

Controller-B

ACL version 2.28

User's Manual

Catalog #100065 Rev.B



Copyright © 1995, 1999 by Eshed Robotec

Catalog #100065 Rev.B

(May 1995) Second Edition (1 2 3 4 5 6 7 8 9 10)

May 1999 - Reprinted / PDF version

Every effort has been made to make this book as complete and accurate as possible. However, no warranty of suitability, purpose, or fitness is made or implied. Eshed Robotec is not liable or responsible to any person or entity for loss or damage in connection with or stemming from the use of the software and/or the information contained in this publication.

Eshed Robotec bears no responsibility for errors which may appear in this publication and retains the right to make changes to the software and manual without prior notice.

ACL is a trademark, and **SCORBOT** a registered trademark, of Eshed Robotec (1982) Ltd.

Read this manual thoroughly before attempting to install or operate the controller. If you have any problems during installation or operation, call your agent for assistance.

Save the original carton and all packing material. You may need them later for shipment.

ESHED ROBOTEC INC.
444 East Industrial Park Drive
Manchester, NH 03109 USA
Tel: 1-800-777-6268
Tel: (603) 625-8600
Fax: (603) 625-2137

Table of Contents

CHAPTER 1	General Information	1-1
	Unpacking and Handling	1-2
	Repacking the Controller	1-2
	Acceptance Inspection	1-3
	Controller Specifications	1-4
	Safety	1-7
	Warnings	1-8
CHAPTER 2	Installation and Configuration	2-1
	Computer/Terminal–Controller Installation	2-2
	Power On	2-2
	Controller Configuration	2-4
	Robot–Controller Installation	2-7
	SCORA-ER 14	2-7
	SCORBOT-ER IX and PERFORMER-MK2	2-7
	SCORBOT-ER VII, SCORBOT-ER V and Vplus	2-7
	Peripheral Devices and Equipment	2-8
	Teach Pendant	2-8
	Install the Teach Pendant Mounting Fixture	2-8
	Connect the Teach Pendant (and Adapter Plug)	2-8
	Teach Pendant Not Connected	2-9
	Inputs/Outputs	2-9
	Remote Servo Drivers	2-9
	Other Devices	2-9
	Auxiliary RS232 Channels	2-10
	Printer	2-10
	Remote Emergency Switch	2-10
	Auxiliary User Control Box	2-11
	Robot Gripper	2-12
	Axis-Driven Gripper	2-12
	Output Driven Gripper	2-13
CHAPTER 3	Controller Functions	3-1
	POWER Switch	3-1
	POWER LED	3-1
	MOTORS Switch	3-2
	EMERGENCY Button and LED	3-2
	RESET Switch	3-3
	USER POWER SUPPLY Terminals	3-3
	OUTPUT Terminals	3-3
	INPUT Terminals	3-3
	INPUT and OUTPUT LEDs	3-4

CHAPTER 4	Inputs/Outputs & User Power Supply	4-1
	Optical Isolation	4-1
	Sink/Source	4-2
	Inputs	4-3
	Maintaining Electrical Isolation	4-3
	Overriding the Controller's Electrical Isolation	4-4
	Connecting the Inputs	4-5
	Activating the Inputs	4-5
	Outputs	4-8
	Maintaining Electrical Isolation	4-8
	Overriding the Controller's Electrical Isolation	4-8
	Connecting the Outputs	4-9
	Relay Outputs	4-9
	Open Collector Outputs	4-9
	Activating the Outputs	4-10
	User Power Supplies	4-13
CHAPTER 5	Controller Components	5-1
	Main Board	5-2
	Diver Card Slots	5-2
	CPU Board Slot	5-2
	RS232 Card Slot	5-3
	I/O Switches	5-3
	PLD – Programmable Logic Device	5-3
	Jumpers	5-3
	Power Supply Connector	5-3
	I/O LED Indicator Connector	5-3
	Input/Output Connector	5-3
	DIP Switches	5-3
	LEDs	5-4
	Motor Power Switch Status Connector	5-4
	CPU Board	5-5
	CPU	5-5
	RAM ICs	5-5
	EPROMS	5-6
	PLD – Programmable Logic Device	5-6
	Battery	5-6
	Coprocesor	5-6
	DIP Switches (RAM and EPROM)	5-6
	Jumper (RAM)	5-6
	Input/Output Board	5-7
	Connector to Main Board	5-7
	Connector to Fan	5-7
	Connector to I/Os.	5-7
	Connector to Motor Power Switch	5-7
	Connector to Power Supply Unit	5-7
	Connector to Emergency Switch	5-7
	Connector to Remote Emergency Circuit	5-7
	DIP Switches (Voltage Reference)	5-7

Switch (Voltage Level)	5-8
DIP Switches (Emergency)	5-8
Axis Driver Card	5-9
Fuses	5-9
LEDs	5-10
Jumpers (Encoders)	5-10
Potentiometers	5-10
Connector Card	5-11
Jumpers (RS232)	5-11
Jumper (Emergency)	5-11
Power Supply	5-12
Transformer	5-12
Diode Bridge	5-13
Capacitor	5-13
Transformer	5-13
Power Supply Circuit Fuses	5-13
Auxiliary RS232 Communication Card	5-14
Jumper (RS232)	5-14

CHAPTER 6 Maintenance and Repairs 6-1

Troubleshooting	6-1
Repair and Upgrade Procedures	6-6
Changing the Voltage Setting	6-6
Opening the Controller Cover	6-6
Replacing Fuses	6-6
AC Power Fuses	6-7
Power Supply Fuses	6-7
Driver Card Fuses	6-7
Removing/Replacing the Battery	6-7
Adding RS232 Channels	6-8
Replacing a Driver Card	6-9
Adding a Driver Card	6-9
Adjusting Driver Card Current Limit	6-10
Expanding RAM	6-10
Upgrading the PLD	6-11

CHAPTER 7 Operation 7-1

Auto Mode	7-1
Teach Mode	7-1
DIRECT Mode	7-1
Manual Keyboard Mode	7-1
Using this Manual	7-2
Activating the Sytem	7-3
Homing the Robot and Peripheral Axes	7-3
Coordinate Systems	7-5
Axis Control Groups	7-6
Servo Control	7-6

Moving the Axes	7-8
“Joint” and “XYZ” Movements	7-8
Activating the Gripper	7-10
Setting the Speed	7-11
Defining and Recording Positions	7-12
Relative Positions	7-14
Listing Positions	7-15
Deleting Positions	7-15
Moving to Recorded Position	7-16
Joint Movement	7-16
Linear Movement	7-17
Circular Movement	7-17
Additional TP Commands	7-18
.	7-18

CHAPTER 8 Programming with ACL 8-1

EDIT Mode	8-1
Help	8-1
Creating a Program	8-2
Writing a Program	8-2
Running a Program	8-3
Program Loop	8-3
Displaying Program Lines	8-4
Halting Program Execution	8-4
Suspend the Program	8-4
Abort the Program	8-5
Stop the Program	8-5
Delaying Program Execution	8-6
Variable Programming	8-6
Mathematical and Logical Functions	8-7
Iteration Functions	8-8
Conditional Functions	8-9
Input and Output Programming	8-10
Displaying Input/Output Status	8-10
Inputs	8-10
Outputs	8-11
Activating Output-Driven Devices	8-11
Sample Program: OUT	8-12
Sample Program: INOUT	8-12
Program Directory	8-14
Multi-Tasking	8-14
Displaying Program Status	8-15
Activating a Program from Another Program	8-15
Simultaneous Execution	8-15
Program Subroutines	8-16
Triggering a Program by I/O	8-16
Saving a Program (Backup) to Disk	8-16
Downloading a Program (Restore) to Controller	8-17

CHAPTER 9	Theory of Control	9-1
	Servo Control	9-1
	Open Loop Control	9-1
	Closed-Loop Control	9-1
	Digital Control	9-2
	Transient and Steady State Responses	9-4
	ACL Controller-B Control Process	9-6
	Trajectory Control	9-7
	Paraboloid	9-7
	Sinusoid	9-7
	Path Control	9-7
	Point-to-Point Control	9-8
	Continuous Path Control	9-8
	The Control Parameters	9-9
	Proportional Control	9-9
	Differential Control	9-10
	Integral Control	9-10
	Proportional–Integral–Differential Control	9-11
	Changing Parameter Values	9-11
APPENDIX A	Fuses	A-1
APPENDIX B	Configuration Summary	B-1
APPENDIX C	Connector Pin Assignments	C-1
APPENDIX D	Emergency System	D-1
	Emergency Activation	D-1
	Remote Emergency Connections	D-1



CHAPTER 1

General Information

This chapter contains instructions for unpacking and inspecting the **Controller-B**.

This chapter also includes the controller's specifications and important safety guidelines and warnings.

* *Read this chapter carefully before you unpack the controller.*

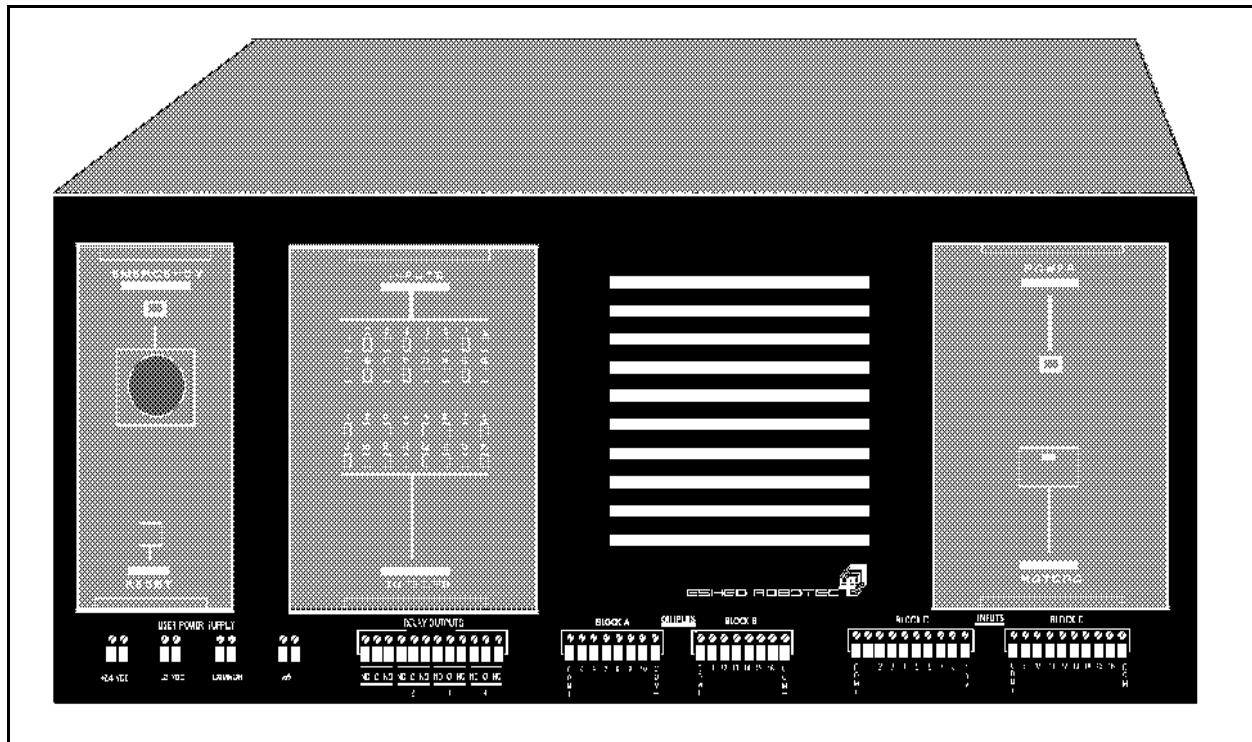


Figure 1-1: ACL Controller-B

Unpacking and Handling

The controller weighs 36 kilos (approximately 80 pounds). Two people are needed in order to lift or move it. Use the handles on the sides of the controller to lift and carry the controller.

Do not grasp the controller on either the front or back panel.

Repacking the Controller

- * *Save the packing materials and shipping carton. You may need them later for shipment or for storage of the controller.*

Be sure all parts are back in place before packing the controller.

The controller should be repacked in its original packaging for transport. If the original carton is not available, wrap the controller in plastic or heavy paper. Put the wrapped controller in a strong cardboard box at least 15 cm (about 6 inches) longer in all three dimensions than the controller. Fill the box equally around the unit with resilient packing material (shredded paper, bubble pack, expanded foam chunks).

Seal the carton securely with strapping or packing tape. Do not use cellophane or masking tape.

Acceptance Inspection

After removing the controller from the shipping carton, examine it for signs of shipping damage. If any damage is evident, do not install or operate the controller. Notify your freight carrier and begin appropriate claims procedures.

The following items are standard components in the **Controller-B** package. Make sure you have received all the items listed on the shipment's packing list. If anything is missing, contact your supplier.

Standard Controller-B Package	
Controller 100/110/220/240VAC	4 driver cards (for 8 axes); (Optional: 3 driver cards for 6 axes)
Cables	Power Cable 100/110/220/240VAC
	RS232 Cable
Emergency By-Pass Plug	Required when TP not connected
ATS (Advanced Terminal Software) Diskette	2 diskettes; one is write-protected
Documentation	<i>Controller-B User's Manual</i>
	<i>ACL Reference Guide</i>
	<i>ATS Reference Guide</i>

Optional Items		
Item	Description	Catalog No.
Teach Pendant	Supplied with: Mounting fixture; Adapter plug for parallel/serial conversion; <i>Teach Pendant User's Manual</i> .	1704
Additional Axis Driver Cards	Each card serves two axes (up to 12 axes supported per controller). Factory-installed if ordered with controller.	45045
Auxiliary Multiport RS232 Board, Cable and Connectors	Card for additional RS232 channels:	45012
	2-port connector cable	40120
	8-port connector cable	40024

Controller Specifications

The following table gives the specifications of the **Controller-B**.

Controller-B Specifications		
Item	Specification	Notes
Type of Control	Stand-alone Real-time Multi-tasking PID (proportional, integral, differential) PWM (pulse width modulation)	Terminal or PC required only for programming stage.
Number of Servo Axes	Maximum: 12	
Groups of Control	12 axes can be divided into 3 groups: Group A Group B Group C (independent axes)	Each group has independent control. Axis interpolation in groups A and B.
Axis Driver Cards	PWM H-bridge drivers 33 KHz, 7 A standard; 10A optional 33-42V (depending on input voltage and load) with tacho feedback	
Path Control	CP (continuous path) Joint Linear Circular Spline User-defined path Tool coordinates	10 ms control cycle parameter. Software controlled acceleration/deceleration. PID and tacho parameters.
Trajectory Control	Paraboloid Sinusoid Open Loop (not for user)	
Speed Control	Speed Travel time	Speed programmed as a percentage of range.
Control Parameters	Servo control Speed, velocity profile, smoothing Axis position error Gripper operation Thermic, impact, limit protection Homing Encoder interface Cartesian calculations	
Power Requirements	100/110/220/240V AC, 50/60Hz, 1500W max.	

Controller-B Specifications		
Item	Specification	Notes
Internal Power Supplies	Servo: 33–42V, 22A (depending on input voltage and load) Digital: 5V/12V/24V User: 12V 2A each (floating); 24V 1.5A each (floating)	
Weight	36kg (approx. 80 lbs)	
Dimensions	479mm (17") L 490mm (19.3") W 206mm (8.1") H	
Ambient Operating Temperature	2°–40°C (36°–104°F)	
CPU	Motorola 68020 Motorola FPU – MC68881	
EPROM	512KB; (expandable to 2MB: future)	
Battery backed-up CMOS RAM	512KB; (expandable to 2MB: future) Approximately 150KB reserved for system Approximately 350KB available for user	
Communication	2 integrated RS232 channels; expandable to 10; Bi-directional parallel port for printer (or fast communication: not yet supported)	
Inputs/Outputs	16 inputs: 2 floating blocks (C and D) 30V maximum Sink/source selectable	
	16 outputs: 4 relays; 12 open collectors in 2 floating blocks (A and B) Sink/source selectable	
Internal BUS	32 bit data, 24 bit address	
Programming Languages	ACL : Advanced Control Language	Using any terminal Using PC with ATS
	SCORBASE Level 5 Software	Using PC
Position Teaching	ACL SCORBASE Teach Pendant	Absolute Relative Cartesian Joint
Multi-tasking	Maximum simultaneous execution: 40 user tasks	
Positioning System	Incremental optical encoders with index pulse (C-pulse)	

Controller-B Specifications		
Item	Specification	Notes
Coordinate System	XYZ coordinates Joint coordinates	
LED Indicators	Main power Inputs/Outputs Servo Power Emergency	On front panel
	Axis power	On rear panel
Safety Features	Emergency switch Motor power switch	On front panel
	Adjustable current limit Short-circuit protection	On all axes
	Thermic, impact and limit software protection Hardware “watchdog” to guard against software/hardware faults.	
Expansion Options	Connector for remote servo drivers; 8-port RS232 card; BUS support for smart peripherals (future)	
Connectors	Inputs/Outputs User power supply	Terminals on front panel
	Axis drivers RS232 channels Teach Pendant Parallel Port Inputs/Outputs, User Power Supply, Emergency Remote Emergency Switch Robot Encoders (ER IX, ER 14, MK2) Robot Power (ER IX, ER 14, MK2) Robot SCORBOT-ER V/SCORBOT-ER Vplus Remote Power Drivers Auxiliary RS232 channels (optional)	On rear panel: D9 connectors D9 connectors D25 connector D25 connector Centronics connectors Terminal block D37 connector 19-pin round connector D50 connector D50 connector D37 connector
Teach Pendant	25 multi-function keys 4 line LCD display; 20 characters per line Emergency Stop pushbutton Deadman switch Auto/Teach selector switch, Full control features	Mounted (full features); Hand-held (no program execution).

Safety

1. Turn off the controller's POWER switch before you do any work within the operating range of a connected robot or automated device.
2. Turn off the controller's POWER switch before you connect any inputs or outputs.
3. Turn off the controller's POWER switch **and** disconnect the power cable from the AC power outlet before you remove any AC power fuses. The power supply units contain dangerously high voltages; the power cable must be disconnected to remove possible shock hazard.
4. The power cable must have a ground connection. If your power outlet does not have a safety ground, do not connect the controller.



*The controller is designed to work with a safety ground connection. **Failure to connect the power cable to a grounded outlet could result in electrical shock.***

5. It is strongly recommended that additional safety devices, such as mushroom emergency buttons and warning lights, be installed in your system.



To immediately abort all running programs and stop all axes of motion, do any of the following:

- press the ABORT key on the teach pendant, or
- use the ACL command A <Enter> or <Ctrl>+A, or
- press the controller's red EMERGENCY button, or
- press the teach pendant's red EMERGENCY button.

Warnings

1. Do not operate the **Controller-B** until you have studied this manual thoroughly.
2. Do not install or operate the **Controller-B** under any of the following conditions:
 - Where a safety ground connection does not exist.
 - Where the ambient temperature drops below or exceeds the specified limits.
 - Where exposed to large amounts of dust, dirt, salt, iron powder, or similar substances.
 - Where subject to vibrations or shocks.
 - Where exposed to direct sunlight.
 - Where subject to chemical, oil or water splashes.
 - Where corrosive or flammable gas is present.
 - Where the power line contains voltage spikes, or near any equipment which generates large electrical noises.
3. Do not plug the controller into the AC power outlet before you are sure that its voltage requirement (as seen on the cover of the AC fuse drawer at the rear of the controller) matches your voltage supply.

If the voltage setting does not match your supply, follow the instructions in Chapter 6 for changing the controller's voltage setting.
4. Do not connect voltage to input ports which exceeds the voltage present on the COM+ terminal of the input block.
5. Never connect voltage from a power supply directly to any open collector outputs (terminals 5–16). The open collector outputs must always be connected to a load. Never connect a load to any voltage exceeding 30VDC.
6. Never drive a current of more than 1.5A through the relay outputs.
Never drive a current of more than 150mA through the open collector outputs.

CHAPTER 2

Installation and Configuration

The installation instructions in this chapter refer to Figure 2-1.

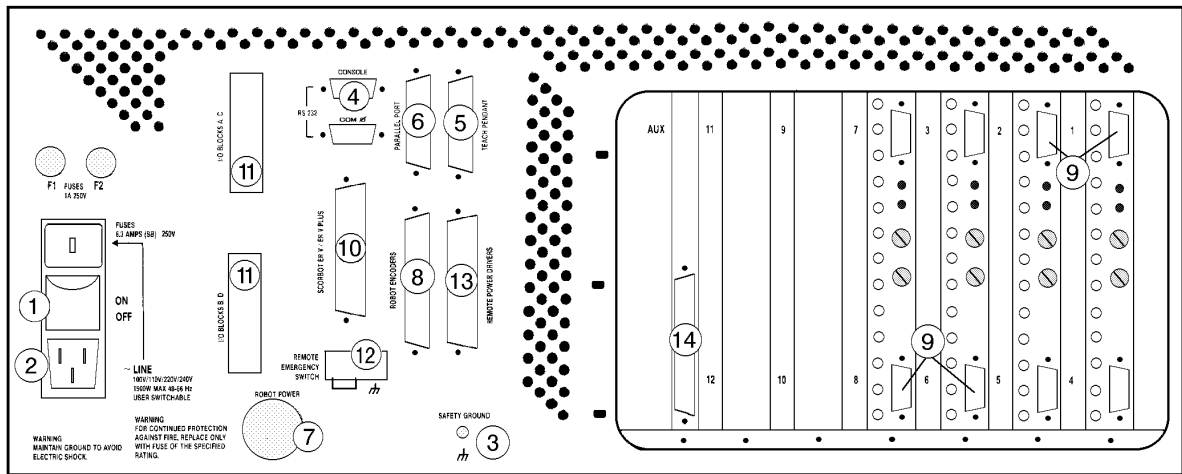


Figure 2-1: Controller-B Back Panel

Legend		
1	Power On/Off Switch	
2	Power Line socket 100/110/220/240VAC	
3	Safety Ground terminal	
4	RS232 channels: Console and COM 0	D9 male
5	Teach Pendant connector	D25 male
6	Parallel Port connector	D25 female
7	Robot Power connector	19-pin round female
8	Robot Encoders connector	D37 male
9	Axis Driver connectors	D9 female
10	SCORBOT-ER V, Vplus connector; RESERVED	D50 male; DO NOT USE
11	I/O Block connectors	36-pin Centronics female
12	Remote Emergency Switch terminal	
13	Remote Servo Driver Command Output connector	D50 male
14	Auxiliary RS232 multipoint (optional)	D37 connector; female

Computer/Terminal–Controller Installation

It is recommended that you connect and check the computer/terminal– controller setup before installing the robot or any other hardware.



Do not connect the controller to the AC power supply until you are sure that the controller's voltage setting matches your voltage supply. If the voltage setting does not match your supply, follow the instructions in Chapter 6 for changing the controller's voltage setting.

1. Make sure the controller power switch (1) is turned off.
2. Connect the AC power cable to the controller (2) and to the AC power supply outlet.
3. Install and configure your computer/terminal and monitor according to the manufacturer's instructions. Make sure the computer's power switch is off.
4. Connect the RS232 cable (D9 connector) to the Console port (4) on the controller and to the RS232 port on the computer/terminal. You may use either COM1 or COM2 on the computer.
5. Connect the computer power cable to an AC power source.
6. If you will *not* be using a teach pendant, connect the Emergency By-Pass plug into the Teach Pendant (D25 connector) port (5).

If you will be using a teach pendant, connect it at this time. Follow the instructions for installing the teach pendant later in this chapter.

It is recommended that you set the teach pendant Auto/Teach switch to AUTO before you power on the system.

7. You may now turn on the system. Follow the instructions in the next section, "Power On."

Power On

1. Once you have made all the required hardware connections, you can power on the controller.
 - Turn on the controller's POWER switch.
 - Turn on the controller's MOTOR power switch.

The green (power and motor) LEDs light up.

2. Turn on your computer, and boot using your own DOS.
3. Insert the **ATS** diskette into one of the computer disk drives, or copy the files from the diskette to the hard drive. Make that drive the default.

4. At the DOS prompt, activate **ATS**.

If the controller is connected to computer port COM1 (default), type:

```
go <Enter>      or  term_acl <Enter>
```

If the controller is connected to computer port COM2, type:

```
go /c2 <Enter>  or  term_acl /2 <Enter>
```

5. Once the software has loaded, the following will appear on your monitor:

```
Advanced Terminal Software for Controller-B version 2.0 (C) ESHED ROBOTEC

---- ROM TEST COMPLETE

SYSTEM READY !
>

<Shift+F10> Backup , <Shift+F8> Print , <Shift+F9> Exit , <Alt+H> Help
1cont↓ 2coff↓ 3home↓ 4run  5move  6move↓ 7teach 8here  9dir↓ 0edit
```

If the prompt does not appear, do one of the following:

- Press <Enter>.
- Make sure the controller is turned on.
- Refer to the section, “Troubleshooting,” in the chapter, “Maintenance.”

The teach pendant display will show:

```
AUTO MODE
```

If AUTO MODE is not displayed, make sure the TP Auto/Teach switch is set to AUTO, and then from the keyboard, type:

```
auto <Enter>
```

6. Perform the controller configuration, as described below.
7. After you have completed the controller configuration, you may connect other hardware to the controller.

Controller Configuration

This section describes the short-form controller configuration which loads default parameter settings according to your responses to the prompts. The procedure described below should be sufficient for you to begin operating the system.

For complete instructions on controller configuration, refer to the *ATS Reference Guide*, and the command CONFIG in the *ACL Reference Guide*.

- * *If the controller has already been in operation, be sure to back-up all data before initiating this configuration procedure.*

The configuration procedure is initiated from the **ATS** main screen by pressing the hot-key combination:

<Ctrl> + <F1>

You are prompted:

```
Controller Configuration
ARE YOU SURE (Y/N)? N
```

Press Y to proceed with the configuration, or
Press N or <Enter> to cancel the configuration.

You are then prompted by a short series of Controller Configuration options.

```
Advanced Terminal Software  version 2.0  (C) ESHED ROBOTEC

Controller Configuration
Robot type : ER-1X / ER-14 / MK2 / OTHER
How many axes are installed (8)? 8
Does the controller have an auxiliary RS232 board (Y/N)? N
Working directory is: c:\ats
Is this correct (Y/N)? Y

WARNING ! USER RAM WILL BE ERASED !!
ARE YOU SURE(Y/N)? N

<Shift+F10> Backup , <Shift+F8> Print , <Shift+F9> Exit , <Alt+H> Help
1con↓ 2coff↓ 3home↓ 4run 5move 6move1 7teach 8here 9dir↓ 0edit
```

- * *Make sure you select the proper options for your installation. Incorrect selections may result in damage to your equipment.*

Robot type: **ER-IX** / ER-14 / MK2 / OTHER

This defines the robot which is connected to the controller.

Use the left and right arrow keys to highlight the name of the robot which is connected to the controller. Then press <Enter> to accept.

Based on your choice of robot (except OTHER), the controller reserves the next available axis for an electrical gripper. For example, when configuring for **SCORA-ER 14**, the gripper axis is defined as axis 5; for **SCORBOT-ER IX** and **PERFORMER-MK2** it is defined as axis 6.

How many axes are installed (8)? ..

This defines the number of axes which can be driven by the controller.

Press <Enter> to accept the default (8), or
Type any other valid number and press <Enter>.

If you have defined the robot type as OTHER, you will also be prompted to define group A:

How many axes in Group A (6)? ..

Press <Enter> to accept the default (6), or
Type any other valid number and press <Enter>.

Does the controller have an auxiliary RS232 board(Y/N)? N

Press Y if the auxiliary multiport RS232 board is installed.
Press N or <Enter> if the board is not installed.

Working directory is: c:\ats

Is this correct (Y/N)?Y

The first time this prompt appears, it shows the DOS directory from which the **ATS** software was activated.

The Working directory must be the directory which contains the parameter files and the **SCORBASE** program file (.CBU files).

If you change the directory definition, it is written to a file named **SETUP.DIR**. Thereafter, whenever **ATS** is loaded, the Working directory is set according to the definition in the **SETUP.DIR** file. Similarly, the **SETUP.DIR** file determines the definition of the Backup directory shown in the Backup Manager screen. **SETUP.DIR** is updated when either the Working directory or Backup directory definition is changed.

Press N if you want to change the directory. The cursor moves to the directory line, prompting you to type and <Enter> a different directory.

Press Y if the directory is correct.

Press <Esc> if you are not sure whether the displayed directory is correct. This will cancel the configuration procedure. Press F10 to access the **ATS** Backup Manager menu to verify the proper directory definition. Or exit to DOS to verify the location of the .CBU files.

WARNING ! USER RAM WILL BE ERASED !!
ARE YOU SURE (Y/N)? N

Press Y to proceed with the configuration.

Press N or <Enter> to cancel the configuration.

After you confirm, **ATS** compares your selections with the controller's current configuration. You are warned of any differences, and again prompted to confirm the configuration.

After you again confirm, **ATS** performs the configuration and loads the proper parameter files in accordance with your selections.

As soon as the configuration is completed, the Peripheral Setup screen appears; this screen is used for defining the devices which are connected to the controller and for loading the parameters for these devices. (The relevant menus in the Peripheral Setup are described later in this chapter, in the sections which provide installation instructions for a gripper and for an auxiliary user control box .)

For definitions not included in the short-form controller configuration procedure—such as axes in control group C, a robot of another make, and memory allocation—you will need to use the **ACL** command CONFIG.

Robot–Controller Installation

After you have completed the computer/terminal setup and verified that the system is functioning, you may proceed with the robot installation.

Perform the robot hardware installation according to the procedure detailed in the chapter, “Installation,” in your robot’s *User’s Manual*.

Special instructions for specific robots are given below:

SCORA-ER 14

When the controller is configured for **SCORA-ER 14** operation, you may connect peripheral axes to any axis connector **except** those labeled 1, 2, 3 and 4, which are reserved for the robot axes.

In addition, **axis 5** is reserved by default for an electrical gripper. If an electrical gripper will not be used, axis 5 can be configured for a peripheral axis by means of the **ACL** command CONFIG.

SCORBOT-ER IX and PERFORMER-MK2

When the controller is configured for **SCORBOT-ER IX** or **PERFORMER-MK2** operation, you may connect peripheral axes to any axis connector except those labeled 1, 2, 3, 4 and 5, which are reserved for the robot axes.

In addition, **axis 6** is reserved by default for an electrical gripper. If an electrical gripper will not be used, axis 6 can be configured for a peripheral axis by means of the **ACL** command CONFIG.

SCORBOT-ER VII, SCORBOT-ER V and Vplus

* *Controller-B does not support SCORBOT-ER VII, SCORBOT-ER V or SCORBOT-ER Vplus.*

Do not attempt to connect these robots to Controller-B !

Peripheral Devices and Equipment

- * *The controller must be turned off before you connect any devices.*

Teach Pendant

- * If you purchased and received the teach pendant together with the **Controller-B**, follow the installation instructions detailed below.
- * If the teach pendant was purchased separately, it was supplied with the “EPROM Upgrade Kit.” To install, you must follow the installation/ upgrade instructions in the appendix of the *Teach Pendant for Controller-B User’s Manual*.

Install the Teach Pendant Mounting Fixture

This special fixture is required for reasons of safety.

Install this fixture safely outside the working range of the robot and peripheral axes.

The fixture has two magnetic strips which activate two magnetic switches on the teach pendant. When the teach pendant is mounted in this fixture, programs can be executed from the teach pendant; when the teach pendant is hand-held, program execution cannot be activated from the teach pendant.

The fixture for mounting the teach pendant is supplied with a set of screws for mounting.

Connect the Teach Pendant (and Adapter Plug)

If you do not intend to connect the teach pendant, skip this section, and go to the following section, “Teach Pendant Not Connected.”

The teach pendant is supplied together with a D25 adapter plug. This adapter enables communication between the teach pendant and the controller; it converts teach pendant serial data to parallel, and controller parallel data to serial.

1. Connect this adapter plug to the port marked Teach Pendant on the controller’s rear panel.
2. Tighten the retaining screws.
3. Connect the connector on the teach pendant cable to the adapter plug.

Teach Pendant Not Connected

- * *When the **Controller-B** teach pendant is not connected to the controller, the Emergency By-Pass Plug must be connected instead.*

This plug prevents the controller from activating EMERGENCY status when the teach pendant is absent.

1. Connect the by-pass plug to the controller's Teach Pendant port.
2. Tighten the retaining screws.

Inputs/Outputs

In addition to connecting I/Os through the terminals on the controller front panel, the I/O Centronics connectors (11) at the rear of the controller may be used to connect external I/O devices directly or by means of Eshed Robotec's **I/O Box**.

Remote Servo Drivers

Controller-B can be used to support and control external servo amplifiers which drive AC motors, DC brushless motors and high power DC motors.

Control of the external drivers is achieved by means of bipolar DC voltage from the controller's REMOTE SERVO DRIVER port (13).

To connect an external DC/AC servo amplifier to **Controller-B**, use the controller's REMOTE SERVO DRIVER port (13).

For encoder signal inputs, use the D9 axis driver ports (9) or the Robot Encoders port (8).

Other Devices

The following items may be used with **Controller-B**, providing these devices are equipped with the proper encoder (contains PC510 or PC510C circuit). These accessories are configured as peripheral axes and connected to the corresponding axis driver ports :

- Linear Slidebases: LSB 20"/24V; LSB 1.0m/24V; LSB 1.5m/24V
- Black Rotary Table 24V
- Gray Conveyor 24V
- XY-Table 24V
- ASRS Carousel
- Motor Kit B

- * *Do not connect any other devices which were previously purchased and operated by means of the original **Controller-A**.*

Auxiliary RS232 Channels

If an auxiliary RS232 communication card (14) has been factory-installed in the controller, be sure the controller is configured accordingly, as described earlier in this chapter.

Make the following cable connections:

- Connect the cable's D37 connector into the Auxiliary RS232 port (14) on the controller.
- Connect the cable's D25 connectors to the corresponding COM ports on the other controllers or computers. The cable will have either two or eight D25 connectors.
- Tighten the retaining screws on the connectors.

If you wish to install an auxiliary RS232 communication card in your controller, follow the instructions described in the section, "Upgrade and Repair Procedures," in Chapter 6.

Printer

To connect a printer to the controller, plug the D25 connector on the printer cable into the controller's Parallel Port (6).

Tighten the retaining screws on the connector.

Remote Emergency Switch

A remote emergency switch (such as a mushroom button) can be connected to the controller. This switch may be incorporated in the user control box (see following section).

- Loosen the screws and remove the wire which shorts the two poles of the Remote Emergency Switch terminal (12) at the rear of the controller.
- Connect the wires of the emergency switch to the Remote Emergency Switch terminal, and tighten the screws.

* *The Remote Emergency Switch terminal is normally closed.*

Remote Emergency outputs also exist between pins 1 and 19 on the I/O connectors (11) on the controller rear panel.

When connecting an emergency switch or device by means of the I/O block connectors, Jumpers JP1 and JP2 on the I/O board must be removed.

Refer to the description of the Emergency button in Chapter 3, and to Appendix D, "Emergency System."

Auxiliary User Control Box

The **ACL** software for **Controller-B** allows you to construct and use an auxiliary control box with the following functions:

- **HOLD/RUN Selector switch**
 - When set to HOLD, axis motion is halted and program execution is suspended.
 - When set to RUN, software program execution and axis motion resume.
- **START Push button**

When pressed, this button will activate an **ACL** user program named START.

Connect these switches to the controller's inputs, as described in the section above, "Inputs/Outputs."

To configure the controller for use with the auxiliary control box, initiate the Peripheral Setup procedure from the **ATS** main screen by pressing the hot-key combination:

<Ctrl> + <F3>

The Control Box Setting menu in the lower right part of the screen allows you to define the I/O settings for the auxiliary user control box.

```
CONTROLLER B - PERIPHERAL SETUP
GROUP A  Axis 1 : SCORBOT-ER IX
GROUP A  Axis 2 : SCORBOT-ER IX
GROUP A  Axis 3 : SCORBOT-ER IX
GROUP A  Axis 4 : SCORBOT-ER IX
GROUP A  Axis 5 : SCORBOT-ER IX
GRIPPER  Axis 6 : Axis-driven Gripper
GROUP B  Axis 7 : XY table axis X (lower)  Hard Home
GROUP B  Axis 8 : XY table axis Y (upper)  Hard Home
GROUP B  Axis 9 : Rotary table (B)        Microswitch
GROUP C  Axis 10 : Speed controlled conveyor belt
          Axis 11 : Not Configured
          Axis 12 : Not Configured

Gripper Connection: None / IO / Axis 6
DC servo gripper

Control Box Setting:
Run/Hold switch:  input # 0
Start push button: input # 0

F1-Change connection  F5-Restore from file  F8-Update controller

<Esc> Exit
```



Be sure your selections are in accordance with the actual installation.

An input defined as 0 indicates that the switch or button is not connected to an input.

Bring the cursor to the input connection option. Type in the number of the input to which the switch or button is connected, from 1 through 16, and press <Enter>.

Press **F8** to accept the definitions, and update the controller parameters.

For more information, refer to the controller configuration procedures in the *ATS Reference Guide*.

Robot Gripper

Install the gripper on your robot according to the instructions supplied with the gripper, or provided in your robot's *User's Manual*.

To configure the controller for use with either an electrical or pneumatic gripper, initiate the Peripheral Setup procedure from the **ATS** main screen by pressing the hot-key combination:

<Ctrl> + <F3>

The short menu in the lower left part of the screen allows you to define the type of gripper used in the robotic system.

"Not Connected" initially appears on the gripper axis line, and a gripper connection option line is also displayed on the screen. The option line permits you to define whether the gripper is axis-driven or output-driven.

Axis-Driven Gripper

Gripper Connection: None / IO / **Axis 6**

To complete the definition of an axis-driven gripper, use the arrow key to highlight the **Axis #** option, and press <Enter>.

A Gripper Type window opens, showing the available types of axis-driven grippers.

Use the arrow key to highlight your selection, and press <Enter> to accept. The window will close, and the selected gripper is displayed below the Gripper Connection option line.

To change the type of gripper, bring the cursor to the line displaying the gripper connection options, and press **F1**. The Gripper Type window will again open, allowing you to change the gripper definition.

Press **F8** to accept the definitions, and update the controller parameters.

For more information, refer to the controller configuration procedures in the *ATS Reference Guide*.

Output Driven Gripper

Gripper Connection: None / **IO** / Axis 6

Configuring the controller for an output-driven gripper causes a pneumatic gripper to respond to the **ACL** commands OPEN and CLOSE and to the teach pendant commands Open Gripper and Close Gripper.

To complete the gripper definition for an output-driven gripper, use the arrow key to highlight the **IO** option, and press <Enter>.

The following option lines will appear, in sequence:

Gripper Connected to Output # **0**

Type in the number of the output to which the gripper is connected, from 1 through 16, and press <Enter>.

State of output for closed gripper **0** / 1

Select 0 for an output OFF signal to close gripper.

Select 1 for an output ON signal to close gripper.

Gripper time delay (1/100 sec.) **80**

This option determines the time required to execute a CLOSE or OPEN gripper command. Press <Enter> to accept the default, or enter a new value for this option.

Press **F8** to accept the definitions, and update the controller parameters.

For more information, refer to the controller configuration procedures in the *ATS Reference Guide*.



CHAPTER 3

Controller Functions

The front panel of the controller contains switches, LEDs and connection terminals for operator use.

The descriptions in this chapter refer to Figure 3-1.

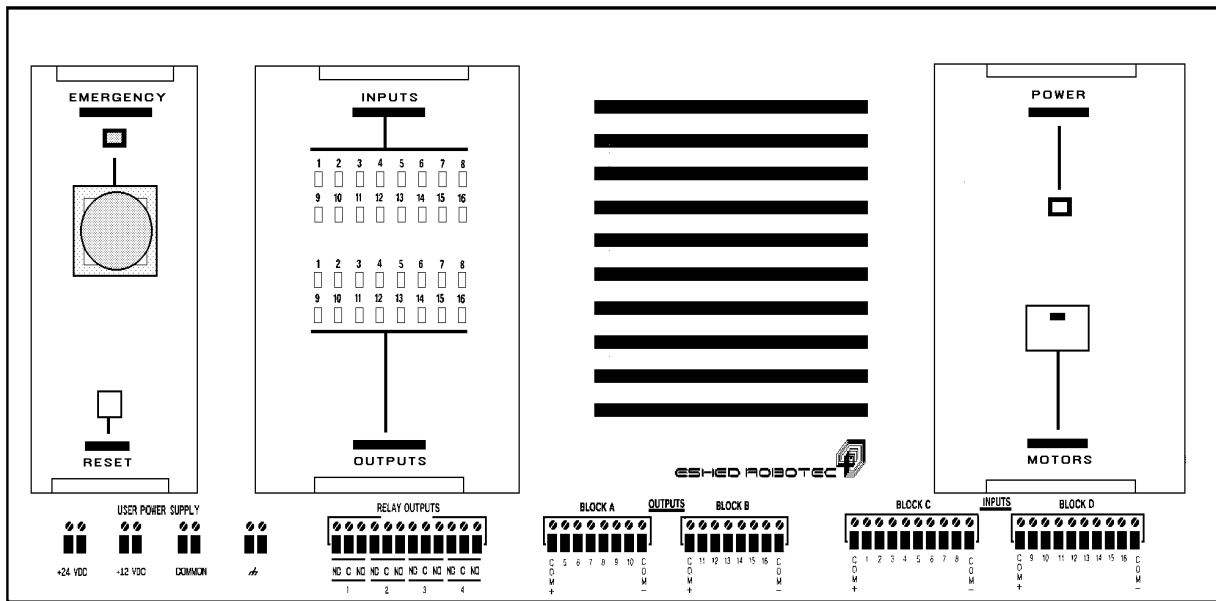


Figure 3-1: Controller-B Front Panel

POWER Switch

This switch is located at the back of the controller.

The POWER switch connects and disconnects 100/110/220/240VAC power to the controller. When the switch is turned on, the system is activated.¹

① POWER LED

This yellow LED lights up when the POWER switch is on. It indicates that power is being supplied to the controller.

② MOTORS Switch

This switch connects and disconnects DC voltage to all the connected motors. A green LED is embedded in the switch, and lights up when the switch is on. The controller's CPU keeps track of the switch's on/off status, and causes the message, "motor power switch is off," to appear when the switch is not depressed.

The MOTORS switch is turned off in the following circumstances:

- To disconnect power to the motors without turning off the controller.
- To prevent possible motion of axes.

When the MOTORS switch is released, any moving axes are brought to a soft stop. Once the MOTORS switch is turned off, the robot motors and all connected axes are unable to move. If the POWER switch remains on, other system components, such as the encoders and I/O's, will continue to function; program execution will also continue, although motor error messages will be displayed.

③ EMERGENCY Button and LED

When the EMERGENCY button is pressed, the following occurs:

- Motor power is disconnected; all motor movement stops; the green MOTORS LED shuts off.
- The COFF (control off) state is activated.
- The red emergency LED lights up.
- An emergency message appears on monitor.
- All running programs (including BACKG) are aborted.
- The inputs and outputs are "frozen" in their current state.
- The user power supply remains in normal operation.
- HOME and CON cannot be activated.
- If it exists, the user program, EMERG, will be executed.

Release the EMERGENCY button by turning it clockwise, as indicated by the arrows on the button.

- When the EMERGENCY button is released, the following occurs:
- An exit emergency message appears on the monitor.
- The red emergency LED shuts off.
- The green MOTORS LED turns on.
- If it exists, the user program BACKG will start running.

To resume normal operation, do the following:

- If necessary, abort the user program EMERG.
- Activate CON.
- Reactivate user programs. (All positions, including Home, remain in memory.)

* *When a **remote emergency switch** is connected to the controller, it functions exactly like the **EMERGENCY** button.*

④ **RESET Switch**

Like the RESET button on a PC computer, pressing this switch restarts the controller, as if it were being turned on.

When the switch is pressed, the following happens:

- All currently running programs are aborted.
- The Home position is erased from memory during the reset.

To resume work following the reset, do the following:

- Home the axes (HOME, HHOME*).
- Reactivate all aborted programs (RUN).

⑤ **USER POWER SUPPLY Terminals**

The USER POWER SUPPLY allows external devices in the user's applications to receive power from the controller. The controller's USER POWER SUPPLY has four terminals:

- +24VDC 2A regulated floating power supply
- +12VDC 2A regulated floating power supply
- Floating Common
- Safety Ground

⑥ **OUTPUT Terminals**

The 16 outputs allow the robotic system to transmit signals to external devices in the robot's environment. The **Controller-B** has two types of outputs: 4 relay outputs and 12 open collector outputs.

Outputs may also be connected by means of the two Centronics 36-pin connector (I/O BLOCK) at the rear of the controller.

⑦ **INPUT Terminals**

The 16 inputs allow the robotic system to receive signals from external devices in the robot's environment.

Inputs may also be connected by means of the two Centronics 36-pin connector (I/O BLOCK) at the rear of the controller.

* *Read Chapter 4 thoroughly before making any output, input or power supply hardware connections.*

⑧ **INPUT and OUTPUT LEDs**

16 green LEDs, corresponding to outputs 1–16, light up when the outputs are on.

16 yellow LEDs, corresponding to inputs 1–16, light up when the inputs are on.

CHAPTER 4

Inputs/Outputs & User Power Supply

Controller-B has 16 inputs and 16 outputs. The IOs are accessible from either the terminals on the controller's front panel or the ports on the controller's rear panel. Two user power supplies are also provided.



Avoid the risk of malfunctions and shorts:

Make sure all wires are connected to either the front terminals or rear ports.

Make sure controller outputs are not connected to each other.

Before describing the specific details of the **Controller-B** inputs and outputs, it is necessary to explain two important features of this controller: optical isolation and sink/source modes.

Optical Isolation

Optical isolators are devices which were designed to solve the problem of interfacing electronic equipment in electrically noisy environments. If a non-isolated direct connection is used, noise entering through the terminals or ground and power lines can disrupt system operation, resulting in the generation of erroneous logic signals. In severely noisy environments, noise may even damage electronic circuitry.

When electrical isolation is maintained, system communication is not affected by common-mode noise.

The optical isolator uses visible or infra-red light to establish communication between local and remote equipment.

The isolator consists of a light emitter on one end and a light detector on the other. When the light emitter side is off (dark), the detector is off. When the light emitter side turns on, the detector changes its state to on.

Sink/Source

The I/O terminals of the **Controller-B** can be configured to work in one of the two modes:

- Source
- Sink

In simple terms, an electrical source is comparable to a water faucet, while an electrical sink is comparable to the tub into which the water flows.

For proper I/O operation, a paired source/sink connection is required, as shown in the following tables:

Controller Input	External Device Output
Source	Sink
Sink	Source

Controller Output	External Device Input
Source	Sink
Sink	Source

To set the sink/source mode of an input/output, set the corresponding switch on the controller’s main board to either the sink or source position.

Figure 4-1 shows the default setup:

- Inputs 1 to 16: Sink
- Outputs 5 to 16: Sink

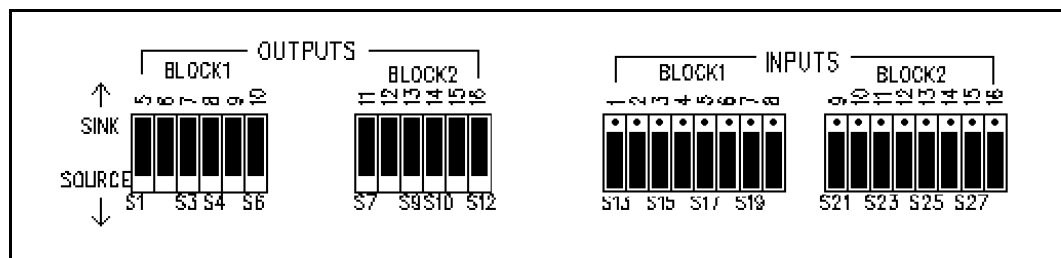


Figure 4-1: Sink/Source Switches—Main Board

Inputs

Controller-B has 16 inputs, which are divided into two optically isolated blocks (Blocks C and D), as shown in Figure 4-2. Each block is floating and can be referenced to a different external device or to the controller's frame ground. In addition, any input can act as a source or as a sink.

The controller input circuits use opto-isolators to ensure proper electrical isolation between the controller and other electronic equipment in the system.

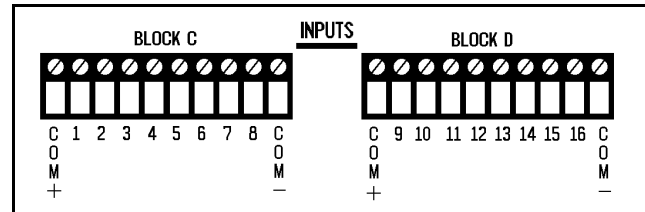


Figure 4-2: Input Terminals on Front Panel

Input terminals are connected to the light emitting side of the opto-couplers. The opto-coupler light detector side is connected to the internal circuits of the controller, and referenced to the controller's ground.

Maintaining Electrical Isolation

To maintain the controller's electrical isolation, do the following:

- Dedicate an isolated input block to one external device.
- Be sure the input block's reference switches are in the EXTERNAL position (default setting). These switches are located on the I/O board, as shown in Figure 4-3/4-3A.
- Connect the COM- block to the external device's ground.
- Connect the COM+ block terminal to the external device's positive voltage terminal.

In this way the light emitter side of the opto-coupler is referenced to the external device's power and ground while the light detector is referenced to the controller's power and ground—and electrical isolation is maintained.



When an input block is set to the EXTERNAL (default) reference mode, do not connect more than one external device to an input block.

Overriding the Controller's Electrical Isolation

If your installation includes an external device which contains opto-isolators on its outputs, you may choose to override the controller's optical isolation to a particular input block. This is accomplished by setting the controller's reference switches to the INTERNAL position, as indicated in Figure 4-3.

Be sure to first disconnect the COM- and COM+ terminals before you change the reference settings to INTERNAL.

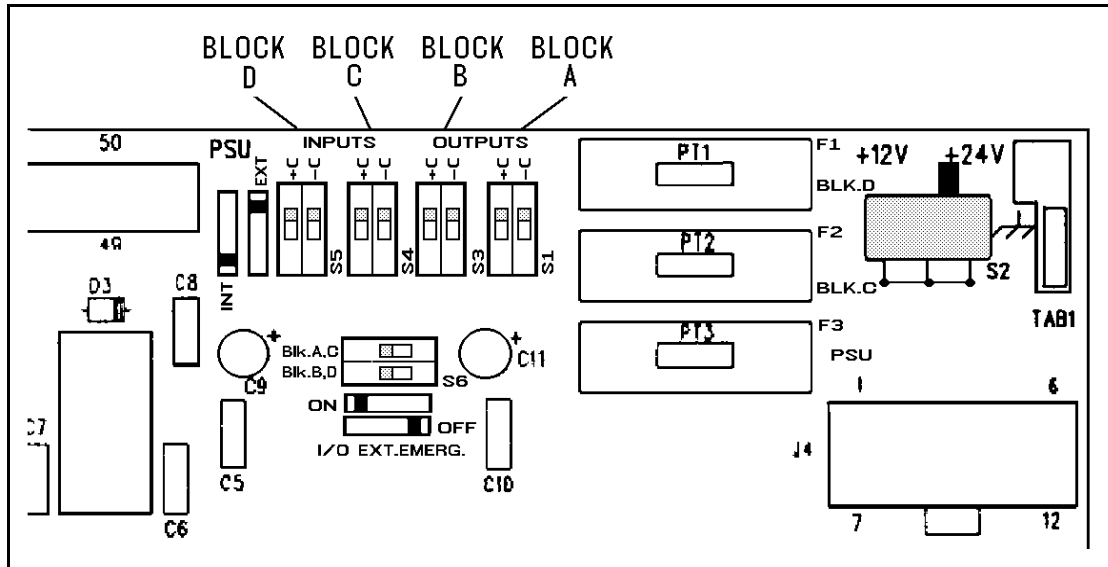


Figure 4-3: Int/Ext Reference Switches -- I/O Board (REV.1)



When an input block is set to the INTERNAL reference mode, do not connect external device power and ground. This will short controller power to device power, and may damage both the controller and the external device.

Connecting the Inputs

Before connecting controller inputs to external device outputs, be sure to check whether the external device output is a sink or a source type output.

Some manufacturers use the terms NPN and PNP.

- An NPN output is a sink-type output.
- A PNP output is a source-type output.

If the external device output is a sink (NPN) type, be sure to set the controller's input switch to its SOURCE position.

If the external device output is a source (PNP) type, be sure to set the controller's input switch to its SINK position.

If the input block switch is set to EXTERNAL reference, make your connections as follows:

Controller	External Device
COM+	Positive supply voltage
Input	Output
COM-	Ground

If you want to set the input block to INTERNAL, be sure to first disconnect the COM- and COM+ terminals.

Activating the Inputs

The inputs are activated by external devices' outputs which are connected to corresponding inputs on the controller.

If the controller's input is configured as SOURCE:

- External voltage of 0VDC to 1.5VDC will turn the input ON.
- External voltage of 3.0VDC to the positive voltage reference (voltage present on the COM+ terminal of the input block) will turn the input OFF.

If the controller's input is configured as SINK:

- External voltage of 3.0VDC to the positive voltage reference (voltage present on the COM+ terminal of the input block) will turn the input ON.
- External voltage of 0VDC to 1.5VDC will turn the input OFF.



Do not connect voltage to input ports which exceeds the voltage present on the COM+ terminal of the input block.

Figures 4-4 through 4-7 show diagrams of the input circuit in both source and sink modes for both external and internal power supply.

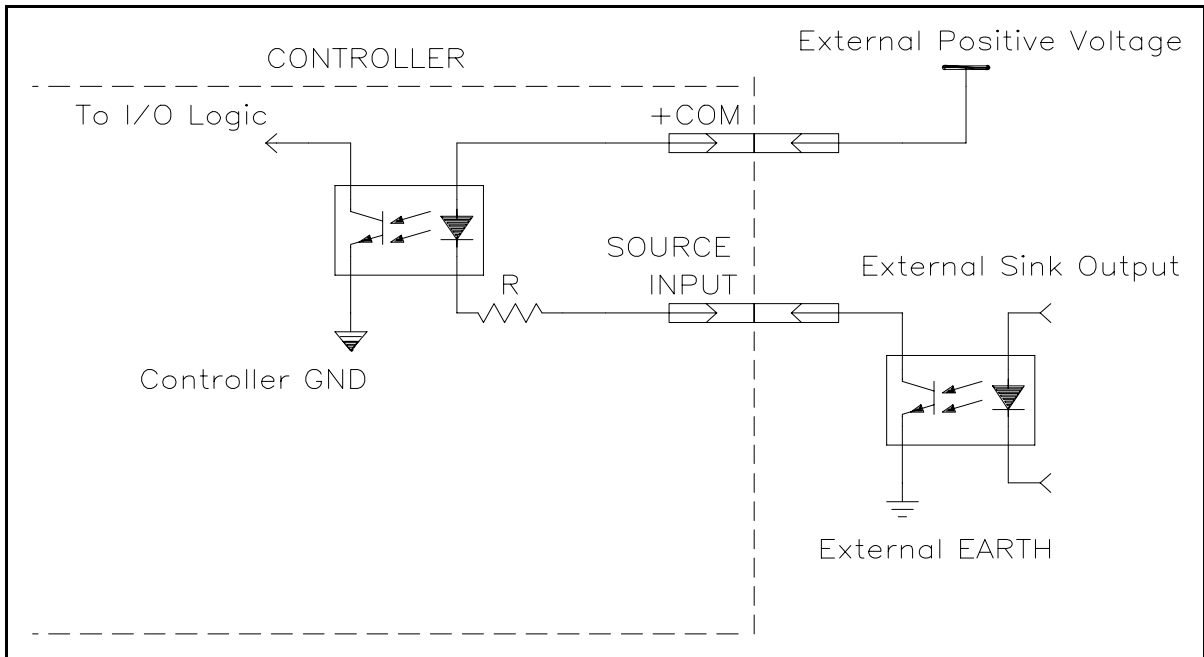


Figure 4-4: Input Circuit (Source) - External PSU

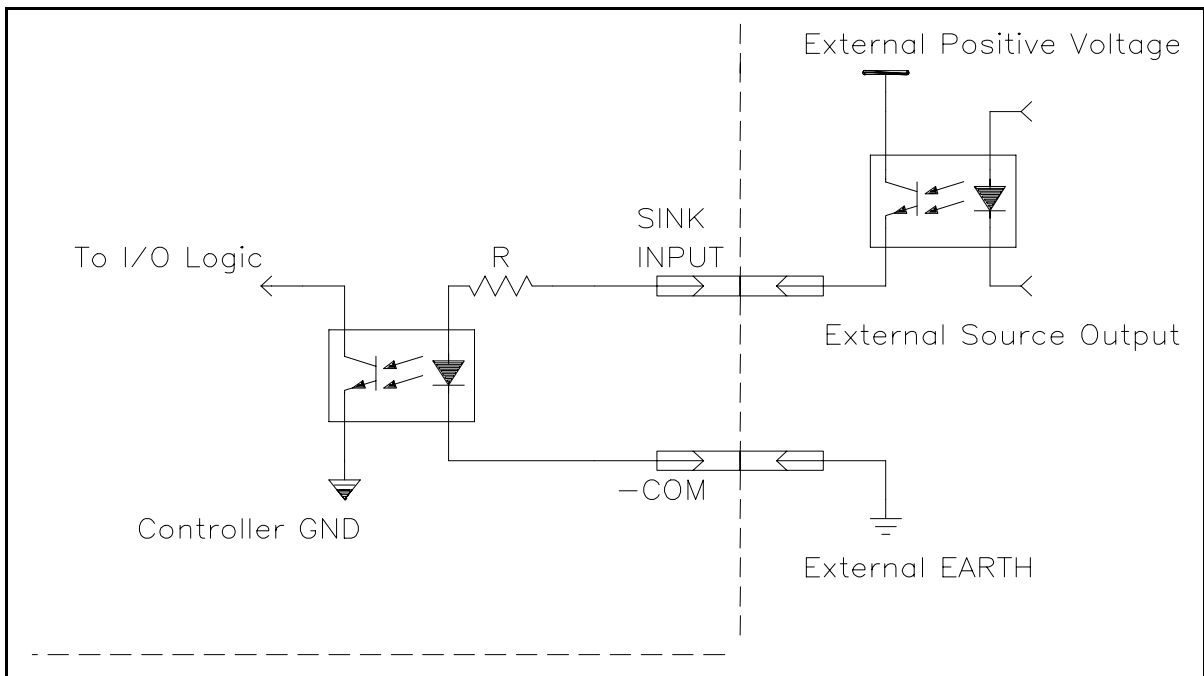


Figure 4-5: Input Circuit (Sink) - External PSU

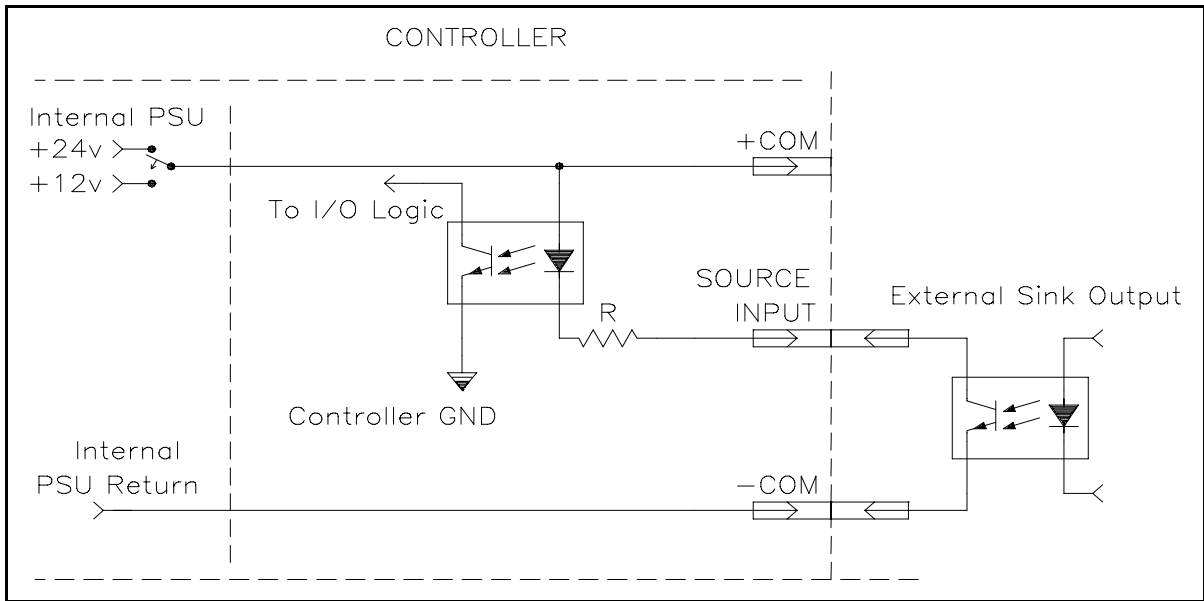


Figure 4-6: Input Circuit (Source) - Internal PSU

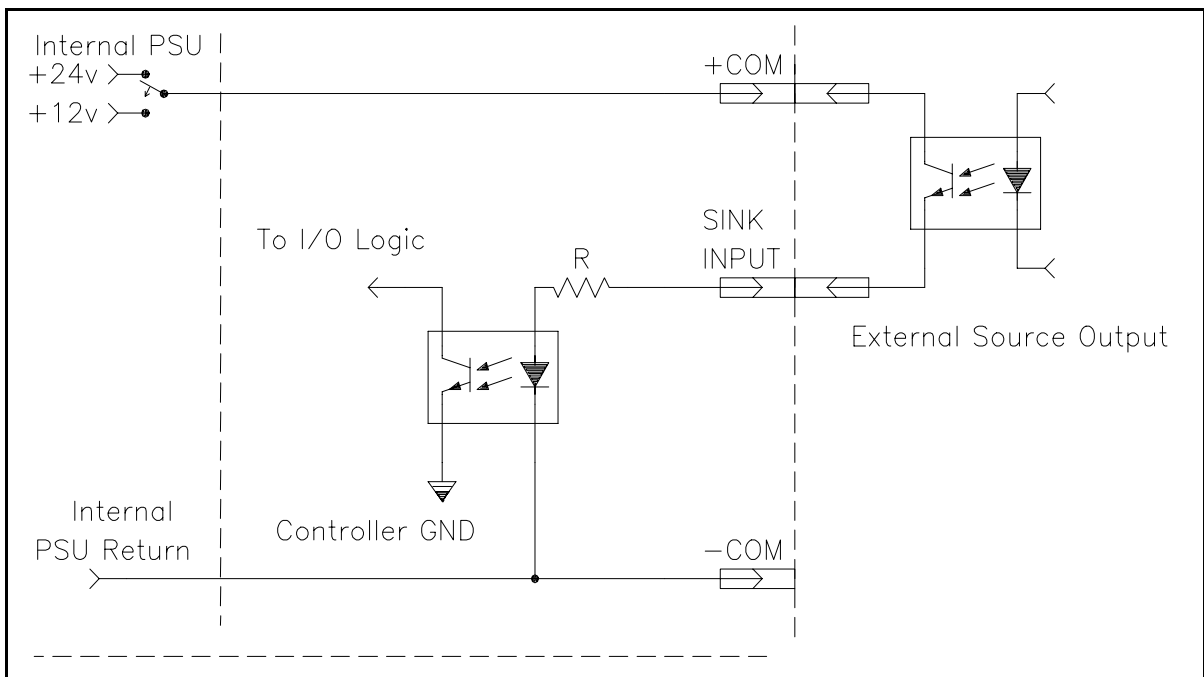


Figure 4-7: Input Circuit (Sink) - Internal PSU

Outputs

Controller-B has 16 outputs. Four are relay outputs with both Normally Open and Normally Closed contacts, as shown in Figure 4-8. The other 12 outputs are open collector types, which are divided into two optically isolated blocks (Blocks A and B), as shown in Figure 4-9. Each block is electrically floating and can be referenced to a different external device or to the controller's frame ground (default: external). In addition, any open collector output can act as a source or as a sink.

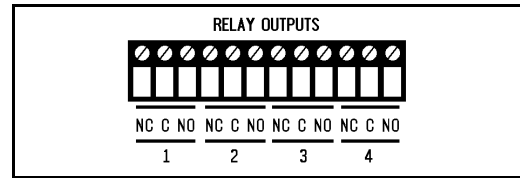


Figure 4-8:
Relay Output Terminals on Front Panel

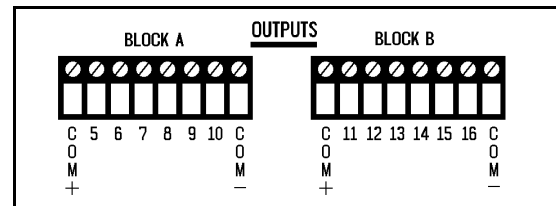


Figure 4-9: Open Collector Output Terminals on Front Panel

Open collector output terminals are connected to the light detector side of opto-couplers and should be referenced to external device power and ground. The opto-coupler light emitting side is connected to the internal circuits of the controller, and referenced to controller's power and ground.

Maintaining Electrical Isolation

To maintain electrical isolation between the controller and an external device, do the following:

- Dedicate an isolated output block to one external device.
- Set the block reference switches to the EXTERNAL position.
- Connect the COM- block terminal to the external device's ground.
- Connect the COM+ block terminal to the external device's positive voltage terminal.



When an output block is set to the EXTERNAL (default) reference mode, do not connect more than one external device to an output block.

Overriding the Controller's Electrical Isolation

If your installation includes an external device which contains opto-isolators on its inputs, you may choose to override the controller's optical isolation to a particular output block. This is accomplished by setting the controller's reference switches to the INTERNAL position, as indicated in Figure 4-3.

Be sure to first disconnect the COM– and COM+ terminals before you change the reference settings to INTERNAL.



When an output block is set to the INTERNAL reference mode, do not connect external device power and ground. This will short controller power to device power, and may damage both the controller and the external device.

When using an open collector output block in its INTERNAL mode, the output voltage level can be set to either +12V or +24V. This is accomplished by setting the switch on the I/O board, as seen in Figure 4-3.

Connecting the Outputs

Before connecting controller outputs to external device inputs, be sure to check whether the external device input is a sink or a source type input.

If the external device input is a sink type, be sure to set the controller's output switch to its SOURCE position.

If the external device input is a source type, be sure to set the controller's output switch to its SINK position.

Relay Outputs

Outputs 1 to 4 include relays in their final stage. Each relay includes the three contact points:

- Common (C)
- Normally Closed (NC)
- Normally Open (NO).

Maximum current allowed: 1.5A



Never drive a current of more than 1.5A through the relay outputs.

Open Collector Outputs

Outputs 5 to 16 include a transistor with an open collector in their final stage. These outputs must be connected to a load; never connect directly to power or ground.

When using an inductive load, such as a solenoid or a relay, connect a reversed biased protection diode across the load. When connecting an open collector output to an input, you can connect directly. Figures 4-4 through 4-7 show the proper I/O connections.

Maximum current allowed: 150mA



Never connect voltage from a power supply directly to any open collector outputs (terminals 5–16). The open collector outputs must always be connected to a load. Never connect a load to any voltage exceeding 30VDC.

Never drive a current of more than 150mA through the open collector outputs.

Activating the Outputs

The outputs are activated by software commands.

The **ACL** command format is as follows:

SET OUT[*n*]=0 Turns output *n* OFF.

SET OUT[*n*]=1 Turns output *n* ON.

Where *n* = 1–16

Figure 4-10 shows the ON and OFF states of the relay outputs.

Figures 4-11 through 4-13 show diagrams of the output circuit in both source and sink modes for both external and internal power supply.

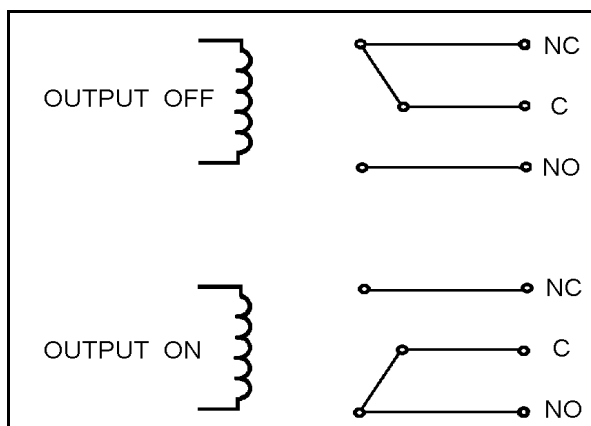


Figure 4-10: Relay Output States

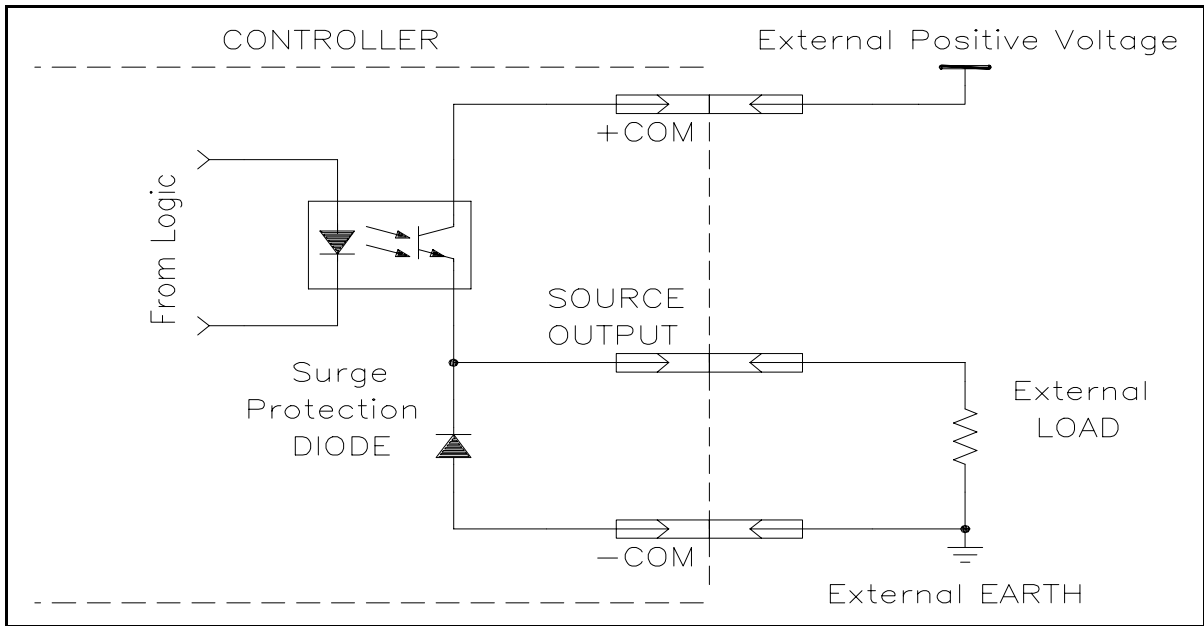


Figure 4-11: Output Circuit (Source) - External PSU

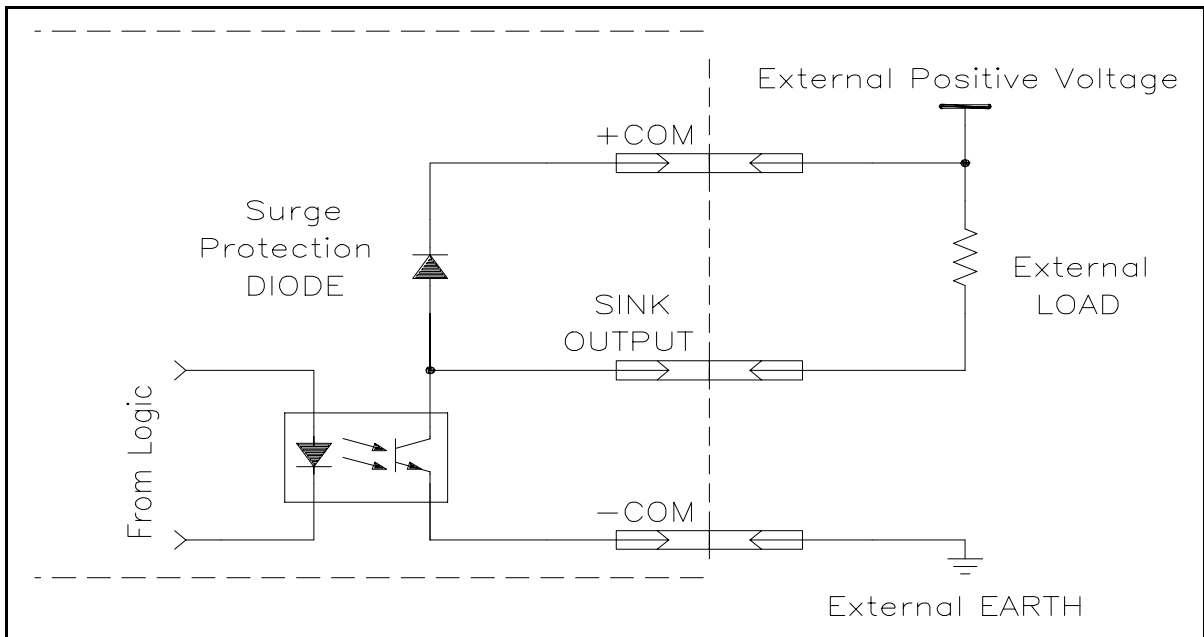


Figure 4-12: Output Circuit (Sink) - External PSU

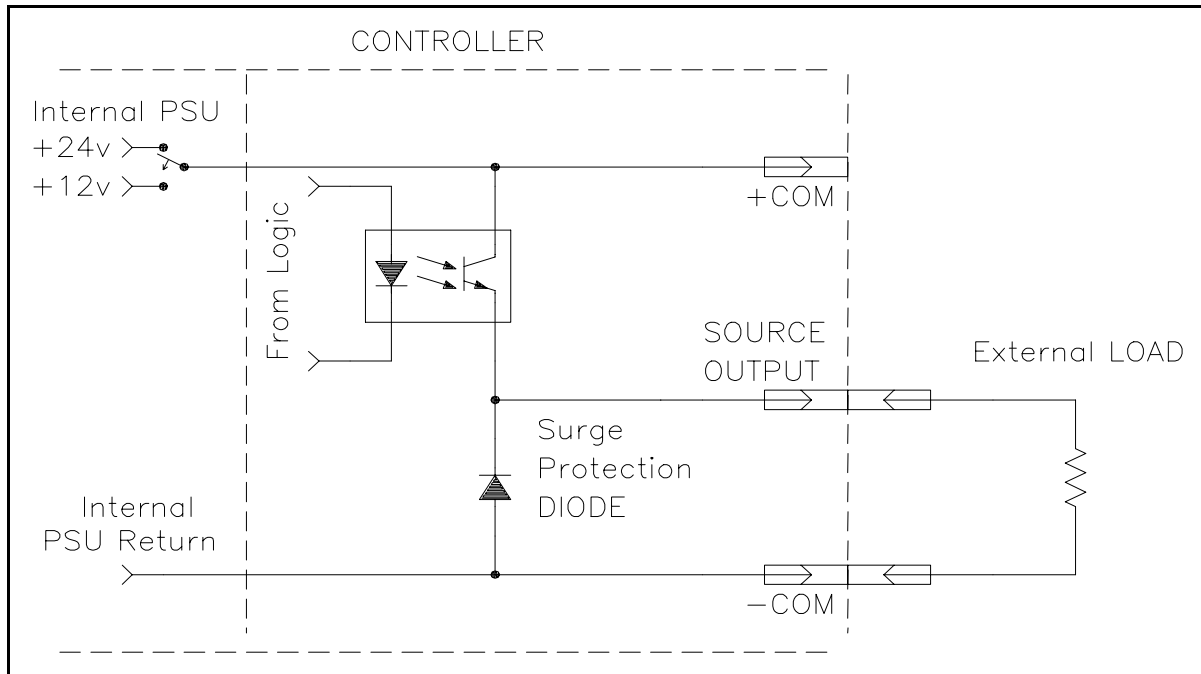


Figure 4-13: Output Circuit (Source) - Internal PSU

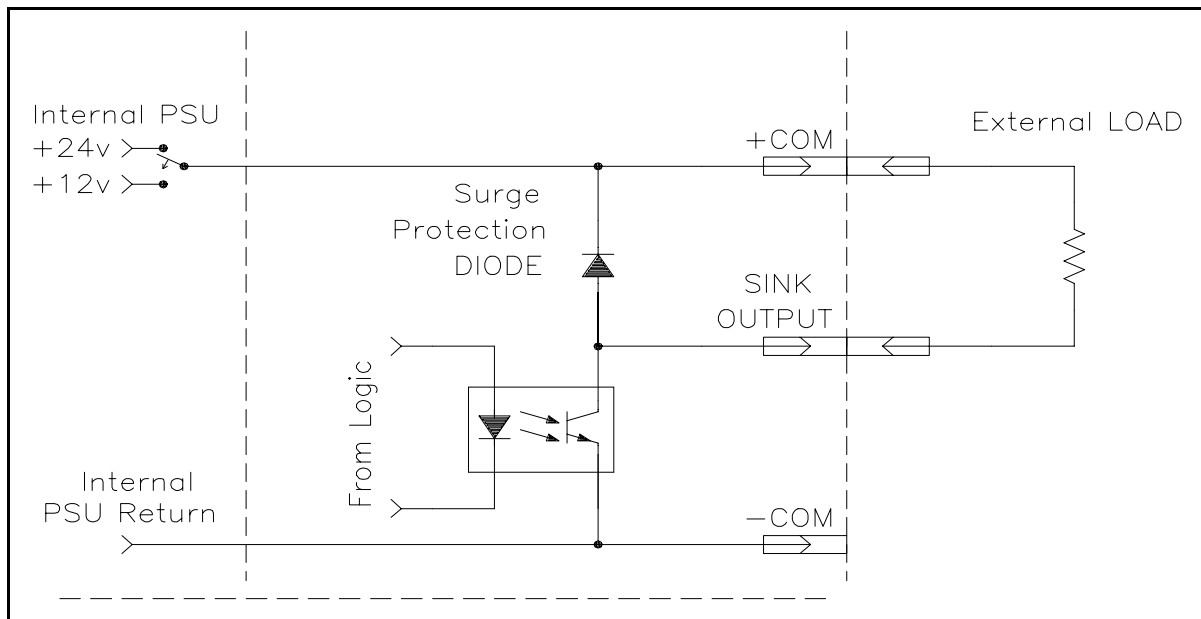


Figure 4-14: Output Circuit (Sink) - Internal PSU

User Power Supplies

Controller-B has two power supplies with floating outputs and common return:

- 24VDC, 1.5A max.
- 12VDC, 2A max.

Frame Ground terminals are connected to earth through the power line cable, and can be connected to the power supply's Common terminal in order to override the floating status of the PSU.

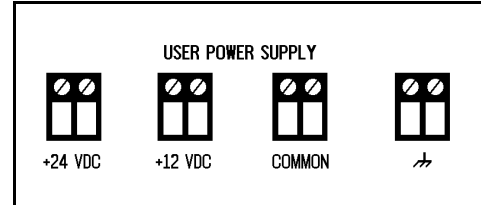


Figure 4-15: User Power Supply Terminals

Setting an output or input block to the INTERNAL reference mode will short the +COM terminal of the block to 24VDC or 12VDC of the user power supply, and will also short the -COM terminal to Common of the user power supply.



CHAPTER 5

Controller Components

This chapter describes the main circuit boards in the **Controller-B**.

The following sections and diagrams describe the most important components on each circuit board. It is recommended that you familiarize yourself with these components. Some of these items may require adjustment or replacement.

The number next to each description indicates the number identifying the item in the accompanying diagram.

Additional information can also be found in other parts of this manual:

- Repair and upgrade procedures are given in Chapter 6.
- Current limits for axis drivers are detailed in Appendix A.
- Jumper and switch configurations are detailed in Appendix B.

Main Board

Refer to Figure 5-1.

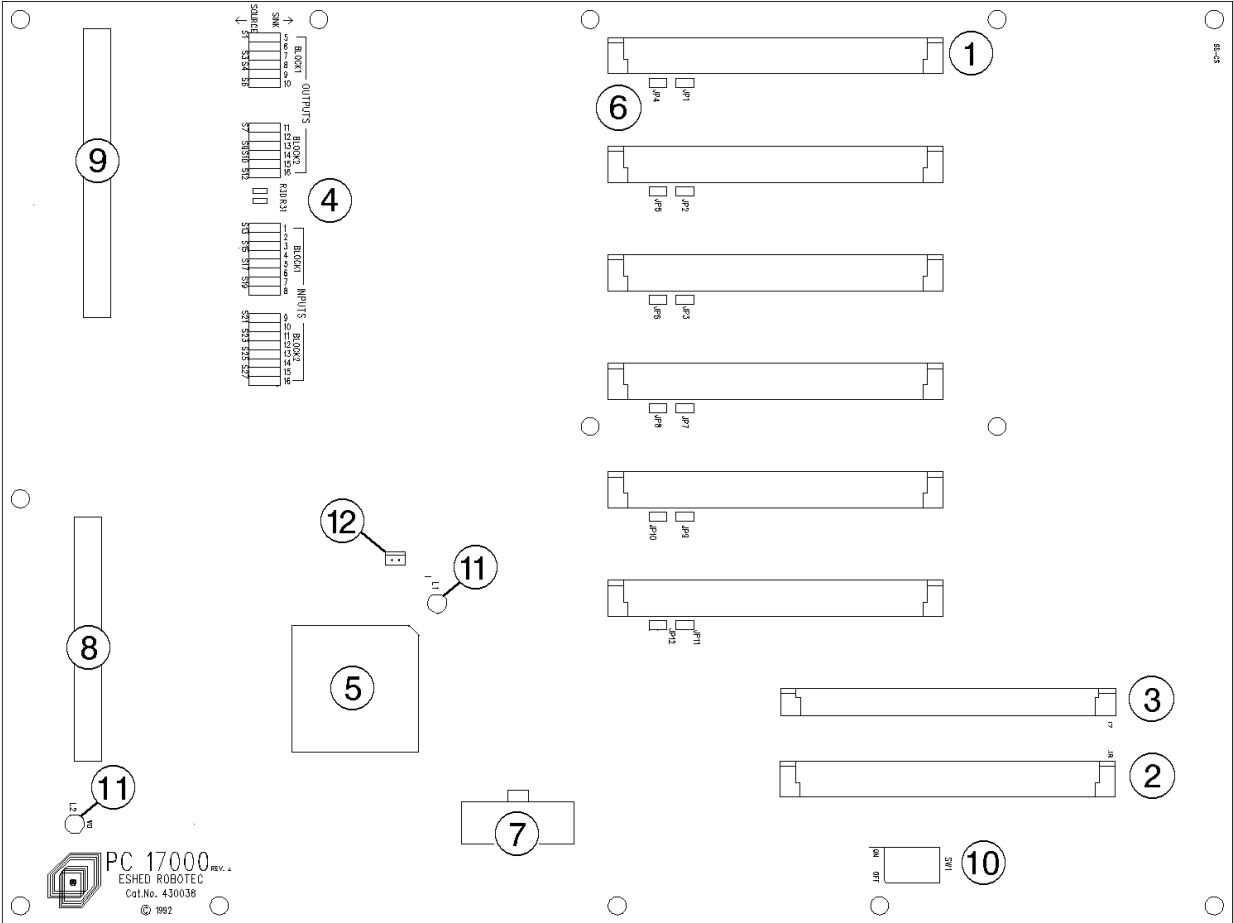


Figure 5-1: Main Board

① **Diver Card Slots**

Six slots for connecting the axis driver cards.

Other cards which are compatible with the controller’s BUS may also be connected.

② **CPU Board Slot**

Slot for connecting the CPU board, which contains CPU, EPROM and battery-backed RAM.

③ **RS232 Card Slot**

Slot for connecting the auxiliary multiport RS232 communication card.
The controller can be equipped with a multiport communication card suited for use in ER-CIM systems and other applications which require additional communication ports.

④ **I/O Switches**

Switches for configuring the sink or source status of each input and open collector output.

Default factory setting:

- Outputs:Sink
- Inputs:Source

⑤ **PLD – Programmable Logic Device**

Performs logic functions equivalent to those of approximately twenty ICs.
Programmed by Eshed Robotec.

⑥ **Jumpers**

Jumpers JP1 through JP12.

Default factory setting: open

Do not alter the status of these switches unless specifically instructed to do so by Eshed Robotec.

⑦ **Power Supply Connector**

Power input from the controller power supply.

⑧ **I/O LED Indicator Connector**

Connector from which flat cable leads to I/O LED indicator circuit.

⑨ **Input/Output Connector**

Connector from which flat cable leads to I/O circuit board.

⑩ **DIP Switches**

Six DIP switches at SW1.

Default factory setting: all switches on.

Do not alter the status of these switches unless specifically instructed to do so by Eshed Robotec.

11 LEDs

Two LEDs on the main board:

- “Watchdog” yellow LED.

Software protection:

The controller protects itself from states it does not recognize; this prevents the axes’ motors from being driven wildly and uncontrollably.

A motor control code which addresses the DAC is executed with regular frequency. If the system fails to address the DAC within a 20ms interval, the motors are inhibited, as indicated by the watchdog LED.

Hardware protection:

The watchdog LED will light up and the controller will shut down power to the motors if any of the following power supply fuses blow out: +12VA, -12VA, +12VDR, -12VDR.

- “Halt” red LED.

In the event of controller hardware failure, the processor will be halted, as indicated by the halt LED.

12 Motor Power Switch Status Connector

Connector from which wires lead to the motor power switch.

CPU Board

Refer to Figure 5-2.

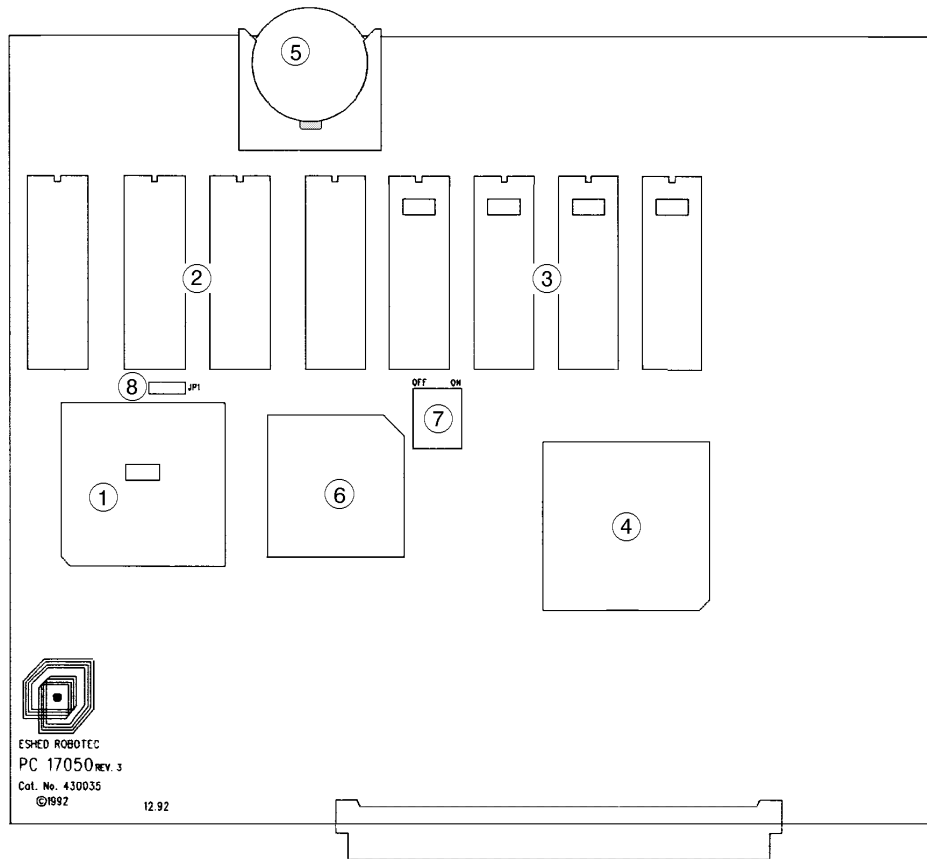


Figure 5-2: CPU Board

① CPU

The central processing unit contains all the circuitry for controlling, timing, performing arithmetic, and addressing data for storage or transmission.

Directs the sequence of operations, interprets the coded instructions, performs arithmetic and logical operations, and initiates the proper commands to the controller circuits for execution. Controls the controller, as directed by the program it is executing.

② RAM ICs

Four battery-backed RAM ICs.

Standard size is 512KB. Expandable to 1MB or 2MB. Contact Eshed Robotec for further details.

③ EPROMS

Four electrically programmable memory ICs, programmed with **ACL**. Standard memory size is 512KBb, including all software.
(Expandable to 1MB or 2MB: not yet available)

To prevent erasure of the EPROM by exposure to UV rays, do not remove the sticker covering the EPROM window.

④ PLD – Programmable Logic Device

Performs logic functions equivalent to those of approximately twenty ICs.
Programmed by Eshed.

⑤ Battery

A 3V lithium battery, for memory back-up. Two-year lifespan.

Replace only with Renata CR2477N 3V battery, available from Eshed Robotec.

⑥ Coprocessor

Assists the CPU in performing fast mathematical calculations. Uses floating point arithmetic.

⑦ DIP Switches (RAM and EPROM)

Switches S1 are used for configuring the size of the RAM and the EPROM.

Default factory settings are shown in bolded typeface.

• RAM

DIP switch S1-1	DIP switch S1-2	RAM Size
ON	ON	512 KB
OFF	ON	1 MB
OFF	OFF	2 MB

• EPROM

DIP switch S1-3	DIP switch S1-4	EPROM Size
ON	ON	512 KB
OFF	ON	1 MB
OFF	OFF	2 MB

⑧ Jumper (RAM)

Jumper JP1 is used for configuring the size of the RAM.

Default factory settings are shown in bolded typeface.

Shorted	RAM Size
Pins 1 and 2	512 KB
Pins 2 and 3	1 MB or 2 MB.

Input/Output Board

Refer to Figure 5-3.

For more information on inputs and outputs, refer to Chapter 4, “Inputs/Outputs.”

- ① **Connector to Main Board**
Connector from which flat cable leads to main board.
- ② **Connector to Fan**
Connector from which flat cable leads to fan.
- ③ **Connector to I/Os.**
Connector from which flat cable leads to I/O connectors at the back of the controller.
- ④ **Connector to Motor Power Switch**
Connector from which flat cable leads to motor power switch.
- ⑤ **Connector to Power Supply Unit**
Connector from which flat cable leads to power supply unit.
- ⑥ **Connector to Emergency Switch**
Connector from which flat cable leads to emergency switch.
- ⑦ **Connector to Remote Emergency Circuit**
Cable from which cable leads to remote emergency circuit (PC17600).
- ⑧ **DIP Switches (Voltage Reference)**
Four switches: S1, S3, S4, S5.
Switches for configuring either INTERNAL or EXTERNAL voltage reference.
Default factory setting: external

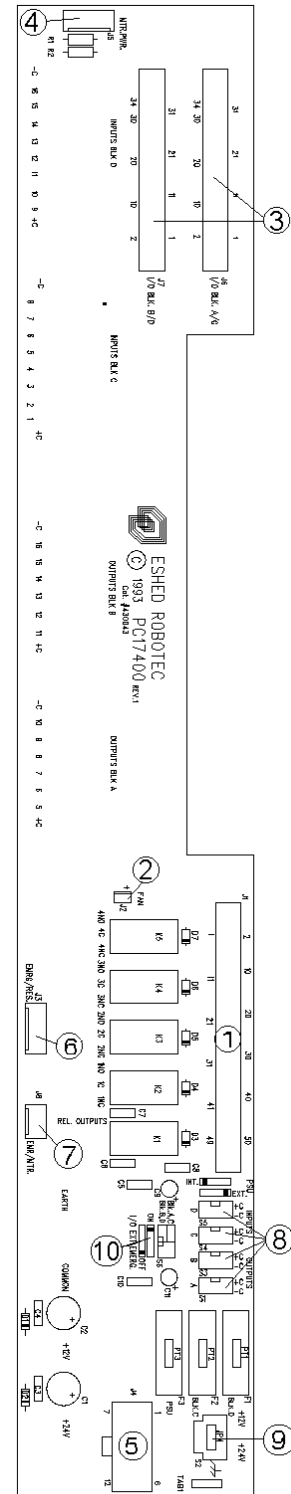


Figure 5-3: Input/Output Board (REV.1)

⑨ **Switch (Voltage Level)**

Switch S2 for configuring the open collector output voltage levels, either +12V or +24V, when the output block is set to the INTERNAL reference.

Default factory setting: 24V

⑩ **DIP Switches (Emergency)**

Two switches: S6-1 and S6-2.

These switches are used for configuring an external emergency switch.

These switches are normally set to ON. They must be set to OFF when connecting remote safety/emergency switches by means of the Centronics I/O connectors on the controller's back panel.

Axis Driver Card

Refer to Figure 5-4.

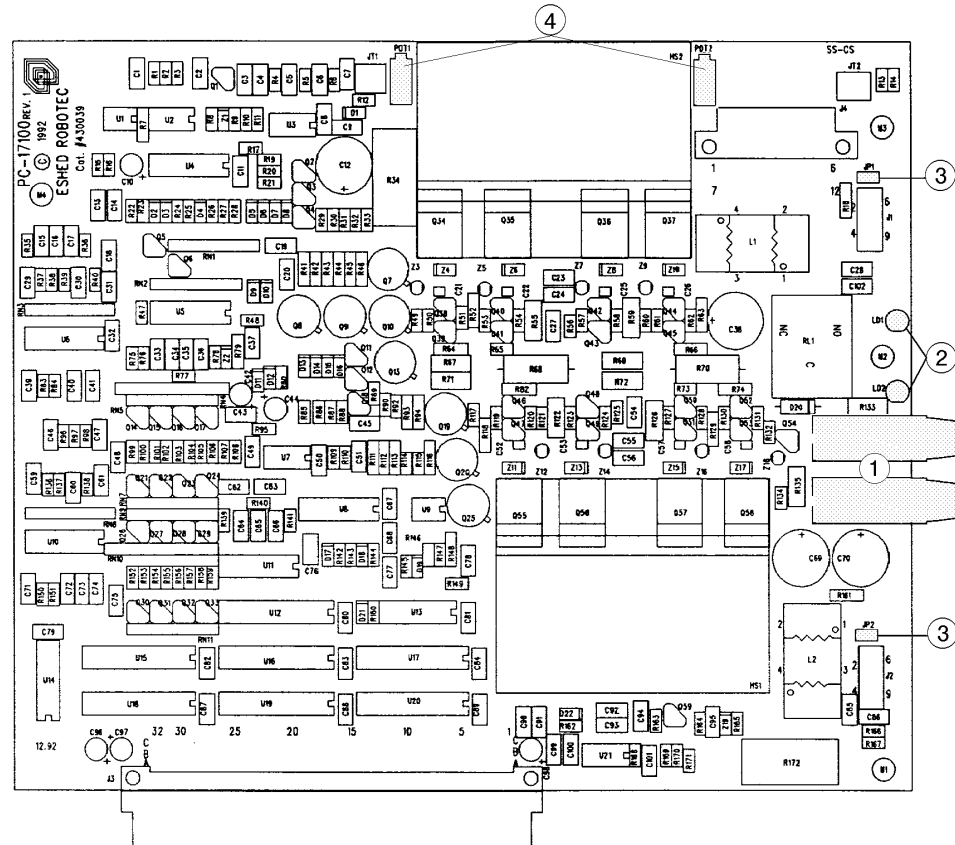


Figure 5-4: Axis Driver Card

① Fuses

“Slow blow” fuses; slightly more than the maximum current that can pass through the axis’ motor.

Robot	Fuse Rating	Function
SCORA-ER 14	8A (SB)	Upper fuse: axes 1,2,3
	4A (SB)	Upper fuse: axis 7 Lower fuse: axes 4,5,6,8
SCORBOT-ER IX	8A (SB)	Upper fuse: axes 1,2,3,7
	4A (SB)	Lower fuse: axes 4,5,8
	2A (SB)	Lower fuse: axis 6
PERFORMER-MK2	10A (SB)	Upper fuse: axes 1,2,3
	4A (SB)	Upper fuse: axis 7 Lower fuse: axes 4,5,8
	2A (SB)	Lower fuse: axis 6

② LEDs

A lit LED indicates the axis driver is receiving power.

An unlit LED indicates one of the following:

- An emergency button has been pressed somewhere in the system.
- The controller's motors switch is off.
- A fuse has blown.

③ Jumpers (Encoders)

Jumpers J1 and J2 are used for configuring the encoders.

Default factory setting: open.

Do not alter the status of these switches unless specifically instructed to do so by Eshed Robotec.

④ Potentiometers

Potentiometers for adjusting motor current limit.

The upper axis is measured by JT1, and adjusted by POT1.

The lower axis is measured by JT2, and adjusted by POT2.

The potentiometer is rotated until the proper current limit is obtained—approximately 1V per 1A.

Connector Card

Refer to Figure 5-5.

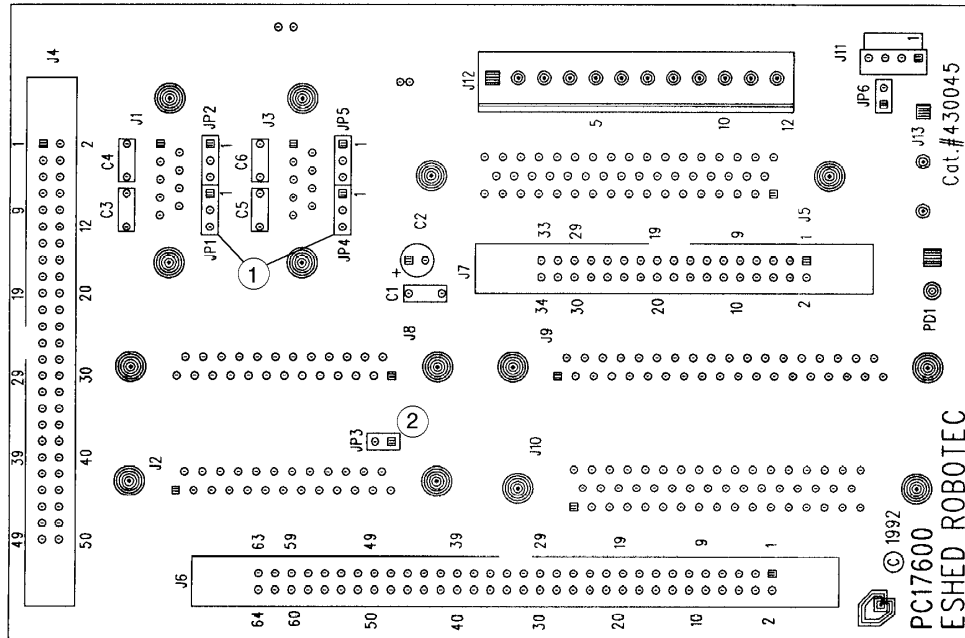


Figure 5-5: Connector Card

① Jumpers (RS232)

Jumpers JP1, JP2, JP4, JP5 are used to configure RS232 communication.

Default factory setting (with standard cable connection, as shown below):

Jumper	Shorted	Controller COM
JP1, JP2	Pins 1 and 2	CONSOLE
JP4, JP5	Pins 1 and 2	COM Ø

Standard RS232 Cable Pinout	
Controller – D9	PC – D25
GND 5	GND 7
TRX 3	RX 3
RX 2	TRX 2
	6,20
	4,5,8

② Jumper (Emergency)

Jumper JP3 must be removed when the controller is equipped with **ACL** EPROM version 2.28 and later.

Power Supply

Refer to Figure 5-6



DO NOT OPEN the power supply housing. Only authorized technicians are permitted to do so.

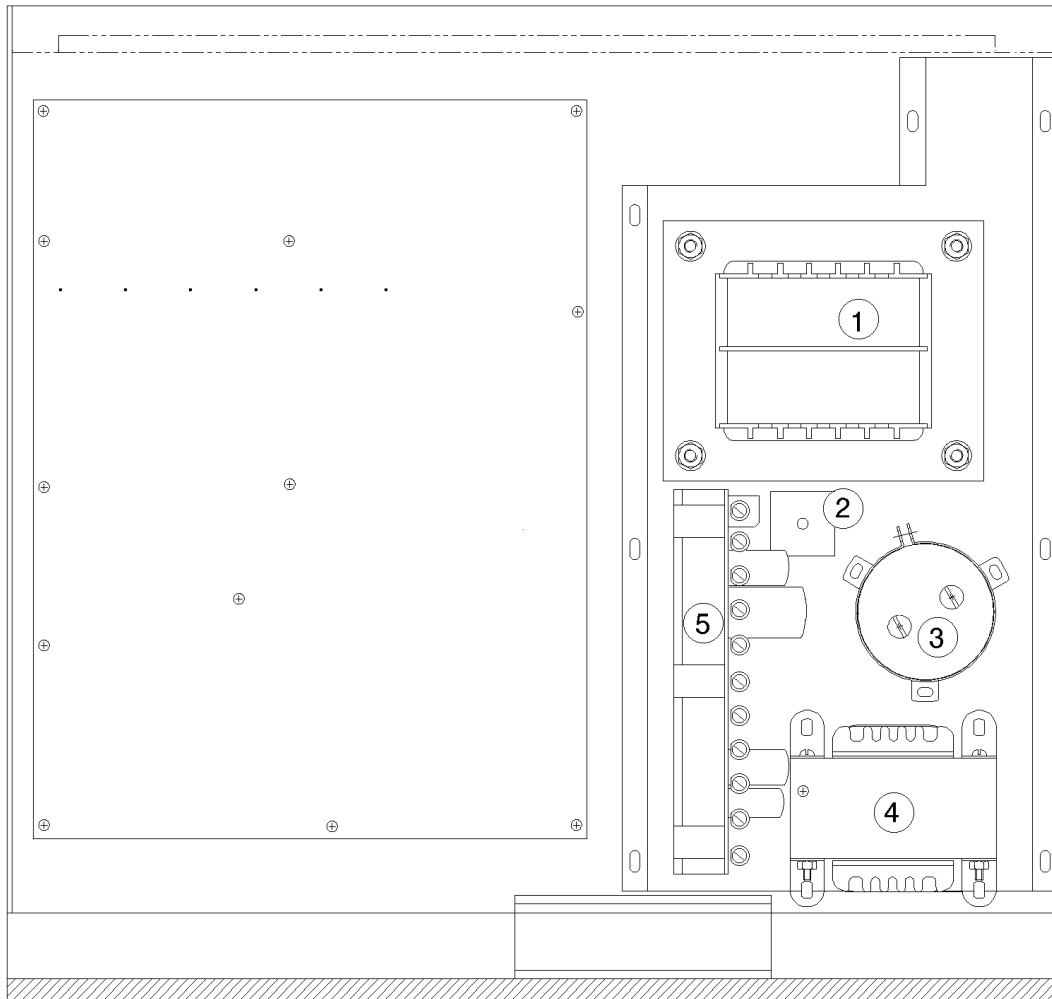


Figure 5-6: Power Supply

① Transformer

- +12V User Power Supply and I/O Board
- +24V User Power Supply and I/O Board
- -40V Robot Power Supply
- $\pm 12V$ to Drivers

② **Diode Bridge**

③ **Capacitor**

27,000 μ F/65V

④ **Transformer**

Logic power +5V, \pm 12V

⑤ **Power Supply Circuit Fuses**

The eleven fuses can be accessed *without removing the cover of the power supply*. Each fuse is properly labeled. Refer to Appendix A for details on each fuse.

Auxiliary RS232 Communication Card

Refer to Figure 5-7.

This is a multiport RS232 communication card. The card has a D37 connector for the connection of a cable with multiple D25 connectors. Two cables are available from Eshed Robotec:

- Cable with eight D25 connectors, marked COM1 through COM8, for connection to corresponding COM ports on computers or other controllers, as required in ER-CIM systems.
- Cable with two D25 connectors, marked COM1 and COM2, for connection to corresponding COM ports on computers or other controllers.

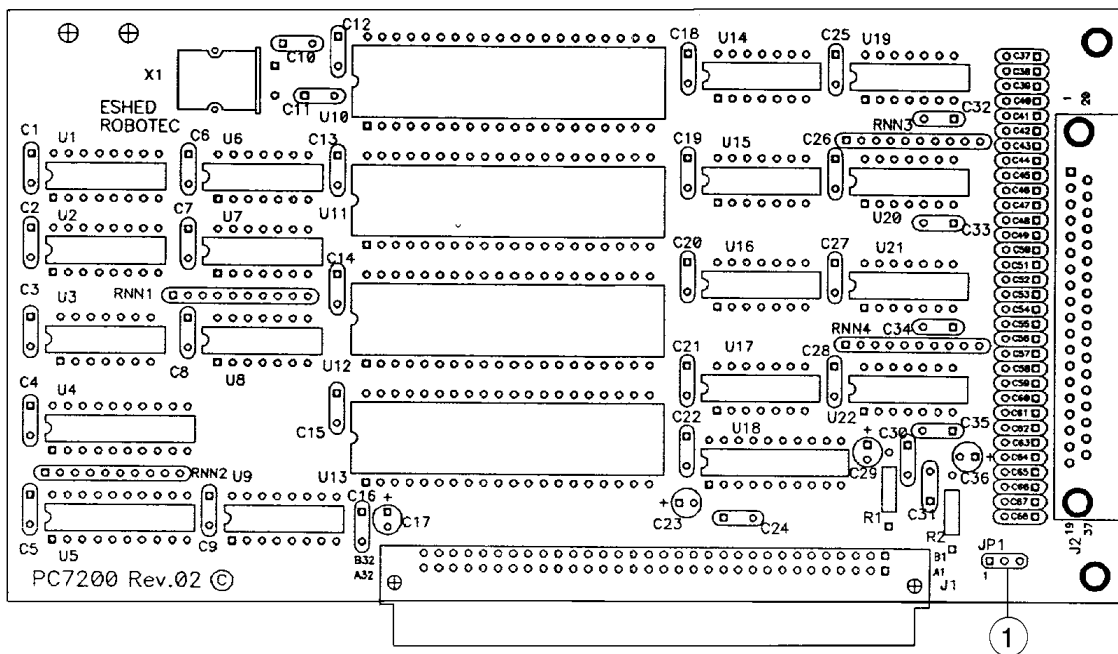


Figure 5-7: Auxiliary Multiport RS232 Communication Card

① Jumper (RS232)

Jumper JP1 allows the software to determine whether or not the communication card has been installed in the controller.

Default factory setting: Pin 1 and pin 2 shorted.

CHAPTER 6

Maintenance and Repairs

All instructions in this chapter can be performed by the user.



Make sure the controller's POWER switch is turned off before you open the controller cover. Make sure the power cable is disconnected from the AC power source before you remove AC power fuses.

Troubleshooting

This section describes possible controller problems and suggestions for correcting them. Complete instructions for removing and replacing controller components are given in the section, “Upgrade and Repair Procedures,” later in this chapter. Refer also to your robot's *User's Manual* for additional troubleshooting instructions.

You may also be able to identify the component responsible for a problem by replacing the suspected faulty component with an identical component from a working system.

1. *A message appears on the screen: AC/Fuse FAILURE*

This message appears when a controller fuse has blown.

It also appears whenever the controller is shut down, either normally or by a power failure.

A message appears on the screen: Motor power switch is OFF

This message appears when a motor or movement commands (for example, CON and MOVE) could not be executed because the MOTOR switch is not depressed,

- Be sure the motors switch is on. Activate CON. Then repeat the command.

2. *Gibberish (garbled text) appear on the screen.*

Enter the **ACL** command ENGLISH to make the system communicate in English.

-
3. *Controller power does not turn on. Power LED does not light. Fans do not rotate.*
-
- Make sure the AC power supply matches the controller's voltage requirement, as seen on the cover of the AC fuse drawer at the back of the controller. If the voltage supply and controller voltage setting do not match, follow the instructions later in this chapter for changing the controller's voltage setting.
 - Make sure AC power is being supplied to the power outlet.
 - Make sure the power cable is connected to both the proper power source and the controller.
 - Using an ohmmeter, measure the resistance of the fuses. If resistance is close to 0 Ω , the fuse is functioning. Check the fuses in the following order:
 - Check the 6.3A fuse (part of power inlet/switch unit).
 - Check fuse F1, and then fuse F2 (near power inlet/switch unit).
 - Turn off the controller, open the controller cover, and check the 0.5A (SB) fuse (marked FAN/POWER RELAYS).

-
4. *No communication between the controller and the computer/terminal. Message appears on the screen: Controller Not Responding*
-
- Make sure the controller's POWER switch is turned on.
 - Make sure the RS232 cable between the controller CONSOLE port and the computer COM port is properly connected. Make sure you have loaded ATS with the proper /C switch. (Refer to installation instructions in Chapter 2.)
 - Make sure the RS232 cable is properly wired.

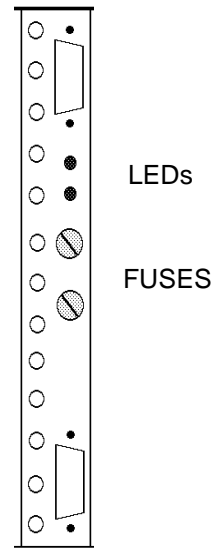
-
5. *No communication between the controller and the teach pendant.*
-
- Make sure the controller's POWER switch is turned on.
 - Make sure the teach pendant cord is properly connected to the teach pendant and to the controller's Teach Pendant port (via the adapter plug).
 - Make sure adapter is securely inserted into the port. Check the green LED on the adapter plug. Make sure it is lit, indicating a working connection.

Do not attempt to open the teach pendant. There are no user-servicable parts inside.

-
6. *Motors suddenly stop. No message on screen. No response to keyboard entries*
-
- Turn off the controller and open up the cover. Turn on the controller. Check the yellow “watchdog” LED on the main board. If it is lit, it indicates that that one of the following fuses on the power supply unit has blown out: +12VA, -12VA, +12VDR, -12VDR.
 - Turn off the controller and check each of these four fuses. Replace the blown fuse.

-
7. *Controller is totally inoperative although all power supplies are working.*
-
- Turn the controller off and on.
 - Turn off the controller and open up the cover. Turn on the controller. Check the red “halt” LED on the main board. If it is lit, the CPU is in “halt” state.
 - Turn off the controller. Remove the CPU board. Make sure the RAM configuring jumper (JP1) and the DIP switches (S1) are set according to the RAM size.

-
8. *Controller functioning but robot arm cannot be activated.*
-
- Make sure controller is in the control off (COFF) state. Then activate the control on (CON) state.
 - Make sure the controller’s MOTORS switch is on and the green LED is lit.
 - Make sure all robot cables are properly connected to the controller.
 - Check whether the driver card LEDs on the controller’s rear panel are lit, indicating that power is being supplied to the axis driver. The upper LED corresponds to the upper axis and the lower LED corresponds to the lower axis, as shown at right.
If LED is not lit, check the driver card fuse. The upper fuse serves the upper axis and the lower fuse serves the lower axis.
 - If the fuse is okay, turn off the controller, and open the cover. Make sure the cable from the power supply unit to the driver cards is properly connected.



9. *Controller does not receive an input signal.*

- To determine whether the problem is in the controller or a user application, enter the **ACL** command: **SHOW DIN**

Zeros and ones appear on the screen (0=OFF and 1=ON). These numbers correspond to the status of the 16 inputs.

- If the Input is configured as source: Short the specific input to ground and again type: **SHOW DIN**
- If the Input is configured as sink: Short the specific input to 12V User's Power Supply, and again type: **SHOW DIN**

If the status of the input changes, the problem is in the user application.

If the status of the input does not change, check the following:

- If the input LED also does not light up, the problem might be in the display card PC7300, or in a faulty connection between the display card and the main board.

10. *I/O display LEDs do not light up.*

- Check whether the flat cable from the main board (J18) to the display card PC7300 is properly connected.

11. *Controller does not give output signal.*

- For Outputs 1-4, check whether the relays have been switched (LED is lit):
 - In output OFF, NC is shorted to COM, NO is disconnected from COM.
 - In output ON, NO is shorted to COM, NC is disconnected from COM.

If outputs have not been switched, check the flat cable connecting the main board (J17) and the I/O card.

- For Outputs 5-16 (open collector) check whether the load and voltage source have been properly connected. (If the supply has been connected directly to the output terminal, the output transistor will blow out immediately).

12. *User programs are not saved in BBRAM. Response to **ACL** command **DIR** is: Directory is Empty.*

- Check the voltage level of the backup battery on the CBU board. A good battery produces 2.7–3.3V. Replace the battery if necessary.
- Make sure the battery is securely set in socket.

-
13. *Emergency LED remains lit even after the Emergency button is released.*
- Make sure the teach pendant is properly connected to the controller via the adapter plug.
 - If a teach pendant is not installed, make sure the by-pass plug is properly connected to the controller.

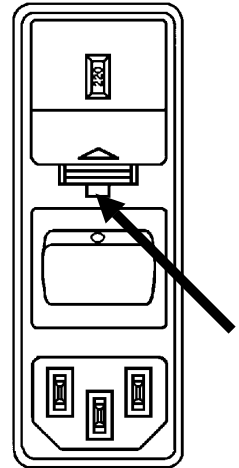
Repair and Upgrade Procedures

Changing the Voltage Setting

If your AC power supply does not match the controller's voltage setting, *do not connect the controller to the AC power supply!*

You must first change the voltage setting of the controller.

1. To change the voltage setting, you must remove the 6.3 fuse holder. Use a small screwdriver to release the fuse drawer cover (as indicated by the arrow in the diagram at right).
2. Pull out the fuse drawer, and set it aside.
3. Again using the small screwdriver, pry out the white voltage switch from its housing.
4. Find the side of the voltage switch marked with the voltage setting you need, and reinsert the switch so that the correct voltage setting is visible. Press the voltage switch into the housing until it snaps into place.
5. Reinsert the fuse drawer and press it until it snaps into place.



Opening the Controller Cover

1. Turn off the controller's POWER switch.
2. Unscrew the 10 Phillips screws which hold the cover.
3. Carefully lift off the cover and set it aside.

Replacing Fuses

There are a number of different fuses in the controller. They are described in Appendix A.



Warning! Before you remove any AC power fuses, turn off the controller's POWER switch, and disconnect the power cable from the AC power source.

AC Power Fuses

These fuses are located at the rear of the controller.

- To remove the main AC power fuses, use a small screwdriver to release the fuse drawer cover (as shown in the diagram above). Pull out the fuse drawer, and remove and replace the fuses. Reinsert the fuse drawer and press it until it snaps into place.
- To remove the AC logic PSU fuses (marked F1 and F2), push down on the fuse holder cover while turning it counterclockwise. Remove the fuse holder and replace the fuse. Reinsert the fuse holder, and while pressing down turn it clockwise, until it is securely in place.

Power Supply Fuses

These fuses are marked and accessed on the cover of the power supply. To change these fuses you must open the controller cover. But, *do not open the power supply cover!*

To remove any of the power supply fuses, push down on the fuse holder cover while turning it counterclockwise. Remove the fuse holder and replace the fuse. Reinsert the fuse holder, and while pressing down turn it clockwise, until it is securely in place.

Driver Card Fuses

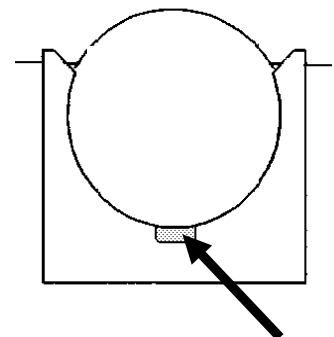
These fuses are accessed from the rear of the controller. To remove any of the power supply fuses, push down on the fuse holder cover while turning it counterclockwise. Remove the fuse holder and replace the fuse. Reinsert the fuse holder, and while pressing down turn it clockwise, until it is securely in place.

Removing/Replacing the Battery

The battery is located on the CPU board. The battery should be replaced when it supplies less than 3.0V.

Replace only with Renata CR2477N 3V battery, available from Eshed Robotec.

1. Turn off the controller.
2. Using a small screwdriver, pry the battery loose (as indicated by the arrow in the diagram at right). Pull out the battery.
3. To replace, slide the new battery into the slot until it snaps into place.



Adding RS232 Channels

An auxiliary RS232 communication card may be installed in the controller to provide additional RS232 communication channels. The cable leading from the card may have either two or eight D25 connectors.

1. First, make sure Pins 1 and 2 are shorted on the card's Jumper JP1.
2. Turn off the controller.
3. Remove the cover of the controller.
4. Remove the screws from the blank AUX bracket at the back of the controller. One screw is on the bottom of the bracket, on the outside of the controller. The other screw is at the top of the bracket, on the inside of the controller. Remove the blank bracket.
5. Before inserting the auxiliary RS232 card, first check that none of the 64 pins in the male DIN connector is bent. Then, make sure the card is directly above the female DIN connector on the main board, and that the metal bracket fits the rear panel. Firmly but gently press the card into the slot.
6. Retighten the bracket screws.
7. Make the cable connections:
 - Connect the D37 connector from the multipoint connector cable to the Auxiliary RS232 port (14) on the controller.
 - Connect the D25 connectors on the multipoint connector cable to the corresponding COM ports on the other controllers or computers.
8. The controller must be reconfigured for the auxiliary RS232 card.

Before you perform the configuration, you must backup to disk the entire contents of the controller, including all parameters.

Power on the system. From the **ATS Backup Manager** menu, select the options "Backup ALL" and "BACKUP to disk (F3)."
9. Perform the configuration, using either of the following methods.
 - Use the command <Ctrl>+F1, as described in the section, "Controller Configuration," in Chapter 2 of this manual; or
 - Use the **ACL** command CONFIG, as described in the *ACL Reference Manual* . 1
10. Reload the contents of the controller, including all parameters, which you backed up to disk. From the **ATS Backup Manager** menu, select the options "Restore ALL" and "RESTORE from disk (F5)."

Replacing a Driver Card

1. Turn off the controller.
2. Remove the cover of the controller.
3. Remove the screws from the driver card bracket at the back of the controller. One screw is on the bottom of the bracket, on the outside of the controller. The other screw is at the top of the bracket, on the inside of the controller.
4. Disconnect the driver power connector from the driver card.
5. Holding the card with two hands, lift it out very carefully.
6. Before inserting the new driver card, first check that none of the 96 pins in the male DIN connector is bent. Then, make sure the driver card is directly above the female DIN connector on the main board, and that the metal bracket fits the rear panel. Firmly but gently press the driver card into the driver card slot.
7. Reconnect the driver power connector.
8. Retighten the bracket screws.

Adding a Driver Card

1. Turn off the controller.
2. Remove the cover of the controller.
3. Remove the screws from the blank bracket on the empty driver card slot at the back of the controller. One screw is on the bottom of the bracket, on the outside of the controller. The other screw is at the top of the bracket, on the inside of the controller. Remove the blank bracket.
4. Before inserting the new driver card, first check that none of the 96 pins in the male DIN connector is bent. Then, make sure the driver card is directly above the female DIN connector on the main board, and that the metal bracket fits the rear panel. Firmly but gently press the driver card into the driver card slot.
5. Connect the driver power connector.
6. Tighten the bracket screws.

Adjusting Driver Card Current Limit

1. Turn off the controller's POWER switch.
2. Connect the common probe of a DC voltmeter to JT1 (for POT1) or to JT2 (for POT2).
3. Turn on the controller.
4. Using a small screwdriver, rotate either POT 1 (for upper axis) or POT 2 (for lower axis). Watch the voltmeter reading, and adjust until the proper value is reached. The voltmeter reading reflects the amperage of the current limit level.

Rotating clockwise reduces the level of the current limit;

Rotating counterclockwise increases the level of the current limit.

Rotate the potentiometer until you obtain a currently limit of approximately 1V per 1A.

Expanding RAM

To expand the RAM to 1MB or 2MB, do the following:

1. Turn off the controller.
2. Lift out the CPU board, on which the RAM ICs are located, from the controller.
3. Using an IC extracting tool, carefully remove the four RAM ICs mounted in U7, U8, U9 and U10.
4. Carefully insert the four new RAM ICs into the empty sockets of U7, U8, U9 and U10.

Insert each chip by aligning the notch and the pins on the chip with the notch and the pin receptacles on the socket. Gently push the chip into the socket until it is firmly seated.

5. Set the DIP switches at S1 on the CPU board as follows:

DIP switch S1-1	DIP switch S1-2	RAM Size
ON	ON	512 KB
OFF	ON	1 MB
OFF	OFF	2 MB

6. Set the jumper at JP1 on the CPU board as follows:

Shorted	RAM Size
Pins 1 and 2	512 KB
Pins 2 and 3	1 MB or 2 MB.

7. Power on the system. Using the **ACL** command CONFIG, configure the controller for the expanded RAM.

Upgrading the PLD

The controller has two programmable logic devices:

- PLD on main board
- PLD on CPU board.

To remove a PLD, use a PLCC extraction tool.



CHAPTER 7

Operation

This chapter introduces you to the basic commands for operating robots by means of both the **ACL** software and the teach pendant.

Auto Mode

*When you operate the system from the **keyboard**, the teach pendant must be switched to **Auto**. When necessary, you are prompted to enter the **ACL** command AUTO.*

Teach Mode

*When you operate the system from the **teach pendant**, the teach pendant must be switched to **Teach**. It is strongly recommended that you always keep the teach pendant in its mount.*

DIRECT Mode

This chapter describes the operation of the robotic system when it is functioning in the **DIRECT** mode. When the system is in **DIRECT** mode, the user has direct control of the axes, and the controller executes commands as soon as they are entered by the user.

When in **DIRECT** mode, the screen prompt appears like this: >_

When the system is operating in **EDIT** mode, commands are entered into a user program, which can be saved and executed at a later time. Program editing procedures are described in Chapter 8.

Manual Keyboard Mode

Manual mode is available when the system is in **DIRECT** mode. The Manual mode enables direct control of the robot axes from the keyboard when a teach pendant is not connected.

When using the keyboard to perform some of the procedures described in this chapter, the system must be in Manual mode.

To activate Manual mode, make sure Auto mode is active. Then hold the <Alt> key and press the character M.

Press: **<Alt> + m**

The system will respond in one of the following ways:

```
MANUAL MODE!      or      MANUAL MODE!  
>_                >_  
JOINT MODE        XYZ MODE
```

The system's response indicates the currently active coordinate system.

To exit Manual mode, the same command is used:

Press: **<Alt> + m**

```
EXIT MANUAL MODE...  
>_
```

Using this Manual

To familiarize yourself with the system, you should read through this chapter and practice entering the commands described in each section.

Nearly all operations described in this chapter can be performed from the keyboard. The steps for using the keyboard are indicated by the heading PC. The teach pendant is optional. The operations which can also be performed from the teach pendant are indicated by the heading TP.

This manual uses the following typographical conventions:

Descriptions of PC operation show user entries in bold, lowercase text. System responses are shown in uppercase text (though the actual screen display may be different.) For example:

Type: **home** **<Enter>**
WAIT!! HOMING...

The system is not case-sensitive. You may use either uppercase and lowercase characters to enter commands and data.

Descriptions of TP operation show the user entries (TP keys) in bold, uppercase text within brackets. System responses are shown in boxed text. For example:

Press: **[CONTROL ON|OFF]**

```
CoffA <enter>  
group:A ax:-- JOINTS
```

Activating the System

Activate the system and load the **ATS** software, as described in the section, “Power On,” in Chapter 2.

Homing the Robot and Peripheral Axes

- * *The Home routine must be executed at the beginning of each working session. Otherwise the controller will not execute the commands for recording positions and moving to recorded positions.*

The location of the robot axes is monitored by encoders which track the amount of movement relative to an initial position. To obtain repeatable robot performance, this initial position—home—must be identical each time the robot is used. Thus, whenever the system is activated, the homing program, which is internally programmed into the controller, must be executed.

During the homing procedure, the robot joints move and search for their home positions, one at a time. The **SCORBOT-ER IX** and **PERFORMER-MK2** robots, for example, are homed in the following sequence: shoulder (axis 2), pitch (axis 4) elbow (axis 3), roll (axis 5), base (axis 1). If a gripper has been defined in the controller setup, the gripper will open and close at the end of the homing routine.

To find its home position, the axis is moved until the microswitch on the joint sends a specific signal to the controller. The axis motor continues to rotate until its encoder produces an index pulse. At that point, the axis is at home.

- * *Before you begin the homing procedure, make sure the robot has ample space in which to move freely and extend its arm.*

TP

First make sure the TP is mounted and switched to Teach mode.

Press: **[RUN] [0] [EXECUTE]**

This instructs the controller to execute Program 0, the robot homing routine. The display panel on the teach pendant will show:

```
HOMING ...
Run 0
group:A ax:-- JOINTS
```

When the Home search is successfully completed, the TP will sound a beep, and the display panel will show:

```
Homing complete
```

If the robot is unable to find a home position in one or more of the axes, you will see a message such as:

```
Home fail [4]
```

To stop the homing while the operation is in progress, press [**ABORT**]. If you abort the homing routine, it must be executed again in its entirety.

The peripheral axes must be homed by means of the **ACL** commands HOME and HHOME, as described below.

PC

First make sure the system has been switched to Auto mode.

To home the robot axes (Group A), use the **ACL** command HOME.

Type: **home <Enter>**
WAIT!! HOMING...

If all axes reach their home position, a message is displayed:

```
HOMING COMPLETE
```

If the homing process is not completed, an error message identifying the failure is displayed:

```
*** HOME FAILURE AXIS 4
```

If the limit switch is found, but not the encoder's index pulse, the following message is displayed:

```
*** INDEX PULSE NOT FOUND AXIS 2
```

To stop the homing while the operation is in progress, use the abort commands:

Type: **A <Enter>**
or press: **<Ctrl>+A**

To home peripheral axes, each axis must be homed individually; for example:

Type: **home 7 <Enter>**
home 8 <Enter>
home 9 <Enter>

To home an axis, such as a slidebase, which uses a hard stop rather than a microswitch, use the **ACL** command HHOME.

Type: **hhome 8 <Enter>**

Coordinate Systems

Robots controlled by **Controller-B** can be operated and programmed in two different coordinate systems: Joint and Cartesian (XYZ) coordinates. For complete information on working with these coordinate systems, refer to Chapter 2 in the *ACL Reference Guide*.

(The TOOL coordinates option is not currently available from the teach pendant.)

TP

First make sure the TP is mounted and switched to Teach mode.

To toggle between the two coordinate systems:

Press: [JOINTS|XYZ]

```
group:A ax:-- XYZ
```

Press
again:

[JOINTS|XYZ]

```
group:A ax:-- JOINTS
```

The display reflects the currently active coordinate system.

Depending on the active coordinate system:

- Manual movements of the axes are executed according to either joint or XYZ coordinates.
- The TP's RECORD POSITION command records positions by executing either HERE or HEREC.

PC

First make sure the system has been switched to Auto mode.

Activate Manual mode.

Then, to activate the Joint coordinate system:

Press: J

```
JOINT MODE
```

To activate the XYZ coordinate system:

Press: x

```
XYZ MODE
```

Axis Control Groups

By default, the controller assumes the robot axes (Group A) are under servo control. The Group Select key allows you to switch control to peripheral axes (Group B), or to an independent axis (Group C), or the gripper (Group G).

TP

First make sure the TP is mounted and switched to Teach mode.

To select the axis control group:

Press: [**GROUP SELECT**]

group:G ax:6 JOINTS

Continue pressing this key for group G (gripper), group B, group C, and again for group A, and so on.

When group C is displayed (it will appear only if the controller has been configured for group C), use the TP numerical keys to enter the number of the independent axis. Then press [**ENTER**].

Note that the TP treats the gripper as a distinct axis control group, group G.

When alternating among control groups, group A will remain in the coordinate system (Joint or XYZ) in which it was last active.

PC

ACL does not have a command for selecting the axis control group. The specific format of each command indicates the axis control group.

Servo Control

The controller must be in the servo control (CON) state for the axes to execute movement commands.

Activating the Home routine will activate CON.

Certain events, such as impact, overheating (thermic error), or activation of an Emergency switch, will automatically disable servo control (COFF). CON must be activated to resume motion and servo control.

While the controller is in the COFF state, you cannot operate the axes.

TP

First make sure the TP is mounted and switched to Teach mode.

To enable and disable control of a selected group or all groups:

Press: [**CONTROL ON|OFF**]

```
CoffA <enter>
group:A ax:-- JOINTS
```

When the key is first pressed, it toggles to CON or COFF for the selected group.

Continue pressing the key for other options. The action to be performed (e.g., CoffB, Con All) will be displayed. If at least one group is enabled, COFF is applied to all groups. If all groups are disabled, CON is applied to all groups.

To accept the displayed option:

Press: [**ENTER**]

```
CONTROL DISABLED
CoffA <enter>
group:A ax:-- JOINTS
```

The text in the fourth line of the TP display is inverted when the selected group is disabled.

PC

First make sure the system has been switched to Auto mode.

The system must be in DIRECT mode (but not Manual mode).

Use the **ACL** commands CON and COFF to enable and disable servo control.

Type: `con <Enter>` Enables control of all axes.

Type: `coff <Enter>` Disables control of all axes.

The format can be altered to enable and disable control of specific groups of axes; for example:

<code>cona</code>	Enables control of robot axes (Group A).
<code>coffb</code>	Disables control of peripheral axes (Group B).
<code>con 9</code>	Enables control of axis 9 (Group C).

Moving the Axes

“Joint” and “XYZ” Movements

When the coordinate system is set to the XYZ mode, movement commands cause linear motion of the tool center point (TCP) along the X, Y and Z axes, while maintaining the tool’s pitch and roll angles relative to the robot’s point of origin.

When the coordinate system is set to the Joint mode, the robot responds to movement commands by moving from one defined point to another.

Peripheral axes always move according to Joint coordinates.

TP

When in Joint mode, the controller recognizes the joint functions of the teach pendant keys.

When in XYZ mode, the controller recognizes the Cartesian functions of the teach pendant keys.

The teach pendant offers the easiest method for moving the robot arm. You simply select an axis, and press the + or – key. Movement continues as long as the key is pressed, or until the axis limit is reached.

Make sure the TP is mounted and switched to Teach mode.

Make sure Group A control is enabled and in Joint mode.

Now move the axes of the robot, in both directions.

Press: [SELECT AXIS] [1|AXIS 1|X] [+]

Press: [–]

The base of the robot (axis 1) will move in both directions.

Press: [SELECT AXIS] [2|AXIS 2|Y] [+]

Press: [–]

The shoulder of the robot (axis 2) will move in both directions.

You may continue selecting axes and moving them.

Before continuing, make sure Group A control is enabled and in XYZ mode.

Now watch how the TP keys now affect the movement of the TCP.

Press: [1|AXIS 1|X] [+]

Press: [-]

The TCP will move along the X-axis.

Press: [2|AXIS 2|Y] [+]

Press: [-]

The TCP will move along the Y-axis.

Press: [3|AXIS 3|Z] [+]

Press: [-]

The TCP will move along the Z-axis.

PC

First make sure the system has been switched to Auto mode.

Control (CON) must be enabled, and Manual mode must be active.

The keys listed below are then used to move the robot.

The axes will move as long as the activating key is depressed, or until a fixed stop is reached. The gripper will either open completely or close completely.

In Joint mode, the keys produce the following movements:

Press:	1, Q	Move axis 1 (base)
	2, W	Move axis 2 (shoulder)
	3, E	Move axis 3 (elbow)
	4, R	Move axis 4 (wrist pitch)
	5, T	Move axis 5 (wrist roll), but not gripper
	6, Y	Closes/Opens <i>electrical gripper</i> (axis 6)

In XYZ mode the following changes in manual movement occur:

Press:	1, Q	TCP moves along X+ and X- axes.
	2, W	TCP moves along Y+ and Y- axes.
	3, E	TCP moves along Z+ and Z- axes.
	4, R	Pitch moves; TCP maintains position.

All other movements are the same as in Joint mode.

While moving the arm, you may alternate between XYZ and Joint modes as often as required.

If peripheral axes are connected, the following keys are also used:

Press:	7, U	Move axis 7
	8, I	Move axis 8
	9, O	Move axis 9
	0, P	Move axis 10
	-, [Move axis 11
	+,]	Move axis 12

Activating the Gripper

In response to the commands to open and close, the gripper goes completely from one state to the other.

TP: First make sure the TP is mounted and switched to Teach mode.

Press: **[OPEN | CLOSE]**

This command functions on both electric and pneumatic grippers.

The **[Open|Close]** key toggles the gripper between its open and closed states. If the gripper was open it will now close, and vice versa. Repeat the command.

PC: Make sure the system has been switched to Auto mode, and control enabled.

The **ACL** commands **OPEN** and **CLOSE** are used for both electric and pneumatic grippers.

Type: **open <Enter>**

Type: **close <Enter>**

Setting the Speed

When executing MOVE (go position) commands, the speed of the robot is defined as a percentage of maximum speed. Speed defined as 100 gives the robot maximum speed, while a speed of 1 is the minimum. When the system is first turned on, the default speed is set at 50, approximately half the robot's maximum speed.

When executing *manual* movements controlled (by the user) from the teach pendant or keyboard, the speed of the robot is relative to the speed setting, and much slower than MOVE command movements.

TP

Make sure Group A control is enabled and in Joint mode.

Now, set the robot's speed to a speed of 30% of maximum joint speed:

Press: **[SPEED%] [3] [0] [ENTER]**

All joint movement commands will be executed at a speed of 30, until a different speed is entered.

Make sure Group A control is enabled and in XYZ mode.

Now, set the robot's speed to 30% of maximum linear speed:

Press: **[SPEEDL%] [3] [0] [ENTER]**

All XYZ movement commands will be executed at a speed of 30, until a different speed is entered.

Note that although SPEEDL value is entered as a percentage, it is converted to a value in millimeters/second.

PC

The ACL commands SPEED and SPEEDL are used to define the speed at which joint movements and linear movements are executed. For example:

speed 50	Sets speed of joints movements of Group A axes to 50% of maximum speed.
speedb 20	Sets speed of joint movements of peripheral axes (Group B) to 20% of maximum speed.
speedl 20	Sets speed of linear, circular and linear spline robot movements to 20 mm/sec.

For more details, refer to the command descriptions in the *ACL Reference Guide*.

Defining and Recording Positions

Defining a position reserves space in controller memory, and assigns it a name.

Recording a position writes coordinate values to the allocated space in controller memory.

Two types of position names are possible:

- Numerical names (such as 3, 22, 101) of up to five digits. Positions with this type of name do not need to be defined before they are recorded; the position recording command automatically defines and records positions with numerical names.
- Alphanumeric names (such as P, POS10, A2). The name may be a combination of up to five characters, and should begin with a letter. These positions cannot be accessed from the teach pendant.

Positions may belong to a vector; that is an array of positions identified by a specific name and an index; for example, PVEC[1] and PVEC[5] are positions in a vector named PVEC. When a vector is attached to the teach pendant (by means of the **ACL** command ATTACH), vector positions can be accessed from the teach pendant by means of their index number.

If you accidentally record coordinates for position 0, execute the Home program. The homing routine records the proper coordinates for position 0.

For more information on positions and position vectors, refer to Chapters 2 and 3 in the *ACL Reference Guide*.

TP

The teach pendant simultaneously defines and records a position for any axis group.

The position is defined for the currently active group, and receives the current values of the axes in that group. The position coordinates are recorded in the currently active coordinate system.

To record a robot position, make sure Group A control is enabled and in Joint mode. Then use the axis movement keys to bring the robot to any location. Record this as position 12.

Press: [RECORD POSITION] [1] [2] [ENTER]

Here 12

You may press up to five digits for the position name. If you use a position name which has already been defined, the new coordinates will overwrite the existing ones.

This command is also used to record positions in a vector. The vector must first be attached (ATTACH) to the TP.

Move the robot to another location and record it as position 13.

PC

Use the **ACL** command DEFP to define a robot position. For example:

Type: `defp A1 <Enter>`

Defines position A1 for the robot.

When a position is defined, it is assigned to a specific axis control group. By default, it is assigned to the robot (Group A) axes. To define a position for Group B, or an independent axis, the command format determines the group to which the position is dedicated. For example:

<code>defpb B24 <Enter></code>	Defines position B24 for Group B.
<code>defpc C3 10 <Enter></code>	Defines position C3 for axis 10.

Now define three robot positions:

Type: `defp A31 <Enter>`
`defp A32 <Enter>`
`defp A33 <Enter>`

The **ACL** command HERE records a position—in *joint coordinates*—according to the current location of the axes.

The **ACL** command HEREC records a position in *Cartesian coordinates*

Remember to activate Manual mode before starting motion, and to exit Manual mode when the motion is completed. Also be sure the position is defined before you attempt to record it.

Move the robot to any location, and record its coordinates for position A31.

Type: `here A31 <Enter>`

Move the robot two more times, and record coordinates for positions A32 and A33.

If you attempt to record a position which has not been defined (for example HERE A34), the system will display an error message.

If you specify a name of a position which has already been recorded (for example, HERE A31), the HERE command will overwrite the existing coordinates with new coordinates.

Relative Positions

TP

Relative positions cannot be recorded by means of the teach pendant.

PC

The **ACL** commands **HERER** and **TEACHR** allow you to record a position as relative to another position, or as relative to the current position of the robot.

To record a position which is relative to another position by *joint coordinates*, move the robot to the relative location and record the position. For example:

Type: **herer A99 A33 <Enter>**

The coordinates of position A99 are actually offset values; that is, the difference in the encoder count at position A31 and at position A99. If the coordinates of position A31 change, position A99 will remain relative to position A31 by the same number of encoder counts.

To record a position relative to the current location of the robot by *joint coordinates*, you are prompted to enter values (encoder counts) for each of the axes. If offset values have already been recorded for this position they will appear in the brackets; you can press <Enter> to accept a displayed value. If no offset values have been recorded, the bracket contains only a dot [.]); you must enter a value. For example, to record a position for the **SCORBOT-ER IX** robot:

Type: **herer A99 <Enter>**
1-- [.] >0 <Enter> Base = no offset
2-- [.] >500 <Enter> Shoulder = 500 counts offset
3-- [.] >250 <Enter> Elbow = 250 counts offset
4-- [.] >0 <Enter> Pitch = no offset
5-- [.] >0 <Enter> Roll = no offset

The command **TEACHR** allows you to record a position which is relative to another position, or relative to the current position of the robot, in *Cartesian coordinates*. **TEACHR** can be easily used to maintain a vertical offset (along the Z-axis) between two positions; for example:

Type: **>teachr over** Relative position OVER will always
X -- [.] > 0 <Enter> be 50mm above the current position
Y -- [.] > 0 <Enter> of the robot.
Z -- [.] > 500 <Enter>
P -- [.] > 0 <Enter>
R -- [.] > 0 <Enter>

ACL has a number of commands for recording position coordinates; they are detailed in the *ACL Reference Guide*, and will not be discussed in this manual.

Listing Positions

PC

To see a list of the defined positions, use the **ACL** command LISTP.

Type: **listp <Enter>**

The list of defined positions is displayed on the screen. Positions 12, 13, A31, A32, A33 and A99 should now appear in the list.

To view the coordinates of position A31, use the **ACL** command LISTPV.

Type: **listpv A31 <Enter>**

Position coordinates are displayed on the screen in the following manner.

```
1:-90039      2:-16861      3:-47555      4:27792      5:862
X:257.986    Y:399.450     Z:514.093     P:-72.743    R:2.273
```

Two sets of values are displayed for robot positions:

- The first line shows the joint coordinates; defined in encoder counts.
- The second line shows the Cartesian (XYZ) coordinates. X, Y and Z are defined in millimeters; P (pitch) and R (roll) are defined in degrees.

Deleting Positions

PC

To delete positions, use the **ACL** command DELP.

Type: **delp A99 <Enter>**

```
DO YOU REALLY WANT TO DELETE THAT POINT? (YES/NO)>_
```

Type: **yes <Enter>.**

```
A99 DELETED.
```

To prevent accidental deletion of a position, you are required to respond by entering the entire word “yes”, followed by <Enter>. Entering any other other character, including Y, is regarded as “no.”

Moving to Recorded Position

Once a position has been recorded, you can easily send the robot (or other devices connected to the controller) to that position.

You can use the PC to move to positions recorded by the TP. Alternately, you can use the TP to move to positions recorded by means of the PC, providing the positions are defined by *numerical* names.

The currently active coordinate system determines whether the movement of the robot (Group A) will be either joint (MOVE) or XYZ (MOVEL).

SPLINE movements are either joint or linear, depending on the type of positions in the vector, regardless of the currently active coordinate system.

For more information on movement commands, refer to the *ACL Reference Guide* and the *Teach Pendant for Controller-B User's Manual*.

Joint Movement

TP

Make sure Group A control is enabled and in Joint mode.

Assuming the robot is at position 13, send the robot back to position 12.

Press: **[MOVE] [1] [3] [EXECUTE]**

Do not release the Execute key until the axes reach the target position.

If the Execute key is released, the movement is stopped immediately, and the command is aborted.

To send all the axes of group A to the home position:

Press: **[MOVE] [0] [EXECUTE]**

PC

Use the **ACL** command MOVE to send the robot to a position.

Assuming the robot is at home, send the robot to position A31.

Type: **move A31 <Enter>**

In this command the robot moves at the current speed setting.

The MOVE command may contain a duration parameter, which is defined in hundredths of a second. To send the robot to position A32 in 10 seconds:

Type: **move A32 1000 <Enter>**

Linear Movement

TP Make sure Group A control is enabled and in XYZ mode.

Assuming the robot is at position 13, send the robot back to the home position in a straight path.

Press: **[MOVEL] [0] [ENTER]**

Do not release the Execute key until the axes reach the target position.

PC To move the TCP in a straight path, use the **ACL** command MOVEL.

For example, send the robot from the home position to position A33.

Type: **move 0 <Enter>**

Type: **movel A33 <Enter>**

Circular Movement

TP Make sure Group A control is enabled.

Assuming the robot is at position 13, send the robot position 12 in a curved path via the home position.

Press: **[MOVEC] [0] [ENTER] [1] [2] [EXECUTE]**

Do not release the Execute key until the axes reach the target position.

Movement is always according to XYZ, regardless of the currently active coordinate system.

PC To move the TCP along a curved path, use the **ACL** command MOVEC

You must specify two positions for MOVEC in order to define the curve. For example, send the robot from the home position to position A31, via position A32.

Type: **move 0 <Enter>**

Type: **movec A31 A32 <Enter>**

Additional TP Commands

The SPLINE, INSERT and DELETE commands are discussed in the *ACL Reference Guide* and the *Teach Pendant for **Controller-B** User's Manual*.

The SINGLE STEP command on the teach pendant is not currently available.

CHAPTER 8

Programming with ACL

This chapter serves as a tutorial to help you become familiar with program editing. To learn how to write and edit a program, you should follow, in sequence, the procedures described in this chapter.

- * *This chapter introduces you to the basic commands for programming in ACL. Many more commands and formats are available in the ACL language. Refer to the ACL Reference Guide for complete lists and descriptions of editing functions and ACL commands.*

EDIT Mode

So far you have learned to operate the robot in the DIRECT mode, in which all commands are executed the moment you press <Enter>.

To write programs which will be executed by the robotic system, you will use the EDIT mode.

Whenever the EDIT mode is active, the screen shows the current program line number and a prompt, such as this: 143 : ?_

The controller assigns the line numbers; they are not user definable.

Help

Quick, on-line help is available while you are working with ACL.

When in EDIT mode, use the command HELP to display a list and brief explanations of all EDIT mode commands.

When in DIRECT mode, use the command HELP *topic* to display a brief explanation of a *topic* (command or subject).

Creating a Program

To create a program, activate the EDIT mode by using the command EDIT, followed by the name you want to call the program. Program names are limited to five characters; for example:

Type:

```
edit aaa <Enter>
      AAA NEW PROGRAM
      DO YOU WANT TO CREATE THAT PROGRAM (Y/N)>
```

Type:

```
y <Enter>
      PROGRAM      AAA
      *****
      25: ?_
```

At the ?_ prompt, you can begin entering program command lines.

Writing a Program

To write a program which will send the robot to each of the positions recorded earlier, enter the following command lines:

Type:

```
moved A31 <Enter>
moved A32 <Enter>
moved A33 <Enter>
exit
      AAA IS VALID
```

Although the command MOVE may be used in EDIT mode, the command MOVED is preferable. MOVED ensures that the robot will accurately reach the target position before continuing to the next command.

The commands MOVEL, MOVEC and SPLINE are also available in EDIT mode. As with the MOVE command, it is preferable to use the command format with the D suffix; that is, MOVELD, MOVECD, SPLINED.

The EXIT command is used to end the current editing session and return the system to DIRECT mode.

Running a Program

When the > prompt is displayed, it indicates the system is in DIRECT mode. To check the program you have just created, do the following:

Make sure control is enabled (CON) and the robot is at its home position.

Type: **run aaa <Enter>**
 DONE

The robot moves to positions A31, A32 and A33, and then stops.

Program Loop

You will now edit the program and add command lines which will cause the program to run in a loop.

Program loops are created by using the companion commands, LABEL and GOTO.

- LABEL *n* marks the beginning of a routine.
- GOTO *n* sends program execution to the line which follows the corresponding LABEL.

Type: **edit aaa <Enter>**
 WELCOME TO ACL EDITOR, TYPE HELP WHEN IN TROUBLE.
 PROGRAM AAA

 25: ?_

The prompt shows the first line of the program. Entering a new command inserts a command line at this point.

Pressing <Enter> without entering a new command simply displays and accepts the line as is, and moves the editor to the next line.

Type: **label 1 <Enter>**
Press: **<Enter>**
Press: **<Enter>**
Press: **<Enter>**
Type: **goto 1 <Enter>**
Type: **exit <Enter>**
 AAA IS VALID

Displaying Program Lines

To view the program you have edited, use the command LIST, followed by the name of the program.

Type: **list aaa <Enter>**

```
          PROGRAM      AAA
*****
25:  LABEL  1
26:  MOVED  31
27:  MOVED  32
28:  MOVED  33
29:  GOTO   1
30:  END
      (END)
```

END marks the end of a program; (END) marks the end of a listing. They are written by the controller; they are not entered by the user.

Halting Program Execution

Bring the robot to its home position, and then run program AAA. The robot moves to positions A31, A32 and A33 in a continuous loop, without stopping.

Since you have now created and executed a program which will run in an endless loop, this section describes the **ACL** commands which are used to halt a program during its execution.

Suspend the Program

The companion commands SUSPEND and CONTINUE, respectively, suspend execution of a program, and then restart it from the point of interruption by executing the next program command line.

Type: **suspend aaa <Enter>**

The robot completes the current movement command and then stops. Program AAA is now suspended.

Type: **continue aaa <Enter>**

The CONTINUE command causes the robot to continue moving from the point where it was halted by a SUSPEND command.

Abort the Program

To immediately abort running programs and stop all axis movement, enter the abort command in either one of the following ways:

Type: **a <Enter>**

Press: **<Ctrl>+A**

```
PROGRAM AAA ABORTED
```

Program AAA can now be reactivated only by means of the RUN command, which will start the program from the beginning.

If several programs are running, and you want to abort only one of them, following the command by the name of the specific program; for example:

a aaa <Enter>

This format aborts the specified program only after the command currently being executed has been completed.

Stop the Program

To include an abort command in a program you are editing, use the command STOP.

The STOP command will abort a program only after all axis movement commands which have already been sent to the controller (movement buffer) have completed execution.

Use the STOP command in one of the following ways:

Type: **stop aaa <Enter>** Aborts only program AAA.

Type: **stop <Enter>** Aborts all running programs.

STOP is available in EDIT mode only.

STOP cannot be used to abort a running program when in DIRECT mode.

Delaying Program Execution

The DELAY command causes program execution to pause for a specified amount of time.

The DELAY command ensures that preceding commands have been properly executed before the next command is executed.

The command format includes a time parameter, n , which is expressed in hundredths of a second; for example, if $n = 150$, the delay is 1.5 seconds.

Edit program AAA. Insert delay commands following each MOVED command line.

```
Press: <Enter>
Press: <Enter>
Type:  delay 200 <Enter>
Press: <Enter>
Type:  delay 200 <Enter>
Press: <Enter>
Type:  delay 200 <Enter>
Press: <Enter>
Type:  exit <Enter>
```

Another ACL command, WAIT, command causes program execution to pause until a certain condition is met.

Variable Programming

Variables are locations in controller memory which are defined by name and hold values. Variables simplify programming by allowing instructions to be executed conditionally and repeatedly.

ACL has a number of system defined variables whose values indicate the status of inputs, outputs, encoders and other control system elements. Some of these variables can accept user defined values. None of these variables can be deleted from the system.

User variables are defined and manipulated by the user, and can be created or deleted as needed. User variables may be either private (local) or global.

- **Private variables** are defined and manipulated in the EDIT mode and recognized only by the specific program in which they are defined.
- **Global variables** can be defined and manipulated in both the EDIT and DIRECT modes, and can be used in any program.

The command DEFINE defines a private variable. Up to twelve variables can be defined in one command. For example:

Type:	define pv <Enter>	Defines PV as a private variable.
	define va vb vc <Enter>	Defines VA, VB and VC as private variables.

The command GLOBAL defines a global variable. Up to twelve variables can be defined in one command. For example:

	global gv <Enter>	Defines GV as a global variable.
	global gva gvb gvc <Enter>	Defines GVA, GVB and GVC as global variables.

Variable names must begin with an alphabetical character and may have up to 5 characters.

The commands DIM[*n*] and DIMG[*n*] are used to define arrays of private and global variable arrays, respectively; *n* defines the dimension (number of elements) in the array.

Mathematical and Logical Functions

The SET command performs mathematical and logical operations on variables. The command format may be one of the following:

```
set var1=var2
set var1=oper var2
set var1=var2 oper var3
```

Where: *var1* is a variable;

var2 and *var3* can be either a variable or a constant.

oper is: Arithmetic operator: + - * /

Algebraic operator: ABS, EXP, LOG, MOD

Trigonometrical operator: COS, SIN, TAN, ATAN

Logical (Boolean) operator: AND, OR, NOT

The simplest format assigns a variable the value of a constant or another variable. For example:

```
set var = 1
set var1 = var2
```

The value of a variable can be the result of an operation performed on another variable. For example:

<pre>set var1 = abs var2</pre>	If the value of <i>var2</i> is -1, <i>var1</i> is set to 1.
--------------------------------	---

The value of a variable can be the result of a mathematical operation performed on either two other variables or another variable and a constant. For example:

```
set var1 = var2 + 1
```

The value of *var1* is greater by 1 than the value of *var2*.

```
set vara = varb * varc
```

The value of *vara* is the result of *varb* multiplied by *varc*.

```
set var = var + 100
```

The result of an operation can equal the same variable, thereby changing its value. The value of *var* now equals the previous value of *var* plus 1000.

Iteration Functions

Many applications require task iteration, or repetition. Variables can be used to produce program loops which repeat a command or commands, thereby avoiding the need for redundant command lines within a program

The command format **FOR *var1* = *var2* TO *var3*** enables a program routine to be executed repeatedly. *Var1* must be a variable; *var2* and *var3* may be either variables or constants. For example, enter the following commands to create program LOOP:

```
edit loop
define var
for var=1 to 10
println "LOOP"
endfor
exit
```

The variable is a counter, which is set initially to 1 and increased by one each time the loop is performed. When the counter value reaches the final value (10 in this example), the loop is performed for the last time.

The ENDFOR command is required to mark the end of the loop.

The PRINTLN command causes comments (text within quotation marks) to be displayed on the screen during program execution. Thus, when you run program LOOP, the word "LOOP" will be displayed 10 times.

By altering the PRINTLN command line you can cause the system to report which loop has been completed. Bring the cursor to the ENDFOR command line. Enter the command DEL; this will delete the preceding command line. Then enter a new command line:

```
println "LOOP " var
```

Make sure you have included a space following the text "LOOP."

The PRINTLN command causes the current value of a variable to be displayed on the screen during program execution. Thus, when you run program LOOP, the following will now appear on the screen.

```
LOOP 1
LOOP 2
LOOP 3
```

. . . and so on, until LOOP 10 is displayed.

Add LABEL and GOTO command lines to program LOOP to cause the program to repeat continuously.

In the section on input/output programming later in this chapter, you will see additional examples of program loops which enable the system to check and respond to the state of the controller's 16 inputs.

Conditional Functions

Many applications require the program to flow according to certain conditions.

The command format **IF *var1 oper var2*** checks the relation between *var1* and *var2*. *Var1* must be a variable; *var2* may be either a variable or a constant. *Oper* is one of the following comparison operators: > < + >= <= <>

When the IF statement is true, the program executes the next line(s), until it reaches an ENDIF command, which marks the end of the conditional routine.

```
if var1=var2
  goto 1
endif
```

The IF statement may, however, be followed by another conditional statement. The next line may be an alternative condition (ORIF) or an additional condition (ANDIF).

```
if var1=var2
  orif var3>10
  goto 2
endif
```

At least one of the two conditions must be true in order for the program to jump to label 2

```
if var1=var2
  andif var3>10
  goto 2
endif
```

Both conditions must be true in order for the program to jump to label 2.

The conditional routine may also contain a routine to be executed when the IF condition is false. The beginning of such a routine begins with the command ELSE.

```

if var1=var2
  goto 2
else
  goto 1
endif

```

If the condition is not true, the program will jump to label 1.

Input and Output Programming

The state of the controller's 16 inputs and 16 outputs is determined by means of two system variables, IN[*n*] and OUT[*n*]; *n* specifies the I/O index; that is, 1–16.

The value of the variable indicates whether the input or output is on or off; when the value of the variable is 1, the input or output is ON; when the value is 0, the input or output is OFF.

Displaying Input/Output Status

The I/O LEDs on the front panel of the controller turn on and off to reflect the status of the inputs and outputs. If you are not close enough to see the controller panel, you may want another means to check the I/O status.

In DIRECT mode, use the following commands to display the status of all 16 inputs and outputs, respectively:

Type: **show din <Enter>** Shows status of the inputs.

Type: **show dout <Enter>** Shows status of the outputs.

The display will indicate the I/O status in the following manner:

```

1>16: 0 1 0 1 0 0 0 0 0 1 0 0 1 1 0 0
O.K.

```

When editing a program, use the command PRINTLN to display the status of a specific input or output during program execution. For example:

println in[5] When this command is encountered during program execution, either 1 or 0 will be displayed (that is, the value of variable IN[5]), depending on the state of input 5;

Inputs

Conditional commands, such as IF and WAIT, are used to read and respond to the state of the inputs. For example, you can use the following routine in a program:

if in[3]=1	If input 3 is ON, then
move A31	Move to position A31.
else	If input 3 is NOT ON (off), then
move A32	Move to position A32.
endif	End of conditional routine.

Outputs

As with inputs, conditional commands can read and respond to the state of the outputs. Commands can also be used to alter the state of outputs.

To change the state of an output—in both DIRECT and EDIT modes—use the SET command. For example:

<code>set out[6]=1 <Enter></code>	Turns ON input 6.
<code>set out[8]=0 <Enter></code>	Turns OFF input 8.

Activating Output-Driven Devices

Pneumatic devices and end effectors are connected to controller outputs and controlled by means of **ACL** output commands.

Assuming a pneumatic device is connected to controller (relay) output 2, use the following command format

<code>set out[2]=1 <Enter></code>	Turn on output 2 to activate device.
<code>set out[2]=0 <Enter></code>	Turn off output 2 to stop device.

In order to activate the pneumatic device from the teach pendant, you need to create two programs (named DVON and DVOFF, for example) which can be called from the teach pendant by means of the **Run** key. Each program contains one of the commands shown above.

- Program DVON contains the command to turn on output 2.
- Program DVOFF contains the command to turn off output 2.

Using the **ACL** command DIR note the identity number of programs DVON and DVOFF. (The command DIR is explained more fully later in this chapter.)

Let's assume programs DVON and DVOFF are identified as program 8 and program 9, respectively. Now, whenever you want to activate the device by means of the teach pendant:

Press: **[Run] [8] [Enter]**

Sample Program: OUT

The following program, named OUT, turns the controller outputs on and off, in succession. Write and run this program. Watch the output LEDs on the controller when you execute this program.

```
PROGRAM      INOUT
*****
DEFINE I
FOR          I = 1 TO 16
  SET OUT[I] = 1
  DELAY     20
ENDFOR
FOR          I = 1 TO 16
  SET OUT[I] = 0
  DELAY     20
ENDFOR
GOTO        1
```

Sample Program: INOUT

The following program, named INOUT, demonstrates program loops and conditional routines. The program contains two loops; one loop has instructions for checking the status and responding to the state of all the inputs; the other loop has instructions for responding when input 16 is on.

In addition, this program listing shows how to include user comments within a program.

Explanatory notes are provided below.

```
PROGRAM      INOUT
*****
DEFINE I
PRINTLN     "this program tests inputs & sets outputs"
PRINTLN
LABEL       1
FOR          I = 1 TO 16
  IF         IN[I] = 1
    * TEST IF INPUT I IS ON
    SET      OUT[I] = 1
    * SET OUTPUT I ON
  ELSE
    SET OUT[I] = 0
```

```

        * SET OUTPUT I OFF
    ENDIF
    DELAY    300
ENDFOR
IF          IN[16] = 1
    * IF INPUT 16 IS ON EXIT FROM PROGRAM
    SET OUT[16] = 0
    PRINTLN    " program inout stopped "
    PRINTLN
    GOTO2
ENDIF
GOTO      1
LABEL     2

```

- PRINTLN comments will be displayed on the screen during program execution.
- PRINTLN without a comment or argument simply enters a carriage return, and brings the screen cursor to the beginning of the next line.
- The variable I is used as the counter for 16 loops.
- FOR starts a program loop which checks state of all 16 inputs.
- The first IF command starts a conditional routine with instructions for responding to the state of an input: if an input is turned on, the output of the same index is also turned on; if the input is turned off; the output is turned off.
- The asterisk * precedes a user comment within a program; the comment is not displayed during program execution.
- ENDIF ends the IF conditional routine.
- ENDFOR ends the FOR loop.
- The second IF command starts a routine which checks and responds to the state of input 16. If input 16 is on, output 16 will not light; the program will go to label 2 and terminate.
- If input 16 is off, the program will go to label 1 and repeat.

When you run program INOUT, simulate an external input by manually shorting the input terminals.

- If the input is configured as sink: short the input to COM+ of the same block.
- If the input is configured as source: short the input to COM- of the same block.

When you short any of inputs 1 through 15, the output with the same index (1–15) will turn on. When you short input 16, the program will stop. Note the messages on the screen during program execution.

Program Directory

The programs you have created are stored in the controller's battery-backed RAM (BBRAM). To view the list of programs in the controller, use the DIR command in DIRECT mode. For example:

Type: `dir <Enter>`

```
name : validity      : identity : priority
AAA  :              : 1      : 5
LOOP :              : 2      : 5
OUT  :              : 3      : 5
INOUT :            : 4      : 5
```

... and so on.

- **Validity:** If the program is valid no message appears. "Not valid" will appear if the program contains a logic error, such as a FOR command without an ENDFOR command.
- **Identity:** This is the controller-assigned program identity number, which is needed for executing a program from the teach pendant. (Since certain controller operations can cause program identity numbers to change, use the DIR command at the beginning of each working session to verify the identity of program which you may want to call from the teach pendant.)
- **Priority:** By default the controller assigns each program a run-time priority of 5, on a scale of 1–10. The user can define a program's priority by means of the PRIORITY or RUN command.

Multi-Tasking

Controller-B is a multi-tasking real-time controller; it can simultaneously execute and control 40 user-defined tasks.

Before you attempt to run your programs simultaneously, edit the LOOP program, and insert the following line after the PRINTLN "LOOP" command line.

```
wait out[4]=1
```

(This will cause the program to pause, allowing you to observe the effect of each program and to enter commands.)

To run programs concurrently, use a RUN command to start execution of each program. For example:

Type: `run aaa <Enter>`
`run out <Enter>`
`run loop <Enter>`

Program AAA takes the robot through a series of movements. Program LOOP causes text to be displayed on the screen. Program OUT turns controller outputs on and off. Watch the robot, the computer screen, and the LED display on the controller while these programs are being executed.

To abort all three programs, use an Abort command.

Displaying Program Status

While programs are running, use the command STAT to view their status. (You can enter the command when the LOOP program pauses.)

Type: **stat <Enter>**

JOB_NAME	PRIORITY	STATUS	POSITION
AAA		5	PEND 7: MOVED
OUT		5	DELAY 56: DELAY
LOOP	5	DELAY	15: WAIT

- PEND: program is executing a movement command.
- DELAY: program execution is currently being delayed.
- SUSPEND: execution has been halted by SUSPEND command.
- POSITION: program's current line number and command.

Activating a Program from Another Program

As indicated throughout this chapter, **Controller-B** enables interaction and synchronization of programs.

Simultaneous Execution

The RUN command can be included in a program in order to start execution of another program. When a running program encounters a RUN *prog* command, both program are executed concurrently.

When several programs are running, those with a higher priority have precedence; those with the same priority share controller CPU time by means of an equal distribution algorithm.

Program Subroutines

Since two programs may conflict with one other, it may be preferable to use the GOSUB command rather than RUN.

Like RUN, the GOSUB command is used to start execution of another program. Unlike RUN, however, when a program encounters a GOSUB *prog* command, the program is suspended until the called program has completed execution. At that point, the first program resumes execution from the line which follows the GOSUB command.

Triggering a Program by I/O

The TRIGGER command can be used to execute another program when a specified input or output is turned off or on. However, it will activate the program only once, regardless of subsequent changes in the I/O state.

Saving a Program (Backup) to Disk

The programs, positions and variables used in your programs will remain stored in the controller's BBRAM. These files are not erased when the controller is turned off, but their contents may be totally or partially erased during certain configuration and restore procedures.

In order to save them to disk, perform the following steps.

1. From the DIRECT mode, press <Shift>+10. The **ATS Backup Manager** menu will appear on your screen.
2. Use the arrow keys to highlight "Backup PROGRAMS" and press <Enter>.

Backup directory: type and <Enter> the name of the drive where you want the file to be saved (it may be a floppy drive disk, or a subdirectory on your hard disk).

File name: type MYPROGS (or any name of up to 8 characters) and press <Enter>. The CBU extension will automatically be written.

3. Press F3 to save the file to disk.

When "DONE" appears, press <Esc> to return to the main **ATS** screen.

Note that this procedure saves *all* programs, positions and variables which are currently in the controller's BBRAM to the file MYPROGS.CBU.

Downloading a Program (Restore) to Controller

Files which have been saved to disk can be downloaded to the controller as needed.

The following steps will download the contents of a file to the controller BBRAM.

1. From the DIRECT mode, press <Shift>+10. The **ATS** Backup Manager menu will appear on your screen.
2. **Backup directory:** Type and <Enter> the name of the drive where the CBU file is located ; it may be a floppy disk drive, or a subdirectory on your hard disk. (You can press F9 (CATALOG) to make sure the file is in the directory.)

Use the arrow keys to highlight “Restore PROGRAMS” and “ADD TO Controller Contents.” Press <Enter> to accept these options.

File name: Type MYPROGS (for example), and press <Enter>. The CBU extension is not needed.

3. Press F5 to load (RESTORE) the file from disk to the controller BBRAM. When “DONE” appears, press <Esc> to return to the main **ATS** screen.



CHAPTER 9

Theory of Control

The function of the controller is to instruct the movements of the robot arm or other devices in the robotic system, to monitor these movements, and to make adjustments automatically in order to correct any errors.

Servo Control

Open Loop Control

In open-loop (non-servo) control, the system does not check whether the actual output (position or velocity) equals the desired output.

In open-loop control systems the controller output signal (U_r) is determined only by the input signal (r). If the system response is incorrectly predicted, or if the output signal is affected by other factors, deviations from the desired state will occur. Since no feedback exists, the system is unable to correct output errors.

In open loop robotic control, power is applied to the motors according to a predefined program. The path and speed cannot be precisely predicted, since they are determined by the torque and load on the motors, and other environmental factors.

Closed-Loop Control

In closed-loop control, the control system measures the output signal (C), compares it with the input (desired) signal (r), and corrects any errors.

Figure 9-1 compares schematic diagrams of open-loop and closed-loop control systems.

In servo control systems, a feedback device, commonly an optical encoder, measures the output (C) (the amount, speed and direction of motor rotation), converts it to an output signal (U_b), and transmits it to the comparator.

A comparator (\otimes) connects the input and feedback signals, produces an error signal equal to the algebraic difference of its two input signals. The comparator output—the error signal—is denoted here as U_e .

The error signal is the most important value in the closed-loop system. The system aims to reduce U_e to the smallest possible value. When $U_e = 0$, the output signal (the actual state) is equal to the input signal (the desired state).

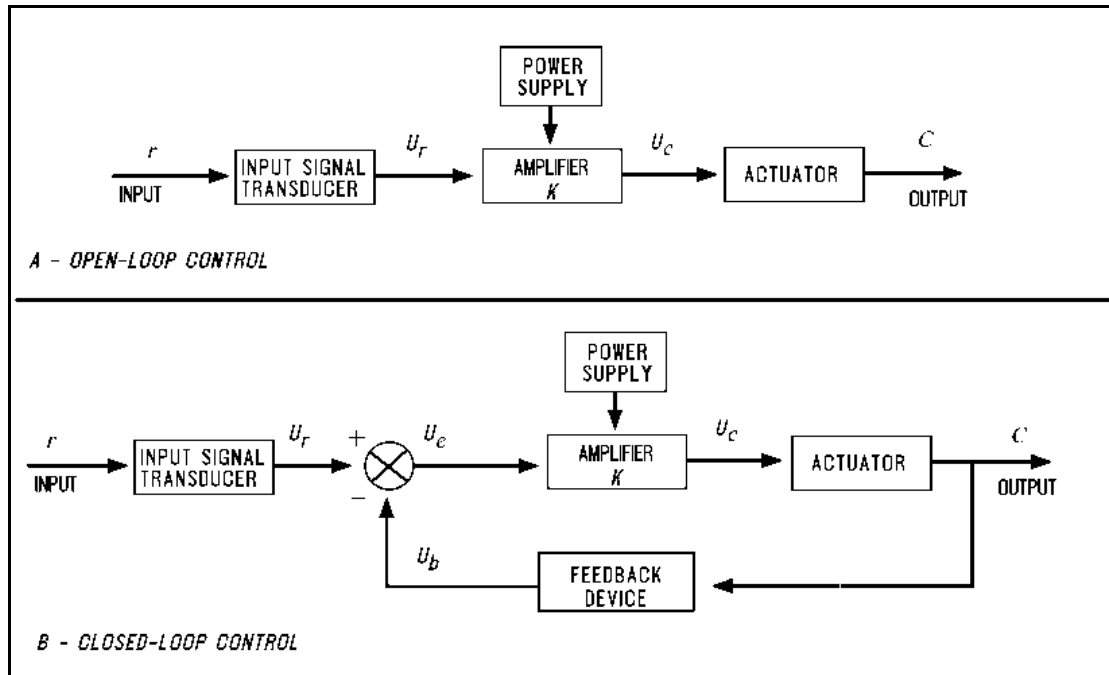


Figure 9-1: Open-Loop and Closed-Loop Control

Digital Control

Unlike analog control systems, in which all signals within the controller are continuous analog signals, digital control systems are those in which some of the signals within the controller are discrete digital signals, due to the presence of microprocessors.

In digital control systems, the controller must be capable of converting between analog and digital signals. For the microprocessor to read an analog signal, the signal must first pass through an Analog to Digital Converter. The ADC samples—that is, reads—the signal at periodic intervals and stores the value for the processor to read. For the microprocessor to transmit an analog signal, it must send the discrete values of the signal to a Digital to Analog Converter. The DAC holds the output continuously until given a new value.

Controllers use microprocessors to calculate the *state* (position, velocity, etc.) *error* (e) for each motor and the *control signal* (U_c) which is sent to the motors to correct the error. The control signal is converted to an analog signal by a DAC and then amplified before driving the motor.

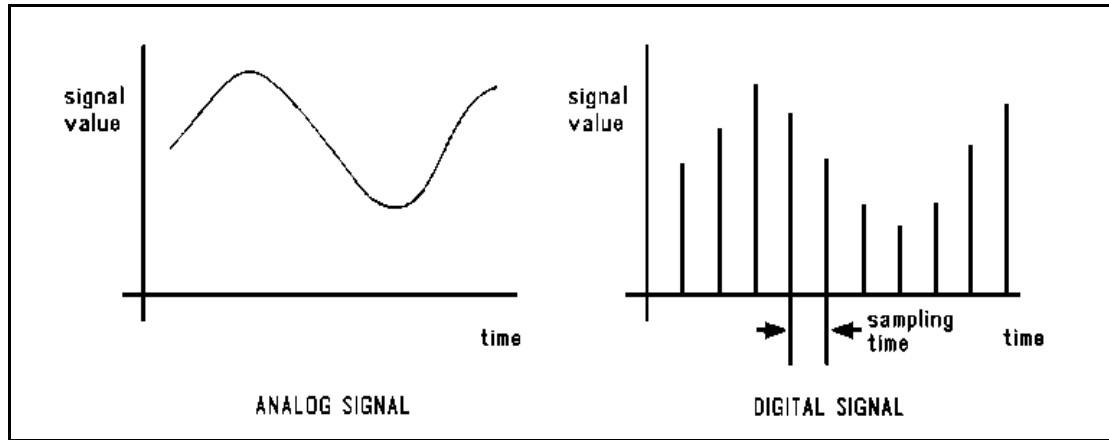


Figure 9-2: Analog and Digital Signals

The feedback device measures the actual state and produces an analog signal. The feedback signal is converted by a ADC so that the processor can read it to compute e .

A digital control system can be programmed to compute any number of control equations. The processor's control program is a continuous loop whose basic steps are as follows:

1. Read desired state from memory.
2. Read actual state from feedback device.
3. Calculate the state error (e).
4. Calculate control signal from control equation.
5. Go back to step 1.

The main difference between digital and analog controllers is the time delay caused by the processor's computations. This time delay is, in effect, the sampling time of the DAC and of the output control signal it produces. If the processor can complete a loop within a few milliseconds, the sampling time will be rapid, and the digital controller will produce an output similar to the equivalent analog controller.

On the other hand, if the processor is slow to make the computations, the controller will be unaware of fast changes in the feedback signal and the control signal will be based on "old" measurements. The greater the delay, the more the response will oscillate, eventually becoming unstable.

Transient and Steady State Responses

When the desired input signal (r) changes suddenly, the system will react in two phases, as shown in Figure 9-3. The initial reaction to a change in the input signal is called the *transient response*. The second part of the reaction is known as the *steady state response*. Once the input signal (r) has remained constant for some time, and the error between the input and output signals has stabilized, the system is said to be in steady state. The transition from transient to steady state is not a cleanly defined break.

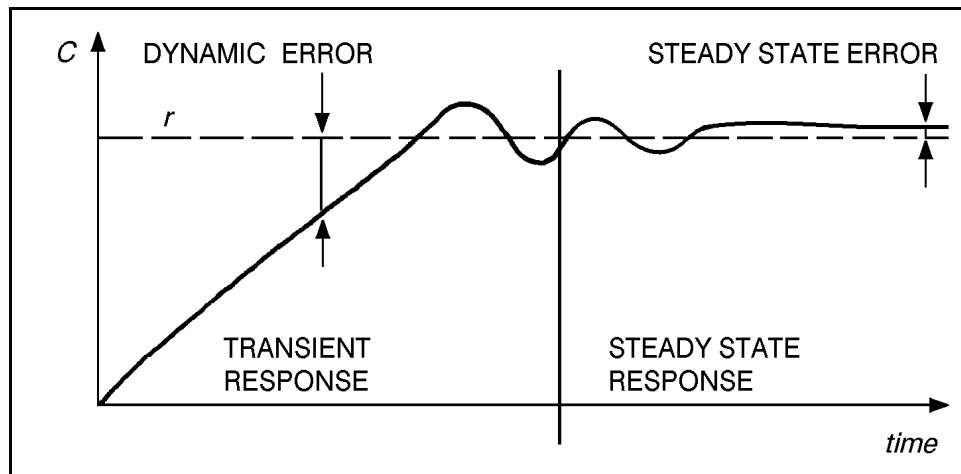


Figure 9-3: Transient and Steady States

This constant error, known as the *steady state error*, should be reduced as much as possible by the control system.

Increasing the amplitude of the controller output signal (that is, increasing the controller gain) can reduce the steady state error and enable a more rapid approach to the steady state value. The greater the controller gain, the faster the system reacts.

However, excessive gain may lead to a phenomenon called *overshoot*—a rise in the controlled value to a point above the desired value, followed by a drop below the desired value, repeated several times before stabilization. This, in effect, causes the actual value to oscillate around the desired value. Further increase of the controller gain may lead to instability of the entire system— that is, uncontrollable *oscillation*.

A control system is *damped* when it reaches steady state without overshoot. A *critically damped* response is the fastest approach to steady state without overshooting; an *overdamped* response is a slow approach to steady state.

Figure 9-4 shows different transient responses.

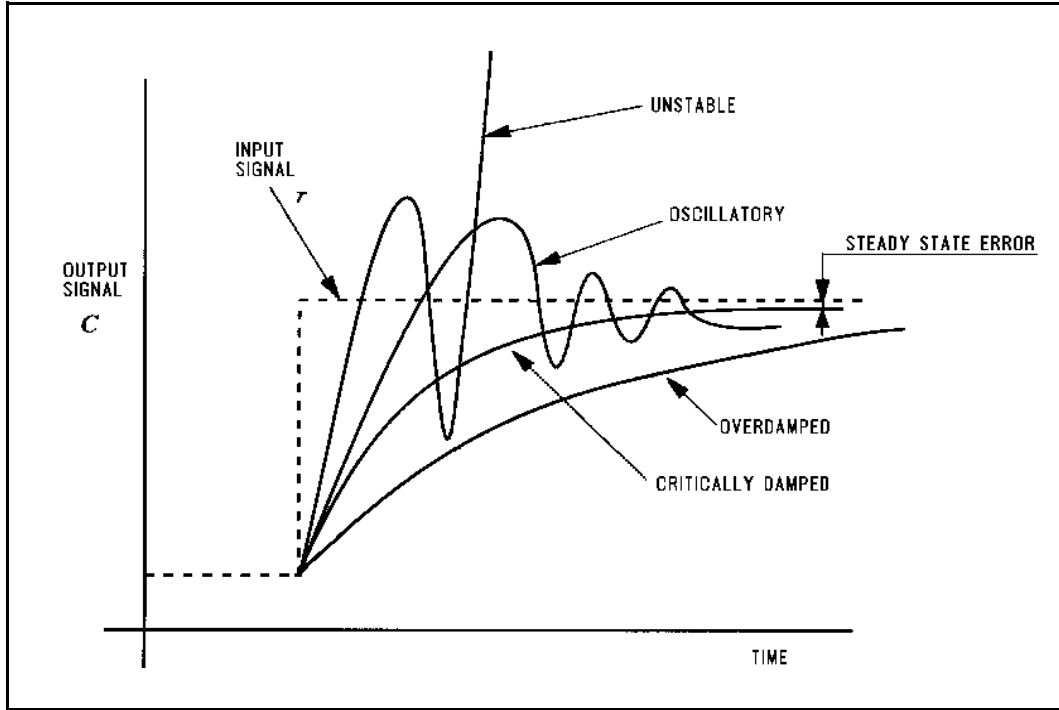


Figure 9-4: Transient State Responses

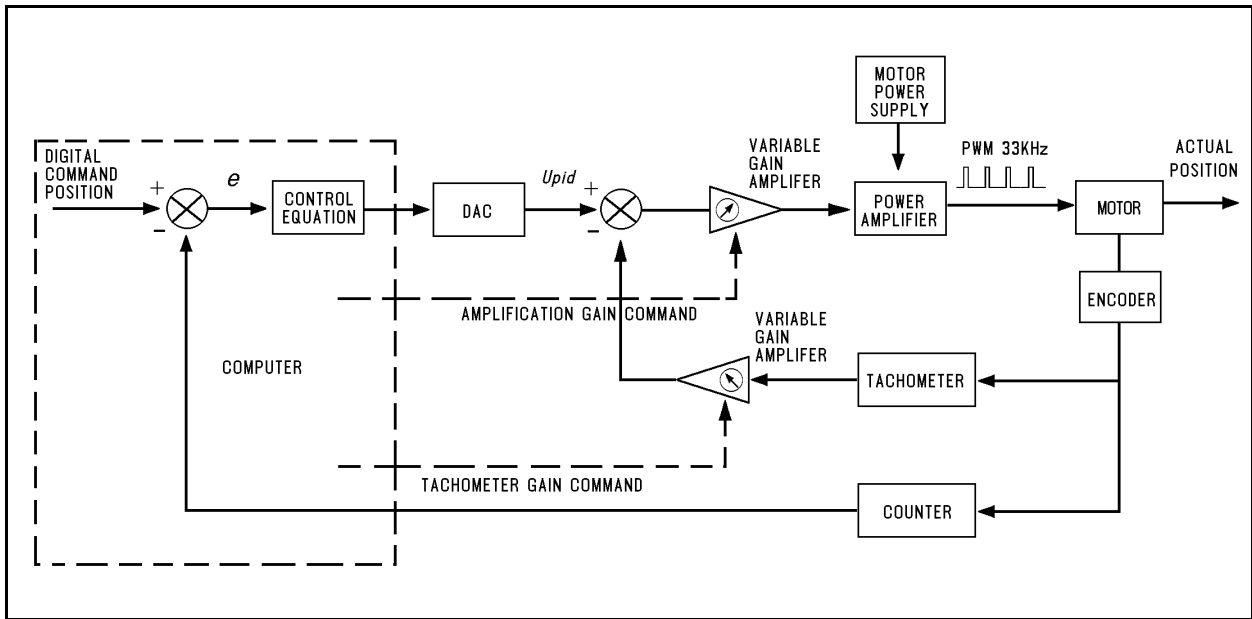


Figure 9-5: ACL Controller-B Control Loop

ACL Controller-B Control Process

The basic steps of the **Controller-B** control loop are described below. Refer to Figure 9-5. The entire control cycle takes about 10ms.

1. The processor calculates the command position and speed once per cycle. It outputs a digital value to the DAC unit in the range of ± 5000 .
2. The DAC unit output is compared to the tachometer output and the difference is amplified to give a PWM output to the motor.
3. The power amplifier drives the motor by switching $\pm 37V$ to it at 33KHz, in a duty cycle proportional to the output of the gain amplifier. The motor cannot react to this high frequency of switching and is therefore affected by only the average value of the voltage.

This method of controlling the time during which current flows through the motor, rather than controlling the value of the current, is known as PWM (Pulse Width Modulation) control. Refer to Figure 9-6.

4. As the motor rotates, the encoder attached to it produces a number of pulses proportional to the amount of rotation.

Once per cycle the processor reads the encoder pulse count and calculates the motor's position and speed (rate of encoder counts).

The processor then compares the actual (output) position and speed values with the desired (input) ones, determines the error values and takes the necessary action to cancel them.

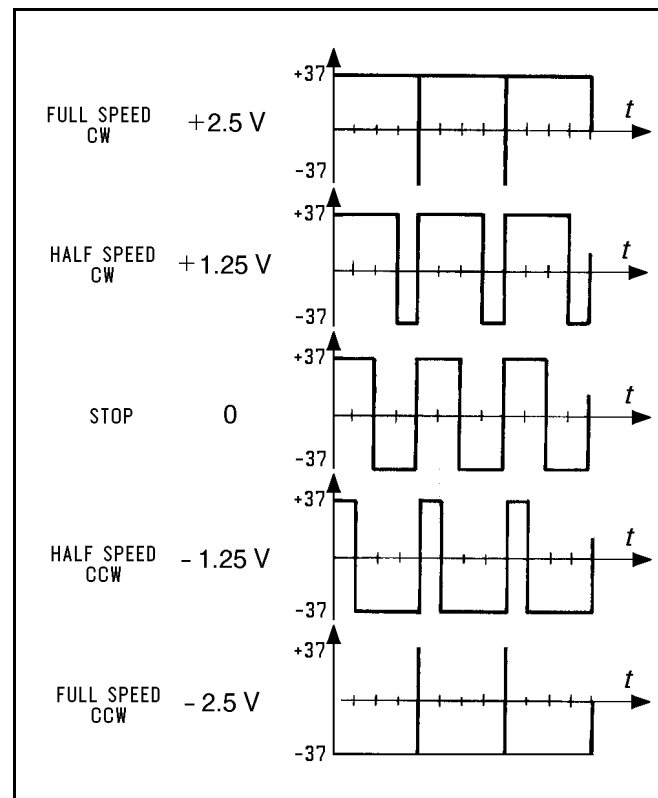


Figure 9-6: ACL-B Controller Control Signals

Trajectory Control

For better path performance (that is, to accurately reach the desired state and avoid overshoots), trajectory control profiles, may be programmed into the control system. **Controller-B** offers two profiles: paraboloid and sinusoid. Refer to Figure 9-7.

Paraboloid

The paraboloid profile causes the motors to accelerate slowly until maximum speed is reached, then decelerate at the same rate.

Sinusoid

The sinusoid profile causes the motors to accelerate and decelerate quickly at the start and end of movement, with a constant speed along the path.

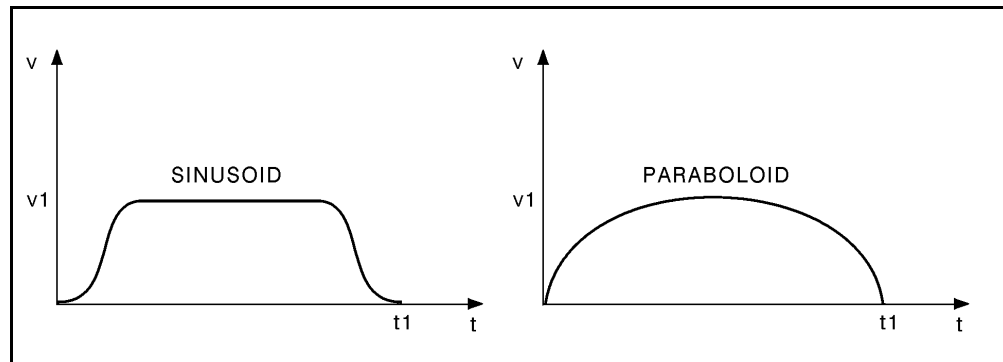


Figure 9-7: Trajectory Control Profiles

Path Control

It is desirable that the path and speed of a robot between taught points be predictable. Ideally, the path between consecutive points is traversed at a constant velocity with defined acceleration and deceleration segments.

Along the path, motion of all joints should be proportional, so that all the joints start and finish moving at the same time. The method of coordinating the movement of the joints so that all joints reach the desired location simultaneously is termed *joint interpolation*.

Point-to-Point Control

Point-to-point control (PTP) involves the positioning of the robot's end effector at given points, without defining the exact path of the end effector between any two points.

Point-to-point control is suitable for applications which require an exact and static position of the end effector at the points where operations will be performed.

In principle, point-to-point control can be used to guide the robot through a large array of positions, thus resulting in a complex path. In order to obtain such a path, points must be defined and recorded in a very close sequence. The number of positions will be limited, however, by the capacity of the control system to maintain positions in memory.

Continuous Path Control

Continuous path control (CP) involves the movement of the end effector between two points along a path defined by a mathematical formula. This method of control is suitable for applications in which the end effector executes operations along a precise trajectory.

During program execution, the control system calculates and plans the path, and instructs the robot motors to move accordingly.

When continuous path control is required, the processor divides the path into short segments, and interpolates the motion of the joints as frequently as possible.

Three type of CP control are possible.

- *Joint Control*: Each axis moves according to the trajectory profile. The gripper path is not defined; only the start and end points are defined. All axes start and stop movement at same time.
- *Linear Path Control*: The axes are coordinated in order to move the TCP (tool center point; tip of the gripper) in a straight line according to the trajectory profile.
- *Circular Path Control*: The axes are coordinated in order to move the TCP along a circular path according to the trajectory profile.

The Control Parameters

In the robotic system controller by **Controller-B**, as in common in closed-loop systems, the controlled value (C) is measured by an optical encoder. The encoder signals serve as feedback to the controller, enabling it to correct any deviations from the desired value.

Since control systems cannot react immediately to the input signal, there will always be a lag between the generation of an error signal and the actual correction of the controlled value.

The PID (proportional, integral, differential) control parameters allow the controller to adapt to various conditions of operation, such as overcoming nonlinear functions in the system.

Proportional Control

The proportional parameter is the gain of the control system. Its value determines the reaction amplitude to position errors.

When a position error exists (that is, the actual motor position is off by a certain amount of encoder counts), the processor multiplies the error by the proportional parameter and adds the product to the DAC value, thereby reducing the error.

The proportional parameter is the parameter in the PID control system which acts most quickly in reducing the position error, especially during motion. It is also the first parameter to respond to position errors when the robot has stopped at a target position.

The greater the proportional parameter, the faster the system responds and reduces the error. But, using too great a value for the proportional parameter will cause the axis to oscillate.

The main disadvantage of proportional control is that it cannot completely cancel the error, because once it has reduced the error it cannot generate enough power to overcome friction in the system and propel the axis to its target position.

Even in steady state, under load, the controlled value (output signal) will always be different from the desired value (input signal). The steady state error can be reduced by increasing the gain, but this will increase the oscillation and reduce stability.

Differential Control

In differential control, the controller output (C) is a function of the rate at which the error (U_e) changes. The faster the rate of change of the error, the greater the controller output (C). In other words, the controller is sensitive to the slope of the error signal.

The differential parameter is responsible for reducing the speed error. The control system calculates the actual speed once per cycle and compares it to the desired value. While the robot is accelerating (during the first part of path) the differential acts as a driving factor.

While the robot is decelerating (during the second, and last, part of path), the differential acts as a braking factor. A good differential setting will result in a clean and smooth motion along the entire path. Lack of the differential will cause overshoot at the end of path. High differential values will cause small vibrations along the path.

In this control method, the controller predicts the value of the error in accordance with the error signal slope, and causes the correction to take place in advance. However, if the error is constant and unchanging, differential control will not be able to reduce the error to zero.

Integral Control

In integral control, all the state errors which have been recorded each cycle are totalled and their sum is multiplied by the integral parameter value.

In integral control, the controller output (C) reduces the error signal (U_e) to zero at a rate proportional to the size and duration of the error. In other words, the greater the error, the greater the controller output; and, the longer the duration of the error, the greater the controller output.

The main advantage of integral control is that the steady state error is always reduced to zero since its value increases each cycle, thus strengthening the control system's ability to react and reduce the error. However, using too great a value for the integral parameter may cause overshoots, while too small a value may prevent the cancellation of a steady state error.

Unlike the proportional parameter, the integral parameter takes effect more slowly and is less noticeable during motion. However, when the axis comes to a complete stop and the proportional parameter can no longer reduce the steady state error, the integral parameter takes over and can cancel the error completely

Proportional–Integral–Differential Control

The PID control method enables optimal exploitation of all three types of control—proportion, integral and differential. In this manner, it creates an output response which follows the input signal closely, without gaps or lags, in both slow and rapid processes, including those in which the load is in a constant state of change. In summary, the PID control parameters serve the following functions:

- *Proportional Parameter*: Enables fast and powerful reactions of the arm to movement commands. Responsible for the repeatability of the motion.
- *Integral Parameter*: Assists the proportional parameter in eliminating steady state errors.
- *Differential Parameter*: Provides the required damping.

Changing Parameter Values

The control system parameters of **Controller-B** are accessed by user commands. However, most parameters may be altered only when the password-protected **PRIVILEGE** mode is active. This feature serves to prevent accidental or incorrect manipulation of servo and other critical parameters.

For more details see the *ACL Reference Guide*.



APPENDIX A

Fuses



Warning! Before you remove any fuses, turn off the controller. Before you remove any AC power fuses, disconnect the power cable from the AC power source.

SYSTEM FUSES		
Fuse Rating	Function	Location
2A (SB*) I/O User +24V	User power supply +24V	Power Supply Unit cover
2A (SB) I/O User +12V	User power supply +12V	
0.5A (SB)	Fan, Power Relays	
0.1A -2VDR	Driver, H-bridge	
1.25A +12VDR		
0.4A -12VA	Analog logic	
0.4A +12VA		
0.6A -12V	Auxiliary boards (RS232)	
0.6A +12V		
2.5A +5V1	Auxiliary boards (RS232)	
2.5A +5V	Main board, CPU board	
Variable fuse rating -35VDC to -40VDC See following table.	2 fuses on each card; one fuse per axis: Upper fuse: axes 1,2,3,7,9,11 Lower fuse: axes 4,5,6,8,10,12	Rear Panel; on driver cards
6.3A (SB) 100/110/220/240V	2 main AC fuses	Rear Panel; next to power switch/inlet
1A 100/110/220/240V	AC voltage to Logic PSU	Rear Panel; next to power switch/inlet; marked F1 and F2

* SB: slow-blow fuse

DRIVER FUSE RATINGS		
Robot	Fuse Rating	Function
SCORA-ER 14	8A (SB)	Upper fuse: axes 1,2,3
	4A (SB)	Upper fuse: axis 7 Lower fuse: axes 4,5,6,8
SCORBOT-ER IX	8A (SB)	Upper fuse: axes 1,2,3,7
	4A (SB)	Lower fuse: axes 4,5,8
	2A (SB)	Lower fuse: axis 6
PERFORMER-MK2	10A (SB)	Upper fuse: axes 1,2,3
	4A (SB)	Upper fuse: axis 7 Lower fuse: axes 4,5,8
	2A (SB)	Lower fuse: axis 6

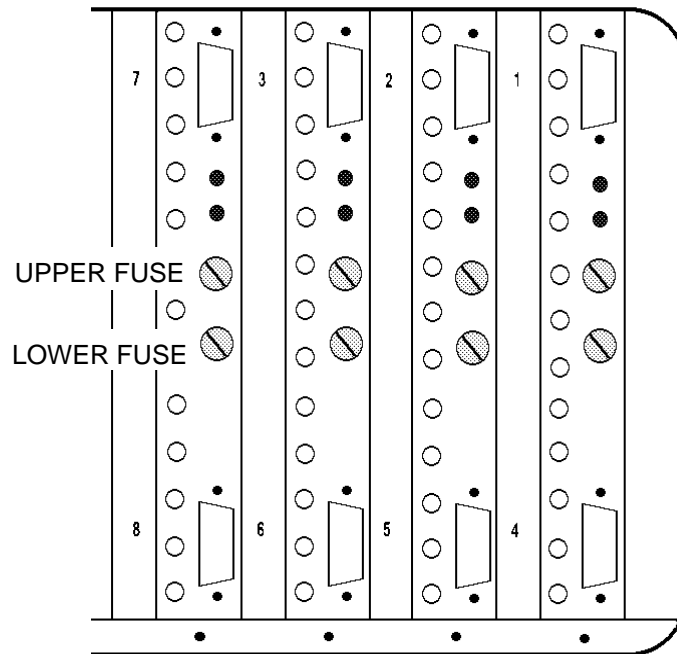


Figure A-1: Driver Card Fuses

APPENDIX **B**

Configuration Summary

JUMPERS			
Board	Jumper	Connection	Description
Main Board	JP1 JP2 JP3 JP4 JP5 JP6	JP7 JP8 OPEN* JP9 JP10 JP11 JP12	Reserved for future use.
CPU Board	JP1	1-2* 2-3	512KB RAM 1MB or 2MB RAM
Axis Driver Card	JP1	OPEN*	SCORBOT-ER IX, SCORA-ER 14
		CLOSE	SCORBOT-ER VII SCORBOT-ER Vplus
	JP2	OPEN*	SCORBOT-ER IX, SCORA-ER 14
		CLOSE	SCORBOT-ER VII, SCORBOT-ER Vplus
Connector Card	JP1	1-2*	crossed RS232 cable; CONSOLE port
	JP2	1-2*	
	JP4	1-2*	crossed RS232 cable; COM 0 port
	JP5	1-2*	
	JP1	2-3	non-crossed RS232 cable; CONSOLE
	JP2	2-3	
	JP4	2-3	non-crossed RS232 cable; COM 0
	JP5	2-3	
	JP3	OPEN	teach pendant connected
		CLOSE	teach pendant not connected
Auxiliary RS232 Card	JP1	1-2*	card installed

* Default factory setting.

SWITCHES			
Board	Switch	Setting	Description
Main Board	S1 – 5	SINK*/SOURCE	Sink/Source status of open collector outputs: Output 5
	S2 – 6	SINK*/SOURCE	Output 6
	S3 – 7	SINK*/SOURCE	Output 7
	S4 – 8	SINK*/SOURCE	Output 8
	S5 – 9	SINK*/SOURCE	Output 9
	S6 – 10	SINK*/SOURCE	Output 10
	S7 – 11	SINK*/SOURCE	Output 11
	S8 – 12	SINK*/SOURCE	Output 12
	S9 – 13	SINK*/SOURCE	Output 13
	S10 – 14	SINK*/SOURCE	Output 14
	S11 – 15	SINK*/SOURCE	Output 15
	S12 – 16	SINK*/SOURCE	Output 16
Main Board	S13 – 1	SINK/SOURCE*	Sink/Source status of inputs: Input 1
	S14 – 2	SINK/SOURCE*	Input 2
	S15 – 3	SINK/SOURCE*	Input 3
	S16 – 4	SINK/SOURCE*	Input 4
	S17 – 5	SINK/SOURCE*	Input 5
	S18 – 6	SINK/SOURCE*	Input 6
	S19 – 7	SINK/SOURCE*	Input 7
	S20 – 8	SINK/SOURCE*	Input 8
	S21 – 9	SINK/SOURCE*	Input 9
	S22 – 10	SINK/SOURCE*	Input 10
	S23 – 11	SINK/SOURCE*	Input 11
	S24 – 12	SINK/SOURCE*	Input 12
	S25 – 13	SINK/SOURCE*	Input 13
	S26 – 14	SINK/SOURCE*	Input 14
	S27 – 15	SINK/SOURCE*	Input 15
	S28 – 16	SINK/SOURCE*	Input 16
Main Board	SW1	1 ON* 2 ON* 3 ON* 4 ON* 5 ON* 6 ON*	Reserved for future use.
CPU Board	S1	1 ON* 2 ON*	512KB RAM
		1 OFF 2 ON	1MB RAM
		1 OFF 2 OFF	2MB RAM

* Default factory setting.

SWITCHES (cont'd)			
Board	Switch	Setting	Description
CPU Board	S1	3 ON* 4 ON*	512KB EPROM
		3 OFF 4 ON	1MB EPROM
		3 OFF 4 OFF	2MB EPROM
Input/Output Board	S1 S3 S4 S5	External*/Internal External*/Internal External*/Internal External*/Internal	PSU internal or external voltage reference Block A Block B Block C Block D
Input/Output Board PC17400 (REV.1)	S6	1 ON*	Remote Emergency Switch Block A, C
		2 ON*	Remote Emergency Switch Block B, D
Input/Output Board	S2	+24V* / +12V	Voltage level of internal PSU

* Default factory setting.

TEST POINTS on Main Board	
Test Point	
TP1	+12VA
TP2	-12VA
TP3	+12V
TP4	+5V
TP5	+5V1
TP7	Ground
TP8	-12V



APPENDIX **C**

Connector Pin Assignments

Connector	Pinout	Function
Console and COM 0 RS232 channels D-type 9-pin plug	1	Not used
	2	RX (receive data)
	3	TX (transmit data)
	4	Not used
	5	Not used
	6	Not used
	7	RTS (request to send)
	8	CTS (clear to send)
	9	Not used
Teach Pendant D-type 25-pin plug	1	+5V1
	2	GND
	3	TP-A1
	4	TP-ED/SP
	5	TP-R/W
	6	TP-D8
	7	TP-D9
	8	TP-D10
	9	TP-D11
	10	TP-D12
	11	TP-D13
	12	TP-D14
	13	TP-D15
	14	TPIN-D8
	15	TPIN-D9
	16	TPIN-D10
	17	TPIN-D11
	18	TPIN-D12
	19	TPIN-D13
	20	TP- IRQ TP
	21	TP-RESET
	22	TP-D14
	23	TP-D15
	24	TP-EMERG
	25	GND

Connector	Pinout	Function
Parallel Port D-type 25-pin socket	1	HS4/STROBE
	2	P-OUT0/DATA0
	3	P-OUT1/DATA1
	4	P-OUT2/DATA2
	5	P-OUT3/DATA3
	6	P-OUT4/DATA4
	7	P-OUT5/DATA5
	8	P-OUT6/DATA6
	9	P-OUT7/DATA7
	10	HS3/ $\overline{\text{ACK}}$
	11	P-IN0/BUSY
	12	P-IN1/PE
	13	P-IN2/SLCT
	14	P-OUT8/ $\overline{\text{AFD}}$
	15	P-IN3/ $\overline{\text{ERROR}}$
	16	P-OUT8/INIT
	17	HS2/SLCT-IN
	18	GND
	19	P-IN4
	20	P-IN5
	21	P-IN6
	22	P-IN7
	23	HST
	24	GND
	25	GND
Robot Power "Burndy" Metal or Bantam 19-pin socket	A	M1_A
	M	M1_B
	B	M4_A
	K	M4_B
	C	M2_A
	L	M2_B
	D	M5_A
	J	M5_B
	E	M3_A
	H	M3_B
	F	M6_A
	G	M6_B
	R	+24VDC
S	24V Ret.	

Connector	Pinout	Function	
Robot Encoders D-type 37-pin plug	1, 2	+5V	
	3, 4, 29	GND	
	30, 37	GND	
	5	1CHA	
	6	1CHB	
	7	1CHI	
	8	COM1	
	9	2CHA	
	10	2CHB	
	11	2CHI	
	12	COM2	
	13	3CHA	
	14	3CHB	
	15	3CHI	
	16	COM3	
	17	4CHA	
	18	4CHB	
	19	4CHI	
	20	COM4	
	21	5CHA	
	22	5CHB	
	23	5CHI	
	24	COM5	
	25	6CHA	
	26	6CHB	
	27	6CHI	
	28	COM6	
	31	HOME-MSW1	
	32	HOME-MSW2	
	33	HOME-MSW3	
	34	HOME-MSW4	
	35	HOME-MSW5	
	36	HOME-MSW6	
	Axis Driver D-type 9-pin socket	1	M-B
		2	+5V1
		3	5V-ENC
4		MSWITCH	
5		COMMON	
6		CHB	
7		CHI	
8		CHA	
9		M-A	

Connector	Pinout	Function
I/O Block A, C 36-pin Centronics socket	1	Emergency AC1
	2	+12V User PSU
	3	+24V User PSU
	4	PSU Common
	5	Output 5
	6	Output 7
	7	Output 9
	8	- COM BLA
	9	+COM BLC
	10	Input 2
	11	Input 4
	12	Input 6
	13	Input 8
	14	PSU Common
	15	Output 1 COM
	16	Output 2 NC
	17	Output 2 NO
	18	Not Used
	19	Emergency AC2
	20	+12V User PSU
	21	+24V User PSU
	22	+COM BLA
	23	Output 6
	24	Output 8
	25	Output 10
	26	PSU Common
	27	Input 1
	28	Input 3
	29	Input 5
	30	Input 7
	31	-COM BLC
	32	Output 1 NC
	33	Output 1 NO
	34	Output 2 COM
	35	PSU Common
	36	Not used

Connector	Pinout	Function
I/O Block B, D 36-pin Centronics socket	1	Emergency BD1
	2	+12V User PSU
	3	+24V User PSU
	4	PSU Common
	5	Output 11
	6	Output 13
	7	Output 15
	8	- COM BLB
	9	+COM BLD
	10	Input 10
	11	Input 12
	12	Input 14
	13	Input 16
	14	PSU Common
	15	Output 3 COM
	16	Output 4 NC
	17	Output 4 NO
	18	Not Used
	19	Emergency AC2
	20	+12V User PSU
	21	+24V User PSU
	22	+COM BLB
	23	Output 12
	24	Output 14
	25	Output 16
	26	PSU Common
	27	Input 9
	28	Input 11
	29	Input 13
	30	Input 15
	31	-COM BLD
	32	Output 3 NC
	33	Output 3 NO
	34	Output 4 COM
	35	PSU Common
	36	Not used
Remote Emergency Switch terminal	1	EMRG-IN
	2	EMRG-OUT
	3	FRAME-GND

Connector	Pinout	Function
Remote Servo Driver Command Output D-type 50-pin plug	1	AGND
	2	DAC0-OUT
	3	DAC1-OUT
	4	DAC2-OUT
	5	DAC3-OUT
	6	DAC4-OUT
	7	DAC5-OUT
	8	DAC6-OUT
	9	DAC7-OUT
	10	DAC8-OUT
	11	DAC9-OUT
	12	DAC10-OUT
	13	DAC11-OUT
	14	AGND
	15	AGND
	16	GND
	17	GND
	18	AGND
	19	AGND
	20	AGND
	21	AGND
	22	AGND
	23	AGND
	24	AGND
	25	AGND
	26	AGND
	27	AGND
	28	AGND
	29	AGND
	30	AGND
	31	AGND
	32	GND
	33	MOTORS-ON
	34	GND
	35	EMRG
	36	INH FAR0
	37	INH FAR1
	38	INH FAR2
	39	INH FAR3
	40	INH FAR4
	41	INH FAR5
	42	INH FAR6
	43	INH FAR7
	44	INH FAR8
	45	INH FAR9
	46	INH FAR10
	47	INH FAR11
	48	INH FAR ALL
	49	GND
	50	GND

Connector	Pinout	Function
Auxiliary RS232 multiport D-type 37-pin socket PC-7200	1	TD0 (transmit data)
	2	RTS0 (request to send)
	3	RX0 (receive data)
	4	CTS0 (clear to send)
	5	GND
	6	TD1
	7	RTS1
	8	RX1
	9	CTS1
	10	TD2
	11	RTS2
	12	RX2
	13	CTS2
	14	GND
	15	TD3
	16	RTS3
	17	RX3
	18	CTS3
	19	TD4
	20	RTS4
	21	RX4
	22	CTS4
	23	GND
	24	TD5
	25	RTS5
	26	RX5
	27	CTS5
	28	TD6
	29	RTS6
	30	RX6
	31	CTS6
	32	GND
	33	TD7
	34	RTS7
	35	RX7
	36	CTS7
	37	----



APPENDIX **D**

Emergency System

Emergency Activation

When the controller's or teach pendant's red EMERGENCY button is pressed, or when any remote emergency device or switch is activated, the system activates its emergency mode.

Refer to Chapter 3, "Controller Functions." for a complete instructions on activating and cancelling the emergency mode.

Remote Emergency Connections

Any number of stand-alone emergency buttons and devices with normally closed contacts can be configured for use with **Controller-B**. The devices may be protective covers, limit switches, alarms, safety mats, and so on.

As shown in Figure D-1, the switches and devices can be connected in either of two ways:

- By means of the Remote Emergency Switch terminal at the rear of the controller.

This terminal is normally closed. To make the connection, loosen screws 1 and 2, and remove the wire which shorts the two poles of the terminal. Then connect the wires of the emergency device to the Remote Emergency Switch terminal, and tighten the screws.

- By means of pins 1 and 19 on the I/O Centronics connectors at the rear of the controller.

When connecting an emergency device by means of the I/O block connectors, *switches S6-1 and S6-2 on the I/O board must be set to OFF.*

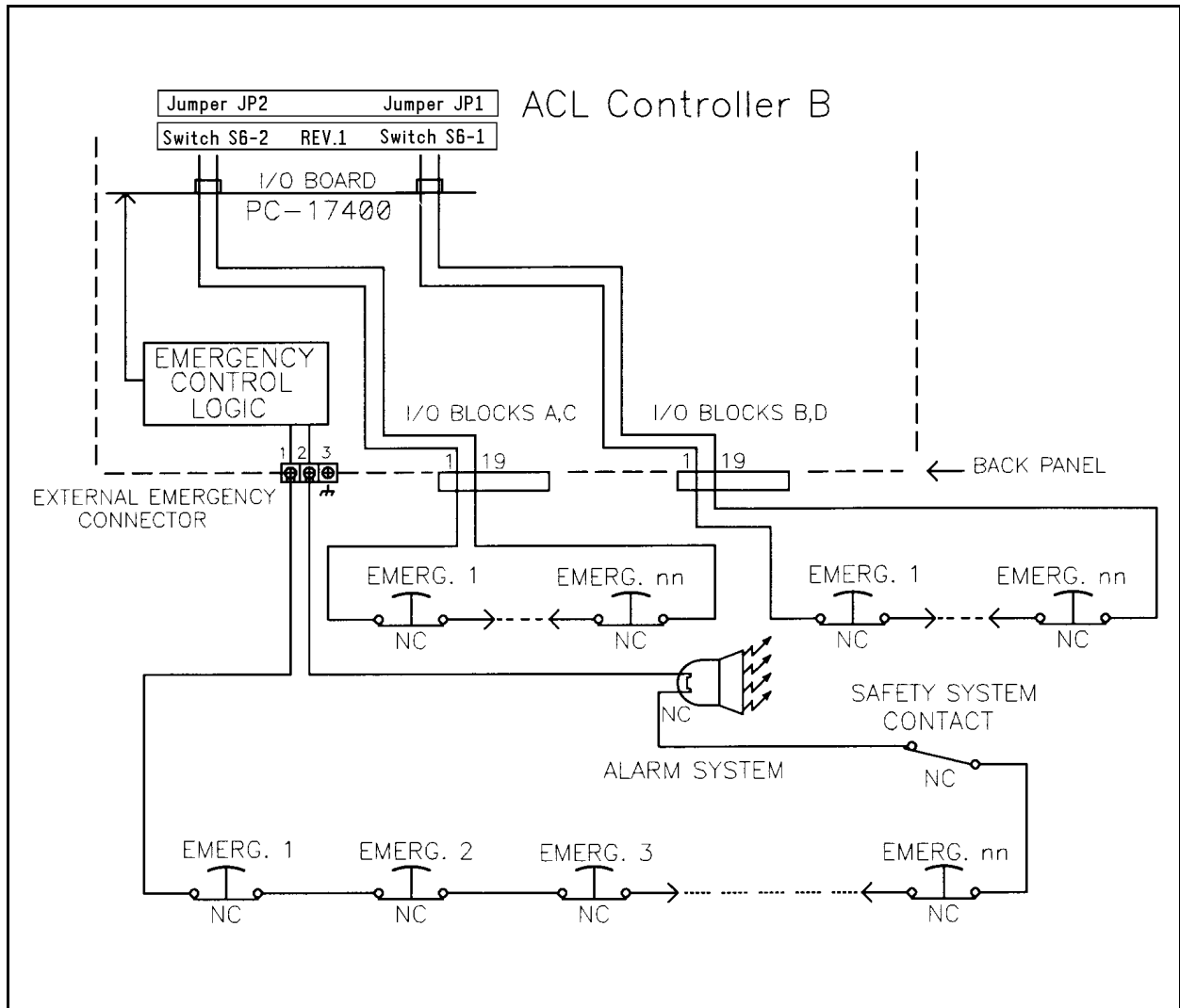


Figure D-1: Emergency System Circuits

