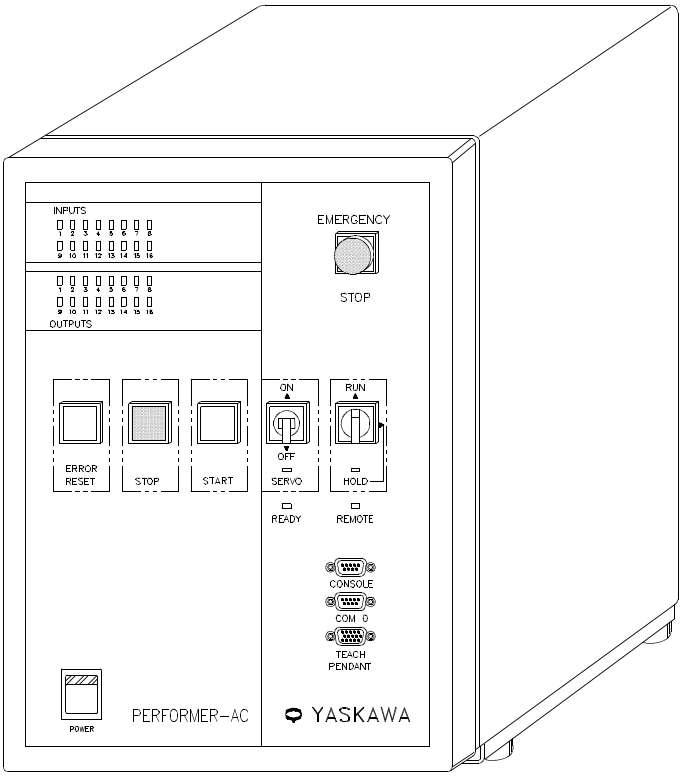
Controller-AC

for Performer-MK3

ACL version AC30.25

User's Manual

Catalog #100120 Rev.02



October 1998 (PDF version)

Catalog #100120 Rev.02

.

Table of Contents

List of Figures

CHAPTER 1

General Information

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.

Make sure you have received all the items listed on the shipment's packing list. If anything is missing, contact your supplier.

Following are the standard components in the **Controller-AC for MK3** package.

Standard Controller-AC for MK3 Package	
Controller	3 driver cards (for 5 axes); (Optional: driver cards for axes 7 and 8)
Cables	Power Cable
	RS232 Cable
ATS (Advanced Terminal Software)	1 diskette; includes **ACLoff-line** software
Documentation	***Controller-AC for MK3*** *User's Manual*
	ACL *Reference Guide*
	ATS *Reference Guide*
	ACLoff *line User's Manual*

The following are optional items for the **Controller-AC for MK3**.

Optional Items	
Item	Description
Teach Pendant	Supplied with: Mounting fixture; *Teach Pendant User's Manual*.
Direct Teaching Handles	Supplied with: *MK3 Direct Teaching Instruction Booklet*
Additional Axis Driver Card	Serves two axes (up to 8 axes supported per controller). Factory-installed if ordered with controller.
Auxiliary Multiport RS232 Board, Cable and Connectors	Card for additional RS232 channels: 2-port connector cable or 8-port connector cable Factory-installed if ordered with controller.

Controller Specifications

The following table gives the specifications of the **Controller-AC for MK3**.

Controller-AC for MK3 Specifications	
Item	Specification
Case	Free-standing, enclosed frame
Weight	20.4 kg
Dimensions	450mm (L) x 280 mm (W) x 403 mm (H)
Cooling	Indirect
Operating Temperature	0°–40°C
Relative Humidity	less than 90°; non-condensing
AC Power Input	200–230V AC; +10%, -15%; 50 Hz or 60Hz; 1300VA max.
Grounding	Less than 0.1 Ohm
Inputs/Outputs	16 inputs (12V–28V)
	16 relay contact outputs
Motor Driver	AC Servos
Position Feedback Devices	Resolvers with absolute positioning (Return to Home or Homing not required)
Number of Axes	5 for robot; additional 3 optional
Acceleration	S-curve profile (smooth acceleration)
User Memory	400 programs; 5000 program lines; 5000 positions (standard configuration)
Programming Language	ACL: Advanced Control Language - Full multi-tasking, up to 40 concurrent tasks for robot and cell control - Provides wide set of instructions for: - Sophisticated trajectory control. - Event driven on-the-fly change of trajectory - Torque control - Automatic Programs for: Restart; Emergency; Background Safety
Program Editing	On-line and off-line
Axis Servo Control	- Fully digital with resolver interface and brake control; - 0.5ms position loop and 127 ms current loop cycle time; - Operation from nominal 300 V DC bus; - Over-current and short-circuit protection; - Software detection of thermic overload of motor; - Emergency stop (limit switches, emergency switch, errors) - Error detection: over-current, DC bus over-voltage; DC bus under-voltage; resolver error; DC failure; watchdog. - Control parameters in EEPROM of each axis driver.

Controller-AC for MK3 Specifications	
Item	Specification
Trajectory Control	Joint, linear and circular interpolation; Continuous path with all movement combinations; On-the-fly change of speed and acceleration; On-the-fly change of trajectory; Path reversal.
Torque Control	User definable upper and lower torque limits (value and sign); User monitoring of output torque.
Cell Control	Emergency connector; 16 Inputs (12V–28V); 16 Outputs (Relay Contact); Multi-tasking (independent cell control while robot is working.)
Direct Teach (optional)	Lead by Nose handles and associated software: - Handle attached to elbow; - Handle attached to roll flange with safety switch and record switch. - Lead by Nose function. - Record positions.
Teach Pendant (optional)	Jogging, teaching positions, testing trajectory, running programs; 25 multifunction keys; 4 line display; Emergency and Dead Man switch.

Control Front Panel

1: Power On/Off switch with lamp.
2: Error Reset push button.
3: Error indicator lamp.
4: Program Stop push button.
5: Program Start push button.
6: Program Running indicator lamp.
7: Servo Enable toggle switch.
8: Servo Enable indicator LED.
9: Hold / Run selector switch.
10: Hold mode indicator LED.
11: Ready indicator LED (watchdog status: blink = fault; steady = ready).
12: Remote indicator LED.
13: Emergency switch with indicator lamp.
14: Input state indicator LEDs.
15: Output state indicator LEDs.
16: Console RS232 port for PC.
17: Auxiliary RS232 port (Com0).
18: Teach Pendant and Direct Teach connector.
19: Front panel fastening screws.

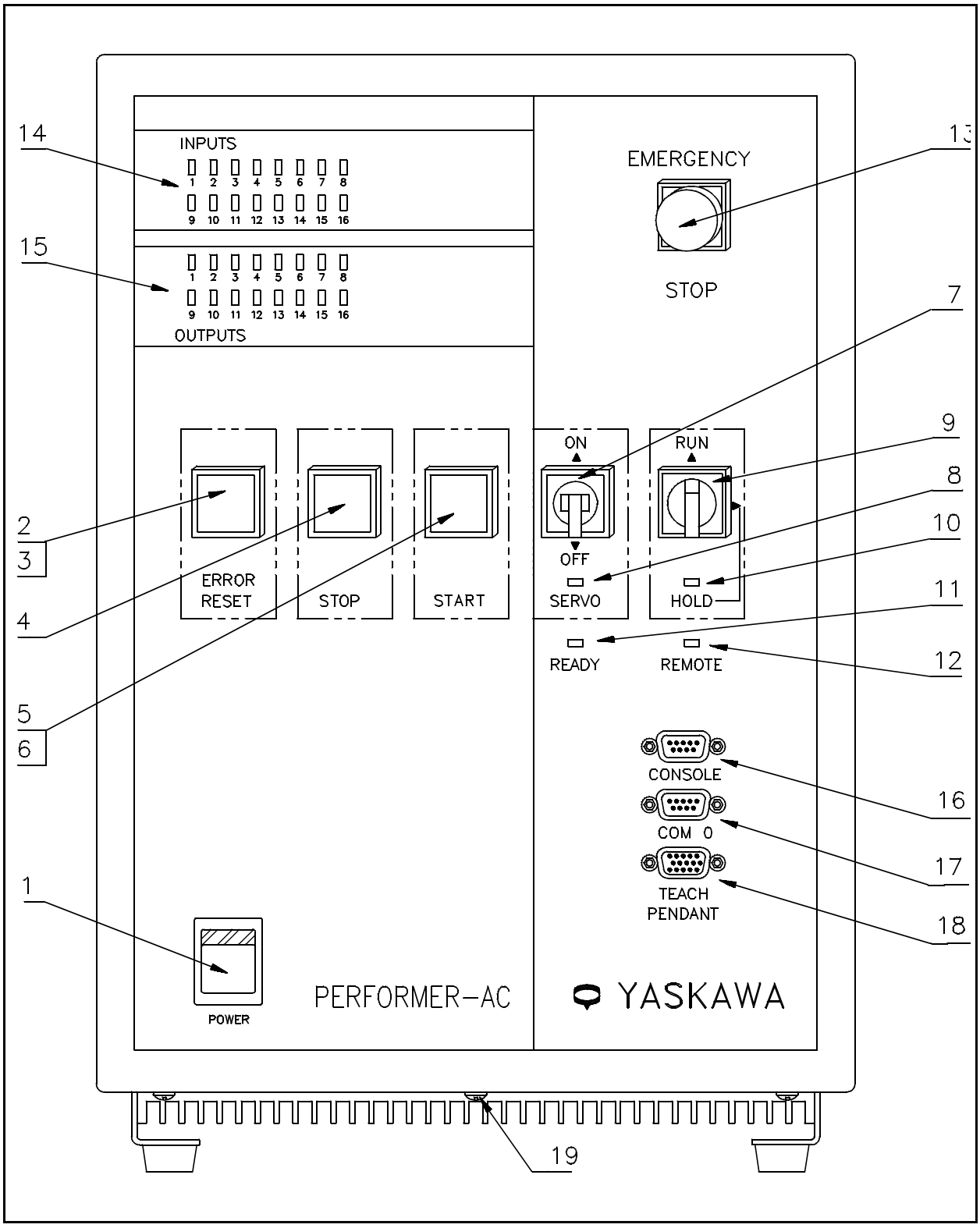


Figure 1-1: Controller Front Panel

Controller Back Panel

21: Power line socket (200-230V).
22: Fuse drawer (two fuses, 8A, slow blow).
23: Ground connection screw.
24: Robot Power connector.
25: Axis 7 Power connector.
26: Axis 8 Power connector.
27: Robot resolvers connector.
28: Axis 7 resolver connector.
29: Axis 8 resolver connector.
30: External Emergency connection terminals.
31: I/O connector block 1.
32: I/O connector block 2.
33: Parallel port.
34: Bracket screws for Dual Power Unit (DPU) cards.
35: Fan
36: Auxiliary RS232 multiport connector (optional)
37: Fan housing screws

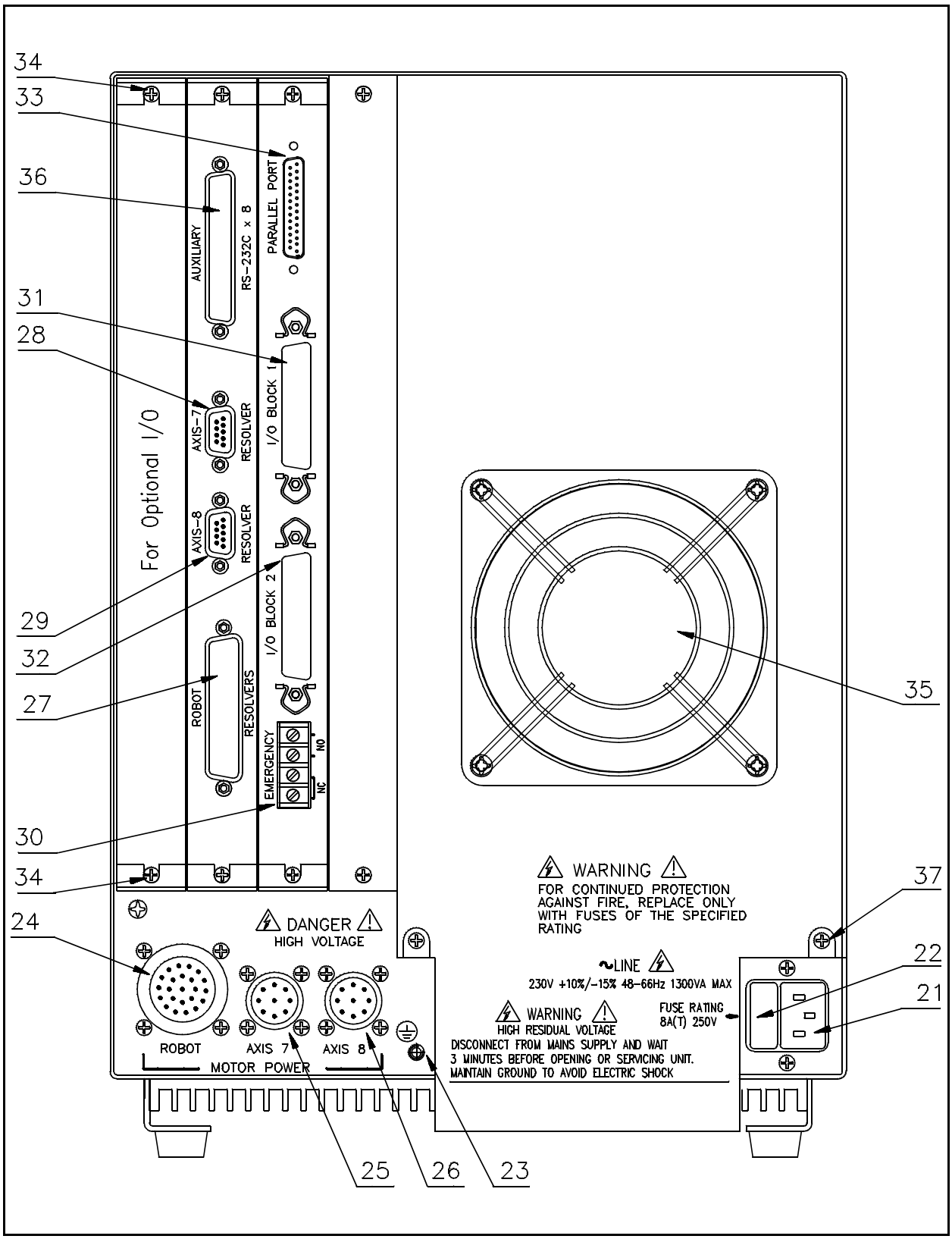


Figure 1-2: Controller Back Panel

Controller Functions

By default, the controller's front panel switches and lights operate in the Panel mode.

By configuring parameters and issuing the ACL command REMOTE, some panel functions can be operated in Remote mode. In Remote Mode, control of some of the controller's front panel functions are transferred to external switches and lamps by means of the system's digital inputs and outputs.

POWER Switch (1)

This switch turns the controller on and off.

A lamp in the POWER switch lights up when the controller is turned on.

☞ *After switching off, wait one minute before switching on again.*

ERROR RESET Push Button (2) and Error Indicator Lamp (3)

When an error occurs in one of the servo axes and the power to that axis and all axes of the same group are disabled, the lamp in the ERROR RESET button lights up.

In order to reenable control to all disabled axes, clear the error by pressing the ERROR RESET button.

To transfer the function of the Reset switch to an external input, set parameter 114 to the input number (LET PAR 114=*n*), and issue the command REMOTE.

To transfer the function of the Error lamp to an external output, set parameter 115 to the output number (LET PAR 115=*n*), and issue the command REMOTE.

STOP Push Button (4)

Pressing this button stops movement of all axes and aborts execution of all programs, except the background program BACKG (if it exists and is running).

The STOP differs from the EMERGENCY button in that it does not disable motors, does not activate the EMERG program, and does not abort the BACKG program.

To transfer the function of the Stop switch to an external input, set parameter 113 to the input number (LET PAR 113=*n*), and issue the command REMOTE.

START Push Button (5) and Program Running Indicator Lamp (6)

Pressing this button starts execution of an ACL user program named START, if it exists. If you have not created a START program, pushing this button will have no effect.

The lamp in the START button lights up whenever *any* program is running.

To transfer the function of the Start switch to an external input, set parameter 16 to the input number (LET PAR 16=*n*), and issue the command REMOTE.

To transfer the function of the Start lamp to an external output, set parameter 117 to the output number (LET PAR 117=*n*), and issue the command REMOTE.

SERVO ON / OFF Toggle Switch (7) and SERVO Indicator LED (8)

When this switch is in the OFF position, control of axes cannot be enabled.

When the SERVO switch is turned ON, the SERVO indicator LED is lit.

To transfer the Servo On function of the switch to an external input, set parameter 14 to the input number (LET PAR 14=*n*), and issue the command REMOTE.

To transfer the function of the Servo LED to an external output, set parameter 119 to the output number (LET PAR 119=*n*), and issue the command REMOTE.

HOLD / RUN Selector Switch (9) and HOLD Indicator LED (10)

When this switch is moved to the HOLD position, all movement of axes and all running programs are suspended. Returning that switch to the RUN position causes all suspended programs and movements to resume.

When the controller is in the HOLD state, the HOLD indicator LED is lit.

To transfer the Hold function to an external input, set parameter 15 to the input number (LET PAR 15=*n*); you do not need to issue the command REMOTE.

For safety purposes, the Hold function on the controller panel always remains active, even when Remote mode is active. This allows the Hold function to be used for safety switches in the work cell, as in safety doors and safety mats.

To transfer the function of the Hold LED to an external output, set parameter 118 to the output number (LET PAR 118=*n*), and issue the command REMOTE.

READY Indicator LED (11)

This light displays the state of an internal watchdog. If an internal error occurs and the controller is unable to function properly, this LED will blink. Normally this LED remains lit.

REMOTE Indicator LED (12)

When the controller is in the Remote Mode, this LED is lit.

To transfer the function of the Remote LED to an external output, set parameter 116 to the input number (LET PAR 116=*n*), and issue the command REMOTE.

EMERGENCY / STOP Push Button and Emergency Indicator Lamp (13)

Pressing this buttons results in the following:

- Immediately stops all arm movements.
- Disables servo control on all axes.
- Activates all motors brakes.
- Runs program named EMERG, if exists.

This button has a mechanical latch. You must turn the buttons slightly in either direction in order to release it.

Digital INPUTS Indicator LEDs (14)

Shows the present state of inputs. When lit, voltage is being applied to that input and the logical state is "1" in the ACL software.

Digital OUTPUTS Indicator LEDs (15)

Shows the present state of outputs. When lit, the contact is closed and the logical state is "1" in the ACL software.

CONSOLE RS232 Connector (16)

This D9 connector is the communication port for the host PC which is used for programming the controller.

COM 0 Auxiliary RS232 Connector (17)

This D9 connector provides communication to external devices in the robot cell, such as a vision system, a programmable logic controller (PLC), barcode reader, and so forth.

TEACH PENDANT / Direct Teach Connector (18)

This D15F high density connector is the communication port for the teach pendant and the Direct Teach device.

When Direct Teach is installed, the Teach Pendant is connected to a connector on the cable of the Direct Teach device, and the Direct Teach cable is connected to the controller's D15F port.

Inputs/Outputs

For more information on reading and operating inputs and outputs, refer to the section,"Input and Output Programming,," in Chapter 5 and to the I/O circuit diagram in Appendix A.

Inputs

The Controller-AC for MK3 has 16 inputs. The logical state of the inputs is as follows:

	Input ON	Input OFF
Hardware	Voltage applied	Voltage Not applied
ACL	1	0
Display	LED on	LED off

Outputs

The Controller-AC for MK3 has 16 relay outputs. Four have both Normally Open and Normally Closed contacts, while the other 12 have only Normally Open contacts. The logical state of the inputs is as follows:

	Output ON	Output OFF
N.O. Contact	Closed	Open
N.C. Contact	Open	Closed
ACL	1	0
Display	LED on	LED off

This page intentionally left blank.

CHAPTER 2

Safety

1. Do not operate the **Controller-AC** until you have studied this manual thoroughly.
2. Do not install or operate the **Controller-AC** under any of the following conditions:
   - Where a safety ground connection does not exist.
   - Where the ambient conditions are below or exceed the specified limits.
   - Where subject to high vibrations or shocks.
   - Where exposed to direct sunlight.
   - Where subject to chemical, oil or water splashes.
   - Where corrosive or flammable gas is present.
3. Before you connect the controller to the AC power outlet, make sure the outlet supplies 220V–230V +10% / -15%.
4. To guarantee safety, be sure the controller and the robot power have a ***ground connection***. (Robot must have a separate wire connected to the ground screw on the controller; controller must have a ground connection.)
5. After turning off the controller, wait at least ***one minute*** before you turn it on again.
6. Turn off the controller's POWER switch:
   - Before you do any work within the working envelope of a connected robot or automated device.
   - Before you connect any inputs or outputs.
7. Turn off the controller's POWER switch ***and disconnect the power cable from the AC power outlet*** before you open or service the controller.

   The power supply units contain dangerously high voltages; the power cable must be disconnected to remove possible shock hazard.

   After disconnecting the controller from the power supply, ***wait three minutes*** before opening or servicing the unit.
8. Never drive a current of more than 1.5A through the relay contact of the digital outputs.

9. 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 controller's red EMERGENCY button or STOP button.
   - Press the teach pendant's red EMERGENCY button.
   - Use the **ACL** command **[Ctrl]+A**.

CHAPTER 3

Installation

The installation instructions in this chapter refer to the diagrams of the controller's front and rear panels, shown in Figures 1-2 and 1-2.

PC/Terminal–Controller Installation

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

Allow adequate clearance around the controller, and be sure air passage to the rear fan is not blocked.

☞ *Do not connect the controller to the AC power supply until you are sure that the supply is in the range 220V–230V +10% / -15%.*

1. Make sure the controller power switch (1) is turned off.
2. Connect the AC power cable to the controller (21) and to the AC power supply outlet.
3. Install and configure your computer/terminal and monitor according to the manufacturer's instructions. Then turn off the computer's power switch.
4. Connect the RS232 cable (D9 connector) to the CONSOLE port (16) 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 be using a teach pendant (with or without Direct Teach handles), install 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. If you will be using a remote emergency switch, install it at this time. Follow the instructions for installing a remote emergency switch later in this chapter.
8. 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 turn on the controller's POWER switch.
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 a directory (ATS) on 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:

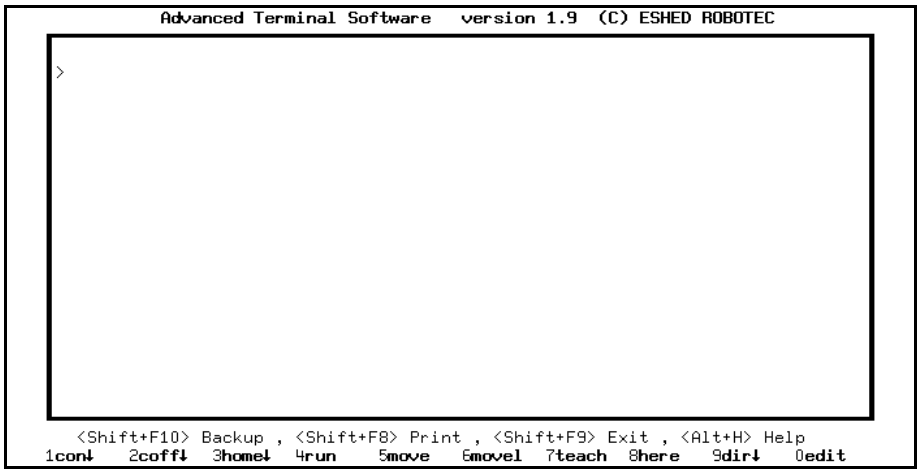
   ```
   term_acl  <Enter>
   ```

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

   ```
   term_acl /c2 <Enter>
   ```

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

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



   - Press <Enter>.
   - If still no prompt appears, 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 only if you are using a non-standard configuration—for example, with additional servo axes at axes 7 and 8, or without a robot. Otherwise, the configuration procedure is not needed.

7. After you have completed the controller configuration, you may connect other hardware to the controller.

Controller Configuration

If you are using a non-standard configuration, you will have to change the controller configuration.

The system displays the controller's default or currently defined values in square brackets, and the range of options in parenthesis. To accept a displayed value, simply press **<Enter>** , as indicated below.

The following example shows how to configure a controller in which a driver card for axes 7 and 8 is installed. (Normally the slot for axis driver 6 remains empty, but the software assumes a card is installed; hence the configuration is for 8 axes.)

To begin the configuration, type:

```
>config <Enter>
(239) SYSTEM CONFIGURATION
!!!WARNING ALL USER PROGRAMS WILL BE ERASED.
ARE YOU SURE ??? [YES/NO/DEFAULT] (DEFAULT)> <Enter>
JOB KILLING PHASE ..........>
ENTER NUMBER OF INPUTS [16] (0-16) > <Enter>
ENTER NUMBER OF OUTPUTS [16] (0-16) > <Enter>
ENTER NUMBER OF AXIS DRIVERS INSTALLED [5] (0-8)>8 <Enter>
ENTER NUMBER OF AUXILIARY RS232 PORTS [0] (0-8) > <Enter>
WHICH TYPE OF ROBOT (0-NONE,2-MK2,3-MK3) [3](0-3)> <Enter>
ENTER NUMBER OF SERVO LOOPS, GROUP A [5] (5-5)> <Enter>
ENTER NUMBER OF FIRST AXIS OF GROUP B [6] (6-8) 7<Enter>
ENTER NUMBER OF SERVO LOOPS, GROUP B [0] (0-3)> 1<Enter>
ENTER NUMBER OF USER PROGRAMS [400] > <Enter>
ENTER NUMBER OF USER PROGRAM LINES [5000] > <Enter>
ENTER NUMBER OF USER VARIABLES [3000] > <Enter>
ENTER NUMBER OF USER POINTS , GROUP A [5000] > <Enter>
ENTER NUMBER OF USER POINTS , GROUP B [0] > 1000<Enter>
ENTER NUMBER OF USER POINTS , GROUP C [0] > 1000<Enter>
ENTER NUMBER OF USER COMMENTS [3000] > <Enter>
Performing configuration, please wait 10 seconds

Available workspace 366012 (Bytes) 100%
Assigned workspace  280511 (Bytes)  76%
```

```
Unused workspace     85501 (Bytes)  23%
O.K.
>
```

For more information on the controller configuration, refer to the command CONFIG in the *ACL Reference Guide*.

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.

Install the robot according to the procedure detailed in your robot's user manual.

To connect the robot to the controller, do the following.

- Connect the robot ground wire to the Ground connection screw (23).
- Connect the robot to the ROBOT RESOLVERS D50 connector (27).
- Connect the robot to the ROBOT circular connector (24).

☞ *The robot must be connected to the controller Ground connection screw.*

Peripheral Devices and Equipment

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

Teach Pendant

1. 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.

2. Connect the teach pendant to the Teach Pendant DB-15 female port on the front of the controller. (Refer to Figure 1-1 #18). Tighten the connector screws.

Inputs/Outputs

The I/O Centronics connectors (31 and 32) at the rear of the controller is used to connect external I/O devices .

Additional RS232 Channels

If an auxiliary RS232 communication card 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 (36) 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.

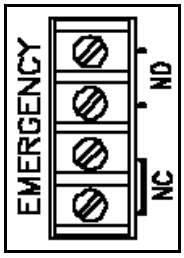
If you wish to install an auxiliary RS232 communication card in your controller, follow the instructions described in the section, "Maintenance Procedures," in Chapter 6.

Remote Emergency Switch

A remote emergency switch (such as a mushroom button) can be connected to the controller.

Refer to the Emergency Circuits diagram in Appendix A.

Remote emergency wiring consists of two independent lines (30):



- A serial line for normally closed switches.
- A parallel line suitable for normally open switches.

You can use either one of these lines, or both for maximum security.

To connect to the N.C. Terminal:

- Loosen the screws and remove the wire which shorts the two poles of the terminal at the rear of the controller.
- Connect the wires of the emergency switch to the N.C. terminal, and tighten the screws.

To connect to the N.O. Terminal:

- Connect the wires of the emergency switch to the N.O. terminal, and tighten the screws.

Remote Emergency outputs also exist between pins 1 and 19 on the I/O Block 1 and Block 2 connectors on the controller rear panel (31 and 32).
In this case N.O. switches should be used.

Refer to the description of the Emergency button in Chapter 1 and to the wiring diagram of the Emergency System in Appendix A.

Peripheral Axes

To install peripheral devices as axes 7 and 8, contact your agent for instructions.

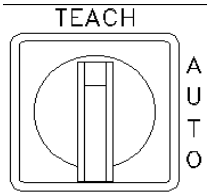
CHAPTER 4

Operation

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

Auto and Teach Modes

*When you operate the system from the **keyboard**, the teach pendant must be switched to **Auto.*** If the system has been operated previously from the TP, you must enter the **ACL** command AUTO in order to return operation control to the keyboard.



*When you operate the system from the **teach pendant**, the teach pendant must be switched to **Teach**.*

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 the next chapter.

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.

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!
>
'J' FOR JOINT, 'X' FOR XYZ, 'Z' FOR TOOL
>
>
JOINT MODE   or   XYZ MODE   or   TOOL 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 Sytem

Activate the system and load the **ATS** software, as described in the section, "Power On," in Chapter 3.

Homing the Robot

☞ The home routine must be executed only when the entire robot-controller system is put into operation for the first time.

Homing is not otherwise required, except when:

- The robot has undergone a maintenance procedure.
- A "Resolver Error" occurred.
- The robot was forcibly moved when its brakes were locked and power was off.

Although the controller's software will automatically detect when the robot needs to be homed, it is recommended, as a precaution, that you always home the robot in these three cases.

☞ *Before you begin the homing procedure, make sure it can be started from the robot's current position.*

During the standard homing procedure, each joint is moved toward one of its limits, where an optoswitch closes a dedicated controller input. After reaching the limit, the controller makes the necessary offset and initialization.

Robot finds its home position by executing the following sequence:

- Search for axis 2 upper limit switch.
- Search for axis 4 upper limit switch, then align axis 4 and axis 3.
- Search for axis 3 lower limit switch.
- Search for roll (axis 5) limit switch.
- Robot moves shoulder, upper arm and forearm to vertical position.
- Base turns clockwise to search for axis 1 limit switch.
- Robot returns to position 0, which is a factory-set reference position.
- Position 0 is recorded in ACL.

Robot Homing Routine

This section describes the standard robot homing procedure.

If limitations in the robot's working envelope prevent you from using the standard homing routine, you can home axes individually, as described in the following section.

PC

Before you begin the homing, make sure axis 3 is not near one of its limits. If it is, use the TP or keyboard manual mode to bring it more within its range.

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 postion, 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
```

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]
```

Homing Individual and Peripheral Axes

PC To home one robot axis only:

Type: **home *n* <Enter>**

where *n* is the number of a robot axis.

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

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

Abort Homing

Besides pressing the controller's EMERGENCY or STOP button, you can use the following methods to stop the homing routine.

If you abort the homing routine, it must be executed again in its entirety.

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

Type: **A <Enter>**

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

TP To stop the homing while the operation is in progress,:

Press: **[ABORT]**

Coordinate Systems

The **Performer-MK3** robot can be operated and programmed in three different coordinate systems: Joint, Cartesian (XYZ) and Tool coordinates. For complete information on working with these coordinate systems, refer to Chapter 2 in the ***ACL** Reference Guide*.

TP

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

To select one of the three coordinate systems:

Press: **[JOINTS|XYZ|TOOL]**

```
group:A ax:– –       XYZ
```

Press again: **[JOINTS|XYZ|TOOL]**

```
group:A ax:– –     JOINTS
```

Press again: **[JOINTS|XYZ|TOOL]**

```
group:A ax:– –     TOOL
```

The display reflects the currently active coordinate system.

Depending on the active coordinate system:

- Manual movements of the axes are executed according to Joint, XYZ or Tool coordinates.
- The TP's RECORD POSITION command records positions by executing HERE and HERER (in Joint mode), HEREC and HERERC (in XYZ mode), and HEREC and HERERT (in Tool mode).

PC

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

Activate Manual mode (press <Alt>+M).

Then, to activate the Joint coordinate system:

Press: **J**

```
JOINT MODE
```

To activate the XYZ coordinate system:

Press: **X**

XYZ MODE

To activate the Tool coordinate system:

Press: **Z**

TOOL MODE

Axis Control Groups

The teach pendant's Group Select key allows you to switch control of the teach pendant to peripheral axes (Group B), or to an independent axis (Group C), or to an electric servo gripper (Group G). By default, the selected group is the robot (Group A).

TP

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

To select the axis control group:

Press: **[GROUP SELECT]**

group:B ax:7 JOINTS

Continue pressing this key for group G (gripper), group B, group C, and again for group A, and so on. If a group is not configured, it will not be displayed.

If the robot is equipped with a gripper which has an AC servo motor and resolver, the controller will treat the gripper as a distinct axis control group, G.

If group C is selected, use the TP numerical keys to enter the number of the independent axis. Then press **[ENTER]**.

When alternating among control groups, group A will remain in the coordinate system (Joint/XYZ/Tool) 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.

Enabling 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:

`cona`	Enables control of robot axes (Group A).
`coffb`	Disables control of peripheral axes (Group B).
`con 9`	Enables control of axis 9 (Group C).

Moving the Axes

Joint, XYZ and Tool Coordinate Systems

When the coordinate system is set to the **XYZ or Tool** mode:

- Movement commands to the X, Y or Z axis result in a linear motion of the **tool center point (TCP)** along the respective axis, while maintaining a constant orientation of the tool.
- Movement commands to the pitch or roll axis will change the orientation of the tool, while maintaining a constant TCP position.

When the coordinate system is set to the **Tool** mode, the X, Y and Z axes are defined as follows. Refer to the figure below.

- The Z-axis is the axis which intersects the flange at its center point, and is perpendicular to it.
- The X-axis is horizontal and perpendicular to the Z-axis.
- The Y-axis is vertical and perpendicular to both the Z and X axes.

When the coordinate system is set to the **Joint** mode, movement commands cause the robot to move one joint. (Peripheral axes always move according to Joint coordinates.)

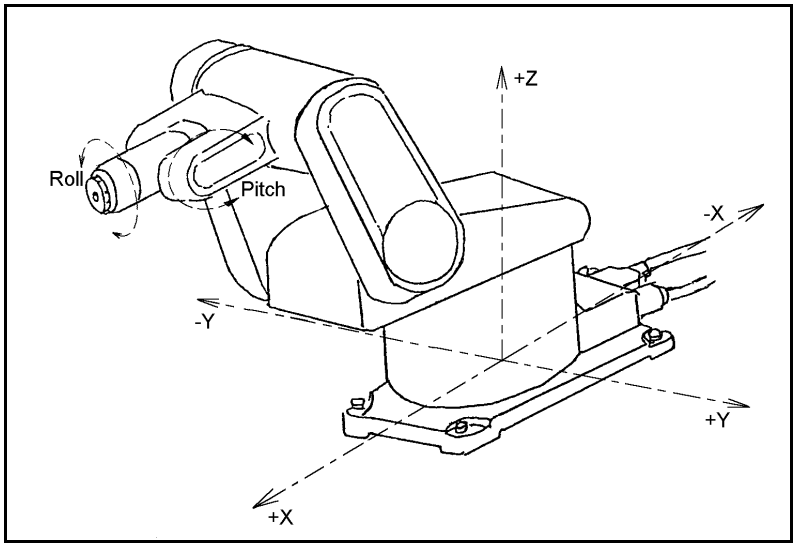


Figure 4-1: XYZ Coordinate System

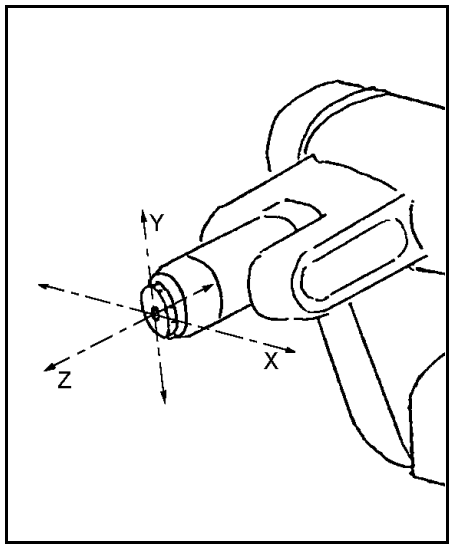


Figure 4-2: Tool Coordinate System

Moving the Axes

TP

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

When in XYZ and Tool mode, the controller recognizes the Cartesian functions of the teach pendant keys. When in Tool mode, XYZ coordinates refer to the Tool coordinates.

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

Press: **[–]**

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

Press: **[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 then Manual mode must be actived (press <Alt>+M).

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)
	6, Y	Moves axis 6; closes/opens electrical gripper

In XYZ and Tool modes 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

Activating the Gripper

Before using the gripper, make sure the value of parameter 274 is set to the number of the output to which the gripper is connected.

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.

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 30% of maximum linear speed 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.

Three types of position names are possible:

- Vector names (such as PVEC[50] and PVEC[10]) of up to five characters and an index. A position vector—an array of positions—can be attached to the teach pendant by means of the **ACL** command ATTACH. The vector positions can then be accessed from the teach pendant by means of their index number.

  For more efficient programming, define position vectors and record positions named according to the vector indices.

- 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.

- 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.

If you accidently 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.

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.

Move the robot to another position.

Press: **[RECORD POSITION]**

Here 13

The teach pendant automatically increments the position number (definition name) during successive record commands.

Press: **[ENTER]**

[RECORD POSITION] records positions according to the currently active coordinate system. Press the key repeatedly to scroll through the options for recording relative positions.

When in Joint Mode:

Press: **[RECORD POSITION]**

Here

Press again: **[RECORD POSITION]**

HereR

When in XYZ or Tool Mode:

Press: **[RECORD POSITION]**

HereC

Press again: **[RECORD POSITION]**

HereRC	if XYZ Mode.
HereRT	if Tool Mode.

[RECORD POSITION] is also used to record positions in a vector. The vector must first be attached (ATTACH) to the TP.

Refer to the ***ACL** Reference Guide* for complete descriptions of the position recording commands.

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:

`defpb B24 <Enter>` Defines position B24 for Group B.

`defpc C3 10 <Enter>` 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 will overwrite the existing coordinates with new coordinates.

The **ACL** commands HERER, HERERC and HERET allow you to record a position as relative to another position.

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.

ACL has a number of commands for recording position coordinates. Refer to the ***ACL** Reference Guide* for more details.

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 User's Manual*.

Joint Movement

TP

To make a joint moment, Group A must be 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

To make a joint moment, Group A must be 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

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 User's Manual*.

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

CHAPTER 5

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, 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. Like the MOVE command, these commands can also receive the D suffix which ensures sequentiality of execution; 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 proecdures for halting a program during its execution.

Suspend the Program

To halt the running program, turn the Run/Hold switch on the controller front panel to **Hold**.

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

To resume execution of the program, turn the Run/Hold switch on the controller front panel to **Run**.

The program will restart from the point of interruption by executing the next program command line. The robot will continue moving from the point where it was halted.

Abort the Program

Besides pressing the controller's STOP button or EMERGENCY button, you can immediately abort running programs and stop all axis movement 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 hundreths of a second; for example, if *n* = 150, the delay is 1.5 seconds.

Edit program AAA. Insert delay commands after 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.

System Variables

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

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:

`set var1 = abs var2`	If the value of *var2* is -1, *var1* 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:

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

`set out[2]=1 <Enter>`	Turn on output 2 to activate device.
`set out[2]=0 <Enter>`	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]** **[EXECUTE]**

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   OUT
            **********************
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 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 apply voltage to any of inputs 1 through 15, the output with the same index (1–15) will turn on. When you apply voltage to 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 editing and backup operations can cause program identity numbers to change, use the DIR command 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

The **Controller-AC for MK3** 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, the **Controller-AC** 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 Interrupt

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.

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.

This page intentionally left blank.

CHAPTER 6

Maintenance

Normal controller operation does not require any maintenance procedures.

Only authorized personnel may service the controller.

☞ *Before you open the controller or perform any service procedure, do the following:*

- *Turn off the controller's power switch.*
- *Disconnect the power cable from the AC power source.*
- *Wait 3 minutes (due to high residual voltage inside the controller).*

☞ The controller contains electrostratic sensitive CMOS devices.
Be sure to observe ESD (electrostratic discharge) regulations for CMOS devices.

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, "Maintenance Procedures," later in this chapter. Refer also to your robot's user 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.

If the procedures described here do not help, contact your local agent.

1. *No communication between controller and PC*

<table>
<tr><td colspan="5">Make sure you loaded ATS with the proper /C switch. (Refer to installation instructions in Chapter 1.)</td></tr>
<tr><td colspan="5">Check the RS232 connection. Make sure the RS232 cable between the controller CONSOLE port and the computer COM port is properly connected.</td></tr>
<tr><td colspan="5">Switch the controller off, wait 1 minute, and switch on again, without exiting ATS.</td></tr>
<tr><td colspan="5">Watch “Ready” Light.</td></tr>
<tr><td colspan="2">“Ready” Light Blinking</td><td colspan="3">“Ready” Light Steady</td></tr>
<tr><td colspan="2">No messages, no prompts.</td><td colspan="2">Message appears:
<code>Default Setup Installed</code></td><td>Message appears:
<code>Driver not responding Axis 1</code></td></tr>
<tr><td colspan="2">Hardware failure.
Remove all DDC and DPU cards from controller, and again turn on controller.
Message appears:
<code>Driver Not Responding</code></td><td colspan="2">Was this the first time you ever turned on the controller?</td><td>Remove DDC-1 card and insert another DDC card in its slot. Exchange the DPU-1 card.</td></tr>
<tr><td>Yes</td><td>No</td><td>Yes</td><td>No</td><td rowspan="2"></td></tr>
<tr><td>CPU board is O.K.

Problem in one of the driver cards. Check each card one at a time by inserting them into DDC/DPU Slot 1. Replace the defective DDC or DPU card .</td><td>CPU board is faulty or memory is corrupted. Remove CPU battery for 15 seconds. Reinsert battery and check again. If CPU is still not working, replace the board.</td><td>This is a normal response the first time the controller is put into operation.</td><td>Check backup battery on CPU card.</td></tr>
</table>

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

The PC does not support Japanese.

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

3. *After controller is powered on, or after the CON command is issued, message appears:* `Driver not responding at axis n.`

- Make sure the DDC card for the specified axis is correctly inserted.
- Exchange the DDC card for the specified axis with a DDC card for another axis.
- If still not responding, change the DPU card for the specified axis.

4. *Message:* `Resolver error axis n`

Check the resolver connection for that axis:

- Turn off the controller, disconnect from power source, and wait three minutes. Disconnect the robot resolvers cable from the Robot Resolver connector on the controller's back panel. Measure the resistance of the resolver winding through the cable. Normal values are approximately 10–100Ω. Refer to Appendix C for the Robot Resolvers connector pin-out. If the resolver connections and resolver are O.K. but the message still appears, replace the DDC card for the axis.
- If problem persists, replace the Dual Digital Controller (DDC) card.
- If problem persists, replace the Resolver Interface (RIC) card.
- If problem persists, replace the Dual Power Unit (DPU) card.

5. *Message:* `Bad connection of robot cable or fan failure.`

- Make sure the robot power cable connector is well secured.
- Check the internal fan and the fan on the back panel. Replace fan, if necessary.

6. *Message:* `DPRAM Watch Dog error axis n`

A malfunction caused a communication interuption between the main CPU card and the DDC.

- Reset the controller: turn it off, wait one minute, and then turn it on again.

7. *Emergency lamp remains lit even after the Emergency button is released.*

- If a teach pendant is in use, make sure the EMERGENCY button on the teach pendant is released.
- Make sure the emergency terminals on the controller back panel have been correctly connected or jumpered.

Maintenance Procedures

☞ *Before you open the controller or perform any service procedure, do the following:*

- *Turn off the controller's power switch.*
- *Disconnect the power cable from the AC power source.*
- *Wait 3 minutes (due to high residual voltage inside the controller).*

☞ The controller contains electrostratic sensitive CMOS devices.
Be sure to observe ESD (electrostratic discharge) regulations for CMOS devices.

Opening the Front Cover

1. Turn off the controller, disconnect from power source, and wait three minutes.
2. Unscrew the 3 Phillips screws which secure the front panel cover. (Refer to Figure 1-1, #19.)
3. Carefully remove the cover and set it aside.

   Be careful not to damage the ribbon cables.

Replacing AC Power Fuses

These fuses are located at the rear of the controller. (Refer to Figure 1-2, #22.)

Turn off the controller, disconnect from power source, and wait three minutes.

To remove the AC power fuses, use a small screwdriver to release the fuse drawer cover. Pull out the fuse drawer, and remove and replace the faulty fuse(s). Reinsert the fuse drawer and press it until it snaps into place.

Removing/Replacing the CPU Battery

The battery is located on the CPU board. (Refer to Figure 6-1.) The battery should be replaced when it supplies less than 3.0V.

1. Turn off the controller, disconnect from power source, and wait three minutes.
2. Remove the front cover of the controller.
3. Using a small screwdriver, pry the battery loose. Pull out the battery.
4. To replace, slide the new battery into the slot until it snaps into place.

   Replace only with Renata CR2477N 3V battery, available from your product supplier.

Replacing or Adding a DDC Card

1. Turn off the controller, disconnect from power source, and wait three minutes.
2. Remove the front cover of the controller.
3. Carefully remove the 4 screws (2 on each bracket) from the metal retaining brackets across the PC boards in the front of the controller. *Do not let loose screws fall into the controller*. Remove the brackets.
4. (Only if replacing a card:) Carefully remove the DDC card from its slot.
5. Before inserting the new card, first check that none of the pins in the connector is bent. Then, make sure the card is directly above the back plane and DPU connectors. Firmly but gently press the card into the connectors

   The retaining brackets secure the PC boards within the controller during shipment. They do not need to be reattached, unless the controller is being shipped.

Replacing DPU Card

1. Turn off the controller, disconnect from power source, and wait three minutes.
2. Remove the fan housing from the rear of the controller. (Refer to Figure 1-2, #35.)
3. Completely loosen the two screws from the DPU card's bracket at the rear of the controller.
4. Using the DPU extracting tool, carefully remove the DPU card from its slot.
5. Before inserting the new card, first check that none of the pins in the connector is bent. Then, make sure the card is directly above the back plane and DDC connectors. Firmly but gently press the card into the connectors.
6. Fasten the card's bracket screws.
7. Press on the corresponding DDC card to make sure it is securely in place.
8. Reattach the fan housing.

Adding a DPU Card

1. Turn off the controller, disconnect from power source, and wait three minutes.
2. Remove the fan housing from the rear of the controller. (Refer to Figure 1-2, #35 and #37.)
3. Remove the two screws from the bracket over the blank slot (DPU slot 4) at the rear of the controller.
4. Before inserting the new card, first check that none of the pins in the connector is bent. Then, make sure the card is directly above the connector slot on the back plane. Firmly but gently press the card into the slot.

6. Fasten the card's bracket screws.
7. Press on the corresponding DDC card to make sure it is securely in place.
8. Reattach the fan housing.

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 external DB-37 connection may have either two or eight D25 connectors.

(Refer to Figure 1-2, #34; Figure 6-3; Figure B-12)

1. Turn off the controller, disconnect from power source, and remove the front cover of the controller.
2. Remove the 4 screws (2 on each bracket) from the metal retaining brackets across the PC boards in the front of the controller. Remove the brackets.
3. From the rear of the controller, disconnect any cables connected to the Resolver card. Loosen the two screws on the resolver card bracket, and remove the resolver card from the controller.
4. Remove the two screws from the plate which covers the empty Auxiliary RS232 port on the Resolver card, and detach the metal plate from the card.
5. Insert the female connector of the RS232 ribbon cable into the Auxiliary RS232 port, so that the connector faces out. Fasten the two hex nuts
6. Position the Resolver card in its slot. Thread the male connector of the RS232 ribbon cable through the hole in the back plane.
7. Carefully reinsert the Resolver card into its slot. Rasten the bracket screws. Reconnect any cables you disconnected previously.
8. From the front of the controller, attach the male connector ribbon cable to the Auxiliary RS232 card.
9. Before inserting the Auxiliary RS232 card, make sure none of the pins in the connector is bent. Then, make sure the card is directly above the connector slot on the back plane. Firmly but gently press the card into the slot.
10. Make the cable connections:
    - Connect the D37 connector from the multiport connector cable to the Auxiliary RS232 port on the controller.
    - Connect the D25 connectors on the multiport connector cable to the corresponding COM ports on the other controllers or computers.

11. 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)."
    - Use the **ACL** command CONFIG, as described in the ***ACL** Reference Manual*.
    - 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)."

Expanding RAM

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

(Refer to Figure 6-1 and to Figure B-3.)

1. Turn off the controller, disconnect from power source, and wait three minutes.
2. Remove the front cover of the controller.
3. Remove (if still in place) the metal retaining brackets across the PC boards in the front of the controller. *Do not let loose screws fall into the controller*.
4. Remove the CDC card with the attached CPU card.
   Be sure you observe ESD (electrostratic discharge) regulations for CMOS devices.
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.

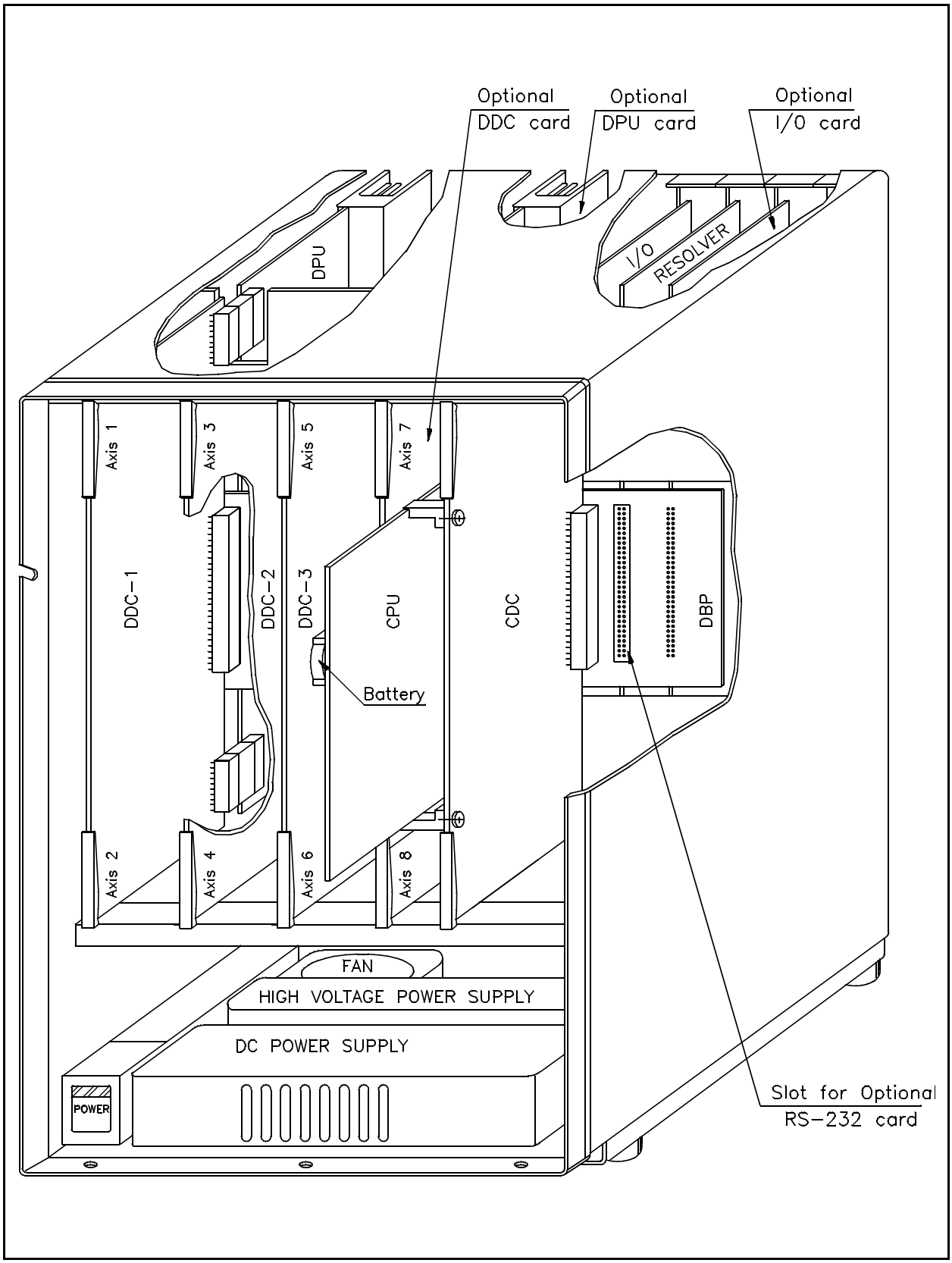


Figure 6-1: Controller Components - front

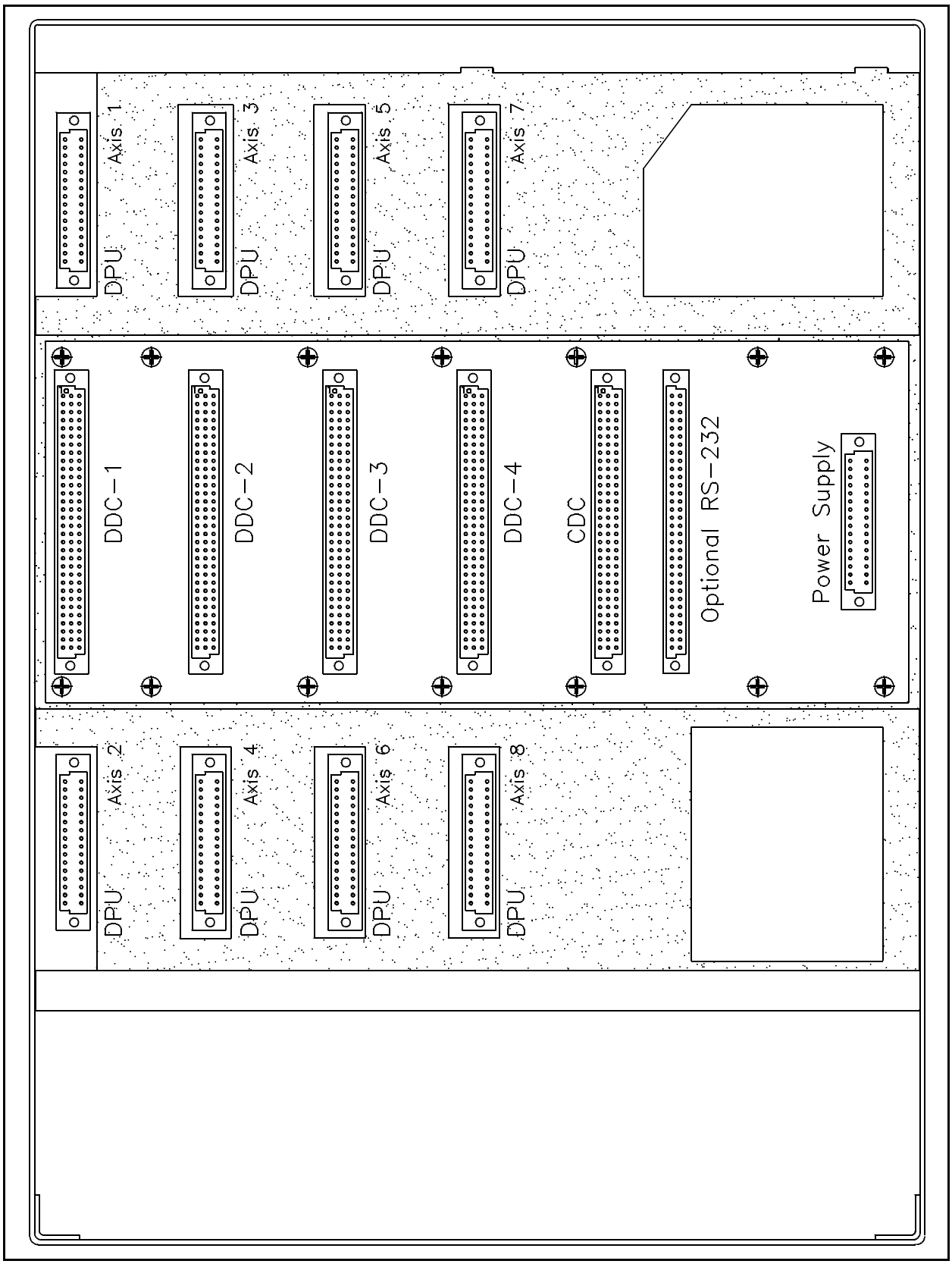


Figure 6-2: Connectors - front

APPENDIX **A**

Circuits

Input Circuit

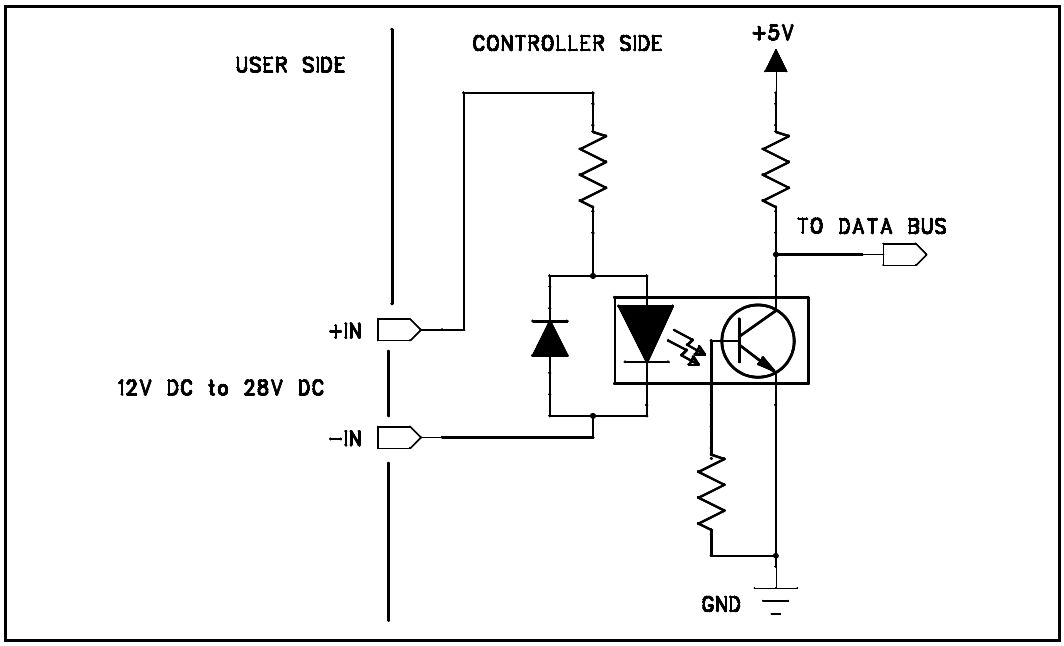


Figure A-1: Input Circuit

Emergency System Circuits

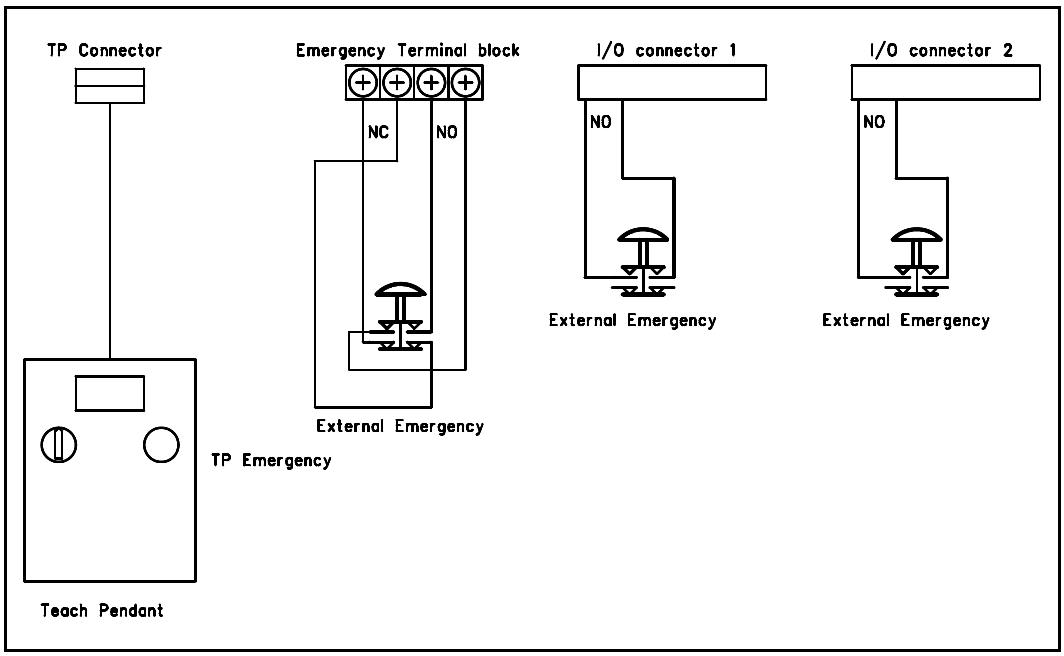


Figure A-2: Emergency System Circuit

APPENDIX **B**

PC Boards

This appendix shows the circuit and connector boards in the **Controller-AC for MK3**.

The **Controller-AC for MK3** contains two back-to-back buses: the Digital Back Plane (DBP) and the Power Back Plane (PBP). They are separated by a metal shield which also serves as their mechanical support. These back planes allow easy insertion of cards from both sides of the controller.

Each Dual Digital Controller (DDC) card mates with its companion Dual Power Unit (DPU) card by means of two direct connectors.

Items marked by numbers in some of the diagrams have accompanying descriptions.

Dual Digital Controller Card (DDC)

Each DDC card controls two AC servo, permanent magnet brushless motors.

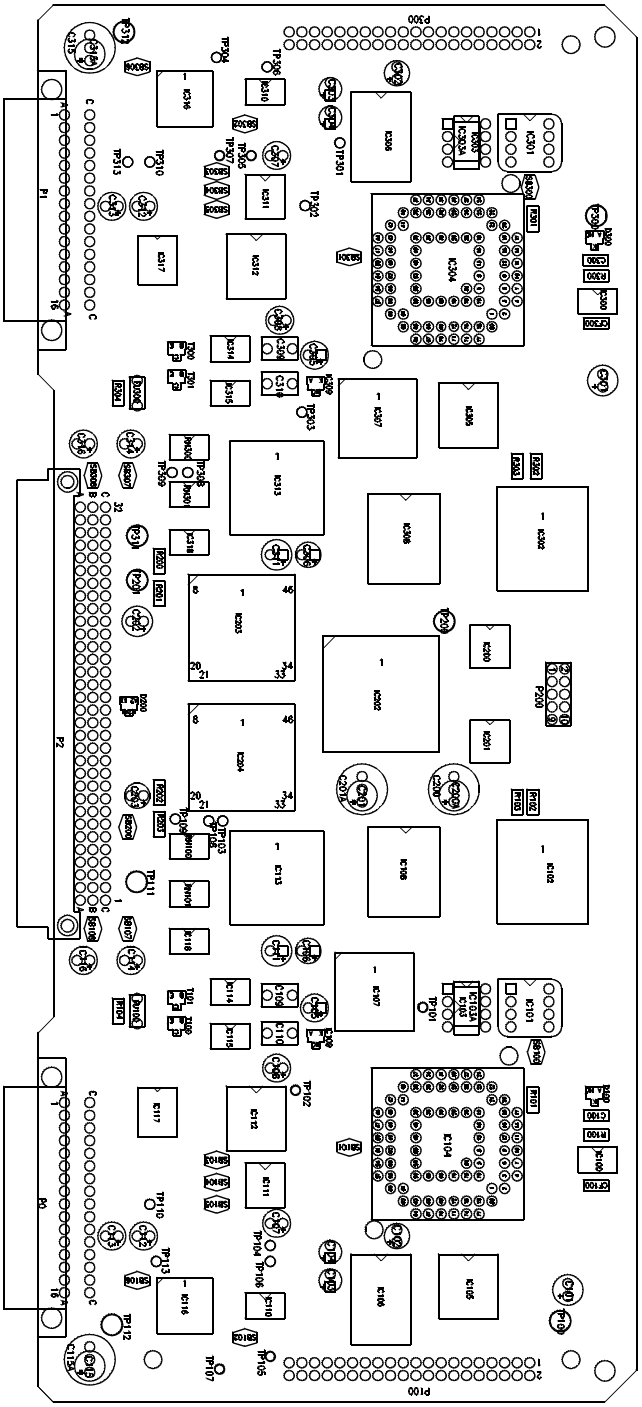


Figure B-1: PC-27150
DDC Card

Dual Power Unit Card (DPU)

Each DPU card accepts PWM controls from the DDC card and directly drives two AC servo, permanent magnet brushless motors.

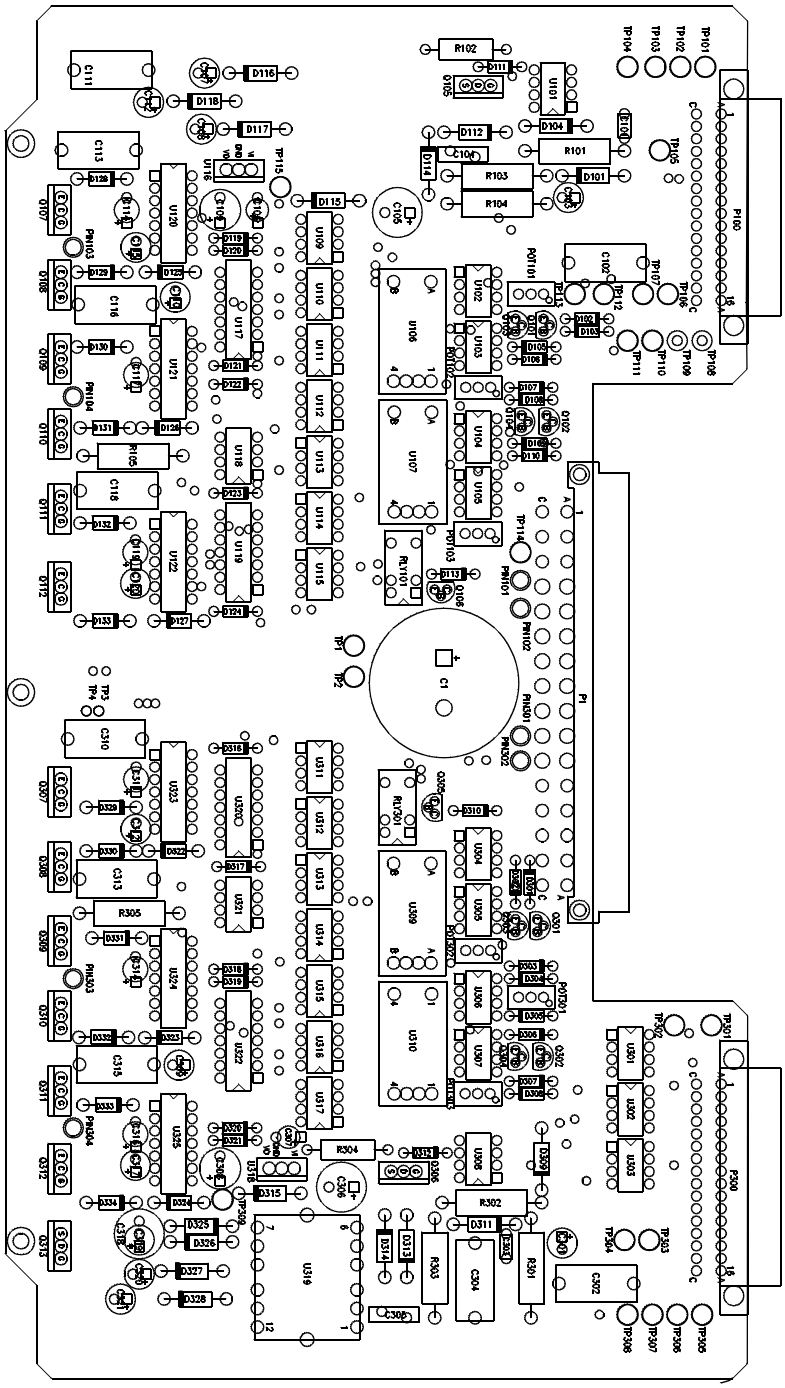


Figure B-2: PC-27100
DPU Card

Main CPU Card (CPU)

The CPU board is piggy-back connected to the Controller Digital card (CDC).

1 68020 CPU with a 68882 floating point unit

2 Four ICs: battery backed-up 512KB RAM

3 Four EPROMS: 512KB programmed with **ACL**.

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

4 PLD

5 A 3V lithium battery, for memory back-up.

6 Coprocessor

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

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

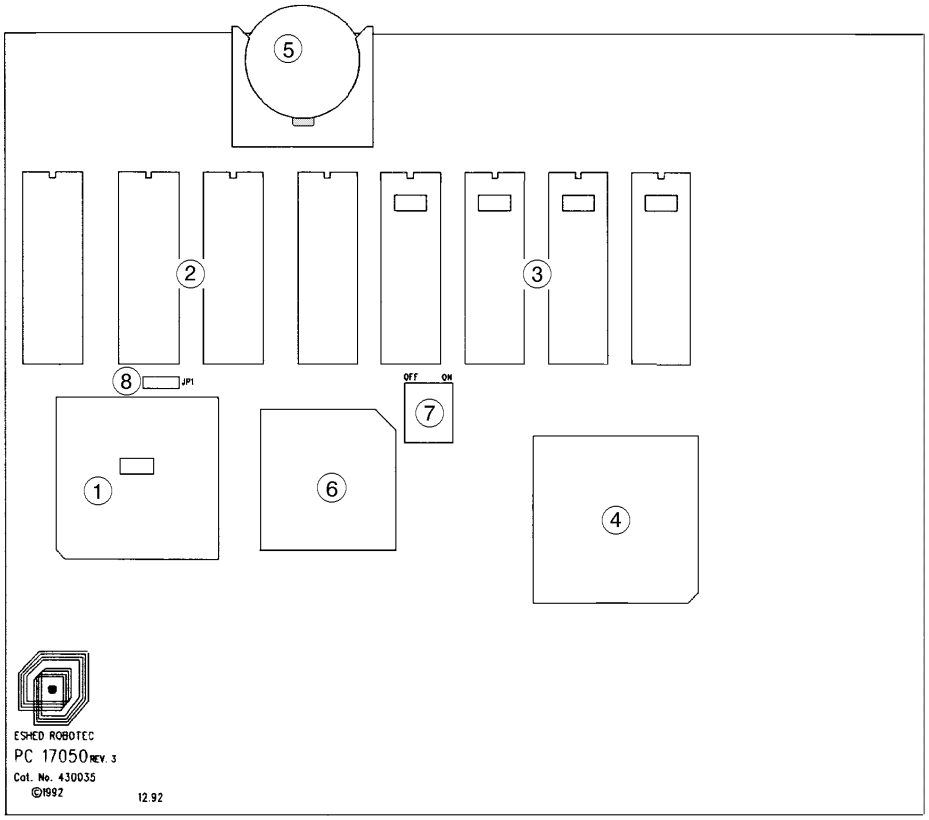


Figure B-3: PC-17050
CPU Card

Controller Digital Card (CDC)

This card contains axis inhibit logic, teach pendant interface, and console and auxiliary RS232 hardware. It also contains a connector slot for the main CPU card and hardware watchdog for the main CPU.

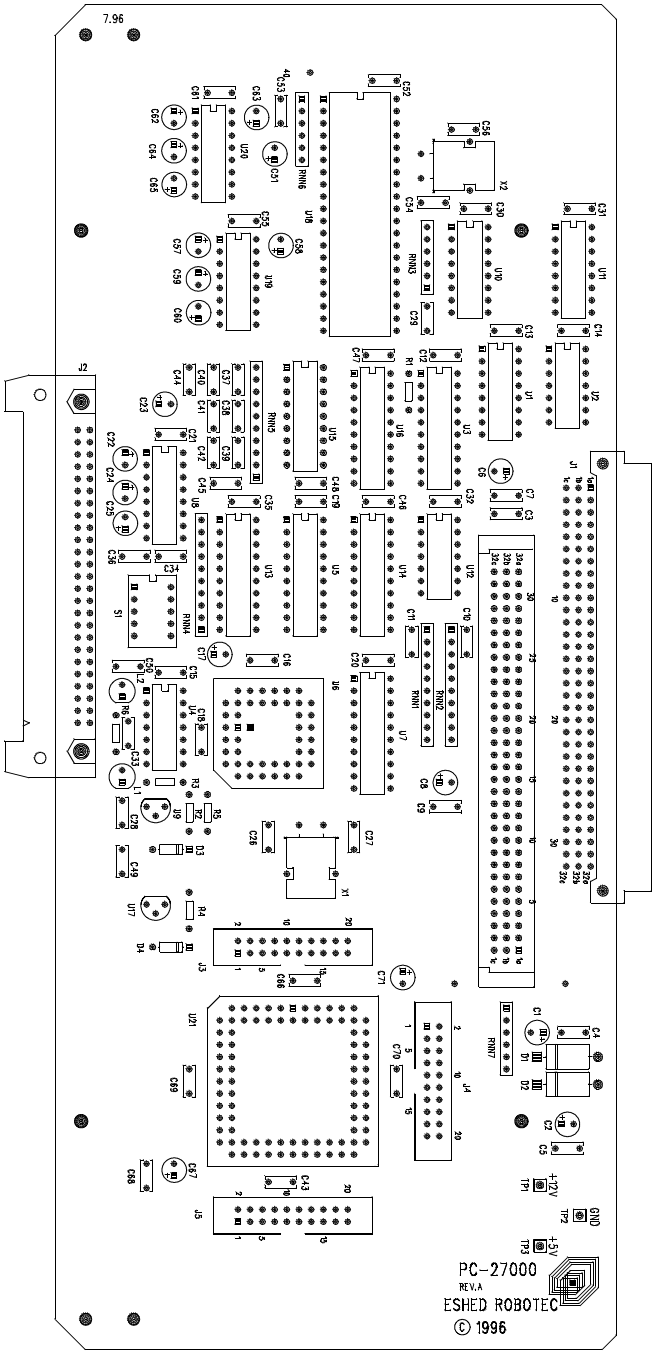


Figure B-4: PC-27000
CDC Card

Back Panel Interface Card (BPI)

This card contains the back panel connectors and some internal system connectors. It also contains all I/O electronic circuits for: 16 optically isolated inputs; 16 output relays; parallel port logic; emergency logic.

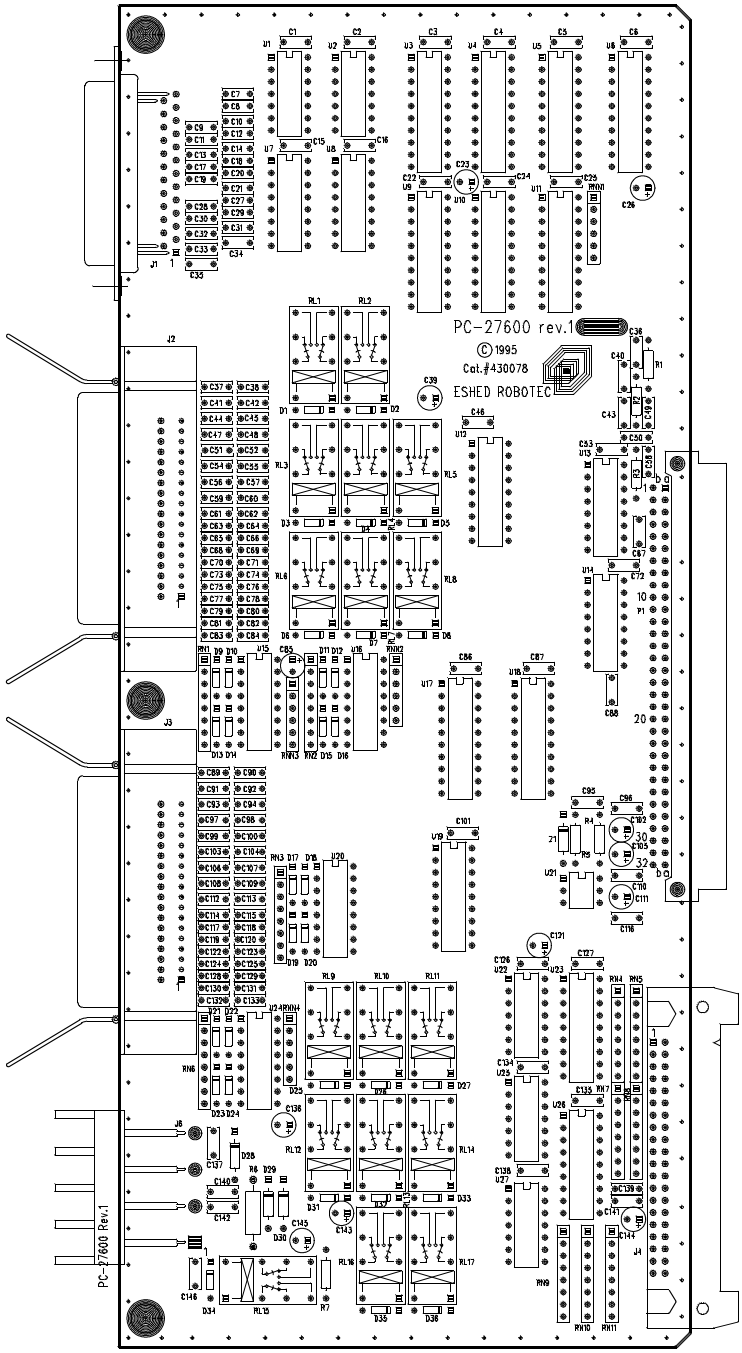


Figure B-5: PC-27600
BPI Card

Resolver Interface Card (RIC)

This card contains the resolver connectors and optically isolated inputs for the axis limits.

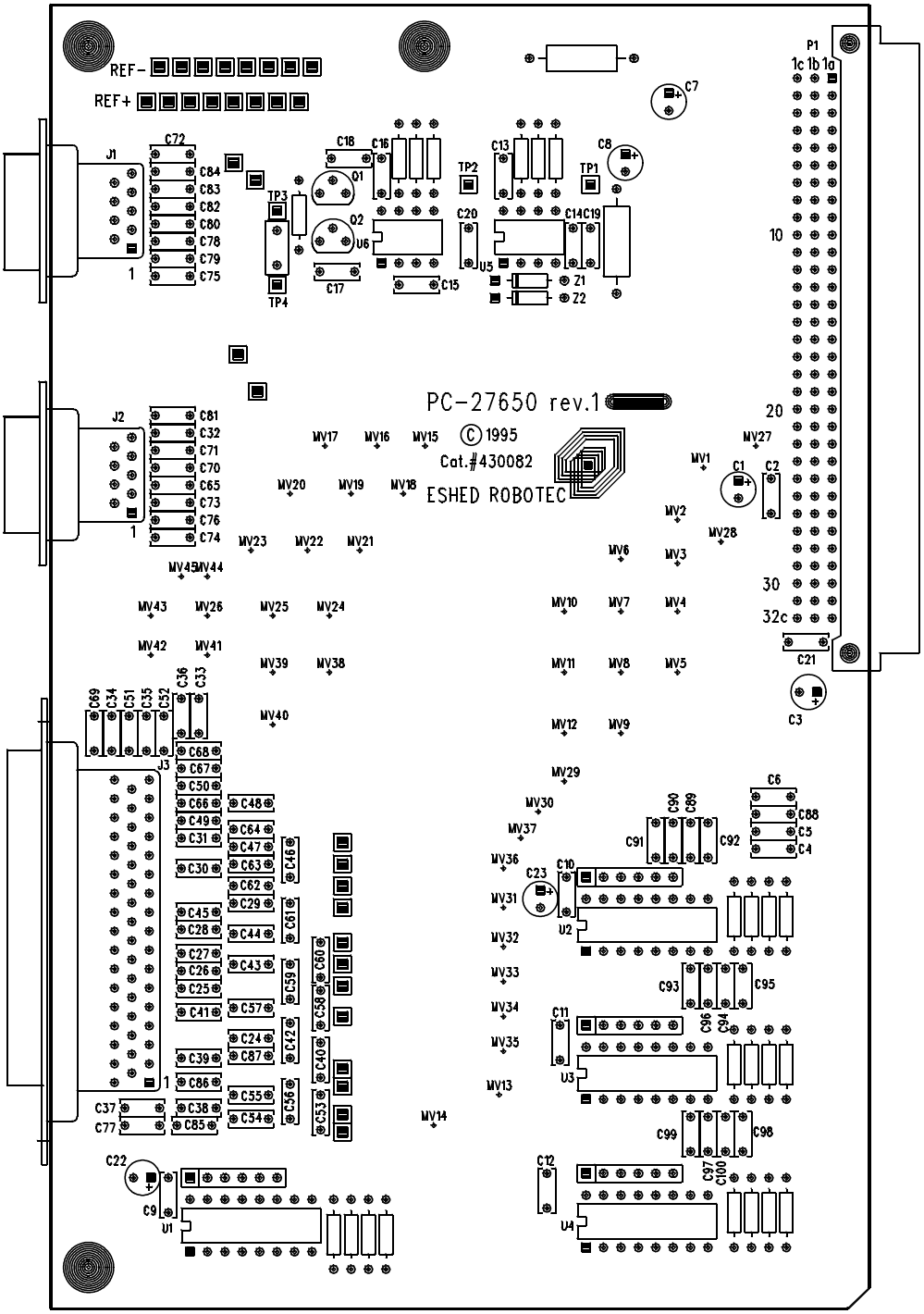


Figure B-6: PC-27650
RIC Card

Front Panel Interface Card (FPI)

This card contains the connectors for: front panel switches and lamps; front panel ports (Console, Com 0 and Teach Pendant); all LEDs on the front panel *except* I/O LEDs.

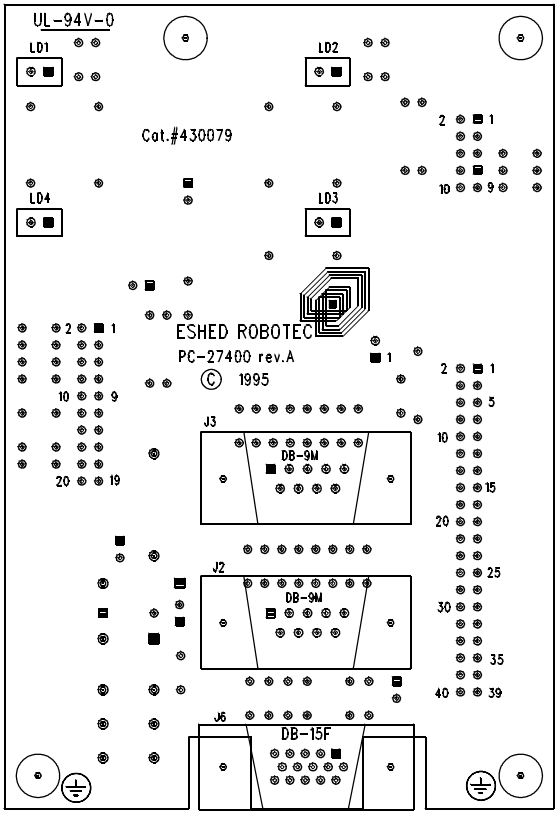


Figure B-7: PC-27400 FPI Card - front

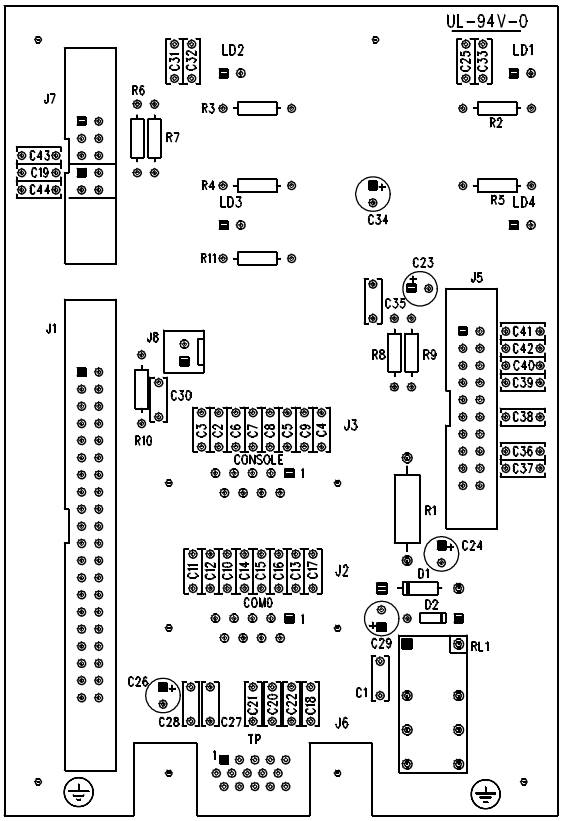


Figure B-8: PC-27400 FPI Card - rear

System Power Supply Card (SPS)

This card supplies high voltage power for the motors. It contains AC fail logic, over-voltage and under-voltage logic.

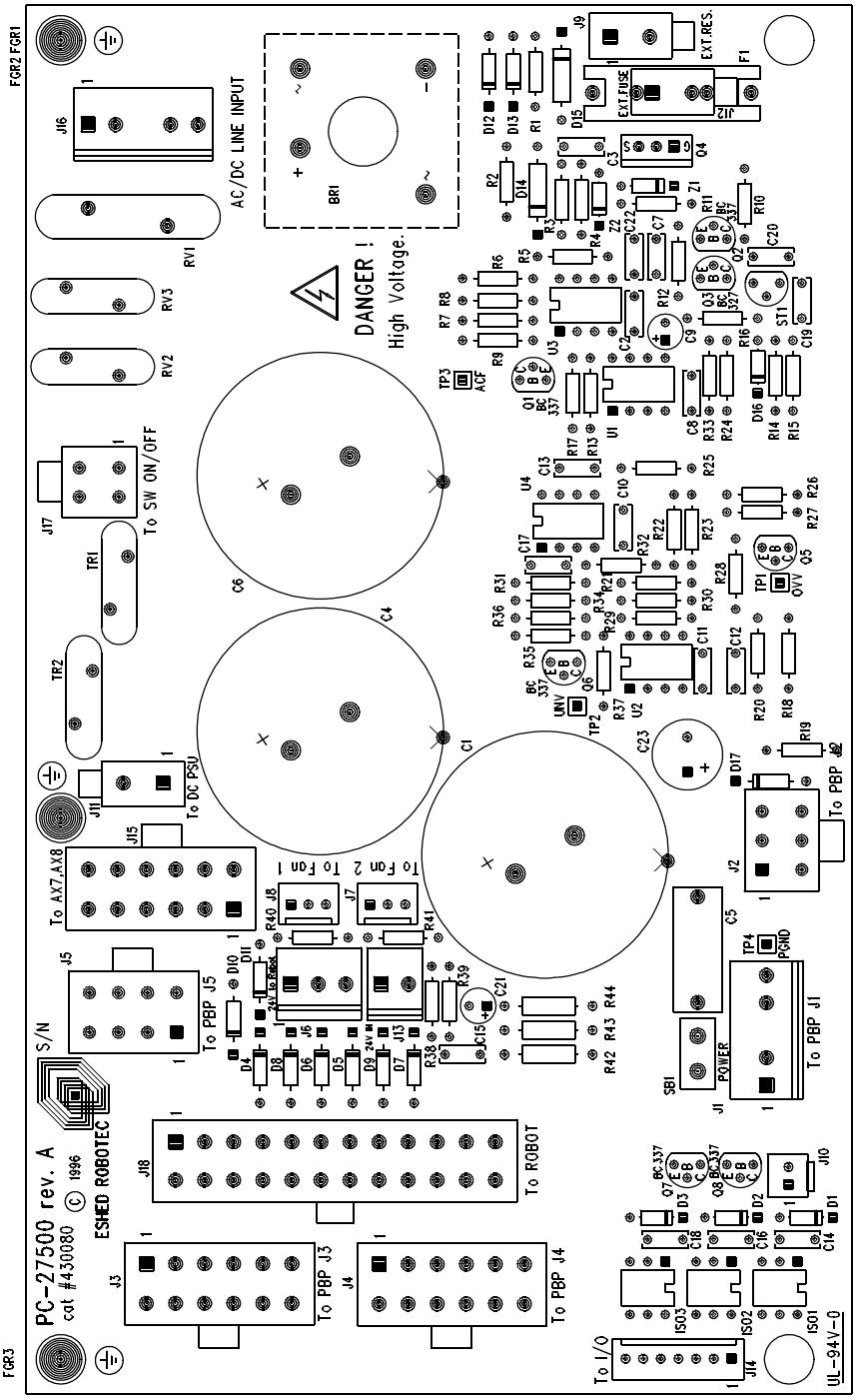


Figure B-9: PC-27500
SPS Card

Digital Back Plane (DBP)

A mainly passive connector back plane, this card contains: 4 driver bus slots; CDC card slot; auxiliary multiport RS232 slot.

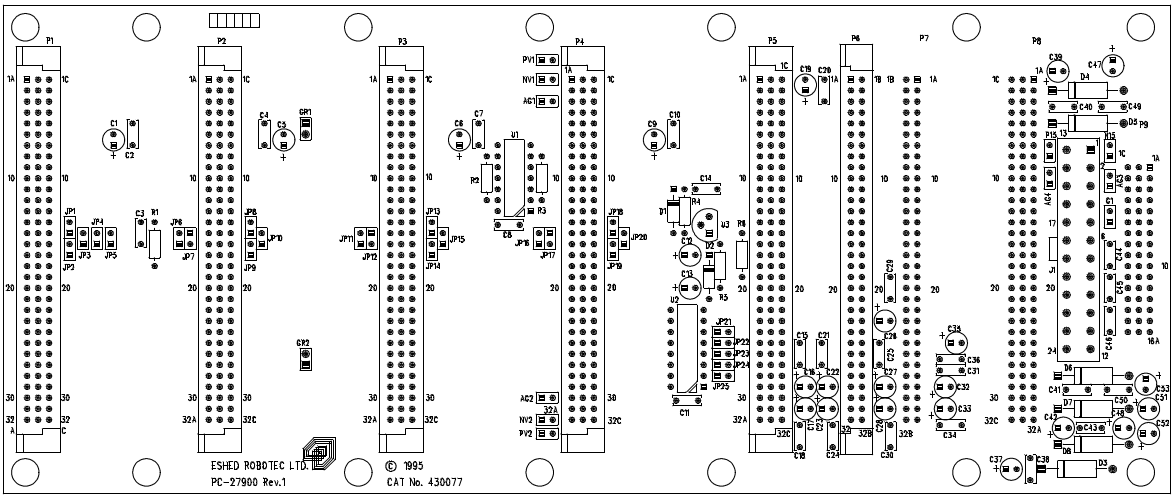


Figure B-10: PC-27900
DBP Card - front

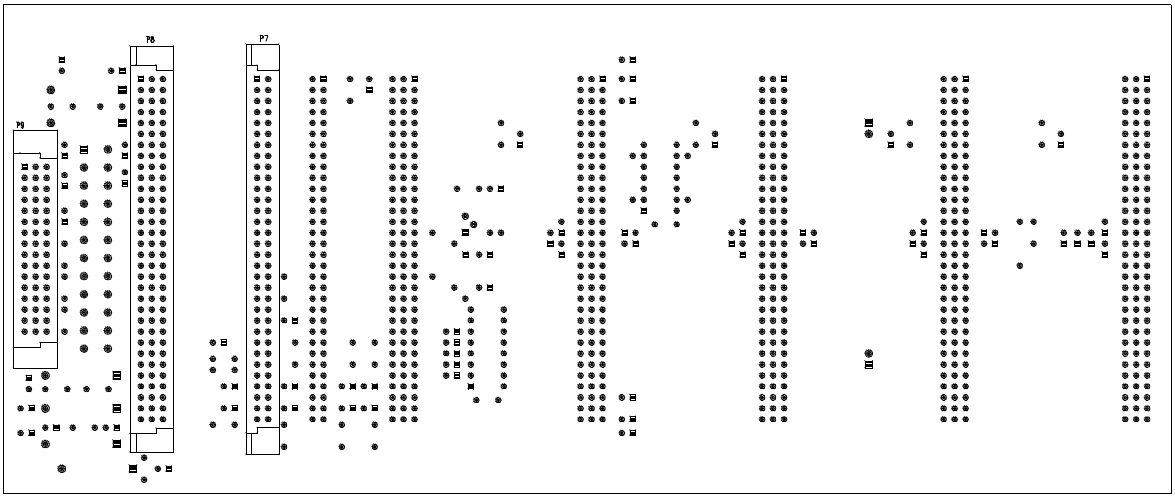


Figure B-11: PC-27900
DBP Card - rear

Power Back Plane (PBP)

This card connects to the high voltage power supply and distributes all high voltage signals. The DPU cards connect to the motor power connectors on the PBP card.

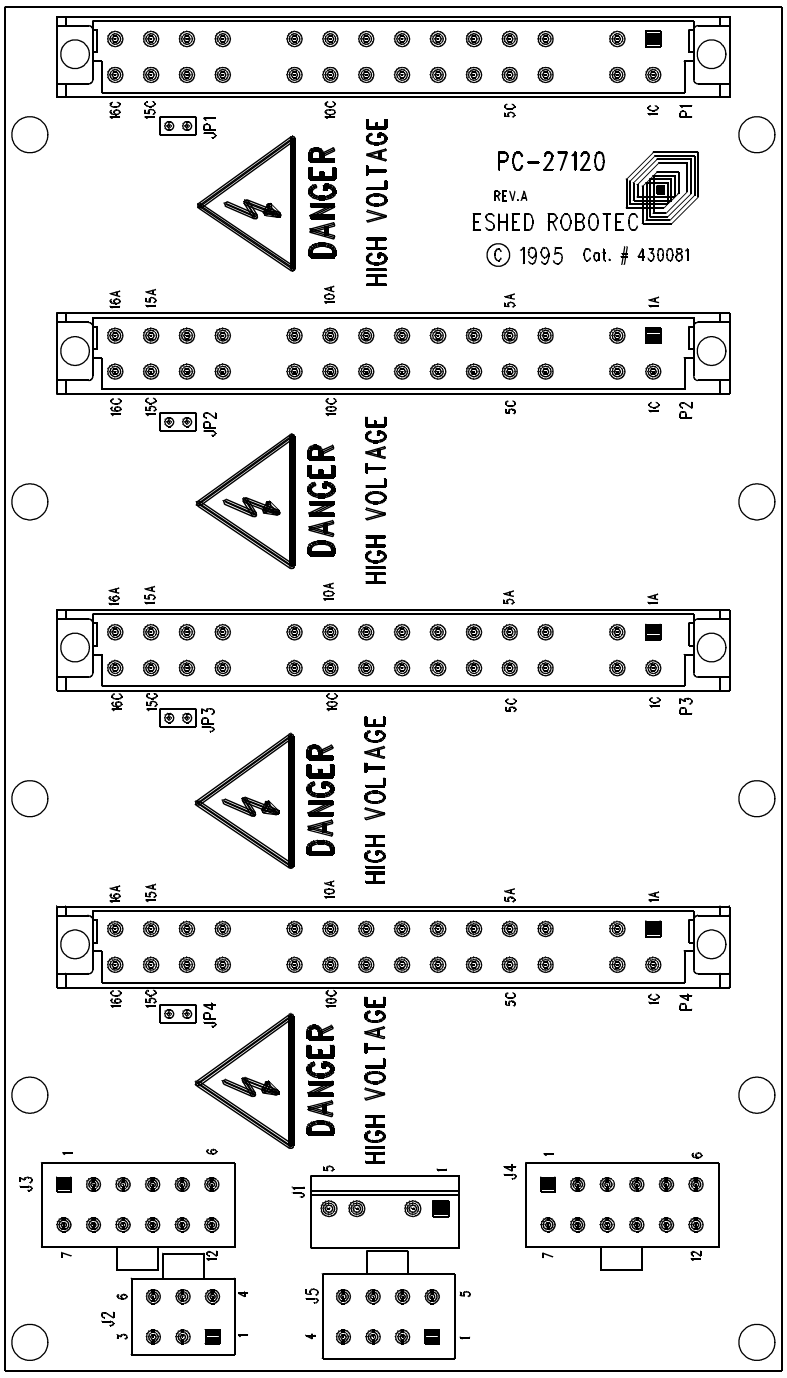


Figure B-12: PC-27120
Power Back Plane

Auxiliary RS232 Card

This is an optional 8-channel RS232 communication card.

1 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.

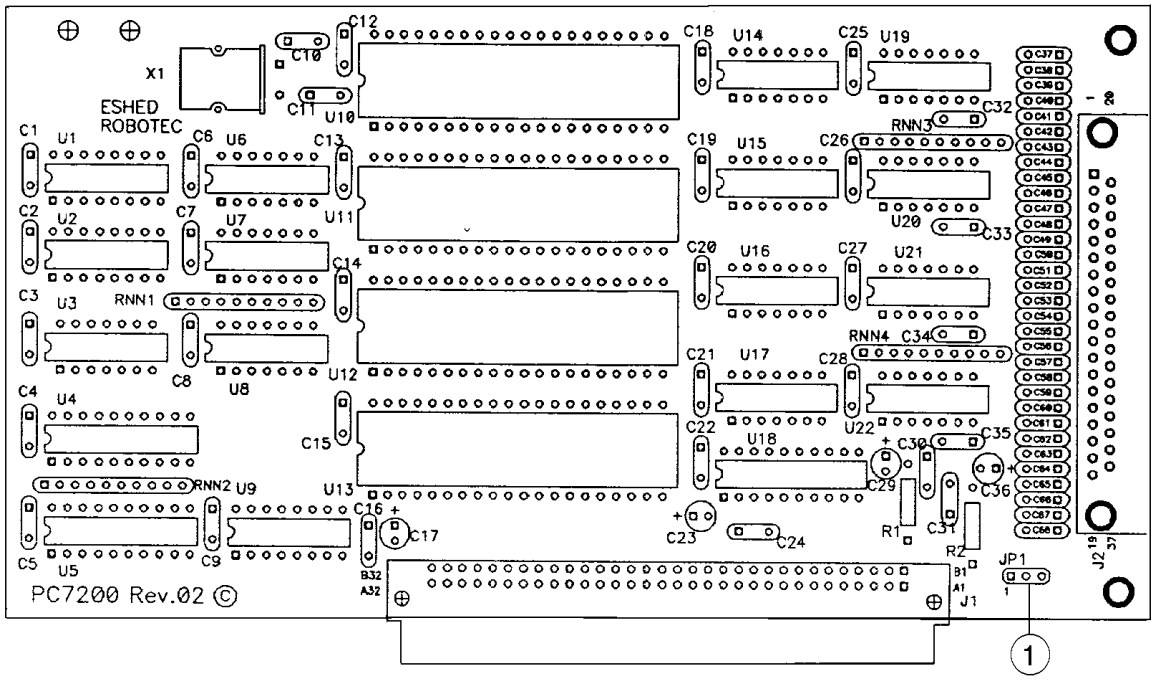


Figure B-13: Aux. RS232 Multiport Communication Card

APPENDIX **C**

Connector Pin-Outs

Teach Pendant

Connector type: DB-15 pin, high density, female

1	GND
2	+5V
3	RECORD
4	+12V
5	RX
6	GND
7	EMERGENCY
8	RECORD RETURN
9	RESERVED
10	DEAD MAN RETURN
11	PULL UP
12	NOT USED
13	RESERVED
14	DEAD MAN
15	TX

Console RS-232

Connector type: DB-9 pin, male

2	RX0
3	TX0
5	GND
7	RTS0
8	CTS0

Auxiliary RS-232

Connector type: DB- 9 pin, male

2	RX1
3	TX1
5	GND
7	RTS1
8	CTS1

I/O Block 1

Connector type: Amphenol-57, 36 pin, female

1	EMRG (NO)
19	EMRG (NO)
2	+IN1
20	– IN1
3	+IN2
21	– IN2
4	+IN3
22	– IN3
5	+IN4
23	– IN4
6	+IN5
24	– IN5
7	+IN6
25	– IN6
8	+IN7
26	– IN7
9	+IN8
27	– IN8
10	NO_O1
28	COM_O1
29	NC_O1
11	NO_O2
12	COM_O2
30	NC_O2
13	NO_O3
31	COM_O3
14	NO_O4
32	COM_O4
15	NO_O5
33	COM_O5
16	NO_O6
34	COM_O6
17	NO_O7
35	COM_O7
18	NO_O8
36	COM_O8

I/O Block 2

Connector type: Amphenol-57, 36 pin, female

1	EMRG (NO)
19	EMRG (NO)
2	+IN9
20	– IN9
3	+IN10
21	– IN10
4	+IN11
22	– IN11
5	+IN12
23	– IN12
6	+IN13
24	– IN13
7	+IN14
25	– IN14
8	+IN15
26	– IN15
9	+IN16
27	– IN16
10	NO_O9
28	COM_O9
29	NC_O9
11	NO_O10
12	COM_O10
30	NC_O10
13	NO_O11
31	COM_O11
14	NO_O12
32	COM_O12
15	NO_O13
33	OM_O13
16	NO_O14
34	COM_O14
17	NO_O15
35	COM_O15
18	NO_O16
36	COM_O16

Robot Motors

Connector type: Burndy UTG-Metalok Bantam, 28 pin
(insert arrrangement: size 20-28, female).

A	M1U
B	M1V
T	M1W
C	M2U
D	M2V
U	M2W
E	M3U
F	M3V
V	M3W
G	M4U
H	M4V
W	M4W
J	M5U
K	M5V
X	M5W
L	M6U
M	M6V
Y	M6W
Z	RESERVED
S	BR1
b	BR2
c	BR3
e	BR4
d	BR5
N	BR6
P	+24VBR
a	24VRET
R	ENABLE 1_6 POWER

ENABLE 1_6 POWER input should be shorted to 24VRET inside the robot.

Axis 7 / Axis 8 Motors

Connector type: Burndy UTG-Metalok Bantam, 8 pin
(insert arrrangement: size 12-8, female)

A	M7(8)U
B	M7(8)V
C	M7(8)W
D	RESERVED
G	BR7(8)
E	+24VBR
H	24VRET
F	ENABLE 7(8) POWER

ENABLE 7(8) POWER input should be shorted to 24VRET inside the accessory.

Robot Resolvers

Connector type: DB-50 pin, female

[	1	RV11
	2	RV12
[	18	RV13
[	19	RV14
[	34	RVR11
[	35	RVR12
[	3	RV21
[	4	RV22
[	20	RV23
[	21	RV24
[	36	RVR21
[	37	RVR22
[	5	RV31
[	6	RV32
[	22	RV33
[	23	RV34
[	38	RVR31
[	39	RVR32
[	7	RV41
[	8	RV42
[	24	RV43
[	25	RV44
[	40	RVR41
[	41	RVR42
[	9	RV51
[	10	RV52
[	26	RV53
[	27	RV54
[	42	RVR51
[	43	RVR52
[	11	RV61
[	12	RV62
[	28	RV63
[	29	RV64
[	44	RVR61
[	45	RVR62
	13	RESERVED
	46	RESERVED
[	14	LSW11
[	15	LSW12
[	30	LSW21
[	31	LSW22

[= Twisted pair

[	47	LSW31
	48	LSW32
[	16	LSW41
	17	LSW42
[	32	LSW51
	33	LSW52
[	49	LSW61
	50	LSW62

[= Twisted pair

Axis 7/ Axis 8 Resolvers

Connector type: DB-9 pin, female

	2	RESERVED
[	7	RV7(8)1
	3	RV7(8)2
[	8	RV7(8)3
	4	RV7(8)4
[	9	RVR7(8)1
	5	RVR7(8)2
[	6	LIM7(8)1
	1	LIM7(8)2

[= Twisted pair

APPENDIX **D**

Direct Teach

Direct Teach (DT) is an optional feature of the Performer-AC controller. It allows you to grasp and lead the Performer-MK3 robot arm to the different positions required by an application, and to record the positions.

Direct Teach has two elements:

- The **Direct Teach software** is programmed onto the controller's EPROMs.
- The two **Direct Teach handles** are attached to the robot arm and used for grasping and moving the arm.

  The first handle is attached to the flange at the end of the robot arm.

  The second handle is attached to the robot elbow and contains a dead man switch, which is pressed (activated) as soon as you grasp this handle. Direct teaching cannot be performed unless this switch is pressed.

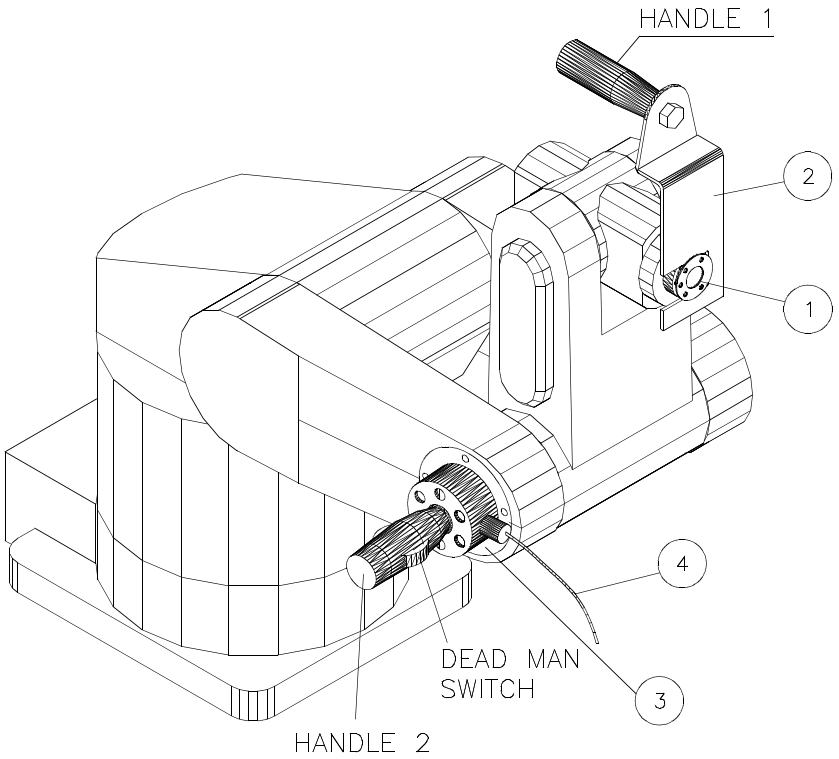


Figure D-1: Direct Teach Handles

To verify that the Direct Teach option is available, power on the controller and watch the screen display. You should see the message:

```
Direct Teach Option : Installed
```

New robots may be stiff and require more force during direct teaching. Robot movement will become easier and smoother after a few hours of running time.

Using DT to Move the Robot

To conduct a direct teach session, do the following.

1. Enable servo control: from the keyboard or from the teach pendant, enter the command **CONA**.
2. ***Move the teach pendant selector switch to the Teach position, and wait at least 2 seconds.***
3. Grasp the direct teach handles and press the dead man switch.
4. Move the robot (and record positions).

Using the Handles

As long as the dead man switch is pressed, you can lead the robot to the desired position.

If you release the dead man switch, the robot will stop and no longer move. You must leave the handles free for at least 2 seconds before you resume the direct teach session.

Be careful not to exert any force on the handles before you have pressed the dead man switch. The direct teach software determines the current payload while the dead man switch is released. Any pressure on the handles before the dead man switch is presssed will be included in the calculated payload value, and will result in extra effort being required to move the robot.

If too much force is applied to the handles before the dead man switch is activated, an error will be detected. You must press the controller's Error Reset button in order to clear the error.

When performing direct teach, always apply pressure gradually in the desired direction. Note that there is a short delay between the time pressure is first applied and the start of robot motion.

If you have moved the robot beyond its working envelope, use the teach pendant or keyboard manual mode (and not DT) to bring the robot back within its limits.

Modes of Movement

Three modes are available:

- Free Axes: all axes are free and can be moved to any position.
- Fixed Pitch: the pitch angle remains fixed the current orientation. All other axes can be moved freely.
- Fixed Pitch and Z: the pitch angle remains fixed at at the current orientation and the Z coordinate remains fixed at the current coordinate. All other axes can be moved freely.

To set a different mode, press the **Select Axis** key on the teach pendant. Keep pressing to scroll through the options: **Free Axes**, **Fixed Pitch**, **Fixed Pitch & Z**.

Using DT to Record Positions

To record positions, do the following:

1. Make sure servo control is enabled:

   ```
   CONA
   ```

2. Attach a position vector to the teach pendant by means of the ACL command ATTACH; for example:

   ```
   DIMP PALT[100]
   ATTACH PALT
   ```

3. Make sure the teach pendant is switched to the Teach mode setting.

4. Press the **Record Position** on the teach pendant. (Press once to record an absolute position; press twice to record a relative position.)

   The teach pendant display will depend on the active coordinate system (Joint, XYZ or Tool); for example:

   ```
   Here PALT _
   ```

5. Enter a position number, ***but do not press*** **Enter**; for example:

   ```
   Here PALT 19
   ```

6. Grasp the Direct Teach handles and move the robot arm to the desired location of position 19.

7. When the robot is at the position, press **Enter** on the TP.

8. The teach pendant will display:

   ```
   Here PALT 19
   Here PALT 20 _
   ```

This indicates that the first position was recorded, and the controller will record the next position as soon as the **Record** switch (or **Enter** key) is pressed again.

A short beep will sound when a position is successfully recorded.

A long beep will sound when an error occurs; for example, bad position number, position out of range, position not in XYZ space.

To enter another command from the teach pendant when the record position (Here...) prompt is displayed, simply press a new command key. The record prompt is erased, and the new command is displayed.