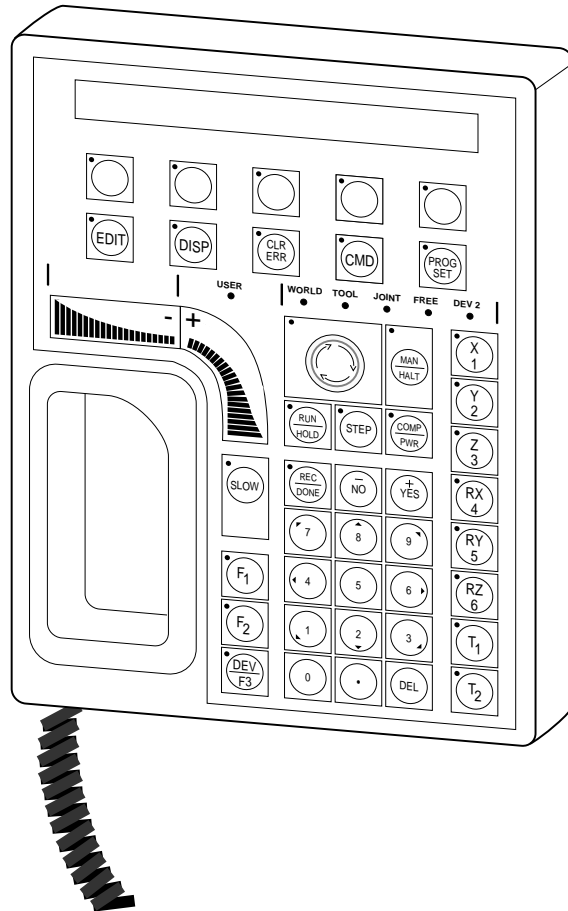
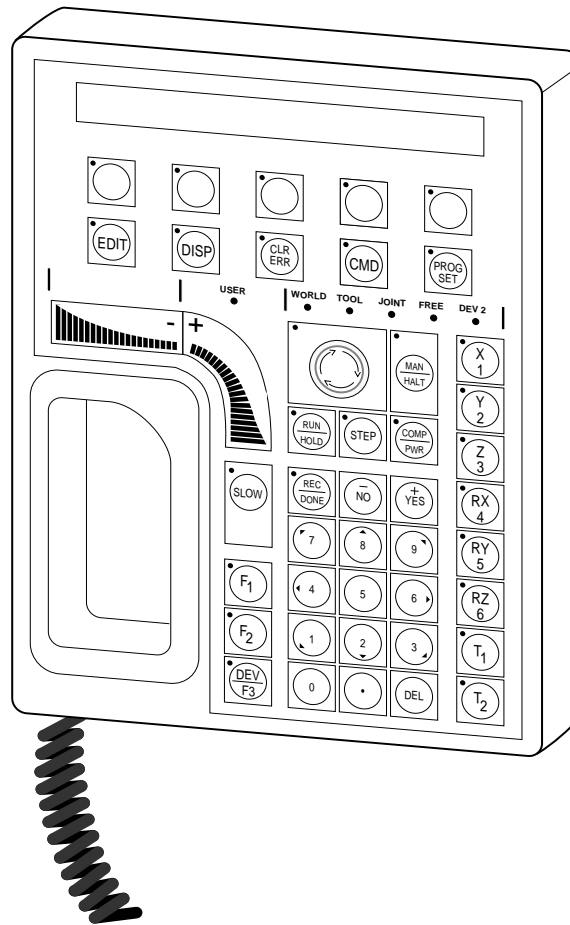

Manual Control Pendant

User's Guide



Manual Control Pendant

User's Guide



Part Number 00101-00010 Rev A
June, 1991



150 Rose Orchard Way • San Jose, CA 95134 • USA • Phone (408) 432-0888 • Telex 171942 • Fax (408) 432-8707
Otto-Hahn-Strasse 23 • 4600 Dortmund 50 • Germany • Phone (49) 231/75 89 40 • Fax (49) 231/75 89 450
7, rue Victor Hugo • 92310 Sèvres • France • Phone 33 (1) 45 34 32 32 • Fax 33 (1) 45 34 76 16

The information contained herein is the property of Adept Technology, Inc. and shall not be reproduced in whole or in part without prior written approval of Adept Technology, Inc. The information herein is subject to change without notice and should not be construed as a commitment by Adept Technology, Inc. This manual is periodically reviewed and revised.

Adept Technology, Inc. assumes no responsibility for any errors or omissions in this document. Critical evaluation of this manual by the user is welcomed. Your comments assist us in preparation of future documentation. A form is provided at the back of the book for submitting your comments.

Copyright © 1991 by Adept Technology, Inc. All rights reserved.

The Adept logo is a registered trademark of Adept Technology, Inc.

Adept, Adept MC, AdeptOne, AdeptThree, AdeptVision, V, and V⁺ are trademarks of Adept Technology, Inc.

Printed in the United States of America

Table of Contents

Chapter 1.	Introduction	1
1.1	This Manual and Related Publications	1
1.2	Compatibility	2
1.3	Notes, Cautions, and Warnings	2
1.4	Safety	3
	Reading and Training for System Users	3
	System Safeguards	4
	Computer Controlled Robots	4
	Manually Controlled Robots	5
	Other Computer Controlled Devices	5
1.5	Customer Service Assistance	6
	Service Calls	6
	Training Information	6
	Application Information	6
	International Customer Assistance	6
Chapter 2.	Manual Control Pendant Basics	7
2.1	Programmer's Pendant vs. Operator's Pendant	7
	Connecting the MCP	10
2.2	MCP Layout	11
	Soft Buttons	12
	Function Buttons	13
	Data Entry Buttons	14
	The REC/DONE Button	14
	The DEL Button	14
	Mode Control and Manual Control Buttons	15
	Speed Bars and Slow Button	15
2.3	Emergency Stop From the MCP	15
2.4	Background Mode	15
Chapter 3.	The MCP Predefined Functions	17
3.1	Introduction	17
3.2	Predefined Function Buttons	17
	Edit Function	18
	Real	18
	Loc	19
	Display Function	20
	Joint Values	20
	World Location	20

Status	21
Binary I/O	21
Last Error	21
The Clear Error Function	22
The CMD Function	23
Auto Start	23
Calib	23
Store All	24
CMD1 and CMD2	24
Prog Set Function	25
New	25
Step	25
Cycle	25
Speed	26
Start	26
Chapter 4. Moving a Robot or Motion Device With the MCP	27
4.1 Introduction	27
Mode Control Buttons	27
Emergency Stop Button	28
COMP/PWR Button	28
MAN/HALT Button	28
Manual Control Buttons	28
Speed Bars	29
Slow Button	29
4.2 Robot States	30
World State	30
Tool State	32
Joint State	35
Free State	38
4.3 Selecting Different Robots	40
Chapter 5. Programming the Pendant	41
5.1 Writing to the Pendant Display	41
The Pendant Display	41
ATTACHing to the Pendant	41
Using WRITE With the Pendant	42
5.2 Detecting User Input	43
Using READ With the Pendant	43
Detecting Pendant Button Presses	43
Keyboard Mode	44
Toggle Mode	44
Level Mode	45
Monitoring the MCP Speed Bar	46
Reading the State of the MCP	47

5.3	Controlling the Pendant	48
	Control Codes for the LCD Panel	48
	The Pendant LEDs	50
	Making Pendant Buttons Repeat Buttons	50
5.4	Auto-Starting Programs with the MCP	52
	WAIT.START	52
5.5	Programming Example: MCP Menu	53
Index	57

List of Figures

Figure 1-1	Impacts and Trapping Points	3
Figure 1-2	Arm Power/High Power & Program Running Lights	4
Figure 2-1	Holding the Programmer's MCP	7
Figure 2-2	Holding the Operator's MCP	8
Figure 2-3	Cradling the Operator's MCP	9
Figure 2-4	MCP Connection	10
Figure 2-5	MCP Layout	11
Figure 2-6	Soft Buttons	12
Figure 2-7	MCP Function Buttons	13
Figure 2-8	Data Entry Keys	14
Figure 3-1	MCP Predefined Function Buttons	17
Figure 3-2	EDIT Function Button	18
Figure 3-3	DISPLAY Function Button	20
Figure 3-4	CLEAR ERROR Function Button	22
Figure 3-5	Command (CMD) Function Button	23
Figure 3-6	Program Set Function Button	25
Figure 4-1	Mode Control Buttons	27
Figure 4-2	Speed Bar	29
Figure 4-3	SLOW Button	29
Figure 4-4	WORLD State (SCARA)	30
Figure 4-5	WORLD State (Cartesian)	31
Figure 4-6	TOOL State	33
Figure 4-7	TOOL State (Six-Axis Robot)	34

Figure 4-8	JOINT State (SCARA)	35
Figure 4-9	JOINT State (Cartesian)	36
Figure 4-10	JOINT State (Six-Axis Robot)	37
Figure 4-11	FREE State (Four-Axis SCARA)	39
Figure 5-1	MCP Button Map	47
Figure 5-2	Pendant LCD Display	49

List of Tables

Table 5-1	Pendant Control Codes	51
-----------	-----------------------------	----

Chapter 1

Introduction

1.1 This Manual and Related Publications

This manual covers the operation of the Manual Control Pendant (MCP). The MCP is used in many different applications, and with many different pieces of equipment. In some cases, the application using the MCP will have functions particular to that application. These functions are documented along with the application.

The MCP has the following primary functions:

- Serve as the primary point of control for initiating and monitoring operations.
- Guide the robot or motion device while teaching locations.
- Support application programs.

Chapter 2 of this manual covers the basics of using the MCP. Chapter 3 covers using the predefined functions of the MCP. These functions allow you to control an Adept system using only the MCP. Chapter 4 covers moving a robot or motion device with the MCP. Chapter 5 covers programming the MCP for use within your own programs.

You should have the following publications handy while learning to use the MCP:

1. The user's guide for any robot or motion device you will use in conjunction with the MCP.
2. The controller user's guide.
3. The user's guides for any applications that employ the MCP (AIM PCB for example).
4. *V and V⁺ Reference Guide* (if you are programming the MCP).

1.2 Compatibility

This manual is for use with manual control pendants operating in conjunction with V/V+ version 10.1 or higher.

1.3 Notes, Cautions, and Warnings

There are three levels of special notation used in this equipment manual. In descending order of importance, they are:



WARNING: If the actions indicated in a WARNING are not complied with, injury or major equipment damage could result. A WARNING will typically describe the potential hazard, its possible effect, and the measures that must be taken to reduce the hazard.



CAUTION: If the action specified in the CAUTION is not complied with, damage to your equipment could result.

NOTE: A NOTE provides supplementary information, emphasizes a point or procedure, or gives a tip for easier operation.

1.4 Safety

Reading and Training for System Users

Adept robot systems include computer controlled mechanisms that are capable of moving at high speeds and exerting considerable force. Like all robot systems¹ and industrial equipment, they must be treated with respect by the system user.

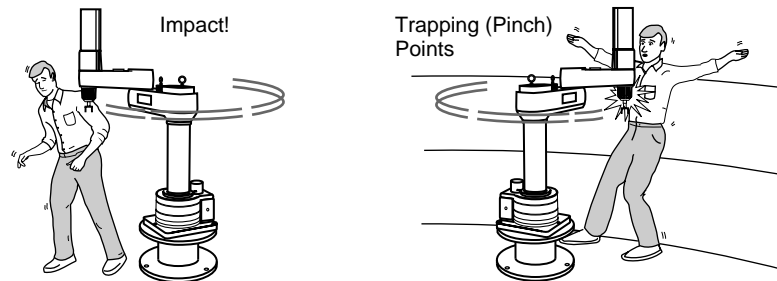


Figure 1-1. Impacts and Trapping Points

We recommend you read the *American National Standard for Industrial Robot Systems—Safety Requirements*, published by the Robotic Industries Association, in conjunction with the American National Standards Institute. The publication, ANSI/RIA R15.06-1986, contains guidelines for robot system installation, safeguarding, maintenance, testing, startup, and operator training. The document is available from the American National Standards Institute, 1430 Broadway, New York, NY 10018.

¹. Or any other motion device controlled by an Adept system.

System Safeguards

Safeguards should be an integral part of robot workcell design, installation, operator training, and operating procedures. Adept robot systems have various communication features to aid you in constructing system safeguards. These include remote emergency stop circuitry, and digital input and output lines.

Computer Controlled Robots

Adept robots are computer controlled, and the program that is running the robot may cause it to move at times or along paths you may not anticipate. When the amber ARM POWER/HIGH POWER² light and the blue PROGRAM RUNNING light on the front of the controller are illuminated, do not enter the workcell because the robot may move unexpectedly.

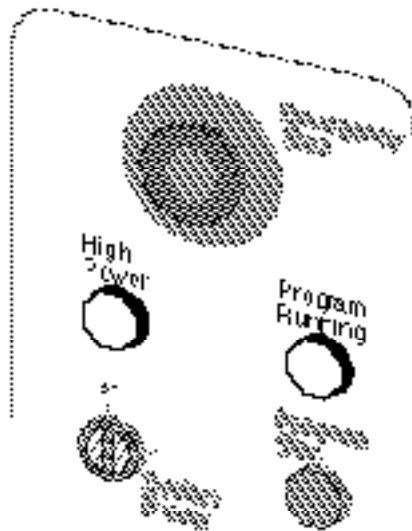


Figure 1-2. Arm Power/High Power & Program Running Lights

². Depending on the controller you have, the term HIGH POWER may be used in place of ARM POWER. The terms are synonymous.

Manually Controlled Robots

Adept robots can also be manually controlled when the amber ARM POWER/HIGH POWER light on the front of the controller is illuminated. When this light is lit, robot motion can be initiated from the terminal or the manual control pendant. If you enter the workcell when this light is illuminated, press the MAN/HALT button on the manual control pendant. This will prevent anyone else from initiating unexpected robot motions from the terminal keyboard.

Other Computer Controlled Devices

In addition, these systems can be programmed to control equipment or devices other than the robot. As with the robot, the program controlling these devices may cause them to operate at times not anticipated by personnel. It is critical that safeguards be in place to prevent personnel from entering the workcell when the blue PROGRAM RUNNING light on the front of the controller is illuminated.



WARNING: Entering the robot workcell when either the amber ARM POWER/HIGH POWER or the blue PROGRAM RUNNING light is illuminated can result in severe injury.

Adept Technology recommends the use of additional safety features such as light curtains, safety gates, or safety floor mats to prevent entry to the workcell while ARM POWER/HIGH POWER is enabled. These devices may be connected using the robot's remote emergency stop circuitry (see the controller user's guide).

1.5 Customer Service Assistance

Service Calls

Adept Technology maintains a fully staffed Customer Service Center at its headquarters in San Jose, CA. Two dedicated phone lines are available for **service calls only**:

(800) 232-3378 from outside California

(800) 232-3379 from within California

When calling Customer Service, please have available the serial number of the controller. If your system includes an Adept robot, have the serial number of the robot handy. The serial numbers can be determined by using the ID command (see the *V and V⁺ Reference Guide*).

Training Information

For information regarding Adept Training Courses, please call (408) 434-5024.

Application Information

There is also a dedicated phone line for assistance with robot, vision, and control applications. For applications assistance call (408) 434-5033.

International Customer Assistance

Europe

For information on training, service, or applications, Adept also has a Customer Service Center in Dortmund, Germany. The phone number is: 0231/75 89 40.

Outside Europe or USA

For information on training, service, or applications, call (408) 434-6267.

Chapter 2

Manual Control Pendant Basics

2.1 Programmer's Pendant vs. Operator's Pendant

Adept motion systems are designed to allow control of the robot or motion device from the Manual Control Pendant (MCP). There are two styles of MCPs, the operator's pendant and the programmer's pendant. The programmer's pendant is designed for use while an application is being written and debugged. Figure 2-1 shows how to hold the programmer's MCP. The operator's MCP is designed for use during normal system operation. Figure 2-2 shows how to hold the operator's MCP.

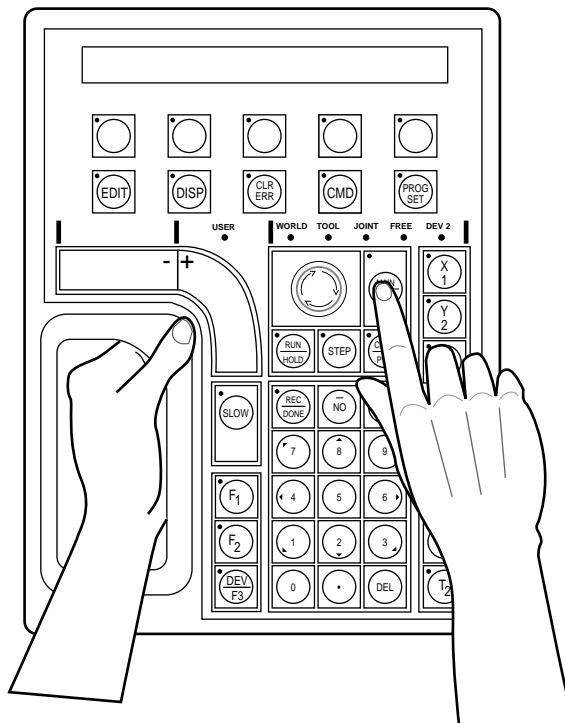


Figure 2-1. Holding the Programmer's MCP

The operator's pendant has a palm activated switch that is connected to the remote emergency stop circuitry of the controller. Whenever this switch is released, ARM POWER is removed from the motion device. When the operator's MCP is not being used, it must be removed from the controller and the pendant jumper plug installed, or placed in the special cradle provided. The cradle retaining clip will keep the arm-power interlock switch depressed when the pendant is not in use. Figure 2-3 shows how to place the operator's pendant in its cradle.

To operate the MCP, put your left hand through the opening on the left-hand side of the pendant and use your left thumb to operate the pendant speed bars. Use your right hand for all the other function buttons. The various button groupings of the pendant are reviewed in this chapter.

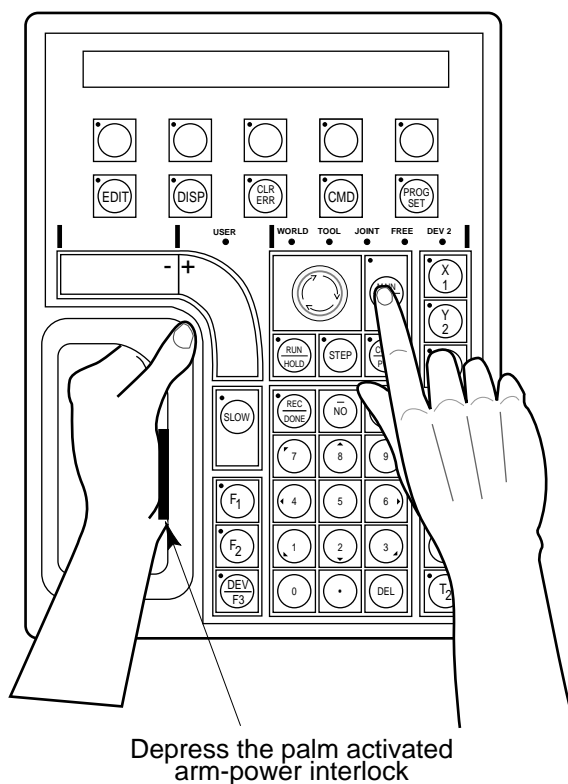


Figure 2-2. Holding the Operator's MCP

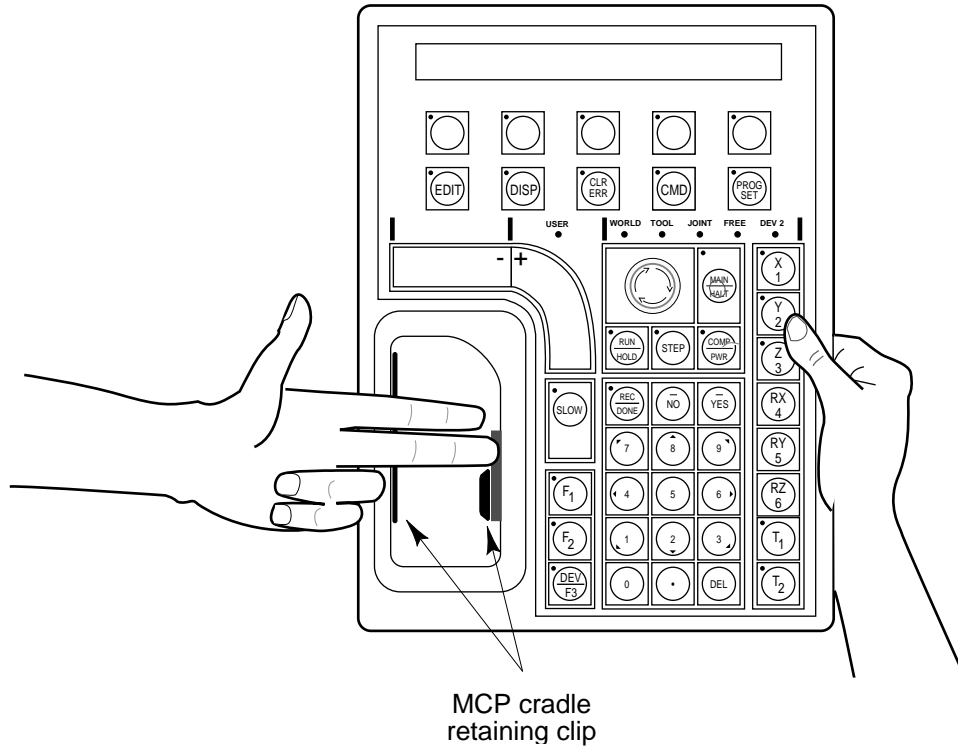


Figure 2-3. Cradling the Operator's MCP



WARNING: The cradle for the operator's pendant **MUST** be mounted outside of the robot or motion device work envelope.

Connecting the MCP

The MCP is connected to the 14-pin connector marked "PENDANT" on the front of the controller. The pendant emergency stop button and the palm-activated arm-power interlock switch are wired into the emergency stop circuitry. Therefore, either the pendant or the pendant jumper plug must be attached to this connector.¹ If neither one is connected, you cannot enable ARM POWER. If the pendant or jumper plug is removed, ARM POWER is shut off. To allow you to attach or remove the pendant without shutting off ARM POWER, the controller is equipped with a "Pendant Remove" button that temporarily bypasses the emergency stop features of the pendant. To attach or remove the pendant, press this button and hold it down while you exchange the pendant connector and the pendant jumper plug. See Figure 2-4.

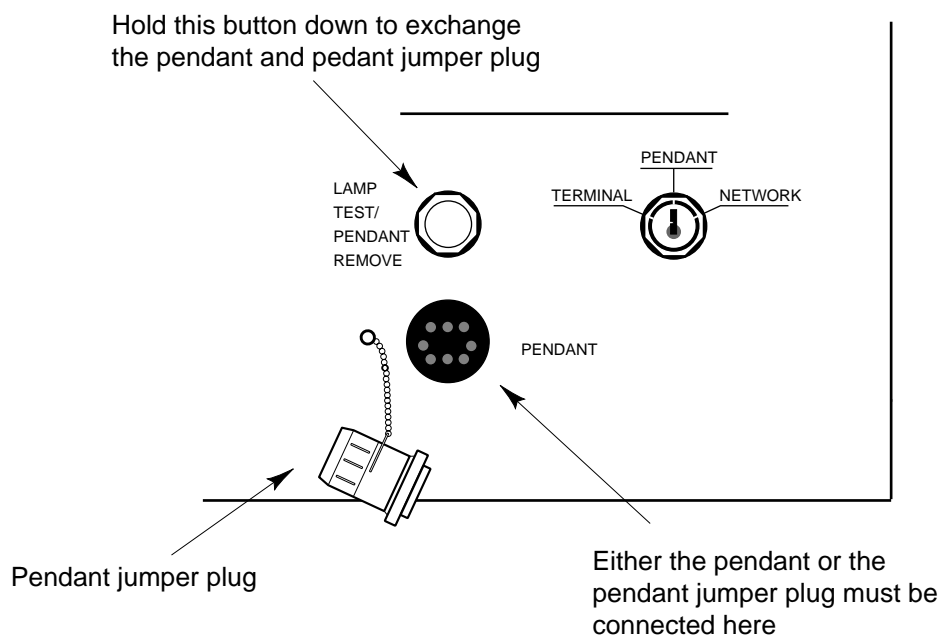


Figure 2-4. MCP Connection

¹. See the user's guide for the Adept MC controller for details on connecting the MCP internally.

2.2 MCP Layout

The major areas of the MCP are shown in Figure 2-5.

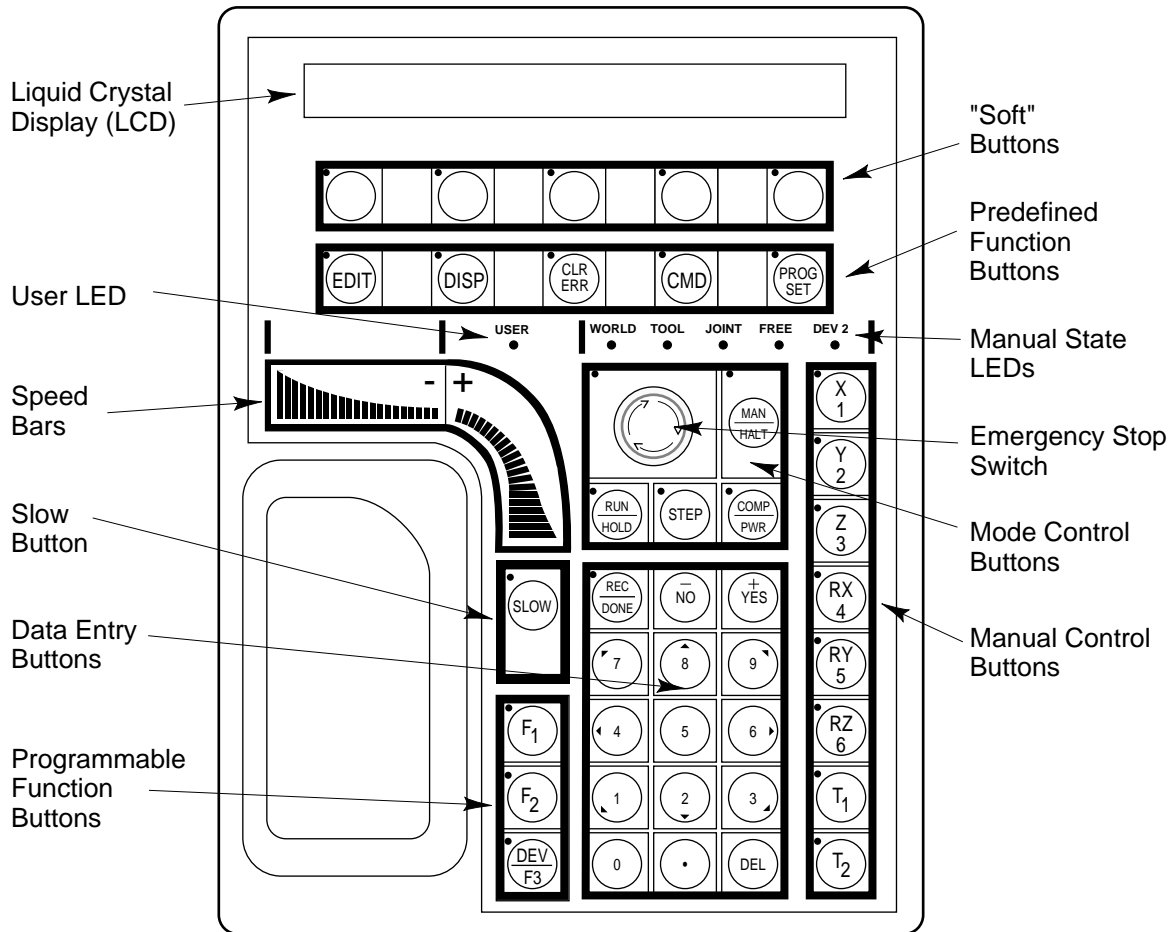


Figure 2-5. MCP Layout

Soft Buttons

The “soft” buttons have different functions depending on the application program being run, or the selection made from the predefined function buttons. Whenever a soft button is active, its function is shown on the bottom line of the pendant display. Because these buttons do not have fixed labels (the labels are defined by the program using the buttons) they are referred to as soft buttons. (Programming the MCP is covered in Chapter 5.) Figure 2-6 shows the soft buttons. In this example, the last two soft buttons have the options “Retry” and “Quit” associated with them.

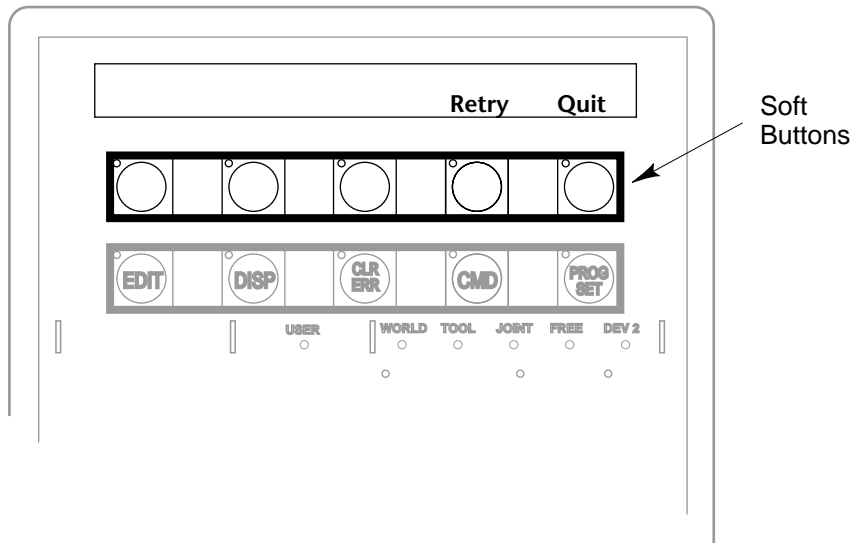


Figure 2-6. Soft Buttons

Data Entry Buttons

The “data entry” buttons are used to input data, normally in response to prompts that appear on the pendant display. The data entry buttons include, +/YES, -/NO, DEL, the numeric buttons (0-9), the decimal point, and the REC/DONE button. These buttons are similar to the numeric keypad on a standard keyboard.

The REC/DONE Button

The REC/DONE button behaves like the Return or Enter key on a normal keyboard. When data entry is complete, pressing REC/DONE sends the entry to the controller. In many cases, applications programs have users press the REC/DONE button to signal that they have completed a task.

The DEL Button

The DEL button acts like the backspace key on a normal keyboard. When data is being entered, it will appear on the pendant display. DEL will delete any characters that appear on the pendant display, but have not been entered using the REC/DONE button. Applications programs may also assign special functions to the DEL button.

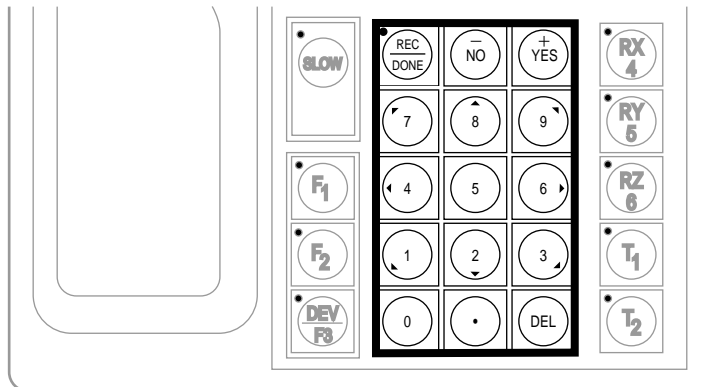


Figure 2-8. Data Entry Keys

Mode Control and Manual Control Buttons

The mode control and manual control buttons are used to control the robot from the pendant. The use of these buttons is covered in Chapter 4.

Speed Bars and Slow Button

The speed bars and slow button are used primarily to move the robot when it is in manual mode. These options are described in Chapter 4.

In some cases, application programs will make special use of the speed bars. See the documentation for the application program for details on how it uses these buttons.

2.3 Emergency Stop From the MCP

To immediately halt program execution and turn off ARM POWER, press the emergency stop button on the MCP. This switch has the same effect as pressing the emergency stop button on the controller.

If you are using the operator's pendant, you may also release the arm-power interlock switch to halt program execution and shut off ARM POWER.

To re-enable arm power after pressing the MCP emergency stop button, turn the emergency stop button to the right (clockwise). The switch is spring loaded and will return to its normal position. If you are using the operator's MCP, depress the palm switch. ARM POWER can now be re-enabled by pressing the "COMP/PWR" button (mode control group), or by entering the ENABLE POWER command from the keyboard.

2.4 Background Mode

The pendant is in background mode when the "USER" LED is not lit and none of the predefined functions are being used. The "USER" LED is lit whenever an application program is making use of the MCP. The MCP will not return to background mode until the program completes execution or is aborted. The LEDs above the predefined function buttons indicate whether the functions are being used. If one of the LEDs is lit, the MCP can be returned to background mode by pressing the REC/DONE key (more than one press may be necessary). The predefined functions are described in Chapter 3.

When the MCP is in background mode, the viewing angle of the LCD can be changed. There are three different angles. Press the "2", "5", or "8" button to select a different viewing angle.

NOTES

Chapter 3

The MCP Predefined Functions

3.1 Introduction

This chapter describes the manual control pendant functions related to:

- Loading and starting programs
- Editing global variables
- Displaying system status

Chapter 4 covers using the MCP to manually control a robot or motion device.

3.2 Predefined Function Buttons

The MCP has five predefined function buttons for your use. They are listed and explained below.

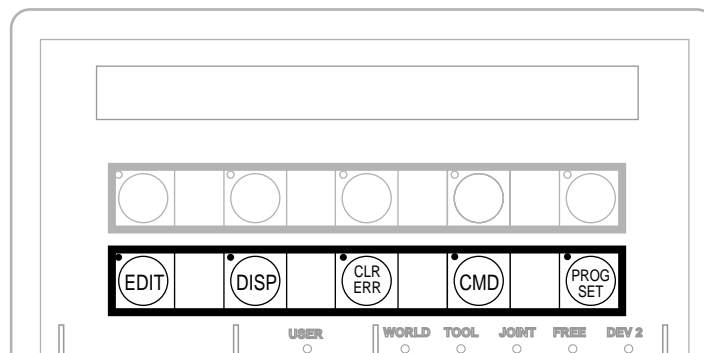


Figure 3-1. MCP Predefined Function Buttons

The Edit Function

The Edit function button allows editing of location variables and real variables that are used by V or V⁺ programs.

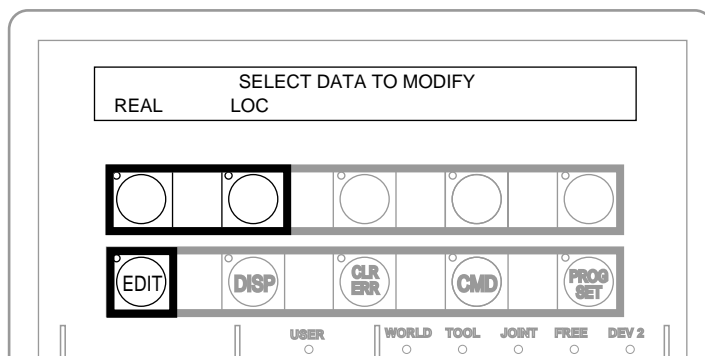


Figure 3-2. EDIT Function Button

Real

Press the “real” soft button and the LCD will display:

```
SELECT REAL VARIABLE TO EDIT
var1   var2   var3   var4   <MORE>
```

“var1”, “var2”, etc., are global variable names. Press the soft button under the variable name to edit that variable. The “<MORE>” soft button is shown only when there are more than five global real variables in system memory. When a variable has been selected, the LCD will show:¹

```
var.name = xxx
CHANGE   TRUE   FALSE
```

Press the “TRUE” soft button to set the variable to the boolean value of true (-1). Press “FALSE” to set the variable to false (0). To change the value of the variable press the “CHANGE” soft button. The LCD will display:

```
var.name = _
CHANGE   TRUE   FALSE
```

The typing cursor has replaced the variable value. Use the data entry buttons to input a new value, and complete the entry by pressing “REC/DONE”.

¹ If the variable being edited is an array, an additional soft button is defined that allows you to specify the index of the array element to edit.

Loc

Press the “LOC” soft button and the LCD will display:

```

      SELECT LOCATION VARIABLE TO EDIT
loc1   loc2   loc3   loc4   <MORE>

```

Press the soft button under the variable name to edit that variable. The “<MORE>” soft button is shown only when there are more than five global location variables in system memory. When a variable has been selected, the LCD will show:²

```

loc.name: X = 500
CHANGE   NEXT   HERE

```

If a precision point is selected, the LCD will show:

```

#loc.name: Jt1 = -210
CHANGE   NEXT   HERE

```

Press the “CHANGE” soft button to change the displayed component of the location variable. The value will disappear and be replaced with the typing cursor. Use the data entry buttons to enter a new value and complete the entry by pressing “REC/DONE”.

Press the “NEXT” soft button to show the next component of the location variable. The location’s X, Y, Z, y, p, and r values will be shown in succession. X, Y, and Z values are given in millimeters; y, p, and r values are given in degrees. If a precision point is being edited, the joint values for all the joints in the robot will be shown in succession.

Press the “HERE” soft button to record the current robot location in the variable being edited.



WARNING: Be extremely careful when changing location values. When the robot moves to a modified location, it could damage equipment in the workcell.

². If the variable being edited is from an array, an additional soft button is defined that allows you to specify the index of the variable to edit.

The Display Function

The Display function button allows either the current joint values, the current world location, the system status, the digital I/O³ status, or the last error message to be displayed on the MCP.

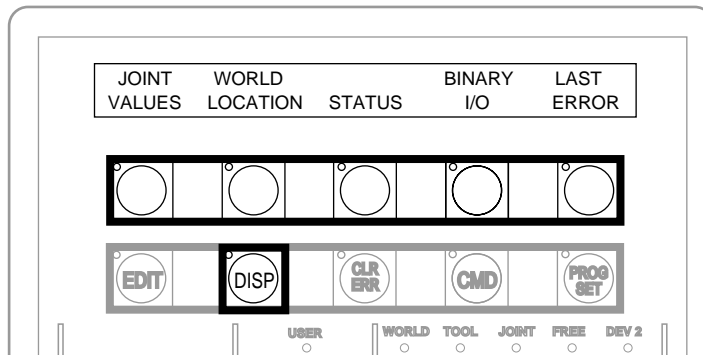


Figure 3-3. DISPLAY Function Button

Joint Values

When this button is pressed, the display will show:

J1 = x.xx J2 = x.xx J3 = x.xx
 J4 = x.xx J5 = x.xx J6 = x.xx

These values represent the current joint positions of the robot or motion device. Values will be shown only for joints the robot or motion device actually has. Rotational joint values are expressed in degrees, and translational joint values are expressed in millimeters. See the next chapter for details on the meaning of values for different joints.

World Location

When this button is pressed, the display will show:

X = xxx.xxmm Y = xxx.xxmm Z = xxx.xxmm
 y = xxx.xx° p = xxx.xx° r = xxx.xx°

The values represent the current location of the robot or motion device in world coordinates. See the next chapter for details on world coordinates.

³ Digital I/O and Binary I/O are synonymous terms. Digital I/O is the preferred term, but occasionally program prompts will use Binary I/O.

Status

When this button is pressed, the display will show:

```

program.name      50          1          0
Program           Speed    Cycle    Left

```

“Program” shows the name of the currently executing or most recently executed program. “Speed” shows the current monitor speed. “Cycle” shows the total number of cycles specified when the program was executed. “Left” shows the number of cycles of the program remaining to execute.

Binary I/O

When this button is pressed, the display will show:

```

---- 1111 0000 ----  ----  ----  ----  ----
+ 0032-0001 -      OUT      IN      SOFT

```

The top line shows the status of the range of digital I/O signals indicated on the second line (1 - 32 in the above example). A “-” indicates the signal has not been configured, a “1” indicates the signal is on, and a “0” indicates the signal is off. The type of signal that is being displayed is indicated by the LED on the soft buttons labeled “OUT”, “IN”, and “SOFT”. The above example shows digital input signals in the range 1 to 32. Signals 25 - 28 are on, signals 21 - 24 are off, and no other signals in this range have been defined.

To display a different range of signals, press the soft buttons under the “+” or “-” labels. The next or previous range of signals will be displayed. Press the “OUT”, “IN”, and “SOFT” soft buttons to display input, output, or soft signal ranges. See the controller user’s guide for details on digital I/O signal ranges.

Last Error

Press “LAST ERROR” to display the error messages generated by V⁺ during the current session. The most recent error will be displayed. The right-most soft button will be labeled “<MORE>”. Pressing this button will cycle back through the error messages generated during the current session.

The Clear Error Function

If the MCP is in the "Pendant" position, or the system switch MCP.MESSAGES is enabled, error messages are sent to the MCP. When an error is sent to the MCP, the MCP will beep, display a blinking error message, and light the LED on the "CLR ERR" button.

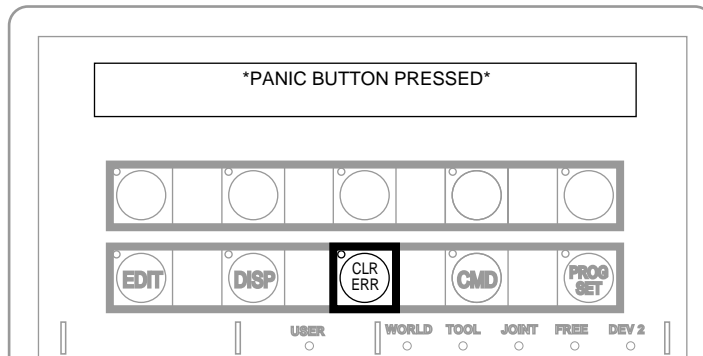


Figure 3-4. CLEAR ERROR Function Button

The "CLR ERR" button must be pressed for operation to continue. Pressing the "CLR ERR" button will clear the error message from the display and return the MCP to the state it was in before the error.

The CMD Function

The CMD function button displays the options AUTO START, CALIBRATE, STORE ALL, CMD1, and CMD2.

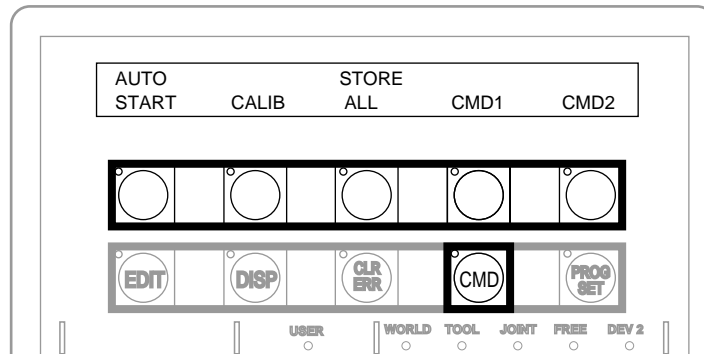


Figure 3-5. Command (CMD) Function Button

Auto Start

The controller key switch must be set to pendant. When “AUTO START” is pressed, the pendant display will show:

```
Enter last two digits of file name:
auto_
```

Enter one or two digits and press REC/DONE. The pendant LCD will prompt you to place the controller key switch in the “TERMINAL” position and press the controller “PROGRAM START” button on the controller. The system will then attempt to load the file AUTOxx.V2 from the default disk, and COMMAND the program “autoxx” (xx refers to the digits you entered). The program file AUTOxx.V2 must reside on the default disk,⁴ and it must contain a monitor command program named “autoxx”. If the file does not exist, or does not contain a correctly named program, the operation will be aborted and the appropriate error message will be displayed on the LCD. For example, if you had entered “9”, the system would attempt to load the file AUTO9.V2 and COMMAND the program “auto9”.

⁴ See the DEFAULT DISK command in the *V and V+ Reference Guide* for details on setting the default disk. See the description of the utility program CONFIG_C in the *Instructions for Adept Utility Programs* for details on setting the startup default disk.

Calib

The controller key switch must be set to pendant. When "CALIB" is pressed, the pendant LCD will prompt you to place the controller key switch in the terminal position and press the controller "Program Start" button. The system will then enable power and calibrate the robot.

Store All

When "STORE ALL" is pressed, the pendant display will show:

```
Enter last two digits of file name:  
STORE auto_
```

Enter one or two digits, press REC/DONE, and all programs and variables in system memory will be stored to a file on the default disk with the name autoxx.v2. For example, if you had entered "11", the file AUTO11.V2 would be created and, all programs and global variables in system memory would be stored to that file.

CMD1 and CMD2

The controller key switch must be set to pendant. When "CMD1" is pressed, the LCD will prompt you to place the controller key switch in the terminal position and press the controller "PROGRAM START" button. The system will then attempt to load the file CMD1.V2 from the default disk, and COMMAND the program "CMD1". The program file CMD1.V2 must reside on the default disk, and it must contain a command program named "cmd1". If the file does not exist, or does not contain a correctly named program, the operation will be aborted and the appropriate error message will be displayed on the LCD. If CMD2 is pressed, the file CMD2.V2 will be loaded and "cmd2" will be COMMANDed.

Prog Set Function

Using the prog set button, you may select a new program to execute, set the starting step number, set how many cycles of the program to perform, set the monitor speed, and start a memory resident application program.

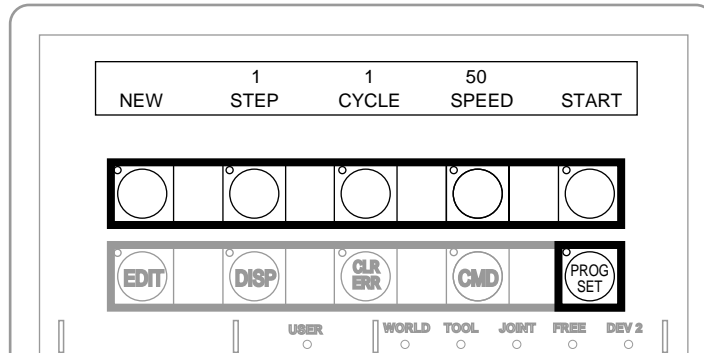


Figure 3-6. Program Set Function Button

New

Press the “NEW” soft button and the LCD will display:

```

                SELECT A NEW PROGRAM
prog1  prog2  prog3  prog4  <MORE>

```

To select a different program, press the soft button under the program name. To see additional programs (if there are more programs), press the “<MORE>” soft button.

Step

Press “STEP” and the step number will blink, and the typing cursor will appear next to the step number. Use the data entry buttons to enter the program step to start execution. Complete the entry by pressing REC/DONE.

Cycle

Press “CYCLE” and the cycle count will blink, and the typing cursor will appear next to the cycle count. Use the data entry keys to enter the number of program cycles to execute. Complete the entry by pressing REC/DONE.

Speed

Press "SPEED" and the current monitor speed will blink and the typing cursor will appear next to the monitor speed. Use the data entry keys to enter a new monitor speed. Complete the entry by pressing REC/DONE.

Start

The "Start" button will work only when the controller key switch is in the pendant position and ARM POWER is enabled (this option cannot be used with DRY.RUN enabled). Press "START" and the LCD will prompt you to put the controller key switch in the "TERMINAL" position and press the "PROGRAM START" button. When the "PROGRAM START" button is pressed, the program displayed above the "NEW" soft button will begin execution.

Chapter 4

Moving a Robot or Motion Device With the MCP

4.1 Introduction

The MCP is used with a robot or motion device primarily to teach robot locations for use in application programs. The MCP is also used with custom applications that employ “teach routines” that pause execution at specified points and allow an operator to teach or re-teach the robot locations used by the program. The Adept AIM software system makes extensive use of the pendant for teaching robot locations.

When you move the robot using the MCP, motion will be in world state, tool state, joint state, or in free state.

When moving in world state, directions are sent from the MCP to move the robot in a Cartesian coordinate system centered at the base of the robot. When moving in tool state, directions are sent from the MCP to move the robot in a Cartesian coordinate system centered at the robot’s end-of-arm tooling location.

In joint state, directions are sent from the MCP to move individual robot joints. In free state, selected joints of the robot are “freed” from servo control so they can be moved by hand.

Mode Control Buttons

The mode control buttons change the state being used to move the robot, switch control of the robot between the MCP and application programs, and enable ARM POWER (when necessary). The buttons are labeled in red.

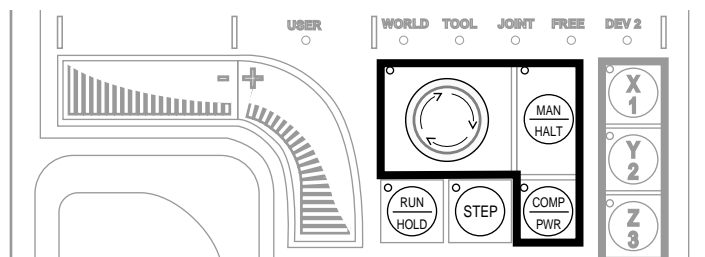


Figure 4-1. Mode Control Buttons

Emergency Stop Button

The emergency stop button will stop program execution and turn off ARM POWER. In the AdeptOne, AdeptThree, and Adept PackOne, the friction brakes will be activated.

COMP/PWR Button

If ARM POWER is enabled, the COMP/PWR button selects computer mode. If ARM POWER is disabled, the COMP/PWR button enables ARM POWER and selects computer mode. In computer mode, an executing program or the system terminal has control of the robot.¹

MAN/HALT Button

When there is no program executing, or a program has paused for a pendant teach routine, pressing the “MAN/HALT” button selects manual mode. In manual mode, the MCP has control of the robot. If a program is executing, the MAN/HALT button will stop program execution (without shutting off ARM POWER).

Manual mode cannot be entered if ARM POWER is off (panic button LED not illuminated). To enable ARM POWER, press the “COMP/PWR” button. The MCP is in manual mode when:

1. The LED on the “MAN/HALT” button is illuminated, and
2. One of the manual state LEDs is also illuminated (the “Manual state” LEDs indicate the type of manual motion that has been selected, either World, Tool, Joint, or Free).

The system will remain in manual mode until ARM POWER is turned off, or the “COMP/PWR” button is pressed. When you have finished moving the robot manually, press the “COMP/PWR” button to return control to the controller. If a program attempts to execute with the MCP in manual mode, the error “Comp mode disabled” will be generated.

When the “MAN/HALT” button is pressed the first time, the MCP will be in world state. Pressing the “MAN/HALT” button again selects the next state to the right (tool, joint, free), eventually wrapping back to the left-most state (world). If manual mode is terminated and re-entered (without turning off system power) the last active state is selected.

Manual Control Buttons

The buttons on the far right side (outlined in blue) are the “Manual control” buttons. When the MCP is in manual mode, these buttons select

¹ If the robot has not been calibrated and ARM POWER is turned on, the MCP emergency stop switch LED will be lit, and both the COMP/PWR and MAN/HALT LEDs will be off.

which robot joint will move, or the coordinate axis along which the robot will move. The X/1, Y/2, Z/3, RX/4, RY/5, and RZ/6 buttons are covered in section 4.2. (The MCP must be in manual mode before a manual control button can be selected.)

Speed Bars

The “speed bars” are used to control the robot’s speed and direction. The joint(s) that will move when the speed bars are pressed depends on the “state” selected with the MAN/HALT button. Press the speed bars with your left thumb. Pressing the speed bars near the outer ends will move the robot faster, pressing the speed bar near the center will move the robot slower. See Section 4.2 for details on positive and negative directions.

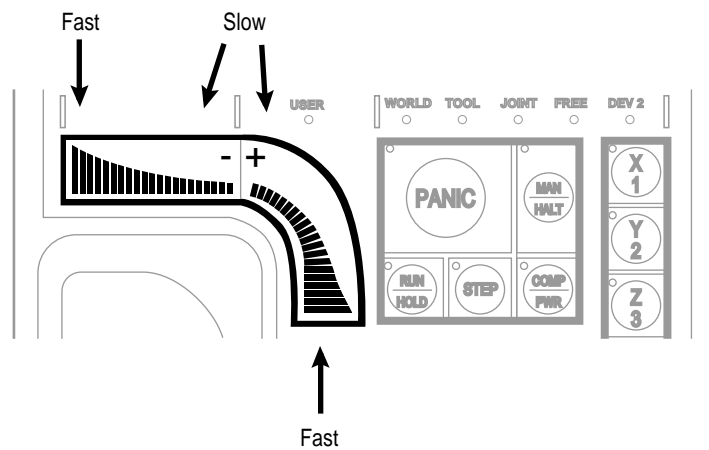


Figure 4-2. Speed Bars

Slow Button

The slow button selects between the two different speed ranges of the speed bars. When the slow button LED is lit, the slower speed range is selected.

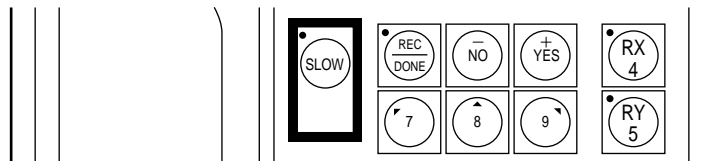


Figure 4-3. SLOW Button

4.2 Robot States

World State

When "world" state is selected, movement in the X, Y, or Z direction is parallel to an axis of the world coordinate system. Before the speed bars will move the robot, an axis of motion must be selected from the manual control buttons. The world coordinate system for a SCARA robot is shown in Figure 4-4. If "X1" is selected, pressing the "+" speed bar will move the robot tool flange in the positive X direction. Pressing the "-" speed bar will move the flange in the negative X direction. The world coordinate system for a Cartesian robot is shown in Figure 4-5. If "X1" is selected, pressing the "+" speed bar will move the robot tool flange in the positive X direction. Pressing the "-" speed bar will move the flange in the negative X direction.

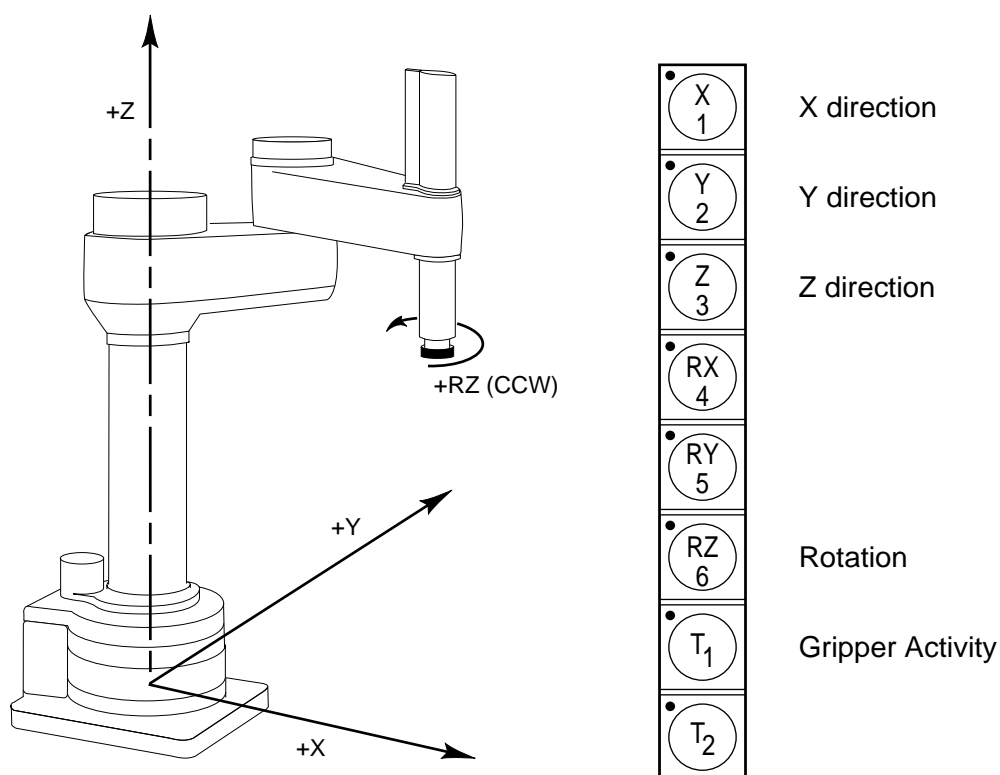


Figure 4-4. WORLD State (SCARA)

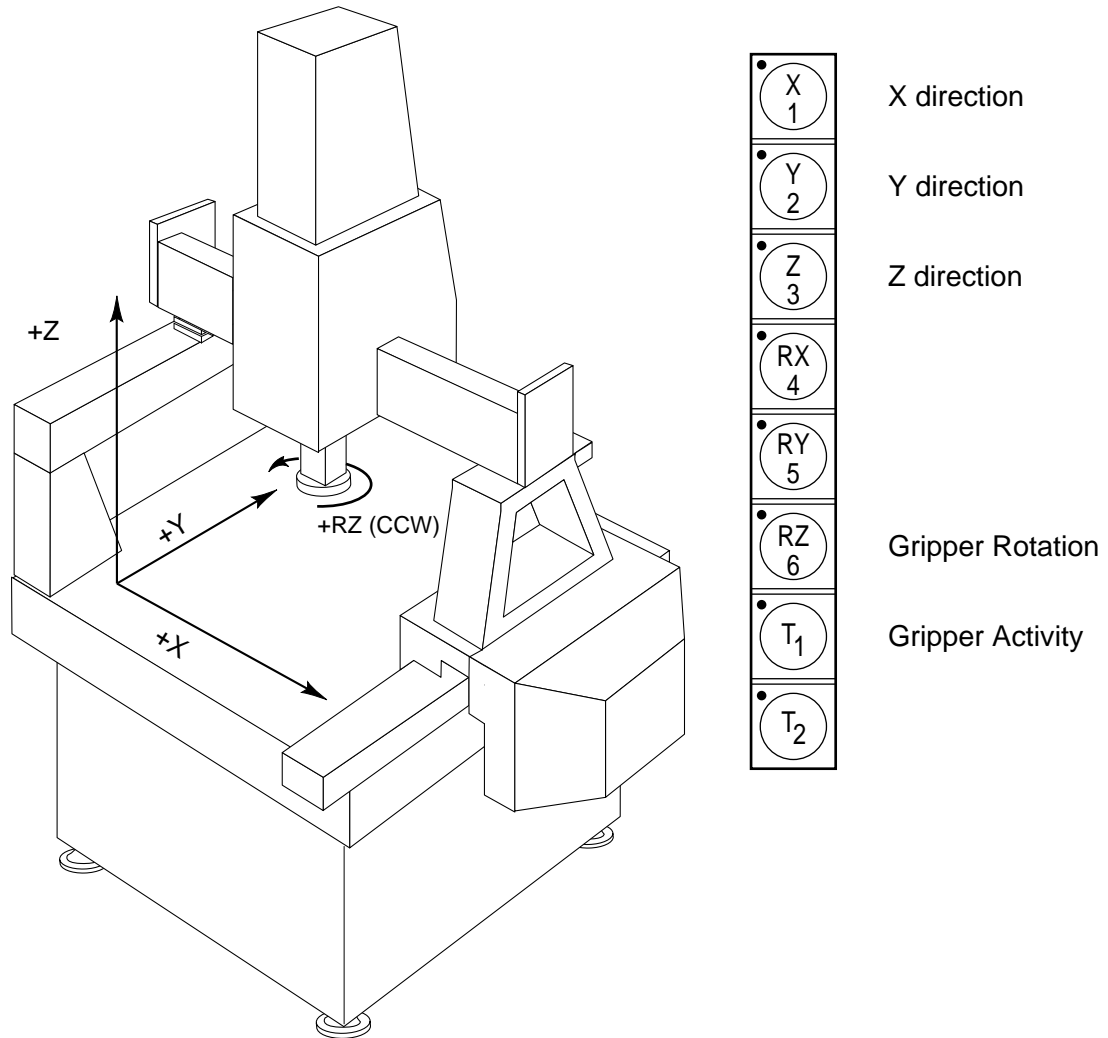


Figure 4-5. WORLD State (Cartesian)

The T₁ button cycles the gripper solenoids. Press anywhere on the “+” side of the speed bar to open the gripper, on the “-” side to close the gripper.

NOTE: This is the most common gripper setup. The gripper solenoids may be configured so they operate differently (or they may not be configured at all). Place your robot in a safe location and cycle the gripper to verify which side of the speed bar opens the gripper.²

². The utility CONFIG_R is used to configure gripper activity. See the *Instructions for Adept Utility Programs*.

Tool State

When “tool” state is selected, movement in the X, Y, or Z direction is along an axis of the tool coordinate system. The tool coordinate system is centered at the robot tool flange with the Z axis pointing away from the flange. The positive X axis is aligned with the center of the tool flange keyway. Before the speed bars will move the robot, an axis of motion must be selected from the manual control buttons. If “X1” is selected, pressing the “+” speed bar will move the robot tool flange in the positive X direction. Pressing the “-” speed bar will move the flange in the negative X direction.

In a four-axis robot, positive rotation of the gripper is clockwise as viewed from above. Figure 4-6 shows the tool state for a four-axis SCARA robot.

Figure 4-7 shows the tool coordinate system on a six-axis robot.

NOTE: Figure 4-6 and Figure 4-7 are drawn with the assumption that the TOOL transformation is set to NULL (all values are 0). If a TOOL transformation is in effect, the tool coordinate system will be offset and rotated by the value of the TOOL transformation. Any motion in tool state will now be relative to the offset coordinate system, and not the center of the tool flange. See the *V and V⁺ Reference Guide* for details on TOOL transformations.

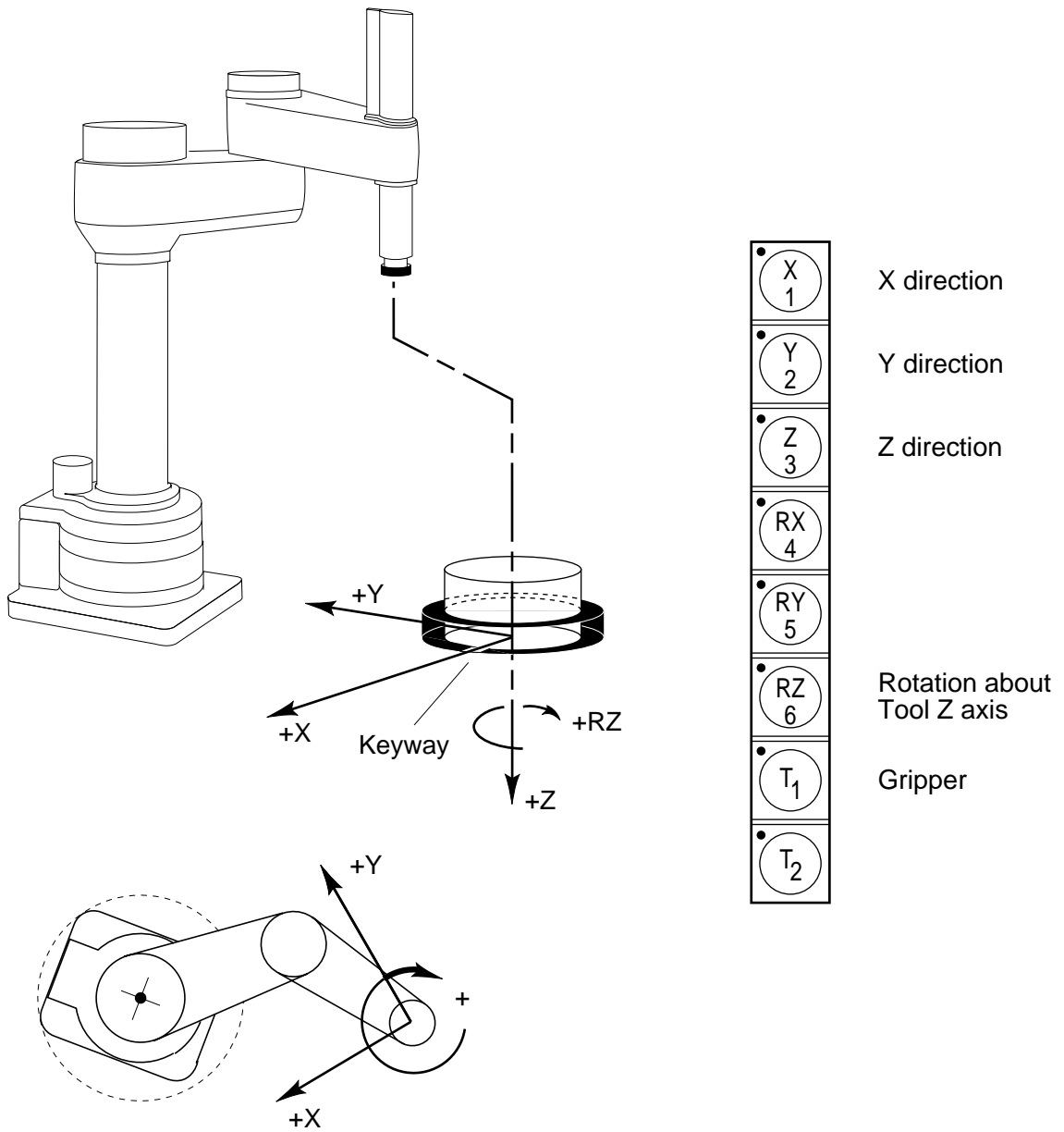


Figure 4-6. TOOL State

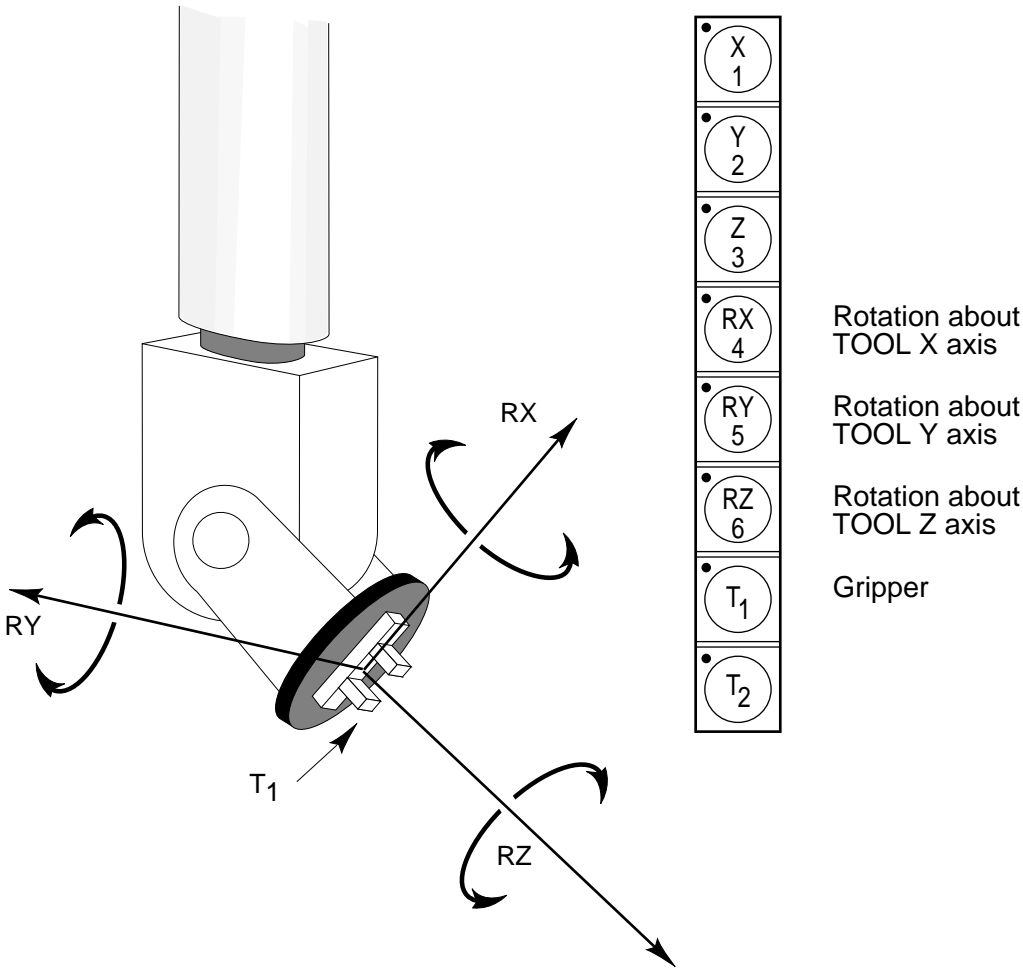


Figure 4-7. TOOL State (Six-Axis Robot)

Joint State

When joint state is selected, movement is about the axis of the specified joint. Figure 4-8 shows an Adept SCARA robot with three rotational joints (joints 1, 2, and 4) and one translational joint (joint 3). Positive rotation of joints 1 & 2 is counter-clockwise as viewed from above. Positive rotation of joint 4 is clockwise as viewed from above. Positive movement of joint 3 is downward. Before the speed bars will move a joint, the correct joint must be selected from the manual control buttons.

Different types of motion devices will have the different joint numbers assigned to their joints. When you first move an unfamiliar robot using joint state, set the monitor speed to 10 or lower, put the robot in a safe area, and carefully move the robot using the different joint numbers to verify how the MCP moves the robot. See the documentation for the motion devices you are using for details on their joint assignments.

Figure 4-9 shows an Adept Cartesian robot with three translational joints and one rotational joint.

Figure 4-10 shows the joint assignments for a typical six-axis robot (as always, the first time you move a robot, carefully verify the joint assignments).

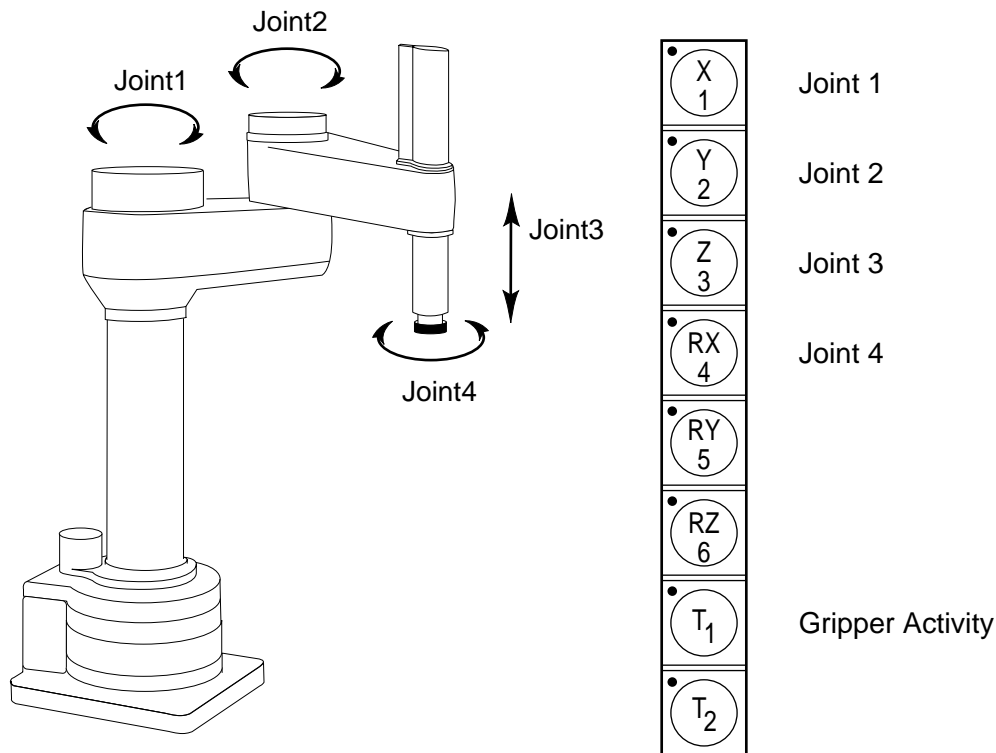


Figure 4-8. JOINT State (SCARA)

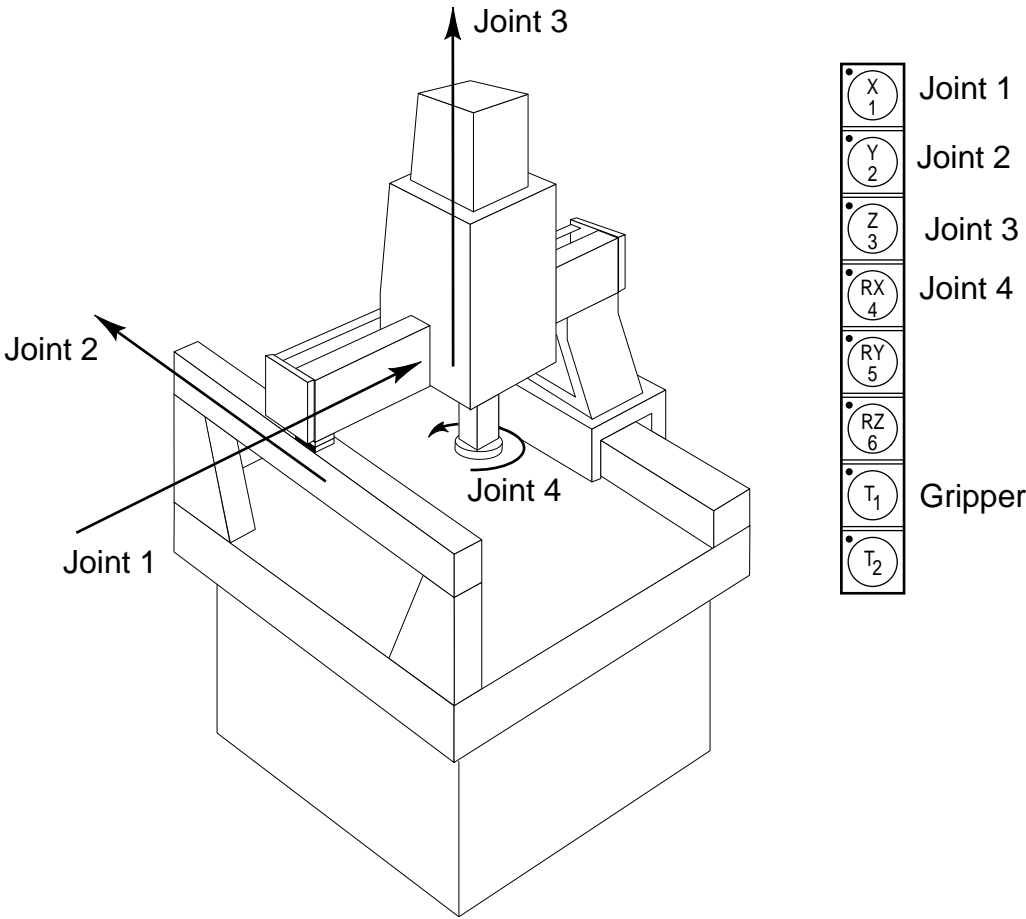


Figure 4-9. JOINT State (Cartesian)

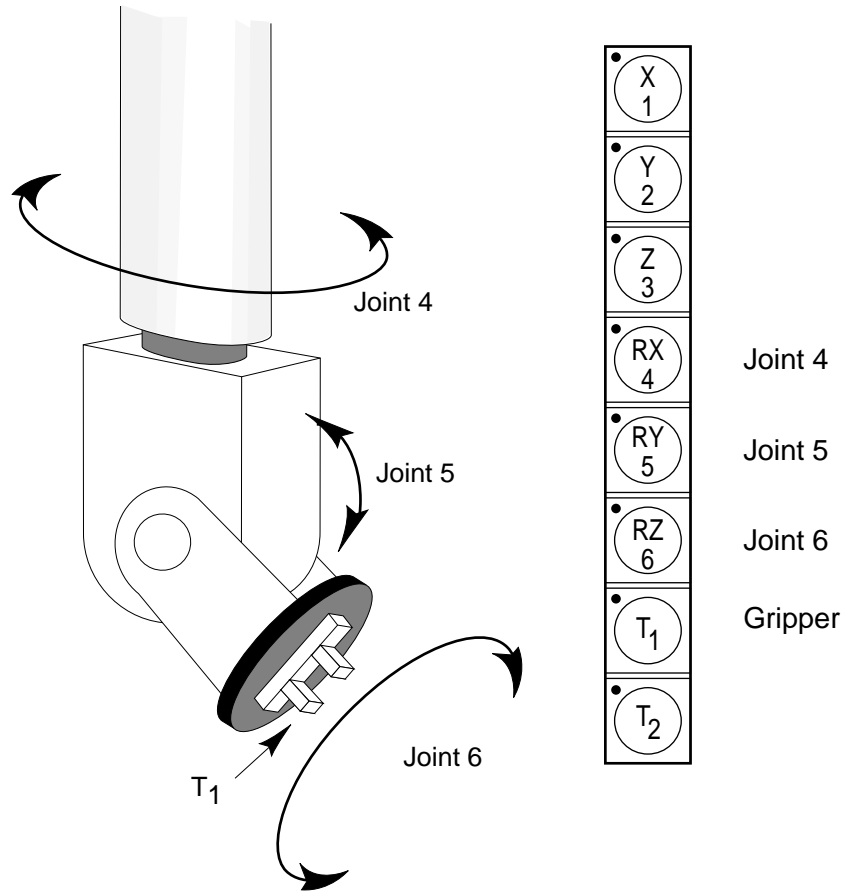


Figure 4-10. JOINT State (Six-Axis Robot)

Free State

When free state is selected, individual joints are freed from servo control, and the robot brakes (if any) are released.³ Unlike the other states, you can make multiple selections from the manual control buttons to free as many joints as required. In some cases, such as joints 1 & 2 on a SCARA robot, multiple joints are freed by selecting a single button. As soon as the "COMP/PWR" button is pressed, or another selection is made from the manual control buttons, all joints are placed back under servo control and will not move freely.

Figure 4-11 shows the free state for a four-axis SCARA robot.

The joint assignments in the free state are the same as the joint assignments in joint state. See Figure 4-7 for the joint assignments in a typical six-axis robot.



WARNING: As soon as a joint is selected from the manual control buttons, the related joint is free to move (in some cases, multiple joints may be freed up). In many cases the weight on the joint will be sufficient to move the joint and cause damage or harm. For example, when joint 3 on a SCARA or Cartesian robot is freed, the joint is free to fall to the end of its travel. In articulated robots, multiple links of the robot may be free to fall when a single joint is freed up. Be extremely careful when selecting a joint in free mode.

³ On the Adept 604-S robot, the joint 3 brake is never released. To move this joint in free state, you must use enough pressure to overcome the friction brake.

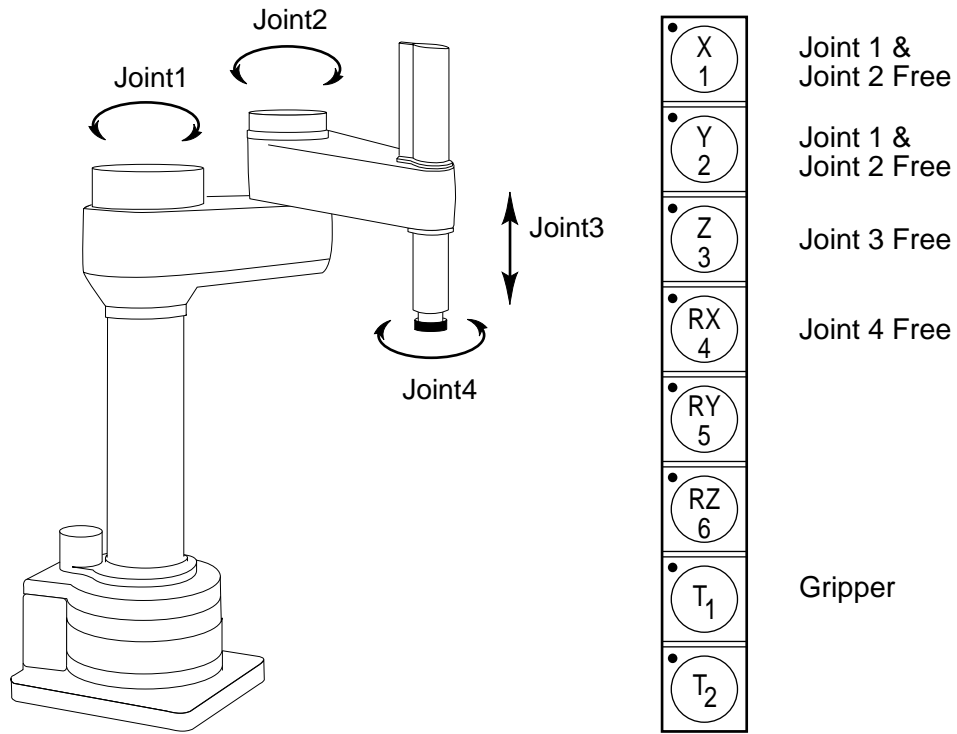


Figure 4-11. FREE State (Four-Axis SCARA)

4.3 Selecting Different Robots

If your system is equipped with more than one robot or motion device, you can select which device the manual control buttons will affect. When manual mode is first entered, device 1 will be selected. To select device number 2, press the DEV/F3 button. Device number 2 will now be selected, and the "DEV 2" light in the manual state LED group will be lit. To select device number 3, press the DEV/F3 button again. Device 3 will be selected, and the "DEV 2" LED will blink. To re-select device 1, press the DEV/F3 button.

Remember: When the "DEV 2" LED is off, device 1 is selected; when it is lit continuously, device 2 is selected; when it flashes, device 3 is selected.

Chapter 5

Programming the Pendant

5.1 Writing to the Pendant Display

This chapter provides an overview of strategies for programming the manual control pendant. You must be familiar with basic V⁺ programming to program the MCP. V/V⁺ programming is not covered in this manual; we assume you are familiar with the V⁺ language. This chapter covers the instructions used specifically for the MCP.

The Pendant Display

The MCP display is a 2-line, 80-character LCD display. It is written to and read from using the WRITE and READ instructions. These instructions are detailed in the *V and V⁺ Reference Guide*. This chapter covers the capabilities of these instructions that pertain to programming the MCP.

ATTACHing to the Pendant

Before an application program can communicate with the MCP, the MCP must first be ATTACHed using the ATTACH instruction. The logical unit number for the MCP is 1. The following instruction will ready the MCP for communication:

```
ATTACH (1)
```

When the MCP is ATTACHed, the “USER” LED on the MCP will be lit.

As with all other devices that are ATTACHed by a program, the MCP should be DETACHed when the program is finished with the MCP. The following instruction will free up the MCP:

```
DETACH (1)
```

When the MCP has been ATTACHed by an application program, the user can interact with the pendant without putting the controller key switch in the “Pendant” position.

The success of an ATTACH or DETACH instruction (or any other I/O instruction) can be checked with the IOSTAT() function. Robust programming includes a check for success of all I/O instructions. For clarity, the examples in this chapter leave this check out.

Using WRITE With the Pendant

The following instructions will display a welcome message on the two lines of the pendant display:

```
$intro = "Welcome to the MCP"  
ATTACH (1)  
WRITE (1) $intro  
WRITE (1) "Instructions to follow...", /S
```

Notice that the second WRITE instruction uses the “/S” qualifier. This qualifier suppresses the carriage return-line feed (<CR-LF>) that is normally sent by the WRITE instruction. If this qualifier was not specified, the first line displayed would have been overwritten. In section 5.3 we will detail the pendant control codes. These codes control the cursor position, the lights on the MCP, and the interpretation of MCP button presses. These codes are sent to the pendant using the WRITE instruction. The “/S” qualifier must be sent with these instructions to avoid overwriting the pendant display.

5.2 Detecting User Input

Input from the pendant can be received in two ways:

- A series of button presses from the data entry buttons (see Figure 2-5) can be read. The READ instruction is used for this type of input.
- A single button press from any of the buttons can be detected. These single button presses can be monitored in three different modes:
 - a. The buttons can be monitored like keys on a normal keyboard.
 - b. The buttons can be monitored in “toggle” mode (“on” or “off”). The state of the button is changed each time the button is pressed.
 - c. The keys can be monitored in “level” mode. The state of the button is considered “on” only when the button is held down.

The PENDANT() function is used to detect button presses in these modes. The KEYMODE instruction is used to set the button behavior.

Using READ With the Pendant

The READ instruction accepts input from the pendant “Data Entry Buttons” (1, 2, 3, 4, 5, 6, 7, 8, 9, 0, ., +, -). A READ instruction expects a <CR-LF> to indicate the end of data entry. On the MCP, this sequence is sent by the “REC/DONE” button (similar to the Enter or Return key on a normal keyboard). The “DEL” button behaves like the “Backspace” key on a normal keyboard. All other pendant buttons are ignored by the READ instruction. Note, the pre-defined function buttons are active and may be used while an attached program is waiting for input.

The instruction line:

```
READ(1) $response
```

will pause the program and wait for input from the pendant. The user must signal the end of input by pressing the “REC/DONE” button. The input will be stored in the string variable “\$response”. The input can be stored as a real variable, but the “+” and “-” buttons must not be used for input.

Detecting Pendant Button Presses

Individual MCP button presses are detected with the PENDANT() function. This function returns the number of the first acceptable button press. The interpretation of a button press is determined by the KEYMODE instruction. See the *V and V+ Reference Guide* for complete details. The basic use of these two operations is described below.

Keyboard Mode

The default mode is “keyboard”. If a PENDANT() instruction requests keyboard input, the button number of the first “keyboard” type button pressed will be returned. See Figure 5-1 for the numbers of the buttons on the MCP. The following code will detect the first soft button pressed:

```
; Set the soft keys to keyboard mode
KEYMODE 1,5 = 0
; Wait for a button press from buttons 1 - 5
DO
    button = PENDANT(0)
UNTIL button < 6
```

The arguments to the KEYMODE instruction indicate that pendant buttons 1 through 5 are to be configured in “keyboard” mode. The “0” argument to the PENDANT() function indicates that the button number of the first “keyboard” button pressed is to be returned.

Toggle Mode

To detect the state of a button in toggle mode, the PENDANT() function must specify the button to be monitored.

When a button is configured as a toggle button, its state is maintained as “on” (-1) or “off” (0). The state is toggled each time the button is pressed. If an LED is associated with the button, it is also toggled. The following code sets the “REC/DONE” button to toggle mode and waits until “REC/DONE” is pressed:

```
; Set the REC/DONE button to toggle
KEYMODE 8 = 1
; Wait until the REC/DONE button is pressed
DO
    WAIT
UNTIL PENDANT(8)
```

The arguments to KEYMODE indicate that MCP button number 8 (the “REC/DONE” button) is configured as a toggle button. The argument to PENDANT() indicates that the state of MCP button 8 is to be read.

Level Mode

To detect the state of a button in level mode, the PENDANT() function must specify the button to be monitored.

When a button has been configured as a level button, the state of the button is “on” as long as the button is pressed. When the button is not pressed, its state is “off”. The following code uses the buttons labeled “2”, “4”, “6”, and “8” (button numbers 45, 47, 49, and 57—don’t confuse the button labels with the numbers returned by the PENDANT function) to move the cursor around the terminal display. The buttons are configured as level buttons so the cursor moves as long as a button is depressed.

```
; Set the REC/DONE button to toggle
KEYMODE 8 = 1

; Set the data entry buttons labeled "2" - "8" to level
KEYMODE 45, 51 = 2
DO
    IF PENDANT(49) THEN
        TYPE /X1, /S           ;cursor right
    END
    IF PENDANT(47) THEN
        TYPE $CHR(8)         ;cursor left (backspace)
    END
    IF PENDANT(51) THEN
        TYPE /U1, /S         ;cursor up
    END

    IF PENDANT(45) THEN
        TYPE $CHR(12)        ;cursor down (line feed)
    END

UNTIL PENDANT(8)
```

Monitoring the MCP Speed Bar

The speed bar on the MCP returns a value from -128 to 127 depending on where it is being pressed. An argument of "-t2" to the PENDANT() function will return the value of the speed bar. The following code displays the state of the speed bar.

```
; Set the REC/DONE button to toggle
KEYMODE 8 = 1

; Display speed bar value until the REC/DONE is pressed
DO
    WRITE(1) PENDANT(-2)
UNTIL PENDANT(8)
```

The "Slow" button is intended to alter the value returned by the speed bar. The following code compresses the range of values returned by 50% whenever the "Slow" button is "on".

```
; Set the REC/DONE button to toggle
KEYMODE 8 = 1

; Do until the REC/DONE button is pressed
DO
    IF PENDANT(36) THEN
        TYPE PENDANT(-2) * 0.5
    ELSE
        TYPE PENDANT(-2)
    END
UNTIL PENDANT(8)
```

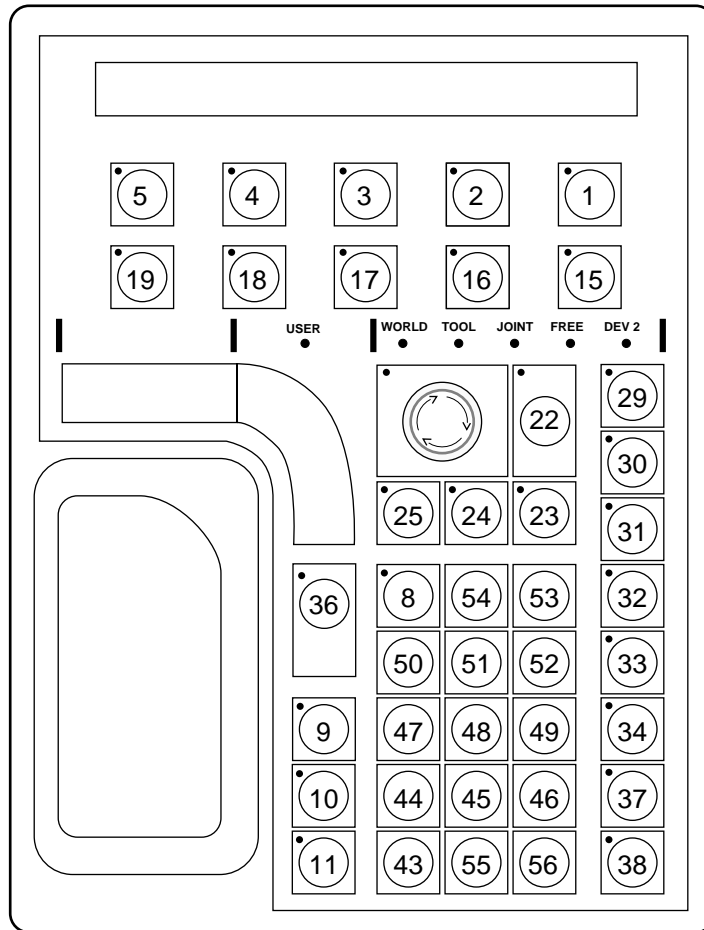


Figure 5-1. MCP Button Map

Reading the State of the MCP

It is good programming practice to check the state of the MCP before ATTACHing to it. The instruction:

```
cur.state = PENDANT(-3)
```

will return a value to be interpreted as follows:

- 1 Indicates that one of the predefined function buttons has been pressed.
- 2 Indicates the MCP is in background mode (not ATTACHed to an application program).
- 3 Indicates an error is being displayed.
- 4 Indicates that the MCP is in USER mode (ATTACHed to an application program).

See Section 5.5 for a program example that checks the MCP state.

5.3 Controlling the Pendant

The MCP responds to a number of control codes that affect the LCD panel, whether or not the buttons are “repeat” buttons, and the LEDs associated with the pendant buttons. The control codes are listed in Table 5-1. The control codes are sent as ASCII values using the WRITE instruction. The normal way to send control codes is to use the \$CHR() function to convert a control code to its ASCII value.

Control Codes for the LCD Panel

To clear the display and position the cursor in the middle of the top line, issue the instruction:

```
WRITE(1)
$CHR(12), $CHR(18), $CHR(20), /S
```

“\$CHR(12)” clears the pendant and places the cursor at position 1 (see Figure 5-2). “\$CHR(18)” indicates that the next value received should be interpreted as a cursor location. “\$CHR(20)” indicates the cursor should be placed at position 20. “/S” must be appended to the WRITE instruction or a <CR-LF> will be sent. Notice that using control code 18 allows you to position the cursor without disturbing existing text.

The following code will place the text “EXIT” in the middle of the bottom line and set the text blinking.

```
WRITE(1) $CHR(18), $CHR(58), "EXIT", /S
WRITE(1) $CHR(18), $CHR(58), $CHR(22), $CHR(4), /S
```

“\$CHR(22)” tells the pendant to start a series of blinking positions starting at the current cursor location and extending for the number of positions specified by the next control code (“\$CHR(4)”). This code will cause any text in positions 58 - 62 to blink until an instruction is sent to cancel the blinking. The following code line dabbles the blink positions:

```
WRITE(1) $CHR(18), $CHR(58), $CHR(23), $CHR(4), /S
```

“\$CHR(23)” tells the pendant to cancel a series of blinking positions starting at the current cursor location and extending for the number of positions specified by the next control code (“\$CHR(4)”).

Text can be made to blink as it is written to the display, regardless of the position the text is in. The following code writes the text "EXIT" to the middle of the bottom line, starts the "E" blinking, and then beeps the MCP:

```
WRITE(1) $CHR(18), $CHR(58), $CHR(2), "E", /S  
WRITE(1) $CHR(3), "XIT", /S  
WRITE(1) $CHR(7), /S
```

"\$CHR(2)" starts blink mode. Any characters sent to the MCP display will blink. Blink mode is canceled by "\$CHR(3)". "\$CHR(3)" cancels blink mode for subsequent characters; it does not cancel blinking of previously entered characters. It also does not cancel blinking of character positions set by control code 22. "\$CHR(7)" causes the pendant to beep.

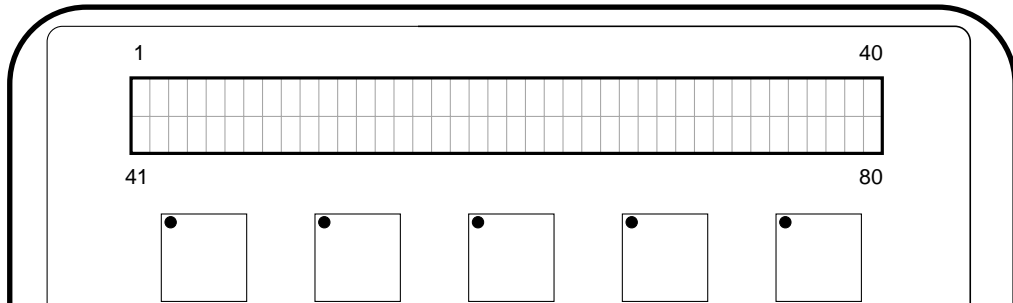


Figure 5-2. Pendant LCD Display

The Pendant LEDs

The LEDs on the soft buttons, the "F" buttons, and the "REC/DONE" button can be lit (either continuously or intermittently). The following code places the text "CLEAR" and "EXIT" over the first two soft buttons, lights the LED over the first soft button, and blinks the light over the second soft button:

```
WRITE(1) $CHR(18), $CHR(41), "CLEAR", /S
WRITE(1) $CHR(9), "EXIT", /S
WRITE(1) $CHR(31), $CHR(5)
WRITE(1) $CHR(30), $CHR(4)
```

"\$CHR(9)" tabs the cursor to the next soft button position. "\$CHR(31)" lights an LED. "\$CHR(30)" starts an LED blinking. The button LED to be lit is specified in the ensuing control code. In the above example, button 5's LED is turned on and button 4's LED is set blinking. The soft buttons, "F" buttons, and REC/DONE button are the only buttons that have programmable LEDs.

Making Pendant Buttons Repeat Buttons

Pendant buttons that are configured as "keyboard" buttons are normally repeat buttons, button presses are recorded as long as the button is held down. The repeat function can be disabled, requiring users to press the button once for each button press they want recorded. The following instruction disables the repeat option for the period (.) button:

```
WRITE(1) $CHR(25), $CHR(55)
```

The repeat option is enabled with the instruction:

```
WRITE(1) $CHR(24), $CHR(55)
```

Table 5-1 lists all the control codes used with the pendant.

Table 5-1. Pendant Control Codes

Single Byte Control Codes		
Code	Function	
1	(Not Used)	
2	Enable blink mode for subsequent characters	
3	Disable blink mode for subsequent characters (characters will still blink if they appear in a blinking position set by code 22)	
4	Display cursor (make the cursor visible)	
5	Hide cursor (make the cursor invisible)	
6	(Not Used)	
7	Beep	
8	Backspace (ignored if cursor is in character position 1)	
9	Tab to next soft button	
10	Line feed (move down in same position, scroll if on line 2)	
11	Vertical tab (move up in same position, do not scroll)	
12	Home cursor and clear screen (cancels any blinking positions, but does not affect blink mode set by code 2)	
13	Carriage return (move to column 1 of current line)	
14	Home cursor (move to character position 1)	
15	Clear from cursor position to end of line	
Double Byte Control Codes		
Code	Function	Second Code
16	(Not Used)	
17	(Not Used)	
18	Position cursor	Cursor position (1-80)
19	(Not Used)	
20	(Not Used)	
21	(Not Used)	
22	Enable blinking positions starting at current cursor location	Number of blinking positions (1-80)
23	Disable blinking position starting at current cursor location	Number of blinking positions (1-80)
24	Enable repeat mode for button	Button number
25	Disable repeat mode for button	Button number
26	(Not Used)	
27	(Not Used)	
28	Turn off pendant button LED	Light number *
29	(Not used)	
30	Start pendant button LED blinking	Light number *
31	Turn on pendant button LED	Light number *

*For soft buttons, "F" buttons, and REC/DONE button only

5.4 Auto-Starting Programs With the MCP

The "CMD" predefined function button provides three options for loading and auto-starting a program from the pendant. These three options are "AUTO START", "CMD1", and "CMD2". The program file requirements for all three options are the same:

1. The file being loaded must be on the default disk. The default disk is specified with the DEFAULT DISK command. The utility CONFIG_C can be used to specify a default disk at startup.¹ See the *Instruction for Adept Utility Programs* for details on running this utility.
2. The file name must correspond to the MCP selection. If "CMD1" is pressed, the disk file must be named CMD1.V2. If "AUTO START" is pressed, the user will be asked to input one or two digits. These digits will be used to complete the file name AUTOxx.V2. A corresponding file name must exist on the default drive.
3. A command program with the same name as the file name (minus the extension) must be one of the programs in the file. If AUTO22.V2 is loaded, the program "auto22" will be COMMANDed. See the *V and V+ Reference Guide* for details on command programs.

WAIT.START

Starting a robot program while the operator is in the workcell can be extremely dangerous. Therefore, Adept has installed the following safety procedure to prevent program startup while an operator is in the workcell. Before a program auto-started from the MCP will begin execution, the operator will have to leave the workcell, put the controller key switch in the terminal position, and press the "Program Start" button. The WAIT.START instruction implements this safety feature. This instruction is automatically included in any programs started with the AUTO START, CMD, CMD1, CMD2, and CALIBRATE buttons on the MCP. You should include this safety feature in any pendant routines you write that initiate monitor command programs that include robot motions. The command "WAIT.START" in a monitor command program will pause execution of a monitor command program until the key switch is correctly set and the PROGRAM START button is pressed. See the *V and V+ Reference Guide* for other uses WAIT.START.



WARNING: For this safety feature to be effective, the controller must be installed outside the workcell.

¹. The default disk is not the same as the boot drive. The boot drive is set in hardware and is used during the boot procedure to specify the drive that contains the operating system. Once the system is loaded, the default disk is the drive and path specification for loading and storing files.

5.5 Programming Example: MCP Menu

The following code implements a menu structure on the MCP.

```
.PROGRAM mcp.main( )
; ABSTRACT: This program creates and monitors a menu structure on the
;   MCP.
;
; INPUT PARMS: None
;
; OUTPUT PARMS: None
;
; GLOBAL VARS: mcp - MCP logical unit
                pe.clr.scr - pendant control code, clear display & home cursor
                pe.cur.pos - pendant control code, set cursor position
                pe.off.led - pendant control code, turn off an LED
                pe.blink.char - pendant control code, start blink position
                pe.noblink.char - pendant control code, disable blink position
                pe.beep - pendant control code, beep the pendant
                pe.tab - pendant control code, tab to next soft button
                pe.on.led - pendant control code, turn on an LED

LOCAL button   ;number of the soft button pressed
LOCAL quit     ;boolean indicating menu structure should be exited

mcp = 1
quit = FALSE
pe.clr.scr = 12
pe.cur.pos = 18
pe.off.led = 28
pe.blink.char = 2
pe.noblink.char = 3
pe.beep = 7
pe.tab = 9
pe.on.led = 31

; Check to see if the MCP is free

    IF PENDANT(-3) <> 2 THEN
        GOTO 100
    END

; Attach to the MCP

    ATTACH (mcp)

; Verify ATTACH was successful

    IF IOSTAT(mcp) <> 1 THEN
        GOTO 100
    END
```

```
DO ;main processing loop

; Display the top level menu

    CALL mcp.disp.main( )

; Get the operator selection (must be between 1 and 5)

    DO
        button = PENDANT(0)
    UNTIL (button < 6)

; Turn on the LED of the selected button

    WRITE (mcp) $CHR(pe.on.led), $CHR(button), /S

; Respond to the menu item selected

    CASE button OF
        VALUE 1:          ; Verify program exit
            CALL mcp.main.quit(quit)

        VALUE 2:
            CALL mcp.option.2( )

        VALUE 3:
            CALL mcp.option.3( )

        VALUE 4:
            CALL mcp.option.4( )

        VALUE 5:
            CALL mcp.option.5( )

; Turn off LED

    WRITE (mcp) $CHR(pe.off.led), $CHR(button)

    END ;CASE button of

    UNTIL quit

; Detach from the MCP

    DETACH (mcp)

100 IF NOT quit THEN      ;Exit on MCP busy
    TYPE /C22, /U15, "The MCP is busy or not connected."
    TYPE "Press the REC/DONE button to clear.", /C5
    END

.END
```

```
.PROGRAM mcp.disp.main( )

; ABSTRACT: This program is called to display a main menu above the five
; soft keys on the MCP. The program assumes the MCP has been attached.
;
; INPUT PARMS: None
;
; OUTPUT PARMS: None
;
; GLOBAL VARS: mcp - MCP logical unit
;               pe.clr.scr - pendant control code, clear display & home cursor
;               pe.cur.pos - pendant control code, set cursor position
;               pe.beep - pendant control code, beep the pendant
;               pe.tab - pendant control code, tab to next soft button

; Clear the display and write the top line

        WRITE (mcp) $CHR(pe.clr.scr), $CHR(pe.cur.pos), $CHR(16), "MAIN MENU", /S

; Write the menu options

        WRITE (mcp) $CHR(pe.cur.pos), $CHR(41), /S
        WRITE (mcp) "Option5", $CHR(pe.tab), "Option4", $CHR(pe.tab), /S
        WRITE (mcp) "Option3", $CHR(pe.tab), "Option2", $CHR(pe.tab), "  Q
UIT", /S

; Beep the MCP

        WRITE (mcp) $CHR(pe.beep), /S

.END
```

```
.PROGRAM mcp.main.quit(quit)

; ABSTRACT: This program responds to a "Quit" selection from the MCP
;   main menu.  It verifies the selection and passes the result.
;
; INPUT PARMS: none
;
; OUTPUT PARMS: quit - Boolean indicating whether a "quit"
;   has been verified
;
; GLOBAL VARS mcp - The MCP logical unit
;   pe.clr.scr - pendant control code, clear display & home cursor
;   pe.off.led - pendant control code, turn off an LED
;   pe.blink.char - pendant control code, start blink position
;   pe.noblink.char - pendant control code, disable blink position
;   pe.tab - pendant control code, tab to next soft button
;
;
quit = FALSE ;assume quit will not be verified

; Display sub-menu and start the "NO" option blinking

WRITE (mcp) $CHR(pe.clr.scr), "Quit. Are you sure?"
WRITE (mcp) $CHR(pe.tab), $CHR(pe.tab), $CHR(pe.tab), " YES", /S
WRITE (mcp) $CHR(pe.tab), $CHR(pe.blink.char), " NO", $CHR(pe.nobl
ink.char), /S
button = PENDANT(0)

; Set quit to true if verified, else turn off the "NO" soft button LED

IF button == 2 THEN
    quit = TRUE
ELSE
    WRITE (mcp) $CHR(pe.off.led), $CHR(1)
END
.END
```


- A**
- ARM POWER/HIGH POWER, 4
- ATTACH, 41
- Auto start
 - from the MCP, 23
- AUTO.V2, 23
- B**
- Background mode, 15
- Binary I/O, see Digital I/O
- Button modes, 43
 - keyboard, 44
 - level, 45
 - toggle, 44
- C**
- Calibrating the robot from the MCP, 23
- Clear error button, 22
- Clear error function, 22
- CMD function, 23
- CMD1, 24
- CMD2, 24
- Communication
 - with the MCP, 41
- Comp/Pwr button, 28
- Compatibility, 2
- Control codes, 48
 - for LCD, 48
- CR-LF
 - suppressing, 42
- Customer Service Assistance
 - phone numbers, 6
- Cycle
 - setting, 25
- D**
- DETACH, 41
- Detecting user input, 43
- Device number, 40
- Digital I/O
 - displaying, 21
- E**
- Edit function, 18
- Editing global variables with the MCP, 18
- Emergency stop, 15
- Emergency stop buttons, 28
- Errors
 - displaying system, 21
- F**
- Free state, 38
- H**
- HIGH POWER/ARM POWER, 4
- I**
- IOSTAT(), 41
- J**
- Joint state, 35
 - with Cartesian robot, 35
 - with SCARA robot, 35
- Joint values
 - displaying, 20
- K**
- Keyboard mode, 44
- L**
- LCD
 - control codes, 48
- Level mode, 45
- Location
 - displaying current robot, 20
- Location variables
 - editing with MCP, 19
- Logical unit number
 - for MCP, 41
- M**
- Man/Halt button, 28
- Manual states
 - free, 38
 - joint, 35
 - tool, 32
 - world, 30
- MCP
 - determining state of, 47
 - viewing angle, 15
- MCP button map, 47
- MCP logical unit number, 41
- Mode control buttons, 27
- Monitor speed
 - setting, 26

P

- PENDANT(), 44
 - used to determine MCP state, 47
 - with speed potentiometer, 46
 - with toggle buttons, 44
- Potentiometer
 - programming, 46
- Precision point, 19
- Predefined function buttons, 17–26
- Priming a new program, 25
- Prog Set function, 25
- Program
 - starting execution, 26
- Program cycles
 - setting, 25
- Program step
 - selecting starting, 25

R

- READ, 43
- Real variables
 - editing with MCP, 18
- Robot location
 - displaying, 20

S

- Safety
 - overview, 3
- Selecting a new program, 25
- Selecting different robots, 40
- Slow button
 - programming, 46
- Speed
 - setting, 26
- Speed bar, 29
- Speed potentiometer
 - programming, 46
- Starting program execution, 26
- State of MCP
 - determining, 47
- Status
 - displaying system, 21
- Step
 - selecting starting, 25
- Store All function, 24
- Storing programs with the MCP, 24
- Suppressing CR-LF, 42
- System safeguards
 - computer controlled devices, 4

T

- Toggle mode, 44

- TOOL state
 - and TOOL transformation, 32
- Tool state, 32
- TOOL Transformation, 32

U

- User input
 - detecting, 43

V

- Viewing angle
 - LCD panel, 15

W

- WAIT.START, 52
- WRITE, 42

00101-00010 Rev A