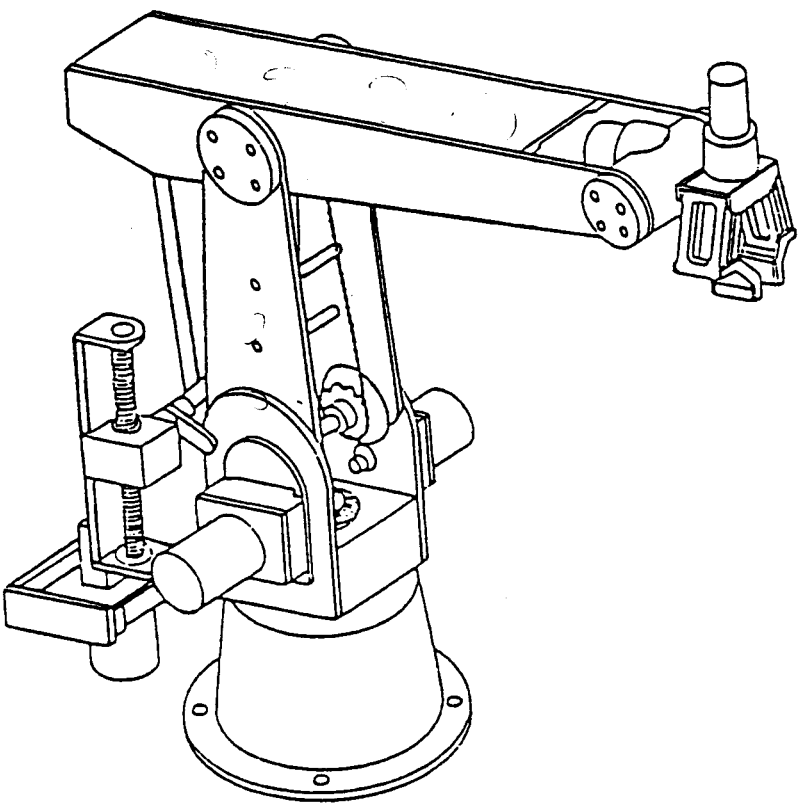
Prof. T. Sobh

SIR-1

USER MANUAL



Scien-Tech Intraco Automation (Pte) Ltd

TABLE OF CONTENTS

Page

CHAPTER 1

INTRODUCTION

SIR-1 is an intelligent multipurpose educational robot, designed and manufactured by SCIEN-TECH INTRACO AUTOMATION (PTE) LTD.

Generally it can be used for following purposes :-

- for teaching and training of robotics
- to simulate industrial robot operations for industrial automation planning
- for research and development of robotic hardware and software.

The basic system configuration of SIR-1 consists of a robot arm, a microprocessor-based controller and a hand-held teach pendant.

It can operate in either teach mode or host mode. In teach mode the user, without any formal programming knowledge, can use the hand-held teach pendant to operate the robot arm and record the arm motions. In host mode the user can program the robot arm to work on custom-designed software using a host computer.

This manual provides user with basic information on SIR-1 robot arm, controller and hand-held teach pendant as well as instructions to operate SIR-1. These instructions are categorized as follows :-

- Setting up and self-tests
- How to power up SIR-1
- How to program SIR-1 in the teach mode
- How to program SIR-1 in the host mode.

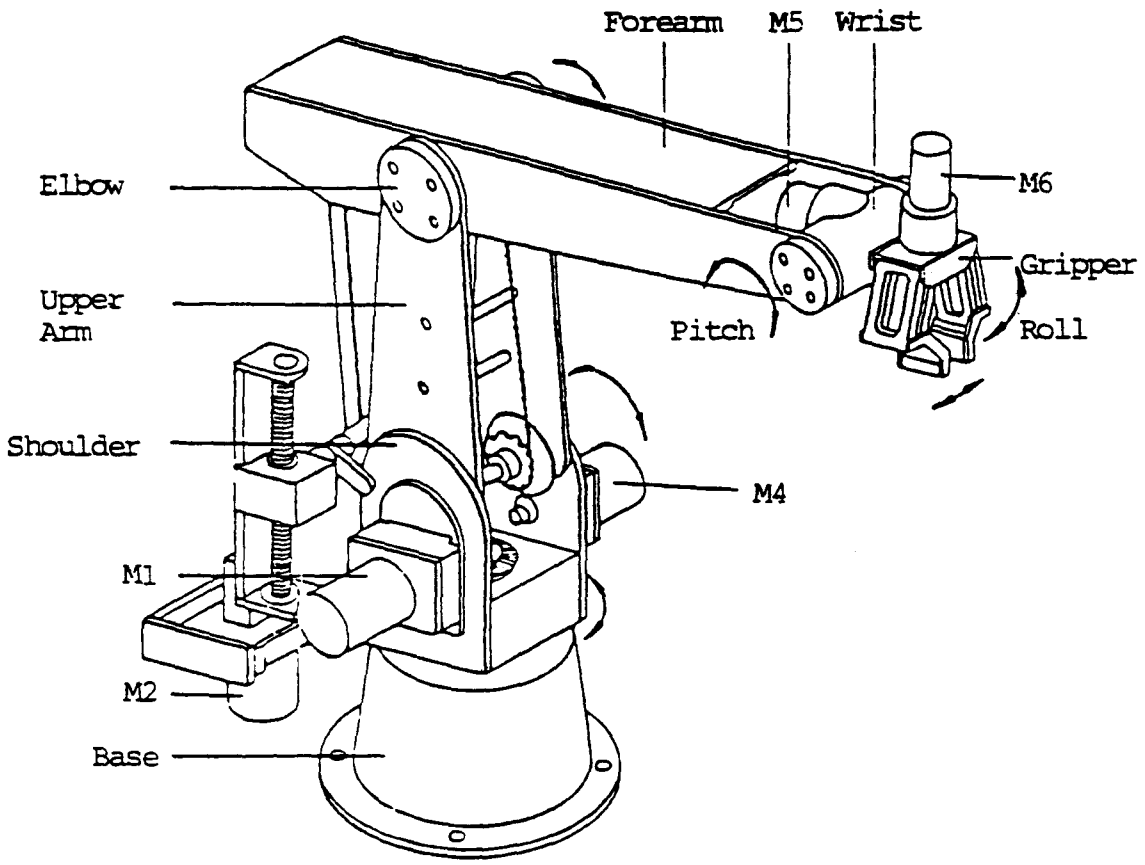
CHAPTER 2

BASIC INFORMATION ON SIR-1

2.1 SIR-1 ROBOT ARM

The articulated SIR-1 robot arm is designed with an open structure so that the user can see all its operating parts.

Figure 2.1 shows the various parts of the SIR-1 robot arm.



M1 - base motor

M2 - shoulder motor

M3 - elbow motor (hidden)

M4 - wrist pitch motor

M5 - wrist roll motor

M6 - gripper open/close motor

Figure 2.1 SIR-1 Robot Arm

2.1.1 Driving Source and Control Mechanism

Gearmotors

The driving source of the robot arm and its grippers come from the six DC gearmotors (M1 to M6) equipped with optical encoders.

The swivel operation of the base is achieved by a 100:1 gearmotor M1 through a gear with a 3:1 gear ratio to increase torque.

In order to acquire accurate and stable positioning capability, both the upperarm and forearm are driven by ball screws activated by the motors M2 & M3 each with a geardown-ratio of 20:1.

The wrist pitch motion is controlled by the motor M4 with gear down ratio of 100:1. Between the wrist and the motor M4 are two roller chains and gears giving a reduction ratio of 4:1.

The wrist roll is controlled by the motor M5 with gear down ratio of 200:1.

The opening and closing of the grippers are activated by motor M6 with a gear down ratio of 200:1 and through a screw/nut mechanism.

Photo sensors and Limit switches

The robot arm has a total number of 11 end-of-travel limit (software limit). The control program will stop the respective motor and bring it back to the safe range, when an end-of-travel limit is being detected.

The robot arm is also equiped with 6 limit switches which act as hardware limits. These are placed just after the software limit and are used to cut off the respective motor power when software limits are exceeded.

The hardware and software limits provide double protection to the mechanical arm.

Table 2.1 gives the number of photo sensors and limit switches at each joint of the robot arm.

Joint	Software Limits	Hardware Limits
Base	1 photo sensor	1 limit switch
Shoulder	2 limit switches	2 limit switches
Elbow	2 limit switches	2 limit switches
Wrist Pitch	1 photo sensor	1 limit switches
Wrist Roll	1 photo sensor	-
Grippers	1 limit switch	-

Table 2.1 Software and Hardware Limits

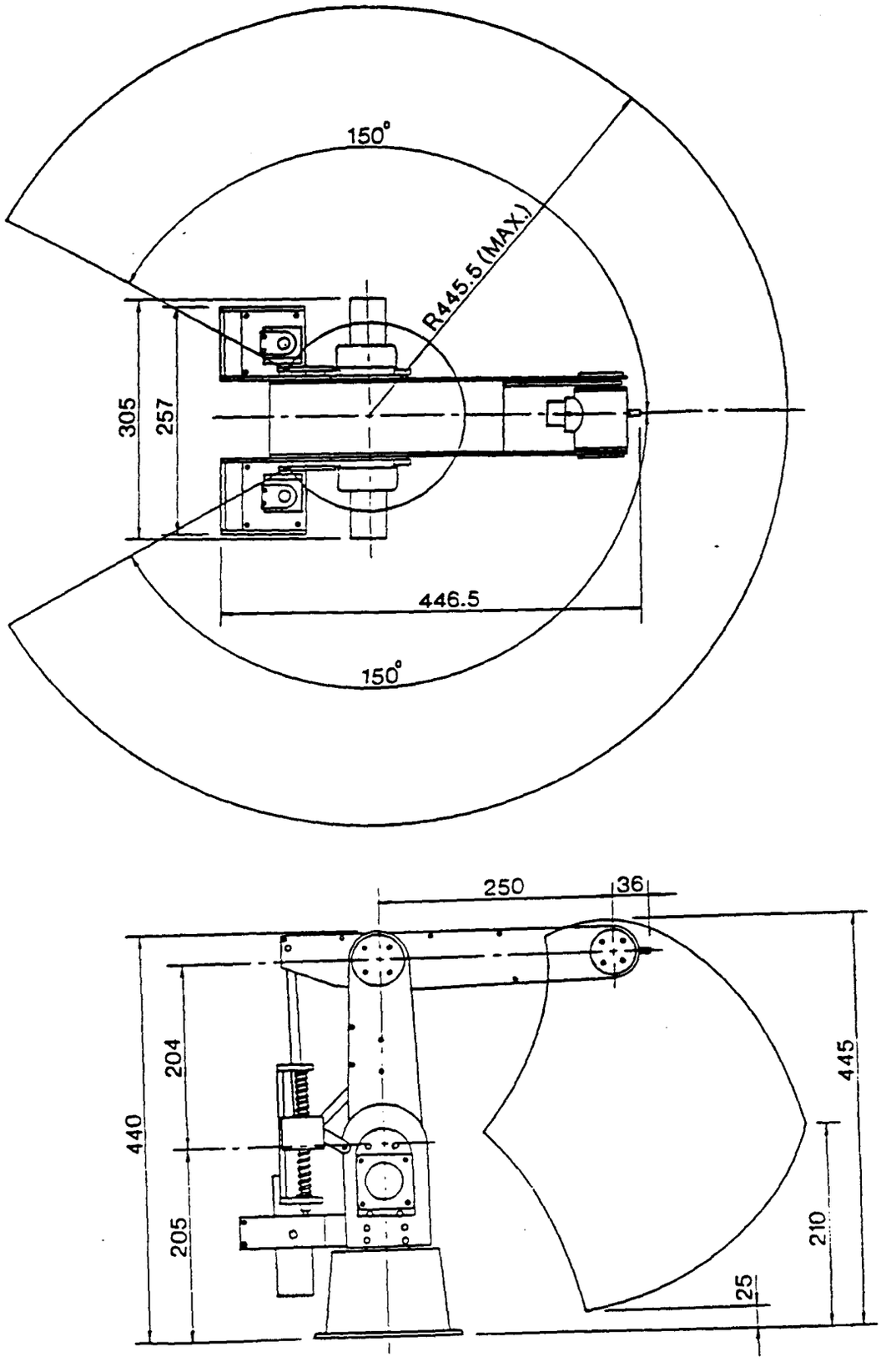
1.2 Specifications of SIR-1 Robot Arm

e specifications of the SIR-1 robot arm are shown in Table 2.2 below

Description		Specifications		
Structure		Articulated		
Degree of Freedom		5 axes		
Drive System		DC servo-gearmotor with incremental digital shaft encoders		
Working Range/ Max. Speed/ resolution	Base	300°	60°/sec	0.24°/step
	Upper Arm	+20°, -110°	9°/sec	0.03°/step
	Forearm	+5°, -85°	6.5°/sec	0.02°/step
	Wrist Pitch	+90°, -90°	45°/sec	0.18°/step
	Wrist Roll	350°	165°/sec	0.9°/step
	Grippers Opening	60mm fully opened	12mm/sec	0.13 mm/step
Combined Max. Speed		460 mm/sec at finger tip		
Reach Range		See Figure 2.2		
Load Capacity		5 Kg (without grippers)		
Repeatability		+/- 0.6 mm		
Ambient Temperature		0°C to 40°C		
Accessories		X-Y table, Linear slide base, rotary table, conveyor, triple finger gripper, magnectic hand et		

Table 2.2 SIR-1 Robot Arm Specification

2.1.3 SIR-1 Working Envelope



All dimensions in mm.

Figure 2.2 SIR-1 WORKING ENVELOPE

2.2 SIR-1 CONTROLLER

The SIR-1 controller is a robot control system which is microprocessor-based to provide programmable and teaching playback functions.

It interprets the commands which the user gives to the arm through the teach-pendant or a host computer.

It permits simultaneous control of 8 DC servo-motors - 6 on the robot arm and 2 on any accessory or accessories such as X-Y table, rotary table, and conveyor etc. It is equipped with many other powerful features such as I/O ports for synchronisation with surrounding equipment and peripherals, cassette interface for mass storage of teaching sequences and data, 8-step speed control for multi-speed setting, user defined home position and self diagnosis.

2.2.1 Switches and Indicators on Control Panel

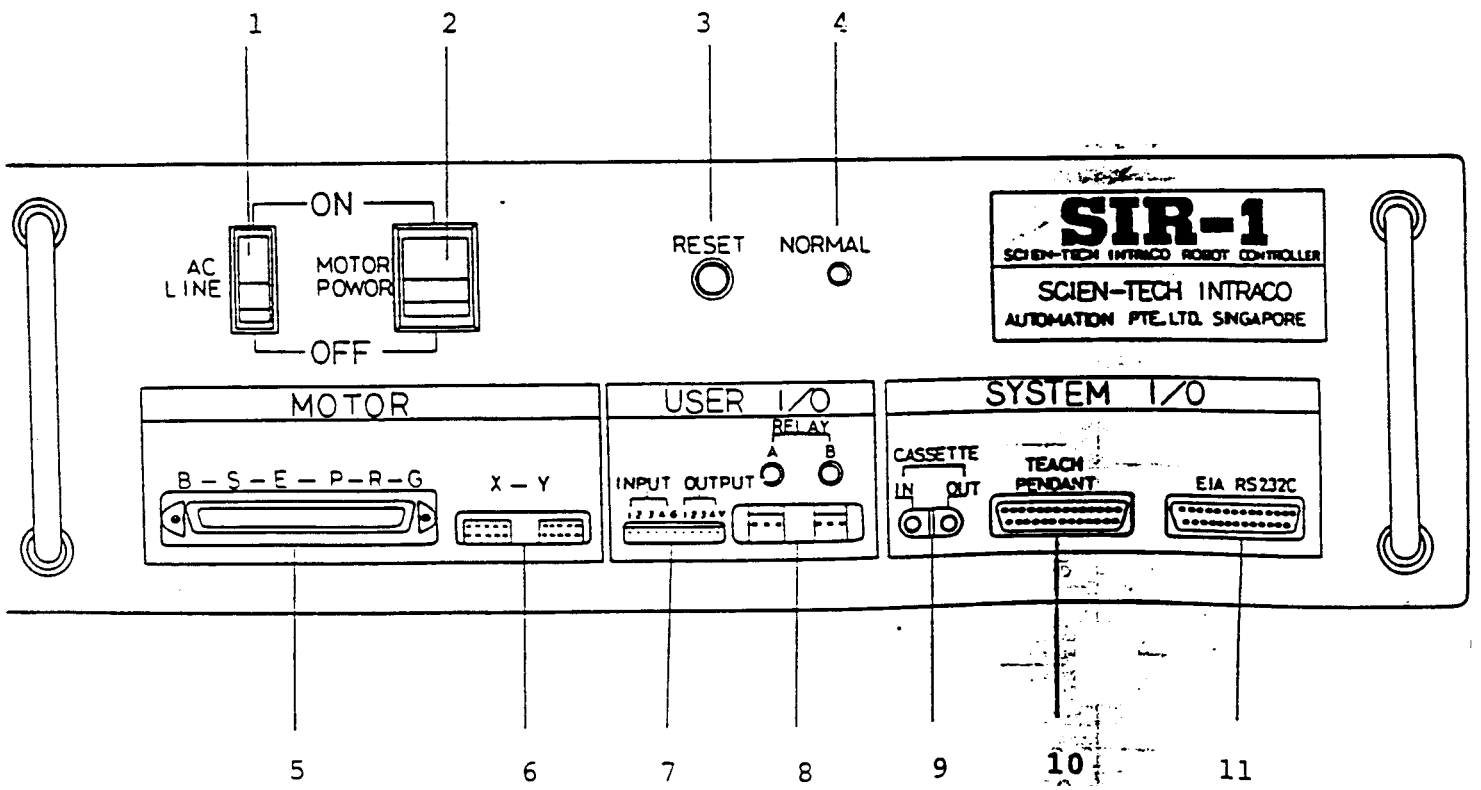


Figure 2.3 Switches and Indicators on Control Panel

CONTROL OR INDICATOR	FUNCTION
1. AC Line Switch	To turn on and off the AC power supply.
2. Motor Power Switch	To switch on and off the power supply to the motors on robot arm and accessories.
3. Reset Switch	To reset the controller.
4. Normal LED	This indicator blinks continuously after power-up if the controller is in normal condition.
5. Robot Arm Port	For connection to the robot arm.
6. Accessory Port	For connection to the accessories up to two axes to operate simultaneously with the robot arm.
7. Input/Output Ports	Four input bits and four output bits are provided, for synchronisation with the surrounding peripherals.
8. Relays Ports and LEDs	Two relay ports with LED indicators, for synchronisation with the surrounding peripherals. The LED will illuminate when the respective relay is turned on.
9. Cassette Interface Port (Optional)	For connection to cassette recorder for recording the teach sequences and data.
10. Teach Pendant Port	For connection to the hand-held teach pendant.
11. RS-232C Serial Interface Port	For connection to any computer with standard RS-232C serial interface for host mode operation.

Table 2.3 Functions of the Switches and Indicators

2.2.2 Specifications of SIR-1 controller

The Specifications of the SIR-1 controller are shown in Table 2.4.

ITEM		DESCRIPTION	
Structure/ Installation		Stand-alone, with cable to robot arm	
Number of Control Axes		Simultaneous 8-axes controls; 6 axes for robot; 2 axes for peripherals	
Memory for Teach Data		4K bytes standard (218 steps min) 32K bytes max (1904 steps min)	
Operating Mode		Teach mode and host mode	
Position Control		Incremental digital position control through optical encoder on each axis	
Speed Control		8 steps	
System Input/Output		Teach pendant, EIA - RS 232C serial interface Cassette interface (Optional)	
User Input/ Output	Digit I/O	4 inputs & 4 outputs	
	Relay	two relays	
Self-Diagnosis Capability		Built-in Self-test Program; Software limit, hardware limit and built-in system alarm	
Power Supply		110/220 VAC +/-10%; 60/50 Hz; 200W	
Weight		6.5Kg (approx.)	
Operating Conditions		Temperature	0° to 45°C
		Humidity	RH 90% max.; non-condensing

Table 2.4 SIR-1 Controller Specifications

2.3 SIR-1 TEACH PENDANT

The SIR-1 Teach Pendant enables the user without formal programming knowledge to program and manipulate the robot arm.

Figure 2.4 shows the control panel of SIR-1 teach pendant.

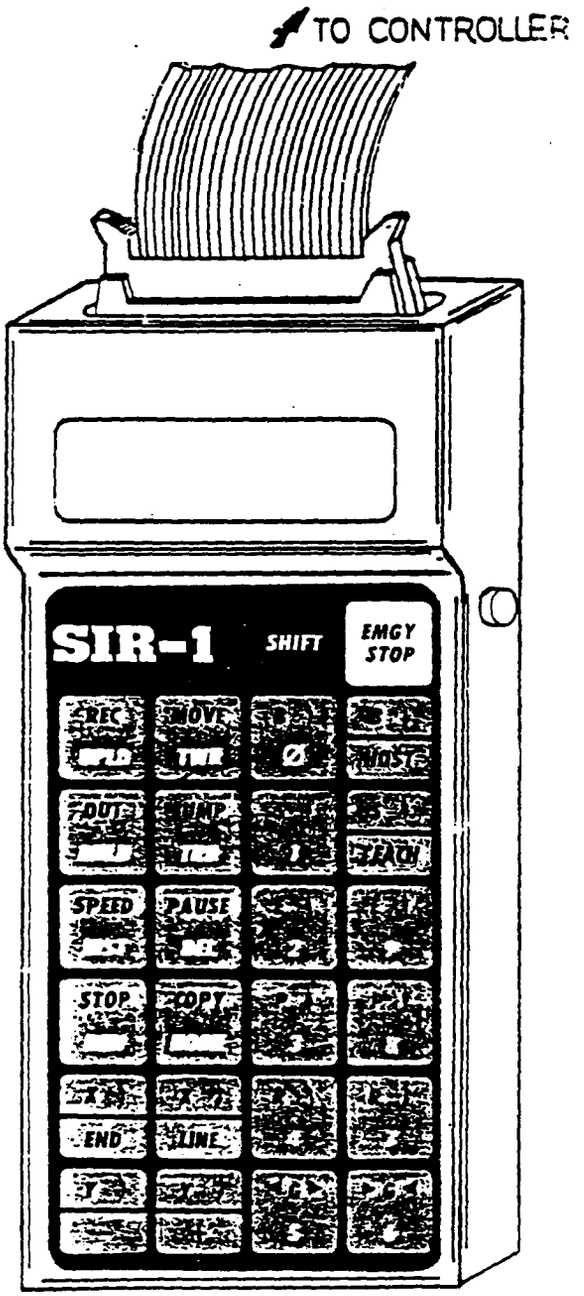


Figure 2.4 SIR-1 Teach Pendant

Function keys

It has 26 keys to perform 22 command functions, 16 joint-moving functions and 10 numeric functions.

For easy operation of the teach pendant, different function keys are identified by different colours.

Command keys

The main command keys are in grey colour.

To activate a command in 'white' letters, the SHIFT key, which is the red button at the right hand side of the teach pendant, must be pressed simultaneously with the command key. The SHIFT indicator will illuminate when the SHIFT key is pressed.

To activate a command in 'black' letters just press the command key only.

Parameter keys

The parameter keys are blue in colour. The joint keys and numeric keys fall under this category. These keys can only be activated after pressing a command key.

CHAPTER 3

SETTING UP AND SELF-TESTS

This chapter describes the procedures to connect the teach pendant and robot arm to the controller as well as to initiate the self-test operation of the controller and the motor/encoder assemblies on the robot arm. For installation of gripper, please refer to Appendix A.

3.1 Connect the teach pendant to the controller

3.1.1 Check to see all switches on the front panel of the controller are at the OFF position.

3.1.2 Connect the teach pendant to the controller using the flat cable.

3.1.3 Check the voltage of the AC power source before plugging in the AC power cord of the controller.

> **IMPORTANT NOTE**
>
> SIR-1 can work on either 110 VAC or 220 VAC by setting a jumper on the power supply board in the controller. Failure to follow the jumper setting may cause permanent damage to both the controller and the power supply.

3.2 Controller Self-test

3.2.1 When the [AC LINE] is turned on, the controller will execute a self-test. If the connections are alright and the controller is in good working condition :

- 3 short 'beep' sounds can be heard.
- AC LINE indicator will illuminate and the NORMAL indicator will flash continuously on the front panel of the controller.
- The message 'IDLE' will appear simultaneously on the teach pendant.

3.2.2 If any fault is detected in the controller during the self-test, the operation will stop and the alarm will sound with an error message appears on the teach pendant.

3.3 Connect the robot arm to the controller

3.3.1 With AC LINE and MOTOR POWER at the OFF position, connect the 50-conductor cable from the robot arm to the controller.

3.3.2 Double-check to ensure that the cable is in place and locked at both ends.

> **IMPORTANT NOTE**
>
> Loose connection on either end of the connector may cause damage to the controller circuit.

3.4 Test the motor and encoder assemblies on the robot arm

3.4.1 Turn on [MOTOR POWER].

3.4.2 Press [SHIFT] on the teach pendant and turn on [AC LINE] simultaneously.

This activates all the active joint motors on the robot arm and accessory ports to perform a short back-and-forth movement.

The name(s) of the active joint(s) will be displayed on the teach pendant leaving the name(s) of the inactive joint(s) blank.

Figure 3.1 below shows the teach pendant display of the various joints when all the joint motors are active.

base shoulder elbow pitch roll gripper

Fig 3.1 Teach Pendant Display

The two dots on the fifth and sixth digit from the left on the display are for motors of accessary port X and Y respectively.

If all the joint motors are inactive, the teach pendant will display 'Err F' and then 'SIR-1'.

CHAPTER 4

POWER-UP

This chapter describes 2 ways to power-up SIR-1 to enter the mode-selection state.

4.1 How to power up SIR-1

Method 1 : Power-up with home-positioning

1.1 Turn on [MOTOR POWER] and then [AC LINE].

If at least 4 big joint motors (B,S,E,P) are active, SIR-1 robot arm will perform a home-positioning sequence and stop at its default home position.

1.2 After home positioning, the active joints' names appear on the teach pendant leaving the inactive ones blank and SIR-1 enters the mode-selection state.

1.3 If all the joint motors are inactive, the teach pendant will display the message 'IDLE' and the alarm will sound in 3-beep sequence.

1.4 If 1.3 happens, power-down SIR-1 by turning off [MOTOR POWER] and [AC LINE].

Method 2 : Power-up for direct entry to mode-selection state

2.1 Turn on [AC LINE] and then [MOTOR POWER]. Teach pendant will display 'IDLE'.

2.2 Press [SHIFT] and [EMGY STOP] simultaneously. A message 'SIR-1' will display on the teach pendant and SIR-1 is in the mode-selection state.

4.2 Mode-selection state

4.2.1 When in mode-selection state, the user can select to operate the robot arm under the teach mode using the teach pendant or the host mode using the host computer.

4.2.2 Chapter 6 to chapter 16 give details of programming using teach pendant. Chapter 17 to chapter 19 give details of programming using the host computer.

CHAPTER 5

RESET AND MOTOR CHECK

This chapter describes how to reset SIR-1 and how to do a motor check.

5.1 Reset

5.1.1 To reset, press [RESET] on the controller or press [SHIFT] and [EMGY] keys on the teach pendant and release [SHIFT] key first.

5.1.2 This causes SIR-1 to abort what it is doing, however the whole user program memory is not affected.

5.2 Motor Check

5.2.1 To do a motor check on all eight motors, press [SHIFT] and [EMGY] and release [EMGY] key first.

5.2.2 The effect is same as resetting, in addition it does a check on all motor. After the check, active joint motors' names will display on the teach pendant leaving the inactive ones blank. SIR-1 is in the mode-selection state.

PROGRAMMING FROM THE TEACH PENDANT

CHAPTER 6

HOW TO ENTER TEACH MODE

This chapter describes how to enter the teach mode operation for programming using the teach pendant.

6.1 How to enter Teach Mode

After SIR-1 has completed its power up initialization sequence as described in chapter 4, SIR-1 enters the mode-selection state.

To go into teach-mode operation press [TEACH] on the teach pendant and "- 0000" will appear on the display and SIR-1 is in the command entry state. The user may proceed to create a teach program.

If you have not done the home positioning or motor action check after power up, press [SHIFT] [TEACH] to enter teach mode. By doing so SIR-1 assumes that all motor are functional.

Before creating any teach programs, the user has to understand the various teach commands and parameters used.

6.2 Teach command keys

Teach commands can be divided into two classes :-

Immediate commands and deferred commands.

Immediate command is a command that execute its function immediately upon pressing its key while a deferred command will be activated only during the execution of the program.

However [HOME] and [MOVE] are both immediate and deferred commands.

Table 6.1 below shows the immediate and deferred command keys.

Immediate command keys	Deferred command keys	Immediate and Deferred command keys
[HOST] [TEACH] [REC] [STOP] [COPY] [END] [LINE] [-] [+] [RUN] [INST] [DEL] [UPLD] [DNLD] [TWR] [TRD]	[JUMP] [PAUSE] [SPEED] [OUT]	[HOME] [MOVE]

Table 6.1 SIR-1 Command Keys

6.3 Parameter keys

Joint control keys and numeric entry keys are parameter keys whose functions are governed by the preceeding command key(s). They cannot be executed without first executing a command key.

Joint-control keys

Table 6.2 shows the joint control keys and their functions.

Joint Control key	Function
B ↑	Base to move in anti-clockwise direction
B ↓	Base to move in clockwise direction
S ↑	Upperarm to move upwards
S ↓	Upperarm to move downwards
E ↑	Forearm to move upwards
E ↓	Forearm to move downwards
P ↑	Grippers to pitch up
P ↓	Grippers to pitch down
R ↑	Grippers to roll in anti-clockwise direction
R ↓	Grippers to roll in clockwise direction
< G >	Grippers to open
> G <	Grippers to close
X ↑	X port motor to move in anti-clockwise direction
X ↓	X port motor to move in clockwise direction
Y ↑	Y port motor to move in anti-clockwise direction
Y ↓	Y port motor to move in clockwise direction

Table 6.2 SIR-1 Joint Control Keys

Numeric entry keys :

There are 10 numeric entry keys from 0 to 9.

CHAPTER 7

BASIC TEACH COMMANDS

This chapter allows the user to give a try at creating a simple teach program using some basic commands such as LINE, MOVE, HOME, REC and END.

7.1 Create a simple teach program 1

7.1.1 Below are the procedures to create a simple teach program for a move sequence

1. Power up SIR-1 as explained in chapter 5 so that SIR-1 does a home-positioning.

2. After the active joints' names display on the teach pendant press [TEACH] to enter the command entry state and "- 0000" appears on the teach pendant. Now can start creating the program as follows :-

Key Entry	Display
- Press [0] [0] [0] [0]	"- 0000"
- Press [MOVE]	"MOVE"
- Press [R↓], hold it down until display shows "R 0050"	"R 0050"
- Press [E↓], hold it down until display shows "E 0100"	"E 0100"
- Press [REC]	"- 0001"
- Press [END]	"END"
- Press [REC]	"- 0002"

3. Before running the program, drive SIR-1 back to its home position by pressing [SHIFT] [HOME] and then [LINE].

4. To run the program, press the starting line number [0] [0] [0] [0] followed by [SHIFT] [RUN].

The sequence just created is to tell SIR-1 to rotate its grippers 50 steps and lower its forearm 100 steps and then stop.

To drive SIR-1 back to its original or home position, the user has to press [SHIFT] [HOME] and then [LINE].

If the user wants the robot arm to return to its home position at the end of the move sequence, the home command should be recorded as a program step as follows :-

Key Entry	Display
- Press [0] [0] [0] [0]	"- 0000"
- Press [MOVE]	"MOVE"
- Press [R↑], hold it down until display shows "R 0050"	"R 0050"
- Press [E↓], hold it down until display shows "E 0100"	"E 0100"
- Press [REC]	"- 0001"
- Press [SHIFT] [HOME]	"HOME"
- Press [REC]	"- 0002"
- Press [END]	"END"
- Press [REC]	"- 0003"

The functions of the commands used in the above teach program are explained below :-

7.2 LINE command - [LINE]

The [LINE] key is used to abort a current command (except MOVE command) so that SIR-1 returns to command entry state and display present line number on the teach pendant.

7.2.1 To enter a line number

Every program step must begin with a 4-digit line number. It is advisable to start your program at line 0000 although you may start at other line.

To enter a line number, just key in a 4-digit number if it is in command entry state, ie. teach pendant displays a 4-digit number. If not press [LINE] first to abort the current command to followed by a 4-digit number.

For example :-

Key Entry	Display
[0] [0] [0] [0]	"- 0000"

Correcting a wrong line number

If a wrong line number has been entered, to correct it, just press the correct number. For example :-

Line number "- 0023" has been entered as "- 0024". To correct it just press [0] [0] [2] [3] and the correct number "- 0023" will replace the wrong one and appears on the display.

Key Entry	Display
[0] [0] [2] [4]	"- 0024" wrong number
[0] [0] [2] [3]	"- 0023" correct number

7.2.2 To abort a current command Input

If the user wishes to abort any current command other than MOVE command, just press [LINE] and SIR-1 returns to the command entry state.

To abort MOVE command during teaching, press [SHIFT] [TEACH].

7.3 MOVE command - [MOVE]

The [MOVE] key is used for moving the joints. [MOVE] must be activated first before moving any joint-control key(s).

For example :-

Key Entry	Display
Press [MOVE]	"MOVE"
Press [R↓], hold it down until display shows "R 0050"	"R 0050"
Press [E↓], hold it down until display shows "E 0100"	"E 0100"

You can activate all eight joints one at a time under MOVE command.

Bi-directional

Each joint can move in two directions :

- up or down for shoulder, elbow and wrist pitch
- anticlockwise or clockwise for base and wrist roll
- open or close for gripper.

To distinguish between these two directions on the display, for up, anticlockwise and open movements a "dot" will appear between the joint name and the step count.

For example :-

To move the shoulder up 50 steps.

Key Entry	Display
Press [MOVE]	"MOVE"
Press [S↑], hold it down until display shows	"S.0050"

- for down, clockwise and close movement, there is no dot between the joint name and step count.

For example :

To move the shoulder down 50 steps.

Key Entry	Display
Press [MOVE]	"MOVE"
Press [S↓], hold it until display shows	"S 0050"

Step-count

Step-count is the number to the right of the joint-name as seen in the example above. It indicates the number of steps the particular joint has moved.

Based on the above example, when [S↓] is pressed continuously, the shoulder will move continuously and the step count will keep increasing from "S 0000" to "S 0001" to "S 0002" to "S 0003" and so on.

To move shoulder 50 steps, release [S↓] key when the display shows "S 0050".

If you press the key momemtarily, the step count will increase by one. If the number overshoot, you can press [S↑] to bring it back.

7.4 HOME command - [HOME]

The [HOME] key is used to drive SIR-1 to its home position. It can be done in 3 ways.

7.4.1 Direct Entry to home position

When SIR-1 is in the command entry state, to drive SIR-1 to its home position, press [SHIFT] [HOME].

When the home position is reached the message "HOME" will display on the teach pendant. The user can record this command as a program step by pressing [REC]. If not, press [LINE] to abort the command and a line number will appear.

7.4.2 Home SIR-1 under MOVE command

If SIR-1 is under MOVE command, SIR-1 can be driven to its home position by pressing [SHIFT] [HOME].

When SIR-1 reaches its home position, the message "MOVE" will display on the teach pendant and it is still under MOVE command.

7.4.3 Re-define home position

After power-up, the robot arm is at its default home position which can be re-defined as follows :-

When SIR-1 is in the command entry state, press [MOVE] followed by the relevant joint-control key(s) to move SIR-1 to its desired position.

Then simply press [HOME] only and this position becomes the new home position of SIR-1. (Note : Do not press [SHIFT] [HOME] because this will cause SIR-1 to move to its default-home position)

7.5 RECORD command - [REC]

7.5.1 Record a program step

This [REC] key is used to record a program step in a teach program.

Example of a program step :-

Key Entry	Display
Press [0] [0] [0] [0]	"- 0000"
Press [MOVE]	"MOVE"
Press [R↑], hold it down until the display shows "R 0050"	"R 0050"
Press [E↓], hold it down until the display shows "E 0100"	"E 0100"

To record this program step press [REC] and the next line number "- 0001" appears.

Key Entry	Display
[REC]	"- 0001"

If a program step is previously entered on this line, it will be overwritten by the new program step.

CHAPTER 8

SPEED COMMAND - [SPEED]

This chapter explains how the SPEED command is used in a teach program.

SIR-1 has a 8-step speed control from 0 to 7, 0 being the slowest and 7 being the fastest.

The speed command allows the user to set the speed of the robot arm motion.

8.1 Create a simple teach program 2

Following are the procedures to create a teach program to set the robot arm to run at 2 different speeds.

Key Entry	Display
- Press [0] [0] [0] [0]	"- 0000"
- Press [SPEED] [7] [REC]	"- 0001"
- Press [MOVE]	"MOVE"
- Press [R⟩], hold it down until display shows "R 0100"	"R 0100"
- Press [REC]	"- 0002"
- Press [SPEED] [3] [REC]	"- 0003"
- Press [MOVE]	"MOVE"
- Press [R⟩], hold it down until display shows "R.0100"	"R.0100"
- Press [REC]	"- 0004"
- Press [END]	"END"
- Press [REC]	"- 0005"

To run the program, press the starting line number [0] [0] [0] [0] and then press [SHIFT] [RUN].

The grippers rotate 100 steps in the clockwise direction and back 100 steps in the anti-clockwise direction, then stop at its original position.

It can be noticed that the grippers move faster in the clockwise direction than in the counter-clockwise.

This is because the speed of the clockwise direction is set at the fastest, that is, 7 while that of the anticlockwise direction is set at a medium speed of 3.

CHAPTER 9

EXAMINE AND EDIT THE COMMANDS

After a program has been created and recorded, it is useful to check through the program line by line and amend any error(s) before running the program.

This chapter describes how to examine and modify teach sequences using the following commands :-

[+], [-], [INST], [DEL] and [COPY].

Before starting the process, user may wish to refer to Appendix C which shows the 7-segment interpretation of the digits and alphabets.

9.1 Examine program steps using [+]

First set the program at its starting line by entering the starting line number.

To start the checking process, press [+] key and the first program step will appear on the display. If the program step is too long (eg. [MOVE] and [JUMP]) and it is not possible to display it on the LED at one time, it is necessary to press [+] a few times to examine the complete step.

9.2 Amend error in parameter using [SHIFT] [+]

During the examination, if any error is detected in the parameter (eg. speed number, step count, jump condition code and output condition code etc), it can be amended by simply entering the new number or press [-] to change the direction of joints motion and then press [SHIFT] [+]. The new number will be entered and the checking process can continue.

9.3 Modify program steps using editing commands : [INST], [DEL] and [COPY]

To modify or correct a program that has been created, a set of editing commands can be used.

As SIR-1 remembers the absolute positions of the points being taught, deletion and insertion of a program step does not affect other move sequences. This make the editing easy and powerful.

Following are the editing commands :

9.3.1 INSERT command - [INST]

[INST] key is used to insert a program step.

To insert a program step, first set the line number which the step will be inserted on the display. Enter the desired command and press [SHIFT] [INST]. (Do not press [REC], otherwise the existing program step will be overwritten).

The step will be inserted into the program and all the subsequent program steps will be shifted downward one line in the memory and the line numbers are changed accordingly.

The next line number after the inserted line will appear on the display.

9.3.2 DELETE command - [DEL]

[DEL] key is used to delete a program step.

To delete a program step, set its line number on the display and press [SHIFT] [DEL].

The step is deleted and all the subsequent program steps will move one line upwards in the memory and their line numbers will be changed accordingly.

9.3.3 COPY command - [COPY]

[COPY] key is used to duplicate an existing program step in the program. It is especially useful for long program steps which are repeated several times in a program.

To do this, first set the line number of the step to be duplicated then press [COPY] and the display will show "C XXXX", where XXXX is irrelevant.

Now input the line number to copy to and press [REC].

If a non-existent line is copied, error message "Err 5" will be displayed and SIR-1 returns to command entry state.

CHAPTER 10

SINGLE-STEPPING A PROGRAM

Before running a program that has been created and recorded, it is advisable to move the arm through the program one line at a time. This process is called program single-stepping ---- one of the most important techniques to debug computer programs.

This chapter describes the procedures to single-step a program as follows:

1. To single-step a program, first set the starting line number and press [SHIFT] [LINE], the first sequence will be executed.

2. On completion of the first sequence the line number of the next sequence to be executed is displayed. By pressing [SHIFT] [LINE] again, the sequence will be executed and the process continues.

3. In this way the user can step through the whole program or some critical parts of the program to make sure that SIR-1 is doing the right sequences.

4. During the process, necessary correction can be made.

CHAPTER 11

RUN A PROGRAM AND STOP A PROGRAM

This chapter describes the procedures to run a program and to stop a program.

11.1 Procedures to run a program

1. To run a program, first enter the starting line number, then press [SHIFT] [RUN].

2. As soon as [SHIFT] [RUN] are pressed, the display on the teach pendant becomes blank and the arm begins to operate as taught.

3. If there is any syntax error in the program or controller detects any error condition, an error code ("Err x") will flash on the display and the alarm will beep three times.

   However, the program execution may or may not abort depending on the situation. Refer to Appendix E for error code interpretation and various remedial actions.

11.2 Procedures to stop a program

1. To stop a program when it is run halfway, simply press [STOP].

2. As soon as [STOP] is pressed, the system will continue the program step it is executing and stop the program upon completion. The teach pendant will show the next line number.

CHAPTER 12

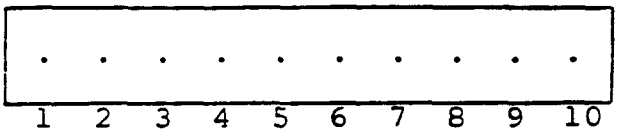
SYNCHRONISATION WITH SURROUNDING EQUIPMENT AND PERIPHERALS

12.1 Input/Output Port and Relay Ports

The microprocess-based controller is equipped with 8 programmable input/output ports and 2 relay ports for synchronising with surrounding equipment and peripherals. The chapter explains how this can be done with a program using jump and out commands.

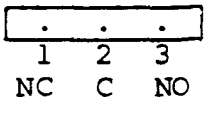
Figure 12.1 shows the pin configuration of various ports.

Input/Output Port



Pin 1	Input Bit 1	All inputs are TTL Compatible
2	Input Bit 2	
3	Input Bit 3	
4	Input Bit 4	
5	Ground	
6	Output Bit 1	TTL : Open Collector VOH : 28V IOL : 40 mA at VOL < 0.5V
7	Output Bit 2	
8	Output Bit 3	
9	Output Bit 4	
10	+5V	

RELAY PORTS A AND B



Contact Specification : 3 Amp. 220 Volt

NC : Normally closed

NO : Normally opened

Relay On : C and NO short circuit
C and NC open circuit

Figure 12.1 Pin Out Configuration of Various Ports

12.2 JUMP command

The JUMP command allows for conditional branching in a program. Hence the jump commands are used in a program to test the user input port's bits which are dependent on the input pattern from the surrounding peripheral/equipment.

12.2.1 Procedures to activate JUMP command

Below shows how jump command is activated :-

To enter the jump command, press [JUMP] and the display shows "J XXXX", enter the line number to jump and then press [REC].

This time the display shows "0 XXXX" awaiting the input of the jump condition code. Table 2.1 shows the jump condition codes.

12.2.2 JUMP conditions codes

Table 12.1 shows the jump condition codes

Jump Condition Code	Input Port Bit			
	4	3	2	1
0	L	L	L	L
1	L	L	L	H
2	L	L	H	L
3	L	L	H	H
4	L	H	L	L
5	L	H	L	H
6	L	H	H	L
7	L	H	H	H
8	H	L	L	L
9	H	L	L	H
10	H	L	H	L
11	H	L	H	H
12	H	H	L	L
13	H	H	L	H
14	H	H	H	L
15	H	H	H	H
16 (for unconditional jump)	X	X	X	X

All input bits are TTL compatible

H means logic high

L means logic low

X means don't care.

Table 12.1 Jump Condition Codes

12.2.3 How to calculate the jump condition code

The place value of the input bits are shown below :

Input bit	4	3	2	1
Place value	8	4	2	1

The jump condition code is the sum of the place value of those input bits which are set high.

For example, jump condition code 14 is the sum of the place value of 8, 4 and 2 of the input bits 2, 3 and 4 respectively which are set at high. The value of input bit set at low is 0.

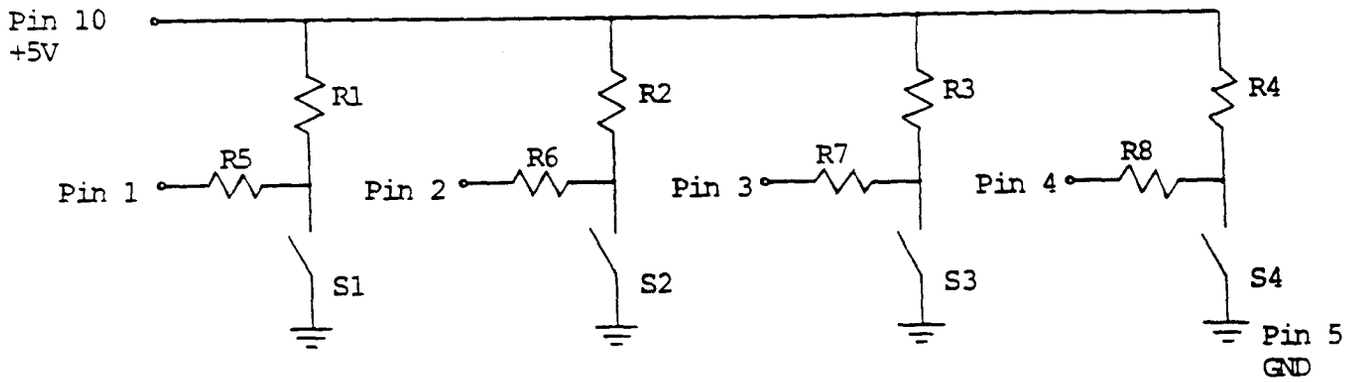
12.2.4 A simple program to illustrate the JUMP command

Below is a single program to illustrate how to use the jump command.

Key Entry	Display
- Press [LINE] [0] [0] [0] [0]	"- 0000"
- Press [MOVE]	"MOVE"
- Press [B⟩], hold it down until the display shows "B.0100"	"B.0100"
- Press [REC]	"- 0001"
- Press [SHIFT] [HOME]	"HOME"
- Press [REC]	"- 0002"
- Press [JUMP]	"J XXXX"
- Press [0] [0] [0] [0]	"J 0000"
- Press [REC]	"O XXXX"
- Press [1] [4]	"O 0014"

- Press [REC]	"- 0003"
- Press [JUMP]	"J XXXX"
- Press [0] [0] [0] [2]	"J 0002"
- Press [REC]	"O XXXX"
- Press [1] [6]	"O 0016"
- Press [REC]	"- 0004"
- Press [END]	"END"
- Press [REC]	"- 0005"

Before trying out the program, connect a simple circuit to the input port on the controller as shown in Fig. 12.1.



R1 to R4 : 1K 1/8W
R5 to R8 : 330 1/8W

Figure 12.1 Circuit for testing input port

For jump condition code 14 the input bit 1 is set low and the other input bits 2, 3 and 4 are set high. Set the input pattern on the circuit so that S1 closes and switches S2, S3 and S4 open.

When the program is run SIR-1 will move its base 100 steps in the anti-clockwise direction and returns to its home position. This move sequence will be repeated over and over again because the jump condition is met.

Now if the input pattern is changed, for example, switch S1 is opened and the other switches are closed, the sequence will stop because the jump condition is not met. The program will go into a loop and keep on checking the jump condition.

12.2.5 How editing affects JUMP command

When a program step is inserted or deleted before the destination line number of a JUMP command, the destination line number of the JUMP command is increased by one or decreased by one accordingly. If the program step is inserted or deleted at the destination line number itself, it is remain unchanged. This done automatically to ensure that the program jump to the correct line number after the editing.

12.3 OUT command

The out command is used for programming 4 TTL level output bits and 2 relays. They can be programmed ON or OFF through the out command.

12.3.1 Procedures to activate OUT command

Below shows how to activate the out command:-

While in teach mode, press [OUT] and the display will show "O XXXX", awaiting for the input of the output condition code. The output pattern from the output bits and relay ports depend on this condition code.

12.3.2 Output Condition Codes

Tables 12.2 shows the output condition codes

k value	relay A	relay B	output bit 4	3	2	1
0	OFF	OFF	0	0	0	0
1	OFF	OFF	0	0	0	1
2	OFF	OFF	0	0	1	0
3	OFF	OFF	0	0	1	1
4	OFF	OFF	0	1	0	0
5	OFF	OFF	0	1	0	1
6	OFF	OFF	0	1	1	0
7	OFF	OFF	0	1	1	1
8	OFF	OFF	1	0	0	0
9	OFF	OFF	1	0	0	1
10	OFF	OFF	1	0	1	0
11	OFF	OFF	1	0	1	1
12	OFF	OFF	1	1	0	0
13	OFF	OFF	1	1	0	1
14	OFF	OFF	1	1	1	0
15	OFF	OFF	1	1	1	1
16	OFF	ON	0	0	0	0
17	OFF	ON	0	0	0	1
18	OFF	ON	0	0	1	0
19	OFF	ON	0	0	1	1
20	OFF	ON	0	1	0	0
21	OFF	ON	0	1	0	1
22	OFF	ON	0	1	1	0
23	OFF	ON	0	1	1	1
24	OFF	ON	1	0	0	0
25	OFF	ON	1	0	0	1
26	OFF	ON	1	0	1	0
27	OFF	ON	1	0	1	1
28	OFF	ON	1	1	0	0
29	OFF	ON	1	1	0	1
30	OFF	ON	1	1	1	0
31	OFF	ON	1	1	1	1

k value	relay		output bit			
	A	B	4	3	2	1
32	ON	OFF	0	0	0	0
33	ON	OFF	0	0	0	1
34	ON	OFF	0	0	1	0
35	ON	OFF	0	0	1	1
36	ON	OFF	0	1	0	0
37	ON	OFF	0	1	0	1
38	ON	OFF	0	1	1	0
39	ON	OFF	0	1	1	1
40	ON	OFF	1	0	0	0
41	ON	OFF	1	0	0	1
42	ON	OFF	1	0	1	0
43	ON	OFF	1	0	1	1
44	ON	OFF	1	1	0	0
45	ON	OFF	1	1	0	1
46	ON	OFF	1	1	1	0
47	ON	OFF	1	1	1	1
48	ON	ON	0	0	0	0
49	ON	ON	0	0	0	1
50	ON	ON	0	0	1	0
51	ON	ON	0	0	1	1
52	ON	ON	0	1	0	0
53	ON	ON	0	1	0	1
54	ON	ON	0	1	1	0
55	ON	ON	0	1	1	1
56	ON	ON	1	0	0	0
57	ON	ON	1	0	0	1
58	ON	ON	1	0	1	0
59	ON	ON	1	0	1	1
60	ON	ON	1	1	0	0
61	ON	ON	1	1	0	1
62	ON	ON	1	1	1	0
63	ON	ON	1	1	1	1

K value : output condition code

Table 12.2 Output Condition Code

12.3.3 How to calculate the output condition code

The place value of the output bits and relay ports are shown below:

	Relay port		Output Bit			
	A	B	4	3	2	1
Place value	32	16	8	4	2	1

The output condition code is the sum of the place value of the relay port and the output bits which are turned on.

For example, to turn on relay B and output bits 1 and 3, the output condition code is number 21 which is the sum of 16, 1 and 4.

12.3.4 A simple program to illustrate the OUT command

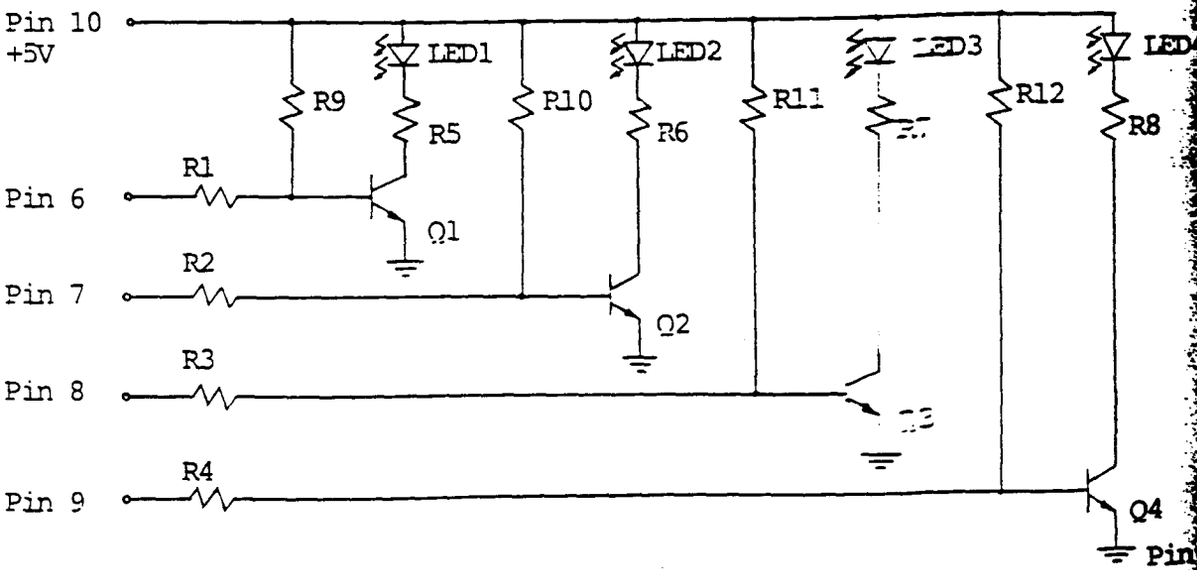
Below is a program to illustrate how to use the OUT command.

Key Entry	Display
- Press [0] [0] [0] [0]	"- 0000"
- Press [OUT]	"O XXXX"
- Press [6] [3]	"O 0063"
- Press [REC]	"- 0001"
- Press [PAUSE]	"P XXXX"
- Press [3] [0]	"P 0030"
- Press [REC]	"- 0002"
- Press [OUT]	"O XXXX"
- Press [0] [0] [0] [0]	"O 0000"
- Press [PAUSE]	"P XXXX"
- Press [3] [0]	"P 0030"

- Press [REC]	"- 0003"
- Press [JUMP]	"J XXXX"
- Press [0] [0] [0] [0]	"J 0000"
- Press [REC]	"J 0016"
- Press [1] [6]	"- 0004"
- Press [END]	"END"
- Press [REC]	"- 0005"

This program is to turn on all the output bits and relays, pause for 3 seconds, then turnoff all the output bits and relays, pause for 3 seconds. This is followed by an unconditional jump back to the origin and the sequence is repeated over and over again.

Before trying out the program, connect a simple circuit to the output ports as shown in Figure 12.2.



R1 to R8 330 1/8W
R9 to R12 4.7K 1/8W
Q1 to Q4 2N2222

Figure 12.2 Circuit for testing output port

When the program is run, LED 1 to 4 and the 2 relay LEDs are blinking at 3 seconds intervals.

This output pattern can be changed by changing the parameters in line 0001 and line 0002.

CHAPTER 13

UPLOAD COMMAND - [UPLD]

The UPLOAD command allows program generated on the hand-held teach pendant to be transferred from the controller to a diskette on a host computer for storage.

13.1 Procedures to activate UPLOAD command

1. Before initiating the UPLOAD command, an Apple II or compatible computer must first be connected to SIR-1 as described in Appendix D.

2. Boot the SIR-1 Demo Program, the screen will display a menu. Select 3 for UPLOAD/DOWNLOAD and the screen will prompt you to enter the slot number of the serial interface card and the disk drive you wish to use.

3. After this, the screen will prompt "Upload binary file to drive 1?", reply [Y] and the message "READY TO RECEIVE" will appear on the screen.

4. Now to initiate the uploading of the program from the controller, set the starting line number of the program on the teach pendent, press [SHIFT] [UPLD] and the display will show "L XXXX", enter the last line number of the program and press [REC]. The display will show "READY".

5. To effect the uploading, press [Y)] on the teach pendant.

6. When the operation is completed the screen will display the file length and whether the checksum is correct.

7. If the checksum is correct, the screen will prompt to enter the file name.

8. After entering the file name, press [RETURN] on the computer keyboard and the program is recorded on the diskette.

9. After the uploading operation, the display on the teach pendant will show "H XXXX" where XXXX is the checksum value. At this time, press [LINE] on the teach pendant will return SIR-1 to the command-entry state.

CHAPTER 14

DOWNLOAD COMMAND - [DNLD]

The download command is to reload a previously uploaded program back into the SIR-1 controller from the diskette on the host computer.

14.1 Procedures to activate the DOWNWARD command

1. All files to be downloaded must be previously uploaded by means of the [UPLD] command because the uploaded file is a memory image of SIR-1's user memory.

2. To initiate the downloading, set the starting line number of the program on the display of the teach pendant, then press [SHIFT] [DNLD]. When the display shows "READY" press [Y)], the display now show "....." and the controller is waiting for the program to be downloaded from the host computer.

3. Now prepare the host computer as in uploading. Boot the SIR-1 Utility Program, when the screen prompts "Upload binary file to drive 1?" reply [N]. The screen will then prompt "Download file to controller?" reply [Y] and enter file name accordingly.

5. When asked whether disk drive and controller ready, reply [Y]. The downloading is then in process. On completion the screen will show the file length, the file name and announce the process is completed.

6. At the same time the teach pendant display will show "PASS" to indicate the downloading is completed.

   If the display shows "Err" it means that there is a checksum error and the downloading data are not correct.

   In either case, press [LINE] to enter the command entry state.

CHAPTER 15

CASSETTE TAPE WRITE COMMAND - [TWR]

This command is used to save programs on an ordinary audio cassette tape.

15.1 Procedures to activate CASSETTE TAPE WRITE command - [TWR]

1. Connect a cable from the earphone jack marked "CASSETTE OUT" on the front panel of the cassette to the cassette recorder's microphone jack.

2. When SIR-1 is in the command entry state set the starting line number of the program to be saved on the display of the teach pendant, press [SHIFT] [TWR].

3. When the display shows "L XXXX". Enter the last line number of the program to save and press [REC].

4. When the display shows "F XXXX". Enter the file name (which should be a three decimal digit number) and press [REC]. The display will prompt "READY".

5. Wind the cassette tape to where the program wil be recorded.

6. Press [RECORD] on the cassette recorder and immediately press [Y] on the teach pendant.

   Now SIR-1 is sending out a header-message via its cassette interface. This header will last for 8 seconds and is used as an identification when the file is read back by SIR-1 using CASSETTE TAPE READ command. After the header-message, the program steps are transmitted to the cassette tape.

   When transmission is completed, a checksum which is a four-digit hexadecimal number "H XXXX" will display on the teach pendant.

7. Press [STOP] on the cassette recorder to stop it.

8. Press [LINE] to enter command-entry state.

CHAPTER 16

CASSETTE TAPE READ COMMAND - [TRD]

This command is used to input program from the cassette recorder to the controller.

16.1 Procedures to activate CASSETTE TAPE READ command - [TRD]

1. Connect a cable from the earphone jack marked "CASSETTE IN" on the front panel to a cassette recorder's earphone or speaker jack.

2. To activate cassette tape read command, press [SHIFT] [TRD] on the teach pendant when SIR-1 is in the command entry state.

3. The display will show "F XXXX", enter file name and press [REC]. The display will prompt "READY", pressing [Y], the display will show "....." and SIR-1 is searching for the desired file from the CASSETTE IN port. Press [PLAY] immediately on the cassette recorder.

   (It is advisable to wind the tape to where just a few seconds before the program's starting point so that SIR-1 does not need to wait too long to receive the program).

4. If the program does not exist on the tape or the process begins at the middle of the program or even worse the wrong file name has been entered, SIR-1 will search for it endlessly. To abort this process, press [SHIFT] [EMGY] on the teach pendant or the [RESET] switch on the controller's front panel.

5. During transmission, the speaker will sound and the display shows various patterns. Upon successful transmission, the display will show 'PASS'.

6. Press [LINE] to enter the command-entry state and manipulate the program just reloaded.

7. If there is any problem with the file eg. checksum not matched, an "Err X" message will appear on the display, where X is a number between 0 to 9 depending on the type of error. See Appendix E for error code interpretation and remedial action.

8. The program is read back to the same place in the memory when it is written to the cassette. All line numbers for the program are remain the same.

CHAPTER 17

HOST MODE COMMANDS

SIR-1 has a set of host mode commands, these commands are sent from the host computer to the SIR-1 controller through the RS-232C serial interface. This enable the user to write their own program to control the robot instead of using the teach pendant. Hence, the power of the host computer can be added to the SIR-1. Next few sections explain the set of host mode commands.

17.1 MOVE command

1. Syntax : M b, s, e, p, r, g, x, y, k <cr>

   b, s, e, p, r, g, x, y represent the parameters for the joint motors, they are step numbers which range from - 32767 to + 32767 though such large numbers are not necessary for operation in SIR-1's effective working area.

   K is the parameter for user output port and relays.

   The K value is the same as the output condition code for the OUT command in the teach mode. Refer to Table 12.2 for selection of K value.

2. MOVE command can activate the following simultaneously:-

   - move the motor(s) a specified number of steps
   - output a pattern through the user output port and turn ON or OFF selected relays.

3. In using the MOVE command (also true for SPEED command), it is not necessary to move all the 8 motors simultaneously.

4. Example

   4.1 To rotate the base motors 50 steps without moving any other motors or setting any output bits and relays. The entry is as follows:-

   M 50, 0, 0, 0, 0, 0, 0, 0 <cr>

   4.2 In this case zeros can be omitted without altering the meaning of the command, so the entry can be simply:

   M 50 <cr>

4.3 However to hold the base, shoulder and elbow fixed while moving all the other motors 50 steps, the entry should be as follows:-

M 0, 0, 0, 50, 50, 50, 50, 50 <cr>

or just:

M , , , 50, 50, 50, 50, 50 <cr>

but cannot be:

M 50, 50, 50, 50, 50 <cr>

because this would be interpreted as a command to hold the grippers X and Y motors fixed while moving all other joint motors 50 steps.

NOTE
Omitting the K value leaves the output port and relays unchanged. It does not set them all to zero.

17.2 SPEED command

1. Syntax : S b, s, e, p, r, g, x, y <cr>

   b, s, e, p, r, g, x, y represent the parameters for the joint motors, they are speed numbers which range from 0 to 7, 0 is slowest and 7 is the fastest speed.

   At power-up, all the motors are set at speed "7" (the fastest speed).

2. SPEED command is used for setting the speed of the joint motors.

   SIR-1 allows every motor to have its own moving speed, that is, all the 8 axes can move at different speeds.

17.3 PAUSE command

1. Syntax : P t <cr>

   t is a number from 0 to 32767. The actual time paused is t x 0.1 second(s) thus the programmable pause duration is from 0.1 second to 3276.7 seconds. (approx. 55 minutes)

2. PAUSE command is used to halt the execution for a specified period of time. Then execution continues with the next program line.

17.4 HOME command

1. Syntax : H <cr>

2. HOME command is used to bring SIR-1 to its soft-home position.

   This position is normally used as a reference point during start-up or when the controller loses the currect position information (ie. the robot does not know where it is) and wants to start all over again from a known position.

17.5 TEACH command

1. Syntax : T <cr>

2. TEACH command is used to activate the teach pendant during programming on the host computer.

3. After this command, the user can operate the joint-control keys on the teach pendant to drive the robot to a desired position.

4. To return to the HOST mode, press [SHIFT] [EMGY] and then [HOST].

5. The controller will send an ACII "3" to the host computer to indicate that the control is passed back to the host computer from the teach pendant.

CHAPTER 18

HOW TO ENTER THE HOST MODE

This chapter describes how to enter the host mode for programming using the host computer.

After SIR-1 has completed its power-up initialization sequence as described in chapter 4, SIR-1 enters the mode-selection state.

To go into host-mode operation press [HOST] on the teach pendant and display shows "....." which indicates that SIR-1 is ready to respond to the host mode commands which are explained in chapter 19.

CHAPTER 19

HOW TO WRITE HOST PROGRAM

There are three elements in writing the host program :

- hand shaking
- sending host mode commands
- receiving return code.

19.1 Hand Shaking

Before sending any host command a hand shaking must be done with the SIR-1 controller first. To do so simply send an ASCII "ESC" to the SIR-1 controller.

Upon receiving the hand shake signal, the controller will reply with an "enabled motor" code (see section 19.3) back to the host computer. At the same time the display on the teach pendant will be changed to "HHHHHH".

After this hand shaking, the communication between the SIR-1 controller and the host computer is established. The controller will keep on scanning the RS-232C port and expecting the host mode commands to be sent from the host computer.

19.2 Sending Commands

Sending host mode commands is simple and straight forward. You can use "PRINT" statement in BASIC to send out any command according to the syntax as explained in chapter 17. The only exception is sending "READ" command. Examples of how to send all the commands are given in the next few sections.

19.3 Receiving Return Codes

At certain time, SIR-1 controller need to return codes to the host computer to inform the host computer about its status. The codes are:

19.3.1 "enabled motor" code

This code is returned when the controller receives the ASCII "ESC" from the host computer.

The code is a number which ranges from 0 to 255. The user has to decode it by changing it to a 8-bit binary number. The LSB (least significant bit) to MSB (most significant bit) represent the working condition of the base, shoulder, elbow, pitch, roll, gripper, X and Y motors respectively.

19.3.2 ASCII "1" & "3" code

Upon receiving a "T" command, the controller replies an ASCII "3" code and goes into the teach mode. After the user manipulate the robot arm to a desire position and return to the host mode by pressing [SHIFT] [EMGY STOP] follow by [HOST], the controller reply an ASCII "1" code and return to host mode.

The user must keep on checking the return code from the controller after sending a "T" command, he may proceed to send the subsequent command only after receiving the return code ASCII "1".

19.4 Host Program #1

Following is a sample host program written on Apple II computer:

```
100  REM  ********************************
110  REM     TO INITIALIZE THE 7710 CARD
120  REM  ********************************
130 D$ =  CHR$ (4)
140  PRINT D$;"PR#2": REM  OUTPUT TO SLOT NO.2
150  POKE  - 16223,27: REM  SET CONTROLLER
160  REM  ********************************
170  REM         MAIN  PROGRAM
180  REM  ********************************
190  PRINT "H"
200  PRINT "S 5,5,7"
210  PRINT "M 100,175,100"
220  PRINT "P 10"
230  PRINT "S ,,6,,7"
240  PRINT "M ,,-150,,-100"
250  PRINT "P 20"
260  PRINT "H"
270  REM  ********************************
280  PRINT D$;'PR#0": REM  OUTPUT TO MONITOR
290  END
```

Line 130 and 140 set the output to slot number 2 which the serial interface card resides.

Line 150 send the hand shaking code ASCII "ESC" to the controller. 27 is the ASCII code for "ESC", please refer to Appendix G ASCII Code Table for details.

Line 190 homes the robot arm if it is not already at the home position.

Line 200 sets the speed for base, shoulder and elbow to 5, 6 and 7 respectively.

Line 210 moves the base, shoulder and elbow certain number of steps.

Line 220 pauses the robot arm for 1 second.

Line 230 sets the speed of elbow and roll.

Line 240 moves the elbow and roll.

Line 250 pauses the robot arm 2 seconds.

Line 260 returns the robot arm to home position.

Line 280 sets the output back to monitor.

Before running this simple program, connect the Apple II computer to the SIR-1 as described in Appendix D and set SIR-1 to the host mode as stated in chapter 18.

After the teach pendant display ".....", boot the SIR-1 Demo Diskette. The screen will display a menu, select 5 to exit and then type RUN PROGRAM#1.

The robot arm will move accordingly and the teach pendant displays "HHHHHH".

19.5 Host Program #2

```
100 D$ = CHR$ (4)
110 PRINT D$;"PR#2"
120 POKE - 16333,27
130 D = PEEK ( - 16333)
140 PRINT D$;"PR#0"
170 REM  ****************************************
180 REM       ROUTINE TO DECODE THE DATA BYTE
190 REM  ****************************************
200 IF D > = 128 THEN  PRINT "Y MOTOR":D = D - 128
210 IF D   = 64 THEN  PRINT "X MOTOR":D = D - 64
220 IF D   = 32 THEN  PRINT "GRIPPER":D = D - 32
230 IF D   = 16 THEN  PRINT "WRIST ROLL":D = D - 16
240 IF D   = 8 THEN  PRINT "WRIST PITCH":D = D - 8
250 IF D   = 4 THEN  PRINT "ELBOW":D = D - 4
260 IF D   = 2 THEN  PRINT "SHOULDER":D = D - 2
270 IF D   = 1 THEN  PRINT "BASE"
280 REM  ****************************************
290 END
```

This program illustrates how to read the "enabled motor" code and how to decode it. After running this program, names of the working motor will be displayed on the screen.

19.6 Host Program #3

```
100  REM  ********************************
110  REM           MAIN  PROGRAM
120  REM  ********************************
130  HIMEM: 36863: REM  SET HIGHEST MEMORY FOR BASIC PROGRAM
140 D$ =  CHR$ (4)
150  PRINT D$;"BLOAD READ": REM  LOAD ASSEMBLY SUBROUTINE
160  PRINT D$;"PR#2"
170  POKE  - 16223,27
180  PRINT "H"
190  PRINT "M 123,45,67,89"
200  PRINT "R": CALL  - 28672: REM  CALL SUBROUTINE AFTER R COMMAND
210  PRINT D$;"PR#0": REM  RETURN OUTPUT TO MONITOR
220  GOSUB 280
230  PRINT D$;"PR#2"
240  PRINT "P 20"
250  PRINT "H"
260  PRINT D$;"PR#0"
270  END
280  REM  ***********************
290  REM     PRINTING THE DATA
300  REM  ***********************
310 AM =  - 28576: REM  BEGINNING ADDRESS OF THE DATA
320 I = 0: REM  SET INDEX TO 0
330 I1 = 0: REM  STRING INDEX
340 D =  PEEK (AM + I)
350  IF D = 0 THEN 430: REM  CHECK FOR LAST DATA
360  IF D = 13 THEN 400: REM  CHECK FOR CARRIAGE RETURN
370 M$(I1) = M$(I1) +  CHR$ (D): REM  CREATING A POSITION STRING
380 I = I + 1: REM  INCREMENT MEMORY INDEX
390  GOTO 340
400 I = I + 1
410 I1 = I1 + 1
420  GOTO 340
430 I1 = I1 - 1
440  FOR C = 0 TO I1
450  PRINT M$(C)
460  NEXT
470  RETURN
```

```
0800                    1  ;***********************************************
0800                    2  ;
0800                    3  ; SUBROUTINE: READ
0800                    4  ;
0800                    5  ;    PURPOSE: TO RECEIVE ALL MESSAGE DATA FROM
0800                    6  ;             THE CONTROLLER AFTER A 'R' COMMAND
0800                    7  ;             IS SEND
0800                    8  ;
0800                    9  ; SUBROUTINE CALLED: CHKDTA
0800                   10  ;                     - TO CHECK IF A BYTE IS
0800                   11  ;                       RECEIVED
0800                   12  ;
0800                   13  ;       FLAG: V - TO INDICATE THAT THE FIRST DATA
0800                   14  ;                 IS RECEIVED
0800                   15  ;
0800                   16  ;***********************************************
0800                   17  ;
0800                   18  ;PROGRAM EQUATES
0800                   19  ;
C080                   20  CMD       EQU $C080           ;+$NO FOR ACIA COMMAND PORT
C081                   21  DATA      EQU CMD+1
AA54                   22  CSWH      EQU $AA54           ;CHARACTER OUTPUT SWITCH
00FE                   23  COUNT     EQU $FE             ;A COUNT
00FF                   24  NODATA    EQU $FF             ;A FLAG TO INDICATE NO MORE DA
TA
0000                   25  ZERO      EQU $00
FFCB                   26  SETV      EQU $FFCB           ;SET V=1
002B                   27  POS       EQU $2B             ;POSITIVE SIGN
002D                   28  NEG       EQU $2D             ;NEGATIVE SIGN
0800                   29  ;
0800                   30  ; M A I N     P R O G R A M
0800                   31  ;
0800                   32  ;
0800                   33  ;INITIALIZATION
0800                   34  ;
9000                   35            ORG $9000
9000 AC 54 AA          36            LDY CSWH            ;LOAD SLOT ADDRESS TO INDEX Y
9003 98                37            TYA                 ;MULTIPLY BY 16 TO GET THE
9004 0A                38            ASL                 ;  $NO INDEX TO ACCESS THE ACI
A
9005 0A                39            ASL
9006 0A                40            ASL
9007 0A                41            ASL
9008 A8                42            TAY                 ;ESTABLISH Y INDEX
9009 A2 00             43            LDX #ZERO           ;SET X=0
900B 86 FF             44            STX NODATA
900D B8                45            CLV                 ;CLEAR V
900E                   46  ;
900E A9 00             47  READ      LDA #ZERO           ;SET COUNT=0
9010 85 FE             48            STA COUNT
9012 20 4C 90          49            JSR CHKDTA          ;CHECK IS DATA IS HERE
9015 70 18             50            BVS CHECK
9017 B9 81 C0          51            LDA DATA,Y          ;LOAD DATA INTO 'A'
901A C9 2B             52            CMP #POS            ;CHECK FOR SIGNS
```

```
901C F0 07      53            BEQ SET
901E C9 2D      54            CMP #NEG
9020 F0 03      55            BEQ SET
9022 4C 0E 90   56            JMP READ
9025 9D 60 90   57  SET       STA STATUS,X        ;STORE FIRST DATA
9028 E8         58            INX
9029 2C CB FF   59            BIT SETV            ;SET V FLAG
902C 4C 0E 90   60            JMP READ
902F C4 FF      61  CHECK     CPY NODATA
9031 F0 0A      62            BEQ FINISH          ;IF NO MORE DATA THEN FINISH
9033 B9 81 C0   63            LDA DATA,Y
9036 9D 60 90   64            STA STATUS,X        ;STORE DATA
9039 E8         65            INX                 ;INCREMENT INDEX
903A 4C 0E 90   66            JMP READ
903D            67  ;
903D A9 00      68  FINISH    LDA #ZERO           ;STORE ZERO TO LAST BYTE
903F 9D 60 90   69            STA STATUS,X
9042 60         70            RTS
9043            71  ;
9043            72  ; SMALL ROUTINE TO CHECK DATA'S ARRIVAL
9043            73  ;
9043 B9 80 C0   74  CHKDTA    LDA CMD,Y           ;CHECK DATA IS HERE
9046 4A         75            LSR
9047 B0 0C      76            BCS DATAIN
9049 50 F8      77            BVC CHKDTA
904B E6 FE      78            INC COUNT
904D A9 FF      79            LDA #$FF
904F C5 FE      80            CMP COUNT
9051 D0 F0      81            BNE CHKDTA
9053 84 FF      82            STY NODATA
9055 60         83  DATAIN    RTS
9056            84  ;
9060            85            ORG $9060
9060            86  STATUS    DFS $40,0
90A0            87            END
```

This program illustrates how to send "R" command and how to receive the data (the absolute positions of the eight joints and the input port status).

An assembly language subroutine "READ" must be called immediately after sending the "R" command as in line 200. This subrontine is used to recive the position data send from the controller. We cannot use Applesoft BASIC because it is not fast enough to receive the data.

This subroutine is resided on the DEMO PROGRAM diskette, Line 150 load the subroutine. The starting address of the subroutine is $9000 (hexadecimal) or -28672 (decimal). Line 200 sends out the "R" command and then run the subroutine.

The subroutine waits for the first sign byte and discards other data. After receiving the sign byte, it starts to receive the subsequent data. When all data is received, a zero is inserted into the last byte to indicate the end of data. The starting address to store these data is $9060 or -28576.

To retrieve the position data, the format of the data stored must be understood. For a motor's absolute position, the first byte is a sign byte "+" or "-", followed by 5 bytes each represents a digit. The last byte is a carriage return. As a result, the position data of a motor consists of 7 bytes. The lenght of the status data for the input port is 4 bytes long, one sign byte, two digit bytes and a carriage return. With eight motor and the input port, the maximum length of data is 60 bytes.

The BASIC subroutine starting from line 280 is used to retrieve the information and print them on the screen. M$(0) is a string variable which represents the absolute position of base, M$(1) for shoulder and so on. To convert the strings into numeric values, you can use VAL statement.

Line 130 specifies the highest memory of the BASIC program to prevent it from overlapping with the assembly language subroutine.

19.7 Host Program #4

```
100  REM  ************************************************
110  REM              M A I N          P R O G R A M
120  REM  ************************************************
130  HIMEM: 36863: REM  SET HIGHEST MEMORY FOR BASIC PROGRAM
140 D$ =  CHR$ (4)
150  PRINT D$;"BLOAD READ"
160  PRINT D$;"PR#2"
170  POKE  - 16223,27
180  PRINT "R": CALL  - 28672
190  GOSUB 480
200  FOR I = 0 TO 7:B(I) =  VAL (M$(I)): NEXT
210  PRINT D$;"PR#0"
220  PRINT : PRINT "YOU MAY NOW MOVE THE ROBOT WITH"
230  PRINT "THE TEACH PENDANT.  WHEN YOU HAVE"
240  PRINT "FINISHED, PLEASE RETURN TO HOST MODE.": PRINT
250  PRINT D$;"PR#2"
260  PRINT "T"
270 EC =  PEEK ( - 16223)
280  IF EC <  > 49 THEN 270
290  PRINT "R": CALL  - 28672
300  GOSUB 480
310  FOR I = 0 TO 7
320 A(I) =  VAL (M$(I))
330 D(I) = A(I) - B(I)
340  NEXT
350  PRINT "H"
360  PRINT D$;"PR#0"
370  PRINT  TAB( 10);"BEFORE"; TAB( 20);"AFTER"; TAB( 30);"DIFFERENCE"
380  FOR I = 0 TO 7
390  PRINT "MOTOR ";I; TAB( 10);B(I); TAB( 20);A(I); TAB( 30);D(I)
400  NEXT
410  PRINT : PRINT "THE EQUIVALENT 'MOVE' STATEMENT IS:"
420  PRINT "M ";D(0);
430  FOR I = 1 TO 7
440  PRINT ",";D(I);
450  NEXT
460  PRINT
470  END
480  REM  ************************************
490  REM        GET DATA FROM MEMORY
500  REM  ************************************
510 AM =  - 28576: REM  BEGINNING ADDRESS OF THE DATA
520 I = 0: REM  SET INDEX TO 0
530 I1 = 0: REM  STRING INDEX
535 M$(I1) = ""
540 D =  PEEK (AM + I)
550  IF D = 0 THEN 630: REM  CHECK FOR LAST DATA
560  IF D = 13 THEN 600: REM  CHECK FOR CARRIAGE RETURN
570 M$(I1) = M$(I1) +  CHR$ (D): REM  CREATING A POSITION STRING
580 I = I + 1: REM  INCREMENT MEMORY INDEX
590  GOTO 540
600 I = I + 1
610 I1 = I1 + 1
620  GOTO 535
630 I1 = I1 - 1
640  RETURN
```

This program illustrates how to use "T" command. It also display the position of a point before teaching and after teaching. It finds the differences and shows the equivalent MOVE command to move to the point being taught from the previous point. It also make use of the same "READ" subroutine as in the previous example.

Line 180 reads the absolute position of the robot arm before being moved and stores them in array B (B for before).

Line 260 sends the "T" command, after which you can manipulate the robot arm to a desired point. You should return to the host mode after doing so.

After sending the "T" command you should keep checking the return code from the controller to be "1" before sending out the subsequent commands. Line 270 and 280 performs this function.

Line 290 reads the position data after moving the robot arm and stores them in array A (A for after).

The program then finds the differences and stores them in array D (D for differences). It homes the robot and print all data on the screen.

Appendix A Robot Arm Adjustment

If properly used, SIR-1 robot needs little maintenance or adjustment. However, certain mechanical parts may need re-tightening or lubricating after continuous operation for a long time.

Following are the adjustment procedures for those parts most likely need adjustment :

1. Closing the body swivel gap

   The body swivel gap can be reduced by loosening the screws then pushing the base motor downward so that the pinion on the motor output shaft moves downwards until it is properly engaged to the bevel gear (see Fig. A-1). Then tighten the screws.

2. Encoder disk adjustment

   Sometimes the encoder disk oscillate after the motion of robot arm has stopped. This can be remedied by tightening the encoder disk. First, loosen the set screw as shown in Fig. A-2 and then push the encoder disk closer to the motor to tighten it. Ensure that you don't over tighten it, overtightening will strain the motor and it may not move during play back.

3. Installation of gripper

   The connector is already fixed on the grippers before shipping to user. Simply insert the grippers to the shaft of the wrist roll motor. Align the flat part of the shaft with the position of the set screw and ensure that there is about 1 mm gap between the connector and wrist and then tighten the set screw. Insert the plug leading out from behind the wrist to the encoder PCB. (See Fig A-3)

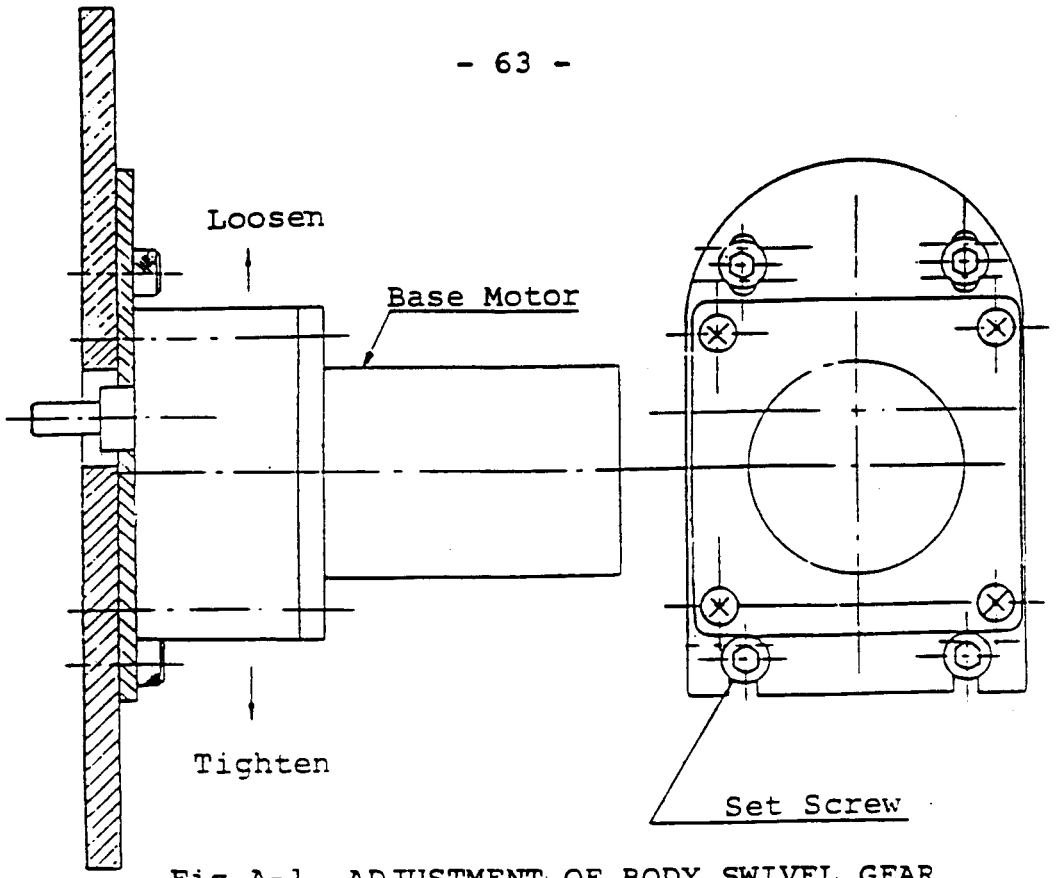


Fig A-1 ADJUSTMENT OF BODY SWIVEL GEAR

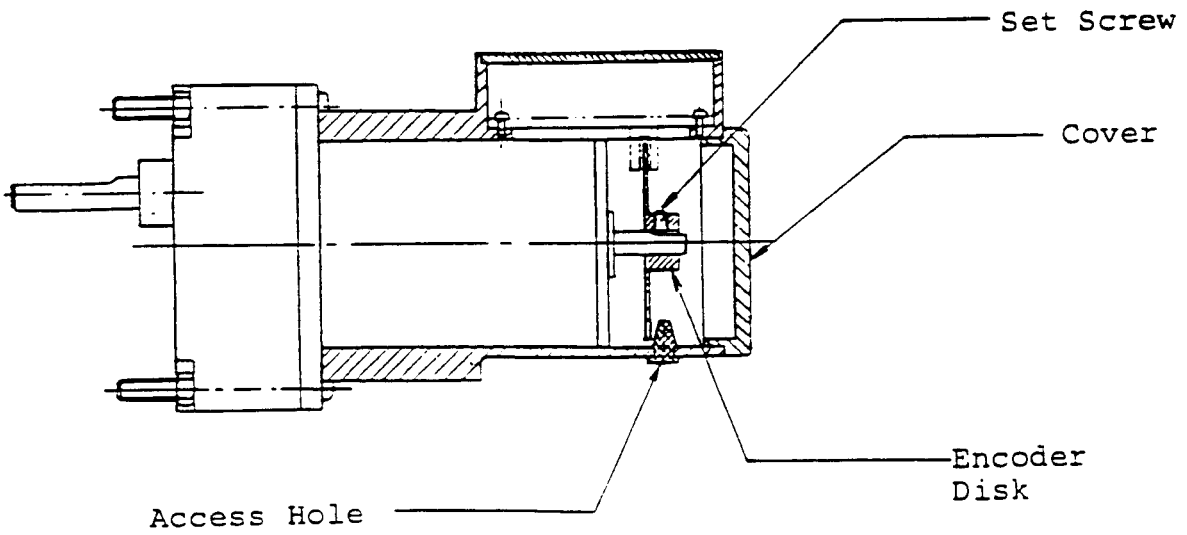


FIG A-2 ENCODER DISK ADJUSTMENT

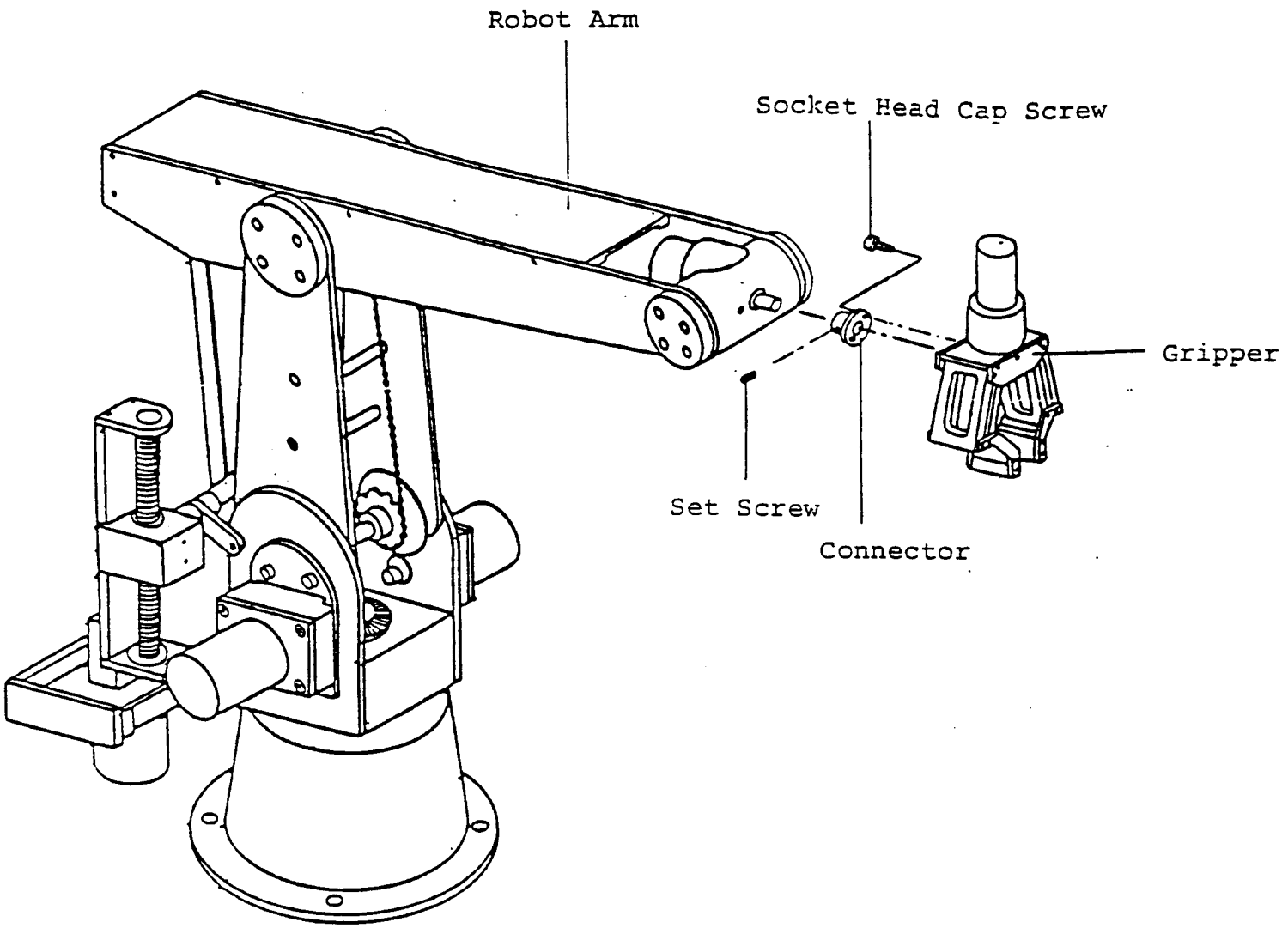


Fig. A-3 INSTALLATION OF GRIPPERS

Appendix B Controller Ajustment

SW1 : It is used to configure the controller to be a DCE (data communication equipment) or a DTE (data terminal equipment). If it is switched forward (view from front panel), pin 2 on RS 232C serial interface port = TxD, pin 3 = RxD (Factory-set condition). If it is switchedbackward, pin 2 on J12 = RxD, pin 3 = TxD.

DW2 : This is the baud rate selection switch. See Table B-1 for further details.

DW2 1	DW2 2	DW2 3	Baud rate selected	Factory-set condition
off	off	off	110	
on	off	off	150	
off	on	off	300	
on	on	off	600	
off	off	on	1200	
on	off	on	2400	
off	on	on	4800	
on	on	on	9600	X

Table B-1 Band Rate Settings

Appendix C 7-Segment Interpretation of the digits and alphabets

DIGIT	PATTERN	ALPHABET	PATTERN	ALPHABET	PATTERN
0		A		L	
1		B		M	
2		C		O	
3		D		P	
4		E		R	
5		F		S	
6		G		T	
7		H		X	
8		I		Y	
9		J			

Appendix D Serial Port Configuration

1. The RS-232C serial port in the SIR-1 controller is originally configure to suit most commonly used personal computers. However, if your computer's serial port does not work in the same way, this Appendix will guide you to configure the controller to suit the computer.

2. Configuring the serial port refers to making sure that your computer can talk to one another. This requires taking care of the following :

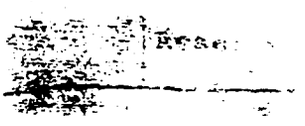
   - electrical connections
   - transmission rate (baud rate)
   - data format
   - standard interface signals

3. Electrical connections

   3.1 On the right side of the controller's front panel you will find two multi-pin connectors. The right most one is the serial port (remember the left one is for teach pendant). SIR-1 permits user to test its serial port even though it is without a host computer, provided that user has a data terminal. But it is important that the transmit and receive lines are properly interfaced.

   3.2 When connected to a computer, the receive line should be on pin 3 of the RS 232C connector of the controller, and the transmit should be on pin 2 and this is the factory-set condition.

   3.3 When connect SIR-1 to a terminal, which is a computer peripheral, these two lines must be swapped. To do this, power off the controller, remove the lid and find a black slide switch SW1 on bottom right of the controller card (view from front panel). Push it backward, replace the lid and power on the controller. Now the receive and transmit lines have already been reversed.



4. Transmission rate

SIR-1 is shipped with the serial port set to operate at a transmission rate of 9600 baud (9600 bits per second) for both send and receive. This rate can be changed to any of seven other standard rates by means of setting DSW6 located on SIR-1's controller card. The available rates and the corresponding switch settings are given in Table A-1. These switches should be changed when power is off, since this switch settings are only read by SIR-1 on power-up only. As with SIR-1, most computers have some means of setting baud rate - either through switches on a circuit board or via commands that can be issued under the computer's operating system or as part of BASIC. The main thing is that both SIR-1 and your computer be configured to operate at the same baud rate; otherwise communication between the two will be impossible.

5. Data format

SIR-1 uses the following data format :

8 bits word length
1 start bit
2 stop bits
no parity bit
full duplex (transmit and receive can happen simultaneously)

Many computers have the above as their "default" format. If doesn't, consult computer manual to learn how to do this.

6. Connection between SIR-1 and Apple II Computer

Insert the serial interface card (CCS 7710A) to slot number two at the expansion slot in the Apple II computer. You may place it at other slot, but the address of the command and data register will be changed. Set the baud rate on the serial interface card to be the same as the SIR-1 controller setteng. The default setting on the controller is 9600.

Connect the serial interface card to the SIR-1 controller by using a pair of 25-pin connectors. The connector should plug into the 25-pin connector labelled EIA-RS232C.

Appendix E Error Codes Interpretation and Remedial Action

Error Code	Error	Problem / Symptom	Remedial Action
1	illegal key entry	An attempt has been made to press a key that is not allowed at this moment. Nothing is affected and SIR-1 is still in the same state before the error occurred.	Press the correct key
2	Checksum error	This error occurs if the calculated checksum is not equal to that transferred with the file during DOWNLOADING or TAPE-READING	(1) repeat the process one more time. (2) check the connections between SIR-1 and the host computer in the case of Downloading. (3) check the connections between SIR-1 and the cassette recorder in the case of Tape-reading. (4) if the error persists, call service personnel.
3	line not found	The requested line does not exist.	Key in the correct line number.
4	reserved	-	-
5	illegal quantity	An attempt has been made to input into SIR-1 a quantity that is not in the legal range. eg. speed number, input condition code, output condition code etc.	input the correct value.
6	Reserved	-	

Error Code	Error	Problem / Symptom	Remedial Action
7	Reserved	-	
8	Out of memory	SIR-1 is running out of memory to store your program sequences. This happens when your input program is too large to fit into the current user RAM space (resulted from [REC], [INSERT] or [COPY] command). Nothing is affected except the last-input line is not entered.	(1) First save the current program into a host computer or a cassette recorder and expand the user RAM. Then powerup and retrieve the saved program. Resume the input process with the line the error occurred. (2) Delete some unwanted program lines from the memory and then continue with the failed line.
9	joint limit reached	A joint has been driven to the extreme of its working range and the motor power to that joint has been cut off.	When Error 9 occurs it will also show which joint has problem. Eg. "B Err 9" means base joint is out of limit. Turn off the motor power. Turn the encoder disk manually until the limit switch is clo ed back. Switch on the motor power and activate the motor so that it is moving away from the limit.

Appendix F : Memory Expension

There are two types of memory chips that can be used in the SIR-1 controller, namely 6116 and 5565. 6116 is 2K bytes type and 5565 is 8K bytes type, both have the access time of 200 NS. The SIR-1 controller comes standard with 4K bytes RAM memory which uses 2 pieces of 6116 - U16 and U17.

You can expand 4K bytes more of memory by simply inserting two more pieces of 6116 at U18 and U19. If you want to expand only 2K bytes you must insert the IC at U18. Please note that after inserting the IC, you should see four empty pins on top of the IC sockets which have 28 pins but the 6116 have only 24 pins.

By using 5565 you can expand the RAM memory to 32K bytes. Do the necessary jumper and switch setting as follows :

i) Select the jumper labelled 6116/5565 which is located right on top of U35 (40 pin IC) to the 5565 position.

ii) Select the switch setting of DW1 which is directly below U35 as follows :

DW1								RAM
1	2	3	4	5	6	7	8	Type
OFF	OFF	OFF	OFF	ON	ON	ON	ON	6116
ON	ON	ON	ON	OFF	OFF	OFF	OFF	5665

Remove the memory chips if they are 6116s and insert 5665 from IC 16 to IC 19.

Note :1. Switch off all the power before you do this.

2. You can not use two types of memory chips at the same time.

Appendix G ASCII Code Table

ASCII Code	Display Screen Character	Keystroke	ASCII Code	Display Screen Character	Keystroke
0		CTRL-@	48	0	0
1		CTRL-A	49	1	1
2		CTRL-B	50	2	2
3		CTRL-C	51	3	3
4		CTRL-D	52	4	4
5		CTRL-E	53	5	5
6		CTRL-F	54	6	6
7	(bell)	CTRL-G	55	7	7
8	(backspace)	CTRL-H or ←	56	8	8
9		CTRL-I	57	9	9
10	(linefeed)	CTRL-J	58	:	:
11		CTRL-K	59	;	;
12		CTRL-L	60	<	<
13	(carriage return)	CTRL-M	61	=	=
14		CTRL-N	62	>	>
15		CTRL-O	63	?	?
16		CTRL-P	64	@	@
17		CTRL-Q	65	A	A
18		CTRL-R	66	B	B
19		CTRL-S	67	C	C
20		CTRL-T	68	D	D
21	(forward space)	CTRL-U or →	69	E	E
22		CTRL-V	70	F	F
23		CTRL-W	71	G	G
24	(cancel line)	CTRL-X	72	H	H
25		CTRL-Y	73	I	I
26		CTRL-Z	74	J	J
27		ESC	75	K	K
28		n.a.	76	L	L
29		CTRL-SHIFT-M	77	M	M
30		CTRL-^	78	N	N
31		n.a.	79	O	O
32	space	space bar	80	P	P
33	!	!	81	Q	Q
34	"	"	82	R	R
35	#	#	83	S	S
36	$	$	84	T	T
37	%	%	85	U	U
38	&	&	86	V	V
39	'	'	87	W	W
40	(	(	88	X	X
41	)	)	89	Y	Y
42	*	*	90	Z	Z
43	+	+	91	[	n.a.
44	,	,	92	\	n.a.
45	-	-	93	]	SHIFT-M
46	.	.	94	^	^
47	/	/	95	_	n.a.

Note: Add 128 to ASCII codes if generated in the Monitor or Integer BASIC keystrokes.
n.a = not available on the Apple II keyboard.