SRS-M1

SMALL INDUSTRIAL ROBOT SYSTEM
TECHNICAL MANUAL

Manual Order Number: SRS-TECHNICAL/M1

Copyright 1987 CRS Plus Inc

REV.	REVISION HISTORY	DATE
-001	Original Issue	12/85
-002	Combine Installation & Technical	2/86
-003	Add additional information	5/86
-004	Version 3.30 Support for data base pointers in ACI	9/86
-005	Improved support for ACI functions Valid for RAP1 version 3.46 or later	2/87
-006	Technical references in Appendix F Support for SRS-M1A	6/87

Additional copies of this manual or other CRS Plus literature may be obtained from:

CRS Plus Inc.
P.O. Box 163 Stn 'A'
Burlington, Ontario
L7R 3Y2
(416)639-0086

The information in this document is subject to change without notice.

CRS Plus Inc makes no warranty of any kind with regard to this material, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. CRS Plus Inc assumes no responsibility for any errors that may appear in this document. CRS Plus Inc makes no commitment to update nor to keep current the information contained in this document.

CRS Plus Inc assumes no responsibility for the use of any circuitry other than circuitry embodied in a CRS product.

CRS Plus Inc software products shall remain the property of CRS Plus Inc.

This manual provides general information, installation instructions, programming information, principles of operation for the CRS Plus SRS-M1 Small Industrial Robot System. Additional information is available in the following documents:

* RAPL Programming Manual

* SRS-M1 Small Industrial Robot System - Tutorial Manual

* SRS-M1 Small Industrial Robot System - IBM-PC Host Interface

CONTENTS

PAGE

CHAPTER 1
GENERAL INFORMATION
System Description 1-1
Robot Arm 1-2
RSC-M1A Robot System Controller 1-3
Teach Pendant 1-5

CHAPTER 2
SPECIFICATIONS
Robot Arm 2-1
Workspace Elevations - Top View 2-2
Workspace Elevations - Side View 2-3
RSC-M1A Robot System Controller 2-4
Teach Pendant 2-5

CHAPTER 3
PREPARATION FOR USE
Unpacking and Inspection 3-1
Robot Base Installation 3-2
Gripper Installation 3-3
RSC-M1A Installation 3-4
Cabling 3-5
Video Terminal Installation 3-6
Printer/Host Computer Installation 3-7
Digital I/O Installation 3-8
Analog Input Installation 3-10

CHAPTER 4
PRECAUTIONS FOR USE
Safety 4-1
Handling of Robot 4-1
Power Fail 4-1

CHAPTER 5
START-UP OPERATIONAL SEQUENCE
Power Up Sequence 5-1
Homing Sequence 5-1

CHAPTER 6
MECHANICAL DESCRIPTION
Robot Arm Construction 6-1
Drive Motors 6-3
Drives and Transmissions 6-4
Arm Assembly Drawings 6-6

CHAPTER 7
ELECTRONIC DESCRIPTION

Introduction 7-1
Motherboard 7-2
D.C. Amplifier Module 7-5
Servo Axis Card 7-7
Arm Power Relay Board 7-9
I/O Termination Board 7-10
Computer Power Supply 7-11
Arm Power Supply 7-11

CHAPTER 8
SOFTWARE DESCRIPTION

Introduction 8-1
Memory Management 8-1
Computational Accuracy 8-1
Path Control - Closed Loop Control 8-1
Path Control - Command Generation 8-2
Input/Output Scanning 8-3
Memory Allocation 8-4

CHAPTER 9
HOMING BRACKET INSTALLATION

Introduction 9-1
Robot Calibration 9-1
Auto Start Program 9-3

APPENDIX A
MONITOR FUNCTIONS

Description A-1

APPENDIX B
ADVANCED COMMUNICATION INTERFACE (ACI)

Introduction B-1
Functional Description B-1
Master Protocol B-1
Enquiry Sequence B-2
Header Sequence B-3
Data Block Transmission Sequence B-5
EOT Sequence B-6
Error Checking and Recovery B-6
ACI Error List B-7
Communication Cycle Abort B-10
Interface Requirements B-10
Preparing The Robot Controller B-10
ACI Monitor Commands B-11
ACI Special Access Codes B-14

APPENDIX C
MAINTENANCE AND TROUBLESHOOTING

Mechanical Maintenance C-1
Service Routines C-2
Electronic Troubleshooting C-8

APPENDIX D
MECHANICAL DRAWINGS

Introduction D-1

APPENDIX E
OPTICAL ENCODER SPECIFICATIONS

Description E-1

APPENDIX F
SRS-M1 ROBOT CONTROLLER TECHNICAL DESCRIPTION

Introduction F-1
Memory Space Allocation F-1
Input/Output Space F-2
Axis Control F-5
Interrupt Control F-6
System Digital Output Useage F-8
System Digital Input Useage F-10
RAPL Software Parameter List F-12
RAPL User Memory Allocation Description F-17

TABLES

TABLE	TITLE	PAGE
2-1	Robot Arm Specifications	2-1
2-2	RSC-M1 Specifications	2-4
2-3	Teach Pendant Specifications	2-5
3-1	SRS-M1 Parts List	3-1
6-1	Robot Joints - Range of Motion	6-2
7-1	RSC-M1A Controller Description	7-1
7-2	Controller Connector Map	7-3
7-3	D.C. Amplifier Description	7-6
7-4	Servo Axes Card Description	7-8
8-4	User Memory Allocation	8-9
B-1	Header Block Description	B-3
B-2	ACI Error Codes	B-7
B-3	ASCII Control codes for the ACI	B-12
B-4	ACI timeout values	B-13
C-1	Mechanical Maintenance	C-1
F-1	8086 Memory Segmentation	F-1
F-2	Motherboard input/output Allocation	F-2
F-3	Axis Control Registers	F-5
F-4	Interrupt Control	F-6
F-5	8086 Interrupt Vector Map	F-7
F-6	System Digital Output Useage	F-8
F-7	System Digital Input Useage	F-10
F-8	RAPL Parameter Table	F-13

FIGURE	TITLE	PAGE
1-1	SRS-M1 Small Industrial Robot System	1-1
1-2	SRS-M1 Robot Arm	1-2
1-3	RSC-M1A Robot System Controller	1-3
1-4	RSC-M1A Robot System Controller (Front Panel)	1-4
1-5	RSC-M1A Robot System Controller (Back Panel)	1-4
1-6	Teach Pendant	1-5
2-1	Workspace Elevation (Top View)	2-2
2-2	Workspace Elevation (Side View)	2-3
3-1	Robot Base Dimensions	3-2
3-2	Robot Gripper Mounting Dimensions	3-3
3-3	RSC-M1 Robot System Controller Dimensions	3-4
3-5	Video Terminal Cable pinout	3-6
3-6	Printer Cable pinout	3-6
3-8	Digital I/O Pinout Description	3-8
3-9	Digital I/O Pinout of Controller	3-9
3-10	Analog Input Pinout Description	3-10
3-11	Analog Input Pinout of Controller	3-10
3-12	Analog Input Connector pin location diagram	3-10
5-1	Homing Marks - Setup Position	5-2
5-2	Locations of Homing Marks on Robot	5-2
5-3	Homing Marks - Home Position	5-2
6-1	Robot Joints	6-1
6-2	Robot Joints - Range of Motion	6-2
6-3	Optical Encoder Schematic	6-3
6-4	Harmonic Drive	6-4
7-1	RSC-M1A Controller	7-1
7-2	Motherboard PCB	7-2
7-3	D.C. Amplifier Module	7-5
7-4	Servo Axis Card	7-7
7-5	I/O Termination Board - Viewed from inside the Controller	7-10
7-6	I/O Termination Board - Viewed from outside the Controller	7-10
8-1	Digital Closed Loop Control	8-7
8-2	Acceleration, Velocity, Motion Profiles	8-8
9-1	Robot position relative to Homing Bracket	9-2
A-1	Zero position	A-3
B-1	ACI Protocol	B-5
F-1	8254 Timer Configuration	F-4
F-2	System Input/Output Useage	F-11

1-3 RSC-M1A ROBOT SYSTEM CONTROLLER

The SRS-M1A Robot System is controlled by the CRS Plus RSC-M1A Robot System Controller.

The Standard RSC-M1A Robot System Controller consists of a 16-bit microprocessor based master controller, resident RAPL programming language, teach pendant, Six (6) D.C. Servo amplifiers, arm power supply, and five (5) servo axis cards.

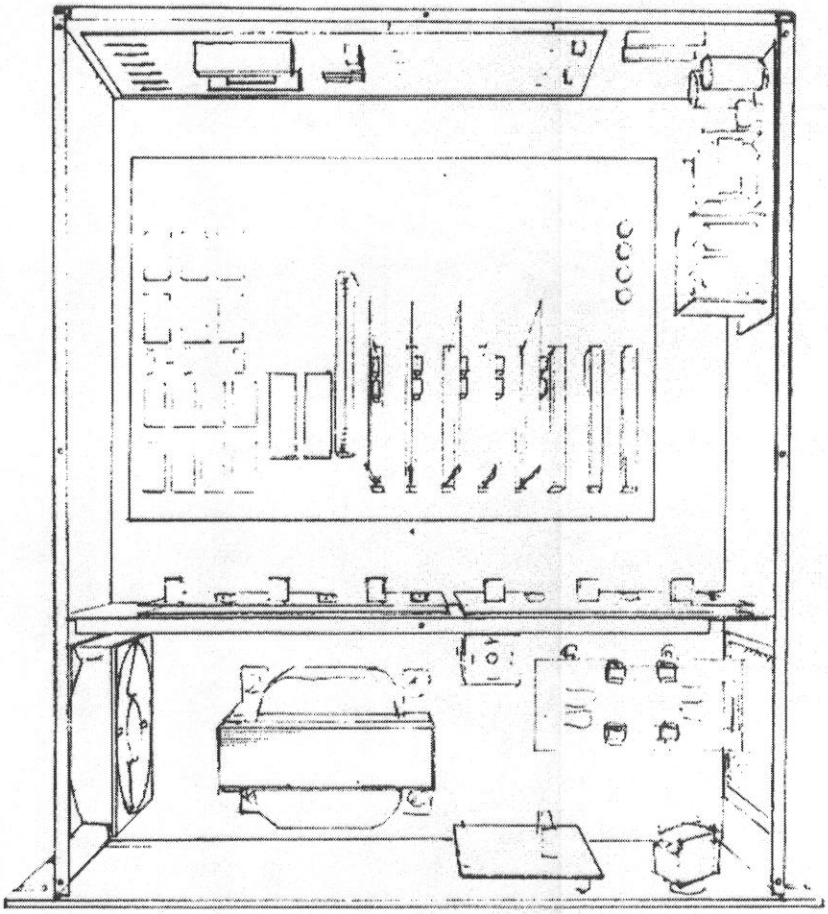


Figure 1-3 RSC-M1A Robot System Controller

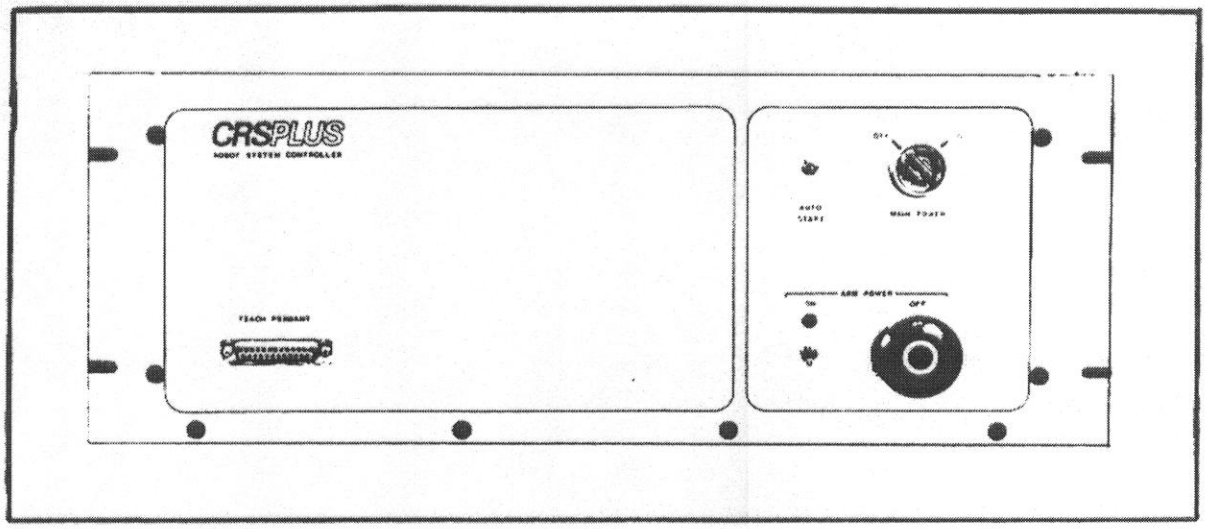


Figure 1-4 RSC-M1A Robot System Controller (Front Panel)

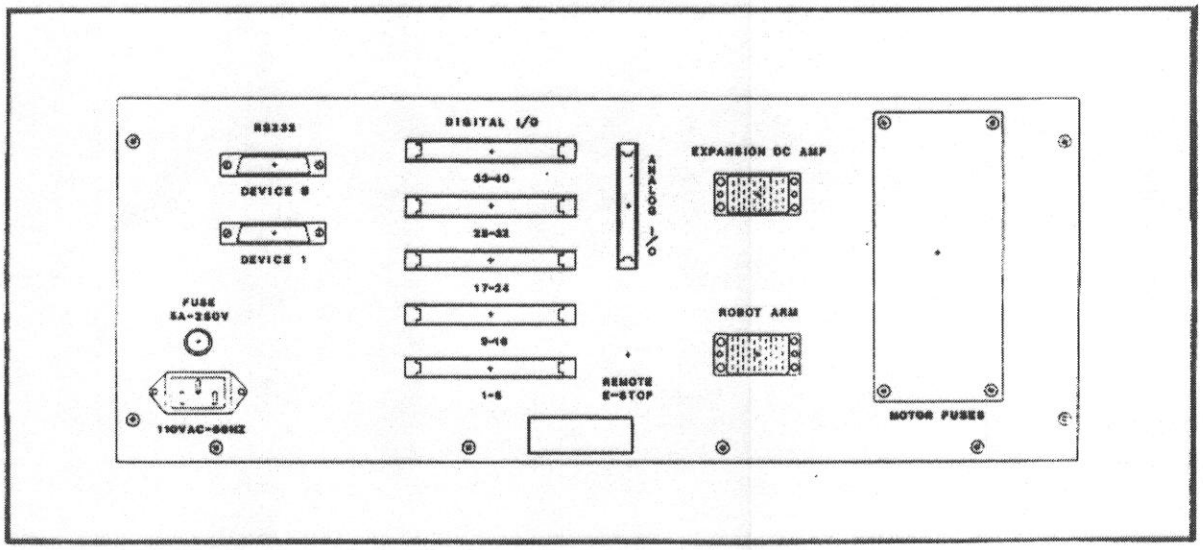


Figure 1-5 RSC-M1A Robot System Controller (Back Panel)

1-4 TEACH PENDANT

The RSC-M1A Robot System Controller can be manually controlled by the teach pendant.

The teach pendant provides manual controls for all 8 joints (Motion), Teaching points, aborting programs, and speed of motion. The teach pendant also provides an LED for displaying error status and gripper control.

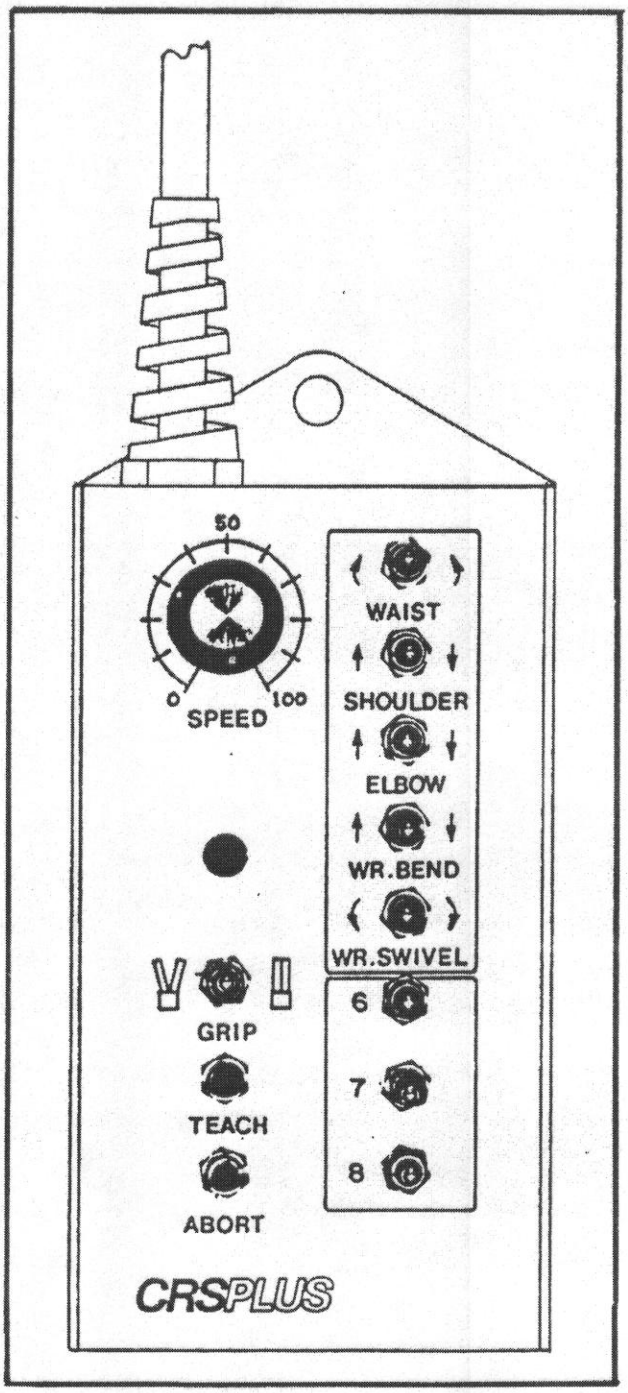


Figure 1-6 Teach Pendant

2-1 ROBOT ARM

Description	Specification
Structure	Articulated - Five DOF
Payload (Maximum Speed)	1.00 Kilogram
Payload (Reduced Speed)	2.00 Kilograms
Reach (Without Gripper)	0.56 Metres
Workspace Dimensions:	
Base Rotation	+/- 175 degrees
Shoulder rotation	+110, -0 degrees
Elbow Rotation	+0, -130 degrees
Wrist Bend	+/- 110 degrees
Tool Roll	+/- 180 degrees
Maximum Loaded Speed	0.50 Metres per second
Joint Speeds:	
Base	60 degrees per second
Shoulder	60 degrees per second
Elbow	60 degrees per second
Wrist	180 degrees per second
Tool	180 degrees per second
Repeatability	+/- 0.13 Millimetres
Joint Worst Case Resolution:	
Base	0.0023 inches
Shoulder	0.0023 inches
Elbow	0.0014 inches
Wrist	0.0013 inches
Tool	0.0013 inches
Drive System	DC Servo Motors with Optical encoders

Table 2-1 Robot Arm Specifications

2-2 WORKSPACE ELEVATION - TOP VIEW

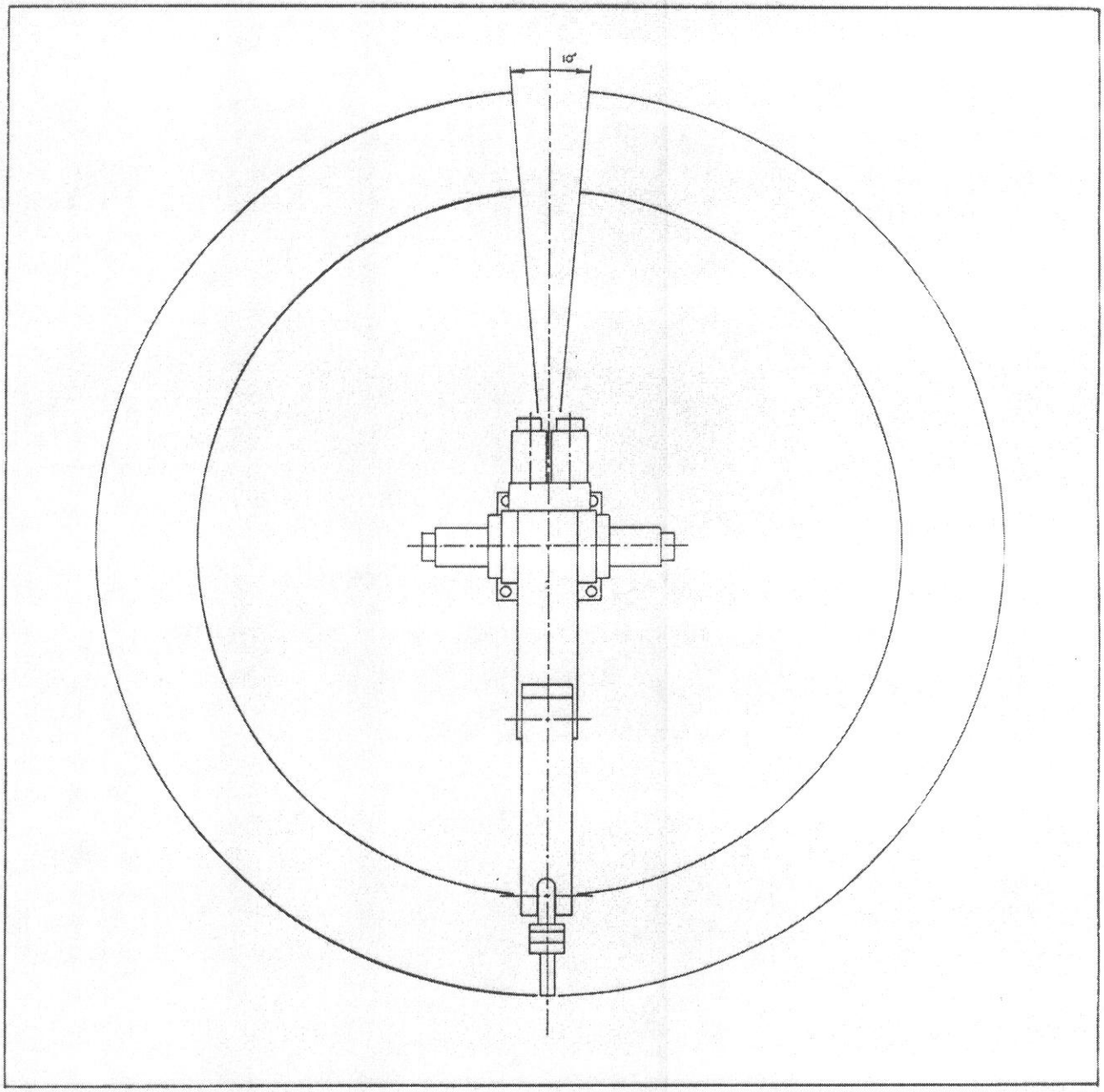


Figure 2-1 Workspace Elevation - Top View

2-3 WORKSPACE ELEVATION - SIDE VIEW

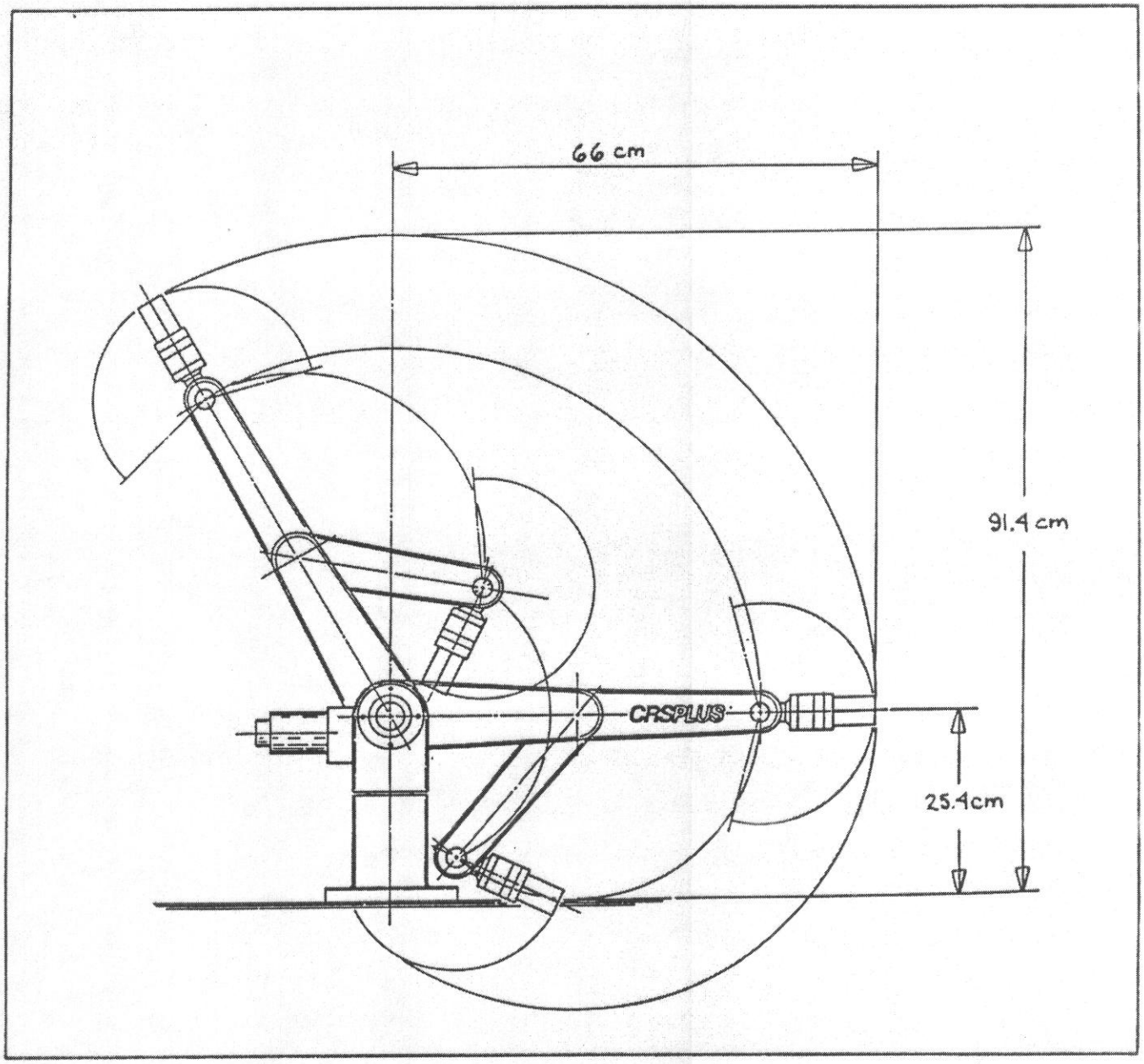


Figure 2-2 Workspace Elevation - Side View

2-4 RSC-M1A ROBOT SYSTEM CONTROLLER

Description	Specification
Control System:	
Microprocessor type	8086
Number of Axes	Five (5) DC Servo Axes
Teaching System:	
Manual	Teach Pendant
Language	RAPL
Off-Line	Optional ACI Interface
Motion:	
Point to Point	Yes - Joint Interpolated
Straight Line	Yes - Possible on all moves
Continuous Path	Yes - 8 axis spline interpolation
Position Detection	Digital Optical Encoders
Speed setting	0% to 150% Programmable
Interfaces:	
Communication	Dual RS-232C, ACI interface
User digital inputs	16 Standard, 40 Optional
User digital outputs	16 Standard, 40 Optional
Hardware Expansion	Peripheral Edge Connector
Programming:	
Language	RAPL
Editor	RAPL Line Editor
User Memory size	8 KByte Std, up to 64 K optional
Power Requirements	100-130 VAC, 50-60 Hz, 3A
Operating Ambient Temp.	0 to 50 degrees Celsius
Dimensions	19" wide x 21" deep x 7" high

Table 2-2 RSC-M1A Robot System Controller Specifications

2-5 TEACH PENDANT

8 Switches for Joint Control 0 to 100 % Speed Control
Abort Button Teach Button
Ready Light Indicator
10 foot cable

Table 2-3 Teach Pendant Specifications

3-1 UNPACKING AND INSPECTION

Upon receipt of your SRS-M1A Small Industrial Robot System check to ensure you have received all the components. See Table 3-1 for the components of the SRS-M1 robot system. If any components are missing contact your local distributor or CRS Plus immediately.

Description	Quantity
Robot Arm	1
Robot System Controller	1
Robot/Controller Cable	1
Teach Pendant	1
Power Cord	1
Spare Fuses (2A)	5
SRS-M1A Installation Manual	1
RAPL Users Guide	1

Table 3-1 SRS-M1A - System Components

3-2 ROBOT BASE INSTALLATION

Install the robot base in accordance to the following procedure:

1. Prepare a mounting platform. Mounting platform must be firmly fastened to floor to prevent vibration.

2. Drill holes in mounting platform using the dimensions shown in figure 3-1.

3. Mount robot base on the mounting platform.

4. Fasten robot base to mounting platform using 4 of a 3/8" diameter, 16 UNC Tap, 1-1/2" long bolts.

5. For future precise relocation use 1/4" diameter dowel pins as shown in figure 3-1.

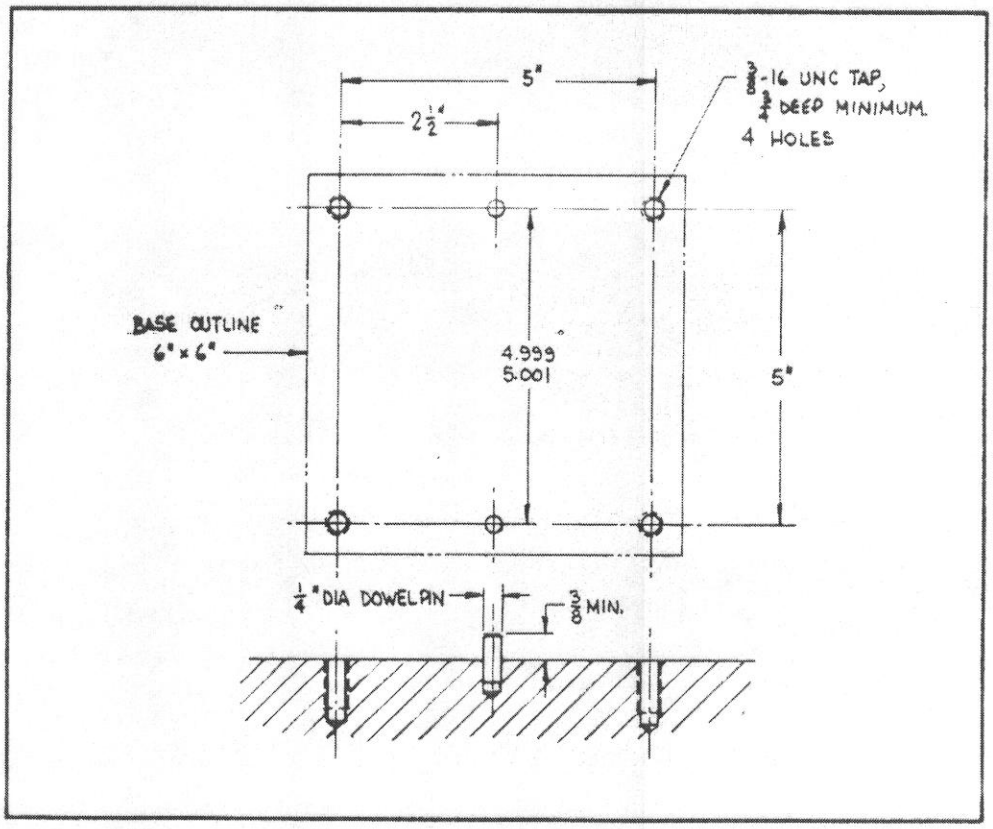


Figure 3-1 Robot Base Dimensions

3-3 GRIPPER INSTALLATION

Install the robot gripper in accordance to the following procedure:

1. Fasten gripper to robot flange using 4, #10 socket head cap screws as shown in figure 3-2.

2. Connect air hose from pneumatic gripper into barbed fittings on the right side of the robot arm. The upper connector is pressurized after executing the CLOSE command (normally open) while the lower one is pressurized after the OPEN command (normally closed). For use with the SRS-SGRIP option, plug the electrical connector from the servo gripper into the receptacle on the left side of the robot arm.

3. For pneumatic operation, connect air supply to the 1/8-NPT air port at the rear of the robot base.

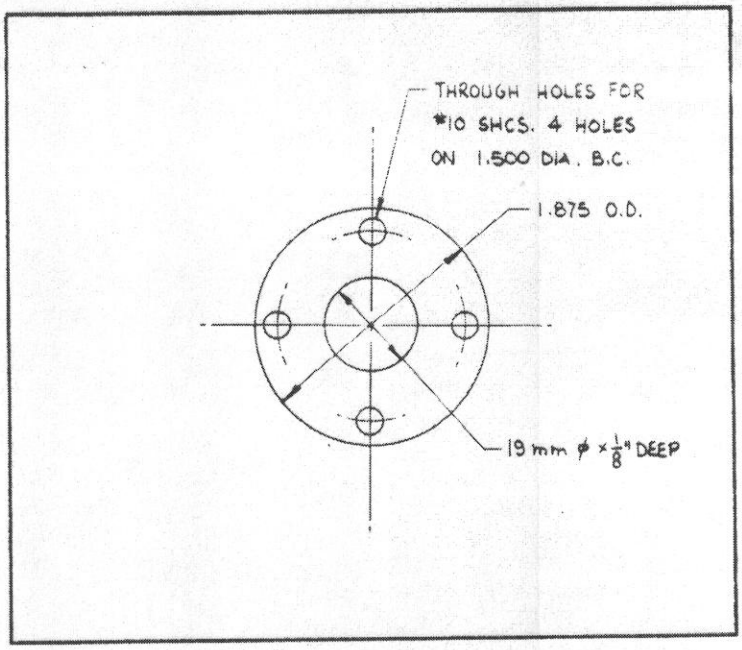


Figure 3-2 Robot Gripper Mounting Dimensions

3-4 RSC-M1A ROBOT SYSTEM CONTROLLER INSTALLATION

Install the robot system controller in accordance to the following procedure:

1. Mount Controller into a 19" rack.

2. Allow no less than a 1" clearance for air flow in front of the grills on the right and left side of the enclosure.

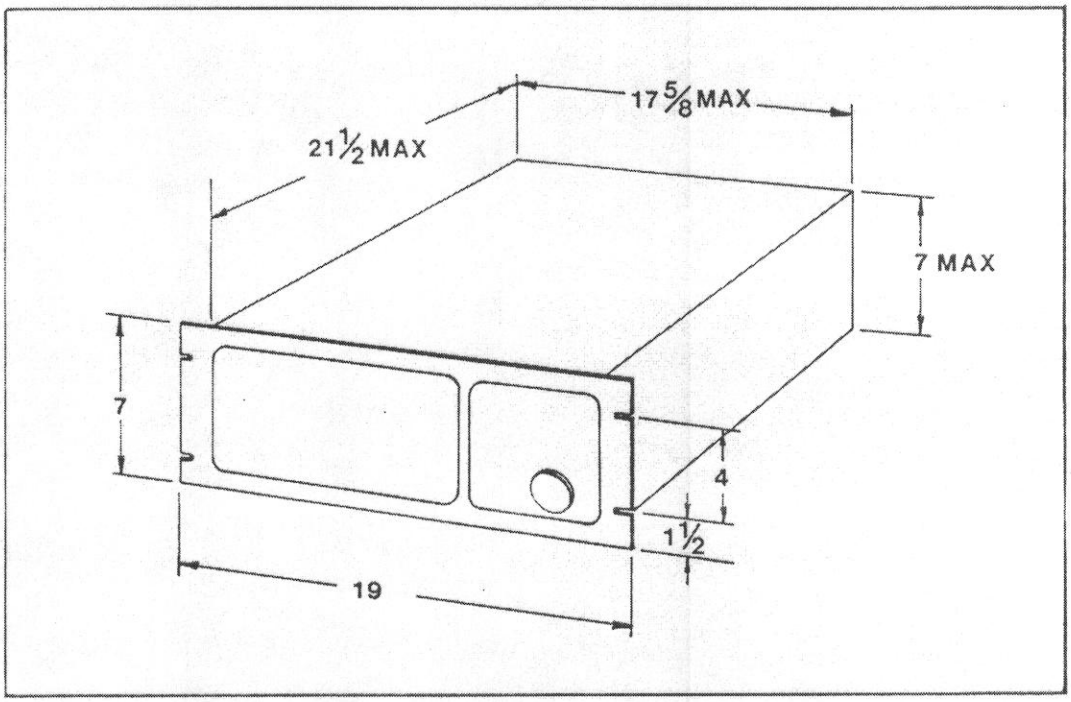


Figure 3-3 RSC-M1A Dimensions

3-5 CABLING

Install the robot cables in accordance to the following procedure:

1. Connect the robot system cable to the robot base. The connector is keyed to prevent incorrect orientation. The cable exits toward the back of robot. Once the connectors are mated correctly tighten screw firmly. Connect the other end of the cable to the rear panel of the RSC-M1A controller. Minimize the bending stress on the panel connector by strain relieving it to a solid object nearby.

2. Connect the teach pendant to the robot system controller front panel. The thumb screws in the cable may be used to provide a firm connection.

3. Connect your video terminal to the robot system controller serial port #1 on back panel.

4. Connect the robot system power cord.

5. Plug power cord into three prong outlet.

3-6 VIDEO TERMINAL INSTALLATION

Install the video terminal in accordance to the following procedure:

1. Connect cable to video terminal and controller with the pinouts shown in figure 3-5.

2. Set Video terminal to 9600 baud, Full Duplex, No Parity, and no auto line feed.

3. Install power cable to Video Terminal and turn on power.

Controller	Video Terminal
2 - Rxd	2 - Txd
3 - Txd	3 - Rxd
4 - CTS*	4 - RTS*
5 - RTS*	5 - CTS*
7 - GND	7 - GND

Figure 3-5 Video Terminal Cable pinout

3-7 PRINTER OR HOST COMPUTER INSTALLATION

Install the printer or host computer in accordance to the following procedure:

1. Connect cable to printer or host computer and controller with the pinouts shown in figure 3-6.

2. Set printer or computer RS-232 to 2400 baud, Full Duplex, No Parity, and XON/XOFF on.

Controller	Printer/Host Computer
2 - Rxd	2 - Txd
3 - Txd	3 - Rxd
4 - CTS*	4 - RTS*
5 - RTS*	5 - CTS*
	6 - DSR
7 - GND	7 - GND
	20 - DTR

Figure 3-6 Printer Cable pinout

Replace existing Tech. Manual pages with these two.

3- 8 DIGITAL I/O INSTALLATION

Install the SRS-DB/16R or equivalent in accordance to the following procedure:

1. Connect cable to digital I/O buffer and controller with the pinouts shown in Figure 3-8 and 3-9.

2. If not using the SRS-DB/16R contact your distributor for information on compatibility with other isolators or systems. Use Ansley 609-5030 connector for connection to controller.

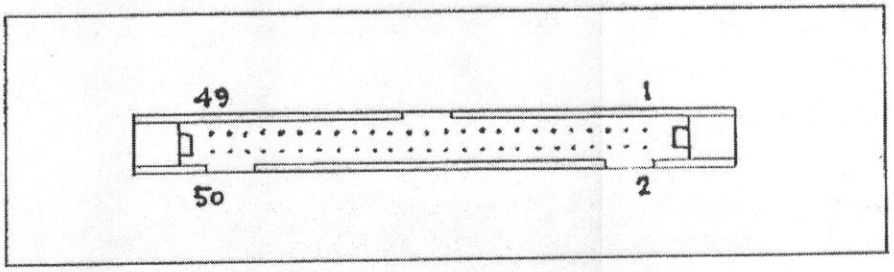


Figure 3-8 Pin orientation for each digital I/O connector

Pin	Description	Pin	Description	Pin	Description
2-50	GROUND	2-50	GROUND	2-50	GROUND
1,49	Vcc	1,49	Vcc	1,49	Vcc
47	INPUT 1	47	INPUT 9	47	INPUT 17
45	INPUT 2	45	INPUT 10	45	INPUT 18
43	INPUT 3	43	INPUT 11	43	INPUT 19
41	INPUT 4	41	INPUT 12	41	INPUT 20
39	INPUT 5	39	INPUT 13	39	INPUT 21
37	INPUT 6	37	INPUT 14	37	INPUT 22
35	INPUT 7	35	INPUT 15	35	INPUT 23
33	INPUT 8	33	INPUT 16	33	INPUT 24
31	OUTPUT 1	31	OUTPUT 9	31	OUTPUT 17
29	OUTPUT 2	29	OUTPUT 10	29	OUTPUT 18
27	OUTPUT 3	27	OUTPUT 11	27	OUTPUT 19
25	OUTPUT 4	25	OUTPUT 12	25	OUTPUT 20
23	OUTPUT 5	23	OUTPUT 13	23	OUTPUT 21
21	OUTPUT 6	21	OUTPUT 14	21	OUTPUT 22
19	OUTPUT 7	19	OUTPUT 15	19	OUTPUT 23
17	OUTPUT 8	17	OUTPUT 16	17	OUTPUT 24
Connector 1-8		Connector 9-16		Connector 17-24	

Pin	Description	Pin	Description
2-50	GROUND	2-50	GROUND
1,49	Vcc	1,49	Vcc
47	INPUT 25	47	INPUT 33
45	INPUT 26	45	INPUT 34
43	INPUT 27	43	INPUT 35
41	INPUT 28	41	INPUT 36
39	INPUT 29	39	INPUT 37
37	INPUT 30	37	INPUT 38
35	INPUT 31	35	INPUT 39
33	INPUT 32	33	INPUT 44
31	OUTPUT 25	31	OUTPUT 33
29	OUTPUT 26	29	OUTPUT 34
27	OUTPUT 27	27	OUTPUT 35
25	OUTPUT 28	25	OUTPUT 36
23	OUTPUT 29	23	OUTPUT 37
21	OUTPUT 30	21	OUTPUT 38
19	OUTPUT 31	19	OUTPUT 39
17	OUTPUT 32	17	OUTPUT 40
Connector 25-32		Connector 33-40	

Figure 3-9 Digital I/O Pinout Description

3-9 ANALOG INPUT INSTALLATION

Install the SRS-AI/8 in accordance to the following procedure:

1. Install optional piggy-back board to motherboard in socket shown in figure 7-2. Connect analog inputs to rear panel of the controller using Ansley 609-3430 or equivalent. Pinout for analog connection is shown in figure 3-10. Additional analog I/O is available with the optional SRS-COMBI expansion card. Figure 3-11 shows the pinout for the remainder of the analog I/O connector if this card is installed.

Pin	Description
1	GROUND
2	GROUND
3	ANALOG 1
4	ANALOG 2
5	ANALOG 3
6	ANALOG 4
7	ANALOG 5
8	ANALOG 6
9	ANALOG 7
10	ANALOG 8

FIGURE 3-10 Analog Input pin description

Pin	Description	Pin	Description
11	GROUND	22	ANALOG IN 15
12	GROUND	23	ANALOG IN 16
		24	ANALOG IN 17
13	ANALOG OUT 2	25	ANALOG IN 18
14	ANALOG OUT 1	26	ANALOG IN 19
		27	ANALOG IN 20
15	GROUND	28	ANALOG IN 21
16	GROUND	29	ANALOG IN 22
		30	ANALOG IN 23
17	ANALOG IN 10	31	ANALOG IN 24
18	ANALOG IN 11	32	ANALOG IN 25
19	ANALOG IN 12		
20	ANALOG IN 13	33	GROUND
21	ANALOG IN 14	34	GROUND

FIGURE 3-11 Analog Input pin description with optional SRS-COMBO expansion card.

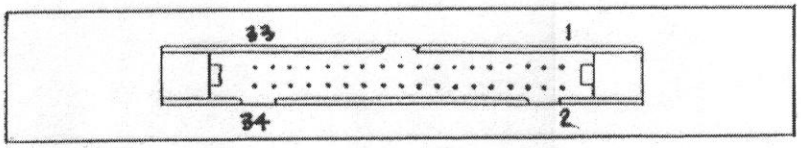


FIGURE 3-12 Analog input connector pin location diagram.

4-1 SAFETY

The SRS-M1A Small Industrial Robot System is capable of fast movement without warning. To prevent personal injury never enter the workspace while robot is under program control.

The RSC-M1A Robot System Controller can cause injury if cover is removed to expose the high voltage inside.

4-2 HANDLING OF ROBOT

Handle the robot with care. The robot is a precise piece of machinery and should be treated with care. When moving the arm, be careful not to apply pressure to any wiring or to the motor encoders. At all times, be careful not to allow metal objects or liquid into the robot arm or controller.

Use caution when applying power to the RSC-M1A Robot System Controller, do not exceed the rated power shown in Table 2-2.

4-3 POWER FAIL

Upon a power failure the RSC-M1A Robot System Controller will automatically remove arm power to prevent robot motion on re-application of power. When a power failure occurs it is neccessary to re-start the robot. Follow the directions in Chapter 5 for the start-up procedure.

5-1 POWER UP SEQUENCE

1. Turn on Video Terminal.

2. Turn on Robot system controller.

3. Manually move robot into the upright position and hold in position.

4. Turn on arm power.

5. Let go of arm.

6. You are now ready for the Homing sequence.

5-2 HOMING SEQUENCE

1. Enter <MANUAL> at keyboard to enter manual mode.

2. Starting with Joint #1, move all five joints to the homing mark positions shown in Figure 5-2. Position each joint such that the homing marks are as shown in Figure 5-1 (arrow point is within the homing range).

3. Enter <HOME> at keyboard. The controller will ask you to confirm that you have moved the robot arm into the starting home range.

4. If robot arm is in starting home range then enter <YES> at keyboard.

5. The controller will move each joint into the home position and display on your screen the status of each joint homing. If a error occurs in any of the joints return to step (1) and repeat complete procedure. If a error occurs in three (3) consecutive attempts contact your local distributor or CRS Plus.

6. Visually inspect each homing marker. Markers should point directly at each other as shown in Figure 5-3. If the arm did not home correctly return to step (1) and repeat complete procedure.

7. The arm did home correctly and you are now ready to start developing and executing programs.

5-2 HOMING SEQUENCE (Con't)

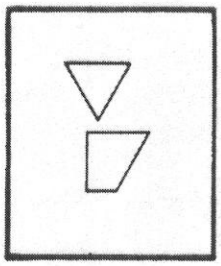


Figure 5-1 Homing Marks - Setup Position

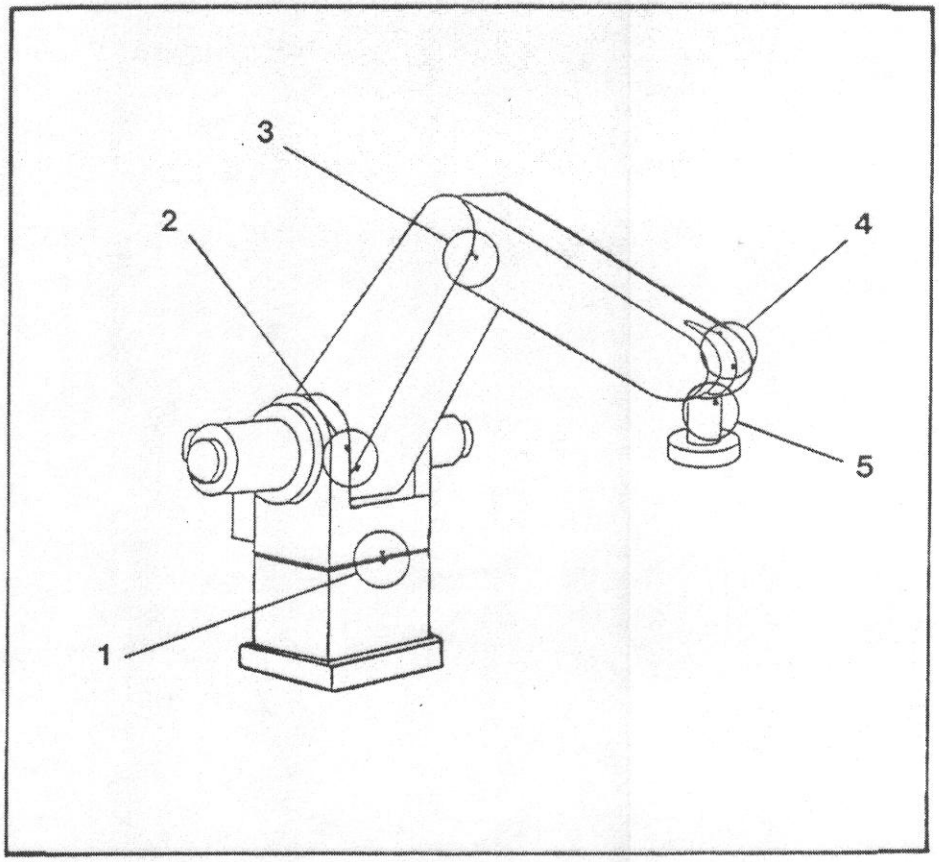


Figure 5-2 Locations of Homing Marks on robot arm

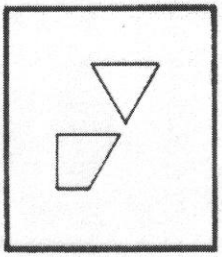


Figure 5-3 Homing Marks - Home Position

6-3 DRIVES AND TRANSMISSIONS (Con't)

Adjustment

Harmonic Drives are not adjustable. The chain drive of joint three may at times need attention. It can be adjusted by means of a chain tensioner at the rear of link one. See APPENDIX C for a detailed description of this procedure.

The wrist drive components are all adjustable in case some play should develop in the wrist. This adjustment should be done only by a qualified technician. Re-lubrication would be needed after adjustment (see APPENDIX C).

Preload in the chain transmission can be adjusted as described in APPENDIX C.

Adjustment of the wrist bevel gears for joint 5 is a difficult procedure and should be only done by a qualified technician. Briefly the procedure is as follows: remove the upper arm cover, remove the link 2 wrist drive chains, remove the wrist from the arm, loosen the set screw holding the wrist cross-shaft in place, slide the cross shaft in or out to alter gear mesh, retighten set screw, reassemble wrist. After reassembly, the robot must be re-calibrated.

7-1 INTRODUCTION

This section describes the RSC-M1A controller hardware as shown in Figure 7-1. Table 7-1 describes the various modules.

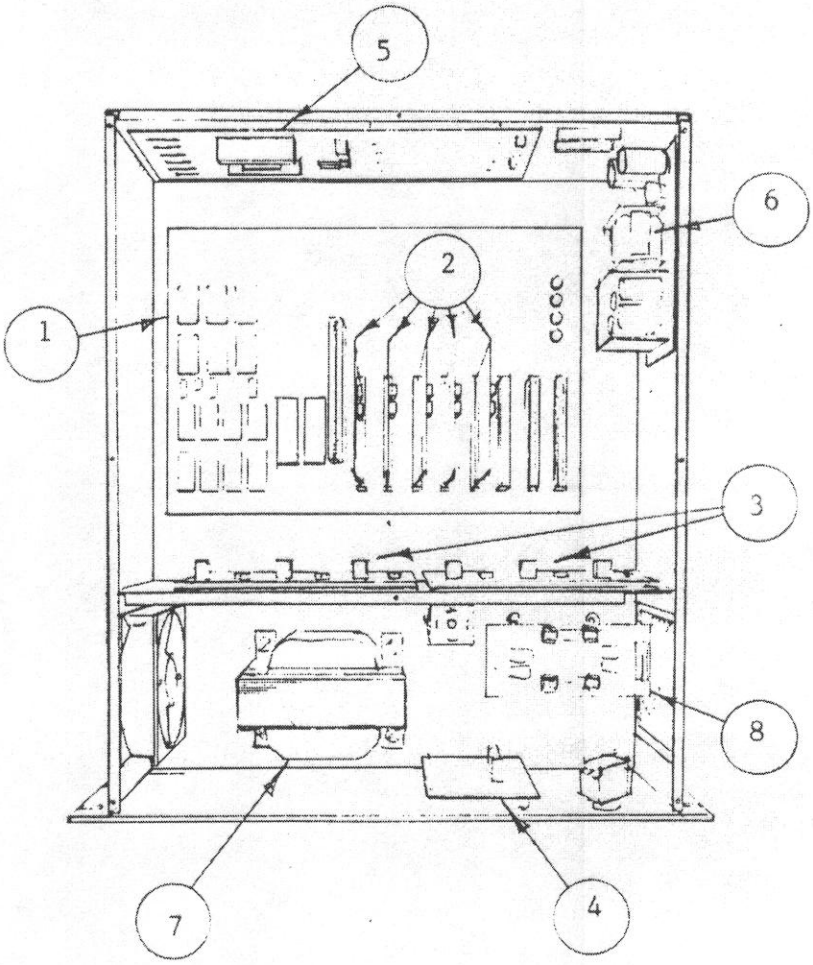


Figure 7-1 RSC-M1A Controller

Item	Quantity	Module Number	Description
1	1	SEC-9-001	Mother Board
2	5	SEC-9-002	Axes cards
3	2	SEC-9-005	D.C. Amplifier boards
4	1	SEC-9-007	Arm Power Relay board
5	1	SEC-9-004	Digital I/O Connection board
6	1	SEC-9-008A	Computer Power Supply
7	1	SEC-9-006A	Arm Power Supply
8	1	SEC-9-010	Arm Power Filter Board

Table 7-1 RSC-M1A Controller Desciption

7-2 MOTHERBOARD

The Motherboard as shown in figure 7-2, is the largest printed circuit board (P.C.B.) in the CRS RSC-M1A controller. The Motherboard has all the needed functions to operate as a computer control system. Table 7-2 describes the Motherboard connectors locations.

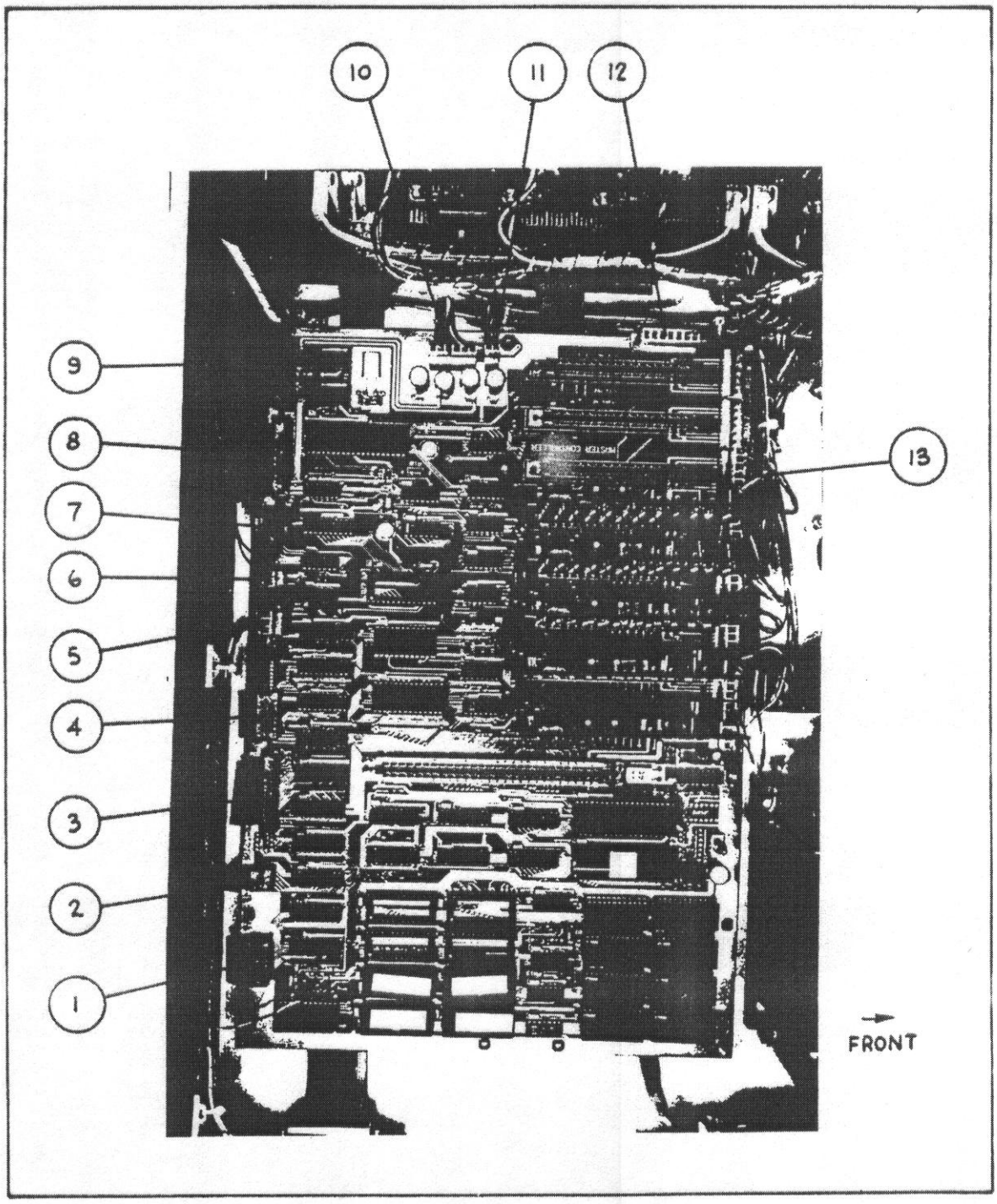


Figure 7-2 Motherboard P.C.B.

7-2 MOTHERBOARD (Con't)

ITEM	FUNCTION
1	18 pin dip for digital I/O numbers 9 to 16.
2	8 pin dip for the arm power relay board.
3	26 pin header for the teach pendant connection.
4	18 pin dip for digital I/O numbers 1 to 8.
5	3 pin connector for the auto-start, arm power, and ground.
6	2 pin connector for the gripper.
7	18 pin dip for the optional analog input.
8	26 pin header for serial I/O channel 1.
9	26 pin header for serial I/O channel 0.
10	6 pin connector for +5VDC, +12VDC, -12VDC, and ground.
11	2 pin connector for the power supply to the encoders.
12	8 pin connector for common ground.
13	Eight 6-pin connectors used for the encoder feedback, and velocity command for each axis.

Table 7-2 Controller Connector Map

The Motherboard contains the following:

1) The 8086-2 16-bit microprocessor and the 8087-2 math co-processor running at a master clock frequency of 7.33 MHz.

2) A programable interrupt controller which is used to support the multi-task operating system. It has 8 levels of interrupts which are assigned as follows:
   - Level 0 is used for the encoders zero crossing signal, used only during homing the robot.
   - Level 1 is used for closing the position loop every 3.8 mSec.
   - Level 2 is used for generating the positional command every 3.8 mSec in the run mode, and 15 mSec. in the manual mode.
   - Level 3 is used for scanning the digital I/O every 40 mSec.
   - Level 4 is to support the TEACH button on the teach pendant.
   - Level 5 is to support the ABORT button on the teach pendant.
   - Level 6 is to support the serial channel 0.
   - Level 7 is to support the serial channel 1.

3) A programable interval timer (PIT) is used to generate fixed time intervals. The PIT has 3 timers which are used as follow:
   - Timer 0 generates the time base for the position loop.
   - Timer 1 generates the time base for the command generation.
   - Timer 2 generates the time base for the I/O scanning.

4) A Dual Asynchronous Receiver/Transmitter (DART) which controls two serial cannels with the RS-232 format having a programmable baud rate of between 50 and 19,200 bps.

7-2 MOTHERBOARD (Con't)

The memory section has up to 128 Kbyte of Read Only Memory which contains the RAPL executive program. 16 Kbyte of Random Access Memory (Expandable to 64 Kbyte) is provided for the user programing and computer "scratchpad" space. The RAM chips have an integral lithium energy source which will retain the memory contents for up to 10 years.

It also has eight (8) axis slots, five of which are used for the SRS-M1A robot axes. The other 3 may be used for extra servo axes including a servo gripper. It also has one peripheral slot which contains the local microprocessor bus and is intended for future system expansion.

The mother board has a 28-pin dip socket for the eight optional analog inputs.

7-3 DC AMPLIFIER MODULE

The D.C. Amplifier Module as shown in Figure 7-3 contains three (3) separate amplifiers, each supplied with +/- 26 VDC. The signal to each amplifier is +/- 10 VDC. The output is the motor voltage of +/- 20 VDC. at 2 amperes each.

Each module has an adjustable gain in the range of x1 to x2. It has no phase reversal and a low crossover distortion.

Table 7-3 refers to figure 7-3 which shows the features of the Amplifier module:

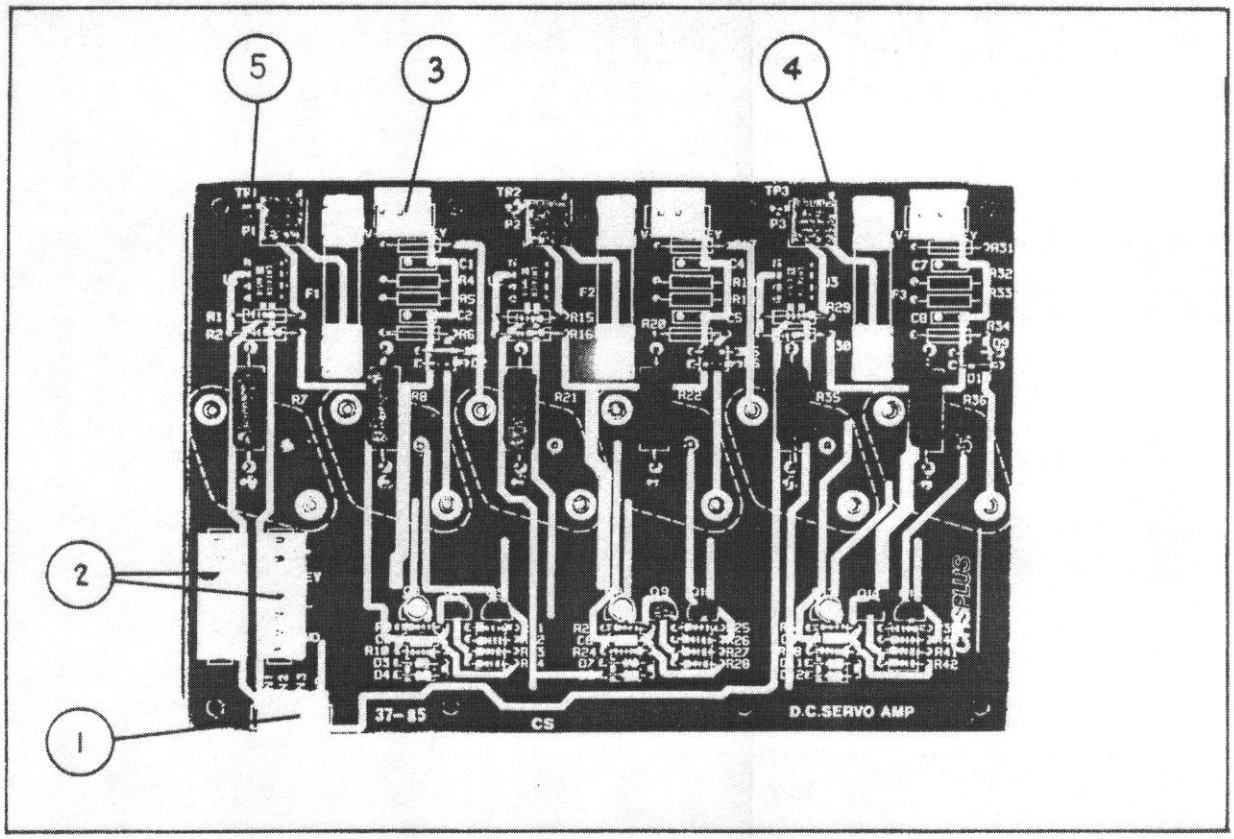


Figure 7-3 D.C. Amplifier Module

7-3 DC AMPLIFIER MODULE (Con't)

ITEM	FUNCTION
1	4 pin connector for the command signals connecting the servo axis output to the amplifier module. The signal level is +/- 10 VDC at 70 ma. The pin-out configuration of this connector is: * * * * IN1 IN2 IN3 GND.
2	Two 7 pin connectors for the power source are used to connect the amplifier module to the voltage regulator, carrying +/- 20 VDC at 10 amps. The pin-out of this connector is as shown below: * * O * * * * + V + V KEY - V - V GND GND
3	Three 3-pin connectors for the motor power output connecting the amplifier module to the D.C. servo motors. The pin-out is as shown below: * * O +VM -VM KEY
4	Three 20K, 20-turn trim pots. The amplifier gain is increased by turning the adjusting screw CCW.
5	Three test points, TP1, TP2, and TP3. The voltages at the each test point corresponds to the motor armature voltage for the respective channel.

Table 7-3 D.C. Amplifier Description

7-4 SERVO AXIS CARDS

The Servo Axis cards as shown in figure 7-4 contain the necessary functions to close the position loop. It takes an incremental digital command from the mother board, converts it to an analog voltage, and send it to the DC Amplifier. It also takes the feedback signal from the incremental optical encoder in a square wave form (channel A, channel B, and a zero crossing index signal). The signal is then shaped and converted to a pulse train, with pulse width of 1 uSec (+/-30%).

Table 7-4 refers to figure 7-4 which shows the features of the Servo Axis Card module:

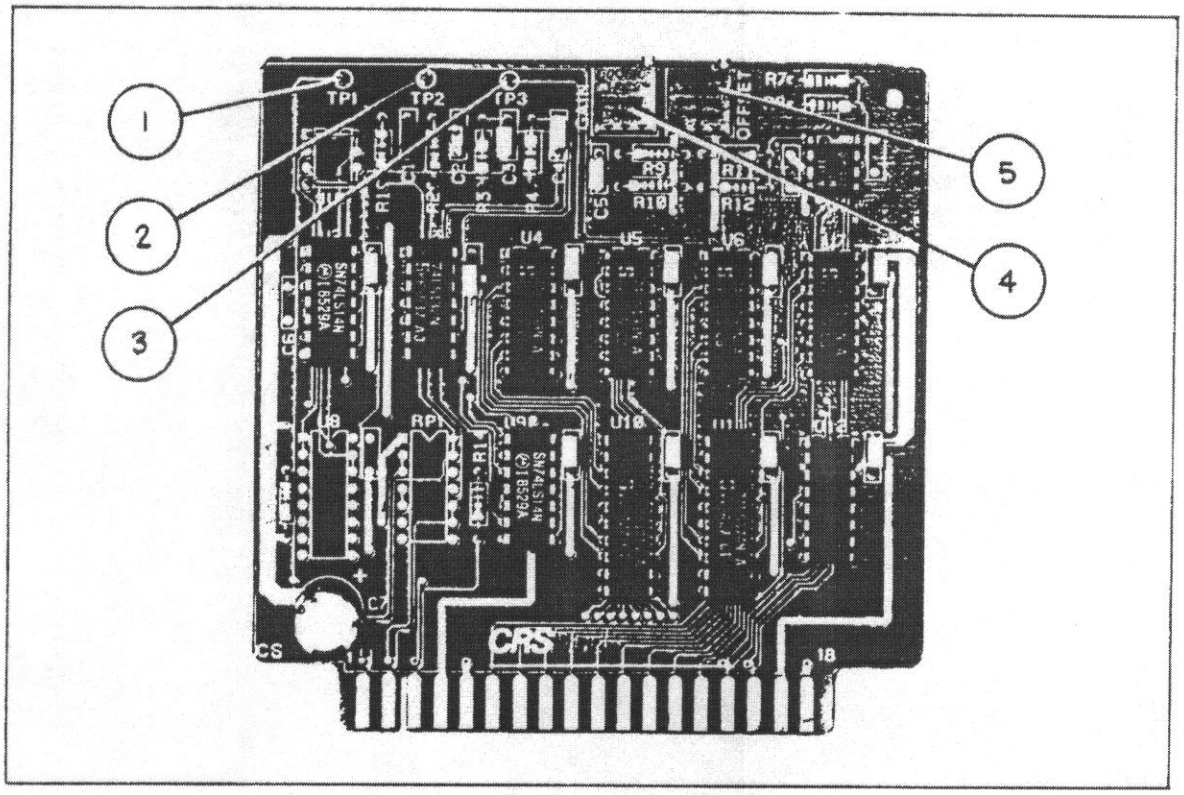


Figure 7-4 Servo Axis Card

7-4 SERVO AXIS CARDS (Con't)

ITEM	FUNCTION
1	TP1 - the analog command voltage. Should be in the range of +/- 10VDC.
2	TP2 - the pulse train for the CW motion of the servo motor. The signal at this test point has a pulse width of 1 uSec (+/-30%) and a period depending on the speed of the servo motor.
3	TP3 - the pulse train for the CCW motion of the servo motor, with the same signal characterstec as TP2.
4	20K, 20 turn gain adjustment potentiometer. Turning the adjusting screw CCW will increase the gain of the output stage.
5	20k, 20 turn Offset adjustment potentiometer. Turning the adjusting screw CW will give a positive voltage offset to the output.

Table 7-4 Servo Axis Card Description

7-5 ARM POWER RELAY BOARD

This board contains the arm power relay, arm power switch, AC power failure signal, arm power "watch-dog" circuitry, and the Auto Start switch. The relay switches the 110/220 VAC to the arm power supply. The function of the relay will only be enabled by the computer if the controller is functioning normally. The continuing safe operation of the computer is checked by the "watch-dog" timer.

The board connects to the mother board and the I/O termination board via two 5-pin and one two-pin connectors. One five-pin connector contains the power supply to the board (+12, +5 VDC and Ground), and the ARM-ON signal to the I/O termination board (this signal goes to the external DC AMP connector). The second contains the power fail signal, AUTO-START signal, Arm Enable, and the Arm On signal to the mother board.

The two-pin connector is used for the remote E-Stop connector on the rear panel.

The following is the pinout for these connectors:

Connector D:

*	*	*	o	*
+12VDC	+5VDC	GND	KEY	Arm power on

Connector E:

*	*	*	*	o
Arm Power	PFI	Auto Start	Arm Ena	KEY

The two E-Stop pins are connected to the 'Arm Power OFF' push button on the front panel. The 'Arm-Power-On' is a signal to indicate to the mother board the status of the arm power (on/off). The Auto Start is used in conjunction with the main power switch on the front panel to execute an AUTO_ST program on power-up.

7-6 I/O TERMINATION BOARD

The I/O termination board is shown in figure 7-6. This board is used to connect the controller to the all external system components. It has connectors for up to 40 digital inputs and 40 digital outputs at TTL levels un-buffered. It also contains the connectors for the robot arm and the external amplifier, the remote E-Stop switch, and the analog I/O. In addition to connectors, it contains the fuse holders for the arm power and servo gripper power fuses.

On the inside of the board at the top (see figure 7-5) there is a header block which is used to select the function of the eighth slot of the mother board. With the header inserted one way, signals from the slot go to and from the servo gripper in the robot arm. Installed in the other way, signals go to the third channel of the external DC amplifier (SRS-DCAMP).

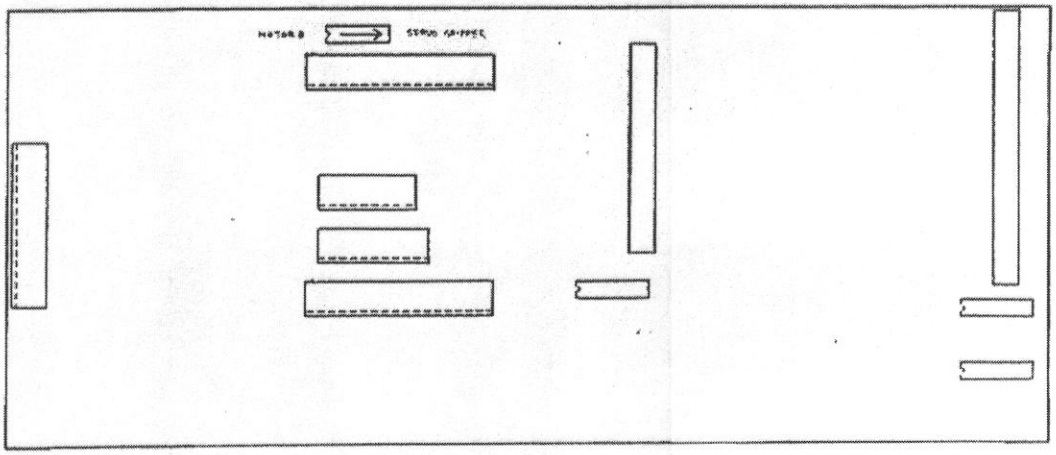


Figure 7-5 I/O Termination Board - Viewed from the inside of the controller box.

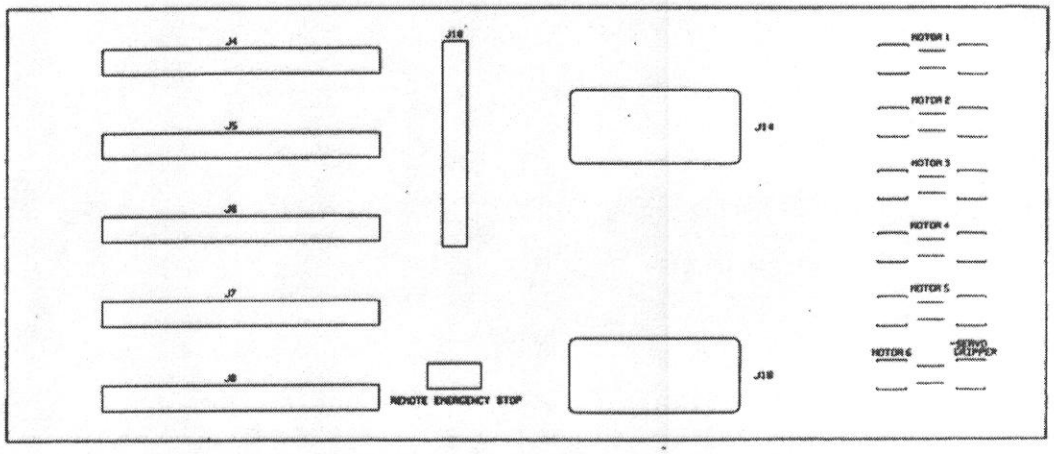


Figure 7-6 I/O Termination Board - Viewed from the outside of the controller box.

7-7 COMPUTER POWER SUPPLY

The computer power supply is a switching mode supply with three output voltages:

+5 VDC at 8.0 A.
+12 VDC at 2.5 A.
-12 VDC at 1.0 A.

The power supply has output overvoltage clamping, and output short circuit protection.

7-8 ARM POWER SUPPLY

The arm power supply consists of a step_down transformer (36 VAC CT. at 13 A), a bridge rectifier, and two filter capacitors. The output of this power supply is +/- 26 VDC at 13 A, and is unregulated.

8-0 INTRODUCTION

This section of the manual is intended for those that are familiar with Intel 16 bit programming practices. Memory elements defined here are defined as Intel memory items and the definitions of these can be found in the appropriate Intel literature.

System parameters are used to change the robots operating characteristics and to observe what the robot is doing at any particular time.

8-1 MEMORY MANAGEMENT

The CRS Robot System Controller uses the Intel 8086 segmented architecture scheme. The Controller contains three different physical types of memory;

1. Four (4) Kilobytes of Low Power CMOS memory:
   Used for Scratchpad use, 8086 stack space, and the interrupt vector space.

2. Twelve (12) Kilobytes of Battery-backed CMOS memory:
   Used for System Parameter setup and user memory space (about eight Kilobytes). This memory can be expanded to 64 Kilobytes.

3. One hundred and twenty-eight (128) Kilobytes of EPROM memory:
   Used for firmware requirements. This memory can be expanded to 256 Kilobytes depending on future software requirements.

All robot memory accesses use checksums to ensure data integrity.

8-2 COMPUTATIONAL ACCURACY

The Robot System Controller software utilizes the Intel 8087 Math co-processor fully for all mathematical calculations. Real numbers are stored in memory as four (4) byte quantities, giving 9 digit precision. Where possible, calculations are made using the full eighty (80) bit (10 byte) capability of the 8087 device, yielding a more accurate result.

8-3 PATH CONTROL - CLOSED LOOP CONTROL

The CRS Robot Controller has complete closed loop DC Servo Control for up to eight (8) axes. Gains can be programmable for a wide range of motion requirements. The closed loop control is achieved in approximately four (4) milliseconds. The control loop consists mainly of a proportional gain which acts on the positional error of each joint of the robot. This error is the difference between the commanded and the actual position of the joint.

Cartesian coordinates are transformed to motor pulse coordinates so that the robot commands can be compared to its actual position for closed loop control.

8-4 PATH CONTROL - COMMAND GENERATION

The robot has three modes of path control; Joint Interpolated, Straight line, and continuous path.

Joint Interpolated

The Joint Interpolated motion guarantees that all joints to be moved will start and stop together. It also guarantees that maximum acceleration and decceleration will not be exceeded for each axis. Command updates occur at four milli-second intervals. Figure 8-2 shows typical acceleration, velocity and position profiles for a joint interpolated robot motion.

Straight Line

In Straight Line, command updates are performed at a slower pace (30 milli-second update time). Acceleration and decceleration in this mode are based on world coordinates instead of individual joints. If wrist motionsare required during the straight line path, then the wrist rotational speeds are continuous throughout the straight line path segment. Joint limit checking is performed during the straight line move, and if any joint exceeds the programmed limits, the motion is immediately terminated, and an error message is displayed.

Continuous Path

The continuous path algorithm utilizes a cubic spline technique which will join the path points, called 'knots' in such a way as to make the velocity of the joints adjust smoothly between paths. The result of this is a motion which stops only at the end of the path. The algorithm will determine the parameters of the paths so that the robot moves through each of the knots. It does not relate to the programmer exactly how the ensuing path will appear. It is up to the programmer to provide sufficient points to define the path if precise path control is required. For straight line motions, it is possible to define cartesian points using RAPL commands (ie., SHIFT or SHIFTA) to provide the knots that will then be introduced into the continuous path algorithm.

The cubic spline algorithm requires the storage of path parameters for the continuous path. These parameters must be maintained until the path has been completed.

The storage requirements for a continuous path are calculated here;

B is the number of bytes required for parameter storage,
K is the number of knots which define the path, and,
A is the number of axes that have been programmed.

$$B = 4*((A+1)*K + K)$$

The RAPL controller utilizes the remaining space in the program buffer as the memory required to maintain this data. Once the path is in progress, no operation that modifies this data area can be executed, or improper operation will result.

It is recommended that the expansion memory option be provided so that larger

paths can be programmed, and more memory can be made available for program storage. To illustrate the memory requirements for the continuous path, up to 256 knots can be programmed using the TPATH command. This would require memory of 10K bytes for a full 8 axis implementation, according to the equation provided previously.

Due to the mathematics of the alogorithm, the programmer must not provide locations that are identical as consecutive elements of the path list. This will cause a math error and processing of the path parameters will stop. The following statement will cause an error.

```
10 CPATH A,B,B,C
```

since the point **B** is used consecutively in the list.

Due to the processing time involved for the path algorithm, vibration may be introduced into the arm during the execution of certain paths. In this case, the gain setting for all joints may be set to approximately 0.5. This results in some loss of path accuracy, but provides a quieter path motion.

For a large number of knots, the processing time of the continuous path parameters can be noticeable. This is not typically a detriment if good programming practices are adhered to. Firstly, it is typical to arrive at a point and wait for either the gripper to close, or some other control action to take place. This delay can be spent preparing the parameters for the next move. Also, while a non-continuous path is being executed, the next continuous path can be calculated.

8-5 INPUT/OUTPUT SCANNING

Digital input/output scanning takes place at approximately 40 milli-second intervals. This is the same interval used for the system clock. The 'ONSIG' input detection and the arm power check occurs at this rate as well.

8-6 MEMORY ALLOCATION

The robot controller has two types of on-line memory. The EPROM memory contains the system firmware, and is up to 256 Kbytes in capacity. The RAM memory is 16 Kbytes long, and is expandable to 64 Kbytes.

The firmware consists of two basic sections. It includes a monitor level structure which defines the hardware of the system. On top of this level, the operating system resides. This is what the programmer sees. It handles the execution of RAPL, along with the generation of robot path coordinates for motion control.

The RAM memory itself is divided into several functional areas. These are described in Table 8-1. Most importantly, its uppermost segment is allotted to the user. It is therefore called User Memory. It is this section of memory which contains user programs, variable storage and location storage. The exact size of user memory depends on the version of RAPL that you are using, since user memory is automatically mapped into whatever space is not required to run the operating system.

Command functions like ALLOCATE and NEW alter the contents of the user memory. These functions are described in detail in the RAPL Programming Manual.

The User Memory is further divided into four main areas. These areas are used to store programs, locations, variables and lastly a reserved area. Each of these areas store data in different structures.

Since the three main types of user memory can be dynamically allocated by the user, the system programmer must access the pointers to these items in memory.

Program Memory

Program storage consists of a program table, which contains the program name, the index into the program buffer where the program resides, and a checksum which is used to confirm the validity of each program's contents before a program is executed. The program buffer is a contiguous byte array which holds all of the user programs.

The program buffer stores programs as sequential ASCII characters. Programs information is either typed through the user terminal, or data which can be downloaded from a Master computer using the ACI interface and support software. Each program must end with an ASCII '$' character as a delimiter. Needless to say, the RAPL editor does not allow the entry of this character from the keyboard. It is up to the system programmer not to allow this character to be downloaded from a host computer as it will confuse the RAPL Program Manager.

Each program is given a spot in the program table, which can fit up to 255 programs. Each robot program description has three fields. The first field is an 8 byte character string which contains the program name. The second field is a word quantity which contains the index in the program buffer to the start of that program. The third field is a word quantity which contains the program checksum. The length of the program buffer is set by the ALLOCATE command. The program buffer is then used by potentially 255 different programs. When a new program is created, there must be space in the program table (ie., there can't

be more than 255 programs at once), and, there must be room left in the program buffer (ie. if all the program memory is used by one program, you will get a buffer full error if you create another program).

Variable Storage

Variables are located in the symbol table. The symbol table has a length defined by the ALLOCATE command. Like the program table, there are three fields for each symbol entry in the symbol table. The first is the eight character string used to store the name. The second is a four byte location which stores the real value of the symbol. The third is a word quantity used to store the checksum.

Location Storage

Locations (both precision, and cartesian types) are stored similarly to symbols in a location table, except that there are ten fields for each location. The first is the name string, as previously described. The next eight are four byte locations used to store the components of each location. Since the robot control can control up to 8 individual axes of motion, this limit was designed in. When a cartesian coordinate is stored, it uses the first six of these 8 fields to store a four byte real number representation of each world component. When a precision point is stored, each of the eight fields store a double size integer value of each axis motor coordinate (in encoder pulse units). The tenth field stores the checksum.

The configuration of the user memory can be described in terms of structures and their elements. The appropriate PL/m-86 definitions would appear as:

```
ProgramTable(ProgramTableSize) structure (
     Name(8)   byte,
     Index     word,
     Checksum  word) at (ProgramTablePointer)

VariableTable(SymbolTableSize) structure (
     Name(8)   byte,
     Value     real,
     Checksum  word) at (SymbolsPointer)

PrecisionPointTable(LocationTableSize) structure (
     Name(8)   byte,
     AxisVal(8) dword,
     Checksum  word) at (LocationsPointer)

CartesianLocation(LocationTableSize) structure (
     Name(8)   byte,
     Coord(8)  byte,
     Checksum  word) at (LocationsPointer)

ProgramBuffer(ProgramBufferSize) byte at (ProgramBufferPointer)
```

The checksums associated with each element of the structures consists of the arithmetic sum of each byte in the structure element. Only the low byte of the checksum is used. The checksum used in the program table is the checksum of the entire program that it references, from the first byte of the program, up to and including the '$' delimiter used to end a program.

Equivalent program elements can be designed for Pascal or C language based host computer interfaces.

Exact addresses for each of the system parameters mentioned above can be found in the memory map for your particular version of RAPL. This information can be found in Appendix F.

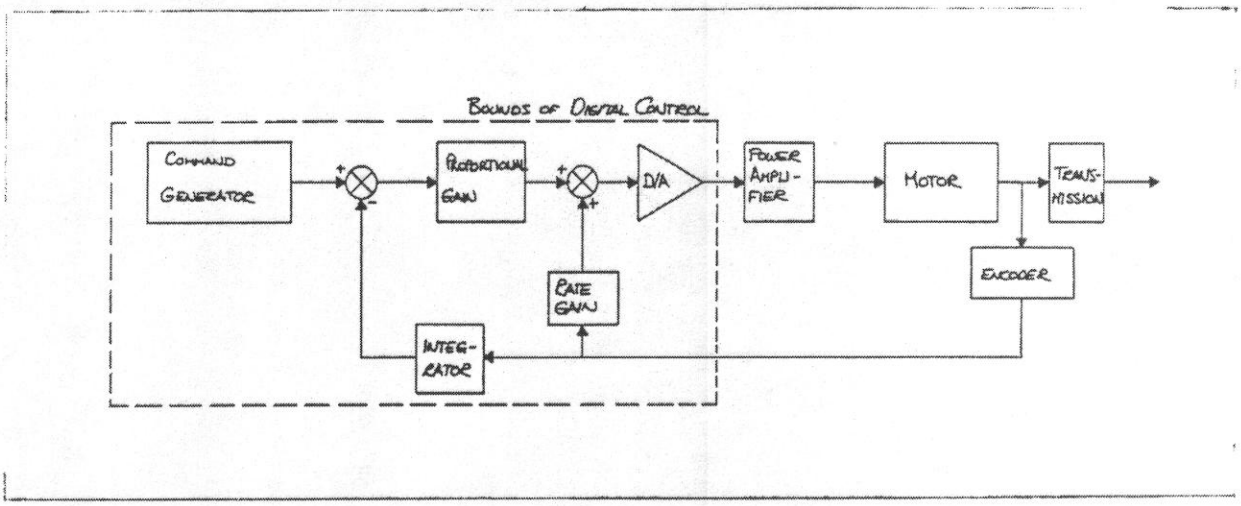


Figure 8-1 Closed Loop Functional Diagram

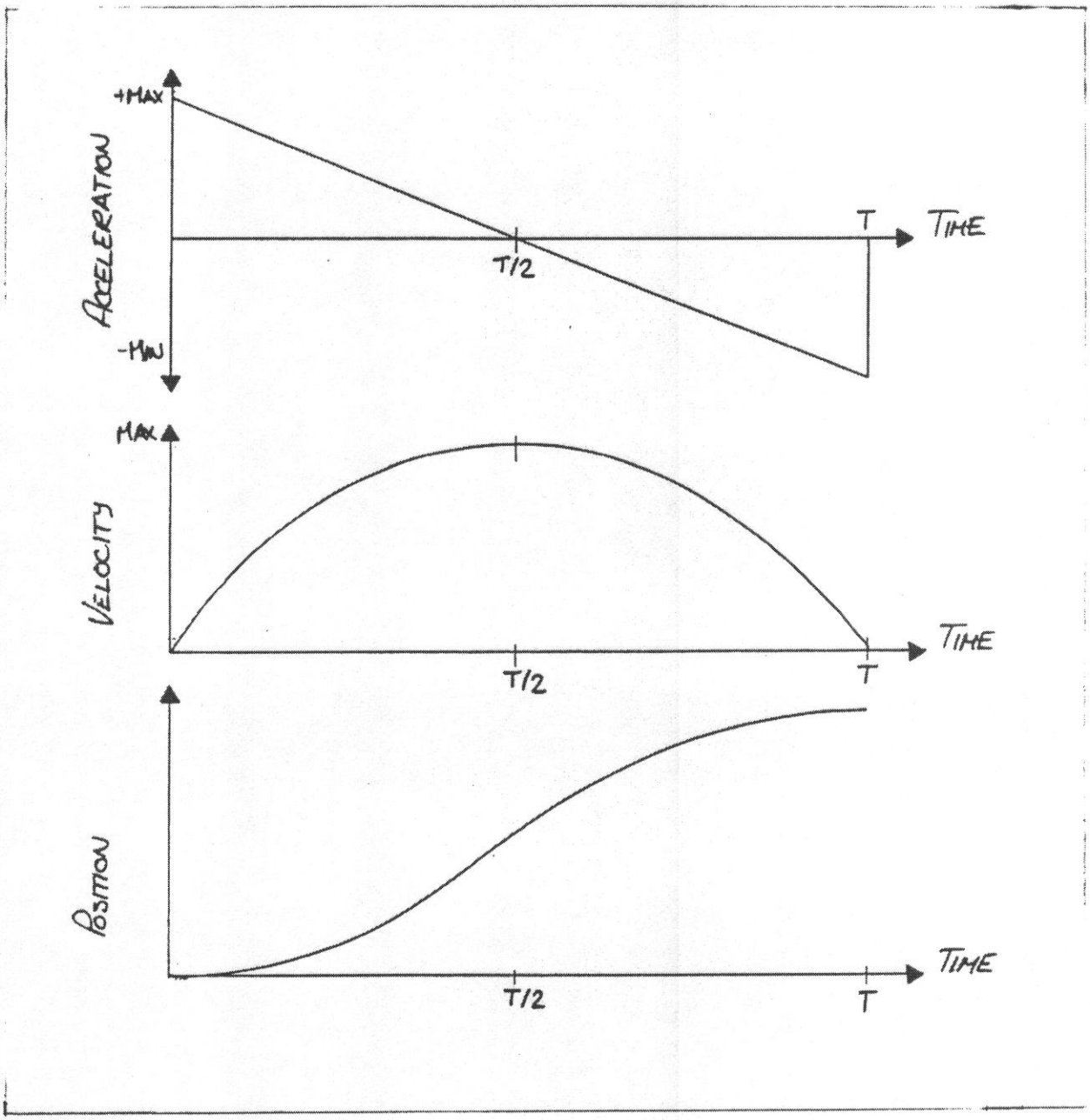


Figure 8-2 Standard acceleration/velocity profiles for joint interpolation

Address (hex)	Description
FFFF:0	Bootstrap address
F000:FFEF	Upper limit of firmware address space
E000:0000	Standard 128K firmware address boundary
C000:0000	Extended 256K firmware address boundary
0000:FFFF	Optional expansion memory upper limit
0000:3FFF	Standard 16K RAM memory upper limit
0000:2000	Approximate start of USER memory (may vary according to firmware release version, and other customer options) Refer to Appendix F.
0000:1FFF	Approximate top of CPU scratchpad space and system parameter space
0000:1000	Start of CPU scratchpad and system parameter space
0000:0FFF	Top of 8086 stack space
0000:0400	Start of 8086 stack space
0000:03FF	8086 vector interrupt space
0000:0000	Bottom of memory

Table 8-1 Memory Map

9-1 INTRODUCTION

The purpose of the Homing Bracket is to allow simple homing of the robot in a situation where the operator may not be trained for, or be able to home it accurately using the factory-supplied marks. Such applications are found in industrial systems and labs where the arm may not be easily accessible.

In order to use the homing bracket, the robot calibration must be changed. This is not difficult and will be described in detail in this manual.

9-2 ROBOT CALIBRATION

The calibration numbers in the robot are five numbers that describe the distance in motor pulses from the true "zero" position of the arm to the HOME position. The HOME position is a set location and is near the READY position as supplied from the factory. To set the calibration, move the robot to within one motor turn of the HOME position, and enter the @@CAL command. Each motor rotates in turn until its index mark is reached. At this mark, it stops. The distance from the zero to this index mark is stored as the calibration number for this axis. When the HOME command is issued, the user must position the arm within the same single turn of the motor of the original calibrated home location. Normally, the homing marks are used for this purpose, however, the homing bracket may also be used.

Home Bracket Installation

The position of the bracket in the robot workspace is important to the success of the procedure. The robot must exit from the bracket using joint commands. This is because until it is homed, the robot can only make relative moves. The most convenient joint to use for exiting is joint 3. The wrist flange must be perpendicular to the upper arm in order to slide out of the bracket. The angle of the support plate on the bracket is designed to position the bracket as close to the robot as possible, and still allow room for typical tooling. The optimum position can be seen in Figure 9-1.

In addition to the bracket itself, the small fixture plate must be installed on the robot wrist flange. Remove any existing tooling and install the plate between the gripper flange and the tooling. It should be installed so that the narrow end points away from the side of the flange nearest the factory homing mark.

Adding this plate between the robot and the gripper changes the position of the arm in accessing locations in the robot task. The distance can be compensated by re-teaching the locations or by adding a tool offset of 0.187 in the X direction. If no tool offset is used already, enter a location NEWTOOL with the value ".187,0,0,0,0,0". Then issue the command "TOOL NEWTOOL". Re-teaching the locations will be a better solution as any time the gripper comes off the robot, it may not be repositioned exactly where it was before.

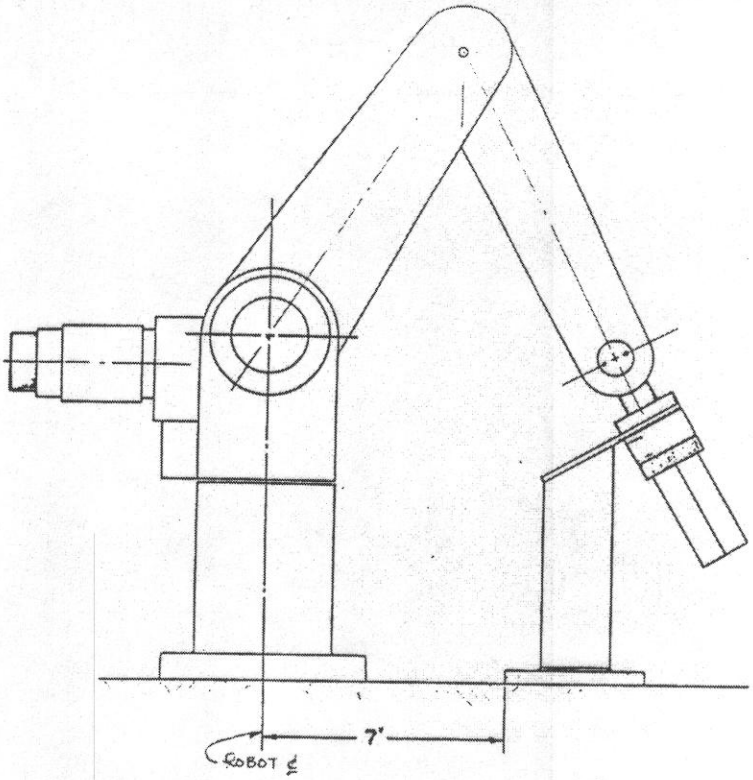


Figure 9-1 Robot Location relative to Homing Bracket

New Calibration

To change the location of the HOME position, start by issuing the HOME sequence using the home marks. This sets the zero location properly without having to measure the true zero location as described in the manual.

The robot must then be moved into the home bracket. This can be done easily with the ARM POWER off. When the arm is in position, turn arm power back on. set the speed to a slow value like 10%. Issue a joint command to move the robot out of the bracket such as "JOINT 3,20". Watch the arm carefully to make sure it leaves the bracket easily without binding. Application of a bit of white grease to the inner edge of the bracket can ease the robot's exit.

Due to interference with external equipment etc. the robot's current location may not be the best for homing (remember, the arm moves small distances during the homing sequence). In this case, use joint commands to move the robot to a clear location. Remember the sequence of moves taken to get here as the robot will have to repeat them each time it is homed. Once at a clear location, use the @@CAL command to recalibrate the robot at the new home location.

There is one note of caution to be added. If during the CAL sequence, an individual move of a joint to get to the encoder index mark takes less than 5 degrees (give or take) of motion, it would be a good idea to drive that joint back by about 1/2 a motor turn to ensure that it sees the correct index mark every time. Such a correction can be used in the homing program (see below).

9-3 AUTO START PROGRAM

The desired result of using a homing bracket is to be able to start the robot automatically. The use of the special program "AUTO_ST" will permit the robot to exit the bracket, move to the location from which it was calibrated, and execute a HOME command. It will then run the application software.

The following auto start program is an example of a typical one. In this case, the subroutine called EXIT will move the arm out of the bracket and home the arm. EXIT is listed below. The application program is called MAINLOOP and when it finishes, the E_STOP program will run.

```
PROGRAM AUTO_ST:
10 TYPE 'If arm in bracket then turn ARM POWER on.'/
30 ONPOWER
40 GOSUB EXIT
45 GOSUB INIT
50 GOSUB MAINLOOP
60 GOSUB E_STOP
70 STOP
$
```

The EXIT subroutine repeats the joint moves to take the arm out of the bracket. Lines 55 and 60 are there to adjust the wrist to make sure that the HOME sequence always finds the correct index mark. The values used in these two lines are emperically determined. Notice the use of low speed in the exit move, line 20.

```
PROGRAM EXIT:
10 SPEED 10
20 JOINT 3,15
40 SPEED 50
50 JOINT 1,-30
55 JOINT 4,-5
60 MOTOR 5,-250
70 HOME
80 RETURN
$
```

The E_STOP subroutine is called whenever a safe condition is breached in the workspace, or when the normal end of the shift occurs. This routine first moves the robot to a safe location (SAFE), The IGNORE command causes the robot to ignore the stop signal which caused it to jump to this routine. It then moves back into the homing bracket, and the ARM POWER is turned off.

```
PROGRAM E_STOP
10 MOVE SAFE
20 IGNORE
30 MOVE ATCAL
40 MOVE PREHOME
50 SPEED 10
60 MOVE FIXTURE
70 FINISH
75 DELAY 1
80 ARM OFF
90 STOP
$
```

CONTENTS

A-1 DESCRIPTION . . . . . A-

A-2 COMMAND DESCRIPTIONS . . . . . A-
@ACCEL . . . . . A-
@APC . . . . . A-
@@CAL . . . . . A-
@CALRDY . . . . . A-
@@DIAG . . . . . A-
@DUMP . . . . . A-
@ESPRIT A-1
@GO A-1
@@GTYPE A-1
@IN A-1
@LOFB A-1
@MAXVEL A-1
@NAPC A-1
@NOA A-1
@OUT A-1
@@RD A-1
@@RE A-2
@@RH A-2
@@RI A-2
@@RN A-2
@@RS A-2
@SUBST A-2
@XLIMITS A-2
@XPULSES A-2
@XRATIO A-2
@ZERO A-2

A-1 DESCRIPTION

All monitor commands require the execution of the **PASSWORD** command to allow access to this level of control. The password is included in the front of tis manual binder on a seperate sheet of paper.

Many of the following commands control software "toggle" switches; that is they turn on or off a software attribute. In this case the command description includes the default condition. This is the condition which the software would return to when a "TEACH" start or cold boot procedure is executed. In some cases this default has changed with different versions of RAPL. In such cases, the variations are listed.

Some of the monitor level functions provide the programmer with a means of controlling and observing the robot memory on a bit level. This enables the programmer to perform advanced diagnostic functions, and more comprehensive remote functions as well. Data stored in memory can be of differing sizes, or word types. Each has a different representation. A list of these data types is given:

Data descriptor	Data type	length in bytes	description
B	byte	1	unsigned quantity 0-255
W	word	2	unsigned quantity 0-65535
I	integer	2	signed, -32767 - 32766
P	pointer	4	hex notation, segment:offset, the segment resides in the upper word, and offset in the lower word.
D	double word	4	unsigned 0 - 4 billion
J	double integer	4	signed, +/- 2 billion
R	real	4	single precision

TABLE A-1 - Data type information as used in RAPL

If mention is made to boolean data, this implies a byte representation whose least significant bit is used to denote a true or false situation. A false situation exists if this bit is zero, and true if it is set to 1.

The monitor functions utilize the full 8086 addressing format. The address consists of a segment and an offset. All monitor functions must operate in the same segment space. Any data output are displayed in decimal format.

@ACCEL

FORMATS:

Source: **@ACCEL (axis#),<value>**
Compiled: /013 (axis#),<value>

DESCRIPTION:

Alter the maximum acceleration rate parameter for any axis. This value can be (0 < value <= 2) and is a real number.

The unit of <value> is in pulses per tick/tick, where a tick is the minimum unit of time used by the command generation software and a pulses is the basic length unit of the servo. For the M1A robot with standard axis cards, the default value is 1 ie. 1/1000 of a motor turn per 0.004^2 $seconds^2$. This gives a maximum acceleration of 62.5 revolutions/sec^2.

APPLICABLE MODES:

{I},{M}

@APC

FORMATS:

Source: **@APC**
Compiled: /120

DESCRIPTION:

Enables Arm Power check. For example, when the arm power is removed, the controller will signal a error message.

DEFAULT CONDITION

ARM Power check is on.

APPLICABLE MODES:

{I},{P},{M}

@@CAL

TOKEN: /019

FORMAT:
@@CAL

DESCRIPTION:

Factory calibrate command used at Factory. Do **NOT** use this command unless servicing the robot.

WARNING

DO NOT USE THIS COMMAND, UNLESS ON THE ADVICE OF CRS PLUS.

EXAMPLE

Step	Action
1	Do a "TEACH" start.
2	Arm power should be on.
3	Robot should be in "MANUAL" mode.
4	Move arm to zero position as shown in figure A-1.
5	Enter "NOMANUAL".
6	Enter "PASSWORD".
7	~~Support arm so it doesn't drop during zeroing.~~ SPEED 80
8	Enter "@ZERO".
9	Enter "JOINT 2,90"
~~10~~	~~Check each joint to ensure they are in "HOME" range. If not repeat.~~
11	Enter "@@CAL Y".

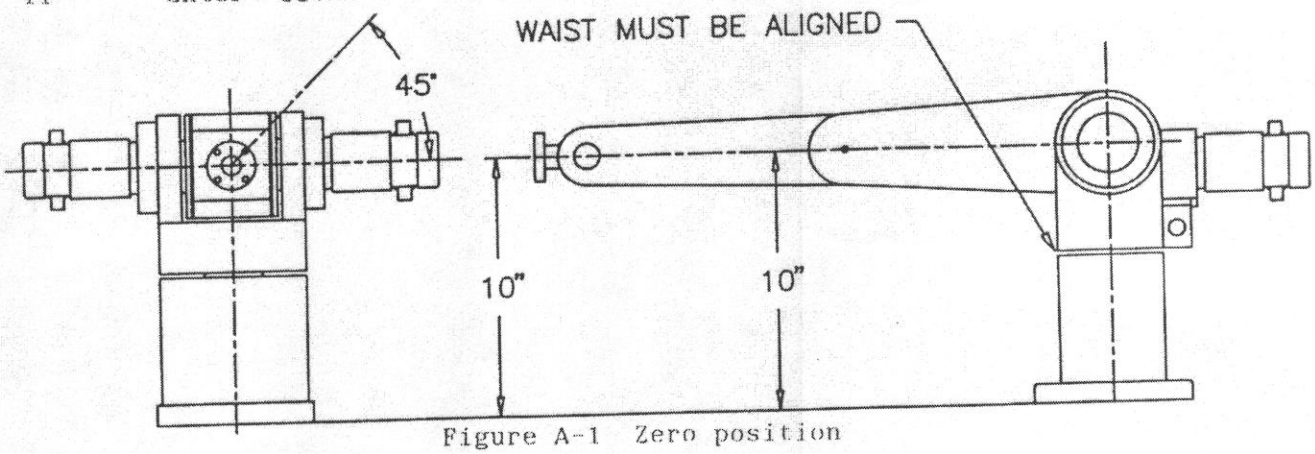


Figure A-1 Zero position

FORMATS:

Source: **@@CAL**
Compiled: /019

DESCRIPTION:

Factory calibrate command used at Factory. Do **NOT** use this command unless servicing the robot.

WARNING

DO NOT USE THIS COMMAND, UNLESS ON THE ADVICE OF CRS PLUS.

EXAMPLE

Step	Action
1	Do a "TEACH" start.
2	Arm power should be on.
3	Robot should be in "MANUAL" mode.
4	Move arm to zero position as shown in figure A-1.
5	Enter "NOMANUAL".
6	Enter "PASSWORD".
7	Support arm so it doesn´t drop during zeroing.
8	Enter "@ZERO".
9	Enter "JOINT 2,90"
10	Check each joint to ensure they are in "HOME" range. If not repeat.
11	Enter "@@CAL Y".

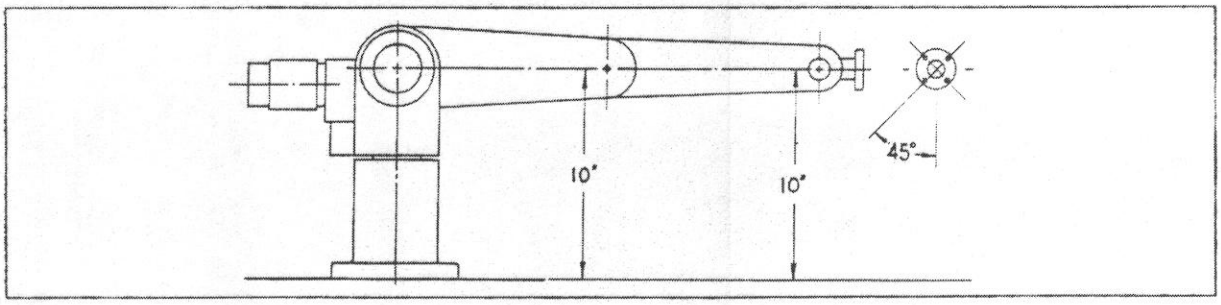


Figure A-1 Zero position

@CALRDY

FORMATS:

Source: **@CALRDY**
Compiled: /144

DESCRIPTION:

The @CALRDY statement will take the arm to the zero position in a joint coordinated fashion. The operator should be aware of the @CALRDY location, and realize that the robot will move to this location in a joint interpolated fashion from its current location when the command is issued. The operator must ensure that a collision will not occur during this motion. It is recommended that the speed be set low and the operator have access to the ABORT button during execution of this command.

APPLICABLE MODES:

{I},{M}

FORMATS:

Source: **@@DIAG**
Compiled: /113

DESCRIPTION:

This is a interactive command that will check the controller hardware an software.

The following is an example of the information displayed:

```
ARE YOU SURE THAT THE FOLLOWING IS TRUE?
ROBOT ARM POWER IS OFF,
YOU HAVE READ THE SERVICE MANUAL
                                                        Respond with "Y"

PIC R/W TEST
                    ** PASSED **
PIT COUNTING TEST
                    ** PASSED **000
                    ** PASSED **001
                    ** PASSED **002
PIT INTERRUPT TEST
                    ** PASSED **000
                    ** PASSED **001
                    ** PASSED **002
8087 TEST
                    ** PASSED **
TEACH PENDANT TEST
READY LAMP!! - "E" to end                               Check READY lamp
                    ** PASSED **
GRIPPER OUTPUT TEST!! - "E" to end                  Listen for valve action
                    ** PASSED **
Input states - "E" to end                       Check Teach Pendant switches
0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  0  1  1  1
Press AUTOSTART                                     Press AUTO-START switch
                    ** PASSED **
Press TEACH                                             Press TEACH button
                    ** PASSED **
Press ABORT                                             Press ABORT button
                    ** PASSED **
```

Analog input test - "E" to end	If analog option installed, check digital values
No Analog to Digital Converter	
Teach pendant analog input test - "E" to end 142	Rotate TP speed pot
** PASSED *	
Encoder Read Test - "E" to end	
# of Axis cards installed = 008 0000 0000 0000 0000 0000 0000 0000 0000	Rotate robot joints, check feedback (HEX #'s displayed)
** PASSED **	
Encoder Clear Test	
Velocity output test - "E" to end	Observe velocity output to amp with
RAM READ/WRITE TEST RAM SIZE :4000h	Check memory size

Table A-2 - output from the @@DIAG command.

APPLICABLE MODES:

{I}

FORMATS:

Source: **@DUMP <B¦W¦I¦P¦D¦J¦R>,<segment:offset>,<# of elements>**
Compiled: /114 <B¦W¦I¦P¦D¦J¦R>,<segment:offset>,<# of elements>

DESCRIPTION:

Contents of memory specified is displayed to the terminal. Depending on the data type, the DUMP command will index through memory with the correct byte count, so that consecutive words of that type are correctly addressed. The data types are as follows:

Data descriptor	Data type	length in bytes	description
B	byte	1	unsigned quantity 0-255
W	word	2	unsigned quantity 0-65535
I	integer	2	signed, -32767 - 32766
P	pointer	4	hex notation, segment:offset, the segment resides in the upper word, and offset in the lower word.
D	double word	4	unsigned 0 - 4 billion
J	double integer	4	signed, +/- 2 billion
R	real	4	single precision

TABLE A-3 - Data type information as used in the @DUMP command.

APPLICABLE MODES:

{I},{M}

@ESPRIT

FORMATS:

Source: @ESPRIT
Compiled: /123

DESCRIPTION:

Toggles editor function for Esprit 6110+ video terminal to allow editing commands to remain on same line. Useful for displaying program and editing while retaining display.

DEFAULT CONDITION

Off for all RAPL versions.

APPLICABLE MODES:

{I},{M}

FORMATS:

Source: **@GO <segment:offset>**
Compiled: /115 <segment:offset>

DESCRIPTION:

This command will execute an 8086 macro assembler program residing at the segment:offset address given in the command.

APPLICABLE MODES:

{I},{M}

@@GTYPE

FORMATS:

Source: **@@GTYPE <A¦S¦M>**
Compiled: /140 <A¦S¦M>

DESCRIPTION:

This command informs the robot controller of the gripper configuration. The argument <A> specifies the air gripper to use, while the argument <S> specifies the servo gripper. The <M> specifies the magnetic gripper for use.

This command should only be used when a new gripper is installed on the arm. When the servo gripper option is installed, the arm power must be turned on. If it is not, then a message will appear indicating to the user to turn it on.

DEFAULT CONDITION

The AIR gripper type is selected in the default condition.

APPLICABLE MODES:

{I},{M},{P}

FORMATS:

Source: **@IN <B¦W>,<port address>**
Compiled: /116 <B¦W>,<port address>

DESCRIPTION:

Contents of the specified input port specified is displayed.

APPLICABLE MODES:

{I},{M}

@LOFB

FORMATS:

Source: **@LOFB <S¦R>**
Compiled: /162 <S¦R>

DESCRIPTION:

Loss of feedback/collision detection check set or reset. When set, this function will check for any motor which is being commanded full speed, while no change in position is obtained over a 0.5 second interval. These symptoms are typical of the following situations:

1. Broken encoder, producing no feedback.
2. Blown fuse.
3. Robot has collided with an object and is binding.
4. Static overload.

DEFAULT CONDITION

For version 3.4.5 and 3.4.6, the default condition is off (Reset). For all versions from 3.5.0 and on, the default condition is on (Set).

APPLICABLE MODES:

{I},{M}

EXAMPLE:

1. To activate collision detection
   @LOFB S <cr>

2. To de-activate collision detection
   @LOFB R <cr>

FORMATS:

Source: **@MAXVEL [axis#],[VELOCITY]**
Compiled: /167 [axis#],[VELOCITY]

DESCRIPTION:

This command is used to tune the maximum value of command velocity to the maximum pulse output rate of the encoder for an extra axis.

If this value is set improperly, the encoder of the extra axis may not be anble to send out pulses fast enough to keep up with the command. The axis may lag behind so badly that the command error may overflow its buffer and the motor will reverse. In less drastic cases, the motor will be poorly synchronized with the robot, particularly at higher command speeds, and it will be difficult to control the axis under manual control.

Determine the value of [VELOCITY] by running the motor at full speed under the given load and amplifier conditions, and measure the encoder speed in RPM. The encoder output can be used as a tach if a scope can be attached to its output. The frequeny along with the opulse count per revolution will give the speed in RPM using the following formula:

$$\text{ENCODER RPM} = \frac{60}{(\text{Pulses/rev})(\text{Pulse-Period}_{\text{sec}})}$$

NOTE

The @XMAXVEL command should be executed after any @XPULSES or @XRATIO command, since the @XMAXVEL command generates internal parameters that are dependant upon all extra axis parameters.

APPLICABLE MODES:

{I},{M}

@NAPC

FORMATS:

Source:	**@NAPC**
Compiled:	/121

DESCRIPTION:

This command will disable the arm power check. For example, if the arm power is removed, the controller will not give a error message.

WARNING:

Without the arm power check flag set, the robot will not go into the limp condition when the arm is turned off. This means that the arm will jump to the location where it was when the arm power was disabled, when the power is restored. To avoid this, use the limp command as soon as the arm power is turned off when the @NAPC command has been issued.

APPLICABLE MODES:

{I},{P},{M}

FORMATS:

Source: @NOA
Compiled: /122

DESCRIPTION:

This command will display the current number of axis, and permit the user to change the number.

DEFAULT CONDITION

For versions 1.0 to 3.50, the default condition is 5 axes. For version 4.00 the controller reads the number of axis cards installed if more than 5, and prompts the user to enter the correct number. The system will not permit further operation until a response has been entered.

APPLICABLE MODES:

{I},{M}

@OUT

FORMATS:

Source:	**@OUT <B¦W>,<port address>,<value>**
Compiled:	/119 <B¦W>,<port address>,<value>

DESCRIPTION:

This command will permit the user to force an output port to a given digital value (byte or word value).

APPLICABLE MODES:

{I},{M}

FORMATS:

Source: @@RD
Compiled: /133

DESCRIPTION:

The ACI interface will be disabled. The serial port #1 will now be considered as any other programmable serial channel.

DEFAULT CONDITION

For versions prior to 3.50, the default condition is ACI disabled. For versions 3.50 and on, the default is ACI enabled.

APPLICABLE MODES:

@@RE

FORMATS:

Source: @@RE
Compiled: /134

DESCRIPTION:

Enable the ACI software. This will dedicate the serial port #1 to the ACI software. No other normal serial input will be allowed on this channel. Output is still allowed, but it will cause problems if an ACI cycle is concurrently running.

DEFAULT CONDITION

For versions prior to 3.5.0, the default condition is ACI disabled. For versions 3.5.0 and on, the default is ACI enabled.

APPLICABLE MODES:

FORMATS:

Source: CCRH
Compiled: /135

DESCRIPTION:

The last header which was read in will be displayed. This will indicate the type of the last (or current) conversation.

APPLICABLE MODES:

{I},{M}

@@RI

FORMATS:

Source: @@RI
Compiled: /136

DESCRIPTION:

To force an initialization of the ACI interface. Any communication in sequence will be aborted. This may cause a loss of synchronization between the slave unit and the master control.

APPLICABLE MODES:

{I},{P},{M}

FORMATS:

Source: ccRN
Compiled: /137

DESCRIPTION:

To select a slave device number for the robot controller. This value can be any value between 1 and 127. The value of 0 is not permitted.

DEFAULT CONDITION

For versions prior to 3.50, there is not asingle value default condition. After 3.50, the default value is 1.

APPLICABLE MODES:

{I},{P},{M}

@@RS

FORMATS:

Source: @@RS
Compiled: /138

DESCRIPTION:

Display the current status of the ACI software. The number of retries, the number of communication failures and the number of successful cycles will be displayed in a continuous fashion on the terminal device. This is a useful debugging tool.

APPLICABLE MODES:

{I},{P},{M}

FORMATS:

Source: **@SUBST <B¦W¦I¦P¦D¦J¦R>,<segment:offset>,<data>**
Compiled: /117 <B¦W¦I¦P¦D¦J¦R>,<segment:offset>,<data>

DESCRIPTION:

Load the new value into the specified memory location. By terminating the new data entry with a space instead of a <CR>, the function will display the next data value in memory, along with its address. The process of entering data can be repeated until a <CR> is entered. The data types are as follows:

Data descriptor	Data type	length in bytes	description
B	byte	1	unsigned quantity 0-255
W	word	2	unsigned quantity 0-65535
I	integer	2	signed, -32767 - 32766
P	pointer	4	hex notation, segment:offset, the segment resides in the upper word, and offset in the lower word.
D	double word	4	unsigned 0 - 4 billion
J	double integer	4	signed, +/- 2 billion
R	real	4	single precision

TABLE A-4 - Data type information as used in the @SUBST command

APPLICABLE MODES:

{I},{M}

@XLIMITS

FORMATS:

Source: **@XLIMITS <Axis #>,<Positive Limit>,<Negative Limit>**
Compiled: /166 <Axis #>,<Positive Limit>,<Negative Limit>

DESCRIPTION:

This command allows the system programmer to enter a joint limit travel for th extra axes. The units of the limits are entered in the same units as are use for the JOINT command. That is, the limits are based on the values that XRATI and XPULSES have at the time the XLIMIT command is used. Whenever the XRATIO c the XPULSES command is re-used, then the XLIMIT command should also be re entered in order to obtain the correct physical joint limits.

The command only works for the extra axes.

FORMATS:

Source: **@XPULSES <Axis #>,<Pulses>**
Compiled: /160 <Axis #>,<Pulses>

DESCRIPTION:

This command permits the programmer to adapt a suitable optical encoder to the extra Axis. See Appendix E of the Technical Manual for a description of the electrical specifications for an encoder.

The arguments provided by the @XPULSE command specify the axis number and the number of encoder pulses per motor revolution. Valid pulse values are between 1 and 32767.

DEFAULT CONDITION

The default value for axes 6 through 8 is 1000 pulses per revolution, the value used for the SRS-MOTOR option.

APPLICABLE MODES:

{I},{M}

EXAMPLE:

1. @XPULSES 6,100

@XRATIO

FORMATS:

Source: **@XRATIO <Axis#>, <Factor>**
Compiled: /161 <Axis#>, <Factor>

DESCRIPTION:

This function provides a scaling factor for any of the extra axes (6, 7 or 8 for use in a JOINT command. It is a Transmission Ratio that relates moto turns to measurement units. The formula to calculate the value for any extr axis would be:

TR = Measurement Units/encoder revolution

The TR can be any value that produces a useable unit of command for th programmer.

DEFAULT CONDITION

The default value of TR for axes 6 through 8 is 1.

APPLICABLE MODES:

{I},{P},{M}

EXAMPLE:

1. A motor uses a 100:1 gear reduction and the programmer wishes to command th motor in degrees of output shaft rotation. The factor would be:

$$TR = \frac{360 \text{ degrees/turn}_{output}}{100 \text{ turns}_{input}/\text{turns}_{output}}$$

$$= 360 / 100$$
$$= 3.60 \text{ output degrees/encoder turn}$$

2. The command would be:

@XRATIO 6,3.6

FORMATS:

Source: @ZERO
Compiled: /022

DESCRIPTION:

The @ZERO command will set the position registers to zero. This command is use when calibrating the robot. Use this command with extreme care. Once the @ZER command is issued, then the real travel limits of the arm may not be protecte by the software limits.

WARNING

DO NOT USE THIS COMMAND, UNLESS ON THE ADVICE OF CRS PLUS.

B-1 INTRODUCTION

The ACI permits external computer systems to communicate with one or more robot controllers on a single RS232 link.

In its basic configuration, this protocol is used to transfer raw data either to or from the master device. The protocol allows any chunk of 8086 memory to be the object of the communication in a segmented addressing technique.

The protocol permits error checking and automatic transmission retries in order to establish a communication link.

B-2 FUNCTIONAL DESCRIPTION

The ACI is a master/slave protocol. All robot controllers are configured as slaves in the network. Any external compter would then have to be a master. Only a master can establish a communication. All the communication initiative must be taken by the master unit.

B-3 MASTER PROTOCOL

The master device must establish a communication link by specifying a slave device number as a target. It must then confirm to the target that it is indeed requesting a communication. After the link has been established, the master must then provide the slave with the information describing the data transfer that is about to take place. After all of the particulars concerning the communication has been transferred, the actual data is then sent. The data is transferred in packages of 128 bytes. Each 'block' of data contains its own start/stop characters and a checksum test byte to validate the data after transfer. Finally, it is up to the master to close the communication link with an EOT character.

A communication consists of four separate blocks. They are described in detail in the following sections.

B-4 ENQUIRY SEQUENCE

The master issues a simple three byte code to the serial line. This code is;

'R' , slave ID + 20h , ENQ

The master must then follow this sequence with two character times of no transmission. This will ensure that the slave has indeed read a valid enquiry sequence, and not some random chunk of another communication. When the slave has interpreted a correct enquiry, it will issue an appropriate response;

'R' , slave ID + 20h , ACK

When the master reads this response it has established the communication link to the target device.

The slave id number is any number between 0 and 7F hexadecimal. Using the ACI monitor command **@@RN**, the programmer can configure each robot controller with an appropriate slave id number.

Adding a value of 20 hex to the slave number provides security that the second byte of this string does not resemble a control code in any way. This way, the slave device will be able to distinguish an enquiry sequence and can establish communications quickly.

The master should attempt to contact the slave only a limited number of times before abandonning the communication. A failure to establish the communications with the slave after three attempts normally indicates a failure in the link. Either the baud rates are not set correctly at either end, or a physical problem exists with the interface wiring.

A delay of 2 character times should be included in the transmission of the enquiry sequence. This delay is placed between the second and third characters of the enquiry sequence. This time delay ensures that a three byte string from a data block cannot be misconstrued as an enquiry sequence. The master control must ensure that this time delay exists, or proper ACI operation cannot be guaranteed.

B-5 HEADER SEQUENCE

The header block consists of a description of the data transfer that will take place. The header block is broken down into the following byte description;

1	SOH	Start of header character
2	SLAVE ID + 20h	Slave identification number
3	MASTER ID + 20h	master identification number
4	READ/WRITE	data read or write selection
5	MEMORY TYPE	memory access type (see special codes)
6	NUMBER OF FULL BLOCKS	number of full blocks transferred
7	NUMBER OF BYTES IN LAST BLOCK	number of byte in last data block
8	MEMORY OFFSET LOW	target memory starting address
9	MEMORY OFFSET HI	
10	MEMORY SEGMENT LO	
11	MEMORY SEGMENT HI	
12	ETX	end of text character
13	LRC	longitudinal redundancy check

Table B-1

All data values are expressed in hexadecimal notation.

The slave ID number is the same that appears in the enquiry sequence.

The Master ID number must be 01 hexadecimal.

The Read/Write byte identifies what type of operation is to be executed. A write operation will transfer data from the slave device to the master. The read operation is the opposite. It transfers data from the master to the slave. The following codes are used:

READ	**01h**
WRITE	**00h**

The memory access type specifier identifies specific areas of memory that the data transfer can take place in. This eases the burden of programming on the programmer in many cases.

The elementary code '00h' is the general purpose memory access code. This code eneabled the programmer to read or write any byte in the 8086 memory space. This makes the command a very powerful and also a potentially hazardous tool to work with. CRS technical staff should be consulted when this command is used.

A list of the special codes available appears in a later section.

The next two byte in the header identify the amount of data to be transferred. This information is considered as the number of full data blocks to be transferred, and the number of bytes which remain in the last data block. The last data block cannot have 0 bytes in it. A full block contains 128 bytes of data. If the number of bytes to be transferred is an exact multiple of 128, then the last block will contain 128 bytes (80 hex). The logic to determine the number of blocks and the number of bytes in the last block is as follows;

```
let N be the total number of bytes to be transferred,
let B be the number of full blocks to be transferred,
let n be the number of bytes in the last block;

B = N / 128
n = N mod 128        ; (ie. the remainder of the division N/128)
if n = 0 then do
     n = 128
     B = B - 1
endif
```

The memory address identified the starting address of the data transfer operation. It defines (in Intel segment/offset format) where in the robot memory the data will be transferred to or from during this operation.

Header Transfer Error Detection

The LRC byte is the longitudinal redundancy check character . It is the sum of bytes 2 through 11 inclusive of the header block. The slave device sums the values of all received characters and will accept the header block only if it reads an LRC that is the same as its own computed value. If the LRC's do not match, then the header block is not accepted, and the slave will issue a **NAK** character in response so that the master control will know to re-try the header block transfer. The master can only have three re-try attempts before the slave device issues an error, and aborts the communication cycle. If the header block is accpted, the slave device will issue an **ACK** character as a response. The slave also expects the correct control codes **SOH** and **ETX** during the transmission, and any deviation from this pattern will cause an error, and the communication will be aborted. Timeout checks are made between the end of the enquiry sequence, and the accpetance of the SOH character, and also the length of time reuqired between sending an SOH and ETX character.

B-6 DATA BLOCK TRANSMISSION SEQUENCE

The data block transfer sequence is determined by the format specified in the header block.

In a data write command, the slave will begin to write the specified data blocks quickly after it issues an ACK in response to receiving the header block. It expects ACK characters from the master after each block is transmitted. Receiving a NAK will provoke a retry of the previous block. After the last block is sent and acknowledged, the slave will issue an EOT. It expects to receive a final EOT from the master, and then will close down the communication. The data blocks are configured in the following manner;

STX + [DATA BYTE #0] + ,.....,+ [DATA BYTE 127] + ETB + LRC

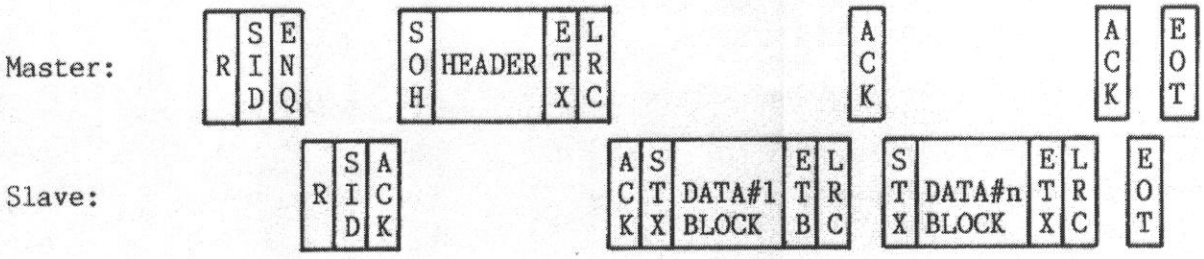
for the full data blocks, and

STX + [DATA BYTE #0] + ,.....,+ [DATA BYTE #n-1] + ETX + LRC

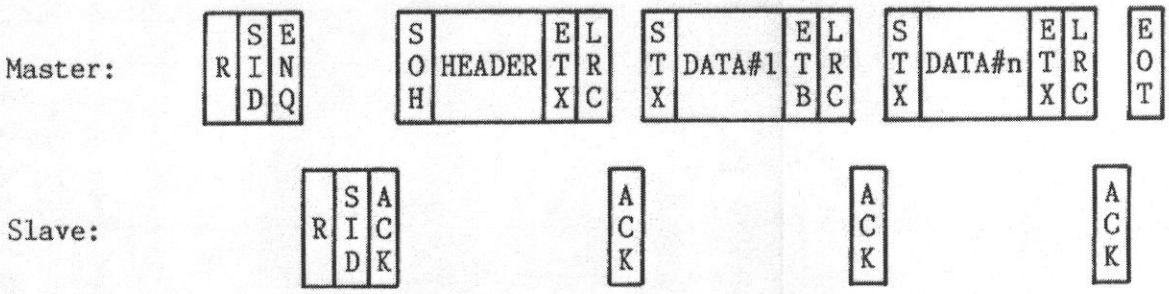
for the last data block.

The last data block uses an ETX character instead of an ETB character in order to signal the very end of the transferred data.

The following diagram illustrates the flow of data between the master and slave devices during a separate read and write cycle.



Similarly, in a read command, the slave will expect blocks of data after the header is aknowledged. It will issue a NAK if it detects an error in transmission. The master will issue an EOT after the last data block is read and acknowledged.



Data Transfer Error Detection

As in the header block transfer, the LRC byte is the sum of all data bytes in the block. Control codes STX, ETX or ETB are not included in the summation. An erroneous LRC check will cause the receiving end to issue a **NAK** character, which will force a re-try of the entire block. Only three re-tries are permitted, whereupon the recieving end should issue an **EOT** and abort the sequence. Timeout checks are made for reception of an STX after a complete header transfer, or a complete previous data block transfer, and also btween accpeting the opening STX of a data block transfer, and reading the ETX or ETB character. Once a data block has been accepted by the ACI in a read operation, the data block is transferred to the target memory address.

B-7 EOT SEQUENCE

The EOT is the final sequence, and is shown in the timing charts above. In the communication, the master ALWAYS has the last word, and the master ALWAYS has the ability to abort the communication. The slave will never knowingly abort the communication once it is started.

B-8 ERROR CHECKING AND RECOVERY

The ACI can handle a range of different communication errors. The two broad classifications of errors are transmission inaccuracies, and character time outs. Character time outs are used when a pending communication cycle has been interrupted by an unforgivable time delay. After a time out condition, the slave ACI issues an EOT then resets for another cycle. When a character is transmitted incorrectly, two things may happen. First, if it results in a checksum error, the data block is retried. This process continues for a programmed number of retries. If still no luck is encountered, then the slave executes a complete cycle reset, ready for the next communication attempt. If a control character is mis-read because of a transmission error, then the cycle is aborted immediately, and the ACI is reset. A timeout error causes an immediate reset of the ACI.

Error Number	Error Description, Cause, and recovery
1	Not used
2	No SOH character found to lead off the header sequence. The SOH character must be the first character of the header. Any other character will cause the ACI to terminate, and the communication cycle will be ended. The ACI will release an EOT character.
3	The ACI encountered a software error when reading in the header block. This is an internal software failure. Contact your CRS Plus representative. The ACI will be reset after this error.
4	A header retry failure was encountered. This means that the ACI tried four times to obtain the header, and was unable to do so due to repeated LRC failures. Usually, this is a sign of a bad interface wiring, or incorrect master control software. If you are using CRS Plus interface software for your host computer, contact your CRS Plus representative.
5	No ETX, or ETB was read at the end of a data block transfer. Remeber that the last data block must have ended with an ETX character. All other data blocks end with an ETB character. The ACI will immediately reset after detecting this error.
6	Not used
7	Not used
8	Data block read retry failure. The ACI terminated due to repeated LRC check failures when reading a data block from the Master.
9	Character timeout. The ACI expected a character, and did not receive one in the allotted time. The ACI will be reset. Check that the Master control did not hang up part way through a transmission, or that the correct number of characters were sent.
10	The ACI read a character other than an STX at the start of a data block. The ACI will be reset.
11	Data block write retry failure. The ACI was unable to deliver a data block to the Master without getting a NAK responce four consecutive times

Table B - 2 ACI Error Codes

Error Number	Error Description, Cause, and recovery
12	A character other than an ACK or NAK was read in response to a data block write. This error can occur on a spurious noise injection into the line, but is a rare error type. If this error persists, then the Master could be responding with an incorrect code. Contact your CRS Plus representative. The ACI will be reset.
13	Timeout error on transmission of a character. The ACI was not able to transmit a character in the allotted time. This timeout would have been caused by a failure in the DART chip (see hardware description) to present a transmit ready signal. The ACI will be reset after this error. Contact CRS Plus if the problem persists.
14	Bad target Id match in header block. The header block must contain the same slave device number that the enquiry sequence had, or this error will be generated. The ACI will be reset.
15	Not used
16	Bad character received when EOT expected. The ACI expects the Master to close all communication with an EOT character. This error is generated if this does not happen, and another character is received instead. Since the communication is already finished by this point anyway, the ACI will be reset.
17	A character other than an ETX was received at the end of a header block. This character must be delivered to the ACI immediately after the LRC character is sent.
18	Timeout on receiving the header SOH character. The Master control, after establishing communications with the slave during the enquiry sequence, has a limited time to send the following header block. See the table of timeout values. A timeout error always causes the ACI to reset.
19	Timeout of recieving the header ETX character. This timeout measures the time it takes to complete the transmission of the header block. Once the SOH has been received, the Master has only a limited time to transmit the whole header block. The ACI will be reset after this error.
20	Timeout on receiving the data block STX character. When the header block has been completed, and the Master has programmed a read cycle for the slave, the Master must deliver the first data block within a time period. Failing to do so will reset the ACI.

Table B - 2 ACI Error Codes

Error Number	Error Description, Cause, and recovery
21	Timeout on data block ETB/ETX character. Similar to error #19, the master has only a limited time to send the entire data block before this error occurs. The ACI will be reset.
22	Timeout on receiving a data block ACK/NAK character. After the slave has transmitted the data block to the Master, the Master must respond with the ACK or NAK character within the preset time limit, or the ACI will reset automatically.
23	Timeout on receiving the final EOT. The slave expects the Master to close thecommunication within a specified time limit. Failure to do so will set this error, and the ACI will be reset.
24	Unexpected EOT character. This error will be set if any expected control character turns out to be an EOT. The Master can close off the communication prematurely by sending an EOT instead of the accepted control code at any time in the sequence.
25	Not used
26	Bad special read code. The read code provided in the header is not supported.
27	Bad special write code. The write code provided in the header is not supported.
28	Bad special memory access code. The memory access code provided in the header is not supported.

Table B - 2 ACI Error Codes (cont'd)

B-9 ABORTING A COMMUNICATION CYCLE

Sending an EOT at any time that a normal control character is to be issued will be interpreted by the slave as a signal to abort the communication. This function may be required at times were an abrupt termination of the link is essential, for instance when attempting to quickly service another slave.

B-10 INTERFACE REQUIREMENTS

The ACI has been tested at baud rates up to and inculding 2400 baud. Faster baud rates may be possible, but are not recommended. Handshaking should be disabled, and parity should be turned off. A data byte size of 8 bits is required, with two stop bits. See the RAPL CONFIG command for setting up serial channel 1 to service the ACI link.

B-11 PREPARING THE ROBOT CONTROLLER FOR A COMMUNICATION CYCLE

The robot controller must be set up to accept the communication. This communication is executed through serial port 1. When this is active, then any robot references to device #1 will not respond, since this protocol takes priority when it is activated.

Error Number	Error Description, Cause, and recovery
21	Timeout on data block ETB/ETX character. Similar to error #19, the master has only a limited time to send the entire data block before this error occurs. The ACI will be reset.
22	Timeout on receiving a data block ACK/NAK character. After the slave has transmitted the data block to the Master, the Master must respond with the ACK or NAK character within the preset time limit, or the ACI will reset automatically.
23	Timeout on receiving the final EOT. The slave expects the Master to close thecommunication within a specified time limit. Failure to do so will set this error, and the ACI will be reset.
24	Unexpected EOT character. This error will be set if any expected control character turns out to be an EOT. The Master can close off the communication prematurely by sending an EOT instead of the accepted control code at any time in the sequence.
25	Not used
26	Bad special read code. The read code provided in the header is not supported.
27	Bad special write code. The write code provided in the header is not supported.
28	Bad special memory access code. The memory access code provided in the header is not supported.

Table B - 2 ACI Error Codes (cont'd)

B-9 ABORTING A COMMUNICATION CYCLE

Sending an EOT at any time that a normal control character is to be issued will be interpreted by the slave as a signal to abort the communication. This function may be required at times were an abrupt termination of the link is essential, for instance when attempting to quickly service another slave.

B-10 INTERFACE REQUIREMENTS

The ACI has been tested at baud rates up to and inculding 2400 baud. Faster baud rates may be possible, but are not recommended. Handshaking should be disabled, and parity should be turned off. A data byte size of 8 bits is required, with two stop bits. See the RAPL CONFIG command for setting up serial channel 1 to service the ACI link.

B-11 PREPARING THE ROBOT CONTROLLER FOR A COMMUNICATION CYCLE

The robot controller must be set up to accept the communication. This communication is executed through serial port 1. When this is active, then any robot references to device #1 will not respond, since this protocol takes priority when it is activated.

B-12 ACI MONITOR COMMANDS

The programmer can access the ACI software through a set of commands available at the terminal. These commands will allow the programmer to monitor any communications to and from the controller. The programmer can also enable or disable the ACI interface.

The commands are only available through the monitor level.

@@RI:
To force an initialization of the ACI interface. Any communication in sequence will be aborted. This may cause a loss of synchronization between the slave unit and the master control.

@@RN:
To select a slave device number for the robot controller. This value can be any value between 1 and 127. The value of 0 is not permitted.

@@RS:
Display the current status of the ACI software. The number of retries, the number of communication failures and the number of successful cycles will be displayed in a continuous fashion on the terminal device. This is a useful debugging tool.

@@RE:
Enable the ACI software. This will dedicate the serial port #1 to the ACI software. No other normal serial input will be allowed on this channel. Output is still allowed, but it will cause problems if an ACI cycle is concurrently running.

@@RD:
The ACI interface will be disabled. The serial port #1 will now be considered as any other programmable serial channel.

@@RH:
The last header which was read in will be displayed. This will indicate the type of the last (or current) conversation.

Character	Hex Code	Decimal Code
STX	02	02
ETX	03	03
ETB	17	23
EOT	04	04
SOH	01	01
ENQ	05	05
ACK	06	06
NAK	15	21

Table B-3 ASCII control codes for the ACI

Event	Timeout (in character times ++)
Enquiry timeout Timeout used to test for characters received during the enquiry sequence.	300
SOH timeout Time permitted between the recognition of a successful enquiry sequence, and the acceptance of the header SOH character	300
Header ETX timeout Time permitted between acceptance of the header SOH character, and the detection of the header ETX character.	300
ACK for data block Time permitted for waiting for an ACK or NAK from the master after the transmission of a data block from the slave.	300
STX data block timeout Time permitted for waiting for the data block STX from the master after a successful acknowledgment of either the header block, or the the previous data block.	300
Data block ETX/ETB timeout Timeout permitted for the complete data block, from acceptance of the STX to the recpetion of either the ETB or ETX character.	300
Final EOT timeout Time permitted between the acknowledgement of the last data block read and the final EOT response, or the last data block write acknowledge from the master, and its response to the slaves final EOT.	300

Table B-4 ACI timeout values

++ Timeouts are given in character times so that the baud rate does not affect the timeout strategy.

B-13 SPECIAL ACI ACCESS CODES

This chapter deals with the special functions available with the ACI version 2 software from CRS Plus. The extra functions located here explain how the system porgrammer can interface to the robot control to obtain information that was previously very difficult to obtain.

The functions explained here will permit the programmer to access robot I/O ports, as well as position and command registers.

The special codes are separated into special write codes (codes 40h to 4Fh) and special read codes (20h to 2Fh). These codes are placed in the access type specifier in the header block.

Special Write Codes

With the exception for code 47h, it is not necessary to specify an address with special write commands. It is important, however to specify the amount of data to be transferred. This is important in all cases.

40h

This special write command will return the most relevant addresses within the robot software. The addresses which are returned will enable the master control to access user memory directly, thereby permitting program upload and downloading. The use of this function will provide master control software which can operate independently from the robot software version. The buffer of information which is returned is in the form of memory pointers (4 byte, segment, offset format) . The lowest byte contains the least significant byte of the pointer. The addresses included in the buffer of information point to the items in the robot memory as described in table F-8, in Appendix F of the technical manual. The buffer of pointer information is the same as that used by the RAPL-BIOS facility, and is not reproduced here. With this special code, no address field need be specified, but since the pointer information consists of more than one full block, any portion of the data can be returned for use, starting with the first element.

41h

Send robot error codes. This special write code will send four bytes to the master indicating the health of the robot control. The buffer of four bytes includes, in this order:

1	AlarmStatus	00	indicates no alarmcondition
		01	indicates that an alarm exists
2	RAPL error code	NN	See the RAPL error manual
3	ACI error	00	No ACI error in effect
		01	ACI error is in effect
4	ACI error code	NN	See the ACI error list. This is the last ACI error detected.

42h

This command will force a write of the user input port images as captured by the RAPL operating system. The memory address in the header block need not be initialized, but the byte count must be specified. That is, it should contain a value of 8. See the technical description of the robot I/O space for a description of the inputs that can be read.

43h

This command will force a write of the user output port images as created by the RAPL operating system. The memory address in the header block need not be initialized, but the byte count must be specified. That is, it should contain a value of 8. See the technical description of the robot I/O space for a description of the inputs that can be read.

44h

This command will cause a write to the host computer of the position command data in the robot controller. The position command data exists as an array of 8 double integer values in memory. No address need be specified in the header block, but a data length of 32 bytes or less should be specified.

45h

This command will cause a write to the host computer of the actual position data in the robot controller. The actual position data exists as an array of 8 double integer values in memory. No address need be specified in the header block, but a data length of 32 bytes or less should be specified.

46h

This command will cause a write to the host computer of the end point of path data in the robot controller. The end point position data exists as an array of 8 double integer values in memory. No address need be specified in the header block, but a data length of 32 bytes or less should be specified.

47h

This command will transfer 8086 I/O input port values to the host computer. The I/O address is specified by the offset portion of the data address field in the header block. The segment field is not used. The number of bytes specified will result in a read of all ports from the specified address. This is a special command in that only one partial block of data can be transferred. The number of bytes in the last block will then correspond to the number of I/O ports to be scanned.

Special Read Codes

Special read codes allow the programmer to load specific areas of the robot memory with data.

20h

This special read command will load up to 128 bytes into the active command input buffer. In this way, and external computer can simulate a data terminal entry. Only text characters will provide a valid response. Any attempt to enter control codes as you would at the interactive level with the terminal will produce a RAPL error. No echoing will be noticed, since all normal echoes will go to the terminal.

C-1 MECHANICAL MAINTENANCE

SCHEDULE

Routine maintenance will help the SRS-M1 to continue operating at peak performance for many years.

This section sets out a maintenance schedule for the mechanical system and details some of the procedures to be followed during maintenance periods.

The recommended maintenance is found in table C.1.

TABLE C.1 - MAINTENANCE SCHEDULE

Maintenance item	Description	Hourly interval
Service Transmission Chains	page C-2	1000
Check Inner Wiring Harness	page C-3	1000
Check arm covers for wear/cracking	page C-4	2000
Re-Lubricate Harmonic Drives	page C-5	2000
Check Motor brush wear	page C-6	5000
Inspect clean and refit Harmonic Drives	page C-7	8000

The first four service procedures above should be done after approximately 500 hours of use, as the chains have a tendency to wear in or stretch fairly rapidly at first. The inner wiring harness should be checked for damage that could have been caused by poor fitting of the harness.

All service items listed below must be done under the following conditions:

1) System power removed.

2) The service is to be done by qualified service personnel only.

SERVICE ROUTINES

ITEM: Service Transmission Chains

INTERVAL: 1000 hours

TOOLS REQUIRED:
- Long allen key driver - 3/32" AF.
- #0 Robertson screwdriver
- Loctite #601 retaining compound
- Heavy molybdenum-based oil

ACTION:

1) With robot arm power on, check chain tension on the slack side of the loop when the robot is carrying its expected load. Play in the chain is obvious when there is a "dead" feel in the slack (unloaded) side of the chain when pressed from side to side. If a dead feel can be detected, procede to adjust the tension as described below.

2) Remove rear lower arm cover, take out all screws on top section of upper arm cover and bend it back to expose the insides of the upper arm. This will give access to all five of the chain tensioners.

3) Using a long handled allen key tighten the screws in the tensioner until moderate finger pressure fails to deflect the chain as described above. It should not take much to do this. Do not overtighten as damage to the chain, sprocket and joint bearings could result.

4) After the chain has been tightened, apply a small drop of Loctite medium strength retaining compound (green colour) to each screw in the tensioner. Use sparingly and ensure that none gets on the chain.

5) Apply a molybdenum compound heavy weight oil to the chain. Use sparingly. Because of the slow speeds the chain must endure, this should be sufficient lubrication. However, under severe operating conditions, a moly grease may be more suitable.

ITEM: Check inner wiring harness

INTERVAL: 1000 hours

TOOLS REQUIRED: - #0 Robertson screwdriver

ACTION:

1) Remove the front shoulder cover using a #0 Robertson screwdriver.

2) Visually inspect the wiring harness inside looking for tears in the wire bundle nylon mesh cover.

3) Rotate the base (joint 1) all the way counter-clockwise and check for over-tightening of the harness. Rotate the other way and ensure that it does not bind anywhere.

4) At each end of travel, inspect the harness for tears as above. If any problems are seen, refer the problem to a factory trained service representative.

5) Replace the cover.

ITEM: Check inner wiring harness

INTERVAL: 1000 hours

TOOLS REQUIRED: - #0 Robertson screwdriver

ACTION:

1) Remove the front shoulder cover using a #0 Robertson screwdriver.

2) Visually inspect the wiring harness inside looking for tears in the wire bundle nylon mesh cover.

3) Rotate the base (joint 1) all the way counter-clockwise and check for over-tightening of the harness. Rotate the other way and ensure that it does not bind anywhere.

4) At each end of travel, inspect the harness for tears as above. If any problems are seen, refer the problem to a factory trained service representative.

5) Replace the cover.

ITEM: Check arm covers for wear/cracking

INTERVAL: 2000 hours

TOOLS REQUIRED: - #0 Robertson screwdriver

ACTION:

1) A visual inspection of all flexible covers should be carried out. any signs of cracking or excesive wear should be considered problems. Excessive wear can be detected as a whitening of the covers.

2) Of course the covers will not affect the performance of the robot, but they are a useful safety feature and protect the robot from dirt and small objects entering the mechanism.

3) Defective covers should be replaced.

ITEM: Re-lubricate the Harmonic Drives

INTERVAL: 2000 hours

TOOLS REQUIRED: - #0 Robertson screwdriver
- Bevel tipped grease gun
- Molybdenum based H/P gear grease

ACTION:

The Harmonic Drives are packed with grease before leaving the factory. During heavy operation, some grease will break down and run out. Re-packing is only necessary every 8000 hours and is a major operation. In the meantime, the supply of grease in each drive should be kept sufficiently high to prevent a lack of lubrication to the spline surfaces. This is described below.

1) There are three Harmonic drives in the M1 robot. Each will need attention. The grease fitting for the waist drive (joint 1) is located on the top of the base casting at the rear.

2) The grease fittingd for the drives for joints 2 and 3 are located behind the front shoulder cover on the front of the cast shoulder supports. They can be accessed by removing the cover.

3) In each case, the ports lead into the chamber around the Harmonic Drive. A grease gun with a standard fitting will be needed to pump the grease into the port. The amount pumped in should be just enough that the back pressure forces the grease back out past the tip of the gun. Pump the grease in slowly to avoid overfilling.

ITEM: Check motor brush wear.

INTERVAL: 5000

TOOLS REQUIRED: - 1/4" slot screwdriver
- 1/4" offset slot screwdriver

ACTION:

1) The brushes for each motor are located on the outside of the rear bell-housing of each unit. The only motor which will give accessability problems is the joint 1 motor. This one is located inside the base casting. Its brushes are accessable with an offset screwdriver.

2) The brushes should be removed, inspected, and replaced one at a time to prevent re-assembling them incorrectly. Putting the wrong brush into the housing will result in the motor running away uncontrollably when power is applied. <u>This would be devastating</u> and must be avoided!

3) The Brushes should be routinely changed every 8000 hours or if the brush length is less than 0.19 inches from the seat of the spring to the tip.

ITEM: Inspect clean and refit Harmonic Drives

INTERVAL: 8000 hours

TOOLS REQUIRED: Field service kit.

ACTION:

NOTE: This repair to be performed only by a factory trained service representative

1) Remove the motor by undoing the four #10 socket cap screws in the mounting flange. Slide the motor carefully out - do not force it. If resistance is encountered, rotating the robot joint while pulling gently on the motor will aid its release.

2) Remove the circular spline. Joints 2 and 3 splines can be removed by simply removing the shoulder support for that axis. The spline for joint one can be removed by prying it with a sharpened, bent screwdriver.

3) Removal of the circular spline permits a clear view of the flex spline teeth. Clean away all grease on the spline teeth of both the flex and circular splines. Inspect the teeth for signs of wear or damage.

4) Clean the grease away from the wave generator on the motor shaft. Inspect the outer surface of the bearing race for signs of wear. Remove the snap ring and take out and clean the oldham coupling. Inspect the mating surfaces for wear and check the coupling mesh for play. Grease the coupling parts prior to re-assembly. It may also be required to re-pin the circular spline to the casting. This will be needed if backlash is apparent in the drive. Drill a 1/8" hole through two of the circular spline mounting holes to a total depth of no more then 0.656 inches. Install a 0.125 dia by 0.625 long roll pin.

5) Re-fit the harmonic drives carefully. Again, installation of the motor can be aided by rotating the joint as you push gently on the motor. Grease the harmonic drives through the grease ports. About 25% more grease is needed in the base drive than joints 2 and 3.

6) Re-calibrate the robot and check the homing marks. Dis-assembly of the drives may change the relative location of the encoder index in which case the homing marks will need to be changed.

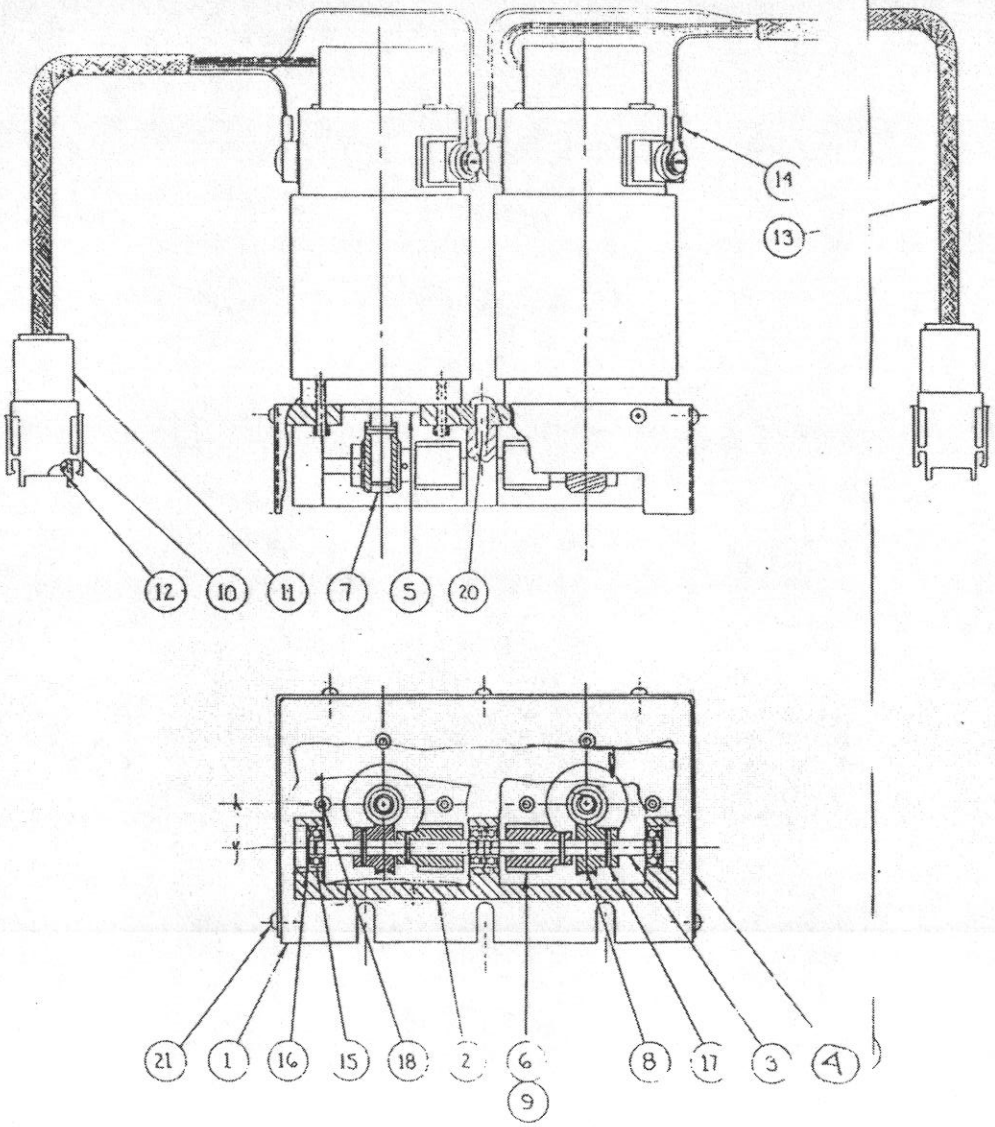
C-2 ELECTRONIC TROUBLESHOOTING

Symptom	Action
1. Video Terminal not active while Main Power on.	1a. Turn Main Power Off 1b. Turn Terminal Power Off. 1c. Check RS232 Cable. 1d. Turn Terminal On. 1e. Press space bar at Terminal. 1f. Turn Robot Controller on. 1g. If still not working try '2'.
2. Video Terminal not active while Main Power on.	2a. Turn Main Power Off 2b. Turn Terminal Power Off. 2c. Turn Terminal On. 2d. Press space bar at Terminal. 2e. Hold Teach button on Pendant. 2f. Turn Robot Controller on.
3. Arm Power can not be turned on.	3a. Issue ARM ON command at terminal. 3b. Press Arm power 'on' switch. 3c. If unsuccessful try '4'.
4. Arm Power can not be turned on.	4a. Ensure J2 (Motherboard) and Arm power board 8-pin plugs are seated properly. 4b. Ensure that +12VDC and ground on arm power board are present.
5. Axes run away when arm power turned on	5a. Check the base plug on the robot. 5b. Check all motor plugs. 5c. Issue a W1 command, move the joint under test by hand, and observe the change in motor pulses on the VDT. If motor pulses are changing in both directions, then check the fuse in the DC amplifier module. 5d. Ensure that the Vcom connectors on the motherboard are seated properly. 5e. Check voltage regulator fuses F1 and F2. 5f. Ensure encoders +5VDC and Ground are present on the motherboard. 5g. Ensure that the encoder plugs on the motherboard are seated properly (AXIS1 to AXIS5).

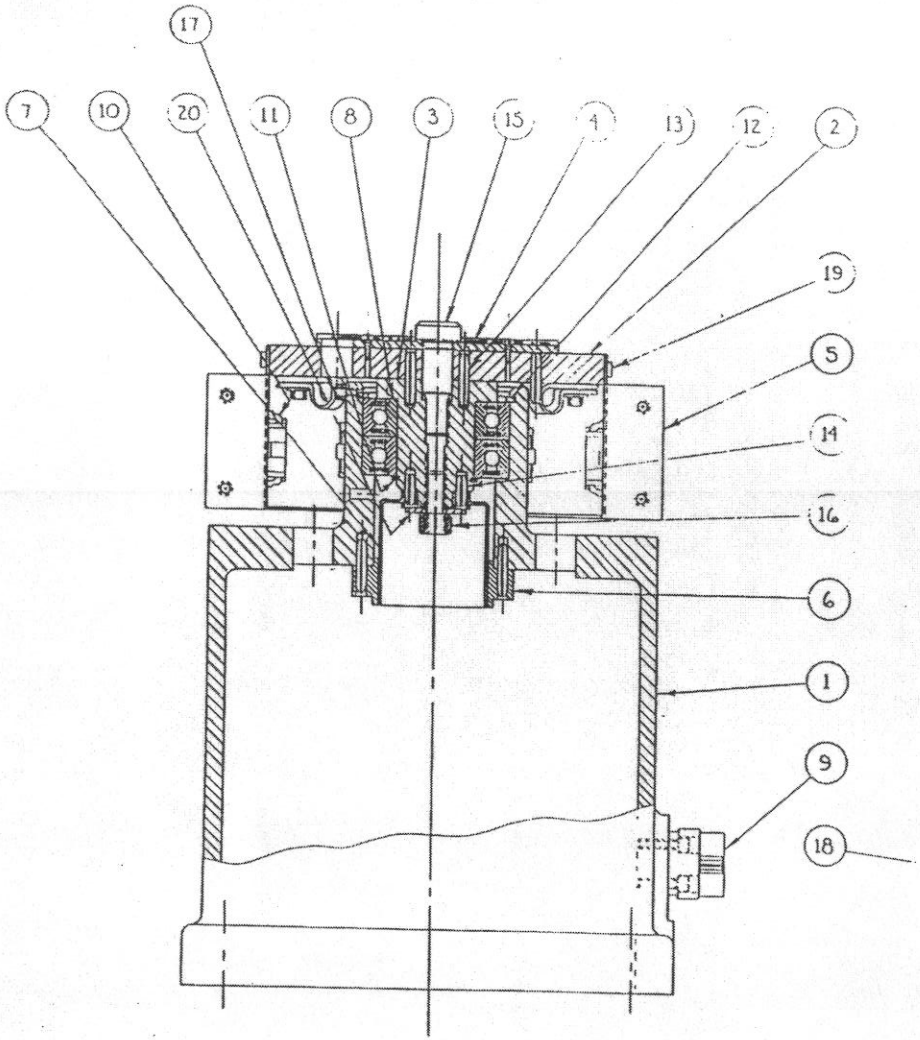
C-2 ELECTRONIC TROUBLESHOOTING

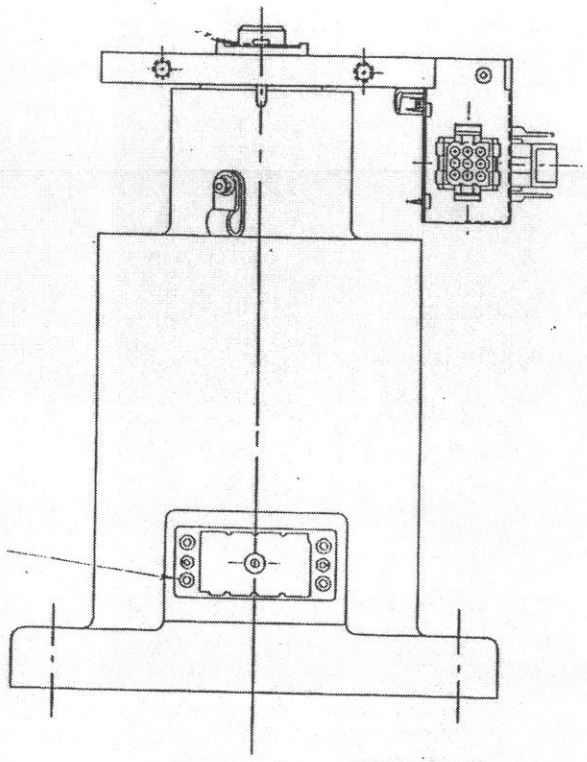
Symptom	Action
6. Arm power is on, but robot is not operational.	6a. Issue a NOLIMP command. 6b. Check voltage level at ESTOP pin J6 on the motherboard. It should be less than 0.8 VDC.
7. Arm power is on, but one (1) joint is not operational.	7a. Turn arm power off. 7b. Check motor plug. 7c. Check fuse in amplifier module 7d. Issue a W1 command, move the joint under test by hand, and observe the change in motor pulses on the VDT. If motor pulses are changing in both directions, then check the fuse in the DC amplifier module. 7e. Check motor cables.
8. Digital I/O not working.	8a. Check cabling.

APPENDIX D
MECHANICAL DRAWINGS

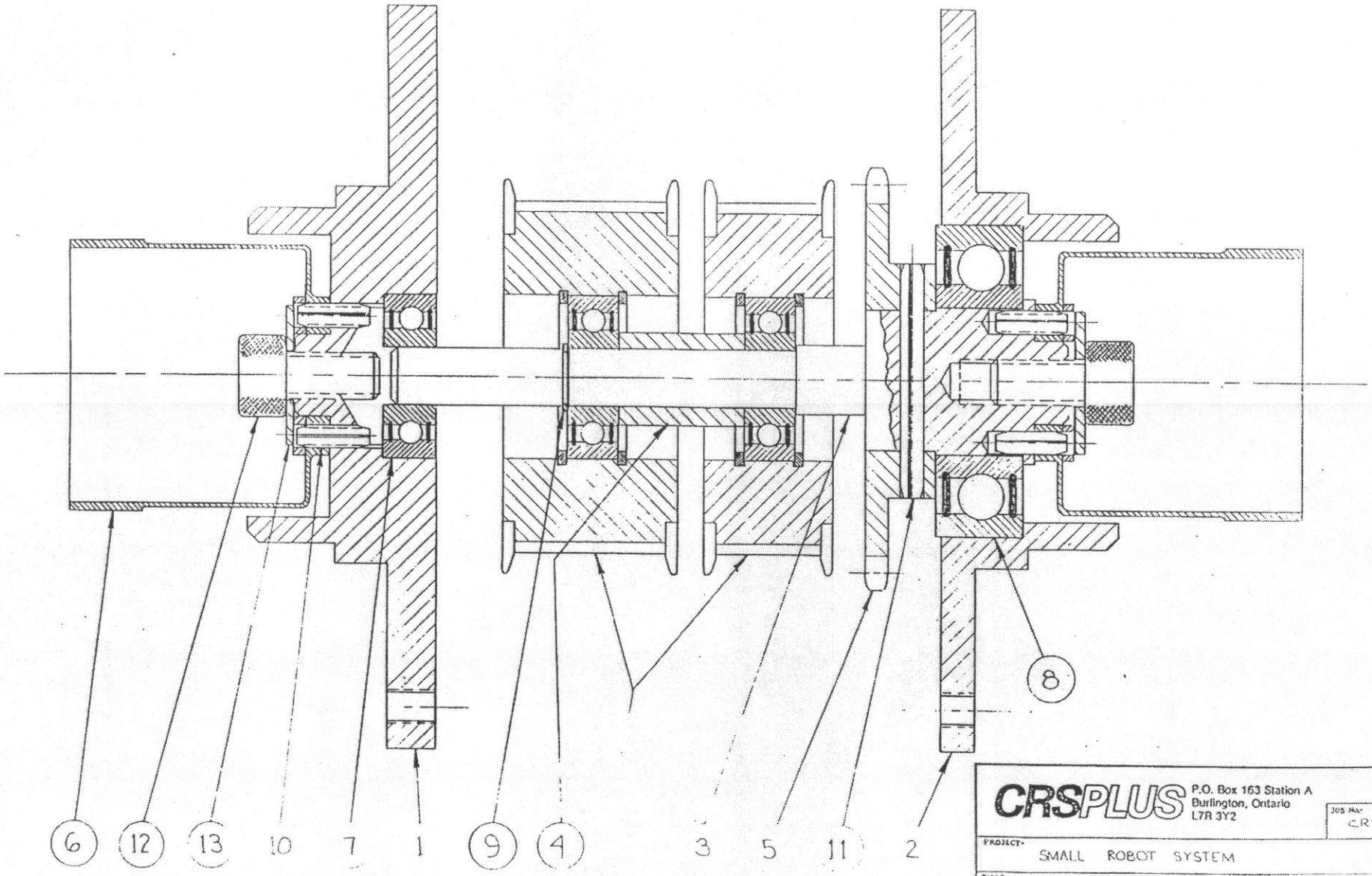


CRS PLUS			CRS 009
SMALL ROBOT SYSTEM			
WRIST MOTOR ASSEMBLY			
	DATE	NAME	DWG NO
DRAWN	OCT 15/85	J. RATHERLEY	SMC-9-002
CHECKED			
SCALE	FULL SCALE		



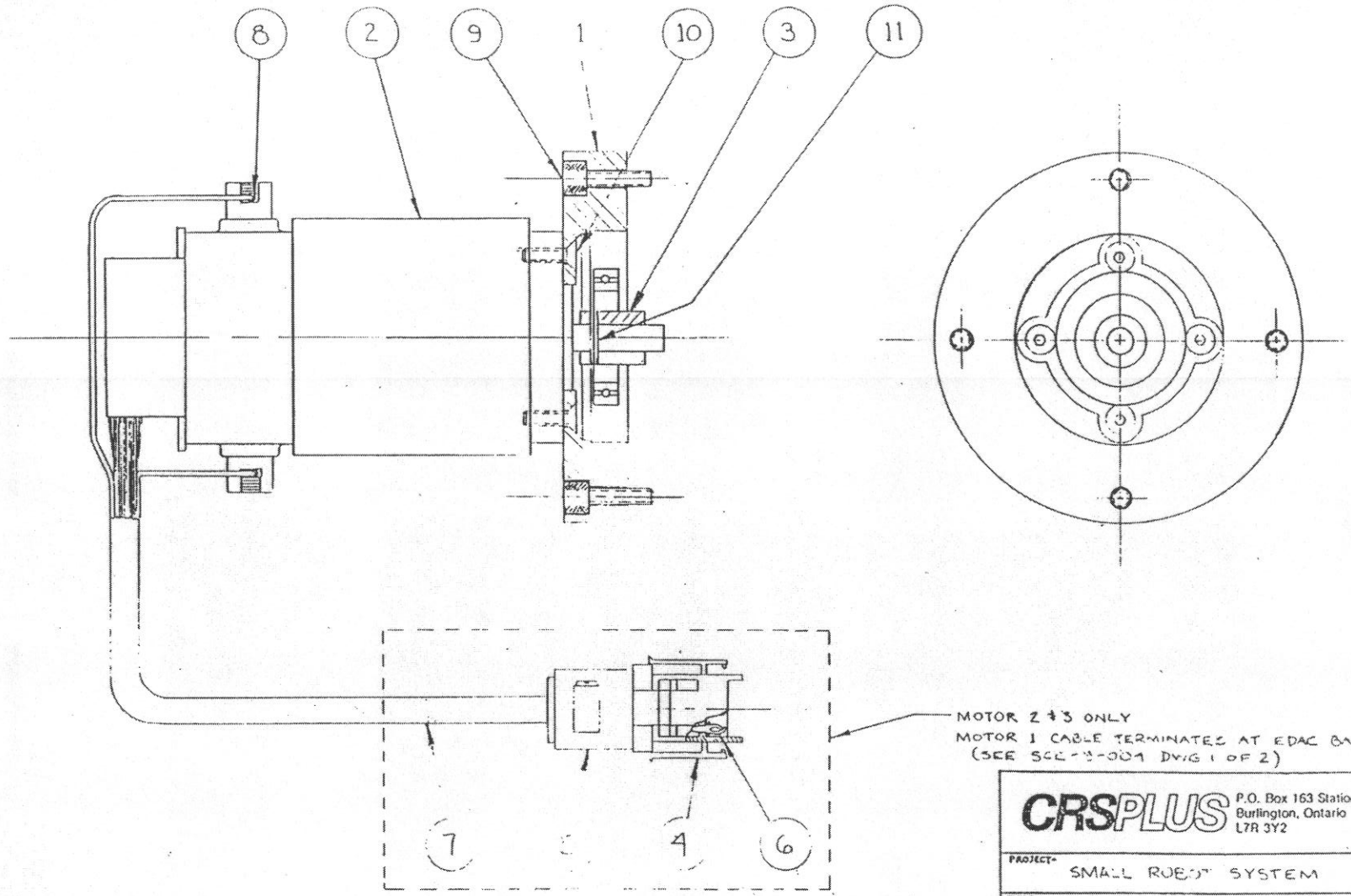


CRS PLUS			CRS 009
PROJECT: SMALL ROBOT SYSTEM			
TITLE: BASE ASSEMBLY			
	DATE	NAME	DWG NO
DRAWN	OCT 10/85	J. MATHERLY	SMC-9-001
CHECKED			



SEL ASSEMB SMA-9-005

CRSPLUS P.O. Box 163 Station A Burlington, Ontario L7R 3Y2			JOB No. CRS 009
PROJECT: SMALL ROBOT SYSTEM			
TITLE: JOINT 2 ASSEMBLY			
	DATE	NAME	DWG. No. SMB-9-006
DRAWN	OCT 9/85	I. HATHERLEY	
CHECKED			
SCALE	2 x FULL SCALE		



CRSPLUS P.O. Box 163 Station A Burlington, Ontario L7R 3Y2

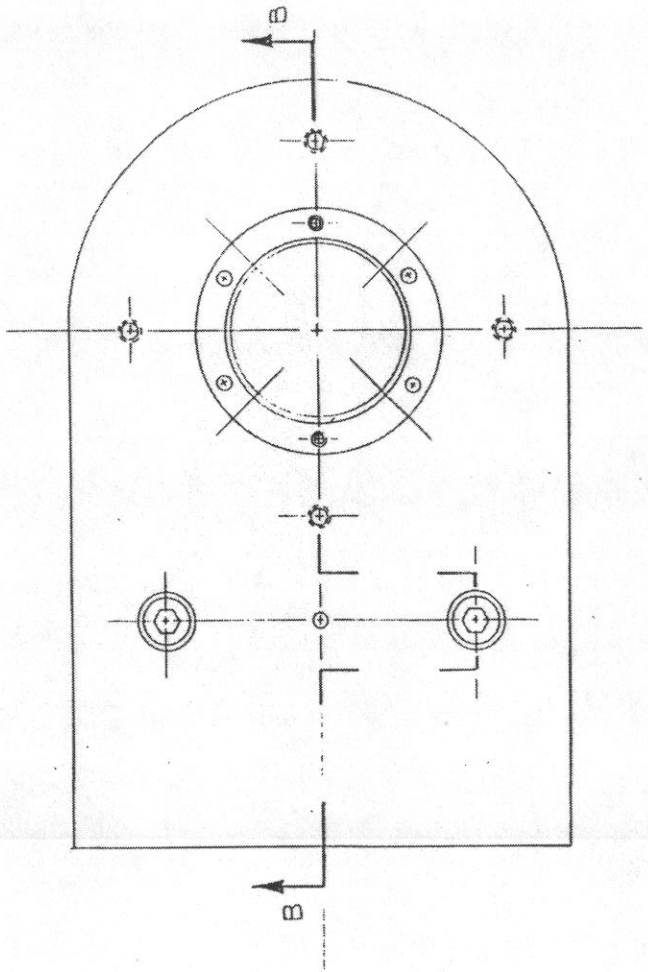
Job No: CRS 009

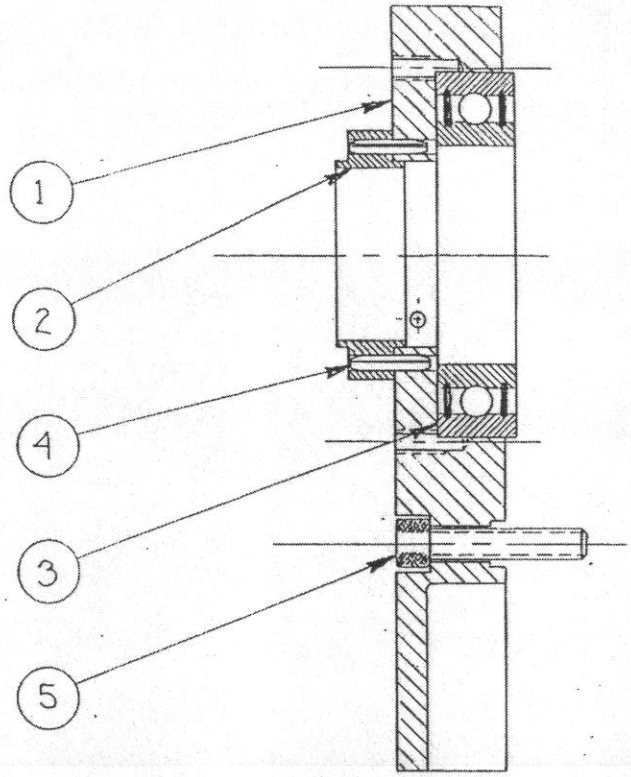
PROJECT: SMALL ROBOT SYSTEM

TITLE: MOT. F/FLANGE ASSEMBLY

	DATE	NAME
DRAWN	OCT 10/84	[illegible]
CHECKED		
SCALE	FULL SCALE	

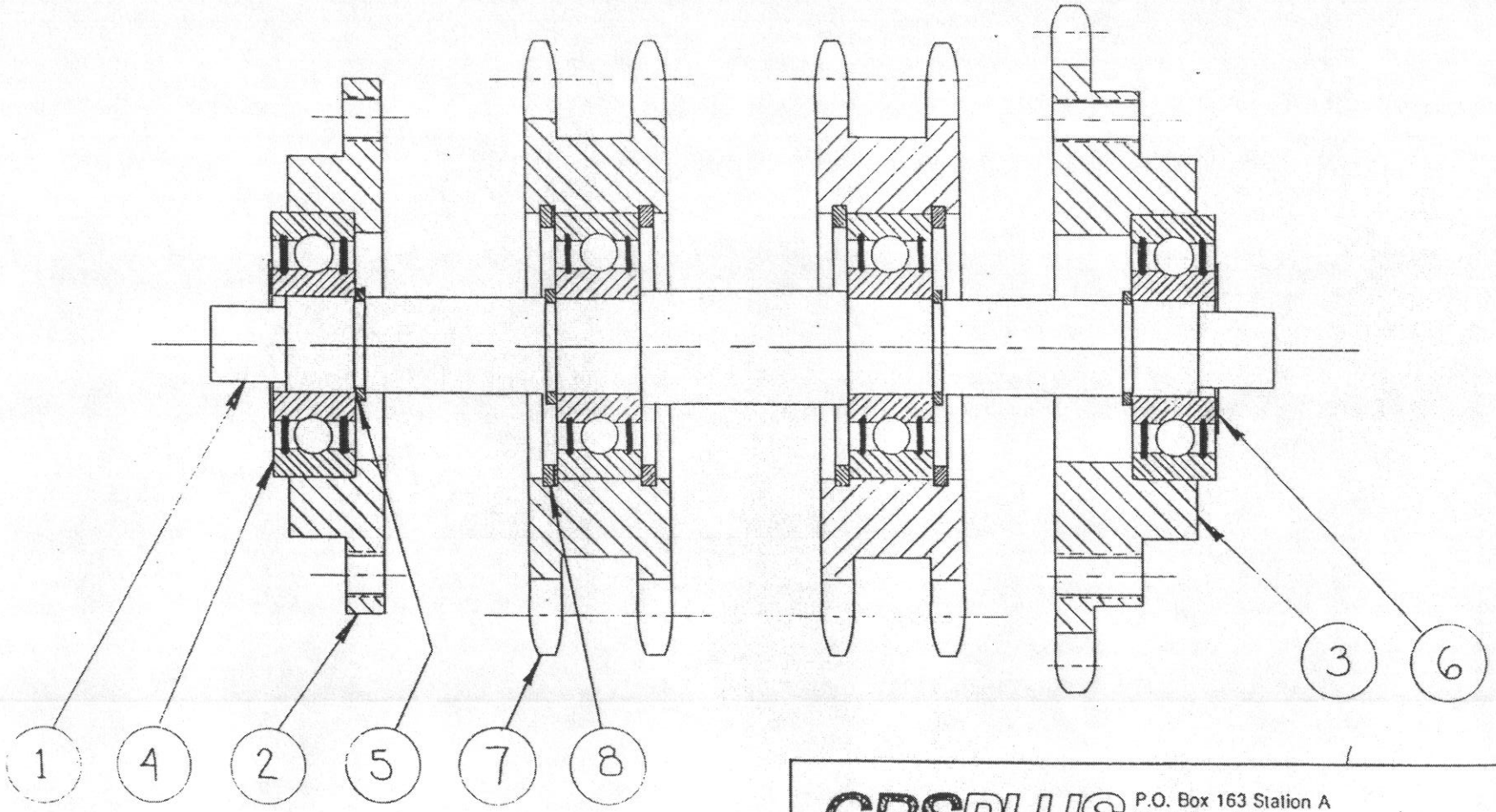
DWG. No: SMB-9-00[illegible]





SECTION B-B

CRSPLUS P.O. Box 163 Station A Burlington, Ontario L7R 3Y2			JOB No: CRS 009
PROJECT- SMALL ROBOT SYSTEM			
TITLE- SHOULDER ASSEMBLY			
	DATE	NAME	DWG. No- SMB-9-004
DRAWN	OCT 9/85	I. HATHERLEY	
CHECKED			
SCALE	FULL SCALE		



CRSPLUS

P.O. Box 163 Station A
Burlington, Ontario
L7R 3Y2

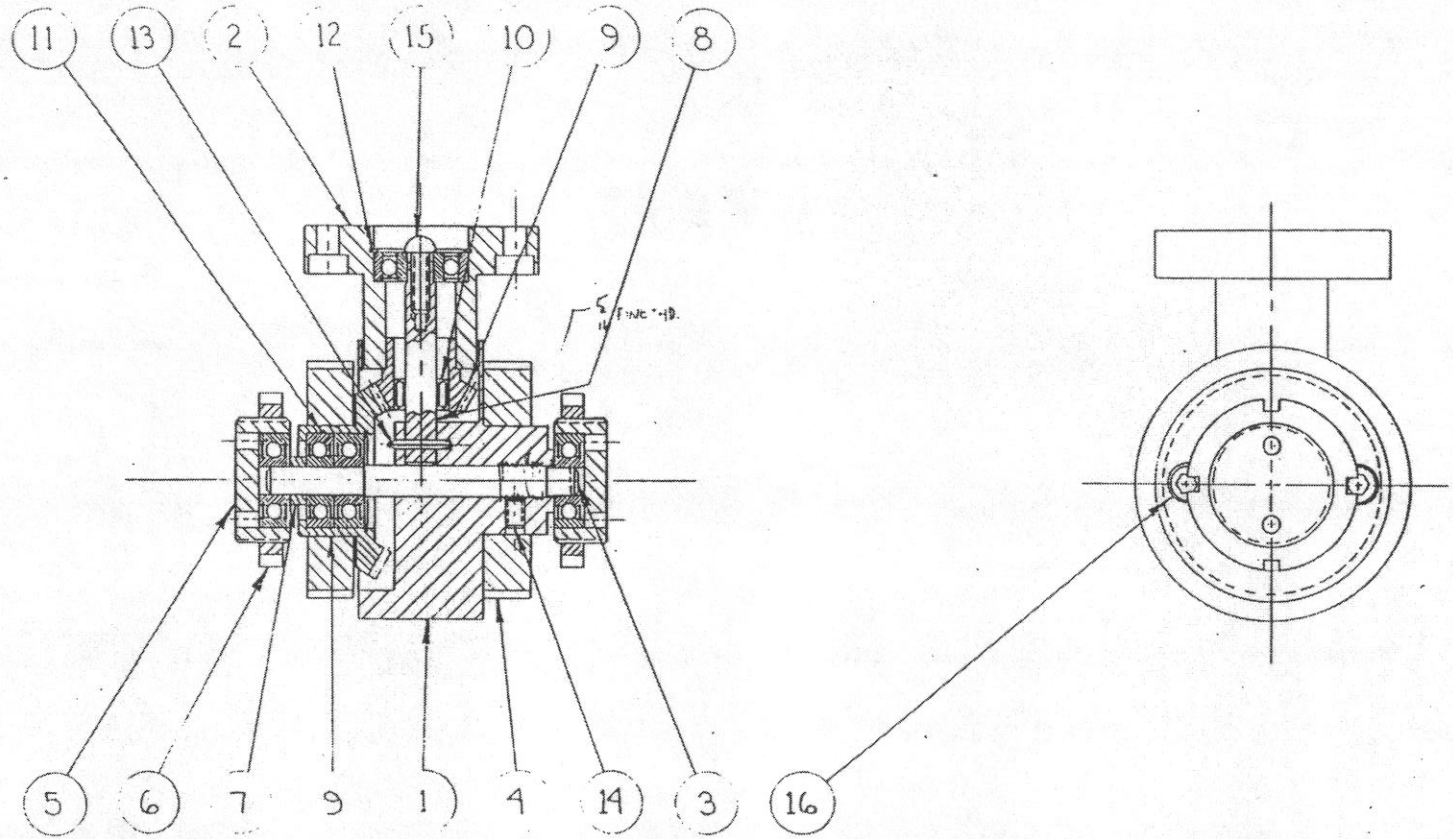
JOB No.- CRS 009

PROJECT- SMALL ROBOT SYSTEM

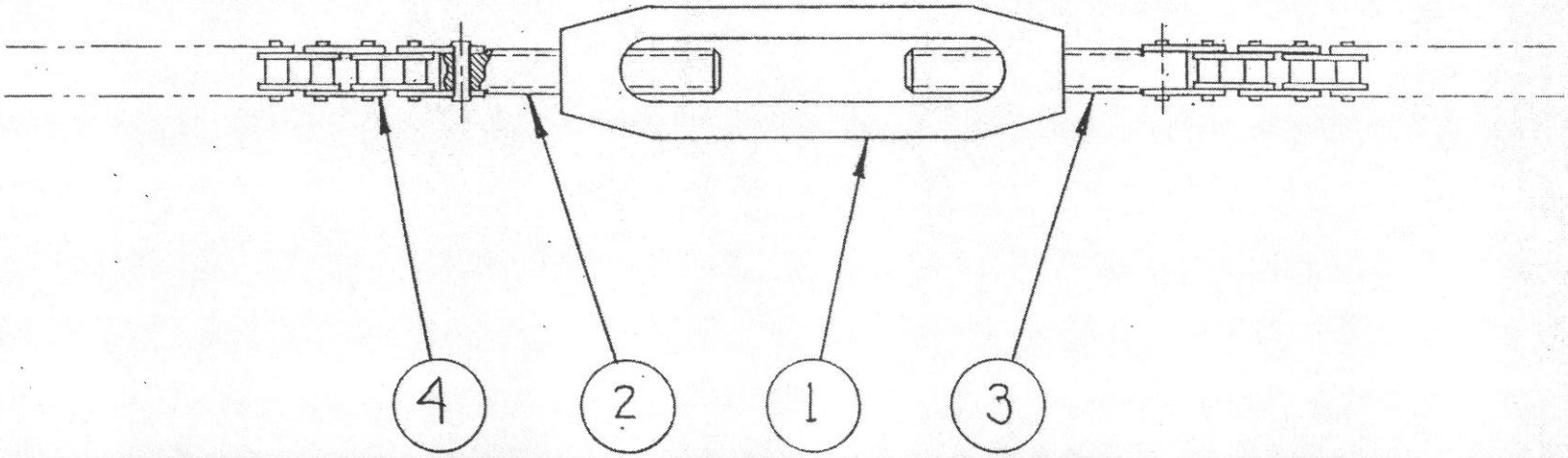
TITLE- JOINT 3 ASSEMBLY

	DATE	NAME
DRAWN	OCT 8/85	I. HATHERLEY
CHECKED		
SCALE	2 X FULL SCALE	

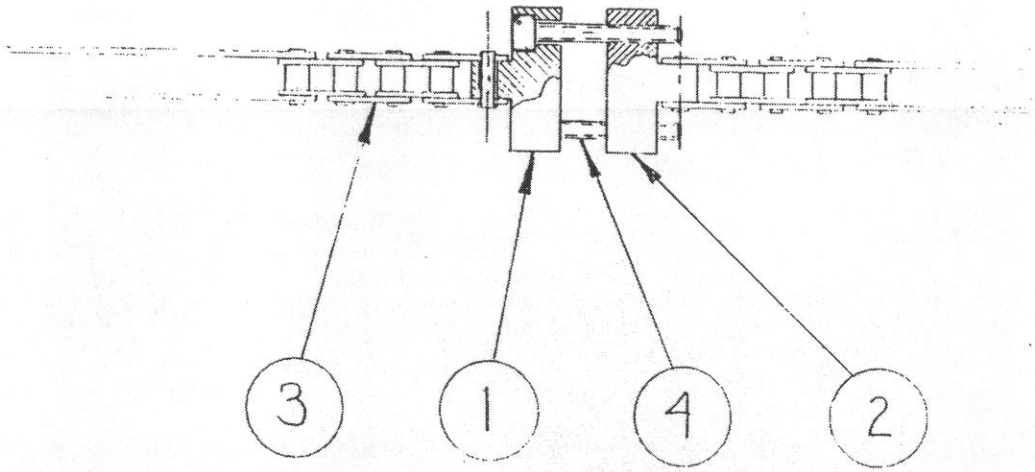
DWG. No.- SMA-9-007



CRSPLUS P.O. Box 163 Station A Burlington, Ontario L7R 3Y2			JOB No: CRS 009
PROJECT- SMALL ROBOT SYSTEM			
TITLE- WRIST ASSEMBLY			
	DATE	NAME	DWG. No- SMB-9-009
DRAWN	OCT 15/85	T. HATHERLEY	
CHECKED			
SCALE	FULL SCALE		



CRSPLUS	P.O. Box 163 Station A Burlington, Ontario L7R 3Y2		JOB No.- CRS 009
PROJECT- SMALL ROBOT SYSTEM			
TITLE- TENSIONER ASSEMBLY			
	DATE	NAME	DWG. No.-
DRAWN	NOV 18/85	I. HATHERLEY	



CRSPLUS

P.O. Box 163 Station A
Burlington, Ontario
L7R 3Y2

JOB No.- CRS 005

PROJECT- SMALL ROBOT SYSTEM

TITLE- TENSIONER ASSEMBLY - LINK 2

	DATE	NAME
DRAWN	JAN 6/86	I HATHERLEY
CHECKED		
SCALE	FULL SCALE	

DWG. No.- SMA 5 01

APPENDIX E
OPTICAL ENCODER SPECIFICATIONS

The SRS-M1 controller can be interfaced to optical encoders with the following specifications;

Two channel output with zero cross signal

Square wave output

5 volt DC power supply with 120 milliamp current typ.

Signal rise time 1.0 microsecond, max

Signal fall time 1.0 microsecond, max

Pulse frequency range DC to 50 Khz Max.

APPENDIX F - SRS-M1A CONTROLLER
TECHNICAL DESCRIPTION

[This APPENDIX contains detailed descriptions of the operation and architecture of the SRS-M1A Controller. It is intended for those who require specialized interfacing or plan operating system software modifications. For access to this document, please contact your distributor or CRS Plus.]