SRS-M1A

SMALL INDUSTRIAL ROBOT SYSTEM
IBM-PC HOST INTERFACE

Manual Order Number: SRS-MAN/COMM

Copyright 1987, 1988 CRS Plus Inc
835 Harrington Ct., Burlington, Ontario, L7N 3P3
CANADA

REV.	REVISION HISTORY	DATE
-001	Original Issue	05/86
-002	Version 2.1 of ROBCOMM	02/87
-003	Version 4.2 of ROBCOMM	11/87

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

CRS Plus Inc.
P.O. Box 163 Station A
Burlington, Ontario
L7R 3Y2
CANADA

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.

PREFACE

This manual provides general information, installation instructions, programming information, and principles of operation for the CRS Plus, IBM-PC to SRS-M1A, Small Industrial Robot System interface package (the ROBCOMM software package). Additional information on other CRS Plus products is available in the following documents:

* RAPL Programming Manual

* SRS-M1 Small Industrial Robot System - Tutorial Manual

* SRS-M1 Small Industrial Robot System - Technical Manual

CONTENTS

CHAPTER 1 - INSTALLATION

1-1 INTRODUCTION 1-1
1-2 STANDARD INSTALLATION 1-2
Floppy-Disk Based System 1-2
Hard-Disk Based System 1-3
1-3 DETAILED SOFTWARE INSTALLATION 1-4
1-4 HARDWARE INSTALLATION 1-6
IBM Configuration 1-6
Cables 1-6
1-5 STARTING THE SYSTEM 1-8
1-6 SETUP OF ROBOT CONTROLLER HARDWARE 1-9
1-7 ROBCOMM DEFAULT CONFIGURATION 1-11
1-8 IBM-PC BAUD RATE SETTING 1-12

CHAPTER 2 - OPERATION

2-1 WINDOWS 2-1
2-2 MENU FORMAT 2-3
Main Menu 2-3
Setup Menu Options 2-4
Terminal Menu 2-8
Baud Rate Menu Options 2-10
Data Save Menu Options 2-11
Load Data Menu Options 2-14
Check Duplicate Menu Options 2-17
Utilities Menu Options 2-19
Memory Menu Options 2-24
2-3 ENDING THE PROGRAM 2-26

CHAPTER 3 - ERRORS

3-1 ERRORS 3-1
Disk Access Errors 3-1
Communication Errors 3-1
Syntax Errors 3-1
RAPL Errors 3-1

CHAPTER 4 - UTILITIES

4-1 INTRODUCTION 4-1
4-2 READCAL 4-2
4-3 LOADCAL 4-3

APPENDIX A - ERRORS

A-1 COMMUNICATION ERROR CODES A-1
A-2 SYNTAX CHECKER ERRORS A-3
A-3 SYSTEM ERRORS A-5
A-4 ROBCOMM ERRORS A-6

APPENDIX B - JOB FILE STRUCTURE

NUMBER - TITLE PAGE

1-1 - Connection from IBM PC RS-232 port to Robot port 1-7
1-2 - Connection to Robot Device 1 port for use with RTS-CTS. 1-7
2-1 - Window placement. 2-1
2-2 - MAIN menu options. 2-4
2-3 - SETUP menu options. 2-5
2-4 - The Configuration menu and display. 2-7
2-5 - TERMINAL menu options. 2-8
2-6 - BAUD RATE menu options. 2-10
2-7 - DATA SAVE Menu, shown ready to save a program. 2-11
2-8 - LOAD DATA Menu showing a program ready to load. 2-14
2-9 - Check Duplicate Menu and Display 2-17
2-10 - UTILITIES Menu showing the Renumber option 2-19
2-11 - Typical MEMORY EDIT display 2-25
2-12 - Typical MEMORY VIEW display 2-26
4-1 - Display during operation of READCAL program. User input **BOLD** . 4-2
4-2 - Display during operation of LOADCAL program. User input **BOLD** . 4-3

1-1 INTRODUCTION

This software package provides the ability to permanently save CRS Plus robot system programs, locations, and variables to an IBM-PC or 100% compatible computer and reload them to the SRS-M1A robot controller. This is achieved through the use of the ACI (ADVANCED COMMUNICATION INTERFACE).

Equipment required for successful operation of the CRS Plus ROBCOMM package is as follows; An IBM-PC or 100% compatible personal computer running version 3.10 or higher of the MS-DOS or PC-DOS operating system. Hardware configuration should include a minimum of 256 kbytes of available RAM memory, at least one 360K disk drive, and a serial port with interrupt capability (Device COM1: or COM2:) for communications.

The software will run on either the PC-XT or PC-AT model of computer. The ROBCOMM package is compatible with Hercules graphics adapters, monochrome text adapters, CGA and EGA colour graphic adapters.

The system software is supplied on two types of diskettes: a 5 1/4 inch and a 3.5 inch both formatted for use with the MS-DOS operating system. The contents of these two diskettes are identical. Users should be fully familiar with the use of the DOS system before attempting to use this software.

To install this system on a standard IBM PC/XT, follow the list given in section 1-2. For other non-standard systems, the other sections provide more detail on the installation of the package. If any of the DOS commands used in the installation procedure are new to you, refer to the DOS User's Guide.

1-2 STANDARD INSTALLATION

In this section, a quick installation procedure is outlined. At each step, a reference is included (in square brackets: []) as to where more detailed information can be found. If problems occur at that step, the user should read the reference information as an aid to solving the problem.

This section contains two sets of instructions: one for a dual floppy disk based system (drives A: and B:), and the other for a system with a hard disk (set to drive C:). For both machines, the instructions assume a single serial port on the PC configured to COM1 and using a 25-pin subminiature male receptacle (DB25-R).

Floppy-Disk Based System

1. Boot the computer from the DOS system disk. [DOS User's Guide]

2. Insert a new diskette into drive B: and Issue the DOS "FORMAT B:/S" command to make a new system diskette. [DOS User's Guide]

3. Replace the DOS diskette with the ROBCOMM diskette and copy all files to the newly formatted diskette in B:. Use the DOS "COPY A:*.* B:" command. Put the new ROBCOMM system diskette into drive A:. Put the CRS diskette away in a safe place. [DOS User's Guide]

4. Connect the female end of the 3' male-female ribbon cable to the 25-pin serial port on the rear panel of the PC. Connect the other end to the "OUT" port of the supplied Serial Switch Box. Connect the two male-male ribbon cables between the Serial Switch Box and the serial device ports on the rear panel of the robot controller: Port A to Device 0, and Port B to Device 1. [Section 1-4 this manual]

5. Put the switch on the Serial Switch Box to B. Turn the robot controller main power to ON. [Section 1-7 of this manual]

6. Put a formatted (non-system) diskette into drive B: to hold working robot files (RAPL programs, locations and variables).

7. From the DOS prompt, type "READCAL<Enter>". Answer the questions as asked: "Y", "2400", "1", "A:". When asked for the serial number of the robot, use the 4-digit number found on the base casting of the robot arm below the main connector. The calibration values for your robot should be stored on the diskette in drive A:. [Section 4-2 of this manual]

8. Run ROBCOMM from drive A: by typing "A:ROBCOMM<Enter>". [Section 1-5 of this manual]

9. When the display appears, press <Ctrl-T> to get into the TERMINAL menu. Set the Serial Switch to A and press <Esc>, then <E> to Emulate. The PC is now emulating a terminal and controlling the robot directly. [Section 2-5 of this manual, RAPL User and Tutorial Manuals for information on RAPL usage]

Hard-Disk Based System

1. Boot the computer from the hard disk. [DOS User's Guide]

2. Insert the ROBCOMM diskette in drive A:.

3. Run the HINSTALL batch program from the CRS diskette in drive A: by typing "A:HINSTALL<Enter>". This program creates a directory called C:\ROBOT for the robot operating files and copies the diskette files to that directory. [DOS User's Guide]

4. Put the CRS diskette away in a safe place.

5. Connect the female end of the 3' male-female ribbon cable to the 25-pin serial (male) port on the rear panel of the PC. Connect the other end to the "OUT" port of the supplied Serial Switch Box. Connect the two male-male ribbon cables between the Serial Switch Box and the serial device ports on the rear panel of the robot controller: Port A to Device 0, and Port B to Device 1. [Section 1-4 this manual]

6. Put the switch on the Serial Switch Box to B. Turn the robot controller main power to ON. [Section 1-7 of this manual]

7. Select a directory in which to store working robot files (RAPL programs, locations and variables) and change to that directory. [DOS User's Guide]

8. From the DOS prompt, type "READCAL<Enter>". Answer the questions as asked: "Y", "2400", "1", "A:", and, when asked for the serial number of the robot, use the 4-digit number found on the base casting of the robot arm below the main connector. The calibration values for your robot should then be stored in the current directory. [Section 4-2 of this manual]

9. Run ROBCOMM by typing "ROBCOMM<Enter>". The install program sets up the DOS PATH to contain the directory "C:\ROBOT" so DOS will find the executable file from any current directory. [Section 1-5 of this manual]

10. When the display appears, press <Ctrl-T> to get into the TERMINAL menu. Set the Serial Switch to A and press <Esc> and then <E> to Emulate. The PC is now emulating a terminal and controlling the robot directly. [Section 2-5 of this manual, RAPL User manual and RAPL Tutorial Manual for information on RAPL usage]

1-3 DETAILED SOFTWARE INSTALLATION

There are three purposes to the installation of the ROBCOMM package. First, the PC must be able to find the system files on a disk. Second, installation must provide a place on a disk for working files. These are the files containing RAPL programs, and locations and variables used by the robot. Third, the communication parameters must be set up so the PC and the robot communicate correctly.

The ROBCOMM package is supplied on two non-system diskettes: one 5.25 inch and one 3.5 inch. Both diskettes contain the following files:

ROBCOMM.EXE	- The ROBCOMM program in DOS ".EXE" (executable) form.
ROBCOMM.CFG	- Configuration file used by ROBCOMM.
READCAL.COM	- Utility to read and store the robot calibration values.
LOADCAL.COM	- Utility to re-load robot calibration values from disk.
HINSTALL.BAT	- Installation batch program to put system on hard disk.
ROBOT.BAT	- Batch file to run ROBCOMM on hard disk.
ROBIVCMD.410	InValid CoMmanD files for each RAPL version
ROBIVCMD.350	
ROBIVCMD.345	
ROBIVCMD.330	
ROBIVCMD.125	
ROBIVCMD.120	
ROBIVCMD.110	
ROBCMD.410	Valid CoMmanD files for each RAPL version
ROBCMD.350	
ROBCMD.345	
ROBCMD.330	
ROBCMD.125	
ROBCMD.120	
ROBCMD.110	

As with any DOS executable or command files, ROBCOMM, READCAL, and LOADCAL are run by typing the file name in the form of a command line. However, DOS must be able to find the file on an accessible disk when the command line is entered.

Floppy-disk Based System

If a Floppy-disk based PC is to be used, the most effective way to operate the system is to create a DOS system disk containing the ROBCOMM system and all associated files. This permits "Booting" the computer from the same disk that will be used to run the program. Most floppy disk based PC computers have two drives: A: and B:. Correctly installed, ROBCOMM uses A: for the system files and B: for working files.

To install ROBCOMM on a DOS system disk, the computer must first be booted using a DOS system diskette. A new diskette to be used as the new ROBCOMM system disk should be inserted into drive B: and the DOS FORMAT command issued as below:

A>FORMAT B:/S<Enter>

The computer system will tell the user to insert a new diskette in drive B: and press the <Enter> key.

WARNING:
THIS PROCEDURE WILL DESTROY ANY INFORMATION ON THE DISK IN DRIVE B:

The '/S' argument in the FORMAT command will install the DOS system files at the start of the diskette to enable the computer to Boot from this diskette.

Next the files must be copied from the CRS Plus diskette onto the new system diskette. This is done by inserting the distribution diskette into drive A: and using the DOS COPY command to copy the ROBCOMM files onto the new system diskette. The command:

A>COPY A:*.* B:<Enter>

may be used. All files will be transferred. The original distribution diskette should be put away for safe keeping and the copy run at all times. The write-protect tab should not be removed from the original system disk.

Refer to section 1-5 for information on starting the ROBCOMM system.

B) Hard Disk Based System

To install the ROBCOMM system on a winchester (hard) disk based system, the ROBCOMM distribution diskette must be inserted into drive A: after the computer is booted normally. It will be useful to create a separate directory on the hard disk in which to keep the ROBCOMM system files. A good name for this directory would be \ROBOT. This directory can be created by typing:

C>MD \ROBOT<Enter>

For more information on directories and the partitioning of the hard disk, refer to the DOS User's Guide. The files from the diskette can be copied to this directory by typing:

C>COPY A:*.* C:\ROBOT<Enter>

Refer to section 1-5 for information on starting the ROBCOMM system.

1-4 HARDWARE INSTALLATION

IBM Configuration

The ROBCOMM package requires a fully IBM PC hardware compatible serial communication adapter. The standard DOS name for such an adapter is "COMn". In most PC systems, one or two of these devices are present. They are then labeled COM1 and COM2. DOS allocates specific hardware IO ports to support these devices and ROBCOMM must know which device is being used to be able to communicate through it. ROBCOMM does not use the standard DOS software communication support as this is too slow for normal operation. It utilizes the following IBM interrupt levels for operation:

Channel	Interrupt Request
COM1	IRQ4
COM2	IRQ3

All standard IBM communication adapters have interrupt capability. The user should consult the system installation manual in order to set the configuration of that port correctly.

Cables

The ROBCOMM interface package includes a Serial Switch Box, a 4-foot male-to-female ribbon cable and two 8-foot male-to-male ribbon cables. The female end of the shorter cable must be connected to the serial (COM1 or COM2) port on the back of the PC. The serial receptacle is usually a male 25 pin DB-25 connector. In some cases, a male DB-9 (9 pin) receptacle is used. In this case a conversion cable will be required. The conversion cable can usually be purchased from the PC dealer. The OUT port of the Serial Switch Box must be connected to the other (male) end of the short cable. The two male-to-male cables must be connected between the switch box and both ports of the robot controller with port A to device #0, port B to device #1.

If the PC has both COM1 and COM2 ports available, the system may be used without the Serial Switch Box. In this case, male-to-female cables (available from CRS Plus or your distributor) can be used to connect the COM1 port to Device 0 of the robot and COM2 to Device 1. Once again, one or two 9 to 25 pin adapter cables may be required.

If wiring lengths greater than supplied and less then 30 feet are required, longer ribbon cables are available. If a non-ribbon cable must be assembled, the connection diagram of figure 1-1 may be used. If RTS-CTS protocol is desired, the connection diagram of 1-2 must be used.

Robot Controller		IBM-PC	
Function	Pin #	Pin #	Function
Rxd	2	2	Txd
Txd	3	3	Rxd
CTS*	4	4	RTS*
RTS*	5	5	CTS*
		6	DSR
GND	7	7	GND
		20	DTR

Figure 1-1 - Connection from IBM PC RS-232 port to Robot port

Robot Controller		IBM-PC	
Function	Pin #	Pin #	Function
Rxd	2	2	Txd
Txd	3	3	Rxd
CTS*	4	4	RTS*
RTS*	5	5	CTS*
		6	DSR
GND	7	7	GND
		20	DTR

Figure 1-2 - Connection to Robot Device 1 port for use with RTS-CTS.

1-5 STARTING THE SYSTEM

The system may be started in one of two ways. It may be executed by typing the system name in the form of a DOS command line, or from a special DOS batch file.

Command Line Operation

For floppy- or hard-disk based systems the ROBCOMM package can be executed from the DOS prompt by typing the name of the .EXE file if this file (ROBCOMM.EXE) is in the current directory or there is a PATH to the directory in which it is stored. The DOS command line would look like this:

```
C>ROBCOMM<Enter>
```

Batch File Operation

A batch file is a series of commands in a text file format. Running the batch file is the equivalent of typing all the commands in the file. The file is executed by entering the name of the batch file as a command line. The DOS User's Manual contains more information about DOS batch files.

If the PC will be dedicated to the robot, it is convenient to use the automatically executing 'AUTOEXEC.BAT' batch file for start-up. This is an example of such a batch file as would be used in a hard-disk based system:

```
DATE
TIME
CD C:\ROBOT\FILES
PATH C:\ROBOT
ROBCOMM
```

This file can be entered by booting the computer normally and typing the following when the prompt appears:

```
COPY CON \AUTOEXEC.BAT<cr>
DATE<cr>
TIME<cr>
CD C:\ROBOT\FILES<cr>
PATH C:\ROBOT<cr>
ROBCOMM<F6> (function key 6 will echo a ^Z when pressed)
```

If the PC will not be dedicated solely to robot use, a batch file located in the "ROOT" directory of the hard disk is a convenient method for starting the program. The CRS floppy disk contains an example of this type called "ROBOT.BAT":

```
PATH C:\ROBOT
ROBCOMM
```

This file can be executed from any current directory by typing "C:\ROBOT".

1-6 SETUP OF ROBOT CONTROLLER HARDWARE

When ROBCOMM starts, the screen will appear as Figure 2-2. In the default condition, the screen is set to operate in monochrome only. The TERMINAL EMULATION mode of ROBCOMM may be used to set up the RAPL controller. If COM1 alone is being used (see section 1-4), proceed normally. However, if COM2 or both COM1 and COM2 are used, ROBCOMM must know it. In this case, the Config option of ROBCOMM must be used before the PC can be used as a terminal. Refer to section 1-7 for the use of this option.

Pressing <Ctrl-T> (control-T, entered by holding down the <Ctrl> key and pressing the <T> key) will bring up the TERMINAL menu which permits ROBCOMM to emulate a terminal. If using the Serial Switch Box, the switch must be set to A and <Esc> (the escape key) pressed (as instructed in the top of the screen) before proceeding.

The default BAUD rate used for both RAPL device #0 and ROBCOMM for terminal emulation is 9600. This will be displayed in the lower right corner of the display. If anything other than 9600 is shown, the "Baud Rate" option from the terminal menu may be used to change the rate to 9600. Refer to section 2-5 for more information on this menu item.

Once the correct baud rate has been selected, the Emulate item will actually emulate the terminal. At this point, the screen will go blank except for the top line which will read:

```
ROBCOMM ENHANCED TERMINAL EMULATION 4.2          CNTRL Z to QUIT
```

The PC system is now emulating a dumb terminal. Turning on the robot controller will permit normal operation of the RAPL system. After the RAPL sign-on message and prompt appear, the RAPL SERIAL command will display the current serial configuration of RAPL.

The ACI features of the RAPL controller must be initialized and enabled before communication with this package can take place. Versions of RAPL above 3.30 default to an ACI-initialized condition. However the user should understand the set-up procedures.

The procedure is outlined below:

1) <u>Configuration of device #1.</u>
   Note: all ACI communication is through device #1 of the robot controller. This port must be set to the correct baud rate, etc. before using the ACI features of the ROBCOMM package. The RAPL CONFIG command is used for this. Serial status can be checked using the SERIAL command. Both commands are part of the RAPL language of the controller.

The default attributes for device #1 are:

Baud rate	- 2400
Parity	- None
# Data bits	- 8
# stop bits	- 1
XON/XOFF	- off
RTS/CTS	- off
Echo	- off

The ACI communication may also be used at 4800 baud for RAPL versions 4.10 and up. Remember that for proper communication, both the PC and the robot must be set to the same baud rate.

2) Initialize the ACI
Use the RAPL "@@RI" (Remote Initialize) command to initialize the ACI. This prepares the RAPL system for ACI use.

3) Enable the ACI
Use the RAPL "@@RE" (Remote Enable) command to enable the ACI. This sets device #1 up for ACI use. Any attempts to use device #1 as anything but an ACI port until the "@@RD" (Remote Disable) has been executed will result in an error. RAPL versions prior to version 3.3 require re-enabling after each teach start.

4) Set the slave number
Each robot controller is treated as a slave in the 'Master/Slave' protocol of the ACI. The controller must be given a slave number between 1 and 127. As this implies, there can be up to 127 slaves communicating with one master. The RAPL "@@RN" (Remote Number) command is used to enter the slave number. The ROBCOMM package defaults to slave number 1 at start up. However, the number can be changed during operation using the ROBCOMM SETUP menu and the default number may be changed using the Config option.

The controller is now ready for operation in the ACI mode.

The status of ACI communication can be checked at any time using the "@@RS" (Remote Status) command of the RAPL system if a separate terminal is connected to serial channel #0. This will display the status of the current ACI communications as described in Appendix A of this manual.

1-7 ROBCOMM DEFAULT CONFIGURATION

On the ROBCOMM distribution diskette is a file called ROBCOMM.CFG. This is a text file containing the default configuration of some of the parameters of the ROBCOMM package. The file contains lines of text, each line having a configuration parameter, an equal sign, and an argument, followed by a carriage-return/linefeed.

The ROBCOMM.CFG file on the distribution diskette will have the following contents:

```
PASSWORD = NEW
PORT     = 1
DELAY    = 2
ROBOT    = 1
ACI      = 2400
TERMINAL = 9600
VERSION  = 410
COLOUR   = OFF
```

PASSWORD sets the ROBCOMM (not RAPL) system password to allow entry into the MEMORY and CONFIGURATION functions of the package. It has a maximum length of fifteen characters.

PORT determines which port will be used for communicating to the robot, selecting 1 causes all communication to take place using COM1, 2 causes all communication to take place using COM2, and B (Both) causes COM1 to be used for serial communication in the TERMINAL EMULATION MODE and COM2 to be used for the ACI, thus bypassing the need for the serial switch box.

DELAY determines how long the ROBCOMM software will display an error message in the ERROR window. The argument can assume a value between 0 and 9, which represent seconds approximately.

ROBOT sets the desired default robot slave number for communication in the ACI mode.

ACI sets the default baud rate for ACI communication mode. This must match the ACI (Device #1) baud rate setting for the robot controller before communication can take place.

TERMINAL sets the default baud rate for terminal emulation communication mode, this must also match the Device #0 baud rate setting for the robot controller.

VERSION sets the current RAPL command list for the check, tokenize, and untokenize utilities.

COLOUR sets the colour or monochrome display attribute.

1-8 IBM-PC BAUD RATE SETTING

The Baud Rate is the rate of transfer of information across the serial communication channel. It is expressed as a number of bits per second transferred and in most cases can be from 30 to 19,200 bps for the RS-232. The hardware which performs the actual communication must be initialized to send and receive characters at the desired rate. Both (or all) communicating devices on the same line must be set to the same baud rate.

There are two different rates used by the ROBCOMM package as there are two modes of communication with the robot controller: ACI and TERMINAL. When the program is started from DOS, the rates will be set to the values defined in the ROBCOMM.CFG file (see section 1-7). Normal values are 9600 for the TERMINAL EMULATION MODE and 2400 for the ACI mode.

The terminal baud rate is set from the BAUD RATE item in the TERMINAL menu using the ROBCOMM package. In this case, direct entry of a value is required. Legal values are 9600, 4800, 2400, 1200, 600, and 300. The result will be seen at the bottom of the display when the ROBCOMM system is in the TERMINAL menu.

The ACI baud rate in ROBCOMM is changed using the BAUD RATE item from the SETUP menu. This will permit several possible values from 300 to 4800 baud.

The most reliable transfer occurs at 2400 baud. However, RAPL Versions 4.10 and up contain an upgrade which makes transfer reliable at up to 4800 baud.

Whenever a baud rate is changed, the user must remember that both ends must be set to the same value. Check the rate of the robot controller by using the RAPL "SERIAL" command to display the parameters of the serial channels. Device #1 is the ACI channel and device #0 is the terminal channel. The rates may be changed using the RAPL "CONFIG" command.

Once the baud rates are selected, they will remain until they are changed or the program is terminated. The values set in the robot controller remain until changed or until a 'Teach Start' is performed.

2-1 WINDOWS

During operation, the system divides the PC display into several distinct areas called windows. These are illustrated in Figure 2-1 and described below. During system operation, the cursor is found in one of the windows at all times. The location of the cursor will tell the user what the system is doing or what it is waiting for.

```
CRS PLUS ROBOT COMMUNICATION PACKAGE 4.2
MENU =  <------------------- MENU WINDOW ------------------------->
WILL =  <-------- INSTRUCTION/MESSAGE/ERROR WINDOW ------------>
                          < LABEL WINDOW >

                          DATA WINDOW

LOG:
                     <- LOG WINDOW ->

     RAPL SYNTAX VERSION 4.10 LOADED
C:\ROBOT\FILES ------------------------- Slave 1  -- Baud Rate  2400
```

Figure 2-1 - Window placement.

MENU WINDOW - This is the area where the menu is displayed. Refer to the description of menu operation (section 2-2) for a full description of the menus and their use. When the cursor is in the MENU window, the system expects the user to make a selection from the currently displayed menu or use the <Ctrl-T> key to select terminal mode.

INSTRUCTION WINDOW - This area contains the current instructions. When a menu item has not yet been selected, this window describes the function of the highlighted item. After an item has been selected, this window contains the instructions relating to the selected function. When the cursor is in the INSTRUCTION window, the system is waiting for input from the user.

MESSAGE WINDOW - This area informs the user what the system is doing whenever no action is expected. For instance, many of the communications which take place involve data transfer times of several seconds. The MESSAGE window then displays what the system is doing. The message flashes to indicate that the system is active. The MESSAGE window has the same physical area as the INSTRUCTION window.

ERROR WINDOW - The ERROR window displays any error messages that occur during operation of the system. The error message is displayed for the time set by the delay parameter in ROBCOMM.CFG and is accompanied with a beep from the computer. In most cases, an error number is also displayed, allowing the user to diagnose the faulty condition. These errors are listed in Appendix A. The ERROR window has the same physical area as the INSTRUCTION window.

DATA WINDOW - The DATA window is a general purpose window. The use of this window depends on the function currently selected. It could be used for display of program listings, file or program directories, or for checking the existence of locations or variables. These functions are described in more detail in later sections of this manual.

LABEL WINDOW - This area describes the current use of the DATA Window.

LOG WINDOW - This area of the screen reports the last eight ROBCOMM operations.

2-2 MENU FORMAT

In operation, ROBCOMM uses a menu format to select a function. Once operation has commenced, the system displays a line of possible functions across the display in the MENU Window (see Figure 2-1). To select one of the functions, the <left cursor> and <right cursor> keys on the standard PC keyboard may be used to move the cursor to the desired function. The system highlights the current selection by displaying it in bold-face as an aid in locating the cursor. An explanation of the highlighted function will appear in the INSTRUCTION window below the MENU window. If the user is satisfied that the highlighted function is the correct one, it is selected by pressing the <Enter> key. The selected function is then executed.

An alternative to the above method is to enter the first character of any function. The function will be executed immediately without displaying the explanation in the INSTRUCTION window. If a letter is typed that does not match the first letter of any current menu item, the system remains unchanged.

Another way to select a function is to use the PC keyboard function keys. The menu items correspond with the function keys as numbered from left to right. Thus, the left most item in the current menu is selected using <F1> etc.. When a function key is pressed it causes the corresponding menu item to be executed without displaying the explanation in the INSTRUCTION window.

The 'path' to a desired function may not be direct. For instance, to save a robot program on disk, the user must first select the DATA SAVE item from the MAIN menu. This brings in the SAVE menu. From this menu the user must then select the PROGRAM item. The system then reads in a list of all robot programs in the robot directory. The user selects the desired program from this last menu (which is actually displayed in the DATA window). The system then takes over and saves the program. Many functions take several such steps to execute.

Main Menu

The MAIN Menu has several options which are shown in Figure 2-2.

Setup -	Changes the IBM communication environment.
Data Save -	Saves robot information to user specified disk files.
Load Data -	Loads information from disk files to the robot.
Utilities -	Modifies RAPL programs and accesses robot memory.
Quit -	Returns control to DOS and ends the ROBCOMM session.

```
CRS PLUS ROBOT COMMUNICATION PACKAGE 4.2
MENU =   Setup      Data Save  Load Data  Utilities  Quit
WILL =   Change Setup Parameters/Emulate Terminal.

LOG:

     RAPL SYNTAX VERSION 4.10 LOADED
C:\ROBOT\FILES                          Slave 1   — Baud Rate  2400
```

Figure 2-2 - MAIN menu options.

Setup Menu Options

Once the system has been started (refer to section 1-5), the cursor will be resting on the SETUP item of the MAIN menu. Pressing the <Enter> key at this time will select the SETUP menu. This contains several options:

Terminal -	Emulate a terminal in the DATA Window.
Baud rate -	Change ACI Communication Baud Rate.
Slave No. -	Change Slave Device.
Drive Spec -	Change Drive and/or Directory for working files.
Config -	Change the system default parameters.
Quit -	Return to Main Menu.

The function of each of these items is explained below. The SETUP sub-menu is shown in figure 2-3.

```
CRS PLUS ROBOT COMMUNICATION PACKAGE  4.2
MENU =  Terminal  Baud Rate  Slave No.  Drive Spec Config     Quit
WILL =  Emulate a Terminal in DATA Window.

LOG:

     RAPL SYNTAX VERSION 4.10 LOADED
C:\ROBOT\FILES                          Slave 1   — Baud Rate  2400
```

Figure 2-3 - SETUP menu options.

Terminal - The TERMINAL EMULATION MODE permits the PC to operate as a 'dumb' terminal. The PC system must be plugged in to device 0 on the robot rear panel as described in section 1-4. The TERMINAL item may be selected from the SETUP menu. ROBCOMM will instruct the user to select the terminal position on the Serial Switch Box and press the <Esc> key if only one communication port is being used. Using Device #1 as a terminal will not work and may cause an error in the ACI software in RAPL. Pressing the <Esc> key makes it more difficult to overlook changing the switch position. Once this is done, the TERMINAL menu will appear in the MENU window (refer to TERMINAL menu section).

The user may also go directly to the TERMINAL menu by pressing <Ctrl-T>, (holding the <Ctrl> key down while at the same time pressing the <T> key) while in any of the menus, except the TERMINAL menu.

Baud Rate - Selecting baud rate will display the BAUD RATE menu, refer to figure 2-6. For a discussion on the ACI baud rate, refer to section 1-8.

Slave Number - The Slave Number refers to the device code that each individual robot has on the ACI link. Valid slave numbers are from 1 to 127. ROBCOMM prompts the user to enter a slave number. It will refuse the value if it is out of range. An accepted value will be indicated by the slave number status field in the bottom row of the display.

Drive Spec - The robot communication package uses disk files to save programs, locations and variables. The disk/directory specifier allows the user to select the drive and the directory on the disk in which to find these ´working´ files. This is called the working directory.

When the system starts, the current default drive is shown in the bottom left corner of the display. When changing active drive only, the working directory will be the current directory of the new drive.

DOS allows the use of sub-directories on a single disk. This can make it easier to organize stored files, keeping directories to smaller more manageable sizes. For more information on the use of these directories, the user should consult a DOS manual.

To change the drive and/or directory, the user should select the DRIVE SPEC menu item from the SETUP menu. The INSTRUCTION window will prompt the user for a new directory name. Entering a letter from A to Z with a colon ´:´ will select only a new drive. To enter a new directory on the same disk as currently used, the user should enter the path with sub--directory names separated by back-slashes ´\´. The new path may also include a new drive letter followed by a colon. When this has been entered, the new selected directory will be shown in the bottom left corner of the display. The following are legal selections:

A:	Will select current directory of drive A
A:\	Will select root directory of drive A:
A:\SHOW	Will select directory \SHOW of drive A:
\SHOW	Will select directory \SHOW of current default drive.
\SHOW\TEST_1	Will select directory \SHOW\TEST_1 of current drive.
.\EXAMPLES or EXAMPLES	Will select the EXAMPLES sub-directory of current directory. For example, if the current directory is called A:\SHOW, the new directory will be A:\SHOW\EXAMPLES.

If a directory which does not exist is selected, ROBCOMM will issue an error message to that effect. It will then provide the opportunity to create a new directory. Pressing the <Y> key for yes in response to the prompt will permit the user to enter the new directory name. The current directory will be changed to the new directory.

Note that when creating new directories, DOS permits only one "level" to be created at a time. For instance, if the directory "\ROBOT" does not exist, "\ROBOT\FILES" cannot be created. ROBCOMM can create "\ROBOT" and then ".\FILES" to create the desired directory.

Config - This selection permits the user to change the default configuration of ROBCOMM. It displays these options in the DATA window (see Figure 2-4) and allows the user to edit them. To protect against unauthorized changes, ROBCOMM requires a password to be entered before permitting changes to be made. The ROBCOMM password is found in the ROBCOMM.CFG file and may be changed using the Config option. If the user enters the wrong password an error message will appear in the ERROR window. When first shipped, the ROBCOMM.CFG file contains the password "NEW".

To exit the configuration screen the user must press the <Esc> key. ROBCOMM asks whether to save the information to the ROBCOMM.CFG file on disk. If the user presses the <Y> key for yes, the disk file will be updated. The user will then be asked whether to update the system with the new values. If the user presses the <Y> key for yes again, then the system is updated to the new values. The config screen values are defined in section 1-7.

```
CRS PLUS ROBOT COMMUNICATION PACKAGE  4.2
MENU =   Terminal   Baud Rate  Slave No.  Drive Spec Config     Quit
     =   Enter new password.
Current System Settings  <Esc> to Quit.

CURRENT SETTINGS
-----------------

PASSWORD NEW
PORTS    1
DELAY    2
ROBOT    1
ACI      2400
TERMINAL 9600
VERSION  410

LOG:

RAPL SYNTAX VERSION 4.10 LOADED     RAPL SYNTAX VERSION 4.10 LOADED
C:\ROBOT\FILES                      Slave 1   — Baud Rate  2400
```

Figure 2-4 - The Configuration menu and display.

Quit - The quit function returns control to the MAIN menu.

Terminal Menu

Entering Terminal from the Setup menu or pressing <Ctrl-T> from any menu will bring up the Terminal menu. If using COM1 or COM2 only, the user must ensure that the Serial Switch Box is set to port A (which must be connected to Device #0) and then press <ESC>. At that point, the Terminal menu will be displayed. It contains three items:

Emulate -	Emulates a terminal in DATA window.
Baud Rate -	Sets the terminal Baud rate.
Quit -	Return to Setup menu.

Note: Check the baud rate display at the bottom right corner of the screen. If device #0 of the robot has been set to the displayed rate, emulation may commence immediately. If the baud rate does not match, emulation will cause a garbled display on the DATA screen.

```
CRS PLUS ROBOT COMMUNICATION PACKAGE  4.2
MENU =  Emulate   Baud Rate  Quit
WILL =  Emulate a Terminal in DATA Window.

LOG:

     RAPL SYNTAX VERSION 4.10 LOADED
C:\ROBOT\FILES                    Slave 1   Baud Rate  9600
```

Figure 2-5 - TERMINAL menu options.

Terminal Menu

Entering Terminal from the Setup menu or pressing <Ctrl-T> from any menu will bring up the Terminal menu. If using COM1 or COM2 only, the user must ensure that the Serial Switch Box is set to port A (which must be connected to Device #0) and then press <ESC>. At that point, the Terminal menu will be displayed. It contains three items:

Emulate -	Emulates a terminal in DATA window.
Baud Rate -	Sets the terminal Baud rate.
Quit -	Return to Setup menu.

Note: Check the baud rate display at the bottom right corner of the screen. If device #0 of the robot has been set to the displayed rate, emulation may commence immediately. If the baud rate does not match, emulation will cause a garbled display on the DATA screen.

```
CRS PLUS ROBOT COMMUNICATION PACKAGE  4.2
MENU =   Emulate    Baud Rate  Quit
WILL =   Emulate a Terminal in DATA Window.

LOG:

     RAPL SYNTAX VERSION 4.10 LOADED
C:\ROBOT\FILES          Slave 1    Baud Rate  9600
```

Figure 2-5 - TERMINAL menu options.

Emulate - The screen will clear except for the top title line and the PC will act as a ´dumb´ terminal. The function keys have predefined RAPL commands assigned to them in this mode to reduce keystrokes. The following is a list of the commands supported:

F1 = EDIT
F2 = RUN
F3 = DIR
F4 = LIST
F5 = ENABLE
F6 = DISABLE
F7 = MANUAL
F8 = TEACH
F9 = STATUS
F10 = NEXT

For details of the operation of these commands, refer to the RAPL User´s Manual.

Pressing <Ctrl-Z> will terminate emulation and return to the Terminal menu.

Baud Rate - The user may select the terminal Baud Rate item and choose the next most appropriate rate. Refer to the discussion in section 1-8 for more detail on this subject.

Quit - Once again the user will be asked to change the switch setting on the Serial Switch Box and press the <Esc> key to verify the action. Control will now be returned to the Setup menu or the menu from which the terminal mode was called.

Baud Rate Menu Options

The BAUD RATE menu consists of the five possible baud rate settings. This permits changing the ACI baud rate to match the rate used in the controller, overriding the default setting for this run of ROBCOMM only. The change will be displayed in the bottom right hand corner of the screen. Upon selecting a baud rate the user is returned to the SETUP menu. As with other menus, pressing the key of the first character of an item in this menu will immediately select that item. Thus pressing <2> will select 2400 baud immediately and return control to the setup menu.

```
CRS PLUS ROBOT COMMUNICATION PACKAGE  4.2
MENU =   4800      2400      1200      600      300
WILL =   Select a Baud Rate from the Menu.

LOG:

     RAPL SYNTAX VERSION 4.10 LOADED
C:\ROBOT\FILES                    Slave 1   — Baud Rate  2400
```

Figure 2-6 - BAUD RATE menu options.

Data Save Menu Options

The second function available from the MAIN menu is the DATA SAVE which gives the user the DATA SAVE menu. This contains several options:

Program -	Save RAPL programs from the robot to disk.
Locations -	Save all locations from the robot to disk.
Variables -	Save all variables from the robot to disk.
Job Save -	Save a combination of the above from robot to disk.
Quit -	Return to Main Menu.

The function of each of these items is explained below. The DATA SAVE sub-menu is shown in figure 2-7.

```
CRS PLUS ROBOT COMMUNICATION PACKAGE  4.2
MENU =  Program   Locations  Variables  Job Save   Quit
     =  Select Program from List below. <Q> to quit.
PROGRAM(S) IN ROBOT MEMORY
DIGIO        ANAIO        OUT          AIO
AUTO_ST      MAIN         LOOP         GO_PALLT
GET_PART     DROPPART

LOG:
RAPL SYNTAX VERSION 4.10 LOADED     RAPL SYNTAX VERSION 4.10 LOADED
DIGIO    SAVED TO DIGIO.ROB         ANAIO    SAVED TO ANAIO.ROB
OUT      SAVED TO OUT.ROB           AUTO_ST  SAVED TO AUTO_ST.ROB
C:\ROBOT\FILES                      Slave 1   — Baud Rate  2400
```

Figure 2-7 - DATA SAVE Menu, shown ready to save a program.

Program - The program function will save a RAPL program from the controller to a file in the current working directory. It first reads the program table in the controller and displays all the robot programs in the DATA window by name. If there are no programs in memory an appropriate message will appear in the ERROR window. The program to save is selected by highlighting the desired name using the cursor keys and pressing <Enter>. If there are more files than will fit in the DATA window, cursoring up or down at the screen limits will display the extra file names.

If the user does not wish to save any of the programs, he may press the <Q> key to abort this function and return to the DATA SAVE menu. If a program is selected, it will be transferred to disk though the ACI link. The MESSAGE window will display the current status and the selected program will scroll through the DATA window before being written to the disk. Successful completion of the save is indicated in the LOG window.

The program is saved to disk using its RAPL name with the extension ".ROB". Thus, if the program in the robot memory is "TEST", the disk file will be called "TEST.ROB". If a file with that name already existed in the directory, ROBCOMM will give the existing file a .BAK extension ("TEST.BAK") and save the new program normally.

If an error occurs during transfer or writing to disk, it will be displayed in the ERROR window and control will return to the Data Save menu.

Location - The location function saves the robot location table to a file in the current working directory. Upon selecting this function, ROBCOMM will prompt the user for a file name in which to save the locations. Pressing the <Enter> key without typing a name will abort the operation and return control to the Data Save menu. If a valid file name is entered, ROBCOMM will save the locations in a file with the .LOC extension added. If a file with that name already exists in the directory ROBCOMM will rename that file with .LBK extension.

During the transfer, an indication of the status will flash in the MESSAGE window. The location names will be displayed briefly in the DATA window and the successful completion will be displayed in the LOG window. If an error occurs during transfer or writing to disk, it will be displayed in the ERROR window.

Variable - The variable function saves the robot variable table to a file in the current working directory. Upon selecting this function, ROBCOMM will prompt the user for a file name in which to save the variables. Pressing the <Enter> key without typing a name will abort the operation and return control to the DATA SAVE menu. If a valid file name is entered, ROBCOMM will save the variables in a file with the .VAR extension added. If a file with that name already exists in the directory ROBCOMM will rename that file with .VBK extension.

During the transfer, an indication of the status will flash in the MESSAGE window. The variable names will be displayed briefly in the DATA window and the successful completion will be displayed in the LOG window. If an error occurs during transfer or writing to disk, it will be displayed in the ERROR window.

Job - In many cases, the contents of the robot memory will all be related to a single job or task. In this case, the user may wish to save the memory, including programs, variables and locations, using a single JOB name. The Job file saved on disk is a text file containing the names of the program, location and variable disk files which, when later loaded into the robot, contain all the information needed to execute the task. For more information on the structure of the JOB file, refer to Appendix B. It is important to realize that a JOB file is only a list of other files which together make up the robot memory contents. These files will be saved on disk during the job save.

To save a job, select the JOB SAVE menu item. ROBCOMM will ask for a job name which will be the name of the job file (ie entering "TASK1<cr>" will create a file called TASK1.JOB in the working directory). If that file already exists the user is prompted whether to overwrite the existing file or to rename the destination file. If neither action is taken the job is aborted and control is returned to the DATA SAVE menu. ROBCOMM will then display the contents of the robot program table in the DATA window to permit the user to select the programs that will be associated with this job. The first item in the DATA window will be "* ALL *" which permits all programs to be included in a single keystroke. Selecting a program (or ALL programs) for inclusion is done by cursoring to the desired item and pressing <Enter>. An asterisk (*) will appear beside each selected program name.

After selecting the programs in the job, press 'Q' to quit. ROBCOMM will then complete the job file, by adding a variable file name and a location file name to the list. The .VAR and .LOC files will have the same name as the .JOB file created. In the example above, they would be called "TASK1.VAR" and "TASK1.LOC" respectively. Unlike the programs, the user has no choice as to which locations and variables will be stored. The robot's location and variable tables will be stored, in separate files, in their entirety.

Each item saved will be logged in the LOG window. Any errors encountered will be reported. If an error is encountered, the JOB SAVE will be terminated and an "ABNORMAL JOB TERMINATION" message will appear in the LOG window.

Quit - The quit function returns control to the MAIN menu.

Load Data Menu Options

The third function available from the MAIN menu is the LOAD DATA which displays the LOAD DATA menu. This contains several options:

Program -	Load RAPL programs from disk to robot.
Locations -	Load all locations from disk to robot.
Variables -	Save all variables from disk to robot.
Job Load -	Save a combination of the above from disk to robot.
Quit -	Return to Main Menu.

The function of each of these items is explained below. The LOAD DATA sub-menu is shown in figure 2-8.

```
CRS PLUS ROBOT COMMUNICATION PACKAGE  4.2
MENU =  Program   Locations  Variables  Job Load   Quit
     =  Select program from list below. <Q> to quit.
.ROB FILES ON DISK:
AIO          ANAIO        BAD          DIGIO
IO           OUT

LOG:

RAPL SYNTAX VERSION 4.10 LOADED
C:\ROBOT\FILES                    Slave 1   Baud Rate  2400
```

Figure 2-8 - LOAD DATA Menu showing a program ready to load.

Program - This function will load a .ROB disk file into the robot memory. Once this item is selected all .ROB files found in the current directory will be displayed alphabetically in the DATA window. The file to be loaded is selected by using the cursor keys. If there are more files than will fit in the DATA window, cursoring up or down at the screen limits will display the extra file names. If the desired file is not present, the <Q> key will abort the sequence and return to the LOAD DATA menu. If there are no .ROB files present, an error message will appear in the ERROR window.

Upon locating the desired file, the <Enter> key selects the file and loads it to the robot. When a file has been selected, the file will be transferred to the robot controller using the ACI, the MESSAGE window will display communication activities, and the selected file will be displayed in the DATA window. If no errors occur, the LOG window will display successful completion.

The program loaded to the robot will have the same name as the disk file. If a program with that name exists in the robot ROBCOMM will prompt the user for a new name. A robot program cannot be deleted or overwritten using ROBCOMM. Pressing <Enter> before entering a new name will abort the load and return control to the LOAD DATA menu. Otherwise the file will be loaded to the robot and given the new name.

Location - The load location function will load a complete location table into the robot memory from a disk file. The DATA window will then display all .LOC files found on disk alphabetically. The file to be loaded is selected by using the cursor keys. If there are more files than will fit in the DATA window, cursoring up or down at the screen limits will display the extra file names. If the desired file is not present, the <Q> key will abort the sequence and return to the LOAD DATA menu. If there are no .LOC files on disk an error message will appear in the ERROR window.

Upon locating the desired file, the <Enter> key selects the file and loads it to the robot. When a file has been selected, the locations in the file will be transferred to the robot controller using the ACI. ROBCOMM performs a location check to determine if locations in the robot have the same name as locations in the file. If any match, the Check Duplicate menu will be displayed. This permits the user to Overwrite, Ignore, or Rename the location in question or Abort the load sequence. If there are no more spaces in the robot's location table, ROBCOMM will report this during this check.

As the file is being transferred, the MESSAGE window will display communication activities. If no errors occur, the LOG window will display successful completion. If an error occurs during transfer or reading from disk, it will be displayed in the ERROR window.

Variable - The load variable function will load the variable table into the robot's memory from a disk file. The DATA window will then display all .VAR files found on disk alphabetically. The file to be loaded is selected by using the cursor keys. If there are more files than will fit in the DATA window, cursoring up or down at the screen limits will display the extra file names. If the desired file is not present, the <Q> key will abort the sequence and return to the LOAD DATA menu. If there are no .VAR files on disk an error message will appear in the ERROR window.

Upon locating the desired file, the <Enter> key selects the file and loads it to the robot. When a file has been selected, the variables in

the file will be transferred to the robot controller using the ACI. ROBCOMM performs a variable check to determine if variables in the robot have the same name as variables in the file. If any names match, the Check Duplicate menu will be displayed. This permits the user to Overwrite, Ignore, or Rename the variable in question or Abort the load sequence. If there are no more spaces in the robot's variable table, ROBCOMM will report this during this check.

As the file is being transferred, the MESSAGE window will display communication activities. If no errors occur, the LOG window will display successful completion. If an error occurs during transfer or reading from disk, it will be displayed in the ERROR window.

Job Load - In many cases, the contents of the robot memory will all be related to a single job or task. In this case, the user may wish to load the memory, including programs, variables and locations, using a single JOB name. The Job file on disk is a text file containing the names of the program, location and variable disk files which, when loaded into the robot, contain all the information needed to execute the task. For more information on the structure of the JOB file, refer to Appendix B. It is important to realize that a JOB file is only a list of other files which together make up the robot memory contents. These files must exist on the disk in order to load the job.

To load a job, select the JOB Load menu item. Once this item is selected all *.JOB files found in the current directory will be displayed alphabetically in the DATA window. The file to be loaded is selected by using the cursor keys. If there are more files than will fit in the DATA window, cursoring up or down at the screen limits will display the extra file names. If the desired file is not present, the <Q> key will abort the sequence and return to the LOAD DATA menu. If there are no *.JOB files present, an error message will appear in the ERROR window.

Upon locating the desired file, the <Enter> key selects that job for loading to the robot. When a file has been selected, ROBCOMM checks the user as to whether he wishes to append the job or not. Appending permits the job to be added to the current contents or the memory. If the user answers no to the "Append? (Y/N)" question, he must then decide to proceed or not. Continuing will clear the current robot memory before loading the job from the disk. If the job is appended, the user may be called upon from time to time during the load to check for duplicate location or variable names. If not, the load will proceed without intervention.

Each item loaded will be logged in the LOG window. Any errors encountered will be reported. If any errors are encountered, the JOB Load will be terminated and an "ABNORMAL JOB TERMINATION" message will appear in the LOG window. The user should be sure that there is enough RAPL user memory allocated before he attempts to load a job.

Quit - The quit function returns control to the MAIN menu.

Check Duplicate Menu Options

The CHECK DUPLICATE menu appears when an duplication error occurs while loading either locations or variables to the robot controller. The problem comes about when the name of an item in the file matches the name of a similar item already in the robot memory. If this happens, the user will be called upon to select an option from this menu.

In any case where the item table is full and cannot accept any new items, a message will be displayed in the DATA window. If this occurs, the controller memory must be re-allocated to leave more space for the items missed (refer to RAPL manual) before attempting to re-load the information.

Overwrite -	Uses the value of the item as found in the file. Overwrite the value in the robot memory.
Rename -	Permits the file item to be loaded under a different name.
Ignore -	Ignores the value on disk. Leaves the memory item intact.
Abort -	Aborts the entire load cycle.

The function of each of these items is explained below. The CHECK DUPLICATE menu is shown in figure 2-9.

```
CRS PLUS ROBOT COMMUNICATION PACKAGE  4.2
MENU =  Overwrite  Rename   Ignore   Abort
WILL =  Rename Location from Disk.

Location :  GGG     Already Exists in Robot
Location :  GGG           IGNORED
Location :  C       Already Exists in Robot
Location :  C           OVERWRITTEN
Location :  A       Already Exists in Robot
Location :  A           RENAMED TO          STOP
Location :  STOP     IGNORED. TABLE FULL
Location :  ABC     Already Exists in Robot
Location :  ABC         OVERWRITTEN
Location :  JJJ     Already Exists in Robot
Location :  JJJ         OVERWRITTEN
Location :  TTT     Already Exists in Robot
Location :  TTT           IGNORED
Location :  EEE     Already Exists in Robot
Location :  EEE

LOG: DIGIO    SAVED TO DIGIO.ROB        ANAIO    SAVED TO ANAIO.ROB
     OUT      SAVED TO OUT.ROB          AUTO_ST  SAVED TO AUTO_ST.ROB
     GET_PART SAVED TO GET_PART.ROB     NO LOCATION(S) TO SAVE
     LOADED LOCATIONS(S)
C:\ROBOT\FILES                          Slave 1   — Baud Rate  2400
```

Figure 2-9 - Check Duplicate Menu and Display

Overwrite - This function loads the item (location or variable) from the disk into the memory space occupied by the old item. The actual transfer will not take place until the check is complete.

Rename - This function renames the item from the file. Then it checks to see if the new name exists in the controller or the list from the disk. If it does, the CHECK DUPLICATE menu is displayed again. If the new name is accepted, the transfer will not take place until the location check is complete. Notice the message in figure 2-9 where a location is renamed to STOP but no room remains in the table. If this should happen, the user must be careful that he is not left without some critical information in memory. If this occurs, the RAPL "ALLOC" command can be used to change the memory partition to permit more of the desired item to be loaded.

Ignore - This function, unlike the overwrite, skips that entry coming from disk, leaving the matching item in the controller intact.

Abort - This function aborts the loading of any locations or variables to the robot controller. It also leaves the disk file unchanged. The abort function returns control to the LOAD DATA menu.

Utilities Menu Options

The ROBCOMM package has a number of utilities which operate on robot programs saved on disk. These functions are available through the Utilities menu. This menu contains several options:

Check -	Checks the syntax of a RAPL program.
Tokenize -	Tokenize a RAPL program.
Untokenize -	Untokenize a RAPL program.
Renumber -	Renumber a RAPL program.
Memory -	Changes to MEMORY menu.
Quit -	Return to Main Menu.

The function of each of these items is explained below. The UTILITIES menu is shown in figure 2-10.

```
CRS PLUS ROBOT COMMUNICATION PACKAGE  4.2
MENU =  Check     Tokenize   Untokenize Renumber   Memory     Quit
     =  Select program from list below. <Q> to quit.
                    .ROB FILES ON DISK:
   ANAIO            AUTO_ST          DIGIO            GET_PART
   OUT

LOG: DIGIO    SAVED TO DIGIO.ROB        ANAIO    SAVED TO ANAIO.ROB
     OUT      SAVED TO OUT.ROB          AUTO_ST  SAVED TO AUTO_ST.ROB
     GET_PART SAVED TO GET_PART.ROB     NO LOCATION(S) TO SAVE
     LOADED LOCATIONS(S)
C:\ROBOT\FILES                          Slave 1   — Baud Rate  2400
```

Figure 2-10 - UTILITIES Menu showing the Renumber option

Check - This function will check the syntax of a robot file on the disk.

Once this item is selected all .ROB files found in the current directory will be displayed alphabetically in the DATA window. The file to be checked is selected by using the cursor keys. If there are more files than will fit in the DATA window, cursoring up or down at the screen limits will display the extra file names. If the desired file is not present, the operation may be aborted using the <Q> key. If there are no .ROB files present, an error message will appear in the ERROR window.

Upon locating the desired file, the <Enter> key selects it for checking. When a file has been selected, the syntax of the program stored in that file will be checked using the syntax information according to the current version of RAPL being used. ROBCOMM also creates a *.LST file, which contains the source file, any the error messages generated, and the information summary table. An indication of what is happening will flash in the MESSAGE window.

As the program is checked, it will scroll through the DATA window. Errors in a line will cause the scroll to stop and wait for a key to be pressed. Pressing <Q> will terminate the check while any other key will continue it. Should any errors occur, the user should verify the correct syntax of the command in the RAPL manual. If a disk read or write error occurs during the check (ie. disk write-protect etc.) the error number will be displayed in the ERROR window.

The syntax check is useful for testing programs written off-line before transferring to the robot memory. Note that no check of program logic or flow is made. This is left to the robot controller once the program is loaded. RAPL features such as TRACE mode, and the run time error display are useful for determining logic errors.

Tokenize - This function will replace each English-like RAPL command in the desired program with a numeric equivalent (a token) which will increase the execution speed of the program once it has been re-loaded into the controller [see RAPL manual].

Once this item is selected all .ROB files found in the current directory will be displayed alphabetically in the DATA window. The file to be tokenized is selected by using the cursor keys. If there are more files than will fit in the DATA window, cursoring up or down at the screen limits will display the extra file names. If the desired file is not present, the operation may be aborted using the <Q> key. If there are no .ROB files present, an error message will appear in the ERROR window.

Upon locating the desired file, the <Enter> key selects it for tokenizing. When a file has been selected, the file will be tokenized according to the command list of the currently loaded RAPL version (see section 2-1, CONFIG option).

When a file has been selected, ROBCOMM will prompt the user, in the INSTRUCTION window, for the name of a destination file. He may enter a

name consisting of no more than 8 characters. The name of the source file selected will be displayed as the default. If the user presses the <Enter> key, the destination file will have the same name as the source but the original source .ROB file will renamed as a .BAK file. If a different name is entered, and a .ROB file by that name is already on disk, it is also renamed to a .BAK file. If no file name is entered the operation is aborted. Once a valid destination file has been entered the ROBCOMM system begins tokenizing the source file, and an indication of what is happening will flash in the MESSAGE window. The tokenized file will scroll through the DATA window and successful completion will be displayed in the LOG window. If a disk read or write error occurs it will be displayed in the ERROR window.

If the program contains errors or unrecognizable RAPL commands (ie. not present in current version), those lines will pass unchanged.

Using a different name as the destination file prevents the loss of the original program, but remember, if another program calls this one as a subroutine, the name must be changed in the calling GOSUB statement as well.

Untokenize - This function will replace the numeric equivalent of the RAPL commands (the tokens) with the original English-like RAPL commands. This will decrease program execution speed but improve program readability (see RAPL manual).

Once this item is selected all .ROB files found in the current directory will be displayed alphabetically in the DATA window. The file to be untokenized is selected by using the cursor keys. If there are more files than will fit in the DATA window, cursoring up or down at the screen limits will display the extra file names. If the desired file is not present, the operation may be aborted using the <Q> key. If there are no .ROB files present, an error message will appear in the ERROR window.

Upon locating the desired file, the <Enter> key selects it for untokenizing. When a file has been selected, the file will be untokenized according to the command list of the currently loaded RAPL version.

When a file has been selected, ROBCOMM will prompt the user, in the INSTRUCTION window, for the name of a destination file. He may enter a name consisting of no more than 8 characters. The name of the source file selected will be displayed as the default. If the <Enter> key is pressed, the destination file will have the same name as the source but the original source .ROB file will renamed as a .BAK file. If a different name is entered, and a .ROB file by that name is already on disk, it is also renamed to a .BAK file. If no file name is entered, the operation is aborted. Once a valid destination file has been entered the ROBCOMM system begins untokenizing the source file, and an indication of what is happening will flash in the MESSAGE window. The untokenized

file will scroll through the DATA window and successful completion will be displayed in the LOG window. If a disk read or write error occurs it will be displayed in the ERROR window.

If the program contains errors or unrecognizable RAPL commands (ie. not present in current version), those lines will pass unchanged.

Using a different name as the destination file prevents the loss of the original program, but remember, if another program calls this one as a subroutine, the name must be changed in the calling GOSUB statement as well.

Renumber - This function will renumber a .ROB file from the current directory. Renumbering may be needed if editing has left no spaces between lines for further changes. It can also be used to improve program appearance for documentation. For example, the program below:

```
5 ; A routine to get the robot clear of the machine
10 DEPART .5
20 MOVE CLEAR1
22 ; Wait for the signal
25 IFSIG -4 THEN 20
30 MOVE SAFE1
32 MOVE SAFE2
34 MOVE SAFE3
40 RETURN
$
```

renumbered with starting number 1000 and increment of 100 will look like this:

```
1000 ; A routine to get the robot clear of the machine
1100 DEPART .5
1200 MOVE CLEAR1
1300 ; Wait for the signal
1400 IFSIG -4 THEN 1200
1500 MOVE SAFE1
1600 MOVE SAFE2
1700 MOVE SAFE3
1800 RETURN
$
```

This format is neater and allows room for insertion of new lines should further editing be necessary.

Once this item is selected all .ROB files found in the current directory will be displayed alphabetically in the DATA window. The file to be renumbered is selected by using the cursor keys. If there are more files than will fit in the DATA window, cursoring up or down at the screen limits will display the extra file names. If the desired file is not present, the operation may be aborted using the <Q> key. If there are no .ROB files present, an error message will appear in the ERROR window.

Upon locating the desired file, the <Enter> key selects it for renumbering. When a file has been selected, ROBCOMM will prompt the user, in the INSTRUCTION window, for the name of a destination file; the name of the source file selected will be displayed as the default. The user may enter a name consisting of no more than 8 characters, or press <Enter> to use the default name. If the default is used, the destination file will have the same name as the source, but the original source .ROB file will renamed as a .BAK file. If a different name is entered, and a .ROB file by that name is already on disk, it is also renamed to a .BAK file. If no file name is entered the operation is aborted.

When a valid destination file has been entered ROBCOMM prompts for a starting line number. This will be the number of the first line of the program in the file. The number entered must be an integer value between 1 and 32767. When that number has been entered, ROBCOMM asks for a skip number. This is the increment for each successive number. This must also be an integer between 1 and 32767.

Once both numbers have been entered, ROBCOMM will create a file containing the newly renumbered program. All RAPL GOTO, IF, IFSIG, and IFSTART commands will have the correct destination line numbers as a result. As the program is being renumbered, an indication of what is happening will flash in the MESSAGE window and the new program will be displayed in the DATA window. Successful completion will be displayed in the LOG window. If a disk read or write error occurs the appropriate message will be displayed in the ERROR window.

Memory Menu Options

The robot Memory option is a very powerful debugging tool. It permits access to the robot memory on a byte level. Several memory functions are available from the MEMORY Menu.

Used in conjunction with Appendix F of the RAPL Technical manual, this Utility can be of enormous benefit for debugging robot system irregularities.

WARNING: It is very easy to create unpredictable robot behaviour in the event that the contents of critical robot memory locations are changed. Consequently, it is advised that only persons with a thorough knowledge of the CRS robot and the M1A series of controllers use these commands. For information on the memory structure of the SRS-M1A, refer to APPENDIX F of the SRS-M1A Technical manual.

The memory function has several options:

Display -	Display robot memory in hex form.
Segment -	Select robot memory segment.
Edit -	Edit robot memory.
View -	View robot pointers.
Quit -	Return to Utilities Menu.

Display - This function displays the contents of the controller memory byte by byte in the DATA window.

ROBCOMM prompts the user for a starting memory address offset, and the number of bytes to display. The DATA window will then be filled with the bytes starting at that address. The DATA window can contain 242 bytes maximum. If the user selects more than this number, then the memory will be displayed one screen at a time, displaying the first 242 bytes then prompting the user to press any key to continue or the <Q> key to quit.

The Display function allows the user to observe the memory, but no provision is made to change the memory contents. Sixteen bytes of data are displayed across each row. Each row starts with the memory address of the first data byte in that row.

Segment - In order to allow speedy entry of memory addresses for the other memory selections, a segment register value that will be used for the other memory functions can be entered using this function. The default value is 40H. For more information on 8086 addressing schemes, refer to the SRS-M1A Technical Manual.

Edit - The edit mode displays and permits editing of one page of memory at a time. When this item is selected, ROBCOMM prompts the user for a starting offset using the current segment (see above).

Memory is displayed in two formats: Hexadecimal in the main portion of the DATA screen and ASCII character format in the right portion (see figure 2-11). When the memory contents are displayed, the cursor is positioned at the first byte in the Hexadecimal part of the page with the byte under the cursor highlighted. The cursor keys can be used to position the cursor in the screen. Pressing any valid hexadecimal key will change the value of the byte at the cursor (also changing the ASCII equivalent). Any byte that has been changed will remain highlighted as long as this page is displayed.

The <page up> and <page down> keys will display the pages of memory before and after the current page of memory while pressing the <Q> key will exit the edit page. However, if any byte of the current page of memory has been changed, (even if the new value is the same as the old), ROBCOMM asks if the user wishes to save the contents of that page to the controller before exiting or displaying the next page.

```
                    CRS PLUS ROBOT COMMUNICATION PACKAGE  4.2
   MENU =   Display    Segment    Edit       View       Quit
        =   Press <Q> to quit.
                                ROBOT MEMORY
0040:2158  31 30 20 2F 30 37 30 20 43 3D 30 0D 32 30 20 2F  10 /070 C=0 20 /
0040:2168  30 37 30 20 43 2B 0D 33 30 20 2F 30 37 33 20 43  070 C+ 30 /073 C
0040:2178  20 54 48 45 4E 20 36 33 30 0D 34 30 20 2F 30 38   THEN 630 40 /08
0040:2188  39 20 2D 43 0D 35 30 20 2F 30 37 32 20 43 20 4C  9 -C 50 /072 C L
0040:2198  54 20 34 30 20 54 48 45 4E 20 32 30 0D 36 30 20  T 40 THEN 20 60
0040:21A8  2F 30 37 35 20 31 30 0D 36 33 30 20 2F 30 38 39  /075 10 630 /089
0040:21B8  20 43 0D 36 34 30 20 2F 30 37 35 20 35 30 0D 24   C 640 /075 50 $
0040:21C8  31 30 20 2F 30 37 30 20 43 3D 30 0D 31 35 20 2F  10 /070 C=0 15 /
0040:21D8  30 37 30 20 4E 55 4D 20 3D 31 0D 32 30 20 2F 30  070 NUM =1 20 /0
0040:21E8  37 30 20 43 2B 0D 33 30 20 2F 30 38 33 20 43 2C  70 C+ 30 /083 C,
0040:21F8  41 0D 34 30 20 2F 30 37 32 20 4E 55 4D 20 45 51  A 40 /072 NUM EQ
0040:2208  20 31 20 54 48 45 4E 20 31 30 30 0D 35 30 20 2F   1 THEN 100 50 /
0040:2218  30 37 30 20 4E 55 4D 3D 31 0D 36 30 20 2F 30 37  070 NUM=1 60 /07
0040:2228  34 20 4F 55 54 0D 37 30 20 2F 30 37 32 20 43 20  4 OUT 70 /072 C
0040:2238  4C 54 20 32 35 20 54 48 45 4E 20 32 30 0D 37 32  LT 25 THEN 20 72

 LOG: DIGIO     SAVED TO DIGIO.ROB          ANAIO     SAVED TO ANAIO.ROB
      OUT       SAVED TO OUT.ROB            AUTO_ST   SAVED TO AUTO_ST.ROB
      GET_PART  SAVED TO GET_PART.ROB       NO LOCATION(S) TO SAVE
      LOADED LOCATIONS(S)                                                  72
 C:\ROBOT\FILES                          Slave 1   — Baud Rate  2400
```

Figure 2-11 - Typical MEMORY EDIT display

View - Appendix F of the Technical manual contains a list of all relevant robot memory pointers. The View function permits the value of these pointers to be displayed in the DATA window.

It is possible to view the desired information by pressing the <Enter> key, when the desired pointer is highlighted. This will cause the contents to be displayed in the box in the right half of the DATA window (or in the full DATA window if there is too much information for the box). Figure 2-12 shows this display mode.

```
CRS PLUS ROBOT COMMUNICATION PACKAGE  4.2
MENU =   Display    Segment    Edit      View      Quit
     =    Press <Q> to Quit.
                    ROBOT POINTERS
Time Array Pointer              0040:1CA6
Acceleration Array Pointer      0040:1CAA        : 255
Temporary Array Pointer         0040:1CAE        : 255
Ctpath Knots Array Pointer      0040:1CB6   ALARM STATUS FLAG: 0
Ctpath Time Array Pointer       0040:1CBA   ALARM NUMBER: 17
Ctpath Acceleration Pointer     0040:1CBE   DIGITAL INPUT IMAGE
Ctpath Temp. Array Pointer      0040:1CB2   1111111111111111
Timer                           0040:1CC2   1110111100000000
Current  Segment                0040:1CC6   1111111111111111
Ctpath Number of Knots          0040:1CCC   1111111100000001
Ctpath Number of Axes           0040:1CCE   00000000
Number of Knots                 0040:1CC8   DEVICE SELECTED: 0
Number of Axes                  0040:1CCA   PROGRAM COMPLETE FLAG: FF
Ctpath Loaded Flag              0040:19B2   LINE NUMBER OF KNOTS: 10
Straight Ln Number of Knots     0040:11E4
LOG: DIGIO    SAVED TO DIGIO.ROB        ANAIO    SAVED TO ANAIO.ROB
     OUT      SAVED TO OUT.ROB          AUTO_ST  SAVED TO AUTO_ST.ROB
     GET_PART SAVED TO GET_PART.ROB     NO LOCATION(S) TO SAVE
     LOADED LOCATIONS(S)
C:\ROBOT\FILES                          Slave 1   — Baud Rate  2400
```

Figure 2-12 - Typical MEMORY VIEW display

2-3 ENDING THE PROGRAM

Most sub-menus have a ´Quit´ item. After performing the desired functions within the menu, selecting ´Quit´ will return control to the main menu. When the user is completely finished, the ´Quit´ item from the main menu will return control to DOS. To prevent accidentally leaving ROBCOMM, the system asks for confirmation before returning to DOS.

After saving files to a disk, the user may wish to back up the newly saved files. The DOS ´COPY´ command may be used to copy working files to a backup disk which can be stored in a safe place.

3-1 ERRORS

Any time an error occurs during operation of the ROBCOMM program, the PC speaker will beep, and an error message will be displayed in the ERROR window. The message is displayed for the time specified in the ROBCOMM.CFG (see section 1.7) before the system returns to the current menu. The default time for this display is two seconds.

Disk Access Errors

All attempts to read or write to the disk in the ROBCOMM system must be done within the rules of the Disk Operating System, ie. DOS. For this reason, the user should be familiar with the working aspects of the MS/DOS system before operating ROBCOMM.

Typical errors that result from improper disk access are listed in Appendix A. In each case, the disk access may be repeated after correcting the fault. If the error is unclear, there is a corresponding error number which can be used to help trace the problem.

Communication Errors

Communication errors result from a fault being detected by the error checking protocol used by the ACI. This type of error results in a communication error message being displayed. The message tells what communication was underway when the error ocurred and the type of error. The latter is in the form of a number. This number can be decoded by looking it up in Appendix A.

When a communication error occurs at the PC end, a corresponding error often occurs at the robot end. This error can be seen by the @@RS command in RAPL. The RAPL error numbers are also listed in Appendix A.

Syntax Errors

This type of error occurs only when using the syntax checker to verify the syntax of the source code in a program. A list of possible errors are listed in Appendix A. Note all syntax errors should be verified against the syntax given in the RAPL manual.

RAPL Errors

During operation of the robot through the Terminal Mode of the ROBCOMM system, the user may encounter RAPL error messages. As an aid to interpreting these messages, refer to the RAPL Programming Manual, Appendix B.

4-1 INTRODUCTION

Two other important utilities are supplied with the ROBCOMM package. These are the READCAL and LOADCAL programs. These are to be used to save the robot calibration values on disk and to restore them in case of a robot memory failure or when changing the arm used by a specific controller. Both of these utilities use the COM1 port exclusively.

The calibration values are the numbers which permit homing of the robot at start-up. These are five numbers which represent the offset from the absolute ZERO position to the HOME position (at the pointed end of each home mark). In order to home itself the robot must move to the HOME position and then load the calibration values into the current position registers. The HOME command performs this function.

If a major memory error occurs, the calibration values may be lost or altered. The values are protected by a checksum: that is, the total of all the numbers in there byte representation. Each time a HOME is attempted, the numbers are totalled and compared to the stored checksum. Any change in any number would cause a 'CAL CHECKSUM' error message to be displayed and prevent the robot from being homed.

The READCAL and LOADCAL functions permit saving of the values for all calibration offsets and the checksum on disk. In case of error, they can then be reloaded from the disk file.

4-2 READCAL

The use of this program is interactive. To use it answer all questions as they come up on the screen. Figure 4.1 shows the sequence of questions used to operate the READCAL package.

```
******************************** WARNING ********************************

This program is intended to save the calibration numbers for the  CRS Plus
robot system.  As such it  overwrites any previously stored file for  this
robot serial number. It saves the current  values without regard for their
correctness.  Is this what  you want? If so respond with a ´Y´ for yes. If
not, any letter will do: Yes

Enter desired baud rate: 2400
Enter slave number of the robot: 1
Enter the disk and optionally, directory
on which to store the file: C:\ROBOT\CALFILES

Enter Serial Number of your robot system: 1082

Calibration successfully stored to C:\ROBOT\CALFILES\SRS_1082.CAL
```

Figure 4-1 - Display during operation of READCAL program. User input **BOLD**

Note the WARNING. Because of the nature of this program, it is destructive to the existing values on disk. Entering the correct serial number will create a file called ´SRS_####.CAL´ where ´####´ is the serial number of the unit, in the specified directory. This file should be copied to a backup disk and kept in a safe place.

If the robot has been recalibrated for some reason, repeat the procedure. If, for instance, the robot is to be used with a homing fixture, the values created in the calibration at the fixture can be stored in another file called for instance ´SRS_FIXT.CAL´. To create this file, enter ´FIXT´ instead of the robot serial number.

4-3 LOADCAL

The use of this program is interactive. To use it answer all questions as they come up on the screen. Figure 4.2 shows the sequence of questions used to operate the LOADCAL package.

```
***************************** WARNING ******************************

This program is intended to restore calibration numbers lost after major
memory failure of the CRS Plus robot system. As such it overwrites
current values without regard for their correctness. Is this what you
want? If so respond with a 'Y' for yes. If not, any letter will do: Yes

Enter desired baud rate: 2400
Enter slave number of the robot: 1
Enter the disk and optionally, directory
on which to find the file: C:\ROBOT\CALFILES

Enter Serial Number of your robot system: 1082

Contents of C:\ROBOT\CALFILES\SRS_1082.CAL loaded successfully
```

Figure 4-2 - Display during operation of LOADCAL program. User input **BOLD**

Note the WARNING. Because of the nature of this program, it is destructive to the existing values in the robot memory. Use it to restore the values in the robot after a memory failure. Entering the correct serial number will search for and load the values in a file called 'SRS_####.CAL' where '####' is the serial number of the unit, in the specified directory.

A-1 COMMUNICATION ERROR CODES

There are many ways in which to cause an error in the execution of this package. The most common of these are communication errors. These errors are listed along with their codes below. The format of the system in case of a communication error is to print a message in the ERROR window such as:

'ERROR in reading Location Table Size - ROBCOMM Error # <code>'

This will be displayed for about 2 seconds and will be accompanied by a beep from the computer speaker. The message tells the user what communication was being attempted and what type of communication error ocurred.

The communication errors that can occur during execution of the ROBCOMM package are listed here:

IBM PC COMMUNICATION ERROR CODES

04 An enquiry sequence was ignored 32 times
12 A header was tried 4 times with no success
16 A header was responded to with neither an ACK nor a NAK.
20 A data block read was attempted from the slave 4 times, with an LRC failure all times.
22 A data block write was attempted to the slave 4 times, with a NAK returning all times.
24 Invalid response to a data block write to the slave; neither an ACK nor a NAK returned.
27 Invalid EOT response from the slave.
28 Invalid STX read in a data block read from the slave.
32 Invalid ETB read in a data block read from the slave.
34 Invalid ETX read in a data block read from the slave.
40 Timeout on receiving a character.
50 Timeout on transmitting a character
61 Bad baud rate selection

RAPL COMMUNICATION ERROR CODES

1 ENQ sequence retry error - not used
2 No SOH to lead off the header
3 Bad buffer index in header sequence - serious software failure!
4 Retry failure - Header
5 No ETX or ETB read to end data read sequence
6 LRC check failure in data transfer
7 LRC failure in header
8 Data block transfer retry failure
9 Time-out on receiving a character
10 Bad character read instead of STX in data block read

11 Retry failure - data block write
12 Bad character (other than ACK or NAK) read in data write acknowledge
13 Timeout on transmission of character - will turn off ACI
14 Bad target ID match in header
15 Bad target ID match in Enquiry
16 Bad character was received, other than the final EOT
17 Bad read of ETX character at end of header block
18 Time-out on receipt of header SOH character
19 Time-out on receipt of header ETX character
20 Time-out on receipt of data block STX character
21 Time-out on receipt of data block ETB/ETX character
22 Time-out on receipt of data block ACK/NAK character
23 Time-out on receipt of final EOT
24 Master sent premature EOT
25 Software error in ENQ sequence
26 Bad special read code
27 Bad special write code
28 Bad special code of some kind

A-2 SYNTAX CHECKER ERRORS

The syntax checker will generate error messages which will be displayed in the DATA window as the check is happening. These messages also appear in the *.LST file created by the check function. There is no error number accompanying the message. If a message appears and the user cannot understand the meaning, he should refer to the RAPL Reference manual for a detailed description of the appropriate command.

POSSIBLE SYNTAX CHECKER ERROR CODES

```
***** WARNING SYNTAX CHECKER ABORTED BY OPERATOR REQUEST!                *****
***** ERROR LINE TRUNCATED. MAXIMUM OF 128 CHARACTERS PER LINE.          *****
***** ERROR LINE NUMBER FOUND TO BE OUT OF ORDER.                        *****
***** ERROR LINE NUMBER FOUND TO BE TOO LARGE.                           *****
***** ERROR LINE NUMBER CANNOT BE ZERO.                                  *****
***** ERROR INVALID LINE. NO LINE NUMBER FOUND.                          *****
***** ERROR UNRECOGNIZABLE COMMAND FOUND.                                *****
***** ERROR NO COMMAND SPECIFIED. ONE MUST BE ENTERED.                   *****
***** ERROR INVALID LINE SPECIFIED FOR LINE TRANSFER.                    *****
***** WARNING COMMAND NOT IN COMMAND LIST.                               *****
***** ERROR COMMAND NOT CLASSIFIED FOR PROGRAM MODE.                     *****
***** ERROR COMMAND RESERVED FOR SYSTEM USE ONLY.                        *****
***** ERROR COMMAND SHOULD HAVE NO ARGUMENTS.                            *****
***** ERROR ARGUMENT xx MUST BE RESERVED WORD 'THEN' ONLY.               *****
***** ERROR ARGUMENT xx ARGUMENT MUST BE -40 - 40 BUT NOT ZERO.          *****
***** ERROR ARGUMENT xx MUST BE RESERVED WORDS 'ON' OR 'OFF'.            *****
***** ERROR ARGUMENT xx OPTION MUST BE AN 'S' ONLY.                      *****
***** ERROR ARGUMENT xx MISSING AN '=' SIGN. MUST ENTER ONE.             *****
***** ERROR ARGUMENT xx MUST BE AN '=' SIGN ONLY.                        *****
***** ERROR ARGUMENT xx TOO MANY ARGUMENTS MAXIMUM OF nn                 *****
***** ERROR ARGUMENT xx MISPLACED RESERVED WORD 'THEN'.                  *****
***** ERROR ARGUMENT xx OPTION MUST BE AN 'I' ONLY.                      *****
***** ERROR ARGUMENT xx MISSING A RESERVED SYMBOL 'R' OR 'D'.            *****
***** ERROR ARGUMENT xx MUST BE A RESERVED SYMBOL 'R' OR 'D'.            *****
***** ERROR ARGUMENT xx MISSING ARGUMENT RANGE -40 - 40.                 *****
***** ERROR ARGUMENT xx MUST BE A '+' SIGN.                              *****
***** ERROR ARGUMENT xx SHOULD BE  A '+' OR '-' SIGN.                    *****
***** ERROR ARGUMENT xx MISSING  A '+' OR '-'  SIGN.                     *****
***** ERROR ARGUMENT xx MISSING ARITHMETIC OPERATOR.                     *****
***** ERROR ARGUMENT xx MUST BE ARGUMENT RANGE nn - nn.                  *****
***** ERROR ARGUMENT xx INVALID LOGICAL OPERATOR FOUND.                  *****
***** ERROR ARGUMENT xx MISSING LOGICAL OPERATOR. MUST ENTER ONE         *****
***** ERROR ARGUMENT xx MUST BE RESERVED WORD  'BY'.                     *****
***** ERROR ARGUMENT xx MIXED LOCATION TYPES FOUND.                      *****
***** ERROR          PROGRAM NOT TERMINATED BY DOLLAR SIGN.              *****
***** ERROR ARGUMENT xx PARAMETER RANGE MUST BE %0 - %7 ONLY.            *****
***** ERROR ARGUMENT xx MISSING PROGRAM NAME.   MUST ENTER ONE.          *****
***** ERROR ARGUMENT xx MISSING VARIABLE NAME.  MUST ENTER ONE.          *****
***** ERROR ARGUMENT xx '-' IS INVALID IN PROGRAM NAME.                  *****
***** ERROR ARGUMENT xx INVALID USE OF AN UNARY OPERATOR. '-'.           *****
```

```
***** ERROR ARGUMENT xx INVALID PROGRAM NAME.  UNREADABLE FORMAT. *****
***** ERROR ARGUMENT xx INVALID VARIABLE NAME. UNREADABLE FORMAT. *****
***** ERROR ARGUMENT xx PROGRAM NAME MUST BE UPPER CASE ONLY.     *****
***** ERROR ARGUMENT xx VARIABLE NAME MUST BE UPPER CASE ONLY.    *****
***** ERROR ARGUMENT xx MISSING LOCATION NAME.  MUST ENTER ONE.   *****
***** ERROR ARGUMENT xx '-' IS INVALID IN LOCATION NAME.          *****
***** ERROR ARGUMENT xx INVALID IMPLIED LOCATION.  INDEX 0 - 255. *****
***** ERROR ARGUMENT xx INVALID LOCATION NAME. UNREADABLE FORMAT. *****
***** ERROR ARGUMENT xx LOCATION NAME MUST BE UPPER CASE ONLY.    *****
***** ERROR ARGUMENT xx ARGUMENT MUST BE STRING INDEXES &1 - &4.  *****
***** ERROR ARGUMENT xx STRING IS TOO LONG MAXIMUM 32 CHARACTERS. *****
***** ERROR ARGUMENT xx EXPECTED STRING MISSING LEADING QUOTE.    *****
***** ERROR ARGUMENT xx STRING NOT TERMINATED BY ENDING QUOTE.    *****
***** ERROR ARGUMENT xx FOUND QUOTE(S) EMBEDDED IN STRING.        *****
***** ERROR ARGUMENT xx TOO MANY CHARACTERS AFTER FINAL QUOTE.    *****
***** ERROR ARGUMENT xx INVALID CHARACTER AFTER FINAL QUOTE.      *****
***** ERROR ARGUMENT xx MISSING ARGUMENT RANGE nn - nn.           *****
***** ERROR ARGUMENT xx MISSING AN INTEGER ARGUMENT.              *****
***** ERROR ARGUMENT xx POSITIVE INTEGER ONLY.   NO NEGATIVES.    *****
***** ERROR ARGUMENT xx ILLEGAL INTEGER VALUE.   NO VARIABLES.    *****
***** ERROR ARGUMENT xx POSITIVE CONSTANT ONLY.  NO NEGATIVES.    *****
***** ERROR ARGUMENT xx INVALID REAL VALUE.  FORMAT UNREADABLE.   *****
***** ERROR ARGUMENT xx MISSING REAL VALUE BEFORE EXPONENT.       *****
***** ERROR ARGUMENT xx INVALID EXPONENT.  UNREADABLE FORMAT.     *****
***** ERROR ARGUMENT xx MUST BE RESERVED WORDS 'UP' OR 'DOWN'.    *****
***** ERROR ARGUMENT xx MUST BE RESERVED WORDS 'FORWARD' OR 'BACKWARD'. *****
***** ERROR ARGUMENT xx MUST BE RESERVED WORDS 'JOI' OR 'CYL'.    *****
***** ERROR ARGUMENT xx MUST BE RESERVED WORDS 'E' OR 'O' OR 'N'. *****
***** ERROR ARGUMENT xx MUST BE RESERVED WORDS 'R','X','N' OR 'B' *****
***** ERROR ARGUMENT xx MUST BE RESERVED WORDS 'E' OR 'N'.        *****
***** ERROR ARGUMENT xx INVALID TABLE ITEM SELECTED.              *****
***** WARNING NON-NUMERIC LINE TRANSFER DETECTED.                 *****
```

A-3 SYSTEM ERRORS

If a system error occurs, a message will be displayed in the ERROR window along with an error number. The number corresponds to the fault as indicated in the following list. The action required by the user depends on the severity of the error. Many errors from this list can occur in "normal" use of the package. As an example, when the user wishes to create a new working directory on his disk, he can enter an illegal entry in the SETUP-DRIVE SPEC menu item. The system will respond with error #02. This requires the user to decide whether to create a new directory or try another name. As such, it is not a serious error. However some errors in this list indicate a serious malfunction of the hardware/software of the PC. The column to the right in the following list indicates the type of each error.

SYSTEM ERROR CODES		ERROR TYPE
-1	OPERATING SYSTEM ERROR	FATAL HARDWARE
01	USER IS NOT OWNER	NETWORK - NOT SUPPORTED
02	NO SUCH FILE OR DIRECTORY	NON-FATAL
03	NO SUCH PROCESS	FATAL SOFTWARE
04	INTERRUPTED SYSTEM CALL	FATAL HARDWARE
05	I/O ERROR	DISK OR COMn I/O
06	NO SUCH DEVICE OR ADDRESS	DISK OR COMn I/O
07	ARGUMENT LIST IS TOO LONG	DOS ERROR - CHECK PATH
08	EXEC FORMAT ERROR	FATAL SOFTWARE
09	BAD FILE NUMBER	DOS FILE HANDLING
10	NO CHILD PROCESS	NETWORK - NOT SUPPORTED
11	NO MORE PROCESS ALLOWED	NETWORK - NOT SUPPORTED
12	NO MEMORY AVAILABLE	HARDWARE (RAM IN PC)
13	ACCESS DENIED	NETWORK - NOT SUPPORTED
14	BAD ADDRESS	FATAL HARDWARE
15	BULK DEVICE REQUIRED	I/O
16	RESOURCE IS BUSY	NETWORK - NOT SUPPORTED
17	FILE ALREADY EXISTS	DISK I/O
18	CROSS DEVICE LINK	NETWORK - NOT SUPPORTED
19	NO SUCH DEVICE	DISK I/O
20	IS NOT A DIRECTORY	DISK I/O
21	IS A DIRECTORY	DISK I/O
22	INVALID ARGUMENT	DISK I/O - CHECK PATH
23	NO MORE FILES(SYSTEM)	DISK I/O
24	NO MORE FILES(PROCESS)	DISK I/O
25	NOT A TERMINAL	NETWORK
26	TEXT FILE IS BUSY	NETWORK
27	FILE IS TOO LARGE	DISK I/O
28	NO SPACE LEFT	HARDWARE (RAM IN PC)
29	SEEK ISSUED TO PIPE	NETWORK
30	READ-ONLY FILE SYSTEM	DISK I/O
31	TOO MANY LINKS	NETWORK
32	BROKEN PIPE	NETWORK
33	MATH FUNCTION ARGUMENT ERROR	FATAL HARDWARE
34	MATH FUNCTION RESULT OUT OF RANGE	FATAL HARDWARE

A-4 ROBCOMM ERRORS

If an attempt to perform a function results in a fault or a possibly dangerous situation, an error or warning will appear in the ERROR window. The following lists the possible messages:

OTHER ERRORS	ERROR TYPE/ACTION
100 CONFIGURATION FILE TOO LARGE.	ROBCOMM.CFG has been altered
101 COMMAND FILE TOO LARGE.	A ROBCMD.* file has been altered
102 INVALID COMMAND FILE TOO LARGE. . . .	A ROBIVCMD.* file has been altered
103 FILE MISSING ENDING DOLLAR SIGN	Add $ character to end of file
104 INVALID FILE SYNTAX	Desired file is wrong type
105 LINE NUMBER OUT OF SEQUENCE . . .	Renumber requires sequential line num´s
106 UNABLE TO DETERMINE LINE NUMBER . .	Line number must be followed by space
107 INVALID LINE SPECIFIED FOR TRANSFER	RAPL redirection illegal dest
108 WARNING NON NUMERIC LINE NUMBER TRANSFER. . . .	renumber may be dangerous
109 RENUMBER HAS FAILED	An error restricts renumbering a file
110 ROBOT´S PROGRAM TABLE IS FULL	Re-allocate for larger Prog Table
111 PROGRAM NAME ALREADY EXISTS IN ROBOT.	Can´t overwrite existing RAPL Prog
112 INSUFFICIENT MEMORY TO LOAD FILE. . .	Re-allocate RAPL for more prog mem.
113 ATTEMPTED TO TOKENIZE OR UNTOKENIZE RESERVED AREA . .	Bad command in file
114 UNABLE TO DETERMINE COMMAND	Bad command in file
115 INSUFFICIENT LOCATION MEMORY.	Re-allocate RAPL for more loc´s
116 INSUFFICIENT VARIABLE MEMORY.	Re-allocate RAPL for more var´s
117 TYPE OF FILE EXCEEDS 512.	DOS error - wrong file type
118 EMPTY FILE REQUESTED.	File has 0 byte size
119 TOKENIZING HAS FAILED	An error restricts tokenizing
120 UNTOKENIZING HAS FAILED	An error restrict un-tokenizing
121 USER AUTHORIZATION FAILURE HAS OCCURRED	Wrong password entered
122 LOCATION TABLE FULL	This location will not be loaded
123 VARIABLE TABLE FULL	This variable will not be loaded

The job file is a standard DOS text file with .JOB as an extension. It has a general structure that is read by ROBCOMM to determine what files, when loaded, constitute the complete user memory for a given robot task or job.

The job file can be broken down into three basic sections: the program file section, the variable file section, and the location file section. The appropriate ROB, VAR and LOC file names are entered in the job file without the corresponding file name extension. The file type is determined by its location within the job file. Each section of the job file is separated by a comma, carriage return, and linefeed.

The program file section can consist of several program file names. Only one robot program name can be on each line of the job file. If there is more than one .ROB entry in the ROB section then each line, except the last, must end with a + character. The ´+´ indicates that more programs follow.

The VAR and LOC section may contain only one file name each. The variable and location files may have any name, but when a Job Save is done, these files will take the same name as the job file but with extensions ".VAR" and ".LOC". For example, the following is a typical job file structure:

```
PROG1+
PROG2,
JOB1,
JOB1
```

The previous example will save the robot program file PROG1 to PROG1.ROB. The location information will be saved in JOB1.LOC, and the variable information will be saved to JOB1.VAR. These file types are the same as those created by the ROBCOMM save data routines. The next example of a job file is more involved:

```
AUTO_ST+
MAIN+
LOOP+
ALL_OFF,
VARS,
LOCS
```

This job file states that robot programs AUTO_ST, MAIN, LOOP and ALL_OFF are to be saved to AUTO_ST.ROB, MAIN.ROB, LOOP.ROB and ALL_OFF.ROB. There are four robot programs in all that define in job task. All are joined by a + as required. The last program line ends with a comma as expected, since the next entry is the variable file name.

CRSPLUS
INDUSTRIAL AUTOMATION

CRS PLUS INC.
830 Harrington Court
Burlington, Ontario
L7N 3N4

Tel (416) 639-0086
Fax (416) 639-4248

February 26, 1990

Dr. Ralph Hsu
Algonquin College
Dept. IMMM
200 Lees Avenue
Ottawa, Ontario
K1S 0C5

Dear Dr. Hsu,

Thank you for completing and returning the REQUEST FOR READER'S COMMENTS form. Comments from valued customers such as yourself, provide us with the means to constantly improve our products.

Your observation of the DIGITAL I/O and ANALOG I/O connectors pin numbering is quite correct. Also, if you have a COMBO card you may have noticed that the numbering of the two analog output channels is also reversed. These typographical errors have already been corrected and I have enclosed the corrected pages from the current TECHNICAL manual. Please use these pages as replacements and discard the incorrect pages to avoid future confusion.

Please be very careful if you are working with these signals direct from the connector. These signals are not buffered or isolated from the controller electronics; any mistakes or shorts will damage the mother board - possibly preventing robot operation.

You also asked a number of questions, the answers to which are:

1. Why is the ROBCOMM section of the user manual empty?

   - That section is pre-labelled for the ROBCOMM software package option user manual. If you have purchased this package, the manual should have been added to that section. If you are not familiar with this package, please refer to the description in the enclosed price list.

II. What software diagnostics are available?

- o There is a diagnostic command built into the RAPL operating system. Please refer to the @@DIAG command in Appendix A of the Technical manual.

III. System repair, schematics, and block diagram?

- o All of these items are found in our SERVICE manual. I have attached a copy of the table of contents and some other examples from this manual. I also marked the sales description in the price list.

IV. Full-screen RAPL program editor?

- o This is not practical because the "screen" is actually a separate, intelligent, device outside the robot controller with an RS-232 serial link between them. It would also make use of the SYNTAX BUILDER impossible.

  This is partially why we created the ROBCOMM and ACI package. Robcomm allows the user to send/receive RAPL programs in ASCII text file form. This permits use of any word processor with an ASCII mode, or a simple text file editor, to create and edit RAPL programs. Robcomm is then used to check the syntax of these files, convert to tokens, and send the program to the robot using our ACI protocol to avoid data errors. The ACI package allows the user to write his own robot communication interface. Critical ACI modules and examples are provided in ANSI 'C' so they can be incorporated into your own programs.

V. What repair services are available?

- o You have several options available, including: Updating RAPL firmware, Updating controller hardware, Re-manufacture of the robot arm, Upgrading to another robot model, Factory repair of damaged equipment, Spare part kits, and Service contracts. These options are in the price list, I have already marked the most popular items for you.

I am sending our latest sales brochure and corporate video tape, and have added your name to our mailing list so you will be notified of future developments.

Again, thank you for completing our questionnaire. We look forward to supplying all your light industrial robot needs in the future.

Sincerely,

CRS Plus Inc.

Richard E Cooke

Richard E Cooke
Customer Support group

/rc
algonqui.nl

Encl.

CRSPLUS
INDUSTRIAL AUTOMATION

CRS PLUS INC.
835 Harrington Court
P.O. Box 163, Station A
Burlington, Ontario
L7R 3Y2
(416) 639-0086
FAX: (416) 639-4248

REQUEST FOR READER'S COMMENTS

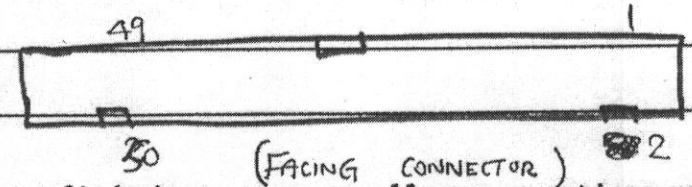
We at CRS Plus make every effort to provide documents that are as elegant as they are practical. The Customer Support Group invites you to take an active role in our documentation process:

NEXT page please

Please specify, by page number, any errors you found in a document.

TECHNICAL SECTION p. 3-8 Fig 3-8 ; p3-10, Fig 3-12

Fig3-8 PIN LAYOUT FOR DIGITAL I/O CONNECTOR IS INCORRECT



ALL ODD NO. ON TOP ROW (1 to 49)

ALL EVEN NO. ON BOT. ROW (2 - 50)

Were you able to find the answers to all your questions using these manuals?

NO, Crucial servicing information is missing - e.g. TEST POINT Voltage, Wave form, DIP switch settings, jumpers configurat.

If not, then what questions do you have?

1) ARE THERE suppose to be MATERIALS UNDER the section ROBCOMM)
I do not have anything under that section in my binder

2) Block diagram of system; electronic schematics for diagnostics & repair; servo axis adjustment procedures; Software diagnostics to check out system

Did you have any difficulty understanding descriptions or wording?

PIN layouts on those connectors were very confusing because of incorrect information.

Use separate sheets as needed. Please attach your name and telephone number so that we can contact you. Mail or FAX this form to the attention of the Customer Support Group.

Thank you for your time, your comments will help keep our manuals and tutorials practical!

DR. RALPH HSU, PROFESSOR
ALGONQUIN COLLEGE, DEPT. IMMM
200 LEES AVE.,
OTTAWA, ONTARIO
CANADA K1S 0C5

TEL: 613-598-3303
FAX: 613-598-3300

Same problem in fig 3-12

Why not keep same orientation of connectors in fig 3-8, 3-12, to reduce confusion!

Also, label "FACING CONNECTOR" could make locating the pins more explicit.