
Retrospective Theses and Dissertations

1984

A 'User-Friendly' Robot Operator Training Aid

Donald A. Washburn
University of Central Florida



Part of the [Engineering Commons](#)

Find similar works at: <https://stars.library.ucf.edu/rtd>

University of Central Florida Libraries <http://library.ucf.edu>

This Masters Thesis (Open Access) is brought to you for free and open access by STARS. It has been accepted for inclusion in Retrospective Theses and Dissertations by an authorized administrator of STARS. For more information, please contact STARS@ucf.edu.

STARS Citation

Washburn, Donald A., "A 'User-Friendly' Robot Operator Training Aid" (1984). *Retrospective Theses and Dissertations*. 4695.

<https://stars.library.ucf.edu/rtd/4695>



A 'USER-FRIENDLY' ROBOT OPERATOR TRAINING AID

BY

DONALD A. WASHBURN
B.S.E., University of Central Florida, 1982

Research Report

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Engineering in the
Graduate Studies Program of the College of Engineering
University of Central Florida
Orlando, Florida

Spring Term
1984

ABSTRACT

There will be an increasing demand for robot operators, since more companies are using robots in production and material handling. In order to train these would-be operators, a 'user-friendly' robot operator training aid was developed. This aid helps the trainee learn how to program a robot using on-line programming, and the trainee also obtains valuable hands-on experience. Since this training aid is user-friendly and has safety features, it requires little or no previous experience with robots or computers and minimal supervision. With this training, the would-be operators will acquire a basic understanding of how they can apply this experience to the programming of large industrial robots.

The RHINO XR-2 robot was used, including some of the peripheral equipment that can be easily interfaced with the RHINO. The system can control eight axes of motion. A controlling program was written in BASIC language for the Radio Shack Model III microcomputer. The computer's keyboard was used to simulate a teach pendant, similar to those used with many industrial robots.

ACKNOWLEDGEMENTS

This project would not have been possible without the support of many people. I would like to especially thank Dr. Gary E. Whitehouse, who has been my mentor throughout my college career and was instrumental in the selection of this graduate research. I also appreciate the help of Dr. John E. Biegel in solving the problems that arose during my research and in proofreading my papers. Furthermore, I am grateful to Dr. Darrell G. Linton for his concern and encouragement during my college career. In addition, I would like to thank all the other professors in the department for their great interest and support.

The RHINO was kept running with the helpful technicians in the lab, especially David Gribben. I would also like to thank the secretaries, especially Lynn and Peggy, for their help and for the use of their computer system.

Finally, my greatest thanks goes to my wonderful family for all they have done for me. Without them, I would not be where I am today.

TABLE OF CONTENTS

INTRODUCTION.....	1
HARDWARE.....	4
SOFTWARE.....	9
USER MANUAL.....	13
SAMPLE EXPERIMENTS.....	22
PROBLEMS ENCOUNTERED.....	24
FUTURE AND IMPROVEMENTS.....	25
CONCLUSIONS.....	26
APPENDICES	
A. Extra Pictures of the RHINO.....	27
B. Sample Program Output.....	30
C. Program Listing of START/BLD.....	33
D. Flowchart for STARTUP/BAS.....	35
E. Program Listing of STARTUP/BAS.....	37
F. Flowchart for TEACH/BAS.....	49
G. Program Listing of TEACH/BAS.....	51
BIBLIOGRAPHY.....	72

INTRODUCTION

Under the Robot Institute of America (RIA) definition, there were 6,300 robots in the United States in 1982. According to a RIA study, 1983 production of robots in the United States was expected to reach 2,400. By 1990 the United States could be making as many as 24,000 robots a year (Keller, 1983). As these robots are put to use, there is and will be a need for robot operators. Therefore, an easy and efficient way to train new operators is needed. Also, training with a small robot should help the trainee overcome any fear of robots.

At present, there are two ways to program a robot, on-line and off-line. A robot usually comes with a teach pendant, a device used to manually control the robot. To teach a given task, the operator uses the teach pendant to move the hand to the required locations. Each location is recorded as the end point of a motion sequence. A task is a set of motion sequences. Once taught the task, the robot can repeat it as many times as necessary. This type of programming is called on-line programming, because the robot must be working on-line (in the location where it will be used). While it is being programmed on-line, the robot can not be used for other purposes. The robot can continue to

work on other tasks while being programmed off-line. Off-line programming has many benefits; however, it is not easily done with today's robots.

On-line programming is the easiest and most used method of programming a robot. Hands-on experience is needed to train the operator to do on-line programming. An inexperienced operator can place people, objects and the robot in danger. Therefore, the operator needs a safe way to get hands-on experience in on-line programming.

After studying this situation, it was determined that a 'user-friendly' robot training aid was needed. The RHINO XR-2 robot was selected because it is a six-axis robot and has several peripheral devices that attach easily. Using the peripheral equipment (a conveyor, a carousel and a linear base), more realistic situations can be developed for the trainee than with a robot alone. Also the RHINO XR-2 interfaces to the controlling computer using only four lines of the RS-232 interface. The use of the RS-232 interface allows the user to choose the controlling computer.

Once developed, this robot operator training aid would be used mostly by University of Central Florida (UCF) engineering students. Dr John E. Biegel is going to use the training aid in the "Robotics in Industry" graduate engineering course. It could also be used to train would-be operators. The engineering students are going to be considered would-be operators or trainees for ease of

explanation. The would-be operator will need only an introduction to the training aid before beginning his/her training. The trainee requires little experience with computers or robots to operate this aid. Also, because this training aid is user-friendly and has safety features, the trainer requires minimal supervision. This training aid will help would-be operators learn about programming a robot so they can be more efficient when using larger industrial robots.

HARDWARE

The hardware of the robot operator training aid does not have to be understood to use the system. But, since some operators may want an understanding of how the hardware works, we have included a brief description.

The six axes of the RHINO XR-2 robot are grasp, wrist-rotational, wrist-azimuthal, forearm, shoulder, and waist. The RHINO XR-2 is shown in figure 1. Six 12 volt servo motors are used to drive the six axes. The linear base, a peripheral device, is used to add a seventh axis of motion to the RHINO system. Using the linear base, the robot can move in either direction along a straight line. The conveyor, another peripheral device, can be used to move objects so that the RHINO can place them on the conveyor as it would in a factory environment. All the peripheral devices used with the RHINO are shown in figures 1 and 2. The carousel is a turntable that can be used to move objects in a circular path. A bin is attached to the carousel by a wiper. A picture of the wiper with a box just about to drop into the bin is shown in figure 3. Objects put onto the table can be deposited into the bin. A table is included in the layout for the RHINO to pick, place, and assemble object(s) upon it.

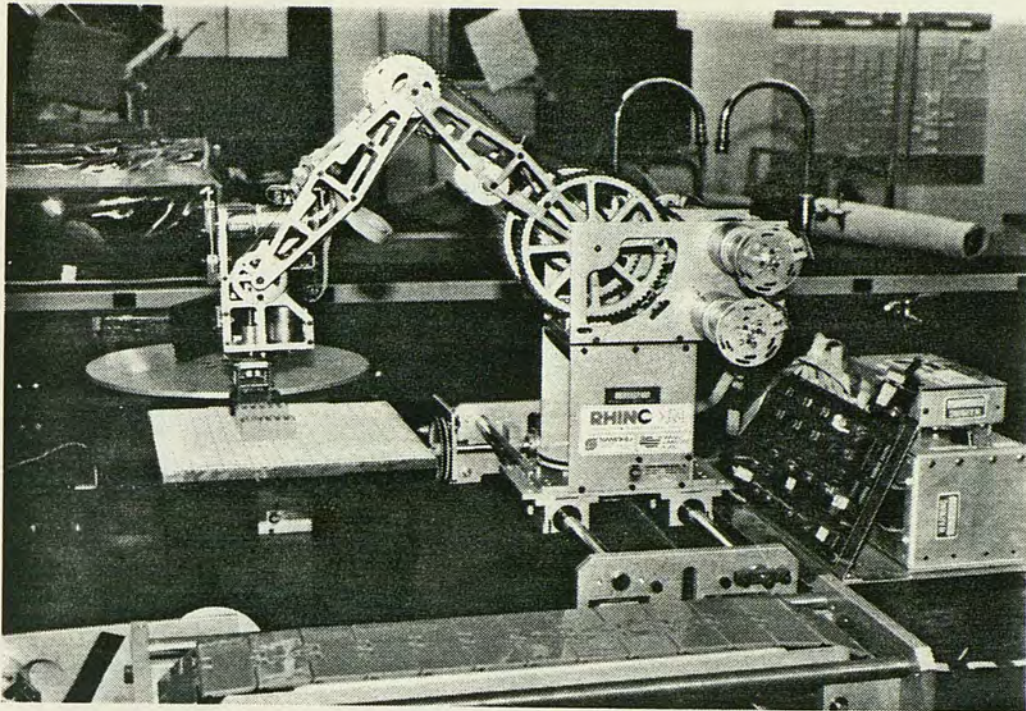


Figure 1. A side view of the RHINO XR-2.

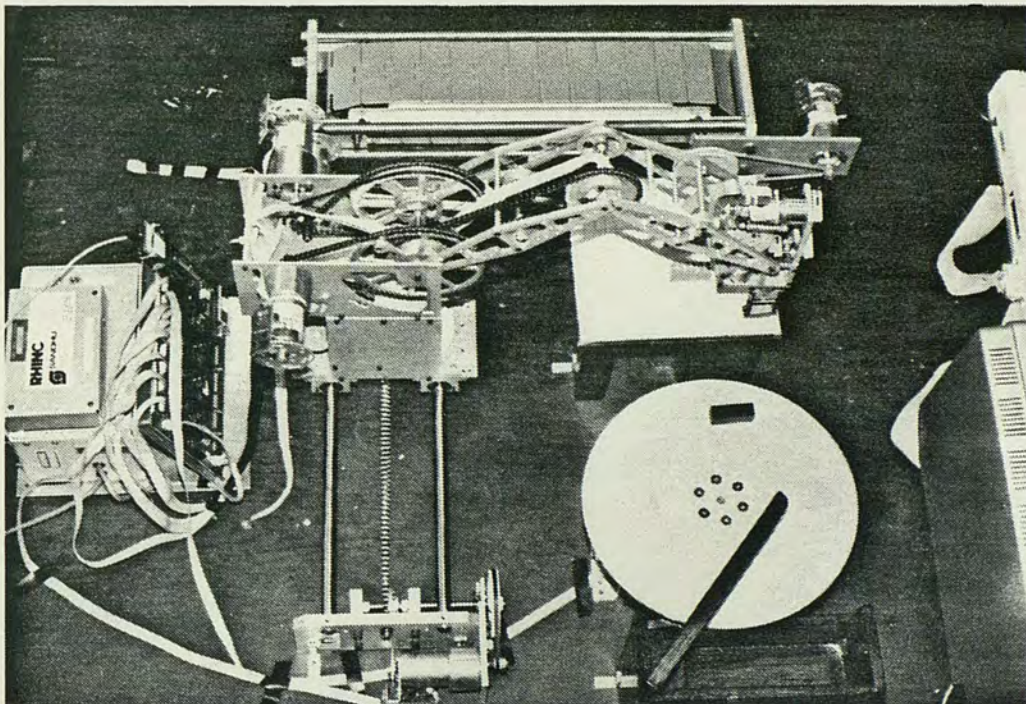


Figure 2. An aerial view of the RHINO XR-2.

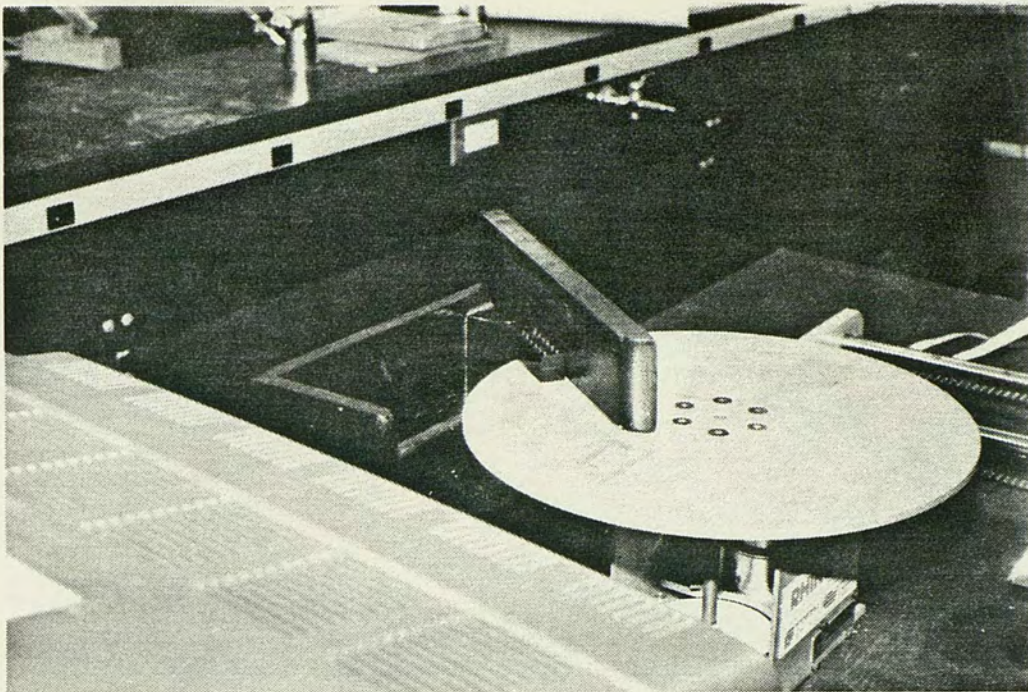


Figure 3. A box, on the carousel, is about to drop into the bin because of the wiper.

To monitor the motion of each axis, an optical encoder is mounted on the drive shaft of the motor. This optical encoder is a round plate, with several pairs of slots which break a beam of light to give an 'on' and 'off' action of the light beam. This 'on' and 'off' action of the light beam created by two slots, which are 90 degrees out of phase, forms a square wave. From the square wave, the number of steps the axis has moved are counted. Using this feature, an axis can be programmed to move a predetermined number of steps.

An Intel 8748 microprocessor is dedicated to controlling the eight axes. The microprocessor, the RS-232 interface, and other components are located on the RHINO

controller board, which is physically attached to the RHINO power supply. The RHINO controller board is shown in figure 4. The microprocessor uses its eight registers to follow the eight axes. Since each register is eight bits, and one bit is needed for the sign, a maximum of 127 steps can be stored by the controller board at one time. Longer motions must be done as a sequence of steps.

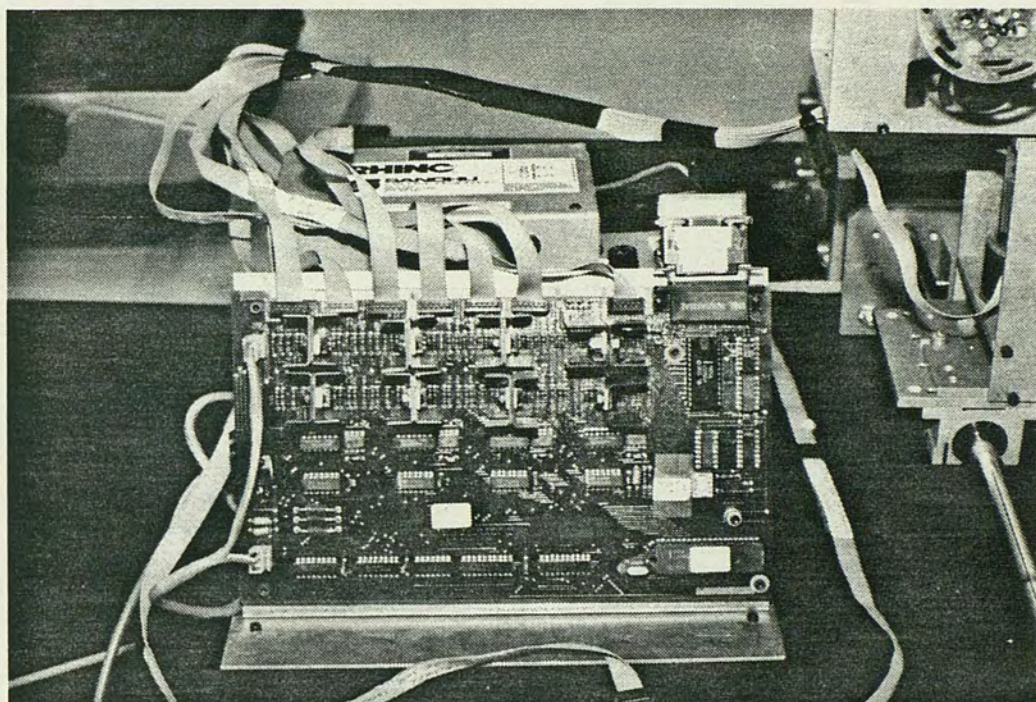


Figure 4. The RHINO controller board and power supply.

Most of the axes have a microswitch along their traverse. In this way, these axes can be reset to their starting locations. Once all axes are reset, the exact location of the system is known. By determining the number of steps each axis drive motor takes before it collides into

itself, the software can prevent the RHINO from being programmed to collide with itself.

The controlling computer communicates with the RHINO controller board by using four lines of the RS-232 interface. The four lines used are transmit data, received data, signal ground, and protective ground. There are four different types of commands used to control the RHINO. The START command is used to move a motor in a specific direction and a specified number of steps. The STOP command is used to stop a motor instantaneously. To obtain the motor status information, the number of steps a motor is to be moved, there is a QUESTION command. Finally, the states of the six microswitches that are used to set the reference position of the axes can be obtained with the STATUS command. All of these commands are transparent to the user of the training aid.

SOFTWARE

A Radio Shack Model III microcomputer was used as the controlling computer for the RHINO XR-2 robot system. The controlling programs were written in the BASIC language and their flowcharts and program listings are included in the appendices. The keyboard of the computer is used to control the RHINO in a manner similar to a teach pendant. The motion of each axis is controlled by four keys; two keys move the axis in a positive direction, and two keys move the axis in a negative direction. Each direction of the axis has two increments of motion, high and low. An increment is the number of steps an axis will move when a key is pressed. The operator may set the number of steps for each increment. The default values are one hundred for 'high' and ten for 'low'. Because of hardware limitations, the increments are limited to a range of one to 127 steps. However, a feature of this software package allows longer moves when only one axis is being moved.

Using the keyboard, the trainee can move the RHINO to different locations and store the point as a step. These steps, when stored, can be run as a routine; a routine being a series of steps. When the robot is being taught the routine the operator moves each axis separately to get to the desired location. When the routine is being run under

computer control, the robot controller moves the robot arm from one point to another by driving one or more axes at the same time. Therefore, the operator must teach the steps or points in the correct sequence to avoid collision during operation, even though there were no collisions during the teaching process. To obtain simpler and smoother motion, it is better to move as few axes as possible in any one step. Pictures of the RHINO while it is running a routine are shown in Appendix A. A routine can have up to one hundred steps. In this way, the operator can create a routine to accomplish a task such as moving a box from a table to a bin. Once created, a routine can be repeated as many times as desired, such as would occur in a factory environment. A routine can also be edited, saved and loaded. A routine can be saved on a disk for demonstration later. The file access mode can be used to get a listing of all routines on the disk. Routines that are no longer needed can be deleted. Since a routine is limited to one hundred steps, routines can be run in a long series called a loop. Therefore, complicated tasks could be broken down into several simple routines. A loop can include up to 20 routines. Each routine can be repeated before going on to the next routine in the loop.

When the system is set up to be used, the axes motors the operator wants to use must be connected to the RHINO controller board. The operator chooses the axes needed to

complete a task. The software package is written in a general form to allow different motor configurations. When initializing the system, the operator inputs the motor configuration that is to be used. Then the RHINO is reset to a starting location by the operator. Once this is done, the operator is ready to enter the RHINO teach mode.

The safety features included in the user-friendly robot operator training aid are useful in helping an inexperienced operator to overcome any fears of damaging any part of the system. The RHINO cannot run into itself when an operator is teaching it a routine. Therefore, it cannot be programmed to run into itself either. If an accidental collision with another object does occur, there is an emergency on/off switch. The emergency switch is shown in figure 5; it is connected to the leg of the table below the microcomputer. The RHINO motor drivers can withstand an overload for short time; therefore if an axis drive motor is stalled and the operator turns the RHINO off quickly, there is little chance of damage to the robot. Also a 'pause' mode can be activated while running a routine. When running with the 'pause' mode on, the RHINO stops after each step so the operator can check to be sure that everything is alright. If there are no problems, the routine can be continued.



Figure 5. The TRS-80 microcomputer, printer and emergency on/off switch.

USER MANUAL

This section is an overview of the controlling software for the robot operator training aid. All the steps that are necessary for the operation of the system and the menus in the software will be explained.

To set up the robot operator training aid, the TRS-80 microcomputer must be turned on by using the switch located under the right side. When the disk drive stops (the red light turns off) the RHINO system disk needs to be inserted in the lower drive with the label side up. For the next step, the operator presses the red reset button on the right side of the TRS-80. Following this, the system asks "Enter Date (mm/dd/yy)?". After an input of the date in this format, 02/15/84, the system asks "Enter time (hh:mm:ss)?". Either the time is entered or the enter button is pressed. After all these steps are taken, the system prints a title, signaling it is ready to proceed.

The system prints the operating procedures and a series of steps that must be completed. This series of screens seen by the operator are shown in Appendix B. One of the steps is connecting the motors into the desired motor configuration. The operator must connect the motor plugs to the desired location on the RHINO controller board. (Note: only motors two, six, seven and eight can be changed or

disconnected). Once this is completed, the operator must enter the motor configuration using the menu shown in figure 6. As can be seen in that figure, the default configuration is displayed on the top of the screen. By entering the locations of the motors that were changed and the motors now connected to these locations, the operator has entered the new motor configuration. The operator enters zero when the correct motor configuration is displayed.

```

                RHINO TEACH MODE
                ROUTINE TITLE EXAMPLE1
) - RUN A LOOP OF ROUTINE          CLEAR - SETS START LOCATION
* - REINITIALIZE THE SYSTEM        PAUSE STATUS - ON
P - CHANGE PAUSE STATUS            1 - 0
@ - RESET THE RHINO (SHIFT @)      2 - 0
{ - ERASE THIS STEP AND BACK ONE (LEFT ARROW) 3 - 0
} - ERASE THIS STEP (RIGHT ARROW)  4 - 0
L - CHANGE INCREMENT VALUES      5 - 0
O - INCREMENT STEP NUMBER (TO RECORD A STEP) 6 - 0
+ - ADD A LONG MOVE TO ONE MOTOR   7 - 0
> - RUN THIS ROUTINE               8 - 0
? - EXIT TEACH MODE                STEP NUMBER - 1
= - DISK FILE ACCESS               HIGH - 100     LOW - 10

```

Figure 6. Menu for the motor configuration.

The final step to be done before beginning use of the teach mode is to reset the RHINO. Motors three through eight have microswitches located along their traverse for resetting. If the microswitch is set, the software checks its location by moving a short distance from the microswitch and then back slowly. However, if the microswitch is not

set, the operator must help the system find the microswitch, since it does not know its location.

Each motor is connected to a gear by a chain or belt, and attached to this gear is an object similar to a round washer. This object is used to reset the microswitch. Therefore, to assure proper resetting, the operator must be sure this object is moving toward the microswitch. The menu used during manual resetting is shown in figure 7. Once the operator is ready to reset the designated motor, he/she presses the B key.

```
MOTOR      - 4  IS BEING RESET

B - BEGIN THE RESETTING OF THE MOTOR
C - CHANGE THE DIRECTION OF THE MOTOR
S - SLOW DOWN NEAR THE SWITCH
F - FASTER, IF SLOW DOWN WAS AT THE WRONG PLACE

X - STOPS THE RESETTING IF IN TROUBLE AND CHANGE DIRECTION

      ALL OTHER KEYS STOP THE MOTION
      PRESS B AGAIN TO CONTINUE
```

Figure 7. Menu for manual resetting of motors.

If the reset device is moving away from the microswitch, the operator must press C to change the direction of the motor. As the object nears the microswitch, the operator presses S to slow down the movement. If the operator slows the movement in the wrong place, he/she can press F to resume the faster speed. Once the motor is reset, the software checks it in the same

manner as a motor that is already reset. If during resetting, the operator gets into trouble, such as if the motor starts to bind, he/she can press X and the motor stops instantaneously.

When motors three through eight are reset, if there is a motor connected to position two, it now must be reset. The menu used to reset motor two is shown in figure 8. If the motor is in the correct location, the operator presses O for O.K. However, if it is not in the correct location the operator presses B to begin resetting and then presses D when done. The other keys used are similar to those used for resetting the other motors. The last motor to be reset is the hand motor and this is done by using the menu shown in figure 9. This is accomplished by using a closed hand as a reference point. If the hand is closed, the operator presses D for done. However, if it is open, the operator must begin resetting by pressing B. The hand slowly moves ten steps, then waits for the operator to press S for stop, meaning hand is closed, or C to continue closing. The operator must be careful not to close the hand too far, since damage to the RHINO could occur. Once the hand is reset, the operator is ready for the RHINO teach mode.

RESET MOTOR 2

B - BEGIN THE RESETTING OF MOTOR 2
O - O.K., THIS IS THE STARTING LOCATION

C - CHANGE DIRECTION OF MOTOR 2
D - DONE, THIS IS THE STARTING LOCATION
F - FAST, MOVE MOTOR FASTER
S - SLOW DOWN NEAR STARTING LOCATION

Figure 8. Menu used for resetting motor 2.

HAND RESET

D - DONE, THE HAND IS ALREADY CLOSED
B - BEGIN RESETTING THE HAND IF NOT CLOSED

S - STOPS HAND WHEN IT IS CLOSED
C - CONTINUE CLOSING WHEN THE HAND STOPS MOVING

Figure 9. Menu used for resetting the hand motor.

After resetting is completed, the RHINO teach mode menu in figure 10 appears on the screen. The only keys explained are the ones not used to move the motor, which were explained in the section on software. By pressing the * key, the system can be reinitialized in one of three ways: delete the current routine, reset the system, or change the motor configuration. The O key increments the step number which causes that location to be an end point when the routine is run. The pause status is changed to either ON or OFF by pressing P. To reset the RHINO, the operator presses shift and @. If a mistake is made in a step, the arrow keys can be used to go back and correct the error. This is done

by erasing the current step (right arrow) or erasing the current step and the previous step (left arrow). The L key is used to change the increment values. To move a motor a long distance, greater than the high increment, the + key is used. When the operator is ready to run the routine, he/she presses the > key. Once entered, the operator is asked how many times to repeat this routine. To exit the teach mode, the operator presses the ? key. The CLEAR key is used to set the start location; this must be done at the start of the routine, at step number one.

```

                MOTOR CONFIGURATION
                1 - GRASP or FINGERS
                2 - CAROUSEL
                3 - WRIST - AZIMUTHAL
                4 - FOREARM
                5 - SHOULDER
                6 - WAIST
                7 - LINEAR BASE
                8 - WRIST - ROTATIONAL
** MAY CHANGE LOCATION OF MOTORS 2,6,7,8 ONLY **
MOTORS AND THERE POSSIBLE LOCATIONS ('#' - DEFAULT LOCATION)
CAROUSEL ('2',6,7,8)           CONVEYOR (2 ONLY)
LINEAR BASE (6,'7',8)         WAIST ('6',7,8)
WRIST - ROTATIONAL (6,7,'8')  NONE - NO MOTOR ATTACHED

ENTER LOCATION TO BE CHANGED (2,6,7,or 8) or 0 if O.K. ?

```

Figure 10. The RHINO teach mode menu.

The other information displayed on the screen is as follows: the PAUSE status, position of each motor in steps, step number, and the high and low increment values. As was explained in the section on software, a loop of routines can be run together to accomplish more complex tasks. The) key is pressed to enter the loop mode. A sample loop menu is

shown in figure 11. This menu is similar to the others and is selfexplanatory. To return to the teach mode, the T key is pressed.

```

                RUN A LOOP OF ROUTINES
                LOOP TITLE - EXAMPLE
ROUTINE TITLE  # REPEATS      ROUTINE TITLE  # REPEATS
1  EXAMPLE1    1              11
2  EXAMPLE2    1              12
3  EXAMPLE1    1              13
4  EXAMPLE3    1              14
5  EXAMPLE1    1              15
6
7
8
9
10

```

A-ADD, C-CHANGE, D-DELETE, L-LOAD, R-RUN, S-SAVE, T-TEACH MODE ?

Figure 11. A sample loop menu.

Once a routine is created, it can be saved by using the disk file access. The = key is used to get into the file access mode, which is shown in figure 12. Using this menu, files can be saved on the disk or loaded from the disk. A remark is stored with the file to add useful comments about the routine or loop. A directory to all files can be obtained, and while in this mode files can be deleted. A sample directory is shown in figure 13. Included in the software is complete error checking. Therefore, if the operator makes a mistake, he/she does not lose everything. Sample errors are shown in figure 14. To distinguish between the routine files and the loop files, an extension

is added to the end of the file name by the software. The extension for routine files is /RHN and loop files is /LOP.

DISK FILE ACCESS MENU FOR ROUTINE' S FILES

1 - SAVE FILE ON DISK 6 - CHANGE THE FILENAME
 2 - LOAD A FILE FROM DISK 7 - CHANGE THE REMARK
 3 - CHANGE THE TYPE OF FILE BEING ACCESSED
 4 - DIRECTORY OF DISK FILES AND DELETING FILES FROM DISK
 5 - RETURN TO TEACH MODE
 ENTER CHOICE ?

FILENAME (8 CHARACTERS OR LESS) - EXAMPLE1
 REMARK (240 CHARACTERS OR LESS)
 THIS IS A REMARK

Figure 12. Disk file access menu.

DRIVE :0	EXAMPLE1/RHN	EXBACK1/RHN	EXAMPLE2/RHN
EXBACK2/RHN	EXAMPLE3/RHN	EXBACK3/RHN	EXAMPLE/LOP
EXBACK/LOP	EXAMPLE4/RHN	EXBACK4/RHN	

THE FILES WITH /RHN AT THE END ARE ROUTINE FILES AND
 /LOP AT THE END ARE LOOP FILES

DID YOU WANT TO DELETE (D) A FILE OR RETURN (R) TO MENU ?

Figure 13. Sample directory of all disk files.

***** DISK ERROR *****

*** FILE NOT FOUND ***

THE FILE NOFILE/RHN TRYING TO BE KILLED DOES NOT EXISTS
PLEASE CHECK THE DIRECTORY AND TRY AGAIN

PRESS ANY KEY TO CONTINUE

***** DISK ERROR *****

*** BAD FILENAME ***

FILENAME 34BAD/LOP IS NOT AN APPROPRIATE FILENAME. TRY AGAIN

PRESS ANY KEY TO CONTINUE

Figure 14. Sample error messages.

SAMPLE EXPERIMENTS

Typical applications have been programmed as laboratory experiments (labs) for the trainee. From these experiments, the trainee will learn some of the programming problems associated with robot applications. Also the trainee will see how time-consuming it is to program a robot. When the trainee has completed some of the laboratory assignments, he/she should be encouraged to develop his/her own routines. This will encourage creativity. The would-be operator will also learn some of the limitations of a robot.

Sample labs were created and stored on the disk for demonstration of the system. The first is a routine to move a box from the table to the conveyor. The motor configuration is the same as the default except the conveyor is connected to motor position two. This routine is named EXAMPLE4 and the box should be located at x 2-3 and y 6-9 on the table. The next routine is EXAMPLE1 and it moves a box from the same location to the carousel which deposits the box into the bin. Finally, a loop was created which moves three boxes from the table to the bin. The motor configuration used for the routine EXAMPLE1 and the loop example entitled EXAMPLE is the default configuration. As can be seen in figure 11, the loop includes EXAMPLE1 for placing the box into the bin and EXAMPLE2 and EXAMPLE3 to

move the next boxes into the location used by EXAMPLE1. The two extra boxes are located at x 6-7, y 6-9 and x 10-11, y 6-9. These examples demonstrate many of the capabilities of the user-friendly robot training aid. Pictures of the system while running these sample labs are included in Appendix A.

PROBLEMS ENCOUNTERED

The problems encountered in this development were typical of the problems encountered when using many of today's industrial robots. The control system on the robot used in this development did not permit control of motor speed. This lack of control results in the inability to conveniently move the hand along a straight line. The lack of acceleration and deceleration capabilities in the system result in instabilities at the beginning and end of axis motion sequences. The resultant vibrations of an axis sometimes result in an incorrect setting of a starting location. Because of the low precision of the RHINO, repeatability of a routine is low. Also some hardware problems have occurred because of overheating.

The only software problem encountered resulted from the lack of memory in the microcomputer. To overcome this lack of memory, the program was divided into two smaller programs and the size of the arrays for storage of axis positions were made smaller. This means that fewer steps are allowed in any routine. Since the BASIC language was used, the movement of the RHINO is sometimes slow. If this program were written in machine code, the robot would run smoother and faster. However, it was felt that the BASIC language was fast enough for learning purposes.

FUTURE AND IMPROVEMENTS

There are a few things that can be done to improve this training aid. Off-line programming could be included by using angles of the joints and 3-D coordinate system. However, the RHINO would be more apt to collide with objects in this type of program. A voltage regulator on the motors could be installed so that the motors would have controllable acceleration and deceleration. By doing this, straight line motion would be obtainable and probably with higher precision. More microswitches could be included and checked during running to see if any steps had been lost. However, new circuits and new interfacing hardware would have to be designed and built.

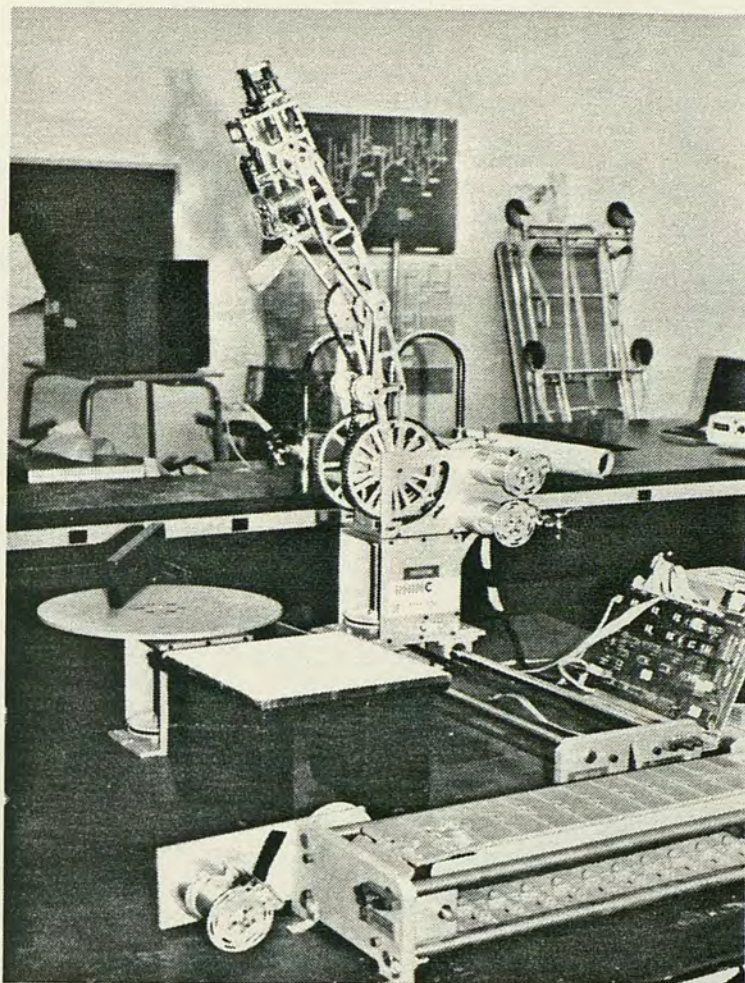
This training aid should be useable by students and would-be operators. Even if new technology in robotics is developed, the trainee will get an introduction to robotics by this aid. Other devices, such as an NC milling machine or NC lathe, could be included to make the system similar to a factory situation.

CONCLUSIONS

There will be a greater need for robot operators in the near future and these would-be operators need to be trained. A 'user-friendly' robot training aid was developed to train operators for the growing number of robots. This aid will help the trainee learn how to program a robot using on-line programming. Since this training aid is user-friendly and has safety features, it requires little or no previous experience with robots or computers and minimal supervision. Therefore, this training aid should be useful to colleges and companies wanting to train robot operators in on-line robot programming. This aid should allay any fear concerning robots, by allowing the operator to develop confidence in controlling a robot.

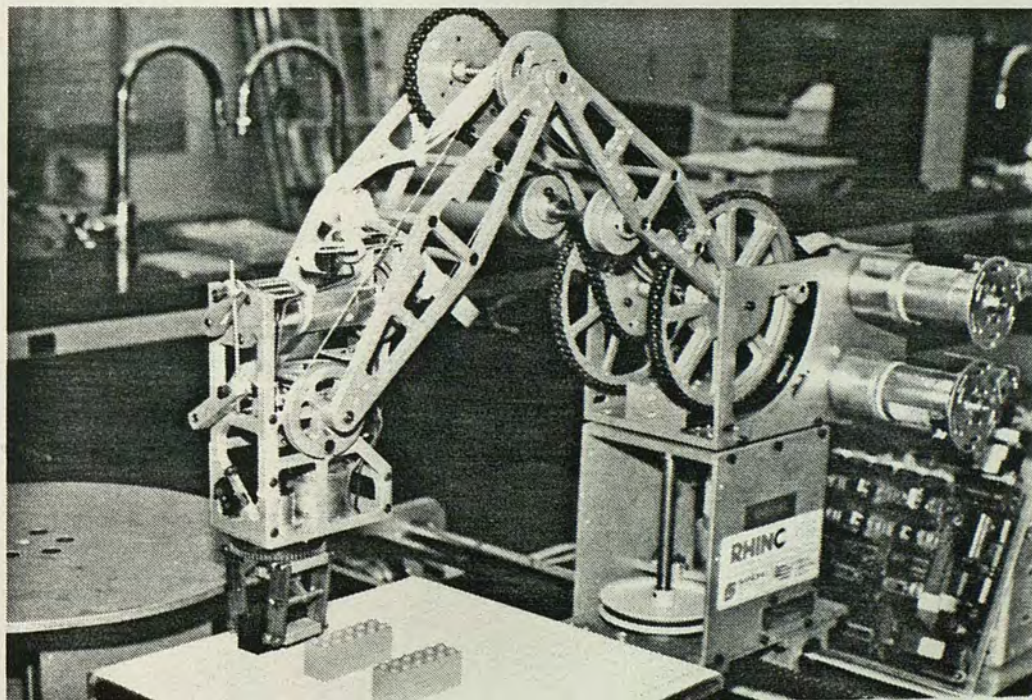
APPENDIX A

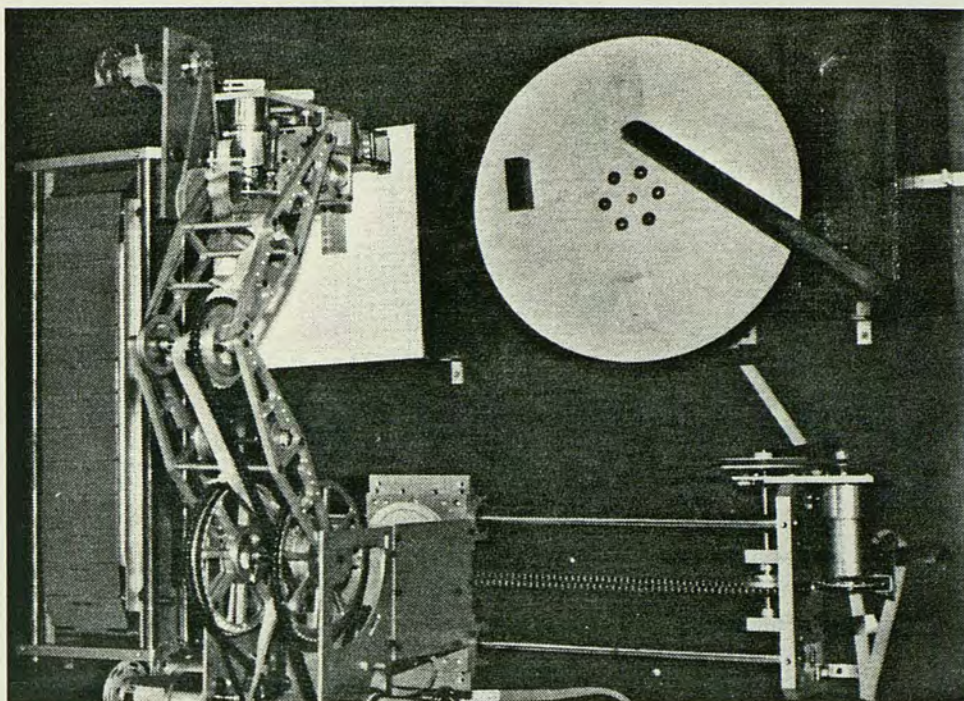
Extra Pictures of the RHINO



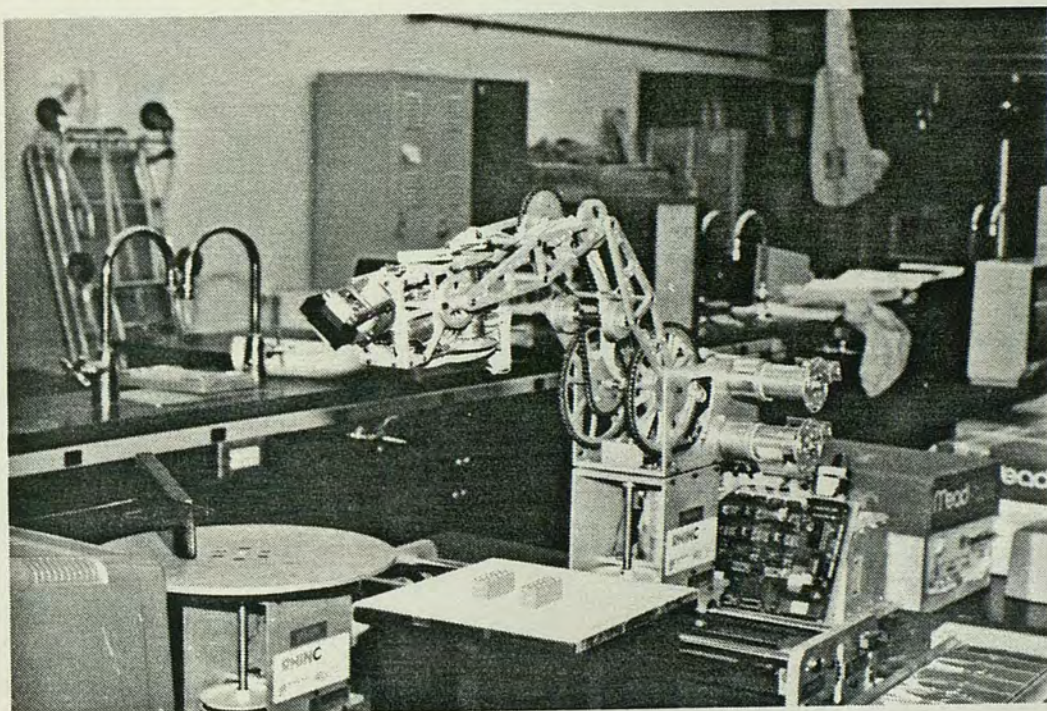
The RHINO is in its
reset position
(to the left).

RHINO picking up a
box while running
EXAMPLE1 (below).





A box is on the carousel ready to be deposited into the bin.



The RHINO is moving a box to a new location.

APPENDIX B

Sample Program Output

Series of screens printed to the user
as an introduction to the system

Welcome to the RHINO XR-2 'User-Friendly' Robot Training Aid. This system should help you understand how to program a robot.

The RHINO is moved by using the keyboard as a teach pendant. Up to eight motors can be connected and controlled. A column of keys is used to move an axis. The number identifying the axis moved is identified by the number key at the top of the column. Thus, 1, Q, A, and Z move motor 1. There are two different increments of movement, high and low.

Row 1	high increment, positive direction	(Example key 1)
Row 2	low increment, positive direction	(Example key Q)
Row 3	low increment, negative direction	(Example key A)
Row 4	high increment, negative direction	(Example key Z)

The maximum high increment is 127. If a longer move is needed, an option in the teach mode menu provides for it.

A routine, which is a series of steps, will be created from the movements you tell the RHINO to store. The RHINO will move from one point to another when the routine is running (Point-to-Point mode), with each point being the end of a step. Therefore when you define a step, remember all joints will move at once, not one at a time as when you programmed it. Once a routine is created, it can be stored on the disk by using the disk access mode (= key). Be sure to increment step # (0 key).

A routine can be run in two modes: (1) The Pause ON mode where the RHINO stops at the end of each step for you to see if it is O.K. (Use this at least the first time you run a routine). (2) In the Pause OFF status the RHINO runs without stopping. The routine can be repeated as many times as needed, but it will go back to the start location between repeats. The maximum number of steps in a routine is one hundred.

To program a complex task, up to 20 routines can be run sequentially as a loop. Each routine can be repeated before going on to the next routine. Experiment with this option. You will find it very useful.

Before entering the teach mode, the motor configuration must be entered and the RHINO reset to the start location.

NOTE: An emergency switch is in the 110V power cord of the RHINO power supply. This must be turned OFF if any motor on the RHINO or peripheral equipment starts to bind. Otherwise, there will probably be damage to the RHINO. There is already a safety feature to prevent the RHINO from colliding into itself. However, it can hit other objects, so be CAREFUL, especially on long moves!

If the emergency switch is used, you must press the red reset button on the computer and start over.

Before proceeding, there are a few things to be checked and done in the following order. Please, these must be done!

1. Is the RS-232 cable plugged into the bottom of the Model III computer and into the RHINO power supply ?
2. Is the emergency switch in ON position ?
3. Is the RHINO power supply plugged into the 110V outlet ?
4. Is the dip switch on the controller board at position 3 ?
5. Connect the motors in the configuration you need. Note you can only change motors connected at positions 2, 6, 7, or 8.
6. Turn the controller logic switch ON (large switch). Note the power supply light should come on, if not check all steps.
7. Turn the motor power switch ON (small switch).

APPENDIX C

Program Listing of START/BLD

Commands used to initialize the system

PROGRAM LISTING OF START/BLD

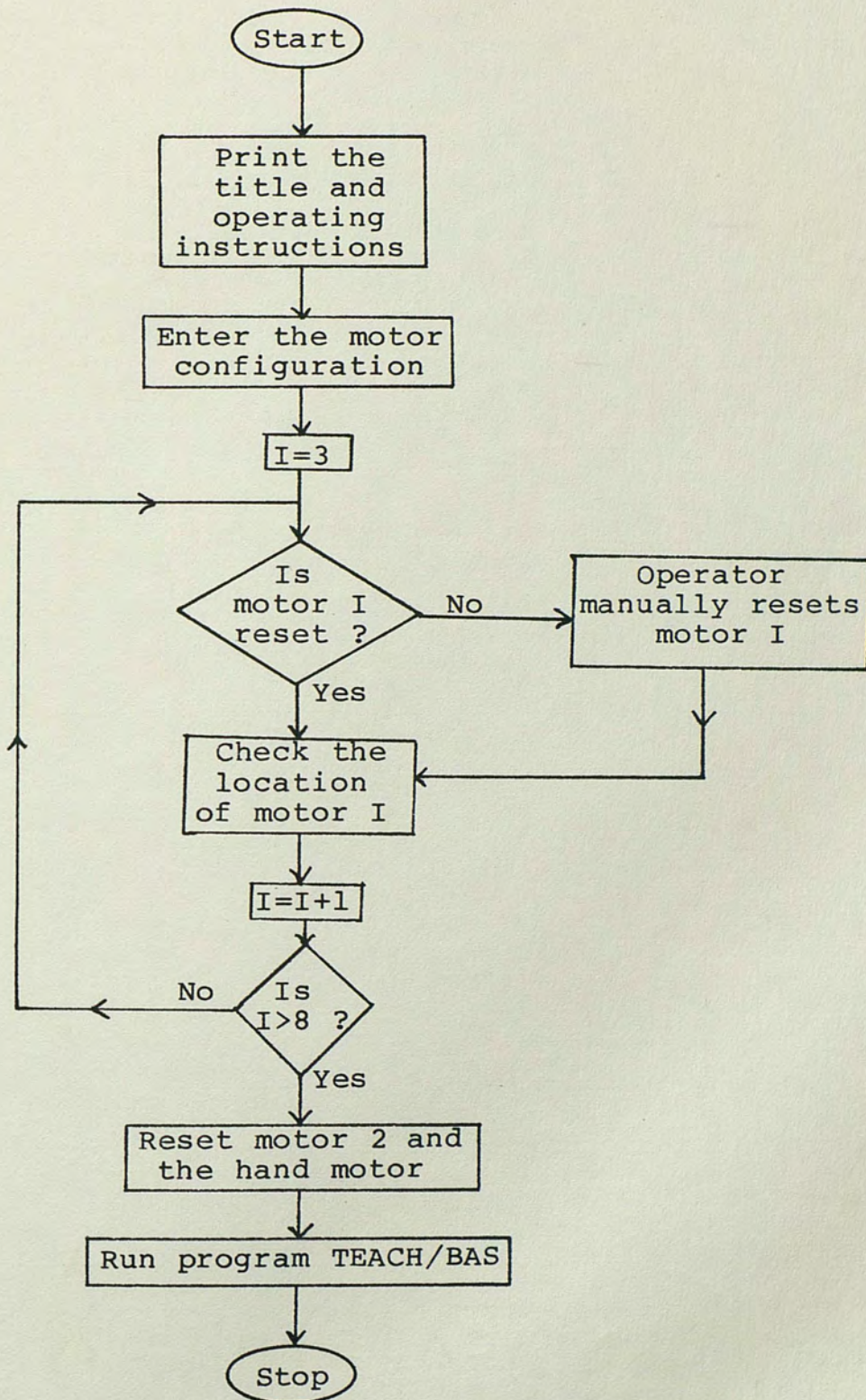
```
SETCOM (BAUD=9600,STOP=2,PARITY=2,WORD=7,WAIT)  
BASIC  
3
```

```
RUN "STARTUP/BAS "
```

APPENDIX D

Flowchart for STARTUP/BAS

FLOWCHART FOR STARTUP/BAS



APPENDIX E

Program Listing of STARTUP/BAS

```

100 '
110 ' *****
120 ' *
130 ' *  ROBOT TRAINING AID      MAY 1983  *
140 ' *                      BY          *
150 ' *          DON WASHBURN,         *
160 ' *
170 ' *****
180 '
190 ' CMD"B","OFF"      'DISABLE THE BREAK KEY
200 ' NOTE : RHINO USES LABELS FOR MOTORS AS A TO H BUT, I
      CHANGED
210 '      THEM TO  NUMBERS 1 TO 8 TO BE EASIER TO USE IN
      TEACH MODE.
220 ' *****
230 ' * PRINT TITLE OUT *
240 ' *****
250 CLS:PRINTCHR$(23):PRINT:PRINT:PRINT
260 PRINT TAB(7)"ROBOT TRAINING AID" : PRINT
270 PRINT TAB(12)"MAY 1983":PRINT:PRINT
280 PRINT TAB(15)"BY"
290 PRINT TAB(10)"DON WASHBURN"
300 FOR Q=1 TO 200
310 A$=INKEY$:IF A$<>" " THEN 400
320 NEXT Q
330 PRINT:PRINT:PRINT:PRINT TAB(3)"PRESS ANY KEY TO
      CONTINUE"
340 A$=INKEY$:IF A$="" THEN 340
350 '
360 ' *****
370 ' * PRINT OUT INFORMATION AND THE START UP STEPS *
380 ' *****
390 '
400 CLS : PRINT"      Welcome to the RHINO XR-2
      'User-Friendly' Robot Training"
410 PRINT"Aid. This system should help you understand how
      to program a" : PRINT "robot."
420 PRINT"      The RHINO is moved by using the keyboard as a
      teach"
430 PRINT"pendant. Up to eight motors can be connected and
      controlled."
440 PRINT"A column of keys is used to move an axis. The
      number"
450 PRINT"identifying the axis moved is identified by the
      number key at"
460 PRINT"the top of the column. Thus, 1, Q, A, and Z move
      motor 1."
470 PRINT"There are two different increments of movement,
      high and low."
480 PRINT" Row 1 high increment, positive direction

```

```

      (Example key 1)"
490 PRINT" Row 2 low increment, positive direction
      (Example key 0)"
500 PRINT" Row 3 low increment, negative direction
      (Example key A)"
510 PRINT" Row 4 high increment, negative direction
      (Example key Z)"
520 PRINT"The maximum high increment is 127. If a longer
      move is needed,"
530 PRINT"an option in the teach mode menu provides for it."
540 GOSUB 550 : GOTO 590
550 PRINT @979,"PRESS ANY KEY TO CONTINUE";
560 FOR KK=1 TO 30 : A#=INKEY$ : IF A#<>" THEN 580 : NEXT
      KK
570 PRINT @979,CHR$(30); : FOR KK=1 TO 10 : NEXT KK : GOTO
      550
580 CLS : RETURN
590 PRINT"      A routine, which is a series of steps, will
      be created"
600 PRINT"from the movements you tell the RHIND to store.
      The RHIND will"
610 PRINT"move from one point to another when the routine is
      running"
620 PRINT"(Point-to-Point mode), with each point being the
      end of a step."
630 PRINT"Therefore when you define a step, remember all
      joints will move"
640 PRINT"at once, not one at a time as when you programmed
      it. Once a"
650 PRINT"routine is created, it can be stored on the disk
      by using the"
660 PRINT"disk access mode (= key). Be sure to increment
      step # (0 key)."
```

670 PRINT" A routine can be run in two modes: (1) The
Pause ON mode"

680 PRINT"where the RHIND stops at the end of each step for
you to see"

690 PRINT"if it is O.K. (Use this at least the first time
you run a"

700 PRINT"routine). (2) In the Pause OFF status the RHIND
runs without"

710 PRINT"stopping. The routine can be repeated as many
times as needed,"

720 PRINT"but it will go back to the start location between
repeats. The"

730 PRINT"maximum number of steps in a routine is one
hundred."

740 GOSUB 550

750 PRINT" To program a complex task, up to 20 routines
can be run"

760 PRINT"sequentially as a loop. Each routine can be

```
repeated before"
770 PRINT"going on to the next routine. Experiment with
this option."
780 PRINT"You will find it very useful."
785 PRINT" Before entering the teach mode, the motor
configuration"
786 PRINT"must be entered and the RHINO reset to the start
location."
790 PRINT" NOTE: An emergency switch is in the 110V
power cord of"
800 PRINT"the RHINO power supply. This must be turned OFF if
any motor on"
810 PRINT"the RHINO or peripheral equipment starts to bind.
Otherwise,"
820 PRINT"there will probably be damage to the RHINO. There
is already a"
830 PRINT"safety feature to prevent the RHINO from colliding
into itself."
840 PRINT"However, it can hit other objects, so be CAREFUL,
especially on"
850 PRINT"long moves!"
860 PRINT" If the emergency switch is used, you must
press the red"
870 PRINT"reset button on the computer and start over." :
GOSUB 550 : PRINT:PRINT" Before proceeding, there
are a few things to be checked and"
880 PRINT"done in the following order. Please, these must
be done!"
890 PRINT:PRINT"1. Is the RS-232 cable plugged into the
bottom of the Model III"
900 PRINT" computer and into the RHINO power supply ?"
910 PRINT"2. Is the emergency switch in ON position ?"
920 PRINT"3. Is the RHINO power supply plugged into the 110V
outlet ?"
930 PRINT"4. Is the dip switch on the controller board at
position 3 ?"
940 PRINT"5. Connect the motors in the configuration you
need. Note you"
950 PRINT" can only change motors connected at positions
2, 6, 7, or 8."
960 PRINT"6. Turn the controller logic switch ON (large
switch). Note the"
970 PRINT" power supply light should come on, if not check
all steps."
980 PRINT"7. Turn the motor power switch ON (small switch)."
```

990 GOSUB 550

```
1000 PRINT:PRINT:PRINT
1010 PRINTTAB(14)"You are now ready to start using the"
1020 PRINT:PRINTTAB(12)"RHINO 'User Friendly' Robot Training
Aid"
1030 PRINT:PRINT:PRINT"Please enter a title for the routine
```

```

you are going to use."
1040 PRINT@576,"TITLE OF ROUTINE (8 characters or less) - ?
";
1050 ?
1060 ? *****
1070 ? * INITIALIZE THE SYSTEM *
1080 ? *****
1090 ?
1100 CLEAR 500
1110 DEFINT A-Z
1120 DEFUSR0=85 'SEND CHARACTERS ON RS-232
1130 DEFUSR1=80 'GET CHARACTERS OFF OF RS-232
1140 ?
1150 LL=8 : GOSUB 1410
1160 F5#=B# : IF F5#="" THEN F5#="RHINO"
1170 GOSUB 3280 'GET MOTOR CONFIGURATION
1180 GOSUB 1560 'RESET RHINO - FIRST TIME
1190 ?
1200 ? *****
1210 ? * *
1220 ? * STORE VARIABLES FOR MAIN PROGRAM *
1230 ? * *
1240 ? *****
1250 ?
1260 ON ERROR GOTO 2970
1270 CLS:PRINT:PRINTCHR$(23):PRINT :PRINT:PRINT :PRINT
TAB(4)"LOADING RHINO TEACH MODE"
1280 OPEN "0",1,"RHINO/SAV"
1290 PRINT #1,F5#
1300 FOR I=1 TO 8 : PRINT #1,NC(I),XC(I) : NEXT I
1310 FOR I=1 TO 4 : PRINT #1,NN(I) : NEXT I
1320 CLOSE
1330 ? *****
1340 ? * LOAD AND RUN MAIN PROGRAM *
1350 ? *****
1360 RUN"TEACH/BAS"
1370 ? STOP ' USED FOR DEBUGGING
1380 ? *****
1390 ? * INPUT ARRAY OF LENGTH LL *
1400 ? *****
1410 B#=""
1420 A#=INKEY# : IF A#="" THEN 1420
1430 IF ASC(A#)=13 THEN RETURN
1440 IF A#="," THEN PRINT " NO COMMAS"; :FOR C=1 TO 2000 :
NEXT C :PRINT" "; :GOTO
1420
1450 IF ASC(A#)=8 THEN L1=LEN(B#):IF L1=0 THEN 1410 ELSE
PRINT A#;:LL=LL+1:IF L1=1 THEN 1410 ELSE
B#=LEFT$(B#,L1-1) : GOTO 1420
1460 IF ASC(A#)>127 OR ASC(A#)<32 THEN 1420
1470 B#=B#+A# : LL=LL-1 : PRINT A#;

```

```

1480 IF LL=0 THEN RETURN ELSE 1420
1490 '
1500 ' *****
1510 ' *
1520 ' *      RESET FOR THE RHINO FOR THE FIRST TIME      *
1530 ' *
1540 ' *****
1550 '
1560 DIM IN(6),TN(6)      'DIMENSION LOCAL VARIABLE
1570 CLS:PRINT:PRINT :PRINT:PRINTCHR$(23) :PRINT: PRINT
      TAB(6)"RESETTING THE RHINO" 'THE FIRST TIME THE RHINO
      IS RESET
1580 GOSUB 2140
1590 FOR CJ=1 TO 6 : IN(CJ)=TN(CJ) : NEXT CJ
1600 FOR J=1 TO 6
1610 IF J>3 THEN IF NN(J-2)=6 THEN 1630
1620 ON (1+IN(J)) GOSUB 2320 ,2210
1630 NEXT J
1640 ' *****
1650 ' * CHECK MOTOR 2 - MOVE IT TO START LOCATION *
1660 ' *****
1670 IF NN(1)=6 THEN 1870 ' NO MOTOR 2 SO CANNOT RESET
1680 CLS:PRINT:PRINTTAB(20)"RESET MOTOR 2":PRINT:PRINT
1690 PRINT"B - BEGIN THE RESETTING OF MOTOR 2":PRINT"D -
      O.K., THIS IS THE STARTING LOCATION" : PRINT:PRINT"C -
      CHANGE DIRECTION OF MOTOR 2"
1700 PRINT"D - DONE, THIS IS THE STARTING LOCATION":PRINT"F
      - FAST, MOVE MOTOR FASTER":PRINT"S - SLOW DOWN NEAR
      STARTING LOCATION":PRINT:PRINT
1710 A#=INKEY$ : IF A#="" THEN 1710
1720 IF A#="O" THEN 1870 ELSE IF A#<>"B" THEN 1710
1730 FOR Q=1 TO 50 : NEXT Q
1740 MT=66:SG=45:CT=100
1750 GOSUB 2560
1760 GOSUB 2690 : IF E=0 THEN FOR GG=1 TO 20:A#=INKEY$:IF
      A#<>" " THEN 1780 ELSE NEXT GG : GOSUB 2560 : GOTO 1760
1770 A#=INKEY$: IF A#="" THEN 1760
1780 IF A#="C" THEN GOSUB 2890 : IF SG=43 THEN SG=45 : GOTO
      1750 ELSE SG=43 : GOTO 1750
1790 IF A#="S" THEN CT=CT-CT/2:IF CT<5 THEN CT=1 : GOTO 1750
      ELSE 1750
1800 IF A#="D" THEN GOSUB 2890 : GOTO 1870 ' DONE
1810 IF A#="F" THEN CT=CT+CT : IF CT<5 THEN CT=10 : GOTO
      1750 ELSE IF CT>127 THEN CT=127 : GOTO 1750 ELSE 1750
1820 GOSUB 2890 : FOR C=1 TO 250 : NEXT C : GOTO 1680
1830 '
1840 ' *****
1850 ' * RESET THE GRASP OR FINGERS (MOTOR 1) *
1860 ' *****
1870 CLS:PRINT:PRINTTAB(20)"HAND RESET":PRINT:PRINT"D -
      DONE, THE HAND IS ALREADY CLOSED":PRINT"B - BEGIN

```

```

RESETTING THE HAND IF NOT CLOSED":PRINT:PRINT"S - STOPS
HAND WHEN IT IS CLOSED":PRINT"C - CONTINUE CLOSING WHEN
THE HAND STOPS MOVING"
1880 A#=INKEY#:IF A#="" THEN 1880
1890 IF A#="D" THEN 2070
1900 IF A#<>"B" THEN 1880
1910 FOR Q=1 TO 50 : NEXT Q           ' + = CLOSED (43)
1920 MT=65:SG=45:CT=10 :GOSUB2560   ' - = OPEN (45)
1930 SG=43:CT=1
1940 FOR I=1 TO 10
1950 GOSUB 2560
1960 FOR II=1 TO 10
1970 A#=INKEY# : IF A#="S" THEN 2070
1980 IF A#="C" THEN 1940
1990 NEXT II
2000 NEXT I
2010 A#=INKEY#:IF A#="" THEN 2010
2020 IF A#="C" THEN 1940
2030 IF A#<>"S" THEN 2010
2040 ' *****
2050 ' *   SET UP RHINO FOR START, OPEN HAND   *
2060 ' *****
2070 FOR I=1 TO 8 :TT(I)=0:NEXT I
2080 MT=65 : SG=45:CT=75:GOSUB2560
2090 RETURN
2100 ' *****
2110 ' *   FIND WHICH MICROSWITCHES ARE CLOSED *
2120 ' *   SET FLAG TN(#) - 0 OPEN, 1 - CLOSED *
2130 ' *****
2140 GOSUB 2800
2150 FOR CJ=1 TO 6 :TN(CJ)=1 :NEXT CJ
2160 FOR CJ=6 TO 1 STEP -1
2170 IR = I-2[(CJ-1) : IF IR<0 THEN 2190
2180 TN(CJ)=0 : I=I-2[(CJ-1) ' SWITCH IS OPEN
2190 NEXT CJ : RETURN
2200 ' *****
2210 ' *   MOTOR RESET BUT CHECK POSTION   *
2220 ' *****
2230 CLS:PRINT:PRINT :PRINTCHR$(23) :PRINT:PRINT :PRINT
TAB(7)"CHECKING MOTOR  ";CHR$(50+J)
2240 MT=66+J:SG=43:CT=30:GOSUB2560
2250 GOSUB 2690 : IF E>0 THEN 2250
2260 FOR C=1 TO 100 : NEXT C
2270 GOSUB 2140 : IF TN(J)=1 THEN 2240
2280 SG=45:GOSUB 2800 :ST=I:CT=1
2290 GOSUB 2800 :IF I<ST THEN RETURN
2300 GOSUB 2560 : GOTO 2290
2310 ' *****
2320 ' *   MOTOR NEEDS TO BE RESET AND THEN CHECKED *
2330 ' *****
2340 MT=66+J : CT=50

```



```

2350 SG=43
2360 CLS : PRINT:PRINT"MOTOR    -  ";CHR$(MT-16);" IS BEING
      RESET":PRINT:PRINT"B - BEGIN THE RESETTING OF THE
      MOTOR":PRINT"C - CHANGE THE DIRECTION OF THE
      MOTOR":PRINT"S - SLOW DOWN NEAR THE SWITCH"
2370 PRINT "F - FASTER, IF SLOW DOWN WAS AT THE WRONG PLACE"
2380 PRINT:PRINT"X - STOPS THE RESETTING IF IN TROUBLE AND
      CHANGE DIRECTION":PRINT:PRINT"    ALL OTHER KEYS STOP
      THE MOTION":PRINT"    PRESS B AGAIN TO CONTINUE""
2390 A#=INKEY#:IF A#<>"B" THEN 2390
2400 FOR Q=1 TO 50 : NEXT Q
2410 GOSUB 2800 : ST=I
2420 GOSUB 2800 : IF I<ST THEN GOSUB 2890 : CLS : GOTO 2230
2430 A#=INKEY# : IF A#<>" " THEN 2460
2440 GOSUB 2690 :IF E=0 THEN FOR GG=1 TO 20 :A#=INKEY#:IF
      A#<>" " THEN 2460 ELSE NEXT GG :GOSUB 2560: GOTO 2420
2450 A#=INKEY#:IF A#="" THEN 2440
2460 IF A#="C" THEN IF SG=43 THEN SG=SG+2 :GOTO 2420 ELSE
      SG=43: GOTO 2420
2470 IF A#="S" THEN CT=1 : GOTO 2420
2480 IF A#="F" THEN CT=50 : GOTO 2420
2490 IF A#="X" THEN GOSUB 2890 : IF SG=45 THEN 2350 ELSE
      SG=45 : GOTO 2360
2500 GOSUB 2890 : FOR C=1 TO 250 : NEXT C : GOTO 2360
2510 '
2520 ' *****
2530 ' * SEND A MOVE TO A MOTOR *
2540 ' *****
2550 '
2560 POKE 16880,MT :X=USR0(0) ' OUTPUT THE MOTOR AND SIGN
      OF MOVE
2570 POKE 16880,SG :X=USR0(0)
2580 CC=CT:IF CC>=100 THEN POKE 16880,49 :X=USR0(0)
      :CC=CC-100 ' CHECK IF OVER 100, IF SO OUTPUT A ONE
2590 W=INT(CC/10) : CC=CC-W*10 ' FIND TENS + ONES
2600 POKE 16880,W+48 : X=USR0(0) ' THEN OUTPUT THEM
2610 POKE 16880,CC+48 :X=USR0(0)
2620 POKE 16880,13 : X=USR0(0) ' OUTPUT A CR TO END THE
      MESSAGE
2630 RETURN
2640 '
2650 ' *****
2660 ' * GET THE NUMBER OF COUNTS LEFT *
2670 ' *****
2680 '
2690 POKE 16880,MT : X=USR0(0) ' OUTPUT THE MOTOR AND A
      '?'
2700 POKE 16880,63 : X=USR0(0)
2710 X=USR1(0) ' GET RESPONSE
2720 E=PEEK(16872)-32 ' A RHINO FIX
2730 IF E<0 THEN E=128+E

```

```

2740 RETURN          ' E = # OF COUNTS LEFT IN MOVE
2750 '
2760 ' *****
2770 ' * GET MIRCOSWITCH'S VALUE *
2780 ' *****
2790 '
2800 POKE 16880,73 : X=USR0(0) ' OUTPUT AN 'I' TO GET THE
STATUS OF MICRO-SWITCHES
2810 X=USR1(0) ' GET RESPONSE
2820 I=PEEK(16872)-32 ' I= VALUE FROM 0 TO 63
2830 RETURN          ' I=0 - ALL CLOSED I=63 - ALL OPEN
2840 '
2850 ' *****
2860 ' * STOP A MOTOR IN MOTION *
2870 ' *****
2880 '
2890 POKE 16880,MT : X=USR0(0) ' STOP MOTOR MT - OUTPUT
MOTOR AND A 'X'
2900 POKE 16880,88 : X=USR0(0)
2910 RETURN
2920 '
2930 ' *****
2940 ' * DISK ERROR SUBROUTINE - FOR SAVING AND LOADING *
2950 ' *****
2960 '
2970 E2=ERR/2+1 ' GET ERROR CODE
2980 IF E2<51 THEN ERROR E2 ' NOT A DISK ERROR
2990 CLOSE:CLS:PRINT:PRINT" ***** DISK ERROR
*****":PRINT
3000 IF E2=53 THEN 3170 ELSE IF E2=54 THEN 3120
3010 IF E2=62 THEN RESUME 3030 ELSE IF E2=65 THEN 3110
3020 RESUME ' CAUSE ERROR TO HAPPEN AGAIN - SAME LINE
3030 ON ERROR GOTO 3040 : KILL "RHINO/SAV" : PRINT"
**** DISK IS FULL ****" : PRINT: GOTO 3050
3040 E2=ERR/2+1 : IF E2<>54 THEN ERROR E2 ELSE PRINT "
**** DISK IS FULL ****":PRINT
3050 PRINT"To elevate this problem you could (1) delete some
old files
using disk access mode or (2) put a new RHINO master disk
(with
room on it) in drive #0 and try saving on it.
NOTE: MUST changethe disk in drive #0 before pressing a key
to
continue, ";
3060 PRINT " also never insert a disk when the red
light is on !! (3) can not made enough room so, must
quit and try again later."
3070 PRINT"ENTER CHOICE (AFTER DOING REQUIRED STEPS) ?";
3080 A$=INKEY$ : IF A$="" THEN 3080
3090 IF A$="1" THEN RESUME 3960
3100 IF A$="2" THEN RESUME 1270

```

```

3110 IF A$<>"3" THEN 3070 ELSE CMD"S"
3120 PRINT "      **** BAD FILENAME ****" :PRINT :PRINT
      "FILENAME ";FF$;" IS NOT AN APPROPRIATE FILENAME SO,
      TRY AGAIN" : PRINT : PRINT: PRINT"PRESS ANY KEY TO
      CONTINUE"
3130 A$=INKEY$: IFA$="" THEN 3130 ELSE RESUME 3960
3140 PRINT "      **** FILE NOT FOUND ****" :PRINT :PRINT
      "FILENAME ";FF$;" DOES NOT EXIST - PLEASE CHECK THE
      DIRECTORY
AND TRY AGAIN"
3150 PRINT:PRINT:PRINT"PRESS ANY KEY TO CONTINUE"
3160 A$=INKEY$: IFA$="" THEN 3160 ELSE RESUME 3960
3170 PRINT "      **** BAD FILE NUMBER ****":PRINT:PRINT
      "THIS IS CAUSE BY NOT HAVING ENOUGH FILES"
3180 PRINT"YOU SHOULD START OVER (REBOOT) AND ANSWER THE
      STARTING QUESTION":PRINTCHR$(34);"HOW MANY FILES ?
      ";CHR$(34);" WITH AT LEAST ONE "
3190 PRINT"GOOD LUCK - PRESS ANY KEY TO CONTINUE "
3200 A$=INKEY$ : IF A$="" THEN 3200 ELSE CMD"I","DO START"
3210 ?
3220 ? *****
3230 ? * *
3240 ? * FIND THE MOTOR CONFIGURATION *
3250 ? * *
3260 ? *****
3270 ?
3280 DIM NN(4),NC(8),XC(8) 'DIMENSION VARIABLES SET IN
      THE SUBROUTINE
3290 DIM AM$(6),NF(4) 'DIMENSION LOCAL VARIABLES
3300 AM$(1)="CAROUSEL" :AM$(2)="CONVEYOR" :AM$(3)="LINEAR
      BASE"
3310 AM$(4)="WAIST" : AM$(5)="WRIST - ROTATIONAL":
      AM$(6)="NONE"
3320 NN(1)=1 : NN(2)=4 : NN(3)=3 : NN(4)=5
3330 NF(1)=2 : NF(2)=6 : NF(3)=7 : NF(4)=8
3340 ? **** PRINT INFORMATION OUT ****
3350 CLS : PRINT TAB(22)"MOTOR CONFIGURATION"
3360 PRINTTAB(20)"1 - GRASP or FINGERS":PRINTTAB(20)"2 -
      ";AM$(NN(1))
3370 PRINTTAB(20)"3 - WRIST - AZIMUTHAL":PRINTTAB(20)"4 -
      FOREARM":PRINTTAB(20)"5 - SHOULDER"
3380 FOR I=2 TO 4
3390 PRINT TAB(20)CHR$(52+I);" - ";AM$(NN(I))
3400 NEXT I
3410 PRINTTAB(6)** MAY CHANGE LOCATION OF MOTORS 2,6,7,8
      ONLY **"
3420 PRINTTAB(1);"MOTORS AND THERE POSSIBLE LOCATIONS ('#'
      - DEFAULT LOCATION)"
3430 PRINT TAB(6)AM$(1);" ('2',6,7,8)";TAB(38);AM$(2);" (2
      ONLY)"
3440 PRINT TAB(6)AM$(3);" (6,'7',8)";TAB(38);AM$(4);"

```

```

      ('6',7,8)"
3450 PRINT TAB(6)AM$(5);" (6,7,'8')";TAB(38);AM$(6);" - NO
      MOTOR ATTACHED"
3460 ' ***** ENTER THE LOCATION ***
3470 PRINT@960,"ENTER LOCATION TO BE CHANGED (2,6,7,or 8) or
      0 if O.K. ? ";
3480 A$=INKEY$;IF A$="" THEN 3480
3490 IF A$="0" THEN 3600
3500 IF A$="2" THEN N=1 : GOTO 3530
3510 IF A$<"6" OR A$>"8" THEN 3480
3520 N=VAL(A$)-4
3530 PRINT@960,CHR$(30); : PRINT@960,"ENTER THE MOTOR TO BE
      AT THE ";A$;" LOCATION ? ";:LL=20:GOSUB 1410
3540 FOR I=1 TO 6 ' CHECK TO FIND WHICH MOTOR SELECTED
3550 IF LEFT$(B$,3)=LEFT$(AM$(I),3) THEN
      NN(N)=I:PRINT@960,CHR$(30);:GOTO 3350
3560 NEXT I : PRINT@960,CHR$(30);:PRINT@960,B$;" IS NOT A
      MOTOR,TRY AGAIN";
3570 FOR I=1 TO 1500 : NEXT I:PRINT@960,CHR$(30); : GOTO
      3470
3580 ' ***** CHECK IF TWO OF MORE MOTORS OF THE SAME *****
3590 ' ***** EXCEPT 6(NONE) ALSO IF CORRECT CONNECTION TO 2
      **
3600 FOR J=1 TO 3
3610     TP=NN(J):IF TP=6 THEN 3650
3620     FOR I=J+1 TO 4
3630         IF TP=NN(I) THEN 3700
3640     NEXT I
3650 NEXT J
3660 FOR I=2 TO 4 : IF NN(I)=2 THEN 3690 : NEXT I
3670 IF NN(1)=1 OR NN(1)=2 OR NN(1)=6 THEN CLS : GOTO 3750
3680 PRINT @960,CHR$(30); : PRINT@960,"AN INCORRECT MOTOR IS
      CONNECTED TO MOTOR 2, PLEASE CORRECT "; : GOTO 3710
3690 PRINT @960,CHR$(30); : PRINT@960,"THE CONVEYOR CAN ONLY
      BE CONNECTED TO MOTOR 2, PLEASE CORRECT"; : GOTO 3710
3700 PRINT@960,CHR$(30);:PRINT@960,"TWO OR MORE LOCATIONS
      HAVE THE SAME MOTOR, PLEASE CORRECT";
3710 FOR C=1 TO 1500:NEXTC :PRINT@960,CHR$(30); :GOTO 3350
3720 ' *****
3730 ' * READ IN LIMITS FOR THIS MOTOR CONFIGURATION *
3740 ' *****
3750 READ NC(1),XC(1)
3760 FOR I=3 TO 5 : READ NC(I),XC(I) : NEXT I
3770 FOR I=1 TO 6 : FOR J=1 TO 4
3780     IF NN(J)=I THEN READ NC(NF(J)),XC(NF(J)) :IF I<6 THEN
      3800
3790     NEXT J :READ C,C
3800 NEXT I : RETURN
3810 ' *****
3820 ' * LIMITS FOR ALL POSSIBLE MOTORS *
3830 ' *****

```

```

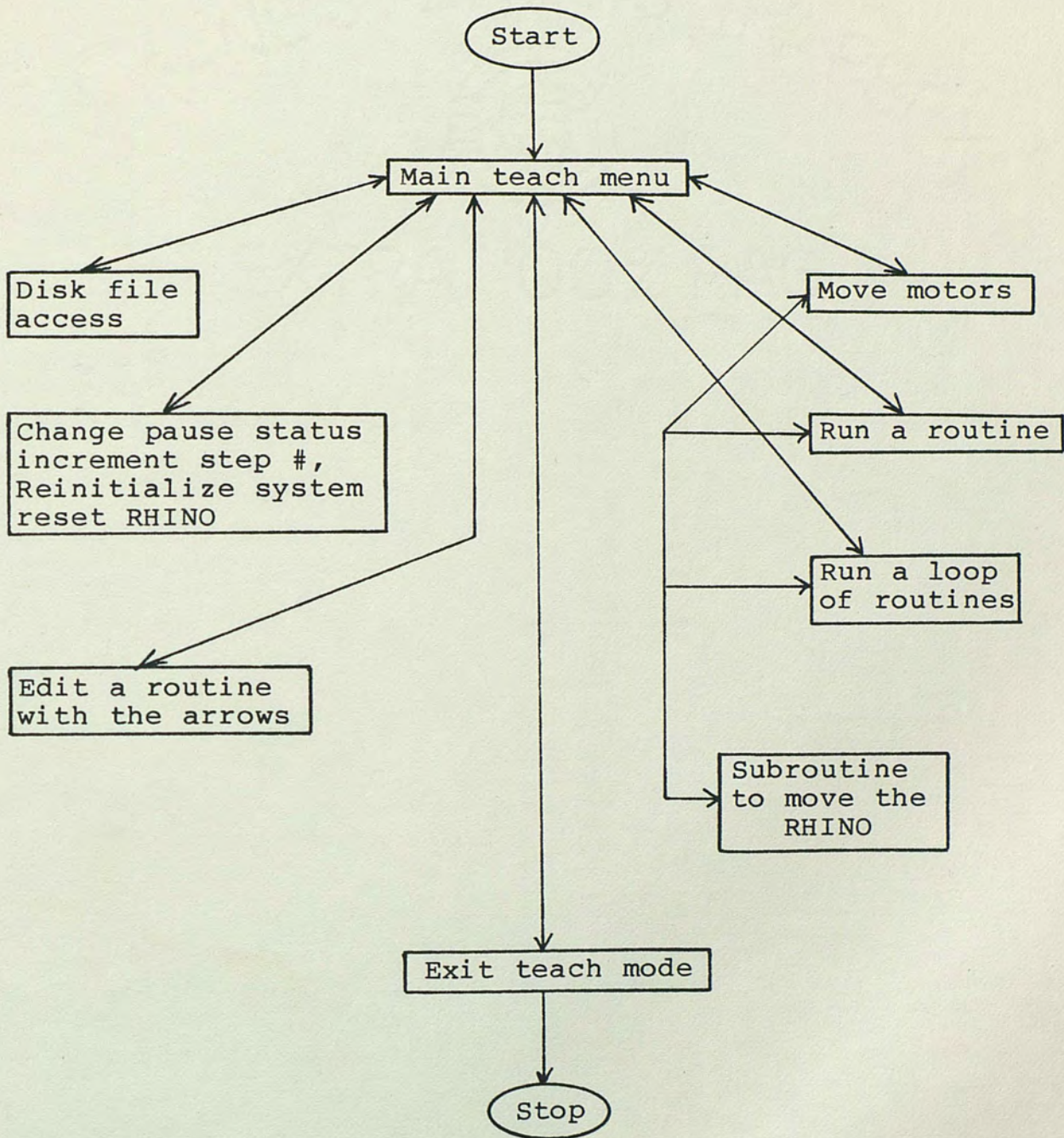
3840 DATA -45,75,-2000,1750           : 'MOTOR
      GRASP,WRIST-AZIMUTHAL (1,3)
3850 DATA -950,1200,-1000,650       : 'MOTOR FORARM,SHOULDER
      (4,5)
3860 DATA -7000,7000,-10000,10000   : 'MOTOR CAR,CON
3870 DATA 0,4750,-1700,700         : 'MOTOR LIN,WAI
3880 DATA -3000,3000,0,0           : 'MOTOR WRI,NONE
3890 DATA 0,0,0,0                   : 'MOTOR NONE REPEATED
3900 DATA 0,0,0,0
3910 '
3920 ' *****
3930 ' *   DISK DIR AND DELETING FILES   *
3940 ' *****
3950 '
3960 ON ERROR GOTO 2970 : CLS : CMD"D:0" : PRINT CHR$(15)
3970 PRINT"THE FILES WITH /RHN AT THE END ARE ROUTINE FILES
      AND":PRINTTAB(15)"/LOP AT THE END ARE LOOP FILES"
3980 PRINT@832,"DID YOU WANT TO DELETE (D) A FILE OR TRY
      SAVING (S) AGAIN ?"
3990 A$=INKEY$:IF A$="" THEN 3990
4000 IF A$="S" THEN 1270 ELSE IF A$<>"D" THEN 3990
4010 PRINT@960,"ENTER THE TYPE OF FILE TO BE DELETED; 1-/RHN
      2-/LOP ?";
4020 A$=INKEY$ : IF A$="" THEN 4020
4030 IF A$<"1" OR A$>"2" THEN 4020 ELSE DK=VAL(A$)
4040 PRINT @960,CHR$(30); : PRINT @960,"ENTER THE FILENAME
      OF THE FILE TO BE DELETED ? ";
4050 LL=8 : GOSUB 1410
4060 IF B$="" THEN PRINT@960,CHR$(30); : PRINT@960"MUST
      ENTER A FILENAME, TRY AGAIN"; : FOR I=1 TO 300 : NEXT C
      : PRINT @960,CHR$(30); : GOTO 3960
4070 FF#=B$+TY$(DK) : PRINT@960,CHR$(30); : PRINT@960,"ARE
      YOU SURE YOU WANT ";FF#;" TO BE DELETED ? ";
4080 A$=INKEY$ : IF A$="" THEN 4080
4090 POKE 16420,1 : POKE 16912,40 : OUT 236,8
4100 IF A$<>"Y" THEN PRINT CHR$(30); : GOTO 3960
4110 PRINT @960,CHR$(30); : PRINT @960,"KILLING
      ";CHR$(244);CHR$(245);CHR$(246);" " ;FF#
4120 KILL FF# : FOR C=1 TO 100 : NEXT C :PRINT@960,CHR$(30);
      : GOTO 3960

```

APPENDIX F

Flowchart for TEACH/BAS

FLOWCHART FOR TEACH/BAS



APPENDIX G

Program Listing of TEACH/BAS


```

100 '
110 ' *****
120 ' *
130 ' *   ROBOT TRAINING AID   MAY 1983   *
140 ' *                       BY         *
150 ' *                   DON WASHBURN   *
160 ' *
170 ' *****
180 '
190 ' NOTE : RHINO USES LABELS FOR MOTORS AS A TO H BUT, I
      CHANGED
200 '       THEM TO NUMBERS 1 TO 8 TO BE EASIER TO USE IN
      TEACH MODE.
210 '
220 ' *****
230 ' * INITIALIZE THE SYSTEM *
240 ' *****
250 '
260 CLEAR 2500
270 DEFINT A-Z
280 DEFUSRO=85      'SEND CHAR ON RS-232
290 DEFUSR1=80     'GET CHAR OFF OF RS-232
300 DIM N(100,8), N2(100,8), MC(8),DC!(8), TT(8),M(8),
      S(8),TY$(2),T2$(2)
310 DIM PP$(2),RN$(20),RN(20),F$(2),M$(2),NC(8),XC(8),NN(4)
320 ' *****
330 ' *   LOAD THE VARIABLE SET IN STARTUP FOR RHINO/SAV *
340 ' *****
350 E5=3 : ON ERROR GOTO 2520
360 OPEN "I",1,"RHINO/SAV"
370 INPUT #1,FF$
380 FOR I=1 TO 8 : INPUT #1,NC(I),XC(I) : NEXT I
390 FOR I=1 TO 4 : INPUT #1,NN(I)      : NEXT I
400 CLOSE
410 ON ERROR GOTO 0
420 '
430 ' *****
440 ' *   SET UP VARIABLES AND DEFAULT VALUES *
450 ' *****
460 '
470 PP$(0)="ON " : PP$(1)="OFF" : M$(1)=" " : M$(2)=" " :
      TY$(1)="/RHN" : TY$(2)="/LOP" : T2$(1)="ROUTINE'S" :
      T2$(2)="LOOP'S" : TL=0 : F$(1)=FF$ : F$(2)=FF$
480 M=1 : B1=100 : B2=10 : YY=0 : PP=0 : LG=0 : FC=0 : FY=0 : Y5=0
      'HIGHEST STEP IN ARRAY AT PRESENT, SET THE TWO
      INCREMENTS, AND SET FLAG VARIABLES
490 FOR I=1 TO 8 : TT(I)=0 : NEXT I
500 FOR J=0 TO 1 : FOR I=1 TO 8
510 N(J,I)=0      ' CLEAR ARRAY FOR STEPS TO BE ADDED AND
520 NEXT I : NEXT J ' MAKE THE START LOCATION THE RESET (ALL

```

```

O)
530 '
540 ' *****
550 ' * *
560 ' * SCREEN DISPLAY *
570 ' * *
580 ' *****
590 '
600 CLS ' PRINT OUT INSTRUCTIONS FOR TEACH MODE ON SCREEN
610 PRINTTAB(24)"RHINO TEACH MODE":PRINTTAB(22)"ROUTINE
    TITLE ";F$(1)
620 PRINT") - RUN A LOOP OF ROUTINE";:PRINTTAB(35);"CLEAR -
    SETS START LOCATION"
630 PRINT"* - REINITIALIZE THE SYSTEM"
640 PRINT"P - CHANGE PAUSE STATUS":PRINT" - RESET THE
    RHINO (SHIFT @)"
650 PRINTCHR$(123)" - ERASE THIS STEP AND BACK ONE (LEFT
    ARROW)":PRINTCHR$(125)" - ERASE THIS STEP (RIGHT ARROW)"
660 PRINT"L - CHANGE INCREMENT VALUES":PRINT"O - INCREMENT
    STEP NUMBER (TO RECORD A STEP)":PRINT"+ - ADD A LONG
    MOVE TO ONE MOTOR"
670 PRINT"> - RUN THIS ROUTINE":PRINT"? - EXIT TEACH MODE"
675 PRINT "= - DISK FILE ACCESS"
680 PRINT@234,"PAUSE STATUS - ";:PRINT@249,PP$(PP);
690 FOR DX=1 TO 8
700 PRINT@241+DX*64,CHR$(48+DX);" - ";
710 NEXT DX
720 PRINT@810,"STEP NUMBER - ";:PRINT@825,M;
730 PRINT@869,"HIGH - ";:PRINT@886,"LOW - ";: PRINT@875,
    STR$(B1);: PRINT@891, STR$(B2);
740 IF FC=1 OR M>1 THEN 1490
750 FOR I=1 TO 50
760 PRINT @960,"MOVE TO THE STARTING LOCATION AND PRESS
    CLEAR (DEFAULT-ALL 0)";
770 FOR C=1 TO 20 :NEXT C : PRINTCHR$(29);CHR$(30); : NEXT I
780 GOTO 1490
790 '
800 ' *****
810 ' * *
820 ' * CHECK TO SEE IF A MOTOR IS MOVING *
830 ' * OVER OR UNDER IT'S LIMITS *
840 ' * AND COUNT THE STEPS MOVED *
850 ' * *
860 ' *****
870 '
880 ZT=NC(T)-TT(T) : IF ZT<0 THEN ZT=TT(T)-XC(T) : GOTO 930
    ' DETERMINE IF OVER OR UNDER THE LIMIT
890 PRINT @960,"MOTOR ";CHR$(48+T);" IS UNDER IT'S LIMIT BY
    ";ZT;" STEPS, PLEASE BE CAREFUL ";
900 FOR C=1 TO 1000 :NEXT C ' TIME DELAY
910 PRINT@960,CHR$(30); : TT(T)=TT(T)-CT ' BECAUSE

```

```

OVER OR UNDER ADD COUNT BACK IN
920 IF LG=0 THEN 1600 ELSE LG=2 :RETURN
930 PRINT @960,"MOTOR ";CHR$(48+T); " IS OVER IT'S LIMIT BY
";ZT;" STEPS, PLEASE BE CAREFUL";
940 GOTO 900
950 ' *****
960 ' * CHECK MOTOR 3+4 FOR LIMITS, SINCE THEY CHANGE *
970 ' *****
980 '
990 ' ***** MOVE 4 ,CHECK 5 IF + OR - *****
1000 IF TT(5)>0 THEN 1070
1010 '
1020 ' ***** 5 IS -, SO D'S LIMITS = -950+.8(5) & 1200+(5)
***
1030 IF TT(T)<0 THEN CP=NC(T)+.8*TT(5) : IF TT(T)>=CP THEN
1110 ELSE ZT=CP-TT(T) : GOTO 890
1040 IF TT(T)<=XC(T)+TT(5) THEN 1110 ELSE
ZT=TT(T)-(XC(T)+TT(5)) :GOTO 930
1050 '
1060 ' ***** 5 IS +, SO D'S LIMITS = -950+1.5(5) & 1200
***
1070 IF TT(T)<0 THEN CP=NC(T)+1.5*TT(5) :IF TT(T)>=CP THEN
1110 ELSE ZT=CP-TT(T): GOTO 890
1080 IF TT(T)>XC(T) THEN ZT=TT(T)-XC(T) : GOTO 930
1090 '
1100 ' ***** MOVE 4 SO CHECK IF 3 LIMITS CHANGED **
1110 IF TT(T)<0 THEN 1140
1120 IF TT(3)<0 THEN CP=NC(3)+1.7*TT(T) : IF TT(3)>=CP THEN
1430 ELSE 1160
1130 IF TT(3)<=XC(3)+TT(T) THEN 1430 ELSE 1160
1140 IF TT(3)<0 THEN CP=NC(3)+TT(T): IF TT(3)>=CP THEN 1430
ELSE 1160
1150 IF TT(3)<=XC(3)+2*TT(T) THEN 1430 ELSE 1160
1160 PRINT@960,"MOTOR 4 IS MOVING INTO MOTOR 3 LIMIT HAVE TO
MOVE 3 FIRST"; : GOTO 900
1170 '
1180 ' ***** MOVE 3, CHECK 4 IF + OR - ****
1190 IF TT(4)<0 THEN 1260
1200 '
1210 ' ***** 4 IS + SO 3'S LIMITS = -2000+1.7(4) & 1750(4)D
***
1220 IF TT(T)<0 THEN CP=NC(T)+1.7*TT(4) : IF TT(T)>=CP THEN
1430 ELSE ZT=CP-TT(T) : GOTO 890
1230 IF TT(T)<=XC(T)+TT(4) THEN 1430 ELSE
ZT=TT(T)-(XC(T)+TT(4)) : GOTO 930
1240 '
1250 ' ***** 4 IS -, SO 3'S LIMITS = -2000+(4) & 1750+2(4)
***
1260 IF TT(T)<0 THEN CP=NC(T)+TT(4): IF TT(T)>=CP THEN 1430
ELSE ZT=CP-TT(T) : GOTO 890
1270 IF TT(T)<=XC(T)+2*TT(4) THEN 1430 ELSE

```

```

      ZT=TT(T)-(XC(T)+2*TT(4)) : GOTO 930
1280 '
1290 ' ***** MOVE 5 BUT CHECK IF 4 LIMIT'S CHANGED ***
1300 IF TT(T)>0 THEN 1330
1310 IF TT(4)<0 THEN CP=NC(4)+.8*TT(T) : IF TT(4)>=CP THEN
1430 ELSE 1350
1320 IF TT(4)<=XC(4)+TT(T) THEN 1430 ELSE 1350
1330 IF TT(4)<0 THEN CP=NC(4)+1.5*TT(T) : IF TT(4)>=CP THEN
1430 ELSE 1350
1340 GOTO 1430
1350 PRINT @960,"MOTOR 5 IS MOVING INTO MOTOR 4 LIMIT HAVE
      TO MOVE 4 FIRST"; : GOTO 900
1360 ' *****
1370 ' *      MOVE MOTOR PER KEY ENTERED      *
1380 ' *****
1390 TT(T)=TT(T)+CT:IF T=4 THEN 990
1400 IF T=3 THEN 1190
1410 IF TT(T)<NC(T) OR TT(T)>XC(T) THEN 880 ' CHECK IF OVER
      LIMITS
1420 IF T=5 THEN 1300
1430 IF LG=1 THEN RETURN ELSE N(M,T)=N(M,T)+CT:IF CT<0 THEN
      SG=45 ELSE SG=43'AND COUNTS THEN DETERMINE SIGN OF MOVE
1440 CT=ABS(CT):MT=T+64:GOSUB 4980 ' MOVE THE MOTOR
1450 GOSUB 5110 : IF E>0 THEN 1450 ' IF DONE ?
1460 ' *****
1470 ' * PRINT OUT # OF STEPS FOR EACH MOTOR *
1480 ' *****
1490 FOR DX=1 TO 8 'PRINT OUT # OF STEPS FOR EACH JOINT
1500 PRINT@245+DX*64,TT(DX);"  ";
1510 NEXT DX
1520 '
1530 ' *****
1540 ' *
1550 ' * GET INPUT FOR KEYBOARD AND TEST TO SEE *
1560 ' * WERE TO GO OR WHICH MOTOR TO MOVE *
1570 ' *
1580 ' *****
1590 '
1600 A$=INKEY$ : IF A$="" THEN 1600
1610 IF A$="L" THEN GOSUB 2260 :GOTO 1600 ' CHECK IF
      SPECIAL COMMAND
1620 IF A$="." THEN YY=1 : GOSUB 4710 : GOTO 1490
1630 IF A$="*" THEN 2760
1640 IF A$="P" THEN GOSUB 2680 : GOTO 1600
1650 IF A$=")" THEN Y5=0 :GOSUB 6690 : GOTO 600
1660 IF A$="?" THEN 3120
1670 IF A$="O" THEN 2150
1680 IF A$=">" THEN GOSUB 3700 : PRINT@825,M; : GOTO 1490
1690 IF A$="=" THEN GOSUB 5400 : GOTO 600
1700 IF A$="+" THEN 3290
1710 IF ASC(A$)=31 THEN GOSUB 2980 : GOTO 1490

```

```

1720 IF ASC(A$)=8 THEN GOSUB 3560 : GOTO 1490
1730 IF ASC(A$)=9 THEN GOSUB 3460 : GOTO 1490
1740 ' *****
1750 ' * CHECK FOR MOTOR KEY NOW *
1760 ' *****
1770 IF A$="1" THEN CT=5 :T=1:GOTO 1390 'FIND WHICH MOTOR
    TO MOVE (T)
1780 IF A$="Q" THEN CT=1 :T=1:GOTO 1390 'AND THE COUNTS
    (CT)
1790 IF A$="A" THEN CT=-1 :T=1:GOTO 1390
1800 IF A$="Z" THEN CT=-5 :T=1:GOTO 1390
1810 IF A$="2" THEN CT=B1 :T=2:GOTO 1390
1820 IF A$="W" THEN CT=B2 :T=2:GOTO 1390
1830 IF A$="S" THEN CT=-B2:T=2:GOTO 1390
1840 IF A$="X" THEN CT=-B1:T=2:GOTO 1390
1850 IF A$="3" THEN CT=B1 :T=3:GOTO 1390
1860 IF A$="E" THEN CT=B2 :T=3:GOTO 1390
1870 IF A$="D" THEN CT=-B2:T=3:GOTO 1390
1880 IF A$="C" THEN CT=-B1:T=3:GOTO 1390
1890 IF A$="4" THEN CT=B1 :T=4:GOTO 1390
1900 IF A$="R" THEN CT=B2 :T=4:GOTO 1390
1910 IF A$="F" THEN CT=-B2:T=4:GOTO 1390
1920 IF A$="V" THEN CT=-B1:T=4:GOTO 1390
1930 IF A$="5" THEN CT=B1 :T=5:GOTO 1390
1940 IF A$="T" THEN CT=B2 :T=5:GOTO 1390
1950 IF A$="G" THEN CT=-B2:T=5:GOTO 1390
1960 IF A$="B" THEN CT=-B1:T=5:GOTO 1390
1970 IF A$="6" THEN CT=B1 :T=6:GOTO 1390
1980 IF A$="Y" THEN CT=B2 :T=6:GOTO 1390
1990 IF A$="H" THEN CT=-B2:T=6:GOTO 1390
2000 IF A$="N" THEN CT=-B1:T=6:GOTO 1390
2010 IF A$="7" THEN CT=B1 :T=7:GOTO 1390
2020 IF A$="U" THEN CT=B2 :T=7:GOTO 1390
2030 IF A$="J" THEN CT=-B2:T=7:GOTO 1390
2040 IF A$="M" THEN CT=-B1:T=7:GOTO 1390
2050 IF A$="8" THEN CT=B1 :T=8:GOTO 1390
2060 IF A$="I" THEN CT=B2 :T=8:GOTO 1390
2070 IF A$="K" THEN CT=-B2:T=8:GOTO 1390
2080 IF A$="," THEN CT=-B1:T=8:GOTO 1390
2090 GOTO 1600 'NOT A MOTOR
2100 '
2110 ' *****
2120 ' * INCREMENT THE STEP COUNTER AND CHECK IF OVER *
2130 ' * THE LIMIT, AND PRINT WARNING IF CLOSE TO LIMIT *
2140 ' *****
2150 M=M+1 : IF M>100 THEN M=100 : PRINT@960,"YOU EXCEED
    THE MAXIMUM STEP SIZE OF 100";:GOTO 2220 ' OVER THE
    LIMIT OF ROUTINE ARRAY
2160 FOR CI=1 TO 8
2170 N(M,CI)=0
2180 NEXT CI

```

```

2190 PRINT @825,M;
2200 IF M<85 THEN 1600
2210 DF=100-M:PRINT@960,"ONLY ";DF;" STEPS LEFT, SO FINISH
UP";
2220 FOR C=1 TO 1500:NEXT C:PRINT@960,CHR$(30);:GOTO1580
2230 ' *****
2240 ' * ENTER NEW INCREMENT VALUES *
2250 ' *****
2260 PRINT@960,"ENTER THE LOW INCREMENT VALUE ? ";
2270 GOSUB 2560 :IF VV=0 THEN PRINT@860,CHR$(30);:GOTO 2260
2280 PRINT@960,CHR$(30); : IF B#<>"" THEN B2=VAL(B#)
2290 PRINT @960,"ENTER THE HIGH INCREMENT VALUE ? ";
2300 GOSUB 2560:IF VV=0 THEN PRINT@960,CHR$(30);:GOTO 2290
2310 PRINT @960,CHR$(30); : IF B#<>"" THEN B1=VAL(B#)
2320 IF B1>B2 THEN 2360
2330 PRINT @960,"HIGH VALUE MUST BE LARGER THAN LOW VALUE,
TRY AGAIN";
2340 FOR DD=1 TO 1000 :NEXT DD
2350 PRINT @ 960,CHR$(30); : GOTO 2260
2360 IF B1> 127 THEN PRINT@960,"THE INCREMENT VALUES MUST BE
LESS THAN 127, TRY AGAIN "; : GOTO 2340
2370 IF B2<1 THEN PRINT@960,"THE INCREMENT VALUES MUST BE
POSTIVE, TRY AGAIN";:GOTO 2340
2380 PRINT@875,STR$(B1);" ";:PRINT@891,STR$(B2);" ";
2390 RETURN
2400 ' *****
2410 ' * INPUT ARRAY OF LENGTH LL *
2420 ' *****
2430 B#=""
2440 A#=INKEY$ : IF A#="" THEN 2440
2450 IF ASC(A#)=13 THEN RETURN
2460 IF A#="," THEN PRINT " NO COMMAS"; : FOR C=1 TO 2000 :
NEXT C : PRINT " "; :
GOTO 2440
2470 IF ASC(A#)=8 THEN L1=LEN(B#):IF L1=0 THEN 2430 ELSE
PRINT A#;:LL=LL+1:IF L1=1 THEN 2430 ELSE
B#=LEFT$(B#,L1-1) : GOTO 2440
2480 IF ASC(A#)>127 OR ASC(A#)<32 THEN 2440
2490 B#=B#+A# : LL=LL-1 : PRINT A#;
2500 IF LL=0 THEN RETURN ELSE 2440
2510 ' *****
2520 ' * INPUT A NUMBER SET FLAG *
2530 ' * VV=0 - BAD NUMBER *
2540 ' * =1 - O.K. *
2550 ' *****
2560 B#="" : BB=0
2570 A#=INKEY$ : IF A#="" THEN 2570
2580 IF ASC(A#)>=48 AND ASC(A#)<=57 THEN B#=B#+A# : BB=1 :
PRINT A#;:GOTO 2570
2590 IF ASC(A#)=13 THEN VV=1 :RETURN
2600 IF BB=0 AND A#="-" THEN B#="-" : BB=1 :PRINT B#; :GOTO

```

```

2570
2610 IF ASC(A#)=8 THEN L1=LEN(B#) : IF L1=0 THEN 2570 ELSE
PRINT A#;:IF L1=1 THEN 2560 ELSE B#=LEFT$(B#,L1-1):GOTO
2570
2620 PRINT @992," BAD INPUT, TRY AGAIN ";
2630 FOR DD=1 TO 400 : NEXT DD
2640 PRINT@960,CHR$(30);:VV=0:RETURN
2650 ' *****
2660 ' * CHANGE PAUSE STATUS *
2670 ' *****
2680 IF PP=0 THEN PP=1 ELSE PP=0
2690 PRINT @249,PP$(PP); : RETURN
2700 '
2710 '
2720 '*****
2730 '* RESET SYSTEM BY DELETING CURRENT ROUTINE OR RERUN *
2740 '******
2750 '
2760 PRINT @960,"1-DELETE CURRENT ROUTINE,2-RESET SYSTEM AND
MOTORS,3-CONTINUE?";
2770 A#=INKEY$ : IF A#="" THEN 2770 ELSE PRINT
@960,CHR$(30);
2780 IF A#="1" THEN 2870
2790 IF A#<>"2" THEN 1600
2800 ' ***** RESET SYSTEM BY RE-RUNING THE PROGRAM *****
2810 PRINT @960,"SURE ABOUT RESETTING SYSTEM, WILL LOSE
CURRENT ROUTINE + LOOP?";
2820 A#=INKEY$ : IF A#="" THEN 2820 ELSE PRINT
@960,CHR$(30);
2830 IF A#<>"Y" THEN 1600
2840 GOSUB 4750 : MT=65:SG=43:CT=75 : GOSUB 4930
2850 CMD"I","DO START"
2860 ' ***** DELETING CURRENT ROUTINE *****
2870 PRINT @960,"SURE ABOUT DELETING CURRENT ROUTINE, DID
YOU SAVE IT FIRST?";
2880 A#=INKEY$ : IF A#="" THEN 2880 ELSE PRINT
@960,CHR$(30);
2890 IF A#<>"Y" THEN 1600
2900 FOR I=1 TO 8
2910 N(1,I)=TT(I) : N(0,I)=0
2920 NEXT I
2930 M=1 : FC=0 : GOTO 610 'RESET FLAG VARIABLES
2940 '
2950 ' *****
2960 ' * CLEAR - SET START LOCATION *
2970 ' *****
2980 IF M=1 THEN 3010 'CHECK IF AT START
2990 PRINT@960,"CAN ONLY SET START POSITION AT THE START OF
THE ROUTINE";
3000 FOR C=1 TO 1000 : NEXT C : PRINT @960,CHR$(30);:RETURN
3010 IF FC=0 THEN 3050 ' CHECK IF ALREADY SET

```

```

3020 PRINT @960,"START POSITION ALREADY SET, DO YOU WISH TO
CHANGE IT ? ";
3030 A#=INKEY# : IF A#="" THEN 3030
3040 IF A#<>"Y" THEN PRINT@960,CHR$(30); : RETURN
3050 PRINT @960,CHR$(30); : PRINT @960,"THIS IS NOW THE
START LOCATION - CONTINUE WITH ROUTINE";
3060 FC=1 : FOR I=1 TO 8
3070 N(0,I)=TT(I) : N(1,I)=0 ' SET START POSITION AND
ERASE STEP 1
3080 NEXT I :FOR I=1 TO 1000 : NEXT I : PRINT @960,CHR$(30);
: RETURN
3090 ' *****
3100 ' * END THE ROUTINE AND SETUP FOR RESTART *
3110 ' *****
3120 PRINT@ 960,"ARE YOU SURE YOU WANT TO END, HAVE YOU
SAVED YOUR ROUTINE ?";
3130 A#=INKEY# : IF A#="" THEN 3130
3140 IF A#<>"Y" THEN PRINT@960,CHR$(30);: GOTO 1600
3150 PRINT@960,CHR$(30);:GOSUB 4750
3160 CT=75:MT=65:SG=43:GOSUB 4980
3170 CLS:PRINT:PRINTCHR$(23):PRINTTAB(4);"MUST TURN BOTH
SWITCHES":PRINT:PRINT TAB(2);"ON THE POWER SUPPLY OFF
!!"
3180 PRINT@768,"PRESS ANY KEY WHEN THEY ARE OFF"
3190 FOR KK=1 TO 25 : A#=INKEY#:IF A#<>" " THEN 3210 : NEXT
KK
3200 PRINT @768,CHR$(30); : FOR KK=1 TO 10 : NEXT KK : GOTO
3180
3210 CLS : 'CMD"B", "ON" ' ENABLE BREAK KEY
3220 CMD"S" ' THE END
3230 '
3240 ' *****
3250 ' * ADD A LONG MOVE TO *
3260 ' * ONE MOTOR *
3270 ' *****
3280 '
3290 PRINT @960,"PRESS THE NUMBER KEY OF THE MOTOR TO BE
USED (0 TO RETURN) ? ";
3300 A#=INKEY# : IF A#="" THEN 3300
3310 IF A#<"1" OR A#>"8" THEN IF A#="0" THEN PRINT
@960,CHR$(30); : GOTO 1490 ELSE 3300
3320 MT=VAL(A#)
3330 PRINT@960,CHR$(30); : PRINT @960,"ENTER THE # OF STEPS
TO BE TAKEN BY MOTOR ";MT;" ? ";
3340 GOSUB 2560 : IF VV=0 THEN 3330
3350 T=MT:CT=VAL(B#)
3360 IF CT=0 THEN PRINT @960,CHR$(30); : GOTO 1490
3370 LG=1 : GOSUB 1390
3380 IF LG=2 THEN PRINT@960,CHR$(30);:LG=0 : GOTO 1600
3390 FOR I=1 TO 8 : MC(I)=0 : NEXT I
3400 MC(MT)=CT : N(M,T)=N(M,T)+CT : GOSUB 4240

```



```

3410 LG=0:PRINT @960,CHR$(30);:GOTO 1490
3420 '
3430 ' *****
3440 ' *      RIGHT ARROW KEY - DELETE THIS STEP      *
3450 ' *****
3460 PRINT @960,"ARE YOU SURE, YOU WANT TO DELETE THIS STEP
(Y or N) ?";
3470 A$=INKEY$ : IF A$="" THEN 3470 ELSE PRINT
@960,CHR$(30); : IF A$<>"Y" THEN RETURN
3480 FOR I=1 TO 8
3490 MC(I)=-N(M,I):TT(I)=TT(I)-N(M,I) : N(M,I)=0
3500 NEXT I
3510 GOSUB 4240 : RETURN
3520 ' *****
3530 ' * LEFT ARROW KEY - DELETE THIS STEP *
3540 ' * AND ONE BACK FROM THIS ONE *
3550 ' *****
3560 PRINT @960,"ARE YOU SURE, YOU WANT TO DELETE THIS STEP
AND ONE BACK (Y,N) ?";
3570 A$=INKEY$ : IF A$="" THEN 3570 ELSE PRINT
@960,CHR$(30); : IF A$<>"Y" THEN RETURN
3580 IF M>1 THEN 3620
3590 PRINT @960,"AT STEP ONE, CAN NOT GO BACKWARDS !";
3600 FOR I=1 TO 1000 : NEXT I
3610 PRINT @960,CHR$(30); : RETURN
3620 GOSUB 3480 : M=M-1 : GOSUB 3480 : PRINT @825,M;" " :
RETURN
3630 '
3640 ' *****
3650 ' * *
3660 ' * RUN A ROUTINE SUBROUTINE *
3670 ' * *
3680 ' *****
3690 '
3700 PRINT@960,"ENTER # OF REPEATS FOR THIS ROUTINE (0 TO
RETURN) ? " : GOSUB2560
3710 PRINT@960,CHR$(30); : IF VV=0 THEN 3700 ELSE
RR=ABS(VAL(B$))
3720 IF RR=0 THEN RETURN
3730 FOR LR=1 TO RR
3740 IF PP=0 THEN PRINT @960,CHR$(30); : FOR DX=1 TO 8
:PRINT@245+DX*64,TT(DX);" " :NEXT DX
3750 IF PP=0 THEN GOSUB 4870 ELSE GOSUB 4880 'RESET THE
RHINO TO THE STARTING LOCATION
3760 LP=1 : IF PP=0 THEN PRINT @984,"RUNNING ";F$;"
ROUTINE"; : PRINT @825,LP;" " :
3770 FOR LP=1 TO M ' LOOP THOUGH ALL MOVES
3780 TP=0 ' SET FLAG
3790 FOR I=1 TO 8
3800 MC(I)=N(LP,I):TT(I)=TT(I)+MC(I) ' STORE FOR MOVE
SUBROUTINE

```

```

3810     IF MC(I)<>0 THEN TP=1  ' IF THIS MOTOR MOVES SET
      FLAG TO 1
3820     NEXT I
3830     IF TP=1 THEN GOSUB 4240  ' CALL MOVE SUBROUTINE IF
      AT LEAST ONE MOTOR MOVED
3840     IF FY=1 THEN 3870
3850     FOR DX=1 TO 8 : PRINT @245+DX*64,TT(DX);"  " ;:NEXT
      DX
3860     PRINT @825,LP;
3870     IF PP=1 THEN 3960
3880     PRINT @960, "THE";LP;"STEP OUT OF";M;"* P-PAUSE
      STATUS, S-STOP, C-CONTINUE ?";
3890     A#=INKEY$; IF A#="" THEN 3890
3900     IF A#="P" THEN GOSUB 2680 : PRINT @960,CHR$(30); : GOTO
      3960
3910     IF A#<>"S" THEN 3960
3920     PRINT@960,CHR$(30);:PRINT @960,"ARE YOU SURE, THE REST
      OF THE STEPS WILL BE ERASED ? ";
3930     A#=INKEY$;IF A#="" THEN 3930
3940     PRINT @960,CHR$(30);
3950     IF A#="Y" THEN M=LP;V=1 :RETURN
3960     NEXT LP
3970     IF RR-LR=<1 OR PP=1 THEN 4030
3980     PRINT@960,CHR$(30); : PRINT@960,LR;"REPEATS OUT
      OF";RR;"* P-PAUSE STATUS, S-STOP, C-CONTINUE ?";
3990     A#=INKEY$;IF A#="" THEN 3990
4000     PRINT@960,CHR$(30);
4010     IF A#="P" THEN GOSUB 2680 : PRINT@960,CHR$(30); : GOTO
      4030
4020     IF A#="S" THEN 4100
4030     NEXT LR
4040     IF PP=0 THEN PRINT@960,CHR$(30);
4050     V=0 : RETURN
4060     '
4070     ' NOTE ERROR FLAG V=0 - WHEN O.K., V=1 - STOP IN MIDDLE
      OF ROUTINE
4080     '
      V=2 - STOP AT END OF ROUTINE
4090     ' ***** IF STOP THE ROUTINE DURING REPEATING *****
4100     PRINT@960,"ARE YOU SURE YOU WANT TO STOP REPEATING NOW
      ? ";
4110     A#=INKEY$ : IF A#="" THEN 4110
4120     PRINT @960,CHR$(30);
4130     IF A#<>"Y" THEN 4030 ELSE V=2 : RETURN
4140     '
4150     ' *****
4160     ' * MOVE THE RHIND *
4170     ' *****
4180     '
4190     ' VARIABLE DEFINITION          MC - ALL MOTOR COUNTS
4200     ' MAX - MAX # OF COUNTS      MM - MOTOR LETTER WITH MAX
4210     ' BJ - # OF MOTORS THAT MOVES F - FRACTION TO BE MOVED

```

```

4220 ' DC - MOTOR COUNTS (AND FRACTION) OF EACH MOTOR THAT
MOVES
4230 ' S - SIGN ARRAY          M - MOTOR ARRAY
4240 BJ=0:MAX=0:MM=0:F=1 ' INITIALIZE THE VARIABLES
4250 FOR I=1 TO 8
4260   J=I-BJ          ' FIND # OF MOTOR THAT MOVES
4270   IF MC(I)=0 THEN BJ=BJ+1:GOTO 4310 ' DOES NOT MOVE
SO NEXT MOTOR
4280   MC(J)=MC(I):M(J)=I+64:IF MC(J)<0 THEN S(J)=45 ELSE
S(J)=43
4290   MC(J)=ABS(MC(J)):IF MC(J)>MAX THEN MAX=MC(J):MM=J
4300   DC!(J)=MC(J)
4310 NEXT I
4320 IF BJ=8 THEN RETURN
4330 N=8-BJ
4340 IF MAX<=127 THEN 4460 ' SHORT MOVE
4350 TM=127 : SM=MAX : S1=0
4360 FOR I=1 TO N ' FRACTION TO BE MOVED EACH TIME
4370   DC!(I)=(TM*DC!(I))/MAX:IF MC(I)>.9*SM THEN S1=S1+1
4380 NEXT I
4390 IF S1>1 THEN 4570
4400 ' ***** THE LOOP TO MOVE THE MOTORS *****
4410 FOR J=1 TO N
4420   CT=INT(DC!(J)*F!+.5)
:MT=M(J):SG=S(J):MC(J)=MC(J)-CT:GOSUB 4980
4430 NEXT J
4440 MT=M(MM):GOSUB 5110 ' CHECK THE LONGEST MOVE
4450 F!=1-E/127:IF E+MC(MM)>127 THEN 4410 ' IF ALMOST DONE
4460 FOR J=1 TO N ' FINISH THE MOVE
4470   CT=MC(J):IF CT<0 THEN 4530 ELSE MT=M(J):SG=S(J):GOSUB
4980
4480 NEXT J
4490 MT=MM ' CHECK THE LONGEST MOVE IF DONE
4500 GOSUB 5110
4510 IF E=0 THEN RETURN ELSE 4500 ' IF SO , RETURN
4520 ' THE END OF SUBROUTINE
4530 MT=M(J):IF S(J)=43 THEN SG=45 ELSE SG=43
4540 GOSUB 4980 :GOTO4480 ' OVER MOVED SO CORRECT
4550 ' **** LOOP TO MOVE THE MOTORS WITH TWO OR MORE
4560 ' **** MOTORS WITH THE SAME COUNTS
4570 FOR J=1 TO N
4580   CT=INT(DC!(J)*F!+.5)
:MT=M(J):SG=S(J):MC(J)=MC(J)-CT:GOSUB 4980
4590 NEXT J
4600 EE=0
4610 FOR J=1 TO N
4620 MT=M(J):GOSUB 5110 :IF E>EE THEN EE=E:MM=J
4630 NEXT J
4640 F!=1-EE/127:IF EE+MC(MM)>127 THEN 4570 ' IF ALMOST
DONE
4650 GOTO 4460

```

```

4660 '
4670 ' *****
4680 ' * IF THE POSTION IS KNOWN USE TT ARRAY FOR RESET *
4690 ' *****
4700 '
4710 PRINT @960,"RESETTING RHIND, 1-START LOCATION,2-SETUP
      (ALL 0's),3-RETURN ?";
4720 A$=INKEY$ : IF A$="" THEN 4720 ELSE PRINT
      @960,CHR$(30);
4730 IF A$="1" THEN GOSUB 4870 : GOTO 4810
4740 IF A$<>"2" THEN RETURN
4750 PRINT@983,"RESETTING THE RHIND";
4760 FOR I=1 TO 8
4770   MC(I)=-TT(I) : IF YY=1 THEN N(M,I)=N(M,I)-TT(I)
4780   TT(I)=0
4790 NEXT I
4800 YY=0 : GOSUB 4240 : FOR I=1 TO 200 :NEXTI
4810 PRINT @978,CHR$(30); : RETURN
4820 '
4830 ' *****
4840 ' *   RESET TO START LOCATION FROM PRESENT LOCATION *
4850 ' *****
4860 '
4870 PRINT @978,"RESETTING TO START LOCATION";
4880 FOR I=1 TO 8
4890 MC(I)=N(O,I)-TT(I) : TT(I)=N(O,I)
4900 IF YY=1 THEN N(M,I)=N(M,I)+MC(I)
4910 NEXT I : GOSUB 4240 : FOR I=1 TO 200 :NEXT I
4920 YY=0 : IF PP=1 THEN RETURN ELSE PRINT @978,CHR$(30); :
      RETURN
4930 '
4940 ' *****
4950 ' *   SEND A MOVE TO A MOTOR *
4960 ' *****
4970 '
4980 POKE 16880,MT : X=USR0(0)   ' OUTPUT THE MOTOR AND SIGN
      OF MOVE
4990 POKE 16880,SG : X=USR0(0)
5000 CC=CT:IF CC>=100 THEN POKE 16880,49 : X=USR0(0)
      :CC=CC-100   ' CHECK IF OVER 100, IF SO OUTPUT A ONE
5010 W=INT(CC/10) : CC=CC-W*10   ' FIND TENS + ONES
5020 POKE 16880,W+48 : X=USR0(0) ' THEN OUTPUT THEM
5030 POKE 16880,CC+48 : X=USR0(0)
5040 POKE 16880,13 : X=USR0(0)   ' OUTPUT A CR TO END THE
      MESSAGE
5050 RETURN
5060 '
5070 ' *****
5080 ' *   GET THE NUMBER OF COUNTS LEFT *
5090 ' *****
5100 '

```

```

5110 POKE 16880,MT : X=USR0(0) ' OUTPUT THE MOTOR AND A
      '?'
5120 POKE 16880,63 : X=USR0(0)
5130 X=USR1(0) ' GET RESPONSE
5140 E=PEEK(16872)-32 ' A RHINO FIX
5150 IF E<0 THEN E=128+E
5160 RETURN ' E = # OF COUNTS LEFT IN MOVE
5170 '
5180 ' *****
5190 ' * GET MIRCOSWITCH'S VALUE *
5200 ' *****
5210 '
5220 POKE 16880,73 : X=USR0(0) ' OUTPUT AN 'I' TO GET THE
      STATUS OF MICRO-SWITCHES
5230 X=USR1(0) ' GET RESPONSE
5240 I=PEEK(16872)-32 ' I= VALUE FROM 0 TO 63
5250 RETURN ' I=0 - ALL CLOSED I=63 - ALL OPEN
5260 '
5270 ' *****
5280 ' * STOP A MOTOR IN MOTION *
5290 ' *****
5300 '
5310 POKE 16880,MT : X=USR0(0) ' STOP MOTOR MT - OUTPUT
      MOTOR AND A 'X'
5320 POKE 16880,88 : X=USR0(0)
5330 RETURN
5340 '
5350 ' *****
5360 ' * SAVE AND LOAD ROUTINES ON *
5370 ' * THE DISK FN=F# *
5380 ' *****
5390 '
5400 TY=1:FY=0:E3=0:E5=0 :ON ERROR GOTO 6340
5410 CLS : PRINT @640,"FILENAME (8 CHARACTERS OR LESS) -
      " ;:PRINT @674,F$(TY)
5420 PRINT "REMARK (240 CHARACTERS OR LESS)"
5430 PRINT M$(TY);
5440 PRINT@10, "DISK FILE ACCESS MENU FOR ";T2$(TY);" FILES
      " ;:PRINT
5450 PRINT "1 - SAVE FILE ON DISK";TAB(30)"6 - CHANGE THE
      FILENAME" : PRINT "2 - LOAD A FILE FROM
      DISK";TAB(30);"7 - CHANGE THE REMARK"
5460 PRINT "3 - CHANGE THE TYPE OF FILE BEING
      ACCESSED";PRINT "4 - DIRECTORY OF DISK FILES AND
      DELETING FILES FROM DISK"
5470 PRINT "5 - RETURN TO TEACH MODE"
5480 PRINT@448,CHR$(30);;:PRINT @448,"ENTER CHOICE ? ";
5490 A$=INKEY$ : IF A$="" THEN 5490 ELSE EC=VAL(A$): PRINT
      A$;
5500 IF EC<1 OR EC>7 THEN 5480
5510 ON EC GOTO 5550 ,5550 , 5530 , 6060 ,5520 ,5630 , 5660

```

```

5520 ON ERROR GOTO 0      :RETURN
5530 IF TY=1 THEN TY=2 ELSE TY=1
5540 GOTO 5410
5550 PRINT @512,"ARE YOU SURE YOU WANT TO ";CHR$(34);: IF
    EC=1 THEN PRINT "SAVE"; ELSE PRINT"LOAD";
5560 PRINT CHR$(34);" ";F$(TY);TY$(TY);" ? ";
5570 A$=INKEY$; IF A$="" THEN 5570
5580 IF A$<>"Y" THEN 5620
5590 POKE 16420,1 : POKE 16912,40 : OUT 236,8
5600 FF$=F$(TY)+TY$(TY); K2=512
5610 ON EC GOSUB 5720 ,5890
5620 GOTO 5410
5630 PRINT@674,CHR$(30);:PRINT@673," ? ";:LL=8 : GOSUB 2430
5640 IF B$<>" " THEN F$(TY)=B$
5650 PRINT@673,CHR$(30);:PRINT @674,F$(TY);:GOTO 5480
5660 PRINT@768,CHR$(30); : PRINT@832,CHR$(30); : PRINT
    @896,CHR$(30); : PRINT@960,CHR$(30); :PRINT@768,"?
    ";:LL=240;GOSUB 2430
5670 IF B$<>" " THEN M$(TY)=B$
5680 PRINT @768,CHR$(30); : PRINT@832,CHR$(30);: PRINT@896,
    CHR$(30); :PRINT@960, CHR$(30); :PRINT @768,M$(TY);
    :GOTO 5480
5690 ? *****
5700 ? *   SAVE A ROUTINE ON THE DISK   *
5710 ? *****
5720 PRINT@K2 ,CHR$(30);:PRINT @K2 ,"   SAVING   ";CHR$(244);
    CHR$(245); CHR$(246);"   ";FF$;
5730 OPEN "D",1,FF$
5740 IF TY=2 THEN 5800
5750 PRINT #1,M$(TY) : PRINT #1,M
5760 FOR I=0 TO M : FOR II=1 TO 8
5770 PRINT #1,N(I,II)
5780 NEXT II,I
5790 CLOSE : RETURN
5800 PRINT #1,M$(TY) : PRINT#1,TL
5810 FOR I=1 TO TL
5820 PRINT#1,RN$(I) : PRINT #1,RN(I)
5830 NEXT I
5840 CLOSE : RETURN
5850 ? *****
5860 ? *   LOAD A ROUTINE - MOVE THE RHINO TO THE LOCATION   *
5870 ? *           IT STOP AT WHEN THIS ROUTINE ENDED.       *
5880 ? *****
5890 PRINT @K2 ,CHR$(30);:PRINT @K2 ,"   LOADING
    ";CHR$(244);CHR$(245);CHR$(246);"   ";FF$;
5900 OPEN "I",1,FF$
5910 IF TY=2 THEN 6010
5920 INPUT #1,M$(TY) : INPUT #1,M
5930 FOR I=1 TO 8 : MC(I)=TT(I) : INPUT #1, N(O,I) :
    TT(I)=N(O,I) : NEXT I
5940 FOR I=1 TO M : FOR II=1 TO 8

```

```

5950 INPUT #1,N(I,II) : TT(II)=TT(II)+N(I,II)
5960 NEXT II,I
5970 CLOSE
5980 IF FY=1 THEN RETURN 'FLAG FOR RETURNING TO RUNNING A
      LOOP
5990 FOR I=1 TO 8 : MC(I)=TT(I)-MC(I) : NEXT I
6000 GOSUB 4240 : RETURN
6010 INPUT #1,M$(TY) : INPUT #1,TL
6020 FOR I=1 TO TL
6030 INPUT #1,RN$(I) : INPUT #1,RN(I)
6040 NEXT I
6050 CLOSE : RETURN
6060 ' *****
6070 ' *   DISK DIR AND DELETING FILES   *
6080 ' *****
6090 E3=1 : CLS : CMD"D:0" : PRINT CHR$(15)
6100 PRINT"THE FILES WITH /RHN AT THE END ARE ROUTINE FILES
      AND":PRINTTAB(15)"/LOP AT THE END ARE LOOP FILES"
6110 PRINT@832,"DID YOU WANT TO DELETE (D) A FILE OR RETURN
      (R) TO MENU ?";
6120 A#=INKEY$:IF A#="" THEN 6120
6130 IF A#="R" THEN E3=0 : GOTO 5410 ELSE IF A#<>"D" THEN
      6120
6140 PRINT@960,"ENTER THE TYPE OF FILE TO BE DELETED;
      1-/RHN, 2-/LOP ?";
6150 A#=INKEY$ : IF A#="" THEN 6150
6160 IF A#<"1" OR A#>"2" THEN 6150 ELSE DK=VAL(A#)
6170 PRINT @960,CHR$(30); : PRINT @960,"ENTER THE FILENAME
      OF THE FILE TO BE DELETED ? ";
6180 LL=8 : GOSUB 2430
6190 IF B#="" THEN PRINT@960,CHR$(30); : PRINT@960,"MUST
      ENTER A FILENAME, TRY AGAIN"; : FOR C=1 TO 1000 : NEXT
      C : PRINT @960,CHR$(30); : GOTO 6090
6200 FF#=B#+TY$(DK) : PRINT@960,CHR$(30); : PRINT@960,"ARE
      YOU SURE YOU WANT ";FF#;" TO BE DELETED ? ";
6210 A#=INKEY$ : IF A#="" THEN 6210
6220 POKE 16420,1 : POKE 16912,40 : OUT 236,8
6230 IF A#<>"Y" THEN PRINT CHR$(30); : GOTO 6090
6240 PRINT @960,CHR$(30); : PRINT @960,"KILLING
      ";CHR$(244);CHR$(245);CHR$(246);" ";FF#;
6250 KILL FF# : FOR C=1 TO 100 : NEXT C :PRINT@960,CHR$(30);
      : GOTO 6090
6260 '
6270 ' *****
6280 ' *   DISK ERROR SUBROUTINE - FOR SAVING AND LOADING *
6290 ' *****
6300 '
6310 ' FLAG STATUS  E3=0 - DISK ACCESSES,1 -
      DIRECTORY/DELETE MODE
6320 '           E5=0 - DISK ACCESSES,1 - RUN LOOPS,2 -
      SAVING AND LOADING LOOPS

```

```

6330 '
6340 E2=ERR/2+1      ' GET ERROR CODE
6350 IF E2<51 THEN ERROR E2      ' NOT A DISK ERROR
6360 CLOSE : CLS : PRINT:PRINT"          *****      DISK ERROR
*****":PRINT
6370 IF E2=53 THEN 6550 ELSE IF E2=54 THEN 6520
6380 IF E2=62 THEN RESUME 6400 ELSE IF E2=65 THEN 6500
6390 RESUME          ' CAUSE ERROR TO HAPPEN AGAIN - SAME LINE
6400 ON ERROR GOTO 6410 : KILL FF$ : GOTO 6420 'KILL THE
FILE STARTED ON THE DISK UNTIL FULL
6410 E2=ERR/2+1 : IF E2<>54 THEN ERROR E2  ' CHECK IF NO
FILE WAS STARTED
6420 PRINT"          ***      DISK IS FULL          ***" :PRINT"To
evaluate this problem you could delete some old files
using
the disk access mode (=) or put a new RHINO master disk
(with "
6430 PRINT"room on it) in drive #0.  Then try saving the
file again
If there is in not enough room to save end program (?) and
try agin later."
6440 PRINT"
NOTE:  MUST change the disk in drive #0 before press a key
to
continue !!! Also never open the disk drive doors when thr
red light is on because the head is in operation !!!"
6450 IF E5=1 THEN PRINT:PRINT"THERE IS AN ERROR IN A ROUTINE
IN THIS LOOP, SO ABORTING TRYING
TO RUN THE REST OF THE LOOP.  PLEASE CORRECT AND TRY AGAIN."
6460 PRINT:PRINT"PRESS ANY KEY TO CONTINUE"
6470 A$=INKEY$ : IF A$="" THEN 6470
6480 IF E5=1 OR E5=2 THEN RESUME 6790  'BACK TO RUN A LOOP
OF ROUTINES
6490 IF E3=0 THEN RESUME 5410 ELSE RESUME 6090
6500 PRINT"          ***      BAD FILENAME
***":PRINT:PRINT"FILENAME ";FF$;" IS NOT AN APPROPRIATE
FILENAME.  TRY AGAIN"
6510 GOTO 6450
6520 IF E5=3 THEN 6590 ELSE PRINT"          ***      FILE NOT
FOUND          ***":PRINT:PRINT"THE FILE ";FF$;" TRYING TO BE
"; : IF E3=0 THEN PRINT"LOADED "; ELSE PRINT"KILLED ";
6530 PRINT"DOES NOT EXISTS
PLEASE CHECK THE DIRECTORY AND TRY AGAIN"
6540 GOTO 6450
6550 PRINT"          ***      BAD FILE NUMBER
***":PRINT:PRINT"THIS IS CAUSE BY NOT HAVING ENOUGH
FILES"
6560 PRINT"YOU SHOULD START OVER (REBOOT) AND ANSWER THE
STARTING QUESTION":PRINTCHR$(34);"HOW MANY FILES ?
";CHR$(34);" WITH AT LEAST ONE "

```



```

6570 PRINT"GOOD LUCK - PRESS ANY KEY TO CONTINUE "
6580 A$=INKEY$ : IF A$="" THEN 6580 ELSE CMD"I", "DO START"
6590 PRINT "          *** FILE NOT FOUND ***
        ":PRINT:PRINT"FILE RHINO/SAV CREATED BY THE FIRST HALF
        OF THIS RHINO TEACH PROGRAM
        IS NOT THERE SO MUST START OVER BY REBOOTING THE SYSTEM"
6600 PRINT:PRINT"PRESS ANY KEY TO CONTINUE"
6610 A$=INKEY$ : IF A$="" THEN 6610 ELSE CMD"I", "DO START"
6620 '
6630 ' *****
6640 ' * *
6650 ' * RUN A LOOP OF ROUTINES *
6660 ' * *
6670 ' *****
6680 '
6690 PRINT @960, "ARE YOU SURE, YOU WANT TO RUN A LOOP OF
        ROUTINES (Y or N) ?";
6700 A$=INKEY$ : IF A$="" THEN 6700 ELSE PRINT@960,CHR$(30);
        : IF A$<>"Y" THEN RETURN
6710 CLS : PRINT: PRINTCHR$(23): PRINT:PRINT: PRINT:
        PRINTTAB(5); "ONE MOMENT PLEASE"
6720 FOR I=1 TO M : FOR II=1 TO 8 ' SAVE THE CURRENT
        ROUTINE SO CAN USE SAME ARRAY FOR RUNNING LOOPS
6730 N2(I,II)=N(I,II)
6740 NEXT II : NEXT I : M2=M
6750 E3=0 : YY=0 : ON ERROR GOTO 6340
6760 ' *****
6770 ' * PRINT OUT SCREEN FOR RUNNING LOOPS *
6780 ' *****
6790 CLS: PRINTTAB(20) "RUN A LOOP OF ROUTINES"
6800 PRINT @86, "LOOP TITLE - ";F$(2);
6810 PRINT:PRINTTAB(6) "ROUTINE TITLE # REPEATS";
        TAB(38); "ROUTINE TITLE # REPEATS"
6820 ' ***** PRINT #'S OUT FOR ROUTINES *****
6830 RY=0 : FOR I=1 TO 10
6840 PRINT @132+RY+64*I, I:PRINT@163+64*I, I+10
6850 IF I=9 THEN RY=-1
6860 NEXT I : IF Y5=1 THEN 7400
6870 ' ***** FIRST SO CHECK IF TO LOAD OR START FRESH
        *****
6880 Y5=1 : PRINT @960, "L-LOAD A LOOP, S-SAME LOOP USED
        BEFORE, OR N-ENTER A NEW LOOP ?";
6890 A$=INKEY$ : IF A$="" THEN 6890 ELSE PRINT
        @960,CHR$(30);
6900 IF A$="L" THEN 7830
6905 IF A$="S" THEN 7400
6910 IF A$<>"N" THEN 6880
6920 ' ***** ENTER A NEW LOOP OF ROUTINES *****
6930 PRINT@960, "ENTER ROUTINE TITLE AND # OF REPEATS, JUST
        PRESS ENTER TO STOP";
6940 LD=136:TL=0

```

```

6950 FOR I=1 TO 10
6960 PRINT @LD+64*I,"? " ; LL=8 : GOSUB 2430
6970 IF B#="" THEN PRINT @960,CHR$(30) ; PRINT@LD+64*I,"
"; : GOTO 7080
6980 TL=TL+1 : RN$(TL)=B$
6990 PRINT@LD+64*I+14,"? " ; GOSUB 2560 : IF VV=0 THEN
PRINT @LD+64*I+15," " ; PRINT @960,"ENTER ROUTINE
TITLE AND # OF REPEATS, JUST PRESS ENTER TO STOP" ;
GOTO 6990
7000 RN(TL)=VAL(B$) : IF RN(TL)=0 THEN RN(TL)=1
7010 PRINT @LD+64*I,STRING$(28,32) ; PRINT
@LD+64*I,RN$(TL) ; PRINT @LD+64*I+15,RN(TL) ;
7020 NEXT I
7030 IF LD=136 THEN LD=LD+32 : GOTO 6950
7040 ' ***** OVER THE MAX OF TWENTY *****
7050 PRINT @960,CHR$(30) ; PRINT@960,"20 LOOP IS ALL THAT
CAN BE ENTERED" ;
7060 FOR I=1 TO 1000 : NEXT I : PRINT CHR$(29) ; CHR$(30) ;
7070 ' ***** WHAT NEXT ? *****
7080 PRINT
@960,"A-ADD,C-CHANGE,D-DELETE,L-LOAD,R-RUN,S-SAVE,T-TEA
H MODE ?" ;
7090 A$=INKEY$ : IF A#="" THEN 7090
7100 PRINT @960,CHR$(30) ;
7110 IF A#="T" THEN 7480
7120 IF A#="R" THEN 7590
7130 IF A#="A" THEN 7280
7140 IF A#="L" OR A#="S" THEN 7830
7150 IF A#="D" OR A#="C" THEN AA$=A$ : GOTO 7180 ELSE 7080
7160 '
7170 ' ***** ENTER LOOP # *****
7180 PRINT @960,"ENTER LOOP # TO BE " ; IF AA#="C" THEN
PRINT "CHANGED ? " ; ELSE PRINT "DELETED ? " ;
7190 GOSUB 2560 : IF VV=0 THEN PRINT @960,CHR$(30) ; GOTO
7180
7200 NU=ABS(VAL(B$)) : IF NU<1 OR NU>TL THEN PRINT@960,
CHR$(30) ; PRINT @960,"MUST BE A USABLE NUMBER" ; FOR
C=1 TO 1000:NEXT C: PRINT@960,CHR$(30) ; GOTO 7080
7210 IF AA#="D" THEN 7330
7220 '
7230 ' ***** CHANGE A LOOP TITLE *****
7240 LD=136 : MU=NU : IF NU>10 THEN LD=168 : NU=NU-10
7250 GOSUB 7750 : GOTO 7080
7260 '
7270 ' ***** ADD A LOOP TITLE *****
7280 IF TL=20 THEN PRINT @960,"CAN NOT HAVE MORE THAT 20
ROUTINES !" ; FOR C=1 TO 1000:NEXTC:PRINT
@960,CHR$(30) ; GOTO 7080
7290 TL=TL+1 : NU=TL : MU=NU : LD=136 : IF TL>10 THEN LD=168:
NU=TL-10
7300 GOSUB 7750 : GOTO 7080

```

```

7310 '
7320 ' *****          DELETE A LOOP TITLE          *****
7330 TL=TL-1
7340 FOR J=NU TO TL
7350     RN$(J)=RN$(J+1) : RN(J)=RN(J+1)
7360 NEXT J : GOTO 6790
7370 '
7380 ' *****          PRINT OUT THE CURRENT LOOP OF ROUTINES
*****
7390 '
7400 IF TL=0 THEN 7080 ELSE LD=136 : T8=TL:T9=0:T7=0 : IF
TL>10 THEN T9=TL-10 : T8=10
7410 FOR I= 1 TO T8
7420     PRINT @LD+64*I,RN$(I+T7); : PRINT
@LD+64*I+15,RN(I+T7);
7430 NEXT I
7440 IF T9>0 THEN T8=T9 : T9=0 : T7=10 : LD=LD+32: GOTO 7410
7450 GOTO 7080
7460 '
7470 ' *****          RETURN TO TEACH MODE          *****
7480 CLS : PRINT :
PRINTCHR$(23):PRINT:PRINT:PRINT:PRINTTAB(7)"ONE MOMENT
PLEASE"
7490 ON ERROR GOTO 0
7500 M=M2 : FOR I=1 TO M : FOR II=1 TO 8 ' LOAD THE CURRENT
ROUTINE BACK IN ARRAY
7510 N(I,II)=N2(I,II)
7520 NEXT II : NEXT I : RETURN
7530 '
7540 ' *****
7550 ' * RUN A LOOP OF ROUTINES *
7560 ' *****
7570 '
7580 ' ***** TL=# OF ROUTINES IN THE LOOP *****
7590 FY=1: TY=1 : E5=1
7600 FOR KL=1 TO TL
7610 FF#=RN$(KL)+"/RHN" : K2=960 : GOSUB 5890
7620 FOR I=1 TO 8 : TT(I)=MC(I) : NEXT I
7630 PRINT @960,CHR$(30); : PRINT @960,"RUNNING ";RN$(KL);"
ROUTINE ";RN(KL);" TIMES";
7640 RR=RN(KL) : GOSUB 3730
7650 IF V=1 OR V=2 THEN 7700 'ERROR ABORTED RUN A ROUTINE
(PF=0)
7660 IF PF=1 THEN 7690
7670 PRINT @960,CHR$(30); : PRINT @960,"END OF ";RN$(KL);"
ROUTINE, DO YOU WANT TO CONTINUE THE LOOP?";
7680 A$=INKEY$ : IF A$="" THEN 7680 ELSE IF A$<>"Y" THEN
PRINT@960,CHR$(30); : GOTO 7400
7690 NEXT KL : FY=0 : PRINT@960,CHR$(30); : GOTO 7400
7700 PRINT @960,CHR$(30); : PRINT @960,"STOPED THE ROUTINE
DURING RUNNING, CORRECT PROBLEM + TRY AGAIN";

```

```

7710 FOR C=1 TO 1000 : NEXT C : PRINT @960,CHR$(30);
7720 FY=0 : GOTO 7400
7730 '
7740 ' *****      CHANGE OR ADD A ROUTINE TITLE AND #
*****
7750 PRINT@LD+NU*64,STRING$(25,32);: PRINT @LD+NU*64,"? ";
:LL=8:GOSUB 2430
7760 IF B$<>" " THEN RN$(MU)=B$
7770 PRINT @LD+NU*64+14,"? "; : GOSUB 2560 : IF VV=0 THEN
PRINT @LD+NU*64+15," " " " : GOTO 7770
7780 RN(MU)=ABS(VAL(B$)) : IF RN(MU)=0 THEN RN(MU)=1
7790 PRINT @LD+NU*64,STRING$(28,32); : PRINT
@LD+64*NU,RN$(MU); : PRINT @LD+64*NU+15,RN(MU);
7800 RETURN
7810 '
7820 ' *****      SAVE AND LOAD A LOOP OF ROUTINES
*****
7830 AA$=A$ : IF AA$="S" THEN PRINT @960,"SAVING "; ELSE
PRINT @960,"LOADING ";
7840 PRINT "LOOPS, W-WITH LOOP TITLE ABOVE, N-NEW TITLE,
R-RETURN ?";
7850 A$=INKEY$ : IF A$="" THEN 7850
7860 IF A$="R" THEN PRINT @960,CHR$(30);:GOTO 7080
7870 IF A$="W" THEN 7920
7880 IF A$<>"N" THEN 7850
7890 PRINT @960,CHR$(30);:PRINT@960,"WHAT IS THE NEW LOOP
TITLE ?"; : LL=8 : GOSUB 2430
7900 IF B$="" THEN PRINT @960,CHR$(30); : GOTO 7080
7910 F$(2)=B$ :PRINT@86,"LOOP TITLE - ";F$(2);" "
7920 FF$=F$(2)+TY$(2): POKE 16420,1 : POKE 16912,40 : OUT
236,8
7930 TY=2 : K2=960 : E5=2 : PRINT @960,CHR$(30); : IF
AA$="S" THEN GOSUB 5720 ELSE GOSUB 5890
7940 GOTO 6790

```

BIBLIOGRAPHY

Hands-On-Introduction to Robotics, The Manual for the XR-1.
Champaign, Illinois: RHINO Robots Inc., 1982.

Kater, David A., and Thomas, Susan J. TRS-80 Graphics.
Peterborough, New Hampshire: BYTE Books, 1982.

Keller, Erik L. "Clever Robots Set To Enter Industry en masse." Electronics 56 (November 17, 1983): 116-129.

TRS-80 Model III Disk System Owner's Manual. Fort Worth, Texas: Tandy Corporation, 1980.

TRS-80 Model III Operation and BASIC Language Reference Manual. Fort Worth, Texas: Tandy Corporation, 1980.