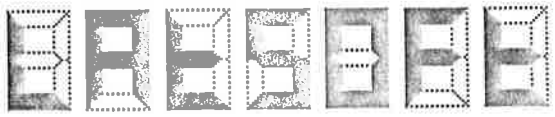


Figure 2-10
Soft Home Position

Pressing the ENTER key erases the motor step count display and updates the move count on the extreme right.

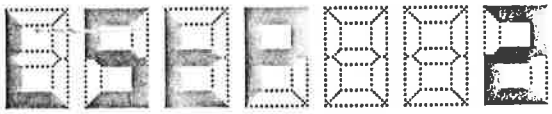


FIGURE 2-11
Learn, Stored position 2

To complete the program, press the END key. The "L" on the left side of the display will disappear, signifying that the LEARN sequence has ended. The END key is always used to change modes (and always returns the system to the PLAY mode). The "P" will re-appear, to indicate that the robot is again in the PLAY mode.

RUN MODE

RUNNING YOUR SINGLE MOTOR MOVE PROGRAM

Now that you have taught the robot a move sequence, you are ready to run the sequence under the control of the teach pendant system. Push the RUN key once and watch the robot carry out your programmed instructions.

While the robot is in the RUN mode, the display reflects it's new condition. As before, the mode is at the left and the move number is on the right. During the first move in the sequence, the display will show:

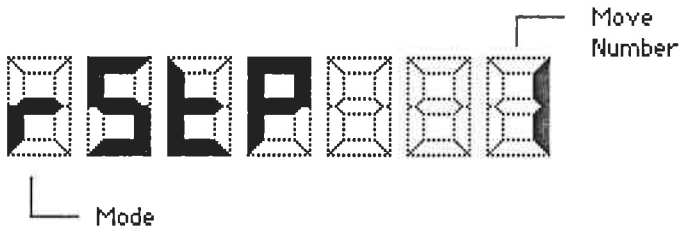


FIGURE 2-12
Run, Stored position 1

While the robot is returning HOME, which is the second move, the display will indicate move two by displaying the following:

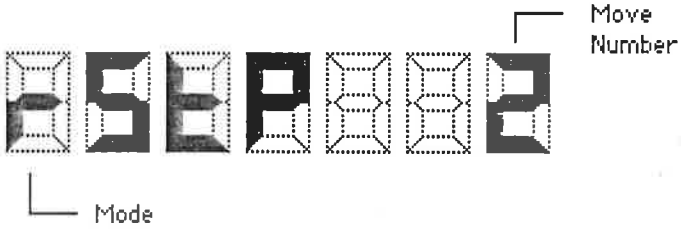


FIGURE 2-13
Run, Stored position 2

Once you have pressed RUN, the robot will execute the program once (unless you have programmed a continuous loop).

If you press the RUN/HALT key, during program execution, the robot stops immediately and goes into the HALT mode. The teach pendant display shows **"rHLt"**. Halt is ended by pressing the RUN key once again. This feature is useful for instructional presentations and for collision avoidance during a run.

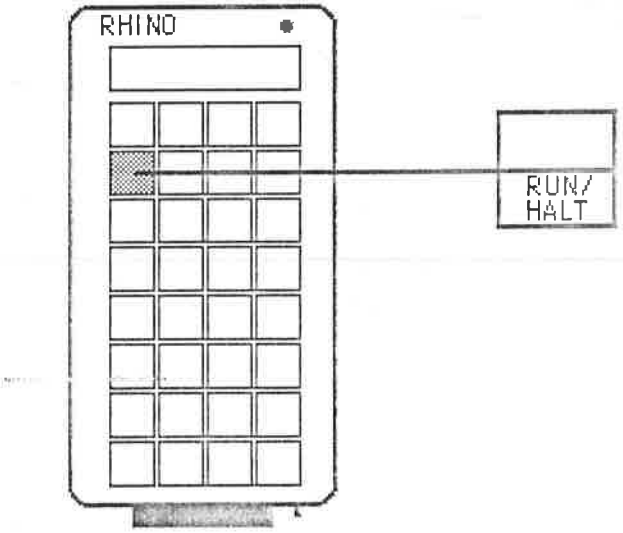


FIGURE 2-14
The RUN/HALT Key

If you press the END key during program execution, the robot stops immediately. Thereafter, the display shows a "P" on the left to indicate that the system has reverted to the PLAY mode. When RUN is pressed after END, the robot will begin the program from the beginning by executing the first point from the robot's present position. If the END key is pressed while in the HALT mode, the system returns to the PLAY mode.

After running a sequence of moves, you can go back into the LEARN mode and add other moves to the end of the previous move sequence. This technique, allows you to make a complex series of moves simply by stringing together short LEARN sequences.

TEACHING MULTI AXIS MOVES

The essential difference between multi axis and single axis moves is in the use of the ENTER command. In single motor moves, you pressed ENTER after one axis had been moved to bring the arm to a given location. In the multi axis move, you press ENTER after you have moved a combination of axes to reach a given point in space. Your preliminary steps are identical, i.e. do a "Hard Home" (if not already done), CLEAR memory (optional), set a "Soft Home" (optional), and enter the LEARN mode.

When planning a multi axis move, keep the following important principles of robot movement in mind:

PRINCIPLE #1.

Start moving your robot from the waist up and move the fingers last. For instance, if you are planning to reach a point which requires waist ("F" motor) and bicep ("E" motor) movement, you start with the waist. As you move the motors, you will see the appropriate displays.

After you have moved the waist 200 counts, your display would show:

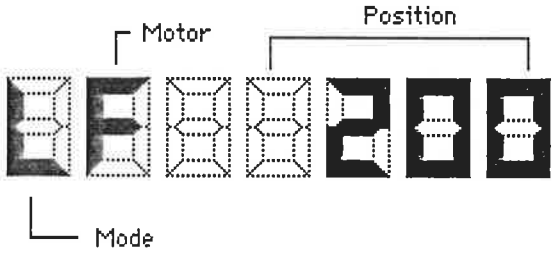


FIGURE 2-15
Learn, F motor, 200 position

There is no direction indicator if the move is in a positive direction (counterclockwise on the "F" motor).

Now switch to the bicep ("E" motor) without pressing the ENTER key. Notice that the display changes. Your display would show the following after a 150 count move up.

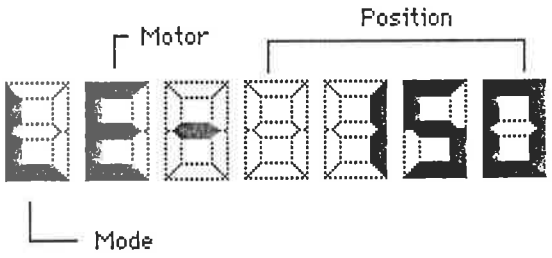


FIGURE 2-16
Learn, E motor, -150 position

Your motor reading has changed from "F" to "E". The number of steps apply to the "E" motor now. At this point you are still at move 0 (zero) because you have not recorded any moves in this sequence even though you have moved two axes in the first move.

Now, if you push the ENTER key, your display will change to:

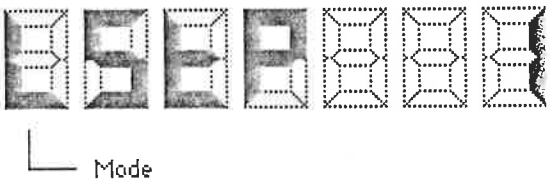


FIGURE 2-17
Learn, Stored position 1

This indicates to the operator that the first move of the sequence has been completed. You could have moved all the other axes before recording the move. Remember that you can also re-adjust an axis before you record a move. Just press the appropriate key on the needed axis. Even after you have recorded the move, don't worry if your position is not exactly as you had wished. You can modify any move later with the EDIT capability, a very useful feature of the Rhino XR teach pendant.

A further point to remember is that the XR robot moves from point to point, not in a specific path. Those motors that have further to go take longer to move than the

motors that have a short way to go. In order to avoid crashes, it is best to remember the points at which you ENTER a move and try to estimate the path that the robot will take. After a while, you will become quite proficient but you may need a little patience in the beginning.

PRINCIPLE #2.

Make the gripper OPEN/CLOSE a separate movement. Think about your own arm. If you wanted to pick up (or put down) an object, you would move your arm to the chosen location and then open or close your hand. In order to appreciate this point, think about what would happen if you did not follow this principle. Your hand would open before your arm had moved to the place where you wanted the object placed, and whatever was in your hand would be dropped along the way. If you wanted to move your arm to a specific location to pick up an object, you would close your fingers on thin air, or on the wrong object, if you closed your fingers before your arm was properly positioned.

If this doesn't make sense, try an experiment. Ask another person to talk you through a move with your own arm while you are blindfolded. If the objective is to pick up an object from a desk top, you will find that your helper will direct you first to turn your body, then to lower your arm, moving it forward and back, and finally to open your hand when you are correctly positioned.

PRINCIPLE #3.

Each move can move any one motor in one direction only. Again, the best way to appreciate this principle is to think about what would happen if you failed to observe it. Suppose that you wanted to set an object down at the far side of a divider. You would raise the arm, move it over the divider, and lower it on the other side. If you did all this in one movement, the arm would take the shortest route to the end location and would crash full force into the divider that you were trying to avoid. As you may remember, the controller combines all the positive and negative motor moves on the same axis in the same movement. The algebraic sum makes the robot more efficient when you are practicing but it can cause problems unless you keep the process in mind. If you wish to make the robot move in two different moves in two separate directions, you must make a separate move in each direction.

PRINCIPLE #4.

Watch how the robot moves from point to point to accustom yourself to its revolute coordinate system. The XR, like many industrial robots, moves from point to point. The arm moves from one location to another, but because of its physical construction, it does not move in a straight line. Until you become accustomed to this property and can accurately predict exactly how the robot will move, it will be best to keep obstacles out of your way.

PRINCIPLE #5.

Avoid complicated, multi-axis moves, in the beginning. It is better to use the ENTER key often and to use many single axis moves. There is less likelihood of a "serious" crash (XR crashes are never really serious but these things are to be avoided) and it is usually easier to EDIT a less complicated move sequence.

PRINCIPLE #6.

When you tell the robot to GO SOFT HOME at the end of a move sequence, it will go to its "Soft Home" position in a single, multi axis move. If this is likely to cause problems, lead the robot to some safe intermediate location, and record the move with the ENTER key, before you give the GO SOFT HOME command. For example, if you set an object down, remember to move to above the object after you have released it. Then push the ENTER key. This will prevent the XR from possibly knocking down the object on its way home.

RUNNING MULTI AXIS MOVES

After you have finished entering the program into the system, press the END key and then the RUN key. Remember that you can HALT the system anywhere within the cycle by pressing RUN a second time and then pressing RUN again when you are ready to resume the cycle. If you press END during a run, the robot will stop immediately and return to the PLAY mode.

If you are planning a lengthy sequence, you may want to program it in sections. You might program a module, run it for testing and possible editing, and then program another module. Under these circumstances, you would not want the robot to cycle continuously. After you had run the sequence once or twice, you would return to the LEARN mode and do some more programming.

Keep in mind that when the robot leaves the RUN mode, it automatically goes into PLAY. If you press a motor move key and move the robot away from its "Soft Home" position and then decide to go back to the LEARN mode, it is best (but not necessary) to press GO SOFT HOME first to make sure that the robot starts from the same "Soft Home" position each time.

Before beginning your second move sequence module, take a look at the teach pendant display. It should show an "L" on the left for LEARN and a number other than "0" on the extreme right. The number should correspond to the final count in your previous move sequence.

SECTION 3

EDIT MODE

FUNCTION OF THE EDIT MODE

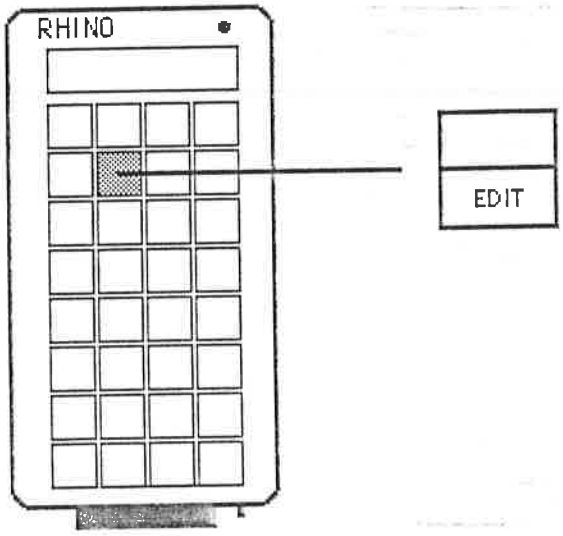


FIGURE 3-1
The **EDIT** key

In the EDIT mode, you can trouble shoot and/or modify a program that has been previously entered into the teach pendant system memory. Using EDIT is similar to writing the original program but easier.

The ability to edit a program is very important for efficient robot operation. In many industrial situations a program may be written off-line and loaded into the robot/controller/pendant system. The programmer, who is removed from the scene, may not know exactly where the robot is to be located and where the vises and other fixtures may be placed. Such factors are very important for the proper operation of a robot and will require relatively minor modifications by the shop floor operator. The operator can bring the program to the plant floor and then make his final adjustments at the actual fixtures.

Another common example would be a program that is working perfectly, but minor changes in the pick up or delivery points require minor program changes. When that

occurs, it is more efficient to edit a program for the motor positions of the specific motor(s) than to re-write the entire program.

Of course, the EDIT mode can not be used unless you have used the LEARN mode to ENTER a program. If you attempt to EDIT a non existant program, the display will remind you that there is no program in memory by displaying "no dAtA". You can enter the EDIT mode only from the PLAY mode.

AN EXAMPLE

To give you an idea of how EDIT works, we have designed a simple example. (You may want to enter this program into your system so that you can follow the text more easily).

Suppose you had entered the following program in LEARN mode on your teach pendant:

Move 1	F	- 300	waist clockwise	
	A		gripper open	ENTER
Move 2	E	250	shoulder down	
	D	250	elbow up	ENTER
Move 3	F	790	waist counterclockwise	ENTER
Move 4	D	- 300	elbow down	
	C	- 250	wrist down	
	B	500	wrist rotate clockwise	ENTER
Move 5	F	600	waist clockwise	ENTER
Move 6	A		gripper close	
	B	- 250	wrist rotate counterclockwise	
	C	300	wrist up	
	F	300	waist clockwise	ENTER
Move 7	GO SOFT HOME			ENTER
	END			

Your display has now returned to PLAY and will display a "P".

USING THE NEXT MOTOR AND PREVIOUS MOTOR KEYS

When you push the EDIT key on the pendant, you will see the following on the display:

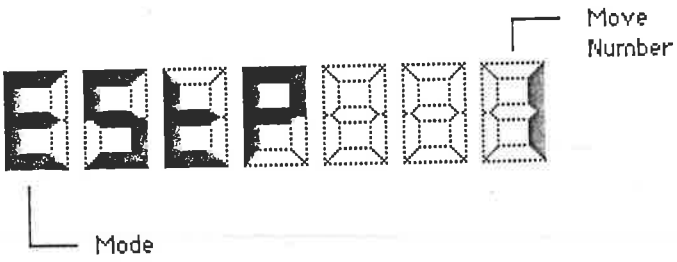


FIGURE 3-2
Edit, Stored position 1

which means that the first move has been executed. In EDIT, you can see a display for every motor or I/O command in every record. To view each motor or I/O segment of a move (record) press NEXT MOTOR or PREVIOUS MOTOR.

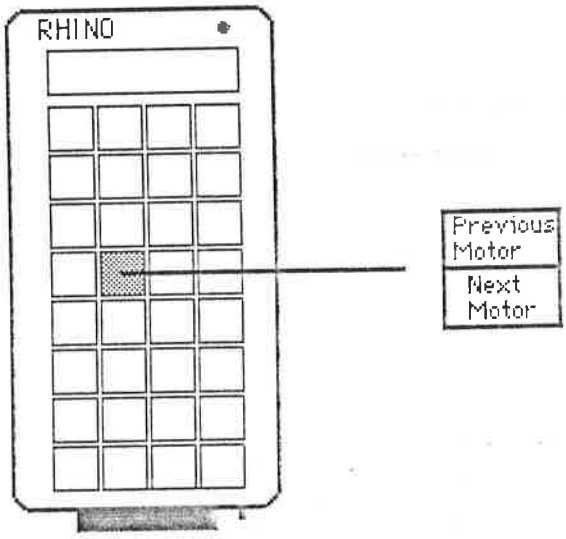


FIGURE 3-3
The PREVIOUS MOTOR/NEXT MOTOR Key

If a specific motor was not moved, it will show 0 in the step section of the display for that motor. The display for a motor that has not moved looks like this:

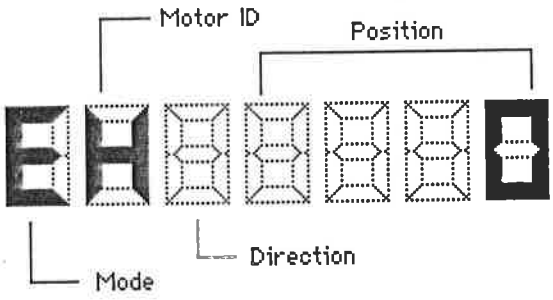


FIGURE 3-4
Edit, H motor, 0 position

The first time NEXT MOTOR is pressed, the status of the A motor (gripper) is displayed as shown:

FIGURE 3-5
Hand open indication.

The capital letter C appears in the second digit since the gripper is a **commanded** position and not a location.

Now press the NEXT MOTOR key again and the display will change to show the move information for the next (B) motor:

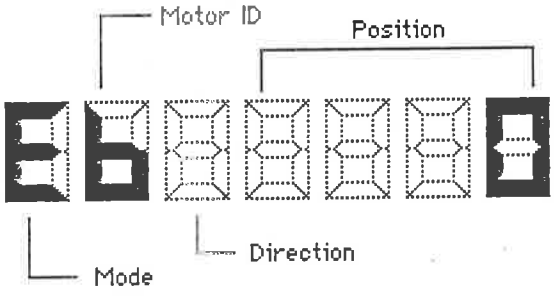


FIGURE 3-6
Edit, B motor, 0 position

This indicates that you did not move the "B" motor before the first press of the ENTER key.

Press the NEXT MOTOR key once again and the display will show:

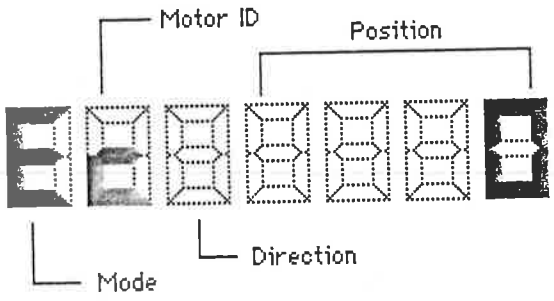


FIGURE 3-7
Edit, C motor, 0 position

This display indicates that the "C" motor was not moved from the hard home position.

Press the NEXT MOTOR key again and again to see the displays for the "D", "E", and "F", "G" and "H" motors. Further pressing of the NEXT MOTOR key will display the I/O status. This will be discussed later. For each motor (except the "A" motor), you will see a set of numbers. For the "A" motor you will see **oPEn** or **CLoSE** only. This shows the condition of the fingers (in a pneumatic emulation).

Now try the PREVIOUS MOTOR key. You will again see displays for motors "G", "F", "E", "D", "C", "B", and "A", moving in the opposite direction from the NEXT MOTOR key.

To EDIT the position of the "F" motor, you simply use the "F" motor keys to move the "F" motor and change the display at the same time. Remember, the NEXT MOTOR and PREVIOUS MOTOR keys will allow you to select a motor within a specific composite 8 motor move. They can not move the system from one composite 8-motor move to another.

CHANGING MOTOR POSITIONS

To modify the motor move values, first make sure that the motor you are addressing is actually connected to the system. This is especially important for the "G" and "H" motors which are usually reserved for accessories such as the conveyor, rotary table, slide base, or X-Y table. If a motor is not connected to a motor port, the display will show "OFF" when the move keys for that motor are pressed. The system automatically disables the keys that do not have a motor connected to them.

In the beginning, until you become familiar with the robotic system, it is advisable to change only motors that were moved (have non zero values). Beginners may find that the arm trajectory changes in unexpected ways when zero value moves are altered. In the first move of the above example, you should change the count for the waist, or "F" motor only (do not modify the other motors at this time).

Let us assume that you decided to change the count for the "F" motor in the first move from -300 to -500. Use the PREVIOUS MOTOR and the NEXT MOTOR keys to display the "F" motor count (if desired), then use the "F" motor move keys to move the axis to the desired motor position. The robot will follow the changes as they are made. The "F" motor display in the first move will now show:

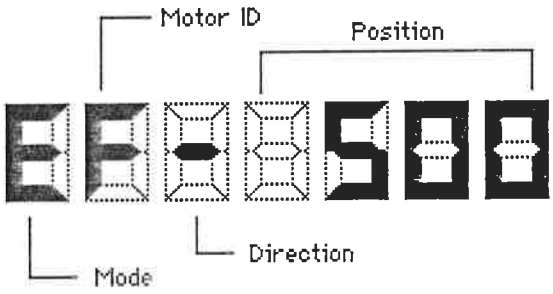


FIGURE 3-8
Edit, F motor, -500 position

Once you have changed a value, you do not have to key ENTER to confirm the change.

Editing one location does not affect the other locations. (The system displays each move as an absolute location from the hard home position. It remembers each move in the computer as an absolute position from the hard home position. Thus editing one move does not effect the others.)

You do not have to display the motor you going to EDIT; once you are in the record you wish to change, moving the appropriate motor will record the new value even if it is currently being displayed. The displaying of the motor however is useful if you want to know the exact location of the new position.

In the next section, we will show you how to step forwards and backwards through the programmed moves.

THE NEXT MOVE/PREVIOUS MOVE KEYS

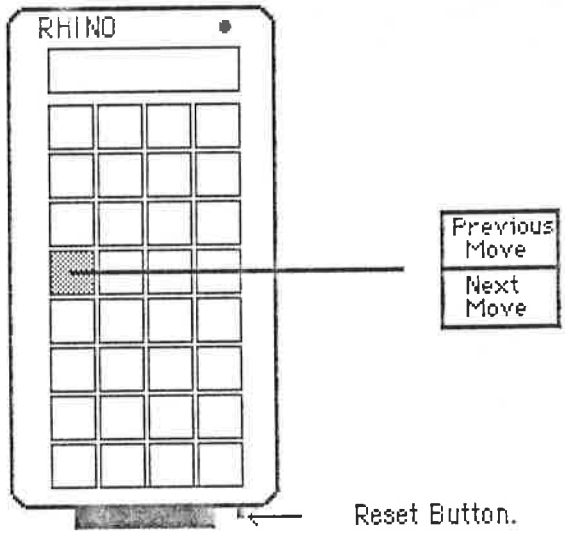


FIGURE 3-9
The PREVIOUS MOVE/NEXT MOVE key

When you are satisfied with the motor values in the first move of the sequence, press the NEXT MOVE key to move to the next composite 8-motor move. When you press the key, the robot will move to the next location. Right after you have pressed the key, you will see the following display:

FIGURE 3-10
Edit, Step 2

Use the NEXT MOTOR and PREVIOUS MOTOR keys just as you did on the first move to display motor counts as needed. You can change the "E" or "D" motors within the second step in the sequence.

If you wish to inspect the last (previous) move (that has already been edited), press the PREVIOUS MOVE key on the pendant. You may press this KEY repeatedly to "Back Up" several steps to see the flow or to modify a sequence. The robot moves back through the moves. After using the PREVIOUS MOVE key to "Back Up" through a series of moves, you can use the NEXT MOVE key to move forward through them to decide if you have modified the sequence to your satisfaction.

Continue to use the NEXT MOVE key until you have moved through the entire program. If you move beyond the last move, your display will show the first move. The editor "wraps around". So, if the first move is in the display and you press PREVIOUS MOVE, the editor will go to the last move.

Another very useful feature of the EDIT function allows you to leave the EDIT mode, at any time without having to step through all the moves, by pressing the END key. This can be very useful if you are working on a long move sequence with many moves and you do not want to step through the entire program.

INSERT MOVE/DELETE MOVE KEYS

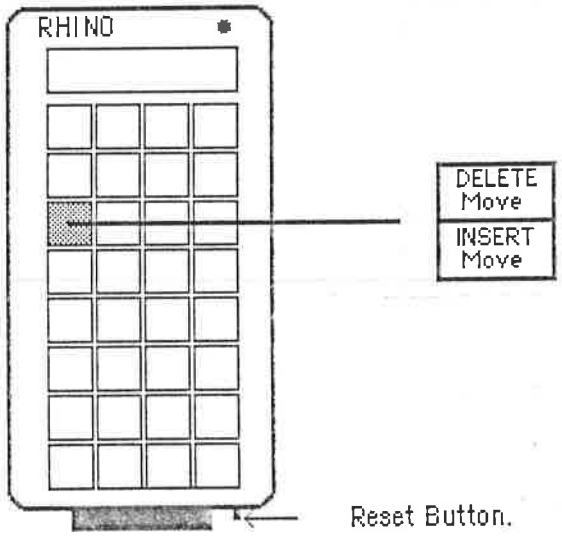


FIGURE 3-11
The DELETE MOVE/INSERT MOVE Key

The use of the INSERT and the DELETE keys enable the user to insert a new move between two existing moves and to delete an existing move.

To INSERT a move, step through the program till you reach the step **PRIOR** to the point at which you want to insert the move. Press the INSERT key. The display will show "EinS XX" where the XX is the number of the new move. For example if you are at step 3 and wish to insert a move following step 3, pressing insert will cause the display to show "EinS 4". Once in the INSERT mode, enter the move as if you were teaching a point in the LEARN mode. All keys will be active and will be recorded till you press the ENTER key. If you want to insert more than one move, the moves have to be inserted one at at time. After inserting the first record, press INSERT again and key in another record.

To DELETE a move, step through the program till you reach the step that you want to delete. Pressing the DELETE key will delete the move record and move the robot to the next move in the sequence. The next step number will be displayed.

Care should be taken when inserting and deleting moves, especially when your application uses I/O. Inserting or deleting a move in which the states of the I/O is changed, may have unexpected results later in the program.

INSERT MOVE/DELETE MOVE KEYS

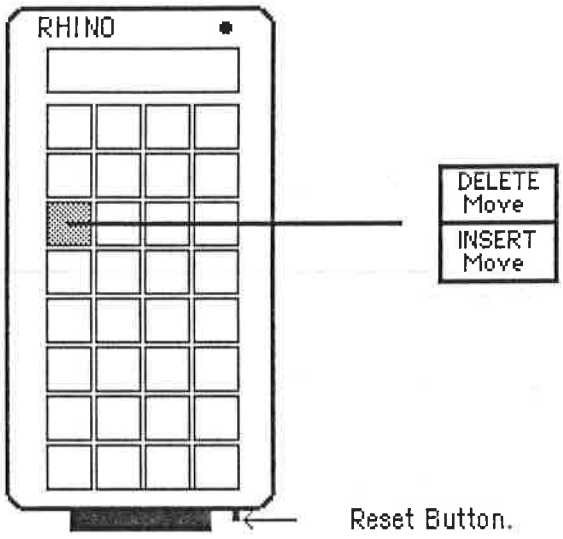


FIGURE 3-11
The DELETE MOVE/INSERT MOVE Key

The use of the INSERT and the DELETE keys enable the user to insert a new move between two existing moves and to delete an existing move.

To INSERT a move, step through the program till you reach the step **PRIOR** to the point at which you want to insert the move. Press the INSERT key. The display will show "EinS XX" where the XX is the number of the new move. For example if you are at step 3 and wish to insert a move following step 3, pressing insert will cause the display to show "EinS 4". Once in the INSERT mode, enter the move as if you were teaching a point in the LEARN mode. All keys will be active and will be recorded till you press the ENTER key. If you want to insert more than one move, the moves have to be inserted one at at time. After inserting the first record, press INSERT again and key in another record.

To DELETE a move, step through the program till you reach the step that you want to delete. Pressing the DELETE key will delete the move record and move the robot to the next move in the sequence. The next step number will be displayed.

Care should be taken when inserting and deleting moves, especially when your application uses I/O. Inserting or deleting a move in which the states of the I/O is changed, may have unexpected results later in the program.

ADDITIONAL COMMENTS ABOUT THE EDITOR

Another useful function of the NEXT MOTOR key is that it functions as a SINGLE STEP key which allows the user to execute his program on step at a time before the program is executed in the RUN mode. When using the NEXT MOTOR key as a single step function, any jump instructions, conditional or unconditional, are ignored; the next physical program step is always executed.

The single step function can also be implemented with the RUN key; however when this key is used, all jump commands are executed as they would be in the RUN mode. Pressing the RUN key while in the EDIT mode moves the robot to the next logical position taking into account any jump instructions be they based on inputs or not.

When single stepping through a program and a WAIT ON INPUT instruction is executed, the system will not proceed until the input condition is matched. You will see a blinking display informing you that the system is waiting on a particular input condition. If you do not have the hardware set up to change the inputs, pressing the NOT WAIT INPUT key will force the system to recognize a matched input condition. You may also put the system into a mode where all waits are disabled. See Appendix III.

SECTION 4

CONTROLLING AND USING I/O

For the robot to fully sense and control its environment, the controller must have I/C capabilities. The Mark III controller has 8 inputs, 8 outputs and two auxiliary motor All of these can be sensed and controlled from the teach pendant through the contr

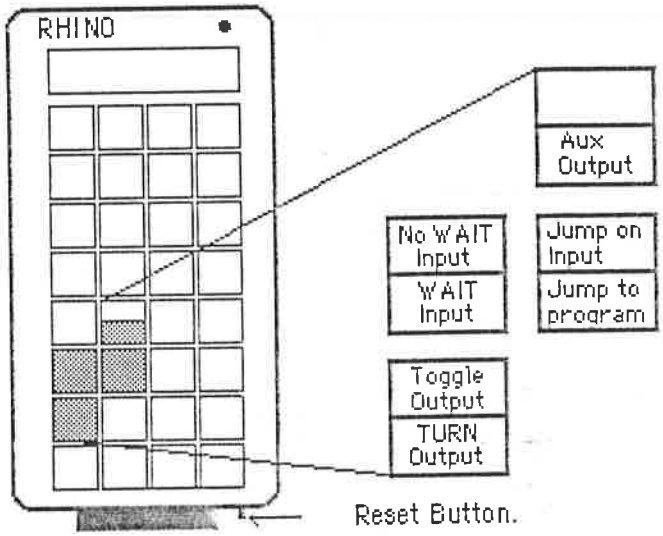


FIGURE 4-1
Keys associated with the Input and Output functions

Before discussing the various commands available for controlling the I/O, let us sp moment to familiarize ourselves with the display concept used by the teach penda commands. Since the display is limited to 7 seven segment digits, an abbreviated is used to show the states of what I/O commands have been programmed into a g teach pendant record.

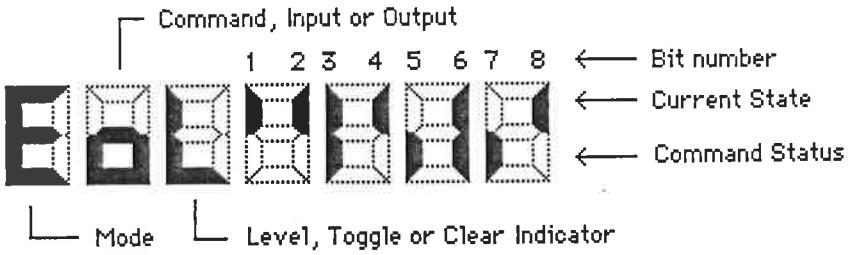


FIGURE 4-4
Edit, Output Level, Indicators

AUX PORTS

Setting the level of the AUX ports requires a two key sequence. To set an Aux port, press the AUX OUTPUT key followed by the 1(ON or OFF) or the 2(ON or OFF) key. Pressing the AUX OUTPUT key immediately after the initial keypress will abort the command. Note that on the MARK III controller the state of both Aux ports is displayed on two LEDs located next to the switches that reverse the polarity of the Aux Ports.

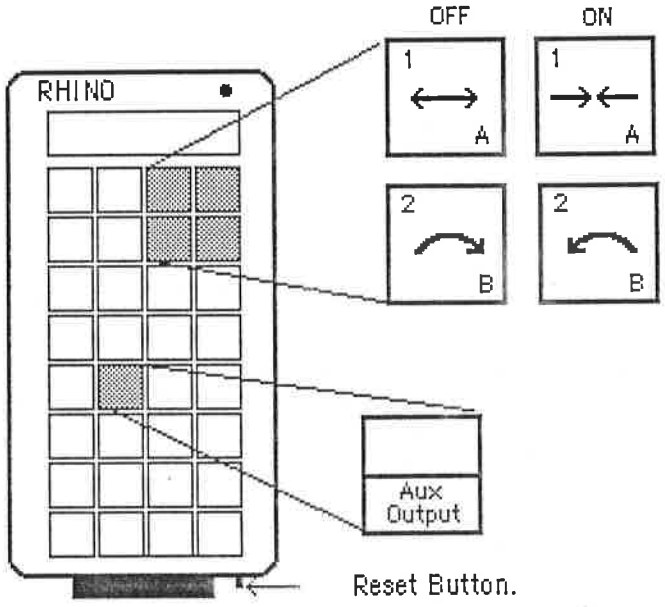


FIGURE 4-5
The AUXILIARY PORT Control Keys

When AUX OUTPUT is pressed, the display shows **Au** preceded by the mode. In the PLAY mode, the display would show the following after the AUX OUTPUT key is pressed.

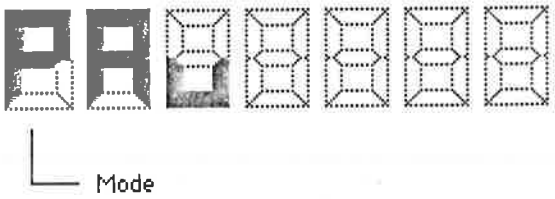


FIGURE 4-6
Play, Aux ports, waiting...

If the **1 ON** key is pressed, the display will change to:

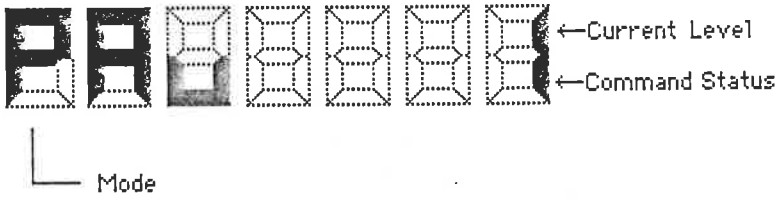


FIGURE 4-7
Play, Aux Ports, Port 1 on

Indicating that AUX 1 is now turned ON. If the AUX OUTPUT key is pressed again and then the **1 OFF** key is pressed, the display will change to:

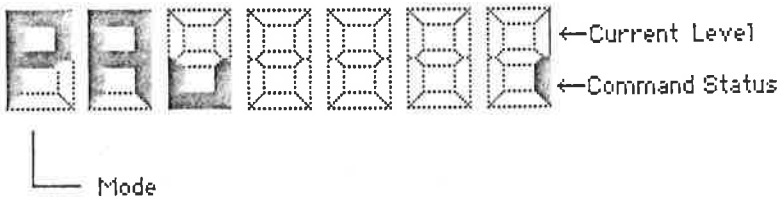


FIGURE 4-8
Play, Aux port, port 1 off

indicating that AUX 1 is turned off and the command to change state is in the current record.

The user can clear an AUX command from a record by using a three key sequence:

<AUX OUTPUT> <CLEAR> <Number 1 or 2, ON or OFF>

Clearing an AUX command from a record will also set that port to an OFF state.

OUTPUT PORTS

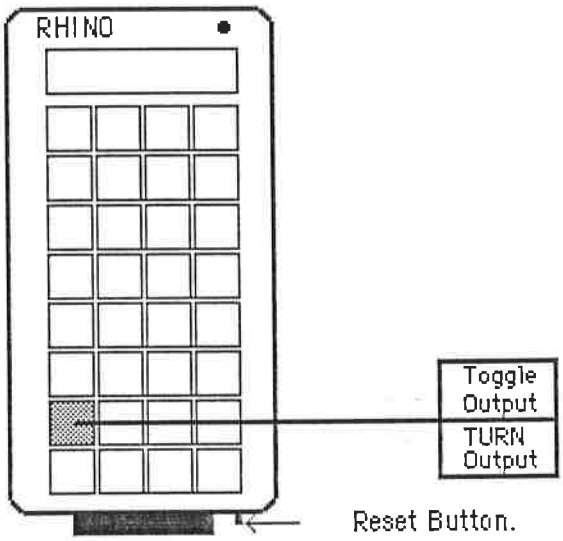


FIGURE 4-9
Keys for Output Control

The teach pendant provides two methods for controlling the 8 output ports, TOGGLE OUTPUT and TURN OUTPUT.

TOGGLE OUTPUT provides a toggled signal. It first sets the output level and then toggles the output level to the opposite state. This operation provides a pulse which can be used to latch data with external hardware or put to other uses where a momentary level is needed.

TURN OUTPUT sets the output level and then leaves it at that level.

TURN OUTPUT

Setting the state of an output line is a two key sequence, beginning with the TURN OUTPUT key. Pressing the TURN OUTPUT key changes the display to:

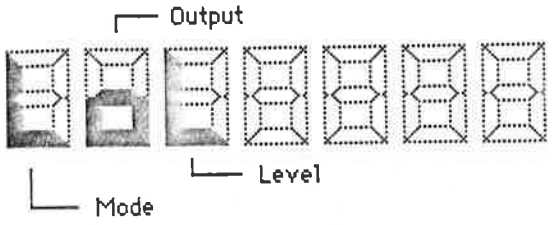


FIGURE 4-10
Learn, Output Level, waiting....

The second key determines which one of the 8 output lines is to be addressed and to what state. For example, to set output port 3 ON (low), press the 3 key under the ON column.

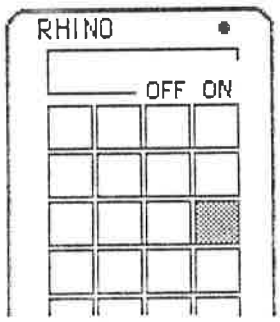


FIGURE 4-11
Key that turns output port 3 ON

The display will now show:

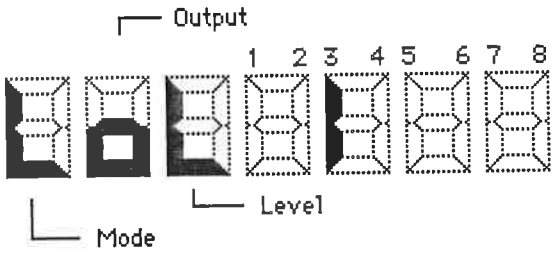


FIGURE 4-12
Learn, Output Level, Port 3 ON

If other output ports were set ON or if other output commands are active in the current record, the display will also indicate their current status.

To clear a record of an output command, press TURN OUTPUT, then press CLEAR and then press the number of the port that is to be cleared. The command will be cleared and the output will be turned OFF.

TOGGLE OUTPUT

The implementation of the TOGGLE OUTPUT command is similar to the implementation of the TURN OUTPUT command. A toggle ON command will set the output ON (low) and then set the output OFF (high) regardless of the initial state of the output. A toggle OFF command will set the output OFF (high) and then set the output ON (low) regardless of the initial state of the output. The pulse width produced by the command is approximately 2 milliseconds.

Note that if the initial state of the output is the same as the active state of the toggle, no pulse will be generated. The output will simply change state.

The clearing of a toggle command from a record is similar to clearing the TURN OUTPUT command; it involves a three key sequence. First press TOGGLE OUTPUT, then CLEAR and then the PORT NUMBER. The command will be cleared and the output will be turned OFF (high).

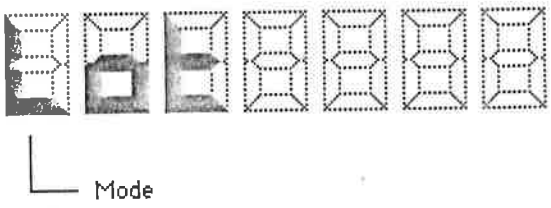


FIGURE 4-13
Output toggle function

WAIT INPUT

The WAIT INPUT function allows the system to wait (enter a holding state) until the programmed input condition is met. The system can be programmed to wait for any one of the 8 input lines to go either high or low.

When using the WAIT INPUT function, the system will wait for the programmed condition as soon as the key is pressed. No keys will be recognized until the specified input condition is met. However, the wait can be overridden by pressing the NO WAIT INPUT key. This will let the system continue from the wait condition just as if the required condition had been met. The programmed command is not canceled.

After the WAIT INPUT key has been pressed, the display will show:

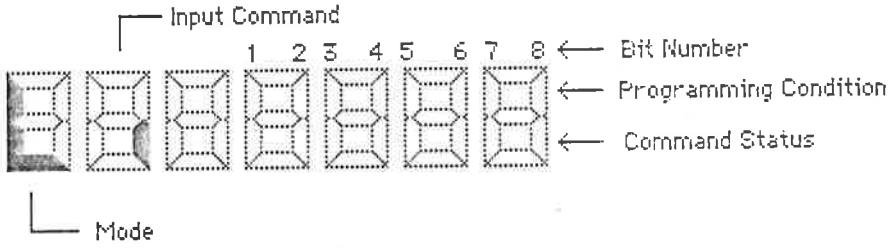


FIGURE 4-14
Learn, Input, waiting....

The upper segment of each bit column indicates the condition that the system will wait for; ON for high and OFF for low. The lower segment indicates that the command for that port is active. Only on those columns that have the lower segment ON does the upper segment have meaning.

After the WAIT INPUT key is pressed, the PORT NUMBER key is pressed using either the

key under OFF or under ON depending on the condition desired. For example, if the system is to wait for the signal on input 4 to go high and the Wait Input then 4-ON keys have been pressed, the display will show:

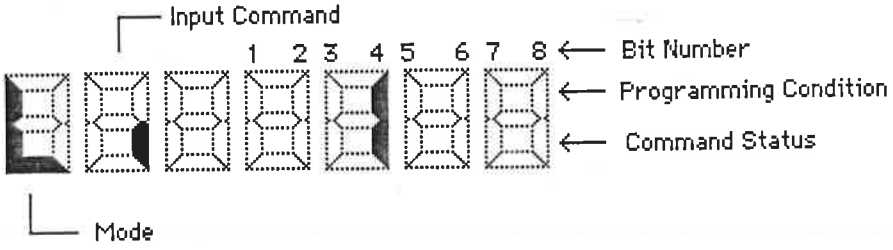


FIGURE 4-15
Learn, Input, Wait for input 4 to be high.

After the Wait on Input key has been pressed, if you want to see what the current actual inputs are, press Shift-*User key before you complete the Wait on Input key sequence. The upper segments of the right 4 digits show the current status of the inputs. Lighted segments are inputs that are ON (high).

NOTE: When programming a sequence of moves and you use the WAIT INPUT function, the input condition specified in the WAIT command must be met before the system can proceed; that is, the system is hung until the WAIT condition is satisfied. During programming, this can be overcome by pressing the NO WAIT INPUT key. No indication is given that the system is waiting for an input condition, therefore, if the system appears to be unresponsive after keying in a WAIT instruction, press NO WAIT INPUT to proceed.

PROGRAM FLOW CONTROL

The version 3.0 teach pendant system allows the user to transfer control to a record other than the one directly after the current record. This jump capability can be unconditional or conditional, based on the conditions of the input ports.

LABELS

In order to make the program branch, we have to be able to tell the program where it is to branch. This is accomplished by using labels which are attached to records that are to be used as destination points. A maximum of eight labels are available. They are labeled with the numbers 1 to 8. To affix a label to the current record, press the ***USER**

key followed by a key numbered 1 to 8. If the label that you ask for is already in use, the display will show **"uSEd"**.

Labels are branched to at the end of a current instruction.

JUMP TO PROGRAM

Jump to Program implements an unconditional branch to a labeled destination record. A two key sequence is used. First press JUMP TO PROGRAM, then the number key corresponding to the destination label. After pressing the two keys, the display will show:

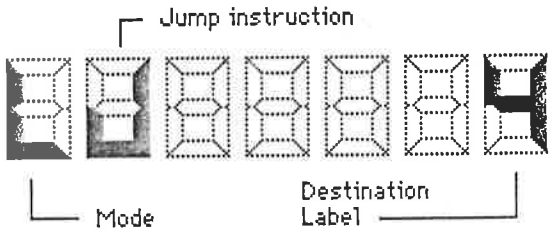


FIGURE 4-16
Learn, Jump instruction, destination = Label 4

To clear a jump instruction from the current record, press JUMP TO PROGRAM followed by CLEAR.

If an attempt is made to program a jump instruction into a program record that already has a JUMP ON INPUT instruction in it, the display will show:

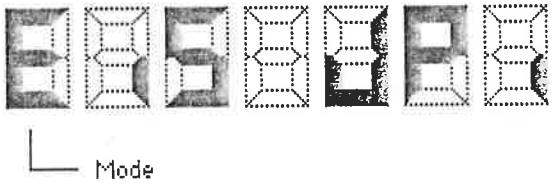


FIGURE 4-17
Edit, Input, Jump already exists in this record.

to indicate that a JUMP ON INPUT instruction already exists in this record. This instruction must be cleared before another jump instruction can be programmed.

JUMP ON INPUT

Jump on input implements a conditional branch capability based on the condition of an input signal. If the condition specified is matched, the jump takes place. If not, the execution of the program continues to the next record.

The JUMP ON INPUT instruction requires 3 parameters. The port number to respond to, the input level of the port and the destination label. The parameters are entered as follows. The first two parameters are entered with one keystroke by pressing the necessary port number under either the ON (high) or OFF (low) column of keys. The destination label is entered by pressing the number key corresponding to the label identification.

After the JUMP ON INPUT key is pressed, the display shows:

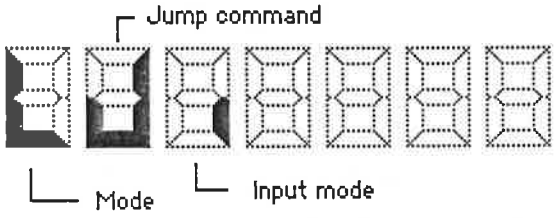


FIGURE 4-18
Learn, Jump(input mode), waiting....

Pressing the input port number ON or OFF results in a display that shows:

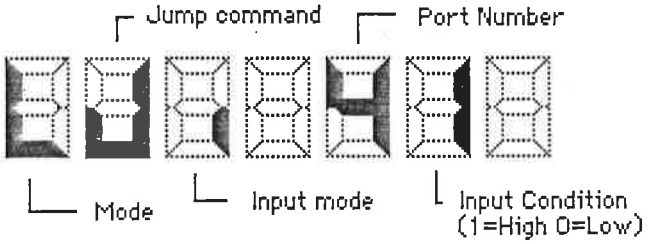


FIGURE 4-19
Learn, Jump, Input 4 high, waiting...

Pressing the destination label number, the display shows:

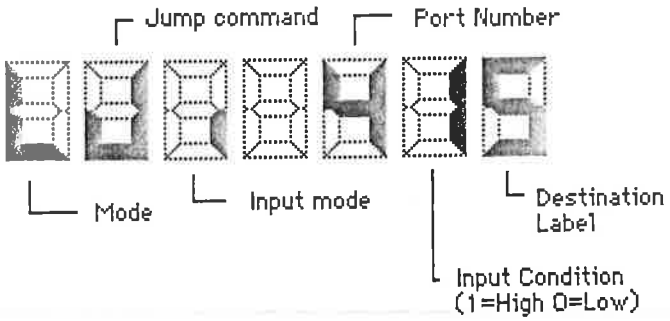


FIGURE 4-20
Learn, Jump on input 4 high, to label 5

To clear a JUMP ON INPUT command from the current record, press JUMP ON INPUT and then press CLEAR.

If an attempt is made to program a JUMP ON INPUT instruction into a program record that already has a jump instruction in it, the display will show:

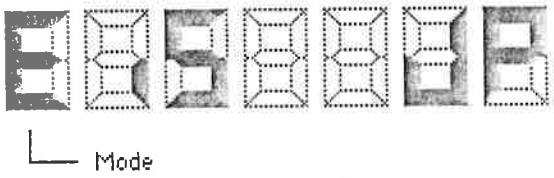


FIGURE 4-21
Edit, Input, Jump instruction already exists in this record.

to indicate that a jump instruction already exists in this record. This instruction must be cleared before another jump instruction can be programmed.

As with the Wait on Input command, you can view the current actual input levels by pressing Shift-*User.

SAVING AND LOADING

a Teach Pendant Program

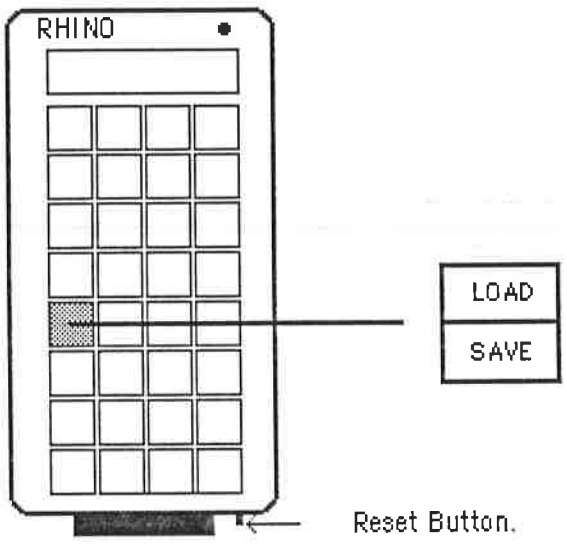


FIGURE 4-22
The LOAD and SAVE keys

SAVE

After a program has been run successfully, you may want to save it on disk for future reference. To do so, you must be in PLAY mode and then use the SAVE key. Since the XR controllers do not have disk storage capability, the information is first transferred to a host computer and then stored on the disk of the host computer. Make sure that the system is properly connected to the host computer with the proper interface. Specific instructions for use with Apple IIe and the IBM-PC can be found in Appendix II.

The host computer must be programmed to properly receive and transmit the LOAD/SAVE data in real time. The information is sent through the computer port of the Mark III controller at

9600 BAUD,
7 data bits,
2 stop bits and
even parity.

As each move is transmitted, it is counted down at the right side of the display. During the transmission to the host computer, the letter "S" will appear at the left of the teach pendant display. The "S" (send) indicates a SAVE to the host computer.

LOAD

To load a move sequence that was previously saved, you must be in the PLAY mode, have executed a hard home instruction since the last power up or reset and have no existing program in memory, ie. the memory must be cleared. During loading, the letter "L" (load) will appear at the left of the teach pendant display and the number of moves will be constantly updated at the right of the display.

If the LOAD or SAVE key is inadvertently pressed when the system is not connected to or configured for a host system, the LOAD/SAVE function can be aborted by pressing END.

THE STALL ROUTINE

When your robot cannot move the full number of steps you have given it with the move keys, the motor stalls and the word "STALL" is put into the teach pendant display followed by the stalled motor identification letter. This can occur during all modes of operation but usually occurs in PLAY, LEARN, and EDIT when the robot runs into an obstruction that stops further motion. The robot can of course, also run into its base, the table top or even into itself.

A motor stall is not a fatal situation. The system is designed so that it attempts to recover from a stall. When a stall is detected the display will change to show a "STALL" and indicate the stalled motor. You will notice that the robot reverses direction and backs away a few encoder positions from the obstacle it had run into. All you now need to do is press the END key to acknowledge the stall. The display will identify the motor on which the stall had occurred and the corrected step count for its encoder. For example, if the "D" motor had stalled at 180 steps, the count after the STALL might show D 150, because the motor had moved 30 steps by backing up.

Occasionally, a STALL occurs during the RUN mode, especially if you had not stepped through your program with the EDIT routine. Or, you might have misjudged the actual path of the robot in executing the move sequence and run into an unexpected obstruction.

If this occurs, the robot will stop and the display will show the move in which the STALL condition occurred. Press END to escape the stall trap. The display will show "PEndrun" (play mode, end run)and the system will return to the PLAY mode. The display message will clear when any key is pressed. You will have to RUN the program again from the PLAY mode after you have edited the crash out of the sequence.

The advantage provided by the STALL routine is that the system does not forget the program that the system had been taught.

If the motors cannot move as directed while the XR is doing either a Go Soft Home or a Go Hard Home, the display will show "noSoFt" or "noHArd". Remove the obstruction and re-enter the appropriate HOME command.

SECTION 5

THE KEYS AND THEIR FUNCTIONS

The teach pendant has a total of 32 keys arranged in eight rows by four columns. Each key except the Shift key serves more than one function. The keys will be described starting at the upper left corner and then working down to the bottom of the keyboard, one column at a time. Multiple key functions will be described where applicable as a part of the first key to be pressed in that sequence. Unshifted keys will be described first. Each key will be described with its function in each of the four modes of operation.

UNSHIFTED KEY FUNCTIONS

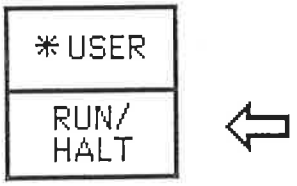
LEARN/ENTER. This key is used to enter the LEARN mode of operation for the teach pendant. Once in the LEARN mode, the key becomes the ENTER key and is used to ENTER the moves.

PLAY. Pressing this key will put the system into the LEARN mode.

LEARN. In the LEARN mode this key serves to ENTER each move into the memory of the pendant system. The key is pressed after all motors and all I/O for the move have been set to their proper positions. Every time that a move is ENTERed, the display is cleared and the step number is displayed on the right side of the display.

EDIT. Used to terminate the INSERT function; the new step is inserted in the program and control is returned to the EDIT mode.

RUN. Not used.



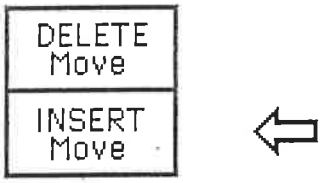
RUN/HALT. This key is used to make the robot execute the move sequence in memory. Once the system is in the RUN mode, the key becomes the HALT key and can be used to stop the operation of the robot. When the robot stops, the key becomes the RUN key again and can be used to restart the robot within the same program.

PLAY. RUN starts the robot system if there is at least one move in the program in memory. The robot executes the sequence once or until it is stopped with either the HALT key or the END key. The RUN mode can be activated from the PLAY mode only. When this key is pressed, the letter "r" appears in the left position of the display. While the robot is in the RUN mode, the right-hand digits of the display will show the number of the current move. If the current move is a DELAY, the display will so indicate.

LEARN. Not used.

EDIT. Used like the Next Move key, however all Jump instructions are enabled. This can be used as a Single Step function.

RUN. Acts as a HALT Command. Stops execution of the current move in progress and puts the robot in a HALT mode. Pressing the key again, restarts the robot and the execution of the program.



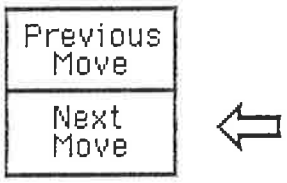
INSERT Move. This key is used in the EDIT mode only and is used to INSERT a new move into the move buffer.

PLAY. Not used.

LEARN. Not used.

EDIT. The INSERT Move key is active only in the EDIT mode. When the INSERT Move key is pressed, a move is inserted immediately **after** the last move that was executed (the current move). The current move in the buffer is moved up (provided that the move buffer is not full) and the word "InS" appears in the display. This INSERTed move is edited to be the new current move.

RUN. Not used.



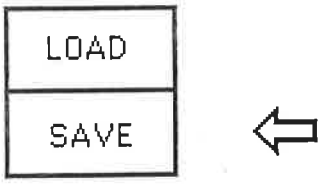
Next Move. This key is used only in the EDIT mode and allows the user to move the editor to the next record. As each record is made current, the robot is moved to the programmed position.

PLAY. Not used.

LEARN. Not used.

EDIT. While in EDIT mode, this key will move the robot to the next physical move in the program sequence. It is best to step through an entire move sequence once before RUNning a program. This will allow you to catch unexpected move and I/O conditions more easily. Jump instructions are not executed but the presence of labels is checked. If the end of memory is reached, the system will wrap around and begin execution at the first record.

RUN. Not used.



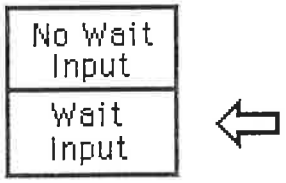
SAVE. The SAVE key allows you to transfer a sequence that you have taught the system, to a host computer so that the information can be saved on disk of the host computer. Appropriate software is provided by Rhino for the Apple IIe and the IBM-PC.

PLAY. The SAVE key is used to move the contents of the Pendant System memory to the host computer for storage on disk. During SAVE a letter "S" will appear in the left digit of the display to indicate a SAVE to the host computer. Put the diskette with the SAVE/LOAD programs on it in the disk drive of the host computer and make the appropriate responses to the system as instructions appear on the screen. Also see information in APPENDIX II.

LEARN. Not used.

EDIT. Not used.

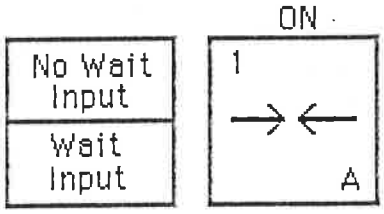
RUN. Not used.



WAIT Input. This key is used to specify that the robot/controller system is to go into a WAIT mode until the specified input line reaches the programmed state. There are 8 input lines. More than one WAIT INPUT command may be specified in a given record.

PLAY. The keypress is not useful in the PLAY mode but may be used to check the operation of an input signal. WAIT Input is a two keystroke instruction. The first key is the WAIT INPUT key. The second key is one of the 16 motor move keys on the right hand side of the pendant. There are 16 input line conditions (8 lines x 2 states) that the system can wait on. The system can be programmed to wait for the lines to go either high or low in

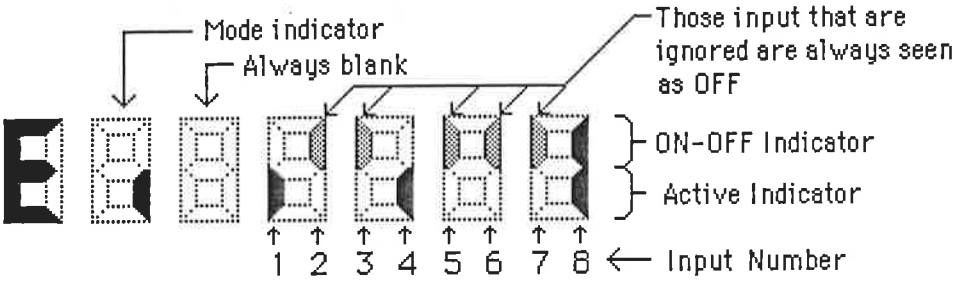
any combination. All inputs are pulled high when the system first comes on. If nothing is connected to an input, it will be seen as being ON (or high). Grounding an input drives it OFF (or low). The following key strokes would program the system to wait for input 1 to go high:



The system will be unresponsive to all key inputs until this input condition is met (except Shift-*User which shows the current actual status of all the inputs and No Wait Input which is used to override the WAIT specification. See information under No Wait Input). If the system is waiting for the programmed input condition to match the actual input level, the lower segment of the display will blink. As soon as the input conditions match, the segment will be ON steady. The segment must be ON steady before proceeding.

LEARN. See PLAY Mode.

EDIT. Wait inputs that have been programmed under the LEARN mode may be modified under the EDIT Mode. To cancel a Wait on Input command from the current record, press the Wait on Input key then CLEAR then the number of the input line you want ignored.

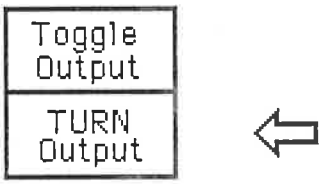


Display indicates INPUT mode
Program is waiting for 1 and 4 to be OFF and 8 to be ON.

WAIT ON INPUT EDITOR DISPLAY.

FIGURE 5-1

RUN. Not used.



TURN Output. This key is used to turn the 8 OUTPUT lines ON and OFF.

PLAY. In the PLAY Mode TURN Output turns the 8 output lines ON and OFF. TURN Output is a two keystroke command. The argument for TURN Output is one of the 16 number keys on the right hand side of the pendant. If the output line is to be turned ON, use the number key in the ON column and if an output is to be turned OFF, use the number key in the OFF column.

LEARN. Same as PLAY.

EDIT. Outputs that have been specified in the LEARN mode may be edited in the EDIT mode.

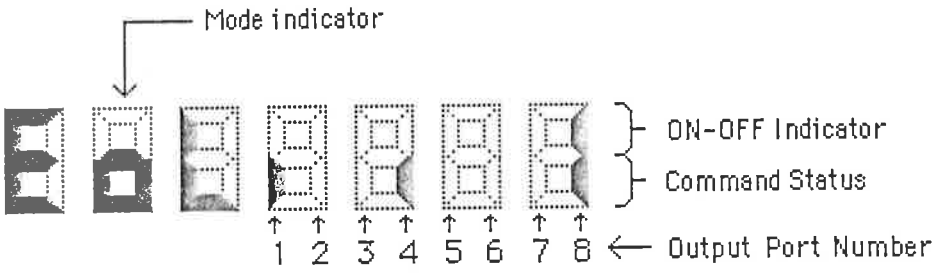
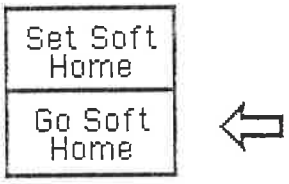


FIGURE 5-2
Editing output function

RUN. Not used.



Go Soft Home. This key sends the robot to the home position that is defined in software. This is a programmable home position as compared to the hard home position defined by the microswitch cam locations; the gripper position is not affected.

PLAY. Sends the system to it's soft home position. If the soft home position has not been defined, the robot moves to the hard home position. (Soft Home and Hard Home are defined as the same point by the system during STARTUP and RESET).

LEARN. Sends the system to the Soft Home position; the gripper is not affected. The move will not be entered till the ENTER key is pressed. This allows you to modify the I/O settings for the move before entering the move. The Soft Home position is stored as an 8 motor move and may be modified with the 16 motor move keys just like any other move.

EDIT. Sends the robot to the Soft Home position. Same operation as in the LEARN mode.

RUN. Not used.

EDIT

PLAY. Puts the system into the EDIT mode and executes the first instruction in the move buffer, provided there are moves to edit.

The EDIT mode permits the alteration of LEARNed motor moves and I/O. To enter the EDIT mode, the robot system must be in the PLAY mode. (Pressing END will bring the system to the PLAY mode from all other modes). When the EDIT key is pressed, the robot makes the first move in the move sequence, and "EStP 1" appears in the display.

By using the **NEXT MOTOR** and **PREVIOUS MOTOR** keys, it is possible to display the current settings for motors "A" to "H", the I/O etc. in the current record. The NEXT MOTOR and PREVIOUS MOTOR keys can not be used to go outside the current move. Wrap around occurs at either end.

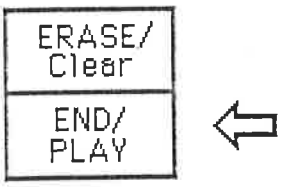
The **NEXT MOVE** and **PREVIOUS MOVE** keys allow single stepping through each program step, ignoring all jump instructions. The **RUN** key is also used for single stepping but enables all jump instructions as they would be in the RUN mode. See separate descriptions for these keys.

See Appendix for layout of Teach Pendant record storage format.

LEARN. Not used.

EDIT. Not used.

RUN. Not used.



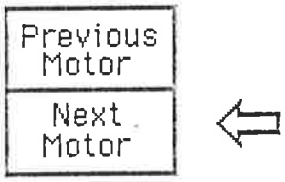
END/PLAY. This key is used to END all modes of operation and return to the PLAY mode. It is also used to acknowledge a STALL condition.

PLAY. Used only if a STALL message appears in the display and is used to acknowledge the STALL condition allowing continued operation after a STALL recovery without loss of memory or of encoder counts in the current move. The END key is also used to abort certain functions like the LOAD and SAVE routines.

LEARN. ENDs the LEARN sequence and puts the system into the PLAY mode. If your sequence contains 149 or more moves, the teach pendant display will indicate that memory is FULL and will automatically place the system into the PLAY mode.

EDIT. ENDs the EDIT sequence and puts the system into the PLAY mode.

RUN. ENDs a RUN sequence. Stops the continuous operation of the robot immediately.



Next Motor. This key is used in the EDIT mode only. It is used to step to the next motor in a move sequence. It is also used to step though the I/O and other functions that make up a move. The function is used only to display the value of each section of the current record; you need not step to that portion of the record to edit it.

PLAY. Not used.

LEARN. Not used.

EDIT. This key moves the editor's display function from the "A" motor to the "B", "C", "D", "E", "F", "G", and "H" motors one at a time, each time it is pressed. After the "H" motor is addressed, the NEXT MOTOR key moves the editor to the INPUT, OUTPUT, LEVEL, OUTPUT TOGGLE, AUXILIARY PORTS, DELAY, JUMP and finally LABEL function. If the Next Motor key is pressed again the system will wrap around to the "A" motor.

RUN. Not used.

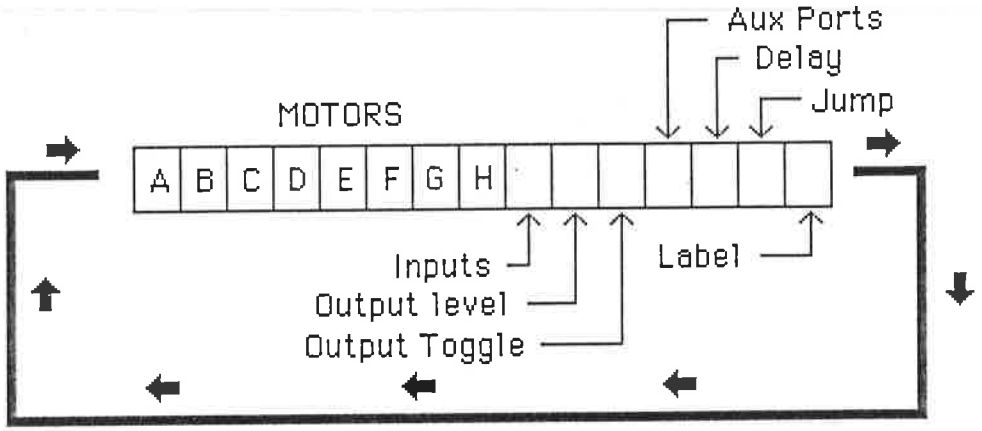
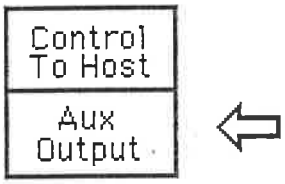
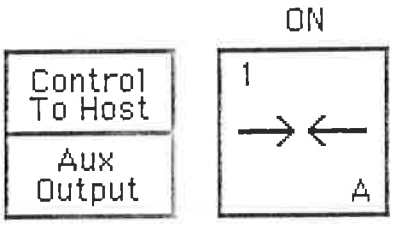


FIGURE 5-3



Aux Output. This key allows control of the 2 AUXILIARY PORTS which may be used to run unencoded motors or power other devices in a ON/OFF manner. The direction of motion of the motors (or polarity of the AUX output voltage) cannot be reversed in software but may be reversed with the switches mounted at the Aux Ports. Note: Never change the position of an AUX switch while that AUX port's power is ON.

PLAY. Aux Output is a two key sequence that allows the two Auxiliary outputs on the Mark III controller to be turned ON and OFF. The keys may be used in the PLAY mode to control the operation of the Aux Ports. In order to turn on Aux Port 1, press AUX OUTPUT followed by the 1-ON key.



LEARN. In the LEARN mode, the Aux Outputs may be turned ON or OFF at any time within the move. However, execution of the Aux commands will follow the standard sequence. See the paragraph that describes the sequence in which a move is executed. All keystrokes described in PLAY are valid except that an Aux Output cannot be toggled in a single move. In any one move it can be turned ON or OFF only.

EDIT. The status of the Aux Outputs can be edited in the EDIT mode.

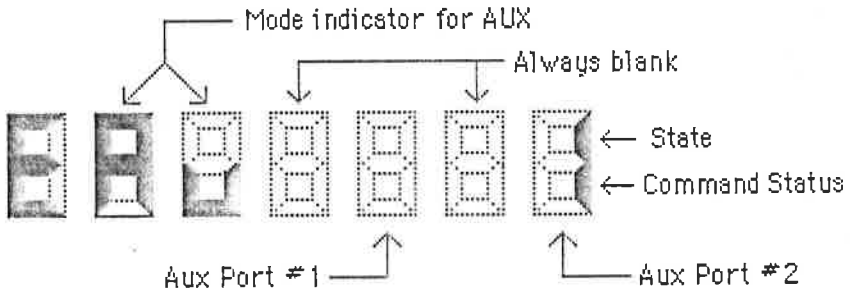
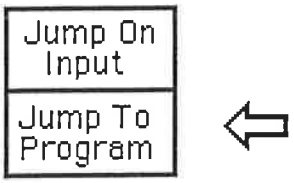


FIGURE 5-4

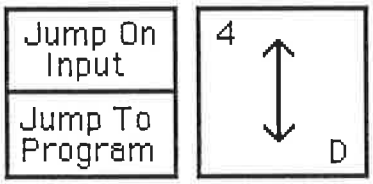
RUN. Not used.



Jump To Program. JUMP TO PROGRAM allows an unconditional branch to one of 8 labels.

PLAY. Not used.

LEARN. Up to 8 labels can reside in memory at any one time. Jump To Program is a two key function that allows the program to execute a program step that has a label attached to it. The Jump To Program key is followed by a number key.



Either the ON or the OFF number keys may be used. The key sequence to jump to label number 4 is Jump To Program followed by one of the number 4 keys.

The Jump To Program key acts like a GOTO instruction in BASIC. It is an unconditional branch instruction that will transfer execution of the program to the record with the corresponding label.

EDIT. Pressing any of the number keys changes the jump instruction destination so that the jump will be to the edited label.

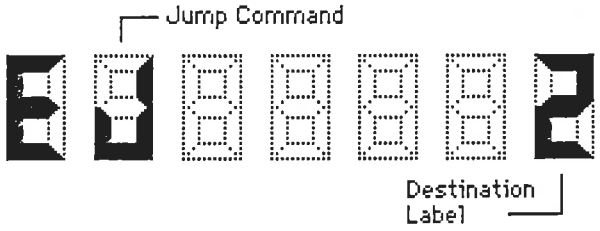
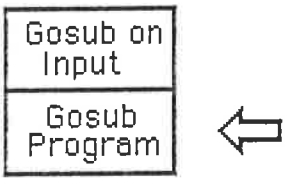
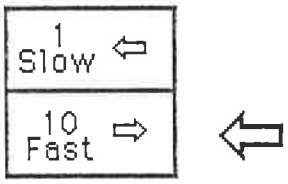


FIGURE 5-5
The Jump command

RUN. Not used.



Gosub Program. Not implemented.



10 Fast -->. The "10 Fast" key is not a programmable key; ie., it cannot be placed into a program. This key serves to set the number of steps that each motor will move for each press of a motor move key. This gives you control over the move resolution of the robot. Also see "1 Slow".

PLAY. The '10' key is the default key on power-up and reset. It is identical in operation to the '1' key except that it causes the addressed motor to move 10 encoder positions for each key press of a motor move key, and thus provides faster movement of the robot. Holding down a motor move key activates the auto-repeat feature of the keyboard and allows easy, rapid movement of the robot arm.

LEARN. Same as Play mode. Does not change the program.

EDIT. Same as Play mode. Does not change the program.

RUN. Not used.

THE 16 MOTOR MOVE KEYS.

The 16 keys on the right half of the TEACH PENDANT are the motor move keys. When the MOVE keys are being used, the length of each move may be specified by pressing the "1" or the "10" key. Holding any key down activates the AUTO REPEAT feature (after one movement of either 1 or 10 steps).

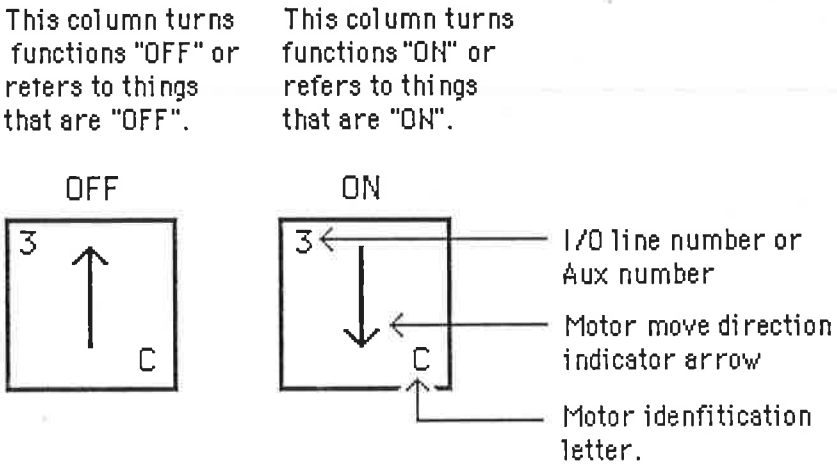


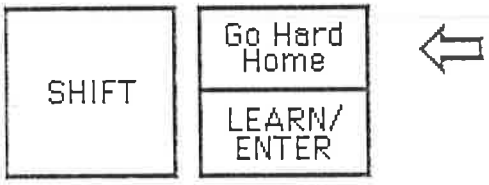
FIGURE 5-6

The "A" motor (gripper) keys operate differently from the others . There are only two positions for the gripper: "CLoSE" or "oPEn". The "CLoSE" key is inactive if the gripper is already closed. Similarly the "oPEn" key is inactive if the gripper is already open. The "CLoSE" state is reached when the motor "A" is stalled and stopped. To grasp an object with the gripper, simply press the key on the right (-><-). When you are ready to release an object, press the other gripper key (<--->). The system takes care of the rest.

When the other 14 motor move keys are pressed, the display shows the absolute position of that motor. The absolute position of a motor is the offset counted in encoder positions from the HOME position which is defined as 0 (zero). The system is constantly checking for stall conditions. If a stall occurs, the system automatically backs up the stalled motor and corrects for the stall. The word "StALL" then appears on the display. The user must press the END key to acknowledge the STALL trap before continuing to operate the robot. Also see END key description.

THE SHIFTED KEY FUNCTIONS

The SHIFTed functions are simultaneous **two-key functions that combine the SHIFT key,** on the bottom left, with another function key from the rest of the keyboard on the teach pendant. It is best to hold the pendant in the left hand so that you can keep your left thumb somewhere near the SHIFT key. The second key must be pressed and released **while the SHIFT key is held down.**



Go Hard Home. This key is used to perform the hard home function.

PLAY. This key performs the HARD HOME function. The hard home function orients the robot to its environment by positioning the robot so that the cams on the various axes are centered on the microswitches for those axes. When you ask for a hardhome routine, the robot goes to the position defined by the cams on it's five axes; ie., hand rotation cam, the shoulder cam, the wrist flex cam, the elbow cam, and the waist cam. The system sets the gripper by first CLOSEing to the stalled position and then OPENing it 80 encoder counts.

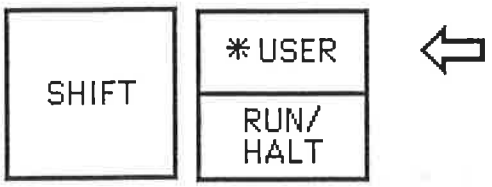
See Appendix III for hard homing the SCARA Arm.

Once the HARD HOME position has been found, the system will set the SOFT HOME position to the HARD HOME position

LEARN. Not used.

EDIT. Not used.

RUN. Not used.



*** USER.** This is a dual function key. In EDIT or LEARN, the key allows attaching a label to the current record. It is also used with the WAIT ON INPUT and JUMP ON INPUT keys to display the current input states.

PLAY. Used with the WAIT ON INPUT key to display the current input states.

LEARN. Used with the WAIT ON INPUT and JUMP ON INPUT keys to display the current input states or by itself to label the current record.

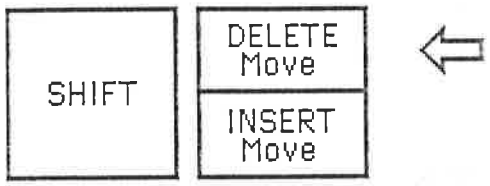
EDIT. In EDIT, the key allows the attaching of a label to the current record. First press the *USER key and then press one of the 16 numbered keys. After *USER key is pressed, the display will show:

Figure 5-7

Pressing a numbered key appends that number to the record to indicate that it is the label for that record. If a label already exists, the display will show:

Figure 5-8

RUN. Not used.



DELETE Move. DELETE Move is an EDIT function, as such it can be used in the EDIT mode only. When a move is DELETEd, the I/O information and the other functions that make up the move (all 24 bytes) are deleted. If important I/O or other information is a part of the deleted byte, this has to be recreated after the DELETE. (Also see INSERT Move)

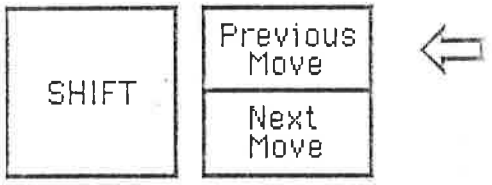
PLAY. Not used.

LEARN. Not used.

EDIT. This key is used to DELETE moves or records from memory, and is active only in the EDIT mode. The process is as follows: Upon pressing DELETE, the current move in the EDIT buffer is erased, and the word "dEL" appears in the right side of the display momentarily. The robot then executes the next move in the sequence.

The advanced user may want to experiment with compressing moves if the buffer is close to full and the move sequence is not yet complete. The user should single step through the program in the EDIT mode to see whether any moves can be combined. After selected adjacent moves are collapsed into one another, DELETE can be used to delete the redundant moves.

RUN. Not used.



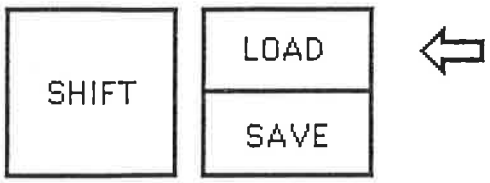
Previous Move. Previous Move works just like Next Move but moves the system in the opposite direction. Like Next Move, Previous Move can be used in the EDIT mode only.

PLAY. Not used.

LEARN. Not used.

EDIT. This key is the mirror image of Next Move. Whereas Next Move forces the system to execute the next succeding program step, the Previous Move key executes the the program step immediately preceeding the current step. It is active only during the EDIT mode.

RUN. Not used.



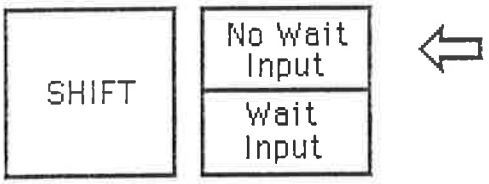
LOAD. The LOAD key allows a program that was previously recorded on the host computer disk to be loaded into the Teach Pendant system. Special software is needed at the host computer in order to use this function. The software provided by Rhino is suitable for the Apple IIe and the IBM-PC.

PLAY. This key is used to LOAD programs from the host computer. The format of data and communication set up is the same as in SAVE. During the LOAD sequence a letter "L" appears in the second digit of the display to indicate a LOAD from the host computer to the teach pendant system. As the moves are received by the system, the move count will be displayed on the right side of the display. Instructions for the SAVE/LOAD diskette and software listings are provided in Appendix II. Before LOADing, the system should have had it's HARDHOME set.

LEARN. Not used.

EDIT. Not used.

RUN. Not used.



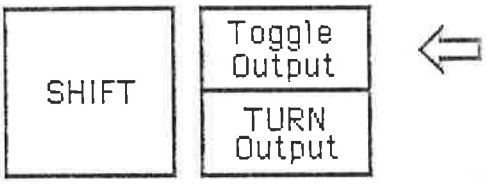
No Wait Input. NO WAIT INPUT is used to override the Wait Input command's input conditions that the system may be waiting on in the PLAY, RUN or EDIT modes.

PLAY. Overrides a WAIT condition and allows the operator to proceed to the next move.

LEARN. Same as in PLAY mode.

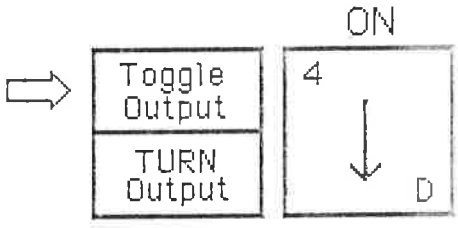
EDIT. Same as in PLAY mode.

RUN. Same as in PLAY mode.



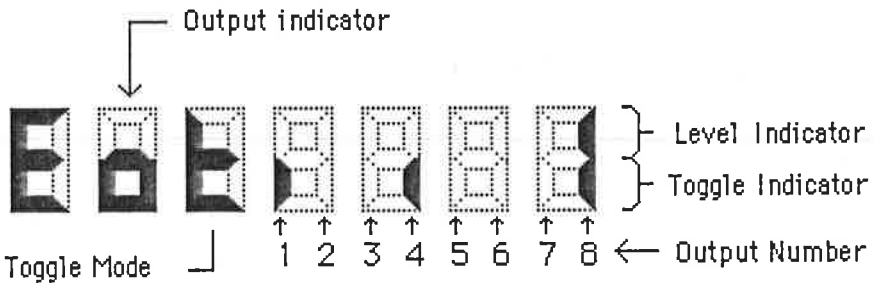
Toggle Output. Toggles one of the 8 output lines to the programmed state, then 2 milliseconds later to the opposite state.

PLAY. Toggle Output is used to toggle the specified Output Line. The toggle is to the programmed state from the present state and then back to the opposite state. Toggle Output is a two key function. The argument for the Toggle Output function is one of the number keys. The key sequence to toggle line 4 ON is: Toggle Output, 4-ON.



The state of the output changes to the programmed state for about 2 milliseconds then switches to the opposite state.

LEARN. The lines to be toggled are specified in the OUTPUT frame. More than one line may be toggled in one move.



Display indicates OUTPUT toggle mode.
Program indicates that Output 8 is toggled ON,
and outputs 1 and 4 are to be toggled OFF.

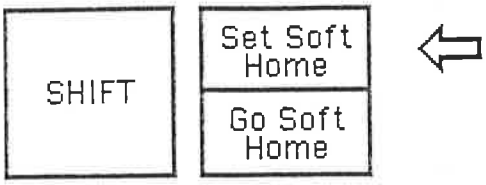
Figure 5-9

The keystrokes are similar to the keystrokes in the PLAY mode.

NOTE: If the programmed toggle level is the same as the initial state, then the toggle function will appear as only a state change, there will be no pulse output.

EDIT. Toggled Outputs may be edited in the EDIT mode.

RUN. Not used.



Set Soft Home. Sets the current position of the robot as the SOFT HOME or software home position for the robot. Soft Home is a convenient, programmable position of the robot that can be accessed with the Go Soft Home key.

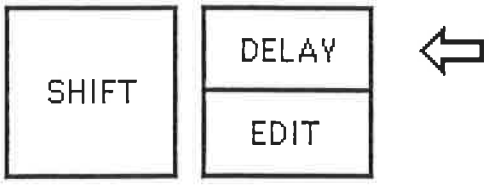
PLAY. The SET SOFT HOME key sets the current position of the robot as the SOFT HOME or software home position for the robot.
SOFT HOME position can be the same as HARD HOME or can be established at any point selected by the user. It is the position from which the user would normally begin to teach a move sequence.

The SOFT HOME position will remain defined until the power is turned off, the reset button is pressed or a new SOFT HOME position is defined. Before SOFT HOME can be defined, a HARD HOME (see GO HARD HOME) routine must have been performed. When the initial (after startup or on reset) HARD HOME routine is performed, it sets SOFT HOME to the HARD HOME position. When the new SOFT HOME position is reached, the Display will show "AtSoFt"

LEARN. Soft Home can be set in the Learn mode. When this is done, the old Soft Home Position is replaced with the new Soft Home Position and all subsequent requests to Go Soft Home will send the robot to the new location.

EDIT. The Soft Home position is seen as just another position of the robot when EDITed. All aspects of the move may be Edited just like one would edit any other move.

RUN. Not used.



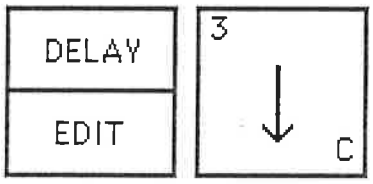
DELAY. Introduces a DELAY of up to 3 seconds into the move sequence.

PLAY. Not used.

LEARN. The DELAY key is used in the LEARN mode to enter a DELAY into the move sequence. The DELAY key is followed by any number key from 1 to 3 which represents the length of delay in seconds. Delays up to 3 seconds may be specified in one move. Longer DELAYs can be entered by using as many moves as may be needed.

RUNning a move sequence with multiple delays can make the system appear to be "hung up". However, the appearance of the word "dLy" in the display tells the user that the system is executing a delay. In the RUN mode, the "dLy" message is followed by the programmed delay length.

DELAY is a two keystroke function. The argument for the DELAY function is a number key from 1 to 3. The value of the number key specifies the length of the delay in whole seconds, for a maximum DELAY in one move of 3 seconds.

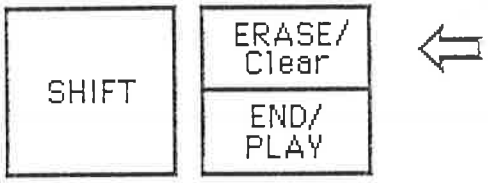


The key sequence to specify a DELAY of 3 second is: DELAY, 3-ON.

Figure 5-10

EDIT. The length of a DELAY may be edited in the EDIT mode. The delay may be modified by changing the number in the display screen by pressing one of the number keys. If the DELAY is to be eliminated completely, press the DELAY key then the ERASE/Clear key.

RUN. Not used.



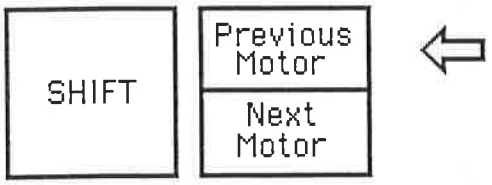
ERASE/Clear. This key is used to clear the memory of the system and to clear DELAYs, Outputs, and Input Wait conditions and Jumps. Each use depends on the mode that the system is in and the display screen that is active within that mode. Most commands can be cleared from a record by pressing the function then the CLEAR key.

PLAY. Clears the memory of the teach pendant system erasing the current program. It is used before you first enter the LEARN mode to make sure that a sequence will not contain any unplanned moves. It can be also be used any time that the pendant memory needs to be erased such as prior to a LOAD.

LEARN. Not used.

EDIT. Can only be used after a function key is pressed, for example if a record has a command to set output 2 ON (low), pressing TURN OUTPUT, CLEAR then 2 will remove the command and resets the output high.

RUN. Not used.



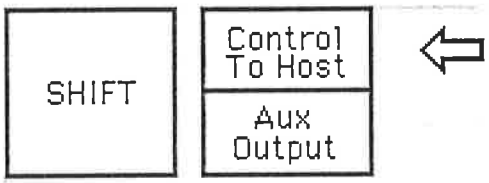
Previous Motor. This key and the Next Motor key are used to display each record segment when Editing a move sequence.

PLAY. Not used.

LEARN. Not used.

EDIT. The PREVIOUS MOTOR key is identical to the NEXT MOTOR key except that the motor moves are accessed in the reverse direction from "H" to "A". After the "A" motor display the system wraps around to display the other functions just like the NEXT MOTOR key function does.

RUN. Not used.



Control To Host. Transfers control of the Teach Pendant system to a host computer.

PLAY. When CONTROL to HOST is pressed, all subsequent control of the system is from the host computer. The host computer becomes the hand held keyboard and sends the key pad codes that match the numbers of the key pad positions. Control is transferred back to the hand held keyboard when the host computer sends a hexidecimal 40 or decimal 64. If the CONTROL TO HOST key was inadvertently pressed, the function can be aborted by pressing the END key.

LEARN. Not used.

EDIT. Not used.

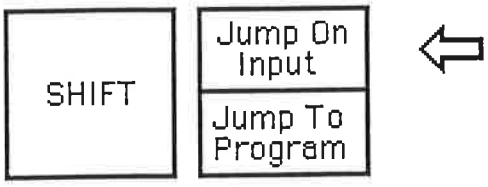
RUN. Not used.

The hexidecimal key numbers are illustrated as follows:

47 27	4F 2F	57 30	5F 38
46 26	4E 2E	56 31	5E 39
45 25	4D 2D	55 32	5D 3A
44 24	4C 2C	54 33	5C 3B
43 23	4B 2B	53 34	5B 3C
42 22	4A 2A	52 35	5A 3D
41 21	49 29	51 36	59 3E
SHIFT KEY	48 28	50 37	58 3F

HEXIDECIMAL KEY CODES

(NO KEY PRESS = 20 HEX)



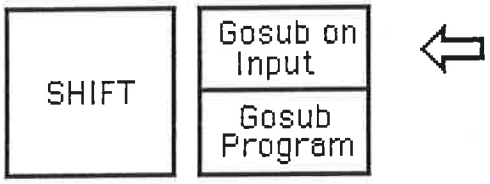
Jump On Input. This is a conditional jump command that uses the condition of the Input lines as an argument. The jump is to one of 8 labeled records in memory. JUMP ON INPUT operates the same as JUMP TO PROGRAM with regards to executing a new move and the same as WAIT ON INPUT with regard to meeting input conditions.

PLAY. Not used.

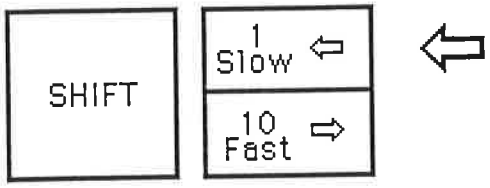
LEARN. The Jump on Input function is a three key function. The first key is the Jump On Input key and the second key is one of the 16 number keys. Jump can be conditional on either a high input or a low input. Use the ON numbers for a high input requirement and an OFF number for a low input requirement. The third key is one of the 8 numbered keys and represents the destination label.

EDIT. The JUMP ON INPUT can be edited in the same manner as the WAIT ON INPUT and the jump commands. Canceling of the command is accomplished by pressing JUMP ON INPUT then CLEAR.

RUN. Not used.



Gosub On Input. Not implemented.



1 Slow <---. 1 Slow is not a programmable key. It determines only whether a move of 1 encoder position or 10 encoder positions will be issued to the robot with each press of a motor move key. **1 Slow** allows you to position the robot more accurately by giving you finer control over the system. There is no effect on the system when in the RUN mode.

PLAY. The '1' key will translate each press of the 14 move keys (see MOVE KEYS) into one encoder position move of the addressed motor (the 1 and 10 keys do not affect the A motor). The robot moves more slowly than in the 10 Fast mode. This is useful for fine robot control. Although the robot moves slowly during positioning, it will move with normal speed when the move sequence is RUN.

LEARN. The "1" key may be used in the LEARN mode to give the operator finer control over the operation of the robot. Although the robot will now move slowly during LEARN, it will move with normal speed when the move sequence is RUN.

EDIT. Same as Learn Mode.

RUN. Not used.

SECTION 6

PENDANT DISPLAYS

The little round, red indicator in the upper right hand corner of the pendant keyboard indicates that power is available at the pendant.

The following pages show the list of all messages that will appear on the display when some specific event takes place:

PLAY MODE

LEARN MODE

EDIT MODE

RUN MODE

GOING TO SOFT HOME

AT SOFT HOME

FAILED TO GET TO SOFT HOME

GOING TO HARD HOME

AT HARD HOME

FAILED TO GET TO HARD HOME

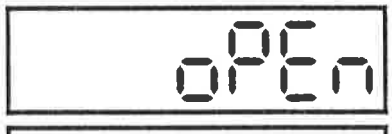
HOST CONTROL

TEACH PENDANT CONTROL

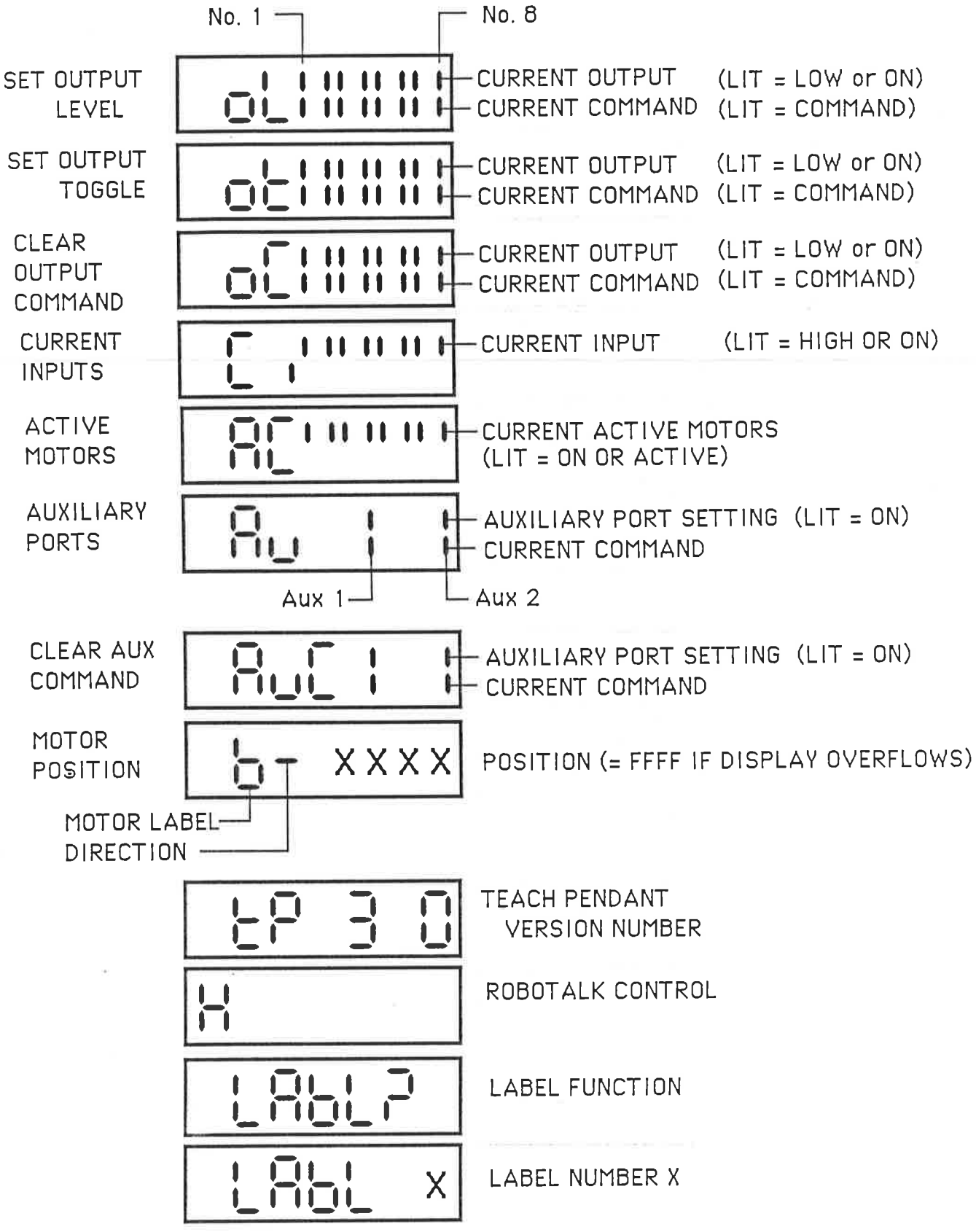
MOTOR POSITION RANGE ERROR

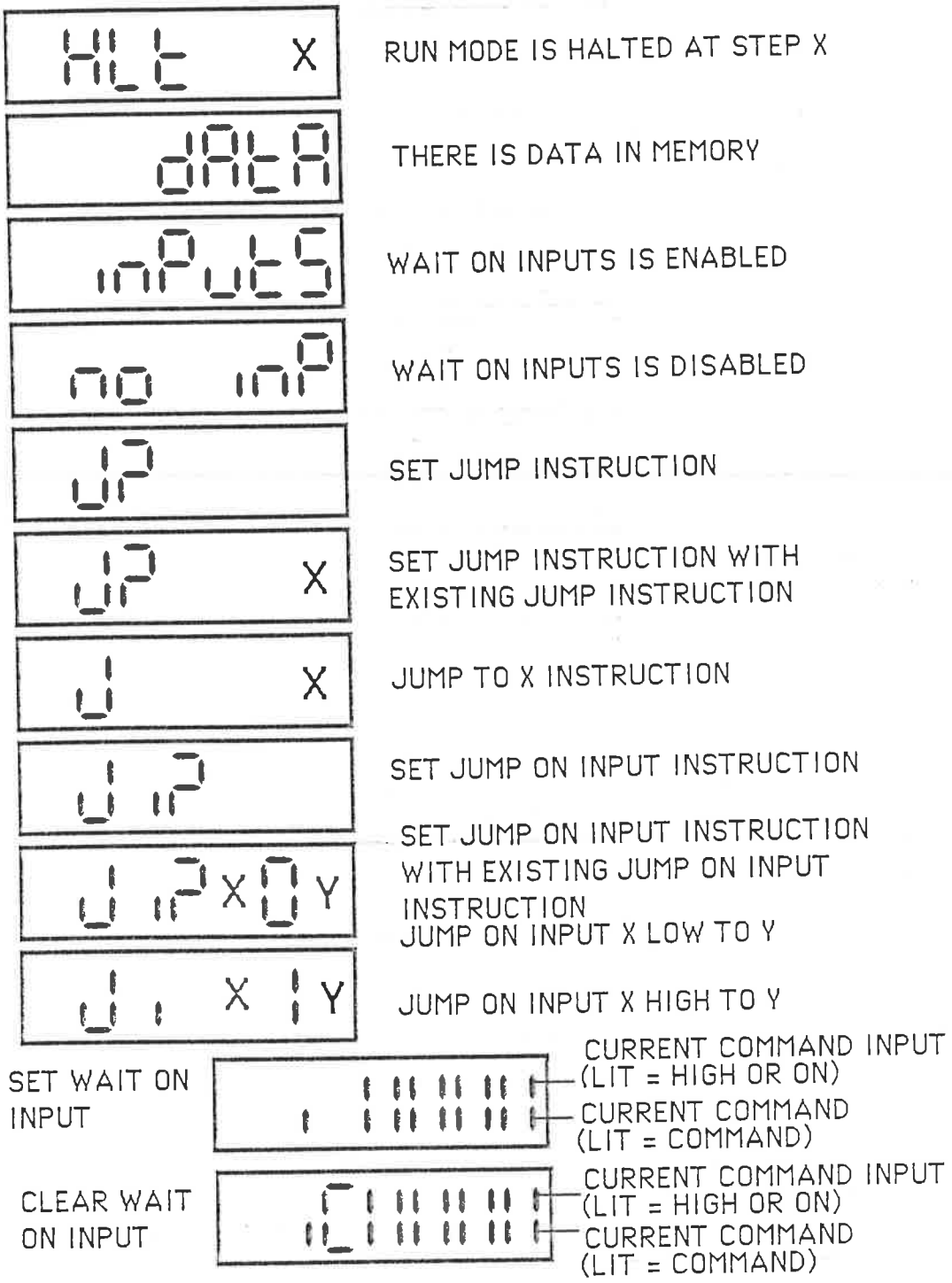
MOTOR POSITION RANGE ERROR

MOTOR OFF

	END EDIT MODE
	END RUN MODE
	NO DATA IN MEMORY
	PRESS ENTER
	MUST BE IN PLAY MODE
	MUST BE IN EDIT MODE
	ERROR or INVALID KEY
	MEMORY IS FULL
	SOFT HOME IS SET
	LABEL IS USED
	I/O IS INACTIVE OR UNAVAILABLE
	I/O IS ACTIVE OR AVAILABLE
	HAND IS CLOSED OR CLOSING
	HAND IS OPEN OR OPENING
	ERASE MEMORY

noLAbL	MISSING LABEL
tESt	TEST OR DEMO PROGRAM
in it	SYSTEM IS INITIALIZING
iS JP	IS A JUMP INSTRUCTION
iS JP i	IS A JUMP ON INPUT INSTRUCTION
r5232	UPLOAD/DOWNLOAD RS-232 ERROR
doHArd	DO A HARD HOME
StALL X	MOTOR X STALLED
dLy X	DELAY X (X=0,1,2,3)
inS XXX	INSERTING STEP NUMBER XXX
Ent XXX	ENTERING STEP NUMBER XXX
StP XXX	EXECUTING OR LEARNING STEP NUMBER XXX
bSt XXX	EXECUTING PREVIOUS STEP NUMBER XXX
dEL XXX	DELETING STEP NUMBER XXX
inC XXX	MOTOR INCREMENT VALUE = XXX (XXX=1,10 OR 100)


No. 1
No. 8
SET OUTPUT LEVEL
CURRENT OUTPUT (LIT = LOW or ON)
CURRENT COMMAND (LIT = COMMAND)
SET OUTPUT TOGGLE
CURRENT OUTPUT (LIT = LOW or ON)
CURRENT COMMAND (LIT = COMMAND)
CLEAR OUTPUT COMMAND
CURRENT OUTPUT (LIT = LOW or ON)
CURRENT COMMAND (LIT = COMMAND)
CURRENT INPUTS
CURRENT INPUT (LIT = HIGH OR ON)
ACTIVE MOTORS
CURRENT ACTIVE MOTORS
(LIT = ON OR ACTIVE)
AUXILIARY PORTS
AUXILIARY PORT SETTING (LIT = ON)
CURRENT COMMAND
Aux 1
Aux 2
CLEAR AUX COMMAND
AUXILIARY PORT SETTING (LIT = ON)
CURRENT COMMAND
MOTOR POSITION
XXXX
POSITION (= FFFF IF DISPLAY OVERFLOWS)
MOTOR LABEL
DIRECTION
TEACH PENDANT VERSION NUMBER
ROBOTALK CONTROL
LABEL FUNCTION
LABEL NUMBER X


HLt X
RUN MODE IS HALTED AT STEP X
dAtA
THERE IS DATA IN MEMORY
inPutS
WAIT ON INPUTS IS ENABLED
no inP
WAIT ON INPUTS IS DISABLED
JP
SET JUMP INSTRUCTION
JP X
SET JUMP INSTRUCTION WITH EXISTING JUMP INSTRUCTION
J X
JUMP TO X INSTRUCTION
J inP
SET JUMP ON INPUT INSTRUCTION
J inP X 0 Y
SET JUMP ON INPUT INSTRUCTION WITH EXISTING JUMP ON INPUT INSTRUCTION
JUMP ON INPUT X LOW TO Y
J X Y
JUMP ON INPUT X HIGH TO Y
SET WAIT ON INPUT
CURRENT COMMAND INPUT (LIT = HIGH OR ON)
CURRENT COMMAND (LIT = COMMAND)
CLEAR WAIT ON INPUT
CURRENT COMMAND INPUT (LIT = HIGH OR ON)
CURRENT COMMAND (LIT = COMMAND)

APPENDIX I

THE MOVE BUFFER DATA FORMAT

Each move is stored as a group of 24 bytes called a **RECORD**. These 24 bytes contain all the information to fully describe the status of the Mark III controller. A record is generated whenever the **ENTER** key is pressed during a Learn or Edit session. A maximum of 149 records can be programmed in a given Teach Pendant file.

The content of a record is initially set in the LEARN mode and can be modified in the EDIT mode. Individual variables within a record are accessed through the use of the NEXT MOTOR and PREVIOUS MOTOR functions (see chapter on Editing). The order in which the variables are accessed is as follows:

1. A motor - Gripper
2. B motor - Wrist Rotate
3. C motor - Hand Azimuth
4. D motor - Elbow
5. E motor - Shoulder
6. F motor - Waist
7. Wait on Inputs
8. Output level
9. Output toggle
10. Auxiliary Motor ports
11. Delay
12. Jump instructions
13. Labels

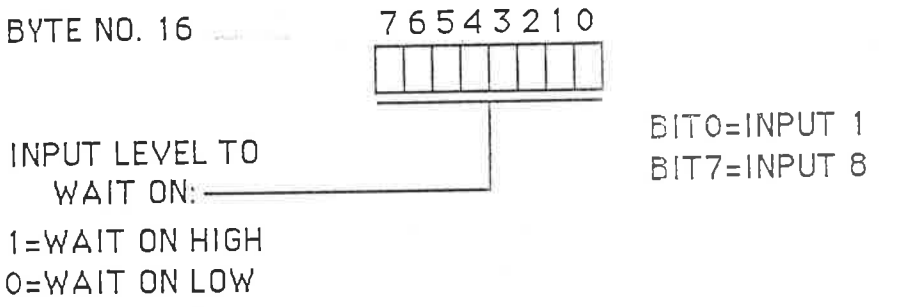
Each record is stored as a binary byte string stored contiguously with the other records comprising a move sequence or Teach pendant program. The absolute positions of motors B through H are stored as 16 bit (2 byte) values in two's compliment format with zero defined as the hard home position. The 24 bytes of a record are defined as follows:

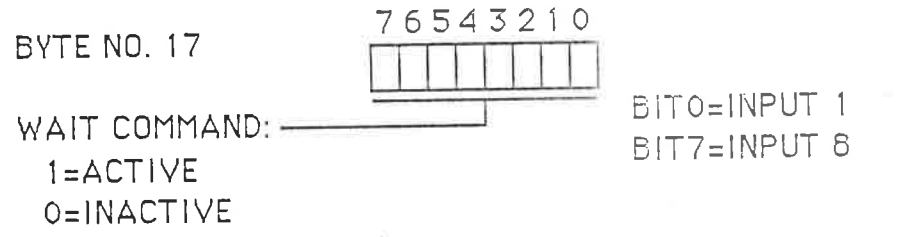
1.	Byte 0:	Unused
2.	Byte 1:	B motor low byte
3.	Byte 2:	C motor low byte
4.	Byte 3:	D motor low byte
5.	Byte 4:	E motor low byte
6.	Byte 5:	F motor low byte
7.	Byte 6:	G motor low byte
8.	Byte 7:	H motor low byte
9.	Byte 8:	Gripper control
10.	Byte 9:	B motor high byte
11.	Byte 10:	C motor high byte
12.	Byte 11:	D motor high byte
13.	Byte 12:	E motor high byte
14.	Byte 13:	F motor high byte
15.	Byte 14:	G motor high byte
16.	Byte 15:	H motor high byte
17.	Byte 16:	Wait On Input Level
18.	Byte 17:	Wait On Input Command Status
19.	Byte 18:	Output Level setting
20.	Byte 19:	Output Toggle Command Status
21.	Byte 20:	Delay Value
		Auxiliary Command Status
		Auxiliary Port level
		Jump On Input level
22.	Byte 21:	Jump destination label
		Jump Command Status
		Jump On Input Command Status
23.	Byte 22:	Record Label
		Jump On Input source port
24.	Byte 23:	Output Command Status

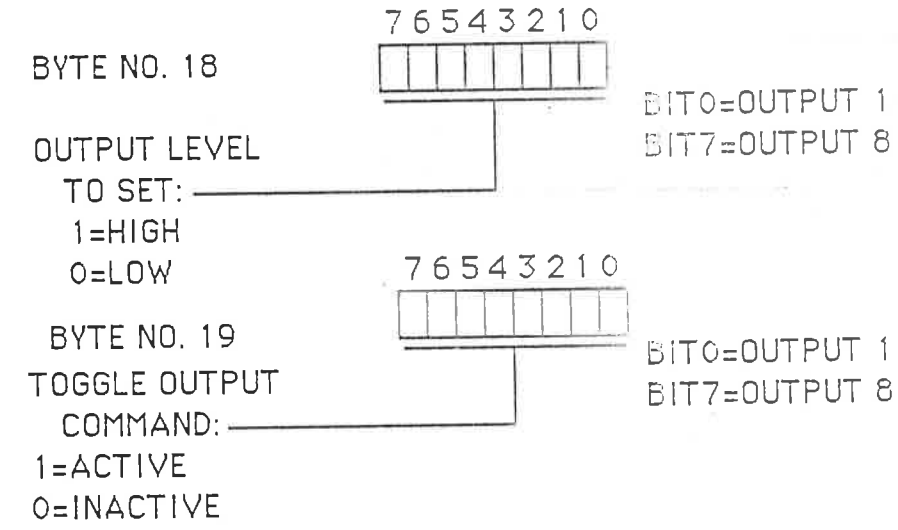
The following diagrams give a pictorial representation of the function of each bit within the 24 byte record. Each byte is stored as shown contiguously within a record and each record is stored contiguously within a Teach Pendant program. When the file is being sent to or from a host computer for Loading/Saving, each byte is divided into two nibbles for transmission. The nibbles are rejoined at either end for storage.

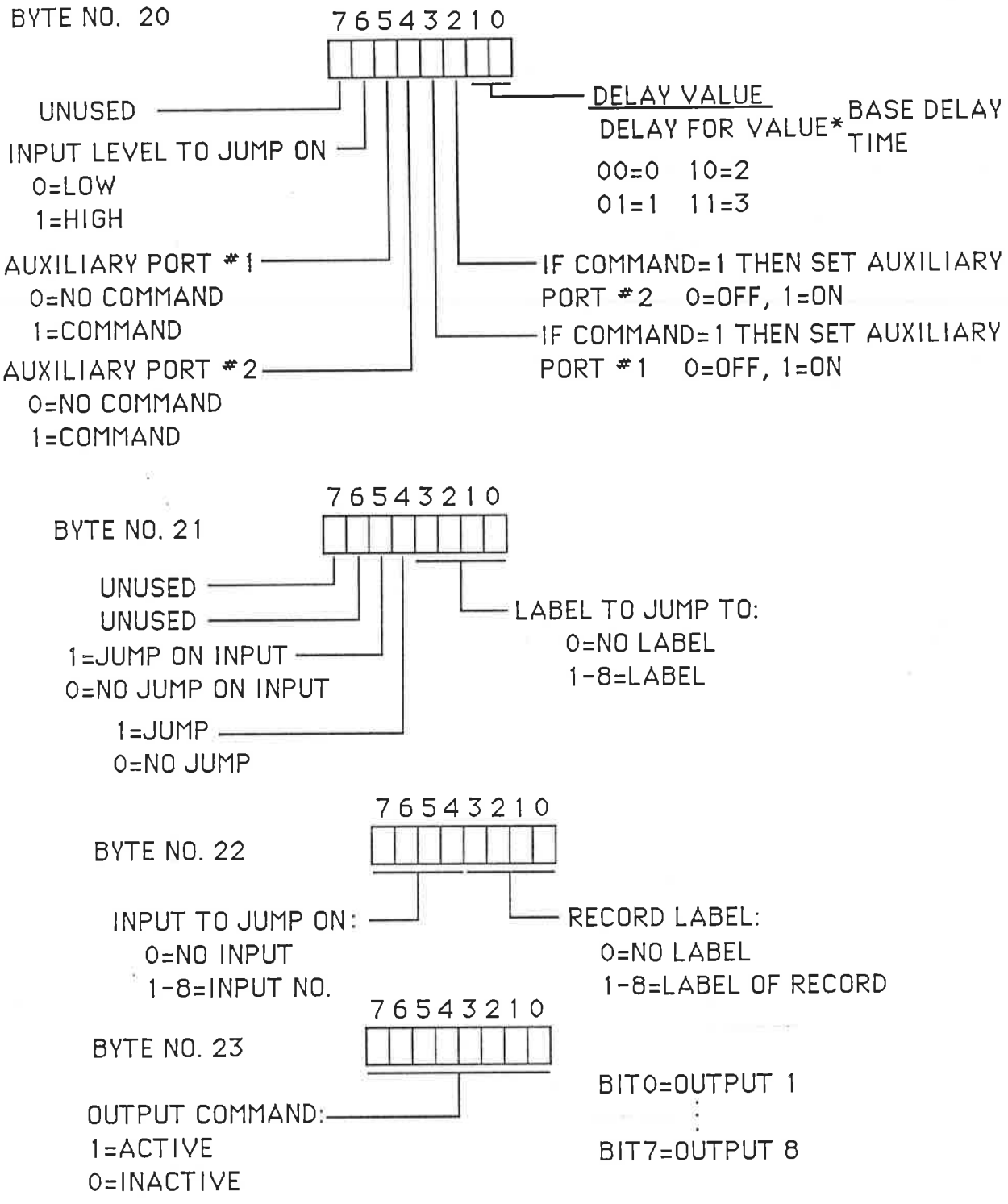
BYTE NO. 0: UNUSED
BYTE NO. 8: GRIPPER 0=NO COMMAND
$4F=79="O"=OPEN HAND
$43=67="C"=CLOSE HAND

BYTE NO. 1: B MOTOR LOW BYTE	BYTE NO. 9: B MOTOR HIGH BYTE
BYTE NO. 2: C MOTOR LOW BYTE	BYTE NO. 10: C MOTOR HIGH BYTE
BYTE NO. 3: D MOTOR LOW BYTE	BYTE NO. 11: D MOTOR HIGH BYTE
BYTE NO. 4: E MOTOR LOW BYTE	BYTE NO. 12: E MOTOR HIGH BYTE
BYTE NO. 5: F MOTOR LOW BYTE	BYTE NO. 13: F MOTOR HIGH BYTE
BYTE NO. 6: G MOTOR LOW BYTE	BYTE NO. 14: G MOTOR HIGH BYTE
BYTE NO. 7: H MOTOR LOW BYTE	BYTE NO. 15: H MOTOR HIGH BYTE








BYTE NO. 20
7 6 5 4 3 2 1 0
UNUSED
INPUT LEVEL TO JUMP ON
0=LOW
1=HIGH
AUXILIARY PORT #1
0=NO COMMAND
1=COMMAND
AUXILIARY PORT #2
0=NO COMMAND
1=COMMAND
DELAY VALUE
DELAY FOR VALUE*
00=0 10=2
01=1 11=3
BASE DELAY TIME
IF COMMAND=1 THEN SET AUXILIARY PORT #2 0=OFF, 1=ON
IF COMMAND=1 THEN SET AUXILIARY PORT #1 0=OFF, 1=ON
BYTE NO. 21
7 6 5 4 3 2 1 0
UNUSED
UNUSED
1=JUMP ON INPUT
0=NO JUMP ON INPUT
1=JUMP
0=NO JUMP
LABEL TO JUMP TO:
0=NO LABEL
1-8=LABEL
BYTE NO. 22
7 6 5 4 3 2 1 0
INPUT TO JUMP ON:
0=NO INPUT
1-8=INPUT NO.
RECORD LABEL:
0=NO LABEL
1-8=LABEL OF RECORD
BYTE NO. 23
7 6 5 4 3 2 1 0
OUTPUT COMMAND:
1=ACTIVE
0=INACTIVE
BIT0=OUTPUT 1
BIT7=OUTPUT 8

tp3 hibyte

SEQUENCE OF CONTROLLER FUNCTIONS

When the Teach Pendant is running a preprogrammed sequence of moves, each record is executed in order. Within each record, each function is executed in a fixed sequence. This sequence is as follows:

1. The system waits for a match of Input Lines if a Wait On Input command is active.

2. The Gripper is opened or closed if programmed.

3. Each of the motors is moved to the absolute position indicated by the motor count for that motor.

4. Any programmed Delay is executed.

5. The system executes the Output Toggle or Output Level commands.

6. The Auxiliary motor ports are set to a level.

7. The system jumps to a new record if programmed by a Jump to Program or Jump On Input command.

APPENDIX II

INSTRUCTIONS FOR THE "SAVE" AND "LOAD" PROGRAM

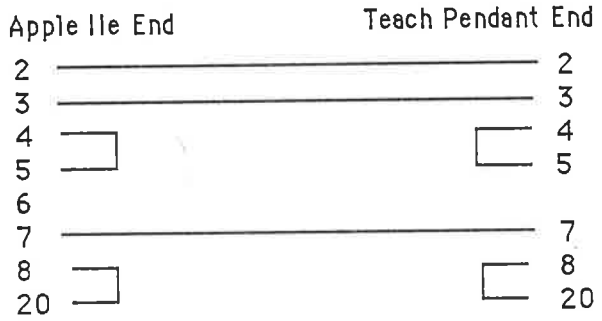
A diskette is provided for the SAVE/LOAD of programs from the teach pendant. A separate disk is available with software on it for each of the following personal computers:

1. APPLE IIe
2. IBM PC

A description of the cable required for SAVEing or LOADing, and instructions to operate the program for each computer follow:

APPLE IIe

The cable required for using the APPLE IIe program is similar to the RS-232C cable that is used to interface the teach pendant to the XR series controller. The cable connections are as follows:



INSTRUCTIONS:

1. Turn off your computer and install your Super Serial card in slot #2 of the Apple IIe.

2. Connect the cable between the controller's computer port and the Super Serial card connector.

3. Place the Mark III controller mode switch in the TEACH PENDANT position.

4. Insert the provided disk in the drive 1, and turn on the power. (Boot up the system).

4. Turn on the Mark III Controller.

5. Press the Mark III reset switch.

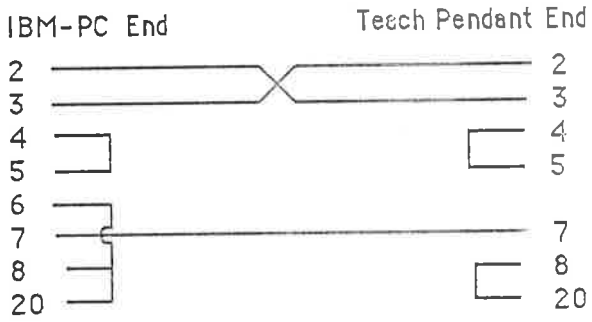
7. Specify the type of the card that you installed in slot #2. (The program requires that 1 of the 3 specified cards be used.)

8. Specify the action (Save/Load) that is required.

9. Follow the program instructions to SAVE or LOAD.

IBM PC

The cable required for using the IBM PC has to have the following configuration:



Note: The IBM adapter provided with the Mark III controller performs the 2-3 crossover required for the IBM-PC. The Apple cable is used for both the Apple IIe and the IBM-PC.

INSTRUCTIONS:

1. Connect the Rhino data cable between the controller's computer port and the serial port of the IBM PC using the IBM Adapter plug on the IBM side of the cable. The serial Asynchronous Communications Card is required.

2. Place the Mark III Mode switch in the TEACH PENDANT position.

3. Turn on the Mark III controller and press the RESET switch.

4. Insert a PC-DOS diskette in drive A. Insert the SAVE/LOAD diskette provided in drive B. Turn on the power and system will be booted up.

5. Enter the current date and time as requested.

6. Type B:<return> to make drive B the default drive.
   Type A:BASICA UPDOWN<return> to run the program.

7. Specify the action (SAVE/LOAD) that is required.

8. Follow the program instructions to SAVE or LOAD.

Note: The program automatically appends a '.MOV' extension to all the filenames which are entered by the user. Therefore the user should enter only the first part of the filename.

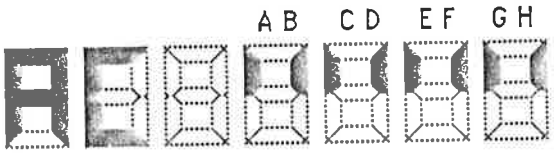
APPENDIX III

SPECIAL FUNCTIONS

A number of functions exist which are not labeled on the Teach Pendant hand unit keypad. Each function is enabled by pressing the SHIFT key and the listed key. The following is a listing of those functions.

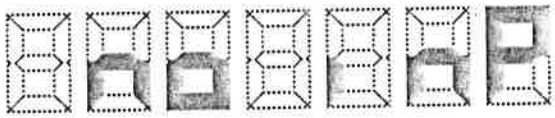
100 Distance key - Like the 1 and 10 Distance functions, pressing the 100 function commands the the system to move 100 encoder positions each time a motor move key is pressed. This function is enabled by pressing SHIFT and the 8-OFF key.

Motor Status - The status of each of the eight motors can be interrogated or changed by the use of this function. Pressing SHIFT and the 2-ON key displays:



Pressing a motor key under the OFF or ON column will turn off or on that motor. Pressing SHIFT and 2-ON immediately after the initial pressing will abort the function and leave all motors in their initial state. NOTE: Activating a motor that is not connected may cause the system to hang if that motor is given a command to move.

Global Disable of Wait on Inputs - This function enables/disables all programmed Wait on Input statements. The default state on power up or reset is **ENABLED**. When activated by pressing the SHIFT key and the 6-OFF key the state of this function is displayed as follows:



Disabled

or

Enabled

Pressing any OFF key will Disable the Wait function and pressing any ON key will enable the function. If the function key is pressed immediately after the initial press, the function is aborted and remains in its initial state.

SCARA MODE - On power up, the teach pendant system defaults to the XR robot mode which assumes that the controller is connected to the Rhino Robot 5 axis arm. However, the system can be configured to properly hard home and operate the Rhino SCARA arm. Before a hard home has been set, press SHIFT and 1-OFF which will cause the display to show: SCArA?. Pressing any ON key at this time will cause the system to enter the SCARA mode. If any OFF key is pressed, the system will return to the XR mode.

In the SCARA mode, the controller will execute a different hard home routine. It is for this reason that the SCARA mode cannot be entered if a hard home has been set. If a hard home has been set and you wish to change from one mode to the other, press RESET and then press SHIFT and 1-OFF followed by an ON key.

The only other difference between the two modes is the Demonstration Program. The SCARA's demo program is written to more effectively show the capabilities of the SCARA design. The SCARA demo program is invoked in the same manner as the XR program as shown below.

XR-2 MODE - The version 3.x version of the Teach Pendant operating system is configured for XR-3 robot. Because of differences in the gripper motor and its encoder, the XR-2 robot's gripper will not fully open when given an OPEN command. After a HARDHOME is completed the user may configure the Teach Pendant system to support the XR-2 gripper by pressing SHIFT and 7-ON. A return to the XR-3 mode is accomplished by pressing SHIFT and 7-OFF. Upon power up and after reset the system defaults to the XR-3 mode.

Teach Pendant Software Version - The version number of the software currently being executed can be displayed by pressing SHIFT and 8-ON. The display will show:

The last digit displayed on your system, in the illustration a zero, may differ depending on

the current release of the Teach pendant operating system.

Demonstration Program - A small Teach Pendant program resides in firmware to demonstrate the robot movements and provide a convenient program file to study the editing functions. Two versions are available, one for the XR and one for the SCARA. Depending on the current robot type, the appropriate version of the DEMO program will be loaded.

Enabling this function requires that the robot has had it hard home position programmed and that the present memory be empty; this can be ensured by pressing SHIFT-ERASE. Press SHIFT and 2-OFF and the display will show the word "tESt" when the loading of the DEMO program into active memory is complete. If the memory is not empty when attempting to load DEMO, the display will show the word "dAtA". If the program loaded successfully, the system is ready to execute by pressing RUN.

The following are the keystrokes that make up the DEMO program for the XR.

Move 1	LABEL 1		
	B 170	wrist rotate clockwise	
	C 320	wrist up	
	D 170	elbow up	
	E 160	shoulder down	
	F 270	waist counterclockwise	
	Aux Output	1-ON	
	Aux Output	2-OFF	ENTER
Move 2	C 140	wrist down	
	D -340	elbow down	
	E 440	shoulder down	
	F -320	waist clockwise	
	Aux Output	1-OFF	
	Aux Output	2-ON	ENTER
Move 3	D 360	elbow up	
	E -100	shoulder up	
	F 10	waist counterclockwise	
	Aux Output	1-ON	
	Aux Output	2-OFF	ENTER
Move 4	Go Soft Home		
	Aux Output 1-OFF		

	Aux Output 2-ON		ENTER
Move 5	C	620	wrist up
	D	250	elbow down
	E	170	shoulder down
	F	-310	waist clockwise
	Aux Output		1-On
	Aux Output		2-OFF ENTER
Move 6	Go Soft Home		
	Aux Output		1-OFF
	Aux Output		2-ON
	Jump To Label 1		ENTER

The following is the keystrokes for the SCARA version of the DEMO program.

Move 1	LABEL 1		
	B	170	wrist rotate clockwise
	D	-340	elbow up
	E	410	shoulder rotate
	Aux Output		1-ON
	Aux Output		2-OFF ENTER
Move 2	C	150	wrist down
	Aux Output		1-OFF
	Aux Output		2-ON ENTER
Move 3	B	0	
	C	0	
	D	0	
	E	0	
	Aux Output		1-ON
	Aux Output		2-OFF ENTER
Move 4	D	-434	
	E	-650	
	Aux Output 1-OFF		
	Aux Output 2-ON		ENTER

Move 5	C	-160	
	Aux Output	1-On	
	Aux Output	2-OFF	ENTER
Move 6	C	0	
	D	0	
	E	0	
	Aux Output	1-OFF	
	Aux Output	2-ON	
	Jump To Label 1		ENTER

APPENDIX IV

GENERAL OPERATING NOTES

When the controller is first turned ON (or is RESET), all the motors are moved two encoder steps in both directions to determine if the motors are connected to the system. Motors that do not respond to this move request are considered to have been disconnected. The keys associated with these motors are disabled by the software. A remote possibility exists that a stalled motor can be seen as a disconnected motor because it cannot move. If a motor is stalled, move it off the stall manually to allow the initialization to take place.

Some functions do not actually execute until the key is released, although the display may show what is going to happen.

The lefthand character of the display always shows the current mode the system is in. P for Play, L for Learn, E for edit, R for run, H for Robotalk or Rhino-VAL and HOST for host.

Invalid key presses will often display the message ERROR or the condition of the message. For example, pressing NEXT MOTOR while in LEARN mode will give a display: L EDIT, signifying the fact that the key is valid only in EDIT mode.

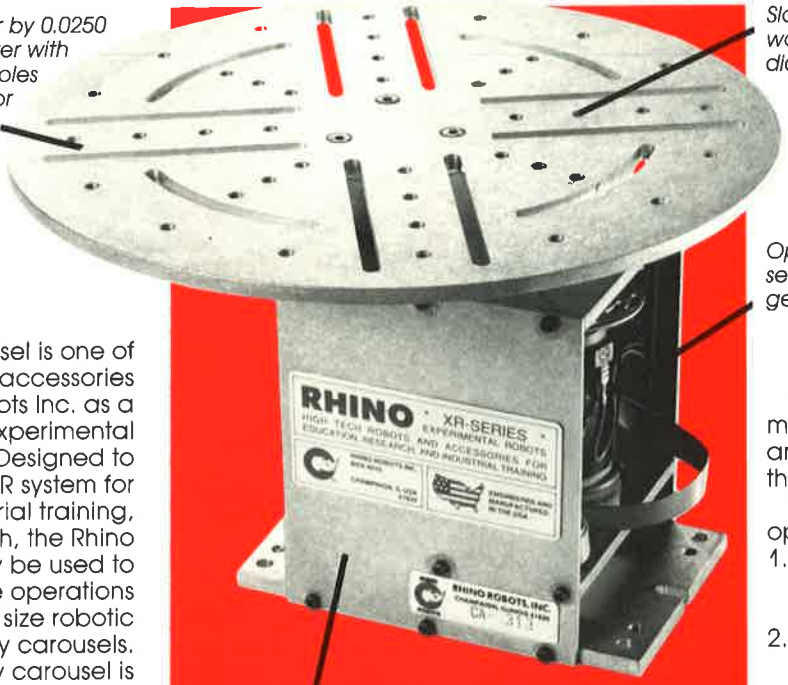
Most multiple key-press functions can be aborted at any time by pressing the same initial key again. For example, to clear an OUTPUT 2-ON command, the sequence would be: TURN OUTPUT, CLEAR and then 2-ON. If the sequence TURN OUTPUT, TURN OUTPUT or TURN OUTPUT, CLEAR, TURN OUTPUT is pressed, no changes will have occurred to the current record.

On reset or power-up, the system automatically checks whether or not the controller can accept I/O commands. If the controller can not accept I/O commands then the function to activate I/O will be disabled.

When editing or deleting, only the current record is affected. For example, if the arm is moved to a position, the hand is closed and the record stored, then the hand is opened and the record stored again and then the outputs are set and the record stored once more, there will be 3 records with the arm in essentially the same position. Going to the first of the 3 records and modifying the position will not affect the other two records. In some cases there may have been more than 3 records stored. It is always a good idea to step through the program several times during editing before actually running the program.

XR Series
Rotary Carousel

12 inch (304.8 mm) diameter by 0.0250 inch (6.4 mm) thick platter with mounting holes and slots suitable for varied applications



Slots and holes accommodate four work stations. Suitable for 0.250 inch diameter by 20 tpi hardware.

Needle bearing and ball bearings on main shaft for smooth operation

Optically encoded DC servomotor with 65.5:1 integral gearbox

Aircraft grade aluminum construction

Rhino's XR rotary carousel is one of the family of robotic accessories provided by Rhino Robots Inc. as a part of the popular XR experimental robotic system. Designed to ance the use of the XR system for education, industrial training, simulation and research, the Rhino XR rotary carousel may be used to emulate almost all the operations executed by the full size robotic rotary carousels.

The platform of the rotary carousel is supported by needle bearings at the top and ball bearings at the bottom which take all the thrust loading of the shaft. Mounting holes in the platform are in the form of a system of holes, straight slots and curved slots that can be used to mount a number of varied devices. The surface can accommodate four stations without difficulty. Like all other Rhino XR accessories, the rotary carousel is constructed of aircraft grade aluminum for ruggedness and durability without adding weight. The rotary carousel platform is belt driven and can travel in either direction in any combination of moves programmed by the user. A microswitch, which allows the system to be reset to a known position after each start up, can be read from the controller and from the host computer.

The servomotor is identical to the motors used to operate the robot and all the motorized accessories that form the XR system.

The rotary carousel may be operated in the following modes:

1. Directly from any one of the eight encoded motor ports on the Rhino 8 axis controller.
2. Directly from one of the auxiliary ports of the Mark III controller. In this mode the control is ON-OFF.

Each of the above modes of operation may be controlled in one of the following ways:

1. From the kernel commands. The slide base may be controlled from the kernels that form the language of the XR controllers.
2. From RoboTalk™. RoboTalk contains commands that may be used to control the rotary carousel either as a motor port accessory or as an auxiliary port accessory.
3. From Rhino-VAL. Rhino-VAL is an emulation of the Unimation robotic language VAL™, providing a subset of the language with 53 commands. The emulation contains the commands necessary to control the rotary carousel as an auxiliary accessory.
4. From the teach pendant system.

RHINO® ROBOTS, INC.

world leader in instructional robotics

Specifications

XR SERIES ROTARY CAROUSEL

APPLICATIONS

The rotary carousel provides the XR series robot with a rotary table with which it can interact intelligently. All applications requiring the emulation of a rotary table will benefit from the addition of a rotary carousel to the system.

The rotary carousel can be operated in conjunction with the Rhino XR robot to simulate innumerable work cells. The most readily apparent applications for the rotary carousel are:

1. Provide the robot with an easily controlled rotary table.
2. Use as a stand alone device to bring parts to the robot arm for manipulation.
3. Systems that exploit the I/O capabilities of the controller can be built around the rotary carousel with or without other Rhino XR accessories.

Example 1: The rotary carousel can be used with an XR robot arm to simulate the operation of most simple indexing table assembly operations. Four work stations are easily put on the carousel. Station one can be the loading and unloading station. The robot unloads the finished part from this station and then loads a new part in the same location. Stations two and three can be simple assembly stations. Station four can be an inspection station that determines if the task performed was properly executed. One or more robots or other devices can interact with the cell.

Example 2. The rotary carousel can be used in an elementary pick and place operation where the robot places a part on one position of the carousel, the carousel moves the part to another location and then the robot removes the part from the second location. A number of other operations can be placed within the basic framework of this simple cell. Almost all inspection tasks are variations of this task and consist of picking up, inspecting and placing. Vision and optical inspection systems can be built around the basic hardware.

WORK CELL DEVELOPMENT

In order to facilitate work cell development, Rhino offers a large number of coordinated accessories that can be controlled through the Rhino XR controller. These accessories include the rotary carousel, a slide base, a tilting rotating carousel, a belt conveyor, an XY table, a 32 key hand held teach pendant with display and a number of assorted gripper attachments.

All motor driven accessories have optically encoded motors with integral gearboxes that can, through the XR controller, be programmed with the teach pendant alone, with a host computer or with the teach pendant in conjunction with a host computer by using Rhino's higher level languages such as RoboTalk™ and Rhino-VAL, Rhino's emulation of Unimation VAL™.

Model	XR Series Rotary Carousel
Part Number	FG-0034
Applications	Education, Industrial Training, Research, Work Cell Modeling, Robotic Simulation
Dimensions	Diameter 12.0 inches (304.8 mm) Height 7.5 inches (190.5 mm)
Weight	6.5 lbs (3.00 kg)
Platform Dimensions	12.0 inches diameter (304.8 mm)
Platform Rotation	Infinite in each direction
Angular Positional Resolution	0.14 degrees per step (theoretical)
Nominal Speed	40.0 degrees per second
Operation Control	Operated through XR controller using any of the encoded ports "A" to "H" to auxiliary ports 1 or 2. Programming control from Rhino teach pendant or through host computer via RoboTalk or other emulated languages.
Warranty	1 year on mechanical parts 6 months on electrical parts
Trademarks	VAL is a trademark of the Unimation Corporation. RoboTalk is a trademark of Rhino Robots Inc.

Rhino Robots Inc. reserves the right to change any and all specifications and prices without prior notice.

Rhino Robots Inc.
308 South State Street
P.O. Box 4010
Champaign, Illinois USA 61820
Telephone: 217-352-8485
TELEX: 3734731 RHINO ROBOTS C

Rhino has representatives throughout the USA, Canada and a large number of other countries. Please call or write for the name of the representative in your area.

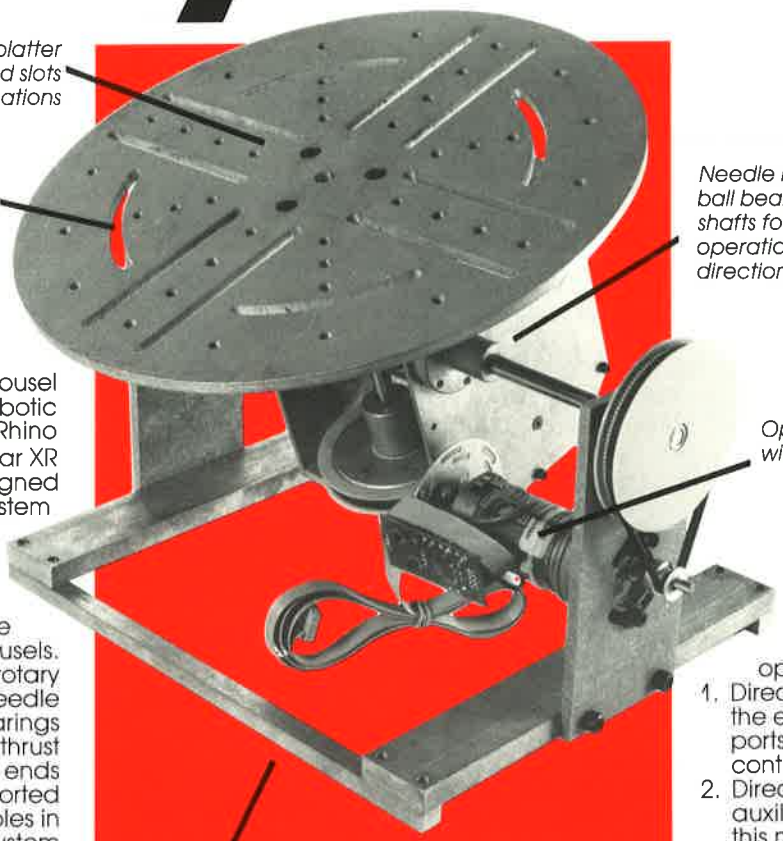
ENGINEERED AND MANUFACTURED IN THE USA

RHINO® ROBOTS, INC.

world leader in instructional robotics

XR Series

Tilting Rotary Carousel



Rhino's XR tilting rotary carousel is one of the family of robotic accessories provided by Rhino Robots Inc. as a part of the popular XR experimental robotic system. Designed to enhance the use of the XR system for education, industrial training, sir tion and research, the Rhino . ilting rotary carousel may be used to emulate almost all of the operations executed by the full size robotic tilting rotary carousels. The platform of the tilting rotary carousel is supported by needle bearings at the top and ball bearings at the bottom which take all the thrust loading of the vertical shaft. Both ends of the horizontal support are supported by needle bearings. Mounting holes in the platform are in the form of a system of holes, straight slots and curved slots that can be used to mount a number of varied devices on the carousel. The surface can accommodate four stations without difficulty. Like all other Rhino XR accessories, the tilting rotary carousel is constructed of aircraft grade aluminum for ruggedness and durability without adding weight. The tilting rotary carousel platform is belt driven and can travel in either direction on the rotational axis and can be tilted 90 degrees in either direction on the tilt axis. Microswitches, which allow the system to be reset to a known position after each start up, can be read from the controller and from the host computer. The servomotors are identical to the motors used to operate the robot and all the motorized ccessories that form the XR system.

The tilting rotary carousel may be operated in the following modes:

1. Directly from any two of the eight encoded motor ports on the Rhino 8 axis controller.
2. Directly from one or both of the auxiliary ports of the controller. In this mode the control is ON-OFF.

Each of the above modes of operation may be controlled in one of the following ways:

1. From the kernel commands. The tilting rotary carousel may be controlled from the kernels that form the language of the XR controllers.
2. From RoboTalk™. RoboTalk contains commands that may be used to control the tilting rotary carousel either as a motor port accessory or as an auxiliary port accessory.
3. From Rhino-VAL. Rhino-VAL is an emulation of the Unimation robotic language VAL™, providing a subset of the language with 53 commands. The emulation contains the commands necessary to control the tilting rotary carousel as an auxiliary accessory.
4. From the teach pendant system.

RHINO® ROBOTS, INC.

world leader in instructional robotics

Specifications

XR SERIES TILTING ROTARY CAROUSEL

APPLICATIONS

The tilting rotary carousel provides the XR series robot with a tilting rotary table with which it can interact intelligently. All applications requiring the emulation of a tilting and rotating table will benefit from the addition of a tilting rotary carousel to the system.

The tilting rotary carousel can be operated in conjunction with the Rhino XR robot to simulate work cells. The most readily apparent applications for the tilting rotary carousel are:

1. Provide the robot with an easily controlled tilting rotary table.
2. Used as an stand alone device to bring parts to the robot arm for manipulation.
3. Systems that exploit the I/O capabilities of the controller can be built around the tilting rotary carousel with or without using other Rhino XR accessories.

Example 1: The rotary carousel can be used with an XR robot arm to simulate the operation of most simple indexing table assembly operations. Four work stations are easily put on the tilting carousel. Station one can be the loading and unloading station. The robot unloads the finished part from this station and then loads a new part in the same location. Stations two and three can be simple assembly stations. Station four can be an inspection station that determines if the task performed was properly executed. One or more robots or other devices can interact with the cell.

Example 2: The rotary carousel can be used to demonstrate how placing a workpiece on a tilting rotary table can substantially increase the ability of the robot to work on the part. The tilting carousel therefore forms the basic table on which all robot manipulations should be planned. The most difficult manipulation is the welding operation because the weld puddle is affected by gravity and welding has to take this into account. As such, the tilting feature is designed specifically for the simulation of welding operations.

WORK CELL DEVELOPMENT

In order to facilitate work cell development, Rhino offers a large number of coordinated accessories that can be controlled through the Rhino XR controller. These accessories include the tilting rotary carousel, a slide base, a belt conveyor, an XY table, a 32 key hand held teach pendant with display and a number of assorted gripper attachments.

All motor driven accessories have optically encoded motors with integral gearboxes that can, through the XR controller, be programmed with the teach pendant alone, with a host computer alone or with the teach pendant in conjunction with a host computer by using Rhino's higher level languages such as RoboTalk and Rhino-VAL, Rhino's emulation of Unimation VAL.

Model	XR Series Tilting Rotary Carousel
Part Number	FG-1004
Applications	Education, Industrial Training, Research, Work Cell Modeling, Robotic Simulation
Dimensions	Diameter 12.0 inches (304.8 mm) Height 9.125 inches (231.8 mm) Width 14.5 inches (368.3 mm)
Weight	11.5 lbs (5.23 kg)
Platform Dimensions	12.0 inches diameter (304.8 mm)
Platform Rotation	Infinite in each direction
Platform Tilt	90 degrees in either direction
Angular Positional Resolution	0.14 degrees per step (theoretical)
Tilting Positional Resolution	0.14 degrees per step (theoretical)
Nominal Rotational Speed	40.0 degrees per second
Nominal Tilting Speed	40.0 degrees per second
Operation Control	Operated through XR controller using any of the encoded ports "A" to "H" or auxiliary ports 1 or 2. Programming control from Rhino teach pendant or through host computer via RoboTalk™ or other emulated languages.
Warranty	1 year for mechanical parts 6 months for electrical parts
Trademarks	VAL is a trademark of the Unimation Corporation. RoboTalk is a trademark of Rhino Robots Inc.

Rhino Robots Inc. reserves the right to change any and all specifications and prices without prior notice.

Rhino Robots Inc.
308 South State Street
P.O. Box 4010
Champaign, Illinois USA 61820
Telephone: 217-352-8485
TELEX: 3734731 RHINO ROBOTS C

Rhino has representatives throughout the USA, Canada and a large number of other countries. Please call or write for the name of the representative in your area.

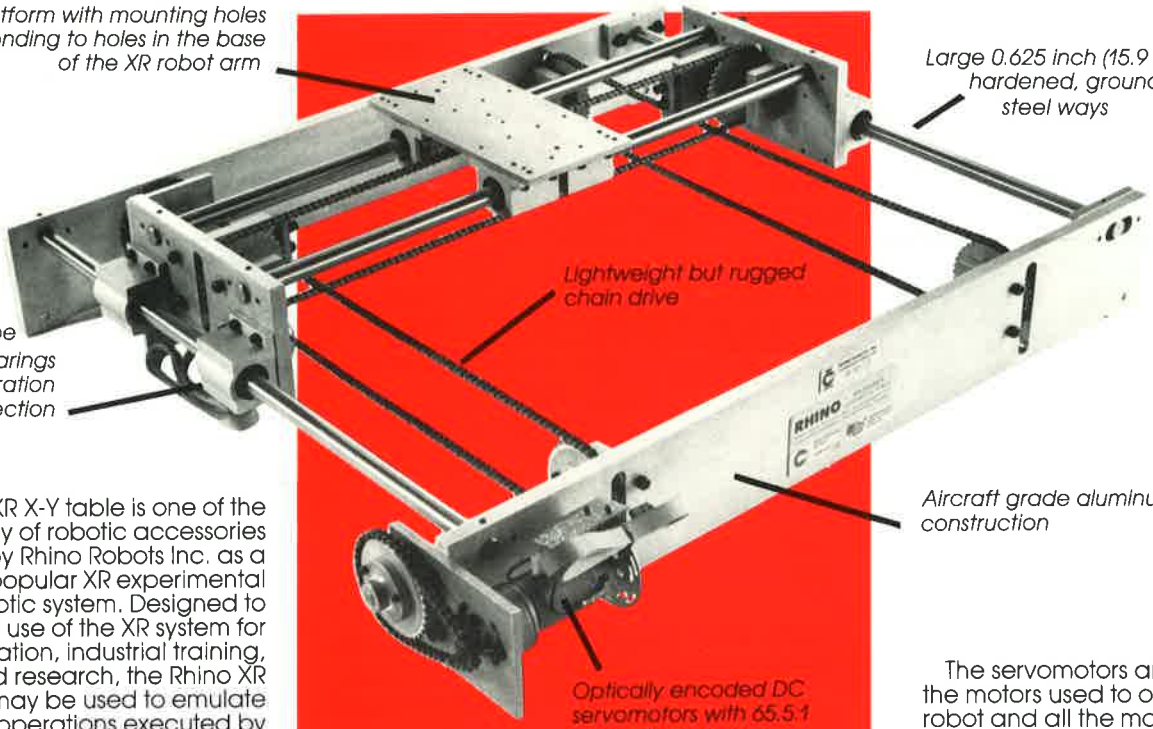
ENGINEERED AND MANUFACTURED IN THE USA

RHINO® ROBOTS, INC.

world leader in instructional robotics

XR Series

X-Y Table



Rhino's XR X-Y table is one of the family of robotic accessories provided by Rhino Robots Inc. as a part of the popular XR experimental robotic system. Designed to enhance the use of the XR system for education, industrial training, simulation and research, the Rhino XR X-Y table may be used to emulate almost all the operations executed by the full size industrial robotic X-Y tables.

The platform of the X-Y table is supported by linear ball bearings that ride on ground and polished steel shafts. The X-Y table is the equivalent of a milling machine table or other machine tool table that the robot would feed and interact with. The X-Y table can be used to carry the robot itself or to mount some other device that forms a part of the robot's work cell. The X-Y table thus forms the basis for providing the XR robot arm with a motorized work platform that can be controlled from the robot controller.

Like all other Rhino XR accessories, the X-Y table is constructed of aircraft grade aluminum to provide ruggedness and durability without adding weight. The X-Y table platform is chain driven and can travel a maximum of 18 inches (457.2 mm) in one direction and 12 inches (304.8 mm) in the other. Microswitches, which allow the system to be reset to a known position after each start up, can be read from the controller and from the host computer.

The servomotors are identical to the motors used to operate the robot and all the motorized accessories that form the XR system.

The X-Y table may be operated in the following modes:

1. Directly from any two of the eight encoded motor ports on the Rhino 8 axis controller.
2. Directly from the two auxiliary ports of the Mark III controller. In this mode the control is ON-OFF.

Each of the above modes of operation may be controlled in one of the following ways:

1. From the kernel commands. The X-Y table may be controlled from the kernels that form the language of the XR controllers.
2. From RoboTalk™. RoboTalk contains commands that may be used to control the X-Y table either as a motor port accessory or as an auxiliary port accessory.
3. From Rhino-VAL. Rhino-VAL is an emulation of the Unimation robotic language VAL™ providing a subset of the language with 53 commands. The emulation contains the commands necessary to control the X-Y table as an auxiliary accessory.
4. From the teach pendant system.

RHINO®

ROBOTS, INC.

world leader in instructional robotics

Specifications

XR SERIES X-Y TABLE

APPLICATIONS

The X-Y table provides the XR series robot with the ability to interact with a motorized table. Strategies for working with machine tool tables can thus be investigated with the X-Y table. All applications requiring the use of a motorized table will benefit from the addition of a X-Y table to the system.

The X-Y table can be operated in conjunction with the Rhino XR robot to simulate innumerable work cell configurations found in the industrial environment. The most readily apparent applications for the X-Y table are:

1. Simulation of palletizing schemes and the development of relevant software.
2. Simulation of a cartesian robotic system.
3. As a stand alone device to bring parts to the robot arm for manipulation.

Example 1: The X-Y table can be used with an XR robot arm to simulate the operation of an automatic palletizing operation in which the robot performs the basic pick and place operations and the X-Y table provides the incremental palletizing movements. Appurtenances that allow the loading and unloading of the pallets can be added to the system.

Example 2: The X-Y table can be used as a machine tool table and the robot can be used to place and remove parts to and from the table as a part of a larger work cell. For advanced experimentation I/O can be added to the system to provide positional feedback and thus add intelligence to the system.

WORK CELL DEVELOPMENT

In order to facilitate work cell development, Rhino offers a large number of coordinated accessories that can be controlled through the Rhino XR controller. These accessories include the X-Y table, a rotary carousel, a tilting rotating carousel, a belt conveyor, a 32 key hand held teach pendant with display and a number of assorted gripper attachments.

All motor driven accessories have optically encoded motors with integral gearboxes that can, through the XR controller, be programmed with the teach pendant alone, with a host computer alone, or with the teach pendant in conjunction with a host computer by using Rhino's higher level languages such as RoboTalk and Rhino-VAL, Rhino's emulation of Unimation VAL.

ENGINEERED AND MANUFACTURED IN THE USA

Model	XR Series X-Y table
Part Number	FG-0037
Applications	Education, Industrial Training, Research, Work Cell Modeling, Robotic Simulation
Dimensions	Length 29.75 inches (755.7 mm) Width 27.00 inches (685.8 mm) Height 6.375 inches (162.0 mm)
Weight	35.0 lbs (15.9 kg)
Platform Dimensions	4.25 x 8.5 inches (108.0 x 215.9 mm)
Platform Travel Area	20 x 22 inches (508 x 558 mm)
Positional Resolution	0.006 inches (0.15 mm) theoretical
Nominal Speed	1.5 inches/sec. (38.1 mm/sec.) in each direction
Operation Control	Operated through XR controller using any of the encoded ports "A" to "H" or auxiliary ports 1 or 2. Programming control from Rhino teach pendant or through host computer via RoboTalk or other emulated languages.
Warranty	1 year on mechanical parts 6 months on electrical parts
Trademarks	VAL is a trademark of the Unimation Corporation. RoboTalk is a trademark of Rhino Robots Inc.

Rhino Robots Inc. reserves the right to change any and all specifications and prices without prior notice.

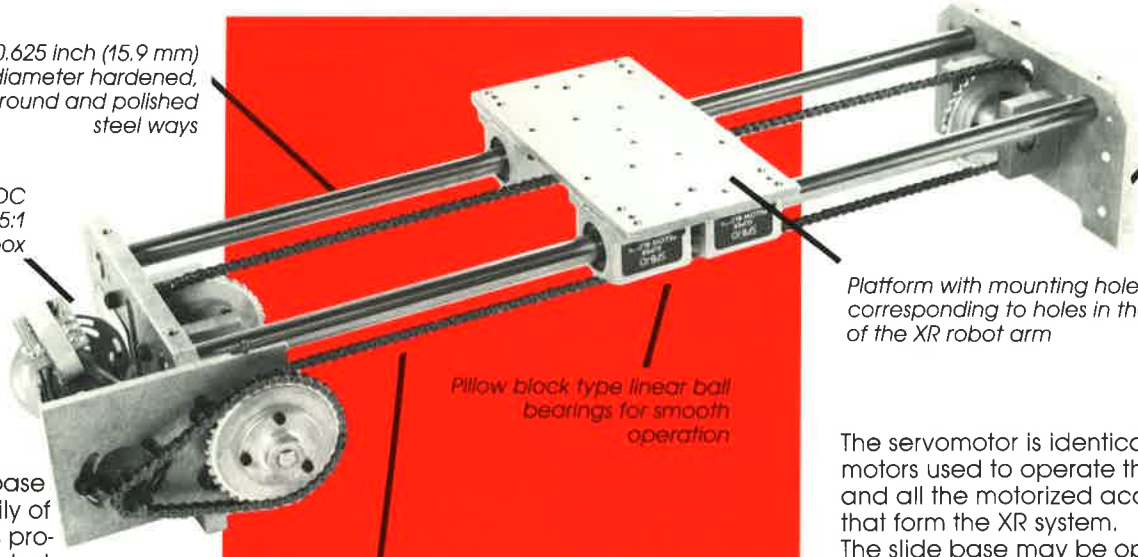
Rhino Robots Inc.
308 South State Street
P.O. Box 4010
Champaign, Illinois USA 61820
Telephone: 217-352-8485
TELEX: 3734731 RHINO ROBOTS C

Rhino has representatives throughout the USA, Canada and a large number of other countries. Please call or write for the name of the representative in your area.

RHINO®
ROBOTS, INC.
world leader in instructional robotics

XR Series

Linear Slide Base



Rhino's XR slide base is one of the family of robotic accessories provided by Rhino Robots Inc. as a part of the popular XR experimental robotic system. Designed to enhance the use of the XR system for education, industrial training, simulation and research, the Rhino XR slide base may be used to emulate almost all the operations executed by the full size industrial robotic slide bases.

The platform of the slide base is supported by linear ball bearings that ride on ground and polished steel shafts. The slide base can be used to carry the robot itself, providing the 5 axis XR robot arm with a 6th axis, or to mount some other device that forms a part of the robot's environment.

Like all other Rhino XR accessories, the slide base is constructed of aircraft grade aluminum for ruggedness and durability without adding weight. The slide base platform is chain driven and can travel a maximum of 22 inches (558.8 mm). A microswitch, which allows the system to be reset to a known position after each start up, can also be read from the controller and from the host computer.

The servomotor is identical to the motors used to operate the robot and all the motorized accessories that form the XR system.

The slide base may be operated in two modes:

1. Directly from any one of the eight encoded motor parts on the Rhino 8 axis controller.
2. Directly from one of the auxiliary ports of the Mark III controller. In this mode the control is ON-OFF.

Each of the above modes of operation may be controlled in one of the following ways:

1. From the kernel commands. The slide base may be controlled from the kernels that form the language of the XR controllers.
2. From RoboTalk™. RoboTalk contains commands that may be used to control the slide base either as a motor port accessory or as an auxiliary port accessory.
3. From Rhino-VAL. Rhino-VAL is an emulation of the Unimation robotic language VAL™, providing a subset of the language with 53 commands. The emulation contains the commands necessary to control the slide base as an auxiliary accessory.
4. From the teach pendant system.

RHINO®

ROBOTS, INC.

world leader in instructional robotics

Specifications

XR SERIES BELT CONVEYOR

APPLICATIONS

The belt conveyor provides the XR series robots with the ability to interact with a belt conveyor to implement material handling strategies.

The belt conveyor can be operated in conjunction with the Rhino XR robot to simulate innumerable conveyor configurations found in the industrial environment. The most readily apparent applications for the linear belt conveyor are:

1. Extend the reach of the robot beyond its normal work envelope.
2. Use as a stand alone device that can bring parts to the robot arm for manipulation.
3. Systems that exploit the I/O capabilities of the controller can be built around the belt conveyor with or without using other Rhino XR accessories.

Example 1: A sophisticated parts sorting system can be created with the conveyor. Sensors and simple passive devices are easily attached to the conveyor to allow a parts sorting cell to be created.

Example 2: The belt conveyor can be used to emulate various full sized operations that use conveyors and robots interactively. These include everything from simple material handling and placing systems to fairly sophisticated, intelligent robot dominated systems that serve automated packaging, palletizing and storage systems.

WORK CELL DEVELOPMENT

In order to facilitate work cell development, Rhino offers a large number of coordinated accessories that can be controlled through the Rhino XR controller. These accessories include the belt conveyor, a rotary carousel, a tilting rotating carousel, a vertical conveyor, an XY table, a 32 key hand held teach pendant with display and a number of assorted gripper attachments.

All motor driven accessories have optically encoded motors with integral gearboxes that can, through the XR controller, be programmed with the teach pendant alone, with a host computer alone or with the teach pendant in conjunction with a host computer by using Rhino's higher level languages such as RoboTalk™ and Rhino-VAL, Rhino's emulation of Unimation VAL™.

ENGINEERED AND MANUFACTURED IN THE USA

Model	XR Series Belt Conveyor
Part Number	FG-864
Applications	Education, Industrial Training, Research, Work Cell Modeling, Robotic Simulation
Dimensions	Length 31.5 inches (800.1 mm) Width 7.25 inches (184.2 mm) Height 4.125 inches (104.8 mm)
Weight	12.0 lbs (5.45 kg)
Max usable conveyor length	25.5 inches (647.7 mm)
Positional Resolution	0.006 inches (0.15 mm) theoretical
Nominal Speed	1.5 inches/sec. (38.1 mm/sec.)
Operation Control	Operated through XR controller using any of the encoded ports "A" to "H" or auxiliary ports 1 or 2. Programming control from Rhino teach pendant or through host computer via RoboTalk or other emulated languages.
Warranty	1 year on mechanical parts 6 months on electrical parts
Trademarks	VAL is a trademark of the Unimation Corporation. RoboTalk is a trademark of Rhino Robots Inc.

Rhino Robots Inc. reserves the right to change any and all specifications and prices without prior notice.

Rhino Robots Inc.
308 South State Street
P.O. Box 4010
Champaign, Illinois USA 61820
Telephone: 217-352-8485
TELEX: 3734731 RHINO ROBOTS C

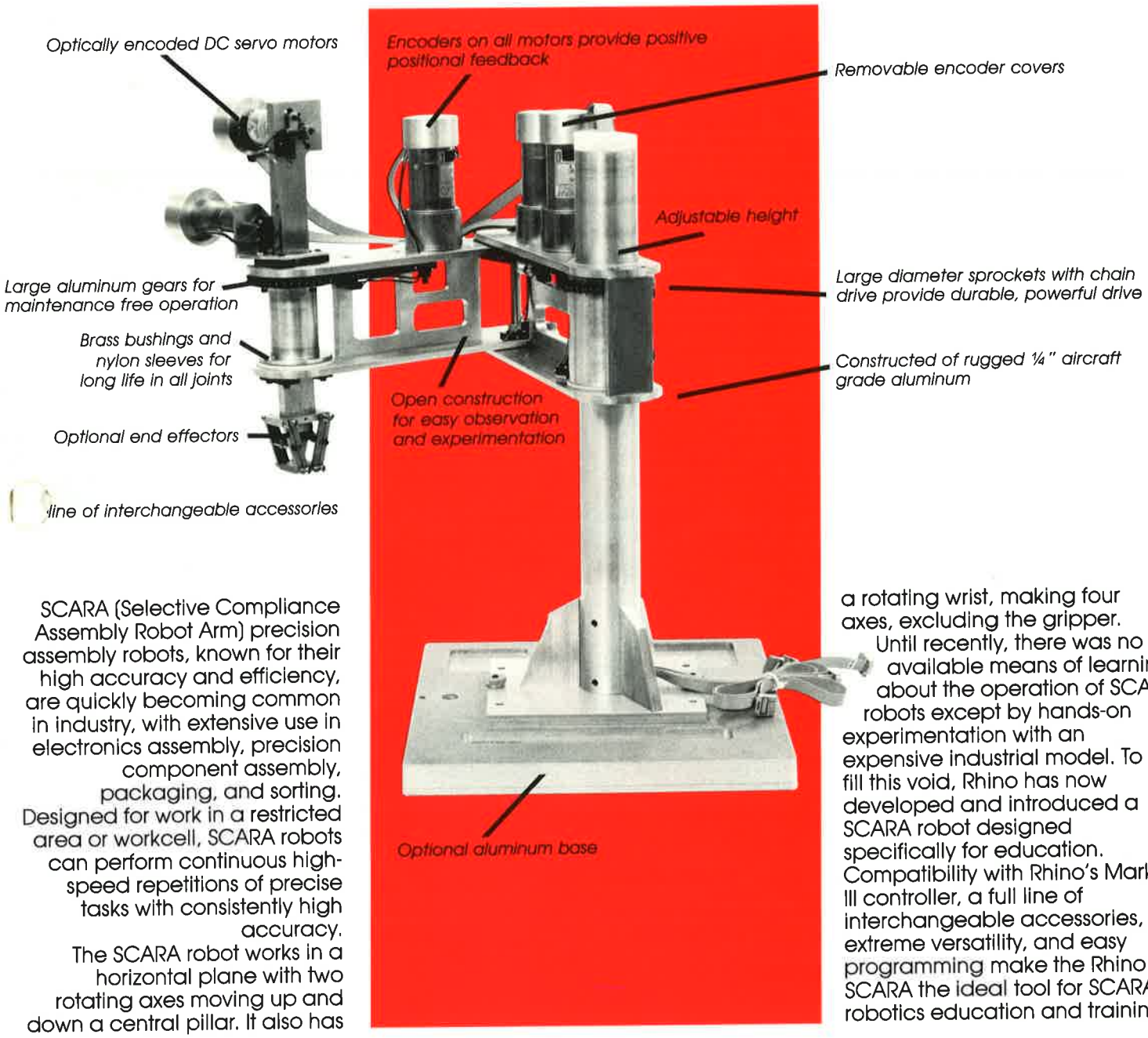
Rhino has representatives throughout the USA, Canada and a large number of other countries. Please call or write for the name of the representative in your area.

RHINO®
ROBOTS, INC.
world leader in instructional robotics

NEW

XR Series

SCARA Robot



SCARA (Selective Compliance Assembly Robot Arm) precision assembly robots, known for their high accuracy and efficiency, are quickly becoming common in industry, with extensive use in electronics assembly, precision component assembly, packaging, and sorting. Designed for work in a restricted area or workcell, SCARA robots can perform continuous high-speed repetitions of precise tasks with consistently high accuracy.

The SCARA robot works in a horizontal plane with two rotating axes moving up and down a central pillar. It also has a rotating wrist, making four axes, excluding the gripper.

Until recently, there was no available means of learning about the operation of SCARA robots except by hands-on experimentation with an expensive industrial model. To fill this void, Rhino has now developed and introduced a SCARA robot designed specifically for education. Compatibility with Rhino's Mark III controller, a full line of interchangeable accessories, extreme versatility, and easy programming make the Rhino SCARA the ideal tool for SCARA robotics education and training.

RHINO®

ROBOTS, INC.

world leader in instructional robotics

Specifications

XR SERIES END EFFECTOR PACKAGE

FEATURES

VERSATILE — The Rhino end effector package allows experimentation with a wide variety of applications.

COST EFFECTIVE — There is no more cost effective means of experimenting with end effector design and applications. Furthermore, significant cost savings are realized by purchasing the end effectors as a complete package, **Model No. FG1125.**

EASILY INSTALLED — Any of the Rhino optional grippers may be installed in minutes for convenient use and experimentation.

DURABLE — Like all other Rhino XR Series accessories, the Rhi end effectors are constructed of aircraft grade aluminum for maximum durability without adding weight.

INTERCHANGEABLE — All of the Rhino optional end effectors are interchangeable with each other and with the standard gripper that comes with the XR Series robotic arm.

End Effector Package Components

Model	Part Number	Specifications	Applications	Items Included
XR Series 2.0 Finger Attachment	FG0707	Opening width—1.5 inches (38.1mm) Depth of width—2.5 inches (63.5mm) Length of gripper—3.7 inches (93.98mm)	Gripping taller and wider objects.	2.0 Finger Attachment
XR Series Triple Finger Attachment	FG0701	Opening diameter—1.5 inches (38.1mm) Depth of width—.5 inches (12.7mm) Placement of 3 fingers—120 degree intervals	Gripping round objects, centering objects for consistent placement, study of multi-fingered grippers.	Triple Finger Attachment
XR Series Narrow Finger Attachment	FG0822	Opening width—1.25 inches (31.75 mm) Depth of reach—.75 inches (19.05mm)	Accessing small spaces that are normally inaccessible, grasping the bridge or handle of an object.	Narrow Finger Attachment
Vacuum Finger Attachment	FG0700	Voltage—12 volt pneumatic solenoid Maximum vacuum—40 pounds (18.18kg) Length—2 inches (50.8mm)	Experimentation with vacuum handling, transporting freshly machined or painted surfaces, gripping samples of paper and other material, palletizing.	Vacuum attachment, circuit board with solenoid vacuum valve, air tubing, instructions for installation (vacuum source not included)
XR Series Magnetic Finger Attachment	FG0710	Voltage—12 volt pneumatic solenoid Length—2 inches (50.8mm)	Experimentation with magnetic handling, lifting small, ferrous objects, lifting objects with smooth surfaces which should not be marred in transport, activating magnetic switches, palletizing.	Magnetic solenoid, circuit board, cable, mounting instructions, and screws
XR Series Moto Dremel Hand	FG0871	Tool model—380 Motor—permanent magnetic field, 115 v. 60 HZ, AC, 0.75 AMPS at low speed, 0.9 AMPS at high speed Speed—variable 5,000 to 28,000 R.P.M. Housing—sturdy, shatterproof nylon cord—grounded, 3 wire type Bearings—ball bearings Weight—12 oz. (.34 kg) Size—7.25 inches (184.15mm) long x 1.875 inches (47.625mm)	Experimentation with cutting, sharpening, cleaning, carving, sanding, polishing, shaping, and grinding.	Moto dremel, hand, cable stringing instructions, mayline cable, documentation (drill bits not included)
XR Series Shovel Finger Attachment	FG0711	Capacity—7.5 cubic inches (190.5 cubic mm)	Scooping up a pile of material or objects such as sand or nuts and bolts, experimenting with wrist movements.	Shovel Finger Attachment
Clam Shell Finger Attachment	FG0713	Opening width—3.0 inches (76.2mm)	Gripping small objects like nuts and bolts.	Clam Shell Finger Attachment
XR Series Rhino Writer Hand	FG0703	Length—8.5 inches (215.9mm) Inside diameter—.6 inches (15.24mm)	Writing letters and numbers, simulating arc welding, continuing path studies.	Writer hand, software disk for Apple IIe, felt tipped pen, mayline cable, cable stringing instructions, manual

Rhino Robots, Inc. reserves the right to change any and all specifications and prices without written notice.

RHINO ROBOTS

Rhino Robots Inc.
308 South State Street
P.O. Box 4010
Champaign, Illinois USA 61820
Telephone: 217-352-8485
TELEX: 3734731 RHINO ROBOTS C

Rhino has representatives throughout the USA, Canada and a large number of other countries. Please call or write for the name of the representative in your area.

Represented locally by: