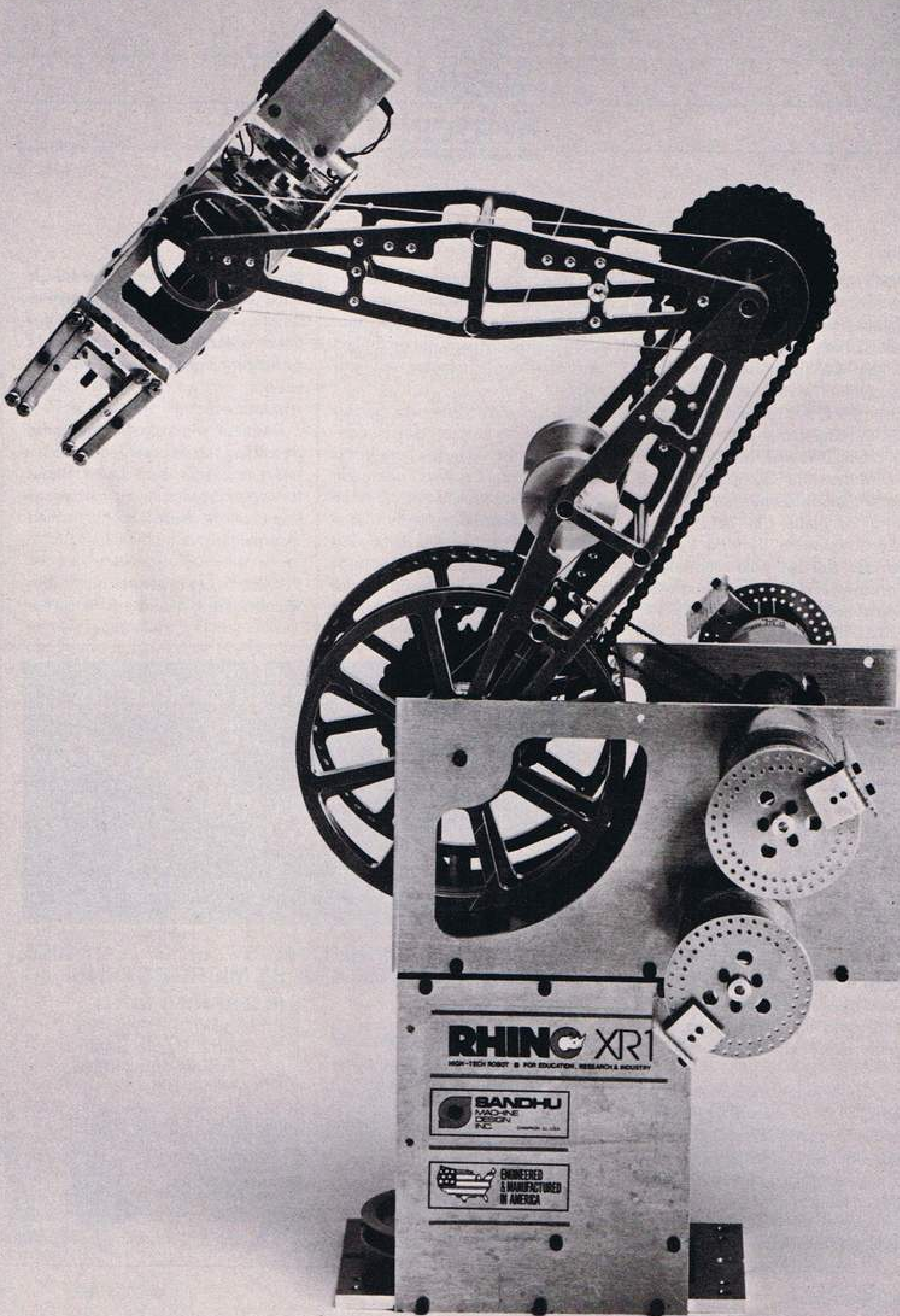

RHINO XR1
HIGH-TECH ROBOT FOR EDUCATION, RESEARCH & INDUSTRY
SANDHU
MACHINE
DESIGN
INC.
ENGINEERED
& MANUFACTURED
IN AMERICA

THE RHINO XR-1
A Hands-On Introduction to Robotics

Harprit Sandhu
Sandhu Machine Design, Inc.
308 S. State
Champaign, Illinois 61820

Opportunities for investigations into the field of robotics have until now been limited either to large research facilities and industrial and educational laboratories, or the small hobby shops in the basements of avid tinkerers. With the introduction of the Rhino XR-1, anyone interested in robotics can have access to a five-axis manipulator that can be completely controlled from any small computer. Hardware, firmware and software are modifiable.

The Rhino XR-1 is specifically designed as a research and education tool for the industrial environment. Modeled after the large industrial robot arms, it is completely digital in design and features information transfers from the computer to the Rhino XR-1 and from the Rhino XR-1 back to the computer. A controller card with an on-board microprocessor handles all the overhead associated with the control of the motors; the control, drive, and logic circuitry needed to monitor the entire robot are all on this one card. The card can also monitor six separate interrupts that can be configured as desired.

Origin

The Rhino XR-1 is a product of Sandhu Machine Design Inc., a firm in Champaign, Illinois, that specializes in the design and fabrication of special machinery. As one might expect, engineers everywhere who are associated with design machinery talk about the "robot revolution," but at our shop, the discussions led to the decision that we would design and market a small, sophisticated robot arm.

The first concrete opportunity to put our plan into effect arose early in 1981. The resources were committed and the result was the Experimental Robot-One, known as the XR-1. The Rhino concept comes from its strength and ruggedness.

Photo 1: The Rhino XR-1 stands 32 inches high and contains six motors, one for each of the five axes of the arm-like mechanism and a sixth to control the fingers. Using four basic commands, the Rhino can be programmed to perform a variety of sophisticated tasks.

Early Specification Ideas

Before a machine can be designed, its personality has to be defined. The following describes the initial personality of the Rhino XR-1. It is not necessarily what we ended up with, it is simply how we began.

It was decided that the Rhino XR-1 should be controlled by any computer with an RS-232C interface, the most common computer interface at this time. This means that, to the computer, the robot is a serial ASCII device no different from a terminal, modem, or printer.

It was further decided for reasons of economy that the full RS-232C interface should not be required. The robot would be designed so that it could be controlled by three wires. The three signals used are:

- Transmitted data, Line 2 (transmitted by the XR-1).
- Received data, Line 3 (received by the XR-1).
- Ground, Line 7. Common data ground.

It should also be possible, we decided, to run Rhino XR-1 from a modem in case someone needed remote capability.

We also wanted a data transmission rate that was adjustable from 300 to 9600 bps. Of these:

- 300 bps gives adequate operation. This rate is used by slow modems.
- 1200 bps is much better. This rate is used by the more expensive modems.
- 9600 bps gives crisp operation. Used for direct connection to a computer on site. This rate is available on almost all computers and is thus more flexible than 19200 bps.

We thought it advisable for the robot to have its own controller card. Since we'd decided on the RS-232C interface, this could make it possible to decrease the overhead on the host computer. Computers, being hundreds of times faster than the robots they control, could then be used for making further calculations and decisions while the robot's movements were being executed.

With this in mind, the Intel 8748 processor was chosen to control the interface, and design considerations were initiated.

We decided to avoid the use of stepper motors because of the following problems:

- Slippage and thus loss of positional information.
- Easy overloading of motors.
- Need for timed control at all times.
- Need for constant power for running and holding the motors.
- Ramping considerations are a must so that the inertial properties of the motors will not be overlooked.
- No longer digital if slippage occurs.

It was further decided that a good small robot should work just like the big industry robots. This would require that optical choppers be placed on all axes of the robot so that exact axes positioning could be repeated indefinitely. Eliminating loss of position information over a long period of time is very important for sophisticated operation and control.

As a rule of thumb, the resolution of the Rhino XR-1 would be about $\frac{1}{32}$ inch and the speed about 20 feet per minute or three seconds for every foot of movement (of the major axes under consideration). (Incidentally, the big computerized numerical control machines run at about 200 to 400 inches a minute when traversing a table — not unlike Rhino XR-1.)

Since the RS-232C interface can use 12V, it was decided that the motors should run at 12V. For the small amount of 5V power needed for the logic components, we used an on-board voltage regulator and stole the 5V from the + 12V side of the power supply.

A bi-polar power supply was chosen because it eases the problems associated with reversing DC motors. The system is implemented on the controller in the following manner. One side of the motor is always grounded, the other side is then connected to either + 12V or − 12V as needed to run the motor forward or backward. The microprocessor makes sure that both cannot be on at the same time. Needless to say, there are other schemes for achieving this result, but this was the most cost effective.

We also decided, for the sake of simplicity, to attempt to construct the robot almost entirely out of $\frac{3}{8}$ inch, $\frac{1}{4}$ inch, $\frac{1}{8}$ inch, and $\frac{1}{16}$ inch aircraft-grade aluminum. We would keep the rod sizes to a few standard shafts around $\frac{1}{4}$ inch, $\frac{5}{16}$ inch, and $\frac{3}{8}$ inch, and keep fastener sizes to a minimum. No rivets would be used and everything would be designed for easy disassembly.

All of this amounted to a big complicated order. However, we decided to build an initial run of 100 robots. If the acceptance in the marketplace was good and there was a need for such a robot, we could easily produce 100 robots a month.

We also determined that the educational and experimental nature of the robot would necessitate that it be designed so that it could be completely disassembled by the purchaser and put back together again. A tool kit would have to be provided to facilitate this process.

There could be no easily breakable parts — they would have to be substantial and solid and designed for a long, useful life. The experimenter would be able to see everything: nothing would be hidden unnecessarily.

In order to keep the cost of the robot down, the use of elaborate bearings would be minimized: not that bearings themselves are expensive, but they are time consuming to fit and retain accurately and, thus, expensive. Our life tests on the robots indicated that if a drop or two of 3-in-1 oil is applied to lubricate each lube point occasionally, the machine should last indefinitely.

It would be unrealistic to expect the base of the robot to be sufficiently heavy to allow the robot to lift heavy objects with its arms fully extended, so it was decided that the base would be designed to be easily attachable to a work surface. (The robot should not be used unless the base is attached to a solid surface.) A hole, a slot, and a threaded hole would be provided for mounting. The robot arm would be built as a space frame which would be extremely light and extremely rigid.

We determined that the body and the base should be fairly simple structures and allowances made so that the belts that drive the various axes could be easily adjustable.

In addition, the unit would have good firmware transparency, so that the person programming the arm would not have to worry about the machine firmware when developing his or her own software.

Finally, the machine would have to be easy to use, easy to program, flexible and versatile, and above all, fun and easy to learn from.

Actual Design Parameters

The Rhino XR-1 robot arm consists of three main components and their related devices:

- the mechanical arm,
- the controller card,
- the power supply (including all related cables and devices).

The Mechanical Arm. The mechanical arm is 32 inches high, and has five degrees of freedom plus controllable fingers. The unit is configured as follows:

- A base upon which the main body of the robot arm rotates. The base contains the motor that rotates the robot arm about the waist and forms a platform for the body to be mounted upon. It may be mounted in any position, and should be bolted down.
- A body element that forms the main structure for the drive mechanisms and mounts the three motors that control the movement of the bicep, the forearm and the wrist azimuthal movement. This section of the device contains the drive mechanisms for the controls, gears, and belts needed to effect the motions of the arm assembly.
- The bicep and the forearm consist of space frames made of $\frac{1}{4}$ inch square aluminum carved out of $\frac{1}{4}$ inch plate and assembled with $\frac{1}{4}$ inch diameter spacers. These form

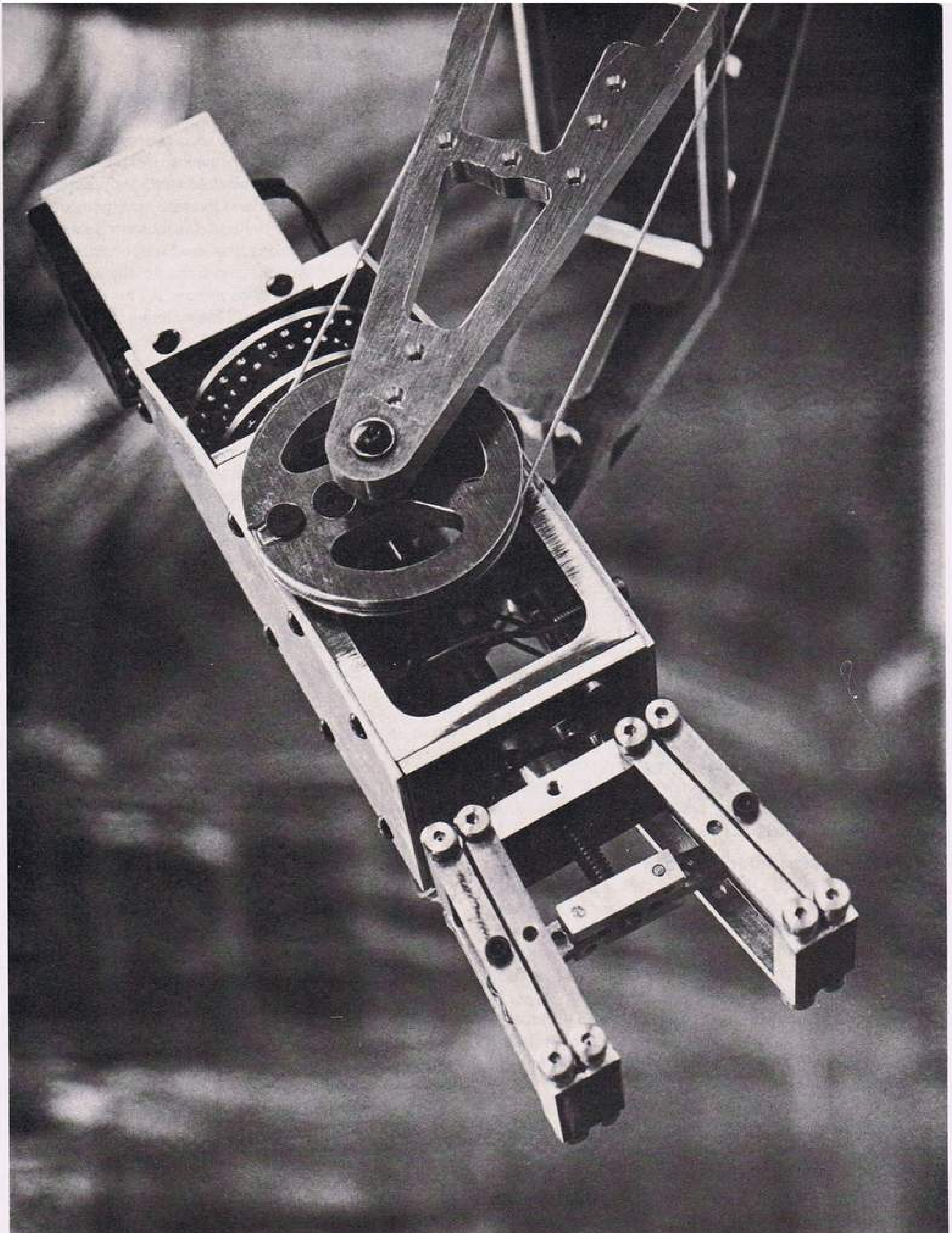


Photo 2: The Rhino's hand is mounted at the end of the forearm. The rotational movement of the hand and the closing of the fingers is controlled by two hand-mounted servomotors. The Rhino can reach up to 32 inches from base to top of fingertips and has a radial reach of 22.25 inches from center of rotation to tip of fingers. The arm's lifting ability and gripping force are approximately 16 ounces.

very rigid space frames and related hardware is mounted on them.

- The hand is mounted at the end of the forearm. The azimuthal movement of the hand is controlled from the body of the Rhino XR-1. The rotational movement of the hand and the closing of the fingers is controlled by two hand-mounted servomotors. One motor controls the rotation of the wrist and the other controls the closing of the fingers.

All six motors on the Rhino XR-1 have optical incremental encoders attached to them. This system of optical choppers provides a very effective form of digital positioning for each of the axes. The choppers for the waist, the shoulder joint, elbow joint, and wrist azimuthal joint run at the full speed of the motors before the motion has been transmitted through the gearbox. These motors can be positioned within $\frac{1}{6}$ of a revolution. The choppers for the two hand motors are mounted on the output shafts of the gearboxes and have a resolution of $\frac{1}{24}$ of a revolution or 15 degrees. (The deluxe hand provides a resolution of about $\frac{1}{10}$ of one degree!)

The Controller Card. The Rhino XR-1's controller card provides complete digital control of eight motors and monitors their optical choppers. It also has an on-board RS-232C interface that receives information from and sends information to the host computer. The ability to monitor the condition of six interrupts is implemented on the controller card: four of these interrupts are configured as microswitch monitors on the Rhino XR-1; the other two are available as needed. If necessary, all six interrupts can be made independent of the operation of the motors. Because the controller card provides such easy control of eight motors on a completely digital level, it can be used effectively by experimenters who wish to develop their own mechanical arms.

Operation of Controller Firmware

The program stored in the 8748 processor chip consists of a scanning routine. This scanning routine scans the host computer output present at the RS-232C interface and scans the operation of the optical choppers on up to eight motors. It also sends information from the Rhino XR-1 to the host computer when information is requested.

Six motors are on the robot. The basic package can control two additional motors. These motors may be coordinated with robot operation. This can be most useful and someday most robots will be able to do this.

The program maintains the motors positions by constantly monitoring the status of the two photo-transistors on each chopper. Essentially, the microprocessor determines the direction in which the motor is rotating and keeps track of how many holes have gone by on the chopper as the rotation proceeds. If the motor overshoots the required number of holes, the motor is automatically reversed and backed up the required number of holes. This provides digital damping. The controller attempts to constantly maintain a zero error position for each motor. This is done automatically and is transparent to the user. New commands are added to the error signal when received.

Should an extremely heavy load cause the arm to drop, the change in count will be picked up by the controller and the controller will move the affected motor in the correct direction to bring the count back to the zero error position. If you move a motor chopper by hand, the motor will automatically start and oppose the direction in which you are moving the chopper. Most observers are surprised at the force required to oppose the operation of the motors. We are using very high reduction ratios, and it is necessary to be careful.

Conversing With the Controller

The host computer can give the Rhino XR-1 controller four separate commands:

- START a specific motor and move it a specific amount. The amount is expressed as a number of chopper holes.
- QUESTION. What is a motor's error condition now? How far is it from its stopping point?
- STATUS. What is the status of the six microswitches (another name for interrupt lines)?
- STOP a specific motor instantly and set its error condition equal to zero.

All commands must end with a carriage return and line feed. The commands are used in the following way:

The Start Command. The start command consists of a two to five character string that is organized as follows:

a. The first character can be from A to H (upper- or lowercase) and identifies one of the eight motors that the card is controlling.
b. The second character can be a plus (+) or minus (−) sign and indicates the direction in which the motor is to run. The plus (+) sign may be omitted.
c. The last three characters can be anywhere from 0 to 127 and indicate the number of counts that the motor is to be moved. The 0 count conveniently defines a null instruction. The count of 95 is the maximum count that should be sent out at any one time. (It is actually possible to send 127, but it is not recommended unless you know exactly what you are doing. More on this later.)

Motor Status Information. The QUESTION to request motor status information from the controller is a two-character string. The first character is a letter from A to H and the second character is a question mark (?). With this instruction we are asking the computer if the motor under consideration has reached its zero error position. If that motor has reached a zero error position, the controller will answer with a hexadecimal 20. If the motor has not reached its zero error position, the character returned will be hexadecimal 20 plus the number of holes still to be traversed, all expressed in hexadecimal notation. Hexadecimal 20 is added to the error condition because some computers use all codes below 20 as control codes and this can cause problems. The information received is valid for a very short time since the motor is running at full speed. Making use of this information has to be done quickly and with some cunning. Essentially, this allows us to send the motor under consideration the next instruction without waiting for the motor to stop. It also allows us to inhibit the next instruction until the error count is low enough so that we will not go over the 127 count which would flip the direction bit.

The fact that hexadecimal 20, decimal 32, has been added to the count returned means that the highest count we can receive is 127 − 32, or 95. After that, the numbers wrap around and 96 is expressed as −32, 97 as −31 and so on to −1 which represents 127. If your computer will not accept numbers below hexadecimal 20 because it uses these numbers as control codes, then you have to limit yourself to keeping all conversations with the robot to numbers below decimal 95. If you never ask a motor to move more than 95 holes, it can never be more than 95 holes from its destination and you will never encounter control codes in your conversations with the Rhino controller card. Note that the wraparound gives you negative numbers from −1 to −32. On some computers, however, the (−) bit is not processed properly and this can cause problems.

Interrupt Status. The controller card has the ability to detect the grounding of six lines. These six lines are organized as the limit switch microswitches on motors C, D, E, F, G, and H, the six motors that allow full power to be applied to them (not the two small hand motors). As presently configured, these six lines are used to detect the microswitch closures for these six motors. They do not necessarily have to detect microswitch closures — they can be used to detect the grounding of these six lines for any purposes. The instruction for asking the status of these limit switches is "I". A question mark is not required. Upon receiving the "I" command, the Rhino controller card responds with a hexadecimal number. In decimal equivalents, if all the microswitches are open, you will get a 95 back from the controller. If any of the microswitches are closed, the number will change as follows:

- If switch C is closed, decrement by 1.
- If switch D is closed, decrement by 2.
- If switch E is closed, decrement by 4.
- If switch F is closed, decrement by 8.
- If switch G is closed, decrement by 16.
- If switch H is closed, decrement by 32.

Thus, if all the switches were closed, we would decrement by 1 + 2 + 4 + 8 + 16 + 32 = 63 and the answer would be 32. The number can never be below decimal 32 (hexadecimal 20), the magic number we wanted to stay above earlier.

Stop a Motor and Set Error Signal to 0. This is a two-character command. The first character consists of a letter from A to H and the second character consists of an X. Thus, if motor C is running or stalled under load and CX is sent to the controller, the controller will immediately set the error signal for motor C to 0 and that motor will be stopped. This

CIRCLE 4

LOGICSIM: For board or IC logic designers

Before you Troubleshoot Another Breadboard Take A Look At Our Professional Logic Simulator

Guarantee funtionality of complex circuits, increase design productivity, cut down product development cost.

Available now is LOGICSIM™ the first low-cost logic simulator with capabilities comparable to industry standard simulators, those costing up to $50,000. LOGICSIM makes complicated design, test and verification easy. Nine logic states, expandable macro network library, ability to simulate TTL/NMOS/CMOS/PMOS, three-state, wired-or and transfer gates, plus numerous other features make it the most powerful and usable design tool on the market. For more information, call (408) 578-8096. Or write:

E/Z ASSOCIATES, 5589 STARCREST DR, SAN JOSE, CA 95123

U.S.A. Price is $460. CP/M based 64K systems (Apple's with CP/M card). CP/M is a registered trade mark of Digital Research Inc. Apple is a registered trade mark of Apple Computer Inc.

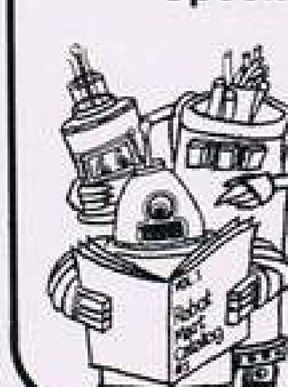
SYM-1

Single-Board Control Board

SYM-1 is the *first choice* for an on-board robotic control computer.

- Powerful 6502 microprocessor
- Bright 6-digit hex LED display
- Audible response keyboard
- 4K byte ROM resident monitor
- 1K byte Static RAM, with sockets for immediate expansion to 4K total expansion to 64K.
- Audio cassette interface with remote control, in either 135 baud (KIM-1 compatible) or 5 speed 1500 baud.
- RS-232 or 20 ma TTY interface
- 15 bidirectional TTL lines for user applications

Special Introductory Offer—$219



THE ROBOT MART

19 West 34 Street,
New York, NY 10001
212/695-5108

Send $3 for 1 year subscription to Robotics Catalog.

CIRCLE 10

command is of particular interest as it allows us to turn off stalled motors and to effect other recovery procedures that are needed from time to time. This command can stop the hand closing motor and remember where the hand stalled on the object being handled.

These four commands constitute a comprehensive language capability between the host computer and the controller for the Rhino XR-1. A very sophisticated capability can be designed with these commands because they give complete control of the motors. The experimenter can design still other commands for his or her own firmware: your imagination is the only limit. The flexibility of the design that we have created is the key to allowing this implementation.

The Intel 8748

The controller design is based on the Intel 8748 microcomputer. The Intel 8748 is a new NMOS chip that allows the design and development of sophisticated control systems with relative ease. The chip consists of:

- 8-bit processor which allows powerful control.
- 1K by 8-bit erasable read-only memory. Ease of programming on-board.
- 64 by 8-bit programmable data memory which provides adequate programmable memory.
- 27 I/O lines, for lots of I/O capability directly from the chip.
- 8-bit timer/event counter. Not used in controller at this time but can be used for fantastic feedback possibilities from the Rhino XR-1. Can be used to ramp motors!
- Single 5 V power supply requirement for convenience.
- More than 90 instructions constitute a powerful instruction set.
- All packaged in 40-pin DIP package.
- Available in both an EPROM and ROM version for economy.

Firmware Operation

The on-board firmware is designed to perform two functions: to provide an RS-232C interface for conversations back and forth with the host computer, and to monitor the operation of eight DC motors and their optical choppers.

The microprocessor on board the controller monitors the situation at the eight DC motors as follows:

The two photo-transistors on the choppers at each of the motors are designed to provide two signals that are 90 degrees out of phase. The two signals are read by the microprocessor as digital signals that look like 00, 01, 11, 10, 00 . . . when the motor is running in one direction, and look like 00, 10, 11, 01, 00 . . . when the motor is running in the other direction. The microprocessor uses the generated sequences to determine the direction in which the motor is turning and to update the current status of the error signal for each of the motor positions.

When the motor reaches its zero error position, the microprocessor automatically turns the power off to that motor. Should the motor be disturbed by a load, or overshoot its destination, the microprocessor will immediately detect the change in the error signal and automatically take corrective action. Once the microprocessor has been told to move a motor a certain number of chopper holes, the execution of the command is completely automatic. The one exception is that an overload or stall condition will cause the microprocessor to keep trying to get to the zero error position. The QUESTION command will tell you when this stall condition is anticipated, and corrective action can be taken automatically under software control. The power supply is not overloaded. The motors can be protected.

For more information on the operation of the firmware, see *A Hands-On Introduction To Robotics, The Manual for the XR-1.*

A Complete Robotics System

The availability of interchangeable hands, additional fingers, and other accessories make the Rhino XR-1 a complete robotics system.

Hands. Two basic types of hands are available for the Rhino XR-1: the standard hand (comes with the Rhino XR-1), and the deluxe hand (currently available by special order, but in the future will be available off the shelf).

The standard or H-1 hand is the lighter and smaller of the two. It has two small plastic servomotors to power the rotation of the wrist and the closing of the fingers.

The deluxe hand is the same width as the standard hand, so it fits in the same space, but it has slightly more depth and less height. It has two all-metal DC gearbox motors that are smaller versions of the four larger motors that run the arm and base. These motors are arranged to give more torque, more power, and better resolution to the accessories that fit on the hands. The choppers run at full motor speed and provide extremely fine control.

Both hands use the same accessories and are interchangeable, except, of course, for the parts that constitute the hands themselves.

Fingers. A comprehensive family of fingers can be ordered, and they will be available off the shelf soon. These fingers and palms, which do the actual grasping of the various manipulated objects, are available in the following configurations:

The *basic fingers* (F-1) come standard with the Rhino XR-1. These consist of a two-finger arrangement in which the tips of the fingers stay parallel to one another. Other items can be attached to these flat fingertips to allow the hand to grasp a variety of objects.

The *1.5X fingers* (F-1.5) are 1.5 times larger than the standard fingers to allow slightly more flexibility and grasp of larger objects. The fingers come with their own wrist plate and nut and easily replace the standard fingers.

The *2X fingers* (F-2) have been increased in size by 2.0 times to allow even more flexibility in grasping larger objects. The fingers come with their own wrist plate and nut and easily replace the standard fingers. As might be expected, increasing the size of the fingers decreases the force at the fingertips.

The *triple fingers* (F-3) are essentially the standard F-1 fingers but with three sets of fingers, which makes it a much better hand for grasping round objects. Additionally, objects tend to center themselves in the hand when grasped, and this is a tremendous plus when it comes to writing pick and place routines. The fingers come with their own tendon shaft, wrist plate, and nut, and easily replace the standard fingers. As with all finger accessories, these can be used on both hand mechanisms.

The *magnetic fingers* (F-MP) consist of a magnetic pickup unit that can be attached to the hand mechanism to pick up objects made of ferro-magnetic materials. There is no need for a tendon on this hand since no closing is involved. A dummy tendon shaft is supplied. Power to the hand is not supplied as such. One of the extra motor lines is selected to power the relay supplied to switch the hand. (Under limited conditions, it is possible for sophisticated users to use the controller power supply for weak applications.) At the owner's option, the magnetic hand can, of course, be hooked up to a separate, powerful power supply (recommended procedure).

The *clamshell fingers* (F-CS) consist of a small clamshell-type arrangement whose operation is similar to the basic hand supplied with the unit. Instead of fingers, however, this arrangement has a clamshell that allows the Rhino XR-1 to pick up objects that would otherwise be difficult to grasp.

The *shovel fingers* (F-SV) consist of a shovel-type hand. The tendon nut is not used as a closing device, but rather as a tilting device not unlike a digging bucket. A dummy tendon shaft is supplied with the unit.

The *hollow body hands* (F-MD) unit consists of a hollow hand body that takes the place of the standard hand and allows the user to position a small MotoDremmel (R) tool in the hand of the Rhino XR-1. This can be used for grinding and carving applications.

Sandhu Machine Design Inc. designs and builds other hands and robots to customer specifications. Please write for quotes. A sketch can be very helpful as can a sample of the object you want the hand to manipulate.

Other Accessories

The Rhino XR-1 system can be supplied with the following accessories, currently built to order, but available off the shelf in the near future.

Carousel. A small carousel that can be easily positioned at the front of the robot is provided as a convenience item to robotics investigators. The carousel consists of a platter to which various items can be attached. It rotates under command from the controller and can be connected to the seventh or eighth motor control point on the controller board. This allows the experimenter to perform complicated pick and place experiments on a moving carousel under the complete control of the host computer (without having to go through all the work that is normally required to hook up a separate carousel and attach switches and power supplies to it).

Conveyor. A small conveyor that can be positioned in front of the robot base is provided as an experimental accessory for serious experimenters. It can be used to simulate larger conveyor systems that might be encountered in an industrial environment. The conveyor is about three inches wide and about 18 inches long, so it is within easy reach of the robot at either end of its travel, and it is powered by a motor that can be run from port 7 or port 8 of the controller board under complete control of the host computer.

Linear Base. For the ambitious experimenter, we provide a linear base that allows the robot to be mounted so it can traverse laterally about 18 inches. The motor on this base can be run from port 6, port 7, or port 8 of the robot. If port 6 is used, the waist is disabled and lateral motion takes the place of the waist operation. If port 7 or port 8 is used, waist operation can continue along with lateral movement of the entire robot.

If the lateral moving option is used, special precautions must be taken to allow the controller and the power supply to follow the robot. These problems are left to the experimenter to solve, as Sandhu Machine Design Inc. does not currently provide any appropriate accessories (although if the demand is sufficiently high, we may provide these accessories in the future). The most economical solution is to use long

WE'RE OPENING THE HIGH FRONTIER.

JOIN US

Robots are the fingers and feet of the future. The leading edge research needed to break out into the frontiers of space is happening now. And people like you are the support of that research.

Your interest in and knowledge of robotics puts you at the front of the push to tomorrow. Find out more about the research the Space Studies Institute in Princeton, New Jersey funds by circling the Reader Service Card Number.

Join us in opening The High Frontier.

CIRCLE 14

cables on the motors. These can be ordered from Sandhu Machine Design in lengths up to 10 feet.

Power Supply. A bipolar power supply is provided with the Rhino XR-1, and this allows the design of a simple reversing mechanism for the DC motors. This is effected by grounding one side of a motor and connecting either + 12 V DC or − 12 V DC to the other side of the motor.

The standard power supply provides 3.5 amps at each voltage. The minimal 5 V needs of the logic components are taken from the + 12 V side of the supply and regulated by an on-board regulator. This is adequate for most purposes, but if extremely heavy use is envisioned, a heavier power supply (7.0 amps) is available.

Experimentation

The Rhino XR-1 is designed to allow experimentation at three separate levels of interest: hardware, firmware, and software.

Hardware. The unit can be easily modified by the experimenter as construction is almost entirely of free machining aluminum that requires minimal shop facilities. The entire unit can be taken apart and assembled at will, and a tool kit is provided. The availability of the extra hands, fingers, conveyors, carousels, and lateral slide base allow still further experimentation.

The removal of one screw detaches the wrist and fingers from the hand. Another hand can be attached immediately and the experiment can proceed.

A motor can normally be detached with two screws. Should the experimenter choose to investigate different motor types, it would be a fairly straightforward matter to attach the new motor to the Rhino XR-1.

The optical encoder disks are attached with three screws, and should the experimenter wish to modify the encoding procedures, the process is simple enough.

And on and on!

Firmware. Detailed listings of the firmware that controls the controller card are provided. The serious experimenter can write new firmware to suit his or her own special interests. A completely new language may be developed, or new commands added. The firmware provided with the Rhino XR-1 gives the guidance you need to see how you might go about developing your own software. Adding to the software is fairly straightforward.

Software. Because the Rhino XR-1 can be run from an RS-232C port from any computer, software can be developed in any higher-level language or machine language. Conversations with the Rhino XR-1 are initiated with simple PRINT commands, none of which are more than five characters long. The host computer can think that the Rhino XR-1 is a serial ASCII printer or similar device. All responses from the Rhino controller are only 1 byte long, so no complicated busy signal and handshake logic is needed or implemented in the controller design. The processor is fast enough to keep up with all commands from the host computer in real-time at 9600 baud, so special circuitry is not required here either.

Availability

The Rhino XR-1 is available for immediate delivery. For more information, write: Sandhu Machine Design Inc., 308 South State St., Champaign, Illinois 61820, or call (217) 352-8485.

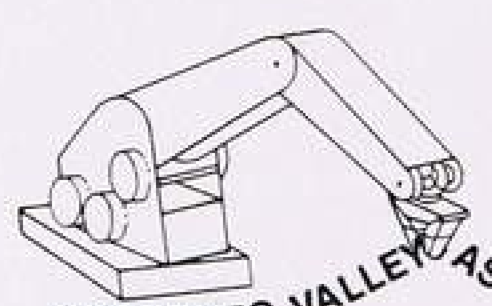
SVA
CONSULTING ENGINEERS
COMPUTER APPLICATIONS

ROBOTICS DEVELOPMENT

Robotic Arms for industrial, scientific and research applications.

Experience in applying the latest in micro-computer technology to San Diego's industrial and business problems since 1974.

- Develop your ROBOTICS capability
- Expand your Education and Training
- Economical and Portable
- apple II and TRS-80 interface
- Call on our experience — 452-0101



$1,995.00

Complete with Interface

SORRENTO VALLEY ASSOCIATES, INC.

11722 Sorrento Valley Rd., San Diego, CA 92121

The **Robotics VI Conference and Exposition** was held March 1 to 4 at Cobo Hall in Detroit, Michigan. For further information, contact the sponsor: Society of Manufacturing Engineers, One SME Dr., POB 930, Dearborn, MI 48128, (313) 271-1500.

For information on **Applied Machine Vision**, April 6 and 7 at the Sheraton Hopkins in Cleveland, Ohio, contact: Robotics International/SME, One SME Dr., POB 930, Dearborn, MI 48128, (313) 271-1500.