

JAMES A. GUPTON JR.

WITH THE UNICORN-1 ROBOT OPERATING under radio control, what now? Why, *computer* control, of course! This part will deal with that subject, although, because of its complexity, only in general terms.

For those of you already involved with computers—micro or otherwise—much of what will be discussed here may seem elementary. For those who have not yet been exposed to that fascinating area of electronics we will try to keep things as simple as possible.

What will be covered here will be the *concepts* involved in having the actions of a robot determined by an electronic device rather than by a human operator. That's where much of the challenge of computer control comes in.

A human can exercise his judgment—without necessarily having to think about it—and change the robot's actions to meet the circumstances. The computer also has to exercise judgment, but before it can do that it must be taught—or programmed—*how* to make judgments; that involves a great deal of highly detailed programming.

For those of you who are unfamiliar with computers, it is not enough just to connect a computer to the robot and say, "Go ahead . . . do your stuff." Every action must be pre-planned, and, more important, every consequence of every action must be considered and the appropriate reaction prepared.

That is one reason why we will not present specific programs for robot control but will, instead, talk about the way those programs will have to function.

Methods of computer-control

To put it broadly, there are three ways that a computer can be used to control the robot's actions.

The first, and simplest, would substitute a computer, *located outside the robot*, for the command consoles described earlier in this series. That computer would be linked to the robot either by cable or by radio.

The program for that system would be fairly simple and would allow the operator to type in a command, to which the robot would respond. For example, entering "GO" or "G" would cause the robot to move forward; "TURN LEFT" or "L" would cause it to turn to the left, and so on.

That elementary program could be modified to operate with a speech-recognition device—several of which are available for a couple of hundred dollars—to allow the robot to respond to the spoken

Computer Control for the UNICORN-1 ROBOT

Part 10—If your robot is going to use computer control, here are some thoughts on the subject that will help you in setting up your system.



word. The vocabulary would be limited (but adequate) but the commands would have to be given to the external computer, not to the robot directly.

The second system would be a program, or series of programs, that would command the robot to perform a predefined sequence of actions.

For example, the robot might be instructed to move forward for ten seconds, stop, raise its right arm in a salute, beep its horn, and then turn around and return to its starting position.

Such programs could become very elaborate, but have a major drawback. Unless the robot is equipped to respond to its environment (and, so far, it isn't) any unknown factor that enters the picture could have serious consequences.

Using the program above as an example, suppose that, unknown to you, the robot is facing a brick wall, five feet in front of it. Shortly after the robot begins to carry out the instructions given to it by the computer, it will run smack into that wall! Not only will that interfere with the rest of the program, but it can also cause damage to the robot and, possibly, the wall. Or maybe, instead of a brick wall, there's a person or a piece of furniture in the way. The overall damage—and its consequences—could be considerably more serious.

In any case where the robot is operating without human intervention, provision must be made for the program to be overridden!

Any program of that nature must contain some means for the human supervisor to stop or alter the robot's actions at any time. That is one reason that the "drop-dead" circuit was included on the latch board (Part 9)—one command would activate that circuit and cause the robot to stop in its tracks, should any unforeseen circumstance arise.

The third method of computer control, and the most fascinating, involves the robot having its own, on-board, computer. The precautions given for the second method also hold here. We'll talk about that method in more detail shortly.

Interfacing

Whichever method is chosen, the robot must be equipped to respond to (and, perhaps, "talk back" to) the computer. Fortunately, the circuits already being used by the robot are designed with that in mind.

There are two formats that computers can use to output data or to receive it: *parallel* and *serial*. The parallel format is always used by the computer internally.

The unit of information that the computer uses for communication is called a *byte*. A byte is made up of eight bits (*binary digits*)—each one either at a logic-"high" or logic-"low" state—and the computer operates on all eight bits at once. Frequently, when a computer is used to operate a printer, the parallel for-

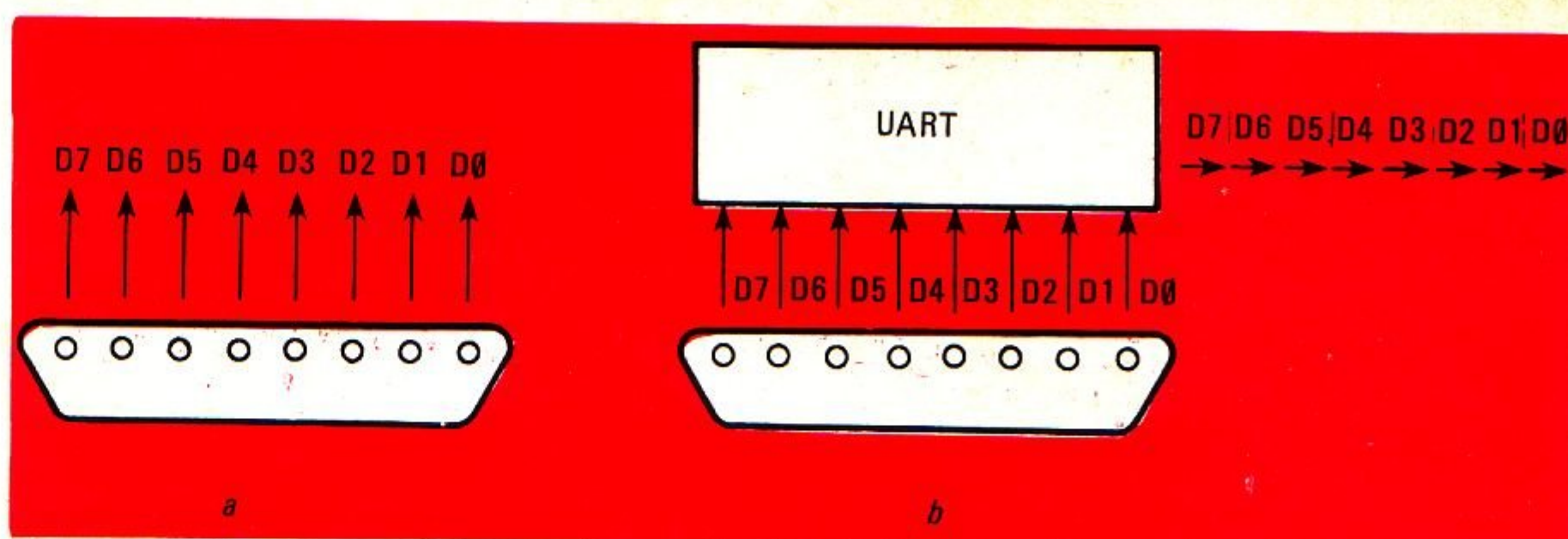


FIG. 83—ALL THE BITS of a byte are sent simultaneously in parallel communications (a). A UART (b) converts parallel data into serial data for transmission over a single line.

mat is used and eight lines are used to connect to the printer—one for each bit of the byte.

On the other hand, sometimes it is convenient—or even necessary—to transmit computer data using only a single line (by telephone, for example). In that case, the serial format is used. The computer takes each byte and sends it out bit-by-bit, one after the other, indicating the beginning and end of each byte. At the other end, the eight bits are received in the order in which they were sent; when they have all arrived, they are used in parallel. Both of those systems are illustrated in Fig. 83. The device that performs the parallel-to-serial and the serial-to-parallel conversions is known as a UART (Universal Asynchronous Receiver/Transmitter). UART's would be used if commands were transmitted to the robot by radio.

If you connect your computer to the robot by means of a cable from the computer's parallel port, it would be a good idea to use twice as many lines as necessary (16) and ground every other one. That will help keep electrical noise from getting mixed in with the data.

(For more information on how computers operate see "Your Own Computer" in the October 1980 issue of *Radio-Electronics* and the article on assembly language computers on page 45 of this issue.)

The decoder-, latch-, and relay-driver-boards in the Unicorn-1 use parallel data. Using the same technique as used with the 7402 IC's on the decoder board, any two bits of an eight-bit word (byte) can be NOR'd or NAND'd to produce a single control bit for the relay-driver board. If you're knowledgeable, more complex and versatile encoding/decoding schemes can be used.

Which computer?

There are two classes of computers that must be considered: those for external use and those that can be mounted on-board the robot.

Almost any computer that has at least one parallel port can be used for the first purpose and it is not our intention to single out one manufacturer's over the other. If you are contemplating buying a computer, refer to the articles mentioned above.

The important thing is that the com-

puter be equipped with a parallel port and that it be flexible enough to meet your needs—present and anticipated. For example, if you are considering using voice control, make certain that there is a speech-recognition board available for your computer.

It should be noted that some computers—such as the Radio Shack *TRS-80* and the Commodore *PET*—do not have parallel ports as such, but that their expansion connectors—frequently used to connect to printers—are actually just that. The thing to look for is eight *data lines*, usually designated "D0" through "D7." If you have those, you have your parallel port.

You will also want a cassette and/or disk interface to allow you to save programs that you have written for the robot.

One thing you should avoid are inexpensive computers that are actually glorified video games. They generally will not have the facilities you need and it will prove difficult (or impossible) to add them.

The other possibility is a single-board computer that can be mounted in the robot. In addition to a parallel port and cassette interface, that computer must also have a hexadecimal ("hex") keypad for programming, and some kind of LED display, if it is not going to be used together with an external computer. An example of how such a computer would be interfaced to the robot is shown in Fig. 84.

A good computer for the purpose is the *KIM-1*. Unfortunately, that computer was recently discontinued; but you may still be able to find one here and there. Other possibilities include the *SYM-1* (a sort of super *KIM*), the *ELF-II* or the *Explorer/85* (keypad version). Again, refer to the article on page 45. Both the *ELF-II* and the *Explorer/85* are manufactured by Netronics, 333 Litchfield Road, New Milford, CT 06776. The *SYM-1* is produced by Synertek Systems Corporation, P.O. Box 552, Santa Clara, CA 95052.

Bear in mind that some of those computers may require a power supply other than 5- or 12-volts DC. In that case a power inverter (see Fig. 85) can be used to turn the robot's 12-volt supply into 117 VAC, which the *computer's* power sup-

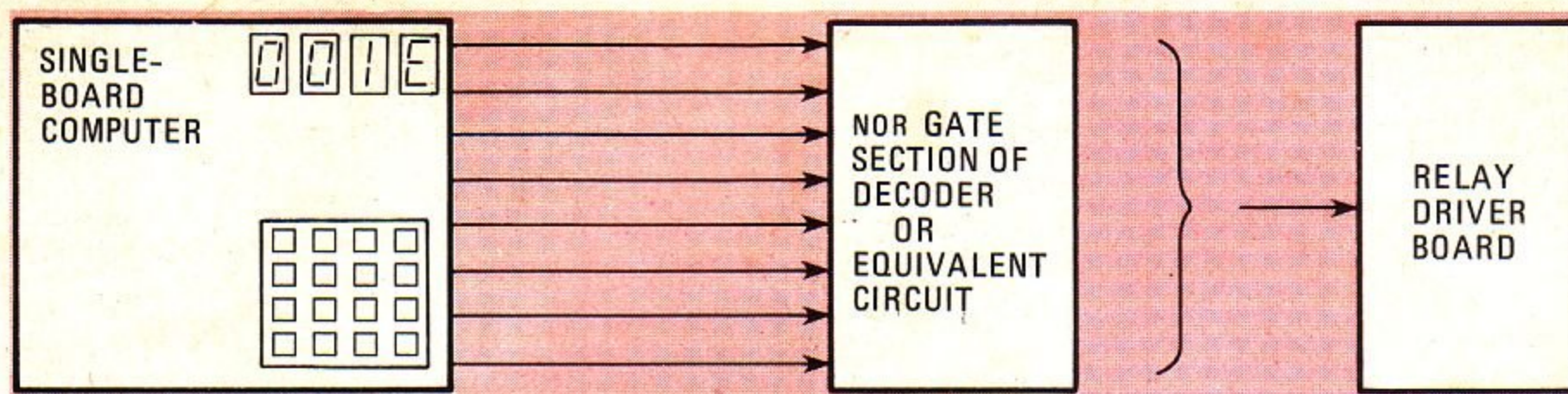


FIG. 84—SINGLE-BOARD COMPUTER can be connected to NOR gate section of latch board or to an equivalent circuit designed to give a single output from a two-bit input. That is only one of many possible schemes.

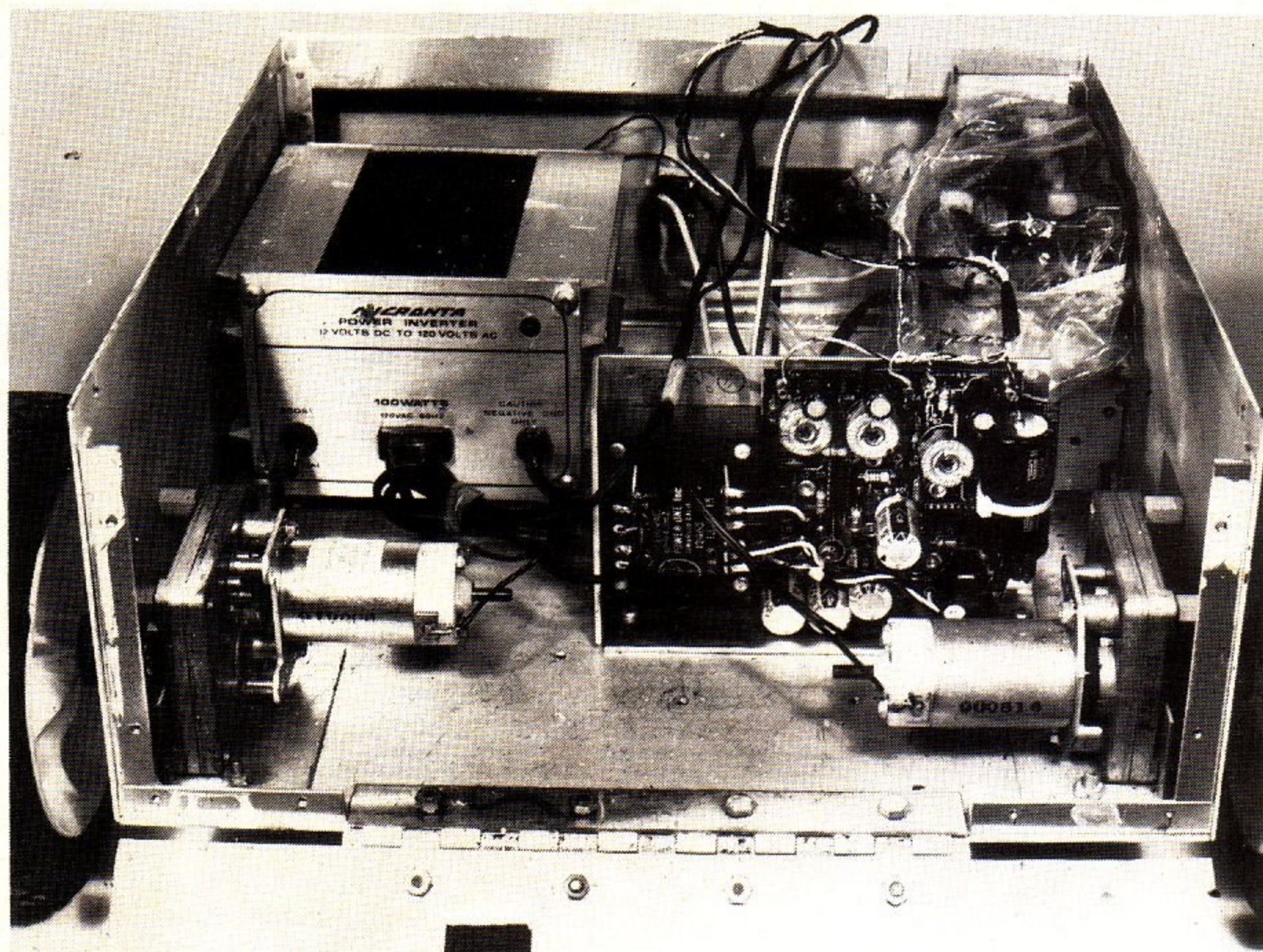


FIG. 85—AN INVERTER (left rear) may be needed if the on-board computer has power requirements other than +5 or +12 volts DC.

ply can then convert readily to its own requirements.

Finally, if you already own a computer but intend to install another in the robot, it would be a good idea to make sure that both computers use the same-type, or compatible, microprocessors. The *KIM-1* and *SYM-1* use the 6502, which is also found in the *Apple II*, *OSI Challenger(s)* and the *PET*, and the *Z-80* in the *TRS-80* is compatible with the *Explorer/85's* 8085.

The 1802, used in the *ELF II* and in RCA's *VIP*, is not normally found in larger computers, but that does not mean that an 1802-based single-board computer should not be used in the robot.

The fact that both of your computers use the same microprocessor means that both of them speak the same language, at the microprocessor level. That, in turn, means that you can use your larger computer to develop and debug (troubleshoot) programs to run on the robot's computer and to *download* (transfer from the larger to the smaller computer) those programs, either directly or, if the cassette interfaces are of the same type, from tape.

The programming itself will also be

easier, since—assuming that your programs are in machine language and not in BASIC—you will be able to use an assembler, making your work go more quickly and also making it easier to follow the flow of the program.

Programming

As you may have gathered by now, it would be impossible to present computer programs for robot-control, there being so many variables involved.

If you are working with an external computer, you will probably want to work in BASIC or another high-level language, using the OUT command, or its equivalent, to transfer data to the robot.

As mentioned above, the on-board computer will almost certainly have to be programmed in machine language. It's more difficult to work with than BASIC, but it *does* have advantages. Programs take up much less memory space, and also run more efficiently. You may even want to write your "big-computer" programs entirely in machine language through the use of an assembler.

This section has of necessity, been sketchy: after all, even books on the subject have not been able to cover the mat-

ter completely.

If you are going to use a computer with your robot, we recommend that you do as much supplementary reading as you can. Personal-computer magazines such as *Byte* magazine and *Interface Age* have had special issues dealing with robots, and the subject comes up frequently there and in other computer publications. Another good source of information that is often overlooked is your local library.

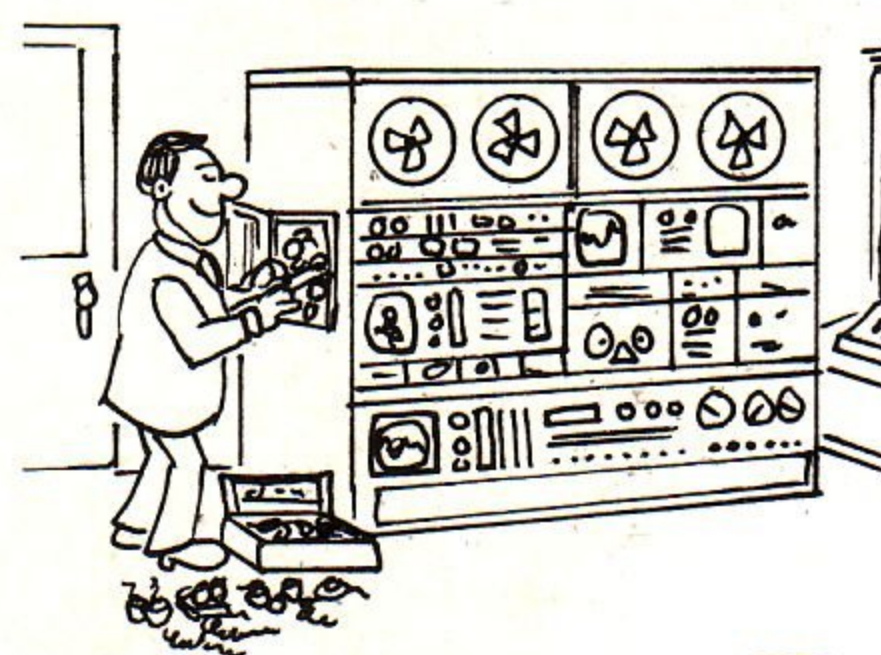
Todd Loofbourrow's book, *How to Build a Computer-Controlled Robot* (Hayden Publishing Company) contains a number of robot-control programs written for the *KIM-1* (or *SYM-1*) as well as a number of more generalized flowcharts. Much of the information presented there may be adaptable to your robot.

A very good—although rather technical—article on "An Interactive Programming Language for Control of Robots" by Li Chen Wang appeared in the September 1977 issue of Dr. Dobb's *Journal of Computer Calisthenics & Orthodontia*. It involves a robotic simulation on a computer's video display and its principles could be adapted to control a "flesh-and-blood" robot. (That issue, #18, Volume II, No. 8, is available in limited quantities from: Dr. Dobb's *Journal*, 1263 El Camino Real, Box E, Menlo Park, CA 94025 for \$2.50, postpaid, second class.) It's worth looking into for readers already familiar with computer programming.

In the next part of the Unicorn-1 series we will take a look at sensors. We will discuss sensors in general, and show you some specific examples that can allow your robot—and the computer that controls it—to respond to the world around it.

We would like to hear about how you're doing with your version of Unicorn-1. Write (and send photographs) to: **ROBOT UPDATE, Radio-Electronics**, 200 Park Avenue South, New York, NY 10003.

R-E



"Charlie's OK at fixing computers. He seldom does any damage that an electronic technician can't repair."

Part 11—The better your robot can respond to the world around it, the more useful it will be. Here are two sensors that will enable the robot to “see” and “feel” objects that are in its vicinity.

UNICORN-1 ROBOT

JAMES A. GUPTON, JR.

UP UNTIL THIS POINT, ANY REACTION THAT the robot has shown to events happening around it have actually been those of its operator. Radio- or computer-control has been possible only to the extent that the operator could observe the robot's environment and make the robot react to it. And, even operating in that way, there has been the danger that the robot could “stumble” into something that could not be seen by the operator.

In this installment of the Unicorn-1 series we'll describe two types of sensors that will enable the robot to detect objects in its immediate vicinity and to react to them.

The first is a contact-type sensor that will give the robot a limited sense of “feel” and allow it to know when it has bumped into something.

There are times, though, when it would be better for the robot to be able to sense when it was *about to bump into* something—running into brick walls is one thing; running into people, another!

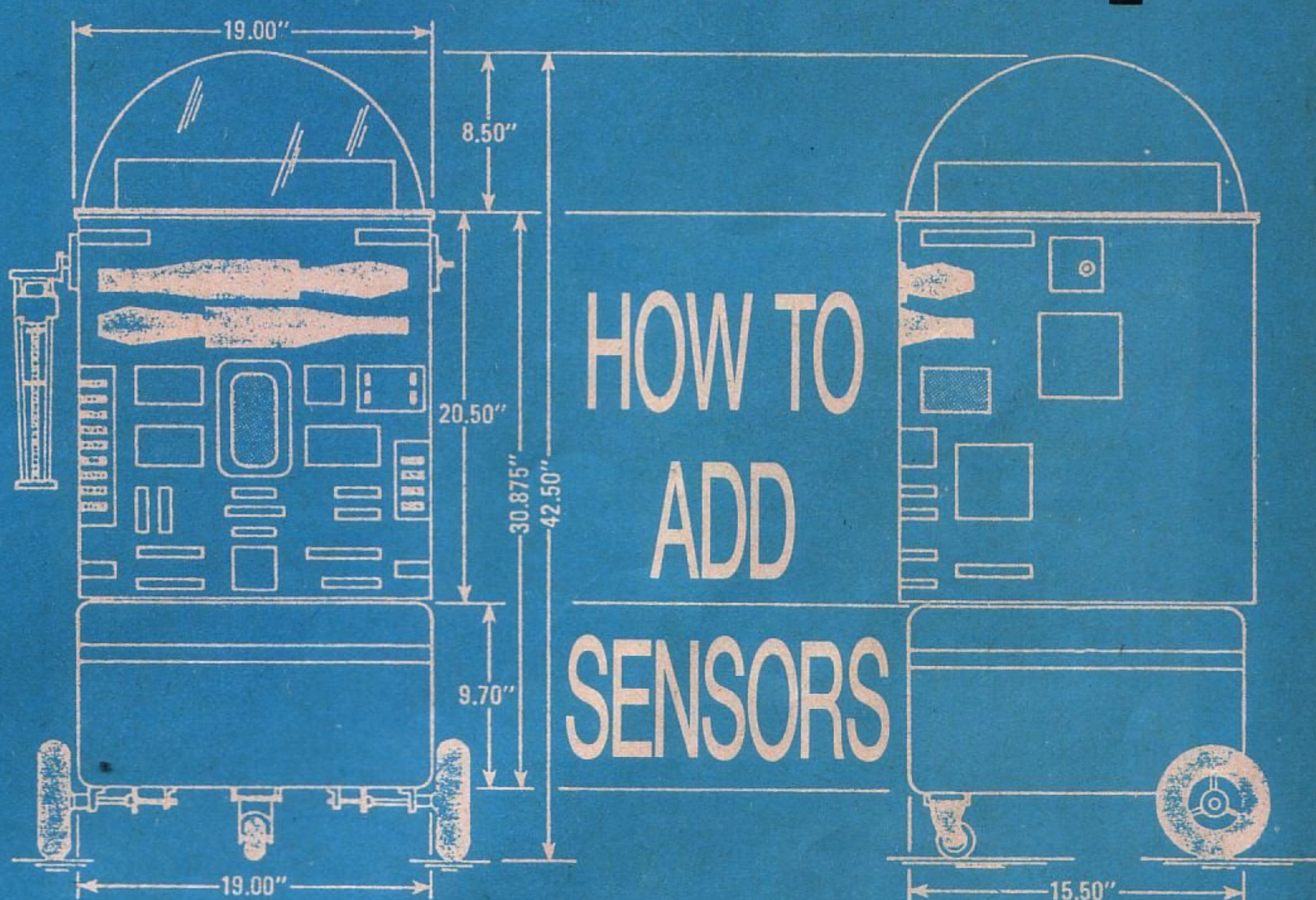
The second sensor, then, will be of the proximity-type, giving the robot a rather restricted sort of “vision.”

Contact sensor

The robot should be equipped with two contact-sensors—front and rear. They are extremely simple in design, as can be seen from Fig. 86, consisting of lever-actuated switches that are connected to rods projecting from the mobility base. Note that the rear sensor-rod is about twice as long as the one for the front sensor. This compensates for the fact that the large driven wheels of the mobility base may project behind it and, naturally, we want the sensor to come into contact with an obstacle before any part of the robot does.

The sensor rods are made from pieces of wire coat hanger, with the paint or lacquer removed to permit good solder joints. The front rod is about 1½-inches long and the rear rod about twice that length. The compression springs can be “liberated” from dried out ball-point pens. The springs are held in place by 4-40 washers soldered to the rods.

A 4-40 cap nut (the kind with a rounded end) can be soldered to the end of each rod to prevent it from scraping or impaling whatever it may come into con-



tact with. Better protection can be provided by applying a liberal amount of silicone sealant to the cap nut to provide a soft, protective surface.

Even better, a small bumper, with a soft covering made from a piece of foam rubber or inner tube, can be constructed and affixed to the end of the sensor rod.

The bushings that fit into the mobility base and allow the sensor rods to move in and out are nothing more than 10-32 \times 3/8 machine screws that have been drilled out with a No. 42 drill bit (use a drill press and vise, if you possibly can) and had their heads filed flat to remove the screwdriver slot. Leave enough head, though, to hold the screw in place. Use 10-32 nuts to secure the bushings to the mobility base.

A helpful hint: Fig. 86 shows a half-inch brass washer soldered to the “inside” end of each sensor rod. (The washers are especially necessary if more than one switch is used for each sensor—see below.) Those washers should be the last part to be attached.

The end-nut (or bumper), spring and 4-40 spring-stop-washer should be attached to the rod first, and the unit inserted into the bushing. Then, using a

wooden block to compress the assembly, and holding the brass washer with a pair of pliers, solder the washer to the end of the rod. Doing this will prevent your getting your fingers burned.

The brackets for the switches can be made from almost any material at hand—metal, plastic, or wood. They support the switches in the proper position and, if two switches are mounted side-by-side, allow the brass washer to contact both switch-levers at the same time.

The original Unicorn-1 used brackets made from scraps of 1½ \times 1½ \times .0625-inch aluminum, bent as shown in Fig. 86, and drilled to accept two 4-40 mounting screws. The section that fits flush with the mobility base need be no larger than ¼-inch if 4-40 hardware is used but should be at least 3/8-inch long for 6-32 hardware.

If the mounting holes in the switches are too small for 4-40 hardware, they can be enlarged with a No. 33 drill bit. Be sure to use a vise and to use either a hand drill or a *very slow* electric drill to prevent damage to the plastic switch case.

The completed front and rear contact-sensor assemblies are shown in Fig. 87. If larger switches are used, mounting brack-

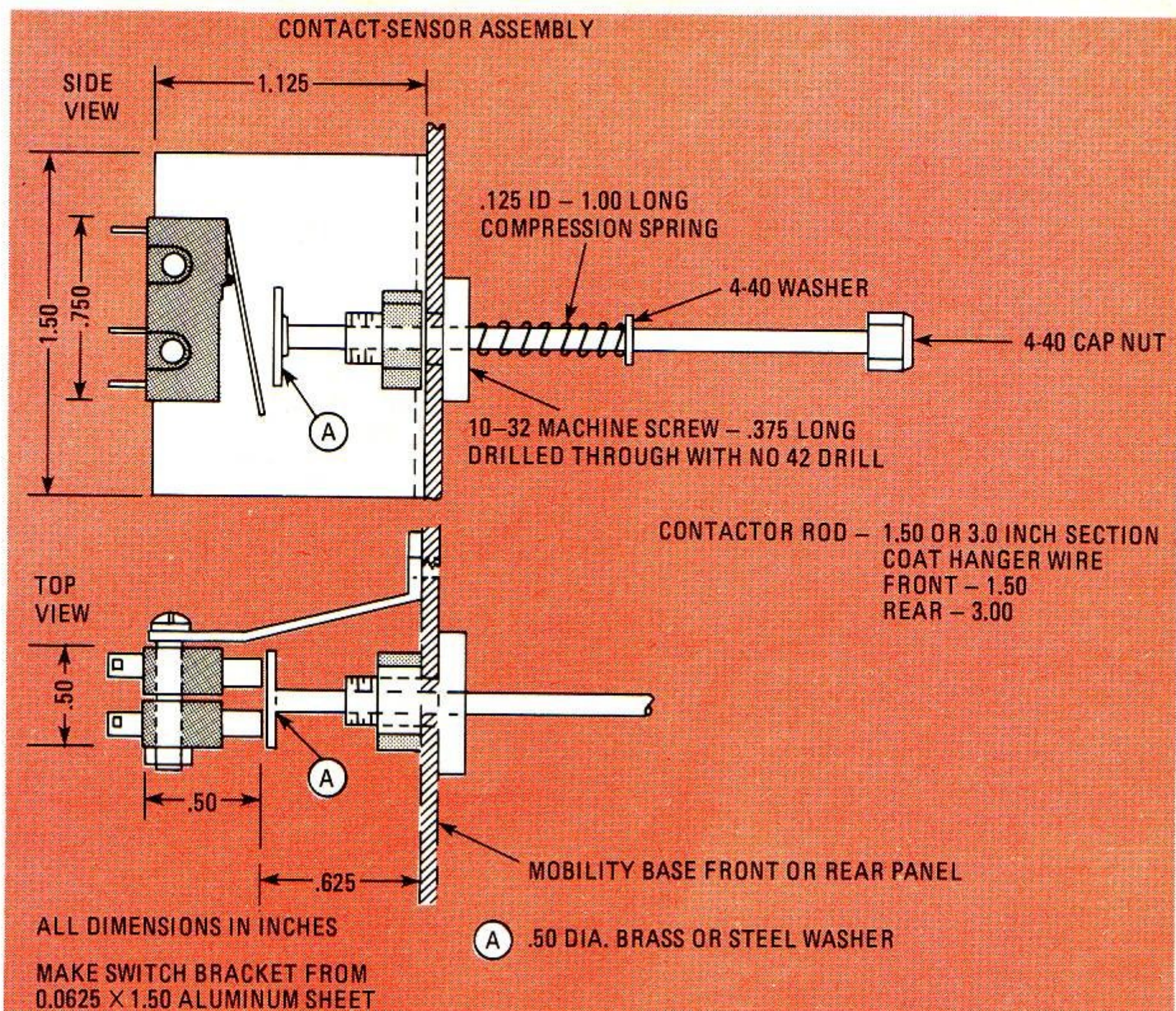


FIG. 86—CONTACT SENSOR tells the robot when it has bumped into something. Cap nut at end of rod should be provided with cushioning material (see text).

ets may not be necessary since the switches can be mounted directly on the bottom plate of the mobility base.

Connection of the switches will be discussed later.

Proximity sensor

While the contact-type sensor described above is extremely useful, there are times when it could prove embarrassing (or worse) to have the robot collide with something. It would be better if it

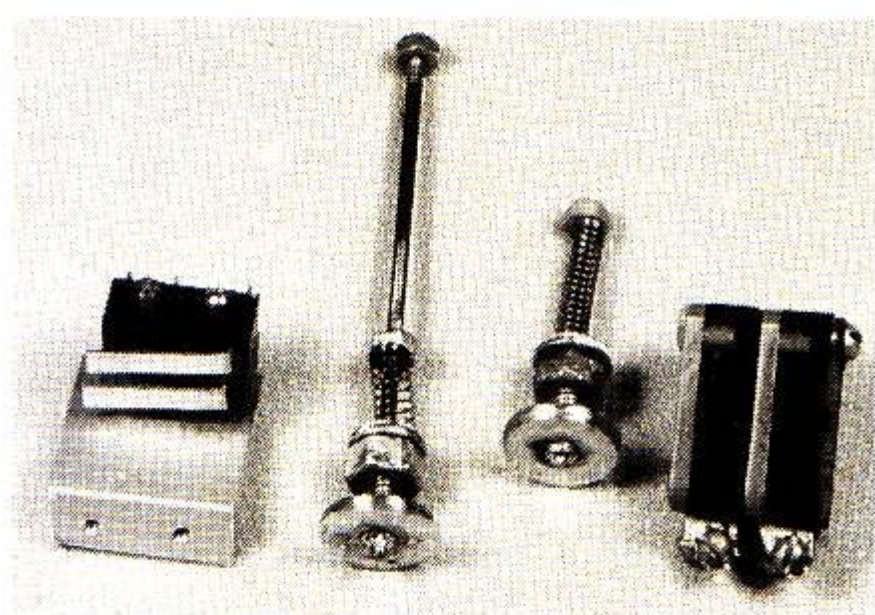


FIG. 87—COMPLETED CONTACT SENSORS used in original Unicorn-1. In this case, dual switch-assemblies were used.

could sense the proximity (nearness) of an object and either stop or, if under computer control, take evasive action.

Figure 88 shows how an infrared-light-type proximity sensor would work. The transmitter, mounted on the robot's right side and angled slightly inward, projects a beam of infrared light that will be reflected by a nearby object to the infrared detector, mounted on the robot's left side and also angled toward the target.

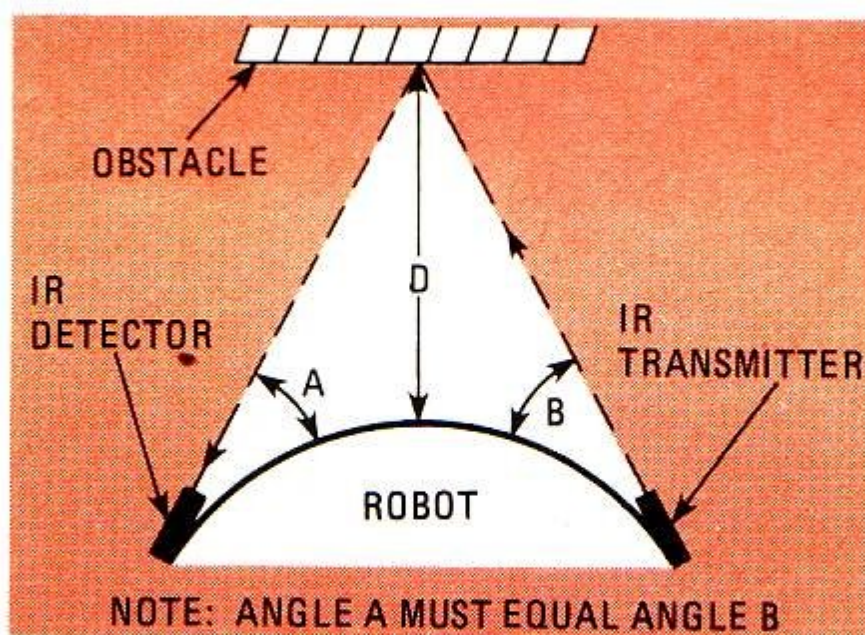
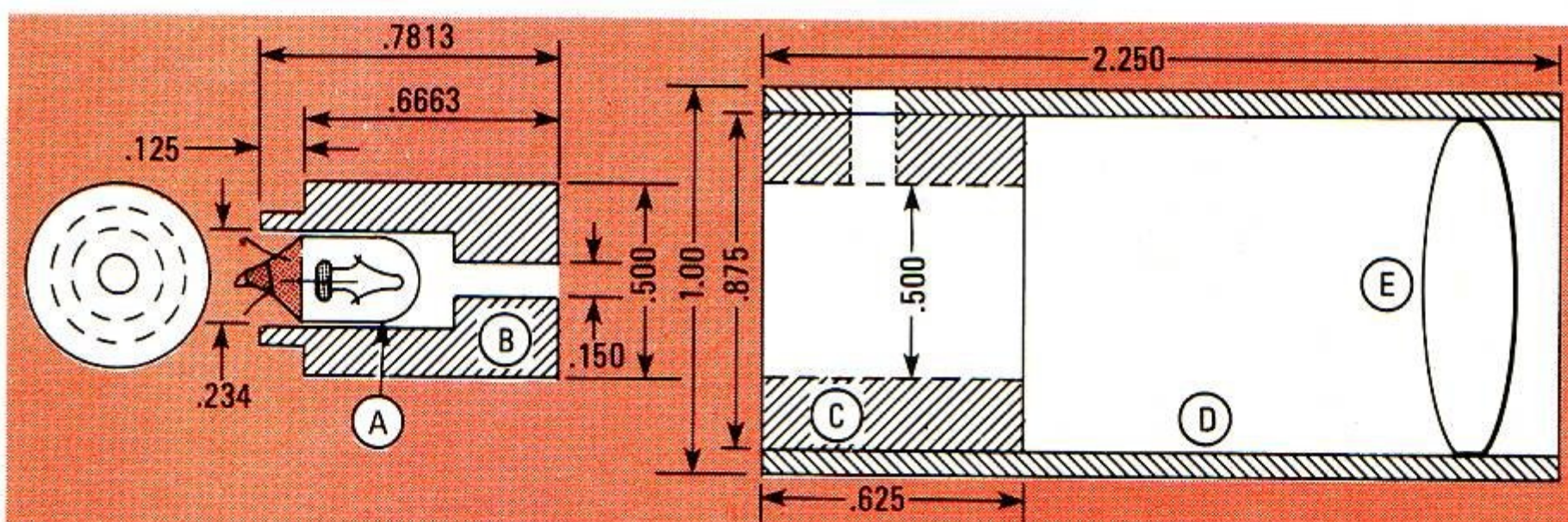


FIG. 88—INFRARED TRANSMITTER AND DETECTOR are mounted on sides of robot's body or dome. Angle A must be equal to angle B.



NOTES:

- (A) IR LAMP TYPE 2174D OR EQUIV.
- (B) IR LAMP HOUSING — .500 DIA. ALUM. ROD
- (C) .875 DIA. ALUM. INSERT — I.D. .502 LENGTH .625
- (D) LENS TUBE — 1.00 DIA., .0625 WALL — ALUM. TUBE
- (E) LENS, DOUBLE CONVEX, FOCAL LENGTH 48mm TO 55mm, DIA. 18mm TO 22mm
- (F) ALL DIMENSIONS IN INCHES EXCEPT FOR LENS

FIG. 89—INFRARED LAMP HOUSING can be held in position in lens tube by set screw, once proper position has been determined.

The distance between the transmitter and the detector, and the angle they form, will determine the distance, D, from the robot that the object can be sensed. The transmitter and detector must be aimed inward at equal angles for accuracy. (Remember that "the angle of reflection equals the angle of incidence;" and, the larger the angle, the farther away—up to about 20 inches in this case—an object can be detected.)

Using infrared light means that the system can be used under almost any lighting conditions since the infrared detector is not very sensitive to visible light. For that matter, the robot could even detect obstacles in the dark—it carries its own "flashlight."

Figure 89 shows the infrared-projector assembly used on Unicorn-1. When used with a lens, the 2174D infrared lamp generates a beam that is usable to a distance of about 20 inches.

The dimensions shown for the lens tube are only approximate, since there are so many variables (lens type, detection distance required, etc.) involved. The best way to find the dimensions you will need is to set up the lamp in its housing at one end of a ruler and to move the lens back and forth until you can see the beam focused into a spot on a screen or sheet of paper placed at distance D—your target distance. Don't forget that D is measured from the front of the robot, and not from the transmitter (or receiver).

Note the distance between the lens and the aperture of the transmitter assembly and make the lens tube that length. Critical adjustments can be made later by adjusting the position of the lamp housing slightly. The final assembly step, before mounting the projector on the robot, is to glue the lens in place in the tube using either epoxy or lens cement. Take care

PARTS LIST—CONTACT SENSORS

Item	Description or quantity	Source
Contact rod	1.5 inches	coat-hanger wire
"	3.0 inches	"
Mobility-base bushing	10-32 \times $\frac{3}{8}$ flat-head screw, (2)	hardware store
"	10-32 nut (2)	"
Cap nut	4-40 (2)	"
Compression spring	.125 I.D., 1 inch long (2)	ball point pen
Washer	4-40 steel (2)	hardware store
"	.5-inch brass (2)	"
Lever-type switch	2 or 4	Radio Shack (catalog No. 275-016) or equivalent
Switch bracket	1.5 \times 1.5 \times .0625 aluminum (2)	scrap or hardware store

PARTS LIST—PROXIMITY SENSOR

Item	Description or quantity	Source
TRANSMITTER:		
Infrared lamp	2174D, 12-volts	electronic-supply house
Lamp housing	.5-inch aluminum rod	hardware store
Lens tube	aluminum tubing, 1-inch O.D. \times 2.25 inches long	"
Lens & lens cement	double-convex, 48-55mm focal length, diam. to fit lens tube	Edmund Scientific 101 E. Gloucester Pike Barrington, NJ 08007

Item	Description or quantity	Source
RECEIVER:		
Sensor housing	5-inch aluminum rod & washer	hardware store
Lens tube	aluminum tubing, 1-inch O.D. \times 2 inches long	"
Lens & lens cement	double-convex, 20-30mm focal length, diam. to fit lens tube	Edmund Scientific
PC board	1 (\$2.50 + \$1.50 S&H if total order less than \$15.00)	Hal-Tronix P.O. Box 1101 Southgate, MI 48195
R1	68 ohms, $\frac{1}{2}$ -watt	
R2	22,000 ohms, $\frac{1}{4}$ -watt	
R3	10,000 ohms, $\frac{1}{4}$ -watt	
R4	4700 ohms, $\frac{1}{4}$ -watt	
IC1	7404 hex inverter	
Q1	2N2222 or equivalent	
Q2	FPT-100 or equivalent (Radio Shack 276-130)	
D1	1N5227 3.6-volt Zener diode	
D2	1N5231 5.1-volt Zener diode	
D3	1N4001, 50PIV, 1-amp diode	
RY1	5-volt DPDT DIP relay (Radio Shack 275-215 or equivalent)	

not to get any of the adhesive on the lens. The completed transmitter assembly is shown in Fig. 90.

A diagram of the infrared-receiver assembly is shown in Fig. 91. As in the case with the transmitter, the dimensions are approximate. To determine the final dimensions, a method similar to the one outlined above is used.

First, cover the aperture of the detector housing with a translucent material, such as *Scotch brand Magic Tape*, to make a focusing screen. Attach the detector housing to a ruler and aim the ruler at a white or light gray surface placed at distance D. When making your final calculations, don't forget about the angles involved! Move the lens back and forth along the ruler until a sharply defined spot is seen on the focusing screen. The distance between the lens and the end of the detector housing will determine the length of the lens tube.

As in the case of the projector, cement the lens to the focusing tube and perform the critical focusing adjustment with the detector housing.

In performing these measurements, the

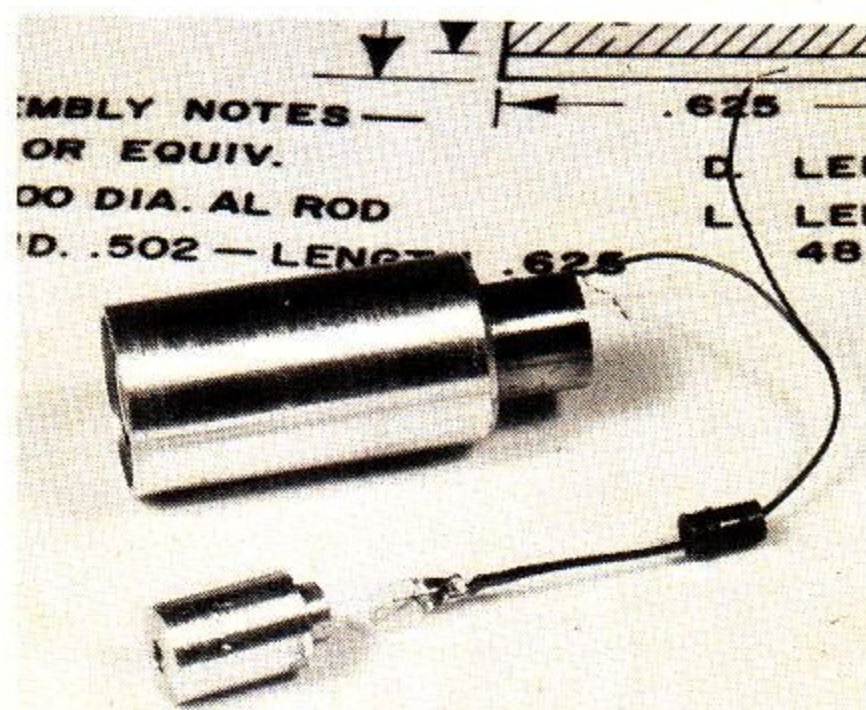


FIG. 90—UNICORN-1's infrared transmitter. Note insulating sleeve for lamp.

projector and receiver assemblies should be placed in the positions they will occupy when mounted on the robot, and be angled accordingly. If this is not done, the results of the measurements will be invalid.

Receiver circuit

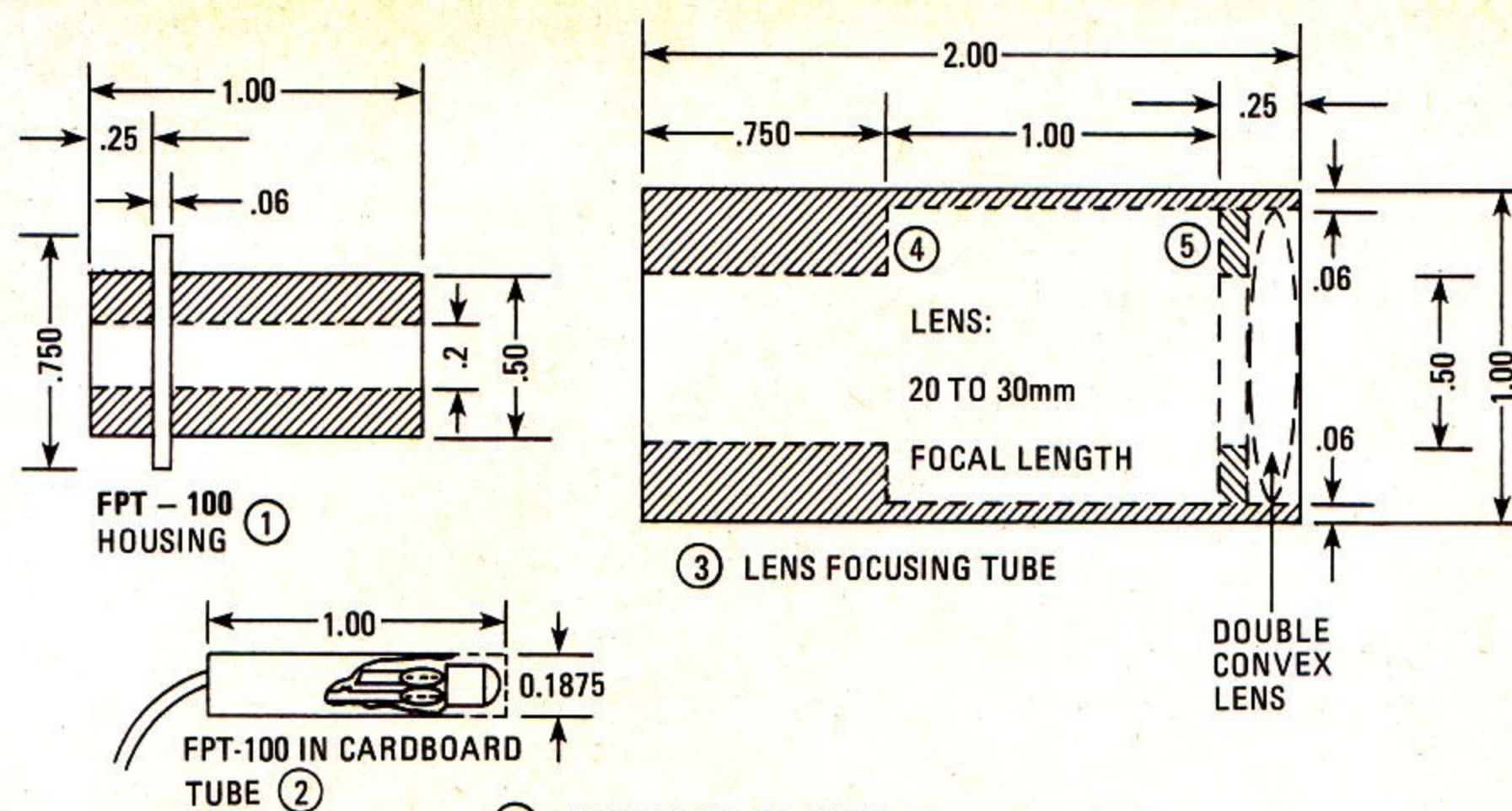
Both the transmitter and receiver can be operated from the robot's 12-volt power supply. A schematic for the receiver is shown in Fig. 92 (and the foil pattern and

parts-placement diagram in Figs. 93 and 94, respectively).

The heart of the receiver is an FPT-100 infrared phototransistor (Radio Shack part No. 276-130 is an acceptable substitute). Its collector is connected to the 12-volt supply through a 10K load resistor, R3. The collector is also connected to pin 1 of IC1 through a 22K series resistor and through a 1N5227 3.6-volt Zener diode (D1). That keeps pin 1 at a logic "high" when the detector is receiving no input.

The IC supply voltage of 5.1 volts is provided by D2, a 1N5231 Zener diode. This diode also provides the coil-voltage for RY1, a DIP relay of the same type used on the relay board described in Part 7 of this series. The circuit operates as follows:

When the infrared sensor, Q2, is at the optimum distance from a reflective obstacle, the reflected infrared light is at a maximum. The sensor is biased into a state of saturation and its collector voltage drops to zero. The 3.6-volts present at pin 1 of IC1 also drops to zero, causing the output at pin 2 to go from 3.6- to five volts (logic "high"). This biases transis-



- ① MAKE FROM .50 ALUM. ROD AND WASHER
- ② CUT FPT-100 LEADS TO .25", SOLDER 18" CONNECTING WIRES TO PC BOARD
- ③ MAKE FROM 1.00 O.D., .06 WALL, ALUM. TUBE
- ④ MAKE FROM .86 O.D. WOOD OR ALUM. ROD
- ⑤ MAKE FROM 1.00 O.D., .50 I.D., FIBER WASHER
- ⑥ ALL DIMENSIONS IN INCHES EXCEPT FOR LENS

FIG. 91—USE A CARDBOARD OR PLASTIC sleeve to prevent FPT-100 leads from shorting to metal housing.

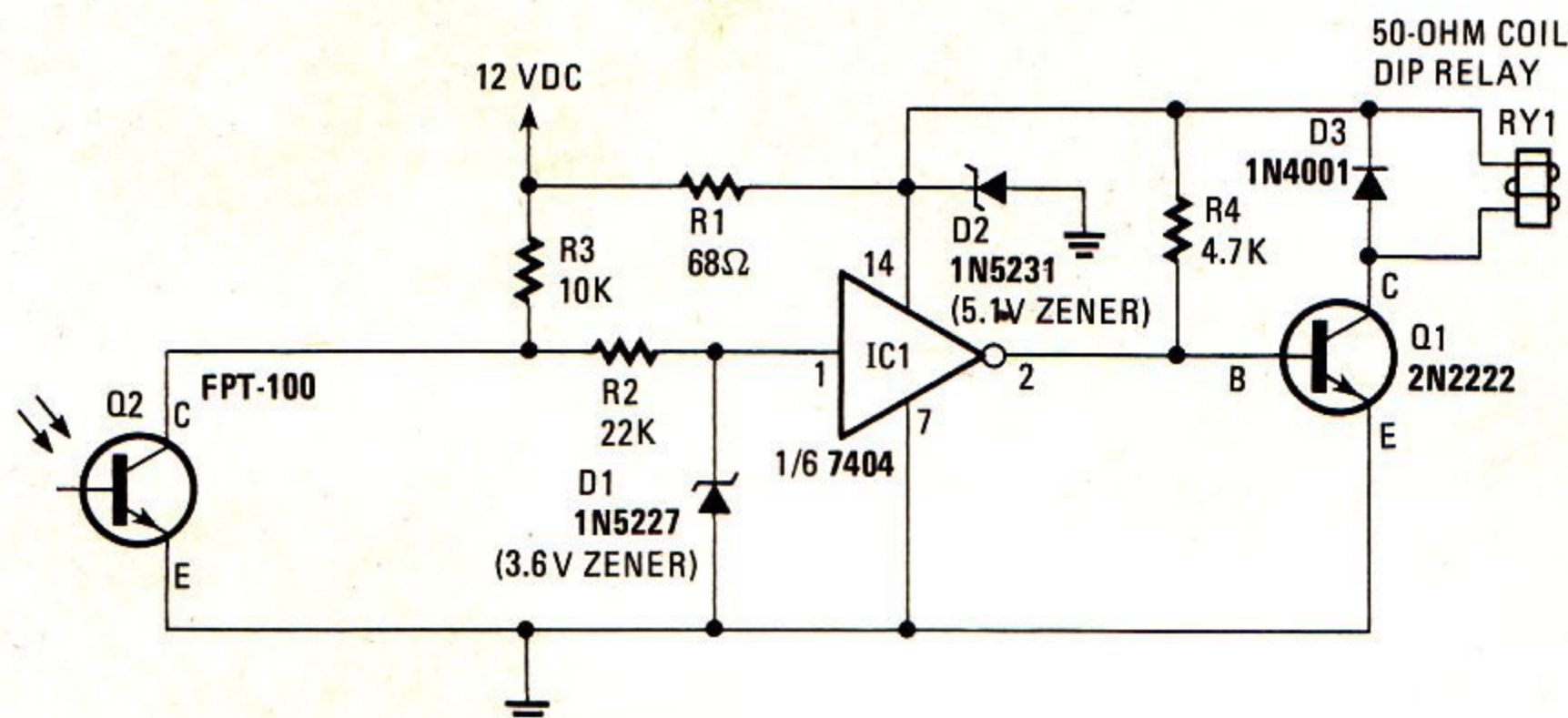


FIG. 92—INFRARED-DETECTOR circuit is simple enough to be built on perforated construction board.

tor Q1, a 2N2222, into saturation, causing current to flow through the coil of the relay and opening the relay's normally-closed contacts, thereby cutting off power to the appropriate control circuitry.

Connection to robot

Depending on how advanced your own robot is, the signals provided by the sensor circuitry can be used in several ways.

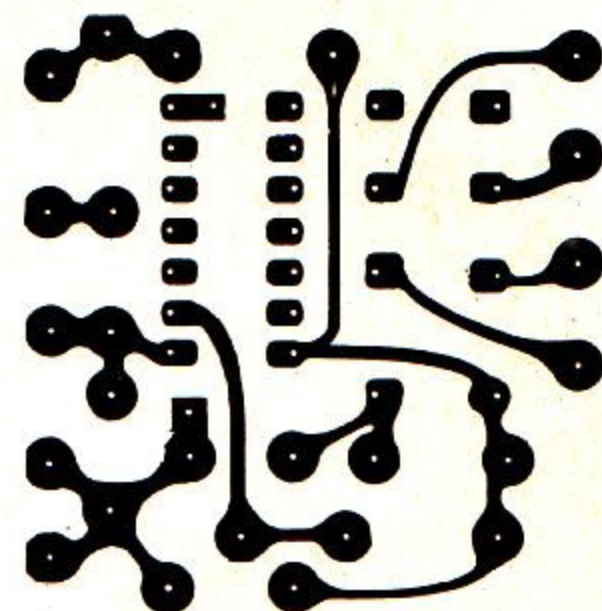
If the robot is still operating at the end of a "tether," the contact-sensor switches and the proximity-sensor relay can simply be connected in series with the motor circuits (like the limit switches) and used to cut power to the motors when an obstacle is detected. This is why you might wish to use two switches each for the front and rear contact-sensors—one switch can control the right-hand wheel, and one the left-hand one. Unused switch or relay

contacts can be used to actuate the robot's horn (or some other audible or visible signaling device) to alert you that it has run into difficulties. Without logic circuits, there's not much more that can be done at this stage.

If the robot is using radio- or computer-control, the output of the detectors can be connected to the appropriate "drop-dead" sections of the latch board (see Part 9) to achieve the same results.

Finally, if you are using a computer, a program can be written to make use of the "drop-dead" signal. For example, the computer could be programmed to respond to that signal and make the robot back up a bit, make a 45-degree turn, and check again for an obstacle. If none were present, it could continue its travel. And that just scratches the surface of the responses that could be programmed.

We've been receiving a lot of correspondence from readers who are building—or contemplating building—their own versions of Unicorn-1. We'd like to see more, along with nice sharp photographs, so we can publish a segment showing off those robots and presenting some of the innovations that you've come up with. Write to **Radio-Electronics**, 200 Park Avenue South, New York, NY 10003 and mark your envelope "ROBOT UPDATE."



1-1/2 INCHES

FIG. 93—YOU CAN ETCH detector board yourself from this pattern. Ready-made boards are also available (see parts list).

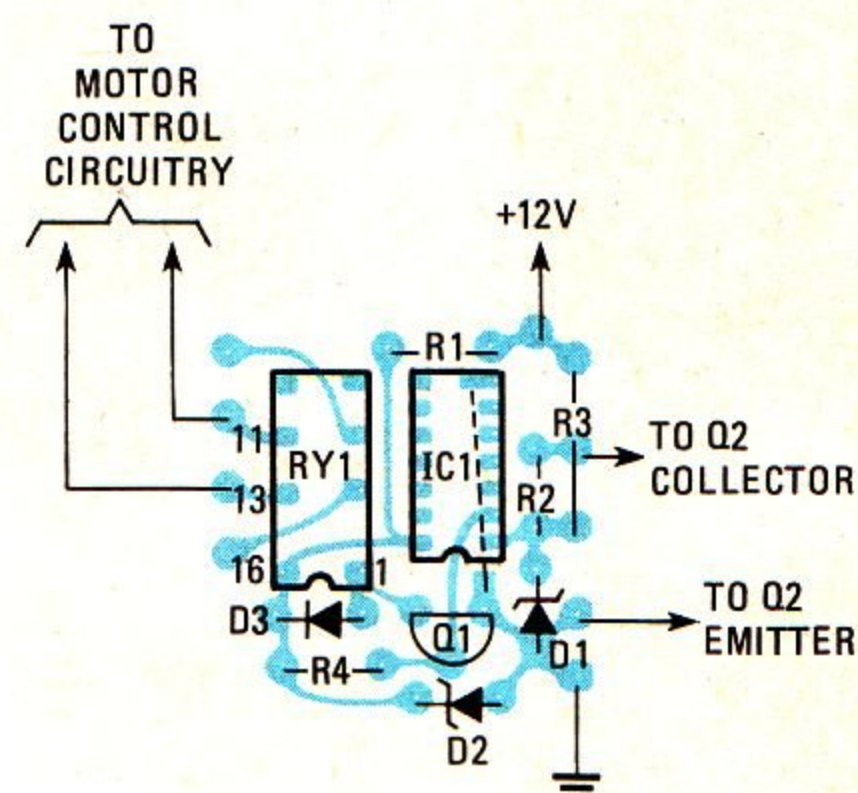


FIG. 94—DETECTOR BOARD is connected to FPT-100 sensor by 18-inch leads. See text for motor-control connections.

The rest is up to you, for this is the end of the Unicorn-1 series. We've shown you how to build a working robot, and how to enhance it with radio- and computer-control. As you continue to work with your robot you'll find its capabilities limited only by your imagination and resources.

Those of you who have built your own robots can take pleasure in knowing that you are advancing the science of robotics. In the near future, much of the hazardous and tedious work now performed by humans will be carried out by robots.

Even now we are seeing robots explore parts of the solar system that man will not visit in person for tens—or hundreds—of years. Enormous progress is being made in creating robots to serve in areas where man's help is either unnecessary or impossible to provide. What will be *your* contribution to the age of robotics? **R-E**