Computer-Controlled ROBOT ARM

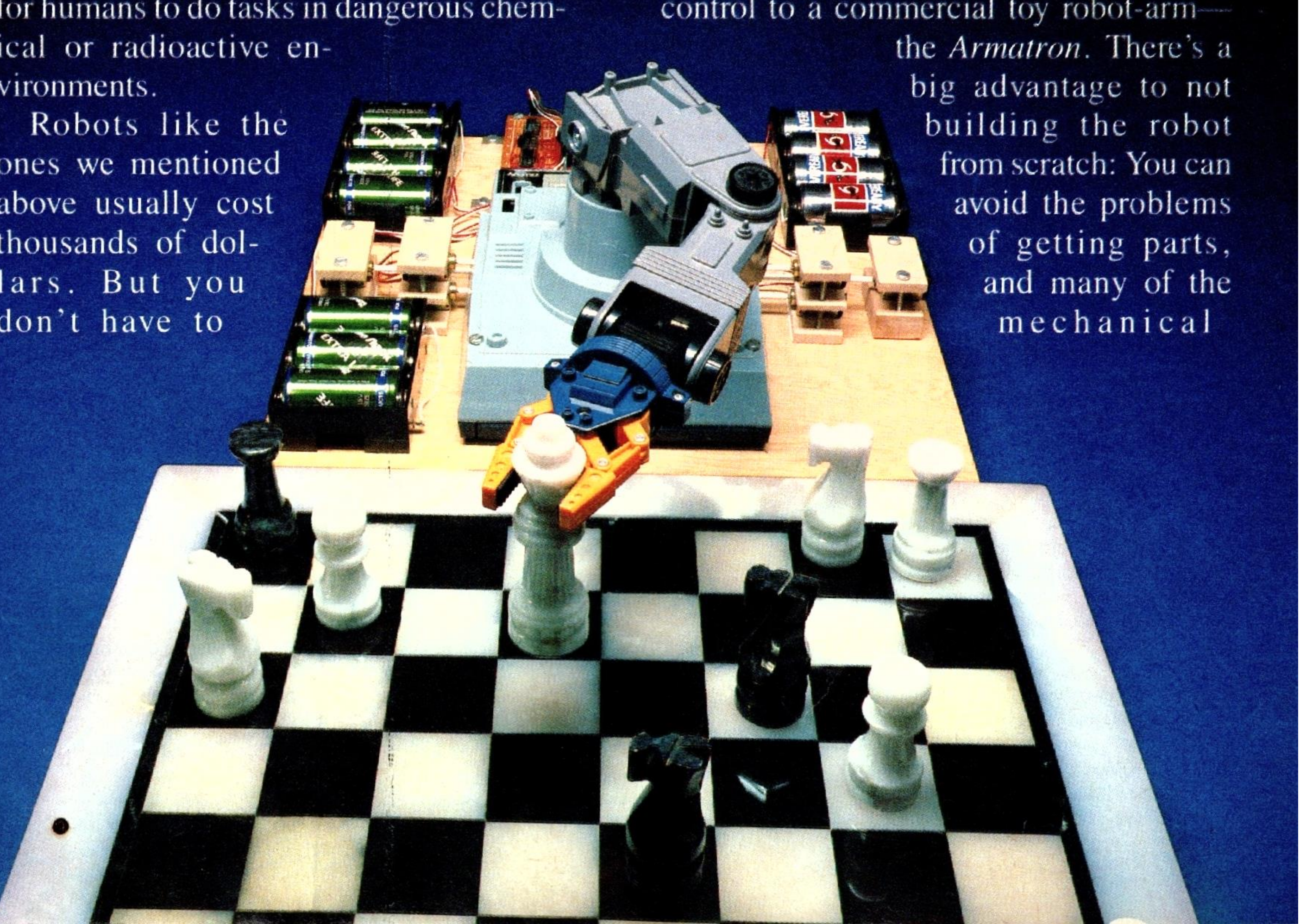
You can experiment with computer-controlled robotics without a lot of cash and without most of the mechanical headaches that usually accompany robot construction.

JIMMY BANAS

ROBOTS NO LONGER EXIST ONLY IN SCIence fiction—they're quickly becoming a part of everyday life. Robots are being used to manufacture cars, assemble electronic circuits, and even to build other robots! They are also helping to eliminate the need for humans to do tasks in dangerous chemical or radioactive environments.

Robots like the ones we mentioned above usually cost thousands of dollars. But you don't have to spend that much if you want to get started experimenting with robots. And you can avoid a lot of the headaches with mechanical construction that robots usually bring to mind.

We'll show you how to add computer control to a commercial toy robot-arm—the *Armatron*. There's a big advantage to not building the robot from scratch: You can avoid the problems of getting parts, and many of the mechanical



problems that usually accompany robot construction and experimenting. That's not to say there's no mechanical work involved—you will have to modify the *Armatron*. But the work is nothing like what you'd have to do to build a robot from scratch.

What is a robot?

Even though humans have dreamed about robots for centuries, the word "robot" is a relatively new one. It was first used in Karel Capek's 1923 play *R. U. R.* (Rossum's Universal Robots) and means "slave worker."

Early attempts at making robots were mechanical endeavors only. The results were basically dolls that looked human but ones that could perform special functions (like writing or playing music) based on a mechanical design of levers and gears.

The advent of the computer and electronic components brought robots the capability to perform much more sophisticated tasks. Some robots can "see" and others can make decisions about their actions. Even so, today's robots—although complicated machines—can perform only relatively simple tasks like pick-and-place operations. It's expected that someday—but not in the very near future—robots will perform much like those made famous in *Star Wars*. Rest assured that we won't be giving our robot that much power. But if you are interested in such endeavors, and make your robot more "intelligent," then we'd like to hear about it.

Robots used in industry are usually defined as programmed manipulators that automatically perform useful work. They can move in different ways; the *anthropomorphic-type* closely simulates the movement of a human through six degrees of freedom—with motions similar to those of the waist, shoulder, elbow, wrist, and fingers. A computer is generally used to command the robot to move to the desired positions.

The *Armatron*

For an investment of about $32, you can be the owner of a robot that is capable of the same maneuvers as industrial robots costing many thousands of dollars. Tomy Toy Company copied the design of industrial robots and built their *Armatron* robot arm. Radio Shack (Tandy) bought the exclusive rights to it. Of course, the *Armatron* robot arm doesn't have the accuracy and repeatability of industrial robots. But we don't think that it's a toy, either.

As purchased, the *Armatron* is controlled by two mechanical joysticks. By pushing the joysticks forward or back and left or right you can rotate the entire arm about its base, raise and lower the arm, bend the elbow, and raise and lower the wrist. By rotating the joysticks you can rotate the wrist and open and close the gripper.

A single motor powers all six joints of the *Armatron* through a cleverly designed planetary gearing system. The joysticks actuate levers that engage gears, giving forward, neutral, and reverse operations for each joint. Despite the clever design, we will remove and discard the joysticks, planetary gearing system, and the motor. We will replace them with six computer-controlled motors that will power the six joints.

The computer/robot system

Figure 1 shows a block diagram of the computer-controlled robot arm. Any computer that has at least a 4-bit output port can be used for control—the author used a Commodore *VIC-20*. Although the programming examples we'll discuss were written for that computer, they can be easily altered to run on other computers.

A 4-to-16 line decoder is used to decode the 4 bits from the computer port into 16 output lines. We need only 12 of the lines—there are six motors, and each needs a forward and a reverse control line. That means that you have 4 lines left over for your own experimentation. How about using those extra lines to control a movable platform?

The twelve outputs of the decoder interface with six motor-controller switches. Figure 2 shows one of those circuits, which are basically transistor switches that turn the motors off or on in either direction.

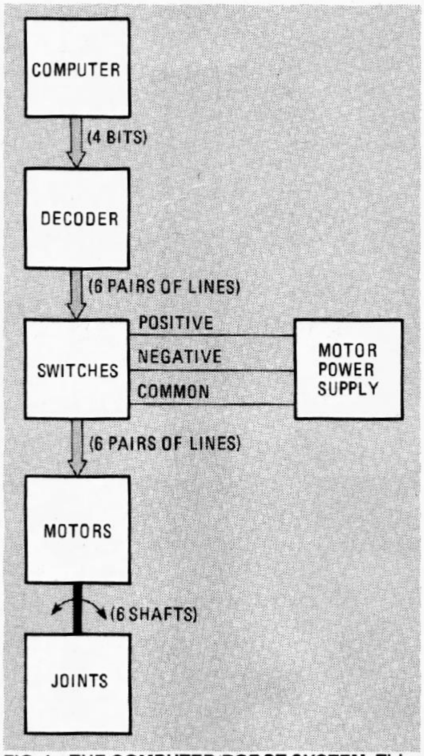


FIG. 1—THE COMPUTER/ROBOT SYSTEM. This block diagram shows us the basic components that make up our computer-controlled robot.

Small DC motors were selected to control the *Armatron* joints because they proved most practical. The motors are inexpensive, small, and don't require large amounts of power. They are also easy to work with. By POKE-ing a number from 1 to 12 to the output port we can drive any of the six joints either forward or reverse. We'll get to the details on that shortly. But now that we have a basic idea of how the system works, let's look at the mechanical interface in detail.

The mechanical interface

The hardest part of converting the *Armatron* into a computer-controlled robot is making the mechanical modifications. But as long as you're careful, you really shouldn't have any problems. The first thing we'll do is remove the existing joystick mechanism, the planetary gearing system, and the motor drive so that our motors can control the arm.

To start, remove the seven screws from the bottom of the *Armatron* and carefully separate the arm and top cover from the bottom tray. Figure 3 shows what you should see. Discard all components on the gear cover except the ring gears, which interface to the arm.

When you remove the gear cover (it's held on by five screws), you will expose the motor, the joystick parts, and the gears that interface with the ring gears. Remove the three screws that hold on the joystick cover. At this point, you can discard the joystick cover and all the joystick components. But do not throw any of the gears away. Next clip the wires at the ends of the battery holder. After you've done that, you should be left with the base and the the planetary gears as shown in Figs. 4 and 5.

The bearing blocks

After all unnecessary parts have been removed, you have to make preparations to interface control motors to the *Armatron* gears. The motors will be mounted outside of the *Armatron* case, so shafts have to be brought out from the gears to the motors. Those shafts are supported by two bearing blocks.

Each bearing block has three holes drilled through it to support three shafts. Each bearing block also has three vertical slots that hold gears that engage with *Armatron* gears.

A bearing block, along with its associated shafts and gears, is shown in Fig. 6 prior to installation in the *Armatron*. Figure 7 shows the dimensions of the bearing blocks. The author arrived at the design by trial and error; the dimensions are fairly critical and should be followed closely.

The bearing blocks can be made from a variety of materials. The author's were made from maple wood. But you can use any material that's easy to drill and cut and yet hard enough for accurate dimensioning. Wood is probably the least ex-

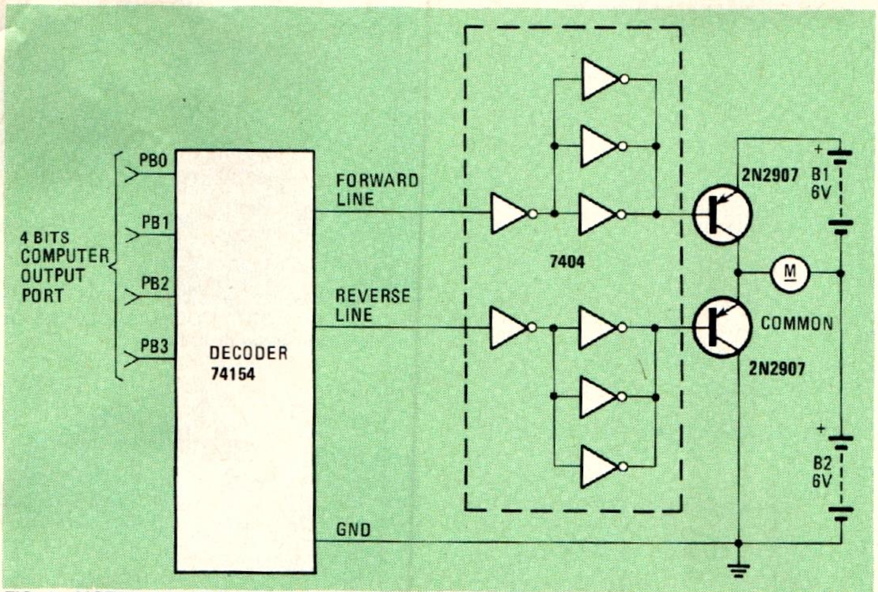


FIG. 2—MOTOR-CONTROLLER CIRCUIT. To control all of the *Armatron*'s joints, you will need six motors. Even so, you will still have 4 unused outputs from the 74154 decoder.

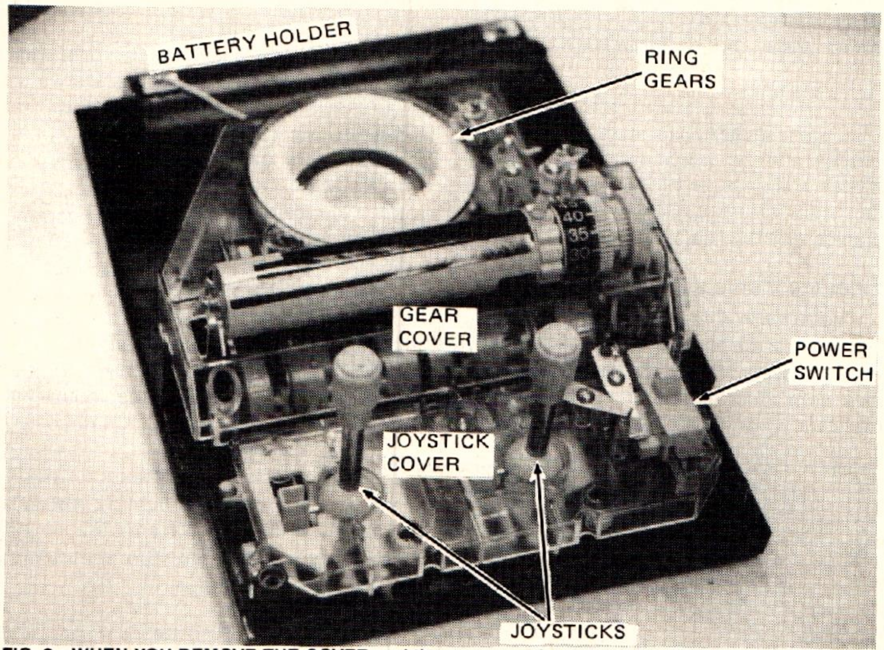


FIG. 3—WHEN YOU REMOVE THE COVER and the arm, you should be left with what is shown here.

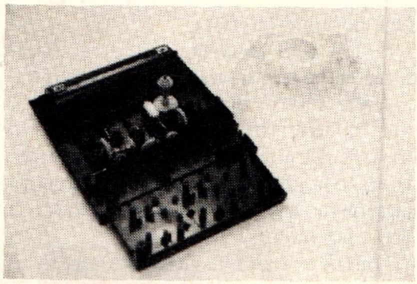


FIG. 4—THE BASE AND GEAR COVER are what are left when the mechanical joystick components are removed.

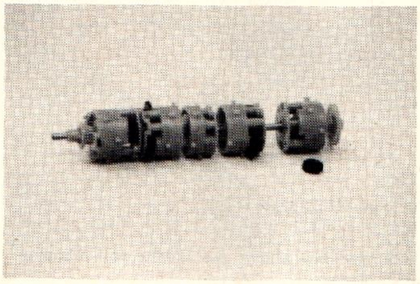


FIG. 5—THE PLANETARY GEARING SYSTEM. Don't throw this out! You'll need six ⅜-inch gears from the system.

pensive way to go. For example, the author purchased a 2 × 2 × 24-inch piece of raw stock maple for about $3 at his local lumber store. A table saw was used to cut the slots; a #29 drill bit was used to drill the holes for the shafts.

Shafts and Gears

Six shafts are needed to transfer power from the motors to the gears. Those shafts

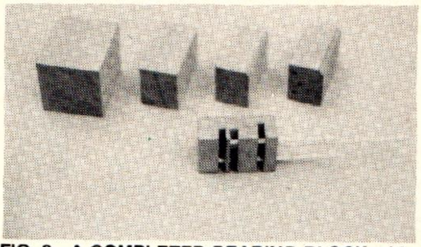


FIG. 6—A COMPLETED BEARING BLOCK with shafts and gears is shown in the foreground. The background shows the progression of fabrication steps.

must connect with the short motor shaft and must fit tightly through the gear that will interface to an arm gear. After much experimentation it was found that plastic rods, found at a local hobby store, made the best shafts. One of the key advantages of plastic rods is that they can bend slightly without breaking. That means that, even if everything is not in precise alignment, they will still work and they won't bind. The rods, with nominal outside diameter of ⅛ inch, have a hole down the middle that allows them to be easily press-fit onto the motor shafts. The lengths of the shafts, as shown in Fig. 8, are determined by the arrangement of the motors next to the *Armatron*. You may want to make yours longer or shorter. For clean cuts on the plastic shafts, use a small tubing cutter.

The gears that fit on the shafts and interface with the *Armatron* gears are were obtained from the planetary gearing system that you removed earlier (Fig. 5). The ability to use these gears is an advantage because they are already sized to mesh with the other gears. The ⅜-inch diameter gears must be removed from the planetary gearing system. You will have to modify the gears slightly by redrilling the center hole to a size just under the diameter of the shaft so that the gear can be press-fit tightly onto the shaft. For best results, use a series of drills of increasing diameter up to ⅛ inch so that you won't offset the hole from the center of the gear as you drill.

The next step is to assemble the shafts and gears in the bearing blocks. The longest shaft belongs in the middle hole and is press fit into a gear in the deepest bearing block slit; the top shaft is press fit into a gear in the middle, shallow slit and the bottom shaft is press fit into a gear in the remaining shallow slit.

Motors and mounting blocks

Each of the six joints is powered by an individual small DC motor. Small motors are necessary because the shafts exiting the *Armatron* must be spaced close together—otherwise they will not be able to fit the bearing-block gears.

To hold the motors, we need five mounting blocks on each side of the *Armatron*. Figure 9 shows the dimensions for the mounting blocks. Use those di-

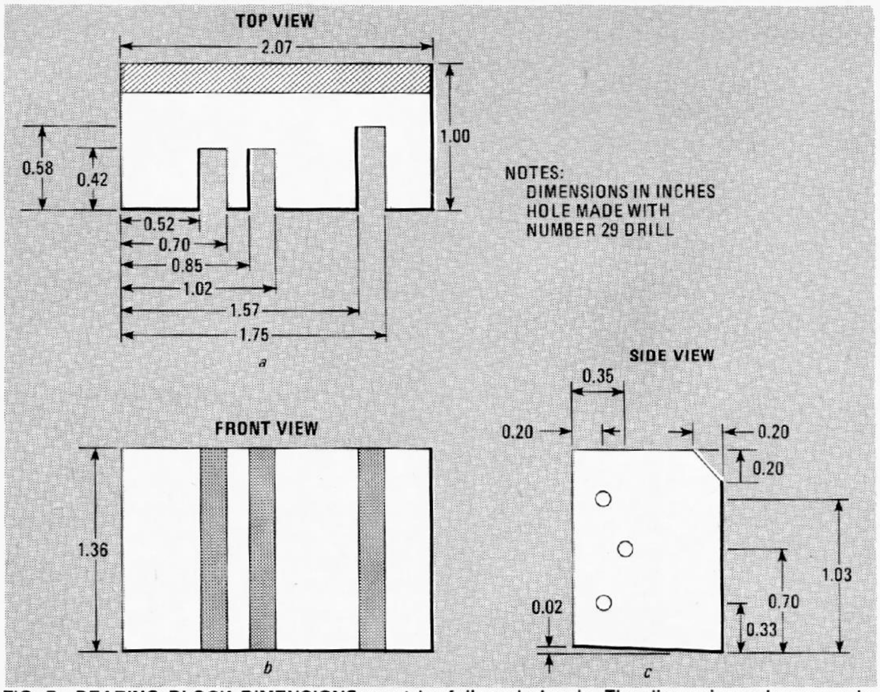


FIG. 7—BEARING BLOCK DIMENSIONS must be followed closely. The dimensions shown are in inches. All holes are made with a #29 drill.

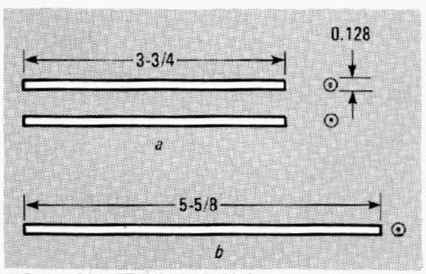


FIG. 8—SHAFT DIMENSIONS. Two sets of three shafts are required.

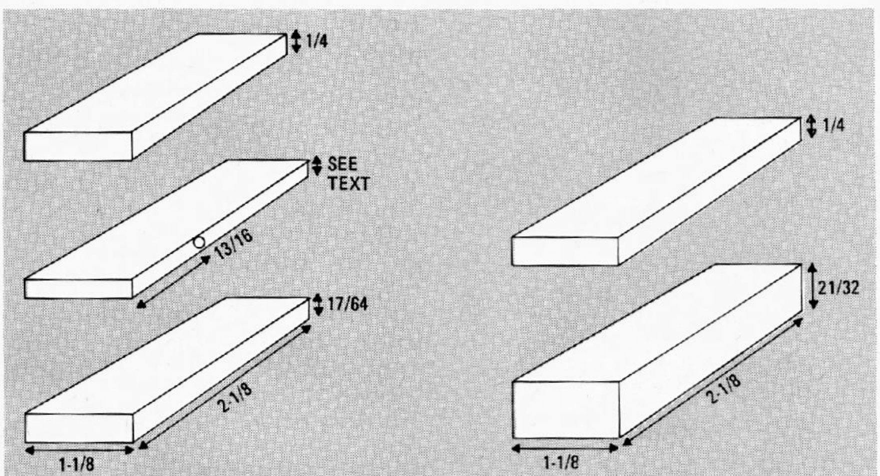


FIG. 9—MOTOR-MOUNTING BLOCK DIMENSIONS. The dimensions shown are for the author's system. You may have to change the dimensions to fit your motors.

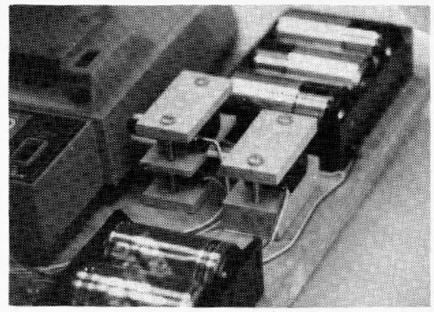


FIG. 10—THE MOTOR-MOUNTING BLOCKS on the right side of the *Armatron*.

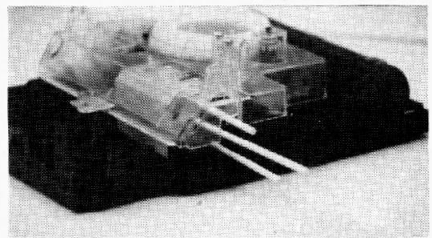


FIG. 11—YOU WILL HAVE TO MODIFY the base and gear cover to accommodate the shafts.

mensions only as a guide—you may have to change your dimensions to fit the motors that you use. The critical block is the spacer between the two motors closest to the *Armatron* (see Fig. 10). That spacer block is made by first drilling a hole using a #29 drill bit through a piece of wood. (The same material used in the bearing blocks can be used in the mounting blocks.) The spacer block is sanded until the width is just slightly thicker than the diameter of the shaft.

Final mechanical modifications

The *Armatron* must be modified to accommodate the bearing blocks and shafts. Figure 11 shows the right side of the *Armatron* with modifications to the base and ring gear cover/holder. A slot must also be cut in each side of the outer cover so that the shafts can protrude. A thin hacksaw blade and file can be used to cut away the plastic to make the modifications.

Final assembly of the mechanical components starts with mounting the bearing blocks in the base. Place each bearing block in the base and match drill two holes; use ½-inch self-tapping screws to secure the bearing blocks to the base. Replace the gear cover with the original screws. Assemble the outer cover and the arm to the base with the original screws. Attach the *Armatron* to a ½ × 12½ × 12½-inch plywood base with four 2¼-inch screws located in the four corners of the *Armatron*. Mount the two motors on each side closest to the *Armatron*. The motors are press fit into the plastic shafts. Finally, mount the third motor on each side of the *Armatron*. To hold down the mounting blocks, use 2¼-inch screws. Do not overtighten the mounting blocks; when operated, final adjustments to the motor positions can be done to achieve proper alignment. You may want to use paper spacers under the bottom mounting blocks so that the motors turn freely.

The electronic interface

Now that the mechanical modifications are complete, we can turn to the easy part: the electronic interface.

Figure 12 shows the complete schematic of the circuit that's required to interface the motors to a computer. As we mentioned earlier, we need 12 control lines to drive the six motors. In other words, two lines—one for forward control and one for reverse—are required for each of the six motors.

Your computer must have at least a 4-bit output port to operate the interface. Those four lines are fed to IC5, a 74154 4-to-16 line decoder. That decoder takes an input and selects one of sixteen lines as an output. (That output depends on the value of the 4-bit word. (If you are familiar with binary numbers, you'll realize that there are 16 possible combinations of 4 bits. The selected output line is driven low,

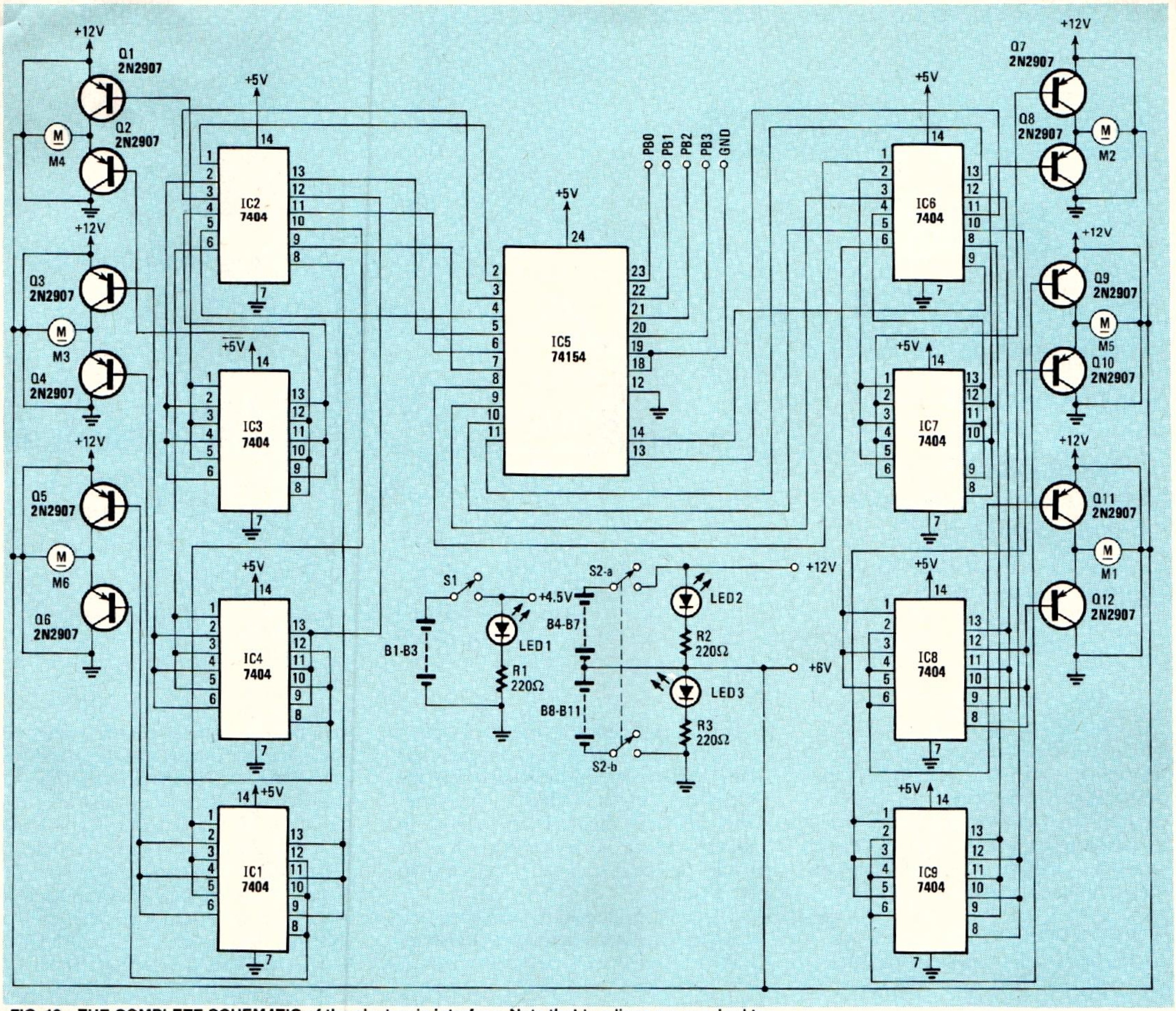


FIG. 12—THE COMPLETE SCHEMATIC of the electronic interface. Note that two lines are required to control each motor. One will cause the motor to go forward, the other will cause it to go in reverse. Motors M1–M3 correspond to the bottom, top, & center of the right side; M4–M6 correspond to the top, bottom, & center of the left side.

while the other 15 lines remain at a high logic level.

Each output line is first connected to a single 7404 inverter and then to three 7404 inverters arranged in parallel. That not only boosts the output current to sufficient levels, it also returns the control signal back to its original logic level. The inverter outputs are fed to the bases of 2N2907 transistors, which act as simple switches and determine the direction of the flow of current through the motor.

For example, if the forward line is low and reverse line is high, then the forward transistor conducts turning the motor in the forward direction. If both inputs are high then neither transistor conducts and the motor remains off. Note that if somehow both lines go low the entire loop is connected putting 12 volts across the two transistors. That will quickly burn them out. Fortunately, using a decoder like the 74154 should prevent that from ever happening.

Two separate switches are used to turn the power supplies on and off. Three LED's are used to show that the power is on. During system operation the switch for the electronic circuit should be turned on before the switch for the motor controllers to insure that the transistors are properly biased. Similarly the switch for the electronic circuit supply should be turned off last.

Circuit layout and final assembly

Circuit layout is not critical and you should feel free to use the construction technique you're most comfortable with. The author's circuit, shown in Fig. 13, was built using a pre-drilled "universal PC board."

A 12/24 pin card-edge connector was used to interface the electronic circuit with the VIC-20 computer. Of course, the connector you use will depend on the computer you use.

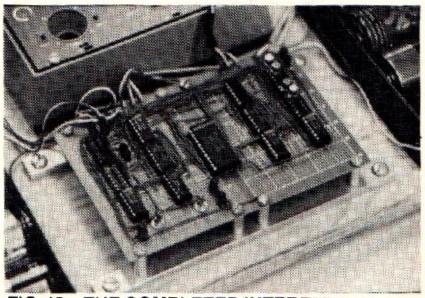


FIG. 13—THE COMPLETED INTERFACE. Universal PC boards were used. Note how the transistors are mounted in IC sockets.

Three separate battery packs, using D-cells, are used for power supplies. Two 6-volt supplies are used for powering the motors; a 4.5-volt supply is used for the integrated circuits.

continued on page 117

ROBOT ARM

continued from page 53

To finish assembly of the system, mount the electronic interface in front of the *Armatron* on the base. Connect the motor leads to the interface. Finally, mount the battery packs to the base.

Software for control

The demonstration program, shown in Table 1, was written to direct the computer to move the robot arm automatically—first rotating the entire arm (joint 1), raising the shoulder (joint 2), bending the elbow (joint 3), bending the wrist (joint 4), rotating the wrist (joint 5), and closing the gripper (joint 6). After the robot does those steps, it will do them again—but in reverse. The demonstration program is written in BASIC. Pressing "D" from the keyboard will exercise the joints in sequence. Pressing "J" will result in the computer asking for a joint number from one to six. Pressing "F" or "R" will cause the joint to move forward or reverse respectively. Finally, pressing 'S' stops the program.

That program, as well as the other to follow, uses the user port of the VIC-20 at memory location 37136. Since we are not inputting information, we must POKE the number 255 into memory location 37138 (POKE 37138,255). Each joint must be turned on for a different length of time in order to perform the demonstration. Those on-times are stored in arrays DF and DR and can be changed if you desire. Because the order in which the motor controllers were wired to the decoder was different than the order of the joints, a software transformation had to be made as indicated in data statement 3010.

The second program, shown in Table 3, controls the joints through the keyboard. Key "J" will ask for a joint number. Pressing keys "F" or "R" will move a joint forward or reverse as long as the key is being pressed. Pressing "S" will end the program.

PARTS LIST

Resistors
R1–R3—220 ohms, ¼ watt, 10%

Semiconductors
IC1–IC4, IC6–IC9—7404 hex inverter
IC5—74154 4-to16 line decoder.
LED1–LED3—Standard red LED
Q1–Q12—2N2907

Other components and materials
S1—SPST switch
S2—SPDT switch
M1–M6—DC motors (6 volts)
B1–B11—1.5 volts, "D" cells
Armatron robot

Miscellaneous: Battery holders, motor shafts, IC sockets, breadboard, edge connector for your computer port, etc.

TABLE 1—AUTOMATIC DEMONSTRATION

```
0 REM * AUTOMATIC
  DEMONSTRATION OF ARM JOINT
  MOTIONS *
1 N=10
2 DIM F(6),R(6),L(6)
3 FOR J=1 TO 6:READ F(J):NEXT J
4 FOR J=1 TO 6: READ R(J):NEXT J
5 POKE 37138, 255:POKE 37136,0
6 FOR J=1 TO 6:READ L(J):NEXT J
9 INPUT A$:IF A$="D" THEN 100
10 PRINT"INPUT JOINT NUMBER"
11 INPUT J
20 GET A$:IF A$=""THEN 20
21 IF A$="D"THEN 100
25 IF A$="F"THEN
   KF=L(J)*2-1:GOSUB 1000:
   GOTO 20
30 IF A$="R"THEN
   KR=L(J)*2:GOSUB 2000:GOTO 20
35 IF A$="J"THEN 10
40 IF A$="S"THEN END
45 GOTO 20
100 FOR I=1 TO 2
101 FOR J=1 TO 6
105 IF I=2 THEN 120
110 KF=2*L(J)-1:GOSUB 1000
115 GOTO 130
120 KR=2*L(J):GOSUB 2000
130 NEXT J
135 NEXT I
140 GOTO 10
1000 FOR T=1 TO N
1005 POKE 37136,KF
1010 FOR Z=1 TO F(J):NEXT Z
1015 POKE 37136,0
1020 FOR Z=1 TO 1:NEXT Z
1025 NEXT T
1030 RETURN
2000 FOR T=1 TO N
2005 POKE 37136,KR
2010 FOR Z=1 TO R(J):NEXT Z
2015 POKE 37136,0
2020 FOR Z=1 TO 1:NEXT Z
2025 NEXT T
2030 RETURN
3000 DATA 12, 12, 15, 15, 75, 75
3005 DATA 12, 12, 15, 15, 75, 75
3010 DATA 4, 5, 2, 1, 6, 3
```

The programs we've showed you so far have moved one joint at a time. However, it is possible to move more than one joint simultaneously. That's of interest to us because we, as humans, move our joints in what is known as *coordinated motion*. You can experiment with different combinations of joints to see what kind of overall motion can be achieved.

The controller circuit we showed you before would be of no use because only one line of the decoder can be active at a time. However, if your computer has an output port with 12 output lines, you can control the motors directly without the decoder. Remember, however, that if both the forward and reverse control lines for a motor go low at the same time, the transistors will be burned out. The circuit shown in Fig. 14 is a simple interface between the computer and motor controller that prevents that from happening.

TABLE 3—KEYBOARD CONTROL

```
0 REM * KEYBOARD CONTROL OF
  ARM JOINT MOTIONS *
1 POKE 37138,255:POKE 37136,0
2 DIM F(6),R(6)
3 FOR J=1 TO 6:READ F(J):NEXT J
4 FOR J=1 TO 6:READ R(J):NEXT J
5 PRINT"INPUT JOINT NUMBER"
10 INPUT J
11 IF J=4 THEN L=1
12 IF J=3 THEN L=2
13 IF J=6 THEN L=3
14 IF J=1 THEN L=4
15 IF J=2 THEN L=5
16 IF J=5 THEN L=6
17 KF=2*L-1:KR=2*L
20 GET A$:IF A$=""THEN 20
25 IF A$="F"THEN GOSUB 100
26 IF A$="R"THEN GOSUB 200
30 IF A$="S"THEN END
35 IF A$="J"THEN 5
40 GOTO 20
100 IF PEEK(197)<>42 THEN 111
106 POKE 37136,KF
107 FOR Z=1 TO F(J):NEXT Z
108 POKE 37136,0
109 FOR Z=1 TO 1:NEXT Z
110 GOTO 100
111 POKE 37136,0
115 RETURN
200 IF PEEK(197)<>10 THEN 211
206 POKE 37136,KR
207 FOR Z=1 TO R(J):NEXT Z
208 POKE 37136,0
209 FOR Z=1 TO 1:NEXT Z
210 GOTO 200
211 POKE 37136,0
215 RETURN
300 DATA 7, 7, 15, 15, 75, 75
310 DATA 12, 12, 12, 12, 75, 75
```

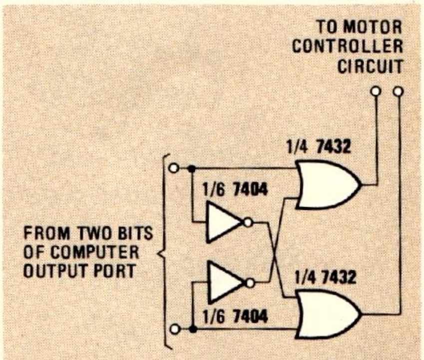


FIG. 14—IF YOUR OUTPUT PORT has enough lines to control the motors, then you can bypass the decoder. This circuit will make sure that you don't burn out your transistors. It should be wired between the output port and the inverters that drive the transistor switches.

Using your robot

When putting your robot to work, keep in mind the following rules that were proposed by Isaac Asimov, in 1950. 1) A robot may not harm a human being or allow a human being to be harmed. 2) A robot must obey the orders given it, unless the orders violate the first rule. 3) A robot must protect itself as long as it does not violate the first and second rules. Keep those rules in mind and have fun with your robot! **R-E**

OOOOOPS!

In our article "Computer-Controlled Robot Arm" in the May 1985 issue, we let a rather embarrising error get into print. Notice that the power and ground lines are shorted together at each motor. Also note that the inputs and outputs of IC3 are reversed. Shown here is a section of the circuit as it should be. Figure 2 in the May issue also shows the correct configuration.

We're sorry those errors got past us, and we hope that they did not cause too many problems.—*Editor*

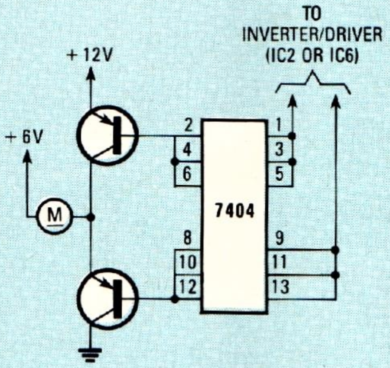


FIG. 1