NOV/DEC 1982 VOL. 4 NO. 6 $3.00 USA/$3.50 CANADA

THE JOURNAL OF INTELLIGENT MACHINES

ROBOTICS AGE™

ARMATRON: A STUDY IN ARM ENGINEERING

Mark Robillard

3 Peach Lane, Townsend, Massachusetts 01469

Anyone who has read my articles over the course of this year has probably noticed that, when it comes to supplying a starting point for robotics, I tend to support the local toy merchant. Toys, although intended for children, are designed by adults. More often than not the designer must surmount many engineering difficulties to accomplish the necessary price/performance ratio. Since this is the beginning of the holiday season and toy advertisements are becoming the most popular show on TV, I find it only fitting that we investigate a new robotics toy.

Photo 1. Complete Armatron from Tomy Toys. Also included with the arm are various shapes and forms used for practice manipulations. Two D cell batteries (not included) provide the motive power.

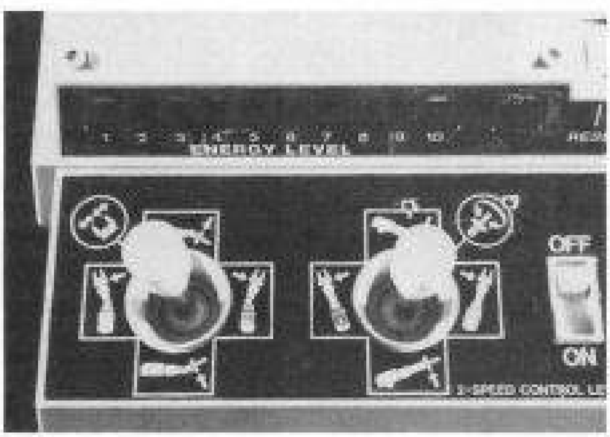


Photo 2. Armatron joystick control platform. These shafts move gears in and out of mesh within the base.

The toy is called Armatron (see photo 1). Tomy Toys of California has imported a highly functional miniature robot manipulator from Japan. Armatron has capabilities usually found only in much more expensive experimental arms. You maneuver Armatron using two joysticks. These controls allow the whole arm to rotate left or right, to raise and lower, to bend at the elbow left and right, and to raise or lower the hand assembly. Turning the joystick shafts rotates the wrist and opens and closes the hand. Not bad for $48.47 at a local discount toy store!

That price puts it at about one-tenth the cost of the closest experimental arm. This does not mean you get one-tenth the capabilities. I have lifted eight ounces straight up with the arm. Its reach is 12 inches, fully extended, yet it can fold back on itself and pick up anything close to its base. It is addictive after you first flip the power switch. If you buy one for yourself, either get another for the kids or don't let them see it.

Mechanics. Photo 2 is a view of the joystick control assembly. These shafts position gears within the unit and have no relationship to the type of control used in computer games. Photo 3 shows how the joystick interacts with the gear mechanism. The large gear in the background is actually a series of independent gears seated into each other. This is called a planetary gear system because the gears rotate like the planets. Each joint has a separate gear.

Photos 3 through 7 show each joint. Photo 6 also provides a complete view of the hand, which is equipped with rubber anti-slip grippers. All these joints are controlled by one rather small motor which is located in the base and powered by two D cell batteries.

Figure 1 depicts the gearing action used to raise and lower the shoulder. The gear ratio increases torque. At the elbow are four shafts with gears. Figure 2 illustrates the interaction and purpose of each gear. Figure 3 explains the gear mechanism of the wrist/hand assembly. The two-pincer hand design uses a unique spring-loaded coupling between its drive gears and the finger mechanisms as shown in figure 4.

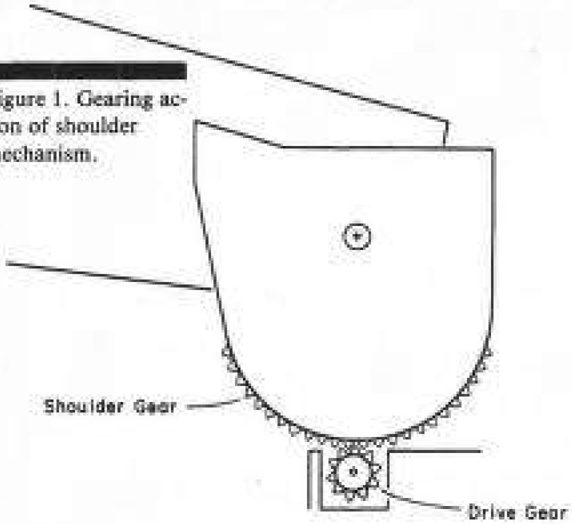


Figure 1. Gearing action of shoulder mechanism.

Automatic Control. Joysticks are fun, but I soon started to think about the possibility of controlling Armatron with a computer. To avoid the complication of cutting up the device's internal mechanisms, I chose to emulate a hand controlling the joysticks rather than replacing them. This approach is not without its problems. What must be accomplished is linear motion, forward and back, left and right; don't forget the turning action to rotate the wrist and operate the hand.

There are a number of possible approaches to the problem, but some are severely limited because of the lack of space in the joystick control area. The first mechanism I built converts the rotary motion of a motor into linear motion. Figure 5 depicts the design. My design imitates the inside of a conventional resistor joystick.

The typical resistor joystick contains two mechanical guideways called bails. These bails guide the stick left or right, depending on which direction the motor is turning. The slot in the bail allows the stick to travel forward and backward as a result of the other motor's action. Two motors provide four directions. I should mention that turning off all power to the motor allows the built-in spring to return the Armatron joystick to the center, or off, position.

Photo 8 shows a wooden frame I constructed to hold the motors and the bails in place. I used heavy wire, such as the kind used to hang drop ceilings, to form the bails. The motor must produce a fair amount of torque. I used Mabuchi RE-56 slot car motors, which are available in most electronics or surplus stores. They mount from the front and are only 1 1/4 inches in diameter.

Motor Interface. Figure 6 is a schematic of a typical motor control circuit which can be used for each motor. The circuit is compatible with standard TTL levels. Refer to the article "Inexpensive Arm-Hand System" (*Robotics Age*, May/June 1982, page 20) for an explanation of motor circuit operation. This

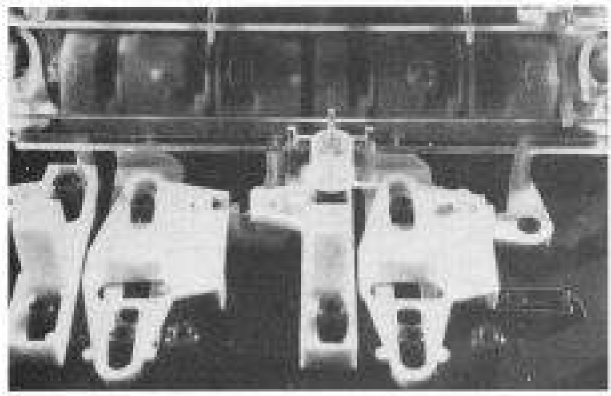


Photo 3. Joystick action of gear meshing. The large cylinder contains a multitude of gears. Moving the joystick causes a small pointer to move on the gear cylinder, allowing certain gears to mesh.

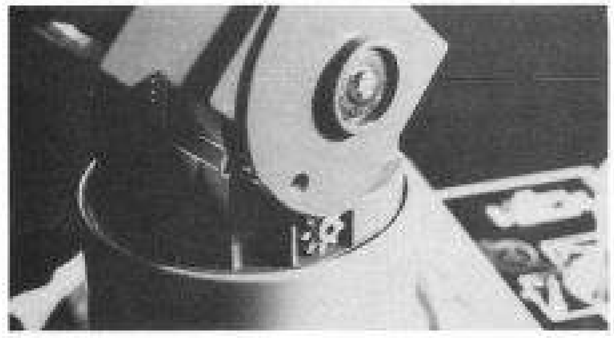


Photo 4. Close-up view of shoulder joint with decorative cover removed.

Photo 5. This view of the elbow joint shows the four gears used to provide motion from elbow to hand. Figure 7 explains each gear.

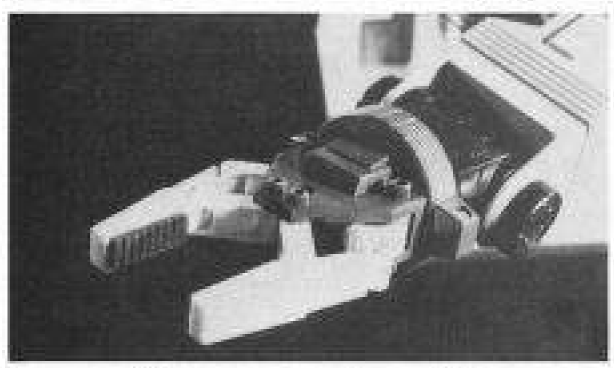


Photo 6. Close-up view of hand/wrist assembly. The hand is capable of opening up to two inches while keeping the jaws parallel at all times.

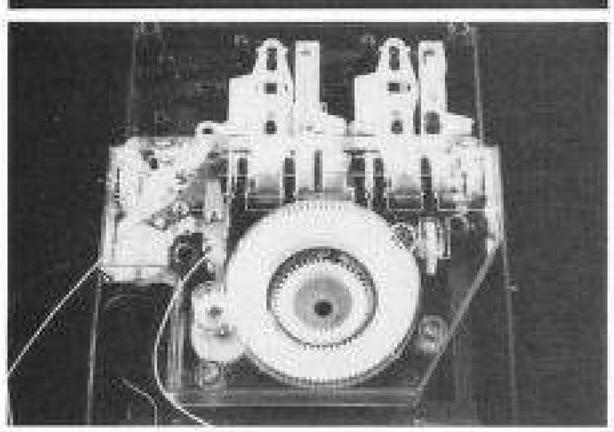


Photo 7. Inside view of the main gear box located in the base. That little motor moves everything. The planetary gear arrangement is visible in center.

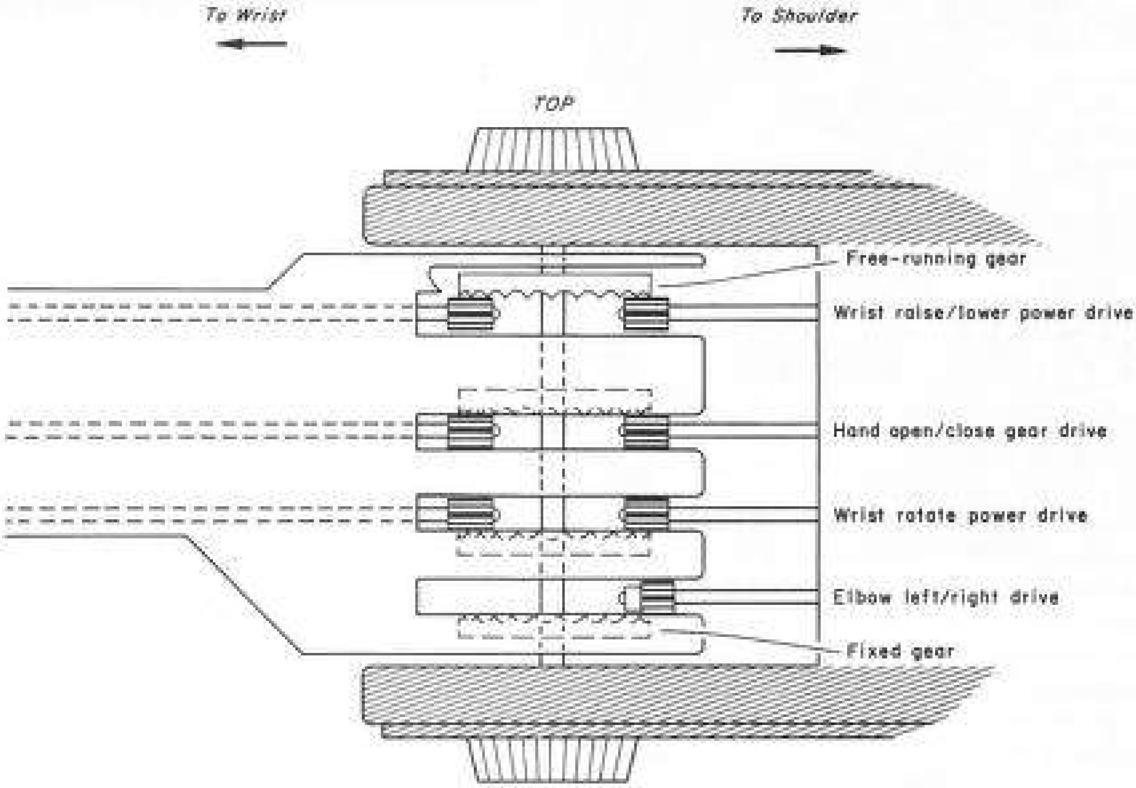


Figure 2. Elbow gear action. The bottom gear moves the elbow left and right. The top gear transfers the wrist raise and lower power. The next to the bottom gear rotates the wrist. The remaining gear transfers open and close power to the hand.

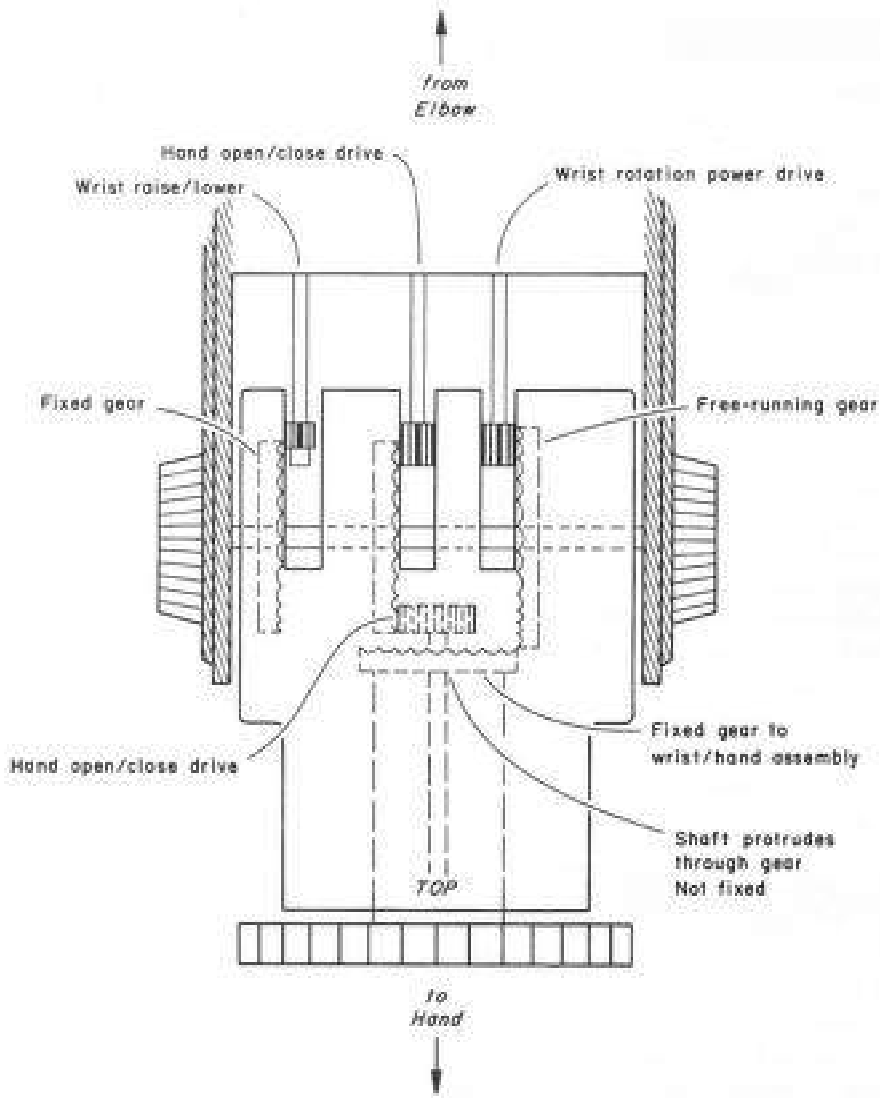


Figure 3. Wrist/hand mechanical action. The center gear opens and closes the hand. The left gear raises and lowers the wrist. The remaining gear rotates the wrist.

circuit is somewhat different, however, since I have used a specific motor control IC(TPQ6002). This IC has both NPN and PNP transistors in it so that only one package is required per motor. It is very inexpensive and is readily available.

Solenoid Control. Another control possibility is to use solenoids. These devices perform linear motion but can only provide one direction of movement. It is necessary, therefore, to use eight solenoids. Because of space limitations on the joystick housing, they must be mounted on a separate platform.

Figure 7 depicts a typical way of mounting the solenoids. An elastic control cable is connected to each of the solenoid plungers, then to the joystick shaft. A means of stopping the shafts from pulling out when they are not energized is also needed.

Solenoid control is by far the more exact method, but it requires more mechanical pieces. The interface to the solenoids is extremely simple, as shown in figure 8. Using high-current peripheral driver ICs simplifies the process quite a bit.

Rotary motion for hand and wrist movements is the toughest part of the emulation. Although the joysticks turn only a few degrees in either direction, they do require quite a strong force to move. I suggest you cement a gear on the top of each stick, then attach a motor with a smaller gear so that it meshes with the first. This increases the torque. Mount the motor so that it travels with the stick.

After all this mechanical effort, you will see that the only disadvantage of Armatron from an external control standpoint is that it is all mechanical inside. I wish that it was programmable like the Milton Bradley Big-Trak.

Computer Control. The computer can control whichever mechanical interface you choose. A parallel port can connect the motor or solenoid drivers and the computer. Each code controls one direction device. All you need is a teach-and-learn program.

I have chosen the following keys to control Armatron:

Key	Function
1	Raise entire arm
2	Lower entire arm
3	Turn entire arm right
4	Turn entire arm left
5	Turn elbow right
6	Turn elbow left
7	Raise wrist
8	Lower wrist
9	Rotate wrist right
0	Rotate wrist left
O	Open hand
C	Close hand

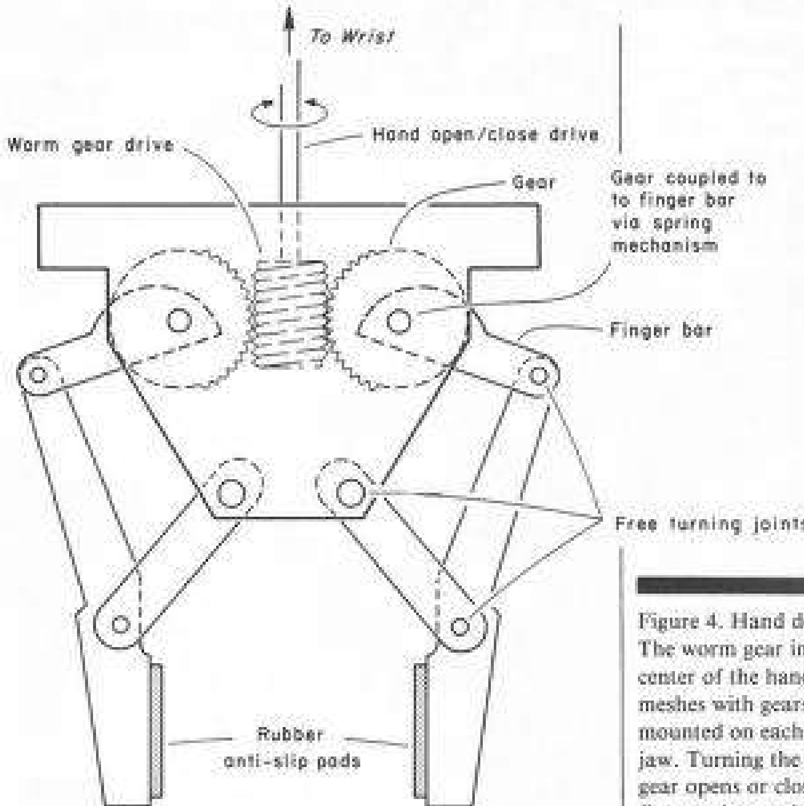


Figure 4. Hand design. The worm gear in the center of the hand meshes with gears mounted on each finger jaw. Turning the worm gear opens or closes the fingers via a spring coupling.

There are four command keys associated with the program:

Key	Command
M	Set manual mode — no memory
A	Set automatic mode — memorize each keypress
B	Begin recall from memory
S	Stop recall from memory

Note: The program listing shown on page 45 is a more advanced version that will prompt the user for commands.

After stopping Armatron in the middle of a memory recall sequence, push B again to reset and start the sequence from the beginning. Be sure, before you start a memory recall sequence, that Ar-

Text continued on page 46.

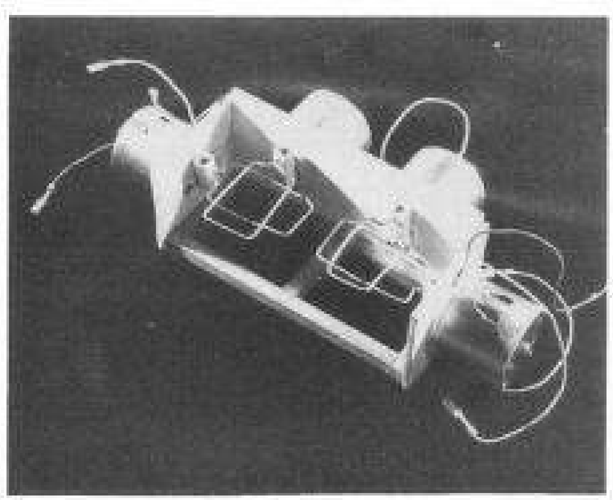


Photo 8. Using motors to produce linear action. This assembly is made up of four RE-56 motors with attached wire bails.

Figure 5. Details of bail construction and motor placement. Motorized bails provide linear motion.

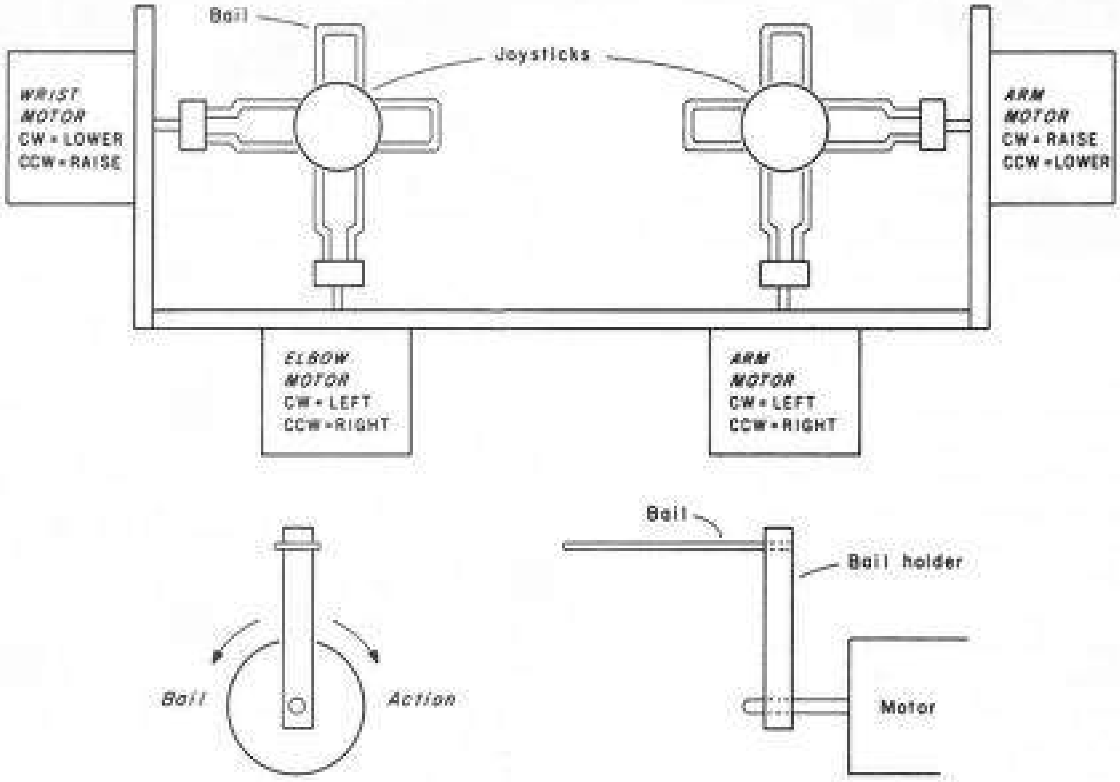
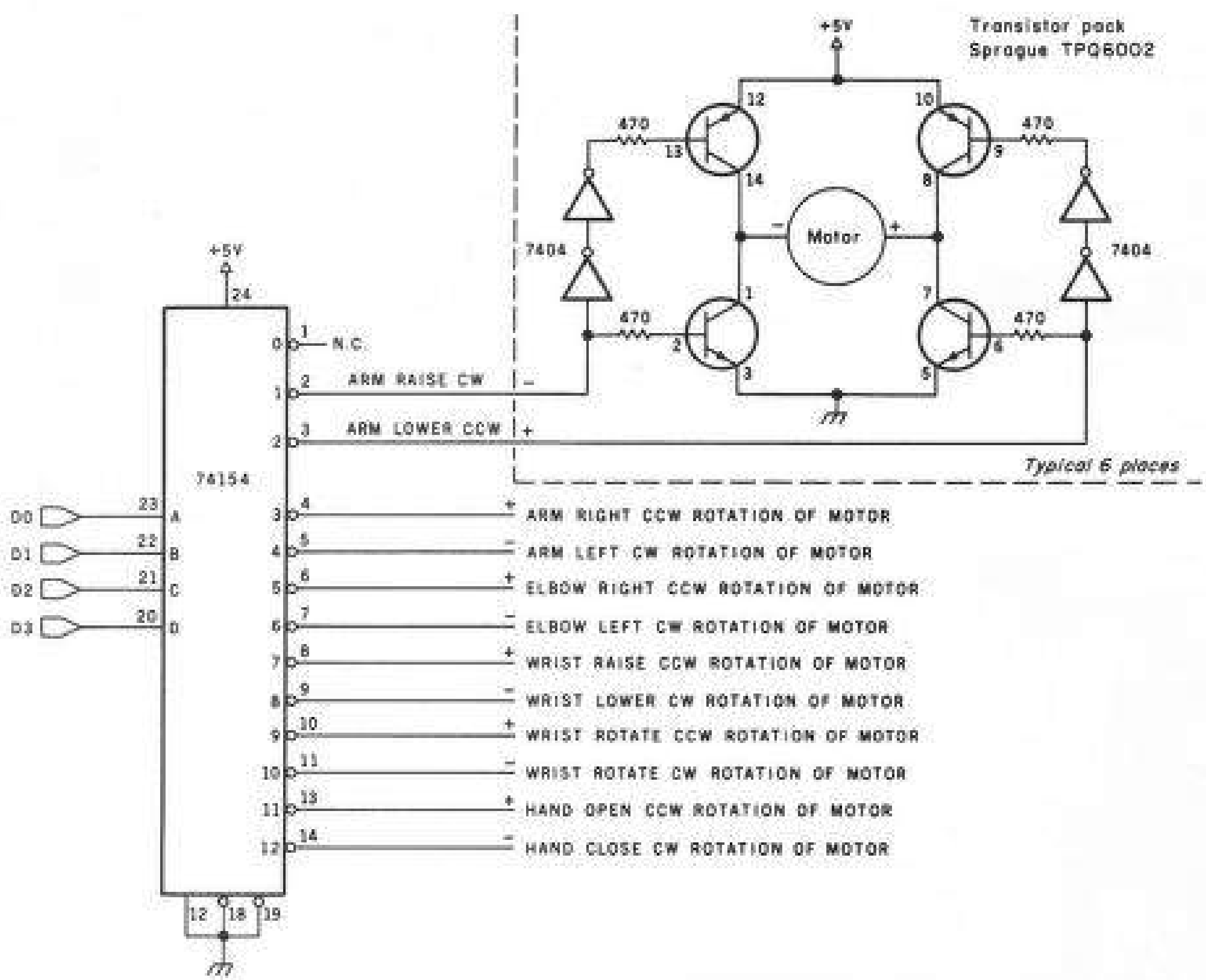


Figure 6. Schematic of motor driver system. Commands in the form of a four-bit number energize one direction at a time.



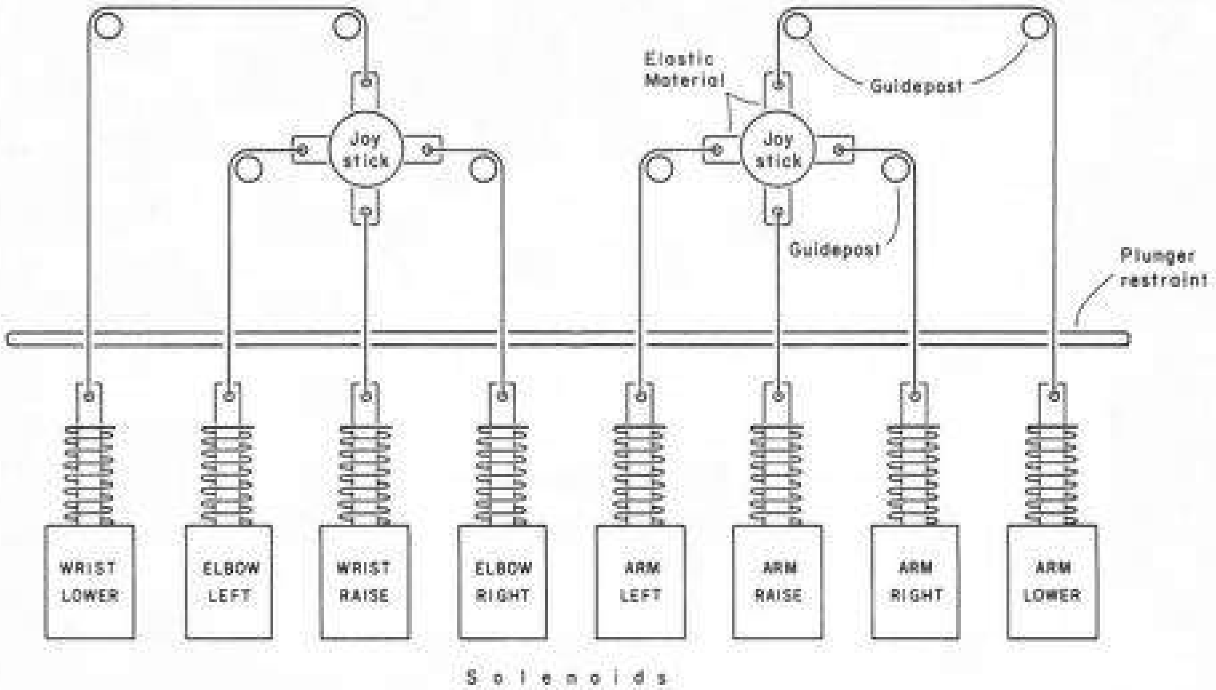
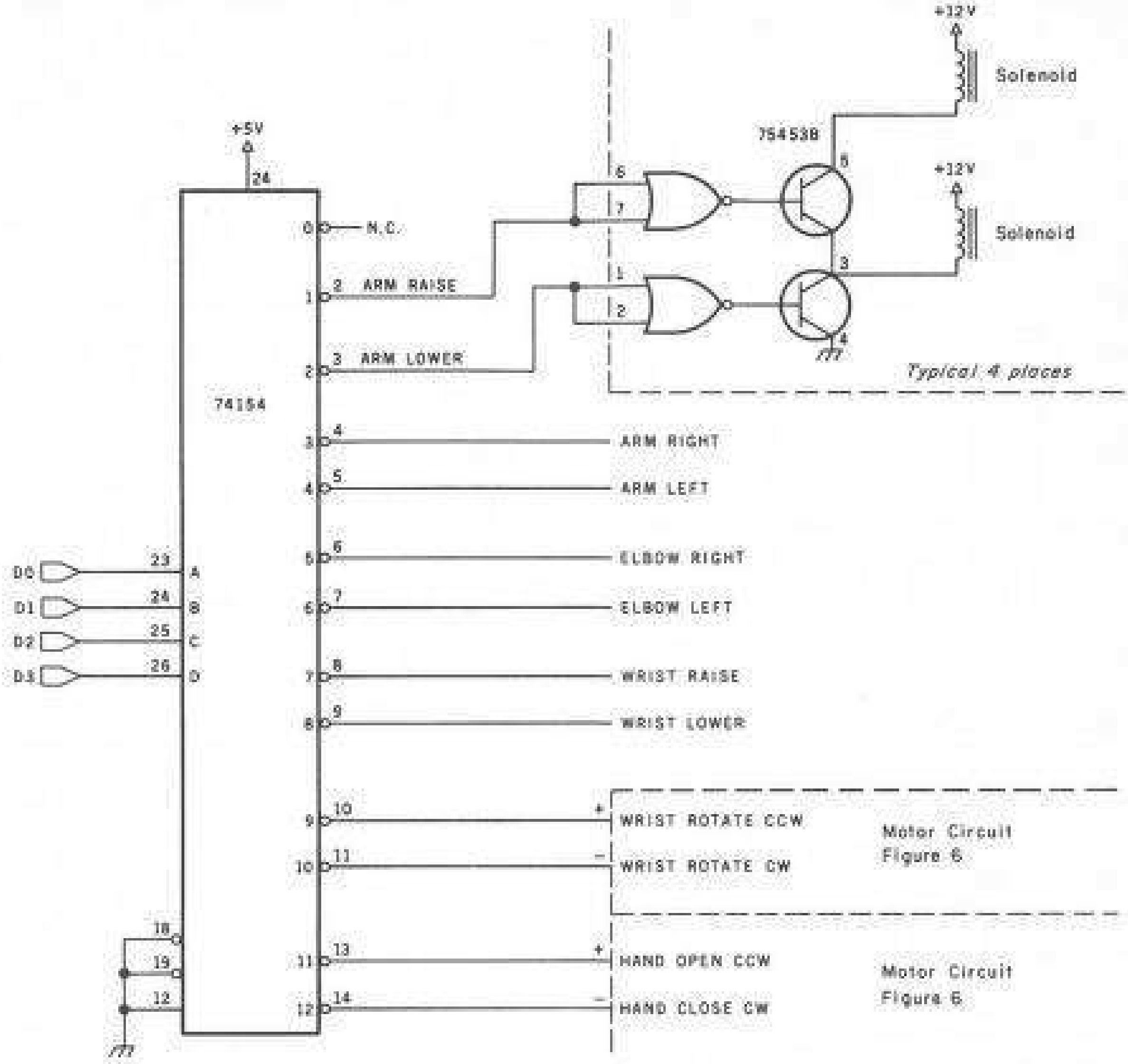


Figure 7. Solenoids can be used to provide linear motion. Energizing a solenoid moves the joystick in the desired direction. Turning off power to the solenoid allows the joystick to return to its center position.

```
1 '
2 '     **AMATRON COMMANDER**
3 '
4 '
5 '    INITIALIZE
6 '
7 '
8 CLEAR 2000
9 ARMOUT=49152
10 DIM CODES$(25): DIM DURATION(25):POKE ARMOUT,0
20 PLACE=1:TIME=0:CLS:KEYVALUE=0
30 PRINT"SELECT THE FOLLOWING"
40 PRINT"1) MANUAL MODE"
50 PRINT"2) AUTOMATIC MODE"
60 A$=INKEY$:IF A$="" THEN GOTO 60
70 IF A$="1" THEN GOTO 100
80 IF A$="2" THEN GOTO 150
90 CLS: GOTO 30
95 '
96 '      MANUAL MODE
97 '
98 '
100 CLS:PRINT"MANUAL INPUT MODE"
110 A$=INKEY$:IF A$="" THEN GOTO 110
120 IF A$="S" THEN POKE ARMOUT,0:PRINT"STOPPED": GOTO 20
130 GOSUB 430  'CONVERT AND SEND
140 GOTO 110
145 '
146 '     AUTOMATIC MODE
147 '
148 '
150 CLS:PRINT"AUTOMATIC MODE"
160 PRINT"SELECT THE FOLLOWING"
170 PRINT"1) TRAIN"
180 PRINT"2) RECALL FROM MEMORY"
190 PRINT"3) RETURN TO MANUAL"
200 A$=INKEY$:IF A$="" THEN GOTO 200
210 IF A$="1" THEN GOTO 249  '     STORE SUBPROGRAM
220 IF A$="2" THEN GOTO 339  '     RECALL SUBPROGRAM
230 IF A$="3" THEN GOTO 100  '     MANUAL MODE
240 GOTO 150
245 '
246 '     STORE SUBPROGRAM
247 '
249 CLS:PRINT"TRAINING"
250 A$=INKEY$:IF A$="" THEN GOTO 250
260 GOSUB 430  '     CONVERT AND SEND
270 CODES$(PLACE)=A$
280 A$=INKEY$
290 IF A$="" THEN TIME=TIME + 1:GOTO 280
300 DURATION(PLACE)=TIME
310 PLACE=PLACE + 1
320 IF PLACE < 25 THEN GOTO 330
322 PRINT"MEMORY FULL"
325 FOR T=1 TO 250:NEXT T
328 GOTO 20
330 TIME=0:GOTO 260
335 '
336 '     RECALL SUBPROGRAM
337 '
339 CLS:PRINT"RECALL FROM MEMORY"
340 A$=CODES$(PLACE)
350 GOSUB 430  '     CONVERT AND SEND
360 TIME=DURATION(PLACE)
370 IF TIME <> 0 THEN TIME = TIME - 1:GOTO 370
375 FOR T=1 TO 70:NEXT T
380 PLACE=PLACE + 1
390 IF PLACE < 25 THEN GOTO 400
392 PRINT"END OF PROGRAM"
395 FOR T=1 TO 200:NEXT T
398 POKE ARMOUT,0:GOTO 20
400 A$=INKEY$
410 IF A$<>"" THEN POKE ARMOUT,0:PRINT"STOPPED":GOTO 20
420 GOTO 340
425 '
426 '     CONVERT AND SEND SUB PROGRAM
427 '
428 '
430 IF A$="D" THEN KEYVALUE = 10:GOTO 470
440 IF A$="O" THEN KEYVALUE = 11:GOTO 470
450 IF A$="C" THEN KEYVALUE = 12:GOTO 470
460 KEYVALUE = VAL(A$)
470 POKE ARMOUT,KEYVALUE:PRINT KEYVALUE
480 RETURN
```

Listing 1. BASIC listing of AMATRON programmable control program written in Microsoft BASIC. ARMOUT is set to a known control location.

Figure 8. Schematic of solenoid interface using same four-bit number system as the motorized circuit.



matron is positioned where you want it to begin. The control program accepts up to 25 different commands in a sequence. You can change this number to suit different memory requirements.

The Program. The Armatron control program (listing 1) begins with an initialization routine. This part of the program sets all variables and memory locations to known values. The system is initialized in manual mode. Any movement key simply activates the appropriate solenoid or motor. Movements are not remembered until the automatic (A) key is pressed. Enter the key scanning portion next. Here the use of the Microsoft BASIC command INKEY is critical. If your version of BASIC does not have this command, you can substitute INPUT A$, but every command key you hit must be followed by a carriage return or ENTER.

Once a key is detected, the command interpreter takes over. It systematically compares what it received with the list of available command buttons.

When in automatic, a string array that contains each entered movement command is formed. First, the key letter, then the duration of activation is stored. When a different key is struck, a new array location is opened. Pauses between commands are not stored.

The B command tells the program to start at the top of the array and begin pulling values of key codes and their lengths, then send them to the operation handler for execution. This is not as simple as it sounds. Study the listing for specific program action and modify to customize it.

This program allows almost the degree of programmability of the motion platform I described in "Constructing An Intelligent Mobile Platform" (*Robotics Age*, July/August 1982, page 41) but it still does not allow total local control. Consider taking the simple Z8 computer system and adapting it for use here. Almost the same parallel interface could exist; only the program would change. How about fitting the motion platform with Armatron, then adding voice command control from last issue? Start working on it. Drop me a line with your results, and in a future article, I will disclose my efforts at fitting it all together. □