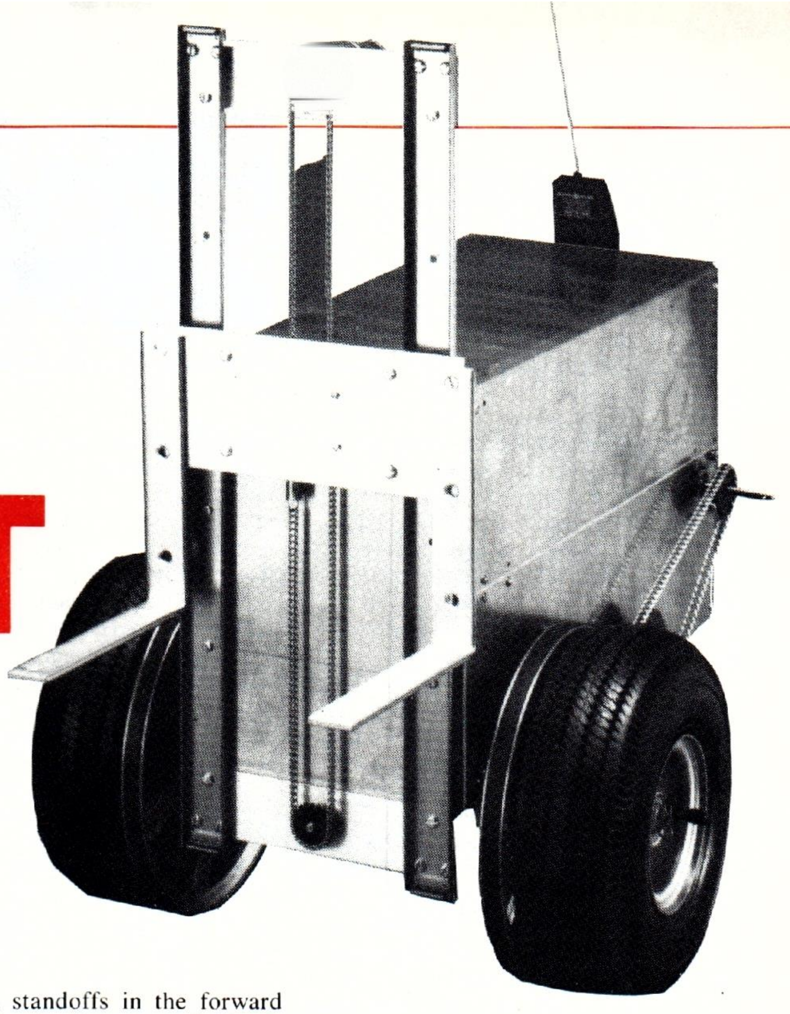
R-E ROBOT



This month we show you how to build the control board.

Part 8 THE PAST TWO MONTHS we've looked at how the robot's control circuitry works. Now its time to get our hands dirty and build the control board.

Construction

Building the control board is a straightforward operation. The double-sided pattern is shown in PC Service. Note that because of its large size, the pattern is shown half size, so it must be enlarged before etching. The board is also available from the supplier mentioned in the Sources box. The supplier provides a board with two-ounce copper, plated-through holes, and a solder mask. If you choose to etch your own board we recommend that you use a blank with two-ounce plating, solder all components on both sides of the board, and install feedthroughs at any unused pads. The reason for the heavier copper is that it provides better power handling capacity and better noise margins.

Once you've either bought or etched the board, check it for power to ground shorts. Those will be very difficult to locate after all of the components have been installed. Then stuff the board following the parts-placement diagram that is shown in Fig. 1.

All of the control-board components can be obtained from most electronics distributors. The Fujitsu relays specified can be found at many relay specialists. If you have trouble finding them, you can substitute units from other manufacturers as long as they have a contact rating of more than 10 amps. If you make any substitutions, you may also need to modify the board to accommodate the substitutes.

The control board is designed to be mounted on standoffs in the forward bulkhead of the chassis as shown in Fig. 2. When mounting the board, it should be oriented so that the terminal strip is located at the top edge of the chassis.

The RPC mounts over the board on one-inch standoffs. Use fixed standoffs at the top edge of the board, and hinged standoffs at the bottom. That will allow the RPC to be swung down and out of the way during troubleshooting.

Holes should be punched in the forward bulkhead for the wires. The motor-power, battery-power, and return wires should all be fed through one hole. All other wires, such as the leads from the shaft encoders, should be fed through a second, separate hole. The return wire from the control board's single-point ground should be as heavy as possible. Also, the motor-power wires should be as heavy as required to handle the current they must carry.

If you are going to use large drive motors with current requirements over 10 amps, the two main switching transistors and their associated diodes may be removed from the circuit board and mounted on the forward bulkhead to take advantage of the huge heat-sinking capacity of the robot's chassis. Use sockets, of course, and connect the sockets to the board with short lengths of heavy-gauge wire. After the control board has been assembled and both it and the RPC have been installed, the forward bulkhead will contain all of the robot's electronics. Now we're ready for bench testing.

Testing

To do the testing you will need a DC supply capable of producing 14–30 volts at 3 amps. If you can't find a suitable supply, you can build one using a high-current transformer, a full-wave bridge, and a suitable filter circuit. Be sure to select diodes (for the bridge) and capacitors (for the filter) whose ratings are appropriate. We built a unit that supplied 18-volts DC at 3 amps for our testing. Whenever the motors were accelerated too quickly, the power supply sagged, the motor relays dropped out, and that brought the motors to a stop. The power supply was completely adequate for testing, however.

Begin testing by connecting the control board to the power supply, but not to the RPC. Apply power and examine the sleep circuit for proper operation. If it is being clocked at 10-Hz as designed, the state of pin 4 should change once every 15 seconds. If all is well, you have confirmed that power is correctly bused to the board.

Proceeding, defeat the sleep circuit by soldering a jumper from R10 to ground, causing RY1 to close. That will energize the system. (Don't forget to remove that jumper when testing and troubleshooting are completed!) Now you should verify that IC9 delivers +5-volts DC and that IC30 delivers +12-volts DC. Also check that +5 volts is available at the correct pins at PL4, the RPC connector.

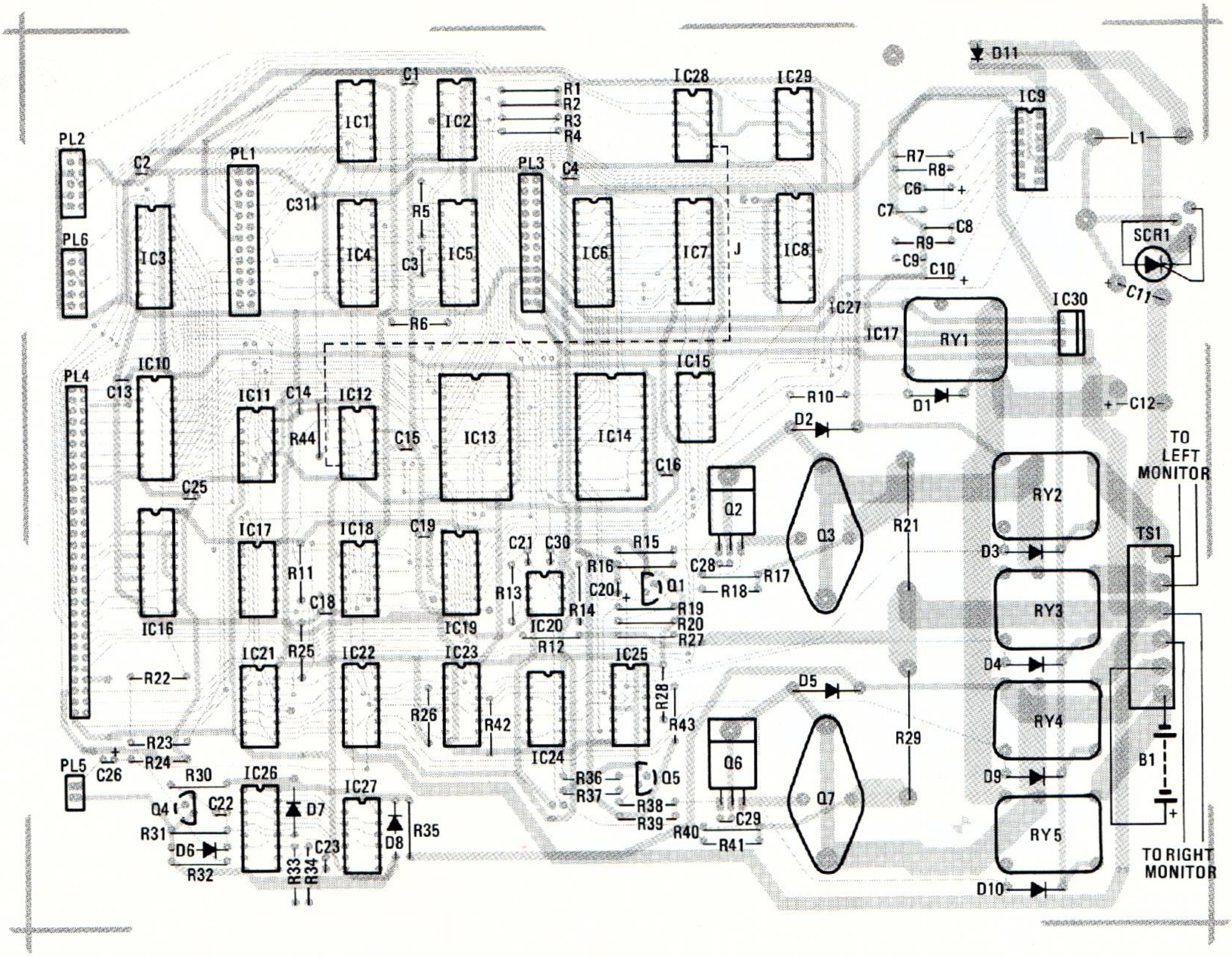


FIG. 1—ALL OF THE CONTROL CIRCUITRY mounts on one double-sided board. Follow this guide when assembling the board; the patterns can be found in PC Service.

PARTS LIST

All resistors ¼-watt, 5%, unless otherwise noted

R1, R4, R6, R7—not used
R2, R12, R16, R18–R20, R22, R23, R26–R28, R30, R34, R36, R37, R39, R41, R44—10,000 ohms
R3—62,000 ohms
R5, R9—15,000 ohms
R8—4700 ohms
R10—220 ohms
R11, R35, R42, R43—1000 ohms
R13, R14—1 megohm
R15, R38—47 ohms
R17, R24, R40—100 ohms
R21, R29—0.1 ohms, 5 watts, 1%
R25, R31–R33—100,000 ohms

Capacitors

C1, C2, C4, C5, C13–19, C22, C25, C27, C31—0.1 µF, monolithic ceramic
C3—100 pF, 50 volts, ceramic disc
C6, C10, C21, C30—2.2 µF, 50 volts, ceramic disc
C7—0.002 µF, 50 volts, ceramic disc
C8—330 pF, 50 volts, ceramic disc
C9—0.047 µF, 50 volts, ceramic disc
C11, C12—2200 µF, 25 volts, electrolytic
C20, C23, C24, C26—10 µF, 16 volts, electrolytic
C28, C29—not used

Semiconductors

IC1, IC2—4051 multiplexer
IC3, IC6—74LS541 octal buffer/line driver
IC4—74LS377 octal D-flip-flop
IC5—ADC0804 8-bit A/D converter
IC7, IC8—74LS374 octal D-flip-flop
IC9—L296 switching regulator (SGS)
IC10—74LS645 octal three-state bus transceiver
IC11—74LS125 quad three-state buffer
IC12—74LS266 quad 2-input exclusive NOR gate
IC13, IC14—8253 programmable interval timer
IC15—74LS32 quad 2-input OR gate
IC16—74ALS520 8-bit comparator
IC17—74LS164 8-bit serial-in/parallel-out shift register
IC18—74LS393 dual 4-bit binary ripple counter
IC19—74LS138 1-of-8 decoder
IC20—LM358 dual op-amp
IC21—74LS259 8-bit addressable latch
IC22—ULN2003 Darlington array
IC23, IC25—4046 PLL
IC24—74LS00 quad 2-input NAND gate
IC26—4060 14-stage ripple counter
IC27—4078 8-input NOR/OR gate
IC28, IC29—dual D-flip-flop
IC30—LM340-12 12-volt regulator
Q1, Q5—2N3906 PNP transistor
Q2, Q6—TIP29A NPN transistor
Q3, Q7—2N3772 NPN transistor
Q4—2N3904 NPN transistor
SCR1—C106Y1 (GE) SCR
D1, D3, D4, D9, D10—1N4001 rectifier
D2, D5, 1N5400 rectifier
D6, D7—1N4148 switching diode
D8—1N754 6.8-volt Zener diode
D11—8R05 Schottky diode (SGS)

Other Components

L1—300 µH
RY1–RY5—DPST relay, 12-volt coil, Fujitsu FBR-631D012 or equivalent
PL1, PL3—26-conductor plug, dual row, 0.025-inch spacing
PL2, PL6—10-conductor plug, dual row, 0.025-inch spacing
PL4—60-conductor right-angle plug, dual row, 0.025-inch spacing
PL5—2-conductor plug, single row, 0.025-inch spacing
TS1—6 connector terminal strip
B1—see text

Miscellaneous: PC board, IC sockets, heat sinks (Thermalloy 601 or equivalent for IC9, Thermalloy 286 or equivalent for IC30), mounting hardware, nuts, bolts, wire, solder, etc.

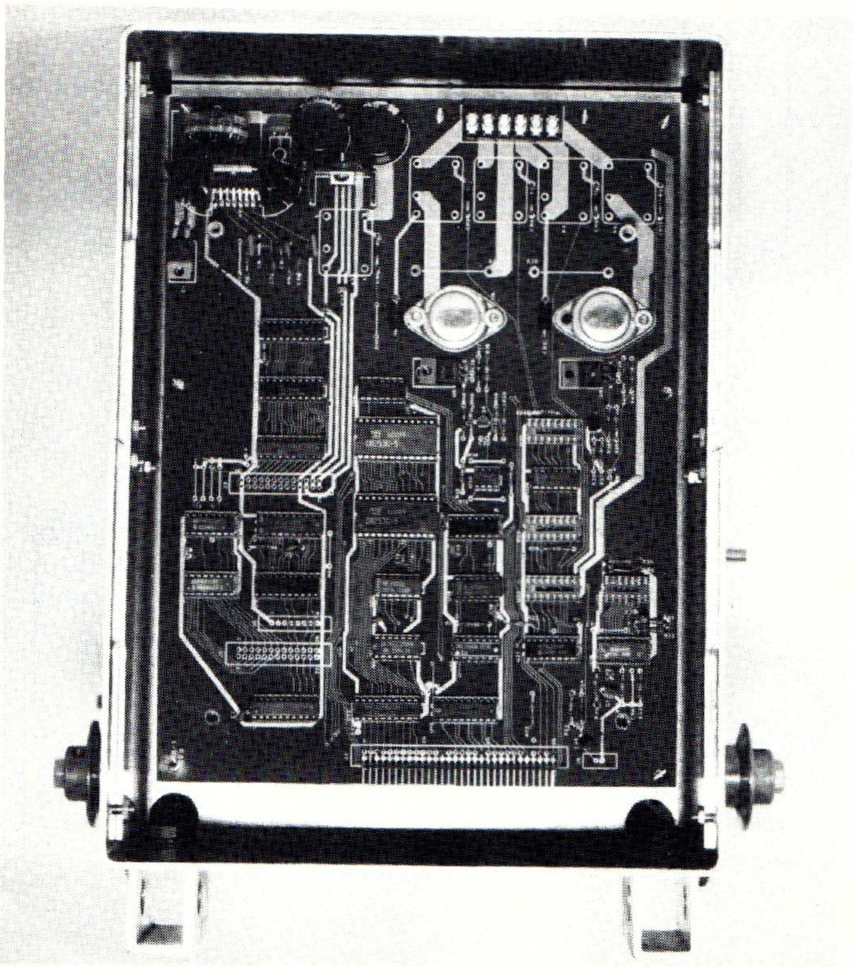


FIG. 2—THE ROBOT'S ELECTRONICS mount on standoffs in the forward bulkhead. The control board is shown here; the RPC mounts above it on hinged standoffs.

TABLE 1—OUTPUT FUNCTIONS

Address	Function
120	Left forward relay
121	Left reverse relay
122	Right forward relay
123	Right forward relay
124	Left motor control enable
125	Right motor control enable
126	Beeper
127	not used

If all is well, connect the RPC. Write the following diagnostic word (in the hex-number base) and execute it:

```
: TEST0 BEGIN 0 127 PC! 1 127 PC!
    ?TERMINAL UNTIL ;
```

As mentioned last time, the scope of this article prevents us from going into a detailed discussion of Forth and its structure. However, note that the while Forth requires statements like the preceding one to be entered as a single line, for space reasons it is impossible for us to show it that way. When you enter such statements, be sure to enter them as single lines or they will not be processed correctly. If you are not familiar with Forth, we recommend the book *Starting Forth*, by Leo Brodie; it is published by Prentice-Hall. You can probably obtain a copy from the Forth Interest Group or at your local computer bookstore.

Let's briefly look at what TEST0 is, and how it works. The colon tells the interpreter to compile the following word called TEST0 into the dictionary. That word is a begin–until loop that will loop until activity from the terminal (?TERMINAL) is detected. The loop itself stores a 0 to port 127H, then stores a 1 to port 127H. Compilation stops at the semicolon and the interpreter returns to the interpretive mode. After compiling TEST0, you can execute your new word simply by typing TEST0 and a carriage return on the keyboard. The word will execute until you touch any key.

During execution, you should observe the output of IC16, the 74ALS520. The address-latching pulse should be about 1 microsecond long, indicating that the wait-state generator is working correctly. Now examine the state of pin 12 of IC21 with an oscilloscope; you should see it toggling. That shows that the RPC and the control board are working together.

Testing the digital inputs and outputs is a very simple process with our operating system in ROM. We have only to write a few diagnostic words and execute them.

The individual outputs can be toggled with the following test word:

```
: TEST1 BEGIN 8 0 DO 0 120 I + PC!
    1 120 I + PC! LOOP 0 UNTIL ;
```

The function of each of the individual outputs is as shown in Table 1.

Next, connect a speaker to the J6 and test the beep function:

```
: DELAY 0 DO 10 0 DO LOOP LOOP ;
: BEEP 1000 0
   DO 1 126 PC! 2 DELAY
        0 126 PC1 2 DELAY
   LOOP ;
```

Notice that here we used "pretty" source-code formatting techniques. That becomes increasingly important as the complexity of our code increases.

continued on page 78

SOURCES

The following are available from Vesta Technology, 7100 W. 44th St., Wheatridge, CO 80033 (303-422-8088): Bare RE-Robot controller board, $41; assembled and tested RE-Robot controller board, $200; assembled and tested RPC, fully populated for the robot function, $294. Add $8.00 shipping per board. Colorado residents add appropriate sales tax. Mastercard and Visa accepted.

Optical endocers (100 counts/revolution, quadrature output) are available from EMC Corp., 373 Hillsboro Way, Goleta, CA 93117 (805-968-3060) for $40 each. California residents must add appropriate sales tax.

TABLE 2

```
: PCX!          ( data address - - - )       ( address is Ø to F )
   SWAP 13Ø PC!                               ( write data to latch )
   F AND DUP  CØ OR 14Ø PC!                   ( write address )
   DUP        4Ø OR 14Ø PC!                   ( set write strobe lo )
   CØ            OR 14Ø PC! ;                 ( set write strobe hi )

: PCX@          ( address - - - data )
   F AND DUP  8Ø OR 14Ø PC!                   ( write addr, read lo )
   13Ø PC@                                    ( get data )
   SWAP CØ OR 14Ø PC! ;                       ( set strobes hi )
```

R-E ROBOT

continued from page 46

The byte-wide input and output ports can also be tested. The following word tests the output latch.

: TEST2 BEGIN 0 150 PC! FF 150 PC!
?TERMINAL UNTIL ;

The parallel input port can be tested with the following test word. Four lines are available to you at PL1:

: TEST3 BEGIN 120 PC@ 10 / . CR
?TERMINAL UNTIL ;

Execute TEST3 and then short some of the inputs to ground. As you short each input, you should see the display on the screen change.

Expansion

The robot can be expanded in various ways. If your expansion project requires full use of the RPC, simply couple your circuits to the RPC bus. To interface the circuitry, you need only duplicate the wait-state generator and the bus-buffer interface described in Part 6 (May, 1987). Select a block of I/O space between Ø1ØØH and EØØØH and start designing.

If your circuit is simple and needs only one or two I/O locations connect it directly to the RERBUS, PL3. Address decoding is accomplished with a single integrated circuit and no bus drivers are needed. For simple digital inputs, digital outputs, and analog inputs, connect the circuit directly to the user connector, PL1.

Operation

Now that we have our electronics in place, it is time to consider the software required to make it all work.

The software commands to be sent to the motor control circuits should follow this sequence:

- Set up timer Ø of each 8253 (left and right wheel control) for mode 3 operation. We write control word 36H to register 3.
- Write a frequency representing a slow speed into timer Ø. We write Ø2ØØH to register Ø.
- Close the forward or reverse relay. Write 1 to location Ø12ØH.
- Now enable the PLL. Write 1 to location Ø124H.

Notice that the relays are closed before the circuit is enabled. That prevents arcing when the contacts close or open.

All those functions are programmed using RCL (*R*obotic *C*ontrol *L*anguage), a sophisticated language that is implemented in Forth. The RCL lets us control the robot's motions and functions using simple commands. Further, because Forth is extensible, RCL is extensible. That means that any code we write becomes part of the language.

That last feature is especially valuable. For instance, to control circuits connected to the RERBUS we have to change the way in which the byte store and byte fetch words operate—it's like writing new PEEK and POKE words in BASIC.

Forth's extensibility allows us to create two new words, PCX! and PCX@, that we can use to access the RERBUS. Those words will operate just like PC! and PC@ but they'll do all of the data manipulation required by the RERBUS. The computer code used to create those words is shown in Table 2.

Notice that we have documented our code with comments to allow you to determine how it operates in case something goes wrong or you want to change it. The comment immediately after the word being defined is a standard Forth-notation comment showing the effect of the word on the stack. For example, PC@ pops one argument off the stack (the address) and pushes one argument on the stack (the data). Next time, we will examine the RCL in greater depth. **R-E**