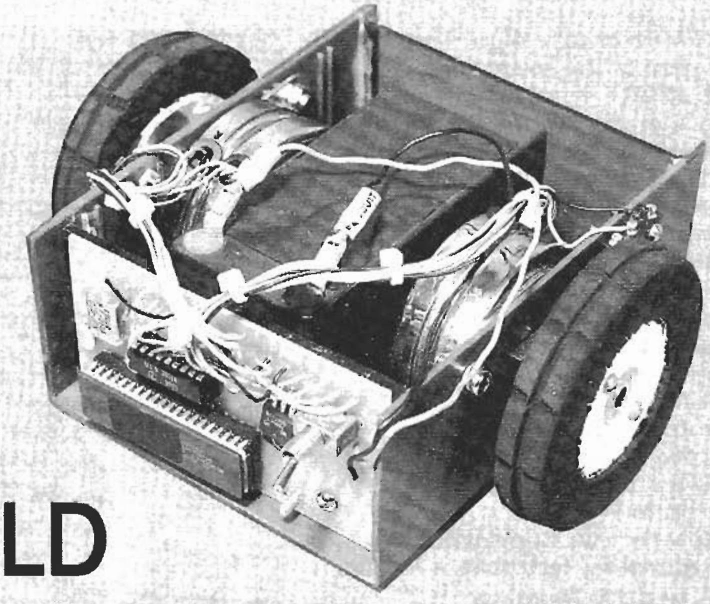


BUILD THE STEPPER-MOTOR ROBOT

FRED EADY

Ken, the friendly robot, is fashioned around an 8748 microcontroller. Build him for less than $100!

PEOPLE TEND TO THINK OF ROBOTS as complex, expensive creatures designed to execute boring, repetitive tasks, day in and day out, for eternity. And sure, there are industrial robots that work like that. But not all robots are boring drudges. In fact, it's possible to build an inexpensive experimental robot that will perform nonrepetitive tasks under program control.

Our robot is called Ken. He's easy to build and fun to program; and in so doing you can learn a great deal about important issues in robotics. Ken is a two-wheeled, battery-powered, free-roaming, obstacle-avoiding robot. Ken's intelligence consists of a single-component microcontroller; his propulsion is provided by two stepper motors driven by a single high-current IC driver. All together, Ken's electrical system consists of eleven electronic and five electro-mechanical devices. (By the way, Ken is not an abbreviation for anything in particular; the author simply thought that it would be rather nice to give the robot some personality.)

Theory of operation

Ken was designed to move freely on a smooth, flat surface. Using the specified stepper motors (or others with a higher torque rating), Ken rolls easily on a tile floor or tight-napped carpet. He has a low center of gravity and is equipped with two oppositely opposed wheels, each driven by a stepper motor. The edge of Ken's smooth plastic case serves as the third point of contact with the floor.

Ken's schematic is shown in Fig. 1. All of his nomadic habits are dictated by IC1, an 8748H microcontroller, that is packaged in a standard 0.6-inch, 40-pin package. In addition to the CPU, the 8748H includes 1024 bytes of EPROM, 64 bytes of RAM, 24 bits of I/O, a crystal oscillator, and an eight-bit timer/counter. For our purposes, we configured IC1 for one 8-bit output port to drive IC2, and one 8-bit input port to monitor motion-sensing switches S2 and S3. Note that Ken doesn't take advantage of all the 8748H's resources, and that leaves you with lots of possibilities for expansion.

Crystal XTAL1 drives the 8748H's internal oscillator; the value of XTAL1 may range anywhere from 1 to 11 MHz. Capacitors C3 and C4 help to start and stabilize the oscillator. However, the author has found that the 8748H will operate reliably without them. Capacitor C1 is used to reset IC1.

Under program control, eight of IC1's I/O ports drive IC2, a ULN2803 high-voltage, high-current octal Darlington transistor array. The ULN2803 can drive a total of eight 500-mA loads with as much as 50-volts DC per load. The ULN2803 has built-in clamp diodes that help suppress stepper-motor noise. ULN2803's can be "stacked" by wiring several of them in parallel. Although doing so was unnecessary for our prototype, you may wish to if you use high-current steppers. Ken's steppers have 12-volt windings that draw approximately 300 mA per winding.

Stepper motor theory

A stepper motor converts electrical pulses into rotational motion. As shown in Fig. 2, a typical stepper motor consists of a permanent-magnet (PM) motor built around two stator cups, with each cup surrounding a separate stator winding. The stator cups form pole pairs that are mechanically displaced by ½ a pole pitch. However, the stator-cup winding pairs displace each other by only ¼ a pole pitch. When the stator windings are energized, the pole pairs are energized in alternation, thus creating North and South magnetic poles. The PM rotor is magnetized, so it aligns on the pole pairs provided by the stator cups.

By changing the polarity of the voltage applied to the stator windings, it is possible to force the rotor to move ¼ a pole pitch. The change in polarity causes opposite poles to attract and like poles to repel, thereby compelling the PM rotor to realign. That realignment is the source of the output shaft's rotational motion. Ken's motors contain 12 pole pairs per stator-winding section, and thus take 48 steps for a complete revolution (7.5 degrees/ step).

Alternately energizing each winding in a predetermined pattern moves the rotor continually in one direction. You can run the pattern backward to reverse rotor motion. Speed of rotation is determined by the rate at which the pattern runs. The standard pattern for Ken's stepper motors is shown in Fig. 3.

Now that we know what drives the steppers, let's take a look at how to drive them.

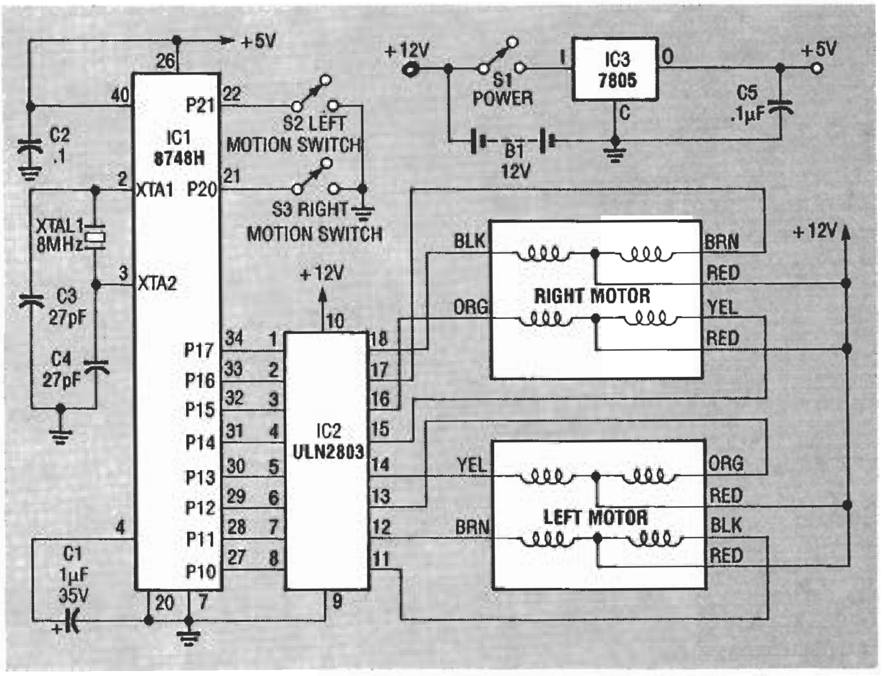


FIG. 1—SCHEMATIC DIAGRAM. Intelligence is provided by IC1, an 8748 single-chip microcontroller.

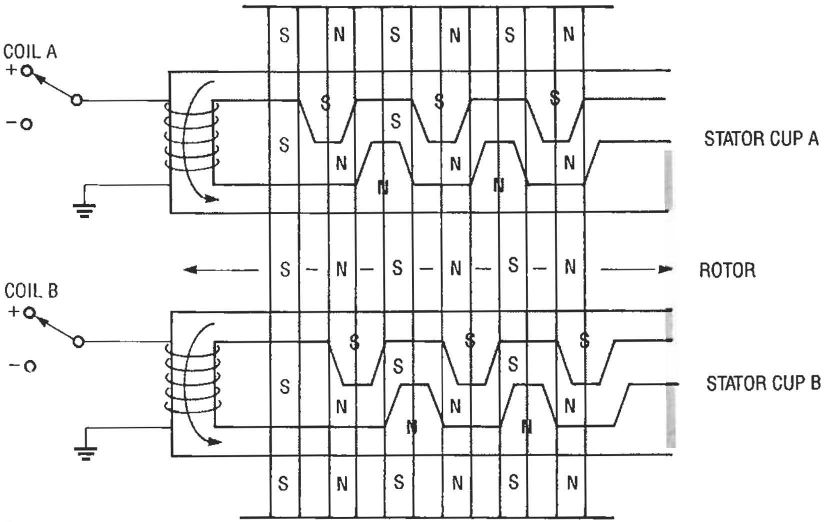


FIG. 2—MECHANICAL DIAGRAM showing the parts of a bipolar stepper motor.

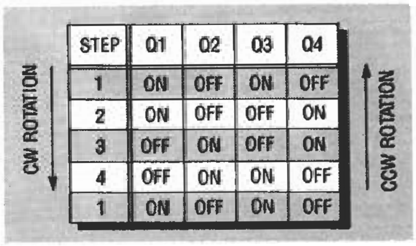
Stepper-motor driver circuit

Figure 4 shows the schematic of a unipolar stepper motor with associated drive transistors. Note that four coils are shown. Each winding consists of two coils wound on the same bobbin per stator half. (By contrast, Fig. 2 depicts a two-coil or "bipolar" stepper.)

You can reverse stator flux in a bipolar motor by reversing the current in the coil. However, doing so requires twice as many drive transistors as a unipolar motor. In addition, as Fig. 5 shows, you must ensure that the circuit does not turn on a series-connected pair of transistors (Q1 and Q2, for example), which would in turn short the power supply.

By contrast, unipolar flux may be reversed by energizing either one coil or the other using a single-ended power supply. In addition, using unipolar motors eliminates the need for extra drive transistors and eliminates the possibility of accidentally shorting the power supply.

As stated earlier, steppers convert electrical pulses into rotational motion. One excellent source of electrical pulses is the output port of our 8748H microcontroller. If one were to equate "on" in Fig. 3 to a binary "1," and "off" to a binary "0," and then apply the pattern to the driver transistors via the microcontroller's output port, the output shaft of the stepper motor would rotate. It's that simple. Unfortunately, we don't have space to print the software listing here, but it's available on the RE-BBS (516-293-2283, 300/1200, 8N1, filename: ROBOT.ASM), and from the source mentioned in the parts list.



STEP	Q1	Q2	Q3	Q4
1	ON	OFF	ON	OFF
2	ON	OFF	OFF	ON
3	OFF	ON	OFF	ON
4	OFF	ON	ON	OFF
1	ON	OFF	ON	OFF

FIG. 3—ENERGIZING SEQUENCE for clockwise and counterclockwise motion.

How Ken maneuvers

Now we understand how Ken moves, but there's still one problem: What if he runs into something? (Earlier we stated that Ken is an obstacle-avoiding robot, but that's not entirely accurate. Ken doesn't really avoid things—he runs into them first!) When Ken confronts an obstinate object, one or both of his wheels will stop. As shown in Fig. 6, the hub of each wheel has two large screws, the heads of which function as cams. As the wheel rotates, one "cam" actuates the associated microswitch every 24 steps (½ revolution). If one of the switches fails to toggle after 24

PARTS LIST

Capacitors
C1—1 µF, 35 volts, tantalum
C2, C5—0.1 µF, disk
C3, C4—27 pF, disk
Semiconductors
IC1—8748H microcontroller
IC2—ULN2803 octal driver
IC3—7805 +5-volt regulator
Other components
B1—12-volt, 1.2 amp-hour gel cell
M1, M2—stepper motor (All Electronics SMT-5, or Airpax A82743-M4)
S1—SPDT toggle switch
S2, S3—SPDT microswitch (All Electronics SMS-90 or equivalent)
XTAL1—8-MHz crystal
Miscellaneous: Heatsink for IC2, case (Radio Shack 270-224 or equivalent), lettuce crisper, brass tubing, rubber tires, PC board, wire, solder, etc.

Note: A kit of electronic parts, not including the case and stepper motors, is available for $39 plus $2 shipping from Fred Eady, 1217 McDonald Street, Fayetteville, TN 37334.

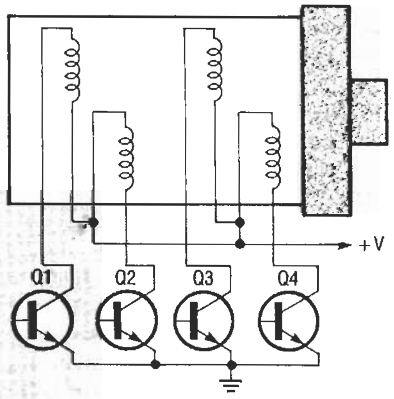


FIG. 4—WIRING DIAGRAM for a unipolar stepper motor.

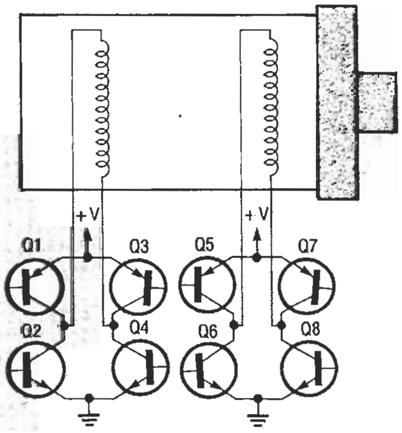


FIG. 5—WIRING DIAGRAM for a bipolar stepper motor. If a series-connected pair of transistors were turned on simultaneously, the power supply would be shorted.

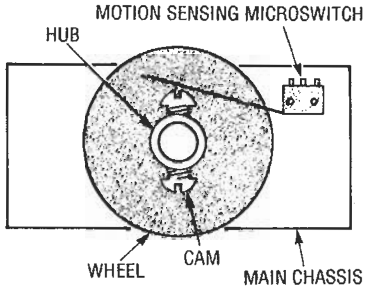


FIG. 6—WHEEL ASSEMBLIES. Two roundhead screws function as cams that actuate the microswitch every half revolution.

step commands from the 8748H, Ken determines that the corresponding wheel has stopped, thus reinforcing the law of physics that states that an object in motion continues in motion until it is interrupted by some external force. Ken, a law-abiding physics student, will then back up and attempt to go around the obstacle.

Ken and Sir Isaac

We know that Ken is intelligent and likes to travel in free space—but how does he do it? Ken's secret is linear force developed by the wheels, which in turn is derived from torque provided by the stepper motors. How do we know how much force is required to move Ken? Mr. Newton put it simply: F = ma, or force equals mass times the acceleration necessary to move that mass.

Ken weighs in at about four pounds; his wheels have a 1.5-inch radius. Taking an arbitrary acceleration of 2 ft./sec^2, let's calculate the force required to move Ken forward at that rate. First, we must convert four pounds to mass by dividing the weight by 1 "g" (32.2 ft./sec^2), leaving 0.124 slug (mass is measured in slugs).

Using our formula, $F = ma = 0.124 \text{ slug} \times 2.00 \text{ ft/sec}^2 = 0.248$ pounds of force to move Ken at 2 ft./sec^2. The rated holding torque of the stepper motor used in Ken is 10.5 ounce-inches. (If you use a different motor, torque may vary.) That value is stated on the motor's data sheet and is determined when the rated DC current is flowing continuously through two of the motor's four windings. Using the formula $F = T/r$ (where F is linear force, T is holding torque, and r is wheel radius), we conclude that $F = T/r = 10.5$ ounce-inches / 1.5 inches, or 7 ounces per wheel. Ken is a two-wheeled creature, so a total of 14 ounces, or 0.875 pounds, of linear force is generated. That's plenty of power (since Ken needs only 0.248 pounds of force), but as with all steppers you lose some force when the motor is rotating.

So, looking again at the data sheet, let's derate the torque by assuming we will be sending 100 pulses per second to the steppers. Doing so derates the maximum torque value by almost 50% and equates to 5.67 ounce-inches of torque. Plugging that value into our formula for linear

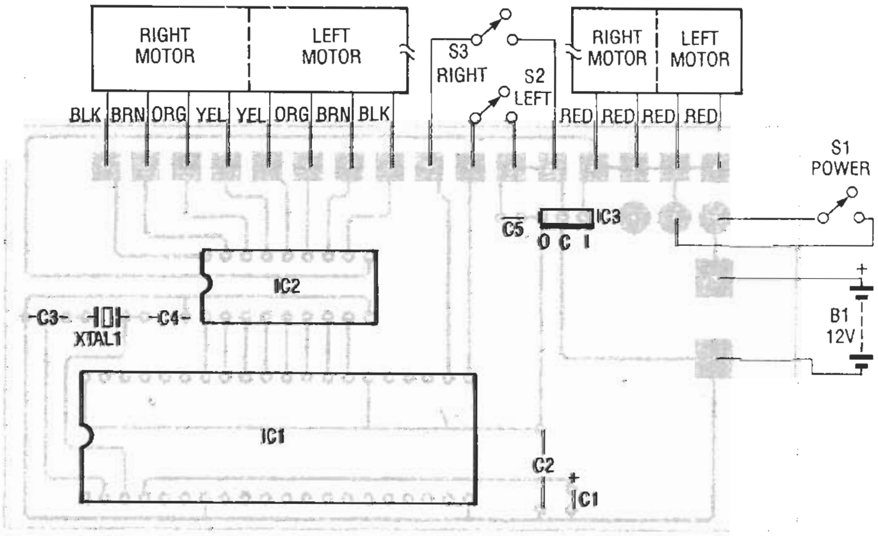


FIG. 7—MOUNT ALL PARTS as shown here. If you use motors other than those specified, you may have to experiment to determine which windings connect to which outputs of IC2.

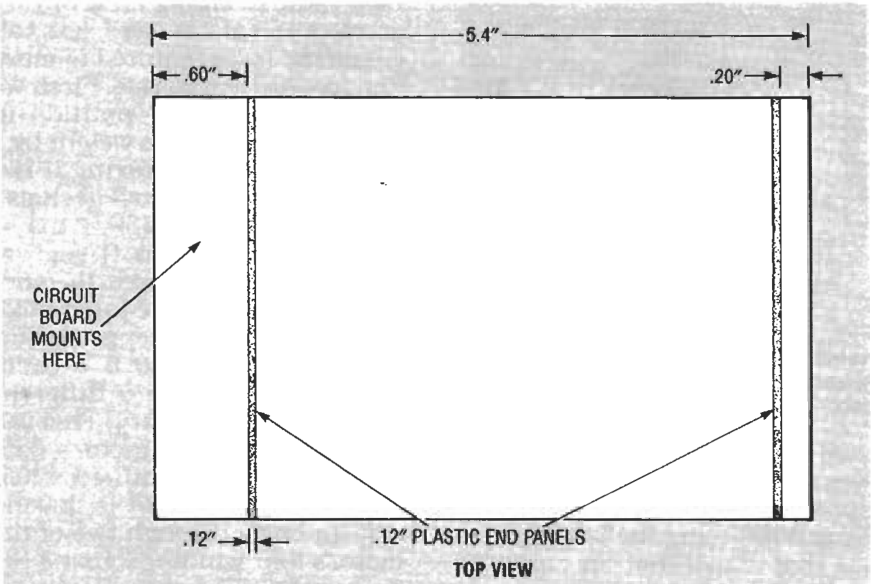


FIG. 8—TOP VIEW. Remove the front and rear ends of the case and glue end panels in place as shown.

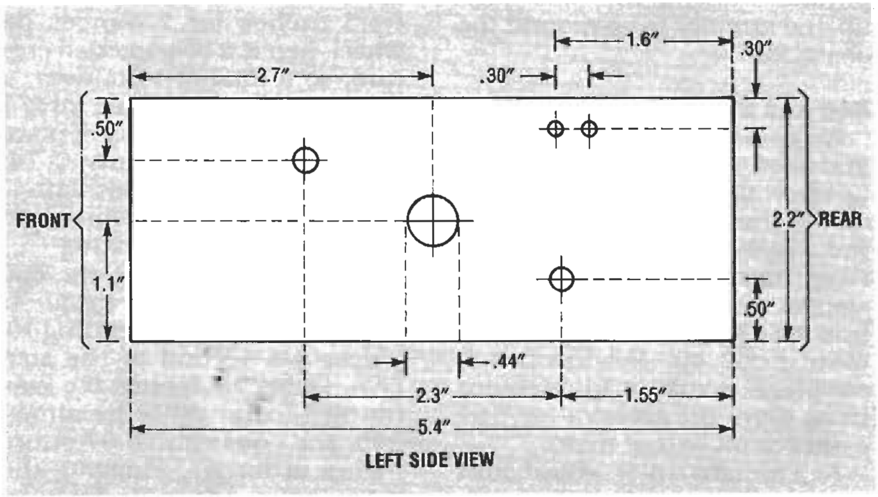


FIG. 9—LEFT SIDE VIEW. The right side is a mirror image.

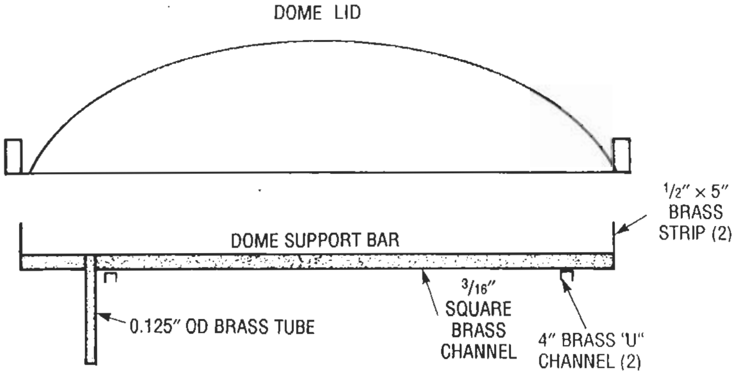


FIG. 10—THE COVER SUPPORT is built from brass channels. Exact dimensions will vary, depending on your cover.

force, we find the result to be 0.473 pounds of linear force, still more than enough.

Enough of this theoretical stuff, let's roll up our sleeves and build Ken.

Electronic assembly

A single-sided PC board allows for convenient construction. Foil patterns are provided, although an etched and drilled board is available.

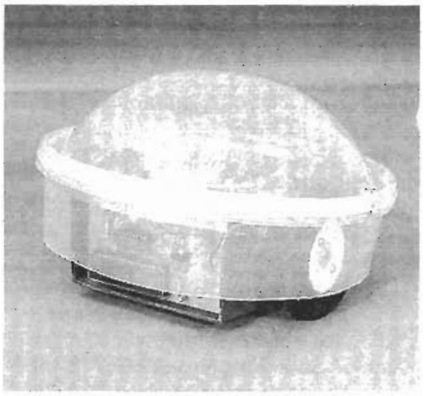


FIG. 11—KEN IS SHOWN HERE with his cover removed.

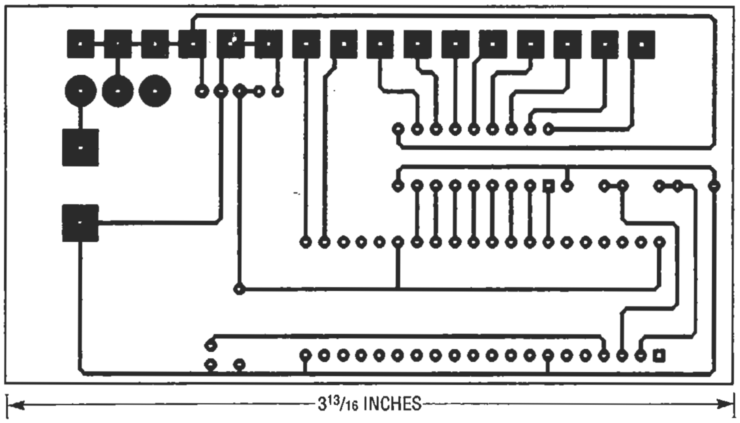
Using the parts layout shown in Fig. 7 as a guide, mount all electronic components. Put a thin coat of heatsink compound on the top and bottom surfaces of IC2, and then slide on a heatsink. Double-check the parts orientation and all solder joints before continuing.

Next connect the stepper motors and microswitches. If you use surplus steppers and cannot determine which winding is which, check the advertisement for an ohms-per-winding value. You can then determine which wires represent independent windings by measuring resistance between various pairs of wires. After you determine which pairs correspond to windings, use trial and error to determine how they should be connected to the PC board. With only four windings, it shouldn't take long to find the right combination.

Mechanical assembly

Ken is built from plastic and brass, although you can use any materials you like. However, If you deviate from our specifications, be sure to recalculate the amount of force required to move your robot and select steppers accordingly.

The body of the prototype consists of a plastic instrument case obtained from a local electronics retailer. That case was selected because it was wide enough to hold both the steppers and B1, a gel cell. We cut off the ends of the case to form a U-shaped channel, and later glued clear red acrylic panels along slots designed for mounting a PC board. You can glue the panels in place using



FOIL PATTERN for Ken's controller board.

modeling cement. The thickness of the panels matches the width of the brass U-channels, discussed below.

Begin assembly by forming and drilling the main body, as shown in Figs. 8 and 9. Next, mount the stepper motors and install the wheels. The wheels may come from inexpensive toys; the author has seen suitable wheels on toys costing less than four dollars. Be sure that the wheels fit the stepper-motor shafts, and that the diameter of the wheels is large enough to raise Ken's body off the floor. Also, use brass or aluminum tubing to shim the stepper-motor shafts, if necessary. In addition, be sure that you can screw or glue some sort of cam surface to the hubs of the wheels to sense wheel motion. The author used oversized set screws for cams.

After mounting the stepper motors and wheels, fit the motion-sensing switches, S1 and S2. Using an ohmmeter to measure continuity, adjust each microswitch to actuate on the high point of the cam surface.

Cover construction

We built a transparent cover for the prototype. (The cover is optional.) We used a lettuce crisper from a local discount department store. First we removed the bottom half of the body, leaving a ring about two inches high. We attached the ring to the domed top of the crisper with a couple of pop rivets. We also custom-built a brass support bar that fits inside the assembly and allows it to rest on Ken's body. Feel free to improvise on our design, but remember to keep the weight down. Also, to provide access to the battery, the cover should not be permanently mounted to the body.

As shown in Fig. 10, the all-brass cover support consists of a long support bar, two thin strips that hold the bar in the lip of the dome, two "U" channels that slip on the end plates of the body, and a vertical tube for stability. Use solder for all mechanical connections. Figure 11 shows Ken with his cover in place.

Final assembly

After checking all connections, connect a 12-volt gel cell to Ken and power him up. Both wheels should turn in the same direction until you stop one (or both). In either case, Ken should reverse both wheels and perform a turn by rotating the wheels in opposite directions. After performing the turn sequence, Ken should again rotate both wheels in the same direction. Also, Ken should turn in opposite directions, depending on which wheel you stop.

Ken in the big world

It's time for Ken's maiden voyage. Just turn him on and let him go. As soon as he runs into something, he should attempt to change direction. Ken is always very busy, and cats love him.

Ken is a simple robot, but the principles of operation are identical to those of large (and expensive!) robots. You can take the basic building blocks presented here and expand on them to build a large and sophisticated device. Have fun, and remember, all of Ken's "intelligence" is under software control. If you don't like the way something works—change it! **R-E**