Section One

Exploring the Human Body

This section is about measuring and testing reactions that occur in the physical body. Although the mind is not in complete control of some of the involuntary physical functions, a certain level of control can be measured and enhanced with proper training using feedback. Heart rate, body temperature, and even reaction time can be controlled to a certain degree with a little training and with the help of some of the devices presented here. Maybe you want to enhance your physical state before an athletic competition? How about showing your friends how you can "beat" a lie-detector test (polygraph)? By training your mind to react based on the output of some of these devices, you can train and enhance the mind and body link and become more in tune with your own physical machine.

This page intentionally left blank

Project One

Biofeedback Device

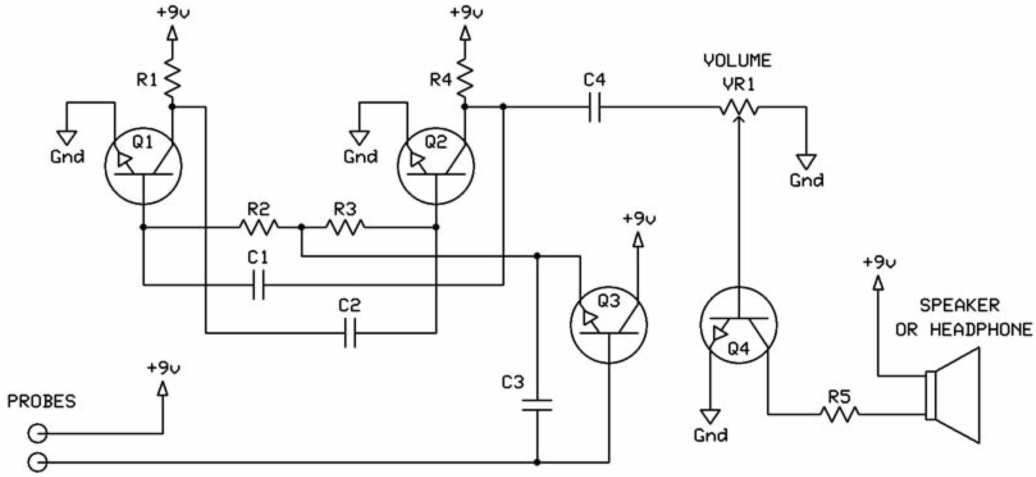
This biofeedback device measures the electrical resistance of the skin and then changes the tone of an audio oscillator depending on the reading. This type of body measurement is also known as *galvanic skin response* (GSR), *electrodermal response* (EDR), *psychogalvanic reflex* (PGR), and *skin conductance response* (SCR). This type of device has been used in the medical field to measure a patient's emotional response and to treat disorders such as phobias, anxiety, and stuttering.

One interesting and controversial use of a biofeedback device is called an *E-meter*, which is used in some forms of the Dianetics and Scientology auditing. This device is formally known as the *Hubbard electrometer*, for the Church of Scientology's founder, L. Ron Hubbard. Interestingly, the Church of Scientology restricts use of the E-meter to trained professionals, seeing it as a religious artifact that can measure the state of electrical characteristics of the static field surrounding the body. The meter is believed to reflect or indicate whether or not the confessing person has been relieved from the spiritual impediment of his or her sins. It can be used only by Scientology ministers or ministers in training, and these devices are manufactured at the Church of Scientology's Golden Era Productions facility in California.

Biofeedback also can be used to measure a person's response to physical activity because the direct result of exertion will be a response in the sweat glands. Maybe you need to learn to speak publicly without breaking a sweat or to beat a lie-detector test? No matter what your "evil genius" motives are, you probably will find ways to use the biofeedback device for your own agenda.

The biofeedback device is a voltage-controlled audio oscillator that increases its frequency as resistance decreases. Thus, the more you sweat, the higher is the pitch of the resulting output. The oscillator also has a volume control so that you won't go insane from the nonstop sound that it produces while in use. For silent operation, the speaker can be removed and the output fed into any multimeter with a frequency-measuring function to display the results in hertz (Hz) rather than an audio signal. Let's review Figure 1-1 to see how the biofeedback device works.

Transistors Q1 and Q2, along with R1, R2, R3, R4, C1, and C2, form a basic audio oscillator that runs on a 9-V battery. To make the tone of the oscillator change in response to voltage, Q3 acts as an amplifier that feeds a voltage back into the circuit between R2 and R3, changing the output frequency. Since the base of Q3 is connected directly to the subject's body along with the 9-V



PARTS LIST FOR THE BIOFEEDBACK AMPLIFIER
Q1, Q2, Q3, Q3 = 2N3904 OR EQUIVALENT NPN R1, R4 = 1K R2, R3 = 10K R5 = 100 OHM C1, C2, C3, C4 = .1 UF DISC (104) VR1 = 10K VARIABLE RESISTOR

Figure 1-1 *Biofeedback device schematic.*

supply, the resistance of the skin creates changes in the voltage feed into the oscillator. To give the audio oscillator some output gain, Q4 is a simple amplifier that can be controlled by turning the variable resistor VR1.

A piezo buzzer also can be used in place of a speaker in case you are tight for cabinet space or don't want the full volume of a speaker. The schematic shown in Figure 1-2 feeds the output from amplifier transistor Q4 through a piezo element rather than to a standard speaker. Piezo buzzers are small coin-sized disks with a bit of crystal attached to one side that respond to voltage changes. These piezo buzzers or elements can be found in cordless phone bases, handsets, and many other devices that beep or blip. The cover of any digital watch is also a piezo buzzer. Figure 1-3 shows a few speakers taken from my junk box along with a piezo buzzer in a plastic case (*bottom right*). Piezo buzzers can be as small as a thumbtack and produce a decent level of sound, so they are great when you need to save space in a black box.

As for the speaker, any large or small speaker with a rating of 4 to 16 Ω will work fine with this circuit, although you probably don't need anything too large. A speaker taken from a dead portable radio would be good. It's always a good idea to breadboard a project before considering soldering the components in case you want to make modifications or test a component

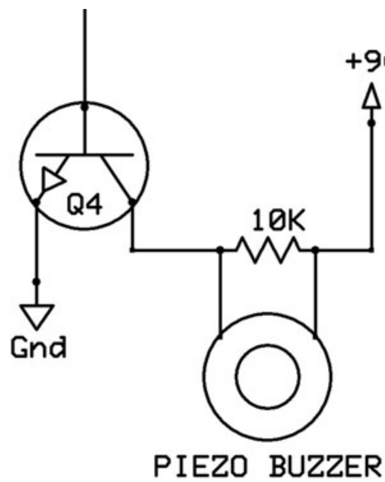


Figure 1-2 *Piezo buzzer alternative output.*

Figure 1-3 *Possible output speaker selections.*

substitution. This circuit is pretty forgiving, so practically any small NPN transistor probably will work. When you are initially testing the breadboard version, just grab hold of two wires for probes because that will work just fine. Figure 1-4 shows the biofeedback circuit being tested on a solderless breadboard.

Once your circuit is working properly, you should hear nothing out of the speaker until you grab hold of the probe wires or short them directly together. When shorted together, the audio oscillator will generate its highest frequency, which will be similar to the output if a person is really sweating profusely. If you do not hear any audio when the probe wires are shorted together, turn the variable resistor VR1 back and forth to make sure that the volume is not turned off. If there is still no output, then you have either a wiring problem or a wrong component. After 30 years as an electronics hobbyist, I have learned that 99 percent of all problems are wiring problems, so recheck the wiring if your circuit fails to work. Once you are happy with operation of the circuit, it can be moved to a more permanent home for installation into a cabinet.

Unless my circuit has a large number of components or integrated circuits (ICs), I always snap off an appropriately sized piece of perf board and then add the wiring on the underside of the board. Copper-clad perf board is also nice, but it can cost a lot more and make it more difficult to change or fix wiring at a later date. Since there are only 13 small semiconductors in the biofeedback device, it fits nicely on a 1×2 in piece of perf board, as shown in Figure 1-5. If you copy the general parts layout that was used on your breadboard, then it will be easier to add the wiring. Also, if you have the spare parts, it is a good idea to leave the breadboard version alive while you build the perf board version so that you can compare the two if something does not work on the final version.

Most of the simple devices in this book will fit into a small plastic hobby project box with room for a battery and necessary switches. Most electronics supply stores will have various sizes

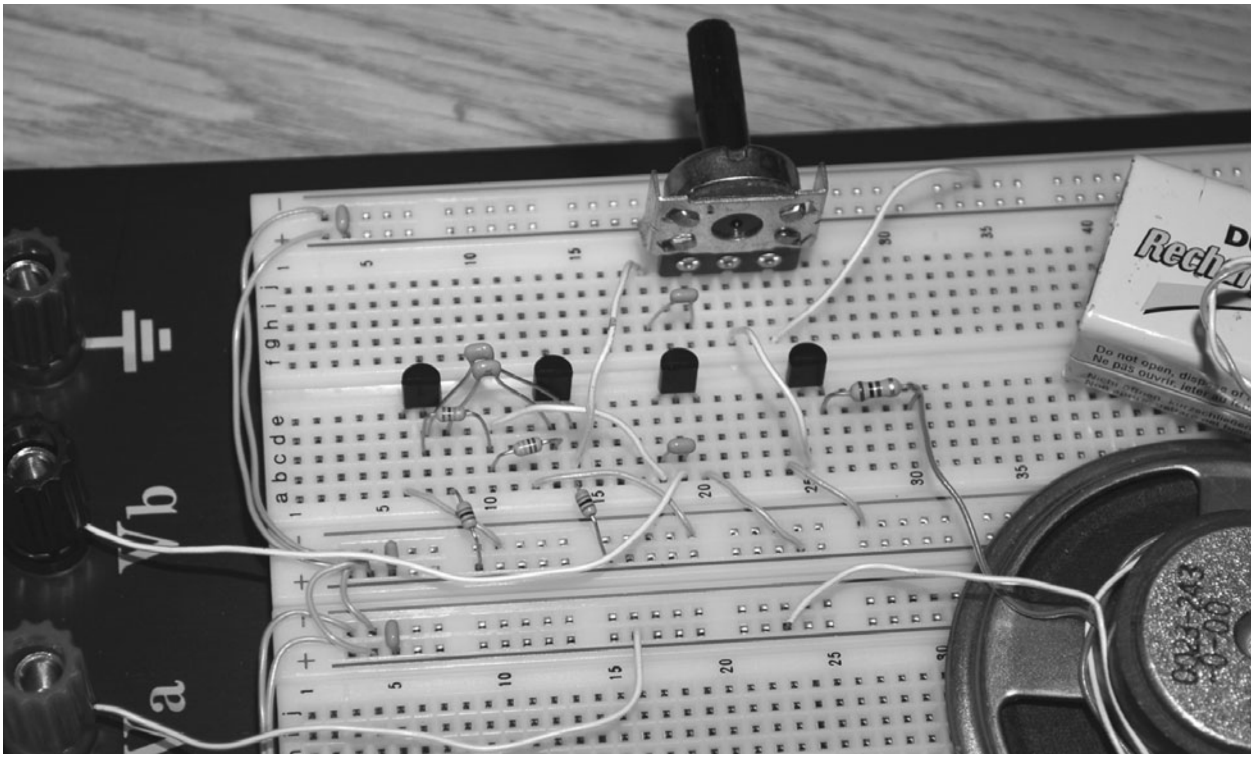


Figure 1-4 *Testing the circuit on a breadboard.*

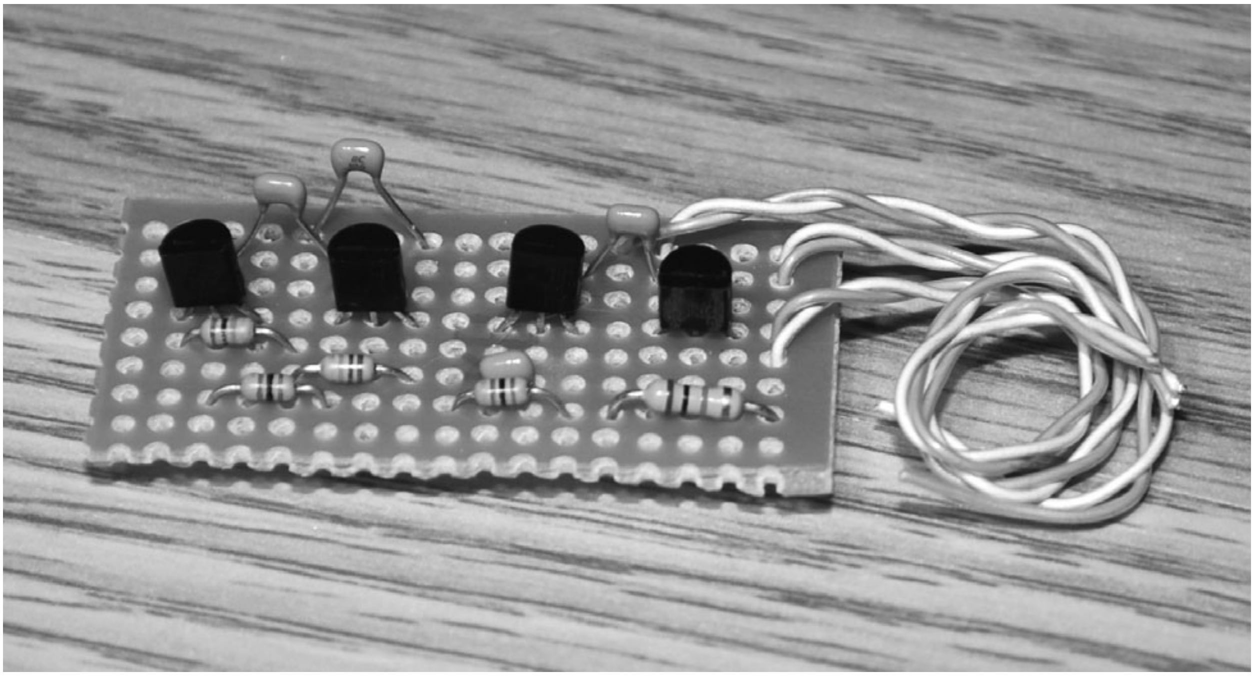


Figure 1-5 *Migration from breadboard to perf board.*

of plastic boxes, and you could even use those gray ABS plastic electrical boxes found at hardware stores to mount your projects. If you are on a really tight budget, you always can get creative and look around the house for a suitable enclosure, such as a soup can or an empty product container. Figure 1-6 shows the battery, variable resistor, on/off switch, and mounting terminals fit into a small black plastic box purchased from RadioShack. I always install an on/off switch between one of the battery wires by default, and a circuit like this can run from a good battery for a long time.

Once you find a project box large enough to hold the battery and hardware, you will need to consider the size and shape of the circuit board. Figure 1-7 shows how I came up with the 2 × 1 in size for the perf board, which was just large enough to fit alongside the battery and contain the 13 components. If the underside of the circuit board is in contact with the battery case or any of the mechanical parts, you can wrap it in electrical tape once it is tested and working to create an insulating barrier. Also, don't forget about all the connecting wires when choosing a cabinet because they can take up a bit of space when it comes time to cram the lid on the box. Now, the biofeedback device is almost ready for use.

Although the biofeedback device works perfectly fine just by grabbing the probe wires between your fingers, this will not be a reliable way to measure your skin resistance because the harder you grip the wires, the more resistance you create. Someone trying to "trick" the unit simply could vary the tone by changing his or her grip on the probes, so some way to attach them to the body will be necessary. Two methods I have found that work very well are copper finger bands and copper plates taped to the subjects' arms. Both finger bands and plates can be made from some copper tubing from the hardware store, as shown in Figure 1-8. The finger bands are small lengths of 1-inch-diameter copper tubing with a slit cut along the length so that they can be safely placed over a finger. The slit allows the diameter to be tweaked, if necessary, by placing a flat-head screwdriver in the slit to widen the opening. This is also a safety release function in case the ring becomes stuck on a finger. The flat plates shown in Figure 1-8 are also bits of the same copper

Figure 1-6 *Finding a suitable cabinet.*

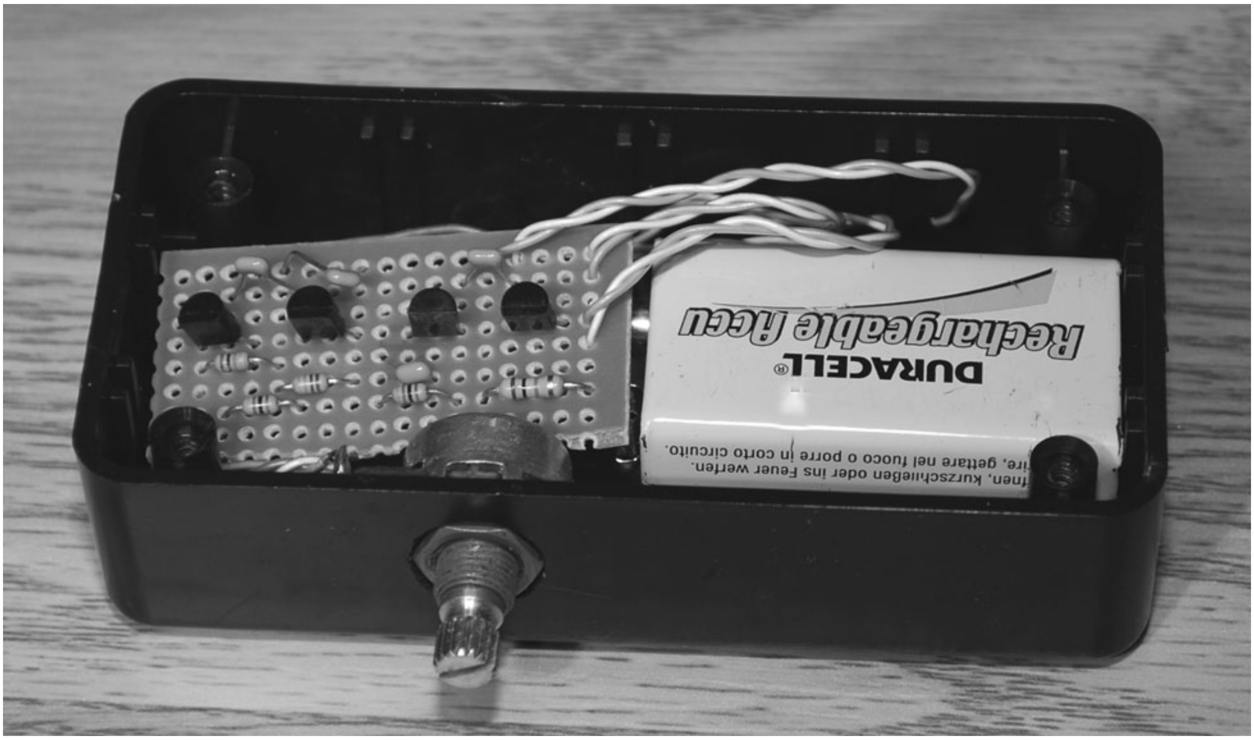


Figure 1-7 *Test-fitting the circuit board.*

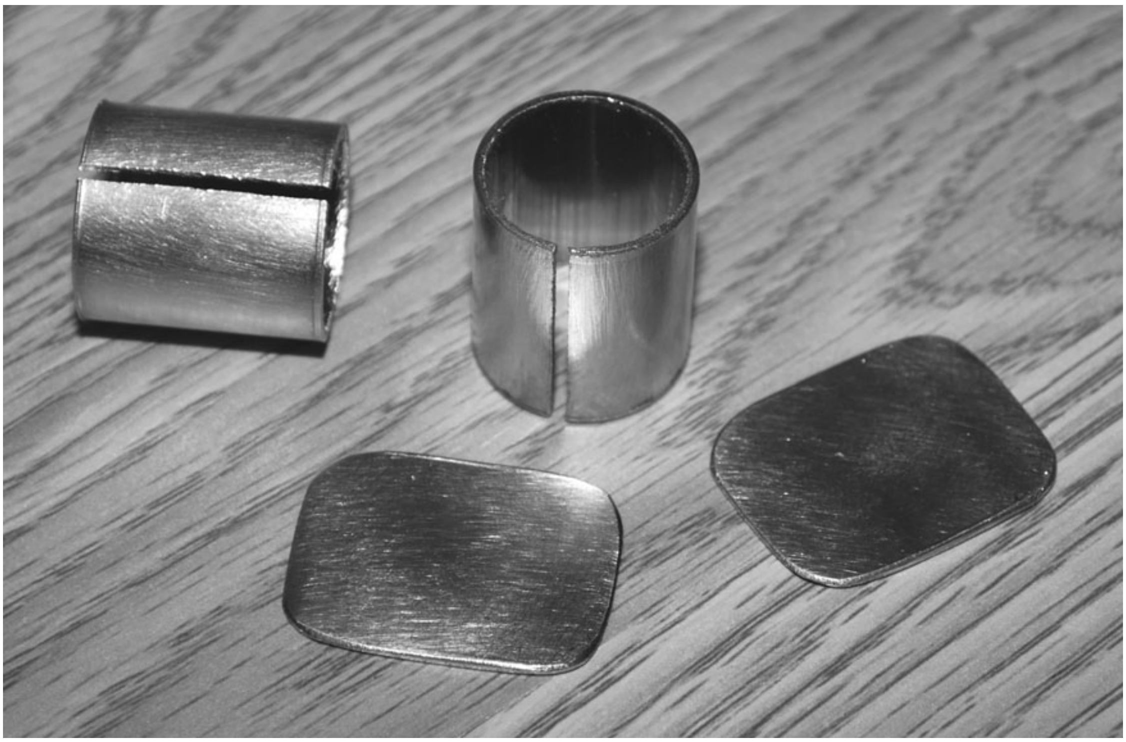


Figure 1-8 *Making finger and body probes.*

Figure 1-9 *Test-fitting the two finger probes.*

tubing hammered flat into plates. All edges should be sanded or filed so that there are no sharp edges, and the copper can be cleaned with some steel wool for best conductivity.

The 1-inch-diameter copper tubing should be a good fit for most people, either a finger or thumb. If you find that the rings are too small or too large, then adjust them by squeezing or prying open the slit to change the diameter. The rings should slide on easily but also make good contact for the biofeedback machine. Some type of probe jelly normally is used to make a better skin connection, but in this case that would defeat the purpose of the machine because it is the resistance between the skin and the probe that is being measured. For this reason, probes are always installed on dry skin.

The quick-disconnect jacks shown in Figure 1-10 are useful because you can easily change the probes when needed. You might even want to try multiple probes in different series or parallel configurations to see what happens. Now that your biofeedback device is ready to use, you must come up with some creative "evil genius" ways to use it. To verify that it is functioning properly, get comfortable, and connect the probes so that you get a steady unchanging tone out of the box. It may take a minute for the tone to stabilize as the moisture between the probes and your skin settles, so relax until the frequency seems steady. Once the tone has not changed, contract your leg muscle as hard as you can so that you exert yourself somewhat, and the tone should increase slowly as your skin resistance changes. Responses will be slow and gradual, with increases in frequency occurring much faster than decreases because moisture evaporates to lower the frequency. For more accurate and quiet readings, you can connect a frequency meter across the speaker output or remove the speaker completely and just measure the frequency.

Figure 1-10 *The completed biofeedback device and probes.*

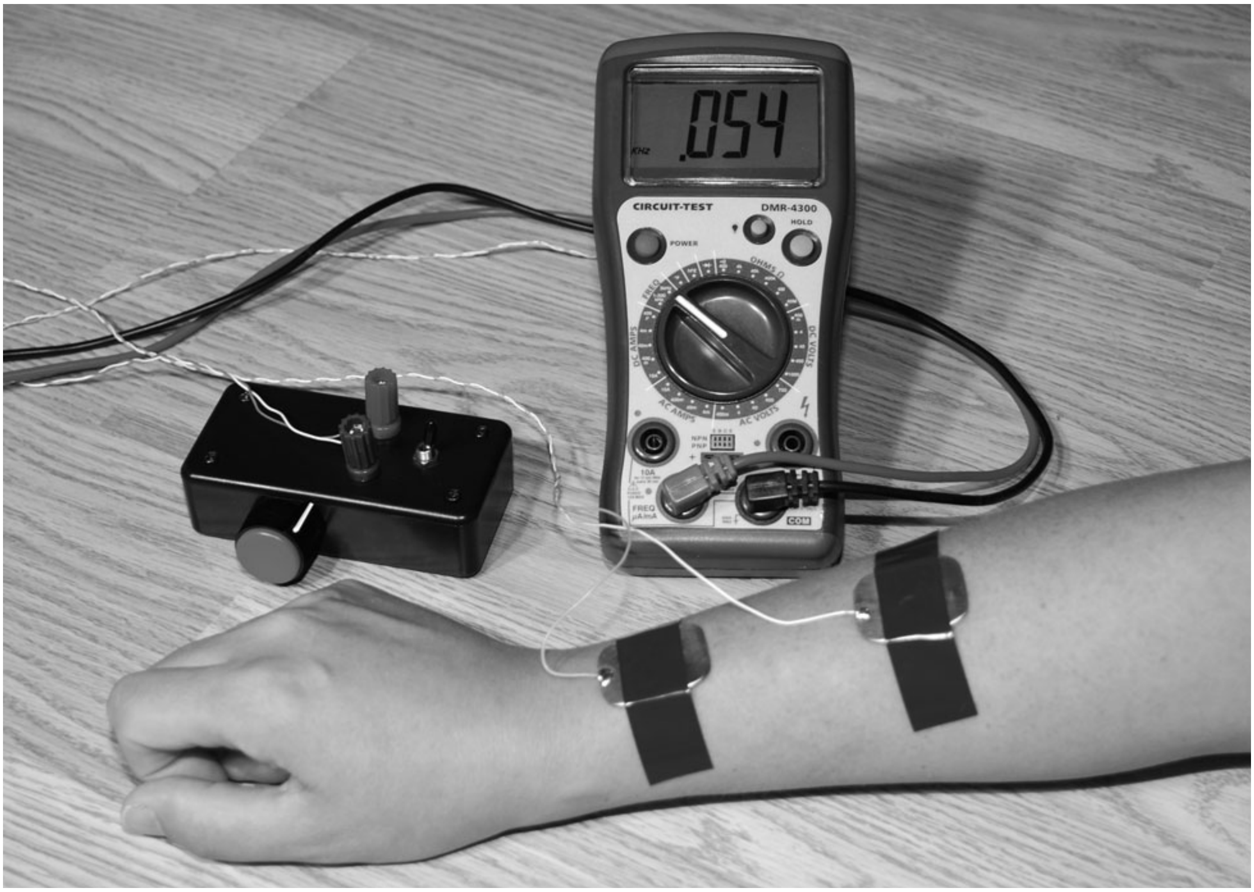


Figure 1-11 *Using a frequency meter for silent operation.*

I added an output jack to my cabinet so that I could plug in an external frequency meter, as shown in Figure 1-11, for more accurate and silent operation of the device. Silent operation may be preferred if you don't want your subject to influence or try to trick the machine. By connecting a frequency meter instead of a speaker, you can see much smaller changes in the frequency, which also may help those who are a little tone deaf when looking for small changes. My multimeter also has a serial port, so I can connect it to a PC and graph the results in real time, making it easy to compare a long-term test over time. Also notice in the figure the use of arm probes, which are held in place by some electrical tape. You may have to play around with the best place to connect the probes for the desired result, but keep in mind that bare skin makes for better testing.

Well, I hope that you have fun with this device and that there are many other things you can do to modify or improve on the design. The placement and types of probes used certainly will affect the results, so get creative and try some now ideas. Maybe a pair of metal spheres held under the armpits? How about some kind of forehead band? Another thing you could try doing is to use the device to check moisture in soil by making probes out of some nails. Add a power supply for continuous operation, and you now have a water flood alarm for your basement by placing the probes in a problem area. I'm sure that any "evil genius" will come up with all kinds of uses and modifications for this device. Have fun!

This page intentionally left blank

Project Two

Reaction Speedometer

This reaction speedometer will measure a person's motor reaction time to a series of lights or a trigger sound. The test begins as soon as the tester flips the switch on the main unit, causing the 10 light-emitting diodes (LEDs) on the box to begin lighting in sequence. The subject is instructed to flip his or her handheld switch as soon as he or she sees either the first LED light or when he or she hears a sound from the optional sound add-on circuit. The fewer LEDs that are on once the subject flips the switch, the higher is his or her reaction time. To increase the difficulty of the test, a variable control allows the LED sequence to be adjusted from a slow crawl to a lightning-fast chase that few will be able to keep up with.

Because this project also includes an optional sound add-on that will send an audio tone once the first few LEDs light up, you can test your subject's motor responses to light, sound, or both at the same time. The reaction speedometer also can be used to hone one's response time for such things as improving in sports, martial arts, or even video games. Daily testing also can give some cues to alertness or the effects of such things as sleep or caffeinated beverages.

The heart of the reaction speedometer is the 74HC4017 (or 74LS4017) decade-counter IC, which can turn on one of 10 outputs in sequence each time a clock pulse is sent. The clock pulses are variable, so the speed of the LED sequence can be controlled to make the test easier or harder.

Figure 2-1 shows the main part of the reaction speedometer schematic without the optional sound output circuit. As you can see, the 4017 decade counter (IC2) is in charge of lighting all the 10 LEDs in sequence, which will happen at a rate controlled by the clock-pulse circuit made from the 555 timer (IC1). There are also two switches—one that allows the tester to start or reset the test sequence and another to allow the subject to freeze the sequence in order to complete the test. A variable resistor (VR1) adjusts the speed of the clock pulses so that the test can be adjusted to the subject's best abilities.

Although the schematic is quite basic, it is always a good idea to first build the device on a solderless breadboard for testing so that you can verify its operation and make any possible modifications before heating up the soldering iron. You also may want to build the optional sound add-on to enhance the test with an audio cue as well as the visual light show.

The optional sound add-on lets the subject respond to an audio tone that will occur as the first few LEDs light up, enhancing the test. The simple schematic shown in Figure 2-2 is an

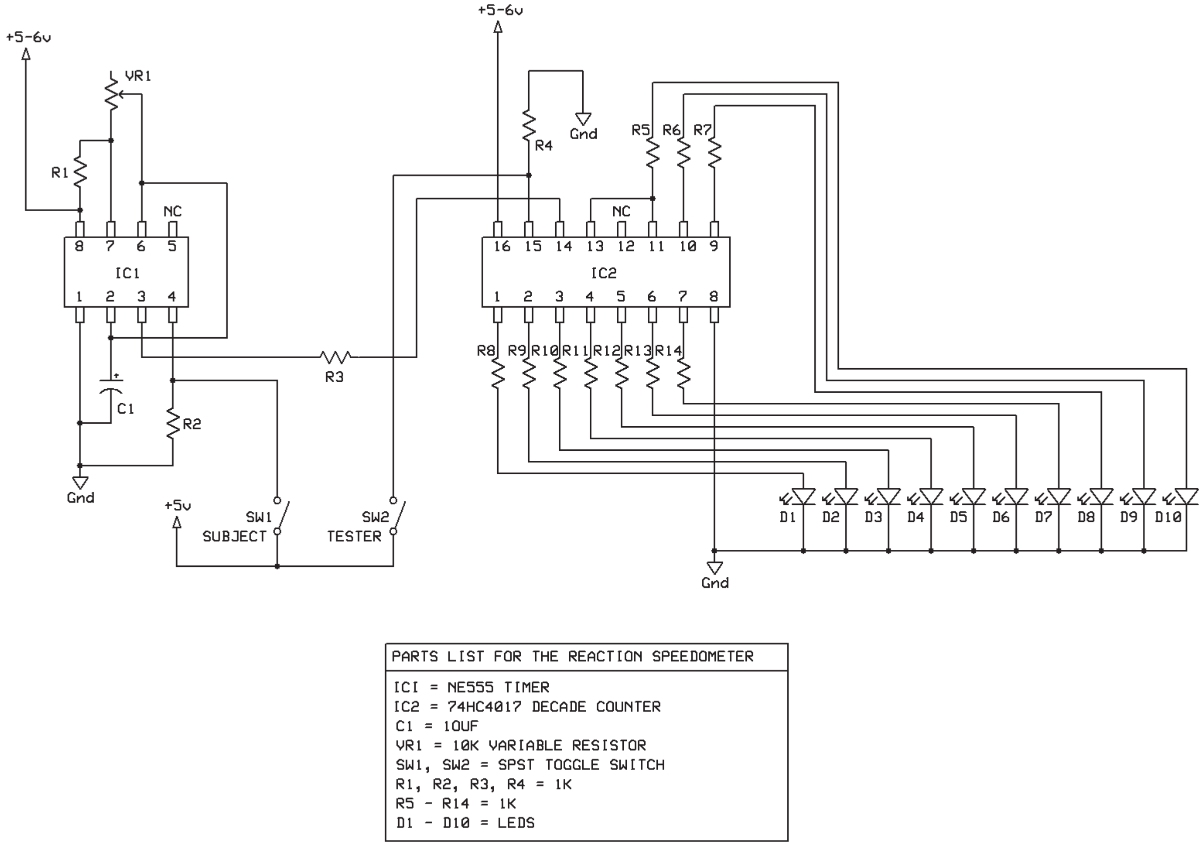


Figure 2-1 *Reaction speedometer schematic.*

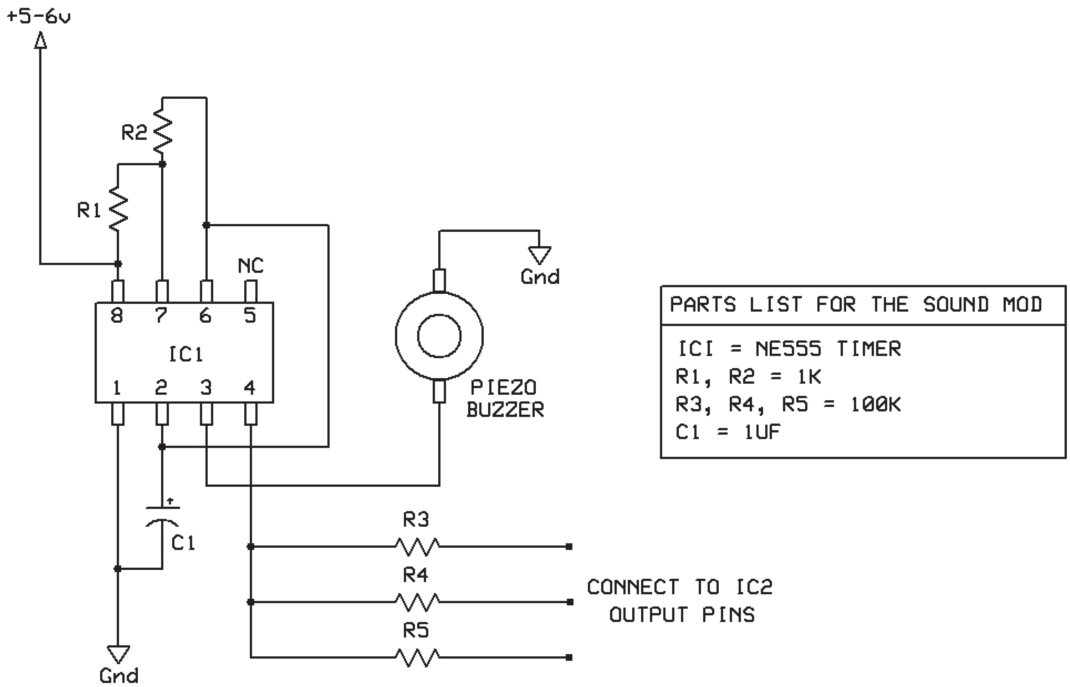


Figure 2-2 *Sound add-on schematic.*

audiofrequency oscillator made from another 555 timer and is practically the same circuit as the original 555 clock circuit shown in the main diagram. The output from the timer is fed into a piezo buzzer so that an audible tone will be heard as soon as the circuit is triggered. Triggering is accomplished via pin 4 and the resistors (R3–R5) that connect to the output pins on the 4017 decade counter.

Depending on which output pins you connect to the sound trigger, you can alter the length and timing of the audio tone as the 4017 steps through its outputs. In my reaction speedometer, I decided to connect to output pins 2, 3, and 4 so that the sound is on for about a third of the time as soon as the test sequence begins. I did not connect to the first output pin because doing so would mean that the sound would be on while the test is in ready (reset) mode, which causes the first LED to remain lit. If you want the tone to last longer, add more connection points. If you want the tone to start later on in the sequence, move the connection points to higher output pins on the 4017 counter. To alter the tone, play around with different values for R1, R2, and C1.

Figure 2-3 shows the reaction speedometer built on a solderless breadboard for initial testing. Notice how each of the three subcircuits (i.e., sound, clock, and counter) are shown in the figure. To make the circuit work, you also will need a pair of single-pole, single-throw toggle switches for the subject and tester. The tester's switch will be mounted to the main box, or it also could be made remote like the subject's switch, to be held in one hand. In one position, the tester's switch will reset the counter, causing the first LED to stay lit. In the other position, the test will start, causing every other LED to light in sequence at a rate controlled by the variable resistor in the 555 clock circuit.

Once the test is in motion, the subject must throw the switch as fast as possible to freeze the test, which will cause the clock to stop. If the test gets all the way to the last LED, the subject has failed, and the test will need to be reset. If the subject cannot beat the test before the last LED is lit, the tester will have to lower the clock rate until the subject can freeze the test somewhere between LED number 2 and LED number 9.

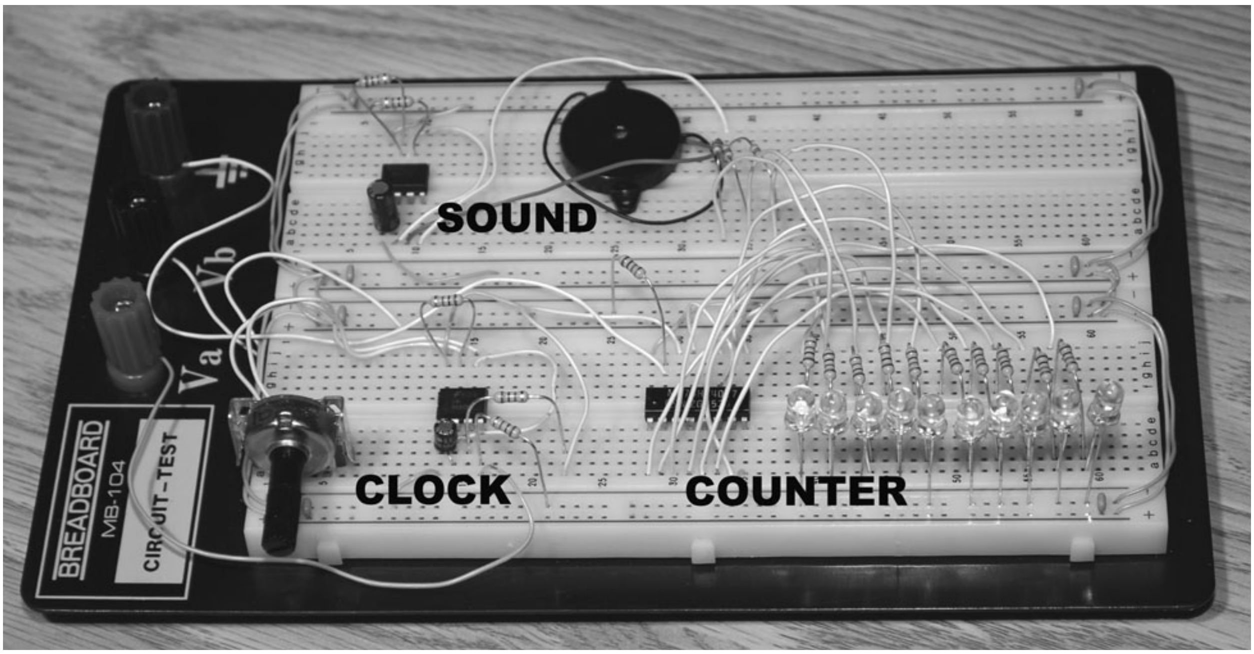


Figure 2-3 *Breadboarded test circuit.*

When you have verified the operation of the circuit and made any necessary modifications, you can move the components to a more permanent home on some perforated board for installation into a cabinet.

Since I added the sound part of the circuit after I built the main circuit, only the clock and counter are on the perforated board shown in Figure 2-4. There probably was enough room to jam the sound add-on onto the board as well, but I thought it might be handy to have the sound board separate to be reused as a generic audio oscillator in some other project later. In my usual style of building a perf board circuit, the component leads are bent on the underside of the board, and all wires are also soldered on the underside of the board. There a quite a few wires coming from the circuit board owing to the 10 LEDs, two switches, variable resistor, and power wires needed.

The sound add-on circuit board shown in Figure 2-5 is completed in the same manner as the original circuit, using a small bit of perf board and adding all the wires to the underside of the board. Also shown in the figure is the small piezo buzzer, which could be replaced by a standard speaker if you wanted by adding a 250-Ω resistor in series with the speaker leads to reduce current draw on the timer output pin. Having the sound circuit on a separate board is handy if you need a simple sound system that can be triggered by a voltage change on some output pin.

There are several ways that you can set up the reaction speedometer for use. The simplest method is to have all the tester's controls in one main box, and place the subject's switch remotely in some type of handheld container. You also could make the tester's start switch the same way so that the LED speedometer is placed between the tester and the subject. A simple handheld switch unit can be made by placing the toggle switch inside a small plastic box or container such as the film container shown in Figure 2-6. The subject now can hold the unit in one hand

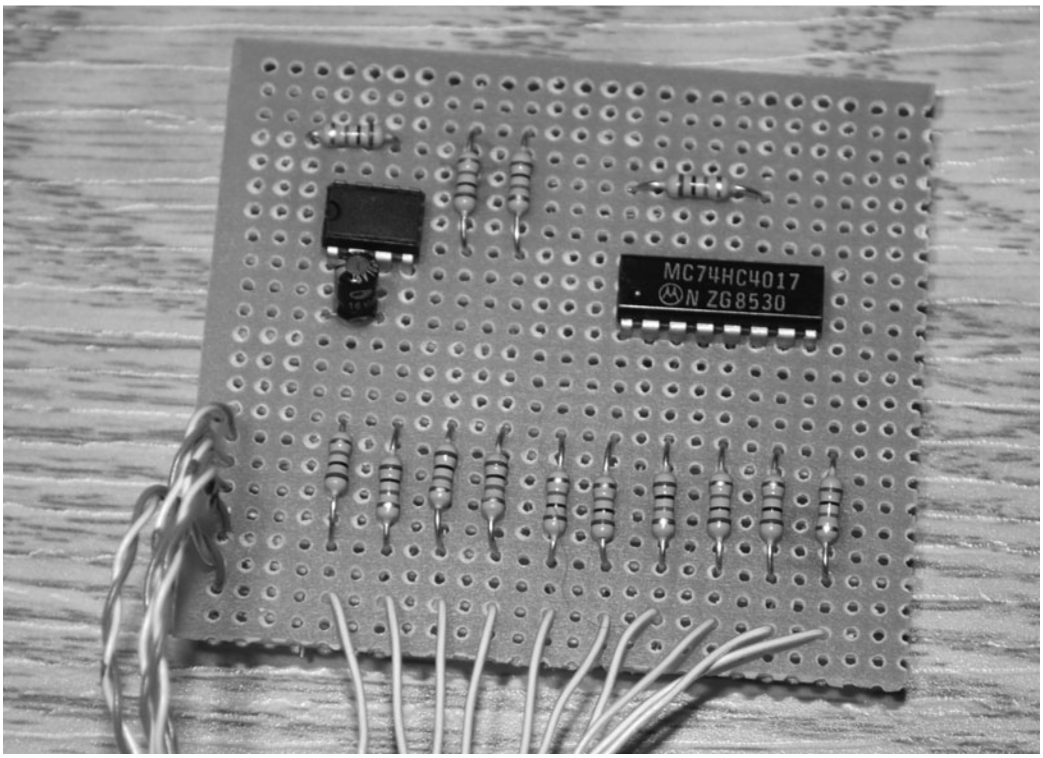


Figure 2-4 *Biofeedback circuit on a perf board.*

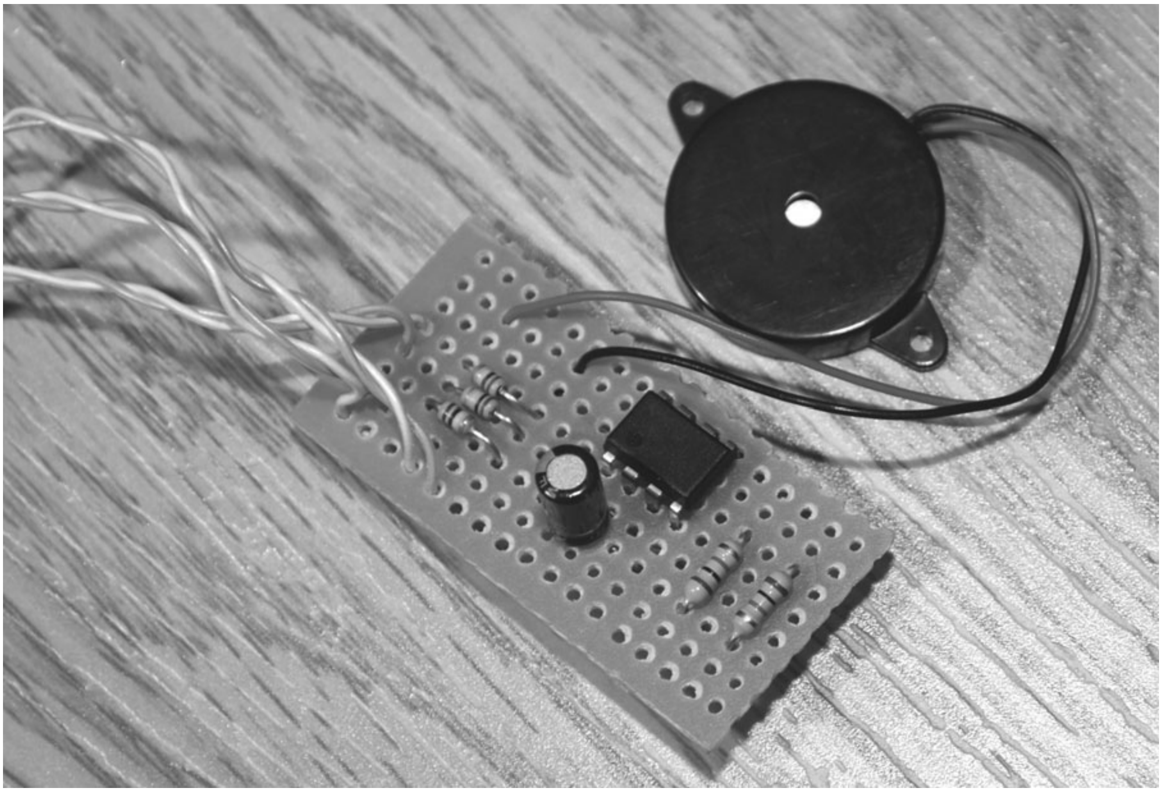


Figure 2-5 *Sound add-on perf board.*

Figure 2-6 *Subject's handheld switch.*

and work the switch with his or her thumb. I kept the tester's switch on the main box, but the advantage to having both switches remote from the main box is that there will be no distractions as the tester flips the switch to start the LED sequence counter.

Both circuits will run from 5 or 6 V, so you can power the unit with either four 1.5-V batteries in series or a larger battery or power pack and a regulator. The four AA batteries shown in Figure 2-7 will give the circuit 6 V and run for a very long time because the unit uses very little power. Although most 74 series logic chips, like the 74HC4017, specify only 5 V, they usually will run with higher voltages, so 6 V is not a problem. For use with a higher-voltage battery, such as a 9-V battery, you certainly will need a regulator to reduce the voltage to avoid damaging the chip.

The simple 5-V regulator schematic shown in Figure 2-8 will allow you to connect a 9- or 12-V battery or a direct-current (dc) wall adapter with a voltage between 9 and 15 V to practically any project in this book requiring 5 V. Because all the circuits in this book use very little power, a heat sink will not be needed because the 7805 regulator will not be working very hard at all. R1 and D1 are optional, but they do let you know when your project is switched on. Having an ON-indicator LED is nice, especially if there is no way to tell if your circuit is on when you put it away.

Figure 2-7 *Six volts from four batteries.*

Figure 2-9 shows the simple 5-V regulator made on a small bit of perf board and ready for use with any of the 5-V projects presented in this book. The ON-indicator LED and current-limiting resistor are not shown on the board because they usually are placed somewhere on the cabinet front, away from the circuit board. If you have modified your circuit to include many more components, then your regulator may need a heat sink to dissipate the extra heat away from the case. If the regulator is so hot that you can't keep your finger on the small metal tab, then you should add some type of heat sink to help cool the device. Any small bit of steel or copper plate usually will be adequate to cool the regulator. If your entire circuit uses more than 1 A, then you probably will have to find a larger regulator than the 7805.

To make the reaction speedometer a little more interesting, I made an oval 10-digit speedometer using Photoshop and then printed it out so that I could glue it to the plastic box. This template also served as a guide when drilling the holes for the 10 LEDs. To drill a perfect hole, start with the smallest drill bit in your kit, and your hole will not wander as you use larger bits. As shown in Figure 2-10, the pilot hole (second drilled hole) is a lot smaller than the hole needed for the $\frac{3}{16}$-inch-diameter LEDs that I am using. It is good practice to drill a small pilot hole anytime you need to make an accurate drill hole in a cabinet. Having multiple holes line up is important when there are a few controls or lights in a row because the accuracy of your work will really show.

The completed reaction speedometer is shown in Figure 2-11, ready to help me train my reflexes to the cutting edge. I can set the speed dial to about half and still pass the test before the last LED lights, but any more than that is just way too

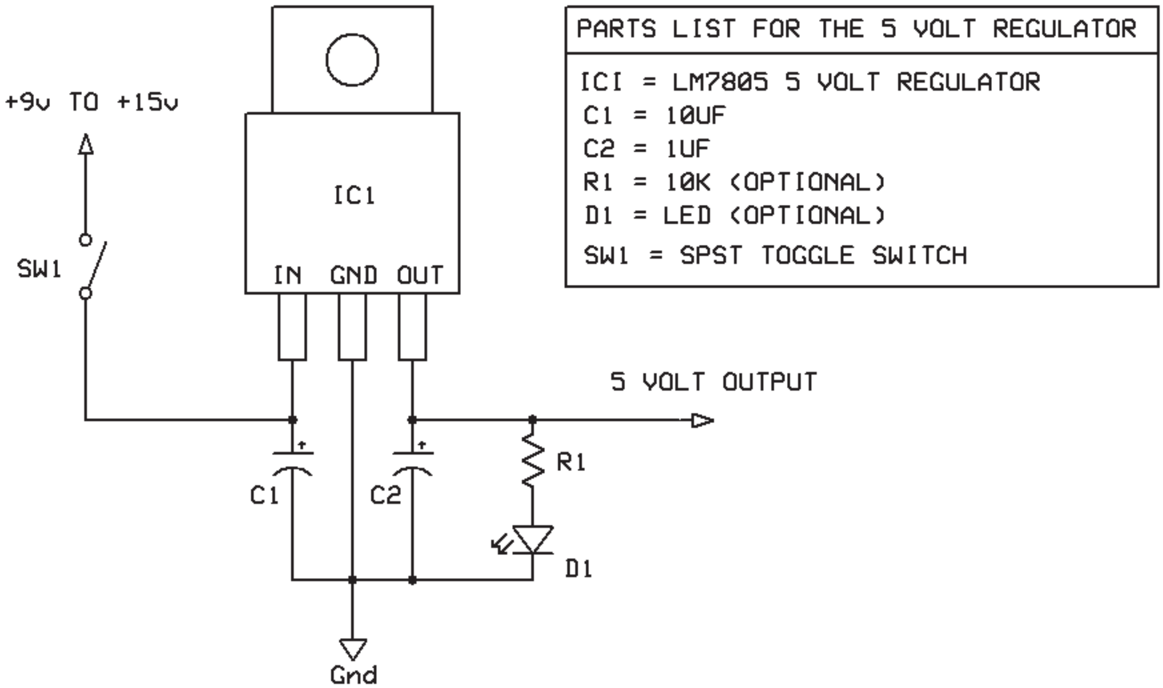


Figure 2-8 *Optional regulator schematic.*

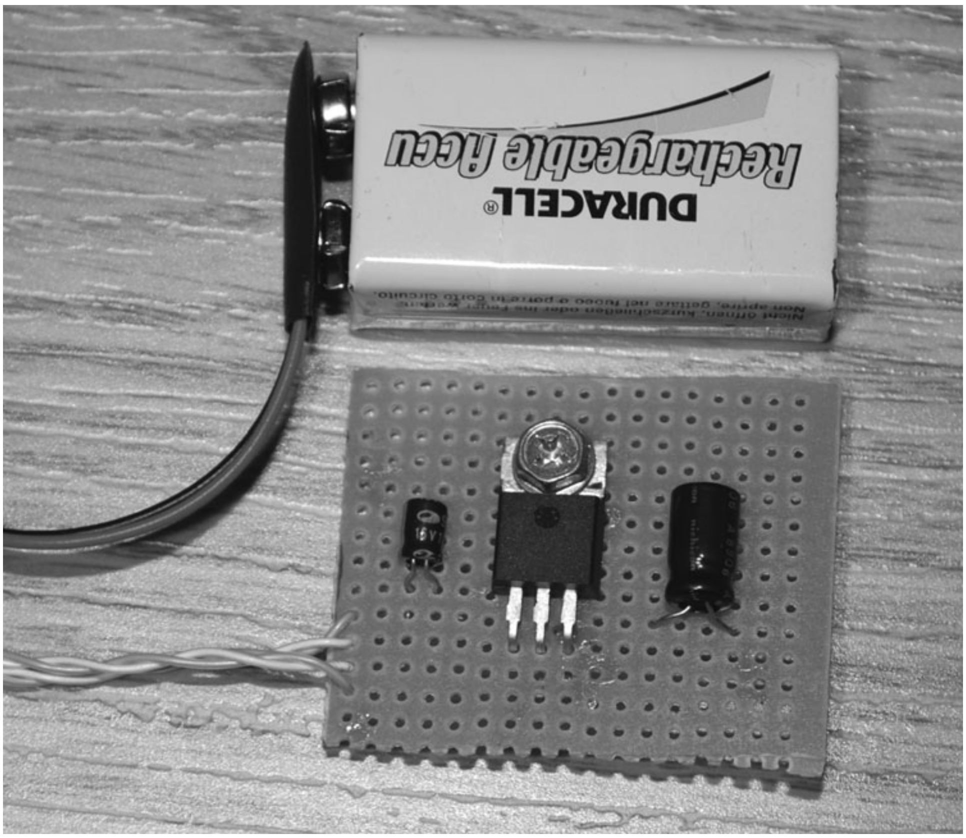


Figure 2-9 *Five-volt regulator breadboard.*

Figure 2-10 *Adding a graphic speedometer.*

Figure 2-11 *Reaction speedometer ready to use.*

fast. I have noticed that after drinking a caffeinated coffee, I can flip the switch between one and two LEDs quicker then when I am not buzzing from caffeine. When I am tired, I notice that my speed decreased by one or two LEDs. To get the most accurate results, I take the test 10 times, add the results, and then divide by 10 to get the real speed average. Another thing I have noticed is that when using only the sound as a trigger, I am faster than when using only the LEDs. I guess my ears are quicker than my eyes!

The reaction speedometer works very well, and there are a lot of interesting modifications you could make to enhance the design. Adding colored LEDs might make the test look more interesting. How about replacing the piezo buzzer with a headphone jack so that you can have your subject wear headphones? Maybe you could design a random trigger using a few more counters and a clock so that you can test yourself. How about a tactile test where the output from one of the LEDs is fed into a transistor or relay that switches a solenoid placed on the subject's body? I am sure you will find many ways to modify and experiment with this simple circuit in order to satisfy your own "evil genius" needs.

This page intentionally left blank

Project Three

Body Temperature Monitor

Body temperature varies throughout the day depending on our mental and physical state. Normally, our bodies remain at a temperature of 98.2°F (36.8°C) when we are awake and not involved in strenuous physical activity. Measuring body temperature during sleep experimentation is particularly useful because our bodies drop to their lowest normal temperatures when we reach the second half of our sleep cycle. This low-temperature point is called the *nadir* and will be about a degree cooler than the normal waking body temperature.

The device presented here is an example of how an inexpensive digital thermometer IC can be connected to a microcontroller to monitor body temperature. Although this project is very simple and displays only a two-digit temperature value, it would be very easy to modify and expand this example to log data, display decimal values, or interface with a computer.

The Maxim DS1621 is just one of many examples of inexpensive and easy-to-use temperature sensors that can be connected to a microcontroller with very minimal effort. Depending on the device, temperature sensors can output data in many different ways. Some examples include serial data, parallel bytes, analog data, and pulse-width modulation. The DS1621 is a two-wire serial interface, so it is well suited to microcontrollers with a low pin count and can be controlled easily by a basic program. As you can see in the body temperature monitor schematic (Figure 3-1), there is not much to it besides the DS1621 thermometer (IC1), an 8-bit Atmega88 microcontroller (IC2), and a pair of seven-segment LED displays. The DS1621 sends its data to the microcontroller every few seconds, and then the data are converted to the nearest degree and displayed on the two LED displays. You could easily modify the code to display the decimal values as well or even change the display to Fahrenheit, although that would require three LED displays if you want to see readings over 99°F. I wanted to keep this example as simple as possible because there are so many different ways you could modify this project.

To keep the number of input/output (IO) pins to a minimum, the two transistors (Q1 and Q2) switch between the dual displays at such a fast rate that they all seem to be on at the same time. This time-sharing trick is how most LED displays work, where there are hundreds or thousands of LEDs to control and only a limited number of connecting wires. There really is no limit to how many seven-segment LEDs you can connect as long as you have sufficient drive current and microprocessor speed and can spare the extra IO pin for each common connection. The

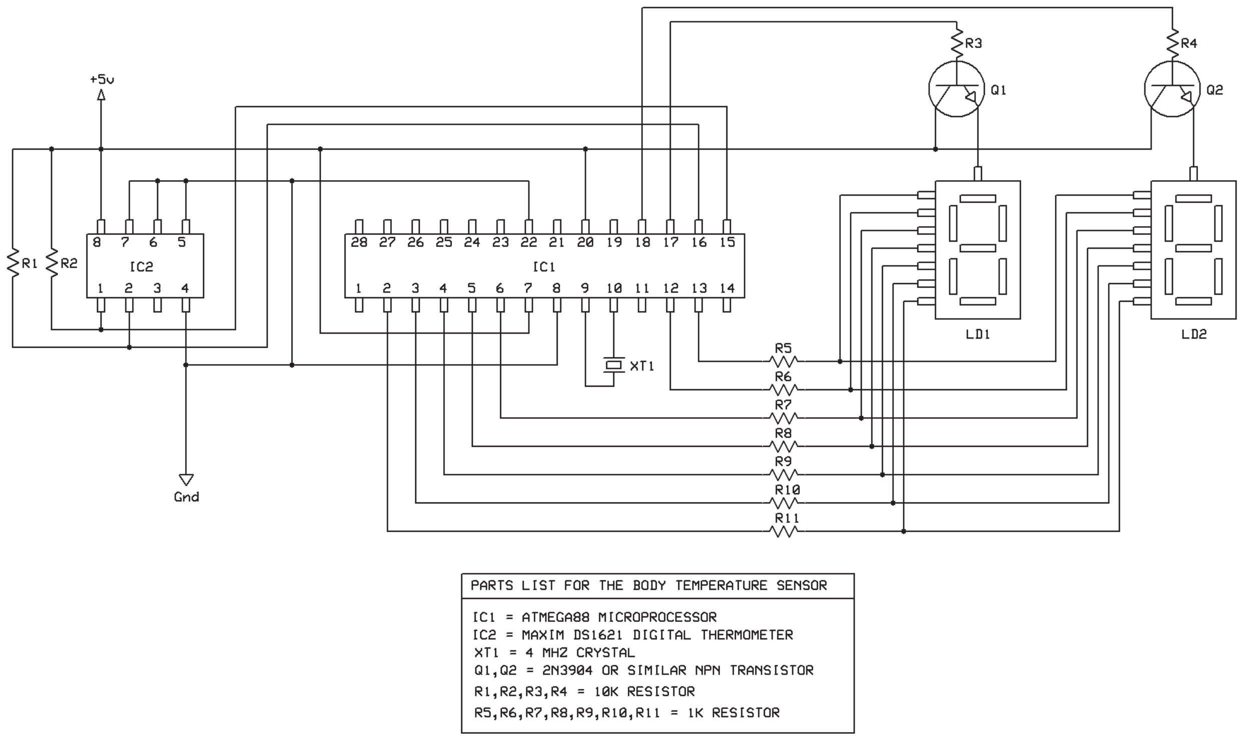


Figure 3-1 *Body temperature monitor schematic.*

seven-segment LED is a very common and inexpensive display that you are already familiar with because it is used in everything from your digital clock to the panel of your microwave oven. There are actually eight segments if you include the decimal point, but we don't use it in this application. These displays come as either stand-alone blocks or chained blocks containing more than a single digit. Some LED displays even have alphanumeric capabilities or multiple "dots" so that they can make any character imaginable. LED displays are either *common cathode* or *common anode*, which means that either the positive connections or the negative connections all go to a common point. It really does not matter which type you use as long as you install them in your circuit so that current is flowing in the proper direction. To use a common-cathode LED display in this circuit, you would have to tie the driver transistors/emitters to ground and connect the common cathode to the collector instead.

Once you have the code compiled and installed in the Atmega88, powering up the breadboard will instantly show you the temperature in your room. Figure 3-2 shows the completed circuit reading 21°C, which is only 1 degree different from what the hallway thermometer was claiming. When I placed my finger on the DS1621, the temperature slowly climbed to 32°C, which seemed about right because I had cold hands, and the temperature outside the body is usually a few degrees colder than inside. For dream research, the actual temperature is not what counts, but rather the variance of temperature over the entire night. Notice the addition of the 7805 regulator on the top right of the breadboard so that the system will run on a 9-V battery.

If you build the circuit compact enough, you may find that it will fit into a cabinet small enough to be placed directly on the body using some kind of elastic strap. I decided to place the electronics in a cabinet but run the temperature

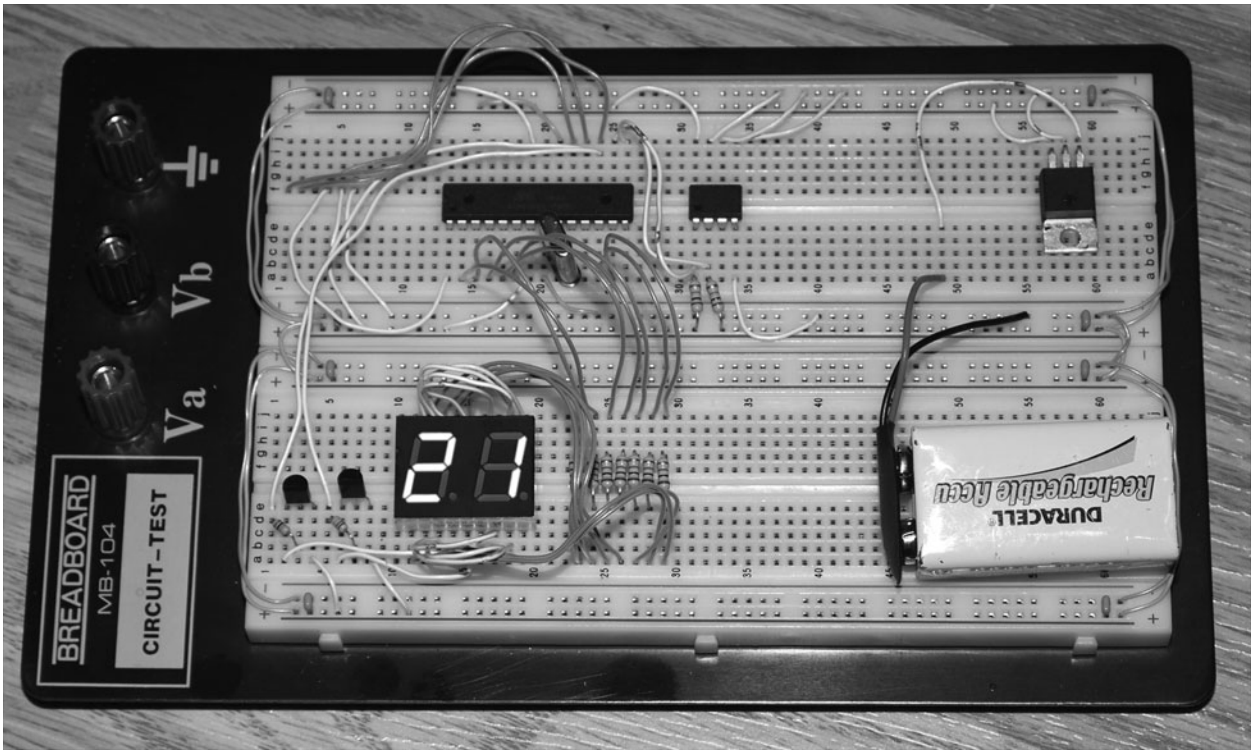


Figure 3-2 *Testing the circuit on a solderless breadboard.*

sensor externally so that it would be more comfortable as a sleep experimentation device. I also wanted a little extra room inside the housing to add some circuitry later that allows a wireless connection through a simple rf transmitter module. Figure 3-3 shows the perf board that carries all the components except for the DS1621 temperature sensor. There will be limits as to how far your temperature sensor can be away from the microcontroller owing to noise and impedance of the wiring, but a few feet should be no problem at all for most devices.

The temperature sensor lives on a tiny bit of perf board with the power and signal wires coming into it (Figure 3-4). I use the top part of a sock to hold the sensor against the subject's arm, where the skin gives good contact with the top of the sensor package. You also might want to run a fine bead of hot glue or nonconductive caulking along the pins of the IC package so that perspiration does not create resistance between the pins of the device if they come in contact with the skin.

The completed body temperature monitor is shown in Figure 3-5, again reading the temperature in my laboratory, which was getting a bit too warm owing to the huge lights I used for making these photographs. Operation of the device could not be more simple; just turn it on and read the temperature, which changes about once per second. I later added one of those microprocessor-compatible rf transmitters to the unit so that I could read the data from the microcontroller into a computer in order to graph the results during the night. Many electronics distributors carry inexpensive rf solutions, which are easy to use and can send serial or parallel data to or from any microcontroller.

The sensor package must come into good contact with the skin to give a reliable reading, and a cut-up sock makes a good armband that keeps the sensor motionlessness and insulates the

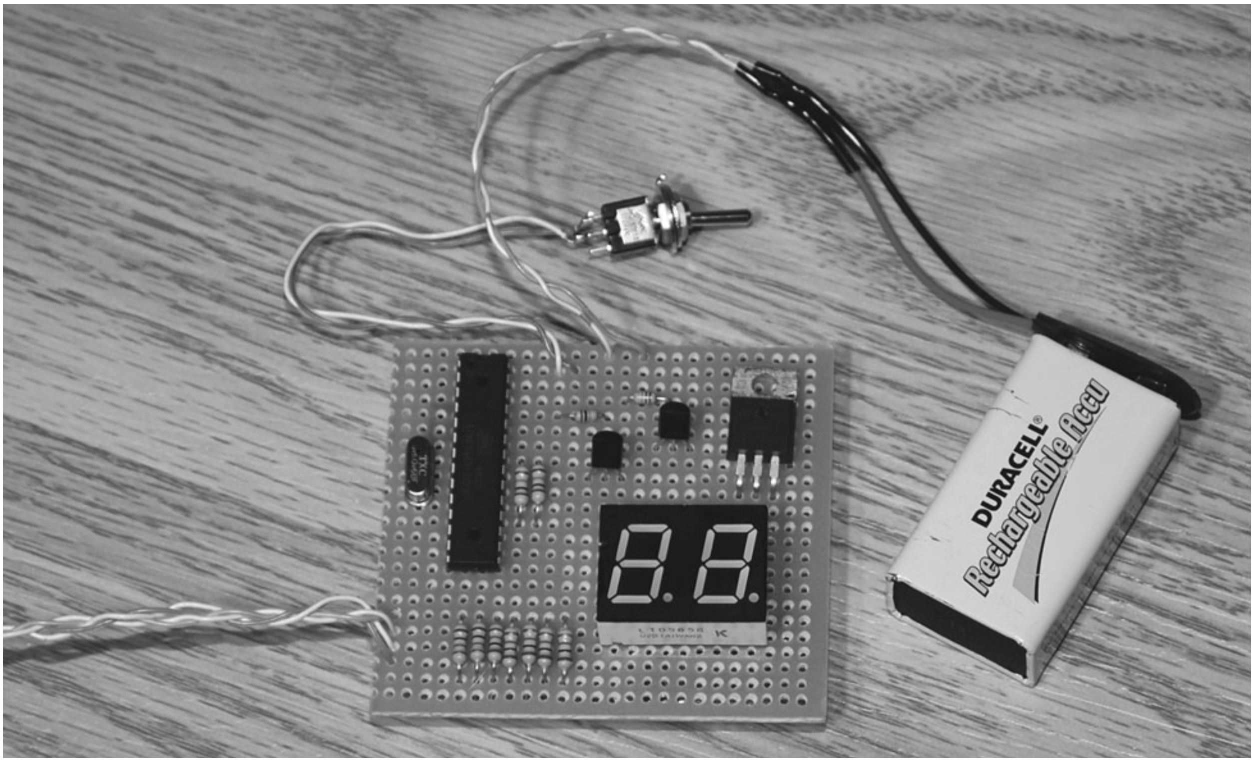


Figure 3-3 *The completed temperature monitor on some perf board.*

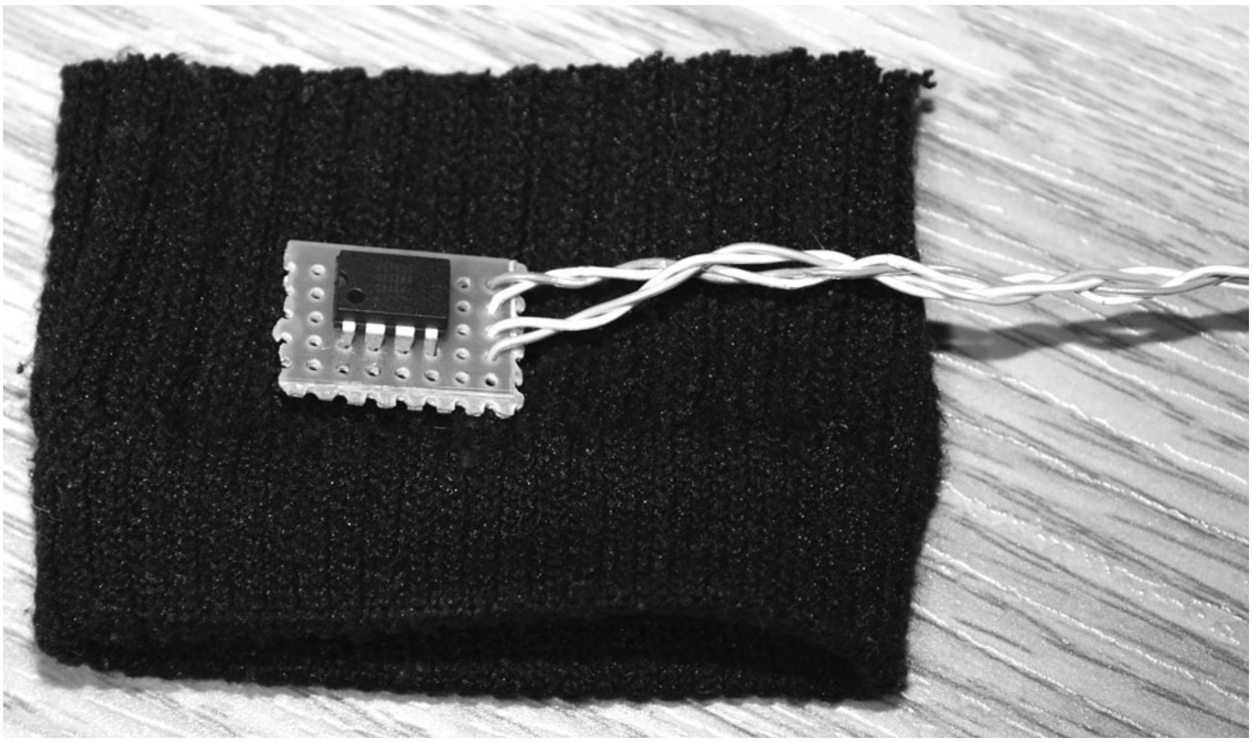


Figure 3-4 *The remotely located temperature sensor.*

Figure 3-5 *The completed body temperature monitor taking a body temperature reading.*

sensor from the outside world. Different parts of the body will give different temperature readings, so don't be alarmed if your sensor shows that you have a very low body-core temperature. The surrounding hardware also will affect the readings, so you might want to consider adding a line to the source code that compensates for the –3 to –5°C difference this device will have compared with a thermometer stuck under your tongue. If your main goal is reading changes in temperature, then it really makes no difference what the sensor shows so long as it varies along with changes in your subject's body temperature.

The complete source code for the body temperature monitor is shown in Listing 3-1 of the appendix and was written in Bascom AVR to keep it as simple as possible. Basic is a great language for fast prototyping and can be ported easily to any platform in a hurry because it is extremely readable. I will explain what each block of code does so that you can understand the workings of the program and port it to whatever microprocessor you plan to use.

[redacted]

The code following this comment is required to tell Bascom that we are going to target the Atmega88 device and that our clock will be an external crystal resonator running at 4 MHz. Telling the compiler your clock speed becomes important when using commands that deal with timing-sensitive routines such as serial transmission or analog-to-digital readings. Defining the device also helps the compiler to generate user errors that have to do with IO pins. In this way, you can't accidentally try to toggle an IO pin that does not exist on the actual device. Critical timing is not an issue in this program, so you can use whatever crystal you happen to have in your parts box.

This block of code sets up the pins that will connect to the LED display (outputs) and to the two-wire serial interface to the DS1621. “Scl” and “Sda” are special Bascom reserved keywords that specify the serial data and serial clock lines.

Basic uses *variables*, which are letters or words used to hold values. I like to use single letters such as *A*, *B*, and *C* for simple programs such as this one, but when you are working on a large, complex program, use of more descriptive variable names is recommended. “TIMER2” or “REDLED1,” for example, would be descriptive variable names that make a lot more sense in a huge block of code. The variables “Msb” and “Lsb” will store the 2 bytes that are returned from the DS1621. “Msb” is the most significant byte and will contain the whole-number value of the current temperature, whereas “Lsb” is the least significant byte containing only the decimal value. “Lsb” is not being used in this version of the code.

This block of code sets up an array of 10 values for the variable “LED.” Although this may seem confusing at first, the values correspond to which of the seven segments will be lighted to display a particular number. To make this a little more confusing, the value in parentheses is actually one higher than the represented numerical value, and to light a segment, we want a low bit, not a high bit, so the value of “Led(9) = 0” says that to display the decimal number 8, we want all bits to be off. This means that all segments light up, and an “8” will be displayed. It can be a bit of a chore computing these values by adding port bits, but once you have done it once for your segment, the hard work is completed.

Everything from here on is going to happen continually until the word "Loop" is reached, which causes program execution to start again where it first encountered the word "Do." This is called an *endless* loop because it never stops unless forced to by another command or an error.

This code block sends the initialization sequence to the DS1621 in two-wire serial format using the built-in "I2C" commands. Having ready-to-use "I2C" basic commands takes a lot of work out of your hands and makes rapid prototyping a snap. The values presented here are only for the DS1621 and are based on the datasheet for the device as well as several code examples found on various forums. The whole-number value is stored in the variable "Msb," and the decimal value is stored in "Lsb" (which is not used). If you are using some other digital thermometer IC, then most likely it will use a different set of commands for communication.

This small bit of code converts the value stored in the variable "Msb" into a pair of bytes ("A" and "B") containing the two digits that make up the value. This is necessary because each LED display can display only the values from 0 to 9. Once variables "A" and "B" are set, the "Ledshow" routine is called.

The "Loop" command causes program execution to jump back to the main routine where the "Do" command was encountered. This is the end of the infinite loop, and all other commands beyond here must be called by either "Goto" or "Gosub" commands.

This is the routine that displays a digit on each of the two LED displays. A little trick called *persistence of vision* is used here to switch between displays so fast that your eyes think they are all on at the same time. As you can see, only one bit of the two ”Portb” pins is on at a single time, and then the variables “A” and “B” are sent to the display for only 2 ms each. Because 2 ms is so fast, it appears that each display is on all the time, and the seven-segment lines can be shared, saving valuable IO overhead. Since Bascom array variables start at 1 not 0, the line “C = A + 1” adds 1 to the array pointer so that the value in “Led(c)” is the same as the decimal value we want. This conversion just makes it easier to understand the code, especially when trying to compute the segment bits from scratch the first time. Once the displays have been lighted for 2 ms each, this routine just “returns” to where it was originally called.

That’s all there is to it! A lot can be accomplished in very few lines using Basic, and since microcontrollers work at nanosecond speeds, you have a lot of power at your fingertips.

Since this is a very plain and simple temperature monitor, you likely will want to modify the code to expand and add your own features, such as a decimal readout and maybe a serial or USB interface to send the data to a computer logging program. Simply by expanding the number of LED displays to four, you could send the type bytes as a more accurate decimal value. Adding a wireless link to send data to a logging program for sleep research or dream-state detection is another interesting modification I plan to do later to this simple device. Using body temperature in conjunction with some of the other sleep-research tools really can increase your ability to monitor and predict your sleep and dream cycle through the night. Next, we will build a simple device to measure respiration.

Project Four

Respiratory Monitor

Respiratory rate varies greatly depending on how much oxygen the body requires. The average person will take 10 to 20 breaths per minute while at rest and between 30 and 40 breaths per minute during strenuous physical activity. During sleep, breathing rate also increases, so adding a respiratory monitor to your arsenal of sleep-research tools would be handy. This project presents a novel method of monitoring a subject's breathing rate that uses the noise picked up by a sensitive microphone and preamplifier to feed a recording device. The resulting waveform is very easy to analyze because it will contain visible bursts of data each time the subject inhales or exhales. Any computer program capable of displaying a waveform along with the time can be used to determine breaths per minute or breaths per hour.

The respiratory monitor is actually an extremely sensitive audio preamplifier that exploits the fact that moving air close to a microphone causes an overload in the audio output. It is this high-gain noise that creates the visible bursts of data that can be seen on the computer screen while viewing the waveform. Exhaling creates the largest spike in data, so you can easily determine how many breaths per minute the data contain by either counting the larger spikes or counting all data spikes and then dividing them by two. Figure 4-1 shows the ultra-high-gain preamplifier, which is fed by an electret microphone. An *electret microphone* is a tiny metal can that contains not only a microphone inside but also a sensitive transistorized amplifier so that the output is already amplified somewhat before it reaches the LM358 op amp (IC1).

An electret microphone is the most common type of microphone you will find in audio recording devices and most computer microphones. Even those large plastic multimedia microphones you can purchase for your computer may contain nothing more than a pair of tiny electret microphones inside, along with a chunk of metal to make the device feel heavy (seriously). Answering machines, telephones, and most other small recording devices all will contain an electret microphone, and you also can purchase them new at most electronics suppliers for a few dollars each. Figure 4-2 shows a few of the many electret microphones that I have collected by ripping apart old electronic devices over the years. Some have a rubber cap or may be sealed in plastic, but inside, you will find the same basic tiny metal can with a hole at one end and a pair of leads or solder spots at the other end. All you need to know is which lead is positive and which is negative, but this is very easy to determine.

PARTS LIST FOR THE RESPIRATORY MONITOR
ICI = LM358 DUAL OP AMP R1,R3,R4 = 10K R2 = 1K C1 = .1 UF CERAMIC C2 = 4.7 UF ELECTROLYTIC VR1 = 500K VARIABLE RESISTOR

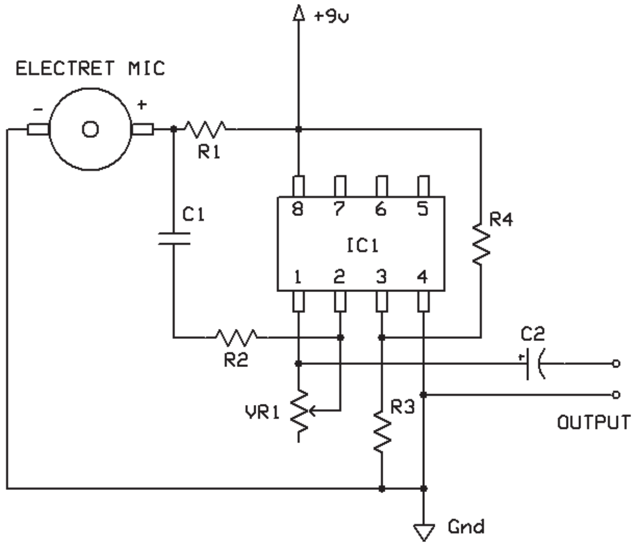


Figure 4-1 *The respiratory monitor schematic.*

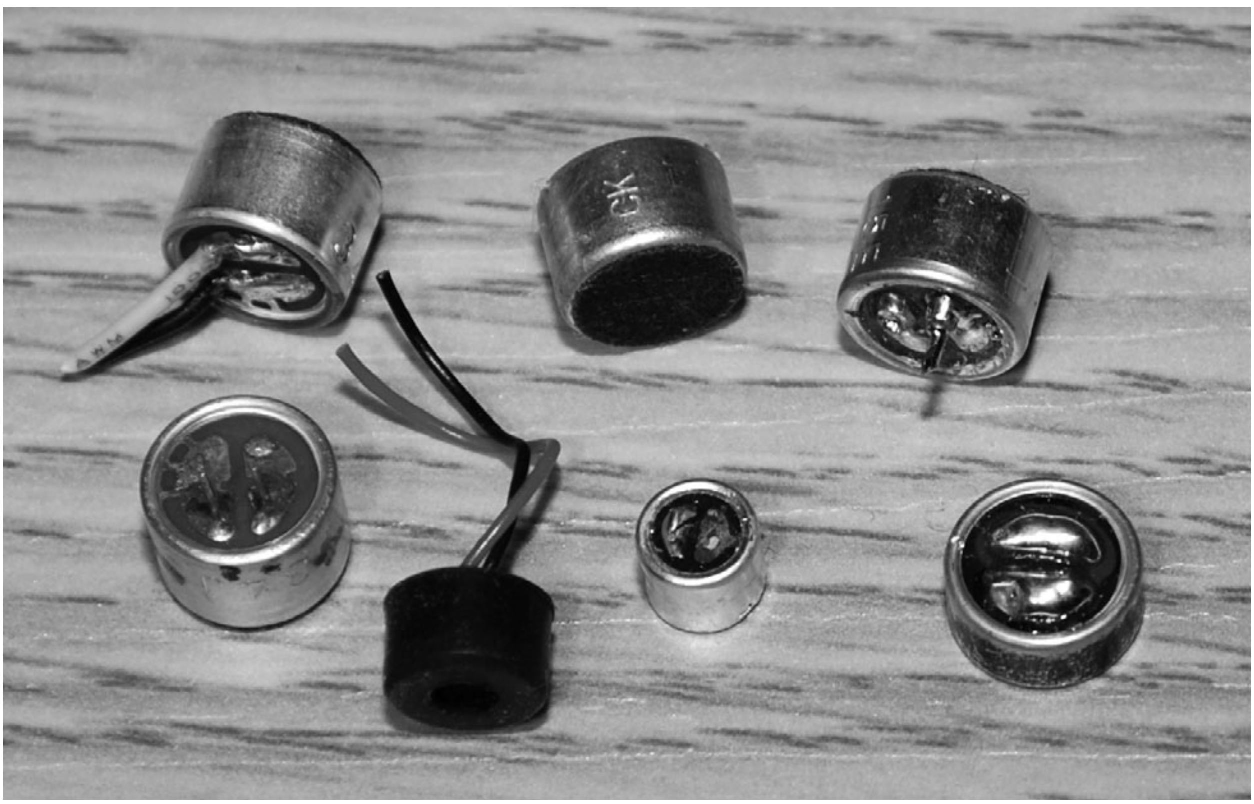


Figure 4-2 *Several electret microphones.*

Figure 4-3 *Electret microphones are polarized.*

An electret microphone requires power to run the small amplifier contained within the tiny metal can. Although the power requirement is very small, you still have to figure out which lead is positive and which one is ground so that you can insert the microphone into your circuit correctly. Looking at the underside of the microphone (Figure 4-3), you usually will see that one lead or solder spot is also connected to the metal can by a small trace at the edge. The side that connects to the can almost always will be the negative, or the ground, lead. The good news is that if there is no visual indication of polarity on your microphone, you simply can try it both ways in the circuit without damaging the microphone in any way. If you have the polarity reversed, you simply will get no output from the microphone.

Since the goal is to position the microphone as close to your subject's nostrils as possible, an inexpensive multimedia headset such as the one shown in Figure 4-4 would be perfect for this project because under that block of foam is just another electret microphone. By using a premade headset microphone, you don't have to take the housing apart because you can simply add a $\frac{1}{8}$ audio jack to connect the microphone to the input of your preamplifier. You also can make a simple

Figure 4-4 *A headset containing an electret microphone.*

headset by bending a coat hanger or simply use a bit of double-sided tape to hold the tiny electret microphone just under the subject's nose to get a good reading.

The breadboarded preamplifier circuit is shown in Figure 4-5 and is so simple that it can be built onto a very small circuit board or without any circuit board at all. The circuit will run on as low as 3 V and as much as 12 V, but a 9-V battery seemed most convenient and would power the preamplifier for a very long time. This preamplifier also makes a decent high-gain microphone amplifier for voice recording, so it can be used for many other experiments needing a sensitive microphone and preamplifier. You also can reduce the circuit further by replacing the variable resistor (VR1) with a fixed resistor with a value between 1 MW and 500K. The 1-MW resistor will give the amplifier the most gain possible, but it may be too much if you do plan to use the device to record voice or audio.

The preamplifier circuit had so few components that I decided to build it without a circuit board to save space and make the compact unit shown in Figure 4-6. I replaced the variable resistor with a 1-MW fixed resistor so that the amplifier would have full gain at all times. Because the LM358 IC was the largest part, it was used as a base to hold the other components into a "blob" circuit that was about half the size of a penny. I also could power the amplifier from a 3-V button cell and place the entire setup inside a bottletop so that only the output leads needed to come from the device. The completed preamplifier performed so well that I even used it for telecommunications over the Internet because it had better gain and clarity than the microphone connected to my laptop. Now all you have to do is find a way to log and analyze the data once the microphone is ready to record your subject's respiration.

The burst of audio data can be seen clearly in Figure 4-7 after recording a few hours of breathing using audio recording software. I use an older version of Sony Sound Forge, but just about any software-based audio recording software that allows you to see waveform data and time will do the job. The larger bursts of data

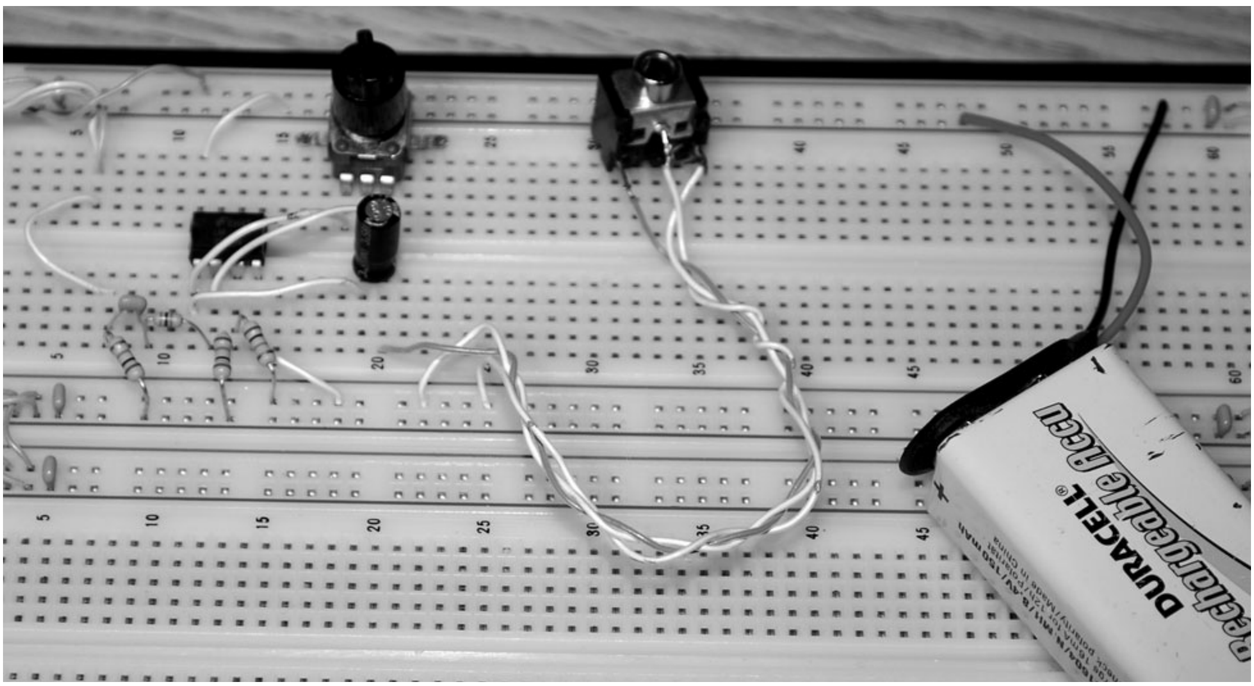


Figure 4-5 *Breadboarded respiratory monitor circuit.*

Figure 4-6 *Building a circuit without a circuit board.*

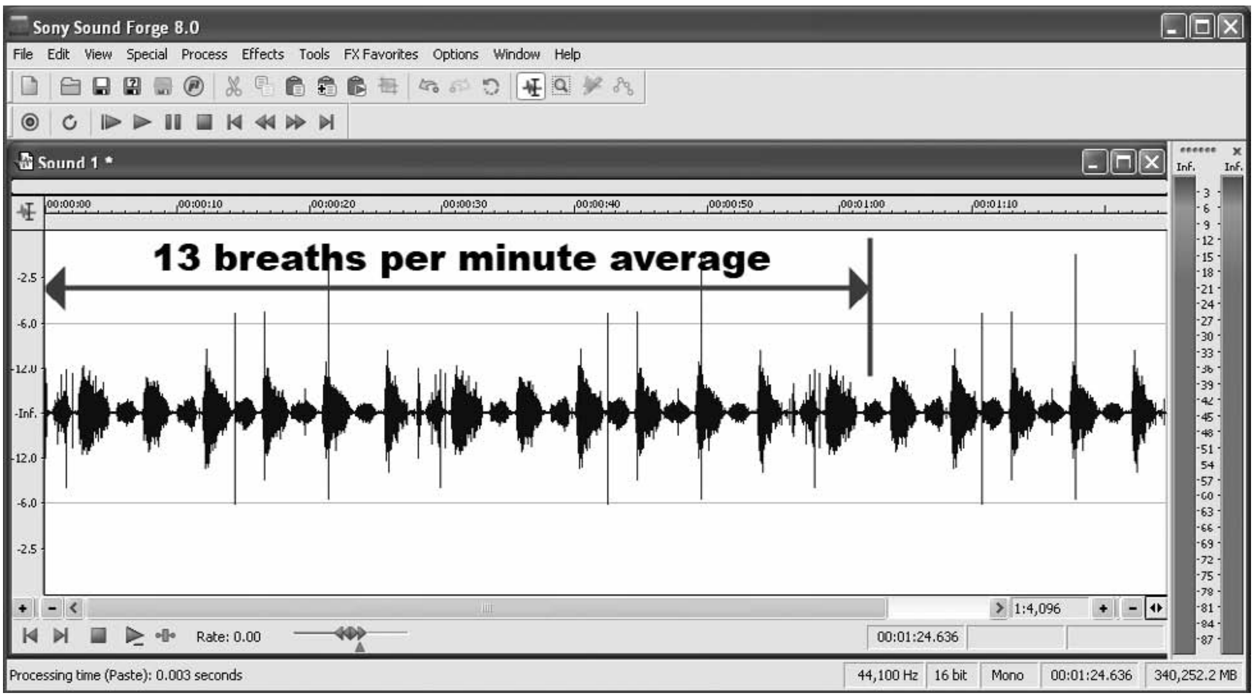


Figure 4-7 *Data bursts shown from the device.*

are exhalations, and the smaller ones are inhalations, so you can just count the bursts for 1 minute and then divide that value by two to get the average respiratory rate. If you are logging several hours through a sleep cycle, then you will easily see the changes to respiratory rate that occur during dream cycles. Breathing becomes shallow and erratic while we dream, so adding respiratory data to your dream laboratory certainly will help you to log your subject's cycles and trigger any external devices you are using. The audio-recording software is the most basic method of using the preamplifier to monitor respiration, and many other programs are available that can perform much more advanced analysis of these simple data.

You also could feed the audio output into an analog-to-digital converter built into a microcontroller and log the respiratory data directly as digital data. A comparator could be set up to read only the exhalation waveform because it is two or three times greater than the inhalation waveform. The circuit and source code presented in the light-sensing lucid-dream mask project in Section Two also could be modified easily to take its input from this circuit if you want to monitor data without the need of a computer. Add a few LED displays, and you could add a counter that simply counts breaths for as long as the circuit is running. Next, we will build a device to monitor heart rate.

Project Five

Heart Rate Monitor

Of all the body responses one could monitor, heart rate is one of the most important because it fluctuates greatly depending on our state of mind and physical condition. Heart rate during physical activity gives us a direct indication of our fitness level, and heart rate changes throughout the night can be clear indicators of when a dream cycle has begun. A heart rate monitor is also a very easy device to use for extended periods because it can be placed on the body in such a way that is not uncomfortable or in the way of any normal activity.

Our heart rate varies with age, gender, and physical condition, but the normally accepted range for adults is between 50 and about 100 beats per minute. While we rest, our heart rate is usually between 50 and 75 beats per minute, and it could climb to over 200 beats per minute while working the body to its near-maximum effort. Heart rate also becomes erratic during sleep cycles, so it can be used to trigger some type of dream experiment or be used in conjunction with many of the other dream-research devices presented in this book to track the dream cycles more accurately throughout the night.

The heart rate monitor project presented here uses a sensitive light-detecting resistor to detect small changes in the light beaming through your finger as your blood pumps through the tiny arteries in the finger. This device is not the same as the heart rate monitors typically used in hospitals because those use skin probes that detect changes in electrical activity as the heart beats. This heart rate monitor actually "sees" your pulse through the body, so it can be used on a finger, toe, or even your earlobe.

This project is a bit more involved than some of the other devices presented in this book, but it still can be built on a small breadboard with a few inexpensive components in a day or two. The completed heart rate monitor is very stable and as accurate as any you would find built into high-quality exercise equipment. Your heart rate will be displayed as a three-digit number on the LED readout, which updates its average once every few seconds. There is much room for improvement and modification owing to the simplicity of the microcontroller code, so you can easily adapt this device to just about any hardware or data-logging device.

The heart rate monitor is a combination of analog and digital circuitry, as shown in Figure 5-1. The LM324 quad op amp (IC1) forms a sensitive amplifier and a low-pass filter that will "lock" onto tiny variations in voltage that fall within the typical heart rate frequencies. The varying voltage comes from the light-dependent resistor (LDR), which changes a very small

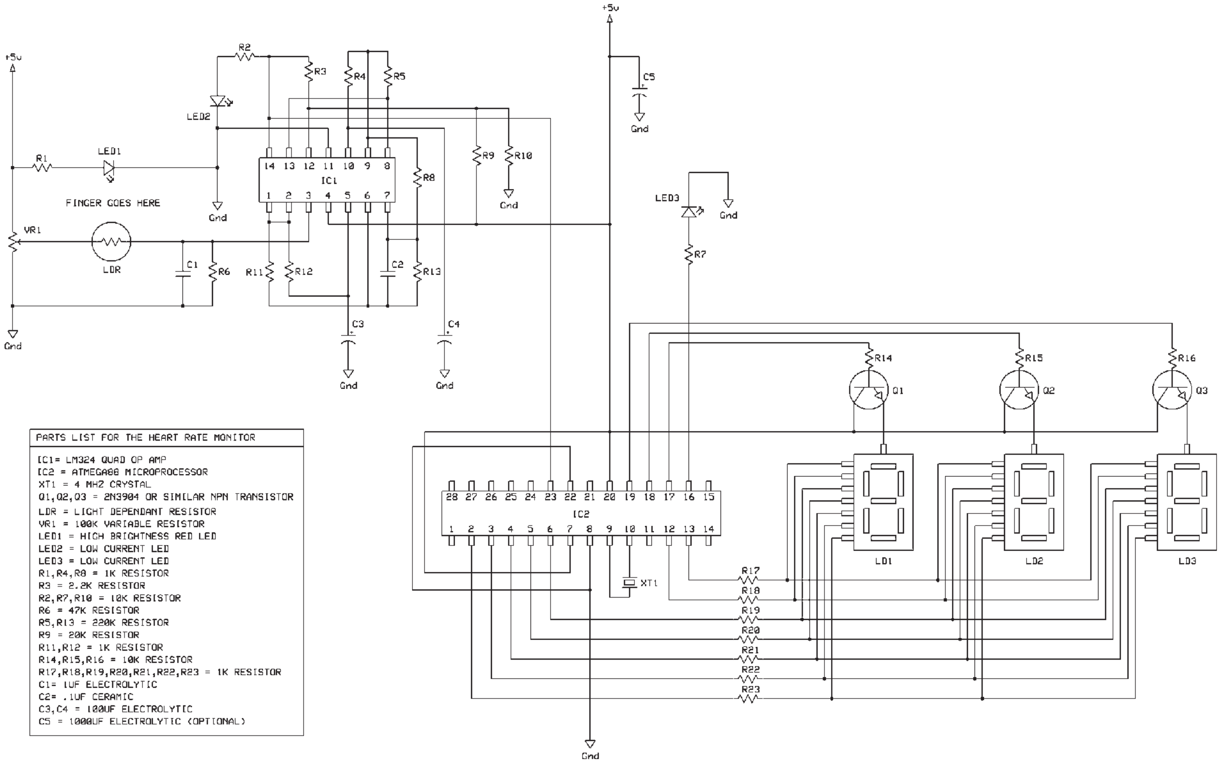


Figure 5-1 *The heart rate monitor schematic.*

amount each time blood pumps through the tiny arteries in between the visible red LED and the LDR surface. After the signal is conditioned by the low-pass filter, it is fed out of the op amp into the analog input of an Atmega88 microcontroller, where the program counts the beats and keeps a running average of beats per minute. The resulting heart rate is displayed as a three-digit number on the triple seven-segment LED displays. Because the microcontroller was added to this device after it was designed, you could decide to leave out the digital part of this project and take the output directly from the op amp and feed it either to an LED that will blink with each heartbeat or to some other data logger or device of your own making. The output from the op amp is very close to being TTL (transistor, transistor logic) compliant, so it could be adapted easily to drive most digital equipment requiring a 5-V TTL signal input. I recommend that you build the analog part of this circuit first because it is the most complex part of the circuit. LED2 will blink each time a heartbeat is detected, so this makes debugging the system much easier. When you get to the digital part of the circuit, another debugging LED (LED3) will blink each time the microcontroller receives a pulse signal from the analog section of this project.

The light-dependent resistor (LDR) is a commonly used semiconductor that can be found in practically every device that needs to respond to some change in ambient light levels. Street lights, security lights, light meters, and even dollar-store night lights such as the one shown in Figure 5-2 will contain an LDR. You also can purchase an LDR from most electronics suppliers, but the dollar-store night light is more convenient, costs about the same as a bare LDR, and gives you a few other bits for your junk box, such as a triac and a few resistors. An LDR is easy to identify because it will be visible to the light it must sense and will look like a tiny button

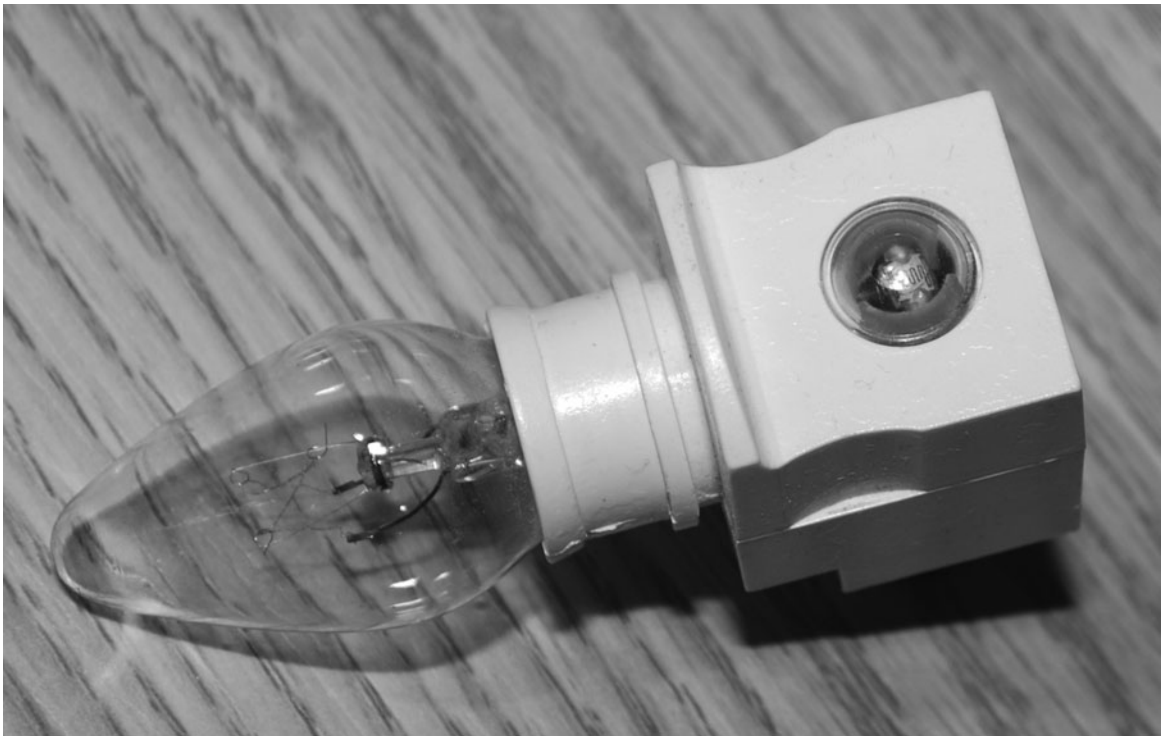


Figure 5-2 *A good source for a light-dependent resistor (LDR).*

with a snakelike pattern on its surface. This can be seen clearly under the tiny plastic eye in the figure.

The snaking track along the surface of the LDR shown in Figure 5-3 is connected to the two pins that exit the device. As light strikes the surface of the LDR, the resistance drops greatly, acting like an analog light switch of sorts. Notice how

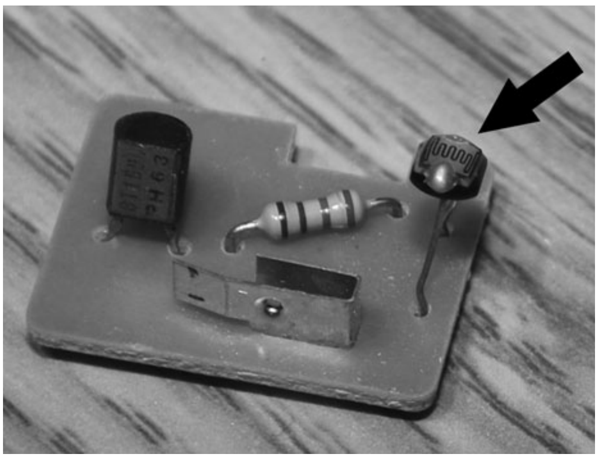


Figure 5-3 *The LDR is easy to identify and extract.*

amazingly simply the circuit ripped from the night light really is—just an LDR feeding a triac through a resistor to turn on the 120-V ac light bulb. To remove the LDR, either use a soldering iron to release it from the circuit, or just bend the leads back and forth until the device breaks free.

The breadboarded heart rate monitor is shown in Figure 5-4, and as you can see, most of the real estate has been taken up by the triple-digit LED display and all its supporting hardware. The analog pulse-detection circuitry at the top of the breadboard consists of the LM324 op amp and a few resistors and capacitors. The large 1000-μF capacitor (C5) shown under the 9-V battery is necessary only if you are using a dc poser adapter or find that the pulse-detection circuit falsely triggers when the LEDs change. The analog circuit is so sensitive that the change in voltage from a single LED coming on actually can trigger the circuit into an oscillation if your power supply is not rock solid. The huge capacitor acts as a power-supply buffer to help filter out any tiny fluctuations caused by the LED display

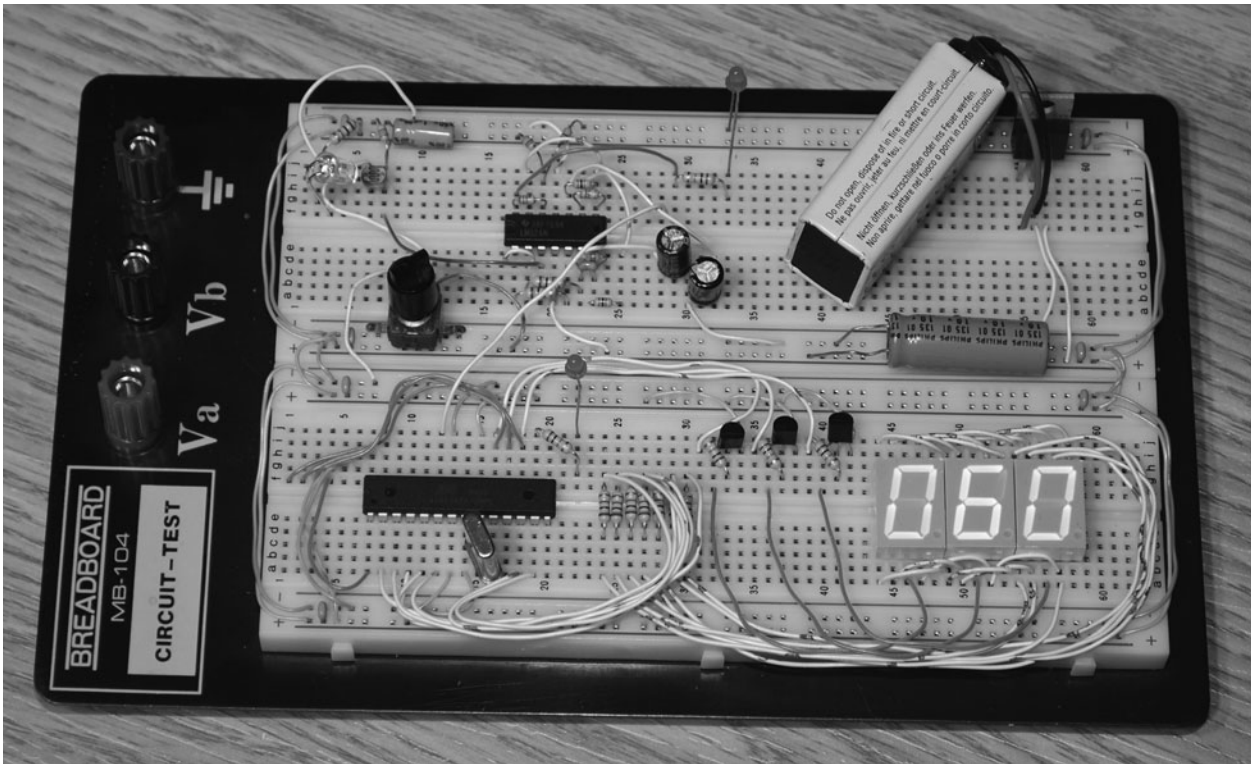


Figure 5-4 *The breadboarded heart rate monitor.*

circuitry. I found that the circuit worked fine without the large capacitor as long as it was running from a battery pack. Using a dc adapter did require the large filter capacitor to be included.

The component to the right of the 9-V battery is an LM7805 regulator to bring the supply voltage down to 5 V, as needed by the microcontroller. If you are building only the analog part of this circuit, it actually can be run from a supply voltage as high as 12 V directly. When first powered up, the LED display will read "060" and not change until a pulse is detected. Every 10 seconds, the average is recalculated, and then the readout is ramped up or down by a value of one on each following pulse. This slow ramp helps to filter out glitches in the average owing to movements of the body that might be detected as pulses. There is a lot of room to improve the microcontroller code because it was made to be short and simple to keep it printable.

To test the analog circuit, place a high-brightness red LED over the LDR, as shown in Figure 5-5, so that there is just enough room to place your finger in between the light and the LDR. You will have to keep your hand extremely still and at the same time take your own pulse using your other hand on your neck. When the

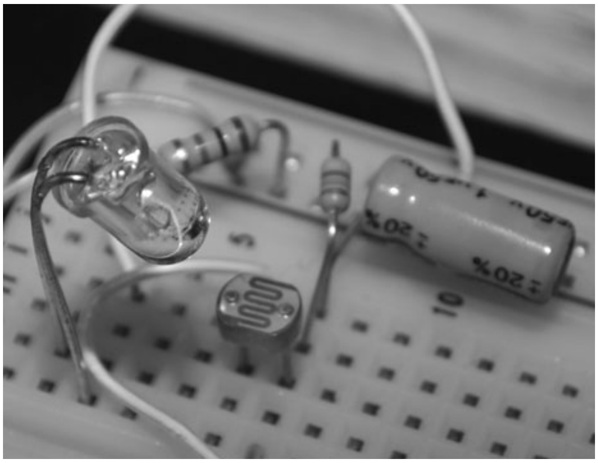


Figure 5-5 *Insert finger here.*

analog circuit is functioning properly, the pulse indicator LED (LED2) will flash almost at the same time that you feel the pulse in your neck.

If you just can't hold your hand still enough over the breadboard, move ahead and build the finger clip, which will eliminate false readings from accidental movement. You also will have to adjust the variable resistor (VR1) to some setting near its center of rotation to get the strongest reading on the pulse-indicator LED. The variable resistor biases the LDR in either the positive or negative direction, so you do not have to be overly concerned about the type of visible red LED you choose. The LED does have to be fairly bright and needs to be red because that is the color that passes through our skin the easiest. White light also will work, but green and blue LEDs will not work very well at all in this system.

I found that the system worked perfectly on the breadboard as long as I could keep my hand still enough not to send false readings to the analog circuit. If I rested the side of my hand on the desk and placed my thumb between the red LED and the LDR, then I rarely got a false reading. Figure 5-6 shows the result of 60 seconds of testing, getting a heart rate of 72 beats per minute, which jives perfectly with what I counted while taking the pulse on my neck. At this point, the analog circuit is fairly quirky when it comes to hand movement or even ambient light changes in the room, so some type of finger clasp will be needed to make the system much more stable.

The hair clip shown in Figure 5-7 just happened to fit around my finger or thumb perfectly, and it was not so tight that it became uncomfortable. There are hundreds of devices you could invent to attach the visible red LED

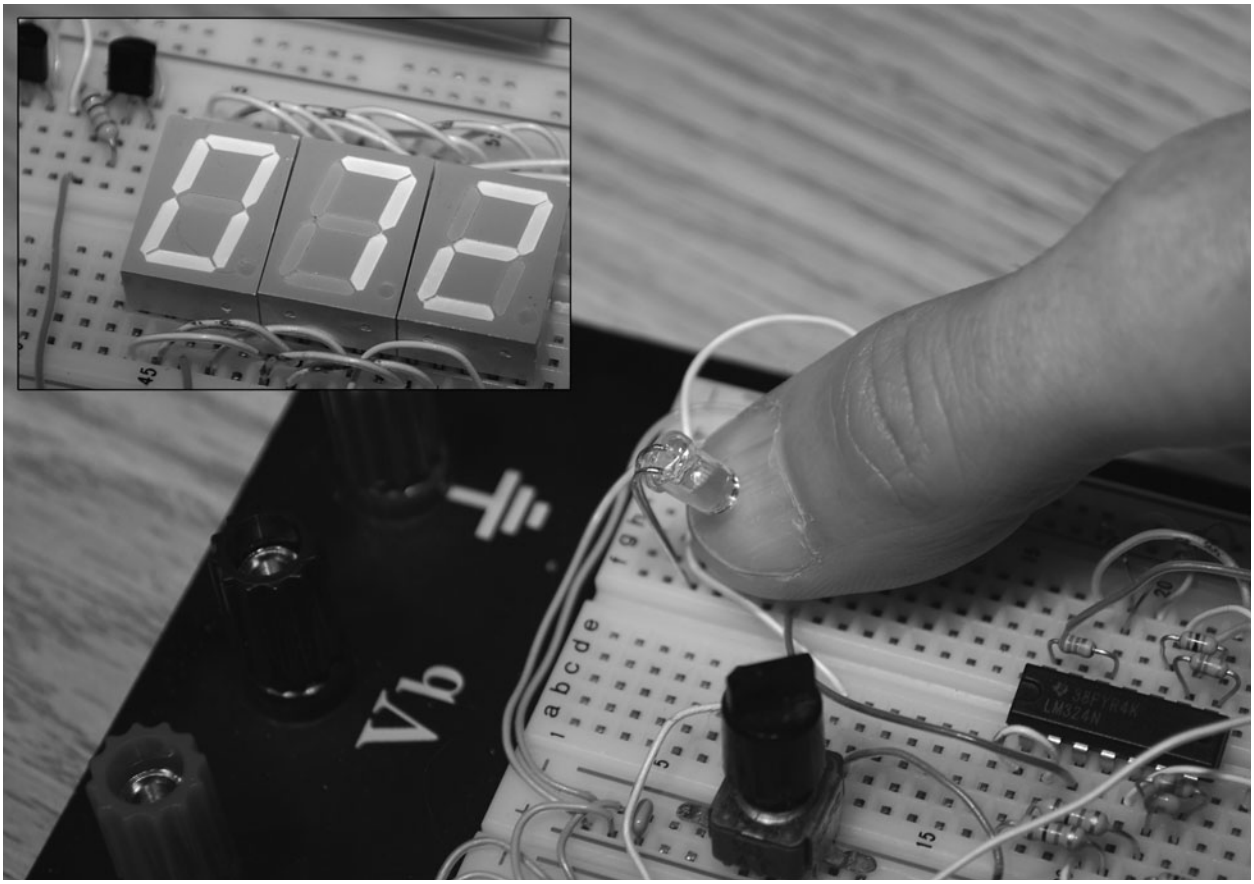


Figure 5-6 *Getting a new reading on the LED display.*

Figure 5-7 *This hair clip fits a finger perfectly.*

and LDR to your body, and remember that this device also will work through your earlobe. I intended to use my heart rate monitor for dream research, so it made the most sense to place the device on my thumb or toe so that it would not be annoying during the night. The method used to position the sensor should be both comfortable and secure enough not to move around easily.

To make the finger clip as compact as possible, the two ends of the hair clip are trimmed as shown in Figure 5-8. The clip now will work on any finger and on a large toe, although it seemed to work best on my thumb. I have poor circulation in my hands and feet, but since there is a large artery in our thumbs, the pulse was strong there no matter how cold my hands felt. Shooting the beam through the area of your fingernail also seemed to be the most effective place to get a strong and solid reading on the pulse-indicator LED.

The LDR and the visible red LED are placed at each end of the hair clip, as shown in Figure 5-9, so that the beam is directed onto the surface of the LDR once the clip is placed around a finger or thumb. My LDR had a small plastic bubble over the top of the detector surface, but some do not. I am not sure if moisture from your skin would affect an LDR without a lens, but if it does, then a small piece of tape or clear plastic placed over the surface of the LDR solves the problem easily. Don't worry if the LED beam

Figure 5-8 *The hair clip is trimmed to make it smaller.*

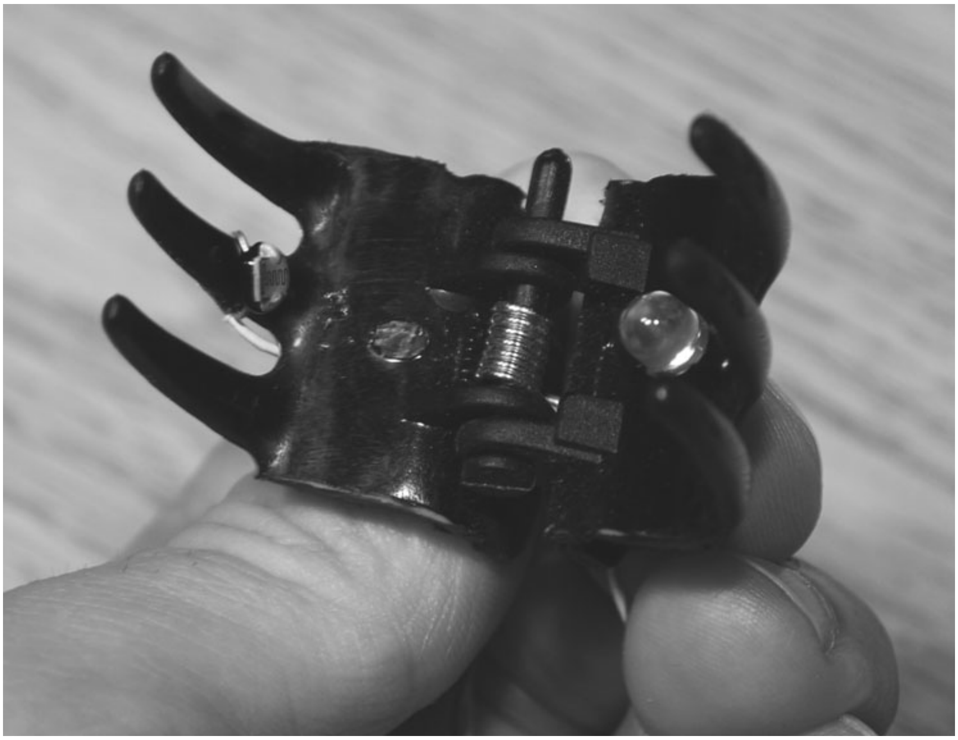


Figure 5-9 *Open jaws reveal the LED and the LDR.*

does not always line up perfectly onto the surface of the LDR because your finger will diffuse the light anyhow before it reaches the sensor. Since the analog circuitry is so sensitive, it takes very little light to actually cause the pulse detector to operate.

To ensure that the finger clip was working as well as or better than the original breadboard sensor, I placed it into the circuit for a reading. The variable resistor (VR1) actually needed a bit of a tweaking because the finger clip was much better aligned and passed more light to the LDR. After 10 seconds, the heart rate display climbed from the default power-on value of 60 to 89 beats per minute as my test subject held still under my massive camera lights for the shot. The elevated pulse was due to straining in the awkward position under 2000 Ω of light for over a minute while I got the shot I wanted.

It's amazing how little it takes to make our pulse shoot up from the normal, "doing nothing" rate. Just getting up from the desk was enough to add 5 beats per minute to my heart rate, although it quickly went back down to just over 60 beats per minute if I was not exerting myself in any way. Now the task of moving the circuit from the breadboard to a more suitable home will begin.

You will need a perf board that is at least 4 × 4 in to contain all the components in this project. However, take your time with the wiring, and start with the analog part of the circuit, just as was done on the breadboard. There are a lot of wires!

To keep the circuit board as small as possible, I replaced the LED displays with the smallest units I could find. If you have some talent with a soldering iron and PCB-making equipment, you probably could shrink the heart rate monitor down to the size of a matchbox, but I prefer the "old school" perf board method of prototyping because it is very easy to modify and correct errors.

Figure 5-10 shows the completed perf board ready to find a new home inside a plastic electrical box found at a local hardware store. The variable resistor (VR1) was replaced with a

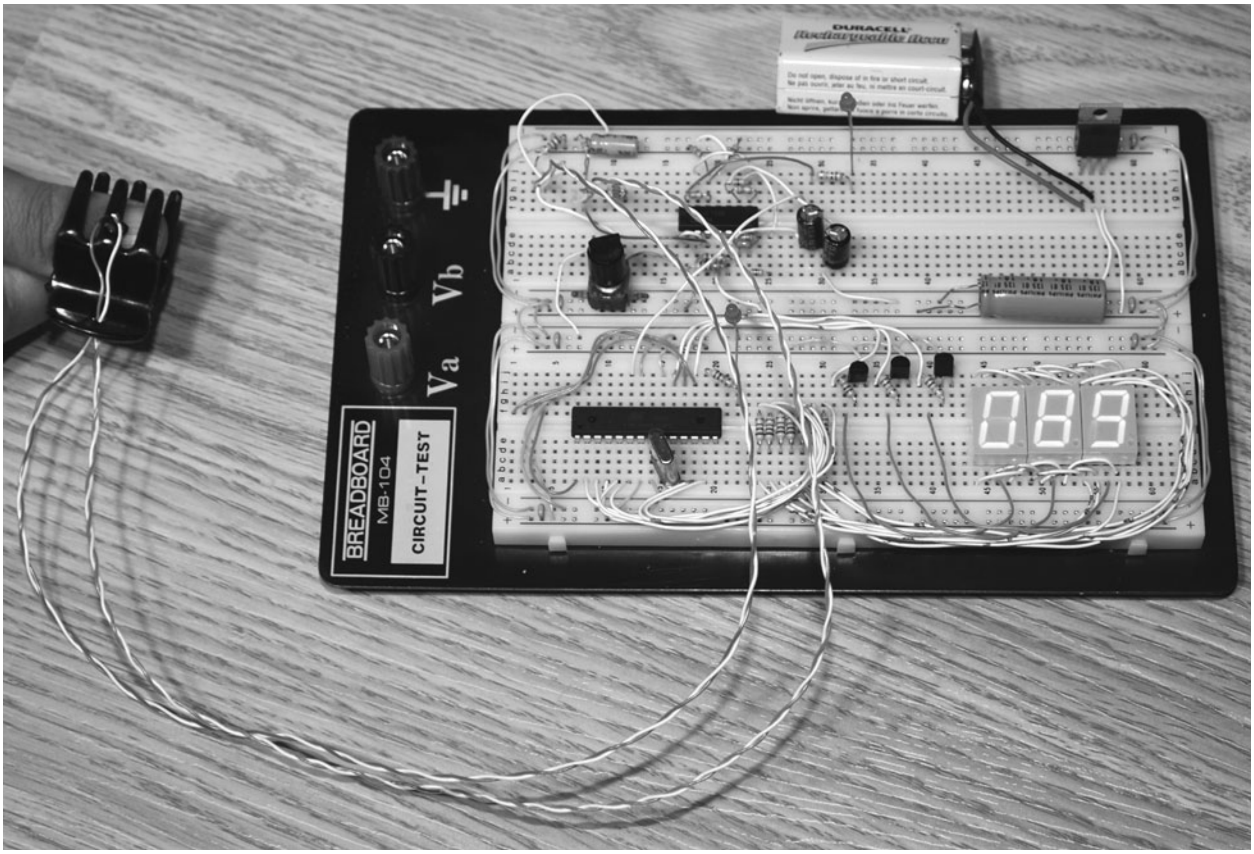


Figure 5-10 *Heart rate monitor on a perf board.*

board-mounted potentiometer because it only required adjustment once for the new clip-mounted optical sensor. If you plan to experiment with different sensor styles, then mounting the variable resistor externally might be a better option because it could require tweaking depending on the angle and amount of light hitting the LDR.

The completed heart rate monitor shown in Figure 5-11 performs as well as any store-bought unit but allows easy modification to adapt to just about any device you can imagine. There is plenty of room inside the cabinet for a small rf module, which would allow the heart pulses to be sent wirelessly to a computer for more advanced data logging. Because there are free IO pins on the microcontroller, it would be very easy to control some other device through a transistor or relay for further enhancements to the hardware. The source code is also very easy to adapt and expand because it was written to be as minimal as possible using the Basic language. Each block of code will be explained so that you can easily convert it to any microcontroller family or language.

Have a read through the complete Basic source code of Listing 5-1 provided in the appendix so that you can get an idea of how the program works. If you are an experienced programmer, then this trivial code probably is something you could write in 15 minutes from scratch, but if you have never written a program in your life, not to worry because Basic is called that for a reason and is easy to understand. The lines in the program listing that start with an apostrophe are comments, and I will explain the code in blocks after each comment. The compiler used is the Bascom AVR, and it is a very good compiler for the Atmega family and includes a very comprehensive instruction set.

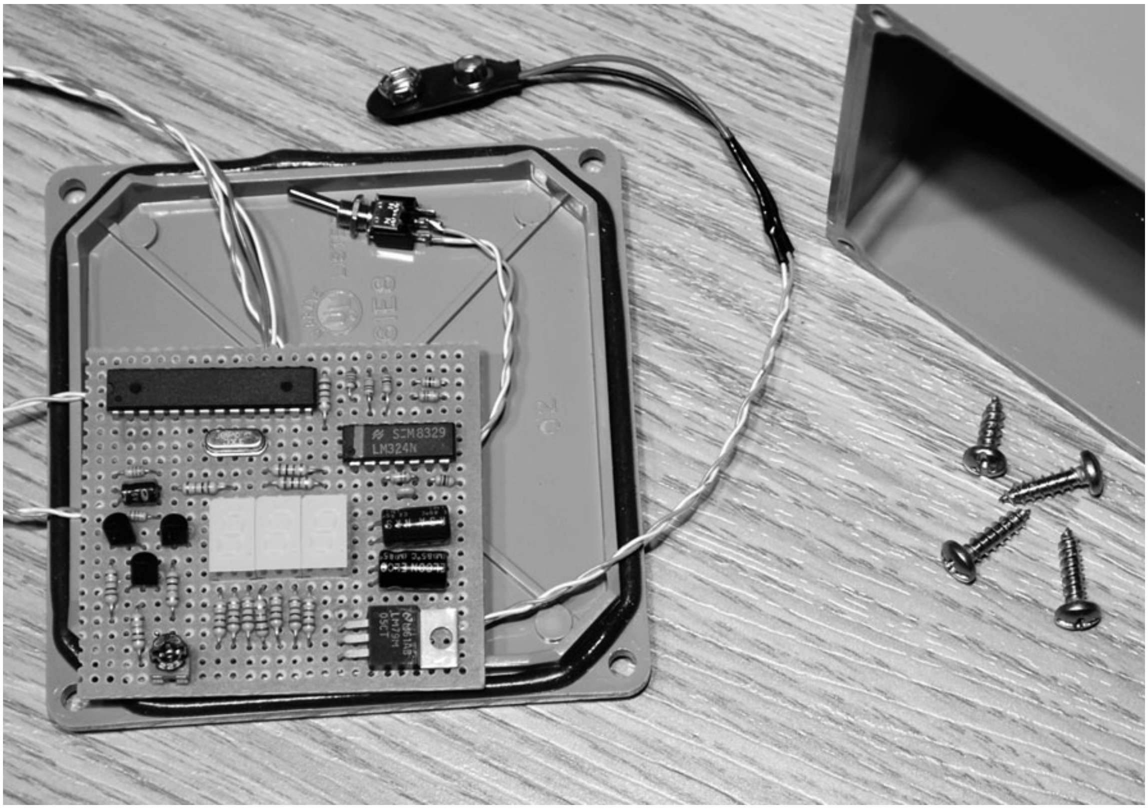


Figure 5-11 *Installing the components into a cabinet.*

Figure 5-12 *The completed heart rate monitor.*

The code following this comment is required to tell Bascom that we are going to target the Atmega88 device and that our clock will be an external crystal resonator running at 4 MHz. Telling the compiler your clock speed becomes important when using commands that deal with timing-sensitive routines such as serial transmission or analog to digital readings. Defining the device also helps the compiler to generate user errors that have to do with IO pins. In this way, you can't accidentally try to toggle an IO pin that does not exist on the actual device.

This block of code sets up the IO pins that will connect to the LED display outputs. You can change these around any way you like to make your wiring as simple as possible, but it is best to keep the LED digits to a single port for simplicity. Port D is being used here.

Basic uses *variables*, which are letters or words used to hold values. I like to use single letters such as *A*, *B*, and *C* for simple programs such as this one, but when you are working on a large, complex program, use of more descriptive variable names is recommended. "TIMER2" or "REDLED1," for example, would be descriptive variable names that make a lot more sense in a huge block of code.

Variables are also defined as the number of bits they are to contain, so in our code, "A" and "B" are 8-bit bytes that can contain a value between 0 and 255. Variable "F" is an integer that can range in value from –32768 to +32767. Variable "D" is a "Word" that can contain a value between 0 and 65535. Although you could just define all variables using larger data types, this is a waste of memory space and will slow down your code.

This block of code sets up an array of 10 values for the variable "LED." Although this may seem confusing at first, the values correspond to which of the seven segments will be lighted to display a particular number. To make this a little more confusing, the value in parentheses is actually one higher than the represented numerical value, and to light a segment, we want a low bit, not a high bit, so the value of "Led(9) = 0" says that to display the decimal number 8, we want all bits to be off. This means that all segments light up, and an "8" will be displayed. It can be a bit of a chore computing these values by adding port bits, but once you have done it once for your segment, the hard work is completed. Your values most likely will be completely different from mine as you "map" out the pins and segments on your LED displays.

This sets up the variable "K" with a default rate that is close to the typical resting adult heart rate. By setting the average to 60 beats per mnute rather than starting it at zero, the calculated value does not have to "ramp" up so far to reach its final value. This will make more sense once you see how the heart rate monitor operates.

The "Start Adc" command tells the compiler to include the code necessary to set up and initiate the onboard analog-to-digital converter on the Atmega88. This will allow us to read in an analog voltage and convert it to a value in order to detect changes in voltage from the LDR sensor.

Everything from here on is going to happen continually until the word "Loop" is reached, which causes program execution to start again where it first encountered the word "Do." This is called an *endless loop* because it never stops unless forced to by another command or an error.

This command reads the analog-to-digital converter on pin 0 of the Atmega88 into variable "D," which is where the output from the LDR is connected. Since the ADC returns a 10-bit reading, values can range from 0 to 1024, which is why variable "D" needed to be a "Word," not a byte.

These three lines compare the current ADC reading "D" with the last known reading "E" by subtracting them into "F." The "Abs(F)" command changes the value in "F" to an absolute nonnegative value so that we only get the difference as a whole number, not a negative number, if "E" were greater than "D." Thus what this does is only give us the difference since the last ADC reading, not the actual value.

This chunk of code acts like a crude low-pass filter that does not allow pulses to be detected that are much faster than what would be considered the upper limit for a human heart rate. Each time a new pulse is detected ("F > 4"), a counter is set to 40 and has to count down to zero before another pulse can be registered.

To aid in debugging the circuit and setting up the optical sensor, a pulse-indicator LED will flash on both the analog part of the circuit and the digital part of the circuit. This piece of code just looks to see if the heart beat filter has been reset and then turns an LED on and off for half the counter cycle.

To display the heart rate as beats per minute, a running counter "H" is incremented until the next pulse is detected. The value in "H" then is divided into a value (9840) that was calculated to give a fairly accurate result of beats per minute. This large number has to do with how many instruction cycles there are between the main loop of this program and took a bit of effort to get using an oscilloscope. Changing the auscultator frequency would completely throw off the calculated heart rate.

To filter out the regain a little more, changes do not happen immediately but are "ramped" up or down on each pulse by a value of 1. By waiting until each pulse is detected ("G=1"), the final value "K" is compared with the calculated value "J" and incremented or decremented accordingly by 1. Without this ramping sequence, a single glitch or false detection would result in a wild and instantaneous change in the LED readout.

Since the LED display contains three individual seven-segment displays and the actual value is contained in the "Word" variable "K," it must be broken down into three digits and sent to the display routine, which expects a value between 0 and 9 to be stored in each variable "A," "B," and "C." By using division and the "Mod" command, the values are taken from the variable "K" and broken into 1s, 10s, and 100s for the display routine.

The "Loop" command causes program execution to jump back to the main routine where the "Do" command was encountered. This is the end of the infinite loop, and all other commands beyond here must be called by either the "Goto" or "Gosub" command.

This is the routine that displays a digit on each of the three LED displays. A little trick called *persistence of vision* is used here to switch between displays so fast that your eyes think that they are all on at the same time. As you can see, only one bit of the three "Portb" pins is on at a single time, and then the variables "A," "B," and "C" are sent to the display for only 2 ms each.

Because 2 ms is so fast, it appears that each display is on all the time, and the seven-segment lines can be shared, saving valuable IO overhead. Since Bascom array variables start at 1 not 0, the line "D = A + 1" adds 1 to the array pointer so that the value in "Led(d)" is the same as the decimal value we want. This conversion just makes it easier to understand the code, especially when trying to compute the segment bits from scratch the first time. Once all three displays have been lighted for 2 ms each, this routine just "returns" to where it was originally called.

I'm sure that you will find many ways to improve the program code and modify this project to suit your needs. With all those unused IO pins and code space, it would be easy to create a lucid dream-detection system by adding an 8-hour timer that looks for dream stages by comparing time with fluctuating heart rate. The addition of a relay or driver transistor could trigger just about any device, creating a complete dream-detection or -induction system that would work just as well as the eyelid movement–sensing device presented earlier. The ability to write Basic code and work with microcontrollers makes rapid prototyping a snap, so there really is no idea you can't set in motion with a little imagination and work. Enjoy!

For the last project in this section, we will continue to explore the reactions of the human body with an effective, sensitive lie-detector device.

Project Six

Lie Detector

A real lie detector, or *polygraph*, is a complex and sensitive piece of electronic equipment that logs several physical responses at the same time as the subject undergoes a battery of questions. A real polygraph, such as the ones a government agency might use, would usually include circuitry to monitor blood pressure, pulse, respiration, breathing rhythms, body temperature and skin conductivity, and even brain waves. The simple version of the lie detector presented here, based on the very old designs, measures only skin conductivity, although it could be used in conjunction with many of the other projects in this chapter to create a much more elaborate lie detector.

The lie-detector schematic shown in Figure 6-1 is surprisingly simple considering that it does a very good job of measuring the subject's skin conductivity. The two transistors form a very-high-gain amplifier that measures a tiny amount of current that will pass along the surface of the subject's skin as he or she holds onto the test probes. Since skin does not conduct if it is completely dry, the more the subject perspires, the higher will be the reading on the analog meter. Since it is a known fact that we perspire a little more when we lie, the tester can grill the subject and then look for small changes on the meter. Since the amplifier is so sensitive and we all have differing levels of skin conductivity, the variable resistor (VR1) can adjust the initial setting so that the meter is pointing to the middle of its range before the test begins. If the subject tries to relax too much, this will cause the meter reading to drop and could be interpreted as the subject trying to trick the test. A higher reading following a question would indicate that the subject may be lying and perspiring more than normal.

The circuit is so simple that you probably can forego the breadboarding process and build it right onto a bit of perf board. As shown in Figure 6-2, there are only two transistors and two resistors on the board. The only change you might need to make is the addition of a resistor in series with your analog meter if it happens to shoot all the way to the end even when the adjustment VR1 is all the way down. If your analog meter fails to move much at all, then you might have to increase the voltage by adding another 9-V battery in series with the current battery to make 18 V. Likely, your meter will be fine with this circuit, but it really depends on the impedance and rated voltage of the tiny coil inside. Most analog meters will respond to a very tiny voltage, even if their readouts say something like "kiloamperes" or "megavolts"!

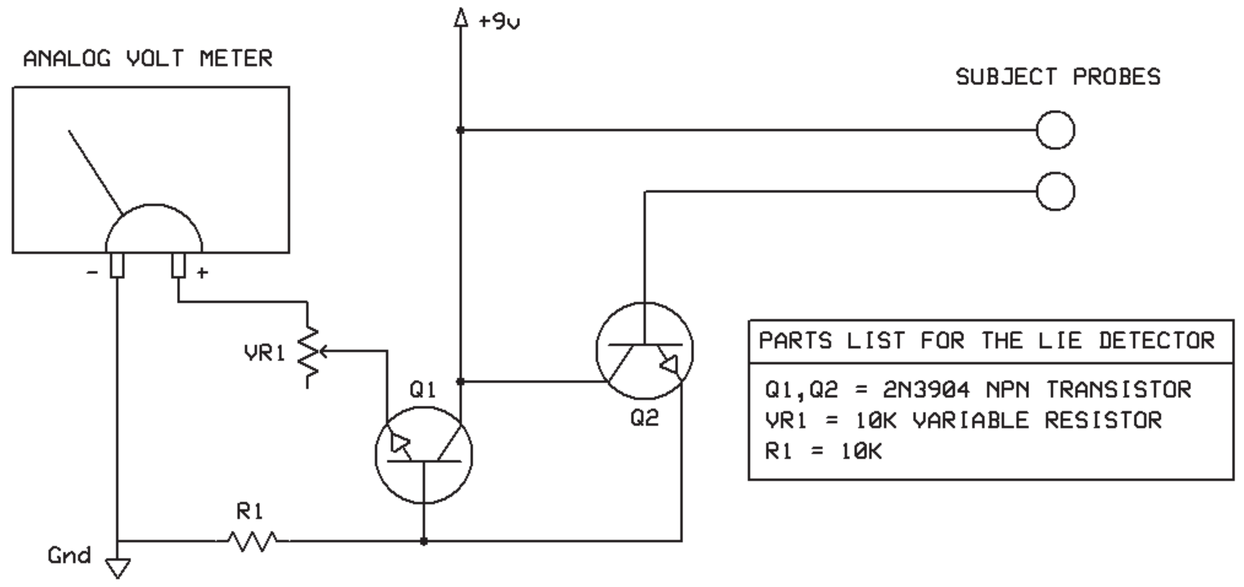


Figure 6-1 *The lie-detector schematic.*

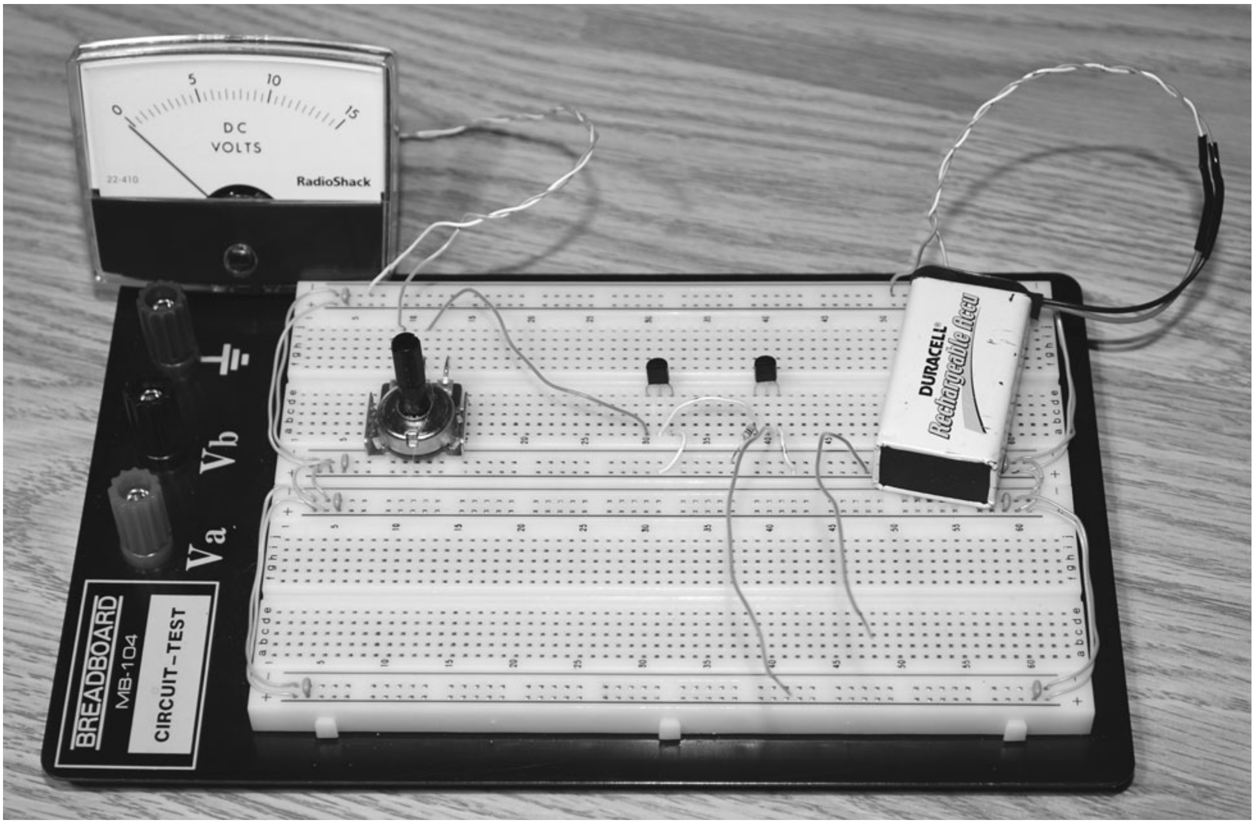


Figure 6-2 *The lie detector on a perf board.*

Analog meters can be quite costly to buy brand new, so you might want to do a little scrounging to find one to play around with. Second-hand shops often carry old stereo gear that may have one or more analog meters. Another source might be an old CB radio, battery tester, analog multimeter, or any other appliance from the 1970s or 1980s that had to display some value. Now that LEDs and semiconductors cost pennies, expensive mechanical display devices are very hard to find on mass-produced consumer electronics.

If you turn up the variable resistor and grab hold of the probe wires, your analog meter should jump to the end of its range in a hurry. If the meter fails to respond, try turning the variable resistor the other way in case you have the wiper lead connected in reverse. If there is still no response (unlikely), connect the probe wires together to drive the amplifier to its maximum value. If the meter still fails to move, then it is either not working or requires a lot more voltage to move the needle. If you are having the opposite problem and can't turn the variable resistor down enough to get the meter below the halfway mark, then add a resistor in series with the meter. A 1K resistor would be a good one to start with. Figure 6-3 shows the response you want to get—the meter at the halfway point after adjusting the variable resistor. You should be able to make the meter swing all the way over by wetting your finger or just by putting a lot of pressure on the probe wires.

Your analog meter likely will have a scale measuring amperage, voltage, or some other value unrelated to our biologic subject. It is actually very easy to make a replacement readout plate just by cutting out something you made in a computer paint program and then affixing it to the original plate. The plastic cover can be popped off by placing a small screwdriver or blade in the tiny slots to pry it off the backing plate, but be careful not to damage the sensitive meter or any of its moving parts once the cover has been removed. Figure 6-4 show the replacement readout plate I made that is a bit

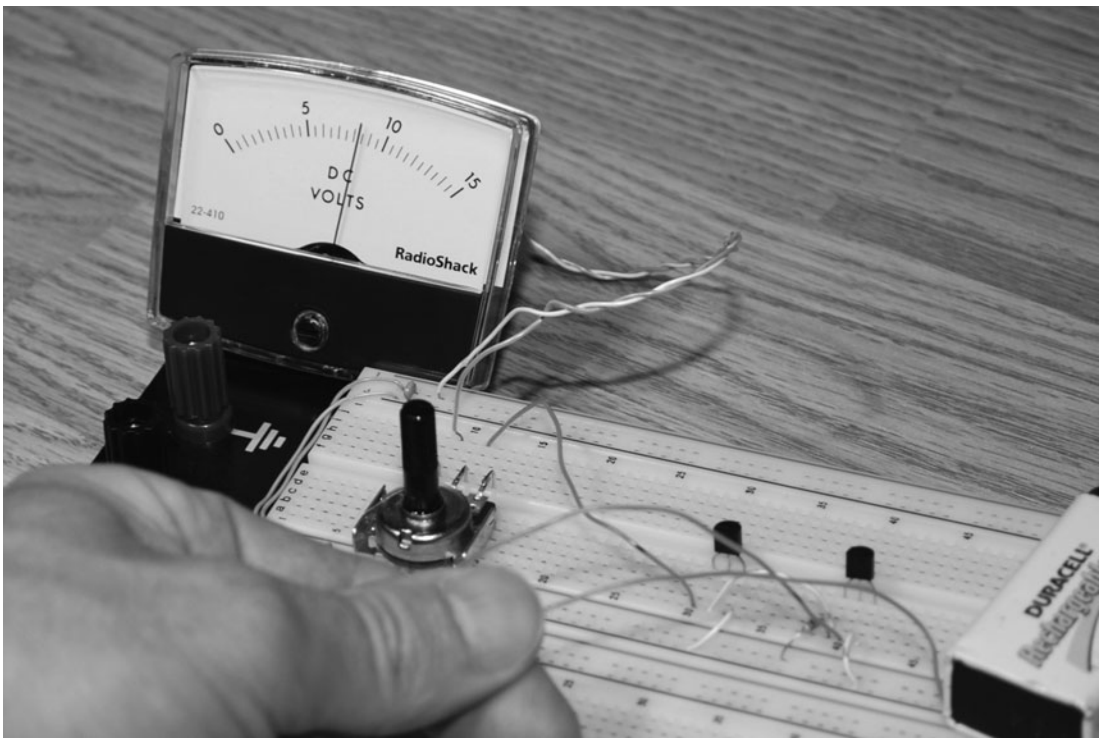


Figure 6-3 *Testing the range of the analog meter.*

Figure 6-4 *Making a replacement readout plate.*

more on the "funny" side of things because I planned to use my lie detector just for kicks. You can make your readout say whatever you like, but try to create three distinct zones so that you can have your needle point to the center point when adjusted properly. Remember, a low reading will indicate either a bad connection between subject and probes or a subject trying to fake the result, and a high reading will indicate above-normal perspiration or a lie.

Once you have your clear plastic cover pried from the backing plate, simply glue or tape your new display over the old one, as shown in Figure 6-5. You do not have to remove the original reading plate because there will be plenty of room between the needle and the original plate to insert your paper display as long as it is glued down flat. I used a small dab from a glue stick to hold the new readout in place so that it can be removed easily at any time without damaging the original plate. Again, be careful when working inside the meter because that tiny needle is very easy to bend or break.

Although you can connect the probes just about anywhere on the subject's body, the old "sweaty

Figure 6-5 *Gluing the new display over the old one.*

palms of a liar" factor certainly applies to this device. I found a few steel rings at the local hardware store and then soldered wires to them, as shown in Figure 6-6, to make a good pickup for the subject's palms. The goal is to affix the probes to your subject in such a way that they make good contact with the skin but do not allow the subject to change the pressure on the probes, which would cause a false readout. By placing the rings on the subject's palms as he or she holds the hands palms up, you get a good connection without allowing the subject to grip or manipulate the pressure between the metal and the skin. If your subject has particularly dry hands, you can try placing the rings further up his or her arms or use elastic to apply a fixed amount of pressure to the probes.

For such a sparsely populated circuit board, there are sure a lot of wires coming from it. The analog meter, power switch, battery, probes, and variable resistor are all connected to the functional circuit shown in Figure 6-7, ready to be installed in some type of cabinet. I also decided to connect my probe leads through a ⅛ stereo jack so that I could create multiple probe sets to allow the use of variously shaped probes. You also could connect multiple probes to your subject as long as they are the same on each side of the body. On a real polygraph machine, many probes are connected to the subject for more accurate results.

There are two ways you can mount the analog meter in your project box: behind a square hole or over the top of the lid with only the "can" stuck through the lid. Making a square hole that looks good requires a bit of patience as well as the notching tool shown in Figure 6-8. To use a notching tool, trace the area to be cut out, and then drill a hole in each corner large enough to insert the tool's cutting blade. The notching tool allows you to "nibble" out small square bites from thin metal or plastic cabinets to create square or straight-edge holes. An easier way to mount an analog meter is just to cut a hole large

Figure 6-6 *Making probes for the subject's hands.*

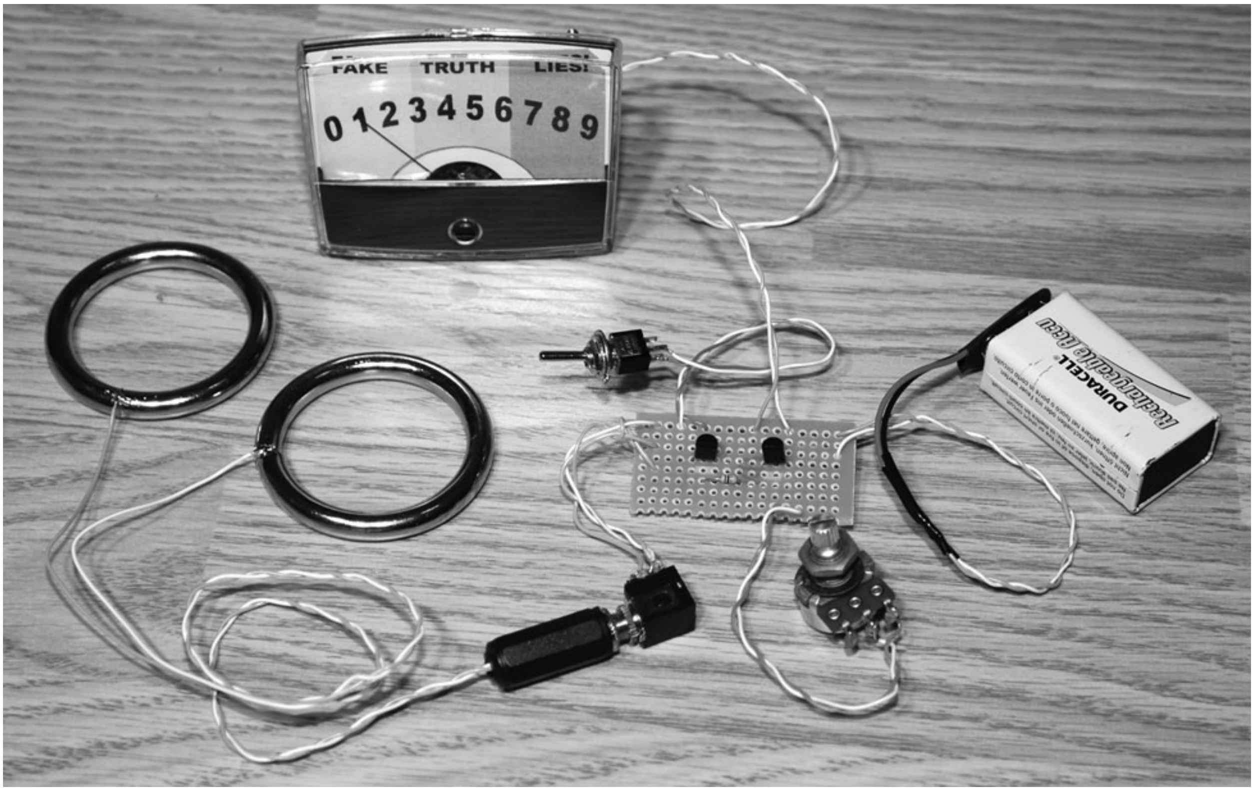


Figure 6-7 *The circuit ready to be installed into a cabinet.*

Figure 6-8 *A notching tool is great for cutting square holes.*

enough for the mechanical body part to fit through the lid, and let the display area cover the hole. This method is simple, but then you see the nondisplay area under the meter, which does not look as professional. Since the plastic cabinet I planned to use was much too thick for the notching tool, I chose the easy way out!

Like Agent Mulder from the *X-Files* once said, “The truth is out there.” And with your new lie detector, you can cut through the deception and misinformation. After a bit of hole drilling, the completed lie detector shown in Figure 6-9 was ready for action. Of course, with my analog meter reading “Fake, Truth, or Lies,” it is obviously more of a party gag than a tool I intend to use to interrogate my “enemy spies.” You actually could combine this device with some of the other biofeedback devices shown earlier in this chapter to create a much more comprehensive lie-detection system, but you also will require the knowledge needed to decode all the feedback received from your subject. The art of using a polygraph is so controversial that some countries do not even consider it a valid test, and often those with knowledge of how the device works can learn to fool the tester.

I’m sure that you will find some useful application of your simple lie-detector unit and have some fun at the same time. If you make the unit look “real” enough, then it might be just as effective at “extracting” the truth as its big brother, the polygraph machine. Sometimes the risk of having a lie exposed is enough! In Section Two we will explore the dreaming world.

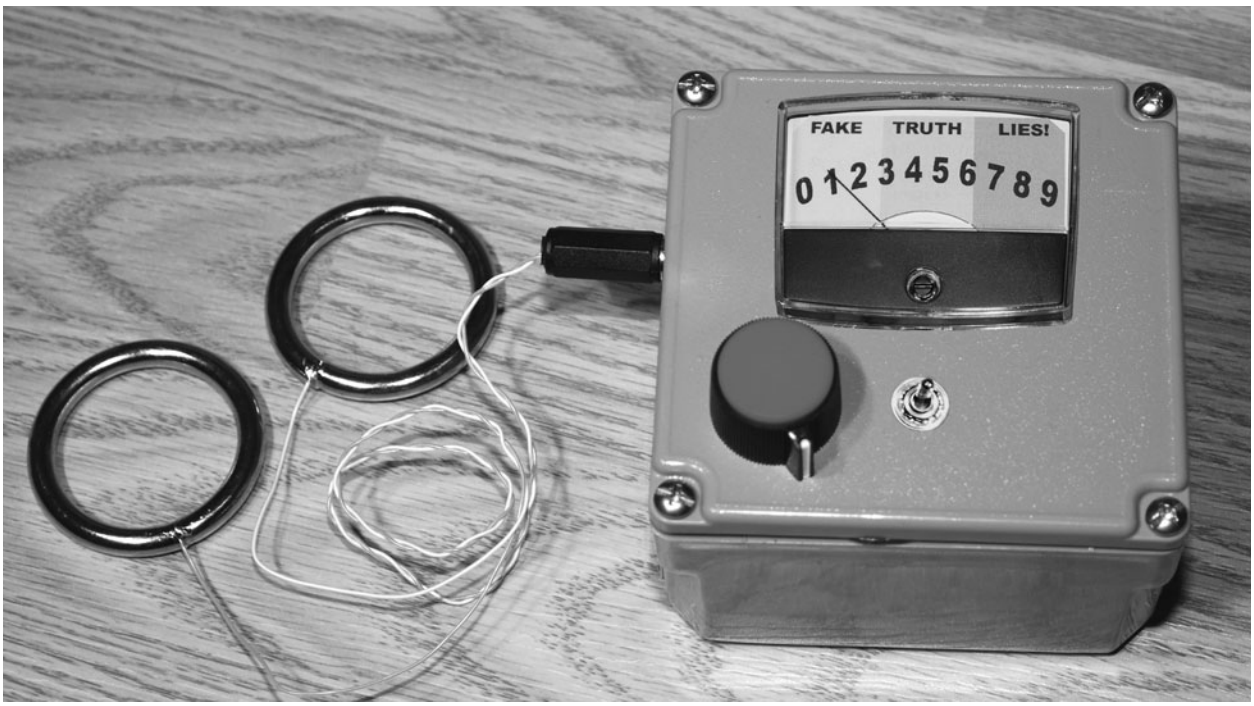


Figure 6-9 *Ready to uncover the truth (or lies)!*

Section Two

The Dreaming World

This section explores the other half of your life that unfolds when the lights are out and you are in that strange place called *dreamland*. You might think that a dream is something that we have no control over, but this is not the case, as will be shown with some of the devices presented here that will allow you to link your sleeping mind with your conscious mind. Imagine being able to enter a world where there are no limits or rules and where even the laws of physics are as elastic as the strange and wondrous atmosphere that surrounds you. Having conscious control over our dream world is something we are all capable of with a little training and help from the electronic world. So get ready to become an "oneironaut" and travel through your dream world, painting the scenery at will and allowing your wildest fantasies to unfurl because this section is all about connecting with the other half of your life.

This page intentionally left blank

Project Seven

White-Noise Generator

White noise can be best described as a hissing sound that an FM radio makes when tuned to a place on the dial where there is no station or static. White noise is a smooth hiss consisting of all audio frequencies played back at random simultaneously, which is why it has no discernible tone. White-noise generators have many uses, some of which include the testing of audio equipment, random-number generation in digital circuits, jamming audio bugs, and many medical uses that we will be able to test and experiment with using our simple device.

White noise is known for its calming effects, which is why many people find it easier to sleep with a fan running or some other source of random noise similar to white noise. Ocean waves, trickling water, and wind through trees are a few other examples of white noise, although they are not as steady as the white noise that will come from our home-built device. Because your ears cannot "lock" onto any specific frequencies or patterns in white noise, the device has the effect of shutting off your ears to distracting noises, just as being outside in the middle of a fresh snowfall on a bright day will give you snow blindness. Sensory deprivation using white noise even has been used as an interrogation aid to temporarily confuse subjects before questioning.

White noise is also used as a treatment for tinnitus and hyperacusis by distracting a patient's hearing so that it always hears something yet at the same time really hears nothing. Maybe you will want to try white noise as a sleep aid in your noisy apartment, or perhaps you want your next hacker gathering to be completely secure so you want to flood your windows with white noise to defeat the laser-bounce eavesdropper. Whatever your plans may be, this extremely easy-to-build white-noise generator will fit the bill and even can be made as a dual-channel stereo system.

The white-noise generator schematic shown in Figure 7-1 is so simple that you might think there is something missing, but the unit works very well and exploits the fact that transistors generate random noise when reversed biased. The reverse-biased emitter-to-base junction of Q1 is fed into a 1-Ω audio amplifier integrated circuit (LM386) so that it can power a speaker or a set of headphones. Because many transistors will act differently in this application, I have added a 100K variable resistor (VR1) to allow for some "tweaking" of the white noise for best results. You can use practically any small-signal NPN transistor in this circuit, and I have found the commonly available 2N3904 to give great results. Another variable resistor (VR2) controls the level

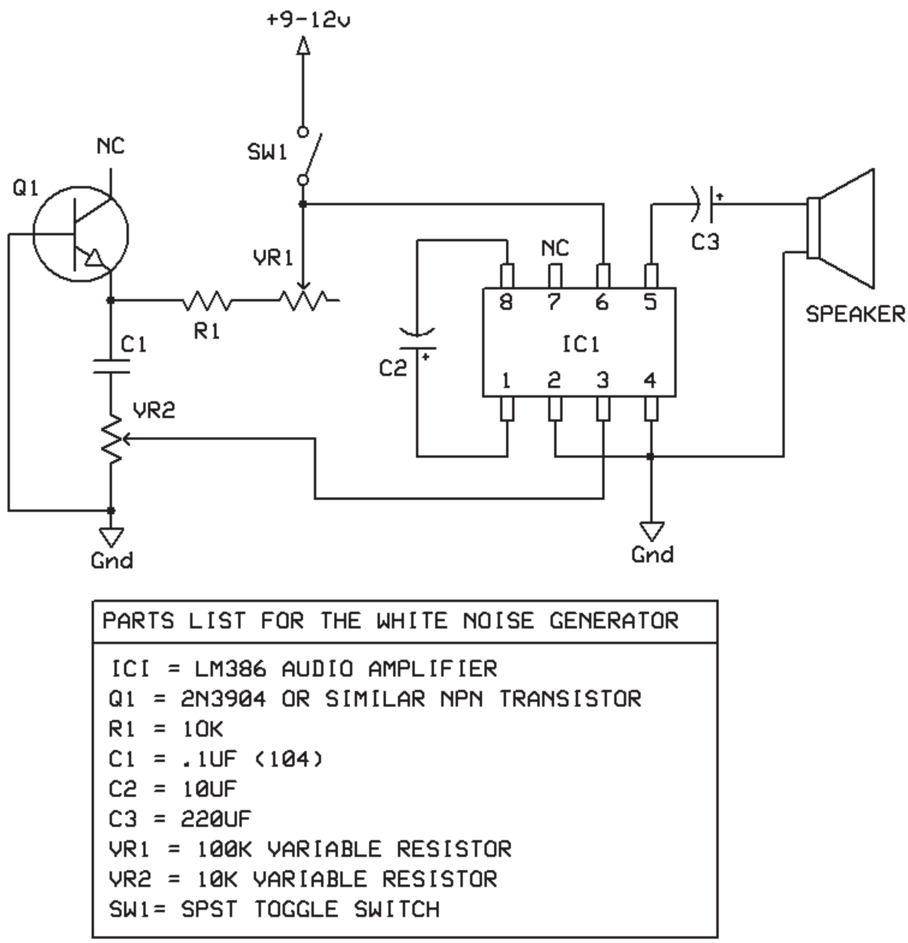


Figure 7-1 *White-noise generator schematic.*

of noise reaching the audio amplifier so that you can adjust the level to suit your needs. Even though the LM386 audio amplifier is just a small 8-pin intergrated circuit (IC), it actually can drive any size speaker you like and is probably more than loud enough to fill a room full of clean white noise.

For a very relaxing white-noise generator that will seem to send noise in all directions, you can build the stereo version shown in Figure 7-2 simply by doubling up the original version and sending the outputs to a pair of stereo headphones, speakers, or the line inputs on your audio amplifier. Because there are two independent noise generators running, the sound will seem to come from all directions, which works amazingly well for audio-deprivation experiments where the subject is wearing headphones or is placed directly between the two speakers.

If you do plan to build the stereo version, try to keep all parts the same make and value so that the sound is even on both channels. Since some transistors may behave slightly differently in this circuit, they should both be exactly the same. To simplify the stereo circuit a bit further, you could replace the LM386 with a stereo amplifier IC to reduce the parts count a bit. If you plan on feeding the output directly to a line input on an amplifier, then you actually can eliminate the entire amplifier section and simply feed the output from the center tap of VR2 directly to your amplifier's input.

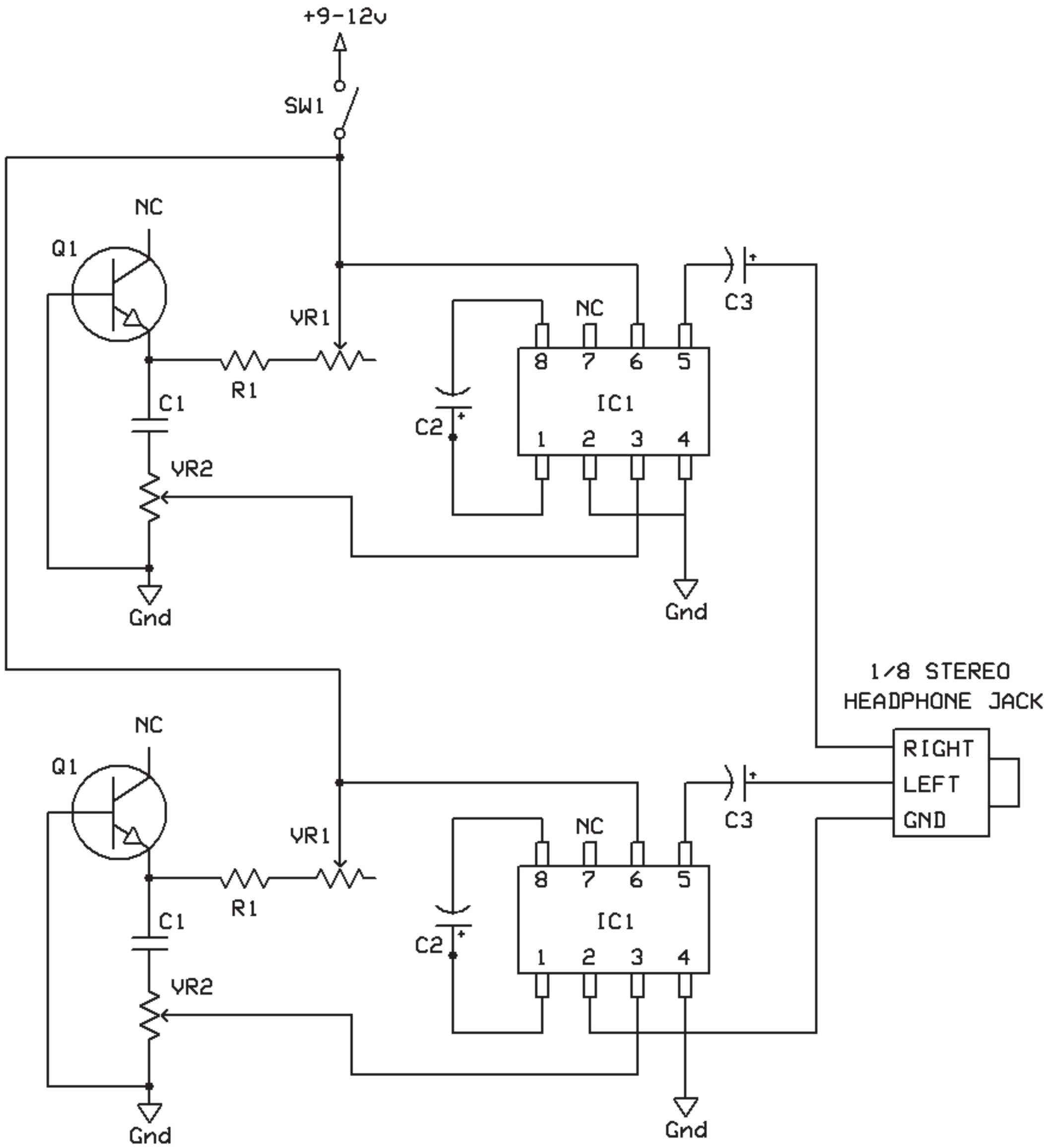


Figure 7-2 *Stereo white-noise generator schematic.*

Figure 7-3 shows what white noise looks like on an oscilloscope or computer input. Since white noise is comprised of every possible frequency at the same time, there is no discernible pattern or tone that can be detected, which is why your mind finds it so relaxing to listen to. White-noise generators like this are often used in digital machines that need to create some kind of chaos for a random output. Digital encryption and lottery machines are prime examples of using white noise as a source for random information because digital circuits cannot generate truly random sequences that never repeat.

The simple circuit can be built on a solderless breadboard (Figure 7-5) in a few minutes. As soon as you power up the circuit, a nice clean hiss should be heard from the speaker if everything is working well. If your noise sounds more like a crackling or spitting oscillator, then try moving VR1 to tweak the sound a little bit. If your noise is still not smooth and crackle-free, just drop in another transistor, and listen to what you get. The

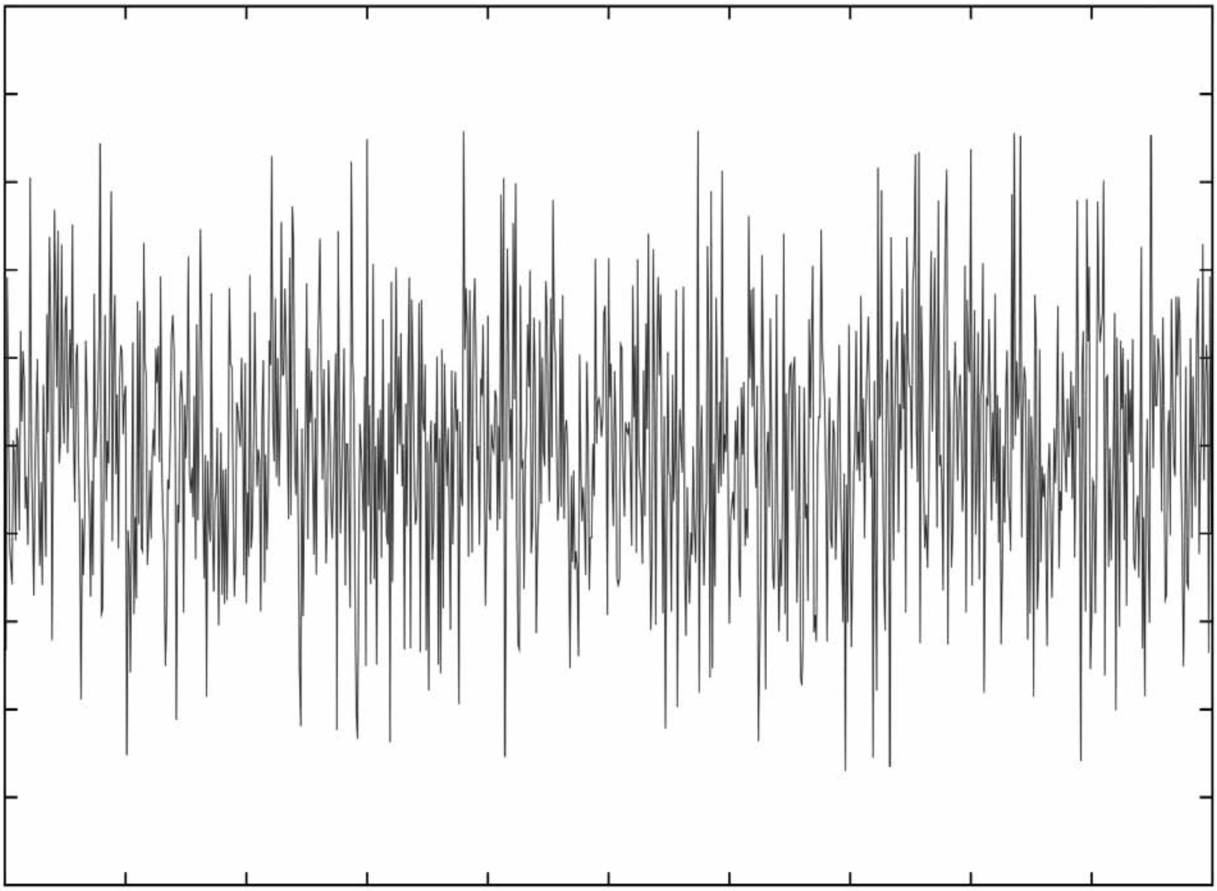


Figure 7-3 *White noise is a random wave.*

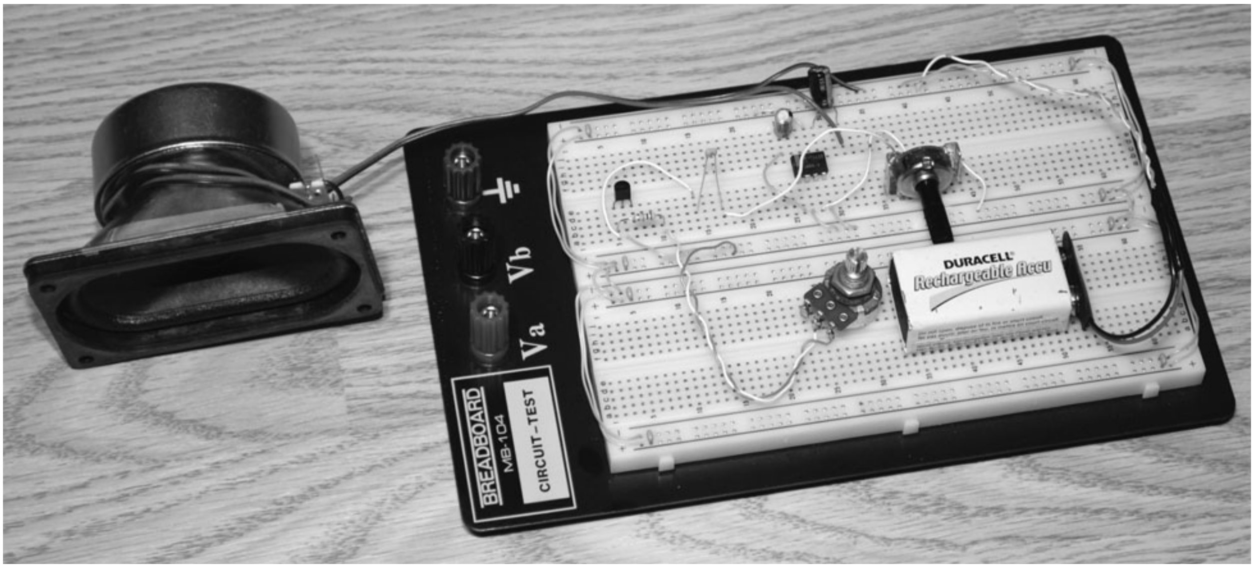


Figure 7-4 *Breadboarding the white-noise generator.*

optimal white noise is a very clean hiss much like a radio with no station or the sound of air leaking out of a tire. Since the LM386 audio amplifier is happy on any voltage from 9 to 18 V, you might want to try a 9-V battery for the absolute cleanest possible output. A dirty power supply will add hum into the circuit.

If you are planning to use headphones, turn down the volume control (VR2) at first because the output of the audio amp can be very loud. If you find the output to be much too loud, remove capacitor C2, and that will change the gain on the LM386 from 200 to only 20, making the output much quieter.

The perf-board version of the white-noise generator shown in Figure 7-5 is extremely small, so it can be installed into just about any cabinet. If you are happy with the “flavor” of white noise after adjusting VR1, then you simply can measure the impedance when it is set correctly and replace it with a fixed resistor to simplify the circuit even further. I decided to leave the tweaking adjustment because it worked somewhat like a single-band equalizer to adjust the sharpness of the white noise.

On a single 9-V battery, the white-noise generator will run all day, although you also could use a 9- or 12-V direct-currebt (dc) adapter for batteryless operation if needed. The 4-inch-diameter speaker shown in Figure 7-5 was fine for a small room, but for a larger room or a more bassy sound, you probably would want to use a larger speaker or one already mounted inside a cabinet. A 10-in speaker in a cabinet will make the white noise much richer and more like wind rather than a hissing radio.

The stereo version of the white-noise generator shown in Figure 7-6 is great for relaxing or messing around with auditory-deprivation experiments. This version has dual noise generators, and both audio amplifiers have capacitor C2 removed so that the output is not too

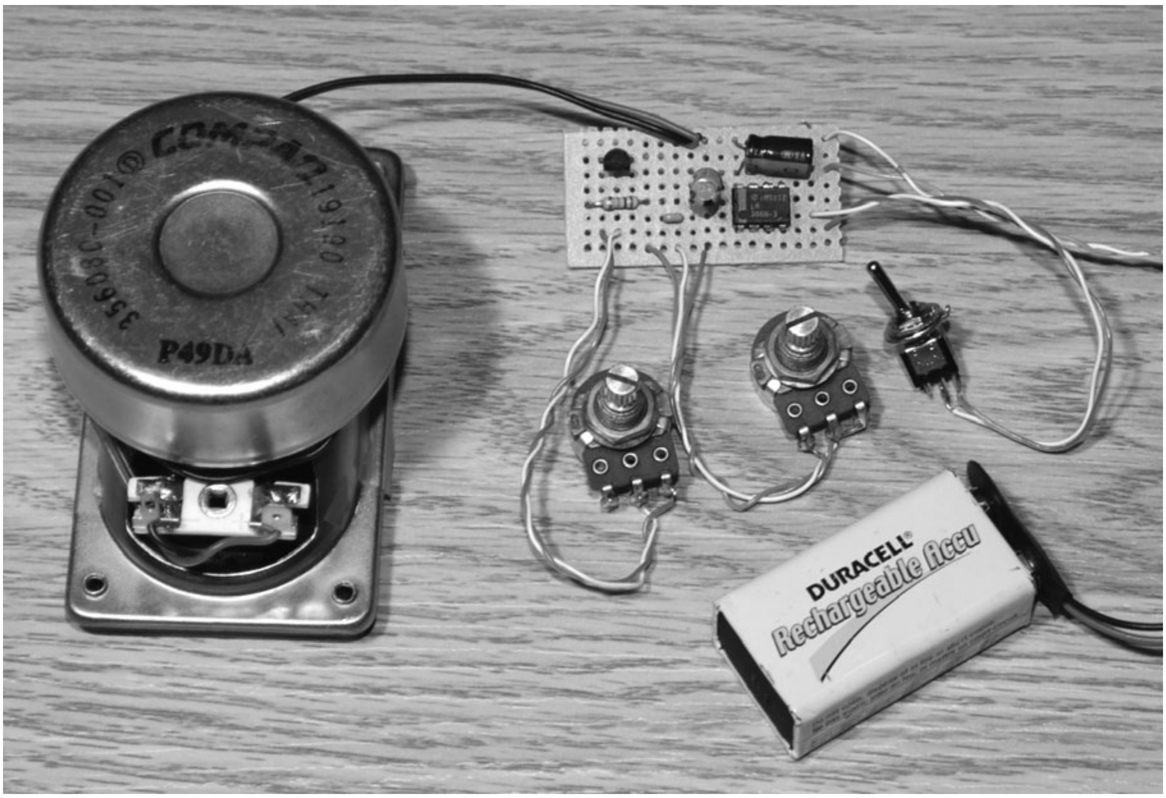


Figure 7-5 *White-noise generator on a perf board.*

Figure 7-6 *Head-mounted stereo version.*

Figure 7-7 *A larger version with more bass response.*

loud for the headphones. To connect this unit to a stereo amplifier for big-room sound, you just need a ⅛-inch stereo-to-RCA-jack adapter so that it can be fed into the line input of any stereo system.

The cabinet-mounted system shown in Figure 7-7 is great for filling a room with rich, bassy white noise that sounds more natural than what comes from a smaller speaker. Because of the full range the quality speaker allows, all the tones are represented rather than just the higher, "hissy" frequencies. This version of the white-noise generator is great for use as a sleeping aid or when you want to work in a noisy environment without being distracted by background noise.

There are many experiments you can do with white noise, so I will leave you to your own "evil genius" devices to have fun with this project now. Some ideas for design might be the addition of a voltage sweep circuit to simulate ocean waves or perhaps the addition of some type of audio equalizer or filter to shape the white noise into pink or brown noise. Next up, we will dive into the world of lucid dreaming.

Project Eight

Introduction to Lucid Dreaming

Imagine being able to immerse yourself in a world where you can do whatever you like without consequence and also have the ability to alter the very laws of physics and control everything you survey. This may sound like an impossible movie scenario, much like *The Matrix*, or a future prediction as to how virtual reality one day may be able to "jack" directly into our consciousness, but let me assure you that this utopian world can be yours right now without installing any plugs on the back of your neck! Lucid dreaming is an unlikely mix of waking reality and a vivid dream, a dream in which you actually can control your surroundings while aware of the dream yet at the same time experience your dreamscape as if it were the real thing.

The scientific community has been studying and documenting the lucid-dream state since the early 1900s, and the term *lucid dreaming* was first used by the Dutch psychiatrist Frederik Van Eeden. Of course, the mainstream scientific community doubted the very existence of lucid dreaming for many decades because it seemed highly unlikely that a person actually could be conscious and dreaming at the same time. Those who are "natural" lucid dreamers, of course, knew otherwise, but how do you prove that your experiences in the "other world" are true lucid dreams? It's much like trying to prove that there is an afterlife because to be in that state means severing all ties to the real world.

In 1978, internationally renowned psychologist Keith Hearne found a way to create a bridge between the lucid dreamer and the real world in an interesting experiment using a polygraph to measure eye movements. The idea was that on entering a lucid state, the dreamer was able to remember to move his or her eyes back and forth 8 or 10 times in the dream. Knowing that every part of the body with the exception of our eyes becomes paralyzed during sleep, Hearne was able to prove that the lucid dreamer was indeed conscious and able to take control while in a true dream state, as measured by an electroencephalograph (EEG), which measures brain wave activity. While doing this groundbreaking experiment, Hearne also determined that a lucid dream occurs during the rapid-eye-movement (REM) state of sleep. He also discovered that a lucid state occurs most often in the first 30 minutes of the last stage of REM sleep (normally in the early-morning hours). This newly discovered information was very important and set the stage to further understand lucid dreaming and direct many more experiments on the subject.

Knowing when a lucid dream is likely to occur along the typical sleep stages makes it easier to perform experiments in which the subject is told to signal the outside world or in which the experiment involves attempting to induce a lucid-dream state on the subject. The most important and obvious signal that we have is the REM cycle that occurs four or five times during a typical night of sleep. Because a lucid dream is most likely to occur during the last REM cycle, we can exploit this information and create various devices designed to detect this REM period by watching the subject's eyelids and then send some type of signal to the dreamer that it is time to become lucid. Of course, there is more to it than this, so let's have a look at what happens during a typical night of sleep in our gray matter.

There are five stages of sleep, each corresponding to a change in brain wave patterns. Brain wave patterns are measure by an ultrasensitive amplifier known as an *electroencephalograph* (EEG). These brain waves show our current state of awareness and change in five typical steps during a night of sleep. Figure 8-1 shows the stages of sleep as well as the occurrence of rapid-eye movements (REMs) through each stage.

The five stages of sleep are shown in Figure 8-2, with REM sleep being the fifth stage, occurring four to five times during a typical 8-hour sleep cycle. REM sleep in adults typically accounts for about 90 to 120 minutes of a night's sleep during an 8-hour period. REM periods are usually quite short at the beginning of the night, with the final one lasting the longest, where most lucid dreams also occur. Body temperature, heart rate, and breathing are quite irregular during REM sleep, which is another method of detection that can be used to trigger lucid-dream hardware. Of course, the most obvious method is to monitor eyelid movements because this is the most direct indication that a dreamer is experiencing REM sleep. During the various sleep stages, the brain frequencies change quite dramatically (as shown by an EEG), corresponding to different states of awareness. Figure 8-2 shows the various brain waves and their frequencies for each stage of sleep.

Notice in Figure 8-2 how the brain waves and their frequencies change during the five cycles of sleep. Alpha waves are the dominant brain waves while a person is awake and relaxed, which is why they are also present during the initial transition between restfulness and stage 1 sleep. As a person drifts deeper into stage 1 sleep, theta waves become the dominant brain waves. These waves also are associated with a trance-like state or are present when a person is doing some

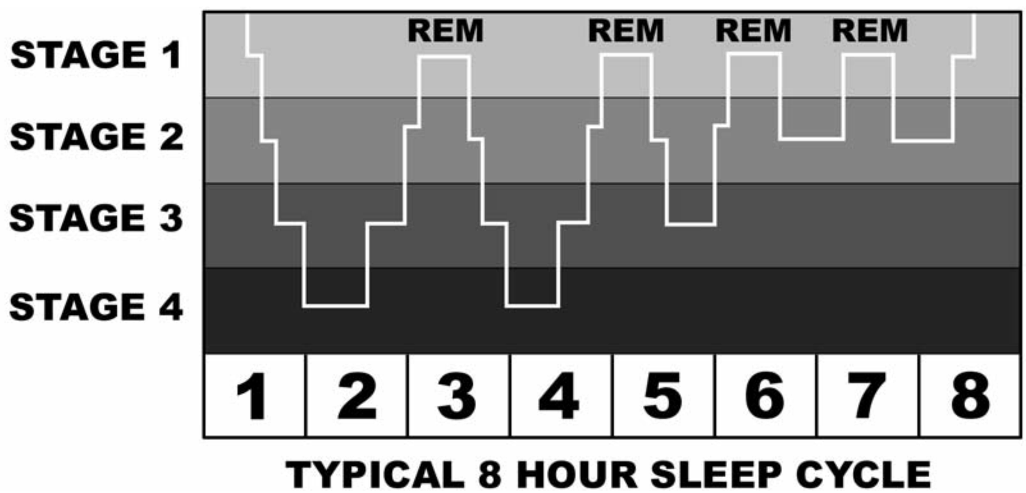


Figure 8-1 *REM frequency during the sleep stages.*

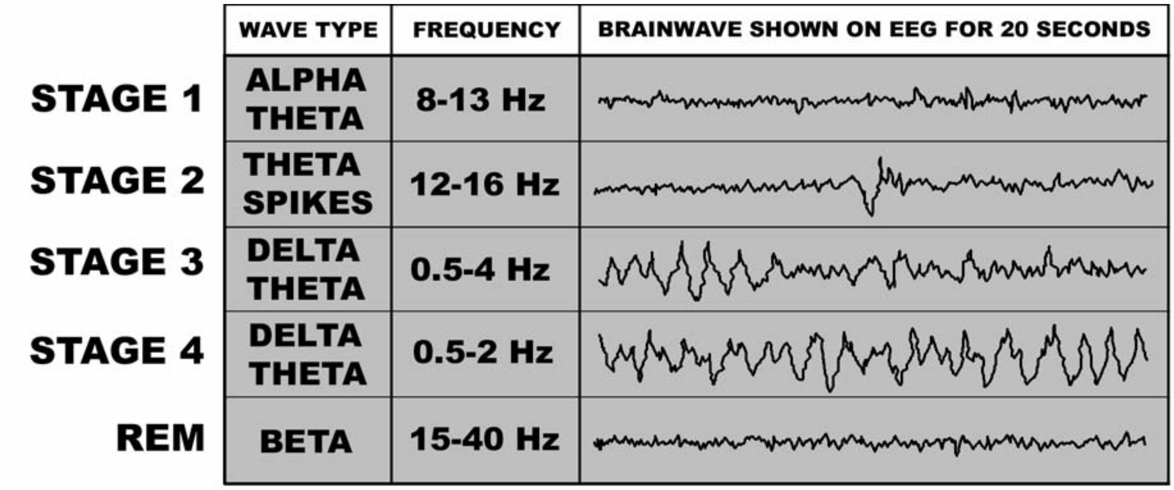


Figure 8-2 *Brain waves during the sleep stages.*

repetitive task where very little thinking is involved. Daydreaming or driving for long distances through a barren landscape most likely would put a person's brain into the theta-wave state. Moving into stage 2, theta waves become dominant and begin to spike and descend at random intervals. The third stage of sleep is considered the onset of deep sleep because theta waves and slower delta waves take over, and wave amplitude is at its highest. Stage 3 of the sleep cycle is where extreme nightmares, bedwetting, or talking out loud occur. Stage 4 sleep is the deepest sleep, with brain waves at their lowest frequency, and this stage happens right before the onset of REM sleep. In stage 4 sleep, more than half the waves are delta waves. In the REM stage of sleep, brain waves become beta waves, which are much the same as when a person is alert or working at some task. Beta waves are produced because the brain is alert and dreaming, yet the entire body is paralyzed with the exception of the eyes. It is thought that without this beta-sleep paralysis, the body might try to act out the dream. It is the REM stage of sleep we are most interested in because this is where you will fall into a lucid dream once you learn to do so.

If you are one of those people who has never experienced a fully lucid dream or don't think you dream very often, don't worry; anyone can learn to lucid dream, and it's likely that you simply forget your dreams as you wake up. Dream recall is the first key to opening the door to the land of lucid dreaming and involves nothing more than keeping a notepad or voice recorder handy so that you can make a short note the second you open your eyes. Have you ever had what you thought was a dreamless night only to have a dream recall triggered by something you saw in the real world that related to the dream? Without that trigger, you would have totally forgotten the dream, but with only something as simple as a visual or auditory cue, you get a total recall. The brain places dream memories in that space that is easily washed away because there really is no point in filling up your memory with random information. I can assure you that if you wake up and write down even one word pertaining to a dream, you will recall many of the details when you look at your note later on. Often, you won't even remember writing in your dream notepad because the brain is in a state that allows memory to slip away in a hurry. A fully lucid dream usually will not disappear from your memory like a typical random or semilucid dream because waking up from your adventure leaves you feeling energized and excited.

Regular sleep patterns also increase the likelihood of a lucid dream because you can use a dual alarm system to wake yourself an hour before your final wake-up time and then drift back to sleep thinking about how much you want to have a lucid dream. This simple technique, along with some of the projects presented later in this chapter, really can increase the number of lucid dreams you have once you learn to deal with them.

Much like riding a bicycle for the first time, your first lucid-dream encounter probably will be short and send you crashing into the land of wakefulness. The instant you realize that you can have total control, you may wake up because of the excitement, or you may not “trust” your dream world enough and scare yourself awake. I remember this happening when I first started studying lucid dreaming and was constantly trying to fly. The first few times I became aware, I woke right up feeling a huge sense of accomplishment and excitement, but that, of course, killed the dream right away. Once I learned to calm down and “get with the program,” I often would try things like jumping into the air or changing my surroundings, which also caused a fast awakening. Because your subconscious knows that flying through the air without wings or breathing underwater is something that obviously would be dangerous, your untrained lucid-dreaming mind often makes you feel unsure of the experience and causes an abrupt end to your adventure. In time, these things also can be overcome.

One of the great lucid-dream researchers of our time, Stephen LaBerge, a Ph.D. in psychophysiology, has written many good books on the subject and has come up with many useful techniques and even devices that can be used to enhance your lucid-dreaming experience. If you would like to expand your knowledge on the subject of lucid dreaming, just Google his name, and you will find more than enough information on his research, books, and inventions, some of which gave me ideas for a few of the projects in this chapter. Along with much groundbreaking research on lucid dreaming, LaBerge has come up with some very simple techniques for enhancing the chance of a lucid dream, as well as ways to prolong the experience.

One of LaBerge’s best-known methods for enhancing the chance of a lucid dream is call the *MILD technique*, which stands for *m*nemonic *i*nduction of *l*ucid *d*reaming and is basically a way of reminding yourself to have a lucid dream. This is done by waking yourself before your normal wake-up time and becoming fully alert by walking around for a few minutes, reminding yourself that you are now going back to a lucid dream. You then crawl back into bed, thinking about the dream you plan to have or recalling the one you were having just as you woke. You must *will* yourself to go back into a dream and become lucid. Surprisingly, this system may work better than most of the electronic devices used to attempt a lucid-dream induction, and it requires only a clock with two alarms or a pair of alarm clocks.

Another simple approach that will help to “train” your brain to recognize a dreaming state is by *testing reality*. Now, this may seem crazy, but often a dreamer simply will forget that it is not normal to be traveling through space in a ship made of cardboard boxes, or the dreamer simply will accept the fact that the dog now talks and drives a motorcycle. By testing reality while you are awake, you get into a pattern that will follow you into the dream world, and when it does, you will begin to realize that reality is bent. Reality testing may be as simple as holding up your hand every few hours to count your fingers because often you cannot do this in a dream or there are an odd number of fingers. You also may ask yourself routinely, “Is this a dream?” during the day because if you do this in an actual dream, your mind will notice things that don’t belong. The key to reality testing is repetition, so I have included in this chapter a simple reality tester that will send you a reminder once every hour or so to perform a simple reality test. When you get used to the device, it will begin to appear in your dreams and could be the key to realizing you are in the "other world."

Some of the other techniques that will be explored as projects in this chapter involve presenting some external stimulus to the dreamer either at opportune times based on sleep patterns or via feedback directly from the body. Using audio to direct a dream is interesting because your mind is often aware of what you hear while dreaming, and if you have ever fallen asleep while watching a movie, you might have had the experience of remembering a dream that had to do with whatever you were listening to at the time. Mixed with lucid dreaming, this approach can become very interesting, especially if you create some type of audio dreamscape and use your own voice to "coach" your dreaming self into a new lucid-dream world. One of the projects in this chapter also will respond to movements of your eyelids, so you can use this feedback in conjunction with a clock to determine when you are in the last REM stage of sleep before waking because this is the opportune time to introduce some type of audio or visual cue to help trigger a lucid dream.

Well, that's enough hype about lucid dreaming for now! Honestly, who wouldn't want to have a virtual reality without limits to control every night (or at least once in a while)? No computer game or thrill-seeking adventure could ever compare to a lucid dream because there are dangers, moral issues, and of course, the very fabric of reality to contend with. When you wake up from a full-blown lucid dream, it won't feel the same as when you wake up from a typical strange dream and just think, "Huh?" Your lucid-dream experience will seem so real at the time that when you finally open your eyes, it will seem like it really happened, and you will remember the experience much like any incredible event in your life. Chances are that you won't be able do lucid dreaming often, so don't worry; it's not like you are going to get addicted to the sleeping world and leave your waking life behind!

Now let's dig into the electronics junk box and put our "evil genius" minds to work so that we can master the other half of our lives—the dream world.

This page intentionally left blank

Project Nine

Waking-Reality Tester

Now here is a device you probably thought you would never need—a little box that helps you to determine if you are awake or dreaming. This may sound like a completely useless gadget, but when it comes to your lucid-dream arsenal, this is actually a very useful tool that can greatly increase your chances and the frequency at which you might have a lucid dream. By itself, the device serves no real function besides being a simple timer that reminds you to practice one of the best-known techniques used to aid in lucid dreaming. Reality testing on a regular basis becomes a habit that is often transferred into your dreaming world, where, of course, reality is much different from that of the waking world. When you ask yourself, "Is this real?" in a dream, often you realize that you are in fact dreaming and then can will yourself into a fully lucid state. This system only works if practiced regularly, so this little timer helps you to get into the habit of testing your current state of reality in order to bring this habit into your dream world.

The reality tester is a covert device that works much like a pager, so you can carry it at all times without having to explain your "weirdness" to those around you. Since the reality tester simply vibrates every hour or so, you can continuously test your state of consciousness without interrupting your day. Eventually, you will begin to do this in your dreams, and when you look around and ask yourself, "Is this the real world?" you will often realize that things are just too strange to be real and then can become lucid. A tried-and-true method of reality testing is to simply count your fingers because often in a dream there will be too many, too few, or you simply will not be able to do it at all. Reading some text is also a good method of reality testing because the dream world often will return scrambled text or something that looks more like ancient hieroglyphics. In a dream, some things will just be plain ridiculous, such as a purple sky or the fact that you are eating your breakfast on a pirate ship made of Lego blocks. In this case, there is no doubt you are dreaming, but you may not even question these oddities unless you are reminded to do so by the reality tester. Once you get in the habit of realizing your shift in reality, you can begin to take control over your dream world and rearrange it as you see fit.

The reality test can be made to vibrate covertly, or it can be adapted to make an audio tone (or both if you like). The main schematic for the reality tester is shown in Figure 9-1 with a small dc motor used as a vibration device.

The 555 timer (IC1) is wired as an adjustable oscillator that sends its pulses to a 74HC4040 14-bit binary counter (IC2) so that the pulse time

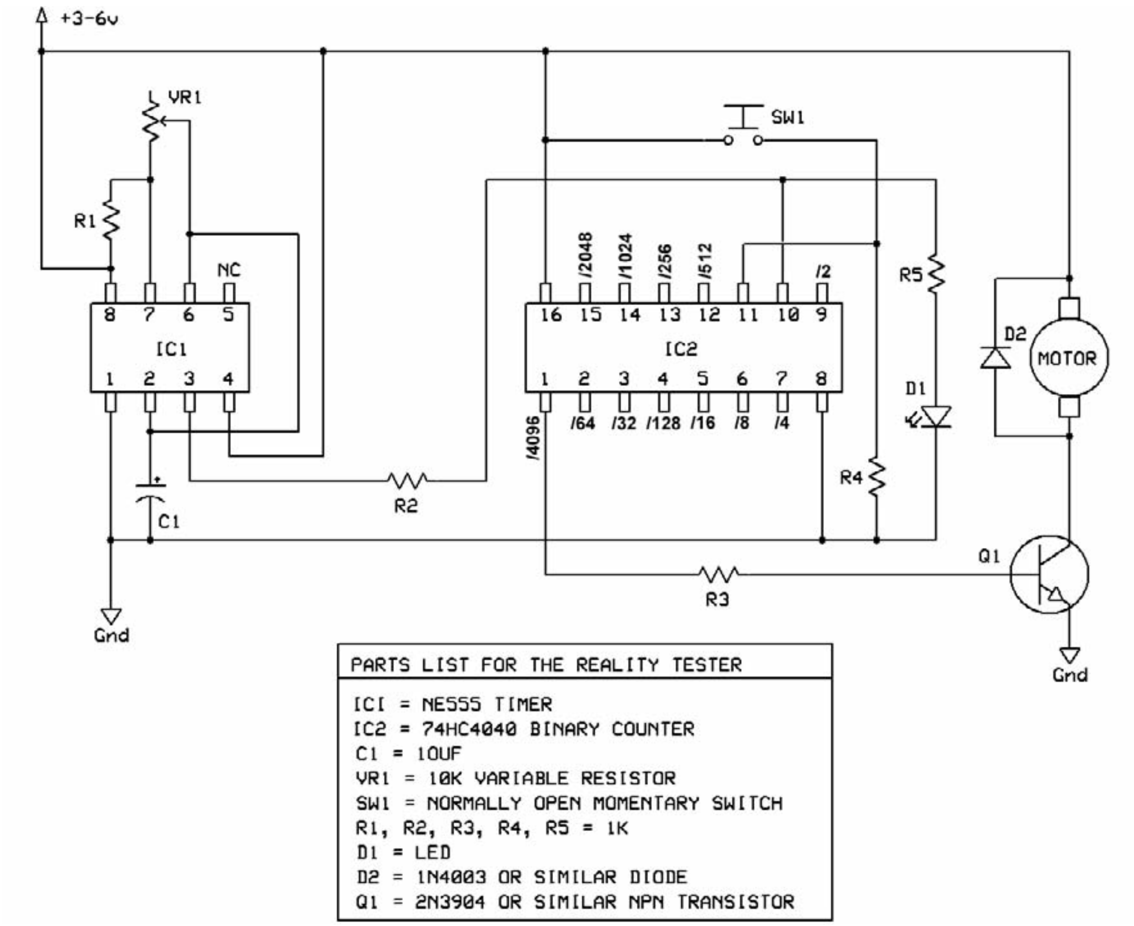


Figure 9-1 *Reality-tester schematic with vibration motor.*

can be divided by one of the following values: 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024, 2048, or 4096. Using the 4040 to divide the pulse time makes it much easier to set up a timer based on minutes or hours because the 555 timer is not very good at outputting long delays. In the schematic, R5 and D1 are used only during setup so that you can use a clock or stopwatch to count the timer pulses and then adjust VR1 to get the light-emitting diode (LED) to flash at about once per second. Once you have a 1-second flash rate, just choose one of the outputs from the 4040 counter to set the interval at which the reality tester will buzz you. In the schematic shown in Figure 9-1, I have chosen pin 1 of the 4040 counter, which is marked 4096, so my reality tester will alert me about once every hour if I set the 555 timer to a pulse rate of about 1 second. To calculate your final time, use the formula

4096/60 (seconds)/60 (minutes) = 1.13 hours

Transistor Q1 is being used as a switch to turn on a motor that operates as a vibration device because it has an unbalanced counterweight on its shaft (more on this later). Switch SW1 is a pushbutton switch that forces you to reset the unit to acknowledge the fact that it is time to test your reality. The reality tester also can be made to output an audio tone by removing the motor, Q1, and R3 and replacing them with the circuit shown in Figure 9-2. As you can see, this circuit is much like the 555 timer clock oscillator in the main schematic except that it oscillates at a higher frequency and then sends its output into a piezo buzzer.

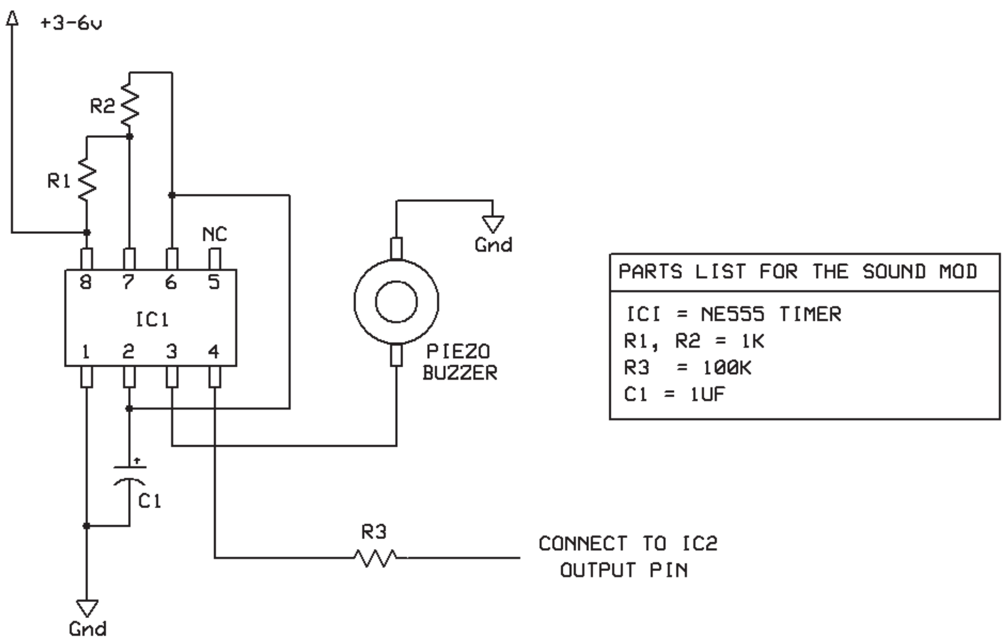


Figure 9-2 *Reality tester audio output modification.*

You could have both the vibrating motor and audio at the same time or add a switch to change between audio and vibrate modes. Having both modes of operation might be good because it is more convenient to use the audio alert mode when working alone at a desk and less embarrassing to use the covert buzz mode when out and about. When you pull a beeping home-built gadget out of your pocket at dinner and try to explain to people that it helps you keep track of your current reality, that might get you some very strange looks!

Now let's work on the mechanical part of the project that will create a vibration very similar to a cell phone or pager. Figure 9-3 shows a number of small dc motors that can be used to create a vibration unit, as well as a few that are actually taken from pagers and cell phone. The two small motors shown at the bottom left of the figure are actual vibration motors, as you can see by the unbalanced counterweights already attached to their output shafts.

Don't worry about tracking down a pager motor because it is extremely easy to make the same kind of thing with any small dc motor and a soldering iron. All you need is a small dc motor (the smaller, the better) and a blob of solder stuck to the end that will cause the motor to vibrate as it spins. These motors can be found in all kinds of kids' toys, CD-ROM or DVD players, video cameras, electric shavers, and many other small appliances. Don't worry about the voltage of the motor, just that it spins up when you place it on your 3- or 6-V battery. Even a motor rated for 24 V probably will spin fast enough on 3 V to create a vibration, but you really don't need a huge motor because the device should be small enough to fit in your pocket when completed. To add an unbalanced counterweight to your motor's shaft, just heat up a blob of solder, as shown in Figure 9-4, and then dip the motor shaft into it and hold it there until the solder solidifies.

As the solder blob cools, try to keep the motor shaft perfectly still so that the blob stays tight on the shaft. When the solder has cooled, add power to your motor and see how it vibrates. Chances are good that the solder blob will not be balanced perfectly on the shaft and that there will be a lot of vibration as the motor spins. Another method of making a counterweight is to find a gear or

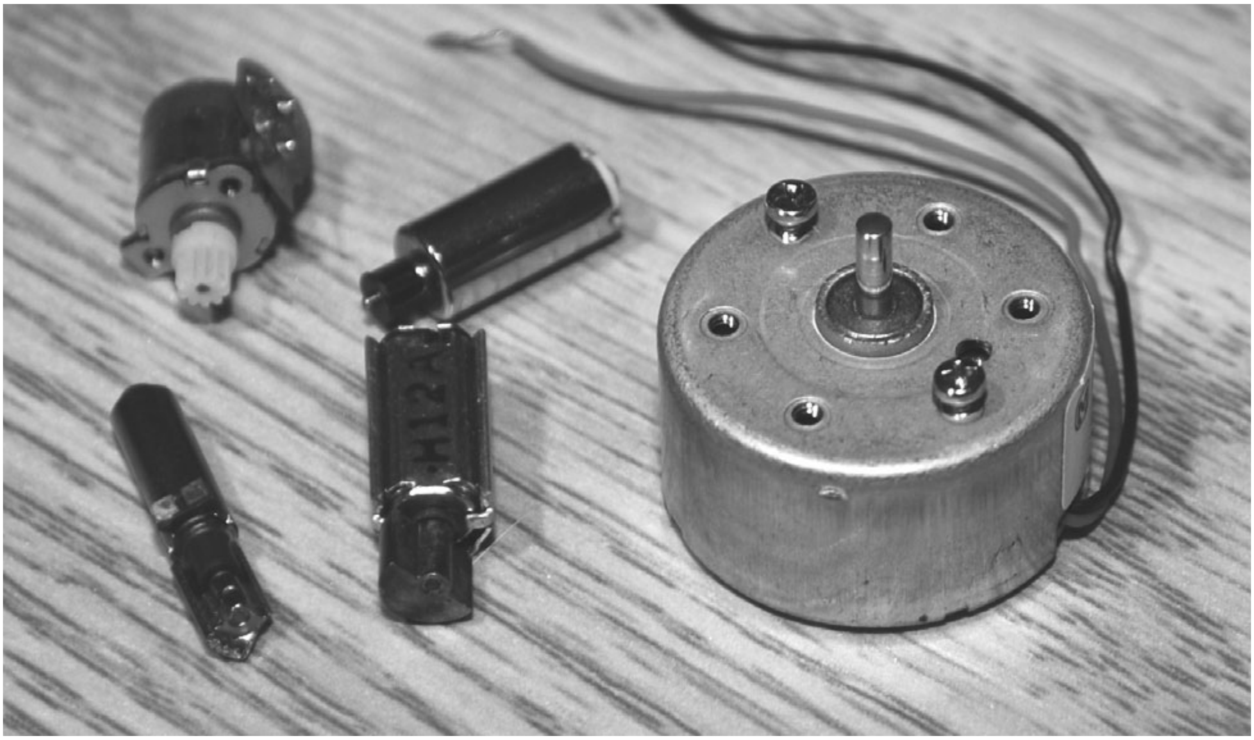


Figure 9-3 *Small dc motors that can make vibrations.*

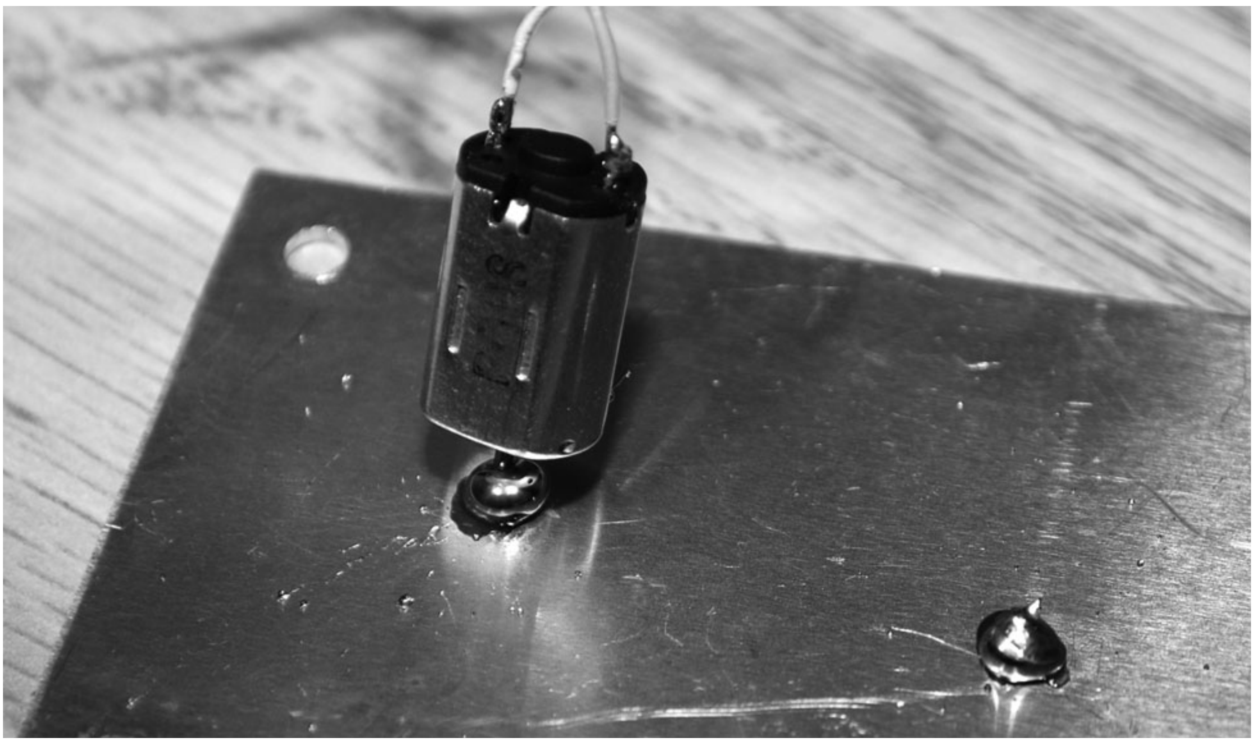


Figure 9-4 *Adding an unbalanced weight to the motor.*

Figure 9-5 *Motor with counterweight attached.*

wheel that fits onto the shaft and file one side away. You will need to solder wires to the power pins on your motor, as shown in Figure 9-5, so that you can connect it to your breadboard and then to your completed circuit.

The breadboarded circuit is shown in Figure 9-6, and you can get a perspective on how small my vibration motor really is compared with the other parts. The 3-V lithium battery will power the 5-V logic chip and timer without any problem and give plenty of juice to the motor to make it vibrate.

The circuit should be built first on a solderless breadboard so that you can adjust the timer properly using the temporary flashing LED (shown by the arrows in Figure 9-6). By adjusting the variable resistor (VR1), you can set the flash rate to about once per second and then "tap" into the 4040 counter to set the overall delay between reminders. Just take the divider number (shown on the pins of the 4040 counter in Figure 9-1) and divide that number by 3600 to get the approximate value in hours between reminders. Once you are satisfied with the time between reminders, you can remove R5 and the LED so that the LED doesn't waste battery power. You also can measure the resistance across the variable resistor once it is removed from your circuit and then replace it with a fixed-value resistor of the same value. Once the circuit is working properly, the components can be transferred to a small perf board for installation into a portable cabinet.

As shown in Figure 9-7, I also added an on/off switch to the system so that battery power could

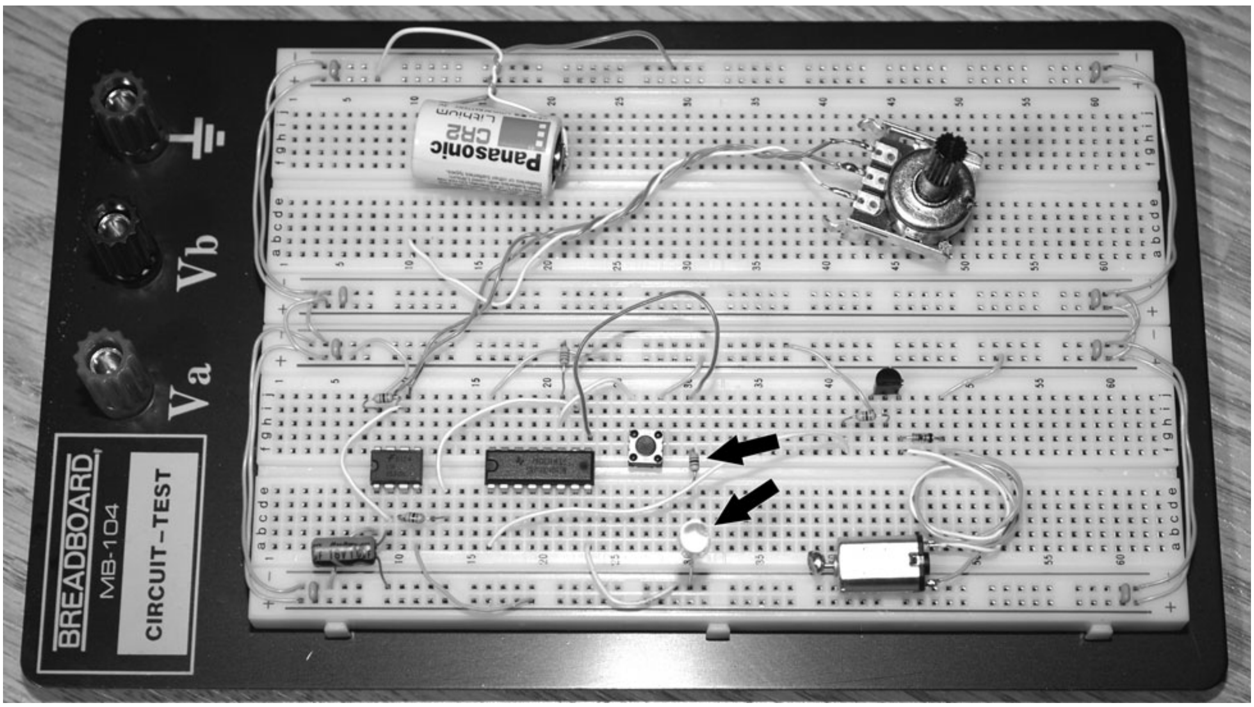


Figure 9-6 *Testing the circuit on a breadboard.*

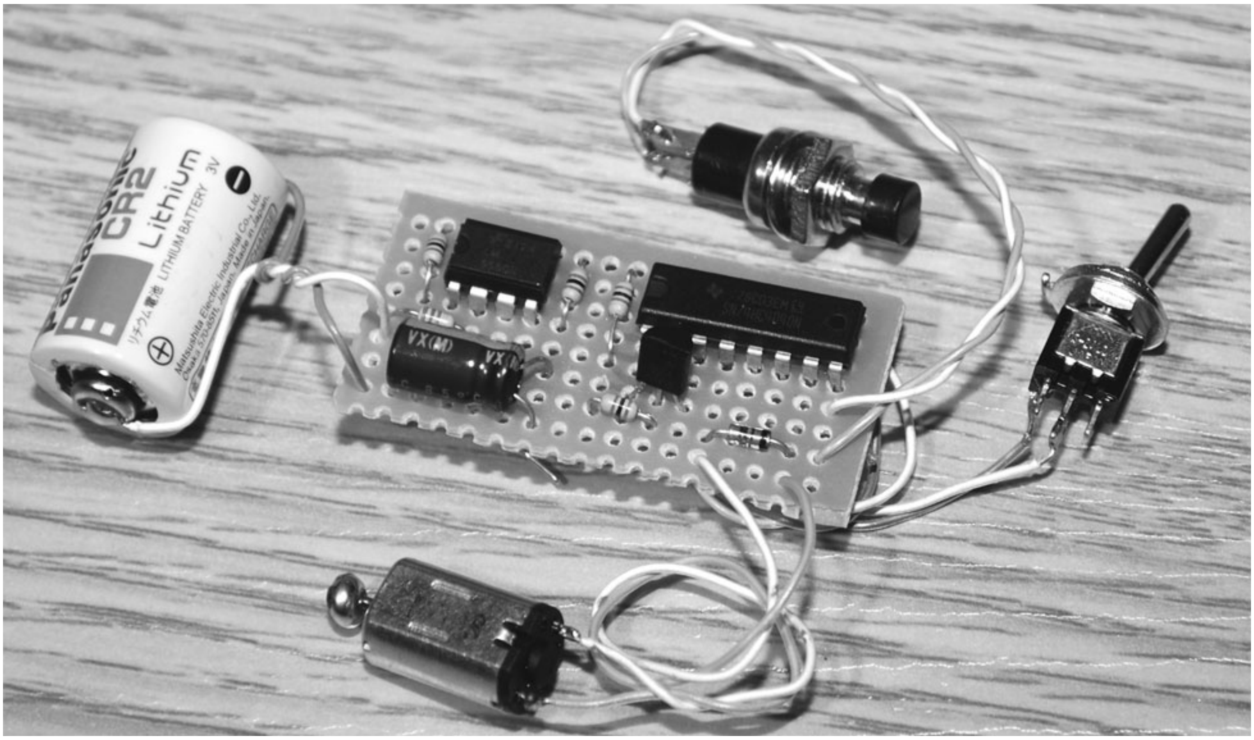


Figure 9-7 *A completed circuit on a perf board.*

be conserved when not busy testing reality. Now all you have to do is find the smallest possible cabinet for your reality tester, and you can begin training your mind to regularly question your current reality. I found a plastic film container that was just perfect to fit the small battery, motor, and circuit board inside, and it fits nicely in a pocket without becoming too annoying. Figure 9-8 shows the tight fit of all the components into the film container, with just enough room for the reset button out the top.

Before I jammed the works into the film container, I wrapped the circuit board with tape so that the battery case and on/off switch would not short out against the component pins. This is not shown in Figure 9-8 because it looked like a mess, but you should consider wrapping your perf board in tape if it is to be jammed into a small container such as this. I also decided to place the on/off switch in the container because I did not want it turned off accidentally when I was carrying the unit around. The reset switch is not a problem because it only works once the buzzer or vibration has already begun. Now you have the lengthy process of testing the unit with it fully assembled. It is painful to wait for the entire cycle, so find something to do and then flip the switch, keeping track of the time you turned on the unit. If you were shooting for an hour delay, then your buzzer or vibration should begin somewhere close to that time, give or take 15 minutes for timer inaccuracies.

The completed reality tester is shown in Figure 9-9, ready to carry around. Of course, I do not recommend that you attempt to bring this device on a plane with you because it surely looks like something dangerous, and trying to explain its purpose to security folks might make you look like a crazy person! Every time you are alerted by the reality tester, press the reset button and then ask yourself, “Is this a dream?” Look around and try to find some text to read, or hold out your hand and count your fingers. In the dreaming world, text almost always will be unreadable, and your fingers will not add up correctly or just look strange somehow. It may take a week or more of

Figure 9-8 *Everything fits into a small film container.*

Figure 9-9 *The reality tester ready for action.*

constant reality checking before this habit spills over into your dreams, but once it does, it will be easy to trigger a lucid-dream state by realizing that you are actually dreaming. You also can expect a few misfires as well as you get excited and wake up too quickly at the prospect of actually controlling the dream. Like all things worthwhile, lucid dreaming is an art that must be practiced to be fully appreciated.

Once you've managed to learn the art of realizing that you are dreaming, you might want to try some other experiments on your sleeping brain. The next project will help you to "direct" your dreams by audio suggestions presented at opportune times during your sleep cycle. Some things that you might want to try are sleep learning, subliminal suggestions, movie recordings, random sounds, and motivational material.

Project Ten

Audio Dream Director

Have you ever fallen asleep while watching a movie, only to awaken a short time later and recall a dream that had something to do with what you were watching? How about waking up early because of some neighborhood distraction, only to realize that you were dreaming about something that was making a very similar sound? Much like a waking mind can react to subliminal messages, the dreaming mind also can be influenced by the senses, sometimes to a greater degree. To exploit this fact, we will build a device that will send you an audio message at the opportune time during your REM sleep phase (about an hour before you wake). This is the time when you are most likely to enter a lucid dream and the best time to remember a dream or make notes for later recall.

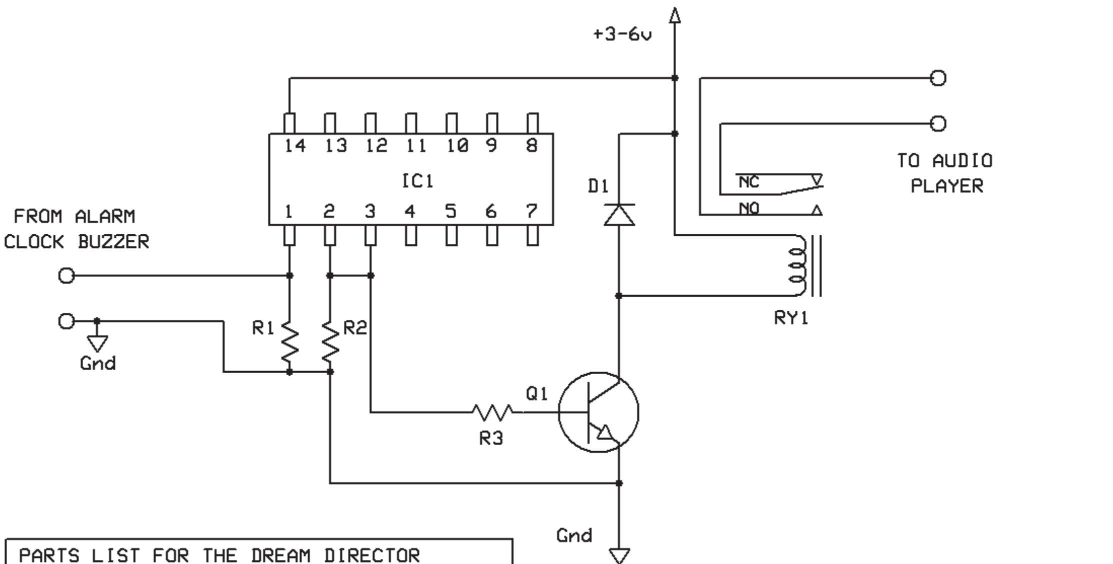
The idea behind this device is quite simple. A hacked alarm clock is modified to trigger some type of audio player that will send you a prerecorded audio file to help you enter a lucid dream or simply "direct" your dreamscape in some direction. There is no direct feedback from the body in this version of the device, but since we know that the last hour of sleep is usually the most important for dream recall and lucid dreaming, the trigger is simply set to an hour before your real wake-up time. You can then prerecord 20 to 40 minutes of audio for playback into a pillow speaker and let your mind wander in the direction of the audio clip. Maybe you want to learn Spanish while you sleep? How about recording your favorite movie and trying to live out the action? Maybe you just want to record yourself speaking about how tonight you will enter a lucid dream? Chances are the audio will affect your dream, although it may do so in ways you did not expect.

When I first experimented with this device, I found that the audio certainly did affect my dreams, but sometimes in ways I was not expecting. I once recorded the unmistakable sounds of a car race, thinking that I would wake from a dream where I was sitting behind the wheel of an Indy car. When I did wake and had dream recall, I remember that the dream was about some weird tiny flying robot, and the car engine sounds actually were the sounds of the little wings flapping as the robots flew past my ears. Although this was not what I expected, it certainly had an effect on my dreams. I also tried my own voice talking to me like some guru about how I was entering a lucid dream, and this worked a few times. On other occasions, I had dreams about making the actual audio file!

The dream director is just an amplified switch that locks down a relay when there is a small change in voltage at the input. The input typically is the output buzzer from an alarm clock. The relay switch then is connected to the play button on some type of audio playback device such as an MP3 player, handheld audio recorder, or even a CD player. Since the actual alarm speaker or buzzer is removed from the hacked clock, you do not wake, and the audio file begins to play through a pillow speaker under your head. Using any computer sound program, you can compile your audio file to slowly ramp up the volume so that you are not awakened suddenly by the sound. If all goes well, your dreaming mind will drift into a dreamscape influenced by the audio file. Because you may remember making the file or have your voice dubbed in to remind you that this is a lucid dream, chances are good that something interesting will happen.

Figure 10-1 shows how simple the relay-trigger part of the dream director really is. IC1 is an OR gate connected in such a way that its own output feeds back into its input and latches it in the ON position once triggered by an external voltage change from your alarm system. The output then is fed into transistor Q1 to drive the relay coil and close the contacts. Using a relay isolates your device from the rest of the circuit and makes it easy to hack just about any audio player with a play button that will start the audio playing. The relay also adds complete isolation from the alarm clock, although I still would recommend that you only use battery-powered devices when "jacking" yourself up at night to your hacked gadgets. Just think about lightning storms, and you will realize why not being plugged into the wall socket at night is a good idea.

You can run the circuit on from 3 to 6 V, and just about any relay with a voltage rating from 5 to 12 V will work. Even some relays rated at 24 V seemed to latch just fine on my system when I was running it on only 3 V. The best way to find a suitable relay is to try activating it with whatever power source you intend to use here. You do not need a large relay because current is very minimal, and your relay needs only a single



PARTS LIST FOR THE DREAM DIRECTOR

IC1 = 74LS32 OR 74HC32 OR GATE
R1, R2 = 100K
R3 = 1K
Q1 = 2N3904 OR SIMILAR NPN TRANSISTOR
RY1 = SINGLE POLE SINGLE THROW RELAY
D1 = 1N4003 OR SIMILAR DIODE

Figure 10-1 *The dream director schematic.*

pole with a pair of normally open contacts. Some of the relays I tested on my 5-V battery are shown in Figure 10-2, so I chose the smallest one, which was taken from an old modem card and rated for 5 V.

This circuit is so simple that you may not even need to test it on a breadboard, but it is certainly easier to make changes this way in case something does not work the way you want it to. Figure 10-3 shows the relay driver ready for testing once I rip the buzzer from my alarm clock and feed it into the logic gate. The reason the circuit latched the relay closed is because once your alarm clock starts feeding the input, it will do so intermittently, which may cause some audio players simply to start playing the same audio file over and over again. By latching the relay, the play signal is sent to your audio device only once.

The completed relay driver is shown built on a small perf board in Figure 10-4. Notice the 5-V regulator installed in the center of the board so that I can run the driver from a 9-V battery. This 7805 regulator is shown in Figure 2-8 and will output a perfect 5 V from a supply of 9 to 12 V. Of course, 3 to 6 V would have been just fine for the logic chip, but 9 V most likely would cook it in a hurry, so the regulator is necessary. When the relay driver is working, any small change in the input voltage will cause the relay to close and not open until the power is removed. To test the board, just connect a 1.5- or 3-V battery to the input wires and listen for the relay to close.

The alarm clock that will trigger the relay board can be just about any type that has a digital alarm connected to a speaker or buzzer with two wires. Figure 10-5 shows two of the clocks I tested, and both worked perfectly with the relay board after removing their buzzer output wires and feeding them to the relay-board input. A battery-operated clock is certainly more preferable than one that will connect your sleep laboratory to the power lines, so take this into consideration when looking for a clock to hack.

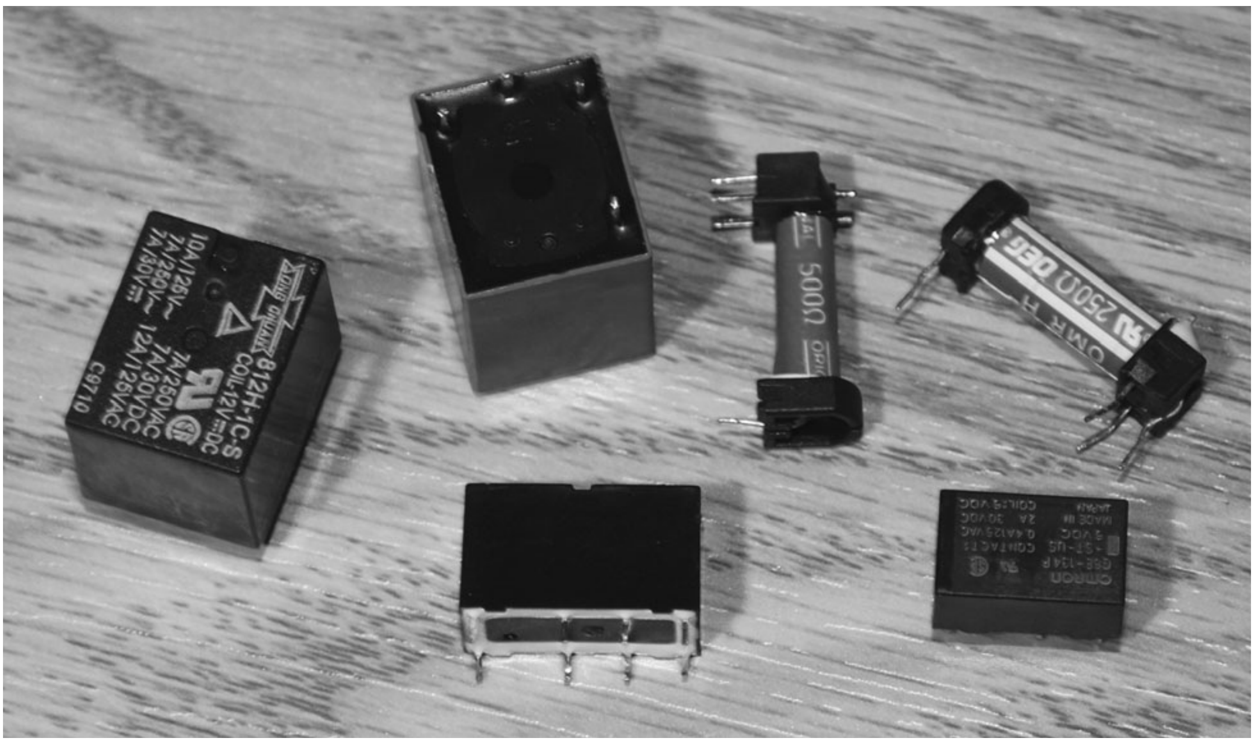


Figure 10-2 *Some relays that will work with this circuit.*

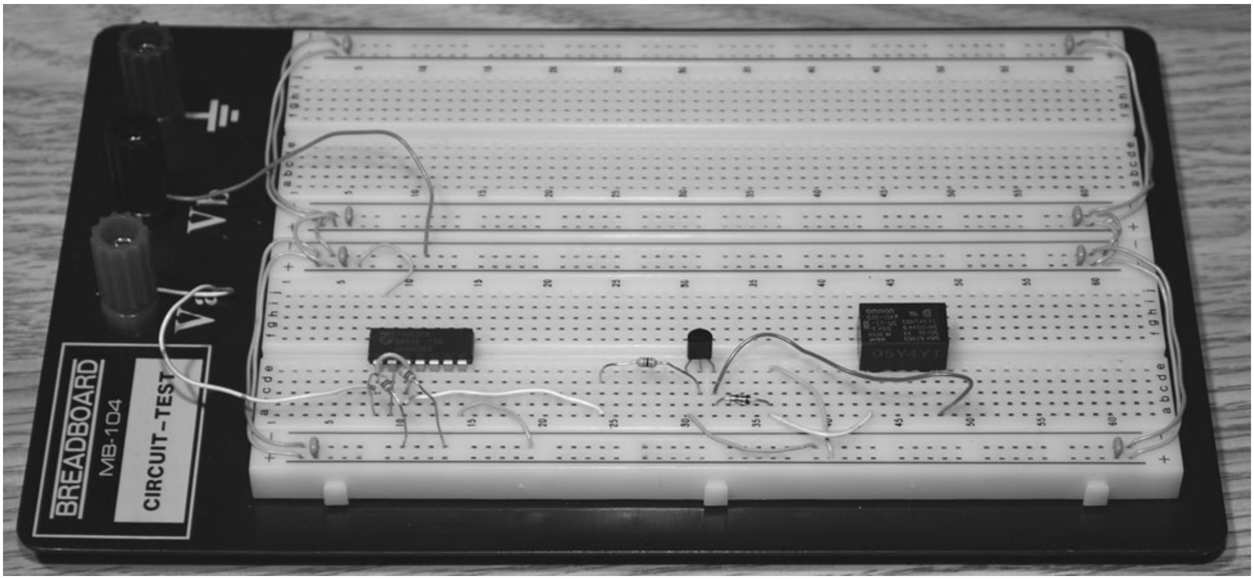


Figure 10-3 *Testing the relay driver on a breadboard.*

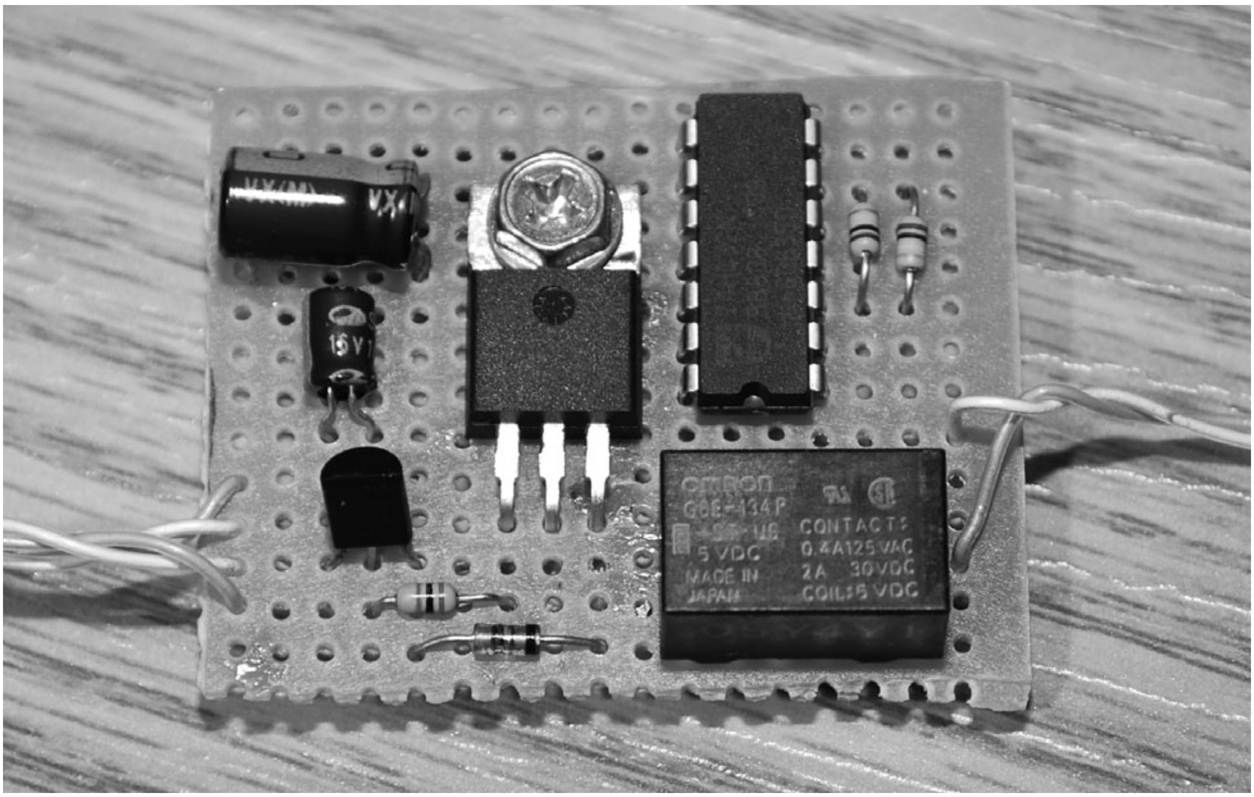


Figure 10-4 *Relay drive board ready for installation.*

Figure 10-5 *Choosing an alarm clock to drive the relay board.*

To trigger the relay board, there needs to be a small voltage change at the input of the logic gate, and this is taken from the output of the alarm clock. You need to identify the speaker, buzzer, or piezo element and remove the two wires from it so that you can install some type of output jack onto the alarm clock's cabinet. Since you don't want the alarm to wake you up, the buzzer must be completely removed from the circuit so that any alarm sounds will be fed silently directly into your relay-board trigger input. Figure 10-6 shows the piezo element found inside the small alarm clock that I pulled apart for this experiment. The two arrows point to the alarm output wires, and

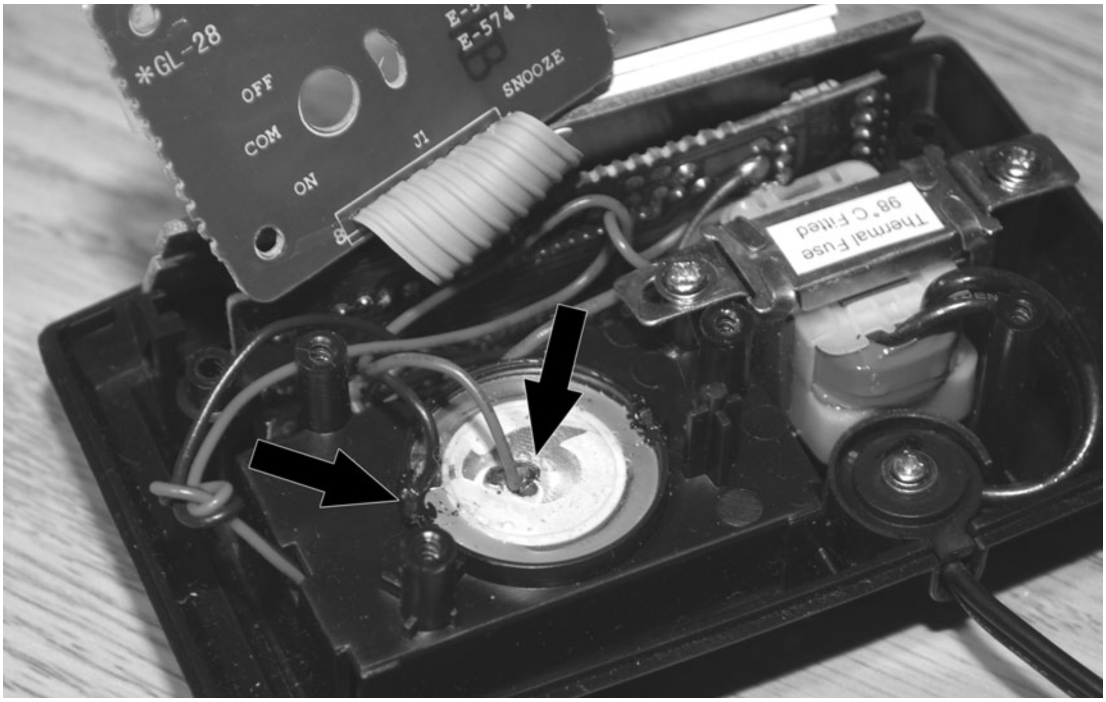


Figure 10-6 *Identifying the alarm output wires.*

there is a positive (red) and negative (black) wire in this unit, which is important because the input to the gate should be a positive-going pulse. Don't worry if your wires are the same color and there is no clear indication as to which one is positive because you can just try them both ways until you hear the relay close when the alarm is going off.

Once you have identified the alarm output wires, connect them to some kind of jack that can be installed in the alarm clock cabinet so that you can plug your relay board or some other experiment into your modified alarm system. Figure 10-7 shows the ⅛-inch mono jack I decided to use for this device, keeping in mind the positive wire going to the center pin on the jack. Again, if you are unsure about the polarity, just guess, and then reverse the wires if the relay fails to close when the alarm is going off. Another idea for testing is to remove the original alarm buzzer or speaker so that you can plug it back into the newly installed output jack. This gives you the ability to hear the alarm for testing purposes or use the clock normally when not experimenting with it.

The newly modified "silent alarm" system is shown in Figure 10-8 with the alarm output jack installed in the side of the cabinet. This little hack is great for a variety of experiments that might require an accurate timer with an output capable of driving a digital circuit or relay into action. By using this alarm set to trigger your dream experiments an hour before your real alarm sounds, you are almost guaranteed to be in the last stage of REM sleep, where lucid dreaming is most likely to occur. Also shown in the figure is the matching male ⅛-inch mono plug that will transfer the alarm voltage into the relay-board trigger input. Now you need to install your relay-board trigger in a cabinet or attempt to jam it into the alarm clock cabinet and leach power from the clock.

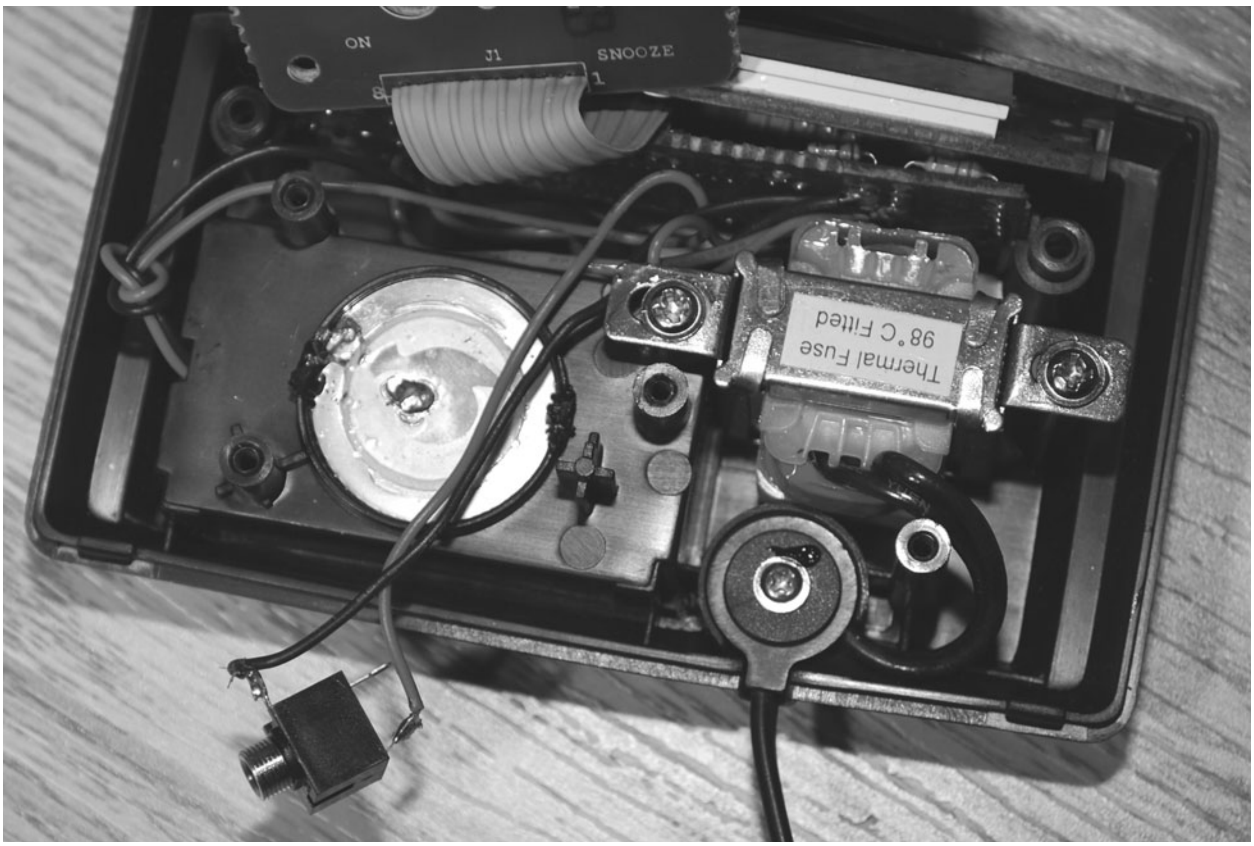


Figure 10-7 *Adding an alarm output jack.*

Figure 10-8 *A universal "silent alarm" for your projects.*

I actually had room in my alarm clock for the relay board and found a 12-V dc output from the power supply, so I could have installed everything right in the clock and fed the 12-V supply into the 5-V regulator in the relay board. Of course, this makes the unit unavailable for other experimentation, so I opted for the typical "black box" installation shown in Figure 10-9, making sure that there was also room for a 9-V battery. I also added an LED to show that the unit was on

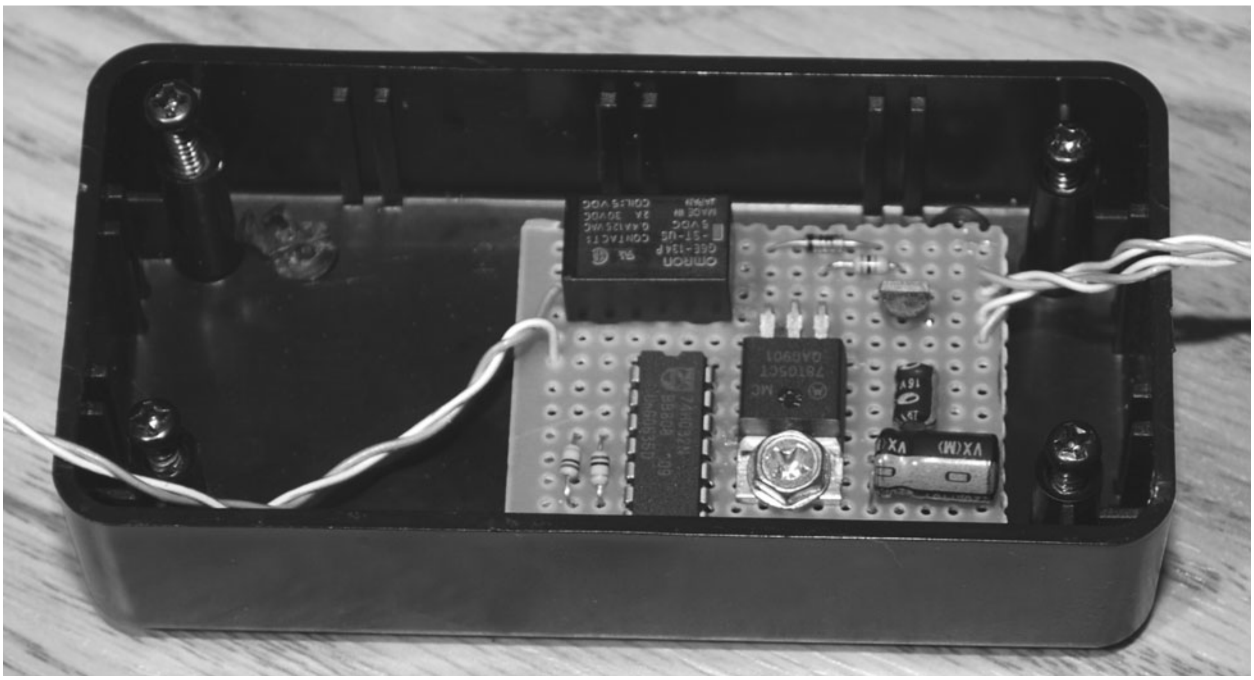


Figure 10-9 *Installing the relay board in a cabinet.*

and a switch to turn it off when not in use. Because the device is on all night, I used a high-value resistor to keep the LED as dim as possible to reduce drain on the battery. A 47K resistor made the LED just barely visible in a dark room.

Figure 10-10 shows the completed relay board and modified alarm clock ready to trigger just about any device through the relay. Because the relay's mechanical contacts carry no voltage and are completely isolated from the rest of the circuit, you are protected somewhat from the alternating-current (ac) supply if you decide to use a plug-in clock, and your external devices are also safe from the circuit if it were to fail because the relay is nothing but a mechanical switch. The RCA-style jack on the top of the box is the direct connection to the relay contact switch so that I can plug devices into the box. I wanted a different jack than the one used to input the alarm so that I wouldn't accidentally mix them up and possibly harm the external devices I plan to plug into the unit.

With your alarm feeding the relay board, all you need to do is connect the relay switch to the play button of some audio device so that it will start playing your prerecorded dream messages when you are in REM sleep. You can hack practically any audio device that will start playing by pressing a single button, so inexpensive personal audio recorders are perfect for the job. I will show you how to hack a digital recorder such as the one shown in Figure 10-11, as well as the old-style mechanical tape-based units that are also quite easy to find and modify. When choosing an audio player, just make sure that it will stay on and respond to a single button press to begin playing your message. Some units have a sleep mode, but often this can be turned off as an option. If your unit will not respond to a single play button, then the device will not work with your relay board. Mechanical tape players always will work because the relay board will control the voltage to the motor while the play button is stuck in the play position. MP3 players usually work as long as they start with a single button press. Usually, the cheaper and larger the unit, the better it will be for hacking.

Figure 10-10 *Alarm clock and relay board ready to use.*

Figure 10-11 *A digital voice recorder is perfect for this project.*

To connect the audio device to your relay board, all you need to do is open the case, find the play-button points, and add two wires back to your relay contacts. Polarity is not important because the relay acts exactly like the original switch on the audio device, and if you have room, you can feed the wires out of the case and not even interfere with the operation of the play button. Figure 10-12 shows how I was able to tap into the play button and sneak the thin wire outside the unit without messing up the original play-button functionality. If you are hacking a very small MP3 player, then you might want to have a magnifying glass handy and use the smallest wire you can find because the contact points will be very small. If you don't care about using the audio device normally, then you can make an ugly hack if necessary, but remember

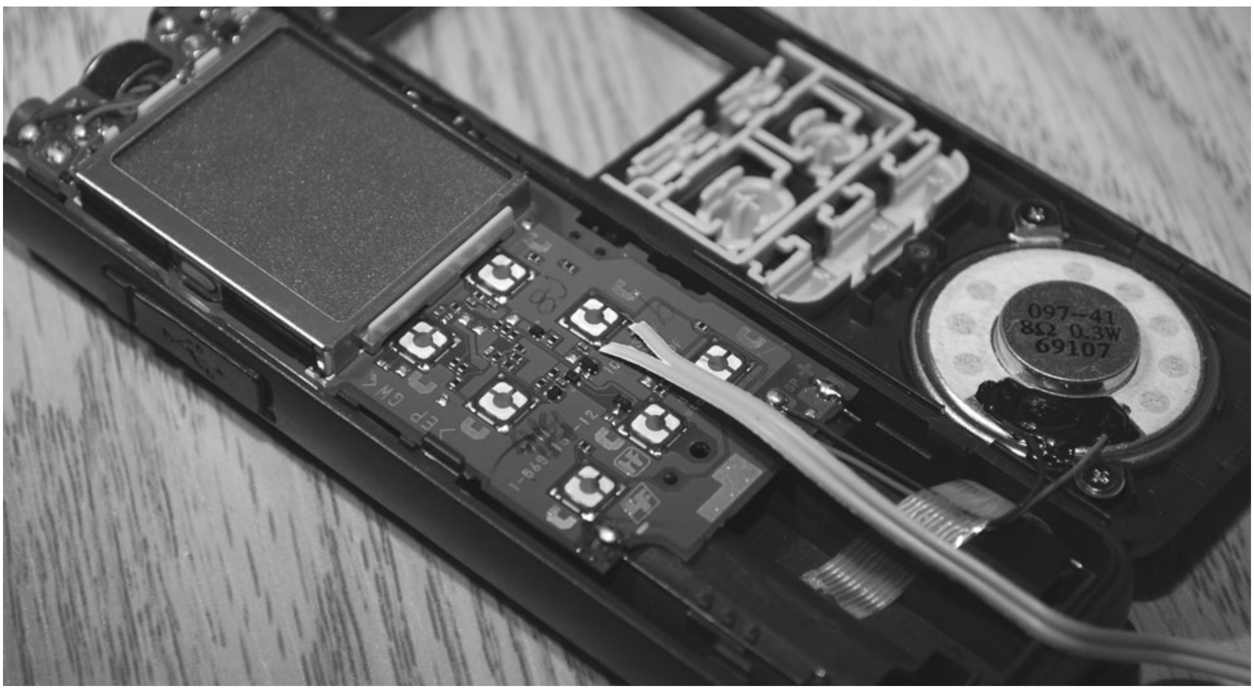


Figure 10-12 *Hacking into the play button.*

that the device will have to be somewhat operational to get your messages into the memory.

Mechanical tape players are very easy to hack and are always guaranteed to work with the relay board because the drive-motor connection will be severed and routed through the relay switch. Open the unit and identify the motor, as shown by the arrow in Figure 10-13. The motor will have only two wires coming from it, and you only need to cut one of them and then feed a pair of wires from each end back to the relay switch. Now you can press play on the recorder and nothing will happen until your relay closes and starts the motor spinning—it's an easy hack! If there is a jack on your recorder with the label "REM" on it, then you are in luck because that's exactly what this jack does: It breaks the connection to the motor so that you can just find a compatible plug and send that directly to your relay switch. Mechanical recorders are nice because they can be found at many stores relatively cheaply and are always easy to hack.

Sound quality is not important here, so you probably can use the same tape over and over again without any problems.

Figure 10-14 shows the completed hack done to the mechanical tape player in order to feed the motor output back into the relay board. I also took this opportunity to add a connection from the original speaker out to another jack so that I could use a pillow speaker rather than placing the device under my pillow directly. This modification reduces the risk of pressing stop or some other button during sleep. Most players already have a headphone jack, so you only have to make the audio-output modification if there is no headphone or external speaker plug on your unit.

Once your relay board is triggering your audio device, you are almost ready to experiment with audio dream induction. The completed unit is shown in Figure 10-15, ready with my favorite movie scenes to influence my dreams. The only question left is, How do you get the audio to your ears in a comfortable way without waking

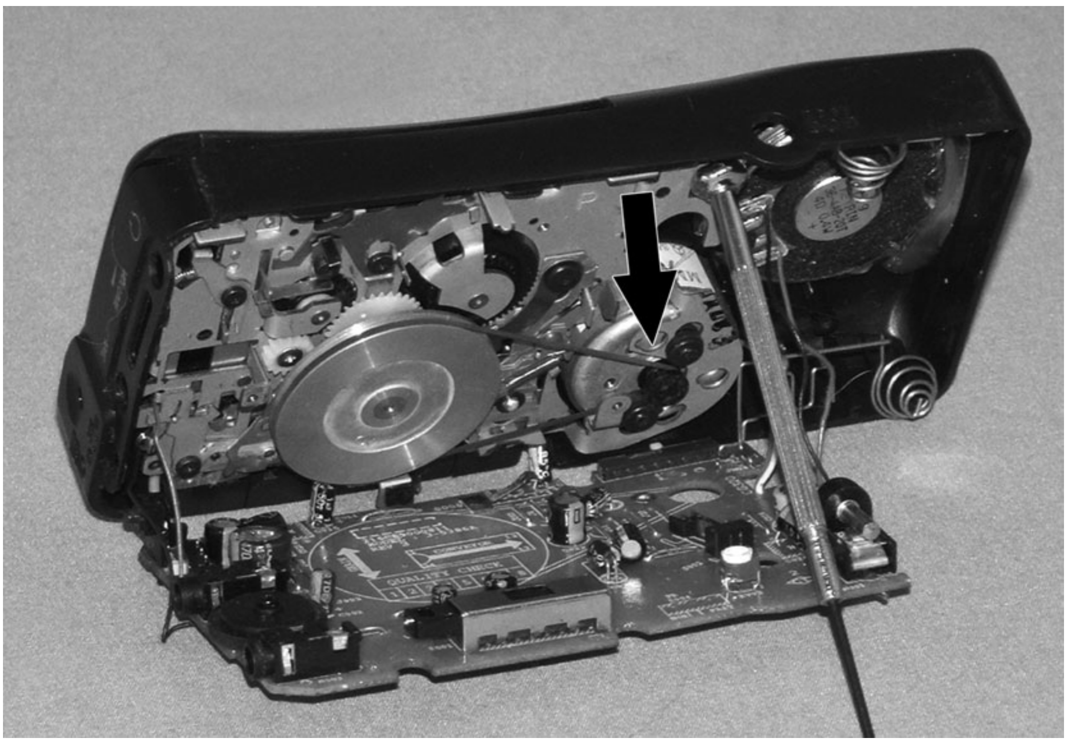


Figure 10-13 *Hacking a mechanical tape player.*

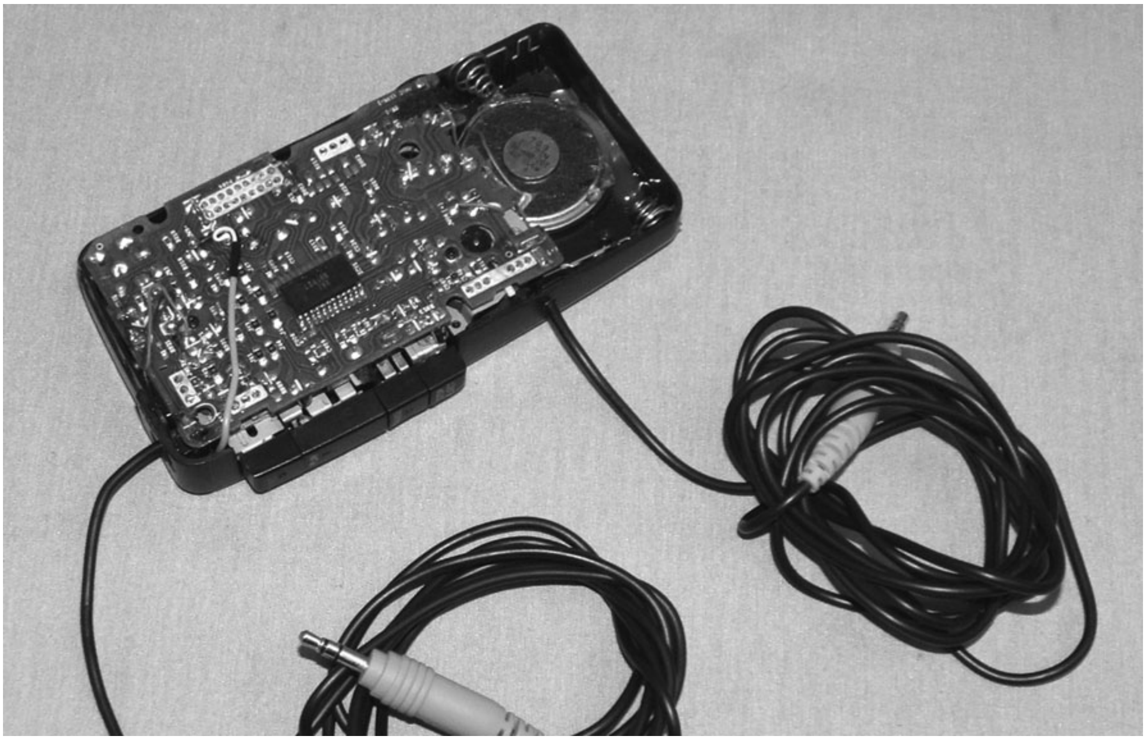


Figure 10-14 *Motor output wires added to the recorder.*

Figure 10-15 *The dream director ready for use.*

yourself up? Headphones are the obvious first choice and ensure that your odd experiments are not forced on anyone else in the room with you, but personally, I can't sleep with headphones stuck to my head, especially when sleeping on my side. If your audio device has a built-in speaker, you could place it directly under your pillow, but most devices do not have that feature, and then you run the risk of pressing buttons while you are asleep. A pillow speaker is very easy to make.

Pillow speakers can be purchased, but that's not my style, so I will show you how to make one that will work perfectly with this and any other audio experiment you might want to try. You can find a small speaker like the one shown in Figure 10-16 in most small radios, audio toys, or even a pair of old headphones. All you need to do is find a small container to fit it into and then solder a headphone jack to the speaker so that it can be plugged into the headphone jack on your audio player. I found a deodorant container that fit the speaker perfectly, so that is what I used.

The completed pillow speaker is shown in Figure 10-17 and also includes a switch that either can be used to shut off the speaker or can be fed back to some other project that requires user feedback. I did not have to make holes in the cabinet for the speaker because it was loud enough just as it was, and it does not take much sound at all at night when your head is right over the speaker. Another simple method for making a pillow speaker is to just take one side of a large headphone system and remove the strap part.

Now you are ready to dive into your favorite movie while you sleep or talk yourself into a lucid dream. The completed rig is shown in Figure 10-18, and it looks more like something from an anarchist's briefcase than something that belongs in a sleep laboratory! Oh well, just don't pack this device in your luggage when going on vacation, and you will be okay. What you plan to record is your business, but here are a few tips to get the best results without waking yourself up.

Figure 10-16 *Making a simple pillow speaker.*

Figure 10-17 *The completed pillow speaker.*

If your audio starts suddenly, it will likely wake you up, especially if your audio file is some fast-paced action scene from your favorite movie. Practically any audio editing software can be used to create a *ramp*, or volume fade-in, over time so that your audio file starts at zero volume and increases gradually in level over a period of a minute or two. Figure 10-19 shows an old version

Figure 10-18 *The completed audio dream director.*

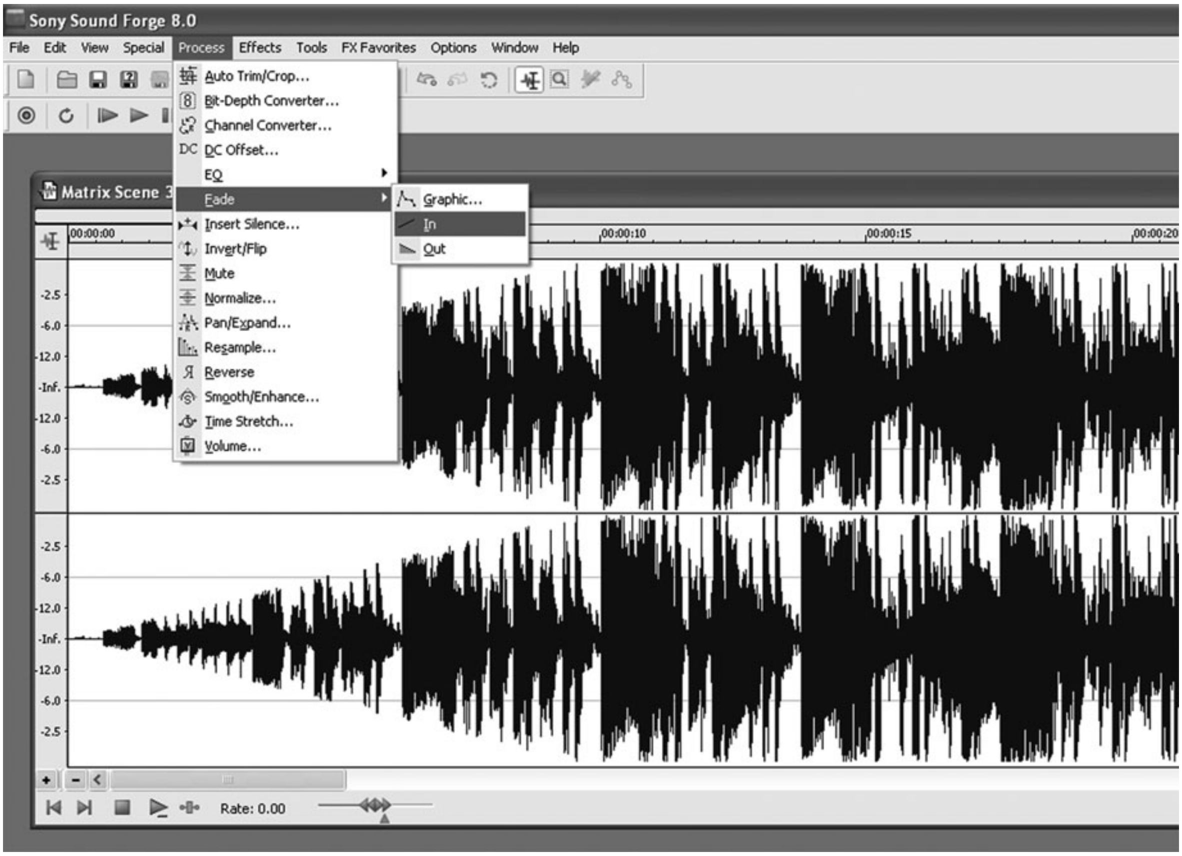


Figure 10-19 *Editing your audio for a gradual volume ramp.*

of Sony Sound Forge applying the "ramp in" filter to a recording of Neo fighting one of the Agents in *The Matrix*. Now the audio file comes in slowly so that it will not wake me from my sleep when the director triggers playback.

I hope that you have fun with this project; I know I did. The audio dream director can be a powerful tool for manipulating your dreamscape or even inducing a lucid dream if used properly. Try your favorite movie scenes or even overdub your own voice telling you to have a lucid dream like some kind of guru. How about sleep learning or self-help content? The possibilities are endless, and as long as you trigger the audio in your last stage of REM sleep and practice proper dream recall, you are guaranteed to see some results with this device. Of course, the dreaming mind is a strange animal, so don't be surprised it that *Matrix* fight scene induces a dream about you watching TV rather than being in the movie!

Project Eleven

Light-Sensing Lucid-Dream Mask

This project will detail the creation of what many consider to be the Holy Grail of lucid-dream hardware. By combining a circuit that monitors eyelid movements with a microcontroller, you end up with a fully self-contained lucid-dream mask or goggles that can detect your REM stage of sleep and send signals to you in the form of flashing lights or trigger any external device such as the audio system shown in the preceding section. By basing the trigger on actual rapid-eye movements, you are guaranteed to get your lucid-dream signals when you need them. These lucid-dream masks, or “dream goggles” as they are sometimes called, have been available for years and were made popular by Stephen LaBerge with his NovaDreamer product.

Our version of the lucid-dream mask will be presented using two different methods of detecting rapid-eye movements directly from the user’s eyelids. The “classic” version will work much the same way that the commercial products do—using an analog amplifier fed by the output of an infrared transmitter and receiver pair to detect small changes in voltage as the dreamer’s eyelids move back and forth. Another method, using a tiny accelerometer IC, actually will detect the eyelid movements directly, which makes the device so much easier to align and get working the first time. I have been experimenting with this alternate dream-mask design for several years now and find it to be more forgiving and accurate than the infrared REM-detection method, even when compared with some commercial units. Of course, you may have reasons to prefer experimenting with the classic system, so both versions will be fully detailed in this project.

In both versions of the dream mask, the output from either the infrared phototransistor or the accelerometer will be fed into a microprocessor for detection so that the signal can be further processed and handled by the user IO code. When 20 or more REM sleep movements have been detected from the sensor, the microprocessor will go into signal mode and begin flashing a visible LED 100 times so that the dreamer can learn to become aware while dreaming and enter a lucid dream. The microprocessor also aids in setup mode, allowing the user to fine-tune the eyelid sensor system by using instant feedback while wearing the mask or goggles. A false-trigger pushbutton is also added so that the unit can be reset if it goes off while he or she is still awake.

This project is presented in two parts because it does require a bit of adjustment and testing to get it set up and working perfectly. The REM-detection mask and related circuit will be built first and tested so that you can decide on which microcontroller to work with or even choose to feed the output directly into a computer for more control and ease of programming. The microcontroller part of this project will be presented next with included source code written in Basic for easy porting to any microcontroller or language. The completed unit will function as well as (or possibly better than) some of the commercially available dream masks, and it will cost you under $20 to build if you already have access to a microcontroller programmer. If you have never worked with microcontrollers such as AVR, PIC, or Basic Stamp, then don't worry; the hobby is inexpensive, easy to learn, and will be explained in more detail in the next part of this project.

Are you ready to become an *oneironaut*, one who travels in dreams? If so, then you will enjoy this project because I have found it to be the most effective way to train yourself to have a lucid dream.

The classic version of the REM-detection unit uses an invisible infrared LED (LED1) to shine light on the user's eyelid so that the reflection can be picked up by a matching phototransistor (PT1) and sent to an amplifier as a changing or modulated voltage. The schematic for this version is shown in Figure 11-1. Invisible light must be used because you don't want to have visible light shone into your eyes all night long. The output from the phototransistor (PT1) is fed into the input of IC1, the ML358 dual op amp. Op amps are great for designing high-gain amplifiers and filters, and because the LM358 contains two op amps, we will be using one as a filter to get rid of noise and the other to greatly amplify the voltage from the phototransistor. The filter is of the low-pass type, which means that a lot of noise induced by ac devices and rf interference will be eliminated so that there will be less noise in the amplified signal. The movement of your eyelids is very slow (1 to 4 Hz), whereas ac hum and other electrical noise will be at a much higher frequency (30 Hz or greater).

The variable resistor (VR1) is used to set the offset of the amplifier so that the signal can be adjusted to compensate for differences in many things, such as ambient light, placement of the

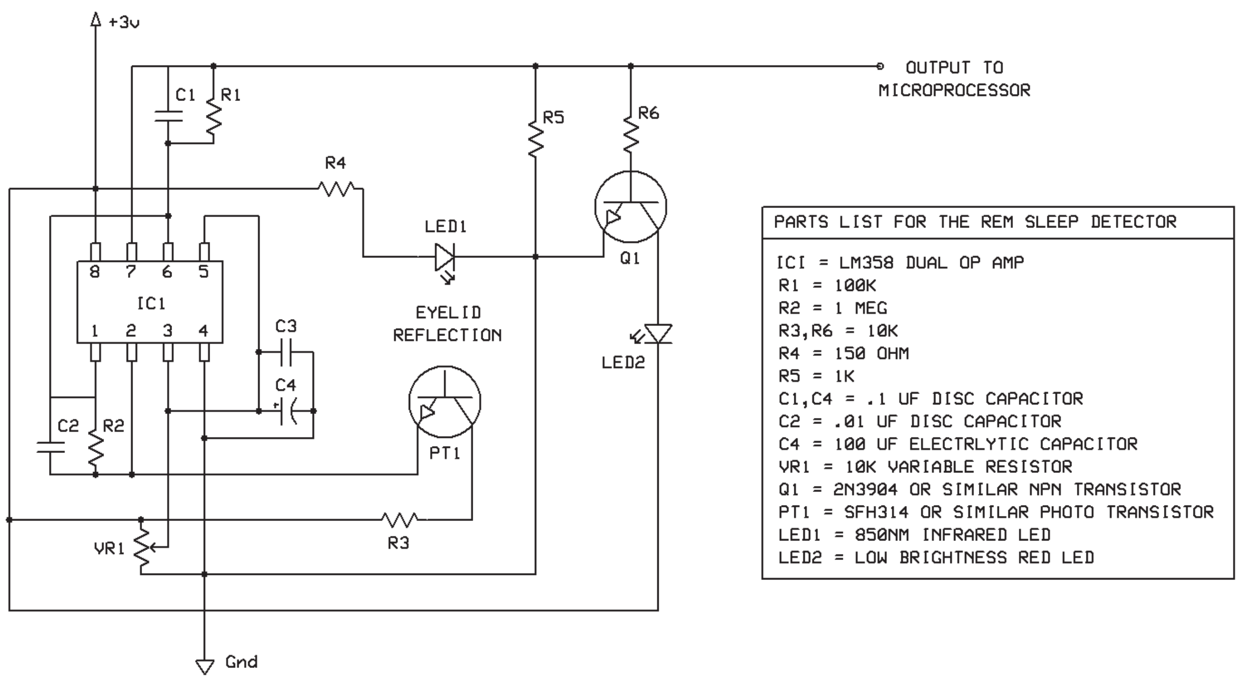


Figure 11-1 *REM-detection device schematic with phototransistor.*

mask on your face, and even the shape of various users' faces. The idea is to set up the system so that slight movements detected on the eyelids return a larger change in voltage so that the microprocessor or computer analog-to-digital converter has an easier time with the raw data. This adjustment of VR1 is critical to operation of the unit and can be quite quirky to get right, so a precision variable resistor is best used because it allows for a lot of fine-tuning.

To aid in the setup of the device, Q1, R6, and LED2 allow a visual indication of the feedback from the amplifier. When you have the detector set up perfectly, the visible LED will flicker slightly as you move your eyelids slightly, an indication of a good change in voltage from the amplifier and filter. Once you are happy with the operation of the unit, Q1, R6, and LED2 can be omitted from the final design because the microprocessor code also will contain a test routine to help you set up the device in real use.

The entire system is designed to run on 3 V dc, so you can power it for several nights on a pair of rechargeable AA or AAA batteries. Also, dc operation keeps unwanted noise from the circuit and isolates your melon from the ac power lines at night. This is good because an electrical storm while you are sleeping could be bad news if your dwelling has a direct strike. I highly recommend that you first build this system on a solderless breadboard because it can be finicky to set up the first time, especially if you decide to use one of the many other op amps available. The LM358 is just a generic op amp that can run from a single voltage supply, but there are many available that would work just fine and possibly give better results. Experimentation is always the goal of this book, so don't be afraid to wander into your own territory and try something different.

The infrared LED and phototransistor pair becomes the heart of this unit, and there are many options available that will work perfectly once you set up the proper distance and reflections spot. The only requirement is that the infrared LED and the phototransistor work at the same wavelength, which will be between 850 and 1050 nm for infrared light. Infrared LEDs from old TV remote controls will work perfectly, and practically any infrared phototransistor will have the correct bandwidth, so it should not be hard to find a pair that work together. RadioShack and many electronics hobby stores will sell them in pairs, and they will look much like the pair shown at the top right of Figure 11-2. The three black-boxed units across the center of the figure are called *position detectors* and contain both the infrared LED and the matching phototransistor in one enclosure. The position detectors work well and are a bit easier to align because they are already set up at the correct angles, but be aware that some of these have a digital output and may not work well in this design. The large unit to the left of the figure has the part number QRB1113 or QRB1114 and was found to work quite well in this design. The unit in the center right of the figure is designed so that an object placed between the two walls will break the beam, and this unit can be hacked apart to reveal the small matched pair, as shown in the bottom of the figure. These beam-breaker units are very common in printers and photocopiers. By using trial and error, you should have no problem selecting the best matched pair from your junk box.

The breadboarded circuit is shown in Figure 11-3, where I use my finger to test the multitude of infrared phototransistor and LED pairs I found in my junk box. The matched pair in this figure was taken from an ear clip heart rate monitor sensor I found on an old exercise bike computer, and it worked very well. Also notice the precision trimpot variable resistor used for VR1. These precision units allow for a very fine control over the entire scale, and they make setup of the amplifier much easier. Another thing to note is that ambient light will have a huge effect on the workings of this circuit, so it is best to set it up in a dark room. Sunlight and many room lighting systems also will contain infrared light, so the phototransistor could act much differently in a

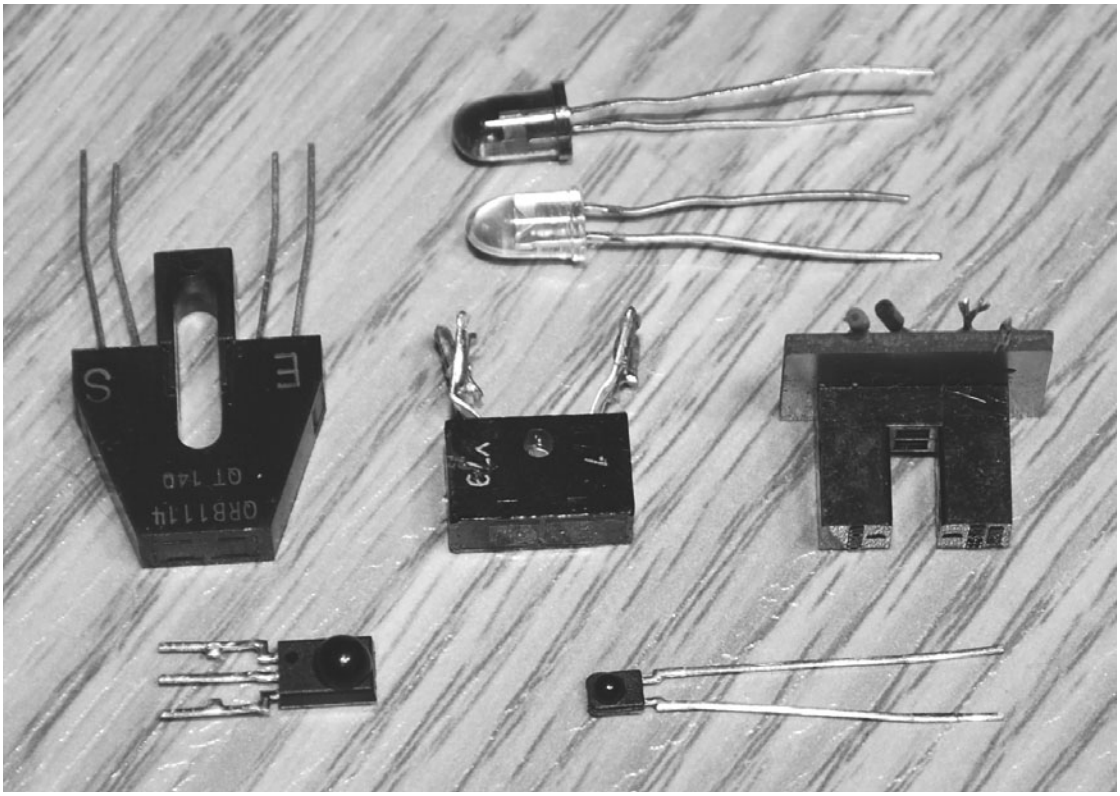


Figure 11-2 *Trying various infrared detectors.*

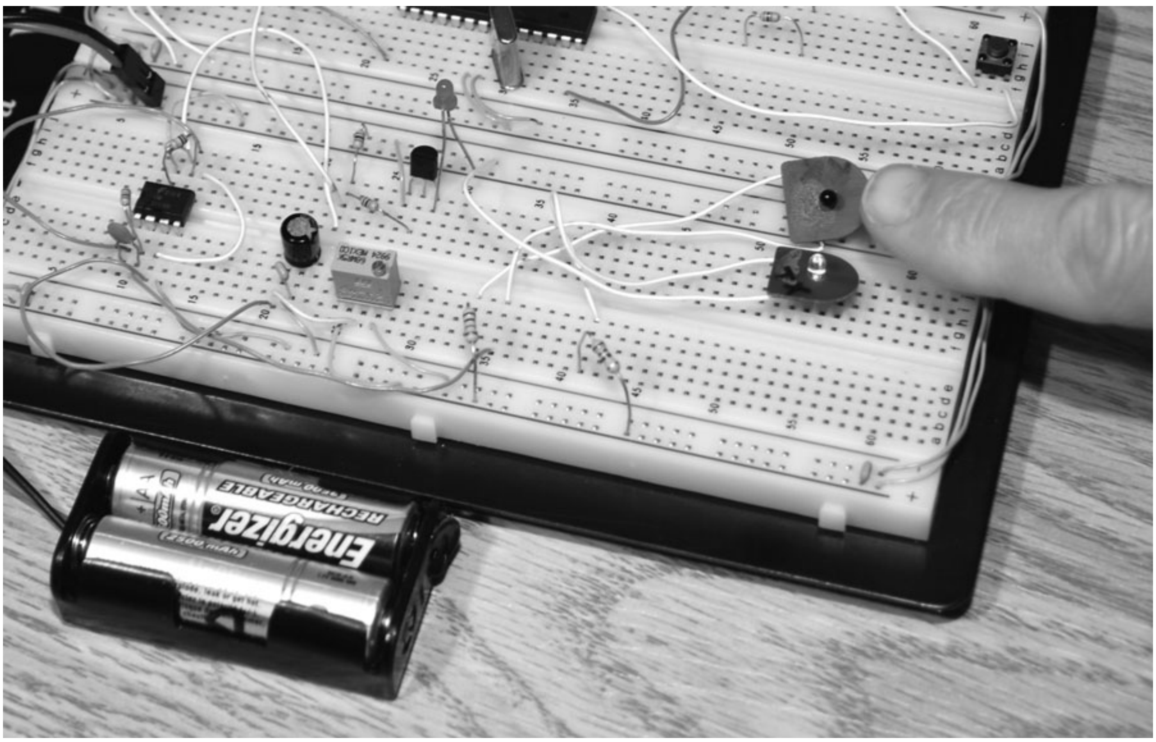


Figure 11-3 *Testing the infrared-detector response.*

lighted room, requiring more setup. Try to point the phototransistor and LED at an angle so that they meet at a point that is about 1 in away. This will be approximately the distance from your eyelid to the matched pair once the unit is built into some type of goggle or mask housing.

Most of these do-it-yourself (DIY) dream masks are built into small swim goggles like those shown in the lower part of Figure 11-4, but I found a ski mask to be so much more comfortable when I was wearing it for extended periods. You want the mounting system to be as loose as possible on your face yet secure enough that it does not shift as you move around in bed. Swim goggles are certainly secure, but the strap seems to be too tight, and the rubber eyepieces are not really designed for comfort as much as they are for creating a watertight seal. If you replace the rubber head strap with some lighter elastic and drill holes in the goggle casings, they are not too bad, but the ski mask is a much better solution and will feel a lot more comfortable and secure. Using a ski mask also gives you a bit of room to mount the battery pack and electronics so that you can make a self-contained unit rather than having wires in your bed.

If you are a junk collector like I am, then you probably have a bunch of infrared beam-breaker-type sensors in your junk pile like the one shown in Figure 11-5. These small units can be broken apart to release the phototransistor and matching LED by carefully prying open the case using a small knife. Before you start hacking, look at the case so that you can identify each component, which will have either an arrow pointing from the LED to the phototransistor or markings that read “AK” (anode, cathode) and “CE” (collector, emitter). If there are no markings at all on the shell, you can just guess by trying each component one way or the other until you get some response. This will not fry the parts because the voltage and current are very low in the circuit.

Figure 11-4 *Selecting a comfortable mounting system.*

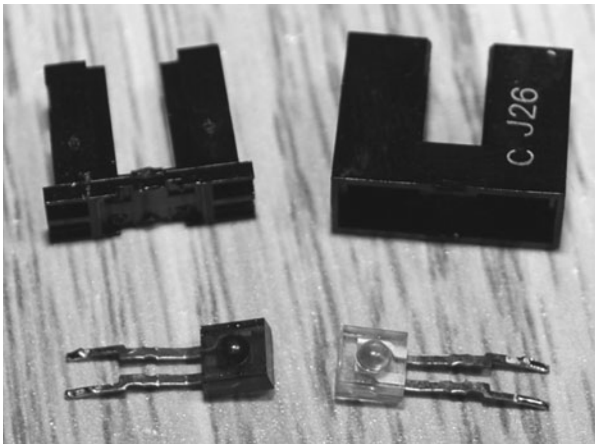


Figure 11-5 *Hacking a photo sensor into its matched pair.*

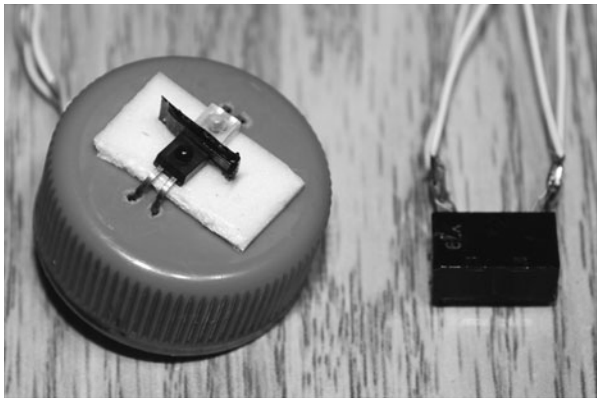


Figure 11-6 *Positioning the photo-sensor components.*

I decided to use the tiny phototransistor and LED that I hacked from the sensor shown in Figure 11-5 because they worked very well. There were no markings on the case, so it took a while to figure out by trial and error that the darker unit was the receiver and the lighter unit was the transmitter. Figure 11-6 shows the matched pair mounted to a bottle cap using some double-sided tape to hold them in place. I also had to cut a bit of the black plastic case out and place it between the two components to keep the beam from reflecting directly from one to the other rather than bouncing off my eyelid first. The small sensor shown on the right in the figure is one of the all-in-one position sensors and is basically the same thing as my bottle-cap hack but ready to be used. You can find these all-in-one units at any electronics supplier by searching for optical position sensors or distance sensors.

After testing various optical-sensor configurations and deciding that the bottle-cap hack was the best one, I placed the components on a small perf board so that the system could be connected to my ski mask for further testing. Figure 11-7 shows the small perf board with the programmed microcontroller (discussed in the last section of this chapter) ready for operation. It is a good idea to get the analog part of this project working before you move on to the microcontroller or computer interfacing because this is the area that will take the most effort to get working properly. I tested my ski mask with the optical sensor while the circuit was still on the breadboard before getting to this stage in the design. If you have an oscilloscope handy, feed the infrared LED with a small 1.5-V watch battery, and then plug the output from the phototransistor into your oscilloscope so that you can see the analog results directly while you wear your face mask and move your eyes back and forth (one closed eye). This will give you a direct reading of the responsiveness of your optical sensor's output. Some careful alignment might be necessary depending on the beam width and field of view of each of the components.

The proximity and angle of the phototransistor and infrared LED are very important for optimal response and may take quite a bit of fine-tuning to get working properly. If you are using a self-contained photo sensor, then only the distance from the edge to your eyelid will be important, but if you are using a separate phototransistor and LED, then both the distance and angle will matter. Figure 11-8 shows my hacked bottle-cap sensor placed inside the ski mask lens at the approximate position where my eye will be. I put on the mask and drew a mark on the lens while looking in the mirror so that I could find this position. A line was drawn around the cap so that the lens could be cut out using a routing bit on a Dremel tool. I could have just glued the cap