

## BUILD CUSTOM LEGO-BASED ROBOTS

---

Once just a playtime toy for youngsters, LEGO parts are now used in schools to teach construction and mechanics, even robotics. In fact, LEGO ‘bots are becoming so common that many schools and libraries around the world host LEGO robot-building races and competitions. The main benefit of a LEGO-based robot: The snap-together design of LEGO pieces allows you to build a robot in just an hour or two, with no cutting or drilling necessary. Changes and improvements are easy to make too, and you can do them “on the fly” as you experiment with new ideas.

There are plenty of books, magazine articles, and Internet sites that discuss the design and construction of all-LEGO robots. So, in this chapter we’ll review another approach: using various LEGO parts as a *basis* for a robot and then augmenting those parts with other construction materials. For example, the robot described in this chapter uses modified hobby servo motors and not the more costly LEGO gear motors. Some gluing and other so-called hard construction is also called for.

### Working with LEGO Parts

---

There are several dozen varieties of LEGO parts, which come in various sizes (not to mention colors). In your robotics work you’ll likely encounter the primary construction pieces detailed in the following subsections.

## BUILDING BLOCKS

The three primary building blocks are shown in Figure 12.1.

*Bricks.* The basic LEGO building block is the brick. Bricks have one or more raised “nubs” (or “bumps”) on the top and corresponding sockets on the bottom. The nubs and sockets mate, allowing you to stack bricks one on top of another.

*Plates.* Plates are like bricks but are half their height. Like bricks, plates have one or more nubs on the top and corresponding sockets on the bottom, and they are made to be stacked together.

*Beams.* Beams are specific to the Technic brand of LEGO parts, such as the LEGO Mindstorms Robotics Invention System, which is detailed in the next two chapters. Beams are bricks that have one row of nubs and then holes down their sides. Beam variations include the L and hooked shape, which can be used for such things as building robotic grippers.

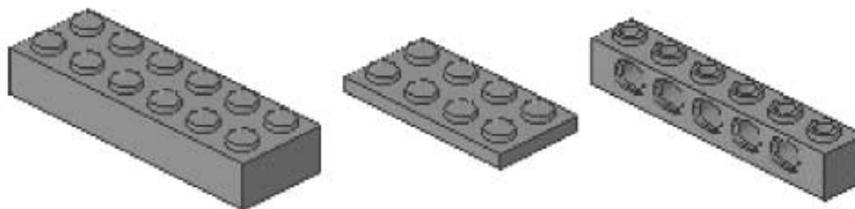
The arrangement of the nubs on bricks and plates define their size. For example, a “2 by 8” brick or plate has two rows and eight columns of nubs. In some LEGO instruction books you’ll see the nubs referred to as “units,” such as “2u-by-8u” (the *u* stands for units). I’ll stick to this style as a quasi-standard throughout the rest of the chapter.

*Building plates* are very large versions of plates, which are available in the 8u-by-16u size. Thin baseplates are also available; these lack sockets underneath.

## CONNECTION PIECES

Figure 12.2 shows several common LEGO parts that are used for connecting things, such as one beam to another or a wheel to a beam. The four most common are as follows:

*Connector peg.* Connector pegs are typically used with the LEGO Technic line to attach beams together. Connector pegs are labeled either 1/2, 3/4, or full—the dif-



**FIGURE 12.1** Bricks, plates, and beams are the primary LEGO pieces used for construction. They are available in various sizes.



**FIGURE 12.2** Connection pieces allow you to snap various LEGO parts together.

ference lies in the length of one side of the connector's middle collar. Some connector pegs are half connector and half axle (we'll discuss axles at the end of this list).

*Friction connector.* Available in a wide variety of sizes and shapes, friction connectors are used with both LEGO Technic and standard parts to connect such things as beams and wheels.

*Bushing.* Bushings hold things, such as axles. So-called half bushings are about half the width of a standard bushing. Some can also be used as miniature pulleys.

## PULLEYS, TIRE HUBS, AND GEARS

LEGO parts are replete with a variety of wheels, pulleys, and gears. Several common varieties are shown in Fig. 12.3. In almost all cases, you attach pulleys, tire hubs, and gears using stud axles and/or friction connectors.

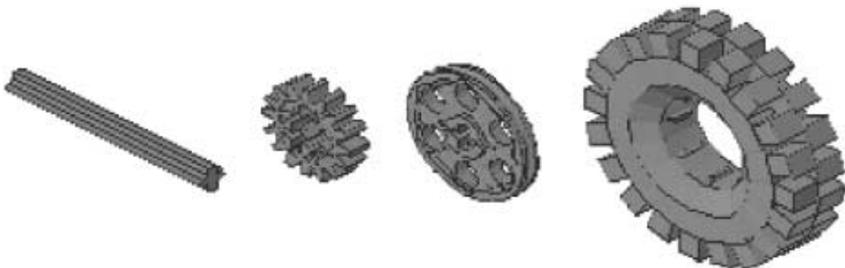
*Wheel hubs.* Wheel hubs are used with any of a number of different-sized rubber or plastic wheels. There are several sizes of wheel hubs.

*Pulleys.* Pulleys are used with string, rubber bands, or O-rings to create a kind of belt. Several sizes of pulleys are available.

*Gears.* There are more than a dozen kinds of LEGO gears. All are designated by the number of teeth they have on their outside circumference. For example, the 8-tooth spur gear has eight teeth; the 14-tooth bevel (slanted) gear has 14 teeth. All gears of the same type (e.g., spur or bevel or crown) are made to mesh with another, and you can often mix one type with another. For example, a spur gear can be effectively used with a crown gear.

*Stud axle.* Stud (or "cross") axles come in various lengths, which are designated by number (e.g., 4, 6, 8, etc.). Axles connect to gears, wheels, and other parts. They have a spine of four studs along their length.

There are also toggle joints, sloped beams, catches, cross blocks, axle connectors, cams, and a raft of others. Check out any good book on LEGO, or see the LEGO Web site for more information on LEGO parts.



**FIGURE 12.3** Use LEGO wheels, pulleys, and gears to enable your robot creations to scoot around the floor.

## Securing Parts

---

LEGO parts are made to snap together. When properly constructed, the snap-together system provides for relatively strong joints. However, the active nature of most robots—machines that are often in motion and perhaps bumping into things—makes snap-on joints less than ideal. If you're not just experimenting with a new, perhaps temporary, design, you may want to think about using alternative means to secure LEGO parts together. You'll also need a different method of securing if you're combining LEGO and non-LEGO parts.

In the next five subsections we discuss methods for securing LEGO parts together. Take note: except for cementing, gluing, and hardware (nuts, bolts, and screws) the remaining methods discussed here—taping, tie wraps, and Velcro—are best used for nonpermanent constructions. These materials do not provide a long-lasting bond, especially for a robot that's always on the go.

### **CEMENTING AND GLUING**

You can cement together nonmovable pieces using ordinary household cement (such as Duco), epoxy, contact, or CA (cyanoacrylate) glues. If you want a very permanent bond, use clear ABS plastic solvent (LEGO parts are made of ABS plastic). ABS solvent cement is available at most home improvement stores, as well as many hobby stores. Another approach is to use hot-melt glue. You can use either the standard or low-temperature variety. (See Chapter 8 for more information on bonding plastic.)

When using glue or epoxy, be sure to rough up the surfaces to be joined. This is especially important when using hot-melt glues. There is less need to rough up the edges when you are using solvent-type cements since the adhesive actually melts the plastic to form a strong joint.

### **TAPING**

Temporary constructions can be taped together using black electrical tape or even strapping tape normally used for box packaging. Do note that most tapes leave a sticky residue. Use acetone to remove the residue.

### **TIE WRAPS**

Tie wraps are used to bundle wires and other loose items. They consist of a single piece of plastic with a catch on one end. You thread the loose end of the tie through the catch. A ratchet in the catch keeps the tie in place. Most ties have a one-way ratchet that cannot be undone; others have a catch “release” so you can reuse the tie. Tie wraps are available in a variety of lengths, thicknesses, and colors. Get an assortment, and use the size that best suits the job.

### **NUTS, SCREWS, AND BOLTS**

You can use small hardware to attach LEGO pieces together. In most cases, you'll need to drill through the plastic to secure the screw or bolt. You can use self-tapping metal screws

if the plastic material is thick. When you use self-tapping screws, be sure to use a drill size slightly smaller than the screw.

### VELCRO (HOOK-AND-LOOP)

You can attach non-LEGO things to LEGO pieces with Velcro (otherwise known by the generic term “hook-and-loop”). Get the kind with self-sticking adhesive backing. Put the hook material on one piece and the loop on the other. Press them together for a very strong—but not permanent—bond.

## Build the LEGO Pepbot

The Pepbot is a small, two-wheeled robot constructed from an assortment of LEGO parts. Motive power for the Pepbot comes from two radio-controlled (R/C) servos and two foam tires, both of which are available at most any hobby store that sells radio-controlled parts and accessories. The “brain” of the Pepbot is a microcontroller—just about any microcontroller will do. In this section, you’ll learn how to construct the Pepbot body using common LEGO parts and how to attach the servo motors and tires. Apart from some special modifications you need to make to the servos, constructing the Pepbot consists of snapping together LEGO parts and cementing things in place with epoxy or hot glue.

We’ll also briefly introduce the idea of using the OOPic object-oriented microcontroller to control the robot. This microcontroller, and others, are covered in more detail in Chapters 31 through 33.

### CONSTRUCTING THE BODY

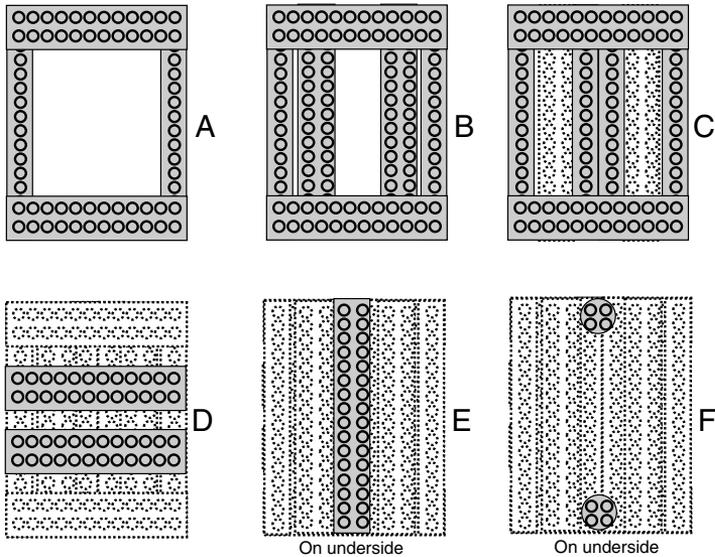
Begin by building the basic frame for the Pepbot. Use the following steps.

#### Frame

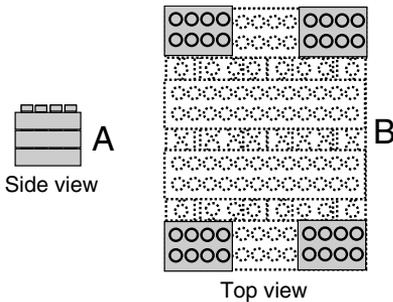
1. Begin by assembling the frame as shown in Fig. 12.4a. Attach two 16u beams to two 2u-by-12u plates, forming a rectangle.
2. On the underside of the frame attach the two 2u-by-16u plates lengthwise (see Fig. 12.4b).
3. Connect two additional 16u beams down the centerline of the bottom of the frame (Fig. 12.4c).
4. Attach two 2u-by-12u plates to the top of the frame, as shown in Fig. 12.4d.
5. Attach a 2u-by-6u plate down the centerline to the top of the chassis, as shown in Fig. 12.4e.
6. Attach a round pad with a 2u-by-2u round brick, then press this assembly into the ends of the centerline beams, on the bottom of the frame (Fig. 12.4f).

#### Side blocks

1. Create four sets of three 2u-by-4u bricks, stacked on top of one another (Figure 12.5a)
2. Attach the assembled side blocks to the frame, as shown in Figure 12.5b.



**FIGURE 12.4** Pepbot main construction. See text for the construction sequence.

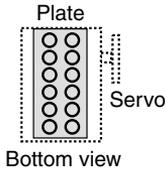


**FIGURE 12.5** Construction of Pepbot side blocks.

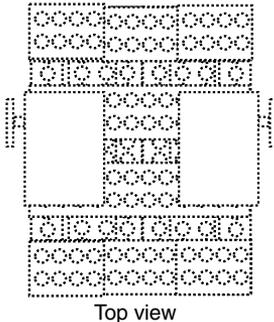
## ATTACHING THE SERVO MOTORS

The Pepbot uses two modified R/C servos for drive motors. In an ordinary servo, rotation is limited by an internal stop inside the motor. By modifying the servo, you can make the motor turn continuously. Modifying servos is not a particularly hard task, but the exact steps will vary depending on the model of servo you are using. See Chapter 20 for details on how to modify several popular R/C servos. For the rest of this section, we'll assume you've already modified two servos and are ready to use them on the Pepbot.

Use epoxy or hot-melt glue to affix two 2u-by-6u plates to the side of each servo casing, as shown in Fig. 12.6. Use only a moderate amount of epoxy or hot-melt glue, as you may need to remove the plate from the servo casing. For one servo, glue the plate to the right side (looking at the servo top down, with the output shaft on the top). For the other servo, glue the plate to the left side. Be careful to align the plates on the servos so they are



**FIGURE 12.6** Attaching servo motors to LEGO plates.



**FIGURE 12.7** Securing the mounted motors to the rest of the robot.

straight, and make sure the plate doesn't block the seam in the case of the servo (this will make it easier to disassemble the servo, if you need to for whatever reason).

Hot-melt glue is the preferred construction technique for this task, as the glue sets quickly—usually in under a minute in normal room temperature. Hold the servo in place until the glue has set. Again, be absolutely sure that the plate is squarely affixed to the servo. After the glue has set and dried, you may attach the servos (now on their motor mounts) to the frame of the robot, as shown in Fig. 12.7, using two 2u-by-12u plates.

## ATTACHING THE WHEELS

The wheels of the Pepbot are lightweight foam tires, which are used in model R/C airplanes. I selected wheels with a 3-inch diameter, which makes the Pepbot travel fairly fast across the floor (hence the name Pepbot). You can use smaller wheels if you wish, but consider the following:

The smaller the wheel, the slower the robot. R/C servo motors turn at about 1–2 revolutions per second, depending on the model (some are slower; some are faster). With 3-inch wheels and a 1.5 rps servo, the Pepbot will travel about 14 inches per second. This speed is calculated by multiplying the diameter of the wheel (3) by pi (3.14), then multiplying that number by the speed of the servo (1.5 rps). That is,  $3 * 3.14 * 1.5$ , or approximately 14. Just as smaller wheels will make Pepbot a little less peppy, larger wheels will make Pepbot travel faster.

The smaller the wheel, the less clearance there is between the bottom of the robot and the floor. Conversely, the larger the wheel, the more clearance there will be. This can be helpful if you run Pepbot over thick carpet or want it to travel over small bumps, like the threshold between a carpeted and a tile room.

The design of the Pepbot will not allow tires smaller than about  $2 \frac{3}{4}$  inches because of the position of the servo motor output shafts. You'll have to redesign Pepbot if you want to use smaller wheels.

To attach the wheels, first insert the standard 1-inch servo plate over the output shaft of the servo. This plate connects firmly to the output shaft using a screw, which is provided with the servo. Then, use epoxy or hot-melt glue (hot-melt glue is preferred) to secure the hub of the wheel to the plate that is connected to the servo. *Be absolutely sure* that the wheel hub is exactly centered over the servo plate; if it is not, the robot will not travel in a straight line.

## ATTACHING THE DECK

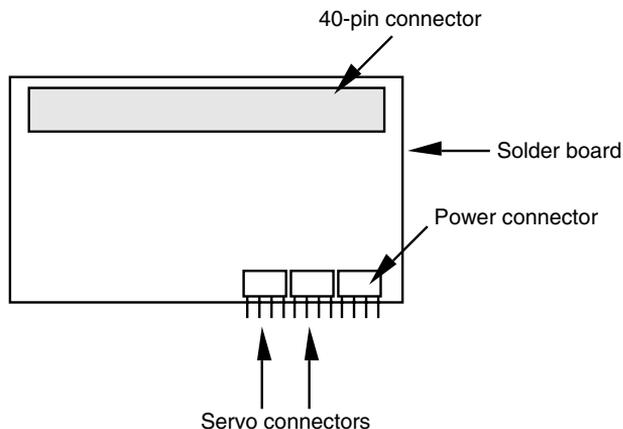
The “deck” is where you place the Pepbot’s batteries and control circuitry. The deck is simply an 8u-by-16u LEGO building plate. A four-AA battery pack is affixed to the deck using double-sided foam tape. The microcontroller board (in this case an OOPic) is attached using small wire tires.

## CONSTRUCTING THE INTERFACE BOARD

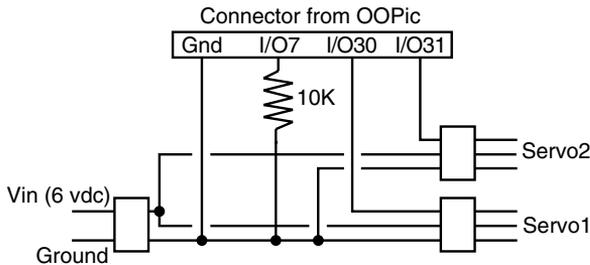
An interface board allows you to easily connect the microcontroller, batteries, and the two servos. It also provides adequate room if you want to add to the Pepbot, such as light sensors to detect light and dark, bumper switches to determine if the robot has hit an obstacle, or a small audio amplifier and speaker to allow the Pepbot to sound off when it wants to.

Construct the interface board by using a small 2-by-3 inch project board, available at Radio Shack. You can use most any construction technique that you like. I used wire-wrapping for the Pepbot prototype. See Fig. 12.8 for a construction diagram; Fig. 12.9 shows the schematic you should follow.

Note the 40-pin cable and connector. This cable attaches to the I/O (input/output) port of the OOPic microcontroller. While the OOPic provides for 31 inputs and outputs, the Pepbot uses only three of them. As you can see, there is plenty of room to expand the Pepbot with sensors, should you wish to do so. See Chapter 29, “Interfacing with



**FIGURE 12.8** Construction plans for the Pepbot interface board. The board leaves plenty of room for expansion.



**FIGURE 12.9** Schematic for the Pepbot interface board.

Computers and Microcontrollers,” for some ideas on creating and interfacing sensors to robots.

## BATTERY POWER

The Pepbot is powered by two separate battery sources, as follows:

A pack of four AA cells provides power to the two R/C servo motors. You may use 1.5-volt alkaline cells or 1.2-volt nickel-cadmium or nickel-metal hydride batteries. However, if you use 1.2-volt cells the servos will run a little slower than under 1.5-volt cells (4.8 volts for the four batteries, as opposed to 6 volts). Under normal use, this battery pack will last for about 30–60 minutes of “play” time, so you may wish to use rechargeable alkalines to save money.

A 9-volt “transistor” battery provides power to the microcontroller. The microcontroller consumes little current, so this battery should last for a long time under normal use.

The separate battery supplies serve an important purpose: R/C servo motors consume a lot of current when they are first turned on, so much, in fact, that the power for the four-battery pack can dip to under 4.5 volts. The OOPic, like most microcontrollers, employs a “brownout” circuit that resets the controller when the voltage falls below a certain level. This prevents the controller from operating in an unstable state. Using the same power pack for both servos and microcontroller can cause erratic behavior in your programs.

For ease of connection, attach a female connector to the end of the battery leads for both the four-AA pack and the 9-volt battery. Both connectors should be the common 0.100-inch pin type. Note that the OOPic microcontroller board already contains a 5 vdc (volts dc) regulator, so no outboard voltage regulator is required. The R/C servos do not require regulation. The ground connections of both battery packs should be connected together.

## CONNECTING THE PIECES

Thanks to the interface board, connecting the servos, batteries, and microcontroller is a snap. The steps are as follows:

1. Connect the four-AA battery pack (observe proper polarity!) to the interface board.

2. Connect both the right and left servos to their respective connections on the interface board. Again, observe correct polarity. One or both servos may “jump” slightly when plugged in, but they should not run.
3. Connect the I/O cable between the OOPic and the interface board.
4. Connect the 9-volt battery to the OOPic microcontroller board.

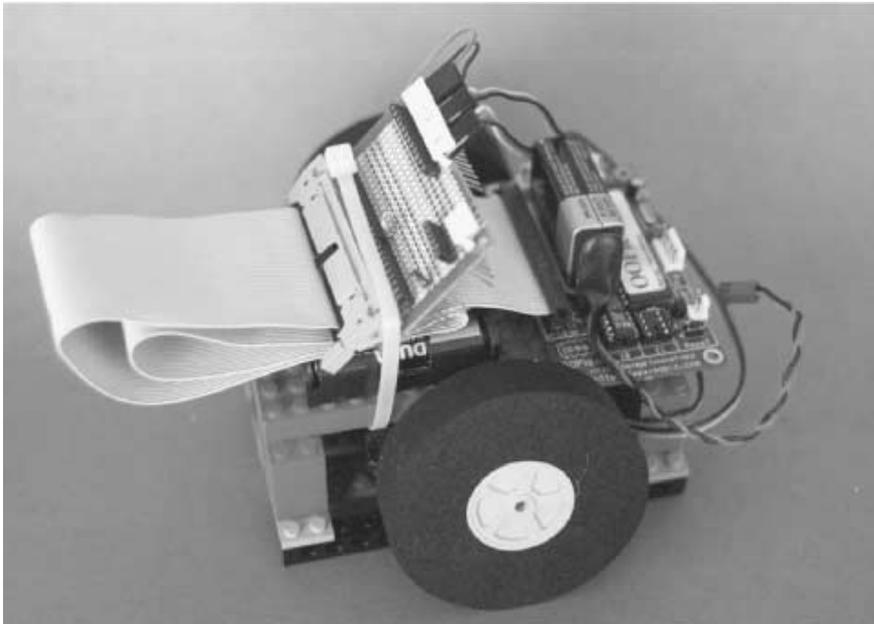
Figure 12.10 shows the completed Pepbot.

## PROGRAMMING THE MICROCONTROLLER

As mentioned earlier, the Pepbot uses an OOPic object-oriented microcontroller—a kind of computer on a single chip. The OOPic is just one of many microcontrollers you can use. Chapter 33 describes the OOPic in more detail, so in this section we’ll just briefly outline the steps for connecting it to the Pepbot. By the way, feel free to use another microcontroller, if that’s your wish. The popular Parallax Basic Stamp can also be used with the Pepbot. Chapter 31 is devoted entirely to using and programming the Basic Stamp.

Listing 12.1 shows the testing program for calibrating the modified servos. Calibration is required to find the “center position” of the servos. In an unmodified servo, the OOPic controls the position of the servo output shaft by using a number from 0 to 63. Zero turns the output shaft of the servo completely one direction, and 63 turns it completely the other direction.

Since the servos in the Pepbot have been modified for continuous rotation (see Chapter 20 for more detail on this technique), the values of 0 and 63 are used to control the direction of the motor. It stands to reason, then, that a value of 31 (midway between 0 and 63)



**FIGURE 12.10** The Pepbot, ready for programming.

should stop the motor. This stop point is calibrated to the value of 31. As detailed in Chapter 20, the modified servo is equipped with an external 5K trimmer potentiometer.

To run the program you must first download it to the OOPic microcontroller. This process is outlined in Chapter 33, and so won't be repeated here. Once downloaded, the program will automatically run, so be sure you have the Pepbot's wheels raised off the floor or table for testing. (*Important! If the OOPic doesn't automatically run the program when downloaded, unplug the programming cable from it. The program should now immediately run.*)

With the servos calibrated, load the program as shown as Listing 1. This is a demonstration program than runs the Pepbot “through its paces,” making it go forward and backward and in turns. Notice that subroutines such as *GoForward* and *HardLeft* are used for motion control. The Pepbot is able to move in all directions by controlling the right and left motors independently—either by turning the motors off or by turning the motors forward or backward. For example, to go forward both motors are told to turn on in the forward direction. To make a “hard” left (spin in place toward the left), the left motor goes in reverse, and the right motor goes forward.

### Listing 12.1

```

Dim S1 As New oServo
Dim S2 As New oServo
Dim CenterPos as New oByte
Dim Button As New oDio1
Dim x as New oByte
Dim y as New oWord

'-----
Sub Main()
CenterPos = 31           ' Set centering of servos
Call Setup
Do
    If Button = cvPressed Then
        ' Special program to calibrate servos
        S1 = CenterPos
        S2 = CenterPos
    Else
        ' Main program (IO line is held low)
        Call GoForward
        Call LongDelay

        Call HardRight
        Call LongDelay

        Call HardLeft
        Call LongDelay

        Call GoReverse
        Call LongDelay
    End If
Loop
End Sub

'-----
Sub Setup()
Button.Ioline = 7           ' Set IO Line 7 for function input
Button.Direction = cvInput ' Make IO Line 7 input
S1.Ioline = 30             ' Servo 1 on IO line 30

```

```
S1.Center = CenterPos           ' Set center of Servo 1
S1.Operate = cvTrue            ' Turn on Servo 1
S2.Toline = 31                 ' Servo 2 on IO line 31
S2.Center = CenterPos         ' Set center of Servo 2
S2.Operate = cvTrue           ' Turn on Servo 2
S2.InvertOut = cvTrue         ' Reverse direction of Servo 2
End Sub

'-----
Sub LongDelay()
    For x = 1 To 200:Next x
End Sub

Sub GoForward()
    S1 = 0
    S2 = 0
End Sub

Sub GoReverse()
    S1 = 63
    S2 = 63
End Sub

Sub HardRight()
    S1 = 0
    S2 = 63
End Sub

Sub HardLeft()
    S1 = 63
    S2 = 0
End Sub
```

Save, compile, and download the program in Listing 12.1. You may need to depress the reset button on the OOPic microcontroller board to prevent the Pepbot from activating prematurely. Place the robot on the floor, then release the reset button. The Pepbot robot should come to life, first going forward, then spinning both to the left and right. It should then turn right, then left, and then finally back up. The program will repeat itself until you disconnect the power.

If you have constructed your Pepbot properly, the robot should trace more or less the same area of the ground for each iteration through the *Do* loop of the program in Listing 12.1. If the Pepbot veers in one direction when it's not supposed to, it could indicate that either one or both wheels were not properly attached to the servo plates. Or it could indicate that the servos are not properly aligned on their mounting plate. Assuming you didn't apply too much epoxy or hot-melt glue, you should be able to use moderate force to remove the wheel and/or mounting plate. Clean off the old epoxy or hot-melt glue, and try again.

Once the Pepbot test program is a success, you can play around with coding your own actions for the robot. With sensors that you can add to the interface board, you can have the Pepbot respond to external stimulus, such as light or touch.

---

## From Here

### *To learn more about...*

Building and programming robots with the Mindstorms Robotics System

### *Read*

Chapter 13, "Creating Functionoids with LEGO Mindstorms Robotics Invention System"

Using and modifying R/C servos

Chapter 20, “Working with Servo Motors”

Connecting robot motors to a computer, microcontroller, or other circuitry

Chapter 29, “Interfacing with Computers and Microcontrollers”

Controlling a LEGO robot using the Basic Stamp

Chapter 31, “Using the Basic Stamp”

Controlling a LEGO robot using the OOPic microcontroller

Chapter 33, “Using the OOPic Microcontroller”