by Jeff Bingham & Lee Magnusson

# Inertial Rolling Robot

*Jeff and Lee's H8/3664-based rolling robot is capable of inertial movement. A DC electric motor is attached to a pendulum and suspended inside an inflated ball, which provides the driving force.*

We live in miraculous times. In today's world, it is now possible for you to design and build devices that even 20 years ago would have bordered on science fiction. Wheeled robots are now relatively common, and for the individual looking for novel projects, new types of motion are often sought.

In this article, we'll outline a mobile robot design that adds a twist to simple rolling (see Photo 1). We'll also discuss the electronics, software, and mechanical issues that we encountered. We hope that you enjoy this robot as much as we have!

## ELECTRONICS

The robot employs a single driven pendulum inside of a sphere. With 2 degrees of freedom, it drives and tilts the ball (see Figure 1). With this geometry, the robot can reach high speeds, steer, and jump. In addition, it doesn't get stuck when it flips over. As a toy, it exhibits enough entertaining motion to be quite endearing and very robust. As a project, it offers an exciting venture into mechanical and electronic design.

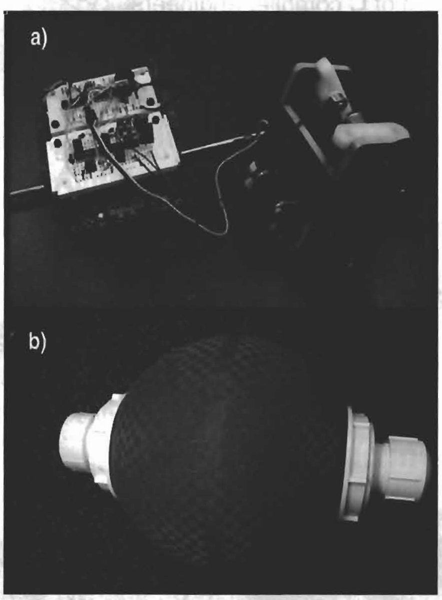


**Photo 1a**—*The robot's inner core consists of an electronics assembly and a mechanical drive assembly.* **b**—*The robot's spherical shell is an inflatable playground ball. The inner core is suspended via a drive axle from the two plastic caps.*

The robot's electronics perform two main functions, actuation for motors and sensors for feedback. There is actuation by means of both a bidirectional brushed DC motor and a digital proportional servo motor (see Figure 2). The two motors provide motive force and turning ability, respectively.

The robot has a pressure sensor and a motor-current sensor, although the latter was not enabled in the current software design. The main processor is a Renesas Technology H8/3664 microcontroller (see Figure 3). We used a protoboard for ease of construction. The DC drive motor is powered by a 12-V battery source consisting of 10 NiMH batteries. The servomotor and the remainder of the electronics are powered by a regulated 5-V source.

## DRIVE MOTOR

We used a brushed DC drive motor that has a+ and a– inputs. Forward drive is produced by connecting the positive input to the motor-voltage supply and the negative input to ground. Reverse drive is produced by switching the two inputs. Power amplification and bidirectional rotation is accomplished by means of a MOSFET H-bridge. International Rectifier IRL3103 MOSFETs with 64 A of continuous drain current were selected because of their low on-state resistance and their ability to handle extreme currents without a heatsink.

The motor has a stall torque of only 2 A, but you can upgrade to a larger one if you want. The average voltage to the motor is varied by switching between high and low states of the H-bridge at a high frequency (10 kHz). This is accomplished by pulse-width modulating the voltage signals to the motor. The MOSFET H-bridge is driven by the two International Rectifier IR2184 Half-bridge MOSFET drivers. The IR2184 is designed for automotive use and will operate with 10 to 25 V. Thus, the same 12-V source from the batteries is used for both the gate-drive supply to this driver as well as the voltage supply for the MOSFET H-bridge.

The MOSFET driver has only a single IN command, which makes for convenient interfacing. The driver will turn on the high-side MOSFET for a logic level of 1 and the low side for a logic level of 0. This behavior is optimal in an H-bridge circuit with an inductive load, because the current

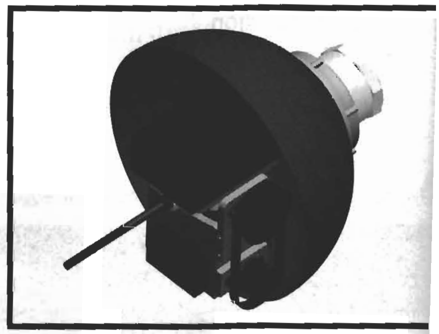

**Figure 1**—*The robot was designed with CAD software to ensure the precise fit of components. In this view, the spherical shell is cut away to show the robot core.*

will automatically continue through the low side of the bridge (see Figure 4). However, by turning on the low-side MOSFET, current can be carried more efficiently through the MOSFET, rather than through the parallel diode. Even though the IR2184 can be connected to the microcontroller directly, a buffer circuit is used to prevent damage to the microcontroller in case of misconnection. The buffer circuit also performs convenient logic functions so that only one PWM channel and one I/O port are necessary on the microcontroller.

Table 1 is a truth table for this logic circuit, which uses a 7404 inverter and two 7402 NOR gates. Thus, the logic circuit sends the PWM signal from one driver to the other, depending on the value at the I/O port. The opposite driver is sent a logic level of 0.

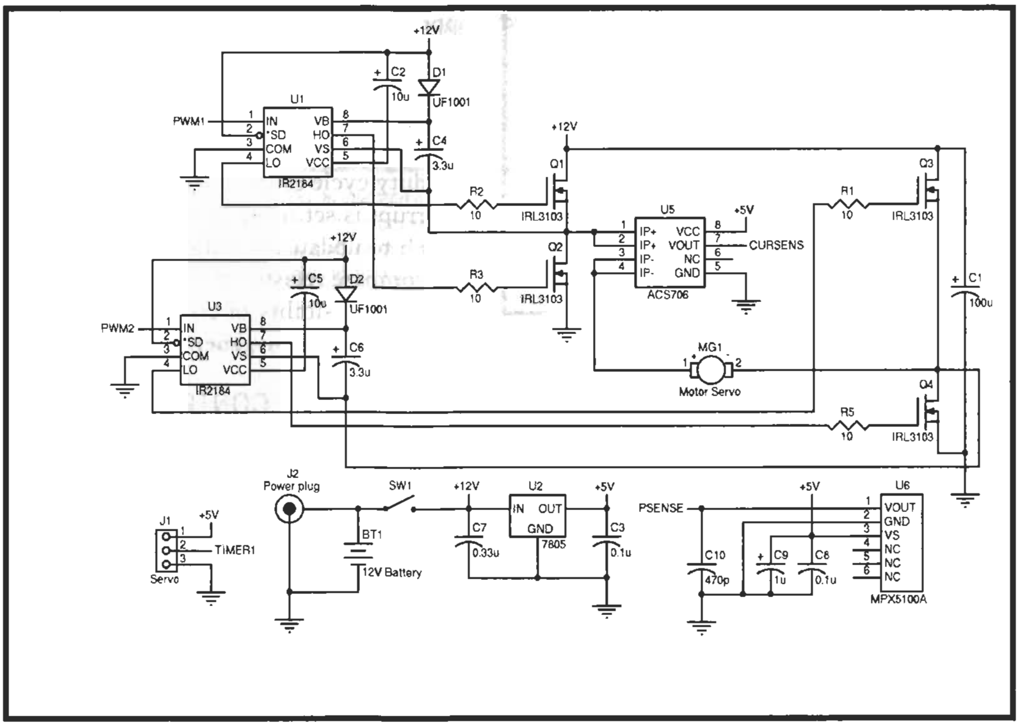


**Figure 2**—*The H8/3664 microcontroller connects to several components that provide control and sensing for the robot. The circuit is basic and easy to connect to most other microcontrollers as well.*

A Hall effect current sensor (Allegro ACS706) is set up in series with the motor in order to measure motor current for control purposes. The sensor's connection is relatively straightforward. It outputs a voltage corresponding to the motor current offset at 2.5 V for zero current, with a sensitivity of 130 mV/A, and has a range of ±15 A. The sensor is connected directly to an ADC port, but it is not enabled for control in software at this time.

We were accidentally shipped a 6-V motor instead of a 12-V motor. The motor worked well for a while, but it eventually burned out. Be sure to purchase a 12-V motor for this electrical design.

## SERVO MOTOR

The servo motor is a standard remote-control hobby servo motor. It has a built-in position feedback loop, based on an internal potentiometer, which enables its position to be commanded directly.

The servo is an extremely easy component to interface with a microcontroller. Power is supplied with a 5-V source and control is provided through a 50-Hz PWM signal. The PWM signal is connected to the FTIOB output port on the H8/3664 through a 7404 inverter, which is used as a buffer.

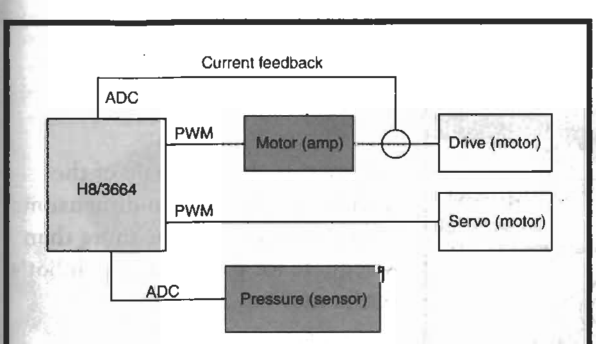


**Figure 3**—*The H8/3664 microcontroller interfaces with PWM output to a drive motor and a steering servomotor. There is input from a pressure sensor and a current sensor on the drive motor, which interfaces through the microcontroller's ADC port.*

## PRESSURE SENSOR

We used a Freescale Semiconductor MPX5100A absolute pressure sensor (with a range from 0 to 16.7 psi) so the inflated ball structure can sense the environment. When the ball hits a barrier, it can be picked up as a pressure transient.

The absolute pressure sensor measures pressure with respect to a perfect vacuum. Thus, the sensor reads both atmospheric and ball pressure. Since atmospheric pressure at sea level is 14.7 psi and the ball may be inflated with 1 psi, the sensor has only a small range that it can sense. However, because impacts of the ball produce both positive and negative changes in pressure, the sensor is effective at detecting collisions. This sensor is easy to interface with the microcontroller: $V_{OUT}$ from the sensor is connected directly to an ADC port. Decoupling and filtering capacitors are recommended in the MPX5100A's datasheet.

## SOFTWARE

The software provides the basic drivers necessary to read sensor information and to control the two motors. The drivers are set up to run in interrupt loops or operate continuously. (Continuous conversions are set up for the A/D sensor readings.) Communication with these low-level drivers is available via function calls. Application functionality is available by pro-

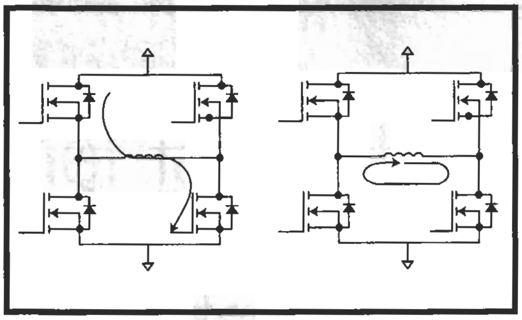

**Figure 4**—*During the PWM on state, current passes through the upper-left MOSFET and then the lower-right MOSFET. When the PWM switches off, current continues through the motor inductance and then through the lower-left MOSFET. To reverse the motor's direction of rotation, the opposite side of the H-bridge is switched.*

gramming the desired robot behavior in the main function and calling low-level driver function calls.

For our robot, the main function contains a simple sinusoidal trajectory of the drive motor and the steering servo motor, followed by a straight trajectory. This trajectory demonstrates some basic capabilities of the robot. Details about the low-level drivers and the complete source listing are posted on the *Circuit Cellar* FTP site.

We experimented with the pressure sensor, but it was not used in the application due to the difficulty of debugging. Sample code is provided, which should theoretically enable the robot to switch from a straight trajectory to a sinusoidal trajectory when an obstacle is hit and thus change direction.

## DC DRIVE MOTOR CONTROL

The code for the DC drive motor and the steering servo motor both use PWM. They're both in the pwm.c file on the *Circuit Cellar* FTP site.

The brushed DC motor must have variable speed and torque capabilities, and it must operate bidirectionally. The motor currently uses voltage control, but it has the capability to use current control. Current control hasn't been implemented in software. The torque in a brushed DC motor is proportional to the current through the motor (related by the motor's torque constant), so it may be advantageous to use current control if you want to control torque.

Voltage control is achieved by a PWM signal to the motor at approximately 10 kHz. This is implemented on the H8/3664 microcontroller using Timer V. Since Timer V does not have a buffered register for changing the duty cycle of the PWM, an interrupt is set up on compare match to update the value in the compare register and eliminate the possibility of erroneous pulses being generated.

## SERVO MOTOR CONTROL

The servo motor is controlled by pulse-width encoding at a rate of 50 Hz. The pulse for standard hobby servos ranges between 1 and 2 ms with 1.5 ms corresponding to the neutral position.

Timer W on the H8/3664 microcontroller is set up to overflow at a frequency of 50 Hz, or a 20-ms period. The period in terms of Timer W counts is 49,152. The compare-match register is set up in buffered PWM mode, so the PWM-output signals can be generated without concern for the timing of writing to the compare-match register. The new value will be automatically loaded at the beginning of the PWM cycle, eliminating the chance for making writes at the wrong time and causing erroneous PWM-signals.

In terms of Timer W counts, 1 to 2 ms corresponds to a GRB value ranging from 2,458 to 4,915, with a neutral position of 3,686. Limit checks are included to ensure that the servo is not overdriven.

A/D conversion for the pressure sensor is accomplished in the adc.c file that's available on the *Circuit Cellar* FTP site. Only one channel is used (AN0) in order to store the value of the current pressure. The ADC is set up to run in continuous operation, and the current value from the pressure sensor can be read at any time.

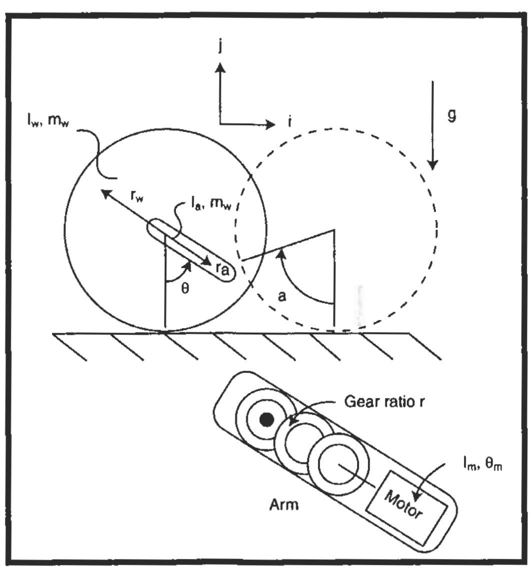


**Figure 5**—*Driving the internal pendulum causes the robot sphere to rotate.*

## MECHANICAL COMPONENTS

In this article, we present our parameters for constructing this robot; however, customization is the spice of life. In order to modify this robot, you should keep a few things in mind. The ideal implementation of this robot has its weight as far away as possible from the robot's driven axle. This allows the robot to convert a maximum amount of momentum or torque into rolling, because the robot essentially moves based on the "falling" of the pendulum. If the pendulum rotates too far, the ball will counter-rotate, so balancing motor torque and pendulum torque is important (see Figure 5).

Simulating the robot's motion can also help. Using the motion equations in Figure 6, different physical parameters and motor-control scenarios can be modeled to determine the optimal design. These equations can be solved using any numerical solver and plotting the output. In order to model the motion in the tilt direction (the plane perpendicular to rolling), the same equations can be simplified and used. However, the simplest and most important analysis of the tilt motion is the turning radius:

$$r_{TURN} = \frac{r_W}{\tan(\varphi)}$$

where φ is the tilt angle of the pendulum. These two-dimensional simulations should be more than adequate for predicting the robot's basic motion.

## MANUFACTURING

We used off-the-shelf mechanical components for this design. Most of the parts can be purchased at your local hardware store.

The mechanical components that you must build must consist of several plates, which when assembled make up a pendulum and the inner-workings of the rolling robot. The pendulum is essentially a box with a series of threaded holes that are used to attach the drive motor and servo-mechanism.

The first step is to cut out the rough shapes of the different plates that make up the pendulum and a custom universal joint. The method of creating these plates depends on the materials you work with. We chose aluminum, which requires machining with a water-jet or milling cutter. However, the body could just as easily be made from polycarbonates or acrylics and cut out with a scroll saw.

Once you cut out the parts, you must drill holes to attach the drive components. Next, tap the drilled holes. Taps can be purchased as a set or individually at your local hardware store. Make sure the taps match the screws used in the project.

| PWM (TMOV port) | I/O (P72 port) | PWM1 | PWM2 |
|---|---|---|---|
| 0 | 0 | 1 | 0 |
| 1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 |
| 1 | 1 | 0 | 0 |

**Table 1**—*A logic circuit provides an interface to a single PWM output and a single digital I/O port on the microcontroller. In this truth table, TMOV and P72 are the outputs on the H8/3664 microcontroller. PWM1 and PWM2 are the inputs to the H-bridge.*

The brackets used to attach the servo pushrod should be made slightly different than the other plates. First, cut out a strip of 18-gauge sheet metal, drill the holes, and then bend the bracket. The part is small enough so that drilling the holes before bending makes the manufacturing process much easier.

The pushrod itself is easily made by cutting a short length of threaded rod and connecting two pushrod clevis pins. The next step is to cut the shafting to length in order to create a drive shaft, gear shaft, and a support shaft. There are several ways to accomplish this. The simplest is to use a hacksaw. Once you cut the shafts to size, carefully round the ends with a file. Any raised burrs will make it difficult to slide anything onto the shaft. Finally, use a die to thread the drive shaft, you should need to thread only about 1" on either end of the shaft.

The final step is to cut two holes into the rubber kick ball. First, determine an axis through the ball. This can be accomplished by deflating the ball and folding it into quarters. The points of the quarters should approximate the location of an axis through the ball. Next, inflate the ball. Using a compass, trace two circles on the ball and use the lock ring from the drain assembly to determine the necessary diameter of the circles.

Lastly, deflate the ball and use a pair of scissors to cut out two holes defined by the circles you have drawn. Make sure you don't cut the ball's plug; otherwise, it will be impossible to inflate.

## ASSEMBLY

Once the mechanical parts are man-

$$\begin{bmatrix} I_a + m_a r_a^2 + I_m & m_a r_a r_w \cos(\theta) + I_m \\ m_a r_a r_w \cos(\theta) + I_m & I_w + (m_a + m_w) r_w^2 + I_m \end{bmatrix} \begin{bmatrix} \ddot{\alpha} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} -m_a g r_a \sin(\theta) \\ m_a r_a r_w \dot{\theta}^2 \sin(\theta) \end{bmatrix} + \begin{bmatrix} T \\ T \end{bmatrix}$$

**Figure 6**—*$I_a$ is the moment of inertia for the pendulum. $m_a$ is the mass of the pendulum. $r_a$ is the length of the pendulum. $\theta$ is the angular position of the pendulum. $\dot{\theta}$ is the angular velocity of the pendulum. $\ddot{\theta}$ is the angular acceleration of the pendulum. $I_w$ is the moment of inertia for the wheel. $m_w$ is the mass of the wheel. $r_w$ is the radius of the wheel. $\alpha$ is the angular position of the wheel. $\dot{\alpha}$ is the angular velocity of the wheel. $\ddot{\alpha}$ is the angular acceleration of the wheel. $I_m$ is the inertia of the motor. T is the applied torque. g is the gravitational constant.*

ufactured and the circuit has been built, you can assemble the components and finish the robot. Assembly is relatively straightforward. After examining the exploded view, drawing it should be an easy task to determine how the pieces fit together (see Figure 7).

The only difficult part of the assembly process involves inserting the mechanical parts in the rubber ball, because some assembly must take place inside the ball. We recommend fully testing the electromechanical assembly before attempting to place it inside the ball.

It is easiest to start by assembling three sides of the pendulum subassembly (M1, M18, and M26). The flanged bearings (M17) can be loosely fitted into the front and back plates, while the back plate spacer (M2) and servo motor can be tightly secured to the back plate. The servo arm is near the bottom of the pendulum. One end of the servo pushrod can be attached to the servo arm at this time and the other end can be attached to the servo pushrod-bracket (M25).

The motor subassembly can be put together by attaching the motor to the motor plate (M22) and the motor plate to the sliding plate (M27). Next, a flanged bearing is loosely installed. Then the bore reducer (M8) is inserted on the motor shaft, a washer is placed on it, and the power sheave (M23) is pressed and tightened onto the bore reducer and motor shaft. After checking that the motor is free to rotate, the flanged bearing can be tightened down. The motor subassembly is attached to the pendulum subassembly by loosely connecting the slide plate (M27) to the pendulum front plate (M18).

The next step is to attach two bearing blocks and the electrical housing (M16) to the universal joint plate (M29). As an additional step, you might want to preassemble the miter gears (M20), drive shaft (M15), and gear shaft (M19) to check the tolerance of the gears and set the alignment of the bearing blocks. Once clearances are checked, the assembly can be returned.

The pendulum subassembly can be finished by attaching the remaining side plate (M26). In addition, one of the bearing blocks can be attached to the back plate (M1) by using the support shaft (M28) with two washers and two collars. Another bearing block can be attached to the front plate (M18) with one miter gear (M20), two washers, the driven sheave (M14), and the gear shaft (M19). Attach the battery boxes with Velcro. The belt (M7) can be placed on the pulleys and tensioned. Then the slide plate (M27) can be tightened down.

Any electrical connections can be made and the pendulum subassembly can be inserted into the ball by slightly stretching the opening. Then the universal joint assembly can be inserted and the drive shaft with the miter gear can be attached. At that point, the universal joint plate can be connected to the bearing blocks on the pendulum along with the servo pushrod bracket.

The final step is to insert the rubber gaskets and the flanged ends of the drain bowls into the ball and bolt the driveshaft to the sink drains with lock nuts. Next, tighten the sink drain lock rings firmly to the ball and apply Teflon tape to the sink-drainpipe connections. Finally, flip on the power switch with a screwdriver, tighten both end caps, inflate the ball, and stand back! (Drawings of each step are posted on the *Circuit Cellar* FTP site.)

## FINDINGS

The rolling robot was an intriguing and difficult project to work on. The mechanical design is novel and nontrivial. However, we found that the

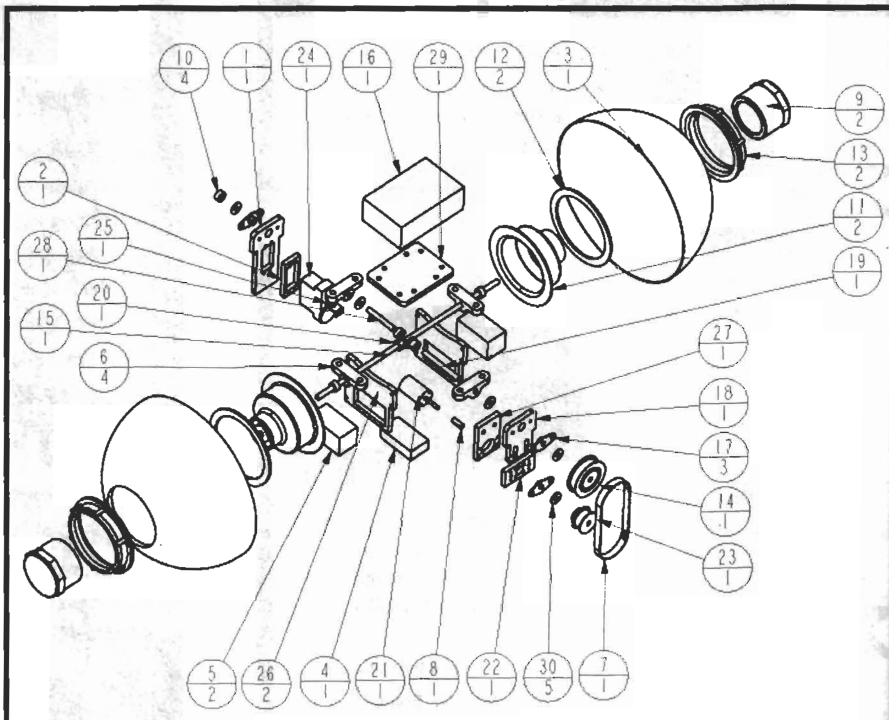

**Figure 7**—*This figure shows an exploded view of the mechanical components of the rolling robot. Numbers in the top half of the bubbles correspond to the parts list that can be downloaded, while numbers in the lower half specify quantity.*

resulting unique motion available with this robot was worth the effort. With that being said, we learned a few things that might be helpful in building a similar rolling robot.

The biggest difficulties were caused by the tight working area inside the rubber ball and not having powerful enough drive components. While assembling the robot inside a rubber ball through small openings is possible, debugging is nearly impossible. Thus, during the debugging process, use a hamster ball or some other clear sphere that allows easy access to the components.

It is also important to ensure that the components used match both the mechanical and electrical requirements. Initially, we burned out the drive motor because we were shipped a 6-V motor, instead of the 12-V motor that we expected. With a little patience and some elbow grease construction of the robot should go smoothly.

We hope that this article encourages you to build a similar robot. We had a lot of fun building ours, and the resulting product is a very interesting accomplishment. Just remember us when you take over the world with your rolling robot army. ■

*Jeff Bingham (bingjeff@gmail.com) has a B.S. in Mechanical Engineering from Idaho State University and is pursuing his Ph.D. in the same field at the Massachusetts Institute of Technology. In his spare time, he likes to build things, such as small autonomous rolling spheres.*

*Lee Magnusson (leem@alum.rpi.edu) has a B.S. in Mechanical and Materials Engineering from Rensselaer Polytechnic Institute and an M.S. in Mechanical Engineering from the Massachusetts Institute of Technology. His interests include electromechanical modeling and design. Lee currently works in the MIT Biomechatronics Lab (http://biomech.media.mit.edu), where he designs the latest in active powered prosthetics.*

## PROJECT FILES

To download the code and additional files, go to ftp://ftp.circuitcellar.com/pub/Circuit_Cellar/2007/200.

## RESOURCE

Freescale Semiconductor,"MPX5100A Datasheet," 2005, www.freescale.com/files/sensors/doc/data_sheet/MPX5100.pdf.

## SOURCES

**ACS706 Linear current sensor**
Allegro Microsystems, Inc.
www.allegromicro.com

**MPX5100A Absolute pressure sensor**
Freescale Semiconductor, Inc.
www.freescale.com

**IR2184 and IRL3103 Half-bridges**
International Rectifier
www.irf.com

**H8/3664 Microcontroller**
Renesas Technology Corp.
www.renesas.com