time. After closing the connectors down onto the ribbon cable, it's a good idea to test for connectivity

You've previewed a digital edition of Make Magazine:

Want to read the entire magazine?

purchase options

browse more magazines

cated to the Chumby's ports, and cut a matching hole in the back of the phone around its original RJ14 jack. Then I used mounting tape to affix the daughtercard to the inside of the phone case, with its ports and jacks facing out the hole.

The metal baseplate of the phone curves up around its perimeter, and its back edge covered the Chumby's recessed power button, located at the bottom of its daughter card. I cut a notch in the metal to allow access to the button (Figure F, previous page). This lets you turn its power off and on with a pen or other small pointed object, which is the way you'd do it anyway; the Chumby is meant to be left on continuously.

Move the Speakers to the Phone Headset

Using a small screwdriver, I easily broke the brittle adhesive that attached the Chumby's speakers to the rear assembly. To put them in the phone's handset, I simply soldered some extension wires insulated with heat-shrink tubing to the RJ14 jacks at each end of the existing coiled phone cable. The cable has 4 wires, so it can drive 2 speakers.

The Chumby's speaker wires are thick and easy to strip and solder, making them a good choice for hacking. Remembering to maintain the polarity, I soldered the pin connectors on the daughtercard for the speakers to the RJ14 jack contacts for the coil cable at the phone end. Then I unscrewed the microphone and speaker ends of the handset and simply lifted the 2 disk-shaped components out.

The Chumby's speakers fit nicely at each end of the handset (Figure G). I wired them to the RJ14 uffed em from

itch anel e with a mby, but o short- rojects. ce switch. he control ker that he hook. forcing me

to concede to a standard panel-mount pushbutton drilled in behind the handset cradle. Soldering the wires was easy, but now I have to disconnect that button whenever I open the phone chassis.

Future Improvements

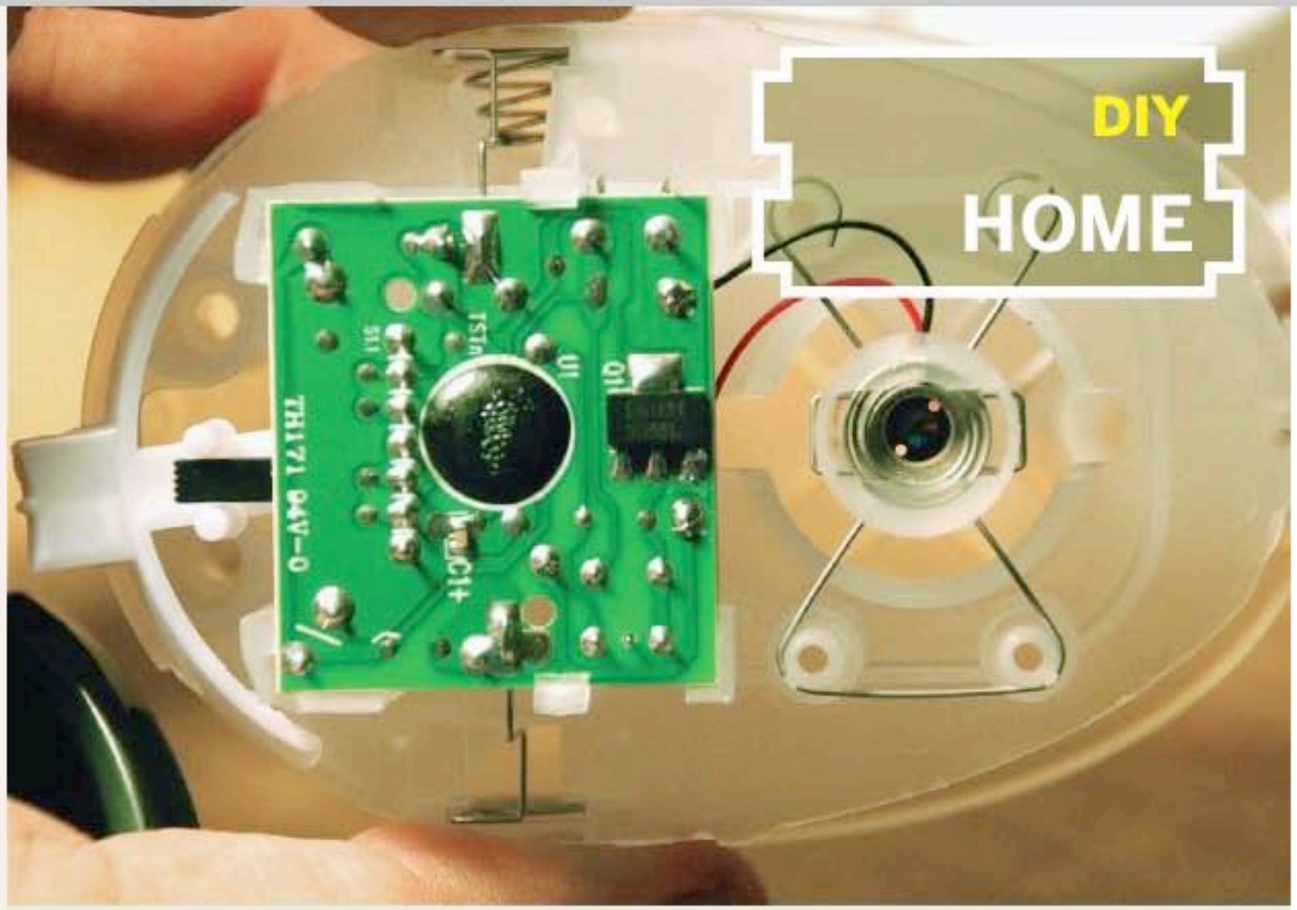
While it's not perfect, it is fun to have a big, retro, important-looking red phone that happens to run Linux, tune internet radio, and display widgets (Figure H). Since the Chumby's schematics and source code are all available, all 3 of the following improvements should be relatively easy work.

» Build an iPod dock into the back of the unit. Chumby can read from a USB-connected iPod (but not iPhone or iPod Touch), and it has a nice touchscreen interface for playing through the speakers.

» Improve access to the power button — likely by splicing out the Chumbilical's leads for that contact.

» Connect a switch and hack software to activate the phone's "hook" buttons. I eventually want the device to launch the internet radio application when you take the handset off the hook. Then, imagine hanging up the red phone and having internet radio automatically turn off. Sweet!

Daniel Gentleman (dan@thoughtfix.com), better known as ThoughtFix, operates two blogs about mobile technology and portable Linux devices.



HACKING THE GLADE WISP

Make your own scent output peripheral from a piezo air freshener. By Wayne Holder

Photography by Wayne Holder

Not long ago, my 11-year-old daughter Belle wanted to create a gadget that would amuse her dog Panda by dispensing different scents for him to sniff. I had no idea how to control the dispensing of fragrances, so we took a trip to the local pharmacy and checked out the electric air fresheners.

Most of them diffused fragrances with heat or fans, but one, the Glade Wisp, claimed to use a microchip to "automatically puff" scented oils into the air. Intrigued, I bought one to see what made it tick.

The Glade Wisp runs off a single AA battery, which powers a vibrating piezoelectric disc that atomizes and disperses aromatic oil in short, smoke-like puffs. The Wisp turns out to be easy to hack — for less than $10 you can make a computer-controlled aromatic atomizer for all sorts of practical and artistic projects.

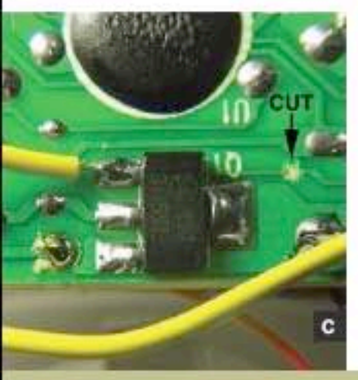
Here's how I modified a Wisp to be controlled by an Arduino board running just a few lines of code.

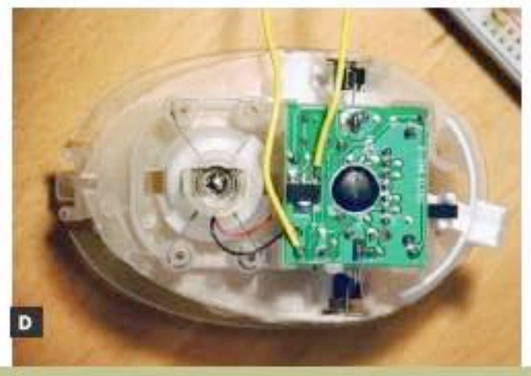
Before you go tearing the Wisp apart, get familiar with how the manufacturer intended it to work. Screw in the scent bottle, remove the red blocking tab from the battery, and watch the unit in operation for a few minutes.

Set the Wisp to its strongest setting, use a desk lamp to illuminate it from the side, and hold a dark piece of paper behind it; you should be able to see white vapor puff out from the top about every 10 to 15 seconds. This will show you what to expect from your modified unit. If you're curious about how the Wisp works, you can read the patent online; see the references for this article at makezine.com/16/diyhome_aroma.

1. Hack in and hook up.

Now let's see what's inside. First, pry off the top shell, starting from the end opposite the adjustment





REWIRING THE WISP: Fig. A: The Glade Wisp Scented Oil Fragrancer. Fig. B: The cover removed, exposing the PC board and scent bottle. Fig. C: The power transistor with its original control trace cut and new control wires attached. Fig. D: The hacked Wisp with new control wires. Fig. E: The Wisp connected to the Arduino controller on a solderless breadboard.

MATERIALS

- **Glade Wisp Scented Oil Fragrancer** There are a few models; I used the one shown here, not the one shaped like a bottom-heavy donut or the more expensive "flameless candle" with the flickering LED.
- **Arduino microcontroller board** Any type that's based on the Atmel ATmega168 chip running at 16MHz will do; I used an Arduino Nano.
- **24-gauge hookup wire**
- **Essential oil, ½oz, fragrance of your choice** The possibilities are vast; Google "fragrance oils" or investigate candle making and soap making suppliers.

TOOLS

- **Computer running Arduino software** from arduino.cc/en/main/software
- **Sharp X-Acto knife**
- **Soldering iron and solder**
- **Wire cutters and strippers**
- **Flat-blade screwdriver**
- **Jeweler's flat-blade screwdriver or drill and small drill bits**
- **Rubbing alcohol**
- **Glue gun and hot glue**
- **Helping hands tool (optional)** with clips and a magnifying glass

switch. The cover is held to the base with snap-fit plastic tabs, so you'll need to get a bit physical and sort of "unzip" the tabs toward the switch end (Figure B). I used a small flat-blade screwdriver.

The base and shell are rather flexible and all the circuit components are attached loosely, so I don't think you can damage the workings unless you slip and rip out one of the wires that attaches the circuit board to the piezoelectric atomizer next to it.

Inspect the PC board and locate a black, rectangular component with 3 leads coming out of one side and 1 big lead on the opposite side. It will probably be marked "3055L." This is a power MOSFET, a type of transistor, and it drives a transformer on the underside of the board, which in turn boosts the battery voltage up to the level needed to power the piezoelectric atomizer disc.

The first step in the mod is to use an X-Acto knife to cut the trace that connects the MOSFET to the control chip that's hidden under the big, round blob of epoxy in the middle of the board.

Severing this connection lets us take over control and vibrate the disc ourselves. With the blob oriented above, I severed the trace just to the upper right of the MOSFET's big lead (Figure C).

Cut 2 lengths of 24-gauge hookup wire long enough to reach from the Wisp to your Arduino board. Connect one wire to the upper left MOSFET lead, and connect the other to the board contact nearest the lower right corner (Figures C and D). Take care not to create short circuits by bridging the MOSFET's pins, and inspect your work carefully. The MOSFET is small, so you may need a "helping hands" tool to assist.

Connect the Arduino's digital output pin D2 to the wire connected to the MOSFET, and connect the Arduino's ground pin to the other lead from the Wisp board (Figure E). That's it for the hardware mods needed to control the Wisp.

2. Program the Arduino.

I used an oscilloscope to probe the Wisp onboard controller's output, and found that it drives the atomization process by generating a 150MHz signal that lasts about 10 milliseconds. We'll program our Arduino board to mimic this signal. The following Arduino code simulates the Wisp's signal pattern, except that it tells the piezoelectric disc to "puff" every 2 seconds instead of every 10–15 seconds.

Reattach the fragrance bottle to the Wisp, then upload and run the code, and if you've wired everything correctly, you should see white puffs coming out of the atomizer immediately.

```
void setup () {
 DDRD = 0xFF;
}

void atomize (char pins) {
 unsigned int ii;
 char kk;
 while (digitalRead(8) == HIGH)
  ;
 for (ii = 0; ii < 2000; ii++) {
  PORTD |= pins;
  for (kk = 0; kk < 12; kk++)
   ;
  PORTD &= !pins;
  for (kk = 0; kk < 12; kk++)
   ;
 }
}

void loop() {
 atomize(0x04);
 delay(2000);
}
```

In the code, the `DDRD` and `PORTD` keywords configure the Arduino's D0–D7 pins as outputs to be controlled directly, by using the Arduino's port manipulation commands. The nested loops in the `atomize()` function toggle the D2 pin (specified by passing in the value `0x04`) on and off a total of 2,000 times, with a very short pause after each change.

The values I chose for the delay loops make the Arduino's output roughly match the Wisp controller's, but with a shorter, 2-second delay between puffs. You may have seen other code that controls the Arduino's digital outputs by calling `digitalWrite()`, but this would be too slow to generate a 150MHz signal.

You can experiment with setting the loops to count up to values other than 2,000 and 12 to see how this changes the atomization process, but note that shorter delay times may not give the circuit's 3,300µF capacitor enough time to fully recharge between puffs, which will result in significantly decreased vapor output.

3. Substitute your own fragrance.

As interesting as it might be to have a computer-controlled air freshener (not very, IMO), the real fun begins when you replace the unit's original fragrance with something more meaningful or exotic, such as, say, the smell of freshly baked cinnamon buns, or hazelnut coffee.

To do this, you need about ½oz of an essential oil in the fragrance or aroma of your choice. The design of this bottle makes it difficult to remove the cap, but it can be done. I used a small, flat-blade jeweler's screwdriver to pry around the cap, pulling it away from the neck and breaking the glue that attached it. You can also drill a small hole in the top of the bottle.

Whichever method you use, you should clean the bottle with rubbing alcohol and let it dry, to remove as much of the old scent as possible. Once you've added your own fragrance, reattach the cap to the bottle with hot glue so it will twist back into place in the Wisp.

4. Further development: Create an aroma orchestra.

Because the Arduino's `PORTD` value lets you write to all of its digital outputs at the same time, one Arduino can control up to 6 Wisps simultaneously. You simply connect each Wisp to a different output pin and pass different values into the `atomize()` function. Using the Arduino programming environment's Serial Monitor feature, you can send keyboard

characters to the Arduino, which lets you create an instrument that plays fragrances, live. Just make sure to avoid using pins D0 and D1, which share their function with the serial port.

For example, the following code reads an input character and uses it to select which of 4 different Wisps to puff. Typing the 2 key commands the Wisp that's connected to pin D2, while typing 3 commands pin D3, and so on.

```
void setup () {
 DDRD = 0xFF;
 Serial.begin(9600);
}

void atomize (char pins) {
 unsigned int ii;
 char kk;
 for (ii = 0; ii < 2000; ii++) {
  PORTD |= pins;
  for (kk = 0; kk < 12; kk++)
   ;
  PORTD &= !pins;
  for (kk = 0; kk < 12; kk++)
   ;
 }
}

void loop() {
 char cc = Serial.read();
 switch (cc) {
 case '2':
  atomize(0x04);
  break;
 case '3':
  atomize(0x08);
  break;
 case '4':
  atomize(0x10);
  break;
 case '5':
  atomize(0x20);
  break;
 }
}
```

To set this up, load the code to the Arduino board, then click the Serial Monitor button, which is the rightmost button at the top of the Arduino's development environment.

Olfactory Displays

Scentovision (1939) movie theater scents introduced at New York World's Fair
Aroma-Rama (1959) movie theater scents from *Behind the Great Wall*
Smell-O-Vision (1960) movie theater scents from *Scents of Mystery*
Odorama (1981) movie scratch-and-sniff cards from John Waters' *Polyester*
iSmell (2001) computer peripheral developed by DigiScents, never marketed
Scent Dome (2004) computer peripheral from TriSenx, trisenx.com
Fragrance Communication System (2005) networked aroma emitters and services for homes, hotels, and theaters, in development at NTT Communications, ntt.com
CineScent (2006) movie theater scents, in development, cinescent.com

This will display a set of controls near the bottom of the window. Select 9,600 baud, then type a number (2–5) into the text box and press Send. This should trigger the corresponding pin, and the Wisp it's connected to.

By using a variation of this code, you can program a set of Wisps to play your own home-theater version of John Waters' notorious Odorama, triggering a "smelltrack" that's synchronized with pictures or scenes of different aromatic subjects appearing onscreen. A small fan that blows the aromas toward the audience might be helpful here.

With some additional work, you can create more advanced scent-enabled applications, such as a network-controlled aroma generator that could receive fragrance-based "mood" messages. It's all up to your imagination and ingenuity.

Meanwhile, Belle and I will be busy working on the device she's designing for Panda, but we hope you'll have fun with this idea, too!

Find the project code and other resources at makezine.com/16/diyhome_aroma.

Wayne Holder has a classical education in computers, tinkering, and building.

Photograph by Ed Troxell

USB MOTION DETECTOR

Turn your PC into an ambush multimedia presenter. By Ken Delahoussaye

Gone are the days when people's interest could be held by simple radio or television. Today we're bombarded with information and we crave interactive experiences that don't waste a single second of our time. Advertisers recognize the difficulty of presenting messages that cut through the clutter, and they've come up with creative ways to capture our attention.

One example: the multimedia kiosk, now common in shopping malls, movie theaters, and airports. Complete with an internal computer, sound card, and video graphics monitor, these dazzle stations can be a powerful advertising tool — especially when they have motion detection circuitry that triggers a video presentation at the precise moment an unsuspecting patron comes near.

This article explains how to construct a USB motion detector that will give your computer this hey-you ability, using a free Windows presentation applet I wrote, USB Multimedia Presenter, so that you can start your own kiosk advertising campaign. You can also use the setup for practical jokes, or just to amaze or amuse your friends.

To interface between the detector and computer, I used an off-the-shelf USB device which requires no drivers to install, since it uses existing Windows drivers. The detector draws all the power it needs from the computer, which further simplifies things. All the parts for the project are easy to find, and if you have basic soldering and mechanical skills, you can put it together in a single evening.

Connect the Detector to the USB Interface

Drill a 3/16" hole in the top of the motion detector enclosure, to accommodate the USB cable. Center