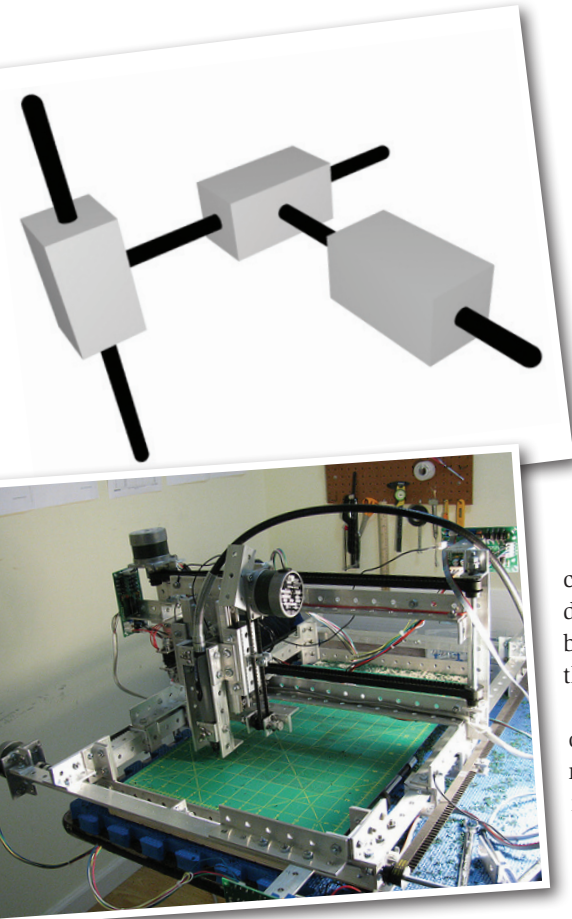# Projects on the Move

**Free/Open Source isn't just about software, it's also about hardware. Open hardware projects are proliferating, and just like open source software, you can get started inexpensively. Here are three projects that are fun, useful, and suitable for beginners to embedded programming, robotics, and building electronic devices.** *By Carla Schroder*

The speed bump for Free/Open Source has always been hardware. Open, freely shareable code and a free open Internet are two of three essential components of the FOSS world. The third is open hardware, because, obviously, all that great software needs hardware. To me, it is silly to encumber any hardware with secret specifications, closed proprietary firmware, non-disclosure agreements (NDAs), or any kind of "intellectual property" nonsense. Hobbyists, hackers, and do-it-yourselfers are drivers of innovation and improvements, although in this new era of "the customer is the criminal," that message is often lost.

Happily, open hardware has seen a small explosion in growth. "Open" covers a lot of ground: licenses, specifications, hardware description language (HDL) code, schematics, and more. Please see the good Wikipedia article "Open-source hardware" for more information on the details of what open hardware means [1].

Everything seems to be controlled by microchips these days, and microchips need to be programmed. These three projects introduce the fundamental concepts and skills of embedded programming and robotics while being fun and inexpensive.

## MiniPOV: Programmable Blinky Lights

The MiniPOV version 3 [2] is awesomely fun. The programmable blinky LED (light-emitting diode) controller might be one of the all-time best introductions to hacking electronics and programming microcontrollers. POV stands for "persistence of vision," which refers to the afterimages retained either by our eyes or brains (controversy rages over which) for about 0.04 seconds, especially of lighted objects. You've probably experienced this with camera flashes and car headlights. POV gadgets take advantage of this to combine movement with flashing LEDs to create animated messages and moving images. A video of a 96-LED POV hack is an impressive example of the possibilities of these devices [3].

As the MiniPOV website states, the project is designed for people who want to learn how to solder, assemble simple electronic kits, program microcontrollers, and make blinky things.

MiniPOV v3 is identical to the MiniPOV v2, except it connects to your PC via a DB9 serial port; version 2 uses a parallel port. Why parallel port, when many modern computers don't even have one? Because it sends signals at 0-5V, which is just right for the Atmel microchip. The serial port sends about 10V. The MiniPOV v3 uses 5.1V zener diodes to reduce those 10V to -0.5V to 5V.

A lot of modern PCs don't have serial ports, either. But, don't worry; you can use a USB-to-serial adapter. You can get the MiniPOV kit and serial adapter for about $30.

## CARLA SCHRODER

**Carla Schroder** is a gadgets enthusiast and Linux expert who has written *The Book of Audacity*, *Linux Networking Cookbook*, *Linux Cookbook*, and hundreds of Linux how-to articles.

The MiniPOV creator, MiT engineer Limor Fried, has been called the "Mac-Gyver of the Day" [4]. She has created several do-it-yourself electronic kits that are sold on the Adafruit Industries website [5], and all specifications are completely open.

One design is not available in kit form: the Wave Bubble RF (radio frequency) jammer. Ms. Fried explains:

*In contemporary Western society, electronic devices are becoming so prevalent that many people find themselves surrounded by technologies they find frustrating or annoying. The electronics industry has little incentive to address this complaint; I designed two counter-technologies to help people defend their personal space from unwanted electronic intrusion.*

The Wave Bubble jams cell phone frequencies, Bluetooth, and WiFi within a 20-foot radius. This design is available online and is distributed under the Creative Commons 2.5 Attribution/Share-Alike license [6]. Do please note that, despite the awesome pleasure and benefit to society of silencing nattering twits, cell phone jammers are illegal in the United States.

Other fun projects can be found on Limor's website [7], such as The Bedazzler – a do-it-yourself, hand-held LED incapacitator that induces nausea, the TV-B-Gone LCD TV zapper, SpokePOV (blinky lights on your bicycle wheels), and the x0xb0x bassline synthesizer.

## Contraptor: DIY Desktop Fab and 3D Printing

If you like Erector sets, Meccano, or Legos, you'll feel right at home with Contraptor, a do-it-yourself fabricator, an open source construction set, a 3D printer prototyper, and a quick prototyper of Cartesian robots. Each different Contraptor kit contains a pegboard and various structural parts (depending on the kit), such as perforated angle irons, rails, bearings, screws, shaft mounts, drive belts, pulleys, and mounts [8].

The structural pieces are only half the story. The other half is making them move and do something – for example, drill holes or cut and shape an object. For this, you need motors and a programmable microcontroller, which are also available on the Contraptor kits order page.

What are Cartesian robots? They are industrial robots whose parts move on three linear axes at right angles to each other (see the top lead image; courtesy of Wikimedia Commons [9]). This type of robot is standard on modern assembly lines. The contemporary twist is the CNC (computer numeric control) machine. In the olden days, skilled machinists controlled the machines by hand. Now it's all done with computers.

So, what does this mean in real life? It means when you have a cool new design idea you can inexpensively whip up a prototype in hours instead of days or weeks. Look at the bottom lead image of this article: The structural parts of this fast driller, which can cut about 20,000 holes in less than six hours, are Contraptor pieces, and the moving parts are controlled by motors that are in turn controlled by an Arduino Duemilanove microcontroller. The Arduino is programmed from a PC. Once upon a time, this would have been a lengthy process costing tens of thousands of dollars. Now it takes just a few hours or days and a few hundred dollars.

The steps from idea to prototype go like this: (1) Create your design in CAD (computer-aided design) software, such as Eagle CAD; (2) generate G-code from CAD; (3) send G-code to the controller firmware.

The signals go to stepper boards, and your awesome Cartesian robot moves and does stuff. G-code is the most commonly used language for CNC machines. And that, in a nutshell, is a simplified overview of modern mass production.

This relatively inexpensive process has many practical applications. The Contraptor website has videos and project examples like an XY plotter, the fast drilling contraption, and a mini-router that uses a Dremel tool to cut precisely shaped and positioned holes in a board. If you're not ready to spend money, a lot of the Contraptor modeling is done using Google SketchUp. You can start new designs or play around with the many freely available Contraptor models hosted on Google 3D Warehouse [10].

What about 3D printing? This part is cool and exciting, because 3D printers are getting cheaper and better. You don't have to route or carve your inspirations because you can print them. Of course, "cheaper" is relative because even 3D printers for home and small-time users still cost several thousand dollars, and
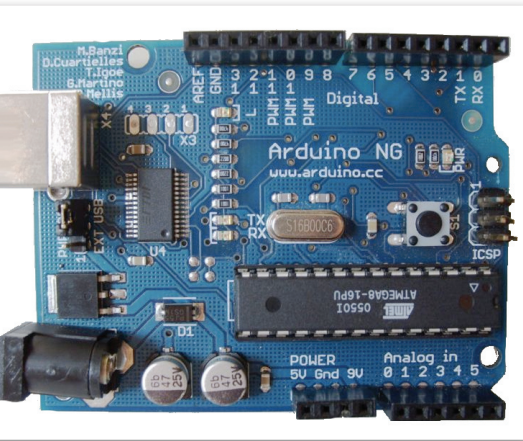
**Figure 1:** The tiny Arduino NG, with USB port and ATMEGA8 microprocessor.

## INFO

[1] Open source hardware: *http://en.wikipedia.org/wiki/ Open-source_hardware*

[2] MiniPOV project page: *http://www.ladyada.net/make/ minipov3/index.html*

[3] 96-LED POV: *http://hackedgadgets. com/2008/11/28/96-led-pov/*

[4] MacGyver of the Day: Limor "Lady-ada" Fried: *http://lifehacker.com/#!5481197/mac gyver-of-the-day-limor-ladyada-fried*

[5] Adafruit Industries: *http://adafruit.com/*

[6] Wave Bubble design: *http://ladyada. net/make/wavebubble/index.html*

[7] limor: *http://ladyada.net/*

[8] Contraptor kits: *http://www. contraptor.org/order-beta-kits*

[9] Cartesian coordinate robot: *http://en.wikipedia.org/wiki/ Cartesian_coordinate_robot*

[10] Google 3D Warehouse, Contraptor models: *http://sketchup.google.com/ 3dwarehouse/search?q=contraptor*

materials are not cheap. Three-dimensional printing starts with a computer file that you (or someone else) create. The printer then "prints" a physical 3D object one cross-sectional layer at a time. A number of different technologies achieve this. An inkjet printing system lays down resin and binder layers and is the only one that can print in color. Other systems use lasers, gels, metals, waxes, liquid polymers, and plasters. The RepRap (replicating rapid prototyper) project is the only FOSS 3D printing system designed for hackers, hobbyists, and community organizers. It works nicely with Contraptor and Arduino. RepRap is designed to replicate itself for fast, cheap production and distribution. Once the plans are created, the only cost is time and materials, because models can be easily distributed over the Internet.

## Arduino: The Beginning

I don't know if Arduino was the first major embedded open hardware project, but I believe it is the most successful, thanks to great design, low cost, and endless flexibility. A basic board is around US$ 30, and US$ 85 buys a bundle of goodies, including LEDs, a small motor, a printed manual, relays, resistors, batches of circuits for all occasions, and other fun stuff. If you don't want to fork over US$ 30 for a ready-made board, the plans are freely available under a Creative Commons license to assemble your own, which would be a great learning experience.

A number of Arduino boards exist for different uses, all fun and inexpensive. The LilyPad is shaped like a disc and is designed to jazz up clothing. Add multicolored LEDs and sound, and turn yourself into a walking multimedia spectacle. The Fio is a wireless controller for all kinds of remote-control mischief… I mean practical, serious applications. Arduino Uno is the latest release of the basic Arduino USB board, and the Mega2560 has twice the memory. Arduino boards control robots, CNC machines, toys, and all kinds of crazy gadgets that people invent.

The genius of the Arduino is a large variety of add-on modules, called shields, and no soldering is required – everything plugs into a solderless breadboard. The Arduino communicates to your PC via USB. It's a tiny little thing that fits into the palm of your hand (Figure 1), but it is large in possibilities.

The brains of the Arduino are its Atmel ATMEGA8 and ATMEGA168 microprocessors. These are single-chip RISC AVR architecture, which was first developed by Atmel in 1996. When you get into embedded programming you'll see AVR a lot. If you're wondering what AVR stands, however, for you won't get a definitive answer. Some say it stands for the initials of its creators, Alf(-Egil Bogen) and Vegard (Wollan)'s RISC processor. Programming these chips means learning the AVR instruction set.

An eye-opener for me was finally understanding that a microprocessor of any kind sits on a circuit board; multiple pins connect it to circuits, which in turn connect it to other processors and device ports. Every one of those pins has a specific purpose, and when you're building something with one of these chips, you must know what each pin is for. The Atmel controllers are integrated processors that include Flash (non-volatile read/write) storage, EEPROM (electrically erasable programmable read-only memory), and SRAM (volatile static random access memory). This means they are self-contained: programs are stored in Flash memory, long-term storage is in EEPROM, and SRAM is for current operations.

Arduino comes with a nice IDE (integrated development environment) that eases you into learning the AVR instruction set. Versions are available for Linux, Mac OS X, and Windows. The steps are pretty much the same as for any programming: First, you write your code, compile it, and transfer your new program to the Atmel microcontroller. Second, you test, debug, and fix as many times as it takes to make it work. Arduino programs are called sketches. The Arduino IDE is pretty nice, with a text box for writing your code, a message area that acts like a live log that gives feedback during coding, and a console that displays activity and error messages. The Arduino website is chock full of examples and instructions.

## YouTube for Inspiration

If you need some inspiration or want to see what other people are doing, just visit YouTube. You'll find dozens of how-to videos, as well as amazing gadgets and inventions. ■■■