# Technology Focus by Richard Cadena

## This month: Love, Tech-Style

"I got through most of the 118-page book in one sitting. What I found is that much of what you can learn from Arduino can be directly applied to entertainment technology."

**I've fallen in love all over again. The object of my affection is a very old friend that I've known ever since I was a kid. She once held my complete fascination, but her allure faded over time. But that's changed now . . .**

Now I call her my favourite pastime, but you might know her as the electronics retailer Radio Shack. I used to spend hours perusing her isles, examining her many parts, and imagining the projects that we could build. It was intoxicating. An integrated circuit chip or two, a bridge rectifier, a handful of resistors and capacitors, a project box to put it in and maybe a few buttons and switches could fire the imagination like no other group of inanimate objects. Throw in an LED indicator or two and there was no end to what you could build - a ring oscillator, or phase shifter, maybe even a pre-amp to drive your guitar into some sweet distortion. As soon as there was enough money in the bank, I would beat a path to her door and lay it on the counter.

But that was the analogue world, long before digital gained sovereignty and cast its long shadow across the store's many thresholds. It took a while, but the takeover of digital gradually changed the landscape, eroding the domination of the many components that feed a techie's dreams, diminishing the fascination that tinkering held for me. Where discrete components once filled most of the shelves in the store, now there were finished products. RC cars instead of servos, motor drive chips and radio kits; cell phones instead of crystal oscillators, inductors, and power supplies; flat panel displays instead of vacuum tubes, rabbit-ear antennas and tuners. It was hard to get excited about someone else's dreamt-up finished product instead of dreaming up your own. The thrill was gone.

Now the romance has been rekindled, thanks to Arduino and Raspberry Pi. Those are not the main course and desert at an Italian café, but two open source computing platforms that are fueling the dreams of techies, hackers, designers, makers and artists alike. These platforms are the current-day equivalent of yesteryear's Heathkit (**http://bit.ly/Heathkit**). They allow you to experiment with firmware in the same way we used to experiment with hardware. So now, whether you're an audio, lighting or video tech, designer or artist, you can learn about and design circuits and apps using current technology in a very simple and inexpensive way. In addition to the documentation, which includes instructions, examples and books, there is a dedicated community of hackers who are happy to help newbies by sharing their projects and knowledge with the community.

Arduino has been around since 2005, and Radio Shack has been selling them since 2011. Ever since then, I've been itching to take it for a spin. I finally got the chance over the holiday break and spent most of Christmas day getting up-to-speed with the basics. With the book *Getting Started with Arduino* by Arduino co-founder Massimo Banzi in one hand, laptop and Arduino kit at the ready with the other, I got through most of the 118-page book in one sitting. What I found is that much of what you can learn from Arduino can be directly applied to entertainment technology.

Learning Arduino teaches you a lot about both hardware and software, which exactly reflects the current state of the entertainment industry. It used to be that every device used hard-wired, dedicated hardware with little or no software. Now that's been turned on its head: many of the new products being developed today in or out of the industry are computers running application-specific software that define the identity of the device.

Take your smart phone, for example. It's a computer running software that allows it to be used as a phone, camera, Rolodex, map, video display, calendar, clock, newspaper, book, calculator and more - depending on the app it's running. In the entertainment industry, many of our devices are computers that become consoles, media servers, visualisers, protocol translators, etc - depending on the application software running on the machine. Some, like media servers, look just like a computer, and others, like consoles, don't look like computers but at their core, that's just what they are. Many of the tools we use - like CAD, visualisers, consoles, media servers - can now run on and/or interface with a laptop and/or tablet. Software is becoming more important in our industry and more products are little more than a bit of hardware and a lot of software.

Some of the best product designers and engineers today are software programmers who earn their keep with a computer keyboard rather than with a soldering iron. As Banzi points out in the book, a computer is "one circuit" with "a thousand behaviours." The behaviour is dictated by the code put in it by a programmer, and it can change with the press of a button or key.

What can you do with Arduino? Almost anything you can imagine. It was originally conceived as a teaching tool to teach students how to programme a computer. And it is a great teacher. It uses a programming language that is based on C and C++, which are some of the most popular computer programming languages in use today. If you

can programme in C or C++, you have a marketable skill. They are powerful languages but the examples in *Getting Started with Arduino* are very easy to follow and very easy to understand. You'll be coding, flashing, fading, sensing and learning in no time at all!

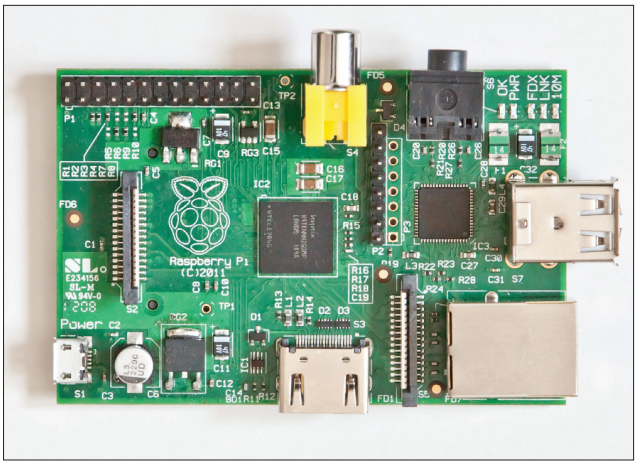
**And software is not all you'll** learn about. There are several of the kits with various parts and components offered on the Arduino website (**www.arduino.cc**) and through other retailers. These kits are very reasonably priced and typically come with the Arduino microcontroller, a breadboard (a circuit board that allows you to quickly and easily make temporary connections for the purpose of testing a circuit design), some jumper wires, LEDs, resistors, switches and various sensors. That alone is enough to help you learn about the basics of hardware, software and concepts like pulse-width modulation (PWM), which is commonly used to dim LEDs. The sensors that are included in some of the kits can help bring you up to speed on the "internet of things," where everything is connected to the internet and can send information about itself like its location, orientation, temperature or anything else that can be sensed. You can also spend a little more money and buy very reasonably priced stepper motors to learn how to programme them to do anything from pan and tilt to precision positioning.

**There are also a variety of** 'shields' or modules that you can buy and interface with the Arduino microcontroller to add more functionality. You can connect it to a network with an Ethernet or WiFi shield, add touchscreen capabilities, give it mobile phone capabilities with a GSM shield, control stepper motors with a motor drive shield, play MP3s with another shield, add multiplexing, a joystick and more. There is even a third party DMX shield to add DMX control capability to the Arduino.

**To programme an Arduino** you have to connect to a computer running software called the Integrated Development Environment (IDE) via USB. The IDE application is where you write the programming code, compile the code (basically, turning it into 0s and 1s so the computer can understand it) and upload it to the Arduino through the USB cable.

**There's another open source** computing platform called Raspberry Pi, and Radio Shack just started selling it around Christmas time. It's similar to Arduino except it's more like a minicomputer with its own operating system, external storage using a micro-SD card, and built-in Ethernet capability. It also has a graphics processor and HDMI output. You can even plug in a keyboard and monitor and use it as a full-blown computer. Using Raspberry Pi is "significantly more complex" than Arduino, according to an online article titled *Arduino vs Raspberry Pi: Which is the Minicomputer for You?* (**http://bit.ly/1aaiMrn**) written by James Bruce.

**If you have little or no** experience with Linux, C or C++, or programming, then Arduino is a great place to start before you step up to a Raspberry Pi. Arduino has lots of easy-to-follow tutorials, fun projects, and a very active online community who are happy to help you get past sticking points - and chances are, there will be at least some sticking points or minor tripping points in the beginning. If you already have some programming skills and you're ready to take it to the next level or you're ready to develop your idea into the 'next big thing', give Pi a try.





Top: The Raspberry Pi.

Above: The Arduino Uno kit includes the Arduino, breadboard, various components, battery case and USB cable. Starbucks *not* included.

**Either way, you'll have fun** and learn at the same time. Computers and control systems are changing so fast that it's a good idea, regardless of your job title, to keep up with the technology. These two platforms are a great way to teach yourself or someone else, and they will open you up to new worlds. Take a look at the Arduino or Raspberry Pi website, or better yet, stroll down the aisle of a Radio Shack. If you're a true techie then you'll fall in love all over again.

> //arduino.cc
> www.raspberrypi.org