

# Maker Faire and (disconnected) Things

I attended [Maker Faire](#) last month. It was my first full Faire though I had been following [Make Magazine](#) from the beginning. For me Maker wasn't so much about the projects as my personal sense of hacker culture in which we can learn by doing and are driven by curiosity rather than business models.

I grew up in a "maker" culture with my father being in business for himself manufacturing products for the TV repair industry among others. I have early memories of the time he manufactured arcade games. I remember Electronic Row on Cortland St in New York and Radio Shack from the days when you'd go there to test vacuum tubes.

The Faire takes a broad view of projects ranging from crafts to sophisticated electronics. It made me think about what is different about electronics and, more to the point, about software and how the concepts of software color my view of the faire.

Electrons make it easy to build systems that are components in larger systems. Software goes even further in using bits as an alphabet. Unlike electrons bits are simply representations of ideas rather than objects with physical limits. We can look at the things we make as ends in themselves. I look at them as starting points for how much more I can do with them.

In junior high school in 1963 I learned to program and I found software resonated with me in a way that hardware projects did not.

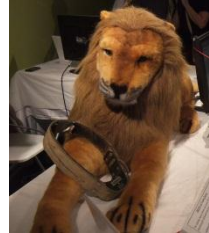
Writing a program to generate prime numbers is not all that removed from building a circuit to make lights blink but I could build far more sophisticated "things" and share them with others. By the time I finished high school in 1966 I was doing production software for online information systems.

How do we get the same kind of leverage using hardware components?

I find that I really enjoyed building things other people use but I've never been satisfied with "just software". This is why I went from mainframes to personal computers as they became available and today I'm excited about the new generation of devices that allow me to use my software skills to make things and make things that do things.

One of the most interesting projects at the faire was a collar for lions (by [Ground Lab](#)) which combined a GPS with a cellular radio and also allowed for downloading Python

code. In effect each lion has a cellular phone number. This gives new meaning to "roaming charges". Fortunately for the lion the cellular carrier is giving them free service.



Taking advantage of cellular technology is the right approach for this project but longer term we need to avoid having to depend on a provider for community infrastructure.

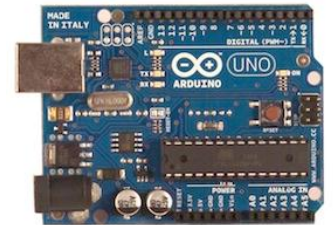
If we are to achieve a real "Internet of Things" we need to understand that the power of the Internet comes from allowing us to focus on our application and not all the stuff in the middle. It's hard to do interesting things if each chip requires a billing relationship with a provider. Worse, each chip must justify its value to a provider.

Fortunately lions are valuable and we can find money to track 2000 lions. The charity model doesn't scale for health care – we shouldn't have to justify the need to connect each medical device one-by-one. This is why we need infrastructure rather than a choice of services.

We have a number of efforts to connect sensors and devices but these tend to be within communities. They solve their local problems but it's difficult to address the larger issue until we can connect devices at any scale.

I've written about this issue in the framing of "[Ambient Connectivity](#)". We need to shift our thinking from a concept of communicating by negotiating a path to the concept of relationships independent of the path.

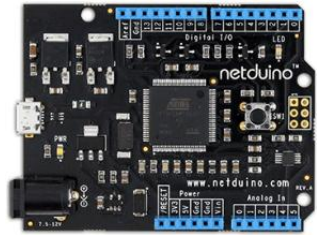
It is an architectural model. We can act as if we have simple connectivity even if we have traditional telecom paths in the middle. Once we've paid for the connection we can act as if it were infrastructure. But it isn't infrastructure so we must not confuse our work-arounds with a full solution.



One of my reasons for going to the Faire was the [Netduino](#). The Netduino was inspired by the [Arduino](#). I am excited by the Arduino because it takes this hardware and democratizes it by aiming at those whose focus is on a project like an animated sign rather than on the electronics itself. It democratizes the technology.

The idea of small programmable devices is not new. They are exciting for those of us who know something about hardware but shift the emphasis to adding capabilities with software. [Parallax](#) with their BASIC stamp has been making such devices for nearly 20 years.

The Arduino [open source hardware](#) experience is related to Chris Anderson's philosophy. The goal is as much to build a community as sell individual devices. Others can add to



the community as they make their own devices. This is not an entirely new idea – Radio Shack in the 1970's found that the existence of competitors helped create a category (electronics hobbyist stores). Open source is another example

of shifting business models towards creating a community.

The Netduino is similar to the Arduino but uses Microsoft's "Micro Framework" and is programmed using the C# language. Whereas I see the Arduino making hardware capabilities accessible without requiring strong hardware or software skills the Netduino strikes me as a gift to those of us who want to take advantage of our software skills.

A full discussion of the choice of programming language and capabilities is beyond the scope of this essay. It's not just the programming language itself but also the hardware and software environment.

Microsoft has heritage as a software company providing tools for the early personal computers. We can see this in the approach it has taken in building C# and the .Net environment which has a set of tools and capabilities that can be mixed and matched. You can get access to native capabilities of the hardware while getting the full capabilities of modern programming language

In addition to the Netduino Microsoft has created its own building blocks. The micro framework was originally used for Microsoft's SPOT watch and meant for creating accessories for PCs but those products did not fare well for want of the right business model. By making those tools available to hackers I expect to see some very interesting results.

The [Mono project](#) is an open source effort built on C# and tools. It can leverage Microsoft's tools or be used without any Microsoft dependency.

Personally I'll use whatever tools I need to get a job done and I find that C# gives me a lot of leverage and the tools provide dynamic assistance. This gives me the freedom to explore ideas and experiment.

Maker Faire and (disconnected) Things/[Bob Frankston](#) 2

One part of the Arduino community is the "shields" or accessory boards that add capabilities. Many of these also work for the Netduino. The shields include boards that allow for simple wireless communications among the boards.

While there were devices at the faire that allow one to "print" physical objects it's hard to make a device like a "watch". I put the word "watch" in quotes because it's not really about time. It's about the form and placement on the wrist – a location that makes it easy to quickly view and reach the device.



In 1993 I wrote a memo at Microsoft I titled "seize the wrist". I was excited to see the Texas Instruments [watch](#). It's what Microsoft's [SPOT](#) watch should've been. There have been a number of wrist-based platforms but the TI piece is useful out-of-the box so it's useful and then I can learn what more I can do with it.

Sony has just announced their [LiveView](#) as a display surface but it's not a hackable device in itself but still exciting. These are not really timepieces as much convenient and connected display surfaces.

The Ti watch uses a simple wireless protocol so you can communicate with the watch. With simple software to relay packets we can connect the device to the world.

What will I do with the "watch"? I don't know. That's the importance of lowering the barrier to hacking and experimenting. Too much of what we call innovation is associated with creating companies and investors. We need to put greater emphasis on creating opportunity to explore and discover. How else will we find answers to questions we haven't even formed yet? Sure a device like the X-Prize can spur some innovation but only along a path already laid out. This is why VC's invest in people who can find new paths.

Even better is when individuals driven by their own needs and curiosity can discover answers to questions people are not yet asking.

This is why leverage is so important. The Netduino in particular builds on billions of dollars of development so I can do an interesting project over a weekend. That's what I've been able to do with software and I'm now increasingly able to move beyond the glass screen and into the physical world.

This is why I'm excited about "open" consumer hardware to include the open Rumba. People have also discovered how to write programs for Canon cameras even if there is no official support. The dirty secret is that all of these devices are computers with standard backplanes and inter-

11/22/2010 11:29

faces and often have diagnostic interfaces needed by the manufacturers for diagnostics and updates. Open APIs are even coming to cars!

These devices in isolation are very interesting but become far more exciting when they start forming relationships. I want to write a simple program that can run on my cell phone to display incoming messages on the watch. What is important is that I can implement this myself and discover what else is possible. I don't need to wait for a manufacturer to decide to market it as a feature. If I do it myself I can learn from experience and rapidly improve the capability.

The key is simplicity – the ability to add such software myself as one of my personal projects. And then I can share it with others without the need to hold it back in order to monetize it.

Today it's hard to experiment with connected devices because it takes so much effort merely to exchange bits. We tend to solve problems within local communities or islands.

This is why I'm putting such emphasis on shifting from telecommunication companies providing services to doing it ourselves. It's the difference between being dependent upon railroads and driving using roads and paths. Knowing that we have roads and sidewalks provides us with the opportunity to travel and exchange goods.

Business or funding models are important. Today's telecom model funds infrastructure by selling high value services. We need a different approach that allows us to focus on the application at hand rather than being limited by a service provider.

I write about this as [Ambient Connectivity](#). The first step is to fund infrastructure. Once we can assume an infrastructure we can then focus on what we do with the opportunity.

And once we can assume connectivity infrastructure we can then focus on taking advantage of relationships. How do we connect a light switch to a light if we are no longer restricted to using a wire? Even better if we could take the same approach whether we put a physical switch on the wall or implemented it as software.

It also brings us back to the real, physical world around us, to the chair I'm sitting in. Think about the motorized seats in your car. Small motors are a great example of what one can do with devices – they don't just make it easier to move the seats, they allow us to use software to control them. Imagine a next generation that has sensors that work to adapt the seat to me and then share the position infor-

mation with the seat in another car. This is not Web 2.0, its reality.

The concept of telecommunications has a pervasive corrosive effect. Returning from Maker I was at JetBlue's T5 terminal at JFK. But I couldn't just connect my devices. I had to login to a web browser simply to click on a screen designed by a lawyer to agree to something I didn't read. I can't use an Arduino or my TI Watch because some lawyers confuse the web with the Internet connectivity.

If we are to achieve the potential of a connected world we need to fund infrastructure as infrastructure and understand that there is a very wide world beyond the web.

One last observation – the attendees at the Faire didn't reflect the "color" of New York. Few Asians, Hispanics, people of color. This is an important issue since the curiosity evinced by Maker attendees is the kind of skill that lays the ground work for success in a changing society.

## Addendum

From the NYT: [With Kinect Controller, Hackers Take Liberties](#)