Eben Upton: Raspberry Pi

Charles Severance

Eben Upton describes why and how he formed the Raspberry Pi Foundation.

In an era where everything is less expensive and smaller, the actual hardware components in our phones, televisions, laptops, game systems, and even thermostats are increasingly buried deep inside sealed glass, plastic, or aluminum cases. Our current technology's designers want it to melt into the background and become an essential yet invisible part of our daily lives—a stark contrast to the technology we used in the 1980s, which was bulky, with lots of parts connected by a snake's nest of cables.

Back then, young people could find their way into programming purely through natural curiosity if they played with the technology they found in their homes. But with today's completely smooth, totally locked up technical marvels, there's very little curiosity among the younger generation to learn more about what's going on "inside."

Eben Upton of the Raspberry Pi Foundation has spent the past eight years trying to produce a $25 (or cheaper) computer specifically designed to show young people what's inside and to inspire them to write programs to, say, control a microwave oven, manipulate a thermostat, or even create their own video a game. To watch the full interview, visit www.computer.org/computingconversations.

Going Over the Cliff

In 2005, Upton, then Director of Studies in Computer Science at St. John's College, Cambridge, noticed a drop in both the number and talent level of incoming computer engineers:

> Looking at the number of people applying to study computer science in 2005/2006 gave me this sudden, horrible realization that we had driven off a cliff, and like Wile E. Coyote, we were looking down and realizing we were in real trouble. I wanted to try and get us back to that 1980s world, where every child who wanted to be a computer programmer had a machine to learn on.

The manufacturing goals were modest—a few hundred units per year—and the initial prototype was a hand-soldered board that resembled an early homebuilt computer but with modern, low-cost hardware:

> It turns out you can take one of those microcontrollers in the Arduino and generate a video signal, if you clock it hard enough and write the right software for it. We started with a piece of Veroboard, an Atmel chip, and a block of SRAM, which gives you a very 1980s computer experience. The lovely thing about it is that because we're talking about 0.1-inch through-hole components, you can build it yourself. In one rainy afternoon, you can go in with a piece of Veroboard and some chips and come out with a computer.

After they created that first prototype called the ABC Micro in homage to the 1980s BBC Micro, Upton and his friends tried to see if children would engage with it:

> When you take it and show it to kids, what you find is that it isn't exciting to them—it isn't contemporary, it isn't modern. Those computers we had in the 1980s—we didn't necessarily get them to program with, we got them to play games and do other things, and they just wheedled their way into our lives. Once they were in our lives, then we learned to program—we didn't necessarily get them with that goal in mind.

Starting Over

In 2006, Upton started working for Broadcom but he continued to work with Cambridge and developing the idea of a low-cost computer in his spare time. He started to explore how the idea of a computer

costing less than $25 might be realized by using a low-cost, low-power Broadcom chip:

> We took the Raspberry Pi concept through a series of Broadcom chips to see whether we could get something compelling. We were very lucky that toward the end of that period, one of these chips had an ARM microprocessor in it, so we made that leap from the special-purpose world, where we had to do all the software development work, to a very general-purpose one, where we made an ARM-based Linux system.

When the last iteration of the Broadcom 2835 included an ARM 11 processor, the stage was set for the Raspberry Pi that we know and love today:

> The BCM 2835 already had most of the other stuff we needed, such as support for HDMI, support for a standard display, a 3D accelerator, video accelerator, camera processor, digital signal processors, and a USB controller. What we found looking at this chip was that we were only a general-purpose processor away from this being a single-chip computer. We were lucky enough to get an ARM 11 core into the final production step of the BCM 2835. The BCM 2835 "grew" an ARM 11, and that's what lets us run Linux.

In 2009, Upton and his friends pooled some funds and formed the non-profit Raspberry Pi Foundation and by 2011, they had designed a board that turned a BCM 2835 into a miniature single-board computer, but the team was still planning on on small-scale manufacturing:

> The thing that really surprised us was the level of interest. It became apparent that we couldn't handle the manufacturing ourselves in the numbers required to meet the demand, which is why, by early 2012,

our entire business changed. Today, we're a very capital-light licensing company. We design the Raspberry Pi, we work on the brand, we work on the software and the hardware prototypes, but all of the manufacturing, the working capital provision, and the logistics are provided for us by partners, such as multibillion-dollar electronic component distributors RS Components and Premier Farnell. That's what has let us go to scale.

We started with a piece of Veroboard, an Atmel chip, and a block of SRAM, which gives you a very 1980s computer experience.

Scaling Up

The Raspberry Pi's popularity soared—the initial manufacturing run of 10,000 units sold out in hours. The brand became instantly recognizable among technologists and people involved in maker movements. The Foundation decided that it should use the opportunity to promote and advance the electronics manufacturing industry in the UK:

> We started like everyone else: when you want to build something cheap, you build it in China. But over the course of the last year, we've steadily re-shored manufacturing to the point where 100 percent of them are built in South Wales. Having these big UK PLCs [public limited companies] as partners has given us the level of support that we needed to make that happen.

With a popular brand and solid marketing and distribution channels, the Raspberry Pi Foundation is turning its attention to the problem that they initially set out to solve—getting young people interested in technology and programming. The

road isn't as direct as it might initially seem, and like previous events in Raspberry Pi's history, it was the people who bought the units and started playing with them who paved the way:

> A lot of what people have done with this has been hardware hacking. We have 32 GPIO pins, which for me was almost a design afterthought, but in practice, a lot of people have been using this as a machine to interact with other hardware. They've put it in boats, planes, and balloons—sent the machine up to the edge of space and used it to automate their homes.

Some of the earliest playful uses of the Raspberry Pi were quite sophisticated and required the skills of an experienced technologist:

> I saw a wonderful example where someone took an old microwave and completely overhauled the user experience with a Raspberry Pi. It had a touch panel on it, it had voice commands, it had a Web interface so you could control it from an iPad, it had a bar code scanner—the user built a database that mapped bar codes to cooking instructions so you could just scan your food, put it in, and the microwave would cook it perfectly. And he had built this thing with a Raspberry Pi.

Into the Classroom

But while the typical 10-year-old child might not be able to program a microwave with a Raspberry Pi, such example applications show just how far we can expand our minds and write programs that change how technology works:

> The Pi has given people access to a level of technology and a platform they can use to do some amazing stuff. The lovely thing is that a lot of what adults are doing with it feeds immediately into the range of projects available for teachers.

If we can show young people a compelling user experience from this tiny computer and demonstrate to them the potential it has for creating new and different technologies that they can control, then we have a reason to get them engaged in programming:

> We take it into schools, and at the end of the session in mixed-ability classes, there's always a group of students with whom you have to pry the Raspberry Pi out of their hands, especially if it's their first experience in programming. Even if they've done something very simple, like change a snake game so that the snake is a different color or make the snake go a little bit faster, that kind of power is intoxicating—you can see it in their eyes when they realize they can make this machine do what they want. For a sizable fraction of those kids, as it was for me, feeling that power is a transformative experience.

The story of Raspberry Pi's development is one of curiosity, serendipity, and unintended consequences, and—judging by the level of adoption and its continued and evolving use—users clearly respond. Just imagine how the availability of this machine will affect the next generation of computer engineers when they start going to college in seven or so years. ∎

Charles Severance, Computing Conversations column editor and Computer*'s multimedia editor, is a clinical associate professor and teaches in the School of Information at the University of Michigan. Follow him on Twitter @drchuck or contact him at csev@umich.edu.*

Selected CS articles and columns are available for free at http://ComputingNow.computer.org.