Small Boards

A choice of ultrasmall boards lets future developers learn how to create and program computers.

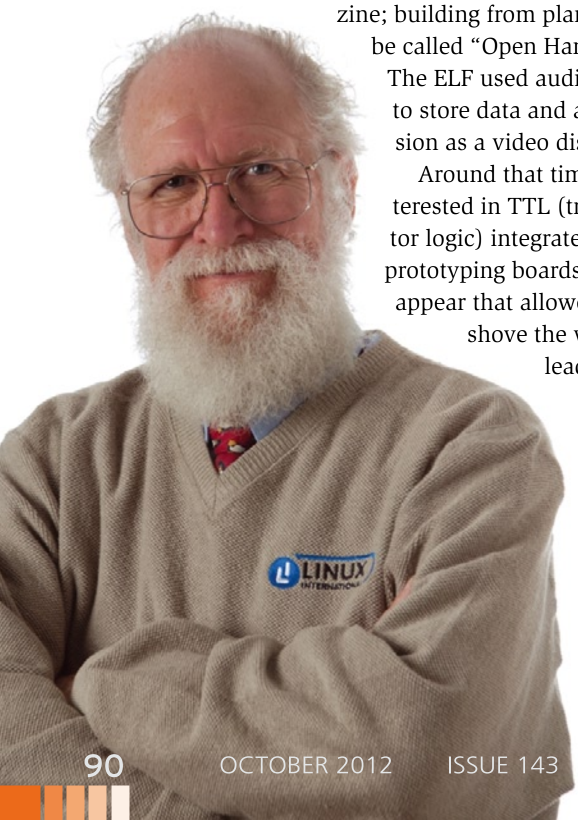
By Jon "maddog" Hall

As some of you know, I started university studying electrical engineering after having three years of electronics in high school. In electronics classes, we spent about half our time taking apart old televisions and radios to get the parts we needed to design and build new ones.

We studied the fundamentals of tubes ("valves" to the British) and learned how to build logic gates out of relays, tubes, and these relatively new things called "transistors" – integrated circuits were far in the future. Engineers also practiced soldering and wire wrapping [1] along with their design skills. I remember paying US$ 1.50 for a single transistor – and more if it was "special," such as a power transistor.

Fast forward to 1975. I worked as a software engineer in a large insurance company, and microcomputers were starting to arrive. A lot of them were in kit form and fairly expensive. I spent a lot of time soldering (and wire wrapping) components together. Often the instructions did not match the circuit diagram, which did not match the printed circuit board layout. Then, my electrical engineering experience started to pay off. I could look at the circuit diagram, see if it was correct, and figure out where the instructions had gone wrong. My experience also helped when the hardware (after much soldering) did not work. I could chase down problems armed only with a vacuum tube voltmeter [2] (alas, I once again show my age because all of these units are now transistorized and "solid state").

A few years later, I was teaching at a small two-year technical college, and I built a computer called the "COSMAC ELF" [3]. The plans were published in the old *Popular Electronics* magazine; building from plans like this would be called "Open Hardware" today. The ELF used audio tape cassettes to store data and an analog television as a video display.



Around that time, I became interested in TTL (transistor-transistor logic) integrated circuitry, and prototyping boards [4] started to appear that allowed you just to shove the wires (whether leads from a resister, capacitor, integrated circuit, or just plain "jumper" wire) into the holes in the board to make a connection. These projects were fairly expensive; I had to pay for the electronic parts and breadboarding equipment as well as the equipment to test the circuits and inject signals into the circuits. Yet, it was still fun to build electronic circuitry.

A while ago, I wrote about the Arduino prototyping hardware platform [5] that you can build yourself from plans or buy preassembled [6]. Today, systems like the BeagleBoard [7]; the newer, smaller, and less expensive BeagleBone [8]; and the Raspberry Pi [9] make building your own projects even easier.

The BeagleBoard, introduced in 2008, currently has a large community of developers around it, with development software that works on all major desktop operating systems and the ability to host several operating systems. Subsequent modifications to the basic board have given it greater capability, and, in the case of the BeagleBone, it has become smaller (to fit in the ever-present Altoids tin) and less expensive (about US$ 84 compared with the higher cost of the BeagleBoard at US$ 149). Of course, the BeagleBoard has more hardware devices to play with than BeagleBone, but the Bone is very, very cute sitting in its Altoids box! The Bone also has "capes," which are daughter boards similar to the Arduino "shields" for expansion.

The Raspberry Pi is the newest and least expensive of the "development systems," costing about US$ 34 (or US$ 25 for the smaller model without Ethernet). In some ways, it is less powerful than the Beagle systems, but in other ways it is more powerful (e.g., a more powerful GPU). For each of these systems the openness of the project allows people to experiment, build new things, and increase their knowledge of electronics and software interaction.

The low cost of the Raspberry Pi would allow educational sponsors to purchase a number of units for grade schools to stimulate interest in computer engineering. The power supplies that are used tend to be non-lethal, and students get to learn how to control real mechanical devices instead of just using computers to play games. ■■■

INFO

[1] Wire wrapping: *https://en.wikipedia.org/wiki/Wire_wrap*

[2] Vacuum tube voltmeter: *http://www.ohio.edu/people/postr/bapix/vtvms.htm*

[3] Cosmac ELF: *https://en.wikipedia.org/wiki/COSMAC_ELF*

[4] Breadboard: *https://en.wikipedia.org/wiki/Breadboard*

[5] "Hack and Beer in Rio" by Jon "maddog" Hall, *Linux Magazine*, December 2010, pg. 89: *http://www.linuxpromagazine.com/Issues/2010/121/Doghouse-Hack-and-Beer-in-Rio*

[6] Arduino: *http://www.arduino.cc/*

[7] BeagleBoard: *http://beagleboard.org/*

[8] BeagleBone: *http://beagleboard.org/bone/*

[9] Raspberry Pi: *http://www.raspberrypi.org/*