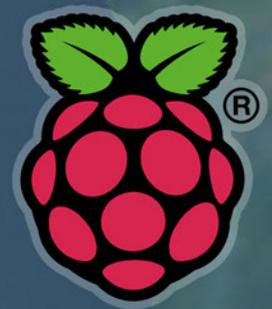


Eben Upton
Founder of the
Raspberry Pi Foundation

Pi PROJECTS

- 32-way Raspberry Pi Cluster
- Live Weather Mapping

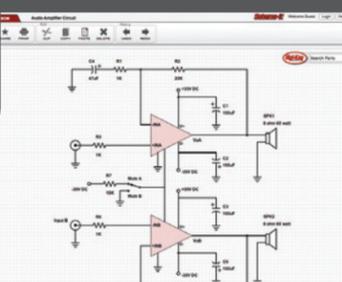
THE RASPBERRY Pi ISSUE



• THE CREDIT CARD-SIZED,
\$25 COMPUTER THAT IS
REVOLUTIONIZING THE
WAY WE EDUCATE.

Bringing your *concepts* to *reality* is as easy as...

1.



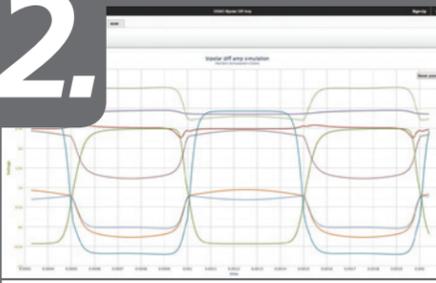
Scheme-it

Create schematics, technical diagrams, and flowcharts using your browser.

- 600+ Symbol Library
- Share Schematics Online
- Export High Quality Images

digikey.com/schemeit

2.



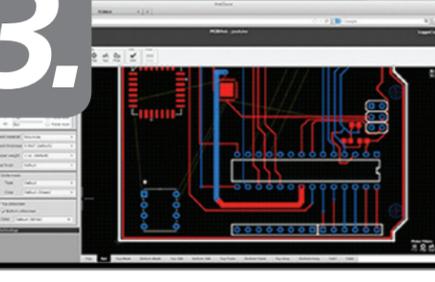
PartSim

Free and easy-to-use circuit simulator that runs in your browser.

- SPICE Simulator
- AC/DC/Transient Sims
- Waveform Viewer

partsim.com

3.



PCBWeb

Full featured online CAD application for designing and manufacturing electronics hardware.

- Schematic Capture
- PCB Layout
- BOM Integration

pcbweb.com



Visit: digikey.com/schemeit • partsim.com • pcbweb.com



Eben Upton
THE RASPBERRY PI FOUNDATION — **4**
 A conversation about how this affordable, credit card-sized computer is changing the way we educate.

Featured Products
 This week's latest products from EEWab. — **10**

RPiCluster
 PhD candidate Josh Kiepert details the process of building a 32-node Raspberry Pi Beowulf cluster. — **14**

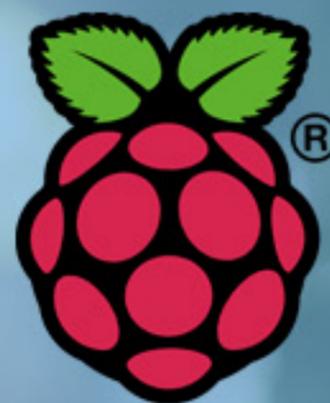
Baking Pi & the Unix OS
 by Alex Toombs — **26**
 Cambridge University's new course uses the Raspberry Pi as a platform to show how operating systems work.

Raspberry Pi Weather Station — **32**
 This project utilizes the Raspberry Pi's simplicity and computing power to continuously monitor temperatures and weather patterns.

The Path to Pi
 by Rob Riemen — **38**
 A history of innovations in open source software and hardware that made the Raspberry Pi possible.

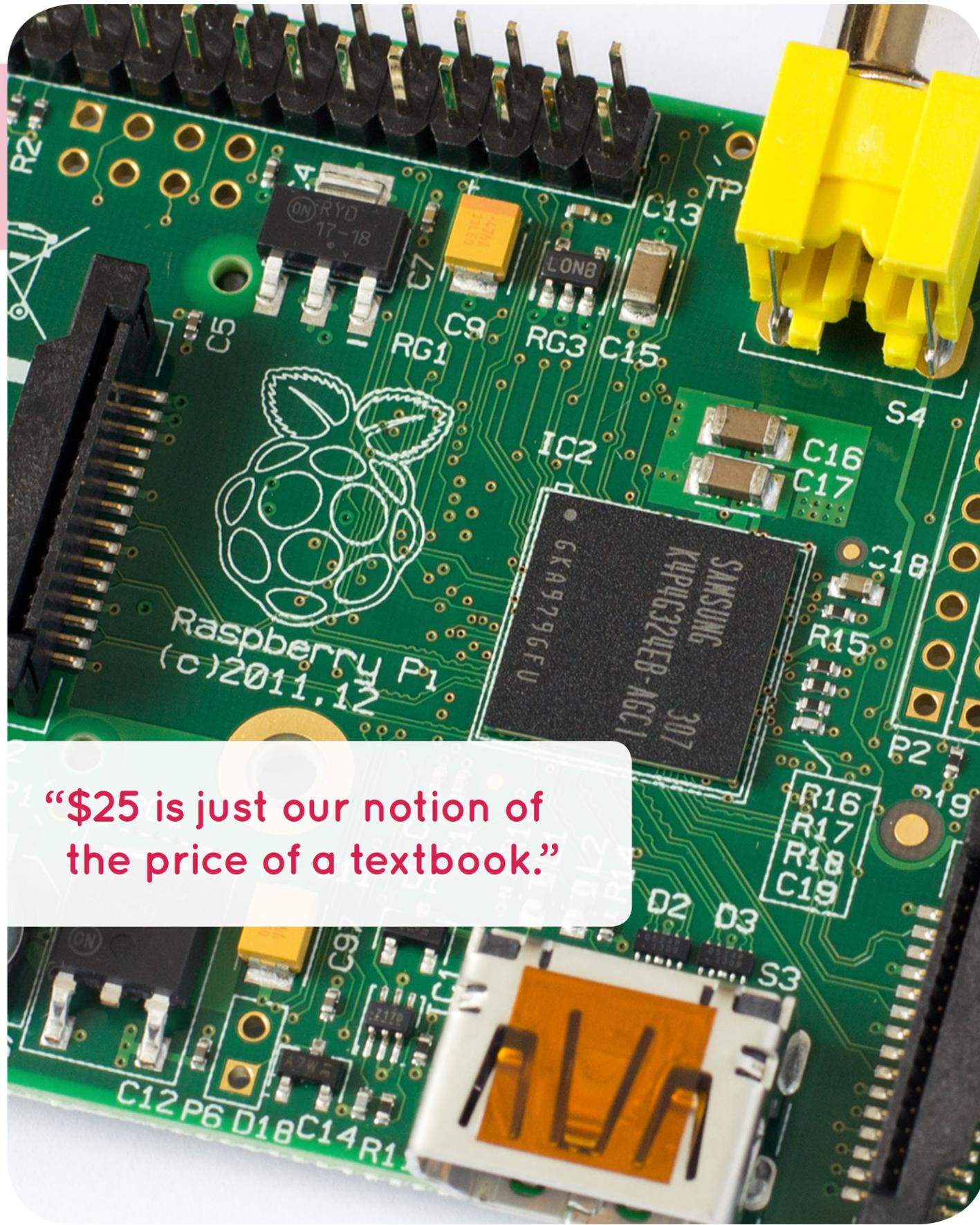
RTZ
Return to Zero Comic — **43**

EBEN UPTON



The Raspberry Pi Foundation is the brainchild of Eben Upton and his colleagues at the University of Cambridge in the UK. The foundation is a non-profit that aimed to produce a remarkably inexpensive, credit card-sized computer--the Raspberry Pi. This computer's simplicity allows it to be a great teaching tool for children and hobbyists alike and is powerful enough to be featured in industrial applications and personal projects. Since its initial release, the Raspberry Pi has taken the computing industry by storm, selling millions of units in its first year. Countless blogs, magazines, and videos have been created to showcase Raspberry Pi projects, potential applications, and computer science lesson plans.

We spoke with Eben Upton about the inspiration behind the Raspberry Pi, some of his favorite projects using the Pi, and the computer's revolutionary impact on computer science education.



“\$25 is just our notion of the price of a textbook.”

Could you talk about the inspiration for creating the Raspberry Pi?

My colleagues from Cambridge and I had this idea that we should try to make a computer that we can give to children so that they can learn to program in the same way that we learned how in the 1980s.

We wanted to produce different models of computers, which are reference to the BBC Micro, one of the successful computers here in the UK. The BBC Micro had a configured Model A, which was an entry-level device and a Model B, which was a full-featured device. We kind of adopted that idea. We initially had the idea of making a \$25 computer, but we realized that we could do a much more fully featured device for \$35. What we developed was a device with a minimal version of it for \$25 and a “deluxe” version of it for \$35, which is where we got Model A and Model B from. We shipped the Model B exclusively for the first year and the Model A kind of rolled in in February of this year.

What is the Raspberry Pi Foundation?

We are UK registered charity. We are kind of an unusual charity in that we don't really accept donations. However, we have received some donations from high-level companies like Google, but those are kind of a rarity. By and large, we trade. Our main source of money is the profits we get from selling Raspberry Pis. The Foundation has a board of trustees and it's ultimately responsible for—on the trading side—the branding and the marketing as well as hardware designs that refine the core device itself. They are also in charge of things like the camera board, which is a peripheral. The Foundation has a bit that makes the money and a bit that spends the money, which is the

education side of it. On the education side, we do some outreach, we sponsor the creation of course material, and we do a little bit of lobbying trying to convince the government that there's a problem with education. It's a very interesting mixture. It actually feels very much like a commercial company, but all of the returns are put back into this charity.

Did the foundation come about from the initial success of the board, or did you have the idea to start the foundation from the very beginning?

The Foundation has been in existence since 2008. One way you could tell which order they came in is the name of the Foundation and product—the Raspberry Pi—is kind of an obsolete name. The “Pi” part of Raspberry Pi is the “Py” in Python. 2008 is the point where we incorporated the foundation and we thought we could build a device that would just run on Python, or a Python equivalent. The foundation didn't just predate the board, it predates the entire concept of this being a Linux box.

What are some of the main differences between the Model A and the Model B?

The Model A only has one USB, no Ethernet, and 256 MBs of RAM. The Model B has two USB ports, Ethernet, and 512 MBs of RAM. This is what I mean when I say you can do an amazingly more powerful device for \$35 than you can for \$25. The \$25 device is way more entry-level that is geared more towards children, but if you are looking for the ability to do some programming and to play videos and games on, then it's the Model B you are looking for, which is a popular choice. We have to chase down to that low price point, but we just have to acknowledge that enough is enough at some point.

“All you need is a balloon and a bit of helium and a Pi. The skies would be a little crowded, but every primary school in the world can have a space program and take pictures of the Earth from near space.”

Was there a lot of thought that went in to try and find that price point?

We actually did things in a funny order. First, we developed the price point, and then we incorporated the foundation, and then we developed the hardware. It's a pretty backwards order when you think about it. \$25 is just our notion of the price of a textbook. What's the thing that you could ask even the poorest child to buy as a school textbook? There are children that are so poor that they can buy a textbook, but there are few enough of them than the amount that schools can subsidize those people. It's essentially an access-driven thing. We looked at what we could build for \$25 and that kind of pegged our entry-level device and then we did just a little bit of optimization because people said to us that the first thing they were going to do with the Raspberry Pi is plug it into a hub to get some more USB and plug it into a network adapter. We tried to figure out if we could add a USB hub and network adapter within a reasonable additional cost. We found that we could.

Could you talk about some of the operating systems that are supported?

The Pi can run on Linux as well as a number of variants of BSD. There have been some efforts to develop there. We also run RISC OS, which was the original ARM operating system from Acorn in the UK back in the 1980s.

We've gotten a lot of questions like, "To what extent does a Pi make a good industrial computer?" It'll be interesting to see. We have seen people putting them into industrial applications,

but I think the industrial applications have to come first, and then the RTOS support will follow, rather than expecting the RTOS vendors to put a lot of investment into this relatively unproven platform.

What have been some of the most unique projects you've seen using the Raspberry Pi board?

There have been a number of zoology-type applications. ZSL, the Zoological Society of London, has just won a half a million-dollar grant from Google to deploy wildlife tracking cameras in Africa that are based on the Pi and the Pi's camera infrastructure. That's a particularly nice one because these are designed to go out and take pictures at night and there have been some talks about getting them to triangulate gunshots. If you hear a gunshot from a poacher, you can dispatch to send warnings.

The one that is still very dear to my heart was done by someone here the UK. This guy puts Raspberry Pis into weather balloons and sends them up to 40Km and takes pictures of the UK from the edge of space. I like this one from an educational point of view because every primary school in the country can afford to do that. All you need is a balloon and a bit of helium and a Pi. The skies would be a little crowded, but every primary school in the world can have a space program and take pictures of the Earth from near space.

People are also using these as a Lego piece to build their own things on top of. In our core education market, you envision people using these as computers, but I think a lot of the industrial and embedded types of things are about using them as a building block. We want to be able to support that.

Do you see a lot of people using the Pi in commercial applications?

We see a number of people using them basically as industrial computers. There's a lot of industrial applications where those are embedded inside a product or just a computer that that can talk to a display, sensors, or actuators in a machine. There are lots of little industrial computers that serve those niches, but the thing is, the Pi is better than almost all of those. I think this is a nascent thing that will take off in the near future.

Do you have an idea of how many Raspberry Pis have been sold?

We have sold around 1.3 million units over 15 months, which is a pretty substantial number. It's enormously gratifying to those of us who thought we would sell 1,000 or 10,000 total. To be selling that kind of number every day is just nuts.

How many people work at the Foundation?

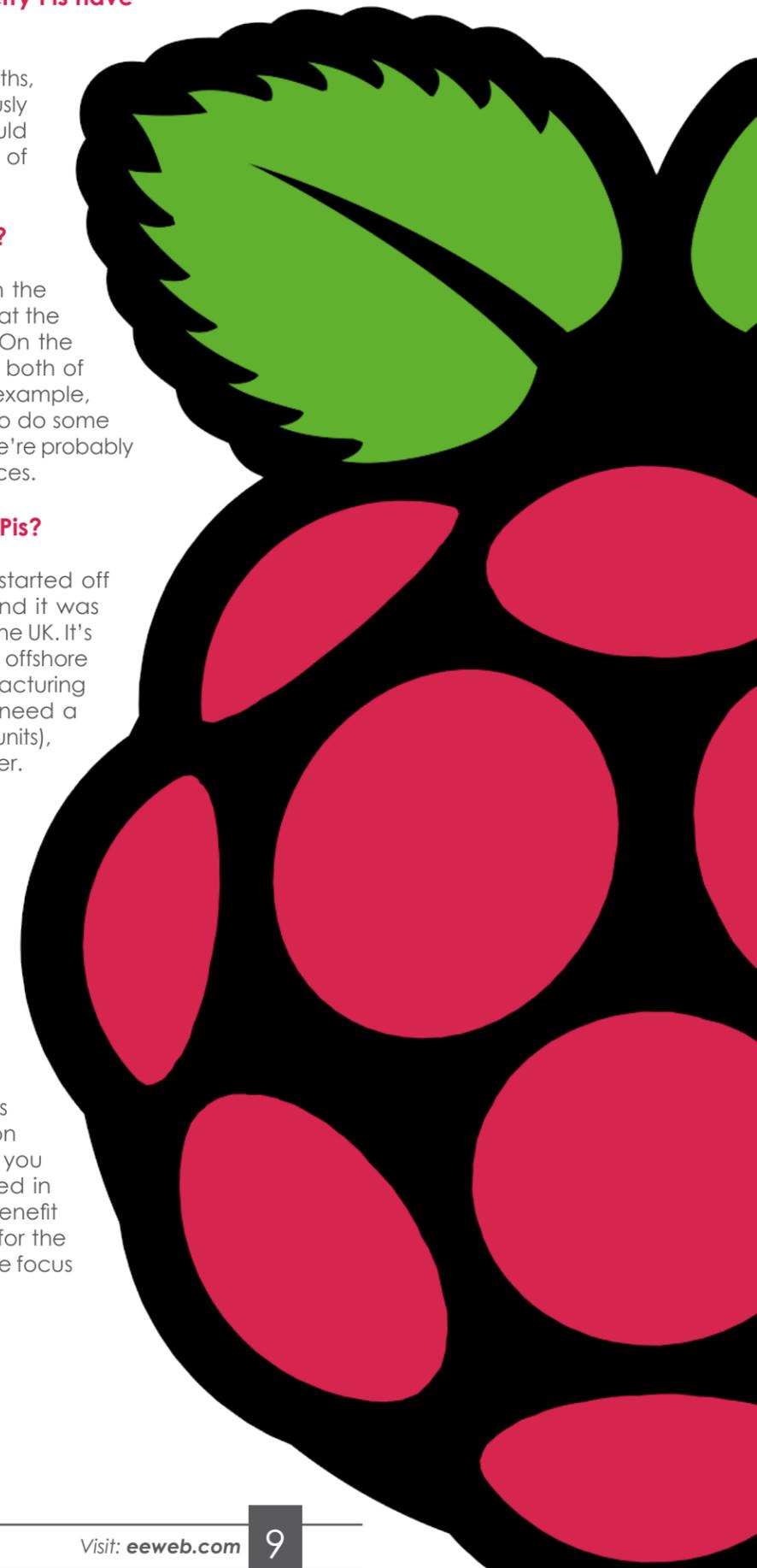
Roughly six. It depends on how you count. On the Foundation side, we have a couple of people at the moment, but we are looking to staff that up. On the business side, we have four or five people. On both of those sides there are a lot of contractors. For example, we have a postdoctoral student that we pay to do some learning platform stuff for us. At any given time, we're probably buying five peoples worth of contractor resources.

Where do you manufacture the Raspberry Pis?

We manufacture them in the Wales, UK. We started off manufacturing entirely in China, but we found it was actually cheaper to manufacture them here in the UK. It's interesting, there's sort of this misperception that offshore manufacturing is cheap. I think if you're manufacturing in staggering volume, then it is cheap. If you need a modest volume (and by modest, I mean 1 million units), I think that manufacturing in the West is cheaper.

How do you see the Raspberry Pi evolving over the next few years?

I think we are going to see a lot of work on software. People keep asking us if we are going to do another one, but I think the models we have already are pretty good. I think there's a lot of work you can do in software and if you look at our website, there are some really nice examples of recent stuff we've done. There have been really significant performance improvements in the applications that people care about, simply through attention to detail. The nice thing about doing that is that you don't often follow the people who've invested in your platform. I want people to continue to benefit off of the software that we're doing. At least, for the foreseeable future, we're going to see a lot more focus on software than hardware. ■



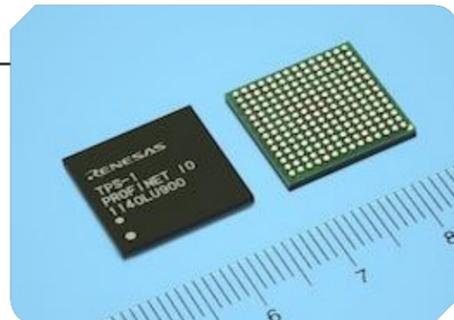


Lowest Power Logic Product Family

NXP Semiconductors today introduced the AXP logic family designed for high-performance, low-voltage and low-power applications. The new AXP family is fully specified at the 0.8-volt supply range, making it the industry's fastest family of devices with the lowest power consumption and highest output drive. Currently available in the smallest MicroPak (leadless) packages, designers now have an excellent balance of power, speed and packaging in standard logic...[Read More](#)

PROFINET I/O Device Chip Interface

Renesas Electronics Corporation announced the availability of the TPS-1 PROFINET IO (input/output) device chip and the related TPS-1 starter kit based on the signed licensing agreement with KW-Software, a company of the Phoenix Contact Group and a leading provider of industrial solutions and technologies. The TPS-1 complements the existing PROFINET technology components ERTEC 200 and ERTEC 400 already available from Renesas Electronics...[Read More](#)

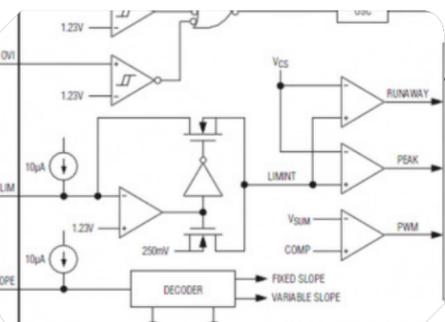


Optical Finger Navigation Lens

Avago's ADBL-A321 Optical Finger Navigation lens is designed for use with our ADBS-A320 and ADBS-A350 Optical finger Navigation Sensor. The ADBL-A321 Optical Finger Navigation lens is a precision-molded optical component which provides the direct illumination and optical imaging necessary for proper sensor operation. This optical finger navigation lens is a polycarbonate and should be handled with care to avoid scratching of the optical surface. Cyanoacrylate-based adhesives should also not be used, as they could cause deformation of the lens material...[Read More](#)

Speaker Amplifier with Audio Processing

The TAS2505 is a low power digital input speaker amp with support for 24-bit digital I2S data mono playback. In addition to driving a speaker amp upto 4-Ω, the device also features a mono headphone driver and a programmable digital-signal processing block. The digital audio data format is programmable to work with popular audio standard protocols (I2S, left/right-justified) in master, slave, DSP and TDM modes. The programmable digital-signal processing block can support Bass boost, treble, or EQ functions...[Read More](#)



AC-DC and DC-DC Converters

The MAX17498A/MAX17498B/MAX17498C devices are current-mode fixed-frequency flyback/boost converters with a minimum number of external components. They contain all the control circuitry required to design wide input voltage isolated and nonisolated power supplies. The MAX17498A has its rising/falling undervoltage lock-out (UVLO) thresholds optimized for universal offline (85V AC to 265V AC) applications, while the MAX17498B/MAX17498C support UVLO thresholds suitable to low-voltage DC-DC applications...[Read More](#)



Laser Diode Driver with Rise Time Control

The PCO-6131 features a user-adjustable variable rise time control. This innovative feature allows the user to adjust the rise time within a range of 2.5μs by means of a PCB-mounted potentiometer, to optimize the driver's rise time for the user's application. The PCO-6131 is based on a hysteretic, average current, switch-mode regulator. This type of regulator is a variable frequency, variable pulse width design which maintains current in an energy storage inductor between a minimum and maximum level. This architecture provides a high performance driver in a small form factor, with high operating efficiency and low stored energy. At 125A output current, the stored energy in the driver is approximately 7 Joules, dramatically lower than the stored energy in comparable linear current sources...[Read More](#)

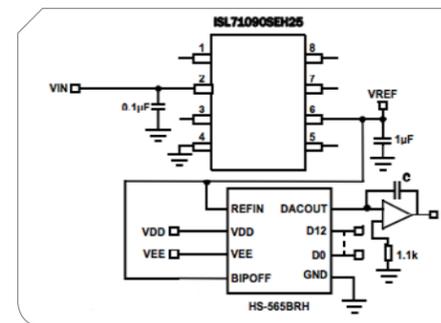
Low-Power High-Speed CMOS Converters

Fujitsu announced the first in a new family of 8-bit, power-efficient, 28nm CMOS converters. The analog-to-digital converter (ADC) addresses the need for large-scale global deployment of single-wavelength, 100Gbps optical transport systems, and provides a solution for future short-range optical and backplane interconnects. Now in its third generation of process technology, the 28nm ADC supports sampling rates from 55 to 70 GSa/s (billions of samples per second) with scalable analog bandwidth...[Read More](#)



Ultra Low-Noise Voltage Reference

The ISL71090SEH25 is an ultra low noise, high DC accuracy precision voltage reference with a wide input voltage range from 4V to 30V, ideal for high-end instrumentation, data acquisition and applications requiring high DC precision where low noise performance is critical. The device uses the Intersil Advanced Bipolar technology to achieve sub 2μVP-P 0.1Hz noise with an accuracy over temperature and radiation of 0.15%...[Read More](#)



Automatic X Capacitor Discharge IC

CAPZero is an IC that completely eliminates the power dissipated by X capacitor discharge resistors. CAPZero is a two terminal device that can be added to any power supply with X capacitor discharge resistors to decrease no-load consumption and provides more standby power. It is a simple product that easily meet safety requirements. When AC voltage is applied, CAPZero blocks current flow in the X capacitor safety discharge resistors, reducing the power loss to less than 5 mW or essentially zero at 230 VAC...[Read More](#)





Low Power CAN Transceivers

Microchip Technology Inc. announced a new family of High-Speed (HS) CAN transceivers, the MCP2561/2. These new devices serve as an interface between a CAN protocol controller and the physical two-wire CAN bus. The new family of MCP2561/2 transceivers provides two device options. The MCP2561 is an HS CAN transceiver in an 8-pin package and features a SPLIT pin. The SPLIT pin helps to stabilize common mode in biased split termination schemes. The MCP2562 is also an HS CAN transceiver available in an 8-pin package and features a Vio pin...[Read More](#)

BeagleBone Black

Mouser Electronics, Inc. has stock and immediate availability of the highly sought-after BeagleBone Black from BeagleBoard.org, a lower-cost, high-expansion focused BeagleBone featuring a Sitara AM335x Cortex A8 ARM processor from Texas Instruments. BeagleBone Black is the exciting new member of the BeagleBoard family, taking popular features from the original BeagleBone, and adding excellent additional features and performance with a faster processor, onboard HDMI. The BeagleBoard platform has a proven track record of facilitating innovation from concept to market and BeagleBone Black's enhanced feature set enables developers to turn ideas into prototypes more easily than ever...[Read More](#)

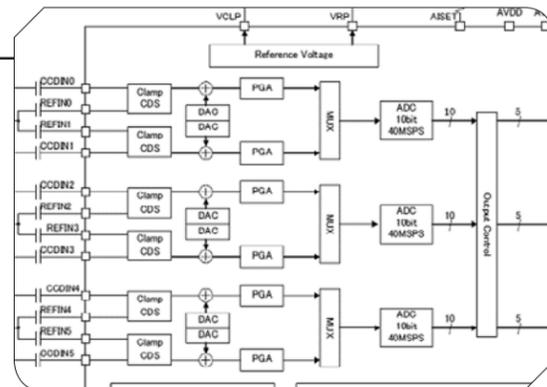


Subminiature Radial Leaded Fuse

The 262/268/269 Series are high-reliability micro fuses, with a 125V rating, very fast-acting type with high breaking capacity. This series is listed under the Department of Defense Quality Product List. The fuses are available from very low amperes of 2mA to 5A and can be plug-in or radial leaded. They can also be usable in AC or DC circuits capable of withstanding and functioning in extreme conditions found in spacecraft or military applications...[Read More](#)

Compatible Analog Pre-Processor

Introducing AKM's 6-Channel Linear Sensor compatible 10 Bit 40 MSPS x 3 Analog Pre-Processor, The AK8448. It is a +3.3 V CMOS, Two Channel input 10 Bit 40MSPS ADC which integrates on-chip Offset Adjust DAC, Gain Adjust PGA and CDS circuit. The AK8448 is analog front end IC for linear image sensor, which integrates six channels of CCD/CMOS sensor interface, three 10-bit 40MSPS analog-to digital converters, offset adjust DACs and programmable gain amplifiers. AK8448 is suitable for scanner applications. The device can also simultaneously sample 6-channel CDS circuits...[Read More](#)



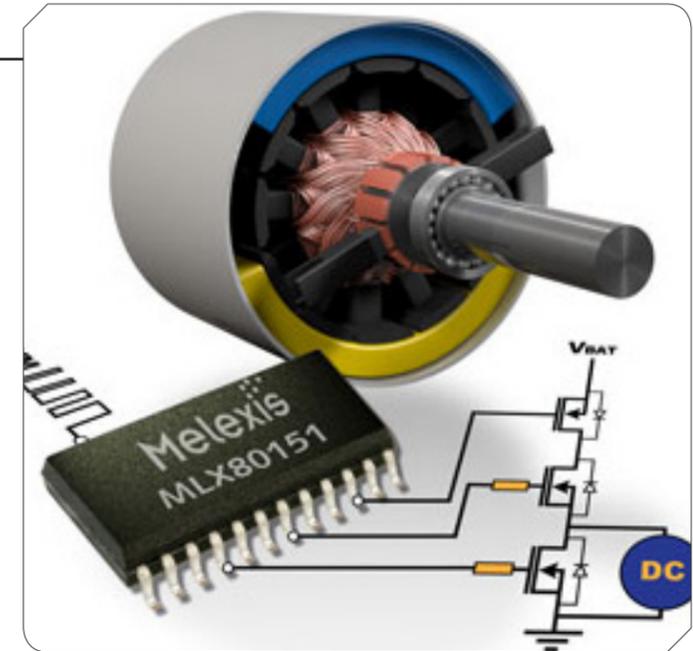
High Current NFET MCU

The MLX80151 is an automotive half bridge motor controller for PWM controlled high current DC pumps and fans.

Some of features and benefits include:

- High temperature motor controller
- AEC-Q100 qualified up to Tamb = 150C
- SIL 3 capable
- High integration level
- Embedded 16bit microcontroller with three 16bit timers
- High Side and Low Side shunt amplifier

The device also has extensive hardware and software diagnostics capabilities...[Read More](#)



Independently Controlled Multi-Channel LED Driver with I²C Interface

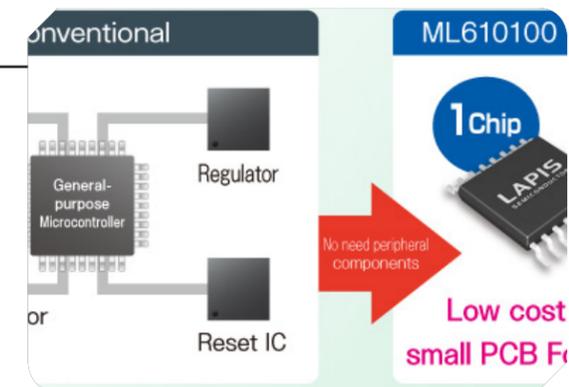


Multi-Channel LED Driver

The A8517 is a programmable multi-output LED driver, which can be used for a variety of LED Driver applications. It integrates a current-mode boost converter with internal power switch and 10 current sinks. The IC operates from 4.5 to 36 V, and is able to withstand up to 40 V load-dump conditions encountered in automotive systems. The I²C interface allows the user to set the LED currents individually, up to 60 mA per LED channel. Adjacent channels may be combined to drive higher-current LED strings...[Read More](#)

LP Micros for Embedded System Control

LAPIS Semiconductor, a ROHM Group Company, is pleased to introduce the ML610Q1xx series of low-cost, feature-rich, low-power microcontrollers. The new Mini Low Power Microcontrollers are especially well suited for embedded applications requiring a very small microcontroller that provides high computational performance and dedicated control functions, yet consumes a minimal amount of power. Applications include motor control, energy harvesting sensors and nodes, audio and speech control, smoke and fire detection, security and access control, battery management, home appliances, lighting control, power monitoring and metering, and many other embedded designs...[Read More](#)





Joshua Kiepert

PhD Candidate at Boise State University

Introduction

The RPiCluster project was started a couple months ago in response to a need during my PhD dissertation research. My research is currently focused on developing a novel data sharing system for wireless sensor networks to facilitate in-network collaborative processing of sensor data. In the process of developing this system it became clear that perhaps the most expedient way to test many of the ideas was to create a distributed simulation rather than developing directly on the final target embedded hardware.



Thus, I began developing a distributed simulation in which each simulation node would behave like a wireless sensor node (along with inherent communications limitations), and as such, interact with all other simulation nodes within a LAN. This approach provided true asynchronous behavior and actual network communication between nodes which enabled better emulation of real wireless sensor network behavior.

For those who may be unfamiliar with a Beowulf cluster before, a Beowulf cluster is simply a collection of similar, (typically) commodity computer hardware based systems, networked together and running some kind of parallel processing software that allows each node in the cluster to share data and computation. Typically, the parallel programming software is MPI (Message Passing Interface), which utilizes TCP/IP along with some libraries to allow programmers to create parallel programs that can split a task into parts suitable to run on multiple machines simultaneously. MPI provides an API that enables both asynchronous and synchronous process interaction. My simulation did not require the computational power of a Beowulf cluster; it simply required several distinct computers on a common network. A cluster environment provided a very convenient development platform for my distributed simulation thanks to its common file system as well as its uniform hardware.

So, why I would want to build a Beowulf cluster using Raspberry Pis? The Raspberry Pi has a relatively slow CPU by modern standards. It has limited RAM, slow USB-based 10/100 Ethernet, and its operating system runs directly on a SD card. None of these "features" are ideal for a cluster computer! Well, there are several reasons. First, when your dissertation work requires the use of a cluster it is nice to ensure that there is one available all the time. Second, RPi's provide a unique feature in that they have external low-level hardware interfaces for embedded systems use, such as I2C, SPI, UART, and GPIO. This is very useful to electrical engineers (like myself) requiring testing of embedded hardware on a large scale. Third, having user-only access to a cluster (which is the case for most student-accessible systems) is fine if the cluster has all the necessary tools installed. If not however, you must then work with the cluster administrator to get things working. Thus, by building my own cluster I could directly outfit it with anything I might need. Finally, RPi's are cheap! The RPi platform has to be one of the cheapest ways to create a cluster of 32 nodes. The cost for an RPi with an 8GB SD card is ~\$45. For comparison,

each node in one of the clusters available to students here at BSU, was about \$1,250. So, for not much more than the price of one PC-based node, I could create a 32 node Raspberry Pi cluster!

One thing to keep in mind when building a cluster with RPi's is to set expectations of performance appropriately. That is, we cannot expect the compute performance to be very good, even when compared to a single modern multi-core desktop computer. This was not a problem for my application because I just needed many independent processors running on a common network to run my distributed application. I didn't need a platform for high performance computing (HPC). Additionally, since the RPi uses an ARM processor, it has a different architecture than PCs, i.e. ARM vs x86. Thus, any MPI program created originally on x86 must be recompiled when deployed to the RPiCluster. Fortunately, this issue is not present for Java, Python, or Perl programs. Finally, because of the limited processing capability, the RPiCluster will not support multiple users simultaneously using the system very well. As such, it would be necessary to create some kind of time-sharing system for access if it ever needed to be used in such a capacity.

The RPiCluster is great if your development is focused only on distributed computing/network programming rather than parallel processing. That is, if the programs being developed for the cluster are distributed in nature, but not terribly CPU intensive. Compute-intensive applications will need to look elsewhere, as there simply is not enough "horse power" available to make the RPi a terribly useful choice for cluster computing. Of course, even compute-intensive applications could be tested on a small scale with the RPiCluster.

Building the System

There are really only five major components needed for a working cluster: computer hardware, Linux OS, an MPI library, an Ethernet switch, and possibly a router. Figure 1 shows the overall network architecture. The RPiCluster design consists of 32 RPi nodes, a 48-port 10/100 switch, Arch Linux ARM, and MPICH3. The total cost of the system was \$1,967.21.

There are several operating systems available as pre-configured Linux images for the RPi: Raspbian "wheezy" (based on Debian) and Arch Linux for ARM (aka "alarm"). I chose to use Arch Linux for the RPiCluster. Arch

Linux was chosen primarily for its minimalist approach which allowed me to install only what I needed. It also resulted in a system that boots in about 10 seconds. There is definitely a learning curve with Arch Linux, but fortunately it has some of the best organized information out there for learning Linux. An excellent place to start is the Arch Linux Beginner's Guide.

As for an MPI implementation, I chose MPICH primarily due to my previous familiarity with it (OpenMPI is also available). On Arch Linux MPICH must be compiled from source, or installed from the Arch User Repository (AUR). I went with the compile-from-source method. You can get the source from mpich.org. OpenMPI is available directly in the Arch Linux extra repository.

One thing I found helpful when creating an Arch Linux image for the RPiCluster was QEMU. QEMU is a CPU architecture emulator and virtualization tool. It is capable of emulating an ARM architecture similar to the RPi. This allows us to boot an RPi image directly on our x86 system. There is one caveat: you need an RPi kernel for QEMU that knows how to deal with the QEMU virtualized hardware, which is not normally part of the RPi kernel image. This involves downloading the Linux kernel source, applying a patch to support the RPi, adding the various kernel features and drivers needed, and compiling.

There are several tutorials available which provide details for running QEMU with an RPi system image. A couple references I found useful are [Raspberry Pi under QEMU](#) and [Raspberry Pi with Archlinux under QEMU](#). Aside from the need for a custom kernel, there are a couple system configuration changes needed within the RPi image to allow it to boot flawlessly. The changes primarily have to do with the fact that RPi images assume the root file system is on /dev/mmcblk0p2 and the boot partition is on /dev/mmcblk0p1. QEMU makes no such assumptions so you have to map /dev/sda devices to mmcblk0 devices on boot. With the RPi system image adjusted and the custom kernel built, starting QEMU is something like the following:

```
$ qemu-system-arm -kernel ./zImage
-cpu arm1176 -m 256 -M versatilepb
-no-reboot -serial stdio -append
"root=/dev/sda2 panic=0 rw" -hda
archlinux-hf-2013-02-11.img
```

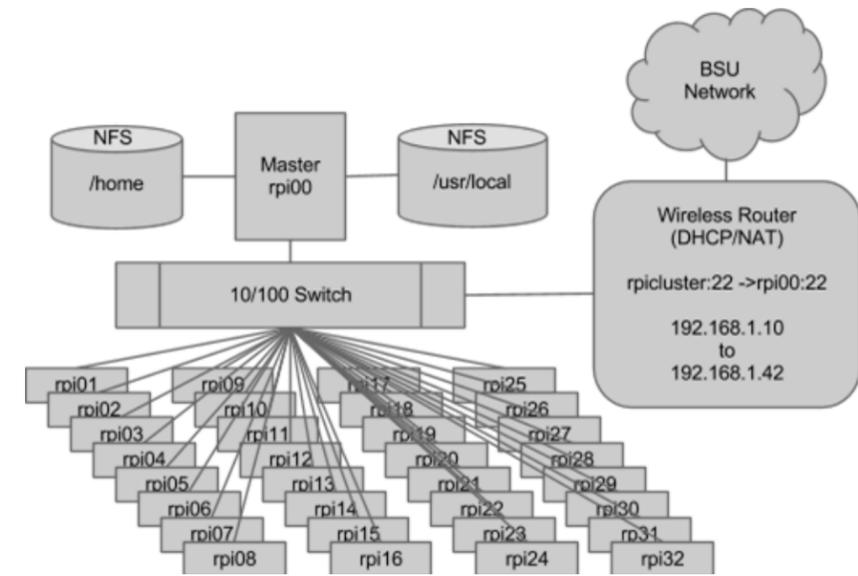


Figure 1: Cluster Design: RPiCluster Network Architecture



Figure 2: Arch Linux ARM under QEMU

Once you have a kernel image (zImage) that is suitable for QEMU you can point it at the new kernel and the RPi system image. Figure 2 shows the boot screen of Arch Linux ARM under QEMU.

With a working Arch Linux image under QEMU, I applied system updates and installed the necessary packages for the cluster environment. Once configuration of the image was completed with QEMU, I was able to simply write the resulting image to each RPi SD card after modifying host name and static IP address, unique to each (I wrote a bash script to do this).

Below are a few of the steps involved for configuring an Arch Linux image for use in a cluster:

1. Change root password from default (root):
passwd
2. Full system update:
pacman -Syyu
3. Install typically needed packages, (bold packages are need for using NFS, OpenMPI, and software development):
pacman -Syy nfs-utils base-devel openmpi sudo adduser gdb
4. Set the timezone for your location (<city>):
timedatectl set-timezone America/<city>
5. Add a new user (<user>) and be sure to add groups wheel, audio, video, and uucp when prompted:
adduser <user>
6. Allow users that are part of the wheel group to use superuser permissions:
chmod 600 /etc/sudoers
7. Uncomment the line: %wheel = ... and save:
nano /etc/sudoers
8. Logout of root and login as the user who will be using MPI and generate ssh keys for password-less login:
\$ ssh-keygen -t rsa
\$ ssh-keygen -t dsa
\$ ssh-keygen -t edsa

Strictly speaking, there is nothing special needed for a working cluster other than MPI and SSH, however a shared file system makes life much easier (e.g. NFS). Cluster setup is basically just configuring a few files so that all nodes can find each other by name on the network and have password-less SSH access to one another. Key configuration files for cluster setup under Arch Linux:

- NFS: /etc/exports
- NFS: /etc/idmapd.conf
- NFS: /etc/fstab
- SSH: /home/<mpiuser>/.ssh/authorized_keys
- Network: /etc/hosts
- Network: /etc/hostname
- Network: /etc/network.d/ethernet-static
- Network: /etc/conf.d/netcfg

For the fine details on NFS, network, and SSH setup, see the Arch Wiki: NFS, Network, SSH.

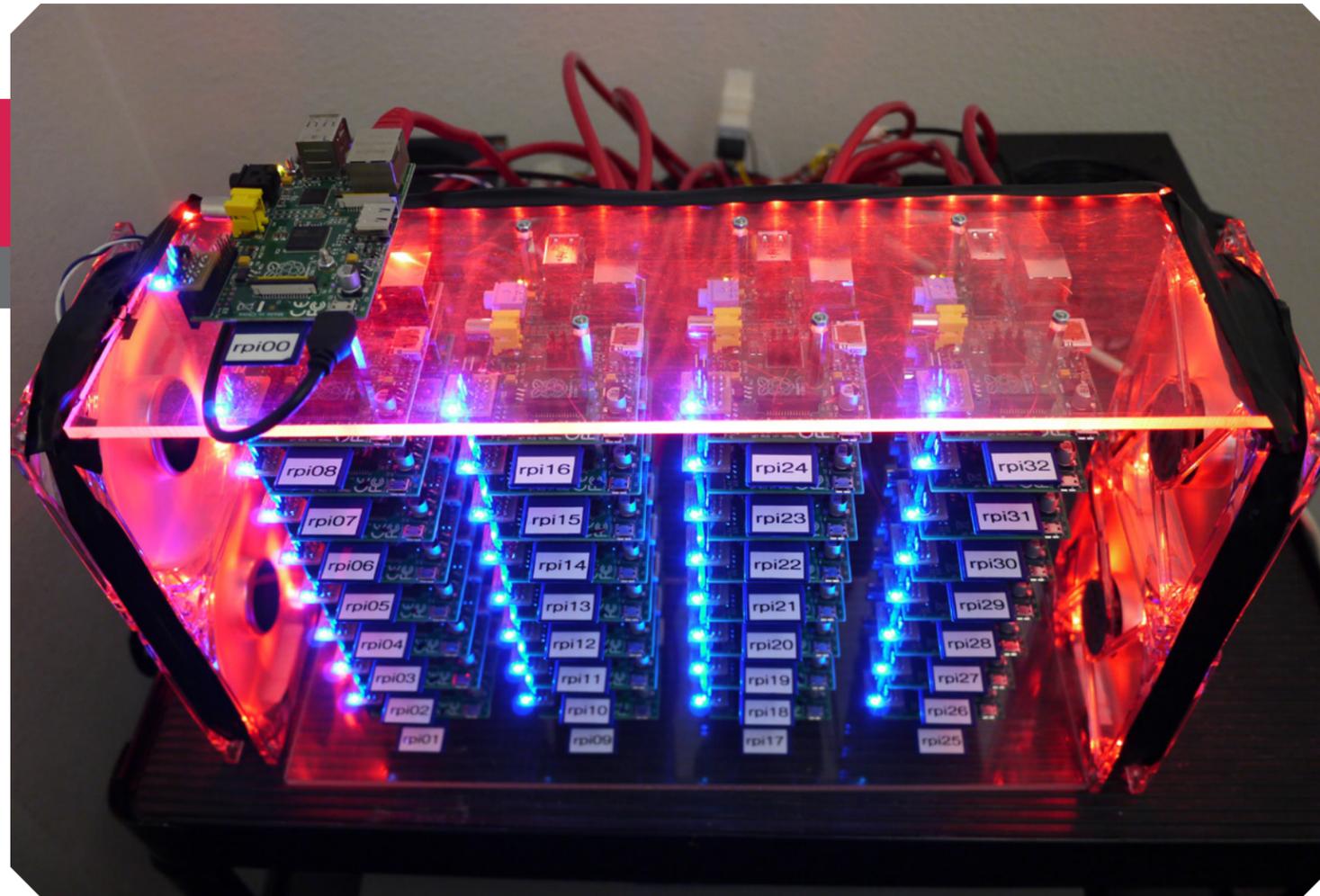
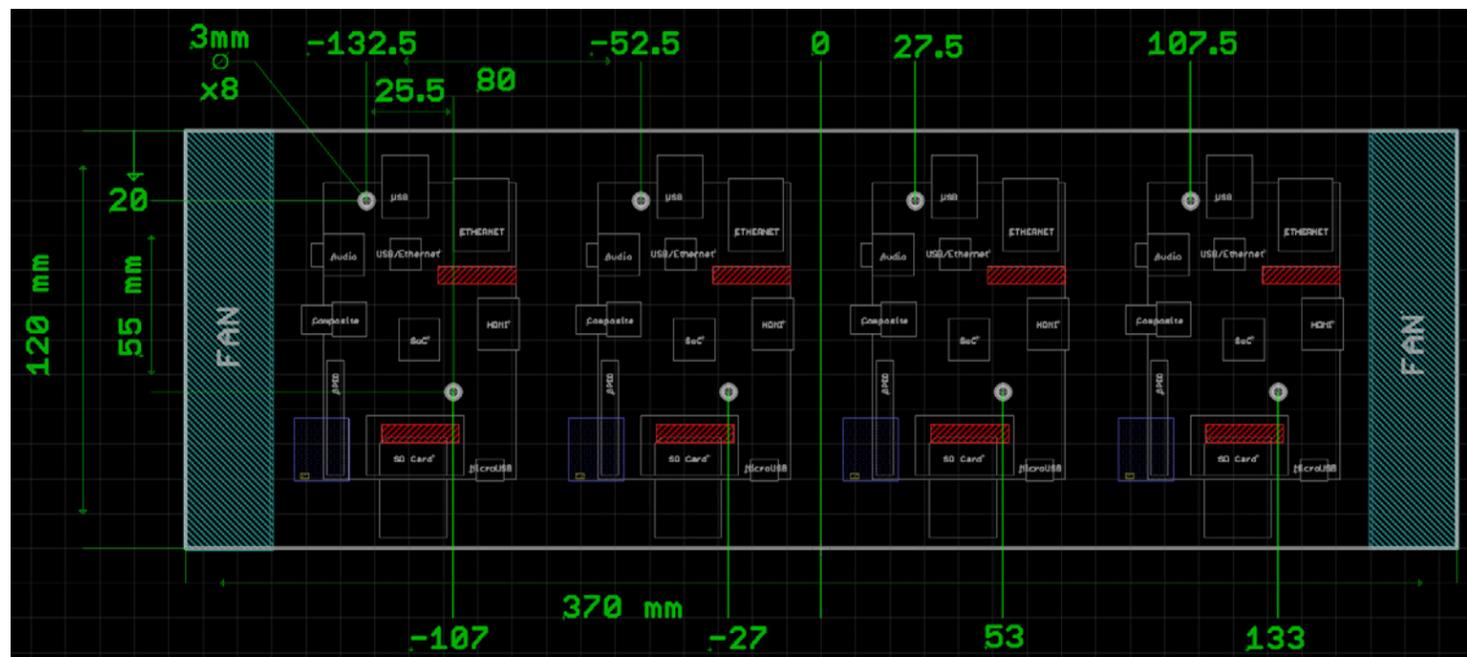


Figure 3a: Front view of the RPiCluster

Figure 3b: Plexiglas Layout (Top/Bottom of RPiCluster Rack)



Rack and Power Design

One aspect of the cluster design that required quite a lot of thought was the rack mounting system and power distribution method. In order to keep the cluster size to a minimum while maintaining ease of access, the RPIs were stacked in groups of eight using PCB-to-PCB standoffs with enough room in between them for a reasonable amount of air flow and component clearance. This configuration suited our needs for power distribution very well since it allowed for a power line to be passed vertically along each stack. Using this orientation, four RPi stacks were assembled and mounted between two pieces of 1/4" Plexiglas. This created a solid structure in which the cluster could be housed and maintain physical stability under the combined weight of 32 Ethernet cables. Figure 3 (a) shows front view of the cluster and its stacks. The Plexiglas layout for mounting each RPi stack was designed using EagleCAD, along with the power/LED PCB. Figure 3 (b) shows the layout.

Cluster setup is basically just configuring a few files so that all nodes can find each other by name on the network and have password-less SSH access to one another.

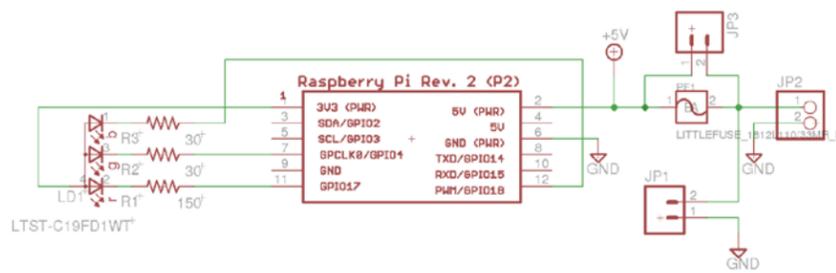


Figure 4a: RPiCluster Power/LED Board Schematic

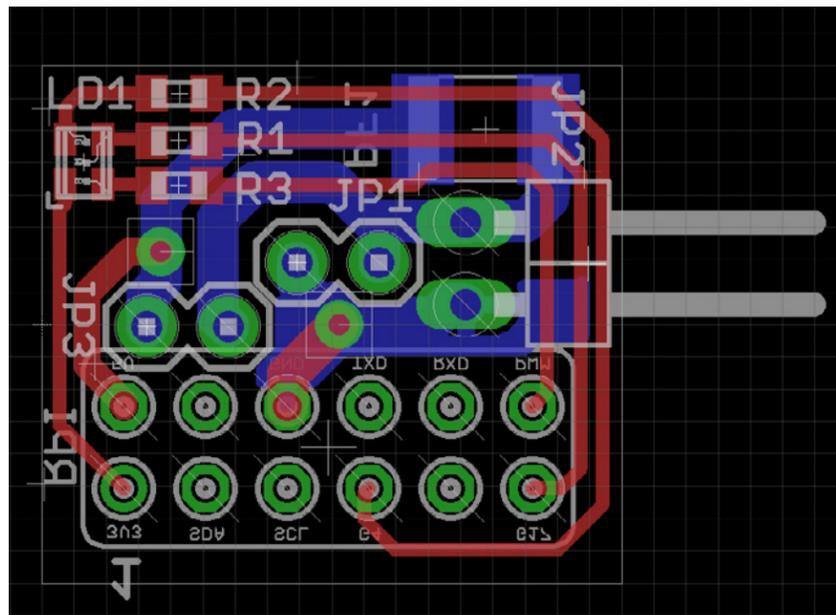


Figure 4b: RPiCluster Power/LED Board PCB Layout

There are two methods of powering an RPi. Each RPi has a microUSB port that enables it to be powered from a conventional powered USB port. Using this method would require a microUSB cable and a powered USB port for every RPi in the cluster, which would add a lot more cabling to deal with. It also means that there would be cabling on both sides of the cluster. Alternatively, there is an I/O header on the side of the RPi which contains a 5V pin that can be used to power the board externally. The latter option was chosen as it would also allow for additional customization and control of each node. A custom PCB was created to fit the I/O header to provide power and an RGB LED (since electrical engineers must, of course, have LEDs to show that their project is working). Figure 4 (a) and (b) show the circuit and PCB layouts, respectively.

As seen in Figure 4(a), aside from a RGB LED and some connectors, there is also a poly fuse (PF1). This was included to maintain short-circuit protection in the event of a board failure. The RPi already has a poly fuse between the USB power connector and the 5V rail. That fuse is bypassed when using the 5V pin to power the board. JP1 provides a 5V interconnect vertically between the RPis in each stack.

With the power being directly distributed via the Power/LED board, it was necessary to find a good source of 5V with sufficient amperage

to drive the whole cluster. Each RPi draws about 400mA at 5V (2W), thus we needed a minimum of 13A of 5V (65W) (and more for overclocking). You could, of course, buy a dedicated high output 5V power supply, but a great option is to use a standard PC power supply. PC power supplies are already designed for high loads on their 5V rails, and they are relatively cheap. The 430W Thermaltake (430W combined output for 5V, 12V, etc) we selected is rated to provide 30A at 5V (150W) and cost \$36. We opted to purchase two supplies to keep the overall load very low on each and allow for overclocking or future expansion.

Final Tweaks

As mentioned earlier, the RPi processor supports overclocking. Another noteworthy feature is that the 512MB of RAM is shared by both GPU and CPU, and the amount of RAM reserved for the GPU is configurable. Both of these features are configurable through modification of parameters in /boot/config.txt. After some testing, the whole cluster was set to run in "Turbo Mode" which overclocks the ARM core to 1GHz and sets the other clocks (SDRAM etc..) to 500MHz. The result was a proportional increase in performance (~30%) and power draw. Since the cluster does not require a desktop environment, the GPU RAM reserve was reduced to 48MB. 48MB seems to be the smallest amount of RAM allowed for the GPU on the current version of Arch Linux ARM for the RPi (it wouldn't boot with anything less).

Performance

As discussed earlier, the RPiCluster was not built with the intent of creating a HPC platform. However, it is still interesting to see where it ended up, right? There are, of course, many methods for measuring the performance of a cluster. Initially, I did some individual CPU tests comparing various platforms with the RPi. Table 1 shows the specifications of the platforms used for comparison. Each platform was compared on the basis of single core/single thread integer/floating point performance using the GNU arbitrary precision calculator: bc.

Figure 5 shows the performance of each platform, where the performance is measured in terms of how little time was required to complete the calculation.

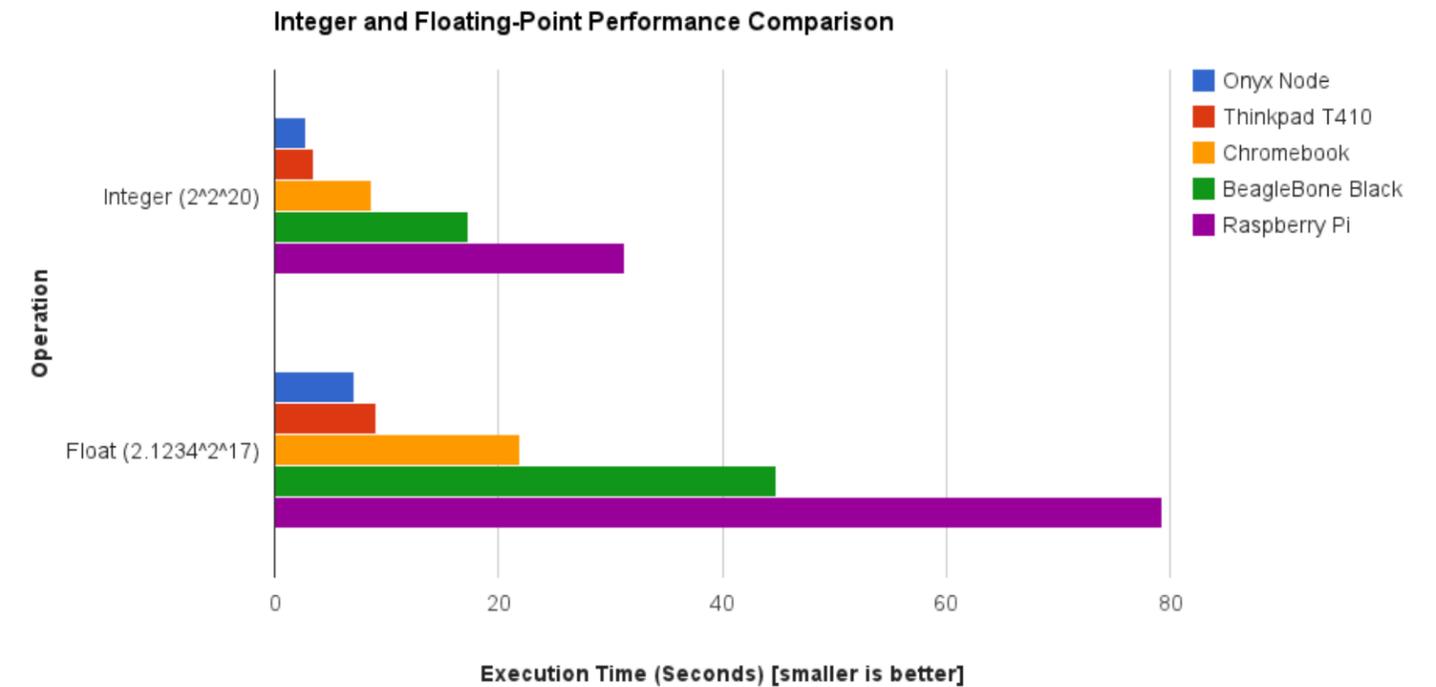


Figure 5: Integer and Floating-Point Performance (using bc)

Of course, after I published some details of the RPiCluster, everyone wanted to know what kind of MPI parallel performance the RPiCluster has. So, I took some time to get HPL (often used for the TOP500 list) working on the RPiCluster to determine overall parallel computing performance. For comparison, I also setup HPL on a single Onyx node (multi-threaded to leverage all the cores). Table 2 shows the resources available and the HPL results after being tuned for peak performance on both platforms.

As is seen in Table 2, the RPiCluster HPL performance is about one quarter that of the BSU cluster node (Onyx). Although, the RPiCluster can solve larger problems due to its greater available RAM.

Conclusion

Overall, the RPiCluster has successfully fulfilled its purpose. Since completing the build I have moved my dissertation work exclusively to the RPiCluster. Its performance has been perfectly acceptable for my simulation needs, and I have had the luxury of customizing the cluster software to fit my requirements exactly. ■

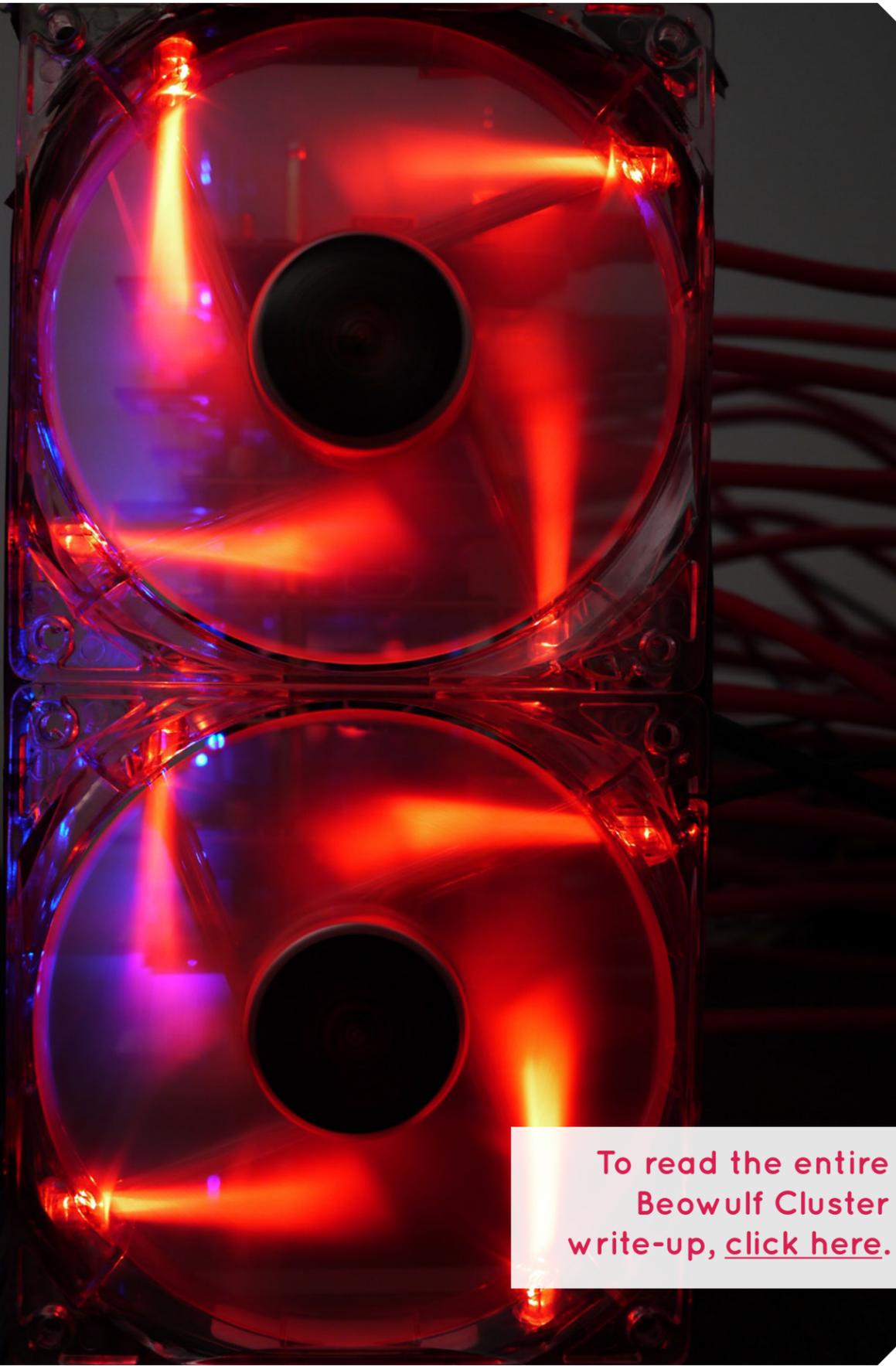
Platform	Cost	OS	CPU	Cores	MHz
BSU Cluster Node	\$1,250	Fedora 16	Intel Xeon E3-1225 (x86_64)	4	3,100
Thinkpad T410	\$1,450	Linux Mint 13	Intel Core i7 M620 (x86_64)	4	2,670
Chromebook	\$250	Arch Linux ARM	Samsung Exynos 5250 ARM Cortex-A15 (ARMv7)	2	1,700
Beagle Bone Black	\$45	Arch Linux ARM	Sitara ARM Cortex-A8 (ARMv7)	1	1,000
Raspberry Pi	\$35	Arch Linux ARM	BCM2/08 ARM11/6JZF-S (ARMv6)	1	700

Table 1: Single Core/Single Thread Integer/Floating-Point Platforms

Feature	Onyx Node	RPiCluster
Processing Cores	4	32
Clock Speed (MHz)	3,100	1,000
Total RAM (GB)	8	16
Available RAM (GB)	5	14.6
Available Disk Space (GB)	85	176
HPL GFLOPS	43.41	10.13

Table 2: Resources/HPL Performance





To read the entire Beowulf Cluster write-up, [click here](#).

Scheme-itSM

Design and Share Schematics Today!



Visit digikey.com/schemeit today!

Digi-Key is an authorized distributor for all supplier partners. New products added daily. © 2013 Digi-Key Corporation, 701 Brooks Ave. South, Thief River Falls, MN 56701, USA

Power Factor Correction Controllers

ISL6730A, ISL6730B, ISL6730C, ISL6730D

The ISL6730A, ISL6730B, ISL6730C, ISL6730D are active power factor correction (PFC) controller ICs that use a boost topology. (ISL6730B, ISL6730C, ISL6730D are Coming Soon.) The controllers are suitable for AC/DC power systems, up to 2kW and over the universal line input.

The ISL6730A, ISL6730B, ISL6730C, ISL6730D are operated in continuous current mode. Accurate input current shaping is achieved with a current error amplifier. A patent pending breakthrough negative capacitance technology minimizes zero crossing distortion and reduces the magnetic components size. The small external components result in a low cost design without sacrificing performance.

The internally clamped 12.5V gate driver delivers 1.5A peak current to the external power MOSFET. The ISL6730A, ISL6730B, ISL6730C, ISL6730D provide a highly reliable system that is fully protected. Protection features include cycle-by-cycle overcurrent, over power limit, over-temperature, input brownout, output overvoltage and undervoltage protection.

The ISL6730A, ISL6730B provide excellent power efficiency and transitions into a power saving skip mode during light load conditions, thus improving efficiency automatically. The ISL6730A, ISL6730B, ISL6730C, ISL6730D can be shut down by pulling the FB pin below 0.5V or grounding the BO pin. The ISL6730C, ISL6730D have no skip mode.

Two switching frequency options are provided. The ISL6730B, ISL6730D switch at 62kHz, and the ISL6730A, ISL6730C switch at 124kHz.

Features

- Reduce component size requirements
 - Enables smaller, thinner AC/DC adapters
 - Choke and cap size can be reduced by 66%
 - Lower cost of materials
- Excellent power factor over line and load regulation
 - Internal current compensation
 - CCM Mode with Patent pending IP for smaller EMI filter
- Better light load efficiency
 - Automatic pulse skipping
 - Programmable or automatic shutdown
- High reliable design
 - Cycle-by-cycle current limit
 - Input average power limit
 - OVP and OTP protection
 - Input brownout protection
- Small 10 Ld MSOP package

Applications

- Desktop computer AC/DC adaptor
- Laptop computer AC/DC adaptor
- TV AC/DC power supply
- AC/DC brick converters

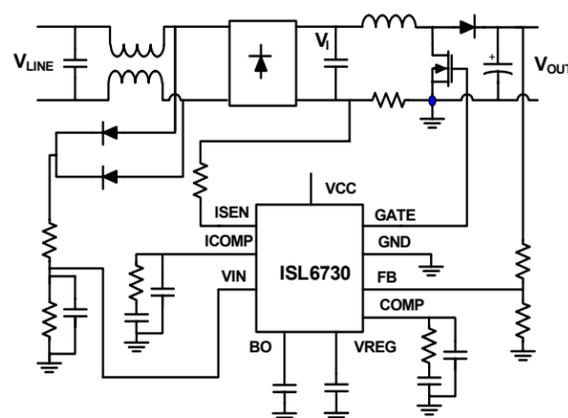


FIGURE 1. TYPICAL APPLICATION

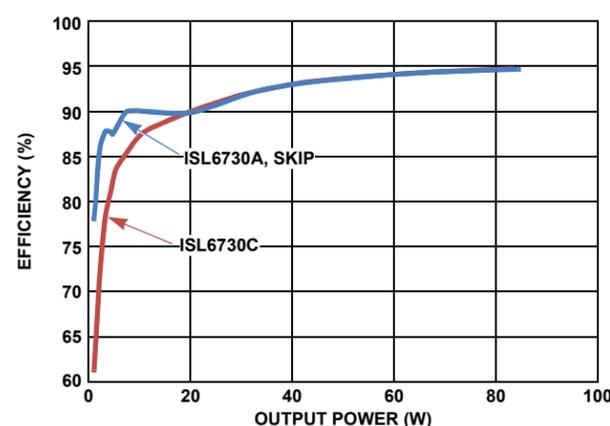


FIGURE 2. PFC EFFICIENCY

TABLE 1. KEY DIFFERENCES IN FAMILY OF ISL6730

VERSION	ISL6730A	ISL6730B	ISL6730C	ISL6730D
Switching Frequency	124kHz	62kHz	124kHz	62kHz
Skip Mode	Yes-Fixed	Yes-Fixed	No	No

Take the Risk out of High Voltage Failure with Certified Avago Optocouplers

IEC IEC 60747-5-5 Certified



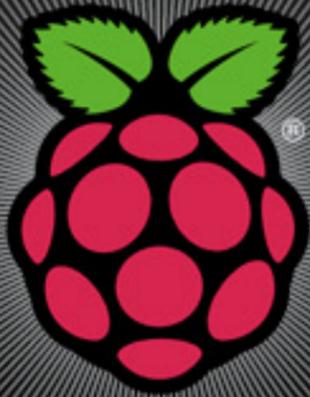
Optocouplers are the only isolation devices that meet or exceed the IEC 60747-5-5 International Safety Standard for insulation and isolation. Stringent evaluation tests show Avago's optocouplers deliver outstanding performance on essential safety and deliver exceptional High Voltage protection for your equipment. Alternative isolation technologies such as ADI's magnetic or TI's capacitive isolators do not deliver anywhere near the high voltage insulation protection or noise isolation capabilities that optocouplers deliver.

For more details on this subject, read our white paper at:
www.avagoresponsecenter.com/672

AVAGO
 TECHNOLOGIES

Your Imagination, Our Innovation
 Sense · Illuminate · Connect



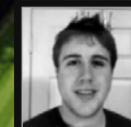


BAKING Pi

CAMBRIDGE UNIVERSITY'S INTRODUCTORY COURSE ON THE

UNIX OS

Electronics as a hobby and knowledge of the boundaries between hardware and software have long been something too expensive and too obfuscated for most people to access. It hasn't been until the last decade or so that many companies launched complete development boards which allow people to have varying degrees of control over both software and hardware -- for purposes of education, automation, and project construction. As of now, Raspberry Pi is the culmination of these efforts, giving people much more power than a similarly-priced Arduino would. While more for the purpose of interfacing with general purpose peripherals over HDMI, ethernet, and USB, the Raspberry Pi still includes multiple general purpose input/output (GPIO) ports-- allowing for a similar level of control. The Raspberry Pi can become a fully featured home theater PC or server—the possibilities are limitless. While people are able to use the Pi for a slew of pre-written applications using pre-baked distributions of Linux, perhaps the most useful thing to do with a Raspberry Pi is to use it as a platform upon which to learn how an operating system functions. Cambridge University's course Baking Pi is an introductory course on Unix designed to do just that.



Alex Toombs
Electrical Engineer

The wealth of information and source code available for Unix kernels make it the perfect place to start considering what an operating system looks like, especially for an integrated platform like the Raspberry Pi.

What is Unix?

Unix is a proprietary operating system originally written in the late 1960s. Written at the famous Bell Labs by a group of pioneering engineers, Unix was based around multiple users on the same system— an idea that holds true today. It was eventually rewritten in C and went on to inspire many other operating systems like Linux and BSD, distributions of which are core to the free software movement. Most of our technology runs off of Unix-like operating systems, including most phones, Apple computers, and even some consumer appliances. Figure 1 to the right shows a condensed history of the Unix operating system and of those systems heavily inspired by the original Unix.

Because Unix and Unix-like operating systems have been around for so long, there are so many facets to the operating system and many tutorials about how to learn and dive into it. And Unix is unparalleled when attempting to interface hardware and software in a computer. Linux, based upon Unix and released in 1991, is a completely open source kernel— where a kernel is an operating system component that manages the resources of the computer and interfaces the software with the available hardware. The wealth of information and source code available for Unix kernels make it the perfect place to start considering what an operating system looks like, especially for an integrated platform like the Raspberry Pi.

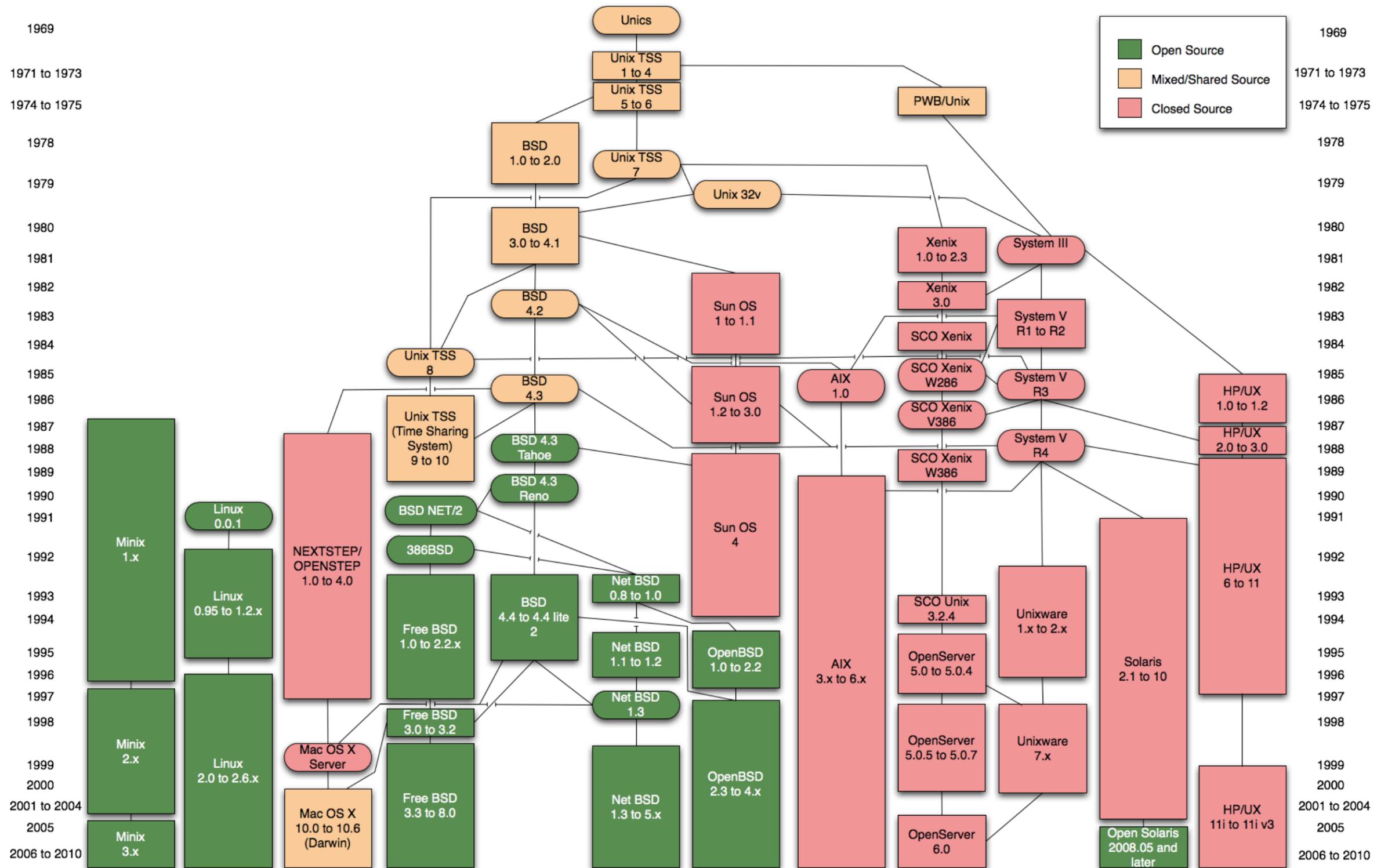


Figure 1: An Incomplete Unix-like OS History (courtesy of Wikimedia user Eraserhead1)

Baking Pi and Unix

Cambridge University's course begins by introducing what an operating system is and what it is written in. They first try to reduce the abstract concept of an operating system, tangentially familiar yet critically important to all of us, by calling it a "very complicated program". They briefly explain one of the core advantages of the Raspberry Pi's uniform hardware. Operating systems may be very similar, but as any Windows user knows, every piece of new hardware requires separate device drivers in order to interface with the OS. Lesson 0 also give a brief intro into assembly, which is a half-step up from binary. All compilers eventually translate your Python, C++, Go, Java, or other language down into assembly and eventually binary, which allow a physical circuit comprised of many transistors to switch on and off in a controlled manner. That level of translation and hardware control is handled by the kernel.

First Programs on the Pi

Lesson 1 directs the user to write a simple program to toggle on the board 'OK' LED, similar to the most basic of Arduino programs. To accomplish this, instructions are written in the ARMv6 assembly variant and compiled with the necessary GNU toolchain. The instruction largely has users copy and paste line by line, as the instructions are unclear. For instance, the lines

```
mov r1,#1
```

```
lsl r1,#18
```

```
str r1,[r0,#4]
```

are very difficult to read, though they (with some initialization lines) allow the user to control the on-board LED. Assembly makes us reflect upon how spoiled we are with our modern high-level languages sometimes.

Raspberry Pi has provided that affordable and standard platform upon which many people will be able to continue to learn about the technology that is at the very root of the devices we use every day.

Eventually, as users gain control over more facets of the blink LED, they can gain an admiration for the engineers that had built entire operating systems (or entire games, such as in the case of the original Roller Coaster Tycoon game) in assembly. The second half of the course even goes on to teach students how to control the screen output from the Pi to display simple images and text manipulations.

Impact of Baking Pi

Personally, the obscurity between hardware and software and the apparent magic behind integrated circuits led me to become an electrical engineer, though I left school lacking a lot of knowledge. I worried that graduating would leave me unable to continue to learn about that all-important bridge of the kernel and operating system. Cambridge has shown just how powerful a basic grasp of assembly can be when one is trying to learn about operating systems. Raspberry Pi has provided that affordable and standard platform upon which many people will be able to continue to learn about the technology that is at the very root of the devices we use every day.

About the Author

Alex Toombs is a recent EE graduate from the University of Notre Dame, concentrating in semiconductor devices and nanotechnology. His academic, professional and research experiences have exposed me to a wide variety of fields; from financial analysis to semiconductor device design; from quantum mechanics to Android application development; and from low-cost biology tool design to audio technology. Alex has recently joined the cloud startup Apcera as a Software Engineer. ■

» **CLICK HERE**
to comment on the article.

PartSim

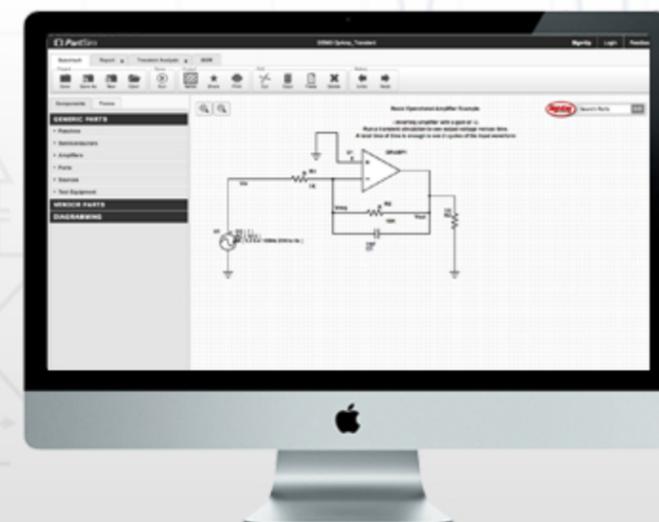
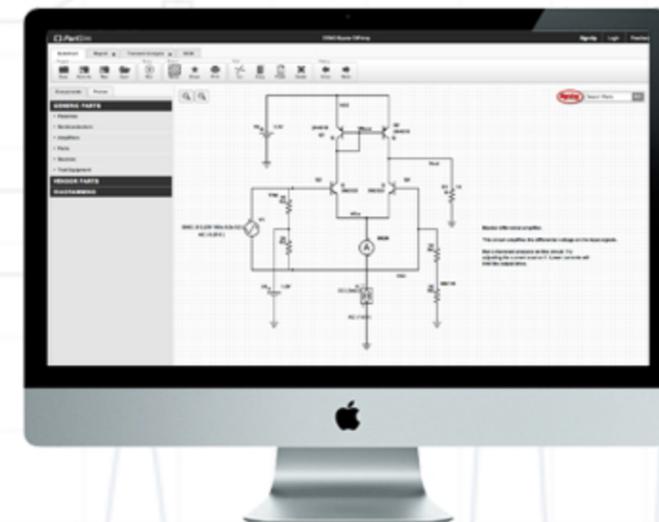
Online Circuit Simulator

PartSim includes a full SPICE simulation engine, web-based schematic capture tool, and a graphical waveform viewer.

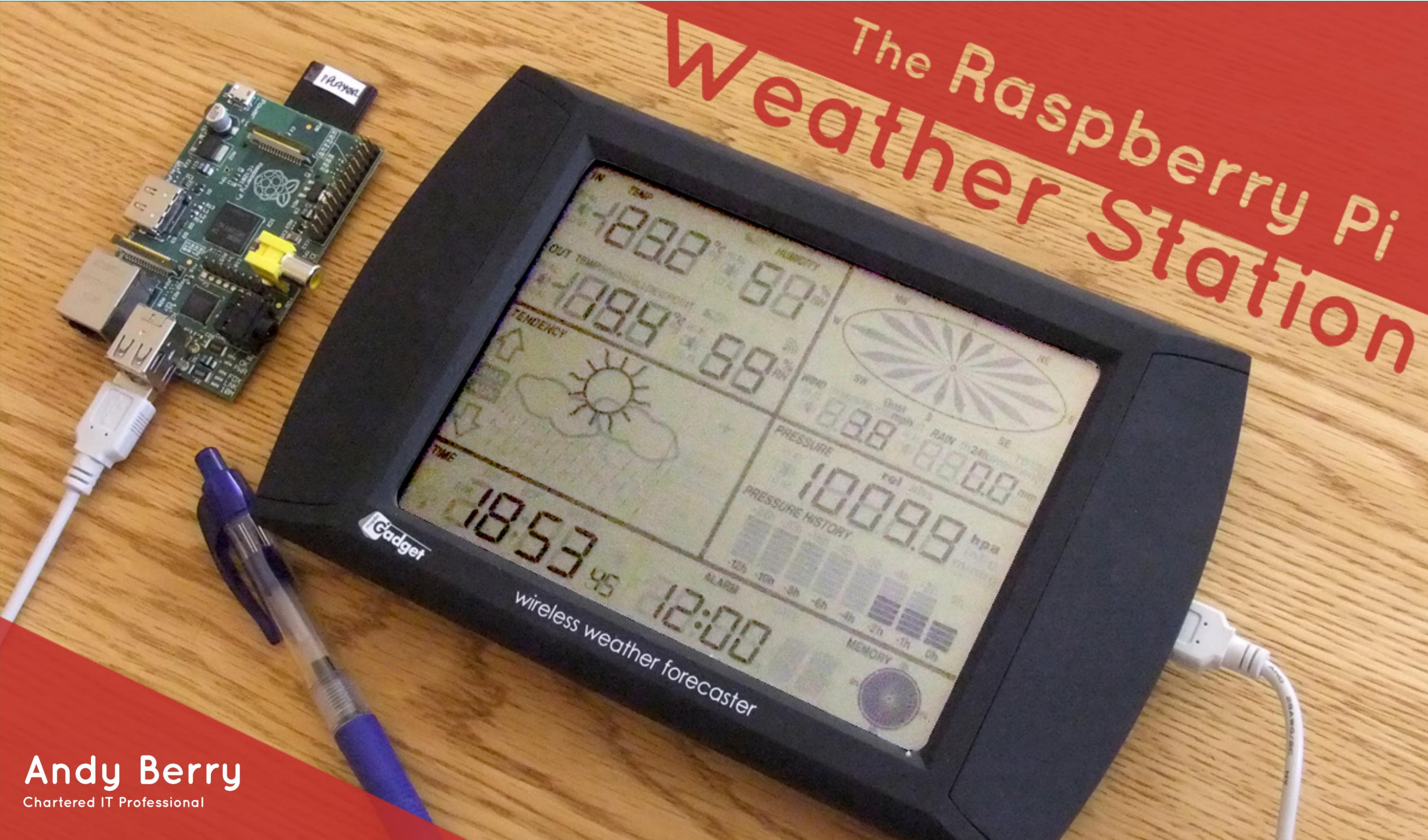
Some Features include:

- Simulate in a standard Web Browser
- AC/DC and Transient Simulations
- Schematic Editor
- WaveForm Viewer
- Easily Share Simulations

Try-it Now!



The Raspberry Pi Weather Station



Andy Berry
Chartered IT Professional

The Hardware

I first became aware of the Pi in late 2011, when Eben announced that a \$25 computer would soon be released. After learning about the computer I realized that the Raspberry Pi would fit well with another idea I had -- to setup a weather station and monitor overnight temperatures. The weather here in England is very unpredictable and I wanted to be more aware of the potential for ice on the roads, as I have a daily 50-mile motorcycle commute.

My first step was purchasing the Maplin Touchscreen USB weather station (although any Fine Offset WH1080 compatible weather station should work). The USB weather station has a rain gauge and temperature sensor and measures wind speed and direction. It communicates to its base station via a wireless connection which has a USB port for direct connection to a computer. This all worked brilliantly using the readily available software Cumulus from sandaysoft.com, and would have been fine not a Raspberry Pi, except for the following reasons:

- * I wanted to visually see the overnight temperatures on a graph.
- * I would have better access to the data if it was stored on a web site.
- * Leaving the PC permanently on was not an option.
- * I wanted a Raspberry Pi to play with.

The Software

It was no use trying to install Cumulus software on the Raspberry Pi, as the software is only available on Windows. I did find some Python software written by [Tim Easterbrook](#) called pywws, however, which worked well.

Getting Started

The software setup and connection to the weather station is actually quite simple, although it's important to read any error message and sort them out (these are usually caused by miss trying Unix commands). It's good to keep in mind during this process that Unix is case sensitive; cd Cython-0.19.1 is different to cd cython-0.19.1.

Though the software setup is simple, the challenging part is to create a website that would hold and display the weather and date.

First you need to load an SD card with the latest copy of Rasbian Wheezy as described by Kyle Olive on his [EEWeb blog](#). The default memory split of 64 will be fine as we are not using any extensive graphics. With the Raspberry Pi up and running you will be ready to connect and install the required programs.

Setting up the Connection

To set up the connection, first you need to install some libraries using git. Git hubs are used for software development and distributed version control. The later version of Rasbian already has git installed, so you can start by creating a directory called "weather" in the pi user home directory to store all the libraries and downloads.

```
** cd ~
** mkdir weather
** cd weather
```

Download the required file and libraries:

```
** git clone https://github.com/jim-easterbrook/pywws.git
** git clone https://github.com/gbishop/cython-hidapi.git
```

Install python-dev to get the header file python.h used by Cython and libusb1.0.9 used by cython-hidapi:

```
** sudo apt-get install python-dev
** wget http://sourceforge.net/projects/libusb/files/libusb-1.0/libusb-1.0.9/libusb-1.0.9.tar.bz2
```

The latest version of the Cython source is 0.19. It can be downloaded from <https://pypi.python.org/packages/source/C/Cython> using the wget command, but check that the version and #md5 is correct.

```
** wget https://pypi.python.org/packages/source/C/Cython/Cython-0.19.1.tar.gz#md5=df4c4fb2b8e3345272e3fc451b76bd630
```



Weather Station Sensor

Expand the source files into their own directory using:

```
** tar xvzf Cython-0.19.1.tar.gz
** tar tar xvjf libusb-1.0.9.tar.bz2
```

Compile and install Cython 0.19.1:

```
** cd ~/weather/Cython-0.19.1
** sudo python setup.py install
** Compile and install libusb:
** cd ~/weather/libusb-1.0.9
** ./configure
** make
** sudo make install
```

Before the cython-hidapi is compiled you will need to change the setup.py file so that it points to the correct location of usb1ib and arm-linux-gnueabi:

```
** cd ~/weather/cython-hidapi
```

Edit the setup.py file using:

```
** nano setup.py
```

and change the line:

```
** os.environ['CFLAGS'] = "-I/usr/include/libusb-1.0"
to os.environ['CFLAGS'] = "-I/usr/local/include/libusb-1.0"
** os.environ['LDFLAGS'] = "-L/usr/lib/i386-linux-gnu-lusb-1.0 -ludev -lrt"
to os.environ['LDFLAGS'] = "-L/usr/lib/arm-linux-gnueabi-hf-lusb-1.0 -ludev -lrt"
```

Copy the libudev file into the /usr/lib directory:

```
** sudo cp /lib/arm-linux-gnueabi-hf/libudev.so.0 /usr/lib/arm-linux-gnueabi-hf/libudev.so
```

Make sure your still in the cython-hidapi directory and run:

```
** sudo python setup.py install
```

Providing everything went according to plan we should now be ready to test the weather station link using pywws. Connect via the USB and run the following:

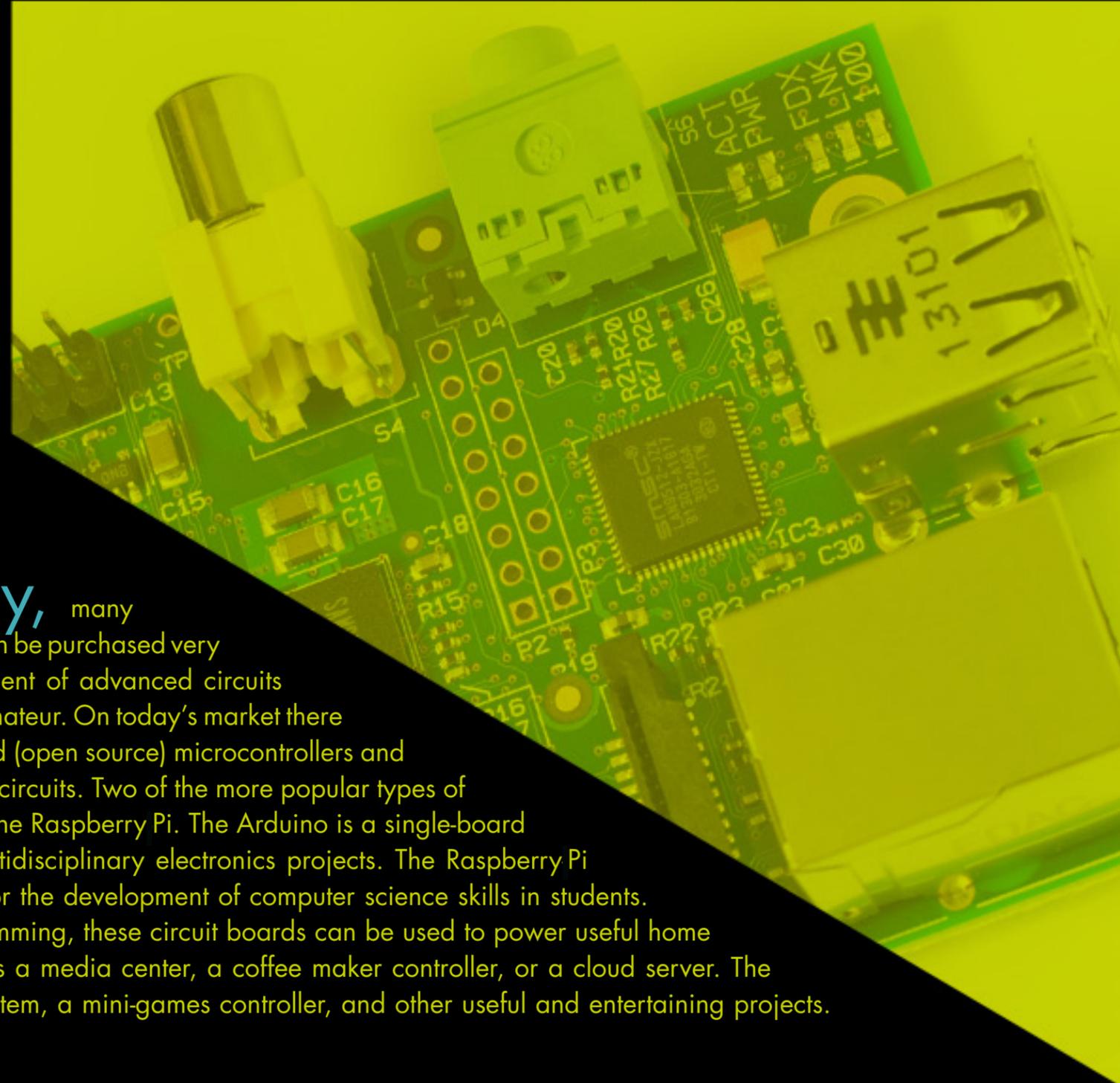
```
** cd ~/weather/pywws
** python -m pywws.TestWeatherStation
```


The Path to Pi

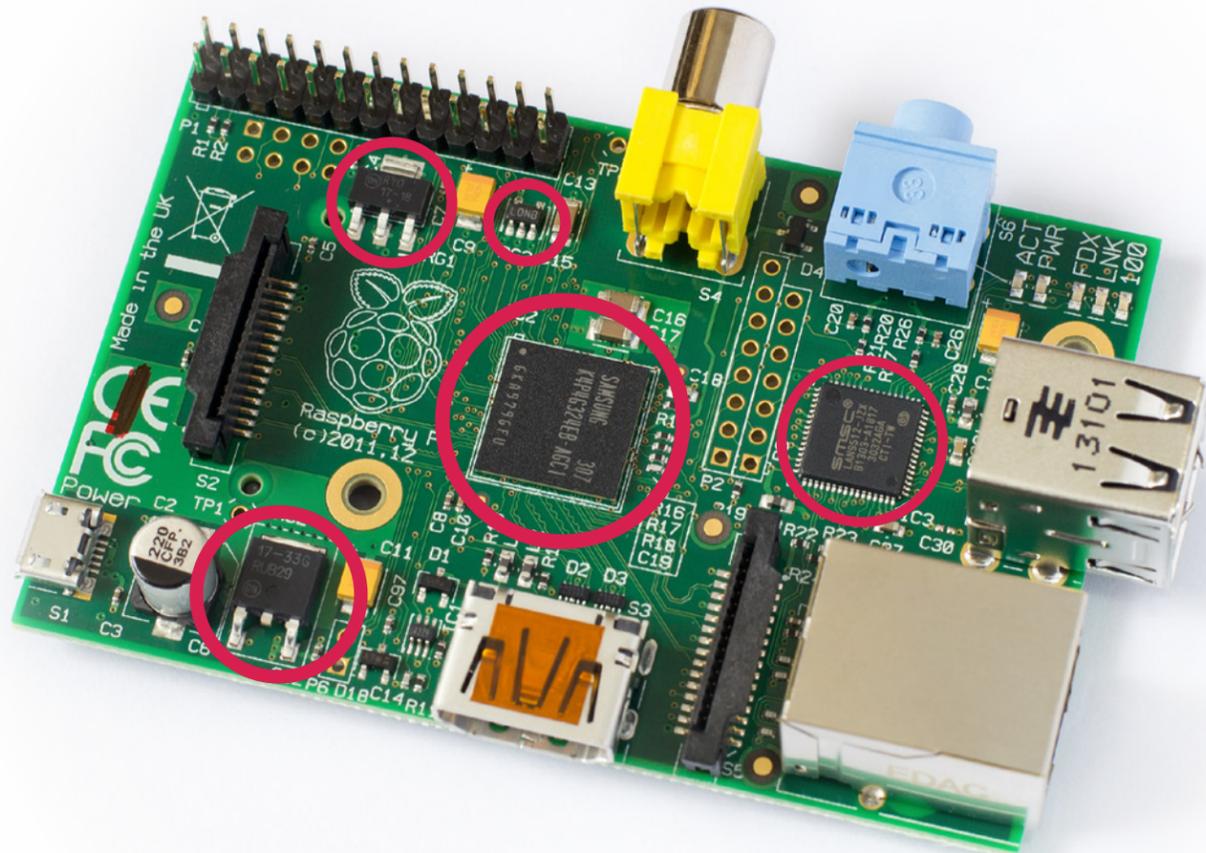
Innovations in open source software and hardware that made the Raspberry Pi possible.



Rob Riemen
Computer Engineering
Student at the University
of Cincinnati



Today, many OEM electronics can be purchased very cheaply, and development of advanced circuits can be accomplished by an amateur. On today's market there are several high quality single-board (open source) microcontrollers and computers that aid in the design of these circuits. Two of the more popular types of single-board electronics are the Arduino and the Raspberry Pi. The Arduino is a single-board microcontroller that aids in the construction of multidisciplinary electronics projects. The Raspberry Pi is a single-board credit card sized computer intended for the development of computer science skills in students. With some background knowledge in electronics and programming, these circuit boards can be used to power useful home projects. For example, a Raspberry Pi could be used to function as a media center, a coffee maker controller, or a cloud server. The Arduino could be used to build a holiday lights controller, a security system, a mini-games controller, and other useful and entertaining projects.



These mini circuit boards have tons of benefits. They help simplify tasks using electronics, and are small and compact so they can be hidden away from view and easily transported. They also help teach owners how circuits and programming works. But how did the circuit board come this far? Advancements in hardware capacity and a trend towards open source electronics both led the way for tiny, powerful, and user-friendly computers and microcontrollers like the Raspberry Pi.

Starting around the middle of the 20th century, engineers used vacuum tubes to construct programmable logic circuitry. These computers would take up whole rooms while only performing a number of simple calculations. Then the invention of the bipolar transistor changed circuitry forever.

It was found that through a base, collector, emitter setup, current flow can be altered to control the function of a circuit. Simplified, these transistors acted as a switch for current flow. Transistors were able to simplify as well as reduce the size of circuits. This made it significantly cheaper and simpler to produce circuits for use in electronics. In the picture above, five powerful IC's are placed onto a circuit board the size of a credit card. The IC's that are singled out are the largest on the board. Generally, because of the computations expected of these larger chips, they require more transistors--which in turn require more silicon space. There are plenty of ICs located on these boards that are slightly visible. The invention of the transistor, coupled with the realization of self-aligned gates, it made credit card sized circuit boards possible in the modern age.

After the transistor revolution came the integrated circuit. The integrated circuit (IC) was only possible after the invention of the transistor. Multiple transistors could be placed in a configuration called a self-aligned gate. This self-aligned gate

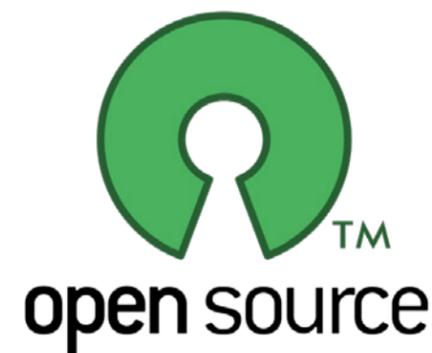
is a phenomenon where a MOSFET transistor uses a gate to mask the doping of gain and source regions of the chip. These gates reduce the capacitance that inhibits the normal functioning of MOSFETs.

The self-aligned gate has led to an increase in computer power by making transistors smaller and faster. Multiple self-align gates make up a single integrated chip to perform functions desired by the user. The invention of the self-align gates and MOSFET transistors pushed the computing age to the next level.

Ever since this development of ICs and transistors, there has been a trend categorized as Moore's Law named after Intel's co-founder Gordon Moore. Every 18 months, the number of transistors on an integrated circuit will double. Up until the past couple years, Moore's law has held, and this increase in transistors and power has kept the cost of electronic parts stagnant while at the same time increasing computing power. This trend has also made it possible to increase computing power of smaller IC's.

Electronic circuits have become cheaper to produce. Manufactures have multiplied computing power, making it possible to have a smartphone that has quadruple the computing power of a laptop produced ten years ago. Mini electronics are found at many stores around town.

The problem with many of the electronics sold today however, is that they are pre-programmed or designed with a specific purpose in mind. The circuits are patented by the developer, a scenario that isn't conducive to open development, or the use of tools such as Arduino or Raspberry Pi.



OPEN SOURCE REVOLUTION

The transmission of ideas freely and openly isn't a new concept. Recipes have been distributed freely for centuries. Henry Ford was trading automotive ideas between other manufactures for years. When computers and electronics became more prevalent, however, the hardware and software wasn't open to the public.

Beginning in the 1980's, many people were becoming disillusioned with the way hardware and software was being distributed. This was the catalyst that sent software developers into action. Many software developers, including Richard Stallman, saw that people need to cooperate with each other in order to continue the advancement of computers. Stallman propelled what would later be called the "open source" movement into the future by establishing the GNU/Linux project.

The GNU project is free in the sense that you can run the program for any purpose, modify the program to suit your needs, redistribute copies, either without payment or for a fee, and you have the freedom to distribute modified versions of the program. From this establishment, the movement lead to important developments regarding "free" software.

The GNU project helped spawn the software package of Linux. Linux is now a freely distributed Unix-like operating system under the model of free software. Late in the 1990's, the source code for Netscape was released in accordance to the free software movement. Because of this distinction that the source code for Netscape was being released, the movement became the "open source" movement.

Open Source software allows anyone to analyze and modify code without legal repercussions. The code can also be distributed freely and openly, which makes it easily available for use in education opportunities.

The open-source movement began churning out royalty free hardware and software, making it easy to develop projects at home. Now it was possible to learn how electronics function while learning how to program at the same time.

Things began to change as electronics and programming were becoming more prevalent. The open-source movement began churning out royalty free hardware and software, making it easy to develop projects at home. Now it was possible to learn how electronics function while learning how to program at the same time. With the increased importance of electronics came the push for faster, cheaper, and larger capacity hardware as well as the open hardware/software movement--making development of electronic projects possible for the everyday user.

Certain source code is now distributed openly and freely. Firefox and Thunderbird are derivatives of Netscape and are still open-source software to this day. This allows anyone to analyze and modify code without legal repercussions. The code can also be distributed freely and openly, which makes it easily available for use in education opportunities.

Programming circuits such as Raspberry-Pi and Arduino also use open source code and operating systems--for Raspberry Pi it is suggested to use Rasperian, the open-source operating system that was specifically developed for the Pi. The open source project allows users to freely share software that can be used to program these devices. This helps in the education of individuals interested in learning more about programming both circuits and computers, the basic foundation of Raspberry Pi and Arduino.

CONCLUSION

Starting with vacuum tubes, and moving to the invention of the transistor, the computer system has made great progress. With the invention of ICs and their implementation of transistors, we have been able to follow Moore's law with the construction of faster circuits. This has paved the way for the development of smaller circuits. Many of today's most useful computers and circuits are the size of a credit card. Being able produce smaller electronics in a cheaper fashion has made hardware available to many people. With this influx of affordable hardware, there has come a need to program the hardware. Software has been in development as long as circuits have been in use. But, there was a time where this software and its uses were heading down the wrong path, in terms of availability. Software usually wasn't available to everybody, and if it was, it had legal implications attached to it. Richard Stallman forged the open source movement by making the GNU/Linux available for all. This was the beginning of the open source movement and from then on many sources of useable software have been used and distributed freely. All of this combined to contribute to the availability of circuit boards like the Raspberry Pi. ■

» **CLICK HERE**
to comment on the article.

Return-to-ZERO



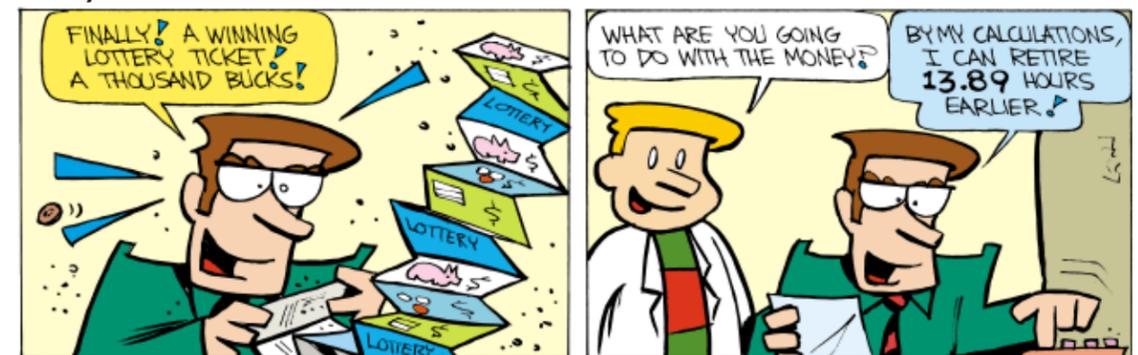
All in a Day's Work



Workaholic



Early Retirement



EEWeb
Electrical Engineering Community

Join Today

eeweb.com/register