USING A RASPBERRY PI AS A MAIN DEVICE IN HIGHER EDUCATION

Kyle Noland, Jamaal Hollins, Ryan Welker,
Joshua Smith, and Lucinda Caughey
Department of Computer Science
University of Illinois Springfield
Springfield, Il 62703
csc@uis.edu

ABSTRACT

There is a growing interest in Computer Science that has traditionally revolved around expensive machinery. The Raspberry Pi indicates that moderately priced and modestly powered machines can be useful in higher education. The Raspberry Pi enables a budget priced option for students to learn about core Computer Science principles that supplement the student’s coursework in such topics as programming, micro-processing, operating systems, network security, web development, and distributed computing. The Raspberry Pi allows students to participate in using a kit computer for learning by cautiously using trial and error practices with the device.

By encouraging students to actively participate in development of projects for the Raspberry Pi, it was possible to experience first-hand how a student approached development while having complete control over the hardware and software. Also by asserting complete control over the operating system environment, the student will be able to determine efficiencies in certain builds of operating systems over other alternatives.

Because of the relatively easy nature of completely starting over with a new operating system by using a new SD card, the student has no need to worry about causing software problems due to development. This peace of mind allows the student to use trial and error without worrying about possible consequences. Usually, this wouldn’t be practical on a daily-use, production environment computer. The Raspberry Pi also allows the student to learn Low-level programming with complete control of the hardware that normally may not be possible in a university core class.

Ultimately, the Raspberry Pi allows any student with the ambition to learn the ability to figure out how a computer really works and have complete control over its abilities. By enabling a student to use many peripheral items that they already have such as a USB keyboard, USB mouse, Ethernet cable, and HDMI cable, the price of effectively using a Raspberry Pi is minimal and an ideal supplement to a student's core curriculum in Computer Science.

Introduction

The primary focus of a Computer Science program should be to develop the knowledge, skills, and abilities (core competencies) sufficient for entry into a career in computing. One of the difficulties many undergraduate students encounter in developing the programming proficiency is its reliance on (relatively) expensive software running on expensive machinery.

Our experience with the Raspberry Pi indicates that these modestly priced / moderately powered machines can be an attractive alternative to traditional desktops and laptops. Coupled to a fully featured virtual environment (such as VMware VSphere), these budget machines can provide students with the environment they need to master core competencies.

In addition to being useable as a programming platform, the nature of these open-source/ 3rd part expandable platforms allow students to supplement coursework in such topics as microprocessing, operating systems, networking, and system security. This research details our experience in developing a kit-based Raspberry Pi solution for use as the primary platform for students in higher education.

Overview

The objective of this research was to explore how the creation of a computing platform based on the Raspberry Pi could be used to enhance undergraduate mastery of core competencies in the areas of:

1. Programming
2. Operating Systems
3. Networking
4. Microprocessors
5. System Security

THE SYSTEM

RASPBERRY PI

The Raspberry Pi computer comes in two configurations (Model A and Model B). Both computers are equipped with an RCA video port, a 3.5mm audio jack, a USB port, a microUSB power port, an SD card slot, an HDMI port and a 700Mhz Broadcom ARM11-based SoC (system on a chip). Model A of the Raspberry Pi has 128MB of RAM and no Ethernet connect. Model B increases the RAM to 256MB, adds a second USB 2.0 port and also add a 10/100 Ethernet port. [1]

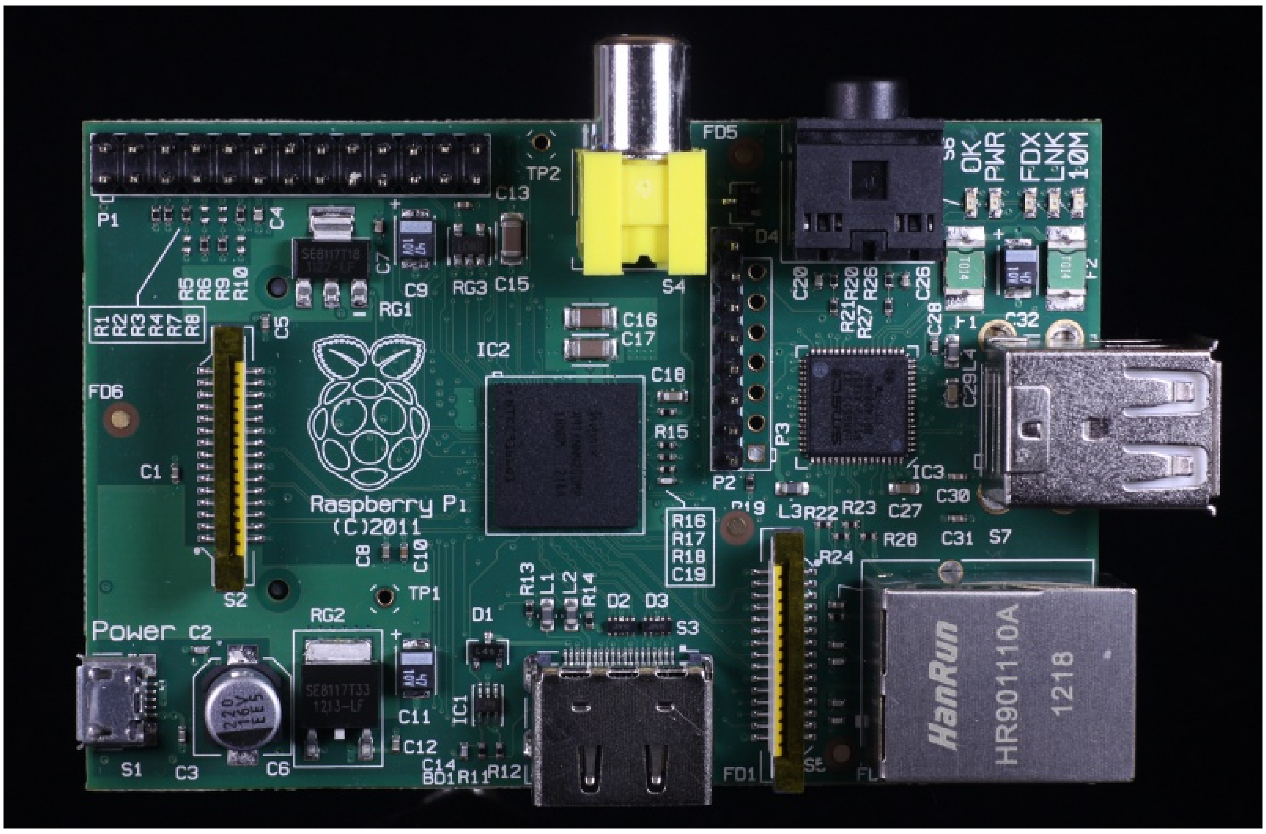

Figure 1: The Raspberry Pi main Circuit Board [2]

Peripherals

In addition to the base board, the following hardware is needed to build a complete working system:

- Power Supply: The power supply for the Raspberry Pi is a standard micro-usb charger.
- Bootable SD Card: The operating system is stored onto an SD memory card. The minimum size for the SD card is 2GB.
- USB Keyboard and Mouse
- HDMI cable and HDMI ready display (or, to use an analogue TV, a composite cable is required with SCART adapter) are required.

Accessories

The following non-essential accessories were also needed to make our system a complete workstation replacement:

- Powered USB Hub: The Raspberry Pi has only two USB ports that are usually connected to a keyboard and mouse. If you wish to connect anything else then you will instead need to connect to a powered USB hub.
- Ethernet cable: An Ethernet cable is required to connect the Raspberry Pi to a router. Either CAT-5e or CAT-6 cables can be used. The port is limited to 100Mbps so there is no benefit to using the (usually) more expensive CAT-6 cables.
- Wireless adapter: The Raspberry Pi (model B) includes an onboard wired Ethernet port, but to use with a Wireless network requires a USB adapter.
- Case: The Raspberry Pi is supplied as a bare board without any case.
- Speakers: While not required if you are using a TV, external speakers may be required for audio when using a monitor. Any standard audio speaker with a 3.5mm plug should work.
- Additional storage: USB flash drives or external USB disk drives can be connected to the Raspberry Pi to increase the amount of storage.

Software

The Raspberry Pi will run a range of OS Distributions and run a variety of software. At power-up, the CPU is offline, and a small RISC core on the GPU is responsible for booting the SoC. Because of this boot process, use of an SD card to boot the RPi is mandatory. The boot order and components are as follows:

1. First stage bootloader - This is used to mount the FAT32 boot partition on the SD card so that the second stage bootloader can be accessed. It is programmed into the SoC itself during manufacture of the RPi and cannot be reprogrammed by a user.
2. Second stage bootloader (bootcode.bin) - This is used to retrieve the GPU firmware from the SD card, program the firmware, then start the GPU.
3. GPU firmware (start.elf) - this allows the GPU to start up the CPU
4. User code - This can be one of any number of binaries. By default, it is the Linux kernel.

A Raspbian Linux image, as well as additional SD card images for other distributions, is available on the foundation's website [3].

Development Projects

With the base system created, projects in the core competencies were developed. The following presents a summary of these.

Programming

RPi Raspian comes pre-loaded with LXDE Window Manager, a lightweight, full-features UI. It also has the Midori web browser and IDLE3, a Python IDE. For a course focused on Python programming, The RPi is fully capable with this base distribution.

Java programming can be accomplished by using Java SE embedded and the Eclipse IDE. GNU gcc and g++ also work on Raspian. For software development courses which required the use of Microsoft Visual Studio or other software, the RPi is able to log into a virtual machine on the University's Virtualization Cluster.

Operating Systems

The process of creating the bootable SD card for the RPi is, itself, an application of basic operating system theory. To explore a different O/S, the user only needs to create a new bootable SD card. Currently we have successfully worked on Rpi's running Raspian, FreeBSD, ArchLinux, and Android.

Networking/ Network Security

The RPi model B has an Ethernet port built in, however, we have also built a system using USB wireless. Both Apache and Cherokee webservers will run on the RPi so it is possible to experiment with web server administration, MySQL, and PHP by creating a personal web server.

It is also possible to create a self-contained, very portable network hacking kit built atop the RPi running Raspian and the PwnPi penetration testing and network security tools software kit.

Microprocessors

The RPi uses the ARMv6 variant of assembler and it can be used to write programs that control the board itself. Programs include enabling and manipulating board's LEDs, graphics theory and creating basic geometric shapes and text, Display manipulation and OS controlled I/O. It is also possible to build your own command line interface and USB driver.

Conclusion

By encouraging students to actively participate in development of projects for the Raspberry Pi, it was possible to experience first-hand how a student approached development while having complete control over the hardware and software. Also by asserting complete control over the operating system environment, the student will be able to determine efficiencies in certain builds of operating systems over other alternatives.

Because of the relatively easy nature of completely starting over with a new operating system by using a new SD card, the student has no need to worry about causing software problems due to development. This peace of mind allows the student to use trial and error without worrying about possible consequences. Usually, this wouldn’t be practical on a daily-use, production environment computer. The Raspberry Pi also allows the student to learn Low-level programming with complete control of the hardware that normally may not be possible in a university core class.

Ultimately, the Raspberry Pi allows any student with the ambition to learn the ability to figure out how a computer really works and have complete control over its abilities. By enabling a student to use many peripheral items that they already have such as a USB keyboard, USB mouse, Ethernet cable, and HDMI cable, the price of effectively using a Raspberry Pi is minimal and an ideal supplement to a student’s core curriculum in Computer Science.

.

References

[1] http://elinux.org/RPi_Hardware retrieved March 8, 2013.

[2] http://elinux.org/images/a/a1/RPi-Front-JPB.jpg retrieved March 8, 2013.

[3] http://www.raspberrypi.org/downloads retrieved February 18, 2013.