ASHESI UNIVERSITY COLLEGE

AFFORDABLE DESKTOP COMPUTING USING LOW END HARDWARE LIKE RASPBERRY PI AND CUBIEBAORD

NII LOUS NELSON

2014

Applied Project

ASHESI UNIVERSITY COLLEGE

AFFORDABLE DESKTOP COMPUTING USING LOW END HARDWARE LIKE RASPBERRY PI AND CUBIEBAORD

By

NII LOUS NELSON

Dissertation submitted to the Department of Computer Science Ashesi University College

In partial fulfillment of the requirement for the award of Bachelor of Science in Management Information Systems

April 2014

Declaration

I hereby declare that this dissertation is the result of my own original work and that no part of it has been presented for another degree in this university or elsewhere.

Candidate's Signature:……………………………………………………………………

Candidate's Name:………………………………………………………

Date:………………………………..

I hereby declare that the preparation and presentation of the dissertation were supervised in accordance with the guidelines on supervision of dissertation laid down by Ashesi University College.

Supervisor's Signature:…………………………………………………………………..

Supervisor's Name:…………………………………………….

Date:…………………………………..

Acknowledgements

I acknowledge with sincere gratitude the financial, emotional and motivational support of my parents, Mr. Aruna Nelson, Mrs. Ayesua Nelson, my two sisters, Shefi and Naa Luisa Nelson for their encouragement and prayers throughout this project.

My appreciation also goes to my supervisor, Mr. Kwadwo Gyamfi Osafo-Maafo of the Faculty of Computer Science, Ashesi University College, for his immense support. The valuable time he spent in guiding me throughout this project.

Nina Chachu, Head Liberian at Ashesi University College, Dr. Sena Agyepong of the Faculty of Business Administration.

To my friends Mawuena Ahadzie, Ebenezer Agoe-Ashiakwei, George Donkor, Kevin Eshun, Amos Akoto, Nakoh Dede Appiah and Araba Buadu, thank you for your time and support in kind.

Finally, I wish to express my sincere appreciation to the Ashesi Community for all it has taught me.

Table of Contents

Abstract

The field of computer science over the past two decades has witnessed a huge leap in the areas of improved systems design, hardware processing power and the development of software that maximizes these advances in systems. This leap can attributed to continuous innovation that drives the field to new heights at an ever speedy pace. The pace at which the field is growing requires educators to be abreast with new and changing technology. It is the desire of educators to have access to physical laboratories and computing centers in which they can physically train students. The challenge faced by educators in executing this task is the high expense that comes with the construction of physical laboratories. This write up stems from a project that aims to tackle this challenge by providing low powered, low cost, portable computing systems for students, laboratories and computing centers using devices such as the Raspberry pi and Cubieboard. Although not a perfect system, it promises to be a desirable solution that can be adopted across a wide range of institutions.

Chapter One

Background

Technological competence is dear to the hearts of students of Ashesi University because it's an important tool that aids them in completing course work and also forms part of a student's learning goals. Since Ashesi's inception in 2002 it has been a keen concern for school authorities to provide computing resources for the student population. Ashesi has undertaken a number of initiatives to see to it that computing resources are readily available to the student body. An Interview with Gloria Kwei of Ashesi's IT department revealed that as at 2011 the university was providing N-computing solutions alongside standalone computers in one of its computer labs [1]. Applications that run on these computers were installed on a server and were accessed through the network. N-computing was an inexpensive solution for the school since all it needed to function well were monitors, a solid server and a stable network, all of which the school had at the time. This system proved to be working up until students started taking out network cables to plug into their own laptops. The current system that works for the university is the provision of standalone computing systems which have applications installed by the IT department. This setup requires a large monetary investment and is affected by rapid technological obsolescence, tedious and time-consuming maintenance, and places limits on physical space. However, current advances in technology has made it possible for powerful computing systems to be built into smaller hardware units. With these new hardware units, networked laboratories and computing centers can be fitted with low powered, low cost and portable computing units which are easy to install, configure and maintain.

This work exploits affordable low end computing solutions as a new way of facilitating and enhancing the delivery of computing solutions to students in primary, secondary and tertiary institutions with a focus on Ashesi University College. The work also highlights how this solution is easier to administer, less expensive and frees resources which otherwise would have been used on other physical systems.

Motivation

As part of the Ashesi University's requirement for graduation, every student has a minimum of forty hours of community service to complete. This project is inspired by the author's service learning project, where he taught basic computer lessons to both second year and final year junior high school students of the Berekuso Primary School. Pupils of the Berekuso Primary School were allotted a time period during the week when they could visit the university and have practical computer sessions. As a result of the limited lecture halls and constraints on scheduling for lectures, the school's computer labs are used as lecture halls. This made it impossible for pupils to have access to the labs for practical sessions. During the author's community service he was compelled to print basic computing procedures on paper to explain to the pupils. At the time of the author's community service laptops couldn't be used to teach because the classrooms in the Berekuso Primary School didn't have access to electricity. The pupils found it difficult to visualize and comprehend the various lessons that were printed and displayed to them on paper. The author's interest in small portable computing units moved him in thinking of a computing solution that is inexpensive, consumes little power, is easy to deploy and maintain and can be used in building computing centers for primary, secondary and tertiary institutions who cannot afford upscale and expensive computer laboratories.

Objectives

This project is a design and systems implementation project that is aimed at providing low powered, inexpensive computing system that student can tinker with, and explore computing processes without fear of breaking an expensive device. The first implementation of this system will be to integrate it into the computer laboratories of Ashesi University and get students of the university to use the system in their everyday computing tasks.

This project is therefore focused on finding a suitable operating system that can smoothly run on the selected hardware device, an operating system that can support applications used by students of the university in completing course work, customizing the operating system to have an interface and navigation system that is user friendly and easy for students to transition to, providing quick setup and configuration tools for students and system administrators of the university and providing a support system that will aid students with difficulty they might encounter using the system.

Related Work

A number of projects around the world have tried to design and implement low cost computing solutions for people mostly in developing countries. This section discusses projects and articles that have in one way or the other tried to tackle the issue of providing low cost computing solutions. The One Laptop per Child project uses low cost laptops to meet the computing needs of children in poor and rural areas of the world. Ultra Lost Computing and Developing Countries, looks at the use of the Raspberry pi and an alternative computing unit for poor communities in the global south of the world. An Affordable Virtual Laboratory Infrastructure to Complement a Variety of Computing Classes, is a project where visualized computing is implemented to provide complementary computers for lab sessions at the University of Michigan.

<u>The One Laptop per Child (OLPC) Project and Its Applicability in Ghana</u>

The OLPC project is an initiative headed by Nicolas Negroponte [2] with the aim of expanding the use of computer technology for school children in mainly the poorer and more rural areas of the world. The project doesn't only seek to narrow the "digital divide", but also seeks to improve educational opportunities for children by providing resources that will engage them to learning through the use of the internet. Devices developed from the project were originally priced at 100 U.S. dollars per unit. In June 2007 a final beta version of the machine, the B4, was produced. In the same year the project planned to develop more computers (numbering into the millions) for deployment in schools of selected countries. Some of these countries include Brazil, Nigeria, Thailand and Uruguay. At the time this article was published Ghana was listed by the OLPC as a country which was currently seeking government support to advance OLPC in the country. For the OLPC, saturation of laptops into communities is an important aspect of their vision. Previous attempts to provide computers to schools resulted in the construction of school computer labs. By providing laptops, students have access to the laptops longer than just the time they spend in school [3].

The fundamental idea of providing cheap laptops to students has been challenged by critics of the OLPC [2]. In their opinion food, water and basic medical care should be considered before OLPC. Negroponte doesn't disagree with this claim, but he argues, the tools needed in providing food, good quality drinking water and any necessity of a community are all bound in education [3]. Another critique of this initiative is the question of how governments of poor nations will cater for textbooks and training of teachers for the classroom if they should spend their monies on laptops. The authors respond to this critique by saying the OLPC project promises to bring whole libraries of textbooks and other educational material directly to children [3]. Other arguments thrown at the project are that, the technology is not right. Instead, a better and cheaper solution will be to provide internet connectivity via net-enabled cell phones which can be accessed in remote areas of the world [3].

XO, is the name of the OLPC laptop hardware that doubles as a laptop and eBook reader. It is 224mm x 228mm x 30mm and weighs 1.5kg. Its design takes into consideration rugged environments, low or no electricity conditions and a 190.5 mm display that is readable in direct sunlight. For choice of operating system the OLPC maintained an open source philosophy and runs on a stripped down version of

Linux. A modest processing power provided by AMD's Geode running at 433 MHz uses only 0.8 Watt of power. The "extreme suspend" technology utilized by the laptop's processor contributes to its low power consumption. On the power front, the overall power consumption of the OLPC laptop is between 0.1 Watt and 3 Watts making it well equipped for condition in which electricity is not reliable. There is no hard disk in the XO because of its intended use in rugged environments. Mass storage is made up of 1 gigabyte flash ROM with 256 of DRAM. An external hard drive can be connected via USB. After a detailed analysis of the OLPC laptop, the Buchele and Owusu-Aning delve into the system's applicability in Ghana. According to an assessment of Ghana's educational sector, Ghana falls within the range of countries for which the OLPC was developed. The authors' claim that the OLPC project will certainly address the problems of access to and quality of education in Ghana. The problem of access will be solved because laptops will enable students to have access to educational material at home and in school. As at 2007 the successes of the OLPC was yet to be accessed since the pilot projects had just kicked off.

Ultra-Low-Cost Computing and Developing Countries. Raspberry Pi and its potential in the "global south"

This project looks at ultra-low cost computing (ULCC) as a solution to the computing needs of developing countries. In the article the mention of ULCC is in reference to the Raspberry Pi which is a device created by Eben Upton, founder of the non-profit Raspberry Pi Foundation, to address the decline in numbers and skill levels of school levers applying to study computer science. In this article the Pi is referred to as a "back to the future" concept to open information technology in ways that will get young people tinkering with technology. To achieve this, the device had to be very low priced and an open design. The articles goes on to talk about the ingenuity with which the Raspberry Pi is designed and the philosophy behind its existence, but goes on to ask if Pi can break out of its computer science teachings and design project ideology to meet the needs of developing countries. It is important to ask this question because of the promise that ULCC “will match the explosion in digital communication with an explosion in data processing capacity”. With the use of ULCC the article makes clear three application opportunities. The first of these is in the area of micro-enterprise and household computing where the Pi in this case can provide access to the internet and standard computing applications for individual enterprises and households. The second opportunity is in the area of technical education. The main idea behind the Pi is to rekindle interest in computer science among school children. In parts of the developing world there is the thirst for IT knowledge in schools, colleges and other educational institutions. Constraints in budgets are the major issues faced by most of these institutions. The Pi might be the answer to their predicaments and also allow students to tinker more with technology. In connection to this second opportunity just discussed the article makes mention of a project in a village in Cameroon that employs this idea. The third and final opportunity is in the area of using the Pi as a monitor or control mechanism in embedded systems that require automation. Examples mentioned by author include using the Pi in developing smart motor controllers that can save power and extend motor life, and climate change measurement devices. But with all these opportunities the spread of computing applications has been limited by design related gaps. The author of the article sets out to claim that “if ULCC can become widespread, they can enable a new computing paradigm: grassroots innovation in which local users are also designers; creating designs that match local needs, resources, and context." But will these opportunities be ever realized, given the history of failures associated with similar initiatives such as the simputer and the red flags raised against the OLPC project? The article goes ahead to outline three activities that need to be effective for success to be achieved. The first of this three is paying close attention to the supply factor. For ULCC to work in developing countries, development of low cost monitors will be key. The issues of production, local distribution and marketing will have to be effectively tackled. The Raspberry Pi Foundation is cited as trying to make this possible but the answer might be in allowing private manufactures to replicate designs and produce. The second activity that needs attention is the demand factor. Why is there a demand for ULCC and if the cost is not the selling point, then what is? The articles goes on to state that tinker – ability might be the major selling point, but the bottleneck facing this selling point is that educational institutions in the developing world "do not want to tinker, they want something that will deliver useful applications. For these institutions the Pi might not be that device that suits their needs. The alternative to this challenge as stated in the article is the opportunity of the Pi being used as a monitoring mechanism in embedded and control devices and providing localized applications that are relevant to people. The third challenge that needs to be considered is the contextual factors. Even if supply and demand challenges are solved, ULCC in developing countries face the challenge of poor access to electricity and limited skill infrastructure. For the author the answer to these challenges possibly lie in the design characteristics of the device. Because of its low power consumption, ULCC can be easily used in places where electrification is a problem. With the issue of limited skill, ULCC can bundled with information that can easily be accessed by people in the developing world. The author concludes by speaking of the promise that ULCC brings with it, but also highlights all the challenges of scaling ULCC to people in very deprived settings. The hope though is that people will come about and re-conceptualize and develop ideas, initiatives and partnership that will make the ULCC philosophy a reality [4].

An Affordable Virtual Laboratory Infrastructure to Complement a Variety of Computing Classes

This project explores virtualized infrastructure as a way of implementing affordable computing to facilitate better classroom experience. The project also highlights that virtualized infrastructure is easier to administer, frees resources for other uses, and provides consistent and repeatable setup. A virtualized lab infrastructure was made up of a virtual computing environment, which uses a power server capable of hosting many instances of virtual machines on a single host computer. In this project the host computer was a Dell PowerEdge Server, consisting of 8 cores, 128 gigabyte of physical RAM, and 2.5 terabytes of physical disk space configured in a RAID 10 array. A machine of this sort is capable of hosting 200 virtual machines simultaneously. Students can log into the host server using a web interface portal which provides both a management interface and a console interface to whatever virtual machine is created by the user. One motivating factor for the implementation of this project is the flexibility of using virtual machines that allows multiple operating systems to coexist on a single physical machine. Also virtualization provides safety in the form of insulated environments where if one machine fails others are not affected. Virtualization also facilitates effective software development for different environments. Not only does virtualization allow software development in different environments or is it safe, it is also cost effective, with a single server machine emulating several machines depending of its configurations. The authors stress on the fact that virtualized machines provide secure, isolated and easily configurable environments in which both students and lecturers find it easy to conduct exercises. With virtualization students are provided with the option of starting from identical setups, resetting their machines when something goes wrong and the ability of being able to have access to a greater number of machines for their exercises. The authors also state some disadvantages of the system. With this implementation it is difficult for lecturers to give students direct experience with the hardware components because working in virtual machines is all done in software. This has effects on some classes such as those requiring Internet appliances such the installation of IP phones which required plugged in Ethernet ports to work. At the time this article was published, virtualization of labs occurred for Operating Systems, Database and Security and Distributed Systems classes.

Considering the various projects discussed in this sections, there is proof that low cost computing is an area of interest to educators around the world. OLPC and the Raspberry Pi project are both targeted towards poor and deprived communities in the developing world. The advantage of the OLPC project is that school children can have access to learning material at school and at home. The laptop hardware is designed just right to withstand the harsh conditions in rural areas of the developing world. For the raspberry pi project, not only can the device be used by children of school going it can also be used in enterprises as point of sale devices, and also in embedded systems. Unlike the OLPC and Raspberry Projects, the project on visualized laps aims to provide complementary computer systems for students. The advantage of this system is the ease of deploying and maintaining both operating systems and software. It is important to note that for this system to be successful a considerable huge amount of money is needed to build labs that are well networked.

Chapter Two

This chapter contains information on the author's proposed steps in designing and implementing the project. It also contains some literature review of various hardware components.

Approach

Requirements

For the execution of the project there is the need to find or build a low end computing device that is inexpensive, easy to maintain and deploy. It is also important to get an operating system that can smoothly run on the selected device without lags or any other problems. The operating system selected must be able to support most if not all of the software requirements of the student body. Ease of customizing the operating system is an important factor in determining the type of operating system to be used. In order to develop a device that suits the computing tasks of the Ashesi student body it is important to gather information on the various academic related tasks and exercises that students of the university use computers to accomplish. Determining the computing needs of the student body will provide enough information to know the precise software to download for the device.

Hardware

Research will be conducted to find a number of low end devices that are below the price tag of 100 U.S. dollars. The research will most likely reveal a number of devices that have been developed by either foundations, universities or individuals. Research will also be conducted to find low end laptops and tablets that meet the price range or are slightly above the 100 U.S. dollars and meet the requirements of the project. Possible hardware components that will be looked at include Raspberry Pi, Cubieboard and Arduino. Details of the finds will be outlined Appendix C.

Finding Operating Systems

Research into hardware components will reveal a number of boards, and low end laptops and computer that will suit the requirements of this project. For each hardware device found, extended research will be conducted to find the different operating systems that run on it. Propriety operating systems such as Microsoft Windows will be highly considered as a suitable operating system for the project since most users are familiar with the operating system and also its architecture. For the purpose of cost saving, open source operating systems such as Ubuntu, Fedora and other variants of Linux distributions will be looked at. Although a number of operating systems will be considered, the ultimate choice will depend on the hardware device selected and the type of operating system it can support.

Finding Software

Based on the finds from research conducted on the operating systems that run on the various hardware devices and that meet the project requirements, the search for software that effectively run on the operating systems will be undertaken. It will be important to note the various ways by which these software can be installed as some might need to be installed through the use of command line code such as yum, apt-get or installed through the downloading of packages or even compiled. In order to compile a list of software that will be used by the three working groups (BA, MIS, and CS) in the university an interview will be conducted between the author and at least five students from every major group from sophomores to seniors.

Customizing UI

Most students of the university use Microsoft Windows operating system on their laptops with a select few using Apples Mac OS and various distributions of Linux. Computers provided by the school also run on Microsoft's Windows platform. In finding an operating system for the project, it is important to note that if the hardware device selected for the project does not run any of these operating systems stated earlier, there will be the need to customize the user interface of the selected operating system so students will find it easy to transition from an interface they already know to what will be installed. For this to be achieved research will be conducted into finding the different window managers and versions that are supported on the various operating systems. Window managers that will be looked at include KDE, XFCE and GNOME.

Software Installation and Maintenance

Software installed on the selected device will need to be maintained and updated periodically. The aim here is to provide an automated means by which software can be installed based on a preferred selection. To achieve this a number of methods can be used. The first of these options is the idea of allowing the user or system administrator to select their preferred settings (window manager, number of desktop screens, user name and password settings, network and other system configurations) and software applications when installing the operating system. This options would be realized by providing the user a list of options that are based on the various workload groupings (i.e. BA, MIS and CS) that will be hard coded into the operating systems installation setup. Users will therefore be able to select the settings and applications they most preferred using a checkbox and install or activate them. A second option of developing a native application that can download and automatically execute terminal commands that configure the operating system and install preferred software will also be highly considered. A web application option, were the user will have to connect to a server and select various software packages based on their workloads is the third option considered. With this option, the user will have to download an executable file that contains terminal commands based on their selection. These commands when run in terminal will automatically install and configure software on the system. The web application will also provide a platform for support users with trouble shooting issues. Support will be in a form of a chat forum user complains can be posted and viewed by other users who might either report the same problem or provide solutions to the problem.

Chapter Three

In implementing this project, the author decided to opt for the Raspberry Pi model B and Cubieboard 2. This decision was made based on the price tag associated with the boards and the ease of acquiring the boards for the project. This chapter describe the actual steps taken in designing and implements the system. Operating system, customization of software, and configurations and support tool are the three sections in this chapter. Operating system section discuss the process of installing images on the Raspberry Pi and Cubieboard. Customization of software, outlines the steps taken in selecting software for the various user groups. Configuration and support tools discuss the means by which software can be installed and maintained.

Hardware, System Design and Implementation

This section describes the architectural design and process of implementing the proposed low end computing unit for the Ashesi student. The fundamental design and implementation aspects will be based on the categorization of students according to workloads, installing software applications determined through interviews on all operating systems that suite the project, development of an easy configuration tool for students and system administrators, and the provision of support services to aid users in troubleshooting.

Operating System

For the Raspberry Pi NOOBS a pre-installed package that provides options for the first time user of the device to select an operating system of their choice was available for download. For users who are familiar with using the device, specific operating system images are provided for download with installation guides. At the time of this project the available operating system images known to the author were Raspbian, Pidora, Openelec, Raspbmc, Risc OS and Arch Linux [5]. From further testing it was clear that Openelec and Raspbmc were not suitable for the project because they did not support a lot of applications but rather best supported multimedia tasks which are only a minute fraction of student tasks. In selecting the right operating system to use, attempts were made to install all images expect Openelec and Raspbmc on an SD card. Out of the four images that were available and could be used for the project, Pidora the fedora remix for the Raspberry Pi was the only image that booted successfully after installation on an 8 gigabyte SD card. Instructions on how to install Pidora on the Raspberry pi can be found in Appendix C.

The Cubieboard 2 unlike the Raspberry Pi provides more operating system options. As at the time of this project the available operating system images known to the author were Arch Linux, Opensuse, core, Mubox, Lubuntu, Xbmc, Gentoo Linux, Mor, Android, Debian, Freebsd, Fedora, Crux, Openwrt and OSSLab [6]. The author was able to get his hands on Lubuntu for NAND, Lubuntu for SD [7], Fedora 18 and 19 for SD card [7], Android which came pre-installed on the Cubieboard 2 and also available for download, and cubiuntu [7] a variant of the Ubuntu linux distribution for the Cubieboard 2. After tedious installation process, Lubuntu 12.04 for NAND was installed on the 4 gigabyte NAND flash, Cubuntu was installed on a 16 gigabyte SD card and Fedora 19 was installed on a 32 gigabyte SD card. All images downloaded booted successfully on the Cubieboard 2. Instructions on how to flush images to NAND and on to install to SD card can be found in Appendix A.

Customization of Software

In determining the user groups for designing and customizing the software on the device, it was important to take into consideration the different year groups of the university. Based on the year groupings of the university, user groups were categorized into freshmen, sophomores, juniors and seniors. Through an interview session the software needs of the user groups was determined. A total of 30 students were interviewed, 10 from each year group. The questions asked students were "What courses did you take last year", "what tasks associated with a course did you require the use of computers", "For each task, please specify the software application used in accomplishing the task". Interviews were conducted on sophomores, juniors and seniors of the university and answers to the questions were based on their previous year. Choice of software was based on the results of the interviews conducted. Tables showing detail breakdown of how software was selected can be found in Appendix B.

Configuration and Support Tools

Research was conducted into numerous configuration options that could be implemented with the system. Out of the three options explained in the approach the option of developing a web application that runs from a browser was selected. For system administrators the application provides images of the different operating system images for download. Instructions on how to install the images are also provided. The application also provides system administrators and users with a list of software which are categorized based on the different course groupings of the university. From this list individuals can select software and download a shell script file which contains terminal commands for installing software selected. The user will have to execute the shell script for automated installation of software. Using gparted users can back up system files on their SD cards. In case of system failure, users can copy their backed up images and get right back to work.

Chapter Four

This sections is an evaluation of the chosen hardware, operating system and software to see if they meet the requirements of the project. The Hardware Comparison section of this chapter evaluates the hardware of the Raspberry Pi and Cubieboard in comparison with the university's current computing system. Operating System discusses the ease of finding and installing operating systems for the Raspberry Pi and Cubieboard. Software Testing evaluates the different software that are compatible with the Raspberry Pi and Cubieboard.

Testing and Evaluation

Hardware Comparison

Ashesi's computer labs and library are equipment with high end machines manufactured by Lenovo and Hewlett Packard. 25 out of the 53 computers accessible to students are all-in-one Lenovo ThinkCentres. These ThinkCentres run an Intel® Core ™ i7 – 37705 with a central processing unit of 3.10 GHz. They are stocked with 8 gigabytes of random access memory and run on 64 bit Windows 7 professional suites. 10 of the computers in the university's labs are Lenovo ThinkCentre standalones which comprise of a monitor and system unit. These standalone ThinkCentres are powered by an Intel ® Core ™ i5 core – 2400 second generation processors running at processing speeds of 3.10 GHz. 4 gigabytes of random access memory and 460 gigabytes of storage space are packed into the system unit. These computers run on Windows 7 professional 64 bit operating system. Computers provided in the library are all-in-one Lenovo ThinkCentre Edges, powered by Intel ® Core ™ i5 -2400 running at processing speeds of 2.50 GHz. These computers also run Windows 7 professional with 4 gigabytes of random access memory. Starting prices for ThinkCentre all-in-one and standalone computers range between U.S $450 and U.S $1,199 [8].

The Raspberry Pi Model B is fitted with A Broadcom BCM2835 system on chip, with an ARM1176JZF-S core central processing unit running at 700MHz. It has 512 megabytes of random access memory and 8 gigabytes of storage space on SD card. The Pi model B cost 22 pounds sterling [9].

The Cubieboard 2 has an AllWinner A20 system on chip with an ARM Cortex-A7 MPCore running at 1 GHz. It’s fitted with two Mali-400MP GPU and CedarX VPU which are able to decode 2160p quadHD video. The Cubieboard 2 comes installed with on board flash storage of 4 gigabyte which can be expanded through the on board SD card slot. This variation of the cubieboard is priced at U.S $86 [10].

From the above description of the university’s current computing system and the two hardware devices selected for this project, it is very clear that the two selected hardware are no way close to the specifications of the schools current system. However, in comparing the raspberry pi and cubieboard, the cubieboard proves to be far ahead of the raspberry pi in terms of processing power and storage options.

Operating System Evaluation

The operating systems for this project will be evaluated based on the ease to find, level of difficulty in installation process and the resource requirements needed for the operating system to run.

Finding operating systems that run on the raspberry board is very easy since all the known stable operating systems can be downloaded from the Raspberry Pi Foundation website. This ease of finding and downloading images can also be said for the cubieboard two. Cubieboard.org provides links to stable images for download and installation. Although the official websites for these devices provide enough resources to run the devices there are other sites which provide images and documentation [11] [12].

Installing Pidora on the raspberry pi was easy once the right software and steps [13] [14] were followed. Win32DiskImager the software used to flash images onto SD cards is a free open source software which is available for download [15]. Flashing Lubuntu to the Cubieboard 2 NAND was an easy process that was well explained in a video tutorial [16]. Flashing both Lubuntu and Cubiuntu images onto SD cards was also an easy process that was well explained in a video tutorial [17]. Flashing Fedora 19 onto SD card for the cubieboard 2 was a little tedious because of extra steps involved in getting the right image to install. This tedious process was simplified by a video tutorial [18].

Software Testing

Software tested for the Raspberry pi and the Cubieboard was categorized into word processing software, programming software, web browsers and multimedia software. The results of student interviews showed that the most preferred word processing software was Microsoft office suite. Microsoft office suite is only available on the Windows and Mac operating system, and thus cannot run on the Linux distributions selected for the project. It was important to find word processing that provided similar functionality as Microsoft office suite. The search for word processing software revealed that Libre office and Open office were two office suites that could be supported by the Pidora, Lubuntu, Cubiuntu and Fedora. For all four Linux distributions Libre office was the easiest to install. The Libre office suite provided word processing tools that were similar in functionality to Microsoft office Word, Excel, PowerPoint and Access. Though similar these tools didn't provided extensive functionality that Microsoft Office suite tools provided. Open office which was the second office suite considered wasn't an easy install on the pidora. On the Lubuntu and Cubiuntu is was difficult finding open office that could be installed alongside Libre office. Installing Open office required that a clean uninstalling of Libre office was performed. Open office worked well on Lubuntu, Cubiuntu and Fedora. Both Libre office and Open office suites provide very similar functionality and thus any of the two can be used in replacement of Microsoft office suite.

Google chrome and Mozilla Firefox were the two web browsers most preferred by students. Chromium which is similar to Google chrome but light weight in nature cannot be installed on Fedora. This is because no stable variations of the browser have yet been added to the Fedora repositories. Chromium is available for Pidora, Lubuntu and Cuibuntu. Firefox on the otherhand is available for Pidora, Lubuntu, Cubiuntu and Fedora. Running Firefox on Pidora reduces the response time of the Operating system. Opening multiple tabs of 8 and above can freeze the Raspberry Pi for a while. For browsing on the Raspberry pi, Midori the default browser for the Pi is best. Firefox has the same speed reducing effect on Lubuntu, Cubiuntu and Fedora on the cubieboard. Unlike the Raspberry pi the cubieboard doesn't freeze but can have the response time of the operating system drastically reduced, especially if multiple tabs of more than 10 are open. Chromium does have this speed reduction effect on Lubuntu, Cubiuntu and Pidora and thus is most recommend.

Results from student interview showed that for programming in java the most preferred IDE's were Dr.Java, Netbeans and Eclipse. Visual studio was most preferred for C# and visual basic programming. Visual studio runs only on Microsoft Windows and cannot be supported by both the Raspberry pi and Cubieboard because both devices don't support the Windows operating system. For java programming Netbeans IDE proved to be suitable for Lubuntu, Cubiuntu and Fedora. This is because a stable version of Netbeans 7.0.1 was available in the repositories. Stable versions of Dr. Java and Eclipse couldn't be found for Lubuntu, Cubiuntu and Fedora for the Cubieboard Two.

For multimedia, students preferred Vlc player. Vlc player works on all operating systems selected for this project. It is important to note that Pidora, Lubuntu, Cubiuntu and Fedora come pre-installed with multimedia application, but Vlc is a better options for both audio and video because it a better video and audio codex library.

The testing stage of this project has revealed that the Pi works at a very slow processing speed in comparison to the Cubieboard. This is clearly because the Cubieboard's hardware requirement far out weigh that of the Raspberry Pi. Attempts to run Firefox and Libre Office simultaneously totally slowed the Pi and lead to a frozen Desktop Screen. Lags and freezing occurred a number of times when software such as Vlc media player and Firefox browser were being run at the same time. The Cubieboard on the other hand is good at handling parallel computing task. Unlike the Raspberry Pi, Firefox and Libre Office running concurrently with no glitches on the cubieboard. It is much easier to find and install software on the Cubieboard two than on the Raspberry Pi. This is because repositories available for Lubuntu, Fedora and Cubiuntu have more software and plugin support than Piodra.

Chapter Five

Conclusion

In the beginning of this project, the author set out to find a low end, low powered and inexpensive hardware device and operating system which could be customized to suit the computing task of Ashesi University Students, The project also focused on determining software tools supported by the selected operating system that were useful in aiding students accomplish their computing task. For system administrators, the project focused on providing quick automated setup and configuration tools. From research a number of hardware devices were discovered and out of these devices the Raspberry PI Model B and Cubieboard two were acquired for the project. For these two hardware devices, various operating systems were downloaded and tried on them. Out of the few operating systems found, Pidora run on the Raspberry Pi, and Lubuntu, Fedora 19 and Cubiuntu run on the Cubieboard two. An interview was conducted to find out the software used by Ashesi Students in completing computing task. Interview results guided the selection of software that were mostly similar in functionality to software stated by students. Testing and evaluation of hardware, operating system and software clearly reveal that the Cubieboard rather than the Raspberry Pi will be a suitable device for the implementation of a low end, low powered computing center for Ashesi University.

Recommendations and Future Research

Due to time constraints the customization of the user interfaces of Pidora, cubiuntu, Lubuntu and Fedora operating systems couldn't be implements. Customization of the window managers of the various operating systems to suit a particular work group as a field for further work. Other low end hardware devices can be also be used to replicate this project to determine if the Cubieboard two is a better hardware device. Research can also be conducted into the development of images that are built to meet the needs of a specific work group. These images can come pre-installed with the desired software and user interface of the group. The development of repositories that have software specially designed to meet the hardware requirements of low end device should be an area of focus. This project set out to provide an automated software setup and configuration system. Research can be conducted into finding and implementing better setup and configuration tools for the Cubieboard two and other low end devices.

References

[1] N. L. Nelson, Composer, *Phone Interview with Gloria Kwei.* [Sound Recording]. 2014.

[2] O. L. P. Child, "One Laptop Per Child," Pentagram and Upstatement, [Online]. Available: http://one.laptop.org/about/people. [Accessed 10 March 2014].

[3] R. O.-A. Suzanne Fox Buchele, "The One Laptop Per Child (OLPC) Project and Its Applicability to Ghana," Accra, 2007.

[4] J. James, "Low-cost computing and related ways of overcoming the global digital divide," *Journal of Information Science,* vol. 6, no. 27, pp. 385 - 392, 2001.

[5] R. P. Foundation, "raspberry pi.org," Raspberry Pi Foundation, [Online]. Available: http://www.raspberrypi.org/downloads/. [Accessed 10 March 2014].

[6] C. Organization, "cubieboard.org," [Online]. Available: http://docs.cubieboard.org/community#linux_community. [Accessed 17 January 2014].

[7] Cubieboard, "Cubieboard.org," [Online]. Available: http://cubieboard.org/download/. [Accessed 22 1 2014].

[8] Lenovo, "Lenovo Desktops & All-in-Ones," [Online]. Available: http://shop.lenovo.com/us/en/desktops/. [Accessed 11 April 2014].

[9] T. Telegraph, "Mini Raspberry Pi computer goes on sale for £22," The Telegraph, [Online]. Available: http://www.telegraph.co.uk/technology/news/9112841/Mini-Raspberry-Pi-computer-goes-on-sale-for-22.html. [Accessed 11 April 2014].

[10] ebay, "ebay," [Online]. Available: http://stores.ebay.com/cubieboard3. [Accessed 11 4 2014].

[11] elinux, "RPi Distributions," 9 April 2014. [Online]. Available: http://elinux.org/RPi_Distributions. [Accessed 11 April 2014].

[12] linux-sunxi, "Bootable OS images," 29 March 2014. [Online]. Available: http://linux-sunxi.org/Bootable_OS_images. [Accessed 11 April 2014].

[13] elinux, "RPi Easy SD Card Setup," 2 March 2014. [Online]. Available: http://elinux.org/RPi_Easy_SD_Card_Setup. [Accessed 11 April 2014].

[14] J. Browning, "So you got a Raspberry Pi: now what?," Engadget, 4 September 2012. [Online]. Available: http://www.engadget.com/2012/09/04/raspberry-pi-getting-started-guide-how-to/. [Accessed 11 April 2014].

[15] S. forge, "Win32 Disk Imager," Source forge, 2014. [Online]. Available: http://sourceforge.net/projects/win32diskimager/. [Accessed 11 April 2014].

[16] ProgramOften, "Flashing Lubuntu 12.04 to NAND," ProgramOften, 4 August 2013. [Online].

Available: http://www.youtube.com/watch?v=LRDseLtEt30. [Accessed 11 April 2014].

[17] ProgramOften, "Installing Cubiuntu A20," ProgramOften, 12 August 2013. [Online]. Available: http://www.youtube.com/watch?v=AsUgEdBszlM. [Accessed 8 Feburary 2014].

[18] ProgramOften, "Preparing the Secure Digital (SD) Card (Fedora 19)," ProgramOften, 31 July 2013. [Online]. Available: http://www.youtube.com/watch?v=AU9ZvLtXLjg. [Accessed 8 Feburary 2014].

[19] TechWatch, "Arduino vs. Raspberry Pi vs. CubieBoard vs. Gooseberry vs. APC Rock vs. OLinuXino vs. Hackberry A10," TechWatch, [Online]. Available: http://techwatch.keeward.com/geeks-and-nerds/arduino-vs-raspberry-pi-vs-cubieboard-vs-gooseberry-vs-apc-rock-vs-olinuxino-vs-hackberry-a10/. [Accessed 12 March 2014].

[20] E. G. M. K. S. W. T. Suleyman Uludag, "AN AFFORDABLE VIRTUAL LABORATORY INFRASTRUCTURE TO COMPLEMENT A VARIETY OF COMPUTING CLASSES," University of Michigan, Michigan, 2012.

Appendix

Appendix A

Installation of operating Systems

This section describes the steps involved in writing and flushing images onto SD cards and NAND for the Cubieboard 2 and Raspberry Pi.

Installing image on an SD for the Raspberry Pi in Windows

1. Download your image from the Raspberry Pi website or any known trusted site.
2. Download Win32DiskImager application
3. Unzip both image and Win32DiskImager into a file directory of your choice
4. Insert your SD card into your card reader - it is important to take note of the drive letter
5. Double click the Win32DiskImager binary to load the program
6. An error message may pop up when Win32DiskImager loads, it is safe to ignore by clicking ok.
7. On the Win32DiskImager interface click on blue folder icon and select the specific image file.
8. It is very important to ensure that the Device listed is the same drive letter as your SD card
9. Once your image file is selected, click Write and yes to confirm the write.
10. Depending on the speed of your computer, the writing might take some time.
11. When the writing is complete, a message will pop up to indicate write successful. At this point it is safe to eject your SD card and boot it from your Raspberry Pi

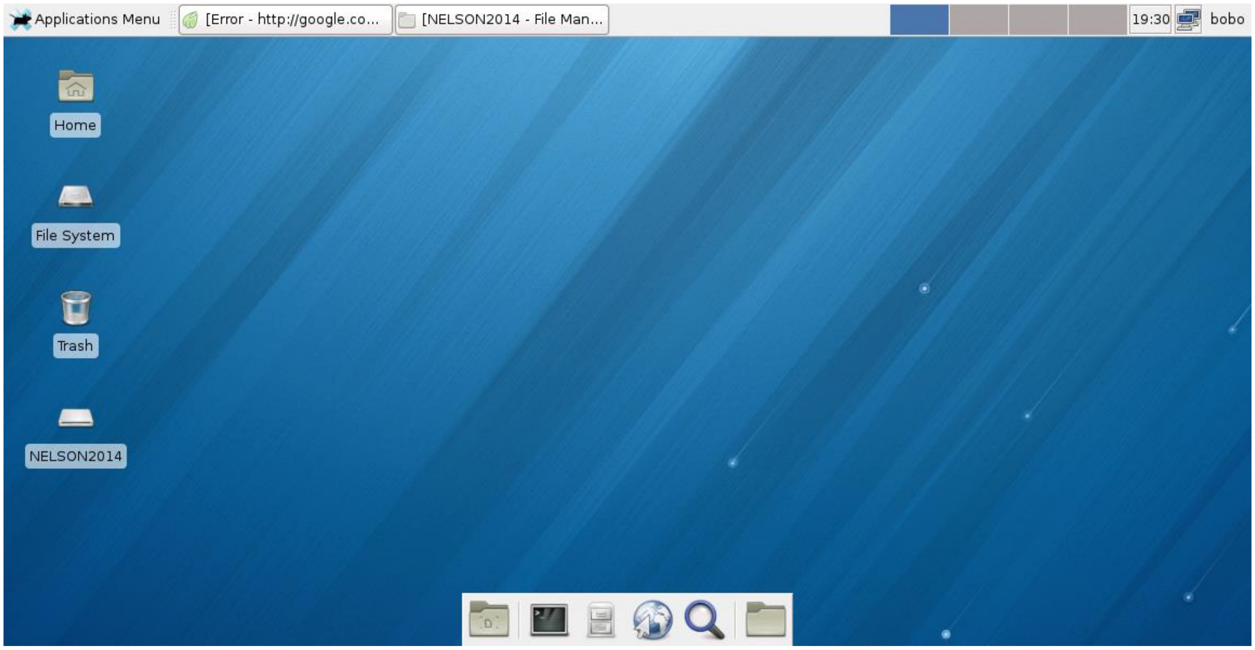


Image of Pidora running on the Raspberry Pi B.

Installing image on an SD for the Raspberry Pi on a MAC or Linux

1. Download your image from the Raspberry Pi website.
2. Extract image from zip folder on the desktop or a directory of your choice (it will be easier installing it from your desktop)
3. On MAC open disk utility to format the SD card as MS DOS FAT, on Linux use gparted to format SD card to FAT32 - it is important to take note of the drive letter.
4. Open terminal to input commands that will install image to SD card
5. Eject SD card from computer
6. Type df-h command and press enter in terminal to display all devices connected to your computer.
7. Insert SD card back into computer, wait for computer to read it and type df-h in terminal again.

8. This will show that a new device has been listed. This device is mostly called disk1s1.
9. Unmount the SD card by typing and running the command: unmounts /dev/disk1s1 (where disk1sl is the name of SD card).
10. In line 3 there was an instruction to note the name of the SD card to help in the process of conversion. Whatever the SD card is named (e.g. /dev/disk1s1), remove the "s1" and add "r" in front of the "disk" so "/dev/disk1" becomes "/dev/rdisk1".
11. In terminal write the image to SD card using the command: sudo dd bs=1m if=~/THE PATH TO THE IMAGE/THE MAME OF THE IMAGE.img of=/dev/YOURDISKNAME". e.g (sudo dd bs=1m if=~/desktop/pidora.img of=/dev/rdisk1 - this is when the image is stored on your desktop and your SD card is called disk1)
12. The installation time depends on the size of your image. A large image will take a few minutes to install.
13. It is important to note that the cursor in terminal not blinking means the image is being writing to the SD card.
14. When writing is complete, eject SD and boot from Raspberry Pi.

Installing image on Cubieboard 2 NAND for windows

1. To get this to work you need usb to micro usb cable
2. Download a live suite program that allows you to flush images into the NAND of the Cubieboard 2. In this tutorial Phoenix is the suite used.

3. Download Cubieboard 2 image and extract to your desktop or a file directory of your choice.
4. Unzip live suite program and install
5. Connect usb to micro usb cable to your pc
6. Now for the trick part. Press and hold the FEL button below the micro usb port and then connect the micro usb part of the cable to the Cubieboard 2. If this is well done the Cubieboard 2 will connect to the pc and its drivers will automatically be installed onto the pc. Disconnect Cubieboard 2 once device drivers are done installing.
7. Once the device drivers are installed open the live suite application (e.g. Phoenix suit)
8. In Phoenix suite click on the firmware button, image button and select the image you want to flush unto the Cubieboard 2 NAND, do not click upgrade
9. Now connect the Cubieboard 2 to the pc using the same technic outlined in step 8 of this tutorial (press and hold FEL button and insert micro usb end). This should be down while phoenix suit is still running on the desktop. If this step is done right a dialog box will pop up asking if a user will like to do a mandatory format. Click yes to write format and write image to NAND
10. Image will be written unto the board. This might take about 5 to 7 minutes. Unplug the Cubieboard 2 from the pc, connect the need peripherals and power the Cubieboard 2.

Image of Cubiuntu running on the Cubieboard 2.

Installing images onto SD card for Cubieboard 2.

1. You need an SD card, an SD card reader to read the micro SD card
2. Plug the micro SD card into the card reader and plug card reader into your pc -it is important to note the drive letter
3. Download Cubieboard 2 image designed for SD card and if zipped, unzip it onto desktop or any file directory of choice.
4. Download Win32DiskImager application, extract and run
5. In Win32DiskImager choice the SD card based on the device letter noted earlier in step two of this tutorial

6. In the Win32DiskImager interface click on blue folder icon and select the specific image file.
7. It is very important to ensure that the Device listed is the same drive letter as your SD card
8. Once your image file is select, click Write and Yes to confirm the write.
9. Depending on the speed of your computer, the writing might take some time.
10. When the writing is complete, a message will pop up to indicate write successful.

Appendix B

1st Year

Software	Total Count
MS Word	15
SPSS	15
Visual Studios	15
MS PowerPoint	12
MS Excel	7
Adobe Reader	4
Prezzi	3
Calculator	3
Offline Dictionary	1
Microsoft Math	1
Math Input Reader	1
R	1
DIA	1
MS Access	1

2nd Year

Software	**Total Count**
MS Word	15
MS Excel	12
MS PowerPoint	12
Adobe Reader	12
Chrome	12
VLC	12
Dr. Java	8
Firefox	6
Eclipse	3
Snip tool	2
SPSS	1
Photo Viewer	1
Foxit Reader	1
Sorting Simulator	1
Corel Draw	1
R	1
Blender	1

3rd Year

Software	Total Count
MS Word	13
MySQL Work Bench	10
MS Excel	9
Xampp	8
Netbeans	8
Notepad++	8
MS PowerPoint	7
Inkscape	6
Balsamiq	6
Indigo Studio	6
Dropbox	6
Dev Cpp	5
Logism	5
phpMyAdmin	5
Eclipse	5
Processing	4
Mars	4
Chrome	4
Photoshop	3
Illustrator	3
Compendium	2
VLC	2
roomeon 3D	1
MS Project	1
Geogebra	1
Codeblocks	1
MS picture manager	1
MS Outlook	1
Prezzi	1

4th Year

Software	Total Count
MS Word	12
Netbeans	9
xampp	9
MS PowerPoint	9
MS Excel	9
Dr. Java	8
mysql workbench	7
notpad++	7
Indigo Studios	6
DevCPP	5
myPhpAdmin	5
Chrome	5
Argo UMLs	4
Adobe Reader	4
Virtual Box	4
DIA	3
dreamweaver	3
pencil	3
Compendium	3
MS Outlook	3
JavaME Mobile Emulator	3
VMWare	3
illustrator	2
komposer	2
JustInMind	2
android ADT	2
Git	2
SPSS	2
LucidChart	1
caret	1
Ripple (chrome store)	1
Live Mail	1
Aptana	1
phpStorms	1
sqlite database browser	1
FileZilla	1
VLC	1
R	1

Appendix C

Arduino Uno

The Arduino Uno is a microcontroller board based on the ATmega328 that has 32 KB of memory. With a maximum length and width of 2.7 and 2.1 inches respectively, the Arduino Uno has 14 digital input/output pins, 6 analog inputs, 16MHz ceramic resonator, USB connection, a power jack, an ICSP header, and a reset button. The device can be powered using a USB connector or with an external power supply. Power can come either from ac AC-to-DC adapter or battery. The board's operating voltage is 5V with a recommended input voltage range of 7-12V. Input voltages limits of the board are estimated to range between 6-20V. Below is a tabulation of the Arduino Uno system specification:

Microcontroller	ATmega328
Operating Voltage	5V
Input Voltage (recommended)	7-12 V
Input Voltage (limits)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	6
DC current per I/O Pin	40mA
DC current for 3.3V Pin	50mA
Flash Memory	32KB (ATmega328) of which 0.5 KB used by bootloader)
SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)
Clock Speed	16 MHZ
Wifi	Optional
HDMI	Not supported
VGA	Not supported
SD card	Optional
Sata	Not supported
USB	1
Video out	Available
Audio out	Via General-purpose input/output (GPIO)

Arduino Due

The Arduino Due is a microcontroller board based on the Atmel Sam3X8E ARM cortex-M3 CPU. It has 54 digital input/output pins; 12 of which can be used as PWM output. It has 12 analog inputs, 4 hardware serial ports, USB connections, a power jack, an SPI header, a JTAG header, a reset button, an 84 MHz clock, 96KB of RAM, and 512KB of memory. The board's operating voltage is 3.3V with a recommended input voltage range of 7-12V. Input voltages limits of the board are estimated to range between 6-16V. Below is a tabulation of the Arduino Due system specification:

Microcontroller	AT91SAM3X8E
Operating Voltage	3.3V
Input Voltage (recommended)	7-12 V
Input Voltage (limits)	6-16V
Digital I/O Pins	54 (of which 12 provide PWM output)
Analog Input Pins	12
Analog Output Pins	2 (DAC)
Total DC Output Current on all I/O lines	130mA
DC current for 3.3V Pin	800mA
DC current for 5V Pin	800mA
Flash Memory	512KB all available for the user applications
SRAM	96KB (two banks: 64KB and 32 KB)
EEPROM	1 KB (ATmega328)
Clock Speed	84 MHZ
Wifi	Optional
HDMI	Not supported
VGA	Not supported
SD card	Optional
Sata	Not supported
USB	1
Video out	Available
Audio out	Via General-purpose input/output (GPIO)

Raspberry Pi

The Raspberry Pi is a tiny and cheap single board computer developed by the UK based Raspberry Pi Foundation. The idea for this credit card sized computer came in 2006, with the aim of promoting the teaching of basic computer science in schools. The Raspberry Pi is manufactured in two board configurations through licensed manufacturing deals with Newark element14, RS Components and Egoman; who produce a version solely distributed in china and Taiwan. The table below shows the specification variation between the two Raspberry Pi models:

	Model A	Model B
System on Chip:	Broadcom BCM2835	
Central Processing Unit:	700 MHz ARM1176JZF-S core (ARM11 family, ARMv6 instruction set)	
Graphic Processing Unit:	Broadcom Video Core IV @ 250 MHz, OpenGL ES 2.0, MPEG -2 and VC-1, 1080p h.264/MPEG-4 AVC high-profile decoder and encoder	
Memory:	256 MB (shared with GPU)	512 MB (share with GPU)
USB 2.0 ports:	1	2
Video Output:	Composite RCA (PAL and NTSC), HDMI, raw LCD panels via DSI	
Onboard network:	None	10/100 Mbit/s Ethernet (8P8C) USB adapter on the third port of USB hub
Low-Level Peripheral:	8 x GPIO, UART, IC bus, SPI bus with two chip selects, IS audio +3.3V, +5V, ground	
Power Ratings:	300 mA (1.5 W)	700 mA (3.5 W)
Power Source:	5 V via MicroUSB or GPIO header	
Size:	85.60 mm x 53.98 mm (3.370 in x 2.125 in)	
Weight:	45 g (1.6 oz)	
Operating Systems:	Arch Linux ARM, Debian GNU/Linux, Gentoo, Pidora(Fedora Remix), FreeBSD, NetBSD, OpenELEC, Raspbian OS, RISC OS, Slackware Linux, Android	
Audio outputs	3.5mm jack, HDMI	
Memory card support	SD, MMC	

Cubieboard

The Cubieboard is a single-board computer, made in Shenzhen, Guangdong, China. It uses the AllWinner A10 and A20 system on chip (SoC), popular on cheap tablets, phones and media PCs. The first prototype boards were sold internationally in September 2012. In June 2013 an enhanced board with an AllWinner A20 was released. The table below shows the different models of the Cubieboard:

	Cubieboard	Cubieboard 2	Cubieboard truck
System on Chip	AllWinner A10 SoC (ARM Cortex-A8 @ 1 GHz CPU, with Mali-400MP GPU and CedarX VPU able to decode 2160p quadHD video).	AllWinner A20 SoC (2 ARM Cortex-A7 MPCore @ 1 GHz CPU, with 2 Mali-400MP GPU and CedarX VPU able to decode 2160p quadHD video).	AllWinner A20 SoC (dual-core ARM Cortex-A7 @ 1 GHz CPU, with Mali-400MP2 GPU).
Memory	512 MB (beta) or 1GB (final) DDR3	512 MB (beta) or 1GB (final) DDR3	2 GB DDR3 @ 480 MHz
Storage	4 GB NAND flash built-in, 1x microSD slot, 1x SATA port	4 GB NAND flash built-in, 1x microSD slot, 1x SATA port	8 GB NAND flash built-in, 1x microSD slot, 1x SATA 2.0 port
Video outputs	HDMI 1080p output	HDMI 1080p output	HDMI 1080p outpu
Networking	10/100 Ethernet connector	10/100 Ethernet connector	10/100/1000 RTL8211E Ethernet connector
Connections	2x USB Host, 1x USB OTG, 1x CIR.	2x USB Host, 1x USB OTG, 1x CIR.	2x USB Host, 1x USB OTG, 1x CIR.
Extra Connections	96 extend pin including I²C, SPI, LVDS	96 extend pin including I²C, SPI, LVDS	54 extended pins including I²C, SPI
Audio	3.5mm	3.5mm	S/PDIF, headphone and HDMI audio out, mic and line-in via extended pins
Dimensions	10 cm × 6 cm	10 cm × 6 cm	11 cm × 8 cm

Cubieboard Two.

Gooseberry

The Gooseberry board is an A10 Allwinner system on chip board produced in the United Kingdom. It has a 1 Ghz CPU which can be overclocked to 1.5 Ghz, and uses the Mali 400 MHz graphics processor. It has an onboard storage of 4 gigabytes and is upgradeable by micro SD memory card of up to 32 gigabytes. The board supports an Ac jack, it has a 3.5 mm earphone jack, a mini USB to connect USB peripheral (mouse, keyboard, USB hub) and to connect the board to a computer for flashing the on board NAND flash, HDMI out to connectthe board to a television or monitor and a micro SD slot for storage expansion. Android is the only fully compatible Operating system for the Gooseberry with a possibility of running Ubuntu with advanced technical Knowledge.

APC Rock, Paper & 8750

The APC Rock, paper and 8750 are chip set boards designed and developed by APC with the aim of providing CPU's that are less expensive, consumer little power and easily connect to the internet. The table below shows the specification variation between the different board models:

Model	Paper	Rock	8750
Chip	VIA ARM Cortex-A9 @800Mhz Processor	VIA ARM Cortex-A9 @800Mhz Processor	VIA ARM 800MHz Processor
Memory	DDR3 512MB Memory 4GB NAND Flash	DDR3 512MB Memory 4GB NAND Flash	DDR3 512MB Memory 2GB NAND Flash
Graphics	Built-in 2D/3D Graphic Resolution up to 720p	Built-in 2D/3D Graphic Resolution up to 720p	Built-in 2D/3D Graphic Resolution up to 720p
Input and Output	HDMI USB 2.0 (x2) microUSB (OTG) Audio out/ Mic in microSD Slot	HDMI ,VGA, USB 2.0 (x2) microUSB (OTG) Audio out/ Mic in microSD Slot	HDMI, VGA, USB 2.0 (x4) Audio out/ Mic in microSD Slot
Network	10/100 Ethernet	10/100 Ethernet	10/100 Ethernet
Debug	20 – pin ARM – JTAG header	20 – pin ARM – JTAG header	None
Expansion	Extra GPIO, SPI and I2C busses on a header	Extra GPIO, SPI and I2C busses on a header	None
Size	204 x 98 x 28mm (W x H x D)	170 x 85mm (W x H) Neo – ITX Standard	170 x 85mm (W x H) Neo – ITX Standard
Software	Firefox OS	Firefox OS	Android

A13 OlinuXino wifi

The Table below outlines the specifications of the OlinuXino A13 wifi board.

CPU	A13 Cortex A8 processor at 1GHz
GPU	3D Mali400 GPU
RAM	512MB
Power	6-16VDC input power supply, noise immune design
USB/USB host	3 + 1 USB Host, 3 available for users 1 for (optional) WIFI RTL8188CU 802.11n 150Mbit module on board
Power connector	1 USB OTG which can power the board
SD Slot	SD-card connector for booting the Linux image
	(optional) 4GB NAND flash
Video output	HDMI,VGA video output – 800 x 600 resolution
Audio Output	Available
Microphone input	Available
Memory	4GB NAND flash
Max Memory	32GB via SD

A10 OlinuXino

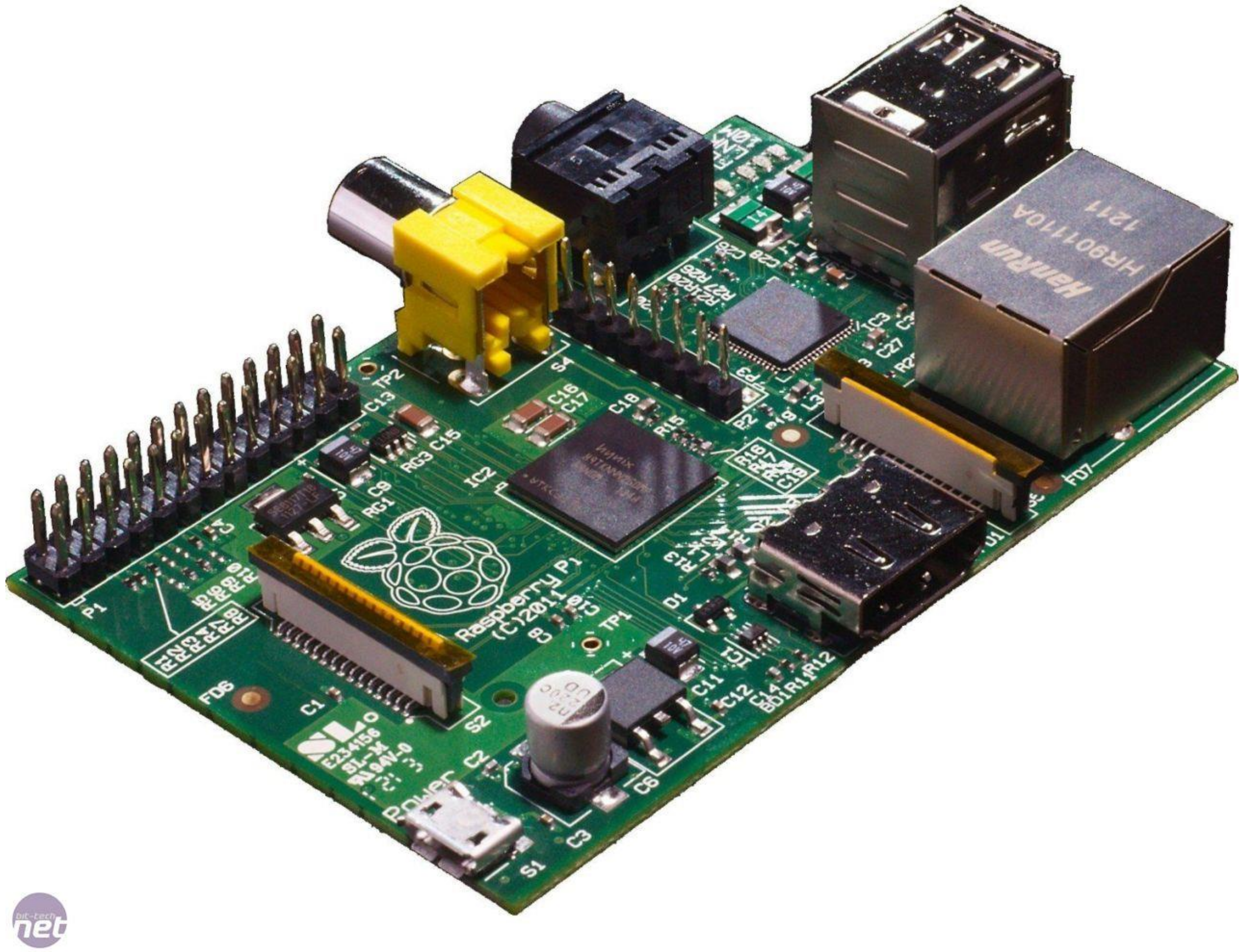
The table below outlines the specifications of the OlinuXino A10 board.

Processor	A10 Allwinner ARM Cortex-A8
Clock Speed	1GHz
GPU	ARM Mali 400
RAM	1GB
Memory	4GB NAND flash
Max Memory	32GB via SD
USB	2
USB host	1
Ethrnet	100Mbit
GPIO	200 GPIO on 0.05” connrctors

Hackberry A10

The table below outlines the specifications of the Hackberry A10 board.

CPU	1.0GHz Allwinner A10 ARM Cortex A8
GPU	Mali 400 with hardware 3D acceleration and hardware video decoding
Serial port	3.3v TTL 4-pin header
Audio input	3.5mm microphone jack
Audio output	Audio over HDMI
USB	2 x USB A 2.0 ports
Internal Storage	4GB NAND storage, 1.5GB available in user partition in Android
External Storage	SDHC card slot supporting up to 32GB
Networking	10/100 Ethernet, Realtek 802.11n WiFi
Memory	DDR3 512MB / 1GB, - 100MB is reserved for the GPU
Boot	Boot from SD card and internal storage via u-boot
OS	Andriod 4.O ICS, Linux support
Digital video output	HDMI up to 1080p
Analog video output	3.5mm composite AV, 3.5mm component Y/Pb/Pr
Power	NEMA 2-pin power adapter included Input AC100 – 240V – 0.4A 50/60Hz Output DC5v



Raspberry Pi