

Club Information

Organization Information

UCLA Linux Users Group is an organization aimed at the promotion of Free and Open Source Software, and associated operating systems, such as Linux. The Linux Users Group specializes in basic system administration, coding in myriad programming languages, as well as hardware and software debugging. LUG provides walk-in IT consultancy for open source systems, taking care to inform and educate clients. The LUG augments this by hosting a series of events and seminars to further educate the general public about free and open source software.

Organization Resources

UCLA Linux Users Group is one of the few student organizations on campus that performs its DNS under the ucla.edu banner [linux.ucla.edu]. The LUG maintains an on campus lounge, and provides its members with workstations, as well as a server cluster. The LUG currently has over 50 members, with 20 core users.

Organization Objective

Upon completion of its first project, namely the Interactive Technology Upgrade, UCLA Linux Users Group wishes to conduct a postmortem of the first phase of this project. This document serves as an analysis of the final product, as well as the steps taken to achieve the implementation. This document will also highlight possible enhancements and upgrades that may be applied to the implementation in the future. Finally, the document will touch upon UCLA LUG's responsibilities regarding the current implementation.

Lead Document Authors

Aamoy Gupta
LUG Administrator
KIHC Lead Software Developer
408-768-2645
aamoyg@linux.ucla.edu

Matt Millar
LUG Member
KIHC Software Developer
831-331-3838
millar@ucla.edu

Steps Taken to Achieve Implementation

Selection of a Mainboard

The first step of the project was to select an appropriate system-on-a-chip microcomputer for the project. The aim here was to pick the board which would offer adequate performance, as well as maximum upgradability. In this case, the Beagleboard xM Revision C was chosen because of its relatively standard and fast processor, HD capable GPU, and RAM size. The following is a table detailing the Beagleboard xM's specifications.

System Component	Specification
RAM	512 MB LPDDR -type
Integrated Graphics	ARM Cortex A8 Neon
Discrete Graphics	C64x+ DSP Core [HD720P]
Processor	ARM Cortex A8 @ 1 GHz [overclockable]
Connectors and I/O	Ethernet 10/100 4 USB 2.0 HDMI S-Video Mini-USB [for power] Standard Voltage Input JTAG Connector RS-232 Serial Port Micro-SD card slot
Expandability	Camera Expansion Port LCD Screen Expansion Port SD Card Expansion

The board was available for \$270, including a case, HDMI cable, and power adapter. The Raspberry Pi and a VIA-based thin client computer were also under due consideration for the role. The Raspberry Pi, though inexpensive, was not selected as its central processing unit is too slow. Thin clients, on the other hand, are powerful yet expensive. As a result, they would be deemed overkill for this project. The BeagleBoard xM could be called the intermediate between the two.

Selection of an Operating System

The next step was to select an operating system for the aforementioned mainboard. There were several options available, outlined below. Also mentioned are the pros and cons of each operating system.

Operating System	Pros	Cons
Angstrom	<ul style="list-style-type: none"> ✓ Bundled with BeagleBoard XM ✓ Mobile operating system with touchscreen, camera, and voice recognition capability 	<ul style="list-style-type: none"> ✓ Shipped with outdated kernel ✓ Virtually no community support ✓ No basic functionality (man pages), Ethernet drivers, graphics drivers have to be manually started
BeagleBoardUbuntu	<ul style="list-style-type: none"> ✓ Full desktop operating system (complete with man pages and useful utilities) ✓ Software support ✓ Full range of packages (including drivers) 	<ul style="list-style-type: none"> ✓ Very difficult to set up. ✓ Some driver conflicts and problems.
Debian CTO	<ul style="list-style-type: none"> ✓ Stable, runs well at kernel level ✓ Can function well as a full or mobile desktop environment 	<ul style="list-style-type: none"> ✓ No support (just a port) ✓ Packages need to be compiled from source (drivers, programs, everything) ✓ A pain to configure and set up.

The operating system chosen in this case was BeagleBoardUbuntu. The main reason for this choice was the support offered by Canonical, as well as the fact that it was easier to maintain and upgrade. Though BeagleBoardUbuntu does not feature the built-in touchscreen and voice recognition libraries like the Angstrom Distribution, it is confirmed that even more advanced solutions can be easily obtained and installed. BeagleBoardUbuntu also has the distinction of being a full desktop operating system.

Content Display Manager

A method had to be devised to display content, such as text, photographs, and video. The following options were considered.

- ❖ Pre-rendered video
 - This would take text, photographs, video, and animations, and render them all together as a video. We dropped this option due to the unstable nature of the dedicated graphics drivers. Without the driver's activated, the board could only manage standard definition video played back on weak integrated graphics. This also does not allow for dynamic insertion of content without a re-render of the entire video. This approach was

tested with the “Trouble with Tribbles” Star Trek TOS episode. Video in HD720P had erratic playback, while standard definition video was able to play unfettered. However, use of standard definition would look unattractive, considering the fact that a 1080P monitor is being used as the display.

- ❖ PDF Slideshow
 - This approach entails the use of PowerPoint-type slides rendered to the lightweight PDF format. The limitation is that media is only limited to text and video. Since PDFs can be easily rendered, dynamic insertion of content is not a problem. This approach has already been tested with a series of heavy PDFs, displaying high definition Star Wars concept art for a week straight.
- ❖ HTML5/JavaScript Based Presentation
 - This was the preferred approach (though it requires the most work). A content management system would be developed using open source JavaScript framework for presentations. The framework would support all sorts of media, including embedded standard definition video and Flash objects. The system is fully dynamic-- new slide objects can be inserted at will [though creating the slides themselves is time consuming without a template]. We have currently developed a slideshow system with banners, pictures, and informational text. Provisions for video and Flash media will be made as part of an update cycle.

Steps Taken to Overcome Limitations

Troubleshooting of Hardware Issues

The board, while formidable on paper, had a series of hardware issues associated with it. The following section outlines the encountered difficulties, and the steps we took in solving them over the course of implementation of the solution.

HDMI Overscan Issue

One of the first problems we encountered was with the display. The top and bottom of the screen were cropped for an unknown reason, which we presumed to be the Beagleboard xM. We tried to correct this by editing the X Window System display parameters, as well as boot parameters of the board. However, none of these attempted fixes worked. Eventually we realized that the problem was due to overscan [we were using a plasma TV as the display]. While discrete graphics cards on laptops account for overscan, the nascent drivers on the Beagleboard xM do not have this feature. We discovered that this overscan issue only held true for televisions and television like displays. We corrected the problem through the use of an LCD monitor as a display instead of a television.

Memory Card Limitations

Another hardware related problem had to do with the memory card. The card bundled with the Beagleboard xM was of the outdated Class 4 type, entailing a severely limited read and write speed. We corrected this issue by upgrading to Class 10 standard, and immediately noticed a large increase in terms of performance, specifically I/O times.

Memory Card Adapter

A design flaw (or perhaps an inconvenience) of the Beagleboard xM is its loose Micro-SD card port. On more than one occasion, the Micro-SD card, which holds the operating system, has simply fallen out while the board was in operation. The only solution to this issue was a complete format, reinstallation, and reconfiguration of the entire operating system. Seeing as the solution itself required over 4 hours to carry out, we felt it necessary to rectify the problem as quickly as possible. We accomplished this by obtaining a case that would surround the Beagleboard xM, minimizing this issue.

Overheating

The board tended to slow down after periods of extended operation due to overheating. We corrected this issue by installing the case, which contains a heat-sink.

Micro-USB Power Limitations

While this is not exactly a problem, it was necessary to purchase a dedicated power supply so that the CPU would not operate in low power mode, and scale itself to a lower frequency.

Troubleshooting of Software Issues

The following is a list of software related issues encountered while trying to develop a solution. Since multiple operating systems were tried and tested, problems exclusive to an operating system are listed under its specific heading.

Angstrom Operating System

The Angstrom Linux distribution is considered the standard operating system for the Beagleboard xM. It can be custom tailored to needs of the developer. We customized an Angstrom image for use with the project to include all necessary drivers, development tools, and extensions for upgrades [most notably libraries for OpenCV, touchscreen capability, and speech recognition]. The following are the issues experienced with the Angstrom Operating System:

- ❖ *Lack of MAN pages*
 - This inhibited development of scripts and usage of critical system utilities. Most programmers are used to having MAN pages for

quick reference. Though this may not inhibit the final product, it prolongs development, and makes attainment of the objective within reasonable amount of time a far cry.

- ❖ *Lack of basic system utilities*
 - System utilities like patch, sudo, and dd were not available to use. This required us to use more primitive utilities to accomplish the task, or perform the task on a different machine altogether. Since many programs and drivers not offered on the Angstrom distribution repositories had to be compiled from source, for the ARM-8 core, this posed a serious hurdle which could not be easily surpassed.
- ❖ *Lack of critical software packages and packages management*
 - Despite being an offshoot of the popular Ubuntu Linux desktop distribution, Angstrom lacked basic packages and horrible package management. In fact, VIM (a standard on most Linux distributions) was not available, and had to be compiled from source. The package manager did not track installed packages properly, and could not handle dependencies, or conduct repairs.
- ❖ *Outdated Kernel*
 - The kernel utilized in the operating system was outdated (Version 2, while the standard is Version 3). This created problems— for example, the network driver listed the Ethernet port on the mainboard as one of the USB ports, among other things.
- ❖ *Broken and Unavailable drivers*
 - One of the claimed advantages of the Angstrom distribution was that it had the ability to be bundled with drivers exclusive to the Beagleboard xM. Unfortunately, this was not the case. Though the drivers were bundled, they were bundled as uncompiled kernel objects. Attempts to compile the drivers resulted in numerous errors. There were cross dependencies between so many of said kernel objects, that activation of the various drivers was impossible. A great deal of effort was spent trying to make the Ethernet driver functional, which was for some reason registering as a USB port. Several scripts were developed to allow the network connection to actually work.

BeagleBoardUbuntu

After our myriad unwanted adventures with the Angstrom distribution, we decided to try BeagleBoardUbuntu. This distribution is a fork of the popular desktop Ubuntu distribution. Many internet sources discouraged BeagleBoard xM users from installing this distribution due to compatibility issues with drivers. However, we decided to try it anyway. We found that the operating

system had much better [and more active] support than that of the Angstrom distribution. We had also found that Canonical and Texas Instruments were offering support as well. The following are the issues that we experienced with the BeagleBoardUbuntu operating system.

❖ *Difficult to set up*

- BeagleBoardUbuntu did not ship with drivers like Angstrom. In order to make use of the more powerful hardware on the BeagleBoard xM, such as the Neon core extensions, as well as the DSPx+ dedicated graphics, it was necessary to obtain drivers and updated kernels from third party vendors and developers. The aforementioned objects then had to be compiled on the board. Thankfully, since BeagleBoardUbuntu is a desktop operating system with a full suite of development tools, this was not an impossible task.

Other Issues

Listed here are software issues that were not exclusive to a particular operating system, but were pervasive throughout the development of the solution.

❖ *MAC Address Issue*

- The BeagleBoard xM does not have a hardware MAC address. Instead, it randomly generates a MAC address upon initialization of the network process. The first attempt to rectify this issue was an attempt to manually set the MAC address variable to a static one. However, this solution did not work, because the network interface had to be taken down and restarted for a manual set. Upon restart, networking would generate an entirely new MAC address. We were finally able to solve the problem by spoofing the MAC address while the network service was running.

❖ *Available Browsers*

- Though the BeagleBoard xM is a powerful platform, it lacks in efficient software tailored to its needs. This especially rang true in terms of its browser, which was necessary for the content display manager, which required and relied upon a modern browser. Unfortunately, no modern browsers are readily available for the BeagleBoard xM. In order to solve this issue, we compiled the resource heavy, yet high performance Chromium browser for the mainboard (this verified version of Chromium is itself, out of date and not considered modern). Coupled with the hardware improvements, this offered a

significant decrease in loading times, making for a viewable platform.

Steps That May Be Taken in the Future

UCLA LUG Responsibilities for Current Implementation

Upon completion of the infographic display, LUG's research division has inherited responsibilities regarding upkeep and further development. The following tasks will continue to be implemented:

- ❖ Dynamic insertion and removal of content [in collaboration with KIHC]
- ❖ Management of critical system software updates
- ❖ Development of extensive documentation
- ❖ Assessment and development of in-house upgrades [if permission is granted to do so]
- ❖ Research for other uses of the developed platform [if permission is granted to do so]

Possible Updates

Depending on the system configuration and the project budget, a variety of updates and upgrades can be implemented to improve system experience. We have assessed that the following upgrades [which were also proposed earlier] are highly feasible and compatible with the current hardware and implementation. All updates would be developed and tested in parallel [the current infographic display would not be affected]. Should testing prove that the upgrade works, then it will be incorporated alongside the current display.

Upgrade	Features	Requirements	Estimated Cost
Remote Access	The system will have a remote connection and control. This allows for direct control of the display, as well as the ability to push updates to the system.	<ul style="list-style-type: none"> ✓ Networking infrastructure [this includes, but is not limited to cabling or WLAN] 	~\$20
Touchscreen	The system will use a touchscreen enabled system to achieve a greater degree of interaction with users. Users will have the ability to control the display through gestures, and thus be able to access	<ul style="list-style-type: none"> ✓ Touchscreen monitor with feedback over USB or HDMI. ✓ Appropriate chipset 	Touchscreen layer on existing monitor: ~\$300 New Touchscreen monitor:

	content that they choose to.		~\$800 and up
Speech Synthesis	This addition will allow the system to provide automatic tours to the users through the use of speech synthesis technologies. The system could be further augmented by hardware that detects users near exhibits.	<ul style="list-style-type: none"> ✓ Speaker network ✓ Speech synthesis library ✓ IR Beams / Motion detectors [optional] 	~\$50 and up Optional add-ons: ~\$100 and up
Speech Processing	Designed to work in tandem with Speech Synthesis, the speech processing module will give users the ability to interact naturally with the system.	<ul style="list-style-type: none"> ✓ Speaker system ✓ Shotgun microphone network ✓ Speech processing library ✓ IR Beams / Motion detectors [optional] 	~\$30

Possible Alternative Applications

As a research oriented organization, it is our belief that the infographic display [along with its various upgrades] could serve multiple purposes. A potential mapping application is being investigated at the moment. Providing a simple service to students as well as conducting research such as this serves to boost KIHC’s reputation at UCLA.

Thank You

Thank you for giving Linux User’s Group the opportunity to conduct research and develop an effective product. We hope that the developed solution is to your satisfaction. We look forward to working with KIHC again to create new systems and maintain existing ones.



KIHC

The Kleinrock Internet History Center at UCLA