

Beaglebone Webcam Server



By

Alexander Corcoran

California Polytechnic State University, San Luis Obispo

Date Submitted: June 10th, 2012

Advisor: Dr. Hugh Smith

1 Abstract

The Beaglebone Webcam Server is a Linux based IP webcam, based on an inexpensive ARM development board, which hosts its own web server to display the webcam feed. The server has the ability to either connect to a wired router, or to act as a wireless access point in order for users to connect and control its functions via any Wi-Fi enabled device.

2 Table of Contents

1	Abstract.....	2
2	Table of Contents	2
3	List of Tables and Figures	3
4	Introduction.....	4
4.1	Overview	4
4.2	Feature Comparisons.....	5
4.3	System Setup and Use	6
5	Background.....	7
5.1	Hardware	7
5.1.1	Beaglebone Development Board.....	7
5.1.2	Logitech C110 USB Webcam	8
5.1.3	Alfa AWUS036NHA USB Wireless Adaptor.....	8
5.1.4	Belkin F4U040V 4-Port USB 2.0 Powered Hub	9
5.1.5	SparkFun Pan/Tilt Bracket and Small Servo.....	9
5.1.6	SparkFun Opto-isolator Breakout.....	10
5.1.7	Cost of Hardware Materials.....	11
5.2	Software.....	11
5.2.1	BeagleBoard Ubuntu	12
5.2.2	Linux UVC driver	12
5.2.3	Ath9k_htc driver	13
5.2.4	Hostapd	13
5.2.5	Isc-Dhcp-Server	13
5.2.6	Lighttpd	13
5.2.7	Mjpg-streamer.....	14
5.2.8	ffmpeg.....	14
5.2.9	Flowplayer.....	14
5.2.10	Servo Control Script	15
5.2.11	Daemon Startup Scripts	15
5.2.12	PHP, JavaScript, and AJAX.....	15
6	Development	17
6.1	Filesystem	17
6.2	Distribution Selection.....	17

6.3	Live Video Streaming: MJPG-streamer.....	18
6.4	Saving and Playback of the Live Stream.....	18
6.5	Configuring Lighttpd.....	19
6.6	Servo Control: GPIO vs. PWM.....	19
6.7	Changing Settings and Executing Processes with PHP	20
7	Appendix: Code.....	21
7.1	/etc/dhcp/dhcp.conf.....	21
7.2	/etc/hostapd/hostapd.conf	21
7.3	/etc/init/mjpg_streamer.conf	21
7.4	/etc/lighttd/lighttpd.conf.....	22
7.5	/etc/rc.local	22
7.6	/var/www/about.php.....	23
7.7	/var/www/cgi-bin/date.php	24
7.8	/var/www/cgi-bin/password.php.....	24
7.9	/var/www/cgi-bin/savevid.php.....	24
7.10	/var/www/cgi-bin/xaxis.php	25
7.11	/var/www/cgi-bin/yaxis.php	25
7.12	/var/www/includes/footer.php.....	25
7.13	/var/www/includes/header.php	26
7.14	/var/www/includes/style.css.....	26
7.15	/var/www/index.php.....	27
7.16	/var/www/ rename.php	27
7.17	/var/www/settings.php	28
7.18	/var/www/stream.php.....	30
7.19	/var/www/videos.php	33
8	Appendix: User Interface.....	34
8.1	Home	34
8.2	Stream	35
8.3	Videos	35
8.4	Settings.....	36
8.5	About	36

3 List of Tables and Figures

Figure 1: Hardware Diagram.....	4
Figure 2: Power Diagram.....	7
Figure 3: Bracket and Webcam	10
Figure 4: Software Flow	12
Table 1: Cost of Materials.....	11

4 Introduction

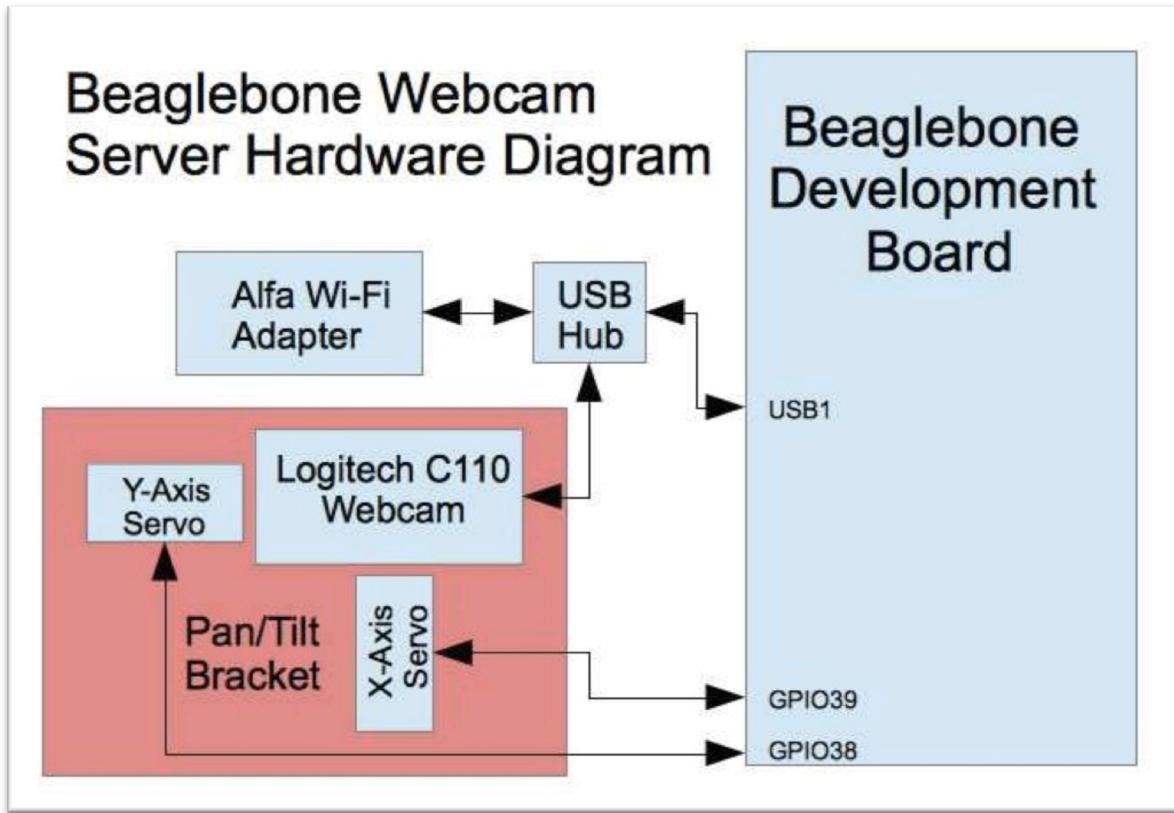


Figure 1: Hardware Diagram

4.1 Overview

The goals of the Beaglebone Webcam Server project are to design and implement an IP webcam using open source software. The project builds upon the rich tools and libraries provided by the Linux community in order to implement features such as: viewing a live webcam stream via a web browser; saving the stream to the server for later viewing, or to download to a client; and controlling two-axis rotation of the webcam via the browser interface. The project focuses on using mature and stable base software components in order to provide a robust client facing user experience.

The web interface is written with JavaScript and PHP, with certain functional sections implemented with AJAX calls to server scripts that manipulate configuration files, or run programs on the server.

4.2 Feature Comparisons

The wireless IP camera market evolved from the wired closed circuit television (CCTV) surveillance camera market, with many of the major networking companies putting out products that focus on using the IP protocol as a medium to transfer video, as opposed to wired CCTV technologies.¹ The Beaglebone has similar features to many IP camera offerings, including pan/tilt, configuration via web interface, and viewing the stream over the Internet, and saving of content², but unique to the Beaglebone Webcam Server is the ability to act as a stand alone access point, and selectively save videos to the system. The cost of the prototype system is also comparable to purchasing a complete system, which means with a streamlined production process, the costs could be reduced below the price of other competitors in the market.

¹ "History of IP Camera Technology." , *Learn the Beginnings of IP Camera Systems*. CCTVSystems.com, 2011. Web. 17 May 2012. <<http://www.cctvsystems.com/history-of-ip>>.

² Selection of similar products in the market:
"IP Camera Features." 1-1. *Cisco Video Surveillance IP Camera User Guide*. Cisco Systems, Inc., 2008. Web. 17 May 2012. <http://www.cisco.com/en/US/docs/security/physical_security/video_surveillance/ip_camera/2500/english/user_guide/ipug.pdf>.

"BL-C230A Tech Specifications." *Panasonic Product Support*. Panasonic, Inc. Web. 17 May 2012. <<http://shop.panasonic.com/shop/model/BL-C230A?support>>.

"DCS-5300." *D-Link Fast Ethernet Network Camera*. D-Link, Inc., 2003. Web. 17 May 2012. <ftp://ftp10.dlink.com/pdfs/products/DCS-5300/DCS-5300_ds.pdf>.

1080p Wireless Network Video Recorder." *NVR Pros HD NVR*. NVR Pros. Web. 17 May 2012. <<http://www.nvrpros.com/HDNVR.html>>.

4.3 System Setup and Use

There are two modes of operation for the system: wired mode and wireless mode. In wired mode, the user needs to disconnect the USB Wi-Fi adaptor and plug the system into a router. In this configuration, the system can be accessed via the Internet for remote motoring of the webcam. For wireless access point mode, the user should disconnect the wired cable, and connect the wireless USB module. Once powered on in this configuration, the system will broadcast an SSID that the user can connect to via a Wi-Fi enabled computer or smart-phone.

In either configuration, to access the web page being hosted on the system, the user must flip the power switch located on the side of the case and wait for the system to power on. Once there is a green light on the webcam, as well as a blue light on the Wi-Fi adapter if being used in wireless mode, the user can connect to the network, navigate to the web address <http://192.168.1.164>, and log in with the user name and password. On the website there are various pages that allow access to the different functions of the server.³

- Home: A small general description of the project, as well as basic descriptions of the functionality of different pages.
- Stream: View a live stream of the webcam, move the webcam along the X and Y-axes, and save a live stream.
- Videos: View a list of videos saved onto the server; as well as play, download or delete any saved video.
- Settings: Change some basic settings of the system including: changing the login user and password, rename a video hosted on the server, and changing the date and time of the system.

³ see Section 8: User Interface for screen shots of the interface

- About: Provides detailed information of how to use the system, an instruction manual.

5 Background

The Beaglebone Webcam Server project required both hardware and software development and component selection.

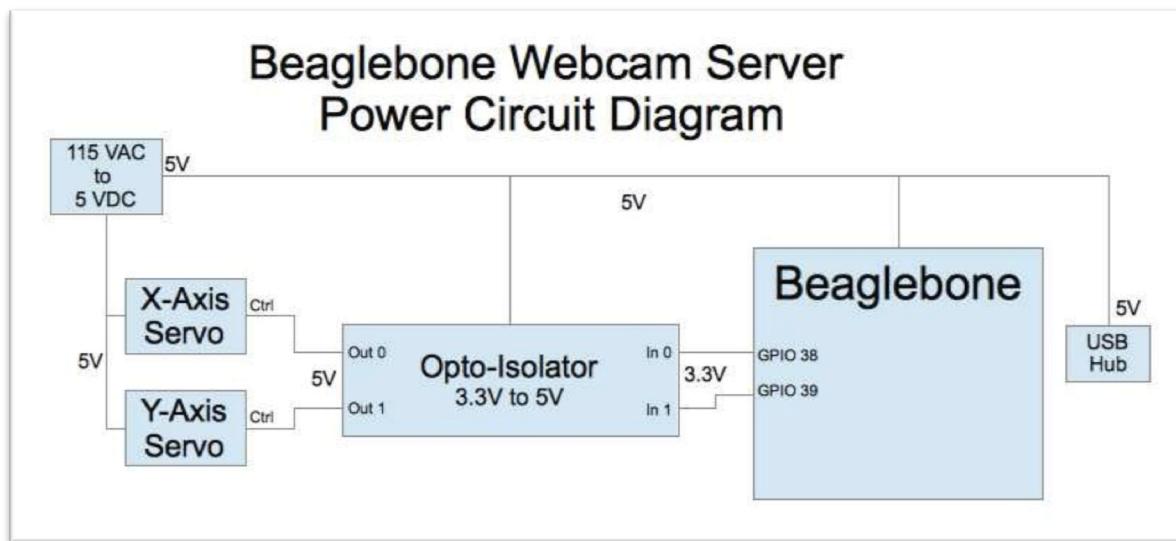


Figure 2: Power Diagram

5.1 Hardware

The entire system runs off of a 5V DC adaptor that provides power to the Beaglebone, USB hub, and servomotors. The Optical Isolator converts the 3.3V GPIO signal to 5V in order to trigger the servos.

5.1.1 Beaglebone Development Board

The Beaglebone ARM development board was selected for the project because of its inexpensive price, and fairly large active user base. There is a great deal of discussion and support on the Internet, and given the open source nature of the hardware, there is complete

documentation and schematics for the system provided by the designers.⁴ The physical board is very small; at 3.4" by 2.1" it can be mounted in a small enclosure with all of the other components. It features a Texas Instruments AM3359 processor that runs at 700MHz, when connected to a 5V power adaptor, as well as 256MB of DDR2 RAM.⁵ The expansion port capability used for this project include GPIO ports to operate the servo motors, and the USB interface in order to communicate between the two peripheral devices and the system.

5.1.2 Logitech C110 USB Webcam

The Logitech webcam features VGA video capture, a USB interface, and a small, compact form factor.⁶ The main feature to be considered in selecting a webcam for this project is compatibility with the UVC (USB Video Class) Linux driver.⁷ Most drivers for Linux need to be written by the open source community because of lack of vendor support, so selecting a support device with a mature driver is essential.⁸

An added benefit of this webcam is its small size and weight, thus allowing it to be mounted to the pan/tilt bracket, and moved by the servomotors.

5.1.3 Alfa AWUS036NHA USB Wireless Adaptor

The Alfa wireless adapter is based on the Atheros AR9271 chipset.⁹ This specific chipset supports Monitor Mode / Access Point mode in the ath9k_htc driver.¹⁰ The Atheros

⁴ <http://beagleboard.org/hardware/design/>

⁵ Coley, Gerald. "BeagleBone Design Specification." *BeagleBone Rev A5 System Reference Manual*. 20. 2 Feb. 2012. Web. 16 May 2012. <http://beagleboard.org/static/BONESRM_latest.pdf>.

⁶ "Logitech Webcam C110." *Webcam C110*. Logitech, Inc. Web. 17 May 2012. <<http://www.logitech.com/en-us/webcam-communications/webcams/devices/8112>>.

⁷ "Supported Devices." *Linux UVC Driver and Tools*. Web. 17 May 2012. <<http://www.ideasonboard.org/uvc/#devices>>.

⁸ "A Linux Driver Project Status Report." *LWN.net*. The Linux Driver Project, 8 Apr. 2008. Web. 17 May 2012. <<http://lwn.net/Articles/276973/>>.

⁹ "Specifications." *AWUS036NHA*. ALFA Networks. Web. 17 May 2012. <<http://www.alfa.com.tw/in/front/bin/ptdetail.phtml?Part=AWUS036NHA>>.

chipset family is one of the most supported in the Linux kernel, and the Alfa wireless adapter has one of the largest power outputs, which is needed in order to be able to provide a strong access point.

5.1.4 Belkin F4U040V 4-Port USB 2.0 Powered Hub

The power output of a the USB port on the Beaglebone is only 500mA¹¹, and the power requirements of the webcam and Wi-Fi adapter are greater than that,¹² so a powered hub is needed. A solution with the minimum amount of available ports is needed in order to be able to fit inside a small enclosure.

5.1.5 SparkFun Pan/Tilt Bracket and Small Servo

In order to implement the pan and tilt functionality of the system, a two-axis system needs to be selected. The bracket and servo combination provided by Spark Fun is the ideal solution to this problem not only mechanically, but also electrically.¹³ The same voltage rail as the main Beaglebone powers the servos, so no conversion is needed in order to provide the correct voltage and necessary current to drive the servomotors. The dual servo setup for the bracket also provides sufficient torque to move the webcam without any issue.¹⁴

¹⁰ "Ath9k_htc Supported Features." *Ath9k_htc*. Linux Wireless. Web. 17 May 2012. <http://linuxwireless.org/en/users/Drivers/ath9k_htc>.

¹¹ Coley, Gerald. "5.6 USB1 Port." *BeagleBone Rev A5 System Reference Manual*. 21. 2 Feb. 2012. Web. 17 May 2012. <http://beagleboard.org/static/BONESRM_latest.pdf>.

¹² Unpowered USB Hub caused kernel panic when both Wifi and Webcam were enabled

¹³ See Figure 3

¹⁴ Output torque: 1.4kg/cm (19.6oz/in), <http://www.sparkfun.com/products/9065>



Figure 3: Bracket and Webcam

5.1.6 SparkFun Opto-isolator Breakout

In order to ensure that the GPIO output of the Beaglebone can trigger the servo, an optical isolator circuit is used to convert the 3.3V output of the GPIO to 5V to drive the servomotors.¹⁵

¹⁵ Coley, Gerald. "6.3.8 GPIO Spec" *BeagleBone Rev A5 System Reference Manual*. 25. 2 Feb. 2012. Web. 16 May 2012. <http://beagleboard.org/static/BONESRM_latest.pdf>.

"BOB-09118." Opto-isolator Breakout. Spark Fun, Inc. Web. 17 May 2012. <<http://www.sparkfun.com/products/9118>>.

5.1.7 Cost of Hardware Materials

Product	Cost
Beaglebone Development Board	\$89
Logitech C110 Webcam	\$20
Alfa AWUS036NHA	\$20
Belkin Powered USB Hub	\$30
2 x Small Servo	\$17
Pan/Tilt Bracket	\$6
Opto-isolator Breakout	\$5
Various Build Materials	\$10
Total Cost	\$197

Table 1: Cost of Materials

5.2 Software

All of the software libraries used in the project are open source and installed from the distribution repository, when available, in order to simplify development, reliability, and reproducibility.

When a user connects to the system via WiFi, hostapd negotiates the connection by authenticating via WPA. Isc_dhcp_server assigns an IP address to the client. When the user navigates to the website, lighttpd serves up the correct page. If playing a saved video, Flowplayer searches for the video in the /var/www/videos directory. During live streaming, the lighttpd queries the web server running inside of Mjpg-streamer and displays the stream images. In order to save videos, ffmpeg takes the live stream from Mjpg-streamer and saves its output in /var/www/videos for Flowplayer to play, or for the user to download.¹⁶

¹⁶ See Figure 4: Software Flow for details

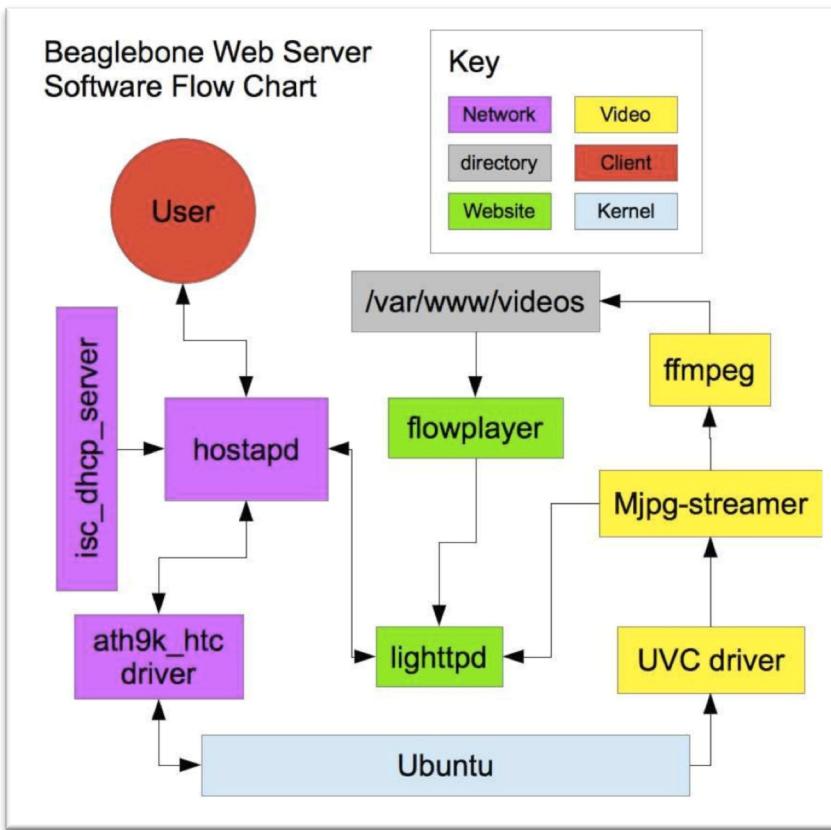


Figure 4: Software Flow

5.2.1 BeagleBoard Ubuntu

Ubuntu is a debian-based Linux distribution ported to the ARM architecture. It is a great choice for development. As one of the most popular distributions, it has a great deal of support and available packages in its repository.

5.2.2 Linux UVC driver

The Linux UVC driver is included in the Linux kernel and is loaded automatically upon booting the system with a webcam attached.

5.2.3 Ath9k_htc driver

This driver supports the AR9271 chipset used in the Alfa USB Wi-Fi module. The specific feature needed by project is the support for Monitor Mode. As an experimental feature, there are only a few drivers that support this mode.¹⁷

5.2.4 Hostapd

Hostapd¹⁸ is a program, used in conjunction with the ath9k_htc driver, which sets up the functionalities of AP/Monitor mode. It allows for different authentication modes such as WEP/WPA/WPA2 as well as any SSID configuration. Simply, it allows clients to connect to the server via Wi-Fi.

5.2.5 Isc-Dhcp-Server

Internet Systems Consortium's DHCP software¹⁹ is the pre-installed DHCP solution chosen by Ubuntu. This software allows any client who connects to the server via hostapd to be dynamically assigned an IP address. Without DHCP software, clients would need to manually assign their client an IP address, which would increase complexity of use, and therefore decrease the possible user base of the system.

5.2.6 Lighttpd

Lighttpd²⁰ is, as its name attests to, a full-featured lightweight web server, which allows the server to provide content to connected clients. The server not only provides the web page content, but also processes any PHP scripts that are needed for the project functionality.

¹⁷ "AP/P2P Modes." *Ath9k_htc - Linux Wireless*. Linux Wireless. Web. 17 May 2012. <http://linuxwireless.org/en/users/Drivers/ath9k_htc>.

¹⁸ <http://hostap.epitest.fi/hostapd/>

¹⁹ <http://www.isc.org/software/dhcp>

²⁰ <http://www.lighttpd.net/>

5.2.7 Mjpg-streamer

Mjpg-streamer²¹ is a software solution that takes raw frame data from the webcam and outputs a stream in the MJPG format, a series of JPEG images. The great feature of this software package is that it contains its own small web server that outputs the stream, as well as a great documentation on how to format and use the stream for different applications.

For this project, any HTTP request that Lighttpd receives for the webcam stream is forwarded to the Mjpg streamer server to process. This allows for a very lightweight capture process that does not use a large amount of CPU cycles converting the raw webcam stream to other more traditional video formats.

5.2.8 ffmpeg

ffmpeg²² is used in this project to convert the live mjpg stream to the FLV format when saving videos. The conversion process is run by a PHP script that executes ffmpeg in order to save a video to the server. The conversion process is extremely CPU intensive, and thus takes slightly longer to convert a video than it does to actually stream the video. This bottleneck was the main reason that ffmpeg was not used to convert the webcam data directly for live streaming.

5.2.9 Flowplayer

Flowplayer²³ is an open source video player that is used to playback saved videos on the server. It allows for pausing and reloading of the video, as well changing playback cursor position.

²¹ <http://sourceforge.net/projects/mjpg-streamer/>

²² <http://ffmpeg.org/ffmpeg.html>

²³ <http://flowplayer.org/>

5.2.10 Servo Control Script

The servo control script is what allows the two servos to move the webcam along the X and Y-axes. This PHP script manipulates system files in order to toggle GPIOs at certain timed intervals in order to produce the correct square wave to move the servos to different positions

5.2.11 Daemon Startup Scripts

Startup scripts ensure that Mjpg-streamer, lighttpd, hostapd, and isc-dhcp-server automatically start on power up of the system. In Ubuntu, the Upstart framework controls this process.²⁴ Upstart uses the concept of services in order to control the running of processes. Mjpg-streamer has a custom upstart configuration file that is used to start the webcam conversion.²⁵

5.2.12 PHP, JavaScript, and AJAX

The entire user interface is written with PHP, JavaScript, and AJAX. PHP is used to search for videos on the server, as well as execute processes on the server needed for video conversion. JavaScript controls all of the buttons and forms, as well as launch some of the PHP that require the entire web page to refresh, such as video deletion. AJAX is used in conjunction with JavaScript in order to execute the PHP scripts that do not require the whole web page to refresh, such as saving a video file.

²⁴ <http://upstart.ubuntu.com/>

²⁵ See Appendix 7.3

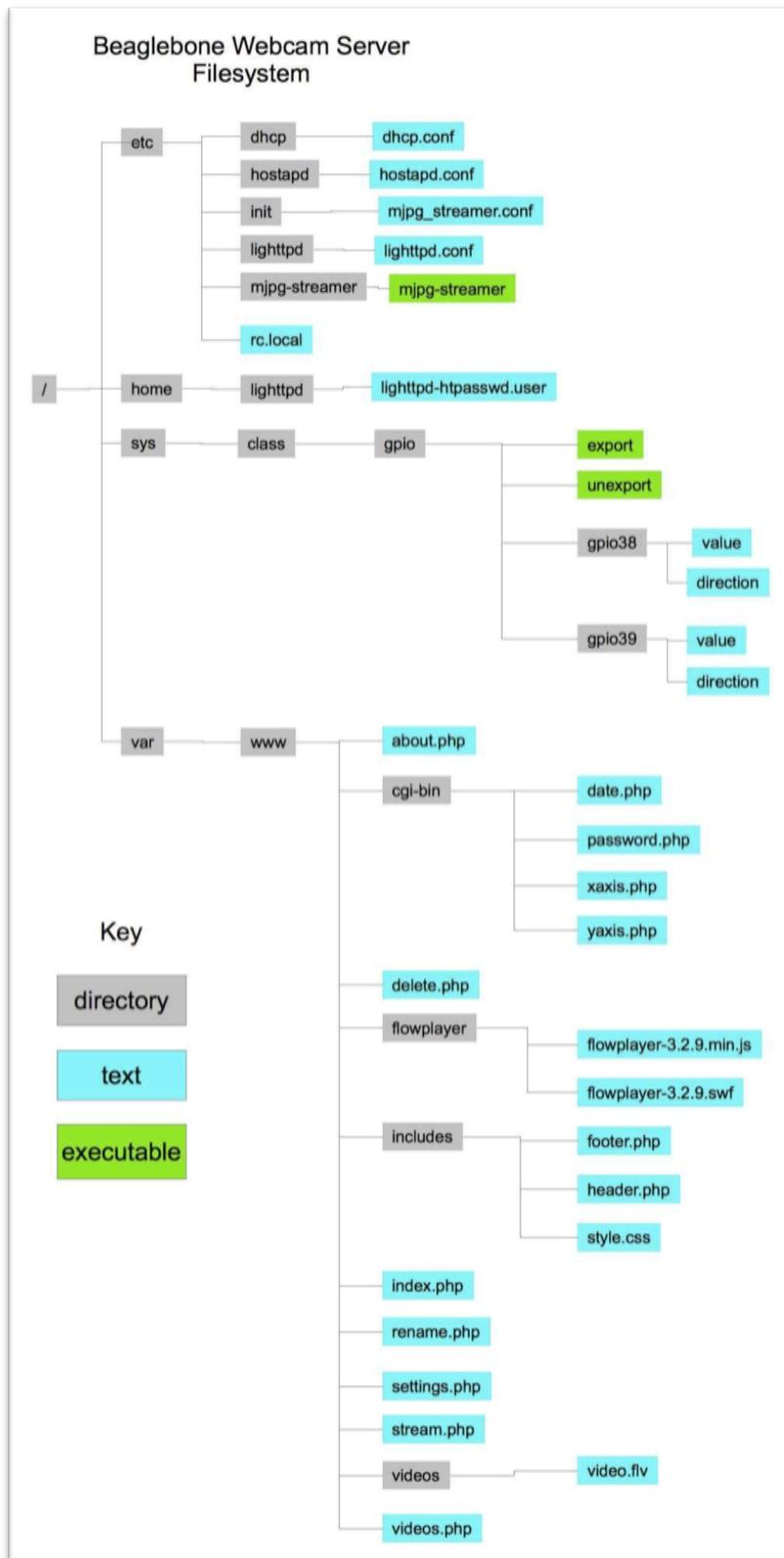


Figure 4: Filesystem

6 Development

The development of the Beaglebone Webcam Server took place during the ten weeks of Spring Quarter 2012 at Cal Poly. During the course of this project, a great deal of time was spent researching different implementations and configurations.

6.1 Filesystem

In addition to the files included with the distribution and files installed with packages, the files that were created or of special interest to the development process are listed in Figure 4.

6.2 Distribution Selection

The Beaglebone documentation suggests using the Angstrom distribution, a minimalist distribution based on the OpenEmbedded build framework, for the kernel.²⁶ Angstrom is a great distribution for optimizing an embedded system with the exact packages and features, but that ability comes at the cost of a steep learning curve, as well as very long compile times for the kernel images and modules. Angstrom did not include the ath9k_htc USB Wi-Fi drivers, and the process to get the kernel headers set up correctly in order to build the module did not have a high success rate. The tight development schedule of this project required a kernel that included all of the modules to be readily available, as well as additional unforeseen packages to be easily installable. The Ubuntu distribution was chosen as the kernel platform for development given its large package repository as built in driver support. Ubuntu ends up having a larger footprint than an optimized Angstrom build, but for development, it perfectly meets the requirements.

²⁶ <http://www.angstrom-distribution.org/building-angstrom> for details about building and configuring Angstrom

6.3 Live Video Streaming: MJPG-streamer

The core feature needed for this project, above all others, was the ability to view a live stream of the webcam through a web browser. The first implementation of this feature was using ffmpeg and ffserver.²⁷ With this setup, there was a five second lag between the webcam input and what was output to the web browser. This discrepancy slowly increased with time until the ffmpeg buffer filled up, causing it to crash. The reason for the lag and eventual crash was the lack of CPU resources to be able to transcode in real time. The solution to this problem was to use a program called Mjpg-streamer.²⁸ The program source was downloaded to the /etc directory and compiled.²⁹ When running, Mjpg-streamer serves up snapshots as well as a live stream via a built in web server set to accept requests from port 8080. In order to embed the image, the Mjpg-streamer documentation provides a JavaScript function that requests a snapshot and embeds the image into the page. For every frame, the JavaScript will replace the previous image with a new image, giving the illusion of a video stream.³⁰ This script was modified to accept a different image source and have the image be able to display correctly via CSS.

6.4 Saving and Playback of the Live Stream

In order to save the live webcam stream for later playback, ffmpeg is used to transcode the stream output from Mjpg-streamer. As mentioned in section 6.3, stream mode is one of the two output methods, the other being snapshot mode, which is used to view the stream in the browser. Selecting a time value from the drop down menu on the stream page will trigger, via AJAX, a php script that executes ffmpeg using the given time value.³¹ This video is outputted as a .flv video, and saved into the videos folder on the web server.³²

²⁷ <http://ffmpeg.org/ffserver.html>

²⁸ http://sourceforge.net/apps/mediawiki/mjpg-streamer/index.php?title=Main_Page

²⁹ See figure 3 for Mjpg-streamer executable location

³⁰ see Appendix 7.18 for image streaming script code

³¹ see Appendix 7.9 for video saving php code

³² see figure 3 for video.flv file in filesystem tree.

When opening the videos page on the web server, a php script will search the videos directory and list the video files on the page.³³ When a video is clicked, the same page is loaded with an additional php variable in the URL, which triggers the script to embed Flowplayer with the selected video. This use of a GET type variable allows the video to be bookmarked for easy playback.

6.5 Configuring Lighttpd

There were three functions that the web server needed to provide, besides serving basic pages. The server needed to provide authentication of users, querystring³⁴ redirection to the Mjpg-streamer server, and php scripting. Authentication is provided by the auth module in lighttpd, and uses the htpasswd³⁵ method for password file generation. The querystring redirection is needed because of the URL format of the Mjpg-streamer web server providing the snapshot and stream functionality. All requests for the root web server directory with a querystring should be redirected to port 8080 to be served by Mjpg-streamer web server. Fast-cgi³⁶ is enabled for php scripts, in order to increase performance by reducing some overhead associated with normal php interfaces.

6.6 Servo Control: GPIO vs. PWM

Servos need a square wave of 50Hz in order to trigger movement. Varying the duty cycle of the wave controls the position of the gear. Ideally, this functionality would be implemented using the hardware based PWM chip on the CPU. In order to be able to access the PWM registers through the filesystem, a specific kernel or kernel module was needed.³⁷ Due to the issues covered in section 6.1, GPIOs were selected as opposed to PWM. The

³³ see Appendix 7.19 for video directory search code

³⁴ www.example.com/? in the URL

³⁵ <http://redmine.lighttpd.net/projects/lighttpd/wiki/Docs:ModAuth> see section 6.7 for htpasswd details

³⁶ <http://redmine.lighttpd.net/projects/lighttpd/wiki/Docs:ModFastCGI>

³⁷ http://linux.org/BeagleBoard/GSoC/2010_Projects/Pulse_Width_Modulation

GPIOs are accessed through the filesystem by writing a port number to the export file.³⁸

GPIO ports 38 and 39 were selected as the ports for the two servos used to move the webcam. Once written to, the directory corresponding to the GPIO port is created. A php script³⁹ will toggle the value of the GPIO, based on a time value passed to the script from the user-selected drop down list value. The script toggles the value file inside of the GPIO port directory.⁴⁰ In order to allow the GPIO filesystem to be able to be accessed by the web server, the files need to be owned by the lighttpd user. This is accomplished by running chown from the rc.local boot-up script.⁴¹

6.7 Changing Settings and Executing Processes with PHP

All of the dynamic functionality of the Beaglebone Web Server is accomplished through PHP scripting, JavaScript calls to php scripts, or JavaScript AJAX calls. In order to change the login user and password, the lighttpd-htpasswd.user file is deleted, and a new file is created by calling the htpasswd program with the new username and password as parameters.⁴² When the system is connected in wired mode, and has access to the Internet upon boot-up, the rc.local script file will update the time. When in wi-fi mode, the time is set to the default time, which is not correct. In order to change the time, the date command needs to be executed.⁴³ Since date needs to have root access, the lighttpd user was added to the sudoers file, only giving access to the date program.

When renaming files, the filename needs to contain valid characters. This is accomplished with a regular expression match in PHP.⁴⁴ The filenames can only contain the letters A-Z, a-z, 0-9, dash, underscore, and dot. The delete video script uses a html POST

³⁸ <http://www.gigamegalog.com/2012/03/16/beaglebone-coding-101-buttons-and-pwm/>

³⁹ see Appendix 7.18 moveServo()

⁴⁰ see Appendix 7.10 or 7.11

⁴¹ see Appendix 7.5

⁴² see Appendix 7.8

⁴³ see Appendix 7.7

⁴⁴ see Appendix 7.9

php script in order to remove a file in order to force the user to navigate back to the videos page to regenerate the list of videos.⁴⁵

All of the PHP scripts that are called with AJAX are placed in the cgi-bin directory. They will feed back a message to display on the page they were called from. They will execute a process, given passed in variables, and return a result as a string.

7 Appendix: Code

This section contains all of the code created for this project. This does not include any of the Linux system files that would be created when installing the other software via the package manager.

7.1 /etc/dhcp/dhcp.conf

```
default-lease-time 600;
max-lease-time 7200;

option domain-name "webcam.com";
authoritative;

subnet 192.168.1.0 netmask 255.255.255.0 {
    range 192.168.1.100 192.168.1.163;
    option routers 192.168.1.164;
    option ip-forwarding off;
    option subnet-mask 255.255.255.0;
    option broadcast-address 192.168.1.255;
    option domain-name "webcam.com";
    option domain-name-servers 192.168.1.164, 192.168.1.164;
}
```

7.2 /etc/hostapd/hostapd.conf

```
wpa_passphrase=beaglebone
ssid=Webcam
driver=n180211
interface=wlan0
auth_algs=3
wpa=2
wpa_pairwise=CCMP
wpa_key_mgmt=WPA-PSK
hw_mode=g
channel=2
macaddr_acl=0
```

7.3 /etc/init/mjpg_streamer.conf

```
# mjpg_streamer - stream jpeg images from UVC webcam
```

⁴⁵ see section 6.4 for details on generation of videos list

```

description      "jpeg streaming for UVC webcam"
start on runlevel [2345]
stop on runlevel [016]

script
    mjpg_streamer -b -i "/etc/mjpg-streamer/input_uvc.so -r 640x480 -f 15 -d /dev/video0" -o "/etc/mjpg-streamer/output_http.so -p 8080"
end script

```

7.4 /etc/lighttpd/lighttpd.conf

```

server.document-root = "/var/www/"
server.port = 80
server.username = "lighttpd"
server.groupname = "lighttpd"
server.bind = "192.168.1.164"
server.tag = "lighttpd"
server.socket = "[::]:80"
server.errorlog        = "/var/log/lighttpd/error.log"
accesslog.filename     = "/var/log/lighttpd/access.log"

server.modules          =
    "mod_access",
    "mod_accesslog",
    "mod_fastcgi",
    "mod_rewrite",
    "mod_proxy",
    "mod_auth",
)
$HTTP["host"] =~ "192.168.1.164" {
$HTTP["url"] =~ "^/" {
auth.backend = "htpasswd"
auth.backend.htpasswd.userfile = "/home/lighttpd/lighttpd-
htpasswd.user"
auth.require = ("/" => (
"method"  => "basic",
"realm"   => "Beaglebone",
"require" => "valid-user"
))
}
}

$HTTP["querystring"] =~ "action=snapshot" {
proxy.server = ( "" => (( "host" => "127.0.0.1", "port" => 8080
)))
}
index-file.names = ( "index.html", "index.php" )
fastcgi.server = ( ".php" => (
"bin-path" => "/usr/bin/php5-cgi",
"socket"   => "/tmp/php.socket"
),
".py" => (
"bin-path" => "/usr/bin/python",
"socket"   => "/tmp/fastcgi.python.socket",
)
)

```

7.5 /etc/rc.local

```

#!/bin/sh -e
#
# rc.local
#
# This script is executed at the end of each multiuser runlevel.
# Make sure that the script will "exit 0" on success or any other

```

```

# value on error.
#
# In order to enable or disable this script just change the execution
# bits.
#
# By default this script does nothing.
#http://www.gigamegablog.com/2012/03/16/beaglebone-coding-101-buttons-
and-pwm/
chown lighttpd:root /sys/class/gpio/export
chown lighttpd:root /sys/class/gpio/unexport
echo 38 > /sys/class/gpio/export
chown lighttpd:root /sys/class/gpio/gpio38/value
chown lighttpd:root /sys/class/gpio/gpio38/direction
#set mux to mode 7, 38 already mode 7
echo 7 > /sys/kernel/debug/omap_mux/gpmc_ad7
echo 39 > /sys/class/gpio/export
chown lighttpd:root /sys/class/gpio/gpio39/value
chown lighttpd:root /sys/class/gpio/gpio39/direction

ntpdate pool.ntp.org
export TZ=America/Los_Angeles

exit 0

```

7.6 /var/www/about.php

```

<?php include ('includes/header.php')?>
<p>This webcam server runs on a Beaglebone and uses a Logitech
webcam.<p>
<h3>Stream</h3> View a live stream of the webcam.
<ul>
<li>In order to rotate the webcam, click on the [Left] and [Right]
buttons to move the camera along the x-axis.</li>
<li>Move the webcam up and down along the vertical axis by clicking
the [Up] and [Down] buttons.</li>
<li>To save a video clip, select a length from the drop down menu, and
then select the [Save Video] button to begin transcoding the video.
when the video is completed, a message will appear below the time
select drop down menu signaling success.</li>
</ul>
<h3>Videos</h3> See a list of all of the saved videos, and free space
left on the server.
<ul>
<li>On this page is listed all of the videos that have been previously
saved through the stream page.</li>
<li>There are three options for each video: 1. Play the video through
the web browser, 2. Download the video to your computer, 3. Select
videos to delete from the server.</li>
<li>when a video is selected to be played in the browser, it is
embedded in the same page, with the other videos listed below it in the
browser window. This page can be bookmarked for later playback, using
the format videos.php?q=filename.</li>
<li>When deleting videos, select the check box that corresponds to the
video in the list, then click on the delete button. Another page will
load, confirming that the video has been deleted.</li>
</ul>
<h3>Settings</h3> Change various settings of the server.
<ul>
<li>Change Login User: Enter a new username and password for the
server, changes take effect immediately.</li>
<li>Rename Video: Select a saved video to rename. Enter the desired new
file name in the text box. Filenames are only allowed to have the
letters A-Z, a-z, numbers 0-9, and underscore _, dash -, and dot. If

```

the new file name does not have the correct .flv extension, it will be added automatically.

Change Date: Enter a new date and time, following the format listed, in order to change the system time to match the current time. This change will be reflected not only in the time listed in the footer of the web pages, but also the default filename of saved videos.


```
<?php include ('includes/footer.php')?>
```

7.7 /var/www/cgi-bin/date.php

```
<?php
$date = $_GET["q"];
exec("sudo date ".$date);
echo "date changed, refresh page to see changes!"; ?>
```

7.8 /var/www/cgi-bin/password.php

```
<?php
$user = $_POST["user"];
$password1 = $_POST["pass"];
$password2 = $_POST["verify"];
$passwordFile = "/home/lighttpd/lighttpd-htpasswd.user";
$cmd = "htpasswd -cb ".$passwordFile." ".$user." ".$password1;

if($password1 !== $password2){
    echo "Passwords did not match!";
}
else{
    clearstatcache();
    if(unlink($passwordFile) == FALSE){
        if(is_file($passwordFile) == FALSE){
            exit("Cannot delete old password file, not a file!");
        }
        else if(is_writable($passwordFile) == FALSE){
            exit("Cannot delete old password file, not writable!");
        }
        else{
            exit("Cannot delete old password file, but is a file and is
writable!");
        }
    }
    $result = exec($cmd);
    echo "Success! New username is: ".$user;
}?>
```

7.9 /var/www/cgi-bin/savevid.php

```
<?php
$time = $_GET['q'];
if($time === 'blank'){
    echo "Please select a recording time.";
}
else{
    date_default_timezone_set('America/Los_Angeles');
    $filename = date("n-j-Y_g:i:sa");
    $command = "ffmpeg -er 4 -y -r 5 -t ".$time." -f mjpeg -i
http://localhost:8080/?action=stream ..../videos/".$filename.".flv
2>&1";
    $output = exec($command, $results);
    echo "Video ".$filename.".flv complete!";
}
?>
```

7.10 /var/www/cgi-bin/xaxis.php

```
<html>
<body>
<?php
$offset = $_GET["q"];
$file=fopen("/sys/class/gpio/gpio39/direction","w") or exit("Unable to
open direction!");
fwrite($file, "out", strlen("out"));
fclose($file);
$file=fopen("/sys/class/gpio/gpio39/value","w") or exit("Unable to open
value!");
$basetime = 20000;
$i=0;
while($i <= 3){
    fwrite($file, '1');
    fflush($file);
    usleep($offset);
    fwrite($file, '0');
    fflush($file);
    usleep($basetime - $offset);
    $i++;
}
fclose($file);?>
</body>
</html>
```

7.11 /var/www/cgi-bin/yaxis.php

```
<html>
<body>
<?php
$offset = $_GET["q"];
$file=fopen("/sys/class/gpio/gpio38/direction","w") or exit("Unable to
open direction!");
fwrite($file, "out", strlen("out"));
fclose($file);
$file=fopen("/sys/class/gpio/gpio38/value","w") or exit("Unable to open
value!");
$basetime = 20000;
$i=0;
while($i <= 3){
    fwrite($file, '1');
    fflush($file);
    usleep($offset);
    fwrite($file, '0');
    fflush($file);
    usleep($basetime - $offset);
    $i++;
}
fclose($file);?>
</body>
</html>
```

7.12 /var/www/includes/footer.php

```
</div>
<div id="footer">
<br>Created by Alexander Corcoran<br>
<?
    date_default_timezone_set('America/Los_Angeles');
    echo "Page last refreshed: ". date("F d, Y H:i:s", time());
?>
</div>
</div>
```

```
</body>
</html1>
```

```
7.13 /var/www/includes/header.php
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head>
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
<meta name="description" content="" />
<meta name="keywords" content="" />
<meta name="author" content="" />
<link rel="stylesheet" type="text/css" href="includes/style.css"
media="screen" />
<title>Beaglebone Webcam Server </title>
</head>
<!-- based on http://www.1stwebdesigner.com/css/how-to-create-php-
website-template/ -->
<body>
<div id="wrapper">
<div id = "header">
<h2>Beaglebone Webcam</h2>
</div>
<div id="nav">
<a href="index.php">Home</a>
<a href="stream.php">Stream</a>
<a href="videos.php">Videos</a>
<a href="settings.php">Settings</a>
<a href="about.php">About</a>
</div>
<div id="content">
```

```
7.14 /var/www/includes/style.css
```

```
body {
background-color:#f1f1f1;
font-family: georgia,sans-serif;
color:#333;
margin:0;
padding:0;
}
#wrapper {
width:960px;
background-color:#f8f8f8;
margin:0 auto;
border-left:1px solid #ccc;
border-right:1px solid #ccc;
}
#header {
width:960px;
height:85px;
margin:0 auto;
margin-bottom:25px;
border-bottom:1px solid #ccc;
border-top:1px solid #ccc;
}
#header h2 {
padding:10px;
}
#nav {
width:960px;
height:40px;
border-bottom:1px solid #ccc;
}
```

```

#nav a {
  display:inline;
  padding:10px;
  text-decoration:none;
  background-color:#f8f8f8;
}
#nav a:hover {
  background-color:#bababa;
  height:80px;
}
#content {
  width:900px;
  float:left;
  padding:10px;
}
#webcam {
  width:800px;
  float:left;
  padding:10px;
}
#sidebar a {
  text-decoration:none;
}
#sidebar li {
  list-style:none;
}
#footer {
  clear:both;
  width:960px;
  height:135px;
  border-top:1px solid #ccc;
}
#footer p {
  padding:10px;
}

```

7.15 /var/www/index.php

```

<?php include ('includes/header.php')?>
<p> Welcome to the Beaglebone Webcam Server </p>

<ul>
<li>To view the live stream click on the Stream button on the top toolbar.</li>
<li>To view any saved videos, select the Videos button on the top toolbar.</li>
<li>To edit any settings for the webcam server, select the Settings button on the top toolbar.</li>
<li>To view details about the webcam server select About from the top toolbar.</li>
</ul>
</img>
<?php include ('includes/footer.php')?>

```

7.16 /var/www/rename.php

```

<?php include ('includes/header.php')?>

<?php
$dir="videos/";
$filename = $_POST['renameFiles'];
$newname = $_POST['newname'];
$fileExtension = ".flv";
if(empty($filename)){

```

```

        echo "Please select a video to rename from the drop down menu in
the settings.";
    }
    else if(empty($newname)){
        echo "Please type a new filename for the selected video.";
    }
    //http://www.webdeveloper.com/forum/showthread.php?t=213200
    else if(!preg_match('/^([a-zA-Z0-9_-]+)$/', $newname)){
        echo "Filename <b>".$newname."</b> contains illegal characters,
filenames can only contain: ";
        echo "<ul><li>A-Z</li><li>a-z</li><li>underscores</li><li>dashes -
</li><li>dots .</li></ul>";
    }
    else{
        //last occurrence of substring in string: for file extension
        $offset = strrpos($newname, $fileExtension);
        if($offset === FALSE){
            echo "Appending .flv extension, ";
            $newname = $newname.$fileExtension;
        }
        //else if( ($offset + strlen($fileExtension)) ===
        strlen($newname)){
            //echo "file extension correct!";
            //}
            if(rename($dir.$filename, $dir.$newname) == FALSE){
                echo "Error, could not rename ".$filename."!";
            }
            echo "Video <b>".$filename."</b> successfully renamed to
<b>".$newname."</b>!";
        }
    ?>
    <?php include ('includes/footer.php')?>

7.17 /var/www/settings.php
<?php include ('includes/header.php')?>

<script type="text/javascript">
function changePassword(){ // This function does the AJAX request
    var xmlhttp;
    var user = document.getElementById('username').value;
    var pass1 = document.getElementById('pwd1').value;
    var pass2 = document.getElementById('pwd2').value;
    var params = "user="+user+"&pass="+pass1+"&verify="+pass2;

        if(confirm('Are you sure you want to change the login user?')){
            if (window.XMLHttpRequest){// code for IE7+, Firefox, Chrome,
Opera, Safari
                xmlhttp=new XMLHttpRequest();
            }
            else{// code for IE6, IE5
                xmlhttp=new ActiveXObject("Microsoft.XMLHTTP");
            }
            xmlhttp.onreadystatechange = function(){
                if (xmlhttp.readyState==4 && xmlhttp.status==200){

                    document.getElementById("success").innerHTML=xmlhttp.responseText;
                }
            }
            xmlhttp.open("POST", "cgi-bin/password.php", true);
            xmlhttp.setRequestHeader("Content-type", "application/x-www-form-
urlencoded");
            //http://www.openjs.com/articles/ajax_xmlhttp_using_post.php
            xmlhttp.send(params);
        }
    }

```

```

}

function changeDate(){ // This function does the AJAX request
    var xmlhttp;
    var date = document.getElementById('dateBox').value;

        if (window.XMLHttpRequest){// code for IE7+, Firefox, Chrome,
        Opera, Safari
            xmlhttp=new XMLHttpRequest();
        }
        else{// code for IE6, IE5
            xmlhttp=new ActiveXObject("Microsoft.XMLHTTP");
        }
        xmlhttp.onreadystatechange = function(){
            if (xmlhttp.readyState==4 && xmlhttp.status==200){

                document.getElementById("dateSuccess").innerHTML=xmlhttp.responseText;
            }
        }
        xmlhttp.open("GET", "cgi-bin/date.php?q="+date, true);
        xmlhttp.setRequestHeader("Content-type", "application/x-www-form-urlencoded");
        xmlhttp.send(null);
    }

function checkFilename(str){
    var patt = /^[a-zA-Z0-9._-]+$/;
    if(str.length == 0){
        document.getElementById('nameError').innerHTML="";
    }
    else if(!patt.test(str)){
        document.getElementById('nameError').innerHTML="Filename can only
        can contain <ul><li>A-Z</li><li>a-z</li><li>0-9</li><li>dash -
        </li><li>dot .</li></ul>";
    }
    else{
        document.getElementById('nameError').innerHTML="";
    }
}
</script>

<p><b>Change Login User</b></p>
Enter Username: <input type="text" id='username' size="20"><br>
Enter Password: <input type="password" id='pwd1' size="20"><br>
Verify Password: <input type="password" id='pwd2' size="20"><br>
<input type="button" value="Change Login" name="submitButton"
onClick="changePassword()"/></button>
<br><span id="success"></span><br>
<p><b>Rename Video</b></p>

<?php
$dir = "videos/";
$fileExtension = ".flv";
$files = scandir($dir);
$numFiles = count($files);
echo "<form action = \"rename.php\" method=\"post\">";
echo "<select name = \"renameFiles\" id=\"fileMenu\">";
echo "<option value= \"\" selected>Saved Videos</option>";
foreach($files as $value){
    if(strpos($value, $fileExtension) !== false){
        echo "<option value =\"".$value."\">".$value."</option>";
    }
}
?>

```

```

</select><br>
New File Name:<input type = "text" id='fin' name="newname"
onkeyup="checkFilename(this.value)" />
<input type="submit" value="Rename Selected"/>
</form>
<span id='nameError'></span><br>
<p><b>Change Date</b></p>
Enter New Date: <input type="text" id='dateBox' size="20"><br>
<input type="button" value="Change Date" name="dateSubmitButton"
onClick="changeDate()"/></button>
<span id='datesuccess'></span><br>
<br>Please use the date format: MMDDHHmmYYYY.ss, where <ul><li>MM is
the two digit month between 01 and 12</li><li>DD is the two digit day
between 01 and 31</li><li>HH is the two digit hour between 00 and 23
(24hr format)</li><li>mm is the two digit minute between 00 and
59</li><li>YYYY is the four digit year</li><li>ss is the two digit
seconds, listed after a . (dot)</li></ul>
<?php include ('includes/footer.php')?>

```

7.18 /var/www/stream.php

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head>
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
<meta name="description" content="" />
<meta name="keywords" content="" />
<meta name="author" content="" />
<link rel="stylesheet" type="text/css" href="includes/style.css"
media="screen" />
<title>Beaglebone Webcam Server </title>
</head>
<!-- based on http://www.1stwebdesigner.com/css/how-to-create-php-
website-template/ -->
<body onload="createImageLayer();">
<div id="wrapper">
<div id ="header">
<h2>Beaglebone Webcam</h2>
</div>
<div id="nav">
<a href="index.php">Home</a>
<a href="stream.php">Stream</a>
<a href="videos.php">Videos</a>
<a href="settings.php">Settings</a>
<a href="about.php">About</a>

```

</div>

```

<div id="content">

<h2>Live Webcam Stream</h2>

<script type="text/javascript">
/* Copyright (C) 2007 Richard Atterer, richard@atterer.net
   This program is free software; you can redistribute it and/or modify
   it
   under the terms of the GNU General Public License, version 2. See
   the file
   COPYING for details. */
//Modified by Alexander Corcoran - acorcora@calpoly.edu

var imageNr = 0; // Serial number of current image
var finished = new Array(); // References to img objects which have
finished downloading
var paused = false;
```

```

function createImageLayer() {
    var img = new Image();
    img.style.position = "absolute";
    img.style.cssFloat = "inherit";
    img.border = 2;
    img.style.zIndex = -1;
    img.onload = imageOnload;
    //img.onclick = imageOnClick;
    img.src = "?action=snapshot&n=" + (++imageNr);
    var webcam = document.getElementById("webcam");
    webcam.insertBefore(img, webcam.firstChild);
}

// Two layers are always present (except at the very beginning), to
// avoid flicker
function imageOnload() {
    this.style.zIndex = imageNr; // Image finished, bring to front!
    while (1 < finished.length) {
        var del = finished.shift(); // Delete old image(s) from document
        del.parentNode.removeChild(del);
    }
    finished.push(this);
    if (!paused) createImageLayer();
}

function imageOnClick() { // Clicking on the image will pause the
//stream
    paused = !paused;
    if (!paused) createImageLayer();
}

//http://www.w3schools.com/php/php_ajax_php.asp modified from
function moveServo(axis, value){ // This function does the AJAX
request
    var xmlhttp;
    var LEFT = 'left';
    var RIGHT = 'right';
    var UP = 'up';
    var DOWN = 'down';
    var CENTER = 'center';
    var XAXIS = 'X';
    var YAXIS = 'Y';
    if( typeof moveServo.xVal == 'undefined' ){
        moveServo.xVal = 1210;
    }
    if( typeof moveServo.yVal == 'undefined' ){
        moveServo.yVal = 1380;
    }
    if (window.XMLHttpRequest){// code for IE7+, Firefox, Chrome,
Opera, Safari
        xmlhttp=new XMLHttpRequest();
    }
    else{// code for IE6, IE5
        xmlhttp=new ActiveXObject("Microsoft.XMLHTTP");
    }
    xmlhttp.onreadystatechange = function(){
        if (xmlhttp.readyState==4 && xmlhttp.status==200){

            document.getElementById("servoResult").innerHTML=xmlhttp.responseText;
        }
    }
    if(axis === 'Y'){//move Y
}

```

```

//accounts for center button not completing if far from center
if(value == DOWN && moveServo.yVal < 2230){
    moveServo.yVal += 170;
}
else if(value == UP && moveServo.yVal > 200){
    moveServo.yVal -= 170;
}
else if(value == CENTER){
    moveServo.yVal = 1380;
}
xmlhttp.open("GET", "cgi-bin/yaxis.php?q="+moveServo.yVal, true);
xmlhttp.send(null);
}
else if(axis === 'x'){//move x
if(value == RIGHT && moveServo.xVal < 1890){
    moveServo.xVal += 170;
}
else if(value == LEFT && moveServo.xVal > 200){
    moveServo.xVal -= 170;
}
else if(value == CENTER){
    moveServo.xVal = 1210;
}
xmlhttp.open("GET", "cgi-bin/xaxis.php?q="+moveServo.xVal, true);
xmlhttp.send(null);
}
else{
    alert("Error!");
}
}

function saveVideo(){ // This function does the AJAX request
    var xmlhttp;
    var time = document.getElementById('timeMenu').value;
    if (window.XMLHttpRequest){// code for IE7+, Firefox, Chrome,
Opera, Safari
        xmlhttp=new XMLHttpRequest();
    }
    else{// code for IE6, IE5
        xmlhttp=new ActiveXObject("Microsoft.XMLHTTP");
    }
    xmlhttp.onreadystatechange = function(){
        if (xmlhttp.readyState==4 && xmlhttp.status==200){

            document.getElementById("vidSaveResult").innerHTML=xmlhttp.responseText;
        }
    }
    document.getElementById("vidSaveResult").innerHTML="Saving Video
for "+time+"s...";
    xmlhttp.open("GET", "cgi-bin/saveid.php?q="+time, true);
    xmlhttp.send(null);
}
</script>
<div id="webcam"><noscript></noscript></div>
<br><br><br><br>
<br><br><br><br>
<br><br><br><br>
<br><br><br><br>
<br><br><br><br>
X-Axis:
<input type="button" value="Left" name="xMax" onclick="moveServo('x',

```

```

'left')"></button>
<!--<input type="button" value="Center" name="xCenter"
onClick="moveServo('X', 'center')"></button>-->
<input type="button" value="Right" name="xMin" onClick="moveServo('X',
'right')"></button>
<br>
Y-Axis:
<input type="button" value="Up" name="yMax" onClick="moveServo('Y',
'up')"></button>
<!--<input type="button" value="Middle" name="yCenter"
onClick="moveServo('Y', 'center')"></button>-->
<input type="button" value="Down" name="yMin" onClick="moveServo('Y',
'down')"></button>
<br><span id="servoResult"></span><br>

<select name = "time" id="timeMenu">
<option value= "blank" selected>Length of video</option>
<option value = "5">5s</option>
<option value = "10">10s</option>
<option value = "15">15s</option>
<option value = "20">20s</option>
<option value = "25">25s</option>
<option value = "30">30s</option>
<option value = "35">35s</option>
<option value = "40">40s</option>
<option value = "45">45s</option>
<option value = "50">50s</option>
<option value = "55">55s</option>
<option value = "60">60s</option>
</select>
<input type="button" value="Save Video" name="saveVid"
onClick="savevideo()"/></button>
<br><span id="vidSaveResult"></span><br>
<?php include ('includes/footer.php')?>

```

7.19 /var/www/videos.php

```

<?php include ('includes/header.php')?>
<script src="flowplayer/flowplayer-3.2.9.min.js"></script>
<?php
$filename=$_GET["q"];
$dir = "videos/";
$fileExtension = ".flv";
$files = scandir($dir);
$numFiles = count($files);
if($filename !== NULL){
    echo "<h1>".$filename."</h1>";

    echo "<p>".date("F d Y H:i:s.", filemtime($dir.$filename))."</p>";
    echo "<a href=\"videos/".$filename."\" style=\"display:block;width:520px;height:330px\" id=\"player\"></a>";
    echo "<script>flowplayer(\"player\", \"flowplayer-3.2.10.swf\");</script>";
}
echo "<h4>Saved Videos</h4>";
/*****
 * http://www.php.net/manual/en/function.disk-free-space.php
*/
echo "Free Space: ";
getBytes = disk_free_space(".");
$si_prefix = array( 'B', 'KB', 'MB', 'GB', 'TB', 'EB', 'ZB', 'YB' );
$base = 1024;
$class = min((int)log($bytes , $base) , count($si_prefix) - 1);
echo sprintf('%1.2f' , $bytes / pow($base,$class)) . ' ' .
$si_prefix[$class] . '<br />';

```

```

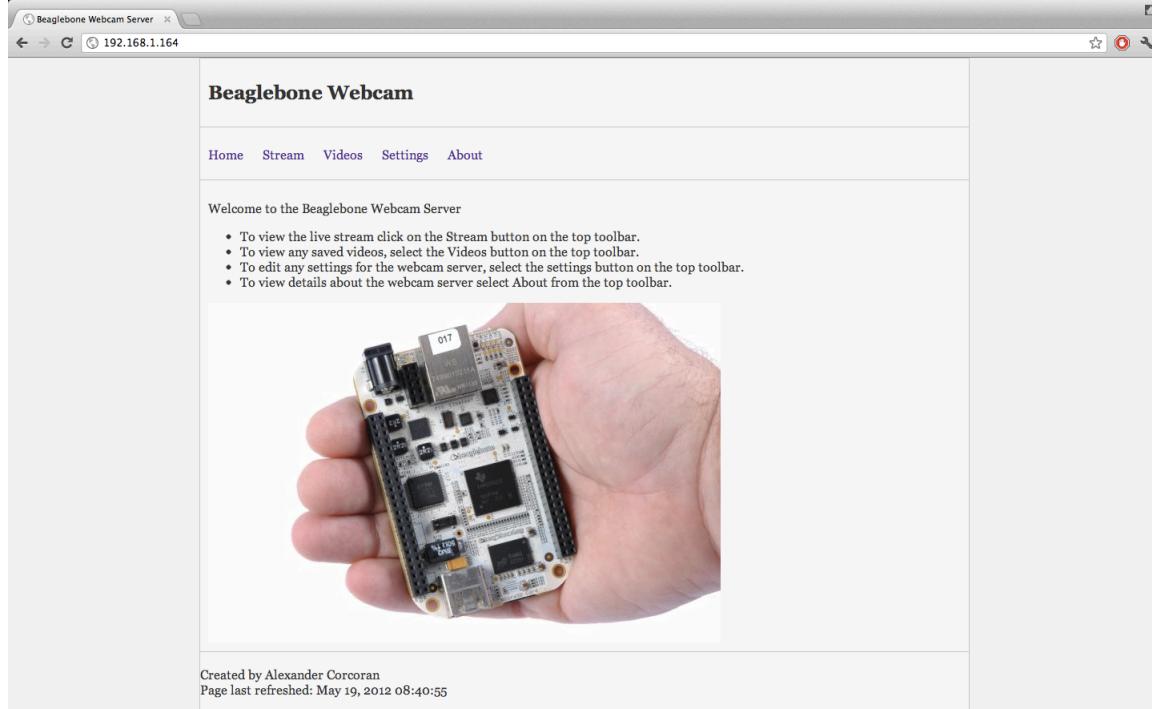
/*****
echo "<form action = \"delete.php\" method=\"post\>";
echo "<ul>";
foreach($files as $value){
    if(strpos($value, $fileExtension) !== false){
        echo "<li>Play: <a href=\"";
        echo "videos.php?q=". $value . "\">". $value . "</a><br>";
        echo "Download: <a href=\"". $dir.$value . "\">". $value . "</a><br>";
        echo "Delete: <input type=\"checkbox\" value=\"\"". $value . "\" name
= \"files[]\"></li>";
    }
}
echo "<input type=\"submit\" value=\"Delete Selected\"/>";
echo "</ul></form><br>";
?>
<?php include ('includes/footer.php')?>

```

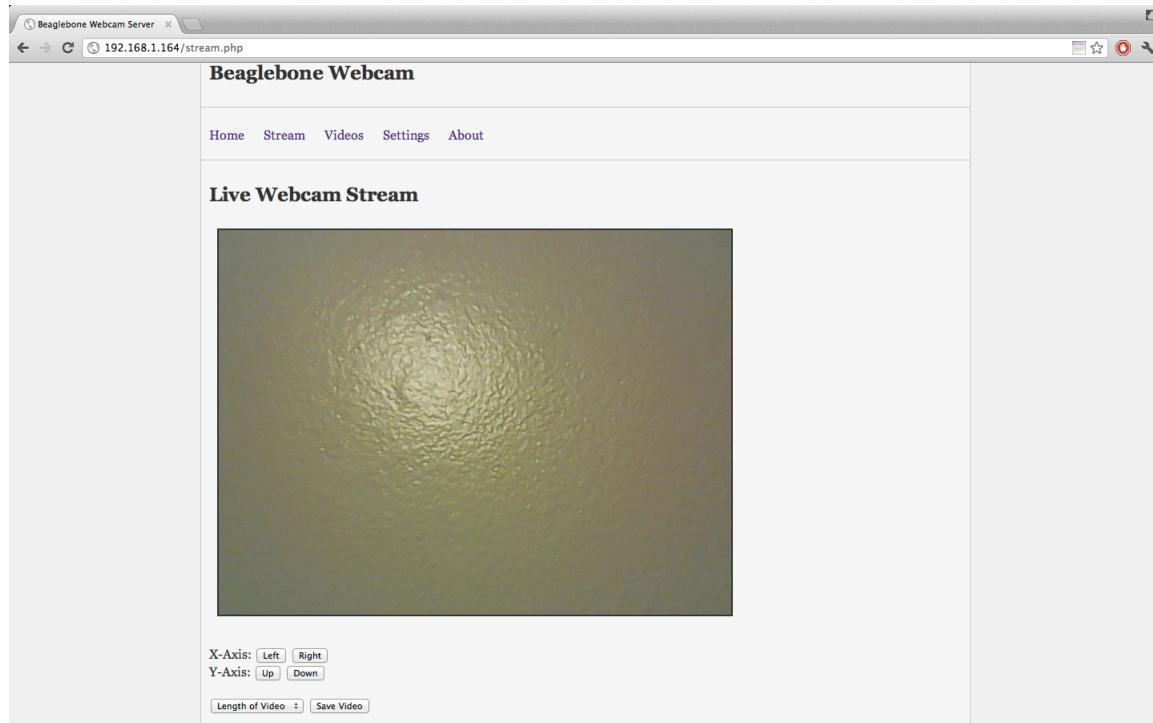
8 Appendix: User Interface

This section includes screenshots from the web browser interface.

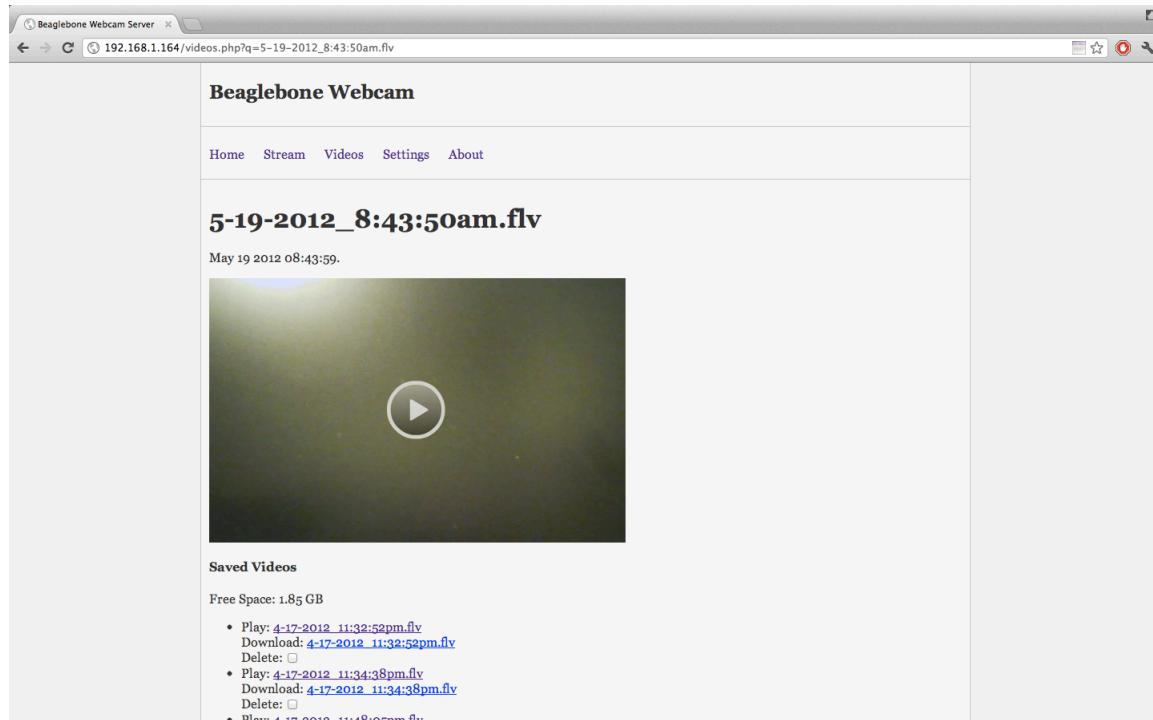
8.1 Home



8.2 Stream



8.3 Videos



8.4 Settings

The screenshot shows a web browser window titled "Beaglebone Webcam Server" with the URL "192.168.1.164/settings.php". The page has a header with links to Home, Stream, Videos, Settings, and About. Below the header are three main sections: "Change Login User", "Rename Video", and "Change Date".

- Change Login User:** Contains fields for Enter Username, Enter Password, Verify Password, and a Change Login button.
- Rename Video:** Contains a dropdown menu for Saved Videos, a New File Name input field, and a Rename Selected button.
- Change Date:** Contains fields for Enter New Date and a Change Date button. A note below explains the date format: MMDDHHmmYYYY.ss, where:
 - MM is the two digit month between 01 and 12
 - DD is the two digit day between 01 and 31
 - HH is the two digit hour between 00 and 23 (24hr format)
 - mm is the two digit minute between 00 and 59
 - YYYY is the four digit year
 - ss is the two digit seconds, listed after a . (dot)

At the bottom of the page, it says "Created by Alexander Corcoran" and "Page last refreshed: May 19, 2012 08:44:27".

8.5 About

The screenshot shows a web browser window titled "Beaglebone Webcam Server" with the URL "192.168.1.164/about.php". The page has a header with links to Home, Stream, Videos, Settings, and About. Below the header is a single section: "Stream".

This webcam server runs on a Beaglebone and uses a Logitech webcam.

Stream

View a live stream of the webcam.

- In order to rotate the webcam, click on the [Left] and [Right] buttons to move the camera along the x-axis.
- Move the webcam up and down along the vertical axis by clicking the [Up] and [Down] buttons.
- To save a video clip, select a length from the drop down menu, and then select the [Save Video] button to begin transcoding the video. When the video is completed, a message will appear below the time select drop down menu signaling success.

Videos

See a list of all of the saved videos, and free space left on the server.

- On this page is listed all of the videos that have been previously saved through the stream page.
- There are three options for each video: 1. Play the video through the web browser, 2. Download the video to your computer, 3. Select videos to delete from the server.
- When a video is selected to be played in the browser, it is embedded in the same page, with the other videos listed below it in the browser window. This page can be bookmarked for later playback, using the format `videos.php?q=filename`.
- When deleting videos, select the check box that corresponds to the video in the list, then click on the delete button. Another page will load, confirming that the video has been deleted.

Settings

Change various settings of the server.

- Change Login User: Enter a new username and password for the server, changes take effect immediately.
- Rename Video: Select a saved video to rename. Enter the desired new file name in the text box. Filenames are only allowed to have the letters A-Z, a-z, numbers 0-9, and underscore _, dash -, and dot. If the new file name does not have the correct .flv extension, it will be added automatically.
- Change Date: Enter a new date and time, following the format listed, in order to change the system time to match the current time. This change will be reflected not only in the time listed in the footer of the web pages, but also the default filename of saved videos