Building Networks and Servers Using BeagleBone

Create your own video theatre and music jukebox using BeagleBone. Make your multimedia library available to all the devices on your network, without spending an excessive amount of money on computer components and software.

You’ll start off by installing the Debian operating system image onto your BeagleBone, and setting up the software required to serve up both audio and video files to any device on the network. Next, you’ll move on to network monitoring software, and install traceroute, MTR (My traceroute), Nmap, and IPtraf-ng, to monitor the traffic on your local network. Once the monitoring software is set up, you’ll create a RAID array to store all your media files. You’ll then go one step further, and set up live and recorded video streaming using a web-based application. Finally, you’ll learn to add Wi-Fi connectivity to your multimedia server by setting up WAP on your BeagleBone system.

Who this book is written for

If you are a developer with BeagleBone experience and want to learn how to use it to set up a network and file server, then this book is ideal for you. To make the most of this book, you should be comfortable with the Linux operating system and know how to install software from the Internet, but you do not have to be a network guru.

What you will learn from this book

- Install and configure the latest Debian image onto your BeagleBone
- Set up simple file services and multimedia sharing services
- Monitor your network traffic to ensure high-speed streaming
- Build and utilize a RAID storage system
- Stream live video action from your BeagleBone
- Make your multimedia server wireless by setting up a DHCP Wi-Fi Server


Bill Pretty      Glenn Vander Veer
In this package, you will find:

- The author's biography
- A preview chapter from the book, Chapter 5 'Streaming Videos'
- A synopsis of the book’s content
- More information on Building Networks and Servers Using BeagleBone

About the Authors

Bill Pretty began his career in electronics in the early 80s with a small telecom start-up company that would eventually become a large multinational. He left this company to pursue a career in commercial aviation in northern Canada. Next, he joined the Ontario Center for Microelectronics, a provincially funded research and development center. Bill left this for a career in the military as a civilian contractor at what was then called Defense Research Establishment Ottawa. Thus began a career that was to span the next 25 years and continues until today.

Over the years, Bill acquired extensive knowledge in the field of technical security and started his own company in 2010. This company is called William Pretty Security Inc. and provides support in the form of research and development to various law enforcement and private security agencies.

Bill has published and presented a number of white papers on the subject of technical security. For a number of years, he was also a guest presenter at the Western Canada Technical Conference, a law enforcement-only conference held every year in western Canada. A selection of these papers is available for download from his website.

Glenn Vander Veer has been an embedded firmware developer for various microprocessors and microcontrollers for the past 20 years. He has been tinkering with electronics and all types of computers for over 30 years now. His interests lie in computer security, audio and video development, and tinkering with various operating systems. This is Glenn's first book, but definitely not his last.
Building Networks and Servers Using BeagleBone

Learn how to build and configure your own network based on the BeagleBone. You will do this in a fun and informative way that will not only teach you networking skills but also result in an impressive project.

What This Book Covers

Chapter 1, *Installing Debian onto Your BeagleBone Black*, introduces how to install Debian onto your BeagleBone. There are two ways to boot the BeagleBone and run the OS.

Chapter 2, *Installing and Configuring Multimedia Server Software*, serves as an installation guide for the software that will be used to store the streamed video and to serve up both the audio and video files to any device on the network, either BB, computers, or tablets/phones.

Chapter 3, *Installing and Configuring Network Monitoring Software*, acts as an installation guide for the software that will be used to monitor the traffic on your local network.

Chapter 4, *Installing and Setting Up a BeagleBone RAID System*, acts as an installation guide for the software that will be used to create a RAID array out of the partitions that you will create on your USB-connected drives.

Chapter 5, *Streaming Videos*, will show you how to set up both live and recorded video streaming, using a web-based application.

Chapter 6, *Setting Up a Wireless Access Point*, shows you how to install and set up a wireless access point or WAP on your BeagleBone system.
In this chapter, we are going to set up both live and recorded video streaming, using a web-based application. What this means for the user is that they can access either their recorded videos or a live stream from their home IP video camera, from their smartphone, or their tablet while being logged on to their home network.

In order to do this, we have to install some additional software on our BeagleBone. The instructions explained in this chapter are based on an excellent tutorial from HowtoForge:


**Installing MySQL5**

The first package that we are going to install is called **MySQL5**.

![All these instructions assume that you have logged in as Root and your network IP address is 192.168.10.127.](image)

To do this, we enter the following command into a terminal (or SSH) window:

`apt-get install mysql-server mysql-client`

Once the software has finished installing, the first thing we have to do is set up a root user password.
In the terminal window, enter `p`. The following message will appear in the window, as shown in the following screenshot:

**New password for the MySQL "root" user:**

![Screenshot of terminal window showing password input]

You will then be asked to repeat the password with the following message, as shown here:

**Repeat password for the MySQL "root" user:**

![Screenshot of terminal window showing password input]

That's it; we are done with MySQL for now.

## Installing Lighttpd

The next piece of software that we will be installing is the actual web server software, called Lighttpd. This is a server package that has been optimized for embedded applications. It has all the functionality that we will need and consumes relatively few computing resources. For a better reference, refer to [http://www.lighttpd.net/](http://www.lighttpd.net/)

"With a small memory footprint compared to other web-servers, effective management of the cpu-load, and advanced feature set (FastCGI, SCGI, Auth, Output-Compression, URL-Rewriting and many more) lighttpd is the perfect solution for every server that is suffering load problems. And best of all it's Open Source licensed under the revised BSD license."

To bring about the installation, we enter the following command in the terminal window:

```bash
apt-get install lighttpd
```

## Disabling Cloud9 services

To prevent the Cloud9 IDE from interfering with your web server, you must disable these services:

```bash
systemctl disable cloud9.service
systemctl disable bonescript.service
systemctl disable bonescript.socket
systemctl disable bonescript-autorun.service
```

Even with these services disabled, you will still be able to use the Cloud9 IDE to develop code.
If you now enter the IP of the BeagleBone, you will see the Lighttpd placeholder page:

You should replace this page with your own web pages as soon as possible.

Unless you changed its configuration, your new server is configured as follows:

- Configuration files can be found in /etc/lighttpd. Please read /etc/lighttpd/conf-available/README file.
- The DocumentRoot, which is the directory under which all your HTML files should exist, is set to /var/www.
- CGI scripts are looked for in /var/lib/cgi-bin, which is where Debian packages will place their scripts. You can enable cgi module by using command "lighty-enable-mod cgi".
- Log files are placed in /var/log/lighttpd, and will be rotated weekly. The frequency of rotation can be easily changed by editing /etc/logrotate.d/lighttpd.

If you get Error 404, then you've forgotten to perform `apt-get update` as I suggested. So, you can either do the update now or try using `index.html` instead.

**Installing PHP5**

Our web-based file browser is a PHP application. Therefore, our next task is to install PHP5 and PHP-FPM. PHP-FPM is a daemon process that runs a FastCGI server on port 9000. The init script for this application is stored at /etc/init.d/php5-fpm/php.ini.

To install these two programs, enter the following command:

```
apt-get install php5-fpm php5
```
Configuring Lighttpd and PHP5

To enable PHP5 in Lighttpd, we must modify /etc/php5/fpm/php.ini and uncomment the line `cgi.fix_pathinfo=1`:

`;http://php.net/cgi.fix-pathinfo
;cgi.fix_pathinfo=1`

Remove the semicolon present at the beginning of the previous line.

The Lighttpd configuration file for PHP `/etc/lighttpd/conf-available/15-fastcgi-php.conf` is suitable for use with spawn-fcgi. However, we want to use PHP-FPM; therefore, we create a backup of the file (named `15-fastcgi-php-spawnfcgi.conf`) and modify `15-fastcgi-php.conf`, as follows:

1. cd /etc/lighttpd/conf-available/
2. cp 15-fastcgi-php.conf 15-fastcgi-php-spawnfcgi.conf
3. nano 15-fastcgi-php.conf

We then add the following code to the config file:

```plaintext
# /usr/share/doc/lighttpd/fastcgi.txt.gz
# http://redmine.lighttpd.net/projects/lighttpd/wiki/Docs:Configuration
Options#mod_fastcgi-fastcgi
## Start an FastCGI server for php (needs the php5-cgi package)
fastcgi.server += ( ".php" =>
    ((
        "socket" => "/var/run/php5-fpm.sock",
        "broken-scriptfilename" => "enable"
    ))
)
```

Lighttpd has a number of different modules that can be enabled by running the `lighttpd-enable-mod <module>` command.

To enable the `fastcgi` configuration, run the following command:

`lighttpd-enable-mod fastcgi`

As a result of running the command, the following messages are displayed in the terminal:

```
Available modules: auth accesslog cgi evasive evhost expire fastcgi
fly-streaming no-www proxy rrdtool simple-vhost ssi ssl status userdir
usertrack fastcgi-php-spawnfcgi fastcgi-php debian-doc
Already enabled modules:
Enabling fastcgi: ok
```
Run /etc/init.d/lighttpd force-reload to enable changes

We proceed further by running the following command:

```
lighttpd-enable-mod fastcgi-php
```

The messages displayed in the terminal will look like the following:

```
Available modules: auth accesslog cgi evasive evhost expire fastcgi flv-streaming no-www proxy rrdtool simple-vhost ssi ssl status userdir usertrack fastcgi-php-spawnfcgi fastcgi-php debian-doc
Already enabled modules: fastcgi
Enabling fastcgi-php: ok
```

This creates the symlinks `/etc/lighttpd/conf-enabled/10-fastcgi.conf`, which points to `/etc/lighttpd/conf-available/10-fastcgi.conf`, and `/etc/lighttpd/conf-enabled/15-fastcgi-php.conf`, which points to `/etc/lighttpd/conf-available/15-fastcgi-php.conf`.

Now, enter the following command:

```
cd /etc/lighttpd/conf-enabled
ls -l
```

You will see the following messages in the terminal:

```
10-fastcgi.conf -> ../conf-available/10-fastcgi.conf
fastcgi-php.conf -> ../conf-available/15-fastcgi-php.conf
```

We then force a reload, as follows:

```
root@beaglebone: /etc/init.d/lighttpd force-reload
```

This causes the terminal to display the following:

```
* Reloading web server configuration lighttpd                           [ OK ]
```

**Testing PHP5**

The document root of the default website is `/var/www`. We will now create a small PHP file (`info.php`) in this directory and call it in a browser:

```
nano Info.php
<?php
phpinfo();
?>
```
The following screenshot shows how Info.php looks in a browser:

![Info.php Screenshot 1](image1)

The file will display lots of useful details about our PHP installation, such as the installed PHP version, as shown here:

![Info.php Screenshot 2](image2)
Streaming Videos

PHP5 is working, and it's working through FPM/FastCGI, as shown in the Server API line. If you scroll down further, you will see all the modules that are already enabled in PHP5. MySQL is not listed here, which means that we don't have MySQL support in PHP5 yet.

**Setting up MySQL support in PHP5**

To get MySQL support in PHP, we can install the `php5-mysql` package. It's a good idea to install some other PHP5 modules as well, as you might need them for your applications. You can search for available PHP5 modules as follows:

```
root@beaglebone:/apt-cache search php5
```

You will see the following messages being displayed:

- `php5-curl` - CURL module for PHP5
- `php5-dbg` - Debug symbols for PHP5
- `php5-dev` - Files for PHP5 module development
- `php5-gd` - GD module for PHP5
- `php5-gmp` - GMP module for PHP5
- `php5-ldap` - LDAP module for PHP5
- `php5-mysql` - MySQL module for PHP5
- `php5-odbc` - ODBC module for PHP5

Pick the modules you need and install them. These are the ones that I installed:

```
apt-get install php5-mysql php5-curl
```

**Xcache** is a PHP opcode cacher for caching and optimizing PHP intermediate code. Xcache can be installed as follows:

```
apt-get install php5-xcache
```

Reload the PHP-FPM service:

```
/etc/init.d/php5-fpm reload
```

Now, reload `http://192.168.10.127/info.php` in your browser and scroll down to the modules section again.
You should now find lots of new modules here, including the `mysql` module:

![MySQL Support](image)

**Creating your own home page**

Now that we have the required software installed and configured, it is time to install our own software. We will start by installing our home page, called `index.html`, in the directory where Lighttpd expects to find it.

The directory name is `/var/www`, as shown here:

![Directory Listing](image)
Streaming Videos

The software uses a modified version of wfb.php, which is a web-based file browser written in PHP. It is available for download at http://cgdave.github.io/webfilebrowser/.

There are three subdirectories:

- The wfbtrash directory is used by wfb to store deleted files. This directory must have write privileges enabled or the delete button on the form will not be available.
- The wfbimages directory is where the icons for the form are stored. If you want to use your own icons, this is where you will put them.
- The video directory is where we will store all our pre-recorded videos. This directory is the cleanest way of keeping our video files separate from the program files.

Creating two modified PHP files

What I have done is created two versions of wfb.php and changed the base directory information in each file.

The following is the code for wfb_video.php:

```php
$basedir = "video";// Base directory = custom directory
$filelinks = true;// Links on files enabled
$basevirtualdir = "video" //video directory
```

File links are enabled so that when the user clicks on a given link, the video will automatically start playing, assuming that the required codecs are installed in the browser.

The following is the code for wfb_files.php:

```php
$basedir = "/";   // Base directory = custom directory
$filelinks = true;// Links on files enabled
$basevirtualdir = "/"; // root directory
```

In this case, the root directory of the file system can be accessed. There doesn't appear to be any way to pass directory information to the PHP code, so we need a different wfb.php file for each location.
When the user enters the location of the media server into their browser, they will be directed to `index.html` on the BeagleBone (in my case, 192.168.10.127).
Streaming Videos

When the user presses the **Recorded Video** button, the following custom file browser screen appears:

![File Browser Screenshot]

The **File Manager** interface is pretty much the same, except that the browser is now pointing to the root directory of the BeagleBone. This version of the PHP file should have password access enabled in order to keep normal users from damaging things.
As we can see in the following screenshot, we have **Root File System Access**:
In both cases, there is an additional area at the bottom of the page that allows the user to do many of the things that they can do in a graphical user interface based system, including uploading files from the Internet.

![File Management Interface]

The user is also able to access the IP cameras that are connected to their local network. In this case, the HTML code of `index.html` must be modified so that it points to the user's camera.

In this case, the IP address is `192.168.10.110` and the default page to view is `jview.htm`. The reader's IP address and default page will probably be different. This information is for a D-Link DCS-933L camera, which requires the user to log in before accessing the video.

```html
<h2>For Realtime Video from your IP Camera Press:</h2>
<p>
<a href="http://192.168.10.110/jview.htm" target="_parent">button>IP Camera 1</button>
</a>
</p>
```

Most, if not all, IP cameras have a built-in web server with a configuration page. What you can configure varies from camera to camera, but the most important part is the camera's IP address. In the following screenshot, you will see that I have configured the camera to use a static IP address. This is so that our home page will always be able to find the camera. My network uses the IP addresses `192.168.10.xxx`, so I set up the camera at `192.168.10.110` in the LAN settings. My default gateway is `192.168.10.1`; yours may be different.
When you access the camera, you must first enter a username and password. The username is `admin` and the password field is blank. You should probably change this in the settings menu.

As you can see in the following screenshot, this particular camera has a number of features that can be SETUP for security monitoring:
Streaming Videos

Once this is done, the user can access the camera from the web application by clicking on the **LIVE VIDEO** button at the top of the screen, as shown here:

There are several buttons at the bottom of the video's window, which allows the user to (digitally) zoom in and out, turn the sound on and off, and enable **Night Mode**. In night mode, the camera will automatically detect when the light level drops and will turn on the Infrared LEDs. Other IP cameras will, of course, have different controls, such as pan, tilt, and optical zoom.
Confi guring a streaming video

In order to provide video streaming from a USB device in our multimedia setup, we have to fi rst install some software that will allow the BeagleBone to act as a server. The software that I chose to use is called MJPG Streamer. If you want to add more functionality than I have here, the instructions are available at http://shrkey.com/installing-mjpg-streamer-on-beaglebone-black/ and at Sourceforge (http://sourceforge.net/projects/mjpg-streamer/).

The fi rst thing we have to do is install the tools and dependencies that we will need to compile the software.

I make it a habit to always do an apt-get update to make sure that all the repositories are up to date. The following commands will install the required tools and dependencies. If your tools are already installed and up to date, they will not be overwritten. All the commands assume that you are logged in as root:

1. Run the following command:
   
   `apt-get install g++ curl pkg-config libv4l-dev libjpeg-dev build-essential libssl-dev vim cmake`

2. Then, run this:
   
   `apt-get install imagemagick`

3. Next, we have to get the code from the website I mentioned earlier:
   
   `wget https://github.com/shrkey/mjpg-streamer/raw/master/mjpg-streamer.tar.gz`

4. Now that we have the tar ball, we can expand it in the directory of our choice:
   
   `tar -xvf ./mjpg-streamer.tar.gz`

5. The next step is to compile the mjpg-streamer code:
   
   `cd mjpg-streamer`

   `make USE_LIBV4L2=true`

   `make install`

6. Now, from the directory that we compiled in, we can run a quick test by typing the following:
   
   `/mjpg_streamer -i "./input_uvc.so" -o "./output_http.so -w ./www"`
Streaming Videos

This will start the server running from port 8080 of the BeagleBone. The `input_uvc.so` file is the input device driver used by `mjpg_streamer`, and `output_http.so` is the output device driver. The final argument is the directory, where the web page is located, that the video should be sent to.

You will see a screen much like what is shown in the following screenshot; the actual messages shown will depend on the capabilities of your particular camera:

```
MJPG Streamer Version: svn rev:
  i: Using V4L2 device.: /dev/video0
  i: Desired Resolution: 640 x 480
  i: Frames Per Second.: 5
  i: Format............: MJPEG
  o: www-folder-path...: ./www/
  o: HTTP TCP port.....: 8080
  o: username:password.: disabled
  o: commands..........: enabled
```

If you now go to the web interface described earlier and click on the Streaming Video button, you should see the output of your webcam.

The following screenshot is from a cheap dollar store camera (it came with an exercise video for about $3.00):
Summary

In this chapter, we installed and set up our web server software as well as support for PHP scripting and SQL.

We also installed the web page that will allow you to view both recorded video on your file server and live video from IP and USB cameras. The web page also features a web-based file browser, similar to those available with graphical user interfaces.

In the next chapter, we will be setting up a Wi-Fi server so that you can connect to your multimedia server from anywhere within its range.
Where to buy this book

You can buy Building Networks and Servers Using BeagleBone from the Packt Publishing website.

Alternatively, you can buy the book from Amazon, BN.com, Computer Manuals and most internet book retailers.

Click here for ordering and shipping details.