

## PiBox - Feature #220

### Create a web server to feed cam video

22 Aug 2013 18:39 - Hammel

<b>Status:</b>	Closed	<b>Start date:</b>	22 Aug 2013
<b>Priority:</b>	Immediate	<b>Due date:</b>	
<b>Assignee:</b>	Hammel	<b>% Done:</b>	100%
<b>Category:</b>	04 - Root File System	<b>Estimated time:</b>	0.00 hour
<b>Target version:</b>	1.0 - Atreides		
<b>Severity:</b>	03 - Medium		
<b>Description</b>			
<p>I need a web server package added to buildroot and I need to install my small web files for it to server up ffmpeg output via crtmpserver.</p> <p>I need to list exactly what has to happen for this to work too.</p>			

#### Associated revisions

---

##### Revision 3ed195ba - 25 Aug 2013 16:05 - Hammel

RM #220: Fix logs directory. Make pkg target depend on main build stamp .\$(MONKEY\_T).

##### Revision 712b05e2 - 30 Aug 2013 13:32 - Hammel

RM #220: Added init script to monkey that uses a stamp file to determine if it will start or not.

##### Revision 4f8ef885 - 30 Aug 2013 13:48 - Hammel

RM #220: Added init script that utilizes a stamp file to determine if the rtmp server should start or not.

##### Revision 35b1d75b - 30 Aug 2013 13:54 - Hammel

RM #220: Add shortcut targets for pkg, pkg-clean.

##### Revision 2ddba13a - 08 Oct 2013 20:31 - Hammel

RM #220: Enabled PHP support in rootfs.

#### History

---

##### #1 - 23 Aug 2013 10:59 - Hammel

- Status changed from New to In Progress

- % Done changed from 0 to 10

Possible web servers:

1. [Monkey](#)
2. Busybox has an httpd server
3. [Mongoose](#)

4. [lighttpd](#)
5. [tinyhttpd](#)
6. [boa](#)

lighttpd, tinyhttpd and boa are available in Buildroot. tinyhttpd can be eliminated because it has no online presence and was just a student project, which means it may not be seeing any active development.

Monkey has a C-API, as does Mongoose. The C-API is appealing in Monkey (less overhead with scripting languages). I've used Mongoose in the past but it's not actively developed. It is very small and doesn't fork. But it might be too small for PiBox's needs. Mongoose was easy enough to learn by reading the code and an example. Monkey has some documentation and a nice web site but I don't see much online about the API.

boa is stable and widely used. Linux Journal has a [short article](#) on using it.

I think what I need to define is what I want to display (images, layout, etc) and determine how I want to do that before picking a server.

## **#2 - 23 Aug 2013 22:17 - Hammel**

- % Done changed from 10 to 20

Assuming I enable PHP in Buildroot and monkey can handle it, I could create a simple RESTful API, as in [this tutorial](#) or [this one](#). For choosing a resource the process would look like this.

- PHP builds page based on
  - Webcam availability
  - USB SD card availability (video files)
- select a file (or webcam)
- Respond with Javascript that connects to daemon to launch ffmpeg
- Start JWPlayer to crtmpserver
- On page exit/close, Javascript sends message to daemon to kill ffmpeg.

Pretty straightforward. I'll have to learn just a little Javascript.

Buildroot's php is 5.3.x, which is sufficient for anything I might want to do.

## **#3 - 24 Aug 2013 16:46 - Hammel**

- % Done changed from 20 to 30

After playing with a number of the available httpd options I settled on Monkey due to it's apparent ease of configuration to enable PHP (re: CGI) support.

I've created a [cross build for monkey](#) and have tested the installed opkg on the target. It comes up enough to know its basically working though the default configuration needs to be adjusted. Configurations are under /etc/monkey, with monkey.conf, plugins.load and plugins/cgi/cgi.conf the most likely candidates for updates.

Once the configuration is fixed I can try writing a simple PHP-based test to make sure it's serving up PHP-based content.

## **#4 - 24 Aug 2013 18:02 - Hammel**

- % Done changed from 30 to 40

The default config essentially works. All I have to do is place my php files under /home/httpd/monkey/php. It doesn't work from the top level directory. There isn't anything in the /etc/monkey/plugins/cgi/cgi.conf that specifies php has to go under the php directory, but that appears to be the case.

I need to change the default config to use port 80 now and force a redirect from index.html to php/index.php and that should be the start of the web server support on PiBox.

#### #5 - 24 Aug 2013 23:12 - Hammel

Some sample javascript for use with HTML4: <http://corehtml5canvas.com/code-live/>

#### #6 - 25 Aug 2013 12:33 - Hammel

- File *crtmpserver.html* added

- File *camff.sh* added

Testing complete feed of webcam, through ffmpeg to crtmpserver across to web browser.

**Setup** under pibox:/home/httpd/monkey

1. Attached

1. *crtmpserver.html* - pulls feed from ffmpeg

2. jwplayer directory

I don't know if this works for tablets, however. I tested on my x86 Fedora 16 desktop.

#### **Requirements**

1. `chown -R nobody.nobody /home/httpd/monkey` (or alternative non-root user)

#### **Status**

The streaming works, but the image is about 4 seconds behind. That's unusable. I need to find a way to speed things up. One possible solution is to use the options I discovered in RM #192: `--intra` and `-qscale 2`

#### #7 - 27 Aug 2013 08:46 - Hammel

- File *camff.sh* added

- File *crtmpserver.lua* added

- File *crtmpserver2.html* added

- % Done changed from 40 to 50

I now have the best working version of this I think I can get for the prototype. I've tweaked how ffmpeg reads from /dev/video and feeds crtmpserver (see updated *camff.sh*). This works fairly well with occasional hiccups from ffmpeg. This may be due to the old version of ffmpeg and I certainly need to try to cross compile an updated ffmpeg to test that theory.

*crtmpserver* seems to work fine (see *crtmpserver.lua*). It doesn't seem to have any crash problems that I can see. It only accepts flv input - the mpegts support is apparently unavailable (at least from the open source version). I tested it's use with flv to a browser with JWPlayer and that part seems okay from the *crtmpserver* side. I've not tested HTML5 support yet.

As for JWPlayer, I'm a bit disappointed. It works great initially with only a 1/2 second lag. This is acceptable for the webcam. However, if you leave it running for a period of time (less than 10 minutes) it starts to lag. I didn't time how long the lag eventually gets but its significant, at times getting as high as 5 seconds. If I pause the player and then press play again, it's immediately back to a 1/2 second lag. So this certainly appears to be an issue with JWPlayer and not *crtmpserver* or ffmpeg. However, I need to find another player to verify this. I don't know what else I can use - perhaps vlc. I need something that can play the stream by circumventing the JWPlayer javascript on the web site by going directly to *crtmpserver*.

The bigger issue I'm having with testing this is network stability. The network seems to die after a period of time. Sometimes quickly, other times it

takes a while. I let it run for hours last night - it eventually died overnight at some point because my ssh sessions went away. Pinging it this morning shows it's back and ssh's work again. So something is not right with either my wireless routers (highly likely) or with PiBox's wireless adapters (I've tried the two that have AP support and both appear to have the same problem, which seems odd).

For the proof of concept what I have now is sufficient. I need to do the following:

1. Put together the web server files in either an opkg or in the PiBox skeleton. I think an opkg seems more appropriate. A web page that selects the webcam would call a PHP script to launch ffmpeg before displaying the JWPlayer content. There needs to be javascript that can connect to a PHP page that shuts down the webcam when the page closes on the client side. That way ffmpeg is only running when the webcam is being viewed, and not using up system resources running when its not being used.

2. Add an init script to the Monkey opkg that tests for a run-stamp. If the run-stamp is found, monkey starts. If not, it doesn't. This will allow me to create a web based front end to configure features in PiBox.

3. Add an init script to the crttmpserver opkg, also with a run-stamp.

Once those are done, this issue can be closed for the first release.

#### #8 - 30 Aug 2013 09:48 - Hammel

- % Done changed from 50 to 60

opkg build created: <https://gitorious.org/piboxwww>

#### #9 - 30 Aug 2013 13:34 - Hammel

- % Done changed from 60 to 70

Added init script that utilizes a stamp file to determine if it will start or not to the Monkey opkg.

#### #10 - 30 Aug 2013 13:49 - Hammel

- % Done changed from 70 to 80

Added init script that utilizes a stamp file to determine if it will start or not to the crttmpserver opkg.

The three (monkey, crttmpserver and piboxwww) now need to be tested to make sure they install cleanly and come up correctly.

#### #11 - 08 Oct 2013 20:07 - Hammel

- Status changed from In Progress to Closed

- % Done changed from 80 to 100

Integration test of crttmpserver, monkey w/piboxwww and piboxd was run today at home and it worked perfectly.

This issue can be closed.

#### Files

crttmpserver.html	1.63 KB	25 Aug 2013	Hammel
camff.sh	453 Bytes	25 Aug 2013	Hammel
camff.sh	2.77 KB	27 Aug 2013	Hammel
crttmpserver.lua	8.38 KB	27 Aug 2013	Hammel

