
derekmolloy.ie[Home](#)[Course Notes](#)[Embedded Systems](#)[General](#)[Blog](#)[Other](#)

Custom Kernel and Ångström Distribution Recipe (SPI/CAN bus Support)

[Home](#) / [Blog](#) / [Beaglebone](#) / Custom Kernel and Ångström Distribution Recipe
(SPI/CAN bus Support)

[< Previous](#) [Next >](#)

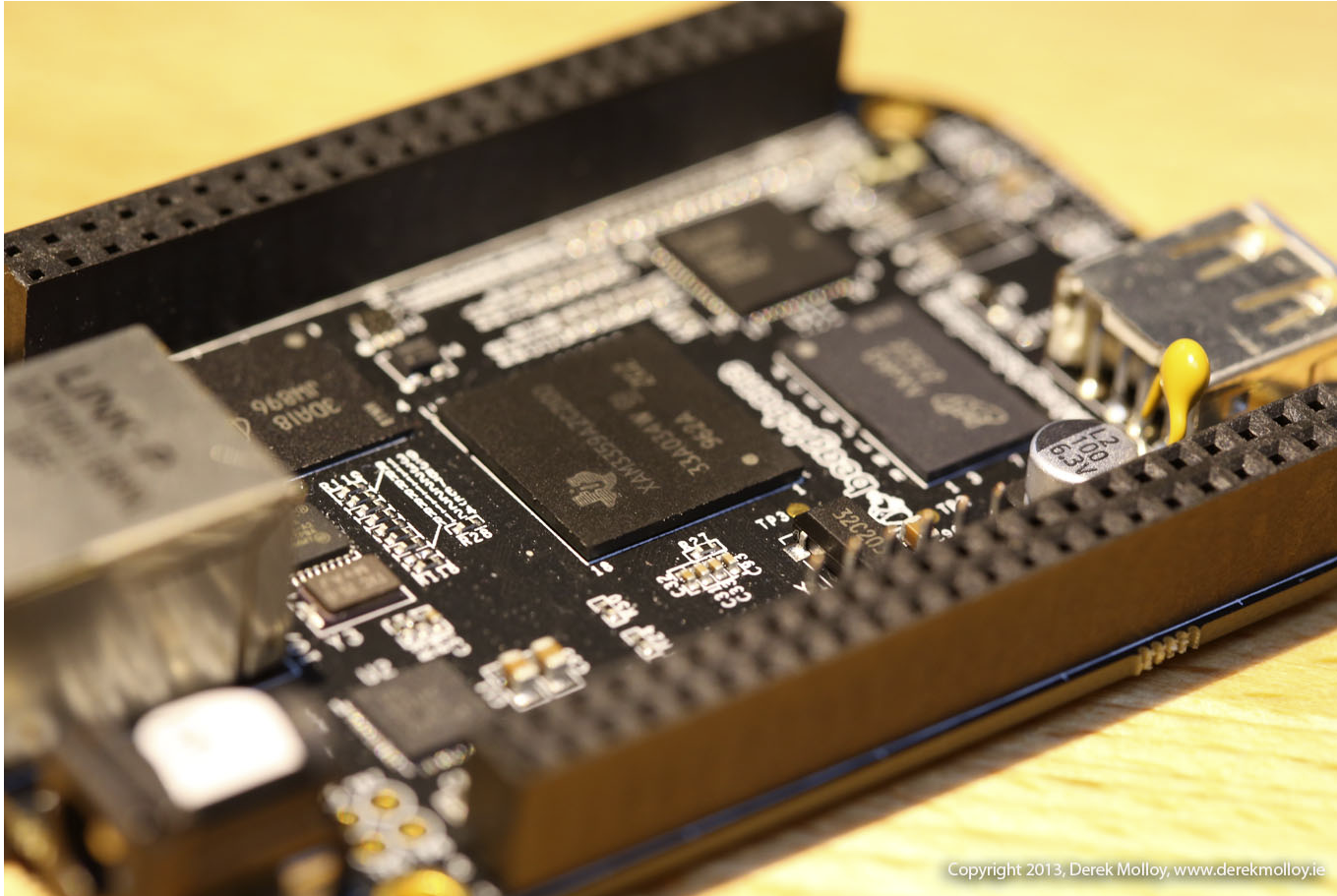
Derek Molloy



Search ...

Tags

Analog Discovery angstrom
avconv **beaglebone**
beaglebone black
bitbake build building c++ C920
connman cpu curl disk distribution
easydriver ffmpeg flash Flip-Flop
git gparted https iso **Java** kernel



Copyright 2013, Derek Molloy, www.derekmolloy.ie

Custom Kernel and Ångström Distribution Recipe (SPI/CAN bus Support)

I have already posted on how we can build a vanilla Angstrom distribution from source at this post: [Building Angstrom for Beaglebone from Source](#). This post looks at how we can add SPI to the kernel and rebuild the Linux kernel and Angstrom distribution. The post isn't specific to SPI, so it may be that you are looking to make a different change to the distribution.

Initial Setup

LED LEDs linux live-cd Logic
Analyzer nmap opencv portscan re-
size resize RTP source stepper
motor UDP vdi Video virtualbox VLC
Wordpress x264

Categories

- › Analog
- › Beaglebone
- › Blog
- › Digital Electronics
- › Embedded Systems
- › General
- › Linux
- › Main Blog
- › Raspberry PI
- › Tools
- › Uncategorized

If you have not done these steps already as explained in the previous post you should:

```
1 git clone git://github.com/angstrom-distribution/setup-scripts.git
2 cd setup-scripts
3 MACHINE=beaglebone ./oebb.sh config beaglebone
4 MACHINE=beaglebone ./oebb.sh update
```

This should have you at the same point as the previous post and ready to go. If you receive any errors at this point then please go back to that post.

Really useful guide on using bitbake for open embedded and how everything works:

http://docs.openembedded.org/usermanual/usermanual.html#chapter_using_bitbake_and_oe

This is a substantial document; however, if you ever want to understand what is going on and how to create your own recipes you need to read this.

Back up the Default Kernel Configuration

Because we are about to replace the default configuration for the Beaglebone it is useful to back it up:

```
1 cd ~/beaglebone/setup-scripts/sources/meta-ti/recipes-kernel/linux/linux-ti33x-psp-3.2/
2
3 cp defconfig defconfig_backup
```

This means that we can always replace it if something goes wrong.

Modify the Kernel Configuration on the Beaglebone using Menuconfig

We can make modifications to the kernel configuration as follows:

```
1 cd ~/beaglebone/setup-scripts/build/tmp-anstrom_v2012_12-eglibc/work/beaglebone-
2 angstrom-linux-gnueabi/linux-mainline-3.8.11-r23a/package/usr/src/kernel
3
4 make menuconfig
```

The directory after build will depend on what your current build directory is. This directory appeared because I

Recent Posts

› Analog Discovery: Getting Started with the Analog and Digital Tools

› Analog Discovery: Getting Started with the Logic Analyzer and Digital Pattern Generator

› Beaglebone: Controlling the on-board LEDs using Java

› Beaglebone: Controlling the on-board LEDs using C++

› Running Java Applications on the Beaglebone Black

have already built the kernel in the last post: [Building Angstrom for Beaglebone from Source](#)

However, if you haven't built a distribution already you could type:

```
1 . ~/.oe/environment-angstromv2012.12
2 bitbake virtual/kernel -c configure
3 ***or***
4 bitbake cloud9-image -c configure
```

depending on your target distribution.

When you execute make menuconfig you should see an interface like in Figure 1. In this case I am going to enable Debug support for SPI drivers. You could be doing this for another reason; for example in Figure 3 I am enabling CAN bus support.



Figure 1: menuconfig user interface

Go to -> Device Driver -> SPI support and then enable "Debug support for SPI drivers" by pressing Y

Archives

- > January 2014

- > December 2013

- > November 2013

- > October 2013

- > July 2013

- > June 2013

- > May 2013

- > April 2013

Professional Secure
File Sharing

TRY IT FREE FOR 30 DAYS

CITRIX
ShareFile



Figure 2: Enable “Debug support for SPI drivers” by pressing ‘Y’

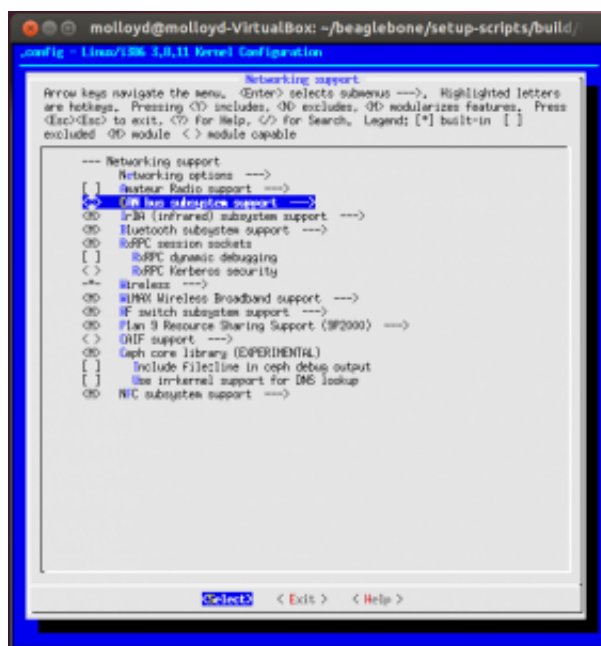


Figure 3: Enabling ‘CAN bus subsystem support’ (Networking Support -> CAN bus subsystem support)

If you were doing what I was doing your screen should appear as in Figure 2. Exit by pressing Exit back to the main menu and then exit completely. You can then copy the config file to your linux-ti33x-psp-3.2 directory so that it won't be lost on a bitbake clean:

```
1 cp .config ~/beaglebone/setup-scripts/sources/meta-ti/recipes-kernel/linux/linux-ti33x-
2 psp-3.2/defconfig
3 cat defconfig | grep CONFIG_SPI_SPIDEV
```

This should come back with CONFIG_SPI_SPIDEV=y

You can then build your distribution with the custom kernel settings you have made by doing something like:

```
1 cd ~/beaglebone/setup-scripts/
2 bitbake cloud9-image
```

to build the standard recipes – however if you want to change the recipe have a look at the next section.

Changing the Angstrom Beaglebone build Recipes

I was having problems with the latest release with nodejs and it is not something I needed for the project I had planned. So the question arose – how do I change the Angstrom build recipe?

Here is one way – it is probably not very elegant:

```
1 cd ~/beaglebone/setup-scripts/sources/meta-ti/recipes-misc/images/
2 cp cloud9-image.bb derek-image.bb
```

I have build my version from the cloud9-image.bb file, except I have removed nodejs and anything else that had nodejs as a dependency. Here is the content of my derek-image.bb file:

```
1 # Image without cloud9 ide but with hw tools installed
2
3 require ti-hw-bringup-image.bb
4
5 FATPAYLOAD = "${datadir}/beaglebone-getting-started/*"
6
7 ROOTFSTYPE_beaglebone = "ext4"
8
```

```
9 IMAGE_INSTALL += " \  
10     systemd-analyze \  
11     mpd \  
12     packagegroup-sdk-target \  
13     vim vim-vimrc \  
14     procps \  
15     beaglebone-tester \  
16     screen minicom \  
17     git \  
18     led-config \  
19     opencv-dev \  
20     cronie-systemd ntpdate \  
21     nano \  
22     minicom \  
23     hicolor-icon-theme \  
24     gateone \  
25     tar \  
26     gdb gdbserver \  
27     mplayer2 \  
28     tslib-tests tslib-calibrate \  
29     iproute2 canutils \  
30     connman-tests \  
31     rsync \  
32     e2fsprogs-mke2fs \  
33     cpufreq-tweaks \  
34     dosfstools \  
35     parted \  
36 "  
37  
38 export IMAGE_BASENAME = "Derek-IDE"
```

Now, to build this recipe:

```
1 cd ~/beaglebone/setup-scripts/  
2 . ~/.oe/environment-angstromv2012.12  
3 bitbake derek-image
```

Note: do not add .bb to the end of the derek-image, even though that is the name of the file.

This should be enough for you to build your own custom distribution. You should then find the build in the directory like “~/beaglebone/setup-scripts/build/tmp-angstrom_v2012_12-eglibc/deploy/images/beaglebone”, which will change according to your build etc. Here is the directory listing:


```

1 molloyd@molloyd-VirtualBox:~/beaglebone/setup-scripts/build/tmp-angstrom_v2012_12-eglib
2 total 524876
3 drwxrwxr-x 2 molloyd molloyd 4096 May 5 13:15 .
4 drwxrwxr-x 3 molloyd molloyd 4096 Apr 21 16:05 ..
5 -rw-r--r-- 1 molloyd molloyd 130297073 May 5 13:14 Angstrom-Derek-IDE-eglibc-ipk-v2012
6 -rw-r--r-- 1 molloyd molloyd 78804972 May 5 13:15 Angstrom-Derek-IDE-eglibc-ipk-v2012
7 lrwxrwxrwx 1 molloyd molloyd 63 May 5 13:15 Derek-IDE-beaglebone.tar.gz -> Angs
8 lrwxrwxrwx 1 molloyd molloyd 63 May 5 13:15 Derek-IDE-beaglebone.tar.xz -> Angs
9 -rw-rwxr-x 1 molloyd molloyd 1535 Mar 25 2011 mkcard.sh
10 lrwxrwxrwx 1 molloyd molloyd 22 May 3 22:37 MLO -> MLO-beaglebone-2013.04
11 lrwxrwxrwx 1 molloyd molloyd 22 May 3 22:37 MLO-beaglebone -> MLO-beaglebone-20
12 -rwxr-xr-x 1 molloyd molloyd 99904 May 3 22:37 MLO-beaglebone-2013.04
13 -rw-rw-r-- 1 molloyd molloyd 37466149 May 3 22:49 modules-3.8.11-r23a-beaglebone.tgz
14 -rw-rw-r-- 1 molloyd molloyd 294 May 5 13:08 README_-_DO_NOT_DELETE_FILES_IN_THI
15 -rwxr-xr-x 1 molloyd molloyd 378948 May 3 22:37 u-boot-beaglebone-2013.04-r0.img
16 lrwxrwxrwx 1 molloyd molloyd 32 May 3 22:37 u-boot-beaglebone.img -> u-boot-bea
17 lrwxrwxrwx 1 molloyd molloyd 32 May 3 22:37 u-boot.img -> u-boot-beaglebone-201
18 lrwxrwxrwx 1 molloyd molloyd 48 May 3 22:49 uImage -> uImage-3.8.11-r23a-beagle
19 -rw-r--r-- 1 molloyd molloyd 4218960 May 3 22:49 uImage-3.8.11-r23a-beaglebone-20130
20 lrwxrwxrwx 1 molloyd molloyd 48 May 3 22:49 uImage-beaglebone.bin -> uImage-3.8

```

You can then deploy this to your SD card using the instructions in my post: [Building Angstrom for Beaglebone from Source](#)

Custom patches for the Beaglebone kernel build and distribution:

All of the beaglebone specific patches are in the directory: `~/beaglebone/setup-scripts/sources/meta-ti/recipes-kernel/linux/linux-ti33x-psp-3.2/beaglebone`

For example, the patches related to the LCD4 Cape are:

```

1 -rw-rw-r-- 1 molloyd molloyd 1494 Apr 20 17:59 0092-beaglebone-add-support-for-LCD4-r
2 -rw-rw-r-- 1 molloyd molloyd 9470 Apr 20 17:59 0093-beaglebone-update-cssp_camera-dri
3 -rw-rw-r-- 1 molloyd molloyd 3231 Apr 20 17:59 0094-beaglebone-make-LCD4-Rev-A1-displ
4 -rw-rw-r-- 1 molloyd molloyd 4271 Apr 20 17:59 0095-beaglebone-convert-LCD4-to-16-bit
5 -rw-rw-r-- 1 molloyd molloyd 4357 Apr 20 17:59 0096-beaglebone-camera-cape-sensor-ori
6 -rw-rw-r-- 1 molloyd molloyd 26765 Apr 20 17:59 0097-beaglebone-improve-performance-of
7 -rw-rw-r-- 1 molloyd molloyd 14360 Apr 20 17:59 0098-beaglebone-Remove-optimization-fr
8 -rw-rw-r-- 1 molloyd molloyd 928 Apr 20 17:59 0099-beaglebone-fix-DOWN-button-on-LCD

```


You can step back one directory and view the bitbake file at: `~/beaglebone/setup-scripts/sources/meta-ti/recipes-kernel/linux/linux-ti33x-psp_3.2.bb`

nano is useful for this as there are a lot of pages:

```
1 ~/beaglebone/setup-scripts/sources/meta-ti/recipes-kernel/linux/linux-ti33x-psp_3.2.bb
2 nano linux-ti33x-psp_3.2.bb
```

At the bottom of a long list can see the 102 beaglebone patches (use CTRL-V to quickly page down) on this date. You would be able to modify the `linux-ti33x-psp_3.2.bb` and add in a custom patch or remove a patch if you felt it was causing difficulty. The process to build is then the same as before.

 Recommend this on Google

By Derek | May 5th, 2013 | Beaglebone, Blog | 10 Comments

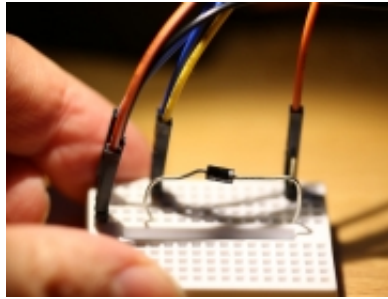
Share This Story, Choose Your Platform!

About the Author: Derek



Dr. Derek Molloy is a Senior Lecturer in the School of Electronic Engineering, Faculty of Engineering & Computing at Dublin City University, Ireland. He lectures in Object-oriented Programming, Embedded Systems, 3D Computer Graphics and Digital Electronics at postgraduate and undergraduate levels. This is his personal blog site.

Related Posts



10 Comments



Joerg May 29, 2013 at 12:14 pm - Reply

here is the same mistake 😊



Derek May 29, 2013 at 1:01 pm - Reply

Thanks Joerg, Fixed.



Yves June 16, 2013 at 7:49 pm - Reply

Hi Derek

I read and followed your article on setting up NTP on the BBB and it worked perfectly. So I read this one as a potential solution to my next problem. It probably would be but there might be a simpler solution. I have flashed my BBB with the latest release from BeagleBoard and had no problem doing it. But when I needed a custom Angstrom distribution, I went to <http://narcissus.angstrom-distribution.org/> to build one online and the resulting image did not boot. I suspect that when I selected BeagleBoard as the computer for which I needed the image, it may have been wrong. Or maybe I needed to select some platform specific pkg or maybe

the 2011.03 release as a start was wrong.

If desperate, I will follow your tutorial and build my own distribution this way but it seems that it would be easier to use the online builder if I knew the right answers. Can you help?

Yves



Tom March 19, 2014 at 6:16 pm - Reply

Hi Matt!

The newer versions of Angstrom generate zImages instead of ulimages. I suggest editing the uEnv.txt in your boot partition on the SD card to remedy the problem. A quick Google Search should find one quickly (I don't want to paste mine here, since I got it from another website).



Víctor Mayoral June 19, 2013 at 6:00 pm - Reply

Hi Derek,

Thanks for this great content. Quick question, how would you change the kernel from the 3.8.11 to the 3.2.x?
Regards,
Víctor.



vinay June 29, 2013 at 11:07 am - Reply

Hi Derek,

thanks for this blog was helpful n i have the same question as Victor i want to get back to kernel 3.2.X? and have u tried BBB kernel 3.8 16x2 character lcd? please help me..
Regards,
vinay

Nicolas July 6, 2013 at 5:17 pm - Reply



Hi Derek

I have ubuntu 12.10 , is this the process I would have to do ti enable the SPI bus on the beagleBone.

Im trying to read the output of an adc IC which connects to the SPI.



bwchoi August 3, 2013 at 3:39 am - Reply

Hi Derek.

Thanks for your beaglebone black blog & youtube.

I am developing a program with qt for bbb.

As you mentioned about qt4-embedded installation youtube, i wish to re-create kernel 3.8 for bbb.

But I can't generate it.

First of all I wish to create cloud9-image.

But it can't with

ERROR: /sources/meta-openembedded/meta-oe/recipes-devtools/nodejs/nodejs4_0.4.12.bb,) do_install)
fail with exit code '1'

But I can generate qt4-demo-imgae.

And some image ok , some image not.

Please help me and advice about minimum kernel environment for qt program in bbb.

Best Regards.

byung-wook choi in korea.



Don Miller September 3, 2013 at 12:28 am - Reply

Hello Derek,

In the backup kernel config step, I am finding defconfig in a slightly different path from your case.

You:

```
cd ~/beaglebone/setup-scripts/sources/meta-ti/recipes-kernel/linux/linux-ti33x-psp-3.2/
```

Me:

```
cd ~/beaglebone/setup-scripts/sources/meta-ti/recipes-kernel/linux/linux-ti33x-psp-3.2/beaglebone
```

I was not sure if this was a mistake, or if the environment has changed a bit since you did your build.

Thanks for all your good work!

Don



Matt October 21, 2013 at 4:52 pm - Reply

Again, this was a great help and following this tutorial I was able to build the systemd-image which is better suited to my needs. However, I want to enable hard floating point support. Could you give any details how to do this?
Many thanks.

Leave A Comment

Comment...

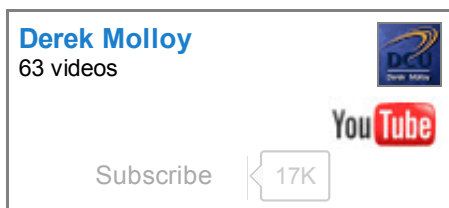
POST COMMENT

Spam Prevention Captcha (apologies, this site is receiving 300+ spam messages per day)

6 + eight =

ABOUT:

This site brings together all of the video content on the Derek Molloy YouTube channel and structures it so that you can follow the videos as lessons. It also integrates associated documentation, datasheets and tools to allow you to get the best from the video series. It also has a blog to allow me to post new videos, articles and useful information that may not be in video form



RECENT COMMENTS

Jason on Qt with Embedded Linux on the Beaglebone

andres on Git and Curl SSL Certificates Configuration on Beaglebone Black

Jose Cabra on The Beaglebone

Feng on GPIOs on the Beaglebone Black using the Device Tree Overlays

Derek on GPIOs on the Beaglebone Black using the Device Tree Overlays

Like Share 61

TAG CLOUD:

Analog Discovery angstrom avconv **beaglebone** **beaglebone black** bitbake build building c++ C920 connman cpu curl disk distribution easydriver ffmpeg flash Flip-Flop git gparted https iso **Java** kernel **LED** LEDs linux live-cd Logic Analyzer nmap opencv portscan re-size resize **RTP source** stepper motor **UDP** vdi Video virtualbox **VLC** Wordpress x264

RECENT TWEETS

I just posted a new Introduction Video on the Analog Discovery USB scope/waveform generator/logic analyzer... <http://t.co/66aG96yzIU> 2 months ago

Just Supported 'NavSpark: GPS GNSS receiver' \$17 for a small powerful crowd-funded GPS rec. <http://t.co/OTMiJkbTzj> #indiegogo via @indiegogo 3 months ago

Follow @DerekMolloyIE 173 followers