

The Red-black tree

Blog of a computer enthusiast and newbie jogger, mad about physics, space science and photography

POSTED BY

PURINDA GUNASEKARA

POSTED ON

OCTOBER 1, 2012

POSTED UNDER

BEAGLEBONE, LINUX, RASPBERRY PI

COMMENTS

3 COMMENTS

Multiple/Second SPI Buses on BeagleBone

How to: Get SPI to work on Beaglebone

If you want just one SPI port running, then most up to date images for beaglebone has enabled them by default.

Location to get pre-compiled images with SPI1 enabled, <http://rcn-ee.net/deb/precise-armhf/v3.2.25-psp20/> (<http://rcn-ee.net/deb/precise-armhf/v3.2.25-psp20/>)

Enabling SPI0

Step 1

- In order to enable userland SPI we need to be able to rebuild the kernel, so grab the Linux source:

```
1 | git clone git://git.kernel.org/pub/scm/linux/kernel/git/stable/linux-stable.git
```

- Install the required tools and libs by running:

```
1 | sudo apt-get install gcc-4.4-arm-linux-gnueabi git ccache libncurses5-dev u-boot-tools lzma
```

I tend to compile the kernel using hardfloat, so arm-linux-gnueabi-hf compiler would be the way to go.

- As per Robert C. Nelson's repo, get the patches required to enable spidev

```
1 | git clone git://github.com/RobertCNelson/linux-dev.git
2 | cd linux-dev
```

- switch to the am33x-v3.2 branch with:

```
1 | git checkout origin/am33x-v3.2 -b am33x-v3.2
```

- Now copy system.sh.sample to system.sh and update system.sh with the settings for your setup:

```
1 | cp system.sh.sample system.sh
```

- For my setup I had to make the following changes to system.sh:

```
1 | uncomment line 14, change from:
2 | #CC=arm-linux-gnueabi-
3 | to:
4 | CC=arm-linux-gnueabi-
5 |
6 | at line 60 update the path to your linux source, change from:
7 | #LINUX_GIT=~/linux-stable/
8 | to:
9 | LINUX_GIT=~/linux-stable/
10 |
11 | uncomment line 70 to set the kernel entry point, change from:
12 | #ZRELADDR=0x80008000
13 | to:
14 | ZRELADDR=0x80008000
15 |
16 | uncomment line 80 to set the BUILD_UIMAGE flag, change from:
17 | #BUILD_UIMAGE=1
18 | to:
19 | BUILD_UIMAGE=1
20 |
21 | and finally at line 89 uncomment and set the path to the SD card, change from:
22 | #MMC=/dev/sde
23 | to:
24 | MMC=/dev/sdd
```

- Build kernel command

```
1 | ./build_kernel.sh
```

Step 2

Note: This step will discuss how to enable SPI0 (SPI1 is enabled by default with latest kernels, or the one you download from above URL which contains precompiled images).

- Goto “/KERNEL/arch/arm/mach-omap2” directory in kernel source which is in the clone you got from “[git://github.com/RobertCNelson/linux-dev.git](https://github.com/RobertCNelson/linux-dev.git) (/ /github.com/RobertCNelson/linux-dev.git)” in above steps.
- Make the following modification after you get to Kernel configuration menu.
 - Note: if you make the change before you get to config menu, the git update process will overwrite it again with the files from repository.

```

1  *** /home/purinda/Software/Linux/linux-dev/KERNEL/arch/arm/mach-omap2/board-am335xevm.c 20:
2  --- board-am335xevm.c 2012-08-09 23:36:24.774040301 +1000
3  ****
4  *** 2275,2280 ****
5  --- 2275,2289 ----
6  };
7
8  static struct spi_board_info bone_spidev2_info[] = {
9  + /* [PG] I added the following struct call in order to get SPI0 working
10 + */
11 + {
12 + .modalias = "spidev",
13 + .irq = -1,
14 + .max_speed_hz = 12000000,
15 + .bus_num = 1,
16 + .chip_select = 0,
17 + },
18 {
19 .modalias = "spidev",
20 .irq = -1,
21 ****
22 *** 3189,3195 ****
23 if(beaglebone_spi1_free == 1) {
24 beaglebone_spi1_free = 0;
25 pr_info("BeagleBone cape: exporting SPI pins as spidev\n");
26 ! setup_pin_mux(spi1_pin_mux);
27 spi_register_board_info(bone_spidev2_info, ARRAY_SIZE(bone_spidev2_info));
28 }
29 if(beaglebone_wlgpio_free == 1) {
30 --- 3198,3207 ----
31 if(beaglebone_spi1_free == 1) {
32 beaglebone_spi1_free = 0;
33 pr_info("BeagleBone cape: exporting SPI pins as spidev\n");
34 ! /* [PG] I made the following pin_mux call in order to get SPI0 working

```

```
35 | ! */
36 | ! setup_pin_mux(spi0_pin_mux);
37 | ! setup_pin_mux(spi1_pin_mux);
38 | spi_register_board_info(bone_spidev2_info, ARRAY_SIZE(bone_spidev2_info));
39 | }
40 | if(beaglebone_wlgpio_free == 1) {
```

Step 3

- Call `./build_kernel.sh` (inside `linux-dev` directory).

Step 4

- Call `./tools/install_image.sh`
 - you may need to modify `system.sh` file according to the changes displayed on above.
 - if you have set the correct memory card device in `system.sh` it will load the compiled driver correctly into the boot partition. In my case the mmc dev was `/dev/sdb`

How to test

- Once you boot the new Kernel you should see two devices in `/dev/` directory as follows when you run the following command

```
1 | ls /dev/spidev* -lah
2 |
3 | Output should be
```

```

4 |
5 | crw----- 1 root root 153, 0 Aug 9 23:58 /dev/spidev1.0
6 | crw----- 1 root root 153, 1 Aug 9 23:58 /dev/spidev2.0

```

- spidev1.0 is spi0 in beaglebone board and spidev2.0 is the spi1(the default spi master active on precompiled kernel images) in beaglebone.
- once you can see these two devices you are ready to test.
- connect pin 18 and 21 in header P9 with a cable, compile and run spidev_test.c in documentation/spi folder in Kernel source (refer to 1st site in references section) to test “spidev1.0”.
- connect pin 28 and 30 in header P9 with a cable, compile and run spidev_test.c in documentation/spi folder in Kernel source (refer to 1st site in references section) to test “spidev2.0”.
- When you do so make sure you change make sure you change the following line (in spidev_test.c) to spidev1.0 or spidev2.0, depending on what you need to test.

```

1 | static const char *device = "/dev/spidev1.1";

```

Example SPI driver with modifications

Modified am335_omap_spi.c file from kernel source (<https://docs.google.com/open?id=0BwVDj5gU8OnTM24wVk9pNmI3d28>)

References:

- <http://www.yellowfeather.co.uk/2012/03/userland-spi-on-the-beaglebone-with-ubuntu/>
(<http://www.yellowfeather.co.uk/2012/03/userland-spi-on-the-beaglebone-with-ubuntu/>)
- <http://dp.nonoo.hu/beaglebone-spi-driver/> (<http://dp.nonoo.hu/beaglebone-spi-driver/>)
- <http://communistcode.co.uk/blog/blogPost.php?blogPostID=1> (<http://communistcode.co.uk/blog/blogPost.php?blogPostID=1>)
- http://e2e.ti.com/support/dsp/sitara_arm174_microprocessors/f/791/p/194808/696323.aspx
(http://e2e.ti.com/support/dsp/sitara_arm174_microprocessors/f/791/p/194808/696323.aspx)

About these ads (<http://en.wordpress.com/about-these-ads/>)

- Families That RV Together Find Joy <http://bit.ly/Nf8Yar>

-
-
-
- <http://bit.ly/Nf8Yar>

- [Beaglebone \(<http://theredblacktree.wordpress.com/tag/beaglebone/>\)](http://theredblacktree.wordpress.com/tag/beaglebone/)
- [Linux \(<http://theredblacktree.wordpress.com/tag/linux/>\)](http://theredblacktree.wordpress.com/tag/linux/)
- [Raspberry Pi \(<http://theredblacktree.wordpress.com/tag/raspberry-pi/>\)](http://theredblacktree.wordpress.com/tag/raspberry-pi/)

3 thoughts on “Multiple/Second SPI Buses on BeagleBone”

1.

Q says:

February 10, 2013 at 8:37 AM

Curious, should:

connect pin 28 and 30 in header P9 with a cable and run spidev_test.c in documentation/spi folder in Kernel source (refer to 1st site in references section) to test spidev1.0.

say...

Connect pins 29, 30 in header P9 to test spidev2.0 ?

If not, it would be useful to test both SPI buses.

Love the article, very helpful man.

Reply

◦

Purinda Gunasekara says:

February 10, 2013 at 8:51 AM

I was too lazy to make that little change :), now its updated.

Reply

2.

Eric Stark says:

May 3, 2013 at 10:29 AM

Thank you for the well-written reference on SPI for the BeagleBone. Do these same instructions work for the BeagleBone Black?

Reply

[Blog at WordPress.com.](#) | [The Zoren Theme.](#)

Follow

Follow “The Red-black tree”

Powered by WordPress.com