**Installing libpruio for use on Beaglebone Black**

**Debian Linux**

**20150130**

1. **Install FreeBasic Compiler**
   1. **Download and uncompress** the BeagleBone Black FreeBasic Compiler (BBB-FBC) package

>wget [http://www.freebasic-portal.de/dlfiles/589/BBB\_fbc-1.00.tar.bz2 tar xjf BBB\_fbc-1.00.tar.bz2](http://www.freebasic-portal.de/dlfiles/589/BBB_fbc-1.00.tar.bz2%20tar%20xjf%20BBB_fbc-1.00.tar.bz2)

>tar xjf BBB\_fbc-1.00.tar.bz2

* 1. **Copy the files and folders** to the appropriate folders so that they are in the command path

>cd BBB\_fbc-1.00

>cp usr/local/bin/fbc /usr/local/bin/

>cp -R usr/local/lib/freebasic /usr/local/lib/

* 1. **Test the BBB-FBC compiler**

>fbc –version

FreeBASIC Compiler - Version 1.01.0 (10-14-2014), built for linux-arm (32bit)

Copyright (C) 2004-2014 The FreeBASIC development team.

1. **Install pruss driver kit**
   1. **Install original am335x-pru-package**

>apt-get install am335x-pru-package

* 1. **Download and uncompress** FB package from FB prussdrv Kit (BBB)

>wget http://www.freebasic-portal.de/dlfiles/539/FB\_prussdrv-0.0.tar.bz2

>tar xjf FB\_prussdrv-0.0.tar.bz2

* 1. **Make a new folder** at the FBC location and copy files to that location

>cd FB\_prussdrv-0.0

>mkdir –p /usr/local/include/freebasic/BBB *#”-p” added here to avoid error msg.*

>cp include/\* /usr/local/include/freebasic/BBB

>cp bin/pasm /usr/local/bin

1. **Install libpruio**
   1. **Download and uncompress** package from libpruio (D/A – I/O)

> wget http://www.freebasic-portal.de/dlfiles/592/libpruio-0.2.tar.bz2

> tar xjf libpruio-0.2.tar.bz2

* 1. **Copy Files to appropriate locations**

>cd libpruio-0.2

>cp src/c\_wrapper/libpruio.so /usr/local/lib

>ldconfig #*configure dynamic linker run-time bindings*

>cp src/c\_wrapper/pruio\*.h\* /usr/local/include

>cp src/config/libpruio-0A00.dtbo /lib/firmware

>cp src/pruio/pruio\*.bi /usr/local/include/freebasic/BBB

>cp src/pruio/pruio.hp /usr/local/include/freebasic/BBB

*Extra Step for Preconditions necessary to run (c) below*:

>cp src/config/libpruio-0A00.dtbo /lib/firmware

* 1. **Activate the PRUSS** by enabling the tree overlay. This must be done every time, after each boot or before running your programs.

~~> echo libpruio > /sys/devices/bone\_capemgr.\*/slots~~

This didn’t work and caused a “file not found” error, so tried:

> echo **libpruio**:**0A00** > /sys/devices/bone\_capemgr.\*/slots *(added version number)*

which results in:

0: 54:PF---

1: 55:PF---

2: 56:PF---

3: 57:PF---

4: ff:P-O-L Bone-LT-eMMC-2G,00A0,Texas Instrument,BB-BONE-EMMC-2G

5: ff:P-O-L Bone-Black-HDMI,00A0,Texas Instrument,BB-BONELT-HDMI

9: ff:P-O-L Override Board Name,0A00,Override Manuf,**libpruio**

or

>echo **BB-BONE-PRU-01** > /sys/devices/bone\_capemgr.\*/slots

which results in:

root@beaglebone:~# cat /sys/devices/bone\_capemgr.9/slots

0: 54:PF---

1: 55:PF---

2: 56:PF---

3: 57:PF---

4: ff:P-O-L Bone-LT-eMMC-2G,00A0,Texas Instrument,BB-BONE-EMMC-2G

5: ff:P-O-L Bone-Black-HDMI,00A0,Texas Instrument,BB-BONELT-HDMI

8: ff:P-O-L Override Board Name,00A0,Override Manuf,**BB-BONE-PRU-01**

* 1. **Test example**

Go to where you uncompressed libpruio-0.2 and run

> sudo src/examples/1 or >./1

Should see a list of 13 samples from 8 Analog Inputs AIN-0 to AIN-7, (in HEX).

These are 12-bit values, (4095, 0x0FFF), but they are left shifted by 4-bits to yield 16-bit values, with a maximum of 65520 (0xFFF0) hex.

0 1 2 3 4 5 6 7 (AIN-7 is tied to a 3.3V voltage divider).

A160 A0C0 A9C0 A170 9860 9860 98B0 F000

A170 A0D0 AA20 A190 9970 9780 9790 EFF0

A110 A0D0 AA00 A110 9990 97E0 97E0 F010

A190 A0D0 AAD0 A160 9AC0 9880 9940 F030

1.136 Volts

1. Prepare system (optional) as described in “Precondtions”.

**Abbreviations**

**PRU** Programmable Real-time

**PRUSS** Programmable Real-time Unit Sub System

**Links/URLs**

* **[http://users.freebasic-portal.de/tjf/Projekte/libpruio/doc/html/\_cha\_preparation.html#SecInstallation](http://users.freebasic-portal.de/tjf/Projekte/libpruio/doc/html/_cha_preparation.html" \l "SecInstallation)**
* **<http://www.freebasic-portal.de/downloads/fb-on-arm/bbb-fbc-fbc-fuer-beaglebone-black-283.html>**
* **<https://github.com/beagleboard/am335x_pru_package>**
* **<http://beagleboard.org/Community/Forums?place=msg%2Fbeagleboard%2F9NYdGWOT_Mg%2F6X0v2XVEeUAJ>**

I've been following the online documentation for libpruio to build and run the libpruio "c\_examples" using the linaro to cross-compile in Eclipse Luna CDT from Windows7-64bit. I've finally gotten it to work for "root", but I've still do not have it working for a non-admin after rebooting, but I know why, (see below). I'll document all this in detail in a few days, but here is a quick summary:

a) libpruio version number is 0A00, not 00A0, (maybe this was a typo??), which required that the version number be specified in /etc/default/capemgr (CAPE=libpruio:0A00), in order to get the overlay installed at boot time. Even running "echo libpruio > /sys/devices/bone\_capemgr.slots" results in "file not found", without changing the first part to "echo libpruio:0A00" because of the version number of the dtbo file.

b) /etc/default/capemgr has to be executable in order to be run by /etc/int.d/capemgr.sh. In my original Beaglebone Black installation this file was not configured for execution. So root changed it "chmod +x /etc/default/capemgr".

c) I had to provide the library paths to three libraries to compile the "1.c" example; libpruio.so; libprussdrv.so; and libtinfo.so. The last two shared object, I copied from the Beaglebone Black to my Windows PC in order to compile within Eclipse using linaro. libprussdrv.so is from FreeBasic, but libtinfo.so is from the am335x\_pru\_package.

d) as discussed in the online libpruio documentation, I created a "pruio" group and changed the group of /dev/uio5 and added myself (non-admin) as a member of this group, which then allows me as non-admin to run the libpruio example. However, /dev/uio5 loses this group membership upon reboot. So I'm currently looking for a way to re-establish this group assignment to /dev/uio5 upon boot so that non-admins can run the code directly, (i.e. "chgrp pruio /dev/uio5" and "chmod g+rw /dev/uio5" as in the libpruio online documentation.

FYI

Status as of: 2015-02-02