Beaglebone-Debian.

by **Computothought** on June 24, 2013

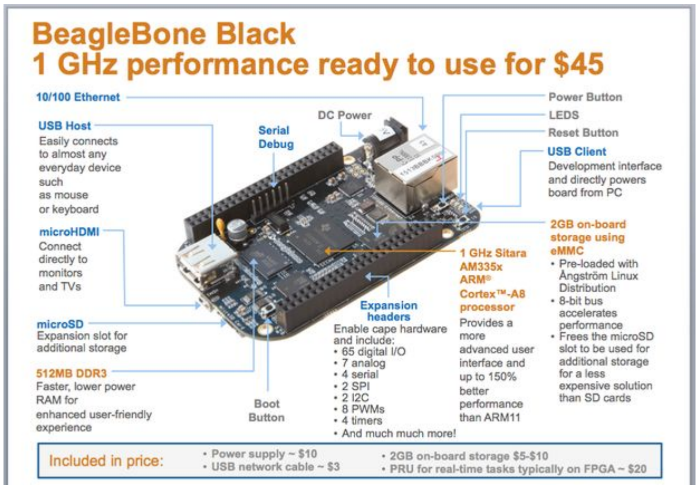
Table of Contents

Author:Computothought Computothought

Educator, technician, unchef, and chief bottle washer. if you can dream it, you can draw it. If you can draw it, you can build it.

Intro: Beaglebone-Debian.

Testing the Beaglebone with Debian.



BeagleBone Black
1 GHz performance ready to use for $45

10/100 Ethernet

USB Host
Easily connects to almost any everyday device such as mouse or keyboard

microHDMI
Connect directly to monitors and TVs

microSD
Expansion slot for additional storage

512MB DDR3
Faster, lower power RAM for enhanced user-friendly experience

Serial Debug

DC Power

Boot Button

Expansion headers
Enable cape hardware and include:
- 65 digital I/O
- 7 analog
- 4 serial
- 2 SPI
- 2 I2C
- 8 PWMs
- 4 timers
- And much much more!

1 GHz Sitara AM335x ARM® Cortex™-A8 processor
Provides a more advanced user interface and up to 150% better performance than ARM11

Power Button

LEDS

Reset Button

USB Client
Development interface and directly powers board from PC

2GB on-board storage using eMMC
- Pre-loaded with Ångström Linux Distribution
- 8-bit bus accelerates performance
- Frees the microSD slot to be used for additional storage for a less expensive solution than SD cards

Included in price:
- Power supply ~ $10
- USB network cable ~ $3
- 2GB on-board storage $5-$10
- PRU for real-time tasks typically on FPGA ~ $20

Step 1: What came in the box.

At least what I was give was just the board and a microusb cable. in the second picture you can see the Beaglebone comes in a larger box than the Raspberry Pi.

Step 2: Mouser video on basic standard install.

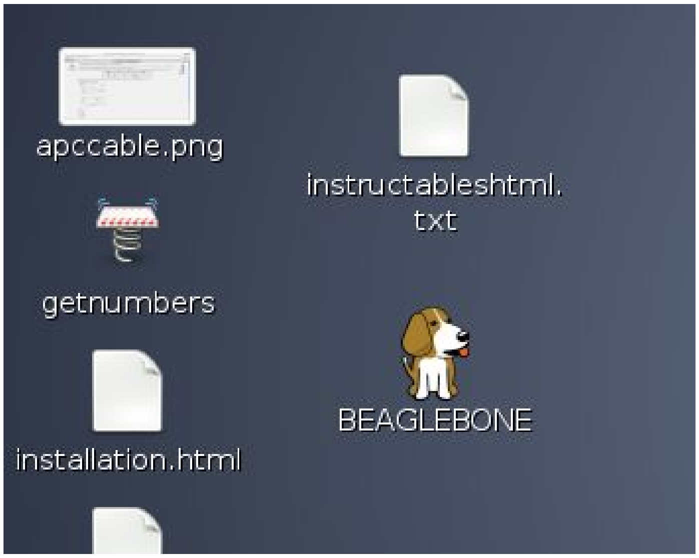
This is ib case you just want to use it as is until you become familliar with the unit.

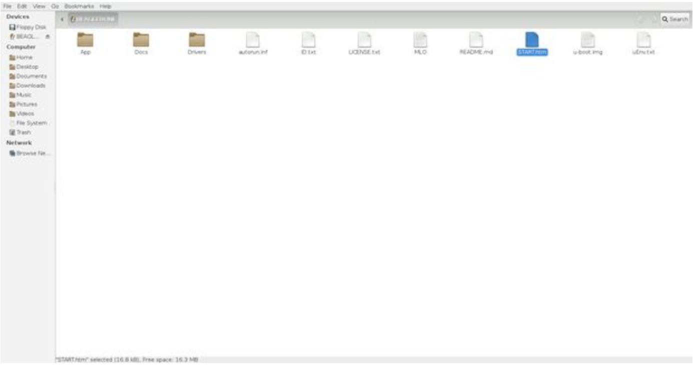
Change the root password on the unit. The web page gave a semifialse ipaddress. So I used the good old pingall.sh script to see what was connect to the network. I knew all ut one of the devices listed. 192.168.1.103 must be the Beaglebon and it was.

```
$ pingall.sh
You are using network 1.
64 bytes from 192.168.1.1: icmp_seq=1 ttl=64 time=2.40 ms
64 bytes from 192.168.1.99: icmp_seq=1 ttl=255 time=4.79 ms
64 bytes from 192.168.1.103: icmp_seq=1 ttl=64 time=0.592 ms
64 bytes from 192.168.1.115: icmp_seq=1 ttl=64 time=0.062 ms
```

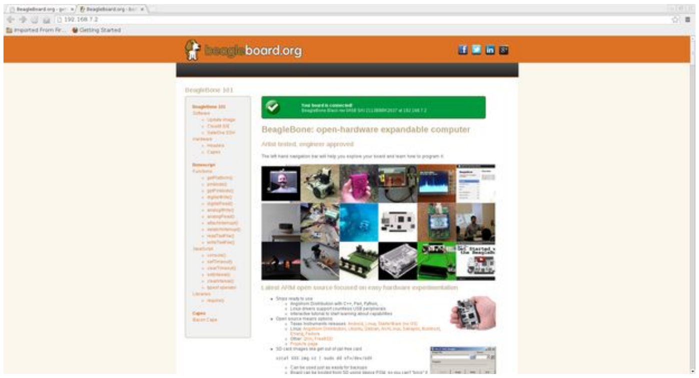
```
$ ssh root@192.168.1.103
The authenticity of host '192.168.1.103 (192.168.1.103)' can't be established.
RSA key fingerprint is 3f:10:b4:e2:be:b4:53:94:f2:8c:48:3d:6f:d1:37:a5.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.1.103' (RSA) to the list of known hosts.
root@192.168.1.103's password:
root@beaglebone:~# passwd
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
# poweroff
```

Shut the system down for now.





"START.htm" selected (16.8 kB), Free space: 16.3 MB


beagleboard.org
BeagleBone 101
BeagleBone: open-hardware expandable computer
Artist tested, engineer approved

Step 3: Basic install.

eMMC: BeagleBone Black

This image can be written to a 1Gb (or greater) microSD card, via 'dd' in linux or the win32 image program linked to on CircuitCo's wiki page. First hold down on the boot select button (next to microSD card) and apply power (same procedure as the official CircuitCo images), it should boot into debian and begin flashing the eMMC, once completed all 4 LED's should be full ON... Simply remove power, remove microSDcard and Debian will now boot from eMMC.

Reference: (this is the script that writes to the eMMC)

```
https://github.com/RobertCNelson/tools/blob/master/scripts/beaglebone-black-copy-microSD-to-eMMC.sh
```

BTW: we are only writing about 500Mb's to the eMMC so the script will only take about 5-6 Minutes after power on.

Image Updated:

- 2013-06-14
  - shellinabox disabled...
  - BeagleBone Black: v3.8.13-bone21 kernel (--uboot bone_dtb)
    - BBB: Now built with the patched 'dtc'...
- 2013-05-29
  - BeagleBone Black: v3.8.13-bone20 kernel
    - BBB: USB hotplug now works...

Get prebuilt image:

```
wget http://rcn-ee.net/deb/flasher/wheezy/BBB-eMMC-flasher-debian-7.0.0-2013-06-14.img.xz
```

Verify Image with:

```
md5sum BBB-eMMC-flasher-debian-7.0.0-2013-06-14.img.xz
74dc3f8949c71621509545bfa06399d9  BBB-eMMC-flasher-debian-7.0.0-2013-06-14.img.xz
```

Follow the "standard" update procedure.

```
http://circuitco.com/support/index.php?title=Updating_The_Software
```

```
Note: It is usally a good ide to use and update the stock firmware before doing any wholsale changes. That way you know the wquipment is ok in case th
```

Step 4: Net installof Debian.

Debian net install might longer but you will have the latest version of the software.
Scripts:

```
git clone git://github.com/RobertCNelson/netinstall.git
cd netinstall
```

Device Options:

```
BeagleBoard --uboot beagle_cx
BeagleBoard xM --uboot beagle_xm
BeagleBone (serial) --uboot bone-serial
BeagleBone (video via cape) --uboot bone-video
BeagleBone Black --dtb am335x-boneblack
PandaBoard --uboot panda
PandaBoard ES  --uboot panda_es
```

You will need a 1GB SD card or greater.

```
Standard System : ~455MB
+ Desktop environment (GNOME) : ~2.9GB
```

Step 5: Debian Wheezy.

Quick Install script for "board"

```
sudo ./mk_mmc.sh --mmc /dev/sdX --uboot "board" --distro wheezy-armhf
```

So For the BeagleBoard xM:

```
sudo ./mk_mmc.sh --mmc /dev/sdX --uboot beagle_xm --distro wheezy-armhf
```

- Options:
  - --firmware : installs firmware
  - --serial-mode : debian-installer uses Serial Port

Demo Image

Debian 7.0.0 (wheezy)

Default username/password:

- username: debian
- password: temppwd

Default root user/password

- user: root
- password: root

Image Updated:

- 2013-07-22
  - Beagle/Panda/Panda ES: v3.7.10-x13 kernel
  - BeagleBone/BeagleBone Black: v3.8.13-bone24 kernel
- 2013-06-14
  - shellinabox disabled...
  - Beagle/Panda/Panda ES: v3.7.10-x12 kernel
    - Panda: WiFi works again...
  - BeagleBone: v3.2.42-psp27 kernel
  - BeagleBone/BeagleBone Black: v3.8.13-bone21 kernel (--uboot bone_dtb)
    - BBB: Now built with the patched 'dtc'...
- 2013-05-29
  - Beagle/Panda/Panda ES: v3.7.10-x10 kernel
  - BeagleBone: v3.2.42-psp27 kernel
  - BeagleBone/BeagleBone Black: v3.8.13-bone20 kernel (--uboot bone_dtb)
    - BBB: USB hotplug now works...

Services Active:

```
Note: Depending on your internal network these may work out the box
Apache, Port 80: http://arm/ (Bone: via usb) http://192.168.7.2
SSH, Port 22: ssh ubuntu@arm (Bone: via usb) ubuntu@192.168.7.2
Getty, Serial Port
```

Active on Boot Script:

- Disable by removing "run_boot-scripts" from small boot partition...*

```
/opt/boot-scripts/<board>.sh
```

Get prebuilt image:

```
wget http://rcn-ee.net/deb/rootfs/wheezy/debian-7.1-console-armhf-2013-07-22.tar.xz
```

Verify Image with:

```
md5sum debian-7.1-console-armhf-2013-07-22.tar.xz
52d53e4cbfd7c227e8f1d952409d8ae3  debian-7.1-console-armhf-2013-07-22.tar.xz
```

Unpack Image:

```
tar xJf debian-7.1-console-armhf-2013-07-22.tar.xz
cd debian-7.1-console-armhf-2013-07-22
```

Install Image:

Quick Install script for "board"

```
sudo ./setup_sdcard.sh --mmc /dev/sdX --uboot "board"
```

"board" options:

- BeagleBoard Ax/Bx - beagle_bx
- BeagleBoard Cx/Dx - beagle_cx
- BeagleBoard xM - beagle_xm
- BeagleBone/Black - bone/bone_dtb
- PandaBoard Ax - panda
- PandaBoard ES - panda_es

So For the BeagleBoard xM:

```
sudo ./setup_sdcard.sh --mmc /dev/sdX --uboot beagle_xm
```

- Additional Options
  - --rootfs <ext4 default>
  - --swap_file <swap file size in MB's>
  - --addon pico <ti pico projector>
  - --svideo-ntsc <use ntsc over dvi for video)
  - --svideo-pal <use pal over dvi for video)

Step 6: Debian Jessie.

Debian Jessie (development snapshot)

Image Updated:

- 2013-07-22
    - Beagle/Panda/Panda ES: v3.7.10-x13 kernel
    - BeagleBone/BeagleBone Black: v3.8.13-bone24 kernel
- 2013-06-14
    - shellinabox disabled...
    - Beagle/Panda/Panda ES: v3.7.10-x12 kernel
        - Panda: WiFi works again...
    - BeagleBone: v3.2.42-psp27 kernel
    - BeagleBone/BeagleBone Black: v3.8.13-bone21 kernel (--uboot bone_dtb)
        - BBB: Now built with the patched 'dtc'...
    - 2013-05-29
    - Beagle/Panda/Panda ES: v3.7.10-x10 kernel
    - BeagleBone: v3.2.42-psp27 kernel
    - BeagleBone/BeagleBone Black: v3.8.13-bone20 kernel (--uboot bone_dtb)
        - BBB: USB hotplug now works...

Get prebuilt image:

```
wget http://rcn-ee.net/deb/rootfs/jessie/debian-jessie-console-armhf-2013-07-22.tar.xz
```

Verify image with:

```
md5sum debian-jessie-console-armhf-2013-07-22.tar.xz
28bf234d63eba8fa94e2c6a1fd28128a  debian-jessie-console-armhf-2013-07-22.tar.xz
```

Unpack image:

```
tar xJf debian-jessie-console-armhf-2013-07-22.tar.xz
cd debian-jessie-console-armhf-2013-07-22
```

Then follow directions show above with the other images...

Flasher

eMMC: BeagleBone Black

This image can be written to a 1Gb (or greater) microSD card, via 'dd' in linux or the win32 image program linked to on CircuitCo's wiki page. First hold down on the boot select button (next to microSD card) and apply power (same procedure as the official CircuitCo images), it should boot into debian and begin flashing the eMMC, once completed all 4 LED's should be full ON... Simply remove power, remove microSDcard and Debian will now boot from eMMC.

Reference: (this is the script that writes to the eMMC)

```
https://github.com/RobertCNelson/tools/blob/master/scripts/beaglebone-black-copy-microSD-to-eMMC.sh
```

BTW: we are only writing about 500Mb's to the eMMC so the script will only take about 5-6 Minutes after power on.

Image Updated:

- 2013-07-22
    - BeagleBone Black: v3.8.13-bone24 kernel
- 2013-06-14
    - shellinabox disabled...
    - BeagleBone Black: v3.8.13-bone21 kernel (--uboot bone_dtb)
        - BBB: Now built with the patched 'dtc'...
- 2013-05-29
    - BeagleBone Black: v3.8.13-bone20 kernel
        - BBB: USB hotplug now works...

Get prebuilt image:

```
wget http://rcn-ee.net/deb/flasher/wheezy/BBB-eMMC-flasher-debian-7.1-2013-07-22.img.xz
```

Verify Image with:

```
md5sum BBB-eMMC-flasher-debian-7.1-2013-07-22.img.xz
86d771af79131526913e3b98089d96c7  BBB-eMMC-flasher-debian-7.1-2013-07-22.img.xz
```

Follow the "standard" update procedure.

```
http://circuitco.com/support/index.php?title=Updating_The_Software
```

Debian Configuration

Serial Ports

Wheezy

edit /etc/inittab and add:

```
T0:23:respawn:/sbin/getty -L ttyO2 115200 vt102
```

SGX Video Acceleration

SGX armel/armhf v3.4.x+

- Note: This is a still a work in progress, but so far all the basic sgx demos seem to work on my Beagle xM C... Thanks to TI for the special armhf binaries!!! -- RobertCNelson 19:48, 17 July 2012 (UTC)
- Test with: Precise/12.04 armhf demo image with v3.4.5-x1, desktop was lxde via: "sudo apt-get install lxde lxde-core lxde-icon-theme"

Re-Build Kernel and SGX Kernel Modules

```
git clone git://github.com/RobertCNelson/stable-kernel.git
cd stable-kernel
git checkout origin/v3.7.x -b tmp
./build_kernel.sh (and then follow the directions as the script runs...)
```

Build kernel

```
./build_kernel.sh
```

Build SGX modules

```
./sgx_build_modules.sh
```

Insert SD card, make sure to modify MMC in system.sh

```
./tools/install_image.sh
```

Place SD card into the device and boot...

```
cd /opt/sgx
sudo tar xf GFX_4.0*_libs.tar.gz
sudo ./install-sgx.sh
```

Reboot, check modules (lsmod):

```
Module                  Size  Used by
bufferclass_ti          5727  0
omaplfb                11512  0
pvrsrvkm              165208  2 bufferclass_ti,omaplfb
```

Blit Test:

```
ubuntu@omap:/usr/bin/armhf/es5.0$ ./sgx_blit_test
------------------ SGX 3D Blit test -----------------
----------------------- Start -----------------------
Call PVRSRVConnect with a valid argument:
OK
Get number of devices from PVRSRVEnumerateDevices:
OK
.... Reported 1 devices
.... Device Number  | Device Type
           0000    | PVRSRV_DEVICE_ID_SGX
Attempt to acquire device 0:
OK
Getting SGX Client info
OK
.... ui32ProcessID:1133
Display Class API: enumerate devices
OK
PVRSRVEnumerateDeviceClass() returns 1 display device(s)
OK
Display Class API: open device
OK
Display Class API: Get display info
OK
.... Name:PowerVR OMAP Linux Display Driver
.... MaxSwapChains:1
.... MaxSwapChainBuffers:1
.... MinSwapInterval:0
.... MaxSwapInterval:1
Display Class API: enumerate display formats
OK
OK
.... Display format 0 - Pixelformat:1
Display Class API: enumerate display dimensions
OK
OK
.... Display dimensions 0 - ByteStride:2560 Width:1280 Height:720
Attempt to create memory context for SGX:
OK
.... Shared heap 0 - HeapID:0x7000000 DevVAddr:0x1000 Size:0x87fe000 Attr:0x2014200
.... Shared heap 1 - HeapID:0x7000001 DevVAddr:0xc800000 Size:0xfff000 Attr:0x2024200
.... Shared heap 2 - HeapID:0x7000002 DevVAddr:0xe400000 Size:0x7f000 Attr:0x2024200
.... Shared heap 3 - HeapID:0x7000003 DevVAddr:0xf000000 Size:0x3ff000 Attr:0x2024200
.... Shared heap 4 - HeapID:0x7000004 DevVAddr:0xf400000 Size:0x4ff000 Attr:0x2014200
.... Shared heap 5 - HeapID:0x7000005 DevVAddr:0xfc00000 Size:0x1ff000 Attr:0x2014200
```

```
.... Shared heap 6 - HeapID:0x7000006 DevVAddr:0xdc00000 Size:0x7ff000 Attr:0x2014200
.... Shared heap 7 - HeapID:0x7000007 DevVAddr:0xe800000 Size:0x7ff000 Attr:0x2014200
.... Shared heap 8 - HeapID:0x7000008 DevVAddr:0xd800000 Size:0x3ff000 Attr:0x2024200
.... Shared heap 9 - HeapID:0x7000009 DevVAddr:0x8800000 Size:0x0 Attr:0x2024200
.... Shared heap 10 - HeapID:0x700000a DevVAddr:0x8800000 Size:0x3fff000 Attr:0x2014200
Display Class API: get the system (primary) buffer
OK
Display Class API: map display surface to SGX
OK
Attempt to create transfer context for SGX:
OK
Do a SRCCOPY blit to the bottom right quadrant of the display:
(bottom right quadrant should be red on blue background):
OK
OK
Do a SRCCOPY blit to the top left quadrant of the display:
(top left quadrant should be striped (r/g/b/w) on blue background):
OK
OK
Do a CUSTOMSHADER blit to the top right quadrant of the display:
(top right quadrant should be yellow):
0xb6acd000 (host) 0xf407000 (device): Device mem for custom shader program
0xb6acb000 (host) 0xf408000 (device): Device mem for texture
USE custom shader program: 0x28841001.c0000000 mov.end o0, sa0
OK
Do a SRCCOPY blit with COLOUR DOWNSAMPLING from ARGB8888 to RGB565
and then present the RGB565 to the bottom right quadrant of the screen
(bottom right quadrant should be a red gradient):
OK
OK
OK
OK
Free the off screen surfaces:
OK
OK
OK
OK
Destroy the transfer context:
OK
Display Class API: unmap display surface from SGX
OK
Destroy Device Memory Context
Display Class API: close the device
OK
Release SGX Client Info:
OK
Disconnect from services:
OK
------------------ SGX 3D Blit test -----------------
------------------------ End ------------------------
```

SGX Legacy armel only upto v3.2.x

NOTE: this only works on BeagleBoard hardware, BeagleBone stuff is in development..

Requirements: stable-kernel (the Demo Images hosted on rcn-ee.net meet this requirement)

```
 https://github.com/RobertCNelson/stable-kernel
```

Note: Due to a bug (seems to only effect older Beagle Bx/Cx boards, use v3.0.8-x3 based kernels)

```
https://github.com/RobertCNelson/stable-kernel/issues/8
oneiric:
wget http://rcn-ee.net/deb/oneiric/v3.0.8-x3/install-me.sh
/bin/bash install-me.sh
```

SDK unPackage Script

Download the latest version of the "create_sgx_package.sh" script

```
2.6.37
wget https://raw.github.com/RobertCNelson/tools/master/x86/ti_omap/create_sgx_package_2.6.37.sh
2.6.38->3.2.x
wget https://raw.github.com/RobertCNelson/tools/master/x86/ti_omap/create_sgx_package_3.2.x.sh
```

Make script executable

```
chmod a+x ./create_sgx_package_*.sh
```

Run script

```
./create_sgx_package_*.sh
```

After Successfully running:

```
:~/temp$ ls
create_sgx_package.sh
GFX_X_XX_XX_XX_libs.tar.gz                      : -> Copy to Beagle (System Libs)
GFX_Linux_SDK.tar.gz                            : -> Copy to Beagle (DEMO's)
Graphics_SDK_setuplinux_X_XX_XX_XX.bin
```

```
SDK
SDK_BIN
```

Beagle: GFX_*_libs.tar.gz

```
tar xf GFX_4_00_00_01_libs.tar.gz  (extracts install-SGX.sh and run-SGX.sh)
./install-SGX.sh (copies necessary SGX libs and startup script)
./run-SGX.sh (force run the new init script, or you can just reboot...)
```

On successful install:

```
Stopping PVR
Starting PVR
Starting SGX fixup for ES2.x (or ES3.x) (or ES5.x xM)
```

Reboot for good measure

```
sudo reboot
```

Beagle: GFX_Linux_SDK.tar.gz

```
tar xf GFX_Linux_SDK.tar.gz
cd GFX_Linux_SDK
tar xf OGLES.tar.gz
```

Test SGX with a DEMO

```
cd OGLES/SDKPackage/Binaries/CommonX11/Demos/EvilSkull
./OGLESEvilSkull
```

Trouble Shooting

```
sudo rm /etc/powervr-esrev
sudo depmod -a omaplfb
sudo /etc/init.d/pvr restart
```

DSP

gst-dsp

- Note: This section is BeagleBoard(-xM) only since the BeagleBone (Black) doesn't have a DSP.

The following Gstreamer elements will be installed:

```
$ gst-inspect | grep dsp
dvdspu:  dvdspu: Sub-picture Overlay
dsp:  dspdummy: DSP dummy element
dsp:  dspvdec: DSP video decoder
dsp:  dspadec: DSP audio decoder
dsp:  dsph263enc: DSP video encoder
dsp:  dspmp4venc: DSP MPEG-4 video encoder
dsp:  dspjpegenc: DSP video encoder
dsp:  dsph264enc: DSP video encoder
dsp:  dspvpp: DSP VPP filter
dsp:  dspipp: DSP IPP
```

Please note that h264 encoder (dsph264enc) will not work because of missing h264venc_sn.dll64P DSP part. According to this message, it is not available due to a licensing restriction.

Requirements: Kernel built with: "CONFIG_TIDSPBRIDGE=m", for reference, here is what rcn-ee.net's image/deb's are configured for:

```
ubuntu@arm:~$ zcat /proc/config.gz | grep TIDSP
CONFIG_TIDSPBRIDGE=m
CONFIG_TIDSPBRIDGE_MEMPOOL_SIZE=0x600000
CONFIG_TIDSPBRIDGE_RECOVERY=y
# CONFIG_TIDSPBRIDGE_CACHE_LINE_CHECK is not set
# CONFIG_TIDSPBRIDGE_NTFY_PWRERR is not set
# CONFIG_TIDSPBRIDGE_BACKTRACE is not set
```

On the xM: if 3.2.x is too jerky, try 3.4.x and use the create_dsp_package.sh script, as the module changed..

Download the latest version of the "create_dsp_package.sh" script

```
wget https://raw.github.com/RobertCNelson/tools/master/x86/ti_omap/create_dsp_package.sh
```

Make script executable

```
chmod a+x ./create_dsp_package.sh
```

Package script:

```
./create_dsp_package.sh
```

Copy DSP_Install_libs.tar.gz to Beagle

Setup network...

Extract:

```
ubuntu@arm:~$ tar xf DSP_Install_libs.tar.gz
```

Install:

```
ubuntu@arm:~$ ./install-DSP.sh
```

What got installed:

```
ubuntu@arm:~$ ls -lh /lib/dsp/
total 7.1M
-rwxr-xr-x 1 root root 1.3M Dec  3 10:56 baseimage.dof
-rwxr-xr-x 1 root root  51K Dec  3 10:56 conversions.dll64P
-rwxr-xr-x 1 root root  13K Dec  3 10:56 dctn_dyn.dll64P
-rwxr-xr-x 1 root root 2.5M Dec  3 10:56 h264vdec_sn.dll64P
-rwxr-xr-x 1 root root 481K Dec  3 10:56 jpegdec_sn.dll64P
-rwxr-xr-x 1 root root 229K Dec  3 10:56 jpegenc_sn.dll64P
-rwxr-xr-x 1 root root 767K Dec  3 10:56 m4venc_sn.dll64P
-rwxr-xr-x 1 root root 890K Dec  3 10:56 mp4vdec_sn.dll64P
-rwxr-xr-x 1 root root 707K Dec  3 10:56 mpeg4aacdec_sn.dll64P
-rwxr-xr-x 1 root root  15K Dec  3 10:56 qosdyn_3430.dll64P
-rwxr-xr-x 1 root root  14K Dec  3 10:56 ringio.dll64P
-rwxr-xr-x 1 root root 9.1K Dec  3 10:56 TSPA_Object_Code_Software_License_Agreement.txt
-rwxr-xr-x 1 root root  53K Dec  3 10:56 usn.dll64P
-rwxr-xr-x 1 root root 245K Dec  3 10:56 vpp_sn.dll64P
```

Building gst-dsp stuff:

If you installed this image via the demo images or netinstall, the gst-dsp build script is installed:

Update arm tools directory to the latest:

```
ubuntu@arm:~$ cd /boot/uboot/tools/
ubuntu@arm:/boot/uboot/tools$ sudo ./update.sh
```

Otherwise, install via git:

```
git clone git://github.com/RobertCNelson/tools.git
```

Change to the pkgs directory

```
ubuntu@arm:~$ cd /boot/uboot/tools/pkgs
(or if just cloned)
ubuntu@arm:~$ cd ~/tools/pkgs
```

Build gst-dsp stuff..

```
./ti-tidspbridge.sh
```

Start dspbridge (or just reboot)

```
sudo /etc/init.d/dsp_init start
```

Test dspbridge

```
sudo dsp-test
```

Playbin:

```
sudo gst-launch playbin2 uri=file://(file)
```

Example: (using http://www.bigbuckbunny.org/index.php/download/ 854x480 mp4)

```
Note: it seems broken in Ubuntu precise armhf...
```

```
sudo gst-launch playbin2 uri=file:///home/USER/big_buck_bunny_480p_surround-fix.avi
```

Building Kernel

Note, this section use to have a lot of details, but maintenance of the two wiki's became a pain, so for now on we will just link to my other pages:

BeagleBone Black

```
http://eewiki.net/display/linuxonarm/BeagleBone+Black
```

Step 7: Upgrade from older versions.

Upgrading from Debian 5 to Debian 6

Upgrading from Debian 5 to Debian 6 report and tutorial.

Upgrading from Debian 6 (Squeeze) to Debian 7 (Wheezy)

Note, this only goes over "armel" -> "armel"... For users wishing to switch from "armel" -> "armhf" just re-install...

```
sudo apt-get update ; sudo apt-get upgrade
sudo sed -i -e "s/squeeze/wheezy/g" /etc/apt/sources.list
sudo apt-get update ; sudo apt-get install -y apt aptitude udev dpkg
sudo aptitude update ; sudo aptitude -y safe-upgrade ; sudo aptitude -y dist-upgrade
```

Related Instructables

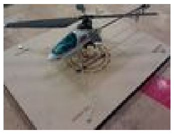
Custom Breadboard Test Leads (Photos) by mpilchfamily

Photo Components & Testing by Josehf Murchison

Toner transfer material print tests (Photos) by The Ideanator

Car Charger To Portable 9vt USB by tdunnick

The Ethical Hacking Process by THAWKK



How to create a test stand for a radio controlled VTOL capable vehicle. by vazquezl31