QNX Software Systems
QNX 6.5.0 SP1 BSP for BeagleBone
User Guide

MPC Data

qnx@mpcdata.com

Revision:	Issue 4
Date:	12 September, 2012
Reference	3220-UD01
Approved:	

Document Change History

Revision	Section	Amendment	Author	Date
Draft 1a	All	Initial version		18th Jan 2012
Draft 1b	Known Issues	Modify List		15th March 2012
Issue 1	All	Updated to Issue 1 after review		15th March 2012
Issue 2a	10.2	Update Network command		3rd April 2012
Draft 3a	All	Updated for QNX 6.5.0 SP1	DHarris	14th Aug 2012
Issue 3	All	Updated for Issue 3	GJones	4th Sept 2012
Draft 4a	9.10, 10.9	Add SD/MMC Driver details	GJones	12th Sept 2012
Issue 4	All	Issued	GJones	12th Sept 2012

© Copyright MPC Data Limited 2012 All trademarks are hereby acknowledged.

Table Of Contents

1. Introduction

This is the Quick Start Guide for the QNX 6.5.0 SP1 BSP for the BeagleBone platform. More information on this platform can be found in the system reference manual (Ref 1).

2. Scope

This revision of the Quick Start Guide describes the BeagleBone for QNX 6.5.0 SP1 produced by MPC Data. This document describes steps to build the BSP and use the included drivers. Further information on developing for QNX can be found on the QNX website (Ref 3)

3. References

1. BeagleBone Reference Manual
   http://beagleboard.org/static/BONESRM_latest.pdf
2. ARM 335x Reference Manual
   http://www.ti.com/litv/pdf/spruh73c
3. Documentation for QNX
   www.qnx.com/developers/docs/
4. QNX instructions for booting TI AM/OMAP based boards
   http://community.qnx.com/sf/wiki/do/viewPage/projects.bsp/wiki/AM_OMAP_boot_resources

4. System Requirements

4.1. Target Requirements

1. Board version: BeagleBone board revision A3.

4.2. Host Requirements

1. QNX Momentics 6.5.0 SP1 SDP
2. Terminal emulation program (Qtalk, Momentics IDE Terminal, HyperTerminal, etc.)
3. USB Cable Standard A plug to Mini B plug
4. SD card and reader on development host
5. Ethernet link

5. System Layout

Start	End	Item
0x40000000	0x7FFFFFFF	Hardware peripherals (See Ref [2])
0x80000000	0x8FFFFFFF	External Memory

6. Getting Started - Building the BSP

The BSP OS image can be built from the source code and binary components contained in this BSP package. For instructions about building a BSP OS image, please refer to the chapter *Working with a BSP* in the QNX *Building Embedded Systems* manual.

7. Connect the Hardware

To control the target board it must be connected to the host PC using a USB Cable Standard A plug to Mini B plug. Load the USB to Serial drivers found in the Drivers folder of the micro SD card provided with the board (BeagleBone Getting Started). This will create a COM port that can be accessed using a normal terminal emulator.
Connect the terminal with the following settings:

baudrate	115200
data	8 bit
parity	None
stop	1 bit
flow control	None

8. Booting the target

8.1. Set up the MMC/SD card

Please note following the steps below will erase all data on the SD card. Ensure data is backed up before continuing!

The target micro SD card should be prepared according to the instructions in reference [4], refer to the section entitled “SD Card Boot”. Then transfer these OS image and bootloader files to the SD card from the host system:

MLO	This is the first stage bootloader (x-loader), which is loaded on startup. It initializes the RAM and loads UBoot.

u-boot.img	This is the second stage bootloader. Make sure you use the image from the "BeagleBone" micro SD, not from the "BeagleBone Getting Started" micro SD.
ifs-ti-beaglebone.bin (built image) or prebuilt-bsp-ti-beaglebone.ifs	The QNX IFS image file , **rename the file on the MMC/SD card to qnx-ifs**

The processor will automatically read the SD/MMC card's FAT filesystem to boot the MLO (x-loader), which will in turn load the UBoot bootloader.

8.2. Boot the image

Enter the following commands at the UBoot prompt to load the OS image.

```
# mmcinfo
# fatload mmc 0 0x81000000 qnx-ifs
# go 0x81000000
```

This will initialize the MMC subsystem, load the IFS from it and branch to its location in RAM. The IFS should then run and finish loading the QNX operating system. The serial will display the following:

```
## Starting application at 0x81000000
```

Please note when you wish to boot a new IFS image, you only need to overwrite the old *qnx-ifs* on the SD card and not redo all the steps in section

To automate copying to memory and running the *qnx-ifs* image, create a file on the micro SD card called *uEnv.txt* and add the following line to it:

```
uenvcmd=mmcinfo;fatload mmc 0 0x81000000 qnx-ifs; go 0x81000000
```

9. Driver Command Summary

9.1. Startup

Buildfile command	startup-beaglebone -v -wb
Required binaries	startup-beaglebone
Required libraries	None
Source location	src/hardware/startup/boards/beaglebone
Notes:	-v enables verbose output (More "v"s increases verbosity)
	-wb enables write-back, read-allocate (no write-allocate) caching

9.2. Serial

Buildfile command	devc-seromap -e -F -b115200 -c48000000/16 0x44E09000^2,72
	reopen /dev/ser1
Required binaries	devc-seromap

Required libraries	libc.so.3
Source location	src/hardware/devc/seromap

9.3. Network

Buildfile command	io-pkt-v4-hc -dam335x deviceindex=0
Required binaries	io-pkt-v4 ifconfig
Required libraries	devnp-am335x.so
Source location	Only released as a prebuilt binary

9.4. Watchdog Timer

Buildfile command	wdtkick
Required binaries	wdtkick
Required libraries	none
Source location	/src/hardware/support/wdtkick

9.5. USB

Buildfile command	io-usb -vvv -d dm816x-mg ioport=0x47401400,irq=18 -d dm816x-mg ioport=0x47401c00,irq=19 waitfor /dev/io-usb/io-usb 4 devb-umass cam pnp
Required binaries	devu-dm816x-mg.so
Required libraries	none
Source location	Only released as a prebuilt binary

9.6. I2C

Buildfile command	i2c-omap35xx-j5 -i 70 -p0x44E0B000 --u0 waitfor /dev/i2c0 i2c-omap35xx-j5 -i 71 -p0x4802A000 --u1 waitfor /dev/i2c1 i2c-omap35xx-j5 -i 30 -p0x4819C000 –u2 waitfor /dev/i2c2
Required binaries	i2c-omap35xx-j5
Required libraries	lib.so.3
Source location	/src/hardware/i2c

9.7. SPI

Buildfile command	SPI0: spi-master -d am335x base=0x48030100,irq=65,edma=1,edmairq=529,edmachannel=17 SPI1: spi-master -d am335x base=0x481A0100,irq=125,edma=1,edmairq=555,edmachannel=43
Required binaries	spi-master
Required libraries	spi-am335x.so
Source location	/src/hardware/spi

9.8. RTC

Buildfile command	rtc rtc –s am335xrtc (saves the current time/date in the hardware)
Required binaries	date
Required libraries	None
Source location	/src/utils/r/rtc

9.9. LED

Buildfile command	dev-leds-beaglebone & waitfor /dev/leds 4
Required binaries	dev-leds-beaglebone
Required libraries	None
Source location	/src/hardware/leds

9.10. SD/MMC

Buildfile command	devb-mmcsd-jacinto5 cam blk automount=hd0t12:/fs/sd0 mmcsd verbose=5,ioport=0x48060100,irq=64,ioport=0x49000000,dma=24,dma=25
Required binaries	devb-mmcsd-jacinto5
Required libraries	libcam.so libc.so.3
Source location	/src/hardware/devb

10. Driver Command Details

For additional information on each command, run the use command followed by the driver name. e.g.

```
use i2c-omap35xx-j5
```

10.1. Serial

Start the driver for the debug UART with this command:

```
devc-seromap -e -F -b115200 -c48000000/16 0x44E09000^2,72
```

10.2. Network

To start network driver, run:

```
io-pkt-v4-hc -dam335x deviceindex=0 verbose
```

You should see following output when you run **ifconfig**

```
# ifconfig
lo0: flags=8049<UP,LOOPBACK,RUNNING,MULTICAST> mtu 33192
        inet 127.0.0.1 netmask 0xff000000
dm0: flags=843<UP,BROADCAST,RUNNING,SIMPLEX> mtu 1500
        address: 00:04:9f:cc:9a:b5
        media: Ethernet autoselect (100baseTX full-duplex)
        status: active
        inet 0.0.0.0 netmask 0xff000000 broadcast 255.255.255.255
```

To manually bring a network interface up, type following command

```
ifconfig dm0 10.150.128.250/24 up
```

To bring up the interface up and automatically acquire an IP address from a DHCP server, enter the following command:

```
dhcp.client
```

Here 10.150.128.250 is the IP address assigned to the target and 24 is the netmask bits (i.e. 255.255.255.0). The LEDs of the RJ45 ports corresponding to an enabled interface shall glow periodically. This can be used to identify the correct port corresponding to each interface.

10.3. USB

The USB host driver can be started using the following commands for both ports.

```
io-usb -vvv -d dm816x-mg ioport=0x47401400,irq=18 -d dm816x-mg
ioport=0x47401c00,irq=19
```

To enable the USB mass storage driver use the following command

```
devb-umass cam pnp &
```

The cam pnp arguments will cause the driver to continue running if it doesn't find any mass storage devices, when any new devices are connected they will be enumerated under /dev.

10.4. SPI

To load an instance of the SPI driver for each bus, use the following commands:

```
SPI 0:
spi-master -d am335x
base=0x48030100,irq=65,edma=1,edmairq=529,edmachannel=17
SPI 1:
spi-master -d am335x
base=0x481A0100,irq=125,edma=1,edmairq=555,edmachannel=43
```

Note that SPI 0 is not enabled by default, as it uses the same pins as I2C1.

10.5. I2C

The I2C driver is required by the board ID and RTC drivers, and so must be loaded before these.

Start the i2c bus 0 interface with the following command:

```
i2c-omap35xx-j5 -i 70 -p0x44E0B000 --u0
```

Start the i2c bus 1 interface with the following command:

```
i2c-omap35xx-j5 -i 71 -p0x4802A000 --u1
```

Start the i2c bus 2 interface with the following command:

```
i2c-omap35xx-j5 -i 30 -p0x4819C000 -u2
```

10.6. RTC

The RTC on the EVM is located within the AM335x processor, and accessed via register writes.

To store the current time & date in the RTC use the following command:

```
rtc -s am335xrtc
```

The kernel clock time and date can be set using the QNX **date** command.

10.7. Board ID

Run the Board Identification utility to view the information stored in the ID I2C EEPROM. This information the BeagleBone serial number and hardware revision

To view the board ID information use the following command:

```
bdid-am335x
```

10.8. LED

Allows the user to change the status of the onboard LEDs.

```
dev-leds-beaglebone &
```

The LEDs are set to match the bit values in the lowest four bits of the work sent to it.

For example

```
echo 9 > /dev/leds
```

Will turn on the first and last LEDs.

10.9. SD/MMC

Allows the user to access the micro SD card.

```
devb-mmcsd-jacinto5 cam blk automount=hd0t12:/fs/sd0 mmcsd
verbose=5,ioport=0x48060100,irq=64,ioport=0x49000000,dma=24,dma=25
```

The above command will start the SD/MMC drivers and automount a type 12 partition if one exists on the card. Other partition types will appear in */dev*, and can be mounted as required using the *mount* command.

11. Capes

The BSP builds with a default set of pins connected to the expansion ports. These can be modified as the user requires by changing the *cape_profile*[] structure in *src/startup/boards/beaglebone/init_pinmux.c* and the call to *init_pinmux_capes* in *src/startup/boards/beaglebone /main.c*

12. Hints and Tips

1. The build file starts a DHCP client when the system boots. It is configured to acquire an address in the background. The boot process then waits for 15 seconds before continuing.

13. Resolved Issues

None

14. Known Issues

1. POR0000155: QNX BSP doesn't boot into QNX when using UBoot from the Angstrom SD/MMC card.

2. POR0000157: The RTC state is lost when power is removed from the board, and battery backup is not available