Freescale Semiconductor
Application Note

Document Number: AN3947
Rev. 0, 11/2009

How to Run the Latest Linux BSP on MPC8313ERDB Rev. Ax Boards

by: Shu Yinbo
System and Application Engineer
Beijing China

1 Introduction

The MPC8313E reference design board (RDB) is a system featuring the PowerQUICC™ II Pro processor that includes a built-in security accelerator. The software board supported package (BSP) for this low-cost, high-performance system solution is available on the Freescale website. The BSP enables the fastest possible time-to-market for development or integration of applications, including printer engines, broadband gateways, no-new-wires home adapters and access points, and home automation boxes.

This document describes how to port the latest BSP to the old hardware platform to use the features added in the latest BSP. It details what you need to know and how to make changes to the latest BSP to match older boards. In the end, the eTSEC and preemptible kernel supports are verified on the MPC8313E-RDB Rev. A4 board.

Contents

2 MPC8313E-RDB Overview

The MPC8313E-RDB reference platform is ideal for hardware and software development for cost-optimized networking applications. The cost-effective MPC8313E communications processor family meets the requirements of several small offices and home offices (SOHO), printing, IP services, and industrial control applications. It leverages Freescale's highly integrated MPC8313E processor built on Power Architecture™ technology and leading-edge external components: a 5-port Gigabit Ethernet switch, USB port, one peripheral component interconnect (PCI) slot, one miniPCI slot, and one SD card memory slot. The high level integration in the MPC8313E helps to lower overall system costs, improve performance, and simplify board design.

This section lists the features of the MPC8313E-RDB.

- CPU—Freescale MPC8313E running at 333/166 MHz; CPU/coherent system bus (CSB)
- Memory subsystem:
  - 128 MB unbuffered DDR2 SDRAM discrete devices
  - 8 MB flash single-chip memory
  - 32 MB NAND flash memory
  - 256 kbit M24256 serial EEPROM
  - SD connector to interface with the SD memory card in SPI mode
- Interfaces:
  - 10/100/1000 BaseT Ethernet ports:
    - eTSEC1, RGMII—Five 10/100/1000 BaseT RJ-45 interfaces using Vitesse™ VSC7385 L2 switch, or one 10/100/1000 BaseT RJ-45 interface using Marvell™ 88E1111 PHY in REVC board
    - eTSEC2, selectable RGMII or SGMII—One 10/100/1000 BaseT RJ-45 interface using Marvell 88E1111 PHY
  - USB 2.0 port—High-speed host and device
  - USB interface—Selectable on-chip PHY or external ULPI PHY interface by SMSC USB3300 USB PHY
  - PCI—32-bit PCI interface running at up to 66 MHz
    - One 32-bit 3.3 V PCI slot connected to the PCI bus
    - One 32-bit 3.3 V miniPCI slot connected to the PCI bus
  - Dual UART ports:
    - DUART interface—Offices supports two UARTs up to 115200 bps for console display
- Board connectors:
  - LCD connectors by GPIO
  - ATX power supply connector
  - JTAG/COP for debugging
- IEEE Std. 1588™ signals for test and measurement
- Real-time clock and thermal sensor on I^{2}C bus

- Programmable LEDs for debug use
- 6-layer PCB routing (4-layer signals, 2-layer power and ground)

3 Preparations

3.1 Host Preparations

3.1.1 System Requirements

A Linux PC must be prepared to use and port the BSP. The LTIB can run on most Linux host distributions that have glibc-2.2.x or later. Any of the following platforms can be used:

- Redhat: 7.3, 8.0, 9.0, Enterprise release 4, Enterprise release 5
- Fedora Core: 1, 2, 3, 4, 5, 8
- Debian: 3.1r0
- Suse: 8.2, 9.1, 9.2, 10.0, 10.2, 10.3
- Ubuntu: 6.10, 7.04, 7.10, 8.04

3.1.2 BSP Preparations

This porting guide applies to both the 20081216 and 20080613 BSP. You can download the BSP from the Freescale website and install either of them on your PC.

After the installation, extract the kernel source using the following command:

```
./ltib –m prep –p kernel-2.6.23-mpc8313erdb_revc.spec
```

3.2 Porting Background

From the beginning you must know the board differences and the why the latest BSP can not run on the older boards.The modifications to the BSP correspond to the changes of the board revisions.

The main changes of the board are provided in Table 1.

Table 1. MPC8313E-RDB Changes

Revision	Descriptions	MPC8313 Chip Revision
REVA	There are two major issues on the REVA board: • On-chip PHY USB signals (DP, DM) are swapped. To use the USB, use a USB cable that swaps the signals (the cable is attached in the REVA package). • NAND flash memory cannot be used as a boot device.	1.0
REVA1	Fixed both major issues on the REVA board. The boot-from-NAND on the REVA1 -RDB has been verified. However, on the current BSP preloaded on REVA1 RDB, the NAND flash memory is empty, and not bootable. Booting from the NAND flash memory will be supported in a future release of the BSP. Software for REVA and REVA1 differs only in the OR1[BCTLD] register setting for NAND flash memory. That is, REVA OR1[BCTLD] is 1; while REVA1 OR1[BCTLD] is 0.	1.0
REVA2	A minor update from REVA1 for mass production. It updated the silkscreen and added a 12-V fan connector (J25) and resistor loading for ATX power. Software can be shared without modification between REVA1 and REVA2.	1.0

Table 1. MPC8313E-RDB Changes (continued)

REVA3	The fixed PMC register issue power management control (PMC) registers cannot be accessed. According to the processor erratum, a 166 MHz CSB frequency must be used. For this reason, some REVA3 and all later boards use 33 MHz instead of 66 MHz as the clock input (check your board U15 oscillator marking). The CORE/CSB/DDR frequency setting is 333/166/333 MHz. However, there are two drawbacks: • PCI bus can run at up to only 33 MHz • PCI/mini-PCI card can run at 66 MHz (with its M66EN pulled up) and must be used. Even the PCI bus on the RDB runs at only 33 MHz. Otherwise, the PCI frequency is further divided and it becomes 16.6 MHz.	1.0
REVA4	Fixed the second drawback issue of REVA3.	1.0
REVB	• Added GTX_CLK125 sourced from the external 125 MHz oscillator. • Added an optional IEEE 1588 connector (P10). • Added three more resistor options (R31–R313) to route three IEEE 1588 signals that are only available in eTSEC1 to the IEEE 1588 connector. • Changed S4 to support LB_POR_CFG_BOOT_ECC_DIS. • Changed SD chip selection signal from SPISEL (GPIO31) to GPIO13.	2.0
REVC	• Added a Marvell 88E1111 PHY. Phy address was assigned to 0x3. Used the same IRQ3 number as the L2 switch. • Added resistor option for RGMII signals route to either L2 switch or Marvell 88E1111 PHY. • Added SGMII support for eTSEC1 if using the added Marvell 88E1111 PHY. (SGMII for eTSEC2 was already supported.) • Added PLL CY23EP05SXC-1 U86 to PHY generated 125 MHz clock. • Changed default TSEC1_GTX_CLK125 clock source to PLL CY23EP05SX-1 instead of the external 125 MHZ oscillator. • Changed U36 1A linear regulator MIC39100-2.5WS to 3A MIC37302WR for higher 2.5V power consumption by additional PHY. • Changed default DAC to 16-bit SPI controlled MAX5203BEUB+ (U47).	2.1

From the table, the Rev. Ax boards use the Rev. 1.0 silicon, Rev. B and Rev. C boards use the Rev. 2.x silicon. For details of the silicon changes, please refer to application note *Migration from MPC8313E Revision 1.0 to Revision 2.x* (document AN3545)

The BSP difference is provided in the Appendix B, “Comparison of the BSP Releases.” You may refer to them for comparison of the BSP releases like the feature added, the bug fixed.

4 DTS Introduction

4.1 Device Tree Generalities

The flat device tree whose format is defined after the Open Firmware specification maintains the kernel entry and bootloader <-> kernel interfaces. It is popularly used in the embedded Linux development today.

This device-tree itself is separated in two blocks, a structure block and a strings block. Both need to be aligned to a 4 byte boundary.

First, described here is the device-tree concept. The device-tree layout is strongly inherited from the definition of the Open Firmware IEEE 1275 device-tree. It is basically a tree of nodes, each node has two or more named properties. A property can have a value or not. It is a tree, therefore each node has only one parent except for the root node who has no parent. A node has two names. The actual node name is generally contained in a property of type "name" in the node property list whose value is a zero terminated string and is mandatory for version 1–3 of the format definition (it is in Open Firmware). Version 16 makes it optional, it can generate it from the unit name defined below. There is also a "unit name" that is used to differentiate nodes with the same name at the same level, it is made up of the node names, the "@" sign, and a "unit address", whose definition is specific to the bus type the node sits on. The unit name does not exist as a property, but it is included in the device-tree structure. It is typically used to represent a "path" in the device-tree. More details about the actual format of these are discussed later.

The kernel PowerPC generic code does not formally use the unit address (though some boards supported code may). The only requirement for the unit address is to ensure uniqueness of the node unit name at a given level of the tree. Nodes without addresses nor possible sibling of the same name (like /memory or /cpus) omit the unit address in the context of this specification, or use the "@0" default unit address. The unit name is used to define a node "full path" that is the concatenation of all parent node unit names separated with "/".

The root node does not have a defined name, and a name property is not required if you are using version 3 or an earlier format. It also has no unit address (no @ symbol followed by a unit address). The root node unit name is therefore an empty string. The full path to the root node is "/".

Every node that represents an actual device (that is, a node that is not only a virtual "container" for more nodes, like "/cpus") is also required to have a "device_type" property indicating the type of node.

Normally, every node that can be referenced from a property in another node is required to have a "linux,phandle" property. Real open firmware implementations provide a unique "phandle" value for every node that the "prom_init()" trampoline code turns into "linux,phandle" properties. However, this is made optional if the flattened device tree is used directly. An example of a node referencing another node via a "phandle" is presented below when laying out the interrupt tree. This "linux,phandle" property is a 32-bit value that uniquely identifies a node. You are free to use any values or system of values, internal pointers, or whatever to generate these, the only requirement is that every node you provide that property have a unique value.

Here is an example of a simple device-tree. In this example, an "o" designates a node followed by the node unit name. Properties are presented with their name followed by their content. The "content" represents an ASCII string (zero terminated) value, while <content> represents a 32-bit hexadecimal value. The various nodes in this example are discussed in the standard. At this point, it gives you an idea of what a device-tree looks like. The "name" and "linux,phandle" properties that are not necessary are presented to give you a better idea of what the tree looks like in practice.

```
/ o device-tree
|- name = "device-tree"
|- model = "MyBoardName"
|- compatible = "MyBoardFamilyName"
|- #address-cells = <2>
|- #size-cells = <2>
|- linux,phandle = <0>
|
o cpus
| | - name = "cpus"
| | - linux,phandle = <1>
| | - #address-cells = <1>
| | - #size-cells = <0>
| |
| o PowerPC,970@0
| |- name = "PowerPC, 970"
| |- device_type = "cpu"
| |- reg = <0>
| |- clock-frequency = <5f5e1000>
| |- 64-bit
| |- linux,phandle = <2>
|
o memory@0
| |- name = "memory"
| |- device_type = "memory"
| |- reg = <00000000 00000000 00000000 20000000>
| |- linux,phandle = <3>
|
o chosen
|- name = "chosen"
|- bootargs = "root=/dev/sda2"
|- linux,phandle = <4>
```

This tree is almost a minimal tree. It contains the minimal set of required nodes and properties to boot a linux kernel. For example, some basic model information at the root, the CPUs, and the physical memory layout. It also includes miscellaneous information passed through and chosen, like in this example, the platform type (mandatory) and the kernel command line arguments (optional).

The /cpus/PowerPC,970@0/64-bit property is an example of a property without a value. All other properties have a value. The significance of the #address-cells and #size-cells properties are explained in the booting_without_of.txt that defines precisely the required nodes, properties, and their content.

The flat device tree creates a such flexibility that the kernel can then probe and match drivers to the device without hard coding all sorts of tables. It also makes it more flexible for board vendors to do minor hardware upgrades without significantly impacting the kernel code or cluttering it with special cases.

4.2 eTSEC and PHY Nodes

Due to evolvement of the MPC8313 silicon and evaluation boards, the interrupt mapping of the eTSEC ports have to be revisited to make it work on older boards. The following section presents a brief introduction to the related parts of the device tree.

4.2.1 MDIO IO Device

The MDIO is a bus that the PHY devices are connected to. For each device that exists on this bus, a child node must be created. Take a close look at the explanation below for defining the PHY node in the device tree.

Required properties:

- reg—Offset and length of the register set for the device
- device_type—Must be "mdio"
- compatible—Must define the compatible device type for the mdio. Currently, this is most likely to be "gianfar"

Example:

```
mdio@24520 {
        reg = <24520 20>;
        device_type = "mdio";
        compatible = "gianfar";

        ethernet-phy@0 {
                ......
        };
};
```

4.2.2 Gianfar-Compatible Ethernet Nodes

Required properties:

- device_type—Must be "network"
- model—Model of the device. Can be "TSEC", "eTSEC", or "FEC"
- compatible—Must be "gianfar"
- reg—Offset and length of the register set for the device
- mac-address—List of bytes representing the Ethernet address of this controller
- interrupts— <a b> where a is the interrupt number and b is a field that represents an encoding of the sense and level information for the interrupt.
- interrupt-parent—The phandle for the interrupt controller that services interrupts for this device.
- phy-handle—The phandle for the PHY connects to this Ethernet controller. It specifies a reference to a node representing a physical layer (PHY) device connected to this Ethernet device. This property is required in case the Ethernet device is connected to a physical layer device.

Recommended properties:

- linux_network-index—This is the intended "index" of this network device. This is used by the bootwrapper to interpret MAC addresses passed by the firmware when no information other than the index is available to associate an address with a device.

Example:

```
ethernet@24000 {
        #size-cells = <0>;
        device_type = "network";
        model = "TSEC";
        compatible = "gianfar";
        reg = <24000 1000>;
        mac-address = [ 00 E0 0C 00 73 00 ];
        interrupts = <d 3 e 3 12 3>;
        interrupt-parent = <40000>;
        phy-handle = <2452000>
};
```

4.2.3 PHY Nodes

Required properties:

- device_type—Must be "ethernet-phy"
- interrupts— <a b> where a is the interrupt number and b is a field that represents an encoding of the sense and level information for the interrupt.
- interrupt-parent—the phandle for the interrupt controller that services interrupts for this device.
- reg—The ID number for the PHY, usually a small integer
- linux,phandle—phandle for this node; likely referenced by an Ethernet controller node.

Example:

```
ethernet-phy@0 {
        linux,phandle = <2452000>
        interrupt-parent = <40000>;
        interrupts = <35 1>;
        reg = <0>;
        device_type = "ethernet-phy";
};
```

5 Porting Device Tree to Rev. Ax boards

As the 20081218 BSP is applied to the Rev. Ax board, the REV Ax boards host the MPC8313E CPU 1.0; The 20081218 Linux BSP was the final release for the MPC8313E-RDB Rev. C board that features the MPC8313E CPU 2.1. The IPIC1 defect was fixed in revision 2.1 this means the interrupt numbers of the eTSEC are different from revision 1.0. So the Linux system freezes or reports errors if you are trying to issue a command such as ping or ifconfig while the latest BSP on the older boards. To avoid such error, you must modify the device tree file named "mpc8313erdb.dts" under "/arch/powerpc/boot/dts" to set correct interrupt vectors for Rev. Ax boards and use "dtc" under "/opt/freescale/ltib/usr/bin" to compile it.

The related parts of mpc8313erdb.dts are changed to the following to make it.

```
mdio@24520 {
        device_type = "mdio";
        compatible = "gianfar";
        reg = <24520 20>;
        #address-cells = <1>;
        #size-cells = <0>;
        phy4: ethernet-phy@4 {
                interrupt-parent = < &ipic >;
                interrupts = <14 2>;
                reg = <4>;
                device_type = "ethernet-phy";
        };

ethernet@24000 {
        device_type = "network";
        model = "eTSEC";
        compatible = "gianfar";
        reg = <24000 1000>;
        ptimer-handle = < &ptimer >;
        local-mac-address = [ 00 00 00 00 00 00 ];

        interrupts = <25 8 24 8 23 8>;
        interrupt-parent = < &ipic >;
        fixed-link = <1 1 d#1000 0 0>;
        sleep = <b00 20000000>;
        fsl,magic-packet;
};

ethernet@25000 {
        device_type = "network";
        model = "eTSEC";
        compatible = "gianfar";
        reg = <25000 1000>;
        ptimer-handle = < &ptimer >;
        local-mac-address = [ 00 00 00 00 00 00 ];
        interrupts = <22 8 21 8 20 8>;
        interrupt-parent = < &ipic >;
        phy-handle = < &phy4 >;
        sleep = <b00 10000000>;
        fsl,magic-packet;
};
```

The device tree file (dts) has to be converted into the binary format device tree file (dtb) so that the Linux kernel can parse it later:

Invoke the device tree compiler to compile dts to dtb format as follows:

~#: /opt/freescale/ltib/usr/bin/dtc -f -b 0 -I dts -O dtb -R 8 -S 0x3000 -o mpc8313erdb.dtb mpc8313erdb.dts

Without touching anything else, the latest Linux BSP works fine on the Rev. Ax boards. However, due to the hardware changes, the SD card support is unavailable on Rev. Ax boards. The details of the Linux deployment are presented in the next section.

6 Running U-Boot and Kernel on the MPC8313E-RDB Rev. A4 board

6.1 Flashing U-Boot

First, the u-Boot is required to be programmed to the target board. If a u-Boot already exists in the NOR flash, either the u-Boot or flash programmer can be used to update itself. If there is no u-Boot, you have to resort to the flash programming tool like the Flash Programmer of CodeWarrior for Power Architecture v8.8.

6.1.1 Updating U-Boot

Download the u-Boot image to the DDR RAM on the board via tftp. Please refer to the instructions below.

```
=> tftp 100000 /tftpboot/u-boot.bin
=> protect off all
=> erase fe000000 fe06ffff
=> cp.b 100000 fe000000 $filesize
=> protect on all
```

6.1.2 Updating U-Boot using CodeWarrior Flash Programmer

Please ensure the board jumper settings are correct against the manual.

http://www.freescale.com/files/32bit/doc/user_guide/MPC8313ERDBUG.pdf

Then, following the steps below to update the u-Boot with the CodeWarrior Flash Programmer.

Steps:

1. Go to Tools>Flash Programmer
2. Click “Load settings”, locate and load the file 8313RDB_NOR_FLASH.xml in the $\Freescale\CodeWarrior PA V8.8 \bin\Plugins\Support\Flash_Programmer\EPPC\83xx
3. Under “Target Configuration” make sure that Connection is “CodeWarrior USB TAP”
4. Under “Flash Configuration” make sure that the Flash Memory Address is 0xff800000
5. Under “Erase/Blank Check”, (de-select all sectors) select the first four sectors (from ff800000 to ff83ffff) and click Erase. Hold shift to select individual sectors.
6. Under “Program/Verify”, click “Use Selected File” and browse for $\images\u-boot.bin
7. Select File Type: Binary/Raw Format
8. Check Restrict Address Range Start: 0xff800000 End: 0xffffffff
9. Check Apply Address Offset and enter 0xff8000000
10. Click on “Program”
11. After the programming is completed, exit the Flash Programmer, by clicking on **Close**. When prompted to save the settings, click on No.

6.2 Configuring the U-Boot for NFS Deployment

After the u-Boot is flashed or re-flashed, please refer to the BSP manual to build your kernel and configure the host. Then, copy the kernel and the dtb file to the /tftpboot directory of the server for downloading later.

Set the boot parameters using the setenv command.

```
setenv ipaddr 10.192.208.245
setenv gatewayip 10.192.208.254
setenv serverip 10.192.208.120
setenv netmask 255.255.255.0
setenv bootargs nfsroot=10.192.208.120:/tftpboot/10.192.208.245
ip=10.192.208.245:10.192.208.120:10.192.208.254:255.255.255.0:mpc8313erdb:eth1:off root=
dev/nfs rw console=ttyS0,115200
saveenv

u-boot information:

U-Boot 1.3.0 (Dec 22 2008 - 11:19:29) MPC83XX

Reset Status: Software Hard, External/Internal Soft, External/Internal Hard

CPU:   e300c3, MPC8313E, Rev: 10 at 333.333 MHz, CSB:  166 MHz
Board: Freescale MPC8313ERDB
I2C:   ready
DRAM:  128 MB
FLASH:  8 MB
NAND:  32 MiB
In:    serial
Out:   serial
Err:   serial
Net:   TSEC0, TSEC1 [PRIME]
=> printenv
ramboot=setenv rootdev /dev/ram;run setbootargs;tftp $ramdiskaddr $ramdiskfile;tftp $loadaddr
$bootfile;tftp $fdtaddr $fdtfile;bootm $loadaddr $ramdiskaddr $fdtaddr
nfsboot=setenv rootdev /dev/nfs;run setbootargs;run setipargs;tftp $loadaddr $bootfile;tftp
$fdtaddr $fdtfile;bootm $loadaddr - $fdtaddr
bootdelay=6
baudrate=115200
loads_echo=1
ethaddr=00:E0:0C:00:95:01
eth1addr=00:E0:0C:00:95:02
hostname=mpc8313erdb
loadaddr=200000
netdev=eth1
ethprime=TSEC1
uboot=u-boot.bin
tftpflash=tftpboot $loadaddr $uboot; protect off 0xFE000000 +$filesize; erase 0xFE000000
+$filesize; cp.b $loadaddr 0xFE000000 $filesize; protect on 0xFE000000 +$filesize; cmp.b
$loadaddr 0xFE000000 $filesize
fdtaddr=400000
fdtfile=mpc8313erdb.dtb
console=ttyS0
setbootargs=setenv bootargs root=$rootdev rw console=$console,$baudrate $othbootargs
setipargs=setenv bootargs nfsroot=$serverip:$rootpath
ip=$ipaddr:$serverip:$gatewayip:$netmask:$hostname:$netdev:off root=$rootdev rw
console=$console,$baudrate $othbootargs
```

```
bootfile=8313uImage
ipaddr=10.192.208.245
gatewayip=10.192.208.254
serverip=10.192.208.120
netmask=255.255.255.0
rootpath=/tftpboot/10.192.208.245
rootdev=/dev/nfs
bootargs=nfsroot=10.192.208.120:/tftpboot/10.192.208.245
ip=10.192.208.245:10.192.208.120:10.192.208.254:255.255.255.0:mpc8313erdb:eth1:off
root=/dev/nfs rw console=ttyS0,115200
stdin=serial
stdout=serial
stderr=serial
ethact=TSEC1

Environment size: 1394/8188 bytes
```

The Linux boot log is attached in appendix for your information.

7 References

Application note *Migration from MPC8313E Revision 1.0 to Revision 2.x.* (document AN3545)

Power_ePAPR_APPROVED_v1.0 Power.org™ Standard for Embedded Power Architecture™ Platform Requirements (ePAPR)

User guide *PowerQUICC MPC8313E Reference Design Board (RDB)* (document MPC8313ERDBUG)

Linux BSP for Freescale MPC8313—Release Notes, (MPC8313E_RDB_BSP_User_Manual-20081226)

BSP installation directory— $/rpm/BUILD/linux/Documentation/ powerpc/booting-without-of.txt

MPC8313E PowerQUICC™ II Pro Integrated Host Processor Device Errata (document MPC8313ECE)

Appendix A Booting Log

```
U-Boot 1.3.0 (Dec 22 2008 - 11:19:29) MPC83XX

Reset Status: Software Hard, External/Internal Soft, External/Internal Hard

CPU:   e300c3, MPC8313E, Rev: 10 at 333.333 MHz, CSB:  166 MHz
Board: Freescale MPC8313ERDB
I2C:   ready
DRAM:  128 MB
FLASH:  8 MB
NAND:  32 MiB
In:    serial
Out:   serial
Err:   serial
Net:   TSEC0, TSEC1 [PRIME]
=> printenv
ramboot=setenv rootdev /dev/ram;run setbootargs;tftp $ramdiskaddr $ramdiskfile;tftp $loadaddr
$bootfile;tftp $fdtaddr $fdtfile;bootm $loadaddr $ramdiskaddr $fdtaddr
nfsboot=setenv rootdev /dev/nfs;run setbootargs;run setipargs;tftp $loadaddr $bootfile;tftp
$fdtaddr $fdtfile;bootm $loadaddr - $fdtaddr
bootdelay=6
baudrate=115200
loads_echo=1
ethaddr=00:E0:0C:00:95:01
eth1addr=00:E0:0C:00:95:02
hostname=mpc8313erdb
loadaddr=200000
netdev=eth1
ethprime=TSEC1
uboot=u-boot.bin
tftpflash=tftpboot $loadaddr $uboot; protect off 0xFE000000 +$filesize; erase 0xFE000000
+$filesize; cp.b $loadaddr 0xFE000000 $filesize; protect on 0xFE000000 +$filesize; cmp.b
$loadaddr 0xFE000000 $filesize
fdtaddr=400000
fdtfile=mpc8313erdb.dtb
console=ttyS0
setbootargs=setenv bootargs root=$rootdev rw console=$console,$baudrate $othbootargs
setipargs=setenv bootargs nfsroot=$serverip:$rootpath
ip=$ipaddr:$serverip:$gatewayip:$netmask:$hostname:$netdev:off root=$rootdev rw
console=$console,$baudrate $othbootargs
bootfile=8313uImage
ipaddr=10.192.208.245
gatewayip=10.192.208.254
serverip=10.192.208.120
netmask=255.255.255.0
rootpath=/tftpboot/10.192.208.245
rootdev=/dev/nfs
bootargs=nfsroot=10.192.208.120:/tftpboot/10.192.208.245
ip=10.192.208.245:10.192.208.120:10.192.208.254:255.255.255.0:mpc8313erdb:eth1:off
root=/dev/nfs rw console=ttyS0,115200
stdin=serial
stdout=serial
stderr=serial
ethact=TSEC1

Environment size: 1394/8188 bytes
```

```
=> tftp $loadaddr $bootfile
Speed: 100, full duplex
Using TSEC1 device
TFTP from server 10.192.208.120; our IP address is 10.192.208.245
Filename '8313uImage'.
Load address: 0x200000
Loading: #################################################################
#################################################################
#################################################################
#################################################################
#################################################################         ##
done
Bytes transferred = 1672264 (198448 hex)
=> tftp $fdtaddr $fdtfile
Speed: 100, full duplex
Using TSEC1 device
TFTP from server 10.192.208.120; our IP address is 10.192.208.245
Filename 'mpc8313erdb.dtb'.
Load address: 0x400000
Loading: ###
done
Bytes transferred = 12288 (3000 hex)
=> bootm $loadaddr - $fdtaddr
## Booting image at 00200000 ...
   Image Name:   Linux-2.6.23
   Created:      2009-08-03   3:44:46 UTC
   Image Type:   PowerPC Linux Kernel Image (gzip compressed)
   Data Size:    1672200 Bytes =  1.6 MB
   Load Address: 00000000
   Entry Point:  00000000
   Verifying Checksum ... OK
   Uncompressing Kernel Image ... OK
   Booting using the fdt at 0x400000
Using MPC8313 RDB machine description
Linux version 2.6.23 (shu@bach-desktop) (gcc version 4.1.2) #3 PREEMPT Mon Aug 3 11:44:44 CST 
2009
console [udbg0] enabled
setup_arch: bootmem
mpc8313_rdb_setup_arch()
Found MPC83xx PCI host bridge at 0x00000000e0008500. Firmware bus number: 0->0
arch: exit
Zone PFN ranges:
  DMA             0 ->    32768
  Normal      32768 ->    32768
Movable zone start PFN for each node
early_node_map[1] active PFN ranges
    0:        0 ->    32768
Built 1 zonelists in Zone order.  Total pages: 32512
Kernel command line: nfsroot=10.192.208.120:/tftpboot/10.192.208.245 
ip=10.192.208.245:10.192.208.120:10.192.208.254:255.255.255.0:mpc8313erdb:eth1:off 
root=/dev/nfs rw console=ttyS0,115200
IPIC (128 IRQ sources) at fdef9700
PID hash table entries: 512 (order: 9, 2048 bytes)
Dentry cache hash table entries: 16384 (order: 4, 65536 bytes)
Inode-cache hash table entries: 8192 (order: 3, 32768 bytes)
Memory: 126148k/131072k available (3348k kernel code, 4772k reserved, 148k data, 94k bss, 152k 
init)
```

```
Mount-cache hash table entries: 512
NET: Registered protocol family 16

PCI: Probing PCI hardware
Generic PHY: Registered new driver
SCSI subsystem initialized
usbcore: registered new interface driver usbfs
usbcore: registered new interface driver hub
usbcore: registered new device driver usb
NET: Registered protocol family 2
IP route cache hash table entries: 1024 (order: 0, 4096 bytes)
TCP established hash table entries: 4096 (order: 3, 32768 bytes)
TCP bind hash table entries: 4096 (order: 2, 16384 bytes)
TCP: Hash tables configured (established 4096 bind 4096)
TCP reno registered
JFFS2 version 2.2. (NAND)  2001-2006 Red Hat, Inc.
io scheduler noop registered
io scheduler anticipatory registered (default)
io scheduler deadline registered
io scheduler cfq registered
Serial: 8250/16550 driver $Revision: 1.90 $ 4 ports, IRQ sharing disabled
serial8250.0: ttyS0 at MMIO 0xe0004500 (irq = 18) is a 16550A
console handover: boot [udbg0] -> real [ttyS0]
serial8250.0: ttyS1 at MMIO 0xe0004600 (irq = 19) is a 16550A
RAMDISK driver initialized: 16 RAM disks of 32768K size 1024 blocksize
loop: module loaded
Intel(R) PRO/1000 Network Driver - version 7.3.20-k2-NAPI
Copyright (c) 1999-2006 Intel Corporation.
Gianfar MII Bus: probed
eth0: Gianfar Ethernet Controller Version 1.3-skbr, 00:e0:0c:00:95:01
GFAR: SKB Handler initialized at CPU#0(max=32)
eth0: MTU = 1500 (frame size=1526, truesize=1800)
eth0: Running with NAPI enabled
eth0: 64/64 RX/TX BD ring size
eth1: Gianfar Ethernet Controller Version 1.3-skbr, 00:e0:0c:00:95:02
GFAR: SKB Handler initialized at CPU#0(max=32)
eth1: MTU = 1500 (frame size=1526, truesize=1800)
eth1: Running with NAPI enabled
eth1: 64/64 RX/TX BD ring size
e100: Intel(R) PRO/100 Network Driver, 3.5.23-k4-NAPI
e100: Copyright(c) 1999-2006 Intel Corporation
Marvell 88E1101: Registered new driver
Marvell 88E1112: Registered new driver
Marvell 88E1111: Registered new driver
Marvell 88E1145: Registered new driver
Fixed MDIO Bus: probed
nor: Found 1 x16 devices at 0x0 in 16-bit bank
 Amd/Fujitsu Extended Query Table at 0x0040
nor: CFI does not contain boot bank location. Assuming top.
number of CFI chips: 1
cfi_cmdset_0002: Disabling erase-suspend-program due to code brokenness.
RedBoot partition parsing not available
physmap-flash nor: Using OF partition information
Creating 5 MTD partitions on "nor":
0x00000000-0x00100000 : "U-Boot"
0x00100000-0x00300000 : "Kernel"
0x00300000-0x00700000 : "Ramdisk"
```

```
0x00700000-0x00780000 : "DTB"
0x00780000-0x00800000 : "JFFS2"
Freescale eLBC NAND Driver (C) 2006-2007 Freescale
NAND device: Manufacturer ID: 0xec, Chip ID: 0x75 (Samsung NAND 32MiB 3,3V 8-bit)
Scanning device for bad blocks
fsl-elbc fsl-elbc.0: Using OF partition information
Creating 6 MTD partitions on "nand":
0x00000000-0x00100000 : "U-Boot-NAND"
0x00100000-0x00900000 : "JFFS2-NAND"
0x00900000-0x00d00000 : "Ramdisk-NAND"
0x00d00000-0x01d00000 : "Reserve-NAND"
0x01d00000-0x01f00000 : "Kernel-NAND"
0x01f00000-0x02000000 : "DTB-NAND"
mpc83xx_spi.0: MPC83xx SPI Controller driver at 0xc9066000 (irq = 21)
usbmon: debugfs is not available
fsl-ehci fsl-ehci.0: Freescale On-Chip EHCI Host Controller
fsl-ehci fsl-ehci.0: new USB bus registered, assigned bus number 1
fsl-ehci fsl-ehci.0: irq 38, io base 0xe0023000
fsl-ehci fsl-ehci.0: USB 2.0 started, EHCI 1.00, driver 10 Dec 2004
usb usb1: configuration #1 chosen from 1 choice
hub 1-0:1.0: USB hub found
hub 1-0:1.0: 1 port detected
Initializing USB Mass Storage driver...
usbcore: registered new interface driver usb-storage
USB Mass Storage support registered.
i2c /dev entries driver
rtc-ds1307 0-0068: rtc core: registered ds1339 as rtc0
WDT driver for MPC83xx initialized. mode:reset timeout=65535 (25 seconds)
mmc_spi spi28672.0: SD/MMC host mmc0, no DMA, no WP, no poweroff
TCP cubic registered
NET: Registered protocol family 1
NET: Registered protocol family 17
rtc-ds1307 0-0068: setting the system clock to 2000-01-09 00:14:24 (947376864)
IP-Config: Complete:
      device=eth1, addr=10.192.208.245, mask=255.255.255.0, gw=10.192.208.254,
     host=mpc8313erdb, domain=, nis-domain=(none),
     bootserver=10.192.208.120, rootserver=10.192.208.120, rootpath=
Looking up port of RPC 100003/2 on 10.192.208.120
PHY: e0024520:04 - Link is Up - 100/Full
Looking up port of RPC 100005/1 on 10.192.208.120
VFS: Mounted root (nfs filesystem).
Freeing unused kernel memory: 152k init
Setting the hostname to mpc8313erdb
Mounting filesystems
Running sysctl
Setting up networking on loopback device:
Setting up networking on eth0:
You need to manually set your nameserver in /etc/resolv.conf
Starting inetd:


        Welcome to Freescale Semiconductor Embedded Linux Environment

!!!!! WARNING !!!!!!!

The default password for the root account is: root
please change this password using the 'passwd' command
```

```
and then edit this message (/etc/issue) to remove this message

mpc8313erdb login: root
Password:
login[787]: root login  on `console'

~ #


~ # ifconfig
eth1      Link encap:Ethernet  HWaddr 00:E0:0C:00:95:02
          inet addr:10.192.208.245  Bcast:10.192.208.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:2634 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1456 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:2566797 (2.4 Mb)  TX bytes:223972 (218.7 Kb)
          Base address:0x2000

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 b)  TX bytes:0 (0.0 b)

~ # ping 10.192.208.120
PING 10.192.208.120 (10.192.208.120): 56 data bytes
84 bytes from 10.192.208.120: icmp_seq=0 ttl=64 time=0.2 ms
84 bytes from 10.192.208.120: icmp_seq=1 ttl=64 time=0.2 ms
84 bytes from 10.192.208.120: icmp_seq=2 ttl=64 time=0.2 ms

--- 10.192.208.120 ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max = 0.2/0.2/0.2 ms
~ #
```

Appendix B Comparison of the BSP Releases

The BSP changes of each revision are presented in Table 2.

Table 2. MPC8313E-RDB BSP Changes

Revision	Description	Corresponding boards
20081226	BSP new features compared to the prior release (20080613 version): --------------- • Final release for Rev 2.1 silicon on Rev C board • Hardware platform: — Support for MPC8313E CPU 2.1 (PVR = 0x80850010, SVR = 0x80B00021) — Support for MPC8313E-RDB Rev C • Linux kernel: — Support for Marvell 88E1111 single port GBE PHY connecting to eTSEC1 and eTSEC2. — Support for SGMII on eTSEC1 and eTSEC2 — IEEE 1588 validation drivers, working on both eTSEC1 and eTSEC2 — Support for IPSec OCF driver	
20080613	This BSP inherits the previous BSP to support the following new features • Final release for REV B board • Hardware platform: — Support for MPC8313E CPU 2.0 (PVR = 0x80850010, SVR = 0x80B00020) — Support for MPC8313E-RDB Rev B • Linux kernel: — Upgraded to 2.6.23 — Support for SD memory card • U-Boot — Upgraded to 1.3.0 • Toolchain — Upgraded to Gcc4.1.2, eglibc-2.5.59, binutils 2.17, CodeSourcery release 78, supporting e300c3/c4 cores with NPTL	
20070831	This BSP inherits the previous BSP to support the following new features • Formally support the REVA3/A4 board • Fix ds1339 RTC support in linux-2.6.20 • Clean up IEEE 1588 code and enhance the IEEE 1588 driver • Update the power management driver • Remove bug fix code for Marvell 88E1111 PHY • Add OCF + IPSEC stuff • Upgrade the toolchain version to gcc-4.0.2-e300c3, glibc-2.3.6, binutils-2.16, • Supporting NPTL thread libraries • Provide preemptible configuration option (disabled by default) • Two flash images: one for GE switch, the other for USB external PHY	

Table 2. MPC8313E-RDB BSP Changes (continued)

20070719	This BSP inherits the previous BSP to support the following new features • Formally support the REVA3 board • CodeWarrior debug and other debug features disabled by default • eTSEC performance improvement features included • Fix eTSEC TX-Flow bug (eTSEC errata 27) • One Flash image to support the internal USB PHY included	
20070428	This BSP inherits the previous BSP to support the following new features • Support 33MHz oscillator • Remove PCI 66MHz support. The oscillator does not support • Support NAND flash booting up • Supports REVA1/A2 boards • eTSEC1 VSC7385 firmware is loaded and enabled in Linux by default • Support SGMII mode in eTSEC2 • Support CORE/CSB/DDR=333/166/333, CORE/CSB/DDR=266/133/266, CORE/CSB/DDR=400/133/266 • Linux kernel • Linux kernel has been updated into 2.6.20 • Performance Monitor with oprofile as application • Support Power Management for TSEC and GTM sleep, deep sleep, and wake-up • EEE1588 PTP validation driver • Support USB in OTG with external PHY, Host in both external and internal PHY, • Gadget in both external and gadget PHY • Support PCI SATA demo	
20070306	• U-Boot 1.1.6 • Enable MPC8313E • Enhanced local bus support • I2C bus • DDR2 32bit memory initialization • DUART • eTSEC1 and eTSEC2 as Ethernet and TFTP port • Flash operation • Reset command • Bootup from NOR Flash • Enable cache and MMU in u-Boot • NAND flash read/write • Linux kernel 2.6.19-rc6 supporting E300c3 core	

THIS PAGE IS INTENTIONALLY BLANK

How to Reach Us:

Home Page:
www.freescale.com

Web Support:
http://www.freescale.com/support

USA/Europe or Locations Not Listed:
Freescale Semiconductor, Inc.
Technical Information Center, EL516
2100 East Elliot Road
Tempe, Arizona 85284
+1-800-521-6274 or +1-480-768-2130
www.freescale.com/support

Europe, Middle East, and Africa:
Freescale Halbleiter Deutschland GmbH
Technical Information Center
Schatzbogen 7
81829 Muenchen, Germany
+44 1296 380 456 (English)
+46 8 52200080 (English)
+49 89 92103 559 (German)
+33 1 69 35 48 48 (French)
www.freescale.com/support

Japan:
Freescale Semiconductor Japan Ltd.
Headquarters
ARCO Tower 15F
1-8-1, Shimo-Meguro, Meguro-ku,
Tokyo 153-0064
Japan
0120 191014 or +81 3 5437 9125
support.japan@freescale.com

Asia/Pacific:
Freescale Semiconductor China Ltd.
Exchange Building 23F
No. 118 Jianguo Road
Chaoyang District
Beijing 100022
China
+86 10 5879 8000
support.asia@freescale.com

For Literature Requests Only:
Freescale Semiconductor Literature Distribution Center
1-800-441-2447 or 303-675-2140
Fax: 303-675-2150
LDCForFreescaleSemiconductor@hibbertgroup.com

Document Number: AN3947
Rev. 0
11/2009

Information in this document is provided solely to enable system and software implementers to use Freescale Semiconductor products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits or integrated circuits based on the information in this document.

Freescale Semiconductor reserves the right to make changes without further notice to any products herein. Freescale Semiconductor makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does Freescale Semiconductor assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. “Typical” parameters that may be provided in Freescale Semiconductor data sheets and/or specifications can and do vary in different applications and actual performance may vary over time. All operating parameters, including “Typicals”, must be validated for each customer application by customer’s technical experts. Freescale Semiconductor does not convey any license under its patent rights nor the rights of others. Freescale Semiconductor products are not designed, intended, or authorized for use as components in systems intended for surgical implant into the body, or other applications intended to support or sustain life, or for any other application in which the failure of the Freescale Semiconductor product could create a situation where personal injury or death may occur. Should Buyer purchase or use Freescale Semiconductor products for any such unintended or unauthorized application, Buyer shall indemnify and hold Freescale Semiconductor and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claim alleges that Freescale Semiconductor was negligent regarding the design or manufacture of the part.

RoHS-compliant and/or Pb-free versions of Freescale products have the functionality and electrical characteristics as their non-RoHS-compliant and/or non-Pb-free counterparts. For further information, see http://www.freescale.com or contact your Freescale sales representative.

For information on Freescale’s Environmental Products program, go to http://www.freescale.com/epp.

freescale™
semiconductor