

# ***Building Linux for the OMAP5912 OSK™ Development Platform***

---

*Rishi Bhattacharya, Steve Kipisz*

## **ABSTRACT**

This report describes the process of building and running Linux on the OMAP Software Development Platform. Tool chain and root file system installation are also included for kernel testing purposes.

## **Contents**

<b>1</b>	<b>Introduction .....</b>	<b>1</b>
<b>2</b>	<b>Requirements .....</b>	<b>1</b>
	2.1 Hardware Requirements .....	1
	2.2 Software Requirements .....	2
<b>3</b>	<b>OMAPxxxx Software Development Platform Boot loader .....</b>	<b>2</b>
<b>4</b>	<b>Getting the Tool Chain, Kernel Source, and Root File System .....</b>	<b>2</b>
<b>5</b>	<b>Installing the Tool Chain.....</b>	<b>2</b>
<b>6</b>	<b>Installing the Kernel Source.....</b>	<b>3</b>
<b>7</b>	<b>Installing the Root File System.....</b>	<b>3</b>
<b>8</b>	<b>Build the Linux Kernel Image.....</b>	<b>3</b>
<b>9</b>	<b>Format the Kernel Image for U-Boot Download .....</b>	<b>4</b>
<b>10</b>	<b>Download and Boot the Kernel .....</b>	<b>4</b>
	<b>References.....</b>	<b>6</b>

## **1 Introduction**

Linux for the OMAP Software Development Platform can be built either by obtaining an ARM tool chain and kernel source from Monta Vista, or by installing an Open Source tool chain and building an Open Source Linux kernel. This report describes the Open Source process.

## **2 Requirements**

### **2.1 Hardware Requirements**

- OMAP5912 OSK (OMAP Starter Kit)

Overwrite this text with the Lit. Number

---

- Serial null-modem cable or serial cable with a null-modem adapter
- PC running RedHat™ Linux

RedHat™ Linux 9.0 was installed on the PC used to build the kernel for this report.

## 2.2 Software Requirements

- Open Source arm linux tool chain
- Open Source Linux kernel source with OMAP support
- ARM Linux root file system

The OMAP5912 OSK was used to test the kernel built.

## 3 OMAP5912 Software Development Platform Boot loader

This report assumes the u-boot boot loader is already installed. Building and installing u-boot is out of the scope of this report and is described in “*U-Boot for the OMAP16xx GSM/GPRS Software Development Platform*” [1].

## 4 Getting the Tool Chain, Kernel Source, and Root File System

Open up your web browser and go to <http://linux.omap.com>. Select the OSK Board and download the tool chain, kernel source, and root file system for the 2.6 Sample Build. The file names are:

Tool Chain	gcc-3.4.0.tar.bz2
Kernel Source	2.6.8-rc3_080804.tar.bz2
Root File System	rootfsosk.tar.bz2

## 5 Installing the Tool Chain

This report follows the same directory conventions as “*Building Linux for the Innovator Development Kit for OMAP™ Platform*” [2]. The source files are downloaded to the /root directory while the tool chain will be installed into the /opt directory.

Change directory to the /opt directory and un-tar the tool chain file.

```
[root@localhost]# cd /opt
[root@localhost]# tar -xjvf /root/gcc-3.4.0.tar.bz2
```

The tool chain will be installed in the /opt/gcc-3.4.0 directory.

## 6 Installing the Kernel Source

Change directory to the /usr/src directory and un-tar the kernel source file.

```
[root@localhost]# cd /usr/src
[root@localhost]# tar -xjvf /root/2.6.8-rc3_080804.tar.bz2
```

The kernel source tree will be installed in the /usr/src/2.6.8-rc3 directory.

## 7 Installing the Root File System

The root file system is hosted on the PC and will be NFS mounted by the kernel at boot up. Create a /data directory, change directory to it, and then un-tar the root file system file.

```
[root@localhost]# mkdir /data
[root@localhost]# cd /data
[root@localhost]# tar -xjvf /root/rootfsosk.tar.bz2
```

The root file system will be installed in the /data/rootfs2.6 directory.

Setting up the PC as an NFS server is out of the scope of this document. See the Linux man pages or any book on installing and configuring Red Hat Linux.

## 8 Build the Linux Kernel Image

Make sure the path to the tool chain is in the current path. The kernel is built in two steps. The first configures the kernel to build for the OSK while the second builds the kernel.

```
[root@localhost]# cd /usr/src/2.6.8-rc3
[root@localhost]# export PATH=/opt/gcc-3.4.0/bin:$PATH
[root@localhost]# make omap_osk_5912_defconfig
[root@localhost]# make
```

The **export** command adds the tool chain path to the current path. The next command configures to build the OSK Sample kernel, while the final **make** makes the kernel.

*Overwrite this text with the Lit. Number*

---

## 9 Format the Kernel Image for U-Boot Download

The file `arch/arm/boot/compressed/vmlinux` is the kernel image, but it is not in a suitable format for U-Boot to download. To convert the file, the `mkimage` tool that comes with U-Boot must be used. See [1] for more information on building and installing U-Boot.

```
[root@localhost]# cd /usr/src/2.6.8-rc3
[root@localhost]# arm-linux-objcopy -O binary -R .note -R .comment -S
arch/arm/boot/compressed/vmlinux linux.bin
[root@localhost]# gzip -9 linux.bin
[root@localhost]# mkimage -A arm -O linux -T kernel -C gzip -a 0x10c08000 -e 0x10c08000
-n "Linux Kernel Image" -d linux.bin.gz uImage.cc
```

## 10 Download and Boot the Kernel

Downloading and booting a kernel is described in [1]. Refer to that document on setting up the SDP and PC for downloading and booting this kernel. After booting the kernel, the terminal emulator should show something like below

U-Boot 1.1.1 (Aug 2 2004 - 18:08:10)

U-Boot code: 11080000 -> 11095D7C BSS: -> 1109A54C

RAM Configuration:

Bank #0: 10000000 32 MB

Micron StrataFlash MT28F128J3 device initialized

Flash: 32 MB

In: serial

Out: serial

Err: serial

Hit any key to stop autoboot: 0

OMAP5912 OSK # bootm 0x100000

## Booting image at 00100000 ...

Image Name: Linux-2.6.8-rc3-omap1

Image Type: ARM Linux Kernel Image (uncompressed)

Data Size: 909608 Bytes = 888.3 kB

Load Address: 10008000

Entry Point: 10008000

Verifying Checksum ... OK

OK

Starting kernel ...

Uncompressing Linux.....

.. done, booting the kernel.

Linux version 2.6.8-rc3-omap1 (root@dual-desktop) (gcc version 3.4.0 20031230 (CodeSourcery ARM Q4 2003)) #3 Sun Aug 8 12:10:31 CDT 2004

CPU: ARM926EJ-Sid(wt) [41069263] revision 3 (ARMv5TEJ)

CPU: D VIPT write-back cache

CPU: I cache: 16384 bytes, associativity 4, 32 byte lines, 128 sets

CPU: D cache: 8192 bytes, associativity 4, 32 byte lines, 64 set

Machine: TI-OSK

Memory policy: ECC disabled, Data cache writethrough

OMAP revision: 3.2 (0x03320200) id: 0x2b58c02f detected as OMAP-5912/1611B

Built 1 zonelists

Kernel command line: console=ttyS0,115200n8 noinitrd rw ip=dhcp root=/dev/nfs nf sroot=192.168.2.37:/home/rishi/rootfs2.6,nolock mem=30M

Total of 128 interrupts in 4 interrupt banks

PID hash table entries: 128 (order 7: 1024 bytes)

Console: colour dummy device 80x30

Dentry cache hash table entries: 4096 (order: 2, 16384 bytes)

Inode-cache hash table entries: 2048 (order: 1, 8192 bytes)

Memory: 30MB = 30MB total

Memory: 28348KB available (1517K code, 424K data, 80K init)

Calibrating delay loop... 95.84 BogoMIPS

Mount-cache hash table entries: 512 (order: 0, 4096 bytes)

CPU: Testing write buffer coherency: ok

NET: Registered protocol family 16

OMAP DMA hardware version 1

DMA capabilities: 000c0000:00000000:01ff:003f:007f

OMAP GPIO hardware version 1.0

Initializing OMAP McBSP system

OMAP OCPI interconnect driver loaded

Power Management fo

NetWinder Floating Point Emulator V0.97 (double precision)

devfs: 2004-01-31 Richard Gooch (rgooch@atnf.csiro.au)

```
devfs: boot_options: 0x0
Initializing Cryptographic API
Serial: 8250/16550 driver $Revision: 1.90 $ 4 ports, IRQ sharing disabled
ttyS0 at MMIO 0xfefb0000 (irq = 46) is a OMAP UART
ttyS1 at MMIO 0xfefb0800 (irq = 47) is a OMAP UART
ttyS2 at MMIO 0xfefb9800 (irq = 15) is a 8250
OMAP serial: activated 3 ports
RAMDISK driver initialized: 16 RAM disks of 8192K size 1024 blocksize
loop: loaded (max 8 devi
PPP generic driver version 2.4.2
smc91x: IOADDR c2000000 doesn't match configuration (300).
smc91x.c: v1.0, mar 07 2003 by Nicolas Pitre <nico@cam.org>
eth0: SMC91C94 (rev 9) at 0xc2000000 IRQ 160 [nowait]
eth0: Ethernet addr: 00:0e:99:02:04:08
OMAP I2C: Driver ver. 1.3
tps65010: version Aug  8 200412:09:00
mice: PS/2 mouse device common for all mice
OMAP Keypad Driver
NET: Registered protocol family 2
IP: routing cache hash table of 512 buckets, 4Kbytes
TCP: Hash tables configured (established 2048 bind 4096)
NET: Registered protocol family 1
eth0: link up
Sending DHCP requests ., OK
IP-Config: Got DHCP answer from 0.0.0.0, my address is 192.168.2.85
IP-Config: Complete:
    device=eth0, addr=192.168.2.85, mask=255.255.255.0, gw=192.168.2.1,
    host=192.168.2.85, domain=WORKGROUP, nis-domain=(none),
    bootserver=0.0.0.0, rootserver=192.168.2.37, rootpath=
Looking up port of RPC 100003/2 on 192.168.2.37
Looking up port of RPC 100005/1 on 192.168.2.37
VFS: Mounted root (nfs filesystem).
Freeing init memory: 80K
init started: BusyBox v1.00-pre8 (2004.03.05-22:18+0000) multi-call binary
```

```
*****
```

```
Starting System Init for OMAP5912OSK
```

```
*****
```

```
Please press Enter to activate this console.
```

```
BusyBox v1.00-pre8 (2004.03.05-22:18+0000) Built-in shell (ash)
```

```
Enter 'help' for a list of built-in commands.
```

```
-sh: can't access tty; job control turned off
```

```
#
```