Configuring OMAP 2430SDP almost from scratch

Nishanth Menon
23 July 2007
Rev 0.3

Disclaimers: All copyrights to rightful owners… and I don't own any ☺. Any damage due to any faults in the document is not my responsibility.

History:
0.3 23 July 2007 – Bit of updates on the switch settings side for UART vs USB modes.
0.2 04 May 2007 – Formatting, sections 4, 5.1, 6.1, 6.2, 7.1.2
0.1 12 Jan 2007 - Created

Table of Contents

---

1. Introduction

Listening to the discussions in the list, I started on getting to use readily available tools to do some development on OMAP- get a kernel and filesystem up.. esp since I did have some spare time while not watching TV. The objective I set was to go from a base board to a booting and useable system in as short a time as possible. Still, it did take me around 6 hrs over a couple of days (late nights rather) with my lack of experience and the research I had to do (udev esp was a "hate first - love later" after this ☺).. But, I guess a 1 to 2 hr window should be sufficient for all of this to get thru… I guess most of the users of the linux-omap list would be experts about things I cover in this document. Hence, I have targeted this document at naïve users like me. This is not an official document, but does strive to be useful to some one, some time. As a result, this document is organized in execution order.

So grab a coffee/beer and let's get our hands dirty.

2. Objectives

To get an OMAP2430 board from bare board to shell # prompt.

3. Target platform

2430 SDP revision 4.0/5.0.

4. Tools used

Sl. No.	Desc	Usage	Source
1.	Code sourcery gcc	Compiler	http://www.codesourcery.com/gnu_toolchains/arm/download.html EABI, for IA32 GNU/Linux
2.	Das U-Boot	Boot loader	http://linux.omap.com/pub/bootloader/2430sdp/source/ u-boot-SEP1106.tar.gz For known reason, the U-Boot (http://sourceforge.net/projects/u-boot) tree still does not have a functional 2430 port in place….
3.	CSST	Flashing Tool	http://focus.ti.com/general/docs/wtbu/wtbugencontent.tsp?templateId=6123&navigationId=12013&contentId=14645&DCMP=WTBU&HQS=Other+OT+omap2430_devplatform As of date, the latest revision is 1.11
4.	Linux Kernel	OS	http://source.mvista.com/git/gitweb.cgi?p=linux-omap-2.6.git;a=log http://www.muru.com/linux/omap/README_OMAP_GIT has instructions on download
5.	Busybox	Filesystem	http://busybox.net/downloads/busybox-1.3.1.tar.bz2 I chose a stable one instead of a new svn copy
6.	Hotplug/udev	Dynamic device management	http://www.us.kernel.org/pub/linux/utils/kernel/hotplug/ hotplug-2004_09_23.tar.bz2 udev-104.tar.bz2

5. Host environment setup

I have a two machine setup: a linux PC for compilation and a windows PC from which I connect to the target and flash it.
Linux: Fedora Core 5 updated with 'yum update'.
Windows: Windows XP.
Network infrastructure: switches, and a dhcp server.

5.1. Linux Host

On the fedora core 5 box which I used to setup the compilation platform, I have installed nfs and tftp-server for use on my target, git and cogito (using yum) for pulling in the kernel and of course a user login.

1. This is the simplest. The compiler comes as a tar file, which I chose to uncompress to '/opt/arm'.
2. edit ~/.bash_profile to set the following variables (mymake is an alias that kind of saves a bit of fingers):

```
PATH=$PATH:$HOME/bin:/opt/arm/arm-2006q3/bin
# to ease up our compilation steps..
alias mymake='make  CROSS_COMPILE=arm-none-linux-gnueabi- V=1 ARCH=arm'
# These are for creating and sending patches.
export GIT_AUTHOR_NAME="Nishanth Menon"
export GIT_AUTHOR_EMAIL="menon.nishanth@gmail.com"
export GIT_COMMITTER_NAME="$GIT_AUTHOR_NAME"
export GIT_COMMITTER_EMAIL="$GIT_AUTHOR_EMAIL"
```

I am hoping here that you do have a ~/bin directory where you'd put your own binaries..

5.2. Windows Host

You need hyperterminal or terraterm to connect to the serial port. You are equally well off with minicom if you have a linux host. But, to flash the uboot, however, you need a windows host for CSST. Install CSST on the windows machine – this is a self installer and installs a quick start guide with requisite information.

6. Bootloader

6.1. Compiling the bootloader

1. Get the required code:
   wget http://linux.omap.com/pub/bootloader/2430sdp/source/u-boot-SEP1106.tar.gz

Note: if you have a proxy, remember to setup the http_proxy variable as required for wget..

2. *mymake omap2430sdp_config*
3. *mymake*

You get a tons of "`warning: target CPU does not support interworking`". Things work despite this ☺

4. If things went right, you will get a 'u-boot.bin' in the root directory. Copy this u-boot.bin to the Windows host.
5. **IMPORTANT**: Copy the u-boot/tools/mkimage to ~/bin (if your $PATH points there) – this is required for "make uImage"

6.2. Flashing the boot loader

This is the step where you'd need either a Lauterbach (which I don't have access to) or CSST. Download csst from the TI website.

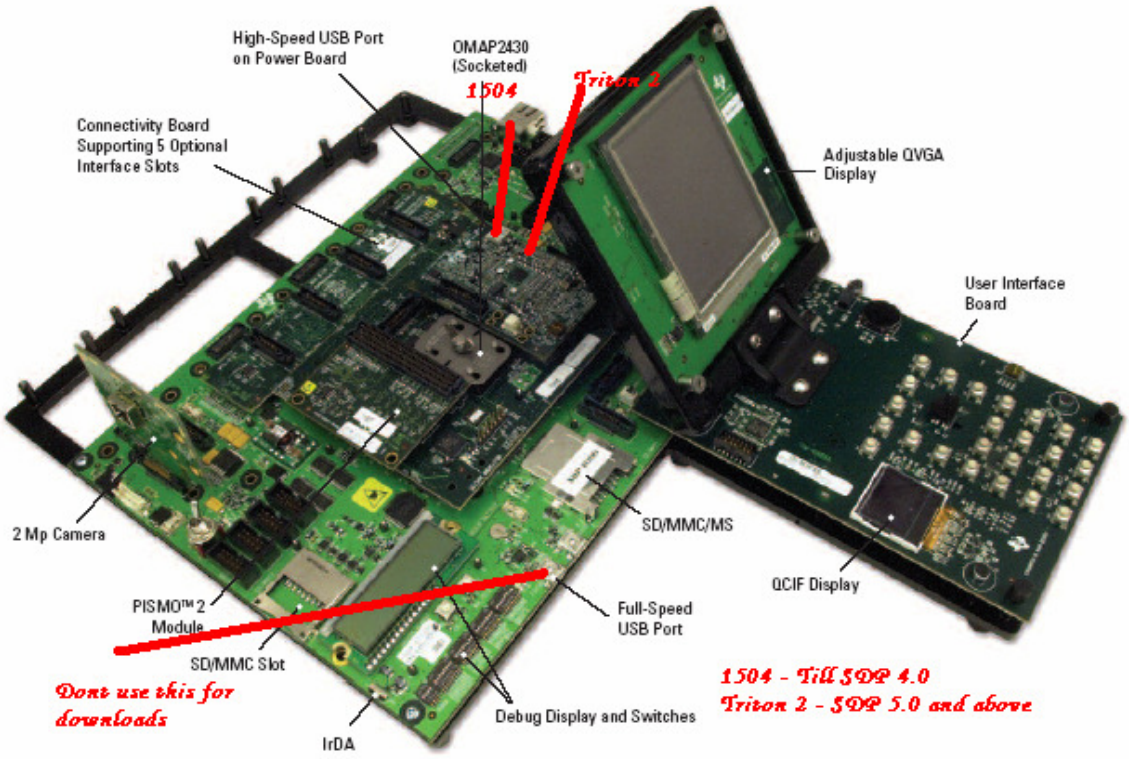
1. Setup your board with the dip switches given in "C:\Program Files\Texas Instruments\CSST_SDP2430_v1.11\docs\CSST_QuickStartGuide_SDP2430.pdf" table 1, page 8
   SPECIAL NOTE: on **SDP5.0, download works only over HSUSB** – this needs page 11 based dip switch

Switches	1	2	3	4	5	6	7	8
S4	OFF	OFF	OFF	OFF	ON	ON	OFF	OFF
S6	**ON**	**OFF**	OFF	OFF	OFF	OFF	OFF	ON
S8	ON	ON	ON	ON	ON	OFF	ON	OFF
S9	ON	OFF	ON	OFF	ON	ON	ON	OFF
S10	ON	ON	ON	ON	-	-	-	-
SW2 - Processor board (5.0)	OFF	OFF	ON	ON	-	-	-	-

UART DOWNLOAD Switches (From CSST documentation) – (Upto 4.0 SDP. 5.0 SDP and above does not support this)

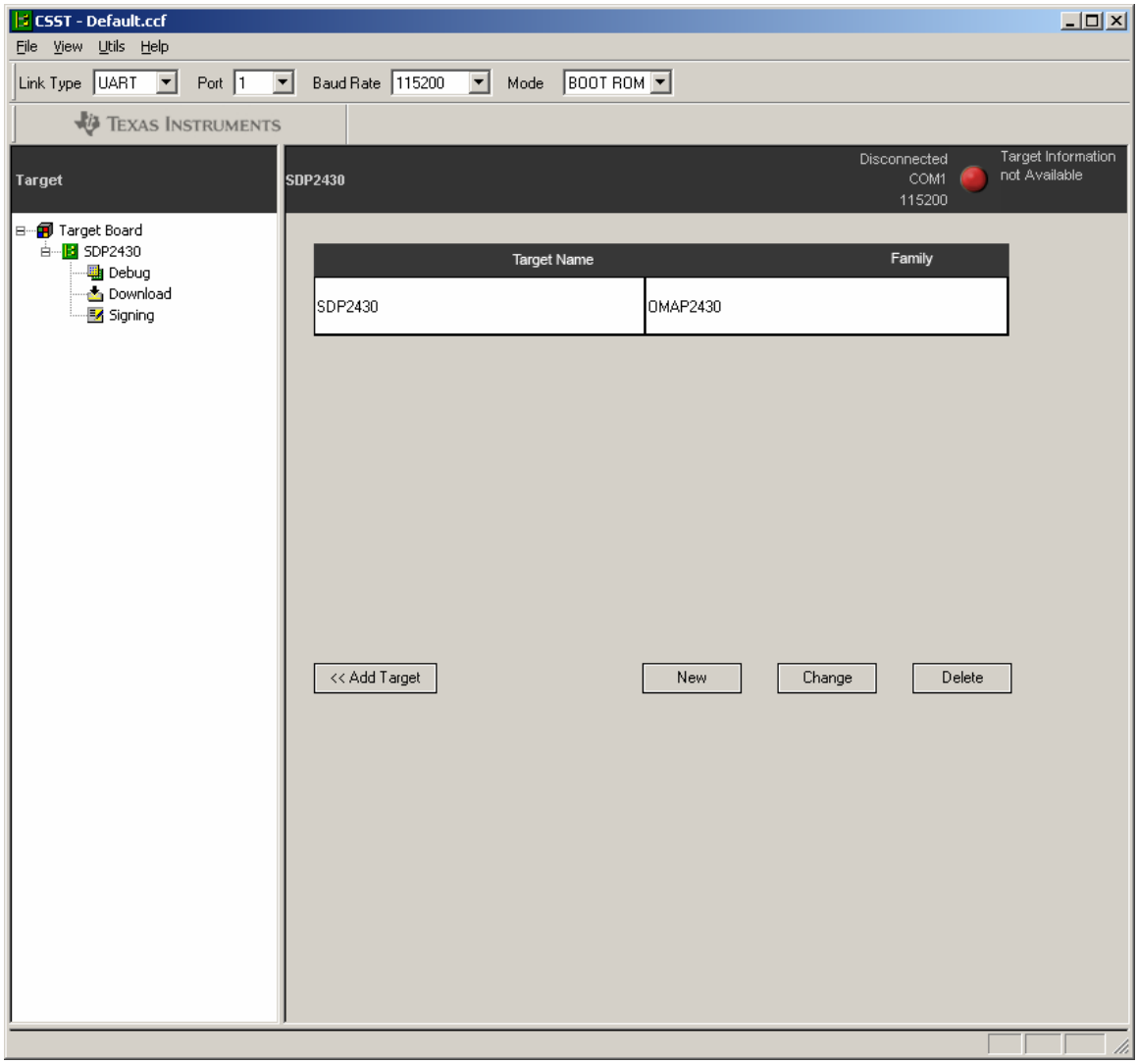
Switches	1	2	3	4	5	6	7	8
S4	OFF	OFF	OFF	OFF	ON	ON	OFF	OFF
S6	ON	ON	OFF	OFF	OFF	OFF	OFF	ON
S8	ON	ON	ON	ON	ON	OFF	ON	OFF
S9	ON	OFF	ON	OFF	OFF	ON	ON	OFF
S10	ON	ON	ON	ON	-	-	-	-
SW2 - Processor board (5.0)	OFF	OFF	ON	ON	-	-	-	-

USB Download switches (From CSST Documentation) –(Most of SDPs support this – including 5.0 SDP)
Port connection for USB:

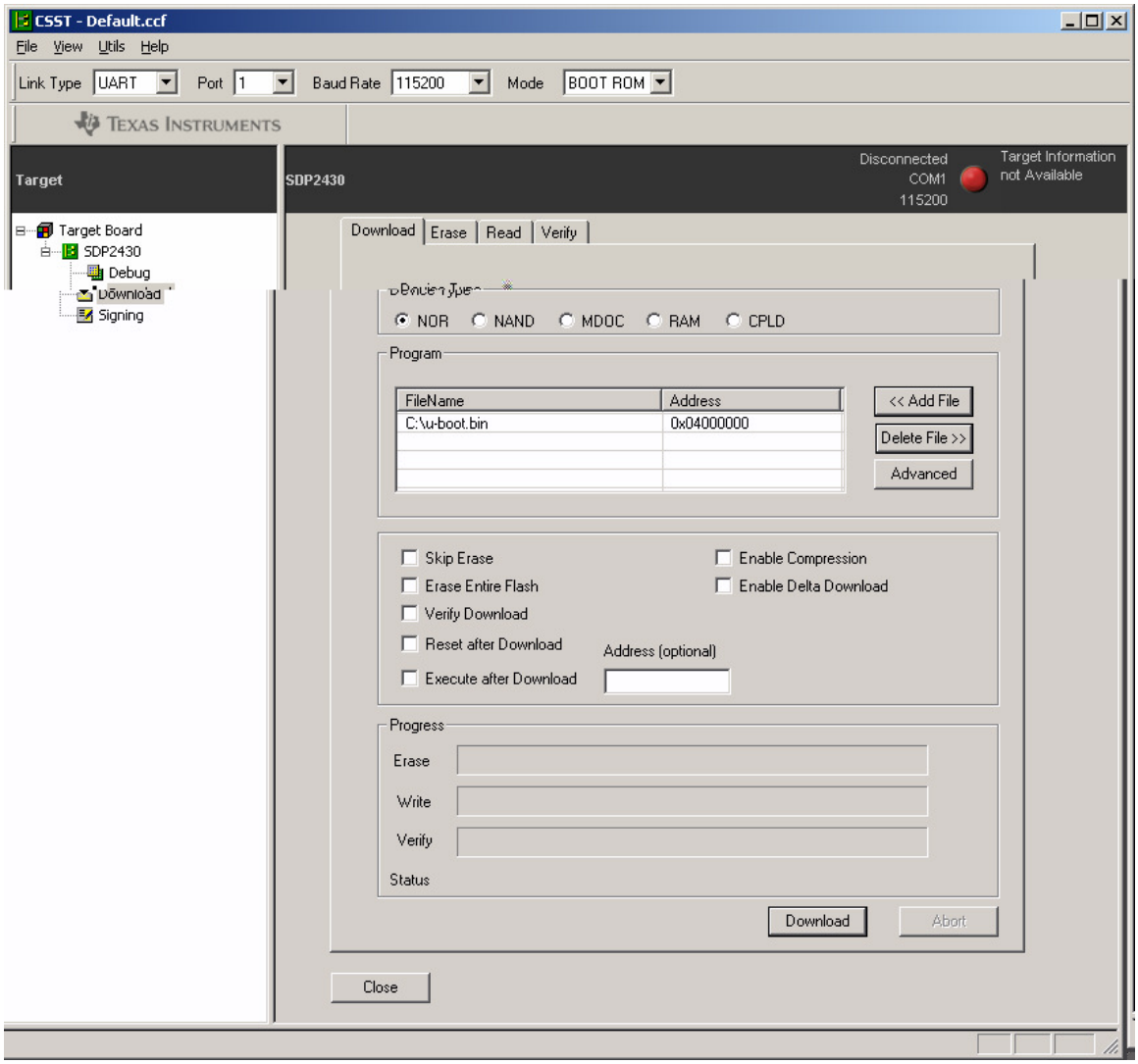


(Image thanks to TI website – Modified with Red labels by me)
UART3 is the uart port closer to the LCD screen.

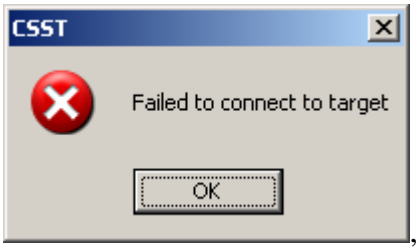
2. Connect your serial cable from windows host to the uart port3 (the one close to the LCD).
3. Start CSST tool



4. Click on “download” from the tree on left.
5. Click on NOR, Add file and select the uboot.bin. and set the address as 0x04000000. The CSST window should look as follows:



6. Now click on the “download” button. It shows “reset the board”. Reset the board and the download should start off



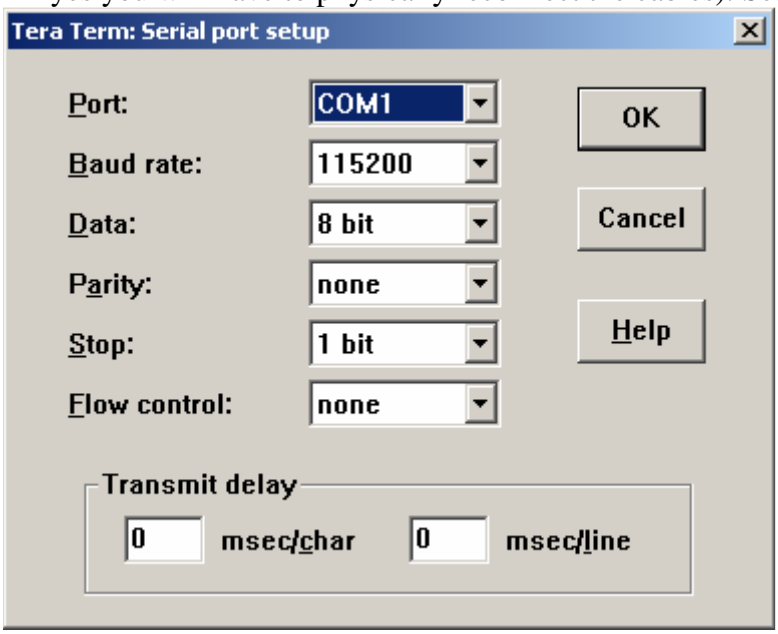
7. If things don’t work out.. you’d see an error , in which case, you’d have to go back to step number 1(check the dip switches and uart connectivity oh and make sure you don’t have terraterm or hyperterminal opened and connected to your com port…)..
8. Things working fine, you will see download progressing and completing successfully.
9. Close csst. You would most probably never need it again normally – u can always use uboot to flash itself..

6.3. Setting up and testing the boot loader

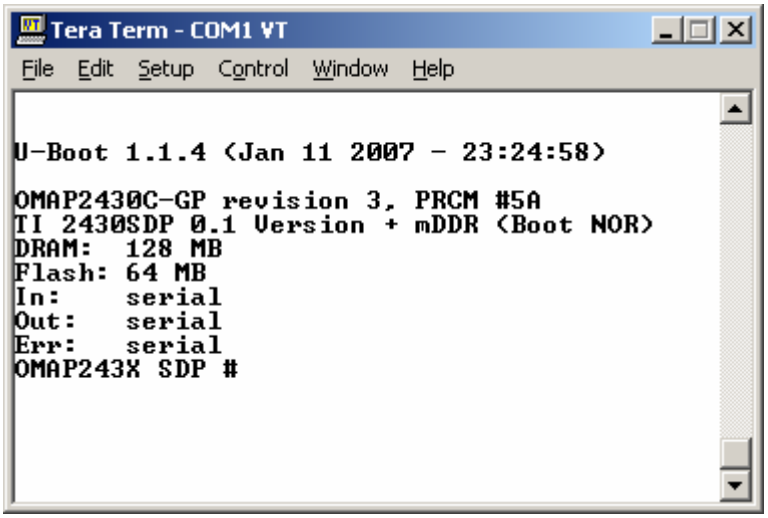
1. Setup your dip switches (sigh again..)..

S10							
1	2	3	4				
ON	ON	ON	ON				
S9							
1	2	3	4	5	6	7	8
ON	ON	ON	ON	ON	ON	ON	OFF
S8							
1	2	3	4	5	6	7	8
ON	ON	ON	ON	ON	ON	ON	OFF
S6							
1	2	3	4	5	6	7	8
ON	ON	OFF	OFF	OFF	OFF	OFF	ON
S4							
1	2	3	4	5	6	7	8
OFF	OFF	OFF	OFF	ON	ON	OFF	OFF

2. Connect your hyper terminal or terraterm to uart1 (the uart port away from LCD – yes you will have to physically reconnect the cables). Set it up to 115200,n,8:



3. Power on the board now.. if you have done good so far you will be rewarded with the U-Boot prompt.



Congratulations… you are now ready to fight the kernel..

7. Linux OS boot up

We will do this in two stages:

1. First the kernel alone
2. Second get the filesystem up and running with udev

Necessities on the linux host: tftp server and a nfs host. For this discussion, we shall assume, the nfs is mounted from /home/fs and tftp server gets files from /tftpboot. Setting up nfs *(/etc/exports)* and tftp-server *(/etc/xinetd.d/tftp)* on the linux host is beyond the scope of this document.

7.1. The Linux Kernel

7.1.1. Download the kernel

1. Get the latest git tree using: *cg-clone http://source.mvista.com/gittrees/linux-omap-2.6.git linux-omap*
    a. Remember to have your http_proxy as git uses wget to download the files… so check this first..

7.1.2. configure the kernel

1. Go to the kernel root directory (linux-omap)
2. *mymake omap_2430sdp_defconfig*
3. *mymake menuconfig*, go to "Kernel Features" and ensure that the following is enabled:
    a. Use the ARM EABI to compile the kernel
    b. Allow old ABI binaries to run with this kernel
    Both these should have a (*) against them – to toggle this, press space
4. Use tab to select exit and come out of the menu -> "`Do you wish to save your new kernel configuration?`" select "yes" and exit

5. *mymake uImage*
6. If things went right, you will get the file arch/arm/boot/uImage. (If it could not create the uImage, then go back to section 6.1point 5.

7.1.3. Verify the kernel

1. From the kernel root directory, do: '*file vmlinux*'. Note the SYSV in the output

```
vmlinux: ELF 32-bit LSB executable, ARM, version 1 (SYSV), statically linked, not
stripped
```

2. Now, copy the uImage to the tftp folder. *cp arch/arm/boot/uImage /tftpboot/*
3. Start your board. In your hyperterminal window – you will see the uboot prompt.
4. Setup the variables serverip to point to your tftpserver ip. And bootfile to point to the path to get the uImage file from.
   a. *setenv serverip 192.168.1.2*
   b. *setenv bootfile "uImage"*

   Note: on some systems when you do not have a dhcp server, you may need to setup the board ip address, gateway and netmask also.
5. Once the settings are completed, give the command '*saveenv*' – this saves it into flash for your future bootups. '*printenv*' prints all the environment variables.☺
6. Give your download a shot.. do '*dhcp*' you should see the download happening:

```
OMAP243X SDP # dhcp
Using MAC Address 08:00:28:01:01:94
BOOTP broadcast 1
*** Unhandled DHCP Option in OFFER/ACK: 43
*** Unhandled DHCP Option in OFFER/ACK: 44
*** Unhandled DHCP Option in OFFER/ACK: 46
*** Unhandled DHCP Option in OFFER/ACK: 43
*** Unhandled DHCP Option in OFFER/ACK: 44
*** Unhandled DHCP Option in OFFER/ACK: 46
DHCP client bound to address 128.247.79.46
TFTP from server 128.247.79.195; our IP address is 128.247.79.46
Filename 'nmenon/uImage'.
Load address: 0x80000000
Loading: #################################################################
         #################################################################
         #################################################################
         #################################################################
         ###############
done
Bytes transferred = 1404308 (156d94 hex)
OMAP243X SDP #
```

7. Type '*bootm*' and press enter. If things work out, you will the kernel boot up..

```
## Booting image at 80000000 ...
   Image Name:   Linux-2.6.20-rc3-omap1-ge450521a
   Image Type:   ARM Linux Kernel Image (uncompressed)
   Data Size:    1404244 Bytes =  1.3 MB
   Load Address: 80008000
   Entry Point:  80008000
   Verifying Checksum ... OK
OK

Starting kernel ...

Uncompressing
Linux.....................................................................
...................... done, booting the kernel.
<5>Linux version 2.6.20-rc3-omap1-ge450521a (xxxxxx@xxxxxxx) (gcc version
4.1.1 (CodeSourcery ARM Sourcery G++ 2006q3-26)) #3 Thu Jan 11 17:08:23 CST 2007
CPU: ARMv6-compatible processor [4107b366] revision 6 (ARMv6TEJ), cr=00c5387f
Machine: OMAP2430 sdp2430 board
Memory policy: ECC disabled, Data cache writeback
<7>On node 0 totalpages: 8192
```

```
<7>  DMA zone: 64 pages used for memmap
<7>  DMA zone: 0 pages reserved
<7>  DMA zone: 8128 pages, LIFO batch:0
<SNIP>
0x00000000-0x00040000 : "bootloader"
<5>0x00040000-0x00060000 : "params"
0x00040000-0x00060000 : "params"
<5>0x00060000-0x00260000 : "kernel"
0x00060000-0x00260000 : "kernel"
<5>0x00260000-0x04000000 : "filesystem"
0x00260000-0x04000000 : "filesystem"
<6>OMAP Keypad Driver
OMAP Keypad Driver
<6>TCP cubic registered
TCP cubic registered
<6>NET: Registered protocol family 1
NET: Registered protocol family 1
<6>NET: Registered protocol family 17
NET: Registered protocol family 17
<6>NET: Registered protocol family 15
NET: Registered protocol family 15
<5>RAMDISK: Couldn't find valid RAM disk image starting at 0.
RAMDISK: Couldn't find valid RAM disk image starting at 0.
<6>Time: 32k_counter clocksource has been installed.
Time: 32k_counter clocksource has been installed.
No filesystem could mount root, tried: No filesystem could mount root, tried:  ext3
ext3
 ext2 ext2 msdos msdos vfat vfat

<0>Kernel panic - not syncing: VFS: Unable to mount root fs on unknown-block(1,0)
Kernel panic - not syncing: VFS: Unable to mount root fs on unknown-block(1,0)
```

Don't worry about the kernel panic about unable to mount the root fs.. This is what we will fix next. ☺

7.2. The Busybox filesystem

I will not try and get some fancy stuff like X-windows or any thing else working, but I will try and get you through some of the basic stuff you'd have to face.. Remember the objective is to get a NFS filesystem up and running.. a ramdisk is also equally possible… but that is just a couple of steps away and pretty easy to handle(a bit of dd,mke2fs and tar cmds later u have that).

7.2.1. Download the filesystem components.

1. Get busybox *: wget http://busybox.net/downloads/busybox-1.3.1.tar.bz2*
2. Get hotplug: *wget http://www.us.kernel.org/pub/linux/utils/kernel/hotplug/hotplug-2004_09_23.tar.bz2*
3. Get udev : *wget http://www.us.kernel.org/pub/linux/utils/kernel/hotplug/udev-104.tar.bz2*
4. Untar all these to get busybox-1.3.1 udev-104 hotplug-2004_09_23 directories

7.2.2. Compile busybox

1. *cd busybox-1.3.1*

2. Edit the *makefile*: line 302 should look like this (I'd like just soft-float.. but what the hell):

```
CFLAGS          := -march=armv6 -mtune=arm1136j-s  -msoft-float -marm -fno-omit-frame-pointer -mapcs -mno-sched-prolog -mabi=aapcs-linux -mno-thumb-interwork
```

3. *mymake defconfig*
4. *mymake menuconfig* Go to "Shells" -> "choose your default shell" and select ash

You can go an edit all the stuff you like in the menu – add remove components you like and dislike.. and finally do an exit and save your configuration..

It might be easier for you to copy my .config for the same. Refer to Appendix A: busybox .config

5. *mymake*
6. *mymake install* (This will create the _install directory under the busybox root and install everything there).
7. check if you compiled busybox properly – run *'file ./_install/bin/busybox'*

```
./_install/bin/busybox: ELF 32-bit LSB executable, ARM, version 1 (SYSV), dynamically linked (uses shared libs), stripped
```

8. *setup the setuid of busybox: chmod 4755 ./_install/bin/busybox*

7.2.3. Setup busybox to work with shared libraries

1. Assuming you had setup busy box as shared library, do: *arm-none-linux-gnueabi-readelf -a _install/bin/busybox | grep "Shared library"*

```
0x00000001 (NEEDED)                     Shared library: [libcrypt.so.1]
 0x00000001 (NEEDED)                     Shared library: [libm.so.6]
 0x00000001 (NEEDED)                     Shared library: [libgcc_s.so.1]
 0x00000001 (NEEDED)                     Shared library: [libc.so.6]
```

2. Locate these files and copy the library archives to the _install/lib directory
   NOTE: A quick and dirty way of doing this is as follows:
   mkdir _install/lib
   cp -rvf /opt/arm/arm-2006q3/arm-none-linux-gnueabi/libc/lib/ _install/lib/*
   cp -rvf /opt/arm/arm-2006q3/arm-none-linux-gnueabi/lib/ _install/lib/*

7.2.4. install hotplug and udev

1. cd to the hotplug directory
2. *cp -rvf etc sbin ../busybox-1.3.1/_install/*
3. cd to the u-dev directory
4. *mymake DESTDIR=`pwd`/../busybox-1.3.1/_install install*
5. *cp udevinfo udevmonitor udevstart udevtest ../busybox-1.3.1/_install/sbin/*
6. *cp etc/udev/debian/* ../busybox-1.3.1/_install/etc/udev/rules.d/*
7. Edit *../busybox-1.3.1/_install/etc/udev/rules.d/devfs.rules*, line 54: it should be:

```
   KERNEL=="ttyS[0-9]*",           NAME="tts/%n" SYMLINK +="ttyS%n"
```

NOTE: these rules give you "devfs" compatible device nodes – I like the ones with the directory organization… for some reason… well… rest is upto you to hack up…

7.2.5. Configuring the system

We now need to create some directories in the system and some initial files.

1. cd to the busybox install directory.

Thanks to http://souptonuts.sourceforge.net/cdrom.htm here are some modified instructions

2. *mkdir -p dev sys proc lib tmp var/lib/misc var/lock var/log var/run var/tmp etc/rc.d/rc.local var/lock/subsys/hotplug dev/pts /root*
3. *ln -s rc.d/init.d/ init.d*
4. *sudo cp -avp /dev/console /dev/ttyS* dev*
   NOTE: you will need root access to create device nodes..
5. *sudo chown root.root bin/busybox*
6. *chmod 1777 tmp var/tmp*
7. Create the additional files: Refer Appendix B: Files in the /etc directory for contents:

etc/busybox.conf
etc/fstab
etc/group
etc/hosts
etc/inittab
etc/login.defs
etc/nsswitch.conf
etc/passwd
etc/securetty
etc/shadow
etc/udhcpd.conf
etc/rc.d/rcE
etc/rc.d/rcS
etc/rc.d/init.d/udev

8. *chmod 400 etc/shadow*
9. *chmod +x etc/rc.d/rcS etc/rc.d/rcE*
10. edit *etc/hotplug/usb.rc*, make line 248 as follows:

rmmod -s

11. *cd /etc/rc.d/rc.local* and run the following commands
    ln -s ../init.d/hotplug S01-hotplug
    ln -s ../init.d/hotplug E01-hotplug
    ln -s ../init.d/udev S01-udev
    ln -s ../init.d/udev E01-udev

7.2.6. Booting off the system

1. Move/copy the _install directory to that mounted off the NFS partition. In my case, it is /home/fs/arm
2. Go to the uboot prompt after resetting the board.
3. run the following command:
4. *setenv bootargs console=ttyS0,115200n8 noinitrd ip=dhcp root=/dev/nfs rw nfsroot=128.247.79.195:/home/fs/arm/,nolock mem=32M*
5. *saveenv*
6. *dhcp;bootm*

```
## Booting image at 80000000 ...
```

```
   Image Name:   Linux-2.6.20-rc3-omap1-ge450521a
   Image Type:   ARM Linux Kernel Image (uncompressed)
   Data Size:    1404244 Bytes =  1.3 MB
   Load Address: 80008000
   Entry Point:  80008000
   Verifying Checksum ... OK
OK

Starting kernel ...

Uncompressing
Linux.....................................................................
..................... done, booting the kernel.
<5>Linux version 2.6.20-rc3-omap1-ge450521a (xxxxxx@xxxxxxx) (gcc version
4.1.1 (CodeSourcery ARM Sourcery G++ 2006q3-26)) #3 Thu Jan 11 17:08:23 CST 2007
CPU: ARMv6-compatible processor [4107b366] revision 6 (ARMv6TEJ), cr=00c5387f
Machine: OMAP2430 sdp2430 board
Memory policy: ECC disabled, Data cache writeback
<7>On node 0 totalpages: 8192
<7>  DMA zone: 64 pages used for memmap
<7>  DMA zone: 0 pages reserved
<7>  DMA zone: 8128 pages, LIFO batch:0
<SNIP>
0x00000000-0x00040000 : "bootloader"
<5>0x00040000-0x00060000 : "params"
0x00040000-0x00060000 : "params"
<5>0x00060000-0x00260000 : "kernel"
0x00060000-0x00260000 : "kernel"
<5>0x00260000-0x04000000 : "filesystem"
0x00260000-0x04000000 : "filesystem"
<6>OMAP Keypad Driver
OMAP Keypad Driver
<6>TCP cubic registered
TCP cubic registered
<6>NET: Registered protocol family 1
NET: Registered protocol family 1
<6>NET: Registered protocol family 17
NET: Registered protocol family 17

<SNIP>
<6>NET: Registered protocol family 1
NET: Registered protocol family 1
<6>NET: Registered protocol family 17
NET: Registered protocol family 17
<6>NET: Registered protocol family 15
NET: Registered protocol family 15
<6>Time: 32k_counter clocksource has been installed.
Time: 32k_counter clocksource has been installed.
<6>eth0: link up
eth0: link up
<5>Sending DHCP requests .Sending DHCP requests .,, OK
 OK
IP-Config: Got DHCP answer from 255.255.255.255, IP-Config: Got DHCP answer from
255.255
.255.255, my address is 128.247.79.46
my address is 128.247.79.46
IP-Config: Complete:IP-Config: Complete:
      device=eth0
      device=eth0, addr=128.247.79.46, addr=128.247.79.46, mask=255.255.254.0,
mask=255.
255.254.0, gw=128.247.78.1, gw=128.247.78.1,
     host=128.247.79.46, domain=sc.ti.com, nis-domain=(none),
     host=128.247.79.46, domain=sc.ti.com, nis-domain=(none),
     bootserver=255.255.255.255,
     bootserver=255.255.255.255, rootserver=128.247.79.195, rootserver=128.247.79.195,
r
ootpath=, rootpath=

<5>Looking up port of RPC 100003/2 on 128.247.79.195
Looking up port of RPC 100003/2 on 128.247.79.195
```

```
<5>Looking up port of RPC 100005/1 on 128.247.79.195
Looking up port of RPC 100005/1 on 128.247.79.195
VFS: Mounted root (nfs filesystem).
VFS: Mounted root (nfs filesystem).
<6>Freeing init memory: 112K
Freeing init memory: 112K
starting up local systems
Mounting a tmpfs over /dev...done.
Creating initial device nodes...done.
done

128.247.79.46 login: root
login[529]: root login on 'tts/0'


BusyBox v1.3.1 (2007-01-12 00:29:58 CST) Built-in shell (ash)
Enter 'help' for a list of built-in commands.

~ $ ls
~ $ ls /dev
console     loop        mtd1ro      mtdblock0   ptmx        ttyS0       vc
full        mem         mtd2        mtdblock1   random      ttyS1       vcc
i2c         mtd0        mtd2ro      mtdblock2   rd          ttyS2       zero
kmem        mtd0ro      mtd3        mtdblock3   tts         ttyS3
kmsg        mtd1        mtd3ro      null        tty         urandom
~ $ echo hello world
hello world
```

8. Trouble shooting the system and some tips..

When things don't always go as planned.. so "**Don't Panic**". Solution is to go step by step.

Solutions… If you are not getting uboot prompt, the dip switches might be the problem, or even you might have forgotten to switch the UART to the correct one..

If it is just a filesystem issue, the kernel should have booted up right – but hangs, then ud' need to double check the steps done to reach till there.. in general, u can use the *init=/bin/sh* option in bootargs and get the shell prompt alone working… that would be one way to ensure that the busybox was compiled right.. If it sees some sort of issues such as "sh: applet not found".. well… check the busybox Makefile for the right options compilation options.. Errors like no init console, implies, the initial /dev/ entries that need to be created by hand was not done or your inittab is wrong.. Debugging udev is kind of a pain.. To give a brief idea: All we do is to have a very small set of devices to start off, then mount a tempfs on /dev and then allow udev to start creating the device nodes as per it's parsing of the udev rules… happy debugging that..
http://www.reactivated.net/writing_udev_rules.html can help more on that.
At times, you may see "nfs: server xyz not responding"… this could be due to two reasons: if the filesystem does come up, then your network is stuffy with packets.. try using a switch and isolate the network traffic.. if the filesystem refused to bootup, you'd want to see more into the etc files and busybox options (EABI kernel?)..

Kernel wise, it is a difficult to debug problem without some sort of ICD tools, but then.. printks can be useful at times… try to use the current kernel – they usually are better fixed. However, for fixes you may find, send the fixes you find to the linux opensource

list as kernel patches.. http://linux.omap.com/mailman/listinfo/linux-omap-open-source and we can all share the joy.. But remember there are around 1000 people on the list.. do a bit of google, read code before wasting 1 sec of all those folks reading/deleting the mail.. ☺

> Note on sending patches to the list: other than the etiquette of a mailing list, some recommendations I have heard:
> *cg-diff*
> *git-format patch 0123456789abcdee..0123456789abcdef*
> *git-am --signoff /tmp/your-new-patch*

Welcome to the world of Linux…. ☺

Appendix A: busybox .config

```
#
# Automatically generated make config: don't edit
# Linux kernel version: 1.3.1
# Thu Jan 11 19:02:05 2007
#
CONFIG_HAVE_DOT_CONFIG=y

#
# Busybox Settings
#

#
# General Configuration
#
CONFIG_NITPICK=y
CONFIG_DESKTOP=y
CONFIG_FEATURE_BUFFERS_USE_MALLOC=y
# CONFIG_FEATURE_BUFFERS_GO_ON_STACK is not set
# CONFIG_FEATURE_BUFFERS_GO_IN_BSS is not set
CONFIG_SHOW_USAGE=y
CONFIG_FEATURE_VERBOSE_USAGE=y
CONFIG_FEATURE_COMPRESS_USAGE=y
CONFIG_FEATURE_INSTALLER=y
CONFIG_LOCALE_SUPPORT=y
CONFIG_GETOPT_LONG=y
CONFIG_FEATURE_DEVPTS=y
# CONFIG_FEATURE_CLEAN_UP is not set
CONFIG_FEATURE_SUID=y
CONFIG_FEATURE_SYSLOG=y
CONFIG_FEATURE_SUID_CONFIG=y
CONFIG_FEATURE_SUID_CONFIG_QUIET=y
CONFIG_FEATURE_HAVE_RPC=y
# CONFIG_SELINUX is not set
CONFIG_BUSYBOX_EXEC_PATH="/proc/self/exe"

#
# Build Options
#
# CONFIG_STATIC is not set
# CONFIG_BUILD_LIBBUSYBOX is not set
# CONFIG_FEATURE_FULL_LIBBUSYBOX is not set
# CONFIG_FEATURE_SHARED_BUSYBOX is not set
# CONFIG_LFS is not set
# CONFIG_BUILD_AT_ONCE is not set

#
```

```
# Debugging Options
#
# CONFIG_DEBUG is not set
# CONFIG_DEBUG_PESSIMIZE is not set
# CONFIG_NO_DEBUG_LIB is not set
# CONFIG_DMALLOC is not set
# CONFIG_EFENCE is not set
CONFIG_DEBUG_YANK_SUSv2=y

#
# Installation Options
#
# CONFIG_INSTALL_NO_USR is not set
CONFIG_INSTALL_APPLET_SYMLINKS=y
# CONFIG_INSTALL_APPLET_HARDLINKS is not set
# CONFIG_INSTALL_APPLET_DONT is not set
CONFIG_PREFIX="./_install"

#
# Busybox Library Tuning
#
CONFIG_PASSWORD_MINLEN=6
CONFIG_MD5_SIZE_VS_SPEED=2

#
# Applets
#

#
# Archival Utilities
#
CONFIG_AR=y
CONFIG_FEATURE_AR_LONG_FILENAMES=y
CONFIG_BUNZIP2=y
# CONFIG_CPIO is not set
# CONFIG_DPKG is not set
# CONFIG_DPKG_DEB is not set
# CONFIG_FEATURE_DPKG_DEB_EXTRACT_ONLY is not set
CONFIG_GUNZIP=y
CONFIG_FEATURE_GUNZIP_UNCOMPRESS=y
CONFIG_GZIP=y
# CONFIG_RPM2CPIO is not set
# CONFIG_RPM is not set
CONFIG_TAR=y
CONFIG_FEATURE_TAR_CREATE=y
CONFIG_FEATURE_TAR_BZIP2=y
CONFIG_FEATURE_TAR_LZMA=y
CONFIG_FEATURE_TAR_FROM=y
CONFIG_FEATURE_TAR_GZIP=y
CONFIG_FEATURE_TAR_COMPRESS=y
CONFIG_FEATURE_TAR_OLDGNU_COMPATIBILITY=y
CONFIG_FEATURE_TAR_GNU_EXTENSIONS=y
CONFIG_FEATURE_TAR_LONG_OPTIONS=y
CONFIG_UNCOMPRESS=y
# CONFIG_UNLZMA is not set
# CONFIG_FEATURE_LZMA_FAST is not set
CONFIG_UNZIP=y

#
# Common options for cpio and tar
#
# CONFIG_FEATURE_UNARCHIVE_TAPE is not set
# CONFIG_FEATURE_DEB_TAR_GZ is not set
# CONFIG_FEATURE_DEB_TAR_BZ2 is not set
# CONFIG_FEATURE_DEB_TAR_LZMA is not set

#
# Coreutils
#
CONFIG_BASENAME=y
CONFIG_CAL=y
```

```
CONFIG_CAT=y
CONFIG_CATV=y
CONFIG_CHGRP=y
CONFIG_CHMOD=y
CONFIG_CHOWN=y
CONFIG_CHROOT=y
CONFIG_CKSUM=y
CONFIG_CMP=y
CONFIG_COMM=y
CONFIG_CP=y
CONFIG_CUT=y
CONFIG_DATE=y
CONFIG_FEATURE_DATE_ISOFMT=y
CONFIG_DD=y
CONFIG_FEATURE_DD_SIGNAL_HANDLING=y
CONFIG_FEATURE_DD_IBS_OBS=y
CONFIG_DF=y
CONFIG_DIFF=y
CONFIG_FEATURE_DIFF_BINARY=y
CONFIG_FEATURE_DIFF_DIR=y
CONFIG_FEATURE_DIFF_MINIMAL=y
CONFIG_DIRNAME=y
CONFIG_DOS2UNIX=y
CONFIG_UNIX2DOS=y
CONFIG_DU=y
CONFIG_FEATURE_DU_DEFAULT_BLOCKSIZE_1K=y
CONFIG_ECHO=y
CONFIG_FEATURE_FANCY_ECHO=y
CONFIG_ENV=y
CONFIG_FEATURE_ENV_LONG_OPTIONS=y
CONFIG_EXPR=y
CONFIG_EXPR_MATH_SUPPORT_64=y
CONFIG_FALSE=y
CONFIG_FOLD=y
CONFIG_HEAD=y
CONFIG_FEATURE_FANCY_HEAD=y
CONFIG_HOSTID=y
CONFIG_ID=y
CONFIG_INSTALL=y
CONFIG_FEATURE_INSTALL_LONG_OPTIONS=y
CONFIG_LENGTH=y
CONFIG_LN=y
CONFIG_LOGNAME=y
CONFIG_LS=y
CONFIG_FEATURE_LS_FILETYPES=y
CONFIG_FEATURE_LS_FOLLOWLINKS=y
CONFIG_FEATURE_LS_RECURSIVE=y
CONFIG_FEATURE_LS_SORTFILES=y
CONFIG_FEATURE_LS_TIMESTAMPS=y
CONFIG_FEATURE_LS_USERNAME=y
CONFIG_FEATURE_LS_COLOR=y
CONFIG_FEATURE_LS_COLOR_IS_DEFAULT=y
CONFIG_MD5SUM=y
CONFIG_MKDIR=y
CONFIG_FEATURE_MKDIR_LONG_OPTIONS=y
CONFIG_MKFIFO=y
CONFIG_MKNOD=y
CONFIG_MV=y
CONFIG_FEATURE_MV_LONG_OPTIONS=y
CONFIG_NICE=y
CONFIG_NOHUP=y
CONFIG_OD=y
CONFIG_PRINTENV=y
CONFIG_PRINTF=y
CONFIG_PWD=y
CONFIG_REALPATH=y
CONFIG_RM=y
CONFIG_RMDIR=y
CONFIG_SEQ=y
CONFIG_SHA1SUM=y
CONFIG_SLEEP=y
```

```
CONFIG_FEATURE_FANCY_SLEEP=y
CONFIG_SORT=y
CONFIG_FEATURE_SORT_BIG=y
CONFIG_STAT=y
CONFIG_FEATURE_STAT_FORMAT=y
CONFIG_STTY=y
CONFIG_SUM=y
CONFIG_SYNC=y
CONFIG_TAIL=y
CONFIG_FEATURE_FANCY_TAIL=y
CONFIG_TEE=y
CONFIG_FEATURE_TEE_USE_BLOCK_IO=y
CONFIG_TEST=y
CONFIG_FEATURE_TEST_64=y
CONFIG_TOUCH=y
CONFIG_TR=y
CONFIG_FEATURE_TR_CLASSES=y
CONFIG_FEATURE_TR_EQUIV=y
CONFIG_TRUE=y
CONFIG_TTY=y
CONFIG_UNAME=y
CONFIG_UNIQ=y
CONFIG_USLEEP=y
CONFIG_UUDECODE=y
CONFIG_UUENCODE=y
CONFIG_WATCH=y
CONFIG_WC=y
CONFIG_FEATURE_WC_LARGE=y
CONFIG_WHO=y
CONFIG_WHOAMI=y
CONFIG_YES=y

#
# Common options for cp and mv
#
CONFIG_FEATURE_PRESERVE_HARDLINKS=y

#
# Common options for ls, more and telnet
#
CONFIG_FEATURE_AUTOWIDTH=y

#
# Common options for df, du, ls
#
CONFIG_FEATURE_HUMAN_READABLE=y

#
# Common options for md5sum, sha1sum
#
CONFIG_FEATURE_MD5_SHA1_SUM_CHECK=y

#
# Console Utilities
#
CONFIG_CHVT=y
CONFIG_CLEAR=y
CONFIG_DEALLOCVT=y
CONFIG_DUMPKMAP=y
CONFIG_LOADFONT=y
CONFIG_LOADKMAP=y
CONFIG_OPENVT=y
CONFIG_RESET=y
CONFIG_RESIZE=y
CONFIG_FEATURE_RESIZE_PRINT=y
CONFIG_SETCONSOLE=y
# CONFIG_FEATURE_SETCONSOLE_LONG_OPTIONS is not set
CONFIG_SETKEYCODES=y
CONFIG_SETLOGCONS=y

#
```

```
# Debian Utilities
#
CONFIG_MKTEMP=y
CONFIG_PIPE_PROGRESS=y
CONFIG_READLINK=y
CONFIG_FEATURE_READLINK_FOLLOW=y
CONFIG_RUN_PARTS=y
CONFIG_FEATURE_RUN_PARTS_LONG_OPTIONS=y
CONFIG_START_STOP_DAEMON=y
CONFIG_FEATURE_START_STOP_DAEMON_FANCY=y
CONFIG_FEATURE_START_STOP_DAEMON_LONG_OPTIONS=y
CONFIG_WHICH=y

#
# Editors
#
CONFIG_AWK=y
CONFIG_FEATURE_AWK_MATH=y
CONFIG_ED=y
CONFIG_PATCH=y
CONFIG_SED=y
CONFIG_VI=y
CONFIG_FEATURE_VI_COLON=y
CONFIG_FEATURE_VI_YANKMARK=y
CONFIG_FEATURE_VI_SEARCH=y
CONFIG_FEATURE_VI_USE_SIGNALS=y
CONFIG_FEATURE_VI_DOT_CMD=y
CONFIG_FEATURE_VI_READONLY=y
CONFIG_FEATURE_VI_SETOPTS=y
CONFIG_FEATURE_VI_SET=y
CONFIG_FEATURE_VI_WIN_RESIZE=y
CONFIG_FEATURE_VI_OPTIMIZE_CURSOR=y

#
# Finding Utilities
#
CONFIG_FIND=y
CONFIG_FEATURE_FIND_PRINT0=y
CONFIG_FEATURE_FIND_MTIME=y
CONFIG_FEATURE_FIND_MMIN=y
CONFIG_FEATURE_FIND_PERM=y
CONFIG_FEATURE_FIND_TYPE=y
CONFIG_FEATURE_FIND_XDEV=y
CONFIG_FEATURE_FIND_NEWER=y
CONFIG_FEATURE_FIND_INUM=y
CONFIG_FEATURE_FIND_EXEC=y
CONFIG_GREP=y
CONFIG_FEATURE_GREP_EGREP_ALIAS=y
CONFIG_FEATURE_GREP_FGREP_ALIAS=y
CONFIG_FEATURE_GREP_CONTEXT=y
CONFIG_XARGS=y
CONFIG_FEATURE_XARGS_SUPPORT_CONFIRMATION=y
CONFIG_FEATURE_XARGS_SUPPORT_QUOTES=y
CONFIG_FEATURE_XARGS_SUPPORT_TERMOPT=y
CONFIG_FEATURE_XARGS_SUPPORT_ZERO_TERM=y

#
# Init Utilities
#
CONFIG_INIT=y
# CONFIG_DEBUG_INIT is not set
CONFIG_FEATURE_USE_INITTAB=y
CONFIG_FEATURE_INIT_SCTTY=y
CONFIG_FEATURE_EXTRA_QUIET=y
CONFIG_FEATURE_INIT_COREDUMPS=y
CONFIG_FEATURE_INITRD=y
CONFIG_HALT=y
CONFIG_MESG=y

#
# Login/Password Management Utilities
```

```
#
CONFIG_FEATURE_SHADOWPASSWDS=y
CONFIG_USE_BB_SHADOW=y
CONFIG_USE_BB_PWD_GRP=y
CONFIG_ADDGROUP=y
CONFIG_DELGROUP=y
CONFIG_ADDUSER=y
CONFIG_DELUSER=y
CONFIG_GETTY=y
CONFIG_FEATURE_UTMP=y
CONFIG_FEATURE_WTMP=y
CONFIG_LOGIN=y
CONFIG_LOGIN_SCRIPTS=y
CONFIG_FEATURE_SECURETTY=y
CONFIG_PASSWD=y
CONFIG_SU=y
CONFIG_SU_SYSLOG=y
CONFIG_SULOGIN=y
CONFIG_VLOCK=y

#
# Linux Ext2 FS Progs
#
CONFIG_CHATTR=y
# CONFIG_E2FSCK is not set
CONFIG_FSCK=y
CONFIG_LSATTR=y
# CONFIG_MKE2FS is not set
# CONFIG_TUNE2FS is not set
# CONFIG_E2LABEL is not set
# CONFIG_FINDFS is not set

#
# Linux Module Utilities
#
CONFIG_INSMOD=y
CONFIG_FEATURE_INSMOD_VERSION_CHECKING=y
CONFIG_FEATURE_INSMOD_KSYMOOPS_SYMBOLS=y
CONFIG_FEATURE_INSMOD_LOADINKMEM=y
CONFIG_FEATURE_INSMOD_LOAD_MAP=y
CONFIG_FEATURE_INSMOD_LOAD_MAP_FULL=y
CONFIG_RMMOD=y
CONFIG_LSMOD=y
CONFIG_FEATURE_LSMOD_PRETTY_2_6_OUTPUT=y
CONFIG_MODPROBE=y
CONFIG_FEATURE_MODPROBE_MULTIPLE_OPTIONS=y
CONFIG_FEATURE_MODPROBE_FANCY_ALIAS=y

#
# Options common to multiple modutils
#
CONFIG_FEATURE_CHECK_TAINTED_MODULE=y
CONFIG_FEATURE_2_4_MODULES=y
CONFIG_FEATURE_2_6_MODULES=y
# CONFIG_FEATURE_QUERY_MODULE_INTERFACE is not set

#
# Linux System Utilities
#
CONFIG_DMESG=y
CONFIG_FEATURE_DMESG_PRETTY=y
CONFIG_FBSET=y
CONFIG_FEATURE_FBSET_FANCY=y
CONFIG_FEATURE_FBSET_READMODE=y
CONFIG_FDFLUSH=y
CONFIG_FDFORMAT=y
CONFIG_FDISK=y
CONFIG_FDISK_SUPPORT_LARGE_DISKS=y
CONFIG_FEATURE_FDISK_WRITABLE=y
# CONFIG_FEATURE_AIX_LABEL is not set
# CONFIG_FEATURE_SGI_LABEL is not set
```

```
# CONFIG_FEATURE_SUN_LABEL is not set
# CONFIG_FEATURE_OSF_LABEL is not set
CONFIG_FEATURE_FDISK_ADVANCED=y
CONFIG_FREERAMDISK=y
CONFIG_FSCK_MINIX=y
CONFIG_MKFS_MINIX=y

#
# Minix filesystem support
#
CONFIG_FEATURE_MINIX2=y
CONFIG_GETOPT=y
CONFIG_HEXDUMP=y
CONFIG_HWCLOCK=y
CONFIG_FEATURE_HWCLOCK_LONG_OPTIONS=y
CONFIG_FEATURE_HWCLOCK_ADJTIME_FHS=y
CONFIG_IPCRM=y
CONFIG_IPCS=y
CONFIG_LOSETUP=y
CONFIG_MDEV=y
CONFIG_FEATURE_MDEV_CONF=y
CONFIG_FEATURE_MDEV_EXEC=y
CONFIG_MKSWAP=y
CONFIG_FEATURE_MKSWAP_V0=y
CONFIG_MORE=y
CONFIG_FEATURE_USE_TERMIOS=y
CONFIG_MOUNT=y
CONFIG_FEATURE_MOUNT_NFS=y
CONFIG_FEATURE_MOUNT_CIFS=y
CONFIG_FEATURE_MOUNT_FLAGS=y
CONFIG_FEATURE_MOUNT_FSTAB=y
CONFIG_PIVOT_ROOT=y
CONFIG_RDATE=y
CONFIG_READPROFILE=y
CONFIG_SETARCH=y
CONFIG_SWAPONOFF=y
CONFIG_SWITCH_ROOT=y
CONFIG_UMOUNT=y
CONFIG_FEATURE_UMOUNT_ALL=y

#
# Common options for mount/umount
#
CONFIG_FEATURE_MOUNT_LOOP=y
# CONFIG_FEATURE_MTAB_SUPPORT is not set

#
# Miscellaneous Utilities
#
CONFIG_ADJTIMEX=y
CONFIG_BBCONFIG=y
CONFIG_CROND=y
# CONFIG_DEBUG_CROND_OPTION is not set
CONFIG_FEATURE_CROND_CALL_SENDMAIL=y
CONFIG_CRONTAB=y
CONFIG_DC=y
# CONFIG_DEVFSD is not set
# CONFIG_DEVFSD_MODLOAD is not set
# CONFIG_DEVFSD_FG_NP is not set
# CONFIG_DEVFSD_VERBOSE is not set
# CONFIG_FEATURE_DEVFS is not set
CONFIG_EJECT=y
CONFIG_LAST=y
CONFIG_LESS=y
CONFIG_FEATURE_LESS_BRACKETS=y
CONFIG_FEATURE_LESS_FLAGS=y
CONFIG_FEATURE_LESS_FLAGCS=y
CONFIG_FEATURE_LESS_MARKS=y
CONFIG_FEATURE_LESS_REGEXP=y
CONFIG_HDPARM=y
CONFIG_FEATURE_HDPARM_GET_IDENTITY=y
```

```
CONFIG_FEATURE_HDPARM_HDIO_SCAN_HWIF=y
CONFIG_FEATURE_HDPARM_HDIO_UNREGISTER_HWIF=y
CONFIG_FEATURE_HDPARM_HDIO_DRIVE_RESET=y
CONFIG_FEATURE_HDPARM_HDIO_TRISTATE_HWIF=y
CONFIG_FEATURE_HDPARM_HDIO_GETSET_DMA=y
CONFIG_MAKEDEVS=y
# CONFIG_FEATURE_MAKEDEVS_LEAF is not set
CONFIG_FEATURE_MAKEDEVS_TABLE=y
CONFIG_MOUNTPOINT=y
CONFIG_MT=y
CONFIG_NMETER=y
CONFIG_RAIDAUTORUN=y
CONFIG_READAHEAD=y
CONFIG_RUNLEVEL=y
CONFIG_RX=y
CONFIG_STRINGS=y
CONFIG_SETSID=y
CONFIG_TASKSET=y
CONFIG_FEATURE_TASKSET_FANCY=y
CONFIG_TIME=y
CONFIG_WATCHDOG=y

#
# Networking Utilities
#
CONFIG_FEATURE_IPV6=y
CONFIG_ARPING=y
CONFIG_DNSD=y
CONFIG_ETHER_WAKE=y
CONFIG_FAKEIDENTD=y
CONFIG_FTPGET=y
CONFIG_FTPPUT=y
CONFIG_FEATURE_FTPGETPUT_LONG_OPTIONS=y
CONFIG_HOSTNAME=y
CONFIG_HTTPD=y
# CONFIG_FEATURE_HTTPD_RELOAD_CONFIG_SIGHUP is not set
# CONFIG_FEATURE_HTTPD_SETUID is not set
CONFIG_FEATURE_HTTPD_BASIC_AUTH=y
CONFIG_FEATURE_HTTPD_AUTH_MD5=y
CONFIG_FEATURE_HTTPD_CONFIG_WITH_MIME_TYPES=y
CONFIG_FEATURE_HTTPD_CGI=y
CONFIG_FEATURE_HTTPD_CONFIG_WITH_SCRIPT_INTERPR=y
CONFIG_FEATURE_HTTPD_SET_REMOTE_PORT_TO_ENV=y
CONFIG_FEATURE_HTTPD_ENCODE_URL_STR=y
CONFIG_IFCONFIG=y
CONFIG_FEATURE_IFCONFIG_STATUS=y
CONFIG_FEATURE_IFCONFIG_SLIP=y
CONFIG_FEATURE_IFCONFIG_MEMSTART_IOADDR_IRQ=y
CONFIG_FEATURE_IFCONFIG_HW=y
CONFIG_FEATURE_IFCONFIG_BROADCAST_PLUS=y
CONFIG_IFUPDOWN=y
CONFIG_FEATURE_IFUPDOWN_IP=y
CONFIG_FEATURE_IFUPDOWN_IP_BUILTIN=y
# CONFIG_FEATURE_IFUPDOWN_IFCONFIG_BUILTIN is not set
CONFIG_FEATURE_IFUPDOWN_IPV4=y
CONFIG_FEATURE_IFUPDOWN_IPV6=y
CONFIG_FEATURE_IFUPDOWN_IPX=y
CONFIG_FEATURE_IFUPDOWN_MAPPING=y
CONFIG_INETD=y
CONFIG_FEATURE_INETD_SUPPORT_BUILTIN_ECHO=y
CONFIG_FEATURE_INETD_SUPPORT_BUILTIN_DISCARD=y
CONFIG_FEATURE_INETD_SUPPORT_BUILTIN_TIME=y
CONFIG_FEATURE_INETD_SUPPORT_BUILTIN_DAYTIME=y
CONFIG_FEATURE_INETD_SUPPORT_BUILTIN_CHARGEN=y
CONFIG_FEATURE_INETD_RPC=y
CONFIG_IP=y
CONFIG_FEATURE_IP_ADDRESS=y
CONFIG_FEATURE_IP_LINK=y
CONFIG_FEATURE_IP_ROUTE=y
CONFIG_FEATURE_IP_TUNNEL=y
CONFIG_FEATURE_IP_RULE=y
```

```
CONFIG_FEATURE_IP_SHORT_FORMS=y
CONFIG_IPADDR=y
CONFIG_IPLINK=y
CONFIG_IPROUTE=y
CONFIG_IPTUNNEL=y
CONFIG_IPRULE=y
CONFIG_IPCALC=y
CONFIG_FEATURE_IPCALC_FANCY=y
CONFIG_FEATURE_IPCALC_LONG_OPTIONS=y
CONFIG_NAMEIF=y
CONFIG_NC=y
CONFIG_NC_SERVER=y
CONFIG_NC_EXTRA=y
CONFIG_NETSTAT=y
CONFIG_NSLOOKUP=y
CONFIG_PING=y
CONFIG_FEATURE_FANCY_PING=y
CONFIG_PING6=y
CONFIG_FEATURE_FANCY_PING6=y
CONFIG_ROUTE=y
CONFIG_TELNET=y
CONFIG_FEATURE_TELNET_TTYPE=y
CONFIG_FEATURE_TELNET_AUTOLOGIN=y
CONFIG_TELNETD=y
# CONFIG_FEATURE_TELNETD_STANDALONE is not set
CONFIG_TFTP=y
CONFIG_FEATURE_TFTP_GET=y
CONFIG_FEATURE_TFTP_PUT=y
CONFIG_FEATURE_TFTP_BLOCKSIZE=y
# CONFIG_DEBUG_TFTP is not set
CONFIG_TRACEROUTE=y
# CONFIG_FEATURE_TRACEROUTE_VERBOSE is not set
# CONFIG_FEATURE_TRACEROUTE_SOURCE_ROUTE is not set
# CONFIG_FEATURE_TRACEROUTE_USE_ICMP is not set
CONFIG_APP_UDHCPD=y
CONFIG_APP_DHCPRELAY=y
CONFIG_APP_DUMPLEASES=y
CONFIG_APP_UDHCPC=y
CONFIG_FEATURE_UDHCP_SYSLOG=y
# CONFIG_FEATURE_UDHCP_DEBUG is not set
CONFIG_VCONFIG=y
CONFIG_WGET=y
CONFIG_FEATURE_WGET_STATUSBAR=y
CONFIG_FEATURE_WGET_AUTHENTICATION=y
CONFIG_FEATURE_WGET_IP6_LITERAL=y
CONFIG_FEATURE_WGET_LONG_OPTIONS=y
CONFIG_ZCIP=y

#
# Process Utilities
#
CONFIG_FREE=y
CONFIG_FUSER=y
CONFIG_KILL=y
CONFIG_KILLALL=y
CONFIG_KILLALL5=y
CONFIG_PIDOF=y
CONFIG_FEATURE_PIDOF_SINGLE=y
CONFIG_FEATURE_PIDOF_OMIT=y
CONFIG_PS=y
CONFIG_FEATURE_PS_WIDE=y
CONFIG_RENICE=y
CONFIG_BB_SYSCTL=y
CONFIG_TOP=y
CONFIG_FEATURE_TOP_CPU_USAGE_PERCENTAGE=y
CONFIG_UPTIME=y

#
# Shells
#
CONFIG_FEATURE_SH_IS_ASH=y
```

```
# CONFIG_FEATURE_SH_IS_HUSH is not set
# CONFIG_FEATURE_SH_IS_LASH is not set
# CONFIG_FEATURE_SH_IS_MSH is not set
# CONFIG_FEATURE_SH_IS_NONE is not set
CONFIG_ASH=y

#
# Ash Shell Options
#
CONFIG_ASH_JOB_CONTROL=y
# CONFIG_ASH_READ_NCHARS is not set
# CONFIG_ASH_READ_TIMEOUT is not set
CONFIG_ASH_ALIAS=y
CONFIG_ASH_MATH_SUPPORT=y
# CONFIG_ASH_MATH_SUPPORT_64 is not set
CONFIG_ASH_GETOPTS=y
CONFIG_ASH_BUILTIN_ECHO=y
CONFIG_ASH_BUILTIN_TEST=y
# CONFIG_ASH_CMDCMD is not set
CONFIG_ASH_MAIL=y
CONFIG_ASH_OPTIMIZE_FOR_SIZE=y
# CONFIG_ASH_RANDOM_SUPPORT is not set
CONFIG_ASH_EXPAND_PRMT=y
# CONFIG_HUSH is not set
# CONFIG_LASH is not set
# CONFIG_MSH is not set

#
# Bourne Shell Options
#
# CONFIG_FEATURE_SH_EXTRA_QUIET is not set
# CONFIG_FEATURE_SH_STANDALONE_SHELL is not set
CONFIG_FEATURE_COMMAND_EDITING=y
# CONFIG_FEATURE_COMMAND_EDITING_VI is not set
CONFIG_FEATURE_COMMAND_HISTORY=15
CONFIG_FEATURE_COMMAND_SAVEHISTORY=y
CONFIG_FEATURE_COMMAND_TAB_COMPLETION=y
CONFIG_FEATURE_COMMAND_USERNAME_COMPLETION=y
CONFIG_FEATURE_SH_FANCY_PROMPT=y

#
# System Logging Utilities
#
CONFIG_SYSLOGD=y
CONFIG_FEATURE_ROTATE_LOGFILE=y
CONFIG_FEATURE_REMOTE_LOG=y
CONFIG_FEATURE_IPC_SYSLOG=y
CONFIG_FEATURE_IPC_SYSLOG_BUFFER_SIZE=16
CONFIG_LOGREAD=y
CONFIG_FEATURE_LOGREAD_REDUCED_LOCKING=y
CONFIG_KLOGD=y
CONFIG_LOGGER=y

#
# Runit Utilities
#
# CONFIG_RUNSV is not set
# CONFIG_RUNSVDIR is not set
# CONFIG_SV is not set
# CONFIG_SVLOGD is not set
CONFIG_CHPST=y
CONFIG_SETUIDGID=y
CONFIG_ENVUIDGID=y
CONFIG_ENVDIR=y
CONFIG_SOFTLIMIT=y
```

Appendix B: Files in the /etc directory

[etc/busybox.conf]

```
[SUID]
su = ssx root.0 # applet su can be run by anyone and runs with euid=0/egid=0
su = ssx        # exactly the same
```

[etc/fstab]

```
# /etc/fstab: static file system information.
#
# <file system> <mount point>   <type>  <options>                       <dump>  <pass>
#/dev/ram0 /      ext2    defaults     0      0
proc  /proc      proc    defaults     0      0
sysfs /sys       sysfs   defaults     0      0
none  /dev/pts   devpts  gid=5,mode=620 0    0
```

[etc/group]

```
root:x:0:root
sshd:x:74:
```

[etc/hosts]

```
127.0.0.1       localhost.localdomain   localhost
# Additional names can be added
#192.168.1.106 squeezel
```

[etc/inittab]

```
# This is run first except when booting in single-user mode.
#

::sysinit:/etc/rc.d/rcS

#
#
#con::respawn:/sbin/getty console
# Put a getty on the serial port
ttyS0::respawn:/sbin/getty -L ttyS0 115200 vt100


#
# /sbin/getty invocations for selected ttys
#
#tty1::respawn:/sbin/getty 115200 tty1
#tty2::respawn:/sbin/getty 115200 tty2
#tty3::respawn:/sbin/getty 115200 tty3
#tty4::respawn:/sbin/getty 115200 tty4
#tty5::respawn:/sbin/getty 115200 tty5
#tty6::respawn:/sbin/getty 115200 tty6
#tty7::respawn:/sbin/getty 115200 tty7
#tty8::respawn:/sbin/getty 115200 tty8
#tty9::respawn:/sbin/getty 115200 tty9
#
#
# Example of how to put a getty on a serial line (for a terminal)
#
#::respawn:/sbin/getty -L ttyS0 9600 vt100
#::respawn:/sbin/getty -L ttyS1 9600 vt100
#
# Example how to put a getty on a modem line.
#::respawn:/sbin/getty 57600 ttyS2
#


# Stuff to do when restarting the init process
::restart:/sbin/init
#

# Stuff to do before rebooting
::ctrlaltdel:/sbin/reboot
::shutdown:/etc/rc.d/rcE
```

```
::shutdown:/bin/umount -a -r
::shutdown:/sbin/swapoff -a
```

[etc/login.defs]

```
# *REQUIRED*
#   Directory where mailboxes reside, _or_ name of file, relative to the
#   home directory.  If you _do_ define both, MAIL_DIR takes precedence.
#   QMAIL_DIR is for Qmail
#
#QMAIL_DIR     Maildir
MAIL_DIR       /var/spool/mail
#MAIL_FILE     .mail

# Password aging controls:
#
#       PASS_MAX_DAYS  Maximum number of days a password may be used.
#       PASS_MIN_DAYS  Minimum number of days allowed between password changes.
#       PASS_MIN_LEN   Minimum acceptable password length.
#       PASS_WARN_AGE  Number of days warning given before a password expires.
#
PASS_MAX_DAYS  99999
PASS_MIN_DAYS  0
PASS_MIN_LEN   5
PASS_WARN_AGE  7

#
# Min/max values for automatic uid selection in useradd
#
UID_MIN                 500
UID_MAX               60000

#
# Min/max values for automatic gid selection in groupadd
#
GID_MIN                 500
GID_MAX               60000

#
# If defined, this command is run when removing a user.
# It should remove any at/cron/print jobs etc. owned by
# the user to be removed (passed as the first argument).
#
#USERDEL_CMD   /usr/sbin/userdel_local

#
# If useradd should create home directories for users by default
# On RH systems, we do. This option is ORed with the -m flag on
# useradd command line.
#
CREATE_HOME    yes
```

[etc/nsswitch.conf]

```
# /etc/nsswitch.conf
#
# Example configuration of GNU Name Service Switch functionality.
# If you have the `glibc-doc' and `info' packages installed, try:
# `info libc "Name Service Switch"' for information about this file.

passwd:         files
group:          files
shadow:         files

hosts:          files dns
networks:       files

protocols:      files
services:       files
ethers:         files
rpc:            files

netgroup:       files
```

[etc/passwd]

```
root:x:0:0:Linux User,,,:/root:/bin/sh
sshd:x:74:74:Privilege-separated SSH:/var/empty/sshd:/bin/false
```

[etc/securetty]

```
# /etc/securetty: list of terminals on which root is allowed to login.
# See securetty(5) and login(1).
console

# for people with serial port consoles
ttyS0

# for devfs
tts/0

# Standard consoles
tty1
tty2
tty3
tty4
tty5
tty6
tty7
tty8
tty9
tty10
tty11
tty12
tty13
tty14
tty15
tty16
tty17
tty18
tty19
tty20
tty21
tty22
tty23
tty24
tty25
tty26
tty27
tty28

# Same as above, but these only occur with devfs devices
vc/1
vc/2
vc/3
vc/4
vc/5
vc/6
vc/7
vc/8
vc/9
vc/10
vc/11
vc/12
vc/13
vc/14
vc/15
vc/16
vc/17
vc/18
vc/19
vc/20
vc/21
vc/22
vc/23
vc/24
```

```
vc/25
vc/26
vc/27
vc/28
vc/29
vc/30
vc/31
vc/32
vc/33
vc/34
vc/35
vc/36

# psudo terminals used by telnet
pts/0
pts/1
pts/2
pts/3
pts/4
pts/5
pts/6
pts/7
```
[etc/shadow]
```
root::12439:0:99999:7:::
sshd:*:11880:0:99999:7:-1:-1:0
```

[etc/udhcpd.conf]
```
# Sample udhcpd configuration file (/etc/udhcpd.conf)
# The start and end of the IP lease block
# The interface that udhcpd will use
interface       eth0            #default: eth0
# The maximim number of leases (includes addressesd reserved
# by OFFER's, DECLINE's, and ARP conficts
#max_leases     254             #default: 254
# If remaining is true (default), udhcpd will store the time
# remaining for each lease in the udhcpd leases file. This is
# for embedded systems that cannot keep time between reboots.
# If you set remaining to no, the absolute time that the lease
# expires at will be stored in the dhcpd.leases file.
#remaining      yes             #default: yes
# The time period at which udhcpd will write out a dhcpd.leases
# file. If this is 0, udhcpd will never automatically write a
# lease file. (specified in seconds)
#auto_time      7200            #default: 7200 (2 hours)
# The amount of time that an IP will be reserved (leased) for if a
# DHCP decline message is received (seconds).
#decline_time   3600            #default: 3600 (1 hour)
# The amount of time that an IP will be reserved (leased) for if an
# ARP conflct occurs. (seconds
#conflict_time  3600            #default: 3600 (1 hour)
# How long an offered address is reserved (leased) in seconds
#offer_time     60              #default: 60 (1 minute)
# If a lease to be given is below this value, the full lease time is
# instead used (seconds).
#min_lease      60              #defult: 60
# The location of the leases file
#lease_file     /var/lib/misc/udhcpd.leases     #defualt: /var/lib/misc/udhcpd.leases
# The location of the pid file
#pidfile        /var/run/udhcpd.pid     #default: /var/run/udhcpd.pid
# Everytime udhcpd writes a leases file, the below script will be called.
# Useful for writing the lease file to flash every few hours.
#notify_file                            #default: (no script)
#notify_file    dumpleases      # <--- usefull for debugging
# The following are bootp specific options, setable by udhcpd.
#siaddr         192.168.0.22            #default: 0.0.0.0
#sname          zorak                   #default: (none)
#boot_file      /var/nfs_root           #default: (none)
# The remainer of options are DHCP options and can be specifed with the
# keyword 'opt' or 'option'. If an option can take multiple items, such
# as the dns option, they can be listed on the same line, or multiple
# lines. The only option with a default is 'lease'.
```

[etc/rc.d/rcE]

```
#!/bin/sh
#End the local scripts
echo "Shutting Down Local Systems"
for i in /etc/rc.d/rc.local/E*
do
   $i stop
done
#umount all before dying
/bin/umount -a
echo "DOne"
```

[etc/rc.d/rcS]

```
#!/bin/sh
echo "starting up local systems"
/bin/mount -a
# below getting rid of ram being mounted ro
/bin/mount -o remount /
#
# The following is for dhcp
#
#ifconfig eth0 0.0.0.0
#/sbin/udhcpc
#
# Instead, if you want static IP address
#
#ifconfig eth0 192.168.1.13  netmask 255.255.252.0
#route add default gw 192.168.1.1
#
# Run ssh daemon
#/sbin/sshd
for i in /etc/rc.d/rc.local/S*
do
   $i start
done
echo "done"
```

[etc/rc.d/init.d/udev]

```
#!/bin/sh -e
# chkconfig: S 4 99
#set -x
PATH="/sbin:/bin:/usr/bin:/usr/sbin"

# defaults
tmpfs_size="10M"
udev_root="/dev"

################################################################################

# we need to unmount /dev/pts/ and remount it later over the tmpfs
unmount_devpts() {
  if mountpoint -q /dev/pts/; then
    umount -l /dev/pts/
  fi

  if mountpoint -q /dev/shm/; then
    umount -l /dev/shm/
  fi
}

# mount a tmpfs over /dev, if somebody did not already do it
mount_tmpfs() {
  if grep -E -q "^[^[:space:]]+ /dev tmpfs" /proc/mounts; then
    return 0
  fi

  # /dev/.static/dev/ is used by MAKEDEV to access the real /dev/ directory.
  # /etc/udev/ is recycled as a temporary mount point because it's the only
  # directory which is guaranteed to be available.
  mount -n --bind /dev /etc/udev
```

```
  echo -n "Mounting a tmpfs over /dev..."
  if ! mount -n -o size=$tmpfs_size,mode=0755 -t tmpfs none /dev; then
    echo " FAILED!"
    echo "FATAL: udev requires tmpfs support, not started."
    umount /etc/udev
    exit 1
  fi

  # using ln to test if /dev works, because touch is in /usr/bin/
  if ln -s test /dev/test-file; then
    rm /dev/test-file
    echo "done."
  else
    echo " FAILED!"
    echo "FATAL: udev requires tmpfs support, not started."
    umount /etc/udev
    umount /dev
    exit 1
  fi

  mkdir -p /dev/.static/dev
  chmod 700 /dev/.static/
  mount -n --move /etc/udev /dev/.static/dev
}

# I hate this hack.  -- Md
make_extra_nodes() {
  [ -e etc/udev/links.conf ] || return
  grep '^[^#]' /etc/udev/links.conf | \
  while read type name arg1; do
    [ "$type" -a "$name" -a ! -e "/dev/$name" -a ! -L "/dev/$name" ] ||continue
    case "$type" in
      L) ln -s $arg1 /dev/$name ;;
      D) mkdir -p /dev/$name ;;
      M) mknod -m 600 /dev/$name $arg1 ;;
      *) echo "links.conf: unparseable line ($type $name $arg1)" ;;
    esac
  done
}

# this function is duplicated in preinst, postinst and d-i
supported_kernel() {
  case "$(uname -r)" in
    2.[012345].*|2.6.[0-7]|2.6.[0-7][!0-9]*) return 1 ;;
  esac
  return 0
}

# Kernels < 2.6.10 break some drivers when udevsend is used as the hotplug
# multiplexer. See #297481 for details.
events_not_ordered() {
  case "$(uname -r)" in
    2.6.[0-9]|2.6.[0-9][!0-9]*) return 0 ;;
  esac
  return 1
}

# shell version of /usr/bin/tty
my_tty() {
  [ -x /usr/bin/readlink ] || return
  [ -e /proc/self/fd/0 ] || return
  readlink /proc/self/fd/0 || true
}

warn_if_interactive() {
  if [ "$RUNLEVEL" = "S" -a "$PREVLEVEL" = "N" ]; then
    return
  fi

  TTY=$(my_tty)
```

```
  if [ -z "$TTY" -o "$TTY" = "/dev/console" -o "$TTY" = "/dev/ttyS0" ]; then
    return
  fi

  printf "\n\n\nIt has been detected that the command\n\n\t$0 $*\n\n"
  printf "has been run from an interactive shell $TTY\n"
  printf "It will probably not do what you expect, so this script will wait\n"
  printf "1 seconds before continuing. Press ^C to stop it.\n"
  printf "RUNNING THIS COMMAND IS HIGHLY DISCOURAGED!\n\n\n\n"
  sleep 1
}

###############################################################################

[ -x /sbin/udevstart ] || exit 0

. /etc/udev/udev.conf

if [ "$UDEV_DISABLED" = "yes" ]; then
  echo "udev disabled on the kernel command line, not started"
  exit 0
fi

if ! supported_kernel; then
  echo "udev requires a kernel >= 2.6.8, not started."
  exit 0
fi

if ! grep -q '[[:space:]]tmpfs$' /proc/filesystems; then
  echo "udev requires tmpfs support, not started."
  exit 0
fi

if [ ! -e /proc/sys/kernel/hotplug ]; then
  echo "udev requires hotplug support, not started."
  exit 0
fi

###############################################################################

udev_root=${udev_root%/}

if [ "$udev_root" != "/dev" ]; then
  echo "WARNING: udev_root != /dev/"

case "$1" in
  start)
    if [ -e "$udev_root/.udevdb" ]; then
      if mountpoint -q /dev/; then
        echo "FATAL: udev is already active on $udev_root."
        exit 1
      else
        echo "WARNING: .udevdb already exists on the old $udev_root!"
      fi
    fi
    mount -n -o size=$tmpfs_size,mode=0755 -t tmpfs none $udev_root
    echo -n "Creating initial device nodes..."
    udevstart
    echo "done."
    ;;
  stop)
    start-stop-daemon --stop --exec /sbin/udevd --oknodo --quiet
    echo -n "Unmounting $udev_root..."
    # unmounting with -l should never fail
    if umount -l $udev_root; then
      echo "done."
    else
      echo "failed."
    fi
    ;;
  restart|force-reload)
```

```
    $0 stop
    $0 start
    ;;
  *)
    echo "Usage: /etc/init.d/udev {start|stop|restart|force-reload}"
    exit 1
    ;;
esac

  exit 0
fi # udev_root != /dev

################################################################################

# When modifying this script, do not forget that between the time that
# the new /dev has been mounted and udevstart has been run there will be
# no /dev/null. This also means that you cannot use the "&" shell command.

case "$1" in
  start)
    if [ -e "$udev_root/.udevdb" ]; then
      if mountpoint -q /dev/; then
        echo "FATAL: udev is already active on $udev_root."
        exit 1
      else
        echo "WARNING: .udevdb already exists on the old $udev_root!"
      fi
    fi
    warn_if_interactive
    if ! events_not_ordered; then
      echo /sbin/udevsend > /proc/sys/kernel/hotplug
    fi
    unmount_devpts
    mount_tmpfs
    [ -d /proc/1 ] || mount -n /proc
    echo -n "Creating initial device nodes..."
    udevstart
    echo "done."
    make_extra_nodes
    ;;
  stop)
    warn_if_interactive
    start-stop-daemon --stop --exec /sbin/udevd --oknodo --quiet
    unmount_devpts
    if [ -d /dev/.static/dev/ ]; then
      umount -l /dev/.static/dev/ || true
    fi
    echo -n "Unmounting /dev..."
    # unmounting with -l should never fail
    if umount -l /dev; then
      echo "done."
      if [-x "/etc/init.d/mountvirtfs" ]; then
      /etc/init.d/mountvirtfs start
      else
      echo "No mountvirtfs"
      fi
    else
      echo "failed."
    fi
    ;;
  restart|force-reload)
    start-stop-daemon --stop --exec /sbin/udevd --oknodo --quiet
    echo -n "Recreating device nodes..."
    udevstart
    make_extra_nodes
    echo "done."
    ;;
  *)
    echo "Usage: /etc/init.d/udev {start|stop|restart|force-reload}"
    exit 1
    ;;
```

```
esac
exit 0
```

<END OF FILE>