Ubuntu 10.04 Startup Guide for the i.MX5X board

freescale™
semiconductor

Content

1 Introduction

This document shows how to run Ubuntu 10.04 (lucid) on an i.MX51 EVK and i.MX53 boards.

The first chapter lists the required hardware equipment and cables

The second chapter is a quick start up guide. Your system might come with a pre-installed Ubuntu 10.04 file system on an SD card.

The third chapter focuses on programming the Ubuntu 10.04 root file system to an SD card.

We will use the i.MX5x acronym to refer to either i.MX51 or i.MX53 whenever instruction applies to both systems.

2 Hardware setup

2.1 MX51 EVK setup

The Figure 1 shows an i.MX51 EVK board and its connectors:

DVI Connector

UART Connector

LVDS Connector

Power Jack

DISP-Card Connector

Ethernet Connector

Micro_USB Connector

Dual-USB_A Connector

MIC_IN Jack

MMC/SD Connector

Audio Jack

Boot swicthes

Mini_PCI Connector

POWER Button SW1

ACCESSORY_CARD Connector

RESET Button SW2
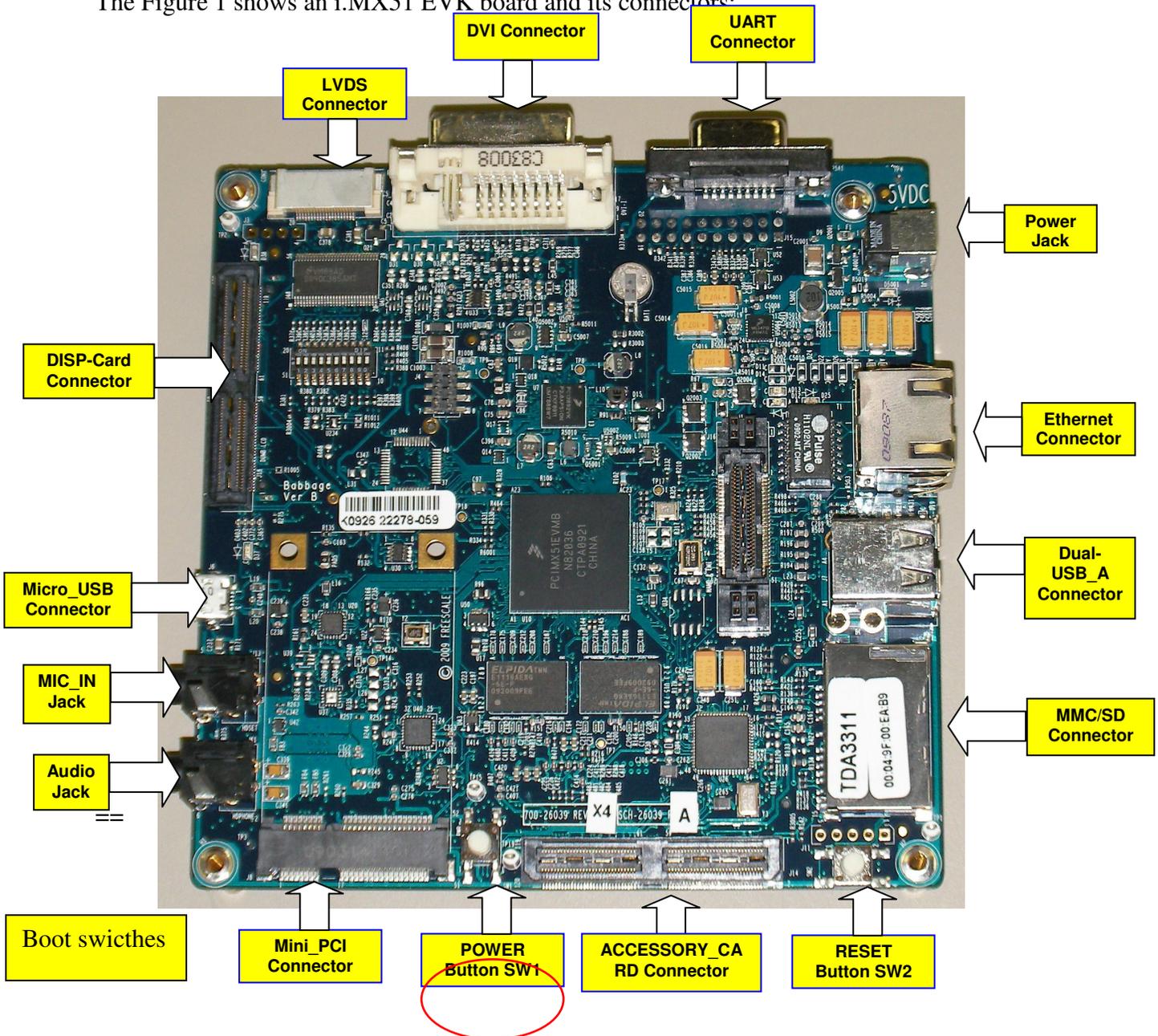
Figure 1: i.MX5x EVK board connectors and switches

The following cables and equipment are required:

+ A DVI cable
+ A monitor with a DVI input. Minimum supported resolution must be 800x600.
+ A DB9 cable for the serial console
+ A USB mouse and keyboard (connect them to the Dual USB connector shown above)
+ Speakers or headphones with a 3.5 mm stereo jack
+ An Ethernet cable
+ A 5V power supply
+ A PC running Windows or Linux

Configure the i.MX51 EVK board to boot off of an SD card by setting the boot switches SW7 and SW8 to ON and by setting the other boot switches to off. For the switches settings of other boot modes, see i.MX51_EVK_Linux_BSP_UserGuide.pdf.

The serial port is configured as follows: 115200 BPS, 8 bits, no parity, 1 stop bit and no flow control. Make sure the serial port terminal running on the PC is configured accordingly.

2.2 MX53 EVK setup
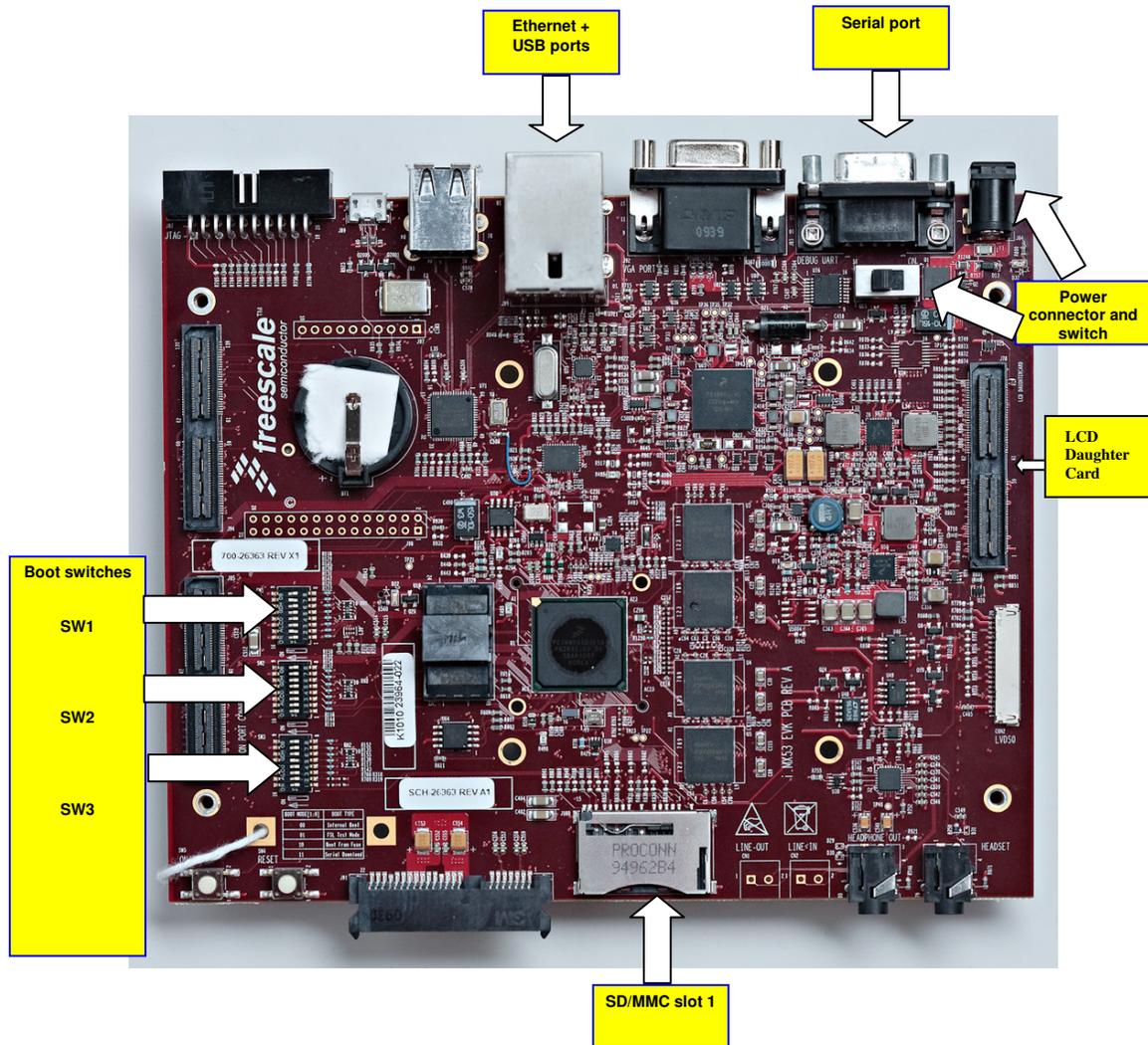
Figure above takes MX53 EVK RevA board (Red color) as the example. MX53 EVK RevB board (Blue color) is similar to RevA board. Please refer to Hardware documentation to know the detailed difference between RevA and RevB boards.

The following cables and equipment are required:

+ A 4 + GB SD card (we recommend SanDisk's)
+ A DB9 cable for the serial console
+ A USB mouse and keyboard (connect them to the Dual USB connector shown above)
+ a WVGA LCD display or DVI daughter board which connects to J78.
+ An Ethernet cable
+ A 5V power supply
+ A PC running Linux, equipped with an SD card reader

Configure the i.MX53 EVK board to boot off of an SD card in upper slot by setting the switches on:

+ SW1 's switches 7 ON. All others set to OFF.
+ All SW2's switches set to off
+ SW3's switch 6 set to on (4-bit)/off (1-bit), all others set to off.

For the switches settings of other boot modes, see i.MX53_EVK_Linux_BSP_UserGuide.pdf.

The serial port is configured as follows: 115200 BPS, 8 bits, no parity, 1 stop bit and no flow control. Make sure the serial port terminal running on the PC is configured accordingly.

2.3 MX53 START setup

Please refer to HW guide to setup MX53 START board.

2.4 MX53 SABRE Tablet setup

Please refer to HW guide to setup MX53 SABRE Tablet board.

3 Quick start up

Your system might come with a pre-installed Ubuntu 10.04 file system on an SD card.

3.1 MX51 Start up

To start the i.MX51 off that SD card, please insert the SD card into the SD/MMC slot 1 located at the bottom of the board (two ports are available). Press the power button SW1 for at least 3 seconds. Messages will be printed out to the serial port, and will show up on the terminal.

3.2 MX53 EVK Start up

To start the i.MX53 off that SD card, please insert the SD card into the SD/MMC slot 1 located at the top side of the board (two ports are available). Turn the power switch ON. Messages will be printed out to the serial port, and will show up on the terminal.

3.3 MX53 START Start up

To start the i.MX53 START off that SD card, please insert the Micro SD card into the slot 1. Press POWER KEY and ensure the LED becomes as blue color. Messages will be printed out to the serial port, and will show up on the terminal.

3.4 MX53 SABRE Tablet Start up

To start the i.MX53 SABRE Tablet off that SD card, please insert the SD card into J13. Press SW5 KEY and ensure the LED becomes as read color. Messages will be printed out to the serial port, and will show up on the terminal.

4 Build Root File System

4.1 Starting from ubuntu Image

If you are using Ubuntu demo image (with bootloader, kernel, lucid rootfs), you can use the following command to flash the images on SD card directly, the /dev/xxx is the SD device. The default display is VGA in the demo image.

```
#gunzip –c  imx53_smd_L2.6.35_11.09.01_ER_demo_image.gz > /dev/xxx
```

To update bootloader and kernel image, run the following commands:

```
sudo dd if=u-boot.bin of=/dev/xxx bs=512 skip=2 seek=2&& sync && sync
sudo dd if=uImage of=/xxx bs=512 seek=2048 && sync && sync
```

Then you can jump to chapter 5.

4.2 Starting from the FSL tar file:

Untar the FSL Ubuntu demo image (for example, the package name is lucid.tar.gz) file onto an SD card (4GB or larger) using a Linux host with an SD interface or using a target Linux system. The instructions below assume an i.MX5x board, running the Freescale Linux BSP.

Copying the boot loader image:

Use dd command in linux server to perform the copying. At first you can run following command to identify the device node assigned to the SD card:

```
$ cat /proc/partitions
major minor  #blocks  name
  8    0  78125000 sda
  8    1  75095811 sda1
  8    2        1 sda2
  8    5   3028221 sda5
  8   32 488386584 sdc
  8   33 488386552 sdc1
  8   16   3921920 mmcblk0
```

In this case, the device node assigned by kernel is /dev/mmcblk0. Then run the following command to copy the u-boot image to the SD card (pick the command that matches the platform you want to use):

```
### for MX51 EVK
sudo dd if=u-boot-bbg.bin  of=/dev/mmcblk0 bs=512 && sync && sync
### for MX53 EVK
```

```
sudo dd if=u-boot-mx53.bin of=/dev/mmcblk0 bs=512 && sync && sync
### for MX53 START
sudo dd if=u-boot-mx53-loco.bin of=/dev/mmcblk0 bs=512 && sync && sync
### for MX53 SABRE Tablet
sudo dd if=u-boot-mx53-smd.bin of=/dev/mmcblk0 bs=512 && sync && sync
```

Please note this operation will delete the partition table present on the medium. If want to update uboot to another version, may run the following command instead:

```
### for MX51 EVK
sudo dd if=u-boot-bbg.bin  of=/dev/mmcblk0 bs=512 skip=2 seek=2 && sync && sync
### for MX53 EVK
sudo dd if=u-boot-mx53.bin of=/dev/mmcblk0 bs=512 skip=2 seek=2 && sync && sync
### for MX53 START
sudo dd if=u-boot-mx53-loco.bin of=/dev/mmcblk0 bs=512 skip=2 seek=2&& sync && sync
### for MX53 SABRE Tablet
sudo dd if=u-boot-mx53-smd.bin of=/dev/mmcblk0 bs=512 skip=2 seek=2&& sync && sync
```

The first 1kB, which includes the partition table, will be preserved.

Copying the kernel image:

The following command will copy the kernel image to the SD card (i.MX5x).

```
sudo dd if=uImage of=/dev/mmcblk0 bs=512 seek=2048 && sync && sync
```

This will copy the uImage to the medium at offset 1MB.

Create the partition table:

```
sudo fdisk /dev/mmcblk0
```

Type the following parameters (each followed by <ENTER>):

```
u       [switch the unit to sectors instead of cylinders]
d       [repeat this until no partition is reported by the 'p' command ]
n       [create a new partition]
p       [create a primary partition]
1       [the first partition]
8192    [starting at offset sector #8192, i.e. 4MB, which leaves enough
space for the kernel, the boot loader and its configuration data]
<enter> [using the default value will create a partition that spans to
the last sector of the medium]
w       [ this writes the partition table to the medium and fdisk exits]
```

The partition for root file system should be located after kernel image. So the first 4M bytes (8192 sectors of 512 bytes) can be reserved for MBR, boot loader and kernel sections.

Create the ext3 or ext4 file system

```
sudo mkfs.ext3 /dev/mmcblk0p1
# or
sudo mkfs.ext4 /dev/mmcblk0p1
```

Mount the SD card and uncompress the root file system files:

```
sudo mount /dev/mmcblk0p1 /mnt/hd
cd /mnt/hd
sudo tar --numeric-owner -xzvf /<path>/lucid.tar.gz
```

Note the option "--numeric-owner" may not be available if you are using busybox. Using this option is mandatory though, so please make sure you use the full blown version of tar.

U-Boot needs to be configured. Please refer to BSP user guide. For example, check whether the MAC address of FEC is configured in fuse by "iim read fecmac". If it's the invalid value, "iim blow fecmac" can be used to blow right MAC address. Please note fuse operation is one-time operation. Or you can use "ethaddr" environment variable to set MAC address.

```
## Set MAC address ###
EVK U-Boot > setenv ethaddr 00:04:9f:00:ea:d3
....
```

The kernel programmed at the offset 1MB can be read using the 'mmc read' command

```
u-boot> mmc read 0 ${loadaddr} <offset> <kernel size in blocks>
```

4.3 Boot with Lucid rootfs:

The U-Boot configurations for **MX51 EVK** booting from SD with **DVI monitor (1024x768 resolution)**:

```
u-boot> setenv bootargs_mmc 'setenv bootargs ${bootargs} console=tty1 root=/dev/mmcblk0p1
rootwait rw ip=dhcp'
u-boot> setenv loadk  'mmc dev 0;mmc read ${loadaddr} 0x800 0x1800'
u-boot> setenv bootcmd_mmc 'run bootargs_base bootargs_mmc; run loadk; bootm'
u-boot> setenv bootcmd 'run bootcmd_mmc'
u-boot> saveenv
```

If using **WVGA panel** for **MX51 EVK**, change the boot command as the following:

```
u-boot> setenv bootargs_mmc 'setenv bootargs ${bootargs} console=tty1
video=mxcdi1fb:RGB565,CLAA-WVGA di1_primary root=/dev/mmcblk0p1 rootwait rw ip=dhcp'
```

The U-Boot configurations for **MX53 EVK** booting from SD with **WVGA panel**:

```
u-boot> setenv bootargs_mmc 'setenv bootargs ${bootargs} console=tty1 root=/dev/mmcblk0p1
rootwait rw ip=dhcp'
u-boot> setenv loadk  'mmc dev 0;mmc read ${loadaddr} 0x800 0x1800'
u-boot> setenv bootcmd_mmc  'run bootargs_base bootargs_mmc; run loadk; bootm'
u-boot> setenv bootcmd 'run bootcmd_mmc'
u-boot> saveenv
```

If using **DVI monitor** for **MX53 EVK**, change the boot command as the following:

```
u-boot> setenv bootargs_mmc 'setenv bootargs ${bootargs} console=tty1 video=mxcdi0fb:RGB24,
1024x768M-16@60 root=/dev/mmcblk0p1 rootwait rw'
```

The U-Boot configurations for **MX53 START** booting from SD with **VGA** output:

```
u-boot> setenv bootargs_mmc 'setenv bootargs ${bootargs} console=tty1 root=/dev/mmcblk0p1
rootwait rw video=mxcdi1fb:GBR24,VGA-XGA di1_primary vga ip=dhcp'
u-boot> setenv loadk  'mmc dev 0;mmc read ${loadaddr} 0x800 0x1800'
u-boot> setenv bootcmd_mmc  'run bootargs_base bootargs_mmc; run loadk; bootm'
u-boot> setenv bootcmd 'run bootcmd_mmc'
u-boot> saveenv
```

The U-Boot configurations for **MX53 START** booting from SATA with **VGA** output:

```
u-boot> setenv bootargs_sata 'setenv bootargs ${bootargs} console=tty1 root=/dev/sda1 rootwait rw
video=mxcdi1fb:GBR24,VGA-XGA di1_primary vga ip=dhcp'
u-boot> setenv loadk  'sata read 0 ${loadaddr} 0x800 0x1800'
u-boot> setenv bootcmd_sata  'run bootargs_base bootargs_sata; run loadk; bootm'
u-boot> setenv bootcmd 'run bootcmd_sata'
u-boot> saveenv
```

The U-Boot configurations for **MX53 SABRE Tablet** booting from SD with **HannStar LVDS** output:

```
u-boot> setenv bootargs_mmc 'setenv bootargs ${bootargs} console=tty1 root=/dev/mmcblk0p1
rootwait rw video=mxcdi1fb:RGB666,XGA di1_primary ldb=di1 ip=dhcp'
u-boot> setenv loadk  'mmc dev 0;mmc read ${loadaddr} 0x800 0x1800'
u-boot> setenv bootcmd_mmc  'run bootargs_base bootargs_mmc; run loadk; bootm'
u-boot> setenv bootcmd 'run bootcmd_mmc'
u-boot> saveenv
```

The U-Boot configurations for **MX53 SABRE Tablet** booting from SATA with **HDMI** output:

```
u-boot> setenv bootargs_sata 'setenv bootargs ${bootargs} console=tty1 root=/dev/sda1 rootwait rw
video=mxcdi0fb:RGB24,1024x768M@60 hdmi di0_primary'
u-boot> setenv loadk  'sata read 0 ${loadaddr} 0x800 0x1800'
u-boot> setenv bootcmd_sata  'run bootargs_base bootargs_sata; run loadk; bootm'
u-boot> setenv bootcmd 'run bootcmd_sata'
u-boot> saveenv
```

Log in as **lucid** (not root), the **password** is **lucid.**

Ensure the below changes are conducted in RFS for the usages.

/etc/init/ttymxc0.conf is created just like the following:

```
# ttymxc0 - getty
#
# This service maintains a getty on ttymxc0 from the point the system is
# started until it is shut down again.

start on stopped rc RUNLEVEL=[2345]
stop on runlevel [!2345]

respawn
exec /sbin/getty -L 115200 ttymxc0
```

The following configurations are added to /etc/rc.local (before the exit 0 line):

```
echo 16000 > /proc/sys/vm/min_free_kbytes
echo 1 1 > /proc/sys/vm/lowmem_reserve_ratio
#echo 2 > /proc/cpu/alignment
chmod -R o+w /sys/class/graphics/
# enable usb auto-suspend. Please remove the following commands if you don't hope USB mouse is
# powered off if no any activities for several minutes
echo auto > /sys/bus/usb/devices/1-1/power/level
echo auto > /sys/bus/usb/devices/2-1/power/level
echo auto > /sys/bus/usb/devices/3-1/power/level
echo auto > /sys/bus/usb/devices/2-1.1/power/level
```

Modify /etc/fstab to remove the swap device, for example:
```
#/etc/fstab: static file system information.
# Use 'blkid \-o value \-s UUID' to print the universally unique identifier
# for a device; this may be used with UUID= as a more robust way to name
# devices that works even if disks are added and removed. See fstab(5).
# <file system> <mount point>  <type> <options>     <dump> <pass>
proc          /proc          proc   nodev,noexec,nosuid 0      0
# / was on /dev/mmcblk1p2 during installation
UUID=24b3fb60-8116-460c-8848-ae00c124d78c /           ext4   errors=remount-ro 0      1
```

/usr/sbin/flash-kernel is modified as the following:

```
#!/bin/sh
echo okay;
```

Delete /mnt/hd/etc/udev/rules.d/70-persistent-net.rules by "rm \-f /etc/udev/rules.d/70-persistent-net.rules".

Check /etc/udev/rules.d/10-imx.rules and ensure the device modes are changed to grant permission for different users. For example,

```
# Miscellaneous devices
KERNEL=="mxc_iim",  MODE="0444", SYMLINK+="mxc_mem"
KERNEL=="mxs_viim", MODE="0444", SYMLINK+="mxc_mem"
KERNEL=="mxc_ipu",  MODE="0666"
KERNEL=="fb0",      MODE="0666"
KERNEL=="fb1",      MODE="0666"
KERNEL=="fb2",      MODE="0666"
KERNEL=="mxc_vpu",  MODE="0666"
SUBSYSTEM=="video", MODE="0666"
KERNEL=="gsl_kmod", MODE="0666"...
```

Add Ubuntu to user groups:

```
sudo usermod \-a \-G audio,video lucid
```

Fix audio (pulse audio appears to be broken):

```
pkill pulseaudio
sudo aptitude -y purge pulseaudio
sudo aptitude -y remove gstreamer0.10-pulseaudio
sudo aptitude -y install xfce4-mixer
sudo aptitude -y install aumix-gtk
```

Note if you remove the pulse audio, some features will not work normally. For example, the "System->Preference->Sound" entry and the Volume Control applet. When entering the applications, there will pop up "waiting for sound system to respond" and then nothing happened. The user can use the applications under "Applications -> Sound & Video -> aumix/mixer" for audio settings.

Install ntp:

```
sudo aptitude \-y install ntp
```

Configure ntp by editing /etc/ntp.conf to define the NTP server of your choice. For example:

```
server 10.208.0.120
```

Install FSL packages. For this you should already have copied all the *.deb files that come along with the Released BSP to the Target Board; or, for example, having them on a USB Key already connected to the target board. The deb files of each release can be

located in the demo image package. (Please install the version built with gcc4.4.4 for Ubuntu 10.04):

```
sudo dpkg \--force-architecture \-i \*.deb
depmod
```

Since GPU is binary release, if the customer uses different kernel version, they need to modify '/lib/modules/<version>/modules.dep' to reflect the change in module dependencies.

Flush data to SD card and reboot:

```
sync
sudo halt
```

Your system is ready to go. The following Freescale debian packages have been installed (available with this release) to allow Ubuntu to operate the i.MX5x specific features like video and graphics acceleration.

Debian package name	Description
imx-lib_<version>_armel.deb	Package name: imx-lib This debian package includes specific libraries for imx platform such as VPU, IPU libraries etc.
lmx-test-<version>-armel.dedb	This debianm package provides test routines
kernel_2.6.35-imx_<version>_armel.deb	Package name: kernel This debian package includes kernel image, kernel modules and relative kernel header files.
firmware-imx_<version>_armel.deb	This debian package includes the firmwares such as VPU, Atheros.
modeps_<version>_armel.deb	Module dependencies
amd-gpu-x11-bin-mx51_<version>_armel.deb	Package name: amd-gpu-x11-bin-mx51 This debian package includes GPU user space library and the applications for x11 on mx5 parts.
libz160-bin_<version>_armel.deb	Package name: libz160-bin

	This debian package include AMD GPU Z160 2D driver
xserver-xorg-video-imx_<version>_armel.deb	Package name: xserver-xorg-video-imx This debian package includes i.MX accelerated video driver for X11.
udev-fsl-rules_<version>_armel.deb	Debian package for i.mx udev rule
Multimedia Debian packages	See release note of Multimedia debian packages

Verify execution of EGL running in an X window by running either 'simple_draw' or 'es11ex' sample applications, both OpenGL-ES samples':

```
simple_draw –f <number of iterations>
es11ex
```

5 Known Issues

5.1 Flash support

- Flash 10 packages are not in default delivery release. Contact Product Marketing to get relative packages.

5.2 GPU support

- X-Acceleration only supports native build in this version. To do native build, do as the following:
  - Download xserver-xorg-video-imx source code to the target
  - ./configure --prefix=/user
  - make
  - make install prefix=<target_rootfs>/usr

6 FAQ

6.1 How to support Touch screen in Ubuntu

The following instructions show how to enable Ubuntu touchscreen in MX53 SMD board, this touch panel is "Hannstar p1003 touch screen" should be aware that only single touch is support by 10.04. The following steps are needed to calibrate the cursor even if the driver is already working.

1. Get evtouch debian package, since the evtouch package in lucid's repository have a bug cause X.org server crash. The package can be found on the demo image bundle of the release with this name. xserver-xorg-input-evtouch_0.8.8-3build1_armel.deb
   Install debian package:
   dpgk –i xserver-xorg-input-evtouch_0.8.8-3build1_armel.deb

2. Add the below sections in /etc/X11/xorg.conf

```
Section "InputDevice"
    Identifier "touchscreen"
    Driver "evtouch"
    Option "Device" "/dev/input/event2"
    Option "DeviceName" "touchscreen"
    Option "MinX" "0"
    Option "MinY" "0"
    Option "MaxX" "4352"
    Option "MaxY" "3200"
    Option "ReportingMode" "Raw"
    Option "Emulate3Buttons" "false"
    Option "Emulate3Timeout" "50"
    Option "SendCoreEvents" "On"
EndSection
```

Please check /var/log/Xorg.0.log and search "P1003" to get right /dev/input path. Then modify "Option Device" (shown as red color) as right path.

Change the "ServerLayout" Section to below:

```
Section "ServerLayout"
    Identifier    "Default Layout"
    Screen        "Default Screen"
    InputDevice "touchscreen"
EndSection
```

3. Use
   sudo service gdm restart

   To restart X server.

4. Here, your touch screen should start to work. If not, the most possible cause is the device number in xorg.conf: Option "Device" is not the right touch screen's device path. Please check kernel log to change the number.

If your LVDS uses eGalax touch screen, do as the followings:

- vi drivers/input/touchscreen/egalax_ts.c and enable single pointer support:

  #define FORCE_SINGLE_POINTER_SUPPORT

- Please check /var/log/Xorg.0.log and search "eGalax" to get right /dev/input path. Then modify /etc/X11/xorg.conf:

```
Section "InputDevice"
Identifier "touchscreen"
Driver "evtouch"
Option "Device" "/dev/input/event2"
Option "DeviceName" "touchscreen"
Option "MinX" "0"
Option "MinY" "0"
Option "MaxX" "32767"
Option "MaxY" "32767"
Option "ReportingMode" "Raw"
Option "Emulate3Buttons" "false"
Option "Emulate3Timeout" "50"
Option "SendCoreEvents" "On"
EndSection
```

- Refer to other steps mentioned in previous section.

6.2 How to test USB Bluetooth Dongle in Ubuntu

The following instructions demonstrate how to test Bluetooth via USB BT dongle (Model: BU_2073-J) in Ubuntu with MX51 EVK board. Please note the below configuration is setup when pulseaudio package is installed.

- Run the system. Then vi ~/.asoundrc

```
pcm.btheadset {
type bluetooth
device 00:07:A4:F3:19:D4 #modify the address as your headset address
profile "auto"
}
```

- Write a script "autorun_bt.headset" to run BT automatically:

```
echo "Kill PulseAudio"
clear
pkill pulseaudio
sleep 1

echo "Start PulseAudio"
clear
pulseaudio &
sleep 1

echo "Enable sound drivers by typing the following into a terminal"
```

```
clear
gksudo hciconfig hci0 voice 0x0060
sleep 1

echo "Now we n tell PulseAudio that your Bluetooth headset exists"
pactl load-module module-alsa-sink device=btheadset
sleep 8
pactl load-module module-alsa-source device=btheadset
sleep 2

echo "Ready"
sleep 2
```

- Run

```
chmod a+x autorun_bt.headset
```

- Insert USB dongle.
- Type "System -> Preferences-> Bluetooth". If USB dongle can not be found, do as follows:

```
/etc/init.d/bluetooth stop
```

- Insert USB dongle again, run:

```
/etc/init.d/bluetooth start
```

- See whether USB dongle is found and connected. If not, connect it manually.
- Run

```
sh autorun_bt.headset
aplay -D btheadset -f s16_le /usr/share/sounds/alsa/Front_Left.wav
```

- The audio can be heard clearly.

6.3 Note for AR3001 Bluetooth usage

To make AR3001 Bluetooth work well, the MAC address must be changed as unique value to avoid conflict with the other:

echo 0060417fXXXX > /lib/firmware/ar3k/30101/ar3kbdaddr.pst

XXXX can be changed to one unique value different from others.

6.4 How to support WiFi in Ubuntu

The following instructions demonstrate how to test WiFi via AR6003 + 3001 Dual band combo card in MX53 SMD:

- Ensure firmware debian package is installed.
- vi /etc/modules to load ath6kl module.
  ath6kl
- Ensure Combo card and WiFi antenna are inserted. Restart the board.
- Enable wireless and select to use wireless connection.