Getting Started With Inflexion UI

Workshop – Installing Inflexion UI on Linux

Software Version: 2.3 for iMX5x Processors

September 2011

Document Revision 1.0

Contractor/manufacturer is:
Mentor Graphics Corporation
8005 S.W. Boeckman Road, Wilsonville, Oregon 97070-7777.
Telephone: 503.685.7000
Toll-Free Telephone: 800.592.2210
Website: www.mentor.com
SupportNet: supportnet.mentor.com/
Send Feedback on Documentation: supportnet.mentor.com/user/feedback_form.cfm

Table of Contents

List of Figures

Introduction

This document explains how to:

- Configure and use a VMware Linux image with i.MX53 Linux BSP installed and configured for the i.MX53 Quick Start Board for Inflexion-based development.
- Build and deploy the demonstration Inflexion Linux application on the iMX53QSB target device

Configuring the development environment

This section explains how to set up your development environment for Inflexion-based development. After completing the steps in this section, you will be ready to create and build Inflexion-based applications on the iMX53QSB.

Prerequisites

To use Inflexion UI for Linux you must have the following available:

- **VMware Linux image**. This VM is available on the iMX53QSB software CD supplied with the hardware device.

  More up to date VM images are available from the Freescale iMX53QSB product page. Note this document is verified against the install CD version of the VM image.

  http://www.freescale.com/webapp/sps/site/prod_summary.jsp?code=IMX53QSB&fpsp=1&tab=Design_Tools_Tab

  .
- **Inflexion 2.3,** for Freescale iMX Processors.
- **TCPIP connection,** to hardware via Ethernet cable (for deploying binaries)
- **Serial or USB cable connection,** to hardware (for establishing hardware IP address)
- **Micro SD card (4 Gig) and HC card reader,** for flashing the bootable rootfs built by LTIB to.

Procedure overview

The overall steps for the installation procedure are:

1. Install and configure the VM development environment.
2. Creating an booting SD card for iMX53QSB
3. Installing the Inflexion runtime library to bootable image
4. Building and deploying the demo application

5. Running the demo application on target

Install and configure the VM development environment

The VM can be found and installed following these instructions:

1. Download the free VMware Player from www.vmware.com.
2. Install the VMware Player on your host PC.
3. Insert the iMX53QSB software CD
4. Extract file vm_Ubuntu.zip from DVD-ROM (it may need up to 9 Gig of space) It will create vm_Ubuntu directory.

   Alternatively unzip the downloaded file if it was obtained from the weblink in the prerequisites section.
5. That directory contains an Ubuntu virtual computer that can be used with VMware Player.
6. To use the virtual computer, navigate into the new directory and double-click the file ending in ".vmx". This will start VMware Player.
   - If VMware Player asks if the virtual computer was "moved" or "copied", please choose "copied". This happens the first time you run the virtual machine, when VMware Player discovers its existence.
   - Select no to the floppy drive connection message if it appears.
7. When the virtual computer is running, you should be automatically logged in. If this does not occur, then log into the virtual machine with the following credentials:

   username= lucid

   password= lucid

The Linux Target Image Builder (LTIB) is the configuration-and-build tool that is used to make binaries for the i.MX target systems, including the i.MX53. Within the VM the LTIB directory for the i.MX53 Linux BSP is located at:

/home/lucid/ltib/

From this point on, the LTIB directory will be referred to as "<ltib>" in example command lines. LTIB has been pre-installed within the VM.

The LTIB installation needs to be set up to produce a rootfs folder.

1. Open a new terminal window and navigate to the <ltib> directory

Adding Linux Multimedia Codecs

Before using LTIB to build the rootfs folder we need to add the multimedia package to the build.

1. On the Freescale website navigate to the product page for the iMX53QSB board.
2. On the **Downloads** tab for your board, under **Codecs and other Algorithms**, find the multimedia codecs (named, for example, "IMX53_11_01_LINUX_MMCODECS").
3. Click the Download button to the right of the package listing.
4. Unpack the contents of the download file **IMX_MMCodecs_11.01_Bundle.tar.gz**
5. From the extracted contents, unpack the **IMX_MMCODECS_11.01.tar.gz**
6. From the extracted contents, unpack the **IMX_MMCODECS_1.9.7_SRC.tar.gz**
7. Create a **pkgs** named folder like the below directory under File System **var/tmp/** and copy contents of the folder created by the final extract to it:

   /var/tmp/pkgs

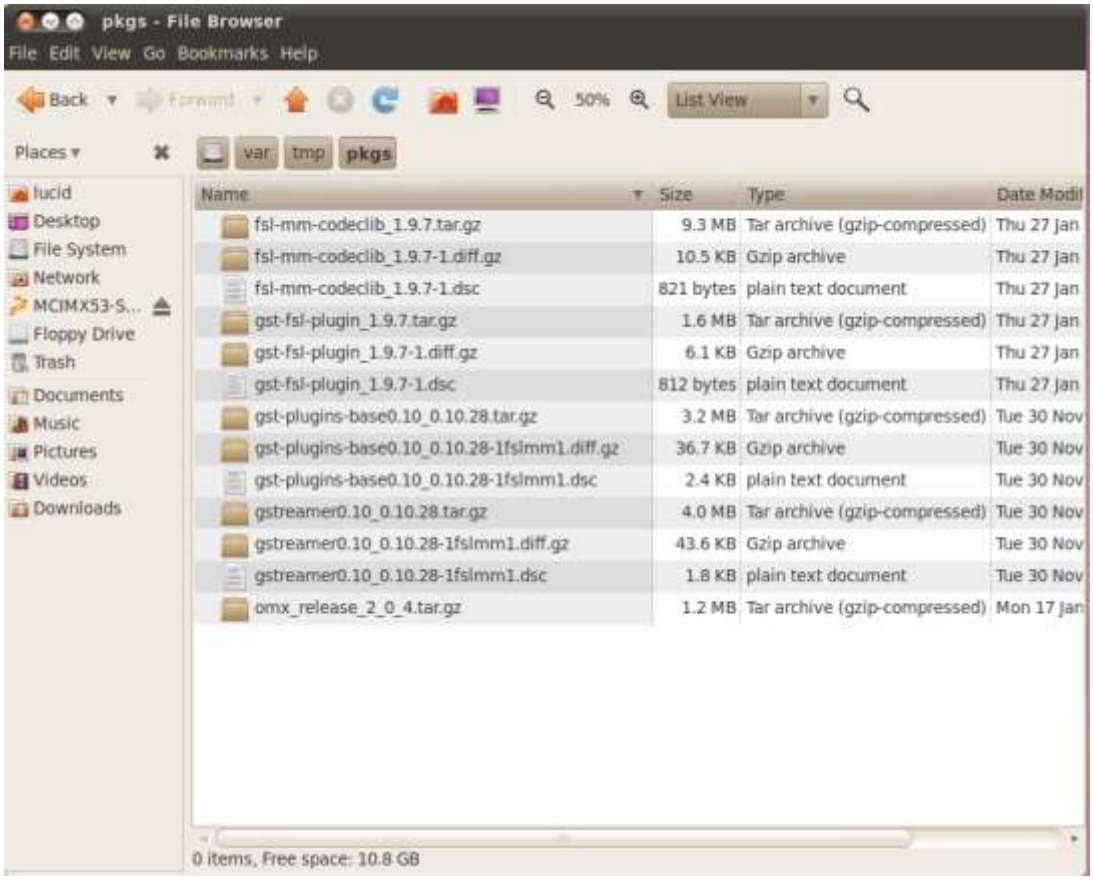

Figure 1: Extracted and copied multimedia codec files

These multimedia packages may need renaming to replace "_" with "-". To rename all the **..._1.9.7.tar.gz** to **...-1.9.7.tar.gz** in one go:

1. Open a new Terminal window
2. Navigate to **/var/tmp/pkgs/**
3. Enter the following to rename the files:

   for file in `ls *_1.9.*`; do echo $file; mv $file `echo $file | sed -e {s/_1.9./-1.9./g}`; done
4. In the LTIB build terminal, change LTIB to pick up the MM 1.9.7 packages:

   sed -i -e {s/1.9.6/1.9.7/g} dist/lfs-5.1/fsl-mm/fsl-mm-codeclib.spec

   sed -i -e {s/1.9.6/1.9.7/g} dist/lfs-5.1/fsl-mm/gst-fsl-plugin.spec

Applying any iMX53QSB patches

Users of the Seiko WVGA Panel with iMX53 Start Board should apply a required patch originally provided in the Freescale Community Forums linked below.

http://imxcommunity.org/group/imx53quickstartboard/forum/topics/seiko-wvga-panel-with-imx53?xg_source=activity

This patch is supplied in the workshop material.

1. From this training workshop directory copy the file **…\Getting Started With Inflexion UI\iMX53QSB Patches\11.05.01_da9052-pressure-key-patch-mx53.tgz**

   To the VM file system:

   \lucid\Downloads
2. Extract the contents of the tgz file.
3. **Note**, this patch is only fully valid with BSP 11.05, follow the below steps for generic application of the patch instead of the ReadMe.txt included in patch files.
4. Copy new '00027-Add-da9052-pressure-key-to-mx53.patch' file to your packages folder (normally File System/opt/freescale/pkgs)
5. Open the file kernel spec file:

   gedit config/platform/imx/kernel-2.6.35.spec.in
6. Add the patch to the kernel spec between the URL and BuildRoot lines, e.g.:

```
URL             : http://www.kernel.org/
Patch27    : 00027-Add-da9052-pressure-key-to-mx53.patch
BuildRoot       : %{_tmppath}/%{name}
Prefix          : %{pfx}
```

7. Add a reference to the patch at the bottom of the file, e.g:

```
./patches/patch-kernel.sh
%patch27 -p1
fi
```

The patch will now be included when LTIB builds. Note if you have already built LTIB you will need to clean the kernel by removing the build RPM folder and force a rebuild. See the patch ReadMe.txt file for guidance.

Building rootfs with LTIB

We are now ready to use the LTIB tool to build the rootfs.

1. In the LTIB build terminal start the LTIB config wizard, on first run it may take a sometime to launch while it configures packages:

   ./ltib -c

2. Wait till LTIB config wizard launches, press to return to confirm Platform Choice, ensure Freescale iMX reference boards is marked with an X:

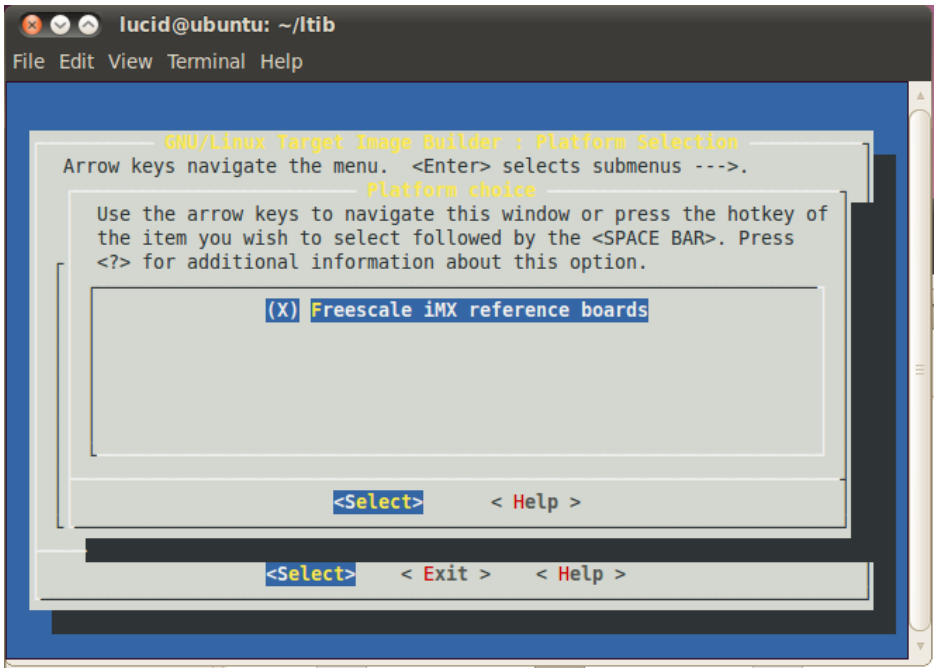

Figure 2: Initial LTIB Configuration screens

3. Press Return, then Esc to exit, saving when prompted.
4. From the next screen under ‘Choose the platform type’ press return to enter selection screen, from the presented list scroll down and change platform type to **imx5x** by pressing Space.

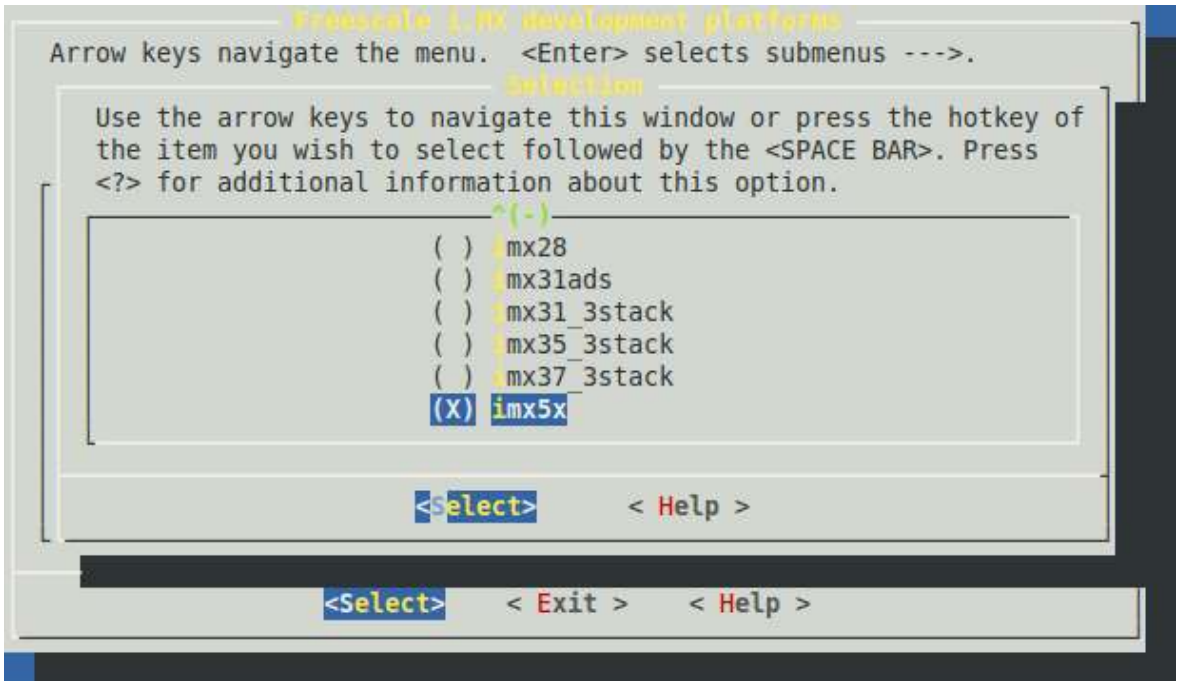

Figure 3: Using LTIB to select the imx5x platform type

5. Scroll down to ‘Selection’ and press Return. From the pop up list use space to select Move to and select ‘Minimum bootable root file system’

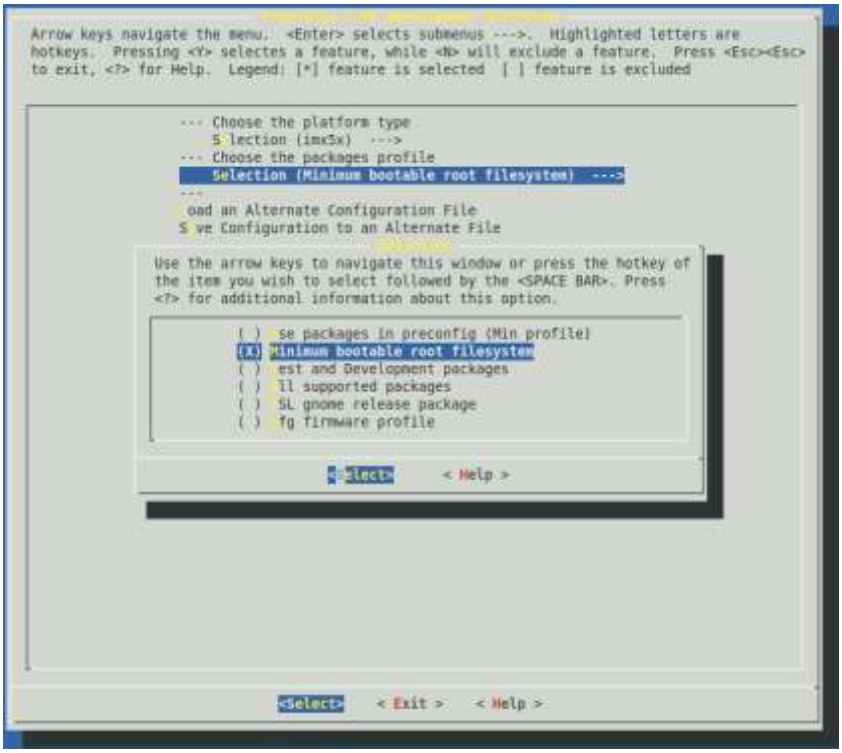

Figure 4: Using LTIB to set bootable properties

6. Press Esc to exit, Save when prompted.
7. From the next screen Scroll to and select board under the ‘Choose your board for U boot’. From the presented list scroll down and using space select `[ ]mx53 loco`

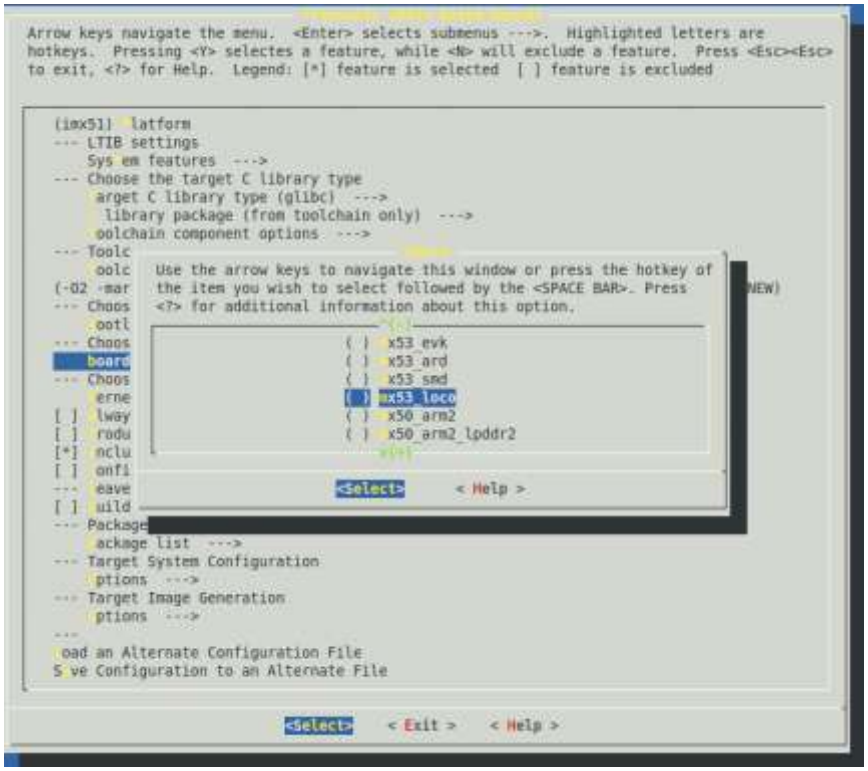

Figure 5: Using LTIB to set the hardware board type

8. Scroll down and under ‘Choose your kernel’ option use space to set:

```
[*] Configure the kernel
```

9. Scroll down and select ‘Package list --->’ using return
10. From the Package selection screen select the following Packages from the list and subsequent sub-lists:

```
[*] amd-gpu-bin-mx51
←     Freescale Multimedia Plugins/Codecs  --->
     [*] gstreamer-fsl-plugins
[*] bash
[*] dropbear ssh client/server
[*] freetype
[*]   gstreamer-plugins-good
[*]   gstreamer-plugins-bad
[*]   gstreamer-plugins-ugly
[*]   gstreamer FFmpeg plugins
[*] libjpeg
[*] libpng
```

```
[*] net-tools
```

11. Select Exit
12. From the list select ‘Target System Configuration Options --->’. From the next screen select (freescale target hostname), change the name in the dialog to ifxtraining

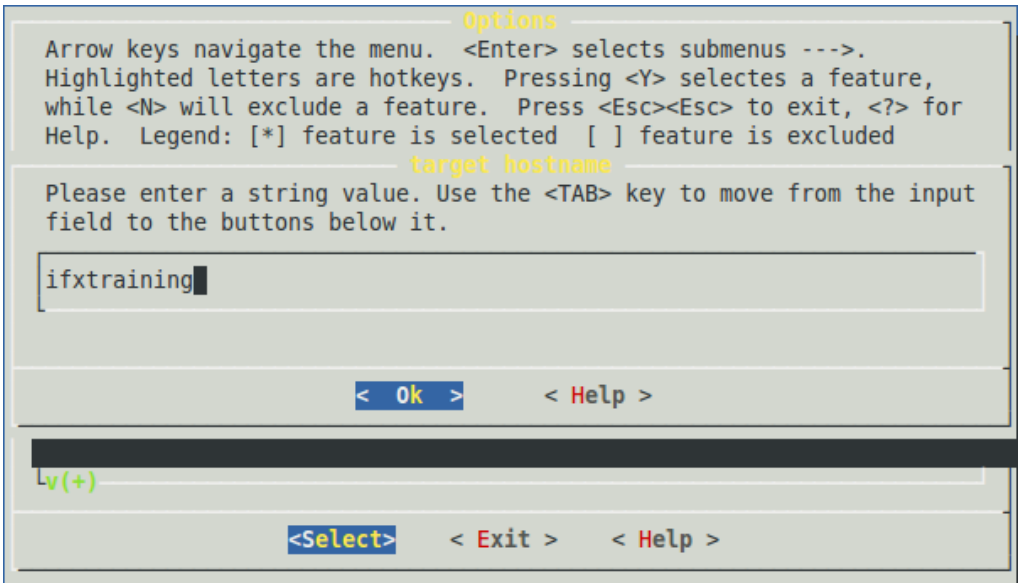

Figure 6: Using LTIB to set the target hostname

13. Scroll down and select ‘( ) load these modules at boot’, in the dialog box enter **gpu** and press return.

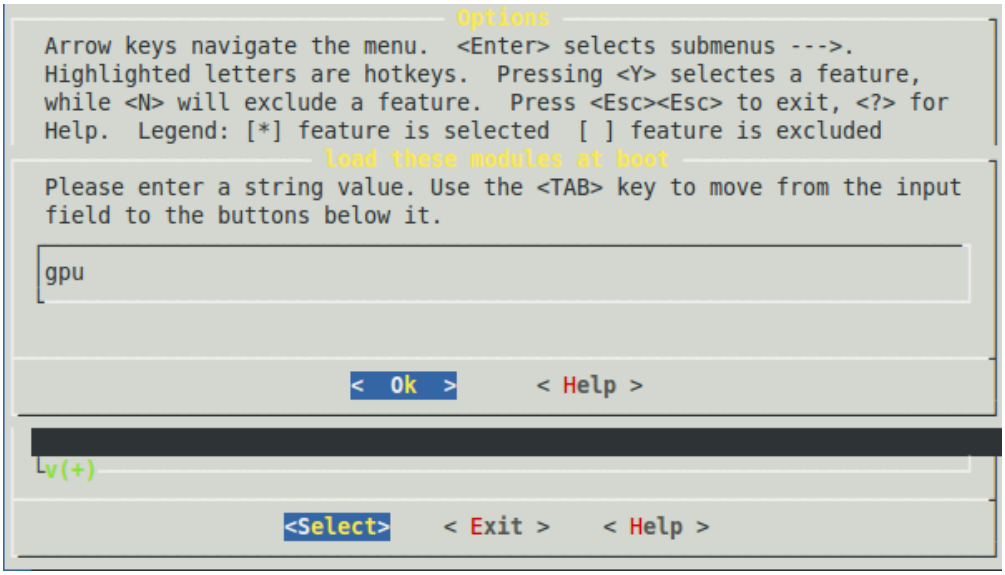

Figure 7: Using LTIB to configure preloaded modules

14. Scroll down to [ ] start networking, press space to select it.
15. Scroll down to ‘Network setup --->’ and press return
16. From the Network setup screen press Space to turn on [ ] get network parameters using chip

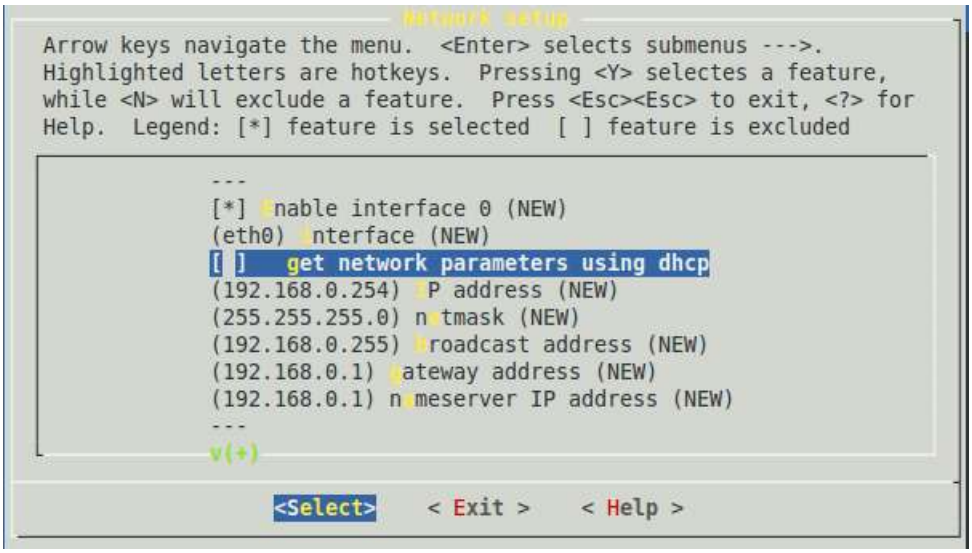

Figure 8: Using LTIB to turn on network parameters

17. Select Exit to return to previous screen.

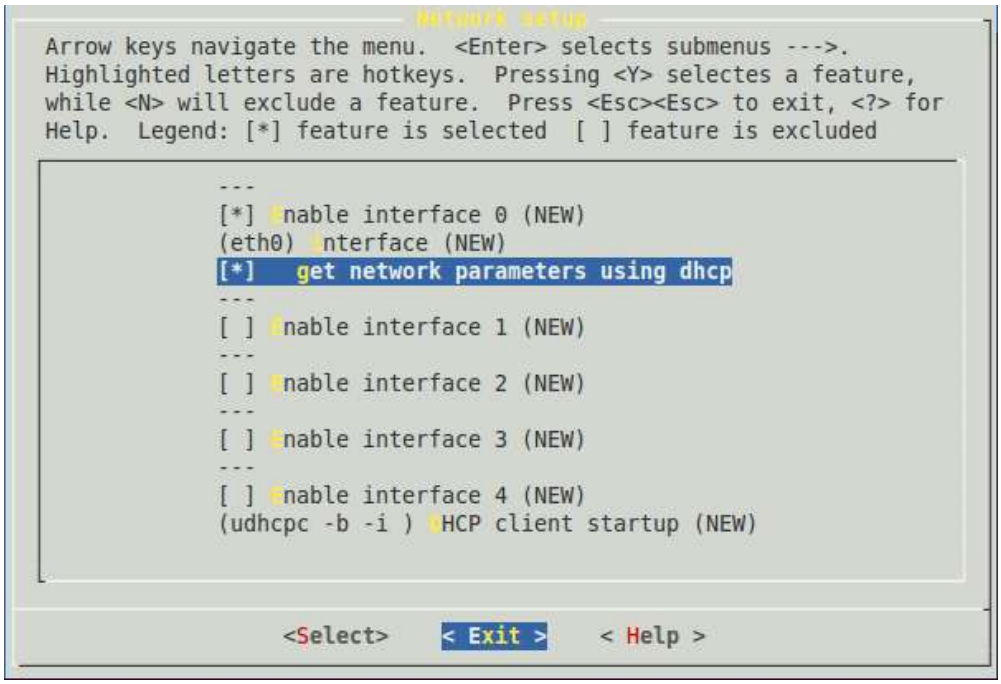

Figure 9: Network parameters set to On

18. Scroll down and turn off '[ ] start inetd'
19. Select Exit and press return
20. From the list select Target Image Generation Options --->
21. Scroll down to Target Image and press return, from the Target Image options scroll down to **ext2.gz ramdisk** and press space to select

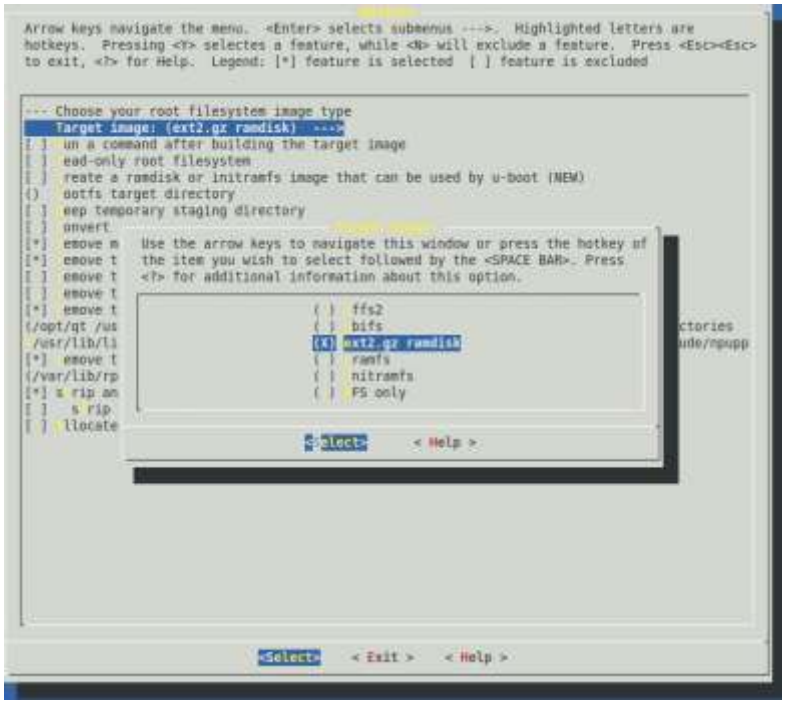

Figure 10: Using LTIB to set the root file system image type

22. Scroll down the list and use space to turn on:

```
[*] remove the /usr/src/ directory
[*] remove the /usr/include directory
```

23. Select Exit twice and then Save.
24. If prompted for the password enter **lucid**, LTIB will now build.

During the build process you will be prompted with a Linux Kernel Configuration box

25. Select General setup ---> From the submenu use space to turn on POSIX message queues

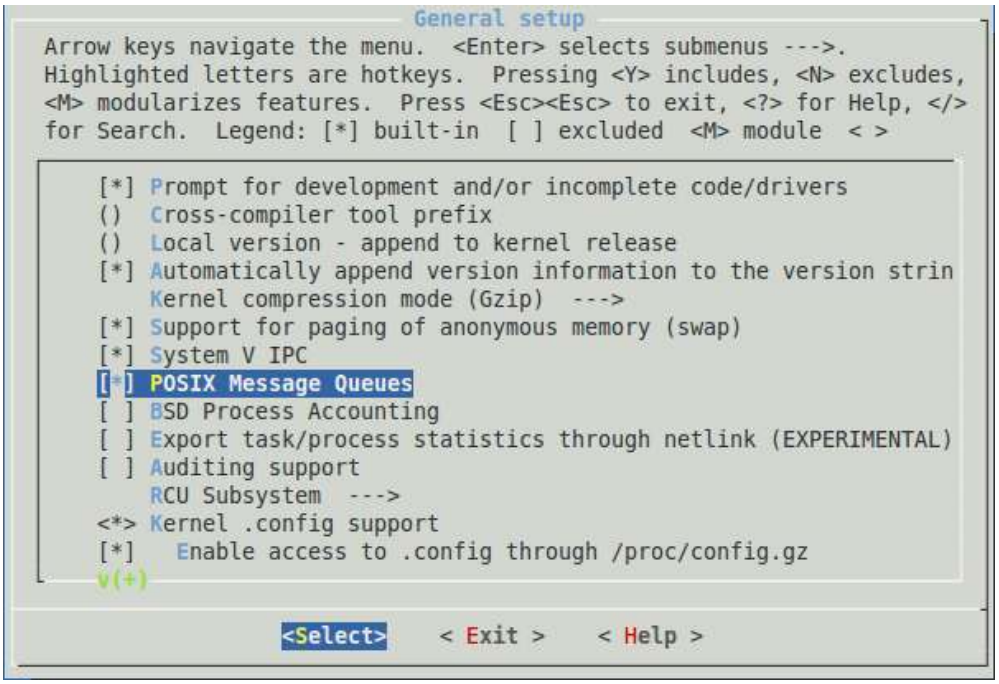

Figure 11: Using LTIB to configure POSIX message queues

26. Exit and Save, the build process will continue.

Once the LTIB build process is complete we can progress on to creating a bootable image from the rootfs output by LTIB.

Creating an booting SD card for iMX53QSB

This section of the document was tested against a 4 Gig micro SD card.

1. In the VM open a new terminal (Ctrl+Alt +t) window
2. Insert SD reader into USB port whilst the VM is running and has mouse focus (the reader will then attach to the VM).
3. Insert micro SD into SD card reader
4. In a terminal type *dmesg* to confirm that the SD card has been seen.

   You should see something like:

```
[10488.178601] scsi 4:0:0:0: Direct-Access     Generic- SD/MMC
1.00 PQ: 0 ANSI: 0 CCS
[10488.180506] sd 4:0:0:0: Attached scsi generic sg2 type 0
[10488.869077] sd 4:0:0:0: [sdb] 7744512 512-byte logical blocks: (3.96
GB/3.69 GiB)
[10488.878023] sd 4:0:0:0: [sdb] Write Protect is off
[10488.878068] sd 4:0:0:0: [sdb] Mode Sense: 03 00 00 00
[10488.878070] sd 4:0:0:0: [sdb] Assuming drive cache: write through
```

5. Unmount the card:

   sudo umount /dev/sdb?

6. Launch command line app fdisk (provides disk partitioning functions):

   sudo fdisk /dev/sdb

7. You will be prompted for Command (m for help), enter the below letters, pressing return after each:

   - *o* – Creates an empty partition table (ignore warning)

   - *n* – Create a new partition
   - *p* – Primary
   - *1* – Partition 1
   - <*enter*> – First Cylinder default start
   - *+512M* – Last Cylinder, Partition is 512M

   - *n* – Create a new partition
   - *p* – Primary
   - *2* – Partition 2
   - <*enter*> – First Cylinder default start
   - <*enter*> – Last Cylinder, to fill the rest of the disk

   - *d* – Delete partition
   - *1* – Partition 1

- *n* – Create a new partition
- *p* – Primary
- *1* – Partition 1
- *+32M* – First Cylinder, offset start by 32M to make space for u-boot and kernel
- <*enter*> – Last Cylinder, to fill the rest of the disk

- *t* – Change partition type
- *1* – Partition 1
- *c* – Set partition 1 to FAT

- *a* – Toggle the bootable
- *1* – Partition 1

- *w* – Write changes and exit

8. Change terminal directory to <ltib>, with rootfs in the CWD

   cd /home/lucid/ltib/
9. Erase the bootloader and environment from the SD card

   sudo dd if=/dev/zero of=/dev/sdb bs=512 seek=2 count=2048
10. Program the bootloader:

    sudo dd if=rootfs/boot/u-boot.bin of=/dev/sdb bs=512 skip=2 seek=2
11. Program the kernel:

    sudo dd if=rootfs/boot/uImage of=/dev/sdb bs=512 seek=2048
12. Format the FAT partition:

    sudo mkfs.vfat /dev/sdb1

The following steps need to be done each time the LTIB BSP rootfs is re-built:

13. Flash the ext2 partition (takes a couple of minutes usually):

    zcat rootfs.ext2.gz | sudo dd of=/dev/sdb2 bs=1M
14. Check and expand the ext2 partition (can take 15-20 minutes):

    sudo e2fsck /dev/sdb2

    sudo resize2fs /dev/sdb2

Mount the card by

15. Opening Places > Home Folder
16. Select 3.4 (or 3.5) GB File system (assuming a 4 Gig card is being used). This mounts the card again.

Installing the Inflexion runtime library to bootable image

This section explains how to deploy the Inflexion Runtime Engine and Inflexion-based applications to your target.

The Inflexion UI needs to be copied to the VM file system, before doing so a touchpanel fix needs to be applied to Inflexion CD install directory.

1. From this training workshop directory copy the file **…\Getting Started With Inflexion UI\iMX53QSB Patches\framework_linux_generic.c**

   To:

   …\InflexionUI-Runtime-2.3\Linux\inflexionui\framework\src

   Replacing the existing file.

The product can now be copied to the VM.

2. Create a new directory in the local VM file system to store the Inflexion UI Runtime and Demos
   e.g./home/lucid/iMX53_builds/Inflexion_2.3
3. From the product install directory copy the following folders to newly created folder above in the VM:

   …\InflexionUI-Runtime-2.3\Linux\demo

   …\InflexionUI-Runtime-2.3\Linux\inflexionui
4. Return to the <ltib> terminal window
5. Copy libinflexionui.so to <SD Card Linux partition>/usr/lib with the below instruction, **Note** below instruction is all one line

sudo cp /home/lucid/iMX53_builds/Inflexion_2.3/inflexionui/libs/armeabi/iMX5x/gles2/libinflexionui.so `mount | awk '/sdb2/ { print $3 }'`/usr/lib/

6. Modify the startup script to disable screen save and run Inflexion by editing the file <SD Card>/etc/rc.d/rc.local:

   sudo gedit `mount | awk '/sdb2/ { print $3 }'`/etc/rc.d/rc.local

   Add this to the bottom of the opened file:

   echo -e "\033c\033[9;0]\033[?33l\033[?25l" > /dev/tty0
   echo 0 > /sys/class/graphics/fb0/blank
   echo 1 > /sys/class/graphics/fb0/power/wakeup

7. Save and exit file.

Building and deploying the demo application

This section explains how to build and deploy the default Inflexion Demo.

1. In a terminal window Navigate to the inflexion demo directory:

   /iMX53_builds/Inflexion_2.3/demo/inflexion_demo

2. Execute the make command with additional make variables to set the root file system and cross compiler (all one command):

   make LTIB_ROOTFS=/home/lucid/ltib/rootfs CROSS_TOOLCHAIN=/opt/freescale/usr/local/gcc-4.4.4-glibc-2.11.1-multilib-1.0/arm-fsl-linux-gnueabi/bin/arm-none-linux-gnueabi-

3. Copy inflexionui_demo to <SD Card Linux partition>/usr/bin:

   sudo cp /home/lucid/iMX53_builds/Inflexion_2.3/demo/inflexion_demo/bin/inflexionui_demo `mount | awk '/sdb2/ { print $3 }'`/usr/bin/

4. Unmount the card:

   sudo umount /dev/sdb?

   sync

5. You can now remove the card and put it in the i.MX53 QSB ready to boot.

On first boot, the bootloader is not configured and the serial terminal will automatically put you into the “U-Boot” environment, so the below steps must be followed

U-Boot setup

The first time the SD card is created, the u-boot parameters will need configuring:

1. Connect the board to your host PC via serial cable.
2. To see serial output open TeraTerm or HyperTerminal on the host PC (not VM) (<Port>, 115200, 8, n, 1, no Flow Control)
3. Connect the power cable to the board and press the power button.
4. The following requires configuring through the serial terminal (copy and paste one line at a time to avoid serial problems):

setenv bootdelay 1

setenv bootcmd 'run bootcmd_SD1 bootcmd_SD2'

setenv bootcmd_SD1 'run bootargs_base bootargs_android bootargs_SD'

setenv bootargs_SD 'setenv bootargs ${bootargs}'

setenv bootcmd_SD2 'mmc read 0 ${loadaddr} 0x800 0x2000; bootm ${loadaddr}'

setenv bootargs_android 'setenv bootargs ${bootargs} init=/init root=/dev/mmcblk0p2 rootwait'

setenv bootargs_base 'setenv bootargs console=ttymxc0,115200 di0_primary video=mxcdi0fb:RGB24,SEIKO-WVGA,bpp=32 gpu_memory=256M calibration'

setenv loadaddr 0x70800000

setenv rd_loadaddr 0x70C00000

saveenv

boot

This last command will start the board booting.

1. Once booted, login via serial terminal as “root” with no password (press enter)
2. Set the root password (one time step):

   passwd
   root
   root

Note: ignore bad password warning

Running the demo application on target

The default Inflexion demo can now be executed

1. In the serial terminal type:

/usr/bin/inflexionui_demo

Note to stop the demo in the serial window use *Ctrl+c*, when deploying an updated demo.