AXIOMTEK

Q7M120-120-EVK
(Q7M120 and Q7B120)

BSP

Linux User's Manual

RoHS

Disclaimers

This manual has been carefully checked and believed to contain accurate information. Axiomtek Co., Ltd. assumes no responsibility for any infringements of patents or any third party's rights, and any liability arising from such use.

Axiomtek does not warrant or assume any legal liability or responsibility for the accuracy, completeness or usefulness of any information in this document. Axiomtek does not make any commitment to update the information in this manual.

Axiomtek reserves the right to change or revise this document and/or product at any time without notice.

No part of this document may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior written permission of Axiomtek Co., Ltd.

CAUTION

If you replace wrong batteries, it causes the danger of explosion. It is recommended by the manufacturer that you follow the manufacturer's instructions to only replace the same or equivalent type of battery, and dispose of used ones.

Table of Contents

This page is intentionally left blank.

Chapter 1
Introduction

The Q7M120 Linux BSP is a collection of binary, kernel patch files, and support files that can be used to create Linux kernel image, and a root file system for Q7M120 development systems.

1.1   Boot Loader

The Q7M120 Linux BSP contains the following U-Boot bootloader binary:
Q7M120 BSP\Image\u-boot.bin

1.2   Linux Kernel Image

This Q7M120 Linux BSP contains a pre-built kernel image based on the 3.0.35 version of the Linux kernel with Q7M120 hardware patch. The Q7M120 kernel image is located at the following path:
Q7M120 BSP\Image\uImage

1.3   Gnome Mobile Root File System

The root file system package provides busybox, common libraries, and other fundamental elements.
The Q7M120 Linux BSP package contains the following rootfs file system:
Q7M120 BSP\Image\rootfs.tar.bz2

The rootfs.tar.bz2 file system includes Freescale specific libraries and gnome GUI. It can be mounted as NFS, or its contents can be stored on a boot media such as SD card or eMMC.

To login, please enter 'root' without password.

1.4   How to boot the Q7M120 EVK board
The boot modes of the Q7M120 EVK board are controlled by the boot configuration DIP switches on the board. To locate the boot configuration switches, see Hardware SPEC manual

Getting started
Connecting the Q7M120 through serial RS-232 console port to see the board's boot messages or connect a monitor via HDMI port
The serial console port is a convenient interface for connecting Q7M120 to PC. First of all, it is very important to make sure that the serial port settings are as follows:
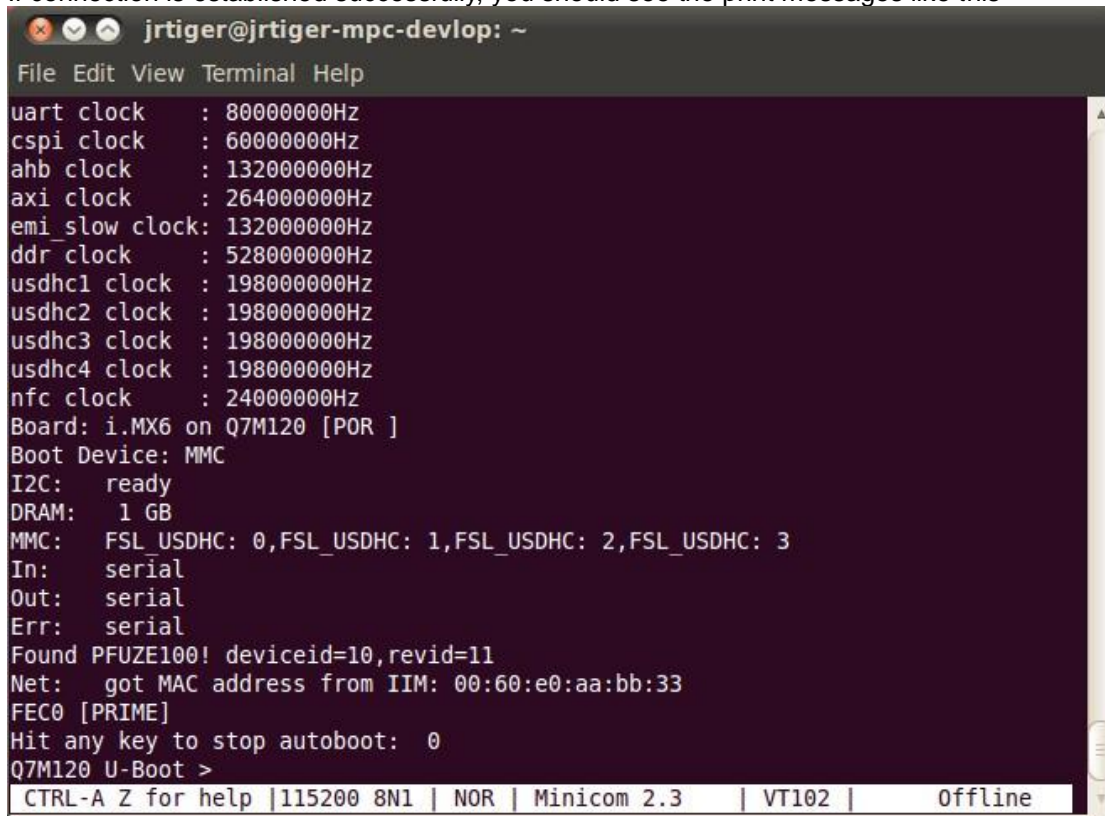
Baudrate: 115200 bps
Parity: None
Data bits: 8
Stop bit: 1
Flow Control: None

If connection is established successfully, you should see the print messages like this

```
⊗ ⊘ ⊙   jrtiger@jrtiger-mpc-devlop: ~

File  Edit  View  Terminal  Help
uart clock      : 80000000Hz
cspi clock      : 60000000Hz
ahb clock       : 132000000Hz
axi clock       : 264000000Hz
emi_slow clock: 132000000Hz
ddr clock       : 528000000Hz
usdhc1 clock  : 198000000Hz
usdhc2 clock  : 198000000Hz
usdhc3 clock  : 198000000Hz
usdhc4 clock  : 198000000Hz
nfc clock       : 24000000Hz
Board: i.MX6 on Q7M120 [POR ]
Boot Device: MMC
I2C:    ready
DRAM:   1 GB
MMC:    FSL_USDHC: 0,FSL_USDHC: 1,FSL_USDHC: 2,FSL_USDHC: 3
In:     serial
Out:    serial
Err:    serial
Found PFUZE100! deviceid=10,revid=11
Net:    got MAC address from IIM: 00:60:e0:aa:bb:33
FEC0 [PRIME]
Hit any key to stop autoboot:  0
Q7M120 U-Boot >
 CTRL-A Z for help |115200 8N1 |  NOR | Minicom 2.3    | VT102 |    Offline
```

To login, please enter 'root' (with no password)

```
freescale login: root
login[3255]: root login on 'ttymxc0'
root@freescale ~$
```

Running the image on the target
This chapter explains how to run an image on the target from downloaded device and NFS.

Running the Image from NFS
To boot from NFS, do as follows:
1. Power on the board.
2. Enter the following commands in the U-Boot prompt:

Q7M120 U-Boot > setenv serverip 10.1.35.155
Q7M120 U-Boot > setenv ipaddr 10.1.35.11
Q7M120 U-Boot > run bootcmd_net

```
😕 😊 😊   jrtiger@jrtiger-mpc-devlop: ~

File  Edit  View  Terminal  Help
emi_slow clock: 132000000Hz
ddr clock        : 528000000Hz
usdhc1 clock     : 198000000Hz
usdhc2 clock     : 198000000Hz
usdhc3 clock     : 198000000Hz
usdhc4 clock     : 198000000Hz
nfc clock        : 24000000Hz
Board: i.MX6 on Q7M120 [WDOG ]
Boot Device: MMC
I2C:   ready
DRAM:   1 GB
MMC:   FSL_USDHC: 0,FSL_USDHC: 1,FSL_USDHC: 2,FSL_USDHC: 3
*** Warning - bad CRC or MMC, using default environment

In:    serial
Out:   serial
Err:   serial
Found PFUZE100! deviceid=10,revid=11
Net:   got MAC address from IIM: 00:60:e0:aa:bb:33
FEC0 [PRIME]
Hit any key to stop autoboot:  0
Q7M120 U-Boot > setenv serverip 10.1.35.155
Q7M120 U-Boot > setenv ipaddr 10.1.35.11
Q7M120 U-Boot > run bootcmd_net
 CTRL-A Z for help |115200 8N1 | NOR | Minicom 2.3    | VT102 |    Offline
```

NOTE: If the MAC address has not burned into fuse, you must set the MAC address to use network in U-Boot.
Q7M120 U-Boot > setenv ethaddr xx:xx:xx:xx:xx:xx

Running the Image from eMMC (Q7M120 default)
```
FEC0 [PRIME]
Hit any key to stop autoboot:  0
Q7M120 U-Boot > run bootcmd_mmc_hdmi
```

Running the Image from SD Card
Q7M120 U-Boot > setenv bootargs_sd 'setenv bootargs ${bootargs} ip=dhcp root=/dev/mmcblk1p1 rootwait'
Q7M120 U-Boot > setenv bootcmd_sd_hdmi 'run bootargs_hdmi bootargs_sd; mmc dev 2; mmc read ${loadaddr} 0x800 0x3000; bootm'
Q7M120 U-Boot > setenv bootargs_hdmi 'setenv bootargs console=ttymxc0,115200 fbmem=24M video=mxcfb0:dev=hdmi,1920x1080M@60,bpp=32 video=mxcfb1:off video=mxcfb2:off'
Q7M120 U-Boot > saveenv
Q7M120 U-Boot > run bootcmd_sd_hdmi

This page is intentionally left blank.

Chapter 2
Board Support Package (BSP)

2.1   Host Development System Installation

2.1.1   Install Host System

1.   Download Ubuntu 10.04 LTS iso image.

2.   Install Ubuntu 10.04.

3.   Sudoers:
     To edit the sudoer's file, please run 'sudo visudo'. At the end of the sudoers file, add the following line; which is needed for using LTIB. This assumes that all your developers have administrator privileges on this host. If that is not the case, add a similar line for each user.

4.   Install host packages needed by LTIB as follows:
     $ sudo aptitude -y install gettext libgtk2.0-dev rpm bison m4 libfreetype6-dev
     $ sudo aptitude -y install libdbus-glib-1-dev liborbit2-dev intltool
     $ sudo aptitude -y install ccache ncurses-dev zlib1g zlib1g-dev gcc g++ libtool
     $ sudo aptitude -y install uuid-dev liblzo2-dev
     $ sudo aptitude -y install tcl dpkg
     $ sudo aptitude -y install asciidoc texlive-latex-base dblatex xutils-dev
     $ sudo aptitude -y install git-core git-doc git-email git-gui gitk

5. Install and configure TFTP server:
    After tftpd is installed, configure it by editing /etc/xinetd.d/tftp. Change the default export path (it is either /usr/var/tftpboot or /var/lib/tftpboot) to / . Or change the default export path to whatever directory you want to download from. Then reboot the hardware.
    $ sudo aptitude -y install tftp tftpd xinetd
    $ sudo vi /etc/xinetd.d/tftp

    Then restart the TFTP server.
    $ sudo /etc/init.d/xinetd restart

6. Install and configure NFS server:
    $ sudo aptitude -y install nfs-common nfs-kernel-server portmap
    To configure nfs server, add lines to /etc/exports as follows:
    /tools/rootfs *(rw,sync,no_root_squash)
    $ sudo vi /etc/exports
    Create a symbolic link to root filesystem which your ltib build.
    $ sudo mkdir /tools
    $ sudo ln -s ~/Project/your_project/ltib/rootfs /tools/rootfs

    Then restart the NFS server.
    $ sudo /etc/init.d/nfs-kernel-server restart

2.1.2   Install LTIB

1.   Build your own project folder.
     $ mkdir -p ~/Project/your_project

2.   Extract the source tar ball with the following command:
     $ tar zxf L3.0.35_4.1.0_130816_source.tar.gz

     Note that you can get this file from Axiomtek official website:

          Download LTIB_IMX6.zip
          Unzip it to get L3.0.35_4.1.0_docs.tar.gz, IMX_AACPD_3.0.7-2.tar.gz,
          IMX_MMCODEC_3.0.35_4.1.0_BUNDLE_CODA.tar.gz and
          L3.0.35_4.1.0_130816_source.tar.gz.

3.   Now change directory to the extracted folder and execute the install script. Then
     you can install it into your own project folder.
     $ cd L3.0.35_4.1.0_130816_source
     $ ./install

4.   Configure and build.
     This command invokes LTIB with its default behavior of performing a build. Since
     the installation has not yet been configured, LTIB will present the configuration
     screen before building. Then, when you exit the configuration screen, LTIB will
     build the target image. Also, at the very first time you run LTIB on a system, it will
     build and install a number of host tools that it will place under /opt/freescale.

     To ask LTIB to show the configuration screen without building afterwards, run this
     command:
     $ ./ltib -m config

5. First time configuration; please select the iMX platform.
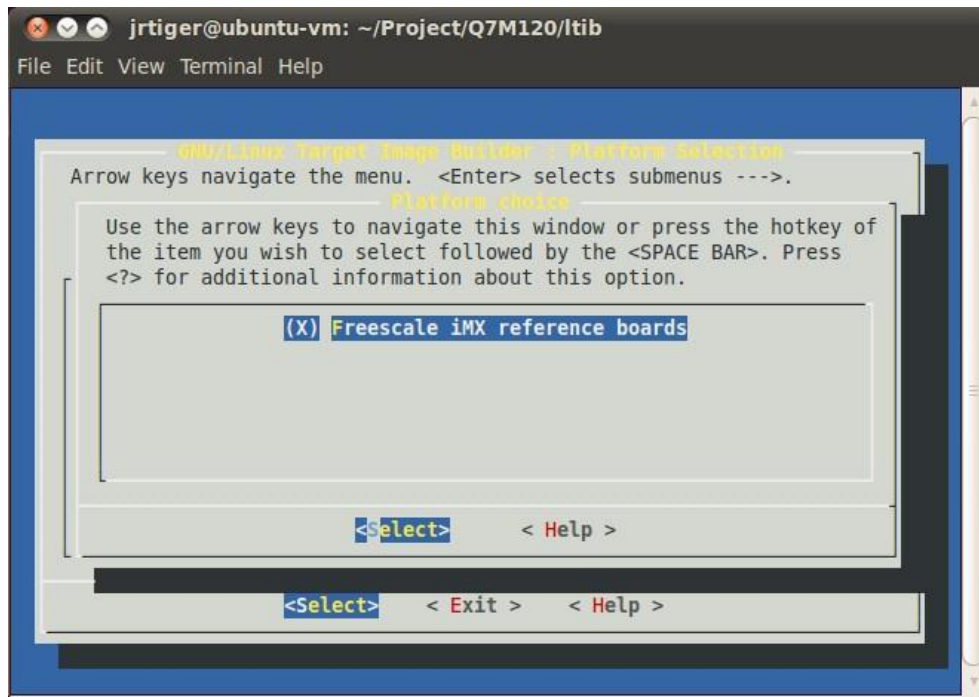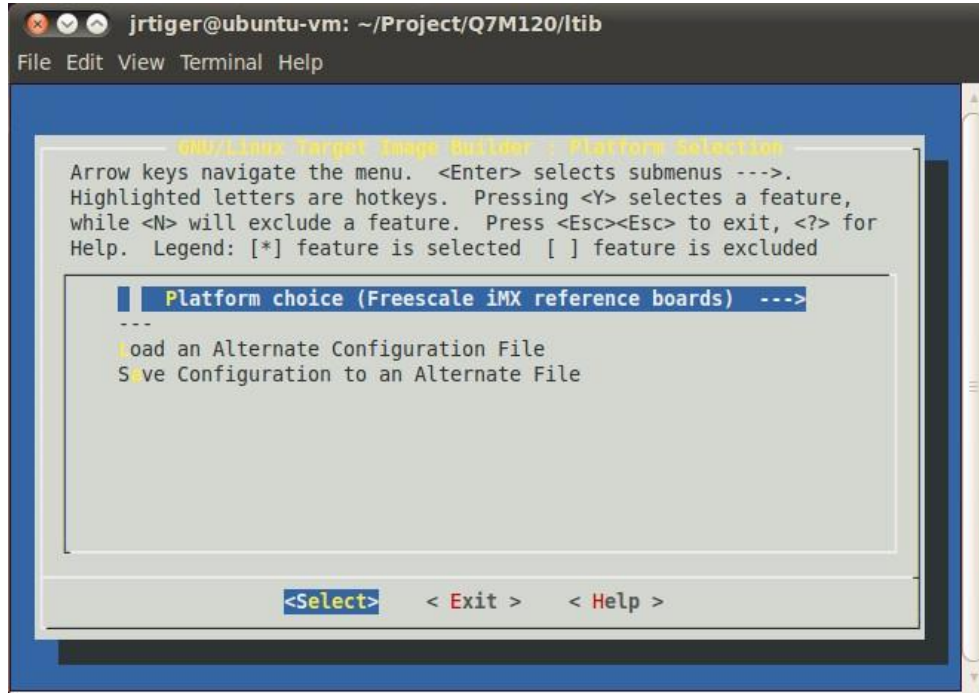
Then save the configuration.
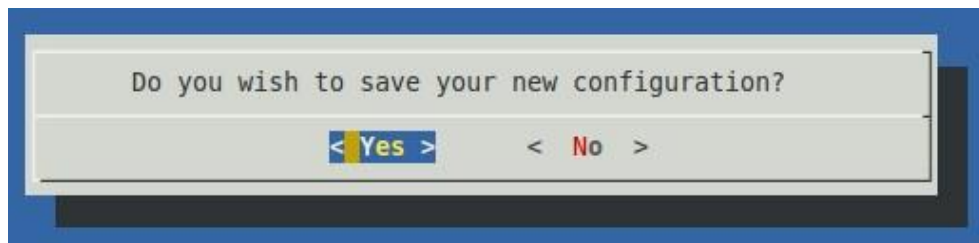
Select the imx6q sub-platform.
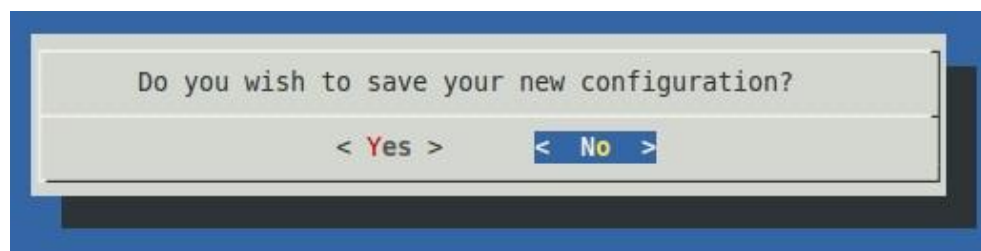
Select packages profile according to your requirement.

Exit the sub-platform selection screen and save the configuration. This will bring you to the main LTIB configuration screen.
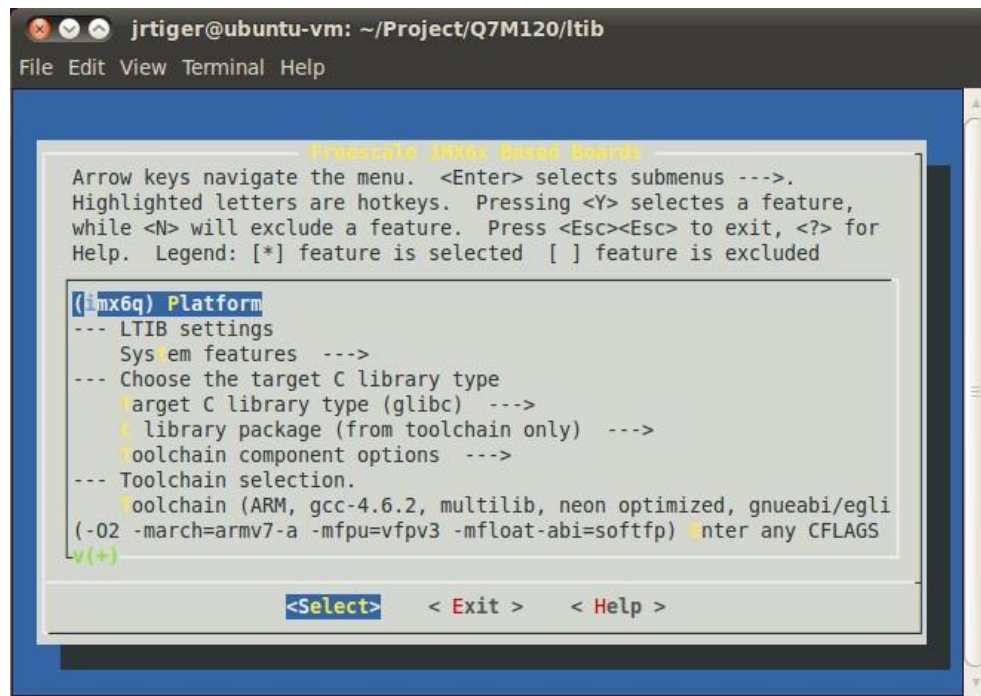
Just exit without saving main LTIB configuration.

6. Create and patch kernel for Q7M120.
    Create the local kernel folder.
    $ mkdir -p ~/Project/Q7M120/kernel

    Extract kernel source to this folder.
    $ tar jxf /opt/freescale/pkgs/linux-3.0.35.tar.bz2 -C ~/Project/Q7M120/kernel

    Extract i.MX platform patches to kernel source folder.
    $ tar jxf /opt/freescale/pkgs/linux-3.0.35-imx_4.1.0.bz2 -C
    ~/Project/Q7M120/kernel/linux-3.0.35

    Enter the kernel source folder and patch the patches.
    $ cd ~/Project/Q7M120/kernel/linux-3.0.35/
    $ ./patches/patch-kernel.sh

    Patch the Q7M120 patches which you can copy from our BSP package.
    $ tar zxf Q7M120_patches_20140401.tar.gz -C
    ~/Project/Q7M120/kernel/linux-3.0.35
    $ cd ~/Project/Q7M120/kernel/linux-3.0.35/
    $ ./patches-q7m120/patch-q7m120.sh

Return to ~/Project/Q7M120/kernel folder and make a symbolic link to kernel source folder.
$ cd ~/Project/Q7M120/kernel
$ ln -s linux-3.0.35 linux

7. Reconfigure LTIB main setting to build kernel to local Linux directory.
$ cd ~/Project/Q7M120/ltib
$ ./ltib -m config

Select kernel setting.
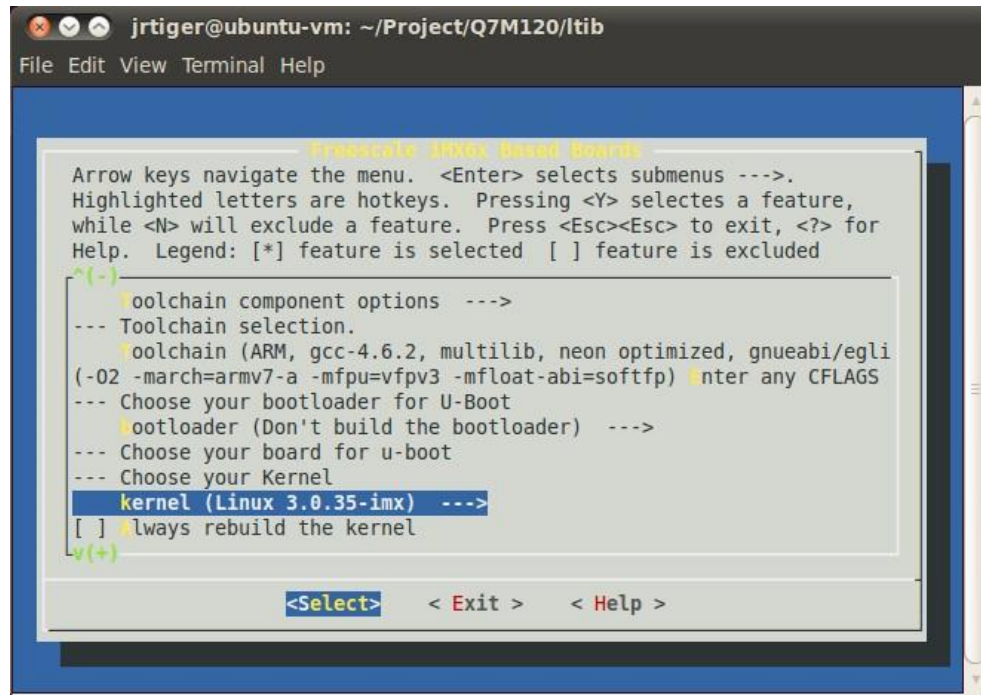
Select Local Linux directory build.
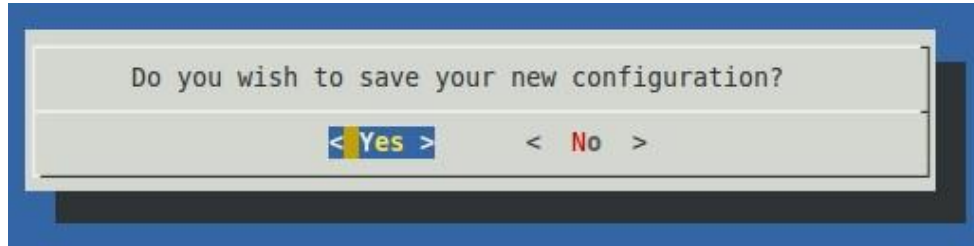

Then enter new linux source directory.

Enter your own linux kernel source folder.
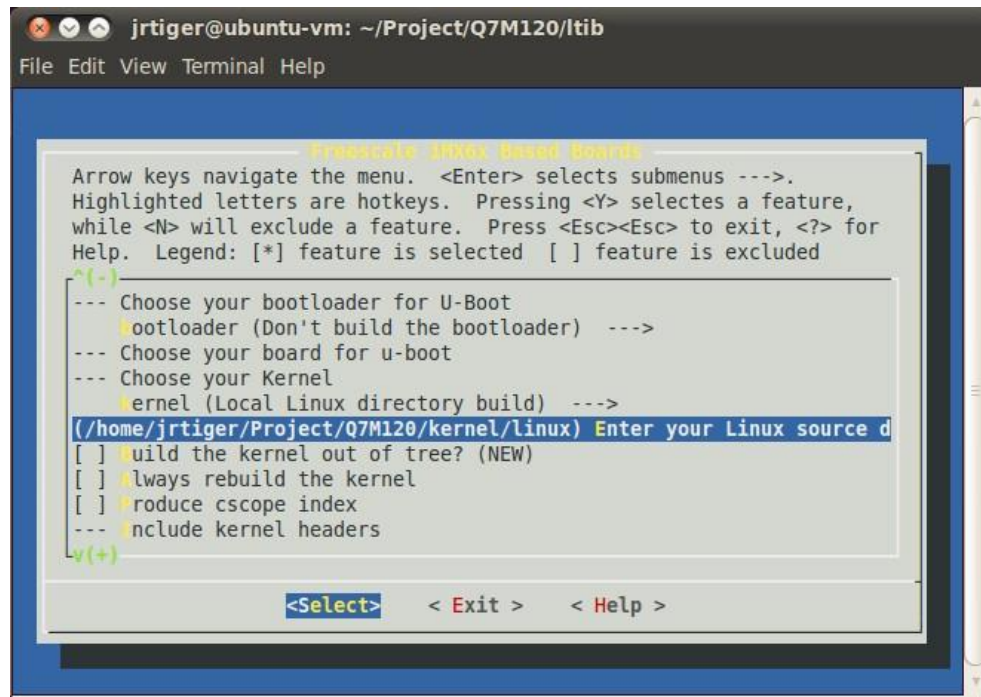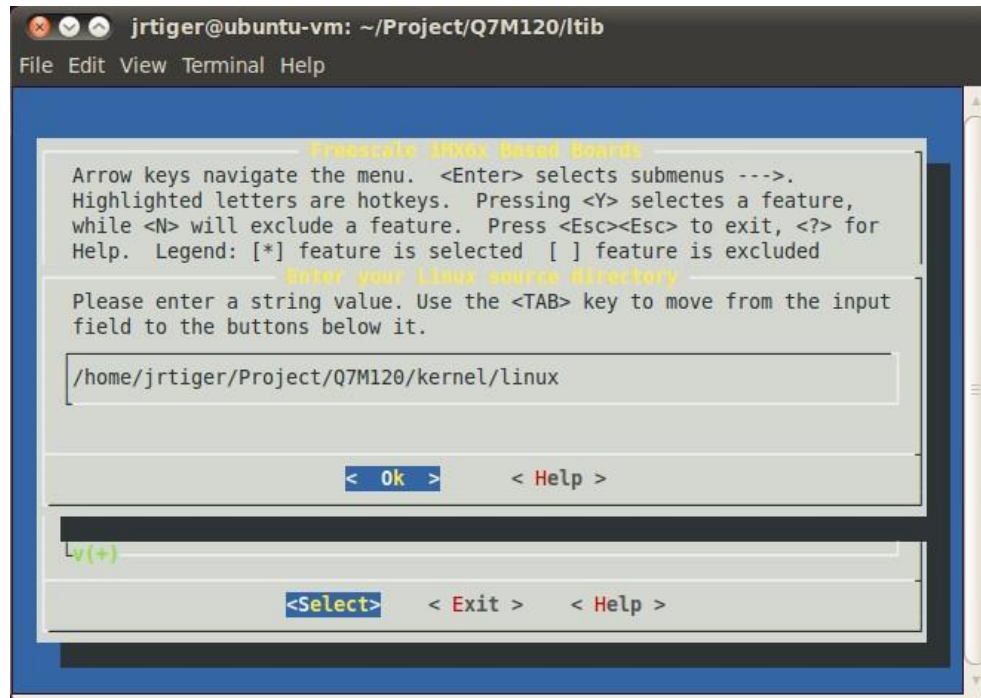
Finally save the configuration.

8. Options:
   Adding iMX6 Multimedia codecs support:
   Note that you can get this file from Axiomtek official website:

   Download LTIB_IMX6.zip
   Unzip it to get L3.0.35_4.1.0_docs.tar.gz, IMX_AACPD_3.0.7-2.tar.gz,
   IMX_MMCODEC_3.0.35_4.1.0_BUNDLE_CODA.tar.gz and
   L3.0.35_4.1.0_130816_source.tar.gz.

   Extract codec file.
   $ tar zxf IMX_MMCODEC_3.0.35_4.1.0_BUNDLE_CODA.tar.gz
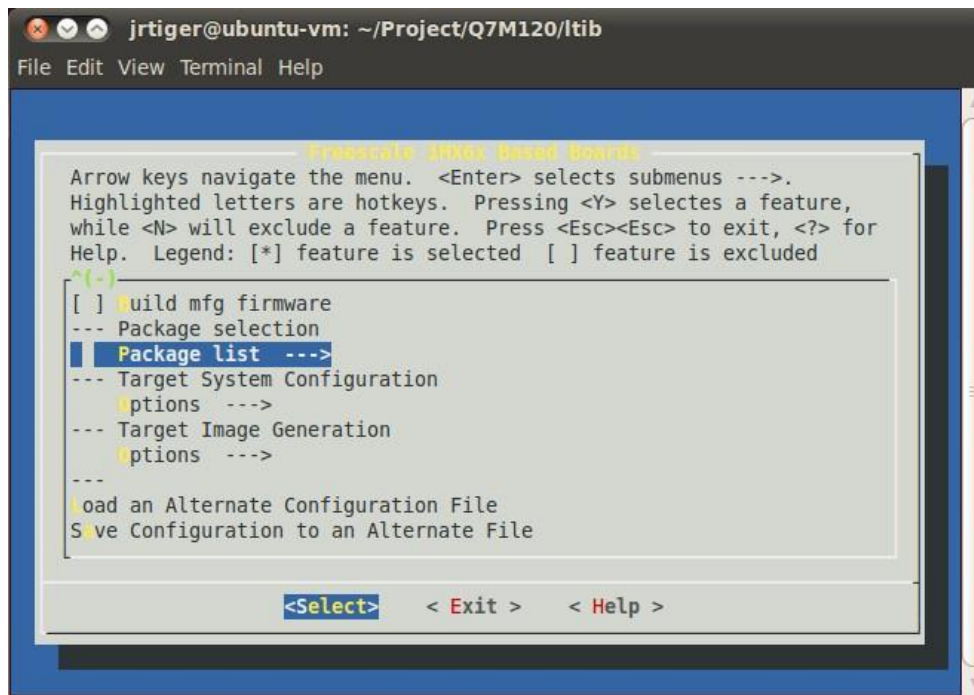 $ tar zxf IMX_AACPD_3.0.7-2.tar.gz
   Copy all tar.gz file from ltib codecs folder to /opt/freescale/pkgs folder.
   $ cp IMX_MMCODEC_3.0.35_4.1.0_BUNDLE_CODA/ltib\ codecs/*.tar.gz
   /opt/freescale/pkgs/
   $ cp IMX_AACPD_3.0.7-2/LTIB\ Codecs/libfslaacpcodec-3.0.7-2.tar.gz
   /opt/freescale/pkgs/
   Then reconfigure LTIB main setting.
   $ cd ~/Project/Q7M120/ltib
   $ ./ltib -m config

   Select Package list.

Select Freescale Multimedia Plugins/Codecs.

Select codecs you want to add.

9. Build the target kernel image and root filesystem.
$ cd ~/Project/Q7M120/ltib
$ ./ltib

After you have completed a build using LTIB, you will have a target root filesystem in the rootfs directory inside the LTIB install directory.

Inside the ~/Project/Q7M120/ltib/rootfs/boot directory, you can find kernel image uImage.

2.2 Deployment

2.2.1 Create a bootable SD Card for Q7M120

1. Copying the Boot Loader image.
    `$ sudo dd if=./u-boot.bin of=/dev/sdb bs=512 seek=2 skip=2 conv=fsync`
   The first 1KB of the SD card, that includes the partition table, will be preserved

2. Copying the Kernel image.
    `$ sudo dd if=./uImage of=/dev/sdb bs=512 seek=2048 conv=fsync`
   This will copy uImage to the media at offset 1MB

3. Copying the Root File System (rootfs).
   First, a partition table must be created.

   To create a partition, at offset 16384 (in sectors of 512 bytes) enter the following command:
    `$ sudo fdisk /dev/sdb`

   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]
   16384   [starting at offset sector #16384, i.e. 8MB, 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]

   Run the following command to format the partition
    `$ sudo mkfs.ext3 /dev/sdb1`
   Copy the target file system to the partition
    `$ mkdir /home/user/mountpoint`
    `$ sudo mount /dev/sdb1 /home/usrt/mountpoint`
   Extract rootfs package to certain directory
    `$ sudo tar jxf rootfs.tar.bz2 -C /home/user/mountpoint`

   NOTE: Make sure tht the device node is correct for the SD Card. i.e. sdb or sdc

   The file system content is now on the media.