



®

AXIOMTEK

SCM120-120-EVK
(SCM120 and SCB120)

BSP

Linux User's Manual



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.

©Copyright 2016 Axiomtek Co., Ltd.

All Rights Reserved

June 2016, Version A2

Printed in Taiwan

Table of Contents

Disclaimers.....	ii
Chapter 1 Introduction.....	1
1.1 Boot Loader	1
1.2 Linux Kernel Image	1
1.3 Gnome Mobile Root File System	1
1.4 Getting Started with SCM120 EVK board	1
Chapter 2 How to Boot the SCM120 EVK Board	3
2.1 How to Boot from SD Card / eMMC.....	3
Chapter 3 Running the Image on the Target	5
3.1 Running the image from NFS.....	5
3.2 Running the image from eMMC (SCM120 default).....	6
3.3 Running the image from SD Card.....	6
Chapter 4 Board Support Package (BSP)	7
4.1 Host Development System Installation	7
4.1.1 Install Host System.....	7
4.1.2 Install LTIB.....	9
4.2 Deployment.....	20
4.2.1 Create a bootable SD Card for SCM120.....	20

This page is intentionally left blank.

Chapter 1

Introduction

The SCM120 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 SCM120 development systems.

1.1 Boot Loader

The SCM120 Linux BSP contains the following U-Boot bootloader binary:
SCM120 BSP\Image\u-boot-scm120q.bin

1.2 Linux Kernel Image

This SCM120 Linux BSP contains a pre-built kernel image and modules based on the 3.0.35 version of the Linux kernel with SCM120 hardware patch. The SCM120 kernel image is located at the following path:
SCM120 BSP\Image\ulImage
And SCM120 kernel module is
SCM120 BSP\Image\modules.tgz

1.3 Gnome Mobile Root File System

The root file system package provides busybox, common libraries, and other fundamental elements.

The SCM120 Linux BSP package contains the following rootfs file system:
SCM120 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.

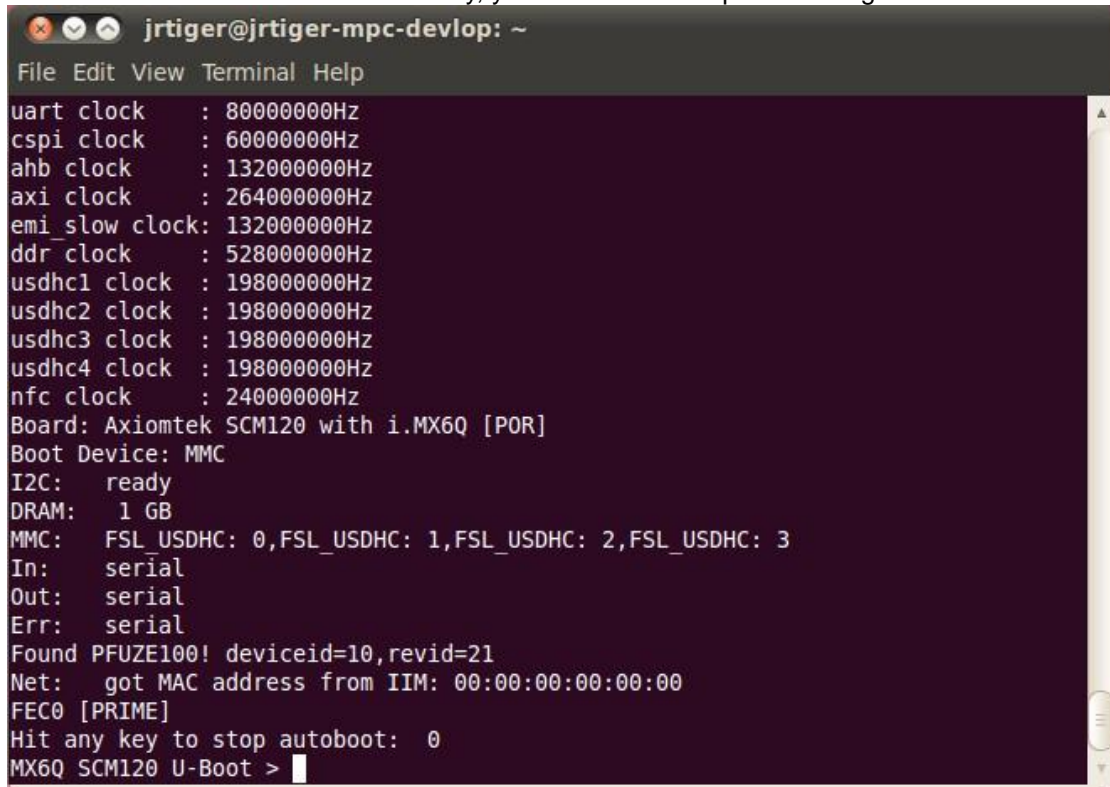
1.4 Getting Started with SCM120 EVK board

Connecting the SCM120 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 SCM120 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-develop: ~
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: Axiomtek SCM120 with i.MX6Q [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=21
Net: got MAC address from IIM: 00:00:00:00:00:00
FEC0 [PRIME]
Hit any key to stop autoboot: 0
MX6Q SCM120 U-Boot >
```

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



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

Chapter 2

How to Boot the SCM120 EVK Board

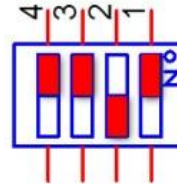
The boot modes of the SCM120 EVK board are controlled by the boot configuration DIP switches and jump setting on the board. To locate the boot configuration switches, see Hardware SPEC manual

2.1 How to Boot from SD Card / eMMC

The following table shows the jump setting for SD/ eMMC boot

SCM120 jumpers are for boot mode selection

Function	Setting
Depend on base board (Default)	SW1(1,3,4) ON SW1(2) OFF



SCB120 jumpers are for boot mode selection.

Function	Setting
Serial Downloader Mode (For MFGTool)	JP1 Open JP2 Open JP3 Open
Boot from eMMC (Default)	JP1 Close JP2 Open JP3 Open
Boot from SD Card	JP1 Open JP2 Close JP3 Close

This page is intentionally left blank.

Chapter 3

Running the Image on the Target

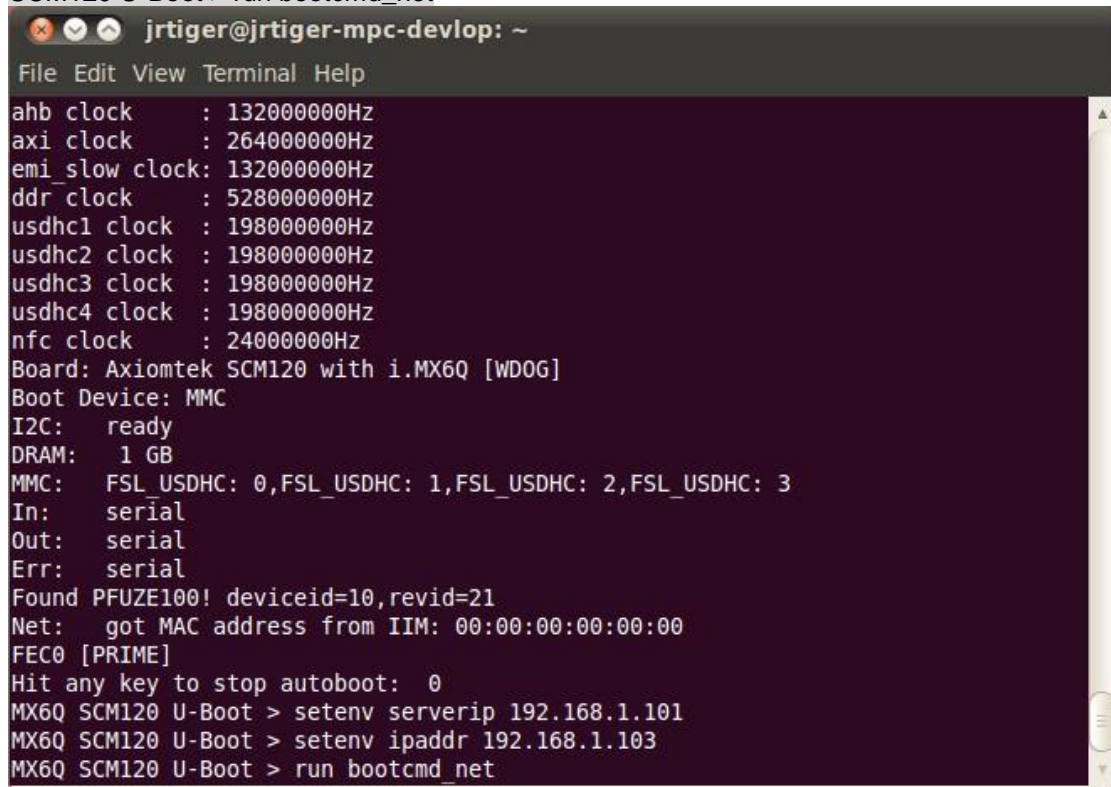
This chapter explains how to run an image on the target from downloaded device and NFS.

3.1 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:

```
SCM120 U-Boot > setenv serverip 192.168.1.101
SCM120 U-Boot > setenv ipaddr 192.168.1.103
SCM120 U-Boot > run bootcmd_net
```



```
jrtiger@jrtiger-mpc-develop: ~
File Edit View Terminal Help
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      : 240000000Hz
Board: Axiomtek SCM120 with i.MX6Q [WDOG]
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=21
Net:   got MAC address from IIM: 00:00:00:00:00:00
FEC0 [PRIME]
Hit any key to stop autoboot:  0
MX6Q SCM120 U-Boot > setenv serverip 192.168.1.101
MX6Q SCM120 U-Boot > setenv ipaddr 192.168.1.103
MX6Q SCM120 U-Boot > run bootcmd_net
```

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

```
SCM120 U-Boot > setenv ethaddr xx:xx:xx:xx:xx:xx
```

3.2 Running the image from eMMC (SCM120 default)

```
SCM120 U-Boot > run bootcmd_emmc  
FEC0 [PRIME]  
Hit any key to stop autoboot: 0  
MX6Q SCM120 U-Boot > run bootcmd emmc
```

3.3 Running the image from SD Card

```
SCM120 U-Boot > run bootcmd_sd  
FEC0 [PRIME]  
Hit any key to stop autoboot: 0  
MX6Q SCM120 U-Boot > run bootcmd sd
```

Tips :

We recommend you to use SD card or NFS boot to develop your applications, and keep eMMC's system when you need debug and speed up development

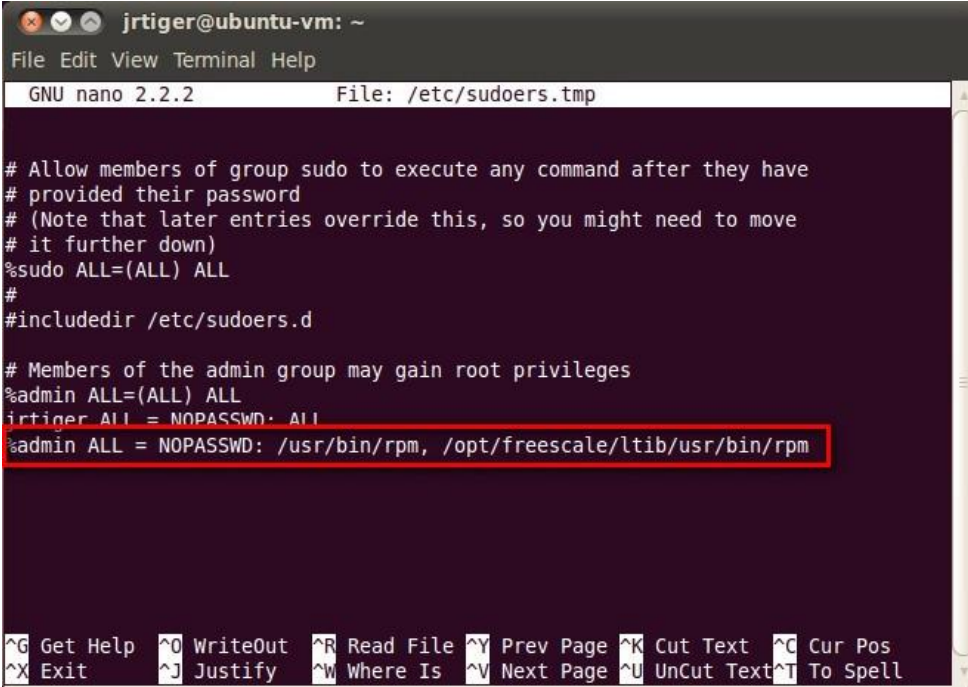
Chapter 4

Board Support Package (BSP)

4.1 Host Development System Installation

4.1.1 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.



```

jrtiger@ubuntu-vm: ~
File Edit View Terminal Help
GNU nano 2.2.2 File: /etc/sudoers.tmp

# Allow members of group sudo to execute any command after they have
# provided their password
# (Note that later entries override this, so you might need to move
# it further down)
%sudo ALL=(ALL) ALL
#
#includedir /etc/sudoers.d

# Members of the admin group may gain root privileges
%admin ALL=(ALL) ALL
jrtiger ALL = NOPASSWD: ALL
%admin ALL = NOPASSWD: /usr/bin/rpm, /opt/freescale/ltib/usr/bin/rpm

^G Get Help ^O WriteOut ^R Read File ^Y Prev Page ^K Cut Text ^C Cur Pos
^X Exit ^J Justify ^W Where Is ^V Next Page ^U UnCut Text ^T To Spell

```

4. Install host packages needed by LTIB as follows:

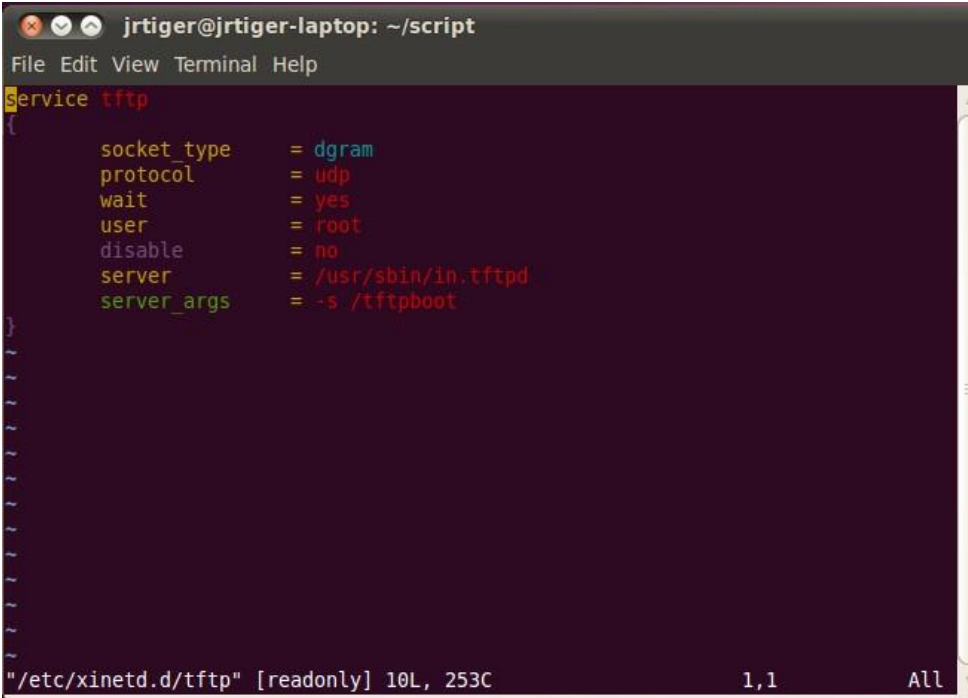

```

$ 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 dlatex 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 tftpd xinetd
$ sudo vi /etc/xinetd.d/tftp
```



```
jrtiger@jrtiger-laptop: ~/script
File Edit View Terminal Help
Service tftp
{
    socket_type      = dgram
    protocol        = udp
    wait            = yes
    user            = root
    disable         = no
    server          = /usr/sbin/in.tftpd
    server_args     = -s /tftpboot
}

"/etc/xinetd.d/tftp" [readonly] 10L, 253C      1,1      All
```

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
```

4.1.2 Install LTIB

1. Build your own project folder.
\$ mkdir -p ~/Project/your_project
2. Extract the source tar ball with the following command:
\$ tar xzf 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

```

jrtiger@ubuntu-vm: ~/Project/Q7M120/source/LTIB/L3.0.35_4.1.0_130816_sc
File Edit View Terminal Help
accompanying written documentation (the "Licensed Software"). In consideration
for Freescale allowing you to access the Licensed Software, you are agreeing to
be bound by the terms of this Agreement. If you do not agree to all of the terms
of this Agreement, do not download the Licensed Software. If at any point you no
longer agree to all the terms of this Agreement, stop using the Licensed
Software immediately and delete all copies of the Licensed Software in your
possession or control. Any copies of the Licensed Software that you have already
distributed, where permitted, and that have not been destroyed, will continue to
be governed by this Agreement. Your prior use of the Licensed Software will also
continue to be governed by this Agreement.

Section 1. Definitions

I have read and accept the EULA (yes|no):
yes

The LTIB files are extracted from a tar file which includes the
prefix ltib. After installation you will find LTIB in:
/home/jrtiger/Project/Q7M120/source/LTIB/L3.0.35_4.1.0_130816_source/ltib

Where do you want to install LTIB ? (/home/jrtiger/Project/Q7M120/source/LTIB/L3
.0.35_4.1.0_130816_source)
~/.Project/Q7M120

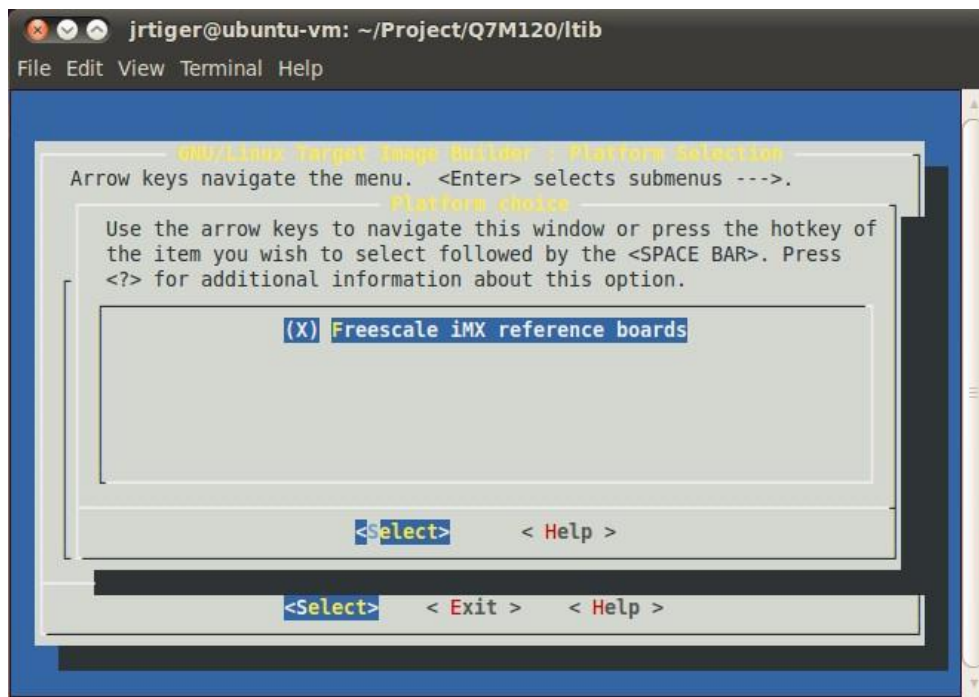
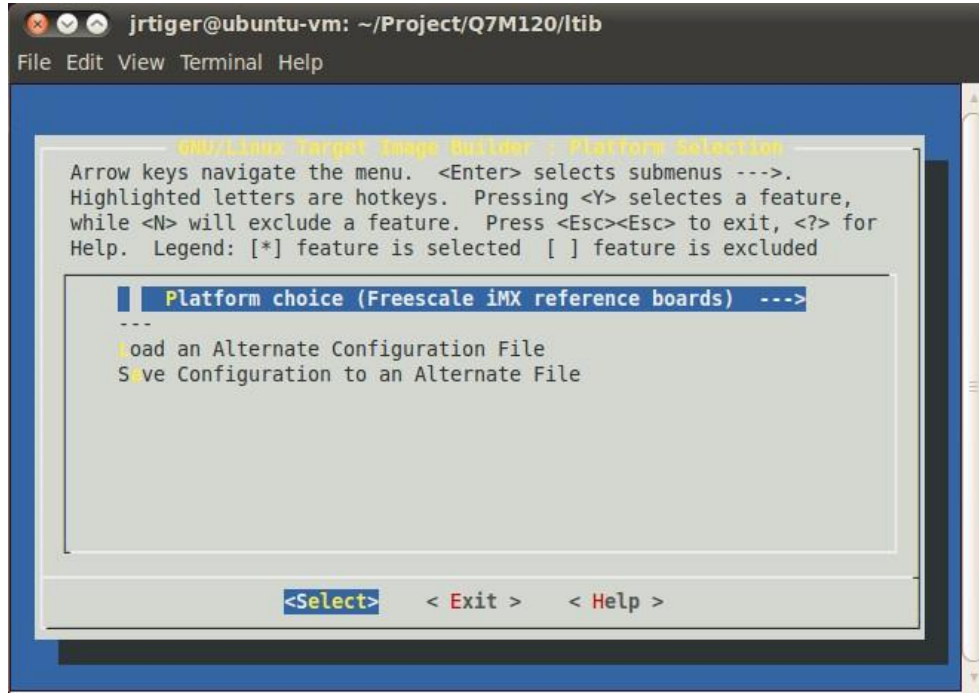
```

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.

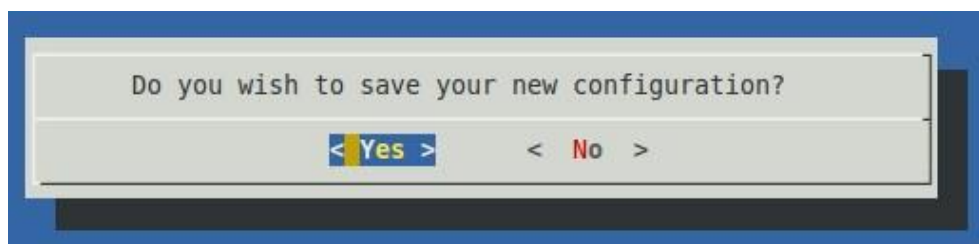
To ask LTIB to show the configuration screen without building afterwards, run this command:

```
$ ./ltib -m config
```

5. First time configuration; please select the iMX platform.



Then save the configuration.



Select the imx6q sub-platform.

```

jrtiger@ubuntu-vm: ~/Project/Q7M120/ltib
File Edit View Terminal Help

----- Freescale i.MX development platforms -----
Arrow keys navigate the menu. <Enter> selects submenus --->.
Selection
Use the arrow keys to navigate this window or press the hotkey of
the item you wish to select followed by the <SPACE BAR>. Press
<?> for additional information about this option.
^(-)
( ) mx31ads
( ) mx31_3stack
( ) mx35_3stack
( ) mx37_3stack
( ) mx5x
(X) imx6q
v(+)
<Select> < Help >
<Select> < Exit > < Help >

```

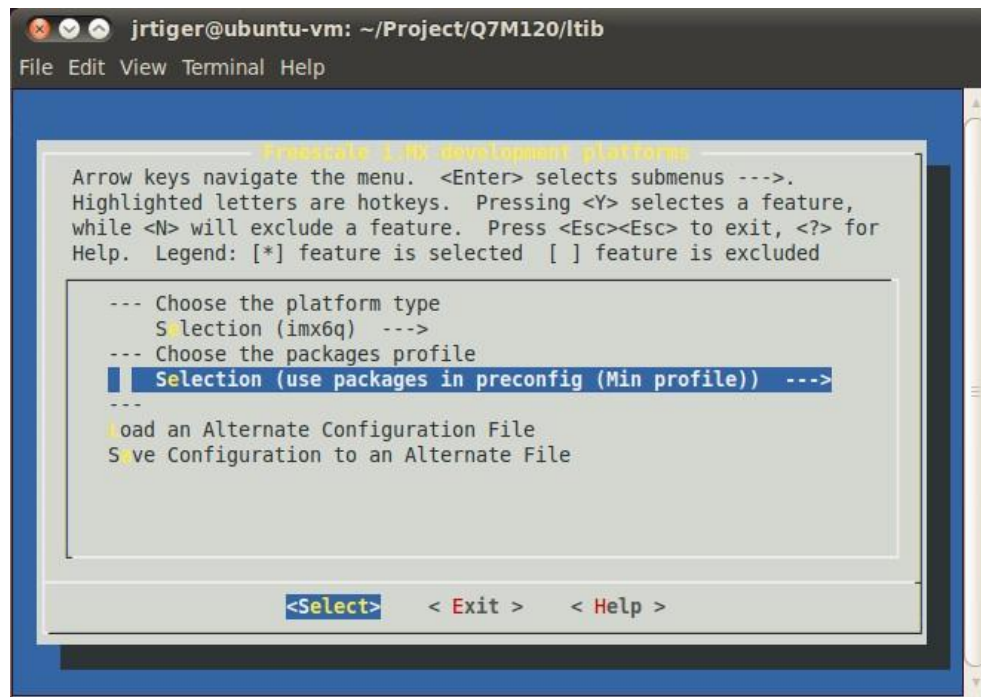
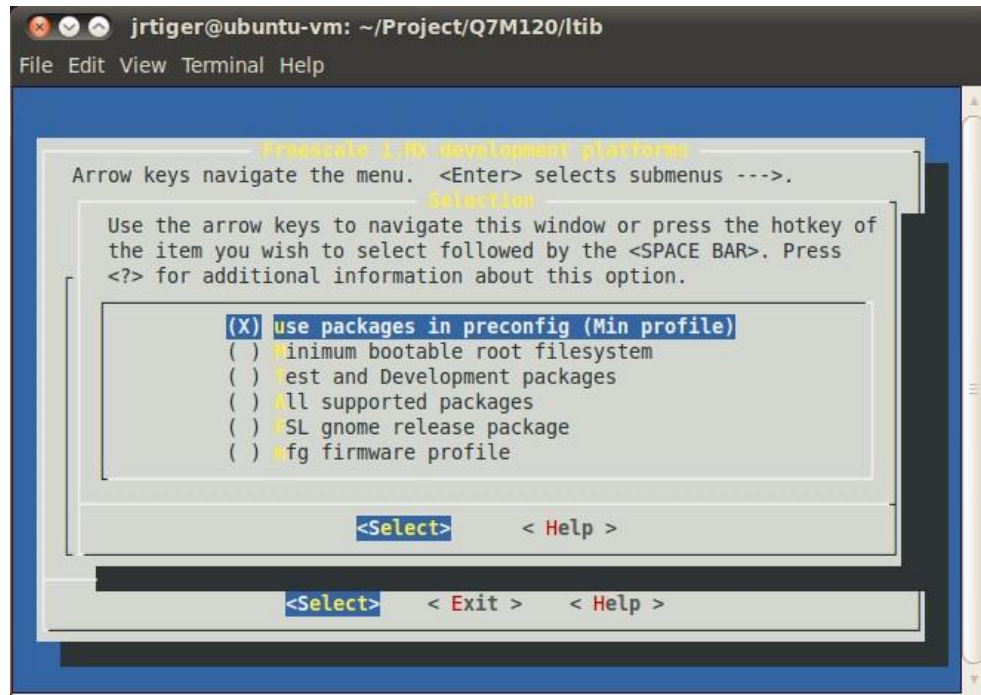
```

jrtiger@ubuntu-vm: ~/Project/Q7M120/ltib
File Edit View Terminal Help

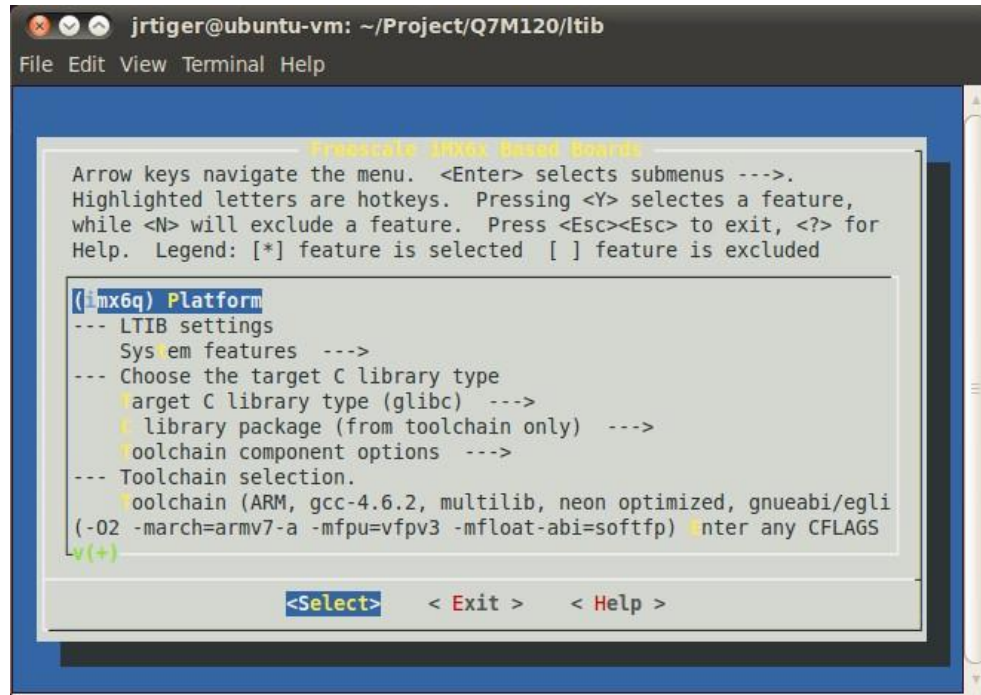
----- Freescale i.MX development platforms -----
Arrow keys navigate the menu. <Enter> selects submenus --->.
Highlighted letters are hotkeys. Pressing <Y> selects a feature,
while <N> will exclude a feature. Press <Esc><Esc> to exit, <?> for
Help. Legend: [*] feature is selected [ ] feature is excluded
--- Choose the platform type
Selection (imx6q) --->
--- Choose the packages profile
Selection (use packages in preconfig (Min profile)) --->
---
Load an Alternate Configuration File
Save Configuration to an Alternate File
<Select> < Exit > < Help >

```

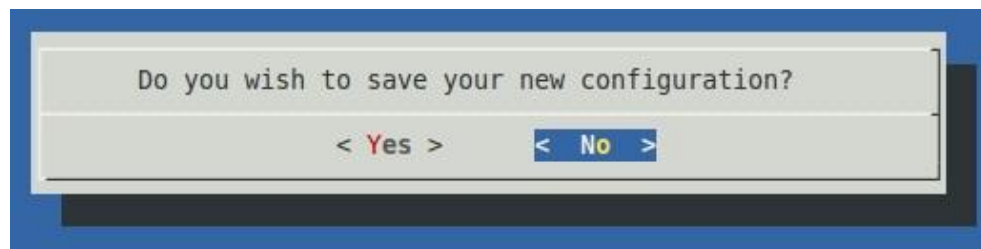
Select packages profile according to your requirement.



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 SCM120.
Create the local kernel folder.

```
$ mkdir -p ~/Project/SCM120/kernel
```

Extract kernel source to this folder.

```
$ tar jxf /opt/freescale/pkgs/linux-3.0.35.tar.bz2 -C ~/Project/SCM120/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/SCM120/kernel/linux-3.0.35
```

Enter the kernel source folder and patch the patches.

```
$ cd ~/Project/SCM120/kernel/linux-3.0.35/  
$ ./patches/patch-kernel.sh
```

Patch the SCM120 patches which you can copy from our BSP package.

```
$ tar zxf SCM120_patches_20141014.tar.gz -C  
~/Project/SCM120/kernel/linux-3.0.35  
$ cd ~/Project/SCM120/kernel/linux-3.0.35/  
$ ./patches-scm120/patch-scm120.sh
```

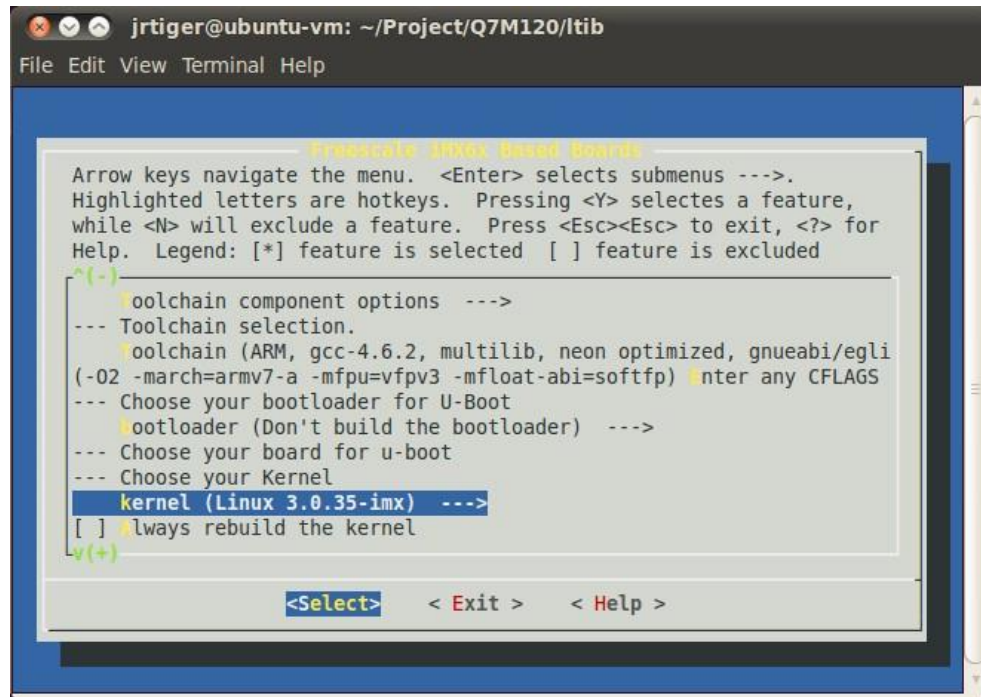
Return to ~/Project/SCM120/kernel folder and make a symbolic link to kernel source folder.

```
$ cd ~/Project/SCM120/kernel
$ ln -s linux-3.0.35 linux
```

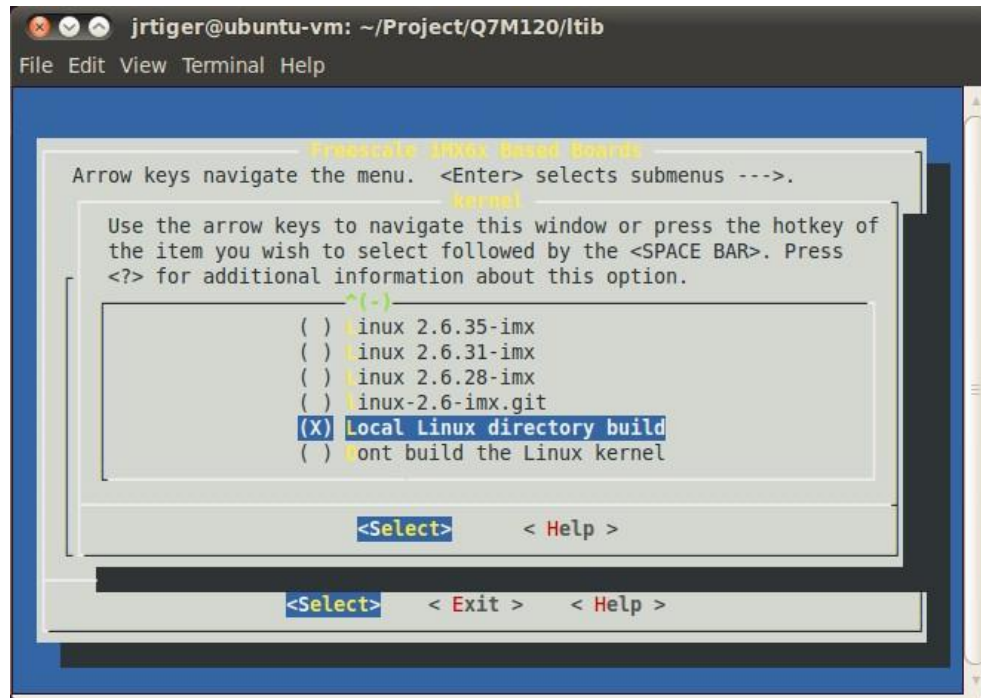
7. Reconfigure LTIB main setting to build kernel to local Linux directory.

```
$ cd ~/Project/SCM120/ltib
$ ./ltib -m config
```

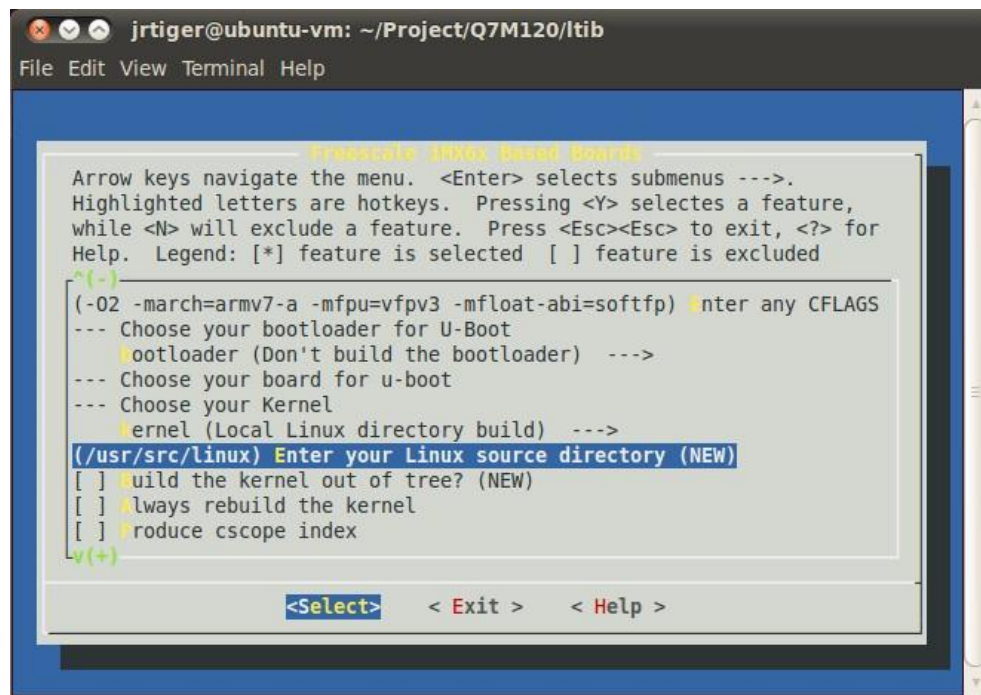
Select kernel setting.



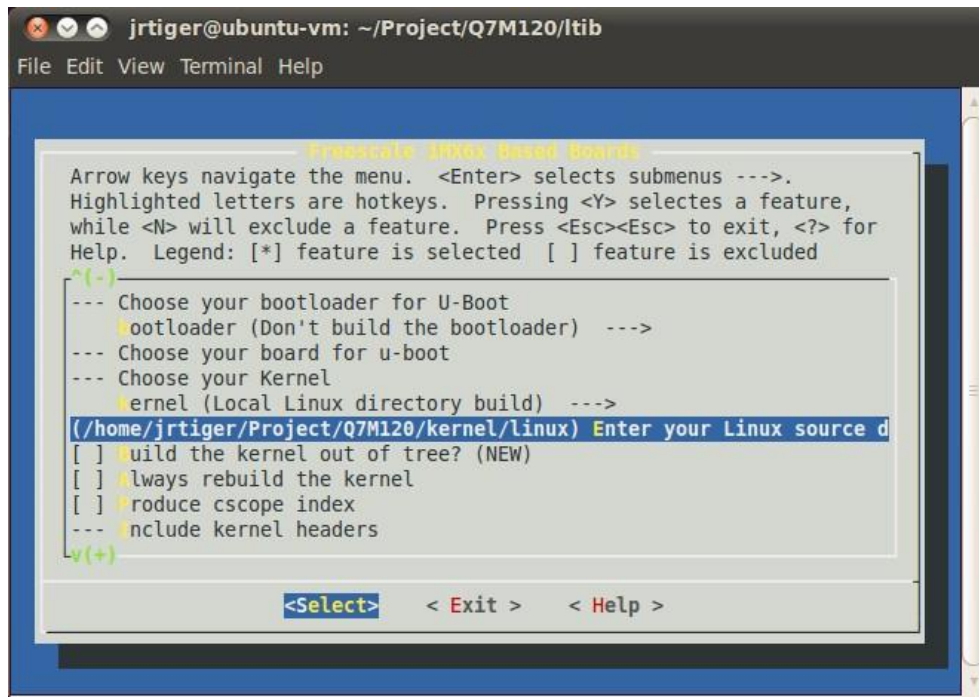
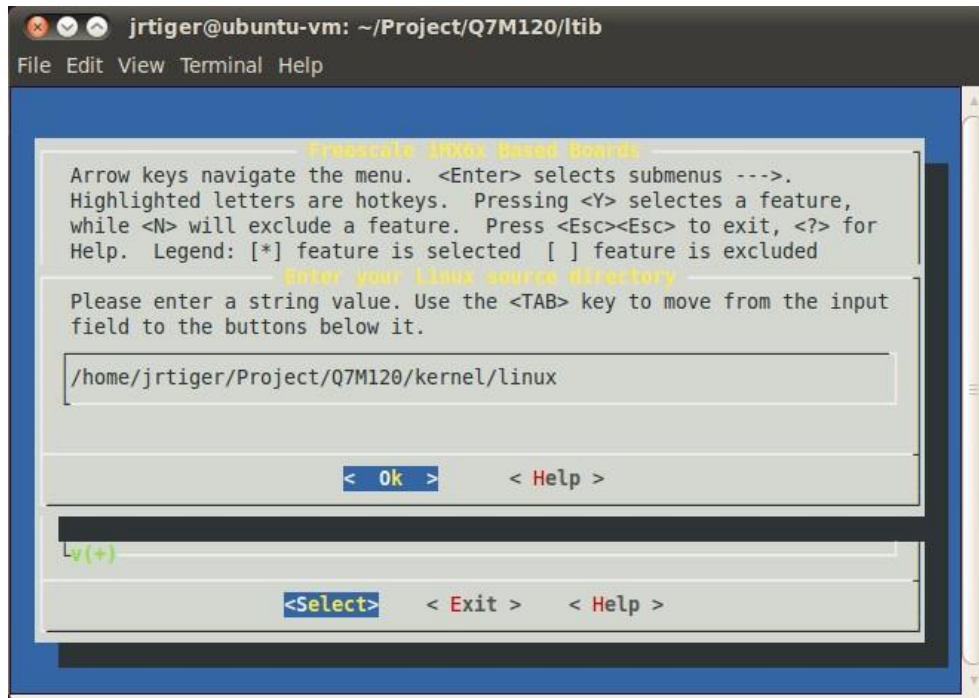
Select Local Linux directory build.



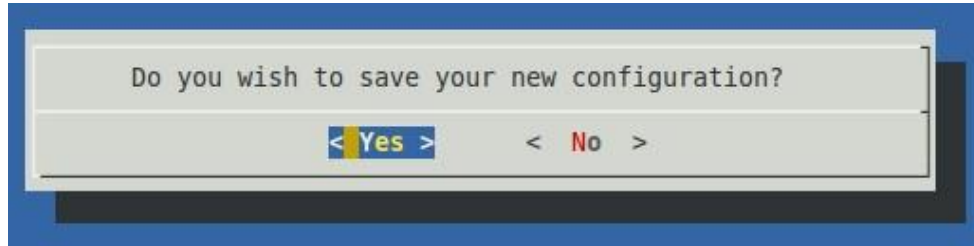
Then enter new linux source directory.



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 xzf IMX_MMCODEC_3.0.35_4.1.0_BUNDLE_CODA.tar.gz
$ tar xzf IMX_AACPD_3.0.7-2.tar.gz
```

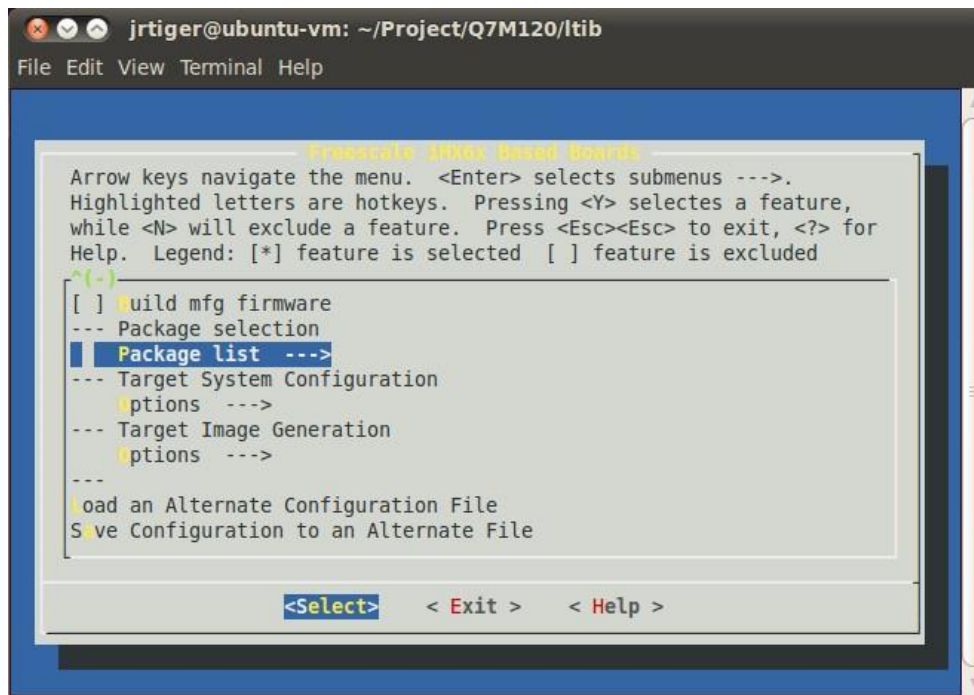
Copy all tar.gz file from ltib codecs folder to /opt/freescale/pkgs folder.

```
$ 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/
```

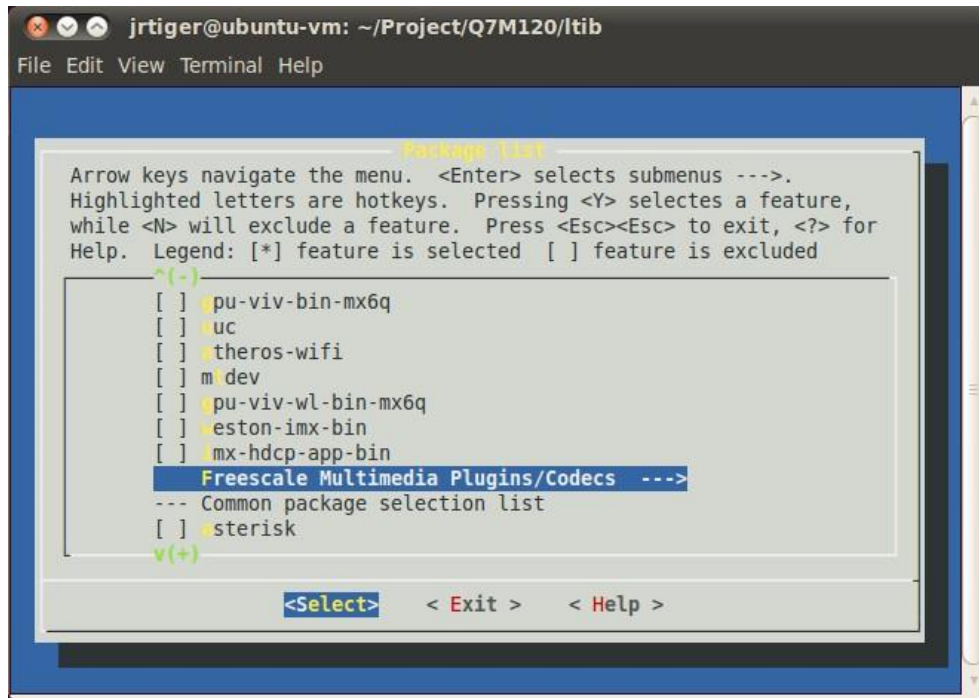
Then reconfigure LTIB main setting.

```
$ cd ~/Project/SCM120/ltib
$ ./ltib -m config
```

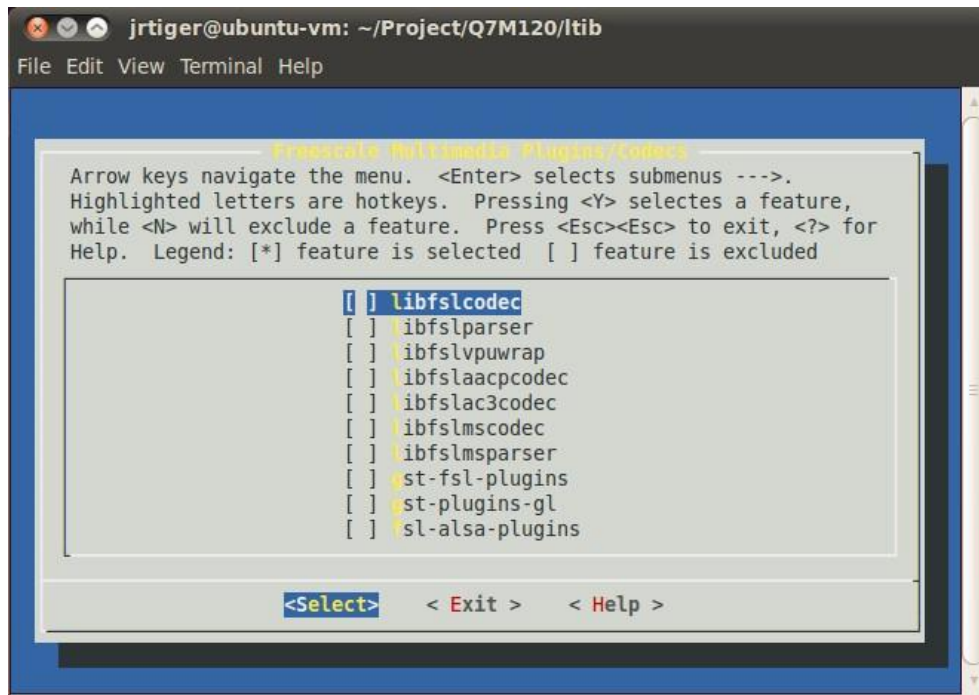
Select Package list.



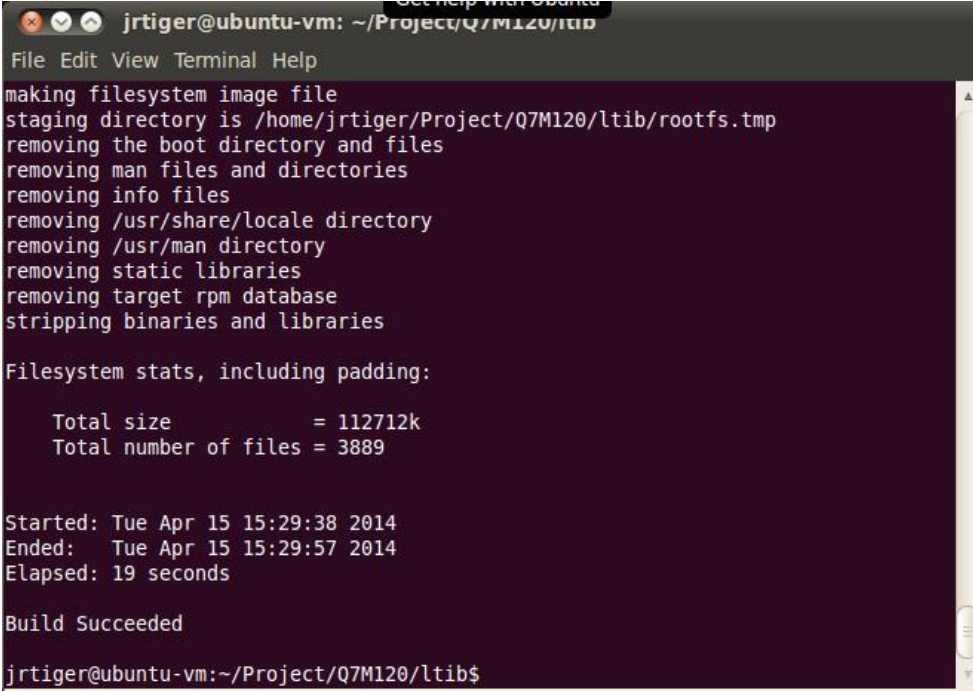
Select Freescale Multimedia Plugins/Codecs.



Select codecs you want to add.



- Build the target kernel image and root filesystem.
\$ cd ~/Project/SCM120/ltib
\$./ltib



```
jrtiger@ubuntu-vm: ~/Project/Q7M120/ltib
File Edit View Terminal Help
making filesystem image file
staging directory is /home/jrtiger/Project/Q7M120/ltib/rootfs.tmp
removing the boot directory and files
removing man files and directories
removing info files
removing /usr/share/locale directory
removing /usr/man directory
removing static libraries
removing target rpm database
stripping binaries and libraries

Filesystem stats, including padding:
    Total size           = 112712k
    Total number of files = 3889

Started: Tue Apr 15 15:29:38 2014
Ended:   Tue Apr 15 15:29:57 2014
Elapsed: 19 seconds

Build Succeeded

jrtiger@ubuntu-vm:~/Project/Q7M120/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/SCM120/ltib/rootfs/boot directory, you can find kernel image ulmage.

4.2 Deployment

4.2.1 Create a bootable SD Card for SCM120

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=./ulmage of=/dev/sdb bs=512 seek=2048 conv=fsync`
This will copy ulmage 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/user/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.