AXIOMTEK

Q7M100 and Q7B100

BSP

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.

ESD Precautions

Computer boards have integrated circuits sensitive to static electricity. To prevent chipsets from electrostatic discharge damage, please take care of the following jobs with precautions:

- Do not remove boards or integrated circuits from their anti-static packaging until you are ready to install them.

- Before holding the board or integrated circuit, touch an unpainted portion of the system unit chassis for a few seconds. It discharges static electricity from your body.

- Wear a wrist-grounding strap, available from most electronic component stores, when handling boards and components.

Trademarks Acknowledgments

Axiomtek is a trademark of Axiomtek Co., Ltd.

Other brand names and trademarks are the properties and registered brands of their respective owners.
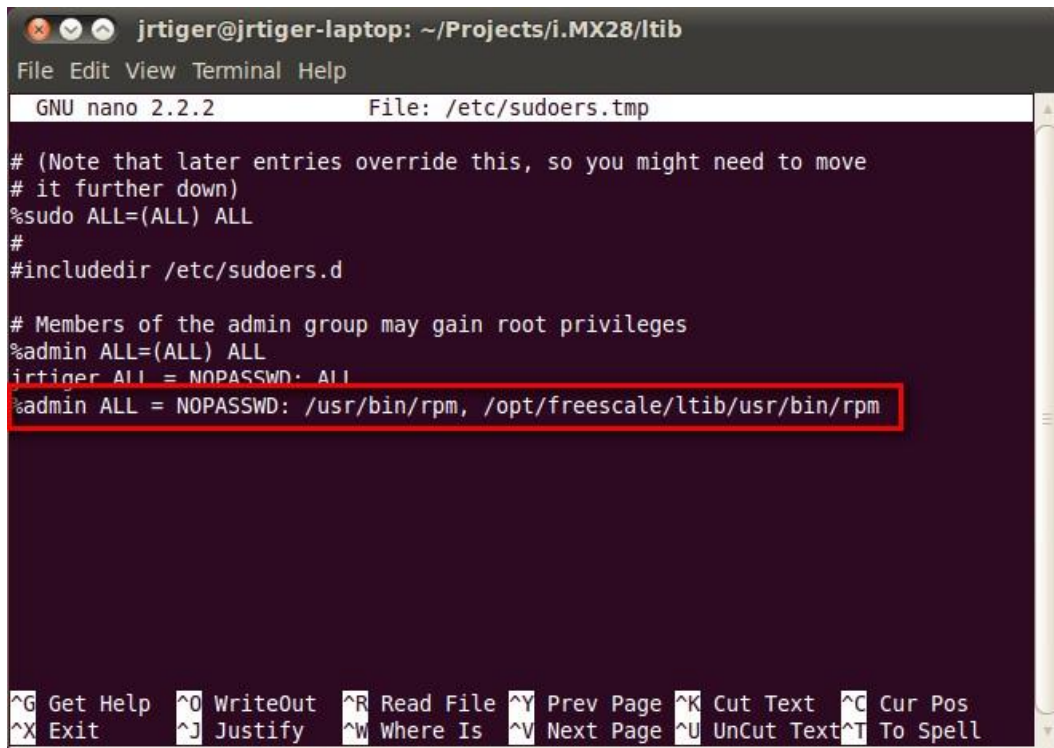
Table of Contents

Chapter 1
Host Development System Installation

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@jrtiger-laptop: ~/Projects/i.MX28/ltib

File  Edit  View  Terminal  Help

  GNU nano 2.2.2              File: /etc/sudoers.tmp

# (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
```
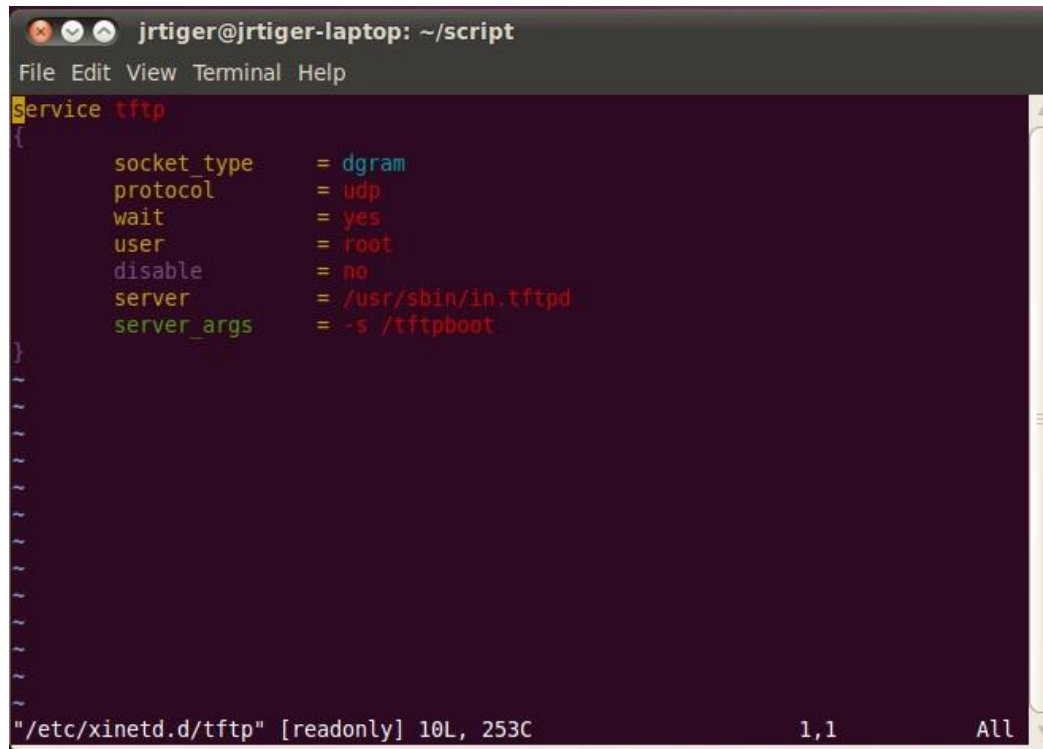
4.   Install host packages needed by LTIB as follows:
      $ sudo aptitude -y install gettext libgtk2.0-dev rpm bison m4 libfreetype6-dev
      $ sudo aptitude -y install libdbus-glib-1-dev liborbit2-dev intltool
      $ sudo aptitude -y install ccache ncurses-dev zlib1g zlib1g-dev gcc g++ libtool
      $ sudo aptitude -y install uuid-dev liblzo2-dev
      $ sudo aptitude -y install tcl

5. Install and configure TFTP server:
    After tftpd 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

```
😣 ⊙ ⊙   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/Q7M100/ltib/rootfs /tools/rootfs

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

1.2   Install LTIB

1.   Build your own project folder.
      $ mkdir -p ~/Project/Q7M100

2.   Extract the source tar ball with the following command:
      $ tar zxf L2.6.35_10.12.01_SDK_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 L2.6.35_10.12.01_ER_source
      $ ./install

```
  ⊗ ⊘ ⊙   jrtiger@ubuntu-vm: ~/L2.6.35_10.12.01_ER_source

File  Edit  View  Terminal  Help

If the Licensed Software includes software and hardware developed by Global
Locate, Inc. ("Global Locate"), and acquired by Broadcom Corporation, you must
separately obtain rights beyond evaluation and demonstration for the Global
Locate software from Broadcom Corporation.


CSR

If the Licensed Software includes software and hardware developed by Cambridge
Silicon Radio, Inc. ("CSR"), you must separately obtain rights beyond evaluation

and demonstration for the CSR software from CSR.

LAOPT27 Nov 12, 2010

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/L2.6.35_10.12.01_ER_source/ltib

Where do you want to install LTIB ? (/home/jrtiger/L2.6.35_10.12.01_ER_source)
~/Project/Q7M100
```

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
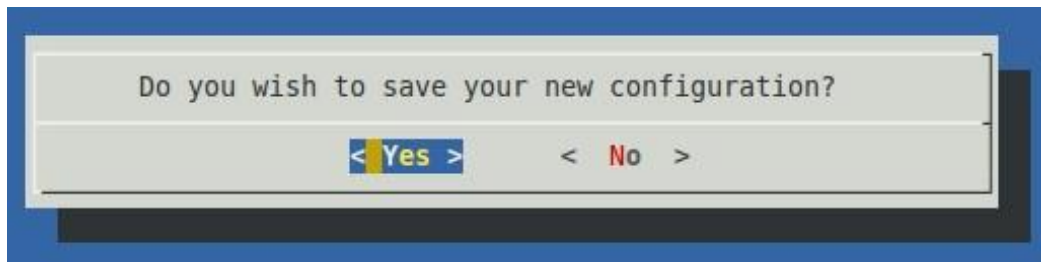   host tools that it will place under */opt/freescale* .

      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.
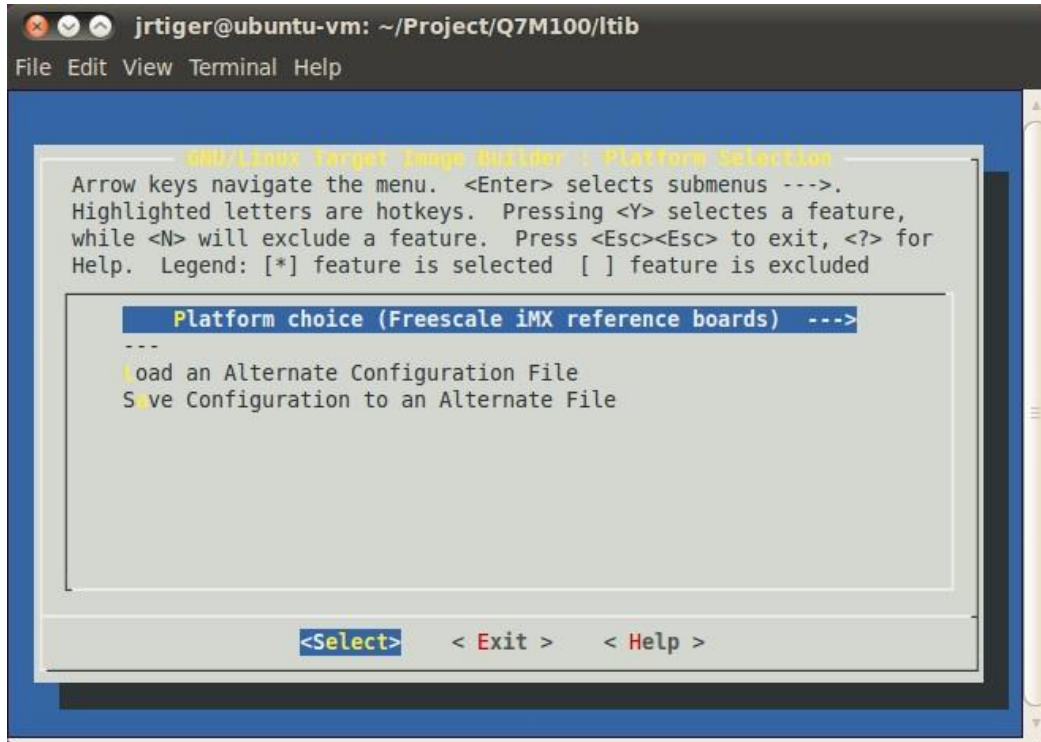
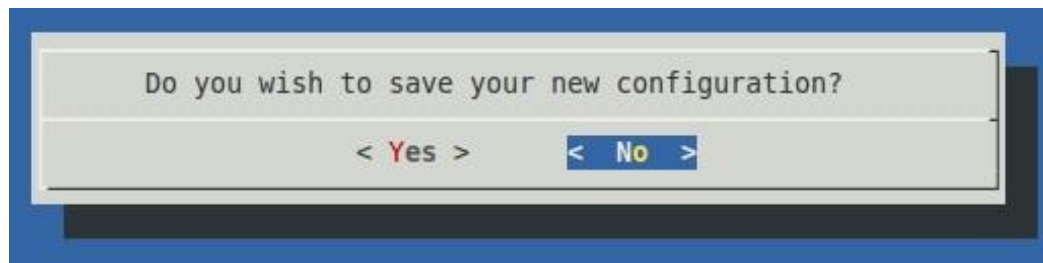Then save the configuration.

Select the imx28 sub-platform.

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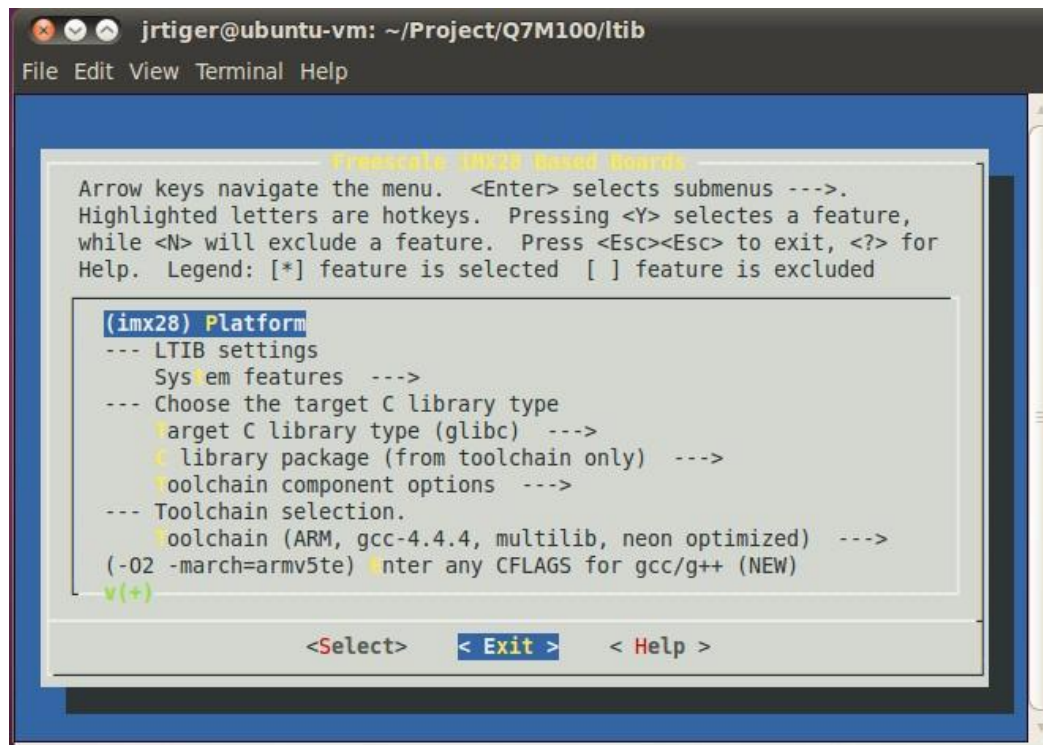
Just exit without saving main LTIB configuration.

6. Create and patch Q7M100 kernel.
    Create the local kernel folder.
    $ mkdir -p ~/Project/Q7M100/kernel

    Extract kernel source to this folder.
    $ tar jxf /opt/freescale/pkgs/linux-2.6.35.3.tar.bz2 -C ~/Project/Q7M100/kernel

    Extract i.MX platform patches to kernel source folder.
    $ tar jxf /opt/freescale/pkgs/linux-2.6.35.3-imx_10.12.01.bz2 –C ~/Project/Q7M100/kernel/linux-2.6.35.3

    Enter the kernel source folder and patch the patches.
    $ cd ~/Project/Q7M100/kernel/linux-2.6.35.3/
    $ ./patches/patch-kernel.sh

    Patch the Q7M100 patches which you can copy from our BSP package.
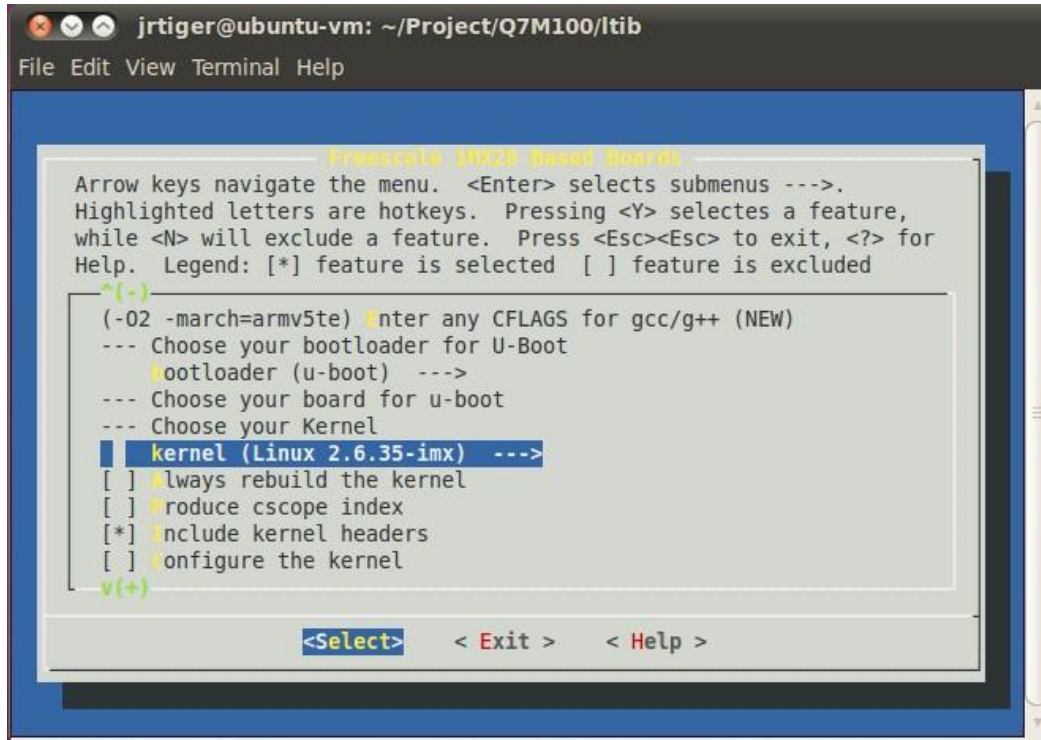 $ patch -p1 < ~/patch-2.6.35.3-Q7M100-013.patch

Return to *~/Project/Q7M100/kernel* folder and make a symbolic link to kernel source folder.
$ cd ~/Project/Q7M100/kernel
$ ln -s linux-2.6.35.3 linux

7.   Reconfigure LTIB main setting to build kernel to local Linux directory.
$ cd ~/Project/Q7M100/ltib
$ ./ltib -m config

Select kernel setting.

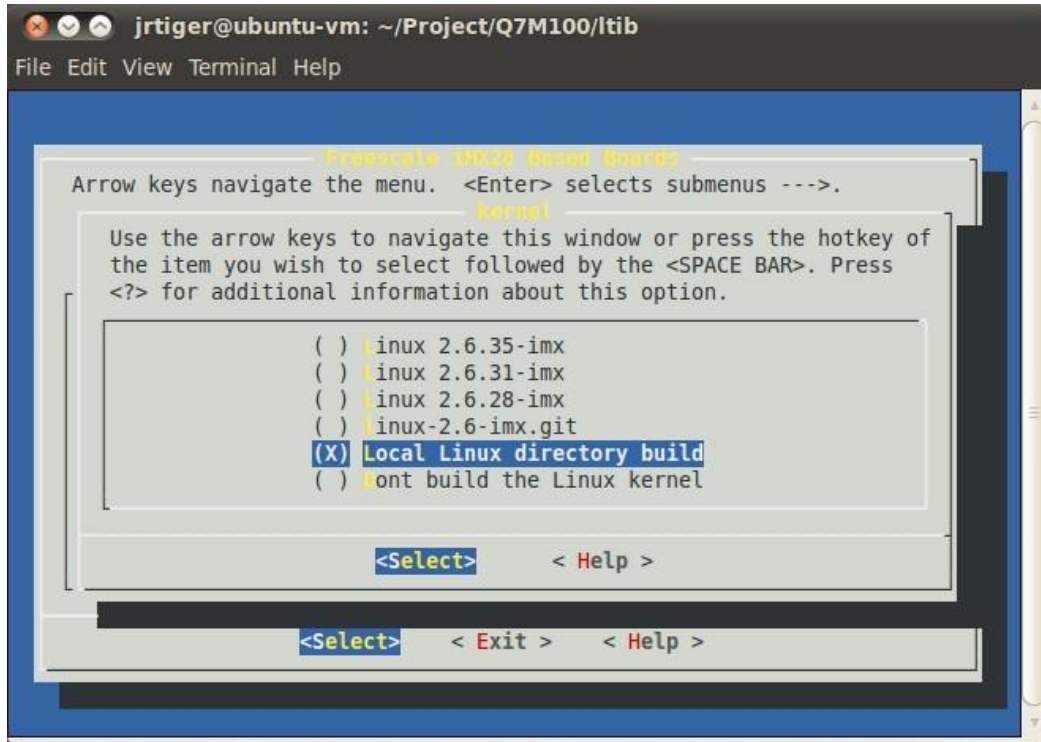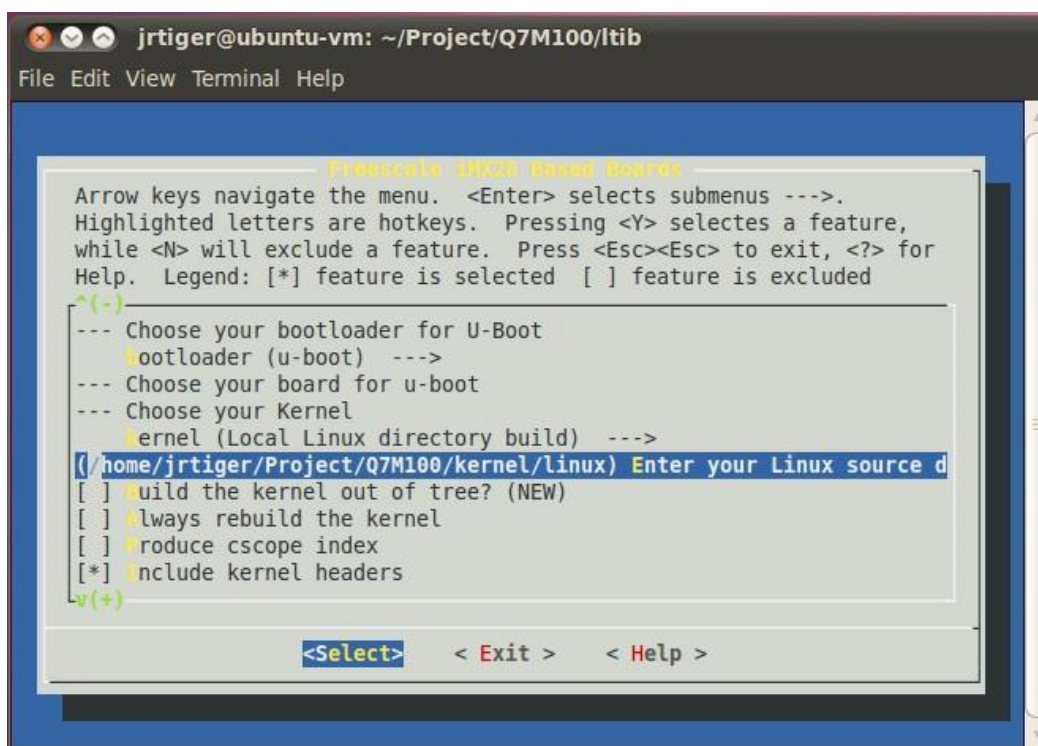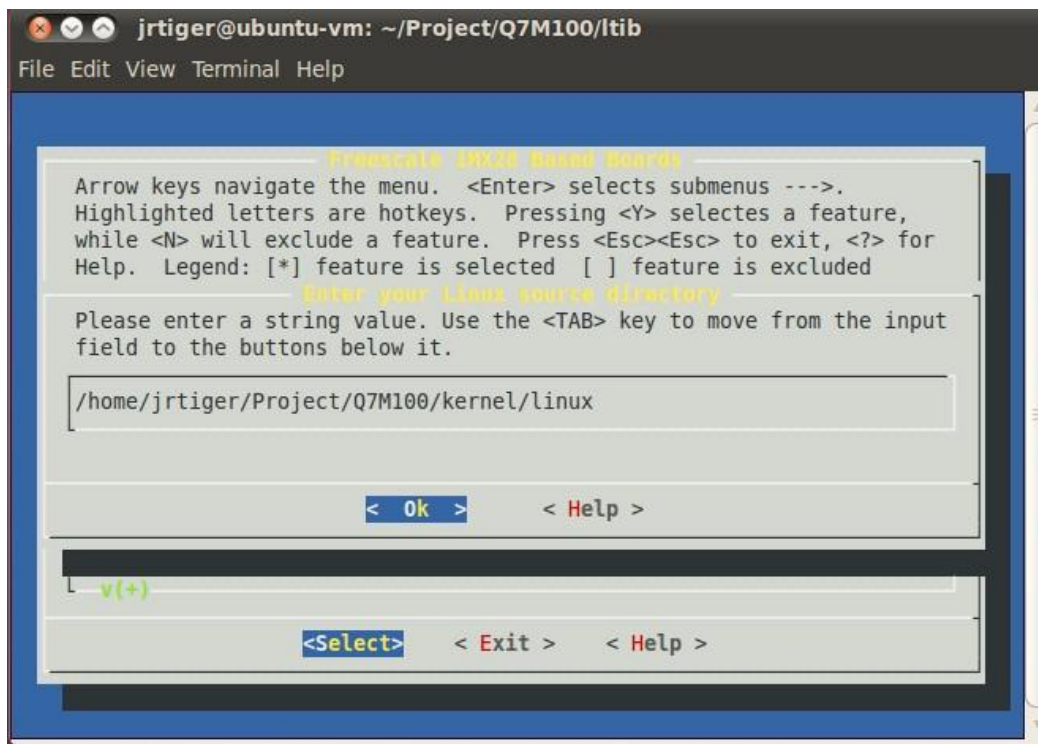Select Local Linux directory build.

Then enter new linux source directory.
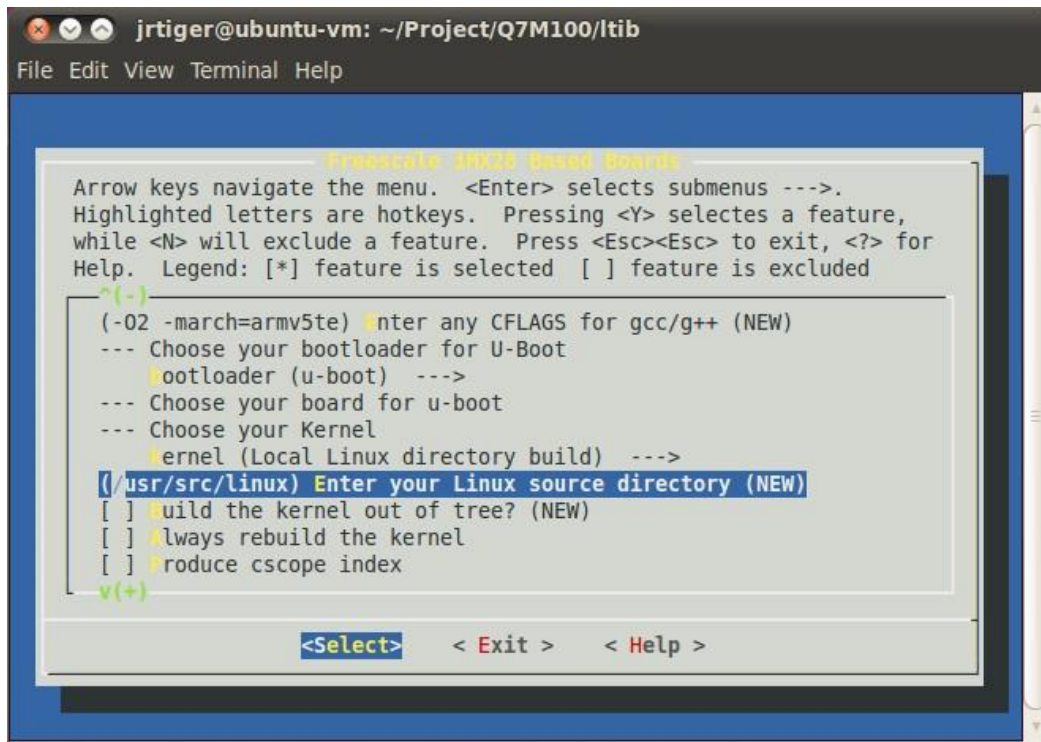
Enter your own linux kernel source folder.

Finally save the configuration.
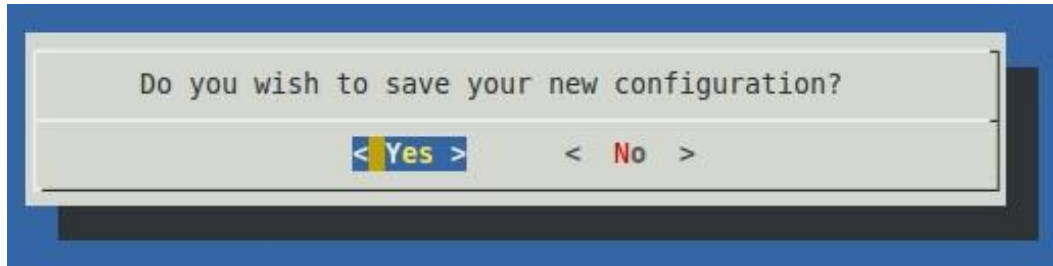
8. Options :
Adding iMX28 Multimedia codecs support:
Extract codec file.
$ tar zxf IMX_MMCODECS_10.12.tar.gz
Copy all tar.gz file to */opt/freescale/pkgs* folder.
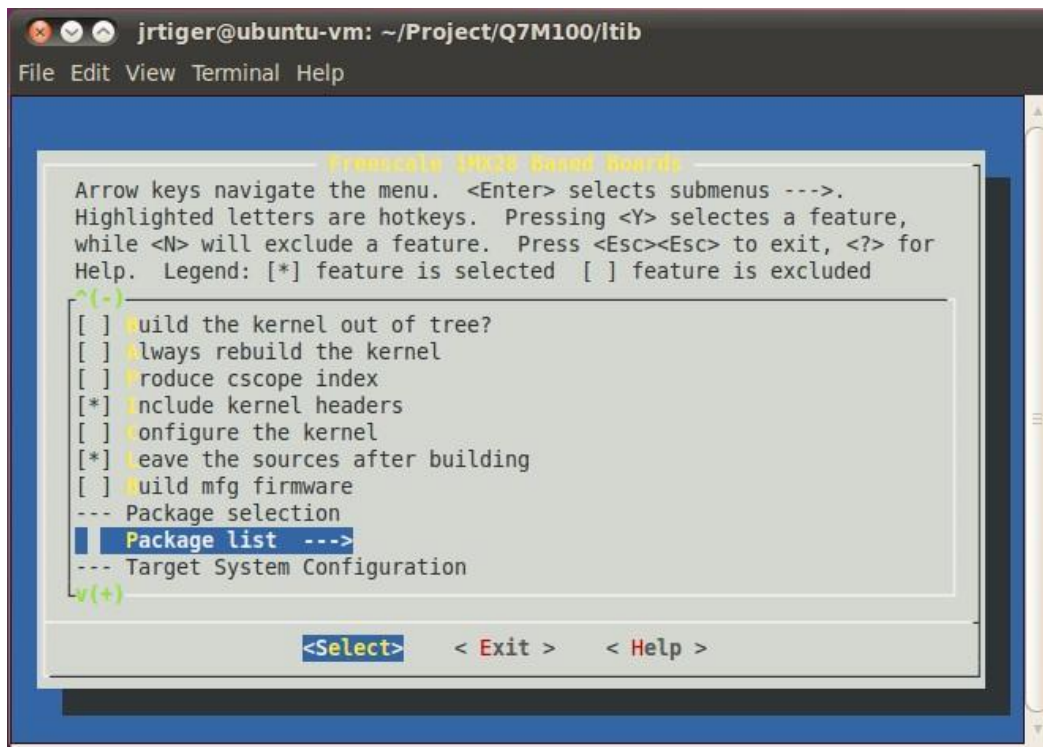$ cp IMX_MMCODECS_10.12/*.tar.gz /opt/freescale/pkgs/
Then reconfigure LTIB main setting.
$ cd ~/Project/Q7M100/ltib
$ ./ltib -m config

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/Q7M100/ltib
   $ ./ltib
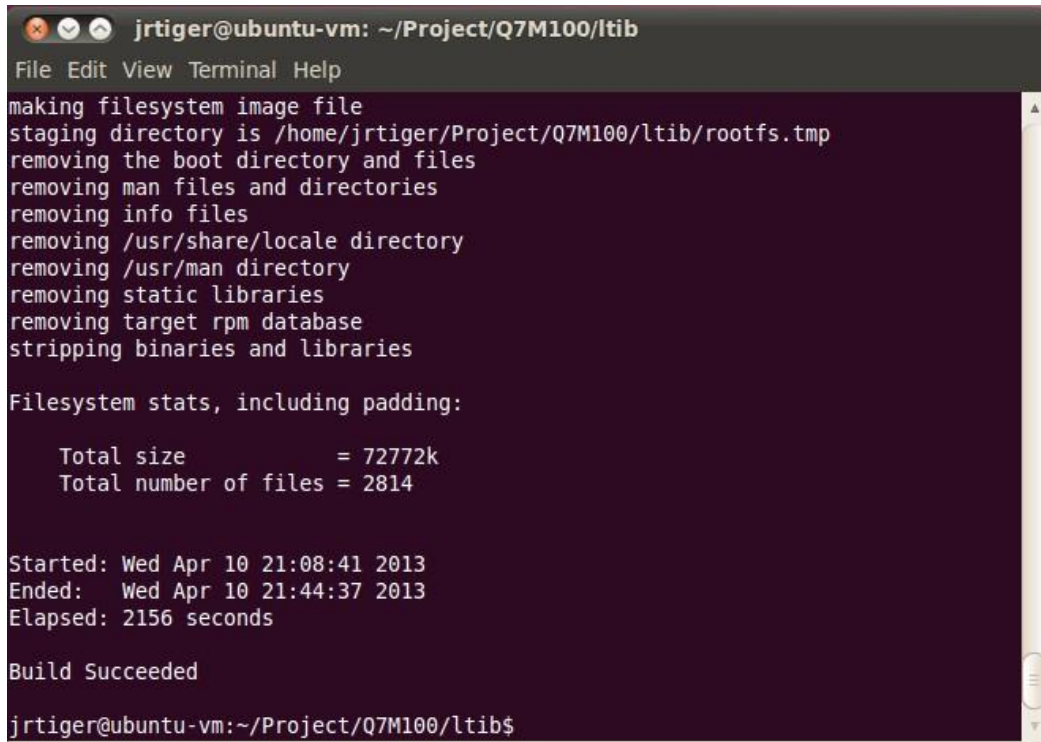
```
jrtiger@ubuntu-vm: ~/Project/Q7M100/ltib
File Edit View Terminal Help
making filesystem image file
staging directory is /home/jrtiger/Project/Q7M100/ltib/rootfs.tmp
removing the boot directory and files
removing man files and directories
removing info files
removing /usr/share/locale directory
removing /usr/man directory
removing static libraries
removing target rpm database
stripping binaries and libraries

Filesystem stats, including padding:

    Total size           = 72772k
    Total number of files = 2814


Started: Wed Apr 10 21:08:41 2013
Ended:   Wed Apr 10 21:44:37 2013
Elapsed: 2156 seconds

Build Succeeded

jrtiger@ubuntu-vm:~/Project/Q7M100/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/Q7M100/ltib/rootfs/boot* directory, you can find kernel image *uImage*.

1.3   Compile Driver and Tools

1.   Compile and build driver of Q7M100 hardware IO devices.
      Change to *ltib* directory.
      $ cd ~/Project/Q7M100/ltib
      Enter ltib shell mode (this is a developer function that provides an environment for compiling and building package).
      $ ./ltib -m shell
      Extract driver source to *ltib/rpm/BUILD* directory.
      LTIB> tar jxf Q7M100-axio.tar.bz2 -C rpm/BUILD/
      Change to *axio* directory.
      LTIB> cd rpm/BUILD/axio
      Build the driver module.
      LTIB> make
      Install the driver module to target root filesystem.
      LTIB> make install

2.   Compile and build Q7M100 library for IO devices (it depends on q7m100.ko.).
      Change to *ltib* directory.
      $ cd ~/Project/Q7M100/ltib
      Enter ltib shell mode (this is a developer function that provides an environment for compiling and building package).
      $ ./ltib -m shell
      Extract source to *ltib/rpm/BUILD* directory.
      LTIB> tar jxf Q7M100-q7_lib.tar.bz2 -C rpm/BUILD/
      Change to *q7_lib* directory.
      LTIB> cd rpm/BUILD/q7_lib
      Build shared library libq7m100.so.1.0.0 and install to target root filesystem.
      LTIB> ./make-lib.sh
      Build RTC function tools.
      LTIB> ./make-test.sh
      Build EEPROM test program.
      LTIB> ./make-eeprom.sh

3.   Compile and build Q7M100 tools.
      Change to *ltib* directory.
      $ cd ~/Project/Q7M100/ltib
      Enter ltib shell mode (this is a developer function that provides an environment for compiling and building package).
      $ ./ltib -m shell
      Extract tools source to *ltib/rpm/BUILD* directory.
      LTIB> tar jxf Q7M100-tools.tar.bz2 -C rpm/BUILD/
      Change to tools directory.
      LTIB> cd rpm/BUILD/tools
      Build the tools.
      LTIB> make

4.   Then copy all binary file to target root filesystem you build. The root filesystem is a directory tree found under *rootfs*.

This page is intentionally left blank.

Chapter 2
U-Boot for Q7M100

2.1   Booting the system with an NFS filesystem

By default, U-Boot is configured to boot from NFS. To boot from NFS, first you must set some configurations. Press any key to break from the boot progress and set configurations.

```
Setup TFTP server IP:
MX28 U-Boot > setenv serverip 192.168.0.100
Setup board IP address:
MX28 U-Boot > setenv ipaddr 192.168.0.5
Setup rootfs path:
MX28 U-Boot > setenv nfsroot /tools/rootfs
Setup boot delay:
MX28 U-Boot > setenv bootdelay 2
Save configurations:
MX28 U-Boot > saveenv
```

Reset the board and kernel will be launched.

2.2   Booting the system from eMMC

First you need to download kernel root filesystem through TFTP server, and install it into eMMC.

```
Setup TFTP server IP:
MX28 U-Boot > setenv serverip 192.168.0.100
Setup board IP address:
MX28 U-Boot > setenv ipaddr 192.168.0.5
Download kernel image through TFTP server
MX28 U-Boot > tftpboot uImage
Using FEC0 device
TFTP from server 192.168.0.100; our IP address is 192.168.0.5
Filename 'uImage'.
Load address: 0x42000000
Loading: FEC: Link is down 7809
######
done
Bytes transferred = 2454628 (257464 hex)
Install kernel image into eMMC:
MX28 U-Boot > mxs_mmc install 0 0x42000000 0x257464 uImage
Done: 4795 (12bb hex) sectors written at 256 (100 hex)
```

Download root filesystem through TFTP server (the file size limitation of this method is 96MB):
```
MX28 U-Boot > tftpboot rootfs.ext2
Using FEC0 device
TFTP from server 192.168.0.100; our IP address is 192.168.0.5
Filename 'rootfs.ext2'.
Load address: 0x42000000
```

```
Loading: FEC: Link is down 7809
######
done
Bytes transferred = 74526720 (4713000 hex)
Install root filesystem into eMMC:
MX28 U-Boot > mxs_mmc install 0 0x42000000 0x4713000 rootfs
Done: 145560 (23898 hex) sectors written at 32768 (8000 hex)
Run system from eMMC:
MX28 U-Boot > run bootcmd_mmc
```

Note: ***We configure 1GB EXT partition for root filesystem. If you want to use full size to deploy your system, try booting from NFS filesystem and then format EXT partition. Extract filesystem and install it to EXT partition as follows:***

Booting with NFS filesystem:
MX28 U-Boot > run bootcmd_net
Format EXT partition as EXT3:
$ mkfs.ext3 /dev/mmcblk0p3
Mount this partition:
$ mount /dev/mmcblk0p3 /mnt/src
Extract filesystem and install it to EXT partition:
$ tar jxf rootfs.tar.bz2 -C /mnt/src
Then reboot and boot it from eMMC.

2.3　　　Reference Document

● uboot_mx28.pdf
● ltib_build_host_setup.pdf