LTIB for i.MX28, a step-by-step guide

Note: This guide shows how to get a target system building and running on an i.MX28 EVK board.

Install LTIB

At the time of this writing, the latest BSP release is L2.6.31_10.08.01.

First, extract the source tar ball. You can do that with this command:

```
% tar -xzf L2.6.31_10.08.01_SDK_source.tar.gz
```

Now change into the extracted folder and execute the `install` script that is inside.

```
% cd L2.6.31_10.08.01_SDK_source
% ./install
```

The installer will ask you to read and accept a EULA and then ask you for the install directory. For this example, choose your home directory.

After the installation, there will be a new directory named "`ltib`" under the install directory you specified. This new directory contains the actual `ltib` Perl script, config files, package specs, and utilities.

From this point on, the LTIB install directory will be referred to as "`~/ltib`" in example command lines, assuming you installed LTIB into your home directory.

Configure and build

To configure and build your LTIB installation the first time, run this command from the `~/ltib` directory:

```
% ./ltib
```

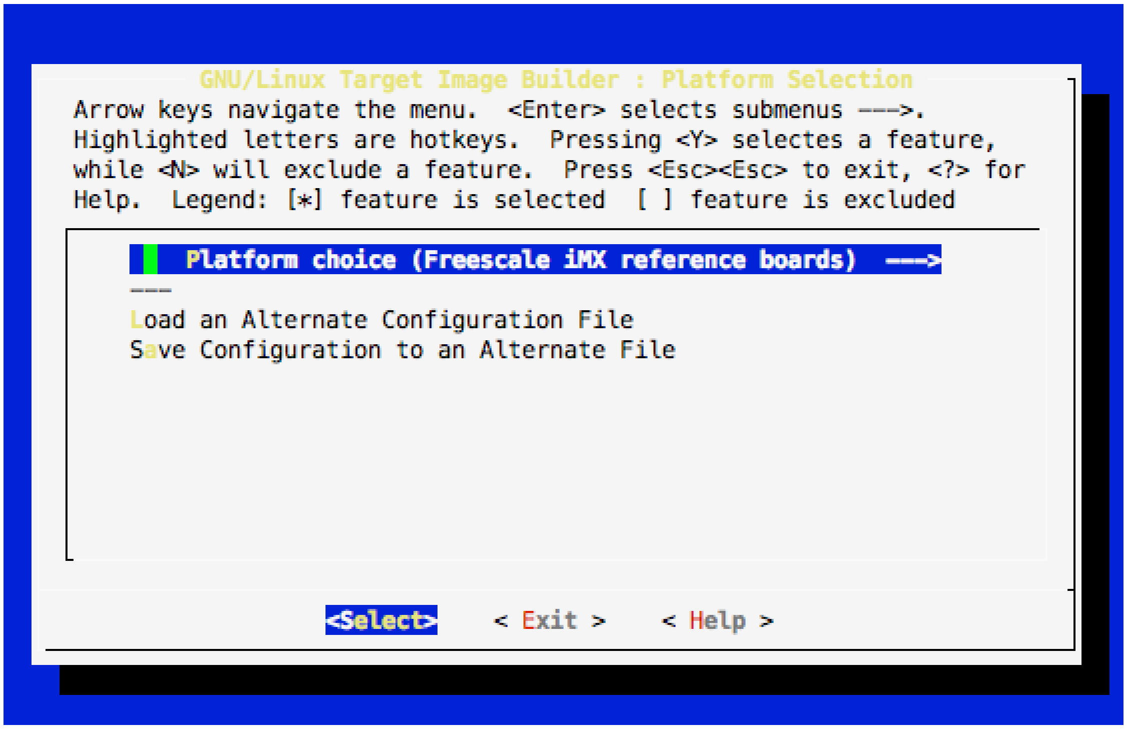
This command invokes LTIB with its default behavior of performing a build. Since the installation has not yet been configured, LTIB will present the curses configuration screen before building. Then, when you exit the configuration screen, LTIB will build the target image. Also, the very first time you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

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

```
% ./ltib -m config
```

First time configuration

The first time you configure a given LTIB installation, you will be asked to select the platform. The first screen that appears looks like this:



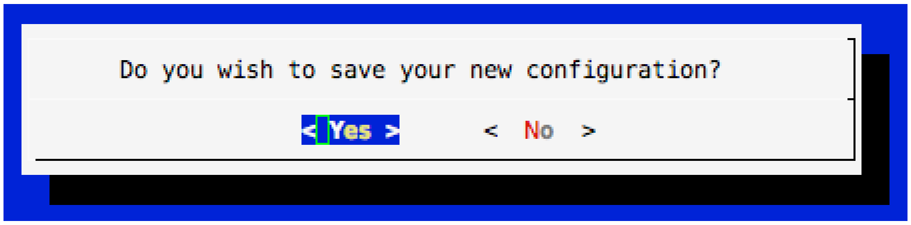
```
        GNU/Linux Target Image Builder : Platform Selection
  Arrow keys navigate the menu.  <Enter> selects submenus --->.
  Highlighted letters are hotkeys.  Pressing <Y> selectes a feature,
  while <N> will exclude a feature.  Press <Esc><Esc> to exit, <?> for
  Help.  Legend: [*] feature is selected  [ ] feature is excluded

      Platform choice (Freescale iMX reference boards)  --->
      ---
      Load an Alternate Configuration File
      Save Configuration to an Alternate File

                 <Select>    < Exit >    < Help >
```

Because the i.MX BSP contains only the i.MX platform, you don't need to change anything on this screen.

On all of the curses configuration screens for both LTIB and the kernel (they use the same `mconf` system), there are a standard set of keys that are important to know. The bottom of each main screen shows **<Select>** and **<Exit>** items. The right and left arrows move between these two items.

When **<Select>** is highlighted, the up and down arrows move between items in the main list. Pressing return on any item with an arrow to the right will open up a sub-screen. Some sub-screens appear as windows on top of the current screen, and some appear as their own screens.

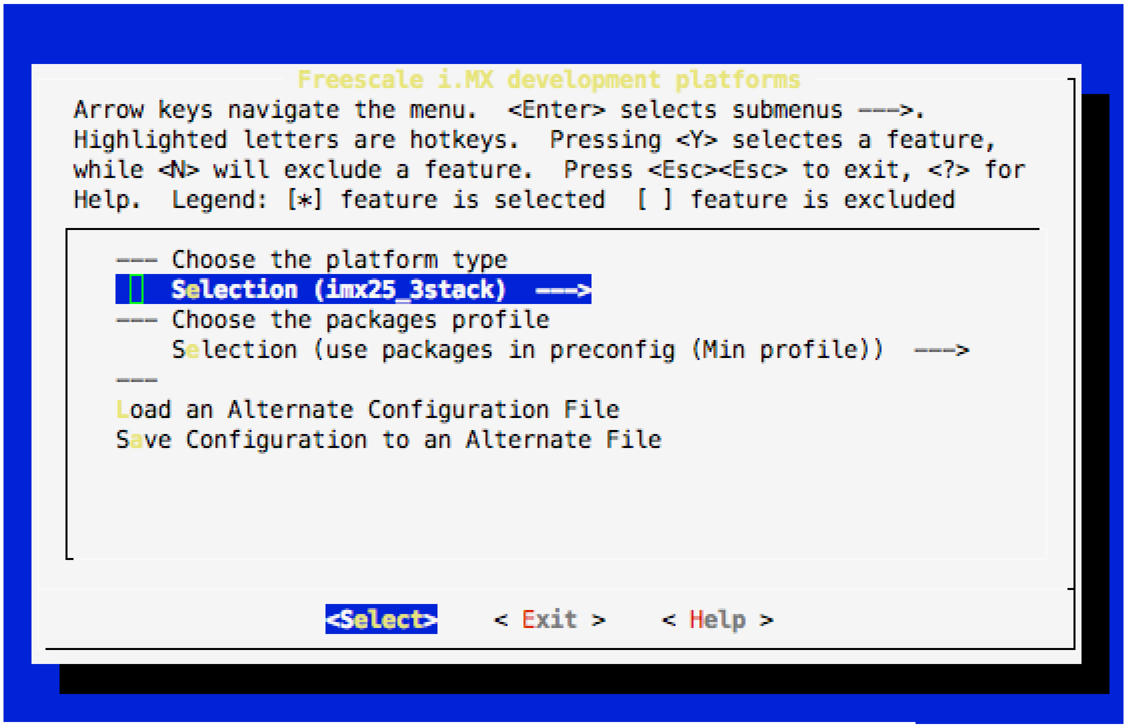
When **<Exit>** is selected, you can hit the return key to exit the current screen. On most main screens, exiting will bring up the following dialog asking if you want to save changes.



```
     Do you wish to save your new configuration?

             < Yes >      <  No  >
```

Here, use the right and left arrow keys to select between **<Yes>** and **<No>**. Press return to activate your selection. Selecting **<No>** will cause LTIB to not continue to the following step.

⭐ If you haven't already, exit the platform selection screen and save the configuration when asked. This will bring up another sub-platform selection screen, shown here:



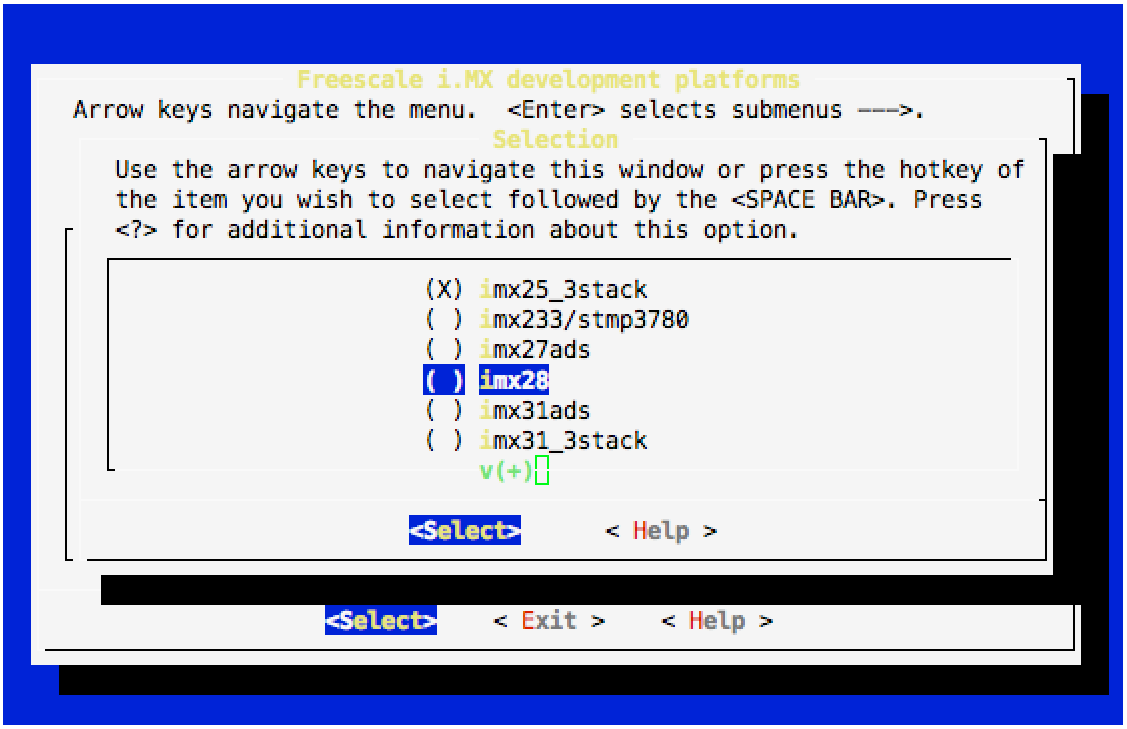
```
                 Freescale i.MX development platforms
  Arrow keys navigate the menu.  <Enter> selects submenus --->.
  Highlighted letters are hotkeys.  Pressing <Y> selectes a feature,
  while <N> will exclude a feature.  Press <Esc><Esc> to exit, <?> for
  Help.  Legend: [*] feature is selected  [ ] feature is excluded

     --- Choose the platform type
         Selection (imx25_3stack)  --->
     --- 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 >
```

The first option available on this screen is used to choose the sub-platform type. For the i.MX platform, the sub-platform is the specific chip family combined with the board type. For the i.MX28, this is the EVK board. The second option on this screen, under the **Choose the packages profile** item, lets you select the default set of packages. Individual packages can be enabled and disabled later, as you need. The default choice is sufficient for now.

If you need to change the sub-platform later, use the command:

```
% ./ltib -m selectype
```

Move the cursor down to the **Selection** item below the **Choose the platform type** line and press return to open up a platform selection menu, shown here:



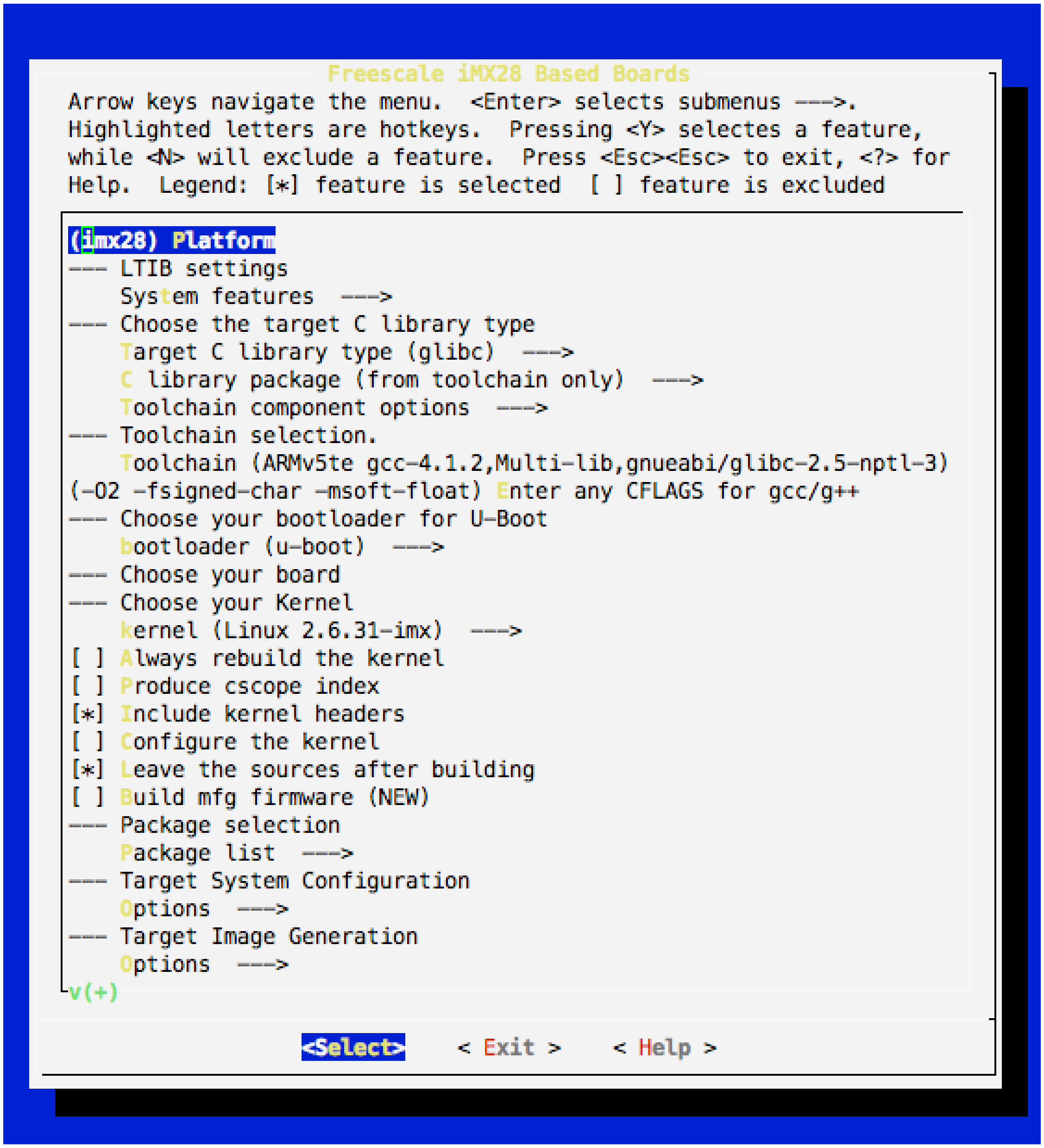
```
                 Freescale i.MX development platforms
  Arrow keys navigate the menu.  <Enter> selects submenus --->.
                              Selection
     Use the arrow keys to navigate this window or press the hotkey of
     the item you wish to select followed by the <SPACE BAR>. Press
     <?> for additional information about this option.

                          (X) imx25_3stack
                          ( ) imx233/stmp3780
                          ( ) imx27ads
                          ( ) imx28
                          ( ) imx31ads
                          ( ) imx31_3stack
                              v(+)

                        <Select>      < Help >

                    <Select>    < Exit >    < Help >
```

⭐ Select the **imx28** platform, which is near the top of the list, and press return to return to the main screen.

⭐ Exit the sub-platform selection screen and save the configuration. This will bring you to the main LTIB configuration screen.

LTIB configuration

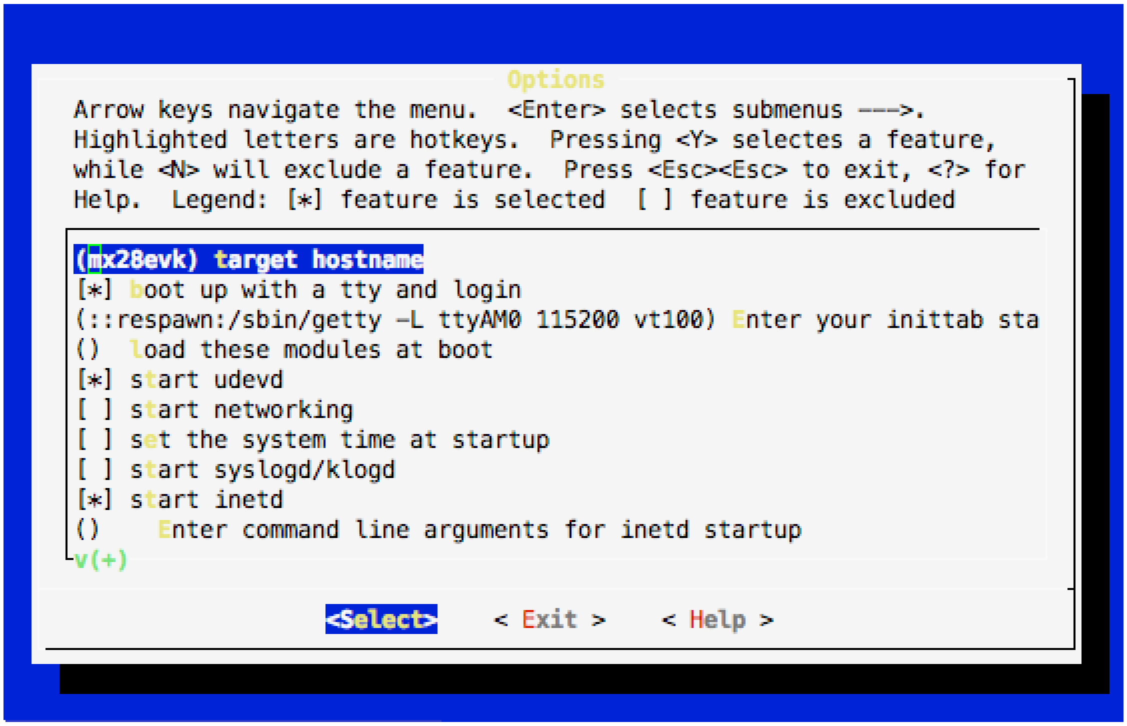
The screen below shows the full contents of the main LTIB configuration screen for the i.MX28.



```
Arrow keys navigate the menu.  <Enter> selects submenus --->.
Highlighted letters are hotkeys.  Pressing <Y> selectes a feature,
while <N> will exclude a feature.  Press <Esc><Esc> to exit, <?> for
Help.  Legend: [*] feature is selected  [ ] feature is excluded

(imx28) Platform
--- LTIB settings
    System features  --->
--- Choose the target C library type
    Target C library type (glibc)  --->
    C library package (from toolchain only)  --->
    Toolchain component options  --->
--- Toolchain selection.
    Toolchain (ARMv5te gcc-4.1.2,Multi-lib,gnueabi/glibc-2.5-nptl-3)
(-O2 -fsigned-char -msoft-float) Enter any CFLAGS for gcc/g++
--- Choose your bootloader for U-Boot
    bootloader (u-boot)  --->
--- Choose your board
--- Choose your Kernel
    kernel (Linux 2.6.31-imx)  --->
[ ] Always rebuild the kernel
[ ] Produce cscope index
[*] Include kernel headers
[ ] Configure the kernel
[*] Leave the sources after building
[ ] Build mfg firmware (NEW)
--- Package selection
    Package list  --->
--- Target System Configuration
    Options  --->
--- Target Image Generation
    Options  --->
v(+)
```

Most items in the main configuration screen can be ignored unless you have a specific reason to modify them. Two exceptions are items related to the kernel: **Always rebuild the kernel** and **Configure the kernel**. If you enable **Configure the kernel**, then the kernel configuration screen will be shown the next time the kernel is built.

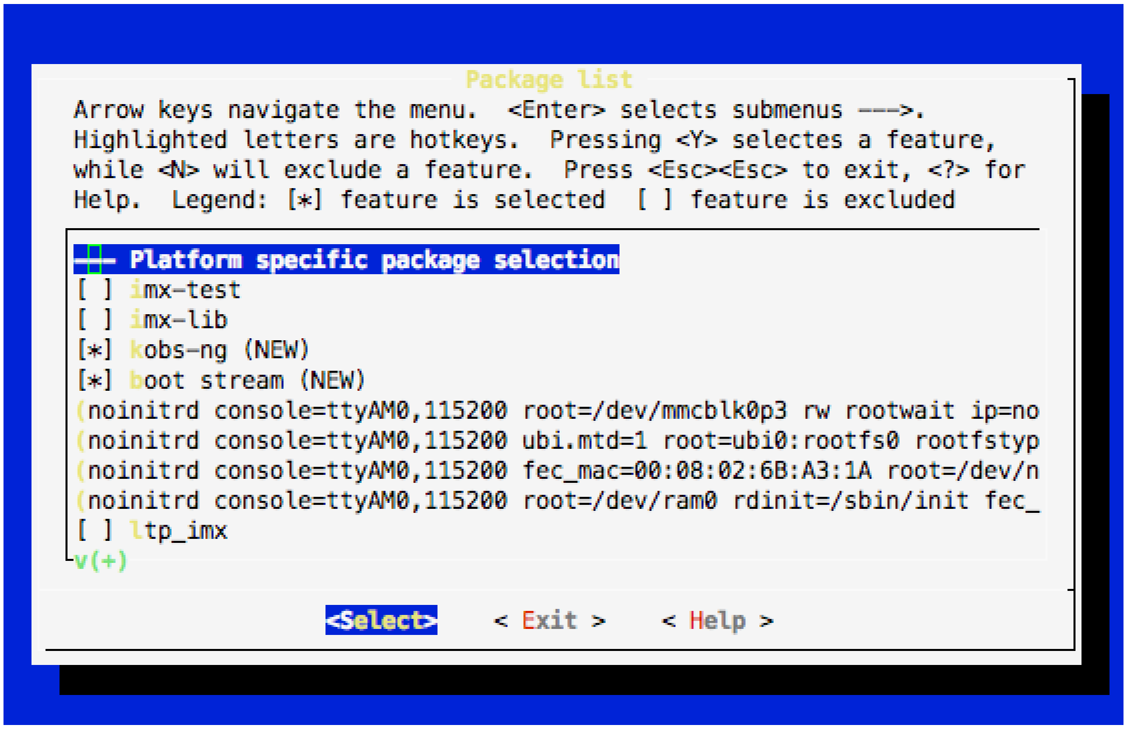
Another important item on the main screen is **Options**. Selecting it will bring up the following secondary configuration screen.



```
Arrow keys navigate the menu.  <Enter> selects submenus --->.
Highlighted letters are hotkeys.  Pressing <Y> selectes a feature,
while <N> will exclude a feature.  Press <Esc><Esc> to exit, <?> for
Help.  Legend: [*] feature is selected  [ ] feature is excluded

(mx28evk) target hostname
[*] boot up with a tty and login
(::respawn:/sbin/getty -L ttyAM0 115200 vt100) Enter your inittab sta
()  load these modules at boot
[*] start udevd
[ ] start networking
[ ] set the system time at startup
[ ] start syslogd/klogd
[*] start inetd
()    Enter command line arguments for inetd startup
v(+)
```

ℹ You do not actually have to enable **start networking** in order to boot from NFS or using networking on the target.

Finally, you can modify settings for individual packages and choose which packages to install by selecting the **Package list** item in the main screen.



```
Arrow keys navigate the menu.  <Enter> selects submenus --->.
Highlighted letters are hotkeys.  Pressing <Y> selectes a feature,
while <N> will exclude a feature.  Press <Esc><Esc> to exit, <?> for
Help.  Legend: [*] feature is selected  [ ] feature is excluded

--- Platform specific package selection
[ ] imx-test
[ ] imx-lib
[*] kobs-ng (NEW)
[*] boot stream (NEW)
(noinitrd console=ttyAM0,115200 root=/dev/mmcblk0p3 rw rootwait ip=no
(noinitrd console=ttyAM0,115200 ubi.mtd=1 root=ubi0:rootfs0 rootfstyp
(noinitrd console=ttyAM0,115200 fec_mac=00:08:02:6B:A3:1A root=/dev/n
(noinitrd console=ttyAM0,115200 root=/dev/ram0 rdinit=/sbin/init fec_
[ ] ltp_imx
v(+)
```

ℹ The package list screen is also where you set the kernel command line options for the i.MX28 when booting Linux directly. When using U-boot, you set the kernel command line in U-boot itself. As shown in the above screen, there are four kernel command lines listed under the **boot stream** package. The first command line is the default, while the other three are optional ones selected by pressing certain keys during system boot. To edit a command line, move the cursor over it and press return.

Run the image on the target system

After you have completed a build by using LTIB, you will have a target root filesystem in the `rootfs` directory inside the LTIB install directory. For instance, if you installed LTIB in your home directory, there will be a `~/ltib/rootfs` directory containing the target rootfs.

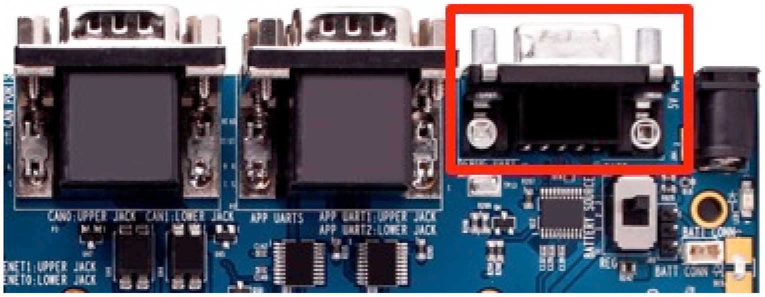
Inside the `~/ltib/rootfs/boot` directory, you will find a set of .sb files. These are the boot streams that the i.MX28 ROM interprets to boot the system.

Boot stream file	Description
`imx28_ivt_linux.sb`	Linux boot stream
`imx28_ivt_uboot.sb`	U-boot boot stream

The Linux boot stream contains only the kernel itself. The rootfs is stored separately on the boot media, or on an NFS server.

U-boot is the open source bootloader supported by Freescale for use with the i.MX28. Note that you only need to use U-boot for development purposes, especially to boot over the network using TFTP/NFS. For the boot media supported directly by the i.MX28 ROM, the ROM itself acts as the bootloader.

To monitor console output from the target, first connect your host computer to the debug UART on the EVK board. You can use either a standard serial port or a USB-to-serial interface. The debug UART is the rightmost DB9 connector when viewing the EVK from top. Finally, run a serial terminal such as `minicom` for Linux or HyperTerminal for Windows. Set the serial port to 115200 baud, 8N1.



The steps below assume you have configured your development host using the standard system setup script included with the BSP. If this is not the case, some of the paths below may be different for you.

Boot from SD/MMC

The first step is to put a boot image onto your SD/MMC card using the method described below.

Imaging with Linux

The BSP includes the `mk_mx28_sd` shell script that makes it very easy to place the build output onto an SD/MMC card. The script will write the selected boot stream, the rootfs, and a 32MB FAT32 partition to the card.

The `mk_mx28_sd` script should always be run from the LTIB install directory. It knows how to find the boot stream files and rootfs from this directory.

Before running the script, you need to determine the device for your SD/MMC card. By default, the script will not execute with the card already mounted, to prevent any possibility of overwriting a non-removable volume such as a hard drive. So, if you don't already know the device for your card, you must insert the card and determine the device, then remove the card before running the script.

Before inserting the card, run this command to see all available SCSI devices:

```
% ls /dev/sd?
```

Now insert the card, and run the above command again. The SD/MMC card will be the new `/dev/sdX` device in the output (where the X is a lower-case letter in alphabetic sequence). This is the device name that you pass to `mk_mx28_sd`. Before running the script, be sure to remove the card. Once you already know your card's device, updating the card contents with a new build will be much faster.

To place Linux onto the SD/MMC card:

```
% cd ~/ltib
% mk_mx28_sd /dev/sdX
```

To place U-boot onto the SD/MMC card, add the `-u` option to the command line:

```
% cd ~/ltib
% mk_mx28_sd -u /dev/sdX
```

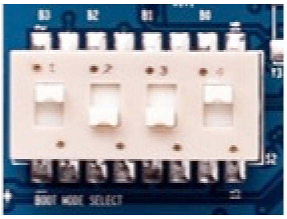
After you invoke the script, it will present a prompt for you to insert your SD/MMC card. Once you have done so, type "yes" to have the script continue and format the card.

Boot the card

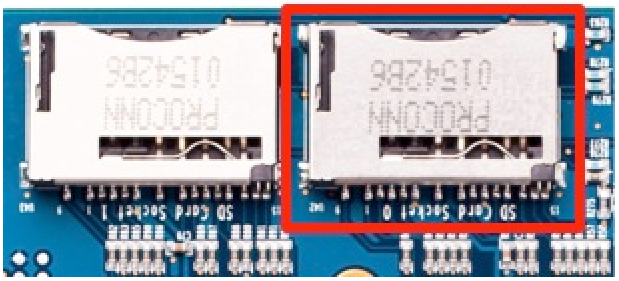
To boot from the SD/MMC card on which you placed Linux or U-boot, first make sure the i.MX28 EVK board is powered off.

Next, set the boot mode on the i.MX28 EVK board for MMC/SD on SSP0. The boot mode DIP switch settings are shown in the table below.

Boot mode	Port	BM3	BM2	BM1	BM0
SD/MMC master	SSP0	1	0	0	1



Then, insert the SD/MMC card into the SSP0 card socket. This is the rightmost socket on the bottom of the EVK board.



Finally, connect power to the board and press the Power button. You will see serial output over the debug UART if you have a terminal program running.

Boot from TFTP/NFS

When developing your application, booting from the network provides a much faster development cycle than alternatives such as copying the built kernel and rootfs to an SD/MMC card. This prevents you from having to repeatedly insert and remove a card and move it between your host and target.

Prepare host

These steps make the build output available so the bootloader and target kernel can access it on the network.

Get the server's address

Later, when you configure U-boot on the target, you will need to know the IP address of your TFTP and NFS server. The following shows the command to use and example output.

```
% ip addr show dev eth0
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UNKNOWN qlen 1000
    link/ether 00:0c:29:1c:3d:b9 brd ff:ff:ff:ff:ff:ff
    inet 10.81.4.235/22 brd 10.81.7.255 scope global eth0
    inet6 fe80::20c:29ff:fe1c:3db9/64 scope link
       valid_lft forever preferred_lft forever
```

This assumes that you have wired Ethernet connected to your host, and are not using WiFi, for instance.

Link kernel image

Create a symbolic link for the kernel image in the rootfs into the TFTP directory.

```
% cd /tftpboot
% ln -s ~/ltib/rootfs/boot/uImage
```

Alternatively, you could simply copy the `uImage` file into `/tftpboot`. However, using a symbolic link will make it so you don't have to copy the kernel every time you run a build.

Setup NFS

These commands will create a link to the `rootfs` directory created by LTIB in order to make the rootfs accessible via NFS from the target.

```
% cd /tools
% rm -rf rootfs
% ln -s ~/ltib/rootfs
% sudo exportfs -a
```

The final `exportfs` command above is to make certain the NFS server is aware of the directory you want to serve. Once you have NFS set up for a particular installation of LTIB, you don't have to modify anything to serve the changes between builds.

Boot into U-boot

Because the i.MX28 ROM does not directly support network booting, you must first boot the target into U-boot. From there, you can download the kernel image via TFTP and execute it. Then the kernel will mount and load the rootfs via NFS.

The first step is to prepare an SD/MMC card with the U-boot boot stream using the instructions in the "Boot from SD/MMC" section above. Be sure to pass the `-u` option to `mk_mx28_sd` so that U-boot is written to the card and not the Linux kernel.

Now follow the steps above for booting from the card with U-boot on it.

When it boots, U-boot will print output similar to the following on the serial terminal:

```
U-Boot 2009.08 (Aug 09 2010 - 18:49:32)

Freescale i.MX28 family
CPU:   454 MHz
BUS:   151 MHz
EMI:   166 MHz
DRAM:  128 MB
MMC:   IMX_SSP_MMC: 0, IMX_SSP_MMC: 1
In:    serial
Out:   serial
Err:   serial
Net:   got MAC address from IIM: 00:04:9f:00:27:22
FEC0
Hit any key to stop autoboot:  0
MX28 U-Boot >
```

When you see the "Hit any key to stop autoboot" message, press return in the serial terminal. U-boot will now show its prompt. If you don't press a key at this stage and let the countdown reach 0, U-boot will execute the contents of the `bootcmd` variable stored in its settings.

ℹ The first time you boot from a freshly imaged card, U-boot may report an error with its stored settings and not show the "Hit any key" message. This is normal and will be rectified once you save the settings.

The entire sequence of booting, and all configuration settings for U-boot are controlled through its environment variables. Normally, these variables are loaded from storage on the SD/MMC card. After reaching the U-boot prompt the first time, you need to set a number of variables that U-boot uses to identify the server and set the kernel command line.

Enter the following commands shown below. The `serverip` variable should be set to the IP address of the host server.

```
MX28 U-Boot > setenv bootargs 'console=ttyAM0,115200n8'
MX28 U-Boot > setenv bootcmd 'run bootcmd_net'
MX28 U-Boot > setenv bootdelay 5
MX28 U-Boot > setenv baudrate 115200
MX28 U-Boot > setenv serverip [Host IP]
MX28 U-Boot > setenv netmask 255.255.255.0
MX28 U-Boot > setenv bootfile uImage
MX28 U-Boot > setenv loadaddr 0x42000000
MX28 U-Boot > setenv nfsroot /tools/rootfs
MX28 U-Boot > setenv bootargs_nfs 'setenv bootargs ${bootargs} root=/dev/nfs ip=dhcp
nfsroot=${serverip}:${nfsroot} gpmi'
MX28 U-Boot > setenv bootcmd_net 'run bootargs_nfs; dhcp; bootm'
```

The configuration provided by these settings will cause U-boot to get an IP address via DHCP. It will then download the kernel from the TFTP server and execute it. When the kernel starts, it will also get an IP via DHCP and then mount the rootfs from the NFS server.

This table describes some of the most important and useful variables:

Variable	Description
`bootdelay`	The number of seconds that U-boot will wait for a keypress when it first boots before automatically booting the kernel.
`bootcmd`	The contents of this variable is executed when the `boot` command is run, or when automatically booting.
`bootargs`	The command line passed to the kernel.
`bootfile`	Name of the kernel file that is downloaded from the TFTP server.
`loadaddr`	Address where the kernel is stored in memory when downloaded from the TFTP server.
`serverip`	The IP address of the host server.

If you take a close look at some of the variables such as `bootargs_nfs`, you will notice that they actually contain commands to modify other variables. In the case of `bootargs_nfs`, it modifies the `bootargs` variable to add a number of kernel command line arguments.

To verify the values of the environment variables, use the `printenv` command as shown in the example here:

```
MX28 U-Boot > printenv
ipaddr=192.168.1.103
netmask=255.255.255.0
bootfile="uImage"
loadaddr=0x42000000
bootargs_mmc=setenv bootargs ${bootargs} root=/dev/mmcblk0p3 rw rootwait ip=dhcp fec_mac=${ethaddr}
bootcmd_mmc=run bootargs_mmc; mmc read 0 ${loadaddr} 100 3000; bootm
ethact=FEC0
bootargs=console=ttyAM0,115200n8
bootcmd=run bootcmd_net
bootdelay=6
baudrate=115200
serverip=10.81.4.117
nfsroot=/tools/rootfs
bootargs_nfs=setenv bootargs ${bootargs} root=/dev/nfs ip=dhcp nfsroot=${serverip}:${nfsroot} gpmi
bootcmd_net=run bootargs_nfs; dhcp; bootm
stdin=serial
stdout=serial
stderr=serial
ver=U-Boot 2009.08 (Aug 09 2010 - 18:49:32)

Environment size: 616/130044 bytes
```

Once all variables are set to the correct values, you can save the entire environment to the SD/MMC card:

```
MX28 U-Boot > saveevn
Saving Environment to MMC...
Writing to MMC(0)... done
```

To boot into Linux from the U-boot command line, simply run the `boot` command. Alternatively, you can reset the EVK and wait for it to automatically boot.