U-Boot for Freescalei.MX28

U-Boot has been ported to the   This document describes how to load, run and build U-Boot images on the board(s).

1 Overview

The U-Boot utility is a multi-platform, open-source, universal boot-loader with comprehensive support for loading and managing boot images, such as the Linux kernel. It supports the following features:

- Network download: TFTP, BOOTP, DHCP, NFS
- Serial download: s-record, binary (via Kermit)
- Memory utilities: copy, dump. crc, check, mtest
- Boot from disk: raw block, ext2, fat, reiserfs
- Interactive shell: choice of simple or "busybox" shell with many scripting features

For more information on U-Boot, refer to http://www.denx.de/wiki/U-Boot/WebHome.

2 Board Dip Switches Setup

Be sure to follow the switch settings required for the board in order to have U-Boot up and running properly.

2.1 i.MX28 Switch Settings

EVK board uses switch S2 to select boot mode. B0, B1, B2, B3 are labelled on the PIN location of switch.

Table 2.1.0.1: i.MX28 Boot Mode Switch Settings

Boot Mode	B3	B2	B1	B0
USB0	0	0	0	0
I2C0	0	0	0	1
SPI2	0	0	1	0
SPI3(Flash)	0	0	1	1
GPMI(NAND)	0	1	0	0
SPI3(EEPROM)	1	0	0	0
SSP0(SD0)	1	0	0	1
SSP1(SD1)	1	0	1	0

3 Image Install Instructions

3.1 Booting the system with an NFS filesystem

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

Setup boot arguments:

```
U-Boot > setenv bootargs 'console=ttyAM0,115200n8'
```

Setup boot command:

```
U-Boot > setenv bootcmd 'run bootcmd_net'
```

Setup NFS boot arguments:

```
U-Boot > setenv bootargs_nfs ‘setenv bootargs ${bootargs} root=/dev/nfs
ip=dhcp nfsroot=${serverip}:${nfsroot} fec_mac=00:04:9F:00:EA:D7’
```

Setup net boot command:

```
U-Boot > setenv bootcmd_net 'run bootargs_nfs; dhcp; bootm'
```

Setup tftp server IP:

```
U-Boot> setenv serverip 10.193.100.158
```

Setup board mac address:

```
U-Boot> setenv ethaddr  00:04:9F:00:EA:D7
```

Setup board IP address:

```
U-Boot> setenv ipaddr   10.193.102.93
```

Setup kernel name:

```
U-Boot> setenv kernel   uImage
```

Setup rootfs path:

```
U-Boot> setenv nfsroot  /data/rootfs_home/rootfs
```

Setup boot delay:

```
U-Boot> setenv bootdelay 2
```

Setup baudrate:

```
U-Boot > setenv baudrate 115200
```

Setup netmask:

```
setenv netmask 255.255.255.0
```

Setup boot file:

```
setenv bootfile uImage
```

Save configurations:

```
U-Boot> saveenv
```

Note:

> *To avoid a conflict of IP address, the* `dhcp` *command can be used for a valid IP address.*

Reset board and kernel will be launched.

3.2 Booting from MMC

3.2.1 Create a u-boot .sb file

Pls follow the following steps:

1. Get imx-bootlets source package and decompress it.

   ```
   tar -zxvf imx-bootlets-src-xx.xx.xx.tar.gz
   ```

2. Copy elftosb2 tool to linux-bootlets directory.

   ```
   cp elftosb2 ./
   ```

3. Add following text in linux_prep/cmdlines/iMX28_EVK.txt

```
Ramdisk: noinitrd console=ttyAM0,115200 root=/dev/ram0 rdinit=/sbin/init
fec_mac=00:08:02:6B:A3:1A
NFS: noinitrd console=ttyAM0,115200 fec_mac=00:08:02:6B:A3:1A root=/dev/nfs
nfsroot=10.193.100.213:/data/rootfs_home/rootfs_mx28 rw ip=dhcp rootwait
SD: noinitrd console=ttyAM0,115200 root=/dev/mmcblk0p3 rw rootwait
fec_mac=00:08:02:6B:A3:1A
```

4. Build out .sb file

```
      make CROSS_COMPILE=/opt/freescale/usr/local/gcc-4.1.2-glibc-2.5-nptl-3/
arm-none-linux-gnueabi/bin/arm-none-linux-gnueabi- BOARD=iMX28_EVK
```

Then you can get an imx28_uboot.sb file.

3.2.2 Create a bootable mmc card

There are two methods to create a bootable mmc card.

1. Use cfimager.exe tool.

- Copy imx28_uboot.sb to Windows.
- Insert a mmc card into an card reader and insert the card reader to computer usb port.
- Run following cmd in Windows command console:

```
cfimager.exe -a -f imx28_uboot.sb -d <removed disk number "X" ,such as H>
```

2. MX28 u-boot provide two new command to create a bootable mmc card, “mxs_mmc format” and “mxs_mmc install“.

- Set Boot Mode dips to 0000.
- Plug 5v power to board. connect USB between board and PC. press Power key. PC should discover an HID USB device.
- Run following cmd in Windows and board should boot up then:

```
sb_loader.exe /f imx28_uboot.sb
```

- Insert an mmc card and run format and install command.

```
MX28 U-Boot > mxs_mmc format 0
WARN: Data on card will get lost with format.
Continue? (y/n)
Done.
MX28 U-Boot > tftpboot imx28_uboot.sb
Using FEC0 device
TFTP from server 10.193.100.211; our IP address is 10.193.102.101; sending
through gateway 10.193.102.254
Filename 'imx28_uboot.sb'.
Load address: 0x42000000
Loading: FEC: Link is down 7809
##########
done
Bytes transferred = 136032 (21360 hex)
MX28 U-Boot > mxs_mmc install 0 0x42000000 0x21360 sbImage
Done: 266 (10a hex) sectors written at 3825666 (3a6002 hex)
MX28 U-Boot >
```

3.2.3 Booting from MMC

Set Boot Mode dips to 1001, insert card, plug 5v power and press power key. Board should boot up now.

```
U-Boot 2009.08-00091-gaca6f56-dirty (Mar 15 2010 - 18:05:56)

Freescale i.MX28 family
CPU:   240 MHz
BUS:   120 MHz
EMI:   166 MHz
DRAM:  128 MB
MMC:   IMX_SSP_MMC: 0
In:    serial
Out:   serial
Err:   serial
Net:   FEC0
Hit any key to stop autoboot:  0
MX28 U-Boot >
```

4 Configuring U-Boot

By default, U-Boot is configured to display the command prompt and receive serial keyboard input on certain UART ports with 115,200-8-N-1 settings.

The system configuration also needs to be set using the `setenv` command. The following is one example of how to set it up by assigning a static IP address to the board. If the network supports BOOTP/DHCP and one wants to use it, set “`true`” when prompted with “`Use BOOTP for network configuration`” in the following example and skip the configurations for `Gateway IP address`, `Local IP address` and `Local IP address mask`. Note that the new configuration doesn’t take effect until the board is reset (reset button or `reset` command).

```
U-Boot> setenv serverip 10.193.100.158
U-Boot> setenv ethaddr  00:04:9F:00:EA:D7
U-Boot> setenv ipaddr   10.193.102.93
U-Boot> setenv kernel   uImage
U-Boot> setenv nfsroot  /data/rootfs_home/rootfs
U-Boot>
```

Each of the parameters should be modified for the specific network and usage. It is very important that the board specifics parameters be set correctly. These parameters are not used by the U-Boot but are used by some operating systems.

Note that some DHCP servers are not configured to support BOOTP requests and as a result a static configuration will be required.

The board specifics parameter is a bit mask of board options. At this time, there are no options and board specifics should be 0.

The configuration can be listed:

```
U-Boot> printenv
bootdelay=3
baudrate=115200
loadaddr=0x42000000
netdev=eth0
ethprime=FEC0
U-Boot_addr=0xa0000000
U-Boot=u-boot.bin
bootargs_base=setenv bootargs console=ttymxc0,115200
bootargs_nfs=setenv bootargs ${bootargs} root=/dev/nfs ip=dhcp
nfsroot=${serverip}:${nfsroot},v3,tcp
bootcmd=run bootcmd_net
bootcmd_net=run bootargs_base bootargs_nfs; tftpboot ${loadaddr} ${kernel};
bootm
prg_U-Boot=tftpboot ${loadaddr} ${U-Boot}; protect off ${U-Boot_addr}
0xa003ffff; erase ${U-Boot_addr} 0xa003ffff; cp.b ${loadaddr} ${U-Boot_addr}
${filesize}; setenv filesize; saveenv
ethact=FEC0
ethaddr=00:04:9F:00:EA:D7
bootargs=console=ttymxc0,115200 root=/dev/nfs ip=dhcp nfsroot=:/opt/eldk/
arm,v3,tcp
ipaddr=10.193.102.93
serverip=10.193.100.158
nfsroot=/data/rootfs_home/rootfs
```

```
kernel=uImage.r450
stdin=serial
stdout=serial
stderr=serial

Environment size: 830/131068 bytes
Boot>
```

5 Using U-Boot

OS images can be downloaded using U-Boot via Ethernet or the serial port. The downloaded images can be immediately decompressed into SDRAM and executed or they can be stored into flash and decompressed into SDRAM at a later time. All of these instructions assume that U-Boot has been installed onto the board and configured.

5.1 Serial Download

To download an image through serial, the terminal should support y moderm protocol, e.g. HyperTerminal in windows. Issue the following command under U-Boot prompt:

```
loady
```

U-Boot now is ready to receive data. To send a file, click on "*Transfer -> Send File -> Ymodem (under Protocol) -> Send...*", choose the file to download.

5.2 Ethernet Download

Note that TFTP is not a reliable protocol and downloads may sometimes fail (with no error message) on busy networks. Always check the byte count of the download to make sure the download is complete. Besides, some EVB/ ADS boards don't have the correct MAC address programmed in the EEPROM which is used by the Ethernet controller.

To download a file via the Ethernet:

1. Set up and configure a TFTP server.
   ```
   setenv ethaddr <board_mac_addr>
   setenv serverip <Server_IP_Addr>
   setenv ipaddr <Board_IP_Addr>
   ```
   (This address can be retrived by command dhcp)
2. Configure the server to load files from the appropriate directories or copy the desired image to the server directory.
3. Test the network configuration:
   ```
   ping <TFTP server IP address>
   ```

The following discuss how to download and execute different kinds of images.

- To download and execute an uncompressed image (check that the number of bytes downloaded is correct):
  ```
  tftpboot ${loadaddr} <file name>
  bootm
  ```

6 Building U-Boot

6.1 Building U-Boot from source

Building U-Boot can be achieved under either Windows using Cygwin or Linux. Here we only supply files for building under Linux. All the following files are included in release package.

Table 6.1.0.1: Resources needed for Building U-Boot

Resource	Description	Source
`u-boot-2009.08.tar.bz2`	U-Boot base source for Linux	pkgs/
`tc-fsl-x86lnx-armeabi-nptl-4.1.2-3.i386.rpm`	Toolchain needed to build U-Boot under Linux	included
`u-boot-v2009.08-imx_10.05.02.tar.bz2`	Freescale U-Boot source patch and binary files.	pkgs/

Unzip the Freescale U-Boot release package `L2.6.31_10.05.02_source.tar.gz.`You will find the U-Boot source patch files and `u-boot-2009.08.tar.bz2` inside `L2.6.31_10.05.02_source/pkgs` .

The following describes how to setup the build environment first and then gives instructions on how to build the U-Boot images.

6.1.1 Setting up U-Boot Build Environment under Linux

To simplify the built process, it is recommended to use Freescale's pre-packaged tools to build U-Boot. The following has been verified working under Redhat 9.0 and Ubuntu 9.10 Linux distribution releases.

1. Locate `u-boot-2009.08.tar.bz2` and `u-boot-v2009.08-imx_10.05.02.tar.bz2` from this package.
2. Under Linux, with root user privilege, install `tc-fsl-x86lnx-armeabi-nptl-4.1.2-3.i386.rpm.` A more simple way is to run ltib in release firstly which will install the toolchain for host server.

6.1.2 Generating U-Boot Image

After the tools are setup properly, follow these instructions to build the U-Boot image:

1. Decompress the `u-boot-2009.08.tar.bz2` base line source code into some directory, etc, your home directory. There should be a `packages` directory under your home if this is done correctly.
   ```
   $ tar jxvf u-boot-2009.08.tar.bz2
   ```
2. Change into the `u-boot-2009.08` subdirectory and apply the patches from the `u-boot-v2009.08-imx_09.12.00.tar.bz2`:
   ```
   $ cd u-boot-2009.08
   ```
   Under Linux, do:
   ```
   $ tar jxvf u-boot-v2009.08-imx_10.05.02.tar.bz2
   $ ./patches/patch-U-Boot.sh
   ```
3. Check for a non-zero return value of the patch command (`echo $?`), or check to see if there are any rejected patch fragments (`find . -name '*.rej' -print`). Either of these findings would indicate that the patch did not apply properly.
4. Build U-Boot. The following commands can be done from any directory and it is recommended to create a separate directory from the source in order to have a clean build.

   For i.MX28, do
   ```
   make CROSS_COMPILE=/opt/freescale/usr/local/gcc-4.1.2-glibc-2.5-nptl-3/
   arm-none-linux-gnueabi/bin/arm-none-linux-gnueabi- mx28_evk_config
   ```

```
            make CROSS_COMPILE=/opt/freescale/usr/local/gcc-4.1.2-glibc-2.5-
nptl-3/arm-none-linux-gnueabi/bin/arm-none-linux-gnueabi-
```

This creates the U-Boot image (`u-boot.bin`) under U-Boot source directory. This image can run from either SDRAM or flash.

7 U-Boot Commands

The following commands are supported for various Freescale boards. The intent is to provide a user-friendly and easy-to-use interface for setting up or using certain hardware features such as NAND flash, MAC address in the EEPROM for CS8900A Ethernet controller and reading/writing e-fuses, etc. This section shows these commands and the corresponding help messages.

7.1 Run

Run an commands in the environment variable(s) 'var'.

```
      run [env_variable]
```

7.2 Net commands

Boot image via network using BOOTP/TFTP protocol

```
      bootp [loadAddress] [[hostIPaddr:]bootfilename]
```

Boot image via network using TFTP protocol

```
      tftpboot [loadAddress] [[hostIPaddr:]bootfilename]
```

Boot image via network using DHCP/TFTP protocol

```
      dhcp [loadAddress] [[hostIPaddr:]bootfilename]
```

Boot image via network using RARP/TFTP protocol

```
      rarpboot [loadAddress] [[hostIPaddr:]bootfilename]
```

7.3 Boot commands

Boot application image from memory

```
      bootm [addr [arg ...]]
```

Boot image via network using TFTP protocol

```
      tftpboot [loadAddress] [[hostIPaddr:]bootfilename]
```

Boot image via network using DHCP/TFTP protocol

```
      dhcp [loadAddress] [[hostIPaddr:]bootfilename]
```

Boot image via network using RARP/TFTP protocol

```
      rarpboot [loadAddress] [[hostIPaddr:]bootfilename]
```

Format MMC device

```
      mxs_mmc format <dev num>
```

Install image to MMC card

```
      mxs_mmc install <dev num> addr size [sbImage|uImage|rootfs]
```

8 Frequently Asked Questions

8.1 Why there is no U-Boot prompt on an internal UART port?

Double check the dip switch settings as mentioned earlier in this document.