VEDA Solutions
inspiring linux learning

U-Boot

- U-boot is an opensource universal bootloader for embedded [know as Das U-Boot].
- Loaded by FSBL from DDR, and it is responsible to load an another application through a serial,
        an ethernet, and Flash memories.
- Performed board, CPU and particular flash specific initialization & configurations.
- Also parses different types of filesystems on many types of storage devices.
- Responsible to pass some setoff parameters called boot parameters to kernel.

U-boot Features :
Source tree
        Well structured tree as like Linus' tree, called ad Denx tree
Autoboot
        Will automatically boot the system on power up or reset of the board
Environment variables
        Env. variables can set, save and even print with respective commands.
Networking
        Supports all possible n/w commands like ping, dhcp, tftp and nfs
O/S loading
        Supports variety of commands to load an O/S
Flash support
        Can read parallel NOR, NAND, SD/MMC, serial NOR, USB flashes
Serial download
        Files can be loaded through serial via loady, loadb

U-Boot commands :
- U-Boot has a set of built-in commands for booting the system, managing memory,
        and updating an embedded system's firmware.
- For a complete list and brief descriptions of the built-in commands,
        he U-Boot prompt, enter either of these commands
        MINI2440 # help (or) ?

Important commands:
        - Flash information (NOR and SPI flash)
                        flinfo
        - NAND flash information
                        nand info
        - Board info structure.
                        bdinfo
        - Shows NAND bad blocks
                        nand bad
        boot   - runs the default boot command, stored in bootcmd
        bootm <address> - starts a kernel image loaded at the given address in RAM
        fatload- loads a file from a FAT filesystem to RAM

tftp	- loads a file from the network to RAM	
ping	- to test the network	

cp[.b, .w, .l]	source target count	- Copies memory contents from address source to target for as many count bytes, words, or long words.
nand[erase, read, write]		- nane read/write commands

Environment variables :

- U-Boot can be configured through environment variables, which affect the behavior of the different commands.
- Environment variables are loaded from ash to RAM at U-Boot startup, can be modified and saved back to flash

Commands to manipulate environment variables:

printenv	- shows all variables
printenv <variable-name>	- shows the value of one variable
setenv <variable-name> <variable-value>	- changes the value of a variable, only in RAM.
saveenv	- saves the current state of the environment to flash

Important U-Boot env variables :

bootcmd	- Defines a command string that is automatically executed when the initial countdown is not interrupted.Executed only when the bootdelay variable is also defined.
bootdelay	- Seconds to wait before running the automatic boot process in bootcmd.
bootargs	- contains the arguments passed to the Linux kernel
serverip	- the IP address of the server that U-Boot will contact for network related
commands .	
ipaddr	- the IP address of target on which u-boot running
ethaddr	- the MAC address, can only be set once
autostart	- if yes, U-Boot starts automatically an image that has been loaded into memory.
baudrate	- a decimal number that selects the console baudrate (in bps).
mtdparts	- This variable allows to share a common MTD partition scheme between U-Boot and the Linux kernel.
hostname	- Target hostname

Building u-boot for mini2440 board t:

- **Download u-boot source**

  git clone git://repo.or.cz/u-boot-openmoko/mini2440.git

- Configure u-boot for target board

  veda@linux # cd mini2440
  veda@linux # make mini2440_config

- Building u-boot for target board

veda@linux **# make CROSS_COMPILE=arm-linux-**
NOTE: export cross compilation toolchain path **PATH=$PATH:$(PATH_OF_CROSSTOOL)**

- After a successful compilation, you should get the following U-Boot images.

       1. **u-boot.bin**   - is a raw binary image
       2. **u-boot**   - is an image in ELF binary format
3. **u-boot.srec**   - is in Motorola S-Record format

```
root@linux:~/elinux/workspace/u-boot-mini2440# ls
api          COPYING   examples      MAKEALL    README           tools
arch         CREDITS   fs            Makefile   rules.mk         u-boot
board        disk      helper.mk     mkconfig   snapshot.commit  u-boot.bin
boards.cfg   doc       include       nand_spl   spl              u-boot.lds
common       drivers   lib           net        System.map       u-boot.map
config.mk    dts       MAINTAINERS   post       test             u-boot.srec
root@linux:~/elinux/workspace/u-boot-mini2440#
```