# SheevaPlug Development Kit - Reference Board Bring-up

This document provides the steps to program the boot flash using an external programmer such as JTAG debugger to bring-up a bare SheevaPlug reference board.

Following files are used which are in the *"SheevaPlug U-Boot"* folder:

1. **kw_dimm400.cmm** – The Lauterbach script to access the initial debug console.
2. **u-boot-rd88f6281Sheevaplug** – The U-Boot elf object file that is loaded on the DRAM by JTAG
3. **u-boot-rd88f6281Sheevaplug_400db_nand.bin** – The binary U-Boot image used to boot from NAND flash

## Procedure

1. Connect the Lauterbach JTAG debugger to the SheevaPlug reference board with Marvell Kirkwood chip 6281(A0).
2. Execute "TRACE32" application.
3. Goto "File" chose "Run Batchfile" point it to the "*kw_dimm400.cmm*" batch file (LauterBach script).
4. Goto "File" chose "Load" point it to the LE "*u-boot-rd88f6281SheevaPlug*" the u-boot elf file
5. Enter command "*R.S PC 670000*"
6. Enter command "*D.S SD:0xD0010000 %LE %LONG 0x01111111*"
7. Enter command "*D.S SD:0xD0010008 %LE %LONG 0x551100*"
8. Goto "Run" then "Go" to execute the loaded u-boot elf file
9. In the board console you will get u-boot prompt
10. Set your IP address
    *setenv ipaddr 192.168.1.XXX*
11. Set your server IP address. Please note that the IP address of the SheevaPlug system and the TFTP server should be in the same subnet.
    *setenv serverip 192.168.1.YYY*
12. Point the TFTP server on the PC to the directory containing the U-Boot files.
13. From console type u-boot command
    *bubt u-boot-rd88f6281Sheevaplug_400db_nand.bin*
14. U-Boot will flash the binary image on the NAND flash found in the system.
15. Type 'reset' in the cosole to restart the device. The new u-boot will be activated and used.

# *Appendix A*

The "mkimage" utility is used to generate a kernel image for U-Boot.

The mkimage utility encapsulates a compressed "uImage" Linux kernel image with header information, CRC32 checksum, etc, for use with the U-Boot bootloader.

mkimage can also be used to create ramdisk images for use with U-Boot, either separated from the Linux kernel image, or combined into one file.

mkimage encapsulates the images with a 64 byte header containing information about target architecture, operating system, image type, compression method, entry points, time stamp, CRC32 checksums, etc.

LINUX Sources, use "mkimage" as follows,

linux-2.6.22.18/arch/arm/boot/Makefile,
---

```
cmd_uimage = $(CONFIG_SHELL) $(MKIMAGE) -A arm -O linux -T kernel \
          -C none -a $(ZRELADDR) -e $(ZRELADDR) \
          -n 'Linux-$(KERNELRELEASE)' -d $< $@
```

mkimage --help
---

```
Usage: mkimage -l image
        -l ==> list image header information
      mkimage -A arch -O os -T type -C comp -a addr -e ep -n name -d
data_file[:data_file...] image
        -A ==> set architecture to 'arch'
        -O ==> set operating system to 'os'
        -T ==> set image type to 'type'
        -C ==> set compression type 'comp'
        -a ==> set load address to 'addr' (hex)
        -e ==> set entry point to 'ep' (hex)
        -n ==> set image name to 'name'
        -d ==> use image data from 'datafile'
        -x ==> set XIP (execute in place)
```