



SheevaPlug Development Kit - USB Flash Recovery from U-Boot

Rev 1.2



USB Flash Recovery from U-Boot

This document provides the instruction to boot Linux on the SheevaPlug from U-Boot using a USB flash key connected to the USB port on the SheevaPlug. The provided set of instructions allows the burning of the filesystem on to the NAND flash.

Following images and material are needed to run this process successfully:

Hardware:

1. USB Flash key with at least 512 MB capacity and ext2 or ext3 filesystem format.

Software:

1. **u-boot-rd88f6281Sheevaplug_400db_nand.bin** – U-Boot binary having the recovery feature enabled. The latest working binary image has been built on April 09, 2009 and is available on the marvell.com website.
2. **flashware.img** – Flashware image which is a concatenation of the ulmage and the squashed (squashfs format) minimal root filesystem.
3. **ulmage.sheeva.xxx** – Latest ulmage that can be used to boot up the kernel on the SheevaPlug.
4. **ubuntu-9.0.5.Release.jffs2** – Filesystem that is to be burned onto the NAND flash.

All the software images except the filesystem '**ubuntu-9.0.5.Release.jffs2**' have been included in the SheevaPlug_USB_Recovery.zip file. See Appendices A, B and C for more information.

1.0 Instructions

1. Copy the flashware image (flashware.img), ulmage (ulmage.sheeva.040309) to the USB key. Also, copy the filesystem image (ubuntu jffs2 or fedora jffs2 or ubuntu ubifs) to the USB key.
2. Upgrade the U-Boot on the plug with the U-Boot binary image in the SheevaPlug_UBoot.zip package file using the 'bubt' command. Select 'n' for the environment option. Reboot the plug.
3. After upgrading the U-Boot, stop the auto-boot to enter the U-Boot prompt. At the U-Boot prompt, set the following parameters:
 - a. `Marvell>> set loadaddr 0x2000000`
 - b. `Marvell>> set mtd1Size 0x300000 (setting mtd1 size to 3MB)`

(Note: If there are many bad blocks observed in mtd1 partition, the user can change the mtd1Size, which is the ulmage partition size, to say 0x400000 or 0x500000 etc. The default is 0x300000)
 - c. `Marvell>> set filesize 0x44ad000 (size of the provided file, flashware.img, in hex bytes)`
 - d. `Marvell>> saveenv`
 - e. `Marvell>> reset`



4. After reset, stop the auto-boot to enter the U-Boot prompt. Connect the USB key, containing the flashware.img, ulmage.sheeva.040309 and the FS images, to the plug. Type the command below:
 - a. *Marvell>> rcvr*
5. The plug should then boot up from the flashware.img image in the USB key. The logs of the boot-up process are shown in Appendix A.
6. At the Linux prompt, do the following:
 - a. *# cat /proc/mtd* → This should list all the 3 mtd partitions for the u-boot, ulmage and rootfs as shown below.
-sh-3.2# cat /proc/mtd
dev: size erasesize name
mtd0: 00100000 00020000 "u-boot"
mtd1: 00300000 00020000 "ulmage"
mtd2: 1fc00000 00020000 "root"
 - b. *# fdisk -l*
 - c. *# mount /dev/sda(USB key) /mnt*
 - d. *# cd /mnt*
 - e. *# flash_eraseall /dev/mtd1*
 - f. *# flash_eraseall -j /dev/mtd2*
 - g. *# nandwrite -p /dev/mtd1 ulmage.sheeva.040309*
 - h. *# nandwrite -p -q /dev/mtd2 <filesystem>*
 - i. *# reboot*
7. Again stop the auto-boot and enter the following at the U-Boot prompt:
 - a. *Marvell>> setenv bootcmd 'nand read.e 0x800000 0x100000 0x300000; bootm 0x800000'*
 - b. *Marvell>> setenv bootargs 'console=ttyS0,115200 mtdparts=nand_mtd:0x100000@0x00000(u-boot),0x300000@0x100000(ulmage)ro,0x1fc00000@0x400000(rootfs)rw root=/dev/mtdblock2'*
 - c. *Marvell>> saveenv*
 - d. *Marvell>> reset*



```
e=squashfs initrd=0x2300000,0x41ad000 ramdisk_size=67252 recovery=usb
serverip=0
.0.0.0
PID hash table entries: 2048 (order: 11, 8192 bytes)
Console: colour dummy device 80x30
Dentry cache hash table entries: 65536 (order: 6, 262144 bytes)
Inode-cache hash table entries: 32768 (order: 5, 131072 bytes)
Memory: 256MB 256MB 0MB 0MB = 512MB total
Memory: 448128KB available (3808K code, 258K data, 124K init)
Mount-cache hash table entries: 512
CPU: Testing write buffer coherency: ok
NET: Registered protocol family 16
```

CPU Interface

```
-----
SDRAM_CS0 ....base 00000000, size 256MB
SDRAM_CS1 ....base 10000000, size 256MB
SDRAM_CS2 ....disable
SDRAM_CS3 ....disable
PEX0_MEM ....base e8000000, size 128MB
PEX0_IO ....base f2000000, size 1MB
INTER_REGS ....base f1000000, size 1MB
NFLASH_CS ....base fa000000, size 2MB
SPI_CS ....base f4000000, size 16MB
BOOT_ROM_CS ....no such
DEV_BOOTCS ....no such
CRYPT_ENG ....base f0000000, size 2MB
```

```
Marvell Development Board (LSP Version KW_LSP_4.2.7_patch2)-- SHEEVA
PLUG Soc
: 88F6281 A0 LE
```

```
Detected Tclk 200000000 and SysClk 400000000
MV Buttons Device Load
Marvell USB EHCI Host controller #0: c088b600
PEX0 interface detected no Link.
PCI: bus0: Fast back to back transfers enabled
SCSI subsystem initialized
usbcore: registered new interface driver usbfs
usbcore: registered new interface driver hub
usbcore: registered new device driver usb
NET: Registered protocol family 2
Time: kw_clocksource clocksource has been installed.
IP route cache hash table entries: 16384 (order: 4, 65536 bytes)
TCP established hash table entries: 65536 (order: 7, 524288 bytes)
TCP bind hash table entries: 65536 (order: 6, 262144 bytes)
TCP: Hash tables configured (established 65536 bind 65536)
TCP reno registered
checking if image is initramfs...it isn't (bad gzip magic numbers);
looks like a
n initrd
Freeing initrd memory: 67252K
RTC registered
Use the XOR engines (acceleration) for enhancing the following
functions:
```



```
o RAID 5 Xor calculation
o kernel memcpy
o kernel memzero
Number of XOR engines to use: 4
cesadev_init(c00117e4)
mvCesaInit: sessions=640, queue=64, pSram=f0000000
Warning: TS unit is powered off.
MV Buttons Driver Load
squashfs: version 3.3 (2007/10/31) Phillip Lougher
squashfs: LZMA support for slax.org by jro
JFFS2 version 2.2. (NAND) © 2001-2006 Red Hat, Inc.
io scheduler noop registered
io scheduler anticipatory registered (default)
Serial: 8250/16550 driver $Revision: 1.90 $ 4 ports, IRQ sharing
disabled
serial8250.0: ttyS0 at MMIO 0xf1012000 (irq = 33) is a 16550A
serial8250.0: ttyS1 at MMIO 0xf1012100 (irq = 34) is a 16550A
RAMDISK driver initialized: 20 RAM disks of 67252K size 4096 blocksize
Loading Marvell Ethernet Driver:
o Cached descriptors in DRAM
o DRAM SW cache-coherency
o Single RX Queue support - ETH_DEF_RXQ=0
o Single TX Queue support - ETH_DEF_TXQ=0
o TCP segmentation offload enabled
o Receive checksum offload enabled
o Transmit checksum offload enabled
o Network Fast Processing (Routing) supported
o Driver ERROR statistics enabled
o Driver INFO statistics enabled
o Proc tool API enabled
o Rx descriptors: q0=128
o Tx descriptors: q0=532
o Loading network interface(s):
o eth0, ifindex = 1, GbE port = 0
o eth1, ifindex = 2, GbE port = 1

mvFpRuleDb (dc750000): 16384 entries, 65536 bytes
Intel(R) PRO/1000 Network Driver - version 7.3.20-k2-NAPI
Copyright (c) 1999-2006 Intel Corporation.
e100: Intel(R) PRO/100 Network Driver, 3.5.17-k4-NAPI
e100: Copyright(c) 1999-2006 Intel Corporation

Warning Sata is Powered Off
NFTL driver: nftlcore.c $Revision: 1.98 $, nftlmount.c $Revision: 1.41
$
NAND device: Manufacturer ID: 0xec, Chip ID: 0xdc (Samsung NAND 512MiB
3,3V 8-bi
t)
Scanning device for bad blocks
Bad eraseblock 832 at 0x06800000
mtd: bad character after partition (r)
3 cmdlinepart partitions found on MTD device nand_mtd
Using command line partition definition
Creating 3 MTD partitions on "nand_mtd":
0x00000000-0x00100000 : "uboot"
```



```
0x00100000-0x00400000 : "uImage"
0x00400000-0x20000000 : "rootfs"
ehci_marvell ehci_marvell.70059: Marvell Orion EHCI
ehci_marvell ehci_marvell.70059: new USB bus registered, assigned bus
number 1
ehci_marvell ehci_marvell.70059: irq 19, io base 0xf1050100
ehci_marvell ehci_marvell.70059: USB 2.0 started, EHCI 1.00, driver 10
Dec 2004
usb usb1: configuration #1 chosen from 1 choice
hub 1-0:1.0: USB hub found
hub 1-0:1.0: 1 port detected
USB Universal Host Controller Interface driver v3.0
usb 1-1: new high speed USB device using ehci_marvell and address 2
usb 1-1: configuration #1 chosen from 1 choice
usbcore: registered new interface driver usblp
drivers/usb/class/usblp.c: v0.13: USB Printer Device Class driver
Initializing USB Mass Storage driver...
scsi0 : SCSI emulation for USB Mass Storage devices
usbcore: registered new interface driver usb-storage
USB Mass Storage support registered.
mice: PS/2 mouse device common for all mice
i2c /dev entries driver
Linux telephony interface: v1.00
Marvell Telephony Driver:
mvBoardVoiceAssembleModeGet: TDM not supported(boardId=0x9)
assembly=-1,irq=-1
mp_check_config: Error, invalid voice assembly mode
md: linear personality registered for level -1
md: raid0 personality registered for level 0
md: raid1 personality registered for level 1
raid6: int32x1      97 MB/s
raid6: int32x2     123 MB/s
raid6: int32x4     125 MB/s
raid6: int32x8     111 MB/s
raid6: using algorithm int32x4 (125 MB/s)
md: raid6 personality registered for level 6
md: raid5 personality registered for level 5
md: raid4 personality registered for level 4
raid5: measuring checksumming speed
   arm4regs   : 1084.000 MB/sec
    8regs     :  754.800 MB/sec
   32regs     :  899.600 MB/sec
raid5: using function: arm4regs (1084.000 MB/sec)
device-mapper: ioctl: 4.11.0-ioctl (2006-10-12) initialised: dm-
devel@redhat.com
dm_crypt using the OCF package.
sdhci: Secure Digital Host Controller Interface driver
sdhci: Copyright(c) Pierre Ossman
mvsdmmc: irq =28 start f1090000
mvsdmmc: no IRQ detect
usbcore: registered new interface driver usbhid
drivers/hid/usbhid/hid-core.c: v2.6:USB HID core driver
Advanced Linux Sound Architecture Driver Version 1.0.14 (Thu May 31
09:03:25 200
7 UTC).
```




```
mvCLAudioCodecRegGet: Error while reading register!
mvCLAudioCodecInit: Error - Invalid Cirrus Logic chip/rev ID!
Error - Cannot initialize audio decoder.at address =0xff<6>ALSA device
list:
#0: Marvell mv88fx_snd ALSA driver
TCP cubic registered
NET: Registered protocol family 1
NET: Registered protocol family 17
md: Autodetecting RAID arrays.
md: autorun ...
md: ... autorun DONE.
RAMDISK: squashfs filesystem found at block 0
RAMDISK: Loading 67249KiB [1 disk] into ram disk... | / - \ | / -
\ | / - \ | / - \ | / - \ | / - \ | / - \ | / - \ | / -
\ | / - \ | / - \ | / - \ | / - \ | / - \ | / - \ | / -
.
.
.
\ | / - \ | / - \ | / - \ | / - \ | / - \ | / - \ | / -
\ | / - \ | / - \ | / - \ | / - \ | / - \ | / - \ | / -
\ | / - \ done.
VFS: Mounted root (squashfs filesystem) readonly.
Freeing init memory: 124K
init started: BusyBox v1.7.0 (2008-02-26 19:25:17 IST)
starting pid 289, tty '': '/etc/init.d/rcS'
starting pid 291, tty '': '/bin/sh'
-sh-3.2# scsi 0:0:0:0: Direct-Access      SanDisk  Cruzer                8.01
PQ: 0 AN
SI: 0 CCS
sd 0:0:0:0: [sda] 15682559 512-byte hardware sectors (8029 MB)
sd 0:0:0:0: [sda] Write Protect is off
sd 0:0:0:0: [sda] Assuming drive cache: write through
sd 0:0:0:0: [sda] 15682559 512-byte hardware sectors (8029 MB)
sd 0:0:0:0: [sda] Write Protect is off
sd 0:0:0:0: [sda] Assuming drive cache: write through
sda: unknown partition table
sd 0:0:0:0: [sda] Attached SCSI removable disk
sd 0:0:0:0: Attached scsi generic sg0 type 0

-sh-3.2#
-sh-3.2#
-sh-3.2#
-sh-3.2#
```




Appendix C

Building flashware.img

This appendix provides the steps to create the *flashware.img* for the USB recovery process. This section assumes that the user has already installed the squashfs utilities on the Linux host machine. You can download the latest utilities from:

<http://squashfs.sourceforge.net/>

Please make sure to build a ulmage with the following support enabled as default:

1. squashfs support
2. initrd support
3. RAMDISK support with the following
 - a. (20) Default number of RAM disks
 - b. (4096) Default RAM disk size (kbytes)
 - c. (4096) Default RAM disk block size (bytes) (NEW)

You can enable the support for the above features in the ulmage by using **'make menuconfig'** command after making the *'mv88f6281_defconfig'* configuration file, while building the kernel and the LSP. Refer to the document *'SheevaPlug Development Kit - Configuring the kernel with LSP for KW_A0_based device-Rev1.2.pdf'* for information in configuring and building the kernel and LSP for SheevaPlug.

Once the ulmage is created with the initrd, RAMDISK and squashfs support, do the following on the Linux machine.

1. Find the size of the ulmage. Usually, the ulmage file is around 2MB.
`# ls -al ulmage.sheeva.xxx`
2. For the recovery procedure, the mtd partition for ulmage is set to be 0x300000, 3MB or 3145728 bytes. Hence we need to create a ulmage.img file which is 3MB in size. Hence we need to use a pad file called ulmage.pad. The size of the ulmage.pad file is obtained by using the following equation:
size = (3145728 bytes) – <size of ulmage>
3. On the linux host, use the following command to create a ulmage.pad file.
`# dd if=/dev/zero of=ulmage.pad bs=(size) count=1`
4. Create the ulmage.img file using the cat command as below.
`# cat ulmage.sheeva.xxxx ulmage.pad >> ulmage.img`
5. Use the rootfsv1.0 from the SheevaPlug_Host_SWsupportPackageLinuxHost.zip to create a squashfs image 'rootfs.img'. The rootfs.img can be created by using the command as below.
`# mkfs.squashfs rootfsv1.0 rootfs.img -b 4096`
6. Concatenate the ulmage.img and rootfs.img image files to get flashware.img.
`# cat ulmage.img rootfs.img >> flashware.img`