Matthias Blaicher

Personal Records

Installing Debian on a Sheevaplug into Flash

July 24, 2012 Microcontroller, Personal, SheevaPlug, Work

Ok, so you are looking for a comprehensive guide on how to install a modern Debian system into the Flash memory of your Sheeva Plug? Your googling has paid of – look no further!

What we will do in this guide:

- Update the bootloader to a recent version which supports direct booting from ubifs partions. So you can just use the stock Debian kernel without any hassle or SD card.
- Install Debian to a ubifs filesystem on the Sheevaplug flash.

Not everything needed in this guide is covered here. It’s assumed you have prepared:

1. A terminal program of your choice which you can use to connect to the debug connection of your Sheeva Plug.
2. You have a working TFTP server for providing the new firmware etc. to your Sheeva. You can probably adapt this guide to use USB or SD cards, though.

Please note that **this guide is specific to the “normal” SheevaPlug**, that is the original SheevaPlug, the eSATA SheevaPlug and the Ionics Nimbus 100. Especially the u-boot update procedure will be different and you can brick your non-normal Sheeva that way!

Updating the bootloader

Let’s first update the bootloader to something useful i.e. a recent built of the vanilla u-boot bootloader. So connect to your Sheeva and stop the boot process by pressing the “any” key:

```
         __  __                      _ _
        |  \/  | __ _ _ ____   _____| | |
        | |\/| |/ _` | '__\ \ / / _ \ | |
        | |  | | (_| | |   \ V /  __/ | |
        |_|  |_|\__,_|_|    \_/ \___|_|_|
 _   _     ____              _
| | | |   | __ )  ___   ___ | |_
| | | |___|  _ \ / _ \ / _ \| __|
```

```
| |_| |___| |_) | (_) | (_) | |_
 \___/     |____/ \___/ \___/ \__|
  ** MARVELL BOARD: SHEEVA PLUG LE

U-Boot 1.1.4 (Jul 14 2009 - 06:46:57) Marvell version: 3.4.16

U-Boot code: 00600000 -> 0067FFF0  BSS: -> 006CF120

Soc: MV88F6281 Rev 3 (DDR2)
CPU running @ 1200Mhz L2 running @ 400Mhz
SysClock = 400Mhz , TClock = 200Mhz

DRAM CAS Latency = 5 tRP = 5 tRAS = 18 tRCD=6
DRAM CS[0] base 0x00000000   size 256MB
DRAM CS[1] base 0x10000000   size 256MB
DRAM Total size 512MB  16bit width
Flash:  0 kB
Addresses 8M - 0M are saved for the U-Boot usage.
Mem malloc Initialization (8M - 7M): Done
NAND:512 MB

CPU : Marvell Feroceon (Rev 1)

Streaming disabled
Write allocate disabled

USB 0: host mode
PEX 0: interface detected no Link.
Net:   egiga0 [PRIME], egiga1
Hit any key to stop autoboot:  0
Marvell>>
```

Since all u-boot settings will be lost during the update it's a good idea to remember the MAC address of your Sheeva :

```
Marvell>> printenv ethaddr
ethaddr=F0:AD:4E:01:49:8F
```

Which is of cause differs for each Sheeva.

You have to options for your new u-boot image. Either a rather outdated "official" 2012.04.01-2 build which can be downloaded here **or you could use my own build** 2013.08.26 (based on uboot 2013.10-rc1-00034-g2b26201) from here. Copy it to your TFTP directory. Now let's flash it:

```
setenv serverip 172.16.0.187 # IP of your TFTP server
setenv ipaddr 172.16.0.15    # IP address for the sheeva
tftpboot 0x0800000 u-boot.kwb
nand erase 0x0 0x60000
nand write 0x0800000 0x0 0x60000
```

After a reset you should be greeted by the new bootloader:

```
U-Boot 2012.04.01 (Jun 01 2012 - 02:27:06)
Marvell-Sheevaplug - eSATA - SD/MMC

SoC:   Kirkwood 88F6281_A1
DRAM:  512 MiB
WARNING: Caches not enabled
NAND:  512 MiB
NAND read from offset 60000 failed -74
*** Warning - readenv() failed, using default environment

In:    serial
Out:   serial
Err:   serial
Net:   egiga0
88E1116 Initialized on egiga0
Hit any key to stop autoboot:  0
Marvell>>
```

As you can see, your old settings are all scrambled. It's a good idea to save back the MAC address now:

```
setenv ethaddr F0:AD:4E:01:49:8F
saveenv
reset
```

Great! You got a great new and powerful bootloader installed which can also boot from ubifs partitions!

Installing Debian

Now it's time to install Debian. Unfortunately we cannot directly install Debian to Flash because it is not supported by the Debian installer. So we first install it to a USB stick and later move it to Flash.

Installing to USB stick

The installation to stick is basically a total standard installation as explained by the well know HOW-TO at cyrius.com. This is the gist of it:

Download the uImage and uInitrd to your TFTP directory and boot it with:

```
setenv serverip 172.16.0.187 # IP of your TFTP server
setenv ipaddr 172.16.0.15    # IP address for the sheeva
tftpboot 0x00800000 uImage
tftpboot 0x01100000 uInitrd

setenv bootargs console=ttyS0,115200n8 base-installer/initramfs-tools/driver-policy=mos
bootm 0x00800000 0x01100000
```

Continue the installation to stick and do not change the partition layout. The installation will take some time, go and grab a coffee.

Transferring image to flash

Boot from USB stick

Once the installation has finished and u-boot is booted again it's time to boot from USB and copy the image to flash.

```
setenv bootargs_console console=ttyS0,115200
setenv bootcmd_usb 'usb start; ext2load usb 0:1 0x00800000 /uImage; ext2load usb 0:1 0x
setenv bootcmd 'setenv bootargs $(bootargs_console); run bootcmd_usb; bootm 0x00800000
run bootcmd
```

This will straightly boot up your Debian installation, so log in with your username and password:

```
Debian GNU/Linux 6.0 sheevaplug ttyS0

sheevaplug login: user
Password:
```

```
Linux sheevaplug 2.6.32-5-kirkwood #1 Sun May 6 16:57:51 UTC 2012 armv5tel

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
user@sheevaplug:~$
```

Now it's time to copy your Debian to Flash. For this you have to become root, either by using su or sudo – depending on the method you choose during installation.

For the installation we still have to format the Flash so let's install the required software (as superuser):

```
apt-get install mtd-utils
```

Now we are able to format the Flash as ubifs and mount it to /mnt/:

```
ubiformat /dev/mtd2 -s 512
ubiattach /dev/ubi_ctrl -m 2
ubimkvol /dev/ubi0 -N rootfs -m
mkfs.ubifs --compr=zlib /dev/ubi0_0
mount -t ubifs ubi0:rootfs /mnt
```

What is left now is copying all the required data to the newly created ubifs partition:

```
mkdir /tmp/rootfs
mount -o bind / /tmp/rootfs/
cp -a /tmp/rootfs/* /mnt/
cp -a /boot /mnt/
```

What should also be adapted is the fstab of the Debian installation in Flash at */mnt/etc/fstab*. It should look like this:

```
# /etc/fstab: static file system information.
#
# Use 'blkid' to print the universally unique identifier for a
# device; this may be used with UUID= as a more robust way to name devices
# that works even if disks are added and removed. See fstab(5).
#
# <file system> <mount point>   <type>  <options>       <dump>  <pass>
proc            /proc           proc    defaults        0       0
ubi0:rootfs  /               ubifs   defaults,noatime,rw                    0 0
tmpfs       /var/run        tmpfs   size=1M,rw,nosuid,mode=0755            0 0
tmpfs       /var/lock       tmpfs   size=1M,rw,noexec,nosuid,nodev,mode=1777 0 0
tmpfs       /tmp            tmpfs   defaults,nosuid,nodev                  0 0
```

Setup u-boot to boot from ubi partition

Reboot your Sheeva and enter u-boot. Make u-boot know about your partition setup and set the boot sequence:

```
# Set uboot to boot from ubi
setenv mtdids nand0=orion_nand
setenv mtdparts mtdparts=orion_nand:0xa0000@0x0(u-boot),0x400000@0x100000(uImage),0x1fb

setenv bootargs_ubi 'console=ttyS0,115200 ubi.mtd=2 root=ubi0:rootfs rootfstype=ubifs'
setenv bootcmd_ubi 'ubi part nand0,2; ubifsmount rootfs; ubifsload 0x1100000 /boot/uIni
setenv bootcmd 'setenv bootargs $(bootargs_ubi); run bootcmd_ubi; bootm 0x00800000 0x01
```

```
8
9 saveenv
```

Congratulations! You've just successfully installed Debian to a modern ubifs partition on your SheevaPlug Flash!

Update (24.06.2013)

You might experience problems when Debian updates your kernel. This is due to a bug. The best workaround is to remove flash-kernel

```
apt-get remove flash-kernel
```

Update (26.08.2013)

Since the "official" u-boot image is rather outdated now, I've build a new one based on git master as of 26.08.2013 (2013.10-rc1-00034-g2b26201) you can get it here. I recommend it over the older u-boot versions. This also fixes problems where the SheevaPlug would not boot from a damaged ubifs partition.

26 thoughts on "Installing Debian on a Sheevaplug into Flash"

Valentin

August 16, 2012 at 11:45

At last! You've basically saved my sanity.
I've been wrestling with the u-boot for two days to figure out how to load ul* from the flash and I completely missed the ubi support in it.

Many thanks!

V.

August 16, 2012 at 12:43

★ mabl

Hi Valentin,

I'm glad this helped you! I've also only discovered the ubi support by accident. Most guides I found stored the image directly (which requires some kind of flashing mechanism) or on a SD card.

One thing I've learnt just yesterday is that uboot will fail to load files from an unclean ubifs file system. So if you are looking for stability you might move /boot to it's own partition.

Best,
Matthias

Valentin

August 16, 2012 at 13:36

Maybe another ubi partition for /boot (as it was originally done). However the current md layout should be changed to increase the space on md1.

V.

Edward

February 6, 2013 at 23:31

Thanks for the guide this helped me a lot. If you want some more space on the internal ubifs use as a mount option during creation and operations "compr=zlib" as explained on this page: http://www.linux-mtd.infradead.org/faq/ubifs.html It helped me save almost 50% of my storage space, also the guruplug is capable of speedy zlib, whereas nand is rather slow, so it speeds up the operations.

★ mabl

February 7, 2013 at 07:33

Hi Edward,

glad this guide helped you! The compression is a very good idea, I don't know why I did not think of it.

Thanks!

March 16, 2013 at 08:58

Tony

I got a sheevaplug which is come with a debian 5. I tried to upgrade the u-boot that you mentioned, it is failed to boot up. I fixed it using esia to reload the u-boot into it. Do you know any version of u-boot that is compatible for the sheevaplug that come with debian 5?

★ mabl

March 25, 2013 at 16:27

Hello Tony,

I don't know about any hardware difference for the Debian 5 Sheevaplug. But I'm no expert. You could have a look at the other u-boots available here:

http://people.debian.org/~tbm/u-boot/2012.04.01-2/

Best,
Matthias

Fred

March 22, 2013 at 23:40

Thanks much Matthias for the tutorial. After reading about the different solutions to upgrade my SheevaPlug from its stock Ubuntu 9.04… and getting nowhere, I'm now a happy Debian 6.0 user

Peter Simonds

August 23, 2013 at 15:46

Your link at the bottom of this page:
"Pingback: Install Debian 7 Wheezy on NAND flash of Sheevaplug eSata | 2nd.homelinux.com",
is a dead link.

I was wanting to install Debian 7 on my NAND. Do I just follow the instructions for Debian 6 and the same bootloader will be fine?

August 23, 2013 at 17:37

Hi Peter,

★ mabl

I'm using Debian 8 with the same instructions. So it ought to work. I've compiled my own u-boot and it looks to be even more stable. Unfortunately I've yet to publish the build or find a new "official" one. But the one linked here does work as well.

PeterCheese

August 26, 2013 at 06:57

Hi Matthias,
Thanks a lot for this great post , it really saves my time and make my Sheeva getting a brand new air from my previous nand content stucked with ubuntu 9.04.
Best regards.

★ mabl

August 26, 2013 at 08:11

Hi Peter,

I'm glad this could be of help. If you run into trouble, don't hesitate to contact me. I've also just uploaded a newer u-boot image. See the article for updates.

Best,
Matthias

Steven

August 29, 2013 at 22:20

Thanks so much for this walk-through and updated u-boot! I just wanted to put out there that when using the 26.08.2013 u-boot.kwb I had to change this line:

setenv bootcmd_ubi 'ubi part nand0,2; ubifsmount rootfs; ubifsload 0x1100000 /boot/uInitrd; ubifsload 0x800000 /boot/uImage'

to

setenv bootcmd_ubi 'ubi part nand0,2; ubifsmount ubi0; ubifsload 0x1100000 /boot/uInitrd; ubifsload 0x800000 /boot/uImage'

Thanks again!

September 9, 2013 at 06:51

Paul

> What should also be adapted is the fstab of the Debian installation in Flash

Having installed Wheezy image, here's what my /etc/fstab looks like:

```
# /etc/fstab: static file system information.
#
# Use 'blkid' to print the universally unique identifier
for a
# device; this may be used with UUID= as a more robust way
to name devices
# that works even if disks are added and removed. See
fstab(5).
#
#
# / was on /dev/sda2 during installation
UUID=9eab00fc-aee0-49dc-96bc-c7c0013d35dd / ext4
errors=remount-ro 0 1
# /boot was on /dev/sda1 during installation
UUID=26c06254-fb45-4c02-b31e-b16e7ec7a06a /boot ext2
defaults 0 2
# swap was on /dev/sda5 during installation
UUID=d8e4f70a-b567-4ce0-9c20-67e2b8d463e1 none swap sw 0 0
/dev/sda1 /media/usb0 auto rw,user,noauto 0 0
/dev/sda2 /media/usb1 auto rw,user,noauto 0 0
/dev/sda3 /media/usb2 auto rw,user,noauto 0 0
/dev/sda5 /media/usb3 auto rw,user,noauto 0 0
```

How would this be adapted to boot from flash?

R Fujii

September 14, 2013 at 23:49

Hmmm. Is it just me, or does *uboot 2013.10-rc1-00034-g2b26201* given abovee missing the mmc device? Even the uboot help is not showing the mmc command.

RS

January 30, 2014 at 21:02

Brilliant & beautifully clear, well laid out instructions that have saved me a shed load of time. I used cyrius.com to do the initial download to USB stick which is "simpler" than ftp and I used his

setenv machid a76
saveenv

to restore the esata machine type after flashing the bootloader.

I also had to use Steven's amendment to "ubi0" and like R Fujii above couldn't get it to boot to mmc with your bootloader.

Nonethless an excellent help – thanks very much

Kyle

March 13, 2014 at 07:35

Thanks for the great instructions! I've got one problem when making the ubifs filesystem: there doesn't seem to be anything in the /boot directory of the root filesystem after installing it to a flash drive. I tried putting the uImage and uInitrd files previously linked in this post into /boot before copying them to the ubifs, but of course it boots to try and install the OS. Am I using the wrong uImage and uInitrd files? Should they have already been present in /boot after installing to the USB drive? Where can I find them?

Thanks!

★ mabl

March 13, 2014 at 08:08

Hi Kyle,

when installing Debian to USB stick, the boot files are installed into their own partition which is mounted under /boot. While under the proposed UBIFS setup, only one partion is used.

Did you execute the "cp -a /boot /mnt/" as proposed in the guide? If not, just boot from usb stick again and execute these steps:

```
ubiattach /dev/ubi_ctrl -m 2
ubimkvol /dev/ubi0 -N rootfs -m
mkfs.ubifs –compr=zlib /dev/ubi0_0
mount -t ubifs ubi0:rootfs /mnt
cp -a /boot /mnt/
```

You should then have the files in /boot.

As it happens I'll install a new Sheevaplug during the next week and crosscheck this guides for errors and changes required for the current Debian.

Matthias

John

April 2, 2014 at 23:08

Hi,

I'm trying to get my SheevaPlug up to date with a clean sweep and install but…
I went through the pretty detailed description on this website and I can't get it to boot.
Here my boot log.
——

U-Boot 2013.10-rc1-00034-g2b26201 (Aug 26 2013 – 07:58:32)
Marvell-Sheevaplug

SoC: Kirkwood 88F6281_A0
DRAM: 512 MiB
WARNING: Caches not enabled
NAND: 512 MiB
In: serial
Out: serial
Err: serial
Net: egiga0
88E1116 Initialized on egiga0
Hit any key to stop autoboot: 0
UBI: attaching mtd1 to ubi0
UBI: physical eraseblock size: 131072 bytes (128 KiB)
UBI: logical eraseblock size: 129024 bytes
UBI: smallest flash I/O unit: 2048
UBI: sub-page size: 512
UBI: VID header offset: 512 (aligned 512)
UBI: data offset: 2048
UBI: attached mtd1 to ubi0
UBI: MTD device name: "mtd=2"
UBI: MTD device size: 507 MiB
UBI: number of good PEBs: 4052
UBI: number of bad PEBs: 4
UBI: max. allowed volumes: 128
UBI: wear-leveling threshold: 4096
UBI: number of internal volumes: 1
UBI: number of user volumes: 1
UBI: available PEBs: 0
UBI: total number of reserved PEBs: 4052
UBI: number of PEBs reserved for bad PEB handling: 40
UBI: max/mean erase counter: 33/19
UBIFS error (pid 0): ubifs_get_sb: cannot open "rootfs", error -22
UBIFS error (pid 0): ubifs_mount: Error reading superblock on volume 'rootfs' errno=-22!

ubifsmount – mount UBIFS volume

Usage:

ubifsmount
– mount ‘volume-name’ volume
UBIFS not mounted, use ubifs mount to mount volume first!
ubifsload – load file from an UBIFS filesystem

Usage:
ubifsload [bytes]
– load file ‘filename’ to address ‘addr’
UBIFS not mounted, use ubifs mount to mount volume first!
ubifsload – load file from an UBIFS filesystem

Usage:
ubifsload [bytes]
– load file ‘filename’ to address ‘addr’
Wrong Image Format for bootm command
ERROR: can’t get kernel image!
Marvell>>

—–

★ mabl

April 3, 2014 at 10:55

Hi John,

have a look at the comment from Steven. It looks like the command needs to be modified for recent uboot versions.

John

April 6, 2014 at 19:48

Hi Matthias,
Thanks for the advise, it helped and it booted but still with problems during the Debian boot up, lots and lots of errors but ended up in a login prompt.
Fix it by installing ‘U-Boot 2012.04.01 (Jun 01 2012 – 02:27:06)’ .
Now it all runs smoothly at first run!
Top guide, thanks!

Cheers

June 27, 2014 at 19:53

Said

Hi Mathias,

Thank you for your great job.
I applied your procedure but I still have errors to load from flash.

Could you help please.

Said

Here is my env variables:

baudrate=115200
bootargs=console=ttyS0,115200 ubi.mtd=2 root=ubi0:rootfs rootfstype=ubifs
bootargs_console=console=ttyS0,115200
bootargs_ubi=console=ttyS0,115200 ubi.mtd=2 root=ubi0:rootfs rootfstype=ubifs
bootcmd=setenv bootargs $(bootargs_ubi); run bootcmd_ubi; bootm 0x00800000 0x01100000
bootcmd_ubi=ubi part nand0,2; ubifsmount ubi0; ubifsload 0×1100000 /boot/uInitrd; ubifsload 0×800000 /boot/uImage
bootdelay=3
ethact=egiga0
ethaddr=F0:AD:4E:00:68:56
filesize=189850
machid=a76
mtddevname=u-boot
mtddevnum=0
mtdids=nand0=orion_nand
mtdparts=mtdparts=orion_nand:0xa0000@0x0(u-boot),0x400000@0x100000(uImage),0x1fb00000@0x500000(rootfs)
partition=nand0,0
stderr=serial
stdin=serial
stdout=serial

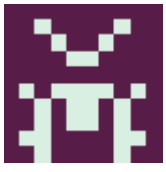
Here are my log:
U-Boot 2013.10 (Oct 21 2013 – 21:06:56)
Marvell-Sheevaplug – eSATA – SD/MMC

SoC: Kirkwood 88F6281_A1
DRAM: 512 MiB
WARNING: Caches not enabled
NAND: 512 MiB
In: serial
Out: serial
Err: serial
Net: egiga0
88E1116 Initialized on egiga0
Hit any key to stop autoboot: 0
UBI: attaching mtd1 to ubi0

UBI: physical eraseblock size: 131072 bytes (128 KiB)
UBI: logical eraseblock size: 129024 bytes
UBI: smallest flash I/O unit: 2048
UBI: sub-page size: 512
UBI: VID header offset: 512 (aligned 512)
UBI: data offset: 2048
UBI: attached mtd1 to ubi0
UBI: MTD device name: “mtd=2″
UBI: MTD device size: 507 MiB
UBI: number of good PEBs: 4047
UBI: number of bad PEBs: 9
UBI: max. allowed volumes: 128
UBI: wear-leveling threshold: 4096
UBI: number of internal volumes: 1
UBI: number of user volumes: 1
UBI: available PEBs: 0
UBI: total number of reserved PEBs: 4047
UBI: number of PEBs reserved for bad PEB handling: 40
UBI: max/mean erase counter: 7/2
UBIFS: mounted UBI device 0, volume 0, name “rootfs”
UBIFS: mounted read-only
UBIFS: file system size: 515063808 bytes (502992 KiB, 491 MiB, 3992 LEBs)
UBIFS: journal size: 9033728 bytes (8822 KiB, 8 MiB, 71 LEBs)
UBIFS: media format: w4/r0 (latest is w4/r0)
UBIFS: default compressor: zlib
UBIFS: reserved for root: 0 bytes (0 KiB)
Loading file ‘/boot/uInitrd’ to addr 0x00000000 with size 7452371 (0x0071b6d3)…
Done
Loading file ‘/boot/uImage’ to addr 0x00000000 with size 1611856 (0x00189850)…
Done
Wrong Image Format for bootm command
ERROR: can’t get kernel image!
Marvell>>

Pingback: Fixing pluggy | Sharing and learning

Pingback: Installing a SheevaPlug | La Fabulosa Vida del Estudiante



January 10, 2015 at 18:04

Nice post. I was checking continuously this blog and I’m impressed! Extremely useful info particularly the last part I care for such info a lot. I was looking for this particular info for a long time. Thank you and best of luck.|

ubuntu

linux

January 27, 2015 at 22:47

I believe that is among the most important info for me. And i’m glad studying your article. But want to remark on few general issues, The site style is perfect, the articles is in reality great : D. Excellent activity, cheers