

# Marcin Juskiewicz

Random mumblings of ARM/AArch64 developer.

## Unbricked my old SheevaPlug

Few months ago one of my friends borrowed SheevaPlug from me. About two weeks later he gave it back — *bricked*... I did not had time to play with it so it landed on shelf.

Yesterday I took it and decided to get it back to live. Requirements:

- bricked SheevaPlug (v1.0 without SATA)
- power cable
- mini usb cable
- usb thumb drive
- OpenOCD ("apt-get install openocd")
- cross compiler ("apt-get install gcc-arm-linux-gnueabi" under Ubuntu)
- U-Boot sources (HEAD of mainline)
- Linux sources (also HEAD of mainline)
- serial terminal (picocom, minicom, screen etc)
- few terminals or terminal multiplexer (I used tmux)

Then:

- Connected power and mini usb cables to SheevaPlug. Desktop recognized usb-serial device as `/dev/ttyUSB1`.
- Connected to it with serial terminal. Nothing appeared there of course ;)
- Run OpenOCD: "`cd /tmp;/sudo openocd -f /usr/share/openocd/scripts/board/sheevaplug.cfg -s /usr/share/openocd/scripts`". SheevaPlug was detected.
- Connected to OpenOCD: "`telnet localhost 4444`".
- Built U-boot:

```
export CROSS_COMPILE=arm-linux-gnueabi-
make mrproper
make sheevaplug_config
make u-boot.kwb
```

- Copied “u-boot” to “/tmp/uboot.elf” and used “reset;sheevaplug\_init;load\_image u-boot.elf;resume 0x00600000” — landed in U-Boot ;)
- There is “sheevaplug\_reflash\_uboot” macro but it was not working for me. So I used U-Boot to flash itself:

```
Marvell>> usb start
Marvell>> fatload usb 0:1 0x0800000 u-boot.kwb
Marvell>> nand erase 0x0 0xa0000
Marvell>> nand write 0x0800000 0x0 0xa0000
Marvell>> reset
```

- Went to [Ångström online image builder](#) and built small busybox based image.
- Unpacked tarball into /tmp/initfs, added /dev/ttyS0 node.
- Built Linux kernel:

```
export CROSS_COMPILE=arm-linux-gnueabi-
make mrproper
make kirkwood_config
make menuconfig (set INITRAMFS_SOURCE to /dev/initfs)
make uImage
```

- Copied “arch/arm/boot/uImage” to USB thumb drive and inserted it into SheevaPlug.
- Booted image:

```
Marvell>> set ethaddr 'c0:ff:ee:c0:ff:ee'
Marvell>> set bootargs 'console /dev/ttyS0,115200 rw'
Marvell>> usb start;fatload usb 0:1 0x800000 /uImage;bootm 0x800000
```

- Landed in nice and small Ångström distribution image ;)
- Went to [Ångström online image builder](#) and built console image (task-base based).
- Built Linux kernel (this time without initramfs):

```
export CROSS_COMPILE=arm-linux-gnueabi-
make menuconfig (unset INITRAMFS_SOURCE)
```

```
make uImage
```

- Copied “arch/arm/boot/uImage” to USB thumb drive and inserted it into SheevaPlug.
- Prepare NAND for UBI:

```
# mount none /dev -t devtmpfs
# udhcpc eth0
# opkg-cl update
# opkg-cl install mtd-utils
# ubiformat /dev/mtd2
# ubiattach -p /dev/mtd2
# ubimkvol /dev/ubi0 -N rootfs -s 490MiB
# ubiupdatevol /dev/ubi0_0 /media/sda1/angstrom-task-base.ubifs
# mount -t ubifs ubi0:rootfs /media/rootfs
# chown -R root:root /media/rootfs
# cp /media/sda1/uImage /media/rootfs/boot
# sync
# reboot
```

- Another reconfiguration in U-Boot:

```
Marvell>> bootargs 'console=ttyS0,115200 rw ubi.mtd=2 rootfstype=ubifs root=ubi:rootfs'
Marvell>> bootcmd 'ubi part nand0,2; ubifsmount rootfs; ubifsload 0x800000 /boot/uImage
;bootm 0x800000'
Marvell>> mtdids 'nand0=orion_nand'
Marvell>> set mtdparts 'mtdparts=orion_nand:512k(uboot),4m@1m(kernel),507m@5m(rootfs)'
Marvell>> save
Marvell>> reset
```

And now my SheevaPlug is operational again. Boots from NAND with latest U-Boot and Linux. There is around 440MB free still on NAND (not counting 4MB partition where kernel was expected to be). I can put it back on shelf now.

The only parts which I needed to compile were U-Boot and Linux kernel. I could skip bootloader and use binary image from Internet but prefer to know what my machines run (and building U-Boot is really easy). Initramfs support in Linux is real live saver as I did not had to play with initrd etc — just

build image and boot it. The only problem was that devtmpfs was not auto mounted (even if option in kernel was selected).

I could also use one of those “easy installers” made by PlugComputer community but I found such solutions more complicated (fetching binaries, finding requirements etc) than the one I used.

Related posts:

1. [How to cross compile ARM kernel under Ubuntu 10.10](#)
2. [Sheevaplug updates](#)
3. [Boot AArch64 in easy way](#)
4. [How to install Ubuntu 13.04 on Chromebook](#)
5. [SD cards die](#)

This entry was posted and tagged [angstrom](#), [brick](#), [linaro](#), [openembedded](#), [sheevaplug](#) on July 13, 2012 [<http://marcin.juszkiewicz.com.pl/2012/07/13/unbricked-my-old-sheevaplug/>] by Marcin Juszkiewicz.

---

[Google+](#) [Blog](#) [Other blogs](#)

---

This website uses cookies. You can accept it or not - your choice.

[Ok, I am fine with it.](#)

No comments yet



Add a comment as John O'Hare