Installation of firmware on ngw100

Update Das U-Boot with JTAGICE mkII

Connect the JTAGICE mkII, and then use "avr32program" from the (Cygwin or Linux) command line prompt:

To erase the ***entire*** flash. It's not possible to only erase part of the flash.

```
> sudo avr32program erase -fcfi@0
```

To program Das U-Boot

```
> sudo avr32program program -F bin -vfcfi@0 uboot.bin
```

To program the file system, additionally

```
> sudo avr32program program -F bin -vfcfi@0 -O 0x20000 ngw_jffs2_root.img
```

NOTE: I found that in order to program the flash memory you must clear it first.

With serial link

These steps will upload the root image to your Network gateway. The root image can be found at Atmel.no beta ware. You can use any terminal program that supports the **Kermit**transfer protocol

in uboot type:

```
Uboot> protect off 0x20000 0x7EFFFF
Uboot> erase 0x20000 0x7EFFFF
Uboot> loadb 0x90000000
## Ready for binary (kermit) download to 0x90000000 at 115200 bps...
```

Now select file transfer from in your terminal program, and upload the file 'ngw_jffs2_root.img' with Kermit as protocol. Be prepared to wait ~25 minutes for the transfer to complete!

```
## Total Size      = 0x00640000 = 6553600 Bytes
## Start Addr      = 0x90000000
```

Note the size and use it as the last argument to cp.b

```
Uboot> cp.b 0x90000000 0x20000 0x640000
Copy to Flash... done
```

```
Uboot> protect on all
Uboot>
```

reboot your network gateway

The Network Gateway will now boot the new root image. The amount of error messages is due to a non upgraded /usr partition. Please follow the steps for upgrading the /usr partition above to complete the Network Gateway upgrade.

I was able to update the U-Boot, /root file system and /usr file system using
JTAG (on STK600) and a SandDisk 2GB SD-card which was formatted to FAT-file
system in windows xp. The /usr file system was written to the serial flash using
SD-card and U-Boot. I did not need to boot from SD-card at all.

First I updated the U-Boot and the /root file system using JTAG. These are both
written in the parallel flash of the NGW100.

In Windows Command Prompt:
Code:

```
avr32program program -e -F bin -vfcfi@0 u-boot.bin
avr32program program -e -F bin -vfcfi@0 -O 0x20000 rootfs.avr32.jffs2-root
```

This writes the U-Boot and /root file system images correctly to the parallel flash. I noticed that if I tried to use Firefox during the programming, the prosess was aborted with an error. So just wait for the programming to finish doing nothing.

After this I copied the /usr image file (rootfs.avr32.jffs2-usr) to the SD card
(just a regular file copy), inserted the SD-card in the NGW100 SD-card reader
and booted to U-Boot.

This is how the "rootfs.avr32.jffs2-usr" image file is written to the serial
flash using U-Boot:

At (U-Boot) runtime, you can detect any SPI flash. This command must be run first before doing any other SPI flash operation.
Code:

```
U-Boot> sf probe 0
SF: Got idcode 1f 28 00
SF: AT45 status register: bc
SF: Detected AT45DB642D with page size 1056, total 8650752 bytes
8448 KiB AT45DB642D at 0:0 is now current device
```

If you look at the physical serial flash chip on the NGW100 you can see it has
the marking "ATMEL 45DB642D-CNU" on it. This is where we want to write.

Initialize the mmc (the SD-card)
Code:

```
U-Boot> mmcinit
```

Read the image file (from SD-card) to external memory (the NGW100 SDRAM)
location 0x90000000
Code:

```
U-Boot> fatload mmc 0:1 0x90000000 rootfs.avr32.jffs2-usr
..reading rootfs.avr32.jffs2-usr
.....................................................................
.....................................................................
.....................................
```

```
8473344 bytes read
```

You will be given the amount of bytes that was read to memory. We need this value in hexadesimal format later. In above case 8473344 = 0x814B00.

If the SD-card is formatted as ext2 file system, use ext2load instead of fatload.

We need to erase the serial flash before writing to it
Code:

```
U-Boot> sf erase 0x0 0x840000
SF: AT45: Successfully erased 8650752 bytes @ 0x0
```

Finally, use the sf write command to write a range of bytes from (SDRAM) memory into some offset into the serial flash.
Code:

```
U-Boot> sf write 0x90000000 0x0 0x814B00
SF: AT45: Successfully programmed 8473344 bytes @ 0x0
```

Above we write 0x814B00 bytes from (SDRAM) location 0x90000000 to serial flash offset 0x0.

Reset and (hopefully) watch the updated linux load.

NGW100 Error: UBoot Can't Get Kernel Image!

May 23rd, 2009 by TheKidd

I did run into a bit of a problem after restoring u-boot and the root filesystem with the JTAG ICE. It appears that the version the board shipped with was setup to boot the linux kernel off of /dev/mtdblock1 at /uImage but the latest build, 2.3.0, had the kernel located at /boot/uImage. I was constantly getting the following error from u-boot:

Wrong Image Format for bootm command

ERROR: can't get kernel image!

UBoot>

After a bit of Googling, I found a blog on the subject which suggested the following:

Booting By Internal Flash

To boot the kernel and root file system off of the internal flash (/dev/mtd1), get into u-boot and enter this:

Uboot> *askenv bootcmd*

Please enter 'bootcmd': *fsload /boot/uImage; bootm*

Uboot> *askenv bootargs*

Please enter 'bootargs': *console=ttyS0 root=/dev/mtdblock1 rootfstype=jffs2*

Uboot> *saveenv*

Uboot> *bootm*