

Einrichtung des NGW100

Andreas Badurczyk

29. April 2010

Inhaltsverzeichnis

1	Vorbedingungen	2
1.1	Benötigte Debian-Pakete	2
2	Erste Verbindung mit dem NGW100	3
2.1	Aktualisierung von Uboot	3
2.1.1	Vorbereiten der SD-Karte	3
2.1.2	Herunterladen der neuen Version	3
2.1.3	Flashen der neuen Version	4
3	Herunterladen und kompilieren von Buildroot	4
4	Einrichtung des NFS-Servers	6
5	Starten des NGW100 über NFS	7
5.1	Speichern der alten Einstellungen	7
6	Setzen der neuen Einstellungen für NFS	7
7	Starten von SD-Karte	8

1 Vorbedingungen

In dieser Anleitung werde ich beschreiben, welche Schritte nötig sind, um buildroot für das NGW100 zu kompilieren und das System per NFS zu starten. Im Anschluss beschreibe ich noch, wie man es von SD-Karte starten kann.

Ich nutze Debian 5, nachdem es auf Arch Linux diverse Kompilierungsprobleme gab und ich das NGW100-Board nicht überreden konnte, das NFS zu nutzen. Unter Debian lief es sofort OOTB.

Anmerkung: Sollte bei einem Befehl ein '\ ' vorkommen, so bedeutet dies einen Zeilenumbruch. Dieser kann beim Ubootterminal Probleme verursachen. Lasst ihn also weg und fügt stattdessen eine lange Zeile ein.

1.1 Benötigte Debian-Pakete

Installiert werden müssen folgende Pakete:

- build-essential
- gettext
- flex
- nfs-kernel-server
- libncurses5-dev
- zlib1g-dev

Alle weiteren Pakete sollten bereits installiert sein.

2 Erste Verbindung mit dem NGW100

Zunächst testen wir die Verbindung auf die serielle Schnittstelle per minicom. Wir benötigen den Gerätenamen des seriellen Anschlusses an unserem Debian-System:

```
$ dmesg |grep tty
[    7.037911] usb 3-1: pl2303 converter now attached
to ttyUSB0
```

Der Geräteiname ist damit /dev/ttyUSB0

Nun starten wir minicom und stellen den Gerätenamen ein. Sobald minicom gestartet ist drückt man Strg+a + o, wählt 'Einstellungen zum seriellen Anschluss', 'a', editieren der ersten Zeile, 'Enter' und abschließend noch 'Speichern als »df«'.

Nun starten wir das NGW100 und minicom sollte uns die Ausgabe anzeigen.

2.1 Aktualisierung von Uboot

2.1.1 Vorbereiten der SD-Karte

Die mitgelieferte Uboot-Version war bei mir zu alt und ich konnte nicht per NFS booten. Deshalb beschreibe ich die Schritte zum Aktualisieren:

Benötigt wird eine SD-Karte, die nicht zu groß ist. Mein erster Test mit einer 2GB-Karte scheiterte, eine alte 64MB-Karte funktionierte dagegen.

Mein NGW100 erkannte keine Fat-Partition, also musste ich die Karte mit ext2 formatieren:

Ermitteln des Gerätenamens:

```
$ cat /proc/partitions
[..]
 8      48      60928 sdd
 8      49      60912 sdd1
```

Erstellen des Dateisystems:

```
$ sudo umount /dev/sdd1
$ sudo mkfs.ext2 /dev/sdd1
```

2.1.2 Herunterladen der neuen Version

Anschließend laden wir das neue Uboot herunter und speichern es auf die SD-Karte. Danach wird die Partition aus dem Dateisystem entfernt. Eine vorkompilierte Version gibt es auf <http://www.atmel.no/buildroot/buildroot-u-boot.html>.

```
$ wget http://www.atmel.no/buildroot/binaries/u-boot\  
/atngw100/flash-upgrade-atngw100-v2008.10.uimg  
$ sudo mkdir -p /mnt/sdd1  
$ sudo mount /dev/sdd1 /mnt/sdd1  
$ sudo cp flash-upgrade-atngw100-v2008.10.uimg /mnt/sdd1  
$ sudo umount /dev/sdd1
```

2.1.3 Flashen der neuen Version

Nun wird die Karte in den Kartenslot des NGW100 gesteckt. Wir brauchen außerdem das Fenster mit minicom. Das NGW100 wird nun neu gestartet und in minicom drücken wir schnell genug die Leertaste. Nun aktualisieren wir:

```
U-Boot> mmcinit  
[...]  
.U-Boot> ext2load mmc 0:1 0x10400000 \  
/flash-upgrade-atngw100-v2008.10.uimg  
.....  
70447 bytes read  
U-Boot> bootm 0x10400000  
[...]  
Press 'y' to continue, or any other key to abort  
Erasing... done  
Programming... done  
Verifying... done  
Flash upgrade successful. Please press reset or  
cycle power.
```

Das war's. NGW100 neustarten und wir können mit buildroot weitermachen.

3 Herunterladen und kompilieren von Buildroot

Ich verwende Buildroot in Version 2010.02. Dies kann unter <http://buildroot.uclibc.org/download.html> heruntergeladen werden. Anschließend wird es entpackt und wir wechseln in das Verzeichnis:

```
$ wget http://buildroot.uclibc.org/downloads/  
buildroot-2010.02.tar.bz2  
$ tar -jxvf buildroot-2010.02.tar.bz2  
$ cd buildroot-2010.02/
```

Erstellen der Standardkonfiguration für das NGW100:

```
$ make atngw100_defconfig
```

Alle Fragen beantworten wir durch drücken von Enter. Jetzt kompilieren wir das Paket:

```
$ make
```

Dies brach bei mir nach einiger Zeit ab.

```
setlocale(LC_CTYPE,"en_US.UTF-8") failed!  
make[1]: *** [extra/locale/c8tables.h] Fehler 1
```

Also aktivieren wir en_US.UTF-8 UTF-8:

```
$ sudo vim /etc/locale.gen  
  
// Entfernen des # vor en_US.UTF-8 UTF-8  
// Abspeichern:  
:wq  
  
$ sudo locale-gen  
Generating locales (this might take a while)...  
  de_DE.UTF-8... done  
  en_US.UTF-8... done  
Generation complete.
```

Jetzt wird wiederum 'make' aufgerufen.
Auch dieses mal bricht es nach einiger Zeit ab:

```
cp: regulaere Datei '/tftpboot' kann nicht angelegt  
werden: Keine Berechtigung  
make: *** [jffs2root] Fehler 1
```

Kleine Änderung per make menuconfig:

```
$ make menuconfig
  Target filesystem options --->
    (/tftpboot)   also copy the image to...
    [ENTER]
    [Zeile entfernen]
    ()           also copy the image to...

// menuconfig beenden und speichern
```

Nun lief es komplett durch. Wir müssen nur eine Kleinigkeit an der `/etc/fstab` anpassen:

```
vim output/target/etc/fstab
//den einzigen Eintrag auskommentieren:
#/dev/mtdblock3 /usr jffs2 defaults 0 0
// Speichern
```

Schließlich kopieren wir den Inhalt nach `/nfs`:

```
$ sudo mkdir /nfs
$ sudo chmod 777 /nfs
$ cp -r output/target/* /nfs
```

4 Einrichtung des NFS-Servers

Anmerkung: Mein Rechner hat die IP-Adresse **192.168.0.3**.

Folgende Zeile ans Ende der Datei `/etc/exports` hängen. Damit haben alle IP-Adressen im Bereich von 192.168.0.0 bis 192.168.0.255 Zugriff.

```
/nfs 192.168.0.0/24(rw, sync, no_root_squash)
```

Anschließend dem NFS-Server mitteilen, dass sich etwas geändert hat und prüfen, ob es übernommen wurde:

```
$ sudo exportfs -r
$ sudo exportfs
/nfs 192.168.0.0/24
```

Testen, ob es gemountet werden kann:

```
$ mkdir /tmp/nfs
$ sudo mount 192.168.0.3:/nfs /tmp/nfs
$ ls /tmp/nfs
bin  boot  config  dev  etc  home  lib  linuxrc
lost+found  media  mnt  opt  proc  root  sbin  sys
tmp  usr  var  www
$ sudo umount /tmp/nfs
```

5 Starten des NGW100 über NFS

Jetzt arbeiten wir wieder mit minicom. Startet es und das NGW100. Schnell genug die Leertaste drücken, um in die Kommandozeile zu gelangen.

5.1 Speichern der alten Einstellungen

Anzeigen der Einstellungen. Ich habe sie mir in eine Textdatei gespeichert, um die Einstellungen ggf. später wiederherzustellen.

```
U-Boot> printenv
baudrate=115200
ethaddr=00:04:25:1C:50:7E
bootdelay=1
ethact=macb0
serverip=172.24.41.2
tftpip=172.24.41.2
eth1addr=00:04:25:1C:50:7F
bootargs=console=ttyS0 root=/dev/mtdblock1 rootfstype=jffs2
bootcmd=fsload;bootm
stdin=serial
stdout=serial
stderr=serial

Environment size: 256/65532 bytes
```

6 Setzen der neuen Einstellungen für NFS

Ich habe folgende Einstellungen gewählt und per saveenv abgespeichert (den Eintrag /nfs/boot/uImage-20100426 muss man natürlich entsprechend anpassen):

```

U-Boot> set bootcmd 'nfs 0x10400000 \
192.168.0.3:/nfs/boot/uImage-20100426; bootm'
U-Boot> set bootargs 'root=/dev/nfs \
nfsroot=192.168.0.3:/nfs/
ip=192.168.0.111:192.168.0.255::255.255.255.0::eth0:off\
console=ttyS0'
U-Boot> set serverip 192.168.100.3
U-Boot> set ipaddr 192.168.0.111
U-Boot> saveenv

```

Anschließend können wir per 'boot' starten. Nach wenigen Sekunden ist das System hochgefahren.

7 Starten von SD-Karte

Wir benötigen Unterstützung für SD-Karten und ext2.

```

$ make linux26-menuconfig
File systems  ——>
  <*> Second extended fs support
Device Drivers  ——>
  <*> MMC/SD/SDIO card support  ——>
    <*> MMC block device driver
    <*> Secure Digital Host Controller Interface support
    <*> Atmel Multimedia Card Interface support

```

Und kompilieren:

```
$ make all
```

Einbinden der SD-Karte und kopieren der Dateien:

```
$ sudo mount /dev/sdd1 /mnt/sdd1
$ sudo cp -r output/target/* /media/sdd1
```

Anpassen der /etc/fstab, entfernen aus dem Dateisystem.


```
$ vim /mnt/sdd1/etc/fstab
// folgende Zeile einfüegen:
/dev/mmcblk0p1 / ext2 defaults 0 0

$ sudo umount /dev/sdd1
```

Nun stecken wir die Karte in den SD-Karten-Slot und starten das NGW100.
Im Uboot geben wir ein:

```
Uboot> askenv bootcmd
Please enter 'bootcmd': mmcinit; ext2load mmc 0:1\
 0x10400000 /boot/uImage; bootm
Uboot> set bootargs 'console=ttyS0 root=/dev/mmcblk0p1\
  rootwait '
Uboot> saveenv
Uboot> boot
```