

## UMTS unter Linux

## UMTS unter Linux

Automatisierung der Internetverbindung mit der UMTS-Karte von Vodafone

Dipl.Ing. Christoph Stockmayer, Schwaig, [sto@stockmayer.de](mailto:sto@stockmayer.de)

### 1. Einleitung

Die grundsätzliche Funktion der Vodafone-UMTS-Karte bei Linux ist in der Literatur ([1], [2]) bereits erschöpfend beschrieben worden. In diesem Artikel soll es darum gehen, wie diese Karte nun möglichst automatisiert betrieben werden kann.

### 2. Kartenmodule

Die UMTS-Karte wird als USB-Device erkannt, ordnet aber den seriellen Treiber noch nicht zu, da in den Konfigurationsdateien Vendor und Produkt nicht bekannt sind. Dies läßt sich manuell lösen mit

```
modprobe usbserial product=0x5000 vendor=0xaf0
```

oder durch Eintrag einer Option in `/etc/modules.conf`

```
options usbserial product=0x5000 vendor=0xaf0
```

oder in der hotplug-Konfiguration `/etc/hotplug/usb.handmap`

```
usbserial 0x0003 0xaf0 0x5000 0x0000 0x0000 0x00 0x00 0x00 0x00 \
          0x00 0x00 0x00000000
```

Damit werden nun beim Einstecken der Karte alle benötigten Module korrekt geladen (s. Kontrolle in `/var/log/messages`).

### 3. PIN-Eingabe

Bevor eine Internetverbindung aufgebaut werden kann, muß die PIN-Eingabe erfolgen. Dies kann über einen kleinen (shell-)Daemon erfolgen, der beim Hochfahren in einer rc-Datei aus `/etc/init.d` gestartet wird:

```
#!/bin/sh
# prüft, ob UMTS-Karte drin ist und meldet PIN an

TTY=/dev/ttyUSB0
TIME1=10
TIME2=20

while true
do
    until stty -a < $TTY > /dev/null 2>&1 # ist Karte da?
    do
        logger "keine UMTS-Karte" # info an syslogd
        sleep $TIME1
    done
    cu umts 2>&1 | logger # anmelden, wenn
                        # Karte da
    sleep 25 # Dauer der
            # Anmeldung
            # normalerweise 20s

    logger "UMTS-Karte initialisiert"
    while stty -a < $TTY > /dev/null 2>&1
    do
        logger "UMTS-Karte"
        sleep $TIME2
    done
done
```

## UMTS unter Linux

```
done  
done 2> /dev/null
```

Dabei wird der eigentliche Anmeldevorgang durch das Kommando `cu` gesteuert, das wiederum durch die Dateien `/etc/uucp/sys:`

```
system umts  
port umts
```

```
und /etc/uucp/port:  
port umts  
device /dev/ttyUSB0  
speed 460800  
dialer umts
```

```
und /etc/uucp/dial:  
dialer umts  
chat "" AT OK ATZ OK AT_OPSSYS=3,2 OK AT+CPIN=9876 OK  
chat-fail BUSY  
chat_fail NO\sCARRIER  
chat_fail ERROR  
chat-fail NO\sDIALTON  
chat-fail SYSTEM\sSUSPEND  
carrier-wait 60  
dtr-toggle true  
complete \d\d+++\d\dATH\r\c  
abort \d\d+++\d\dATH\r\c
```

beschrieben wird. Dabei wird die maximale Übertragungsrate (460800 Bit/s), der Mode (1,2= nur UMTS; 3,2= lieber UMTS) und die Chat-Folge festgelegt.

### 4. Aufbau der Internetverbindung

Die Verbindung baut das Skript `umts_start` auf:

```
#!/bin/sh  
# startet Internetverbindung (Abbruch mit STRG-C oder kill)  
  
TTY=/dev/ttyUSB0  
SPEED=460800  
  
route del default  
stty $SPEED < $TTY  
pppd $TTY $SPEED file /etc/ppp/options.umts
```

Dabei hat die `/etc/ppp/options.umts`-Datei folgende Parameter:

```
460800  
debug  
connect /etc/ppp/umts-chat  
crtscts  
local  
user d2  
nomagic  
novj  
novjccomp  
nodetach
```

## UMTS unter Linux

```
ipcp-accept-local
ipcp-accept-remote
noauth
defaultroute
```

und der Chat (/etc/ppp/umts-chat) wählt sich ins Netz ein und startet dann den pppd-Prozeß:

```
exec chat
TIMEOUT 5
ECHO ON
ABORT '\nBusy\r'
ABORT '\nError\r'
ABORT '\nNo answer\r'
ABORT '\nNo carrier\r'
ABORT '\nNo dialtone\r'
ABORT '\nRinging\r\n\r\nRinging\r'
'' \rAT
TIMEOUT 12
SAY "\ndefining PPP context...\n"
OK ATD*99*1*#
TIMEOUT 22
SAY "\nWaiting for connect...\n"
CONNECT ""
SAY "\nConnected."
```

Damit steht die Internetverbindung und die Defaultroute zur Verfügung. Ev. muß noch auch die Namensserver in /etc/resolv.conf geachtet werden. Durch Beendigung des pppd-Prozesses - mit STRG-C oder `kill `ps -e | grep pppd | awk '{ print $1 }'`` wird die Verbindung wieder beendet und kann jederzeit wieder durch das Skript aufgebaut werden.

### 5. Literatur:

- [1] „Daten-Düse“, Linux-Magazin 10/2004, S.82
- [2] „Linux and the Vodafone Mobile Connect (UMTS) 3G/GPRS Datacard“, Kai Engert,  
<http://www.kuix.de/umts/vodafone>

zur Person:

Christoph Stockmayer ist seit 20 Jahren freiberuflicher Trainer in den Gebieten Programmierung, C/C++/Java/Perl, OOA/OOD und im gesamten UNIX/Linux-Sektor. Er ist SuSE Certified Linux-Trainer und Lehrbeauftragter an der FH Nürnberg. Er betreut außerdem Programmier- und UNIX/Linux-Projekte.