

# SlackForumWiki : UMTSVerbindungMitPPPD

[Categories](#) :: [PageIndex](#) :: [RecentChanges](#) :: [RecentlyCommented](#) :: [Login/Register](#)

:: Search:

## Wie man mit pppd eine Internetverbindung ueber ein UMTS-Modem erstellt

### Vorwort

Das Wissen fuer dieses Tutorium kommt leider nicht von mir.

Ich habe hier im Grunde nur das [Tutorium](#) von Philipp Heumos genommen und es abgewandelt.

Vielen Dank fuer sein freundliches Einverständnis hierzu.

Die ganzen Eingaben auf der Konsole werden alle als Benutzer "root" ausgeführt.

### Kontrolle der Hardware

Zuerst wird mal das Modem an den Rechner angeschlossen.

Danach wird kontrolliert, ob das System das Modem erkennt. Der Befehl

lsusb listet alle angeschlossenen USB Geräte auf:

```
root@rechner/# lsusb
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 001 Device 002: ID 12d1:1003 Huawei Technologies Co., Ltd. E220 HSDPA Modem / E270 HSDPA/HSUPA
Bus 002 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
.
.
.
```

Hier in der zweiten Textzeile ist schon ein Modem. In diesem Falle ein web'n'walk Stick IV. Auch wenn ein Modem fest im Gerät verbaut ist (z. B. Netbooks oder Notebooks) hilft lsusb unter Umständen. In meinem Netbook war es intern per USB angeschlossen und wurde damit auch angezeigt.

Als naechstes erfolgt die Kontrolle, ob die benötigten Gerätedateien vorhanden sind.

```
root@rechner/# ls -l /dev/ttyUSB*
crw-rw---- 1 root dialout 188, 0 Apr  5 09:36 /dev/ttyUSB0
crw-rw---- 1 root dialout 188, 1 Apr  5 09:36 /dev/ttyUSB1
```

Das Gerät /dev/ttyUSB0 langt im Prinzip schon. Die Hardware ist also vorhanden und erkannt.

## Kontrolle der benötigten Module

Das einzige Modul, das wir hier benötigen ist das Modul usbserial. Die Prüfung auf das Vorhandensein geschieht durch:

```
root@rechner/# lsmod | grep usbserial
usbserial          32522  1 option
```

Das ist also auch geladen. Damit sind alle Teile vorhanden, die durch Linux selbst für die Hardware gestellt werden. Als nächstes kommen wir zur Konfiguration des pppd.

## Konfiguration des pppd

Hier erstellen wir zuerst mal die Datei /etc/ppp/peers/umts mit dem Editor der Wahl. Eventuell sind noch benötigte Verzeichnisse anzulegen. Die Datei hat folgenden Inhalt:

/etc/ppp/peers/umts (line 1)

```
1. hide-password
2. noauth
3. connect "/usr/sbin/chat -V -f /etc/ppp/umts.chat"
4. debug
5. /dev/ttyUSB0
6. 460800
7. defaultroute
8. noipdefault
9. noccp
10. nobsdcomp
11. local
12. user " "
13. password " "
14. usepeerdns
15. connect-delay 10000
16. persist
```

Anschliessend brauchen wir noch ein 'Chat-script':

```
/etc/ppp/umts.chat (line 1)
```

```
1. ABORT BUSY
2. ABORT 'NO CARRIER'
3. ABORT VOICE
4. ABORT 'NO DIAL TONE'
5. ABORT 'NO ANSWER'
6. ABORT DELAYED
7. # Modeminit
8. ' ' 'ATZ'
9. TIMEOUT 5
10. OK AT+CPIN?
11. 'READY-AT+CPIN=[pin]'- ' '
12. TIMEOUT 20
13. 'OK' 'AT+CGDCONT=1, "IP", "[ap]"
14. 'OK' 'ATDT=[tel]'
15. CONNECT \d\c
```

In diesem Listing müssen jetzt noch ein paar Ersetzungen gemacht werden:

- Zeile 11: [pin] - Wird durch die eigene PIN der verwendeten SIM-Karte ersetzt
- Zeile 13: [ap] - Wird durch die eigene AP ersetzt. Oft ist sie "internet", aber **Vorsicht:** Diese Angabe dient- oder dient der Tarifwahl. Hier kann man unter Umständen durch eine falsche AP Geld verlieren.
- Zeile 14: [tel] - Hier kommt die Telefonnummer hinein. Hier gibt es - nach meinem Wissen - nur zwei Möglichkeiten:
  - \*99#
  - \*99\*\*\*1#

## Netzwerk einstellen

Hier ist jetzt nur noch die Datei /etc/resolv.conf zu sichern und einen Link nach /etc/ppp/resolv.conf zu erstellen.

Also; zu erst die Sicherung der originalen Datei:

```
root@rechner/# mv /etc/resolv.conf /etc/resolv.conf.bak
```

Anschliessend legen wir einen Link an, der auf die richtige Datei zeigt. Das müssen wir leider

machen, da pppd diese Datei standartmäßig in /etc/ppp/ anlegt.

```
root@rechner/# ln -s /etc/ppp/resolv.conf /etc/resolv.conf
```

## Verbindung herstellen

Die Verbindung wird einfach durch folgenden Aufruf hergestellt:

```
root@rechner/# pppd call umts detach
```

Nach einigen Sekunden steht die Verbindung dann schon.

## Verbindung prüfen


Prüfen kann man die Verbindung dann durch einen 'ping'. Sinnvollerweise pingt man einen Server an, bei dem man davon ausgehen kann, das er immer läuft. Die großen Suchmaschinen bieten sich da an.

```
root@rechner/# ping www.google.de
PING www.l.google.com (209.85.149.103) 56(84) bytes of data.
64 bytes from ber01s02-in-f103.1e100.net (209.85.149.103): icmp_req=1 ttl=57 time=29.8 ms
64 bytes from ber01s02-in-f103.1e100.net (209.85.149.103): icmp_req=2 ttl=57 time=29.8 ms
64 bytes from ber01s02-in-f103.1e100.net (209.85.149.103): icmp_req=3 ttl=57 time=29.9 ms
64 bytes from ber01s02-in-f103.1e100.net (209.85.149.103): icmp_req=4 ttl=57 time=30.2 ms
64 bytes from ber01s02-in-f103.1e100.net (209.85.149.103): icmp_req=5 ttl=57 time=30.0 ms
64 bytes from ber01s02-in-f103.1e100.net (209.85.149.103): icmp_req=6 ttl=57 time=30.1 ms
^C
--- www.l.google.com ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5007ms
rtt min/avg/max/mdev = 29.846/30.017/30.267/0.272 ms
```

Wenn die Pakete - wie hier - wieder zurückkommen, haben wir eine Verbindung zum Internet und sollten Surfen koennen.

[CategoryTutorials](#)

[CategoryHardware](#)

[Source] [History] 2011-04-05 11:31:04  Owner: [MaschinenHans](#)

Your hostname is `ip5b41d7db.dynamic.kabel-deutschland.de`

Valid XHTML :: Valid CSS: :: Powered by WikkaWiki