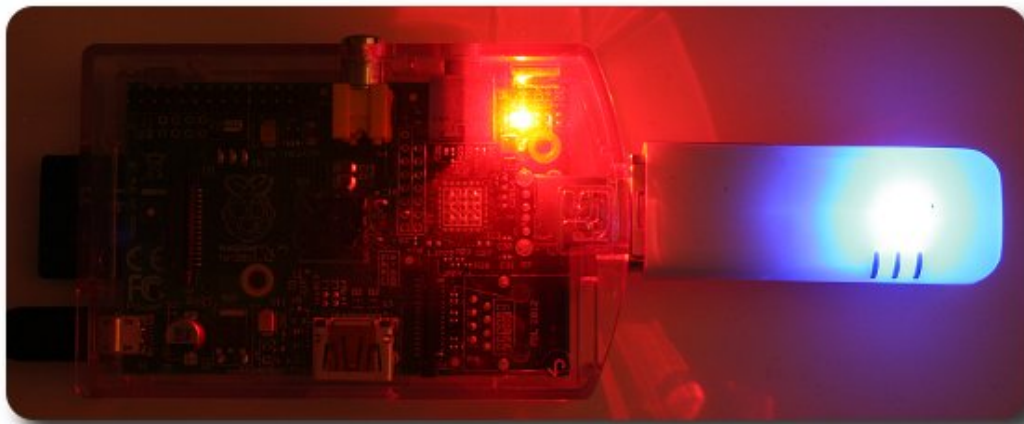


Raspberry Pi – via UMTS into the internet – configuration | setup

Posted on **24. February 2013** by **Henry Koch**



Raspberry Pi Model A connected via UMTS!

I've bought a relatively cheap, used 'MEDIONmobile' UMTS stick on Ebay and I've separately purchased a 'Aldi Talk Starter Pack' at a German Aldi market.

Aldi offers a tariff with 150MB UMTS speed and thereafter GPRS speed, for 3.99€ a month with a possibility to cancel the contract monthly.

It will 'only' used the E-Plus network, but that is relatively well available on my location.

SHORT SUMMARY: WHAT WAS NECESSARY TO GO ONLINE?

1. install wvdial
2. modify /etc/wvdial.conf
3. creating a Batch script with 2 lines, to set the SIM card password
4. customize /etc/network/interfaces
5. open connection

1. INSTALL WVDIAL

```
pi@raspberrypi - $ sudo apt-get install wvdial
.
.
Success! You can run "wvdial" to connect to the internet.
(You can also change your configuration by editing /etc/wvdial.conf)

pi@raspberrypi - $
```

2. MODIFY /ETC/WVDIAL.CONF

For this, a little bit of preliminary work is necessary.

It is important to know something about the path through which the UMTS device will be addressed.

After plugging the stick – enter the command 'lsusb' to see whether the key has been detected.

```
pi@raspberrypi ~ $ lsusb
Bus 001 Device 005: ID 12d1:1003 Huawei Technologies Co., Ltd. E220 HSDPA Modem / E230/E:
pi@raspberrypi ~ $
```

Attention: In the Raspberry Pi Model B it didn't work to insert the stick directly into the Pi, since my power supply wasn't strong enough. (I've used a 5Volt 700mA power supply.)

But the operation via a USB hub went smoothly.

The Model A can handle it, probably because they self consumes by default less power.

The last entries in /var/log/messages, will show the path through which the UMTS modem is added.

In my case: [/dev/ttyUSB0](#).

```
Feb 20 15:17:26 raspberrypi kernel: [ 10.527167] usb 1-1.3: new high-speed USB device 1
.
.
Feb 20 15:17:26 raspberrypi kernel: [ 11.313066] usbcore: registered new interface driver
Feb 20 15:17:26 raspberrypi kernel: [ 11.323480] USB Serial support registered for GSM
Feb 20 15:17:26 raspberrypi kernel: [ 11.334289] option 1-1.3:1.0: GSM modem (1-port)
Feb 20 15:17:26 raspberrypi kernel: [ 11.345030] usb 1-1.3: GSM modem (1-port) converted
Feb 20 15:17:26 raspberrypi kernel: [ 11.354849] option 1-1.3:1.1: GSM modem (1-port)
Feb 20 15:17:26 raspberrypi kernel: [ 11.375418] usb 1-1.3: GSM modem (1-port) converted
```

Now I've configured the /etc/wvdial.conf for my 'Aldi – Medionmobile' access.

```
pi@raspberrypi /etc $ cat wvdial.conf
[Dialer Defaults]
Phone =
Username =
Password =
New PPPD = yes

[Dialer eplus]
Modem = /dev/ttyUSB0
Phone = *99#
Username = eplus
Password = gprs
Init3 = AT+CGDCONT=1,"IP","internet.eplus.de","0.0.0.0"
ISDN=0
Auto Reconnect=on
Stupid Mode=off
Idle Seconds=0
Auto DNS=on
```

3. CREATING A BATCH SCRIPT WITH 2 LINES, TO SET THE SIM CARD PASSWORD

Who has secured the SIM card with a PIN, must ensure that the UMTS modem knows these PIN, before dialing.

Therefore serves the following bash script:

```
pi@raspberrypi /usr/local/bin $ sudo cat setPIN.sh
#!/bin/sh
echo "AT+CPIN=4711\n\r" > /dev/ttyUSB0
```

And now make sure so that only Root may have a look inside.

```
pi@raspberrypi /usr/local/bin $ ls -al
insgesamt 352
-rwx--x--x 1 root staff 50 Feb 24 13:14 setPIN.sh
```

4. /ETC/NETWORK/INTERFACES ANPASSEN

Finally, 4 lines coming into the /etc/network/interfaces.

```
pi@raspberrypi - $ sudo cat /etc/network/interfaces ab auto ppp0
auto lo

iface lo inet loopback
iface eth0 inet dhcp

auto ppp0
iface ppp0 inet wvdial
provider eplus
pre-up /usr/local/bin/setPIN.sh

allow-hotplug wlan0
iface wlan0 inet manual
wpa-roam /etc/wpa_supplicant/wpa_supplicant.conf
iface default inet dhcp
```

pre-up /usr/local/bin/setPIN.sh -> refers to the script for PIN transfer.

5. OPEN CONNECTION

The connection builds up automatically immediately after booting, assuming that the Stick is plugged.

Independent of the automatically solution, the connection can be established and closed, with the the following commands:

```
pi@raspberrypi - $ sudo ifup ppp0
pi@raspberrypi - $ sudo ifdown ppp0
```

Raspberry Pi – via UMTS into the internet – configuration | setup was last modified:

February 22nd, 2017 by Henry Koch

This entry was posted in [Raspberry Pi](#) and tagged [Raspberry Pi](#), [UMTS](#) by [Henry Koch](#). Bookmark the [permalink \[http://www.henrykoch.de/en/raspberry-pi-via-umts-into-the-internet\]](http://www.henrykoch.de/en/raspberry-pi-via-umts-into-the-internet) .

32 THOUGHTS ON "RASPBERRY PI – VIA UMTS INTO THE INTERNET – CONFIGURATION | SETUP"

Michael

on **6. May 2013 at 10:35** said:

Hi, dank für die Anleitung. Das Thema ist bei mir etwas aus dem Ruder gelaufen. Das Problem war zunächst, dass der Stick nicht erkannt wurde. Nach Anpassungen klappte dies dann. Das nächste Problem war, dass für den Stick zwar der Internet S Tarif gebucht werden kann. Genutzt werden kann er im Stick jedoch nicht. Trotz Buchung und Belastung nimmt der Stick den Tarif nicht an und bucht stattdessen eine TAGESFLAT24. Die Rücksprache mit der Hotline ergab, dass der INTERNET S gar nicht über den WebStick genutzt werden kann, sondern NUR über Smartphone. Einzig der INTERNET XL Tarif für 14,95 ist hier nutzbar. Für gelegentlichen Internetzugriff des RasPI ist mir das jedoch zu teuer. Mal sehen welche Möglichkeiten es noch gibt.

Gruß Michael

Henry

on **24. May 2013 at 08:53** said:

Ich kenne mich mit den Tarifen nicht wirklich gut aus. Ich bezahle 3,99 Euro im Monat für einen Aldi Talk UMTS Volumentarif von 150 MB im Monat. Danach wird auf GPRS Geschwindigkeit reduziert. Die 150 MB reichen für meine Zwecke mehr als aus. Auch GPRS Geschwindigkeit würde genügen. Was natürlich Voraussetzung ist, ist das man EPLUS Empfang hat. Es geht aber auch mit anderen Anbietern. Z.Bsp. Fyve.de bietet 100 MB pro Monat für 5 Euro im D2 Netz an.

Das der Surfstick mit der Raspberry Pi Online gehen kann, ist erwiesen. Das funktioniert bei mir seit März tadellos. Das Problem liegt wohl eher beim Provider der SIM Karte.

Raimo

on **1. June 2013 at 19:33** said:

Hallo und vielen Dank für die Schrittfür-Schritt-Anleitung!

Leider wird mein Huawei Surfstick (auch eplus) als Ethernet-Device erkannt und nicht als GSM-Modem! Hast du eine Idee, wie ich da vorgehen könnte? Danke im Voraus!!

Raimo

PS: Hier der Ausschnitt aus den Log-Messages:

Jun 1 18:27:20 raspberrypi kernel: [8564.951958] usb 1-1.2: new high-speed USB device number 9 using dwc_otg

Jun 1 18:27:20 raspberrypi kernel: [8565.053423] usb 1-1.2: New USB device found, idVendor=12d1, idProduct=14db

Jun 1 18:27:20 raspberrypi kernel: [8565.053456] usb 1-1.2: New USB device strings: Mfr=2, Product=1, SerialNumber=0

Jun 1 18:27:20 raspberrypi kernel: [8565.053472] usb 1-1.2: Product: HUAWEI HiLink

Jun 1 18:27:20 raspberrypi kernel: [8565.053484] usb 1-1.2: Manufacturer: HUAWEI

Jun 1 18:27:20 raspberrypi kernel: [8565.060062] cdc_ether 1-1.2:1.0: eth1: register 'cdc_ether' at usb-bcm2708_usb-1.2, CDC Ethernet Device, 58:2c:80:13:92:63

Jun 1 18:27:21 raspberrypi logger: usb_modeswitch: switched to 12d1:14db on 001/009

Henry

on **22. June 2013 at 08:04** said:

Unter http://wiki.ubuntuusers.de/USB_ModeSwitch kannst du eine Anleitung finden, wie du verhindern kannst, dass dein Linux einfach eigenmächtig das Ethernet-Device anstatt des UMTS Modems anzieht.

Sandy

on **16. June 2013 at 21:45** said:

Vielen Dank für Deine Klasse Anleitung!

Eine Frage habe ich ... kannst Du Dir erklären, warum bei mir am Model A mit aktiven Hub der WLAN-Stick abgeschaltet wird, sowie das Modem (S4011) eine Verbindung herstellt? Hab alles so konfiguriert, wie Du es hier erklärt hast ...

Gruß,
Sandy.

Henry

on **22. June 2013 at 07:56** said:

Der beschriebene Verbindungsaufbau ist so konfiguriert, dass die UMTS Karte die Internetverbindung aufbaut, sobald sie aktiv wird.

Das ist im Schritt 4 'etc/network/interfaces anpassen' eingebaut.

Da dein Rechner in der Regel nur eine IP Adresse nach außen hat, ist auch nur eine Netzwerkverbindung zur selben Zeit aktiv. Das bedeutet, sobald UMTS dazu kommt, ist WLAN raus.

Dieses Verhalten kannst du verhindern, indem du die Verbindung manuell aufbaust und somit auf den 'Service' innerhalb der /etc/network/interfaces verzichtest. Das wären dann die letzten beiden Befehle in dem Artikel: `sudo ifup ppp0` | `sudo ifdown ppp0`

Stefan

on **31. August 2013 at 13:08** said:

Hi, super Erklärung! Eine Frage: Bei mir wird bei Einwahl über PPP (UMTS-Stick) die Systemuhrzeit des RPi nicht aktualisiert (kein NTP). Woran kann das liegen, bzw. gibt es hierzu Lösungen? Danke. Beste Grüße Stefan

Henry

on **31. August 2013 at 17:58** said:

Ja da bin ich auch darüber gestolpert.

Die Raspberry Pi hat keine eingebaute Systemuhr.

Beim Bootvorgang wird eine aktive Netzverbindung vorausgesetzt und sich über die Verbindung die Systemzeit von einem NTC Server gezogen.

Es gibt kleine Zusatzmodule, die die Systemzeit ohne Netzwerkverbindung bereitstellen.

Ich habe da zwei liegen, aber noch nicht eingebaut, so dass ich auch noch nichts zu sagen kann.

Hier ist einer der vielen Links wie man das RTC Modul zum Laufen bringt:

<http://learn.adafruit.com/adding-a-real-time-clock-to-raspberry-pi/wiring-the-rtc>.

Sicherlich gibt es eine Möglichkeit die NTC Abfrage gleich nach dem die Netzwerkverbindung per UMTS aktiv ist nochmal machen kann.

Habe ich ehrlich gesagt noch nicht versucht. – Sollte es aber mal tun :-)

JayJay

on **27. December 2013 at 13:33** said:

Hab noch keinen Pi... Aber normal müsste

`~$ sudo ntpdate ptbtime1.ptb.de`

die Zeit bei vorhandener Internetverbindung holen.

Henry

on **22. October 2014 at 18:47** said:

Danke für die Info!

Das werde ich gleich mal ausprobieren.

Dann würde es auch Sinn machen direkt auf der PI via MySQL Daten zu sammeln und mit GNUPlot o.ä. zu Grafiken bauen zu lassen.

Mir fehlte dazu bisher der Zeitbezug.

Anonymous

on **13. September 2013 at 00:11** said:

Irgendeine Idee, wie die Verbindung ppp0 an eth0 weiterzuleiten?

Henry

on **14. September 2013 at 20:59** said:

Ich verstehe die Frage nicht wirklich. Aus Betriebssystemsicht ist es doch nicht unbedingt wichtig welches Gerät die Verbindung ins Netz eröffnet. Hauptsache Sie ist da. Was soll es bringen eine UMTS Karte, wie eine kablegebundene Ethernetkarte anzustreichen?

Michael

on **17. June 2015 at 18:46** said:

Ich denke mal er möchte die UMTS Verbindung auf der eth0 freigeben.

Ben

on **15. September 2015 at 03:20** said:

Ich schätze er meinte eher den Datenfluß der beiden zu verbinden. Also eine Bridge und so den Pi zu einem UMTS-WLAN-Hotspot zu verwandeln.

Threx

on **15. September 2013 at 17:14** said:

Hallo Henry,
danke für die Hinweise!

Ich überlege mir auch einen Raspberry Pi zuzulegen und würde ihn auch gerne per Surfstick betreiben. Dazu habe ich noch eine Frage: Du schreibst, dass der direkte Anschluss des Surfstick an das Raspberry Pi Model B nicht funktionierte und die Stromversorgung zusammenbrach.

Meinst Du, man kann das Problem "einfach" lösen, indem man ein stärkeres Netzteil an den Pi anschließt oder ist es eher wahrscheinlich, dass auf der Platine der Strom nicht hinreichend weitergeleitet werden kann, um einen Surfstick über die beiden USB-Stecker an der Platine

direkt (d. h. ohne Hub mit eigener Zusatzstromversorgung) zu versorgen? Ziel der Fragestellung: Raspberry Pi B mit Surfstick ohne zweiten Stecker für ein Hub betreiben...

Würde mich auch schon über eine qualifizierte Spekulation zum Thema freuen...

Viele Grüße

Threx

Henry

on **15. September 2013 at 19:59** said:

Hallo Threx,

ich gehe stark davon aus, dass ein stärkeres Netzteil das Problem lösen wird.

Ich habe ein Netzteil, was zum Laden eines Samsung Telefons gedacht ist verwendet und das hat 5V 700mA, was nicht all so sehr viel ist.

Die Pi, die ich in meiner Garage betreibe, um Energiedaten zu verarbeiten, wird von einem DC DC Wandler direkt über eine 12V Autobatterie betrieben.

Dort habe ich noch nicht festgestellt, dass es zu einem Spannungsabfall kommt.

Sicherlich gibt es auch Surfsticks, die von Hause aus weniger Strom verbrauchen.

Es kommt auf einen Versuch an.

Ich würde es mit einem stärkeren Netzteil probieren und sehen was passiert.

Schau wir mal, dann sehn wir's schon :-)

Ben

on **15. September 2015 at 03:27** said:

Ich betreibe den Pi mit einem 2A Netzteil (modernes Handyladegerät) und kann damit gleichzeitig einen Edimax-WLAN-Stick (max 500mA) und einen Huawei-UMTS-Stick (ebenfalls max 500 mA) betreiben.

Die angegebene maximale Stromaufnahme findet man übrigens mit **sudo lusb -v**

simo

on **16. October 2013 at 21:41** said:

kann man auch mehrere Sticks betreiben?

Henry

on **19. October 2013 at 18:20** said:

Hallo Simo,

die Pi hat werkseitig 2 USB Ports und die kleine Version ohne Ethernetanschluß hat

einen USB Port.

Mit einem USB Hub läßt sich das nicht beliebig, aber doch sehr erweitern.

Jedes eingesteckte USB Gerät wird vom System erkannt, oder muß dem bekannt gemacht werden.

Es sollte also möglich sein auch 2 UMTS Sticks parallel an die PI zu stecken.

Online gehen wird es aus meiner Sicht aber nur über eine Vebindung funktionieren, also nicht parallel.

Grüsse Henry

stephan

on **17. November 2013 at 00:19** said:

Hallo Henry,

ich versuche jetzt schon seid 4 Monaten meinen Pi über das UMTS Modem online zu bekommen. Ich habe das Huawei E270 Modem und schon alle möglichen wvdial.conf aus dem internet ausprobiert. Über Ubuntu 12. und 13 klappt der Verbindungsaufbau über den networkmanager einwandfrei. Leider nicht bei dem Raspberry. Das einzige was mir angezeigt wird ist die NO CARRIER Meldung. Das erste mal nach dem Neustart findet das modem den Carrier, verliert die Verbindung und kehrt zu dem NO CARRIER Trying again zurück. So langsam weiß ich nicht mehr weiter :D Config, und Anbieter sind die selben. Wheezy sogar schon neu auf dem Raspberry installiert und nach der Installation direkt wvdial installiert.. Leider selbes Ergebnis irgend einen Rat ?

stephan

on **18. November 2013 at 19:06** said:

Hallo zusammen, ich versuche vergeblich seit Monaten mit meinem Raspberry über ein Huawei E270 Umts Modem online zu gehen. Ich habe schon diverse wvdial.conf aus dem Internet mit den Daten von eplus gefüttert aber leider nur mit Teilerfolg. Ich bekomme unter Ubuntu 12 Ubuntu und 13 mit dem installierten Networkmanager ohne Probleme eine Verbindung. Andererseits über wvdial nur das Modem initialisiert. Sobald die Verbindung aufgebaut werden soll findet das Modem keinen Carrier. Pin von der Simkarte ist abgeschaltet. weiß jemand einen Rat oder sollte ich ein anderes Modem ausprobieren ?

Henry

on **24. November 2013 at 19:04** said:

Hallo Stephan,

das erste was mir einfällt, ist das evtl die Spannung zusammenbricht, wenn der UMTS Stick online gehen will. Hast Du / Haben Sie schonmal versucht einenaktiven USB Hub zwischen Raspberry Pi und den UMTS Stick zu hängen?

Ich hatte das Problem am Anfang, dass die Energie meines Handyladegerätes nicht

ausreichte.

Sollte es mit aktiven USB Hub auch nicht gehen, wird es nötig etwas tiefer in die Konfiguration abzutauchen.

Nach der Schilderung vermute ich das Problem bei der Übergabe des Pin's.

Möglicherweise wurde die Deaktivierung der Pin Abfrage nicht von jeder Komponente als solche verstanden.

Grüsse Henry

stephan

on **30. November 2013 at 00:53** said:

Zunächst mal sorry wegen dem doppel post. Die Spannungsversorgung läuft über einen Aktiven Hub und der Pincode der Sim Karte ist ausgeschaltet. Unter Ubuntu 12 und 13 bekomme ich den Webstick nur über den installierten Networkmanager ans laufen. Unter beiden Betriebssystemen leider nicht über Wvdial.

Frank

on **18. December 2013 at 14:54** said:

SURFSTICK:

Also ich habe festgestellt, dass die Surfsticks sehr verschieden sind.

Wer hat mit welchen Surfsticks denn Erfolg gehabt. Da die Dinger nicht billig sind, würde man so schneller ans Ziel kommen.

Der Huawei E173 – funktioniert der am PI? Die meisten Router die ich gekauft hatte, kamen damit nicht zurecht.

Boris

on **8. February 2014 at 13:05** said:

Moin Moin,

schöne Darstellung.

Ich habe mir einen web n walk Stick zugelegt, wegen der schlechte Netzabdeckung in meinem Wohnort. Mein Raspberry B erkennt diesen aber nicht. Mit lsusb taucht er nicht auf. wvdial berichtete gleich nach dem Setup eine Fehlverbindung und verwies mich auf wvdial.conf. Der Stick ist CDFS formatiert, könnte das der Grund sein?

alle Gute

Boris

Franz Otto

on **7. March 2014 at 18:10** said:

Hallo,

dank der guten Anleitung habe ich alles so konfiguriert, dass mein Huawei Stick zumindest angesprochen wird und auch den Carrier entdeckt.

leider verliert er das Signal (siehe Terminal Output unten)

Hab ich da irgendwas in der config (siehe ganz unten) falsch gemacht ?

Ich gehe über o2 postpaid ins Internet

Bin um jeden Tip dankbar

Franz Otto

Terminal output:

*****Beginn*****

pi@raspberrypi ~ \$ wvdial o2

→ WvDial: Internet dialer version 1.61

→ Cannot set information for serial port.

→ Initializing modem.

→ Sending: ATZ

ATZ

OK

→ Sending: AT+CGDCONT=1,"IP","internet","0.0.0.0"

AT+CGDCONT=1,"IP","internet","0.0.0.0"

OK

→ Modem initialized.

→ Sending: ATDT*99#

→ Waiting for carrier.

ATDT*99#

CONNECT

→ Carrier detected. Waiting for prompt.

→ Connected, but carrier signal lost! Retrying...

→ Sending: ATDT*99#

→ Waiting for carrier.

^BOOT:23260771,0,0,0,64

ATDT*99#

NO CARRIER

→ No Carrier! Trying again.

→ Sending: ATDT*99#

→ Waiting for carrier.

ATDT*99#

NO CARRIER

→ No Carrier! Trying again.

→ Sending: ATDT*99#

*****Ende Output*****

config Datei :

*****Beginn*****

wvdial.conf[Dialer Defaults]

Phone =

Username =

Password =

New PPPD = yes

[Dialer o2]

Modem = /dev/ttyUSB1

Phone = *99#

Username =

Password =

Init1 = ATZ

#Init2 = ATQ0 V1 E1 S0=0 &C1 &D2 +FCLASS=0

Init3 = AT+CGDCONT=1,"IP","internet","0.0.0.0"

Phone = *99#

Username = { }

Password = { }

Stupid Mode = 1

ISDN = 0

Auto Reconnect=on

Stupid Mode=off

Idle Seconds=0

Auto DNS=on

*****Ende Config*****

Ferchland

on **9. November 2016 at 14:37** said:

Das war auch mein Problem. Habe einen Surfstick ZTE 190MF (19d2:0117 ZTE WCDMA Technologies MSM)

Dieser Stick meldet laut dmesg 3 Devices an (ttyUSB0, ttyUSB1 und ttyUSB2) letzteres muss benutzt werden und die üblichen wvdial-Init-Strings müssen weggelassen werden. Meine wvdial.conf für O2 Netzclub:

[Dialer Defaults]

New PPPD = yes

Modem = /dev/ttyUSB2

[Dialer netzclub]

Stupid mode = on

Init3 = AT+CGDCONT=1,"IP","pinternet.interkom.de"

Auto DNS = on

Idle Seconds=0

ISDN=0

Password = internet

Auto Reconnect = off
Username = internet
Phone = *99#

LukyLuke
on **12. May 2014 at 10:01** said:

Hallo leute. Erstmal, guter Beitrag!

Leider habe ich das problem mit wvdial, dass ich der bei mir keine IP bekommt...

Nach dem aufruf von wvdial bekomme ich folgendes:

...

No CARRIER

...

Call Ready

CONNECT

→ Carrier detected. Starting PPP immediately.

→ Starting pppd at Fri May 9 15:17...

→ Pid of pppd: 2272

So und das wars, eigentlich sollte er mir jetzt hier, nach dem er einen Carrier gefunden hat, eine IP abrufen und mir das auch ausgeben.

Leider ist danach aber Schluss und es läuft nichts mehr.

Muss man in der Netzwerkeinstellung noch was anpassen, damit er eine ppp verbindung aufbauen kann?

Vielen Dank,
Luke

Nils
on **9. December 2014 at 09:40** said:

hallo danke für deine Anleitung.

doch leider nichts.

hier die fehler und kernel:

wo liegt der fehler ?

Kernel

Dec 8 19:11:39 www3 kernel: [783.791383] usb 1-1.3: Product: ZTE WCDMA Technologies MSM

Dec 8 19:11:39 www3 kernel: [783.791398] usb 1-1.3: Manufacturer: ZTE, Incorporated

Dec 8 19:11:39 www3 kernel: [783.791411] usb 1-1.3: SerialNumber: P671A2TMED010000

Dec 8 19:11:39 www3 kernel: [783.795959] usb-storage 1-1.3:1.3: USB Mass Storage device detected

```

Dec 8 19:11:39 www3 kernel: [ 783.800404] scsi2 : usb-storage 1-1.3:1.3
Dec 8 19:11:39 www3 kernel: [ 783.892512] usbcore: registered new interface driver usbserial
Dec 8 19:11:39 www3 kernel: [ 783.892642] usbcore: registered new interface driver
usbserial_generic
Dec 8 19:11:39 www3 kernel: [ 783.892746] usbserial: USB Serial support registered for
generic
Dec 8 19:11:39 www3 kernel: [ 783.951147] usbcore: registered new interface driver option
Dec 8 19:11:39 www3 kernel: [ 783.951347] usbserial: USB Serial support registered for GSM
modem (1-port)
Dec 8 19:11:39 www3 kernel: [ 783.952120] option 1-1.3:1.0: GSM modem (1-port) converter
detected
Dec 8 19:11:39 www3 kernel: [ 783.960187] usb 1-1.3: GSM modem (1-port) converter now
attached to ttyUSB0
Dec 8 19:11:39 www3 kernel: [ 783.960640] option 1-1.3:1.1: GSM modem (1-port) converter
detected
Dec 8 19:11:39 www3 kernel: [ 783.970794] usb 1-1.3: GSM modem (1-port) converter now
attached to ttyUSB1
Dec 8 19:11:39 www3 kernel: [ 783.971239] option 1-1.3:1.2: GSM modem (1-port) converter
detected
Dec 8 19:11:39 www3 kernel: [ 783.972577] usb 1-1.3: GSM modem (1-port) converter now
attached to ttyUSB2
Dec 8 19:11:40 www3 logger: usb_modeswitch: switched to 19d2:0117 on 001/007
Dec 8 19:11:40 www3 kernel: [ 784.800468] scsi 2:0:0:0: Direct-Access ZTE MMC Storage
2.31 PQ: 0 ANSI: 2
Dec 8 19:11:40 www3 kernel: [ 784.803904] sd 2:0:0:0: [sda] Attached SCSI removable disk
Dec 8 19:11:40 www3 kernel: [ 784.803996] sd 2:0:0:0: Attached scsi generic sg0 type 0

```

```
#####
```

```

pi@www3 ~ $ cd /dev/
pi@www3 /dev $ ls -la |grep ttyUSB
lrwxrwxrwx 1 root root 7 Dez 8 19:11 gsmmodem -> ttyUSB2
crw-rw—T 1 root dialout 188, 0 Dez 8 19:11 ttyUSB0
crw-rw—T 1 root dialout 188, 1 Dez 8 19:11 ttyUSB1
crw-rw—T 1 root dialout 188, 2 Dez 8 19:11 ttyUSB2

```

```

root@www3:/var/log# ifup ppp0
/usr/local/bin/setPI.sh: 2: /usr/local/bin/setPI.sh: gt: not found
AT+CPIN=****

```

```

/usr/local/bin/setPI.sh: 2: /usr/local/bin/setPI.sh: /dev/ttyUSB2: Permission denied
Failed to bring up ppp0.
root@www3:/var/log#

```

```

root@www3:/etc# lsmod
Module Size Used by
option 28349 0
usb_wwan 8408 1 option
usbserial 26684 2 option,usb_wwan

```

```

root@www3:/var/log# lsusb zte MF190 von >O2
Bus 001 Device 002: ID 0424:9514 Standard Microsystems Corp.

```

Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 001 Device 003: ID 0424:ec00 Standard Microsystems Corp.
Bus 001 Device 007: ID 19d2:0117 ZTE WCDMA Technologies MSM

```
root@www3:/var/log# wvdial o2
-> WvDial: Internet dialer version 1.61
-> Initializing modem.
-> Sending: ATZ
ATZ
OK
-> Sending: AT+CGDCONT=1,"IP","pinternet.interkom.de","0.0.0.0"
AT+CGDCONT=1,"IP","pinternet.interkom.de","0.0.0.0"
ERROR
-> Bad init string.
-> Initializing modem.
-> Sending: ATZ
ATZ
OK
```

Vielleicht seht ihr was ?

Gast

on **8. January 2015 at 05:16** said:

In der setPIN.sh muß zwischen dem CPIN-kram (in Anführungszeichen) und /dev/ttyUSB0 ein Größerzeichen sein, nicht "KaufmännischesUnd g t Semikolon", da hat mal wieder jemand html-escaping vermurkst. ;)

Das AT+CGDCONT schlägt danach nur fehl, weil die pin halt noch nicht eingegeben wurde.

kasi

on **1. April 2015 at 23:13** said:

Ich bin nicht mal soweit gekommen – wie in der Anleitung beschrieben. Bei mir ist schon die Installation von wvdial nicht machbar, er bricht bei mletzten Punkt der Konfiguration ab und landet im Nirvana.

Hate einer von euch beim Raspberry auch den Hänger bei der Installation gehabt ?

Hat jemand eine Ahnung was da falsch läuft ?

Gruss Kasi

Bench

on **9. August 2015 at 23:44** said:

Es gibt vom Netzclub (www.netzclub.de) eine kostenlose Prepaid-SIM mit 100MB Highspeed-Traffic im Monat. Finanziert wird das durch Werbe-SMS und -Emails, aber ich persönlich finde das bei dem ganzen sonstigen Spam und einem gut angelegten Spamfilter nicht schlimm. Habe die Karte jedoch (bisher ;)) nur in meinem Smartphone ausprobiert...

