Linux ACM driver v0.16

Copyright (c) 1999 Vojtech Pavlik <vojtech@suse.cz>

Sponsored by SuSE

0. Disclaimer

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Should you need to contact me, the author, you can do so either by e-mail - mail your message to <vojtech@suse.cz>, or by paper mail: Vojtech Pavlik, Ucitelska 1576, Prague 8, 182 00 Czech Republic

For your convenience, the GNU General Public License version 2 is included in the package: See the file COPYING.

1. Usage

The drivers/usb/class/cdc-acm.c drivers works with USB modems and USB ISDN terminal adapters that conform to the Universal Serial Bus Communication Device Class Abstract Control Model (USB CDC ACM) specification.

Many modems do, here is a list of those I know of:

- 3Com OfficeConnect 56k
- 3Com Voice FaxModem Pro
- 3Com Sportster
- MultiTech MultiModem 56k
- Zoom 2986L FaxModem
- Compaq 56k FaxModem
- ELSA Microlink 56k

I know of one ISDN TA that does work with the acm driver:

- 3Com USR ISDN Pro TA

Some cell phones also connect via USB. I know the following phones work:

- SonyEricsson K800i

Unfortunately many modems and most ISDN TAs use proprietary interfaces and thus won't work with this drivers. Check for ACM compliance before buying.

To use the modems you need these modules loaded:

```
usbcore.ko
uhci-hcd.ko ohci-hcd.ko or ehci-hcd.ko
cdc-acm.ko
```

After that, the modem[s] should be accessible. You should be able to use minicom, ppp and mgetty with them.

2. Verifying that it works

The first step would be to check /sys/kernel/debug/usb/devices, it should look like this:

```
T:  Bus=01 Lev=00 Prnt=00 Port=00 Cnt=00 Dev#=  1 Spd=12  MxCh= 2
B:  Alloc=  0/900 us ( 0%), #Int=  0, #Iso=  0
D:  Ver= 1.00 Cls=09(hub  ) Sub=00 Prot=00 MxPS= 8 #Cfgs=  1
P:  Vendor=0000 ProdID=0000 Rev= 0.00
S:  Product=USB UHCI Root Hub
S:  SerialNumber=6800
C:* #Ifs= 1 Cfg#= 1 Atr=40 MxPwr=  0mA
I:  If#= 0 Alt= 0 #EPs= 1 Cls=09(hub  ) Sub=00 Prot=00 Driver=hub
E:  Ad=81(I) Atr=03(Int.) MxPS=   8 Ivl=255ms
T:  Bus=01 Lev=01 Prnt=01 Port=01 Cnt=01 Dev#=  2 Spd=12  MxCh= 0
D:  Ver= 1.00 Cls=02(comm.) Sub=00 Prot=00 MxPS= 8 #Cfgs=  2
```

```
    P:  Vendor=04c1 ProdID=008f Rev= 2.07
    S:  Manufacturer=3Com Inc.
    S:  Product=3Com U.S. Robotics Pro ISDN TA
    S:  SerialNumber=UFT53A49BVT7
    C:  #Ifs= 1 Cfg#= 1 Atr=60 MxPwr=  0mA
    I:  If#= 0 Alt= 0 #EPs= 3 Cls=ff(vend.) Sub=ff Prot=ff Driver=acm
    E:  Ad=85(I) Atr=02(Bulk) MxPS=  64 Ivl=  0ms
    E:  Ad=04(O) Atr=02(Bulk) MxPS=  64 Ivl=  0ms
    E:  Ad=81(I) Atr=03(Int.) MxPS=  16 Ivl=128ms
    C:* #Ifs= 2 Cfg#= 2 Atr=60 MxPwr=  0mA
    I:  If#= 0 Alt= 0 #EPs= 1 Cls=02(comm.) Sub=02 Prot=01 Driver=acm
    E:  Ad=81(I) Atr=03(Int.) MxPS=  16 Ivl=128ms
    I:  If#= 1 Alt= 0 #EPs= 2 Cls=0a(data ) Sub=00 Prot=00 Driver=acm
    E:  Ad=85(I) Atr=02(Bulk) MxPS=  64 Ivl=  0ms
    E:  Ad=04(O) Atr=02(Bulk) MxPS=  64 Ivl=  0ms
```

The presence of these three lines (and the Cls= 'comm' and 'data' classes) is important, it means it's an ACM device. The Driver=acm means the acm driver is used for the device. If you see only Cls=ff(vend.) then you're out of luck, you have a device with vendor specific-interface:

```
    D:  Ver= 1.00 Cls=02(comm.) Sub=00 Prot=00 MxPS= 8 #Cfgs=  2
    I:  If#= 0 Alt= 0 #EPs= 1 Cls=02(comm.) Sub=02 Prot=01 Driver=acm
    I:  If#= 1 Alt= 0 #EPs= 2 Cls=0a(data ) Sub=00 Prot=00 Driver=acm
```

In the system log you should see:

```
    usb.c: USB new device connect, assigned device number 2
    usb.c: kmalloc IF c7691fa0, numif 1
    usb.c: kmalloc IF c7b5f3e0, numif 2
    usb.c: skipped 4 class/vendor specific interface descriptors
    usb.c: new device strings: Mfr=1, Product=2, SerialNumber=3
    usb.c: USB device number 2 default language ID 0x409
    Manufacturer: 3Com Inc.
    Product: 3Com U.S. Robotics Pro ISDN TA
    SerialNumber: UFT53A49BVT7
    acm.c: probing config 1
    acm.c: probing config 2
    ttyACM0: USB ACM device
    acm.c: acm_control_msg: rq: 0x22 val: 0x0 len: 0x0 result: 0
    acm.c: acm_control_msg: rq: 0x20 val: 0x0 len: 0x7 result: 7
    usb.c: acm driver claimed interface c7b5f3e0
    usb.c: acm driver claimed interface c7b5f3f8
    usb.c: acm driver claimed interface c7691fa0
```

If all this seems to be OK, fire up minicom and set it to talk to the ttyACM device and try typing 'at'. If it responds with 'OK', then everything is working.