

Packet Radio via Soundkarte unter Linux

Ausgabe:
03.07.2025

Dieses Dokument wurde erzeugt mit
BlueSpice

Seite von

Packet Radio via Soundkarte unter Linux

Inhaltsverzeichnis

- [1 Das Projekt](#)
- [2 Der Source Code](#)
- [3 Der kompilierte Treiber](#)
- [4 Starten bzw. Aufrufen des Treibers](#)

Das Projekt

Dieser (USB) Soundkartentreiber von OE5DXL befindet sich in der Entwicklung und soll es ermöglichen mit 2 Kanälen mehrere Modems zugleich unter Linux zu initialisieren.

Als KISS Treiber sind bis zu 16 Modems von 1baud bis 28kbaud möglich. Der Equalizer ermöglicht einen Vollduplexbetrieb bei Verwendung eines getrennten Senders und Empfängers.

Weiterer Vorteil ist die Möglichkeit des "**Multibaud**" Digi, also mehrere Geschwindigkeiten FSK AFSK gemischt. In Stereo kann so theoretisch ein multibaud FSK AFSK KISS, als auch AXUDP AX. 25 Modem betrieben werden.

Der Source Code

[Soundmodem Modula2 und C Übersetzung Sourcecode](#)

Der kompilierte Treiber

Hier im ZIP der fertig kompilierte Soundmodem Treiber zum Download: [Soundmodem-bin](#)

UDP Filter und RAW-Monitor Konverter [udpbox-bin](#)

Starten bzw. Aufrufen des Treibers

mit oss testen 1200 + 9600 baud monitor (ohne kiss oder udp)

```
./afskmodem -f 32000 -M 0 -c 0 -b 1200 -M 1 -c 0 -b 9600 -a -g
```

mit alsa:

```
aoss ./afskmodem -f 32000 -M 0 -c 0 -b 1200 -M 1 -c 0 -b 9600 -a -g
```

APRS mit Xastir KISS-Interface, PTT auf ttyS0:

```
aoss ./afskmodem -i /tmp/soundmodem -t /dev/ttyS0 -f 32000 -M 0 -i
```

Xastir

```
"interface" > "interface control" > "add" "serial kiss tnc"
"add" "tnc port" /tmp/soundmodem
"interface control" "start"
```

2-Frequenz-halbduplex-Digi mit 1200 / 1200+9600Bd xnet mit UDP und LPT PTT:

Bei UDP ist die Startreihenfolge egal, die Programme können auch auf verschiedenen Rechnern laufen

```
sudo nice -n -19 aoss ./afskmodem \
-p /dev/parport0 -f 44100 -c 2 -s 9 -l 256 -b 6 -e 7 \
-C 0 -b 1 -r 300 -C 1 -b 2 \
-M 0 -c 1 -b 1200 -q 200 -U 127.0.0.1:9200/9210 -m 0 \
-M 1 -c 0 -b 1200 -H 40 -q 200 -U 127.0.0.1:9201/9211 -m 0 \
-M 2 -c 0 -b 9600 -a -g -q 200 -U 127.0.0.1:9202/9212 -m 0
```

linuxsnet (XNET) AUTOEXEC.NET

```
attach ip0 axudp 1 1 19200 d9210 127.0.0.1
attach ip1 axudp 2 1 19201 d9211 127.0.0.1
attach ip2 axudp 3 1 19202 d9212 127.0.0.1
po 1 baud 1200
po 2 baud 1200
po 3 baud 9600
```

XNET mit KISS und TTY ptt (XNET nach dem Modem starten!)

```
aoss ./afskmodem \
-t /dev/ttyS0 -f 24000 -i /tmp/soundmodem \
-c 2 -s 9 -l 256 -b 6 -e 7 -C 0 -r 300 \
-M 0 -c 1 -b 1200 -q 200 -m 0 \
-M 1 -c 0 -b 1200 -H 40 -q 200 -m 0 \
-M 2 -c 0 -b 9600 -a -g -q 200 -m 0
```

linuxsnet AUTOEXEC.NET

```
attach sdev0 kiss 1 3 38400 /tmp/soundmodem
po 1 baud 1200
po 2 baud 1200
po 3 baud 9600
```

144.800MHz 1200Bd, 70cm 1200+9600Bd xnet, aprsdigi, aprsd und udpbox: APRS hört auf allen Userzugängen und sendet zum IGATE.

Senden auf 144.800 nur APRS Messages.

Auf dem 1200Bd 70cm Zugang normales PR + APRS.

1. Modem sendet alle Ports zu udpbox Port 920x und hört auf Port 921x:

(auf langsamen Rechnern oder bei hoher CPU last hilft Priotität mit nice oder renice erhöhen)

```
sudo nice -n -19 aoss ./afskmodem \
-p /dev/parport0 -f 44100 -c 2 -s 9 -l 256 -b 6 -e 7 \
-C 0 -b 1 -r 300 -C 1 -b 2 \
```

```
-M 0 -c 1 -b 1200 -q 200 -U 127.0.0.1:9200/9210 -m 2 \  
-M 1 -c 0 -b 1200 -H 40 -q 200 -U 127.0.0.1:9201/9211 -m 2 \  
-M 2 -c 0 -b 9600 -a -g -q 200 -U 127.0.0.1:9202/9212 -m 2
```

2. XNET empfängt von udpbox und sendet direkt zum Modem:
linuxsnet AUTOEXEC.NET

```
attach ip0 axudp 1 1 19300 d9210 127.0.0.1  
attach ip1 axudp 2 1 19301 d9211 127.0.0.1  
attach ip2 axudp 3 1 19302 d9212 127.0.0.1  
po 1 baud 1200  
po 2 baud 1200  
po 3 baud 9600
```

3. udpbox empfängt vom Modem (Port 920x) in AXUDP und aprsdigi (9401)
sendet UI von allen Modem Funkports zu aprsdigi (Port 9400)
sendet alle UI zu aprsd auf 192.168.1.1:9000
sendet UI von aprsdigi von Port 9401 zu Modem Funkport 2
sendet nur "APRS Messages" (-f p58) zu Modem Funkport 1

```
./udpbox -R 0.0.0.0:9200 -m 127.0.0.1:9400 -m 192.168.1.1:9000 -r 127.0.0.1:93:9300 \  
-R 0.0.0.0:9201 -m 127.0.0.1:9400 -m 192.168.1.1:9000 -r 127.0.0.1:93:9301 \  
-R 0.0.0.0:9202 -m 127.0.0.1:9400 -m 192.168.1.1:9000 -r 127.0.0.1:93:9302 \  
-M 0.0.0.0:9401 -r 127.0.0.1:9211 -f p58 -r 127.0.0.1:9210 -v
```

4. aprsdigi (Konfiguration nur zum Test)

```
./aprsdigi -v -D --flood WIDE --trace TRACE --notx --int udp:127.0.0.1/9400/16 \  
--tx --norx --int udp:127.0.0.1/9401/16:NOCALL-10,RELAY,WIDE,TRACE
```

Startreihenfolge egal, mit & am Ende der Kommandozeile laufen die Programme im Hintergrund

Dieses Projekt ist Open Source - Haftung, Verantwortung und Spaß übernimmt jeder selbst.