

## Linux und Amateur Packet Radio



# Funkpaketpost



### Inhaltsverzeichnis

1	Einleitung .....	2
2	Das Betriebssystem GNU-Linux .....	3
3	Möglichst schnell online gehen .....	4
4	Linksammlung .....	5

---

## Einleitung

---

Amaterurfunk, Packet Radio und Linux verbindet eine gemeinsame Geschichte die bis in die Anfänge der beiden letzten zurückreicht. So gibt es seit der Version 2.0 des Linux Betriebssystemkerns einen Netzwerktreiber der tief im System verankert ist: den **AX.25** Protokoll Treiber, wobei das **A** hier tatsächlich für Amateur und nicht etwa Audio steht.

Zu dieser Zeit, in den frühen 90ern des letzten Jahrhunderts gehörte es noch zum guten Ton, dass man ein Amaterurfunkrufzeichen hatte wenn man Technik affin war. Und so sollte es nicht allzu sehr verwundern, dass Kernel Entwickler und Funkamateurl *Alan Cox* mit Rufzeichen *G4PTS*, wie man zum Beispiel aus der *man-Page* der Software *axcall* erfahren kann, unter Anderem die Quellen für den AX25 Kernel-Treiber beigesteuert hat.

*Packet Radio* selbst erblickte ein gutes Jahrzehnt vor Linux, rund um das Jahr 1978 (1) das Licht der Welt. Zu dieser Zeit kamen langsam *Personal Computer* in Umlauf die zunächst häufig mit dem Betriebssystem *DOS* ausgestattet waren. *DOS* hatte damals noch die Bedeutung von **Disk Operating System** und stand noch nicht für die gefürchtete Netzwerkattacke *Denial of Service*. Es war die Zeit der aufkommenden Telefonmodems und der Begriff des Computer Netzwerkes hatte noch nicht Eingang in die Welt der Personal Computer gefunden.

Es war aber auch die Zeit in der die Menschen Software als etwas Eigenständiges zu begreifen begannen. Dadurch, dass Software eine Existenz unabhängig von einem physischen Träger hat weil sie leicht zu kopieren ist fürchteten die Einen um die Früchte ihrer Arbeit gebracht zu werden während Andere in der Kopierbarkeit eben gerade einen großen Vorteil sahen, weil man Software dadurch beliebig oft teilen konnte und trotzdem Jeder die ganze Software behielt.

Es gab aber auch Mischformen: So erkannten findige Leute, dass man sich die leichte Kopierbarkeit zunutze machen konnte um Vertriebskosten einzusparen, wenn auch zu dem Preis, dass Nutzer der Software diese eben genauso leicht weitergeben konnten. Indem man aber nur die maschinenlesbaren Bitmuster herausgab konnte man wenigstens andere ProgrammiererInnen daran hindern dass sie die Programme verändern konnten. Die Befürchtung war nämlich, dass auf diese Weise die Arbeit die man in die Entwicklung gesteckt hatte von Anderen als die Eigene ausgegeben werden konnte und man so um die Früchte der eigenen Arbeit gebracht wäre. Weil man die Kopierbarkeit durch die Anwender ohnehin nicht effektiv unterbinden konnte, so ermutigte man die Anwender sogar solche Software weiterzugeben und auf diese Weise Werbungskosten und Vertriebskosten zu sparen. Diese Art Software nannte man "Share Ware" und deren Erzeuger hofften durch das Angebot von Premium Versionen mit essentiellen Zusatzfunktionen schließlich Gewinn zu machen. Die Verfechter der freien Variante waren übrigens nicht durchgehend Altruisten die nur der Welt etwas schenken wollten oder Hobbyisten die nur zu ihrem Vergnügen programmieren, sie hatten ganz einfach ein anderes *Geschäftsmodell*, nämlich eines das auf der Idee basierte, dass man sein Einkommen nicht dadurch erzielt, dass man einfach für die Vervielfältigung von Bitmustern entlohnt wird sondern durch Support- und Beratungsleistung der Anwender.

Warum ist diese Thematik in einer Einleitung zum Thema *Linux und Amateur Packet Radio* so wichtig? Vielen Nutzern der für Packet Radio verfügbaren Software war es nämlich offenbar genug, dass sie die Software, wie man so sagt: *gratis* bekommen konnten. Deshalb finden wir

auch heute wenn wir nach Software für Packet Radio suchen zwar noch eine Menge Programme die man sich aus dem Internet laden kann, aber leider sind für die meisten dieser Programme keine Quellcodes mehr verfügbar. Das führt nun dazu dass niemand sie an die durch die Weiterentwicklung von Hardware und Betriebssystemen veränderten Umstände anpassen kann, außer der Originalautor macht das, sofern er nicht das Interesse verloren hat oder er es nicht mehr kann weil er bereits ein SK ist.

Nun wird man zwar auch im Umfeld von *Linux* den **Gratis-Software** Typus finden, er übt sich aber hier in starker Zurückhaltung. In Linux ist der Typus **Freie-Software** deutlich stärker vertreten. Von freier Software spricht man wenn man die Freiheit hat

- die Software auszuführen, wie man möchte, für jeden Zweck,
- die Funktionsweise der Software zu untersuchen und eigenen Bedürfnissen anzupassen,
- die Software weiterzuverbreiten und damit seinen Mitmenschen zu helfen sowie
- die Software zu verbessern und diese Verbesserungen zu veröffentlichen.

Es handelt sich dabei um die von der Free Software Foundation definierten Kriterien an der man freie Software erkennt.

In dieser Artikelserie soll nun versucht werden zu zeigen wie die verschiedenen Aufgaben von *Packet Radio* ausschließlich mit freier Software realisiert werden können. Auch in diesem Zusammenhang wird man auf *veraltete* Software stoßen und auf Probleme für die (noch) keine Komponenten vorhanden sind. Aber anders als bei *Share Ware* kann man freie Software zumindest im Prinzip immer an aktuelle Verhältnisse anpassen und da Packet Radio im Linux Umfeld aus vielen kleinen Teilprogrammen zusammengesetzt wird braucht man für fehlende Funktionen nur jeweils vergleichsweise kleine Zusatzprogramme schreiben. Die Möglichkeit teil-veraltete Software zu aktualisieren wurde habe ich beispielsweise bereits selbst wahrgenommen indem ich das Software **soundmodem** (2) von Thomas Sailer aktualisiert habe.

## Das Betriebssystem GNU-Linux

---

Hier ist, hoffentlich verständlicherweise, nicht der Ort die Grundlagen des Umgangs mit Linux zu erklären. Das Internet ist aber voll mit Informationen. Wer nur mal *schnell* probieren will ob das überhaupt etwas für ihn/sie ist dem sei die *Knoppix* CD (3) ans Herz gelegt.

Es folgen nun ein paar allgemeine Ratschläge ohne in die Tiefe zu gehen.

Da Gnu-Linux nicht nach dem Prinzip: *one size fits all* funktioniert hat man immer wieder einmal die *Qual der Wahl*. Die Software Pakete von Gnu-Linux bekommt man üblicherweise als **Distribution**. Es ist für viele Neueinsteiger aber verblüffend, dass Gnu-Linux nicht *das eine Ding* ist sondern als Sammlung mit verschiedenen Schwerpunktsetzungen verfügbar ist. Trotzdem gilt häufig, dass Informationen die für eine Distribution gelten, mit Vorsicht angewandt, auch in einer anderen Distribution nützlich sein können.

Distributionen sind aber nicht der einzige Weg an Software zu kommen, in letzter Zeit kommen immer mehr Pakete in Umlauf die unabhängig von der Distribution installiert werden können. Wie auch bei anderen Betriebssystemen ist man dabei gut beraten möglichst sorgfältig die Quelle aus der man die Software installiert zu prüfen um böse Überraschungen zu vermeiden.

Dann gibt es natürlich noch den Weg den man bei *freier Software* immer beschreiten können sollte: Die Installation aus den Paketquellen.

Ich setzte in der Folge für meine *Anleitungen* die '*Debian* (4) Distribution voraus. Das ist eine der am weitesten verbreiteten Distributionen die auch die Basis für andere Distribution wie zum Beispiel *Ubuntu* ist. Trotzdem soll diese Einschränkung nicht als Wertung missverstanden werden. Ich fordere Euch, die Amateurfunk Gemeinde, auf diese Artikel um Hinweise zu ergänzen wie ein bestimmtes Problem in einer anderen Distribution zu lösen wäre, sollte es Unterschiede geben. Das hier ist ja schließlich ein *Wiki* in dem wir unser Wissen zusammentragen.

Als **Hardware** kann es für den Anfang ein nicht mehr ganz taufrischer PC durchaus tun. Ihr seid möglicherweise überrascht wenn ihr seht was in dem alten Ding noch drin steckt wenn ihr eine leichtfüßige Distribution wie zum Beispiel *Lubuntu* (5) installiert. Diejenigen, die einen Raspberry haben können entweder das originale Image installieren oder auch ein in jüngerer Zeit verfügbar gewordenes Debian Image mit dem dann *ganz normale* Debian Pakete aus dem Hauptrepository nachinstalliert werden können.

Wie bereits gesagt ist dies ein zwar interessantes aber viel zu weites Feld weshalb wir uns nun auf unser Hauptthema, '*Linux und Packet Radio* konzentrieren wollen.

## Möglichst schnell online gehen

Wer auf dieser Seite gelandet ist will wahrscheinlich möglichst schnell mit Linux ins Packet Radio Netz kommen um ein erstes Erfolgsgefühl zu erleben. Nun, je nach Voraussetzungen mag das schwerer oder leichter sein. Wer zum Beispiel von Windows kommt und von seinem Equipment bereits weiß, dass es geeignet ist, sollte auf keine großen Hürden stoßen. Anders mag das sein wenn noch keine Vorerfahrungen vorhanden sind.

In diesem Abschnitt soll deshalb versucht werden das Ziel möglichst schnell anzusteuern, wenn auch auf Kosten des tieferen Verständnisses. Ich habe vor das später nachholen.

Die erste Frage die wir beantworten müssen ist: Wo ist der nächste *Packet Radio Knoten*? Für Österreich versucht das UKW Referat eine aktuelle Liste der Digipeater (7) bereitzustellen. Aus dieser Liste ist leider nicht herauszulesen ob es sich beim angegebenen Digipeater um einen **APRS Digipeater** oder um einen **Packet Node** handelt. In diesem Wiki finden sich auf der allgemeinen [Packet Radio](#) Seite dazu Hinweise. Es ist jedenfalls eine gute Idee, sobald man eine potentielle Frequenz ermittelt hat seinen Receiver abzuhören und die Empfangslage festzustellen. Digipeater senden Baken in der Betriebsart aus in der sie auch gearbeitet werden können. Die folgenden Hörbeispiele verdeutlichen was zu erwarten ist:

1. <https://wiki.oevsv.at/wiki/Datei:afsk12.mp3>

1200 Baud Audio Frequenzmodulation (AFSK)

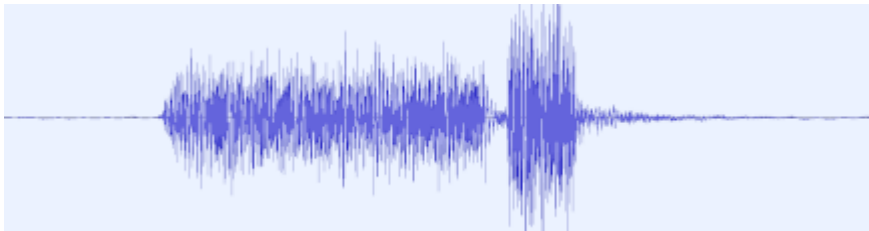
2. <https://wiki.oevsv.at/wiki/Datei:fsk48.mp3>

4800 Baud Frequenzmodulation (FSK) nach FM Demodulator

3. <https://wiki.oevsv.at/wiki/Datei:fsk96.mp3>

9600 Baud frequenzmodulation (FSK) nach FM Demodulator

Es lohnt sich darauf hinzuweisen, dass diese Hörbeispiele mit einem Mikrofon vom Lautsprecher des Receivers aufgezeichnet wurden. In der folgenden Abbildung ist deutlich ein kurzer Rauschpegel zu erkennen bevor der *Squelch* den Empfang stummschaltet:

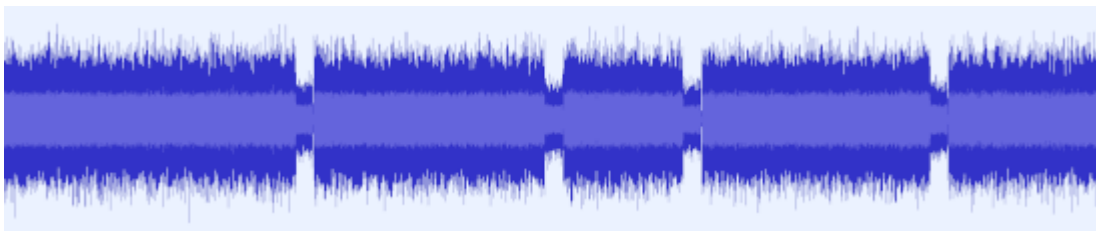


Auf eine weitere Besonderheit sollte man deshalb achten: Hört man den Empfang mit offener Rauschsperrung ab so kann man das eigentliche Signal sehr leicht überhören wie das folgende Hörbeispiel zeigt:

- [https://wiki.oevsv.at/wiki/Datei:4xfsk96\\_open\\_sq.mp3](https://wiki.oevsv.at/wiki/Datei:4xfsk96_open_sq.mp3)

9600 Baud FSK mit offener Rauschsperrung

In der folgenden Abbildung ist zu erkennen was hier passiert:



Da es sich um FM Empfang handelt sehen wir bei offener Rauschsperrung einen hohen Rauschpegel. Erst bei Vorhandensein des Trägersignales sehen wir den Signalpegel durch ein Absinken der Amplitude. Wie wir in einem vorhergehenden Beispiel gehört haben empfinden wir das *9600Bd* Signal als Rauschen. Da unser Ohr den Unterschied zwischen im Signalgehalt aber nicht wahrnehmen kann hören wir nur das Absinken der Amplitude. Ein verlässlicher Indikator ist in diesem Fall also nur der Blick auf das S-Meter unseres Transceivers.

Der nächste Schritt besteht nun darin unser Funkgerät fit zu machen. Dabei gibt es verschiedene Wege die man beschreiten kann.

1. Arbeiten mit Hardware TNC und Modem oder
2. arbeiten mit Soundkarte und Software TNC oder
3. arbeiten mit SDR und Transceiver ...

... wird fortgesetzt ...

## Linksammlung

Die (unvollständige) Sammlung der Links verweist typischerweise auf Seiten in englischer Sprache. Leider sind viele der Seiten seit langem nicht mehr gepflegt und möglicherweise auch schon vom Netz verschwunden. In diesem Fall bewährt sich häufig eine Suche im Internetarchiv (6).

- Linux Amateur Radio AX.25 HOWTO [1]

- Direwolf [\[2\]](#)
- linux-ax25.org [\[3\]](#)
- Linux und Amateur Radio [\[4\]](#)