

Inhaltsverzeichnis

1. Linux und Schmalband Packet Radio mit Terminal	14
2. Benutzer:Oe1rsa	6
3. Linux und Amateur Packet Radio	10

Linux und Schmalband Packet Radio mit Terminal

[Versionsgeschichte interaktiv durchsuchen](#)

[VisuellWikitext](#)

Version vom 22. April 2022, 18:38 Uhr (Q uelltext anzeigen)

[Oe1rsa](#) ([Diskussion](#) | [Beiträge](#))

(Neu angelegt, in Arbeit.)

Markierung: [Visuelle Bearbeitung](#)

Version vom 22. April 2022, 18:45 Uhr (Q uelltext anzeigen)

[Oe1rsa](#) ([Diskussion](#) | [Beiträge](#))

(Artikel weitergeschrieben)

Markierung: [Visuelle Bearbeitung](#)

[Zum nächsten Versionsunterschied →](#)

Zeile 20:

<pre>

Zeile 20:

<pre>

+

**sudo apt install libhamlib-utils
libhamlib-doc sox**

+

</pre>

+

+

**Wir könnten den Transceiver damit
über die CAT Schnittstelle steuern
und insbesondere die PTT Steuerung
vornehmen. In meinem Beispiel
mache ich aber davon keinen
Gebrauch sondern beschränke mich
darauf den RTS Pin der seriellen
Schnittstelle zu schalten.**

+

+

**Ein Hinweis scheint angebracht: Falls
die Standard Pegellage des RTS Pins
bei eurem Transceiver *ungünstig*
liegt, so kann es sein, dass der
Transceiver sofort in den
Sendebetrieb geht sobald das Kabel
angeschlossen wird. Schließt das
Kabel deshalb erst an sobald die
Software eingerichtet ist.**

-

**sudo apt install libhamlib-utils
libhamlib-doc sox**

+

**Generell ist es kein Fehler zunächst
einmal mit einem Multimeter die
Pegel zu überprüfen bevor man das
Funkgerät einstellt.**

```
<\pre>
```

Wir könnten den Transceiver damit über die CAT Schnittstelle steuern und insbesondere die PTT Steuerung vornehmen. In meinem Beispiel mache ich aber davon keinen Gebrauch sondern beschränke mich darauf den RTS Pin der seriellen Schnittstelle zu schalten.

+

+

+

+

+

Nun kann man das Kommando `rigctl` benutzen um manuell die PTT Funktion zu testen. Bei mir lautet das:

```
<pre>
```

```
rigctl -p /dev/ttyUSB1 -P RTS
```

```
<\pre>
```

Welches die richtige serielle Schnittstelle ist müsst ihr zuvor herausfinden. Ich empfehle dazu sich einmal anzusehen welche Schnittstellen es im System gibt solange der USB Dongle nicht angeschlossen ist. Das geht mit

+

+

+

+

+

```
<pre>
```

```
ls /dev/ttyUSB*
```

```
</pre>
```

+

Dann schließt man den Dongle an und gibt das Kommando erneut. Die Schnittstelle die dazugekommen ist muss die Gesuchte sein. Es gibt auch noch andere Methoden die dann zum Einsatz kommen sollten wenn man seine Installation dauerhaft machen will. Dazu mehr bei der Einrichtung des "soundmodem".

+

+

''' wird fortgesetzt wenn ich herausgefunden habe wie man preformatierte Texte korrekt setzt '''

Version vom 22. April 2022, 18:45 Uhr



In diesem Artikel versuche ich zu zeigen, wie man mit Linux *Bordmitteln* eine Verbindung zu einem Packet Radio Knoten aufbauen kann. Wir kommen damit zum Sendebetrieb. Die Beschreibung eines geeigneten Adapterkabels und den Empfang von Signalen habe ich im Artikel [Linux und Amateur Packet Radio](#) beschrieben. Ich werde in diesem Artikel auf Software für die man den **WINE** Emulator (1) benötigt bewusst verzichten. Nicht, dass mich jemand falsch versteht: **WINE** ist ein großartiges Projekt und es ist absolut nichts falsch daran es zu verwenden, speziell dann wenn es keine andere Lösung gibt. Wir wollen es eben mal ohne *Alkohol* versuchen, hi.

Obwohl auch auf Kurzwelle nicht ganz unmöglich, so wird Schmalband Packet Radio normalerweise im 2m und 70cm Band über Transceiver gemacht, die nur FM Modulation beherrschen. Darauf konzentrieren wir uns auch.

Sehen wir uns an, was wir benötigen, so stoßen wir darauf, dass das ein so genannter *TNC*, ein **T**erminal **N**ode **C**ontroller ist. Die Aufgabe eines TNC's ist es Daten, die wir in paketierte Form übergeben, unter Kontrolle eines Protokolles, in unserem Fall *AX.25*, an eine Gegenstelle zu übermitteln.

Solche TNC's waren ursprünglich, in den 80ern des 20. Jahrhunderts, reine Hardwarelösungen und deshalb auch recht unflexibel. Für die Abwicklung von Protokollen sind Computer prädestiniert und seitdem sie ausreichend hohe Geschwindigkeit haben können sie auch die Aufgaben der analogen Signalverarbeitung übernehmen, die in so einem TNC anfallen. Wer schon mit digitalen Übertragungsverfahren im Amateurfunk zu tun hatte weiß, dass oft eine Soundkarte ausreicht. So ist es auch in unserem Fall.

Trotzdem geht es nicht ganz ohne Hardware. Eine Verbindung zwischen Rig, also unserem Transceiver, und dem Computer muss her. Diese Verbindung besteht aus zwei Teilen:

1. Eine analoge Schnittstelle für Audio Übertragung und
2. eine digitale Schnittstelle für die *PTT* Steuerung, also die Sendertastung.

Die Beschreibung eines geeigneten Kabels findet sich, wie bereits gesagt, in dem Artikel [Linux und Amateur Packet Radio](#). Wir werden in diesem Abschnitt nun zunächst den Sendefall vorbereiten. Es ist hilfreich wenn euch zusätzlich zum Transceiver ein Kontrollempfänger zur Verfügung steht. Ein Empfänger der FM demodulieren kann ist ausreichend. Wer einen Empfänger hat der auch SSB fähig ist auf 2m oder 70cm, der kann darüberhinaus den exakten Hub seiner Aussendung einstellen. Das Verfahren dazu heisst *Besselnul* und ich werde es eventuell an anderer Stelle im Wiki beschreiben. Wie gesagt, das ist aber nicht unbedingt nötig.

Wir installieren die **hamlib** und **sox**. Dabei handelt es sich einerseits um eine Bibliothek, die es Applikationsprogrammen erlaubt verschiedene Rigs über eine einheitliche Schnittstelle anzusprechen und andererseits um ein vielseitiges Audio Werkzeug.

```
sudo apt install libhamlib-utils libhamlib-doc sox
```

Wir könnten den Transceiver damit über die CAT Schnittstelle steuern und insbesondere die PTT Steuerung vornehmen. In meinem Beispiel mache ich aber davon keinen Gebrauch sondern beschränke mich darauf den RTS Pin der seriellen Schnittstelle zu schalten.

Ein Hinweis scheint angebracht: Falls die Standard Pegellage des RTS Pins bei eurem Transceiver *ungünstig* liegt, so kann es sein, dass der Transceiver sofort in den Sendebetrieb geht sobald das Kabel angeschlossen wird. Schließt das Kabel deshalb erst an sobald die Software eingerichtet ist.

Generell ist es kein Fehler zunächst einmal mit einem Multimeter die Pegel zu überprüfen bevor man das Funkgerät einstöpselt.

Nun kann man das Kommando `rigctl` benutzen um manuell die PTT Funktion zu testen. Bei mir lautet das:

```
rigctl -p /dev/ttyUSB1 -P RTS  
<\pre>
```

Welches die richtige serielle Schnittstelle ist müsst ihr zuvor herausfinden. Ich empfehle dazu sich einmal anzusehen welche Schnittstellen es im System gibt solange der USB Dongle nicht angeschlossen ist. Das geht mit

```
<pre>  
ls /dev/ttyUSB*
```

Dann schließt man den Dongle an und gibt das Kommand erneut. Die Schnittstelle die dazugekommen ist muss die Gesuchte sein. Es gibt auch noch andere Methoden die dann zum Einsatz kommen sollten wenn man seine Installation dauerhaft machen will. Dazu mehr bei der Einrichtung des *soundmodem*.

wird fortgesetzt wenn ich herausgefunden habe wie man preformatierte Texte korrekt setzt

Linux und Schmalband Packet Radio mit Terminal: Unterschied zwischen den Versionen

[Versionsgeschichte interaktiv durchsuchen](#)

[Visuell Wikitext](#)

Version vom 22. April 2022, 18:38 Uhr (Q
uelltext anzeigen)

[Oe1rsa](#) ([Diskussion](#) | [Beiträge](#))

(Neu angelegt, in Arbeit.)

Markierung: [Visuelle Bearbeitung](#)

Version vom 22. April 2022, 18:45 Uhr (Q
uelltext anzeigen)

[Oe1rsa](#) ([Diskussion](#) | [Beiträge](#))

(Artikel weitergeschrieben)

Markierung: [Visuelle Bearbeitung](#)

[Zum nächsten Versionsunterschied →](#)

Zeile 20:

<pre>

Zeile 20:

<pre>

+

**sudo apt install libhamlib-utils
libhamlib-doc sox**

+

</pre>

+

+

**Wir könnten den Transceiver damit
über die CAT Schnittstelle steuern
und insbesondere die PTT Steuerung
vornehmen. In meinem Beispiel
mache ich aber davon keinen
Gebrauch sondern beschränke mich
darauf den RTS Pin der seriellen
Schnittstelle zu schalten.**

+

+

**Ein Hinweis scheint angebracht: Falls
die Standard Pegellage des RTS Pins
bei eurem Transceiver *ungünstig*
liegt, so kann es sein, dass der
Transceiver sofort in den
Sendebetrieb geht sobald das Kabel
angeschlossen wird. Schließt das
Kabel deshalb erst an sobald die
Software eingerichtet ist.**

**sudo apt install libhamlib-utils
libhamlib-doc sox**

-	+ Generell ist es kein Fehler zunächst einmal mit einem Multimeter die Pegel zu überprüfen bevor man das Funkgerät einstöpselt.
	+
	+ Nun kann man das Kommando <code>rigctl</code> benutzen um manuell die PTT Funktion zu testen. Bei mir lautet das:
	+
	+ <code><pre></code>
	+ <code>rigctl -p /dev/ttyUSB1 -P RTS</code>
<code><\pre></code>	<code><\pre></code>
- Wir könnten den Transceiver damit über die CAT Schnittstelle steuern und insbesondere die PTT Steuerung vornehmen. In meinem Beispiel mache ich aber davon keinen Gebrauch sondern beschränke mich darauf den RTS Pin der seriellen Schnittstelle zu schalten.	+ Welches die richtige serielle Schnittstelle ist müsst ihr zuvor herausfinden. Ich empfehle dazu sich einmal anzusehen welche Schnittstellen es im System gibt solange der USB Dongle nicht angeschlossen ist. Das geht mit
	+
	+ <code><pre></code>
	+ <code>ls /dev/ttyUSB*</code>
	+ <code></pre></code>
	+
	+ Dann schließt man den Dongle an und gibt das Kommando erneut. Die Schnittstelle die dazugekommen ist muss die Gesuchte sein. Es gibt auch noch andere Methoden die dann zum Einsatz kommen sollten wenn man seine Installation dauerhaft machen will. Dazu mehr bei der Einrichtung des "soundmodem".
	+

+ ''' wird fortgesetzt wenn ich
herausgefunden habe wie man
preformatierte Texte korrekt setzt '''

Version vom 22. April 2022, 18:45 Uhr



In diesem Artikel versuche ich zu zeigen, wie man mit Linux *Bordmitteln* eine Verbindung zu einem Packet Radio Knoten aufbauen kann. Wir kommen damit zum Sendebetrieb. Die Beschreibung eines geeigneten Adapterkabels und den Empfang von Signalen habe ich im Artikel [Linux und Amateur Packet Radio](#) beschrieben. Ich werde in diesem Artikel auf Software für die man den **WINE** Emulator (1) benötigt bewusst verzichten. Nicht, dass mich jemand falsch versteht: **WINE** ist ein großartiges Projekt und es ist absolut nichts falsch daran es zu verwenden, speziell dann wenn es keine andere Lösung gibt. Wir wollen es eben mal ohne *Alkohol* versuchen, hi.

Obwohl auch auf Kurzwelle nicht ganz unmöglich, so wird Schmalband Packet Radio normalerweise im 2m und 70cm Band über Transceiver gemacht, die nur FM Modulation beherrschen. Darauf konzentrieren wir uns auch.

Sehen wir uns an, was wir benötigen, so stoßen wir darauf, dass das ein so genannter *TNC*, ein **T**erminal **N**ode **C**ontroller ist. Die Aufgabe eines TNC's ist es Daten, die wir in paketiierter Form übergeben, unter Kontrolle eines Protokolles, in unserem Fall *AX.25*, an eine Gegenstelle zu übermitteln.

Solche TNC's waren ursprünglich, in den 80ern des 20. Jahrhunderts, reine Hardwarelösungen und deshalb auch recht unflexibel. Für die Abwicklung von Protokollen sind Computer prädestiniert und seitdem sie ausreichend hohe Geschwindigkeit haben können sie auch die Aufgaben der analogen Signalverarbeitung übernehmen, die in so einem TNC anfallen. Wer schon mit digitalen Übertragungsverfahren im Amateurfunk zu tun hatte weiß, dass oft eine Soundkarte ausreicht. So ist es auch in unserem Fall.

Trotzdem geht es nicht ganz ohne Hardware. Eine Verbindung zwischen Rig, also unserem Transceiver, und dem Computer muss her. Diese Verbindung besteht aus zwei Teilen:

1. Eine analoge Schnittstelle für Audio Übertragung und

2. eine digitale Schnittstelle für die *PTT* Steuerung, also die Sendertastung.

Die Beschreibung eines geeigneten Kabels findet sich, wie bereits gesagt, in dem Artikel [Linux und Amateur Packet Radio](#). Wir werden in diesem Abschnitt nun zunächst den Sendefall vorbereiten. Es ist hilfreich wenn euch zusätzlich zum Transceiver ein Kontrollempfänger zur Verfügung steht. Ein Empfänger der FM demodulieren kann ist ausreichend. Wer einen Empfänger hat der auch SSB fähig ist auf 2m oder 70cm, der kann darüberhinaus den exakten Hub seiner Aussendung einstellen. Das Verfahren dazu heisst **Besselnul** und ich werde es eventuell an anderer Stelle im Wiki beschreiben. Wie gesagt, das ist aber nicht unbedingt nötig.

Wir installieren die **hamlib** und **sox**. Dabei handelt es sich einerseits um eine Bibliothek, die es Applikationsprogrammen erlaubt verschiedene Rigs über eine einheitliche Schnittstelle anzusprechen und andererseits um ein vielseitiges Audio Werkzeug.

```
sudo apt install libhamlib-utils libhamlib-doc sox
```

Wir könnten den Transceiver damit über die CAT Schnittstelle steuern und insbesondere die PTT Steuerung vornehmen. In meinem Beispiel mache ich aber davon keinen Gebrauch sondern beschränke mich darauf den RTS Pin der seriellen Schnittstelle zu schalten.

Ein Hinweis scheint angebracht: Falls die Standard Pegelung des RTS Pins bei eurem Transceiver **ungünstig** liegt, so kann es sein, dass der Transceiver sofort in den Sendebetrieb geht sobald das Kabel angeschlossen wird. Schließt das Kabel deshalb erst an sobald die Software eingerichtet ist.

Generell ist es kein Fehler zunächst einmal mit einem Multimeter die Pegel zu überprüfen bevor man das Funkgerät einstöpselt.

Nun kann man das Kommando `rigctl` benutzen um manuell die PTT Funktion zu testen. Bei mir lautet das:

```
rigctl -p /dev/ttyUSB1 -P RTS  
<\pre>
```

Welches die richtige serielle Schnittstelle ist müsst ihr zuvor herausfinden. Ich empfehle dazu sich einmal anzusehen welche Schnittstellen es im System gibt solange der USB Dongle nicht angeschlossen ist. Das geht mit

```
<pre>  
ls /dev/ttyUSB*
```

Dann schließt man den Dongle an und gibt das Kommand erneut. Die Schnittstelle die dazugekommen ist muss die Gesuchte sein. Es gibt auch noch andere Methoden die dann zum Einsatz kommen sollten wenn man seine Installation dauerhaft machen will. Dazu mehr bei der Einrichtung des *soundmodem*.

wird fortgesetzt wenn ich herausgefunden habe wie man preformatierte Texte korrekt setzt

Linux und Schmalband Packet Radio mit Terminal: Unterschied zwischen den Versionen

[Versionsgeschichte interaktiv durchsuchen](#)

[Visuell Wikitext](#)

**Version vom 22. April 2022, 18:38 Uhr (Q
uelltext anzeigen)**

[Oe1rsa](#) ([Diskussion](#) | [Beiträge](#))

(Neu angelegt, in Arbeit.)

Markierung: [Visuelle Bearbeitung](#)

**Version vom 22. April 2022, 18:45 Uhr (Q
uelltext anzeigen)**

[Oe1rsa](#) ([Diskussion](#) | [Beiträge](#))

(Artikel weitergeschrieben)

Markierung: [Visuelle Bearbeitung](#)

[Zum nächsten Versionsunterschied →](#)

Zeile 20:

```
<pre>
```

Zeile 20:

```
<pre>
```

+ **sudo apt install libhamlib-utils
libhamlib-doc sox**

+ **</pre>**

+

+

+

+

**Wir könnten den Transceiver damit
über die CAT Schnittstelle steuern
und insbesondere die PTT Steuerung
vornehmen. In meinem Beispiel
mache ich aber davon keinen
Gebrauch sondern beschränke mich
darauf den RTS Pin der seriellen
Schnittstelle zu schalten.**

**Ein Hinweis scheint angebracht: Falls
die Standard Pegellage des RTS Pins
bei eurem Transceiver *ungünstig*
liegt, so kann es sein, dass der
Transceiver sofort in den
Sendebetrieb geht sobald das Kabel
angeschlossen wird. Schließt das
Kabel deshalb erst an sobald die
Software eingerichtet ist.**

```
sudo apt install libhamlib-utils  
libhamlib-doc sox
```

-	+ Generell ist es kein Fehler zunächst einmal mit einem Multimeter die Pegel zu überprüfen bevor man das Funkgerät einstöpselt.
	+
	+ Nun kann man das Kommando <code>rigctl</code> benutzen um manuell die PTT Funktion zu testen. Bei mir lautet das:
	+
	+ <code><pre></code>
	+ <code>rigctl -p /dev/ttyUSB1 -P RTS</code>
<code><\pre></code>	<code><\pre></code>
- Wir könnten den Transceiver damit über die CAT Schnittstelle steuern und insbesondere die PTT Steuerung vornehmen. In meinem Beispiel mache ich aber davon keinen Gebrauch sondern beschränke mich darauf den RTS Pin der seriellen Schnittstelle zu schalten.	+ Welches die richtige serielle Schnittstelle ist müsst ihr zuvor herausfinden. Ich empfehle dazu sich einmal anzusehen welche Schnittstellen es im System gibt solange der USB Dongle nicht angeschlossen ist. Das geht mit
	+
	+ <code><pre></code>
	+ <code>ls /dev/ttyUSB*</code>
	+ <code></pre></code>
	+
	+ Dann schließt man den Dongle an und gibt das Kommando erneut. Die Schnittstelle die dazugekommen ist muss die Gesuchte sein. Es gibt auch noch andere Methoden die dann zum Einsatz kommen sollten wenn man seine Installation dauerhaft machen will. Dazu mehr bei der Einrichtung des "soundmodem".
	+

''' wird fortgesetzt wenn ich
+ herausgefunden habe wie man
preformatierte Texte korrekt setzt '''

Version vom 22. April 2022, 18:45 Uhr



In diesem Artikel versuche ich zu zeigen, wie man mit Linux *Bordmitteln* eine Verbindung zu einem Packet Radio Knoten aufbauen kann. Wir kommen damit zum Sendebetrieb. Die Beschreibung eines geeigneten Adapterkabels und den Empfang von Signalen habe ich im Artikel [Linux und Amateur Packet Radio](#) beschrieben. Ich werde in diesem Artikel auf Software für die man den **WINE** Emulator (1) benötigt bewusst verzichten. Nicht, dass mich jemand falsch versteht: **WINE** ist ein großartiges Projekt und es ist absolut nichts falsch daran es zu verwenden, speziell dann wenn es keine andere Lösung gibt. Wir wollen es eben mal ohne *Alkohol* versuchen, hi.

Obwohl auch auf Kurzwelle nicht ganz unmöglich, so wird Schmalband Packet Radio normalerweise im 2m und 70cm Band über Transceiver gemacht, die nur FM Modulation beherrschen. Darauf konzentrieren wir uns auch.

Sehen wir uns an, was wir benötigen, so stoßen wir darauf, dass das ein so genannter *TNC*, ein **T**erminal **N**ode **C**ontroller ist. Die Aufgabe eines TNC's ist es Daten, die wir in paketiierter Form übergeben, unter Kontrolle eines Protokolles, in unserem Fall *AX.25*, an eine Gegenstelle zu übermitteln.

Solche TNC's waren ursprünglich, in den 80ern des 20. Jahrhunderts, reine Hardwarelösungen und deshalb auch recht unflexibel. Für die Abwicklung von Protokollen sind Computer prädestiniert und seitdem sie ausreichend hohe Geschwindigkeit haben können sie auch die Aufgaben der analogen Signalverarbeitung übernehmen, die in so einem TNC anfallen. Wer schon mit digitalen Übertragungsverfahren im Amateurfunk zu tun hatte weiß, dass oft eine Soundkarte ausreicht. So ist es auch in unserem Fall.

Trotzdem geht es nicht ganz ohne Hardware. Eine Verbindung zwischen Rig, also unserem Transceiver, und dem Computer muss her. Diese Verbindung besteht aus zwei Teilen:

1. Eine analoge Schnittstelle für Audio Übertragung und

2. eine digitale Schnittstelle für die *PTT* Steuerung, also die Sendertastung.

Die Beschreibung eines geeigneten Kabels findet sich, wie bereits gesagt, in dem Artikel [Linux und Amateur Packet Radio](#). Wir werden in diesem Abschnitt nun zunächst den Sendefall vorbereiten. Es ist hilfreich wenn euch zusätzlich zum Transceiver ein Kontrollempfänger zur Verfügung steht. Ein Empfänger der FM demodulieren kann ist ausreichend. Wer einen Empfänger hat der auch SSB fähig ist auf 2m oder 70cm, der kann darüberhinaus den exakten Hub seiner Aussendung einstellen. Das Verfahren dazu heisst **Besselnul** und ich werde es eventuell an anderer Stelle im Wiki beschreiben. Wie gesagt, das ist aber nicht unbedingt nötig.

Wir installieren die **hamlib** und **sox**. Dabei handelt es sich einerseits um eine Bibliothek, die es Applikationsprogrammen erlaubt verschiedene Rigs über eine einheitliche Schnittstelle anzusprechen und andererseits um ein vielseitiges Audio Werkzeug.

```
sudo apt install libhamlib-utils libhamlib-doc sox
```

Wir könnten den Transceiver damit über die CAT Schnittstelle steuern und insbesondere die PTT Steuerung vornehmen. In meinem Beispiel mache ich aber davon keinen Gebrauch sondern beschränke mich darauf den RTS Pin der seriellen Schnittstelle zu schalten.

Ein Hinweis scheint angebracht: Falls die Standard Pegelung des RTS Pins bei eurem Transceiver **ungünstig** liegt, so kann es sein, dass der Transceiver sofort in den Sendebetrieb geht sobald das Kabel angeschlossen wird. Schließt das Kabel deshalb erst an sobald die Software eingerichtet ist.

Generell ist es kein Fehler zunächst einmal mit einem Multimeter die Pegel zu überprüfen bevor man das Funkgerät einstöpselt.

Nun kann man das Kommando `rigctl` benutzen um manuell die PTT Funktion zu testen. Bei mir lautet das:

```
rigctl -p /dev/ttyUSB1 -P RTS
<\pre>
```

Welches die richtige serielle Schnittstelle ist müsst ihr zuvor herausfinden. Ich empfehle dazu sich einmal anzusehen welche Schnittstellen es im System gibt solange der USB Dongle nicht angeschlossen ist. Das geht mit

```
<pre>
ls /dev/ttyUSB*
```

Dann schließt man den Dongle an und gibt das Kommando erneut. Die Schnittstelle die dazugekommen ist muss die Gesuchte sein. Es gibt auch noch andere Methoden die dann zum Einsatz kommen sollten wenn man seine Installation dauerhaft machen will. Dazu mehr bei der Einrichtung des *soundmodem*.

wird fortgesetzt wenn ich herausgefunden habe wie man preformatierte Texte korrekt setzt

Linux und Schmalband Packet Radio mit Terminal: Unterschied zwischen den Versionen

[Versionsgeschichte interaktiv durchsuchen](#)

[Visuell Wikitext](#)

**Version vom 22. April 2022, 18:38 Uhr (Q
uelltext anzeigen)**

[Oe1rsa](#) ([Diskussion](#) | [Beiträge](#))

(Neu angelegt, in Arbeit.)

Markierung: [Visuelle Bearbeitung](#)

**Version vom 22. April 2022, 18:45 Uhr (Q
uelltext anzeigen)**

[Oe1rsa](#) ([Diskussion](#) | [Beiträge](#))

(Artikel weitergeschrieben)

Markierung: [Visuelle Bearbeitung](#)

[Zum nächsten Versionsunterschied →](#)

Zeile 20:

```
<pre>
```

Zeile 20:

```
<pre>
```

+ **sudo apt install libhamlib-utils
libhamlib-doc sox**

+ **</pre>**

+

+

+

+

**Wir könnten den Transceiver damit
über die CAT Schnittstelle steuern
und insbesondere die PTT Steuerung
vornehmen. In meinem Beispiel
mache ich aber davon keinen
Gebrauch sondern beschränke mich
darauf den RTS Pin der seriellen
Schnittstelle zu schalten.**

**Ein Hinweis scheint angebracht: Falls
die Standard Pegellage des RTS Pins
bei eurem Transceiver *ungünstig*
liegt, so kann es sein, dass der
Transceiver sofort in den
Sendebetrieb geht sobald das Kabel
angeschlossen wird. Schließt das
Kabel deshalb erst an sobald die
Software eingerichtet ist.**

```
sudo apt install libhamlib-utils  
libhamlib-doc sox
```

-	+ Generell ist es kein Fehler zunächst einmal mit einem Multimeter die Pegel zu überprüfen bevor man das Funkgerät einstöpselt.
	+
	+ Nun kann man das Kommando <code>rigctl</code> benutzen um manuell die PTT Funktion zu testen. Bei mir lautet das:
	+
	+ <code><pre></code>
	+ <code>rigctl -p /dev/ttyUSB1 -P RTS</code>
<code><\pre></code>	<code><\pre></code>
- Wir könnten den Transceiver damit über die CAT Schnittstelle steuern und insbesondere die PTT Steuerung vornehmen. In meinem Beispiel mache ich aber davon keinen Gebrauch sondern beschränke mich darauf den RTS Pin der seriellen Schnittstelle zu schalten.	+ Welches die richtige serielle Schnittstelle ist müsst ihr zuvor herausfinden. Ich empfehle dazu sich einmal anzusehen welche Schnittstellen es im System gibt solange der USB Dongle nicht angeschlossen ist. Das geht mit
	+
	+ <code><pre></code>
	+ <code>ls /dev/ttyUSB*</code>
	+ <code></pre></code>
	+
	+ Dann schließt man den Dongle an und gibt das Kommando erneut. Die Schnittstelle die dazugekommen ist muss die Gesuchte sein. Es gibt auch noch andere Methoden die dann zum Einsatz kommen sollten wenn man seine Installation dauerhaft machen will. Dazu mehr bei der Einrichtung des "soundmodem".
	+

''' wird fortgesetzt wenn ich
+ herausgefunden habe wie man
preformatierte Texte korrekt setzt '''

Version vom 22. April 2022, 18:45 Uhr



In diesem Artikel versuche ich zu zeigen, wie man mit Linux *Bordmitteln* eine Verbindung zu einem Packet Radio Knoten aufbauen kann. Wir kommen damit zum Sendebetrieb. Die Beschreibung eines geeigneten Adapterkabels und den Empfang von Signalen habe ich im Artikel [Linux und Amateur Packet Radio](#) beschrieben. Ich werde in diesem Artikel auf Software für die man den **WINE** Emulator (1) benötigt bewusst verzichten. Nicht, dass mich jemand falsch versteht: **WINE** ist ein großartiges Projekt und es ist absolut nichts falsch daran es zu verwenden, speziell dann wenn es keine andere Lösung gibt. Wir wollen es eben mal ohne *Alkohol* versuchen, hi.

Obwohl auch auf Kurzwelle nicht ganz unmöglich, so wird Schmalband Packet Radio normalerweise im 2m und 70cm Band über Transceiver gemacht, die nur FM Modulation beherrschen. Darauf konzentrieren wir uns auch.

Sehen wir uns an, was wir benötigen, so stoßen wir darauf, dass das ein so genannter *TNC*, ein **T**erminal **N**ode **C**ontroller ist. Die Aufgabe eines TNC's ist es Daten, die wir in paketiierter Form übergeben, unter Kontrolle eines Protokolles, in unserem Fall *AX.25*, an eine Gegenstelle zu übermitteln.

Solche TNC's waren ursprünglich, in den 80ern des 20. Jahrhunderts, reine Hardwarelösungen und deshalb auch recht unflexibel. Für die Abwicklung von Protokollen sind Computer prädestiniert und seitdem sie ausreichend hohe Geschwindigkeit haben können sie auch die Aufgaben der analogen Signalverarbeitung übernehmen, die in so einem TNC anfallen. Wer schon mit digitalen Übertragungsverfahren im Amateurfunk zu tun hatte weiß, dass oft eine Soundkarte ausreicht. So ist es auch in unserem Fall.

Trotzdem geht es nicht ganz ohne Hardware. Eine Verbindung zwischen Rig, also unserem Transceiver, und dem Computer muss her. Diese Verbindung besteht aus zwei Teilen:

1. Eine analoge Schnittstelle für Audio Übertragung und

2. eine digitale Schnittstelle für die *PTT* Steuerung, also die Sendertastung.

Die Beschreibung eines geeigneten Kabels findet sich, wie bereits gesagt, in dem Artikel [Linux und Amateur Packet Radio](#). Wir werden in diesem Abschnitt nun zunächst den Sendefall vorbereiten. Es ist hilfreich wenn euch zusätzlich zum Transceiver ein Kontrollempfänger zur Verfügung steht. Ein Empfänger der FM demodulieren kann ist ausreichend. Wer einen Empfänger hat der auch SSB fähig ist auf 2m oder 70cm, der kann darüberhinaus den exakten Hub seiner Aussendung einstellen. Das Verfahren dazu heisst **Besselnul** und ich werde es eventuell an anderer Stelle im Wiki beschreiben. Wie gesagt, das ist aber nicht unbedingt nötig.

Wir installieren die **hamlib** und **sox**. Dabei handelt es sich einerseits um eine Bibliothek, die es Applikationsprogrammen erlaubt verschiedene Rigs über eine einheitliche Schnittstelle anzusprechen und andererseits um ein vielseitiges Audio Werkzeug.

```
sudo apt install libhamlib-utils libhamlib-doc sox
```

Wir könnten den Transceiver damit über die CAT Schnittstelle steuern und insbesondere die PTT Steuerung vornehmen. In meinem Beispiel mache ich aber davon keinen Gebrauch sondern beschränke mich darauf den RTS Pin der seriellen Schnittstelle zu schalten.

Ein Hinweis scheint angebracht: Falls die Standard Pegelung des RTS Pins bei eurem Transceiver **ungünstig** liegt, so kann es sein, dass der Transceiver sofort in den Sendebetrieb geht sobald das Kabel angeschlossen wird. Schließt das Kabel deshalb erst an sobald die Software eingerichtet ist.

Generell ist es kein Fehler zunächst einmal mit einem Multimeter die Pegel zu überprüfen bevor man das Funkgerät einstöpselt.

Nun kann man das Kommando `rigctl` benutzen um manuell die PTT Funktion zu testen. Bei mir lautet das:

```
rigctl -p /dev/ttyUSB1 -P RTS  
<\pre>
```

Welches die richtige serielle Schnittstelle ist müsst ihr zuvor herausfinden. Ich empfehle dazu sich einmal anzusehen welche Schnittstellen es im System gibt solange der USB Dongle nicht angeschlossen ist. Das geht mit

```
<pre>  
ls /dev/ttyUSB*
```

Dann schließt man den Dongle an und gibt das Kommand erneut. Die Schnittstelle die dazugekommen ist muss die Gesuchte sein. Es gibt auch noch andere Methoden die dann zum Einsatz kommen sollten wenn man seine Installation dauerhaft machen will. Dazu mehr bei der Einrichtung des *soundmodem*.

wird fortgesetzt wenn ich herausgefunden habe wie man preformatierte Texte korrekt setzt