

Inhaltsverzeichnis

1. TCE Komponenten	27
2. Benutzer:OE2WAO	6
3. Packet Radio via Soundkarte unter Linux	10
4. TCE Software	31

TCE Komponenten

[Versionsgeschichte interaktiv durchsuchen](#)
[Visuell Wikitext](#)

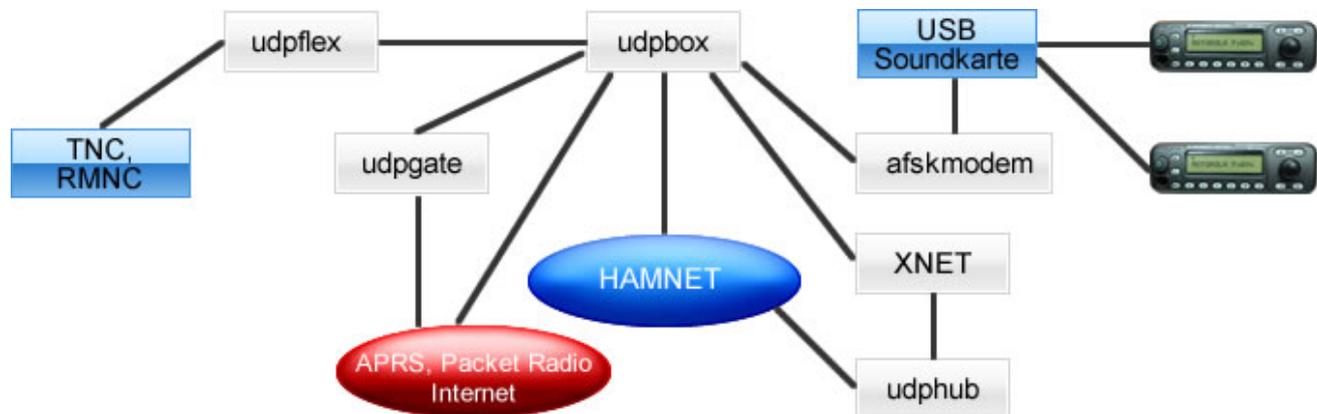
Version vom 8. Dezember 2014, 19:27 Uhr (Quelltext anzeigen)
 OE2WAO ([Diskussion](#) | [Beiträge](#))
[← Zum vorherigen Versionsunterschied](#)

Aktuelle Version vom 16. August 2015, 12:04 Uhr (Quelltext anzeigen)
 OE2WAO ([Diskussion](#) | [Beiträge](#))
[K \(→udpgate\)](#)

Zeile 17:	==udpgate==	Zeile 17:	==udpgate==
–	[[Datei:XZR-conn.PNG 200px thumb left Connection Tab sample]]	+	[[Datei:XZR-conn.PNG 200px thumb Connection Tab sample]]
	Das UDPGATE ist ein APRS Server, welcher die Netzwerkebene des APRS Datentransports übernimmt. Also bspw. die Serverfunktionalität für Benutzer bereitstellen, sowie eine Verbindung zum APRS IS oder nächsten APRS Server (UDPGATE) herstellen. 		Das UDPGATE ist ein APRS Server, welcher die Netzwerkebene des APRS Datentransports übernimmt. Also bspw. die Serverfunktionalität für Benutzer bereitstellen, sowie eine Verbindung zum APRS IS oder nächsten APRS Server (UDPGATE) herstellen.
	Dabei agiert er bei der Verbindung nach der Priorität der Einträge in der Serverliste. Ist der erste Server nicht erreichbar, wird der nächste Server in der Liste versucht zu erreichen. In regelmässigen Abständen wird jedoch erneut versucht, die in der Liste zuvorgelegenen Server nach dem Prioritätsprinzip zu erreichen, und verlustfrei wieder rückzuverbinden. 		Dabei agiert er bei der Verbindung nach der Priorität der Einträge in der Serverliste. Ist der erste Server nicht erreichbar, wird der nächste Server in der Liste versucht zu erreichen. In regelmässigen Abständen wird jedoch erneut versucht, die in der Liste zuvorgelegenen Server nach dem Prioritätsprinzip zu erreichen, und verlustfrei wieder rückzuverbinden.
	Das Modul verfügt über ein eigenes Webinterface welches default unter "serverIP:14501" erreichbar ist. 		Das Modul verfügt über ein eigenes Webinterface welches default unter "serverIP:14501" erreichbar ist.
–			
–			
–			
	==udprfnet==		==udprfnet==

Aktuelle Version vom 16. August 2015, 12:04 Uhr

Im AFU Tinycore Image sind unter anderem amateurfunkspezifische Programme enthalten.



Inhaltsverzeichnis

1	udpbox	29
2	udphub	29
3	udpgate	29
4	udprfnet	29
5	afskmodem	30

udpbox

Die UDPBOX stellt das zentrale Bindeglied zwischen den einzelnen Programmen dar. Sie empfängt und verteilt entsprechend die UDP Pakete.

So ist es bspw. möglich die auf 2m empfangenen APRS Pakete zu filtern, auf 2m wieder auszugeben, und zusätzlich alle (oder gefilterte) APRS Meldungen auf dem Dualband 70cm Packet Radio Digipeater auszusenden.

Zudem beherrscht die UDPBOX die leicht unterschiedlichen Arten in den Protokollen AX25 und TNC2 MONITOR.

Zur Übersicht steht für die APRS Funktion auch ein kleiner Webserver bereit: [http:// HOSTNAME:14501](http://HOSTNAME:14501)

udphub

Der UDPHUB ist ein Hilfsprogramm für XNET, welches die IP Beschränkung umgeht, indem es sich selbst zwischen Benutzer und XNET stellt, und die AXUDP Pakete entsprechend verteilt. Dabei bleibt der Ursprungspfad (IP) des Benutzer eine Woche (einstellbar) gespeichert, und der Benutzer kann bei lokal gestartetem Programm auch ohne aktiven Connect in dieser Zeitspanne von anderen Benutzern kontaktiert werden, genauso als ob man per HF QRV wäre.

udpgate

Das UDPGATE ist ein APRS Server, welcher die Netzwerkebene des APRS Datentransports übernimmt. Also bspw. die Serverfunktionalität für Benutzer bereitstellen, sowie eine Verbindung zum APRS IS oder nächsten APRS Server (UDPGATE) herstellen.

Dabei agiert er bei der Verbindung nach der Priorität der Einträge in der Serverliste. Ist der erste Server nicht erreichbar, wird der nächste Server in der Liste versucht zu erreichen. In regelmässigen Abständen wird jedoch erneut versucht, die in der Liste zugelegenen Server nach dem Prioritätsprinzip zu erreichen, und verlustfrei wieder rückzuverbinden.

Das Modul verfügt über ein eigenes Webinterface welches default unter "serverIP:14501" erreichbar ist.

Server: DE2XZR-10 Port: 14580 [udpgate 0.46] Maxusers: 50 http://151 Uptime: 16d16:26:44

ID	IP	Port	Call	Status	Source	Target	TX														
1	127.0.0.1	14501	DE2XZR-10	OK	14580	14580	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	192.168.1.1	14501	DE2XZR-10	OK	14580	14580	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	192.168.1.2	14501	DE2XZR-10	OK	14580	14580	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	192.168.1.3	14501	DE2XZR-10	OK	14580	14580	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	192.168.1.4	14501	DE2XZR-10	OK	14580	14580	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	192.168.1.5	14501	DE2XZR-10	OK	14580	14580	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	192.168.1.6	14501	DE2XZR-10	OK	14580	14580	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	192.168.1.7	14501	DE2XZR-10	OK	14580	14580	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	192.168.1.8	14501	DE2XZR-10	OK	14580	14580	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	192.168.1.9	14501	DE2XZR-10	OK	14580	14580	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11	192.168.1.10	14501	DE2XZR-10	OK	14580	14580	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12	192.168.1.11	14501	DE2XZR-10	OK	14580	14580	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
13	192.168.1.12	14501	DE2XZR-10	OK	14580	14580	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14	192.168.1.13	14501	DE2XZR-10	OK	14580	14580	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
15	192.168.1.14	14501	DE2XZR-10	OK	14580	14580	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
16	192.168.1.15	14501	DE2XZR-10	OK	14580	14580	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
17	192.168.1.16	14501	DE2XZR-10	OK	14580	14580	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
18	192.168.1.17	14501	DE2XZR-10	OK	14580	14580	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
19	192.168.1.18	14501	DE2XZR-10	OK	14580	14580	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
20	192.168.1.19	14501	DE2XZR-10	OK	14580	14580	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
21	192.168.1.20	14501	DE2XZR-10	OK	14580	14580	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Connection Tab sample

udprfnet

Das UDPRFNET Modul ist eine experimentelle Software für eine intelligente APRS Paketverteilung unter Digipeatern. Ziel ist das gesamte Netz als einen großen RX darzustellen und auch weiter entfernt empfangene APRS Pakete vom dort gebietsmässig nahegelegenen Digi per Radiusdefinition auf 144.800MHz wieder aussenden zu lassen. Dabei bilden mehrere Serververbindungen untereinander das Prinzip der Redundanz.

afskmodem

Das AFSKMODEM ist ein [digitales Soundmodem](#), welches die Pakete in eine (A)FSK Modulation wandelt und der Soundkarte zuführt. Der Name soll jedoch nicht verwirren, es sind auch je nach Soundkarte Geschwindigkeiten > 28kBaude FSK möglich.

[<< zurück zu Einstellungen & Bedienung](#)

TCE Komponenten und Benutzer:OE2WAO: Unterschied zwischen den Seiten

VisuellWikitext

Version vom 8. Dezember 2014, 19:27 Uhr (Quelltext anzeigen)
 OE2WAO ([Diskussion](#) | [Beiträge](#))

Aktuelle Version vom 9. August 2020, 23:41 Uhr (Quelltext anzeigen)
 OE2WAO ([Diskussion](#) | [Beiträge](#))
 (Die Seite wurde neu angelegt:
 „https://oe2wao.info“)

Zeile 1:

- **Im AFU Tincore Image sind unter anderem amateurfunkspezifische Programme enthalten.**

-
-
- **[[Datei:Udpboxes.jpg]]**
-
- **==udpbox==**
-
- **Die UDPBOX stellt das zentrale Bindeglied zwischen den einzelnen Programmen dar. Sie empfängt und verteilt entsprechend die UDP Pakete.**

- **So ist es bspw. möglich die auf 2m empfangenen APRS Pakete zu filtern, auf 2m wieder auszugeben, und zusätzlich alle (oder gefilterte) APRS Meldungen auf dem Dualbaud 70cm Packet Radio Digipeater auszusenden.**

- **Zudem beherrscht die UDPBOX die leicht unterschiedlichen Arten in den Protokollen AX25 und TNC2 MONITOR.**

- **Zur Übersicht steht für die APRS Funktion auch ein kleiner Webserver bereit: http:// HOSTNAME:14501**

Zeile 1:

https://oe2wao.info

+

-

==udphub==

-

Der UDPHUB ist ein Hilfsprogramm für XNET, welches die IP Beschränkung umgeht, indem es sich selbst zwischen Benutzer und XNET stellt, und die AXUDP Pakete entsprechend verteilt. Dabei bleibt der Ursprungspfad (IP) des Benutzer eine Woche (einstellbar) gespeichert, und der Benutzer kann bei lokal gestartetem Programm auch ohne aktiven Connect in dieser Zeitspanne von anderen Benutzern kontaktiert werden, genauso als ob man per HF QRV wäre.

-

==udpgate==

-

[[Datei:XZR-conn.
PNG|200px|thumb|left|Connection Tab sample]]

-

**Das UDPGATE ist ein APRS Server, welcher die Netzwerkebene des APRS Datentransports übernimmt. Also bspw. die Serverfunktionalität für Benutzer bereitstellen, sowie eine Verbindung zum APRS IS oder nächsten APRS Server (UDPGATE) herstellen.
**

-

**Dabei agiert er bei der Verbindung nach der Priorität der Einträge in der Serverliste. Ist der erste Server nicht erreichbar, wird der nächste Server in der Liste versucht zu erreichen. In regelmäßigen Abständen wird jedoch erneut versucht, die in der Liste zuvorgelegenen Server nach dem Prioritätsprinzip zu erreichen, und verlustfrei wieder rückzuverbinden.
**

– Das Modul verfügt über ein eigenes Webinterface welches default unter "serverIP:14501" erreichbar ist.

–

–

–

–

– ==udprfnet==

–

– Das UDPRFNET Modul ist eine experimentelle Software für eine intelligente APRS Paketverteilung unter Digipeatern. Ziel ist das gesamte Netz als einen großen RX darzustellen und auch weiter entfernt empfangene APRS Pakete vom dort gebietsmässig nahegelegenen Digi per Radiusdefinition auf 144.800MHz wieder aussenden zu lassen. Dabei bilden mehrere Serververbindungen untereinander das Prinzip der Redundanz.

–

– ==afskmodem==

–

– Das AFSKMODEM ist ein [[Packet Radio via Soundkarte unter Linux | digitales Soundmodem]], welches die Pakete in eine (A)FSK Modulation wandelt und der Soundkarte zuführt. Der Name soll jedoch nicht verwirren, es sind auch je nach Soundkarte Geschwindigkeiten > 28kBaud FSK möglich.

–

–

– [[TCE Software | << zurück zu Einstellungen & Bedienung]]

Aktuelle Version vom 9. August 2020, 23:41 Uhr

<https://oe2wao.info>

TCE Komponenten und Packet Radio via Soundkarte unter Linux: Unterschied zwischen den Seiten

VisuellWikitext

Version vom 8. Dezember 2014, 19:27 Uhr (Quelltext anzeigen)
 OE2WAO ([Diskussion](#) | [Beiträge](#))

Aktuelle Version vom 22. November 2019, 18:48 Uhr (Quelltext anzeigen)
 OE5HPM ([Diskussion](#) | [Beiträge](#))
 (→[Der Source Code](#))

<p>Zeile 1:</p> <p>– Im AFU Tyncore Image sind unter anderem amateurfunkspezifische Programme enthalten.
</p>	<p>Zeile 1:</p> <p>+ [[Kategorie:Packet-Radio und I-Gate]]</p> <p>+ == Das Projekt ==</p> <p>+ Dieser (USB) Soundkartentreiber von OE5DXL soll es ermöglichen mit 2 Kanälen (L und R der Soundkarte) mehrere Modems zugleich unter Linux zu initialisieren.
</p> <p>+ 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.
</p> <p>+ Weiterer Vorteil ist die Möglichkeit des ""Multibaud"" Digl, also mehrere Geschwindigkeiten FSK AFSK gemischt (bspw. 1k2 2k4 4k8 9k6 auf einer QRG).</p> <p>+ In Stereo kann so theoretisch ein multibaud FSK AFSK KISS, als auch AXUDP AX.25 Modem betrieben werden.</p> <p>+ [[Bild:Soundmodem-box.gif Soundmodem Schema]]</p> <p>+ ==Der Source Code==</p>
<p>– [[Datei:Udpboxs.jpg]]</p>	

				<p>[https://github.com/oe5hpm/dxIAPRS] https://github.com/oe5hpm/dxIAPRS
</p>
				<p>[http://gitlab.oe5xbl.ampr.org/oe5hpm/dxIAPRS/]http://gitlab.oe5xbl.ampr.org/oe5hpm/dxIAPRS/</p>
-	==udpbox==	+	==Der kompilierte Treiber==	
				<p>[[Media:soundmodem i386 linux.zip Soundmodem-bin]] - Der fertig kompilierte Soundmodem Treiber</p>
-	<p>Die UDPBOX stellt das zentrale Bindeglied zwischen den einzelnen Programmen dar. Sie empfängt und verteilt entsprechend die UDP Pakete.</p> <p>
</p>	+	<p>[[Media:udpbox i386 linux bin.zip udpbox-bin]] - UDP Filter und RAW-Monitor Konverter und (neu) mit aprs-digi, Bake, User-Message-Receiver</p>	
-	<p>So ist es bspw. möglich die auf 2m empfangenen APRS Pakete zu filtern, auf 2m wieder auszugeben, und zusätzlich alle (oder gefilterte) APRS Meldungen auf dem Dualband 70cm Packet Radio Digipeater auszusenden.</p> <p>
</p>			
-	<p>Zudem beherrscht die UDPBOX die leicht unterschiedlichen Arten in den Protokollen AX25 und TNC2 MONITOR.</p> <p>
</p>			
-	<p>Zur Übersicht steht für die APRS Funktion auch ein kleiner Webserver bereit: http:// HOSTNAME:14501</p>			
-	==udphub==	+	<p>[[Media:udpgate i386 linux bin.zip udpgate-bin]] - I-Gate, APRS-IS, APRS-Server mit Rangefilter, HTML-Statistik, Log</p>	

<p>Der UDPHUB ist ein Hilfsprogramm für XNET, welches die IP Beschränkung umgeht, indem es sich selbst zwischen Benutzer und XNET stellt, und die AXUDP Pakete entsprechend verteilt. Dabei bleibt der Ursprungspfad (IP) des Benutzer eine Woche (einstellbar) gespeichert, und der Benutzer kann bei lokal gestartetem Programm auch ohne aktiven Connect in dieser Zeitspanne von anderen Benutzern kontaktiert werden, genauso als ob man per HF QRV wäre.</p>	<p>[[Media:udphub i386 linux bin.zip udphub-bin]] - axudp Hub zum HAMNET-PR-Login ohne IP Beschränkung</p>
<p>==udpgate==</p>	<p>[[Media:udpflex i386 linux bin.zip udpflex-bin]] - Interface com-port (/dev/ttySxx) mit KISS oder RMNC bidirektional auf axudp</p>
<p>[[Datei:XZR-conn.PNG 200px thumb left Connection Tab sample]]</p>	<p>==Starten bzw. Aufrufen des Treibers==</p>
<p>Das UDPGATE ist ein APRS Server, welcher die Netzwerkebene des APRS Datentransports übernimmt. Also bspw. die Serverfunktionalität für Benutzer bereitstellen, sowie eine Verbindung zum APRS IS oder nächsten APRS Server (UDPGATE) herstellen.</p>	<p>mit oss testen 1200 + 9600 baud monitor (ohne kiss oder udp)</p>
<p>Dabei agiert er bei der Verbindung nach der Priorität der Einträge in der Serverliste. Ist der erste Server nicht erreichbar, wird der nächste Server in der Liste versucht zu erreichen. In regelmäßigen Abständen wird jedoch erneut versucht, die in der Liste zuvorgelegenen Server nach dem Prioritätsprinzip zu erreichen, und verlustfrei wieder rückzuverbinden.</p>	<p>./afskmodem -f 32000 -M 0 -c 0 -b 1200 -M 1 -c 0 -b 9600 -a -g</p>

- Das Modul verfügt über ein eigenes Webinterface welches default unter "serverIP:14501" erreichbar ist.

+ mit als:

+ 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-Diqi 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 -q -q 200 -U 127.0.0.1:9202/9212 -m 0

		+ <code>linuxsnet (XNET) AUTOEXEC.NET</code>
		+ <code>attach ip0 axudp 1 1 I9200 d9210 127.0.0.1</code>
		+ <code>attach ip1 axudp 2 1 I9201 d9211 127.0.0.1</code>
		+ <code>attach ip2 axudp 3 1 I9202 d9212 127.0.0.1</code>
		+ <code>po 1 baud 1200</code>
		+ <code>po 2 baud 1200</code>
		+ <code>po 3 baud 9600</code>
-	<code>==udprfnet==</code>	
-	<p>Das UDPRFNET Modul ist eine experimentelle Software für eine intelligente APRS Paketverteilung unter Digipeatern. Ziel ist das gesamte Netz als einen großen RX darzustellen und auch weiter entfernt empfangene APRS Pakete vom dort gebietsmässig nahegelegenen Digi per Radiusdefinition auf 144.800MHz wieder aussenden zu lassen. Dabei bilden mehrere Serververbindungen untereinander das Prinzip der Redundanz.</p>	<p>XNET mit KISS und TTY ptt (XNET nach dem Modem starten!)</p>
		+ <code>aoss ./afskmodem \</code>
		+ <code>-t /dev/ttyS0 -f 24000 -i /tmp /soundmodem \</code>
		+ <code>-c 2 -s 9 -l 256 -b 6 -e 7 -C 0 -r 300 \</code>
		+ <code>-M 0 -c 1 -b 1200 -q 200 -m 0 \</code>
		+ <code>-M 1 -c 0 -b 1200 -H 40 -q 200 -m 0 \</code>
		+ <code>-M 2 -c 0 -b 9600 -a -g -q 200 -m 0</code>
		+ <code>linuxsnet AUTOEXEC.NET</code>
		+ <code>attach sdev0 kiss 1 3 38400 /tmp /soundmodem</code>

	+ po 1 baud 1200
	+ po 2 baud 1200
	+ po 3 baud 9600
- ==afskmodem==	
- Das AFSKMODEM ist ein [[Packet Radio via Soundkarte unter Linux digitales Soundmodem]], welches die Pakete in eine (A)FSK Modulation wandelt und der Soundkarte zuführt. Der Name soll jedoch nicht verwirren, es sind auch je nach Soundkarte Geschwindigkeiten > 28kBaud FSK möglich.	+ 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 \

	+ <code>-M 2 -c 0 -b 9600 -a -a -a 200 -U 127.0.0.1:9202/9212 -m 2</code>
- [[TCE Software << zurück zu Einstellungen & Bedienung]]	+ 2. XNET empfängt von udpbox und sendet direkt zum Modem:

	+ <code>linuxsnet AUTOEXEC.NET</code>
	+ <code>attach ip0 axudp 1 1 I9300 d9210 127.0.0.1</code>
	+ <code>attach ip1 axudp 2 1 I9301 d9211 127.0.0.1</code>
	+ <code>attach ip2 axudp 3 1 I9302 d9212 127.0.0.1</code>
	+ <code>po 1 baud 1200</code>
	+ <code>po 2 baud 1200</code>
	+ <code>po 3 baud 9600</code>
	+
	+ 3. udpbox empfängt vom Modem (Port 920x) in AXUDP (9401)

	+ <code>sendet alle UI zu aprsd auf 192.168.1.1:9000
</code>
	+ <code>sendet nur "APRS Messages" (-f p58) zu Modem Funkport 1
</code>
	+ <code>./udpbox -R 0.0.0.0:9200 -m 192.168.1.1:9000 -r 127.0.0.1:93:9300\</code>
	+ <code>-R 0.0.0.0:9201 -m 192.168.1.1:9000 -r 127.0.0.1:93:9301\</code>
	+ <code>-R 0.0.0.0:9202 -m 192.168.1.1:9000 -r 127.0.0.1:93:9302\</code>
	+ <code>-M 0.0.0.0:9401 -r 127.0.0.1:9211 -f p58 -r 127.0.0.1:9210 -v</code>
	+
	+ 4. udpbox als aprs-digi (Beispiel)

	+ <code>-R empfang axudp auf Port 9000</code>

- +
- + **-u bestätige und speichere User Messages an OE0AAA-12 in File /tmp/msg12.txt**
- +
- + **-f filtere für die überlastete 144.800 je nach Geschmack Data-Typen weg**
- + **wie z.b. garnicht aprs-frames, thirdparty-messages, Status-Meldungen**
- + **weg. Die Zahlen sind der Dezimalwert des 1. Bytes der Nutzdaten**
- + **(siehe aprs Protokollbeschreibung APRS101.pdf)**
- +
- + **-x filtere Frames mit TCPIP oder NOCALL weg**
- +
- + **-p sende nur (soweit vom Absender richtig adressierte) Frames die nach direkt gehört aussehen (first hop digi), sende aber den Rest vom Pfad nach Protokoll modifiziert mit für weitere Hops**
- + **füge das digicall OE0AAA-11 zur korrekten Pfad aufzeichnung ein**
- + **anderfalls bliebe der digi unsichtbar.**
- + **Es werden alle Adressierungsarten akzeptiert einschliesslich der effizientesten mit Destination-SSID. Damit kann bei (insbesondere von Mobilstationen gesendeten) Frames 14 byte "WIDE1-1,WIDE2-2..." oder etwa 30% eingespart werden.**
- +

- + **-t filtere gleichbleibende Texte (sprich nervige Baken) 27min weg,**
- + **lasse aber (retryende) User Messages nach 28s durch**
- +
- + **-b sende Bake aus dem File aprsbeacon.txt alle 300s, aber ebenfalls**
- + **gefiltert, also wenn der Text nicht zb. durch neue Wetterdaten ersetzt**
- + **wurde, alle 30min**
- +
- + **-k Filtere alles (ausser User-msg) ausserhalb Umkreis koordinate(grad) /radius(km)**
- +
- + **-c sende zu Monitorzwecken (nc -l -u -p 2000) den sendefertigen Inhalt**
- + **mit Linefeed an Rechner 192.168.1.24: 2000**
- +
- + **-r sende das gleiche(-e) zum Modem als axudp 127.0.0.1:9100**
- +
- + **-v sagt was es tut und warum auf dem standard output**
- +
- + **./udpbox -v -u OE0AAA-12:/tmp /msg12.txt -R 0.0.0.0:9000**
- + **-f d59,60,125,65-83,85-90,97-122 -x TCPIP,NOCALL -d OE0AAA-11**
- + **-p 5,6,7,8,9 -t 1680,28 -b 300: aprsbeacon.txt -k 48.2/-13.1/40**
- + **-c 192.168.1.24:2000 -e -r 127.0.0.1: 9100
**

- +
- +

Man kann noch eine Kopie der ungefilterten rx Daten im monitorformat
- +

an zb. aprsd für Igate senden (-m 127.0.0.1:9304)
- +

Weitere parameter siehe -h
- +
- +

Bakentext File Beispiel:
- +

(Hier sollte man vorsichtig sein um keine Alarmsymbole zu erwischen
- +

aber Call und Koordinaten ausbessern)
- +

Es wird nur die 1. Zeile des Files gesendet, das File kann aber jederzeit
- +

zb. von einem Messwert Programm modifiziert werden.
- +
- +

OE0AAA-11>TEST,TRACE2-2:!9000.00 N/18000.00E#PHG3750Test Digi
- +
- +

dazu auf 266MHz Geode CPU mit billiq-USB-Sound"karte" optimierter Modemstart
- +

(-e 50 leichtes rx Hochpassfilter wegen dumpfem nf-Ausgang beim Rx)
- +
- +

aoss /home/tc/afskmodem -f 24000 -e 8 -t /dev/ttyS0
- +

-l 128 -b 1 -M 0 -U 127.0.0.1:9000 /9100 -m 0 -e 50 &
- +
- +

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

- +
- +
- + **I-Gate mit udpgate**
- +
- + **Rx-Igate kompatibel nach:**
- + **http://wiki.ham.fi/APRS_iGate_properties#APRS-IS_connection_2**
- +
- + **./udpgate -R 0.0.0.0:9000 -t 14580 -s OE0AAA-10 -n 10:netbeacon.txt**
- + **-q www.db0anf.de 14580 -p /etc/pass.txt -f "m/30" -l 6:udp.log -w 14501**
- +
- + **-t TCP Port für Connects mit Aprs-Gaffern wie xastir oder weitere igates**
- +
- + **-s Call des Servers**
- +
- + **-n alle 10 Min Netzbake mit Server Position**
- + **File Inhalt (bitte richtige Koordinaten an den gleichen Spalten eingeben!)**
- + **grad minuten.minutenkommas, "/" und "&" ist das Aprs-Symbol**
- +
- + **!8959.00N/01300.20E&Igate Nordpol**
- +
- + **-q Connect zum APRS-IS Netz oder anderem udpgate (Befehl wiederholen dann werden die Server bei Linkausfall der Reihe nach mit 30s Pause versucht)**
- +

- + **-p** **passwort oder Filename mit Passwort damits in der Kommandozeile unsichtbar ist**
- +
- + **-f** **Filterparameter werden zum Server gesendet**
- +
- + **-l** **Loggt Connects, Frames (gute, gefilterte, duplikate) je nach loglevel**
- + **(das File wird nach jeder Zeile geschlossen und kann gekürzt /gelöscht werden)**
- +
- + **-w** **www Port**
- +
- +
- + **[[Bild:udpdate-html.gif|center|www-Statistik-Beispiel]]**
- +
- +
- + **
**
- + **Dieses Projekt ist Open Source - Haftung, Verantwortung und Spaß übernimmt jeder selbst.**

Aktuelle Version vom 22. November 2019, 18:48 Uhr

Inhaltsverzeichnis

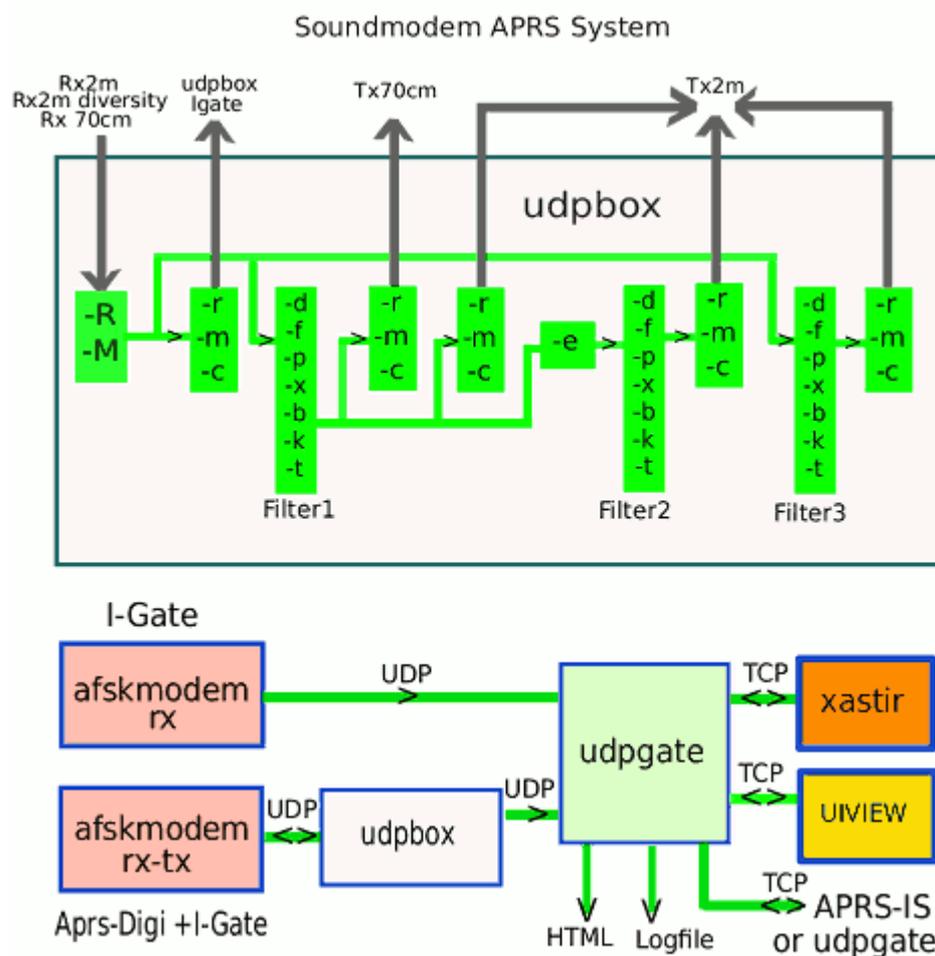
1 Das Projekt	22
2 Der Source Code	22
3 Der kompilierte Treiber	22
4 Starten bzw. Aufrufen des Treibers	23

Das Projekt

Dieser (USB) Soundkartentreiber von OE5DXL soll es ermöglichen mit 2 Kanälen (L und R der Soundkarte) 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 (bspw. 1k2 2k4 4k8 9k6 auf einer QRG). In Stereo kann so theoretisch ein multibaud FSK AFSK KISS, als auch AXUDP AX.25 Modem betrieben werden.



Der Source Code

[1]<https://github.com/oe5hpm/dxIAPRS>

[2]<http://gitlab.oe5xbl.ampr.org/oe5hpm/dxIAPRS/>

Der kompilierte Treiber

Soundmodem-bin - Der fertig kompilierte Soundmodem Treiber

udpbox-bin - UDP Filter und RAW-Monitor Konverter und (neu) mit aprs-digi, Bake, User-Message-Receiver

```
udpgate-bin - I-Gate, APRS-IS, APRS-Server mit Rangefilter, HTML-Statistik, Log
```

```
udphub-bin - axudp Hub zum HAMNET-PR-Login ohne IP Beschränkung
```

```
udpflex-bin - Interface com-port (/dev/ttySxx) mit KISS oder RMNC bidirektional auf axudp
```

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 als:

```
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 l9200 d9210 127.0.0.1
attach ip1 axudp 2 1 l9201 d9211 127.0.0.1
attach ip2 axudp 3 1 l9202 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 l9300 d9210 127.0.0.1
attach ip1 axudp 2 1 l9301 d9211 127.0.0.1
attach ip2 axudp 3 1 l9302 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 (9401)
 sendet alle UI zu aprsd auf 192.168.1.1:9000
 sendet nur "APRS Messages" (-f p58) zu Modem Funkport 1

```
./udpbox -R 0.0.0.0:9200 -m 192.168.1.1:9000 -r 127.0.0.1:93:9300\  

-R 0.0.0.0:9201 -m 192.168.1.1:9000 -r 127.0.0.1:93:9301\  

-R 0.0.0.0:9202 -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. udpbox als aprs-digi (Beispiel)
 -R empfangen axudp auf Port 9000

-u bestätige und speichere User Messages an OE0AAA-12 in File /tmp/msg12.txt

-f filtere für die überlastete 144.800 je nach Geschmack Data-Typen weg wie z.B. garnicht aprs-frames, thirdparty-messages, Status-Meldungen weg. Die Zahlen sind der Dezimalwert des 1. Bytes der Nutzdaten (siehe aprs Protokollbeschreibung APRS101.pdf)

-x filtere Frames mit TCPIP oder NOCALL weg

-p sende nur (soweit vom Absender richtig adressierte) Frames die nach direkt gehört aussehen (first hop digi), sende aber den Rest vom Pfad nach Protokoll modifiziert mit für weitere Hops füge das digicall OE0AAA-11 zur korrekten Pfad aufzeichnung ein anderfalls bliebe der digi unsichtbar. Es werden alle Adressierungsarten akzeptiert einschliesslich der effizientesten mit Destination-SSID. Damit kann bei (insbesondere von Mobilstationen gesendeten) Frames 14 byte "WIDE1-1,WIDE2-2..." oder etwa 30% eingespart werden.

-t filtere gleichbleibende Texte (sprich nervige Baken) 27min weg, lasse aber (retryende) User Messages nach 28s durch

-b sende Bake aus dem File aprsbeacon.txt alle 300s, aber ebenfalls gefiltert, also wenn der Text nicht zb. durch neue Wetterdaten ersetzt wurde, alle 30min

-k Filtere alles (ausser User-msg) ausserhalb Umkreis koordinate(grad)/radius(km)

-c sende zu Monitorzwecken (nc -l -u -p 2000) den sendefertigen Inhalt mit Linefeed an Rechner 192.168.1.24:2000

-r sende das gleiche(-e) zum Modem als axudp 127.0.0.1:9100

-v sagt was es tut und warum auf dem standard output

```
./udpbox -v -u OE0AAA-12:/tmp/msg12.txt -R 0.0.0.0:9000\  

-f d59,60,125,65-83,85-90,97-122 -x TCPIP,NOCALL -d OE0AAA-11\  

-p 5,6,7,8,9 -t 1680,28 -b 300:aprsbeacon.txt -k 48.2/-13.1/40\  

-c 192.168.1.24:2000 -e -r 127.0.0.1:9100
```

Man kann noch eine Kopie der ungefilterten rx Daten im monitor-format an zb. aprsd für lgate senden (-m 127.0.0.1:9304) Weitere parameter siehe -h

Bakentext File Beispiel: (Hier sollte man vorsichtig sein um keine Alarmsymbole zu erwischen aber Call und Koordinaten ausbessern) Es wird nur die 1. Zeile des Files gesendet, das File kann aber jederzeit zb. von einem Messwert Programm modifiziert werden.

```
0E0AAA-11>TEST,TRACE2-2:!9000.00N/18000.00E#PHG3750Test Digi
```

dazu auf 266MHz Geode CPU mit billig-USB-Sound"karte" optimierter Modemstart (-e 50 leichtes rx Hochpassfilter wegen dumpfem nf-Ausgang beim Rx)

```
aoss /home/tc/afskmodem -f 24000 -e 8 -t /dev/ttyS0\  
-l 128 -b 1 -M 0 -U 127.0.0.1:9000/9100 -m 0 -e 50 &
```

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

I-Gate mit udpgate

Rx-Igate kompatibel nach: http://wiki.ham.fi/APRS_iGate_properties#APRS-IS_connection_2

```
./udpgate -R 0.0.0.0:9000 -t 14580 -s 0E0AAA-10 -n 10:netbeacon.txt\  
-g www.db0anf.de 14580 -p /etc/pass.txt -f "m/30" -l 6:udp.log -w 14501
```

-t TCP Port für Connects mit Aprs-Gaffern wie xastir oder weitere igates

-s Call des Servers

-n alle 10 Min Netzbake mit Server Position File Inhalt (bitte richtige Koordinaten an den gleichen Spalten eingeben!) grad minuten.minutenkommas, "/" und "&" ist das Aprs-Symbol

```
!8959.00N/01300.20E&Igate Nordpol
```

-g Connect zum APRS-IS Netz oder anderem udpgate (Befehl wiederholen dann werden die Server bei Linkausfall der Reihe nach mit 30s Pause versucht)

-p password oder Filename mit Passwort damits in der Kommandozeile unsichtbar ist

-f Filterparameter werden zum Server gesendet

-l Loggt Connects, Frames (gute, gefilterte, duplikate) je nach loglevel (das File wird nach jeder Zeile geschlossen und kann gekürzt/gelöscht werden)

-w www Port

[www-Statistik-Beispiel](#)

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

TCE Komponenten: Unterschied zwischen den Versionen

[Versionsgeschichte interaktiv durchsuchen](#)
[Visuell Wikitext](#)

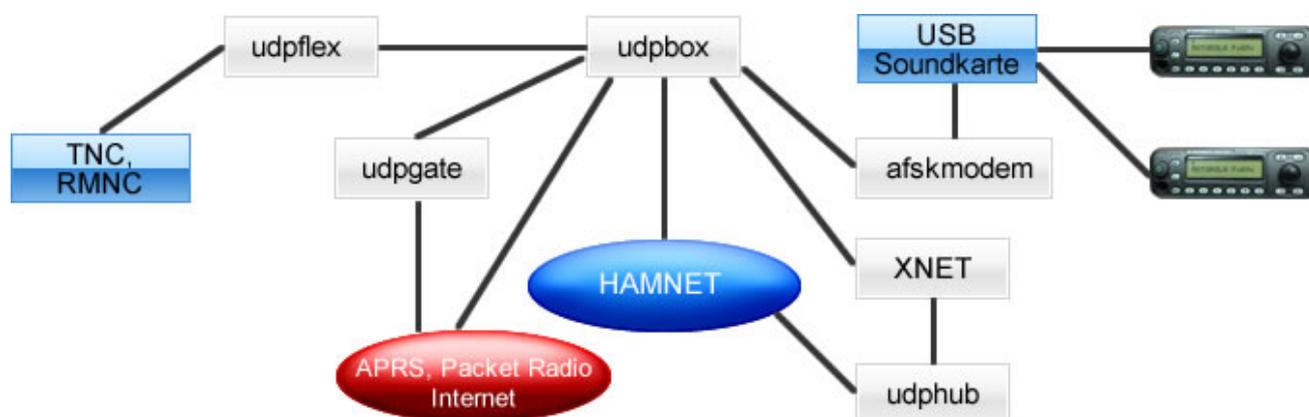
Version vom 8. Dezember 2014, 19:27 Uhr (Quelltext anzeigen)
 OE2WAO ([Diskussion](#) | [Beiträge](#))
[← Zum vorherigen Versionsunterschied](#)

Aktuelle Version vom 16. August 2015, 12:04 Uhr (Quelltext anzeigen)
 OE2WAO ([Diskussion](#) | [Beiträge](#))
[K \(→udpgate\)](#)

Zeile 17:	==udpgate==	Zeile 17:	==udpgate==
-	[[Datei:XZR-conn.PNG 200px thumb left Connection Tab sample]]	+	[[Datei:XZR-conn.PNG 200px thumb Connection Tab sample]]
	Das UDPGATE ist ein APRS Server, welcher die Netzwerkebene des APRS Datentransports übernimmt. Also bspw. die Serverfunktionalität für Benutzer bereitstellen, sowie eine Verbindung zum APRS IS oder nächsten APRS Server (UDPGATE) herstellen. 		Das UDPGATE ist ein APRS Server, welcher die Netzwerkebene des APRS Datentransports übernimmt. Also bspw. die Serverfunktionalität für Benutzer bereitstellen, sowie eine Verbindung zum APRS IS oder nächsten APRS Server (UDPGATE) herstellen.
	Dabei agiert er bei der Verbindung nach der Priorität der Einträge in der Serverliste. Ist der erste Server nicht erreichbar, wird der nächste Server in der Liste versucht zu erreichen. In regelmässigen Abständen wird jedoch erneut versucht, die in der Liste zuvorgelegenen Server nach dem Prioritätsprinzip zu erreichen, und verlustfrei wieder rückzuverbinden. 		Dabei agiert er bei der Verbindung nach der Priorität der Einträge in der Serverliste. Ist der erste Server nicht erreichbar, wird der nächste Server in der Liste versucht zu erreichen. In regelmässigen Abständen wird jedoch erneut versucht, die in der Liste zuvorgelegenen Server nach dem Prioritätsprinzip zu erreichen, und verlustfrei wieder rückzuverbinden.
	Das Modul verfügt über ein eigenes Webinterface welches default unter "serverIP:14501" erreichbar ist. 		Das Modul verfügt über ein eigenes Webinterface welches default unter "serverIP:14501" erreichbar ist.
-			
-			
-			
	==udprfnet==		==udprfnet==

Aktuelle Version vom 16. August 2015, 12:04 Uhr

Im AFU Tinycore Image sind unter anderem amateurfunkspezifische Programme enthalten.



Inhaltsverzeichnis

1	udpbox	29
2	udphub	29
3	udpgate	29
4	udprfnet	29
5	afskmodem	30

afskmodem

Das AFSKMODEM ist ein [digitales Soundmodem](#), welches die Pakete in eine (A)FSK Modulation wandelt und der Soundkarte zuführt. Der Name soll jedoch nicht verwirren, es sind auch je nach Soundkarte Geschwindigkeiten > 28kBaud FSK möglich.

[<< zurück zu Einstellungen & Bedienung](#)

TCE Komponenten und TCE Software: Unterschied zwischen den Seiten

VisuellWikitext

Version vom 8. Dezember 2014, 19:27 Uhr (Quelltext anzeigen)
 OE2WAO (Diskussion | Beiträge)

Aktuelle Version vom 27. Mai 2022, 21:14 Uhr (Quelltext anzeigen)
 OE2WAO (Diskussion | Beiträge)
 K
 Markierung: Visuelle Bearbeitung

Zeile 1:

Im AFU Tynycore Image sind unter anderem amateurfunkspziefische Programme enthalten.

Zeile 1:

- + **==Komponenten / Module==**
- +
- + Im AFU Tynycore Image sind unter anderem amateurfunkspziefische Programme enthalten.

- + **[[TCE Komponenten | >> Nähere Details zu den einzelnen Komponenten]]**
- +
- + **[[Datei:Udpboxs.jpg]]**
- +
- + **==Einstellungen==**
- +
- + **Eine kleine Dokumentation für die notwendigen Betriebseinstellungen befindet sich im Verzeichnis**
- + **/home/tc/readme**
- +
- + **Voreingestellt im Image sind folgende Werte:
**
- + **Fixe IP: 192.168.1.50/24 (zu ändern entweder über die X11 Oberfläche oder in /opt/eth0)**
- +

- **[[Datei:Udpboxs.jpg]]**


```

+ # Haupt IP Adresse
+ ifconfig eth0 44.143.40.90 netmask
  255.255.255.240
+ # Zusätzliche/optionale IP Adresse
  für lokale Konfiguration
+ ifconfig eth0:0 192.168.1.50
+ # Gateway
+ route add default gw 44.143.40.94
+ # Nameserver
+ echo nameserver 44.143.40.30 > /etc
  /resolv.conf
+ echo nameserver 44.143.168.30 >>
  /etc/resolv.conf

```

Der UDPHUB ist ein Hilfsprogramm für XNET, welches die IP Beschränkung umgeht, indem es sich selbst zwischen Benutzer und XNET stellt, und die AXUDP Pakete entsprechend verteilt. Dabei bleibt der Ursprungspfad (IP) des Benutzer eine Woche (einstellbar) gespeichert, und der Benutzer kann bei lokal gestartetem Programm auch ohne aktiven Connect in dieser Zeitspanne von anderen Benutzern kontaktiert werden, genauso als ob man per HF QRV wäre.

===Starten und Stoppen von Modulen===

```

+ Die einzelnen Softwaremodule
  können per Befehl gestartet oder
  gestoppt werden. Syntax: ./modul
  [start|stop|status]<br>
+ Bspw. XNET (Neu)Start:
+ ./snet start
+ APRSdigi (Neu)Start:
+ ./igate start
+ Ubox (Neu)Start:

```

		+ <code>./ubox start</code>
		+ <code>Hub (Neu)Start:</code>
		+ <code>./hub start</code>
		+ <code>L2XNET(Neu)Start:</code>
		+ <code>./l2xnet start</code>
		+ <code>Modem (Neu)Start:</code>
		+ <code>./modem start</code>
-	<code>==udpgate==</code>	+ <code>===Uhrzeit synchronisieren (HAMNET Time-Server)===</code>
		+ <code>sudo ntpclient -c 1 -s -h 44.143.243.254</code>
		+ <code>sudo hwclock --systohc</code>
-	<code>[[Datei:XZR-conn.PNG 200px thumb left Connection Tab sample]]</code>	+ <code>===Zum Abschluss speichern===</code>
-	<p>Das UDPGATE ist ein APRS Server, welcher die Netzwerkebene des APRS Datentransports übernimmt. Also bspw. die Serverfunktionalität für Benutzer bereitstellen, sowie eine Verbindung zum APRS IS oder nächsten APRS Server (UDPGATE) herstellen.
</p>	+ <code>""WICHTIG !!""
</code>
-	<p>Dabei agiert er bei der Verbindung nach der Priorität der Einträge in der Serverliste. Ist der erste Server nicht erreichbar, wird der nächste Server in der Liste versucht zu erreichen. In regelmässigen Abständen wird jedoch erneut versucht, die in der Liste zugelegenen Server nach dem Prioritätsprinzip zu erreichen, und verlustfrei wieder rückzuverbinden.
</p>	+ <p>Einstellung im System finden immer im RAM statt. Um diese dauerhaft auf den Festplatten- bzw. CF-Speicher zu schreiben, muss dies eigens veranlasst werden. Entweder beim Herunterfahren in der grafischen Oberfläche (X11) selbst mit der BACKUP Option, oder ferngesteuert (SSH) mittels dem Befehl</p>
		<code>(sudo su)*</code>

- Das Modul verfügt über ein eigenes Webinterface welches default unter "serverIP:14501" erreichbar ist.

+

+ filetool.sh -b

+ *(ohne "sudo su" nur solange der Befehl nicht zuvor einmal als root ausgeführt wurde)

+ ==Vorgefertigte Varianten==

+

Die Vielfalt in der Zusammenstellung der einzelnen Komponenten erlaubt eine größere Zahl an unterschiedlichen Konfigurationen. Zum leichteren und schnelleren Einsatz am Digistandort bietet [http://www.oe2wao.info OE2WAO] unter Bekanntgabe des geplanten Digi Rufzeichens mehrere Standard Varianten vorgefertigt zur Auswahl.

- ==udprfnet==

+

+ ""Variante 1""

+

+ * XNET Multibaud Packet Radio Digi

+

+ - Multibaud (1k2 2k4 4k8 9k6) User Zugang auf einer Frequenz

+

+ - variabler HAMNET Zugangsport für sämtliche IP Adressen (User Access)

+

+ - weitere AXUDP HAMNET Links können konfiguriert werden

+

+ * APRS Server und Digi mit IGATE

+

+ - 1k2 RX und TX inkl. Message Gateway

+

+ - 1k2 und 9k6 RX auf Packet Radio User Zugang

+

+ - optional 300bd RX (und TX) für Kurzwellen APRS

	<ul style="list-style-type: none"> + - 1k2 TX auf Packet Radio User Zugang von direkt gehörten APRS Stationen
	<ul style="list-style-type: none"> + - 9k6 TX auf Packet Radio User Zugang aller auf HF 1k2 gehörten APRS Pakete
<p>- Das UDPRFNET Modul ist eine experimentelle Software für eine intelligente APRS Paketverteilung unter Digipeatern. Ziel ist das gesamte Netz als einen großen RX darzustellen und auch weiter entfernt empfangene APRS Pakete vom dort gebietsmässig nahegelegenen Digi per Radiusdefinition auf 144.800MHz wieder aussenden zu lassen. Dabei bilden mehrere Serververbindungen untereinander das Prinzip der Redundanz.</p>	<p>+ ""Variante 2""</p>
	<ul style="list-style-type: none"> + * XNET Multibaud Packet Radio Digi
<p>- ==afskmodem==</p>	<ul style="list-style-type: none"> + - Multibaud (1k2 2k4 4k8 9k6) User Zugang auf einer Frequenz
	<ul style="list-style-type: none"> + - variabler HAMNET Zugangsport für sämtliche IP Adressen (User Access)
	<ul style="list-style-type: none"> + - weitere AXUDP HAMNET Links können konfiguriert werden
	<p>+ ""Variante 3""</p>
<p>- Das AFSKMODEM ist ein [[Packet Radio via Soundkarte unter Linux digitales Soundmodem]], welches die Pakete in eine (A)FSK</p>	<ul style="list-style-type: none"> + * APRS Server und Digi mit IGATE
	<ul style="list-style-type: none"> + - 1k2 RX und TX inkl. Message Gateway
	<ul style="list-style-type: none"> + - optional 300bd RX (und TX) für Kurzwellen APRS

- **Modulation wandelt und der Soundkarte zuführt. Der Name soll jedoch nicht verwirren, es sind auch je nach Soundkarte Geschwindigkeiten > 28kBaud FSK möglich.**

- [\[\[TCE Software | << zurück zu Einstellungen & Bedienung\]\]](#)

+

- [\[\[TCE Tinycore Linux Projekt | << Zurück k zur TCE Projekt Übersicht\]\]](#)

Aktuelle Version vom 27. Mai 2022, 21:14 Uhr

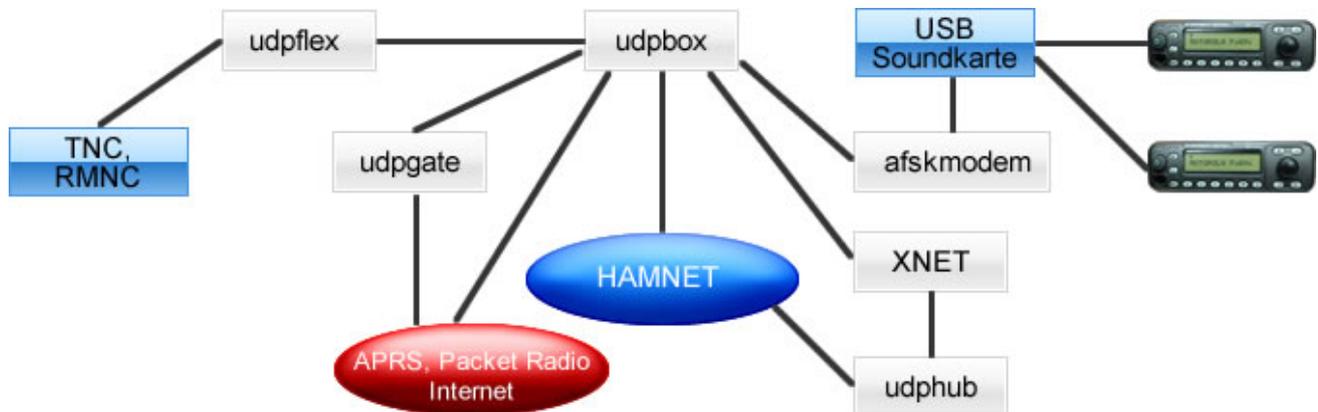
Inhaltsverzeichnis

1 Komponenten / Module	38
2 Einstellungen	38
2.1 Netzwerk einstellen	38
2.2 Starten und Stoppen von Modulen	39
2.3 Uhrzeit synchronisieren (HAMNET Time-Server)	39
2.4 Zum Abschluss speichern	40
3 Vorgefertigte Varianten	40

Komponenten / Module

Im AFU Tincore Image sind unter anderem amateurfunkspezifische Programme enthalten.

>> [Nähere Details zu den einzelnen Komponenten](#)



Einstellungen

Eine kleine Dokumentation für die notwendigen Betriebseinstellungen befindet sich im Verzeichnis

```
/home/tc/readme
```

Voreingestellt im Image sind folgende Werte:

```
Fixe IP: 192.168.1.50/24 (zu ändern entweder über die X11 Oberfläche oder in /opt/eth0)
```

Zu startenden Programme und Optionen (ähnlich autoexec.bat in MS Betriebssystemen) befinden sich in nachfolgender Datei, und müssen zur korrekten Funktion editiert werden:

```
/opt/bootlocal.sh (im Grundzustand sind sämtliche Programme mit '#' auskommentiert)
```

Default Zugang für SSH (unter MS Windows am Besten mit [putty](#))

```
User: tc
Pass: 12345678
```

Netzwerk einstellen

Netzwerkkonfiguration anhand des Beispiels von OE2XZR (IP Adressen müssen auf eigene Bedürfnisse geändert werden).

```
Datei: /opt/eth0.sh
```

```
#!/bin/sh
# DHCP deaktivieren
pkill udhcpd
# Haupt IP Adresse
ifconfig eth0 44.143.40.90 netmask 255.255.255.240
# Zusätzliche/optionale IP Adresse für lokale Konfiguration
ifconfig eth0:0 192.168.1.50
# Gateway
route add default gw 44.143.40.94
# Nameserver
echo nameserver 44.143.40.30 > /etc/resolv.conf
echo nameserver 44.143.168.30 >> /etc/resolv.conf
```

Starten und Stoppen von Modulen

Die einzelnen Softwaremodule können per Befehl gestartet oder gestoppt werden. Syntax: `./modul [start|stop|status]`

Bspw. XNET (Neu)Start:

```
./snet start
```

APRSdigi (Neu)Start:

```
./igate start
```

Ubox (Neu)Start:

```
./ubox start
```

Hub (Neu)Start:

```
./hub start
```

L2XNET(Neu)Start:

```
./l2xnet start
```

Modem (Neu)Start:

```
./modem start
```

Uhrzeit synchronisieren (HAMNET Time-Server)

```
sudo ntpclient -c 1 -s -h 44.143.243.254
sudo hwclock --systohc
```

Zum Abschluss speichern

WICHTIG !!

Einstellung im System finden immer im RAM statt. Um diese dauerhaft auf den Festplatten- bzw. CF-Speicher zu schreiben, muss dies eigens veranlasst werden. Entweder beim Herunterfahren in der grafischen Oberfläche (X11) selbst mit der BACKUP Option, oder ferngesteuert (SSH) mittels dem Befehl

```
(sudo su)*  
filetool.sh -b
```

- *(ohne "sudo su" nur solange der Befehl nicht zuvor einmal als root ausgeführt wurde)

Vorgefertigte Varianten

Die Vielfalt in der Zusammenstellung der einzelnen Komponenten erlaubt eine größere Zahl an unterschiedlichen Konfigurationen. Zum leichteren und schnelleren Einsatz am Digistandort bietet [OE2WAO](#) unter Bekanntgabe des geplanten Digi Rufzeichens mehrere Standard Varianten vorgefertigt zur Auswahl.

Variante 1

- * XNET Multibaud Packet Radio Digi
 - Multibaud (1k2 2k4 4k8 9k6) User Zugang auf einer Frequenz
 - variabler HAMNET Zugangsport für sämtliche IP Adressen (User Access)
 - weitere AXUDP HAMNET Links können konfiguriert werden
- * APRS Server und Digi mit IGATE
 - 1k2 RX und TX inkl. Message Gateway
 - 1k2 und 9k6 RX auf Packet Radio User Zugang
 - optional 300bd RX (und TX) für Kurzwellen APRS
 - 1k2 TX auf Packet Radio User Zugang von direkt gehörten APRS Stationen
 - 9k6 TX auf Packet Radio User Zugang aller auf HF 1k2 gehörten APRS Packete

Variante 2

- * XNET Multibaud Packet Radio Digi
 - Multibaud (1k2 2k4 4k8 9k6) User Zugang auf einer Frequenz
 - variabler HAMNET Zugangsport für sämtliche IP Adressen (User Access)
 - weitere AXUDP HAMNET Links können konfiguriert werden

Variante 3

- * APRS Server und Digi mit IGATE
 - 1k2 RX und TX inkl. Message Gateway
 - optional 300bd RX (und TX) für Kurzwellen APRS

[<< Zurück zur TCE Projekt Übersicht](#)