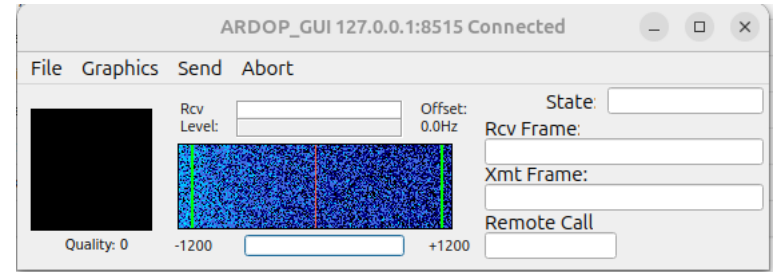


# Winlink mit Pat und ARDOP

Installation auf einem Linux-PC ohne Wine



DF3FY  
24.10.2024  
Version 2

# Hintergrund

- Schwacher Hardware kann mittels Linux frische Kraft gegeben werden. Darüber hinaus gibt es User, die besonderen Wert auf Open-Source-Software legen.
- Verzichtet man auf etwas Komfort und Übertragungsgeschwindigkeit, so ist das Linux-ARDOP-Modem ARDOPC mit der Linux-Mailing-Software Pat eine schlanke Kombination, um Winlink-Mails zu versenden und zu empfangen. Neben VARA- sind auch genügend ARDOP-Gateways vorhanden.
- Pat ist hier zu bekommen:  
<https://getpat.io/>
- Das Modem sowie das GUI dazu gibt es im Downloadbereich folgender Seiten:  
<https://www.cantab.net/users/john.wiseman/Documents/ARDOPC.html> und  
<https://github.com/pflarue/ardop/releases/tag/1.0.4.1.3> (neuere Version, u.a. reduced PTT delay)
- Pat kann statt ARDOP auch VARA (dieses Modem dann mit Wine) nutzen. RMS-Express kann in jedem Fall durch Pat wirksam ersetzt werden.

# Installation auf Debian oder UBUNTU

- Die meisten Linux-Distributionen verwenden die .deb-Paket-Verwaltung.
- Distributionen mit .deb-Paket-Namensendungen basieren auf Debian, z.B. UBUNTU, MINT, MX
- Wer Debian oder Ubuntu nutzt, findet im Internet den wohl breitesten Support.
- Es empfiehlt sich, grundsätzlich Ausgaben mit LTS (long term support) zu nutzen, aktuell z.B. Ubuntu 24.04 oder Debian 12.
- Die Installation von Pat und ARDOP-Modem nebst GUI ist bei Debian-basierten Distributionen gleich.

# Verwendung des Terminals

- Öffnen Sie das Terminal und erlangen Sie Root-Rechte (Befehle mit Administrator- bzw. Root-Rechten sind in **rot**, Befehle mit normalen User-Rechten in **grün** angegeben.)
- **su -**  
bei Debian oder
- **sudo su -**  
bei Ubuntu, Mint oder MX. (Bei diesen Versionen ist sudo immer direkt vorhanden.)
- Schritt 1 sollte ein Update des vorhandenen Linux-Systems sein.  
**apt update**  
**apt upgrade**
- Temporäre Root-Rechte erlangt man – wenn sudo implementiert ist – mit **sudo mein\_gewünschter\_Befehl** (Das Terminal braucht somit vorher nicht auf Root gebracht zu werden.)

# 32-Bit-Architektur

- Linux-Systeme werden heute fast immer mit 64-Bit-Architektur angeboten. Die 32-Bit-Fähigkeit muss für 32-Bit-Programme nachträglich zugefügt werden.
- Die Linux-ARDOP\_GUI gibt es (will man nicht selbst kompilieren) nur als 32-Bit-Version.
- Das ARDOP Modem ist als 64-Bit (ardopc64) und 32-Bit-Version (ardopc) vorhanden.
- Daher mittels des folgenden Befehls für die 32-Bit-Architektur sorgen:  
`dpkg --add-architecture i386 && apt update`

# Installation der nötigen Pakete

- Die runter geladenen Pakete finden sich im Ordner Downloads.

`cd Downloads`

- `apt install ./pat_0.16.0_linux_amd64.deb` (Pat-Datei-Name an die tatsächliche Version anpassen)  
`cp ardopc /usr/local/bin/` (oder die Version ardopcf\_amd64\_Linux\_64 umbenannt nach Download auf ardopcf)  
`cp ARDOP_GUI /usr/local/bin`  
`chmod +x /usr/local/bin/ARDOP_GUI`  
`chmod +x /usr/local/bin/ardopc`  
`apt install libqt5widgets5:i386`  
`apt-install hamlib-utils` (für das Sub-Programm rigctld zur TRX-Steuerung)
- Es empfehlen sich aber gleich auch noch andere Afu-Digimode-Programme.

`apt install fldigi flrig`

# Konfiguration von Pat

Nach Eingabe von  
`pat configure`

ist die sich im  
Terminal öffnende  
Datei (rechts) auf  
die eigenen  
Verhältnisse  
anzupassen. Mit  
`strg o` speichern  
und mit `strg x` die  
Datei verlassen.

(Die Datei wird automatisch im  
Persönlichen Ordner  
unter `.config/pat/config.json`  
abgelegt.)

```
{
  "mycall": "DL0ABC",
  "secure_login_password": "mein Passwort",
  "auxiliary_addresses": [],
  "locator": "JO300Q",
  "service_codes": [
    "PUBLIC"
  ],
  "http_addr": "localhost:8080",
  "motd": [
    "Open source Winlink client - getpat.io"
  ],
  "connect_aliases": {
    "telnet":
      "telnet://{{mycall}}:CMSTelnet@cms.winlink.org:8772/wl2k"
  },
  "listen": [],
  "hamlib_rigs": {
    "K2": {"address": "localhost:4532", "network": "tcp"}
  },
  "ax25": {
    "port": "wl2k",
    "beacon": {
      "every": 3600,
      "message": "Winlink P2P",
      "destination": "IDENT"
    }
  },
  "rig": "K2"
}
```

```
"serial-tnc": {
  "path": "/dev/ttyUSB0",
  "serial_baud": 4800,
  "hbaud": 1200,
  "type": "Kenwood"
},
"ardop": {
  "addr": "localhost:8515",
  "arq_bandwidth": {
    "Forced": false,
    "Max": 500
  },
  "rig": "K2",
  "ptt_ctrl": true,
  "beacon_interval": 0,
  "cwid_enabled": true
},
"pactor": {
  "path": "/dev/ttyUSB0",
  "baudrate": 57600,
  "rig": "FT-897",
  "custom_init_script": ""
},
"telnet": {
  "listen_addr": ":8774",
  "password": ""
},
}
```

```
"varahf": {
  "host": "localhost",
  "cmdPort": 8300,
  "dataPort": 8301,
  "bandwidth": 2300,
  "rig": "K2",
  "ptt_ctrl": true
},
"varafm": {
  "host": "localhost",
  "cmdPort": 8300,
  "dataPort": 8301,
  "bandwidth": 9600,
  "rig": "FT-897",
  "ptt_ctrl": true
},
"gpsd": {
  "enable_http": false,
  "allow_forms": false,
  "use_server_time": false,
  "addr": "localhost:2947"
},
"schedule": {},
"version_reporting_disabled":
false
}
```

# Serielle Schnittstelle konfigurieren

- Dieser Vorgang stellt manchmal einen Fallstrick dar. Zunächst stellt man sicher, dass normale User generell die serielle Schnittstelle zum Lesen und Schreiben nutzen dürfen.  
`chmod 666 /dev/ttyS0`
- Man steckt nun den Schnittstellen-Dongle in einen USB-Port und fragt ihn im Terminal ab mit  
`dmesg | grep tty`  
Ausgabe z.B.: `usb 1-1: pl2303 converter now attached to ttyUSB0.`  
`chmod a+rw /dev/ttyUSB0`  
`usermod -a -G dialout $USER` (statt \$ USER den eigenen User-Namen nehmen, klein geschrieben)
- Moderne Linux-Kernel kennen fast alle USB-Chips. Es kann aber in Ausnahmefällen sein, dass zum Kernel das passende Modul für einen speziellen Dongle noch geladen werden muss. Dazu mit eingestecktem Dongle:  
`lsusb`  
Ausgabe z.B. : `ID 067b:2303 Prolific Technology, Inc. PL2303 Serial Port / Mobile Phone Data Cable`  
Diese Info nutzen wir, um den Kernel zu „informieren“:  
`modprobe usbserial vendor=0x067b product=0x2303`  
Sollte nach einem Neustart diese Information verloren sein, so wird das Kernelmodul nicht automatisch geladen oder steht auf einer Blacklist. In extrem seltenen Fällen ist etwas selbst zu kompilieren. Normalerweise reicht aber der richtige Umgang mit den verfügbaren Modulen. Siehe auch  
<https://wiki.ubuntuusers.de/Kernelmodule/>      <https://wiki.debian.org/Modules>



# rigctld nutzen

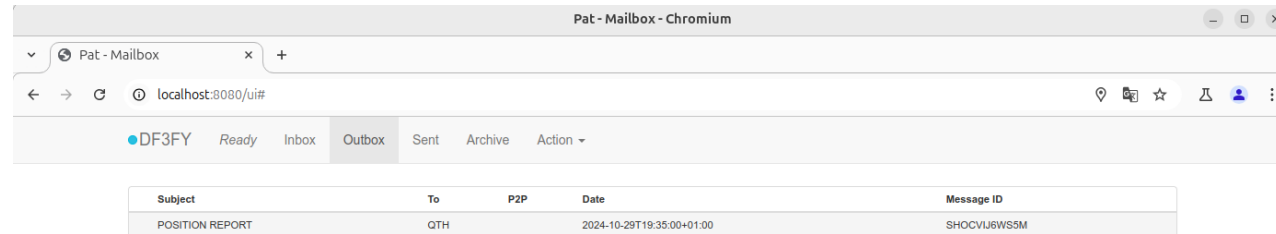
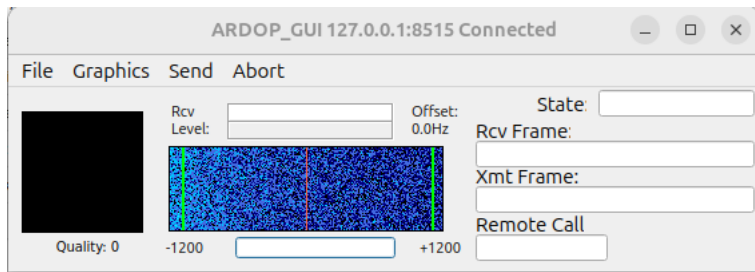
- Mit folgendem Befehl holen wir uns die Info, mit der rigctld den TRX anspricht  
`rigctld -l`
- In der angezeigten Liste steht die Kennnummer unseres TRX.  
(Beispiel Elecraft K2: 2021)
- Mit der Angabe im Startscript (siehe S. 13) von

```
rigctld --model=2021 -r /dev/ttyUSB0 -s 4800
```

liegen alle Informationen zur Steuerung des TRX durch Pat vor.  
Es ist ein K2, über ttyUSB0 angeschlossen und mit einer Baudrate von 4800 zu steuern.

# Start Pat User-Interface, Modem und GUI

- Pat kann nicht nur im Befehlszeilen-Terminal genutzt werden, sondern auch sehr komfortabel mit einem GUI, angezeigt im Browser.
- Die Befehle im Startscript (S. 13)  
`pat -listen "ardop,telnet" http`  
mit nachfolgender Öffnung der Browser-Seite  
`chromium 'http://localhost:8080'`  
bewirken diesen Vorgang.
- Folgende Befehle starten das eigentliche ARDOP-Modem sowie das Modem-GUI.  
`ardopc 8515 hw:0,0` (8515 ist die Portnummer, hw:0,0 die Bezeichnung des Moduls der Soundkarte, welches individuell zu identifizieren ist)  
`ARDOP_GUI`



# Soundsystem identifizieren und einbinden

- Die Terminal-Abfrage

`aplay -l`

gibt Auskunft über die verbaute Soundkarte sowie deren Komponenten.

```
erhard@erhard-HP-250-G8-Notebook-PC: ~  
erhard@erhard-HP-250-G8-Notebook-PC:~$ aplay -l  
**** Liste der Hardware-Geräte (PLAYBACK) ****  
Karte 0: sofhdadsp [sof-hda-dsp], Gerät 0: HDA Analog (*) []  
  Sub-Geräte: 1/1  
  Sub-Gerät #0: subdevice #0  
Karte 0: sofhdadsp [sof-hda-dsp], Gerät 3: HDMI1 (*) []  
  Sub-Geräte: 1/1  
  Sub-Gerät #0: subdevice #0  
Karte 0: sofhdadsp [sof-hda-dsp], Gerät 4: HDMI2 (*) []  
  Sub-Geräte: 1/1  
  Sub-Gerät #0: subdevice #0  
Karte 0: sofhdadsp [sof-hda-dsp], Gerät 5: HDMI3 (*) []  
  Sub-Geräte: 1/1  
  Sub-Gerät #0: subdevice #0  
Karte 0: sofhdadsp [sof-hda-dsp], Gerät 31: HDA Analog Deep Buffer (*) []  
  Sub-Geräte: 1/1  
  Sub-Gerät #0: subdevice #0  
erhard@erhard-HP-250-G8-Notebook-PC:~$
```

Uns interessiert hier die analoge Komponente HDA Analog, die die Gerätenummern #0 #0 führt.

Hiermit erklärt sich die Bezeichnung `hw:0,0` im Modem-Startbefehl bzw. `plughw:0,0` `plughw:0,0`. Manchmal greift auch nur `hw:0.0` `hw:0,0`.

Das alles hängt von der vorhandenen Soundkarte ab. Hier muss ausprobiert werden, auch mit den Lautstärkeinstellungen. Reichen die Einstellungen im normalen Sound-Menü nicht: Dann im User-Terminal: [alsamixer](#)

# Abtastrate anpassen

- Das ARDOP-Modem arbeitet mit 12000 Abtastungen/s.
- Übliche Soundkarten können unter Linux mit einer ganzen Bandbreite von Abtastraten z.B. von 8000 bis 48000 Abtastungen/s arbeiten.
- Dem Linux-Soundsystem ALSA (advanced linux sound architecture) ist nun per zusätzlicher ALSA-Konfigurationsdatei mitzuteilen, die Abtastrate des Modems auf die Standardrate von 48000 zu „interpolieren“. Der folgende Befehl leistet das Gewünschte.

```
pcm.ARDOP {type rate slave {pcm "hw:0,0" rate 48000}}
```

- Hierzu starten wir das User-Terminal und geben `nano` ein. Es öffnet sich eine einfache Textdatei im Terminal auf der Ebene des Persönlichen Ordners. In diese kopieren wir den „Ratenanpass-Befehl“ hinein. Mit strg o wird gespeichert und die dann erfragte Dateibezeichnung als `.asoundrc` angegeben. Den Punkt vorm Dateinamen nicht vergessen. Mit strg x wird der Texteditor nano wieder verlassen.
- Mehr zum Audio-Thema unter <https://wiki.debianforum.de/Audiokonfiguration>

# Startscript anfertigen

Damit die Anwendung funktioniert, müssen mehrere Dinge gestartet werden, nämlich Pat, rigctld, Browser, Modem und Modem-GUI. Hierzu legen wir uns ein Startscript mit dem Inhalt des nachstehenden Kastens auf den Desktop des PCs.

```
#!/bin/bash
ardopc 8515 hw:0,0 &
rigctld --model=2021 -r /dev/ttyUSB0 -s 4800 &
pat -listen "ardop,telnet" http &
chromium 'http://localhost:8080' &
ARDOP_GUI
```

Haben wir ein ardopc64, dann steht im Script eben ardopc64 statt ardopc, wie im o.a. Beispiel. Gleiches gilt für den Browser, statt chromium könnte hier auch firefox stehen. Bei rigctld natürlich die Parameter des eigenen TRX verwenden! Alternativer Ladebefehl `ardopcf 8515 plughw:0,0 plughw:0,0 &` bei dem neueren Modem (zweiter Downloadlink).

User-Terminal öffnen und mit `cd Schreibtisch` auf die Ebene des Desktops bringen.

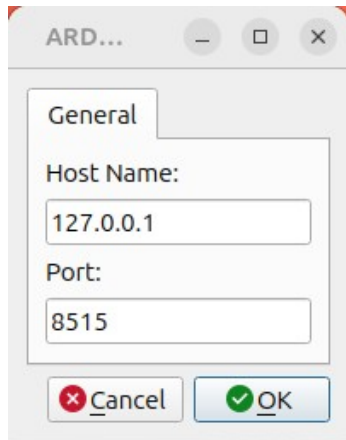
`nano` eingeben und hier den Inhalt des Kastens rein kopieren. Mit `strg o` wieder speichern und als Dateinamen `Startpat.sh` vergeben. Mit `strg x` den Editor nano verlassen.

Wir suchen nun die Datei `Startpat.sh` auf dem Desktop und klicken mit der rechten Maustaste drauf, dann unter Eigenschaften „Als Programm ausführbar“ wählen. Je nach PC-HW ist mit dem Modem-Start etwas zu experimentieren.

# Starten des Scriptes

Auf dem Desktop mit der rechten Maustaste auf das Script klicken und „Als Programm ausführen“ wählen.

Es kommt u.a. ein Fenster des Modem-GUI hoch und benötigt die Befüllung zweier Felder:



ARD... [min] [max] [close]

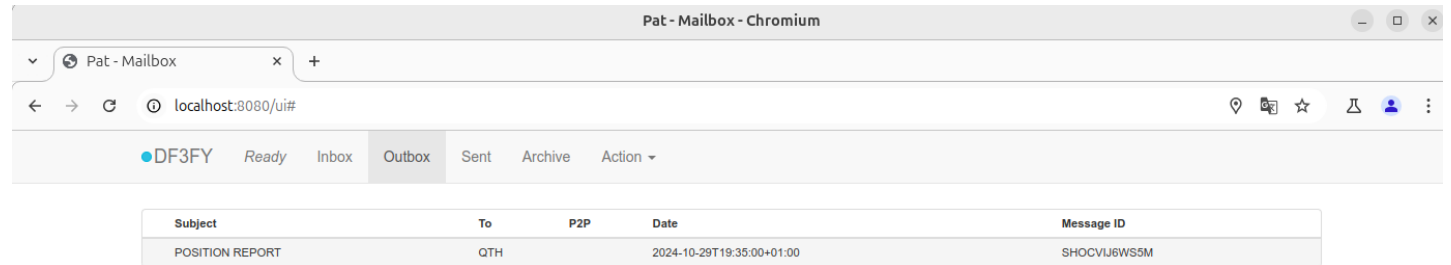
General

Host Name:

Port:

[Cancel] [OK]

Das war schon alles. Die Bedienung von Pat im Browser ist selbsterklärend.



Pat - Mailbox - Chromium

Pat - Mailbox x +

localhost:8080/ui#

●DF3FY Ready Inbox Outbox Sent Archive Action

Subject	To	P2P	Date	Message ID
POSITION REPORT	QTH		2024-10-29T19:35:00+01:00	SHOCVJ6WS5M

# Beenden des Scriptes

Beim Beenden der Programme ist zu bedenken, dass nach dem Schließen der ARDOP-GUI und des Browsers im Hintergrund noch Pat, ardopc sowie rigctld weiter laufen. Startet man nun wieder das Startscript, so laufen Pat und ardopc doppelt mit der Folge, dass sich diese nicht mehr richtig miteinander verbinden. Man muss in so einem Fall diese Teile mittels der Systemüberwachung oder dem Taskmanager manuell beenden oder sich als User kurz ab- und wieder anmelden. Ein kompletter Neustart des PCs ist nicht nötig.

Man kann sich aber auch ein Ende-Script auf den Desktop legen. Die Vorgehensweise ist analog zum Start-Script. Sollte killall bei Debian noch nicht funktionieren, dann mit

`apt install psmisc`  
installieren.

```
#!/bin/bash  
killall ardopc &  
killall rigctld &  
killall pat &  
killall chromium &  
killall ARDOP_GUI
```

# Ansteuerung von VARA im Startscript

- Hierzu im Startscript einfach den Zeiger auf die VARA.exe statt auf ardopc und ARDOP\_GUI legen (VARA muss vorher mit Wine installiert worden sein). Zwecks Auswahl der Modems kann man sich zwei Startscripts auf den Desktop legen.

```
#!/bin/bash  
wine /home/meinusername/.wine/drive_c/VARA/VARA.exe &  
rigctld --model=2021 -r /dev/ttyUSB0 -s 4800 &  
pat http &  
chromium 'http://localhost:8080'
```

- Statt `meinusername` ist natürlich der eigene User-Name (klein geschrieben) zu nehmen.



# Vielen Dank !

Diese Unterlage wurden nach bestem Wissen erstellt.

Die „Rezepte“ habe ich mehrfach getestet.

Dennoch kann ich keinerlei Gewähr oder Haftung übernehmen.

Ich bin für Anregungen, Hinweise und Verbesserungsvorschläge dankbar. Linux lebt wie der Amateurfunk vom Mitmachen und Kommunizieren.

Viel Spaß und Erfolg wünscht  
Erhard, DF3FY  
e-Mail: df3fy(at)darc.de

[https://www.darc.de/fileadmin/filemounts/distrikte/g/ortsverbaende/25/Notfunk/20241104\\_DF3FY\\_Vortrag\\_Winlink\\_mit\\_Pat\\_und\\_ARDOP.pdf](https://www.darc.de/fileadmin/filemounts/distrikte/g/ortsverbaende/25/Notfunk/20241104_DF3FY_Vortrag_Winlink_mit_Pat_und_ARDOP.pdf)

<https://www.darc.de/der-club/distrikte/g/ortsverbaende/25/notfunk/>