

Einsatz der Linux-Software „Pat“ als Ersatz für den Windows-WinLink-Client „RMS Express“

Die die Installation des WinLink-Clients auf einem Linux-Rechner mittels Wine ist relativ aufwendig. Das SW-Modem VARA wird weiterhin benötigt; hierfür gibt es noch keine Linux-Alternative. Die Installation von VARA mittels Wine gestaltet sich allerdings weniger aufwendig (keine weitreichenden .dotnet- / Visual C++ Bibliotheken nötig) wie bei RMS-Express.

Die Linux-Software „Pat“ kann daher ein sinnvoller Ersatz sein, v.a. bei älterer HW (leistungsschwächere Plattformen, Raspberry-PI ...).

Nachfolgend werden Root-Befehle in **roten** Schrifttypen, User-Eingaben in **grün** angegeben.

1. Updaten des PC - im Terminal

(Root-Befehle in **roten** Schrifttypen, User-Eingaben in **grün**)

```
sudo apt update
```

```
sudo apt upgrade
```

Bei Debian bringt man zweckmäßiger Weise durch „su –“, das Terminal vorab in den Root-Modus und kann sich dann bei nachfolgenden Root-Befehlen das vor die Befehle zu schreibende „sudo“ sparen (siehe auch Kapitel 3).

2. Ersatz Firefox

Bei weniger performanter Hardware kann man den Browser Firefox durch Chromium ersetzen (Chromium ist schlanker und schneller, besser für schwache HW)

```
sudo apt remove firefox
```

```
sudo apt install chromium
```

 (oder chromium-browser ; je nach Distribution, z.B. bei Ubuntu)

Hat man genug Speicherplatz können beide Browser auch parallel installiert bleiben – dann „removed“ man eben den anderen Browser nicht.

3. Installation von Pat

Die Software ist über

<https://getpat.io/>

zu beziehen.

Die runter geladene Datei (aktuell) pat_0.15.1_linux_amd64.deb (gibt es auch als .rpm oder Quellcode) liegt üblicherweise im Ordner Downloads.

Zur Installation der in „Downloads“ liegenden Datei ist das Terminal in den Root-Modus auf unterste Ebene zu bringen, dies erfolgt mit

su - (z.B. bei Debian)

oder

sudo su - (z.B. bei Ubuntu)

Mittels des Terminals wird jetzt das Paketverwaltungsprogramms „apt“ zur Installation genutzt. Apt erkennt notwendige Abhängigkeiten und installiert diese gleich mit; z.B. Hamlib, falls diese Bibliotheken noch nicht auf dem PC vorhanden sein sollten.

apt install /home/meinusername/Downloads/./pat_0.15.1_linux_amd64.deb

(statt meinusername ist der eigene Username - in Kleinschreibung - zu nutzen)

Die Punkt/Strich-Notation vor der Dateibezeichnung sorgt dafür, dass Apt genau diese installiert und nicht etwa eine ältere Version, die möglicherweise aus dem Bestand der Distribution ladbar ist.

4. Prüfen, ob Hamlib schon richtig installiert ist

Im Terminal eingeben:

rigctld -l

Wenn positiv, erscheint eine Liste mit Nummern für die meisten Funkgeräte. Wenn negativ, nachinstallieren:

sudo apt install libhamlib-utils

Die Funktion „rigctld“ ist in Hamlib enthalten und ermöglicht die Steuerung des TRX (Frequenzen, Bänder, Mode, PTT). Sie wird mit folgender Befehlszeile im Terminal im Hintergrund aktiviert:

rigctld -m 1023 -r /dev/ttyUSB0 -s 9600

Die Nummer „1023“ referenziert den TRX; hier im Beispiel ist es der Yaesu FT-897 und ttyUSB0 ist der aktive USB-Anschluß mit der Baudrate 9600.

5. Pat prüfen

Im Terminal eingeben:

pat http

Wenn ok, kommt u.a. "Starting HTTP service (<http://localhost:8080>)" hoch. Im Browser

findet sich bei gestartetem Pat unter `http://localhost:8080` alles zur Bedienung von Pat. Um Pat zu beenden, einfach das Terminal und den Browser schließen.

6. Pat konfigurieren

Der Befehl

`pat configure`

öffnet im Terminal die Konfiguration. Die Einträge zwischen den lila-farbenen Anführungszeichen sind zu kontrollieren und individuell zu editieren. Mit der Eingabe von `<strg> <o>` werden die Eingaben gespeichert und mit `<strg> <x>` wird der Editor wieder geschlossen. Durch diesen Vorgang entsteht eine **Datei** im persönlichen Ordner unter `.config/pat/config.json`. Diese Datei kann mittels Doppelklick geöffnet und ebenfalls immer wieder geändert werden. Nachfolgend der Inhalt einer solchen `config.json`-Datei, angepasst auf den FT-897. Der Anpassungsbedarf ist gelb markiert. Wenn die Konfiguration nicht gleich auf Anhieb klappen sollte, lohnt ein Blick auf die Ausgaben im Terminal.

```
-----
{
  "mycall": "DL1ABC",
  "secure_login_password": "Meinpasswort",
  "auxiliary_addresses": [],
  "locator": "JO12AB",
  "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": {
    "hamlib":{"address": "localhost:4532", "network": "tcp"}
  },
  "ax25": {
    "port": "wl2k",
    "beacon": {
      "every": 3600,
      "message": "Winlink P2P",
      "destination": "IDENT"
    },
    "rig": "FT-897"
  },
  "serial-tnc": {
    "path": "/dev/ttyUSB0",
    "serial_baud": 9600,
    "hbaud": 1200,
    "type": "Kenwood"
  },
}
```

```

"ardop": {
  "addr": "localhost:8515",
  "arq_bandwidth": {
    "Forced": false,
    "Max": 500
  },
  "rig": "FT-897",
  "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": "FT-897",
  "ptt_ctrl": true
},
"varafm": {
  "host": "localhost",
  "cmdPort": 8300,
  "dataPort": 8301,
  "bandwidth": 0,
  "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
}

```

7. VARA

Das SW-Modem Vara wird weiterhin benötigt. Es ist nur als Windows-Programm in 32-Bit-Architektur verfügbar. Unter Linux kann man es mittels Wine nutzen. In Wine ist zu beachten, dass die aus Visual-Basic stammende Datei MSVBVM60.DLL in Großschreibung im persönlichen Ordner unter .wine/drive_c/windows/system32 enthalten ist.

Die Datei MSVBVM60.DLL sollte nach Installation von vb6run mittels Winetricks vorliegen. Voraussetzung ist, dass natürlich Wine und Winetricks installiert sind, in Kurzform nachfolgend beschrieben:

```
apt install wine
dpkg --add-architecture i386 && apt update && apt install wine32
apt install winetricks
chmod 666 /dev/ttyS0
usermod -a G dialout $USERNAME (statt $USERNAME den eigenen Linux-User-Namen nehmen (klein geschrieben))
usermod -a -G tty $USERNAME
usermod -a -G audio $USERNAME
```

(Die letzten vier vorstehenden root-Befehle erlauben dem Linux-User die Nutzung der seriellen Schnittstellen und Audiooutput.)

Geben Sie nun im User-Modus ein (user-Befehle sind grün, root-Befehle rot)

```
WINEARCH=win32 winecfg
winetricks vb6run
winetricks sound=alsa
```

Mit

```
wine vara_setup.exe
```

wird dann noch Vara selbst installiert. Nach dem Runterladen liegt die Vara_Installations.exe im Download-Ordner, in welchem man sich mit dem Terminal befinden muss.

In `.wine/drive_c/windows/system32` muss die Datei `pdh.dll` in der richtigen Version liegen, sonst wird VARA beim ersten Start sofort wieder abstürzen. Um das zu verhindern ist eine neue Datei `pdh.dll (151,3 kB)` ebenfalls in den Ordner `.wine/drive_c/windows/system32` einzufügen; die dort bereits vorhandene ungeeignete `pdh.dll`-Datei wird bei diesem Vorgang ersetzt. Diese neue Datei ist unter

<http://files.k6eta.com/pdh.dll.zip>

zu bekommen. Unter nachstehendem Link sind weitere Informationen zum Thema nachlesbar.

<http://k6eta.com/linux/installing-rms-express-on-linux-with-wine>

An dieser Stelle empfehle ich alternativ, die Wine-Version aus WineHQ zu verwenden und nicht, die Versionen, die in den Repositorien der Distributionen verfügbar sind. In dem Papier `Winlink_Vara_Linux_v4.1.pdf` habe ich das diesbezügliche Verfahren beschrieben. Wine liegt bei WineHQ in der Version 9.0 vor. Verwendet man Vara in der aktuellen Version und Wine aus WineHQ, so läuft das SW-Modem ohne weitere Maßnahmen „out of the box“.

8. Anlegen eines Start-Skript "WinLink.sh"

Um mit Pat e-Mails über KW-Funk mittels des SW-Modems VARA verschicken zu können, müssen vier Programme gestartet werden: 1. Hamlib/rigctld, 2. Pat, 3. ein Browser sowie 4. VARA.

Hierzu legt man einfach eine bash-Datei auf den Desktop und schreibt die Befehle zum Starten dieser Programme da rein.

Eine solche Datei kann man z.B. durch Öffnen des Text-Editors „gedit“ und Eintrag folgender Zeilen (#!/bin/bash ist die erste Zeile, die andern starten die vier Programme) erzeugen. Je nach Linux Distribution hat man z.B. ein Gnome-Terminal oder z.B. auch ein xfce4-Terminal. Diese Informationen sind dem Skript mit zu geben.

```
#!/bin/bash
wine /home/meinusername/.wine/drive_c/VARA/VARA.exe &
gnome-terminal -e "rigctld -m 1023 -r /dev/ttyUSB0 -s 9600" &
gnome-terminal -e "pat http" &
firefox 'http://localhost:8080'
```

(die http-Adresse ist sicherheitshalber in 'http://localhost:8080' (Hochkomma) zu setzen, um Fehlinterpretationen auszuschließen)

Beim Speichern nennt man diese Start-Datei z.B. „Pat.sh“ (Dateibezeichnung ohne Anführungszeichen schreiben, das .sh besagt, dass es ein Shell-Script ist). Wenn man statt **firefox** einen anderen Browser hat, dann muss man natürlich diesen ansprechen z.B. mit **chromium**. Analog muss man beim Terminal schreiben, um welchen Typ es sich handelt; hier im Beispiel ist es das **gnome-terminal** und kein **xfce4-terminal**.

Mit der rechten Maustaste klickt man nun auf die Datei – und unter „Eigenschaften“ dann auf „Zugriffsrechte“. Hier ist „Datei als Programm ausführen“ zu wählen.

Dieser Vorgang geht auch im Terminal. Liegt die Skriptdatei z.B. auf dem „Schreibtisch“ so verfährt man nach dem Start des Terminals folgendermaßen:

```
cd Schreibtisch
chmod +x Pat.sh
```

Jetzt kann man alles, was man für WinLink braucht, durch „Ausführen der der .sh-Datei als Programm“ zusammen starten. Im sich öffnenden Browser findet man eine für e-Mail übliche Umgebung. Unter der Schaltfläche „Ready“ ist dann nur bei „Transport“ noch der Modus der Übertragung auszuwählen – für KW eben VARA HF.

Wenn alles gut läuft und man die Informationen aus den sich öffnenden Terminals nicht (mehr) benötigt, kann man die Hintergrundprogramme natürlich auch ohne die Terminals starten – die Start-Skripte vereinfachen sich dann entsprechend:

```
#!/bin/bash
wine /home/meinusername/.wine/drive_c/VARA/VARA.exe &
rigctld -m 1023 -r /dev/ttyUSB0 -s 9600 &
pat http &
chromium 'http://localhost:8080'
```

Im vorstehenden Beispiel wird der Chromium-Browser genutzt.

Achtung: Wenn man seine Pat-Session durch Schließen des Browsers beendet, laufen die Hintergrundprogramme Pat und rigctld weiter, was nicht weiter schlimm ist. Allerdings ist das ein Fallstrick wenn man das Ganze erstmalig einrichtet und Änderungen an den Konfigurationen vornimmt. Die getätigten Änderungen greifen nämlich nicht, solange die Programme nicht geschlossen und neu gestartet werden. Viele User führen dann unnötige Neustarts des gesamten PCs durch und verschwenden damit Zeit. Hier reicht es, die Programme einfach im Taskmanager/Systemüberwachung (meistens unter Hilfsprogramme im Anwendungsmenü zu finden) zu beenden.

9. Winlink-Mails via Internet verschicken und abrufen

Hat der PC eine Internetverbindung (WLAN / Ethernet-Anschluss), dann kann man sein Winlink-Postfach auch mit Telnet bedienen. Das Fenster in Pat ist dann mit seinem Rufzeichen und einem CMS (Core Message Server) zu befüllen. Beispiel:

Connect to remote node... ✕

(select alias) ▾

transport: Telnet ▾ target: wl2k

address: DL1ABC:CMSTelnet@cms.winlink.org:8772

Radio Only

telnet://DL1ABC:CMSTelnet@cms.winlink.org:8772/wl2k

Show RMS list Cancel Connect

Viel Spaß und Erfolg wünscht Erhard, DF3FY