

Nutzung von WinLink und VARA auf Linux-PCs

Für die Nutzung der originären Windows-Software WinLink und VARA auf Rechnern mit einem Linux-Betriebssystem, ist die Installation der Adaptionsschicht „Wine“ erforderlich.

Inhaltsverzeichnis

1. Einführung	2
2. Installation von Wine	2
3. Vorbereitung der WinLink- und Vara-Installation	4
4. Installation der Anwendungen WinLink und VARA	5
5. Konfiguration von WinLink und VARA	7
6. Weitere Windows-Amateurfunkanwendungen auf Basis von Wine	8
7. Nutzung von WineHQ statt der Wine-Versionen aus den Distributionen	9

1. Einführung

Die Wine-Umgebung ist eine Portierung der Windows-Programmschnittstellen auf Linux und MacOS. Da es sich um eine Simulation der Schnittstellen handelt, erfordert der Installationsvorgang etwas mehr Aufwand als unter einer Windows-Umgebung, führt aber zu einer einwandfreien Funktionsfähigkeit ohne Verlust an Performance.

Im Folgenden wird die Vorgehensweise auf Basis der Debian-basierten Linux-Distributionen wie UBUNTU, Debian, Linux-Mint u.v.m. beschrieben - sowohl für die Installation von Wine als auch die Installation von WinLink und VARA.

Es hat sich gezeigt, dass die in den Software-Repositories der Distributionen vorhandenen Wine-Versionen allesamt meistens aktuell sind und ein Anlegen der WineHQ-Software-Quelle zwecks Zugriff auf die aktuellste Wine-Version nicht erforderlich ist – zumal es sich bei WinLink und VARA auch noch um 32-Bit-SW-Architekturen handelt. Wir reden hier also nicht über die neuste Gaming-Software.

Für die Installation ist eine permanente Internetverbindung erforderlich, da hier auf die Online-Repositories der jeweiligen Linux-Distribution zugegriffen wird.

Bei der Installation von Linux-Software-Paketen hat man die Wahl zwischen der Nutzung von Befehlszeilen in einem Terminal oder der Verwendung eines Programms mit graphischer User-Oberfläche (GUI) z.B. Synaptic.

In dieser Anleitung wird die Vorgehensweise mit Terminal beschrieben, da sie schneller zum Ziel führt und durch die Verwendung von „copy & paste“ keine tieferen Linux-Kenntnisse der Anwenders erforderlich sind. Hierzu einfach die wahlweise in rot oder grün geschriebenen Befehle mit der Maus markieren, kopieren und in das Terminal-Fenster einfügen.

2. Installation von Wine

Öffnen Sie das Terminal und erlangen Sie Root-Rechte (Administrator-Rechte)

su -

bei Debian oder

sudo su -

bei Ubuntu oder – Mint.

Kopieren Sie subsequent jede rot markierte Befehlszeile in das Terminal und bestätigen mit **Return**. Wenn der ausgelöste Vorgang abgeschlossen ist, erscheint wieder die Eingabeaufforderung und die nächste Zeile kann eingegeben werden.

Mit dem folgenden Befehl wird die Liste der installierbaren Software aktualisiert:

apt update

apt dist-upgrade

Dieser letzte Befehl dauert ggfs. etwas länger, wenn das letzte Update bereit einige Zeit zurück liegt.

```
dpkg --add-architecture i386 && apt update && apt install wine32
```

Nach Durchführung dieser Befehle geben Sie **exit** ein oder schließen Sie das Terminal-Fenster und starten es anschließend erneut. Dieser Vorgang führt dazu, dass das Terminal nun wieder im User-Modus geöffnet ist und nicht mehr im Root-Modus (Administrator-Modus).

Geben Sie im User-Modus ein (User-Befehle sind in Grün markiert)

```
WINEARCH=win32 winecfg
```

In der nun geöffneten Wine-Konfiguration auf „ok“ klicken. Dieser Befehl stellt in Wine die Architektur auf Windows-32-Bit-Architektur ein. Mit dem Befehl

```
grep '#arch' ~/.wine/system.reg
```

kontrollieren Sie, ob als Ausgabe „arch=win32“ erscheint.

Nun wird das Terminal wieder in den Root-Modus gebracht und zwar wieder wie oben mit

```
su -
```

bzw.

```
sudo su -
```

Jetzt werden noch weitere Pakete installiert und Systemeinstellungen vorgenommen; letztere erlauben den Linux-Usern die Benutzung der seriellen Schnittstellen des PCs.

```
apt install exe-thumbnailer winetricks zenity
```

```
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
```

Nun wird das Terminal geschlossen und wieder neu gestartet, um mit normalen User-Rechten fortzufahren.

Jetzt beenden wir die Super-User-Sitzung mit: **exit**

Kontrolle als was ich eingeloggt bin: Der Befehl **whoami** zeigt Ihren Login-Namen an.

Beispiel:

```
root@rechner:/home/erhard# whoami
root
root@rechner:/home/erhard# exit
exit
erhard@rechner:~$ whoami
erhard
erhard@rechner:~$
```

3. Vorbereitung der WinLink und VARA-Installation

Geben Sie nachfolgend wieder Zeile für Zeile ein und warten jeweils die Abarbeitung ab.

```
winetricks win7  
winetricks sound=alsa  
winetricks -q dotnet35sp1  
winetricks -q dotnet40  
winetricks -q dotnet48  
winetricks vb6run  
winetricks vcrun2015
```

Die letzten drei Befehle brauchen relativ lange zur Durchführung, bitte nicht die Geduld verlieren.

Bei neueren Distributionen und Nutzung von WineHq habe ich festgestellt, dass nur

```
winetricks -q dotnet48
```

installiert werden muss, dotnet40 wird automatisch von dotnet48 nachgezogen.

Das Durchlaufen der .dotnet-Pakete sowie der vcrun2015 ist eine ziemliche Oper - es hagelt Fehlermeldungen, man hat das Gefühl, das Ganze ist eingefroren - einfach ruhig abwarten, bis der Eingabe-Prompt wieder erscheint. Falls irgendwelche Fenster – auch zu Fehlermeldungen - hoch kommen, einfach auf ok bzw. bestätigen klicken und fortfahren lassen.

Mit der Eingabe von

```
winetricks
```

ohne jegliche weitere Angabe kommt die Winetricks-Oberfläche hoch, mit der man kontrollieren kann, ob die dotnets sowie die vb- und vc-runs auch wirklich installiert sind.

Wenn sich das Fenster öffnet, auf "ein Programm installieren" klicken und auf ok. Wenn dann die Liste erscheint, auf "Abbrechen" klicken, nämlich nur da nach kommt ein Fenster hoch, wo man oben "Windows-DLL installieren" auswählen kann. In der sich dann öffnenden Liste findet man (im Erfolgsfall) durch runter scrollen die vier installierten Windows-System-Programme.

Bitte in jedem Fall kontrollieren, ob dotnet35sp1, dotnet40, dotnet48, vb6run sowie vcrun2015 installiert sind. Ansonsten geht WinLink nicht.

Man kann natürlich auch alle fünf Windows-System-Pakete über diese Winetricks-Oberfläche installieren, es dauert aber länger.

4. Installation der Anwendungen WinLink und VARA

Alle folgenden Eingaben werden ebenfalls im Terminal mit normalen User-Rechten vorgenommen.

Hierbei gehe ich davon aus, dass die beiden Installationsdateien im Ordner "Downloads" liegen; diese erhält man von

<https://www.winlink.org/WinlinkExpress>

sowie

<https://rosmodem.wordpress.com/>

Die aktuellen Dateinamen nach dem Download (Stand 23.01.24) lauten:

VARA HF v4.8.1 Setup.zip und **Winlink_Express_install_1-7-12-0.zip**.

Nach dem Entpacken lauten die Dateinamen **Winlink_Express_install.exe** sowie **VARA setup (Run as Administrator).exe**, sie liegen in extra Unterordnern. Man kann die beiden entpackten Dateien nun in den Ordner Downloads schieben. Zuvor muss man die Dateibezeichnung der VARA-Datei umbenennen, da das Terminal die Klammer-Symbole nicht akzeptiert, z.B. in **VARAsetup.exe**.

Der Befehl

```
cd Downloads
```

bringt das Terminal auf die Ebene „Downloads“ und mit

```
wine ./Winlink_Express_install.exe
```

(Dateiname ggfs. an deine Wirklichkeit anpassen) sowie

```
wine ./VARAsetup.exe
```

(Dateiname ggfs. an vorliegende Verhältnisse anpassen)

wird die eigentliche Installation gestartet. Der Verlauf ist so, wie bei allen Windows-Programmen. Man kann auch Desktop-Symbole anlegen lassen - die Installationsprogramme fragen das ab.

VARA wird beim ersten Start sofort wieder abstürzen. Um dass zu verhindern

muss die Datei **pdh.dll (151,3 kB)** in den Ordner

```
„~/wine/drive_c/windows/system32“
```

eingefügt werden und ersetzt die dort bereits vorhandene pdh.dll-Datei. Diese Datei ist unter

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

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

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

Die Datei

MSVBVM60.DLL

befindet sich ebenfalls in diesem Ordner „wine/drive_c/windows/system32“, möglicherweise aber in Kleinschreibung als

msvbvM60.dll

Bitte kontrollieren und ggfs. einfach durch Umbenennen in Großbuchstaben setzen und möglichen Fehlern wird vorgebeugt.

(Anmerkung: Die Dateinamen, die mit einem Punkt beginnen (wie z.B .wine/drive_c/windows/system32) sind sogenannte verborgene Dateien/Verzeichnisse, die mit Setzen eines Häkchens in den Einstellungen des Linux-Dateimanagers sichtbar gemacht werden können.)

Jetzt sollte auch VARA laufen (Vara-HF-SW-Modem) - die SW ist wirklich gut, fast so gut wie ein externes Pactor4-HW-Modem.

Wenn nach/bei der Installation von WinLink sinngemäß die Meldung erscheinen sollte „Windows-Version ist nicht passend für die Anwendung“, so öffnet man den Wine-Konfigurator mittels der Terminal-Eingabe

winecfg

und fügt die WinLink-Anwendung, die mit „RMS Express“ bezeichnet ist, hinzu. Hierzu „Anwendung hinzufügen“ auswählen und die RMS Express.exe Startdatei (nicht die Installationsdatei unter Downloads) hinzufügen. Sie ist unter drive_c direkt im Ordner RMS Express zu finden. RMS Express.exe wird dann mit aufgelistet. Nun markiert man RMS Express.exe und wählt unter Windows-Version „Windows-8.1“ aus und klickt auf „Anwenden“. Dann startet man die Installation erneut. Das Terminal merkt sich bereits eingegebenen Befehle. Mittels der ▲ (Pfeil nach oben) – Taste kommt man an die jeweils vorher genutzten Befehle.

5. Konfiguration von WinLink und VARA

Für WinLink ist ein Account und für VARA eine SW-Lizenz erforderlich. Beides ist für eine geringe Gebühr erhältlich.

WinLink

<https://www.winlink.org/>

VARA

<https://rosmodem.wordpress.com/>

Nach der Installation sind die Lizenz-Keys einzutragen.

Es empfiehlt sich, WinLink so einzustellen, dass beim Start einer VARA-Session in WinLink, WinLink nicht automatisch VARA startet. Hierzu in WinLink nach Eröffnung einer Vara HF WinLink Session unter „Settings“ und dort unter „Vara TNC Setup“ das Häkchen bei „Automatically launch Vara TNC when session is opened“ entfernen. Man muss nun vorm Start von WinLink VARA manuell starten. Das hat bei schwächerer PC-Hardware den Vorteil, dass es nicht zu einem Toggeln zwischen beiden Anwendungen kommt.

In einigen Fällen kann es passieren, dass Wine nicht automatisch die richtigen Soundkarten anspricht: Hier ist ein Link zu den Einstellungen der Soundkarte für Wine.

<https://www.marcopeter.ch/2023/01/12/ubuntu-22-04-audio-treiber-in-wine-auswaehlen/>

Alternativ zu meinen obigen Beschreibungen kann man auch im REGEDIT nicht "alsa", sondern "pulse" eintragen.

Wenn die PTT oder CAT über USB gesteuert werden, ist ein symbolischer Link in der Datei

`~/wine/dosdevices` zu erstellen.

Hierzu steckt man zunächst den Schnittstellen-Dongle in eine USB-Buchse und stellt mit der Eingabe von `sudo dmesg | grep tty` fest, wie das Teil heißt – Ausgabe ist z.B. `usb 1-1: pl2303 converter now attached to ttyUSB0`.

Im Terminal jetzt nur noch

```
ln -s /dev/ttyUSB0 ~/wine/dosdevices/com2
```

ausführen, wenn com2 genutzt werden soll.

Dieser Befehl erzeugt einen sogenannten symbolischen Link auf eine andere Datei. Man kann damit von einer anderen Stelle im Dateisystem auf die selbe Datei zugreifen.

Ist dieser Name im Zielverzeichnis jedoch bereits vorhanden, so erhält man eine Fehlermeldung.

Es ist sehr einfach, mit dem folgenden Befehl zu überprüfen, welche Dateinamen bereits vorhanden sind:

```
ls -lha ~/.wine/dosdevices/
```

Die weiteren Einstellungen sind selbsterklärend und hängen von dem zu verwendenden Transceiver ab.

6. Weitere Windows-Amateurfunkanwendungen auf Basis von Wine

Wenn man - wie beschrieben - Wine für WinLink und VARA frisch auf Linux installiert hat, so ist auch die Installation von weiteren, für den Amateurfunk interessanten Windows-Programmen kein Problem mehr, z.B. MMANA-GAL zur Antennenberechnung oder der Mini-Ringkernrechner sind gute Beispiele hierfür. Besonders empfehlenswert ist die zusätzliche Installation von VARA Chat. Hiermit kann man sich direkt von PC zu PC ohne Gateway und ohne ein Mail-Formular verständigen und auch Dateien austauschen. Vara Chat benötigt keinen zusätzlichen Lizenz-Key.

<https://rosmodem.wordpress.com/>

Bei der Installation der Programme wie oben beschrieben wieder einfach im Terminal mit normalen User-Rechten die Installation starten.

Wenn die Installationsdatei unter Downloads liegt:

```
cd Downloads
```

```
wine MMANA_GAL_DE.exe (Beispiel)
```

Benötigt ein Windowsprogramm die Funktionalität „HTML-Seiten anzeigen“ – z.B. um Hilfedokumente – zu präsentieren, so ist das Programm „Gecko“ zu installieren.

Unter

<https://wiki.winehq.org/Gecko>

findet man für seine Wine-Version und für die 32-Bit-Architektur das passende Paket zum Download. Mittels der Terminal-Eingabe

```
wine --version
```

erfährt man seine Wine-Version und kann sich so das passende Gecko-Paket auswählen.

Hat man eine 6.0.n er Wine-Version, so findet man nach dem Download im Ordner Downloads eine Datei namens **wine-gecko-2.47.2-x86.msi**. Diese Datei ist nun zu installieren. Hierzu im Terminal

wine uninstaller

eingeben und in der sich öffnenden Anwendung auf „Installieren“ klicken. Es öffnet sich dann ein weiteres Fenster, worin man seine unter Downloads befindliche Datei aufsuchen kann. Bei der Auswahl des Datei-Typs ist das entsprechende Dropdown-Menü auf „alle Dateien“ zu setzen, per default werden nur .exe-Dateien angezeigt.

Mit dieser Methode können auch noch andere System-Programme zu Wine hinzu installiert werden.

7. Nutzung von WineHQ statt der Wine-Versionen aus den Distributionen

Es ist nicht auszuschließen, dass sich die Winlink-Client-Software RMS-Express nicht mittels der Wine-Pakete aus den Distributionen installieren lässt. Einige User haben über diese Schwierigkeiten berichtet. Bei Ubuntu 22.04 LTS hat Wine den Stand 6.0.3. Diese Wine-Version ist relativ alt. Es ist daher angezeigt, die WineHQ-Pakete zu nutzen und auch für RMS-Express einen getrennten Wine-Prefix anzulegen.

An dieser Stelle muss man definitiv festhalten, dass die Installation von RMS-Express unter Linux mit Wine kein Selbstläufer ist. Wenn man es aber geschafft hat, funktioniert das Ganze absolut problemlos.

Bevor man überhaupt etwas von Wine installiert, sind die Paketquellen von WineHQ zu installieren. Damit wird erreicht, dass die Wine-SW immer mit automatisch aktualisiert wird. Wenn man schon das Wine aus der Distribution installiert haben sollte, so ist es besser, alles Wine wieder zu deinstallieren und mit WineHQ frisch zu beginnen. Dazu löscht man auch den Ordner `.wine` im persönlichen Verzeichnis.

Xubuntu-Version 23.10

Ich habe das Verfahren mit der aktuellen Xubuntu-Version 23.10 nachvollzogen. Nachstehend führe ich die Schritte auf, die bei mir am 23.01.24 zum Erfolg geführt haben.

Die aktuelle WineHQ Beschreibung kann man hier ergänzend nachvollziehen.
<https://wiki.winehq.org/Download>

Anlegen der Paketquellen:

```
sudo dpkg --add-architecture i386
sudo mkdir -pm755 /etc/apt/keyrings
sudo wget -O /etc/apt/keyrings/winehq-archive.key
https://dl.winehq.org/wine-builds/winehq.key
```

(Achtung, die blauen https-Adressen gehören an die Befehle, sind nur mit Leertaste getrennt, rutschen hier im Text in eine neue Zeile. Bitte mit C&P ins Terminal übertragen.)

```
sudo wget -NP /etc/apt/sources.list.d/
```

```
https://dl.winehq.org/wine-builds/ubuntu/dists/mantic/winehq-mantic.sources
```

```
sudo apt update
```

```
sudo apt install --install-recommends winehq-stable
```

Nach diesen Schritten ist das Wine von WineHQ auf dem PC.

Nach Eingabe von

```
WINEARCH=win32 winecfg
```

entsteht im Persönlichen Ordner ein Unterordner mit der Bezeichnung `.wine`. Das ist der Standard-Wine-Prefix.

Mit

```
grep '#arch' ~/.wine/system.reg
```

Überzeugt man sich, dass wirklich die 32-Bit-Architektur gezogen wurde.

Programme, die mittels des Befehls

```
wine meinprogramm.exe
```

installiert werden, landen dort.

Wie schon oben für die „normale“ Wine-Installation beschrieben, bitte noch folgende Installationen bzw. Einstellungen vornehmen:

```
apt install exe-thumbnailer winetricks zenity
```

```
usermod -a -G dialout $USERNAME (statt $USERNAME den eigen Linux-User-Namen nehmen (klein geschrieben))
```

```
usermod -a -G tty $USERNAME
```

```
usermod -a -G audio $USERNAME
```

Für das verzwickte Programm RMS-Express kann man einen separaten Wine-Prefix anlegen. Dieser Befehl erstellt den Prefix `.wine-rms` und installiert direkt `dotnet48`. Das `dotnet35sp1` wird offenbar bei neuen Versionen nicht mehr benötigt.

```
WINEARCH=win32 WINEPREFIX="$HOME"/.wine-rms' winetricks dotnet48
```

(Bitte mit C&P ins Terminal übertragen!)

Wenn man keine anderen Wine-Architekturen außer der win32 benötigt, kann man den „Leitweg Wine-Prefix“ weglassen. Also einfach eingeben:

```
winetricks dotnet48
```

Erst jetzt kommen die anderen Komponenten hinzu. Hierbei muss – damit das Ganze im neuen Prefix .wine-rms landet – immer der Ziel-Prefix mitgegeben werden.

```
WINEPREFIX="$HOME"/.wine-rms' winetricks vb6run
```

oder nur `winetricks vb6run`

```
WINEPREFIX="$HOME"/.wine-rms' winetricks vcrun2015
```

oder nur `winetricks vcrun2015`

```
WINEPREFIX="$HOME"/.wine-rms' winetricks sound=alsa
```

oder nur `winetricks sound=alsa`

```
WINEPREFIX="$HOME"/.wine-rms' winetricks win7
```

oder nur `winetricks win7`

Nun checken wir, ob im neuen Wine-Prefix unsere dotnets sowie vb6run und vcrun2015 angekommen sind:

```
WINEPREFIX="$HOME"/.wine-rms' winetricks
```

oder nur `winetricks`

Es sollte sich nun die GUI von Winetricks öffnen und uns nach dem eingangs beschriebenen Verfahren die Prüfung gestatten.

Wenn alles vorhanden ist, können wir in den Ordner Downloads wechseln und die Installation von RMS-Express vornehmen. Unbedingt auch jetzt den Prefix mitgeben oder – verwendet man keinen Prefix – diesen weglassen.

```
cd Downloads
```

```
WINEPREFIX="$HOME"/.wine-rms' wine winlink_express_install.exe
```

oder nur [wine winlink_express_install.exe](#)

(Die Bezeichnung der .exe Installations-Dateien so umbenennen, dass die Klammern im Namen des Originals nicht mehr enthalten sind.)

Es empfiehlt sich Vara ebenso in den selben Wine-Prefix zu installieren. Das aktuelle Vara lief bei mir ohne die weiteren eingangs beschriebenen Maßnahmen „out of the Box“.

Debian 12

Das gleiche Verfahren habe ich am 25.01.24 mit Debian 12 erfolgreich durchgeführt.

Man muss hierzu natürlich von WineHQ die Debian-Quellen anlegen:

```
sudo dpkg --add-architecture i386
sudo mkdir -pm755 /etc/apt/keyrings
sudo wget -O /etc/apt/keyrings/winehq-archive.key
https://dl.winehq.org/wine-builds/winehq.key
```

(Achtung, die blauen **https-Adressen** gehören an die Befehle, sind nur mit Leertaste getrennt, rutschen hier im Text in eine neue Zeile. Bitte mit C&P ins Terminal übertragen.)

```
sudo wget -NP /etc/apt/sources.list.d/
https://dl.winehq.org/wine-builds/debian/dists/bookworm/winehq-bookworm.sources
```

```
sudo apt update
```

Nun kann man auch in Synaptic nachgucken und feststellen, dass winehq-stable als Installationspaket sichtbar ist. Mit

```
sudo apt install --install-recommends winehq-stable
```

ist die Installation des aktuellsten Wine erledigt. Das Weitere dann genau wie zuvor für Xubuntu beschrieben.

Viel Glück und viel Spaß

vy 73 Erhard, DF3FY