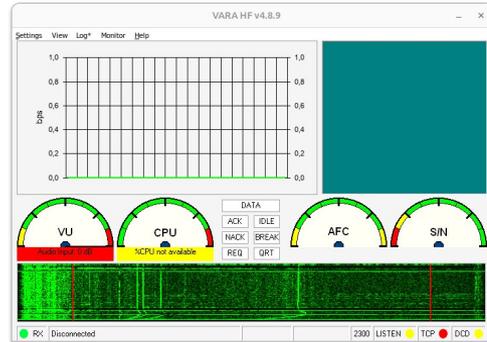


Winlink und VARA unter Linux

Installation mittels der Adaptionsschicht Wine



The WineHQ logo, featuring a wine glass filled with red wine, next to the text "WINEHQ". Below the logo is a section titled "Aktuelle Versionen" (Current Versions). It lists the "Stabile Version:" (Stable Version) as "Wine 9.0 (Neuerungen (en))" and the "Entwicklungsversion:" (Development Version) as "Wine 9.19 (Neuerungen (en))". Below this is a section titled "Über" (About) with the text "Erfahren Sie mehr über das Wine-Projekt." (Learn more about the Wine project.)

Hintergrund

- Für die Nutzung der originären Windows-Software WINLINK und VARA auf Rechnern mit einem Linux-Betriebssystem ist die Installation der Adaptionsschicht „Wine“ erforderlich.
- WINLINK und VARA sind mit Windows gewachsen und basieren auf der 32-Bit-Struktur.
- Wine „übersetzt“ die Windowsbefehls- und Programmstrukturen in die von Linux.
- Bei richtiger Installation und Konfiguration ist mit Wine eine gleichwertige Nutzung dieser Windowsprogramme möglich.

Installation von Wine 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 Wine auf Linux ist bis auf die Wahl der Paketquellen nahezu immer 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 oder Mint.
- Schritt 1 sollte ein Update des vorhandenen Linux-Systems sein.
apt update
apt upgrade

32-Bit-Architektur und Paketquellen

Linux-Systeme werden heute fast immer als 64-Bit-Architektur angeboten. Die 32-Bit-Fähigkeit muss für 32-Bit-Windowsprogramme nachträglich zugefügt werden.

```
dpkg --add-architecture i386 && apt update  
mkdir -pm755 /etc/apt/keyrings  
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.)

```
wget -NP /etc/apt/sources.list.d/  
https://dl.winehq.org/wine-builds/ubuntu/dists/noble/winehq-noble.sources
```



UBUNTU

Vorgehensweise bei Debian (1/2)

Bevor man WineHQ auf Debian nutzen kann, ist die Quellenliste um nicht-Debian- und am besten auch gleich um nicht-freie-Quellen zu erweitern (ist bei MX-Linux, ein Debian mit anderem UI, nicht nötig):

Im Terminal

`nano /etc/apt/sources.list`

eingeben und die Datei um die gelb markierten Einträge ergänzen.

Debian

```
-----  
# deb cdrom:[Debian GNU/Linux Buster - .. - Official amd64 xfce-CD Binary-1 20171013-  
13:08]/ bookworm main contrib non-free non-free-firmware  
deb http://ftp.de.debian.org/debian/ bookworm main contrib non-free non-free-firmware  
deb-src http://ftp.de.debian.org/debian/ bookworm main contrib non-free non-free-firmware  
deb http://security.debian.org/debian-security/ bookworm/updates main contrib non-free non-free-firmware  
deb-src http://security.debian.org/debian-security/ bookworm/updates main contrib non-free non-free-firmware  
# stretch-updates, previously known as 'volatile'  
deb http://ftp.de.debian.org/debian/ bookworm-updates main contrib non-free non-free-firmware  
deb-src http://ftp.de.debian.org/debian/ bookworm-updates main contrib non-free non-free-firmware  
-----
```

Mit Tasten „strg o“ Inhalt speichern und mit „strg x“ die Datei im Terminal wieder verlassen und apt neu laden:

`apt update`

Vorgehensweise bei Debian (2/2)

Wie bei Ubuntu nur andere Quellen bei WineHQ.

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

```
mkdir -pm755 /etc/apt/keyrings
```

```
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.)

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

```
https://dl.winehq.org/wine-builds/debian/dists/bookworm/winehq-bookworm.sources
```



Debian

Installation der WineHQ-Pakete

In den Schritten zuvor haben wir die Paketquellen von WineHQ angelegt und können damit die stets aktuellste Version von Wine verwenden. Die nächsten Befehle machen die WineHQ-Pakete in der apt-Paketverwaltung verfügbar und damit installierbar.

apt update

apt install winehq-devel (bei Ubuntu)

apt install winehq-stable (bei Debian)

(Die Version winehq-stable ist derzeit noch nicht für Ubuntu 24.04 verfügbar – nach einer Reifezeit wird auch dort winehq-stable verfügbar sein. Für uns hat das keine funktionale Bedeutung.)

Wine auf dem PC entfalten

Root-Terminal schließen und wieder starten (oder statt schließen **exit** eingeben), um in den User-Modus zu gelangen

```
WINEARCH=win32 winecfg
```

Prüfen, ob Wine nun in der 32-Bit-Version vorhanden ist

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

Weitere Pakete und Konfigurationen

apt install exe-thumbnailer winetricks zenity

usermod -a -G dialout \$USERNAME (statt \$USERNAME den eigenen Usernamen in klein nehmen)

usermod -a -G tty \$USERNAME (statt \$USERNAME den eigenen Usernamen in klein nehmen)

usermod -a -G audio \$USERNAME (statt \$USERNAME den eigenen Usernamen in klein nehmen)

chmod a+rw /dev/ttyUSB0 (USB-Dongle der Steuerleitung zum TRX eingesteckt)

winetricks win7

winetricks sound=alsa

winetricks -q dotnet48

winetricks vb6run

winetricks vcrun2015

Die Winetricks-Befehle (**nicht im Root-Modus machen !**) dauern in der Durchführung – Geduld also !

Installation von Winlink und VARA

Nach dem Download von RMS-Express (Winlink) und VARA liegen die gepackten Dateien in der Regel im Ordner „Downloads“.

Mit der rechten Maustaste auf hier entpacken / entpacken klicken.

In den korrespondierenden neuen Unterordnern mit der rechten Maustaste auf die Installations.exe-Dateien klicken und „Öffnen mit Wine Windows-Programmstarter“ wählen (Ubuntu) bzw. „Öffnen mit rundll32“ (Debian).

Alternativ im User-Terminal (mit dem Terminal auf der Ebene des Ordners (`cd Downloads`)) die .exe-Dateien zur Installation starten, z.B.

```
wine Winlink_Express_install.exe
```

Serielle Schnittstelle konfigurieren (1/4)

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.

Es empfiehlt sich zum Prüfen des Dongles / der Schnittstelle zum TRX erst mal originäre Linux-Amateurfunk-Programme zu installieren und zu testen, z.B. flrig mittels des Befehls

```
apt install flrig
```

Hiermit wird vermieden, bei einer Fehlfunktion der Sst. mit einer Fehlersuche bei Wine zu starten.

Serielle Schnittstelle konfigurieren (2/4)

Wir wissen nun wie das Teil heißt und können der Wine-Welt diese Information über einen symbolischen Link mitteilen

```
In -s /dev/ttyUSB0 ~/.wine/dosdevices/com5
```

Es kann sein, dass com5 (als Beispiel) vergeben ist, dann wird eben com3 genommen u.s.w., besser als das Suchen ist aber mit

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

vorher zu gucken, was worauf verlinkt ist. Wenn man sieht, dass dort u.a. com5 -> /dev/ttyUSB0 steht, dann ist com5 richtig.

Serielle Schnittstelle konfigurieren (3/4)

Ein Debian-System mit aktuellem Kernel wird nach der Maßnahme auf S. 13 die COM-Ports in Wine richtig nutzen können und damit die Steuerung eines TRX erlauben.

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 dem 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 verfügbaren Modulen. Siehe auch

<https://wiki.ubuntuusers.de/Kernelmodule/>

<https://wiki.debian.org/Modules>

Serielle Schnittstelle konfigurieren (4/4)

Nun prüfen wir, ob der Kernel mit dem Dongle interagiert:

```
dmesg | grep pl2303
```

Es folgt hierauf eine Ausgabe etwa wie

```
[16961.801274] usbcore: registered new interface driver pl2303
```

```
[16961.801309] usbserial: USB Serial support registered for pl2303
```

```
[16961.801351] pl2303 1-1:1.0: pl2303 converter detected
```

```
[16961.801980] usb 1-1: pl2303 converter now attached to ttyUSB0
```

Damit ist auf Linux-Ebene alles getan.

Eintrag in die Registry

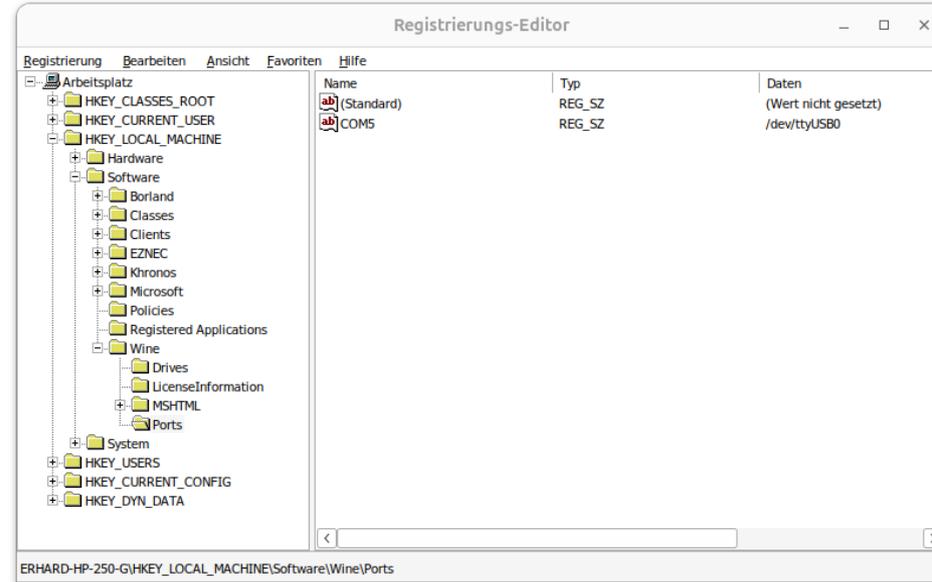
Bei Ubuntu ist die Verlinkung des tty-Devices auf den COM-Port noch mittels des Registrierungseditors nachzutragen. Bei Debian 12 ist das nicht erforderlich. Mit

`wine regedit`

öffnen wir den Windows-Registry-Editor und tragen unter `HKEY_LOCAL_MACHINE / Software / Wine / Ports` z.B. die Bezeichnung `COM5` und den Wert `/dev/ttyUSB0` ein (was wir verlinkt haben). Hierzu im Editor auf „Bearbeiten“ und dann auf „Neu“ sowie „Zeichenfolge“ klicken. Jetzt ins aufgegangene Feld `COM5` schreiben und Return. Mit der rechten Maustaste nun auf den neu entstandenen Eintrag gehen und dann auf „Ändern“ und als Wert `/dev/ttyUSB0` eintragen. Nach Schließen des Editors beendet der Befehl

`wineserver -k`

diese Aktion.



Nachträgliche Installation von Gecko

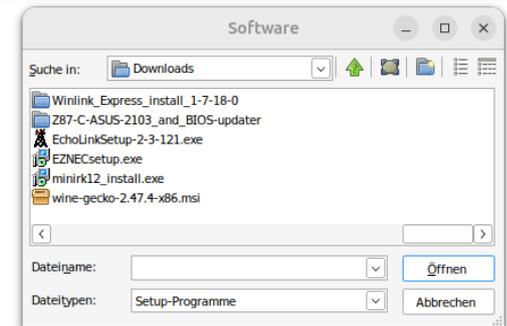
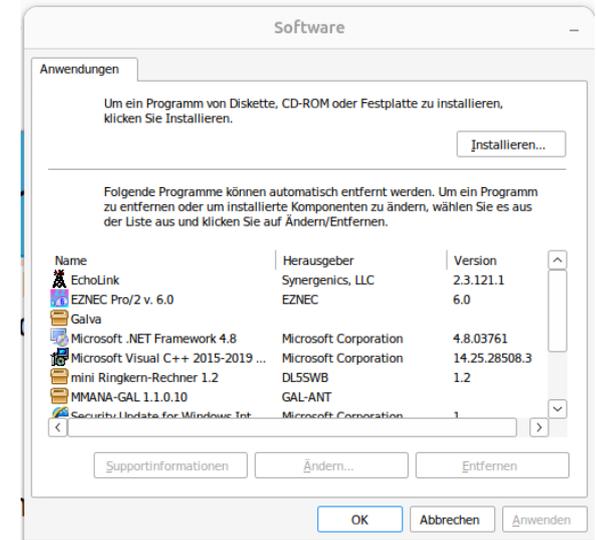
Gecko dient zum Anzeigen von HTML-Dokumenten z.B. Hilfetexte bei Windowsprogrammen. Wenn sich Gecko nicht automatisch bei der ersten Nutzungsanforderung installiert, kann man diese SW manuell installieren: Man lädt von der Seite

<https://gitlab.winehq.org/wine/wine/-/wikis/Gecko>

die 32-Bit-Version z.Zt. wine-8.6 – current herunter (wine-gecko-2.47.4-x86.msi). Im Terminal öffnet man den Uninstaller, mit welchem man SW deinstallieren und installieren kann:

wine uninstaller

Mit Klick auf Installieren gelangt man auf sein Datei-System und kann die .msi-Datei auswählen und per Klick auf Öffnen installieren.



Einrichtung von sudo bei Debian (1/2)

Im Gegensatz zu Ubuntu ist bei Debian der sudo-Befehl nicht standardmäßig eingerichtet. Aus Sicherheitsgründen arbeitet Debian in der Regel mit getrennten User- und Administrator (root)-Passwörtern.

Viele User ziehen aus Gründen der Einfachheit die Verwendung von sudo vor – auch findet sich in den meisten Erklärungen im Internet die Verwendung von sudo.

Für die automatisierte Installation von WineHQ auf Debian und Ubuntu habe ich Skripte geschrieben (die nötigen Befehle sind darin aufgelistet), die die Verwendung von sudo voraussetzen. Die Skripte führen die Befehle sequenziell aus, so werden Fehler bei C&P oder beim Abtippen vermieden. Wer das Skript bei Debian verwenden möchte, muss vorher dort noch sudo einrichten. Umschalten zum Root-Terminal (im User-Terminal `su` – eingeben und Return):

`apt install sudo`

`adduser fritz sudo` (statt fritz den eigenen User-Namen klein geschrieben nehmen)

`exit` (vom Root-Terminal zum User-Terminal schalten)

Zur Überprüfung, ob die Maßnahme gegriffen hat (vorher am System abmelden und wieder anmelden) eingeben:
`sudo -v`

Wenn nach dieser Eingabe das abgefragte User-Passwort akzeptiert wird, ist alles ok.

Einrichtung von sudo bei Debian (2/2)

Je nach Vorkonfiguration kann es sein, dass das auf der vorigen Seite beschriebene Verfahren nicht greift. Dann geht man folgendermaßen vor: Im User-Terminal Root-Rechte holen mit

`su`

Dann nochmals den eingeloggten User (Beispiel fritz) mit folgendem Befehl zur Gruppe sudo hinzufügen:

`usermod -aG sudo fritz`

nach der Eingabe von

`visudo`

erscheint im Terminal der Inhalt der sudo-Konfigurationsdatei und darin fügt man an der Stelle `# User privilege specification` unter der Zeile die mit `root ...` beginnt eine neue Zeile hinzu, nämlich

`fritz ALL=(ALL:ALL) ALL`

mit `strg o` wird gespeichert und mit `strg x` wird die Datei wieder geschlossen. Das Root-Terminal wird verlassen mit `exit`

Nach Aus- und wieder Ein-Loggen wird geprüft, ob sudo nun greift, z.B. mit

`sudo apt update`

Skript zur Installation von WineHQ auf Debian

```
#!/bin/bash
sudo apt update && sudo apt upgrade
sudo dpkg --add-architecture i386 && apt update
sudo mkdir -pm755 /etc/apt/keyrings
sudo wget -O /etc/apt/keyrings/winehq-archive.key https://dl.winehq.org/wine-builds/winehq.key
sudo wget -NP /etc/apt/sources.list.d/ https://dl.winehq.org/wine-builds/debian/dists/bookworm/winehq-bookworm.sources
sudo apt update
sudo apt install winehq-stable
WINEARCH=win32 winecfg
sudo apt install exe-thumbnailer winetricks zenity
sudo usermod -a -G dialout fritz
sudo usermod -a -G tty fritz
sudo usermod -a -G audio fritz
sudo chmod a+rw /dev/ttyUSB0
winetricks win7
winetricks sound=alsa
winetricks -q dotnet48
winetricks vb6run
winetricks vcrun2015
sudo chmod 666 /dev/ttyS0
ln -s /dev/ttyUSB0 ~/.wine/dosdevices/com5
sudo reboot
```

- a) statt **fritz** den eigenen User-Namen klein geschrieben eingeben
- b) vorher gemäß S. 6 die sources.list ergänzen
- c) vorher sudo einrichten
- d) gemäß S. 13 ermitteln, ob com5 oder ein anderer Port richtig ist

Skript zur Installation von WineHQ auf Ubuntu

```
#!/bin/bash
sudo apt update && sudo apt upgrade
sudo dpkg --add-architecture i386 && apt update
sudo mkdir -pm755 /etc/apt/keyrings
sudo wget -O /etc/apt/keyrings/winehq-archive.key https://dl.winehq.org/wine-builds/winehq.key
sudo wget -NP /etc/apt/sources.list.d/ https://dl.winehq.org/wine-builds/ubuntu/dists/noble/winehq-noble.sources
sudo apt update
sudo apt install winehq-devel
WINEARCH=win32 winecfg
sudo apt install exe-thumbnailer winetricks zenity
sudo usermod -a -G dialout fritz
sudo usermod -a -G tty fritz
sudo usermod -a -G audio fritz
sudo chmod a+rw /dev/ttyUSB0
winetricks win7
winetricks sound=alsa
winetricks -q dotnet48
winetricks vb6run
winetricks vcrun2015
sudo chmod 666 /dev/ttyS0
ln -s /dev/ttyUSB0 ~/.wine/dosdevices/com5
Reboot
```

- a) Statt fritz den eigenen User-Namen klein geschrieben eingeben !
- b) gemäß S. 13 ermitteln, ob com5 oder ein anderer Port richtig ist

Verwendung der Skripte

Kopiere den Text von `#!/bin/bash` bis `sudo reboot` in den Texteditor und benenne die Datei als `Skript_Debian.sh` bzw. `Skript_Ubuntu.sh`.

Trage statt **fritz** den eigenen User-Name in Kleinschreibung ein sowie den passenden com-Port ein!

Klicke mit der rechten Maustaste auf das Dateisymbol und setze unter Eigenschaften die Einstellung auf „Als Programm ausführbar“.

Öffne das Terminal und gehe dort hin, wo das Skript liegt, z.B mit `cd Schreibtisch` und starte das Skript mit der Eingabe (Beispiel Ubuntu-Skript) `./Skript_Ubuntu.sh`

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\(at\)darcd.de](mailto:df3fy@darcd.de)