

Compilierung und Installation SVXLINK aus den Github-Sourcen

0. Vorwort

Hier folgt eine Anleitung, wie man auf einem Standard-Linux-System [SVXLINK aus den Github-Sourcen](#) selber compilieren und installieren kann. Ich beziehe mich mit dieser Anleitung auf Linux/Debian 11 (Bullseye), eingeschlossen Raspian für den Raspberry Pi, andere Distributionen werden hier nicht berücksichtigt.

Auch hier gilt, das ist nix für reine Anfänger, denen noch Grundlagenwissen in Sachen Linux fehlt. Grundlegende Kenntnisse im Umgang und der Bedienung eines Linux-System sind also zwingende Voraussetzung, um die folgenden Schritte umsetzen zu können.

1. Vorbereitung

Zuerst bringen wir mal unser System auf aktuellen Stand:

```
$ sudo apt-get update
$ sudo apt-get upgrade
$ sudo apt-get dist-upgrade
```

Dann installieren wir alle notwendigen Tools und Bibliotheken, die zur Compilierung des SVXLINK benötigt werden. Das muss auf dem jeweiligen Linux nur EINMAL erledigt werden:

```
$ sudo apt-get install build-essential git
$ sudo apt-get install libasound2-dev g++ gcc make cmake groff gzip doxygen
tar graphviz
$ sudo apt-get install libsigc++-dev libsigc++-2.0-dev
$ sudo apt-get install libspeex-dev libspeexdsp-dev libopus-dev
$ sudo apt-get install libpopt-dev
$ sudo apt-get install libasound2-dev libgcrypt20-dev libgsm1-dev
$ sudo apt-get install librtlsdr-dev libjsoncpp-dev
$ sudo apt-get install tcl-dev
$ sudo apt-get install libcurl4-openssl-dev
$ sudo apt-get install gpod libgpod-dev
```

Wenn es bisher noch keinen User `svxlink` gab, müssen wir diesen anlegen und einige Gruppenzuweisungen machen. Gibt den User `svxlink` schon, dann die erste Befehlszeile weglassen (`useradd`), aber trotzdem die drei folgenden (`usermod`) ausführen. Auch das ist auf dem jeweiligen System nur EINMAL zu erledigen:

```
$ sudo useradd svxlink
$ sudo usermod -a -G gpio svxlink
$ sudo usermod -a -G audio svxlink
$ sudo usermod -a -G plugdev svxlink
```

Um zu prüfen, ob der User `svxlink` schon existiert und wenn ja, ob die Gruppen schon zugewiesen wurden, macht man folgendes:

```
$ id svxlink
```

und prüft die Ausgabe des Befehls. Da kann man alles überprüfen.

2. Beziehen der Quellen von Github und Compilierung

Zuerst prüfen wir mal, ob es das Verzeichnis `/usr/local/src` gibt. Sollte das nicht der Fall sein, legen wir es an, ansonsten überspringen wir diesen Schritt:

```
$ sudo mkdir /usr/local/src
```

Jetzt geht es weiter:

```
$ sudo chmod 777 /usr/local/src
$ cd /usr/local/src
$ sudo git clone https://github.com/sm0svx/svxlink.git
$ cd svxlink/src/
$ sudo mkdir build
$ cd build
$ sudo cmake -DCMAKE_INSTALL_PREFIX=/usr -DSYSCONF_INSTALL_DIR=/etc -
DLOCAL_STATE_DIR=/var -DUSE_QT=NO -DWITH_SYSTEMD=yes ..
$ sudo make
$ sudo make doc
$ sudo make install
$ sudo ldconfig
```



Sollte unter `/usr/local/src` schon ein Verzeichnis `svxlink` vorhanden sein, löschen wir es am besten vor dem Befehl `git clone`:

```
$ cd /usr/local/src
$ ls -l
drwxr-xr-x  9 root root 4096  9. Jan 18:52 svxlink
$ sudo rm -fr svxlink
```

Das installiert SVXLINK ins Linux, Konfiguration ist unter `/etc/svxlink` zu finden, Zusatzprogramme wie die TCL-Dateien unter `/usr/share/svxlink`. Definitionen wie die Pfadangabe für das Logfile sind in der Datei `/etc/default/svxlink` zu finden und können dort ggf. angepasst werden.

3. SVXLINK konfigurieren

Bevor wir SVXLINK überhaupt starten können, muss jetzt die Konfiguration der `/etc/svxlk/svxlk.conf` an Euer System angepasst werden. Hier **bitte die Dokumentation des SVXLINK studieren** und die korrekten Einstellungen vornehmen, je nachdem was eingerichtet werden soll (Repeater oder FM-simplex-System, ECHOLINK etc.). Eine Beispiel-Konfiguration kann es eigentlich nicht geben, zu verschieden dürfte die Hardware sein, speziell was Soundinterfaces und GPIO-Steuerungen für PTT und/oder SQL angeht. **Das müsst ihr dann den entsprechenden Hardware-Dokumentationen entnehmen und dementsprechend im Setup umsetzen** .

 SVXLINK ist keine Plug'n'Play-Anwendung, sich damit ausgiebig zu beschäftigen ist ein Muss, das kann Euch niemand abnehmen, wenn ihr es alles selbst machen wollt !

4. SVXLINK für automatischen Start vorbereiten

Ich empfehle, um Probleme mit gewissen Abläufen beim Systemstart zu vermeiden, den Autostart des SVXLINK, der standardmäßig mit dem `systemd` erfolgt, etwas zu verzögern. Ein guter Wert wären ca. 30sec.

Dazu bedienen wir uns ebenfalls bei den Möglichkeiten des `systemd`.

```
$ sudo nano /lib/systemd/system/svxlk.timer
```

und fügen in diese `svxlk.timer` folgendes ein:

```
[Unit]
Description=Startup delay SVXLINK

[Timer]
OnBootSec=30
Unit=svxlk.service

[Install]
WantedBy=multi-user.target
```

und speichern diese. Anschließend aktivieren wir diese, damit sie beim Systemstart aktiviert wird:

```
$ sudo systemctl enable svxlk.timer
```

Eine Aktivierung der eigentlichen `svxlk.service` darf bei Einsatz einer zugehörigen `svxlk.timer` nicht erfolgen ! Es wird nur die `svxlk.timer` aktiviert.

Wer mehr darüber wissen will, muss sich mit den Möglichkeiten des `systemd` beschäftigen und dessen Dokumentation lesen.

Um zu prüfen, ob SVXLINK korrekt gestartet wurde, sehen wir uns einfach mal dessen Log an:

```
$ sudo tail -n 500 -f /var/log/svxlk
```

Das zeigt immer die letzten 500 Zeilen der Logdatei vom SVXLINK an, raus kommt man da immer mit

STRG-C.

5. weitere Arbeiten

SVXLINK enthält per Default keine Sound-/Sprachdateien für die Sprachausgabe. Es gibt Sprachpacks, die aber teilweise mit lizenzpflichtigen Text-to-Speech-Applikationen erstellt wurden und damit nicht einfach frei zum Download bereitgestellt werden können. Man könnte selber solche Sprachdateien erstellen mit Open-Source-Tools wie picotts oder man muss eben ggf. eine entsprechende Lizenz erwerben, um schon vorhandene Sprachpacks verwenden zu können. Hier müsst ihr leider selber auf die Suche gehen, wenn es benötigt wird.

Das Format dieser Sprachdateien ist `.wav/11kHz/mono/16bit` und müssen in einer Art Struktur nach `/usr/share/svxlink/sounds/de_DE/` kopiert werden, wobei `de_DE` der eingestellten Sprache in der `svxlink.conf` entspricht, denn SVXLINK ist multilingual:

```
.
├── Airport
├── AnnounceLogic
├── Core
├── Default
├── DtmfRepeater
├── EchoLink
├── Help
├── MetarInfo
├── Own
├── Parrot
├── PhoneLogic
├── PropagationMonitor
├── SelCallEnc
├── SipLogic
├── TclVoiceMail
├── TrafficInfo
└── WeatherInfo
```

Weiter gehe ich aber auf die Sprachunterstützung nicht ein.

Weiterhin gibt es unter `/usr/share/svxlink/events.d` sog. TCL-Scripts, mit deren Hilfe man Ereignis-basierend das SVXLINK anpassen kann. Ein Ereignis wäre z.B. `SQL_OPEN`, also wenn die Rauschsperrung öffnet. Da es sich bei TCL ("Tool command language") um eine Programmier- bzw. Scriptsprache handelt, muss man also lernen, wie diese Sprache angewendet wird, um eigene Anpassungen vornehmen zu können. Das kann an dieser Stelle hier leider nicht erfolgen, da es grundlegende Programmierkenntnisse und -fähigkeiten voraussetzt.

6. Update SVXLINK aus den Github-Quellen

In gewissen Abständen aktualisiert der Programmautor vom SVXLINK, Tobias/SMOSVX, den Programmcode. Wie wir ein Update machen können, beschreibe ich hier. Dazu bedienen wir uns dem Befehl `git pull`, der unser lokales Quellcode-Verzeichnis auf den Stand von Github bringt. Voraussetzung ist also, dass unter `/usr/local/src` ein Verzeichnis `svxlink` existiert, was früher

einmal mittels `git clone` erzeugt wurde.

```
$ sudo systemctl stop svxlink.service svxlink.timer
$ cd /usr/local/src/svxlink
$ sudo git pull
$ cd src
$ sudo rm -fr build
$ sudo mkdir build
$ cd build
$ sudo cmake -DCMAKE_INSTALL_PREFIX=/usr -DSYSCONF_INSTALL_DIR=/etc -
DLOCAL_STATE_DIR=/var -DUSE_QT=NO -DWITH_SYSTEMD=yes ..
$ sudo make
$ sudo make doc
$ sudo make install
$ sudo ldconfig
$ sudo systemctl daemon-reload
$ sudo systemctl start svxlink.service
```

 **HINWEIS:** Wurde innerhalb von `/usr/local/src/svxlink` irgendeine Datei verändert, schlägt `git pull` fehl. Das sog. lokale Repository weicht dann von dem auf Github ab und kann aufgrund lokaler Änderungen nicht mehr so einfach geupdated werden. Der Abgleichmechanismus ist damit praktisch beschädigt. In diesem Fall löschen wir also das `/usr/local/src/svxlink` einfach komplett:

```
$ sudo rm -fr /usr/local/src/svxlink
```

und beginnen wieder bei Pkt. 2 !



Bei einem erneuten `sudo make install` bleiben die Daten unter `/etc/svxlink/` erhalten, die `svxlink.conf` und andere Daten werden also nicht überschrieben.

Anders sieht das bei den TCL-Scripts unter `/usr/share/svxlink/events.d` aus, wenn man dort in den Original-Dateien Veränderungen vorgenommen hat, werden diese überschrieben. Um das zu vermeiden, ist unter `/usr/share/svxlink/events.d` ein weiteres Verzeichnis `local` anzulegen und die geänderten TCL-Scripte dort hinein zu kopieren. Dieses Verzeichnis wird nicht überschrieben.

73 Heiko, DL1BZ

From:

<http://kb.amft-it.de/> - **Amateurfunk - Knowledge Base und Wiki by DL1BZ**

Permanent link:

<http://kb.amft-it.de/doku.php?id=kb-afu:svx-compile>

Last update: **18.01.2023 17:46**

