

Entwicklungssystem installieren Lazarus und Android

(Bernd Jung am 29.02.2024)

Aus dem Internet wird Debian heruntergeladen:

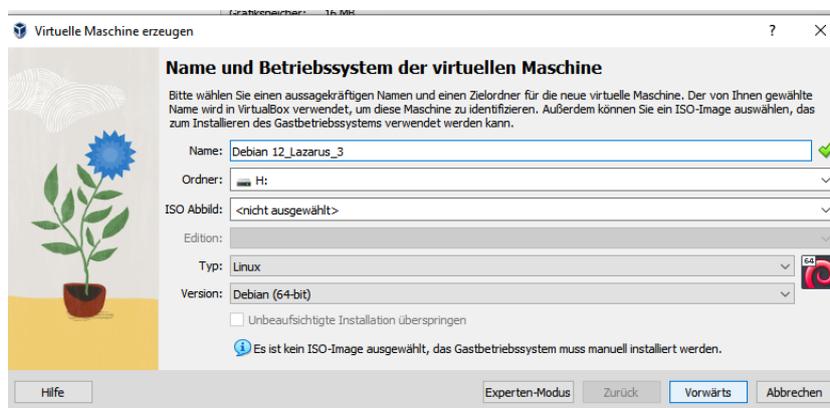
<http://cdimage.debian.org/cdimage/>

VirtualBox Maschine einrichten

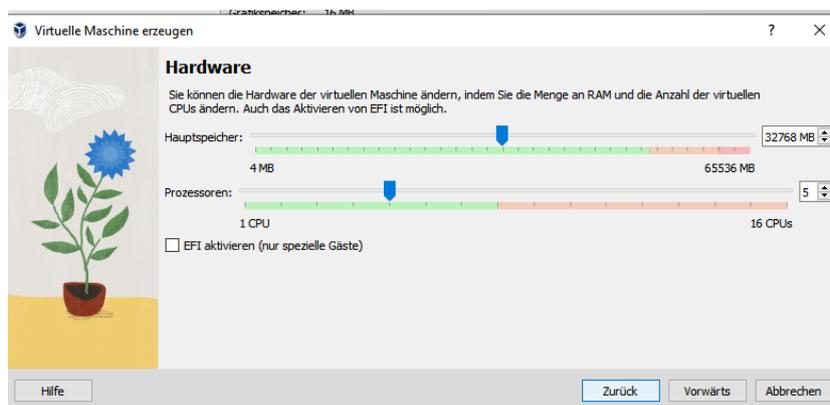
Debian soll unter Virtual-Box installiert werden. Für das passende Betriebssystem findet man Virtualbox im Internet unter:

<https://www.virtualbox.org/wiki/Downloads>

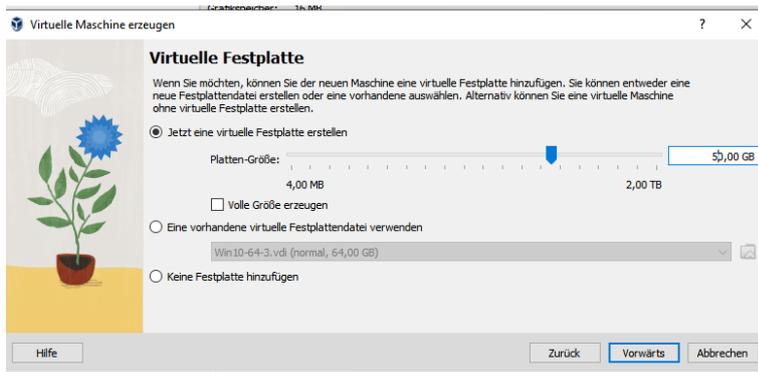
Nach der Installation von Virtualbox wird nun eine neue virtuelle Maschine erzeugt:



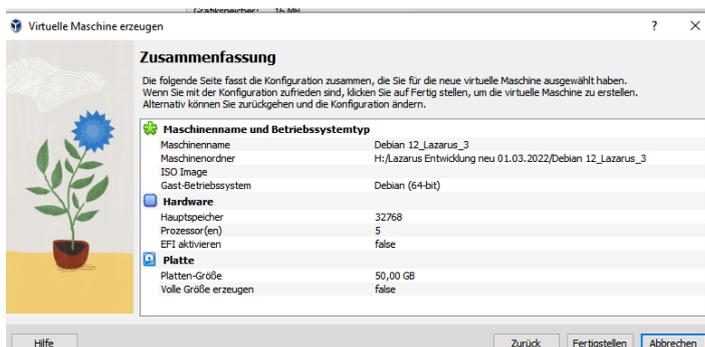
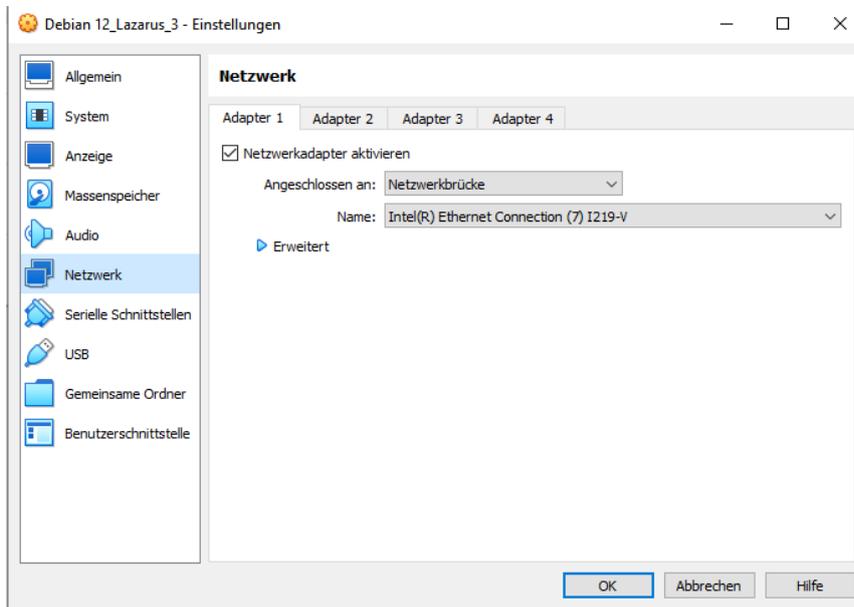
Als Hauptspeicher empfehle ich mindestens 8 GB, wenn das der eigene Computer hergibt.



Eine Festplattengröße von 30 GB sollte ausreichend sein, wenn außer der Entwicklungsumgebung Lazarus nicht zusätzlich laufen soll.



Die Netzwerkkarte habe ich als Brücke konfiguriert, damit eine IP-Adresse aus dem eigenen Netz über DHCP verwendet wird. Es ist dann einfacher die Daten z.B. mit dem Smartphone, das auch über WLAN mit dem Netz verbunden ist, auszutauschen.

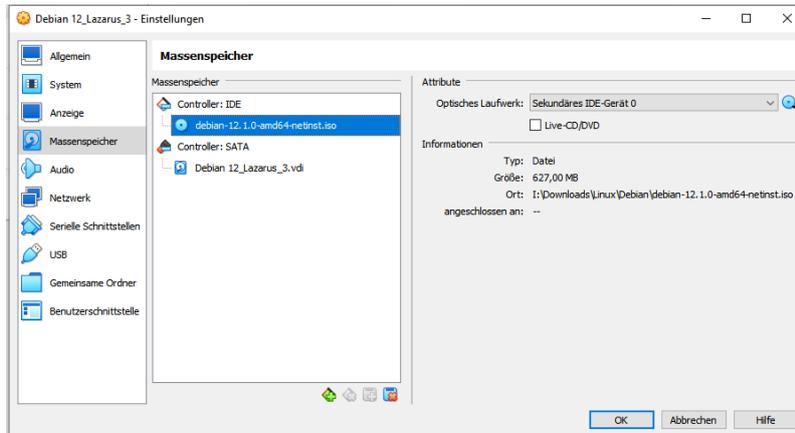


Grundinstallation Debian

Aus dem Internet wird Debian heruntergeladen:

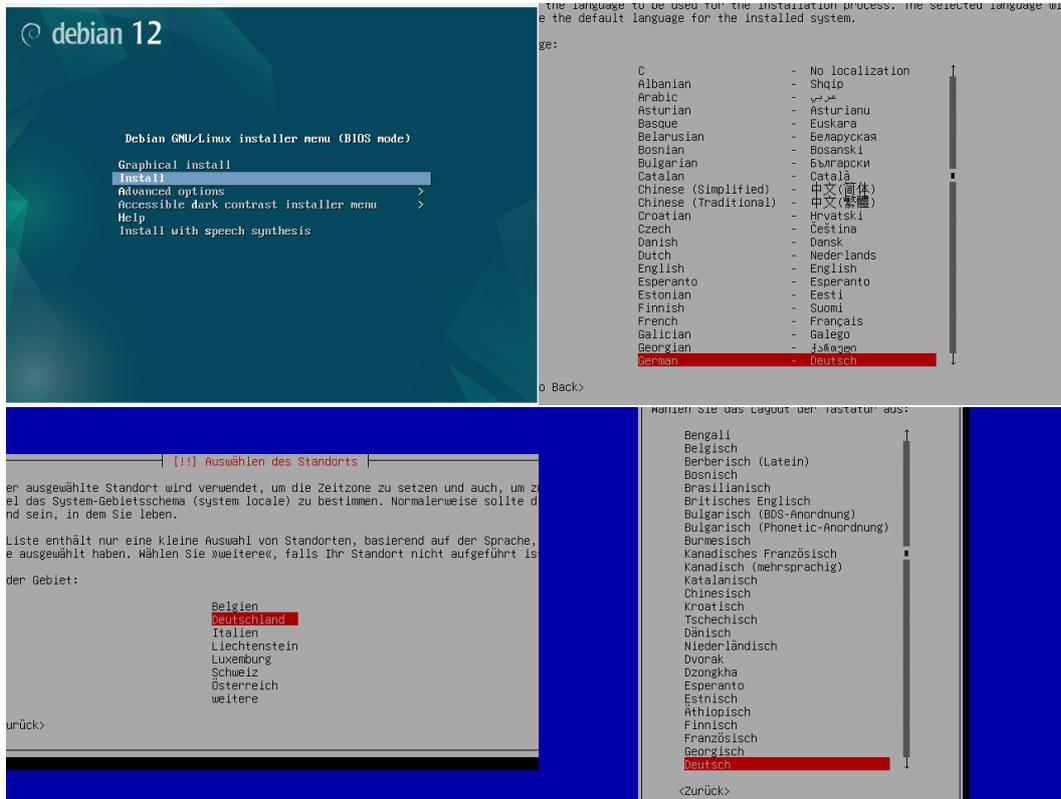
<http://cdimage.debian.org/cdimage/>

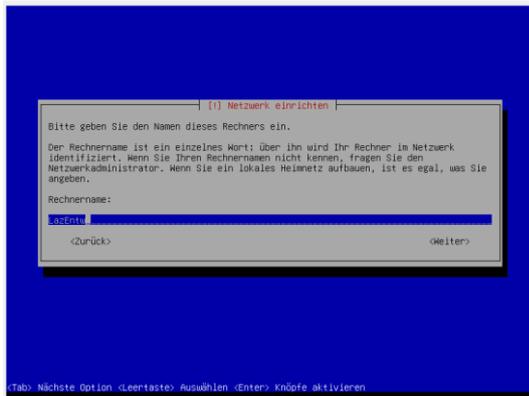
Dann wird die heruntergeladene ISO-Datei in das CD-Laufwerk unserer virtuellen Maschine eingelegt und von dieser CD gestartet.



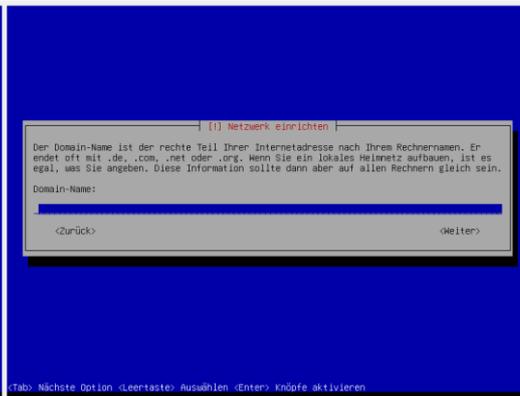
Nun erfolgt die Grundinstallation von Debian ohne allen Schnickschnack. Es werden erst mal nur Standard-Systemwerkzeuge und ein ssh-Server benötigt. Wenn man die grafischen Desktops mit auswählt, dann kommen auch noch Zusatzprogramme mit auf die Platte, die wir ggf. gar nicht benötigen. Den grafischen Desktop xfce installieren wir darum lieber später separat.

Eine Installation könnte dann so ablaufen:





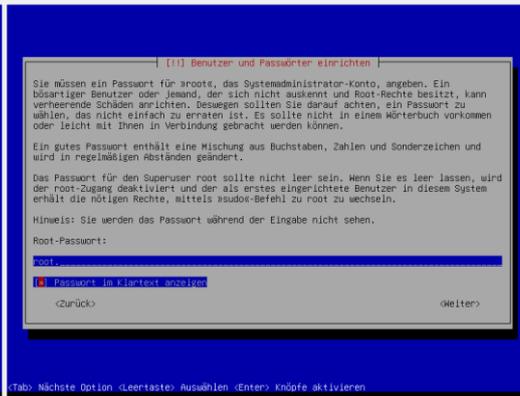
Tab: Nächste Option <Leertaste> Auswählen <Enter> Knöpfe aktivieren



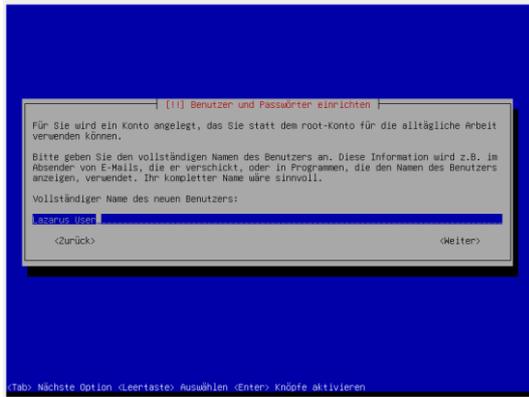
Tab: Nächste Option <Leertaste> Auswählen <Enter> Knöpfe aktivieren



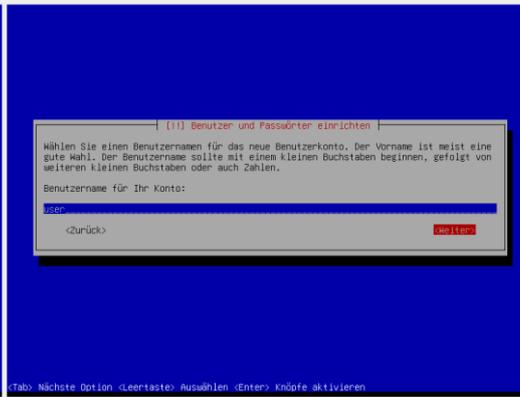
Tab: Nächste Option <Leertaste> Auswählen <Enter> Knöpfe aktivieren



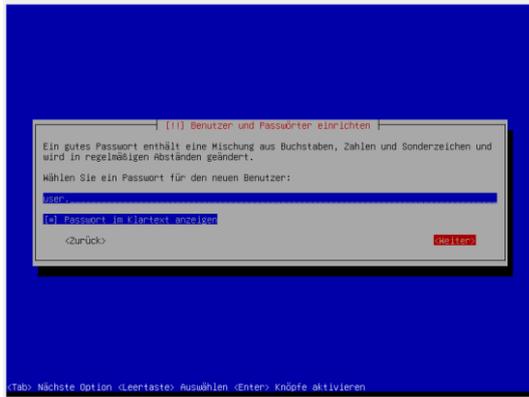
Tab: Nächste Option <Leertaste> Auswählen <Enter> Knöpfe aktivieren



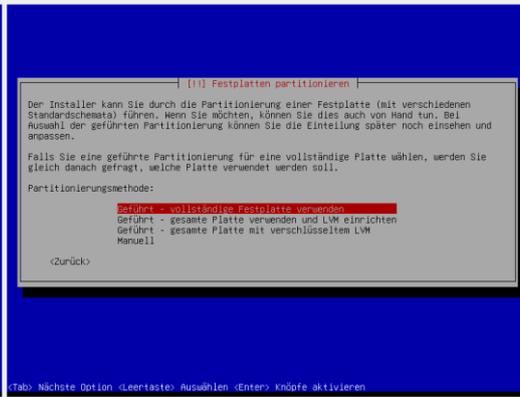
Tab: Nächste Option <Leertaste> Auswählen <Enter> Knöpfe aktivieren



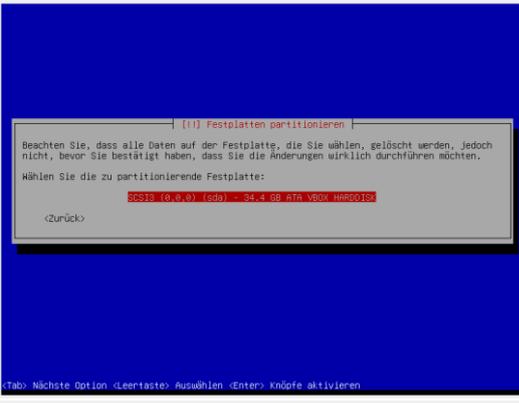
Tab: Nächste Option <Leertaste> Auswählen <Enter> Knöpfe aktivieren



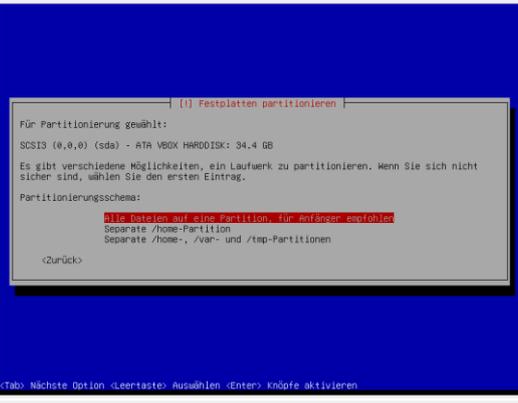
Tab: Nächste Option <Leertaste> Auswählen <Enter> Knöpfe aktivieren



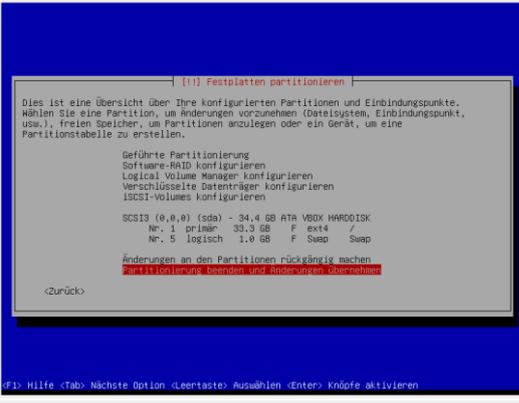
Tab: Nächste Option <Leertaste> Auswählen <Enter> Knöpfe aktivieren



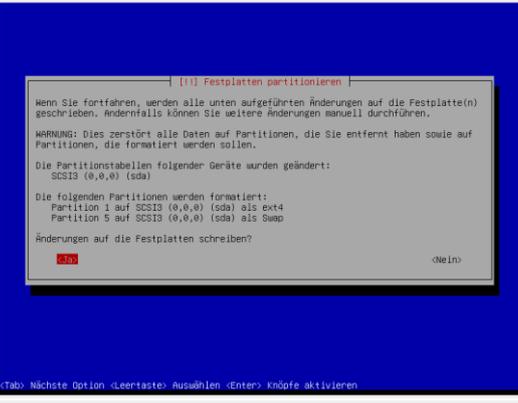
Tab: Nächste Option <Leertaste> Auswählen <Enter> Knöpfe aktivieren



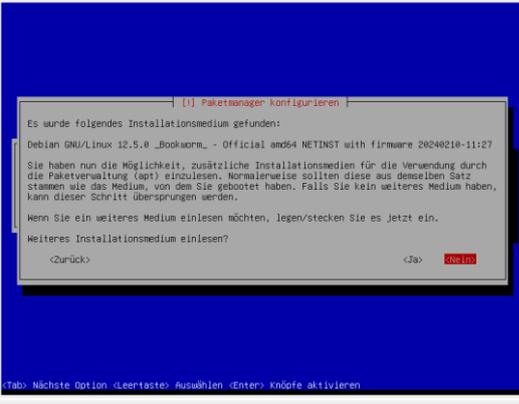
Tab: Nächste Option <Leertaste> Auswählen <Enter> Knöpfe aktivieren



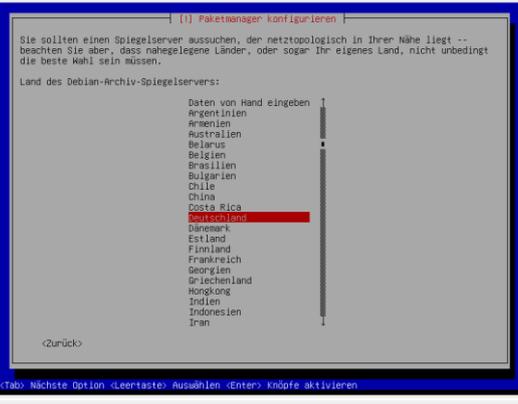
F1) Hilfe <Tab> Nächste Option <Leertaste> Auswählen <Enter> Knöpfe aktivieren



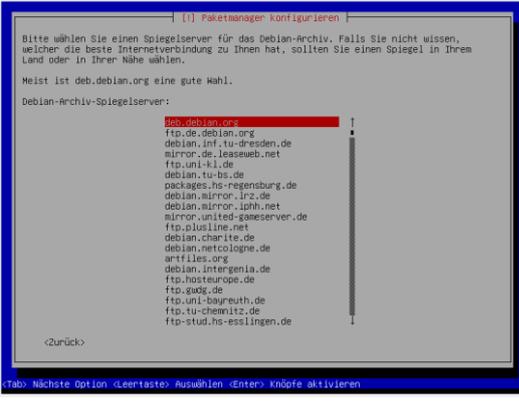
Tab: Nächste Option <Leertaste> Auswählen <Enter> Knöpfe aktivieren



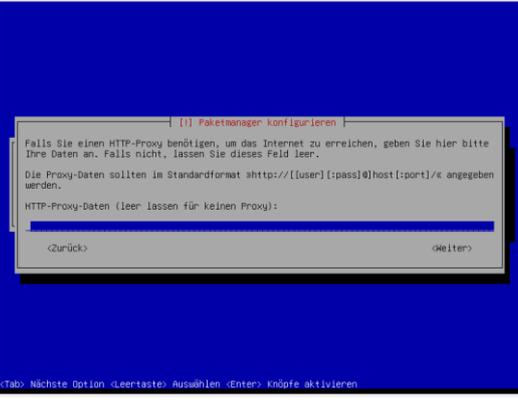
Tab: Nächste Option <Leertaste> Auswählen <Enter> Knöpfe aktivieren



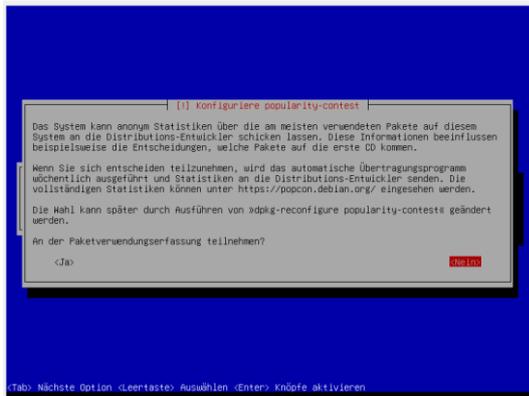
Tab: Nächste Option <Leertaste> Auswählen <Enter> Knöpfe aktivieren



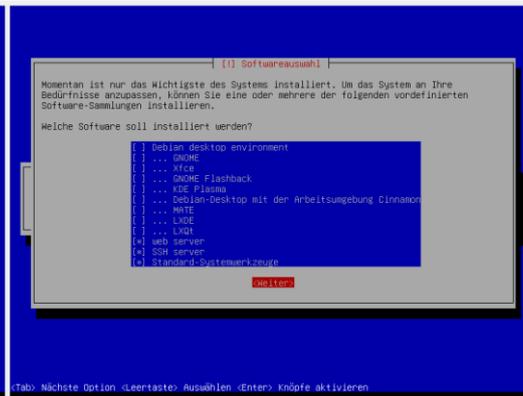
Tab: Nächste Option <Leertaste> Auswählen <Enter> Knöpfe aktivieren



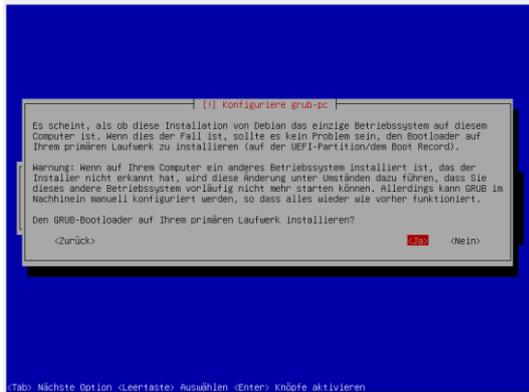
Tab: Nächste Option <Leertaste> Auswählen <Enter> Knöpfe aktivieren



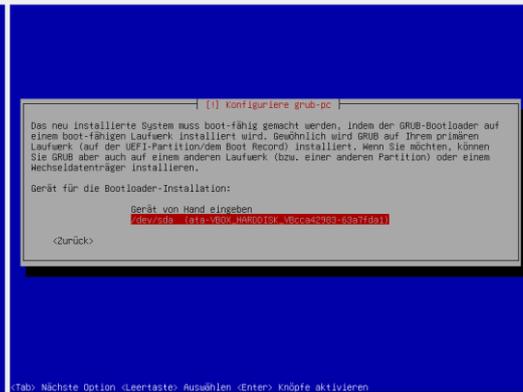
Tab: Nächste Option <Leertaste> Auswählen <Enter> Knöpfe aktivieren



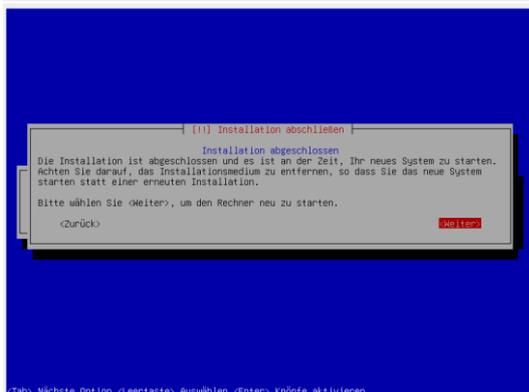
Tab: Nächste Option <Leertaste> Auswählen <Enter> Knöpfe aktivieren



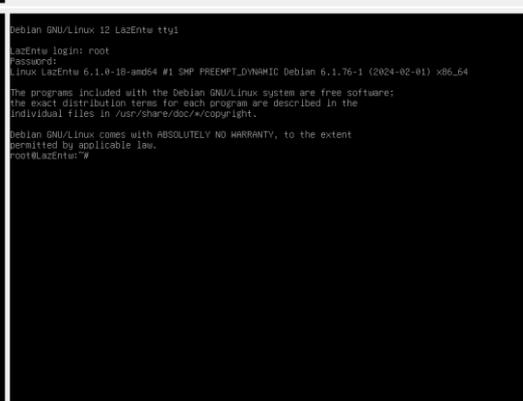
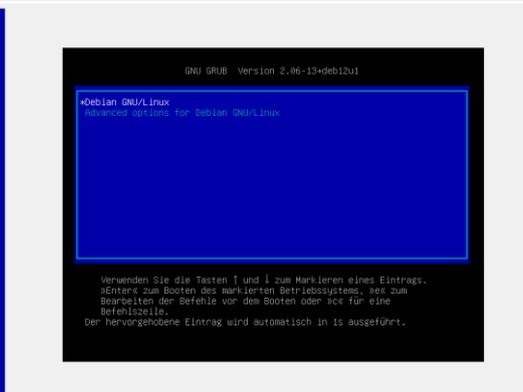
Tab: Nächste Option <Leertaste> Auswählen <Enter> Knöpfe aktivieren



Tab: Nächste Option <Leertaste> Auswählen <Enter> Knöpfe aktivieren



Tab: Nächste Option <Leertaste> Auswählen <Enter> Knöpfe aktivieren



Debian auf den aktuellen Stand bringen

Ich ermittle die IP-Adresse um mich mit ssh verbinden zu können, da dann die Kommandos über die Zwischenablage leichter eingefügt werden können. (Einfach mit der „rechten Maustaste“ bzw. wenn der Midnight -Commander läuft mit „Umschalt+rechte Maustaste“)

```
ip addr  
apt update && apt -y upgrade && apt -y autoremove && apt -y autoclean
```

Midnight-Commander und andere Tools installieren

```
apt -y install mc  
apt -y install curl tree bc zip
```

Bereits installiertes ssh für root freischalten

```
mcedit /etc/ssh/sshd_config
```

```
...  
#PermitRootLogin prohibit-password  
PermitRootLogin yes  
...
```

Installation SSH-Filesystem

```
apt -y install sshfs
```

Installation sudo

```
apt -y install sudo  
adduser user sudo
```

Grafische Oberfläche installieren

```
apt -y install xfce4
```

Automatische Anmeldung Linux einrichten

```
mc -e /usr/share/lightdm/lightdm.conf.d/01_debian.conf
```

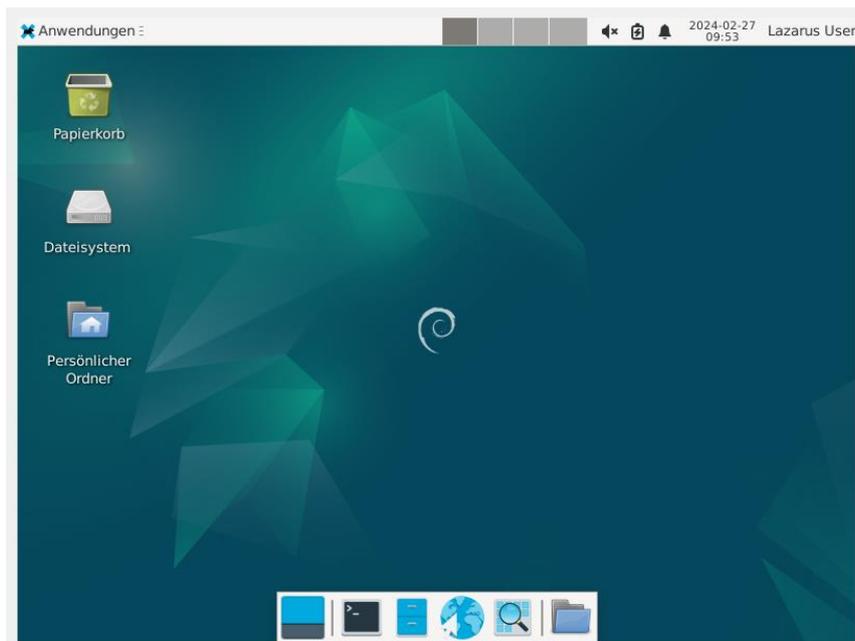
```
[SeatDefaults]  
autologin-user=user  
autologin-user-timeout=0
```

Power-Manager installieren, um Bildschirmschoner ausschalten zu können

```
apt -y install xfce4-power-manager
```

Zusätzliche Tools für XFCE

```
apt -y install synaptic  
apt -y install firefox-esr claws-mail  
reboot
```



Virtualbox Gast-CD installieren

```
apt -y install build-essential
apt -y install linux-headers-$(uname -r)
```

Gast cd einlegen über Vboxmenü

```
umount /dev/sr0
mount /dev/sr0 /media/cdrom0
/media/cdrom0/VBoxLinuxAdditions.run
```

Damit der User auch auf den gemeinsamen Ordner zugreifen kann

```
usermod -a -G vboxsf user
reboot
```

Subversion installieren

```
apt -y install git
apt -y install subversion
apt install thunar-vcs-plugin
```

Weitere Bibliotheken für Lazarus nachinstallieren

```
apt update && apt -y upgrade && apt -y autoremove && apt -y autoclean
apt -y install libx11-dev
apt -y install libgdk-pixbuf2.0-dev
apt -y install libpango1.0-dev
apt -y install libgtk2.0-dev
apt -y install libgtk-3-dev
apt install -y libxtst-dev libatk1.0-dev libghc-x11-dev freeglut3-dev
apt install -y gdb
apt -y install libqt5pas-dev
apt -y install gtk2-engines
apt -y install libsqlite3-dev
```

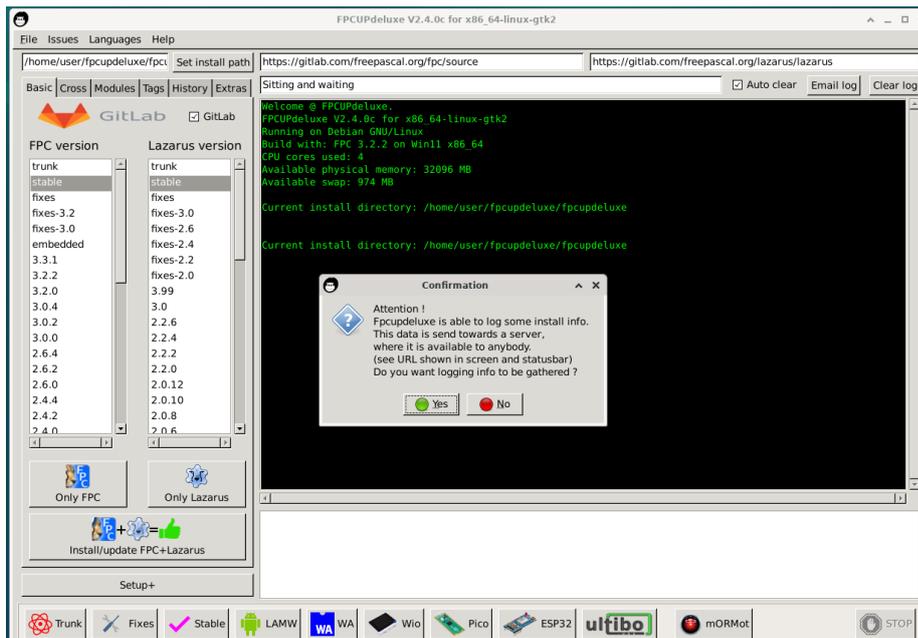
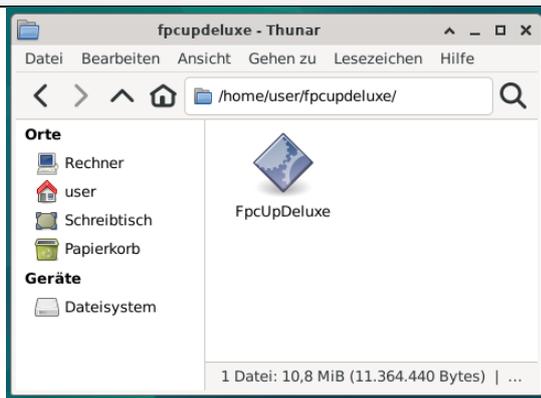
Installation Lazarus und FreePascal

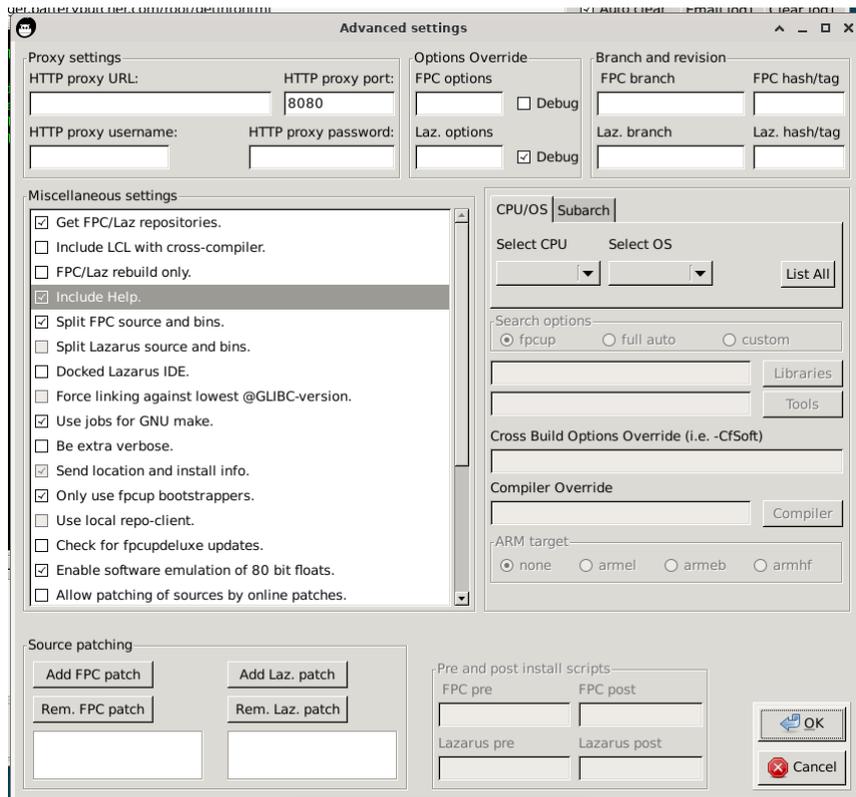
fpcupdeluxe hier downloaden: <https://github.com/newpascal/fpcupdeluxe/releases/latest>

- Immer den neuesten Download verwenden!
- Den Download von FPCUPdeluxe am besten in einen neuen leeren Ordner kopieren.
- Die kopierte Datei am besten im Datei-Manager mit einen Doppelklick starten (eventuell Ausführbar machen).

https://github.com/LongDirtyAnimAlf/fpcupdeluxe/releases/download/v2.4.0c/fpcupdeluxe-x86_64-linux

```
su - user
cd
mkdir -p ~/fpcupdeluxe
wget
https://github.com/LongDirtyAnimAlf/fpcupdeluxe/releases/download/v2.4.0c/fpcupdeluxe-x86_64-linux -O ~/fpcupdeluxe/FpcUpDeluxe
chmod a+x ~/fpcupdeluxe/FpcUpDeluxe
```





Bei „Set install path“ den gewünschten Pfad eingeben wo Lazarus installiert werden soll, der vorher angelegt wird. (/home/user/Lazarus_stable)

Um die Hilfe zu installieren, muss man auf „Setup+“ drücken und dort bei Include Help (default=no) einen Haken setzen!

Um die neueste stabile Version zu installieren einfach die anderen Einstellungen so lassen und den Button „Install/update FPC+Lazarus“ drücken.

Am Ende der Installation sollte so etwas stehen:

```

SUCCESS: installation by fpcupdeluxe complete !

Fpcupdeluxe has created a shortcut link in your home-directory to start Lazarus.
Shortcut-link: Lazarus_Lazarus_stable
Lazarus MUST be started with this link !!
Fpcupdeluxe has also (tried to) create a desktop shortcut with the same name.

```

Fpcupdeluxe schließen und Lazarus durch Anklicken des neuen Icons auf dem Schreibtisch starten.



Auf Tools/Options/General gehen und unter Language German(de) auswählen. Mit okay verlassen und Lazarus neu starten. Jetzt ist Lazarus auf Deutsch.

Installation von Android Entwicklungsumgebung

Setup Android Development Environment on Linux (FPCUPdeluxe / LAMW)

Android-Zielarchitektur: 32-bit ARM und 64-bit aarch64

Libraries und Tools installieren (als root)

```
apt install -y openjdk-17-jdk
apt install -y android-tools-adb
```

Als Benutzer anmelden

```
su - user
cd
```

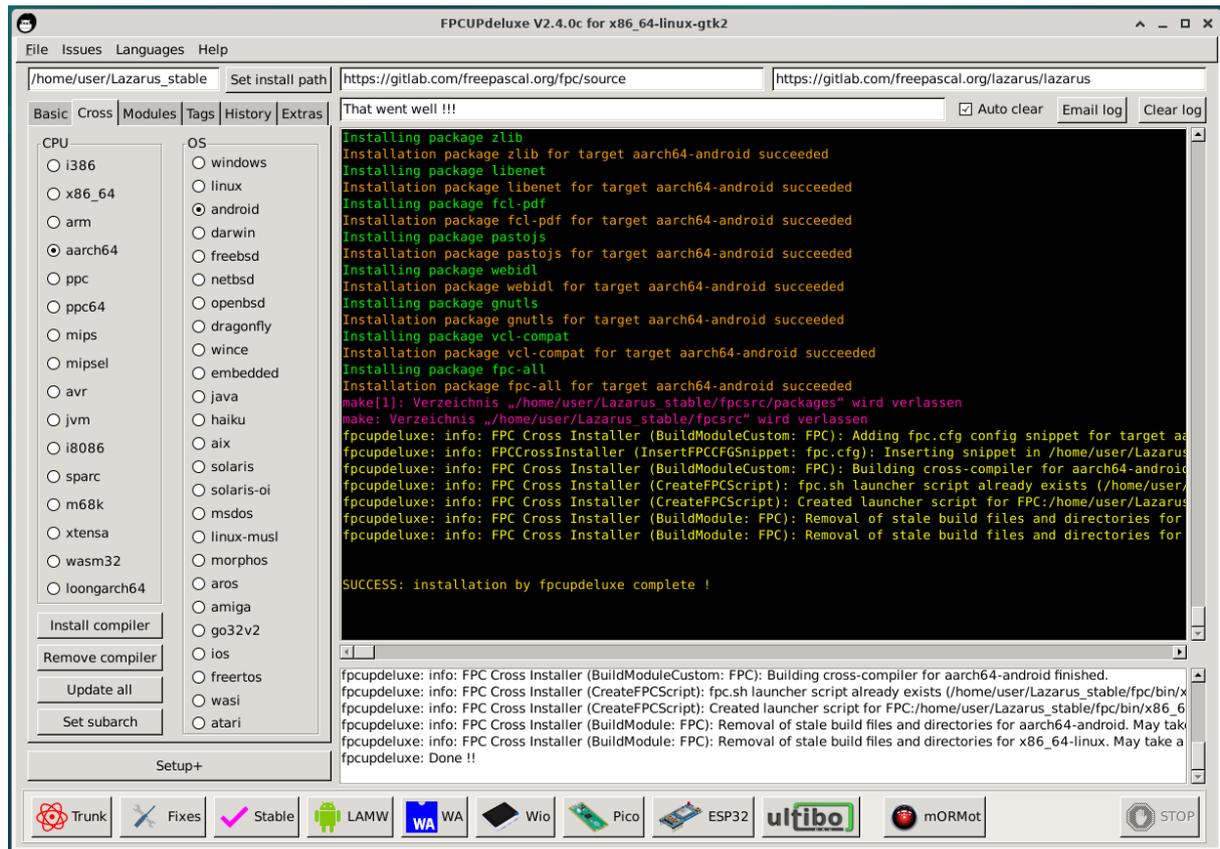
Download und entpacken der Android SDK Command Line Tools für Linux von <https://developer.android.com/studio/#downloads>

```
cd ~/Downloads
wget "https://dl.google.com/android/repository/commandlinetools-linux-11076708_latest.zip"
mkdir -p ~/Android/sdk
cd ~/Android/sdk
unzip ~/Downloads/commandlinetools-linux-11076708_latest.zip
mv cmdline-tools tools
cd ~/Android/sdk/tools/bin/
./sdkmanager --licenses --sdk_root=/home/user/Android/sdk
```

#Alle Lizenzbestimmungen mit "y" bestätigen (kommt mehrmals)!

```
mkdir ~/Android/sdk/ndk-bundle
./sdkmanager "build-tools;34.0.0" "platforms;android-34" "tools" "ndk-bundle" "extras;android;m2repository" --
sdk_root=/home/user/Android/sdk
```

Installation CrossCompiler



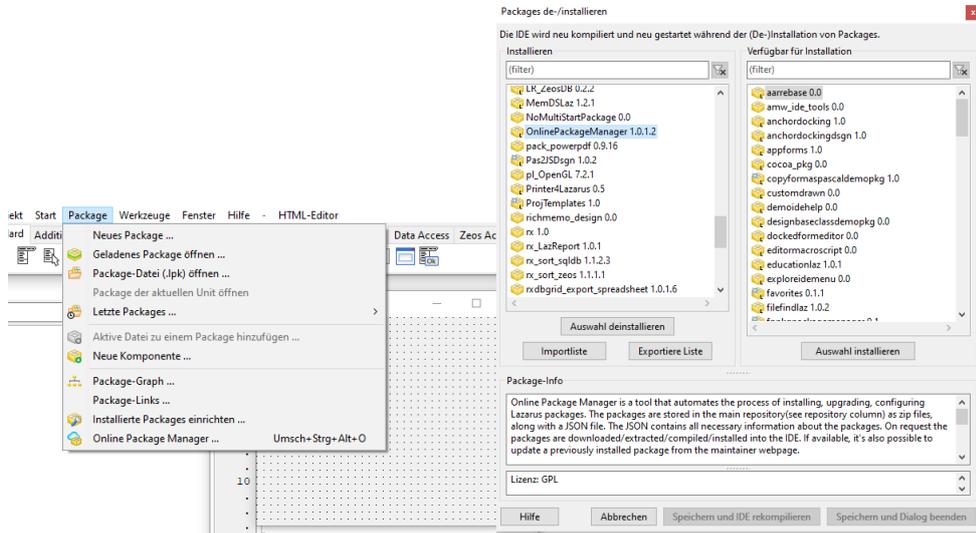
Folgende Konstellationen CPU /OS müssen installiert werden:

- i386 / android
- Arm / android
- Aarch64 / android

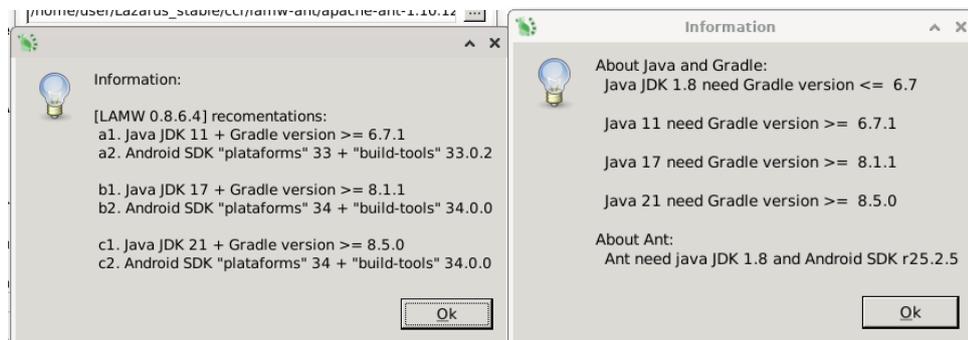
Man kann auch noch andere Cross-Compiler gleich mit installieren, wenn man das möchte (z.B. für Windows).

Installation LAMW

Danach erfolgt die Installation von LAMW. Das kann über FPUUpDeluxe oder mit dem Onlinepaketmanager im Lazarus geschehen. Mit FPUUpDeluxe hatte ich Probleme, weil die Versionen irgendwie nicht zueinander passten. Daher nahm ich den Onlinepaketmanager. Sollte dieser nicht im Menü vorhanden sein, dann wird der über den internen Paket-Manager einrichten/aktivieren.



Zum Erstellen der APK-Pakete gibt es zwei Programme (Ant und Gradle). Hier muss man auf die Versionen achten!



Da wir die Java JDK Version 17 installiert haben, wird Ant wahrscheinlich nicht funktionieren. Ich habe es trotzdem installiert.

Download (<https://ant.apache.org/bindownload.cgi>) und entpacken in /home/user/Lazarus_stable/ccr/lamw-ant/.

Der Download von Gradle (<https://gradle.org/releases/>) wird in das Verzeichnis /home/user/Lazarus_stable/ccr/lamw-gradle entpackt.

Einstellen der Pfade

Die Pfade sind jetzt:

Java JDK: /usr/lib/jvm/java-17-openjdk-amd64

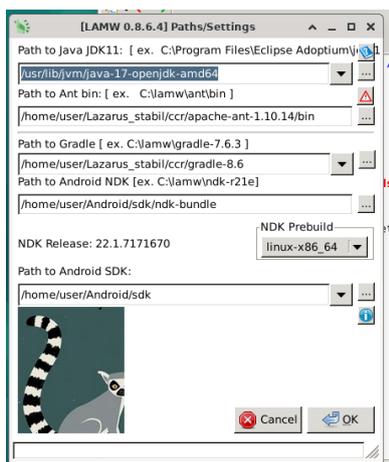
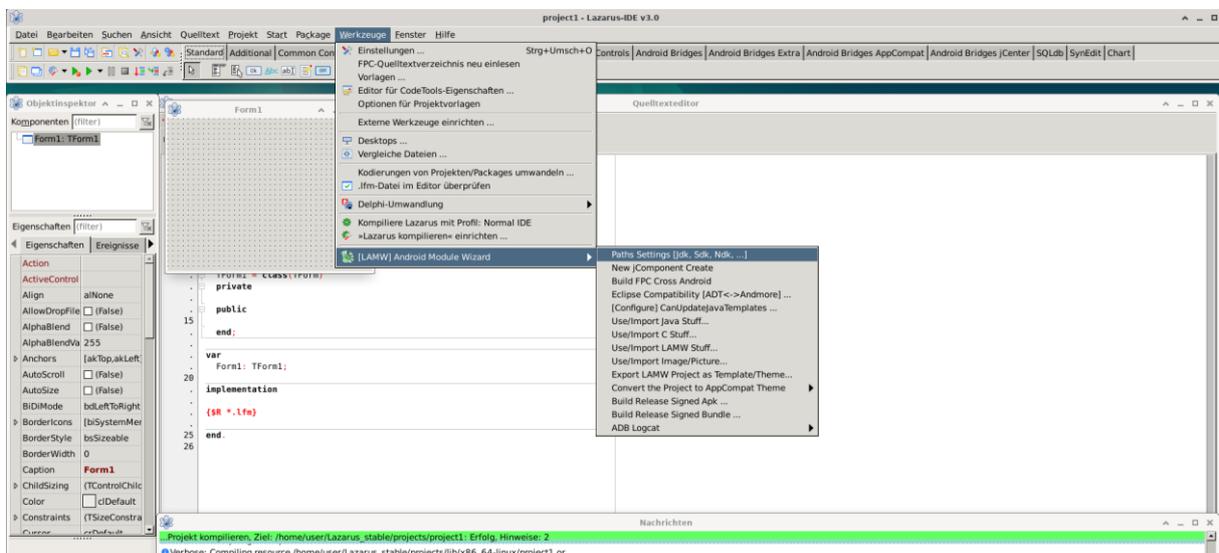
Ant: /home/user/Lazarus_stable/ccr/lamw-ant/apache-ant-1.10.12/bin

Gradle: /home/user/Lazarus_stable/ccr/lamw-gradle/gradle-6.8.3

Android NDK: /home/user/Android/sdk/ndk-bundle

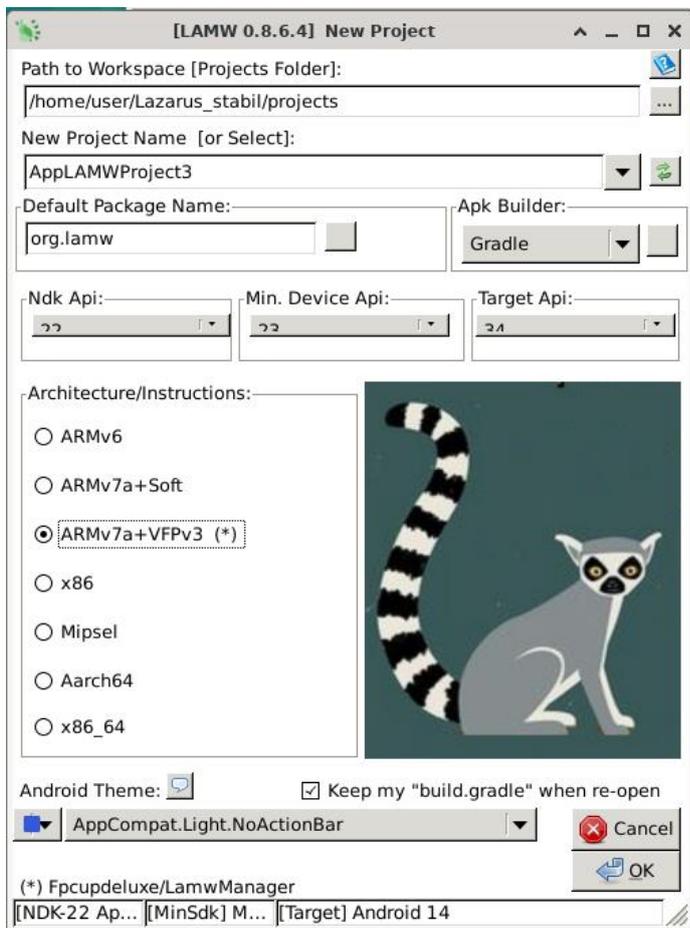
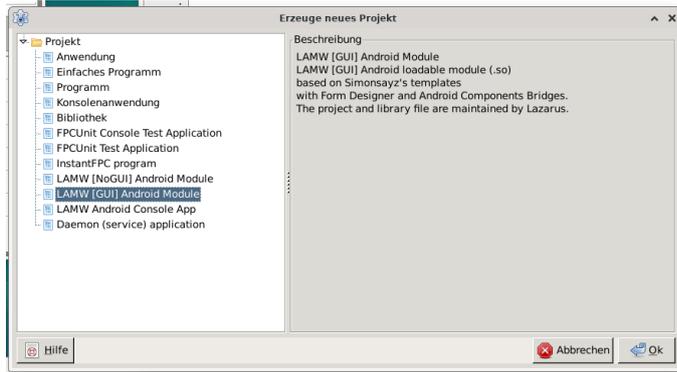
Android SDK: /home/user/Android/sdk

Im Lazarus müssen nun die richtigen Pfade eingestellt werden:



Erstes Projekt erstellen

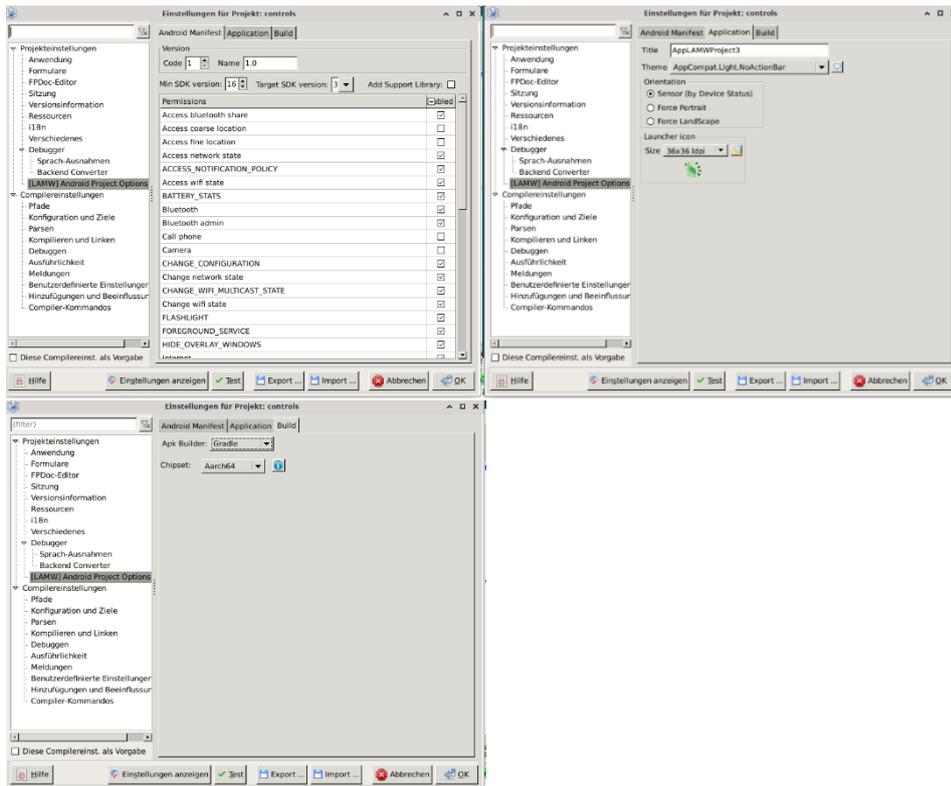
Damit ist die Installation fertig. Ein erstes Projekt erstellt man über Projekt → neu:



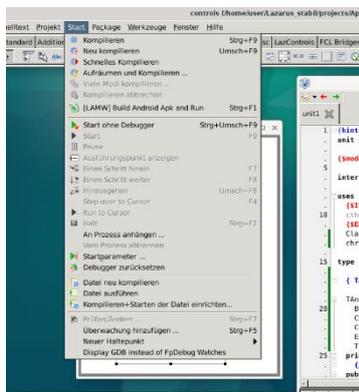
Es funktionierten für 32-Bit Arm-Prozessoren der Eintrag „ARMv7a-VFPv3“ und 64-Bit der „Aarch64“.

Dann erscheint das Form als Modul, auf den dann die Komponenten gezogen werden. Es werden nicht die normalen Komponenten, sondern die Komponenten aus „Android Bridges“, „Android Bridges Extra“, „Android Bridges AppCompat“ und „Android Bridges jCenter“ verwendet!

Projekteinstellungen, wie Rechtevergabe, Ausrichtung, Icon und APK Builder über „[LAMW] Android Project Options“ in den Projekteinstellungen.



Apk-Datei erzeugen über „Start“ → „[LAMW] Build Android Apk und Run“



Die apk-Datei befindet sich dann unter

~/Lazarus_stabil/projects/AppLAMWProject3/build/outputs/apk/debug

Die Apk-Datei wird auf das Smartphone kopiert und dann installiert.

Bei der Installation müssen drei Dinge geachtet werden:

- Das Smartphone muss zulassen, dass APK-Dateien aus fremden Quellen installiert werden dürfen.
- Wenn eine Sicherheits-Meldung kommen sollte, dann muss man unter den Details „Trotzdem installieren“
- Die notwendigen Berechtigungen müssen für die App freigegeben sein. Am Besten ist es, die App einmal zu starten und danach sollte diese bei „Zugriff verweigert“ in der Liste erscheinen. Hier kann man das dann freigeben.

Wie das bei den jeweiligen Smartphones funktioniert ist von Hersteller und OS-Version unterschiedlich und kann deshalb nicht genau beschrieben werden.