MULTIBOOT-DVD!
NEU: Linux Mint 20
PLUS: 5 weitere Top-Systeme
• Ubuntu Mate 20.04
• Manjaro 20 LXQT
• Fedora 32
• Rescatux 0.73
EXKLUSIV & NEU
LinuxWelt-Surfsystem 5/20

Multiboot-DVD mit 6 Top-Systemen

5/2020
August / September

4 196691 608505 05

Deutschland 8,50 €
Schweiz 16,90 sfr · Österreich + Benelux 9,45 €

LINUX WELT

Einsteigertipps
Virenschutz für Linux und Windows

Linux immer im Griff

Mit diesen Tipps läuft Ihr System reibungslos

- Linux ohne Wartezeit starten
- Systemanmeldung automatisieren
- SSD als Datenturbo nutzen
- Direkte Verknüpfungen zu jeder Aufgabe erstellen

u.v.m.

Heimnetz absichern!

Mit IP-Fire schützen Sie Ihr komplettes Netzwerk

Desktop-Geheimtipp

XFCE: Deutlich schneller als Gnome & Co.

Mediacenter im Eigenbau

Mit Kodi alle Medien wiedergeben

Nie mehr Terminal-Befehle!

Mit diesen Programmen lösen Sie auch schwierige Aufgaben per Mausklick

NEU: Linux Mint 20

Neue Themes, verschlüsselte Datenübertragung, verbesserte Monitorskalierung, überarbeitete Statusleiste …

MULTIBOOT-DVD!
NEU: Linux Mint 20
PLUS: 5 weitere Top-Systeme
• Ubuntu Mate 20.04
• Manjaro 20 LXQT
• Fedora 32
• Rescatux 0.73
EXKLUSIV & NEU
LinuxWelt-Surfsystem 5/20

Neu & aktualisiert:

50 Handbücher für Ihr Linux

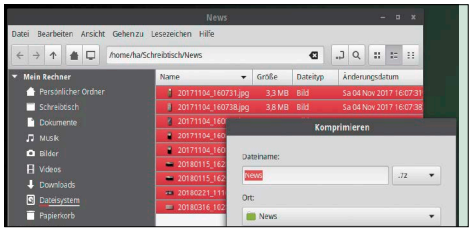
7800 Seiten neues Linux-Know-how für Mint, Ubuntu, Netzwerk, Libre Office, Bash u.v.m.

Infotainment
Datenträger enthält nur Lehr- oder Infoprogramme

Linux immer im Griff

Diese Tricks steigern das Tempo am Linux-Desktop und passen das System optimal an: So starten Sie schneller, senken die Festplattenaktivität und beschleunigen Programme.

S. 28



Terminal vs. Tool

Mit diesen Desktoptools erledigen Sie wichtigen Aufgaben per Mausklick und kommen fast ohne Terminal aus.

S. 42

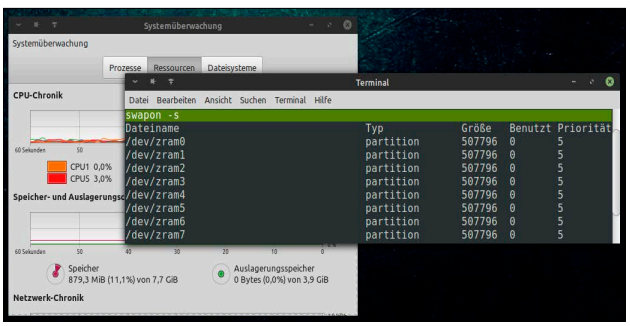
Kodi für Profis

Praxistipps für Kodi: So nutzen Sie alle Funktionen der Medienzentrale.

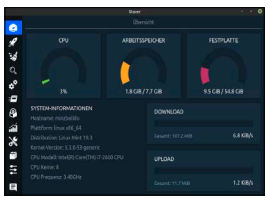
S. 58

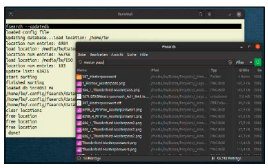
Grundlagen

Special I – Linux immer im Griff



Special II – Produktive Desktoptools





Die Highlights der DVD

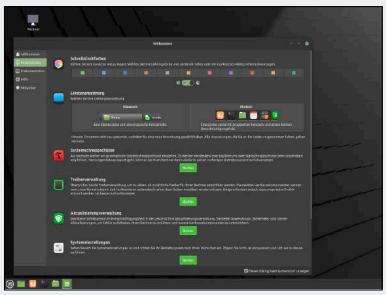
Auf Heft-DVD: 4 x Linux-Desktop plus praktische Livesysteme

Neben den drei unten beschriebenen Desktopdistributionen startet die Heft-DVD den Installer des aktuellen Fedora 32 mit freier Oberflächenwahl. Das LinuxWelt-Surfsystem 5/20 ist eine sichere Livelösung für den Internetbesuch und Rescatux 0.73 repariert defekte Grub-Bootumgebungen.

S. 10

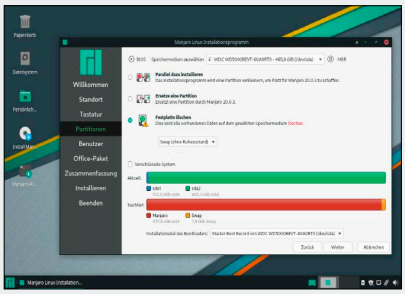
Linux Mint 20 (Cinnamon)

Die Hauptedition mit Cinnamon ist das Flaggschiff der Mint-Trilogie. Cinnamon bringt einige Neuheiten wie die „Willkommen"-Themenwahl



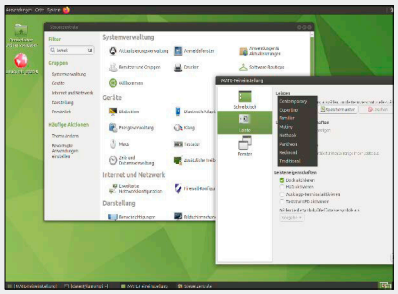
Manjaro 20 (LXQT)

Manjaro ist das einzige Arch-Derivat, das am Desktop dauerhaft punkten kann. Von der Heft-DVD startet Manjaro in der LXQT-Variante.



Ubuntu Mate 20.04 LTS

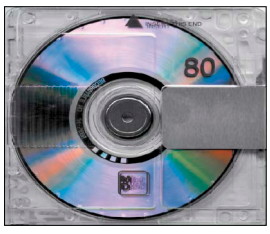
Die Mate-Edition behauptet sich im Ubuntu-Zoo als unprätentiöse Variante neben eigenwilligeren Desktops wie Gnome, KDE oder Budgie.



Software

Netzwerk & Hardware

Praxis



Standards

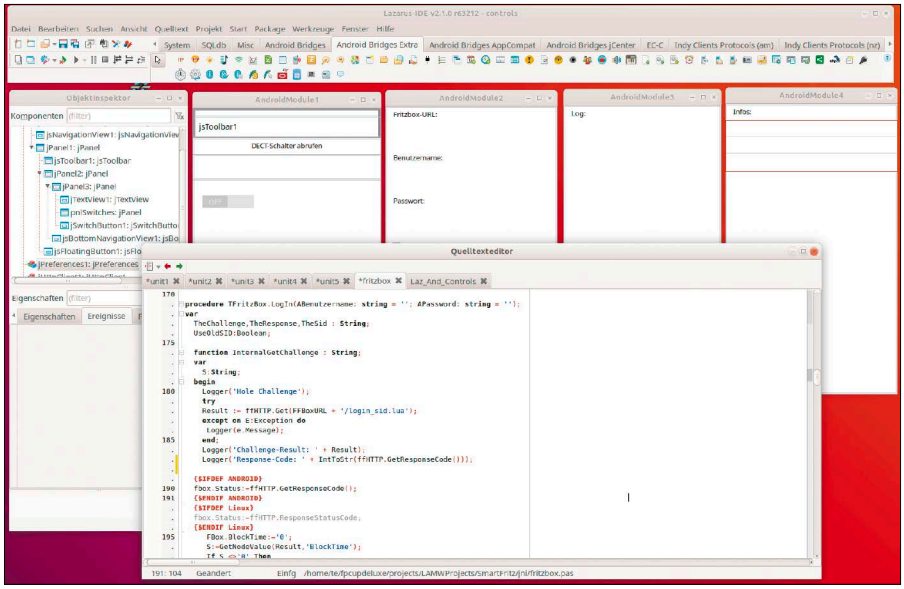
Android-Apps programmieren

Software für Android lässt sich in vielen Programmiersprachen entwickeln. Mit Lazarus und Free Pascal gelingt das vergleichsweise einfach, auch wenn die Installation der nötigen Tools zunächst einige Vorbereitungen erfordert.

VON THORSTEN EGGELING

Die bevorzugten Programmiersprachen für Android-Apps sind Java und Kotlin. Beide werden von Googles offizieller Entwicklungsumgebung Android Studio unterstützt. In diesem Artikel soll es jedoch um den Free Pascal Compiler (FPC) gehen. Zusammen mit der Entwicklungsumgebung Lazarus und einigen Erweiterungen lassen sich Android-Apps relativ komfortabel erstellen. Wir erläutern die nötigen Schritte am Beispiel eines Projekts, mit dem sich eine Fritzbox steuern lässt.

Service: Den Quellcode, die fertig kompilierten Beispielprogramme und weitere Informationen finden Sie über *https://m6u.de/lamw*.



IDE für Linux und Android: Wer Programme in Pascal entwickeln möchte, findet mit Lazarus eine optisch etwas angestaubte, aber technisch aktuelle und funktionsreiche Lösung.

1. Das leisten Lazarus und FPC

Free Pascal (FPC, *www.freepascal.org*) ist der Open-Source-Nachfolger von Turbo Pascal und weitestgehend mit Delphi kompatibel (früher Borland, heute Embarcadero). Pascal ist klar strukturiert, Variablen müssen immer deklariert werden. Das erleichtert Programmieranfängern, Fehler zu vermeiden.

Lazarus (*www.lazarus-ide.org*) ist die grafische Oberfläche (IDE, Integrated Development Environment) für FPC. Ein Kernbestandteil ist der grafische Formulardesigner, über den Sie Bedienelemente im Programmfenster unterbringen und das Layout gestalten. Vom Formulardesigner gelangen Sie schnell zum Quelltexteditor, etwa per Doppelklick auf eine Schaltfläche im Formular. Hier bringen Sie den Programmcode unter, der etwa beim Klick auf eine Schaltfläche ausgeführt werden soll. Lazarus kann aus dem gleichen Quellcode Binärdateien für mehrere Betriebssysteme erstellen, neben Linux auch für BSD, Windows, Mac-OS und den Raspberry Pi. Die Programme laufen in der Regel ohne Abhängigkeiten, unter Linux/BSD müssen jedoch die GTK2-Bibliotheken installiert sein, was aber bei vielen Systemen standardmäßig der Fall ist.

Ein Lazarus-Projekt lässt sich für Android nicht ganz so einfach umsetzen, da nicht die gleichen Bedienelemente verfügbar sind. Am Pascal-Code muss man jedoch kaum etwas ändern, wenn man Programmlogik und Verweise auf die grafische Oberfläche so weit wie möglich trennt. Ein Nachteil: Android-Apps lassen sich in Lazarus nicht schrittweise debuggen, wodurch Fehler schwerer zu ermitteln sind.

Formulardesigner: Grafische Oberflächen lassen sich unter Lazarus mit ein paar Mausklicks erstellen. Für Android müssen Sie die Elemente jedoch manuell ausrichten.

2. Lazarus/FPC unter Linux einrichten

Für die komfortable Installation verwenden Sie das Tool Fpcupdeluxe, das Sie über *https://m6u.de/fpcup* herunterladen. Für 64-Bit-Linux wie Ubuntu 18.04, 20.04 oder Linux Mint 19.3 laden Sie die Datei „fpcupdeluxe-x86_64-linux“ herunter.

Schritt 1: Installieren Sie zuerst die folgenden Pakete:

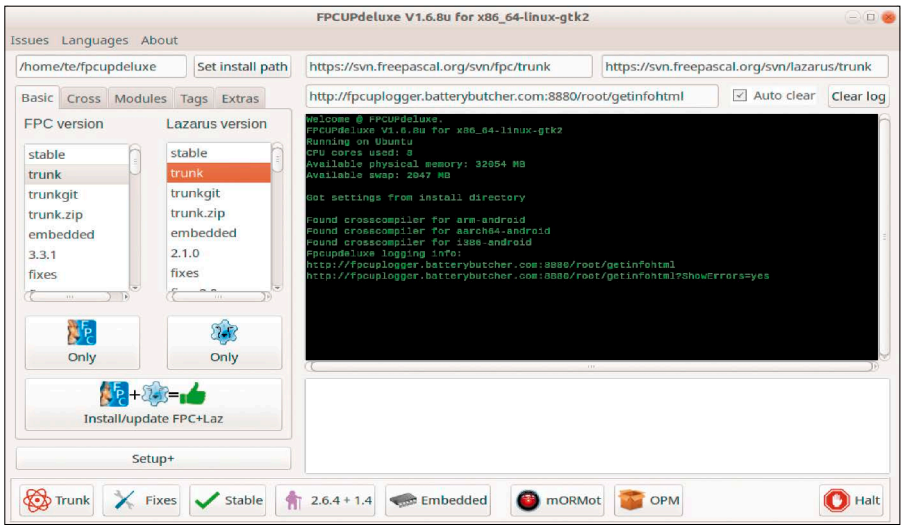
```
sudo apt install libx11-dev libgdk-
  pixbuf2.0-dev libpango1.0-dev
  libgtk2.0-dev subversion git
  freeglut3-dev
```

Schritt 2: Starten Sie dann „fpcupdeluxe-x86_64-linux“. Standardmäßig lassen sich darüber per Klick auf „Stable“ die Versionen FPC 3.0.4 und Lazarus 2.0.8 installieren. Das ist ausreichend, wenn Sie 32-Bit-Apps für Android erstellen möchten. Die laufen auf so gut wie allen Android-Geräten. Wer auch 64-Bit-Apps benötigt, klickt auf „Trunk“ (neuere Versionen, noch in der Entwicklung).

Das ist erforderlich, wenn Sie Apps bei Google Play veröffentlichen wollen. Wir haben für unsere Beispiel-App die Trunk-Version verwendet.

Schritt 3: Nach der Installation starten Sie Lazarus probeweise über das Desktopicon. Da wir noch weitere Tools benötigen, beenden Sie Lazarus und kehren zu Fpcupdeluxe zurück.

Schritt 4: Gehen Sie auf die Registerkarte „Cross“, aktivieren Sie die Optionen „arm“ und „android“ und klicken Sie auf „Install compiler“. Für die 64-Bit-Version aktivieren Sie „aarch64“ und installieren auch diesen Compiler.



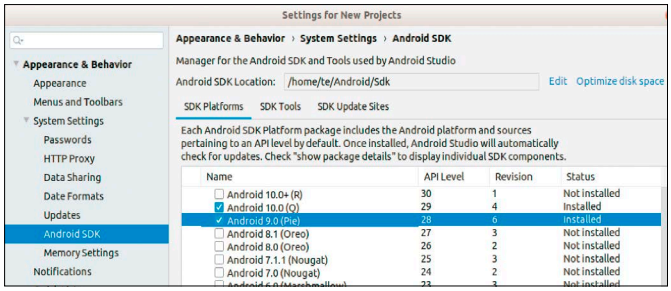
Einfache Installation: Das Tool Fpcupdeluxe hilft bei der Einrichtung unterschiedlicher Lazarus/FPC-Versionen und kann auch die für Android nötigen Crosscompiler installieren.

Schritt 5: Wechseln Sie auf die Registerkarte „Modules“. Wählen Sie nacheinander „ECCControls“, „hashlib4pascal“ und „lamw“ und klicken Sie jeweils auf „Install module“.

3. Android-Tools installieren

Laden Sie Android Studio für Linux über *https://developer.android.com/studio* herunter. Entpacken Sie die Datei und verschieben Sie diese in den Ordner „android-studio“ in Ihr Home-Verzeichnis. Starten Sie „~/android-studio/studio.sh“. Folgen Sie den Anweisungen des Installationsassistenten. Sie können alle Voreinstellungen unverändert übernehmen.

Gehen Sie im Fenster „Welcome to Android Studio“ unten rechts auf „Configure → SDK



Tools für Android-Entwickler: Die gewünschten Programme und Bibliotheken richten Sie über Android Studio ein. Sie benötigen die Entwicklungsumgebung auch für den Android-Emulator.

ANDROID-APPS STARTEN UND TESTEN

Über Lazarus lassen sich Apps an einen Emulator oder ein per USB angeschlossenes Android-Gerät übertragen und starten: Auf dem Android-Gerät aktivieren Sie dafür die Entwickleroptionen. Dafür gehen Sie in den Einstellungen auf „Telefoninfo“ und tippen schnell hintereinander bis zu zehnmal auf „Build-Nummer“, bis die “Entwickleroptionen“ angezeigt werden. Bei neueren Geräten führt der Weg beispielsweise über „Mein Gerät → MIUI-Version“. Die „ Entwickleroptionen“ befinden sich hier meist unter „Weitere Einstellungen“. Aktivieren Sie „USB-Debugging“ und – wenn vorhanden – „Installieren über USB“ und „USB-Debugging (Sicherheitseinstellungen)“. Die verfügbaren Optionen und Beschriftungen können je nach Gerät abweichen. Einen Emulator richten Sie über Android Studio ein. Gehen Sie im Startbildschirm auf „Configure → AVD Manager“ und klicken Sie auf „+ Create Virtual Device“. Übernehmen Sie die Vorauswahl „Pixel 2“. Klicken Sie auf „Next“ und klicken Sie beim Systemimage, „Pie“ (API-Level 28) auf „Download“. LAMW unterstützt zur Zeit nur die API-Level bis 28. Starten Sie den Emulator über die „Play“-Schaltfläche.

Erstellen Sie die App in Lazarus über „Start → [LAMW] Build Android Apk and Run“ oder Strg-F1. Im Emulator ist der Internetzugang möglich, nicht aber der Zugriff auf das lokale Netzwerk. Deswegen lässt sich unsere Beispiel-App hier nur unzureichend testen. Wer den Emulator trotzdem verwenden möchte, kann unter Linux eine IP-Umleitung einrichten. Wie das geht, lesen Sie unter *https://m6u.de/lamw* nach.

Manager“ und aktivieren Sie „Android 9.0 (Pie)“. Wechseln Sie dann auf die Registerkarte „SDK Tools“ und aktivieren Sie „NDK (Side by side)“, „Android SDK Commandline Tools (latest)“ sowie „Google Play services“: Nach Klick auf „OK“ folgen Sie den weiteren Anweisungen des Assistenten.

4. Ein erstes Android-Projekt erstellen

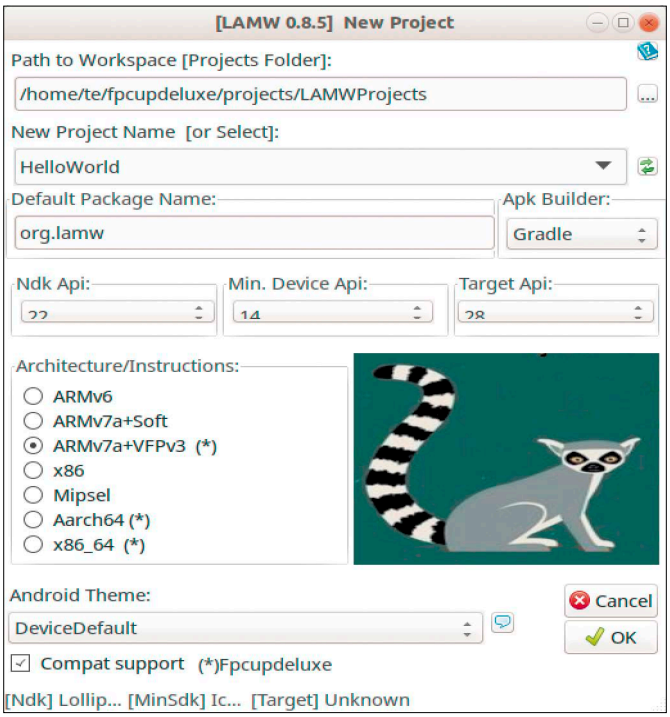
Starten Sie Lazarus über das Desktopicon. Das Programm startet standardmäßig mit einen neuen Projekt und zeigt ein leeres Formular und den Quelltexteditor. Gehen Sie auf „Tool → Options“ und dann auf „Environment → General“. Stellen Sie als Sprache „Deutsch [de]“ ein, dann beenden Sie Lazarus über „File → Quit“ und starten das Programm neu.

Schritt 1: Gehen Sie auf „Werkzeuge → [LAMW] Android Module Wizard → Path settings“. Unter „Path to Java JDK“ geben Sie „/home/[User]/android-studio/jre“ aus Ihrem Home-Verzeichnis an. Unter „Path to Android NDK“ gehört zur Zeit der Pfad „/home/[User]/Android/Sdk/ndk/21.2.6472646“. Darunter wählen Sie die Option „>11“. Klicken Sie auf „OK“, um die Änderungen zu speichern. Die Einstellungen gelten für alle Projekte und müssen nur einmal angepasst werden.

Schritt 2: Klicken Sie im Menü auf „Projekt → Neues Projekt“. Wählen Sie „LAMW [GUI] Android Module“ und klicken Sie auf „OK“. Tippen Sie unter „New Project Name“ eine aussagekräftige Bezeichnung ein, beispielsweise „HelloWorld“. Unter „Default Package Name“ können Sie „org.lamw“ belassen. Wenn Sie die App veröffentlichen wollen, müssen Sie eine eigene Bezeichnung angeben (siehe *https://developer.android.com/studio/build/application-id*). Die weiteren Einstellungen sehen Sie in der Abbildung auf dieser Seite.

Schritt 3: Klicken Sie auf „OK“ und danach auf „Speichern“. Sie sehen jetzt das Fenster des Formdesigners mit dem Titel „AndroidModule“ sowie den Quelltexteditor und den Objektinspektor.

Schritt 4: In der Lazarus-Hauptleiste stehen die Komponenten auf den Registerkarten zur Verfügung, die ein „Android“ in der Bezeichnung tragen. Gehen Sie auf „Android Bridges“, klicken Sie auf das Icon mit dem Button („jButton“) und klicken Sie in das Fenster des Formdesigners. Bauen Sie außerdem ein Textfeld ein („jEditText“).



Neues Android-Projekt: Ändern Sie die Einstellungen wie abgebildet. Lazarus/LAMW verwenden bei aktiviertem „Compat support“ API 28 (Android 9).

Die Elemente lassen sich auf dem Formular beliebig positionieren.

Schritt 5: Per Doppelklick auf „jButton1“ erzeugen Sie das zugehörige Ereignis „jButton1Click“ im Editor. Zwischen „begin“ und „end“ fügen Sie die Zeile

```
jEditText1.Text:='Hello World';
```

ein. Mit Strg-F9 („Start → Kompilieren“) erstellen Sie das Programm und mit Strg-F1 („Start → [LAMW] Build Android Apk and Run“) erzeugen Sie das Installationspaket („apk“). Das Programm wird automatisch auf den laufenden Emulator oder ein angeschlossenes Android-Gerät übertragen und gestartet (siehe Kasten „ Android-Apps starten und testen“).

Schritt 6: Unter Android öffnet sich die App mit einer Titelleiste, die den Namen der App zeigt („HelloWorld“). Darunter ist die Schaltfläche zu sehen, das Textfeld aber nicht. Daher hat es auch keine sichtbare Auswirkung, wenn Sie die Schaltfläche antippen. Der Grund dafür liegt in der Besonderheiten der Android-Apps. Wegen der unterschiedlichen Bildschirmgrößen sowie Hoch- und Querformat richten sich die Bedienelemente relativ zueinander aus. Angaben dazu, wie das geschehen soll, fehlen bisher in der App. Deswegen liegen alle Elemente übereinander, die Schaltfläche verdeckt das Textfeld. Wie die Elemente in der Entwurfsansicht angeordnet sind, spielt keine Rolle. Um das zu ändern, klicken Sie „jEditText1“ an. Im Objektinspektor gehen Sie auf „PosRelativeToParent“ und setzen Häkchen vor „rpLeft“ und „rpTop“. Klicken Sie die Schaltfläche „jButton1“ an, wählen Sie im Objektinspektor hinter „Anchor“ den Eintrag „jEditText1“, bei „PosRelativeToAnchor“ setzen Sie ein Häkchen vor „raBelow“. Hinter „PosRelativeToParent“ aktivieren sie „rpLeft“. Bei beiden Elementen stellen Sie hinter „LayoutParamWidth“ den Wert „lpMatchParent“ ein.

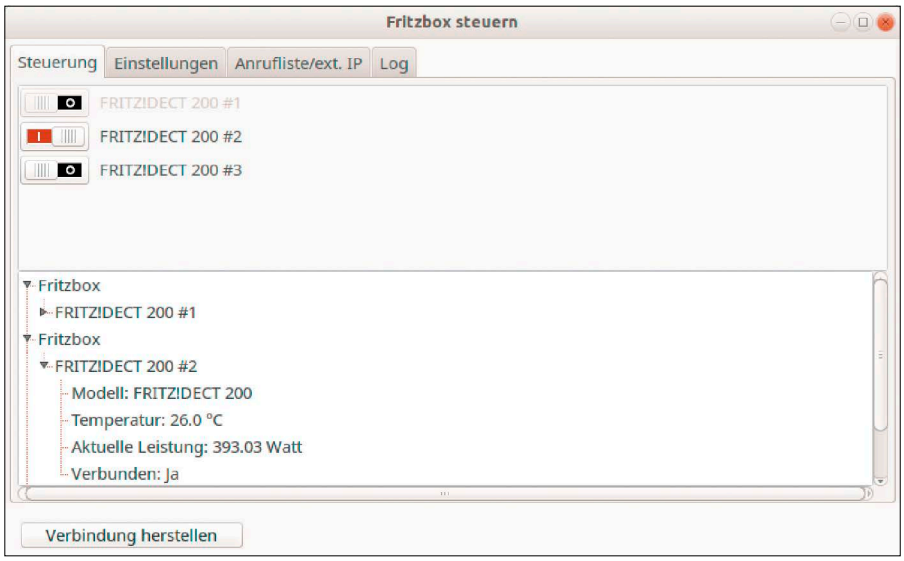
„jEditText1“ befindet sich im Formdesigner jetzt links oben, „jButton1“ wird im Containerelement (Parent, AndroidModule1) an der linken Seite ausgerichtet und unterhalb des Anchor-Elements „jEditText1“. Wenn Sie die App mit Strg-F1 neu starten, entspricht das Ergebnis der Entwurfsansicht. Ein Klick auf die Schaltfläche ersetzt den Text in „jEditText1“ durch „Hello World“. Durch „lpMatchParent“ füllen beide Elemente die gesamte Breite des Bildschirms aus. Das gilt auch, wenn Sie sich das Smartphone in den Modus „Querformat“ drehen. Die Ausrichtung der Elemente ist eine Android-Eigenart und in Android Studio auch nicht besser gelöst. Hinzu kommt, dass Android mehrere Layoutansichten verwenden kann und es eigene Controls etwa für die App-Leiste, eine Toolbar oder die Tab-Navigation gibt. Eine grundsätzliche Einführung liefert *https://developer.android.com/design*.

5. Demo-Apps ausprobieren

Für den Lazarus Android Module Wizard (LAMW) ist nicht viel Dokumentation verfügbar. Bei Problemen und Fragen ist das Forum *https://forum.lazarus.freepascal.org* eine gute Anlaufstelle. Infos zu Android finden Sie nach Klicks auf „Programming“, „Operating Systems“ und „Android“.

Hilfreich sind die Demo-Apps, die Sie in Ihrem Home-Verzeichnis unter „fpcupdeluxe/ccr/lamw/demos“ finden. Die Projektdatei finden Sie jeweils im Ordner „jni/controls.lpi“. Gehen Sie zuerst auf „Projekt → Projekteinstellungen“ und dann unter „Projekteinstellungen“ auf „[LAMW] Android Project Options“. Hinter „Target SDK version“ stellen Sie den Wert „28“ ein, auf der Registerkarte „Build“ wählen Sie „Gradle“ und „ARMv7a+VFPv3“. Danach kompilieren Sie die App mit Shift-F9 („Neu kompilieren“) und starten sie mit Strg-F1.

Hinweis: Strg-F1 funktioniert manchmal nicht zuverlässig. Öffnen Sie dann nach Shift-F9 oder Strg-F9 („Kompilieren“) den Ordner der App in einem Terminal und starten Sie „gradle-local-build.sh“ und danach „gradle-local-run.sh“.



Linux zuerst: Unter Linux lässt sich ein Programm ausprobieren und schrittweise debuggen. Die meisten Funktionen werden dann auch in der Android-App keine Probleme bereiten.

6. Linux-Programm als Grundlage verwenden

Es empfiehlt sich, vor der Android-App zuerst ein Linux-Programm mit den gleichen Funktionen zu erstellen. Das Programm lässt sich dann Schritt für Schritt debuggen, was die Suche nach Fehlern erleichtert. Den Quelltext für unser Linux-Beispielprojekt „SmartFritzSchalter“ können Sie über *https://m6u.de/lamw* herunterladen. Er ist ausführlich kommentiert, weshalb wir hier darauf nicht weiter eingehen. Die Datei „fritzbox.pas“ enthält alle Aufrufe, die die Fritzbox betreffen.

Testen lässt sich das Projekt mit allen Funktionen nur, wenn Sie eine Fritzbox und einen Smarthome-Schalter besitzen („Fritz!DECT 200“). Das Programm kann aber als Beispiel für weitere Funktionen auch die Anruferliste anzeigen und die externe IP-Adresse der Fritzbox melden.

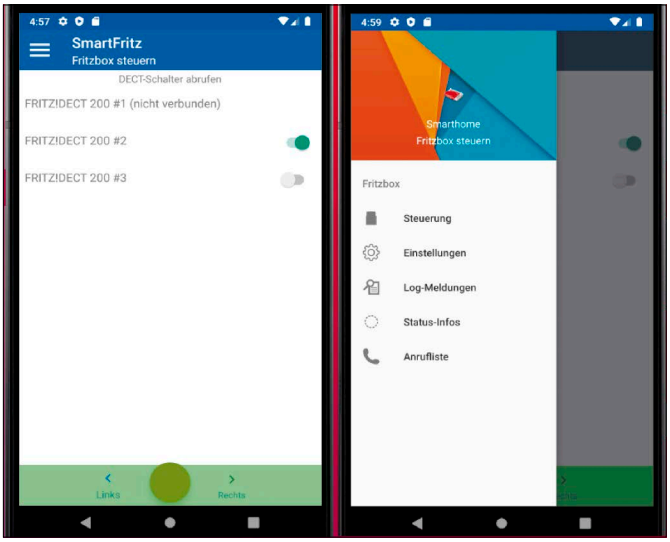
Hinweis: Standardmäßig fordert eine Fritzbox beim Aufruf im lokalen Netzwerk über *http://192.168.178.1* oder *http://fritz.box* nur ein Passwort an.

Erfolgt der Zugang über das Internet und eine HTTPS-verschlüsselte Verbindung, müssen Benutzername und Passwort konfiguriert sein. Das selbst signierte SSL-Standardzertifikat der Fritzbox wird vom Linux-Programm akzeptiert, nicht jedoch von der Android-App. Wie Sie eine Ausnahme definieren, lesen Sie unter *https://m6u.de/lamw*.

7. Die Android-App zur Fritzbox-Steuerung

Die Android-App mit dem Namen „SmartFritz“ enthält fast die gleichen Funktionen wie das Linux-Programm. Um die Unterschiede zu verdeutlichen, haben wir die Codeteile für Android zwischen „{$IFDEF ANDROID}...{$ENDIF ANDROID}“ und gepackt und die für Linux zwischen „{$IFDEF Linux}...{$ENDIF Linux}“. In der Regel betrifft das Anweisungen, die sich auf Bedienelemente beziehen, die bei Android anders heißen und andere Optionen bieten. Die App besteht aus mehreren Modulen, im Android-Sprachgebrauch „Activities“ genannt. Die Start-Acitivity „AndroidModule1“ zeigt nach der Anmeldung die Schalter. Diese werden dynamisch erzeugt, je nachdem, wie viele vorhanden sind. Die anderen Module dienen für die Eingabe der Anmeldeinformationen, Ausgabe von Logeinträgen, Infos zu den Schaltern (Leistung, Temperatur) und für die Anruferliste.

Die Navigation erfolgt über ein Menü („jsNavigationView1“), eine Leiste am unteren Rand („jsBottomNavigationView1“) oder Fingergesten („jPanel2FlingGesture“). Die Anmeldung erfolgt über die runde rote Schaltfläche, die nach erfolgreichem Log-in die Farbe ändert. ■



Fritzbox steuern: Die App zeigt die verfügbaren Fritz-DECT-Geräte an, die sich auch umschalten lassen. Zu den Einstellungen und Logmeldungen gelangen Sie auch über ein Menü.