

# REVOLUTION PI

Release Notes  
Bullseye 06/2023

30.06.2023

# REVOLUTION PI

**KUNBUS GmbH**

Heerweg 15 C

73770 Denkendorf

Germany

+49 (0)711 400 91 500

# 1 Release Notes Bullseye 06/2023

## Raspberry Pi OS Bullseye

Unser Image basiert auf Raspberry Pi OS 2023-05-03, welches wiederum auf Debian Bullseye basiert. Zuvor basierten unsere Images noch auf Debian Buster. Entsprechend ändern sich einige grundlegende Funktionen des Betriebssystems (z. B. grafische Oberfläche und Netzwerkkonfiguration). Details über die Änderungen an Raspberry Pi OS kannst Du diesen Release Notes entnehmen:

- [https://downloads.raspberrypi.org/raspios\\_armhf/release\\_notes.txt](https://downloads.raspberrypi.org/raspios_armhf/release_notes.txt) für Bullseye armhf
- [https://downloads.raspberrypi.org/raspios\\_arm64/release\\_notes.txt](https://downloads.raspberrypi.org/raspios_arm64/release_notes.txt) für Bullseye arm64

An Raspberry Pi OS haben wir Anpassungen vorgenommen: Unter anderem haben wir unnötige Pakete entfernt, APT-Repositories nachgerüstet und Pakete von KUNBUS installiert. Details sind hier zu finden:

<https://github.com/RevolutionPi/imagebakery>

Das Image enthält alle Paket-Updates, die bis Mai 2023 von der Raspberry Pi Foundation und bis Juni 2023 vom Revolution Pi Projekt veröffentlicht wurden. Es verwendet wie bisher Kernel 5.10.152 mit dem RT Patch Version 75.

## Unterstützung für arm64-Architektur

Erstmals stellen wir das Image nicht nur in einer 32-Bit-Variante zur Verfügung (armhf-Architektur), sondern auch in einer 64-Bit-Variante (arm64-Architektur). Die 64-Bit-Variante ist kompatibel mit allen RevPi-Produkten außer dem RevPi Core (mit Compute Module 1).

Falls Du eigene Programme auf deinem RevPi einsetzt, musst Du sie für die arm64-Architektur neu kompilieren. Verwendest Du proprietäre Software, die nur als 32-Bit-Binary zur Verfügung steht, musst Du in der Regel das 32-Bit-Image verwenden. Wenn Du unsicher bist, ob Du das 32-Bit- oder 64-Bit-Image installieren sollst, dann hilft Dir vielleicht diese Auflistung der Vor- und Nachteile:

<https://www.raspberrypi.com/news/raspberry-pi-os-64-bit/>

Aus Kompatibilitätsgründen liefern wir neue Geräte zunächst weiterhin mit dem 32-Bit-Image aus.

## Ende der Unterstützung des RevPi Core (mit Compute Module 1)

Mit Bullseye endet die offizielle Unterstützung des RevPi Core in der ersten Version (PR100102). Aufgrund der gestiegenen Hardware-Anforderungen verschiedenster Komponenten im Bullseye-Release ist ein stabiler Betrieb mit ausreichender Kapazität für eigene Applikationen nicht sinnvoll möglich. Kunden, die auf neuere Softwareversionen angewiesen sind, empfehlen wir einen Umstieg auf die RevPi Core S/SE Familie.

## Leichtere Auffindbarkeit des RevPi im Netzwerk

Bisher musste man Tools wie Angry IP Scanner bemühen, um einen neu installierten RevPi im Netzwerk zu finden. Beginnend mit Bullseye annonciert der RevPi seine Dienste nun per Multicast DNS (auch bekannt als mDNS, Avahi, Zeroconf oder Bonjour).

Um den RevPi über Multicast DNS zu erreichen, wird seine Seriennummer benötigt. Der WebStatus ist dann über <https://revpiNNN.local> erreichbar, wobei NNN der Seriennummer entspricht. Beispiel: Wenn die

Seriennummer 4815162342 lautet, kann der WebStatus über `https://revpi4815162342.local` angesprochen werden. Eine SSH-Verbindung ist unter der Adresse `revpi4815162342.local` möglich.

Falls Du die Seriennummer gerade nicht zur Hand oder den RevPi neu geflasht hast, kannst Du Dir mit folgendem Linux-Befehl alle lokalen RevPis anzeigen lassen, die den WebStatus-Service anbieten:

```
avahi-browse -tr _revpiwebstatus._tcp
```

Die Ausgabe sieht beispielsweise so aus:

```
1 + eth0 IPv4 RevPi Webstatus @ RevPi123          _revpiwebstatus._tcp local
2 + eth0 IPv4 RevPi Webstatus @ RevPi124          _revpiwebstatus._tcp local
3 = eth0 IPv4 RevPi Webstatus @ RevPi123          _revpiwebstatus._tcp local
4   hostname = [RevPi123.local]
5   address = [192.168.1.123]
6   port = [41443]
7   txt = []
8 = eth0 IPv4 RevPi Webstatus @ RevPi124          _revpiwebstatus._tcp local
9   hostname = [RevPi124.local]
10  address = [192.168.1.124]
11  port = [41443]
12  txt = []
```

Auf Apple-Geräten kann stattdessen die App "Flame Services Browser" verwendet werden, welche kostenlos für alle Plattformen (macOS, iOS und iPadOS) im App-Store verfügbar ist. Diese besitzt eine Suchfunktion und listet alle RevPi und deren Dienste im Netzwerk auf. Durch einen Klick auf den entsprechenden Dienst kann z. B. der WebStatus direkt geöffnet werden: <https://apps.apple.com/de/app/flame-services-browser/id325206381>



**Abschalten der Funktion:**

Wenn Multicast-DNS unerwünscht ist, z. B. weil sich der RevPi aus Sicherheitsgründen still verhalten soll, so lässt sich der Dienst mit folgenden Befehlen dauerhaft deaktivieren und eine ungewollte Reaktivierung verhindern:

```
sudo systemctl stop avahi-daemon.service
sudo systemctl disable avahi-daemon.service
sudo systemctl mask avahi-daemon.service
```

## Eigenes TLS-Zertifikat für jeden RevPi

Bisher verwendeten alle RevPi dasselbe selbst-signierte TLS-Zertifikat, um den Browser-Zugriff auf WebStatus und PiCtory abzusichern. Zur Verbesserung der Sicherheit wird nun für jeden RevPi ein eigenes Zertifikat generiert. Die Generierung erfolgt automatisiert durch den systemd-Service `revpi-cert-wizard.timer`. Die Laufzeit des Zertifikats beträgt wie bisher 10 Jahre. Eine Woche vor Ablauf wird es automatisch erneuert.



Selbst signierte TLS-Zertifikate können von Browsern nicht verifiziert werden. Diese zeigen daher eine Warnung an, weil der Herausgeber des Zertifikats nicht erkannt werden kann. Dies hat keinen Einfluss auf die Verschlüsselung, es ist lediglich keine automatische Prüfung der Authentizität des Zertifikats möglich. Ohne eine Plausibilitätsprüfung durch den Nutzer wird daher eine Verwendung von selbst-signierten Zertifikaten zur Kommunikation über ein öffentliches Netzwerk nicht empfohlen.



Die automatisierte Erneuerung des selbst-signierten Zertifikats lässt sich wie folgt abschalten:

```
sudo systemctl stop revpi-cert-wizard.timer
sudo systemctl disable revpi-cert-wizard.timer
sudo systemctl stop revpi-cert-wizard.service
sudo systemctl disable revpi-cert-wizard.service
```



Wenn Du das bisher standardmäßig ausgelieferte selbst-signierte TLS-Zertifikat beibehalten möchtest, kannst Du die beiden folgenden Dateien aus einem unserer bisherigen Images kopieren:

```
/etc/ssl/certs/revpi-self-signed.pem
/etc/ssl/private/revpi-self-signed.key
```

Das kann etwa dann sinnvoll sein, wenn Du in deinem Browser bereits eine Ausnahme für das bisherige Standard-Zertifikat hinterlegt hast und diese weiterverwenden möchtest.

Der CertWizard ist Teil des Pakets `revpi-webstatus`. Das dabei installierte Programm `revpi-cert-wizard` kannst Du auch verwenden, um ein eigenes selbst-signiertes TLS-Zertifikat zu erzeugen. Das Programm muss mit root-Rechten ausgeführt werden. Information über das Tool bekommst du mit dem Aufruf: `revpi-cert-wizard --help`.

## WebStatus und PiCtory ausschließlich über verschlüsselte Verbindung

Bisher konnten WebStatus und PiCtory über eine unverschlüsselte Verbindung im Browser aufgerufen werden. Jetzt ist das nur noch verschlüsselt möglich.

Unverschlüsselte Zugriffe auf Port 80 bzw. 41080 werden nach einem Hinweis auf den verschlüsselten Port 41443 umgeleitet.

Außerdem wird im Auslieferungszustand der Port 443, welcher der Standard für verschlüsselte Verbindungen ist, auf 41443 umgeleitet.



Wenn Du die Standard-Ports 80 und 443 für eigene Anwendungen verwenden möchtest, musst Du lediglich das Paket `revpi-webstatus-redirect` deinstallieren.

WebStatus und PiCtory bleiben weiterhin über Port 41443 verfügbar.

## NetworkManager als Ersatz für dhcpcd

Beginnend mit unserem Bullseye Image erfolgt die Netzwerkkonfiguration nun mit NetworkManager anstelle des bisher verwendeten dhcpcd. Dadurch verbessert sich die Bootzeit geringfügig.

Standardmäßig wird den Interfaces eine IP-Adresse per DHCP zugewiesen. Wenn das unerwünscht ist, können IP-Adressen entweder über die Datei `/etc/network/interfaces` oder mit dem Kommandozeilenprogramm `nmtui` zugewiesen werden. Alternativ ist eine Konfiguration auf der graphischen Benutzeroberfläche mit dem Programm `nm-connection-editor` möglich. Hierfür ist das Paket `network-manager-gnome` zu installieren. Eine Anleitung zur Einrichtung statischer IP-Adressen und anderer gängiger Konfigurationen findet sich z. B. im Debian Wiki:

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

Der NetworkManager erlaubt darüber hinaus die einfache Konfiguration von drahtlosen Netzwerken (graphisch wie auch auf der Kommandozeile).

## UTC als neue Standard-Zeitzone

Die Standard-Zeitzone unseres Images wurde von Europe/Berlin auf UTC geändert. Dies trägt unserer stetig wachsenden internationalen Nutzerschaft Rechnung. Danke für diesen Verbesserungsvorschlag an unseren Kunden Kees Jan Koster!

## Software von Drittanbietern

### Node-RED

Node-RED wird jetzt in Version 3 mit dem Image ausgeliefert. Die Node.js Version wurde auf 18 (LTS) aktualisiert, die RevPi-Nodes werden nun als Debian-Paket verteilt.

Das Node-RED-Changelog für die Version 3 ist unter folgendem Link zu finden:

<https://github.com/node-red/node-red/blob/5d698d66d0857e65b08be41b212b7bcafe5f8261/CHANGELOG.md>

Der Zugriff auf Node-RED läuft nun über den integrierten apache2-Webserver des Revolution Pi. Wenn der gewohnte Port 1880 für den Aufruf verwendet wird, macht der Browser eine Weiterleitung auf die verschlüsselte Verbindung zum Port 41880. Diese Verbindung verwendet das selbst-signierte Zertifikat des CertWizard (siehe oben). Sollte dieses Verhalten nicht gewünscht sein, kann einfach das Paket `revpi-nodered-proxy-apache` deinstalliert werden. Dann ist Node-RED nur noch auf dem lokalen System über Port 1881 verfügbar.



Die Node-RED-Oberfläche läuft auf 127.0.0.1 Port 1881, ist ausschließlich an localhost gebunden und von außen nicht erreichbar. Wenn dies geändert werden soll, kann die Datei `/usr/lib/systemd/system/nodered.service` angepasst werden.



Erstellte Knoten wie `websocket in`, `http in` oder `tcp/udp in` werden, wie gewohnt, an alle Schnittstellen gebunden und sind von außen erreichbar.

### logi.RTS nicht mehr Teil des Repositories

Beginnend mit dem Bullseye-Image wird das logi.RTS Paket nicht mehr durch KUNBUS gepflegt, sondern muss direkt beim Hersteller logi.cals heruntergeladen werden.

Eine Installationsanleitung wird von logi.cals bereitgestellt unter:

<https://help.logicals.com/display/LC3UserDocuDE/logi.RTS+auf+Raspberry+Pi+installieren+und+starten#logi.RTSaufRaspberryPi+installierenundstarten-logi.RTSaufRaspberryPiinstallieren>.

### TeamViewer nicht mehr vorinstalliert

Für das 64-Bit-Image steht TeamViewer nicht zur Verfügung. Auf dem 32-Bit-Image kann es wie folgt nachinstalliert werden:

```
sudo apt update && sudo apt install teamviewer-revpi
```

## Reorganisation von RevPi-Paketen

### pimodbus

Die "Modbus Organization" hat in ihrer Dokumentation die Begriffe "Master-Slave" durch "Client-Server" ersetzt. Daher haben wir die Namen der Debian-Pakete angepasst:

`pimodbus-master -> revpi-modbus-client`

`pimodbus-slave -> revpi-modbus-server`

Offizielle Presse-Erklärung der Modbus Organization:

<https://modbus.org/docs/Client-ServerPR-07-2020-final.docx.pdf>

### revpi-sos und revpi-config

Das Tool `revpi-sos` zum Erzeugen eines SOS-Report für unseren Support wurde vom Debian-Paket `revpi-tools` in das eigenständige Paket `revpi-sos-report` verschoben.

Das Tool `revpi-config` wurde vom Debian-Paket `revpi-webstatus` in das Paket `revpi-tools` verschoben.

### Watchdog-Dateien

Die Dateien zur Benutzung des Watchdog auf dem RevPi Connect befinden sich nun unter `/usr/share/doc/revpi-tools/` anstatt wie bisher unter `/home/pi/connect/`.

### Dateisystem-Vergrößerung

Um den Flash-Vorgang zu beschleunigen ist unser Image auf die minimale Größe geschrumpft und wird beim ersten Booten auf die volle Größe des eMMC-Massenspeichers erweitert. Bisher geschah diese Dateisystem-Vergrößerung in dem Tool `revpi-factory-reset`. Stattdessen geschieht sie nun über einen neuen Dienst namens `firstboot.service`. Der Dienst wird ausschließlich beim ersten Booten des Systems ausgeführt und startet das entsprechende Shell-Skript, welches unter folgendem Pfad zu finden ist: `/usr/share/revpi/firstboot/resize-fs.sh`.

## Upgrade eines vorhandenen Buster-Image

Die Upgrade-Prozedur entspricht der Raspberry Pi Dokumentation:

<https://www.raspberrypi.com/documentation/computers/os.html#using-apt>

Im Einzelnen:

- ▷ Ändere die Einträge in `/etc/apt/sources.list` sowie von `/etc/apt/sources.list.d/*.list` von "buster" auf "bullseye". Beispiel für `/etc/apt/sources.list`:

```
1 deb http://raspbian.raspberrypi.org/raspbian/ bullseye main contrib non-free rpi
2 # Uncomment line below then 'apt-get update' to enable 'apt-get source'
3 #deb-src http://raspbian.raspberrypi.org/raspbian/ bullseye main contrib non-free rpi
```

- ▷ Lies anschließend den Inhalt der Repositories neu ein und starte das Upgrade:

```
1 sudo apt update
2 sudo apt full-upgrade
```

⇒ Es ist möglich, dass bei `sudo apt full-upgrade` dieser Fehler auftritt:

```
1 The following packages have unmet dependencies:
2 libc6-dev : Breaks: libc6-dev (< 8.4.0-2~) but 8.3.0-6+rpi1 is to be installed
3 E: Error, pkgProblemResolver::Resolve generated breaks, this may be caused by held packages.
```

- ▷ Zur Behebung des Fehlers musst Du das Paket `libc6-dev` mithilfe des Befehls `sudo apt install libc6-dev` manuell installieren.

- Damit nach dem Upgrade die Netzwerk-Interfaces weiterhin die klassischen Interface-Namen behalten (z. B. `eth0`), ist eine manuelle Intervention nötig. Dazu bietet sich das Tool `raspi-config` an. Wähle hier den Menüpunkt 6 Advanced Options, dann A4 Network Interface Names. Mit No werden die klassischen Namen genutzt, bei der Auswahl von Yes werden die neueren Namen für die Interfaces genutzt.
- Um nach dem Upgrade weiterhin Zugriff auf WebStatus über Port 80 zu erlangen, muss eine Umleitung eingerichtet werden. Dies geschieht durch die Installation des Pakets `revpi-webstatus-redirect`. Nach der Installation ist ein Neustart des Apache-Services mithilfe von `sudo systemctl reload apache2` nötig.
- Nach dem Upgrade des Lite-Image (das normale Image betrifft dies nicht) wird die `dhcpcd`-Konfiguration nicht mehr funktionieren, da sich der Ort der ausführbaren Datei geändert hat. Die Konfiguration kann entweder in `/etc/systemd/system/dhcpcd.service.d/wait.conf` angepasst oder die Datei einfach gelöscht werden.
- Dieses Script ermöglicht einen automatisierten Wechsel auf den Network-Manager:  
<https://gist.github.com/nbuchwitz/06876674ebd0b2bab429f1a8506271a1>

### Node-RED-Upgrade



## ACHTUNG

**Die direkte Ausführung von aus dem Internet heruntergeladenen Scripten kann potenzielle Schäden verursachen.**

Es wird empfohlen, das Script vorher herunterzuladen, auf Fehler zu prüfen, und es erst dann auszuführen.

▷ Führe das Update Node-RED auf 3.x und Node.JS 18 durch:

```
1 bash <(curl -sL https://raw.githubusercontent.com/node-red/linux-installers/master/deb/update-nodejs-and-nodered) --node18
```

▷ Installiere RevPi-Nodes aus dem Debian-Paket:

```
1 apt install node-red-contrib-revpi-nodes
```

▷ Entferne lokale RevPi-Nodes:

```
1 rm -R /home/pi/.node_red/node_modules/node-red-contrib-revpi-nodes
```

▷ Starte Node-RED neu:

```
1 systemctl restart nodered
```