

[aaronrombaut.com](https://www.aaronrombaut.com)

RTL-SDR Blog V4 on Raspberry Pi OS

3–4 minutes

- [Introduction](#)
- [Purge Old Drivers](#)
- [Prepare to Build from Source](#)
- [Block the Default Drivers from Loading](#)
- [Test the Installation](#)
- [Conclusion](#)

Introduction

I do have to admit that using the word *blog* with a physical device is a little odd to me. Nonetheless, this little dongle is quite a nice addition to a new amateur radio operator like myself. Even if you *do not have* a license, you can try this out since it is a receiving only device. One of the nice things about the RTL-SDR Blog V4 is that with the right software, you can see the spectrum with the waterfall graph. This can help in tuning to an active frequency or alternatively, help in tuning to an inactive frequency.

Purge Old Drivers

The first thing to do is purge *any* previously installed drivers. These

may have been a part of the OS or they may have come in with other software.

```
sudo apt purge ^librtlsdr
```

Funny story (now, but certainly not at the time) about the next block. On the blog's page for the [Quick Start Guide](#), the line cuts off at /usr/. Well guess what happens when you put a `sudo rm -rvf` in front of it? That's right, that system was quickly rebuilt afterwards.

Please be sure to copy the *entire* block of code!

```
sudo rm -rvf /usr/lib/librtlsdr* /usr/include/rtl-  
sdr* /usr/local/lib/librtlsdr* /usr/local/include/  
rtl-sdr* /usr/local/include/rtl_* /usr/local/bin/  
rtl_*
```

Prepare to Build from Source

We will need to ensure we can checkout and build the files.

```
sudo apt-get install libusb-1.0-0-dev git cmake pkg-  
config build-essential
```

```
git clone https://github.com/rtlsdrblog/rtl-sdr-  
blog
```

Change into the directory.

```
cd rtl-sdr-blog/
```

Make a new **build** directory and change into it.

```
mkdir build; cd build
```

The next steps will configure the software for your system and you will also perform the install.

```
cmake ../ -DINSTALL_UDEV_RULES=ON
```

```
make
```

```
sudo make install
```

```
sudo cp ../rtl-sdr.rules /etc/udev/rules.d/
```

```
sudo ldconfig
```

Block the Default Drivers from Loading

The RTL-SDR Blog V4 is based on the DVB-T. Linux will try to load these automatically. The next command will ensure a permanent block of these drivers takes place. This will survive a reboot.

```
echo 'blacklist dvb_usb_rtl28xxu' | sudo tee --  
append /etc/modprobe.d/blacklist-  
dvb_usb_rtl28xxu.conf
```

Test the Installation

After installing and optionally restarting the Raspberry Pi (I had to restart), you can test the installation and configuration. The dongle is slightly on the bigger side and the Raspberry Pi crams all four USB ports together. This may be tricky and depending on your peripherals, may require a USB extension cord to not put stress on the ports. That is the one downside to the Raspberry Pi for me, wires and wires and wires! (Also, why the heck are there no USB-C ports on this thing...in 2024?)

Type the following command to test the device.

```
rtl_test
```

To quit the test, type the following.

Ctrl + C

Conclusion

At this point, you should have a fully installed RTL-SDR Blog V4 installed. The next thing to do would be to figure out the [antenna configuration](#) for what you want to receive and install some other software (like [Gqrx](#) or SDR++, maybe?) to take advantage of the receiving capabilities.
