# RTL-SDR Blog V4 Users Guide

This page serves to inform users about information relevant specifically to the RTL-SDR Blog V4. We encourage users to start with our quickstart guide in order to install the drivers and some common software first.

## RTL-SDR Blog V4 Drivers

Because of required changes we made to the circuitry, new drivers are required for the RTL-SDR Blog V4. These drivers are backwards compatible with the RTL-SDR Blog V3 and older / other branded / generic RTL-SDR units as well.

The upstream Osmocom code now has included our changes to make the RTL-SDR Blog V4 work. However, you may still need to update the drivers to the latest version.  Replacing the drivers is very simple as we'll describe below. Below we prefer to use our own RTL-SDR Blog fork of the drivers as we have added additional features and enhancements for our V3/v4 models.

If you do not update or replace the drivers, then the V4 will simply not function correctly. Either you will get no signals, or signals may appear at the wrong frequency, or appear corrupted in some way.

We are working with various software providers to support the V4 natively, but for now if your software doesn't support it natively please use the instructions below.

## Windows

### Almost all software (Confirmed to work on SatDump, SDR#, SDR-Console V3, SDR++, Welle.io, RTL1090v2)

On Windows, updating the driver is usually just a matter of replacing a .dll file with one from out RTL-SDR Blog Driver Release on GitHub. Most software will use the rtlsdr.dll file in the x64 folder, but notably x86 programs like SDR#, welle.io and RTL1090 require the dll from the x86 folder.

In our quickstart guide we have included instructions on how to replace the dll for SDR#, SDR++ and CubicSDR.

For any other software the procedure will be similar:

1. Download the Releases.zip file from our RTL-SDR Blog Driver GitHub releases page.
2. Copy the rtlsdr.dll file into the folder of the software you are using. Usually this will be replacing an existing rtlsdr.dll file.
3. On some computers you may also need to copy the msvcr100.dll and pthreadVC2.dll file into the folder too. It does not hurt to copy these in anyway, even if you don't need them.

If for some reason the RTL-SDR Blog release is not working for you, please try the latest Osmocom build at https://ftp.osmocom.org/binaries/windows/rtl-sdr which is now also compatible.

## SDR# (SDRSharp)

SDR# already downloads the RTL-SDR Blog branch when the install-rtlsdr.bat file is run. So simply re-run install-rtlsdr.bat on an existing installation, or download a new installation and run install-rtlsdr.bat as you usually would.

## HDSDR & SDRUno

For HDSDR and SDRUno please use the ExtIO from https://github.com/hayguen/ExtIO_RTL/releases. Extract this ExtIO DLL file into your HDSDR or SDRUno folder and choose it if HDSDR asks for the ExtIO on startup.

**Make sure that you disable "IF AGC"** as this does not work correctly on the V4.

Alternatively for HDSDR and SDRUno you can also use extio_rtl_tcp which will allow you to connect to an rtl_tcp server.

## Spektrum

1. Download the latest Release.zip from https://github.com/rtlsdrblog/rtl-sdr-blog/releases.
2. Go into the spektrum\lib folder and rename the file librtlsdr.dll to librtlsdr.dll_old, or just delete librtlsdr.dll.
3. Copy rtlsdr.dll, msvcr100.dll and pthread.dll from the x64 folder into the spektrum\lib folder.
4. Rename rtlsdr.dll to librtlsdr.dll

# Linux (Debian)

On Linux the procedure is also simple quite simple and just involves removing any existing RTL-SDR drivers, and installing our version.

1. Purge the previous driver:

```
2. sudo apt purge ^librtlsdr
   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_*
```

3. Install the RTL-SDR Blog drivers:

```
4. sudo apt-get install libusb-1.0-0-dev git cmake
5. git clone https://github.com/rtlsdrblog/rtl-sdr-blog
6. cd rtl-sdr-blog
7. mkdir build
8. cd build
9. cmake ../ -DINSTALL_UDEV_RULES=ON
10.  make
11.  sudo make install
12.  sudo cp ../rtl-sdr.rules /etc/udev/rules.d/
   sudo ldconfig
```

13. Blacklist the DVB-T TV drivers.

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

14. Reboot

## Alternative Debian Package Installation Method

If you have a system reliant on the Debian packages (eg. FlightRadar24, FlightAware, ADSBExchange images) you can update them directly using this method:

```
sudo apt update
sudo apt install libusb-1.0-0-dev git cmake
sudo apt install debhelper

git clone https://github.com/rtlsdrblog/rtl-sdr-blog
cd rtl-sdr-blog
sudo dpkg-buildpackage -b --no-sign
cd ..

sudo dpkg -i librtlsdr0_*
sudo dpkg -i librtlsdr-dev_*
sudo dpkg -i rtl-sdr_*
```

# Linux (arch)

On arch distributions this package is now available https://aur.archlinux.org/packages/rtl-sdr-blog-git

# MacOS

Unfortunately MacOS will most likely require that software providers update their package files themselves as packages come with standard drivers prepackaged. SDR++ and SDRAngel and GQRX are now compatible. We are still awaiting for a reply and a new release from CubicSDR.

### SDR++ MacOS

The latest nightly build is now compatible with the RTL-SDR Blog V4 and MacOS Intel systems. Get it here https://github.com/AlexandreRouma/SDRPlusPlus/releases/tag/nightly

Note that to run it you may need to use the "Show Package Contents" menu -> then browse to Contents -> MacOS -> sdrpp, and run that file. Otherwise you may get an error.

## SDRAngel MacOS

SDRAngel is now compatible with the RTL-SDR Blog V4 and MacOS Intel systems. Get the latest dmg file here https://github.com/f4exb/sdrangel/releases

## GQRX MacOS

GQRX is now compatible with the RTL-SDR Blog V4 and MacOS Intel systems. The current release is only available as a nightly build, so you will need to log into a GitHub account and download the .dmg file from https://github.com/gqrx-sdr/gqrx/actions/runs/6244124081

Note that to activate the bias tee in GQRX, set the device string as "rtl=0,bias=1". Be careful to not add a space after the comma.

### Compiling RTL-SDR on MacOS

If your software can make use of rtl_tcp, you can install the latest RTL-SDR and rtl_tcp via this method.

First make sure you have installed homebrew and xcode. Open a terminal and run:

```
brew uninstall rtl-sdr

brew install cmake
brew install libusb
brew install pkgconfig
git clone https://github.com/rtlsdrblog/rtl-sdr-blog
cd rtl-sdr-blog
mkdir build
cd build/
cmake ../
make LIBRARY_PATH=/usr/local/lib
sudo make install

rtl_test -t
```

### Now you should be able to run:

```
rtl_tcp -a 0.0.0.0
```

# Android

### SDR Touch / RTL-SDR TCP Driver

The SDR Driver App on Android has now been updated and officially supports the RTL-SDR Blog V4. Get the driver 'SDR Driver' app from the Play store, or update it if you already had it installed.

With this app you can now use any software that relies on the SDR Driver app, such as SDR Touch, Dump1090, ADS-B Radar, MagicSDR, Airband Radio, HAM FM Radio, FM Radio, RTL SDR AIS Driver, SDR Radio, Wavesink.

The new SDR Driver App also respects the bias tee force ON EEPROM setting. So if you want the bias tee always on, please consult the bias tee instructions further down on this page.

### SDR++

We are still in the process of awaiting an official SDR++ release of the updated Android app. In the meantime you can (**at your own risk**) use this fork https://github.com/hasan7/SDRPlusPlus/actions/runs/6319349153 created by a V4 customer. Ignore the error, and download the sdrpp_android artifact which is an APK.

### Other Apps

We are still in the process of awaiting releases from the devs of SDR++, SDRAngel, and the official Android SDR Driver app.

# Updating the Drivers on the FlightRadar24, FlightAware Raspberry Pi4 Images

Set up your FlightRadar24 image as normal, then either log into SSH or attach a monitor. The FlightRadar24 image has default credentials pi/raspberry, pi/flightware for FlightAware and pi/adsb123 for ADSBExchange. Not on the FlightAware to enable SSH, you need to create a file with filename "ssh" on the boot folder of the SDcard.

Once logged into the terminal, run the following commands to update the librtlsdr packages.

```
sudo apt update
sudo apt install libusb-1.0-0-dev git cmake
sudo apt install debhelper

git clone https://github.com/rtlsdrblog/rtl-sdr-blog
cd rtl-sdr-blog
sudo dpkg-buildpackage -b --no-sign
cd ..

sudo dpkg -i librtlsdr0_*
sudo dpkg -i librtlsdr-dev_*
sudo dpkg -i rtl-sdr_*
```

Now reboot your Pi4 and it should work.

## Forcing the Bias Tee on FlightRadar24, FlightAware

Note that this will only work once you have installed the RTL-SDR Blog drivers as per the instructions above. Forcing the bias tee on will not work with any other drivers.

### FlightRadar 24

```
sudo systemctl stop fr24feed
sudo systemctl stop dump1090-mutability.service
rtl_eeprom -b 1
```

### FlightAware

```
sudo systemctl stop dump1090-fa.service
rtl_eeprom -b 1
```

### ADSBExchange

```
sudo systemctl stop readsb.service
rtl_eeprom -b 1
```

After changing the EEPROM reboot your Pi 4.

## RTL-SDR Blog HF Mode

Unlike the RTL-SDR Blog V3, there is no need to activate direct sampling mode to receive HF. You can instead simply tune to an HF frequency and it will work. Activating direct sampling mode on a V4 dongle will yield no results.

## Software Selectable Bias Tee

V1 and V2 of our dongles included a bias tee which could manually be enabled by opening the case and soldering two pads on the PCB together. V3 and V4 introduces a 4.5V bias tee that can be toggled entirely in software. The bias tee can continuously pull up to 180 mA of current.

***WARNING:*** *Before using the bias tee please ensure that you understand that you should not use this option when the dongle is connected directly to a DC short circuited antenna unless you are using an LNA. Although the bias tee circuit is dual protected against accidental shorts with a thermal self-resetting fuse and overcurrent protection on the LDO, short circuiting the bias tee for an extended period of time (days) could damage the LDO or fuse permanently. Only use it while connected to an actual powered device, like an LNA, active antenna or the SpyVerter.*

*To make things clearer: DC Short Antenna -> LNA -> Coax -> V3(bias tee on) is absolutely fine. What's not good and makes no sense anyway is DC Short Antenna -> Coax -> V3(bias tee on). DC Short Antenna -> Coax -> V3(bias tee off) is fine.*

Note that the legacy DVB-T TV drivers will activate the bias tee by default. On Linux ensure that you have properly blacklisted the DVB-T drivers. More info on how to blacklist on the Linux section on the quickstart guide.

When the bias tee activates you should be able to see a small red LED illuminate, visible through a tiny hole next to the SMA port.

### Windows

In most software there is an option to enable "Offset Tuning". This option is irrelevant for R820T/R860/R828D based dongles. Instead we have repurposed it as a switch for our bias tee. This will work as long as you are running our RTL-SDR Blog drivers.

To enable or disable the bias tee, simply activate or deactivate the "Offset Tuning" option. In SDR# this option is available in the RTL-SDR Configure menu, next to the gain settings.

Alternatively you can activate the bias tee via the command prompt by using the rtl_biast.exe program available in our driver release. In a command prompt simply change directories into the folder where you have extracted Release.zip, and run "rtl_biast -b 1" to turn the bias tee ON, and "rtl_biast -b 0" to the turn bias tee OFF.

Note that rtl_biast will only run if there is no other software accessing the RTL-SDR, so make sure to close other software first.

## Linux

If you have installed our RTL-SDR Blog drivers, you can simply run "rtl_biast -b 1" at a terminal to activate the bias tee, or "rtl_biast -b 0" to deactivate it.

## Forcing the Bias Tee to be Always ON

By setting an EEPROM flag you can force the bias tee to always be ON. This is useful for software programs that don't have a bias tee option. To do this simply run the following command in a Linux terminal where you have installed the RTL-SDR Blog drivers, or a Windows command prompt where you extracted our driver release.

```
rtl_eeprom -b 1
```

The unplug and replug the dongle in after.

Note that the bias tee will only be forced on if the software os system is using our RTL-SDR Blog drivers. If any other driver is being used, the bias tee EEPROM flag will simply be ignored.