

# Installation & Configuration **Rosixehird RTL-SDR Radio Receiver**



# Table of Contents

<b>1 - Package Update</b>	<b>3</b>
<b>2 - Dependency Installation</b>	<b>3</b>
<b>3 - Downloading and Installing RTL-SDR</b>	<b>4</b>
<b>4 - Driver Dependency Installation</b>	<b>5</b>
<b>5 - Driver Installation</b>	<b>5</b>
<b>6 - Usage</b>	<b>6</b>

## 1 - Package Update



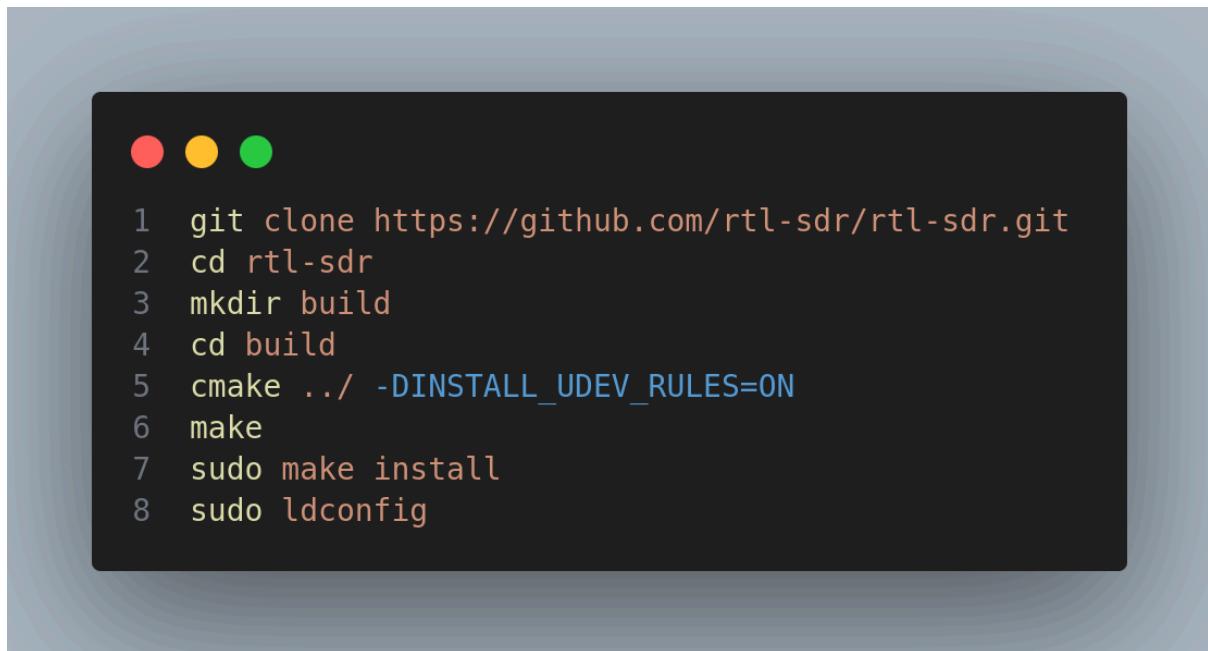
```
1 sudo apt update && sudo apt upgrade -y
```

## 2 - Dependency Installation



```
1 sudo apt install git cmake build-essential libusb-1.0-0-dev
```

## 3 - Downloading and Installing RTL-SDR



```
● ● ●

1 git clone https://github.com/rtl-sdr/rtl-sdr.git
2 cd rtl-sdr
3 mkdir build
4 cd build
5 cmake .. -DINSTALL_UDEV_RULES=ON
6 make
7 sudo make install
8 sudo ldconfig
```

We start by cloning the official repository, then create a directory to perform the installation using **CMake**.

Once the installation is complete, your receiver works properly, but a specific decoder is still missing to use the ThermoPro TX-2C sensor.

## 4 - Driver Dependency Installation



```
1 sudo apt install libtool libusb-1.0-0-dev librtlsdr-dev autoconf automake
```

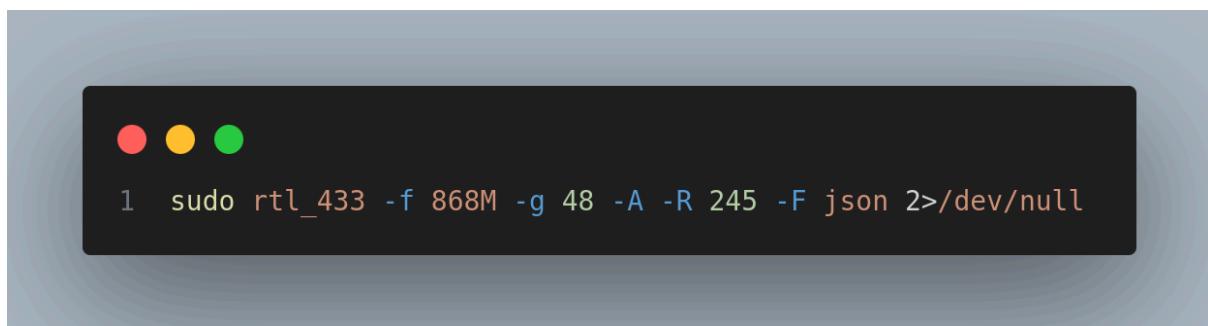
## 5 - Driver Installation



```
1 git clone https://github.com/merbanan/rtl_433.git
2 cd rtl_433
3 mkdir build && cd build
4 cmake ..
5 make
6 sudo make install
```

We start by cloning the repository containing the `rtl_433` software. Next, we create a build directory and install the software inside it using **CMake**.

## 6 - Usage



This command alone allows you to capture the temperatures sent by your ThermoPro TX-2C.

As for the parameters:

**-f** : sensor frequency (e.g., **868M** for 868 MHz).

**-g** : receiver gain (in dB). **0** = automatic gain, on this hardware, it can go up to approximately 48 dB

**-A** : pulse analyzer — displays detected pulses (timings, widths, gaps) and helps with debugging and reverse-engineering protocols.

**-R** : selects a specific decoder/protocol (in this case, **-R 245** targets decoding for the TX-2C)

**-F json** : forces the output to be in JSON format

**2>/dev/null** : redirects the error output (stderr) to **/dev/null**, which is useful for keeping only the standard output (in this case, the JSON) and removing status or diagnostic messages