

RTL-SDR Con Routers



This tutorial and future versions of it can be downloaded from the website of the
Radio Club Utiel

<https://www.ea5rca.es/>

INTRODUCTION

In order to have as little signal loss as possible between our RTL-SDR receiver and your antenna, we have several possible options:

- We can use a low-loss coaxial cable, at the cost of increasing the cost of the installation.
- We can also bring the SDR closer to the antenna using a USB extender, but depending on the distance between it and the computer, we will need an amplified cable (which is not cheap either) and it can cause problems.
- Another very common configuration consists of connecting the SDR to a Raspberry Pi, and placing these next to the antenna.
Then to simplify the installation, by means of the same Ethernet connection cable with the computer, we can upload the power (Power over Ethernet, PoE) necessary for the Raspberry + SDR set.

Comment that personally I think that the best option would be to be able to buy an antenna with SDR built-in (powered by PoE), and if it does not yet exist, we will soon see it in both receivers and transceivers, starting with QRP equipment, which requires less power supply.

Imagine how simple and efficient it would be to install an antenna on the terrace and lower an ethernet cable to our home and then connect either a computer or a console that would allow us to control the equipment.

HARDWARE CHOICE

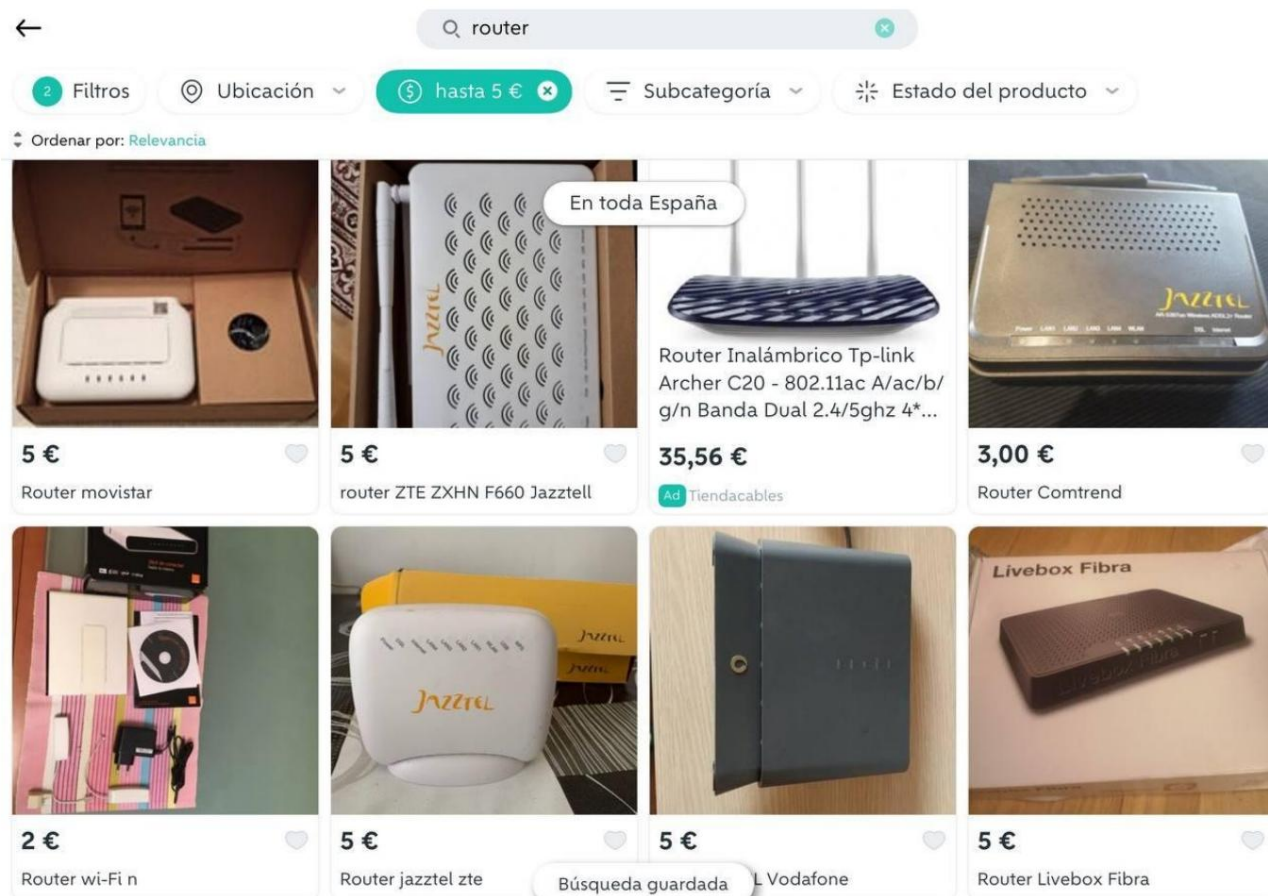
For this tutorial, we have chosen the option of connecting the SDR as close as possible to the antenna and using a Raspberry or other device that performs the interface function, connect to the computer using an Ethernet cable with PoE.

As today we are in the so-called semiconductor crisis, in which Raspberrys and similar equipment have increased their price considerably, what we are going to do is use an old router, to which we will install a firmware distribution called OpenWrt (<https://openwrt.org/>).

This is nothing new but given the circumstances I think it is the most appropriate, since apart from having fun messing around, we contribute to the environment by reusing obsolete equipment for the use that was created and that would otherwise end up in the container.

There are multiple models of routers in which OpenWrt can be installed (that I remember, the one mentioned in this tutorial is the third model in which I do it), but we also need it to incorporate a USB port and also to simplify the process, we are going to use a device to which you only have to load the firmware, without the need to make any hardware modifications, since although they are usually not very complex to implement, it could be an impediment for certain users.

Another premise when selecting the router is that it be low cost, and for this a quick search was made on an On-Line purchase and sale website, in which the amount of five euros was marked as the maximum price:



Finally, the AR-5387un router, from the Comtrend brand, was chosen:



DISCLAIMER

As always, remember that everything mentioned in this document is experimental and could cause unevaluated damage, with the person who carries it out being solely responsible.

Also indicate that changing the firmware could make the used router unusable, and therefore it is recommended to do it with equipment in which, if this were to happen, it would not cause any damage.

INSTALLING OPENWRT

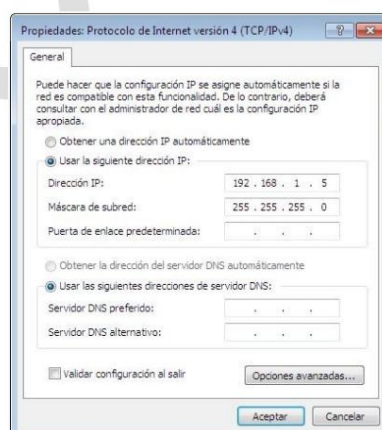
The OpenWrt installation process on this router is very simple and we have it detailed on this website <https://openwrt.org/toh/comtrend/ar-5387un>.

The first thing we have to do is download the firmware binary file that we want to install on our router, which as of today is --> **"openwrt-21.02.1-bcm63xx-generic comtrend_ar-5387un-squashfs-cfe.bin"**:

https://downloads.openwrt.org/releases/21.02.1/targets/bcm63xx-generic/openwrt-21.02.1-bcm63xx-generic-comtrend_ar-5387un-squashfs-cfe.bin

Now to be able to connect to the router when it is in "flashed mode", we have to configure our computer with a static IP address, from the range of the one the router has in this mode (which is 192.168.1.1), so we will give our computer, for example, the IP address 192.168.1.5.

Depending on the operating system that we use, this will be done in a different way, but as an example for Windows 7 (I normally use GNU Linux and for this tutorial I had to throw from an old computer that incorporates this system), we would enter the configuration of the network card and in the properties of the IPv4 protocol we would configure it as follows:



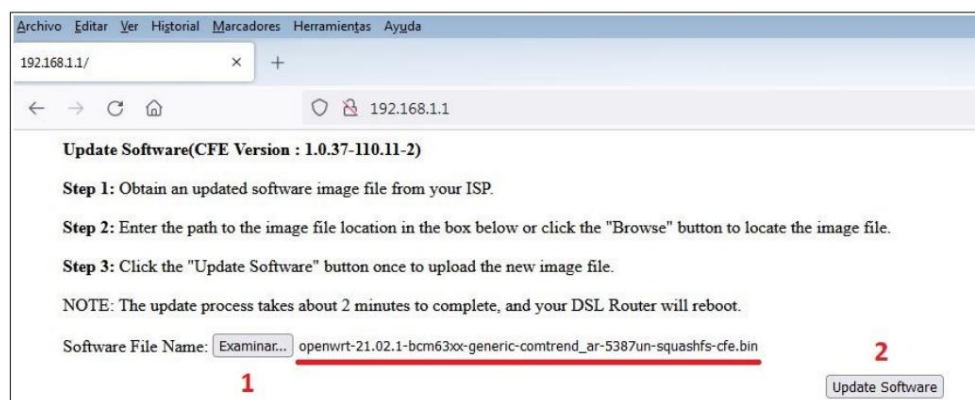
Comment that in the indications of the OpenWrt Web, they also configure the Default Gateway, with the IP of the router 192,168,1,1, but I have not needed to do it.

NOTE: At the end of this tutorial it is indicated how to configure a static IP in other operating systems.

Now it is time to put the router in flashing mode, and for this, with it being off, using a clip or something similar, we activate the Reset button that is located between the USB connector and the power switch, we turn on the router and We keep pressing the Reset button for about twenty seconds while it boots.

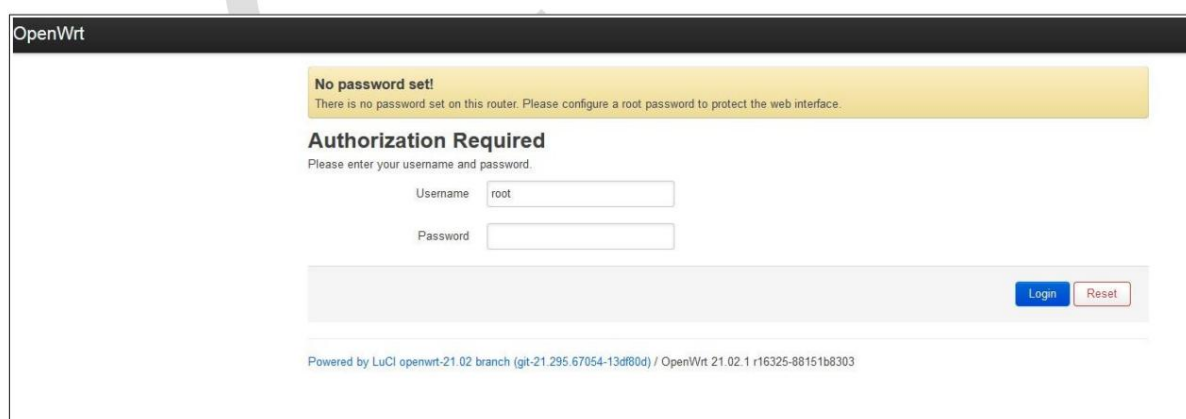
By means of an Ethernet cable we will connect the computer with any of the four ports available to the router, called the LAN interface.

If everything has gone well, when entering the IP address in the browser of the computer that we are using, the page that allows us to change the firmware of the router will appear:



Now we click on the **“Browse...”** button and select the OpenWrt firmware file that we had previously downloaded and click on **“Update Software”**.

After several minutes the router will boot with the new OpenWrt firmware installed:



As soon as it starts, it asks us to put a password on the administrator account (root). It is advisable to do it to get into the habit of setting passwords to administrator accounts, and so that it is not constantly reminding us.

We write the new password and press the **“Login”** button.

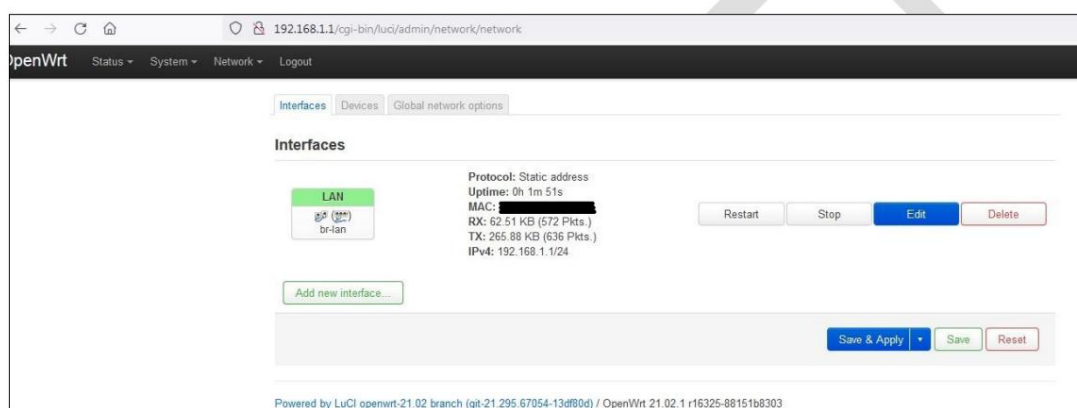
ADD NEW SOFTWARE

Once OpenWrt is installed on our router, since we want to use it as an interface for a USB RTL-SDR stick, we will have to load the corresponding software to carry out this task.

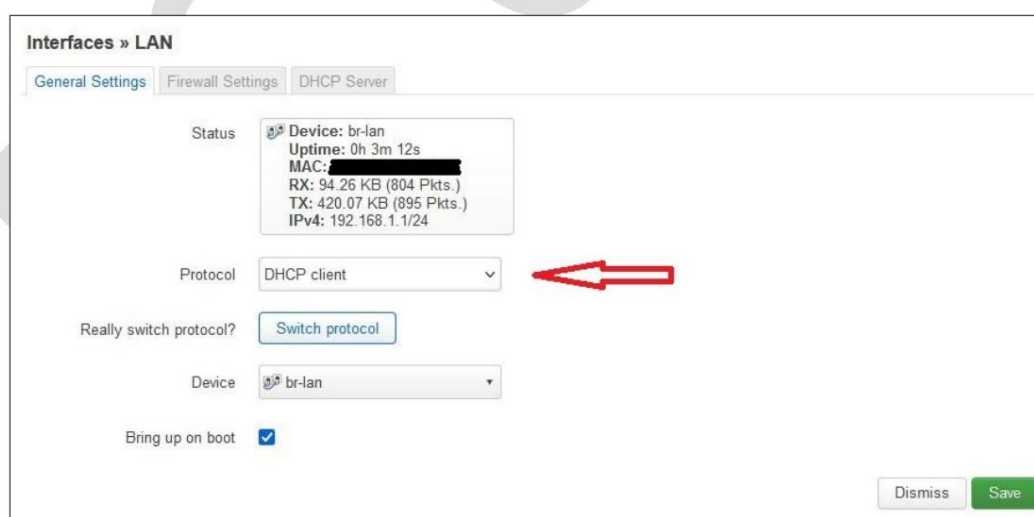
In order to add the software to the router we have to connect it to the Internet, but since it still has a fixed IP address (192.168.1.1), we have to do something so that it can establish a connection with the software servers.

What we will do is change this fixed IP address of the LAN network for a DHCP client, so that the router that we have in our home (with Internet access), can assign it a dynamic IP address and thus we will have access to the Internet in the router Comtrend.

To do this we enter the **Network Menu** , **Interfaces** and press the **Edit** button of the interface **"LAN"**:



Now in the **Protocol** drop-down menu , we will change “**Static address**” to “**DHCP client**”:



Press the "**Save**" button and for this change to take effect, once we go back to the Menu Interfaces, we will press the “**Save & Apply**” button.

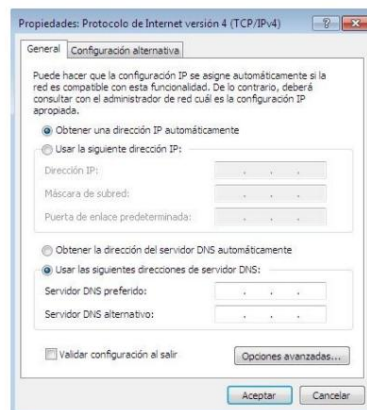
After a few seconds, the router will have changed its configuration and will be waiting for another computer to assign it a dynamic IP, and then communication with the computer to which it was connected will be lost.

In order to reestablish the connection with the Comtrend router, we have to connect the computer with which we are working, to the same local network in which it is now connected, which is the one that will provide us with Internet access.

The most common thing is that we connect them to the router that we have in our house, which will normally have four Ethernet ports.

Although the Comtrend router is now ready to establish a connection, the computer still has a fixed IP (192,168,1,5) and it is also necessary to change it to a dynamic IP configuration.

In a similar way to how we have previously entered the configuration of our network card, now we do it so that it becomes dynamic:

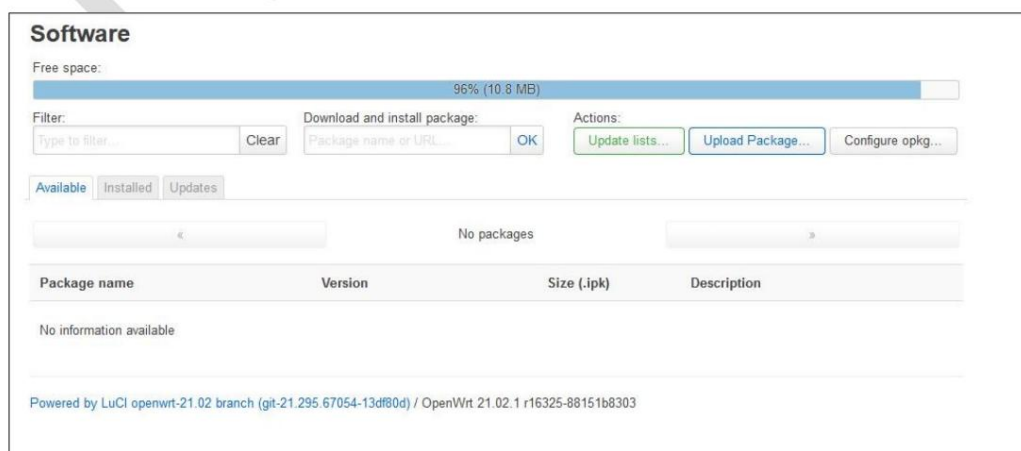


When we have all this working (we can access “https://www.google.es/” in our browser to confirm that the computer is connected correctly), we have to connect to the Comtrend router again, but now we do not know the IP address what he has, and we'll have to find out. There are several ways to do it (we are not going to go into it in depth so as not to lengthen this tutorial even more), but the simplest thing is to enter the Web that our router has with Internet access and search for the information of the clients connected to it.

There are also APPs that allow you to identify all the devices connected to a local network.

Once we know the IP of the Comtrend router, we will access it with the browser and enter the password that we have previously assigned.

Now we will enter the “**System**” Menu , “**Software**” and press the “**Update lists...**” button.

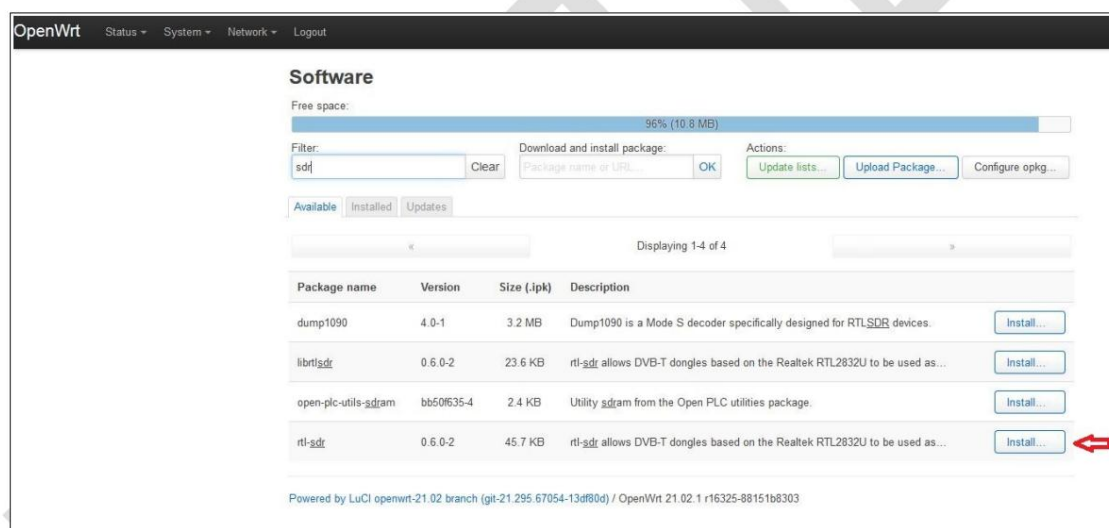


After a few seconds, a list with all the software available for this version of OpenWrt will be loaded.

In some of the tests that I have carried out, there have been times when an error appeared when performing this operation, but after resetting the router ("System" **Menu**, "**Reboot**"), it was solved:



Now to filter between all the available software, in the "**Filter**" box we write "**sdr**" and what we are looking for will appear:



Since what we want to install is the rtl-sdr function, we will have to install the packages that appear in the second and fourth rows.

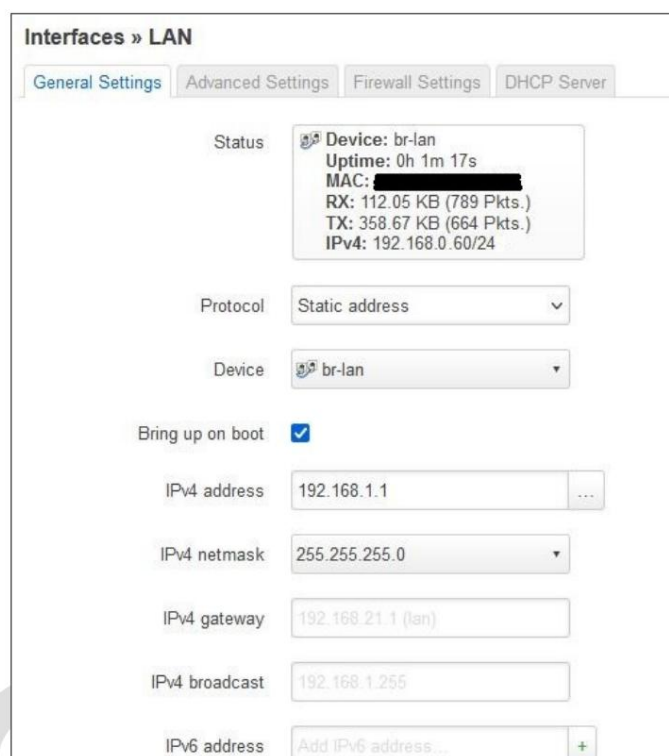
If we install the "**rtl-sdr**" package first (fourth row), the "**librtlsdr**" library package will also be installed.

The most observant will have seen that the "**dump1090**" package is also available , which is another thing to take into account and that will allow the most curious to continue investigating ;)

Don't worry, we're almost done.

If we wanted to leave the Comtrend router connected to our local network, we would not have to change anything else in said router, but since this project is designed to possibly be installed on the terrace (in a waterproof box, etc.) connected to an RTL-SDR, which in turn will be connected to an antenna, I think that as a precaution it is best to leave it isolated from the rest of the equipment with a network connection in our home.

To leave the router separate from our home network, we will set a static IP on the router again, with which we will have to leave the parameters of our LAN interface, as they were before:



The screenshot shows the 'Interfaces » LAN' configuration window in Mikrotik WinBox. The 'General Settings' tab is active. The interface 'br-lan' is selected. The status box shows: Device: br-lan, Uptime: 0h 1m 17s, MAC: [redacted], RX: 112.05 KB (789 Pkts.), TX: 358.67 KB (664 Pkts.), IPv4: 192.168.0.60/24. The Protocol is set to 'Static address'. The Device is 'br-lan'. 'Bring up on boot' is checked. The IPv4 address is '192.168.1.1'. The IPv4 netmask is '255.255.255.0'. The IPv4 gateway is '192.168.21.1 (lan)'. The IPv4 broadcast is '192.168.1.255'. The IPv6 address field is empty with a '+ Add IPv6 address...' button.

EYE! With this step, if we make a mistake and configure it wrong, we will not be able to connect to the router and we will have to start over from the beginning.

On the computer we will also have to set the static IP 192.168.1.5 again.

START RTL-SDR APPLICATION

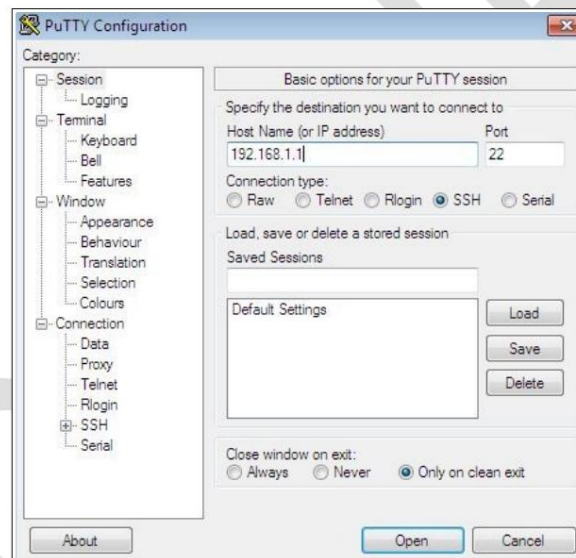
Now that we have the RTL-SDR program installed on the router, we will have to start it manually.

It would be interesting to automate this task so that when you connect the router, it will start automatically, so I encourage you to look for a possible solution to do it, and I would appreciate it if you indicate it to me so I can add it to this tutorial.

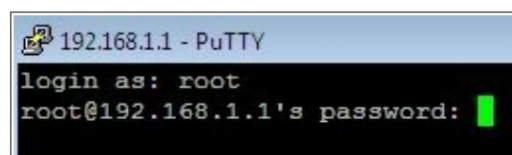
Now we are going to connect to the router via SSH, through a client that uses this communication protocol.

As I usually use the **PuTTY** program (<https://www.putty.org/>), is the one with which I am going to explain it to you.

Once the SSH client is installed, we will indicate the IP that the router has (192,168,1,1), leaving for default **port 22**, and press “**Open**”:



If the connection is established correctly, we will be asked to indicate the user (**root**) and the password that we have previously established:



If it is the first time that we connect to the router via SSH, before this we will get a security warning that we will have to confirm by pressing the "Yes" button:



After this, we will have access to the router command line:



Now we can start the rtl-sdr application indicating the corresponding IP of the router, executing the following command:

rtl_tcp -a 192.168.1.1

The default port is "1234", but if we wanted to change it, the "-p" option would be added with the corresponding port.

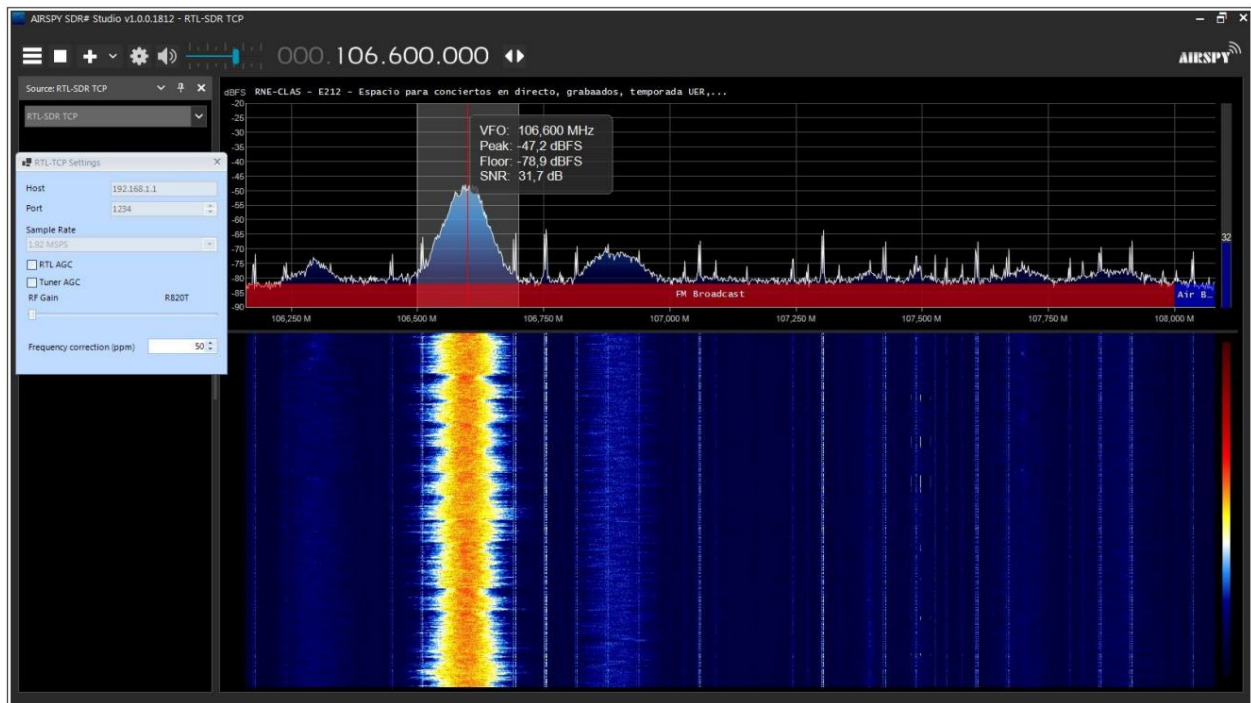
If we had more than one device (I haven't tried this yet), we also have the "-d" option, to change the device number to which we want to assign a certain port, so that for example, to change to port 4321 with device 1, we would write:

rtl_tcp -a 192.168.1.1 -p 4321 -d1

In case of doubts with the necessary parameters or to be able to test that the application correctly recognizes the SDR-RTL receiver, we have the following command available:

rtl_test

And if everything has gone well so far, we can finally start the SDRSharp program (<https://airspy.com/download>) and configure it so that it links to the rtl_sdr application that is running on the router:



Comment that not to extend more, has not entered into issues such as connecting to the router by WiFi, activate DHCP, etc., etc, but we have other options available to explore.

As I mentioned before, it would be interesting to try to automate the startup process of the rtl_tcp application, as soon as the router is connected.

To finish, indicate that although this tutorial has focused on the use of a router to manage an SDR receiver, think of all the possible things that we could do, from, for example, the installation of a Torrent client, to what our imagination is able to reach and all this for a ridiculous cost.

Soon I intend to record a video showing all this working:

<https://www.youtube.com/channel/UCIkuypr7SyEUe0pEkld5xw>

Any questions, suggestions, corrections etc., you can contact me at the following email:

eb5abt@gmail.com

Configure static IP in Ubuntu

We have to enter the **Settings application**, select **Network** and click on the cogwheel of the wired network.

Now in the **IPv4** tab we select Manual IPv4 Method and set the desired IP and netmask:

The screenshot shows the 'Cableada' (Wired) network settings window in Ubuntu. The 'IPv4' tab is active. Under 'Método IPv4', the 'Manual' option is selected. Under 'Direcciones', the first entry has 'Dirección' set to 192.168.1.5 and 'Máscara de red' set to 255.255.255.0. The 'Puerta de enlace' field is empty. Buttons for 'Cancelar', 'Cableada', and 'Aplicar' are at the top.

Configure static IP in Windows 10

Go to **Control Panel, Network and Internet, Network and Sharing Center**, and click **Change adapter settings**. On this, with the right mouse button click on **Properties** and then select **Properties Internet Protocol version 4 (TCP/IPv4)** and set the required IP and subnet mask:

The screenshot shows the 'Propiedades: Protocolo de Internet versión 4 (TCP/IPv4)' dialog box in Windows 10. The 'General' tab is selected. Under 'Puede hacer que la configuración IP se asigne automáticamente si la red es compatible con esta funcionalidad', the 'Usar la siguiente dirección IP' option is selected. The 'Dirección IP' field is set to 192.168.1.5, the 'Máscara de subred' is set to 255.255.255.0, and the 'Puerta de enlace predeterminada' is empty. Under 'Puede hacer que la configuración de servidor DNS se asigne automáticamente si la red es compatible con esta funcionalidad', the 'Usar las siguientes direcciones de servidor DNS' option is selected. The 'Servidor DNS preferido' and 'Servidor DNS alternativo' fields are empty. The 'Validar configuración al salir' checkbox is unchecked. Buttons for 'Aceptar' and 'Cancelar' are at the bottom.