THE HOBBYIST'S GUIDE TO THE RTL-SDR

REALLY CHEAP SOFTWARE DEFINED RADIO

CARL LAUFER

THE HOBBYIST'S GUIDE TO THE RTL-SDR:

REALLY CHEAP SOFTWARE DEFINED RADIO

A GUIDE TO THE RTL-SDR AND CHEAP SOFTWARE DEFINED RADIO BY THE AUTHORS OF THE RTL-SDR.COM BLOG

TABLE OF CONTENTS

PREFACE

In February 2012 the first FM radio signal was received with an RTL2832U RTL-SDR dongle using custom SDR drivers. Since then tens of thousands of hams, security researchers, hackers, makers, tinkerers, students and electronics enthusiasts have purchased RTL-SDR dongles to use as a very cheap software defined radio.

This book is intended to be a comprehensive guide for hobbyists on the use of the RTL-SDR dongle. The book consists mainly of tips to get the best out of your RTL-SDR and tutorials for some of the various interesting projects that can be done using the dongle.

The information and tutorials in this book are up to date at the time of writing. Because SDR technology and its supporting software is evolving at such a fast pace, we cannot guarantee that they will work without the need for some tweaking in the future. We will do our best to keep this book updated and will aim for an update schedule of every 3 - 4 months. To receive the updates you will need to manually contact Kindle support by email and request the update. We cannot push updates out automatically as this would disrupt peoples bookmarks, highlights and notes. You can see the edition the book is up to on the Amazon sales page.

If you discover any mistakes, missing information or just have any feedback on the book please feel free to contact me at rtlsdrblog@gmail.com.

If you are unsatisfied with this book in any way, remember that it can always be refunded through Amazon within 7 days of the purchase. But if you enjoy this book we would very much appreciate it if you were to leave us good review on the Amazon store page.

Tips for reading on Kindle: Be sure to adjust the font size settings to your preference as the default font size can be very large. All Kindle readers have this setting in their options or tool bars. It is also recommended to read this book in two column mode if reading on a PC. The free Calibre software is a good alternative to the Kindle software for reading this book. This book has DRM disabled, so feel free to use Calibre to convert it into a PDF or what ever document format you prefer.

INTRODUCTION

WHAT IS SOFTWARE DEFINED RADIO (SDR)?

In traditional hardware radios, the mathematical operations required to decode and process radio signals are performed using analogue circuitry.

Recently, computers have become powerful enough to perform the required mathematical calculations in software, hence the term software defined radio.

This has led to the creation of advanced radios that previously required complicated analogue hardware now being able to be implemented easily in software. Advanced radio capabilities such as wideband tuning and waterfall displays are now available at much lower costs.

WHAT IS THE RTL-SDR?

The RTL-SDR is an extremely cheap software defined radio which is based on mass produced DVB-T TV (Digital HD TV) USB receiver dongles that have the RTL2832U chip in them. Back in 2012 it was discovered by hardware hacker Eric Fry, Linux driver developer Antti Palosaari and the Osmocom team (who were developing their own SDR) that the RTL2832U chip had a special mode which enabled it to be used as a general wideband SDR.

Today, by using custom software drivers, a commonly found and low cost TV dongle (under $20 USD) can be turned into a sophisticated SDR receiver with features that would have until recently cost in the hundreds to thousands of dollars.

Of course, the performance of these dongles will not match a dedicated SDR, but they perform extremely well for the price and almost all hobbyist projects that can be done with expensive radios or SDRs can also be done with the RTL-SDR. Also because the RTL-SDR has become so ubiquitous it also has the advantage of a large community, meaning that there is a greater amount of free software available.

Owning a wideband SDR opens up many interesting possible projects and avenues to explore. Applications of the RTL-SDR include the following, some of which will be discussed in more depth in the project tutorials chapter.

- Listening to unencrypted Police/Ambulance/Fire/EMS conversations.
- Listening to aircraft traffic control conversations.

- Tracking aircraft positions like a radar with ADS-B decoding.
- Decoding aircraft ACARS short messages.
- Scanning commercial trunked radio conversations.
- Decoding unencrypted digital voice transmissions.
- Tracking maritime boat positions like a radar with AIS decoding.
- Decoding POCSAG/FLEX pager traffic.
- Scanning for cordless phones and baby monitors.
- Tracking and receiving meteorological agency launched weather balloon data.
- Tracking your own self launched high altitude balloon for payload recovery.
- Receiving ham radio balloons.
- Decoding wireless temperature sensors and wireless power meter sensors.
- Listening to VHF amateur radio.
- Decoding ham radio APRS packets.
- Watching analogue broadcast TV.
- Sniffing and analysing GSM signals.
- Using the RTL-SDR on your Android device as a portable radio scanner.
- Receiving GPS signals and decoding them.
- Using the RTL-SDR as a spectrum analyzer.
- Receiving NOAA and Meteor-M weather satellite images.
- Listening to satellites and the ISS.
- Listening to unencrypted military communications.
- Radio astronomy - observing the hydrogen line and solar activity.
- Monitoring meteor scatter.
- Listening to FM radio and decoding RDS information.
- Listening to DAB broadcast radio.
- Using the RTL-SDR as a panadapter for your traditional hardware radio.
- Decoding taxi mobile data terminal signals.
- Listening to amateur radio hams on SSB with LSB/USB modulation.
- Decoding digital amateur radio ham communications such as CW/PSK/RTTY/SSTV.
- Receiving HF weatherfax.
- Receiving digital radio monodiale shortwave radio (DRM).
- Listening to international shortwave radio.

- Looking for RADAR signals like over the horizon (OTH) radar, CODAR and HAARP signals.
- Using the RTL-SDR as a high quality atmospheric noise entropy source for random number generation.
- Using the RTL-SDR as a noise figure indicator.
- Reverse engineering unknown RF protocols.
- Triangulating the source of a signal.
- Searching for RF noise sources.
- Characterizing RF filters and measuring antenna SWR.

WHAT EQUIPMENT DO I NEED TO GET INTO RTL-SDR?

All that is really needed is the following commonly found equipment:

- An RTL-SDR dongle.
- An antenna for your project (purchased or home built).
- Some coax cable and an adapter to connect the RTL-SDR to the antenna.
- A computer (at least a dual core CPU is recommended), or Linux embedded PC.
- RTL-SDR software and drivers (most software is free).

Furthermore, you can get more advanced with some extras:

- An upconverter to receive the HF bands (0 - 30 MHz).
- A low noise amplifier (LNA) to improve reception.
- Preselector filters to reduce out of band interference and noise.

Places to buy some of these products can be found at http://www.rtl-sdr.com/buy-rtl-sdr-dvb-t-dongles/.

RTL-SDR TECHNICAL SPECIFICATIONS

- 22-2200 MHz tunable range (approx. depends on tuner model).

- 3.2 MHz max bandwidth (~2.8 MHz stable).
- 8-bit ADC (~7 ENOB) giving a little under ~50 dBs of dynamic range.
- < 4.5dB noise figure LNA.
- 75 Ohm input impedance.

RTL-SDR ADC

ADC is an acronym for “Analogue to Digital Converter”. It is a microchip that reads in an analogue signal voltage and then digitizes it. The more bits an ADC has, the more accurate the digitization can be. For example an 8-bit ADC can scale the analogue input voltage into values between -127 and +127, whereas a 12 bit ADC can scale input voltage from -2047 to +2047.

With a low bit ADC certain small details in the analogue input, such as weak signals may be lost during digitization. This is especially true if there are strong and weak signals nearby. **The RTL-SDR has an 8-bit ADC**, which is fairly low, but just large enough to give decent performance.

Dynamic range is the range between the largest and smallest possible values. The dynamic range of an ADC can be calculated approximately with: number_of_bits * 6 dB. This gives the RTL-SDR approximately 50 dB of dynamic range. However, the dynamic range can be made slightly better due to the oversampling trick that can be performed in software.

Some experimental tests shown in this link http://www.rtl-sdr.com/comparison-several-sdrs-degradation-broadcast-fm-frequencies/ show that the RTL-SDR does indeed have a dynamic range that is slightly better than 50 dBs.

RTL-SDR BANDWIDTH

The maximum bandwidth of the RTL-SDR is 3.2 MHz, however the largest stable bandwidth is either 2.4 MHz or 2.8 MHz depending on your PC. Setting the bandwidth too large can cause samples to be lost on slow PCs, which causes choppy audio.

Most RTL-SDR compatible software will let you choose your bandwidth which is sometimes referred to as sample rate as well. Although the sample rate and bandwidth are not the same thing, in the RTL-SDR setting the sample rate to 2 Msps (Mega samples per second) will give you 2 MHz of bandwidth. Setting it to 2.8 Msps will give you 2.8 MHz of bandwidth. (If you are familiar with Nyquist you might wonder how 2 Msps can give 2 MHz; this is because the RTL-SDR uses I/Q sampling with two ADCs).

The bandwidth is the size of the real time frequency spectrum that you can see at any one time.

INPUT IMPEDANCE

As RTL2832U dongles are intended for use with TV hardware, they all have an **input impedance of 75 Ohms**. Most amateur and professional radio equipment runs on 50 Ohm cabling, connectors and adapters.

You might think this impedance mismatch will be a problem, however the signal loss due to the mismatch is minimal, equating to less than 0.2 dB.

RTL-SDR MINIMUM PC SPECIFICATIONS

To use SDR#, the most popular Windows software for the RTL-SDR any modern PC with a dual core processor, at least 1 GB of memory and Windows XP or newer should be sufficient. The PC must also have a USB 2.0 or newer port. Although the minimum specs are quite low, beware that older and low specced PCs will struggle and may need to have the bandwidth and FFT resolutions reduced to be able to cope.

Slower PCs and embedded microcontroller based computers like the Raspberry Pi can be used with efficient command line software.

RTL-SDR COMPATIBLE DONGLES

Almost any DVB-T dongle with the RTL2832U chip can be used with the RTL-SDR drivers. However, **one must pay close attention to the tuner chip used in the dongle**. The type of tuner chip used defines the frequency range of the dongle.

There are two commonly used tuner chips. These are the **R820T/R820T2 and the E4000**. There are also the less common FC0013 and FC0012 and the even less common R828D and FC2580. In the table below the frequency range of each tuner is shown.

Tuner	Min Freq (MHz)	Max Freq (MHz)
R820T/R820T2	24	1766

E4000	52	2200
FC0012	22	948.6
FC0013	22	1100
R828D	24	1766
FCI FC2580	146 - 308	438 - 924

The R820T and its improved variant the R820T2 are the most commonly desired and purchased RTL-SDR variants. The R820T2 is a newer variant of the R820T tuner which has better sensitivity compared to the R820T. The E4000 used to be the most popular since they were the first available, but Elonics, the manufacturer has closed down making these chips rare and expensive. But don’t fret at having missed out on cheap E4000 dongles as the R820T/R820T2 tuners are the better overall performers in most cases. Other tuners variants are less common, especially online, but some compatible tuner variants can sometimes be found in local stores in the EU. These tuners are not as common for a reason - they don’t perform as well as the R820T/R820T2’s.

Based on price and availability, currently we recommend the **R820T2** tuner, unless you require the higher frequencies that the E4000 tuner gives. If you don’t need these higher frequencies then buy an R820T2, as there is no other real advantage to the E4000 tuner.

TIPS FOR BUYING RTL-SDR DONGLES

There are several online places to buy RTL-SDRs. Firstly, we at RTL-SDR.com now sell our own RTL-SDR dongles. We sell R820T2 units with several improvements made to the stock units including use of an SMD oscillator and improved component tolerances. The improved component tolerances ensure that the device is operating optimally according to its design and the SMD oscillator offers lower frequency offset and drift. Our main package also includes two telescopic antennas with a magnetic mount antenna base. The magnetic mount antenna base has higher quality coaxial cable compared to other vendors and the telescopic antennas go from 7cm to 20cm and 20cm to 1.5m. As for warranty we will replace any unit that arrives faulty without the need for returns. We ship from the USA ensuring fast delivery times for our US customers. See our Buy RTL-SDR Dongles page online for information on purchasing dongles from us http://www.rtl-sdr.com/buy-rtl-sdr-dvb-t-dongles/.

If we are out of stock, or you would like an alternative choice the second place that we recommend is the Nooelec Amazon store. Like us, Nooelec are also RTL-SDR experts and they can provide installation support over the phone should you need it. Nooelec also ship from the United States ensuring fast shipping. When buying from Amazon you also get the extra Amazon buyers protection and lenient returns policy.

Dongles are also available on eBay. However, you must be careful when buying on eBay as sometimes some sellers misrepresent their product. This is especially the case with the more expensive E4000 tuner dongles. Some sellers advertise E4000 dongles cheaply, when in fact they are actually R820T dongles. Beware of words like "E4000 Upgrade Version"; in most cases unless the price is very high (over $50 for a single unit) you will receive an R820T or FC0012/13. Additionally, when buying on eBay most dongles priced cheaply will come directly from China, meaning slow shipping and no returns policy if you receive a faulty dongle. Finally even dongles advertised as FC0012/13 also often turn out to be R820Ts.

Aliexpress.com is also another place to purchase these dongles. However, the same warnings as with Ebay apply here, plus shipping and processing times tend to be much slower.

R820T/R820T2 PACKAGES

The commonly bought R820T/R820T2 dongle comes in many packages. The most common and fit for most purposes is the black dongle with MCX connector.

There is also the 'nano' package available which is very tiny at about 1cm x 1cm in size.

There are also these white dongles which have PAL (Belling-Lee) connectors.

While they all have similar performance, we recommend the ones with the MCX connectors such as the standard and nano packages. MCX connectors have less insertion loss at GHz frequencies which is important for applications like ADS-B. See the Antenna Adapter Guide Section for information about insertion loss. Note that it has also been reported that the 'nano' models get hotter causing greater frequency instability. Advanced users may want the dongle with the belling lee connector as these connectors are easier to desolder and replace with a better connector like BNC.

OTHER SDRS WORTH MENTIONING

Although there is not yet any SDR that is more value for money that the RTL-SDR, there are several more advanced SDR radios available for purchase. Better SDRs can come with improvements like front end filters, larger ADCs, wider bandwidths, wider frequency ranges and possibly transmit capabilities. Larger ADCs give better dynamic range and sensitivity and front end filters reduce out of band interference. We list some the most popular alternative SDRs below.

	Freq Range (MHz)	**Bandwidth (MHz)**	**ADC (Bits)**	**Comments**	**Price (USD)**
Funcube	0.150 - 260 410 - 2050	0.192	16	RX. Filters	200
HackRF	30 - 6000	20	8	RX/TX	299
BladeRF	300 - 3800	40	12	RX/TX	400, 650

USRP B200/B210	70 - 6000	56	16	RX/TX	675/1100
SockRock Ensemble II	0.180 - 3	0.192	Soundcard	RX/TX. Filters.	67 (TX) 89 (RX/TX)
AirSpy	24 - 1750	10	12	RX. Filters.	199
Mirics (Msi3101)	64 - 108 162 - 240 470 - 960	5	12	RX	85
SDR Play	100 – 380 430 – 2000	8	12	RX. Filters.	149

For reception only dongles there is the AirSpy which has improved performance over the RTL-SDR with its improved ADC bit size and good RF design. The Funcube dongle is another device similar to the RTL-SDR, but with improved RF design and some switched front end filters. The Mirics Msi3101 is another dongle like the RTL-SDR as it was intended for DVB-T TV but also has SDR capabilities. Currently Mirics TV dongles are difficult to find and if you can find them they are quite expensive with a poor frequency range. Another SDR is the SDR Play which uses the Mirics Msi3101 chip in its design and also has switched first order filters. In most cases according to some tests at http://www.rtl-sdr.com/comparison-several-sdrs-degradation-broadcast-fm-frequencies/, the Airspy is the best performing RX only device, followed by the Funcube and potentially the SDR Play.

In the mid to high end of the SDR hardware spectrum we have the HackRF/BladeRF and USRPs which are all capable of receive and transmit. The HackRF is however only capable of half duplex operation, meaning that it can only either transmit or receive at any one time. The BladeRF and USRP radios are full duplex meaning that they can receive and transmit at the same time. The USRP B210 is even capable of receiving and transmitting on two channels at the same time while the B200 is full duplex on only one channel.

If you are mainly interested in the HF bands where ham radio is not active, you may want to get a SDR dedicated to the ham HF frequencies. There are many of these HF SDRs available these days and they have much better HF performance compared to the more general purpose SDRs listed above. On our blog we have a very large list of many alternative SDRs at http://www.rtl-sdr.com/roundup-software-defined-radios/. One popular low cost HF only SDR is the Softrock Ensemble II receiver (can also transmit with the appropriate kit) which comes as a kitset. You will need good soldering and construction skills to complete the kit. It has excellent shortwave reception performance.

For those just wanting a taste of what it might be like to own a HF SDR, there is the excellent online WebSDR http://websdr.ewi.utwente.nl:8901/ which is a great receiver for those wanting to just look at what might be in the HF bands.

SOFTWARE DEFINED RADIO BASIC THEORY

This book is titled *“The Hobbyist’s Guide”* and as such we will only only very lightly touch on SDR algorithm theory. Below we will very simply explain SDR theory, but please note that there is a lot more to the theory and design of SDR systems than what is described below.

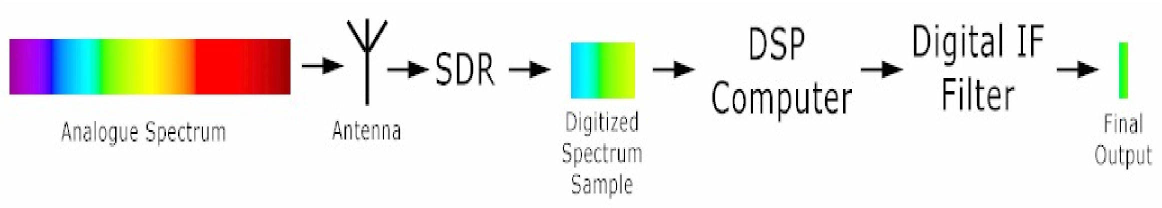
A software defined radio simply works by receiving an analogue radio signal with an antenna and then using an Analogue to Digital Converter (ADC) to digitize the signal. The digitized signal can then be worked on in digital signal processing software on a standard PC.

In practice an ADC will only work up to a certain frequency. The RTL2832U is a type of ADC which works up till 28.8 MHz. To digitize higher frequency signals we need a mixer stage to convert all received frequencies down to the frequencies that the ADC can use. This is the job of the tuner chip (e.g R820T/E4000).

In very brief, the total process is that the signal is received by an antenna, amplified with an internal LNA, mixed by the tuner, filtered to removed aliases, amplified again enough for the ADC to be able to read the signal and then digitized with the ADC. Once digitized the digital I/Q data is sent over USB into the PC where signal processing algorithms are used to demodulate and/or decode the signal.

At any one time, an SDR receives and processes a “chunk” of the RF spectrum, e.g perhaps 2 MHz worth of bandwidth, or more for more advanced SDR’s. Within this chunk of bandwidth there may be many signals. The software digital signal processing side of things handles the tuning of individual signals within this chunk of bandwidth.

We might think of the following heavily simplified analogy with the color spectrum. Let's say that from the color spectrum (purple to red) we want to use our SDR to "listen" to the green signal. First we use our antenna to receive the entire spectrum. The SDR digitizes a section of it (how big the section is depends on the SDR's bandwidth), and sends the digitized section to the computer running DSP software. The DSP software then uses a digital IF filter algorithm to extract only the green signal that we want to listen to.



Advanced: The RTL-SDR uses I/Q sampling, where two ADCs are used, one for the minus part from the DC offset and one part for the positive part of the DC offset. This is how we can get 3.2 MHz of bandwidth from a 3.2 Msps ADC sampling rate despite the Nyquist limit. (If an ADC can sample at 3.2 Msps, then the maximum bandwidth it can digitize is 3.2/2 = 1.6 MHz due to Nyquist.)

If you are interested in a more in depth treatment of SDR theory, the ARRL has a good article at http://www.arrl.org/files/file/Technology/tis/info/pdf/020708qex013.pdf and the University of Colorado College of Engineering and Applied Science has an excellent in depth lab sheet on the SDR theory of the RTL-SDR in particular at http://www.eas.uccs.edu/wickert/ece4670/lecture_notes/Lab6.pdf.

SETTING UP AND USING YOUR RTL-SDR

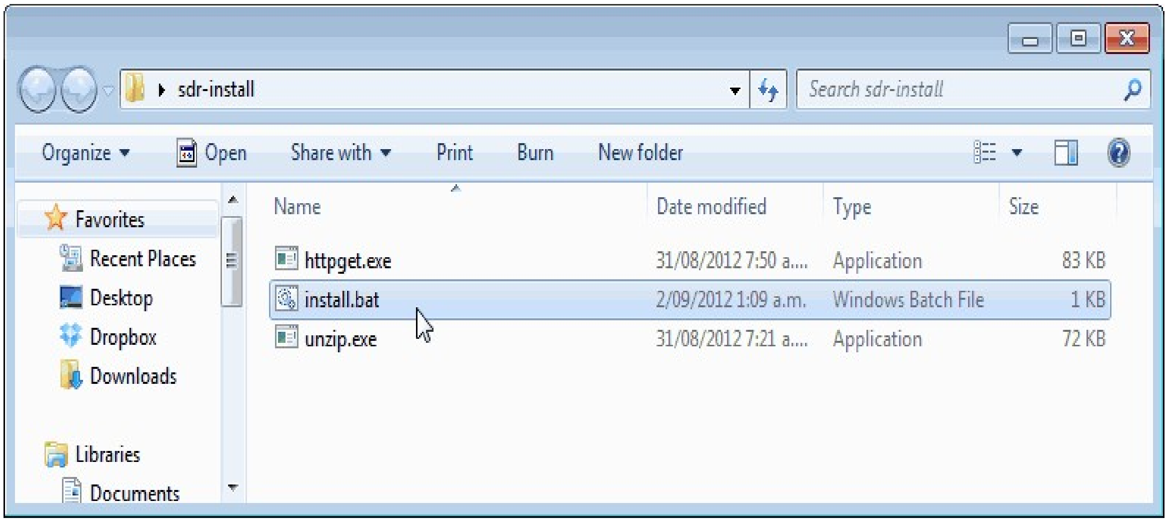
SDR# SETUP GUIDE (TESTED ON WINDOWS VISTA/7 + XP)

1. Purchase an RTL-SDR dongle. The cheapest and best for most applications is the R820T2 dongle. See http://www.rtl-sdr.com/buy-rtl-sdr-dvb-t-dongles/ for more information on the best places to buy one.

2. SDR# (pronounced SDR Sharp) is the easiest to use and most popular free SDR receiver software that is compatible with the RTL-SDR. Go to www.sdrsharp.com and go to the downloads page to download SDR#.
   Note that you must have the Microsoft .NET 3.5 redistributable installed to use SDR#. Modern Windows PCs usually have this installed by default, but older PCs running XP may need this to be installed. It can be downloaded from http://www.microsoft.com/en-gb/download/details.aspx?id=21.

3. Extract (unzip) the sdr-install folder from the zip file to a folder on your computer.

4. Double click on install.bat from within the extracted folder. This will start a command prompt that will download SDR# and all the files required to make SDR# work with the RTL-SDR. Everything will be placed into a new folder within the sdr-install folder called "sdrsharp". The command prompt will automatically close when it is done.

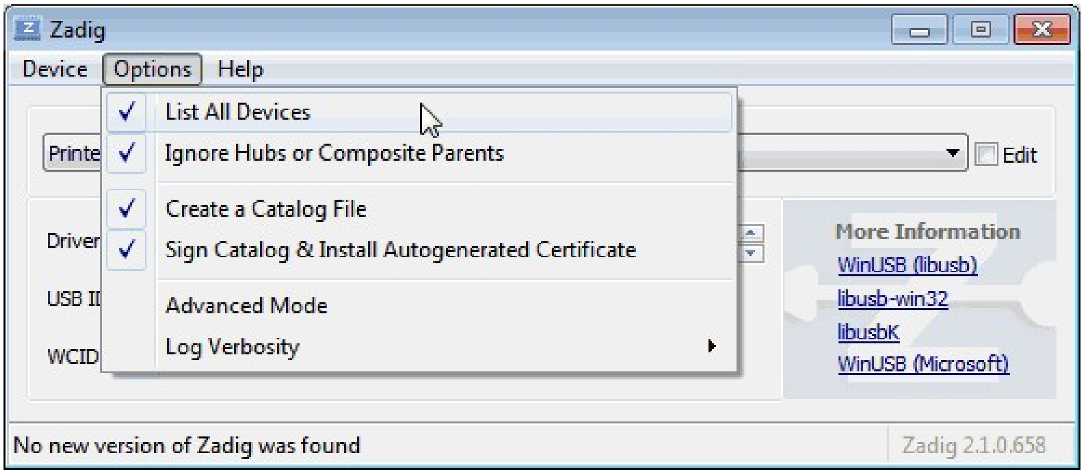


Name	Date modified	Type	Size
httpget.exe	31/08/2012 7:50 a....	Application	83 KB
install.bat	2/09/2012 1:09 a.m.	Windows Batch File	1 KB
unzip.exe	31/08/2012 7:21 a....	Application	72 KB

5. Plug in your dongle and do not install any of the software that it came with (if any), but ensure you wait for the plug and play popup that tries to install it to finish. If necessary, uninstall any DVB-T software drivers you've installed from the CD that some dongles come with.

6. Open the newly created sdrsharp folder and find the file zadig.exe. Right click this file and select "Run as administrator" if using Windows Vista/7. If you are using Windows XP, download the XP version of Zadig from http://zadig.akeo.ie/.

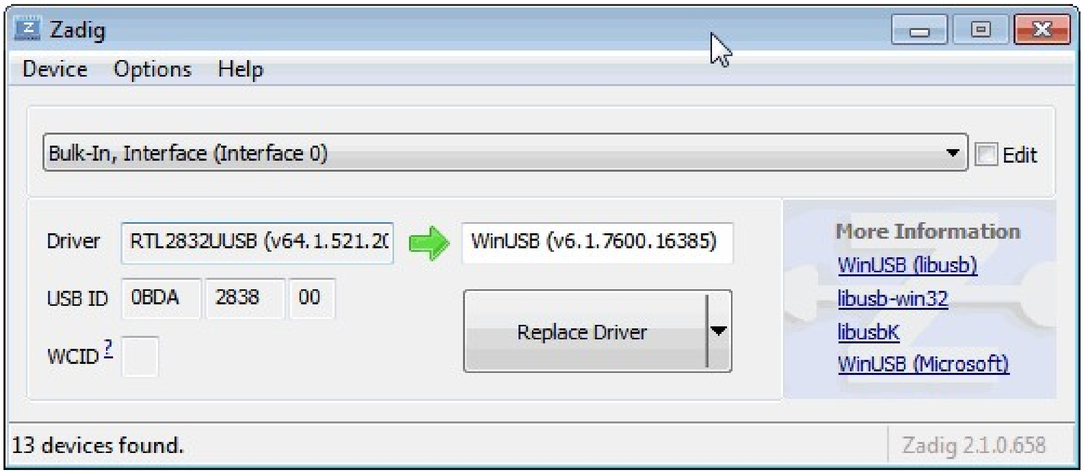
7. In Zadig, go to **Options -> List All Devices** and make sure this option is checked.



No new version of Zadig was found

8. Select **"Bulk-In, Interface (Interface 0)"** from the drop down list. Ensure that WinUSB is selected in the box next to where it says Target. (Note on

some PCs you may see something like RTL2832UHIDIR or RTL2832U instead of the bulk in interface. This is also a valid selection). (Do *not* select “Bulk-In, Interface (Interface 1)” or “USB Receiver (Interface 0)” however).



Bulk-In, Interface (Interface 0)

☐ Edit

Driver RTL2832UUSB (v64.1.521.2(→ WinUSB (v6.1.7600.16385)

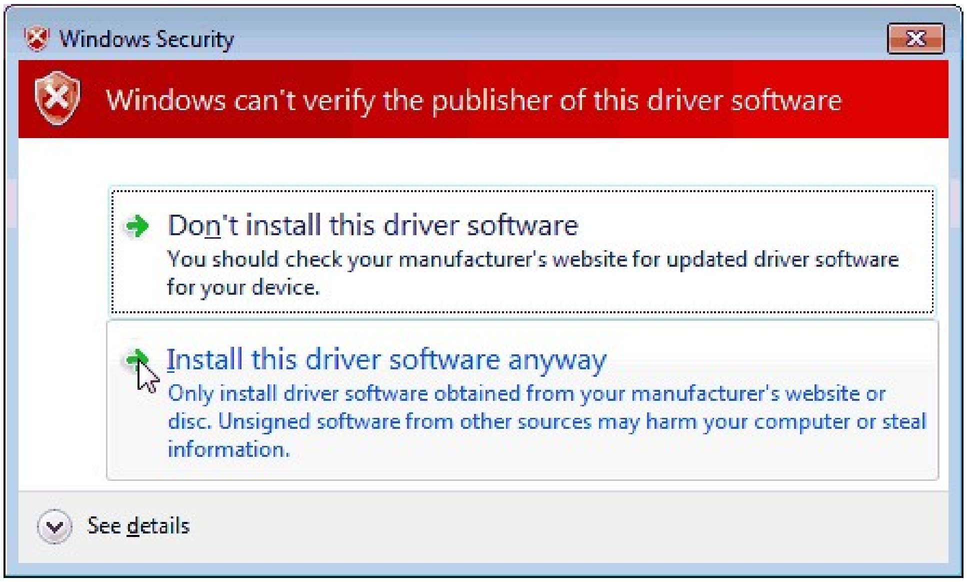
USB ID 0BDA 2838 00

WCID ?

13 devices found.

Zadig 2.1.0.658

9. Click Install (or Replace) Driver. You might get a warning saying that the publisher cannot be verified, but just accept it by clicking on Install this driver software anyway. This will install the drivers necessary to run the dongle as a software defined radio. Note that you *may* need to run zadig.exe again if you move the dongle to another USB port, or want to use two or more dongles together.



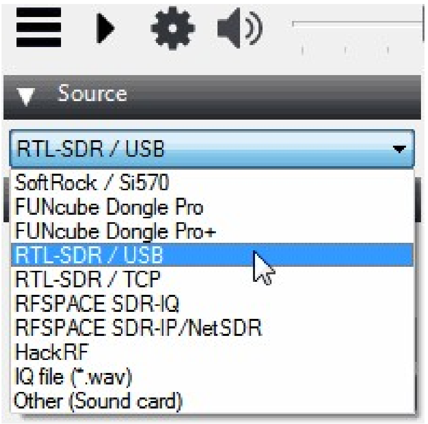
Windows can't verify the publisher of this driver software

Don't install this driver software
You should check your manufacturer's website for updated driver software for your device.

Install this driver software anyway
Only install driver software obtained from your manufacturer's website or disc. Unsigned software from other sources may harm your computer or steal information.

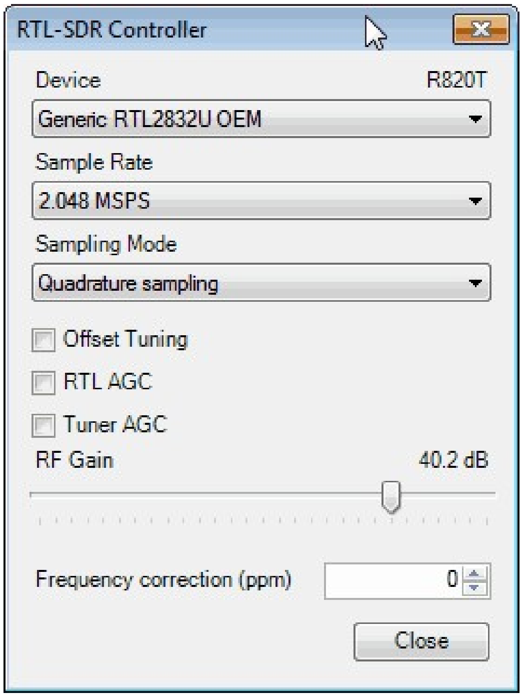
10. Open SDRSharp.exe which is in the same folder as Zadig. Set the drop down box at the top left under the Source partition to **‘RTLSDR / USB’**. Press the

Play button ▶. Your RTL-SDR software radio should now be playing some static and showing an RF spectrum and waterfall display! If everything has worked you should be able to start tuning to frequencies using the numbers at the top of the program.



11. Click on the Configure button ⚙ at the top of the SDR# window to bring up the configure menu.

12. By default the RTL-SDR's gain is set to zero. Increase this gain by moving the RF Gain slider up, or by enabling the Tuner AGC option.


RTL-SDR Controller

Device R820T

Generic RTL2832U OEM

Sample Rate

2.048 MSPS

Sampling Mode

Quadrature sampling

□ Offset Tuning

□ RTL AGC

□ Tuner AGC

RF Gain 40.2 dB

Frequency correction (ppm) 0

Congratulations! You're RTL-SDR should be set up and running now. To test it out we first suggest tuning to a broadcast FM radio station (the music stations you listen to in your car). In most countries these stations are between 88 MHz - 108 MHz. To listen to these stations be sure to select the WFM mode in the top left of SDR#. See the sections below for more information on the use of SDR# and its various settings.

TROUBLESHOOTING AND COMMON QUESTIONS GUIDE

I GET THE ERROR "NO COMPATIBLE DEVICES FOUND" WHEN TRYING TO START THE DONGLE IN SDR#

Long low quality USB extension cables can sometimes cause this error. Some USB 3.0 ports are also incompatible with the dongle and cause this error. One user has had luck with this error by installing zadig from safe mode. Finally, there is a small chance that the dongle is actually faulty. If the dongle produces the same error on multiple computers the dongle is probably faulty and should be refunded or replaced.

ZADIG TAKES A LONG TIME TO INSTALL THE DRIVER, THEN FAILS

You have probably not run zadig in administrator mode. Make sure to right click zadig, and select "Run as Administrator".

I DON'T SEE BULK-IN, INTERFACE (INTERFACE 0)

Ensure **Options->List All Devices** is checked. Some people report seeing something else other than the bulk in interface. If you installed the DVB-T drivers it may also show up as the brand of your dongle or something prefixed with "rtl". Those selections should work too. In rare cases you may receive a faulty dongle that will not show up in Zadig no matter what USB port or computer you try it on. You should ask for a replacement in this case.

I DON'T SEE RTL-SDR/USB IN SDRSHARP

You may have downloaded a version without RTL-SDR support. Check that you have downloaded SDR# from the official website.

USB 3.0 PORTS DON'T WORK

Unfortunately many USB 3.0 controllers are buggy and don't work with some devices. Generally, USB 3.0 works fine with the RTL-SDR, but there are some controllers that will just not recognise the dongle. In this case use a USB 2.0 port instead.

WHEN I RUN INSTALL.BAT A CMD/DOS WINDOW FLASHES ON THE SCREEN BRIEFLY THEN DISAPEARS. NOTHING IS DOWNLOADED OR INSTALLED.

There seems to be a bug or misconfiguration with some versions of Windows which causes any batch file to be unable to execute. One way around this is to install SDR# and the RTL-SDR drivers manually. For instructions on how to do this please see Appendix D: Misc, Manual Installation of SDR#.

ZADIG GIVES "SYSTEM POLICY HAS BEEN MODIFIED TO REJECT UNSIGNED DRIVERS" ERROR IN WINDOWS 8

Windows 8 can cause signed driver issues with zadig. Some users report getting the error: "System policy has been modified to reject unsigned drivers". To solve this download the newer Zadig 2.1+ drivers from http://zadig.akeo.ie/. Note the SDR# install.bat file now always downloads the latest Zadig.

RECEPTION IN SDR# SEEMS VERY POOR/RECEIVER IS INSENSITIVE

Make sure you have increased the RF gain slider which can be accessed by clicking on the configure button. Also, in poor reception areas using the stock antenna indoors may not be sufficient. Though rare, another cause of insensitivity is a dongle blown by electrostatic discharge. It is possible to receive a blown dongle from the factory. If this is the case the dongle should be returned for a replacement.

SDR# GIVES ERROR "APPLICATION FAILED TO INITIALIZE PROPERLY (0XC0000135). CLICK OK TO TERMINATE."

This might mean that you do not have the Microsoft .NET 3.5 Framework installed which is needed for SDR# to run. Download it from http://www.microsoft.com/en-gb/download/details.aspx?id=21.

SDR# GIVES ERROR "CANNOT ACCESS RTL DEVICE"

You may have tuned to a frequency that is out of range of the RTL-SDR. Tune back to a known supported frequency like 000.100.000.000 and press play.

I GET THE ERROR "1 COMPATIBLE DEVICES HAVE BEEN FOUND BUT ARE ALL BUSY"

On some computers the USB 3.0 ports can be buggy and will not work with the dongle. Use a USB 2.0 port instead. Also, this message sometimes occurs after the computer has been suspended. To fix it simply disconnect and reconnect the dongle. Sometimes it may also be necessary to reinstall the Zadig drivers.

WHEN RUNNING INSTALL.BAT I GET ERRORS LIKE "THE SYSTEM CANNOT FIND THE FILE SPECIFIED" AND THE SDRSHARP FOLDER IS NOT DOWNLOADED

This is probably because you did not unzip the files and are trying to run install.bat from within the zip file. Make sure you extract or move the unzip.exe, install.bat and httpget.exe files out of the zip file into another folder, before running install.bat.

THE DONGLE CONSTANTLY DISCONNECTS FROM THE USB PORT

First test to make sure that it is not the fault of a dodgy USB extension cable by plugging the dongle directly into the PC or into another high quality extension cable. If it still disconnects the dongle may be faulty and you should ask for a replacement.

THE DONGLE WON'T CONNECT TO THE PC AND ON MODELS WITH AN LED THE LED DOES NOT ILLUMINATE

The dongle is faulty and should be replaced.

ZADIG WON'T RUN

Some users report that Zadig crashes upon opening for them. An alternative driver installer can be found at http://visualgdb.com/UsbDriverTool/. Use this tool to install the libUSB - WinUSB drivers.

THE AUDIO SOUNDS CHOPPY AND THE SPECTRUM DISPLAYS ARE SLOW

Your PC may not be powerful enough to run SDR# at the settings you have chosen. Single core machines especially may have trouble. To reduce CPU load reduce the sample rate, reduce the FFT display resolution, disable the FFT spectrum analyzer and waterfall and reduce the filter order.

AUDIO SOUNDS CHOPPY OR DOESN'T WORK OVER WINDOWS RDP

Select directsound instead of MME in the audio out settings of SDR#.

WHY IS THERE IS A CONSTANT SPIKE IN THE MIDDLE OF THE SPECTRUM THAT WON'T GO AWAY?

This is normal and is due to the design of the RTL-SDR. If you have an R820T and are using SDR# the spike can be removed by enabling the "Correct IQ" checkbox. For an E4000 you can enable offset tuning in the SDR# configure menu.

WHAT IS BULK IN INTERFACE-1?

This is the infrared (IR) interface for the remote control which is not required when using the dongle as an SDR.

IT SEEMS THAT MY PC IS NOT POWERFUL ENOUGH TO RUN SDR# AS IT USES NEAR 100% CPU

For graphical GUI SDR software like SDR#, at least a dual core processor is recommended. If you have a borderline decent CPU and still experience high CPU usage, try reducing the sample rate to 1 Msps or less, reducing the FFT display resolution (or turning it off), turning off Correct IQ and reducing the filter order.

I USED ZADIG BUT IT BROKE MY KEYBOARD/MOUSE/OTHER DEVICE SOMEHOW

This is because you probably clicked the install button in Zadig after selecting the wrong device in the drop down menu. Make sure you select the correct RTL-SDR device, (Bulk-In Interface, Interface 0) first. Zadig overwrites previous drivers. To get the old drivers back you should be able to do so in Windows device manager, update driver software.

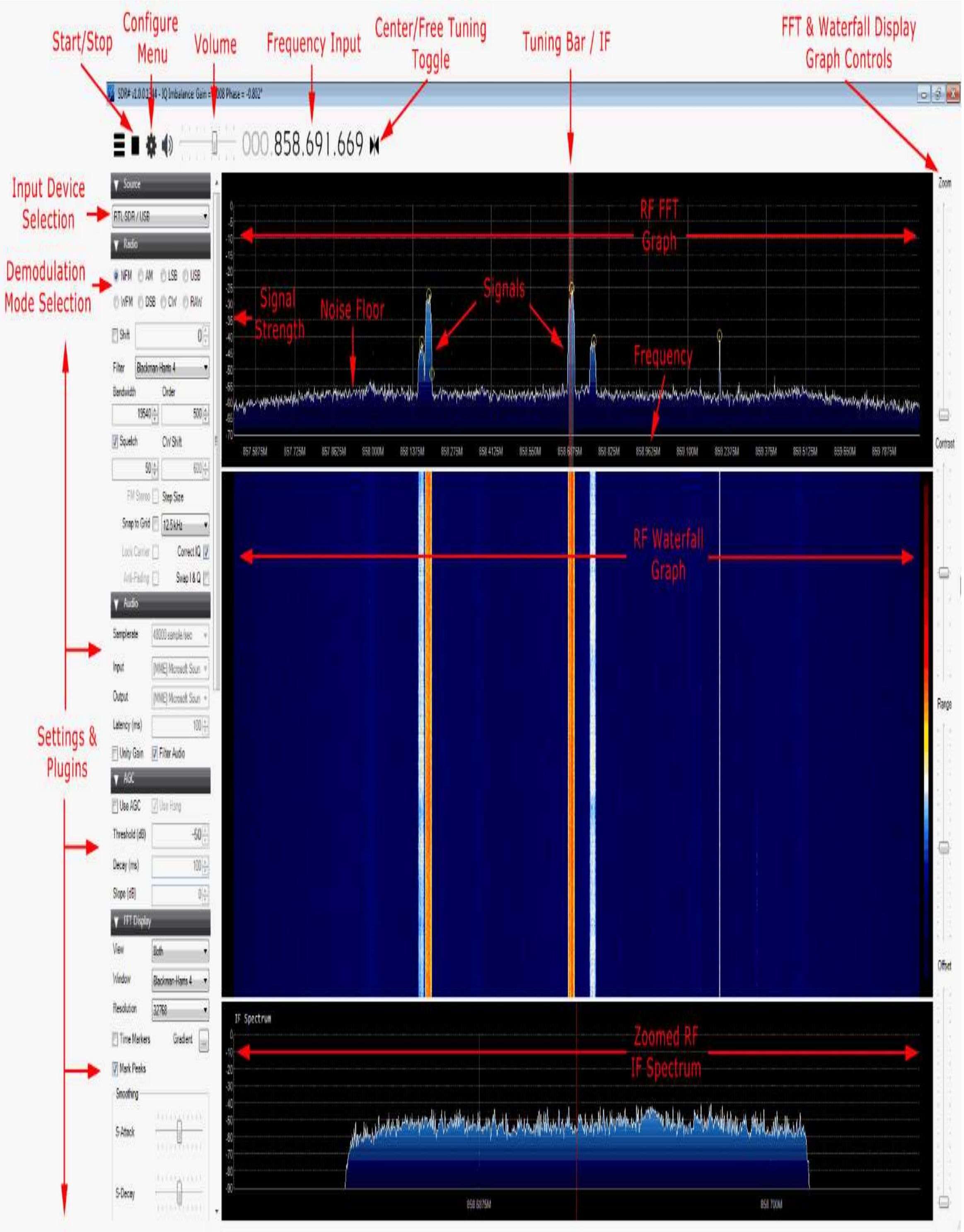
MY R820T2 RTL-SDR SHOWS UP AS AN R820T WHEN I RUN THE DIAGNOSTIC TOOL RTL_TEST

The R820T2 and R820T are identical electronically except for some minor changes in the maximum IF filter widths. The different filters may be the reason the R820T2 has better performance. Thus a R820T2 will show up as an R820T on the PC, since there is no distinction between them in the digital part of the circuit. You can confirm that you have a R820T2 by checking the markings on the chip.

SDR SHARP USERS GUIDE

SDR# is currently the most popular SDR program used with the RTL-SDR. It is easy to set up and simple to use. To install SDR#, go through the RTL-SDR setup guide shown above. Below we explain some of the settings and displays in SDR#.

Upon opening SDR# you will be greeted with this screen shown below. Here we have highlighted the main parts of SDR#.



SDR# v1.0.0.1344 - IQ Imbalance: Gain = 1.008 Phase = -0.802°

000.858.691.669

Source

RTL-SDR / USB

Radio

● NFM ○ AM ○ LSB ○ USB

○ WFM ○ DSB ○ CW ○ RAW

□ Shift 0

Filter Blackman-Harris 4

Bandwidth 19540 Order 500

☑ Squelch 50 CW Shift 600

FM Stereo □ Step Size

Snap to Grid □ 12.5 kHz

Lock Carrier □ Correct IQ ☑

Anti-Fading □ Swap I & Q □

Audio

Samplerate 48000 sample/sec

Input [MME] Microsoft Soun

Output [MME] Microsoft Soun

Latency (ms) 100

□ Unity Gain ☑ Filter Audio

AGC

□ Use AGC ☑ Use Hang

Threshold (dB) -50

Decay (ms) 100

Slope (dB) 0

FFT Display

View Both

Window Blackman-Harris 4

Resolution 32768

□ Time Markers Gradient

☑ Mark Peaks

Smoothing

S-Attack

S-Decay

After opening SDR# for the first time, we suggest you immediately remember to perform the following steps (if you don't know what some of these steps are, continue reading further below for more information):

1. Increase the RF gain from zero to a higher value in the configure menu.
2. Reduce the range slider on the right of the SDR# window to about -70.
3. Increase the FFT resolution to at least 16384 (ideally 32768 or higher if your computer is powerful enough) under the “FFT Display” heading on the left menu to give yourself a clearer higher resolution spectrum and waterfall image.
4. Enable the “Correct IQ” setting to remove the center spike if using an R820T/R820T2, or enable “Offset Tuning” in the configure menu if using an E4000/FC0012/13.
5. Turn off the “Snap to grid” setting, or adjust the PPM offset accordingly.

See below for more information on these settings and others.

MAIN SETTINGS AND WINDOWS

Note that there is a distinction in SDR# between settings that affect the software side, and settings that affect the hardware side. All hardware side settings can be found in the Configure Menu / RTL-SDR Controller window which can be accessed with the cog icon ⚙. In here are settings to control things like the RF gain and sample rate / bandwidth of the RTL-SDR. To optimise reception, you need to adjust settings in this window.

Most of the settings found in the main windows of SDR# affect the software digital signal processing (DSP) side of things. To optimise processing of the signal you need to adjust these DSP settings.

PLAY/STOP BUTTON

This button is used to start and stop the SDR.

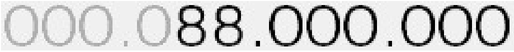
SOURCE

This is a drop down menu which is used to select the SDR input device being used. If you are using an RTL-SDR, select RTL-SDR/USB. Be sure to **NOT** select RTL-SDR/TCP unless you are using a remote server with rtl_tcp.

CONFIGURE MENU / RTL-SDR CONTROLLER

Clicking this button ⚙ opens up the configure menu. In here you can change settings like the sample rate (bandwidth) and RF gain. See further down for more information about these settings.

FREQUENCY INPUT



000.088.000.000

Use the mouse to set the desired frequency you wish to listen to here. You can either click on the tops and bottoms of each individual number to increase or decrease the value, or simply hover over the number you want to change and use the mouse wheel to alter the value.

The frequency input is divided into 4 sections with each section containing 3 values (e.g. 000.000.000.000). The first section represents GHz frequency values, the second MHz, the third kHz and the last Hz. For example to tune to a radio station at 88.6 MHz we would enter 000.088.600.000 into the frequency input. To tune to 861.5475 MHz we would enter 000.861.547.500. To tune to 1.575 GHz we would enter 001.575.000.000. To tune to 500 kHz (with an upconverter and appropriate offset shift set (discussed later)) we would tune to 000.000.500.000.

VOLUME / AF GAIN

Set the volume level of your output speakers or audio piping device here.

RF SPECTRUM / FFT DISPLAY

This part of the window shows the RF spectrum as a graph in real time visually. Active signals will appear as peaks on this graph.

RF WATERFALL

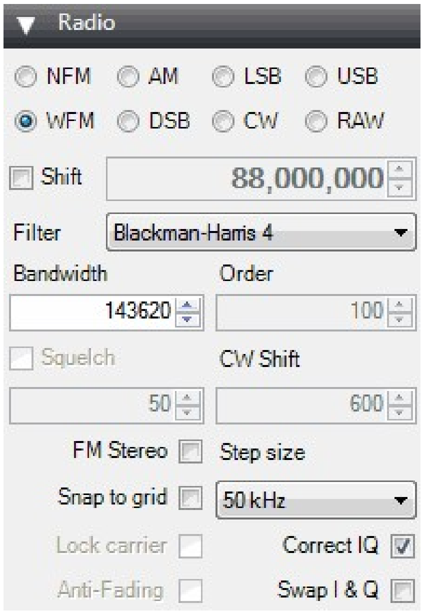
This part of the window shows the RF spectrum graph spread over time with new data at the top and old data at the bottom, just like a waterfall.

TUNING BAR

The vertical red line in the RF spectrum graph shows where on the RF spectrum the RTL-SDR is currently tuned to. Within the currently active chunk of instantaenous bandwidth the tuning can be altered by simply using the mouse to click and drag the red line, or just by clicking elsewhere in the RF spectrum.

The shaded rectangular area around the red line shows the bandwidth of the tuned area (don’t confuse this with the bandwidth/sample rate that is set in the configure menu). The bandwidth should be set so that it covers the area of the signal that is tuned. The bandwidth can be adjusted by using the mouse to simply drag the edges of the shaded area in or out.

RADIO TAB



▼ Radio

○ NFM ○ AM ○ LSB ○ USB

● WFM ○ DSB ○ CW ○ RAW

□ Shift 88,000,000

Filter: Blackman-Harris 4

Bandwidth: 143620 | Order: 100

□ Squelch: 50 | CW Shift: 600

FM Stereo □ | Step size: 50 kHz

Snap to grid □

Lock carrier □ | Correct IQ ☑

Anti-Fading □ | Swap I & Q □

Here you can choose what type of demodulation mode the signal at your currently tuned frequency should use.

NFM - Narrowband Frequency Modulation. Commonly used mode used by walkie talkie radios, weather radio and most VHF/UHF digital signals.

WFM - Wideband Frequency Modulation. This is the mode that broadcast FM stations use (e.g. radio music stations).

AM - Amplitude Modulation. Used by broadcast AM stations that are receivable by normal shortwave radios and also used by air band voice frequencies used by aircraft and air traffic control. Some digital signals also use AM.

LSB/USB - Lower Side Band/Upper Sideband. Used in the HF band by ham radio users to transmit voice and data efficiently with small bandwidths.

CW - Continuous Wave. Used for listening to morse code.

DSB - Double Side Band. Not commonly used in normal operation.

RAW - Raw IQ signal. Almost never used.

SHIFT

This box offsets the tuned frequency by the amount entered. This is useful if you are using an upconverter. For example if you have an upconverter with a 100 MHz oscillator, you would set the shift to be -100,000,000 (don't forget the minus sign). Without the shift, when using an upconverter to tune to a signal at 9 MHz you would need to actually tune to 100 + 9 = 109 MHz. With the shift set, you can tune to 9 MHz as normal. If you have an upconverter with a 125 MHz oscillator you would tune to 125 + 9 = 134 MHz, or set the shift to -125,000,000.

BANDWIDTH

Recommended Default Setting: NFM/AM: 12500, WFM: 250000

This is the width of the shaded part of the tunable area. You can set it manually here, or by dragging the edges with the mouse as described under the tuning bar description.

FILTER

Recommended Default Setting: Blackman-Harris 4

Changes the filter type used. Different filters have different shapes. The filter is used to select the highlighted signal in the RF window. Blackman-Harris is usually the best filter to choose and this setting almost never needs to be changed.

ORDER

Recommended Default Setting: 500

Changes the filter order. You may notice when using low filter orders that signals outside of the tuned bandwidth can still be heard. Larger filter orders “tighten” or “sharpen” the band pass filter used within the tuned bandwidth thus preventing signals outside of the tuned bandwidth from being heard. You will want to increase the filter order when there are strong signals near to your tuned area. Using higher filter orders can cause a greater load on the CPU, so slow PCs may need to reduce this value. Usually there is no need to adjust the order, but see the setting the filters section for more information about situations when you might want to adjust it.

SQUELCH

Recommended Default Setting: OFF

Squelch is used to mute the audio when the signal strength is below the specified value. A larger value requires a stronger signal to unmute the audio. It is useful for when listening to speech as the sound of static when no one is talking will be muted. Note that the value of the squelch does not correspond to the dB signal strength values used in the RF spectrum display. You will simply need to experiment with different squelch values until you get the desired result.

CW SHIFT

Recommended Default Setting: 600

Mainly useful for when receiving CW (morse code) as it specifies the offset between CW transmit and receive frequencies.

FM STEREO

Recommended Default Setting: OFF

Will enable stereo output for broadcast radio WFM signals.

SNAP TO GRID / STEP SIZE

Recommended Default Setting: OFF

In many bands frequencies are allocated at a fixed distance apart. For instance in most countries air band signals are spaced 25 kHz apart (or 8.33 kHz in some countries). Turning on snap to grid can help with tuning by causing the tuning bar to snap directly to a signal. However, to use this with the RTL-SDR the PPM frequency offset correction must be set correctly, otherwise the frequencies may not line up. The snap to step size can be set in the “Step Size” pull down menu.

CORRECT IQ

Recommended Default Setting: ON

Should usually be selected as ON. This setting removes the small but annoying centre spike that is present with R820T/R820T2 RTL-SDR dongles.

LOCK CARRIER

Recommended Default Setting: OFF

Only active in AM or DSB mode. Allows for synchronous AM reception which can significantly improve reception and also automatically centers the signal. The centering is not shown in the main RF spectrum but can its effect can be seen in the Zoom FFT IF Spectrum window. Turn this on for better AM reception, but may increase CPU usage.

ANTI-FADING

Recommended Default Setting: OFF

Can be used when “Lock Carrier” is activated. Takes advantage of the symmetry of AM signals which helps with weak signals when they may be fading in and out. Turn this on for better AM reception, but may increase CPU usage.

SWAP I & Q

Recommended Default Setting: OFF

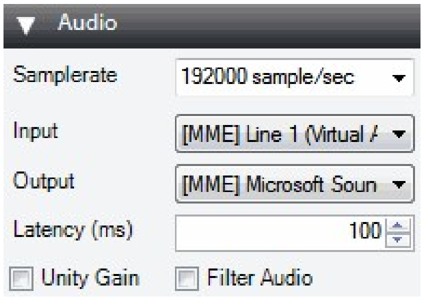
If you are using SDR# as a panadapter, some hardware radios may have the I & Q signals swapped and need this option checked.

MARK PEAKS

Recommended Default Setting: OFF

Simply marks any peak in the RF spectrum with a circle.

AUDIO TAB



▼ Audio

Samplerate 192000 sample/sec

Input [MME] Line 1 (Virtual /

Output [MME] Microsoft Soun

Latency (ms) 100

□ Unity Gain □ Filter Audio

SAMPLE RATE

Recommended Default Setting: 48000

Sets the sample rate of your sound card. Some decoding software may require a specific sample rate to be set. Usually the default value should be fine for general listening. This can normally only be set before pressing the play button for the first time.

INPUT

Recommended Default Setting: ANYTHING

Specifies the input sound card when using SDR# with the “Other (Sound Card)” source. Use mainly with sound card based software defined radios. In normal use with an RTL-SDR this does not need to be set.

OUTPUT

Recommended Default Setting: Your sound card

Sets the audio output device. By default it is set to your speakers. If you are passing the audio to a decoder program here you would choose your virtual audio pipe (VAC/VB Cable) to send the audio to.

UNITY GAIN

Recommended Default Setting: OFF

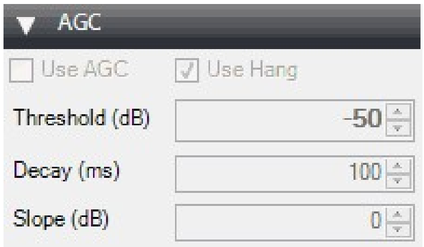
Should normally be unchecked as it sets the audio gain to 0 dB.

FILTER AUDIO

Recommended Default Setting: OFF

Improves voice signals by low pass filtering the audio, removing high pitched hiss and interference. Turn this off if decoding digital signals.

AGC (AUTOMATIC GAIN CONTROL) TAB



▼ AGC

□ Use AGC ☑ Use Hang

Threshold (dB) -50

Decay (ms) 100

Slope (dB) 0

Note that in some modes the AGC tab will be greyed out.

USE AGC

Recommended Default Setting: ON

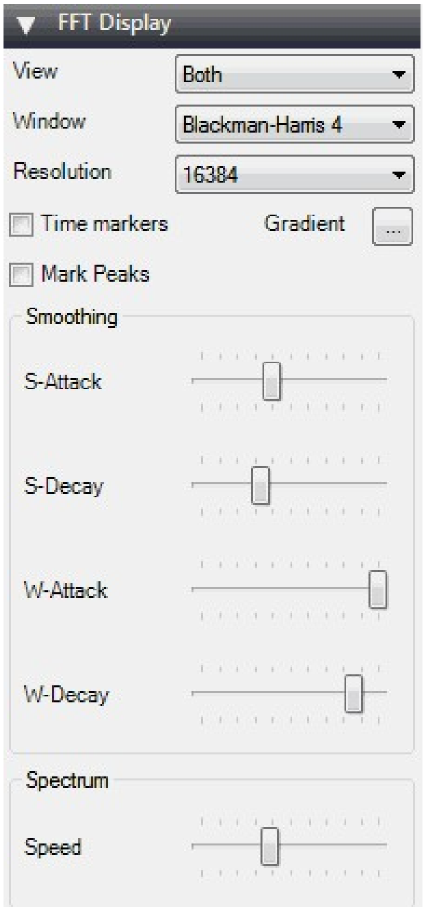
Turns on the automatic gain control. The AGC will attempt control the audio volume level so that loud sounds are not too loud and quiet sounds are not too quiet. The default settings work well for voice audio signals. It is especially useful to turn this on when listening to AM/USB/LSB signals as strong signals in these modes may sound distorted otherwise.

USE HANG / THRESHOLD / DECAY / SLOPE

Recommended Default Setting: -50, 100, 0

Allows you to modify the default AGC behaviour, though in most cases the defaults are fine.

FFT DISPLAY



FFT Display

View: Both

Window: Blackman-Harris 4

Resolution: 16384

□ Time markers Gradient ...

□ Mark Peaks

VIEW

Recommended Default Setting: Both

Set it to view both the RF spectrum and the waterfall, or only one of them, or none at all. Removing the waterfall may be useful on older PCs with slow processing hardware.

WINDOW

Recommended Default Setting: Blackman-Harris 4

Sets the type of filtering algorithm to use on the FFT, the default of Blackman-Harris 4 is the best in most cases.

RESOLUTION

Recommended Default Setting: 32768

Increasing the resolution will increase the quality of how the signal looks in the RF display and waterfall. Using a higher resolution may be useful when fine tuning, as high resolutions will allow you to see the peaks and structure of a signal much more clearly. Beware that high resolutions can slow your PC down and can cause trouble especially with single core machines. Generally, a value of at least 32768 should be used if your PC can handle it.

TIME MARKERS

Recommended Default Setting: OFF

Adds time markers on the waterfall display, so you know at what time a particular signal was broadcasting.

GRADIENT

Allows you to customize the colors used in the waterfall display.

MARK PEAKS

Recommended Default Setting: OFF

Adds a circular marker on every signal peak on the RF spectrum.

S-ATTACK / S-DECAY

Changes the amount of smoothing and averaging done in the RF spectrum display.

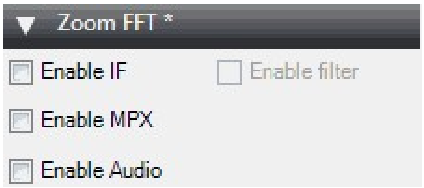
W-ATTACK / W-DECAY

Changes the amount of smoothing and averaging done in the waterfall display.

SPEED

Changes how fast the RF spectrum and waterfall updates.

ZOOM FFT



▼ Zoom FFT *

□ Enable IF □ Enable filter

□ Enable MPX

□ Enable Audio

Zoom FFT is a plugin that comes by default with SDR#. It creates a zoomed in RF spectrum display of the tuned IF bandwidth at the bottom of SDR#.

ENABLE IF

Creates a “zoomed in” RF spectrum around the area of your tuned IF bandwidth. Allows you to see the signal structure with much greater resolution.

ENABLE FILTER

If Enable IF is checked, then you can enable a special adjustable IF filter. This filter allows you to filter the left and right side of the tuned IF bandwidth individually. See the setting the filters section below more more information on its use.

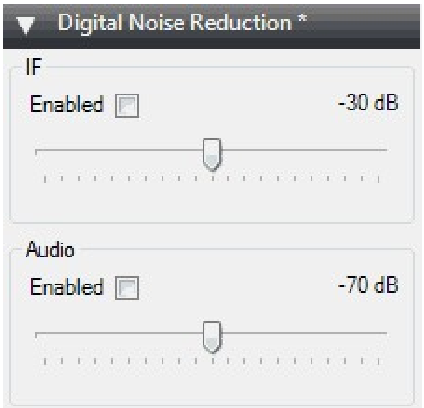
ENABLE MPX

Enables you to see the MPX spectrum of a broadcast FM radio station. Broadcast FM is encoded in a special baseband audio format called MPX. It contains a mono section, a pilot tone and a stereo section, as well as sometimes subcarrier sections for data like RDS and special radio services like SCA. If you were to try and view the audio baseband with the "Enable Audio" button, you wouldn't see the MPX structure because SDR# would have processed it into normal audio, and discarded the subcarriers and other sections.

ENABLE AUDIO

Allows you to see the audio (baseband) spectrum.

DIGITAL NOISE REDUCTION (DNR)



▼ Digital Noise Reduction *

IF

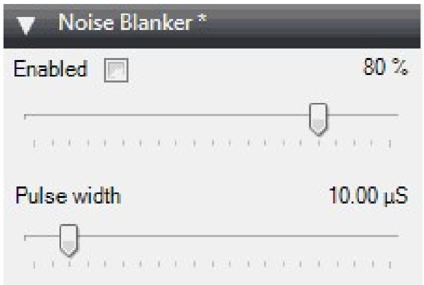
Enabled □ -30 dB

Audio

Enabled □ -70 dB

It is useful to turn on digital noise reduction when listening to noisy analogue voice signals. This setting will attempt to reduce the background 'hiss' sound. There are two DNR options available, IF and Audio. The IF uses the noise reuduction algorithm on the IF signal and the Audio option does it on the output audio signal. Usually the IF digital noise reduction works best and should be tried first, though a combination of both may work best. The sliders control the strength of the algorithm applied.

NOISE BLANKER



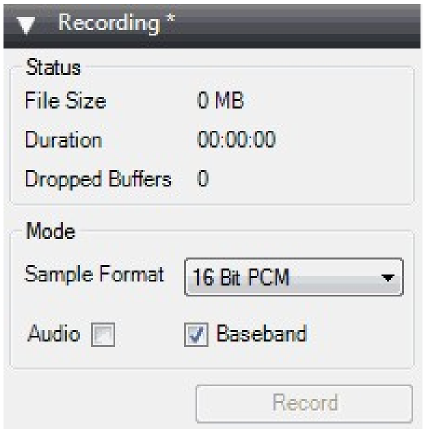
▼ Noise Blanker *

Enabled □ 80 %

Pulse width 10.00 µS

The noise blanker is an algorithm that can be turned on to help reduce impulsive and pulsing like noise from sources like spark gaps. Examples of this type of noise may come from motors, electricity lines and switching power supplies. Using this option can really help when receiving a weak signal amongst noise.

RECORDING TAB



File Size 0 MB
Duration 00:00:00
Dropped Buffers 0
Sample Format 16 Bit PCM
Audio □ ☑ Baseband

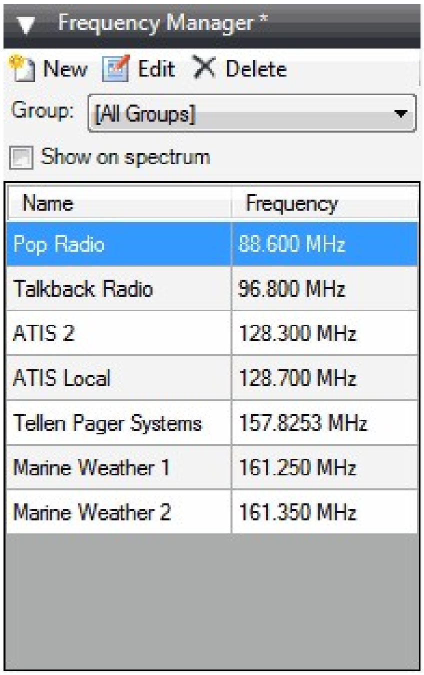
The recording tab allows you to make I/Q and Audio recordings. The sample format allows you to choose the level of recording quality needed. Since the RTL-SDR is only about 8-bits, we can select the lowest 8 Bit PCM option. Using only 8-bits saves a significant amount of disk space.

An I/Q recording is a recording of the entire bandwidth you are currently tuned to. It saves data in that bandwidth so you can replay it at a later time. An I/Q recording can be made by checking the “Baseband” check box. Note that I/Q recordings can use up a lot of disk space, so make sure to watch the File Size and Duration status counters. I/Q recordings can be played back in SDR# by selecting IQ File (*.wav) from the source menu. If you receive a lot of dropped buffers, then your PC or disk may not be fast enough to process the recording.

The audio coming out of the speakers can be recorded by checking the “Audio” check box. This will record audio to a wav file.

All recorded files are stored in the same folder as the SDR# executable.

FREQUENCY MANAGER TAB



Group: [All Groups]

□ Show on spectrum

Name	Frequency
Pop Radio	88.600 MHz
Talkback Radio	96.800 MHz
ATIS 2	128.300 MHz
ATIS Local	128.700 MHz
Tellen Pager Systems	157.8253 MHz
Marine Weather 1	161.250 MHz
Marine Weather 2	161.350 MHz

The frequency manager allows you to save any frequencies of interest in a database. A new frequency can be added to the database by clickong in the New button. This will add a frequency with the current tuned frequency and settings like bandwidth. You can edit the frequencies name and put it into a group for easy management.

If you check the “Show on spectrum” check box then your saved frequencies will be displayed in the RF spectrum.

RIGHT HAND SIDE SLIDERS

ZOOM

Recommended Default Setting: Zoomed out

Moving this slider will cause the RF spectrum and waterfall to zoom in on your tuned IF bandwidth area in order to see a signal closeup. However, the more you zoom in, the lower the resolution will seem. An alternative to zooming is to either reduce the sample rate, or to use the special decimation drivers which are discussed later in the Optimizing Tuning section. These alternative methods will preserve the visual resolution and allow you to see the signal structure much more clearly.

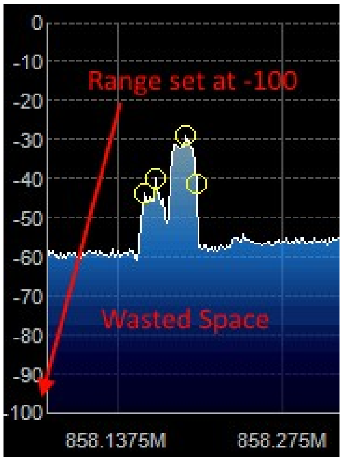
CONTRAST

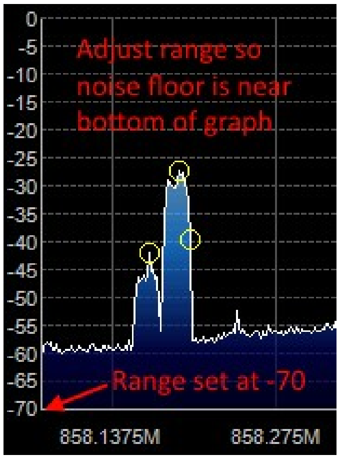
Adjusts the contrast of the waterfall. Adjust it so that signals clearly stand out from the background noise.

RANGE

Recommended Default Setting: -70

Modifies the dB level range shown on the left (vertical) axis of the RF spectrum window. You should adjust this so that the noise floor sits near the bottom of the RF spectrum window. This will allow signals to be more visible in the FFT RF spectrum and waterfall displays. As the RTL-SDR has a dynamic range of approximately 50 dB (plus a little more after oversampling/decimation), you will not need a range much higher than 0 to -70 dB. This setting will also affect the contrast in the waterfall and may help make weak signals easier to spot.

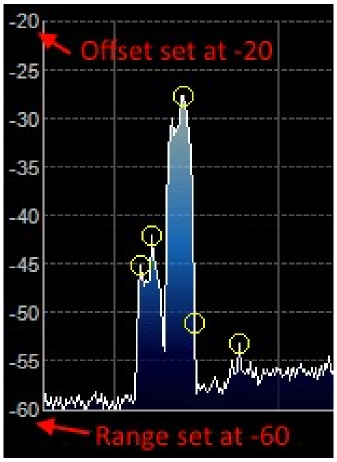




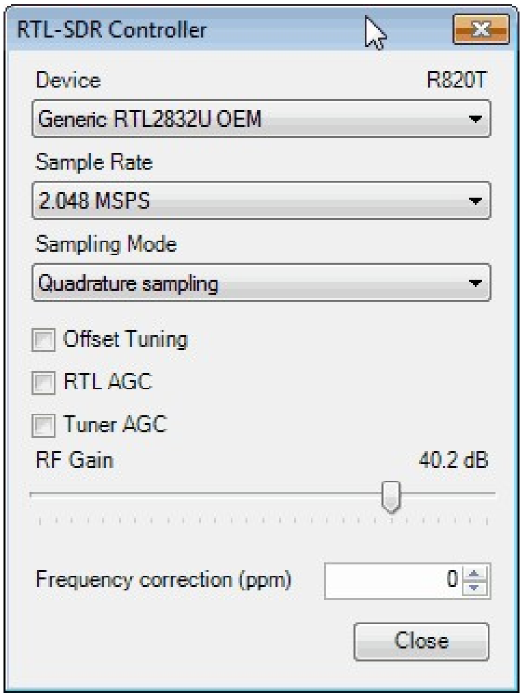
OFFSET

Recommended Default Setting: 0

Adds an offset to the dB level range in the RF spectrum window. The offset is added to the top value on the dB level range in the RF spectrum. Usually there is no need to adjust this, but if you want to get really good contrast on weak signals, adjust this along with the range so that the signal height is the same height as the vertical axis. This setting will also affect the contrast in the waterfall and may help make weak signals easier to spot.



CONFIGURE WINDOW



RTL-SDR Controller

Device R820T

Generic RTL2832U OEM

Sample Rate

2.048 MSPS

Sampling Mode

Quadrature sampling

□ Offset Tuning

□ RTL AGC

□ Tuner AGC

RF Gain 40.2 dB

Frequency correction (ppm) 0

The SDR# configure window can be accessed by clicking on the cog icon ⚙ at the top of the main SDR# window. The settings in this window affect the actual RF performance of the RTL-SDR, and should be set correctly to optimise the signal to noise ratio (SNR). The configure window has several options which are described below.

DEVICE

If you have multiple RTL-SDR dongles plugged in, the device drop down menu allows you to choose betwen them.

SAMPLE RATE

Recommended Default Setting: 2.048 or 2.4 MSPS

Lets you choose the size of the instantaenous bandwidth the RTL-SDR should display. Generally settings of up to 2.8 MSPS work well on most PCs, but if you have a slow PC you may want to reduce this. A larger sample rate may improve reception on weak signals as this allows for more oversampling. We recommend a default rate of 2.048 or 2.4 MSPS.

SAMPLING MODE

Recommended Default Setting: Quadrature Sampling

Use Quadrature sampling for normal operation. The direct sampling selections should be used when using a modded device in the direct sampling mode. See the direct sampling section for more information.

OFFSET TUNING

Recommended Default Setting: OFF IF R820T/2, ON IF E4000/FC0012/13

Only useful for the E4000/FC0012/13 tuners. Selecting this will get rid of the large spike in the center of the spectrum that is present with the E4000/FC0012/13 zero IF tuners.

RTL AGC

Recommended Default Setting: OFF

Enables the automatic gain control system on the RTL2832U chip. This is normally not useful as selecting this usualy degrades reception.

TUNER AGC

Recommended Default Setting: OFF

Enables the automatic gain control system on the tuner chip. Can be useful for general browsing, but it is almost always better to set the gain manually.

RF GAIN

Recommended Default Setting: Adjust for best performance

This slider can be used to set the tuner RF gain manually. Will not be active if Tuner AGC is checked.

FREQUENCY CORRECTION (PPM)

Recommended Default Setting: Your dongles unique PPM offset

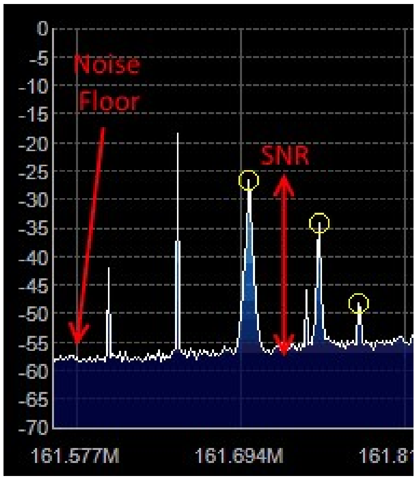
Allows you to correct the frequency offset that RTL-SDRs have from having low quality crystal oscillators. See the setting the PPM correction section further down for more information.

SETTING THE RF GAIN

There are three RF gain settings that can be found by clicking on the Configure button. RTL AGC turns on the RTL2832U chips internal automatic gain control (AGC) algorithm. Tuner AGC enables the RTL-SDR tuners AGC. The AGC’s attempt to automatically optimize the SNR of the signals. Finally, the gain slider can be used to manually set the gain.

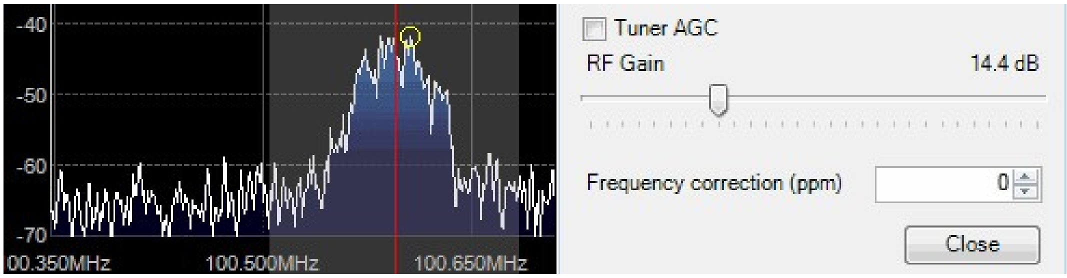
The AGCs used in the RTL-SDR are designed to be used with wideband DVB-T signals, and do not work very well with narrowband signals. We reccomend using manual gain control to optimize the gain of a signal, however for casual browsing turning on Tuner AGC may suffice. RTL AGC is almost never used as it tends to just introduce a lot of unwanted noise.

The goal when setting the RF gain manually is to try and get the signal to noise ratio (SNR) as high as possible. This means that the maximum signal strength should be high, but the noise floor should also be as low as possible.



When increasing the gain, there will come a point at which the noise floor begins to rise faster than the signal strength rises. This is the point at which you should stop increasing the gain.

You can calculate the signal to noise ratio by subtracting the peak signal height from the height of the noise floor. Below we show an example of the effect that increasing the RF gain has on a signals SNR.

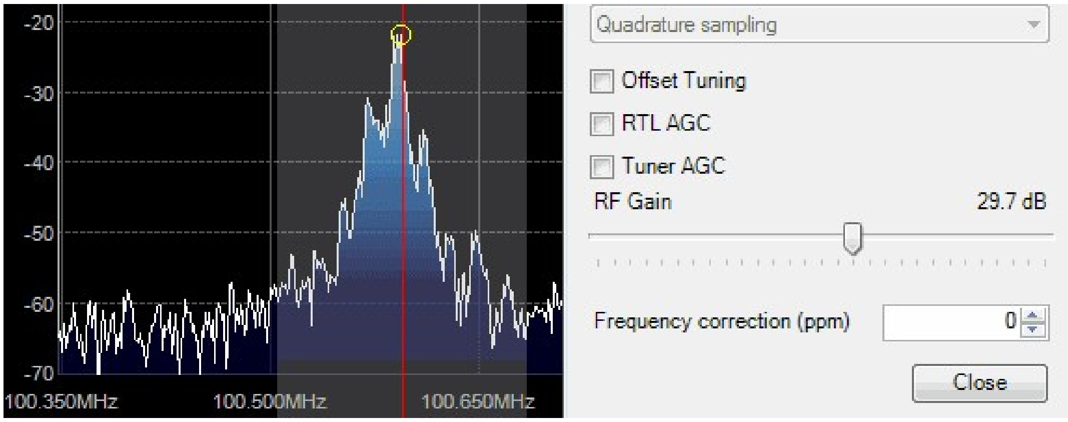


□ Tuner AGC

RF Gain 14.4 dB

Frequency correction (ppm) 0

SNR: -41 - -65 = 24 dB



Quadrature sampling

□ Offset Tuning

□ RTL AGC

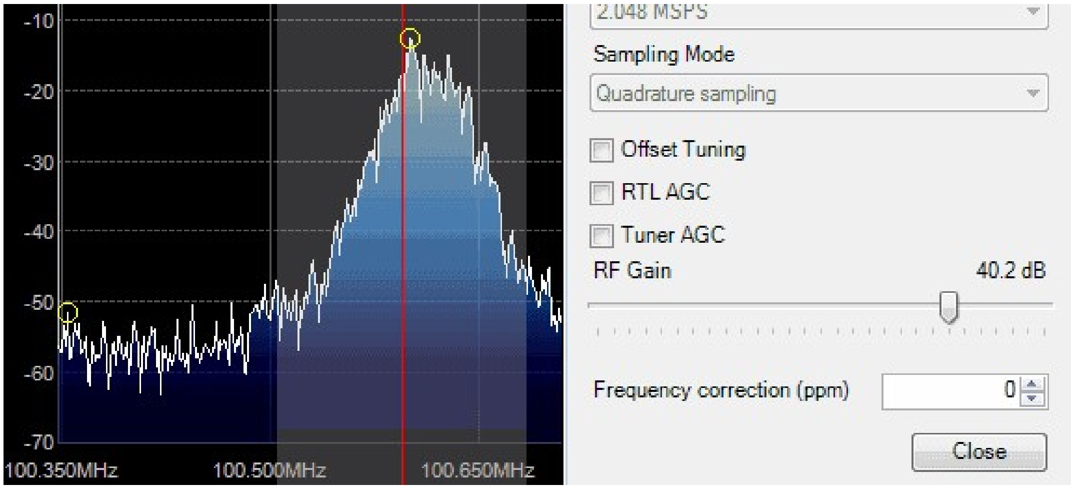
□ Tuner AGC

RF Gain 29.7 dB

Frequency correction (ppm) 0

Close

SNR: -21 - -65 = 44 dB



2.048 MSPS

Sampling Mode

Quadrature sampling

□ Offset Tuning

□ RTL AGC

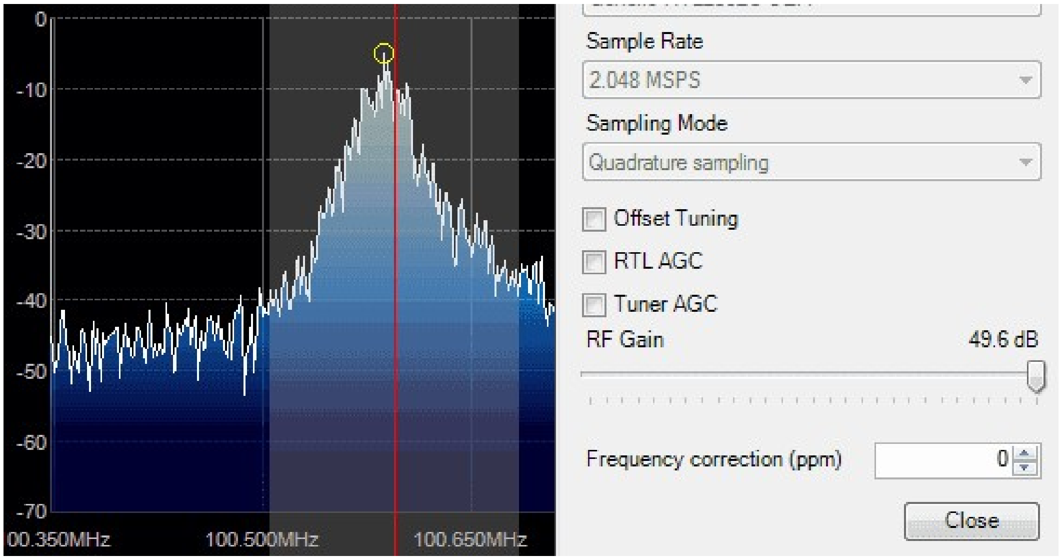
□ Tuner AGC

RF Gain 40.2 dB

Frequency correction (ppm) 0

Close

SNR: -12 - -57 = 45 dB



Sample Rate

2.048 MSPS

Sampling Mode

Quadrature sampling

□ Offset Tuning

□ RTL AGC

□ Tuner AGC

RF Gain 49.6 dB

Frequency correction (ppm) 0

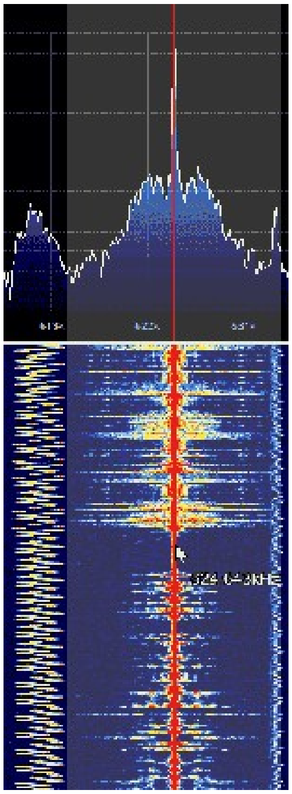
Close

SNR: -5 - -45 = 40 dB

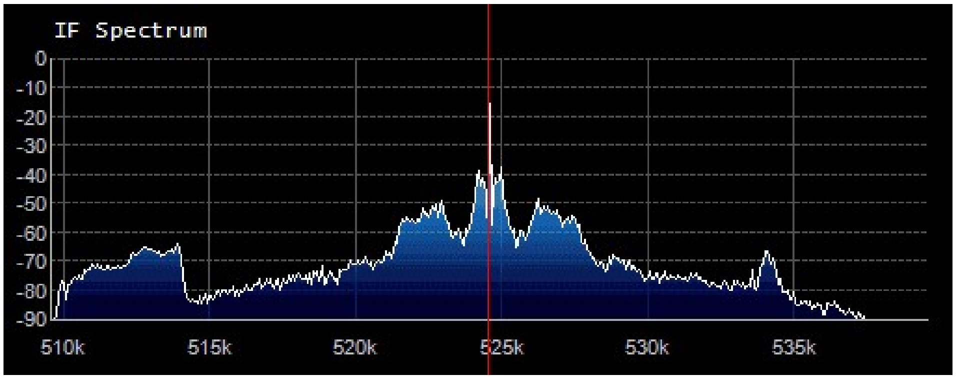
From the above images we see that as we raise the gain the SNR increases, but if we raise it too much the SNR actually begins to decrease. For this signal the gain was optimal at around 40.2 dB which gave us a SNR of 45 dB.

SETTING THE FILTERS

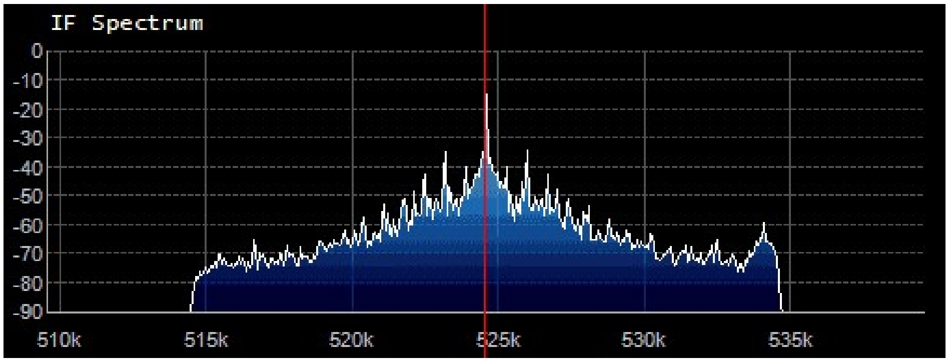
Usually adjusting the filters is not required. However, sometimes there may be a nearby interfering signal which can be tuned out with software filtering. We can see the effect that the filter order has using the Zoom FFT IF Spectrum plugin which is supplied with the standard SDR#. To activate this plugin, simply click on the “Enable IF” box in the left menu under the “Zoom FFT” heading. The Zoom FFT spectrum shows the signal in the tuned area after the band pass filtering. You can use it to investigate the effects of different filter orders and filter types. For example, here we are tuned to an AM radio signal. Notice how just to the left of this particular AM signal is another signal which appears to be some sort of radar and to the right is another unknown source of interference.



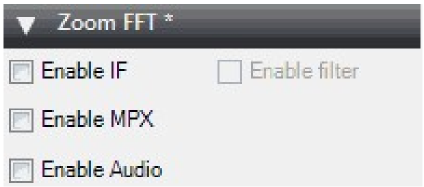
With the filter order set to 10 the radar signal can be heard in the audio and seen in the left of the Zoom FFT IF spectrum plugin window, even though it it outside of the tuned bandwidth.



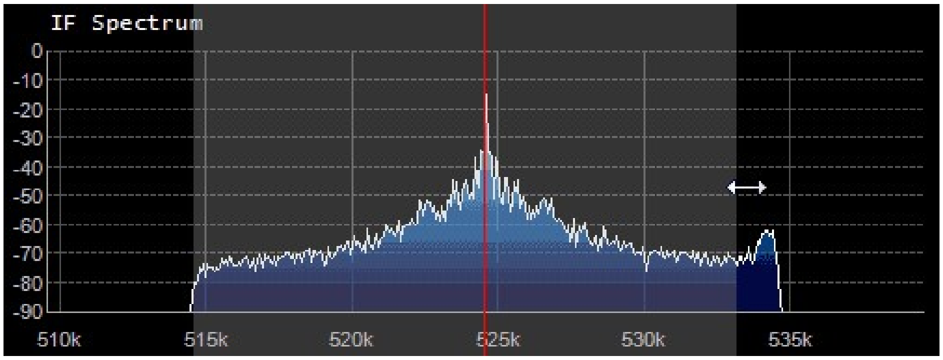
By increasing the filter order to 500, the filter 'sharpens' and now signals outside of the tuned bandwidth are blocked out.



In this AM radio signal there is also an interferer on the inside of the tuned bandwidth on the right of the signal. This interferer can be heard in the output audio as a subtle hissing sound. Newer versions of SDR# have a secondary filter which can be turned on in the Zoom FFT plugin in the left menu by checking the box labelled 'Enable Filter'.



The secondary filter can be used to help tune out interfering signals within the tuned area by altering the bandwidth for each side independently. To do this simply click and drag on the filter edge on one side of the Zoom FFT spectrum window and drag the filter in or out. Note that using this secondary filter will also increase CPU usage.



SETTING THE PPM CORRECTION

PPM is a term that stands for parts per million and is a measure that can be used to determine the accuracy of a crystal osccillator. The crystal oscillator is the “clock” or “heartbeat” of a digital circuit. High quality crystal oscillators have a low PPM value.

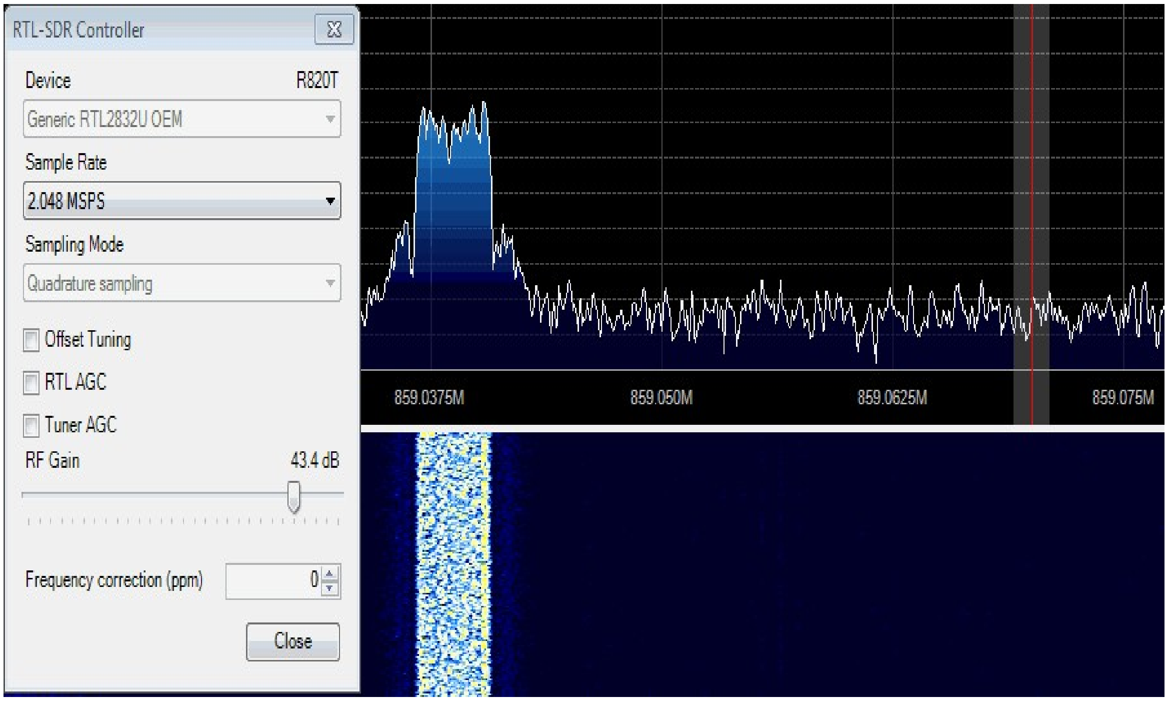
As the RTL-SDR dongles are mass produced with low quality 28.8 MHz crystal oscillators, the tuned frequency can be out by up to +-150 PPM or even more. Some RTL-SDRs now come with improved crystals with lower PPM tolerances. Ones that come with a surface mount (SMD) crystal will usually have a tolerance of +-30 PPM. Ones that come with a TCXO oscillator will usually have a tolerance of +-1 PPM or less, and normally won’t require any frequency correction to be set.

A large tolerance can mean that a known frequency that you are tuning to may not be exactly at the frequency that you expect it to be at. You may find that the frequency will always be off by a few kilohertz which can be very frustrating for narrow band signals. At small frequencies (HF), the frequency offset won’t be so visible, however at VHF and UHF frequencies the frequency offset will be larger. PPM works like this: at 1 kHz a 1 PPM offset will only be 0.001 Hz. However at 1 MHz the offset will be 1 Hz, at 10 MHz the offset will be 10 Hz, at 100 MHz the offset will be 100 Hz and so on. So when using an RTL-SDR with large PPM offset, signals you tune to in the VHF/UHF ranges may not be where you expect them to be. This linear increase in frequency offset as the tuned frequency increases is also why you cannot correct the crystal offet with the “Shift” box, which is instead used for setting upconverter offsets.

Fortunately, SDR# has an option to correct the PPM frequency offset that you may encounter. To set the PPM correction, you will first need to know the true frequency of a known radio signal, preferably a constant narrow band signal such as a trunking channel, or something like a strong ATIS signal. To find such frequencies, you can look them up online in your countries national frequency database or on a website such as radioreference.com.

Tune to the known frequency in SDR#. You will probably find that the actual tuned frequency is offset from the known signal. Zoom into the signal a little using the zoom slider. Click the Configure button. Now adjust the PPM offset value box accordingly so that the signal lines up exactly with the red tuning line. The PPM offset on most dongles is almost always a positive value between 0 and around 150. Remember to wait for the dongle to warm up after first plugging it in as changes in temperature can affect the PPM offset.

The images below show the effect of settings the PPM. The first image shows what happens when you tune to a known signal at 859.070 MHz without having the PPM set correctly. The tuning bar is far off the actual frequency. In the second image a PPM correct of 36 is set, and now the tuning bar correctly sits in the middle of the known signal's frequency.



RTL-SDR Controller

Device R820T

Generic RTL2832U OEM

Sample Rate

2.048 MSPS

Sampling Mode

Quadrature sampling

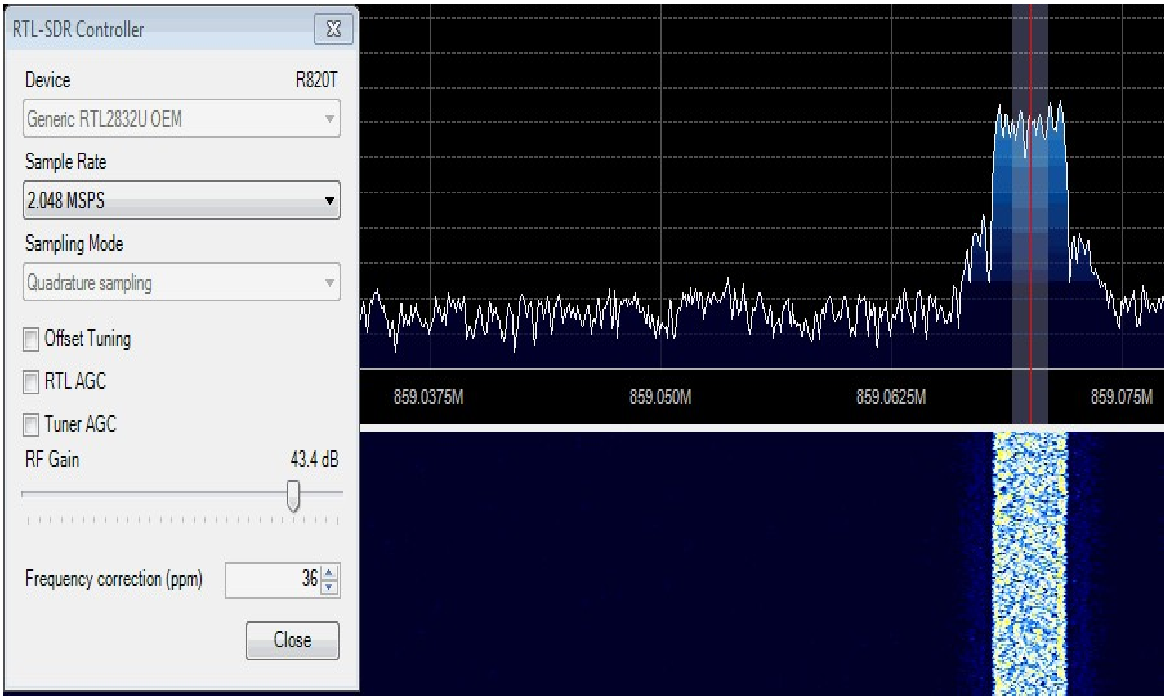
□ Offset Tuning

□ RTL AGC

□ Tuner AGC

RF Gain 43.4 dB

Frequency correction (ppm) 0



RTL-SDR Controller

Device R820T

Generic RTL2832U OEM

Sample Rate

2.048 MSPS

Sampling Mode

Quadrature sampling

☐ Offset Tuning

☐ RTL AGC

☐ Tuner AGC

RF Gain 43.4 dB

Frequency correction (ppm) 36

Close

Another way to get an accurate PPM offset value is to use a Linux program called Kalibrate. This command line program uses mobile phone GSM signals to accurately detect the dongles offset. Obviously you will need GSM signals in your area and an antenna which can receive them for this to work. A tutorial on this can be found in the Calibrating the RTL-SDR section of this book. However, note that we strongly recommend the manual method as Kalibrate can sometimes give inaccurate results.

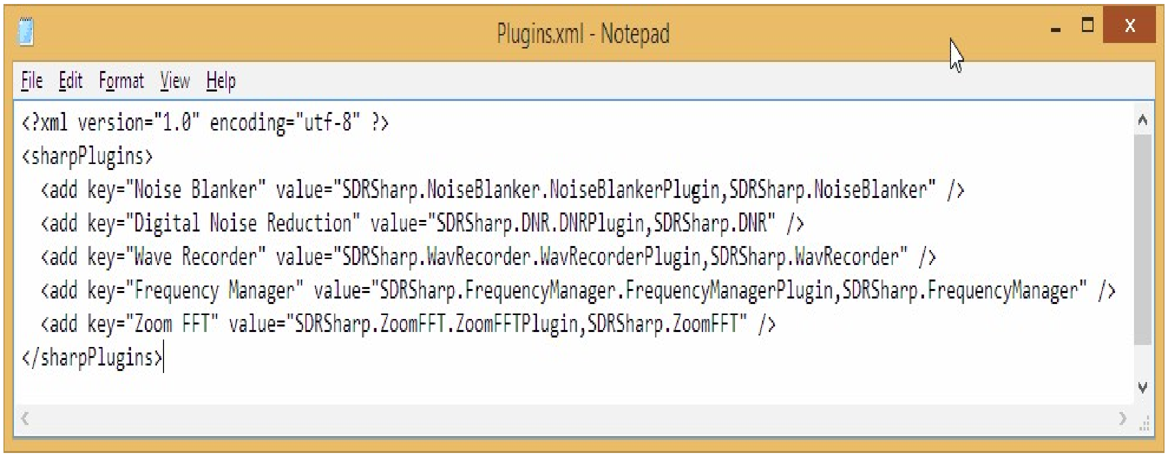
SDR SHARP PLUGINS

There are many plugins available for SDR# that extend its functionality. We recommend visiting http://www.rtl-sdr.com/sdrsharp-plugins/ and http://www.sdrsharp.com/#plugins for a list and brief overview of these plugins.

To install plugins you will most often need to copy a .dll file to the SDR# directory and add an entry to the Plugins.xml text file. This file is in the SDR# directory. You will need to open this file in Notepad or another text editor to read it.

The readme file that usually comes in the plugin zip file will usually tell you what line to add. The line should be added between the <sharpPlugins> </sharpPlugins> tags in the Plugins.xml file.

Note that some older plugin readme files may still instruct you to add the line to the SDRSharp.exe.config file. This was used in older SDR# versions and is now incorrect. The line should be added to the plugins.xml file now.



```
<?xml version="1.0" encoding="utf-8" ?>
<sharpPlugins>
  <add key="Noise Blanker" value="SDRSharp.NoiseBlanker.NoiseBlankerPlugin,SDRSharp.NoiseBlanker" />
  <add key="Digital Noise Reduction" value="SDRSharp.DNR.DNRPlugin,SDRSharp.DNR" />
  <add key="Wave Recorder" value="SDRSharp.WavRecorder.WavRecorderPlugin,SDRSharp.WavRecorder" />
  <add key="Frequency Manager" value="SDRSharp.FrequencyManager.FrequencyManagerPlugin,SDRSharp.FrequencyManager" />
  <add key="Zoom FFT" value="SDRSharp.ZoomFFT.ZoomFFTPlugin,SDRSharp.ZoomFFT" />
</sharpPlugins>
```

SDR# MINOUTPUT SAMPLE RATE

The maximum bandwidth of the tuned area for NFM in SDR# is restricted to 32 kHz. However, for some applications that bandwidth is too small. The maximum bandwidth for an NFM/USB/LSB signal can be adjusted by changing a setting in the SDRSharp.exe.Config file. Open this file in a text editor and find the following line.

```
<add key="minOutputSampleRate" value="32000" />
```

Change this line to the following to increase the maximum bandwidth.

```
<add key="minOutputSampleRate" value="100000" />
```

OTHER GENERAL PURPOSE SCANNING SOFTWARE

Once you’ve gotten the hang of the simple and easy to use SDR#, you may want to try other similar, but perhaps not as user friendly programs.

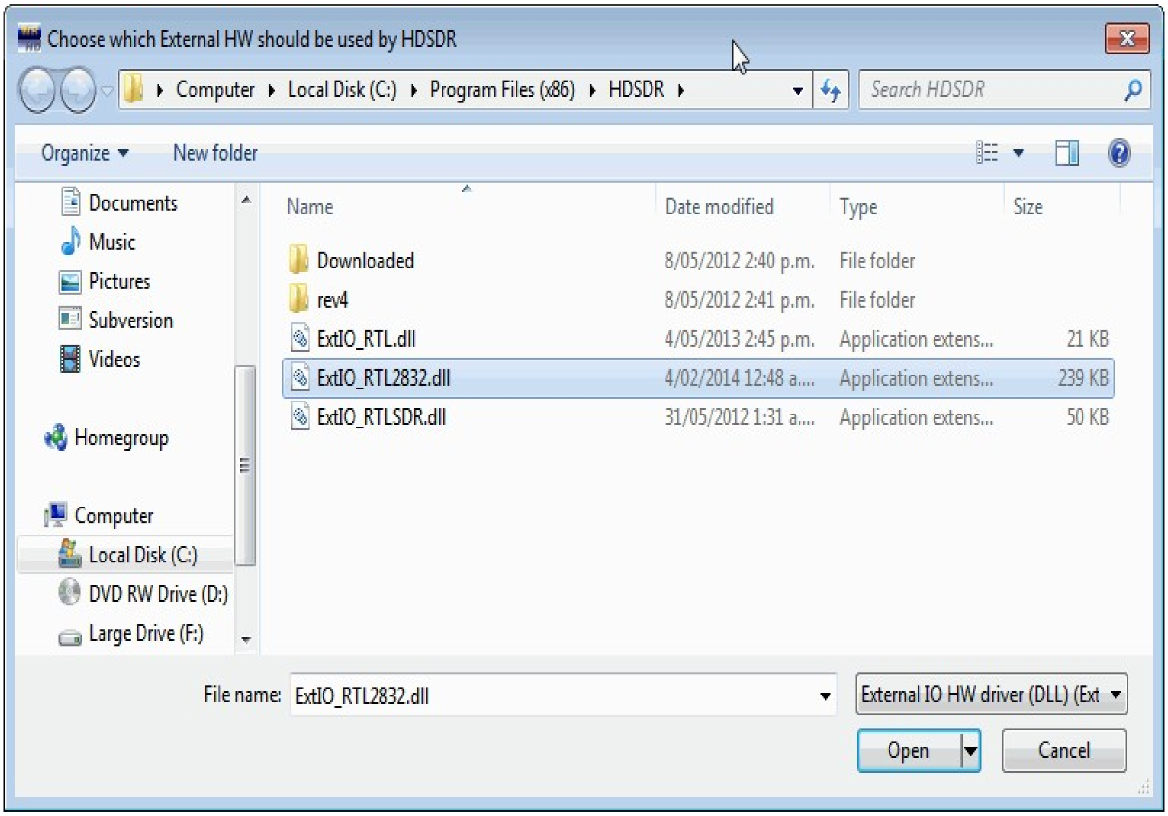
HDSDR SETUP GUIDE

HDSDR is an advanced SDR GUI similar to SDR#. It is based on the old WinRAD SDR program that was popular in the past with older SDRs.

Along with a FFT display and waterfall, HDSDR has some extra features. Users will find an Audio FFT and audio waterfall display on the bottom of the screen. The output audio can also be bandpass filtered by dragging the filter borders on the audio FFT display. Bandpass filtering the audio can really help clean up a noisy signal. The audio processing also supports placing of notch filters either manually or automatically. There are also noise reduction and noise blanker features and an automatic frequency centring algorithm which will automatically centre the signal, so you don’t need to click exactly in the centre of a signal. Traditional ham radio users will also enjoy the S-units signal strength meter and the built in frequency manager.

We recommend installing HDSDR after going through the quickstart guide and installing the drivers and SDR#. Then to install HDSDR on Windows follow the instructions below:

1. If you have not done so already, download and install zadig from http://zadig.akeo.ie/ and install the WinUSB drivers for RTL-SDR using the instructions in the SDR# setup guide above.
2. Download HDSDR from http://hdsdr.de/, using the download button at the bottom of the page.
3. Use the installer you just downloaded to install HDSDR.
4. Download the ExtIO_RTL2832U.dll dll file from https://app.box.com/s/7tpiy8r6qo2bbhdxtt4k (Mirror: http://bit.ly/1sXgFTX).
5. Copy the ExtIO_RTL2832U.dll file into the HDSDR install folder which is by default set to C:\Program Files (x86)\HDSDR.
6. Open HDSDR. You might be asked to select a .dll file. Choose the ExtIO_RTL2832U.dll file you just copied over and then click Open.



Choose which External HW should be used by HDSDR

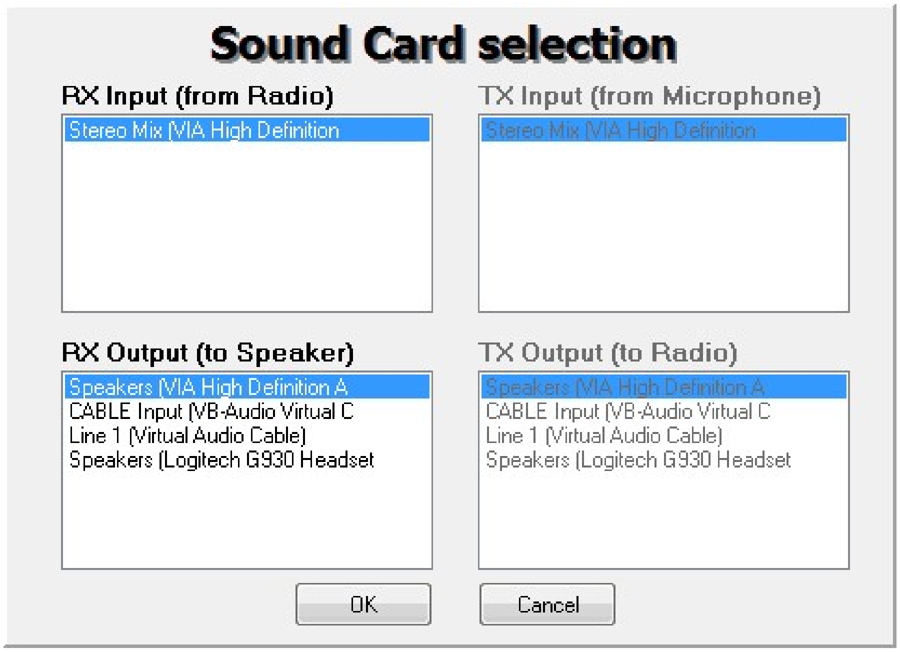
Computer ▸ Local Disk (C:) ▸ Program Files (x86) ▸ HDSDR ▸

Name	Date modified	Type	Size
Downloaded	8/05/2012 2:40 p.m.	File folder	
rev4	8/05/2012 2:41 p.m.	File folder	
ExtIO_RTL.dll	4/05/2013 2:45 p.m.	Application extens...	21 KB
ExtIO_RTL2832.dll	4/02/2014 12:48 a....	Application extens...	239 KB
ExtIO_RTLSDR.dll	31/05/2012 1:31 a....	Application extens...	50 KB

File name: ExtIO_RTL2832.dll

External IO HW driver (DLL) (Ext

7. Choose your output sound card by clicking on the Soundcard button Soundcard [F5] in the bottom left corner, or alternatively by pressing F5. The only important setting here is the “RX Output (to Speaker)” setting which you should set to your speakers, or desired audio piping software.



Sound Card selection

RX Input (from Radio)

Stereo Mix (VIA High Definition

TX Input (from Microphone)

Stereo Mix (VIA High Definition

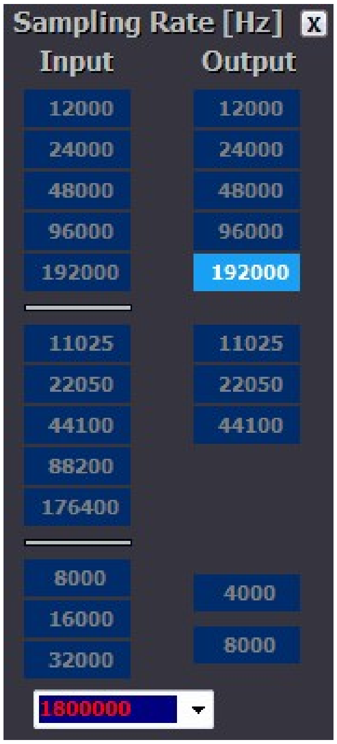
RX Output (to Speaker)

Speakers (VIA High Definition A
CABLE Input (VB-Audio Virtual C
Line 1 (Virtual Audio Cable)
Speakers (Logitech G930 Headset

TX Output (to Radio)

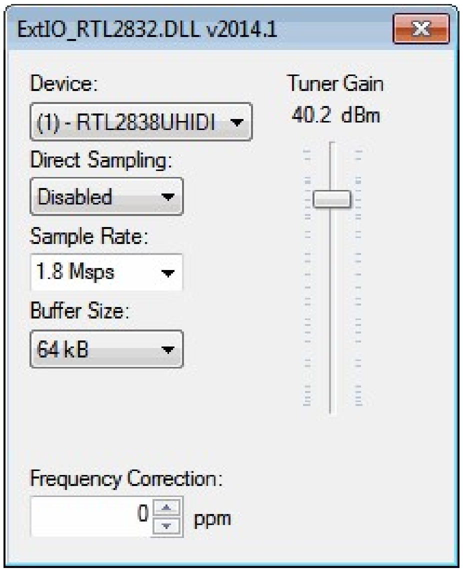
Speakers (VIA High Definition A
CABLE Input (VB-Audio Virtual C
Line 1 (Virtual Audio Cable)
Speakers (Logitech G930 Headset

8. Click on the Bandwidth button Bandwidth [F6] or alternatively press F6. Choose an output Sampling Rate of 48000 Hz for general use, or 192000 for wideband FM radio.



9. Press Start or alternatively press F2. This will start the SDR.

10. To set the RTL-SDR sample rate, gain and frequency correction click on the ExtIO button ExtIO.



ExtIO_RTL2832.DLL v2014.1

Device: (1) - RTL2838UHIDI

Direct Sampling: Disabled

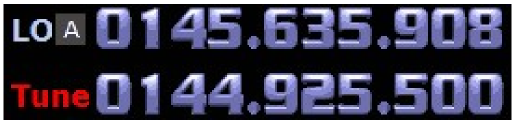
Sample Rate: 1.8 Msps

Buffer Size: 64 kB

Tuner Gain 40.2 dBm

Frequency Correction: 0 ppm

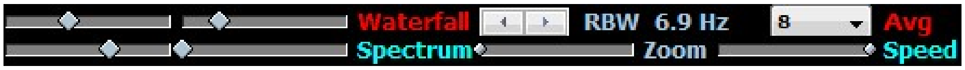
11. To tune to a station, change the Local Oscillator (LO) frequency to a frequency near the frequency you are interested in. Then tune to the desired frequency either by clicking in the RF spectrum, or using the numbers next to the red word Tune.



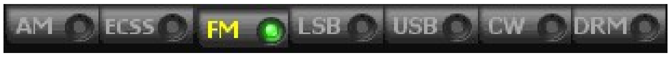
LO A 0145.635.908

Tune 0144.925.500

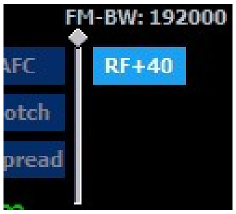
12. You can zoom in and out of the spectrum by using the Zoom slider which is to the left of the word zoom.



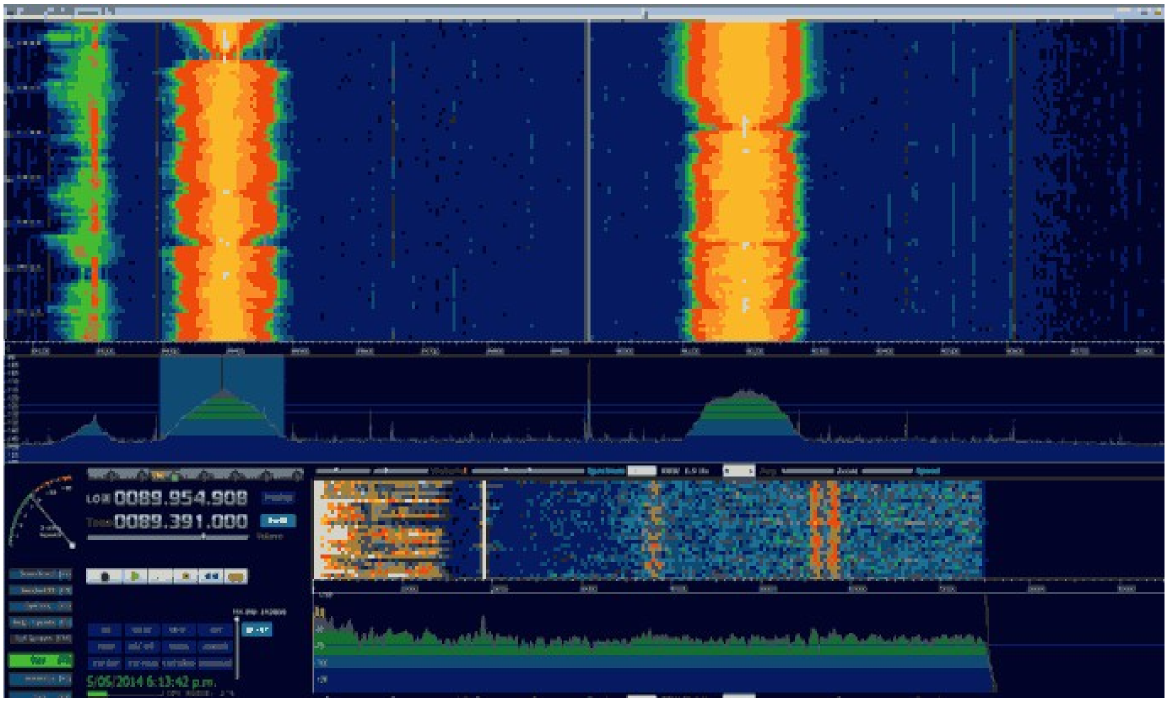
13. The mode can be altered by clicking on the mode buttons.



14. After clicking on the FM mode button, the FM bandwidth can be modified with the FM-BW slider. If this slider is missing you may need to click on the FM mode button again.



15. To listen to a typical wideband broadcast FM station, you will need to change the audio sampling rate to 192000 Hz. Do this by clicking on the Bandwidth button Bandwidth [F6] or alternatively by pressing F6, and the selecting the output sampling rate as 192000 Hz and then using the FM-BW slider to increase the bandwidth.



On the top of HDSDR is the RF waterfall and RF spectrum. On the bottom left we have the controls. On the bottom right is the audio waterfall and audio spectrum. The audio bandwidth can be bandpass filtered by dragging the red filter edges on the audio RF spectrum (the bottom graph).

SDR-RADIO.COM V2 SETUP GUIDE

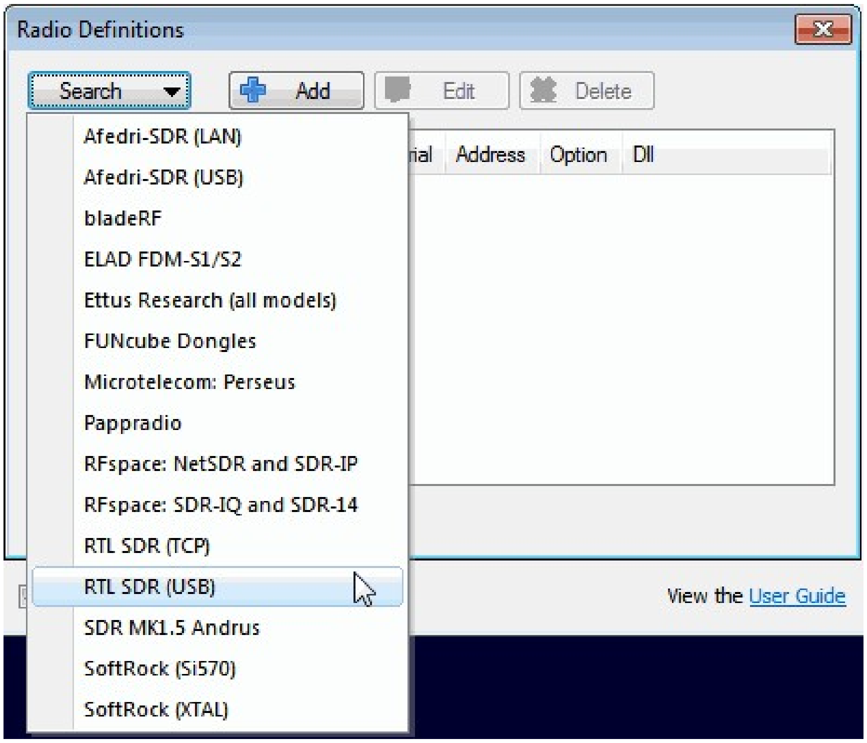
SDR-Radio V2 (sometimes also referred to as SDR-Console) is a popular SDR program with many advanced features. As such is it a fair amount more difficult to learn and use compared to SDR# and HDSDR. Be sure you install version 2 and not V1.5 as only V2 has RTL-SDR support.

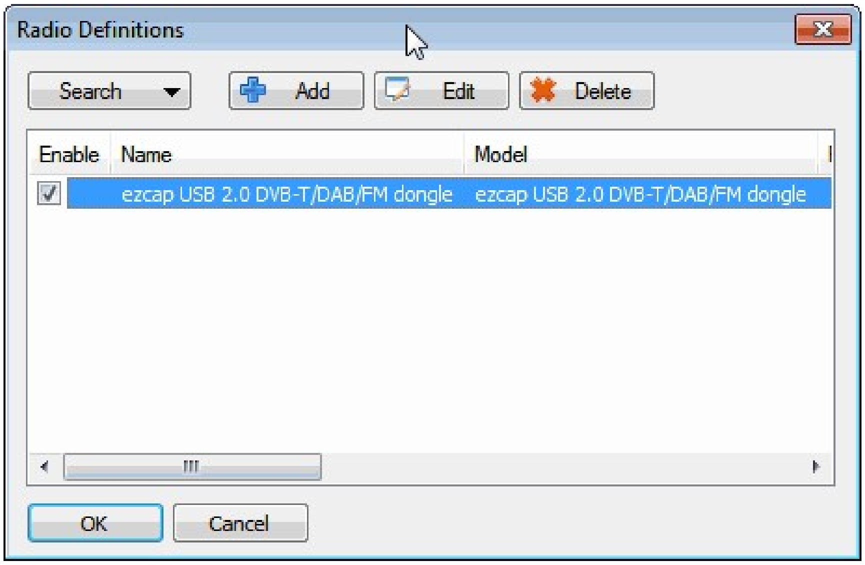
Like HDSDR, not only does SDR-Radio have a RF FFT signal and waterfall display, but also an optional audio spectrum FFT and waterfall display. Built in are also several DSP features like a noise blanker, noise reduction filter, notch filter and squelch options. The EMNS noise reduction filter is particularly good at automatically cleaning up and clarifying voice signals.

To add to the feature list, SDR-Radio also has built in PSK, RTTY and RDS decoders and also comes with a satellite tracker. Furthermore, SDR-Radio has an excellent remote server application which will allow you to easily set up and connect to a remote RTL-SDR server over a network or the internet. Finally, SDR-Radio is capable of listening to up to six signals in the same chunk of visible spectrum at a time, or in other words it can have six simultaenous VFO's.

To install SDR-Radio for the RTL-SDR follow the steps below.

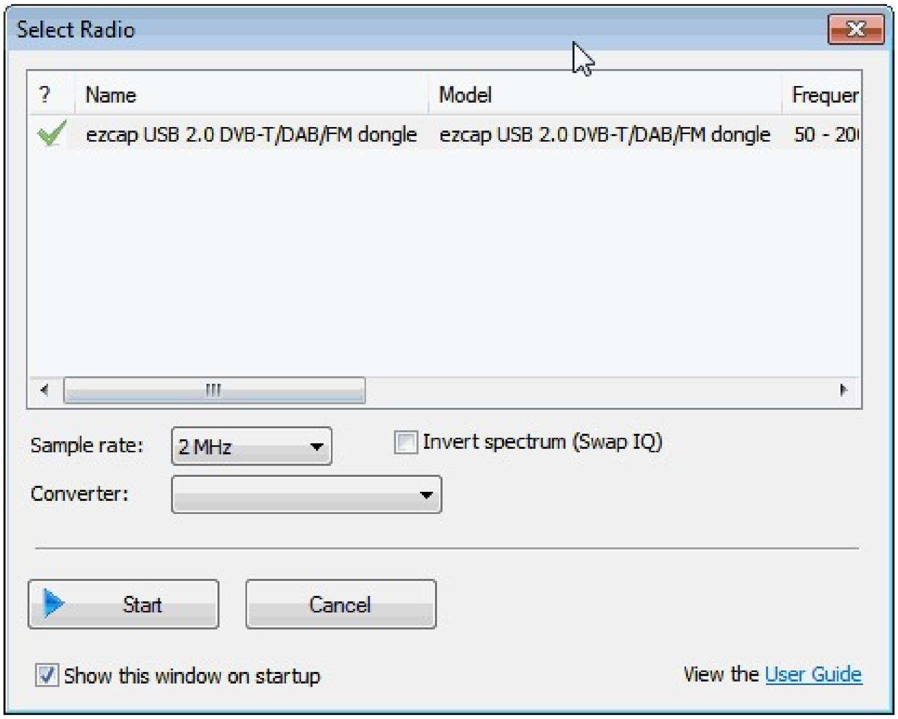
1. If you have not done so already, download and install zadig from http://zadig.akeo.ie/ and install the WinUSB drivers for RTL-SDR using the instructions in the SDR# setup guide above.
2. Download the SDR-Radio installer from http://v2.sdr-radio.com/.
3. Use the installer to install SDR-Radio.
4. Download from the bottom of http://www.aa5sh.com/?page_id=65 (Mirror: http://bit.ly/1oPcWTm) the SDRSourceRTL2832U.dll, rtlsdr.dll and libusb-1.0.dll dll files. Copy them into either the C:\Program Files\SDR-RADIO-PRO.com (64 bit), or C:\Program Files (x86)\SDR-RADIO-PRO.com (32 bit) folder depending on what version of Windows you have.
5. Open SDR-Radio. Upon opening it you will be greeted with the Select Radio screen and a prompt saying "List is empty - add radio definition now?" Click Yes. If this prompt does not display, click the + Definitions button.
6. In the new window open the Search drop down menu and select RTL SDR (USB). After clicking it the RTL-SDR will be added to the Radio Definitions list. Click OK.





Enable	Name	Model
☑	ezcap USB 2.0 DVB-T/DAB/FM dongle	ezcap USB 2.0 DVB-T/DAB/FM dongle

7. Click on the RTL-SDR to select it, choose your desired sample rate then click Start.



Select Radio

?	Name	Model	Frequer
✓	ezcap USB 2.0 DVB-T/DAB/FM dongle	ezcap USB 2.0 DVB-T/DAB/FM dongle	50 - 20

Sample rate: 2 MHz

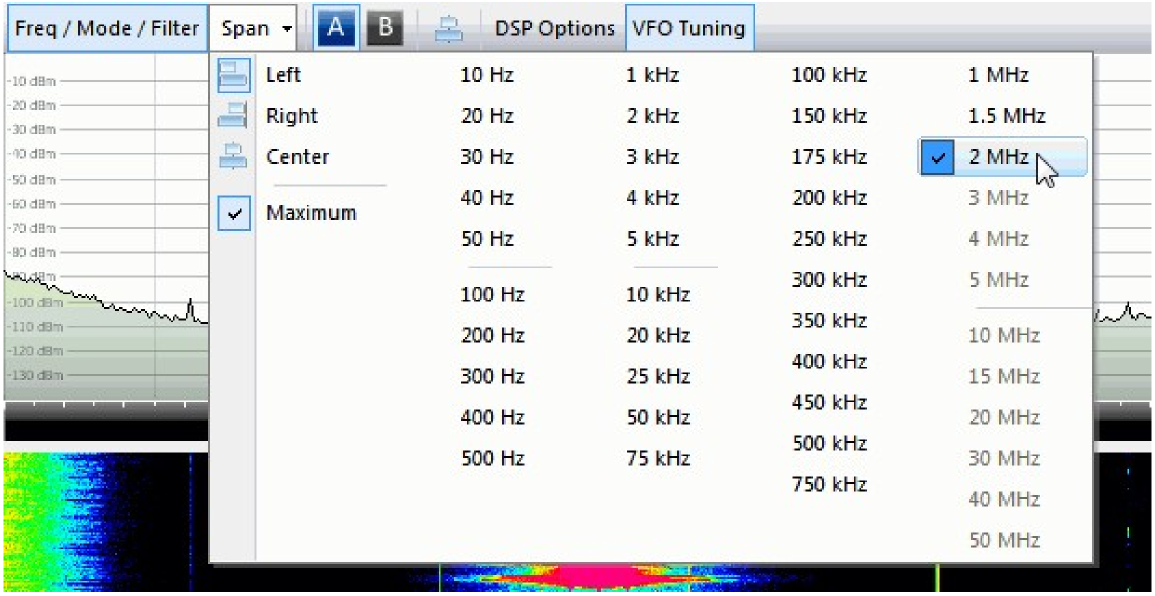
☐ Invert spectrum (Swap IQ)

Converter:

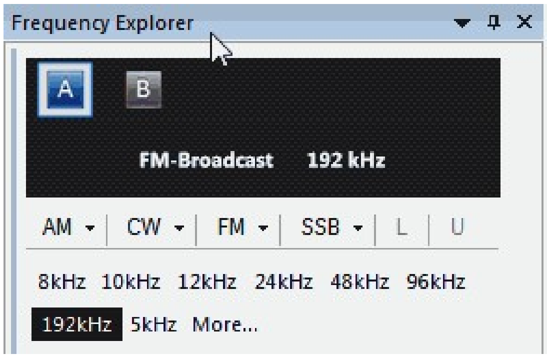
☑ Show this window on startup

View the User Guide

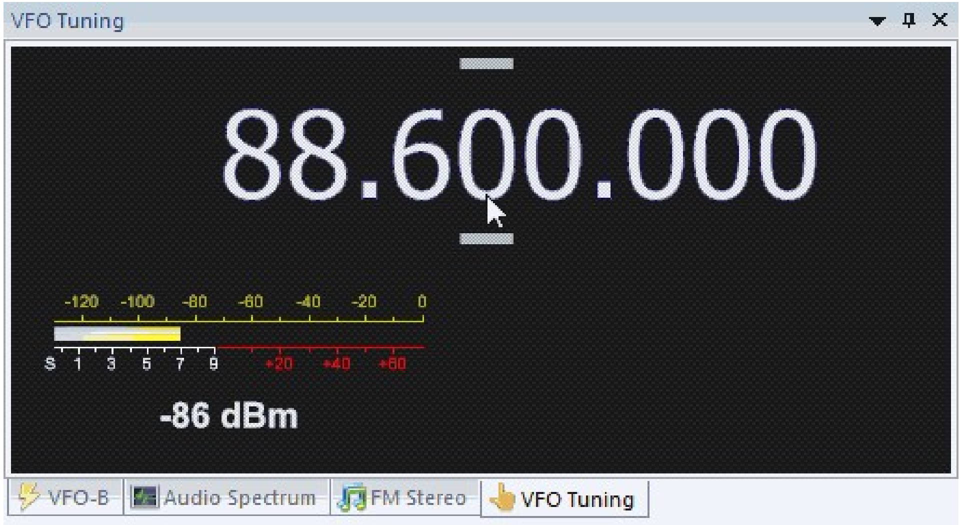
8. Click on the Span button and adjust the span to the sample rate bandwidth you chose in the last step. This will let you see the whole spectrum.



9. To change the receive mode and bandwidth use the left menu under the frequency tab.

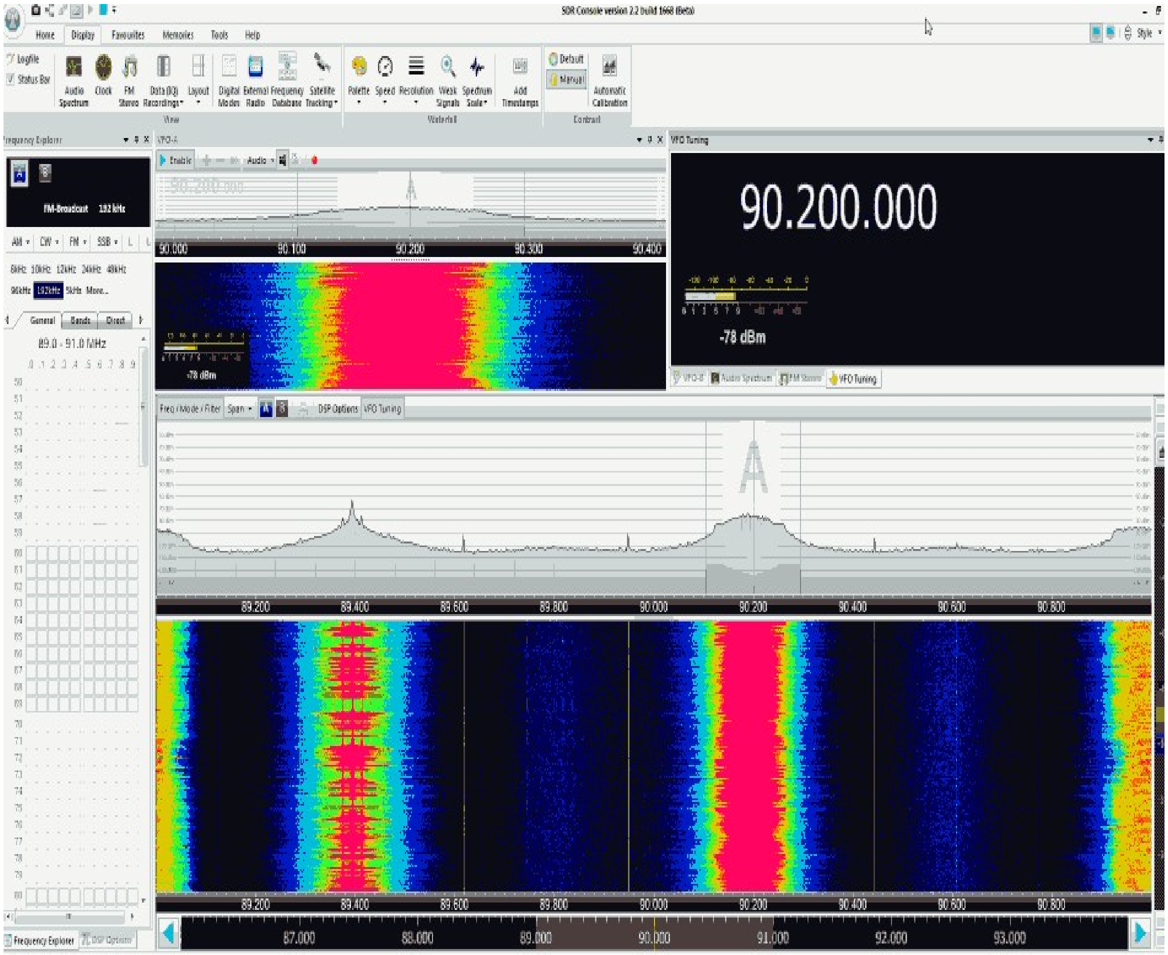


10. To adjust the frequency, use the VFO tuning box on the right side. If you don’t see any numbers in this box you may need to increase the size of the VFO tuning box or maximise the program.



11. Be sure to adjust the gain settings using the RF gain button RF: Automatic on the top toolbar which is under the Home tab. By default it is set to automatic.

12. To adjust the waterfall colors so that signals can be more clearly seen go to the Display tab up the top and then click on the Automatic Calibration button Automatic Calibration on the very top right.



SDR-Radio has its main wideband waterfall on the bottom of the screen and directly above it is the RF spectrum. Above that is another smaller waterfall and RF spectrum which is simply the tuned frequency zoomed in.

GQRX

GQRX is a software defined radio received GUI for OSX and Linux. It is similar to SDR# in terms of usability, but has less features. It also has a built in AFSK1200 decoder. To install GQRX on Linux follow the instructions shown below.

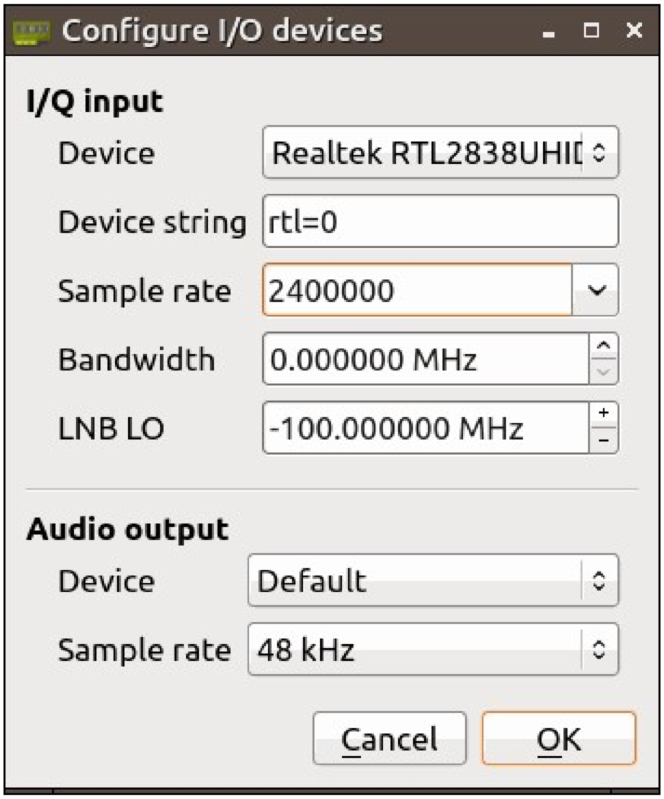
The easiest way to install GQRX is from the respositories. Simply use:

```
sudo apt-get install gqrx-sdr
```

If you want to install GQRX from source you will need to have GNU Radio 3.7 and the RTL-SDR drivers from libosmocore installed first. See the Github page for a list of all required dependencies https://github.com/csete/gqrx. After installing all the dependencies it is a simple matter of running the following.

```
git clone https://github.com/csete/gqrx.git gqrx.git
cd gqrx.git
mkdir build
cd build
qmake ..
make
sudo make install
```

You can then run the software by typing `gqrx` into a terminal to start GQRX. Upon starting GQRX for the first time you will be greeted with a screen asking to configure I/O devices. Simply select the Realtek RTL2832UUHID option from the Device menu and choose your desired sample rate. The other settings can be left as default.



Configure I/O devices

I/Q input

Device: Realtek RTL2838UHII

Device string: rtl=0

Sample rate: 2400000

Bandwidth: 0.000000 MHz

LNB LO: -100.000000 MHz

Audio output

Device: Default

Sample rate: 48 kHz

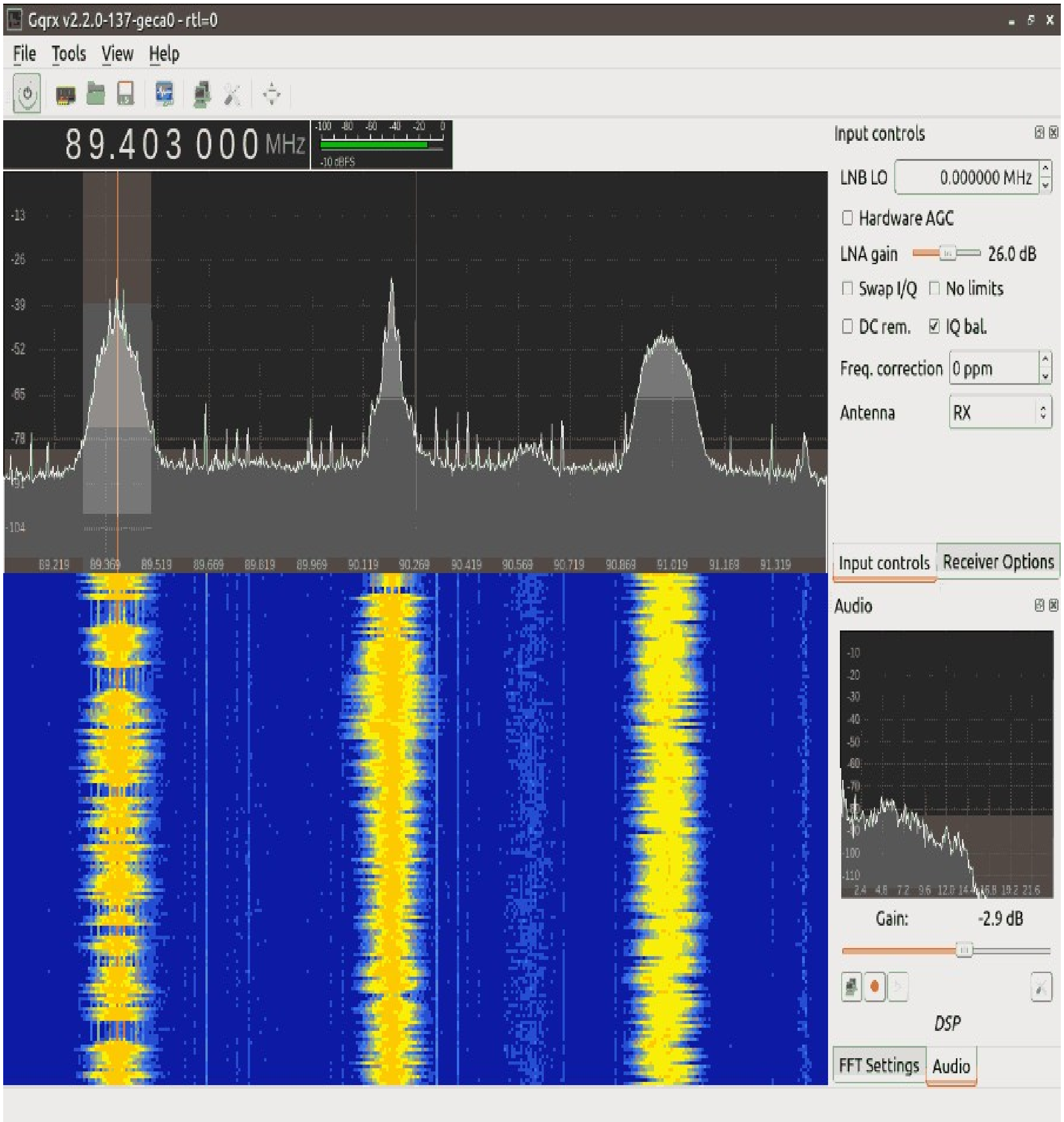
Next press the Start DSP button ⏻ which is in the top left corner. Adjust the frequency using the mouse and the numerical values in the top left corner. The mode and filter bandwidth can be adjusted in the top right under the Receiver Options tab. The filter bandwidth can also be quickly changed by holding down the CTRL key + scrolling the mouse wheel. The RF gain can be set in the top right under the Input controls tab by sliding the LNA gain.

INSTALLING GQRX ON OSX

The easiest way to install GQRX on a Mac computer running OSX is by using Macports. Macports is a package manager for OSX systems that has many Linux ported software available for download. To use Macports you will first need to have the XCode installed, which can be downloaded from the OSX App store. If you need guidance with installing XCode there is a guide available on the Macports website at http://guide.macports.org/#installing.xcode. Once you have XCode installed, you can download Macports for you version of OSX from https://www.macports.org/install.php.

Now to install GQRX from Macports, open a terminal window and type `sudo port install gqrx`. Installation can take a long time (a few hours) as GQRX requires the installation and compilation of GNU Radio too. Once finished you should be able to find GQRX installed in your dash.

If for some reason Macports does not work for you, then we suggest installing a premade (though outdated) GQRX .dmg file from http://eliasoenal.com/2012/09/30/osx-port-of-the-awesome-gqrx-sdr-software/.



Gqrx v2.2.0-137-geca0 - rtl=0

89.403 000 MHz

Input controls

LNB LO 0.000000 MHz

□ Hardware AGC

LNA gain 26.0 dB

□ Swap I/Q □ No limits

□ DC rem. ☑ IQ bal.

Freq. correction 0 ppm

Antenna RX

Audio

Gain: -2.9 dB

CUBICSDR

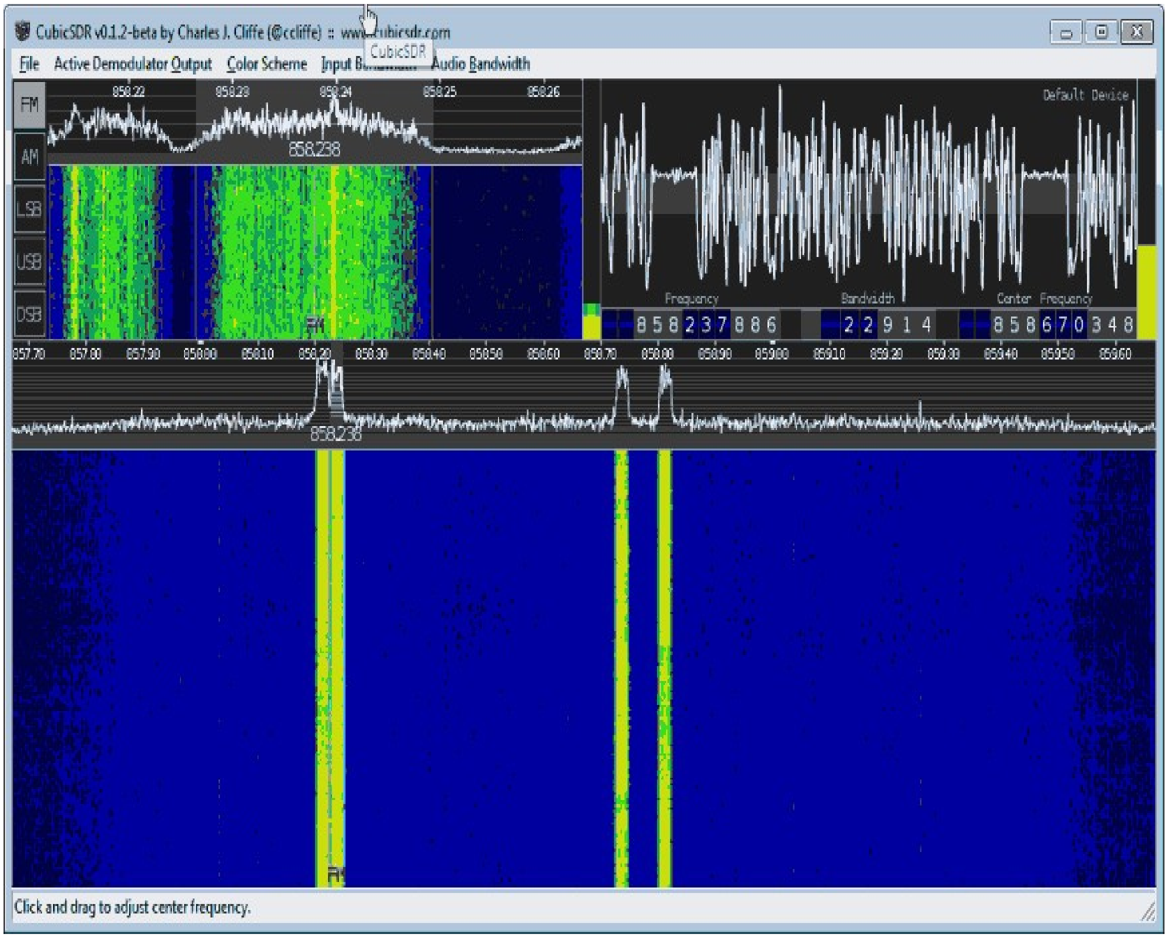
CubicSDR is a relatively new sdr receiver program which is showing good promise. It if fully cross platform supporting Windows, Linux and OSX. Setting it up is extremely simple.

1. If you have not done so already, download and install zadig from http://zadig.akeo.ie/ and install the WinUSB drivers for RTL-SDR using the instructions in the SDR# setup guide shown above.

2. Go to www.cubicsdr.com and download the latest version for your OS from the downloads page.
3. Open the installer and follow the prompts. This will install the software as well as all the appropriate RTL-SDR dll files needed.
4. Open CubicSDR from the start menu, or Linux/OSX equivalent.

If your RTL-SDR is plugged in, CubicSDR will automatically start and the spectrum will be visible. Simply click in the bottom waterfall area to tune.

The screen is broken up into several parts. In the top left is a zoomed in RF spectrum and waterfall display. On the right is an audio spectrum display and the main section on the bottom is a wideband RF spectrum and waterfall display. The center frequency can be changed by using the center frequency adjustment in the center right of the screen and the bandwidth and tuned frequency may also be changed here. Bandwidth can also be changed by clicking and dragging the edges of the tuned area in the main waterfall display. The mode can be selected with the top left vertical bar. The sample rate bandwidth can be changed with the input bandwidth menu. To easily set the PPM offset, hold the ALT button whilst hovering over the frequency input.



Click and drag to adjust center frequency.

LINRAD

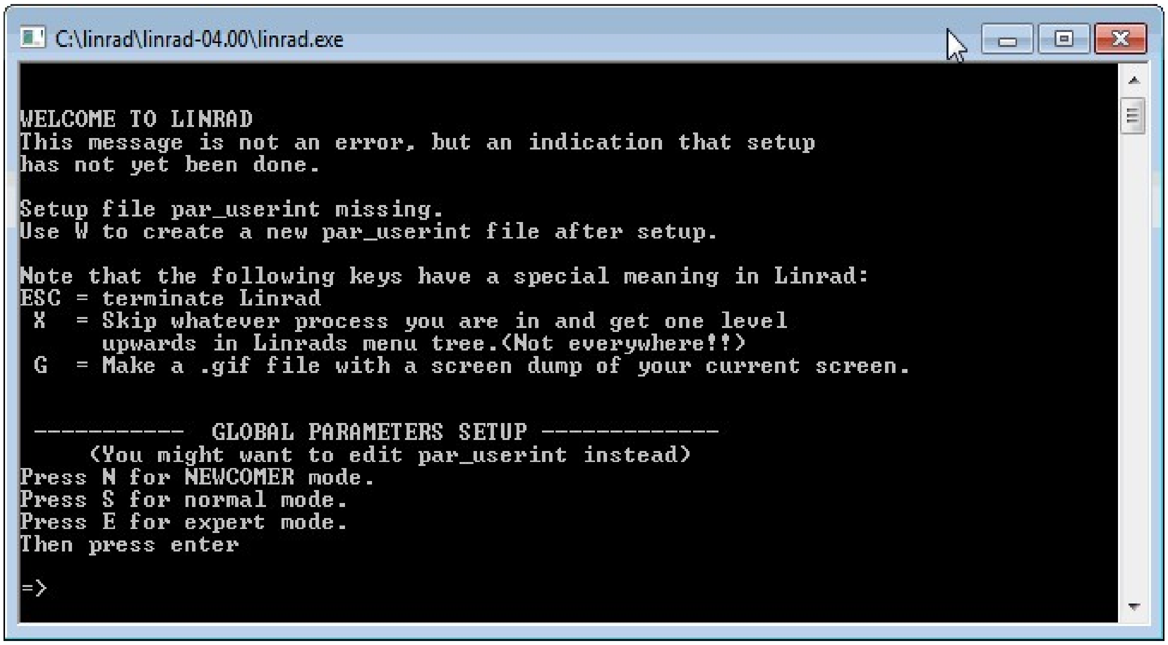
Linrad is a SDR receiver program that has some very advanced features and can also run on older PCs (Pentium 4's). While Linrad has advanced features it also has a very difficult learning curve and is for advanced users only. This tutorial will show you how to set up Linrad on Windows and test it on Wideband FM radio. More advanced features require I/Q and filter calibration which requires a pulse generator. A good tutorial on setting up these calibrations can be found at https://www.youtube.com/watch?v=rPQARPt6r_8, which is a tutorial video by the programmer of Linrad.

1. If you have not done so already, download and install zadig from http://zadig.akeo.ie/ and install the WinUSB drivers for RTL-SDR using the instructions in the SDR# setup guide shown above.

2. Go to http://www.sm5bsz.com/linuxdsp/linrad.htm and download and run the Linrad DLL installer for Windows (setup-linrad-dll-package-03.exe (804985 bytes)).

3. Download and run the setup program for Windows setup-linrad-04.00.exe (843728 bytes).

4. Open Linrad. You will be greeted with the following command prompt window.



```
WELCOME TO LINRAD
This message is not an error, but an indication that setup
has not yet been done.

Setup file par_userint missing.
Use W to create a new par_userint file after setup.

Note that the following keys have a special meaning in Linrad:
ESC = terminate Linrad
 X  = Skip whatever process you are in and get one level
      upwards in Linrads menu tree.(Not everywhere!!)
 G  = Make a .gif file with a screen dump of your current screen.


 -----------  GLOBAL PARAMETERS SETUP -------------
     (You might want to edit par_userint instead)
Press N for NEWCOMER mode.
Press S for normal mode.
Press E for expert mode.
Then press enter

=>
```

5. Type in S, then press enter to open Linrad in normal mode.

6. Next you will be asked to enter the font scaling (size). Choose lower numbers for smaller screens and larger numbers for larger screens.

7. Next choose the process priority as 0 for NORMAL priority.

8. Next choose N to specify the window size in percentage. Then choose 100% width and around 90% for the height.

9. Now the Linrad screen will open. Press the W key to save the options you've chosen so far.

10. Press the U key to configure the input device, then press A to get to the RF input device select screen. Press H to set the input device as the RTL2832. (The RTL-SDR must be connected for Linrad to detect it).

11. Choose 2400000 as the sampling speed. Select 0 for normal I/Q tuner operation and then select 0 for Auto gain mode. If you know your frequency error in PPB (parts per billion) enter it now, otherwise just enter 0.

12. Now press B to set up the output soundcard. Type N to not select PortAudio as the output. Now look on the list you are presented with and find your speakers, or virtual audio pipe that you'd like to use.

13. Press X to get to the main menu and then press W to save your settings.

14. Now press E to get to the FM demodulation screen. Select the default values in the next few screens by pressing Enter.

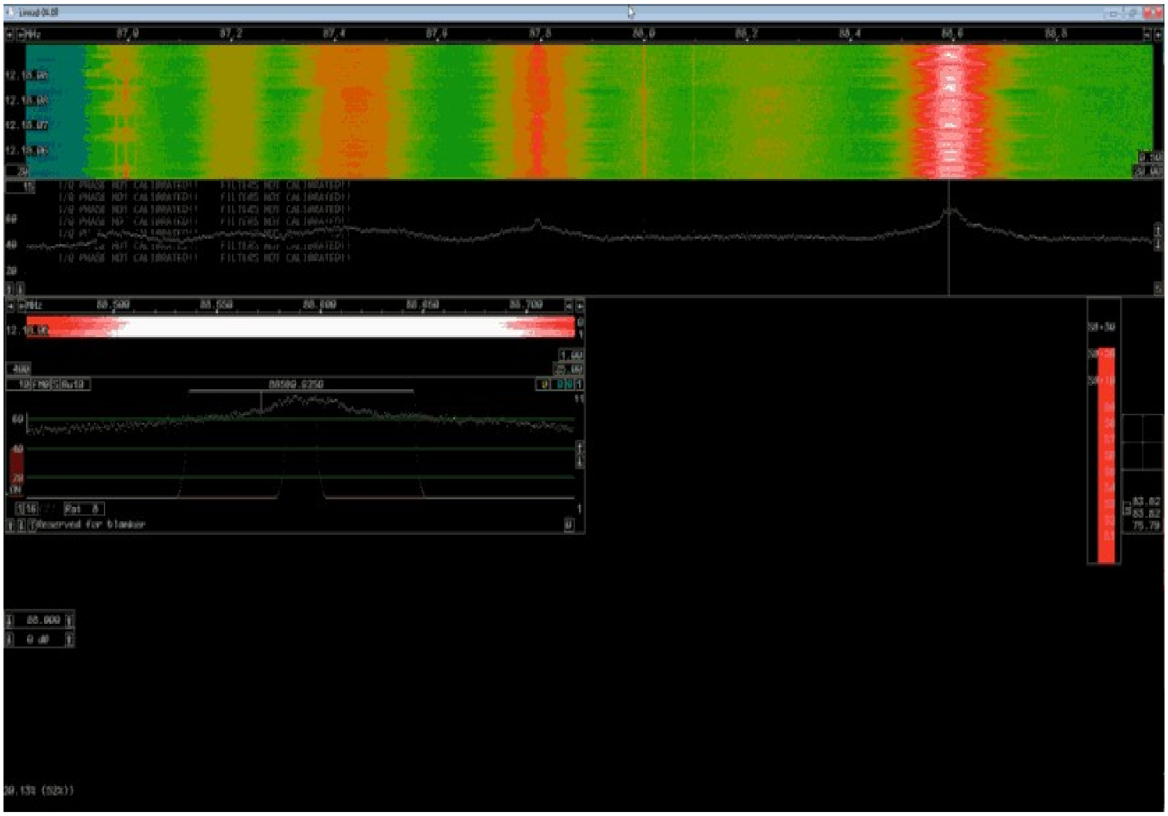
15. In the new receiver screen that shows, first click on the very top bar showing the frequency range. On the left of the bar, enter a value of 0 and on the right enter an arbitrarily large value (eg 1000). Click Apply. This will select the maximum bandwidth frequency range for the waterfall.

16. To increase the waterfall speed, change the value in the bottom left of the waterfall to a lower value, such as 9.

17. Press X then P to get to the parameters screen. Click on the First FFT Amplitude text and set the new amplitude to 20. This will lower the noise floor.

18. If your PC is struggling to keep up, press X then P and change the First FFT Bandwidth setting to 1 kHz.

19. To tune to a station, click on it's peak in the top wide RF spectrum. Alter it's bandwidth by dragging the yellow bar in the lower RF spectrum. Drag to the right to increase the bandwidth for wideband FM.



```
I/Q PHASE NOT CALIBRATED!!    FILTERS NOT CALIBRATED!!
I/Q PHASE NOT CALIBRATED!!    FILTERS NOT CALIBRATED!!
I/Q PHASE NOT CALIBRATED!!    FILTERS NOT CALIBRATED!!
I/Q PHASE NOT CALIBRATED!!    FILTERS NOT CALIBRATED!!
I/Q PHASE NOT CALIBRATED!!    FILTERS NOT CALIBRATED!!
I/Q PHASE NOT CALIBRATED!!    FILTERS NOT CALIBRATED!!
I/Q PHASE NOT CALIBRATED!!    FILTERS NOT CALIBRATED!!
```

INSTALLING THE RTL-SDR DRIVERS ON LINUX

If you want to get started with the RTL-SDR on a Linux system the first step is to install the drivers. First update your distribution using the following.

```
sudo apt-get update
```

Now if not installed already, install git, cmake and libusb which is the USB driver library required for RTL-SDR.

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

Now download, compile and install the RTL2832U Osmocom drivers. (Note that we also recommend trying out Keenerds drivers which can be cloned using the git link https://github.com/keenerd/rtl-sdr.git in place of the osmocom one)

```
git clone git://git.osmocom.org/rtl-sdr.git
cd rtl-sdr/
mkdir build
```

```
cd build
cmake ../ -DINSTALL_UDEV_RULES=ON
make
sudo make install
sudo ldconfig
sudo cp ../rtl-sdr.rules /etc/udev/rules.d/
```

In some cases you may now need to restart Linux.

Plug in the dongle and then run `rtl_test -t` to test the dongle installation. If rtl_test returns strange characters under the found devices heading, or you get errors like the following:

```
Kernel driver is active, or device is claimed by second instance of librtlsdr.
In the first case, please either detach or blacklist the kernel module
(dvb_usb_rtl28xxu), or enable automatic detaching at compile time.
usb_claim_interface error -6
Failed to open rtlsdr device #0.
```

Then this means that the RTL2832U DVB-T TV drivers that are included in newer Linux versions are currently installed and activated. Disable them by running the following command.

```
sudo rmmod dvb_usb_rtl28xxu
```

Note that running this command will only stop the DVB-T drivers once. When you remove and reconnect the dongle or restart Linux, the DVB-T drivers will be reloaded and you'll need to run this line again.

For a more permanent solution create a text file "blacklist-dvb_usb_rtl28xxu.conf" in /etc/modprobe.d. In that file put the text `blacklist dvb_usb_rtl28xxu`. An easy way to do this is to simply type the following into a terminal:

```
sudo echo "blacklist dvb_usb_rtl28xxu" > /etc/modprobe.d/blacklist-dvb_usb_rtl28xxu.conf
```

After blacklisting the DVB-T drivers you can unplug and reconnect the dongle and it should now default to using the RTL-SDR drivers, though in some cases you may need to restart Linux first for the blacklist to take effect.

If you get strange characters showing up when you run rtl_test, it probably indicates that you did not properly install the .rules file with the `cp ../rtl-sdr.rules /etc/udev/rules.d/` command.

RTL_FM

Rtl_fm is a lightweight command line based WBFM/AM/FM/SSB audio receiver for the RTL-SDR which runs on both Windows and Linux. It is provided as part of the official RTL-SDR release from http://sdr.osmocom.org/trac/wiki/rtl-sdr. For Windows download the pre-compiled reldeb.zip release from the above link. For Linux follow the Installing RTL-SDR Drivers on Linux guide above. On Linux rtl_fm is installed as part of the RTL-SDR driver install procedure shown above. You may also wish to try out keenerds updated versions for which Linux and precompiled Windows versions can be found at https://github.com/keenerd/rtl-sdr (GitHub for Linux compilation) and http://igg.kmkeen.com/builds/ (Windows binaries).

To use the play command used in the examples below you will need to install sox or aplay. Sox can be obtained on Debian Linux with sudo apt-get install sox. On a Mac use port install sox. On Windows it can be downloaded from http://sourceforge.net/projects/sox/. You will need to add the folder sox is installed to to the Windows PATH, so it can be accessed from anywhere in the command prompt. You may wish to add the folder with rtl_fm in it to the PATH as well so that it too can be accessed from anywhere in command prompt. In this link is a tutorial showing how to add directories to the PATH http://geekswithblogs.net/renso/archive/2009/10/21/how-to-set-the-windows-path-in-windows-7.aspx. On Windows you will need to add a -d flag at the end of the command to select the default audio device. Additionally, on Windows the ‘play’ command is replaced by the ‘sox’ command.

EXAMPLES OF RTL_FM USE

Rtl_fm needs to be piped into an audio player that can play raw audio such as sox or aplay. We need to make one distinction in terminology for rtl_fm first. When we refer to the bandwidth in rtl_fm, we are referring to the tuned area bandwidth (for example the shaded area in SDR#), NOT the wide band instantaneous bandwidth. Below we list some of the possible rtl_fm flags and then show some examples of it in use.

RTL_FM FLAGS

```
-M set the receive mode (wbfm, fm, am, lsb, usb, raw) (default is fm)
-s bandwidth/sample rate of tuned area (not entire spectrum)
-g gain (default is automatic)
-l squelch level
-t squelch delay
-p ppm offset
-F use better filters (0 - 9)
-r output sample rate (very low quality resampling)
-d dongle number (for use with multiple dongles)
-A arctan computation (std, fast, lut)
-E special modes (dc, deemp, direct, offset)
```

For the full list type in rtl_fm -h.

RECEIVING WBFM EXAMPLE

Rtl_fm in WBFM mode uses an audio output sample rate of 32k.

```
rtl_fm -M wbfm -f 88.6M | play -r 32k -t raw -e s -b 16 -c 1 -V1 -
```

For better quality audio output the audio sample rate can be resampled to 48 kHz using the -r flag as follows.

```
rtl_fm -M wbfm -f 88.6M -r 48k | play -r 48k -t raw -e s -b 16 -c 1 -V1 -
```

Instead of sox, Alsa aplay can also be used as an alternative as follows. On some Linux computers aplay results in better sounding audio.

```
rtl_fm -M wbfm -f 88.6M -r 48k | aplay -r 48k -f S16_LE
```

On Windows use the following.

```
rtl_fm -M wbfm -f 88.6M | sox -r 32k -t raw -e s -b 16 -c 1 -V1 - -d
```

Also, instead of using -M wbfm, it is possible to also use -M fm and to specifically specify the bandwidth with -s and the deemphasis option with -E. This method sometimes produces better sounding audio.

```
rtl_fm -M fm -f 88.6M -s 200k -r 32k -E deemp | play -r 32k -t raw -e s -b 16 -c 1 -V1 -
```

NARROW BAND FM EXAMPLE

```
rtl_fm -M fm -f 161.649M -s 12k -g 50 -l 70 | play -r 12k -t raw -e s -b 16 -c 1 -V1 -
```

AM EXAMPLE

```
rtl_fm -M am -f 127.7M -s 12k -g 50 | play -r 12k -t raw -e s -b 16 -c 1 -V1 -
```

For all examples shown above the quality of the received signal at the expense of increased computation time can be increased by using the -F option. Use a number between 0-9 after -F, where higher numbers are higher quality downsampling filters. Even using a value of -F0 will improve audio quality.

SCANNER EXAMPLE

Rtl_fm can also be used for scanning multiple frequencies for activity. Simply add multiple -f frequencies of the channels that you wish to scan. Rtl_fm will continue scanning until it finds an active signal. An example is shown below.

```
rtl_fm -M fm -f 161.649M -f 161.925M -f 161.825 -s 12k -g 50 -l 70 | play -r 12k -t raw -e s -b 16 -c 1 -V1 -
```

You can also specify a frequency range and step size to scan through. For example below to scan the air band you could use the following,

```
rtl_fm -M am -f 118M:137M:25k -s 12k -g 50 | play -r 12k -t raw -e s -b 16 -c 1 -V1 -
```

CHANGING THE AUDIO SAMPLE RATE WITH SOX

Rtl_fm has the -r flag for adjusting the output sample rate. However, it is a poor quality conversion and cannot upsample. If you need to change the sample rate of the output of rtl_fm for whatever reason, you can use sox as follows.

```
rtl_fm -M fm -f 161.649M -s 12k -g 50 -l 0 | sox -r 12k -t raw -e s -b 16 - -traw -r 48k -e s -b 16 -c 1 - | play -
r 48k -t raw -e s -b 16 -c 1 -V1 -
```

An example for better sounding WBFM and with aplay for output

```
rtl_fm -M fm -f 88.6M -s 170k | sox -r 170k -t raw -e s -b 16 - -traw -r 48k -e s -b 16 -c 1 - | aplay -r 48k -f
S16_LE
```

The full documentation on rtl_fm can be found at http://kmkeen.com/rtl-demod-guide/index.html.

If you want to stream rtl_fm audio over a network connect see the MP3 streaming tutorial in the projects section. Also to use rtl_fm with digital demodulation software like multimonng, see the MultimongNG section later in this book.

INSTALLING GNU RADIO

GNU Radio is a popular Linux based digital signal processing platform that is compatible with the RTL-SDR. GNU Radio is a large complicated program and can sometimes be difficult to install. Below we show the most common methods.

Important Note: Do not run an installation process where GNU Radio is compiled on an embedded device running on an SDCard. The amount of writes GNU Radio does during the installation process can destroy the card. After installation GNU Radio can be started by typing `gnuradio-companion` at a terminal window.

If you happen to be running GNU Radio in an Ubuntu virtual machine, you may need to adjust some settings found in `/usr/local/etc/gnuradio/conf.d/gr-audio-alsa.conf`. You may need to alter these values to period_time = 0.10 and nperiods = 32 if you experience choppy audio.

Note that unless you use the install script shown below, the RTL-SDR Linux drivers should be pre-installed for GNU Radio to work with the RTL-SDR. See Installing RTL-SDR Drivers on Linux. If you don’t want to go through the process of install GNU Radio there are several Linux distributions and Live DVDs that now have GNU Radio preinstalled. See the GNU Radio Live DVDs section for more information.

INSTALL SCRIPT

The easiest way to install the latest version of GNU Radio on almost any Linux OS is to use Marcus Leech’s install script. This script will also install various other related libraries such as RTL-SDR and OsmoSDR. We prefer this method as it is the easiest and most likely to succeed. To use the script all that is required is to type the following into terminal.

```
sudo apt-get update
wget http://www.sbrac.org/files/build-gnuradio && chmod a+x ./build-gnuradio && ./build-gnuradio
```

The script will ask a few questions which you should answer ‘yes’ or ‘y’ to. It will download and install all the required dependencies as well as GNU Radio. After a few minutes to an hour plus the script will stop and ask you to enter your administrator password in order for it to use sudo, so you will need to remember to periodically check the scripts progress. Also note that the script will take around 1-3 hours to complete depending on your internet and PC speeds.

If for compatibility reasons you need to install the older GNU Radio 3.6.5.1, then you can use the -o flag as follows.

```
wget http://www.sbrac.org/files/build-gnuradio && chmod a+x ./build-gnuradio && ./build-gnuradio -o
```

Note that if you get an error like “Fetching Gnu Radio via GIT...Could not find gnuradio/gnuradio-{core,runtime} after GIT checkout”, then try run the script again. Sometimes the GNU Radio git server has trouble serving the files, but it will usually start to work after a few tries. If you consistantly get this error after many tries, open the build-gnuradio script in a text editor and search for the line http://git.gnuradio.org/git/gnuradio.git and replace it with https://github.com/gnuradio/gnuradio.git.

PYBOMBS

If for some reason Marcus Leech’s script does not work for you, then you can try Pybombs which is another method (and the officially recommended one) that can be used to install the latest GNU Radio from source. Below we show the commands on how to install GNU Radio using this method.

```
sudo apt-get update
sudo apt-get install git
git clone git://github.com/pybombs/pybombs
cd pybombs
sudo ./pybombs config
```

After running config you will be asked to configure several options. Leave them all as the default *except* for the installation prefix which you should set as /usr/local. To use the default options simply press enter at each line. Now install gnu radio and gr-osmosdr using pybombs with the following commands.

```
sudo ./pybombs install gnuradio
sudo ./pybombs install gr-osmosdr
sudo ldconfig
```

Note that in some cases you may need to run `./pybombs install gnuradio` first without sudo privileges and then after receiving an error with sudo. The installation process will take around 1-3 hours to install depending on your PC and internet connection speeds.

If you want to install GNU Radio from your distributions repository, rather than from source you can first check what version will be installed with

```
apt-cache policy gnuradio-dev
```

If the version is modern enough (3.7.3+), then you can force Pybombs to install GNU Radio from the respositories by running the following before running ./pybombs install gnuradio.

```
./pybombs config forcebuild ' '
```

If you can install from the repositories you can save a significant amount of time during installation.

PACKAGE MANAGER

The standard apt-get procedure can also be used to install GNU Radio, but these pre-packaged versions are often not the latest version. The easiest way to install a fairly modern version of GNU Radio (3.7+) is to use Ubuntu 14.04 or newer. Note that even if you use a newer version of Linux, you still won't get the newest version of GNU Radio which is why we recommend either of the first two methods over this method. To install GNU Radio from repositories first use the following command at a terminal to check what version of GNU Radio will be downloaded.

```
apt-cache policy gnuradio-dev
```

If the version of GNU Radio is 3.7.3 or higher then you can simply install GNU Radio with the following.

```
sudo apt-get install gnuradio
```

Installation will take about half an hour when installing from the repositories.

LINUX LIVE DVDS WITH PREINSTALLED RTL-SDR SOFTWARE

There are several Linux distributions available now that contain the RTL-SDR drivers and related software pre-intalled and some are even bootable as a Live

DVD. A live DVD will allow you to boot Linux from a DVD or USB drive without the need to actually commit to installing Linux. Most distributions also come with a wide range of amateur radio and other RF tools. Out of the list below we recommend Andy's Ham Radio Linux or the GNU Radio Live DVD the most.

KB1OIQ - ANDY'S HAM RADIO LINUX

Andy's Ham Radio Linux is an amateur radio centric distribution. It contains rtl-sdr drivers, Fldigi, NBEMS, Gpredict, earthtrack, xcwcp and qrq, XLog and cqrlog, flrig and grig, xnec2c, fl_moxgen, aa-analyzer, owx, VOACAP, glfer, Xastir, gqrx, gEDA, GNU Radio Companion, quisk, direwolf, linamc, FreeDV, wsjt-x, Micro-Fox 15 Config, and a TinyTrak3 configuration program.

http://sourceforge.net/projects/kb1oiq-andysham/

GNU RADIO LIVE DVD

A Linux distribution base on GNU Radio programs. Contains rtl-sdr drivers, GNU Radio release 3.7.3, gqrx, Mode-S/ADSB aircraft transponder receiver, GrOsmoSDR GNU Radio blocks, GNSS-SDR.

http://gnuradio.org/redmine/projects/gnuradio/wiki/GNURadioLiveDVD

KALI LINUX

Kali Linux is a security and penetration testing centric distribution. Kali Linux comes with GNU Radio and Wireshark preinstalled.

http://www.kali.org/

PENTOO

Pentoo is another security and penetration testing centric distribution. Comes with GNU Radio and RTL-SDR drivers preinstalled.

http://www.pentoo.ch/

PORTEUS

Porteus is a very light weight distribution optimized for small size, speed and efficiency.

http://www.porteus.org/

BEAGLEBONE BLACK SDR IMAGE FILE

This is a useful image file designed to be used with the BeagleBone black embedded Linux PC. The image file contains the Ubuntu 14.04 OS along with preinstalled SDR software and drivers like the RTL-SDR drivers and GNU Radio. The amount of software preinstalled is expanding, but at the time of writing it has the following preinstalled: GNURadio 3.7, keenerd's rtlsdr bundle, gqrx, multimode, LTE-Cell-Scanner, LTE-Tracker, multimon, rtl_flex_noX, SuperKuh's Dongle Logger – pyrtlsdr, rtl_433, SDR-J, rtl_sdr wide spectrum analyzer, DSD 1.7, RTLAMR, RTL_FM_Python.

http://www.kd0cq.com/2014/08/packed-full-beaglebone-black-img-file-rtl-sdr-gnuradio-gqrx-lots-more-on-ubuntu-14-04/

RTL-SDR MISC. INFORMATION

RTL-SDR CRYSTAL TOLERANCE

The RTL-SDR is a mass Chinese manufactured device with poor crystal tolerances with up to a +-150 PPM offset. The crystal oscillator in the RTL-SDR is its clock source (the heartbeat of the circuit). The RTL-SDR uses a 28.8 MHz oscillator which may be out by a few kilohertz. The end effect is that known frequencies may not be exactly where you expect them to be. See the Calibrating the RTL-SDR section or the Setting the PPM offset section for information on how to calibrate the RTL-SDR to remove the frequency offset.

Some RTL-SDRs come with upgraded crystals. Ones that use an SMD crystal will probably have a tolerance of around +-30 PPM, and ones with a TCXO will have a tolerance of +-1 PPM. RTL-SDRs with a TCXO will not need any PPM offset to be set, or will only need an offset of 1 or 2.

RTL-SDR DC SPIKE

The E4000 dongle (and FC0012/13) will have a large spike in the centre of the spectrum no matter where it is tuned. This is expected as it due to the way the E4000 is designed. To remove the DC spike with the E4000 and FC0012/13 dongles, the “Offset Tuning” option in the SDR# configure menu can be used. Offset tuning will do nothing for the R280T/R820T2.

The R820T/R820T2 also has a small yet noticeable DC spike at the centre, but usually clicking Correct IQ in SDR# is enough to completely remove this.

RTL-SDR SPURS

Due to the design of the dongle there will always be spurious (unwanted) signal noise spikes at multiples of 28.8 MHz which is the frequency of the internal oscillator. You may also see strong spurs at multiples of the USB clock frequency of 48 MHz and multiples of the USB data rate of 480 MHz.

There is unfortunately not much that can be done about this other to know where the spikes are and to ignore them. An oscillator spur looks like a thin solid line on the RF spectrum or waterfall. It will also likely sound like a single audio tone on the LSB/USB modes. To check if a signal is a spur from the oscillator, simply divide the suspect signals frequency by 28.8 MHz. If the result is an integer it is probably a spur. A spur from something like USB may look like many spurs clustered together.

RTL-SDR ADC BIT DEPTH

The number of physical bits used by the RTL2832U ADC is 8, however the effective number of bits of an ADC is usually lower, and for the RTL-SDR may only be around 7.

RTL-SDR SENSITIVITY

The RTL-SDR with an R820T tuner has an approximate average sensitivity of about -130 dBm over it's receivable range. Below are some sensitivity results performed by HB9AJG. The full report of his results can be found at http://f6fvy.free.fr/rtl_sdr/Some_Measurements_on_E4000_and_R820_Tuners.pdf.

Frequency (MHz)	Sensitivity E4000 (dBm)	Sensitivity R820T (dBm)
24	N/A	-127
52	-139	-134
110	-139	-134
145	-141	-134
435	-139	-135
700	-136	-136
1000	-129	-137

OPTIMIZING TUNING

The RTL-SDR receives best when a signal is tuned in near the centre (but not directly on the center) of the spectrum / waterfall. There is a slight sensitivity drop off near the spectrum's edges.

OPTIMIZING DECIMATION - IMPROVING RECEPTION QUALITY IN SOFTWARE

Decimation is a clever way get more bits out of an ADC by oversampling a signal and then processing it in software. Decimation works essentially by sampling the same data point several times and then averaging it, thus reducing the quantization noise of an ADC. Every time a sample is decimated, we gain about 0.5 bits of resolution. By using decimation on the 8-bit ADC of the RTL-SDR it is possible to achieve an effective 9-10 bits of final resolution, thus improving the signal to noise ratio.

Decimation works by oversampling. Thus, to increase the amount of decimation performed the RTL-SDR sample rate should be set as large as possible. Software like SDR# automatically performs decimation depending on the sample rate, and IF (tuned area) bandwidth. By setting the sample rate to a high value such as 2.4 MSps or 2.8 MSps you are increasing the amount of decimation performed and thus improving the quality of the signal received.

In SDR# the amount of decimation used can be calculated by halving the sample rate until it is smaller than the IF bandwidth. For example, if we are listening to a broadcast FM station with a bandwidth of 250 kHz, then with a 2.4 MSps sample rate we would have 2400/2 = 1200, 1200/2 = 600, 600/2 = 300, so decimation would have been done 3 times, and we would gain about 1.5-bits of resolution. For a narrowband FM station with a bandwidth of about 12.5 kHz, we would have 2400/2 = 1200, 1200/2 = 600, 600/2 = 300, 300/2 = 150, 150/2 = 75, 75/2 = 37.5, 37.5/2 = 18.75. Thus we would have decimated 7 times gaining about 3.5 extra bits of resolution.

The effect of decimation cannot be seen in the RF spectrum or waterfall as it would limit the amount of visible bandwidth. However, there are special drivers which can enable the RF spectrum and waterfall to see what the decimated signal looks like. See the custom gain control and decimation drivers section.

RTL-SDR IMAGES

When using a bandwidth of 0.25 MHz there will be images of strong signals at every 250 kHz. When using a bandwidth of 1 MHz, there will be images of strong signals at every 1 MHz intervals. And so on.

An image is essentially an unwanted duplication of a real signal appearing at a frequency multiple. The only true way to get rid of these is to use a preselector on the front end. See the Preselector Filters section for more information.

RTL-SDR TEMPERATURE FREQUENCY DRIFT

As the dongle warms up the crystal oscillator will so too causing the received frequency of signals to drift over time. Once the dongle reaches thermal equilibrium the frequency drift will stabilize. Usually this is not a problem for most applications if you just remember to allow a few minutes for the dongle to heat up after first plugging it in. The final frequency offset can be adjusted by setting the PPM offset value.

Typically dongles with the SMD crystal tend to drift a lot less compared to the through hole can type oscillators.

USING MULTIPLE DONGLES WITH THE SAME SERIAL NUMBER

Some RTL-SDR dongles may all have the same serial number. Most software won’t worry about the serial number, but some software required each dongle that is plugged in to have a unique serial number. This can mean that if there are multiple dongles plugged in, the software will only recognise the first dongle.

The serial number is stored in the RTL-SDR’s EEPROM which is a seperate chip on the PCB. Note that some dongles come without an EEPROM installed, and thus their serial number is fixed and cannot be changed. If your dongles have an EEPROM, then you can modify the serial number by using rtl_eeprom which is a command line EEPROM programming tool for RTL2832 based DVB-T receivers. Instructions for using rtl_eeprom can be found at http://manpages.ubuntu.com/manpages/trusty/man1/rtl_eeprom.1.html. Simply use the -s string to make the serial number a unique value for each RTL-SDR that you want to connect.

MAX NUMBER OF DONGLES PER USB PORT

The theoretical limit of a USB 2.0 bus controller with a *device* throughput of 256 Mbit/s would be about 5 dongles. In practice, the maximum limit will likely be 3 to 4. If you want to connect more dongles you can buy a PCIe USB card to add more USB 2.0 controllers.

USING THE RTL-SDR ON A LINUX VIRTUAL MACHINE

The RTL-SDR works well on virtual Linux operating systems running through VMWare Player. Virtual box tends to not work very well due to its poor USB implementation. When running in VMWare Player, make sure you set the USB host speed to USB 2.0 in the VM settings (it's set to USB 1.1 by default) and ensure that you give the VM sufficient RAM and processing power for your applications. We reccommend setting the virtual machine to have at least 2 processing cores and at least 4 GB of RAM.

Often it is good to use a Virtual Machine as some Linux RTL-SDR projects require installation of the older version 3.6 of GNU Radio, and some require installation of GNU Radio 3.7+. The two GNU Radio's cannot be installed together on the same system as they conflict with one another. But with a virtual machine you can run two seperate instances of Linux with different versions of GNU Radio installed.

Some ready to go VMWare Linux images can be found at http://www.traffictool.net/vmware/. We prefer to use Lubuntu 14.04 and above, but you may choose your favorite distribution.

INPUT STATIC PROTECTION DIODE

After the antenna input the dongles have a diode that will shunt any large voltages produced by static electricity to ground. This helps to protect the sensitive front end of the tuner chip (R820T/E4000 etc). In the past to save costs some dongle manufacturers did not include the diode on the circuit, which caused damage to many dongles after people connected them to antennas with a static charge on them. These days there have been no recent reports of dongles coming without static protection diodes, though it still pays to check. See further down for an image of a typical RTL-SDR board with the input protection diode labelled.

Even though a dongle has a static protection diode it is not a guarantee that the dongle will be protected. Always discharge outdoor antennas before plugging them in.

RTL-SDR CURRENT USAGE

Current use varies between tuner models and even dongle brands. We measured current use using an inline USB current measurement dongle. The full sized R820T uses approximately 60 mA of current when not in use and 260 mA of current when in use. The mini R820T dongle uses about 60 ma when not in use and 240 mA when in use. The Terratec E4000 model needs approximately 60 mA when not in use and about 170 mA when in use.

REMOTE CONTROL

Some resellers of RTL-SDR dongles opt to include a remote control with the dongle. This sometimes causes confusion as the remote control appears to do nothing. The remote control is only used when the dongle is used as a DVB-T TV tuner, with the official manufacturer drivers. It serves no purpose when used with the SDR drivers.

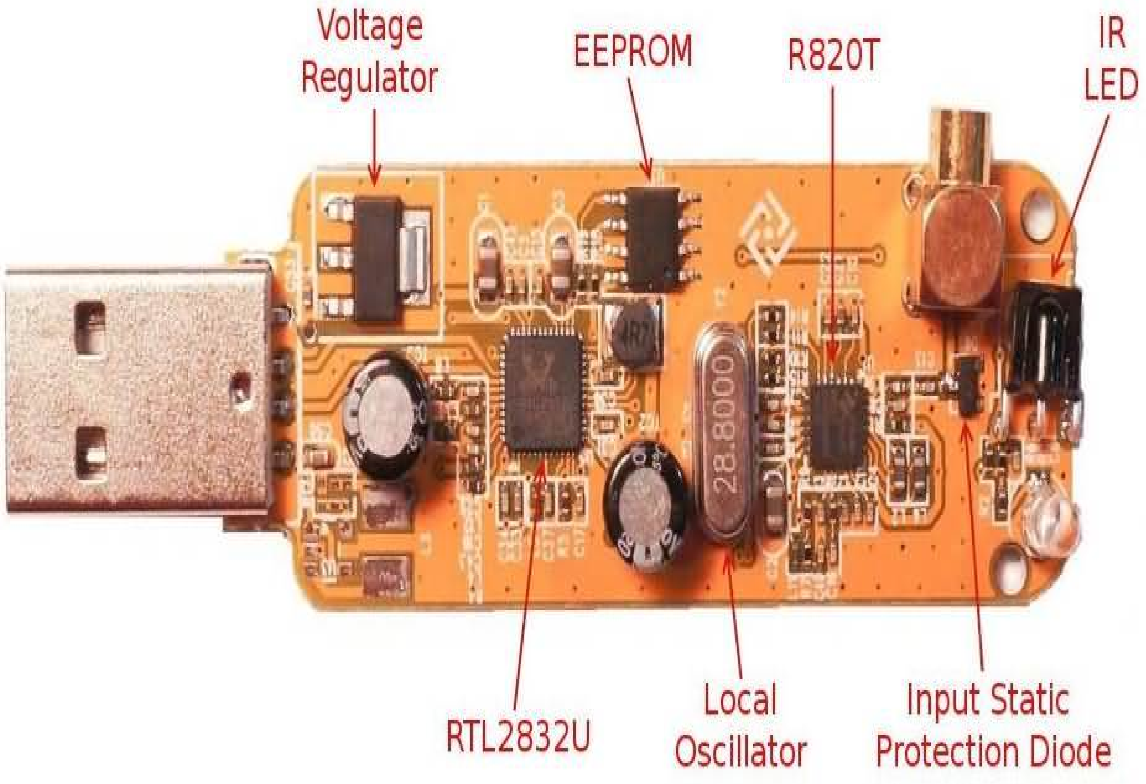
INCLUDED CD

Some resellers also opt to include a small CD with the RTL-SDR dongle containing some DVB-T drivers and TV software on it. Most of the time the DVB-T software on these CDs is pirated and will stop working after a few weeks or months due to the copy protection. If you want to use the RTL-SDR as an SDR then you should not install anything from these CDs anyway. If you do want to watch DVB-T TV, then see the DVB-T TV Linux or Windows Guide section in this book.

DEAD ON ARRIVAL FAILURE RATE

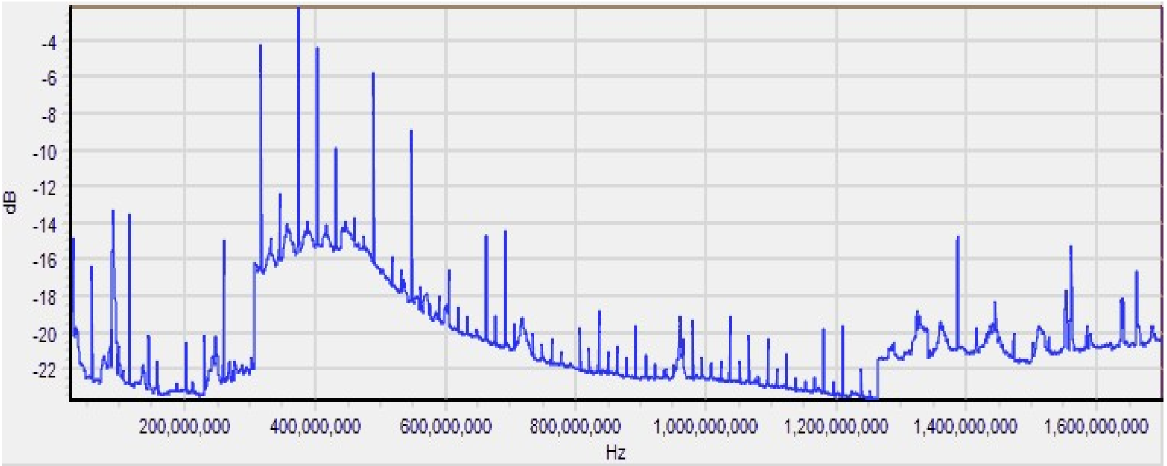
Although quality control is slowly improving, at the moment we estimate the dongle dead on arrival failure rate to be at around 1-2%. This means that out of every 100 dongles you buy, there is likely to be 1 or 2 that arrive in a faulty condition from the factory. See our Troubleshooting Guide in the previous chapter for information on determining if your dongle arrived faulty.

IMAGE OF A TYPICAL RTL-SDR DONGLE CIRCUIT BOARD



RTL-SDR NOISE FLOOR

Here is an image showing a frequency sweep over the RTL-SDRs entire range from 24 MHz to 1.7 GHz with no antenna attached and the gain set to maximum. As can be seen from the graph, the RTL-SDR spurs are clearly visible. These large spurs mainly appear at multiples of 28.8 MHz, indicating that they are from the local oscillator. We can also see from the graph that the noise floor of the RTL-SDR is not flat. At around 305 MHz the noise floor rises sharply and then slowly ramps down again before it rises a little again at 1.264 GHz.



RTL-SDR IMPROVEMENTS AND MODIFICATIONS

LOW NOISE AMPLIFICATION

The amplifier used in the dongle is while considered poor quality for serious work, still ‘low noise’ enough for most applications. However there are third party external LNAs (**L**ow **N**oise **A**mplifiers) that can be used to improve the signal quality. A poor amplifier like the one in the RTL-SDR with a noise figure of <4.5dB will introduce noise causing the signal to be degraded, especially as the gain is increased. The term noise figure is used to measure the amount of noise an amplifier will introduce into the signal. By using a high quality LNA with a noise figure <1dB this noise can be reduced.

One good low cost LNA that we recommend is the LNA4ALL. Its link can be found on our Buy RTL-SDR Dongles page at http://www.rtl-sdr.com/buy-rtl-sdr-dvb-t-dongles/. There are also LNA kits available on Ebay, however these tend to require good surface mount soldering skills. A DIY LNA design for the RTL-SDR can be found here https://github.com/loxodes/rtl-sdr-lna. Habsupplies in the UK also supply high performance filtered LNA’s. Their website is http://ava.upuaut.net/store/index.php?route=product/category&path=72_73.

When positioning the LNA, you want it to be as close to the antenna as possible. The main goal of the LNA is to amplify the signal at the antenna, in order to help reduce the effects of noise that can enter through the coax transmission cable and to help reduce attenuation over long cable runs. In other words, it makes the received signals bigger at the antenna so that any noise or losses introduced by the coax cable will be negligible.

A LNA will be of most use at frequencies above 50 MHz. At HF frequencies (<50 MHz) the ambient RF noise is so high that a high quality LNA won’t make much difference compared to a more ordinary linear amplifier. The manufacturer of the LNA4ALL also makes the LNA4HF which uses a more linear amplifier and is designed for HF operation.

Don’t be tempted to buy standard TV amps as these will perform no where near as well as a proper modern broadband LNA like the LNA4ALL, although if you use very long runs of coax cable they still may help.

Some LNA’s can be powered with “phantom power”. This is where the required DC 5V is injected into the coaxial cable and is used to power the LNA or other device through the cable. The LNA4ALL can be modified for phantom power, and the HAB supplies LNA’s all support phantom power by default. It is possible to modify the RTL-SDR to provide phantom power, see the Phantom Power heading below.

RECEIVING LF/MF/HF (0 - 30 MHZ)

The lowest frequency that a R820T dongle can reliably tune to is about 24 MHz. No RTL-SDR dongle will tune to the entire LF/MF/HF band (0-30 MHz) without modifications or extensions. There are three methods that will allow your RTL-SDR to receive LF/MF/HF frequencies.

1. The direct sampling mode modification.
2. An upconverter circuit.
3. Modified experimental drivers.

1) DIRECT SAMPLING MODE

The RTL-SDR can be told to run in a special “direct sampling” mode, which with a small hardware mod allows the dongle to tune to the LF/MF/HF frequencies between 0 - 28.8 MHz where ham radio and many other interesting signals are found. No upconverter circuit is required with this mod.

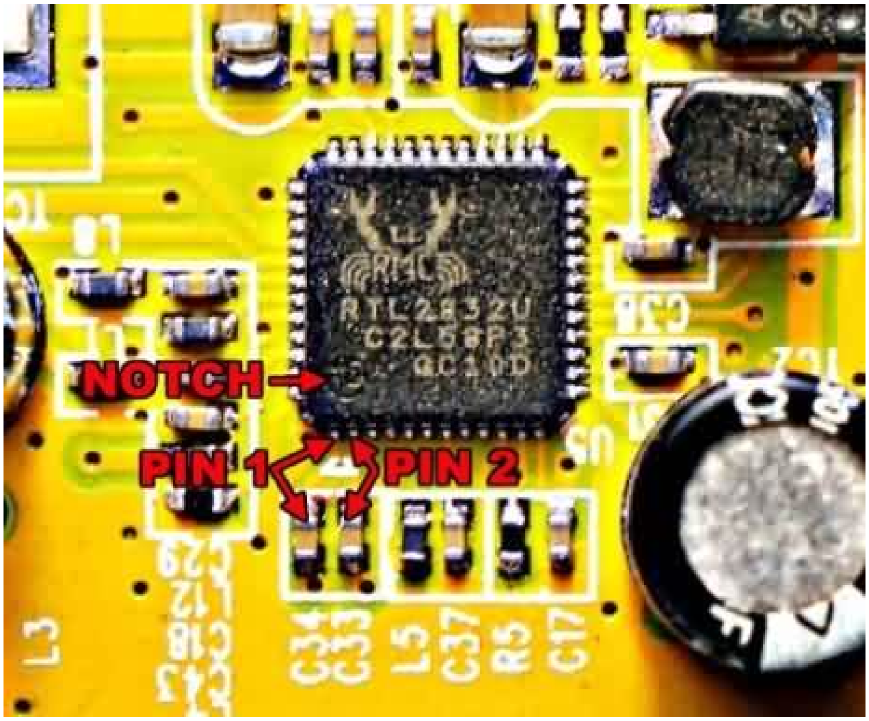
The difficulty with direct sampling is that a hardware modification to the dongle is required. Performance with the direct sampling mod can be the same as, or even better than with an upconverter - assuming appropriate low pass/band pass filtering is used. Upconverters can introduce conversion losses.

The direct sampling mod essentially bypasses the tuner chip and allows RF signals to directly enter the RTL2832U chip by *directly attaching an antenna to the RTL2832U microchip pins*. Care must be taken with this mod as it bypasses any electrostatic protection the dongle has. However, in practice damaging the chip through static discharge is a rare occurrence if care is taken.

INSTRUCTIONS

To do this mod, you will need decent soldering skills. It involves opening the dongle casing, and soldering a “random wire” or “long wire antenna” to either pin one or two, or alternatively pin four or five of the RTL2832U chip. Even a short wire a few meters long will be sufficient for picking up broadcast AM stations. Some people report that even pressing a damp finger onto the pin will pick up signals. However, for most amateur radio signals, a longer wire is probably required.

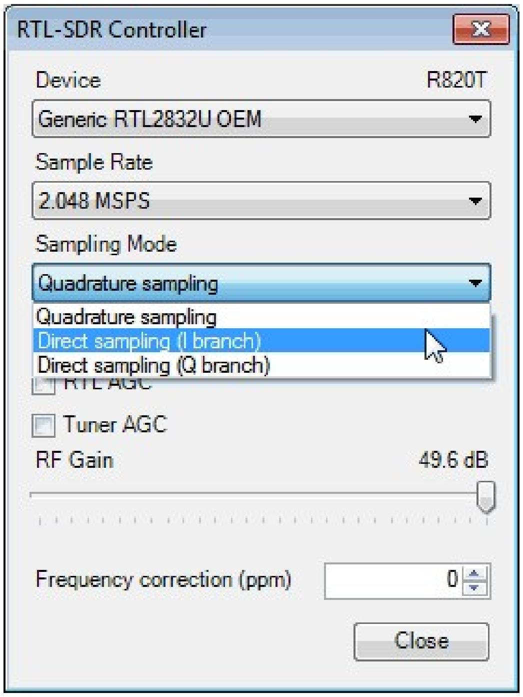
Pins one and two are located below the circle indentation on the RTL2832U chip. It might be easier to solder the wire to one of the two capacitors which are connected directly to pin one and two.



If you are using an R820T based dongle, it may be best to use pins four or five for this mod. Due to the R820T's circuit architecture these pins on the RTL2832U are actually unused under normal operation. Using pins four or five will allow you to use normal quadrature sampling for VHF/UHF and direct sampling as well for HF on the same dongle. The main problem with pins four and five is that they do not have a capacitor connection like pins one and two do, thus making them very hard to solder to. Reddit user pskato has taken a photo of his soldering (http://img42.imageshack.us/img42/3272/htjo.jpg) to pins four and five and can be used as a reference. Some newer R820T2 RTL-SDR dongles made in the first quarter of 2015 and onwards now come with easy to solder to connections for pins 4 & 5. These dongles make performing the direct sampling mod significantly easier.

Once you have performed the hardware mod, to activate it you will need to change some software settings. In SDR#, go to the configure menu and change the sampling mode to “Direct sampling (I branch)”, or ”Direct sampling (Q branch)”. The I branch corresponds to pins one and two and the Q branch corresponds to pins four and five. Pin three is not connected to anything. Direct sampling can also be enabled in HDSDR by using the direct sampling option found in the ExtIO settings window (same window where you choose the sampling rate). To enable direct sampling in GQRX, add the string “direct_samp=1” to the device string input box in the Configure I/O devices window.

If a wire antenna is connected to one of these pins and the correct direct sampling branch is selected, you will be able to receive signals between 0 - 14.4 MHz. Above this you will see images of the 0 - 14.4 MHz range due to Nyquist aliasing. If you want to receive frequencies between 14.4 and 28.8 MHz you will need to build a band pass filter for this range. See the Designing Simple Filters with RFSim99 section below for information on how to design some simple but effective low pass and bandpass filters.



RTL-SDR Controller

Device: R820T

Generic RTL2832U OEM

Sample Rate: 2.048 MSPS

Sampling Mode: Quadrature sampling

□ RTL AGC

□ Tuner AGC

RF Gain: 49.6 dB

Frequency correction (ppm): 0

IMPROVEMENTS

The direct sampling mod described above can be improved on further. Since the RTL2832U uses two pins to create a differential input (+ and - signals) for each I and Q input, a balun can be used to connect both pins one and two or pins four and five to the antenna. Using a balun also has the added advantage of giving you additional antenna static discharge protection. Essentially this mod involves finding or winding your own balun and connecting the balun to the two pins and the other end of the balun to your antenna. A balun is easily made by using a ferrite toroid core such as the FT370-43 and some enamelled wire, both readily available from hobbyist electronics stores. Most experimenters have used a trifilar balun winding.

The balun can also be used to match the 50 or 75 Ohm antenna input to the input impedance of the RTL2832U. The input impedance of the RTL2832U is not known for certain, but most people place it at around 200 ohms. Thus, a 4:1 impedance conversion for the balun seems to be the accepted value which at 200 Ohms would match a 50 Ohm antenna/feedline. However note that some people have reported better performance with larger impedance conversions on the order of 8:1 or 10:1.

Here Mikig shows a schematic for the direct sampling mod on an E4000 tuner http://mikikg.files.wordpress.com/2012/08/rtl2832u-dc-mods.pdf, Mikig recommends using a Minicircuits T1-6T-KK81+ RF transformer (http://www.minicircuits.com/pdfs/T1-6T-KK81.pdf) as the balun for best performance. Here is another writeup by I6IBE on his direct sampling mod http://www.radioamatoripeligni.it/i6ibe/rtl2832hf/dongle.htm, in his implementation he used a bifilar balun. An article about the direct sampling mod in Japanese can be found here http://blog.livedoor.jp/bh5ea20tb/archives/4263275.html, this article has good images that show a neat and efficient way to wind a trifilar balun by twisting the three wires together first. Interestingly, the author of the Japanese article recommends that 10 windings gave him the best performance, indicating that perhaps the RTL2832U input impedance is higher than the expected 200 Ohms.

Most experimenters of this mod find that FM interference can be a problem and thus low pass filters might be necessary for good performance. Some users also report that adding in a low noise amplifier (LNA) can vastly help improve reception. This is because the tuner chip is bypassed in direct sampling. Under normal operation is the tuner chip gives amplification to the received signals through its internal amplifier.

These days there are modified RTL-SDR dongles being sold with the direct sampling mod having been already applied and with things like filters and amplifiers added. Some good pre-modified dongles are as follows:

KN0CK: http://www.kn0ck.com/HF_SDR/

DXPatrol: http://www.ct1ffu.com/site/?option=com_content&view=article&id=178&Itemid=104

Janielectronics: http://janielectronics.com/index.php?route=product/product&path=18&product_id=103

Soft66RTL: http://zao.jp/radio/soft66rtl/

ZJChao: http://amzn.com/B00MC9WPWS

2) UPCONVERTERS

Upconverters move LF/MF/HF frequencies ‘up’ into the receivable range of the RTL-SDR. An upconverter is connected to your antenna before the RTL-SDR dongle and simply increases the frequency of all received signals by about 100 MHz. Good upconverters for HF use will have a low pass filter included helping reduce interference from signals above 30 MHz.

The amount that an upconverter upconverts can matter. Some upconverters use a 100 MHz local oscillator which would add 100 MHz to all frequencies. However, some users have complained that the 100 MHz puts some popular ham bands directly on top of broadcast FM stations. Although these FM stations shouldn’t appear after the upconversion, sometimes images can still show up. Theoretically, the low pass filter used in most upconverters should prevent any interference from FM bands, but some very strong stations can overcome the filter. Most upconverters now use a local oscillator frequency of 125 MHz, which avoids FM broadcast stations.

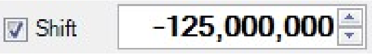
There are now several commercial upconverters available. One high quality and very popular upconverter that we recommend is the Nooelec Ham-it-up. This upconverter is based on an opensource design and it is of very high quality production and decent price. It increases the frequency of all signals by 125 MHz, and has a low pass filter installed. See www.rtl-sdr.com/buy-rtl-sdr-dvb-t-dongles/ for information on purchasing the Ham-it-up.

Other popular choices include the CT1FFU upconverter http://www.ct1ffu.com/site/index.php?option=com_content&view=article&id=178&Itemid=104 and the Janilab upconverter http://janielectronics.com/index.php?route=product/product&product_id=97. One RTL-SDR blogger has done a big writeup on all the upconverters that can be used with the RTL-SDR here http://blog.kf7lze.net/2012/09/14/round-up-of-rtlsdr-upconverter-choices/.

HOW TO USE AN SDR RECEIVER WITH AN UPCONVERTER

To use an SDR receiver like SDR# with an upconverter you will need to tune to your desired signal + the upconverters local oscillator frequency. For example, the ham-it-up upconverter has a local oscillator frequency of 125 MHz, so to see a signal at 6 MHz you would tune to 125 MHz + 6 MHz = 131 MHz.

If you don’t want to always have to do the math, you can set an offset in the software receiver. The offset setting will allow you to tune directly to your desired frequency. In SDR# you can set the ‘Shift’ value to -125,000,000 for an upconverter with a 125 MHz oscillator (don’t forget the minus sign).



☑ Shift -125,000,000

3) EXPERIMENTAL DRIVERS

There are experimental drivers by programmer Oliver Jowett available that will allow the R820T tuner to tune all the way down to around 1 MHz. This driver is experimental and is still being updated. In the future it may be included in the standard RTL-SDR driver. The driver can be downloaded from Oliver Jowett’s experimental GitHub repository https://github.com/mutability/rtl-sdr/. A precompiled Windows binary for SDR# can be found at http://bit.ly/W2Zzqv (Mirror: https://dl.dropboxusercontent.com/u/43061070/rtlsdr_OLIVERHFMOD.dll). Simply download this file, rename it to rtlsdr.dll and then copy it into the SDR# folder overwriting the old one (you may wish to back up the original file first). Now you can open SDR# and tune to HF frequencies. Assuming you have an HF antenna plugged in (such as a simple random wire antenna) you should be able to easily receive some broadcast AM stations at around 0.5 to 2 MHz.

Testers report that sensitivity is best from 14 MHz and higher, but acceptable performance can be found with lower frequencies. This mod is no where near as sensitive as the true direct sampling hardware mod or with an upconverter, but using an LNA like the LNA4HF and a low pass filter can significantly improve reception.

RTL-SDR dongles that use the R820T2 tuner will also see much better performance than those with the R820T tuner when using this experimental modification. This is because the R820T2 has a much larger IF filter which better makes use of the mechanism that this mod is using to receive these HF frequencies.

DIRECT SAMPLING WITH NO HARDWARE MOD DISCOVERY

Note that the experimental driver discussed below has been superseded by the drivers by Oliver Jowett discussed above which give significantly better reception performance. We recommend using Oliver’s drivers for HF reception now, but in rare cases this mod may give better performance on some frequencies.

There is also a discovery found by Reddit forum user Anonofish. Anonofish found that when tuning to frequencies between 3686.6MHz - 3730MHz, the

E4000 RTL-SDR dongle acted as if it was in direct sampling mode, passing the radio signal directly past the tuner and into the RTL2832U chip. This method worked through local oscillator leakage. Unfortunately this was patched in the newer rtl-sdr drivers.

However, there is still a way to test this. An RTL-SDR developer, tejeez has merged code into another developer, Keenerds experimental rtl-sdr branch available at https://github.com/keenerd/rtl-sdr. This can be compiled on Windows or Linux. To compile RTL-SDR on Windows we have compilation instructions section in the next chapter.

If you don’t want to compile this, there is a precompiled replacement rtlsdr.dll file for SDR# available here http://www.bit.ly/1iNu8Vb (Mirror: https://dl.dropboxusercontent.com/u/43061070/rtlsdr_NOHARDWAREMOD_VC.dll). To use this file, download it, rename it to rtlsdr.dll, then copy it into the SDR# folder overwriting the old one. Now in SDR#, enable the RTL AGC. For this dll file the RTL AGC function has been replaced with the direct sampling hack. After enabling this, tune to an HF frequency. Try the AM bands first which will be around 0.5-2 MHz. You will be able to tune up till around 14.4 MHz. Make sure you have a HF antenna plugged into the dongle in the normal antenna port. This hack appears to work best with the R820T tuner and is also reported to work even better with FC0013 and R820T2 tuners.

Since this hack came out many people have reported being able to get acceptable performance after adding an LNA and/or low pass filter.

POSITIONING YOUR RTL-SDR DONGLE

Traditionally, desktop scanner radios are situated next to the user - in the house, or in the ham radio shack. The antenna is then usually situated a fair distance away, usually on the roof of the house or up a mast. This type of setup can require long runs of coaxial cable to connect the radio to the antenna. This can be a problem as the longer the coaxial cable run between the antenna and radio is, the more signal you will lose due to attenuation and noise.

With the RTL-SDR, the radio hardware is the dongle itself. The dongle converts the analogue signal from the antenna into a digital signal which is sent to a PC over USB. USB cables transmit digital data and digital data does not degrade RF signal quality. Therefore the RTL-SDR dongle should be situated right next to the antenna (or as close as possible), replacing the lossy coaxial cable in favour of non-lossy digital USB cable instead.

However, there is a problem with USB cables. For USB 2.0 the maximum length of a USB extension cable is about 5 meters (16 feet) and for USB 3.0 about 3 meters. This limitation is due to digital data timing requirements (although poor quality cable may also not work due to voltage drops). The USB protocol expects a sent data packet to be received within a certain time frame. If the cable is too long, the data will not be received within the expected time as the data packet takes time to travel through the cable.

A 5 or 3 meter cable may not be long enough to reach between your PC and antenna. Fortunately, there are two solutions around this problem.

1. Use an active USB repeater cable or hub.
2. Use an embedded computer like the Raspberry Pi or a remote computer to transmit the signal wirelessly or over ethernet cable to your PC.

1) USE AN ACTIVE USB REPEATER CABLE OR HUB

An active USB repeater cable works by using a repeater circuit every few meters on a USB cable. The repeater circuit will receive a USB data packet, reset the timing and then send it further down the line once again at full speed. The maximum length using repeaters is recommended to be 30m for USB 2.0 and about 18m for USB 3.0.

If you have a USB hub, you can also use it as a repeater with non-active cables. Just place 5 meters of USB cable from the PC to the hub and then another 5 from the hub to dongle.

Active USB cables or hubs can be found cheaply on Amazon or Ebay, see our Buy RTL-SDR dongles page http://www.rtl-sdr.com/buy-rtl-sdr-dvb-t-dongles/.

Note: Do not use USB to Ethernet extenders. These extenders simply convert the USB cable into Ethernet cable, which is capable of long cable runs at **USB 1.1** speeds only. USB 1.1 speeds are too slow for the RTL-SDR dongle, which requires at least USB 2.0.

2) USE A MINI EMBEDDED OR REMOTE COMPUTER TO TRANSMIT THE SIGNAL WIRELESSLY TO YOUR PC

This is an interesting solution where a small embedded PC such as a Raspberry Pi or Beaglebone running a Linux server is placed next to the dongle and antenna. The embedded PC is plugged into the RTL-SDR dongle, which then can be to used to transmit the raw IQ radio data via TCP/IP (over the internet or your network). Transmission can be done wirelessly over WiFi, or over wired Ethernet using a program developed for the RTL-SDR called *rtl_tcp*.

One thing to be concerned about with this solution is the bandwidth of your TCP network connection. Since rtl_tcp uses quite a large amount of data depending on the bandwidth you have chosen, you will need a fast network connection. For 2 Msps / 2 MHz of bandwidth, the commonly used 100Mbit wired connection will usually suffice. For WiFi, an N rated connection is probably required. In practice you will probably find that you will need to drop the sample rate to around 1 MSps. If network speed proves to be a problem with your setup, we recommend using the streaming fm guide in the tutorials section to stream FM rather than raw IQ data.

Installing RTL-SDR on an embedded computer uses the same method as installing it on a regular Linux PC. To install the RTL-SDR drivers on a Raspberry Pi or other Linux based embedded PC follow the Linux installation instructions shown in a previous section.

It may also be wise to use the modified “rtl_fm_python” from https://github.com/th0ma5w/rtl_fm_python. This software is a modified version of rtl_fm which allows the tuned frequency and modulation mode in rtl_fm to be modified remotely via a simple web interface. This is very useful for remote embedded servers.

Refer to the RTL_TCP projects section below for information on how to set up and use rtl_tcp effectively.

Note for Raspberry Pi users - since the Raspberry Pi uses a USB based ethernet controller, its ethernet speed may not be sufficient to keep up with the data rate of rtl_tcp when a high sample rate is used. For best performance we recommend using a slightly more powerful embedded computer such as a Beaglebone Black.

UPGRADING THE RTL-SDR DRIVERS IN SDR#

KEENERDS DRIVERS

SDR# comes with the standard RTL-SDR drivers written by Osmocom. While these drivers are good, there is now a newer branch of experimental drivers by developer Keenerd that are much better overall. With Keenerds drivers you can expect a slightly lower noise floor and less out of band interference from strong signals. There are also other improvements such as faster retuning times which may help when using the drivers with scanner plugins. While the osmocom team is working on updating the official drivers, a final release date of these drivers is unknown. To update the drivers follow these steps:

1. Go to http://igg.kmkeen.com/builds/ and download the latest zip file. This zip file contains the latest version of Keenerds RTL-SDR drivers for Windows.
2. Open the downloaded zip file and open the sdrsharp folder.
3. Copy all the non .exe files into the sdrsharp folder, making sure to replace any duplicate files.
4. Now in the sdrsharp folder delete the rtlsdr.dll file.
5. Rename the librtlsdr.dll file to rtlsdr.dll.

The new drivers are now installed and ready to be used. Take note that if you at any time update SDR# by running install.bat again, Keenerds drivers will be overwritten and you will need to perform the above steps again.

CUSTOM GAIN CONTROL AND DECIMATION RTL-SDR DRIVERS

Keenerds drivers and the Osmocom drivers that come by default with the RTL-SDR are good, but they hide some inner workings of the gain stages. In the RTL-SDR there are actually three seperate gain stages, the LNA gain, the Mixer gain and the VGA gain. The gain slider used in the standard drivers is simply a function of these three gains. For users experienced with the RTL-SDR it would be better if each gain stage could be controlled manually. In some cases this can allow better tuning by adjusting the gains perfectly so that no overloading occurs. A modified driver which exposes all three gain stages can be download from http://www.rtl-sdr.ru/page/novaja-modifikacija-sdr-rtls-sdr-drajvera, note that this page is in Russian so you may need to use Google translate.

This driver also offers a special decimation feature. Decimation is a clever method used to essentially increase the number of effective bits in an ADC by oversampling the signal. When decoding to audio, SDR# and other similar software does decimation automatically, but it does not apply it to the visual waterfall. These drivers allow you to apply decimation to the visual RF spectrum and waterfall allowing you to more clearly see a signal and its structure. Another benefit is that decimation will allow you to use a larger bandwidth, and thus get more decimation, but still maintain a small enough visible bandwidth so that tuning is made easier.

DONGLE SHIELDING AND NOISE REDUCTION

The RTL-SDR dongles have a problem in which they are susceptible to receiving interfering signals that enter via some way other than through the antenna. The solution to this problem is to simply enclose the dongle in a metal box which acts as an RF shield. The USB connector on the dongle and the shield of the RF port should make good electrical contact with the box. Therefore, the metal used in the box should be as conductive as possible - copper is the best, but also the most expensive. Some metals like aluminium may be anodized which can severely reduce their conductivity.

If you are enclosing the RTL-SDR in a box together with other components such as an LNA or upconverter, be sure to shield the RTL-SDR separately from these other components. It has been shown that the RTL-SDR dongle itself emits RFI which can interfere and be amplified in the other devices.

Avoid using the RTL-SDR dongle near power supplies for devices such as laptops and mobile phones. These power supplies typically use something called a ‘switch mode’ power supply which is prone to introducing lots of RF noise into the nearby vicinity. If you are running the dongle on a laptop, ensure you are running on the battery and not power supply. Desktop PCs also use switch mode power supplies so the best PC to use the RTL-SDR on is a laptop running on battery. Switch mode RFI will usually look like multiple solid thin lines on the waterfall or as multiple fuzzy lines.

If a battery powered laptop is not viable, it is a good idea to use a standard or active USB cable to get the dongle as far away from the PC as possible to reduce PC RF noise interference. See the Positioning Your RTL-SDR Dongle section for more information.

COMMON SOURCES OF NOISE

- Power lines. A faulty power line which is arcing electricity can create huge amounts of white noise which can drown out signals.
- Switch mode power supplies.
- Most electrical devices, such as monitors, TVs, appliances etc.

- Ethernet cables. Unshielded Ethernet cables can output huge amounts of RFI.
- Car alternators.
- The dongle itself. You will see spikes at integer multiple of 28.8 MHz (e.g 28.8 MHz, 2 x 28.8, 3x28.8 and so on). These spikes come from the local oscillator used in the dongle.
- Universal Serial Bus (USB). USB typically uses a 48 MHz clock, and you may see spikes at multiples of these frequencies. You may also see very large noise spikes at 480 MHz, 960 MHz and 1440 MHz which are multiples of the 480 MHz USB data transfer rate.
- Ethernet over power. Devices which send ethernet over housing power cables can really introduce a lot of noise. Some internet providers also transmit broadband over unshielded power cables which takes out most HF and some VHF frequencies.

DONGLE USB CABLE GROUNDING IMPROVEMENT

According to the USB spec, the RTL-SDR dongle has an electrical grounding design flaw (though note that this USB spec is sometimes debated). In USB devices the USB cable shielding is usually not meant to be connected to the RF ground on the device, but rather to a separate ground. However, in the RTL-SDR, the USB cable shield connection *is* connected to the RF ground. This connection can cause the USB cable to act as an antenna, allowing unwanted RF signals to enter through the cable into the ground of the dongle.

Fortunately, there is an easy fix for this design flaw. The fix is to simply remove the shield connection from either the USB cable, or the dongle itself and then create a secondary ground such as a bunch of foil wrapped around the dongle with the USB cable ground connected to the foil and not connected to the dongle ground. Or, if you are enclosing the RTL-SDR into a metal case, connect the USB cable ground to the metal case instead. We recommend using a non-anodized aluminium or other conductive metal box for shielding.

This simple fix can make a huge difference especially if you are having front end overload problems with signals that are too strong, such as local broadcast FM stations.

TESTING CHANGES TO NOISE PERFORMANCE

Without a proper method for testing it can be difficult to determine if your shielding efforts actually have any impact on the noise received. One easy way to test this is with the rtl_power software. The rtl_power software is described in more detail in a later section Heatmap Band Scan. To do a noise analysis, first download rtl_power which comes as part of the official Osmocom RTL-SDR release of which a Windows version can be downloaded from http://sdr.osmocom.org/trac/attachment/wiki/rtl-sdr/RelWithDebInfo.zip. If you are running on Linux and have installed the RTL-SDR drivers, rtl_power can be accessed from terminal.

Next plug in your dongle, and connect a 75 Ohm dummy load to the antenna input. A dummy load can be purchased online or from your local electronics shop for a few dollars. To take a noise scan over the entire 24 MHz to 1.7 GHz range of the R820T dongle use the following code.

```
rtl_power -f 24M:1.7G:1M -g 50 -i 15m -1 noise.csv
```

This will scan the entire spectrum once using maximum gain for 15 minutes and output the data to noise.csv. The scan needs to run for at least 15 minutes in order to correctly average the data. Longer scans will result in more accurate scans.

To plot the data you can open the CSV file in Excel or equivalent software and create a scatter plot of the last column of data. To compare changes we recommend doing a scan before you make any changes and then one after you have made the changes in order to compare the two.

References:

http://sinnet3000.blogspot.com.br/2013/03/reducing-emi-in-rtl-sdr.html

http://imgur.com/a/pQ8dM

USB CABLE FERRITES

Another simple but highly effective improvement is to add clip on USB ferrites to the USB extension cable. These ferrites are specifically designed to remove interfering signals from USB cables. Make sure you get the clip on ferrites that are intended for USB cables. Other wrongly sized ferrites or wrap around toroid ferrites may actually harm performance by filtering the actual desired digital USB signals out.

Clip on ferrites can be bought cheaply from Amazon. See http://www.rtl-sdr.com/buy-rtl-sdr-dvb-t-dongles/ for a link.

PRESELECTOR FILTERS

The RTL-SDR like most low cost wideband SDR radios has no preselector at the front end. A preselector is an electronic analogue hardware circuit which filters out any frequency that is not in the filters passband. There are several different types of preselectors such as bandpass, lowpass, highpass or bandstop filters.

- Bandpass Filter: Only allow signals within a certain frequency range to pass. For example between 1 GHz and 1.2 GHz for ADS-B.
- Low Pass Filter: Only allow signals below a certain frequency to pass.
- High Pass Filter: Only allow signals above a certain frequency to pass.
- Bandstop Filter: Blocks a certain frequency range (e.g. 88-108 MHz), which is useful for blocking strong interfering signals like pagers and the FM broadcast band.

Using a preselector on a wide band device like the RTL-SDR would mean that only a small area of the 24-1700 MHz receivable range would be usable. However, preselectors will drastically reduce interference in the tuned frequencies. Most analogue radios and high end SDRs have built in preselectors. It is difficult to have preselectors in wideband radios like the RTL-SDR as you would need to have multiple hardware preselectors for each chunk of tunable frequencies. This is because you would need different values for the analogue filter components depending on the frequency ranges you are interested in.

But still, if you are building a system for a single purpose, like for only receiving ADS-B signals for instance, you will want to use a preselector to drastically reduce radio interference and improve your signal to noise ratio. In this YouTube video https://www.youtube.com/watch?v=wcr-PX662Go the experimenter went from 800 ADS-B messages received per second to 1200 messages per second with a bandpass preselector enabled. Most good upconverters will also use a lowpass preselector filter to filter out interfering signals above about 30 MHz.

A ready made ADS-B filter which includes a preamp can be bought from http://ava.upuaut.net/store/index.php?route=product/product&product_id=85.

If you find that you have interfering signals from a strong signal source, such as pagers or the FM broadcast band you can use a band stop filter. Instead of just passing a specific band, a band stop filter will block or “stop” a certain range of frequencies. Broadcast FM bandstop filters are sometimes referred to as “FM

Traps”. A very cheap but effective band stop filter is the MCM Electronics FM Trap. See our Buy RTL-SDR dongles page online for links http://www.rtl-sdr.com/buy-rtl-sdr-dvb-t-dongles/.

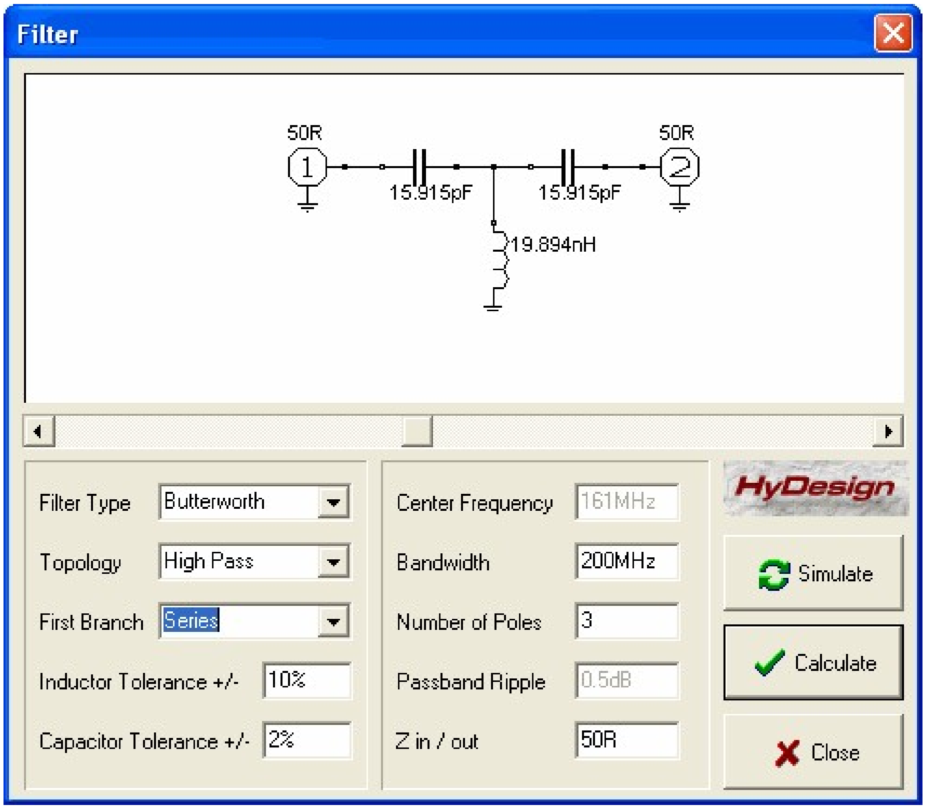
Surface mount and some inline filters can be purchased from Minicircuits http://www.minicircuits.com/products/Filters.shtml. Most surface mount filters can also be purchased from minicircuits with a evaluation board containing female SMA connectors. Without the evaluation board you will need good soldering skills to connect the band pass filter to a breakout board. Unfortunately, filters from Minicircuits are often quite expensive. Low cost ADS-B preselectors can be bought on Ebay, just search for “B1602 SAW filters Epcos”. For UHF frequencies like ADS-B, homemade PCB based hairpin filters can also be easily constructed, see http://www.w1ghz.org/filter/Recipes_for_Printed_Hairpin_Filters.pdf.

We note that more expensive SDRs like the Funcube Dongle Pro+ have built in preselectors which greatly increase the cost, but also improve performance.

DESIGNING SIMPLE FILTERS WITH RFSIM99

Homemade low pass, high pass and band pass filters can be constructed out of capacitors and self wound inductors. All you need are the capacitors, some enamelled wire to create the inductors and a soldering iron. These types of simple capacitor and inductor based filters will work well for HF and possibly low VHF frequencies, but will really struggle to work properly at frequencies above around 300 MHz. Above 300 MHz you will need to use components like SAW or cavity filters. To design capacitor and inductor based filters a tool like RFSim99 can be used, which can be downloaded from http://www.electroschematics.com/835/rfsim99-download/. Note that you will need a 32-bit OS like Windows XP to run this program. This limitation can be worked around using Windows Virtual XP mode or by running XP in a Virtual Machine.

To design a filter simply open RFSim99 and then go to **Tools->Design->Filter. E**nter the parameters for the filter you need, then press Calculate then Simulate to see the graph of simulated results.



Filter Type: Butterworth

Topology: High Pass

First Branch: Series

Inductor Tolerance +/-: 10%

Capacitor Tolerance +/-: 2%

Center Frequency: 161MHz

Bandwidth: 200MHz

Number of Poles: 3

Passband Ripple: 0.5dB

Z in / out: 50R

Here are some simple tips for filter design.

- **Filter Type** - Here you can choose between a Butterworth or Chebyshev filter. Both filters use the same amount of components. The difference between the two is that a Chebyshev filter can have a tighter roll off at the expense of some ripples in the pass band. A tighter roll off means that frequencies near the pass band will have greater rejection.
- **Topology** - Here you can choose between a Low pass, High pass or Bandpass filter design.
- **First Branch** - Sets whether the first component is connected in series or in parallel.
- **Tolerances** - Tolerances of the components you are using.
- **Centre Frequency** - If you are designing a bandpass filter then this is the centre of the frequency range that you would like to pass.
- **Bandwidth** - For a bandpass filter the amount of bandwidth you would like in the pass area, for low pass and high pass filters the start or end of the filter pass/stop band.

- **Number of Poles** - Increasing the number of poles in the filter increases the number of components required, but also makes the filter response have a tighter roll off.
- **Passband Ripple** - Only used when using the Chebyshev filter type. Increasing this value tightens the filter roll off, but increases the rippling in the pass band.
- **Z in / out** - The impedance of your filter, set to 50 Ohms or 75 Ohms for most applications.

The inductors can be made by winding some enamelled wire around a cylindrical object like a pen. To calculate the inductance for a self made air wound inductor you can use the inductor design tool in RFSim99. To access this tool go to **Tools->Component->Inductor**. Enter the length, diameter and number of turns, then press calculate. Adjust the inputs until you get the inductance specified by the filter design.

COAX STUB FILTER

One easy alternative to a using a hardware preselector is to use a “coax stub filter”. A coax stub filter is a type of notch filter. A notch filter is just a band stop filter with a very small “stop” area. These sub filters are excellent for ‘notching out’ (blocking) strong interfereing pager frequencies.

To make a coax stub filter you will need a “T” junction coax connector. The coax from the antenna will enter the T junction from the bottom of the T and exit out one of the sides. On the remaining side there will be a length of coax cable, called a “stub”. The length of the stub will determine the frequency of the notch.

To calculate the stub length use this formula, where freq is the frequency you would like to block and VF is the velocity factor of the particular coax cable you are using.

```
stub length = 300/freq (MHz) x VF x ¼
```

The formula converts the band stop frequency into a wavelength first, then adjusts the length using the coaxial cable velocity factor which will slow down the speed of light in the cable and then finally multiplies by ¼ to get the ¼ wavelength.

Related Links

www.nerdsville.blogspot.com/2012/05/rtlsdr-part-4-cutting-out-noise-using.html

ADDING PHANTOM POWER / BIAS TEE

It is possible to modify the RTL-SDR so that it can provide 5V DC power up the coaxial line which can be used for powering devices such as LNA's and active antennas. To perform this modification, all you need is some careful soldering skills and a 1uF to 4.7uF surface mount inductor to create a bias tee. A bias tee allows DC power to exist on the coaxial cable, but prevents it from entering the dongle where it could cause damage. A simple bias tee involves using a capacitor to block DC entering the dongle, and an inductor to block RF entering the power supply.

To perform the mod all that is required is that the BAV99 ESD protection diode should be removed, and then one end of the inductor should be soldered on to the center conductor of the coax output and the other end of the inductor should be connected to the 5V USB pin via a wire. For a more clear description see this post by Fabio: http://fabiobaltieri.com/2014/06/22/lna/.

An inductor should behave as a short circuit to RF frequencies. However, real world inductors are not perfect, and choosing the perfect inductor for a bias tee can be difficult, especially with a simple bias tee design involving one inductor like this one since there is a tradeoff to make. When choosing an inductor there are two main parameters you should worry about, the inductance and the self resonant frequency (SRF). Higher inductances block RF frequencies better, but they will have a lower SRF. The SRF is the frequency at which an inductor blocks the best, but at higher frequencies it stops working as an inductor, and it's parasitic capacitance takes over. An inductor will still work above its SRF, but it's blocking ability will reduce as the frequencies increase. You can think of an inductor above its SRF as a capacitor that also allows DC to pass.

Generally, if you are only interested in higher frequencies such as ADS-B at 1090 MHz, it is best to choose a smaller inductance value such as 220 nH, but with a high SRF at around 1 GHz. However, in most cases an inductor with a value between 1uF to 4.7uF will work just fine if it has a reasonably high SRF (100-200 MHz). A 1uF to 4.7uF inductor should be able to cover the entire frequency range of the RTL-SDR with RF losses below 1dB across the spectrum.

If you perform this mod you will need to take care not to accidentally short the antenna port, or use a DC shorted antenna. Shorting the USB 5V supply could cause the inductor to fry, and could damage your USB port.

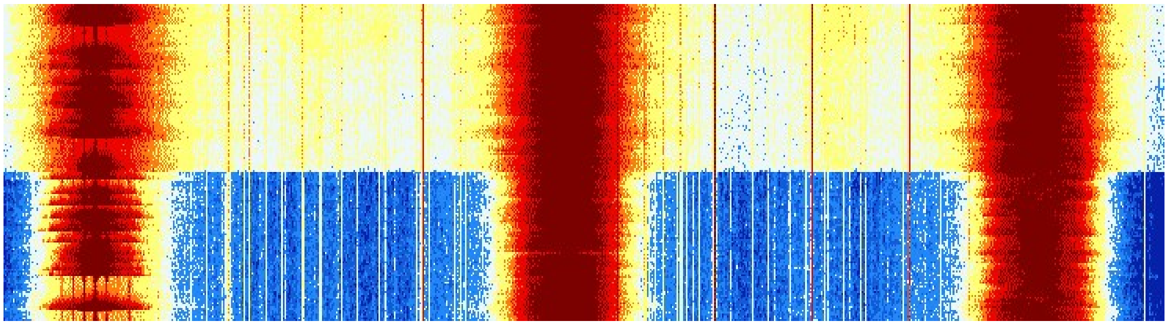
IMPROVING THE STOCK ANTENNA

Most RTL-SDR packages come with a small single length antenna. Some manufacturers provide a telescopic antenna instead, which is much better that the single length whip. But if you didn't get the telescopic antenna or don't have another antenna and you're stuck with the standard stock antenna, here are some quick tips that can drastically improve its performance. These tips are also applicable to other similar whip antennas.

GROUND PLANE

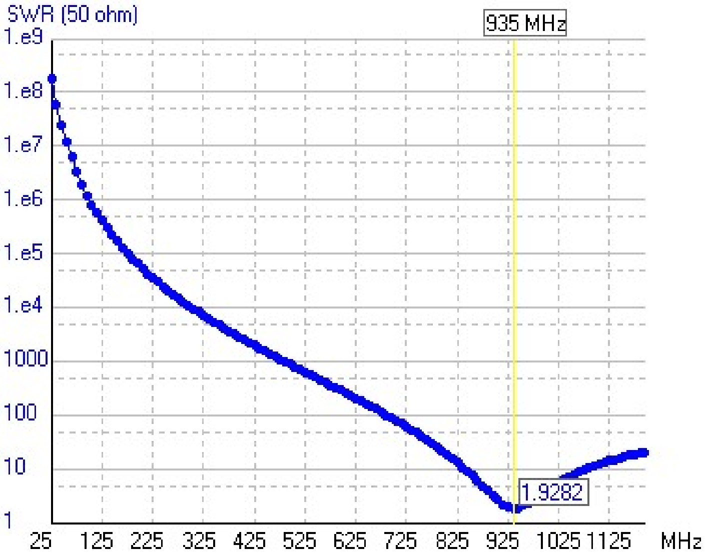
The performance of the stock antenna that comes with many RTL-SDR units can be improved simply by placing it on a flat metallic surface that has a radius similar to or larger than the length of the whip, which for the standard antenna provided with most dongles is around 11.5 cm. Even if you can only find a metal disc with a smaller radius, it will help in most cases to improve reception. The metallic surface could be anything - most easily it could be something like the lid from a coffee tin, or the tin cover of a metal cookie or chocolate box.

Placing it on a metallic surface completes the antenna as a quarter wave ground plane. This shifts the optimum SWR point to a lower frequency and also usually reduces it too. See the SWR section in the Antenna chapter for more information about SWR. Below we show a waterfall image at broadcast FM frequencies comparing the antenna without (top) and with (bottom) a metallic ground plane with radius 12 cm. The noise floor is drastically reduced with the ground plane.

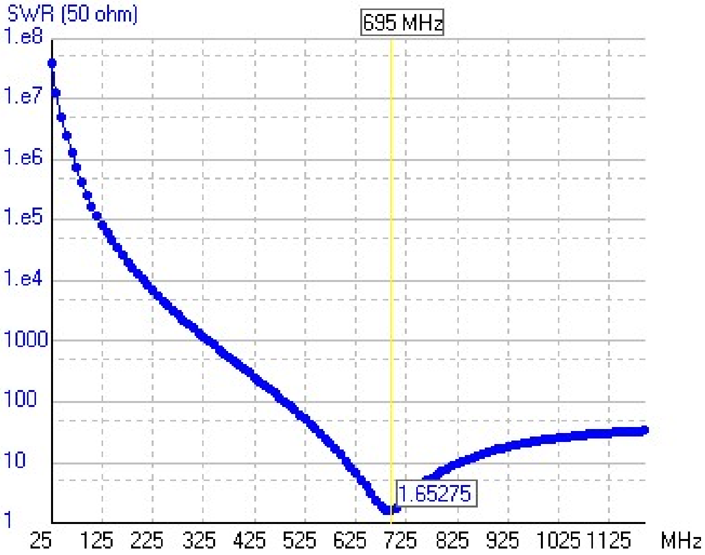


Below we show some 4NEC2 SWR simulations of the stock whip. We can see that for the stock whip with no ground plane (other than the 1 cm radius magnetic base ground plane that is part of the antenna) the antenna is resonant at about 935 MHz and the SWR is very high for lower frequencies. When used with a 6 cm ground plane the SWR is reduced and the resonant frequency has been reduced to 695 MHz. With a ground plane with a 11.5cm radius the resonant frequency is reduced to 645 MHz, but it is a case of diminishing returns.

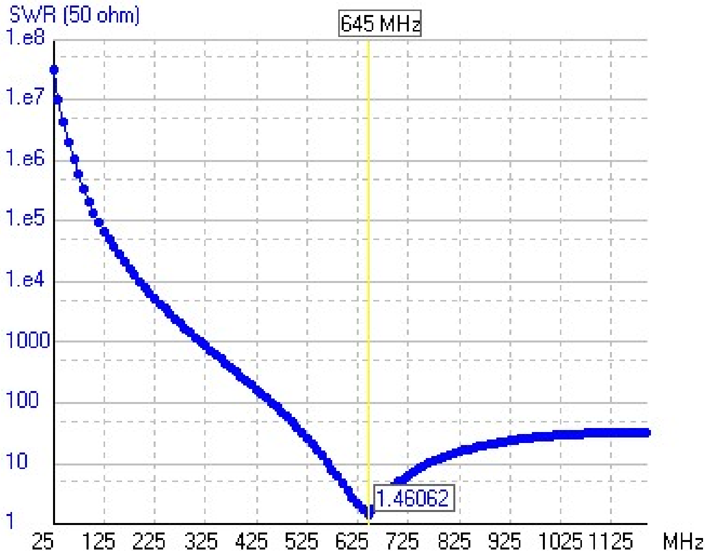
Thus to conclude, to improve the performance of the stock antenna at frequencies below 935 MHz, place the antenna on a flat metallic surface to create a ground plane.



SWR of the stock whip with no added ground plane.



SWR of the stock whip with a 6cm radius ground plane.



SWR of the stock whip with a 11.5cm radius ground plane.

ADS-B

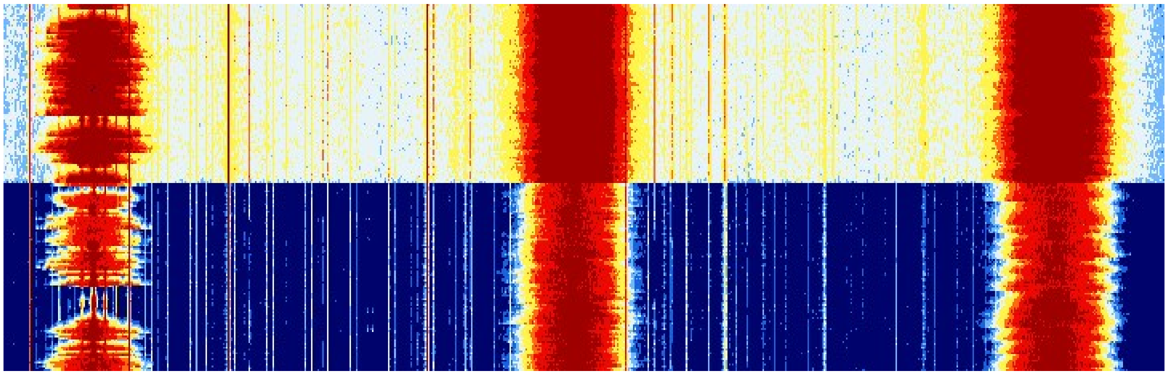
To improve the stock antennas performance at 1090 MHz for ADS-B, cut the whip length down to around 6.9 cm and then use a ground plane with a 6.9 cm radius.

COAX CHOKE

Another improvement that can drastically improve performance is to create a common mode coax choke, which in ham circles is also known as the so called “ugly balun”. A choke will essentially stop unwanted currents in the coax shield. This helps to remove the effect of the coax cabling becoming a part of the antenna and undesirably skewing the radiation pattern. See the Antenna Chapter for more information about radiation patterns.

To create a coax choke all you need to do is find a cylindrical object about 2 - 3 cm in diameter, like a marker pen and create about 5 - 10 turns of coax cabling around it. If you have a ferrite core such as a clip on ferrite use this as the winding centre for even better performance. With a ferrite core you can use less turns for the same performance as more turns without the ferrite. Make the winding as close to the antenna base as possible. Below we show an image of an example winding with a USB ferrite.

Below we show the effect a common mode choke can have on reception. The lower half of the image is with the choke in place using a clip on ferrite with four windings. The upper half shows reception with no choke in place. As can be seen, the noise floor is significantly reduced.



REPLACE STOCK ANTENNA COAX CABLE

Many RTL-SDR users say that the coax cable on the stock antenna is bad, but in actuality it is not that terrible for VHF frequencies because it is only 1m long. The coax used in most units is a type of low quality RG174 with a very thin braid. We measured the insertion loss over a 1m stock antenna cable and found that from 0 - 600 MHz the loss was around 0.5dB - 2dB and from 700 to 900 around 3 dB. At 1 GHz and higher the loss rises significantly and becomes around 6dB - 8dB. Below we show a close up of the stock antenna coax cable with the outer sheathing removed. Note that there are some stock antennas come with a much worse cable which is just a twisted pair of thin wires, these usually have a PAL connector instead of an MCX one and have significantly higher losses at higher frequencies.

It is possible to replace the stock coax with a better cable. Simply peel off the sticker at the bottom of the antenna and use a flat head screw driver to pry open the metallic disc a the bottom. This will expose the cable which can then be unsoldered.

CALIBRATING THE FREQUENCY OFFSET OF THE RTL-SDR

The easiest way to calibrate the RTL-SDR is by tuning to a known frequency and then adjusting the PPM offset until the signal is centred. Make sure to do this after the dongle has been running for a while as the dongles are known to have a "warm up" time, where the PPM offset will change as the temperature slowly increases. In SDR# the PPM offset setting can be adjusted from the "Configure" menu.

KALIBRATE

It is also possible to calibrate the RTL-SDR automatically using a command line program called Kalibrate. However, note that Kalibrate is sometimes known to give inaccurate results and instead we strongly recommend manually determining the PPM offset with a known signal. Kalibrate requires that there are local GSM signals receivable by your antenna to do the calibration. GSM signals are used because they transmit frequency correction data which Kalibrate can use to determine the clock offset. Kalibrate can be downloaded from https://github.com/steve-m/kalibrate-rtl. A pre-built Windows version can be found at http://rtlsdr.org/files/kalibrate-win-release.zip - note that you may need to copy in the newer rtlsdr.dll and libusb-1.0.dll files from the official RTL-SDR release from the osmocom site to get it to work. To compile and install kalibrate on Linux use the following instructions.

1. Install the prerequisites

```
sudo apt-get install libtool autoconf automake libfftw3-dev
```

2. Download the kalibrate-rtl repository from the GitHub https://github.com/steve-m/kalibrate-rtl either by using git clone or by downloading and extracting the zip file from the page.

```
git clone https://github.com/steve-m/kalibrate-rtl
```

3. In a terminal navigate to the kalibrate folder. Type the following compilation instructions into the terminal.

```
./bootstrap && CXXFLAGS='-W -Wall -O3'
./configure
make
sudo make install
```

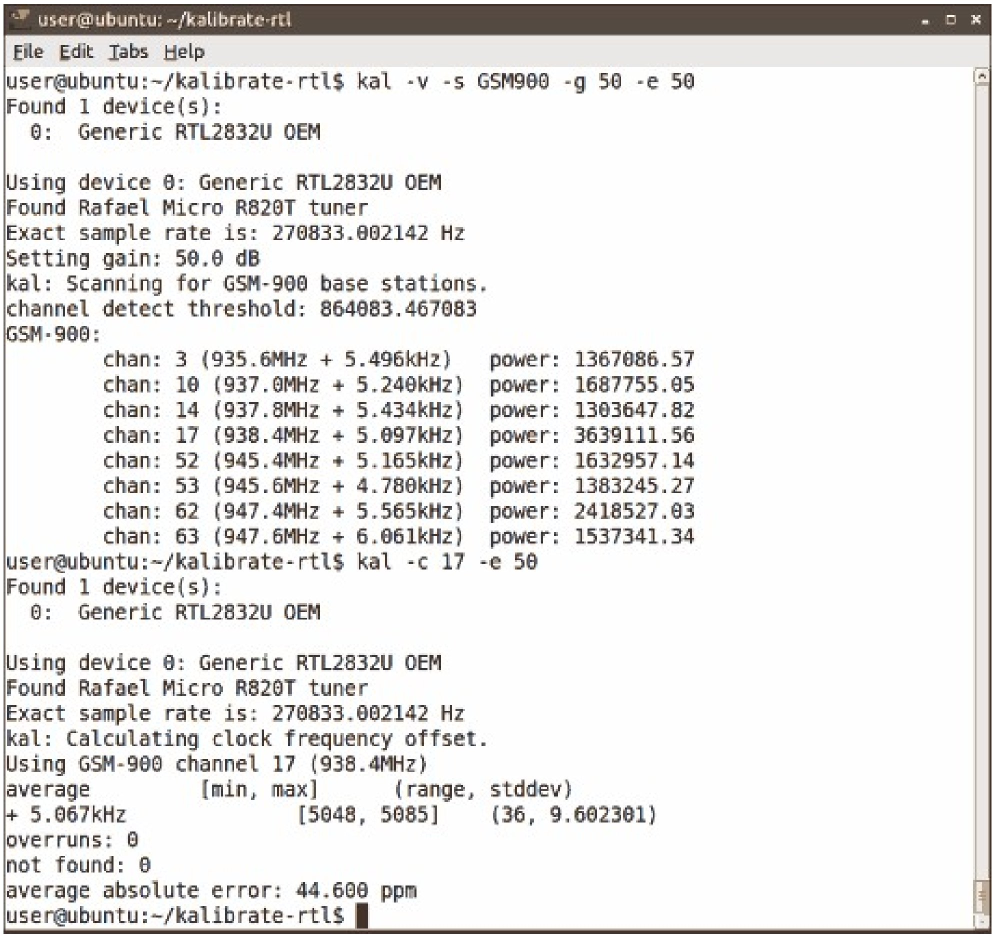
4. Now kalibrate is ready to use by using the command shown below. Replace GSM900 by GSM850, GSM-R, GSM900, EGSM, DCS or PCS depending what GSM systems and frequencies are in use in your particular country. Kalibrate will run for a while searching for GSM stations. Also, set the gain parameter -g to whatever setting works best for you.

```
kal -v -s GSM900 -g 50
```

5. Once kalibrate has found some GSM channels, choose a channel number with strong power from the output displayed in the terminal and then run the following command to determine the RTL-SDR frequency offset in PPM.

```
kal -c channel
```

If kalibrate has trouble finding GSM stations or you find that the PPM offset is giving strange values, you may need to specify to kalibrate an initial PPM offset guess to help it out. You can do this using the -e flag. A good starting guess might be in the region of 30 to 50 ppm.



```
user@ubuntu:~/kalibrate-rtl$ kal -v -s GSM900 -g 50 -e 50
Found 1 device(s):
  0:  Generic RTL2832U OEM

Using device 0: Generic RTL2832U OEM
Found Rafael Micro R820T tuner
Exact sample rate is: 270833.002142 Hz
Setting gain: 50.0 dB
kal: Scanning for GSM-900 base stations.
channel detect threshold: 864083.467083
GSM-900:
        chan: 3 (935.6MHz + 5.496kHz)   power: 1367086.57
        chan: 10 (937.0MHz + 5.240kHz)  power: 1687755.05
        chan: 14 (937.8MHz + 5.434kHz)  power: 1303647.82
        chan: 17 (938.4MHz + 5.097kHz)  power: 3639111.56
        chan: 52 (945.4MHz + 5.165kHz)  power: 1632957.14
        chan: 53 (945.6MHz + 4.780kHz)  power: 1383245.27
        chan: 62 (947.4MHz + 5.565kHz)  power: 2418527.03
        chan: 63 (947.6MHz + 6.061kHz)  power: 1537341.34
user@ubuntu:~/kalibrate-rtl$ kal -c 17 -e 50
Found 1 device(s):
  0:  Generic RTL2832U OEM

Using device 0: Generic RTL2832U OEM
Found Rafael Micro R820T tuner
Exact sample rate is: 270833.002142 Hz
kal: Calculating clock frequency offset.
Using GSM-900 channel 17 (938.4MHz)
average         [min, max]      (range, stddev)
+ 5.067kHz              [5048, 5085]    (36, 9.602301)
overruns: 0
not found: 0
average absolute error: 44.600 ppm
user@ubuntu:~/kalibrate-rtl$ 
```

LTE SCANNER

An alternative to Kalibrate is LTE Scanner, which uses mobile phone LTE basestation signals to get a PPM correction value rather than GSM signals. You will need to have LTE signals in your area and know their approximate frequencies. To install LTE Scanner use the following.

```
sudo apt-get install libboost-all-dev
sudo apt-get install libitpp-dev
sudo apt-get install libfftw3-3
git clone git://github.com/Evrytania/LTE-Cell-Scanner.git
cd LTE-Cell-Scanner
mkdir build
cd build
cmake ..
make
sudo make install
```

Note that if you get errors about FFTW when trying to run cmake and are using a 32-Bit Linux OS, you will need to browse to LTE-Cell-Scanner/cmake/Modules and use a text editor to edit FindFFTW.cmake. In the

FIND_LIBRARY(FFTW_LIBRARY block you will need to add the folder /usr/lib/i386-linux-gnu.

To run LTE-Scanner use the following, replacing the start and end frequencies with values appropriate for your own country.

```
CellSearch --freq-start 749e6 --freq-end 799e6
```

The program will run until an LTE signal is found and it will return the PPM correction.

REMOVING THE IR DIODE

The standard RTL-SDR unit comes with an IR LED diode which is used under DVB-T TV operation for the remote control. This diode has long legs which can act as an antenna picking up strong undesired signals. A simple improvement to the dongle is to simply cut off this IR LED diode from the PCB board. See the Image of an RTL-SDR Circuit Board section for info on where the IR LED is.

REDUCING USB NOISE ON HF

Placing a 1 uF (micro farad) tantalum capacitor across the USB +5V and GND lines can help to reduce reduce noise particularly at HF frequencies. To do this, simply open the dongle casing and solder the 1 uF capacitor across the two outer pins on the USB plug. Be very careful that you connect the polarized tantalum capacitor the correct way by using a multimeter to measure the voltage first. Also be careful not to short anything and try to keep the capacitor leads as short as possible. If done carefully you should be able to easily snap the casing back on.

This helps as capacitors will act as a short to RF signals which should not be on the USB DC power line. A value of 1 uF is enough for almost any frequency to be shorted to ground. A tantalum capacitor should be used as they have lower parasitic inductance compared to electrolytic capacitors and can thus pass higher frequencies much better.

STABILIZING THE FREQUENCY OFFSET

The cheap and low quality oscillator used in most RTL-SDR dongles is okay for most applications, including almost all the projects discussed in the tutorials below. However, some demanding applications like GPS and radio astronomy projects require very stable and very small frequency offsets. The local oscillator will drift in frequency as the dongle heats up. One simple way to stabilize the dongle is to remove its casing or to drill extra ventilation holes. Another method that has been suggested is to add solder to the heat pads under the RTL2832U and R820T chips on the bottom of the dongle circuit board to improve heat dissipation. The author of the SDR for Mariners blog at http://sdrformariners.blogspot.com has also suggested that the dongle be submersed in a can of oil to stabilize the temperature. Immersing the RTL-SDR in a bucket of sand is another method for temperature stabilization.

The best (but also most expensive) way to stabilize the frequency offset is to replace the oscillator with a higher quality one that has a lower PPM offset. You could go even further and replace it with a temperature controlled oscillator (TCXO) which will maintain a stable frequency even as the temperature changes. There are now several vendors that sell TCXO modified dongles and these can be found on our Buy RTL-SDR dongles page at http://www.rtl-sdr.com/buy-rtl-sdr-dvb-t-dongles/.

RTL-SDR PROJECT TUTORIALS

These tutorials are written with the RTL-SDR in mind but most will also be valid for other SDRs and even some hardware radios.

AUDIO PIPING

Most of these tutorials performed in Windows require that something called an audio pipe be set up. An audio pipe is simply a program which routes the audio output from your SDR receiver (e.g SDR#, HDSDR, SDR-Radio) to the decoding program. See Appendix A: Audio Piping for information on how to set up audio piping methods like stereo mix, VB Cable or Virtual Audio Cable.

GENERAL FREQUENCY GUIDE

Here we list some common international radio signals and the frequencies they can be likely be found at. However, note that some of these frequencies could be different for other countries.

Signal	Frequency (MHz)	Type	Description
AM Broadcast Radio	0.535 - 1.605	AM Voice	
Amateur Radio	0 - 30	SSB/Digital Data	Ham radio users use voice and digital data signals
Shortwave Radio	3 - 30	AM Voice/DRM	International radio broadcasts
Simple Walkie Talkies & Cordless Phones	40 - 49	NFM Voice	
CB Radio	26.965 - 27.405	AM Voice	Radio usually used by truckers

Analogue TV	54 - 88	PAL/NTSC	Being phased out in many countries
Radio Controlled Toys	72 - 76	Digital Data	
Broadcast FM	88 - 108	WFM Voice	
Air Traffic Control	108 - 136	AM Voice	Voice communications from airplanes and air traffic towers
ATIS	110 - 129	AM Voice	Voice that repeats weather and other information to pilots
ACARS	131.550	AM Data	System for sending and receiving short messages from aircraft
VDL2	136.975	USB Data	Eventual Replacement of ACARS
NOAA/Meteor-M Satellites	136 - 138	NFM Data	Transmits weather satellite images
VHF Amateur Radio	144 - 148	NFM Voice/NFM Data	Band dedicated for Ham radio
NOAA Weather	~162.4	NFM Voice	Voice weather report
Rail	159-162	NFM Voice/NFM Data	
Maritime	156 - 162	NFM Voice	
Automatic Identification System (AIS)	161.9175 162.025	NFM Data	Used to track ships like a radar

Pagers	157	NFM Data	Messaging System
Analogue TV	174 - 216	Data	PAL or NTSC TV
Digital Audio Broadcast (DAB)	~174 - 239	DAB Data	Digital Radio
Military Communications (USA)	225 - 380	Voice/Data	
Trunking Radio	~400	NFM Voice/Data	Commercial walkie talkies
Weather Balloons (Radiosondes)	~400	NFM Data	Meteorological agency launched weather balloons
Amateur Radio	420 - 450	NFM Voice/Data	
ISM Band	~433	Digital Data	Licence free low power band. Mainly weather stations/power meters etc
LTE	~700	Digital Data	Cell phone fast data connection
Trunking Radios	~860	NFM Voice/Data	
GSM	850 900 1800 1900	Digital Data	Cell phones voice and data
Cordless Phones	900	Digital Voice	Usually with DECT encoding

ADS-B	1090	Digital Data	Used to track aircraft like a radar
GPS	1575	Digital Data	Global Positioning System

ACARS RECEIVING GUIDE

INTRODUCTION TO ACARS

ACARS is an acronym for **A**ircraft **C**ommunications **A**ddressing and **R**eporting **S**ystem. It is a digital communications system that commercial aircraft use to send and receive short messages to and from ground stations. Some messages you might hear on ACARS are OOOI events (messages reporting changes in major flight phases such as **O**ut of the gate, **O**ff the ground, **O**n the ground and **I**nto the gate), digital information sent to the aircraft avionics and text messages to the flight crew. Some ACARS messages will also contain aircraft GPS coordinates, but if your goal is aircraft radar, the ADS-B aircraft radar section will be of more interest to you. Each ACARS message also contains the aircraft registration and fight identification number.

Standard ACARS transmits at a VHF frequency of 131.550 MHz, but there are various other frequencies also in use depending on your location. There are also ACARS messages transmitted via the Inmarsat Satellite network and on the HF bands, but these bands are not as commonly used as the VHF band.

With the RTL-SDR and some decoding software, the ACARS messages around you can be received and displayed live on your PC. As ACARS transmits in the VHF band, reception is very nearly line of sight. But as aircraft fly high in the sky, with a good antenna you should be able to receive messages from high altitude aircraft that are up to 200 miles away. Because ACARS uses vertically polarized signals we recommend a discone, collinear, quarter wave ground plane or j-pole antenna. See the antenna guide section in this book for more information on these antennas. An LNA and bandpass filter may also help but are not necessary unless you have poor reception. See the LNA section and preselector section for more information.

So why receive ACARS? Well apart from watching the messages that come through, most decoder software will also look up the received aircraft registration number on the internet and display information and a photograph of the airplane transmitting. This is a great aid to aircraft spotters and anyone who just wants to see what is flying in the air around them.

In the table below we show some common ACARS frequencies.

Freq (MHz)	REGION / COUNTRY

131.550	Primary worldwide
129.125	USA and Canada Additional
130.025	USA and Canada Secondary
130.425	USA Additional
130.450	USA and Canada Additional
131.125	USA Additional
131.450	Japan/China Primary
131.475	Air Canada Company
131.525	European Secondary
131.725	European Primary
136.700	USA Additional
136.750	USA Additional
136.800	USA Additional
136.900	European Secondary
136.850	SITA North America

136.750	European New
131.580	European New

ACARS TUTORIAL (WINDOWS)

For most of this tutorial we will assume that you have an RTL-SDR dongle installed and working on a Windows PC. If not, please follow the Quickstart Installation Guide section.

There are four commonly used ACARS decoder software programs. These are Acarsdeco2, PlanePlotter, AirNav ACARS 2 and ACARSD. Below we present tutorials on each software. At the moment we currently recommend Acarsdeco2 as the best software to use.

For using PlanePlotter, AirNav ACARS 2 and ACARSD you will also need to have installed a Windows audio piping program installed. See Appendix A: Audio Piping for more information.

First, we will try to receive ACARS messages in SDR# or another similar program such as SDR-Radio to confirm that they are receivable.

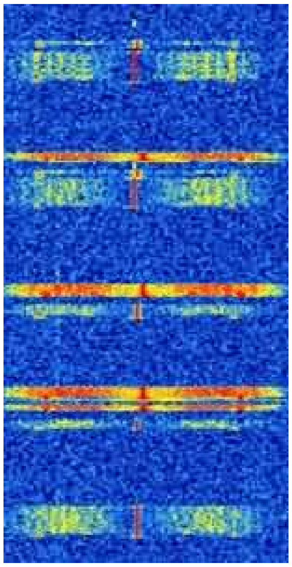
1. Open a general SDR program such as SDR# or SDR-Radio and tune to an ACARS frequency such as 131.500 MHz. Note that we highly recommend using SDR-Radio with multiple VFO's or the multiple VFO plugin for SDR# if you are wanting to use PlanePlotter, AirNav ACARS2 or ACARSD and wanting to track multiple ACARS frequencies.

2. You should see some ACARS signals on the RF spectrum and waterfall. Adjust the RF gain settings so that the ACARS signal is strong and free of out of band interference.

3. Adjust the tuned bandwidth of your receiver to around 5-10 kHz, so that the bandwidth just covers the entire ACARS signal.

4. Set the receiver mode to **AM**, turn off squelch and turn off any noise reduction filters. Using audio AGC may or may not work well, you will need to experiment with what works best. If you are looking to combine multiple ACARS channels on the same audio channel with multiple VFO's it may be best to enable the squelch.

At this point you should be able to see some ACARS signals appearing in the spectrum. ACARS signals come in bursts, and will look something like the waterfall image shown below. If you are outputting the audio to speakers you will also hear a sort of screeching sound when a ACARS packet is received.



In the next sections we show tutorials on how to use the four ACARS decoders mentioned above. We highly recommend using Acarsdeco2 however.

ACARCSDECO2

Acarsdeco2 is a Windows, Linux, Raspberry Pi and OSX compatible command line based multi channel ACARS decoder that directly interfaces with the RTL-SDR. This means that no audio piping is required, and no programs like SDR# are required. In this tutorial we will show how to use the Windows version, but use of the Linux, OSX and Raspberry Pi versions should be very similar. We believe that Acarsdeco2 is probably the best decoder for ACARS available as it has good decoding performance, is capable of decoding up to 3 ACARS channels simultaenously and much easier to set up with multiple ACARS streams than by using multipe VFO's in a program like SDR-Radio of the SDR# multiple VFO plugin. Also, while Acarsdeco2 is a command line program it also has a web interface that can be used to browse the received ACARS messages.

1. First, using SDR# or another program discover your dongles PPM frequency offset and best gain settings appropriate for receiving ACARS signals. Adjust the gain settings until ACARS signals are received strongly without any interference. Record the dongles PPM offset and optimal gain, then close SDR#. If you need help determining the PPM offset see the Setting the

PPM Offset section of this book. Determining the optimal gains and an accurate PPM offset are critical for the operation of Acarsdeco2.

2. Next, download the latest Windows version of Acarsdeco2 from http://forum.planefinder.net/threads/acarsdeco2-up-to-3-channels-acars-sdr-receiver-for-rtl2832-dongle.157/. Unzip the files into a folder on your PC.

3. Now open a command prompt by going to the start menu and typing in “cmd” under search and then opening cmd.exe.

4. Browse to the folder where you have unzipped the Acarsdeco2 files in command prompt. For example, if you stored the files in F:\acarsdeco2, you would type

```
F:
cd acarsdeco2
```

5. Now type the following command to start the software.

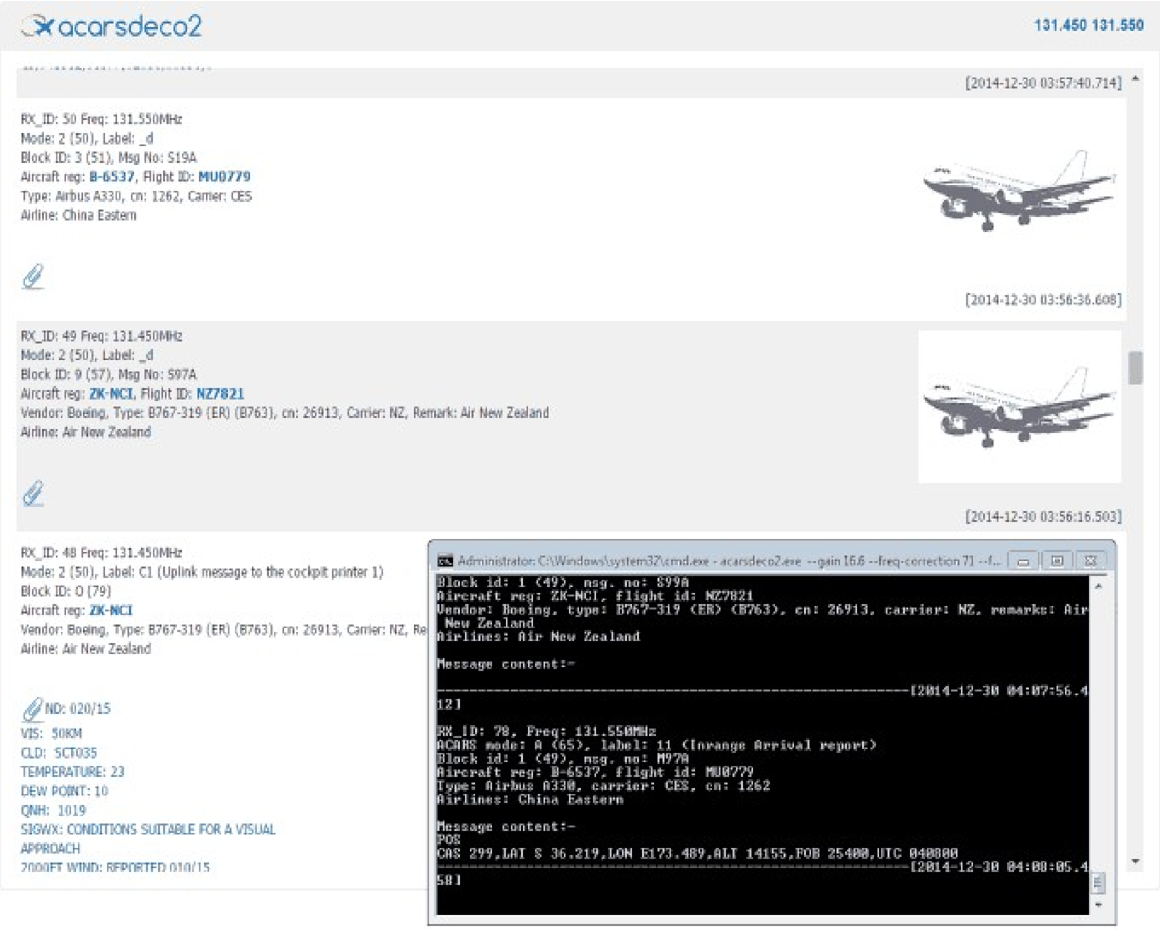
```
acarsdeco2.exe --gain 38.0 --freq-correction 65 --freq 131550000 --freq 131725000 --http-port 8090 --vrs-url
http://192.168.1.10:80 --udp 127.0.0.1:9742 --net 30008
```

6. You should set the flag values (the words with the double dashed before them - explained below) appropriately to your conditions. The last three flags (--vrs-url, --udp and --net) can be omitted for simple operation unless you wish to use those features explained in the flags description below.
--gain - the optimal gain setting you found in step 1
--freq-correction - the PPM offset of your dongle (very important, must be set accurately)
--freq - you can add up to three ACARS frequencies by specifying more than one --freq flag
--http-port - the http port to be used for the Acarsdeco2 web interface
--vrs-url - if you are running Virtual Radar Server at the same time with ADS-B decoding, you can get the images of the aircraft to show up in the Acarsdeco web interface
(not required if not using Virtual Radar Server)
--udp - set the UDP port for transmitting Acarsdeco2 data to PlanePlotter (not required if not using PlanePlotter)
--net - set the TCP port for transmitting Acarsdeco2 data to SBS Basestation (not required if not using BaseStation)

7. Now with acarsdece2 running, open a web browser and browse to http://127.0.0.1:8090 and you should be able to see your ACARS messages being received in real time.

8. If you want to interface with PlanePlotter, you can go into PlanePlotter and go to Options->I/O settings and then deselect “Acars reception from audio input” and select UDP/IP data from net, making sure that the port is set to 9742. Now when you start PlanePlotter it will gather data from Acarsdeco2.

To get real aircraft images to show up in the Acarsdeco web interface you must be running Virtual Radar Server and be receiving the same aircraft with another RTL-SDR dongle and with ADS-B data.



[2014-12-30 03:57:40.714]

RX_ID: 50 Freq: 131.550MHz
Mode: 2 (50), Label: _d
Block ID: 3 (51), Msg No: S19A
Aircraft reg: B-6537, Flight ID: MU0779
Type: Airbus A330, cn: 1262, Carrier: CES
Airline: China Eastern

[2014-12-30 03:56:36.608]

RX_ID: 49 Freq: 131.450MHz
Mode: 2 (50), Label: _d
Block ID: 9 (57), Msg No: S97A
Aircraft reg: ZK-NCI, Flight ID: NZ7821
Vendor: Boeing, Type: B767-319 (ER) (B763), cn: 26913, Carrier: NZ, Remark: Air New Zealand
Airline: Air New Zealand

[2014-12-30 03:56:16.503]

RX_ID: 48 Freq: 131.450MHz
Mode: 2 (50), Label: C1 (Uplink message to the cockpit printer 1)
Block ID: 0 (79)
Aircraft reg: ZK-NCI
Vendor: Boeing, Type: B767-319 (ER) (B763), cn: 26913, Carrier: NZ, Re

ND: 020/15
VIS: 50KM
CLD: SCT035
TEMPERATURE: 23
DEW POINT: 10
QNH: 1019
SIGWX: CONDITIONS SUITABLE FOR A VISUAL
APPROACH
2000FT WIND: REPORTED 010/15

Administrator: C:\Windows\system32\cmd.exe - acarsdeco2.exe --gain 16.6 --freq-correction 71 --f...

```
Block id: 1 (49), msg. no: S99A
Aircraft reg: ZK-NCI, flight id: NZ7821
Vendor: Boeing, type: B767-319 (ER) (B763), cn: 26913, carrier: NZ, remarks: Air
 New Zealand
Airlines: Air New Zealand

Message content:-

------------------------------------------------------------[2014-12-30 04:07:56.4
12]

RX_ID: 78, Freq: 131.550MHz
ACARS mode: A (65), label: 11 (Inrange Arrival report)
Block id: 1 (49), msg. no: M97A
Aircraft reg: B-6537, flight id: MU0779
Type: Airbus A330, carrier: CES, cn: 1262
Airlines: China Eastern

Message content:-
POS
CAS 299,LAT S 36.219,LON E173.489,ALT 14155,FOB 25400,UTC 040000
------------------------------------------------------------[2014-12-30 04:08:05.4
58]
```

PLANEPLOTTER

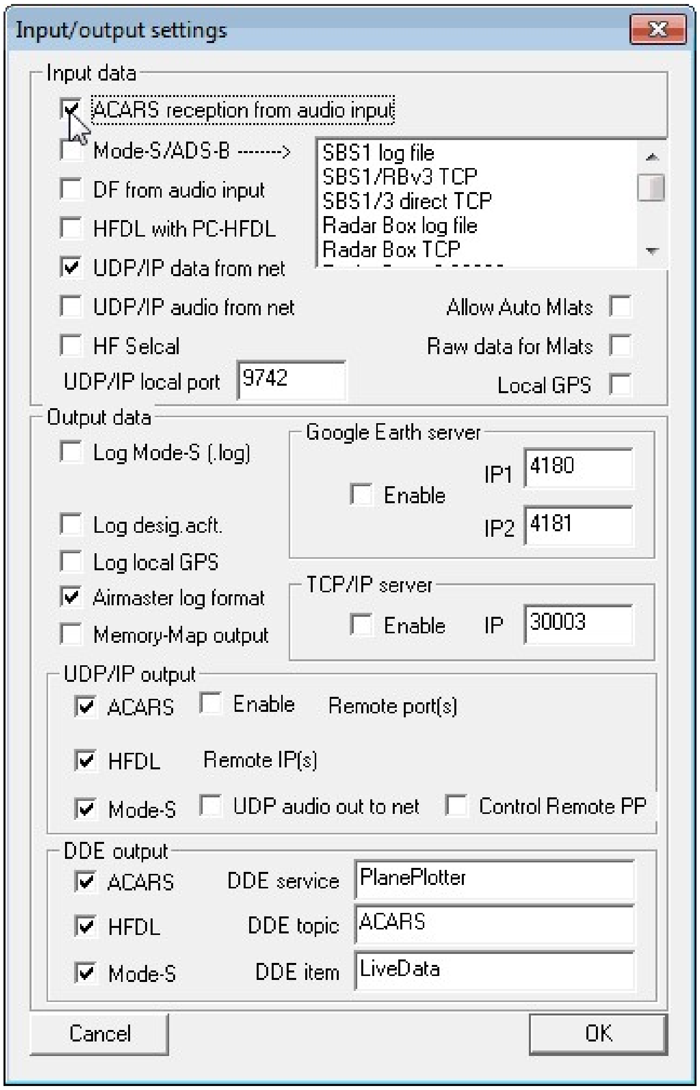
The first one is PlanePlotter, which is a paid software package that costs 25 Euros. However, there is also a free 21-day trial version available. PlanePlotter also has the advantage that it can be used for ADS-B plotting in a later project. To use PlanePlotter for ACARS, follow the instructions below.

1. Download and install PlanePlotter from http://www.coaa.co.uk/planeplotter.htm.

2. In your SDR receiver such as SDR#, set the audio piping method you want to use as the audio output.

3. Open PlanePlotter. Click on the “I/O Settings” button 🔧.

4. Ensure that the option “ACARS reception from audio input is checked”. Push OK.



Input/output settings

Input data

☑ ACARS reception from audio input
☐ Mode-S/ADS-B -------> SBS1 log file / SBS1/RBv3 TCP / SBS1/3 direct TCP / Radar Box log file / Radar Box TCP
☐ DF from audio input
☐ HFDL with PC-HFDL
☑ UDP/IP data from net
☐ UDP/IP audio from net — Allow Auto Mlats ☐
☐ HF Selcal — Raw data for Mlats ☐
UDP/IP local port 9742 — Local GPS ☐

Output data

☐ Log Mode-S (.log)
☐ Log desig.acft.
☐ Log local GPS
☑ Airmaster log format
☐ Memory-Map output

Google Earth server: ☐ Enable — IP1 4180 — IP2 4181

TCP/IP server: ☐ Enable — IP 30003

UDP/IP output: ☑ ACARS ☐ Enable Remote port(s) — ☑ HFDL Remote IP(s) — ☑ Mode-S ☐ UDP audio out to net ☐ Control Remote PP

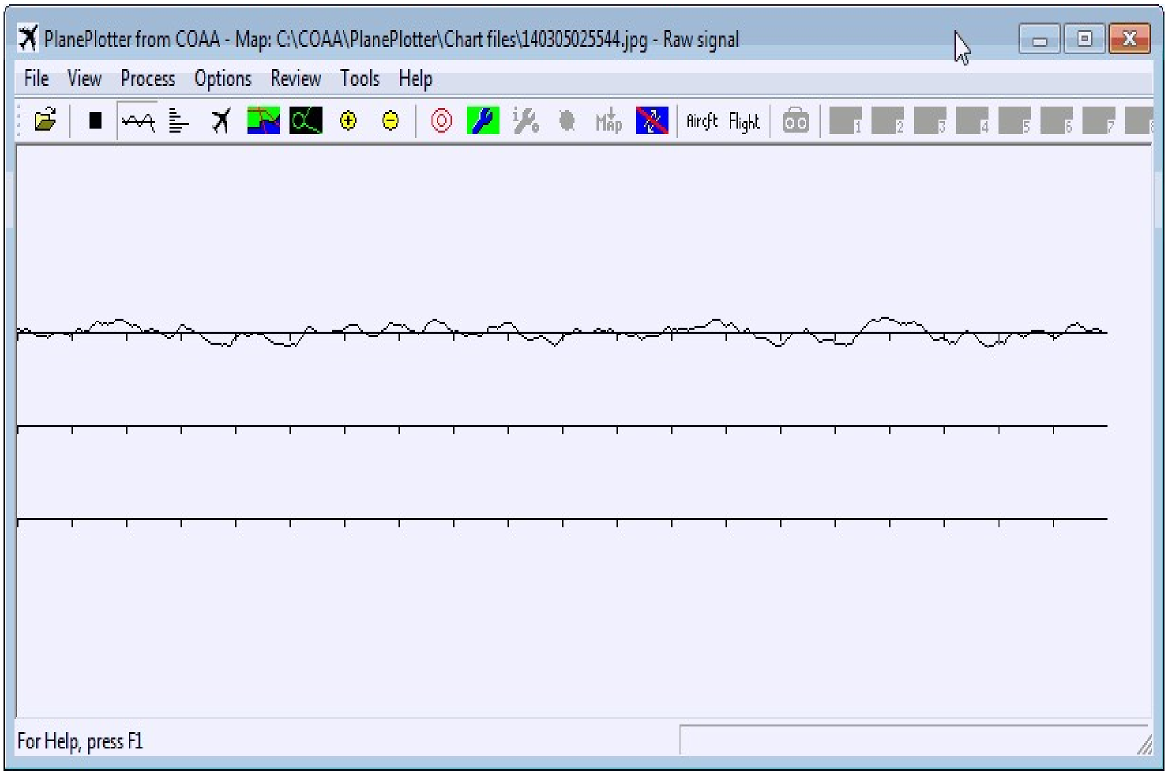
DDE output: ☑ ACARS — DDE service PlanePlotter — ☑ HFDL — DDE topic ACARS — ☑ Mode-S — DDE item LiveData

5. Go to **Options->Audio->ACARS** and select the audio pipe you are using in your SDR receiver.

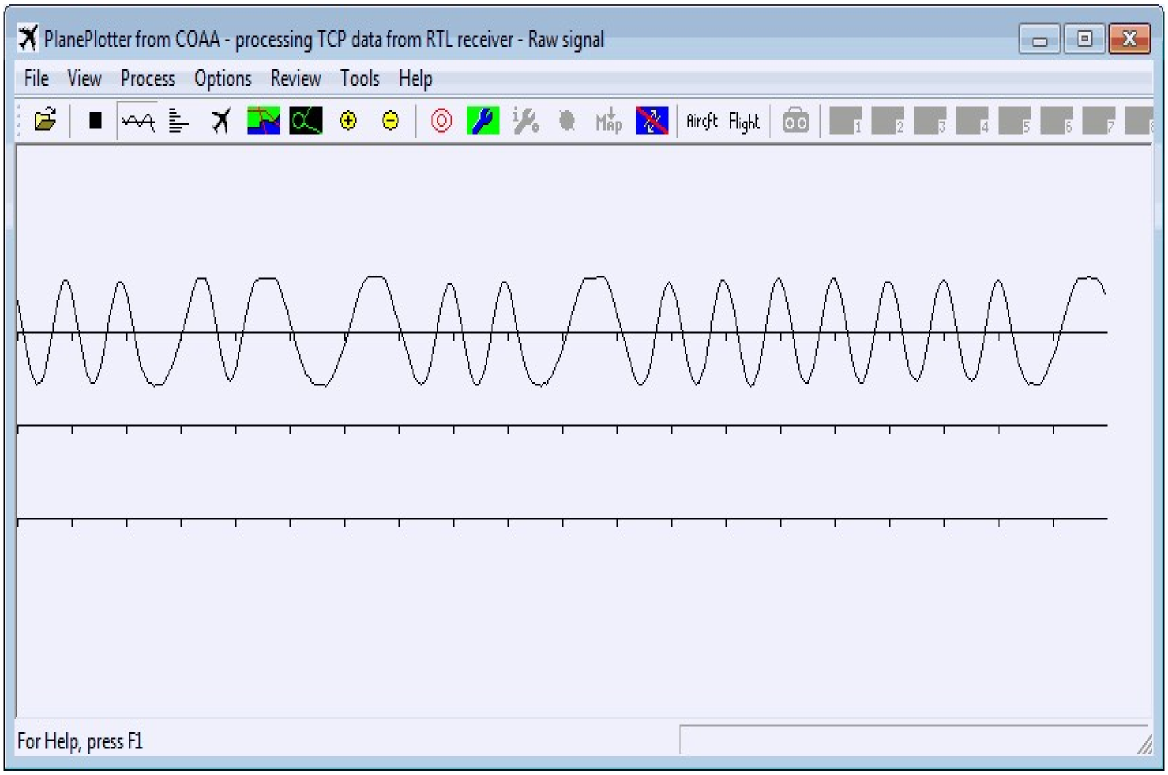
6. Click on the “Start” icon ●.

7. Press the “Signal” icon ∿. This shows a graph of the signal’s audio volume. Ensure the signal is not too loud. You will know the volume is too loud if the graph begins clipping/saturating when ACARS packets are received (signal will look squarish if it is saturating).

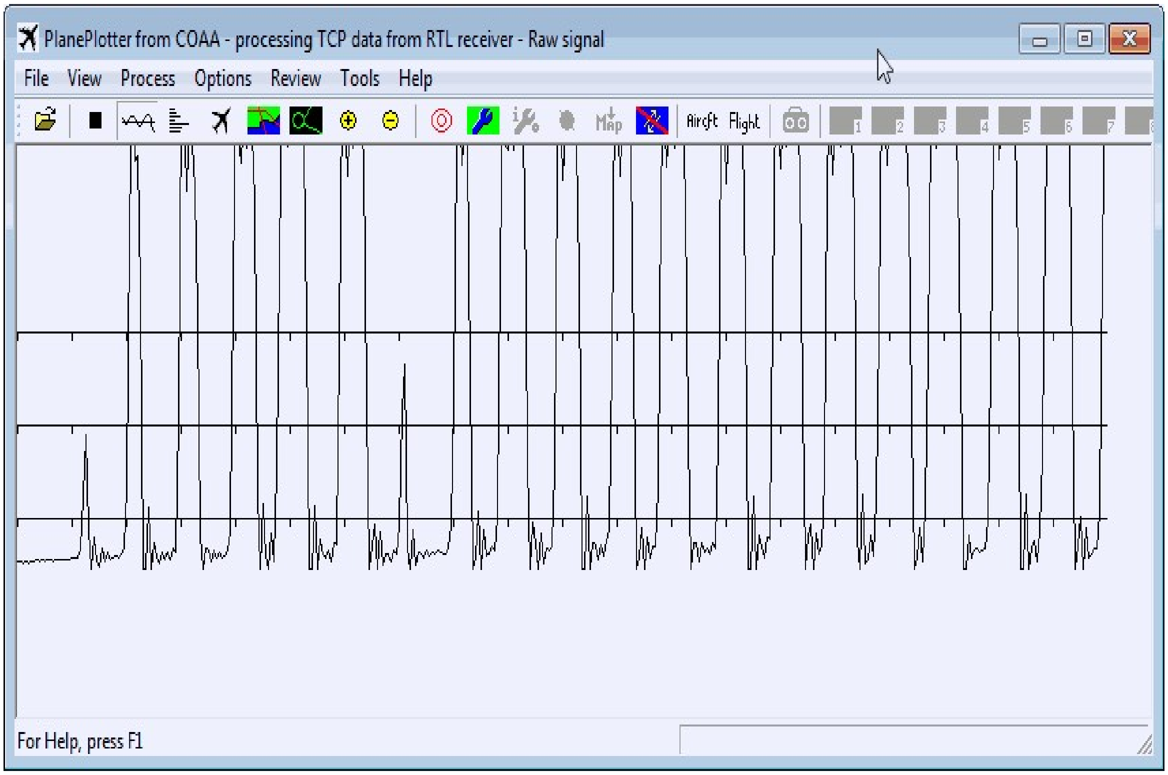
Below is an example of an ideal idle volume.


PlanePlotter from COAA - Map: C:\COAA\PlanePlotter\Chart files\140305025544.jpg - Raw signal
File View Process Options Review Tools Help
Map
Aircft Flight
1 2 3 4 5 6 7 8
For Help, press F1

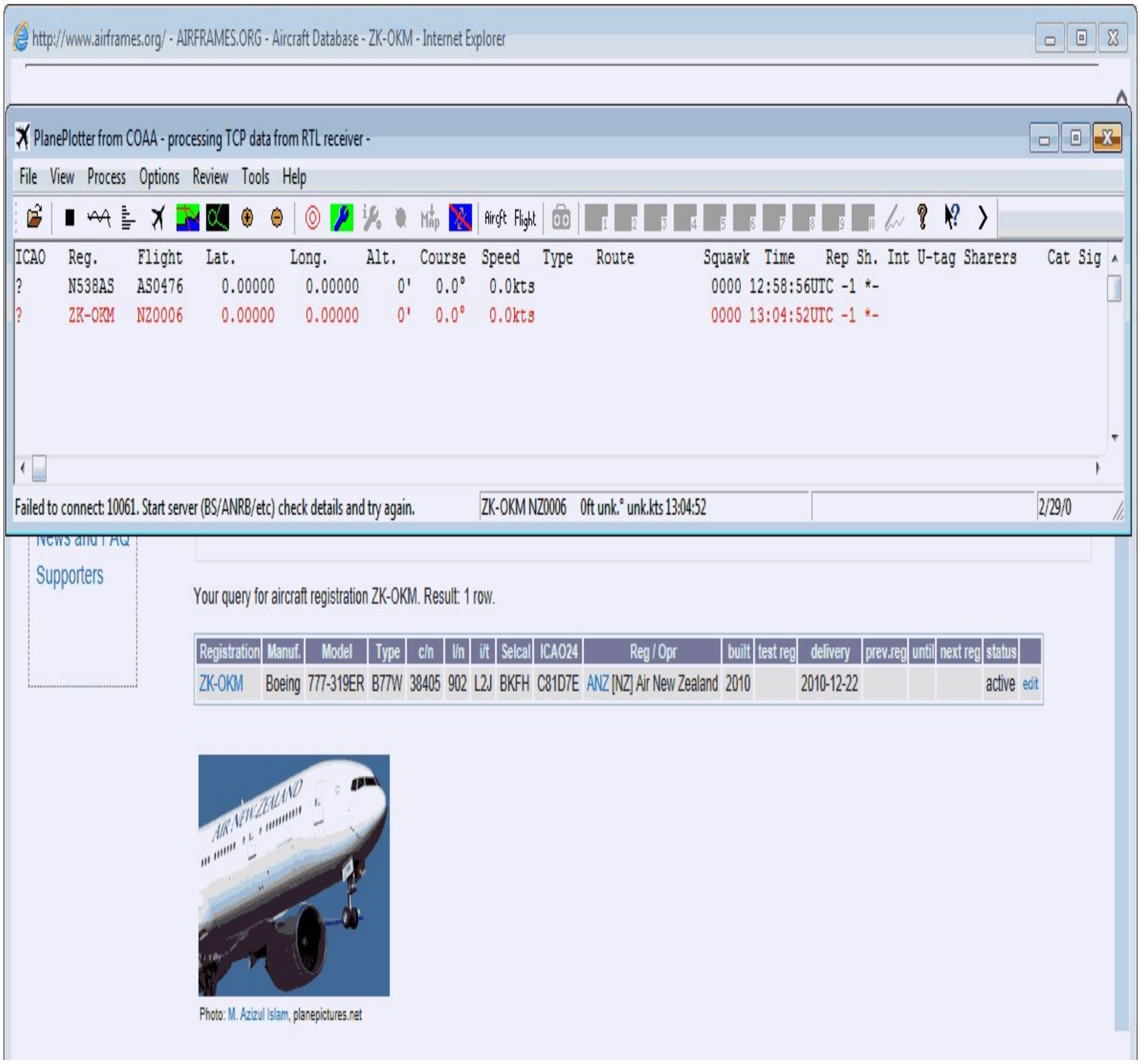
Below is an example of ideal volume when an ACARS packet is sounding.


PlanePlotter from COAA - processing TCP data from RTL receiver - Raw signal
File View Process Options Review Tools Help
Aircft Flight
1 2 3 4 5 6 7
For Help, press F1

Here is an example of an ACARS packet that has its volume set too loud.


PlanePlotter from COAA - processing TCP data from RTL receiver - Raw signal
File View Process Options Review Tools Help
Aircft Flight
1 2 3 4 5 6 7
For Help, press F1

At this point ACARS messages should be successfully decoding in PlanePlotter. Right clicking on an entry in the Aircraft View window ✈ will bring up an internet browser with information about that particular aircraft.



http://www.airframes.org/ - AIRFRAMES.ORG - Aircraft Database - ZK-OKM - Internet Explorer
PlanePlotter from COAA - processing TCP data from RTL receiver -
File View Process Options Review Tools Help
Aircft Flight
News and FAQ
Supporters

```
ICAO  Reg.    Flight  Lat.      Long.     Alt.  Course  Speed  Type  Route       Squawk  Time          Rep Sh. Int U-tag Sharers   Cat Sig
?     N538AS  AS0476    0.00000   0.00000   0'    0.0°   0.0kts                   0000  12:58:56UTC -1 *-
?     ZK-OKM  NZ0006    0.00000   0.00000   0'    0.0°   0.0kts                   0000  13:04:52UTC -1 *-
```

Failed to connect: 10061. Start server (BS/ANRB/etc) check details and try again. ZK-OKM NZ0006 0ft unk.° unk.kts 13:04:52 2/29/0

Your query for aircraft registration ZK-OKM. Result: 1 row.

Registration	Manuf.	Model	Type	c/n	l/n	i/t	Selcal	ICAO24	Reg / Opr	built	test reg	delivery	prev.reg	until	next reg	status	
ZK-OKM	Boeing	777-319ER	B77W	38405	902	L2J	BKFH	C81D7E	ANZ [NZ] Air New Zealand	2010		2010-12-22				active	edit

Photo: M. Azizul Islam, planepictures.net

AIRNAV ACARS

Another commercially available ACARS decoder is the one by AirNav, known as AirNav ACARS decoder 2. This product comes with a 30 day trial but costs $39.95 USD every six months, which makes this the priciest option available. It is available at http://www.airnavsystems.com/ACARS/. The advantage with this decoder is that it is very easy to set up and has good decoding capabilities on par with PlanePlotter. To use AirNav ACARS all that is required is to:

1. Download and install the AirNav ACARS Decoder 2 program.

2. Ensure your default audio device is set correctly in Windows sound recording properties. Acarsd will use the first default audio device found.

3. In your SDR receiver, set the audio piping method you want to use as the audio output.

4. Open the program. AirNav will select the default audio device.

5. Adjust the volume in your SDR receiver or Windows such that the volume meter in the bottom right corner is green and does not enter the red when an ACARS packet is heard.

ACARSD

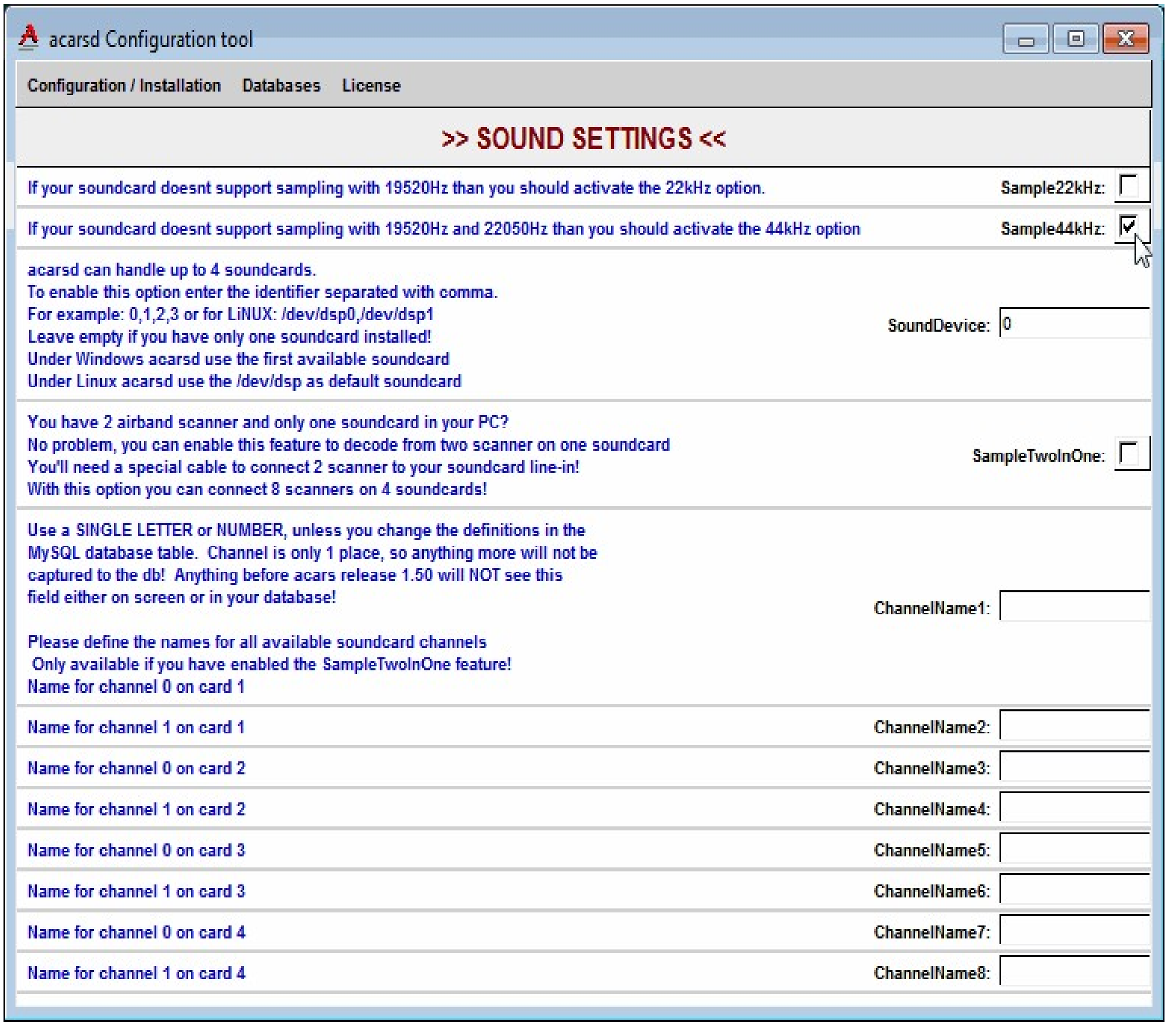
Another program for decoding ACARS signals is Acarsd. Acarsd has the advantage of being completely free and also being available for Linux. Acarsd can be downloaded from www.acarsd.com. We find that the performance of Acarsd is not as good as the two paid options, but some other users have reported that Acarsd can work just as well if not better than the paid options. Acarsd works best when the audio sample rate of the audio piping device is set to 44.1 kHz. See the setting the sample rate section for information on how to change this. To use acarsd follow these instructions.

1. Go to the acarsd website and download and install acarsd. Make sure to follow the instructions on the website for installing as it is an unconventional install procedure.

2. Ensure your default audio device is set correctly in Windows sound recording properties. Acarsd will use the first default audio device found. Also ensure the Playback and Recording sample rates are set to the same value of 44100 Hz.

3. In the acarsd folder run setup.exe.

4. Under Configuration / Installation enable the 44 kHz option.



acarsd Configuration tool

>> SOUND SETTINGS <<

If your soundcard doesnt support sampling with 19520Hz than you should activate the 22kHz option. Sample22kHz: □

If your soundcard doesnt support sampling with 19520Hz and 22050Hz than you should activate the 44kHz option Sample44kHz: ☑

acarsd can handle up to 4 soundcards.
To enable this option enter the identifier separated with comma.
For example: 0,1,2,3 or for LiNUX: /dev/dsp0,/dev/dsp1
Leave empty if you have only one soundcard installed!
Under Windows acarsd use the first available soundcard
Under Linux acarsd use the /dev/dsp as default soundcard

SoundDevice: 0

You have 2 airband scanner and only one soundcard in your PC?
No problem, you can enable this feature to decode from two scanner on one soundcard
You'll need a special cable to connect 2 scanner to your soundcard line-in!
With this option you can connect 8 scanners on 4 soundcards!

SampleTwoInOne: □

Use a SINGLE LETTER or NUMBER, unless you change the definitions in the MySQL database table. Channel is only 1 place, so anything more will not be captured to the db! Anything before acars release 1.50 will NOT see this field either on screen or in your database!

Please define the names for all available soundcard channels
Only available if you have enabled the SampleTwoInOne feature!
Name for channel 0 on card 1 ChannelName1:

Name for channel 1 on card 1 ChannelName2:

Name for channel 0 on card 2 ChannelName3:

Name for channel 1 on card 2 ChannelName4:

Name for channel 0 on card 3 ChannelName5:

Name for channel 1 on card 3 ChannelName6:

Name for channel 0 on card 4 ChannelName7:

Name for channel 1 on card 4 ChannelName8:

5. Go to **Configuration / Installation->Write new acarsd.ini file**. Then exit the configuration tool.

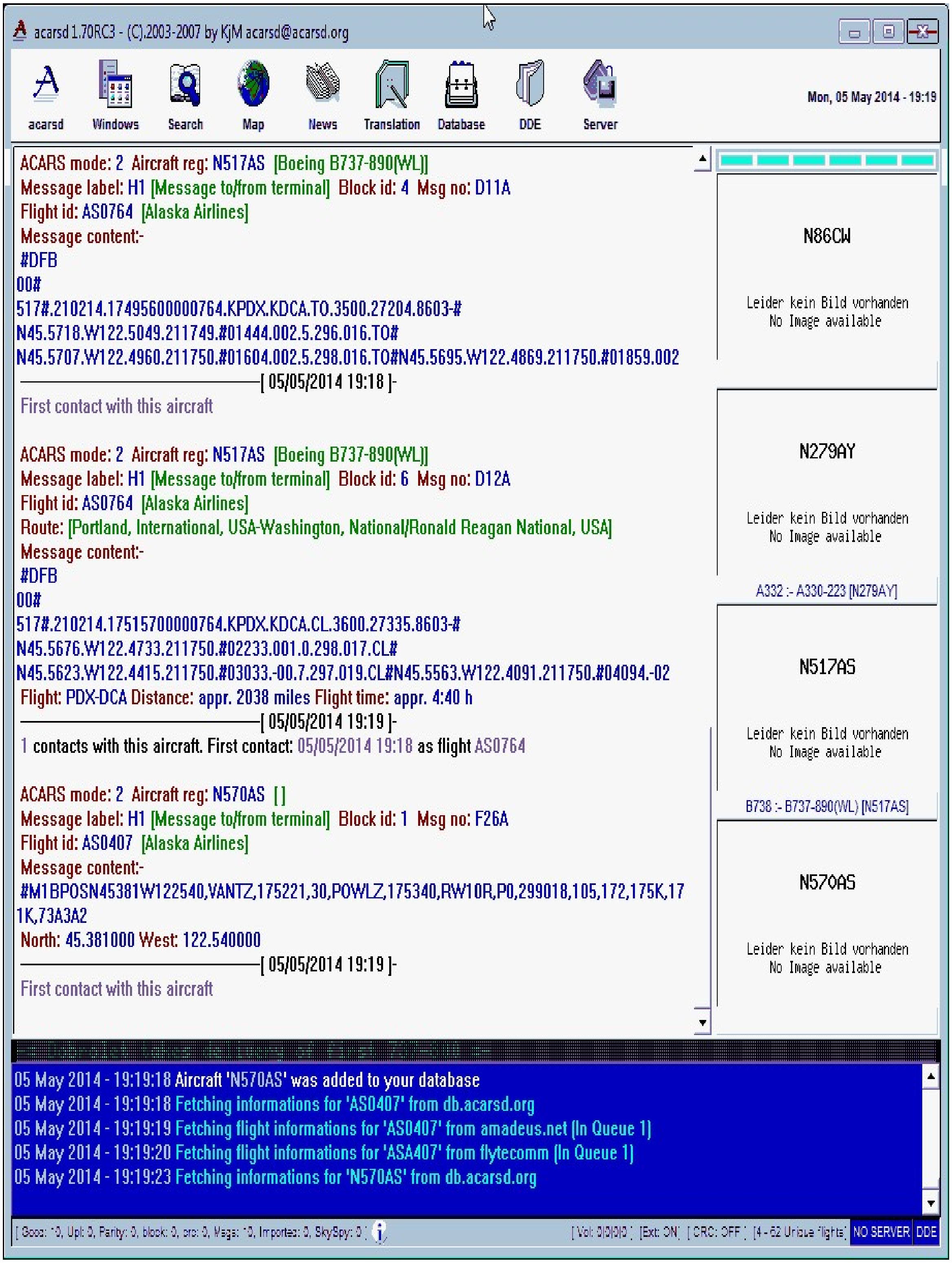
6. In your SDR receiver, set it to output to the audio piping method you have set as your default audio device.

7. Run acarsd.exe in your acarsd folder to open the program.

8. Adjust the volume in your SDR receiver or in Windows so that the volume reading in the bottom right corner of the acarsd window is at around 0-10 when there is no ACARS signal being broadcast. Acarsd is very picky with the volume levels being too loud so make sure you set this correctly.

9. Decoded ACARS packets should begin to show up.

The number of decoder passes can be changed in **Windows->acarsdgui settings**. Setting it higher may improve decode performance.


acarsd 1.70RC3 - (C).2003-2007 by KjM acarsd@acarsd.org
acarsd
Windows
Search
Map
News
Translation
Database
DDE
Server
Mon, 05 May 2014 - 19:19
N86CW
Leider kein Bild vorhanden
No Image available
N279AY
Leider kein Bild vorhanden
No Image available
A332 :- A330-223 [N279AY]
N517AS
Leider kein Bild vorhanden
No Image available
B738 :- B737-890(WL) [N517AS]
N570AS
Leider kein Bild vorhanden
No Image available
NO SERVER
DDE

```
ACARS mode: 2  Aircraft reg: N517AS  [Boeing B737-890(WL)]
Message label: H1 [Message to/from terminal]  Block id: 4  Msg no: D11A
Flight id: AS0764  [Alaska Airlines]
Message content:-
#DFB
00#
517#.210214.17495600000764.KPDX.KDCA.TO.3500.27204.8603-#
N45.5718.W122.5049.211749.#01444.002.5.296.016.TO#
N45.5707.W122.4960.211750.#01604.002.5.298.016.TO#N45.5695.W122.4869.211750.#01859.002
-----------------------------[ 05/05/2014 19:18 ]-
First contact with this aircraft

ACARS mode: 2  Aircraft reg: N517AS  [Boeing B737-890(WL)]
Message label: H1 [Message to/from terminal]  Block id: 6  Msg no: D12A
Flight id: AS0764  [Alaska Airlines]
Route: [Portland, International, USA-Washington, National/Ronald Reagan National, USA]
Message content:-
#DFB
00#
517#.210214.17515700000764.KPDX.KDCA.CL.3600.27335.8603-#
N45.5676.W122.4733.211750.#02233.001.0.298.017.CL#
N45.5623.W122.4415.211750.#03033.-00.7.297.019.CL#N45.5563.W122.4091.211750.#04094.-02
Flight: PDX-DCA Distance: appr. 2038 miles Flight time: appr. 4:40 h
-----------------------------[ 05/05/2014 19:19 ]-
1 contacts with this aircraft. First contact: 05/05/2014 19:18 as flight AS0764

ACARS mode: 2  Aircraft reg: N570AS  [ ]
Message label: H1 [Message to/from terminal]  Block id: 1  Msg no: F26A
Flight id: AS0407  [Alaska Airlines]
Message content:-
#M1BPOSN45381W122540,VANTZ,175221,30,POWLZ,175340,RW10R,P0,299018,105,172,175K,17
1K,73A3A2
North: 45.381000 West: 122.540000
-----------------------------[ 05/05/2014 19:19 ]-
First contact with this aircraft
```

```
05 May 2014 - 19:19:18 Aircraft 'N570AS' was added to your database
05 May 2014 - 19:19:18 Fetching informations for 'AS0407' from db.acarsd.org
05 May 2014 - 19:19:19 Fetching flight informations for 'AS0407' from amadeus.net (In Queue 1)
05 May 2014 - 19:19:20 Fetching flight informations for 'ASA407' from flytecomm (In Queue 1)
05 May 2014 - 19:19:23 Fetching informations for 'N570AS' from db.acarsd.org
```

```
[ Good: 10, Upl: 0, Parity: 0, block: 0, crc: 0, Msgs: 10, Imported: 0, SkySpy: 0 ]    [ Vol: 0|0|0|0 ]  [Ext: ON]  [ CRC: OFF ]  [4 - 62 Unique flights]
```

LINUX ONLY ACARS DECODERS

There are various free open source terminal based ACARS decoders available for Linux. The two best are rtl_acars_ng and acarsed. Both have a significant advantage over the Windows based ACARS solutions as these programs will connect directly to the RTL-SDR and so do not require any audio piping. Furthermore, they will both also listen to and simulataenously decode more than one ACARS channel. Acarsdec is the more feature rich software, but both programs should have similar performance as they are based on the same code base.

These programs are great for running on an embedded device such as a Raspberry Pi.

ACARSDEC

To run acarsdec, download and extract tar package from http://sourceforge.net/projects/acarsdec/ or use the following at the Linux terminal.

```
wget http://sourceforge.net/projects/acarsdec/files/latest/download?source=files -O acarsed.tar.gz
tar -zxvf acarsed.tar.gz
```

Install the following dependencies as well as the librtlsdr libraries if you haven't already and compile the program using `make`. See the installing RTL-SDR Drivers on Linux section for more information.

```
sudo apt-get install libsndfile1-dev
sudo apt-get install libasound2
sudo apt-get install sqlite3 libsqlite3-dev
cd acarsdec-3.0
make
```

To run the program on an RTL-SDR dongle monitoring two frequencies use the following command, where -p specifies the PPM offset for your dongle, -r specifies the RTL-SDR device number if you have more than one RTL-SDR connected and the values after specify ACARS frequencies you wish to monitor. Note that since ACARS signals are narrowband, it is very important the get the PPM offset set correctly. See `./acarsdec -h` for a full list of options.

```
./acarsdec -p 24 -r 0 131.550 131.450
```

You can also send the output of acarsdec through a network connection to a PC running PlanePlotter. To do this, first find out the IP address of the PC running PlanePlotter. Then on the Linux device use the following where you should replace IP_ADDR with the ip address of the computer running PlanePlotter.

```
./acarsdec -n IP_ADDR:9742 -r 0 131.550 131.350
```

Next in Windows and in PlanePlotter ensure that under **Options->I/O Settings** you have the “UDP/IP data from net” checkbox checked, then press the green start button. As ACARS packets are received on the Linux computer running acarsdec, they will also show in the PlanePlotter messages window. Using this you could have a remote embedded PC such as a Raspberry Pi running acarsdec feeding PlanePlotter from a distance.

RTL_ACARS_NG

Rtl_acars_ng is a much simpler ACARS command line decoder. To run rtl_acars_ng simply download the zip file from https://github.com/gat3way/rtl_acars_ng, extract it then run `make` as shown below. To get help run `./rtl_acars_ng -h`.

```
sudo apt-get install libusb-dev
git clone https://github.com/gat3way/rtl_acars_ng
cd rtl_acars_ng
make
```

Then you can run the program using the following.

```
./new_rtl_acars -f 131.550M
```

ADS-B RECEIVING GUIDE (TRACKING AIRCRAFT)

ADS-B INTRODUCTION

Modern planes carry on board something called an Automatic Dependent Surveillance-Broadcast (ADS-B) Mode-S transponder. This transponder periodically broadcasts location and altitude information to air traffic controllers and other aircraft. The system was put into place to replace traditional RADAR systems which work by bouncing a radio signal off the plane and listening for the echo. Traditional RADAR is not always reliable and so ADS-B was invented. ADS-B uses an accurate onboard GPS receiver and broadcasts the GPS location data to ground controllers and other aircraft via radio giving more accurate and stable data. ADS-B is also used to help aircraft avoid collisions by sending a warning or taking automatic emergency action if the flight computer detects a possible collision from ADS-B data.

Currently, ADS-B data is completely unencrypted so the RTL-SDR can be used to listen to these ADS-B signals, which can then be used to create your very own home aircraft radar system. Compared to dedicated commercial ADS-B receivers which can go for between $200 and $1000, the $20 RTL-SDR is very attractive for the hobbyist in terms of price. ADS-B receive performance with the RTL-SDR is also still quite good. However, note that the RTL-SDR probably shouldn’t be relied on for ADS-B navigation in a real aircraft for safety reasons. Also, if you are using a transmit capable SDR, please do not ever transmit on the ADS-B frequency as this could have serious repercussions. ADS-B signals can be found at a frequency of 1090 MHz.

ADS-B is based on something called Mode S which provides the location data for ADS-B. There is also Mode A which provides an identification code and Mode C which provides the aircraft's pressure altitude. These other modes are also receivable by some decoders.

In the USA some small aircraft prefer to use an alternative protocol to ADS-B called Universal Access Transceiver (UAT) which transmits at 978 MHz. UAT has some extra features for pilots compared to ADS-B. In addition to location information UAT provides a Traffic Information Service (TIS/B) which allows pilots to see what ground control sees on their traditional RADAR system. It also provides a Flight Information Service-Broadcast (FIS/B) which includes weather and other information. It seems that most small aircraft in the USA prefer to use the UAT system due to it's lower cost and additional features. Currently UAT decoding is not as well supported as ADS-B decoding, however there are now two decoders available which will be discussed below.

It has been discovered by the RTL-SDR community, that the R820T/R820T2 tuner has the best sensitivity out of all the tuners at this frequency. The E4000 and other tuners do not perform as well in comparison. It is recommended that you obtain an R820T or R820T2 tuner if you want to set up ADS-B decoding with the RTL-SDR.

There are public websites like flightradar24.com which display ADS-B data through the internet as well. They receive ADS-B data through volunteers who have bought dedicated ADS-B receiver hardware, or are using an RTL-SDR and special software.

ADS-B TUTORIAL

For this tutorial we will assume that you have an RTL-SDR dongle installed and working on a Windows PC. If not, please follow the Quickstart Guide shown in a previous section. Aside from this you will need the following:

- A vertically polarized antenna tuned to 1090 MHz.
- Software for receiving and decoding ADS-B.
- Software for displaying ADS-B location data.

ADS-B RECEPTION GUIDE

With a good antenna the RTL-SDR could be capable of receiving aircraft ADS-B signals from over 250 nm away, that is about 460 km or 290 miles away.

Many people have found that in most cases the stock antenna that comes with the dongle is already fairly good at picking up ADS-B signals. Its performance will depend on how far away you are from the aircraft and what your local RF interference is like. To improve the stock antenna you can cut the whip down to 6.9 cm and put it on top of a metallic ground plane. However, if you really want to increase your ADS-B receiving range, you will need a properly tuned antenna placed up high. There are of course commercial ADS-B antennas you can buy, but the best option for keeping in the cheap spirit of RTL-SDR is to build your own.

ADS-B operates with vertical polarization at 1090 MHz which is a line of sight frequency meaning that you need to have a clear unobstructed path between the antenna and the transmitter on the aircraft. Since ADS-B signals are generally quite strong and come from aircraft flying up high, reception is quite easy and most vertically polarized antennas tuned for 1090MHz will work well. Using a poorly tuned or poorly made antenna will result in a lack of range for your air radar.

For ADS-B we recommend either a quarter wave ground plane, j-pole, vertical dipole or collinear antenna. A discone may also work somewhat well, but a dedicated tuned antenna is best. See the Antenna Guide for more information about these antennas.

You should also ensure that your coax feed line (the length of coaxial cable between the antenna and dongle) is high quality and as short as possible. At gigahertz frequencies long runs of cheap coax tend to reduce signal strength significantly. Use coax cable intended for satellite TV installations such as RG-6 or RG-8 as these cables are designed to work well at gigahertz frequencies. See the Coaxial Cable Guide section for more information.

Also ensure you are using connectors on your coax cable that can handle gigahertz frequencies. See the Antenna Adapter Guide in the antennas section for information about connectors. We recommend BNC, N or SMA connectors. Poor connectors at gigahertz frequencies could net up to 1-3 dB of loss per connector.

A low noise amplifier (LNA) such as the LNA4ALL placed at the antenna may help with signal loss particularly if you have a long run of feed line. See the LNA section for more information. Using an active USB extension cable and placing the dongle close to the antenna may be a better option to consider. Some people have even placed a mini embedded computer like a Raspberry Pi running an RTL-SDR at the antenna and then used WiFi or an Ethernet cable to send decoded ADS-B data to a PC.

If you have strong broadcast stations near by, or have other sources of interference, we recommend using a preselector bandpass filter for 1090 MHz. A low cost ADS-B bandpass filter can also be bought from 9A4QV here http://www.adsbfilter.blogspot.com. Another good commonly used and easy to build filter is a hairpin filter on a PCB board http://www.rtl-sdr.com/homemade-ads-b-filter/. It is also easy to use a 1090 MHz SAW filter component which can be cheaply purchased from Ebay for around $10. http://www.ebay.com/itm/B1602-SAW-filters-Epcos-1090MHz-ADS-B-AirNav-SBS-/110619017914. These components are small and require decent soldering skills. See https://steve-m.de/pics/ads-b_saw_filter.jpg for an example of one wired up. For a more complete solution we reccomend the HABAMP which comes with a 1090 MHz SAW filter and an LNA built into a nice metal box. The HABAmp is available at http://ava.upuaut.net/store/index.php?route=product/product&product_id=85.

SAW bandpass filters can have high insertion losses. In many cases a simple high pass filter is all that is needed as these have significantly lower insertion losses and also cost less. A high pass filter that passes from about 1 GHz should be sufficient in most situations, and will block out strong broadcast FM and pager interference, however it will not block out strong cellular signals, which may or may not be a problem in your area. Something like a SHP-1000 filter from minicircuits should work well. See the preselector section for more information.

In this video https://www.youtube.com/watch?v=wcr-PX662Go, the uploader noticed an increase from 800 frames/s to 1200 frames/s when a preselector for 1090 MHz was used.

A high pass filter which blocks out frequencies below 1 GHz may also be a suitable alternative to a bandpass filter. Most interference will be from frequencies below 1 GHz anyway.

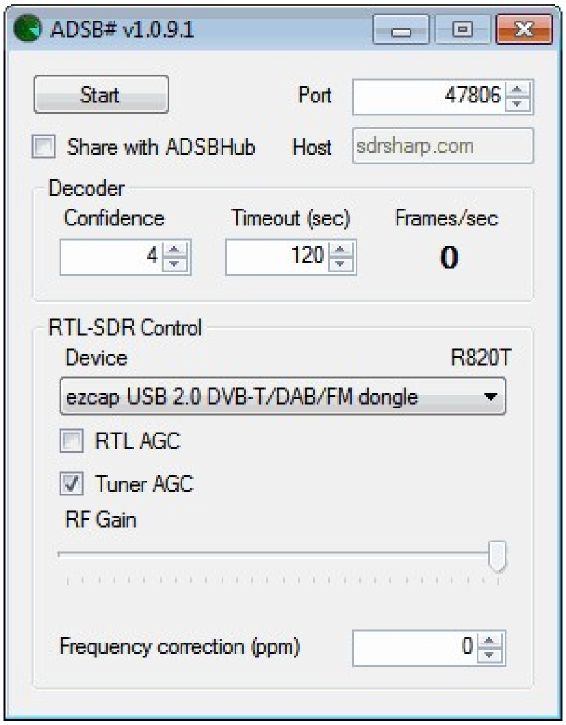
SOFTWARE GUIDE

Currently, there are multiple software options available for listening and decoding ADS-B signals with the RTL-SDR. Almost all software is free. Note for the Windows based software you will need to have installed the RTL-SDR drivers via zadig first. See the Quickstart guide if you have not done this.

The most commonly used ADS-B decoders are ASDB#, RTL1090, dump1090 and modesdeco2. For best performance we currently recommend dump1090, but for best ease of use we recommend ADSB#. For the most complete all in one package we recommend using modesdeco2.

ADSB# (WINDOWS)

ADSB# is a Windows based 1090 MHz ADS-B decoder developed by the programmer of SDR#. It has a simple graphic user interface and is probably the easiest decoder to use overall. To use ADSB#, download it from the main page at www.sdrsharp.com. ADSBSharp.exe can be found in the same folder as SDR#.



Port 47806

□ Share with ADSBHub

Host sdrsharp.com

Confidence 4

Timeout (sec) 120

Frames/sec 0

Device R820T

ezcap USB 2.0 DVB-T/DAB/FM dongle

□ RTL AGC

☑ Tuner AGC

Frequency correction (ppm) 0

To run ADSB# simply double click on ADSBSharp.exe. If your RTL-SDR dongle is plugged into your PC, you should be able to select it under the “Device” drop down menu. You can then click the start button to begin decoding ADS-B data. Be sure to also set the gain. Tuner AGC works well for a quick test and RTL AGC should almost never be used. Manual gain settings most often work the best, and setting it to maximum gain with no AGC options selected is usually the best choice.

If you have a good antenna set up, you should begin to see numbers other than zero showing up under Frames/sec. This indicates that you are receiving ADS-B data.

ADSB# decodes the ADS-B data and then sends location data via TCP/IP. Another program such as Virtual Radar Server, ADSBScope or PlanePlotter (explained later) is needed to visualize the data on a map. If you have several antennas in different locations, you can combine your data in this way using the adsbhub.exe program which is part of the ADSB# zip file download.

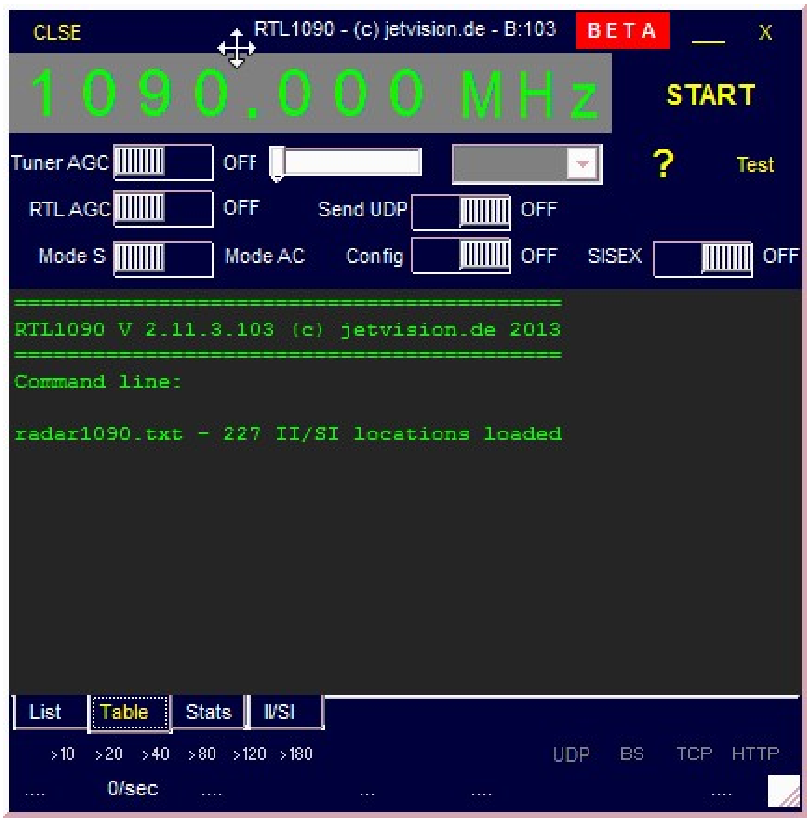
The frequency correction (ppm) setting is the same as the setting in SDR# which is used to correct for the RTL-SDR dongles frequency inaccuracy. Since ADS-B signals are not particularly sensitive to frequency accuracy, this setting won’t make much difference even if you set it correctly.

In ADSB#, when ADS-B data is received from an aircraft a confidence counter is incremented. The confidence counter rises as more packets from the same aircraft are received. Once the confidence is greater than the confidence value set in the GUI, the aircraft location data will be sent over TCP/IP. Confidence is used to prevent false aircraft locations showing up from noise which was accidentally decoded as a valid ADS-B signal. Higher confidence levels may be necessary in noisy environments, but in most situations a confidence of four will work well. The timeout setting defines how long the decoder will wait before checking the confidence of the aircraft again, the default of 120s seems to work fine.

RTL1090 (WINDOWS)

Another Windows 1090 MHz ADS-B decoder with graphical user interface is RTL1090 which can be downloaded from http://rtl1090.web99.de. The easiest way to download and install RTL1090 is to use the IMU (Installer and Maintenance Utility), which will automatically download all the required files. The IMU saves the work of having to copy over several dll files and even guides you through installing the drivers with zadig. Note that if you've already installed the drivers through zadig via the Quickstart guide you can ignore the zadig installation in the IMU. Currently, there is a new version of RTL1090 in beta that can also be downloaded from their website. In this tutorial we use the beta version, but the standard version could also be used.

After downloading RTL1090 it can be opened by clicking on rtl1090.exe in its folder, or if you installed it in the start menu. Upon opening you will be greeted with a screen like the one below. To start decoding press the start button.



```
==========================================
RTL1090 V 2.11.3.103 (c) jetvision.de 2013
==========================================
Command line:

radar1090.txt - 227 II/SI locations loaded
```

To access the options screen click on the OPEN button in the top left corner of the RTL1090 Window. RTL1090 has some special features. It can decode MODE A/C as well as MODE S. Mode A sends an aircraft identification number and Mode C sends the aircraft altitude, MODE S is the most useful one as it has the location information. Mode AC decoding can be activated by flipping the switch in the options screen over to the Mode AC side. RTL1090 also has a built in spectrum viewer which can be activated by flipping the SISEX switch, this is useful for verifying that ADS-B signals are being received. Finally, RTL1090 also has a 1-bit checksum error checker which can improve decoding. The error checker can be activated in the Config menu which can be opened by flipping the Config switch in the options screen. Error checking is disabled by default because it can be taxing on slower PCs. The recent beta 3 version of RTL1090 also comes with a built in graphical radar display.

As with ADSB#, we recommend playing with the manual gain settings or running with Tuner AGC checked and RTL AGC unchecked. Maximum gain with no AGC enabled is usually the best choice.

The authors of RTL1090 also have an experimental version of their software called RTL-DUO, which utilizes two RTL-SDR dongles for improved reception. It can also be downloaded from http://rtl1090.web99.de.

DUMP1090 (LINUX/MAC/WINDOWS)

This is a Linux/OSX/Windows compatible command line ADS-B decoder. It is very useful for Linux based embedded devices such as the Raspberry Pi. With dump1090 you could set up a Raspberry Pi decoder that feeds data to a web server. Most people report that dump1090 has the best decoding performance out of all the ADS-B decoders.

The latest and most up to date branch of dump1090 is maintained by Malcolm Robb and can be downloaded from its git repository at https://github.com/MalcolmRobb/dump1090.

Dump1090 also has a hub mode, which will allow multiple dump1090 instances in different physical locations to connect together and combine data streams. This might be an option if you are not interested in sharing your data publically such as with adsbhub or flightradar24.

Dump1090 also has a graphical display web interface that uses Google maps built into the code. It can be accessed through a web browser.

To install dump1090 on Linux, use the following commands assuming you have already installed the librtlsdr library.

```
git clone https://github.com/MalcolmRobb/dump1090
cd dump1090
make
```

Dump1090 can also be compiled on OSX, but you will need to ensure that you have a compiler and make-tools installed with access to the dev environment and Xcode. Once compiled you can run the following.

```
./dump1090 --interactive --net
```

This will start a screen that shows all received aircraft and also a Google maps web server that can visually display received aircraft. To see the map go to http://localhost:8080 in a web browser, or if you’d like to access it from another PC use http://IP_ADDR:8080, where IP_ADDR is the ip address of the computer running dump1090.

You may wish to also enable the `--aggressive` and `--fix` flags which will be able to decode more ADS-B packets at the expense of increased CPU usage. The aggressive mode enables aggressive error correction and is mainly useful for low air traffic areas as it may cause many false positives under high traffic conditions. The fix option enables 1-bit error correction. More options can be found by typing `./dump1090 -h` at the terminal. Note that dump1090 automatically chooses the maximum gain setting if no gain is set, which is usually the best choice.

Dump1090 can also receive Mode AC messages together with Mode S by using the flag --modeac.

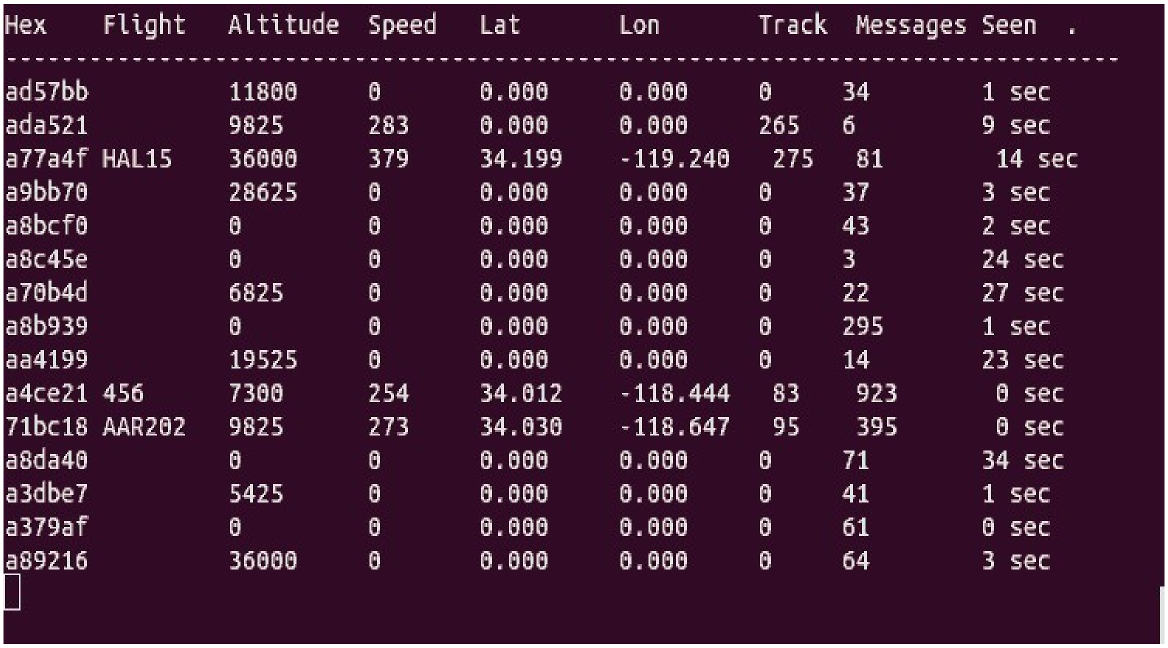
To use dump1090 with a program like Virtual Radar Server (described below) you can set it to output data in Beast format on port 30002 as follows.

```
dump1090.exe --interactive --net --net-ro-port 30002 --net-beast --mlat
```

To install dump1090 on Windows follow these steps:

1. Download the dump1090 zip file from the Github download link at https://github.com/MalcolmRobb/dump1090.
2. Download the official RTL-SDR Windows release from http://sdr.osmocom.org/trac/attachment/wiki/rtl-sdr/RelWithDebInfo.zip.
3. Copy the libusb-1.0.dll, rtlsdr.dll and pthreadVC2-w32.dll files from the official RTL-SDR Windows release zip file to the dump1090 folder.
4. Rename pthreadVC2-w32.dll to pthreadVC2.dll.
5. Double click on dump1090.bat.

The dump1090 Windows batch file contains the line dump1090.exe --interactive --net --net-ro-port 30002 --net-beast --mlat which starts an interactive web server and beast mode output which will allow programs like Virtual Radar Server (described below) to connect to it using port 30002. You may also wish to experiment with adding the --aggressive and/or --fix flags to the batch file which can be edited with a text editor like Notepad.



```
Hex    Flight   Altitude  Speed   Lat       Lon       Track  Messages Seen  .
--------------------------------------------------------------------------------
ad57bb          11800     0       0.000     0.000     0      34         1 sec
ada521          9825      283     0.000     0.000     265    6          9 sec
a77a4f HAL15    36000     379     34.199    -119.240   275    81         14 sec
a9bb70          28625     0       0.000     0.000     0      37         3 sec
a8bcf0          0         0       0.000     0.000     0      43         2 sec
a8c45e          0         0       0.000     0.000     0      3          24 sec
a70b4d          6825      0       0.000     0.000     0      22         27 sec
a8b939          0         0       0.000     0.000     0      295        1 sec
aa4199          19525     0       0.000     0.000     0      14         23 sec
a4ce21 456      7300      254     34.012    -118.444   83     923        0 sec
71bc18 AAR202   9825      273     34.030    -118.647   95     395        0 sec
a8da40          0         0       0.000     0.000     0      71         34 sec
a3dbe7          5425      0       0.000     0.000     0      41         1 sec
a379af          0         0       0.000     0.000     0      61         0 sec
a89216          36000     0       0.000     0.000     0      64         3 sec
```

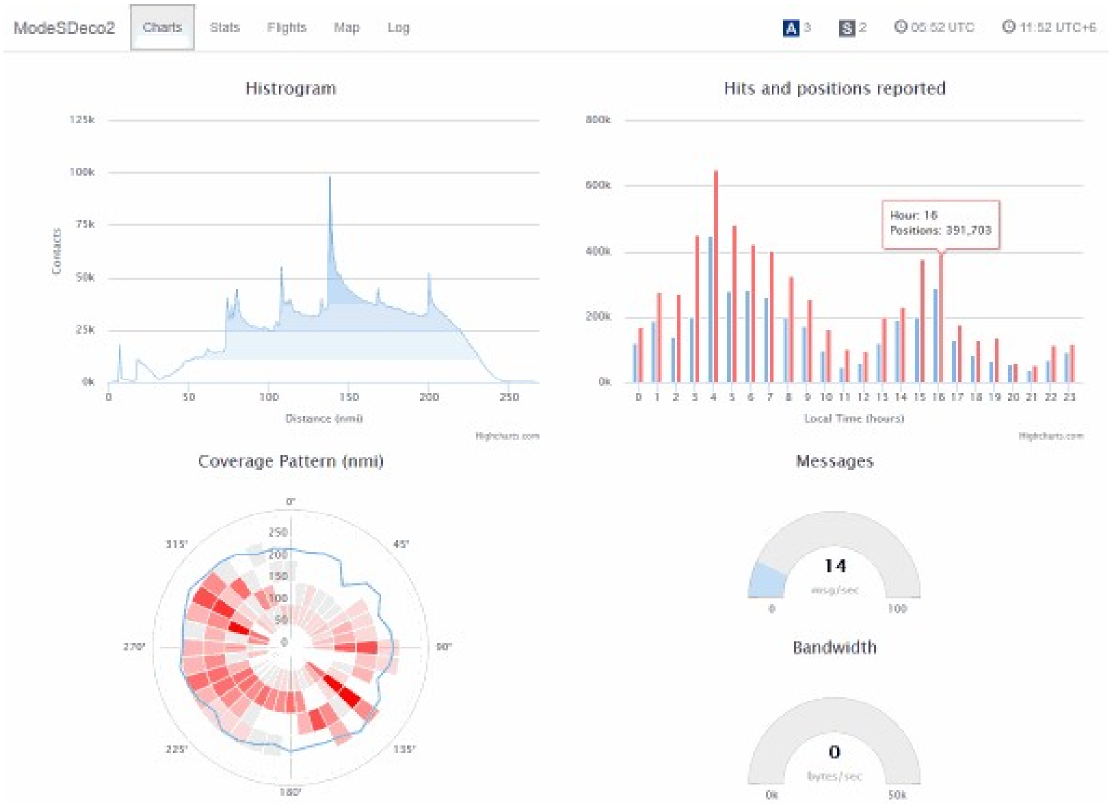
MODESDECO2

ModesDeco is a Windows/Linux/OSX command line based ADS-B decoder. It can receive Mode S and Mode A/C data simultaneously and can interface with Basestation and other graphical display software directly. It comes with a very nice web base GUI that can display maps of flights, and ADS-B stats as well. Modesdeco2 doesn't have an official download or web page, but you can download the latest version from the last page on the forum post at http://radarspotting.com/forum/index.php/topic,2978.0.html. Note that you may need to install the Visual C++ 2012 Redistributable from http://www.microsoft.com/en-us/download/details.aspx?id=30679, make sure you download the x86 version of the redistributable and not the x64 version.

To start decoding with modesdeco2, open a command line prompt from Windows or Linux, navigate to the directory you've extracted modesdeco2 to and run the program using the following command.

```
modesdeco2.exe --gain 49.6 --web8088
```

After running this command you can then browse to http://127.0.0.1:8088 in your web browser to open the web interface.



If you don’t want to use the web interface, and instead want to feed the data to another program such as basestation you can use the following command where lat:lon is an optional setting for your current receiver station latitude and longitude.

```
modesdeco2.exe --gain 49.6 --net 10001 --sbs10001 --location lat:lon
```

This mode outputs data in the base station format. To output in other formats such as AVR that are usable by other graphical radar displays like Virtual Radar Server, use the following command.

```
modesdeco2 --gain 49.6 --sbs10001 10001 --avr 44044 --msg 30003 --location lat:lon --filter-nocountry --rbs --verbose
```

If you want modesdeco2 to be able to display flight extra information such as registration numbers, models, airlines, flight routes, airplane images, silouheutes you’ll need to supply the appropriate commands to the input command.

To get the registration numbers, models, airline information, silhouettes you can use a ready made basestation.sqb file which can be downloaded from http://www.sbs-resources.com/download/index.html. Download and install this program, and the basestation.sqb file will be extracted to C:\SBS-resources\Files\Registration. The silouettes will be stored in C:\SBS-resources\Files\SilhouettesLogos. Then use the modesdeco2 flags --db and --silhouettes to specify these paths. If you want to see the commercial flight routes on the map, then you can download a prepopulated flightroute.sqb data base from this Yahoo! group: https://groups.yahoo.com/neo/groups/PP-logs-and-routes/files, save the flightroute.sqb file into a folder, and let modesdeco2 know about its location with the --frdb flag.

```
modesdeco2.exe --gain 49.6 --web8088 --db c:\SBS-resources\Files\Registration --silhouettes c:\SBS-resources\Files\SilhouettesLogos --frdb c:\flightroutedb\flightroute.sqb
```

COCOA1090 (MAC OSX)

This is a ADS-B decoder written for Mac OS X. It can be downloaded from http://www.blackcatsystems.com/software/cocoa1090.html. It requires use of rtl_tcp.

GR-AIR-MODES (GNU RADIO)

Gr-air-modes is a GNU Radio program for decoding ADS-B. The gr-air-modes project website is at https://github.com/bistromath/gr-air-modes. Gr-air-modes can output to a .kml file and then the aircraft can be viewed in real time 3D in Google Earth.

MODESMIXER

Modesmixer is a little different to the other decoders in that it is not actually a decoder. Rather it is a command line tool that can be used to combine the outputs from multiple decoders and then output a single combined stream. This is useful if you have multiple computers running ADS-B decoding software and would like to combine the results. Modesmixer can be downloaded from a forum post at http://radarspotting.com/forum/index.php/topic,2978.msg15240.html#msg15240. ModeSMixer can mix data from multiple ADS-B decoders, such as from a mix from dump1090, RTL1090 or ADSB#.

For example, to combine two data sources together, one from a remote PC on your network and one from your local PC, you'd specify the two IP addresses of the ADS-B data sources and specify the output data format you would like to use. See the example below. The full list of arguments and output formats available from Modesmixer can be found by using `modesmixer2 -h` at the command line. If you want to use the web interface like in modesdeco2 you can also supply the --web flag.

```
modesmixer2 --inConnect 192.168.1.127:30005 --inConnect 127.0.0.1:30005 --outserver beast:31001 --web 8088
```

DUMP978

dump978 is identical to dump1090, except that instead of decoding ADS-B at 1090 MHz, it decodes the similar UAT protocol at 978 MHz. UAT is used in the

USA most commonly by small aircraft which find it cheaper and more useful to operate. dump978 is current in it’s experimental beta release stages so it’s performance may not be perfect. It can be downloaded from https://github.com/mutability/dump978.

GRAPHICAL RADAR DISPLAY SOFTWARE FOR WINDOWS

Most of the decoders mentioned above send the decoded ADS-B data through a local (or public if desired) network connection. There are a few software options available for receiving this network data and displaying it. Here we show four of the most popular, three are free, and one is paid with a trial option. To use these programs you will need to be running your ADS-B decoder at the same time as these programs.

VIRTUAL RADAR SERVER (WINDOWS)

Virtual Radar Server (VRS) is Windows and Linux Mono based free software program which displays plane positions in a web browser using Google maps. The map can then be shared with other people over the internet if desired. Virtual Radar Server can be downloaded from http://www.virtualradarserver.co.uk/.

If you want to view other public Virtual Radar Servers, search Google using the string inurl:"VirtualRadar/GoogleMap.htm" to find many public Virtual Radar Servers.

1. First download and install VRS from its website. Then open VRS from the start menu.

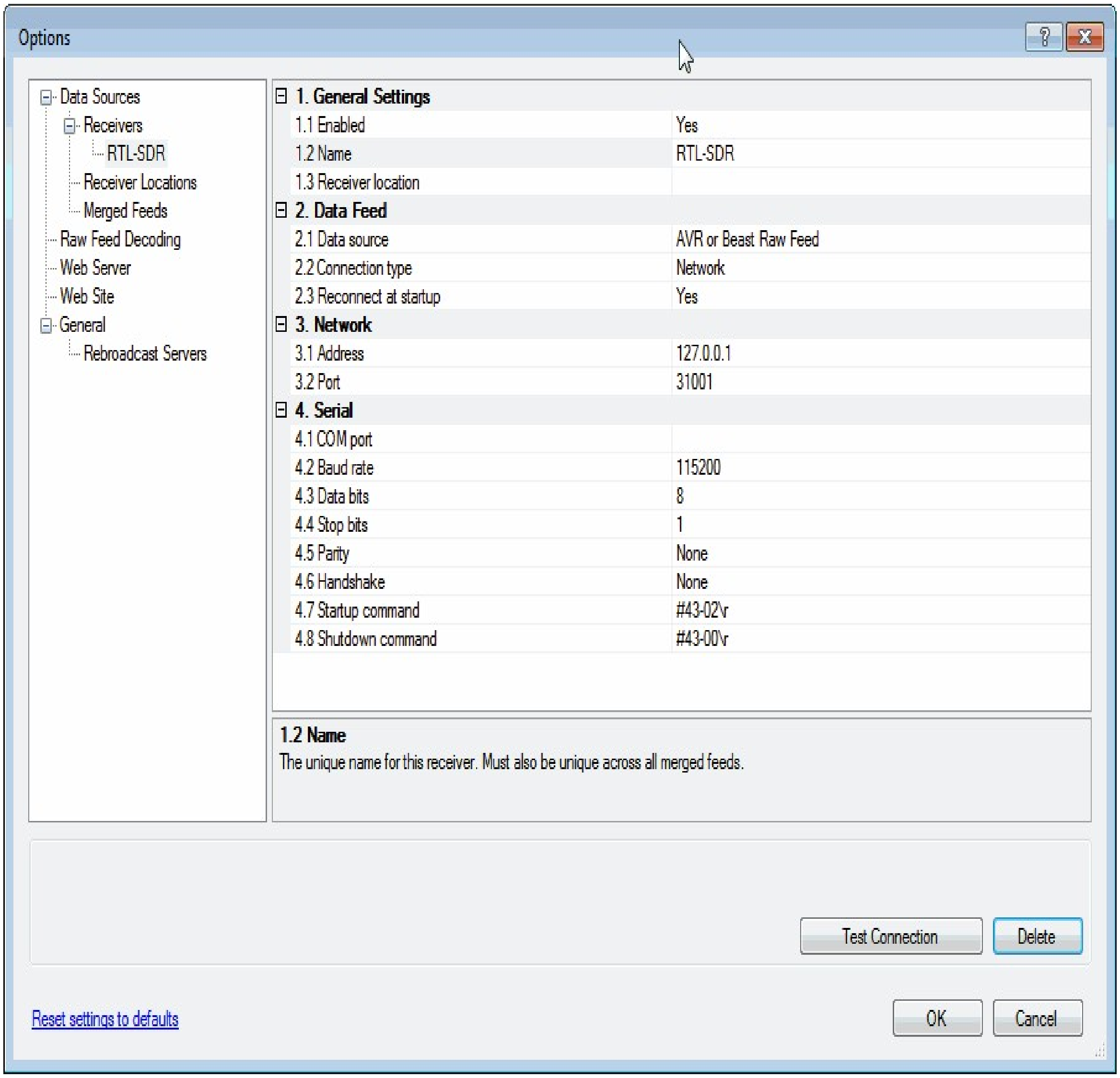
2. Go to **Tools->Options**.

3. Click on Receivers, then New. The properties of this receiver will show.

4. Name your Receiver RTL-SDR under General Settings.

5. Change the data source to “**AVR or Beast Raw Feed**”.

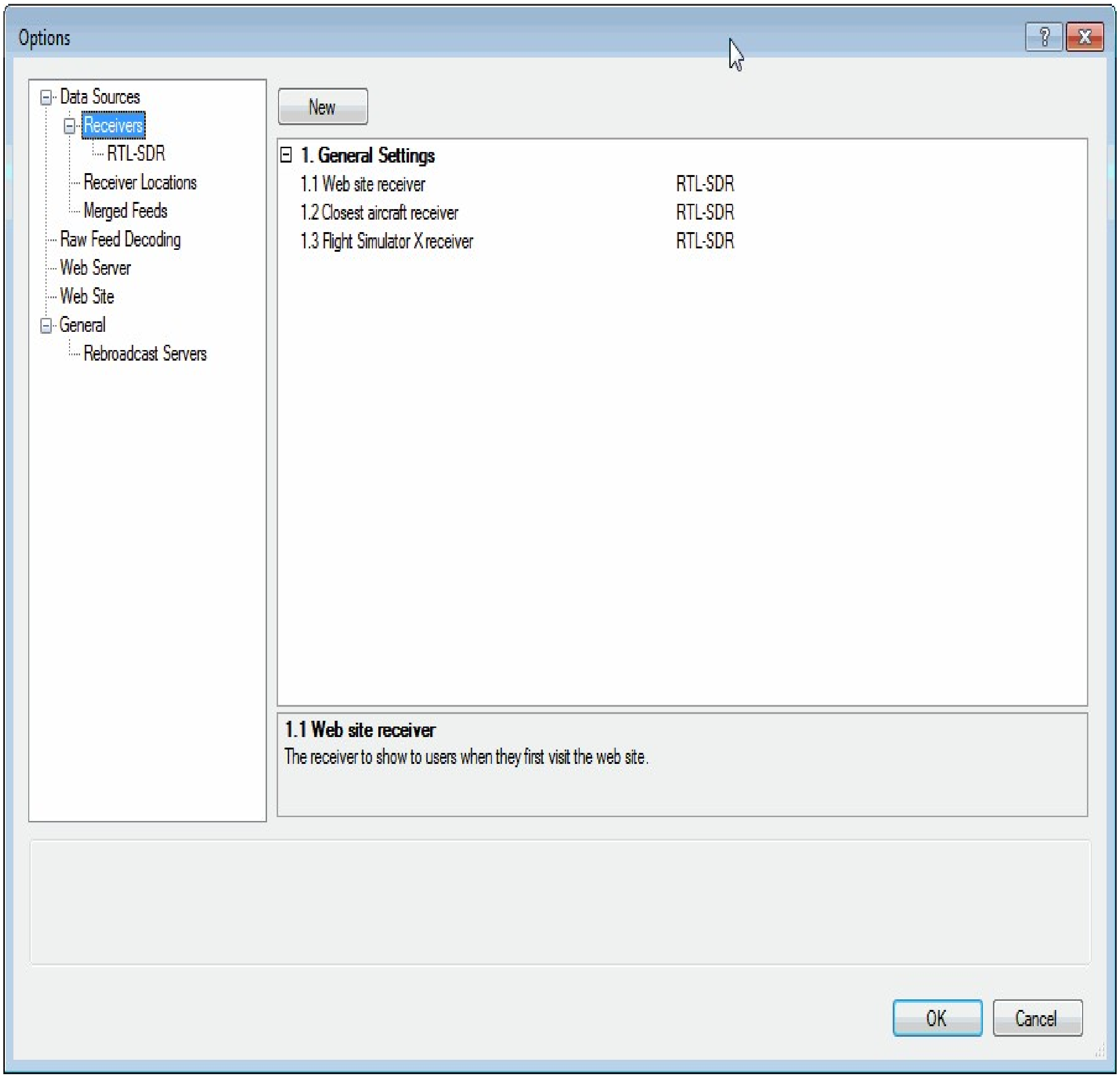
6. Keep the IP Address at 127.0.0.1 (unless you are using a remote server), but change the port number to 47806 if using ADSB#, or 31001 if using RTL1090. (Or set the port to a custom value in ADSB#). The following picture shows the settings you should be using if using RTL1090.



1. General Settings	
1.1 Enabled	Yes
1.2 Name	RTL-SDR
1.3 Receiver location	
2. Data Feed	
2.1 Data source	AVR or Beast Raw Feed
2.2 Connection type	Network
2.3 Reconnect at startup	Yes
3. Network	
3.1 Address	127.0.0.1
3.2 Port	31001
4. Serial	
4.1 COM port	
4.2 Baud rate	115200
4.3 Data bits	8
4.4 Stop bits	1
4.5 Parity	None
4.6 Handshake	None
4.7 Startup command	#43-02\r
4.8 Shutdown command	#43-00\r

1.2 Name
The unique name for this receiver. Must also be unique across all merged feeds.

7. Click on Receivers and set the “Web site receiver”, “Closest aircraft receiver” and “Flight Simulator X receiver” to the RTL-SDR receiver you’ve just created. Press OK.

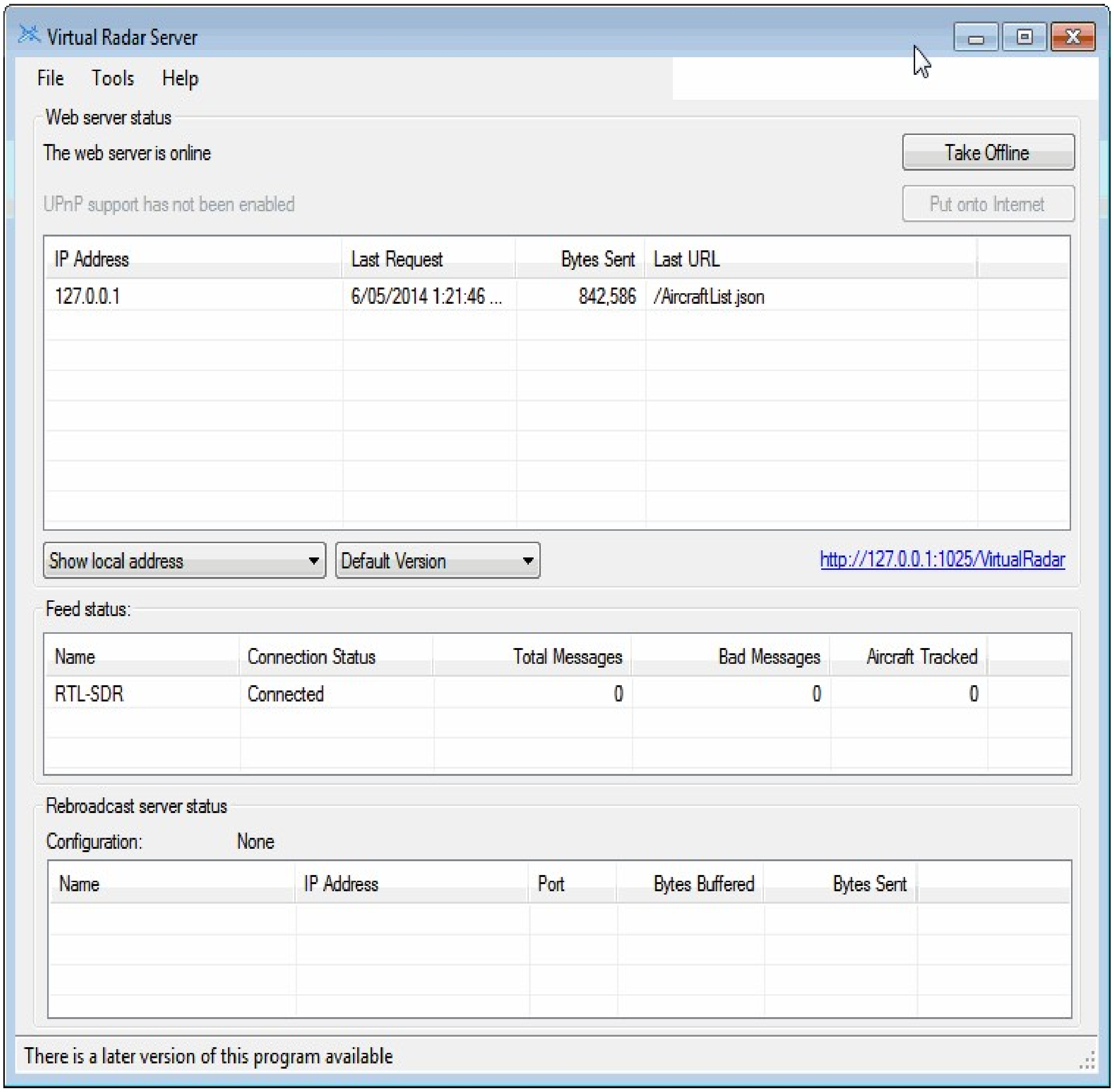


1. General Settings

1.1 Web site receiver	RTL-SDR
1.2 Closest aircraft receiver	RTL-SDR
1.3 Flight Simulator X receiver	RTL-SDR

1.1 Web site receiver
The receiver to show to users when they first visit the web site.

8. Now click on the link to http://127.0.0.1:1025/VirtualRadar in the middle right of the VRS main window to be taken to a web page with a special VRS Google map loaded.



Web server status

The web server is online

UPnP support has not been enabled

IP Address	Last Request	Bytes Sent	Last URL
127.0.0.1	6/05/2014 1:21:46 ...	842,586	/AircraftList.json

http://127.0.0.1:1025/VirtualRadar

Feed status:

Name	Connection Status	Total Messages	Bad Messages	Aircraft Tracked
RTL-SDR	Connected	0	0	0

Rebroadcast server status

Configuration: None

Name	IP Address	Port	Bytes Buffered	Bytes Sent

There is a later version of this program available

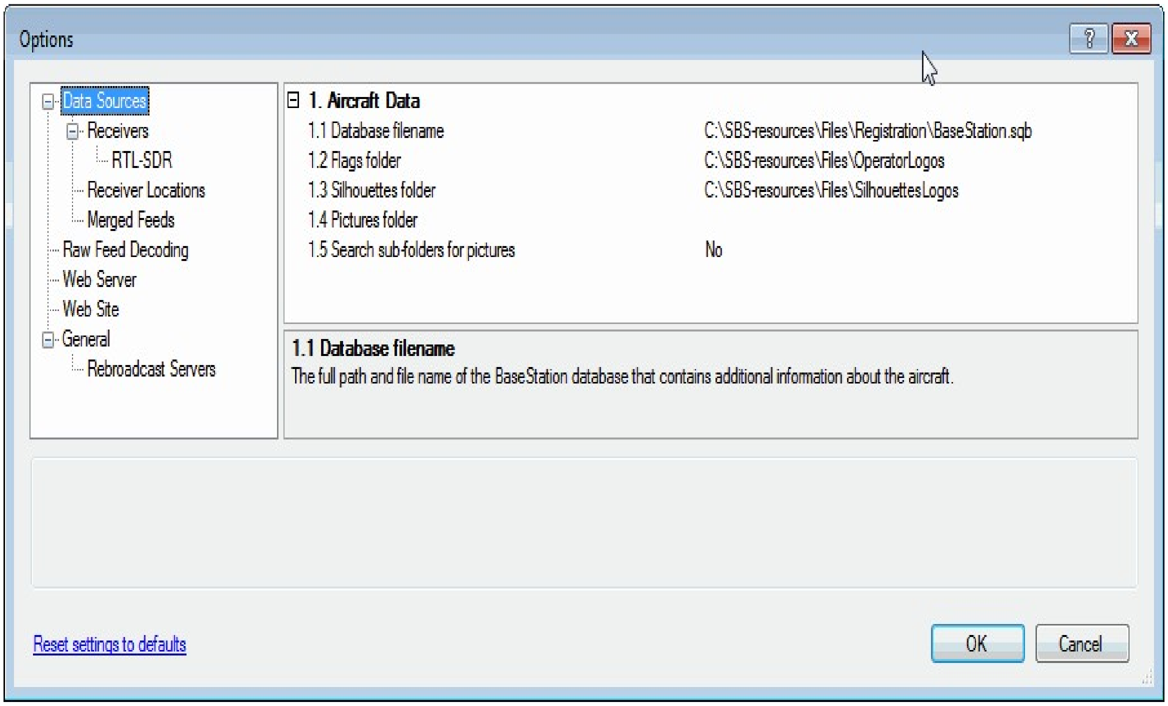
9. You should now be seeing a map. Zoom out and scroll to your location on the map. You should be able to see planes appearing if you are getting good ADS-B frame decodes from your ADS-B decoder program.

Now if you want to be able to see the aircraft registration number, operator, manufacturer, model, flag and silhouette information you will need to install a pre-made lookup database called basestation.sqb. To do this follow these instructions.

1. Go to http://www.sbs-resources.com/download/index.html and download and install the setup file. This will install the required files to a folder in C:\SBS-resources by default. If you change the default install folder be sure

to change the file paths in steps 3, 4 and 5. (An alternative basestation.sqb can be found here http://planebase.biz/bstnsqb)

2. Now in Virtual Radar Server go to **Tools->Options** and click on the Data Sources tab.
3. Set the Database file name to C:\SBS-resources\Files\Registration\BaseStation.sqb.
4. Set the Flags Folder to C:\SBS-resources\Files\OperatorLogos.
5. Set the Silhouettes folder to C:\SBS-resources\Files\SilhouettesLogos
6. Click OK. The database is now set up.

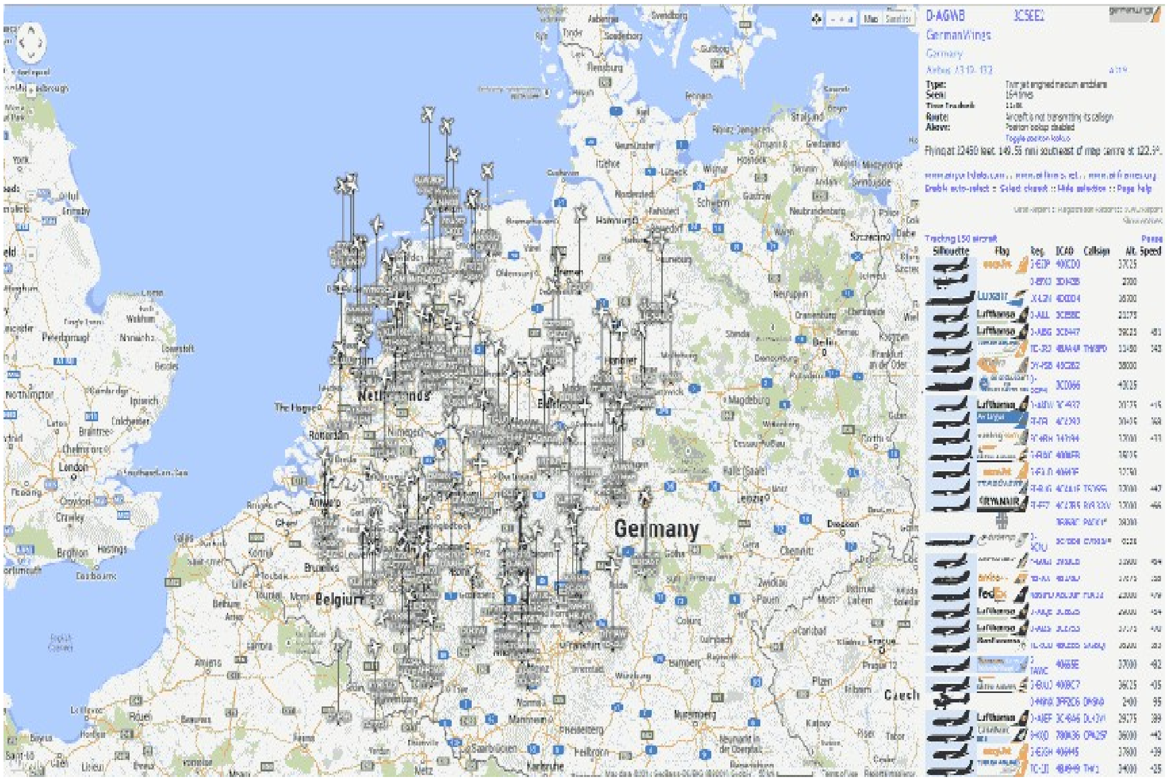


1. Aircraft Data

1.1 Database filename	C:\SBS-resources\Files\Registration\BaseStation.sqb
1.2 Flags folder	C:\SBS-resources\Files\OperatorLogos
1.3 Silhouettes folder	C:\SBS-resources\Files\SilhouettesLogos
1.4 Pictures folder	
1.5 Search sub-folders for pictures	No

1.1 Database filename
The full path and file name of the BaseStation database that contains additional information about the aircraft.

Below is an example of what the web interface of a Virtual Radar Server that has been set up looks like.



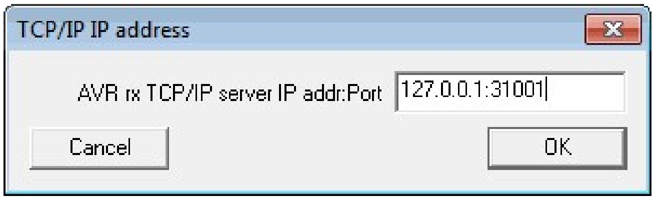
PLANEPLOTTER

PlanePlotter is sophisticated commercial software that has a free 21 day trial period. After the trial period a licence can be purchased for 25 Euros. PlanePlotter is also able to combine ACARS and ADS-B information together if you have two RTL-SDR sticks and can also be used to share data with the popular online ADS-B aggregator flightradar24.com. PlanePlotter can be downloaded from http://www.coaa.co.uk/planeplotter.htm.

One big advantage to using PlanePlotter is that it has a multilateration option which can be unlocked either by regularly contributing ADS-B data to their servers, or by paying 12 euros a year. Multilateration is a method which is used to estimate positions of planes that are broadcasting ADS-B signals without position data. It works by using data contributed from multiple locations and users to essentially triangulate plane positions.

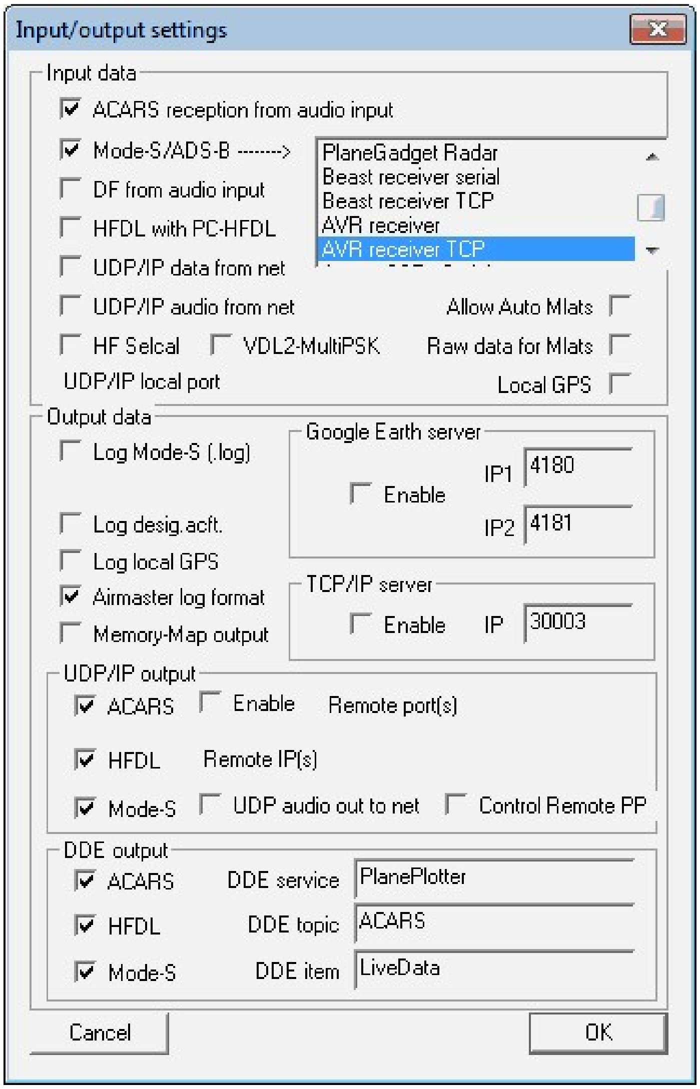
PlanePlotter is easy to set up with the RTL-SDR, see the instructions below.

1. Download and install PlanePlotter from the website. Open PlanePlotter.
2. Go to **Options->Mode-S receiver -> AVR Receiver -> TCP/IP Client**. Set the address and port to 127.0.0.1:31001.


TCP/IP IP address

AVR rx TCP/IP server IP addr:Port 127.0.0.1:31001

3. Go to **Options-> I/O Settings** and click the checkbox next to MODE-S/ADS-B and then select AVR receiver TCP in the box.


Input/output settings

Input data

☑ ACARS reception from audio input

☑ Mode-S/ADS-B -------->

PlaneGadget Radar
Beast receiver serial
Beast receiver TCP
AVR receiver
AVR receiver TCP

☐ DF from audio input

☐ HFDL with PC-HFDL

☐ UDP/IP data from net

☐ UDP/IP audio from net

Allow Auto Mlats ☐

☐ HF Selcal ☐ VDL2-MultiPSK

Raw data for Mlats ☐

UDP/IP local port

Local GPS ☐

Output data

☐ Log Mode-S (.log)

☐ Log desig.acft.

☐ Log local GPS

☑ Airmaster log format

☐ Memory-Map output

Google Earth server

☐ Enable

IP1 4180

IP2 4181

TCP/IP server

☐ Enable IP 30003

UDP/IP output

☑ ACARS ☐ Enable Remote port(s)

☑ HFDL Remote IP(s)

☑ Mode-S ☐ UDP audio out to net ☐ Control Remote PP

DDE output

☑ ACARS DDE service PlanePlotter

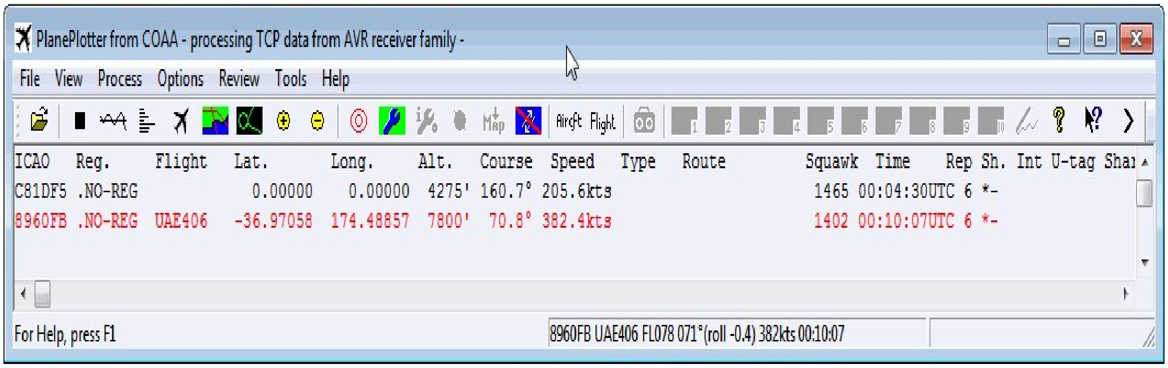
☑ HFDL DDE topic ACARS

☑ Mode-S DDE item LiveData

4. Set the port number in ADSB# to 31001 (or use a different port in step 2), or use RTL1090 which has the default TCP/IP port set at 31001. Turn on ADSB# or RTL1090.

5. Back in Planeplotter click the green start button ●.

6. You can now switch to the Aircraft view ✈ to see a list of detected aircraft.

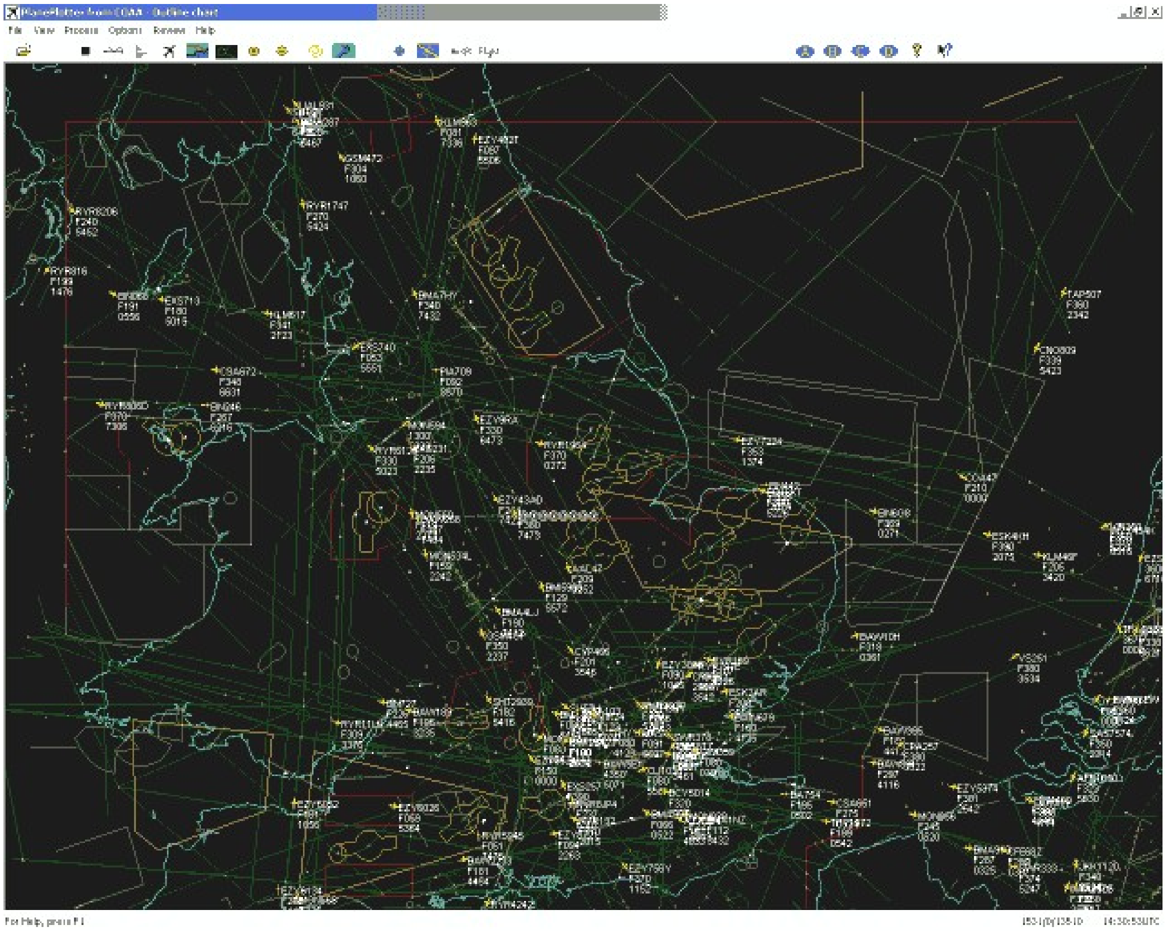


PlanePlotter from COAA - processing TCP data from AVR receiver family -
File View Process Options Review Tools Help
Airft Flight

```
ICAO   Reg.     Flight   Lat.       Long.      Alt.   Course  Speed     Type  Route        Squawk  Time          Rep Sh. Int U-tag Shai
C81DF5 .NO-REG            0.00000    0.00000   4275'  160.7°  205.6kts                     1465   00:04:30UTC   6  *-
8960FB .NO-REG  UAE406  -36.97058  174.48857   7800'   70.8°  382.4kts                     1402   00:10:07UTC   6  *-
```

For Help, press F1 | 8960FB UAE406 FL078 071°(roll -0.4) 382kts 00:10:07

7. Switch to Chart View 🗺, and use the mouse wheel to zoom in and out. Scroll to where some aircraft are showing and then click on the Download Map Button Map to download a Google map. Every time you move or zoom click this button again to re-download a new map. You can also click the satellite button 🌍 to download a satellite view.



PlanePlotter from COAA - Outline chart
File View Process Options Review Help
For Help, press F1
14:30:53UTC

Using PlanePlotter it is also possible to view aircraft in three dimensions using Google Earth. To view the aircraft in Google Earth, you will need to enable the Google Earth Server and have Google Earth Installed. Google Earth can be downloaded from https://www.google.com/earth/.

1. Go to **Options->I/O Settings** and click the enable checkbox in the Google Earth Server boxed area and press OK. Note to access I/O settings you'll need to first stop PlanePlotter decoding by pressing the stop button ■ if it is running.
2. Restart ADS-B reception by pressing the green start button ● once again.
3. Now navigate using Windows Explorer to the folder that you have installed PlanePlotter to. The default installation path is C:\Program Files (x86)\COAA\PlanePlotter. In this folder locate the google_aircraft.kml file and double click it. This file should open automatically with Google Earth and display a top down view of aircraft. You can also open google_cockpit.kml for simulated cockpit views of the aircraft.

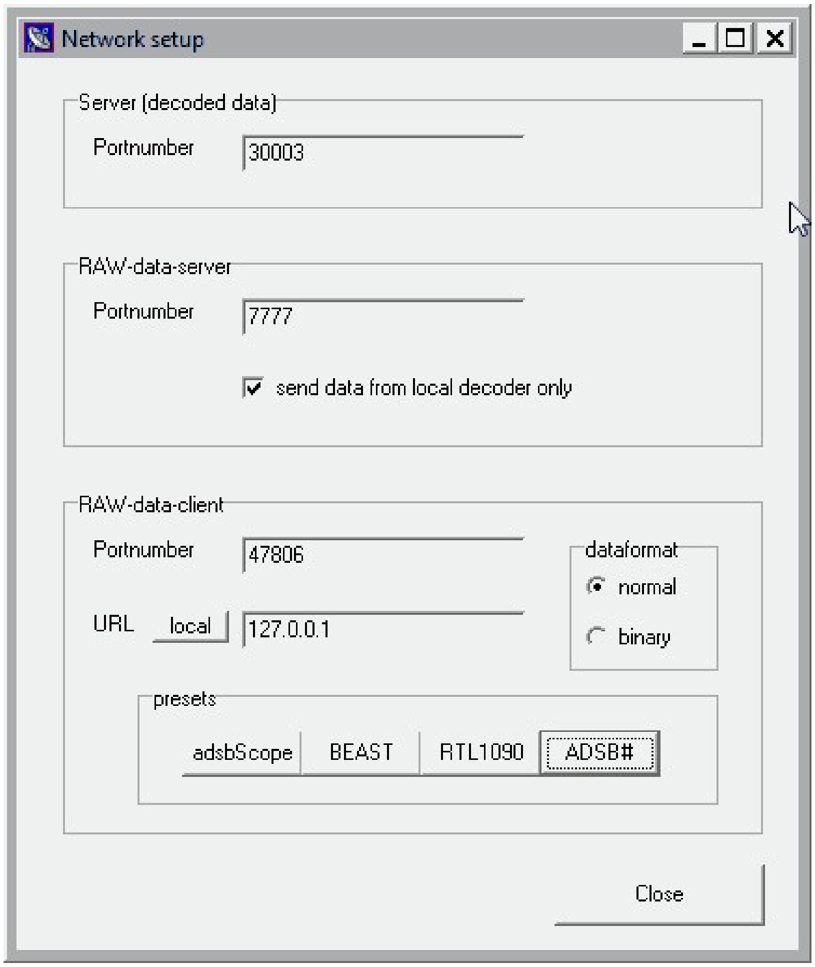
To install the basestation.sqb database like with Virtual Radar Server follow the instructions outlined at http://planeplotter.pbworks.com/w/page/17117301/Databases.

ADSBSCOPE

AdsbSCOPE is a free ADS-B radar program which can be downloaded from http://www.sprut.de/electronic/pic/projekte/adsb/adsb_en.html#downloads under the link that says "ZIP-File with all relevant data (13 MB)". Some people might prefer adsbSCOPE over the other programs shown above as it is free and not web browser based. The latest version of adsbSCOPE also has easy setup options for both ADSB# and RTL1090. To install and use adsbSCOPE follow the instructions below.

1. Download and unzip adsbSCOPE to a folder. Then navigate to the pc_software/adsbscope/26 folder. Run adsbscope26_16384.exe to open the program.
2. Go to **other->Network->Network setup**.
3. The following screen will be shown. Under presets click either on ADSB# or RTL1090 depending on which decoder you have chosen to use. Click on the local button as well, unless you are using a shared server, in which case enter

the URL of the server. If you have altered the ports on either decoder set the Portnumber in the RAW-data-client boxed area to be the same.



Network setup

Server (decoded data)
Portnumber 30003

RAW-data-server
Portnumber 7777
☑ send data from local decoder only

RAW-data-client
Portnumber 47806
URL 127.0.0.1
dataformat
● normal
○ binary

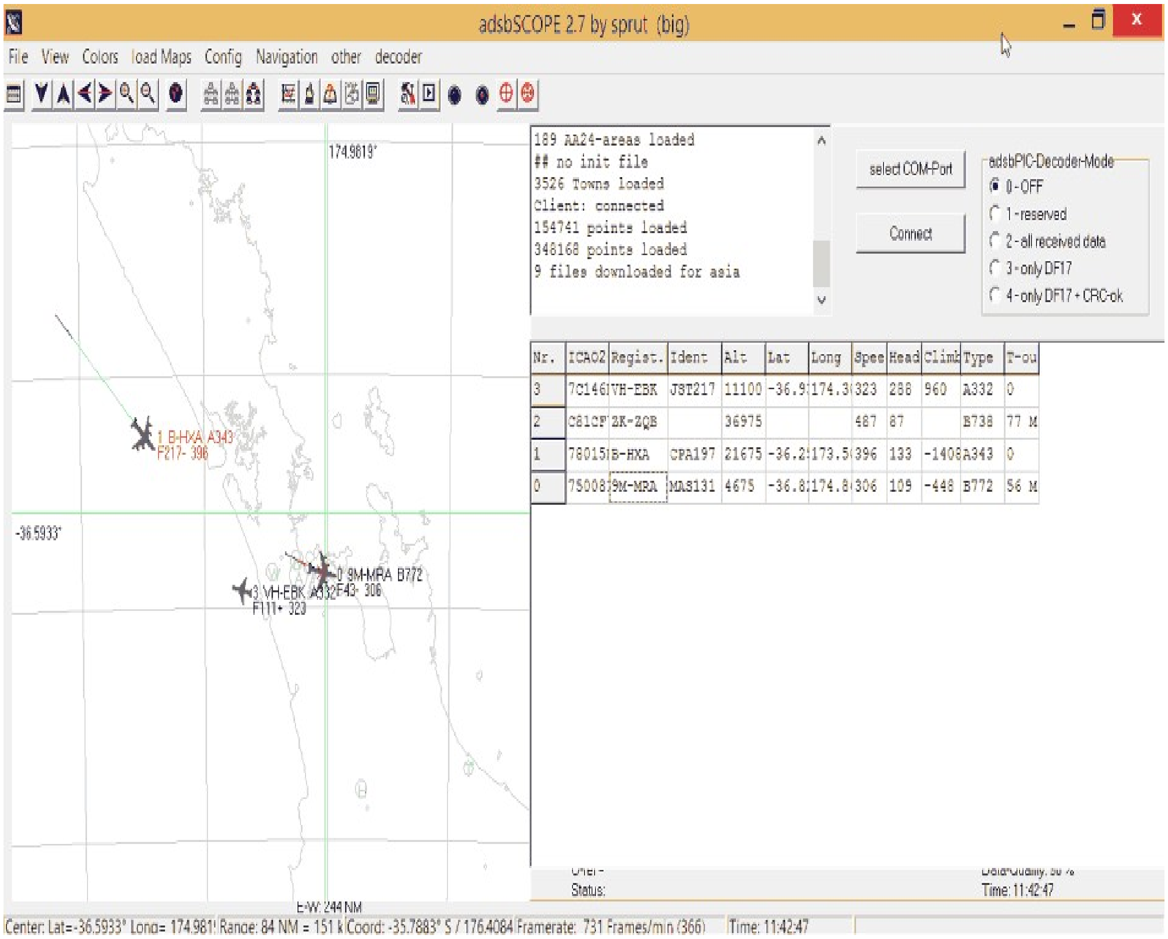
4. Now turn on RTL1090 or ADSB#.

5. In adsbSCOPE click the start RAW-data CLIENT button . After pressing it the button will become coloured . ADS-B data should now be shown in adsbSCOPE.

6. You can use the mouse to pan around the map and the zoom buttons to zoom in and out. You can also click the map button to download a map of the area currently shown in the screen.

7. Use the mouse and zoom tools to centre the cross hairs on the location of your receiver.

8. Go to **Navigation->set Receiver Location** to set the receiver location to the current position of the cross hair.



```
189 AA24-areas loaded
## no init file
3526 Towns loaded
Client: connected
154741 points loaded
348168 points loaded
9 files downloaded for asia
```

adsbPIC-Decoder-Mode
- ● 0 - OFF
- ○ 1 - reserved
- ○ 2 - all received data
- ○ 3 - only DF17
- ○ 4 - only DF17 + CRC-ok

Nr.	ICAO2	Regist.	Ident	Alt	Lat	Long	Spee	Head	Climb	Type	T-ou
3	7C146	VH-EBK	JST217	11100	-36.9	174.3	323	288	960	A332	0
2	C81CF	ZK-ZQB		36975			487	87		B738	77 M
1	78015	B-HXA	CPA197	21675	-36.2	173.5	396	133	-1408	A343	0
0	75008	9M-MRA	MAS131	4675	-36.8	174.8	306	109	-448	B772	56 M

Status:
Time: 11:42:47

Center: Lat=-36.5933° Long= 174.981 Range: 84 NM = 151 k Coord: -35.7883° S / 176.4084 Framerate: 731 Frames/min (366) Time: 11:42:47

ADSBScope can also create a max range plot which can over a period of time show how well your antenna is working. This will draw a line to each of the furthest aircraft seen over time. To turn it on simply follow these steps.

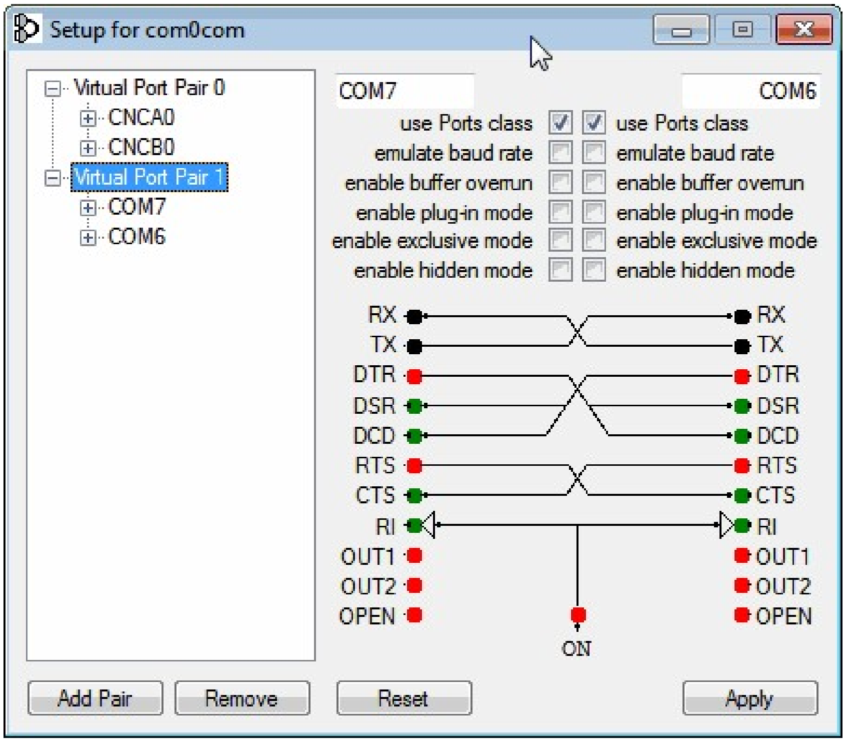
1. Go to **Config->Maximum Range** and enable ‘show all altitudes’.
2. Ensure that **View->Maximum Range** is checked.

BASESTATION

BaseStation is one of the original ADS-B display software packages. It was around when there were only dedicated ADS-B hardware receivers available. Many users prefer this software because it is what they are used to and it has a professional radar-like display.

To get basestation to work with RTL1090 or ADSB#, follow these instructions. To get it to work with Modesdeco you can skip to step 4 where Basestation is opened - we highly recommend using modesdeco as the decoder if you are wanting to use basestation due it's ease of use.

1. Download and install the com-port emulator com0com http://sourceforge.net/projects/com0com/. During installation you may have some problems with unsigned driver issues. To get around this you will need to disable signed driver checks for your particular OS after installing com0com and then restart your PC.
   After launching com0com for the first time take note of the numbers of the COM ports under Virtual Port Pair 1. They will be different depending on your PC setup. Here we have COM6 and COM7.



Setup for com0com

- Virtual Port Pair 0
  - CNCA0
  - CNCB0
- Virtual Port Pair 1
  - COM7
  - COM6

COM7			COM6
use Ports class	☑	☑	use Ports class
emulate baud rate	□	□	emulate baud rate
enable buffer overrun	□	□	enable buffer overrun
enable plug-in mode	□	□	enable plug-in mode
enable exclusive mode	□	□	enable exclusive mode
enable hidden mode	□	□	enable hidden mode

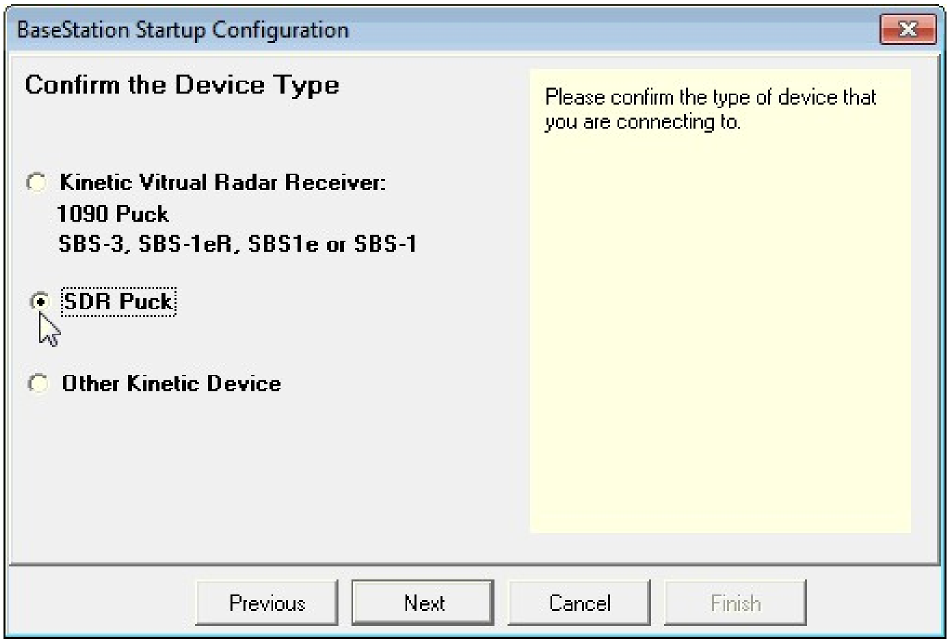
2. Download com2tcp from http://mode-s.66ghz.com/ and extract the exe file into a folder. In that folder create a batch file (To create a batch file simply create a .txt file, copy the command into the text file, then in Windows explorer rename the batch_file.txt to batch_file.bat). One batch file should contain the command in the following line and another containing the command in the second line. These commands are for RTL1090. If you want

to use ADSB# instead of RTL1090 change the port 31001 to the ADSB# default port of 47806 and change the `--sbsbin` flag to `--sbs.`

```
com2tcp --baud 3000000 --parity o \\.\COM6 127.0.0.1 31001
```

```
com2tcp --baud 3000000 --ignore-dsr --sbsbin \\.\COM7 10001
```

3. Now, run both batch files and then open RTL1090. RTL1090 should detect the TCP connection and the TCP light will turn green.
4. Open BaseStation.
5. An initial popup wizard will show. Follow the wizard and set your location and then your network address to 127.0.0.1 port 10001. When you get to the "Confirm the Device Type" screen first ensure you have RTL1090/ADSB# with com2tcp or Modesdeco running on port 10001 with ADS-B frames being received. Then choose SDR Puck and then next.



BaseStation Startup Configuration

Confirm the Device Type

Please confirm the type of device that you are connecting to.

○ Kinetic Vitrual Radar Receiver:
1090 Puck
SBS-3, SBS-1eR, SBS1e or SBS-1

● SDR Puck

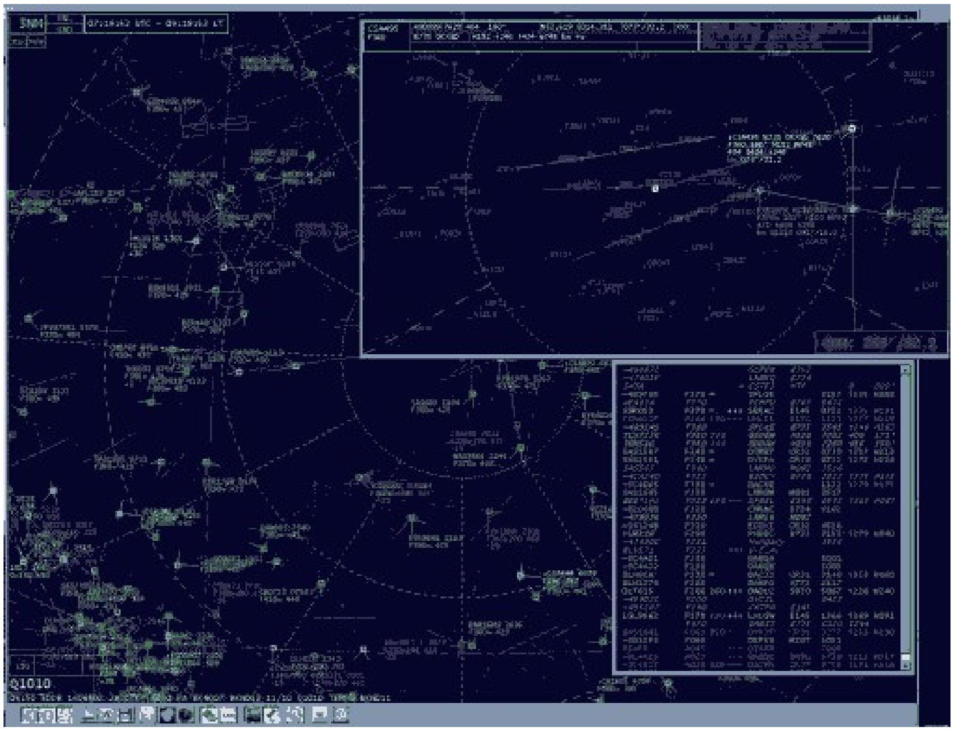
○ Other Kinetic Device

6. The Wizard will check to see if it can receive ADS-B packets so ensure your decoder is up and running and is seeing ADS-B data. If it is successful Basestation will open. If no frames are received during this time, the wizard may fail and you will need to try again.

Now Basestation should be set up and received aircraft should be showing on the screen.

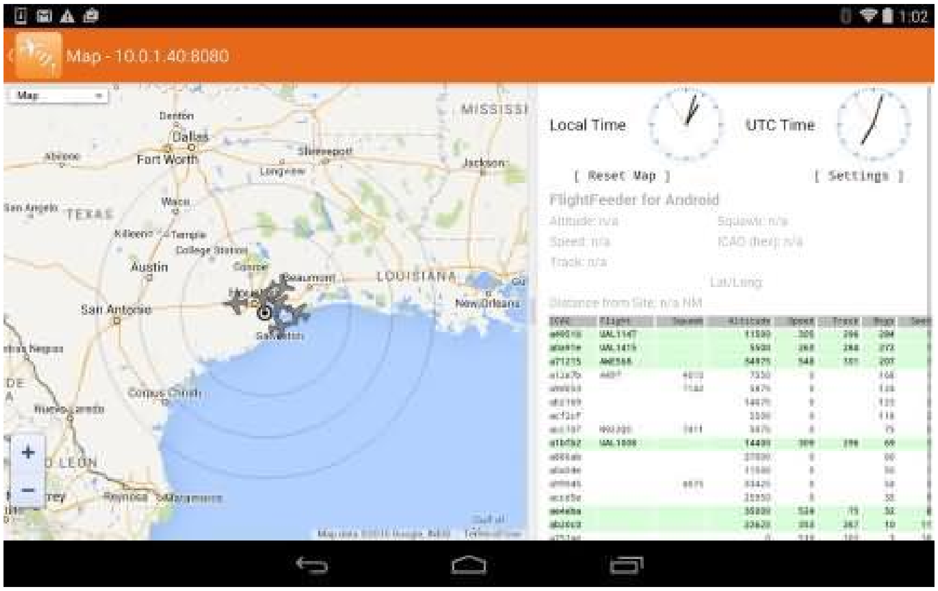
GLOBE-S

A lightweight radar viewer intended to be used with RTL1090. A version of Globe S is built into the beta 3 version of RTL1090. It can be downloaded from http://rtl1090.web99.de/homepage/index.php?way=1&site=READOUT&DERNAME=Globe-S%20RTL1090&dm=rtl1090&USER=rtl1090&goto=1&XURL=&WB=1&EXTRAX=X&PIDX=104245.



MONITORING ADS-B ON AN ANDROID DEVICE

There are several ADS-B apps available for Android, however the one we most reccommend using is the FlightAware FlightFeeder Android App. To download this app search for “FlightAware Feeder” on the Google Play store. This app uses an RTL-SDR to receive ADS-B signals on your Android device, and plots received aircraft on a map. In addition it not only receives ADS-B at 1090 MHz, but it can also receive and decode UAT at 978 MHz at the same time by toggling between the two frequencies. Finally, this app will also automatically broadcast the ADS-B data on your local network, so that the data can be accessed and plotted on a remote device such as a PC. It will also automatically send the data to the flightaware.com network, which is a crowd sourced ADS-B mapping site that combines the data from hundreds of ADS-B contributers.


Map - 10.0.1.40:8080

Local Time

UTC Time

[Reset Map]

[Settings]

FlightFeeder for Android

Altitude: n/a

Squawk: n/a

Speed: n/a

ICAO (hex): n/a

Track: n/a

Lat/Long:

Distance from Site: n/a NM

ICAO	Flight	Squawk	Altitude	Speed	Track	Msgs	Seen
[illegible]	UAL1147		11500	305	296	284	
[illegible]	UAL1415		5500	268	284	272	
a71275	AAE568		54875	548	101	207	
[illegible]	[illegible]	4010	7350	0		168	
[illegible]		7142	5875	0		126	
[illegible]			14075	0		123	
[illegible]			3300	0		118	
[illegible]	[illegible]	7411	5875	0		75	
[illegible]	UAL1008		14400	309	296	69	
[illegible]			27800	0		60	
[illegible]			11500	0		50	
[illegible]		6575	33425	0		58	
[illegible]			23350	0		35	
[illegible]			35000	524	75	32	8
[illegible]			22625	353	267	10	17
[illegible]			0	318	101	3	16

FEEDING DATA TO FLIGHTRADAR24.COM AND FLIGHTAWARE.COM

flightradar24.com and flightaware.com are services which crowd source ADS-B flight information from numerous sources around the world and share the aggregated data on their website. You can use your RTL-SDR ADS-B radar set up to contribute to these services by either using PlanePlotter, or downloading special software from their website. If you are a regular contributor, you will be rewarded with access to the sites premium features and with flightaware they will even buy you a free copy of PlanePlotter. Most commonly this software is set up on an embedded PC such as an Raspberry Pi.

Instructions for contributing to flightradar24 can be found at http://www.flightradar24.com/dvbt-stick.

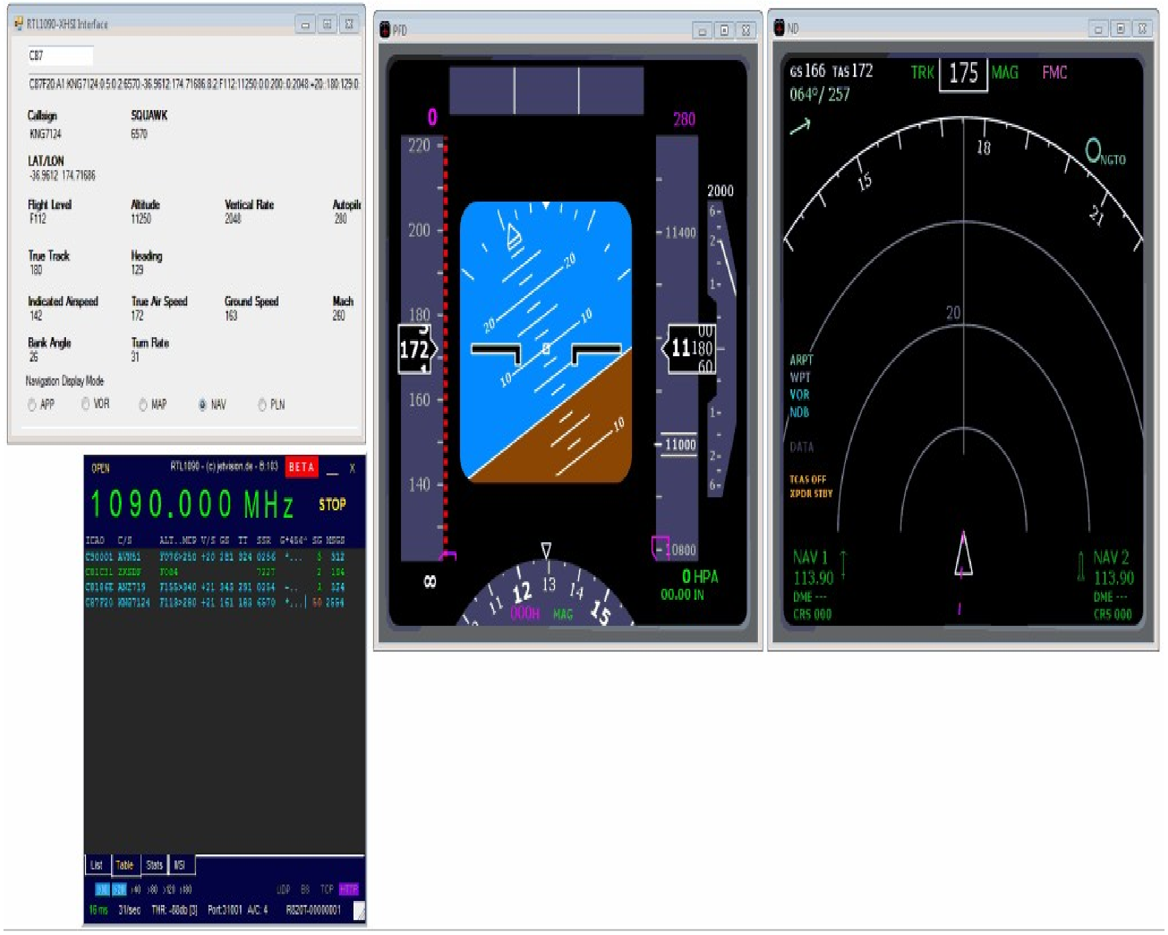
Instructions for contributing to flightaware can be found at https://flightaware.com/adsb/piaware/.

REAL TIME COCKPIT INSTRUMENT DISPLAY USING ADS-B DATA

It is possible to use received ADS-B data together with special software to display a real time cockpit view similar to what the pilot in the actual aircraft would see. To do this a program called XHSI can be used, which is a program used by Flight simulator enthusiasts to display cockpit data. Another program called XHSI-Interface can then be used to interface between RTL1090 and XHSI. To do this follow the steps below.

1. Download and install XHSI from http://sourceforge.net/projects/xhsi/.
2. Download the precompiled XHSI interface called RTL1090-XHSI from http://www57.zippyshare.com/v/49667810/file.html (Mirror: http://bit.ly/1zqJNpN)
3. Open RTL1090 and go into the configure menu. Change the Table 2 name under the HTTP server table names section to "tableb". Save and close the configure window.
4. Start RTL1090 (RTL1090 must be started first).
5. Start RTL1090-XHSI.
6. To begin monitoring data from a plane, enter the aircrafts ICAO number into the text box at the top of RTL1090-XHSI. Once entered you should begin to see that aircrafts ADS-B data show in RTL1090-XHSI.
7. Now open XHSI and it should automatically begin to use the ADS-B data.

If you want the ground symbols such as Waypoints, NavAid and VORs to be displayed in the Navigation Display window, download the X-Plane 9.00-9.70 data from http://data.x-plane.com/get_data.html. Extract the data to a folder and in XHSI set the AptNav Resources directory to the directory where you have extracted the data.



C87

C87F20 A1 KNG7124 0.5 0.2 6570 -36.9612 174.71686 8.2 F112 11250 0 0 200 0 2048 +20 180 129 0

Callsign
KNG7124

SQUAWK
6570

LAT/LON
-36.9612 174.71686

Flight Level
F112

Altitude
11250

Vertical Rate
2048

Autopilo
280

True Track
180

Heading
129

Indicated Airspeed
142

True Air Speed
172

Ground Speed
163

Mach
280

Bank Angle
26

Turn Rate
31

Navigation Display Mode

○ APP ○ VOR ○ MAP ● NAV ○ PLN

```
ICAO   C/S     ALT..MCP V/S GS  TT  SSR  G*456^ SG MSGS
C90001 AVN61   F016>250 +20 181 024 0256 *...    3  312
C81C31 ZKSDF   F004                 7227         2  104
C8106E ANZ719  F156>040 +21 343 291 0254 -..     1  124
C87F20 KNG7124 F118>280 +21 161 183 6570 *...|  50 2664
```

MONITORING MILITARY AIRCRAFT

While military aircraft have ADS-B Mode S transponders, they generally do not transmit position information. Nevertheless, there are military aircraft spotting enthusiasts who enjoy receiving and logging ADS-B signals from military aircraft. Even though position info isn't received, the spotters can still obtain a military aircraft's hex code, registration number, aircraft type, the base station location and altitudes.

The enthusiasts at milaircomms.com have custom ADS-B software for the Raspberry Pi and Windows called MilAirComms1090 which logs military aircraft data and uploads and aggregates the data from many volunteers on their website. Because there are volunteers logging data all over the world (mostly in the USA and Europe), it is possible to find out where military aircraft have been in the past through the logs. You can find more information about this project at http://milaircomms.com/mil_air_modes_logger.html.

Similar software for another similar website at http://www.live-military-mode-s.eu/ can be found at http://www.live-military-mode-s.eu/share/V4.0/ with a tutorial at http://www.jc-images.de/docs/RTL1090_Mode_S_Logger_Virtual_Radar_Server_Configuration.pdf.

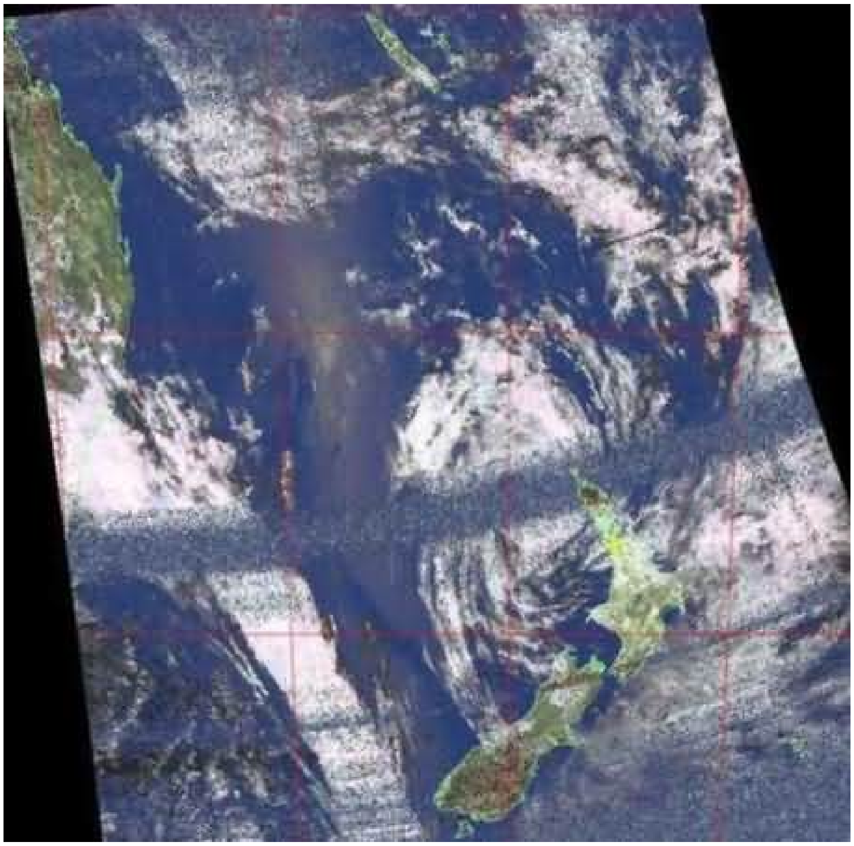
ADS-B FOR GLIDERS AND HELICOPTERS: FLARM

Small aircraft like gliders and some helicopters use a modified version of ADS-B called FLARM. FLARM is an acronym for FLight Alarm system. FLARM is a low cost and low power consumption ADS-B alternative which is often used by light aircraft for collision avoidance. It was invented because gliding enthusiasts needed a cheaper and easier to install version of the ADS-B collision avoidance system that would work with the close proximity flying that is common to gliders.

Although the signals are much weaker, with the right antenna it is possible to track these light aircraft just like commercial aircraft. This might be interesting for those living near a gliding club. FLARM signals are transmitted at 868 MHz and are effectively weaker by 100-1000 times compared to standard ADS-B signals, but are still very much receivable.

The Open Glider Network at http://wiki.glidernet.org/start are glider tracking enthusiasts who use RTL-SDRs and custom software to track gliders in real time. An example of aggregated live glider tracking can be seen on their website at http://live.glidernet.org/ and http://live.glidernet.org/3D/. The Open Glider Network project is mostly active in Europe.

NOAA WEATHER SATELLITE (APT) GUIDE



INTRODUCTION TO NOAA WEATHER SATELLITES

Everyday the American NOAA (National Oceanic and Atmospheric Administration) weather satellites pass over your location multiple times a day. As they pass over each NOAA weather satellite broadcasts an Automatic Picture Transmission (APT) signal which contains a live weather image of your area. The RTL-SDR dongle combined with a good antenna, an SDR program like SDR# and a decoding program can be used to download and display these live images.

There are currently three active NOAA satellites: NOAA 15, NOAA 18 and NOAA 19. Each NOAA satellite transmits at around 137 MHz with a right hand circularly polarized (RHCP) signal. This means that a right hand circularly polarized antenna is required for good reception.

The NOAA weather satellites use a transmission protocol known as Automatic Picture Transmission (APT) which was developed specifically for use on weather satellites. It is an analogue transmission that is somewhat similar to the HF Fax mode used on the HF bands. APT is transmitted in grayscale, but software can be used to colorize the image.

The NOAA satellites only pass overhead at certain times of the day, broadcasting a signal. These signals appear at around ~137 MHz and only when a satellite is passing overhead. Each satellite uses a different frequency. Their frequencies are shown below.

- NOAA 15 - 137.6200 MHz
- NOAA 18 - 137.9125 MHz
- NOAA 19 - 137.1000 MHz

This tutorial will show you how to set up a NOAA weather satellite receiving station, which will allow you to gather several live weather satellite images each day. Other SDRs and even some hardware radio scanners can also work with this tutorial, provided that the radio in question has a large IF bandwidth (30 kHz +) and a discriminator tap.

NOAA WEATHER SATELLITE RECEIVE TUTORIAL

To set up a NOAA weather satellite receive station you will need:

- An audio piping method. See Appendix A: Audio Piping if you don't have one set up.
- A right hand circularly polarized antenna tuned to 137 MHz.
- Software such as WXtoImg for decoding the APT signal.

NOAA WEATHER SATELLITE ANTENNAS

NOAA weather satellites transmit a right hand circularly polarized signal at around 137 MHz. For satellite reception it is desirable to have a circularly polarized antenna with gain directed towards the sky. The two most common antennas that are used for this type of signal are the turnstile and quadrifilar helix antennas.

A turnstile antenna is simply two dipoles aligned at right angles to each other and connected in a way so that their signals are 90 degrees out of phase, giving the circular polarization. The turnstile is also sometimes known as a cross dipole. If you are searching for turnstile antenna build tutorials, beware that there are two types of turnstiles. There are normal and axial mode turnstiles. For APT satellite reception we want the axial mode turnstile which directs most of its radiation towards the sky.

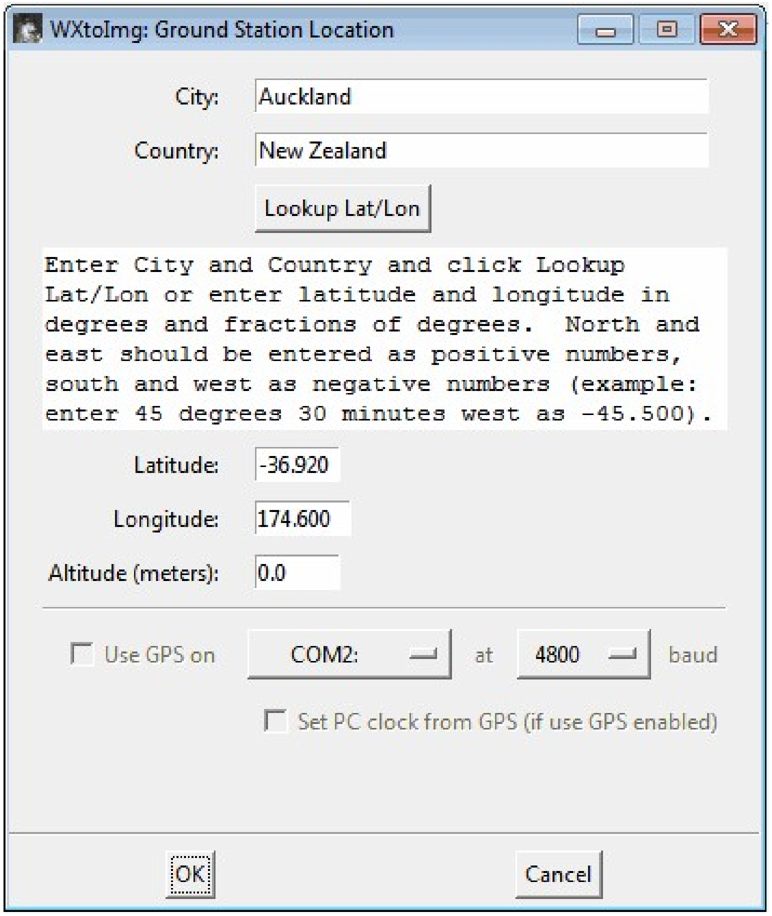
A Quadrifilar Helix (QFH) is a circularly polarized antenna that can be constructed out of PVC pipe and coax cable, or copper pipe. Most people report that the QFH antenna has slightly superior reception compared to the turnstile, but a few people have reported better reception with the turnstile. It may pay to experiment with both if you find your first choice does not work so well. Another antenna that works well with the NOAA APT satellites is the double cross antenna (DCA). It is basically four dipoles arranged in a certain way to produce circular polarization. It is possible that other antennas such as discones and quarter wave ground planes may also receive the NOAA signal, however if it is not a circularly polarized antenna the signal will fade in and out heavily. See the antenna guide for more information about antennas.

Many people have reported that an LNA preamp at the antenna and a broadcast FM bandstop filter can significantly help improve reception. See the LNA section and preselector section for more information.

WXTOIMG SOFTWARE TUTORIAL

WXtoImg is a free weather satellite decoding program for Windows, Linux, OSX and even the Raspberry Pi. It can be used to decode the APT signal and also tell you the frequencies and times of the satellite passes. There is also a paid version of WXtoImg which can unlock more features, however it is not required for basic usage use with RTL-SDR. To use WXtoImg and SDR# together follow the instructions below.

1. First, download and install WXtoImg from http://www.wxtoimg.com/.
2. Open WXtoImg and then set your Ground Station Location, (which is the coordinates of your antenna) by going to **Options -> Ground Station Location**. The city you are in should suffice, but you can be more accurate by entering in an exact latitude, longitude and altitude if you want. The location information is used to determine when a satellite will pass over your sky.


WXtoImg: Ground Station Location

City: Auckland

Country: New Zealand

Lookup Lat/Lon

```
Enter City and Country and click Lookup
Lat/Lon or enter latitude and longitude in
degrees and fractions of degrees.  North and
east should be entered as positive numbers,
south and west as negative numbers (example:
enter 45 degrees 30 minutes west as -45.500).
```

Latitude: -36.920

Longitude: 174.600

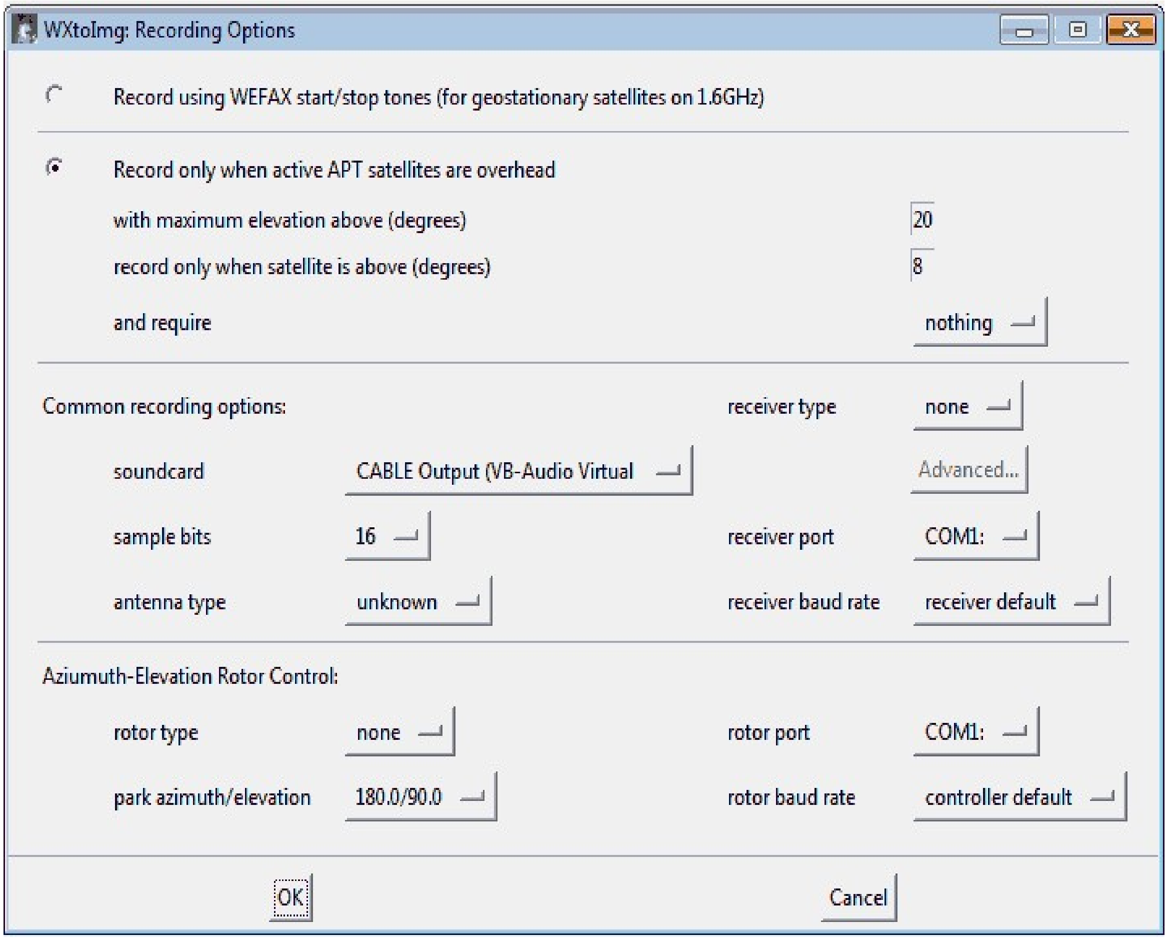
Altitude (meters): 0.0

☐ Use GPS on COM2: at 4800 baud

☐ Set PC clock from GPS (if use GPS enabled)

3. Set the audio piping method that you are using in WXtoImg. Go to **Options -> Recording Options** and ensure the correct audio device is selected under the soundcard option.

4. Also, in this window you can adjust the “Record only when active APT satellites are overhead” “with maximum elevation above (degrees)” and “record only when satellite is above (degrees)” settings. You may want to reduce the default values if you have an antenna with a good view of the sky and find that WXtoImg stops recording or doesn’t start fast enough even though the APT signal is present in SDR#.
Satellite passes with higher maximum elevations will get better reception as they will get a more unobstructed pathway to the antenna. When the satellite is low on the horizon, reception will probably be bad due to obstructions. You should set these values according to your terrain.



WXtoImg: Recording Options

○ Record using WEFAX start/stop tones (for geostationary satellites on 1.6GHz)

● Record only when active APT satellites are overhead

with maximum elevation above (degrees) 20

record only when satellite is above (degrees) 8

and require nothing

Common recording options:

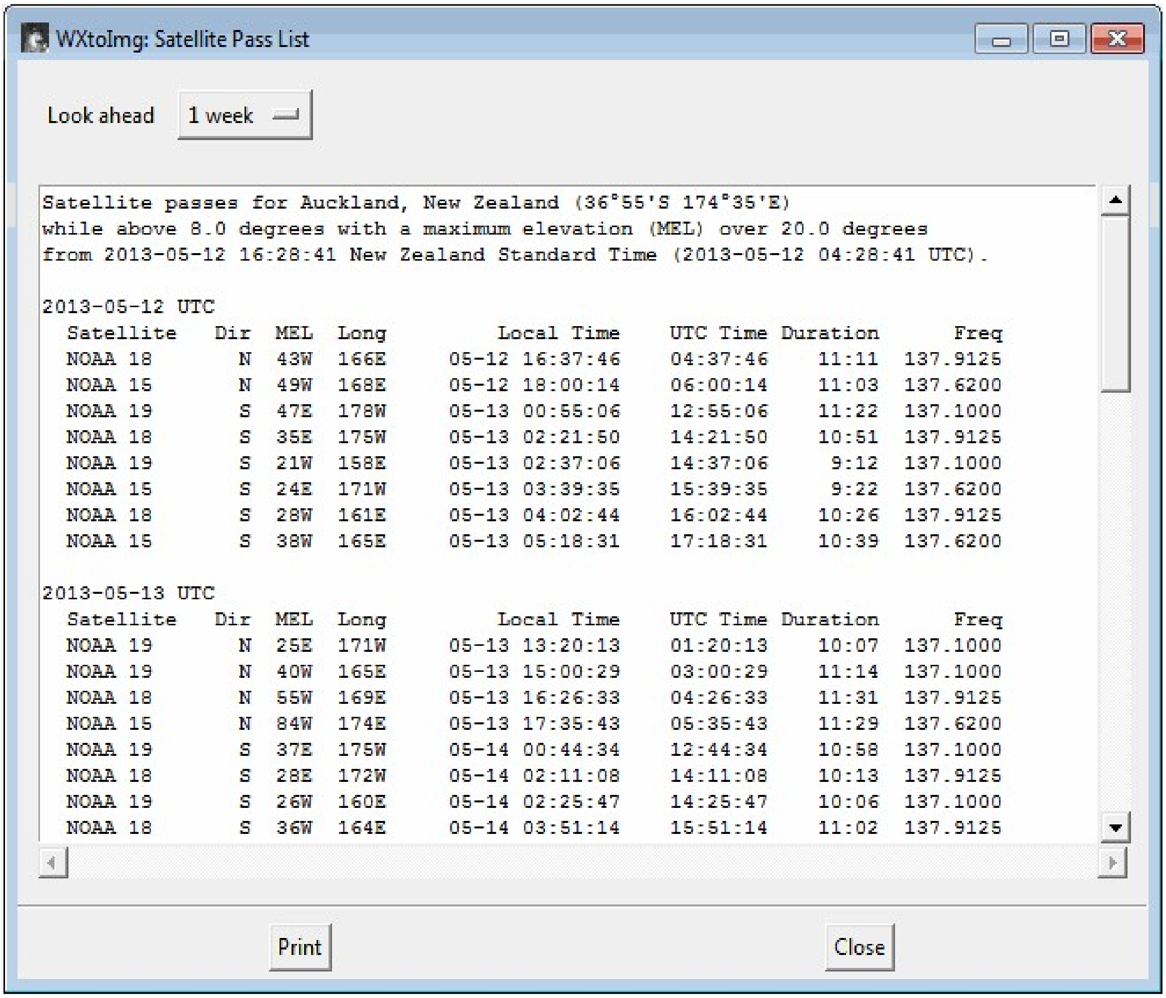
		receiver type	none
soundcard	CABLE Output (VB-Audio Virtual		Advanced...
sample bits	16	receiver port	COM1:
antenna type	unknown	receiver baud rate	receiver default

Aziumuth-Elevation Rotor Control:

rotor type	none	rotor port	COM1:
park azimuth/elevation	180.0/90.0	rotor baud rate	controller default

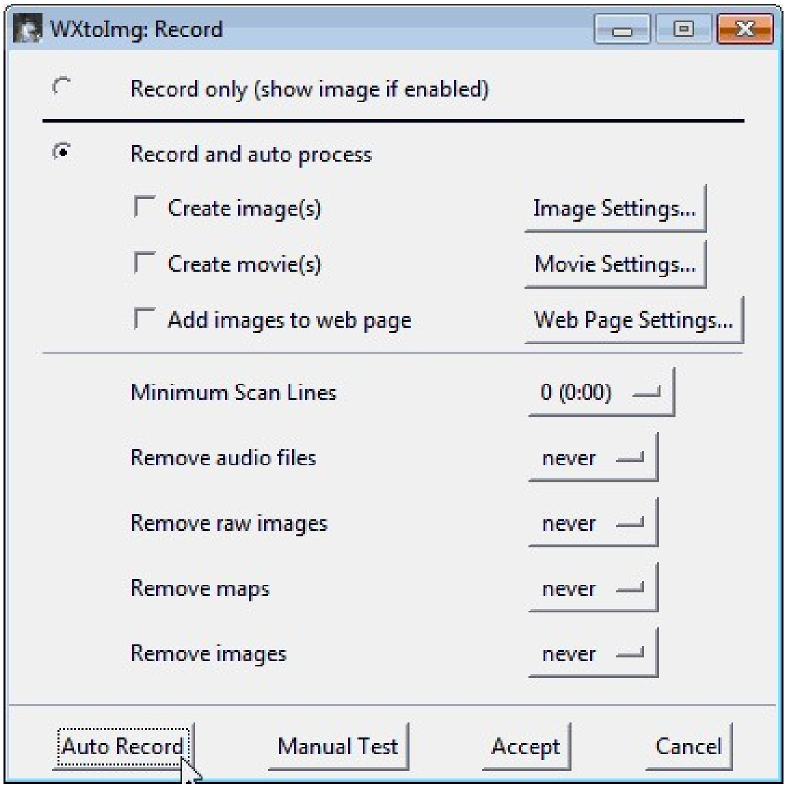
5. Now you will need to update your Kepler file. This file contains the information about satellite locations which need to be periodically updated because satellites drift in their orbit over time. Go to **File -> Update Keplers** to do this. Make sure you have an internet connection for the update.

6. Now you can go to **File -> Satellite Pass List** to find a time when a NOAA satellite will be passing overhead. Take note of the frequency of the pass as well.



```
Satellite passes for Auckland, New Zealand (36°55'S 174°35'E)
while above 8.0 degrees with a maximum elevation (MEL) over 20.0 degrees
from 2013-05-12 16:28:41 New Zealand Standard Time (2013-05-12 04:28:41 UTC).

2013-05-12 UTC
  Satellite   Dir  MEL  Long        Local Time     UTC Time Duration      Freq
  NOAA 18       N  43W  166E     05-12 16:37:46    04:37:46    11:11  137.9125
  NOAA 15       N  49W  168E     05-12 18:00:14    06:00:14    11:03  137.6200
  NOAA 19       S  47E  178W     05-13 00:55:06    12:55:06    11:22  137.1000
  NOAA 18       S  35E  175W     05-13 02:21:50    14:21:50    10:51  137.9125
  NOAA 19       S  21W  158E     05-13 02:37:06    14:37:06     9:12  137.1000
  NOAA 15       S  24E  171W     05-13 03:39:35    15:39:35     9:22  137.6200
  NOAA 18       S  28W  161E     05-13 04:02:44    16:02:44    10:26  137.9125
  NOAA 15       S  38W  165E     05-13 05:18:31    17:18:31    10:39  137.6200

2013-05-13 UTC
  Satellite   Dir  MEL  Long        Local Time     UTC Time Duration      Freq
  NOAA 19       N  25E  171W     05-13 13:20:13    01:20:13    10:07  137.1000
  NOAA 19       N  40W  165E     05-13 15:00:29    03:00:29    11:14  137.1000
  NOAA 18       N  55W  169E     05-13 16:26:33    04:26:33    11:31  137.9125
  NOAA 15       N  84W  174E     05-13 17:35:43    05:35:43    11:29  137.6200
  NOAA 19       S  37E  175W     05-14 00:44:34    12:44:34    10:58  137.1000
  NOAA 18       S  28E  172W     05-14 02:11:08    14:11:08    10:13  137.9125
  NOAA 19       S  26W  160E     05-14 02:25:47    14:25:47    10:06  137.1000
  NOAA 18       S  36W  164E     05-14 03:51:14    15:51:14    11:02  137.9125
```

7. When the time comes for the satellite to appear, open WXtoImg and then go to **File->Record**, and click on Auto Record. The recording and decoding will begin automatically when it is time for the satellite to appear on the horizon. It will automatically stop when it goes out of view according to the times in the satellite pass list. If for some reason the recording doesn't start in time, you can click on the Manual Test button to immediately begin decoding.



WXtoImg: Record

○ Record only (show image if enabled)

● Record and auto process

□ Create image(s)

□ Create movie(s)

□ Add images to web page

Minimum Scan Lines 0 (0:00)

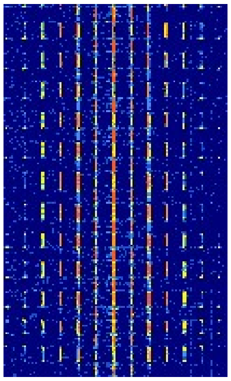
Remove audio files never

Remove raw images never

Remove maps never

Remove images never

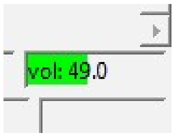
8. Open SDR# and select the audio piping method you are using under the Audio Output drop down box and then tune to the frequency that the satellite will be broadcasting at. Adjust the gain settings in SDR# under the Configure button so that you get good reception of the signal. Set the receive mode to WFM, filter bandwidth to 34 kHz and Filter Audio set to OFF. It may also be useful to ensure Snap to Grid is unchecked.



9. As the RTL-SDR is not frequency accurate and also due to the Doppler effect, the signal may not be at the exact frequency it should be

at. Just adjust the frequency in SDR# until the tuning bar is centred on the satellite signal.

10. Adjust the AF Gain in SDR# and/or Windows volume settings so that the volume bar in the bottom right hand corner of WXtoImg shows a green colour.



vol: 49.0

11. WXtoImg should now be decoding and showing the weather satellite image as it is received. You may need to periodically adjust the frequency to centre the signal as the Doppler effect will cause it to move. However, with RTL-SDR adjusting for the Doppler shift is not critical as the filter bandwidth can be simply set larger than 34 kHz (try 36 - 40 kHz) so that it is large enough to receive the entire signal even as it as it shifts.

12. Once the image has been fully received, you can play with the options under the Enhancements and Projection menu in order to add false colour and enhance the received image.

ORBITRON TUTORIAL

Using a program called Orbitron, SDR# can be made to automatically tune to NOAA and other satellites as they appear on your horizon. Also, although not entirely necessary for the RTL-SDR Orbitron will also help to compensate for satellite frequency drift due to the doppler effect.

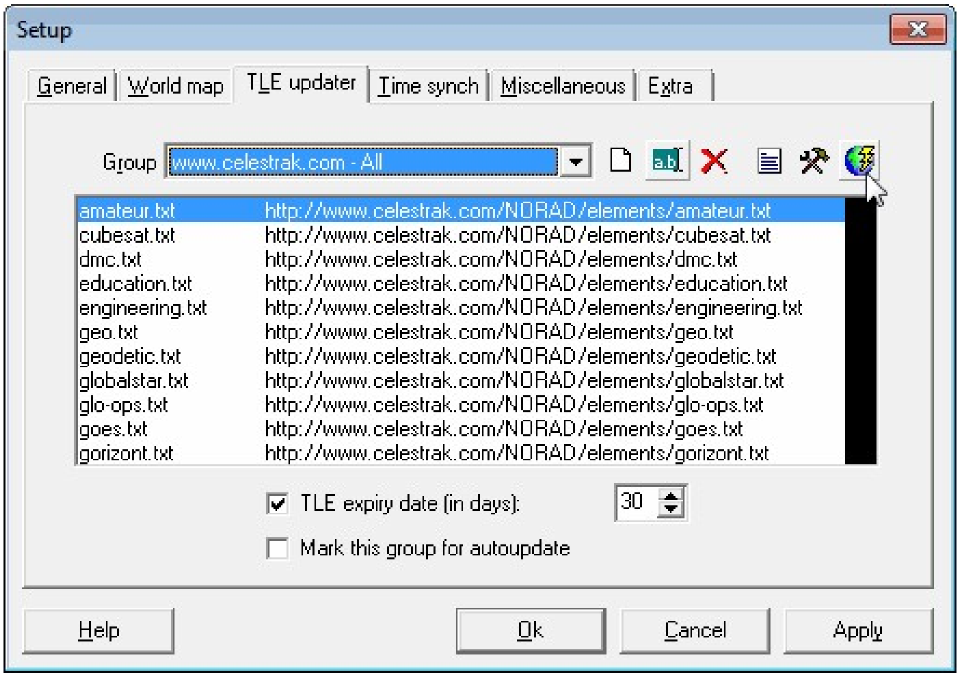
1. Download and install Orbiton from http://www.stoff.pl/.

2. Download the Orbitron DDE drivers from http://www.stoff.pl/orbitron/files/mydde.zip (Mirror: http://bit.ly/YR4hdd). Extract the files to the Orbitron installation directory (likely in Program Files (x86)/Orbitron).

3. Download the SDR# Orbitron plugin from http://public-xrp.s3.amazonaws.com/SatelliteTracker2.zip (Mirror: http://bit.ly/1u04xmV). The readme file comes with installation instructions, but some of the instructions did not work for us. You can try follow their instructions or the ones below.

4. Extract the SDR# Orbitron plugin files to the SDR# directory. With notepad or another text editor open Plugins.xml. Within the <sharpPlugins> </sharpPlugins> tags add the line <add key="SatelliteTracker" value="SDRSharp.SatelliteTracker.SatelliteTrackerPlugin,SDRSharp.SatelliteTracker" /> just below it.

5. Open Orbitron in Administrator Mode (if in Windows Vista/7/8), by right clicking it and selecting Run as Administrator. Orbitron may open in full screen mode. Press Alt+Enter to exit full screen if you wish. You will probably also be initially presented with a TLE file update screen. You can leave all the boxes as default. Click on the update button, which is the icon with a globe and lightning bolt 🌐. Orbitron will download the new TLE files. The TLE files contain the satellite orbit information and will need to be periodically updated every few days. Running Orbitron in Administrator mode is important, as otherwise the TLE files may not be able to be written to. If you don't see the setup screen, then you can get to it by clicking on the Setup icon ⚒, in the main Orbitron tab.



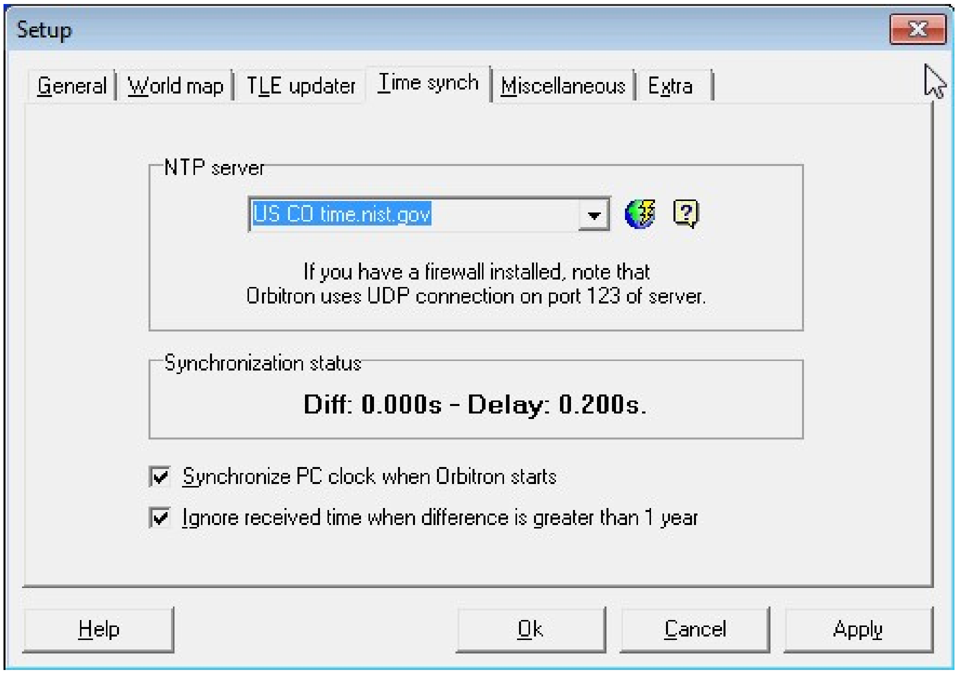
Group www.celestrak.com - All

```
amateur.txt        http://www.celestrak.com/NORAD/elements/amateur.txt
cubesat.txt        http://www.celestrak.com/NORAD/elements/cubesat.txt
dmc.txt            http://www.celestrak.com/NORAD/elements/dmc.txt
education.txt      http://www.celestrak.com/NORAD/elements/education.txt
engineering.txt    http://www.celestrak.com/NORAD/elements/engineering.txt
geo.txt            http://www.celestrak.com/NORAD/elements/geo.txt
geodetic.txt       http://www.celestrak.com/NORAD/elements/geodetic.txt
globalstar.txt     http://www.celestrak.com/NORAD/elements/globalstar.txt
glo-ops.txt        http://www.celestrak.com/NORAD/elements/glo-ops.txt
goes.txt           http://www.celestrak.com/NORAD/elements/goes.txt
gorizont.txt       http://www.celestrak.com/NORAD/elements/gorizont.txt
```

☑ TLE expiry date (in days): 30

☐ Mark this group for autoupdate

6. In order to have Orbitron accurately track the satellites it is important that your Windows PC time is accurate. Orbitron comes with a method to synchronise your PC time to the NTP servers, which provide accurate time. In the setup screen click on the Time Synch tab, and click on the Synchronise PC clock button 🌐 to automatically synchronise the time. You may also wish to select the Synchronise PC clock when Orbitron starts checkbox if your PC is always connected to the internet.



NTP server

US CO time.nist.gov

If you have a firewall installed, note that
Orbitron uses UDP connection on port 123 of server.

Synchronization status

Diff: 0.000s - Delay: 0.200s.

☑ Synchronize PC clock when Orbitron starts

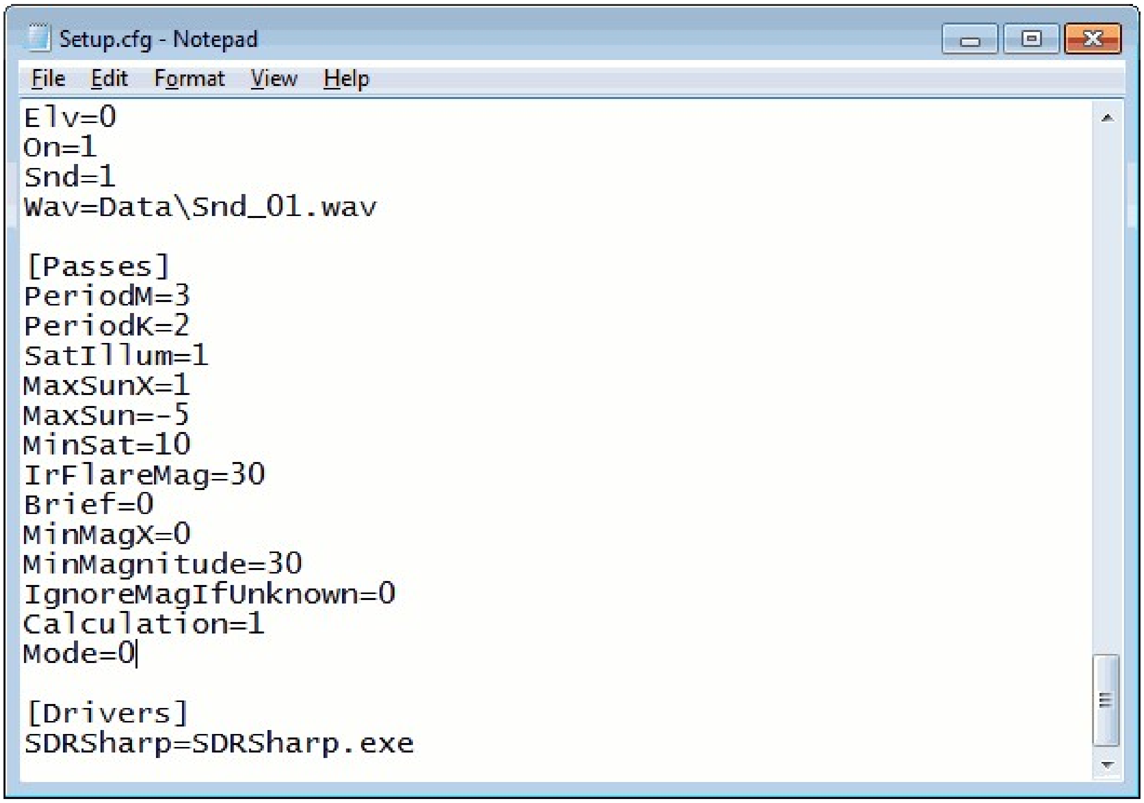
☑ Ignore received time when difference is greater than 1 year

7. Close Orbiton. Now open Notepad in Administrator mode, by right clicking its shortcut in the Start Menu and clicking on Run as Administrator.

8. In Notepad, go to **File->Open** and browse to your Orbitron\Config folder. Orbitron is probably installed in “Program Files (x86)\Orbitron”. Open Setup.cfg.

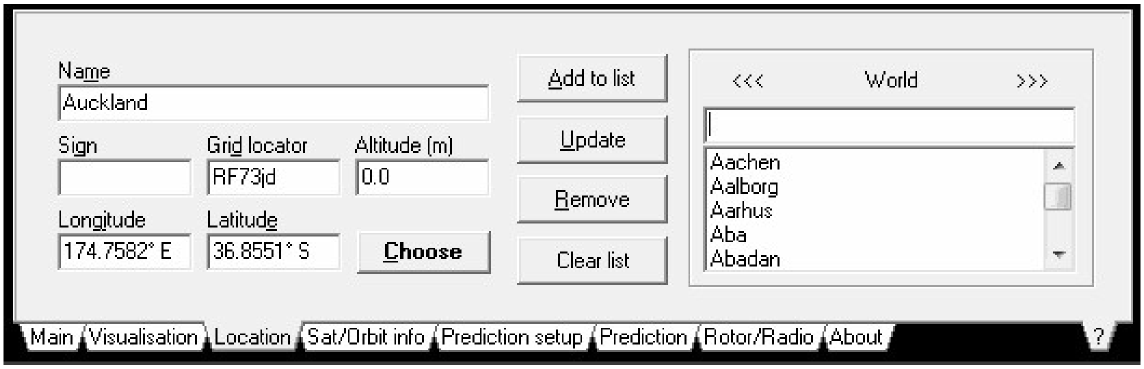
9. At the bottom of the Setup.cfg text file, add these two lines.

```
[Drivers]
SDRSharp=SDRSharp.exe
```

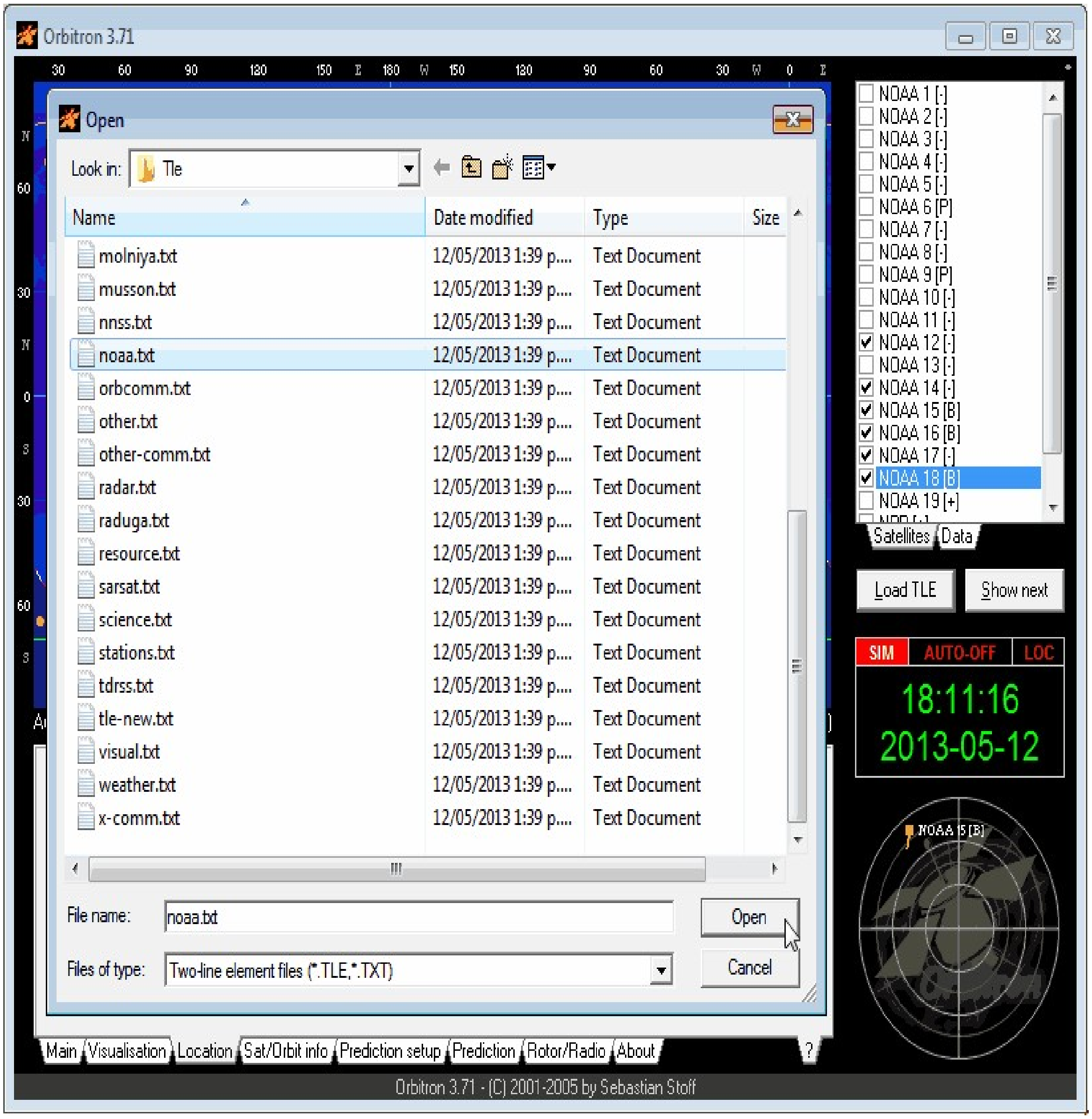


```
Elv=0
On=1
Snd=1
Wav=Data\Snd_01.wav

[Passes]
PeriodM=3
PeriodK=2
SatIllum=1
MaxSunX=1
MaxSun=-5
MinSat=10
IrFlareMag=30
Brief=0
MinMagX=0
MinMagnitude=30
IgnoreMagIfUnknown=0
Calculation=1
Mode=0

[Drivers]
SDRSharp=SDRSharp.exe
```

10. Now open Orbitron and set your home location, by clicking the location tab on the bottom. You can select your city on the right side if you don't know your exact longitude and latitude.


Name: Auckland
Sign:
Grid locator: RF73jd
Altitude (m): 0.0
Longitude: 174.7582° E
Latitude: 36.8551° S

11. Next click on Load TLE and load the noaa.txt file, or the file for whatever other satellite you are interested in tracking.



Look in: Tle

Name	Date modified	Type	Size
molniya.txt	12/05/2013 1:39 p....	Text Document	
musson.txt	12/05/2013 1:39 p....	Text Document	
nnss.txt	12/05/2013 1:39 p....	Text Document	
noaa.txt	12/05/2013 1:39 p....	Text Document	
orbcomm.txt	12/05/2013 1:39 p....	Text Document	
other.txt	12/05/2013 1:39 p....	Text Document	
other-comm.txt	12/05/2013 1:39 p....	Text Document	
radar.txt	12/05/2013 1:39 p....	Text Document	
raduga.txt	12/05/2013 1:39 p....	Text Document	
resource.txt	12/05/2013 1:39 p....	Text Document	
sarsat.txt	12/05/2013 1:39 p....	Text Document	
science.txt	12/05/2013 1:39 p....	Text Document	
stations.txt	12/05/2013 1:39 p....	Text Document	
tdrss.txt	12/05/2013 1:39 p....	Text Document	
tle-new.txt	12/05/2013 1:39 p....	Text Document	
visual.txt	12/05/2013 1:39 p....	Text Document	
weather.txt	12/05/2013 1:39 p....	Text Document	
x-comm.txt	12/05/2013 1:39 p....	Text Document	

File name: noaa.txt

Files of type: Two-line element files (*.TLE,*.TXT)

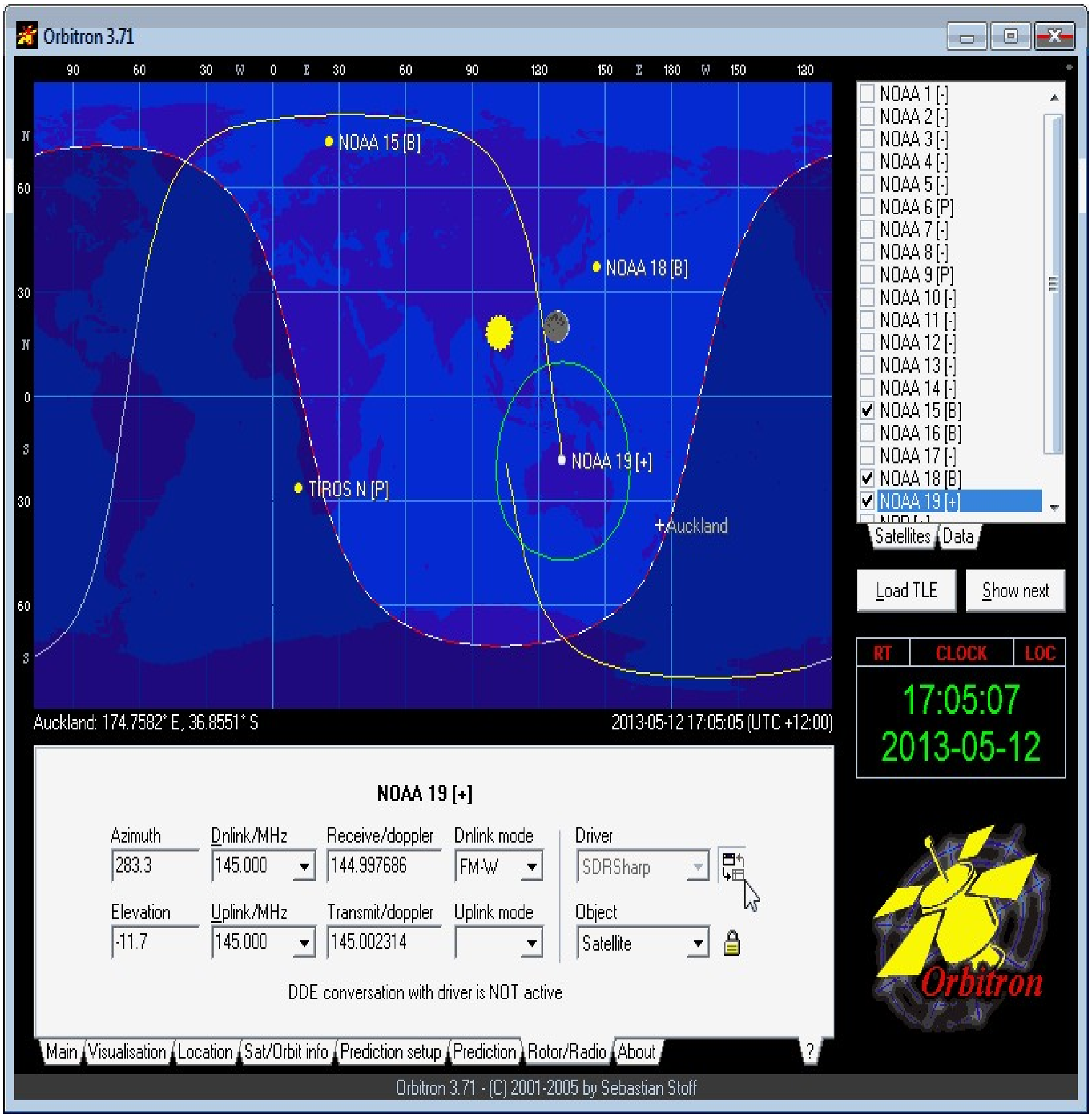
Open

Cancel

□ NOAA 1 [-]
□ NOAA 2 [-]
□ NOAA 3 [-]
□ NOAA 4 [-]
□ NOAA 5 [-]
□ NOAA 6 [P]
□ NOAA 7 [-]
□ NOAA 8 [-]
□ NOAA 9 [P]
□ NOAA 10 [-]
□ NOAA 11 [-]
☑ NOAA 12 [-]
□ NOAA 13 [-]
☑ NOAA 14 [-]
☑ NOAA 15 [B]
☑ NOAA 16 [B]
☑ NOAA 17 [-]
☑ NOAA 18 [B]
□ NOAA 19 [+]

18:11:16
2013-05-12

12. For weather satellite images we are interested in NOAA satellites 15, 18 and 19, as they are the only ones working, so place a check next to those. Double clicking on a satellite name will select it and show it in the map window.



Auckland: 174.7582° E, 36.8551° S

2013-05-12 17:05:05 (UTC +12:00)

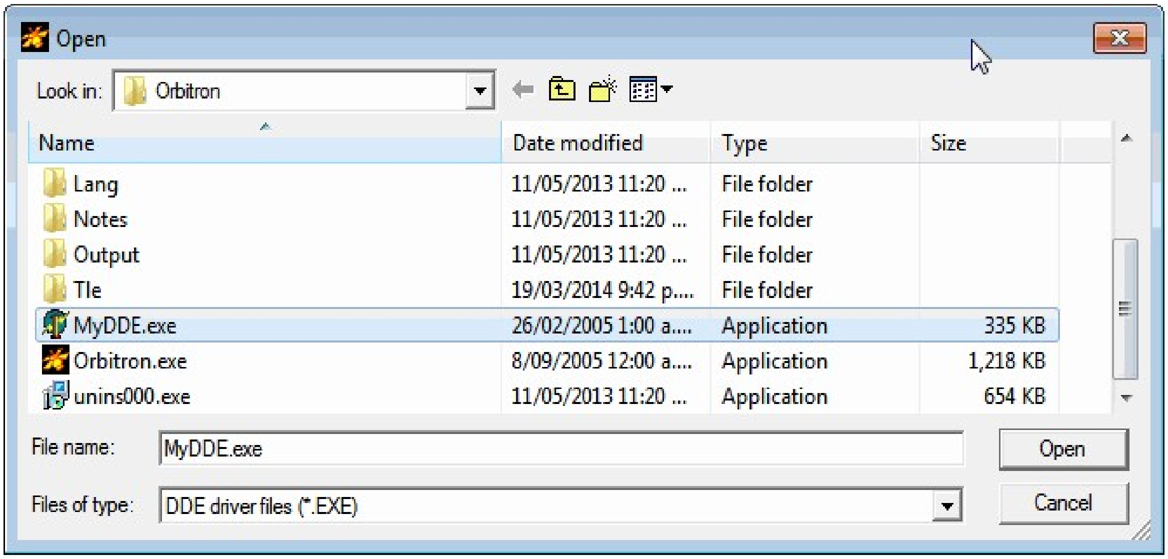
NOAA 19 [+]

Azimuth	Dnlink/MHz	Receive/doppler	Dnlink mode	Driver
283.3	145.000	144.997686	FM-W	SDRSharp
Elevation	Uplink/MHz	Transmit/doppler	Uplink mode	Object
-11.7	145.000	145.002314		Satellite

DDE conversation with driver is NOT active

Orbitron 3.71 - (C) 2001-2005 by Sebastian Stoff

13. Now go to the Rotor/Radio tab, and set the "Dnlink" mode to FM-W and the Driver to SDRSharp. Click the icon with two windows next to the Driver drop down box and make sure it is pressed in. Upon pressing this button for the first time you may be asked to install a Driver. Click Yes and then browse to the Orbitron folder and select the MyDDE.exe file you copied over earlier.



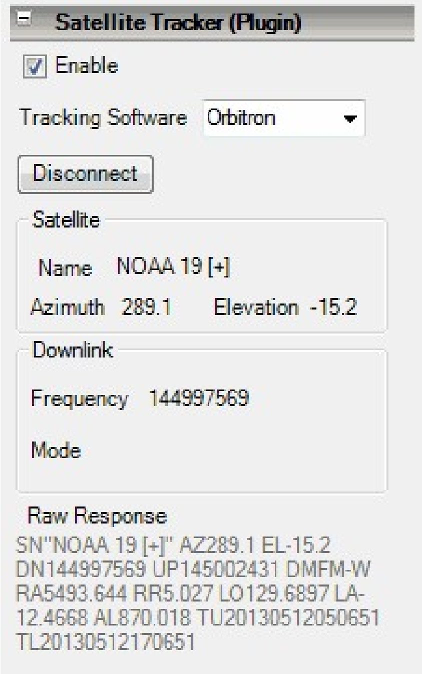
Look in: Orbitron

Name	Date modified	Type	Size
Lang	11/05/2013 11:20 ...	File folder	
Notes	11/05/2013 11:20 ...	File folder	
Output	11/05/2013 11:20 ...	File folder	
Tle	19/03/2014 9:42 p....	File folder	
MyDDE.exe	26/02/2005 1:00 a....	Application	335 KB
Orbitron.exe	8/09/2005 12:00 a....	Application	1,218 KB
unins000.exe	11/05/2013 11:20 ...	Application	654 KB

File name: MyDDE.exe

Files of type: DDE driver files (*.EXE)

14. Open SDR#, press Play and then head to the Orbitron Plugin. Put a check next to Enable, set the Tracking Software option to Orbitron and then click connect. Now double clicking on a satellite name in Orbitron should automatically set the correct doppler corrected frequency in SDRSharp.



☑ Enable

Tracking Software: Orbitron

Satellite

Name: NOAA 19 [+]

Azimuth: 289.1 Elevation: -15.2

Downlink

Frequency: 144997569

Mode:

Raw Response

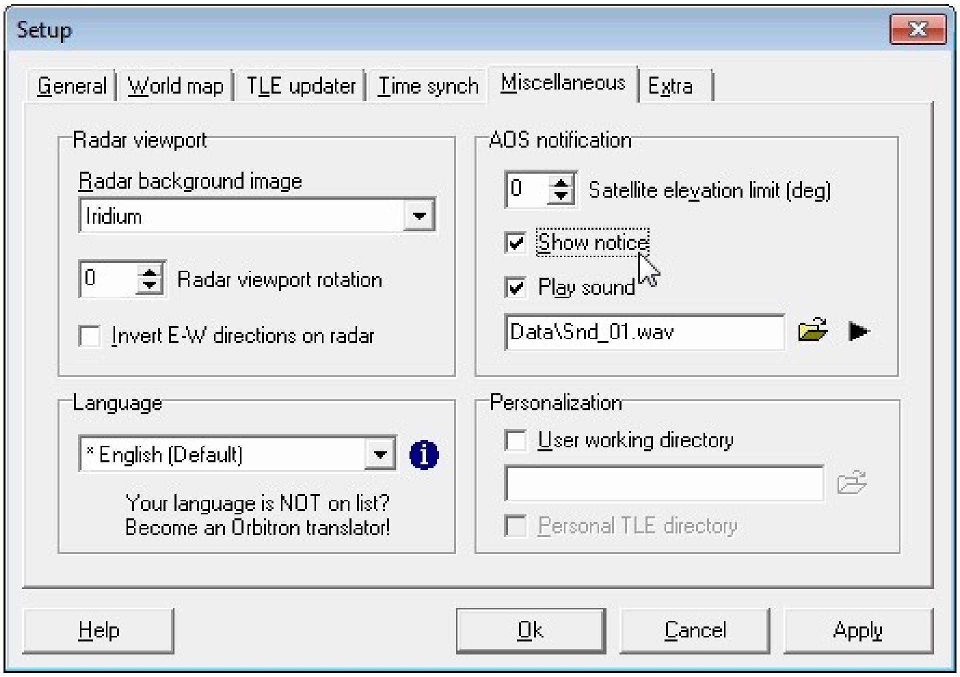
```
SN"NOAA 19 [+]" AZ289.1 EL-15.2
DN144997569 UP145002431 DMFM-W
RA5493.644 RR5.027 LO129.6897 LA-
12.4668 AL870.018 TU20130512050651
TL20130512170651
```

15. SDR# should now snap to the correct frequency and adjust for the doppler effect automatically. You will still need to manually set the correct filter bandwidth and mode to WFM. Also, since the RTL-SDR is not

frequency accurate, you will need to adjust the PPM correction in the SDR# configure box to ensure the signal is centred.

16. If you want Orbitron to make SDR# automatically tune to a satellite as soon as it comes into range, go to the Main Tab in Orbitron and click on the Setup button ⚒ (looks like a crossed hammer and spanner).

17. Go to the Miscellaneous tab and ensure that AOS Notification: Show Notice is selected, with the elevation set to 0. (Increase the elevation if you only want to start tuning to the frequency when the satellite is higher in the sky and thus gives you better reception).



Radar viewport

Radar background image: Iridium

0 Radar viewport rotation

□ Invert E-W directions on radar

AOS notification

0 Satellite elevation limit (deg)

☑ Show notice

☑ Play sound

Data\Snd_01.wav

Language

* English (Default)

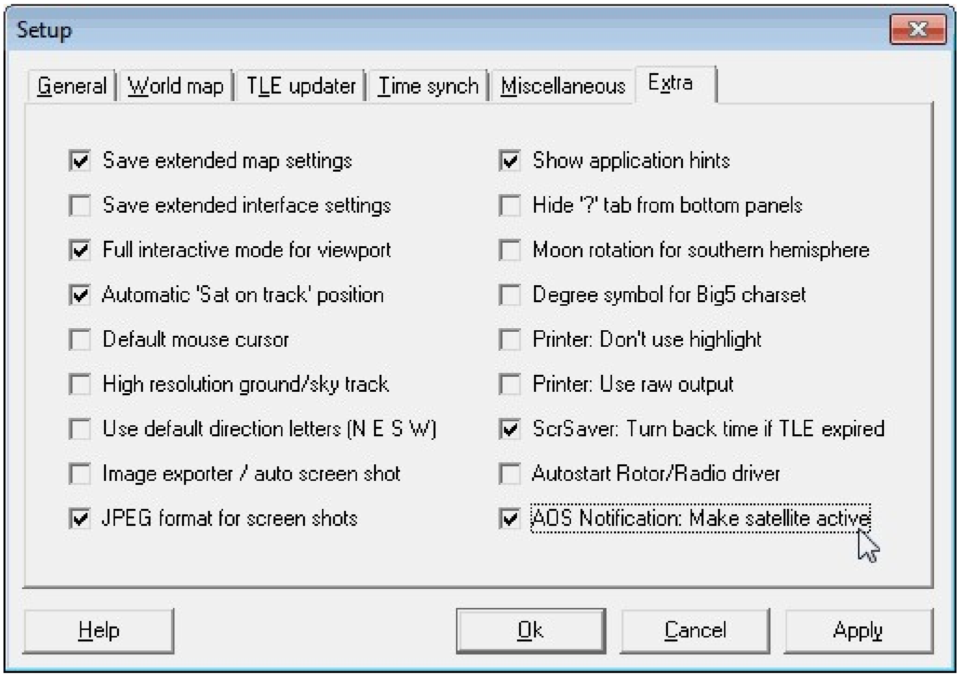
Your language is NOT on list?
Become an Orbitron translator!

Personalization

□ User working directory

□ Personal TLE directory

18. Go to the Extra tab and ensure that AOS Notification: Make satellite active is checked.



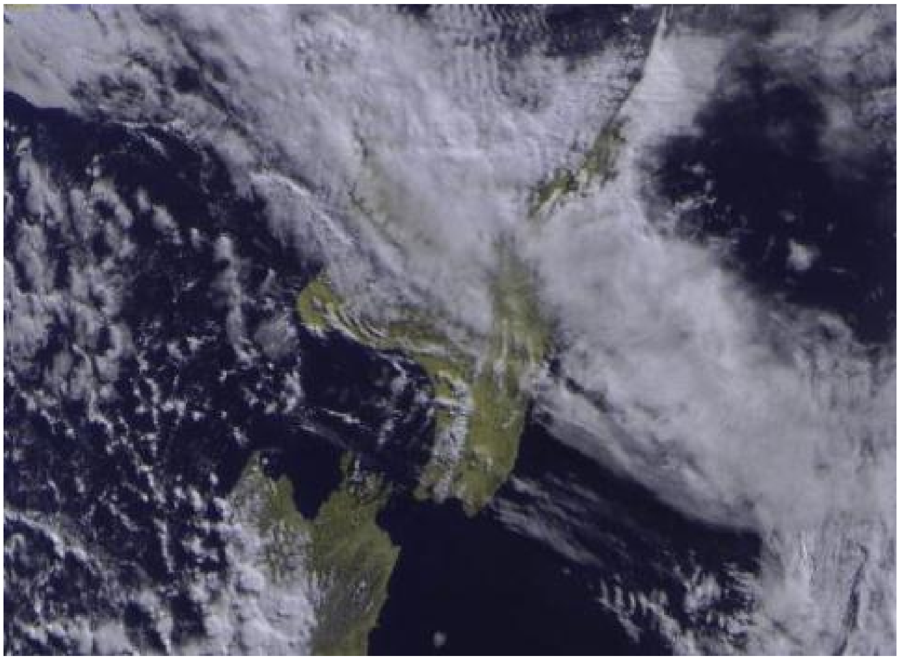
☑ Save extended map settings
□ Save extended interface settings
☑ Full interactive mode for viewport
☑ Automatic 'Sat on track' position
□ Default mouse cursor
□ High resolution ground/sky track
□ Use default direction letters (N E S W)
□ Image exporter / auto screen shot
☑ JPEG format for screen shots

☑ Show application hints
□ Hide '?' tab from bottom panels
□ Moon rotation for southern hemisphere
□ Degree symbol for Big5 charset
□ Printer: Don't use highlight
□ Printer: Use raw output
☑ ScrSaver: Turn back time if TLE expired
□ Autostart Rotor/Radio driver
☑ AOS Notification: Make satellite active

19. Finally, if desired WXtoImg can be made to automatically output a live webpage of the latest weather satellite images. This option can be found in WXtoImg under **Options->Auto Processing Options->Web Page Settings**.

19.

METEOR-M RUSSIAN LRPT WEATHER SATELLITE GUIDE



INTRODUCTION TO METEOR-M

The Meteor-M N2 is a polar orbiting Russian weather satellite that was launched on July 8, 2014. Its main missions are weather forecasting, climate change monitoring, sea water monitoring/forecasting and space weather analysis/prediction.

The satellite is currently active with a Low Resolution Picture Transmission (LRPT) signal which broadcasts live weather satellite images, similar to the APT images produced by the NOAA satellites. LRPT images are however much better as they are transmitted as a digital signal with an image resolution 12 times greater than the aging analogue NOAA APT signals. Compare the image of New Zealand shown at the beginning of this section to the one shown in the NOAA APT tutorial section above this one to see the difference in resolution. Some example Meteor weather images can be found at http://meteor.robonuka.ru/septembers-gallery/ and the satellite can be tracked in Orbitron or online at http://www.satview.org/?sat_id=40069U. During the day the weather image is generated using visible light and at night it is generated using infrared.

The RTL-SDR and other SDRs like the Funcube along with some free software can be used to receive and decode these images. LRPT images from the Meteor-M N2 are transmitted with a right hand circularly polarized signal at around 137.1 MHz, so any satellite antenna such as a turnstile or quadrifilar helix which are commonly used with the NOAA weather satellites can be used. See the antenna guide section for more information. As LRPT is a digital signal, reception strength must be good, so ensure you are using a good satellite antenna placed up high with a unobstructed view of the sky. In some poor reception situations an LNA at the antenna and some bandpass or bandstop filters to remove interference may also be needed.

RECEIVING METEOSAT WEATHER SATELLITE IMAGES: WINDOWS TUTORIAL

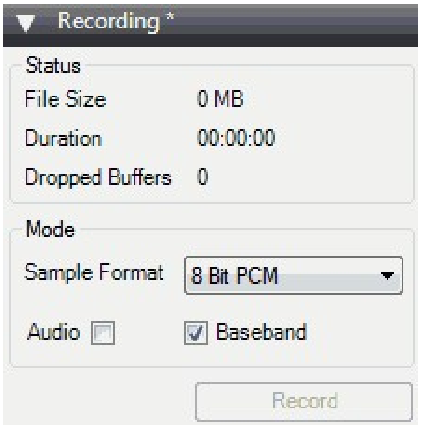
In order to receive these images on Windows you will first need to download and install the following software.

- Audacity. A free audio editing software package. Download from http://audacity.sourceforge.net/.
- LRPT BPSK/QPSK DeWAVulator. Download from https://www.dropbox.com/s/qq1fjyitpa3j14o/software.zip (Mirror: http://bit.ly/Yyicop). Extract the LRPT.exe file, the other two files in the zip archive are not required.
- LRPTOfflineDecoder. Download the latest version from http://meteor.robonuka.ru/soft/ (Mirror of 2014.09.01.0006 version: http://bit.ly/1lUKS4Y).

Since this satellite is quite new, the decoding process is not yet as refined as with the NOAA APT satellites. The steps below outline the process.

1. Open SDR# and in the configure menu change the sample rate of the RTL-SDR to 0.9 Msps.
2. When the Meteor-M2 is about to pass over (use Orbitron to determine when), tune to 137.1 MHz in SDR#. Make sure “Correct IQ” is set and if you are using an E4000 or FC0012/13 based dongle ensure that offset tuning is enabled.
3. Once the signal appears, first make sure it is centred as much as possible in the waterfall. Then record an 8-bit PCM baseband IQ file using the SDR#

Recording plugin, which is included in the standard SDR# install by default.



▼ Recording *

Status

File Size 0 MB

Duration 00:00:00

Dropped Buffers 0

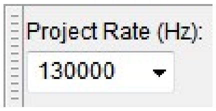
Mode

Sample Format 8 Bit PCM

Audio □ ☑ Baseband

4. After the satellite has passed and the signal has died down, stop the recording and then open the resulting .wav file in Audacity (Remember that SDR# saves its recordings in the same folder as the executable file).

5. Change the sample rate to 130000 Hz in the low left hand corner of Audacity.

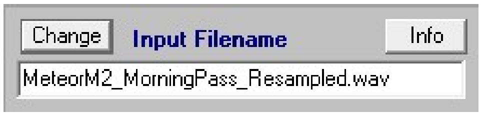


Project Rate (Hz):

130000

6. Export the resampled file as a new wav file by going to **File->Export**. Then choose the format as WAV (Microsoft) signed 16 bit PCM. Save the new wav file to a convenient folder.

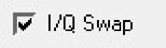
7. Now open the LRPT.exe program and load in the resampled .wav file that you have just exported from Audacity by clicking the change button in the Input Filename box on the left of the program.



Input Filename

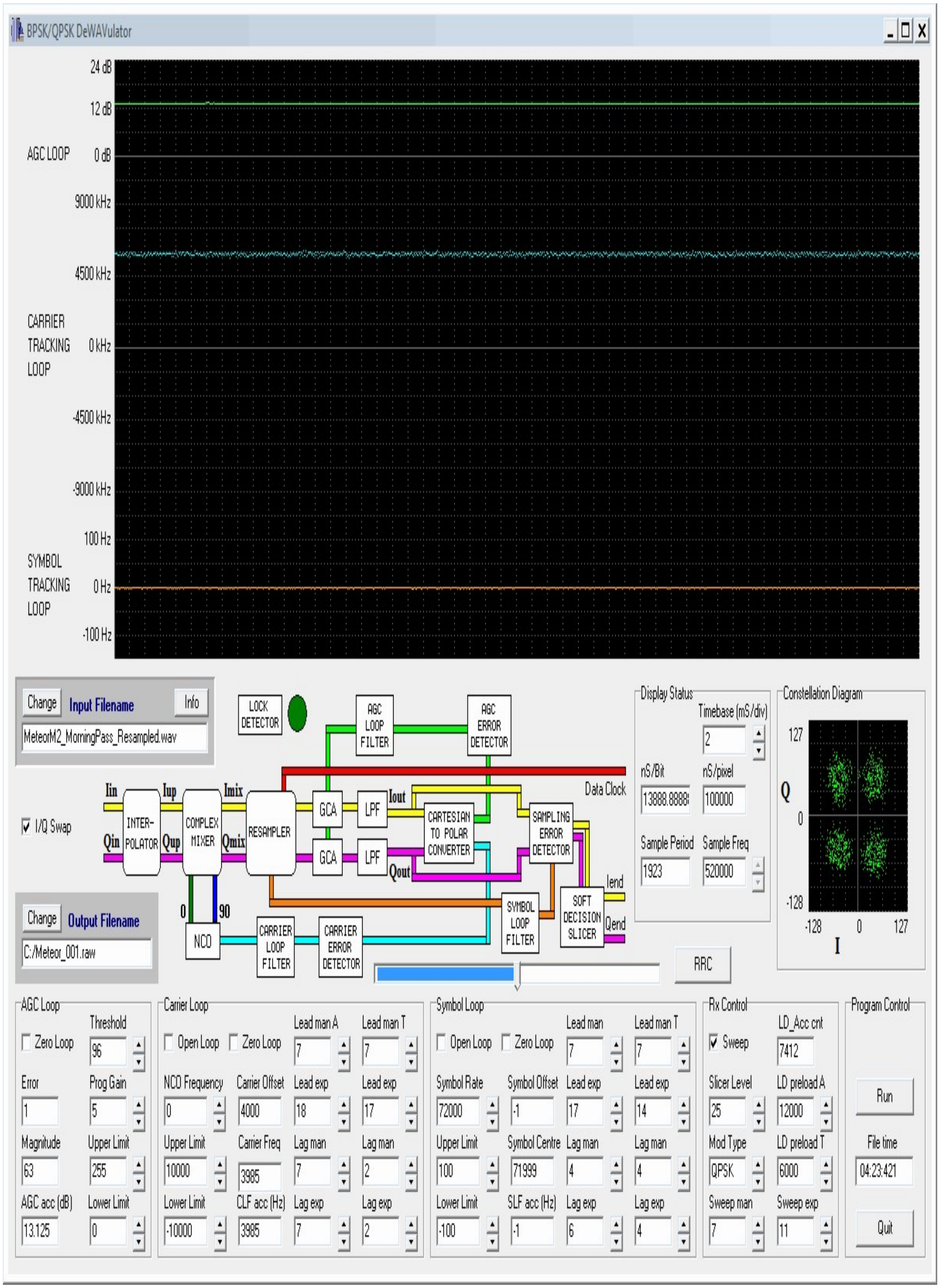
MeteorM2_MorningPass_Resampled.wav

8. Ensure that Swap IQ is selected.



☑ I/Q Swap

9. Now manually move the slider in the middle of the window to the point at which the signal was the strongest, which will probably be somewhere near the centre. Press the start button in the lower right hand corner.



Input Filename: MeteorM2_MorningPass_Resampled.wav

☑ I/Q Swap

Output Filename: C:/Meteor_001.raw

Display Status
Timebase (mS/div): 2
nS/Bit: 13888.8888
nS/pixel: 100000
Sample Period: 1923
Sample Freq: 520000

AGC Loop
□ Zero Loop
Threshold: 96
Error: 1
Prog Gain: 5
Magnitude: 63
Upper Limit: 255
AGC acc (dB): 13.125
Lower Limit: 0

Carrier Loop
□ Open Loop □ Zero Loop
Lead man A: 7
Lead man T: 7
NCO Frequency: 0
Carrier Offset: 4000
Lead exp: 18
Lead exp: 17
Upper Limit: 10000
Carrier Freq: 3985
Lag man: 7
Lag man: 2
Lower Limit: -10000
CLF acc (Hz): 3985
Lag exp: 7
Lag exp: 2

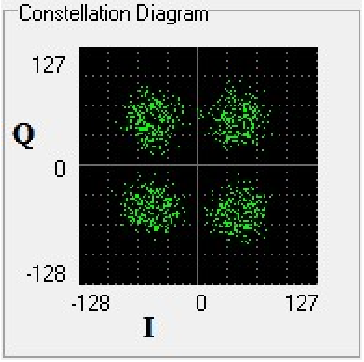
Symbol Loop
□ Open Loop □ Zero Loop
Lead man: 7
Lead man T: 7
Symbol Rate: 72000
Symbol Offset: -1
Lead exp: 17
Lead exp: 14
Upper Limit: 100
Symbol Centre: 71999
Lag man: 4
Lag man: 4
Lower Limit: -100
SLF acc (Hz): -1
Lag exp: 6
Lag exp: 4

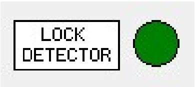
Rx Control
☑ Sweep
LD_Acc cnt: 7412
Slicer Level: 25
LD preload A: 12000
Mod Type: QPSK
LD preload T: 6000
Sweep man: 7
Sweep exp: 11

Program Control
File time: 04:23:421

10. Check the constellation diagram on the right side of the program. If it shows four clusters of dots, then the signal was received properly. If you don’t see it, try moving the slider around until you find a spot where the four clusters show. The Lock Detector light should also turn green.

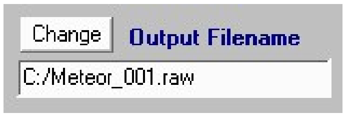
11. If it looks like an X or a square, then try changing the symbol rate from 72000 to 80000. If you see the four clusters of dots then, then unfortunately the satellite was in 80K mode which can not currently be decoded.





12. If you did see the four clusters of dots in the constellation diagram at some point then you are okay to proceed. Bring the slider back to the start position.

13. Choose the save location of the decoded RAW output file by clicking the Change button next to the text Output Filename on the left of the program. By default the output file will be saved as C:\Meteor_001.raw



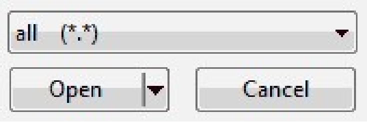
Output Filename

C:/Meteor_001.raw

14. Start the decoding process by clicking on the Start button in the lower right corner of the program. Keep an eye on the slider and wait until it has reached the end, then press stop. The program will not stop by itself, so you must remember to click the stop button. After stopping quit the program using the quit button.

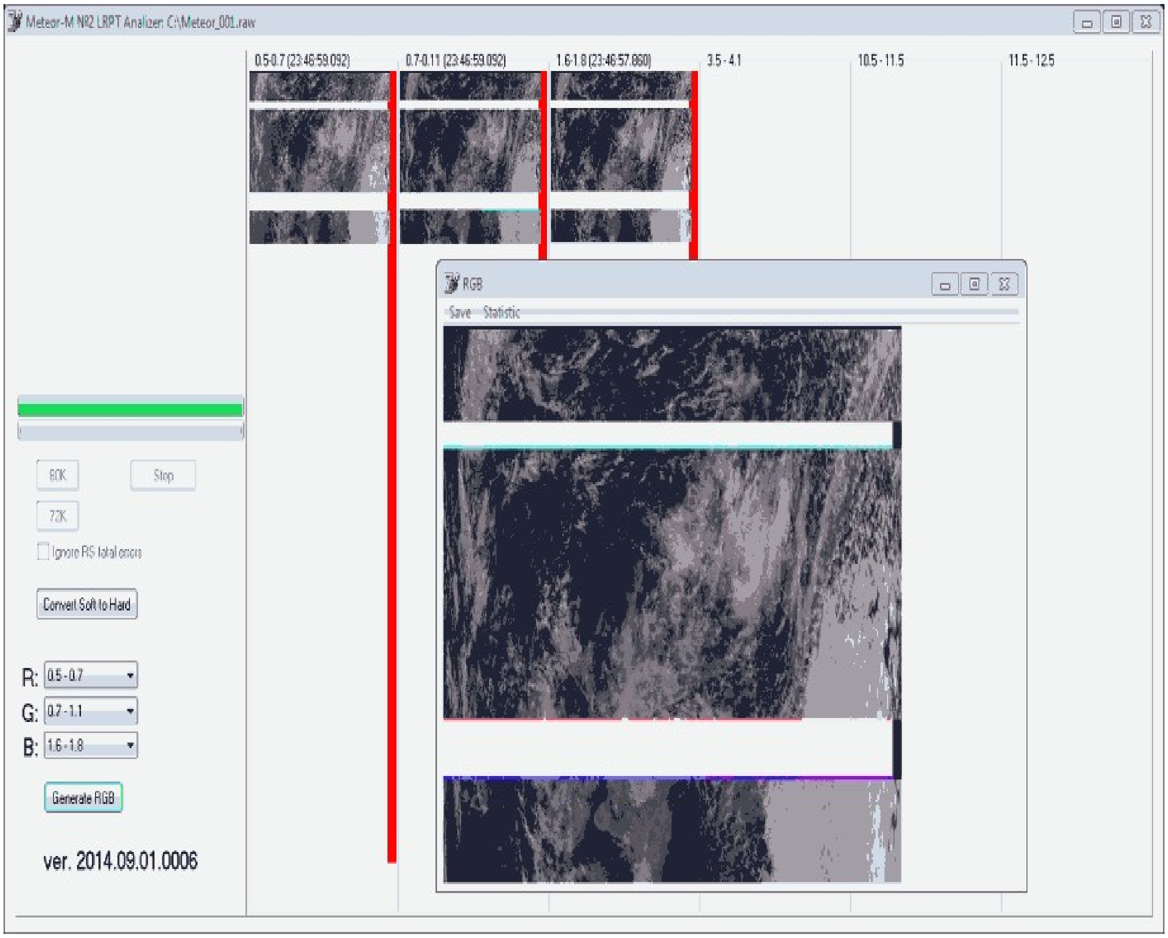
15. Now to actually extract the image from the processed file we need to use the LRPToffLineDecode.exe decoder software. Open the decoded RAW file

from LRPT.exe in LRPToffLineDecode.exe by clicking on the 72K button (or possibly 80K button in the future when this option is available). When opening it you will need to change the file type filter to All (*.*) in order to be able to see the .raw file.


all (*.*)

15. Once opened the image will automatically begin generating. Generating may take around 10 to 15 minutes.

16. After processing is complete, click on the “Generate RGB” button to see the final image. Note that if you notice that one of the images is not in the first three columns, you will need to adjust the R,G,B drop down boxes to choose another column. If the IR sensor was active during daytime the third image might be in the last column. To get a colour image G and B should both be selected as 0.7-0.11.


Meteor-M N2 LRPT Analizer: C:\Meteor_001.raw

R: 0.5 - 0.7

G: 0.7 - 1.1

B: 1.6 - 1.8

ver. 2014.09.01.0006

Note that LRPT.exe was actually designed for the Meteor-M1 satellite which is now inactive. Because of this there might be some artefacts that appear in the final image like black bands.

RECEIVING METEOSAT WEATHER SATELLITE IMAGES IN REAL TIME: WINDOWS

Thanks to new software developments, it has recently become possible to decode Meteor M2 LRPT images in real time. This method is still a little difficult to set up, but as well as getting real time results, it also results in better satellite locks and thus better images. To use real time decoding you will need the satellte tracking software Orbitron, the QPSK decoder plugin for SDR#, a DDE orbitron interface plugin for SDR#, and a special version of LRPTDecoder. This method will also record a backup I/Q file of the LRPT signal, which can then be run through the offline method (shown above), just in case the real time method does not work. Note that this method is fairly new and may still be developing and changing fast.

Note that this tutorial is based heavily on Alex's (happysat) original tutorial which can be found on our blog at http://www.rtl-sdr.com/rtl-sdr-tutorial-decoding-meteor-m2-weather-satellite-images-in-real-time-with-an-rtl-sdr/. If you need more help, want to learn more about real time decoding, or have trouble with this tutorial, please consult his tutorial as well.

DOWNLOAD THE SOFTWARE

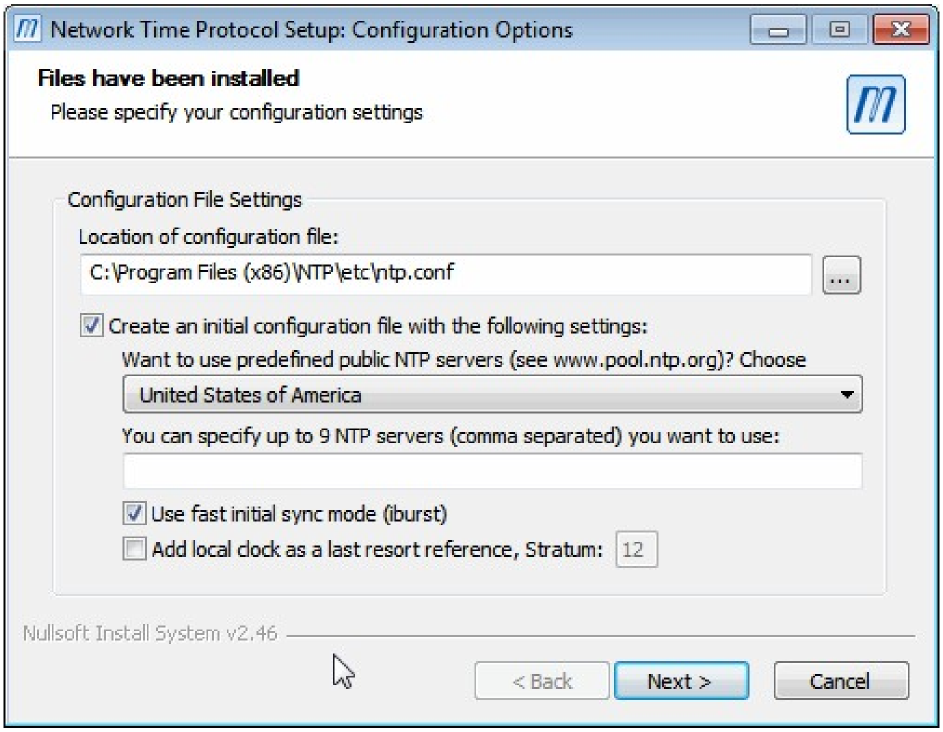
1. First download the QPSK plugin compatible LRPTdecoder software from here http://meteor.robonuka.ru/for-experts/amigos/downloads/amigos-compatable-lrptofflinedicoder/ and extract it to the folder C:\AMIGOS\. Note that although the file name is still called LRPToffLineDecoder.exe, it is actually an online decoder.

2. Next download the DDE orbitron interface plugin and the QPSK demodulator from http://www.rtl-sdr.ru/page/komplekt-plaginov-dlja-priema-sputnikov. Also optionally download the IF recorder plugin from http://www.rtl-sdr.ru/page/dobavlen-novyj-plagin-if-recorder if you want to be able to make a backup recording of the LRPT signal, just in case the online method fails for some reason. Note that these pages are in Russian, so use Google translate. The download links are at the bottom of the page. Extract the contents of these plugin zip files to your SDR# installation folder and add the magicline to the plugins.xml file. If you need help with installing plugins, see the installing SDR# plugins section.

SYNCHRONISING WINDOWS TIME

To improve signal locking with Meteor M2 it is critical that the Orbitron doppler correction be accurate, and for that Windows time needs to be accurate. This is because Orbitron uses the current time to calculate where the satellite should be in space, and thus what doppler correction is needed for your location. As explained in the NOAA tutorial, Orbitron contains its own method to update PC time from the NTP servers. However, this only activates once when you click the update button. Time correction isn't as critical for NOAA satellites as it is for Meteor M2, because NOAA satellites are analog and don't need to lock onto a digital signal. We suggest you download an NTP automatic update program which will automatically syncronise the PC clock with NTP time every few seconds. The software can be downloaded from https://www.meinbergglobal.com/english/sw/ntp.htm#ntp_stable, choose the download for NTP for Windows XP and newer, with IPv6 support. Note that before running the installer make sure that Orbitron is closed, as it's own NTP update procedure may prevent the setup from completing successfully. Below we show how the setup options should be configured.

1. During the setup make sure you choose the closest location to you where it says "Want to use predefined public NTP servers (see www.pool.ntp.org)? Choose.



Network Time Protocol Setup: Configuration Options

Files have been installed

Please specify your configuration settings

Configuration File Settings

Location of configuration file:

C:\Program Files (x86)\NTP\etc\ntp.conf

☑ Create an initial configuration file with the following settings:

Want to use predefined public NTP servers (see www.pool.ntp.org)? Choose

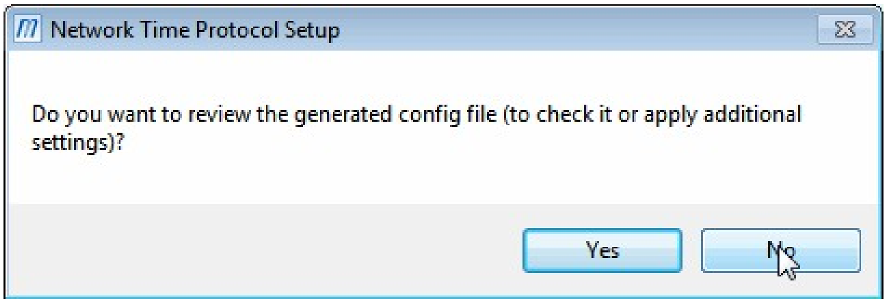
United States of America

You can specify up to 9 NTP servers (comma separated) you want to use:

☑ Use fast initial sync mode (iburst)

☐ Add local clock as a last resort reference, Stratum: 12

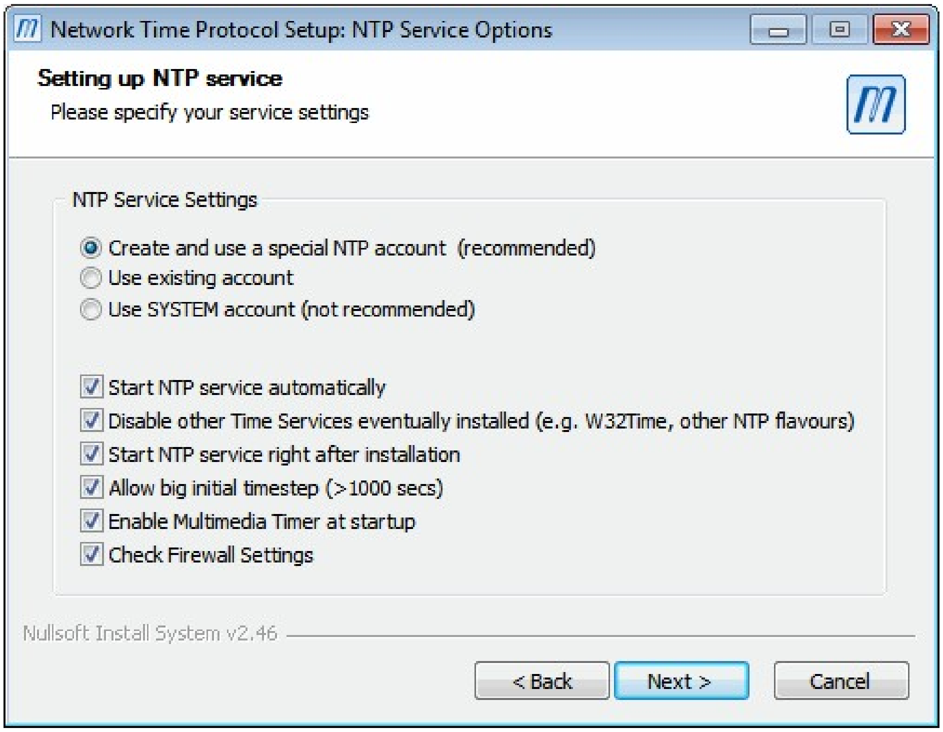
2. You can simply click no at the next screen.



Network Time Protocol Setup

Do you want to review the generated config file (to check it or apply additional settings)?

3. The next screen can also be left as default as shown in the image below.



Network Time Protocol Setup: NTP Service Options

Setting up NTP service

Please specify your service settings

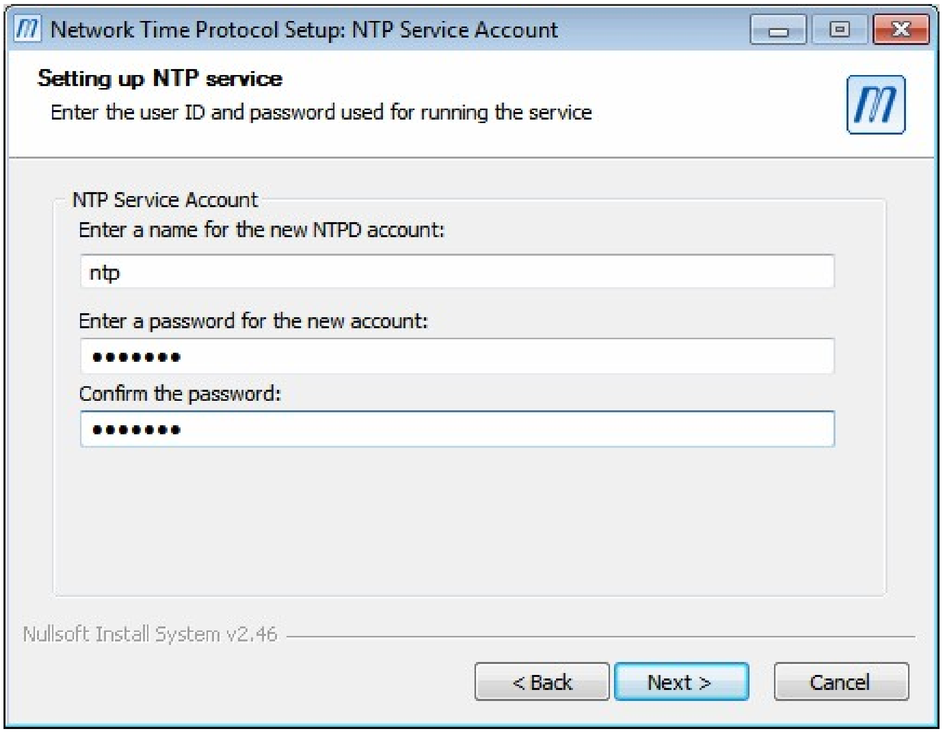
NTP Service Settings

● Create and use a special NTP account (recommended)
○ Use existing account
○ Use SYSTEM account (not recommended)

☑ Start NTP service automatically
☑ Disable other Time Services eventually installed (e.g. W32Time, other NTP flavours)
☑ Start NTP service right after installation
☑ Allow big initial timestep (>1000 secs)
☑ Enable Multimedia Timer at startup
☑ Check Firewall Settings

Nullsoft Install System v2.46

4. In the next screen you will be required to create an NTP account on your PC. Simply enter a password, and remember it in case you need to alter some settings in the future.



Network Time Protocol Setup: NTP Service Account

Setting up NTP service

Enter the user ID and password used for running the service

NTP Service Account

Enter a name for the new NTPD account: ntp

Enter a password for the new account: •••••••

Confirm the password: •••••••

5. Click next and the NTP time service should install and be automatically started. If you receive an errror about the NTP time service not being able to be started, then ensure you have closed Orbitron and rerun the installer. After installing NTP you may also need to manually turn off Windows time syncronisation which does a very poor job compared to NTP. In control panel go to Date and Time settings, and ensure that Windows syncronisation is turned off.

If you need further help with the setup of NTP see a more in depth guide at http://www.satsignal.eu/ntp/setup.html.

ORBITRON AND SDR# SETUP

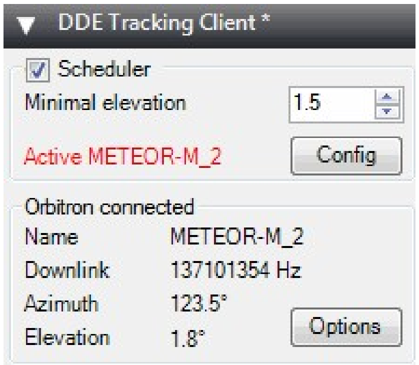
Below we explain the steps needed to set up Orbitron and SDR# to receive and decode Meteor M2 in real time. We also present some optional steps that will allow you to set up an automatic scheduler so that your system will automatically tune to the Meteor M2 frequency and open the decoders when the satellite begins to pass overhead.

1. Go to C:\Orbitron\Config\setup.cfg and open this file with a text editor like Notepad. At the end of the file add the following lines, making sure to edit

the file path to the exact location of your own SDR# installation folder and the SDRSharpDriverDDE file which is included with the SDR# DDE plugin.

```
[Drivers]
SDRSharp=C:\YOUR_SDRSHARP_INSTALL_FODLER\SDRSharp\SDRSharpDriverDDE.exe
```

2. Now open Orbitron and update your TLE files as shown in step 5 of the Orbitron guide above.
3. Set the refresh interval to 1 second by using the drop down menu in the Main Orbitron tab. By default it is set to 5 seconds. 1 ▾
4. Select Meteor M2 as the satellite to track by loading the weather.txt TLE file and choosing Meteor-M2 in the satellite selection area on the right.
5. (Optional if you want scheduling) Set up Orbitron to automatically track and notify when the satellite comes into range as shown in steps 17 - 19 in the Orbitron tutorial above. This will allow Orbitron to begin automatically tracking when the satellite appears on your horizon, above a certain elevationm and to notify the DDE plugin scheduler in SDR#. Make sure you set the AOS notification elevation limit to 0.
6. Now open SDR# and press Play.
7. Go to the Rotor/Radio tab in Orbitron.
8. Set the Dnlink/MHz frequency to 137.1 MHz and the Dnlink mode to FM-W.
9. Choose the driver as SDRSharp and then click the run driver icon to make Orbitron connect to SDR#.
10. Now in SDR# go to the DDE tracking Client plugin. You should see the status showing that Orbitron is connected.



▼ DDE Tracking Client *

☑ Scheduler

Minimal elevation 1.5

Active METEOR-M_2 Config

Orbitron connected

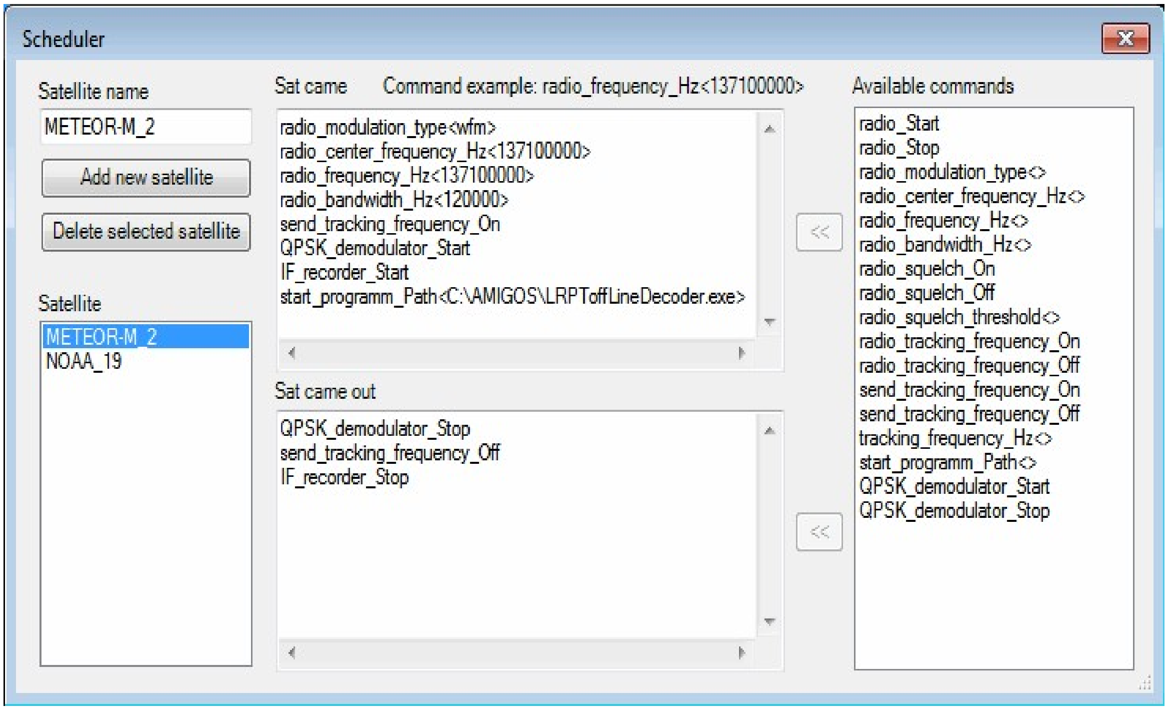
Name METEOR-M_2

Downlink 137101354 Hz

Azimuth 123.5°

Elevation 1.8° Options

11. (Optional if you want scheduling) The DDE plugin can be made to automatically activate and start the LRPT decoders when the satellite comes overhead by using the scheduler which can be activated by checking the Scheduler box. To use the scheduler you first need to create a settings file that will tell the plugin what settings to set in SDR# and what decoding programs to start when the satellite passes over head. To do this for Meteor M2 we can simply download the example settings file from the QPSK and DDE Tracker plugin download page, and place it into the SDR# folder. The file can be downloaded from http://www.rtl-sdr.ru/page/komplekt-plaginov-dlja-priema-sputnikov, it is the last file called ddeschedule.zip. Note that you may still need to adjust the path to the LRPTofflinedecoder.exe file. Also you will need to set the minimal elevation. When the satellite reaches this elevation threshold (higher in the sky is better because you get better reception usually), then the decoders will start. You'll want to set this to an elevation at which you start getting a strong enough signal to actually be able to decode the image.



Sat came:

```
radio_modulation_type<wfm>
radio_center_frequency_Hz<137100000>
radio_frequency_Hz<137100000>
radio_bandwidth_Hz<120000>
send_tracking_frequency_On
QPSK_demodulator_Start
IF_recorder_Start
start_programm_Path<C:\AMIGOS\LRPToffLineDecoder.exe>
```

Sat came out:

```
QPSK_demodulator_Stop
send_tracking_frequency_Off
IF_recorder_Stop
```

Available commands:

```
radio_Start
radio_Stop
radio_modulation_type<>
radio_center_frequency_Hz<>
radio_frequency_Hz<>
radio_bandwidth_Hz<>
radio_squelch_On
radio_squelch_Off
radio_squelch_threshold<>
radio_tracking_frequency_On
radio_tracking_frequency_Off
send_tracking_frequency_On
send_tracking_frequency_Off
tracking_frequency_Hz<>
start_programm_Path<>
QPSK_demodulator_Start
QPSK_demodulator_Stop
```

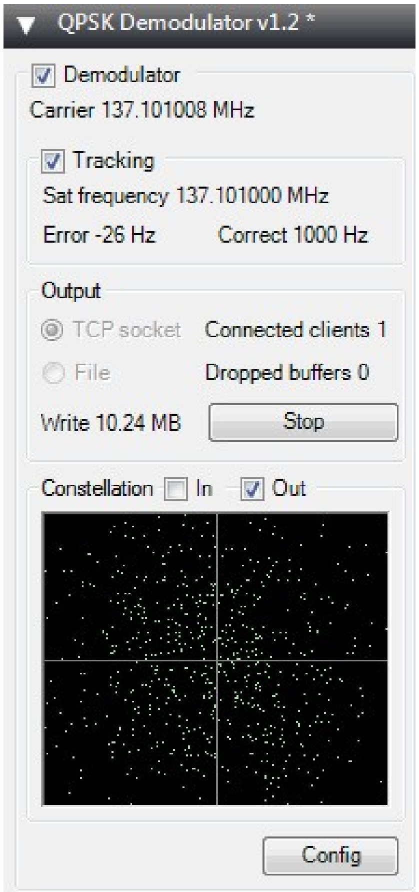
The "Sat came" box is for the commands that will be run when Orbitron notifies the DDE plugin that the satellite is overhead. The "Sat came out" boxis for when Orbitron notifies the DDR plugin that the satellite pass is over. Below we list the meanings of the commands specified for Meteor M2.

radio_modulation_type<wfm> - Sets the modulation mode in SDR# to WFM.
radio_center_frequency_Hz<137100000> - Sets the center frequency in SDR# to 137.1 MHz.
radio_frequency_Hz<137100000> - Sets the tuned frequency in SDR# to 137.1 MHz.
radio_bandwidth_Hz<120000> - Sets the WFM bandwidth to 120 kHz, wide enough to cover the LRPT signal.
send_tracking_frequency_On - Enables the tracking feature on the QPSK plugin which can correct for small doppler effects and improve lock.
QPSK_demodulator_Start - Starts the QPSK demodulator.
IF_recorder_Start - Starts the IF recorder. Note that this is optional. If you don't want to record IF files, then remove this as well as the IF_recorder_Stop command below.
start_program_Path<C:\AMIGOS\LRPToffLineDecoder.exe> - Starts the LRPT decoder. Change this to the location of the modified LRPTofflineDecoder if you don't place it in C:\AMIGOS. Note do not use a path with spaces in it.
QPSK_demodulator_Stop - Stops the QPSK demodulator.
send_tracking_frequency_Off - Stops the QPSK demodulator from doing doppler tracking.
IF_recorder_Stop - Stops the IF recording. Note that this is optional. If you don't want to record IF files, then remove this, as well as the IF_recorder_Start command above.

QPSK PLUGIN AND LRPTOFFLINEDECODER SETUP

The QPSK plugin communicates with the modified LRPToffLineDecoder via a local TCP connection.

1. In the QPSK plugin in SDR# make sure that TCP socket is selected under Output. (Note that if you have a slow PC, real time decoding may not work correctly. In this case you can still record a demodulated output file that can be put through LRPToffLineDecoder later for offline processing. To do this select the File option instead.)



QPSK Demodulator v1.2 *

☑ Demodulator

Carrier 137.101008 MHz

☑ Tracking

Sat frequency 137.101000 MHz

Error -26 Hz Correct 1000 Hz

Output

● TCP socket Connected clients 1

○ File Dropped buffers 0

Write 10.24 MB

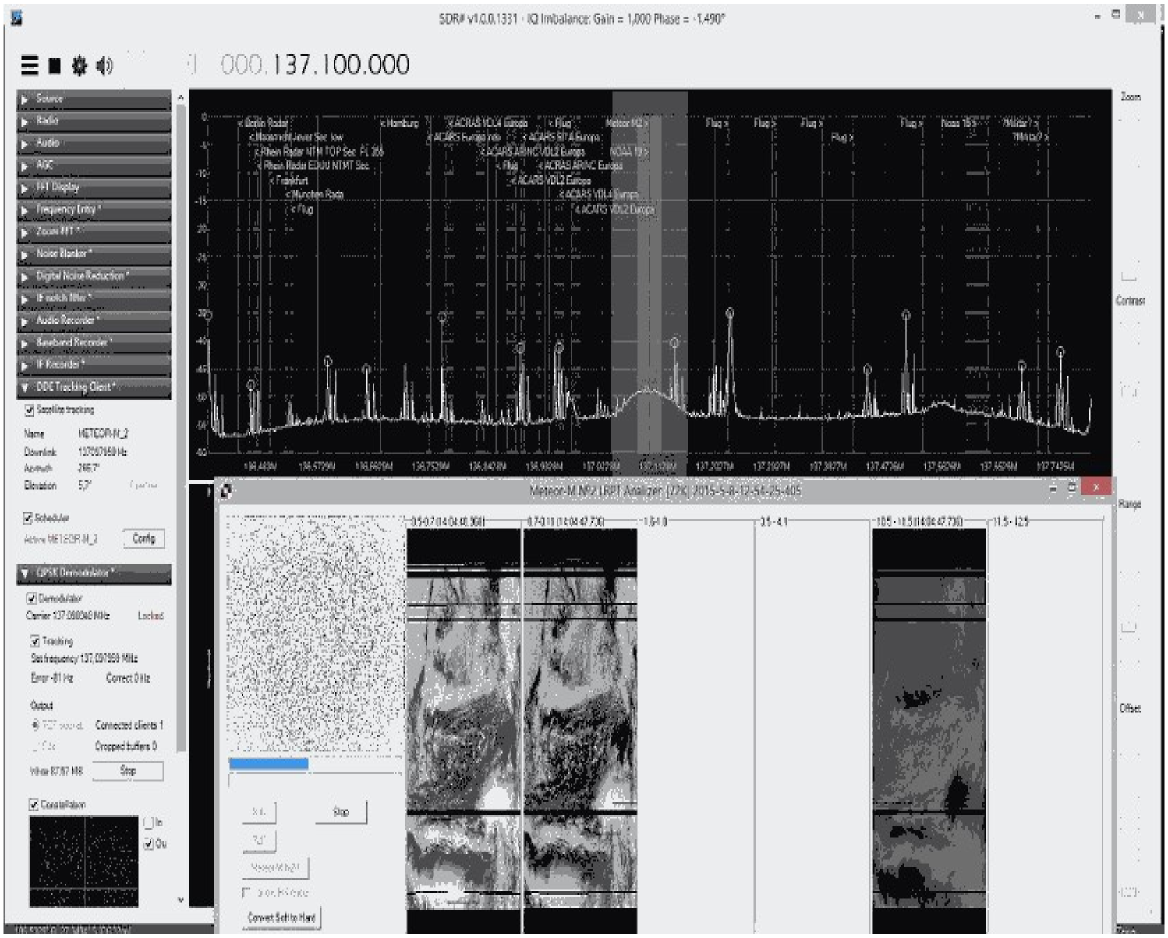
Constellation □ In ☑ Out

2. If you are not using the scheduler, then wait for Meteor M2 pass. Make sure the Tracking checkbox is checked. When it is time for the satellite to pass overhead, then connect Orbitron to the DDE plugin so that it tracking the Meteor M2 frequency automatically and then click the check box next to Demodulator. This will begin the decoding.

3. Then finally open the modified LRPToffLineDecoder.exe to see the image download in real time.

4. If you are using the scheduler in the DDE plugin, then when the satellite passes overhead Orbitron will automatically notify the DDE tracking and scheduler plugin which will in turn start the QPSK decoder and LRPT decoder automatically.

5. To check if the QPSK decoder has a lock on the signal, click the Out checkbox to show the constellation. If the constellation shows four distinct clusters of dots in the four corners then a lock has been acheived and you should begin to see an image form in thr LRPToffLineDecoder software.



RECEIVING METEOSAT WEATHER SATELLITE IMAGES: LINUX TUTORIAL

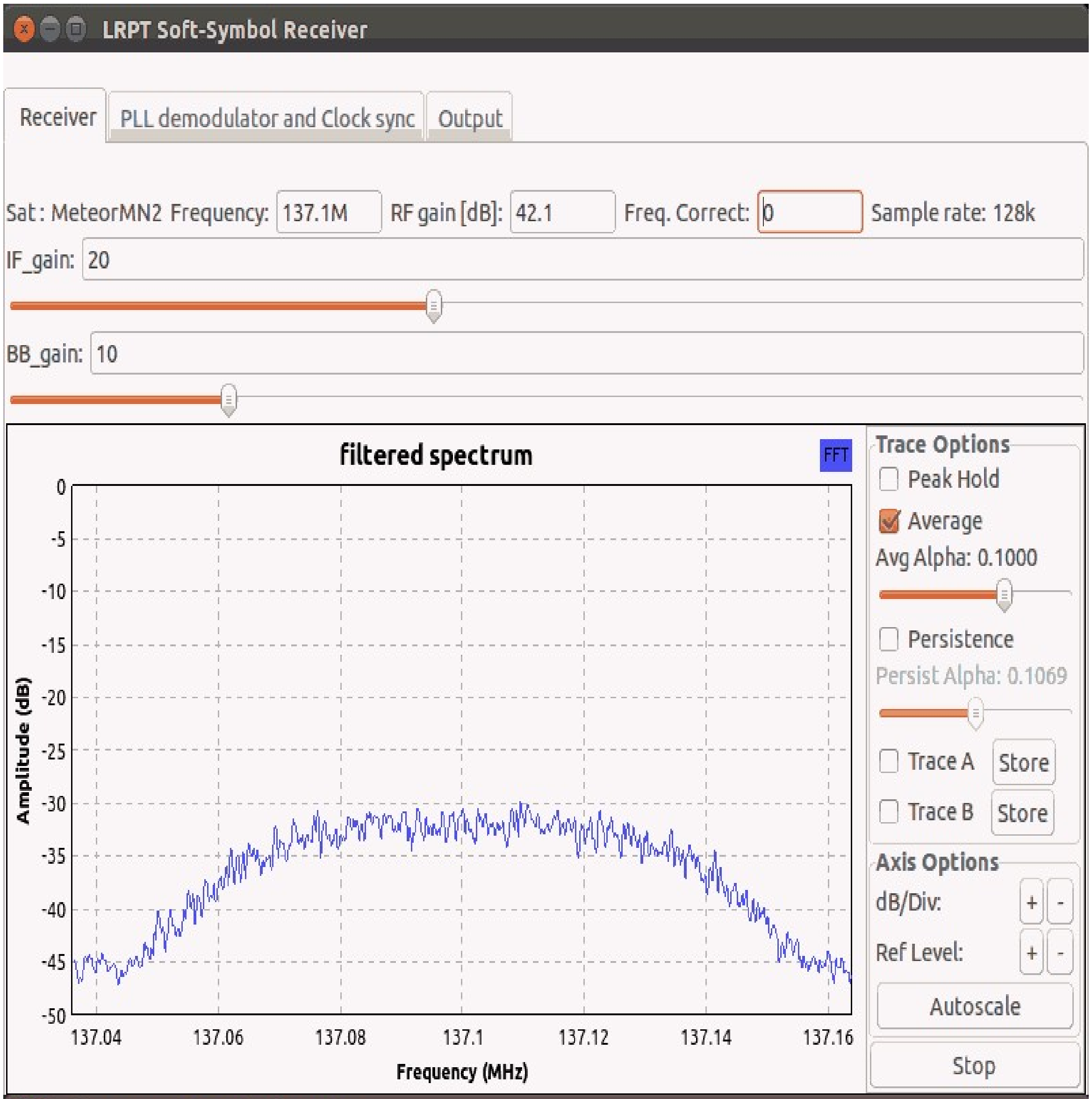
An alternative to the Windows offline method that involves less steps is to use Linux software that will do the receiving and decoding at the same time. To do this you will need a Linux system with GNU Radio 3.7+ and Python installed. See the GNU Radio Live DVDs section for a list of distributions that have GNU Radio preinstalled in them or the GNU Radio installation instructions section for information on how to install GNU Radio. Note that the final step for generating the actual images must still be done in Windows using the LRPToffLineDecode.exe decoder as shown in steps 15 and 16 in the Windows tutorial above.

First download the required GNU Radio python scripts from https://www.dropbox.com/s/par34n42m1r68k3/meteor_rx.zip (Mirror: http://bit.ly/YAhuq6), extract them to a folder and then use the following instructions.

1. First using Orbitron or another satellite tracker wait until it is time for a satellite pass to begin. When the time comes run the python script using `python meteor_qpsk_rx_rtl.py`.

2. Centre the signal in the graph using the “freq. correct” input box which is where you should enter your dongles PPM offset value.

3. Wait for the signal to noise ratio of the Meteor-M2 signal to rise above 5dB. When it does switch to the constellation graph in the “PLL demodulator and Clock sync” panel. If the signal is strong enough and has locked, the constellation should look like four clusters of dots. If it is still too weak the dots will be randomly scattered. If the constellation looks like a circle, increase the “PLL alpha” setting until you see four clusters of dots, then reduce the PLL alpha setting back down until you get to ‘1m’.
   Below we show images of a strong Meteor-M2 signal being received and what the constellation should look like when it has locked.



LRPT Soft-Symbol Receiver

Sat : MeteorMN2 Frequency: 137.1M RF gain [dB]: 42.1 Freq. Correct: 0 Sample rate: 128k

IF_gain: 20

BB_gain: 10

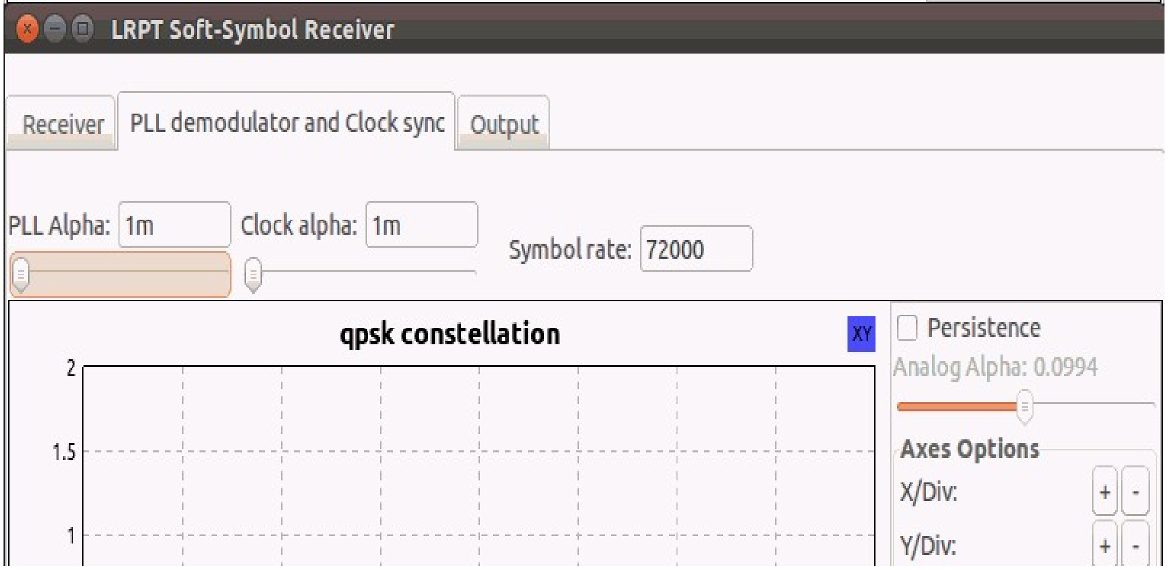
☐ Peak Hold

☑ Average

Avg Alpha: 0.1000

☐ Persistence

Persist Alpha: 0.1069

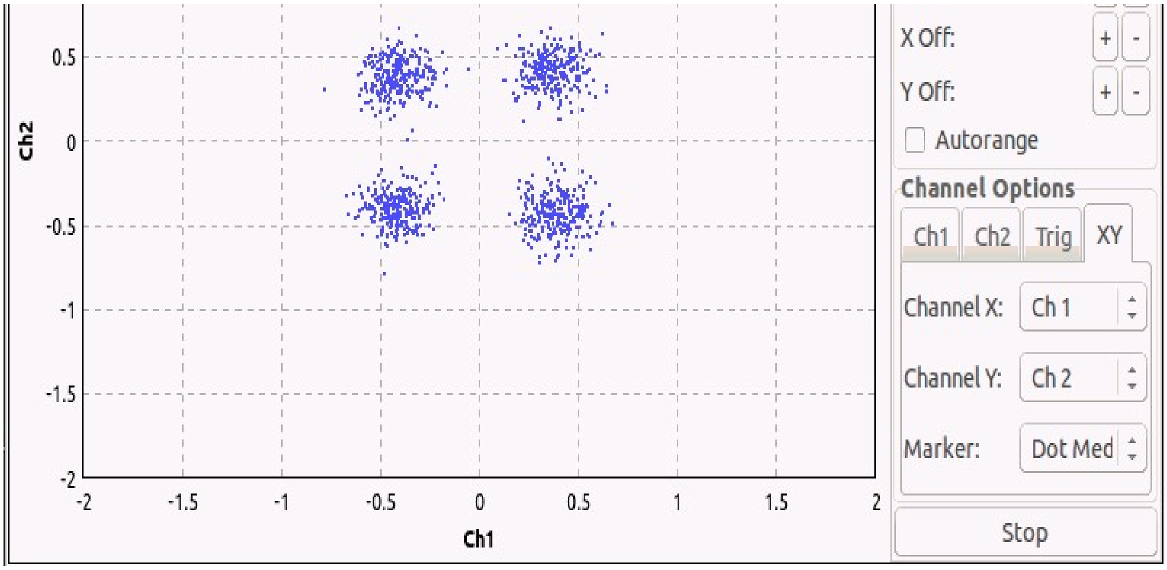


LRPT Soft-Symbol Receiver

PLL Alpha: 1m Clock alpha: 1m Symbol rate: 72000

☐ Persistence

Analog Alpha: 0.0994

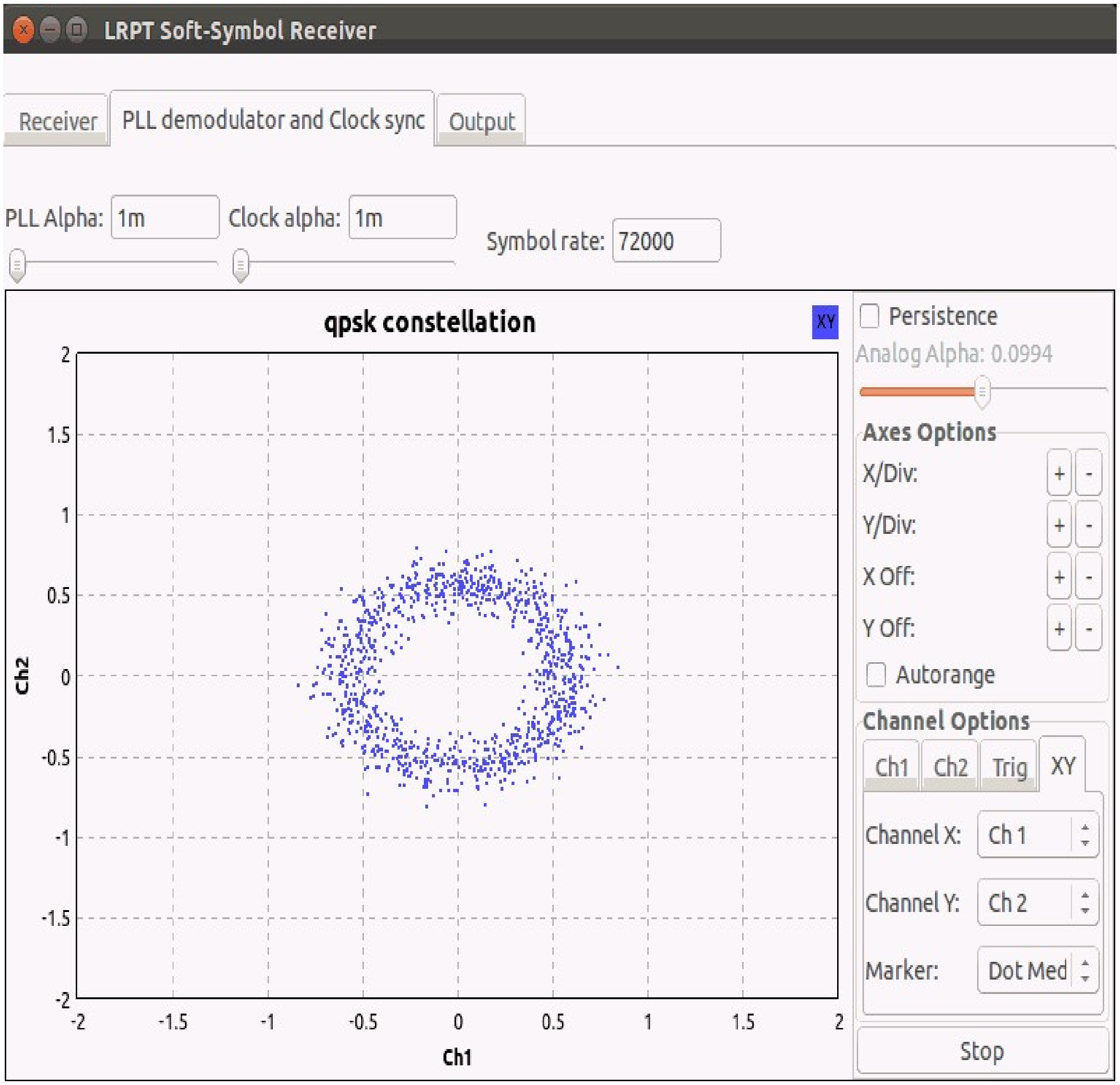


Channel X: Ch 1

Channel Y: Ch 2

Marker: Dot Med

4. If you happen to lose lock throughout the transmission and the constellation becomes a circle, then increase PLL alpha and then decrease it again when the four dots reappear. Always try to keep PLL alpha as low as possible. Changing some settings like the frequency or frequency offset might cause lock loss to occur. The image below shows an example of what the constellation will look like when loss of lock occurs.



PLL Alpha: 1m

Clock alpha: 1m

Symbol rate: 72000

□ Persistence

Analog Alpha: 0.0994

□ Autorange

Channel X: Ch 1

Channel Y: Ch 2

Marker: Dot Med

5. Once the pass has finished close the program.

6. There should now be a .wav and a .s file saved on your Linux desktop. Save the .s file to a flash drive or on a drive that is accessible by Windows.

7. Shutdown Linux and boot into Windows. Then run Lrptofflinedecoder.exe decoder using the .s file to generate the RGB image as shown in steps 15 and 16 of the Windows tutorial.

REMOVING THE FISHEYE EFFECT OF THE COLOR IMAGE

Because the images are generated with a fisheye lens, the satellite image may look distorted. It is possible to stretch the image to make it look more normal using the Smooth Meteor software for Windows available at http://myweb.tiscali.co.uk/wxsatellite/meteor3m.htm. You can also use David Taylors free Windows LRPT Image Processor program which will remove the fisheye effect as well as do color correction, his program is available at http://www.satsignal.eu/software/LRPT-processor.html.

REFERENCES

This tutorial was adapted from Happysats pdf tutorial available at http://www.rtl-sdr.com/wp-content/uploads/2014/09/Happysat_Meteosat_LRPT_Tutorial.pdf.

WEATHER BALLOON (RADIOSONDE) GUIDE

Around the world meteorological weather balloons are launched twice daily. Once launched they continuously transmit weather telemetry to a ground station using something called a radiosonde. The RTL-SDR combined with a decoding program can be used to intercept this telemetry and display the weather data on your own computer. You will be able to see real time graphs and data of air temperature, humidity, pressure as well as the location and height of the balloon as it makes its ascent and descent.

Note that radiosonde decoding may be difficult in many places in the USA. Recently the USA switched to Vaisala radiosondes, some of which are decodable. Unfortunately, they appear to be using the more difficult to receive frequency of 1680 MHz rather than the more common 403 - 406 MHz range used by the rest of the world. It is possible that some stations use the lower frequency band however. SondeMonitor is able to decode Digital RS92SGP, Analogue RS92AGP, Analogue RS80, Analogue 92KL, Graw DFM-06, MeteoModem M2K2/M10 and Meteolabor SRS-C34 radiosonde protocols.

This tutorial is also applicable to other software defined radios such as the Funcube dongle, HackRF, BladeRF or even hardware radios with discriminator taps, but the RTL-SDR is the cheapest option that will work.

RADIOSONDE RECEIVING TUTORIAL

RADIOSONDE ANTENNA TUTORIAL

In Europe and the Pacific, most radiosondes transmit at a frequency that is between 403 and 406 MHz. In the USA and Asia they most commonly transmit in the 1680 MHz band between 1668.4 and 1700 MHz. You will need an antenna capable of receiving those frequencies. Antennas good for the 403 to 406 MHz band could be a quarter wave groundplane, dipole, j-pole, turnstile or a QFH antenna. Circularly polarized antennas tend to work better. If you have good signal some of the previously mentioned antennas may also work well for the 1680 MHz band if sized for that frequency. A Yagi antenna works especially well for the 1680 MHz band if you require more gain from a weak signal and are willing to track the radiosonde by hand or other means. See the antenna guide for more information about these antennas.

An LNA and preselector may also aid reception. See the LNA section and preselector section for more information.

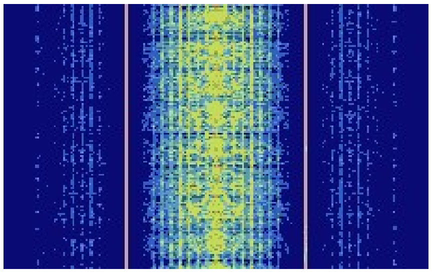
SOFTWARE TUTORIAL

In order to receive and decode weather balloon radiosondes you will need to ready five things.

- An antenna capable of picking up signals near 403 to 406 MHz (or 1680 MHz).
- A decoding program called SondeMonitor which is able to decode radiosonde telemetry.
- The knowledge of when and optionally where weather balloon radiosondes are launched in your area.
- The knowledge of what type of radiosonde protocol is used in your area (can sometimes be guessed with trial and error).
- An audio piping method. See Appendix A: Audio Piping for more information if you don’t have one set up.

Weather balloons are usually launched 45 minutes before the official observation time of 0000 UTC and 1200 UTC, though times may be different in some countries. You will need to either Google or check with your local meteorological agency for a more exact time of when they are launched in your area. You will also need to check with your local meteorological agency on the type of radiosonde that they are using. They are often happy to oblige this information.

One of the most commonly used radiosonde signals is the Vaisala RS92SGP digital signal. A waterfall example of this signal is shown below. It has a bandwidth of around 5 kHz.



To decode weather balloon radiosondes follow these steps:

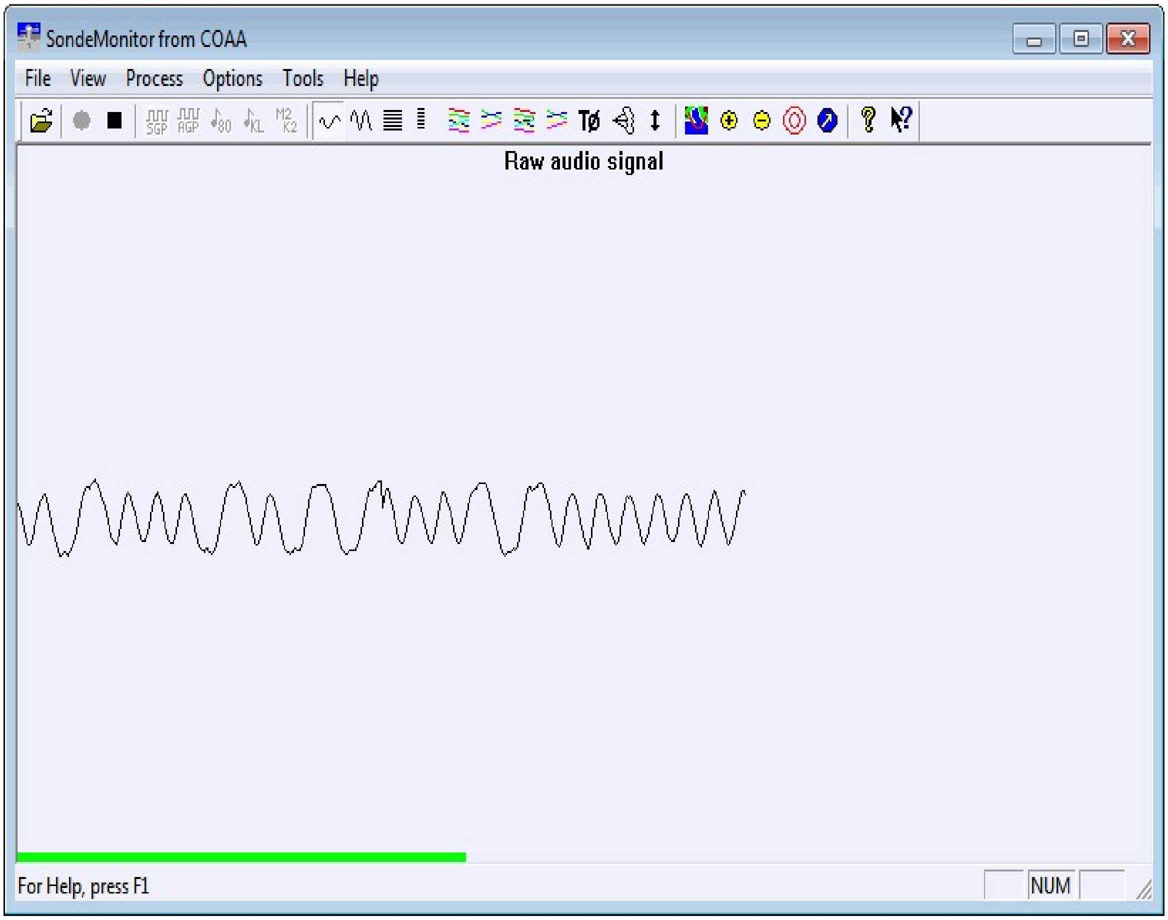
1. Download and install SondeMonitor. SondeMonitor is paid software with a 25 euro price tag, but comes with a 21 day trial. Download SondeMonitor from http://www.coaa.co.uk/sondemonitor.htm.

2. When a weather balloon is due to be launched, open SDR# or another radio receiver GUI program, set your audio piping method under the Audio Output drop down box and tune to a frequency between 403 and 406 MHz (or 1680 MHz). Scan around and look for a radiosonde signal, which should be a narrowband signal of around 5 kHz bandwidth. If you find one tune to it and adjust the RF gain settings for best performance. Set the receive mode to NFM and filter audio to OFF. Adjust the filter bandwidth so that it just covers the signal.

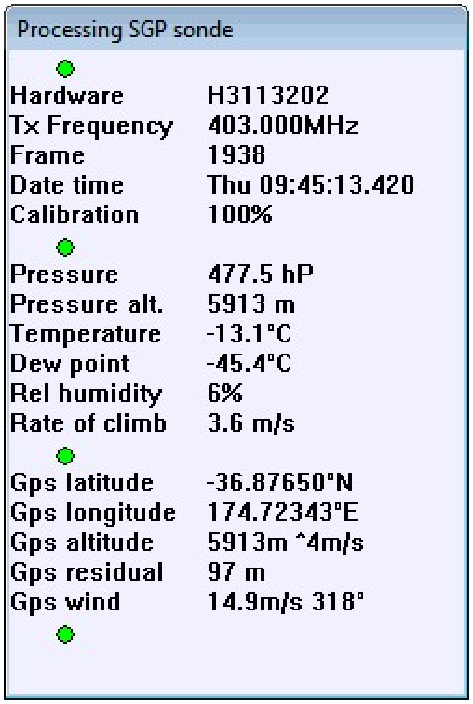
3. Open SondeMonitor and go to **Options -> Audio -> Audio Source** and choose the audio piping method you are using.

4. Select the radiosonde decoding protocol used in your area. These can be selected with these icons SGP AGP 80 KL M2 K2 which are to the right of the start and stop icons. Hovering over the icons will show which protocol they represent. If you don’t know what protocol your area uses, you can try each one one at a timeand hope that one works

5. Now start the decoding by clicking the green start circle ●. A telemetry data window will pop up. Next, click on the raw signal icon ∿. Adjust the audio volume using the audio gain in SDR# or Windows volume settings so that the audio graph is loud enough to be visible, but not too loud as to cause clipping. Clipping is when the signal is too loud and the waveform begins to look squarish. An ideal volume is shown below.



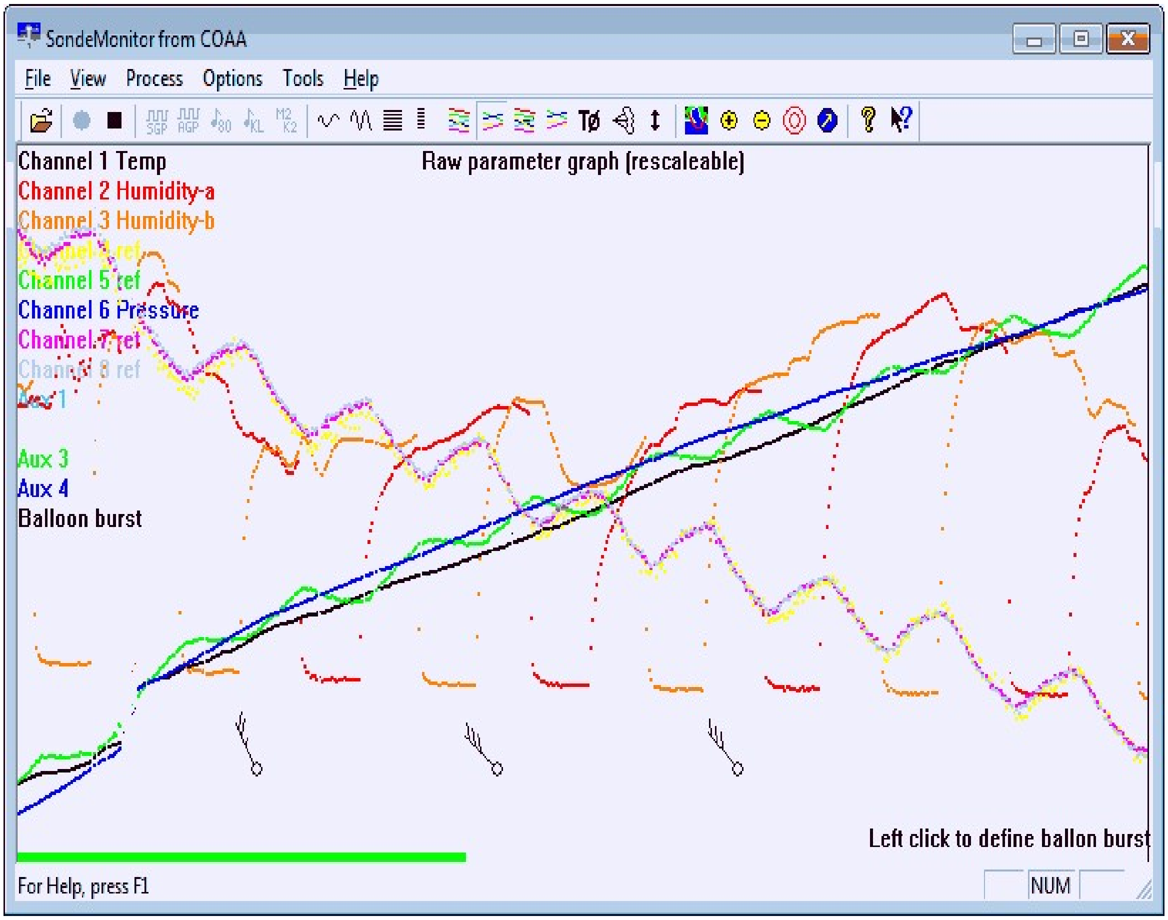
6. At this point, if your signal reception is good and you have selected the correct decoding method, the telemetry window will show data and have four green circles. Note that SondeMonitor requires some initial time to calibrate. The amount of the calibration process completed can be seen in the red bar in the lower left corner of the main window, which will slowly turn green as calibration completes. The calibration status can also be tracked in the telemetry window. If calibration does not complete, or takes a very long time, it means your signal strength is not good enough.



Processing SGP sonde

Hardware	H3113202
Tx Frequency	403.000MHz
Frame	1938
Date time	Thu 09:45:13.420
Calibration	100%
Pressure	477.5 hP
Pressure alt.	5913 m
Temperature	-13.1°C
Dew point	-45.4°C
Rel humidity	6%
Rate of climb	3.6 m/s
Gps latitude	-36.87650°N
Gps longitude	174.72343°E
Gps altitude	5913m ^4m/s
Gps residual	97 m
Gps wind	14.9m/s 318°

7. Click on the Graph 2 icon to see telemetry graphs. You can click the Autoscale icon ↕ to ensure that all the data is displayed on the screen.



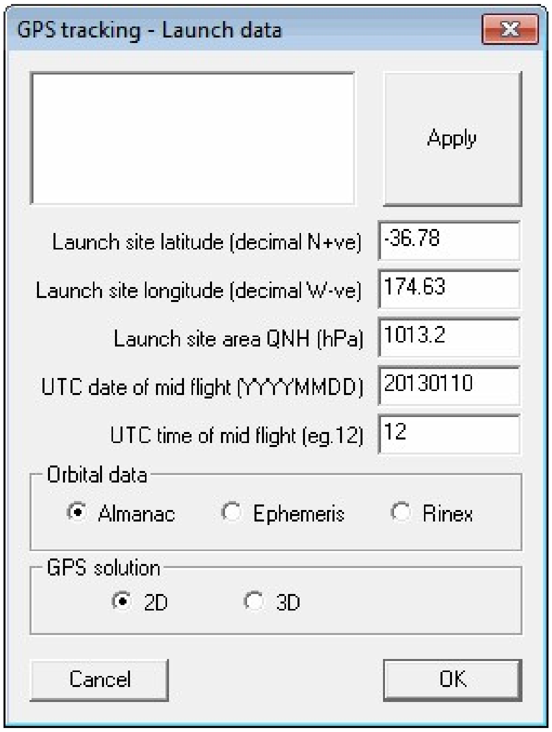
GPS RADIOSONDE LOCATION TRACKING

For most launches, the weather balloon's live location and altitude can be tracked live in Google Earth. However, some radiosonde protocols such as the RS92SGP do not transmit latitude and longitude GPS data directly. Instead, they transmit raw GPS data which must be converted into longitude and latitude on the receiving end (your computer). To do this, the radiosonde's starting coordinates, UTC launch date and UTC time and an up to date GPS almanac are required. The almanac is a data file that stores information about the GPS satellite locations.

The MeteoModem M2K2/M10, Graw DFM-06 and Meteolabor SRS-C34 are radiosondes which transmit already decoded GPS location data. The following steps are only for the other radiosondes which transmit raw GPS data, such as the RS92SGP.

1. Download an up to date Almanac from the US Coast Guard Nav Centre at http://www.navcen.uscg.gov/?pageName=gpsAlmanacs. You will want to get the current SEM Almanac with the .al3 extension.

2. In SondeMonitor, go to **Tools -> GPS Arm**. Enter the Latitude and Longitude of the weather balloon's launch location. You will also need to enter the UTC date of the flight and the approximate UTC time at mid flight. Make sure the date and time is in UTC time. Weather balloons last about 2 to 4 hours, so just estimate a mid flight time. Select Almanac as your Orbital data. The other options can be left as default.



Launch site latitude (decimal N+ve) -36.78

Launch site longitude (decimal W-ve) 174.63

Launch site area QNH (hPa) 1013.2

UTC date of mid flight (YYYYMMDD) 20130110

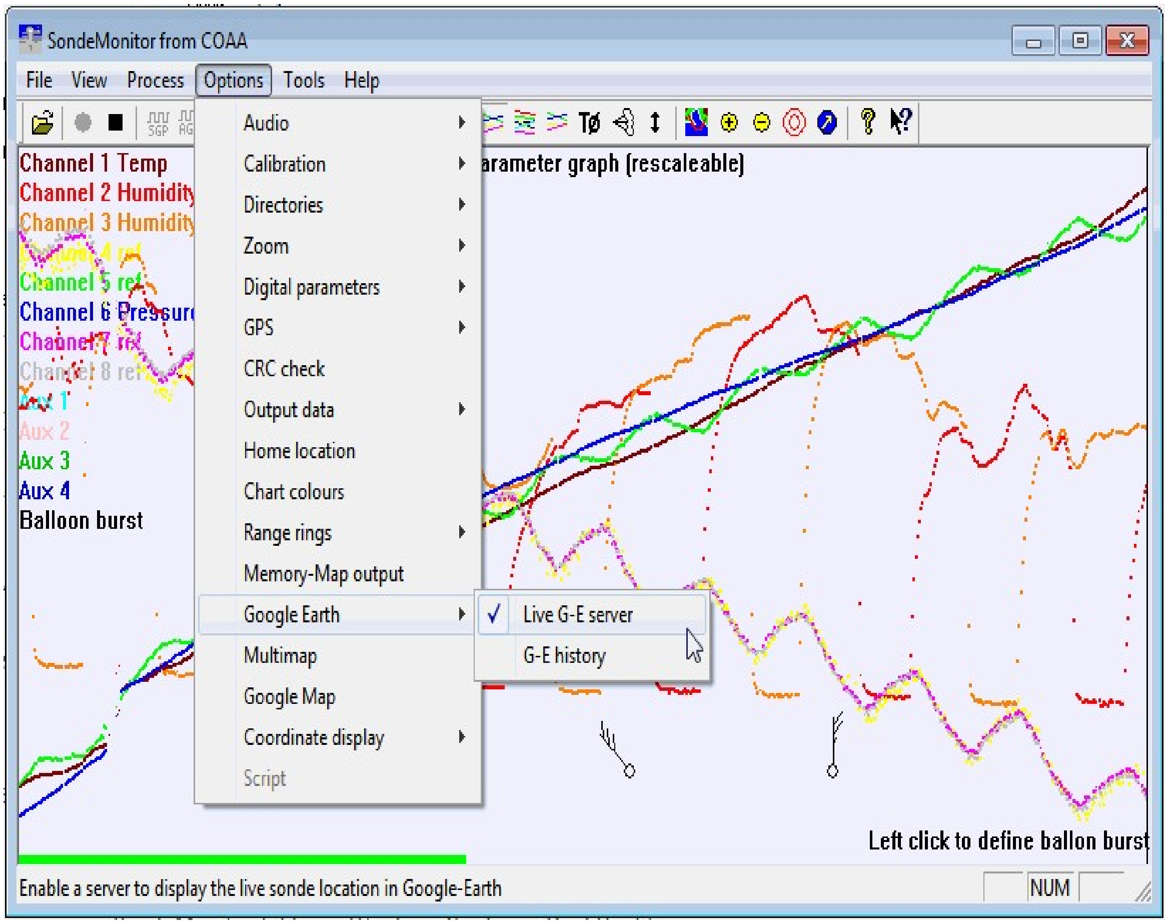
UTC time of mid flight (eg.12) 12

Orbital data: ● Almanac ○ Ephemeris ○ Rinex

GPS solution: ● 2D ○ 3D

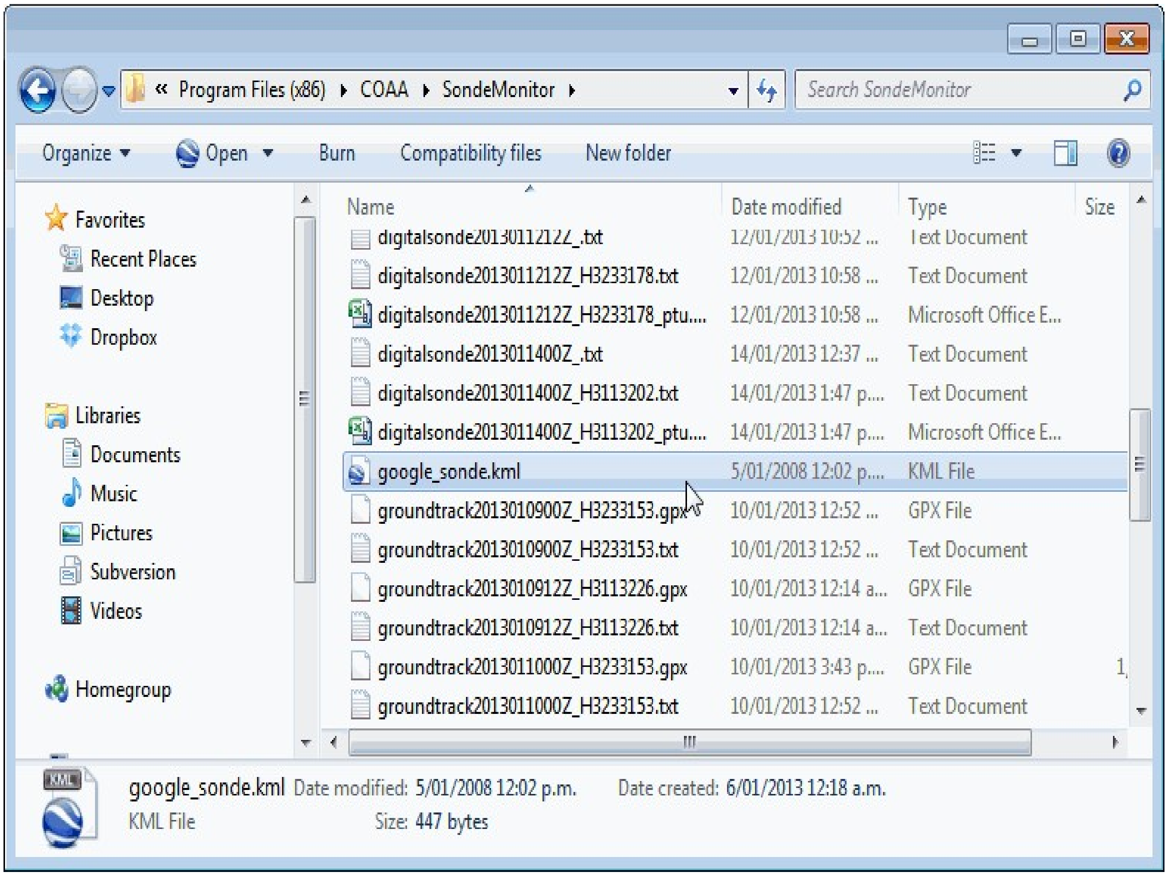
3. Once you press OK, a file selection dialog box will pop up. Browse to the folder where your current .al3 Almanac file is stored in and open it. Now, if you have set the Almanac, start coordinates and date and time set up correctly, the GPS Residual in the SondeMonitor telemetry status window will be to within a couple of hundred meters. The smaller the residual, the more accurate the weather balloon location is. If you have gotten any parameters wrong, the residual will be very large.

4. To see the balloon in Google Earth, go to **Options -> Google Earth** and ensure that Live G-E Server has a check next to it by clicking it if it does not.



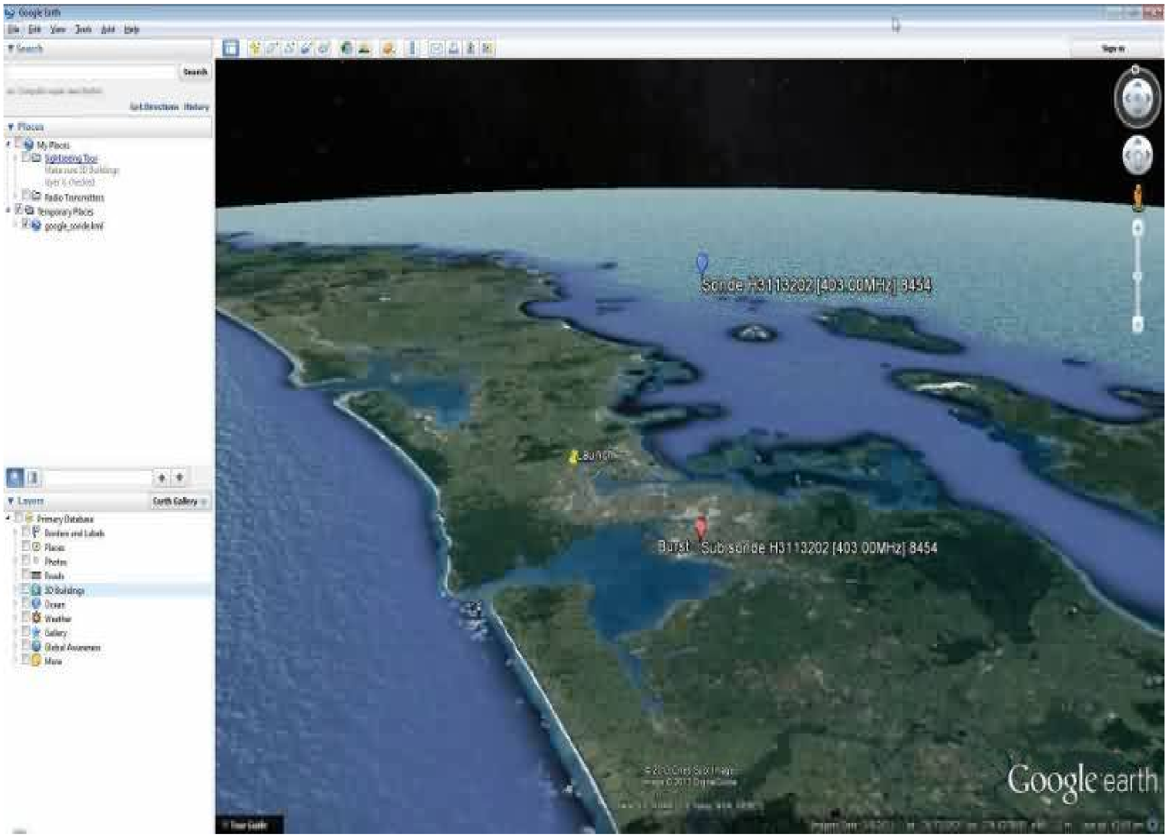
Enable a server to display the live sonde location in Google-Earth

5. Now open your SondeMonitor installation folder in Windows Explorer. Find the file google_sonde.kml and open this file with Google Earth. You should now be able to see the weather balloon's launch location and current live location in Google Earth.



Name	Date modified	Type	Size
digitalsonde2013011212Z_.txt	12/01/2013 10:52 ...	Text Document	
digitalsonde2013011212Z_H3233178.txt	12/01/2013 10:58 ...	Text Document	
digitalsonde2013011212Z_H3233178_ptu....	12/01/2013 10:58 ...	Microsoft Office E...	
digitalsonde2013011400Z_.txt	14/01/2013 12:37 ...	Text Document	
digitalsonde2013011400Z_H3113202.txt	14/01/2013 1:47 p....	Text Document	
digitalsonde2013011400Z_H3113202_ptu....	14/01/2013 1:47 p....	Microsoft Office E...	
google_sonde.kml	5/01/2008 12:02 p....	KML File	
groundtrack2013010900Z_H3233153.gpx	10/01/2013 12:52 ...	GPX File	
groundtrack2013010900Z_H3233153.txt	10/01/2013 12:52 ...	Text Document	
groundtrack2013010912Z_H3113226.gpx	10/01/2013 12:14 a...	GPX File	
groundtrack2013010912Z_H3113226.txt	10/01/2013 12:14 a...	Text Document	
groundtrack2013011000Z_H3233153.gpx	10/01/2013 3:43 p....	GPX File	1,
groundtrack2013011000Z_H3233153.txt	10/01/2013 12:52 ...	Text Document	

google_sonde.kml Date modified: 5/01/2008 12:02 p.m. Date created: 6/01/2013 12:18 a.m.
KML File Size: 447 bytes



ADDITIONAL LINKS AND REFERENCES:
Video Demo: https://www.youtube.com/watch?v=Mz2id1cBmjs

Results by another experimenter: http://nerdsville.blogspot.com/2014/09/radiosonde-decoding.html

MARINE AUTOMATIC IDENTIFICATION SYSTEM (AIS) GUIDE

Large ships and passenger boats are required to broadcast an identification signal containing GPS position, course, speed, destination and vessel dimension information. This signal is used to help prevent collisions between boats as it acts as a sort of boat radar system, similar to the ADS-B system used for aircraft. The system is known as the "Automatic Identification System" or AIS for short.

There are dedicated AIS receivers intended to be used on boats, or by hobbyists, but they can be expensive. A radio scanner or the cheap RTL-SDR can be used to receive these signals and with the help of decoding software, ship positions can be plotted on a map.

This tutorial will show you how to set up an AIS receiver with the RTL-SDR. Most parts of this tutorial are also applicable to other software radios, such as the funcube dongle and HackRF, or even regular hardware scanners if a discriminator tap is used, but the RTL-SDR is the cheapest option.

Safety Warning: This probably should not be used a navigational aid on a boat as the field reliability of the RTL-SDR or other software radios is not proven. This guide is intended for land based scanner hobbyists.

AIS TUTORIAL

To set up an AIS ship radar on a windows system you will need:

- An audio piping method. See Appendix A: Audio Piping for more information if you don't have one set up.
- A vertically polarized antenna tuned to 162MHz.
- Software for decoding the AIS signals.

We will assume you have the RTL-SDR dongle set up and working already. If you have not bought a dongle yet, see our Buy RTL-SDR page at www.rtl-sdr.com/buy-rtl-sdr-dvb-t-dongles/ for information and the check out the Quick Start Guide section shown earlier in this book for an easy setup procedure with SDR#. You will also need to have an audio piping method installed and set up. Audio piping will allow the audio from a SDR receiver program to be passed to a decoding program. See Appendix A: Audio Piping for information about audio piping methods.

AIS ANTENNA GUIDE

AIS signals are broadcast at both 161.975 MHz and 162.025 MHz and have a maximum range of approximately 75 kilometres. So if your radio setup is more than 75 kilometres away from any boats, you will probably not be able to receive AIS signals, though in some cases with good visibility they can travel much further. AIS is also a line of sight signal, meaning that AIS signals could be blocked if there are large buildings or mountains between your antenna and the boats. Because of this reason it is important to put your antenna as high up as possible.

A good antenna for AIS is a collinear antenna. The collinear antenna has very high omnidirectional gain directed towards the horizon. This means it will receive signals best from sources that are near the horizon, which is where ships will be. Other antennas such as a quarter wave ground plane, j-pole, dipole or discone may also work well. If the majority of ships in your area are focused in one direction only, a Yagi antenna may be a good choice. A Yagi is a very directional antenna - it will only pick up signals in the direction it is pointed. The advantage is that with directionality we can get a large signal gain. A Yagi is also usually only sensitive to a small frequency range, so a properly tuned Yagi will also help to keep interfering signals out. See the Antenna Guide section for more information about these antennas.

A bandpass filter with a centre frequency near 162 MHz such as a surface mount one from http://www.minicircuits.com/products/filters_sm_bandpass.shtml may also be useful. Alternatively, a bandstop filter to filter out the FM broadcast band and pagers may help. A pager bandstop filter may be particularly useful as pager frequencies in some countries tend to be at around 157 MHz, near AIS frequencies. A coax stub filter may be a cheap solution for handling pager interference.

An LNA such as the LNA4ALL at the antenna may also help receive weak signals. See the LNA section and preselector section for more information.

SOFTWARE TUTORIALS

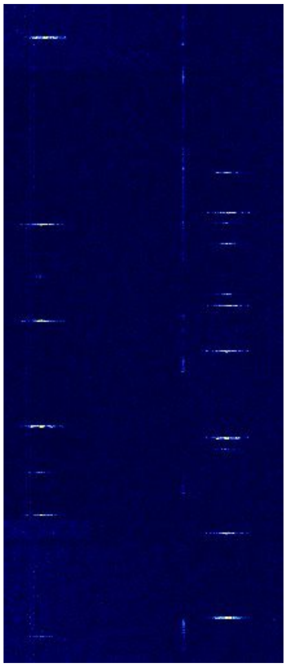
1. Set up two audio pipes in Windows using either Virtual Audio Cable or VBCable. Ensure that the sample rates are set to 48 kHz. See the setting the sample rate section for information on how to change this.

2. Use a SDR receiver such as SDR# or SDR-Radio to tune to an AIS signal. In your SDR receiver set your audio piping method in the Audio output drop down box and tune to an AIS frequency (161.975 MHz or 162.025 MHz). The signals may not appear exactly on the AIS frequencies, since the RTL-SDR is not frequency accurate. Just tune manually until the signals are properly centred, or correctly set your PPM offset.
   For AIS we highly recommend using SDR-Radio as it will allow you to tune to both AIS channels simultaneously. If using SDR-Radio, set up two VFOs to listen to both AIS channels. Ensure that each one outputs to a separate audio piping device.

3. Adjust the RF gain until you get good reception of the AIS signals. You want to adjust the RF gain such that the signal is strong, but the noise floor is low.

4. Set the receive mode to NFM and expand the bandwidth so that it just covers the AIS blips on the waterfall. The bandwidth will be either around 12.5 kHz or 25 kHz (there are two AIS standards with different bandwidths). Turn off the squelch and any DSP algorithms. If using SDR# set Filter Audio to OFF.

5. AIS signals look like small horizontal lines on the waterfall, as is shown on the image below (if you live in a high marine traffic area there will be many more horizontal lines). If you are listening to the AIS signals through your speakers, they will just sound like blips of noise. The two AIS frequencies broadcast the same information and are used in commercial AIS receivers for redundancy.



Once you have AIS reception set up you can then use a program for AIS decoding. There are several programs useful for decoding and displaying AIS information. ShipPlotter is a commercial program with a 21-day trial. AISMon + OpenCPN or AISDispatcher are free programs, but are a little harder to set up.

OPENCPN + AISMON TUTORIAL

AISMon is a free AIS decoding program. It can be downloaded from the AISMon Yahoo group files section at http://groups.yahoo.com/group/aismon/. You may need to first join this group using a Yahoo account to access the files section. The Yahoo group also contains a sample AIS .wav file, which when used with stereo mix, can be used to test both AISMon and ShipPlotter.

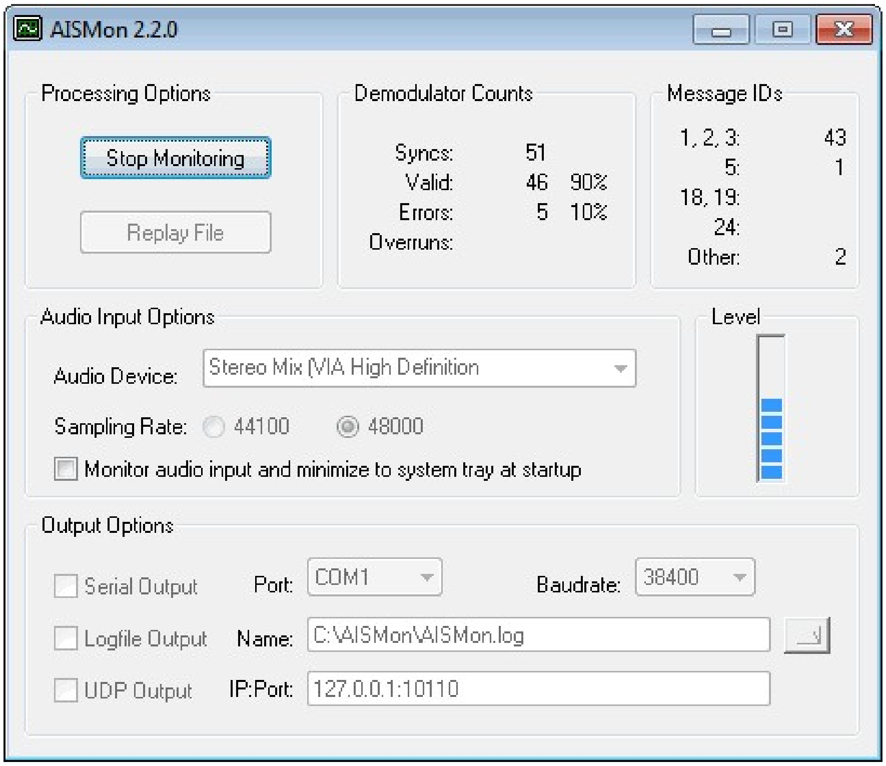
One advantage to AISMon over ShipPlotter (shown in the next section) is that we can open two instances of the program and use them to decode the two AIS channels simultaneously which will improve the ship update rate.

1. Open AISMon, set the audio piping device to the one you have chosen to use and set the sampling rate to 48000 (assuming you've set your audio piping device to use a sample rate of 48000 as well, see the setting the sample rate section for more information). Press start monitoring.

2. Adjust the volume in your SDR receiver software and/or the Windows volume settings until the level meter in AISMon reads at about halfway.

3. If everything is working you should begin to see numbers appear in the "Demodulator Counts" section of AISMon. If you receive many demodulator errors, check that you have properly centered the AIS signal on the tuning bar and are using a bandwidth wide enough to cover the entire AIS signal.

4. Open a second instance of AISMon and repeat steps 1-3 for the second audio pipe if you are using SDR-Radio with two VFO's.



Syncs:	51	
Valid:	46	90%
Errors:	5	10%
Overruns:		

1, 2, 3:	43
5:	1
18, 19:	
24:	
Other:	2

Audio Device: Stereo Mix (VIA High Definition

Sampling Rate: ○ 44100 ● 48000

□ Monitor audio input and minimize to system tray at startup

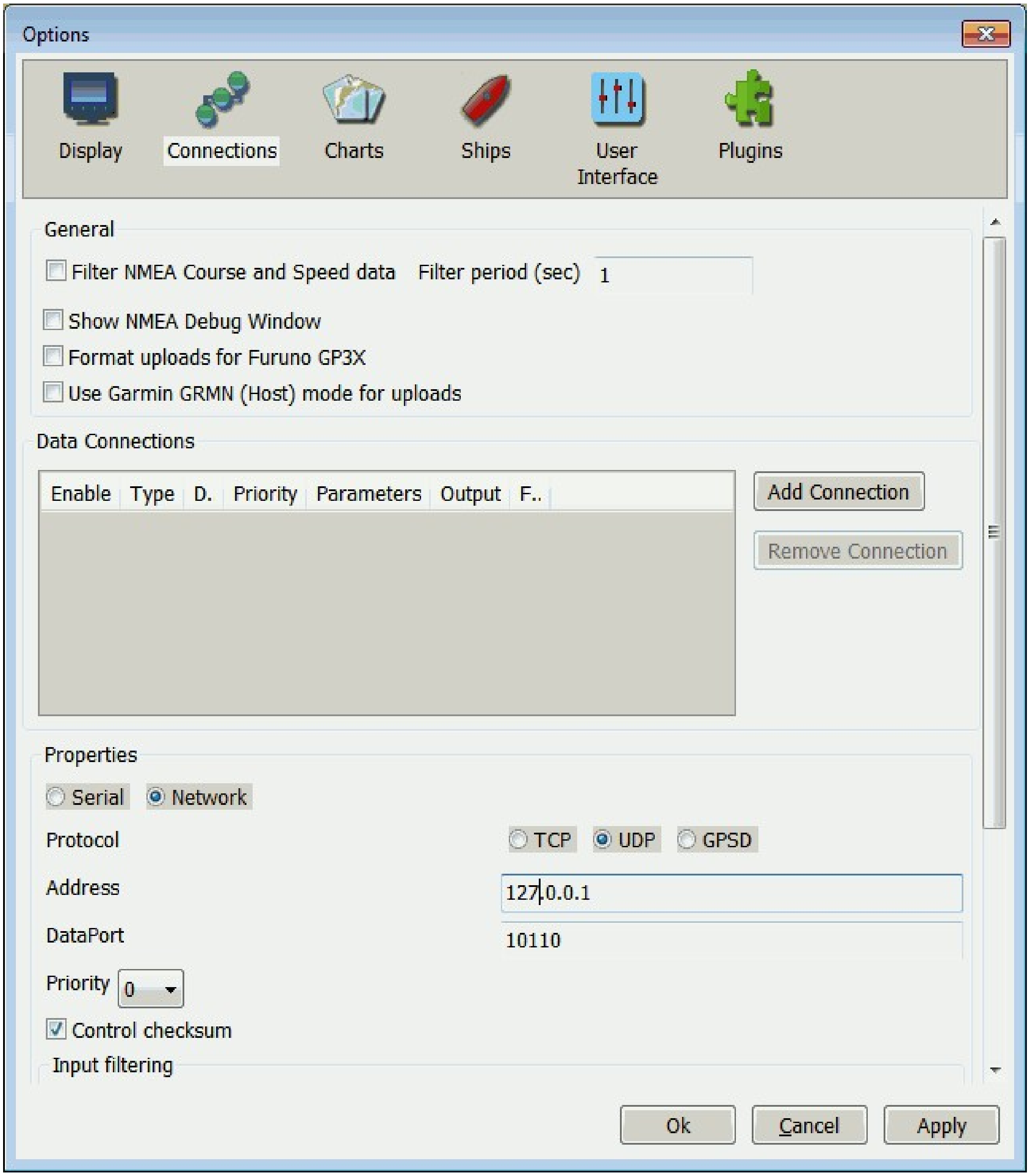
□ Serial Output Port: COM1 Baudrate: 38400

□ Logfile Output Name: C:\AISMon\AISMon.log

□ UDP Output IP:Port: 127.0.0.1:10110

By itself, AISMon does not display any ship information. To view ship information you will need another free open source program called OpenCPN which can be downloaded from http://opencpn.org/ocpn/. OpenCPN is a chart plotting and navigation program which can read the NMEA information that is output by AISMon and plot the ship positions on a map.

1. Download and install OpenCPN from http://opencpn.org/ocpn/.
2. Open OpenCPN and then click on the Options button and go to the Connections tab.
3. Under Data Connections, click on Add Connection and add a Network UDP connection. Use an address of 127.0.0.1 and a port of 10110. Click Apply and OK. If you are using SDR-Radio and are using two audio outputs, add a second UDP connection identical to the first one, but with a port number of 10120 or any other unused port.



General

□ Filter NMEA Course and Speed data   Filter period (sec) 1

□ Show NMEA Debug Window

□ Format uploads for Furuno GP3X

□ Use Garmin GRMN (Host) mode for uploads

Data Connections

Enable	Type	D.	Priority	Parameters	Output	F..

Properties

○ Serial ● Network

Protocol ○ TCP ● UDP ○ GPSD

Address 127.0.0.1

DataPort 10110

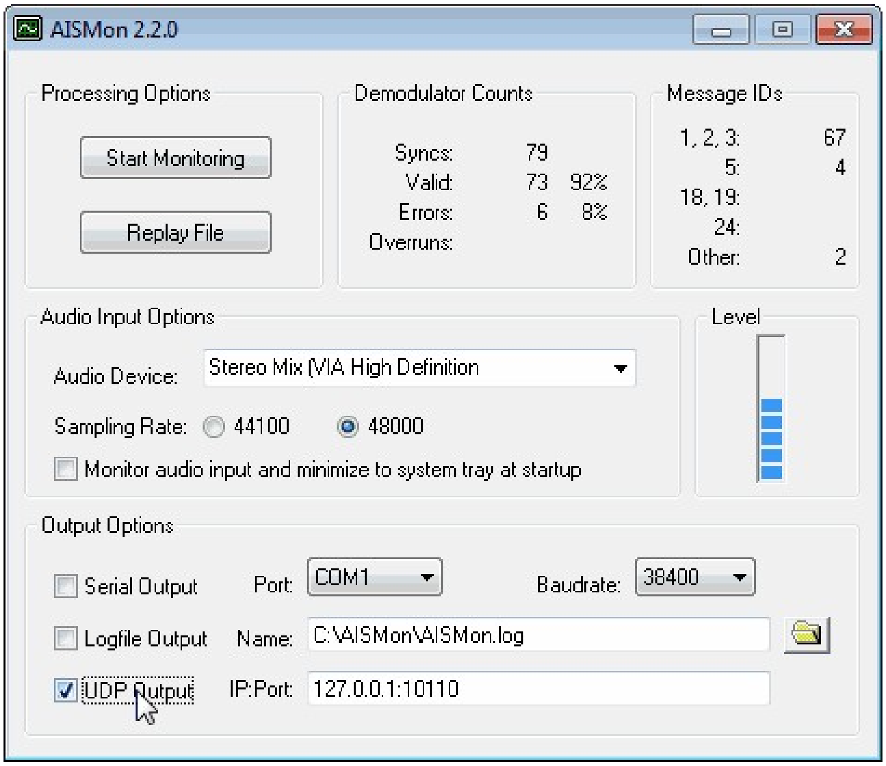
Priority 0

☑ Control checksum

Input filtering

4. Now back in AISMon under Output Options, check UDP Output, and enter 127.0.0.1:10110 into the IP:Port box. (You will need to stop monitoring first to access these options if monitoring is currently active).

5. If you are running SDR-Radio and have two audio outputs for each AIS channel, you can open two AISMon instances where each one uses a separate audio device. Set the UDP output port of the second AISMon instance to 10120 or whatever port you have chosen to use in OpenCPN.



Syncs: 79
Valid: 73 92%
Errors: 6 8%
Overruns:

1, 2, 3: 67
5: 4
18, 19:
24:
Other: 2

Audio Device: Stereo Mix (VIA High Definition

Sampling Rate: ○ 44100 ● 48000

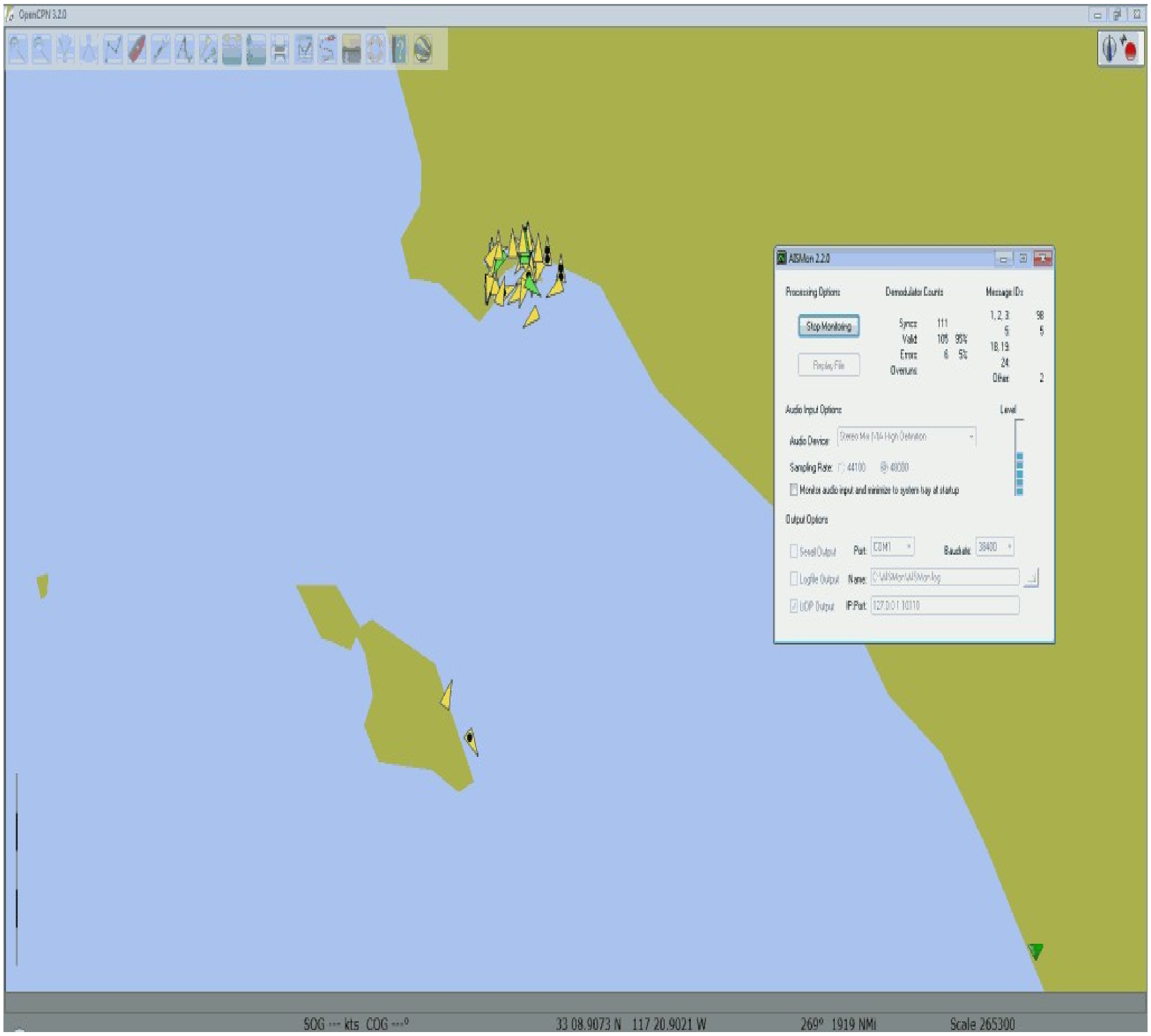
□ Monitor audio input and minimize to system tray at startup

□ Serial Output Port: COM1 Baudrate: 38400

□ Logfile Output Name: C:\AISMon\AISMon.log

☑ UDP Output IP:Port: 127.0.0.1:10110

6. You can now click Start Monitoring again. If everything has been set up correctly, ships will begin to appear in the OpenCPN map.



Processing Options

Demodulator Counts		
Syncs	111	
Valid	105	95%
Errors	6	5%
Overruns		

Message IDs	
1, 2, 3	98
5	5
18, 19	
24	
Other	2

Audio Input Options

Audio Device: Stereo Mix (VIA High Definition

Sampling Rate: ○ 44100 ● 48000

□ Monitor audio input and minimize to system tray at startup

Level

Output Options

□ Serial Output Port: COM1 Baudrate: 38400

□ Logfile Output Name: C:\AISMon\AISMon.log

☑ UDP Output IP:Port: 127.0.0.1 10110

SOG --- kts COG ---° 33 08.9073 N 117 20.9021 W 269° 1919 NMi Scale 265300

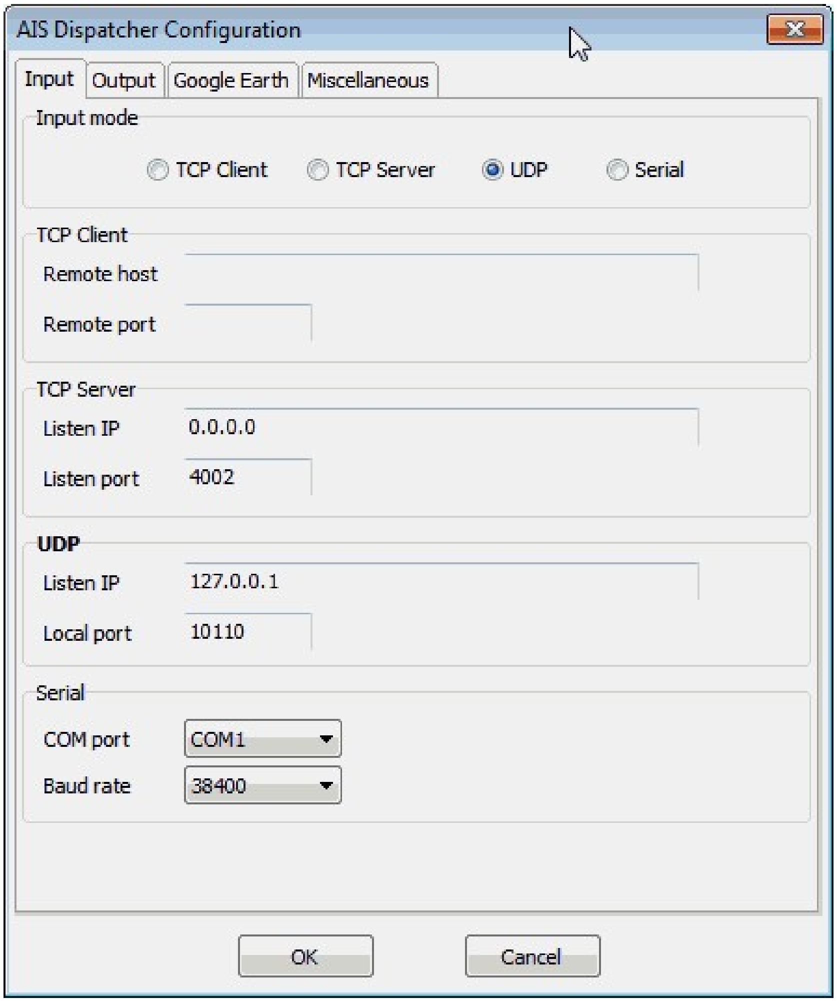
If you want more accurate maps or charts you will need to follow the instructions on the OpenCPN website for downloading charts for your particular location.

AIS DISPATCHER + AISMON TUTORIAL

Another simple and easy to use free program for plotting ships with AISMon is AIS Dispatcher. Download it from http://www.aishub.net/aisdispatcher-description.html. It is available for Windows, Linux and MacOS. After downloading and extracting the AIS Dispatcher zip file to a folder, follow these instructions to set it up.

1. Open and start AISMon and your favourite SDR Receiver and tune to an AIS channel as explained in the above OpenCPN + AISMon tutorial.
2. Open AIS Dispatcher by double clicking on AISDispatcher.exe in the extracted folder. Click on the Configuration button.

3. In the Input tab set the UDP Listen IP to 127.0.0.1 and the Local Port to 10110, or to whatever port you have chosen to use in AISMon.



Input mode

○ TCP Client ○ TCP Server ● UDP ○ Serial

TCP Client

Remote host

Remote port

TCP Server

Listen IP 0.0.0.0

Listen port 4002

UDP

Listen IP 127.0.0.1

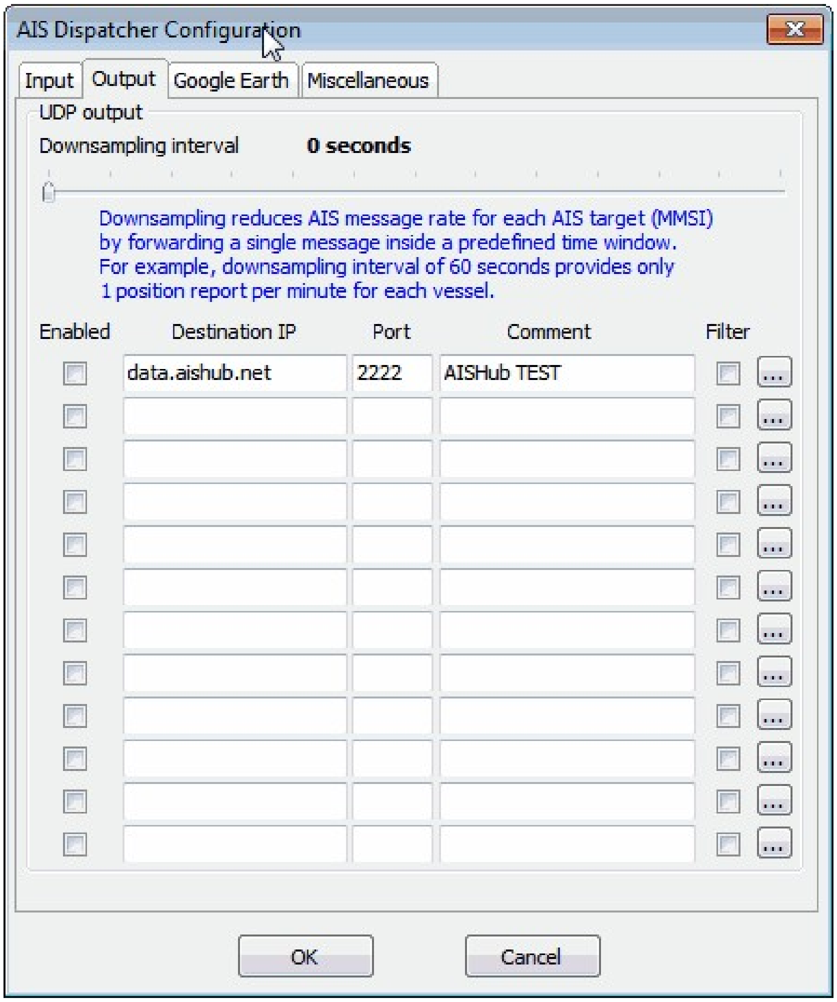
Local port 10110

Serial

COM port COM1

Baud rate 38400

4. In the Output tab disable any UDP outputs if you don't want to share your data. Press OK.



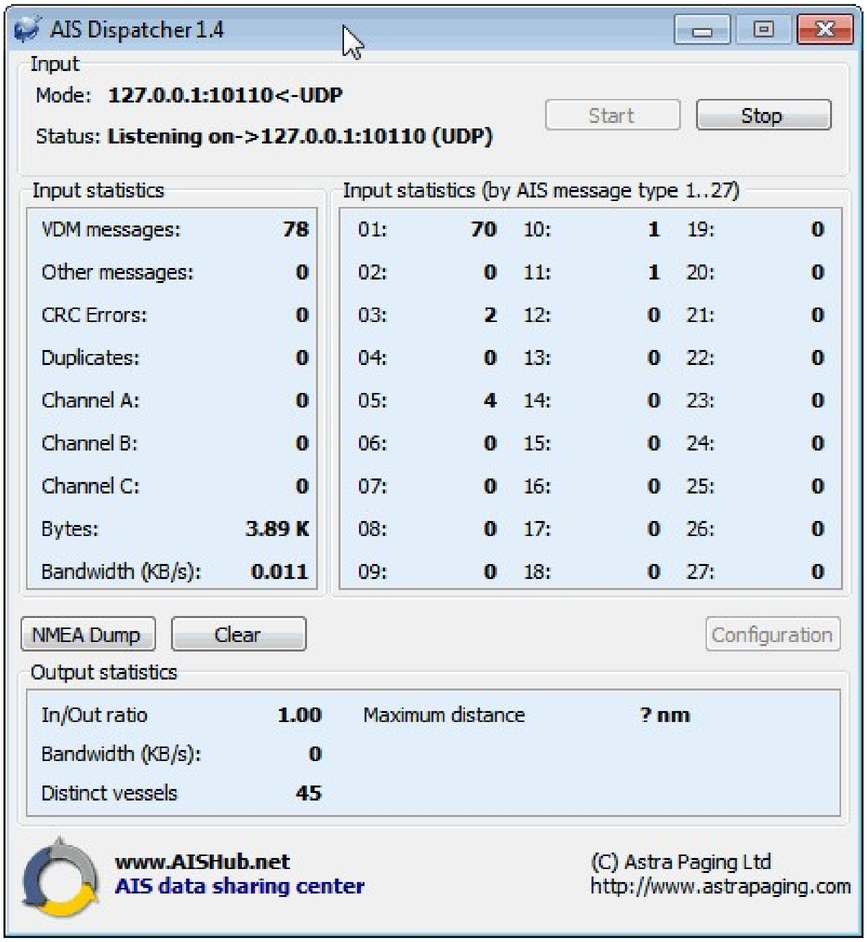
UDP output

Downsampling interval **0 seconds**

Downsampling reduces AIS message rate for each AIS target (MMSI) by forwarding a single message inside a predefined time window. For example, downsampling interval of 60 seconds provides only 1 position report per minute for each vessel.

Enabled	Destination IP	Port	Comment	Filter	
□	data.aishub.net	2222	AISHub TEST	□	...
□				□	...
□				□	...
□				□	...
□				□	...
□				□	...
□				□	...
□				□	...
□				□	...
□				□	...
□				□	...
□				□	...

5. In the main AIS Dispatcher window click the Start button to begin listening to the decoded AISMon messages.



Input

Mode: **127.0.0.1:10110<-UDP**

Status: **Listening on->127.0.0.1:10110 (UDP)**

Input statistics

VDM messages:	**78**
Other messages:	**0**
CRC Errors:	**0**
Duplicates:	**0**
Channel A:	**0**
Channel B:	**0**
Channel C:	**0**
Bytes:	**3.89 K**
Bandwidth (KB/s):	**0.011**

Input statistics (by AIS message type 1..27)

01:	**70**	10:	**1**	19:	**0**
02:	**0**	11:	**1**	20:	**0**
03:	**2**	12:	**0**	21:	**0**
04:	**0**	13:	**0**	22:	**0**
05:	**4**	14:	**0**	23:	**0**
06:	**0**	15:	**0**	24:	**0**
07:	**0**	16:	**0**	25:	**0**
08:	**0**	17:	**0**	26:	**0**
09:	**0**	18:	**0**	27:	**0**

Output statistics

In/Out ratio	**1.00**	Maximum distance	**? nm**
Bandwidth (KB/s):	**0**		
Distinct vessels	**45**		

www.AISHub.net
AIS data sharing center

(C) Astra Paging Ltd
http://www.astrapaging.com

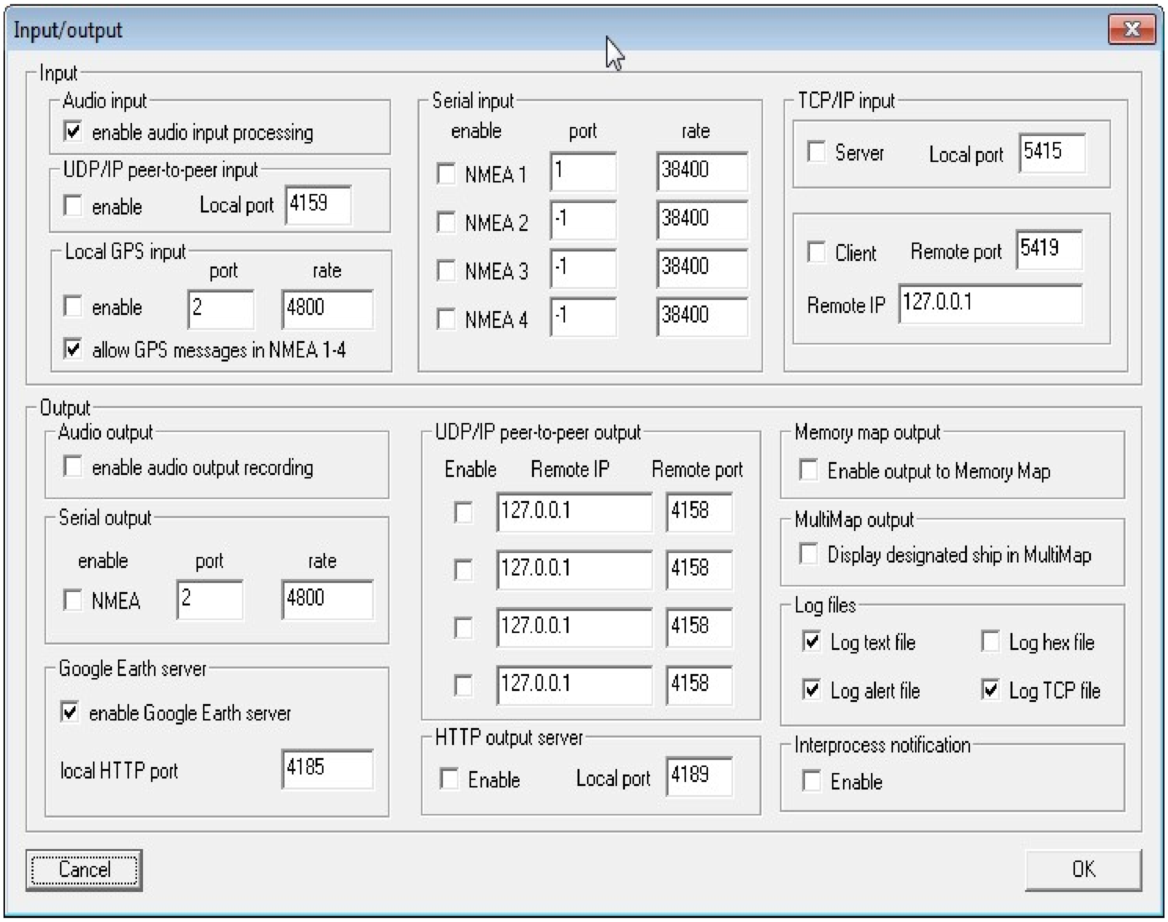
6. Back in the AISDispatcher folder open the AISDispatcher.kml file in Google Earth to view the ships on the globe.

SHIPPLOTTER TUTORIAL

ShipPlotter is a software tool that can decode and plot the location data stored in AIS signals. ShipPlotter is commercial software and costs 25 euros for personal use, but has a 21-day trial. ShipPlotter is only capable of listening to one AIS channel at a time, but could also be used with an external decoder like AISMon. Instructions on using ShipPlotter are shown below.

1. Download and install ShipPlotter from http://www.coaa.co.uk/shipplotter.htm.
2. Open ShipPlotter. Go to **Options->Audio->SoundCard** and select your audio piping method.

3. Under **Options->I/O Settings**, ensure that enable audio input processing is checked.



Input/output

Input

Audio input
☑ enable audio input processing

UDP/IP peer-to-peer input
□ enable Local port 4159

Local GPS input
□ enable port 2 rate 4800
☑ allow GPS messages in NMEA 1-4

Serial input

enable	port	rate
□ NMEA 1	1	38400
□ NMEA 2	-1	38400
□ NMEA 3	-1	38400
□ NMEA 4	-1	38400

TCP/IP input
□ Server Local port 5415
□ Client Remote port 5419
Remote IP 127.0.0.1

Output

Audio output
□ enable audio output recording

Serial output
□ NMEA port 2 rate 4800

Google Earth server
☑ enable Google Earth server
local HTTP port 4185

UDP/IP peer-to-peer output

Enable	Remote IP	Remote port
□	127.0.0.1	4158
□	127.0.0.1	4158
□	127.0.0.1	4158
□	127.0.0.1	4158

HTTP output server
□ Enable Local port 4189

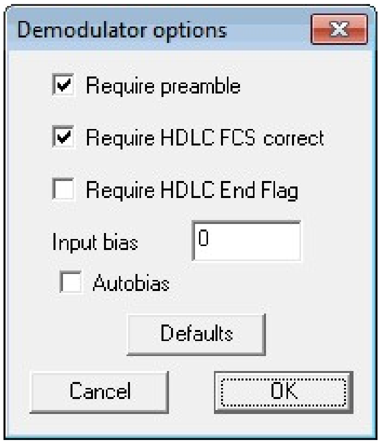
Memory map output
□ Enable output to Memory Map

MultiMap output
□ Display designated ship in MultiMap

Log files
☑ Log text file □ Log hex file
☑ Log alert file ☑ Log TCP file

Interprocess notification
□ Enable

4. Ensure the Demodulator options in **Options->Demodulator** are set to the default values, with ‘Require Preamble’ and ‘Require HDLC FCS correct’ both checked.

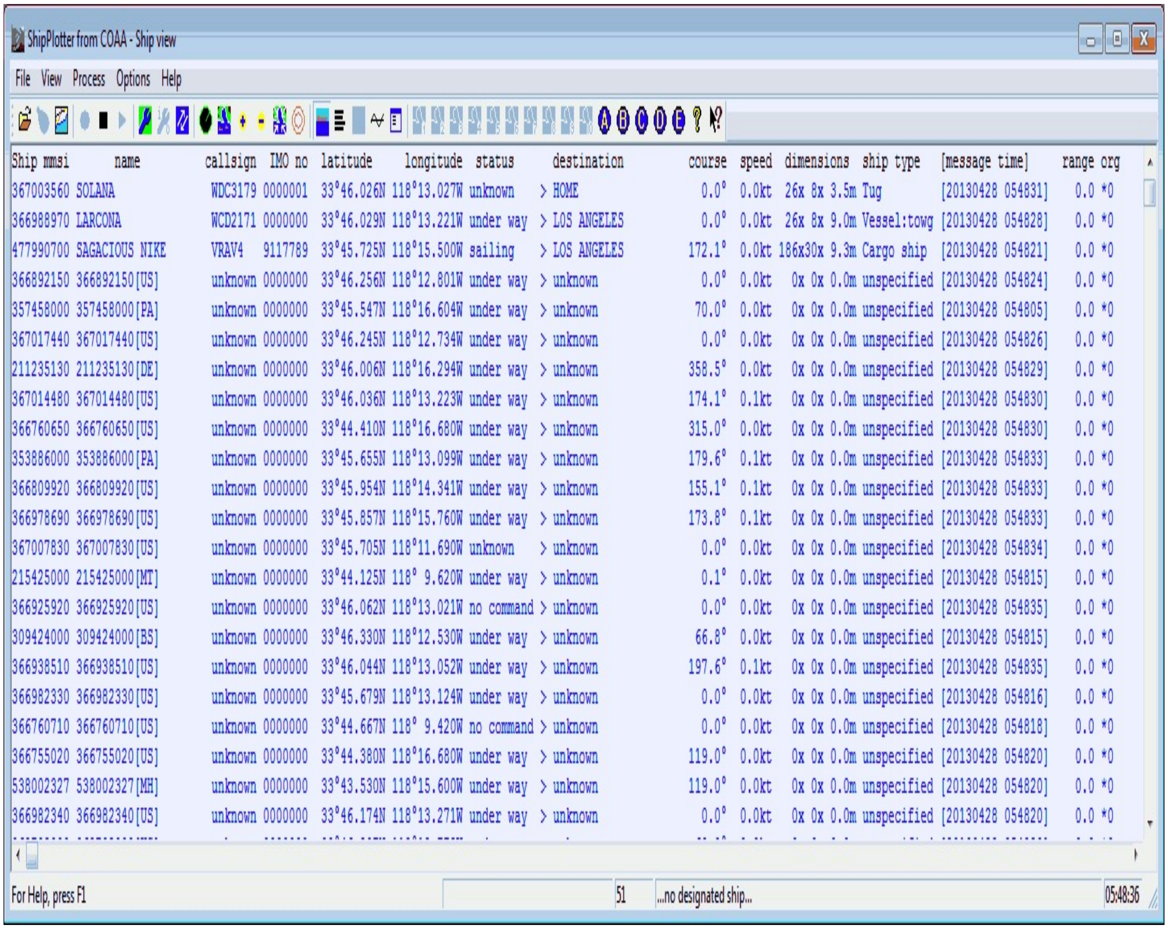


Demodulator options

☑ Require preamble
☑ Require HDLC FCS correct
□ Require HDLC End Flag
Input bias 0
□ Autobias

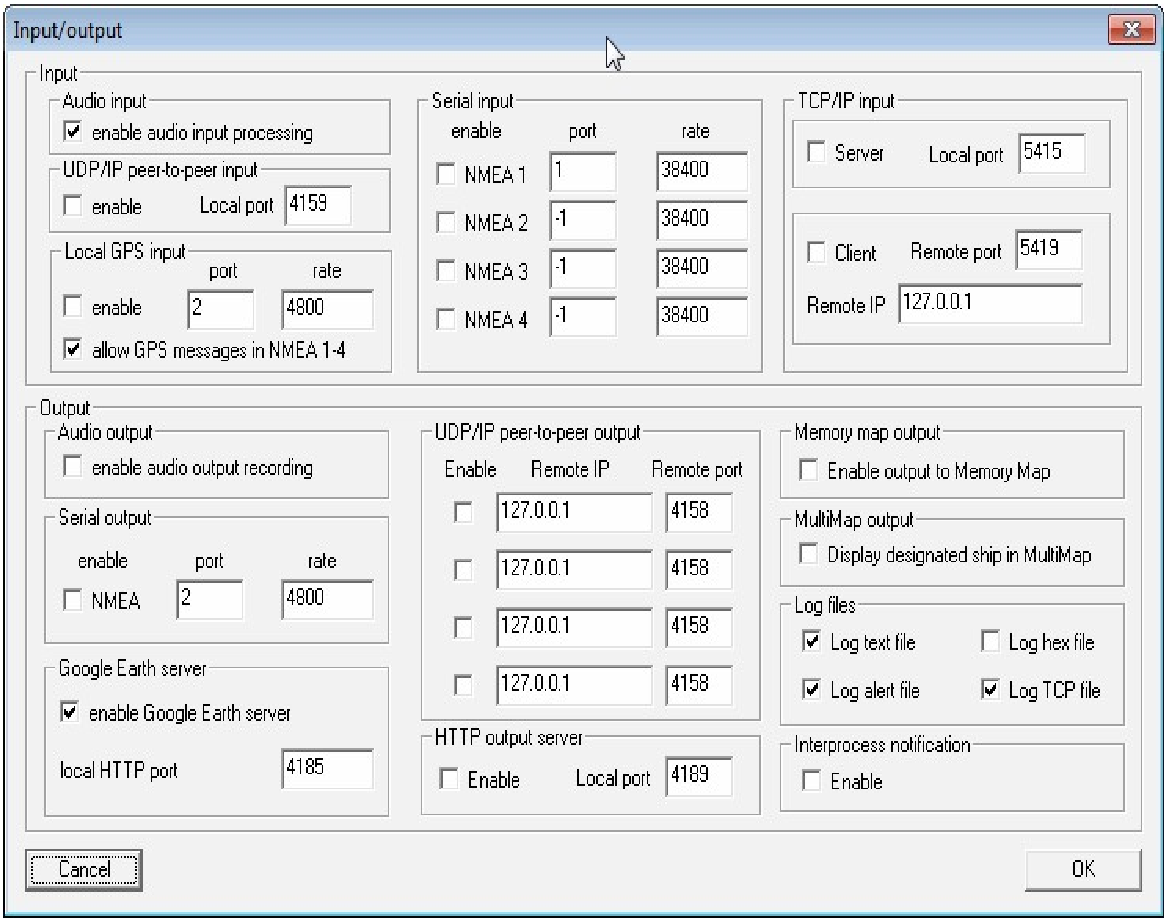
5. Push the green ‘Start’ button ●, and then click on the ‘Raw’ icon ∿ which looks like a horizontal line with a squiggle going through it. This will show a waveform of the input audio. Ensure the audio levels are adequate and not clipping, by making sure the waveform peaks at about halfway on the graph by adjusting the AF gain volume settings in SDR#, or the windows volume settings.

6. Now you should be able to click on the ‘Ships’ and ‘Messages’ icons on the toolbar to see the decoded AIS information. To see ships visually, you will need to follow the ShipPlotter instructions for downloading charts.


ShipPlotter from COAA - Ship view
File View Process Options Help
For Help, press F1
51
...no designated ship...
05:48:36

Ship mmsi	name	callsign	IMO no	latitude	longitude	status	destination	course	speed	dimensions	ship type	[message time]	range	org
367003560	SOLANA	WDC3179	0000001	33°46.026N	118°13.027W	unknown	> HOME	0.0°	0.0kt	26x 8x 3.5m	Tug	[20130428 054831]	0.0	*0
366988970	LARCONA	WCD2171	0000000	33°46.029N	118°13.221W	under way	> LOS ANGELES	0.0°	0.0kt	26x 8x 9.0m	Vessel:towg	[20130428 054828]	0.0	*0
477990700	SAGACIOUS NIKE	VRAV4	9117789	33°45.725N	118°15.500W	sailing	> LOS ANGELES	172.1°	0.0kt	186x30x 9.3m	Cargo ship	[20130428 054821]	0.0	*0
366892150	366892150[US]	unknown	0000000	33°46.256N	118°12.801W	under way	> unknown	0.0°	0.0kt	0x 0x 0.0m	unspecified	[20130428 054824]	0.0	*0
357458000	357458000[PA]	unknown	0000000	33°45.547N	118°16.604W	under way	> unknown	70.0°	0.0kt	0x 0x 0.0m	unspecified	[20130428 054805]	0.0	*0
367017440	367017440[US]	unknown	0000000	33°46.245N	118°12.734W	under way	> unknown	0.0°	0.0kt	0x 0x 0.0m	unspecified	[20130428 054826]	0.0	*0
211235130	211235130[DE]	unknown	0000000	33°46.006N	118°16.294W	under way	> unknown	358.5°	0.0kt	0x 0x 0.0m	unspecified	[20130428 054829]	0.0	*0
367014480	367014480[US]	unknown	0000000	33°46.036N	118°13.223W	under way	> unknown	174.1°	0.1kt	0x 0x 0.0m	unspecified	[20130428 054830]	0.0	*0
366760650	366760650[US]	unknown	0000000	33°44.410N	118°16.680W	under way	> unknown	315.0°	0.0kt	0x 0x 0.0m	unspecified	[20130428 054830]	0.0	*0
353886000	353886000[PA]	unknown	0000000	33°45.655N	118°13.099W	under way	> unknown	179.6°	0.1kt	0x 0x 0.0m	unspecified	[20130428 054833]	0.0	*0
366809920	366809920[US]	unknown	0000000	33°45.954N	118°14.341W	under way	> unknown	155.1°	0.1kt	0x 0x 0.0m	unspecified	[20130428 054833]	0.0	*0
366978690	366978690[US]	unknown	0000000	33°45.857N	118°15.760W	under way	> unknown	173.8°	0.1kt	0x 0x 0.0m	unspecified	[20130428 054833]	0.0	*0
367007830	367007830[US]	unknown	0000000	33°45.705N	118°11.690W	unknown	> unknown	0.0°	0.0kt	0x 0x 0.0m	unspecified	[20130428 054834]	0.0	*0
215425000	215425000[MT]	unknown	0000000	33°44.125N	118° 9.620W	under way	> unknown	0.1°	0.0kt	0x 0x 0.0m	unspecified	[20130428 054815]	0.0	*0
366925920	366925920[US]	unknown	0000000	33°46.062N	118°13.021W	no command	> unknown	0.0°	0.0kt	0x 0x 0.0m	unspecified	[20130428 054835]	0.0	*0
309424000	309424000[BS]	unknown	0000000	33°46.330N	118°12.530W	under way	> unknown	66.8°	0.0kt	0x 0x 0.0m	unspecified	[20130428 054815]	0.0	*0
366938510	366938510[US]	unknown	0000000	33°46.044N	118°13.052W	under way	> unknown	197.6°	0.1kt	0x 0x 0.0m	unspecified	[20130428 054835]	0.0	*0
366982330	366982330[US]	unknown	0000000	33°45.679N	118°13.124W	under way	> unknown	0.0°	0.0kt	0x 0x 0.0m	unspecified	[20130428 054816]	0.0	*0
366760710	366760710[US]	unknown	0000000	33°44.667N	118° 9.420W	no command	> unknown	0.0°	0.0kt	0x 0x 0.0m	unspecified	[20130428 054818]	0.0	*0
366755020	366755020[US]	unknown	0000000	33°44.380N	118°16.680W	under way	> unknown	119.0°	0.0kt	0x 0x 0.0m	unspecified	[20130428 054820]	0.0	*0
538002327	538002327[MH]	unknown	0000000	33°43.530N	118°15.600W	under way	> unknown	119.0°	0.0kt	0x 0x 0.0m	unspecified	[20130428 054820]	0.0	*0
366982340	366982340[US]	unknown	0000000	33°46.174N	118°13.271W	under way	> unknown	0.0°	0.0kt	0x 0x 0.0m	unspecified	[20130428 054820]	0.0	*0

7. An easier method than downloading charts is to display the ships in Google Earth. To get ShipPlotter to work with Google Earth, you must first enable the Google Earth server in **Options->I/O Settings** by ensuring the checkbox next to “enable Google Earth server” in the lower left is checked. The HTTP port can be left as default.



Input/output

Input

Audio input
☑ enable audio input processing

UDP/IP peer-to-peer input
☐ enable Local port 4159

Local GPS input
port rate
☐ enable 2 4800
☑ allow GPS messages in NMEA 1-4

Serial input

enable	port	rate
☐ NMEA 1	1	38400
☐ NMEA 2	-1	38400
☐ NMEA 3	-1	38400
☐ NMEA 4	-1	38400

TCP/IP input
☐ Server Local port 5415
☐ Client Remote port 5419
Remote IP 127.0.0.1

Output

Audio output
☐ enable audio output recording

Serial output
enable port rate
☐ NMEA 2 4800

Google Earth server
☑ enable Google Earth server
local HTTP port 4185

UDP/IP peer-to-peer output

Enable	Remote IP	Remote port
☐	127.0.0.1	4158
☐	127.0.0.1	4158
☐	127.0.0.1	4158
☐	127.0.0.1	4158

HTTP output server
☐ Enable Local port 4189

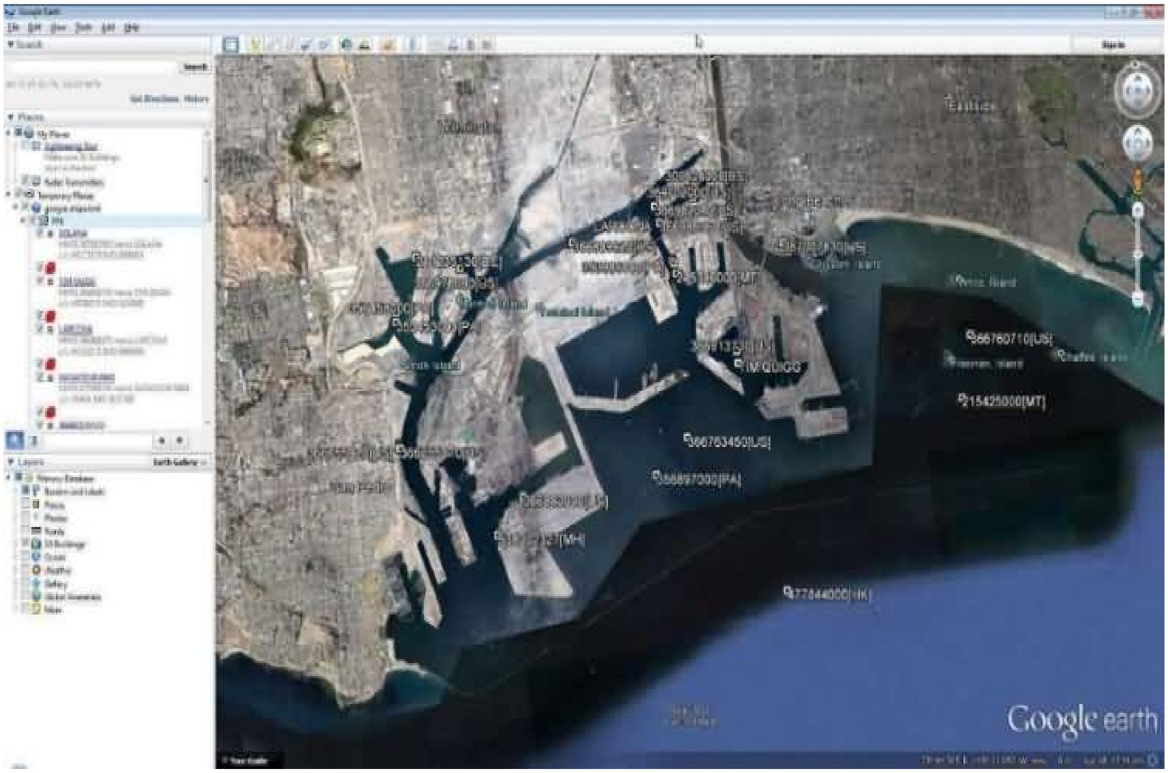
Memory map output
☐ Enable output to Memory Map

MultiMap output
☐ Display designated ship in MultiMap

Log files
☑ Log text file ☐ Log hex file
☑ Log alert file ☑ Log TCP file

Interprocess notification
☐ Enable

8. Then you can go to the folder ShipPlotter was installed to (most likely in ‘Program Files (x86)/COAA/ShipPlotter’) and open the google_ships.kml file in Google Earth. Note that you will need to open ShipPlotter and begin decoding AIS signals by pressing the green start button BEFORE opening the google_ships.kml file, otherwise ships will not show up.



GR-AIS

GR-AIS is a GNU Radio based AIS decoder which can be downloaded from https://github.com/bistromath/gr-ais.

AISDECODER

AISDecoder is a Linux and Windows compatible AIS decoder which can output NMEA positions for use in OpenCPN. Download AISDecoder from the forum thread at http://forum.aishub.net/ais-decoder/ais-decoder-beta-release/.

To use on Windows a precompiled binary file is provided in the thread. After downloading AISDecoder, start up SDR# or any other general purpose SDR receiver and tune to an AIS channel. Next open a command prompt and open aisdecoder using the following line:

```
aisdecoder -h 127.0.0.1 -p 10110 -a winmm -l
```

This will start aisdecoder with it outputing data to 127.0.0.1 on port 10101. Aisdecder will listen to your default audio device so make sure SDR# is outputing to the correct device and it is set as the default in Windows sound recording properties. Adjust the volume levels until aisdecoder reads the volume at around 50-70%.

To use on Linux, first compile the program with the following:

```
tar zxvf aisdecoder.tar.gz
cd aisdecoder
mkdir build
cd build
cmake ../ -DCMAKE_BUILD_TYPE=Release
make
```

Now open two terminal windows or tabs. In one terminal window create a FIFO buffer with the following command.

```
mkfifo /tmp/aisdata
```

In the same terminal window use the following which will tune to an AIS channel and use sox to convert the sample rate into 48 kHz which is required by aisdecoder. It then outputs the raw audio to the FIFO. Remember to set any PPM offset required for rtl_fm by using the `-p` flag and you may need to fine tune the gain setting with `-g`.

```
rtl_fm -M fm -f161.975M -s12k -p48 -g50 | sox -r12k -traw -es -b16 - -traw -r48k -es -b16 /tmp/aisdata
```

In the second terminal window run the command

```
./aisdecoder -h 127.0.0.1 -p 10110 -a file -c mono -d -l -f /tmp/aisdata
```

Now in OpenCPN you can add a UDP receiver with port 10110.

To test an AIS example wav file use the following.

```
sox LALBSnippet.wav -t wav - | sox -twav - -traw -r48k -eunsigned-integer -b16 /tmp/aisdata
```

As an alternative to using a FIFO buffer you can also directly output the audio to the speakers and make aisdecoder listen to that.

```
rtl_fm -M fm -f 161.975M -s 12k -g 50 -l 0 | sox -r12k -traw -es -b16 - -traw -r48k -es -b16 -c1 - | play -r 48k -t raw -e s -b 16 -c 1 -V1 -
```

And then run the following to make aisdecoder listen to the audio using pulse audio.

```
./aisdecoder -h 127.0.0.1 -p 10110 -apulse -c mono -d -l
```

PNAIS

For Windows there is also the PNAIS program available from https://sites.google.com/site/f4eyuradio/ais-decoder. PNAIS will directly connect to an RTL-SDR dongle and will output NMEA data via UDP so it can easily connect to OpenCPN as shown in the AISMon tutorial.

AISREC

At the time of writing this (Dec 2014) the full version of AISRec has not been released. AISRec is a two channel Windows AIS decoder that directly connects to the RTL-SDR and outputs data via UDP for use with software like OpenCPN just like the other decoders. Currently only a trial version is available that will decode one AIS channel and will time out after 20 minutes or 5000 messages. We are unsure how much the full version will cost. In our tests with the trial version AISRec showed itself to be a very good decoder as it was able to spot many more ships compared to any other decoder shown in the above sections.

AISRec can be downloaded by going to their demo YouTube video at https://www.youtube.com/watch?v=bm8UctAjpjw and using the download links in the description. Don’t be put off by the download site which is in Chinese, as it is fairly obvious on how to download the software (hint: after entering the code, click on the button with the downwards pointing arrow).

Note that we discovered that the software doesn’t use a PPM correction setting as expected. Instead it uses a frequency shift setting. To set the shift in the AISRec.ini file, we had to calculate freqshift = 162.025 MHz – frequency of the second AIS channel as shown in SDR# with no PPM correction set.

MARINETRAFFIC.COM

Another possible method for displaying AISMon data is to share your UDP data to the marinetraffic.com website and view the ships on their shared map. You can do this by simply using the marinetraffic.com IP address to send the UDP traffic to. Further instructions can be found at http://www.marinetraffic.com/ais/addyourarea.aspx?level1=150#6. Note, be careful that you do not send delayed AIS data to marinetraffic.com, such as with the sample AIS file from the AISMon Yahoo group.

REFERENCES AND ADDITIONAL LINKS

Hak5 Tutorial Video on AIS: https://www.youtube.com/watch?v=-ZznkOfVivo&feature=youtube_gdata

DECODING VHF DATA LINK MODE 2 (VDL2)

INTRODUCTION TO VDL2

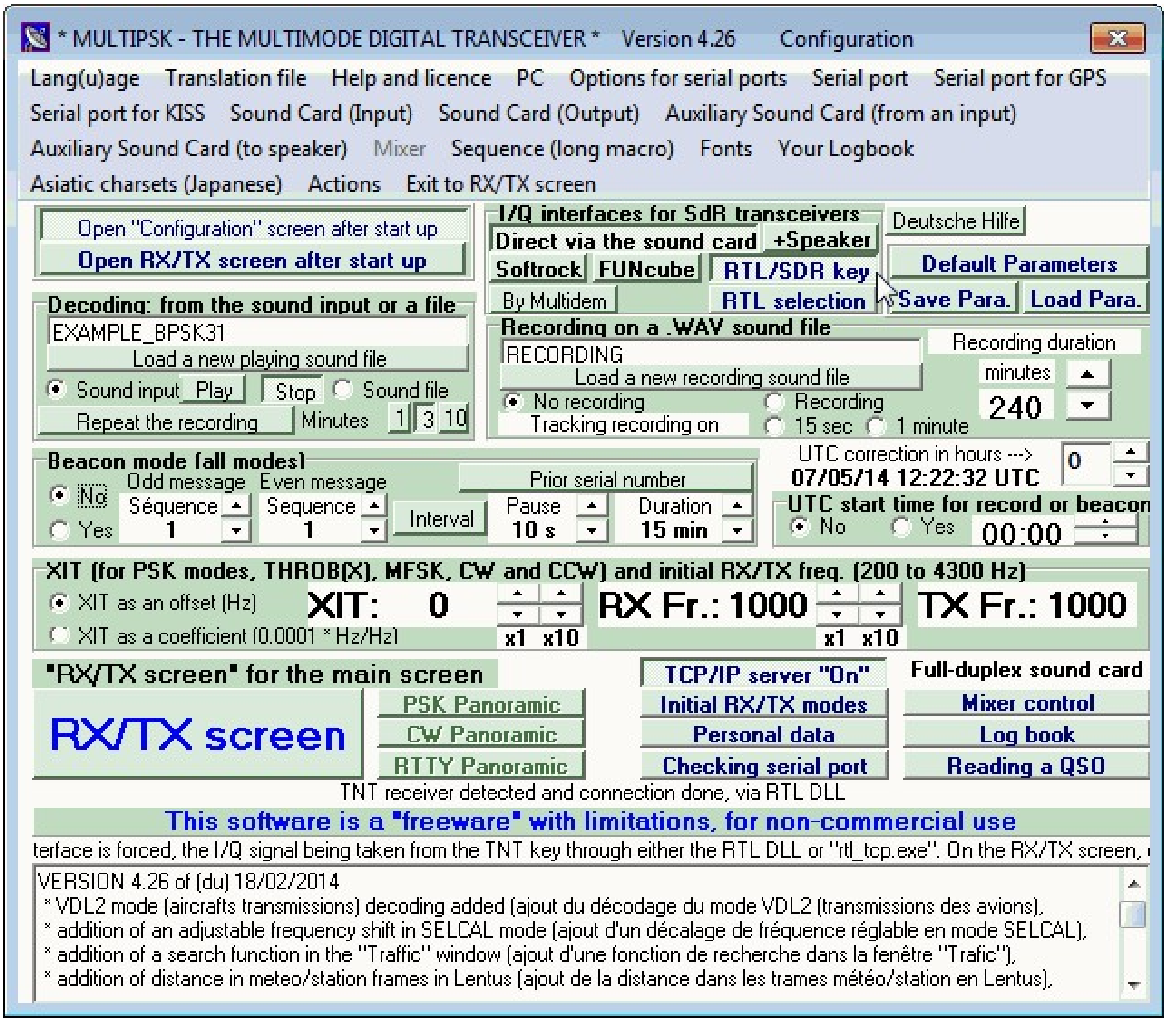
The VHF Data Link mode 2 (VDL2) is a new transmission mode used on aircraft for sending short messages, position data (similar to ADS-B) and also for allowing traffic controllers to communicate to pilots via text and data. VDL2 is intended to eventually replace ACARS. It is found at a worldwide frequency of 136.975 MHz.

MultiPSK can be used to decode VDL2 and then its data can be sent to PlanePlotter for plotting plane positions if GPS data is sent.

DECODING VDL2 WITH MULTIPSK

MultiPSK is a sophisticated multi mode decoder that can connect directly to the RTL-SDR and is capable of decoding VDL2 signals. It is free, but to decode "professional" modes such as VDL2 the paid version is required. However, MultiPSK provides 5 minutes of VDL2 use per opening of the program for testing. To use MultiPSK follow these steps.

1. Download and extract the multiPSK zip file to a folder.
2. Open MULTIPSK.exe
3. You will be greeted with a complicated looking screen. On this screen we need to ensure the RTL/SDR key button (near the top middle) is pressed in.



EXAMPLE_BPSK31

RECORDING

240

UTC correction in hours ---> 0

07/05/14 12:22:32 UTC

Séquence 1 Sequence 1 Pause 10 s Duration 15 min

UTC start time for record or beacon 00:00

XIT: 0 RX Fr.: 1000 TX Fr.: 1000

TNT receiver detected and connection done, via RTL DLL

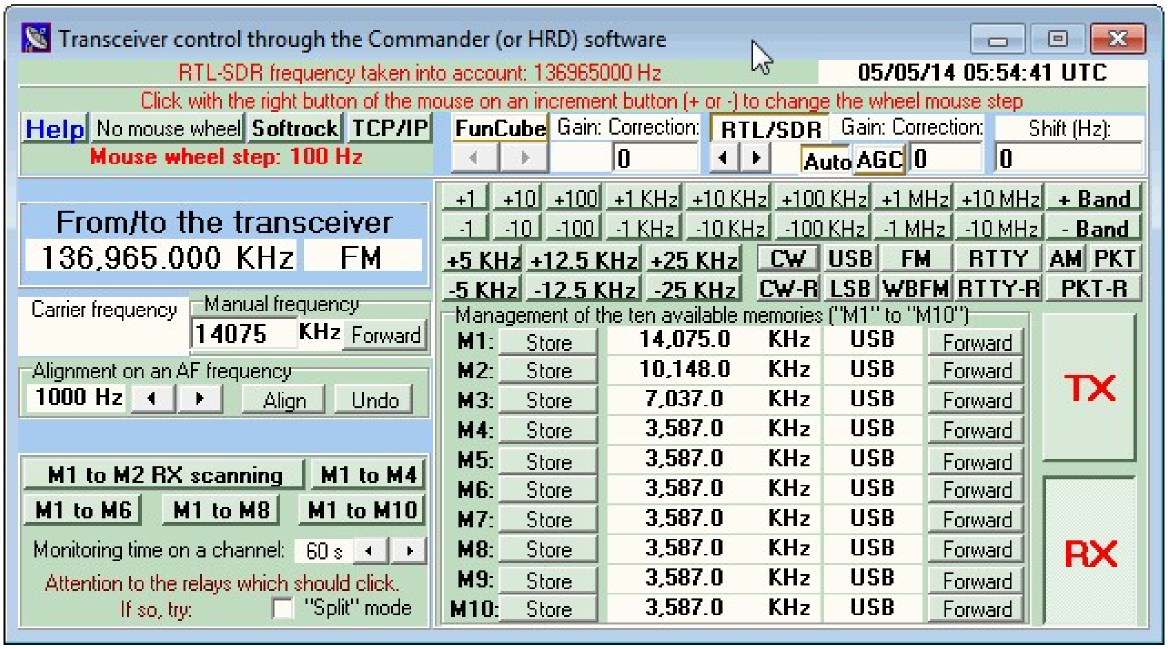
This software is a "freeware" with limitations, for non-commercial use

terface is forced, the I/Q signal being taken from the TNT key through either the RTL DLL or "rtl_tcp.exe". On the RX/TX screen,

VERSION 4.26 of (du) 18/02/2014
* VDL2 mode (aircrafts transmissions) decoding added (ajout du décodage du mode VDL2 (transmissions des avions),
* addition of an adjustable frequency shift in SELCAL mode (ajout d'un décalage de fréquence réglable en mode SELCAL),
* addition of a search function in the "Traffic" window (ajout d'une fonction de recherche dans la fenêtre "Trafic"),
* addition of distance in meteo/station frames in Lentus (ajout de la distance dans les trames météo/station en Lentus),

4. Click on the RX/TX screen button in the bottom left. This will start the RTL-SDR and a waterfall and RF spectrum should show.

5. Click on the Transceiver button Transceiver in the top left area of the main screen. This will bring up the transceiver control screen. Here you can adjust the RTL-SDR gain and set the PPM correction which is shown near the in the top right of the transceiver screen. If you are unsure as to what to set the gain, use the Auto Gain by ensuring the Auto button is pressed. You can enter a PPM correction in the text box under the word “correction”. Close the transceiver screen.



Transceiver control through the Commander (or HRD) software

RTL-SDR frequency taken into account: 136965000 Hz

05/05/14 05:54:41 UTC

Click with the right button of the mouse on an increment button (+ or -) to change the wheel mouse step

Mouse wheel step: 100 Hz

FunCube Gain: Correction: 0

RTL/SDR Gain: Correction: 0

Shift (Hz): 0

From/to the transceiver

136,965.000 KHz FM

Carrier frequency

Manual frequency 14075 KHz Forward

Alignment on an AF frequency 1000 Hz

Management of the ten available memories ("M1" to "M10")

M1:	Store	14,075.0	KHz	USB	Forward
M2:	Store	10,148.0	KHz	USB	Forward
M3:	Store	7,037.0	KHz	USB	Forward
M4:	Store	3,587.0	KHz	USB	Forward
M5:	Store	3,587.0	KHz	USB	Forward
M6:	Store	3,587.0	KHz	USB	Forward
M7:	Store	3,587.0	KHz	USB	Forward
M8:	Store	3,587.0	KHz	USB	Forward
M9:	Store	3,587.0	KHz	USB	Forward
M10:	Store	3,587.0	KHz	USB	Forward

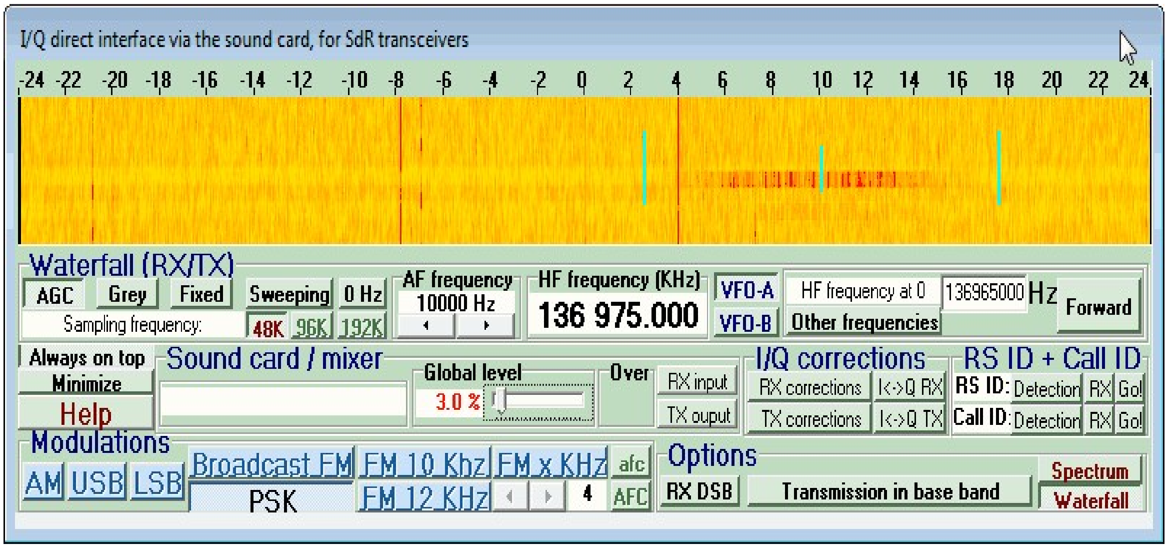
Monitoring time on a channel: 60 s

Attention to the relays which should click.

If so, try: □ "Split" mode

6. In the floating window with the waterfall/spectrum change the HF frequency at 0 to 136965000 Hz (note *not* 136975000 Hz) and then press the "Forward Button".

7. Set the "AF frequency" to 10000 Hz. Do this by first clicking under the number 10 in the waterfall and then fine tuning it using the left and right buttons under the AF frequency controls.



I/Q direct interface via the sound card, for SdR transceivers

Sampling frequency: 48K

AF frequency 10000 Hz

HF frequency (KHz) 136 975.000

HF frequency at 0 136965000 Hz

Global level 3.0 %

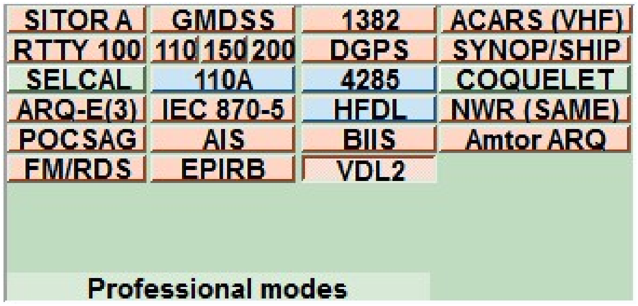
RS ID: Detection

Call ID: Detection

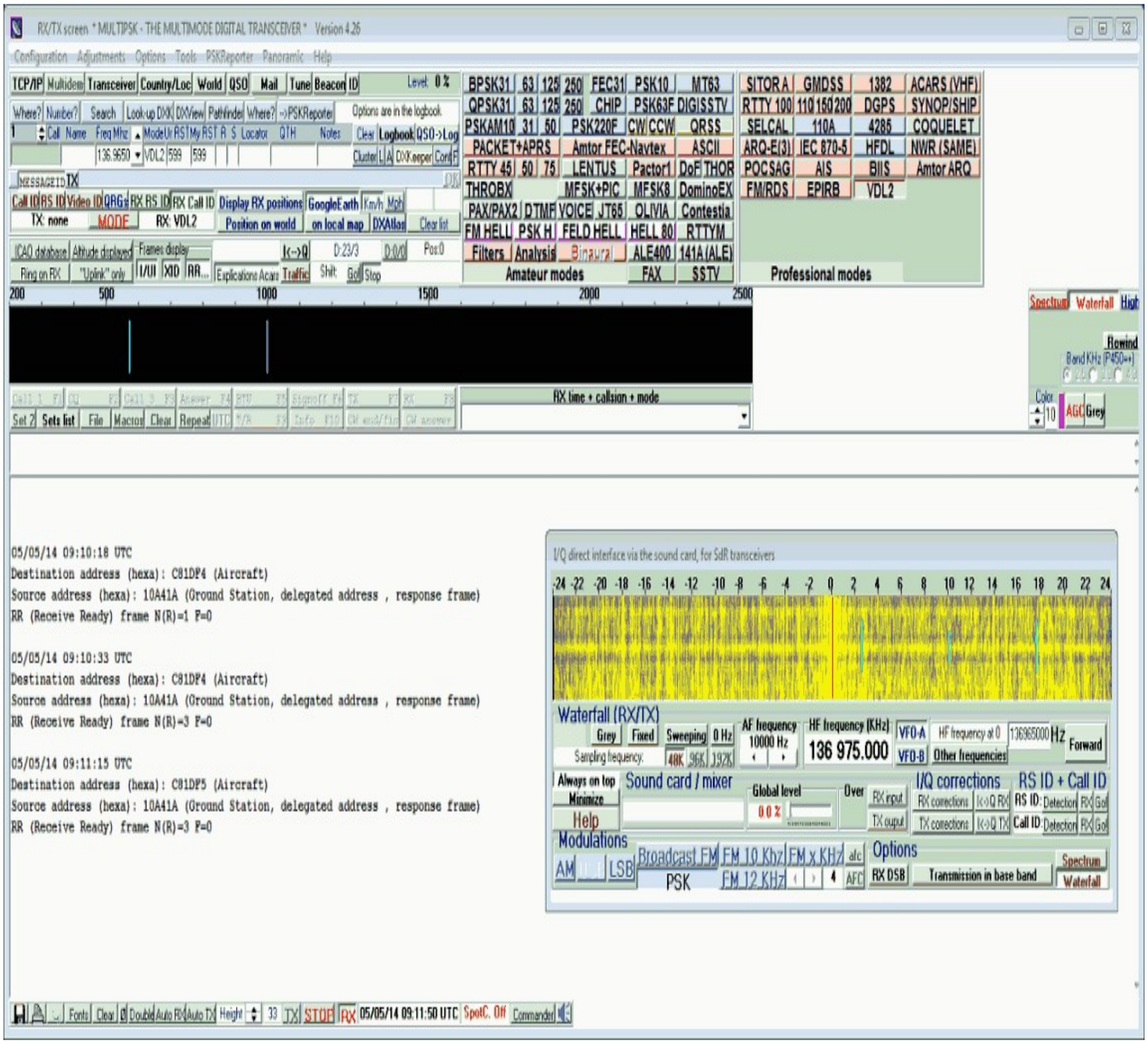
4

8. Now click on the VDL2 button on the very right of the main window, where it says Professional modes. If you can't see this you might need to maximise

the main window.



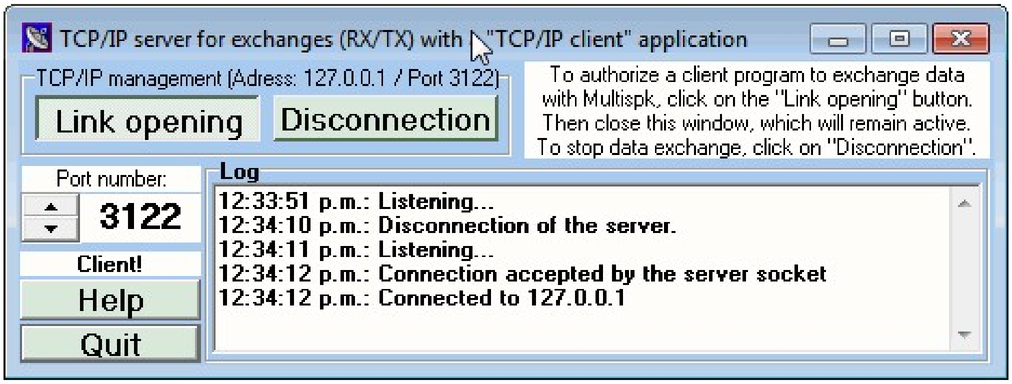
9. Now VDL2 decoding will begin and you should see VDL2 messages in the main window when a message is received.


RX/TX screen * MULTIPSK - THE MULTIMODE DIGITAL TRANSCEIVER * Version 4.26
Configuration Adjustments Options Tools PSKReporter Panoramic Help
RX: VDL2
Amateur modes
Professional modes
I/Q direct interface via the sound card, for SdR transceivers
Waterfall (RX/TX)
HF frequency (KHz)
136 975.000
Sound card / mixer
I/Q corrections
RS ID + Call ID
Modulations
Options
05/05/14 09:11:50 UTC

```
05/05/14 09:10:18 UTC
Destination address (hexa): C81DF4 (Aircraft)
Source address (hexa): 10A41A (Ground Station, delegated address , response frame)
RR (Receive Ready) frame N(R)=1 F=0

05/05/14 09:10:33 UTC
Destination address (hexa): C81DF4 (Aircraft)
Source address (hexa): 10A41A (Ground Station, delegated address , response frame)
RR (Receive Ready) frame N(R)=3 F=0

05/05/14 09:11:15 UTC
Destination address (hexa): C81DF5 (Aircraft)
Source address (hexa): 10A41A (Ground Station, delegated address , response frame)
RR (Receive Ready) frame N(R)=3 F=0
```

MultiPSK can also interface with PlanePlotter in order to plot VDL2 data on a map, if coordinates are transmitted. To do this follow these instructions.

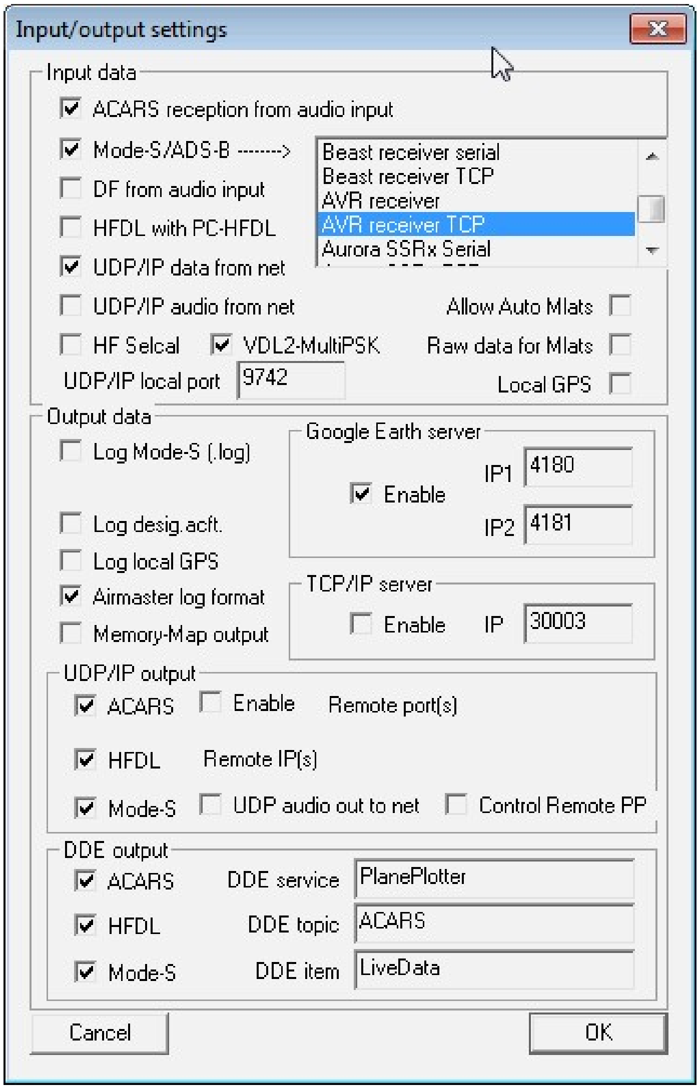
1. In the main MultiPSK window click the TCP/IP button TCP/IP in the top left corner. Make sure the port number is set to 3122.


TCP/IP server for exchanges (RX/TX) with "TCP/IP client" application
TCP/IP management (Adress: 127.0.0.1 / Port 3122)
Link opening
Disconnection
Port number:
3122
Client!
Help
Quit
Log

To authorize a client program to exchange data with Multispk, click on the "Link opening" button. Then close this window, which will remain active. To stop data exchange, click on "Disconnection".

```
12:33:51 p.m.: Listening...
12:34:10 p.m.: Disconnection of the server.
12:34:11 p.m.: Listening...
12:34:12 p.m.: Connection accepted by the server socket
12:34:12 p.m.: Connected to 127.0.0.1
```

2. Now in PlanePlotter go to I/O options 🔧 and ensure that the checkbox VDL2-MultiPSK is checked.



Input/output settings

Input data

☑ ACARS reception from audio input

☑ Mode-S/ADS-B -------> Beast receiver serial / Beast receiver TCP / AVR receiver / **AVR receiver TCP** / Aurora SSRx Serial

☐ DF from audio input

☐ HFDL with PC-HFDL

☑ UDP/IP data from net

☐ UDP/IP audio from net Allow Auto Mlats ☐

☐ HF Selcal ☑ VDL2-MultiPSK Raw data for Mlats ☐

UDP/IP local port 9742 Local GPS ☐

Output data

☐ Log Mode-S (.log)

☐ Log desig.acft.

☐ Log local GPS

☑ Airmaster log format

☐ Memory-Map output

Google Earth server: ☑ Enable IP1 4180 IP2 4181

TCP/IP server: ☐ Enable IP 30003

UDP/IP output

☑ ACARS ☐ Enable Remote port(s)

☑ HFDL Remote IP(s)

☑ Mode-S ☐ UDP audio out to net ☐ Control Remote PP

DDE output

☑ ACARS DDE service PlanePlotter

☑ HFDL DDE topic ACARS

☑ Mode-S DDE item LiveData

Cancel OK

3. Start PlanePlotter by pressing the green start button. Now any messages received by MultiPSK will appear in PlanePlotter as well.

You can also use the free program VDL2-Display available from http://www.rstools.info/downloads-2.html which will display the received VDL2 information in a clearer format. To use VDL2-Display follow the instructions below.

1. In MultiPSK click on the TCP/IP button TCP/IP in the top left corner of the receive screen. Change the port number to 3125. Disconnect and then reconnect the connection.

2. Open Display-Launcher.exe then open VDL2 Display.

3. Click the connect the Connect #1 button in VDL2 Display to connect to MultiPSK. Now received VDL2 messages and aircraft will show in VDL2 Display.

REFERENCES AND ADDITIONAL LINKS

PlanePlotter MultiPSK VDL2 Tutorial:
http://planeplotter.pbworks.com/w/page/75894179/VDL%20mode%202

AIR TRAFFIC CONTROL / SCANNING GUIDE

Air traffic control voice channels can be easily scanned using the RTL-SDR and a scanner plugin for SDR#. A scanner plugin rapidly switches between frequencies looking for an active signal. When an active signal is found it stays on that signal until the transmission is over. This tutorial can be adapted to any radio system that requires scanning.

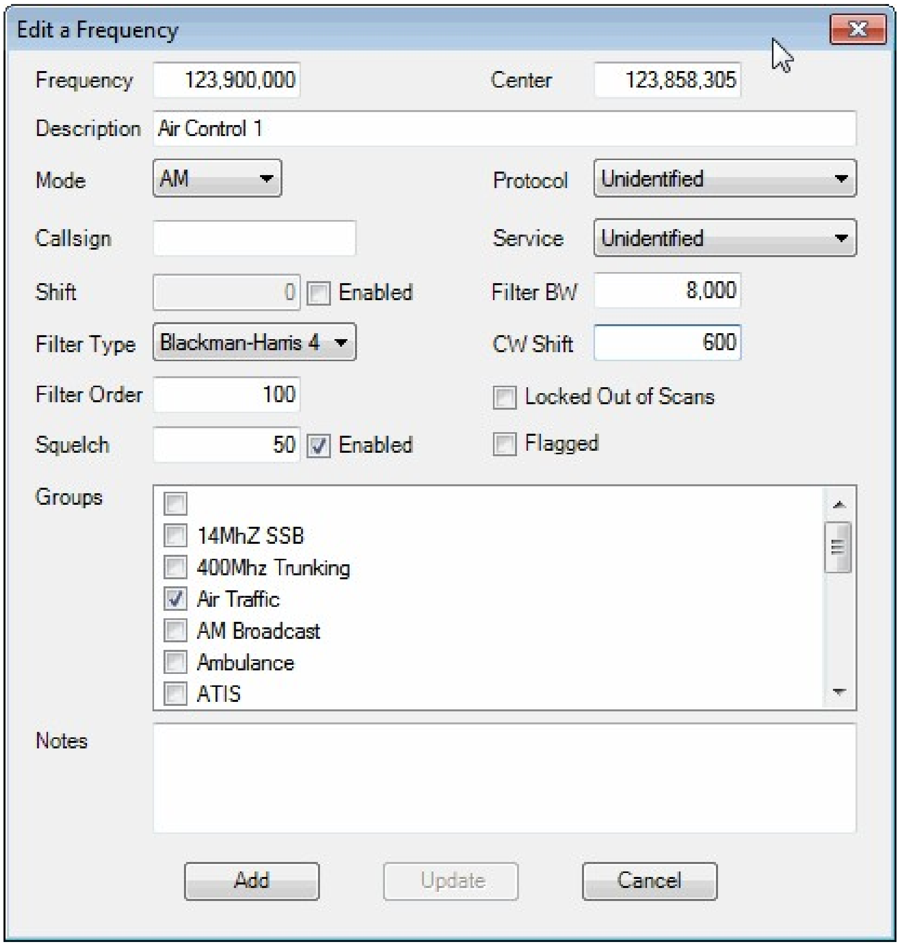
SDR# FREQUENCY SCANNER

The SDR# frequency scanner plugin we will use can be downloaded from www.sdrsharpplugins.com. It has an automatic installer so there is no need to edit any config files. Simply extract the zip file to a folder and then run PluginsSetup.exe making sure to follow the on screen instructions.

KNOWN FREQUENCY SCANNING

This will quickly scan a list of known frequencies for an active signal. You will need to find a list of frequencies used by Air Traffic control in your area. You can also manually search for active frequencies in the spectrum. Another way may be to create a heatmap or use the scanner metrics plugin to find active frequencies. See the heatmap tutorial in a later section.

1. In the frequency manager plugin tab, go to **Manage -> Groups** and create a group called Air Traffic.
2. Now to add your frequencies first tune to that frequency, then click the Edit button. In the window that pops up make sure to set the Mode to AM for air traffic and the Filter BW to 8000. Put a checkmark next to the Air Traffic group box. Finally, click Add.



Edit a Frequency

Frequency 123,900,000 Center 123,858,305

Description Air Control 1

Mode AM Protocol Unidentified

Callsign Service Unidentified

Shift 0 □ Enabled Filter BW 8,000

Filter Type Blackman-Harris 4 CW Shift 600

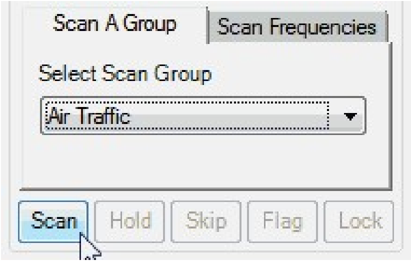
Filter Order 100 □ Locked Out of Scans

Squelch 50 ☑ Enabled □ Flagged

Groups
□
□ 14MhZ SSB
□ 400Mhz Trunking
☑ Air Traffic
□ AM Broadcast
□ Ambulance
□ ATIS

Notes

3. To scan a group, first select that group in the Scan A Group tab. Then click the Scan button. The “minimum signal strength” slider should be set higher than the noise floor, but not too high to miss weak signals. Remember that a value of 0 indicates a very strong signal and a value of -130 indicates a very weak signal. You may need to tweak the signal strength slider until only valid signals and not noise is recognised.



Select Scan Group
Air Traffic

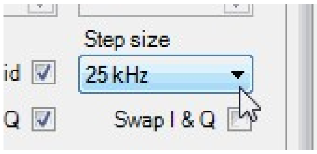
You can edit the scan speed by clicking on the spanner icon and then adjusting the “Radio Settle Time in Milliseconds” slider under the Performance tab.

GENERAL FREQUENCY SCANNING

General frequency scanning is when you want to scan very quickly over a range of frequencies to find unknown active signals. While the frequency scanner plugin used above can do this, for this type of scanning we prefer to use the "Fast Scanner Plugin", from http://rtl-sdr.ru/page/skaner-poisk-novyh-chastot which we find to work better. Note that this page is in Russian, but the download link is placed near the bottom of the article and the plugin is in English. Install the plugin according to the plugin install procedure.

To use the fast scanner plugin follow these steps.

1. Before starting to scan you will need to know the spacing of frequencies in the band you want to scan. For example, air band signals are spaced at 25 kHz (or 8.33 kHz in some countries) apart. You will need to set this spacing as your step size using the setting in the Radio Tab in SDR# which is the tab containing the AM/FM/SSB mode selections. The scanner will increment the frequency using this signal spacing.



Step size: 25 kHz

2. Go to the fast scanner frequency plugin window. Choose either "screen" or "manual" mode in the drop down menu. Screen mode will scan for active frequencies that are currently displayed on the screen and manual mode will let you scan between an arbitrary minimum and maximum frequency.

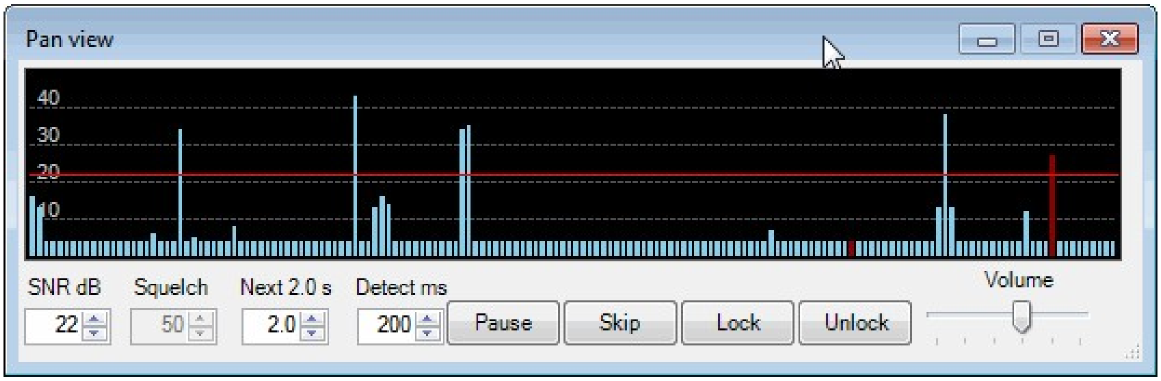
3. Press Scan. A "Pan View" scanner window will pop up. This window shows a frequency spectrum of signals.

4. The horizontal red line indicates the power threshold a signal needs to cross to be classed as an active signal that the scanner will stop at. The height of the line can be adjusted using the SNR dB setting box.

5. You can press the "Skip" button to skip to the next active signal when scanning.

6. You can ignore certain signals such as noise spikes or data channels by clicking on them with your mouse in the pan view frequency spectrum, or by dragging the mouse in the pan view frequency spectrum and using the Lock

and Unlock buttons. When a frequency in the frequency spectrum is coloured red it is ‘locked’ and will be ignored by the scanner.



While scanning the fast scanner will also create a table of active signals that it has found in the plugin window. It shows the time that they have been active and the frequency they are at.

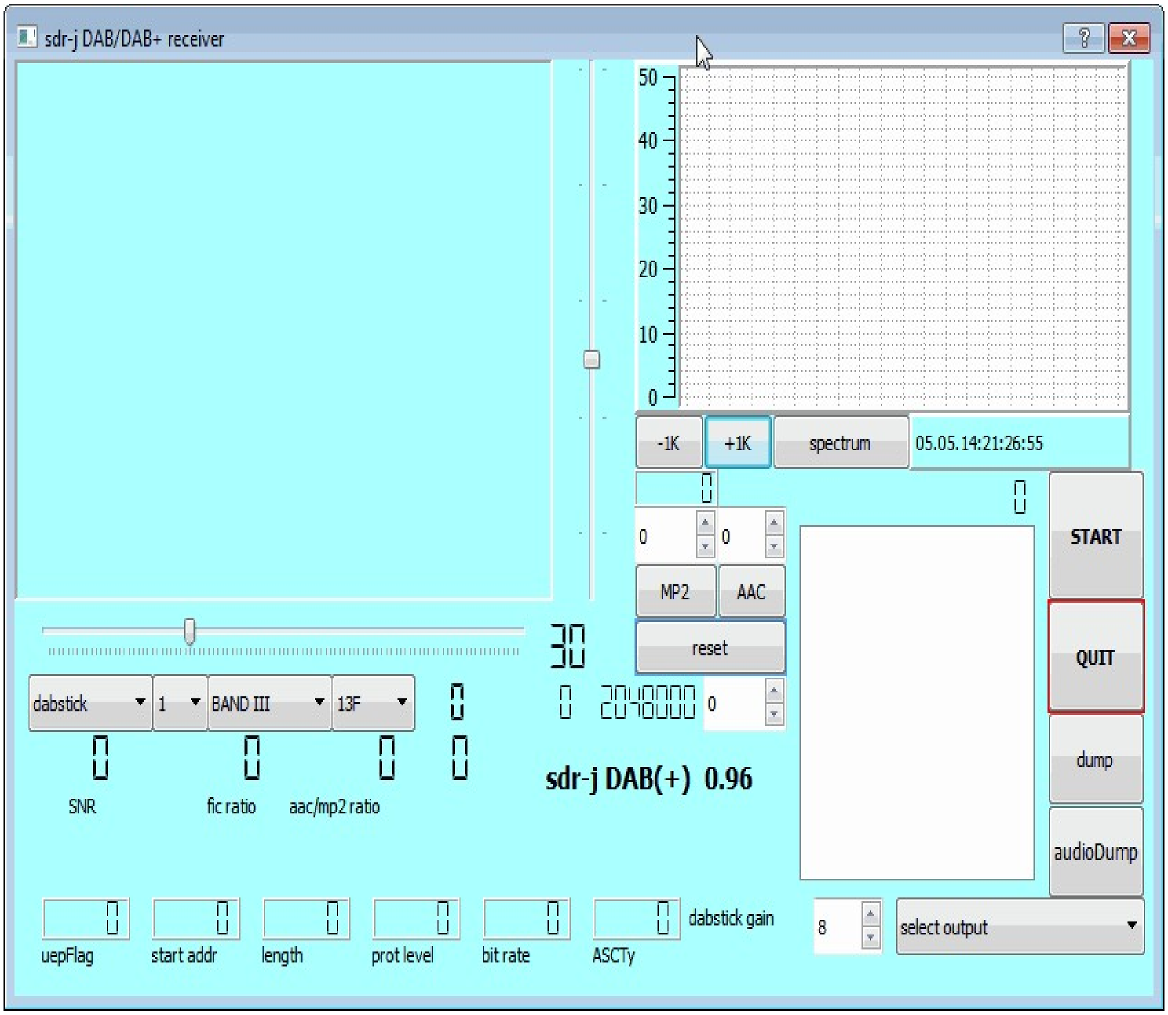
DIGITAL AUDIO BROADCASTING (DAB) RADIO GUIDE

Digital Audio Broadcasting (DAB) is a digital radio technology used to broadcast commercial radio stations. It is an alternative to broadcast wideband FM stations that is used in many countries, especially Europe. The DVB-T TV software from the CD provided with some dongles actually has DAB decoding software, however this software requires the installation of the official drivers which would prevent any general SDR use.

DAB RECEIVE TUTORIAL WITH SDR-J

Instead, to listen to DAB radio with the SDR drivers installed, a program called SDR-J can be used. SDR-J is a Windows and Linux program that is capable of receiving DAB and DAB+. It also comes with an FM receiver, however this won't be used in our tutorial. SDR-J can be downloaded from http://www.sdr-j.tk/. Make sure to download the **dabstick-radio** zip file.

1. Download SDR-J and extract the zip file to a folder.
2. Open the dabreceiver.exe file.
3. If your RTL-SDR dongle is plugged in and properly set up the SDR-J DAB screen will show.



dabstick
1
BAND III
13F
dabstick gain 8

4. Make sure that “dabstick” is selected on the left-most pull down menu.

5. Choose the BAND and DAB spectrum block designator in the two right-most pull down boxes on the left of the screen. In the screenshot above we used BAND III and the block 13F. These settings will be different for each country. Check out http://www.wohnort.org/dab/ for information on block designations in your country.

6. In the bottom right select the audio output you wish to use, which will usually be your speakers.

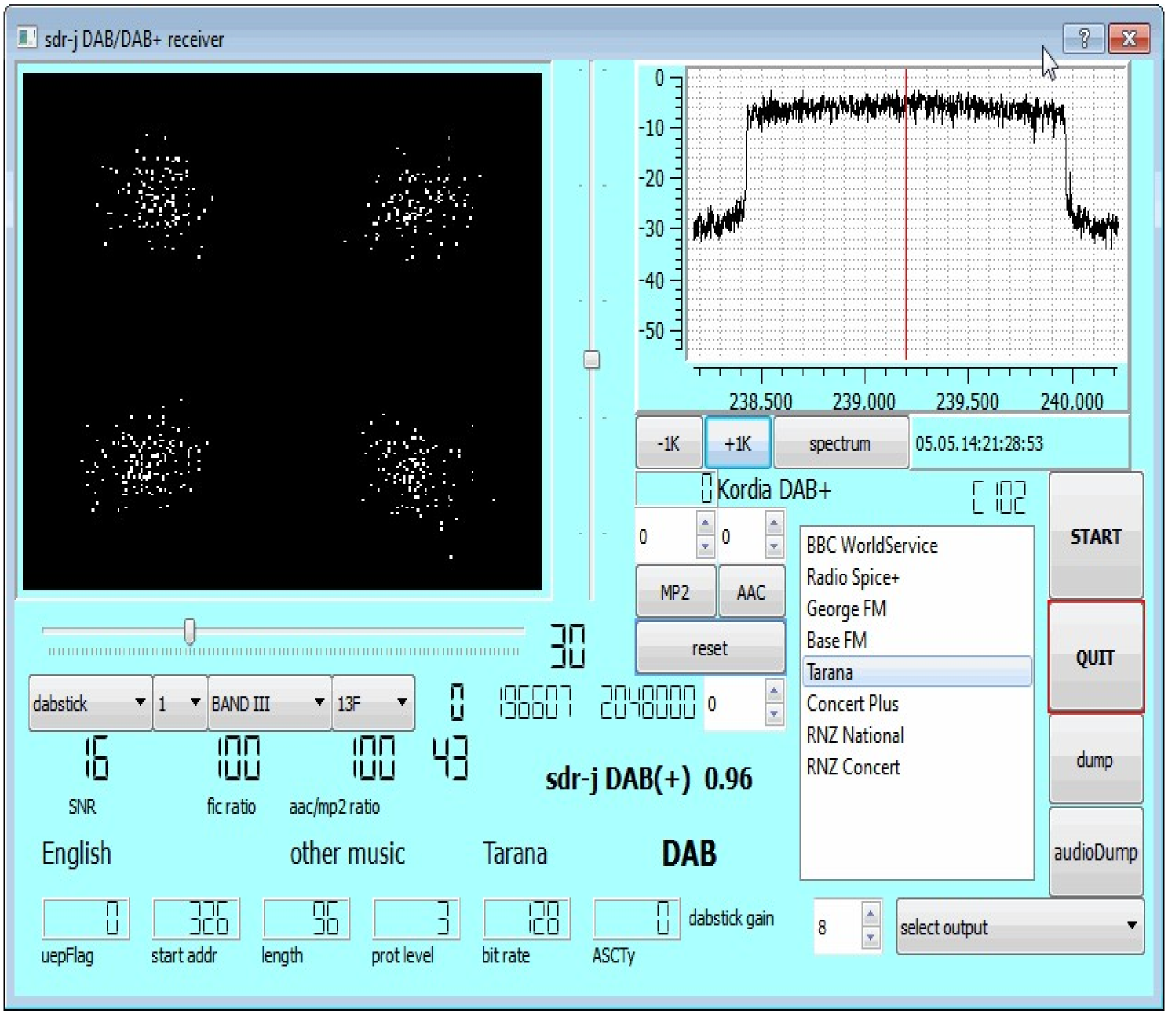
7. Click Start.

8. Adjust the gain in the bottom right corner so that the DAB RF signal in the top right is strong and high above the noise floor. You can use the vertical slider to the left of the RF signal to adjust the left axis range. If the signal is strong the constellation diagram on the left should look like four clusters of

dots. If the dots are not clustering and are just random, then the signal is not strong enough.

9. At this point if you have chosen a DAB block which is broadcasting, the white box on the right side will begin to populate with radio stations.

10. Click on a radio station to begin playing it. Note that sometimes changing the number box above the AAC button works better to select radio stations.



sdr-j DAB/DAB+ receiver

dabstick, 1, BAND III, 13F

0 156607 2048000 0

30

Kordia DAB+ C102

05.05.14:21:28:53

- BBC WorldService
- Radio Spice+
- George FM
- Base FM
- Tarana
- Concert Plus
- RNZ National
- RNZ Concert

SNR 16, fic ratio 100, aac/mp2 ratio 100, 43

sdr-j DAB(+) 0.96

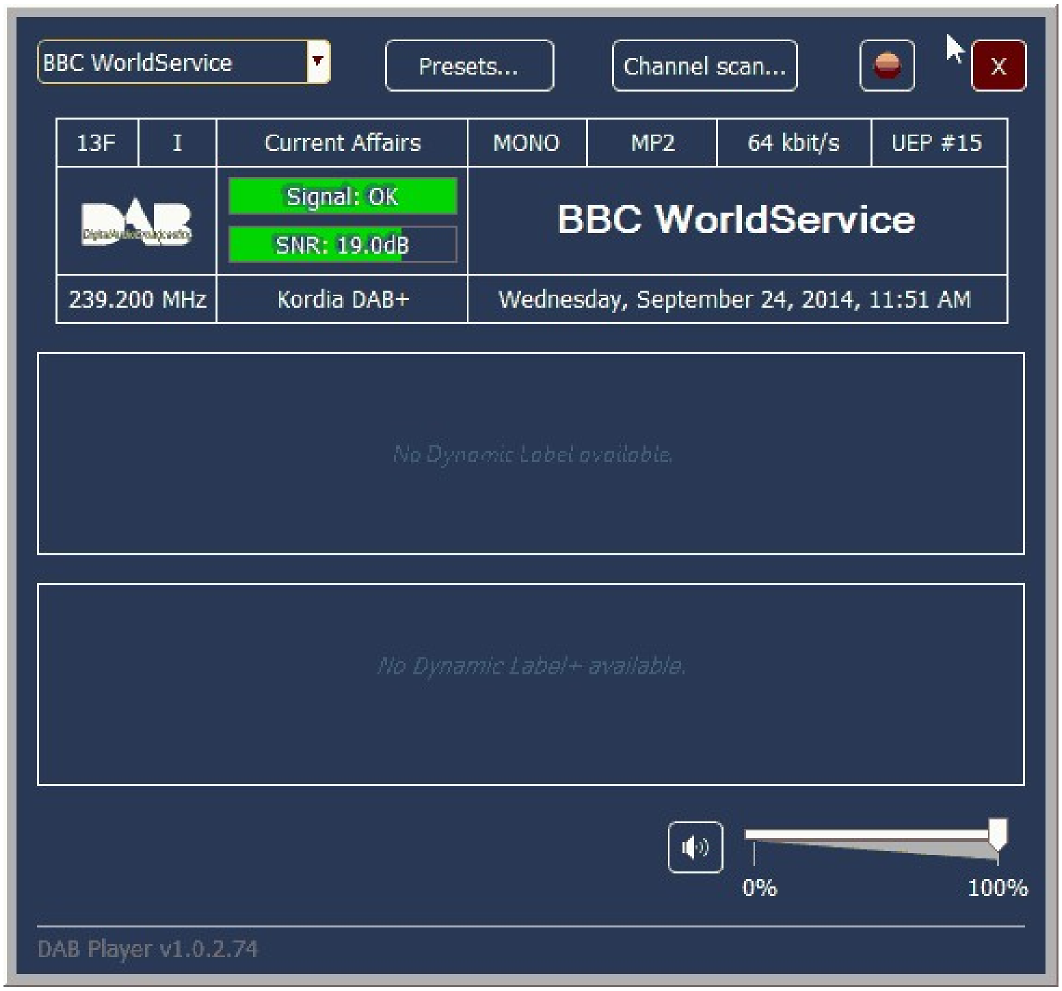
English, other music, Tarana, DAB

uepFlag 0, start addr 326, length 96, prot level 3, bit rate 128, ASCTy 0, dabstick gain 8

DAB RECEIVE TUTORIAL WITH DAB PLAYER

An alternative DAB player called "DAB Player by Andreas Gsinn) that uses the official manufacturer RTL2832U drivers can be downloaded from http://www.ukwtv.de/cms/downloads-aside/281-dab-player-von-andreas-gsinn.html. Note that this page is in German so just download the zip file with the words "DAB_Player" in it, and unzip the files into a folder on your PC. If you don't have the official drivers they can be downloaded from http://download.buytra.com/driver/SPC-0155-Driver.zip (Mirror: http://bit.ly/1igDacW). See the DVB-T on Windows section for more information about using the official drivers. The DAB Player website also provides a link to the official drivers for the R820T in the Treiber2.zip download. Since one of the RTL-SDR's official purpose was for listening to DAB radio, DAB Player seems to work a little better than SDR-J as it used the official manufacturer drivers. These are different to the SDR drivers and will require that you uninstall the SDR drivers first.

Note that the language of this program is by default in German, but it can be changed to English by right clicking on the top half of the program and then going to **Sprache->en-US English (United States) / English (United States)**. After opening the program go to Channel Scan and choose the DAB channel used in your country. If you are unsure of what channel is used, you can specify start and end channels and the program will scan through them all looking for DAB radio. Once it finds a channel and the scan ends, click on Save and then exit the channel search. Now you can select the DAB radio station in the top left drop down menu.



13F	I	Current Affairs	MONO	MP2	64 kbit/s	UEP #15

Signal: OK

SNR: 19.0dB

BBC WorldService

239.200 MHz	Kordia DAB+	Wednesday, September 24, 2014, 11:51 AM

No Dynamic Label available.

No Dynamic Label+ available.

DAB Player v1.0.2.74

RECEIVING ANALOGUE TV (PAL/NTSC) GUIDE

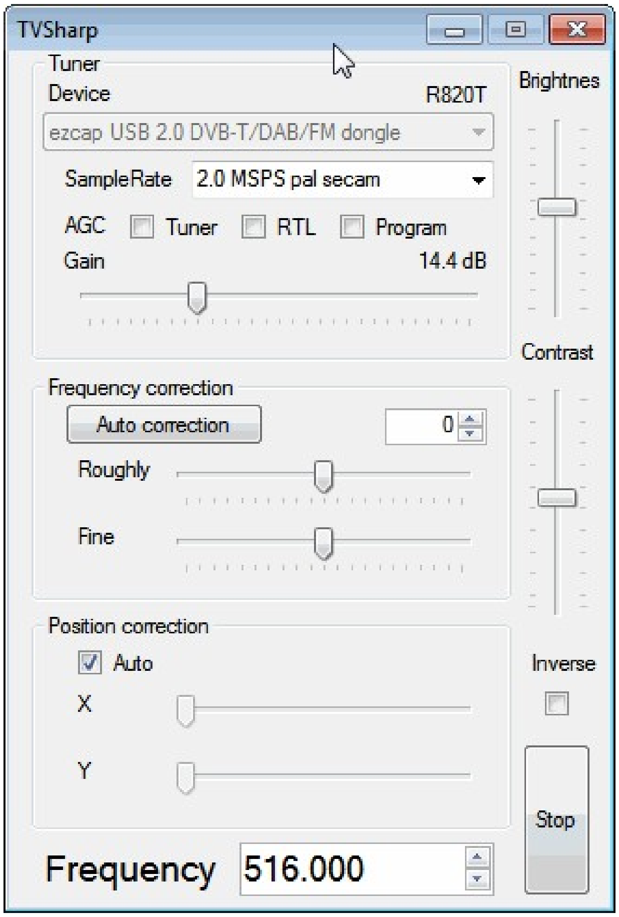
Most countries have begun phasing out transmission of Analogue TV, instead favouring the switch to digital transmission standards like ATSC (USA) or DVB-T (most of world). In the countries that still transmit Analogue TV the RTL-SDR can be used to receive it.

In SDR mode the RTL-SDR is incapable of receiving the entire PAL or NTSC video bandwidth, which is 5 MHz for PAL systems and 4.2 MHz for NTSC. However, it is still capable of receiving part of the luminance (black and white) part of the PAL or NTSC signal.

A program which decodes PAL and NTSC is TVSharp. The latest version of TVSharp can be downloaded from http://rtl-sdr.ru/page/tvsharp-analogovoe-tv-na-rtl-tjunere/tvsharp-updated-version-1-2/ This page is in Russian, so use Google translate or just click on the link at the bottom of the page to download the zip. (Mirror 1: http://sdrts.amoti.ru/download/view.download/4/8) (Mirror 2: http://bit.ly/1hMBZmy).

To use TVSharp simply open it and then click the start button with your RTL-SDR dongle plugged in. Under sample rate choose either the NTSC or PAL option depending on the TV broadcast standard in your country and choose the 2.0 MSPS option which usually works best. Then use the frequency text box to tune to a known analogue TV station. Using a higher sample rate will increase the bandwidth sampled, improving picture quality, but setting it too high seems to cause reception problems. You may need to adjust the gain to get good reception.

The program also has automatic frequency and position correction options built in which will help to prevent image scrolling and non centred images. You may need to wait until the dongle has warmed up and stabilized its frequency offset before you see a stable image.



Device R820T

ezcap USB 2.0 DVB-T/DAB/FM dongle

SampleRate 2.0 MSPS pal secam

AGC □ Tuner □ RTL □ Program

Gain 14.4 dB

0

☑ Auto

Frequency 516.000

There is also an NTSC decoder for GNU Radio available which may be useful if you want to learn about NTSC decoding. The tutorial is written in Japanese, but Google Translate is good enough to get the required information out of it. It can be found at http://bit.ly/XUdEIv.

GUIDE TO LISTENING TO TRUNKED RADIO: ANALOGUE AND DIGITAL P25

In an inefficient radio system, one talkgroup (e.g one company) might use a single frequency for radio communications. However, this is very inefficient as the frequency may not be in use for the majority of the time. It would be more efficient if multiple talkgroups could use the same frequency, but without disrupting other talkgroups while it is in use.

In a trunked radio system, a small set number of voice frequencies are shared between a large number of talkgroups. Each radio constantly receives and transmits to a special computer controlled control channel run by the radio service provider. When the talk button on a radio is pressed, the radio quickly communicates to a remote computer run by the service provider via the control channel and determines a vacant voice frequency that the talkgroup can use. This helps to make radio frequency allocations more efficient.

Because a talkgroup might switch between various voice frequencies often, listening to a conversation can difficult for radio scanners that cannot decode the control channel. The software program Unitrunker can be used to decode the control channel and follow a voice conversation as it hops across various frequencies. With two RTL-SDR dongles you can set up a trunking receiver station. What follows below is a tutorial on how to set this up.

UNITRUNKER VOICE TRUNKING FOLLOWING TUTORIAL

To set up a trunking station we will assume that you have been through the Quickstart guide in this book and thus have a working RTL-SDR setup. In addition you will need:

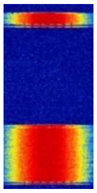
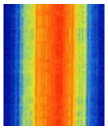
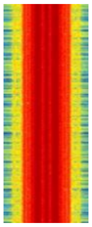
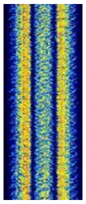
- 2x R820T RTL2832U RTL-SDR dongles. See the Buy RTL-SDR dongles page at http://www.rtl-sdr.com/buy-rtl-sdr-dvb-t-dongles/ if you don’t already have two. You will need one dongle to receive the control channel and one dongle to receive the voice channel. Only the R820T appears to be supported at the moment.
- The Unitrunker software (Download and install the latest version from http://www.unitrunker.com/download)
- One antenna that splits into two connections for the two dongles, or two seperate antennas that can receive the trunking frequencies you are interested

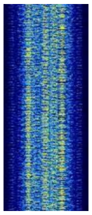
in.

Now follow these steps to set up Unitrunker with the RTL2832U.

SETTING UP THE SIGNAL (CONTROL CHANNEL) RECEIVER

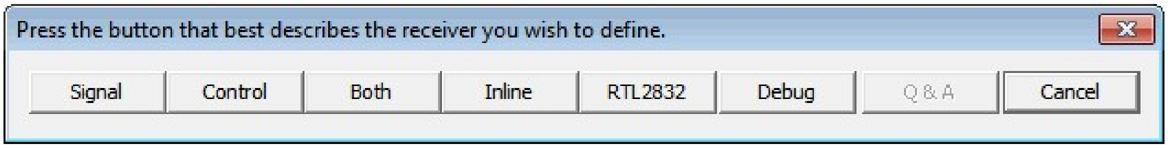
First determine the frequency of a trunking control channel that you are interested in monitoring. You can use SDR# or a SDR receiver program to search for these first. They will be signals that are continuously transmitting. Here are some examples of some common trunking modes. The are commonly found in the 400 and 800 MHz bands, but this may be different depending on your country.

Protocol	Waterfall Image
P25	
DMR/MotoTRBO	
MPT1327	
Motorola Type II Smartnet	

EDACS96	

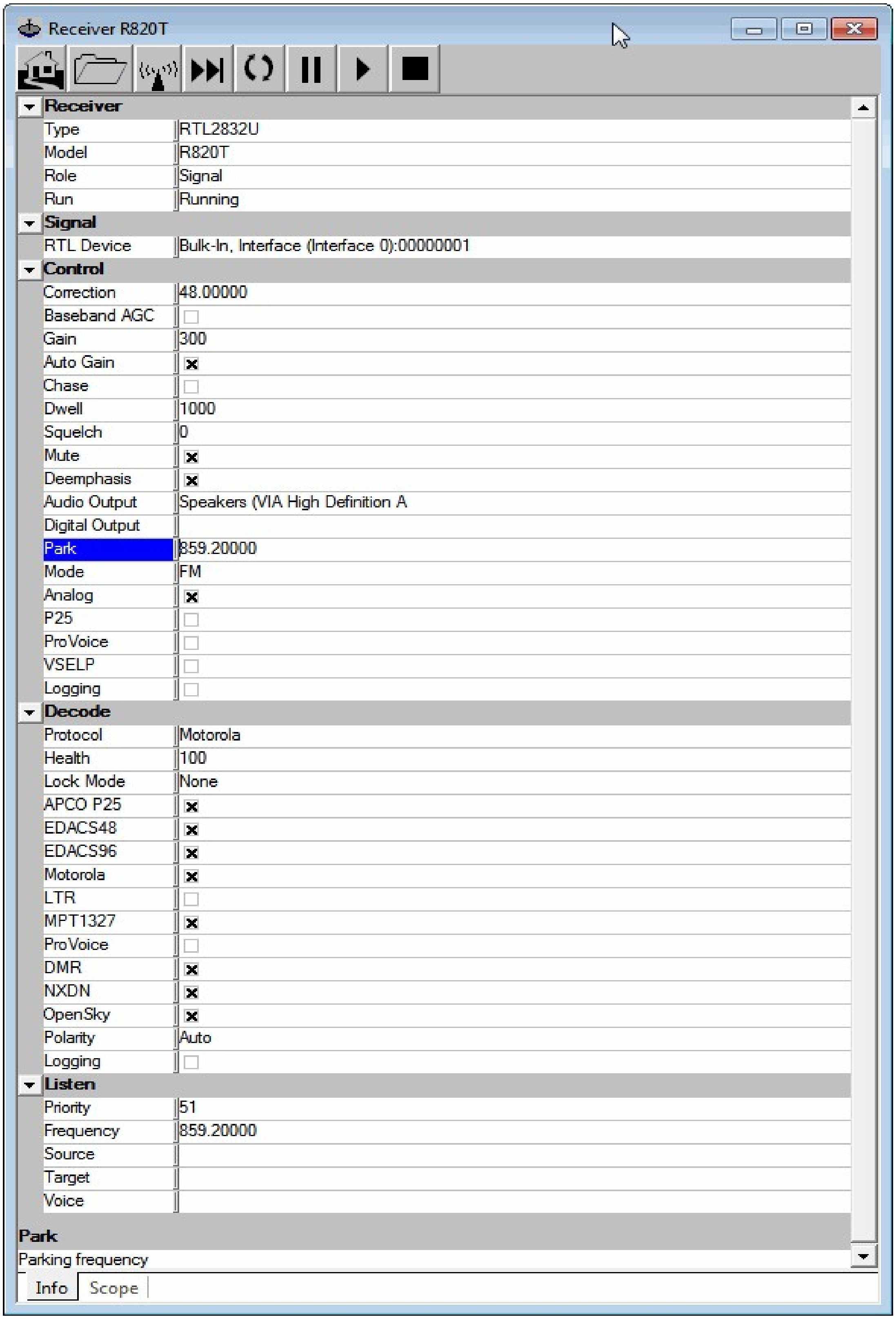
To use Unitrunker follow these instructions:

1. Plug in both dongles and then open Unitrunker.
2. Press the ‘Add new receiver’ button + and then on the bar that pops up click on the RTL2832 button.


Press the button that best describes the receiver you wish to define.

3. In the new window that pops up change the Role to Signal.
4. Under the Signal heading change the RTL Device to the stick that will be used for the control channel.
5. Type in your dongles frequency PPM offset in Correction. Note it is critical that you get the PPM correction correct.
6. Make sure that Mute is checked.
7. Change the Park value to the frequency of the trunking channel that you wish to monitor.
8. Under Decode check all the trunking protocols that you are interested in listening to.
9. Now at the top of this window press the Play button ▶. The Protocol entry under Decode should now show the protocol detected, and a new window with site information should soon pop up and the value of the Protocol setting under the Decode heading should change to the trunking protocol detected.

10. If the window does not pop up, check the Health under the Protocol entry. If it is below 100, either increase the Gain value (can be set to a value 0 - 500) or turn on the Auto Gain setting. If still no new window pops up, set the Audio Output setting to your speakers and uncheck the mute button. Listen to see if the trunking channel is being correctly played through your speakers. If not, you may need to fine tune the frequency with the PPM correction setting or you may be tuned incorrectly.



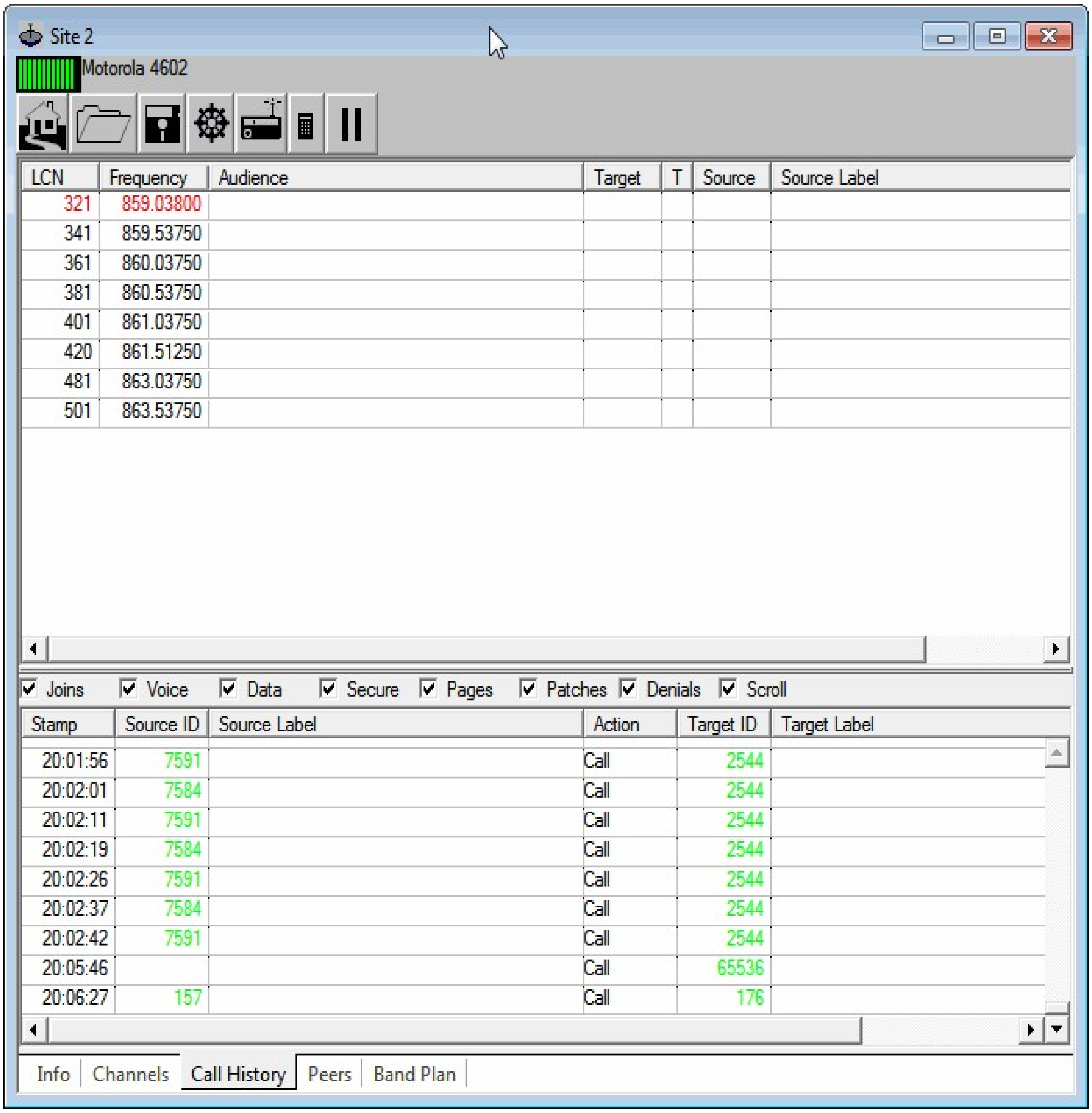
Receiver R820T

Receiver	
Type	RTL2832U
Model	R820T
Role	Signal
Run	Running
Signal	
RTL Device	Bulk-In, Interface (Interface 0):00000001
Control	
Correction	48.00000
Baseband AGC	☐
Gain	300
Auto Gain	☒
Chase	☐
Dwell	1000
Squelch	0
Mute	☒
Deemphasis	☒
Audio Output	Speakers (VIA High Definition A
Digital Output	
Park	859.20000
Mode	FM
Analog	☒
P25	☐
ProVoice	☐
VSELP	☐
Logging	☐
Decode	
Protocol	Motorola
Health	100
Lock Mode	None
APCO P25	☒
EDACS48	☒
EDACS96	☒
Motorola	☒
LTR	☐
MPT1327	☒
ProVoice	☐
DMR	☒
NXDN	☒
OpenSky	☒
Polarity	Auto
Logging	☐
Listen	
Priority	51
Frequency	859.20000
Source	
Target	
Voice	

Park

Parking frequency

11. If the signal is strong enough a new Site window should pop up. This new window shows the frequencies in use, and logs each call. If the frequencies are not showing, click the 'Wizard' button 🖩 whose icon looks like a calculator. Then choose the system used in your country. If unsure, try 'Standard' first.



LCN	Frequency	Audience	Target	T	Source	Source Label
321	859.03800					
341	859.53750					
361	860.03750					
381	860.53750					
401	861.03750					
420	861.51250					
481	863.03750					
501	863.53750					

☑ Joins ☑ Voice ☑ Data ☑ Secure ☑ Pages ☑ Patches ☑ Denials ☑ Scroll

Stamp	Source ID	Source Label	Action	Target ID	Target Label
20:01:56	7591		Call	2544	
20:02:01	7584		Call	2544	
20:02:11	7591		Call	2544	
20:02:19	7584		Call	2544	
20:02:26	7591		Call	2544	
20:02:37	7584		Call	2544	
20:02:42	7591		Call	2544	
20:05:46			Call	65536	
20:06:27	157		Call	176	

12. (Optional) If you have a RadioReference.com account, you can click on the gear icon and enter your RadioReference account creditials. This will download information provided by RadioReference.com, including site information and some talk group labels.

13. Close the Receiver R820T options window for the signal channel and go back to the main Unitrunker Window.

SETTING UP THE VOICE CHANNEL RECEIVER

1. Now that we have added the signal receiver and decoder we will add the voice receiver.

2. Press the ‘Add new receiver’ button + once again and then on the bar that pops up click on the RTL2832 button.

3. This time for the role choose ‘Voice’.

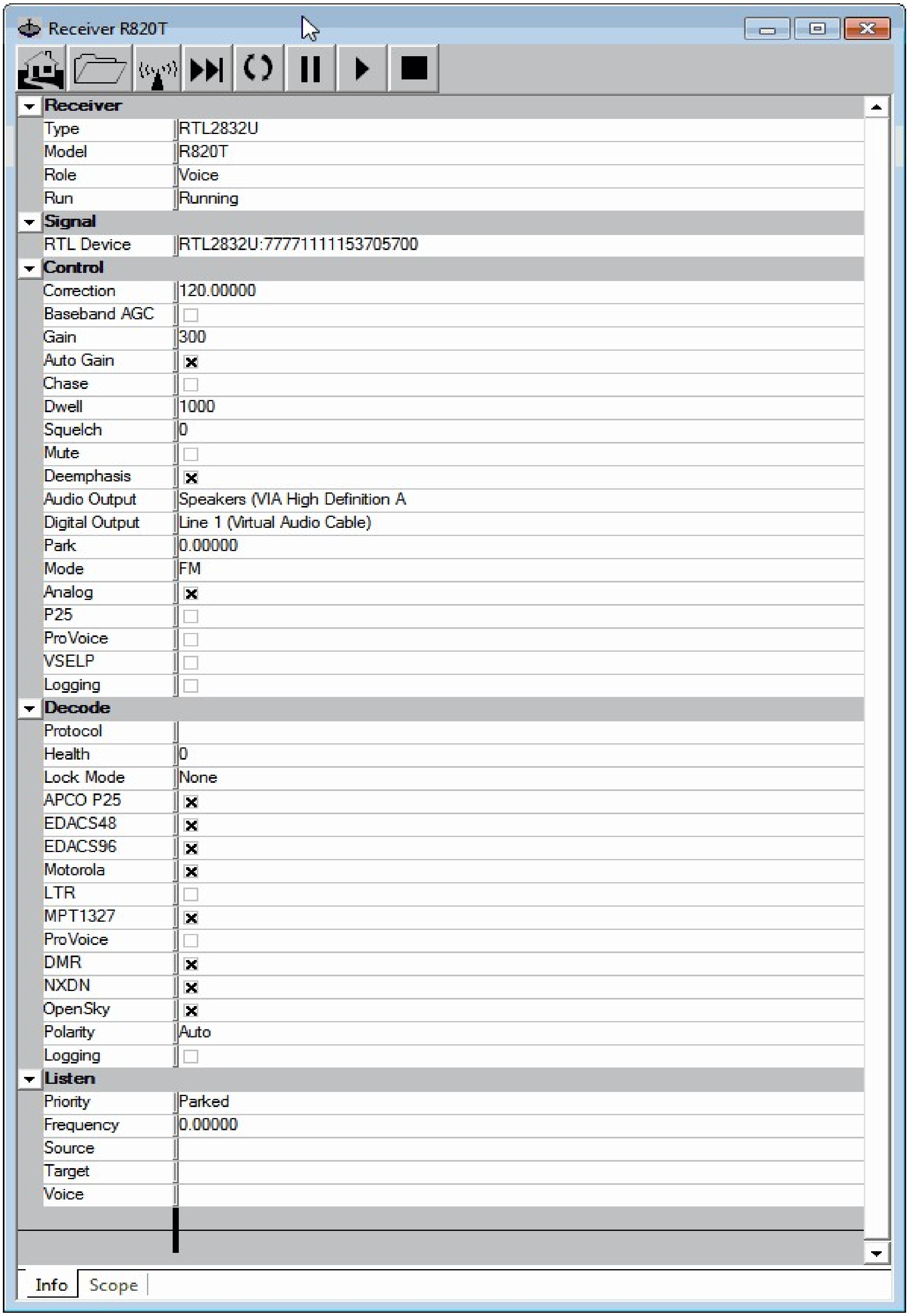
4. For the RTL Device, choose your second RTL-SDR dongle.

5. Set the gain and PPM offset for this dongle appropriately. Note that setting the correct PPM offset is critical for correct operation.

6. Choose the Audio Output as your speakers and make sure that Mute is *not* checked.

7. Set the ‘Park’ frequency to zero. This is the frequency the receiver will revert to if there is no activity.

8. If desired, set the squelch to a value that is similar to what you might use in SDR#. The squelch will stop static from playing when there is no voice signal. You may wish to leave this at zero for now, and then tune it with trial and error later when you have everything going. If you are using trial and error try a value around or try somewhere around 40 - 60 first.



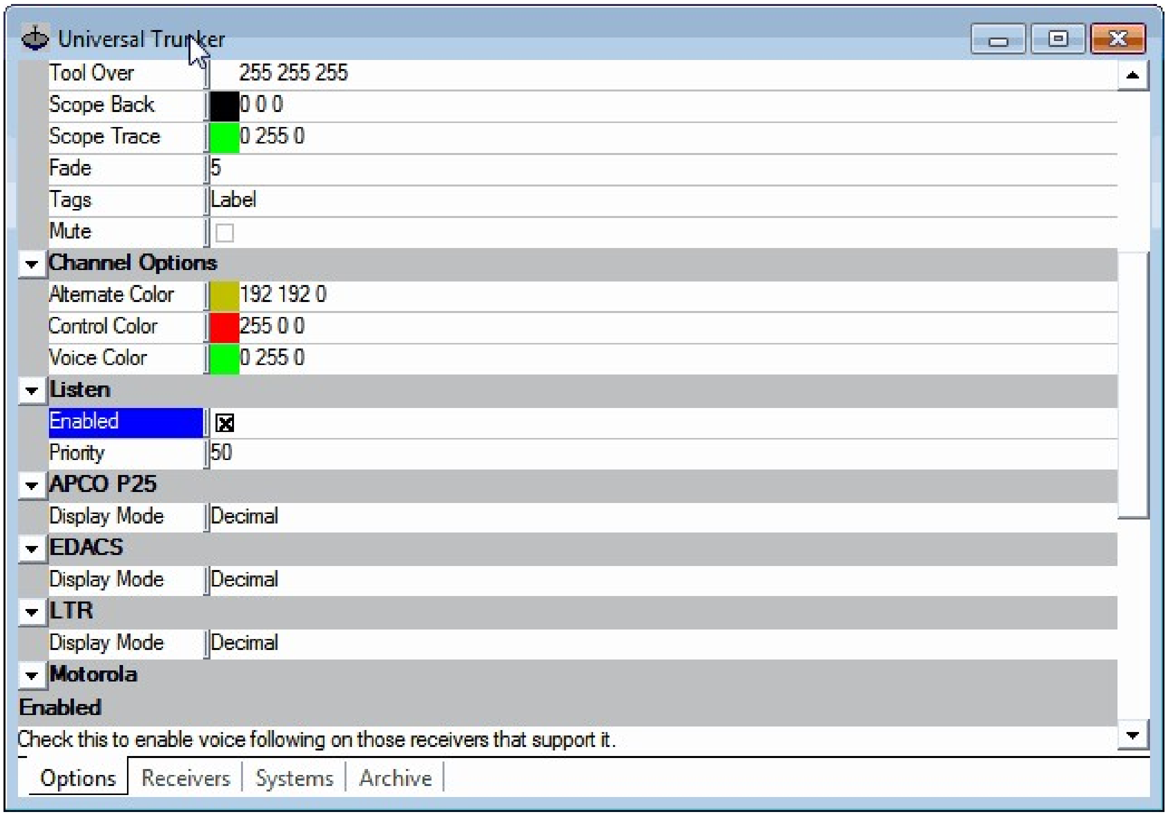
Receiver	
Type	RTL2832U
Model	R820T
Role	Voice
Run	Running
Signal	
RTL Device	RTL2832U:77771111153705700
Control	
Correction	120.00000
Baseband AGC	□
Gain	300
Auto Gain	☒
Chase	□
Dwell	1000
Squelch	0
Mute	□
Deemphasis	☒
Audio Output	Speakers (VIA High Definition A
Digital Output	Line 1 (Virtual Audio Cable)
Park	0.00000
Mode	FM
Analog	☒
P25	□
ProVoice	□
VSELP	□
Logging	□
Decode	
Protocol	
Health	0
Lock Mode	None
APCO P25	☒
EDACS48	☒
EDACS96	☒
Motorola	☒
LTR	□
MPT1327	☒
ProVoice	□
DMR	☒
NXDN	☒
OpenSky	☒
Polarity	Auto
Logging	□
Listen	
Priority	Parked
Frequency	0.00000
Source	
Target	
Voice	

9. Press the Play button ▶.

ENABLING UNITRUNKER FOLLOWING

Now we will make sure that Unitrunker is set to allow the voice receiver to follow calls from the signal channel.

1. Go back to the main Unitrunker window and click on the “Options” tab at the bottom of the window.
2. Make sure that “Enabled” is checked under Listen. This will allow the voice channel to follow the trunking information from the control signal.



Tool Over	255 255 255
Scope Back	■ 0 0 0
Scope Trace	🟩 0 255 0
Fade	5
Tags	Label
Mute	□
Channel Options	
Alternate Color	🟨 192 192 0
Control Color	🟥 255 0 0
Voice Color	🟩 0 255 0
Listen	
Enabled	☒
Priority	50
APCO P25	
Display Mode	Decimal
EDACS	
Display Mode	Decimal
LTR	
Display Mode	Decimal
Motorola	

Enabled

Check this to enable voice following on those receivers that support it.

At this point your trunking receiver should now be up and running!

PRIORITY AND LOCKOUT

If you want to listen to one or more than one talkgroup, but also block others that you are not interested in, then you can use either the “Lockout” feature or the priority settings. This will allow you to mute the talkgroups you are not interested in.

The lockout feature will mute a particular talkgroup. To use it go to the Call History tab in the Site window that pops up. Double click on the group that you want to lockout (note double click on the audience or target column) and check

Lockout which will be under the “Listen” heading on the settings window that pops up.

Priority allows you to prioritise which talk groups you want to listen to in the situation where more than one group is transmitting at once. Priority can be set just above the Lockout check box described in the paragraph above. A priority of 1 is the highest, and a priority of 50 is the lowest. Talkgroups with a higher priority will be listened to over those with a lower priority when more than one talkgroup is talking at once. All talkgroups are given a priority of 50 by default. The priority threshold can be set in the Unitrunker Options under the Listen heading. Any talkgroup with a priority under this threshold can be listened to. So for example if your wanting to listen to emergency services (police, fire, ambulace) only, but the trunking channel is shared with other non-emergency services groups too, then you could set the priority of the police, fire and ambulance talkgroups all to 49. Then you could set the listen threshold to 49, which would cause all the other non emergency service groups to be blocked.

TROUBLESHOOTING

If you have trouble using two dongles on the same computer due to the computer not being able to differentiate between the two dongles, then you may need to set one dongle to use a different serial number. This can be done with rtl_eeprom which comes as part of the official Osmocom RTL-SDR release from http://sdr.osmocom.org/trac/wiki/rtl-sdr. See this page for rtl_eeprom instructions http://manpages.ubuntu.com/manpages/trusty/man1/rtl_eeprom.1.html.

ALTERNATIVE TRUNK FOLLOWING SOFTWARE

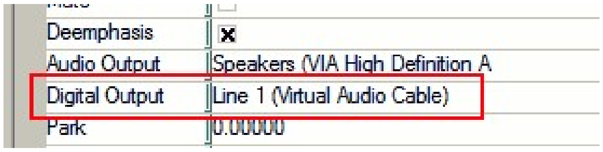
Apart from Unitrunker there are two other software packages for trunking that will work with the RTL-SDR.

One is Trunk88 available from http://wiki.trunk88.com/. Trunk88 is a Windows programs that can monitor Motorola trunked radio systems and can directly interface with an RTL-SDR dongle.

Another program is SDRTrunk which is available from https://code.google.com/p/sdrtrunk/. It is a Java based program for trunk tracking multiple analogue and digital trunking channels using multiple RTL-SDR dongles. Because it is Java based it can run on either Windows or Linux. Currently it supports Fleetsync II, LJ-1200, LTR, LTR-Net, MDC-1200, MPT-1327, Passport and 1200 ANI/GPS. Fleetsync 1200 GPS is a system used by some Taxi companies to track the location of their vehicles. SDRTrunk can be used to receive these signals and plot the Taxi’s on a map.

DECODING DIGITAL VOICE (P25/DMR/MOTOTRBO/NXDN/PROVOICE)

Decoding of unencrypted digital voice signals like P25/DMR/Mototrbo can be set up in the same way as the analogue system shown above. However, instead of outputting audio to the speakers, audio should be output to a virtual audio pipe such as VAC or VB Cable using the Digital Output audio selection in Unitrunker. Then DSD+ should be used to listen to the audio pipe. See information in [Appendix A: Audio Piping]() for more information if you don't have any audio pipes set up.



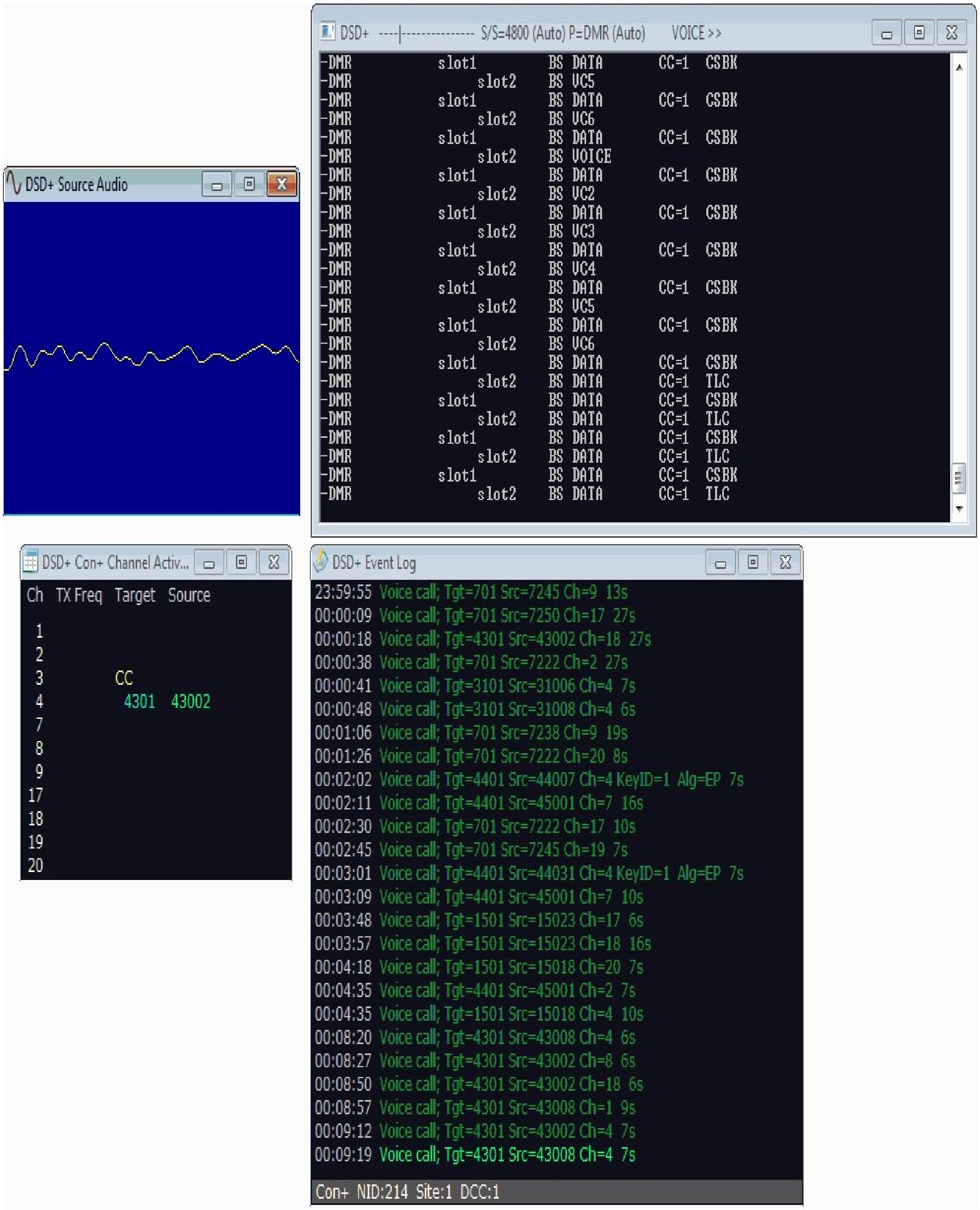
Deemphasis	☒
Audio Output	Speakers (VIA High Definition A
Digital Output	Line 1 (Virtual Audio Cable)
Park	0.00000

DSD+ is a program that is capable of decoding digital voice protocols such as P25, DMR/Mototrbo, NXDN and ProVoice. It is not capable of decoding encrypted digital voice channels. DSD+ is similar to the open source DSD version 1.6, but it has vastly improved decoding and much better voice quality. A direct link to the latest version of DSD+ can be found here http://www.dsdplus.com/, we suggest you download from this link. An active discussion on DSD+ and it's development can be found on the RadioReference forums at http://forums.radioreference.com/voice-control-channel-decoding-software/.

To use DSD+ download and extract the zip file into a folder on your PC. Then simply double click on DSDPlus.exe to start the DSD+ GUI. DSD+ will listen to your default sound device, so ensure that you have set this correctly to your chosen audio pipe used in Unitrunker in the Windows sound recording properties. Adjust the volume levels such that the source audio waveform window shows good sound levels, but is not too loud such that the waveform begins to look squarish.

Make sure that the Deemphasis option in Unitrunker is turned ***OFF*** (no x in the checkbox) for the voice receiver to make it work with DSD+. There seems to currently be a bug where sometimes the deemphasis filter needs to be turned off then on again to truly turn it off, so just be aware of this issue.

When DSD+ is running correctly you will see a bunch of text scrolling in the command prompt window when a digital voice signal is active. The text 'voice' will show on the right when a voice signal is playing. The Event Log will show a log of all calls and data that were received, and the Channel Activity window shows the current calls being made.



DSD+ Source Audio
DSD+ ----|-------------- S/S=4800 (Auto) P=DMR (Auto) VOICE >>
DSD+ Con+ Channel Activ...
DSD+ Event Log

```
-DMR          slot1          BS DATA       CC=1  CSBK
-DMR                slot2    BS VC5
-DMR          slot1          BS DATA       CC=1  CSBK
-DMR                slot2    BS VC6
-DMR          slot1          BS DATA       CC=1  CSBK
-DMR                slot2    BS VOICE
-DMR          slot1          BS DATA       CC=1  CSBK
-DMR                slot2    BS VC2
-DMR          slot1          BS DATA       CC=1  CSBK
-DMR                slot2    BS VC3
-DMR          slot1          BS DATA       CC=1  CSBK
-DMR                slot2    BS VC4
-DMR          slot1          BS DATA       CC=1  CSBK
-DMR                slot2    BS VC5
-DMR          slot1          BS DATA       CC=1  CSBK
-DMR                slot2    BS VC6
-DMR          slot1          BS DATA       CC=1  CSBK
-DMR                slot2    BS DATA       CC=1  TLC
-DMR          slot1          BS DATA       CC=1  CSBK
-DMR                slot2    BS DATA       CC=1  TLC
-DMR          slot1          BS DATA       CC=1  CSBK
-DMR                slot2    BS DATA       CC=1  TLC
-DMR          slot1          BS DATA       CC=1  CSBK
-DMR                slot2    BS DATA       CC=1  TLC
```

```
Ch  TX Freq  Target  Source

 1
 2
 3           CC
 4            4301   43002
 7
 8
 9
17
18
19
20
```

```
23:59:55  Voice call; Tgt=701 Src=7245 Ch=9  13s
00:00:09  Voice call; Tgt=701 Src=7250 Ch=17  27s
00:00:18  Voice call; Tgt=4301 Src=43002 Ch=18  27s
00:00:38  Voice call; Tgt=701 Src=7222 Ch=2  27s
00:00:41  Voice call; Tgt=3101 Src=31006 Ch=4  7s
00:00:48  Voice call; Tgt=3101 Src=31008 Ch=4  6s
00:01:06  Voice call; Tgt=701 Src=7238 Ch=9  19s
00:01:26  Voice call; Tgt=701 Src=7222 Ch=20  8s
00:02:02  Voice call; Tgt=4401 Src=44007 Ch=4 KeyID=1  Alg=EP  7s
00:02:11  Voice call; Tgt=4401 Src=45001 Ch=7  16s
00:02:30  Voice call; Tgt=701 Src=7222 Ch=17  10s
00:02:45  Voice call; Tgt=701 Src=7245 Ch=19  7s
00:03:01  Voice call; Tgt=4401 Src=44031 Ch=4 KeyID=1  Alg=EP  7s
00:03:09  Voice call; Tgt=4401 Src=45001 Ch=7  10s
00:03:48  Voice call; Tgt=1501 Src=15023 Ch=17  6s
00:03:57  Voice call; Tgt=1501 Src=15023 Ch=18  16s
00:04:18  Voice call; Tgt=1501 Src=15018 Ch=20  7s
00:04:35  Voice call; Tgt=4401 Src=45001 Ch=2  7s
00:04:35  Voice call; Tgt=1501 Src=15018 Ch=4  10s
00:08:20  Voice call; Tgt=4301 Src=43008 Ch=4  6s
00:08:27  Voice call; Tgt=4301 Src=43002 Ch=8  6s
00:08:50  Voice call; Tgt=4301 Src=43002 Ch=18  6s
00:08:57  Voice call; Tgt=4301 Src=43008 Ch=1  9s
00:09:12  Voice call; Tgt=4301 Src=43002 Ch=4  7s
00:09:19  Voice call; Tgt=4301 Src=43008 Ch=4  7s

Con+  NID:214  Site:1  DCC:1
```

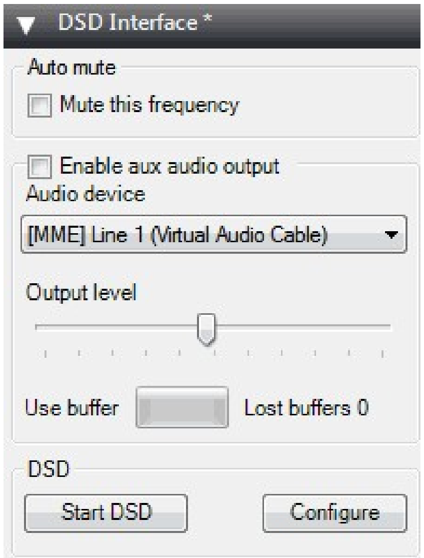
SOME TIPS

- For digital voice you may need to play with the Windows volume settings of your audio pipe to improve decoding.
- You will require a decent antenna that is capable of receiving the trunking frequencies you are interested in.
- Unitrunker can be coupled with a radioreference.com account to get trunk group names.
- Stereo mix can be used instead of VAC or VBCable, but you will be hearing both the digital signal as well as the decoded voice at the same time. Also, the decoded voice audio will be pumped back into DSD+ causing a detrimental feedback loop.

DECODING DIGITAL VOICE IN SDR#

Sometimes it might be more desirable to just decode a digital signal through SDR# if there is a single signal you are interested in, or you are just not interested in trunk following with Unitrunker.

1. Install DSD+ as described in previous the paragraphs.
2. Next, to make things easy download the DSD+ SDR# GUI plugin from http://www.rtl-sdr.ru/page/novyj-plagin-1. This site is in Russian, but the download is the link at the bottom of the post. The plugin helps simplify the operation of DSD+ by providing a nice GUI. Of course it is possible to do all this without the GUI and through the command line instead if you prefer.



DSD Interface *

Auto mute

□ Mute this frequency

□ Enable aux audio output

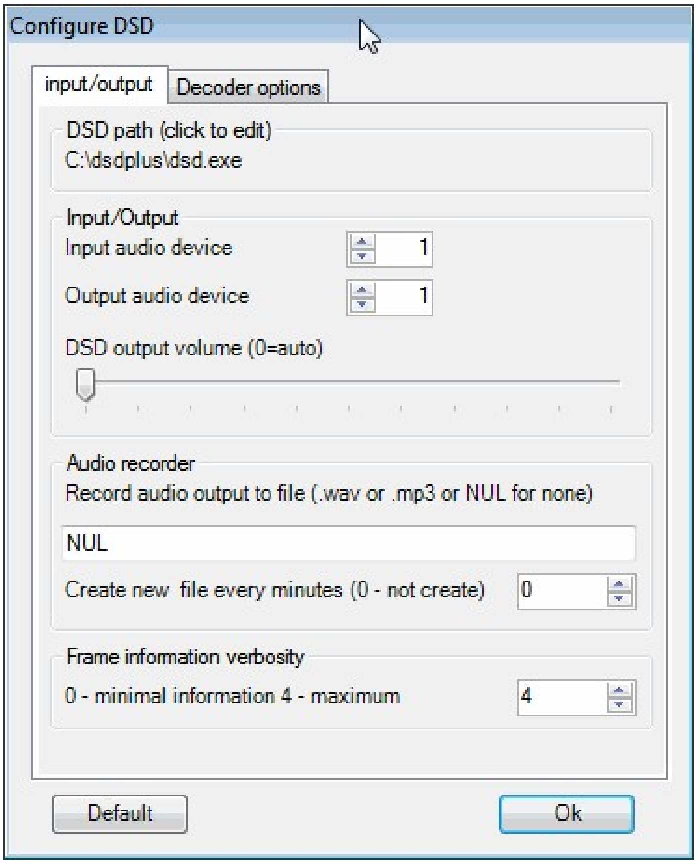
Audio device: [MME] Line 1 (Virtual Audio Cable)

Output level

Use buffer

Lost buffers 0

3. Once installed into SDR#, you will need to tell the plugin the location of where your DSD+ is installed to. To do this, in the plugin click the configure button and then under DSD Path, click on the default path and find dsd.exe in your file system. You will also need to set the input and output audio devices. The device that is set as the *default* audio device in the Windows audio devices window (accessed by right clicking the sound icon in the taskbar) will be device number one. Here we have set our default input audio device as a VAC audio pipe and the default output audio device as the speakers. See information in Appendix A: Audio Piping for more information if you don't have any audio pipes set up.
   In the decoder option tab you can also edit the options for the digital signal that you want to decode, but the default automatic options will work fine in most cases.



Configure DSD

DSD path (click to edit)
C:\dsdplus\dsd.exe

Input/Output
Input audio device 1
Output audio device 1
DSD output volume (0=auto)

Audio recorder
Record audio output to file (.wav or .mp3 or NUL for none)
NUL
Create new file every minutes (0 - not create) 0

Frame information verbosity
0 - minimal information 4 - maximum 4

4. Now to decode a digital signal, first tune to that signal in SDR#, setting the mode to NFM and making sure that 'Filter Audio' is *not* checked. Make sure that any other filters like digital noise reduction are also off.

5. Ensure that the audio output device in SDR# is set to the device you are using as your audio pipe to DSD+.
6. In the DSD+ plugin, click “Start DSD”. Ensure the volume levels in SDR# are loud enough so that the Lvl displayed at the top of the DSD+ command line window is around 50%.
7. At this point DSD+ should be working properly and decoding the digital audio.

TUNING DSD+

Note that DSDTuner currently does not seem to work with the latest versions of DSD+.

DSD+ contains many tunable options that are shown in the help screen ‘dsd -h’. These options are for improving vocal quality which can be tuned automatically with a special program called DSDTuner. DSDTuner simply iterates through all possible options and values for those options and then selects the best.

DSDTuner must be built from source from https://github.com/dreinhold/dsdtune. We have also uploaded a Windows binary at https://dl.dropboxusercontent.com/u/43061070/dsdtune.exe for ease, but you will probably need MingW installed with the path correctly set to run it. To compile this program you must have MingW with gcc installed.

To compile DSDTune, download or git clone it from https://github.com/dreinhold/dsdtune. Assuming you have MingW installed, in command prompt run `gcc -Wall -o dsdtune.exe utils.c dsdtune.c`. This will generate dsdtune.exe.

To use DSDTune first open DSD+ and listen to a digital voice channel. Whilst decoding press the ‘r’ key on your keyboard to begin logging a raw audio file. Press ‘r’ again after about one minute to stop recording. Next copy the dsdtune.exe file to the DSD+ folder. Then run `dsdtune -i DSDPlus-Raw-Input_2013-12-28@140307.wav`, where the wav file is the file you just recorded. Also be sure to pass the decoder options to dsdtune if you use any in DSD+ (e.g. -f, -fn, -fN, -fr).

MOTOTRBO LOCATION REQUEST RESPONSE PROTOCOL (LRRP) DECODING

Some MotorTRBO radios contain a GPS receiver within them that is used to send GPS locations over radio to base. This is useful for tracking a fleet of vehicles for instance. The protocol used to send these locations out is called Location Request Response Protocol or LRRP for short and it is contained within a standard MotorTRBO/DMR signal. DSD+ can be used to decode LRRP signals and show them on a map.

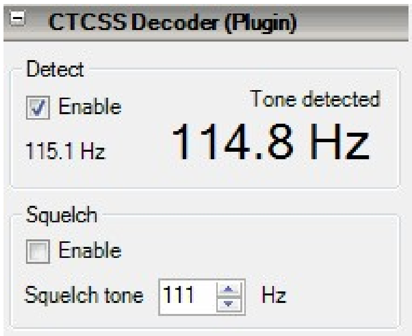
To show decoded LRRP coordinates on a map when using DSD+, simply open LRRP.exe which is in the same directory as DSDPlus.exe. Then tune to a MotoTRBO signal just as you would when listening to voice. As LRRP signals are received they will start to display on the map. You can use the mouse to pan and zoom around the map to look at your local area.

Note that many LRRP radio users use a third party GPS software system which cannot be decoded by DSD+. If you do not see any coordinates in the DSD+ event log when an LRRP event is shown in the event log then this may be the case.

CTCSS TRUNKED RADIO

CTCSS is an acronym for **C**ontinuous **T**one-**C**oded **S**quelch **S**ystem and is a squelching system that is used in shared two way radio systems (walkie-talkies). It is common for a single voice radio channel to be shared over a number of user groups for frequency use efficiency. CTCSS uses a special tone within the voice audio that identifies each group of users. This tone is used to prevent a group hearing radio chatter from another group sharing the same channel.

A SDR# plugin by the name of CTCSS decoder can be used to detect and use CTCSS tones for squelching. It can be downloaded from http://rtl-sdr.ru/page/ctcss-detektor-shumopodavitel (Mirror 1: http://sdrts.amoti.ru/ctcss_detektor_shumopodavitel_) (Mirror 2: http://bit.ly/1iV44Me).



CTCSS Decoder (Plugin)

Detect

☑ Enable

Tone detected

115.1 Hz

114.8 Hz

Squelch

□ Enable

Squelch tone 111 Hz

DCS TRUNKED RADIO

DCS is an acronym for Digital Code Squelch and is a system very similar to CTCSS in that it allows for radio user sharing by ensuring that radio users are not bothered by communications not intended for them. DCS is sometimes also known as Continuous Digital-Coded Squelch System (CDCSS).

A SDR# DCS decoder and squelcher plugin can be downloaded from http://www.rtl-sdr.ru/page/novyj-plagin-dcs-decoder.

LTR TRUNKED RADIO

Logic trunked radio is a type of trunked radio system that does not use a control channel. Instead each repeater has its own controller whose control signal is modulated into the voice signal. LTR is known as a distributed trunking system as opposed to a dedicated trunking system which uses a signal control channel.

LTR can be analysed using a Windows command line program known as LTR Analyzer which can be downloaded here http://home.ica.net/~phoenix/wap/LTR/. Download LTR-Analyzer.zip and extract the files to a folder. However, note that this program will not do much more than showing information about the LTR signal.

To use LTR analyzer simply set your audio pipe as the default device in Windows recording settings, set the audio pipe output in your favourite SDR receiver and then tune to an LTR signal. Open LTR analyzer by double clicking on LTR-Analyzer.exe in the folder. Upon opening it you'll be asked to select the audio input device. Choose your audio piping method. Now increase the volume settings on your SDR receiver until the MODULATION: NO AUDIO changes to a percentage.

FLEETSYNC II

Fleetsync II is a protocol that can transmit status information as well as GPS coordinates. It is usually transmitted along with a voice signal, and can be identified by a short sound that plays before voice. Fleetsync II can be decoded with SDR Trunk which is available https://code.google.com/p/sdrtrunk/. SDRTrunk will display decoded GPS messages on a map.

PAGER DECODING GUIDE

The RTL-SDR combined with SDR# and a POCSAG/Flex capable decoding software application can be used to decode pager messages. With this setup you can receive pager messages from all pager users on the system.

If you don’t know what a pager is, since they are now uncommon, here is a brief explanation from Wikipedia:

A pager is a wireless telecommunications device that receives and displays numeric or text messages, or receives and announces voice messages.

Not many people use pagers these days with mobile phone text messaging being more common, but pagers are still popular with doctors, some fire and ambulance agencies and various service companies as they tend to be more reliable and have greater coverage.

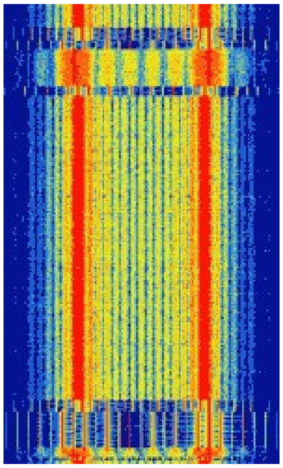
In most countries it is perfectly legal to receive messages from pagers, as they are plain text unencrypted. BUT it is illegal to act on the information received. Please respect your local laws especially as sensitive data is often transmitted through pagers.

TUTORIAL

While directed at the RTL-SDR, this tutorial may also be useful for use with other software defined radios such as the Funcube dongle and HackRF, or even traditional hardware radios with a discriminator tap.

Since pager signals are usually transmitted at a very strong power level, usually almost any antenna will work to receive them, even the stock antenna that comes with the dongle. Pager frequencies differ among different countries. Usually they will be anywhere from 137 - 160 MHz, around ~450 MHz, or around 900 MHz. Most commonly they are found at around 150 MHz. Check radioreference.com or Google for pager frequencies in your area.

Pagers typically use POCSAG or Flex encoding and the signal waterfall will look something like the example shown below. They also have a distinctive sound on NFM. For a sound example see the signal identification guide at www.sigidwiki.com.



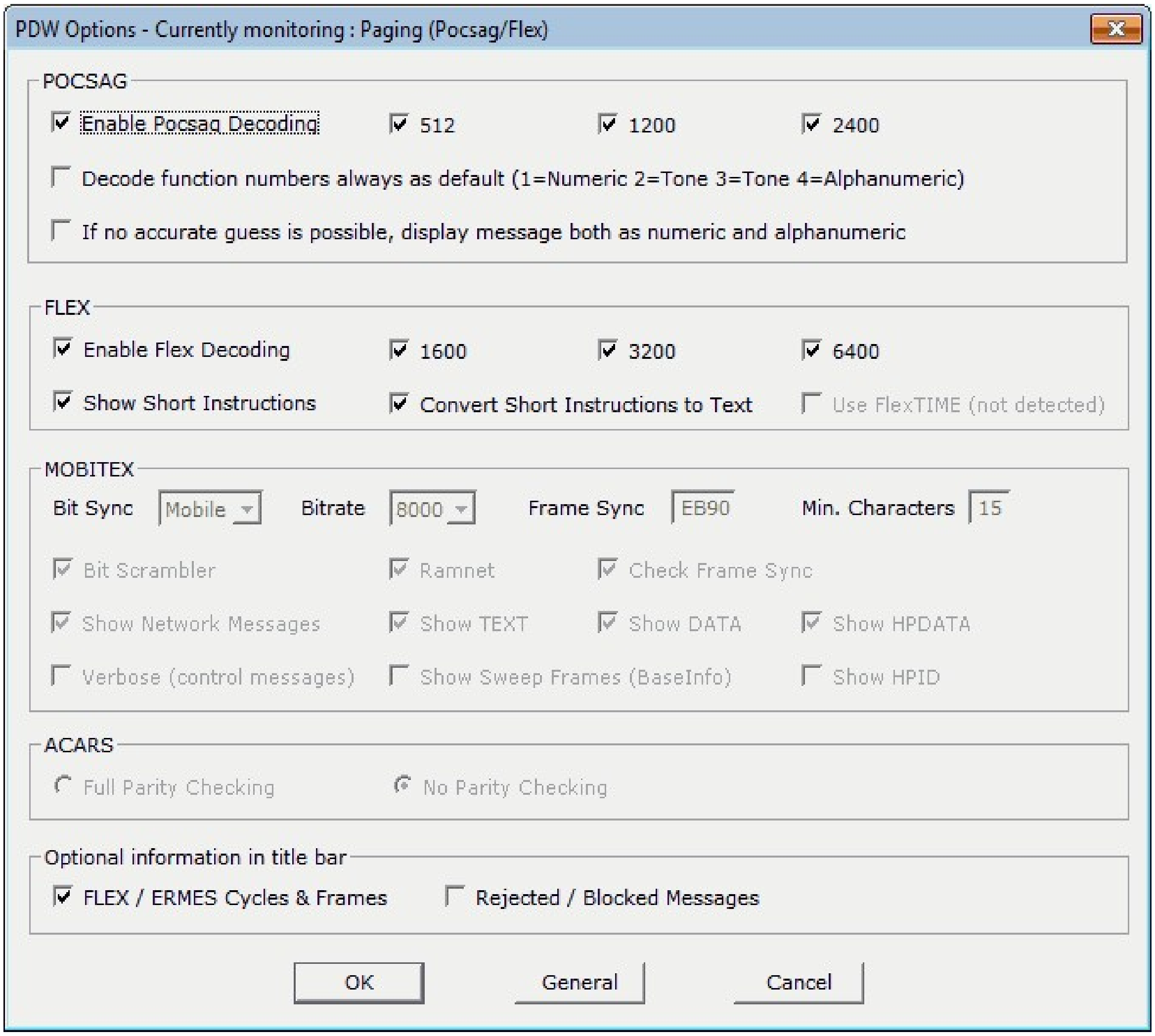
For this tutorial, you will need to have an RTL-SDR dongle set up and working with SDR# or another SDR receiver program. We will assume you have this much done already. If you not, visit the Buy RTL-SDR dongles page at http://www.rtl-sdr.com/buy-rtl-sdr-dvb-t-dongles/ and follow the Quickstart Guide at the beginning of this book.

You will also need to have an audio piping method installed and set up. Audio piping will allow the audio from SDR# or your favorite SDR receiver software to be passed to a decoding program. Either stereo mix, or Virtual Audio Cable, or VBCable may can be used. See Appendix A: Audio Piping if you need more information. Also before starting ensure your audio piping method is set to a sample rate of 44100 Hz. See the setting the audio sample rate section if you need help doing this.

To decode the POCSAG or Flex signals, you need need to download and install a free program called PDW. PDW can be downloaded from http://www.discriminator.nl/pdw/index-en.html. After downloading and installing PDW follow these steps.

1. Open your favourite SDR receiver software such as SDR# and set the output audio to use your preferred audio piping method and then start the radio.
2. Tune to a pager POCSAG/Flex signal. Set the receive mode to NFM, filter bandwidth to 12500 Hz, turn squelch OFF, turn OFF any DSP or noise reduction algorithms and if in SDR# set filter audio to OFF. Adjust the RF gain settings until good reception is achieved. Since pager signals are so strong sometimes you may actually need to reduce the gain to ensure that the signal is not saturating.

3. Open PDW. You may initially receive some errors, but they can be safely ignored. Go to **Options -> Options** and click "Enable Pocsag Decoding". Ensure the 512, 1200 and 2400 boxes are all checked. Also, ensure Enable Flex Decoding is enabled and that the 1600, 3200 and 6400 boxes are all checked. Press OK.



PDW Options - Currently monitoring : Paging (Pocsag/Flex)

POCSAG

☑ Enable Pocsag Decoding ☑ 512 ☑ 1200 ☑ 2400

□ Decode function numbers always as default (1=Numeric 2=Tone 3=Tone 4=Alphanumeric)

□ If no accurate guess is possible, display message both as numeric and alphanumeric

FLEX

☑ Enable Flex Decoding ☑ 1600 ☑ 3200 ☑ 6400

☑ Show Short Instructions ☑ Convert Short Instructions to Text □ Use FlexTIME (not detected)

MOBITEX

Bit Sync: Mobile Bitrate: 8000 Frame Sync: EB90 Min. Characters: 15

☑ Bit Scrambler ☑ Ramnet ☑ Check Frame Sync

☑ Show Network Messages ☑ Show TEXT ☑ Show DATA ☑ Show HPDATA

□ Verbose (control messages) □ Show Sweep Frames (BaseInfo) □ Show HPID

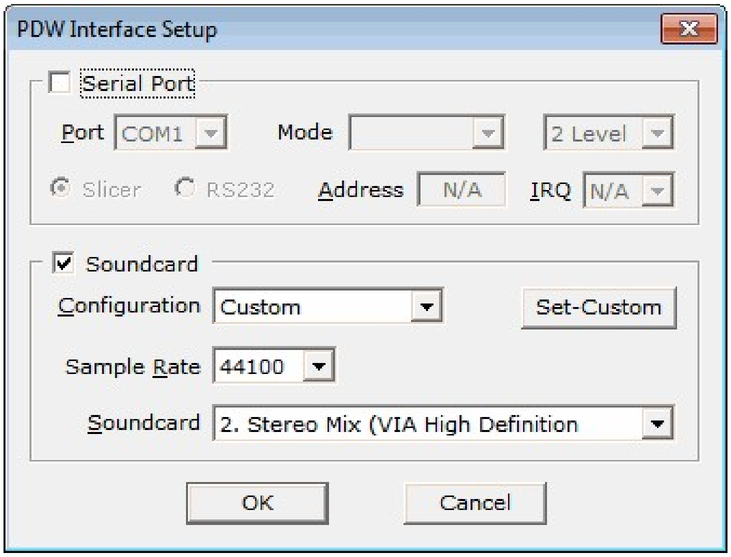
ACARS

○ Full Parity Checking ● No Parity Checking

Optional information in title bar

☑ FLEX / ERMES Cycles & Frames □ Rejected / Blocked Messages

4. Go to **Interface -> Setup**. Enable the Soundcard checkbox, set the Configuration to Custom and choose your audio piping method in the Soundcard drop down box. If you only have one audio piping method enabled in the Windows recording properties, it will automatically choose that method. Press OK.



□ Serial Port

Port COM1 Mode 2 Level

● Slicer ○ RS232 Address N/A IRQ N/A

☑ Soundcard

Configuration Custom

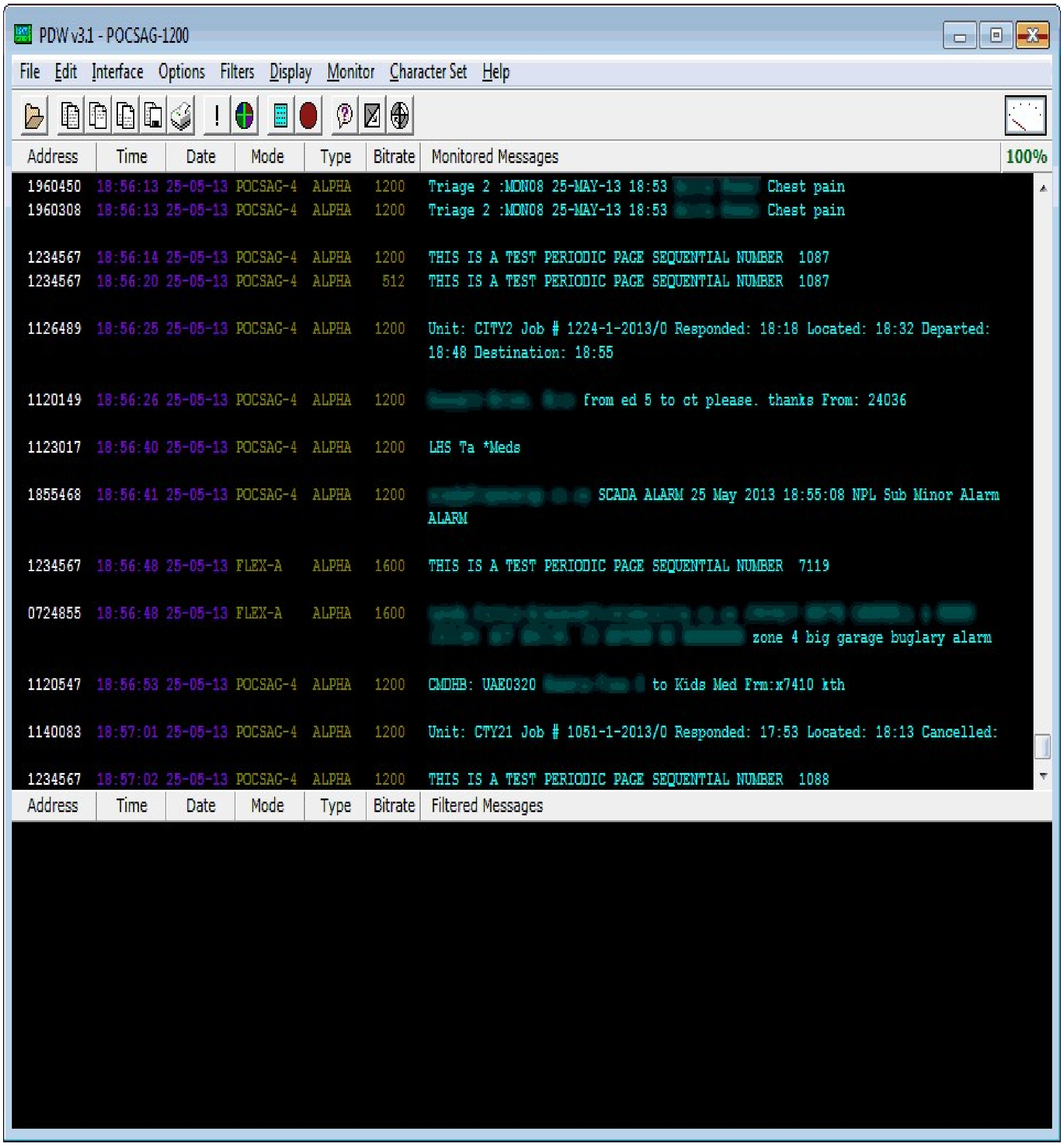
Sample Rate 44100

Soundcard 2. Stereo Mix (VIA High Definition

5. Go to the Monitor menu and ensure POCSAG/FLEX is ticked.

Now, if everything is set up correctly, the pager audio from SDR# should be being sent to PDW. In the top right hand corner of PDW, there should be a volume gauge. You will need to adjust the AF gain volume settings in your SDR receiver software and/or the Windows volume settings so that the volume meter goes up when a pager signal is sent.

The percentage sign under the meter shows the percentage ratio of good to bad decodes. 100% means all the received messages were decoded properly.


PDW v3.1 - POCSAG-1200
File Edit Interface Options Filters Display Monitor Character Set Help
100%
Filtered Messages

```
Address  Time     Date     Mode     Type   Bitrate Monitored Messages
1960450  18:56:13 25-05-13 POCSAG-4 ALPHA  1200    Triage 2 :MON08 25-MAY-13 18:53 [redacted] Chest pain
1960308  18:56:13 25-05-13 POCSAG-4 ALPHA  1200    Triage 2 :MON08 25-MAY-13 18:53 [redacted] Chest pain

1234567  18:56:14 25-05-13 POCSAG-4 ALPHA  1200    THIS IS A TEST PERIODIC PAGE SEQUENTIAL NUMBER  1087
1234567  18:56:20 25-05-13 POCSAG-4 ALPHA   512    THIS IS A TEST PERIODIC PAGE SEQUENTIAL NUMBER  1087

1126489  18:56:25 25-05-13 POCSAG-4 ALPHA  1200    Unit: CITY2 Job # 1224-1-2013/0 Responded: 18:18 Located: 18:32 Departed:
                                                   18:48 Destination: 18:55

1120149  18:56:26 25-05-13 POCSAG-4 ALPHA  1200    [redacted] from ed 5 to ct please. thanks From: 24036

1123017  18:56:40 25-05-13 POCSAG-4 ALPHA  1200    LHS Ta *Meds

1855468  18:56:41 25-05-13 POCSAG-4 ALPHA  1200    [redacted] SCADA ALARM 25 May 2013 18:55:08 NPL Sub Minor Alarm
                                                   ALARM

1234567  18:56:48 25-05-13 FLEX-A   ALPHA  1600    THIS IS A TEST PERIODIC PAGE SEQUENTIAL NUMBER  7119

0724855  18:56:48 25-05-13 FLEX-A   ALPHA  1600    [redacted]
                                                   [redacted] zone 4 big garage buglary alarm

1120547  18:56:53 25-05-13 POCSAG-4 ALPHA  1200    CMDHB: UAE0320 [redacted] to Kids Med Frm:x7410 kth

1140083  18:57:01 25-05-13 POCSAG-4 ALPHA  1200    Unit: CTY21 Job # 1051-1-2013/0 Responded: 17:53 Located: 18:13 Cancelled:

1234567  18:57:02 25-05-13 POCSAG-4 ALPHA  1200    THIS IS A TEST PERIODIC PAGE SEQUENTIAL NUMBER  1088
Address  Time     Date     Mode     Type   Bitrate Filtered Messages
```

OTHER DECODERS

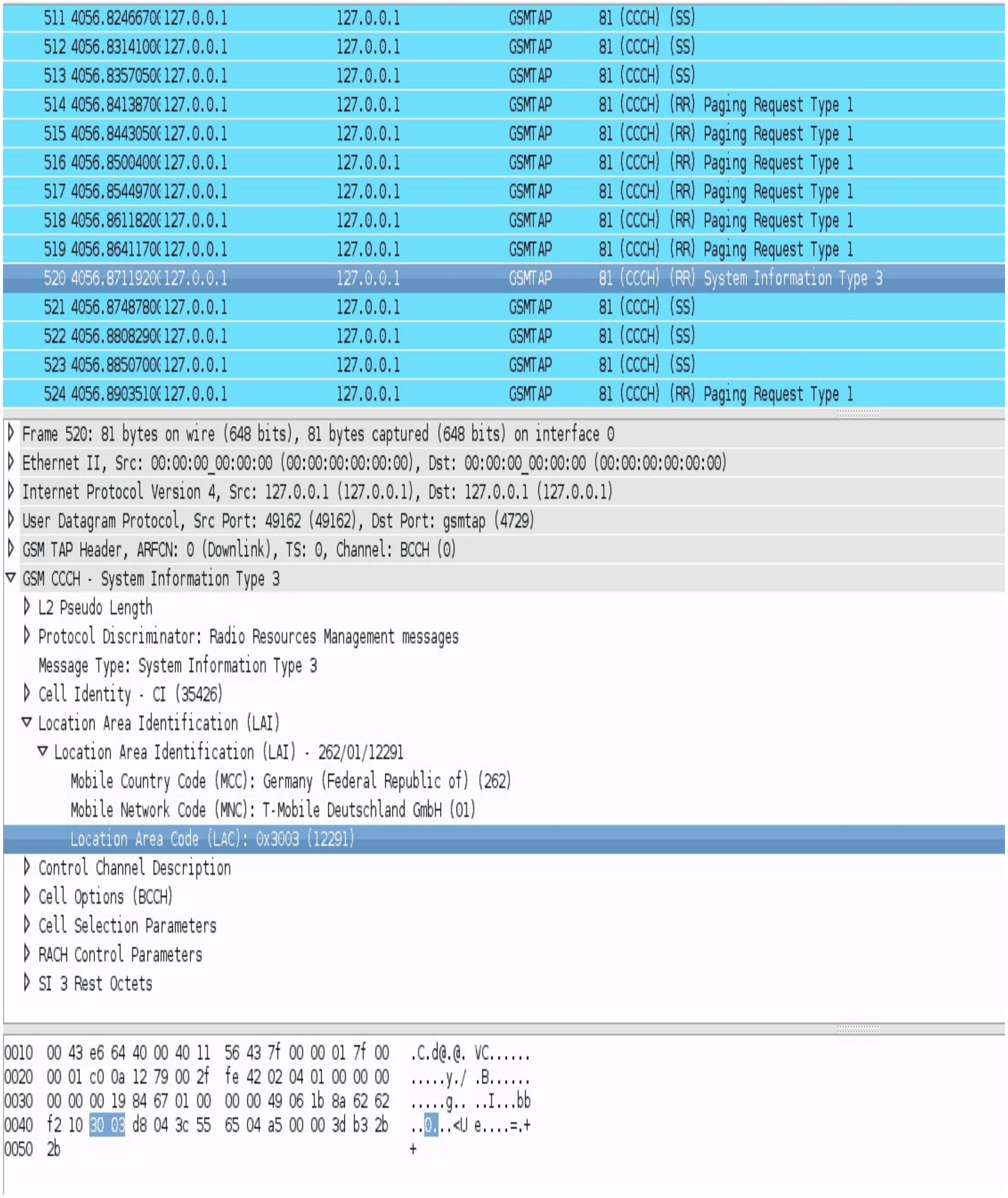
MultimonNG is also capable of decoding POCSAG and can also be run on Windows, Linux and even on an embedded device running Linux. At the command line in Linux use rtl_fm piped into multimon-ng to decode POCSAG and display it to the terminal. In the example code below replace the frequency of 158.1M and the PPM offset with a pager frequency used in your area and your dongles PPM offset value.

```
rtl_fm -f 158.1M -s 22050 -p 49 | multimon-ng -t raw -a POCSAG512 -a POCSAG1200 -a POCSAG2400 -f alpha /dev/stdin
```

See the rtl_fm guide or the MultimonNG guide if you need more information.

ANALYZING GSM SIGNALS

The RTL-SDR can be used to analyze cellular phone GSM signals using Linux based tools gr-gsm and Wireshark. The **G**lobal **S**ystem for **M**obile communications (GSM) is the communications protocol that is used by many mobile phones. The data in GSM signals such as text messages and voice data are of course encrypted, however it is possible to analyze the GSM system used in your area as there are multiple plain text system messages that can be received with an RTL-SDR. Note that with this tutorial you will not be able to receive any sensitive personal data. Below we show an image that shows some of the information that can be received.



```
511 4056.8246670(127.0.0.1          127.0.0.1          GSMTAP    81 (CCCH) (SS)
512 4056.8314100(127.0.0.1          127.0.0.1          GSMTAP    81 (CCCH) (SS)
513 4056.8357050(127.0.0.1          127.0.0.1          GSMTAP    81 (CCCH) (SS)
514 4056.8413870(127.0.0.1          127.0.0.1          GSMTAP    81 (CCCH) (RR) Paging Request Type 1
515 4056.8443050(127.0.0.1          127.0.0.1          GSMTAP    81 (CCCH) (RR) Paging Request Type 1
516 4056.8500400(127.0.0.1          127.0.0.1          GSMTAP    81 (CCCH) (RR) Paging Request Type 1
517 4056.8544970(127.0.0.1          127.0.0.1          GSMTAP    81 (CCCH) (RR) Paging Request Type 1
518 4056.8611820(127.0.0.1          127.0.0.1          GSMTAP    81 (CCCH) (RR) Paging Request Type 1
519 4056.8641170(127.0.0.1          127.0.0.1          GSMTAP    81 (CCCH) (RR) Paging Request Type 1
520 4056.8711920(127.0.0.1          127.0.0.1          GSMTAP    81 (CCCH) (RR) System Information Type 3
521 4056.8748780(127.0.0.1          127.0.0.1          GSMTAP    81 (CCCH) (SS)
522 4056.8808290(127.0.0.1          127.0.0.1          GSMTAP    81 (CCCH) (SS)
523 4056.8850700(127.0.0.1          127.0.0.1          GSMTAP    81 (CCCH) (SS)
524 4056.8903510(127.0.0.1          127.0.0.1          GSMTAP    81 (CCCH) (RR) Paging Request Type 1
```

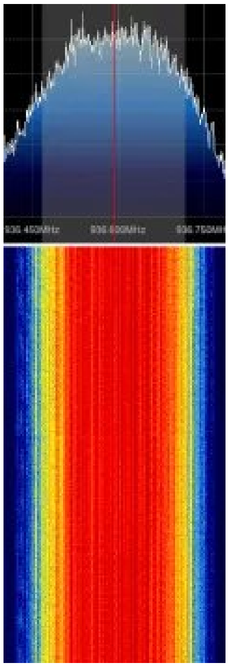
```
▷ Frame 520: 81 bytes on wire (648 bits), 81 bytes captured (648 bits) on interface 0
▷ Ethernet II, Src: 00:00:00_00:00:00 (00:00:00:00:00:00), Dst: 00:00:00_00:00:00 (00:00:00:00:00:00)
▷ Internet Protocol Version 4, Src: 127.0.0.1 (127.0.0.1), Dst: 127.0.0.1 (127.0.0.1)
▷ User Datagram Protocol, Src Port: 49162 (49162), Dst Port: gsmtap (4729)
▷ GSM TAP Header, ARFCN: 0 (Downlink), TS: 0, Channel: BCCH (0)
▽ GSM CCCH - System Information Type 3
  ▷ L2 Pseudo Length
  ▷ Protocol Discriminator: Radio Resources Management messages
    Message Type: System Information Type 3
  ▷ Cell Identity - CI (35426)
  ▽ Location Area Identification (LAI)
    ▽ Location Area Identification (LAI) - 262/01/12291
        Mobile Country Code (MCC): Germany (Federal Republic of) (262)
        Mobile Network Code (MNC): T-Mobile Deutschland GmbH (01)
        Location Area Code (LAC): 0x3003 (12291)
  ▷ Control Channel Description
  ▷ Cell Options (BCCH)
  ▷ Cell Selection Parameters
  ▷ RACH Control Parameters
  ▷ SI 3 Rest Octets
```

```
0010  00 43 e6 64 40 00 40 11  56 43 7f 00 00 01 7f 00   .C.d@.@. VC......
0020  00 01 c0 0a 12 79 00 2f  fe 42 02 04 01 00 00 00   .....y./ .B......
0030  00 00 00 19 84 67 01 00  00 00 49 06 1b 8a 62 62   .....g.. ..I...bb
0040  f2 10 30 03 d8 04 3c 55  65 04 a5 00 00 3d b3 2b   ..0...<U e....=.+
0050  2b                                                 +
```

Note: This tutorial can be quite difficult for those without good Linux experience. To avoid trouble, in this tutorial we describe the steps from a fresh install of Ubuntu 14.04.

The first step is to find out at what frequencies GSM signals are at in your area. For most of the world, the primary GSM band is 900 MHz, in the USA it starts from 850 MHz. If you have an E4000 RTL-SDR, you may also be able to find

GSM signals in the 1800 MHz band for most of the world and 1900 MHz band for the USA. Open up SDR# or your favourite SDR receiver software and scan around the 900 MHz (or 850 MHz) band for a signal that looks like the waterfall image below. This image shows a non-hopping GSM downlink signal. Check the Signal Identification Wiki (www.sigidwiki.com) for a GSM sound sample. Note down the strongest GSM frequencies you can find.



The rest of the tutorial is performed in Linux and for this we assume that you have basic Linux skills in using the terminal. Here we start from a fresh install of Ubuntu 14.04 on a virtual machine but this tutorial may also work on other Linux installations including ones that have GNU Radio preinstalled, but we cannot guarantee it.

INSTALLATION OF GR-GSM

This tutorial is heavily based on the instructions from the gr-gsm GitHub readme at https://github.com/ptrkrysik/gr-gsm**.**

1. The easiest way to install gr-gsm is to use Pybombs. Pybombs will automatically install gr-gsm, and all the required dependencies including GNU Radio.

```
git clone https://github.com/pybombs/pybombs.git
cd pybombs
sudo ./pybombs config
```

2. After running config you will be asked above several options. Leave everything as the default *except* for the install prefix which should be set to /usr/local. You can select the default options by simply pressing enter at each line.
3. Check to see if your Linux distribution has a GNU Radio 3.7.3 or higher package in its repository with the following. Ubuntu 14.10 should have GNU Radio 3.7.3+ in its repositories, but older versions will not.

```
apt-cache policy gnuradio-dev
```

4. If it does use GNU Radio 3.7.3+ then use the following code to install GNU Radio via the repository package, rather than from source which will save significant amounts of installation time. Do not install from repositories if the GNU Radio version is older.

```
./pybombs config forcebuild ' '
```

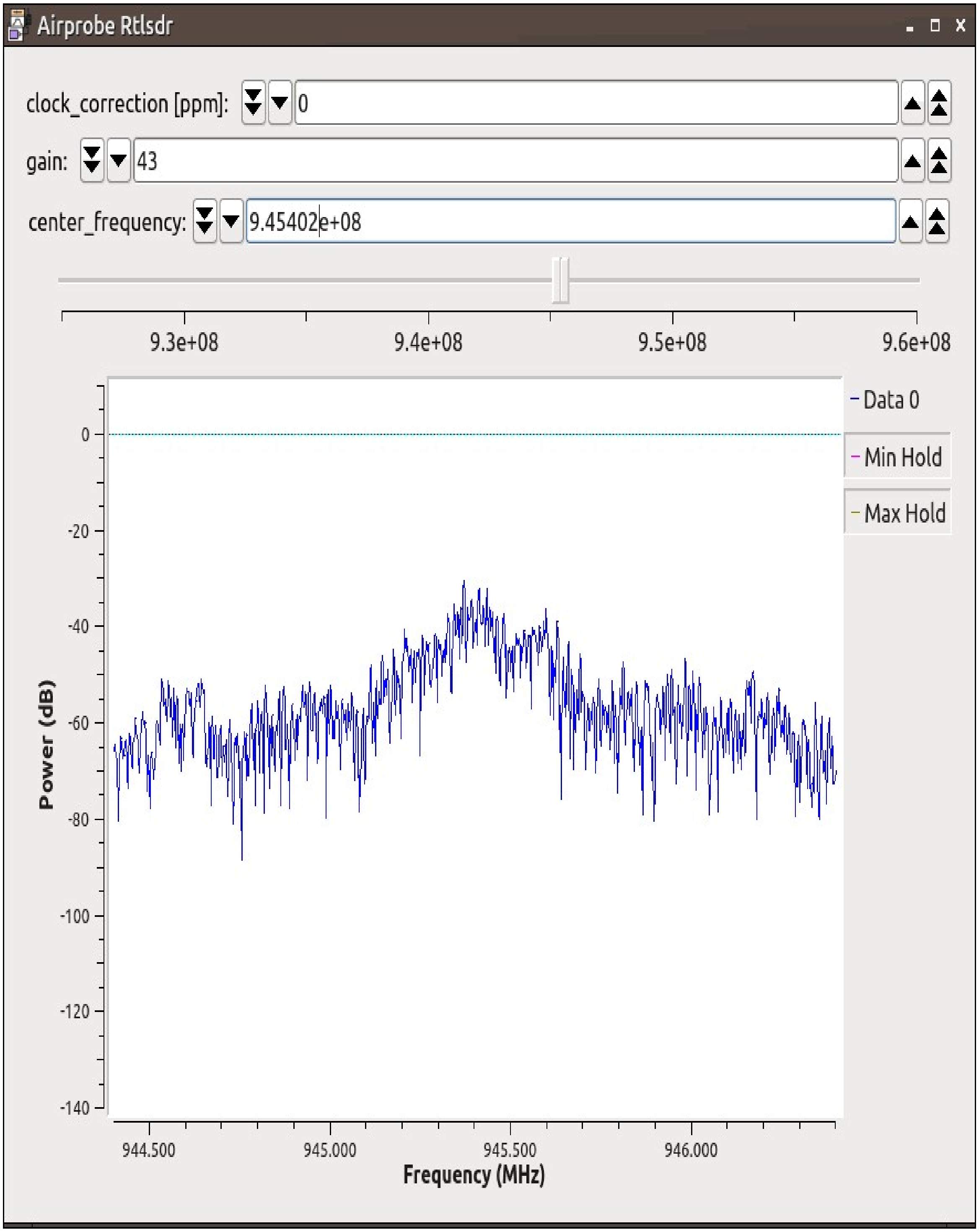
5. Now install gr-gsm by running

```
./pybombs install gr-gsm
```

6. Once installed you can start gr-gsm from anywhere by simply running the following.

```
airprobe_rtlsdr.py
```

7. In the window that pops up you can set the frequency of a GSM signal. For example in this image we tune to a GSM signal at 945.402 MHz, which we would enter under center_frequency as 9.45402e+08. Be sure to also adjust the gain and PPM settings if needed.



clock_correction [ppm]: 0

gain: 43

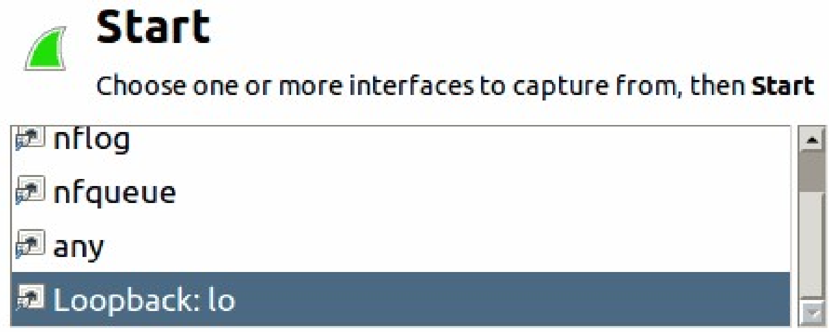
center_frequency: 9.45402e+08

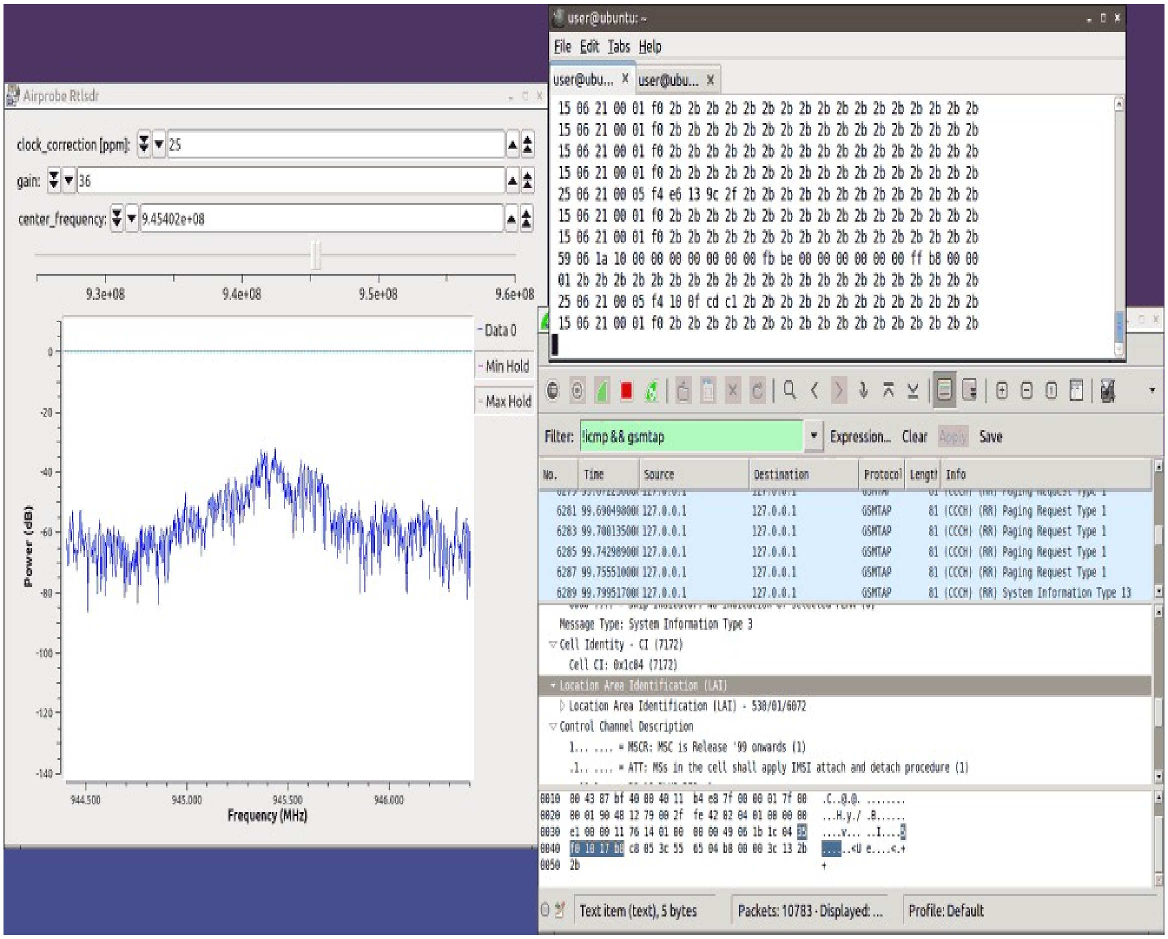
8. Now run Wireshark by typing into a terminal

```
sudo wireshark
```

9. Since gr-gsm dumps data into a UDP port, we must set Wireshark to listen to this. Under the Start button in Wireshark, first set the capture interface to Loopback:lo and then press Start. Then in the filter box near the top of the Wireshark window, type in !icmp && gsmtap. This will ensure that only airprobe GSM data is displayed.

   Note that you can also start wireshark by using `sudo wireshark -k -Y '!icmp && gsmtap' -i lo` which will automatically start wireshark in the loopback mode with the gsmtap filter activated.





clock_correction [ppm]: 25

gain: 36

center_frequency: 9.45402e+08

```
15 06 21 00 01 f0 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b
15 06 21 00 01 f0 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b
15 06 21 00 01 f0 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b
15 06 21 00 01 f0 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b
25 06 21 00 05 f4 e6 13 9c 2f 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b
15 06 21 00 01 f0 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b
15 06 21 00 01 f0 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b
59 06 1a 10 00 00 00 00 00 00 00 fb be 00 00 00 00 00 00 ff b8 00 00
01 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b
25 06 21 00 05 f4 10 0f cd c1 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b
15 06 21 00 01 f0 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b
```

Filter: !icmp && gsmtap

```
No.   Time          Source     Destination  Protocol Length Info
6281  99.69049800(  127.0.0.1  127.0.0.1    GSMTAP   81 (CCCH) (RR) Paging Request Type 1
6283  99.70013500(  127.0.0.1  127.0.0.1    GSMTAP   81 (CCCH) (RR) Paging Request Type 1
6285  99.74298900(  127.0.0.1  127.0.0.1    GSMTAP   81 (CCCH) (RR) Paging Request Type 1
6287  99.75551000(  127.0.0.1  127.0.0.1    GSMTAP   81 (CCCH) (RR) Paging Request Type 1
6289  99.79951700(  127.0.0.1  127.0.0.1    GSMTAP   81 (CCCH) (RR) System Information Type 13
```

```
   Message Type: System Information Type 3
 ▽ Cell Identity - CI (7172)
    Cell CI: 0x1c04 (7172)
 ▾ Location Area Identification (LAI)
  ▷ Location Area Identification (LAI) - 530/01/6072
 ▽ Control Channel Description
    1... .... = MSCR: MSC is Release '99 onwards (1)
    .1.. .... = ATT: MSs in the cell shall apply IMSI attach and detach procedure (1)
```

```
0010  00 43 87 bf 40 00 40 11  b4 e8 7f 00 00 01 7f 00   .C..@.@. ........
0020  00 01 90 48 12 79 00 2f  fe 42 02 04 01 00 00 00   ...H.y./ .B......
0030  e1 00 00 11 76 14 01 00  00 00 49 06 1b 1c 04 03   ....v... ..I.....
0040  f0 10 17 b0 c8 05 3c 55  65 04 b8 00 00 3c 13 2b   ......<U e....<.+
0050  2b                                                 +
```

Text item (text), 5 bytes | Packets: 10783 · Displayed: ... | Profile: Default

If you have problems installing gr-gsm via Pybombs, you can install GNU Radio first via the Marcus Leech install script and then install gr-gsm as follows.

```
git clone https://github.com/ptrkrysik/gr-gsm.git
cd gr-gsm
mkdir build
cd build
cmake ..
make
sudo make install
sudo ldconfig
```

INSTALLATION AND TESTING OF AIRPROBE

If for some reason you cannot get the newer and more up to date gr-gsm software to run, you can still use the older Airprobe software which gr-gsm is based on to analyze GSM signals. The install procedure for this method is a little longer. Note that you only need to go through the following procedure if you cannot get gr-gsm to work. We recommend trying to get gr-gsm to work first as its decoding performance is much better and more stable compared to Airprobe.

1. First, install GNU Radio using the Marcus Leech install script.

```
sudo apt-get update
wget http://www.sbrac.org/files/build-gnuradio && chmod a+x ./build-gnuradio && ./build-gnuradio
```

2. Install wireshark which is a program that will be used to display the data that is output by Airprobe.

```
sudo apt-get install wireshark
```

3. Install some dependencies required by Airprobe and libosmocore that may not have been installed by the GNU Radio installation script.

```
sudo apt-get install git-core autoconf automake libtool g++ python-dev swig libpcap0.8-dev pkg-config
liblog4cpp5-dev
```

4. Airprobe also requires that the libosmocore libraries are installed. They can be installed with the following procedure.

```
git clone git://git.osmocom.org/libosmocore.git
cd libosmocore
autoreconf –i
./configure
make
```

```
sudo make install
sudo ldconfig
```

5. Clone the Airprobe git repository using the following command.

```
git clone git://git.gnumonks.org/airprobe.git
```

6. Because the GNU Radio installation script installs the latest version of GNU Radio (version 3.7+), we need to use a patch by an RTL-SDR.com commenter named neeo/zmiana. This is because Airprobe has not been updated in a while and still expects GNU Radio 3.6. Download the patch file from http://speedy.sh/NBRYB/zmiana3.patch (Mirror: http://bit.ly/1v8e3T2) and move it into the Airprobe directory. (Note another option instead of patching may be to install the older GNU Radio version 3.6 by using the -o flag on the GNU Radio installation script)
7. In the Airprobe directory apply the patch by using the following patch command. After this command you should see several lines of text indicating that the patching was successful.

```
cd airprobe
patch -p1 < zmiana3.patch
```

8. Install gsm-receiver by using the following commands.

```
cd airprobe/gsm-receiver
./bootstrap
./configure
make
```

9. Next we will test if Airprobe installed correctly by downloading a pre-recorded cfile, which contains a GSM signal. (A mirror of cfile is available at http://bit.ly/1snLGCw.)

```
cd airprobe/gsm-receiver/src/python
wget --no-check-certificate https://svn.berlin.ccc.de/projects/airprobe/raw-
attachment/wiki/DeModulation/capture_941.8M_112.cfile
```

10. Open a second terminal window or tab and open wireshark by typing `sudo wireshark` into a second terminal window. Because we are running Wireshark as sudo, some errors and warnings may pop up, but these can be

safely ignored. Note that we are running Wireshark as sudo in order to get access to the local loopback interface.

11. Since Airprobe dumps data into a UDP port, we must set Wireshark to listen to this. Under the Start button in Wireshark, first set the capture interface to Loopback:lo and then press Start. Then in the filter box near the top of the Wireshark window, type in gsmtap. This will ensure that only airprobe GSM data is displayed.
12. Note that you can also start wireshark by using `sudo wireshark -k -Y 'gsmtap' -i lo` which will automatically start wireshark in the loopback mode with the gsmtap filter activated.
13. Back in the first terminal that is in the Airprobe/gsm-receiver/src/python directory, type in the following.

```
./go.sh capture_941.8M_112.cfile
```

14. If everything installed correctly, you should now be able to see the sample GSM data in wireshark.

RECEIVING A LIVE GSM CHANNEL

1. To decode a live channel using a RTL-SDR type the following into terminal where -f specifies the frequency of your GSM signal and -g species the RTL-SDR gain. The -s flag is used here to set the sample rate to 1.0 MSPS, which seems to work much better than the default of 1.8 MSPS.

```
./gsm_receive_rtl.py -s 1e6 -f 936.6M -g 24
```

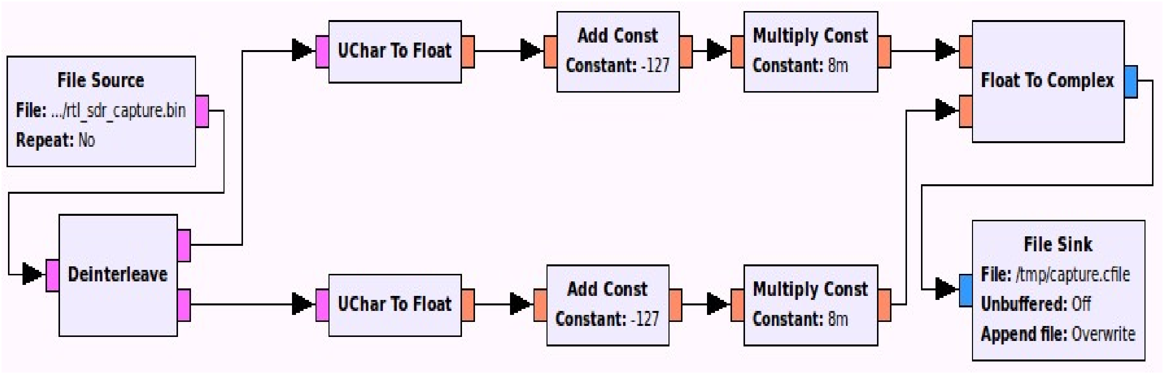
2. A new window will pop up. Within a few seconds you should see hex data scrolling in the terminal window and some GSM data should begin to show in Wireshark. If you get “cannot decode” errors or errors like ERR: conv_decode 11 then try and better center the GSM signal in the RF display window. It may help to use Kalibrate and determine your dongles exact frequency offset to help get a better centering. You should also wait for your dongle to warm up so that the frequency offset stabilizes. You can also try clicking in the center of the GSM signal in the spectrum to try and tune it. If you still have tuning errors try capturing an offline recording as shown below instead.

CAPTURING A CFILE WITH THE RTL-SDR

1. First save a rtl_sdr raw .bin data file using rtl_sdr as shown below where -s is the sample rate, -f is the GSM signal frequency and -g is the gain setting.

```
rtl_sdr /tmp/rtl_sdr_capture.bin -s 1.0e6 -f 938.345M -g 24
```

2. Next, download this GNU Radio Companion (GRC) 3.7 flow graph from http://bit.ly/1ulL6VP. This flow graph will convert the rtl_sdr .bin file into a .cfile. In the flow graph set the file source block to the /tmp/capture.bin file and set the file output in the file sink block to a file called /tmp/capture.cfile. Also, make sure that 'Repeat' in the file source block is set to 'No'.



File Source
File: .../rtl_sdr_capture.bin
Repeat: No

Deinterleave

UChar To Float

Add Const
Constant: -127

Multiply Const
Constant: 8m

UChar To Float

Add Const
Constant: -127

Multiply Const
Constant: 8m

Float To Complex

File Sink
File: /tmp/capture.cfile
Unbuffered: Off
Append file: Overwrite

3. Now execute the flow graph by clicking on the execute button which is the one that looks like the play button. The flow chart *will not* stop by itself when it's done, so you will need to monitor the output cfile file size until it stops growing, then stop the flowgraph by clicking on the stop button. Note that the output .cfile file size will be about four times larger than the input .bin file and that processing may take a while especially if your input .bin file is large.

4. Copy the capture.cfile over the to the airprobe/gsm-receiver/src/python folder.

5. Next open Wireshark as shown before and run the command shown below. GSM data should begin to show in Wireshark.

```
./go.sh capture.cfile 64
```

DECODING GSM MESSAGES AND VOICE

It is actually possible to decode text messages and voice from your own cell phone by getting your personal decryption codes from your SIM card. This process is significantly more involved that the above tutorial, but a good starting point to learning how to do this might be this tutorial http://domonkos.tomcsanyi.net/?p=418.

LISTENING TO AND ANALYZING TETRA SIGNALS

TETRA is an acronym for Terrestrial Trunked Radio and is a digital walkie talkie voice protocol that is often used by government agencies, public safety, rail staff, transport services and the military. The TETRA protocol is in some ways similar to the GSM protocol used in mobile phones. Most countries apart from the United States have a TETRA system in place. A large list of countries that use TETRA can be found at http://en.wikipedia.org/wiki/Terrestrial_Trunked_Radio#TETRA_usage.

Unencrypted TETRA signals can be listened to using an RTL-SDR and Linux software. Even if the TETRA signal is encrypted you can still use some tools to analyze some of the plain text system messages and data sent by the TETRA system just like with GSM. Below we show a tutorial that shows how to listen to and view system data from TETRA signals. Note that most of this tutorial is performed in Linux and we assume that you have some decent Linux experience in installing software and using the command line.

Also note that the TETRA decoder currently requires the older version 3.6 version of GNU Radio to be installed. If you already have GNU Radio 3.7+ installed, we reccommend installing GNU Radio 3.6 on a fresh install of Linux as the two versions may conflict. Another thing to note is that it is easier to install the required software on a 32-bit version of Linux. If you are using a virtual machine to run Linux you will probably already be using a 32-Bit version, but if you are running on bare metal you will probably have a 64-bit version. For this tutorial we suggest using a virtual machine to run a 32-bit Linux on Windows. If using a virtual machine we also strongly suggest to set it to have at least two processors.

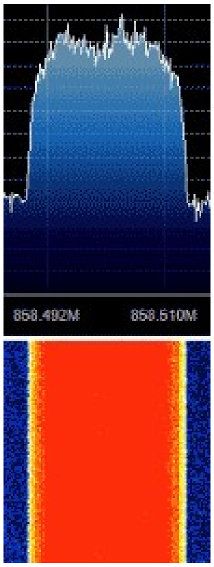
This tutorial is based heavily on the official tutorial written by the author of telive software. That tutorial can be found on the GitHub download of telive at https://github.com/sq5bpf/telive titled telive_doc.pdf. If you haveany issues, we suggest you check out the FAQ in that document.

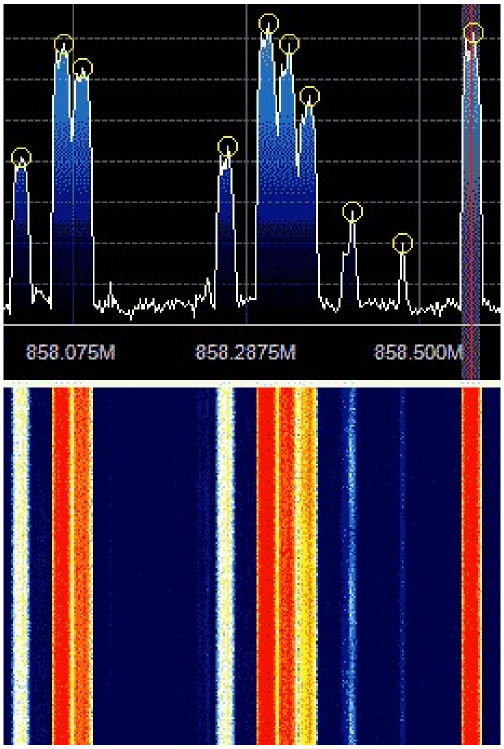
LISTENING TO UNENCRYPTED TETRA

INSTALL DECODER SOFTWARE

1. First using SDR#, your favorite SDR receiver software or a frequency database find the frequency of a TETRA singal in your location. A TETRA

signal is about 25 kHz wide and looks like this zoomed in example image shown below and zoomed out in the second image. There may be several TETRA signals grouped in close proximity to one another. Record the center frequencies of any strong TETRA signals you find.





1. Next install some prerequisites required by the TETRA decoder.

```
sudo apt-get install vorbis-tools
sudo apt-get install sox
sudo apt-get install alsa-utils
sudo apt-get install libncurses-dev
```

2. Now install GNU Radio 3.6. The easiest way to install this older version of GNU Radio is to use Marcus Leech’s installation script with the -o flag. See the GNU Radio installation guide section for more information. Installation of GNU Radio can take a number of hours so be patient.

```
cd ~
wget http://www.sbrac.org/files/build-gnuradio && chmod a+x ./build-gnuradio && ./build-gnuradio -o
```

3. Next install the libosmocore-sq5bpf libraries which is a patched version of the original libosmore libraries.

```
cd ~
git clone https://github.com/sq5bpf/libosmocore-sq5bpf
cd libosmocore-sq5bpf
autoreconf -i
./configure
make
sudo make install
sudo ldconfig
```

4. Now install the osmo-tetra-sq5bpf libraries with the following.

```
cd ~
git clone https://github.com/sq5bpf/osmo-tetra-sq5bpf
cd osmo-tetra-sq5bpf
cd src
make
```

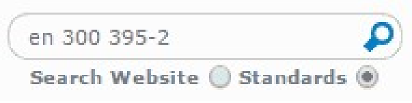
5. Next install telive which is the front end software for the TETRA decoder where YOURUSER.YOURGROUP should be replaced with the username and group that you are currently logged in to on your Linux system. In most cases it can just be YOURUSER.YOURUSER. Run ls -l in your home directory to see what username and group your files are using.

```
git clone https://github.com/sq5bpf/telive
cd telive
make
sudo mkdir /tetra
sudo chown YOURUSER.YOURGROUP /tetra
sh install.sh
```

INSTALL THE TETRA COECS ONTO YOUR SYSTEM

Now we need to install the TETRA codecs so that we can actually hear the TETRA audio being played. Note that if you are using a 64-bit version of Linux you will first need to set your system to use a 32-bit compiler. The following instructions assume that you have a 32-bit Linux OS installed.

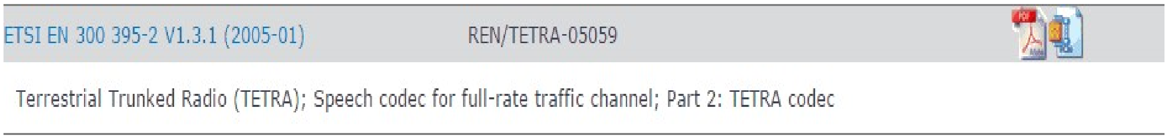
1. Go to http://pda.etsi.org/.
2. In the top right enter as a search term "en 300 395-2" and click the button to select Search Standards.



en 300 395-2

Search Website ○ Standards ●

3. Start the search.
4. Find the search result labelled as REN/TETRA-05059.



ETSI EN 300 395-2 V1.3.1 (2005-01) REN/TETRA-05059

Terrestrial Trunked Radio (TETRA); Speech codec for full-rate traffic channel; Part 2: TETRA codec

5. Click on the winzip icon (looks like a white page with a yellow file cabinet on it) to the right of the result to download en_30039502v010301p0.zip.
6. Move this zip file into ~/osmo-tetra-sq5bpf/etsi_codec-patches.
7. In a terminal browse to ~/osmo-tetra-sq5bpf/etsi_codec-patches.
8. Now run the following commands to unzip the file with lower case letters, patch the codecs, compile and copy the compiled codecs over to the /tetra/bin folder.

```
unzip -L en_30039502v010301p0.zip
patch -p1 -N -E < codec.diff
cd c-code
make
cp cdecoder sdecoder /tetra/bin
```

If you have a 64-bit Linux install you will need to use a seperate procedure to step 8 shown above. These steps show how to set up a 32-bit build environment in a 64-bit Ubuntu OS.

1. Use the following instructions instead of step 8 shown above.

```
sudo su
mkdir /build32
apt-get update
apt-get install debootstrap
debootstrap --variant=buildd --arch i386 wheezy /build32
mkdir /build32/root/codec
```

2. Now move the en_30039502v010301p0.zip file and the codec.diff file to /build32/root/codec and compile the codec using the following commands.

```
cd /home/user/osmo-tetra-sq5bpf/etsi_codec-patches
cp en_30039502v010301p0.zip /build32/root/codec
cp codec.dff /build32/root/codec
chroot /build32
apt-get install unzip
cd root/codec
unzip -L en_30039502v010301p0.zip
patch -p1 -N -E < codec.diff
cd c-code
make
exit
cd /build32/root/codec/c-code
cp cdecoder sdecoder /tetra/bin
exit
```

RUNNING THE DECODING SOFTWARE

Now that everything is installed we can run the software to decode some TETRA signals.

1. Open a terminal window and browse to ~/osmo-tetra-sq5bpf/src and run ./receiver1 1.

```
cd ~/osmo-tetra-sq5bpf/src
./receiver1 1
```

2. Open a second terminal window or tab and open a specially sized xterm window using the following.

```
/usr/bin/xterm -font fixed -bg black -fg white -geometry 203x60
```

3. In the xterm window, browse to ~/telive and run ./rxx.

```
cd ~/telive
./rxx
```

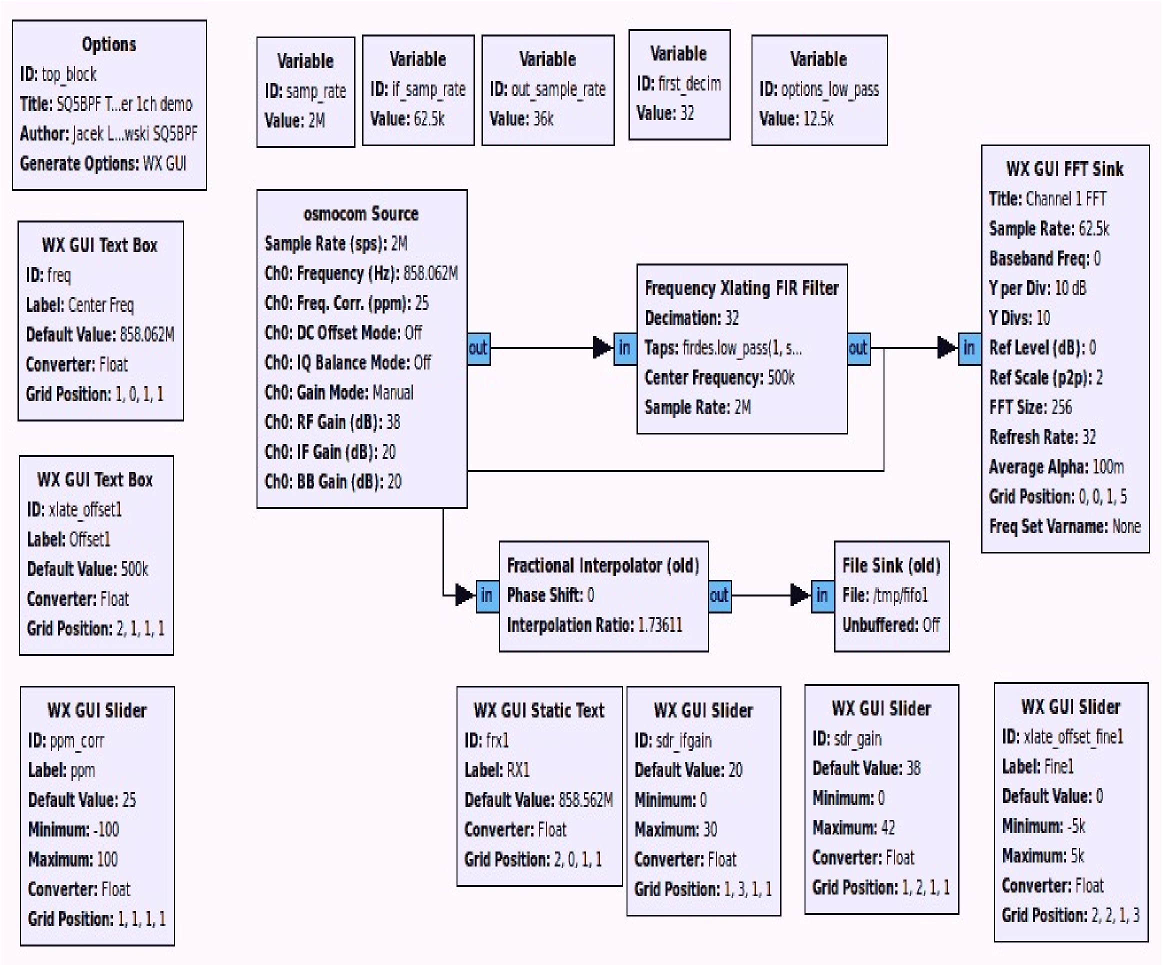
4. Open another terminal window or tab and browse to /tetra/bin and run ./tetrad.

```
cd /tetra/bin
./tetrad
```

5. Open another terminal window or tab and open GNU Radio Companion by typing the following.

```
gnuradio-companion
```

6. In GNU Radio open the telive_1ch.grc file which is found in ~/telive/gnuradio-companion. You may also use telive_1ch_simple.grc which will show a graphical FFT spectrum with click to tune to help you with tuning at the expense of some extra CPU usage.
7. On the left of the GNU Radio flowgraph, double click on the variable labeled Center Freq and change the frequency to the frequency of your TETRA signal that you found earlier subtracted by 500 kHz. E.G. If your TETRA signal was found at 858.562 MHz, you'd type your center frequency as 858.562 MHz - .500 MHz = 858.062 MHz. This is because the default offset value is set to 500 kHz.
8. Double click the variable labelled ppm and input your dongles particular PPM correction value.



Options
ID: top_block
Title: SQ5BPF T...er 1ch demo
Author: Jacek L...wski SQ5BPF
Generate Options: WX GUI

Variable
ID: samp_rate
Value: 2M

Variable
ID: if_samp_rate
Value: 62.5k

Variable
ID: out_sample_rate
Value: 36k

Variable
ID: first_decim
Value: 32

Variable
ID: options_low_pass
Value: 12.5k

WX GUI Text Box
ID: freq
Label: Center Freq
Default Value: 858.062M
Converter: Float
Grid Position: 1, 0, 1, 1

osmocom Source
Sample Rate (sps): 2M
Ch0: Frequency (Hz): 858.062M
Ch0: Freq. Corr. (ppm): 25
Ch0: DC Offset Mode: Off
Ch0: IQ Balance Mode: Off
Ch0: Gain Mode: Manual
Ch0: RF Gain (dB): 38
Ch0: IF Gain (dB): 20
Ch0: BB Gain (dB): 20

Frequency Xlating FIR Filter
Decimation: 32
Taps: firdes.low_pass(1, s...
Center Frequency: 500k
Sample Rate: 2M

WX GUI FFT Sink
Title: Channel 1 FFT
Sample Rate: 62.5k
Baseband Freq: 0
Y per Div: 10 dB
Y Divs: 10
Ref Level (dB): 0
Ref Scale (p2p): 2
FFT Size: 256
Refresh Rate: 32
Average Alpha: 100m
Grid Position: 0, 0, 1, 5
Freq Set Varname: None

WX GUI Text Box
ID: xlate_offset1
Label: Offset1
Default Value: 500k
Converter: Float
Grid Position: 2, 1, 1, 1

Fractional Interpolator (old)
Phase Shift: 0
Interpolation Ratio: 1.73611

File Sink (old)
File: /tmp/fifo1
Unbuffered: Off

WX GUI Slider
ID: ppm_corr
Label: ppm
Default Value: 25
Minimum: -100
Maximum: 100
Converter: Float
Grid Position: 1, 1, 1, 1

WX GUI Static Text
ID: frx1
Label: RX1
Default Value: 858.562M
Converter: Float
Grid Position: 2, 0, 1, 1

WX GUI Slider
ID: sdr_ifgain
Default Value: 20
Minimum: 0
Maximum: 30
Converter: Float
Grid Position: 1, 3, 1, 1

WX GUI Slider
ID: sdr_gain
Default Value: 38
Minimum: 0
Maximum: 42
Converter: Float
Grid Position: 1, 2, 1, 1

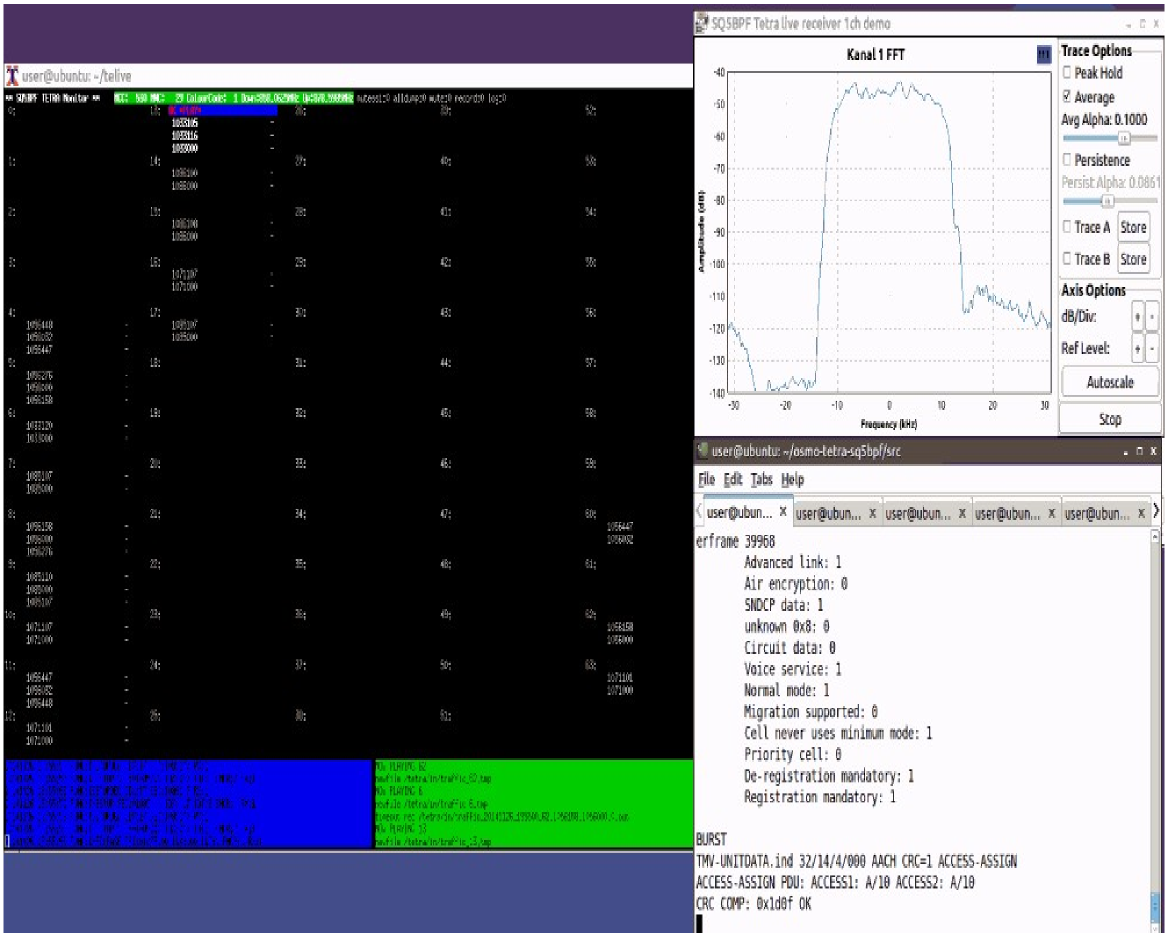
WX GUI Slider
ID: xlate_offset_fine1
Label: Fine1
Default Value: 0
Minimum: -5k
Maximum: 5k
Converter: Float
Grid Position: 2, 2, 1, 3

9. Execute the flowgraph by clicking on the cog icon on GNU Radio Companion toolbar.

At this point you should confirm that you see a strong rectangular TETRA signal in the FFT window that pops up. If you do, switch back to your first terminal window where you ran ./receiver1 1. You should confirm that you see system data scrolling by. If there is no data scrolling by, try adjust the gain and PPM offset in the FFT window.

If data is scrolling and the system is not encrypted you should start to hear voice audio. If a system is encrypted, the terminal window with the system data will show Air encryption: 1. If a system is capable of encryption, the terminal window with the system data will show Air encryption: 1. However, note that even if it shows this, there is still a possibility that encryption has not been enabled.

If you want to log all voice communications you can by pressing “shift+R” in the telive window. This will log .ogg audio files to /tetra/out. You can also enable a text log with “l”. If you happen to close the GNU Radio FFT window and want to run the program again, you will need to restart the ./receiver1 1 program.



```
erframe 39968
        Advanced link: 1
        Air encryption: 0
        SNDCP data: 1
        unknown 0x8: 0
        Circuit data: 0
        Voice service: 1
        Normal mode: 1
        Migration supported: 0
        Cell never uses minimum mode: 1
        Priority cell: 0
        De-registration mandatory: 1
        Registration mandatory: 1

BURST
TMV-UNITDATA.ind 32/14/4/000 AACH CRC=1 ACCESS-ASSIGN
ACCESS-ASSIGN PDU: ACCESS1: A/10 ACCESS2: A/10
CRC COMP: 0x1d0f OK
```

With telive it is also possible to receive two or four TETRA channels simultanouesly, as long as the signals are near enough to one another so that they can be received with one dongle at the same time. To create a two channel receiver follow these steps:

1. Open a terminal window and browse to ~/osmo-tetra-sq5bpf/src and run ./receiver1 1 and ./receiver 2 in seperate terminal windows or tabs.

```
./receiver 1
./receiver 2
```

2. Now follow up to step 2 to 5 in the instructions shown above for a single channel receiver.

3. In GNU Radio open the telive_2ch.grc file which is found in ~/telive/gnuradio-companion.

4. Now in the GNU Radio file you will need to adjust the center and Offset2 frequencies for the TETRA signals you wish to monitor. Leave Offset1 with the 500 kHz value alone and calculate Offset2 for the second signal. To do this you will need to calculate the frequency distance between the center frequency and the second TETRA channel. For example if you have a TETRA signal at 858.061.500 MHz and 858.086500 MHz, you'd calculate your center frequency as 858.0615 - .500 MHz = 857.5615 MHz, and the second offset would be calculated as 858.086500 MHz - 857.5615 MHz = .525 MHz = 525000.

5. Edit the "Center Freq" and "Offset2" labels to the values you calculated in the previous step.

Instructions for setting up a four, five and six channel multi network receiver can be found in the telive_doc.pdf file.

TROUBLESHOOTING

On some installs after a reboot you may get the error "ImportError: /usr/local/lib/libgnuradio-osmosdr-0.0.3git.so.0.0.0: undefined symbol: bladerf_set_correction" on the last line of a big error message when trying to run ./receiver1 1. This is due to some sort of complication with GQRX. The way to solve this problem is to uninstall GQRX and its GNU Radio 3.7 dependancies with the following command.

```
sudo apt-get remove --auto-remove gqrx-sdr
```

ANALYZING TETRA

Even if the TETRA signal is encrypted, it is still possible to look at the system data being sent in a similar way to GSM signals to analyze a TETRA signal.

1. If not already installed, install Wireshark using the following command at a terminal window.

```
sudo apt-get install wireshark
```

2. Now perform the steps above in the Listening to TETRA tutorial section.

3. Open a second terminal window or tab and open wireshark by typing `sudo wireshark` . Because we are running Wireshark as sudo, some errors and

warnings may pop up, but these can be safely ignored. Note that we are running Wireshark as sudo in order to get access to the local loopback interface. Since Airprobe dumps data to a UDP port, we must set Wireshark to listen to this.

4. Under the Start button in Wireshark, first set the capture interface to “Loopback: lo” and then press Start. Then in the filter box, type in gsmtap. This will ensure that only TETRA data is displayed. At this point TETRA system data should be showing in Wireshark.

4.

RADIO ASTRONOMY GUIDE

Combined with a suitable antenna and LNAs the RTL-SDR has been used as a astronomical tool to:

- Observe the hydrogen line and galactic plane.
- Detect meteors entering the atmosphere.

OBSERVING THE HYDROGEN LINE AND GALACTIC PLANE

The Hydrogen line can be observed at 1420 MHz (21cm) by pointing a high gain directional antenna at the interstellar medium. The interstellar medium is the matter that exists in between stars. It is made up of a large amount of neutral hydrogen which emits electromagnetic radiation at 1420.40575177 MHz. Although this emission is rare for a single hydrogen atom, the interstellar medium is made up of a huge amount of neutral hydrogen which makes this emission easy to detect with an appropriate radio and antenna.

By observing the hydrogen line over a moving sky it is possible to detect the galactic plane. This is because there is more hydrogen in the plane of our galaxy than there is in empty space. When the antenna points towards the centre of the galaxy, the intensity of the signal at the hydrogen line frequency will rise.

Additionally, it is also possible to observe the doppler shift of the arms of our Milky Way galaxy galaxy. As the galaxy is spinning, the hydrogen line emissions from the spinning arms will be doppler shifted away from 1420 MHz. When pointed in the right direction, it is possible to observe multiple arms as multiple spectral power peaks near 1420 MHz. This is because inner arms will be spinning at a different speed to the outer arms, causing a difference in their hydrogen line doppler shifts. This is one way that astronomers were able to figure out the shape of our galaxy.

Items you will need to observe the hydrogen line:

- Large satellite dish (>1m), high gain Yagi antenna, Helical antenna or Horn antenna.
- 1 - 2 LNA Preamps and 1 - 3 line amps.
- 1420 MHz bandpass filter.

Marcus Leech was one of the first people to experiment with radio astronomy and the RTL-SDR and he has an excellent writeup here http://www.sbrac.org/files/budget_radio_telescope.pdf. He writes that to observe the hydrogen line with the RTL-SDR you will need at least a 1 meter large dish. In most countries with satellite TV, large used dishes can be found relatively cheaply. If you have difficulty obtaining a dish, other experimenters have had success with non dish based directional antennas such as a 22 element Yagi http://www.y1pwe.co.uk/RAProgs/HLRrtl.pdf which may be cheaper and easier to build. Another had excellent success with a quad 22-element Yagi http://www.y1pwe.co.uk/RAProgs/HLRrtl2U.pdf. Other directional antennas such as helical and horn antennas can also work. An example of a horn antenna used with the RTL-SDR for radio astronomy can be found here http://rishi-patel.blogspot.com/2013/10/summary-of-horn-antenna-project.html. More examples of hydrogen line experiments can be found in the following links:

- 3M Dish: http://lea.hamradio.si/~s57ra/index.php/radio-astronomy/hydrogen-line-with-rtl-sdr
- 3.6M Dish: http://lea.hamradio.si/~s53rm/Radio%20Astronomy.htm

The Hydrogen line is observed by pointing the directional antenna/dish at the sky and tuning the radio to 1420 MHz. Because the hydrogen line signal is very weak, there is a need for several amplifiers after the antenna in order to get enough signal strength such that the receiver can actually detect it. Often two or three line amps are used and one very low noise amplifier is placed directly after the antenna. There is often also a need for good band pass filtering around the 1420 MHz region to remove out of band interference from man made sources. With the right equipment, the hydrogen line is then observed by taking spectral power measurements over a period of time and then integrating or averaging the data.

To average the data you need software that can average the FFT spectrum over a period of time. The RAFFT.exe software from http://y1pwe.co.uk/RAProgs/index.html can be used to do this. There is also the Linux GNU Radio based simple_ra software from https://www.cgran.org/wiki/simple_ra. Useful instructions on the use of simple_ra can be found at http://superkuh.com/rtlsdr.html#simple_ra, though at the moment it seems simple_ra is only available for the older version 3.6 of GNU Radio. Linrad is another program that can be used to average the spectrum, see S53MM's project at http://lea.hamradio.si/~s53rm/Radio%20Astronomy.htm. On this page http://lea.hamradio.si/~s57ra/index.php/radio-astronomy/hydrogen-line-with-rtl-sdr there is a link to a modified version of a spectrum analyzer used for the Beaglebone Black embedded PC that can be used for hydrogen line observation. The rtl_power software that comes with the official osmocom RTL-SDR driver releases can also be used for radio astronomy projects like observing the hydrogen line. The main function that software for observing the hydrogen line needs is that it needs to record frequency power levels over a wide bandwidth for a set amount of time, which is exactly what rtl_power does. Although more advanced software will have options for calibration and options for setting the units to Kelvin which is the units that radio astronomers prefer to work with.

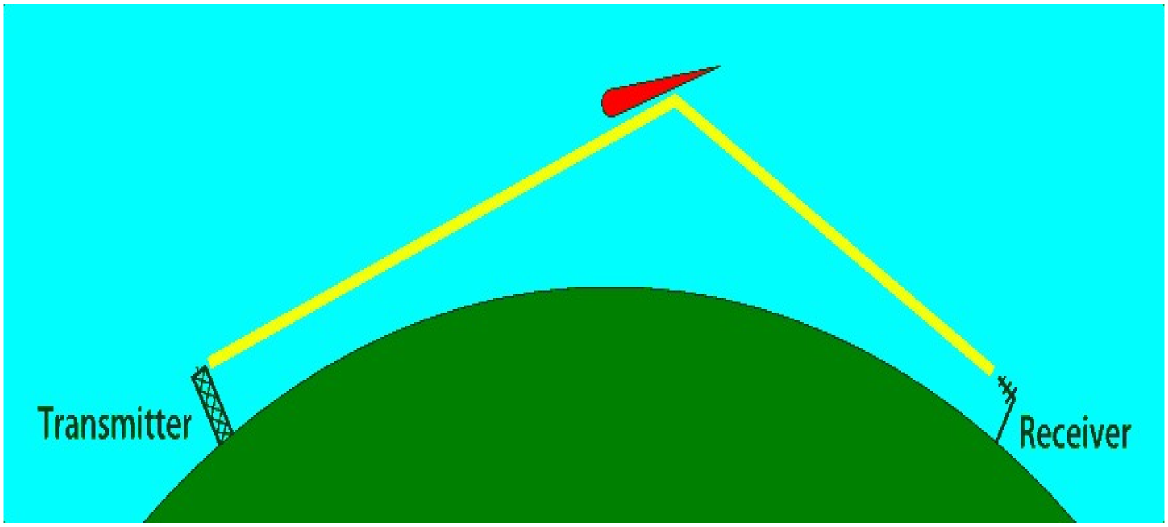
Here are some more technical papers where the RTL-SDR was used for hydrogen line observation:

http://www.haystack.mit.edu/edu/undergrad/VSRT/VSRT_Memos/071.pdf
http://www.haystack.mit.edu/edu/undergrad/srt/pdf%20files/2013_HigginsonRollinsPaper.pdf

METEOR SCATTER DETECTION

The RTL-SDR can be used to detect meteors entering the atmosphere. Meteor scatter detection works by detecting signals from strong but distant broadcast radio stations by listening for radio reflections from meteor trails. As a meteor enters the atmosphere, it ionizes the air behind it leaving a trail of RF reflective ionized air. Note that the RF signals do not reflect off the meteor itself, but rather the ionized air that it leaves behind for a few seconds.

To do meteor detection there needs to be a broadcasting station with a strong carrier signal that is between 300 to 1200 km away from your receiving antenna. This is far enough so that the signal is not received too strongly directly, but close enough so that a meteor in the sky can reflect the signal down to your receiver. Frequencies in the analogue TV spectrum at 50-80 MHz are the best to use for meteor scatter. However as analogue TV transmitters are being reduced due to digital TV switch overs the FM band at 88 - 108 MHz can also be used as an alternative and possibly also digital TV transmissions at UHF frequencies, although UHF frequencies will give poorer results. If you live in Europe the powerful Graves radar in France which transmits at 143.050 MHz is the best choice as it transmits a powerful carrier only signal that is perfect for this type of project. In the USA there are also several powerful radars which can be used, but you will need to determine which ones are near your area. The SNOTEL system that operates in several North American states at 40.670 MHz may be another transmitter to investigate.



As meteor scatter signals can be weak you should use a high gain Yagi-Uda antenna tuned to the band you are monitoring with it pointed towards the sky in the direction of the broadcasting station. Non directional antennas can also work, but may not have enough gain for good performance. Some people have reported success with magnetic loop antennas. An LNA is also desirable. A detected meteor will appear on a RF spectrum waterfall as a curved or sloping blip and will produce a kind of whistle tone if the receiver is set to the USB mode.

A good paper by amateur radio astronomer Marcus Leech describing meteor scatter detection with the RTL-SDR can be found here http://www.sbrac.org/files/meteor_forward_scatter.pdf. Some more papers by a Dr. Morgan that introduce amateur meteor scatter detection in general and with the RTL-SDR can be found at http://www.britastro.org/radio/projects/meteorproj.html. There is also an interesting article by Sky at Night magazine discussing meteor scatter and showing how to build a Yagi for the Graves radar here http://www.britastro.org/radio/downloads/SkyAtNight/BBC_SatN_HOWTOOBSERVEMETEORS.pdf.

While it is fun to monitor meteors in the spectrum by eye in standard SDR software like SDR#, it is more useful to monitor them over a long period of time using additional meteor detection software. Some software will even automatically count the number of meteor “pings” it detects. The software packages Baudline, HROFFT, rtl_power, Spectrum Lab or meteor_detector are discussed below and can be used to monitor a small bandwidth around the transmitter frequency for meteor scatter detection.

RTL_POWER

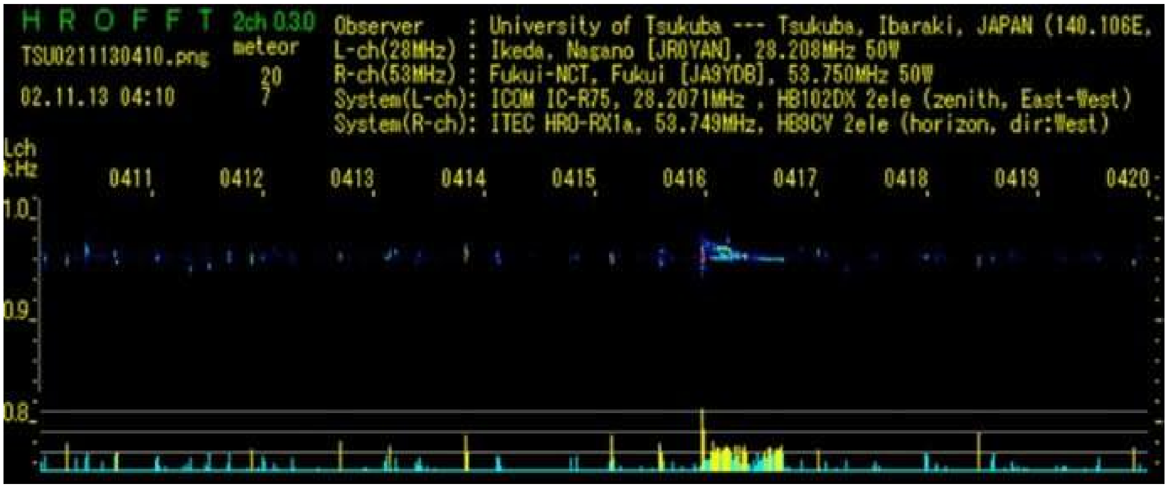
To monitor a small bandwidth around a transmitter at 53 MHz for 1 hour the following rtl_power command could be used. Note that for meteor scatter it is best to use a very small binning size and small bandwidth in order to get a high resolution. See the Heatmap Bandscan Tutorial for more information on using rtl_power.

```
rtl_power -f 52.995M:53.005M:1 -g 50 -i 1 -e 1h meteor_scatter.csv
```

HROFFT

To monitor meteor scatter, a Windows program called HROFFT can be used with the RTL-SDR. HROFFT will monitor the audio spectrum of your tuned frequency and output colorgrams (audio FFT images) every 10 minutes. HROFFT can be downloaded from http://www.bcmeteors.net/index.php/radio-methods/87-radio-detection-basics?showall=&start=4. To use HROFFT follow the instructions below.

1. Download and unzip HROFFT to a folder.
2. Open the header.txt file and enter the information it asks for.
3. Open your favourite SDR receiver like SDR# and set your preferred audio piping method as the output.
4. Tune to the centre frequency of your distant transmitter station, then tune a few kHz higher. Set the receive mode to USB and increase the tuned area bandwidth to the maximum.
5. Open HROFFT.
6. Adjust the volume in SDR# and the signal level in HROFFT so that the blue signal levels shown at the bottom of the graph sit below the first line. A meteor will be counted as detected if the blue signal becomes stronger than the first horizontal bar so you may wish to adjust the volume and signal level settings accordingly so that only actual meteor reflections trigger this bar.
7. After seeing a few meteor echoes, you may wish to adjust the audio bandwidth so that it covers just the bandwidth of the signals you are detecting. The grey bar on the left vertical axis shows the bandwidth, which is adjustable by the f1 and f2 settings. This will help detect meteors better as the signal strength is taken from the sum of signals from this bandwidth.



```
H R O F F T  2ch 0.30  Observer      : University of Tsukuba --- Tsukuba, Ibaraki, JAPAN (140.106E,
TSU0211130410.png  meteor  L-ch(28MHz)  : Ikeda, Nagano [JR0YAN], 28.208MHz 50W
                     20    R-ch(53MHz)  : Fukui-NCT, Fukui [JA9YDB], 53.750MHz 50W
02.11.13 04:10        7    System(L-ch): ICOM IC-R75, 28.2071MHz , HB102DX 2ele (zenith, East-West)
                           System(R-ch): ITEC HRO-RX1a, 53.749MHz, HB9CY 2ele (horizon, dir:West)
```

Meteors will usually appear as short blips, but larger echoes can occur from big meteors. Long trails may indicate radio reflections from aircraft. More information about analyzing HROFFT images can be found here http://www.amro-net.jp/about-hro/analyze-hrofft.htm.

BAUDLINE

On Linux the frequency domain FFT baudline can also be used for recording meteor scatter activity. Use it to search for blips in the waterfall. Download Baudline from http://www.baudline.com/download.html. Baudline can be started using the following example.

```
rtl_sdr -f 91M -s 2400000 -g 50 - | ./baudline -reset -samplerate 2400000 -channels 2 -format u8 -quadrature
-stdin
```

SPECTRUM LAB

Another excellent software program for use in detecting meteors is Spectrum Lab which can be downloaded freely at http://www.qsl.net/dl4yhf/spectra1.html. A guide to using Spectrum Lab for meteor detection with the RTL-SDR can be found in this document http://www.britastro.org/radio/downloads/TECHNIQUES_FOR_USING_THE_RTL_Dongle.pdf. Some additional documents on meteor detection by the same author of the previous paper can be found at http://www.britastro.org/radio/projects/meteorproj.html.

METEOR_DETECTOR

Meteor_detector is a GNU Radio program which can be used to detect meteor scatter. It used to be available from https://www.cgran.org/, but it seems that the site is currently down at the time of writing this (January 2015). We suggest you check back regularly for updates, or contact patchvonbraun on IRC freenode ##rtlsdr for the code.

SOLAR INTERFEROMETRY

The RTL-SDR can be used as a solar interferometer for measuring changes in the sun, such as sunspots. Interferometry is simply a method that can be used to connect multiple smaller radio telescope antennas together in a way that they approximate a larger antenna.

Superkuh has an 11 GHz solar interferometer that he has been building. Details about that can be found at http://superkuh.com/rtlsdrinterferometer.html.

OTHER SIMPLE RADIO ASTRONOMY EXPERIMENTS

More simple radio experiments can be performed using a simple 18 inch directv satellite dish with LNB and an RTL-SDR. These components can be used to make something that has been termed the Itty Bitty Radio Telescope. A low noise block (LNB) amplifies and converts GHz high frequencies down to lower frequencies which the RTL-SDR can receive. For example, an LNB might convert a 12 GHz signal down to 1.2 GHz.

Using such a simple system the difference in the noise floor between the empty sky, the sun, or a persons body can be detected. Experiments can be found here http://www.setileague.org/articles/lbt.pdf.

A link to an Itty Bitty Radio Telescope based on the RTL-SDR can be found here http://www.stargazing.net/david/radio/itty_bitty_radio_telescope.html.

As for other software there is also the RTL Bridge software available at http://cygnusa.blogspot.com/2014/07/rtl-meets-radio-skypipe-and-rs.html. This software allows an RTL-SDR dongle to interface with two other programs by the author of RTL Bridge called Radio-SkyPipe and Radio-Sky Spectrograph. Radio-Sky Spectrograph allows waterfall spectral visualization of phenomena such as solar flares while Radio-SkyPipe is a strip chart data collection program which allows sharing digitally sampled analogue data in real-time over the internet or network. A strip chart is essentially a frequency power plot.

DOPPLER PASSIVE RADAR

It is possible to use the RTL-SDR as a type of simple doppler scatter passive radar system. Doppler scatter radar works in a very similar way to meteor scatter detection that is described in the section above about radio astronomy. By tuning to a distant but powerful RF transmitter, aircraft and other objects can be observed from the reflections they send towards your receiver. Rtl_power can be used as a simple tool for passive radar with the RTL-SDR.

First, find the frequency of a very strong broadcast RF transmitter that is about 300 to 1200 kilometres away from your radios position. Examples of such transmitters might be broadcast FM stations or TV stations. Ideally, the frequency should be in the VHF band, but UHF frequencies can also work. The signal should be strong enough so that you can just barely see the pilot tone on the waterfall but weaker signals may still work.

Using rtl_power, monitor a small bandwidth around the transmitter pilot frequency. For example, the following will monitor a small bandwidth with high resolution for about 1 hour. After monitoring finishes, you can plot a waterfall to see the radar blips. See the Heatmap band scan section for information on plotting a waterfall using rtl_power.

```
rtl_power -f 674.230M:674.233M:1 -g 50 -i 1 -e 1h radar.csv
```

You may be interested in this discussion on a non-english forum which discusses the different "signatures" different types of aircraft leave on a doppler reflection http://www.oh7ab.fi/foorumi/viewtopic.php?f=21&t=295&sid=2abe828d8ad8379eb9d4447d2cb1dc15&start=80.

DUAL COHERENT PASSIVE RADAR

Using two RTL-SDR dongles running under the same clock source it is possible to create a "dual coherent channel" receiver (aka multistatic receiver, coherent multichannel receiver), which can be used for creating a passive radar that shows planes on a display similar to that of a real radar system. Dual coherent passive radar also works by using a strong distant transmitter to provide the reflections. But, by having two RTL-SDR dongles connected to the same clock source and two directional antennas used together with some clever math, a passive radar can be created. The clock source can simply be from another RTL-SDR dongle. Simply take two RTL-SDR dongles, remove the clock from one dongle, and use a cable to connect the clock from the other dongle to the dongle with the clock removed.

Unfortuneately, we havn't seen any code for this type of project publically released yet, only demo videos are available, see the additional links.

Additional Links

http://www.rtl-sdr.com/passive-radar-dual-coherent-channel-rtl-sdr/

http://www.rtl-sdr.com/rtl-sdr-based-coherent-multichannel-receiver/

http://www.rtl-sdr.com/rtl-sdr-based-passive-multistatic-radar-used-track-aircraft/

HF MODES DECODING GUIDE

To receive HF signals you will need an RTL-SDR dongle with an upconverter, have the direct sampling mod enabled or be using one of the experimental HF drivers. See the Receiving LF/MF/HF Guide section earlier in the book for more information. Alternatively, you could use a HF capable SDR radio such as the Funcube Dongle or SoftRock SDR. You will also need to have a HF antenna, most VHF and UHF antennas will not pick up HF signals at all. See the Antenna Guide section for more information about HF antennas.

SHORTWAVE BROADCAST RADIO

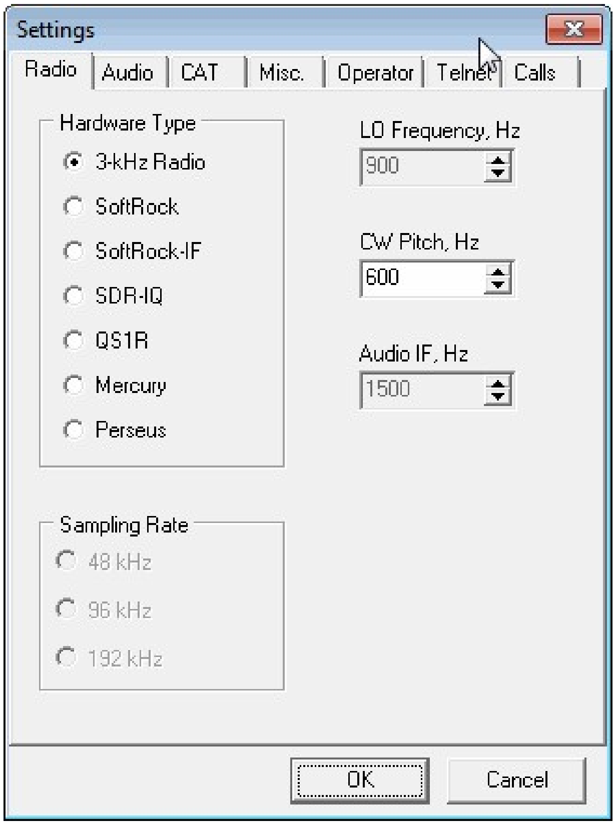
Shortwave broadcast radio is simply international AM radio that is sent over HF. All you need to receive this is a decent HF antenna and a HF enabled RTL-SDR. Sometimes the hobby of searching for and listening to new shortwave radio stations is called SWLing (short wave listening). A list of common shortwave radio stations can be found here http://support.radioshack.com/support_tutorials/communications/swave-6.htm.

CW (MORSE CODE)

CW is an acronym for Continuous Wave and is a communications method more commonly known as Morse Code. It is often used by ham radio hobbyists for making long distance contacts. Morse code is easy for humans to decode, but it actually quite difficult for computers to decode accurately. Nevertheless, there are several decent CW decoders available.

All decoders require the audio to be piped into them from a general purpose SDR receiver. Most SDR receiver programs will have a CW mode, but if they don’t then using USB mode is also okay.

The best CW decoder is called CWSkimmer. CWSkimmer costs $75 but comes with a free 30 day trial. CW Skimmer can be downloaded from http://www.dxatlas.com/cwskimmer/. To use CW Skimmer set the hardware type to 3-kHz Radio or SoftRock IF in the settings menu. Set the signal I/O Device to your audio piping method. Also ensure the Audio output button 🔊 in the main window is set to off (not pressed in).



Hardware Type

● 3-kHz Radio
○ SoftRock
○ SoftRock-IF
○ SDR-IQ
○ QS1R
○ Mercury
○ Perseus

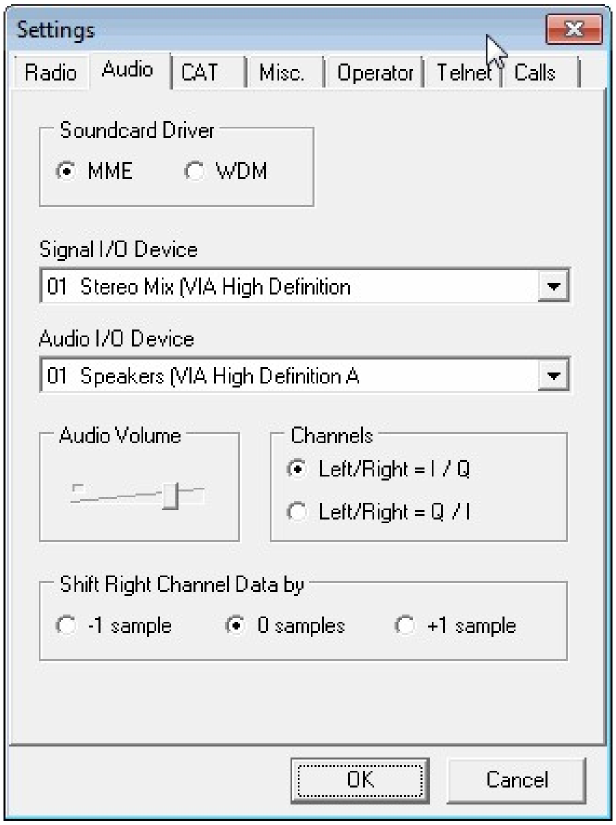
Sampling Rate

○ 48 kHz
○ 96 kHz
○ 192 kHz

LO Frequency, Hz: 900

CW Pitch, Hz: 600

Audio IF, Hz: 1500



Soundcard Driver

● MME ○ WDM

Signal I/O Device: 01 Stereo Mix (VIA High Definition

Audio I/O Device: 01 Speakers (VIA High Definition A

Audio Volume

Channels

● Left/Right = I / Q
○ Left/Right = Q / I

Shift Right Channel Data by

○ -1 sample ● 0 samples ○ +1 sample

Free alternative CW decoders with decent performance are CW Decoder from http://www.hotamateurprograms.com/downloads.htm, CWGet from http://www.dxsoft.com/en/products/cwget/ and also MultiPSK. Pipe audio from SDR# or another SDR receiver into these programs to enable CW decoding.

Fldigi from http://www.w1hkj.com/Fldigi.html and MultimonNG are also capable of decoding CW, however their decoders are not as good as CW Skimmer or the ones mentioned above.

A list of CW frequencies can be found at http://www.arrl.org/band-plan.

RTTY

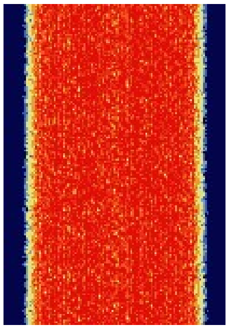
RTTY is an acronym for Radioteletype and is a way for radio amateurs to send text messages over radio, similar to CW but at faster speeds. A list of RTTY frequencies can be found at http://www.arrl.org/band-plan.

RTTY can be transmitted at various baud speeds and it is up to the receiver to choose the correct speed. The most common baud rate used by amateur radio hobbyists is 45 baud. Weather services and other commercial services also transmit information using RTTY but they most commonly use 50 baud. NATO military services prefer transmission at 75 or 100 baud. Note that military transmissions are usually encrypted.

A good program for RTTY is TrueTTY http://www.dxsoft.com/en/products/truetty/. Be sure to set the baud rate under the Speed menu.

STANAG 4285

STANAG 4285 is a HF digital text data link mode used mostly by military naval services. The most common transmitter of STANAG 4285 signals is the French Navy who continuously transmit a test signal that repeats the French equivalent of the quick brown fox “VOYEZ VOUS LE BRICK GEANT QUE J EXAMINE PRES DU GRAND WHARF”. These signals can be found all over the HF spectrum and have a bandwidth of 2500 Hz. A list of STANAG 4285 frequencies can be found at http://qrg.globaltuners.com/?q=STANAG+4285&s=1. The image below shows what a Stanag 4285 signal looks like zoomed in.

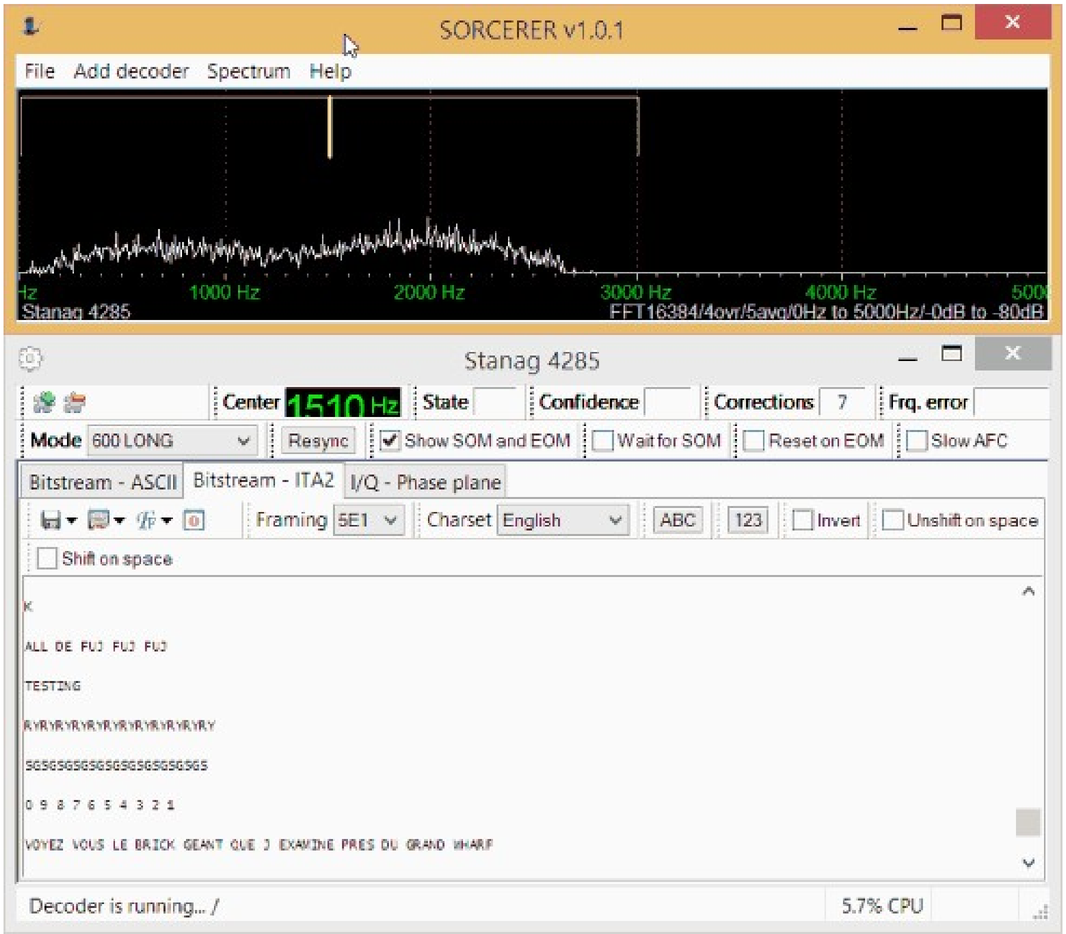


Tips - To test the software mentioned below go to the www.sigidwiki.com signal identification guide and play the example STANAG 4285 audio using Windows stereo mix as your default audio piping method. The programs should be able to decode from this example audio.

DECODING STANAG 4285: SORCERER

Sorcerer is a decoder that used to be advanced commercial software, but is now abandonware that can be freely downloaded from http://www.radioaficion.com/HamNews/archivo/vagabundos-del-dial/5814-sorcerer-decoder.html.

1. Tune to a STANAG 4285 signal in your favourite general SDR program using USB with a bandwidth of around 2500 Hz. Output the audio to you preferred audio piping method.
2. To use Sorcerer, download the zip file from the above link and extract the single exe file to a folder.
3. Open Sorcerer by double clicking on the exe file. Go to **File->Options** and choose your audio piping method in the Soundcard pull down box.
4. Go to **Add Decoder ->PSK->Stanag 4285**. Double click on Stanag 4285 to start that decoder.
5. Centre the yellow bar by clicking in the middle of the signal. The right edge of the yellow bar should be at of near 3000 Hz and the left yellow edge should be near 0. You may need to experiment with the tuning of the yellow bar to start getting decodes.
6. For the most common STANAG 4285 signals choose the Mode as 600 LONG and the Framing as 5N1. Go to the Bitstream - ITA2 tab to see the decoded output after a few seconds. Also note that some signals are sent using 5E1 framing. If you continuously receive gibberish text, try 5E1 framing.



Center 1510 Hz
Corrections 7
Mode 600 LONG
☑ Show SOM and EOM
□ Wait for SOM
□ Reset on EOM
□ Slow AFC
Framing 5E1
Charset English
□ Invert
□ Unshift on space
□ Shift on space

```
K

ALL DE FUJ FUJ FUJ

TESTING

RYRYRYRYRYRYRYRYRYRYRYRY

SGSGSGSGSGSGSGSGSGSGSGSGS

0 9 8 7 6 5 4 3 2 1

VOYEZ VOUS LE BRICK GEANT QUE J EXAMINE PRES DU GRAND WHARF
```

Decoder is running... /

DECODING STANAG 4285: SIGMIRA

Another program called Sigmira can also be used to decode STANAG 4285. Sigmira can be downloaded from http://www.saharlow.com/technology/sigmira/ in both Windows and Linux versions. After installing Sigmira you will need to download the updated features.dat file from the website and place it in the Sigmira folder as is described on the download website. If this isn’t done Sigmira will fail to open. If at a later date Sigmira fails to open, you may need to redownload and replace the features.dat file.

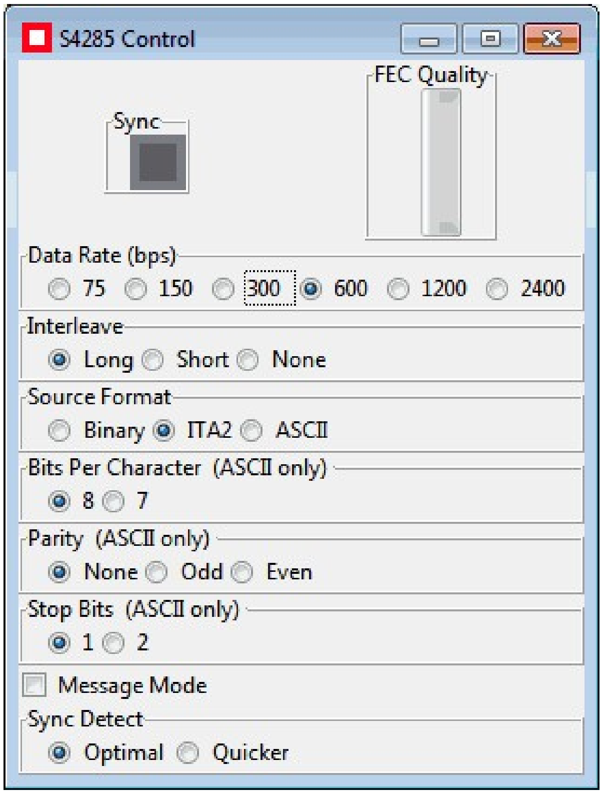
1. To use Sigmira, open it and go to **Source -> Soundcard**. Sigmira is now listening to your default audio piping method set in your Windows recording options. Note that there appears to be a bug in Sigmira which mutes the audio piping method for some reason. You will need to go into your Windows sound properties for the audio pipe to unmute it if this happens.

2. Under Mode, select S4825.

3. Tune to a STANAG 4285 signal in your favourite general SDR program using USB with a bandwidth of around 2500 Hz. Output the audio to the audio piping method used.

4. If the STANAG 4285 control panel is not visible go to **View->S4285 Control** and click it. Also ensure the Rx Text window is showing under **View->Rx Text**.

5. In the S4285 control panel, for the French Navy signal select the "Data Rate" as 600 bps, the "Interleave" to Long and the "Source Format" as ITA2 as shown in the screenshot below.



Data Rate (bps)
○ 75 ○ 150 ○ 300 ● 600 ○ 1200 ○ 2400

Interleave
● Long ○ Short ○ None

Source Format
○ Binary ● ITA2 ○ ASCII

Bits Per Character (ASCII only)
● 8 ○ 7

Parity (ASCII only)
● None ○ Odd ○ Even

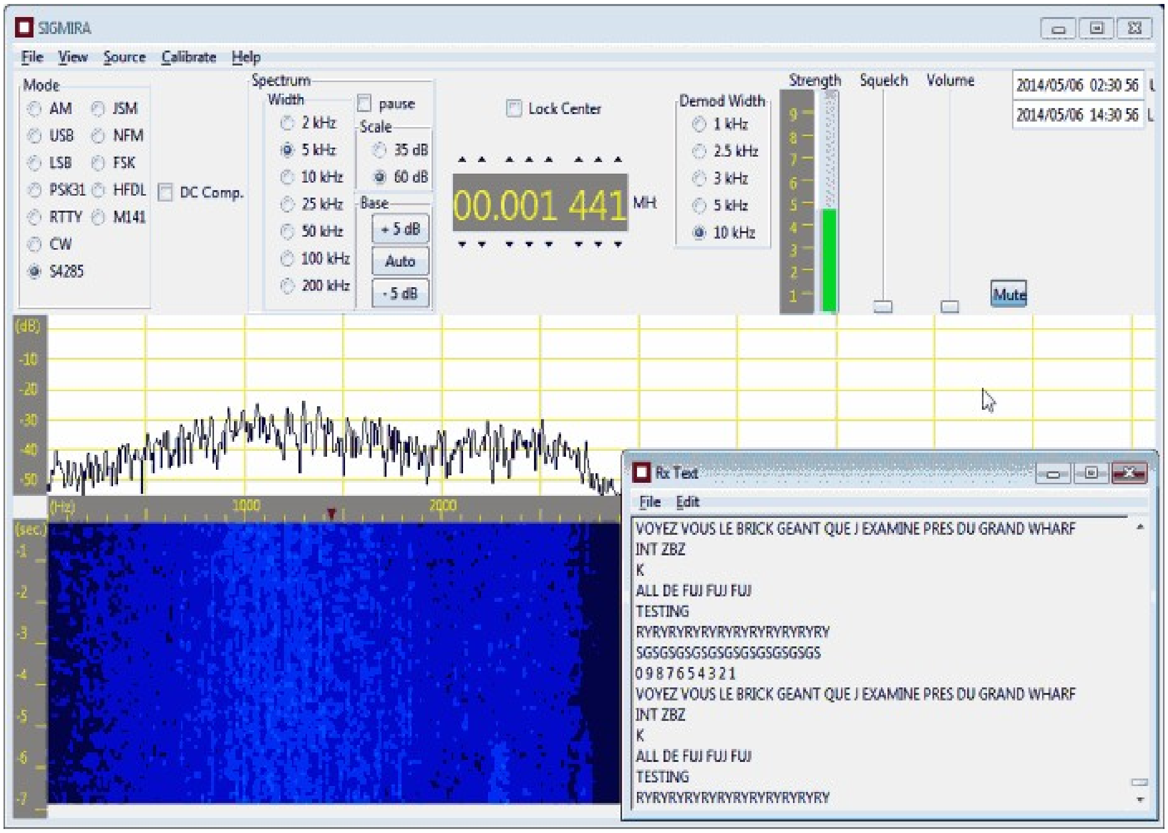
Stop Bits (ASCII only)
● 1 ○ 2

□ Message Mode

Sync Detect
● Optimal ○ Quicker

6. Now in the main SIGMIRA window set the spectrum width to 5 kHz and the Demod Width to 3 kHz or higher.

7. Click in the centre of the signal shown in the audio spectrum and waterfall display to place a red arrow in the middle of the signal. If centred correctly the Sync box in the S4285 control panel will light green, the FEC quality bar will turn green and messages will show in the Rx Text window. It may take some trial and error to correctly centre the red arrow.



Mode: ● S4285

Width: ● 5 kHz

Scale: ● 60 dB

00.001 441 MH

Demod Width: ● 10 kHz

2014/05/06 02:30 56

2014/05/06 14:30 56

```
VOYEZ VOUS LE BRICK GEANT QUE J EXAMINE PRES DU GRAND WHARF
INT ZBZ
K
ALL DE FUJ FUJ FUJ
TESTING
RYRYRYRYRYRYRYRYRYRYRYRYRY
SGSGSGSGSGSGSGSGSGSGSGSGS
0 9 8 7 6 5 4 3 2 1
VOYEZ VOUS LE BRICK GEANT QUE J EXAMINE PRES DU GRAND WHARF
INT ZBZ
K
ALL DE FUJ FUJ FUJ
TESTING
RYRYRYRYRYRYRYRYRYRYRYRYRY
```

DECODING STANAG 4285: OTHER DECODERS

MultiPSK can also be used to decode STANAG 4285, however it’s decoding performance is not as good as Sigmira or Sorcerer.

SSTV

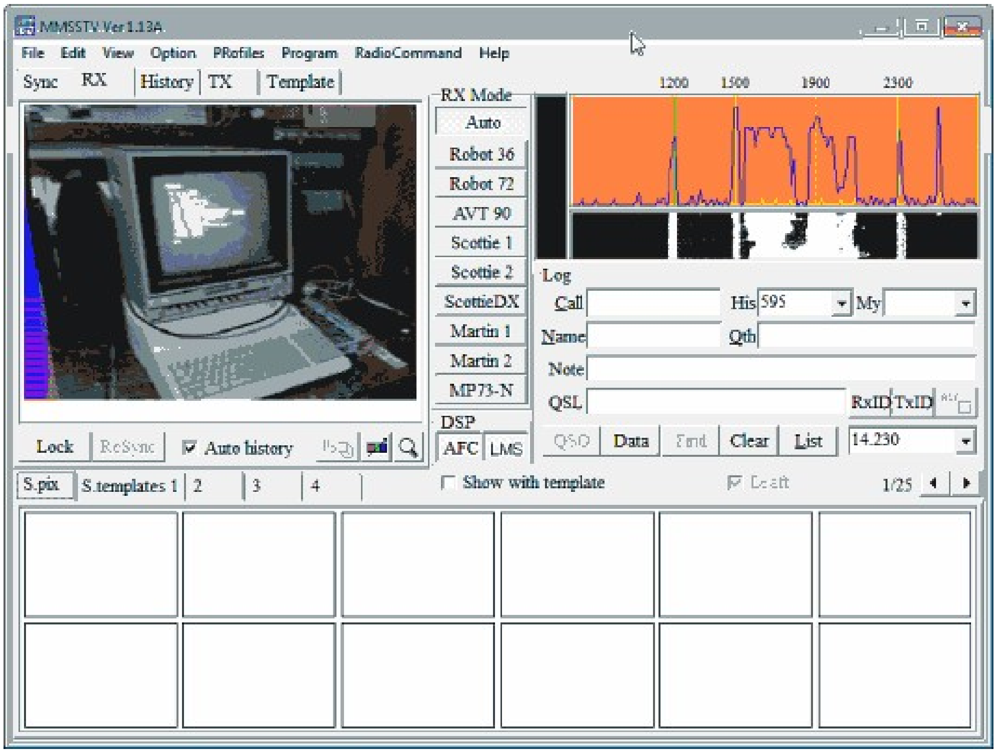
SSTV is an acronym for Slow Scan TV and is a mode used by amateur radio hobbyists to send small calling card images mainly on HF. There are several SSTV formats available, with Scottie S1 and Martin S1 being the most common. SSTV is most commonly found on or near the following frequencies in MHz.

- 3.730
- 7.171
- 14.230 (most popular freq)
- 21.340
- 28.680
- 145.5

SSTV can also be found transmitting in some countries at around 245.800 to 294.075 MHz by Brazilian and Mexican radio pirates who hijack military satellites. The pirates hijack a military satellite transponder to transmit SSTV images and other signals. More information on this activity can be found here http://www.qsl.net/py4zbz/ec/ec.htm.

A good free SSTV decoder is MMSSTV from http://hamsoft.ca/pages/mmsstv.php. MMSSTV will automatically detect the SSTV mode used. To decode SSTV using MMSSTV, use USB mode in your SDR receiver and pipe the audio to MMSSTV using your preferred audio piping method.

Some SSTV audio samples can be found at https://www.youtube.com/watch?v=Ke5UB2HLaZ4.


MMSSTV Ver 1.13A

His 595

14.230

☑ Auto history

□ Show with template

☑ Draft

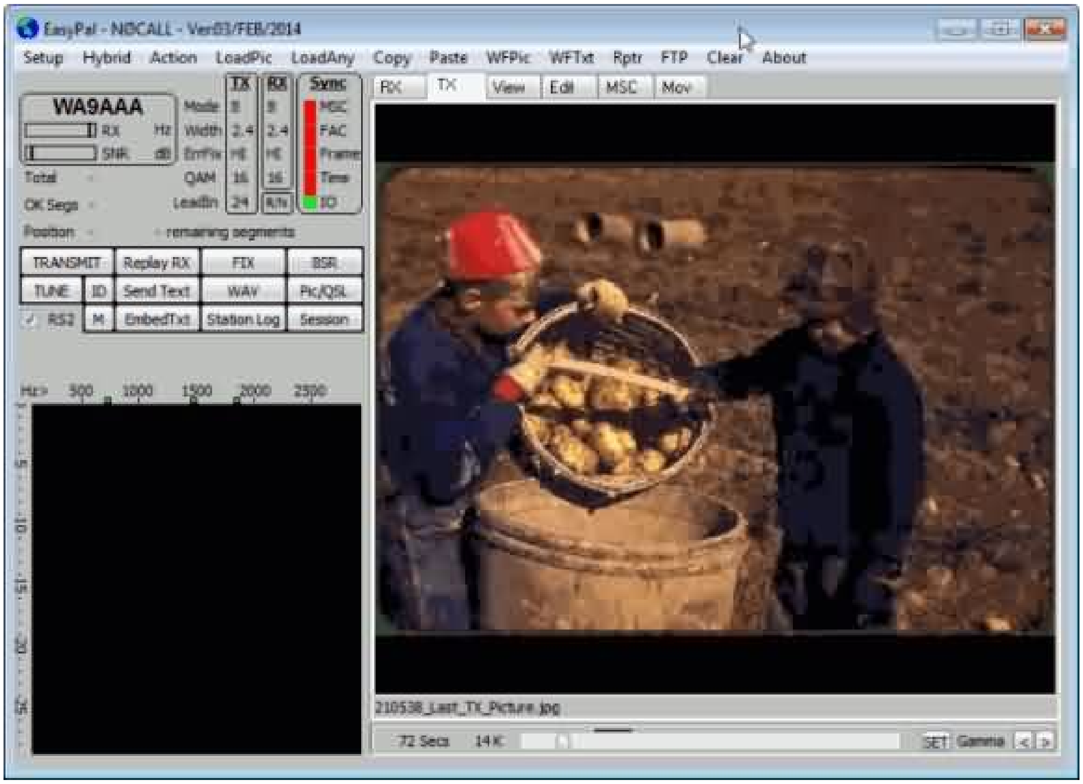
1/25

DIGITAL SSTV

Digital SSTV is a modern SSTV mode that uses a digital signal to transmit high quality images on HF. Digital SSTV is most commonly found on or near the following frequencies in MHz.

- 3.733
- 7.058 (Europe)
- 14.233 (most popular freq)
- 21.337

A good free digital SSTV decoder is EasyPal which can be downloaded from http://www.kc1cs.com/.



WSPR

WSPR (pronounced as ‘whisper’) is an acronym for weak signal propagation report. It is a type or signal that is used for sending and receiving very weak signals on HF between amateur radio operators.

An excellent WSPR tutorial for the Funcube Dongle (but also applicable to the RTL-SDR with upconverter/direct sampling mod) can be found here http://www.nerdsville.blogspot.com/2013/03/wspr-using-funcube-dongle-pro.html.

MARINE HF MODES

Marine HF modes are signals that are generally intended for mariners out at sea. These modes are mainly used to send things like weather reports, but it can be interesting to receive them on land. Below we present some common marine HF modes.

WEATHERFAX (HFFAX) DECODING GUIDE

WeatherFax (sometimes also known as WEFAX or HFFAX) is a HF mode used by meteorological agencies to transmit images of weather information charts. It is a service mainly used by mariners at sea. There is also a Japanese newspaper known as “Kyodo News” which transmits entire pages of its daily newspaper via HFFAX. Weatherfax frequencies are different in every country but there is a large document of radio fax frequencies for many countries that can be found from the NOAA website here http://www.nws.noaa.gov/om/marine/rfax.pdf.

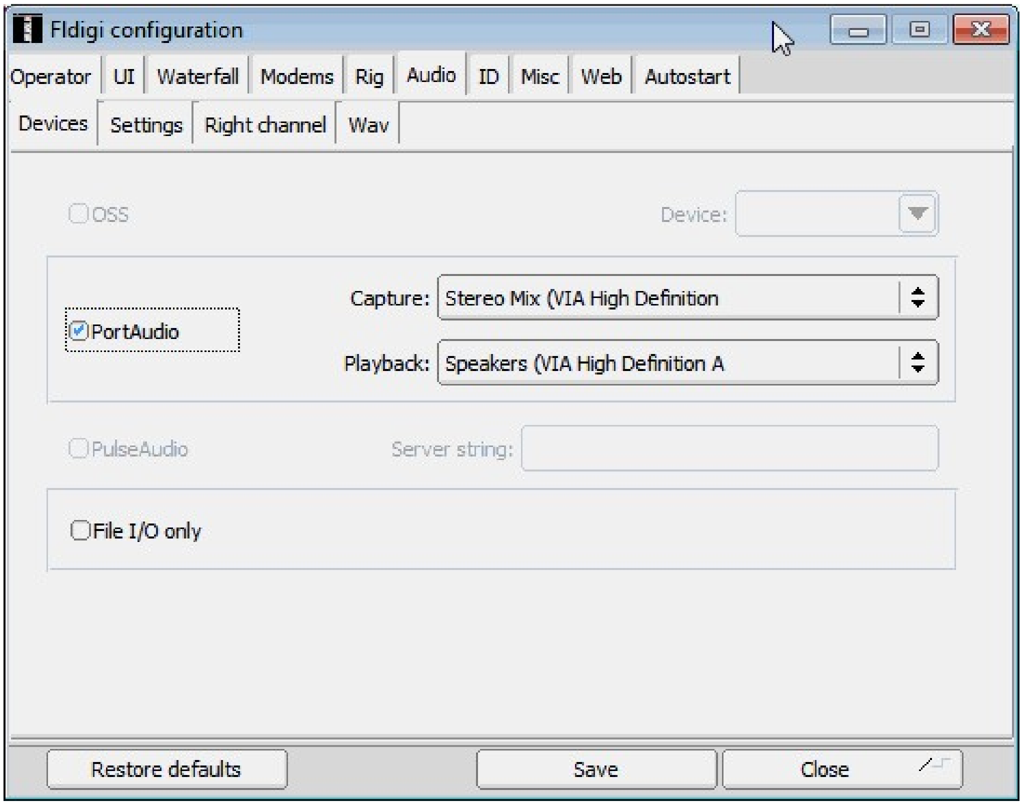
To decode HFFAX use a general SDR receiver like SDR# or SDR-Radio and pipe the audio into a decoder program called Fldigi which is capable of decoding and displaying HFFAX images. Fldigi can be downloaded from http://www.w1hkj.com/download.html. To receive weatherfax follow these instructions.

1. Open your favorite SDR receiver and set the audio output method to your preferred audio piping method.

2. Tune to a weatherfax signal in your SDR receiver. Use Upper Side Band (USB) mode and set the bandwidth to about 1900 Hz.

3. Open Fldigi and go to **Configure -> Sound Card**.

4. Change the Capture audio device to the audio piping method you have chosen.



□ OSS Device:

☑ PortAudio

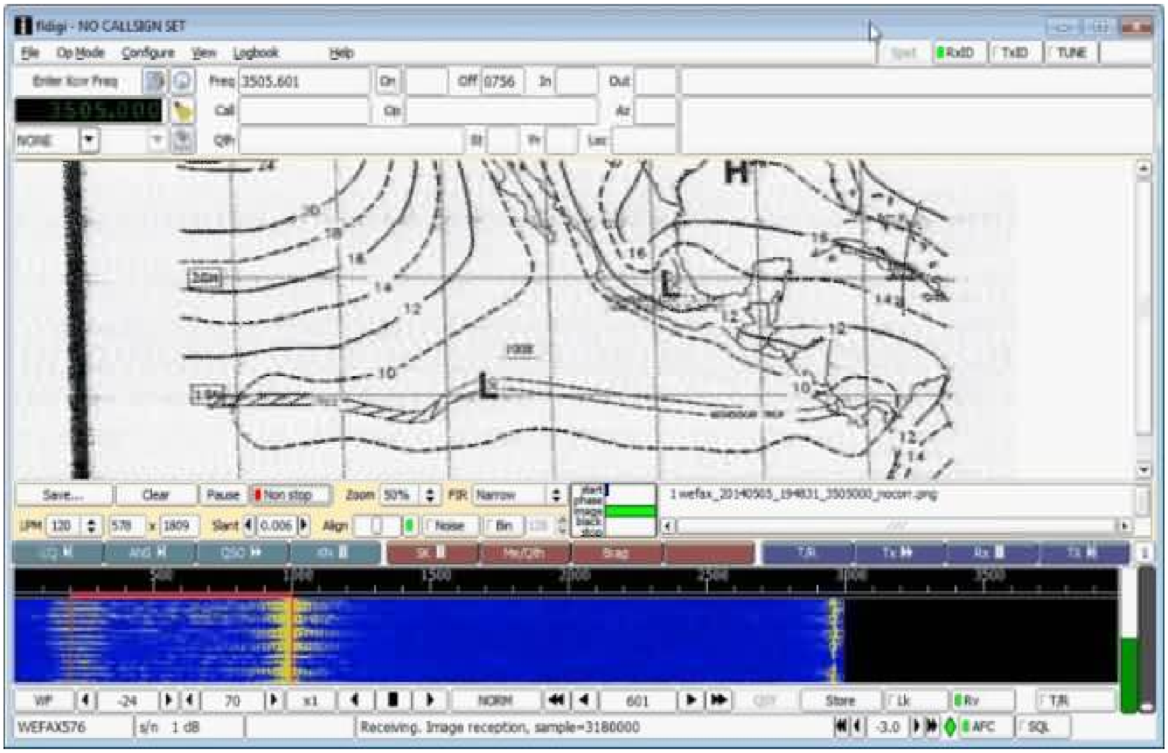
Capture: Stereo Mix (VIA High Definition

Playback: Speakers (VIA High Definition A

□ PulseAudio Server string:

□ File I/O only

5. Now go to **Op Mode -> WEFAX -> WEFAX-IOC576**.
6. Click the Automatic Frequency Control (AFC) button AFC in the bottom right corner so that the light is green.
7. When a WEFAX signal starts to play Fldigi will automatically start decoding.



Receiving. Image reception, sample=3180000

8. You may want to adjust the Slant option if you find that the image is slanted to one side or another.

If for some reason fldigi failed to automatically start, you can click the ‘Non-stop’ button to manually start decoding.

For the next few remaining modes we will introduce them first and then show how to decode them in a program called MultiPSK in a later section.

SYNOP

Synop (Surface Synoptic Observations) is a type of RTTY signal used on HF by the German meteorological service Deutscher Wetterdienst for sending out weather reports. The reports are generally intended for ships at sea. SYNOP message use 5 digit codes to encode a set number of weather parameters. It can usually be received around Europe. When tuning to a synop signal you should use LSB mode. Some SYNOP frequencies and transmission times are shown below.

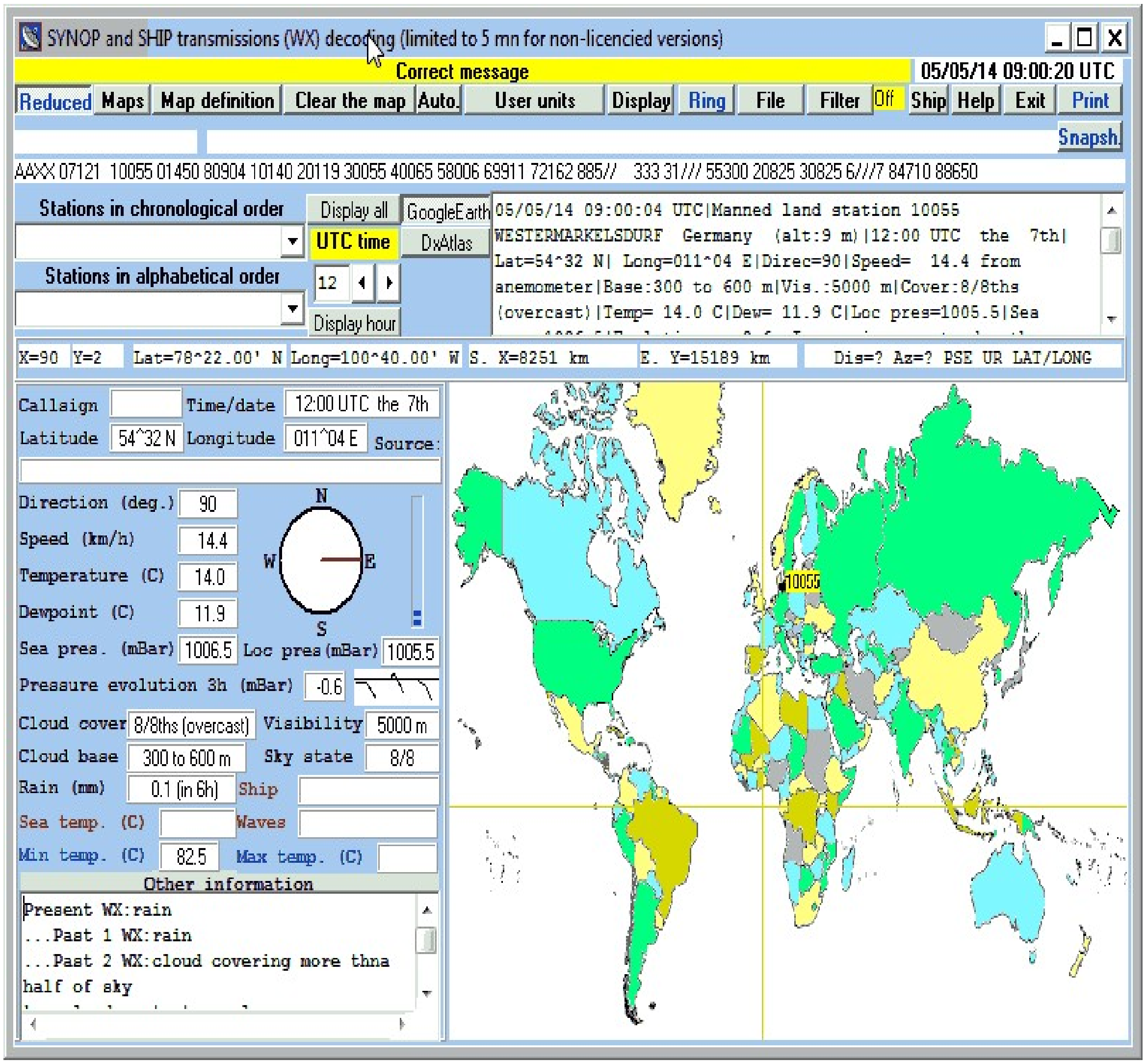
FREQ (kHz)	Station	Time (UTC)
147.300	DDH 47	0500 - 2200

4583.000	DDK 2	0000 - 2400
7646.000	DDH 7	0000 - 2400
10100.800	DDK 9	0000 - 2400
11039.000	DDH 9	0500 - 2200
14467.300	DDH 8	0500 - 2200

A typical raw SYNOP message might look like:

```
09/04/14 11:44:04 UTC|Manned land station 06180 EKCH  KOEBENHAVN\KASTRUP  Denmark  (alt:5
m)|18:00 UTC  the  7th| Lat=55^38 N| Long=012^40 E|Direc=130|Speed=  11.1 from anemometer|Base:600
to 1000 m|Vis.:30 km|Cover:3/8ths|Temp= 15.0 C|Dew=  9.1 C|Loc pres=1006.4| Message:AAXX 07184
06180 02580 31306 10150 20091 30064 41|
```

Here is an example of a SYNOP message displayed in the MultiPSK GUI.



```
AAXX 07121  10055 01450 80904 10140 20119 30055 40065 58006 69911 72162 885//   333 31/// 55300 20825 30825 6///7 84710 88650
```

```
05/05/14 09:00:04 UTC|Manned land station 10055
WESTERMARKELSDURF  Germany  (alt:9 m)|12:00 UTC  the  7th|
Lat=54^32 N| Long=011^04 E|Direc=90|Speed=  14.4 from
anemometer|Base:300 to 600 m|Vis.:5000 m|Cover:8/8ths
(overcast)|Temp= 14.0 C|Dew= 11.9 C|Loc pres=1005.5|Sea
```

```
X=90  Y=2   Lat=78^22.00' N Long=100^40.00' W S. X=8251 km    E. Y=15189 km    Dis=? Az=? PSE UR LAT/LONG
```

Callsign		Time/date	12:00 UTC the 7th
Latitude	54^32 N	Longitude	011^04 E
Source:			
Direction (deg.)	90		
Speed (km/h)	14.4		
Temperature (C)	14.0		
Dewpoint (C)	11.9		
Sea pres. (mBar)	1006.5	Loc pres (mBar)	1005.5
Pressure evolution 3h (mBar)	-0.6		
Cloud cover	8/8ths (overcast)	Visibility	5000 m
Cloud base	300 to 600 m	Sky state	8/8
Rain (mm)	0.1 (in 6h)	Ship	
Sea temp. (C)		Waves	
Min temp. (C)	82.5	Max temp. (C)	

Other information

```
Present WX:rain
...Past 1 WX:rain
...Past 2 WX:cloud covering more thna
half of sky
```

GMDSS DSC

GMDSS DSC (Global Maritime Distress and Safety System Digital Selective Calling) is a system used on HF for ships to send out a distress or safety alert. Listen to it in USB mode. GMDSS DSC can be found at the following frequencies in kHz.

- 2187.5 (main)
- 4207.5
- 6312.0
- 8414.5 (main)
- 12577.0
- 16804.5

An example GMDSS DSC message might look like:

```
<Selective call to a particular individual station>
Called MMSI station address: 371371000 [Ship] (Panama)
Category: Safety
MMSI self-identifier: 005030001 [Coast station: Charleville/Wiluna RCC Australia] (Australia)
Telecommand 1: Test
Telecommand 2: No information
No control by check sum.
Date and time of decoding: 09/04/2014 10:29:32
```

NAVTEX

Navtex (Navigational Telex) is a system used by mariners on HF for sending text messages that are directly printed onto paper using a Navtex machine. The messages usually contain information about maritime weather. Navtex uses USB mode and it is typically transmitted at 518 kHz and 490 kHz. An example Navtex message might look like:

```
...PLEASE REFER TO COASTAL WATERS FORECASTS (CWF) AVAILABLE
THRU NOAA WX RADIO AND OTHER MEANS FOR DETAILED
COASTAL WATERS FORECASTS...
.SYNOPSIS FOR MID ATLC WATERS...A LOW PRES AREA WILL DEVELOP
AND
PASS JUST N OF THE WATERS TONIGHT AND THEN MOVE NE LATER MON
AND
MON NIGHT WHILE STRENGTHENING. STRONG HIGH PRES WILL BUILD E
ACROSS THE NRN WATERS MON NIGHT THRU TUE NIGHT...AND PASS NE
OF THE AREA WED. A COASTAL FRONT WILL FORM OFF THE CAROLINA
COAST TUE NIGHT. LOW PRES WILL DEVELOP AND MOVE NE ALONG THE
CAROLINA COAST LATE WED.
```

SITOR

SITOR (Simplex Teletype Over Radio) is a system used by mariners for sending text messages over HF radio. There is SITOR-A and SITOR-B. SITOR-A is used for point to point communications and SITOR-B is used for broadcasting. The difference between the two is that SITOR-A will request a repeat of the message if it contains errors, whereas SITOR-B uses forward error correction to correct errors. SITOR-B is also the protocol used by Navtex.

DIFFERENTIAL GPS (DGPS)

GPS accuracy used to be purposely degraded by the US military to prevent enemies from using it against them. To get around this limitation, differential GPS (DGPS) was used by GPS users who needed the greater accuracy, such as maritime users. It works by measuring the difference between the GPS location and a known location and then broadcasting the differences to the receivers. DGPS is still used these days to further increase GPS accuracy. The accuracy of regular GPS is about 15m, with DGPS it can be as good as 10cm. DGPS is usually transmitted at 283.5 - 325.0 kHz around waterways and can be decoded using MultiPSK.

An example of a typical DGPS reception on MultiPSK is as follows.

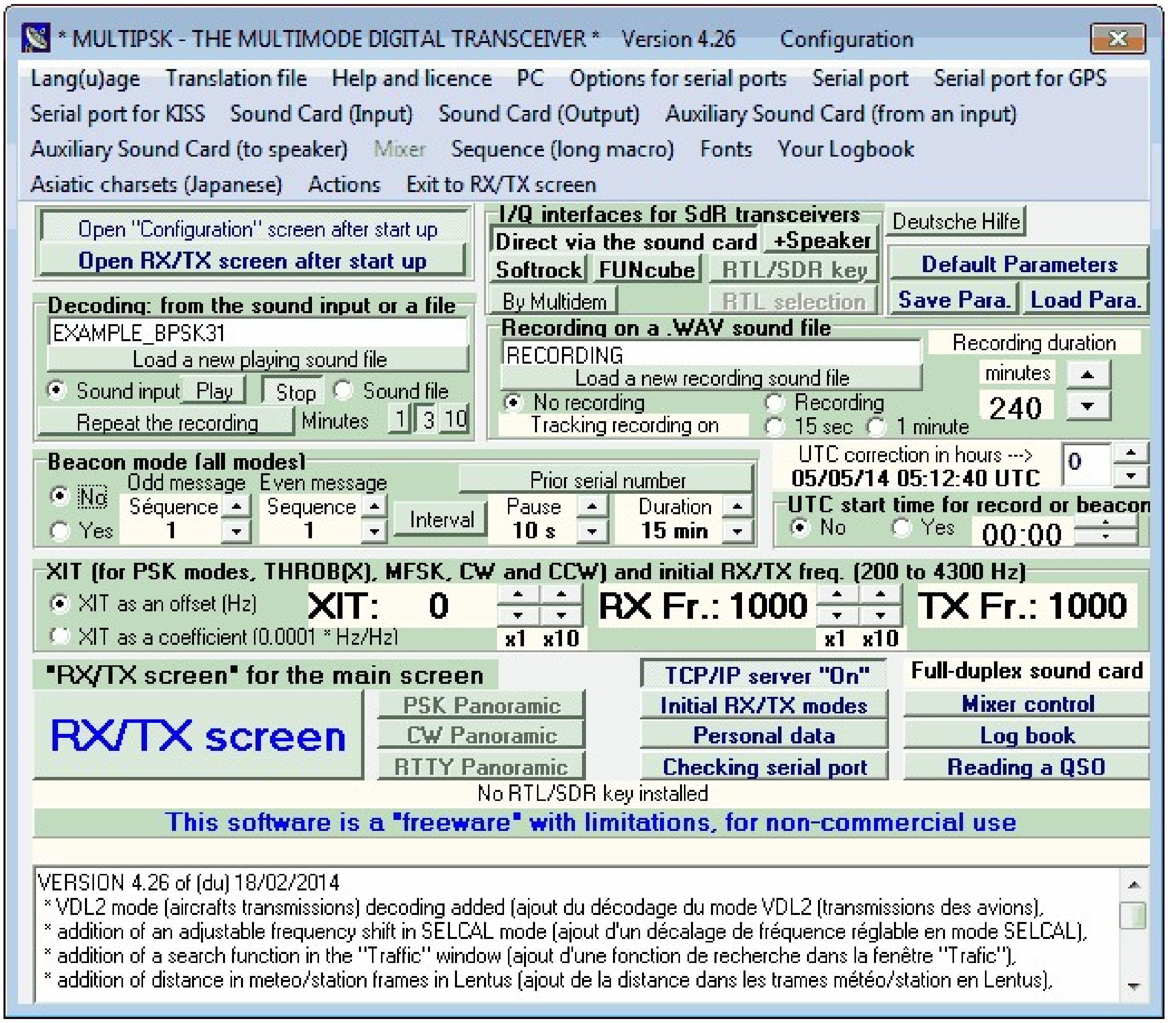
```
09/04/2014 12:55:26
Message type       : 9 (GPS partial correction set)
Station number     : 741 (Dziwnow POL 283.5 KHz TXID 481 100bps/Chejin Dan Lt KOR 292.0 KHz
TXID 670 200bps)
Z-count            : 5973 ( 59 mn 43.8 s )
Sequence count     : 5
Number of words    : 5
Health             : 6 (Transmission not monitored)
Sat. ID|SF|UDRE|Pseudorange corr.  |Range rate corr.|IOD|CRC
22     |0 |<=1m|    -14.58 m       |  -0.006 m/s    |25 |OK
31     |0 |<=1m|    -14.20 m       |  -0.006 m/s    |30 |OK
27     |0 |<=1m|    -33.34 m       |   0.020 m/s    |41 |OK
```

DECODING MARINE MODES WITH MULTIPSK

MultiPSK is a program capable of decoding Synop, GMDSS DSC, SITOR-A and Navtex. MultiPSK can be downloaded from http://f6cte.free.fr/index_anglais.htm. To decode Synop, GMDSS DSC and SITOR-A you will need to purchase the professional version of MultiPSK. However, each professional mode has a 5 minute trial limitation for testing. Navtex can be decoded indefinitely on the free version.

To decode using MultiPSK you can either use MultiPSK to connect to the RTL-SDR directly, or use MultiPSK to listen to the output of the sound card or a virtual audio cable. Here we will show the soundcard/virtual audio cable method. If you want to know how to connect directly to the RTL-SDR in MultiPSK see the VDL2 tutorial section.

1. Open your favourite SDR receiver program like SDR# or SDR-Radio.
2. Set the SDR receiver to use your favourite audio piping method and tune to the marine signal you wish to decode using USB mode.
3. Open MultiPSK. You’ll first be greeted with the following screen.


* MULTIPSK - THE MULTIMODE DIGITAL TRANSCEIVER * Version 4.26 Configuration
Lang(u)age Translation file Help and licence PC Options for serial ports Serial port Serial port for GPS
Serial port for KISS Sound Card (Input) Sound Card (Output) Auxiliary Sound Card (from an input)
Auxiliary Sound Card (to speaker) Mixer Sequence (long macro) Fonts Your Logbook
Asiatic charsets (Japanese) Actions Exit to RX/TX screen
Open "Configuration" screen after start up
Open RX/TX screen after start up
I/Q interfaces for SdR transceivers
Direct via the sound card
+Speaker
Softrock
FUNcube
RTL/SDR key
By Multidem
RTL selection
Deutsche Hilfe
Default Parameters
Save Para.
Load Para.
Decoding: from the sound input or a file
EXAMPLE_BPSK31
Load a new playing sound file
● Sound input
Play
Stop
○ Sound file
Repeat the recording
Minutes 1 3 10
Recording on a .WAV sound file
RECORDING
Load a new recording sound file
● No recording
○ Recording
Tracking recording on
○ 15 sec ○ 1 minute
Recording duration
minutes
240
Beacon mode (all modes)
● No
○ Yes
Odd message Even message
Séquence 1
Sequence 1
Interval
Prior serial number
Pause 10 s
Duration 15 min
UTC correction in hours --> 0
05/05/14 05:12:40 UTC
UTC start time for record or beacon
● No ○ Yes 00:00
XIT (for PSK modes, THROB(X), MFSK, CW and CCW) and initial RX/TX freq. (200 to 4300 Hz)
● XIT as an offset (Hz)
○ XIT as a coefficient (0.0001 * Hz/Hz)
XIT: 0
x1 x10
RX Fr.: 1000
x1 x10
TX Fr.: 1000
"RX/TX screen" for the main screen
RX/TX screen
PSK Panoramic
CW Panoramic
RTTY Panoramic
TCP/IP server "On"
Initial RX/TX modes
Personal data
Checking serial port
Full-duplex sound card
Mixer control
Log book
Reading a QSO

No RTL/SDR key installed

This software is a "freeware" with limitations, for non-commercial use

VERSION 4.26 of (du) 18/02/2014
* VDL2 mode (aircrafts transmissions) decoding added (ajout du décodage du mode VDL2 (transmissions des avions).
* addition of an adjustable frequency shift in SELCAL mode (ajout d'un décalage de fréquence réglable en mode SELCAL).
* addition of a search function in the "Traffic" window (ajout d'une fonction de recherche dans la fenêtre "Trafic").
* addition of distance in meteo/station frames in Lentus (ajout de la distance dans les trames météo/station en Lentus).

4. Go to the Sound Card (Input) menu and select your audio piping method.

5. Make sure that the “Direct via the sound card” button is pressed in and that the RTL/SDR key button is not.

6. Click on the big RX/TX screen button in the bottom left.

Now you are ready to begin decoding signals. But first for modes like Synop, we need to set up the maps.

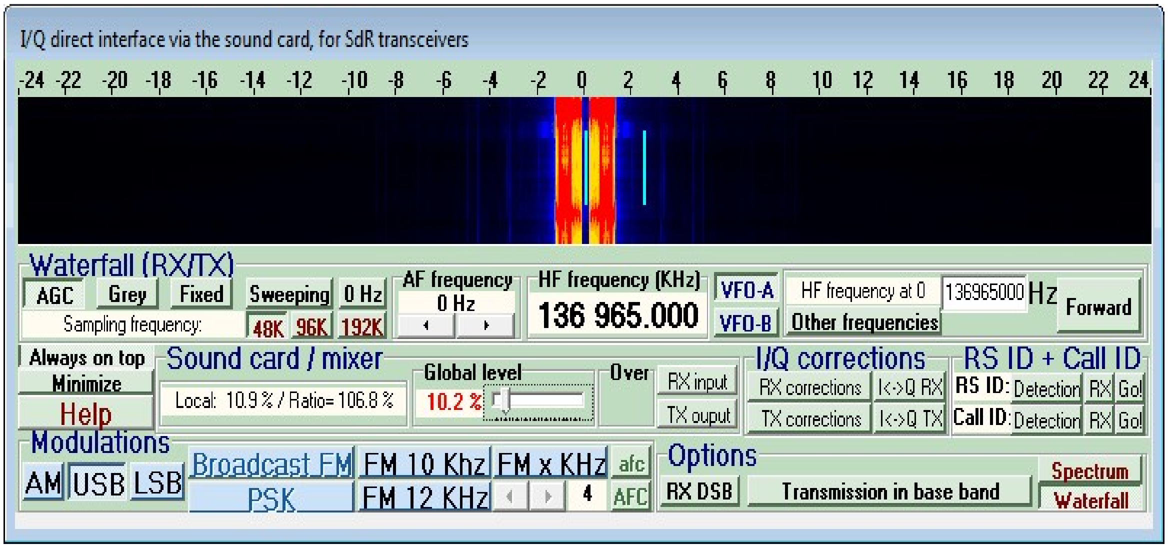
1. Download maps.zip from the multiPSK webpage (http://f6cte.free.fr/index_anglais.htm).

2. Make a folder named “maps” within the multipsk folder and then extract the contents of the maps.zip file into it.

USING MULTIPSK

1. Choose a decoder by clicking on the buttons up in the top right of the Window.

2. In the "I/Q direct Interface via the sound card, for SdR transceivers" window set the AF frequency to 0 Hz.


I/Q direct interface via the sound card, for SdR transceivers

AF frequency: 0 Hz

HF frequency (KHz): 136 965.000

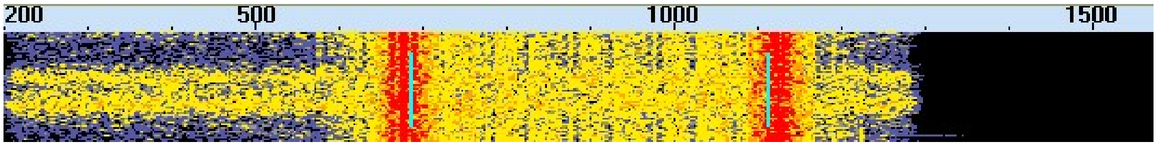
HF frequency at 0: 136965000 Hz

Sampling frequency: 48K 96K 192K

Local: 10.9 % / Ratio= 106.8 %

Global level: 10.2 %

3. In the RF spectrum or waterfall in the main MultiPSK window, click on the right peak of the signal. You can switch between RF spectrum and waterfall using the buttons on the right.



4. If you don't see the right peak, you may need to increase the frequency range of the RF spectrum/waterfall. On the right of the RF spectrum/waterfall click the 4.3 radio button in the Band kHz box.

Band KHz (P450=+)
● 2.5 ○ 3.3 ○ 4.3

DIGITAL RADIO MONDIALE (DRM) GUIDE

INTRODUCTION TO DRM

Digital Radio Mondiale (DRM) radio is a type of digital shortwave radio signal that is used by international shortwave radio broadcasters. It provides superior audio quality compared to AM signals by using digital audio encoding. With an upconverter (or direct sampling mod/experimental drivers), a good antenna and decoding software, the RTL-SDR can be used receive and decode DRM signals. This tutorial is also applicable to other software defined radios that can receive HF, such as the Softrock and Funcube dongle.

HOW TO RECEIVE AND DECODE DRM SIGNALS

To receive DRM with RTL-SDR, you will need to prepare four things.

- A HF upconverter, dongle modified with the direct sampling mod or be using the experimental HF driver.
- A HF antenna.
- The DREAM DRM decoding software with AAC decoder.
- An audio piping method (Virtual Audio Cable/VBCable). See Appendix A: Audio Pipes.

DREAM is a free opensource DRM decoder. Head to the DREAM download page which can be found at http://sourceforge.net/apps/mediawiki/drm/index.php?title=Main_Page.

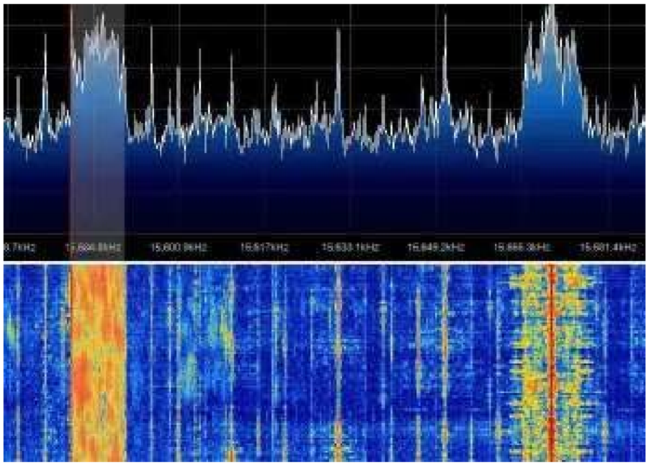
Before you start you will need to ensure your preferred audio piping method is set to use 48 kHz as the sample rate. See the Setting the Sample Rate section for more information.

1. Download DREAM and extract the zip file into a folder.
3. Due to software licence reasons the required AAC audio decoder can not be shipped with the DREAM binary file. You can follow the instructions on the DREAM download page to compile your own faad2_drm.dll decoder. But as not everyone has compiling experience, a precompiled faad2_drm.dll download for windows can be found at this megaupload link http://bit.ly/1mZivhY (Mirror: https://dl.dropboxusercontent.com/u/43061070/faad2_drm.dll). Note that using this file in some countries may not be legal due to patent laws. Place the faad2_drm.dll file into your DREAM folder.
3. Now you can open SDR# or your favourite SDR receiver software and set your desired audio pipe as the output audio device.

4. Tune to a DRM signal. DRM signals use upper side band (USB) and have a bandwidth of 10 kHz, so apply these settings to your SDR software as well. Audio AGC can be left on, but it may need to be experimented with in order to get the best decoding performance.

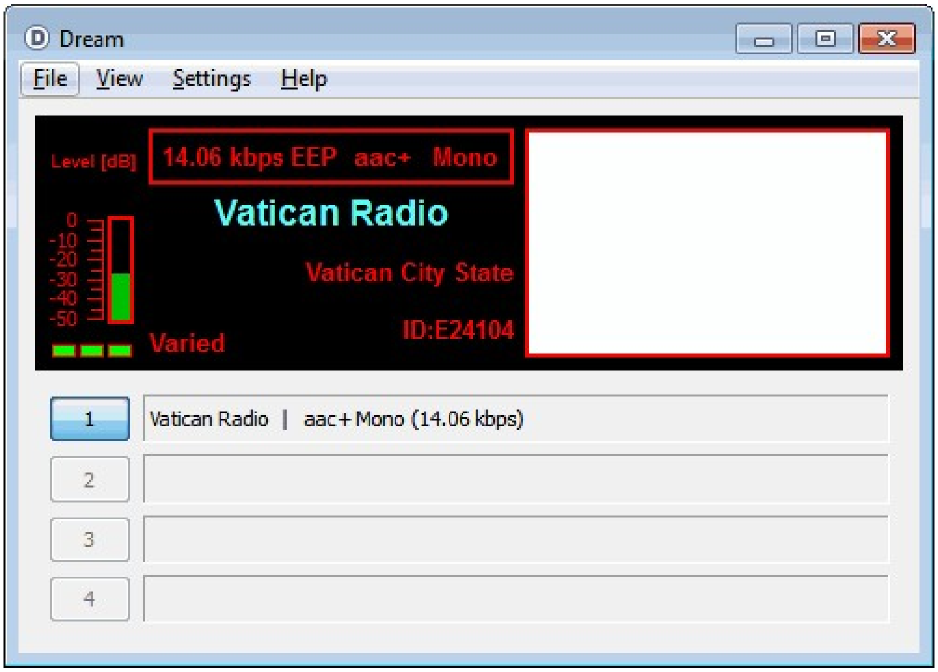
5. If using SDR# you should also experiment with the filter order. Usually a large filter order of 100+ works well especially if there are other strong signals nearby.

6. On a waterfall a DRM signal looks like this signal shown on the left of the image below. Compare it to the normal shortwave AM signal shown on the right of the waterfall.



7. Now, open DREAM and then go to **Settings->Sound Card->Signal Input->Device** and set your preferred audio pipe as the input device. Also, ensure that **Settings->Sound Card->Signal Input->Sample Rate** is set to 48000Hz.

8. Try to get the green “Level [dB]” bar in DREAM to be near the centre by adjusting the volume settings in your SDR receiver program or in Windows. If everything is set up correctly, you should see three green bars underneath the volume meter and start seeing information about the DRM radio station you are tuned to in the window and also begin to hear some audio. If you hear no audio make sure you’ve placed the faad2_drm.dll file in the correct directory.



14.06 kbps EEP aac+ Mono

Vatican Radio

Vatican City State

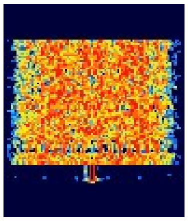
Varied

ID:E24104

1 Vatican Radio | aac+ Mono (14.06 kbps)

HFDL DECODING GUIDE

HFDL is an acronym for High Frequency Data Link. It is similar to ACARS but sent over HF frequencies. A list of HFDL frequencies can be found at http://qrg.globaltuners.com/?q=hfdl. To decode HFDL, the SDR receiver must be set to USB mode and tuned to be 1440 Hz below the centre of the HFDL packet. The bandwidth should be set as 2800 Hz. If you are unfamiliar with what an HFDL signal looks like, check out the zoomed in example waterfall image below.



To decode HFDL a program known as PC-HFDL can be used. PC-HFDL is not free software as it costs $35 USD, but it comes with a free 10 minute trial per opening. It is probably the best HFDL decoder available at the moment. PC-HFDL can be downloaded from http://www.chbrain.dircon.co.uk/. To use PC-HFL with the RTL-SDR follow these steps.

1. Open SDR# or your favourite SDR receiver software and set the audio output to your preferred audio piping method.
2. Tune to a HFDL station using Upper Side Band (USB) mode.

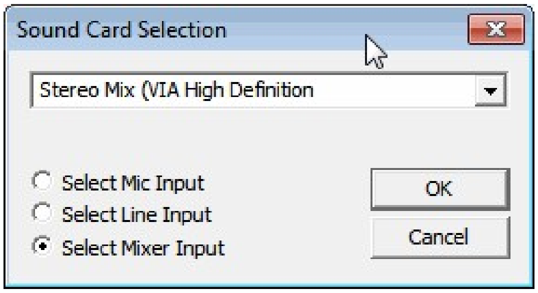
3. Ensure that the signal is tuned to 1440 Hz below the centre of the HFDL packets. This is important.

4. Set the bandwidth to 2800 Hz.

5. Open PC-HFDL.

6. Go to **System-Options -> Soundcard Configuration**.

7. Choose the audio piping method you are using in your SDR receiver and click on Select Mixer Input.

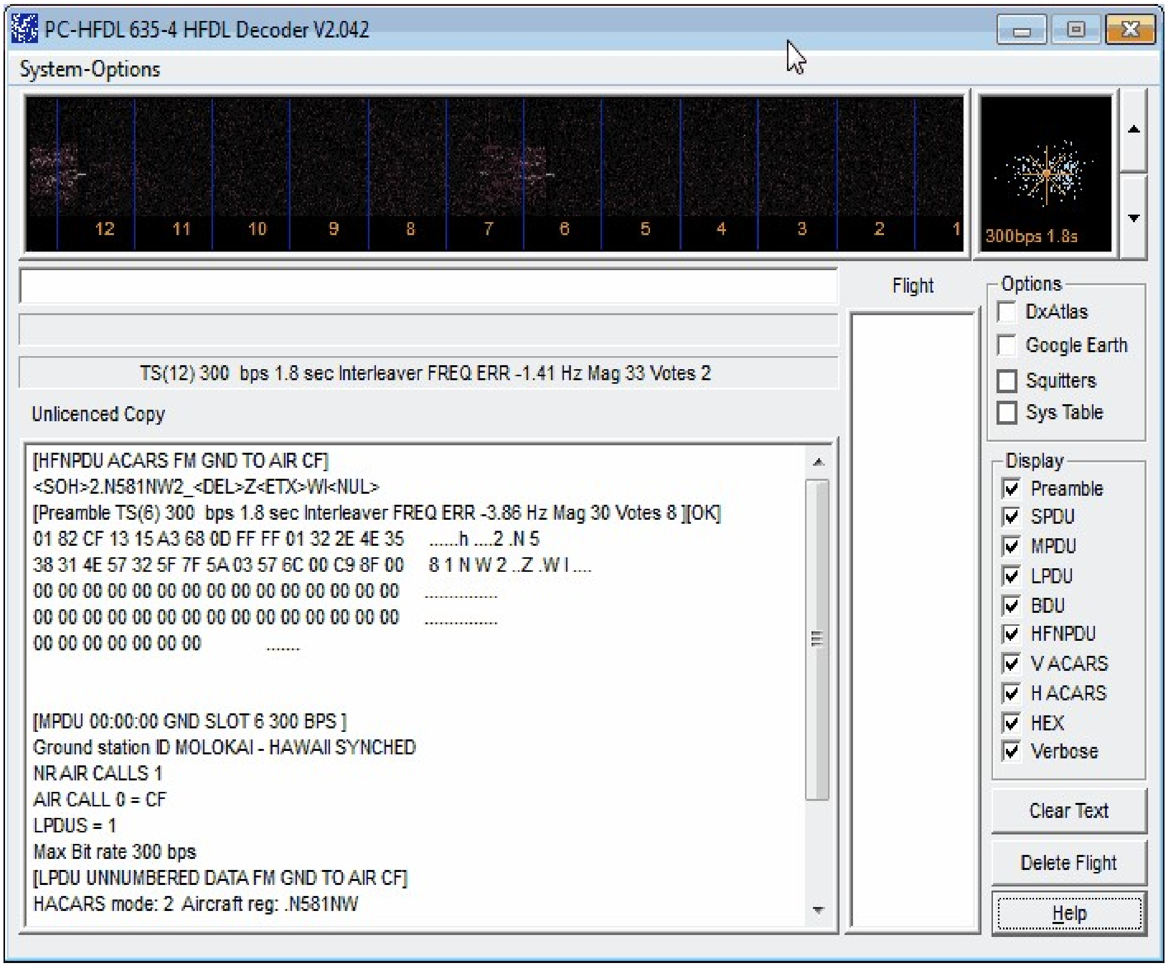


Sound Card Selection

Stereo Mix (VIA High Definition

○ Select Mic Input
○ Select Line Input
● Select Mixer Input

8. When a HFDL packet is received, PC-HFDL will decode it and display it in its main window.



TS(12) 300 bps 1.8 sec Interleaver FREQ ERR -1.41 Hz Mag 33 Votes 2

Unlicenced Copy

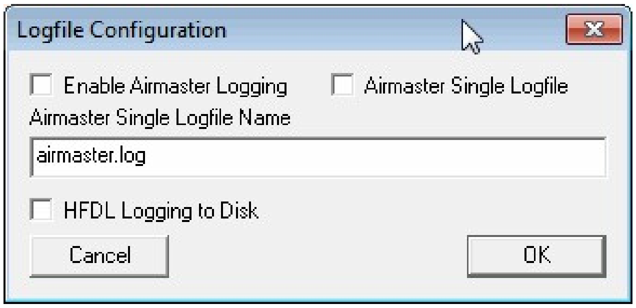
```
[HFNPDU ACARS FM GND TO AIR CF]
<SOH>2.N581NW2_<DEL>Z<ETX>WI<NUL>
[Preamble TS(6) 300  bps 1.8 sec Interleaver FREQ ERR -3.86 Hz Mag 30 Votes 8 ][OK]
01 82 CF 13 15 A3 68 0D FF FF 01 32 2E 4E 35     ......h ....2 .N 5
38 31 4E 57 32 5F 7F 5A 03 57 6C 00 C9 8F 00     8 1 N W 2 ..Z .W l ....
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00     ...............
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00     ...............
00 00 00 00 00 00 00            .......


[MPDU 00:00:00 GND SLOT 6 300 BPS ]
Ground station ID MOLOKAI - HAWAII SYNCHED
NR AIR CALLS 1
AIR CALL 0 = CF
LPDUS = 1
Max Bit rate 300 bps
[LPDU UNNUMBERED DATA FM GND TO AIR CF]
HACARS mode: 2  Aircraft reg: .N581NW
```

Options
□ DxAtlas
□ Google Earth
□ Squitters
□ Sys Table

Display
☑ Preamble
☑ SPDU
☑ MPDU
☑ LPDU
☑ BDU
☑ HFNPDU
☑ V ACARS
☑ H ACARS
☑ HEX
☑ Verbose

You can further extend the information received by PC-HFDL by using a second program called PC-HFDL-Display. This program has a large database of aircraft and will resolve aircraft registration numbers and show you the aircraft type and airline they fly for.

1. Download PC-HFDL-Display from http://www.rstools.info/downloads-2.html.
2. Extract the zip file for a folder.
3. To set up PC-HFDL-Display, first in PC-HFDL go to **System-Options->Logfile Configuration**.
4. Then in PC-HFDL enable “HFDL Logging to Disk” by checking the box.



Logfile Configuration

□ Enable Airmaster Logging □ Airmaster Single Logfile

Airmaster Single Logfile Name

airmaster.log

□ HFDL Logging to Disk

5. Now open PC-HFDL-Display by opening Display-Launcher.exe and then clicking on the PC-HFDL-Display button. Note that the first time you run Display-Launcher.exe you may need to run it as administrator.

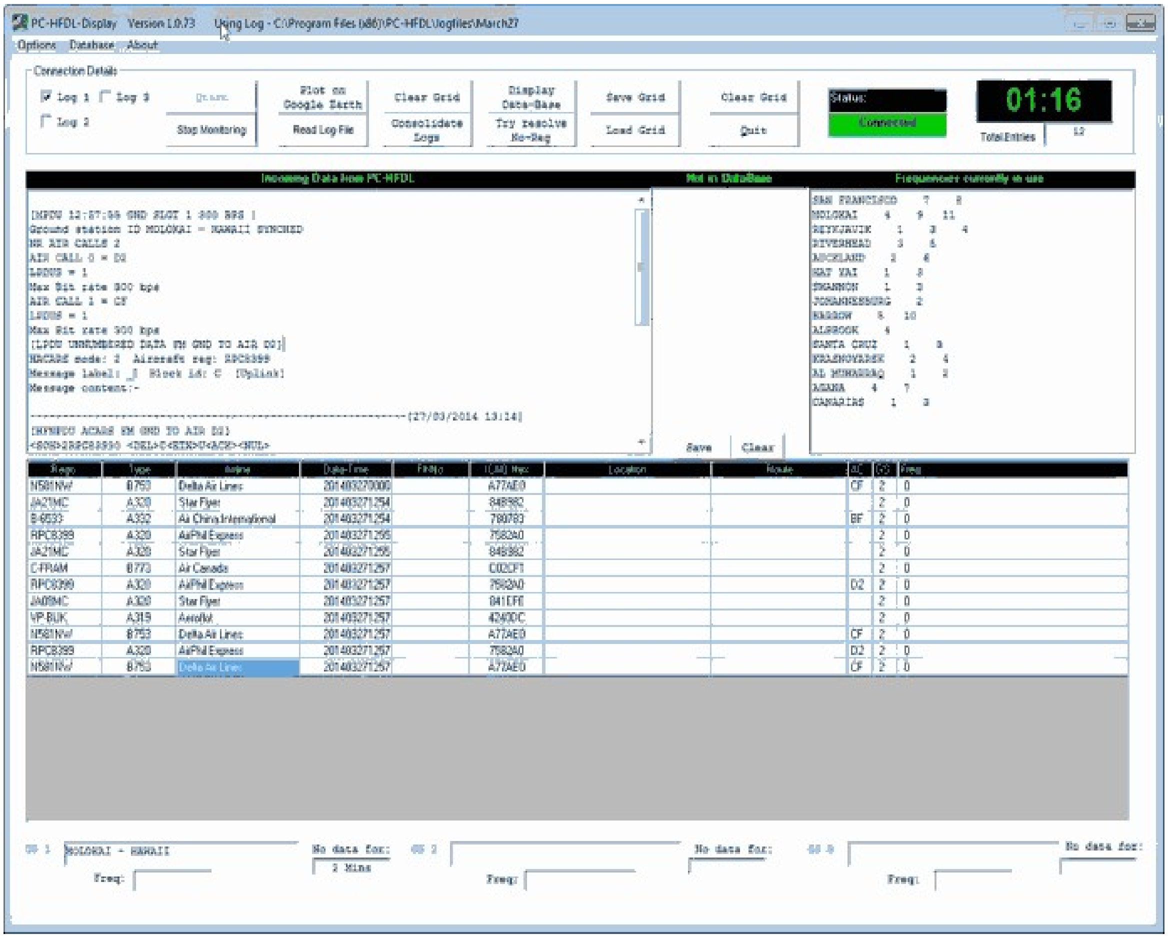
6. If a screen pops up, click ‘Use this Data’. You may also get a warning about not having a log path setup, you can ignore this for now.

7. Go to **Options->PC_HFDL Log Path 1**.

8. Navigate to the folder where PC-HFDL is installed to and enter the logfiles folder.

9. Choose the date of the current logfile that you wish to monitor. Note that PC-HFDL must have decoded at least one HFDL packet first before a logfile will be written.

10. Ensure that Log 1 is checked under connection details and then click the Start button.



```
[MPDU 12:57:56 GND SLOT 1 300 BPS ]
Ground station ID MOLOKAI - HAWAII SYNCHED
NR AIR CALLS 2
AIR CALL 0 = D2
LPDUS = 1
Max Bit rate 300 bps
AIR CALL 1 = CF
LPDUS = 1
Max Bit rate 300 bps
[LPDU UNNUMBERED DATA FM GND TO AIR D2]
HACARS mode: 2  Aircraft reg: RPC8399
Message label: _] Block id: C  [Uplink]
Message content:-

-------------------------------------------------------------(27/03/2014 13:14]
[HFNPDU ACARS FM GND TO AIR D2]
<SOH>2RPC83990 <DEL>C<ETX>U<ACK><NUL>
```

```
SAN FRANCISCO    7    8
MOLOKAI    4    9   11
REYKJAVIK    1    3    4
RIVERHEAD    3    5
AUCKLAND    2    6
HAT YAI    1    3
SHANNON    1    3
JOHANNESBURG    2
BARROW    5   10
ALBROOK    4
SANTA CRUZ    1    8
KRASNOYARSK    2    4
AL MUHARRAQ    1    2
AGANA    4    7
CANARIAS    1    3
```

Regn	Type	Airline	Date-Time	Flt No	ICAO Hex	Location	Route	AC	GS	Freq
N581NW	B753	Delta Air Lines	201403270000		A77AE0			CF	2	0
JA21MC	A320	Star Flyer	201403271254		848982				2	0
B-6533	A332	Air China International	201403271254		780783			BF	2	0
RPC8399	A320	AirPhil Express	201403271255		7582A0				2	0
JA21MC	A320	Star Flyer	201403271255		848982				2	0
C-FRAM	B773	Air Canada	201403271257		C02CF1				2	0
RPC8399	A320	AirPhil Express	201403271257		7582A0			D2	2	0
JA09MC	A320	Star Flyer	201403271257		841EFE				2	0
VP-BUK	A319	Aeroflot	201403271257		42400C				2	0
N581NW	B753	Delta Air Lines	201403271257		A77AE0			CF	2	0
RPC8399	A320	AirPhil Express	201403271257		7582A0			D2	2	0
N581NW	B753	Delta Air Lines	201403271257		A77AE0			CF	2	0

GS 1 MOLOKAI - HAWAII No data for: 2 Mins Freq:

GS 2 No data for: Freq:

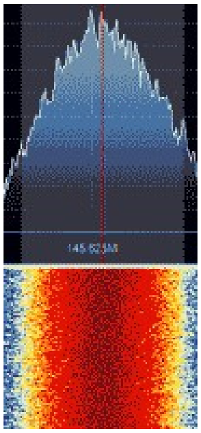
GS 3 No data for: Freq:

MULTIPSK

MultiPSK is another decoder that can decode HFDL messages. HFDL decoding requires the paid version of MultiPSK. The Display Launcher used for PC-HFDL can also be used with MultiPSK. Just use the HFDL-Display button instead of the PC-HFDL-Display one and connect to MultiPSK via TCP/IP as shown in other MultiPSK tutorials above.

D-STAR

D-STAR is a VHF digital amateur voice radio mode that is usually heard at around 145.670 MHz with a bandwidth of 6.25 kHz. D-STAR uses repeaters that go through the internet which allows for global communications. A zoomed in D-STAR waterfall image looks like the image shown below.



Digital Speech Decoder (DSD) version 1.7 (or newer if reading in the future) can be used to decode D-STAR audio. A precompiled DSD 1.7 for Windows can be downloaded at http://bit.ly/1nPU87E (Mirror 1: https://dl.dropboxusercontent.com/u/43061070/dsd-1.7.rar) (Mirror 2: http://bit.ly/RF0yLw). Download the zip file and extract the files to a folder.

To use DSD 1.7 for D-STAR use the following steps.

1. Set your preferred audio piping device to be the default Windows recording device in the Windows sound recording properties. We recommend using either VB Cable or Virtual Audio Cable for D-STAR.

2. Open your favourite SDR receiver software like SDR# or SDR-Radio and set your audio piping device and turn off any DSP options. If using SDR#, make sure that filter audio is set to off. Tune to the D-STAR frequency at 145.670 MHz and set the mode to NFM and the bandwidth to 6.25 kHz.

3. Open a Command Prompt by going to **Start->All Programs->Accessories->Command Prompt**, or by simply typing cmd into the start menu search bar.

4. In command prompt navigate to the folder that you have extracted DSD1.7 to. For example if you extracted DSD 1.7 to c:\radio\dsd17 you would type

“cd c:\radio\dsd17”.

5. Run DSD1.7 with the command `dsd -i /dev/dsp -o /dev/dsp -fd`
6. DSD will begin to decode the D-STAR audio and play the voice audio through your speakers.

6.

COMPILING THE LATEST DSD VERSION ON WINDOWS

Digital Speech Decoder (DSD) is open source software which is under active development. As a result to be able to use the latest version of the software you will need to be able to compile it yourself. On Linux compilation is fairly easy with the instructions from https://github.com/szechyjs/dsd/wiki/Installation which can be pretty much copy and pasted into a terminal window. Compilation on Windows is a little more involved so here we provide a tutorial. The easiest way to compile DSD is with the Cygwin environment.

Note that if you are not interested in compiling the latest DSD versions, a precompiled version of DSD 1.7.0 can be downloaded from http://bit.ly/1nPU87E (Mirror 1: https://dl.dropboxusercontent.com/u/43061070/dsd-1.7.rar) (Mirror 2: http://bit.ly/RF0yLw), though these will probably be older versions. Remember that for most digital audio we recommend DSD+ which can be found at http://www.rtl-sdr.com/improved-digital-voice-p25-decoding-dsd/. However, DSD 1.7 is the only software that will currently work for decoding D-STAR.

1. Download an install the Cygwin environment from https://www.cygwin.com/. Make sure to download the 32-bit version and not the 64-bit one. Cygwin is an environment that allows compilation of Linux software on a Windows platform.

2. During the installation you will be asked if you would like to install some packages. You will need to install the following packages: gcc-core, gcc-g++, libgcc1, libsndfile, libsndfile-devel, libsndfile1, cmake, make, fftw3, libfftw3-devel, libfftw3_3, liblapack-devel, liblapack0, git and wget. To install them simply type their names into the search bar and expand the menus until you find the right name. Then click on the text Skip to change it to the current version number.
   Also remember the installation location of Cygwin. By default it should be C:/cygwin.

3. Open the Cygwin terminal from the start menu. Now we will download and compile the required pre-requisites. (Note that if you have multiple compilers on your system when running the cmake command you should set

cmake to use the “Unix Makefiles” generator with cmake -G “Unix Makefiles” ..

4. Download mbelib (the P25 decoder library) with the following:

```
git clone https://github.com/szechyjs/mbelib.git
cd mbelib
mkdir build
cd build
cmake ..
make
make install
```

5. Next install ITPP.

```
cd ~
wget -O itpp-latest.tar.bz2 http://sourceforge.net/projects/itpp/files/latest/download?source=files
tar xjf itpp-latest.tar.bz2
cd itpp-4.3.1 (or whatever the latest version is)
mkdir build
cd build
cmake ..
make
make install
```

6. Finally compile DSD.

```
cd ~
git clone https://github.com/szechyjs/dsd.git
cd dsd
mkdir build
cd build
cmake ..
make
```

7. Now in Windows browse to the cygwin installation directory (default directory is C:\cygwin) and then browse to C:\cygwin\home\NAME\dsd\build, where NAME will be your user name on your Windows PC.

8. In the build folder you should be able to find dsd.exe. Copy dsd.exe into another folder on your computer. Now, if you were to go into command prompt and try to run dsd.exe you would receive an error. This is because we need to move some .dll files into the same folder as dsd.exe first.

9. Browse to C:\cygwin\bin and copy cygwin1.dll, cygsndfile-1.dll, cyggcc_s-1.dll, cygstdc++-6.dll, cygFLAC-8.dll, cygogg-0.dll, cygvorbis-0.dll, cygvorbisenc-2.dll, cygfftw3-3.dll, cyggomp-1.dll, cyggfortran-3.dll and finally cygquadmath-0.dll into the same folder as dsd.exe.
10. From C:\cygwin\home\NAME\itpp-4.3.1\build\itpp copy cygitpp-8.dll into the same folder as dsd.exe.
11. From C:\cygwin\home\NAME\mbelib\build copy cygmbe-1.dll into the same folder as dsd.exe.
12. From C:\cygwin\lib\lapack copy cygblas-0.dll into the same folder as dsd.exe.

Now DSD should be able to run from the Windows command line. Remember that with DSD 1.7+ you need to specify the input and output audio devices. For example if you are using the default input and output audio devices you would run `dsd -i /dev/dsp -o /dev/dsp`.

RTL-SDR FOR ANDROID

At the time of writing this section in this book there are currently five Android RTL-SDR apps available on the Google Play marketplace.

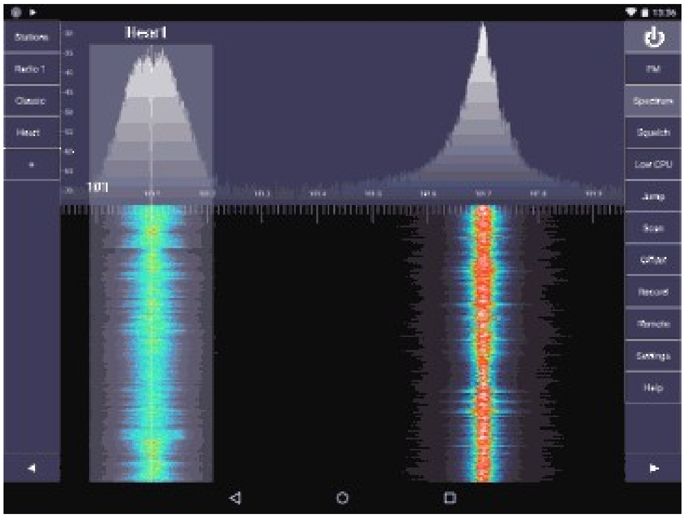
To use RTL-SDR on Android, you need a fairly powerful device that runs at least Android 4.0. You also need a special USB OTG (on-the-go) cable adapter. See the Buy RTL-SDR dongles page for a link to places where you can buy OTG cables http://www.rtl-sdr.com/buy-rtl-sdr-dvb-t-dongles/. Note that some mobile devices do not have enough power to run the RTL-SDR dongle by themselves. If this is the case you will need to buy a USB OTG cable that can be powered externally via something like a battery pack. You may want to consider a powered USB OTG cable regardless because running the RTL-SDR on a mobile device can quickly drain the battery. It may pay to use an external battery pack connected to the powered USB OTG cable if using RTL-SDR on Android away from a power source.

SDR TOUCH

SDR Touch was the first RTL-SDR app for Android released. It is a general purpose SDR GUI similar in function to programs like SDR# as it has an RF spectrum and waterfall display and allows tuning of any frequency in multiple modes. The app is capable of tuning in WFM, NFM, AM, SSB (USB/LSB) and CW signals. It cannot decode digital signals like DRM or DAB/DAB+.

SDR Touch can be downloaded from the Google Play store by searching for “SDR Touch”. There is a trial version and a paid version which can be bought for around $10 USD. More information about the app can be found at http://www.sdrtouch.com/. There is also an ongoing developers discussion thread about SDR Touch going on at XDA http://forum.xda-developers.com/showthread.php?t=2108053.

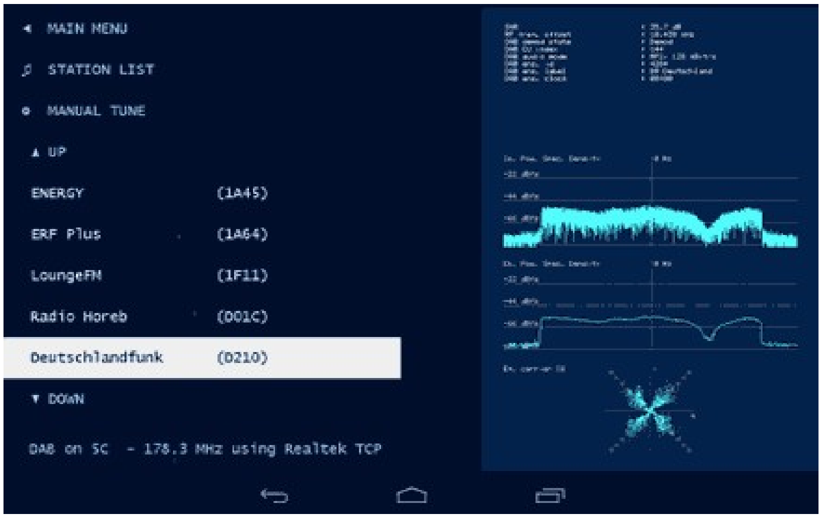
Play store link: https://play.google.com/store/apps/details?id=marto.androsdr2



WAVESINK

Wavesink is an Android program similar to SDR Touch, but it is restricted to decoding Broadcast FM, FM RDS, DRM and DAB/DAB+ audio only. It is the only Android app that can decode DRM, DAB/DAB+ with the RTL-SDR. Audio quality on either of the modes is excellent as they have spent a lot of time optimizing audio quality. The app also comes with an alternative easy to use GUI for music mode which is useful for vehicles that use an Android based entertainment system. Wavesink costs around $10 USD, but a free trial version is available for download.

Play store link: https://play.google.com/store/apps/details?id=de.ses.wavesink



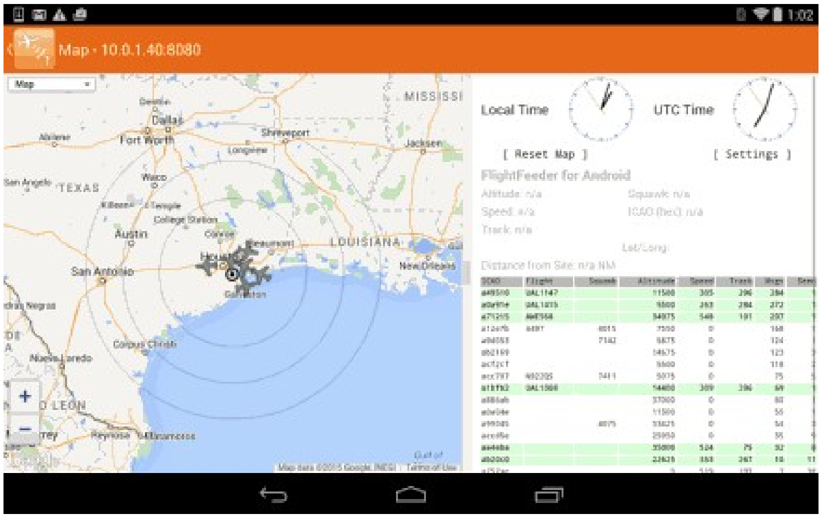
ENERGY (1A45)
ERF Plus (1A64)
LoungeFM (1F11)
Radio Horeb (D01C)
Deutschlandfunk (D210)

DAB on 5C - 178.3 MHz using Realtek TCP

FLIGHTAWARE FLIGHTFEEDER

A very good app for decoding ADS-B signals and plotting them on a map. See the ADS-B tutorial section for more information on receiving ADS-B. Also capable of feeding data to the flightaware.com shared network. As a bonus also decodes UAT signals at 978 MHz, and will automatically switch between the two frequencies.

Play store link: https://play.google.com/store/apps/details?id=com.flightaware.android.flightfeeder



FlightFeeder for Android

Altitude: n/a Squawk: n/a

Speed: n/a ICAO (hex): n/a

Track: n/a

Lat/Long:

Distance from Site: n/a NM

ICAO	Flight	Squawk	Altitude	Speed	Track	Msgs	Seen
a4b510	UAL1147		11500	305	296	284	1
a0a9fe	UAL1075		9500	263	284	272	1
a71215	AAL1568		34075	548	191	207	1
a12a7b	4491	4015	7500	0		168	1
a0d053		7142	5875	0		124	[illegible]
ab2199			14675	0		123	[illegible]
ac7jc1			5500	0		118	[illegible]
acc797	N922GS	7411	5075	0		75	[illegible]
a18f92	UAL1568		14400	309	296	69	1
a88dab			37000	0		60	[illegible]
a0e04e			11500	0		55	1
a99585		4075	33625	0		54	[illegible]
acc8be			25050	0		55	[illegible]
ae44ba			35000	524	75	52	8
ab2000			22625	353	267	10	11
[illegible]			5	519	193	1	[illegible]

USB ADSB RTL-SDR

USB ADSB RTL-SDR is a program which is capable of receiving ADS-B signals and plotting them on a map. See the ADS-B tutorial for more information on receiving ADS-B. This app costs around $0.99 USD.

Play store link: https://play.google.com/store/apps/details?id=com.wilsonae.android.usbserial

ADS-B RECEIVER

Another ADS-B receiver app available for Android. This software can also be used to display live aircraft data in another Android app called Avare, which is an app that displays offline FAA aviation charts. This app costs about $1.99 USD but has a free trial version available. More information about the app can be found at http://hiz.ch/index.php/home/adsb-receiver.

Play store link: https://play.google.com/store/apps/details?id=bs.Avare.ADSB

SDR WEATHER

This app receives weather radio and Emergency Alert System (EAS) alerts using the RTL-SDR.

Play store link: https://play.google.com/store/apps/details?id=org.thecongers.sdrweather.

RECEIVING 10 GHZ BEACONS

A 3 cm or 10 GHz band is used by amateur radio operators usually for simple beacons and communications.

To receive beacons at such high frequencies a Ku band “low noise block downconverter” (LNB) is required. Fortunately these can be had for very cheap on Amazon. See the rtl-sdr.com products page at http://www.rtl-sdr.com/buy-rtl-sdr-dvb-t-dongles/ for more information.

The LNB will receive the 10 GHz signal and then convert it down to a frequency that is receivable by the RTL-SDR. Often you will need to combine a LNB with a satellite dish to get good performance from distant 10 GHz signals.

There are several amateur radio beacons at 10 GHz that can be received by doing this. Also, there is the communications technique known as “rain scatter” which allows 10 GHz signals to be received by bouncing a signal off a precipitation cloud.

Additional Information:

http://www.rtl-sdr.com/rain-scatter-10-ghz-reception-rtl-sdr/
http://www.rtl-sdr.com/receiving-10-ghz-reflected-moon-beacon-rtl-sdr/

GNU RADIO INTRODUCTION

GNU Radio is a Linux based advanced graphical programming tool for digital signal processing (DSP). It can interface with the RTL-SDR and perform real time decoding.

This guide is intended to teach the basics of GNU Radio by helping you to create your first GNU Radio RTL-SDR program. If are not running a Linux distribution with GNU Radio preinstalled and you have not yet installed in, please see the GNU Radio installation guide section earlier in the book.

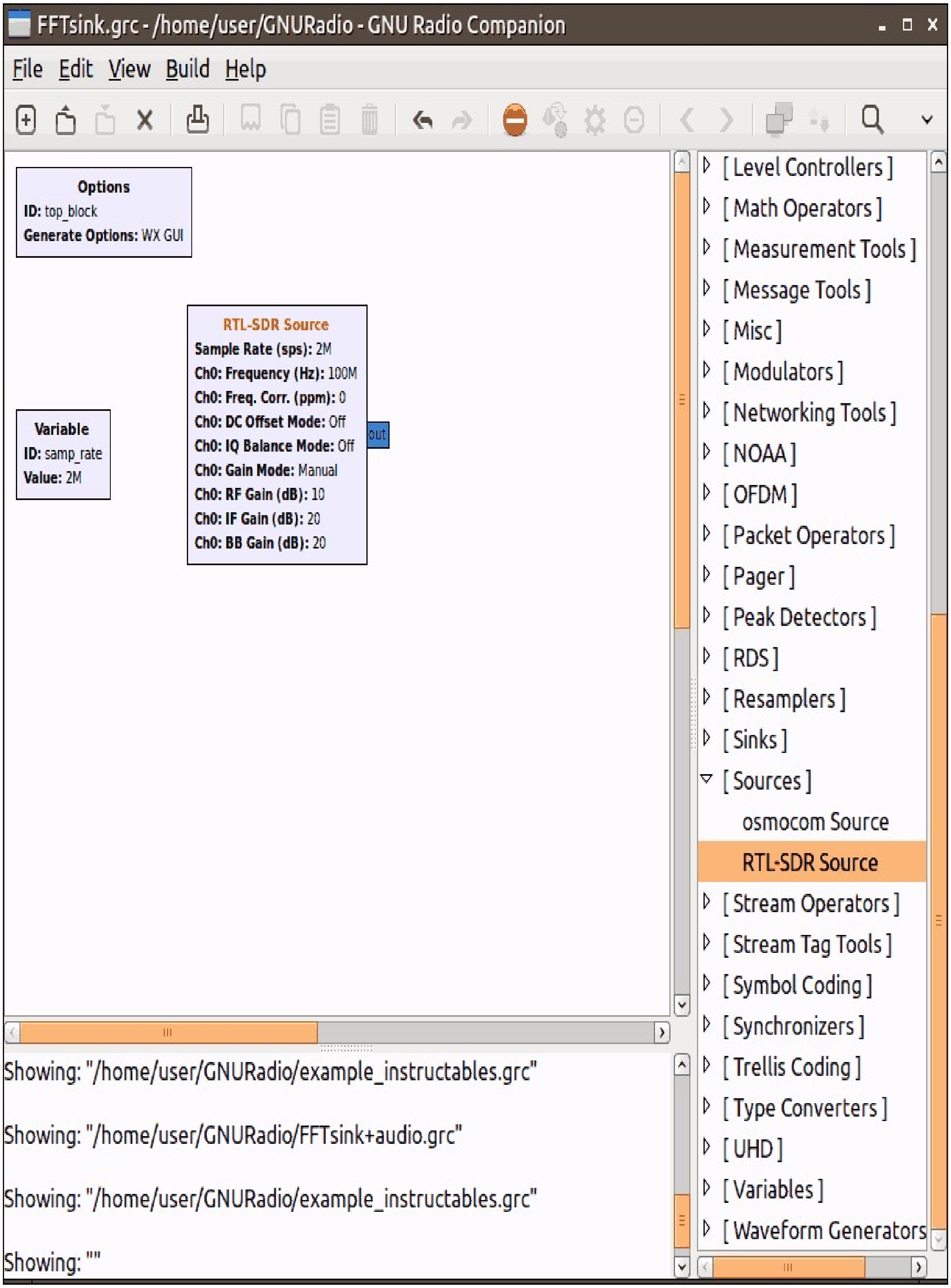
GNU RADIO PROGRAM: FFT DISPLAY

The first program we'll make is a simple FFT RF spectrum display using GNU Radio.

1. First open GNU-Radio Companion by typing the following into a terminal.

```
gnuradio-companion
```

2. On the right side of the GNURadio Companion you'll see a list of GNU Radio blocks that can be added. Locate the Sources menu and expand it. Click and drag the 'RTL-SDR Source' into the main GNU Radio window.
3. Next double click the Variable box with the ID: samp_rate. Change the samp_rate (sampling rate) to 2 MHz by typing in a value of 2e6.
   Math Tip: The number 2e6 is equivalent to 2 x 10^6 or 2M or 2000000.

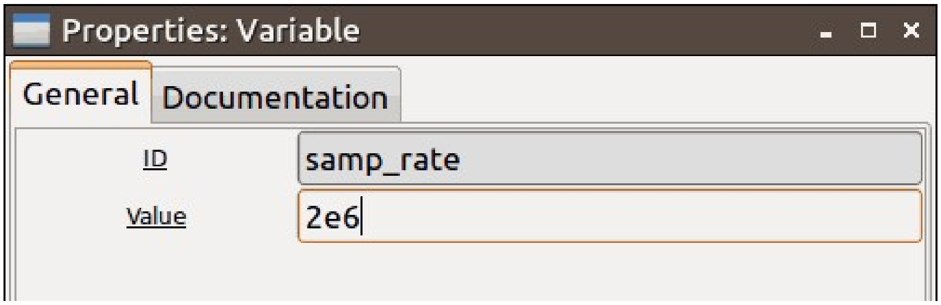


Options
ID: top_block
Generate Options: WX GUI

RTL-SDR Source
Sample Rate (sps): 2M
Ch0: Frequency (Hz): 100M
Ch0: Freq. Corr. (ppm): 0
Ch0: DC Offset Mode: Off
Ch0: IQ Balance Mode: Off
Ch0: Gain Mode: Manual
Ch0: RF Gain (dB): 10
Ch0: IF Gain (dB): 20
Ch0: BB Gain (dB): 20

Variable
ID: samp_rate
Value: 2M

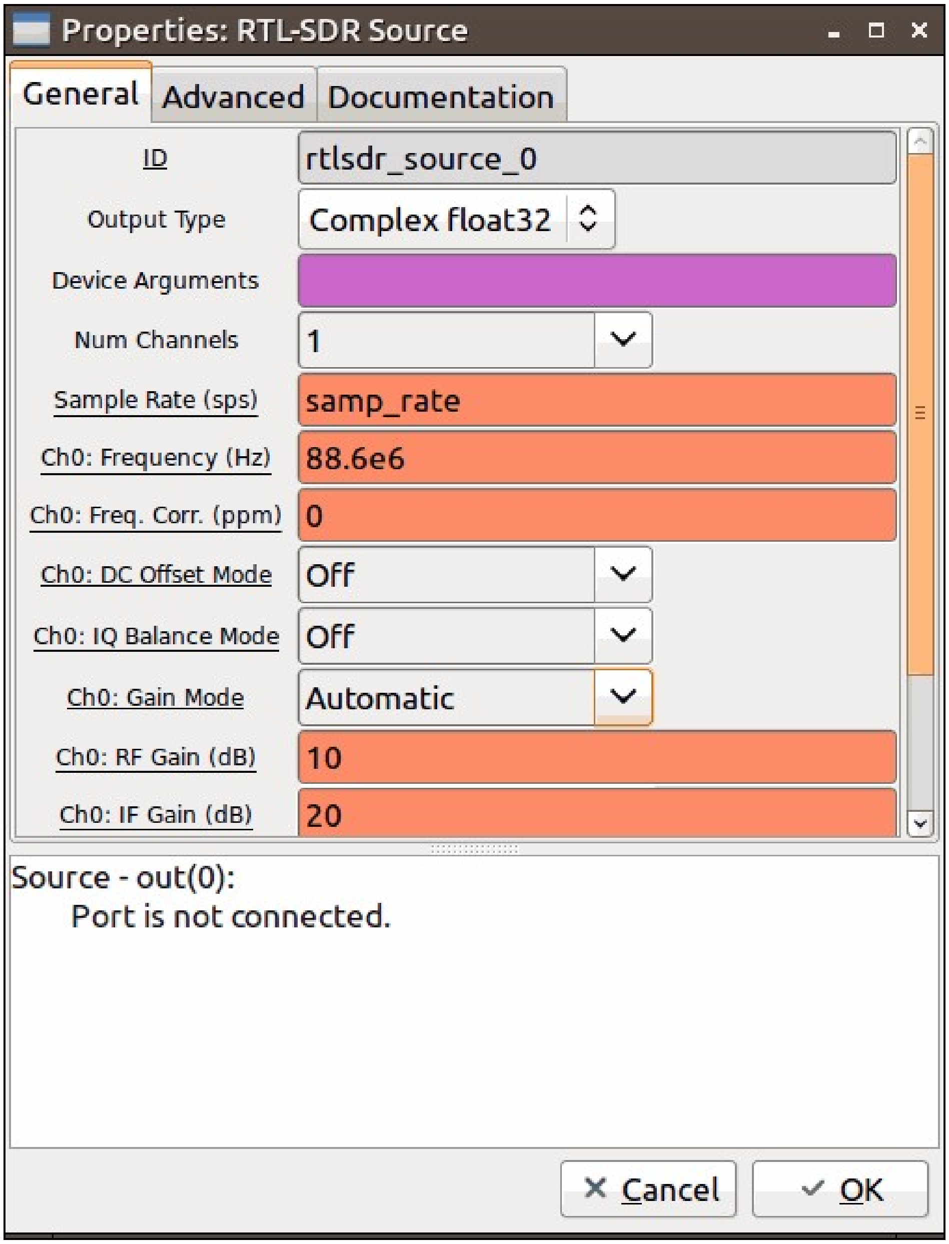
```
Showing: "/home/user/GNURadio/example_instructables.grc"

Showing: "/home/user/GNURadio/FFTsink+audio.grc"

Showing: "/home/user/GNURadio/example_instructables.grc"

Showing: ""
```



ID: samp_rate

Value: 2e6

4. Next double click on the RTL-SDR source block that you’ve just added. Change the Ch0: Frequency (Hz) value to the frequency of a local broadcast FM station. For example here it is set to 88.6 MHz by using the value 88.6e6. Also set the Gain Mode to Automatic. Finally, set the PPM offset under Ch0: Freq. Corr. (ppm) to your dongles PPM offset value if desired.

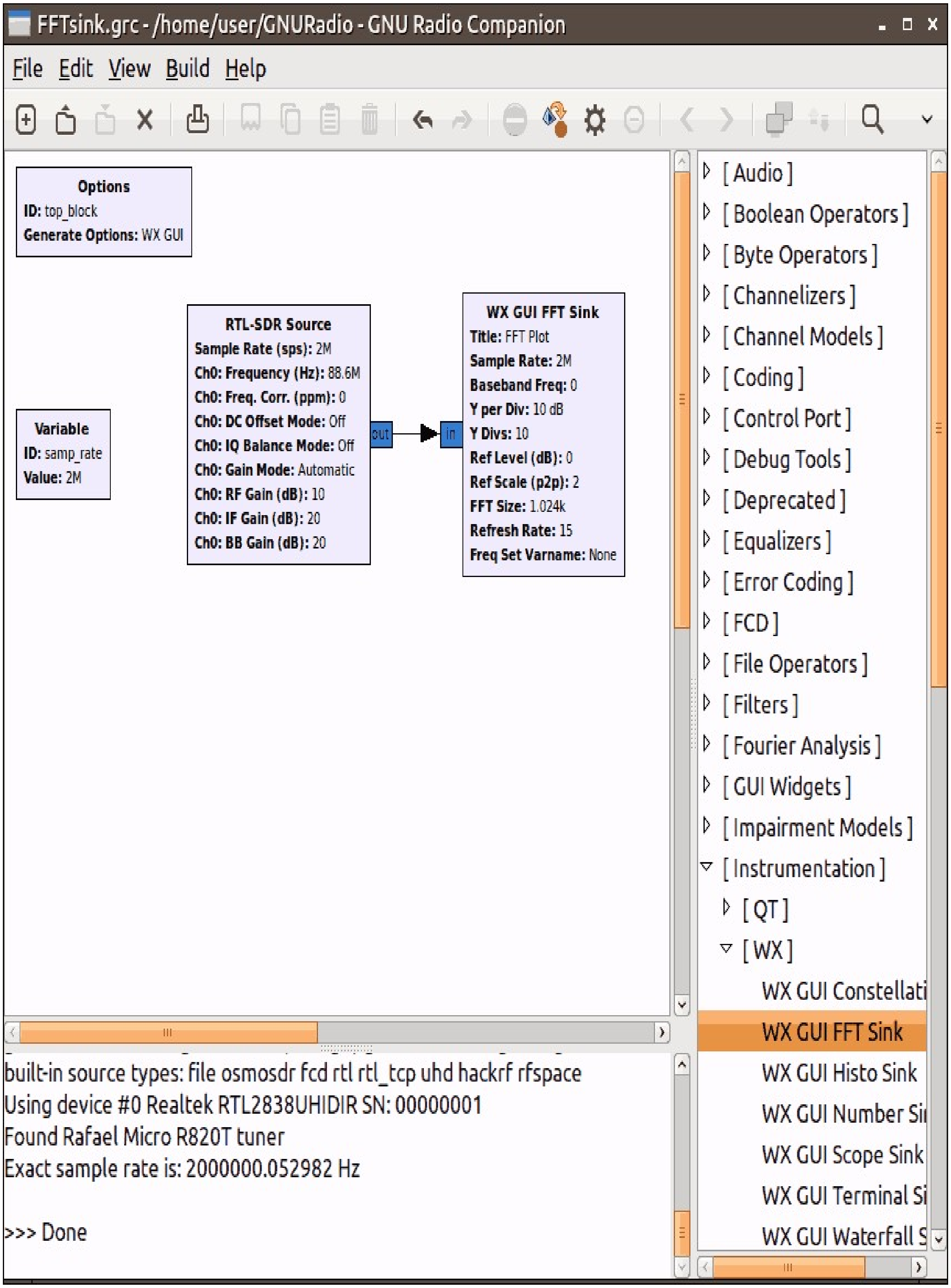


ID: rtlsdr_source_0
Output Type: Complex float32
Device Arguments:
Num Channels: 1
Sample Rate (sps): samp_rate
Ch0: Frequency (Hz): 88.6e6
Ch0: Freq. Corr. (ppm): 0
Ch0: DC Offset Mode: Off
Ch0: IQ Balance Mode: Off
Ch0: Gain Mode: Automatic
Ch0: RF Gain (dB): 10
Ch0: IF Gain (dB): 20

```
Source - out(0):
    Port is not connected.
```

5. Next in the block selection window pane go to **Instrumentation -> WX -> WX GUI FFT Sink** and drag this into the main Window. Place this block to the right of the RTL-SDR Source block.

6. Connect the two blocks together by first clicking on the blue out node of the RTL-SDR Source and then clicking on the blue in node of the WX GUI FFT Sink block.



Options
ID: top_block
Generate Options: WX GUI

Variable
ID: samp_rate
Value: 2M

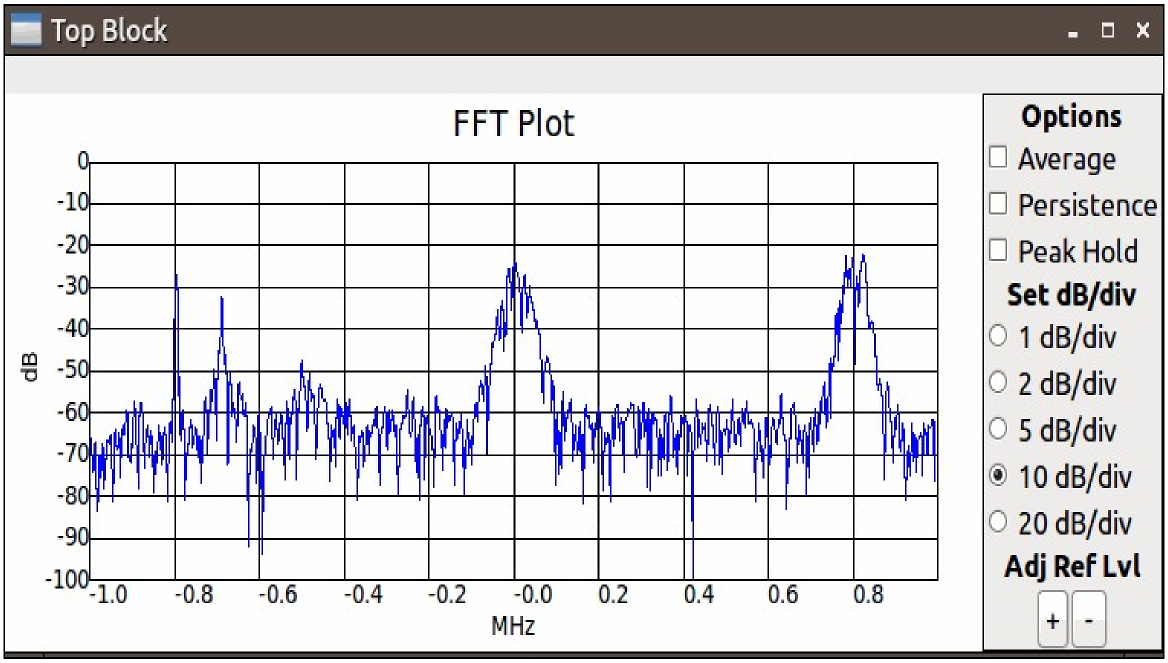
RTL-SDR Source
Sample Rate (sps): 2M
Ch0: Frequency (Hz): 88.6M
Ch0: Freq. Corr. (ppm): 0
Ch0: DC Offset Mode: Off
Ch0: IQ Balance Mode: Off
Ch0: Gain Mode: Automatic
Ch0: RF Gain (dB): 10
Ch0: IF Gain (dB): 20
Ch0: BB Gain (dB): 20

WX GUI FFT Sink
Title: FFT Plot
Sample Rate: 2M
Baseband Freq: 0
Y per Div: 10 dB
Y Divs: 10
Ref Level (dB): 0
Ref Scale (p2p): 2
FFT Size: 1.024k
Refresh Rate: 15
Freq Set Varname: None

```
built-in source types: file osmosdr fcd rtl rtl_tcp uhd hackrf rfspace
Using device #0 Realtek RTL2838UHIDIR SN: 00000001
Found Rafael Micro R820T tuner
Exact sample rate is: 2000000.052982 Hz

>>> Done
```

7. Now click on the Generate Flowgraph button. First you will be prompted to save. Call this project FFTSink.grc. Then click the execute flowgraph button ⚙. (Note please be aware that in the very latest version of GNU Radio these icons have changed.)

The result will be a live RF spectrum plot with your chosen frequency n the centre.



GNU RADIO PROGRAM: WBFM RECEIVER

Now we will extend the FFT display we just made to also play Wide Band FM (WBFM) audio. First a GNU Radio tip: To easily find blocks go to **View->Find Blocks**. A search bar will appear on the top right side which will make finding blocks much easier.

1. Open the FFTSink.grc file project you created in the previous tutorial then go to file save as and save the project as a new file called FFTsink+audio.grc.
2. Add in a ‘Rational Resampler’ Block and connect it to the RTL-SDR Source. Double click it and set the parameters as follows.
   - Interpolation = 1
   - Decimation = 4.

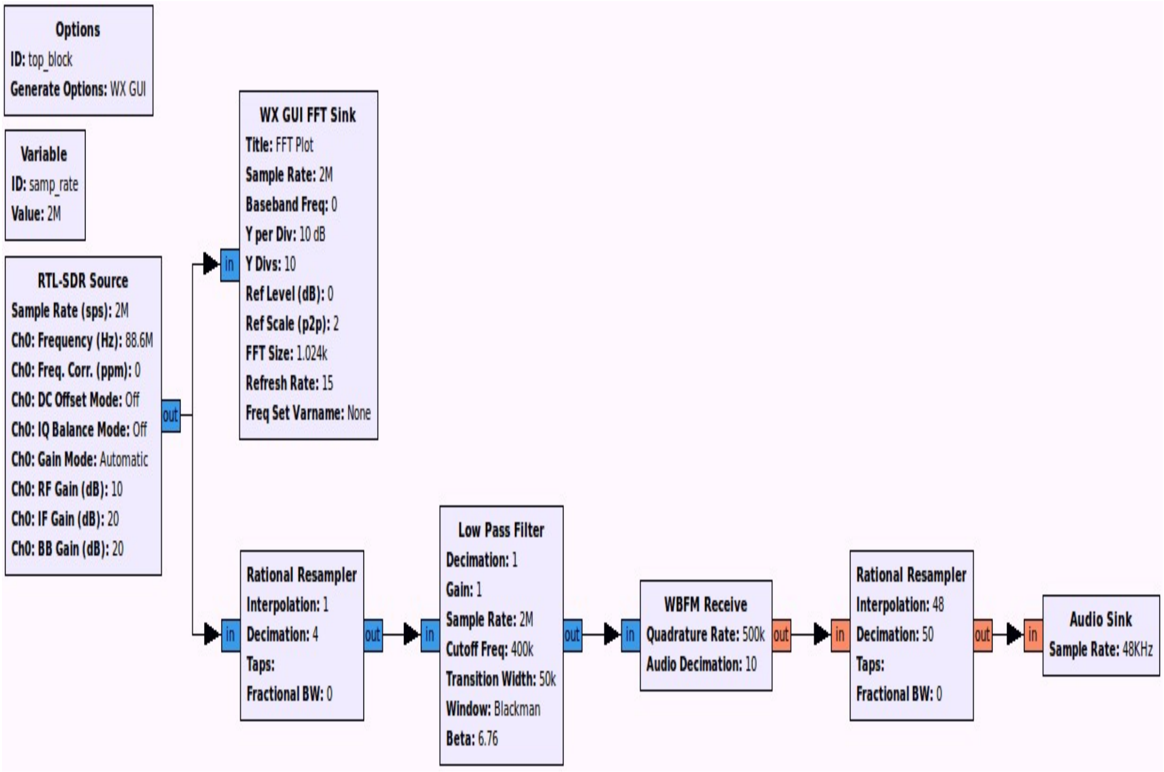
3. Add in a ‘Low Pass Filter’ block and connect it to the Rational Resampler. Set the parameters as follows.
    - Cutoff Freq = 400e3
    - Transition Width = 50e3

4. Add in a ‘WBFM Receive’ Block. Connect it to the Low Pass Filter. Set the parameters as follows.
    - Quadrature Rate = 500e6
    - Audio Decimation = 10

5. Add in a second ‘Rational Resampler’ block. Connect it to the WBFM Receiver. Set the parameters as follows.
    - Type = Float->Float (Real Taps)
    - Interpolation = 48
    - Decimation = 50

6. Add an ‘Audio Sink’ block. Connect it to the Rational Resampler. Set the parameters as follows.
    - Sample Rate = 48e3

7. Press the Execute Button. You should now hear audio playing.



Options
ID: top_block
Generate Options: WX GUI

Variable
ID: samp_rate
Value: 2M

RTL-SDR Source
Sample Rate (sps): 2M
Ch0: Frequency (Hz): 88.6M
Ch0: Freq. Corr. (ppm): 0
Ch0: DC Offset Mode: Off
Ch0: IQ Balance Mode: Off
Ch0: Gain Mode: Automatic
Ch0: RF Gain (dB): 10
Ch0: IF Gain (dB): 20
Ch0: BB Gain (dB): 20

WX GUI FFT Sink
Title: FFT Plot
Sample Rate: 2M
Baseband Freq: 0
Y per Div: 10 dB
Y Divs: 10
Ref Level (dB): 0
Ref Scale (p2p): 2
FFT Size: 1.024k
Refresh Rate: 15
Freq Set Varname: None

Rational Resampler
Interpolation: 1
Decimation: 4
Taps:
Fractional BW: 0

Low Pass Filter
Decimation: 1
Gain: 1
Sample Rate: 2M
Cutoff Freq: 400k
Transition Width: 50k
Window: Blackman
Beta: 6.76

WBFM Receive
Quadrature Rate: 500k
Audio Decimation: 10

Rational Resampler
Interpolation: 48
Decimation: 50
Taps:
Fractional BW: 0

Audio Sink
Sample Rate: 48KHz

EXPLANATION OF WBFM FLOW GRAPH

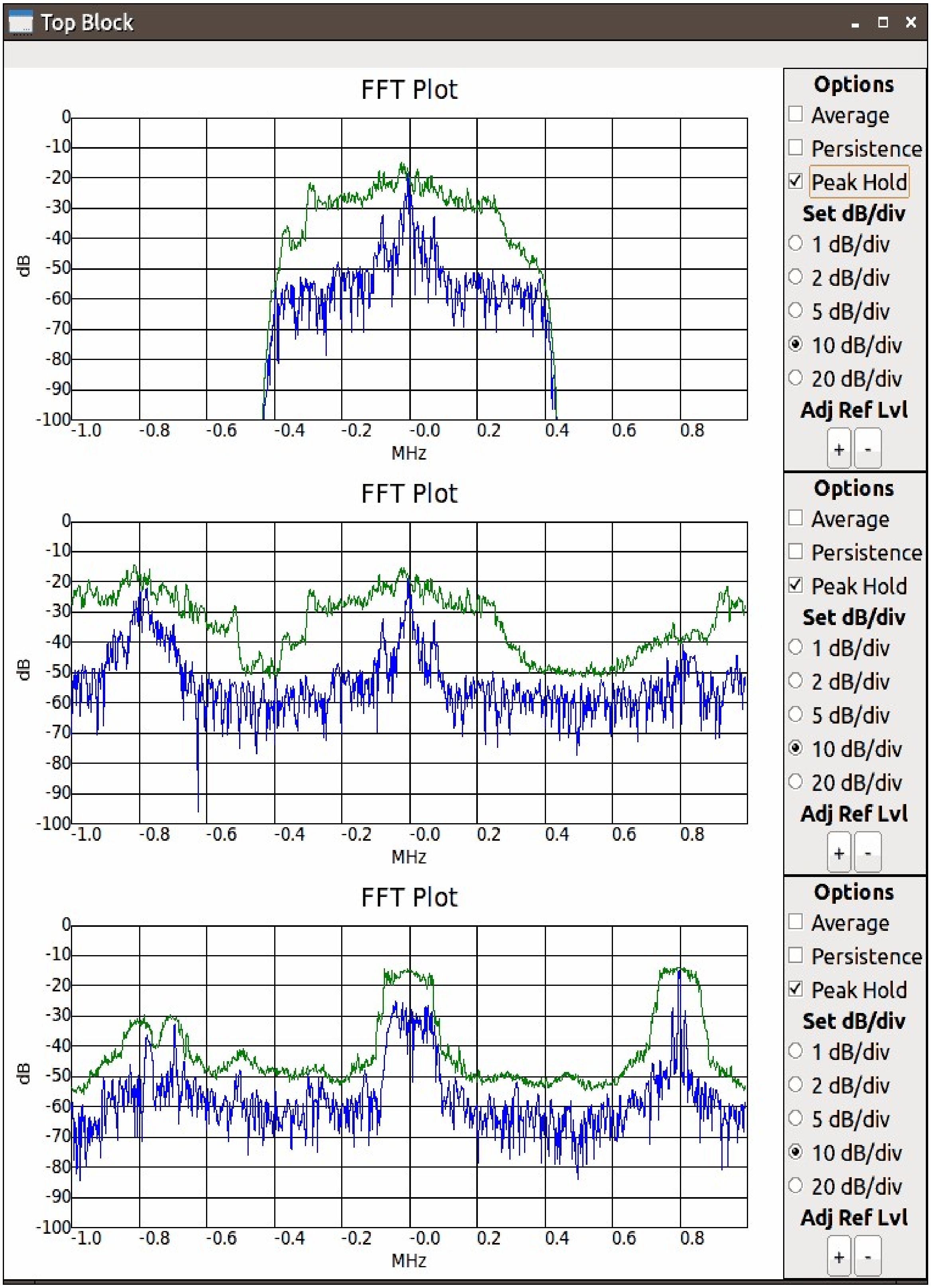
The first thing that happens in the flow graph is that the signal from the RTL-SDR is decimated by four times. Decimation is a way to get higher resolution data (higher effective number of quantization bits) out of an ADC. Essentially, by decimating by four, we reduce our 2 Msps sample rate to 0.5 Msps, but gain about one extra bit on our ADC. We need to be careful of the amount we decimate by, as decimating by a wrong value could cause neighbouring signals to overlap onto the signal of interest. For a tutorial on decimation see http://www.atmel.com/images/doc8003.pdf.

After decimation we use a low pass filter to select only the WBFM signal that we are interested in, which is at the centre frequency. This is needed as nearby WBFM frequencies are received in the same bandwidth, but we want to listen to only one. The bandwidth of a typical WBFM radio station is about 200 kHz. After decimating by four, the bandwidth of our WBFM signal is multiplied by four to 800 kHz. The lowpass filter cutoff frequency is measured for only one side of the bandwidth, so here we set it to 400 kHz. The lowpass filter transition width defines how ‘sharp’ our filter is. The smaller the transition width, the more square the low pass filter becomes. However, the smaller the transition width, the greater the number of calculations required and thus the greater CPU time required. With a transition width of 100e3 GNU Radio uses approximately 33% CPU, a transition width of 10e3 uses 45% CPU and a transition width of 1e6 tries to use over 100% of the CPU.

Next comes the WBFM Receiver block which handles the WBFM to audio decoding. We set the Quadrature Rate at 500e6 because 0.5 Msps is our sample rate after decimation. We also set the audio decimation as 10, as this will bring the 0.5 Msps decimated signal down to a 50 kHz output audio signal. This can then easily be converted to the 48 kHz sample rate that our soundcard needs. Decimation could also be done in the next rational resampler block, but the resampling performed in the WBFM block is of higher quality.

As the soundcard needs 48 kHz audio we need to convert our 50 kHz audio signal down to 48 kHz. We do this by first taking the signal down to 1000 Hz by decimating by 50. Then by interpolating by 48 we get a final sample rate of 48 x 1000 = 48 kHz. This is exactly the amount our sound card needs.

To learn more we recommend connecting FFT sinks to the output of the first rational resampler and to the output of the low pass filter to see how they affect the signal as shown in the graphs below. The top graph is the low pass filtered signal, the middle graph is the decimated signal and the bottom graph is the raw signal.



Options

□ Average
□ Persistence
☑ Peak Hold

Set dB/div

○ 1 dB/div
○ 2 dB/div
○ 5 dB/div
● 10 dB/div
○ 20 dB/div

Adj Ref Lvl

+ -

Options

□ Average
□ Persistence
☑ Peak Hold

Set dB/div

○ 1 dB/div
○ 2 dB/div
○ 5 dB/div
● 10 dB/div
○ 20 dB/div

Adj Ref Lvl

+ -

Options

□ Average
□ Persistence
☑ Peak Hold

Set dB/div

○ 1 dB/div
○ 2 dB/div
○ 5 dB/div
● 10 dB/div
○ 20 dB/div

Adj Ref Lvl

+ -

GNU RADIO PROGRAM: COMPILING A WBFM-RDS RECEIVER

For this tutorial we will show you how to download, build and run a ready made GNU Radio program. We will download the GR-RDS program, which allows you to not only listen to WFM broadcast radio, but to also decode the RDS signals inside of them. To get started follow the instructions below.

Clone the gr-rds program from Github using the following command.

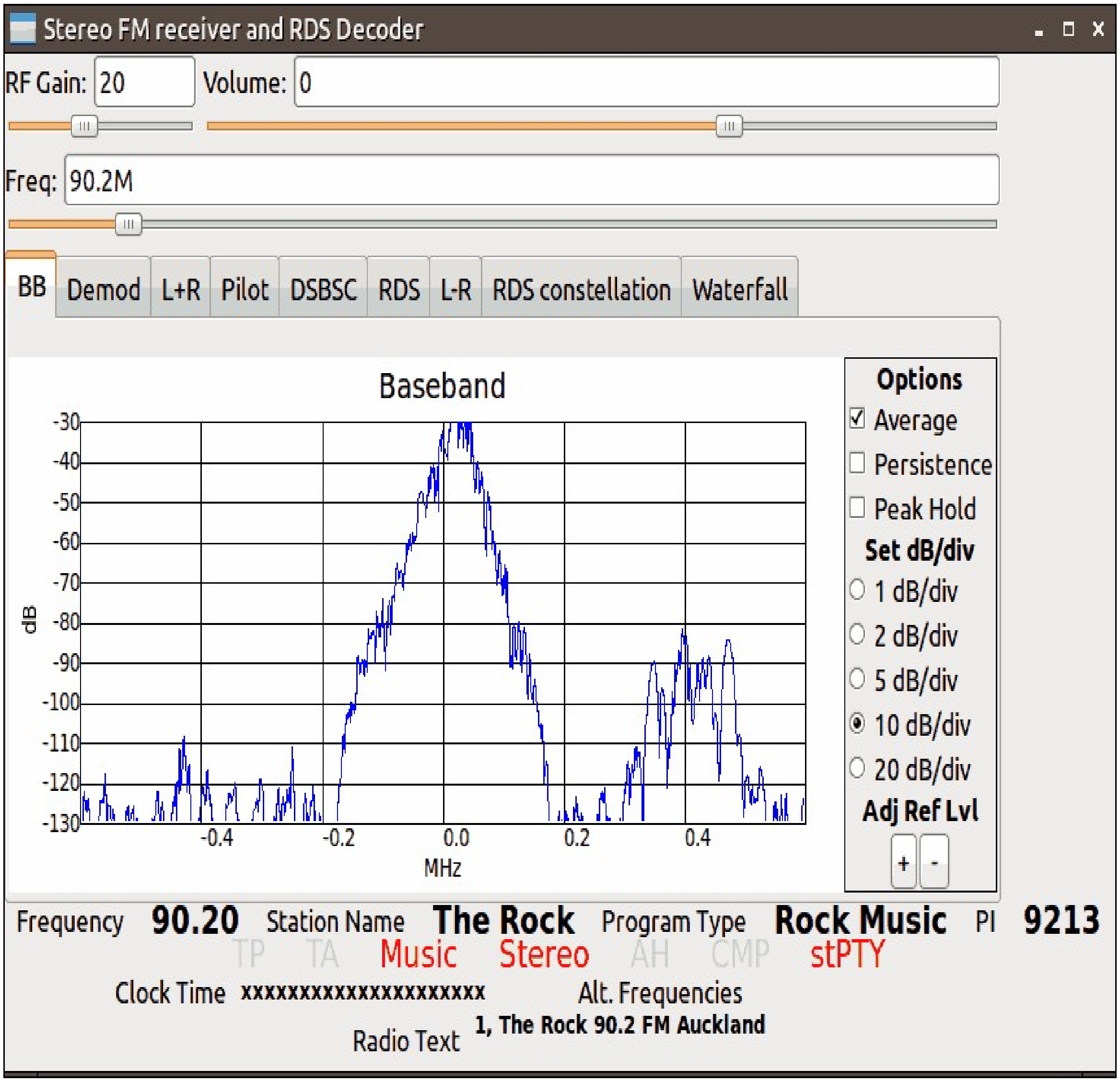
```
git clone git://git.github.com/bastibl/gr-rds
```

Now build gr-rds with the following.

```
cd gr-rds
mkdir build
cd build
cmake ..
make
sudo make install
sudo ldconfig
```

Now you can run gr-rds by opening `gnuradio-companion rds_rx.grc` or running the python file as shown below.

```
cd ..
cd apps
python rds_rx.py
```



Stereo FM receiver and RDS Decoder

RF Gain: 20 Volume: 0

Freq: 90.2M

☑ Average
□ Persistence
□ Peak Hold

Set dB/div

○ 1 dB/div
○ 2 dB/div
○ 5 dB/div
● 10 dB/div
○ 20 dB/div

Adj Ref Lvl

Frequency 90.20 Station Name The Rock Program Type Rock Music PI 9213

TP TA Music Stereo AH CMP stPTY

Clock Time xxxxxxxxxxxxxxxxxxxxx Alt. Frequencies

Radio Text 1, The Rock 90.2 FM Auckland

REFERENCES AND ADDITIONAL LINKS:

More GNU Radio Tutorials: http://files.ettus.com/tutorials/

GNU Radio Video Tutorial Series: https://www.youtube.com/playlist?list=PL618122BD66C8B3C4

A comprehensive video course on GNU Radio by Micheal Ossmann, inventor of the HackRF SDR can be found at: https://greatscottgadgets.com/sdr/.

COMPILING RTL-SDR DRIVERS FROM SOURCE ON WINDOWS

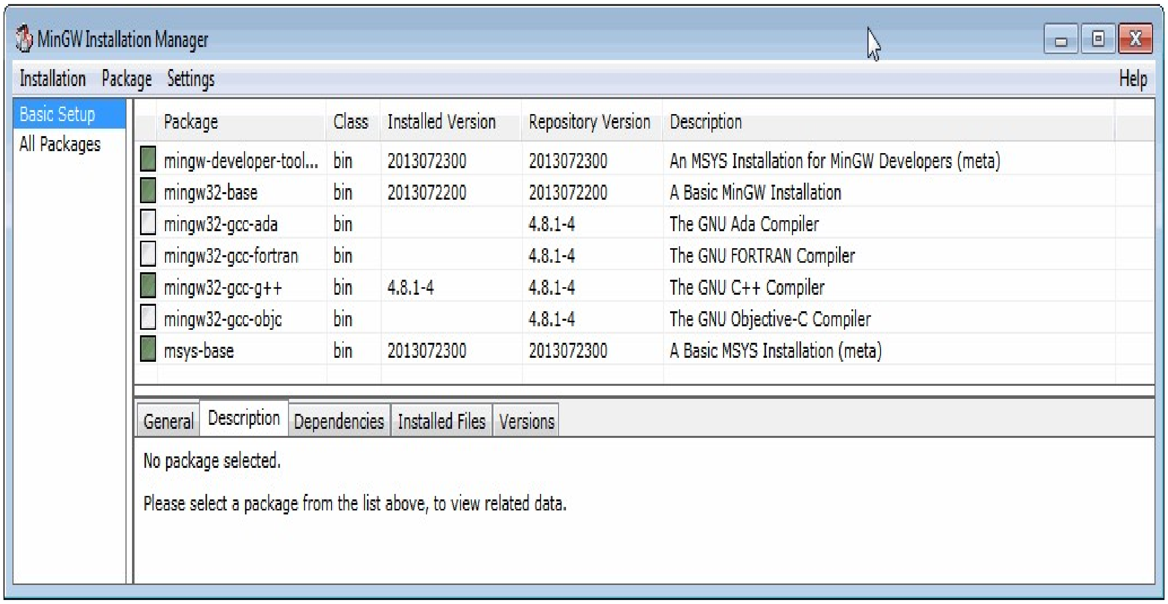
Compiling the RTL-SDR drivers is a useful skill to have as it allows you to test the latest updates to the driver or to try out various experimental RTL-SDR branches. One such regularly updated experimental branch is keenerds, which can be found here https://github.com/keenerd/rtl-sdr.

Note that if you simply want to use the RTL-SDR, compiling the drivers is not necessary. Just follow the Quickstart guide from the first section. Compiling the drivers on Linux is simple and instructions are provided on the official osmocom page at http://sdr.osmocom.org/trac/wiki/rtl-sdr. Compilation on Windows is a little more complicated so below we present a tutorial.

COMPILING WITH MINGW

This tutorial shows how to compile from source using MingW, a tool for compiling Linux programs on Windows. For other users to use the compiled files they will need to also have MingW set up on their system. MingW can be downloaded from http://www.mingw.org/. Make sure you set the MingW path in Windows correctly if it does not set it automatically (http://stackoverflow.com/questions/5733220/how-to-add-mingw-bin-directory-to-my-system-path).

1. Install MingW making sure to install the developer tools, base and g++ compilers.



	Package	Class	Installed Version	Repository Version	Description
🟩	mingw-developer-tool...	bin	2013072300	2013072300	An MSYS Installation for MinGW Developers (meta)
🟩	mingw32-base	bin	2013072200	2013072200	A Basic MinGW Installation
□	mingw32-gcc-ada	bin		4.8.1-4	The GNU Ada Compiler
□	mingw32-gcc-fortran	bin		4.8.1-4	The GNU FORTRAN Compiler
🟩	mingw32-gcc-g++	bin	4.8.1-4	4.8.1-4	The GNU C++ Compiler
□	mingw32-gcc-objc	bin		4.8.1-4	The GNU Objective-C Compiler
🟩	msys-base	bin	2013072300	2013072300	A Basic MSYS Installation (meta)

No package selected.

Please select a package from the list above, to view related data.

2. Download and install CMake and be sure to check add to path during the installation process. http://www.cmake.org/.

3. Download libusb1.0 http://sourceforge.net/projects/libusbx/files/releases/.

4. Download pthreads ftp://sourceware.org/pub/pthreads-win32/pthreads-w32-2-9-1-release.zip.

5. Extract the libusb1.0 zip file to a folder like c:\libraries\libusb-include and pthreads to c:\libraries\pthreads-include.

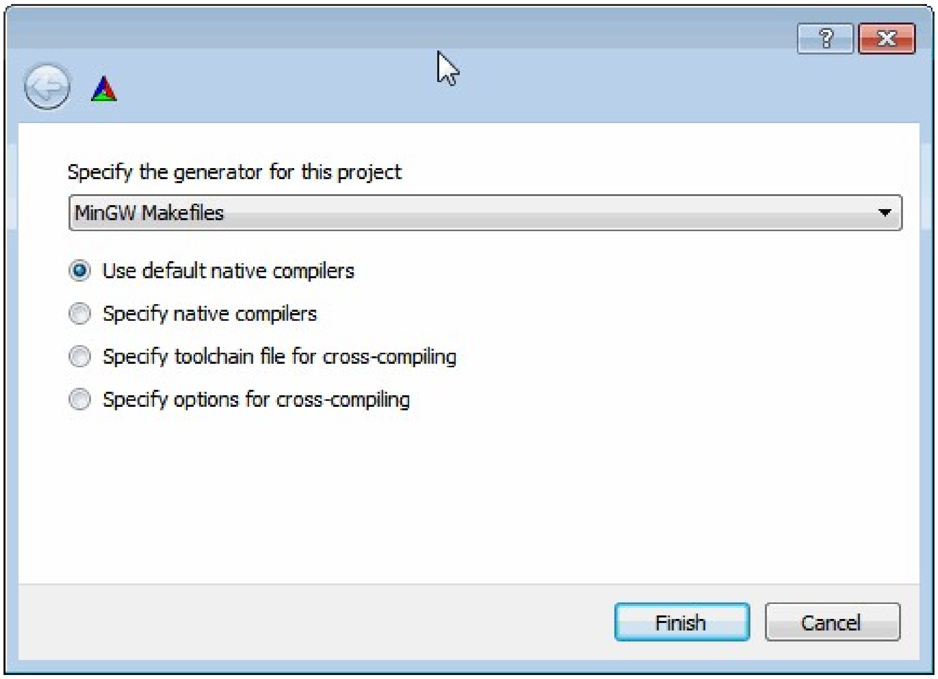
6. Download the RTL-SDR source using git at git://git.osmocom.org/rtl-sdr or from https://github.com/steve-m/librtlsdr by downloading the zip file or doing a git clone. Extract to a directory like c:\rtl-sdr.

7. Open cmake-gui from the start menu.

8. Set the source code folder as c:\rtl-sdr

9. Set the binary build folder as c:\rtl-sdr\build. Press Configure, you will be asked to create the build directory. Click yes.

10. Choose “MinGW Makefiles” as the compiler for this project and “Use default native compilers” and then click Finish.



Specify the generator for this project

MinGW Makefiles

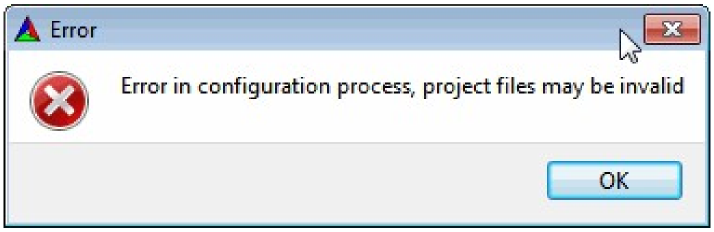
● Use default native compilers

○ Specify native compilers

○ Specify toolchain file for cross-compiling

○ Specify options for cross-compiling

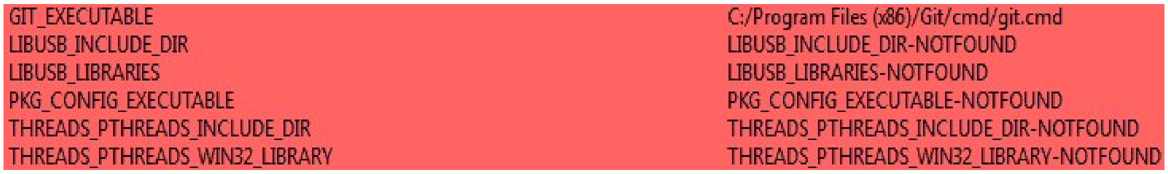
11. You may receive an error, but don't worry about it yet.



Error

Error in configuration process, project files may be invalid

12. Now make sure the advanced check box is checked. Scroll to the bottom of the window. You should see 5 library names that say that their directory was not found.



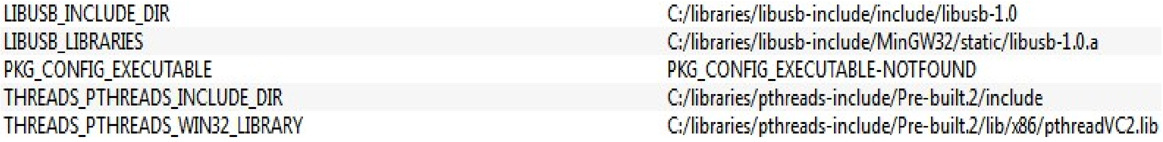
GIT_EXECUTABLE	C:/Program Files (x86)/Git/cmd/git.cmd
LIBUSB_INCLUDE_DIR	LIBUSB_INCLUDE_DIR-NOTFOUND
LIBUSB_LIBRARIES	LIBUSB_LIBRARIES-NOTFOUND
PKG_CONFIG_EXECUTABLE	PKG_CONFIG_EXECUTABLE-NOTFOUND
THREADS_PTHREADS_INCLUDE_DIR	THREADS_PTHREADS_INCLUDE_DIR-NOTFOUND
THREADS_PTHREADS_WIN32_LIBRARY	THREADS_PTHREADS_WIN32_LIBRARY-NOTFOUND

13. Set LIBUSB_INCLUDE_DIR to c:\libraries\libusb-include\include\libusb-1.0\

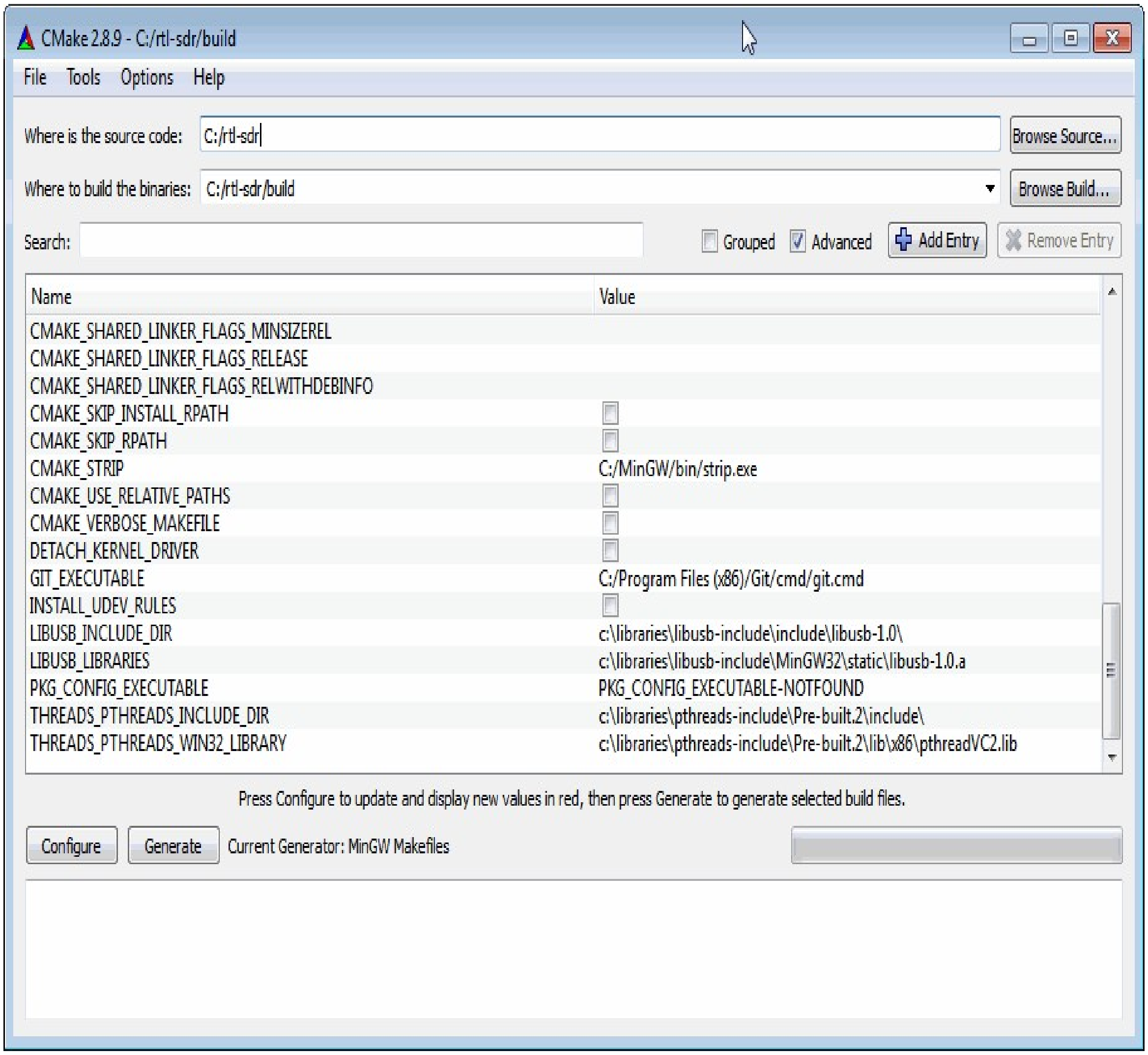
14. Set LIBUSB_LIBRARIES to c:\libraries\libusb-include\MinGW32\static\libusb-1.0.a

15. Set THREADS_PTHREADS_INCLUDE_DIR to c:\libraries\pthreads-include\Pre-built.2\include\

16. Set THREADS_PTHREADS_WIN32_LIBRARY to c:\libraries\pthreads-include\Pre-built.2\lib\x86\pthreadVC2.lib



LIBUSB_INCLUDE_DIR	C:/libraries/libusb-include/include/libusb-1.0
LIBUSB_LIBRARIES	C:/libraries/libusb-include/MinGW32/static/libusb-1.0.a
PKG_CONFIG_EXECUTABLE	PKG_CONFIG_EXECUTABLE-NOTFOUND
THREADS_PTHREADS_INCLUDE_DIR	C:/libraries/pthreads-include/Pre-built.2/include
THREADS_PTHREADS_WIN32_LIBRARY	C:/libraries/pthreads-include/Pre-built.2/lib/x86/pthreadVC2.lib



Where is the source code: C:/rtl-sdr

Where to build the binaries: C:/rtl-sdr/build

Search:

☐ Grouped ☑ Advanced

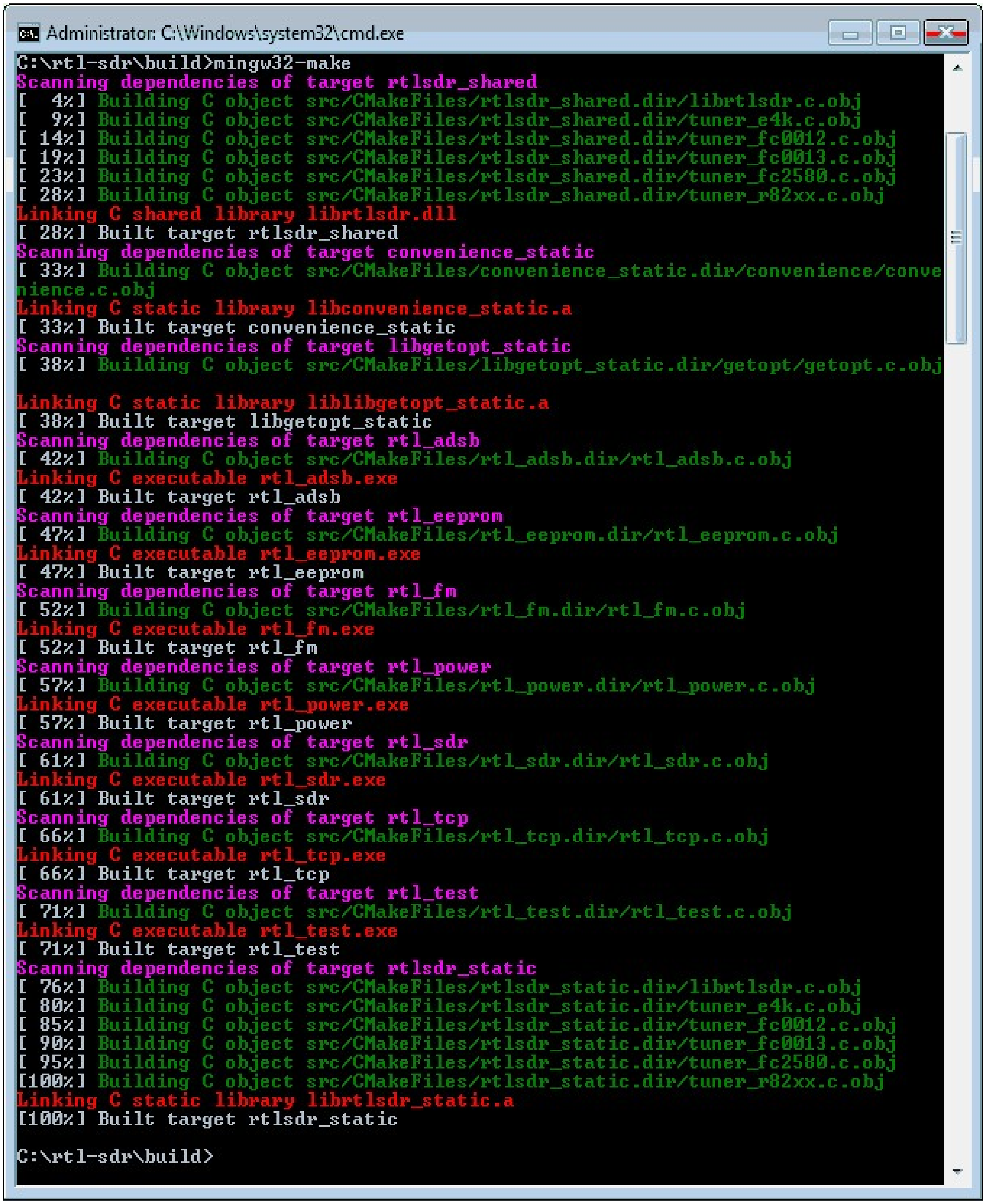
Name	Value
CMAKE_SHARED_LINKER_FLAGS_MINSIZEREL	
CMAKE_SHARED_LINKER_FLAGS_RELEASE	
CMAKE_SHARED_LINKER_FLAGS_RELWITHDEBINFO	
CMAKE_SKIP_INSTALL_RPATH	☐
CMAKE_SKIP_RPATH	☐
CMAKE_STRIP	C:/MinGW/bin/strip.exe
CMAKE_USE_RELATIVE_PATHS	☐
CMAKE_VERBOSE_MAKEFILE	☐
DETACH_KERNEL_DRIVER	☐
GIT_EXECUTABLE	C:/Program Files (x86)/Git/cmd/git.cmd
INSTALL_UDEV_RULES	☐
LIBUSB_INCLUDE_DIR	c:\libraries\libusb-include\include\libusb-1.0\
LIBUSB_LIBRARIES	c:\libraries\libusb-include\MinGW32\static\libusb-1.0.a
PKG_CONFIG_EXECUTABLE	PKG_CONFIG_EXECUTABLE-NOTFOUND
THREADS_PTHREADS_INCLUDE_DIR	c:\libraries\pthreads-include\Pre-built.2\include\
THREADS_PTHREADS_WIN32_LIBRARY	c:\libraries\pthreads-include\Pre-built.2\lib\x86\pthreadVC2.lib

Press Configure to update and display new values in red, then press Generate to generate selected build files.

Current Generator: MinGW Makefiles

17. Press Configure. After configuring is successfully completed press build.

18. Now in CMD prompt, go to the c:\rtl-sdr\build directory. Type “mingw32-make” to compile.



```
C:\rtl-sdr\build>mingw32-make
Scanning dependencies of target rtlsdr_shared
[  4%] Building C object src/CMakeFiles/rtlsdr_shared.dir/librtlsdr.c.obj
[  9%] Building C object src/CMakeFiles/rtlsdr_shared.dir/tuner_e4k.c.obj
[ 14%] Building C object src/CMakeFiles/rtlsdr_shared.dir/tuner_fc0012.c.obj
[ 19%] Building C object src/CMakeFiles/rtlsdr_shared.dir/tuner_fc0013.c.obj
[ 23%] Building C object src/CMakeFiles/rtlsdr_shared.dir/tuner_fc2580.c.obj
[ 28%] Building C object src/CMakeFiles/rtlsdr_shared.dir/tuner_r82xx.c.obj
Linking C shared library librtlsdr.dll
[ 28%] Built target rtlsdr_shared
Scanning dependencies of target convenience_static
[ 33%] Building C object src/CMakeFiles/convenience_static.dir/convenience/conve
nience.c.obj
Linking C static library libconvenience_static.a
[ 33%] Built target convenience_static
Scanning dependencies of target libgetopt_static
[ 38%] Building C object src/CMakeFiles/libgetopt_static.dir/getopt/getopt.c.obj

Linking C static library liblibgetopt_static.a
[ 38%] Built target libgetopt_static
Scanning dependencies of target rtl_adsb
[ 42%] Building C object src/CMakeFiles/rtl_adsb.dir/rtl_adsb.c.obj
Linking C executable rtl_adsb.exe
[ 42%] Built target rtl_adsb
Scanning dependencies of target rtl_eeprom
[ 47%] Building C object src/CMakeFiles/rtl_eeprom.dir/rtl_eeprom.c.obj
Linking C executable rtl_eeprom.exe
[ 47%] Built target rtl_eeprom
Scanning dependencies of target rtl_fm
[ 52%] Building C object src/CMakeFiles/rtl_fm.dir/rtl_fm.c.obj
Linking C executable rtl_fm.exe
[ 52%] Built target rtl_fm
Scanning dependencies of target rtl_power
[ 57%] Building C object src/CMakeFiles/rtl_power.dir/rtl_power.c.obj
Linking C executable rtl_power.exe
[ 57%] Built target rtl_power
Scanning dependencies of target rtl_sdr
[ 61%] Building C object src/CMakeFiles/rtl_sdr.dir/rtl_sdr.c.obj
Linking C executable rtl_sdr.exe
[ 61%] Built target rtl_sdr
Scanning dependencies of target rtl_tcp
[ 66%] Building C object src/CMakeFiles/rtl_tcp.dir/rtl_tcp.c.obj
Linking C executable rtl_tcp.exe
[ 66%] Built target rtl_tcp
Scanning dependencies of target rtl_test
[ 71%] Building C object src/CMakeFiles/rtl_test.dir/rtl_test.c.obj
Linking C executable rtl_test.exe
[ 71%] Built target rtl_test
Scanning dependencies of target rtlsdr_static
[ 76%] Building C object src/CMakeFiles/rtlsdr_static.dir/librtlsdr.c.obj
[ 80%] Building C object src/CMakeFiles/rtlsdr_static.dir/tuner_e4k.c.obj
[ 85%] Building C object src/CMakeFiles/rtlsdr_static.dir/tuner_fc0012.c.obj
[ 90%] Building C object src/CMakeFiles/rtlsdr_static.dir/tuner_fc0013.c.obj
[ 95%] Building C object src/CMakeFiles/rtlsdr_static.dir/tuner_fc2580.c.obj
[100%] Building C object src/CMakeFiles/rtlsdr_static.dir/tuner_r82xx.c.obj
Linking C static library librtlsdr_static.a
[100%] Built target rtlsdr_static

C:\rtl-sdr\build>
```

Finally, librtlsdr.dll can then be found in C:\rtl-sdr\build\src

COMPILING USING VISUAL C++ 2010

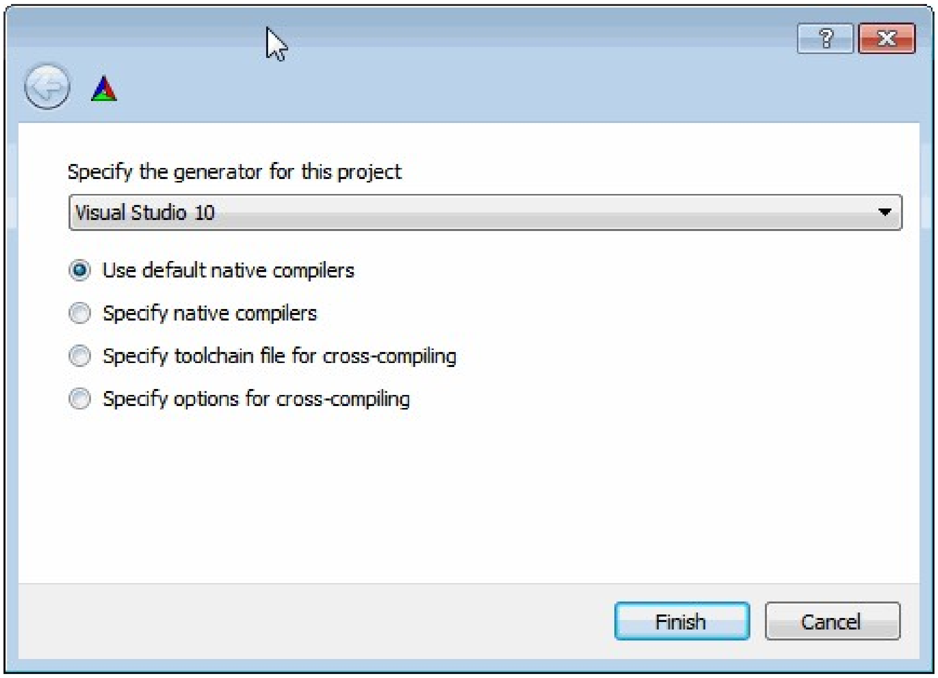
The RTL-SDR drivers can also be compiled using Visual Studio. Compiling this way may be more portable as most people will be likely to already have the required runtimes as they are usually included with Windows.

1. Install the Windows SDK. Here is the download link for Windows 7 http://www.microsoft.com/en-us/download/details.aspx?id=8279.

2. Install Visual C++ 2010. Compilation should also work on newer Visual C++ versions. http://www.visualstudio.com/en-us/downloads#d-2010-express.

3. You may also need to install service pack 1 for Visual C++ 2010 http://www.microsoft.com/en-us/download/details.aspx?id=4422.

4. Follow the same steps for CMAKE as when compiling for MingW as shown in the above section, but instead choose Visual Studio 10 when specifying the generator.



Specify the generator for this project

Visual Studio 10

● Use default native compilers

○ Specify native compilers

○ Specify toolchain file for cross-compiling

○ Specify options for cross-compiling

5. Add the same libraries as with the MingW compile but instead for LIBUSB_LIBRARIES use c:\libraries\libusb-include\MS32\static\libusb-1.0.lib.

6. Press Configure then Generate.

7. Now go to c:\rtl-sdr\build\ in Windows explorer and open the rtlsdr.sln file with Visual C++ Express 2010.

8. In the GUI change the Build from Debug to Release in the top menu of Visual Stuio, then press F7 to build the files.

The rtlsdr.dll and other built files can then be found in c:\rtl-sdr\build\src\Release

PANADAPTER GUIDE

A panadapter is a device that connects to a traditional analogue hardware radio and displays a visualization of the spectrum through a RF spectrum and waterfall display. Most RTL-SDR software shows the RF spectrum and waterfall by default so the RTL-SDR can be used as a super cheap panadapter. This can give you the great performance of a dedicated hardware radio and the visual RF display common to software defined radios.

Creating an RTL-SDR panadapter is radio hardware dependant as you will need to discover a way to connect the RTL-SDR antenna input into the first IF of your hardware radio. Once the IF output is connected to the antenna input, you simply tune to the IF frequency using the RTL-SDR to visualize the signals on the waterfall.

Some older or cheaper radios may need a little hardware hacking to access this IF output and some newer radio will have the IF as an available out port on the back of the radio. Most hardware radios will also allow you to control it via GUI software like Omnirig HDSDR if you have a control cable.

Here is an excellent tutorial showing the steps needed to set up a RTL-SDR panadapter for the ICOM IC-7600 radio http://www.ab4oj.com/icom/ic7600/we1x_rtl_sdr3_0.pdf. The tutorial should also be valid for other radios with an easy IF output.

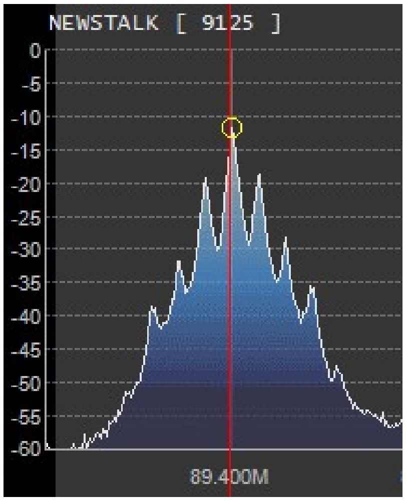
A good tutorial on doing this for the Yaesu FT-857 can be found on this YouTube video https://www.youtube.com/watch?v=tshlXgarnBQ#t=164. Another video showing the performance of the FT-857 with RTL-SDR panadapter can be found here https://www.youtube.com/watch?v=wqcGACGPtrM**.** An excellent pdf showing the modifications required for many Yaesu radios is here http://www.radioamatoripeligni.it/i6ibe/pdf/panadapter.pdf (note in Italian but with many good images).

Here is a tutorial for the Heathkit HR-10 which does not have an easy IF output https://www.youtube.com/watch?v=BRVAT9jG4kg.

RDS RECEIVING GUIDE

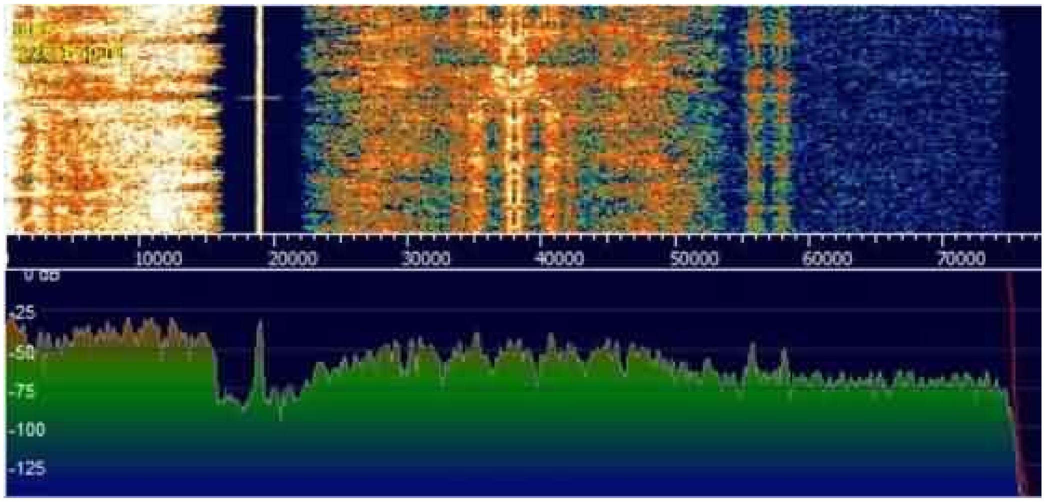
RDS stands for Radio Data System and is a digital signal embedded into broadcast FM signals. It is used by radio stations to display the name of the radio station and/or current song playing on an LCD screen.

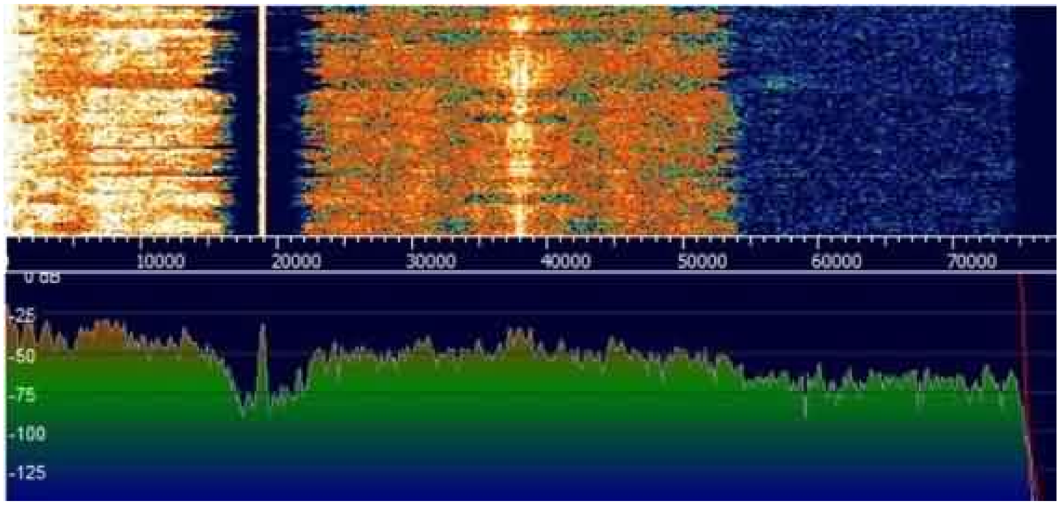
SDR# automatically decodes RDS data in the top left of the RF spectrum when tuned to a broadcast radio station with the WFM mode set.



NEWSTALK [9125]

If you are unsure if a radio station has RDS or not you can check it in SDR# by using the “Enable MPX” option in the Zoom FFT plugin. This will open a new RF display showing the FM MPX Spectrum. If RDS is used on a radio station, you should be able to see a peak at about 57k. On HDSDR the RDS peak is much easier to see. You can check visually in HDSDR’s audio spectrum to see if the station you are tuned to has an RDS signal. In the example screenshot below there is a thin signal to the right of the spectrum - this is the RDS signal. An example of an FM station without RDS is shown below it.



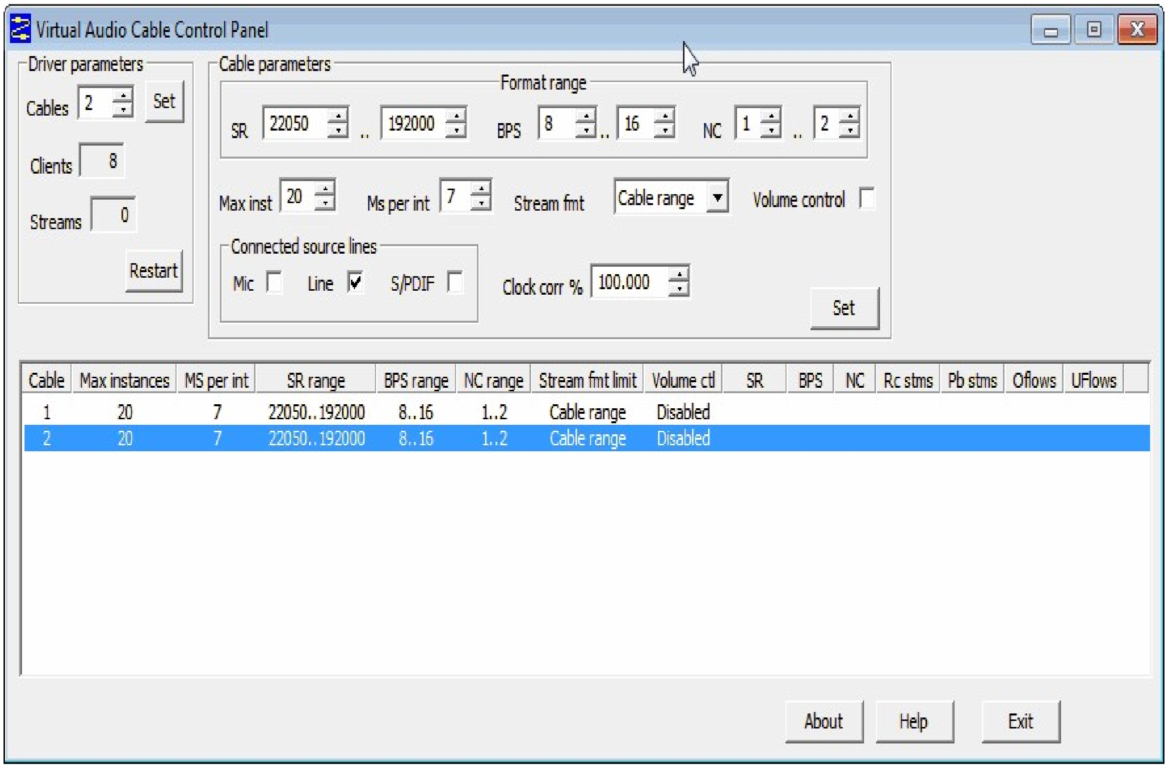


Another program often used for RDS decoding is RDS Spy which is a Windows program. RDS Spy has better sensitivity and analysis options so it is better for dxing (very long range distance reception). RDS Spy can be downloaded from http://rdsspy.com/. RDS Spy will only work with SDR# if a special plugin is used. This is because by default SDR# removes the RDS tone from the output audio. RDS Spy can also work with HDSDR. The first step to decoding RDS with RDS Spy is to set up Virtual Audio Cable or Hi-Fi Cable with the correct sample rate. Only Virtual Audio Cable and Hi-Fi Cable can be used as they are the only virtual cable's that can support a sample rate of 1920000, which is what RDS Spy needs to work. To set up Hi-Fi Cable, simply set the sample rate of Input and Output Hi-Fi cables to 1920000 in the Windows Playback and Recording properties. Use the following instructions to correctly set up Virtual Audio Cable.

1. You will need to have downloaded and installed Virtual Audio Cable (VAC) first. Since the trial version supports up to three free cables so it will be sufficient for this task. The watermark audio played by the trial version of VAC will not pose a major problem.

2. Go to **Start Menu -> Virtual Audio Cable -> Control Panel**. Note you may need to open the virtual audio cable control panel as administrator.

3. Create two audio cables which have a maximum sample rate of 192000 as shown in the screenshot below (two cables are needed if using HDSDR, only one if using SDR#). Be sure to click on the Set button after changing the options sample rate to make your settings permanent.



Virtual Audio Cable Control Panel

Driver parameters

Cables 2 Set

Clients 8

Streams 0

Restart

Cable parameters

Format range

SR 22050 .. 192000 BPS 8 .. 16 NC 1 .. 2

Max inst 20 Ms per int 7 Stream fmt Cable range Volume control □

Connected source lines

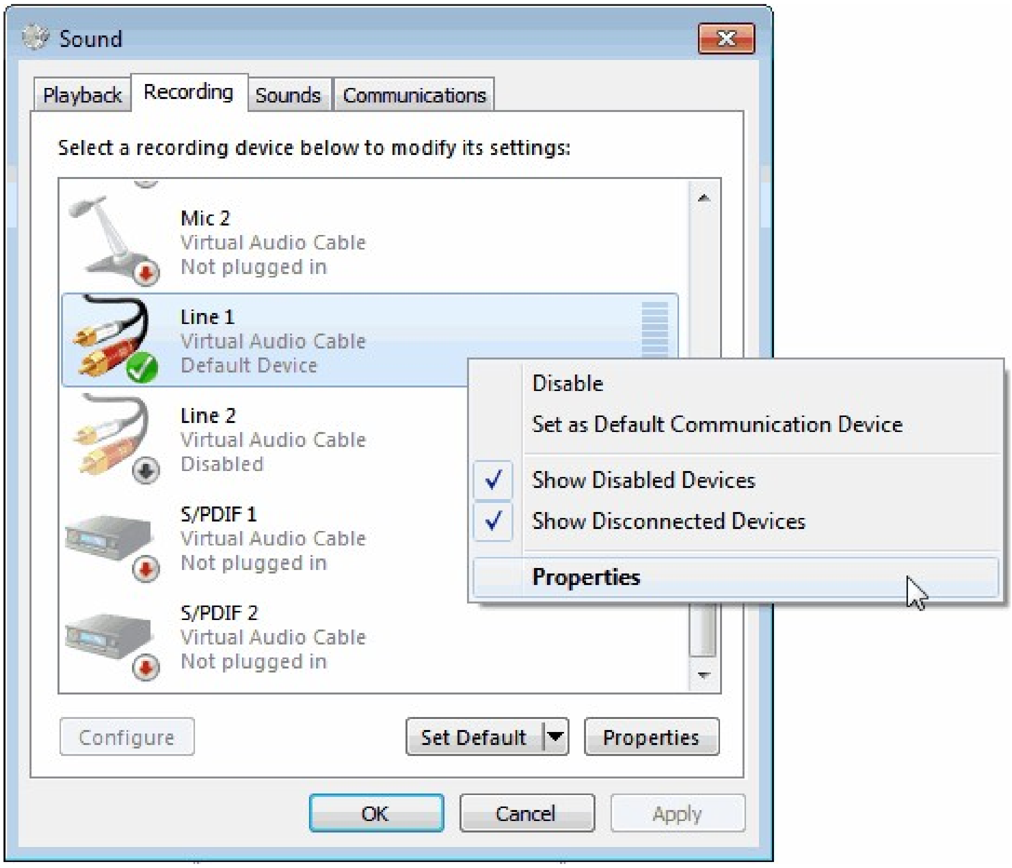
Mic □ Line ☑ S/PDIF □

Clock corr % 100.000

Set

Cable	Max instances	MS per int	SR range	BPS range	NC range	Stream fmt limit	Volume ctl	SR	BPS	NC	Rc stms	Pb stms	Oflows	UFlows
1	20	7	22050..192000	8..16	1..2	Cable range	Disabled							
2	20	7	22050..192000	8..16	1..2	Cable range	Disabled							

4. In the Windows recording and playback options set each virtual audio cable device to a sample rate of 192000 Hz in the advanced options. You can access the advanced options by right clicking on the virtual audio cable device, clicking properties and then going to the Advanced tab. Set the default format to a setting with 192000 Hz in it. Do this for **ALL** virtual audio cable devices in both the Recording **AND** Playback tabs.



Select a recording device below to modify its settings:

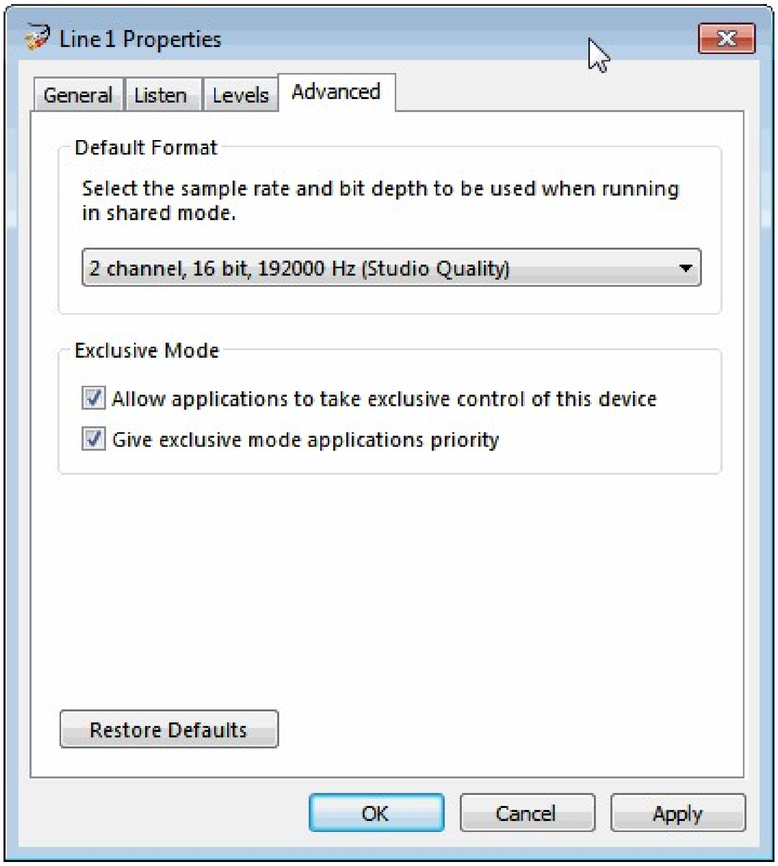
Mic 2
Virtual Audio Cable
Not plugged in

Line 1
Virtual Audio Cable
Default Device

Line 2
Virtual Audio Cable
Disabled

S/PDIF 1
Virtual Audio Cable
Not plugged in

S/PDIF 2
Virtual Audio Cable
Not plugged in



Default Format

Select the sample rate and bit depth to be used when running in shared mode.

2 channel, 16 bit, 192000 Hz (Studio Quality)

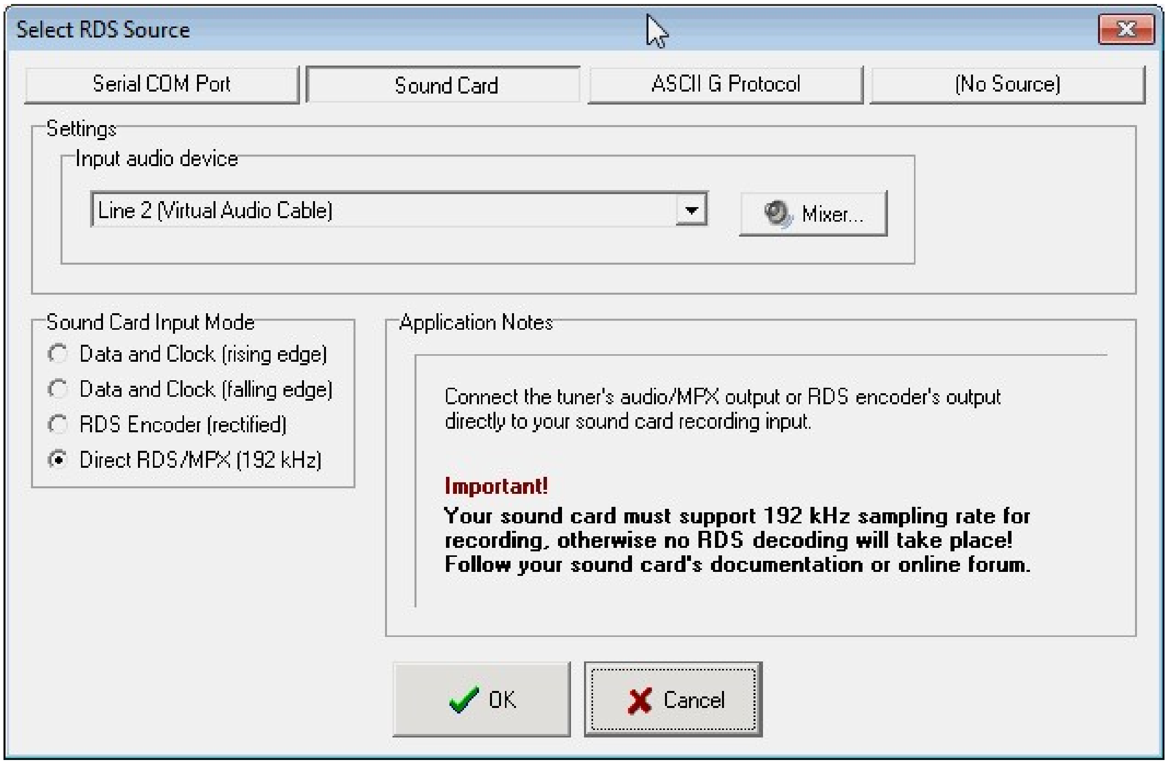
Exclusive Mode

☑ Allow applications to take exclusive control of this device

☑ Give exclusive mode applications priority

Now to use RDS Spy with SDR# use the following intructions:

1. Download the MPX Output SDR# plugin from http://rtl-sdr.ru/page/vse-dostupnye-na-etom-sajte-plaginy-s-kratkim-opisaniem (note that this page is in Russian, so use Google translate if necessary), and install it into SDR#.
2. Open SDR# and set the output audio device to the Virtual Audio Cable that you have set up with a 1920000 sample rate. Tune to a broadcast FM station which has RDS using the WFM mode.
3. In the MPX Output plugin select the virtual audio cable as the audio device and click "Enable MPX output".
4. Open RDS Spy and go to Configure -> Select RDS Source.
5. Select the Sound Card tab and then under Input audio device select the virtual audio cable device and select the Sound Card Input Mode as "Direct RDS/ MPX (192 kHz). Press OK.



Settings

Input audio device: Line 2 (Virtual Audio Cable)

Sound Card Input Mode

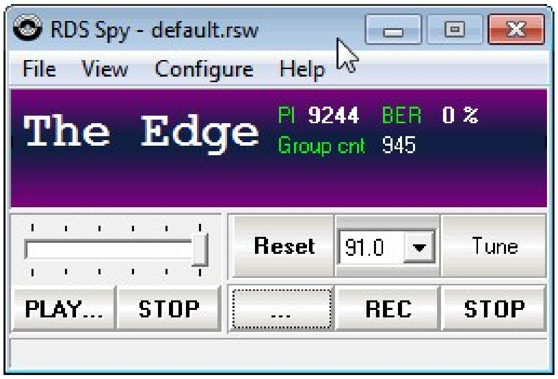
○ Data and Clock (rising edge)
○ Data and Clock (falling edge)
○ RDS Encoder (rectified)
● Direct RDS/MPX (192 kHz)

Application Notes

Connect the tuner's audio/MPX output or RDS encoder's output directly to your sound card recording input.

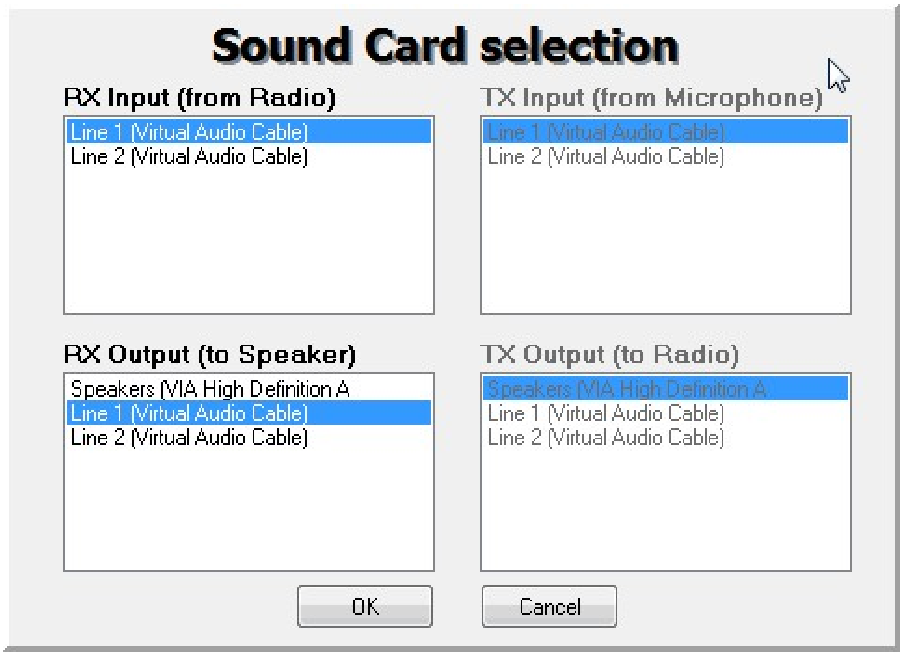
Important!
Your sound card must support 192 kHz sampling rate for recording, otherwise no RDS decoding will take place! Follow your sound card's documentation or online forum.

6. RDS Spy should now be decoding RDS signals.

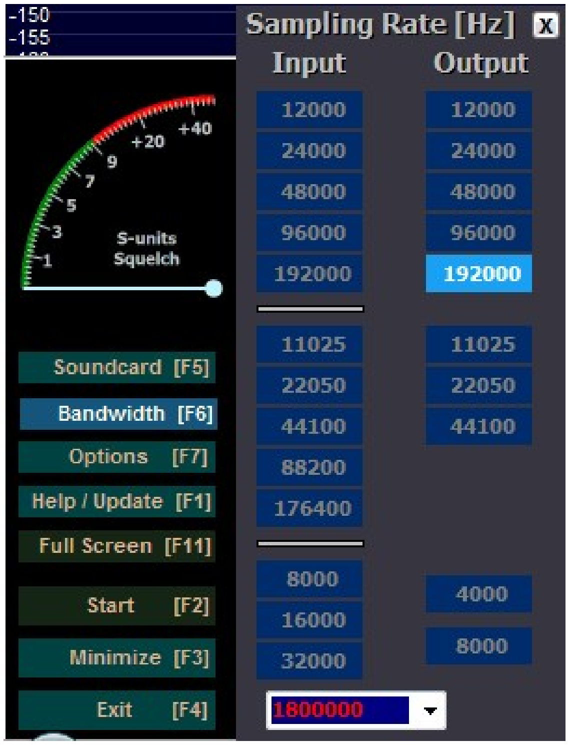


As an alternative, you can use RDSSpy with HDSDR. To use HDSDR you must have set up two Virtual Audio Cables with a sample rate of 1920000 as shown in the set up instructions above.

1. Open HDSDR.
2. In HDSDR click the Soundcard button in the lower left corner and then select the first virtual audio cable device under RX Output (to Speaker).


Sound Card selection
RX Input (from Radio)
Line 1 (Virtual Audio Cable)
Line 2 (Virtual Audio Cable)
TX Input (from Microphone)
Line 1 (Virtual Audio Cable)
Line 2 (Virtual Audio Cable)
RX Output (to Speaker)
Speakers (VIA High Definition A
Line 1 (Virtual Audio Cable)
Line 2 (Virtual Audio Cable)
TX Output (to Radio)
Speakers (VIA High Definition A
Line 1 (Virtual Audio Cable)
Line 2 (Virtual Audio Cable)
OK
Cancel

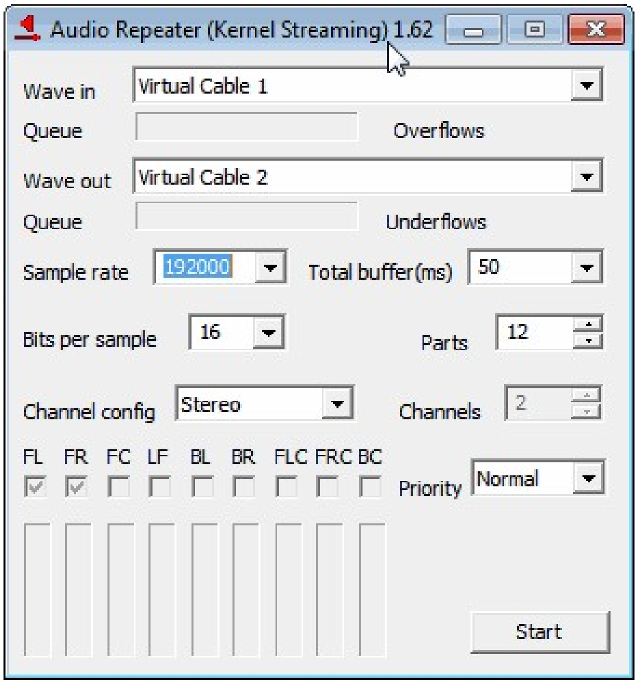
3. Click the Bandwidth button in the lower left corner and then set the Output to 192000.



4. Now click Start in HDSDR and tune to a broadcast FM station that has RDS. You can check visually in HDSDR's audio spectrum to see if the station you are tuned to has an RDS signal. In the example screenshot below there is a thin signal to the right of the spectrum - this is the RDS signal. An example of an FM station without RDS is shown below it.

5. Go to **Start Menu -> Virtual Audio Cable -> Audio Repeater (Kernel Streaming).**

6. Set the Wave in to the first virtual audio cable and the Wave out to the second virtual audio cable. Also set the sample rate to 192000. Click Start. If you start getting overflows (increasing numbers next to the word 'Overflows'), stop the audio repeater and increase the Total buffer(ms) setting until the overflows stop.



Audio Repeater (Kernel Streaming) 1.62

Wave in: Virtual Cable 1

Queue | Overflows

Wave out: Virtual Cable 2

Queue | Underflows

Sample rate: 192000 | Total buffer(ms): 50

Bits per sample: 16 | Parts: 12

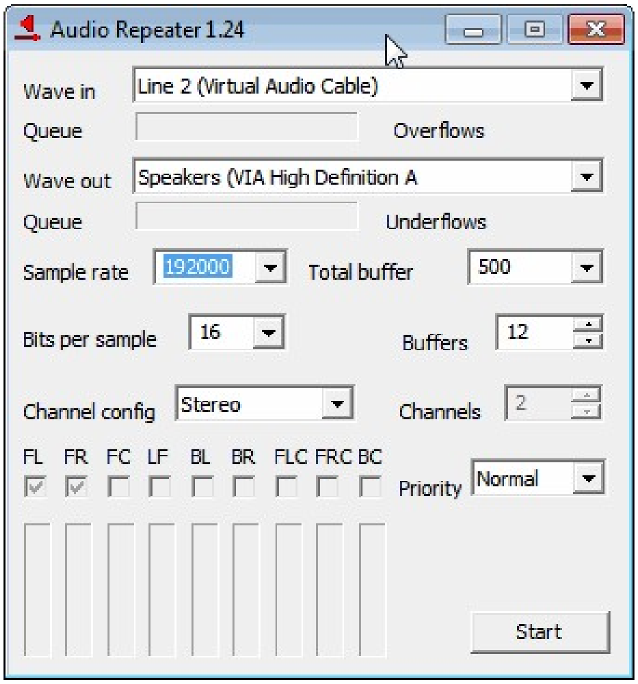
Channel config: Stereo | Channels: 2

FL	FR	FC	LF	BL	BR	FLC	FRC	BC
☑	☑	□	□	□	□	□	□	□

Priority: Normal

7. OPTIONAL (If you want to listen to the FM audio): Go to **Start Menu -> Virtual Audio Cable -> Audio Repeater (MME).**

8. OPTIONAL CONTINUED: Set the Wave in to the second virtual audio cable and set the Wave out to your speakers. This will allow you to simultaneously listen to the FM station audio.



Audio Repeater 1.24

Wave in: Line 2 (Virtual Audio Cable)

Queue | Overflows

Wave out: Speakers (VIA High Definition A

Queue | Underflows

Sample rate: 192000 | Total buffer: 500

Bits per sample: 16 | Buffers: 12

Channel config: Stereo | Channels: 2

FL	FR	FC	LF	BL	BR	FLC	FRC	BC
☑	☑	□	□	□	□	□	□	□

Priority: Normal

9. Open RDS Spy and go to **Configure -> Select RDS Source**.

10. Select the Sound Card tab and then under Input audio device select the second virtual audio cable device and select the Sound Card Input Mode as “Direct RDS/MPX (192 kHz). Press OK.

11. RDS Spy should now be decoding RDS signals.

For those interested there is also a GNU Radio RDS decoder implementation which is also extremely useful for learning how RDS is decoded. It is available at https://www.cgran.org/wiki/RDS.

LISTENING TO SCA/SCMO FM SUBCARRIER CHANNELS

In some countries standard broadcast FM radio contains “subcarriers” which contain either data or additional audio services. An example of one such data subcarrier is RDS which is explained in the tutorial above. In some countries an audio SCA subcarrier exists and is used to provide extra transmissions for things like reading services for the blind. It is possible to listen to these audio SCA transmissions using a combination of SDR# with the MPX Output plugin or HDSDR, Virtual Audio Cable and SDR#. The MPX Output plugin for SDR# can be downloaded from the list at http://rtl-sdr.ru/page/vse-dostupnye-na-etom-sajte-plaginy-s-kratkim-opisaniem (note that this page is in Russian, so use Google translate if necessary).

You can check if there are audio SCA services in your country by using the FM MPX Spectrum in SDR#, or the audio spectrum waterfall in HDSDR, just like what is shown in the above RDS tutorial. An audio SCA signal will be to the right of any RDS signal.

HDSDR or SDR# with the MPX Output plugin must be used as the RTL-SDR receiver as the raw FM audio with subcarriers is distorted by SDR# or SDR-Radio when the WFM mode is used because the subcarriers are truncated. It would be okay if the NFM mode could be used, however the maximum bandwidth of the NFM modes in SDR# or SDR-Radio are not large enough to cover the width of a broadcast FM signal. You also need to ensure that Virtual Audio Cable of Hi-Fi Cable, not VBCable is not used as the audio pipe. This is because VBCable can only support up to a 96 kHz bandwidth and a 196 kHz bandwidth is required to see the SCA subcarriers. Ensure that the Virtual Audio Cable you are using is set to have a maximum sample rate of 192000 Hz, as is shown in the previous RDS tutorial.

1. Open HDSDR of SDR# with the MPX Plugin and tune to a broadcast FM signal with an audio SCA subcarrier. Ensure that the MPX audio is outputting to Virtual Audio Cable. (in SDR# choose the Virtual Audio Cable as the output in the MPX Output plugin).

2. If using HDSDR, adjust the Audio Bandpass filter in HDSDR so that it is fully wide up to at least 96 kHz. (This is adjusted by using the in the red lines in the RF spectrum display at the bottom of the HDSDR window)

3. Open a new copy of SDR# and set the receiving device to “Other (Sound Card)”. Set the audio input under the “Audio” heading in the SDR# sidebar to your Virtual Audio Cable and set the Output to your speakers.

4. Start SDR# by clicking on play the play button.

5. You should now be seeing a typical WFM audio signal in SDR# that is mirrored. You should notice the L+R mono audio signal from DC to 15 kHz, the 19 kHz stero pilot tone and the L+R stereo signal at 23 kHz to 53 kHz. Depending on the configuration of the broadcaster you may also see an RDS signal at 57 kHz, another data signal at 67 kHz and an SCA audio signal at 92 kHz.

6. To decode an audio SCA signal simply choose the DSB mode in SDR# and center the tuning on the SCA signal which should be at 92 kHz.

6.

APRS GUIDE

APRS is an acronym for Automatic Position Reporting System and is a packet based system used by Amateur radio hobbyists. APRS is used to transmit data such as short messages, announcements, weather station data and also to report GPS coordinates of things like transmitters and moving vehicles.

APRS data is often collected internationally on an internet server through an iGate. Some iGates also selectively retransmit internet APRS data to the local RF network. Using the internet and iGate transceivers APRS messages can be transmitted overseas.

APRS is assigned to different frequencies around the world, but is usually around the region of 144 MHz. A table showing common APRS frequencies is shown below.

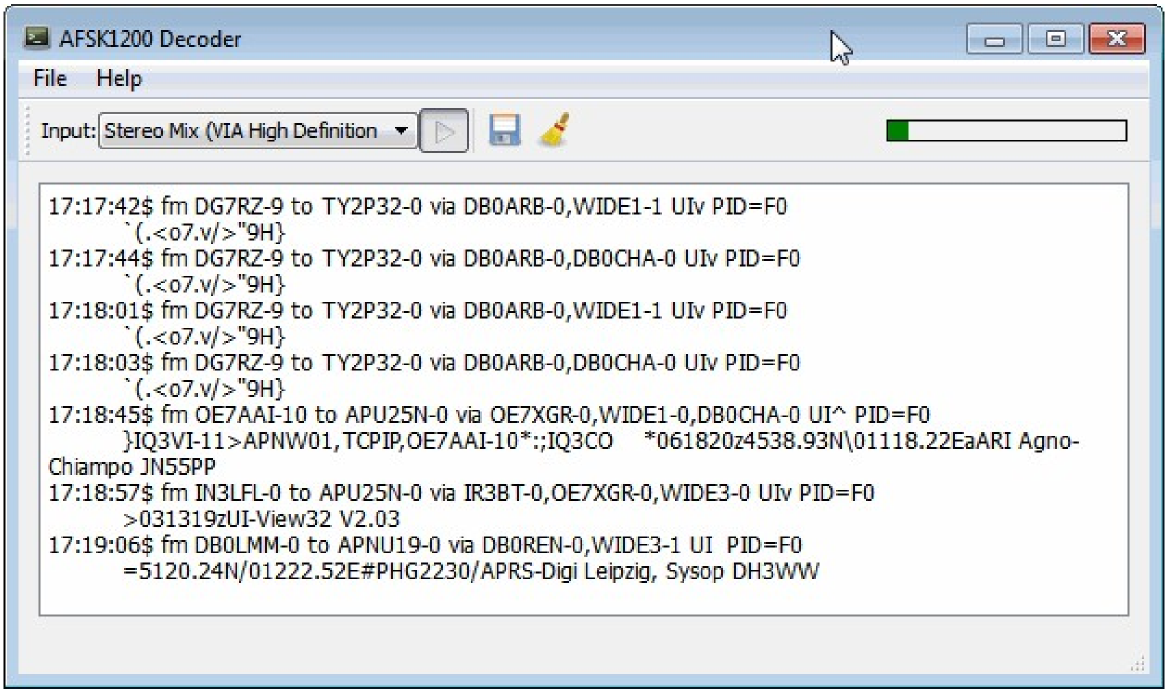
FREQ (MHz)	**REGION / COUNTRY**
144.390	Colombia, Chile, Indonesia, North America
144.575	New Zealand
144.660	Japan
144.800	South Africa, Europe, Russia
144.930	Argentina, Uruguay
145.175	Australia
145.570	Brazil
145.525	Thailand

QTMM AFSK1200

Since APRS is transmitted using Asynchronous Frequency Shift Keying (AFSK) at 1200 bits/s the software program QTMM AFSK1200 can be used to decode the messages. QTMM AFSK1200 can be downloaded from http://sourceforge.net/projects/qtmm/. To use AFSK1200 follow the instructions below.

1. Open SDR# or your favourite SDR receiver and set the audio output to your preferred audio piping method.
2. Tune to a local APRS frequency and set the mode to NFM. Set the bandwidth large enough to cover the packet width.
3. Open AFSK1200 and under Input select the audio piping method you are using.
4. Press the Play button ▷ to begin listening.

Decoded message will show up in the main window.

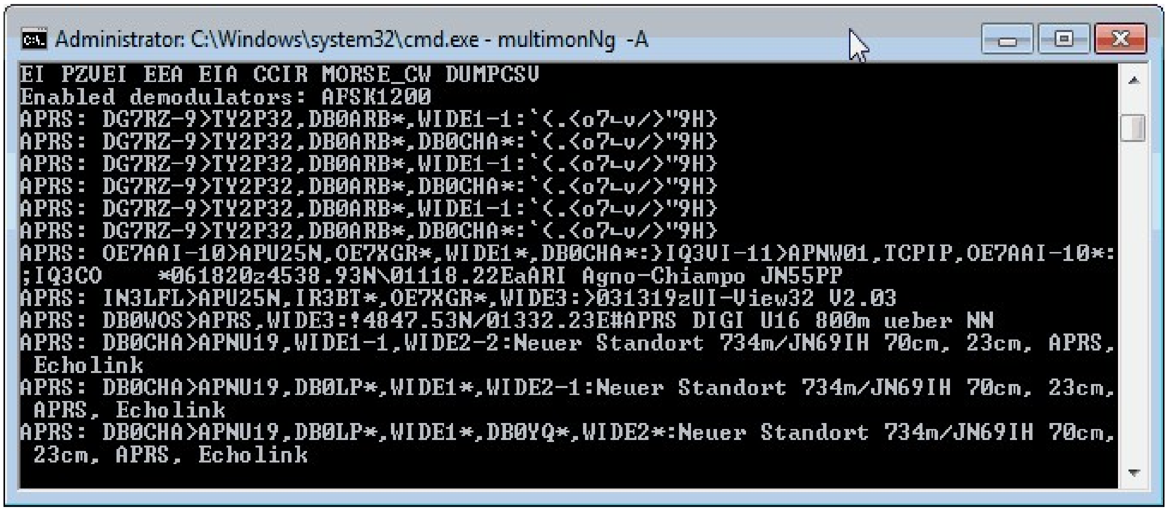


Input: Stereo Mix (VIA High Definition

```
17:17:42$ fm DG7RZ-9 to TY2P32-0 via DB0ARB-0,WIDE1-1 UIv PID=F0
        `(.<o7.v/>"9H}
17:17:44$ fm DG7RZ-9 to TY2P32-0 via DB0ARB-0,DB0CHA-0 UIv PID=F0
        `(.<o7.v/>"9H}
17:18:01$ fm DG7RZ-9 to TY2P32-0 via DB0ARB-0,WIDE1-1 UIv PID=F0
        `(.<o7.v/>"9H}
17:18:03$ fm DG7RZ-9 to TY2P32-0 via DB0ARB-0,DB0CHA-0 UIv PID=F0
        `(.<o7.v/>"9H}
17:18:45$ fm OE7AAI-10 to APU25N-0 via OE7XGR-0,WIDE1-0,DB0CHA-0 UI^ PID=F0
        }IQ3VI-11>APNW01,TCPIP,OE7AAI-10*:;IQ3CO    *061820z4538.93N\01118.22EaARI Agno-
Chiampo JN55PP
17:18:57$ fm IN3LFL-0 to APU25N-0 via IR3BT-0,OE7XGR-0,WIDE3-0 UIv PID=F0
        >031319zUI-View32 V2.03
17:19:06$ fm DB0LMM-0 to APNU19-0 via DB0REN-0,WIDE3-1 UI  PID=F0
        =5120.24N/01222.52E#PHG2230/APRS-Digi Leipzig, Sysop DH3WW
```

MULTIMONNG

MultimonNG can also be used to decode APRS / AFSK1200. Use the command multimonng -A. This will only decode APRS messages and not APRS pings. If you want to see everything use multimonng -a AFSK1200.



```
EI PZVEI EEA EIA CCIR MORSE_CW DUMPCSV
Enabled demodulators: AFSK1200
APRS: DG7RZ-9>TY2P32,DB0ARB*,WIDE1-1:`(.<o7└v/>"9H}
APRS: DG7RZ-9>TY2P32,DB0ARB*,DB0CHA*:`(.<o7└v/>"9H}
APRS: DG7RZ-9>TY2P32,DB0ARB*,WIDE1-1:`(.<o7└v/>"9H}
APRS: DG7RZ-9>TY2P32,DB0ARB*,DB0CHA*:`(.<o7└v/>"9H}
APRS: DG7RZ-9>TY2P32,DB0ARB*,WIDE1-1:`(.<o7└v/>"9H}
APRS: DG7RZ-9>TY2P32,DB0ARB*,DB0CHA*:`(.<o7└v/>"9H}
APRS: OE7AAI-10>APU25N,OE7XGR*,WIDE1*,DB0CHA*:}IQ3VI-11>APNW01,TCPIP,OE7AAI-10*:
;IQ3CO    *061820z4538.93N\01118.22EaARI Agno-Chiampo JN55PP
APRS: IN3LFL>APU25N,IR3BT*,OE7XGR*,WIDE3:>031319zUI-View32 V2.03
APRS: DB0WOS>APRS,WIDE3:!4847.53N/01332.23E#APRS DIGI U16 800m ueber NN
APRS: DB0CHA>APNU19,WIDE1-1,WIDE2-2:Neuer Standort 734m/JN69IH 70cm, 23cm, APRS,
 Echolink
APRS: DB0CHA>APNU19,DB0LP*,WIDE1*,WIDE2-1:Neuer Standort 734m/JN69IH 70cm, 23cm,
 APRS, Echolink
APRS: DB0CHA>APNU19,DB0LP*,WIDE1*,DB0YQ*,WIDE2*:Neuer Standort 734m/JN69IH 70cm,
 23cm, APRS, Echolink
```

On Linux rtl_fm can be used as the receiver by using the line shown below. The output could be further piped into other software.

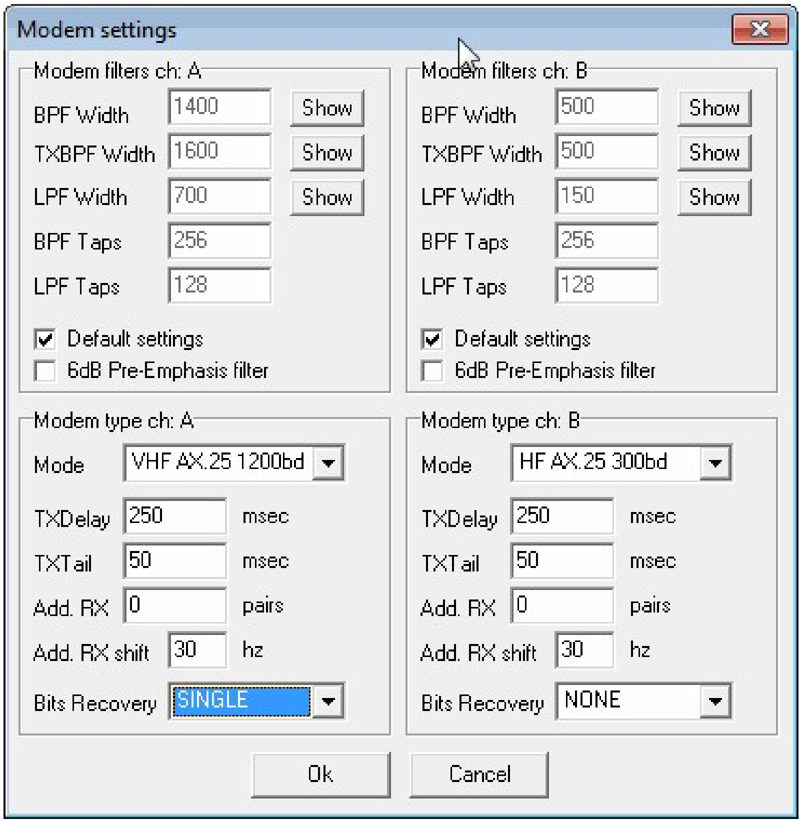
```
rtl_fm -f 144.390M -s 22050 | multimon-ng -t raw -a AFSK1200 -f alpha -A /dev/stdin
```

See the multimonNG tutorial section for more information about multimonNG.

APRSISCE/32

APRSISCE/32 is an advanced APRS program that is capable of iGate functionality and plotting received APRS GPS coordinates on a map. APRSISCE/32 can be downloaded from the downloads section at http://aprsisce.wikidot.com/. To get APRS working with APRSISCE/32 you will need to also download the UH7HO soundmodem software from http://uz7.ho.ua/packetradio.htm. Download the sm2ch57.zip file and extract it to a folder.

1. Open SDR# or your favourite SDR receiver and set the audio output to your preferred audio piping method. Tune to an APRS frequency.
2. In Soundmodem go to **settings->Modem.**
3. Change the Modem type ch:A to VHF AX.25 1200bd.



Modem settings

Modem filters ch: A

BPF Width	1400	Show
TXBPF Width	1600	Show
LPF Width	700	Show
BPF Taps	256	
LPF Taps	128	

☑ Default settings

□ 6dB Pre-Emphasis filter

Modem filters ch: B

BPF Width	500	Show
TXBPF Width	500	Show
LPF Width	150	Show
BPF Taps	256	
LPF Taps	128	

☑ Default settings

□ 6dB Pre-Emphasis filter

Modem type ch: A

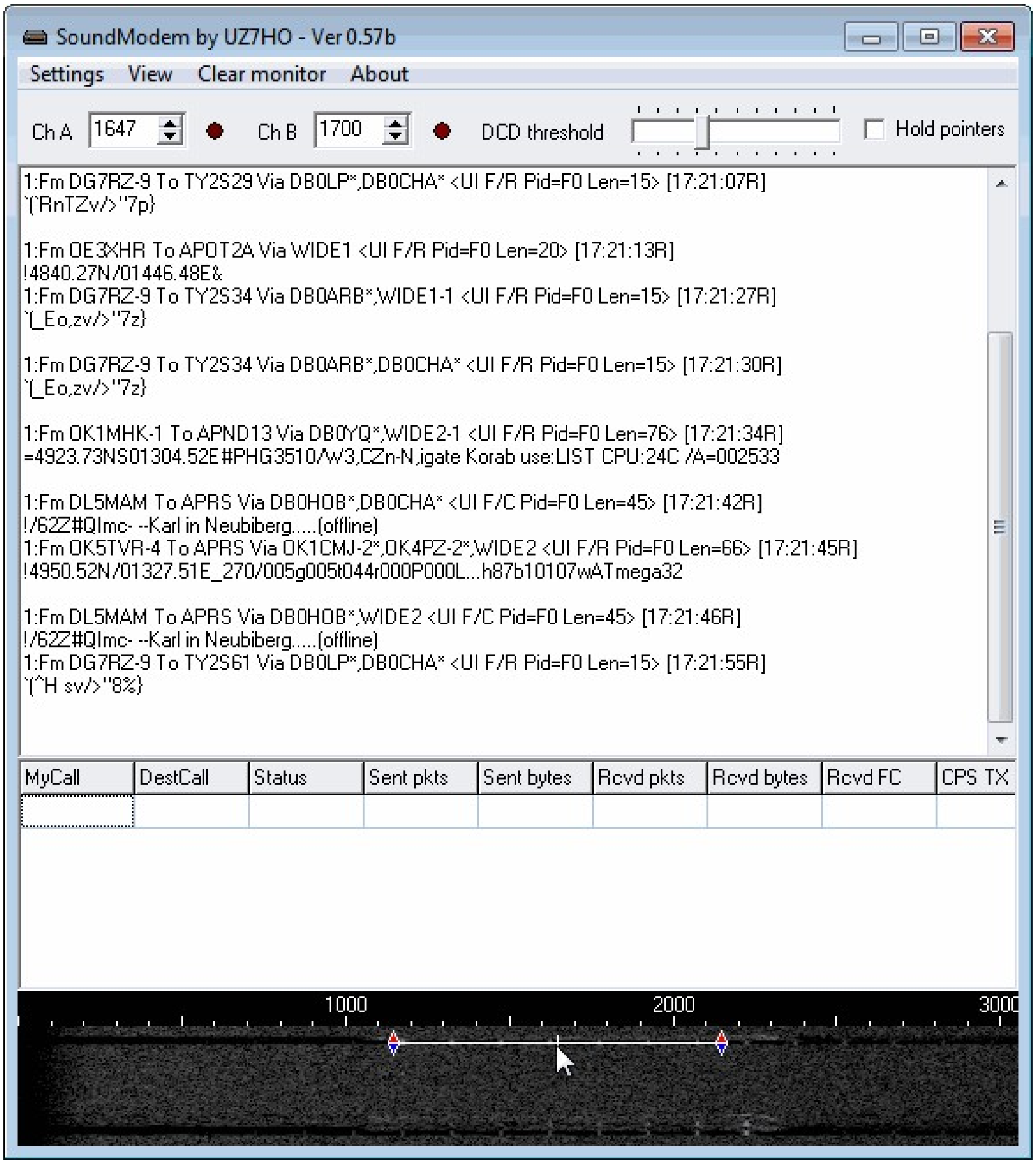
Mode	VHF AX.25 1200bd	
TXDelay	250	msec
TXTail	50	msec
Add. RX	0	pairs
Add. RX shift	30	hz
Bits Recovery	SINGLE	

Modem type ch: B

Mode	HF AX.25 300bd	
TXDelay	250	msec
TXTail	50	msec
Add. RX	0	pairs
Add. RX shift	30	hz
Bits Recovery	NONE	

4. Set the input audio device by going to **Settings->Devices** and the choose your audio piping device next to “Input Device”.

5. Centre the tuning bar at the bottom of the SoundModem window on the APRS signal.



Ch A 1647 Ch B 1700

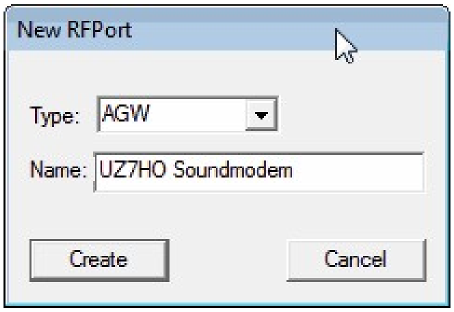
```
1:Fm DG7RZ-9 To TY2S29 Via DB0LP*,DB0CHA* <UI F/R Pid=F0 Len=15> [17:21:07R]
`(`RnTZv/>"7p}

1:Fm OE3XHR To APOT2A Via WIDE1 <UI F/R Pid=F0 Len=20> [17:21:13R]
!4840.27N/01446.48E&
1:Fm DG7RZ-9 To TY2S34 Via DB0ARB*,WIDE1-1 <UI F/R Pid=F0 Len=15> [17:21:27R]
`(_Eo,zv/>"7z}

1:Fm DG7RZ-9 To TY2S34 Via DB0ARB*,DB0CHA* <UI F/R Pid=F0 Len=15> [17:21:30R]
`(_Eo,zv/>"7z}

1:Fm OK1MHK-1 To APND13 Via DB0YQ*,WIDE2-1 <UI F/R Pid=F0 Len=76> [17:21:34R]
=4923.73NS01304.52E#PHG3510/W3,CZn-N,igate Korab use:LIST CPU:24C /A=002533

1:Fm DL5MAM To APRS Via DB0HOB*,DB0CHA* <UI F/C Pid=F0 Len=45> [17:21:42R]
!/62Z#QImc- --Karl in Neubiberg.....(offline)
1:Fm OK5TVR-4 To APRS Via OK1CMJ-2*,OK4PZ-2*,WIDE2 <UI F/R Pid=F0 Len=66> [17:21:45R]
!4950.52N/01327.51E_270/005g005t044r000P000L...h87b10107wATmega32

1:Fm DL5MAM To APRS Via DB0HOB*,WIDE2 <UI F/C Pid=F0 Len=45> [17:21:46R]
!/62Z#QImc- --Karl in Neubiberg.....(offline)
1:Fm DG7RZ-9 To TY2S61 Via DB0LP*,DB0CHA* <UI F/R Pid=F0 Len=15> [17:21:55R]
`(^H sv/>"8%}
```

6. When you open APRSISCE32 for the first time it will ask you for your call sign. If you simply wish to use APRSISCE32 as a map for received APRS packets, type something random into the MyCall-ssid box.

7. If you just want to see which messages you are receiving via your antenna only, you will want to disable ARPS-IS in order to stop APRS messages being received via the internet. Go to **Enables->ARPS-IS Enabled** and uncheck this setting.

8. In APRSISCE32 go to **Configure->Ports->New Port.**

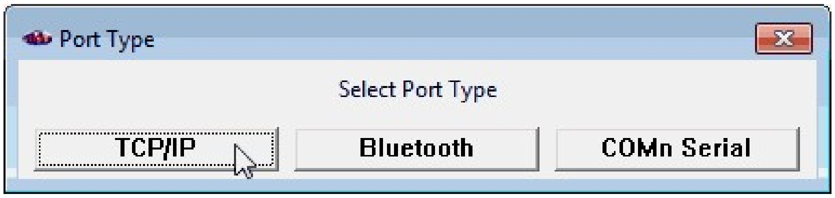
9. Change the type to AGW, set the name as UZ7HO soundmodem.


New RFPort

Type: AGW

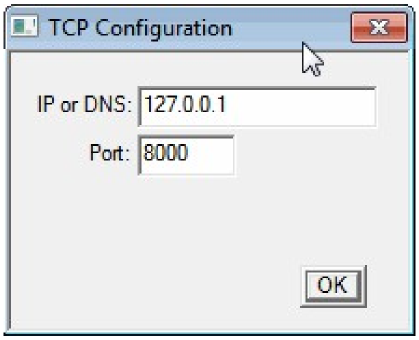
Name: UZ7HO Soundmodem

10. For port type choose TCP/IP.


Port Type

Select Port Type

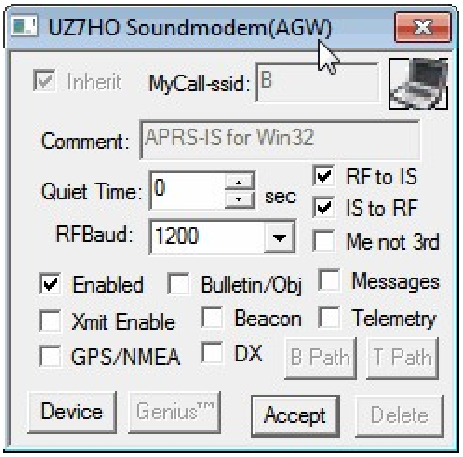
11. Set the IP to 127.0.0.1 and the port to 8000.


TCP Configuration

IP or DNS: 127.0.0.1

Port: 8000

12. Uncheck RF to IS and IS to RF, unless you are a ham who understands and wants to use these options.


UZ7HO Soundmodem(AGW)

☑ Inherit MyCall-ssid: B

Comment: APRS-IS for Win32

Quiet Time: 0 sec

RFBaud: 1200

☑ RF to IS

☑ IS to RF

□ Me not 3rd

☑ Enabled □ Bulletin/Obj □ Messages

□ Xmit Enable □ Beacon □ Telemetry

□ GPS/NMEA □ DX

13. Go to **View->ALL** and make sure that this is checked.
14. APRSISCE32 will now be listening to the UZ7HO soundmodem.

In APRSISCE32 you can move around with the mouse and zoom in and out with the mouse wheel. There is also a zoom bar on the left of the map.

Note that it is also possible to use the AGWPE packet engine software that is discussed below to send APRS data to APRSISCE/32 using the same setup.

AGW PACKET ENGINE AND TRACKER

Another good set of programs for decoding and displaying APRS positions on a map is the AGW Packet Engine and AGW Tracker. The packet engine is what decodes the APRS packets received by the RTL-SDR and AGW tracker is what displays them on a map.

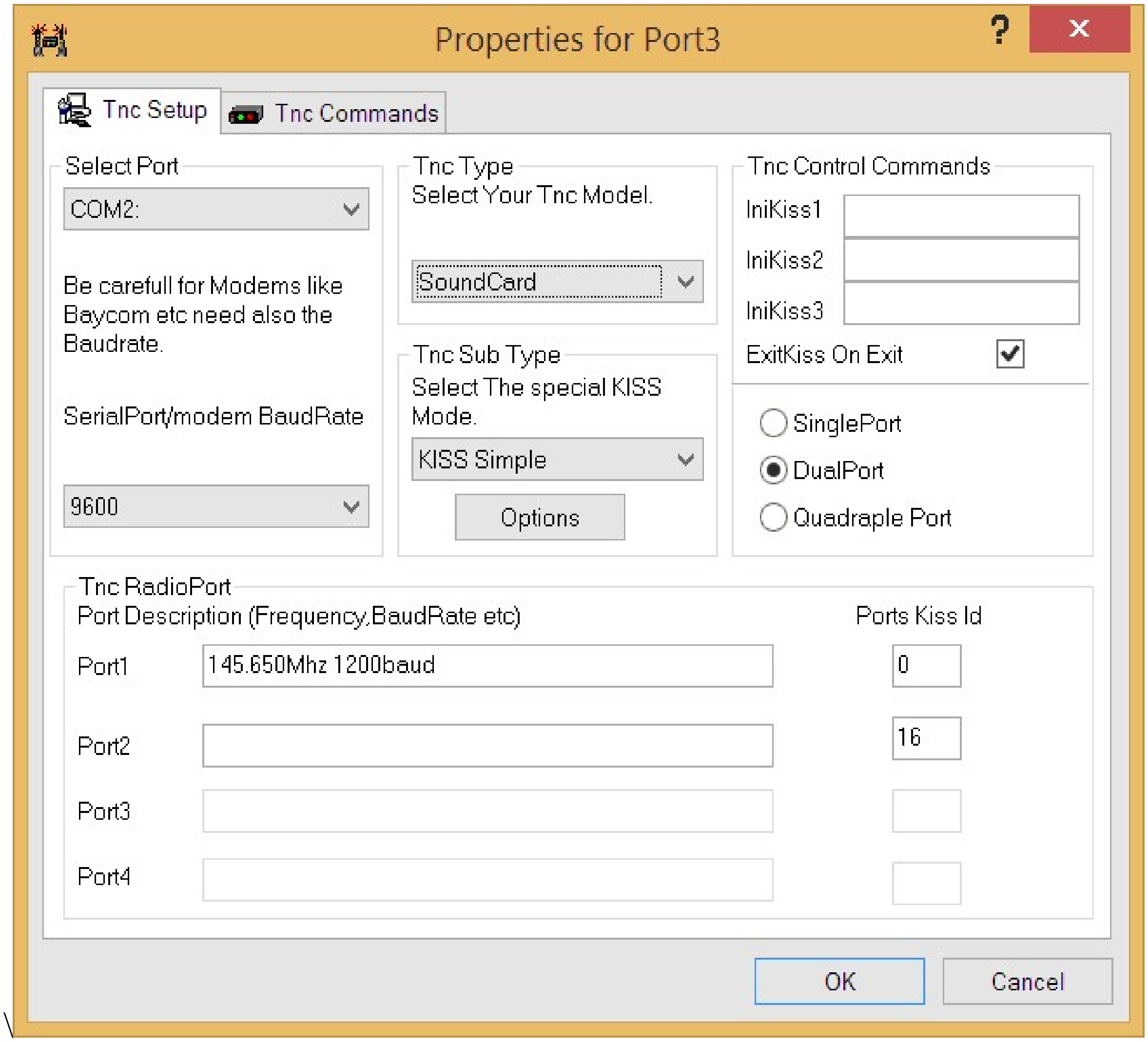
Download AGW Packet Engine and AGW Tracker from http://www.sv2agw.com/downloads/default.htm. AGW Packet Engine comes with a Hamware licence, essentially meaning that it’s free. AGW Tracker is licensed as Shareware with an unlimited trial time which means it is completely free but you can pay for it if you want. To set these programs up follow these instructions.

1. Open SDR# or your favourite SDR receiver and set the audio output to your preferred audio piping method. Tune to an APRS frequency.
2. Open the AGW Packet Engine exe file. Note you may need to open it as an administrator to be able to set the audio pipe.

3. Right click on the AGE Packet Engine icon in the taskbar and go to Properties.

4. Click on New Port.

5. In the new window that pops up choose the Tnc Setup tab, and under Tnc Type select ‘SoundCard’.



Properties for Port3

Tnc Setup | Tnc Commands

Select Port

COM2:

Be carefull for Modems like Baycom etc need also the Baudrate.

SerialPort/modem BaudRate

9600

Tnc Type

Select Your Tnc Model.

SoundCard

Tnc Sub Type

Select The special KISS Mode.

KISS Simple

Options

Tnc Control Commands

IniKiss1

IniKiss2

IniKiss3

ExitKiss On Exit ☑

○ SinglePort

● DualPort

○ Quadraple Port

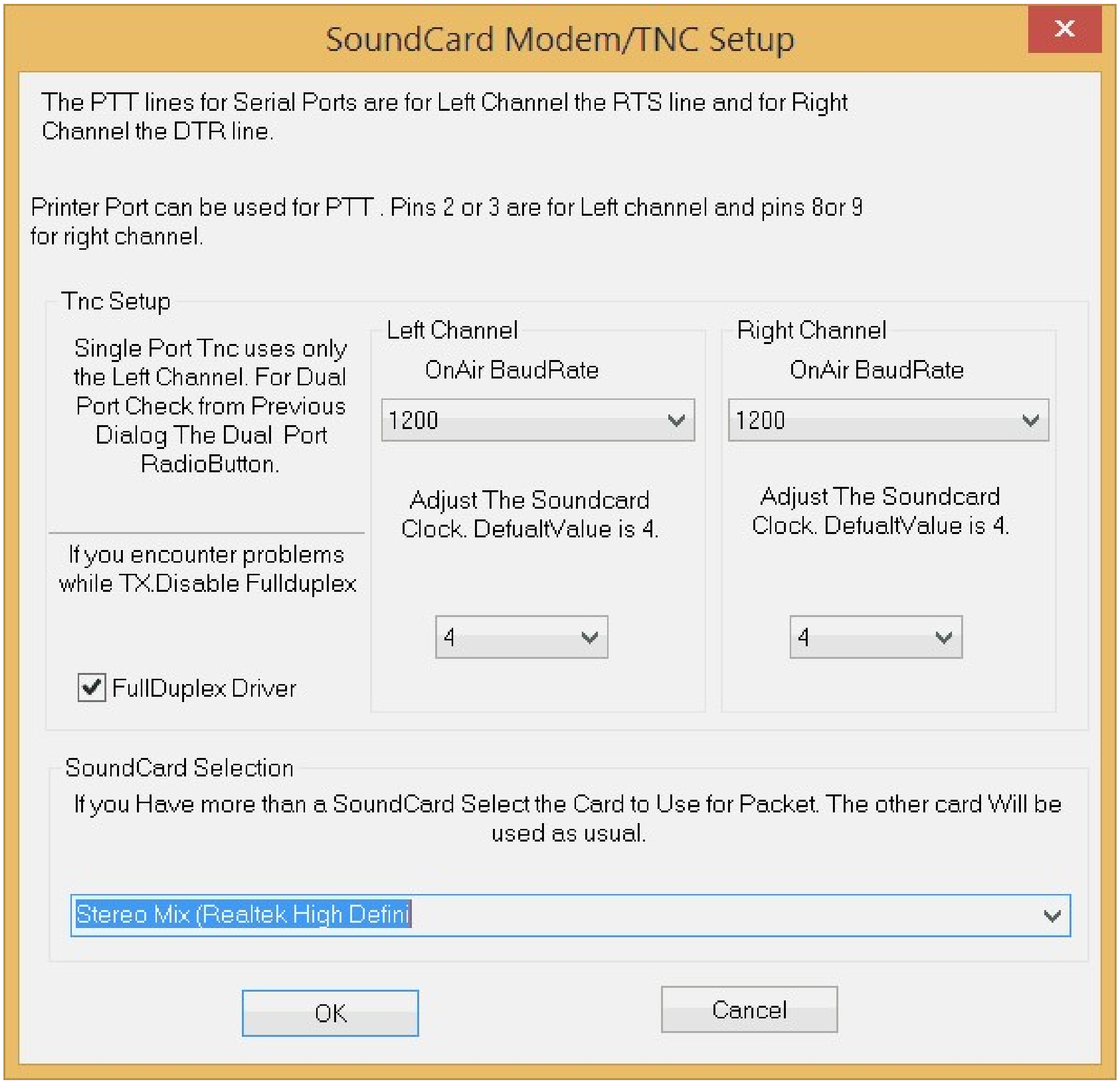
Tnc RadioPort

Port Description (Frequency,BaudRate etc)		Ports Kiss Id
Port1	145.650Mhz 1200baud	0
Port2		16
Port3		
Port4		

OK | Cancel

\

6. Now another window will pop up. In this window under the “SoundCard Selection” heading choose your audio piping method.



SoundCard Modem/TNC Setup

The PTT lines for Serial Ports are for Left Channel the RTS line and for Right Channel the DTR line.

Printer Port can be used for PTT . Pins 2 or 3 are for Left channel and pins 8or 9 for right channel.

Tnc Setup

Single Port Tnc uses only the Left Channel. For Dual Port Check from Previous Dialog The Dual Port RadioButton.

If you encounter problems while TX.Disable Fullduplex

☑ FullDuplex Driver

Left Channel

OnAir BaudRate: 1200

Adjust The Soundcard Clock. DefualtValue is 4.

4

Right Channel

OnAir BaudRate: 1200

Adjust The Soundcard Clock. DefualtValue is 4.

4

SoundCard Selection

If you Have more than a SoundCard Select the Card to Use for Packet. The other card Will be used as usual.

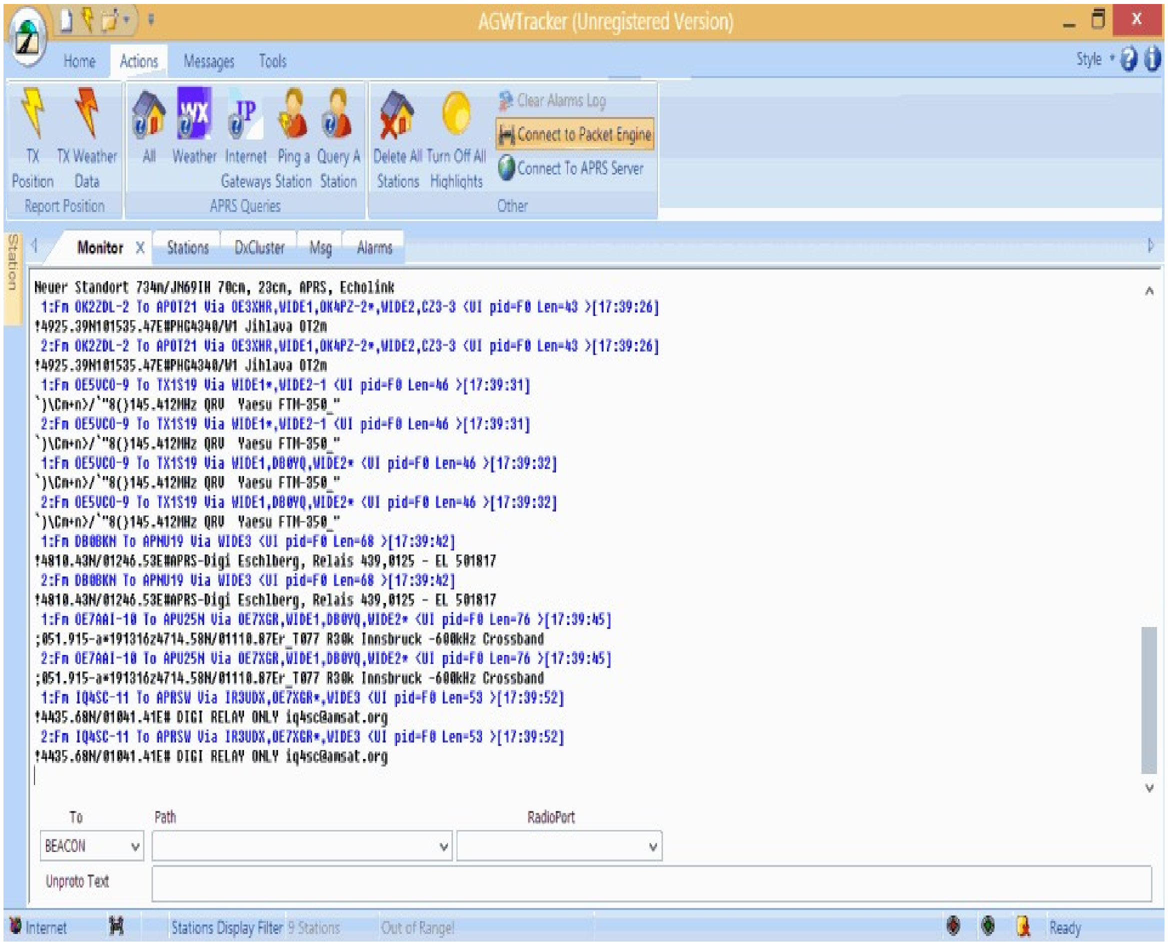
Stereo Mix (Realtek High Defini

7. After making these changes you will need to restart the program.

8. Open AGWETracker. You can ignore the startup options, but if you're a ham who wants to use this as an iGate you might want to set them up.

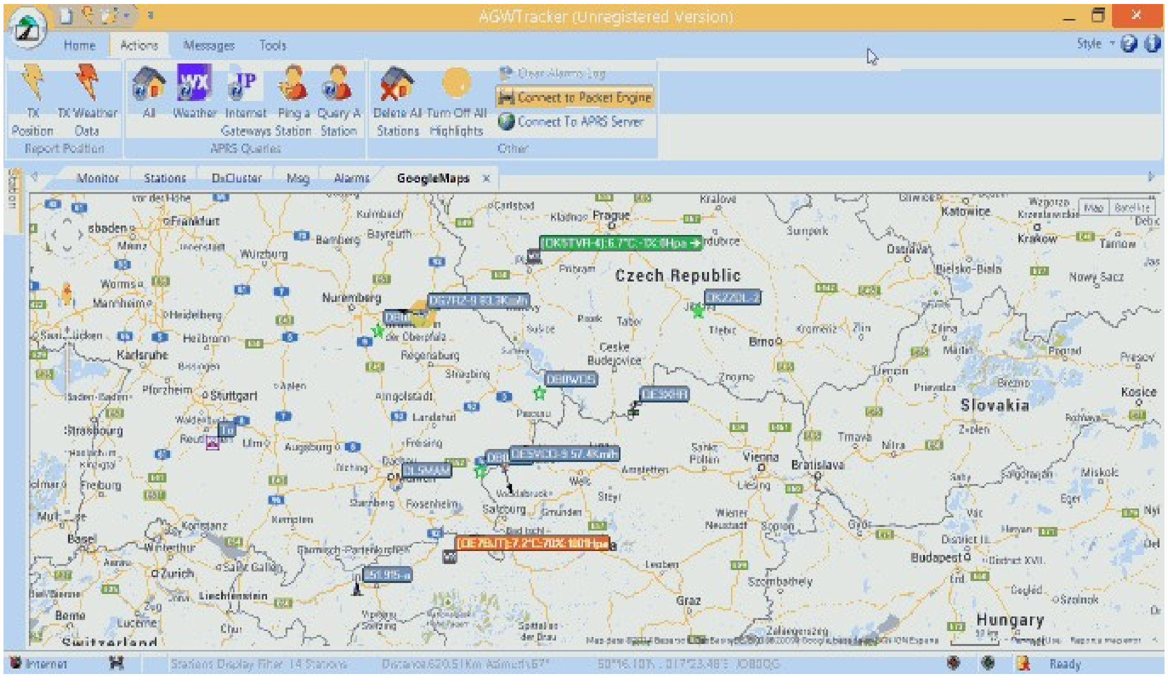
9. Go to the actions tab and ensure that the Connect to packet engine button Connect to Packet Engine is pressed in. Also ensure Connect to APRS Server is *not* pressed in if you don't want to receive APRS packets via the internet.

10. When APRS packets are received by AGW Packet Engine the decoded messages will show up in the monitor tab. Note that you may need to adjust your volume settings so that packets are clearly received.



```
Neuer Standort 734m/JN69IH 70cm, 23cm, APRS, Echolink
 1:Fm OK2ZDL-2 To APOT21 Via OE3XHR,WIDE1,OK4PZ-2*,WIDE2,CZ3-3 <UI pid=F0 Len=43 >[17:39:26]
!4925.39N101535.47E#PHG4340/W1 Jihlava OT2m
 2:Fm OK2ZDL-2 To APOT21 Via OE3XHR,WIDE1,OK4PZ-2*,WIDE2,CZ3-3 <UI pid=F0 Len=43 >[17:39:26]
!4925.39N101535.47E#PHG4340/W1 Jihlava OT2m
 1:Fm OE5VCO-9 To TX1S19 Via WIDE1*,WIDE2-1 <UI pid=F0 Len=46 >[17:39:31]
`)\Cm+n>/`"8()145.412MHz QRV  Yaesu FTM-350_"
 2:Fm OE5VCO-9 To TX1S19 Via WIDE1*,WIDE2-1 <UI pid=F0 Len=46 >[17:39:31]
`)\Cm+n>/`"8()145.412MHz QRV  Yaesu FTM-350_"
 1:Fm OE5VCO-9 To TX1S19 Via WIDE1,DB0YQ,WIDE2* <UI pid=F0 Len=46 >[17:39:32]
`)\Cm+n>/`"8()145.412MHz QRV  Yaesu FTM-350_"
 2:Fm OE5VCO-9 To TX1S19 Via WIDE1,DB0YQ,WIDE2* <UI pid=F0 Len=46 >[17:39:32]
`)\Cm+n>/`"8()145.412MHz QRV  Yaesu FTM-350_"
 1:Fm DB0BKN To APNU19 Via WIDE3 <UI pid=F0 Len=68 >[17:39:42]
!4810.43N/01246.53E#APRS-Digi Eschlberg, Relais 439,0125 - EL 501817
 2:Fm DB0BKN To APNU19 Via WIDE3 <UI pid=F0 Len=68 >[17:39:42]
!4810.43N/01246.53E#APRS-Digi Eschlberg, Relais 439,0125 - EL 501817
 1:Fm OE7AAI-10 To APU25N Via OE7XGR,WIDE1,DB0YQ,WIDE2* <UI pid=F0 Len=76 >[17:39:45]
;051.915-a*191316z4714.58N/01110.87Er_T077 R30k Innsbruck -600kHz Crossband
 2:Fm OE7AAI-10 To APU25N Via OE7XGR,WIDE1,DB0YQ,WIDE2* <UI pid=F0 Len=76 >[17:39:45]
;051.915-a*191316z4714.58N/01110.87Er_T077 R30k Innsbruck -600kHz Crossband
 1:Fm IQ4SC-11 To APRSW Via IR3UDX,OE7XGR*,WIDE3 <UI pid=F0 Len=53 >[17:39:52]
!4435.68N/01041.41E# DIGI RELAY ONLY iq4sc@amsat.org
 2:Fm IQ4SC-11 To APRSW Via IR3UDX,OE7XGR*,WIDE3 <UI pid=F0 Len=53 >[17:39:52]
!4435.68N/01041.41E# DIGI RELAY ONLY iq4sc@amsat.org
```

11. To view received stations on a map, go to the Stations Tab, then right click on a station and select locate and show on map. In the window that pops up choose your preferred map (Google, Bing or Yahoo! Maps). You can then zoom out to see all the other stations.



OTHER APRS SOFTWARE

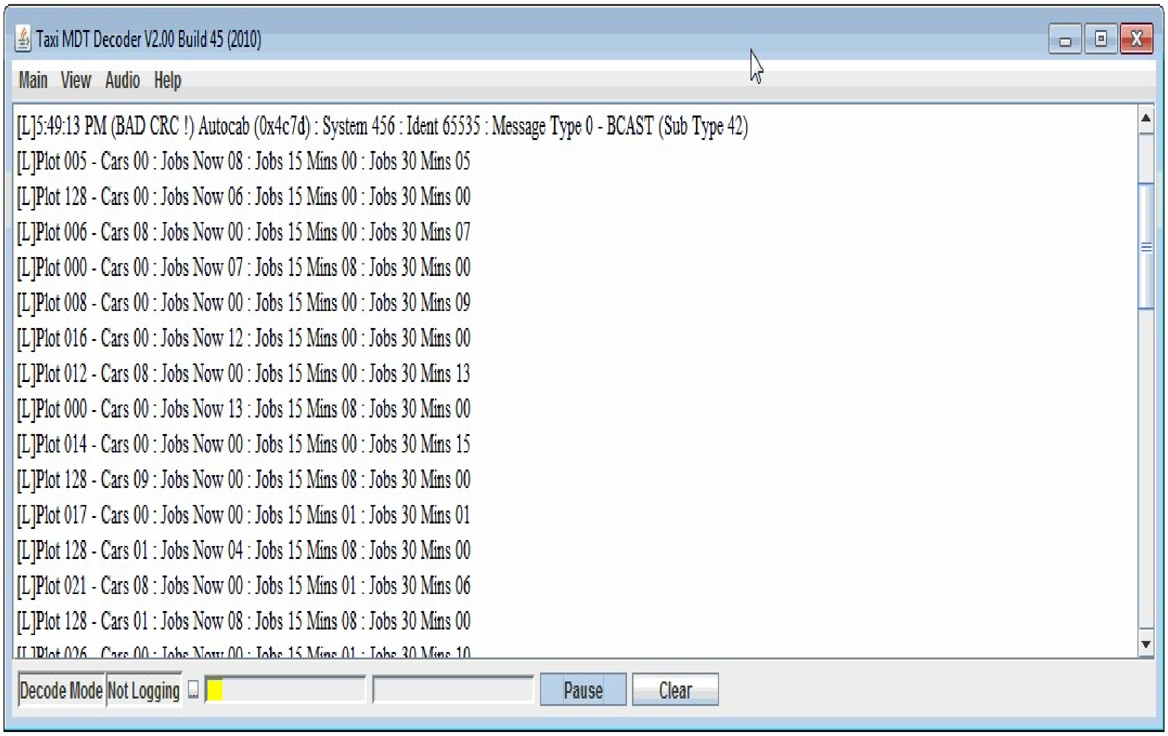
On Linux there is Xastir http://xastir.org/ which can be used with the RTL-SDR. Using Xastir with the RTL-SDR is not as easy as simply piping the output of MultimonNG into Xastir. A script to feed the APRS data to Xastir is required. Fortunately, KJ6VVZ has written a good tutorial on this which is available at http://kj6vvz.com/xastir-with-rtl-sdr.html.

For amateur radio hobbyists, the RTL-SDR can also be set up as an RX iGate by using the command line python software pymultimonaprs which can be downloaded from https://github.com/asdil12/pymultimonaprs. A good tutorial showing how to do this (in Polish but can be Google Translated) can be found here http://sq7mru.blogspot.com/2013/08/aprs-igate-rx-z-tunera-dvb-t.html.

MOBILE DATA TERMINAL (MDT) DECODING GUIDE

MDT's are Mobile Data Terminals and are often used by Ambulances and Taxi's to display real time information about jobs, which the drivers can then accept notifying the dispatchers. There is a Java based Taxi MDT Decoder that can be downloaded from http://sourceforge.net/projects/taxidecoder/. It is capable of decoding Auriga and Autocab MDT's which are commonly used in the UK.

To use taxi decoder simply tune to a MDT radio signal in your favourite SDR receiver software and pipe the audio into taxi decoder. To choose the audio pipe in Taxi Decoder Go to **Audio->Audio Devices**. Adjust the volume settings in your SDR receiver and in Windows so that the audio level shows a green bar at the bottom of the window when a signal is playing.



```
[L]5:49:13 PM (BAD CRC !) Autocab (0x4c7d) : System 456 : Ident 65535 : Message Type 0 - BCAST (Sub Type 42)
[L]Plot 005 - Cars 00 : Jobs Now 08 : Jobs 15 Mins 00 : Jobs 30 Mins 05
[L]Plot 128 - Cars 00 : Jobs Now 06 : Jobs 15 Mins 00 : Jobs 30 Mins 00
[L]Plot 006 - Cars 08 : Jobs Now 00 : Jobs 15 Mins 00 : Jobs 30 Mins 07
[L]Plot 000 - Cars 00 : Jobs Now 07 : Jobs 15 Mins 08 : Jobs 30 Mins 00
[L]Plot 008 - Cars 00 : Jobs Now 00 : Jobs 15 Mins 00 : Jobs 30 Mins 09
[L]Plot 016 - Cars 00 : Jobs Now 12 : Jobs 15 Mins 00 : Jobs 30 Mins 00
[L]Plot 012 - Cars 08 : Jobs Now 00 : Jobs 15 Mins 00 : Jobs 30 Mins 13
[L]Plot 000 - Cars 00 : Jobs Now 13 : Jobs 15 Mins 08 : Jobs 30 Mins 00
[L]Plot 014 - Cars 00 : Jobs Now 00 : Jobs 15 Mins 00 : Jobs 30 Mins 15
[L]Plot 128 - Cars 09 : Jobs Now 00 : Jobs 15 Mins 08 : Jobs 30 Mins 00
[L]Plot 017 - Cars 00 : Jobs Now 00 : Jobs 15 Mins 01 : Jobs 30 Mins 01
[L]Plot 128 - Cars 01 : Jobs Now 04 : Jobs 15 Mins 08 : Jobs 30 Mins 00
[L]Plot 021 - Cars 08 : Jobs Now 00 : Jobs 15 Mins 01 : Jobs 30 Mins 06
[L]Plot 128 - Cars 01 : Jobs Now 08 : Jobs 15 Mins 08 : Jobs 30 Mins 00
[L]Plot 026 - Cars 00 : Jobs Now 00 : Jobs 15 Mins 01 : Jobs 30 Mins 10
```

MORE INFORMATION AND REFERENCES:

http://www.rtl-sdr.com/decoding-taxi-mobile-data-terminal-signals-with-rtl-sdr/

USING MULTIMONNG ON LINUX AND WINDOWS

MultimonNG is a command line program available on both Windows and Linux and can be run on an embedded computer like the Raspberry Pi. It is capable of decoding a wide range of protocols such as the following.

```
POCSAG512 POCSAG1200 POCSAG2400
EAS
UFSK1200 CLIPFSK AFSK1200 AFSK2400 AFSK2400_2 AFSK2400_3
HAPN4800
FSK9600
DTMF
ZVEI1 ZVEI2 ZVEI3 DZVEI PZVEI
EEA EIA CCIR
MORSE CW
```

MultimonNG can be obtained from https://github.com/EliasOenal/multimon-ng/. The Linux compile instructions can also be found at the above link. To run MultimonNG on Windows download a precompiled version here: https://dl.dropboxusercontent.com/u/43061070/multimonNG.exe (Mirror: http://bit.ly/1iNvmQf).

To run MultimonNG on Windows first download and extract the zip file to a folder on your computer. Then follow these instructions.

1. Open your favourite SDR receiver such as SDR#.
2. Set your audio output to your favourite audio pipe. Note that on Windows multimonng will by default listen to your default audio piping method set in Windows sound recording properties.
3. Open a Windows command prompt by going to **Start->All Programs>Accessories->Command Prompt**.
4. In command prompt change directory using the ‘cd’ command to where MultimonNG is installed. e.g. cd c:/tools/multimonng
5. You can now run multimon-ng using the command multimonng -a XXX where XXX should be replaced with the protocol you’d like to decode. You can specify multiple decoders if you wish.

On Linux (or Windows if desired) you will need to pipe in FM audio with a sample rate of 22050 into multimonNG. You can use rtl_fm to do this. See the example below.

```
rtl_fm -f 158.1M -s 22050 -p 49 | multimon-ng -t raw -a POCSAG512 -a POCSAG1200 -a POCSAG2400 -f
alpha /dev/stdin
```

Or alternatively if you want to use a different bandwidth / sample rate, either use the -r flag in rtl_fm, or use sox for resampling and use any sample rate desired.

```
rtl_fm -M fm -f 157.918M -s 12k -g 30 -l 0 -p 49 | sox -traw -r12k -es -b16 -c1 -V1 - -traw -es -b16 -c1 -r22k
- | multimon-ng -t raw -a POCSAG512 -a POCSAG1200 -a POCSAG2400 -f alpha /dev/stdin
```

USING RTL_TCP

Rtl_tcp is a command line program that works on both Windows and Linux. It is used to send an IQ stream from a remote RTL-SDR server over TCP/IP. In other words it allows you to connect to a remote RTL-SDR dongle over a network. SDR# is one such program that can connect to a rtl_tcp server.

In most cases you will need a fast network connection to use rtl_tcp as it can use a significant amount of network bandwidth. To calculate the bandwidth required we need to multiply the chosen SDR sample rate by the number of bits per sample. The number of bits in the raw RTL-SDR IQ stream is 8 bits for the I stream and 8 bits for the Q stream. So we can use the following calculation.

Bandwidth (MBit/s) = Sample Rate (Ms/s) x (8+8)bits

So for example, running at the maximum sample rate of 3.2 Msps we would need 3.2 x 16 = 51.2 MBits/s. For a more conservative 2 Msps we need 2 x 16 = 32Mbit/s.

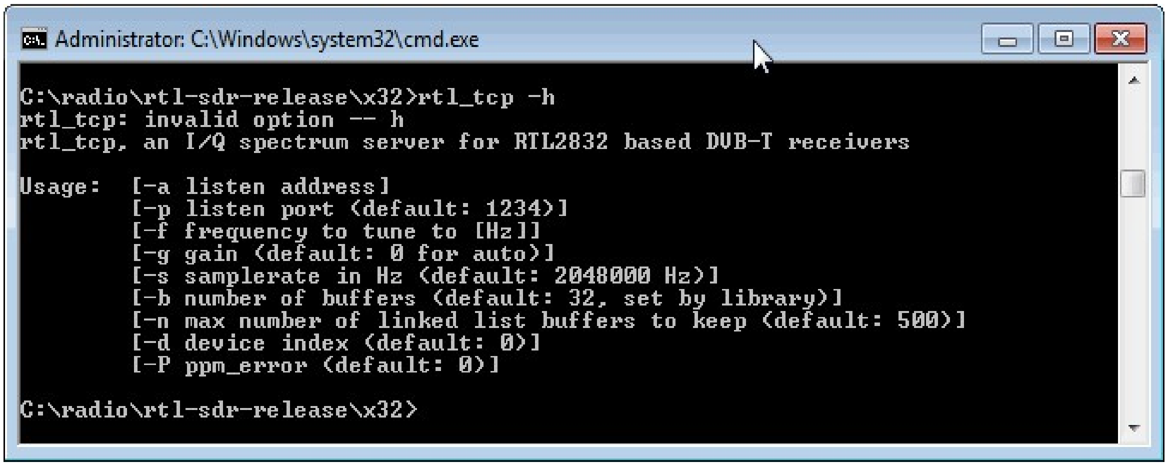
Standard ethernet networks run at 100 Mbits/s so they are sufficient (assuming the network is not heavily used by other applications and overheads are taken into account). For comparison USB 2.0 runs at 480 Mbits/s. Wireless 802.11a and 802.11g networks run at up to 54Mbits/s, however this is the theoretical maximum and they never usually run this fast. The newer 802.11n wireless standard can run at up to 300 Mbits/s and should be sufficient for the maximum sample rate.

Note that other factors like network congestion, ethernet cabling length, WiFi connection quality, number of WiFi users and embedded device speed might affect your performance. To test your network speed on Windows try the LAN Speed test software http://www.totusoft.com/downloads.html. Realistically, you should expect to use a lower sampling rate such as 1Msps, especially if you are using an embedded device like the Raspberry Pi. You should expect WiFi to be problematic if there are any other users or if there is significant interference.

To use rtl_tcp on Windows follow the instructions below.

1. Download the rtl_sdr precompiled binaries for Windows from http://sdr.osmocom.org/trac/attachment/wiki/rtl-sdr/RelWithDebInfo.zip.
2. Extract the RelWithDebInfo.zip file to a folder.

3. Using the Windows command prompt navigate to the folder you have extracted the zip file to.
4. Navigate to the x32 folder.
5. Run rtl_tcp -h for a list of available commands.



```
C:\radio\rtl-sdr-release\x32>rtl_tcp -h
rtl_tcp: invalid option -- h
rtl_tcp, an I/Q spectrum server for RTL2832 based DVB-T receivers

Usage:  [-a listen address]
        [-p listen port (default: 1234)]
        [-f frequency to tune to [Hz]]
        [-g gain (default: 0 for auto)]
        [-s samplerate in Hz (default: 2048000 Hz)]
        [-b number of buffers (default: 32, set by library)]
        [-n max number of linked list buffers to keep (default: 500)]
        [-d device index (default: 0)]
        [-P ppm_error (default: 0)]

C:\radio\rtl-sdr-release\x32>
```

To use SDR# with rtl_tcp:

1. Find your local ip address by typing ipconfig at a command prompt. Your ip address will be listed under a ethernet apadter: local area network connection heading.
2. At the command prompt use the following where IP_ADDRESS should be the IP address of the computer you are running rtl_tcp on.

```
rtl_tcp -a IP_ADDRESS
```

3. In SDR#, select RTL-SDR/TCP and in the configure menu enter the IP Address of the server. The sample rate and gain controls can be adjusted from SDR#.
4. Press Play.

To use SDR Touch with rtl_tcp:

1. At the command prompt use the following (where -s sets the sample rate). Note that the sample rate for SDR Touch must be set as 1024000.

```
rtl_tcp -a IP_ADDRESS -s 1024000
```

2. Open SDR Touch.
3. Scroll down on the menu to the Remote button. Touch this button.
4. Enter the IP address and port as ip_address:port and then touch connect.

Rtl_tcp is automatically installed during the RTL-SDR Linux driver installation procedure and it can be used in the same way as on Windows on Linux. (Note that in Linux the command to find your ipaddress is “ifconfig”). There appears to be a bug in newer Linux kernels that can cause stuttering problems with rtl_tcp due to power saving features. This can be fixed by reverting to an older kernel or by running a process that will use 100% CPU time such as running “top -d .01”.

STREAMING MP3 FM WITH LINUX: RTL_FM

If you just want to stream a FM audio station over the network, you can use rtl_fm and pipe the output into sox to convert it into a mp3 file and then use socat to output it to the network. On the receiving end, netcat can be used to connect to the TCP/IP stream.

The bandwidth required for this is far less than with rtl_tcp as it only needs about 8kb/s at most which almost any network including the internet and WiFi can easily handle.

On Ubuntu sox, socat and netcat can be installed with the following.

```
sudo apt-get install sox
sudo apt-get install socat
sudo apt-get install netcat
```

The commands to encode to mp3 and to play the received audio come from sox, so you will need to install sox on both the server and listener computers. You may also need to install the sox mp3 libraries on the listener side with `sudo apt-get install libsox-fmt-mp3` if you hear no sound output.

You can test if sox is working on a signal with a command that will play the mp3 compressed audio from the RTL-SDR to your speakers. If you hear the audio with this command then the mp3 compression is working.

```
rtl_fm -M fm -f 161.649M -s 12k -g 50 -l 70 | sox -traw -r12k -es -b16 -c1 -V1 - -tmp3 - | play -t mp3 -
```

To send the audio over TCP/IP use the following socat command.

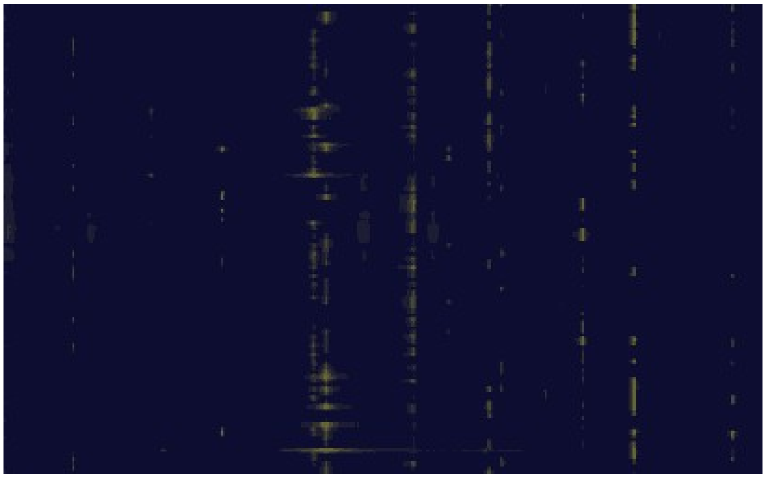
```
rtl_fm -M fm -f 161.649M -s 12k -g 50 -l 70 | sox -traw -r12k -es -b16 -c1 -V1 - -tmp3 - | socat -u - tcp-listen:8080
```

On the listener side use the following command.

```
netcat IP_ADDRESS 8080 | play -t mp3 -
```

Where you should change the IP_ADDRESS to the ip address of the server. You should also change the frequencies, bandwidths and other appropriate rtl_fm flags to those required by the signal you are listening to.

HEAT MAP BAND SCAN



A heatmap can help you discover where in the frequency spectrum signals are being received over a long period of time. To do this, the Linux and Windows software program rtl_power which is included with the official RTL-SDR package (Windows: http://sdr.osmocom.org/trac/attachment/wiki/rtl-sdr/RelWithDebInfo.zip) can be used. However, note that we highly recommend that you use Keenerds updated version of rtl_power available from http://igg.kmkeen.com/builds/. Keenerds drivers and rtl_power version completes scans significantly faster than the original drivers. This tutorial will be based on Windows, but it can easily be adapted to work on Linux as running rtl_power in Linux is much easier.

To create a heatmap we need to use a tool called rtl_power. This tool logs average frequency power over a wide bandwidth using binning. To explain binning we use an example. Let's say we are logging 1 MHz of bandwidth and choose to use a bin size of 2 kHz. Then each pixel in the heatmap image would represent the average power over the 2 kHz bin size. Therefore, the heatmap image would be 500 pixels wide.

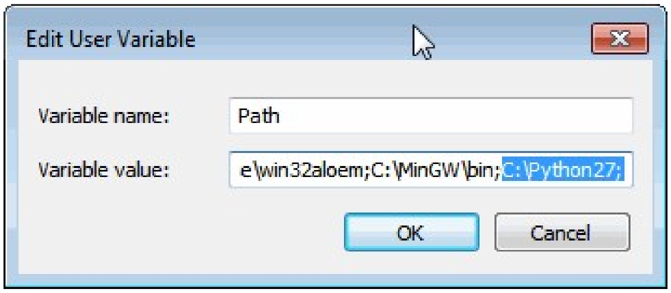
To create a heatmap in Windows follow the instructions below. The instructions can also be adapted for use in Linux. In Linux rtl_power is installed as part of the RTL-SDR driver installation procedure.

1. Download the official RTL-SDR release for Windows from http://sdr.osmocom.org/trac/attachment/wiki/rtl-sdr/RelWithDebInfo.zip and unzip the files into a folder.

2. Download the latest version of heatmap.py from https://github.com/keenerd/rtl-sdr-misc/blob/master/heatmap/heatmap.py (Older Mirror: http://kmkeen.com/tmp/heatmap.py.txt), and save it to a text file with a .py extension in the same folder as rtl_power in the x32 folder.

3. Install Python 2.7 x32 from http://www.python.org/download/. You must be careful to download the older 2.7 version and also ensure that it is the x32 version as well. This is because the PIL (Python Imaging Library) needs this version. Also take note of the path that you install Python to.

4. In Windows right click on Computer in your start menu and go to Properties. Select Advanced System Settings and then click on Environment Variables. In the top window where it says User Environment Variables locate the "Path" variable. Add the path to your newly installed Python directory.



Edit User Variable

Variable name: Path

Variable value: e\win32aloem;C:\MinGW\bin;C:\Python27;

5. Install Python Image Libraries from http://www.pythonware.com/products/pil/. Make sure you download the Python Imaging Library 1.1.7 for Python 2.7 for Windows.

6. Using the Windows command prompt navigate into the directory where you have extracted the official RTL-SDR release and navigate to the x32 folder.

7. Run rtl_power. As an example here is a line that will log 50 MHz of bandwidth with 2000 Hz sized bins over the frequencies 150 MHz to 200 MHz.

```
rtl_power -f 150M:200M:2k -g 20.7 logfile.csv
```

8. When you want to finish running rtl_power, press CTRL+C to quit.

9. To generate the heatmap type the following.

```
python heatmap.py logfile.csv image.png
```

Note that there appears to be a bug with current rtl_power versions which causes a value of -1.#J to be sometimes written to the CSV file. If you get an error related to this when running heatmap.py, simply open the CSV file in a text editing program and do a find and replace of -1.#J and replace it with -1.00.

RTL_POWER FLAGS

More flags can be found by using `rtl_power -h` at the command line, but here we list an explain the most important flags.

-f lower:upper:bin_size Specifies the frequency range to log over. bin_size specifies the width in Hz that each pixel will represent.

-i integration_interval The time that the average power for each bin will be summed over.

-g tuner gain Gain of the RTL-SDR.

-c crop percentage The centre on a chunk of scanned spectrum is higher quality than the edges. This flag specifies how much of the edges should be discarded. Larger values give better results, but slows scanning down. Specify as an decimal (0.2) or percentage (20%). Recommended values are between 0.2 and 0.5.

-F downsampler filters A value from 0 - 9. Larger gives better results. Configures what downsampling filter to use. Higher values use more CPU.

-e exit_timer The length of time that rtl_power should run.

-1 single shot mode Perform one run and then exit.

RTL_POWER GUI

There is a Windows GUI available for rtl_power called rtl_pan. It is available from http://sourceforge.net/projects/guiforrtlpower/. It is capable of generating a heatmap like heatmap.py.

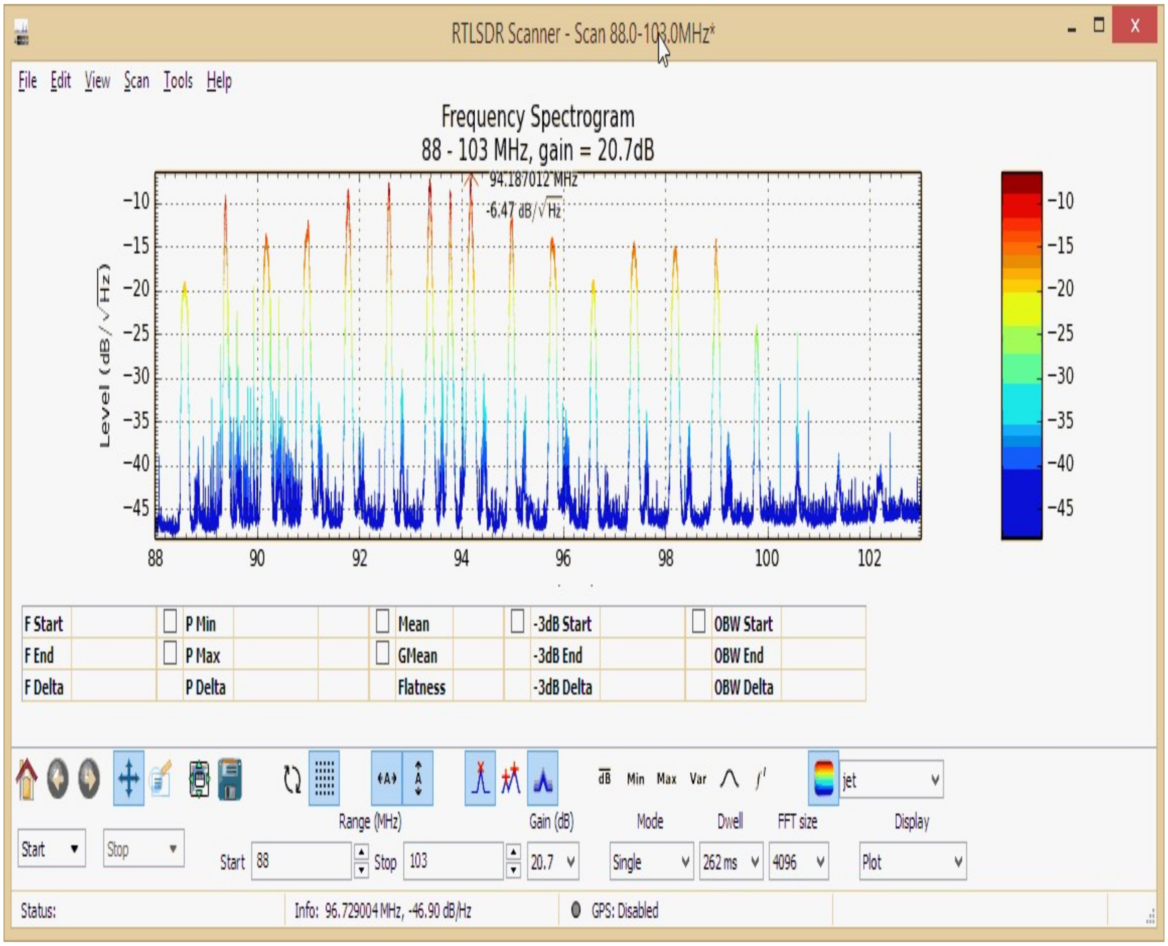
To use rtl_pan, download the latest rtlpan.exe file from the above link and place it in the same folder as rtlsdr.dll, libusb-1.0.dll and rtl_power.exe. Then double click rtlpan.exe to start the program. After starting the program you can select your desired scanning frequency range, bin size, tuner gain and ppm offset. Running the software over time will produce the heatmap which can be saved to an image by clicking the photo icon in the top right corner. At the time of writing this tutorial there is currently no options to select more advanced options like crop size and downsample filters.

RTLSDR SCANNER

RTLSDR scanner is a Windows tool that can generate a FFT plot over a wide bandwidth. It can be downloaded from http://eartoearoak.com/software/rtlsdr-scanner. There is a Windows installer available that will automatically download and install the various dependencies that are required for the software to run. When you get to the "Choose components" part of the installer, make sure that all the dependencies are selected for installation if you are unsure what your PC already has installed.

After installing RTLSDR Scanner, simply enter a range in the Start and Stop boxes at the bottom of the window and set the desired gain. Press Start to begin scanning. The mode can be set to either continuous or single. In continuous mode the scanner will continuously refresh the spectrum every time it finishes one pass, in single mode the scanner will performance one scan and then stop. The dwell time indicates how long the scanner will average over a range before recording the power values. Setting it larger will help clean up noisy graphs, but will slow down scanning. The FFT size indicates the FFT resolution used during scanning. Higher values will give higher resolution graphs which show the structure of signals more clearly, but will cause scanning to be slower. RTLSDR Scanner can also display the band scan as a spectrogram or 3D spectrograph by changing the settings in the Display pull down box.

The latest version of RTLSDR Scanner can also connect to an external GPS receiver and be used to create radio strength heatmaps by driving around with the RTL-SDR and GPS logging data. This can be useful for radio direction finding, which is the act of trying to locate the physical location of a radio transmitter. See the Radio Direction Finding section below for more information.



F Start		□ P Min		□ Mean		□ -3dB Start		□ OBW Start	
F End		□ P Max		□ GMean		-3dB End		OBW End	
F Delta		P Delta		Flatness		-3dB Delta		OBW Delta	

Start ▾ Stop ▾

jet

Start 88 Stop 103

Gain (dB): 20.7

Mode: Single

Dwell: 262 ms

FFT size: 4096

Display: Plot

Status:

Info: 96.729004 MHz, -46.90 dB/Hz

● GPS: Disabled

RADIO DIRECTION FINDING WITH RTLSDR SCANNER

As mentioned in the last section, RTLSDR Scanner has functionality that allows the creation of radio strength heatmaps which can be used for radio direction finding. In other words this simply means that you can find the location of a radio transmitter by driving around with a GPS enabled laptop running RTLSDR Scanner. As you drive around the signal strength will be correlated against your GPS position allowing a signal strength heatmap to be created.

RTLSDR Scanner is compatible with external GPS receivers that can output NMEA GPS data serially via RS232, USB or Bluetooth. It is also compatible on Linux with GPSd, a Linux GPS daemon.

If your laptop does not have a built in GPS receiver you will need to use an external receiver. The most common GPS receiver that most people already own is an Android or iPhone smartphone. In this tutorial we show how to use a common Android or iPhone smartphone as the GPS receiver on a Windows laptop with bluetooth. The instructions here are specifically for Android, but they can easily be adapted for a jailbroken iPhone by using the TryGPSOut app available on the Cydia store for $1 USD.

1. Go to the Google Play store on your Android phone and download the app titled “Bluetooth GPS Output” by Meowsbox. This is a paid app but the free version will run for about 10 minutes giving you enough time to verify that the app works as intended. The full version costs around $1 USD and can be found on the play store. There are other similar apps for free that you can try, but we were only able to get this particular app to work without trouble.

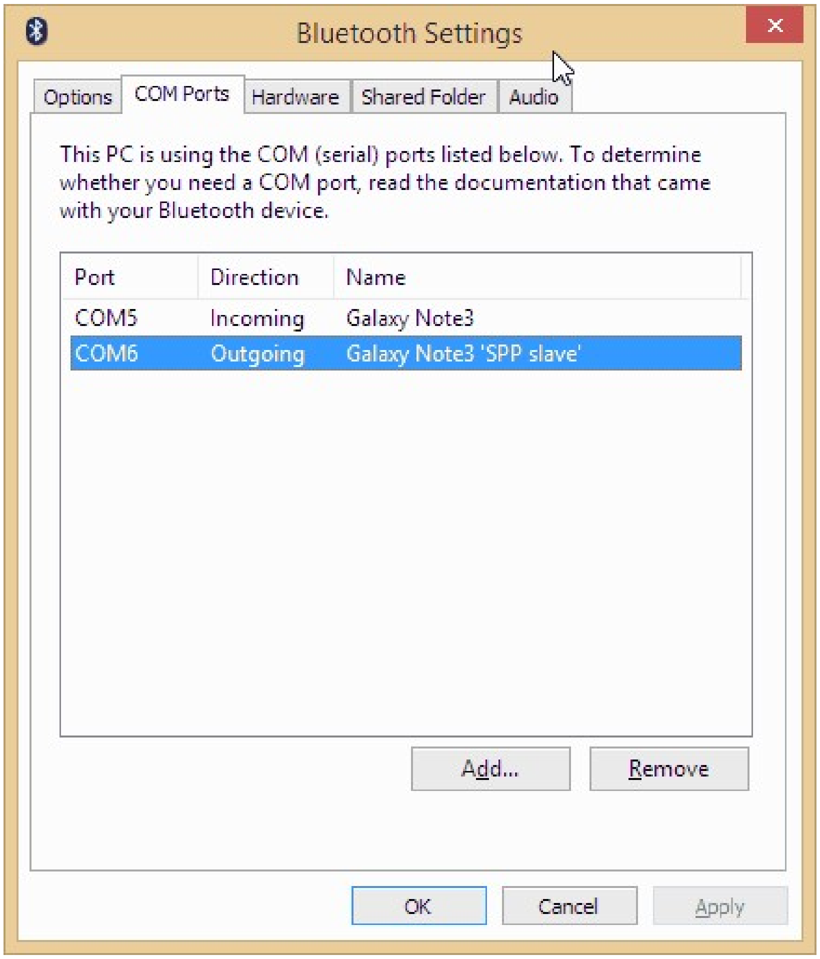
2. Now enable bluetooth on your laptop and phone.

3. Open the Bluetooth GPS app and press the settings button and choose Make Discoverable.

4. Now your phone will become visible to your laptop. Pair the two devices. Note that even though the devices are paired, the app will still show “not connected” in the top right corner. This is normal.

5. In Windows go to Control Panel, click on “Change Bluetooth Settings” which will be under the Devices and Printers Heading.

6. Go to the COM Ports tab. Take note of which COM port has the ‘Outgoing’ direction. In the case of the screenshot shown below it is COM6.

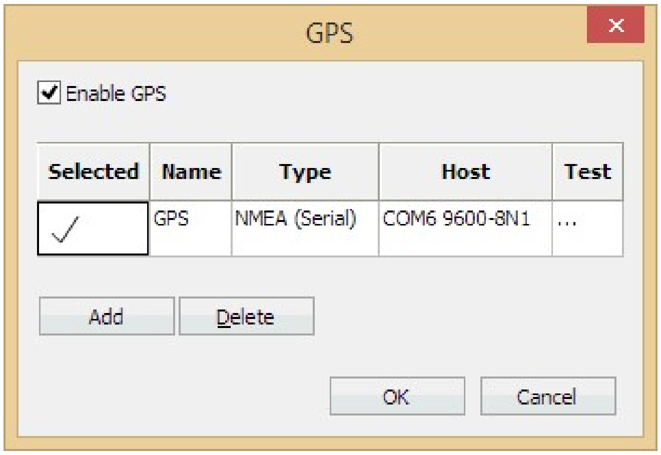


This PC is using the COM (serial) ports listed below. To determine whether you need a COM port, read the documentation that came with your Bluetooth device.

Port	Direction	Name
COM5	Incoming	Galaxy Note3
COM6	Outgoing	Galaxy Note3 'SPP slave'

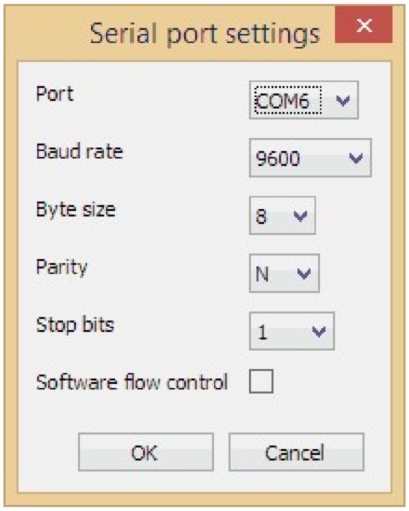
7. Plug in your RTL-SDR dongle into the laptop and open RTLSDR Scanner.

8. In RTLSDR Scanner go to **Edit->GPS**. Check the Enable GPS box and click the Add button. In the added box choose the Type as NMEA (Serial), the Host as the COM port you identified in the previous steps as the ‘Outgoing’ port and under host set the baud rate to 9600 with parity setting left as default.



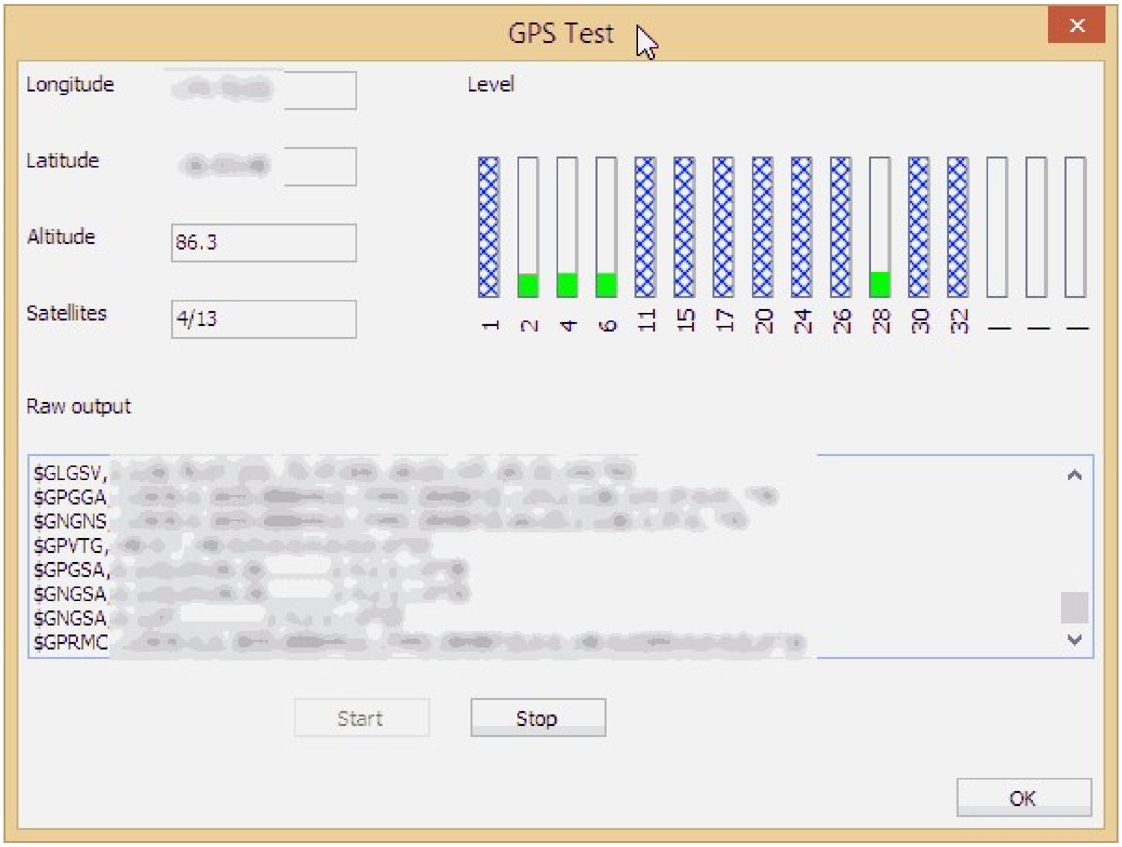
☑ Enable GPS

Selected	Name	Type	Host	Test
✓	GPS	NMEA (Serial)	COM6 9600-8N1	...



Port	COM6
Baud rate	9600
Byte size	8
Parity	N
Stop bits	1
Software flow control	□

9. Click the three dots in the Test column. In the new window that pops up click Start. If everything is working correctly, you should be able to see your GPS coordinates in the top left two boxes and NMEA data should be streaming through in the bottom box.



Longitude [redacted]

Latitude [redacted]

Altitude 86.3

Satellites 4/13

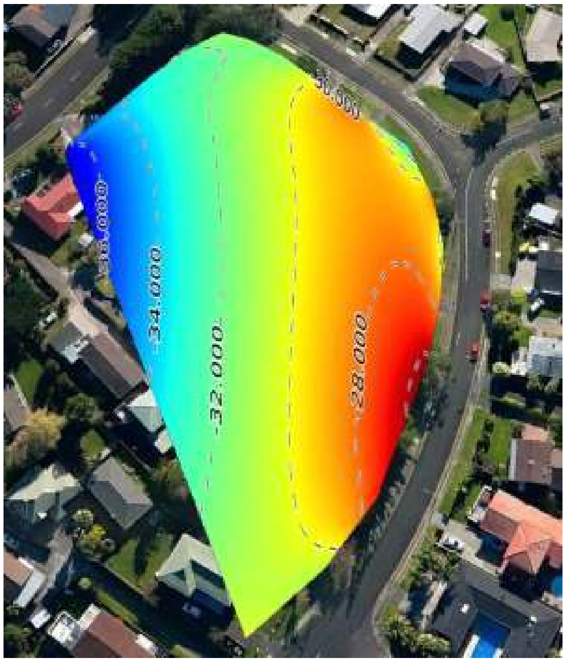
Raw output

```
$GLGSV, [redacted]
$GPGGA [redacted]
$GNGNS [redacted]
$GPVTG, [redacted]
$GPGSA, [redacted]
$GNGSA [redacted]
$GNGSA [redacted]
$GPRMC [redacted]
```

10. Now start a continuous scan in RTLSDR Scanner and drive around. Be sure to set the bandwidth relatively small around the signal you are interested in to get better resolution. You might also try adjusting the dwell and FFT sizes to get faster scans.

11. Once you have finished, press Stop. Then go to **File -> Export Map**. Choose the frequency in kHz you would like to plot on the heatmap and choose the bandwidth in kHz around that frequency that you would like to include in the averaging. For example if you were wanting the heatmap of a 100 kHz wide signal at 88.6 MHz, you would set the centre frequency as 88600 and the bandwidth as 50.

12. Then click export and export the file as a .kmz file. Now you can open this .kmz file in Google Earth to see the heatmap.



DECODING EMERGENCY ALERT SYSTEM (EAS) SAME MESSAGES

In the USA and Canada, and some other coutnries there is an Emergency Alert System (EAS) which is commonly used to alert the public to local weather emergencies such as tornadoes, flash floods and severe thunderstorms. Local EAS weather alerts are encoded with the SAME (Specific Area Message Encoding) protocol. They are transmitted on the local weather radio frequency (NOAA weather radio if in the USA) and some weather radio's are capable of decoding the EAS SAME data. Dsame is an EAS decoder that can be used with the RTL-SDR to receive and output full EAS weather messages. It can be downloaded from https://github.com/cuppa-joe/dsame. An example of a dsame message is as follows:

> The National Weather Service in Pleasant Hill, Missouri has issued a Required Weekly Test valid until 12: 30 PM for the following counties in Kansas: Leavenworth, Wyandotte, Johnson, Miami, and for the following counties in Missouri: Clay, Platte, Jackson, Cass. (KEAX/ NWS)

You can listen to an example EAS SAME message at our post on http://www.rtl-sdr.com/new-eas-same-weather-alert-decoder/.

To decode EAS messages you will need an RTL-SDR, multimon-ng and Python set up and running on your PC (the EAS decoder will run on Windows or Linux). See the section on multimon-ng and the heatmap bandscan section if you need help with setting up Python. To run the program you can either use rtl_fm or SDR# as the input sound source.

To run it with SDR# or another GUI based program in Windows:

1. Download both dsame and multimon-ng, and place them into the same folder. Multimon-ng can also be downloaded from the dsame Github page.

2. In the Windows sound settings set your preferred audio piping method to be the default device. (multimon-ng will automatically listen to the default device).

3. Open up SDR# or a similar program. Set the audio output to go to your audio piping method in the SDR# audio settings and tune to a weather radio station in NFM mode.

4. Open up a Command prompt and use the "cd" command to browse to the folder where you have saved dsame and multimon-ng.

5. Use the following command to begin listening and decoding:

```
dsame.py --source scritps\ source-soundcard-multimon-ng.bat
```

6. Now when an EAS message plays it will show in the terminal.

To decode using rtl_fm on Windows or Linux (the instructions below assume Windows, but if in Linux simply use the .sh script instead of the .bat script):

1. Copy rtl_fm and it’s supporting files to the same directory as multimon-ng and dsame (if in Windows).
2. Use a text editor to open the text file “source-rtl-fm-multimon-ng.bat” which is stored in the scripts folder. Set the PPM and FREQ values appropriately for your RTL-SDR dongle and EAS station frequency.
3. Open up a Command prompt (or terminal in Linux) and browse to the folder where you have saved dsame.
4. Use the following command to begin listening and decoding

```
dsame.py --source scritps\ source-rtl-fm-multimon-ng.bat
```

You can also set dsame to only listen to messages meant for your area by entering the correct SAME code for your location with the --same flag. See the Github page for a full list of commands.

TRAIN TELEMETRY DATA DECODING

Freq (MHz)	Signal Type
452.9375	Head of Train (HOT)
457.9375	End of Train (EOT)
457.9250	Distributed Power Unit (DPU)

Head of Train (HOT) and End of Train (EOT) signals are used on trains to transmit telemetry data such as brake line pressure and to monitor accidental separation of the train. An End of Train device is used to send the EOT signal which the HOT unit in the head locomotive will pick up.

Distributed Power Unit (DPU) signals are control signals that are used to control remote DPU’s on long trains. DPU’s are locomotives which are placed in the middle or rear of a train to help more evenly distribute pushing and pulling power over the entire train.

HOT, EOT and DPU signals can be decoded and monitored using the RTL-SDR, SDR receiver software such as SDR#, audio piping and some special decoding software. The decoding software goes by the names of softEOT and softDPU and can only be downloaded after registering for a Yahoo! Group at https://groups.yahoo.com/neo/groups/SoftEOT/. Registering may take a few hours as you must wait for a moderator to approve your application. These signals are most commonly found in the USA, but may be in some other countries too.

You will probably only be able to receive these signals if you live near a train yard or busy train route. To use softEOT or softDPU follow the instructions below.

1. Download the softEOT and/or softDPU zip files from the files section of the Yahoo! Group and extract them to a folder.

2. Ensure you have set the default audio device for your chosen audio piping method in the Windows sound recording settings.

3. Open your favourite SDR receiver GUI such as SDR# or SDR-Radio.

4. Set the audio output to the audio piping method you are using.

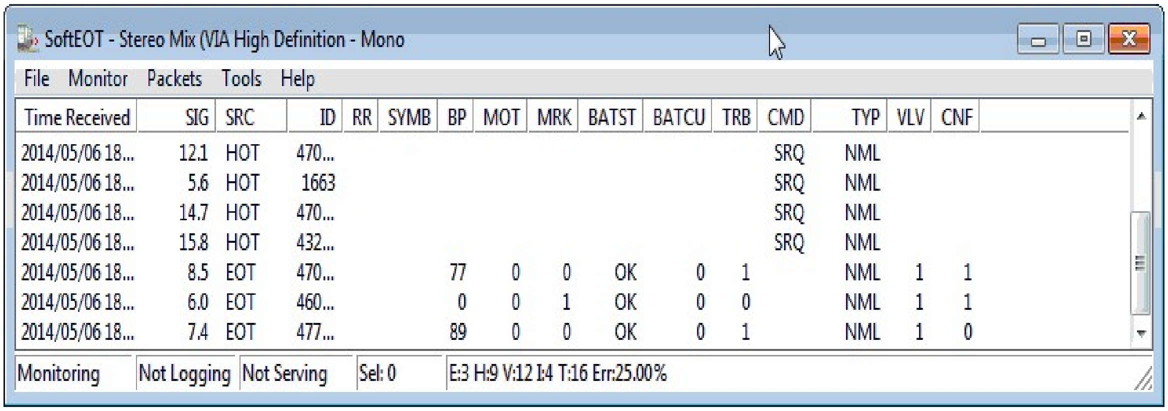
5. Tune to either a HOT, EOT or DPU frequency.

6. Open softEOT or softDPU.

7. To begin monitoring go to **Monitor -> Start Monitoring**.

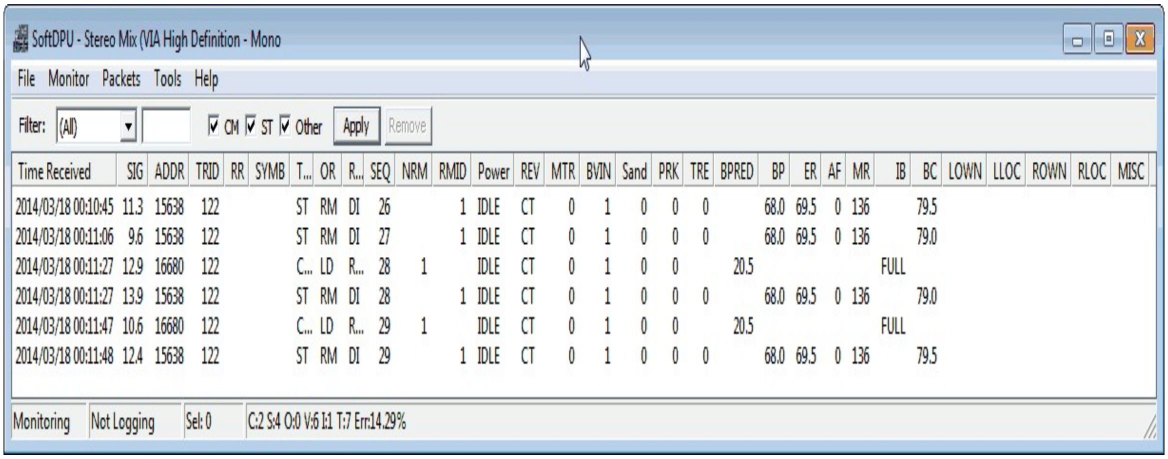
8. Go to **Monitor -> Audio Level Meter** and then adjust the volume in your SDR program or Windows such that the audio level is at around 5-10%, however the exact level isn't critical.

Decoded messages will begin to show.



Time Received	SIG	SRC	ID	RR	SYMB	BP	MOT	MRK	BATST	BATCU	TRB	CMD	TYP	VLV	CNF
2014/05/06 18...	12.1	HOT	470...									SRQ	NML		
2014/05/06 18...	5.6	HOT	1663									SRQ	NML		
2014/05/06 18...	14.7	HOT	470...									SRQ	NML		
2014/05/06 18...	15.8	HOT	432...									SRQ	NML		
2014/05/06 18...	8.5	EOT	470...			77	0	0	OK	0	1		NML	1	1
2014/05/06 18...	6.0	EOT	460...			0	0	1	OK	0	0		NML	1	1
2014/05/06 18...	7.4	EOT	477...			89	0	0	OK	0	1		NML	1	0

Monitoring | Not Logging | Not Serving | Sel: 0 | E:3 H:9 V:12 I:4 T:16 Err:25.00%



Filter: (All) ☑ CM ☑ ST ☑ Other

Time Received	SIG	ADDR	TRID	RR	SYMB	T...	OR	R...	SEQ	NRM	RMID	Power	REV	MTR	BVIN	Sand	PRK	TRE	BPRED	BP	ER	AF	MR	IB	BC	LOWN	LLOC	ROWN	RLOC	MISC
2014/03/18 00:10:45	11.3	15638	122			ST	RM	DI	26		1	IDLE	CT	0	1	0	0	0		68.0	69.5	0	136		79.5					
2014/03/18 00:11:06	9.6	15638	122			ST	RM	DI	27		1	IDLE	CT	0	1	0	0	0		68.0	69.5	0	136		79.0					
2014/03/18 00:11:27	12.9	16680	122			C...	LD	R...	28	1		IDLE	CT	0	1	0	0		20.5					FULL						
2014/03/18 00:11:27	13.9	15638	122			ST	RM	DI	28		1	IDLE	CT	0	1	0	0	0		68.0	69.5	0	136		79.0					
2014/03/18 00:11:47	10.6	16680	122			C...	LD	R...	29	1		IDLE	CT	0	1	0	0		20.5					FULL						
2014/03/18 00:11:48	12.4	15638	122			ST	RM	DI	29		1	IDLE	CT	0	1	0	0	0		68.0	69.5	0	136		79.5					

Monitoring | Not Logging | Sel: 0 | C:2 S:4 O:0 V:6 I:1 T:7 Err:14.29%

If you live in Austria, Belgium, Denmark, France, Germany, Ireland, Netherlands or Spain another decoder that can work for trains in those countries is COAA's TrainPlotter which can be downloaded from https://www.coaa.co.uk/trainplotter.htm. TrainPlotter is not free but comes with a 21 day free trial. TrainPlotter decodes train telemetry signals which are sent at frequencies around 460 MHz using NFM. Use the same process of receiving the signal in a SDR receiver like SDR# using NFM mode and piping the audio into TrainPlotter.

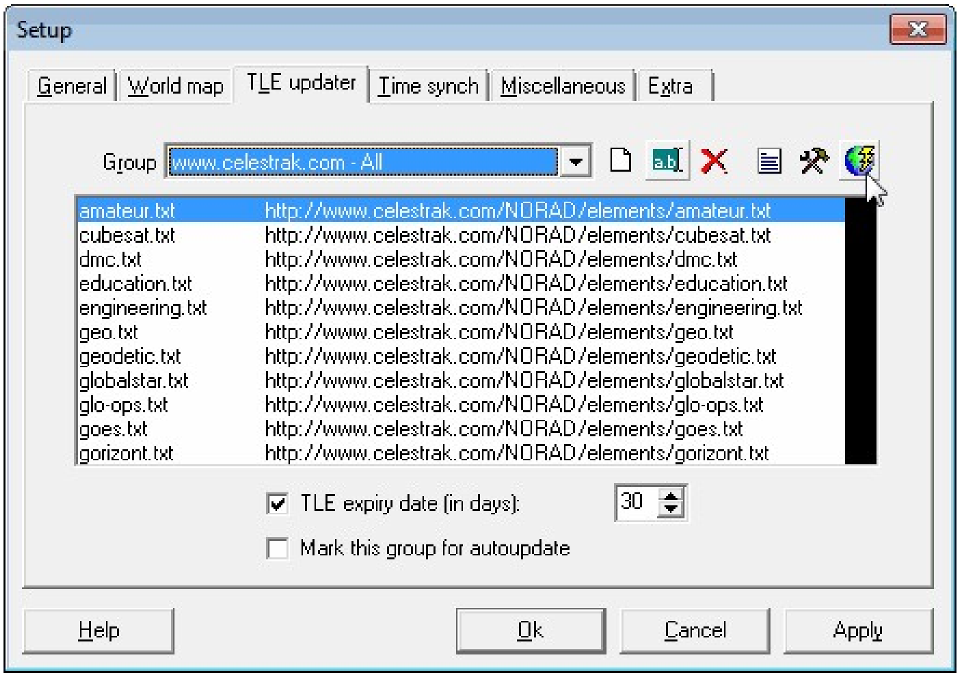
DECODING FUNCUBE SATELLITE TELEMETRY

The Funcube-1 (a.k.a AO-78) is an educational amateur radio “cubesat” satellite. It’s signal is fairly easy to receive, as it passes by almost every location on earth three times in the morning and three times at night transmitting telemetry at around 145 MHz. With an RTL-SDR, decent satellite antenna and the Funcube Telemetry software you can decode the Funcube-1’s telemetry data. For antennas we recommend a turnstile or QFH antenna tuned to 145 MHz, but a simple dipole antenna can also work. See the Antenna Chapter for more information on suitable antennas.

To determine when the Funcube-1 will pass over at your location, use the free Orbitron software. It can be downloaded from http://www.stoff.pl/. Once installed follow these instructions to set up Funcube-1 tracking.

1. Open Orbitron in Administrator Mode (if in Windows Vista/7/8), by right clicking it, and selecting Run as Administrator. Orbitron may open in full screen mode. Press Alt+Enter to exit full screen if you wish.

2. You will probably also be initially presented with a TLE file update screen. You can leave all the boxes as default. Click on the update button 🌍. Orbitron will download the new TLE files. The TLE files contain the satellite orbit information and will need to be periodically updated every few days. Running Orbitron in Administrator mode is important, as otherwise the TLE files will not be able to be written to.



Group www.celestrak.com - All

amateur.txt	http://www.celestrak.com/NORAD/elements/amateur.txt
cubesat.txt	http://www.celestrak.com/NORAD/elements/cubesat.txt
dmc.txt	http://www.celestrak.com/NORAD/elements/dmc.txt
education.txt	http://www.celestrak.com/NORAD/elements/education.txt
engineering.txt	http://www.celestrak.com/NORAD/elements/engineering.txt
geo.txt	http://www.celestrak.com/NORAD/elements/geo.txt
geodetic.txt	http://www.celestrak.com/NORAD/elements/geodetic.txt
globalstar.txt	http://www.celestrak.com/NORAD/elements/globalstar.txt
glo-ops.txt	http://www.celestrak.com/NORAD/elements/glo-ops.txt
goes.txt	http://www.celestrak.com/NORAD/elements/goes.txt
gorizont.txt	http://www.celestrak.com/NORAD/elements/gorizont.txt

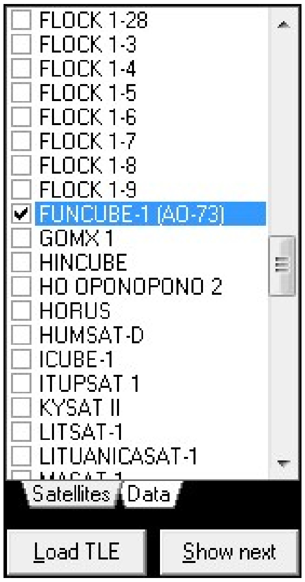
☑ TLE expiry date (in days): 30

☐ Mark this group for autoupdate

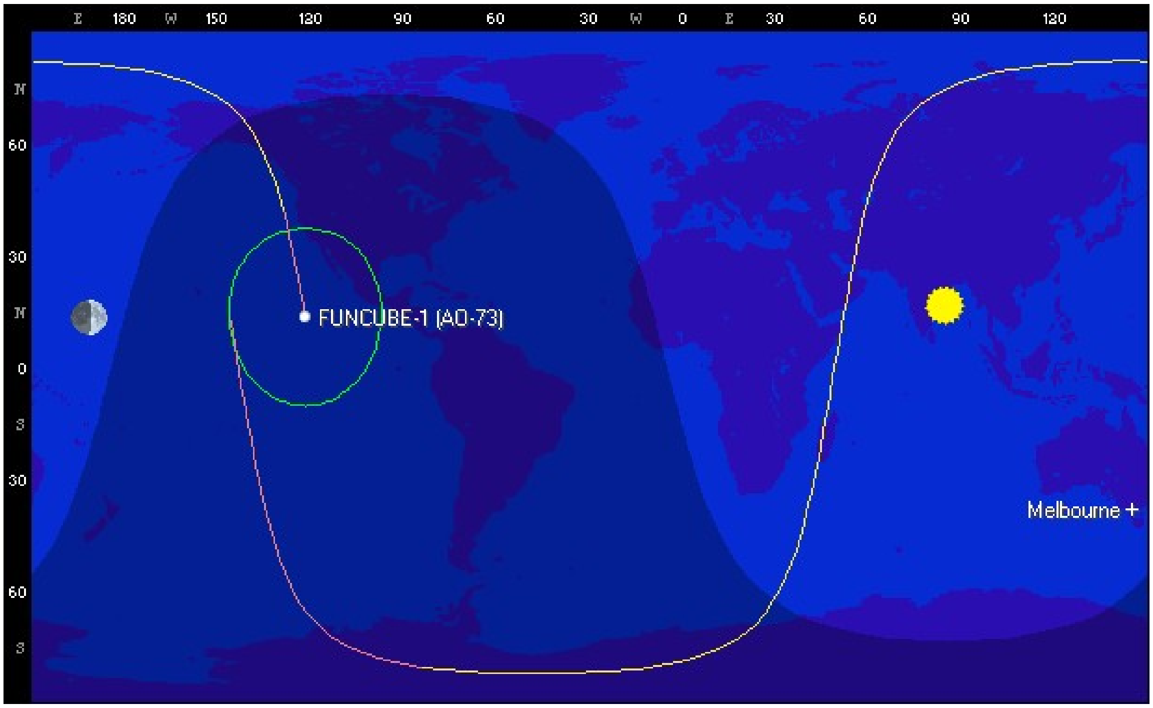
3. Now click the Load TLE button Load TLE on the right of the Orbitron window.

4. Open Cubesat.txt.

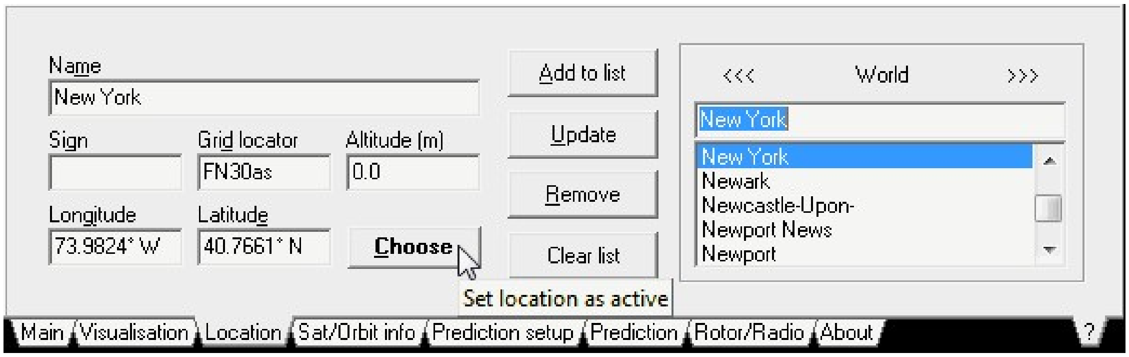
5. There will now be a list of satellites in the top right of the Orbitron window. Scroll down and place a check next to Funcube-1 (AO-73). The current position of the satellite will show in the Orbitron map.



☐ FLOCK 1-28
☐ FLOCK 1-3
☐ FLOCK 1-4
☐ FLOCK 1-5
☐ FLOCK 1-6
☐ FLOCK 1-7
☐ FLOCK 1-8
☐ FLOCK 1-9
☑ FUNCUBE-1 (AO-73)
☐ GOMX 1
☐ HINCUBE
☐ HO OPONOPONO 2
☐ HORUS
☐ HUMSAT-D
☐ ICUBE-1
☐ ITUPSAT 1
☐ KYSAT II
☐ LITSAT-1
☐ LITUANICASAT-1



6. To predict when the satellite will be overhead, first set your location by clicking on the location tab at the bottom of the Orbitron window. Either enter your lat/long coordinates, or find your city in the countries list on the right. Click the Choose button when done.



Name: New York

Sign:

Grid locator: FN30as

Altitude (m): 0.0

Longitude: 73.9824° W

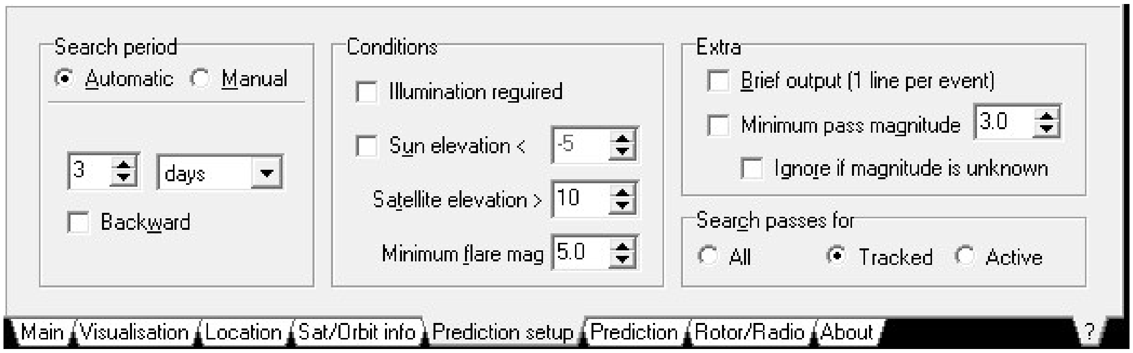
Latitude: 40.7661° N

World

New York

- New York
- Newark
- Newcastle-Upon-
- Newport News
- Newport

7. Now click on the Prediction setup tab and uncheck “Illumination Required”.



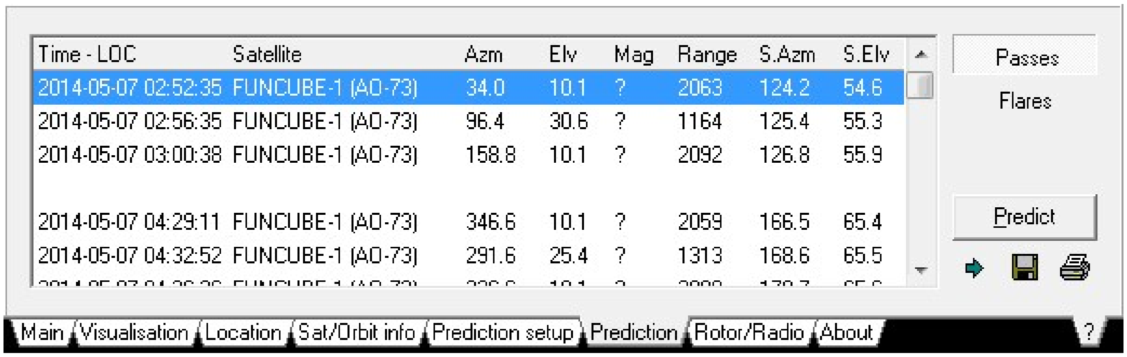
Search period
● Automatic ○ Manual
3 days
□ Backward

Conditions
□ Illumination required
□ Sun elevation < -5
Satellite elevation > 10
Minimum flare mag 5.0

Extra
□ Brief output (1 line per event)
□ Minimum pass magnitude 3.0
□ Ignore if magnitude is unknown

Search passes for
○ All ● Tracked ○ Active

8. Go to the Prediction tab and then click on the Predict button. Now you will see the local times at which the Funcube will pass overhead.

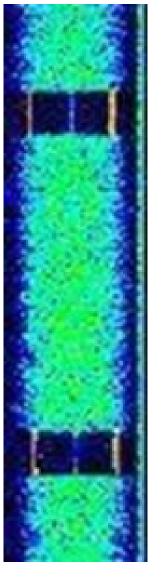


Time - LOC	Satellite	Azm	Elv	Mag	Range	S.Azm	S.Elv
2014-05-07 02:52:35	FUNCUBE-1 (AO-73)	34.0	10.1	?	2063	124.2	54.6
2014-05-07 02:56:35	FUNCUBE-1 (AO-73)	96.4	30.6	?	1164	125.4	55.3
2014-05-07 03:00:38	FUNCUBE-1 (AO-73)	158.8	10.1	?	2092	126.8	55.9
2014-05-07 04:29:11	FUNCUBE-1 (AO-73)	346.6	10.1	?	2059	166.5	65.4
2014-05-07 04:32:52	FUNCUBE-1 (AO-73)	291.6	25.4	?	1313	168.6	65.5

You may also want to look at the NOAA satellite Orbitron guide which can show you how to interface Orbitron with SDR# to automatically tune to the Funcube signal.

To decode the Funcube-1 telemetry follow these instructions.

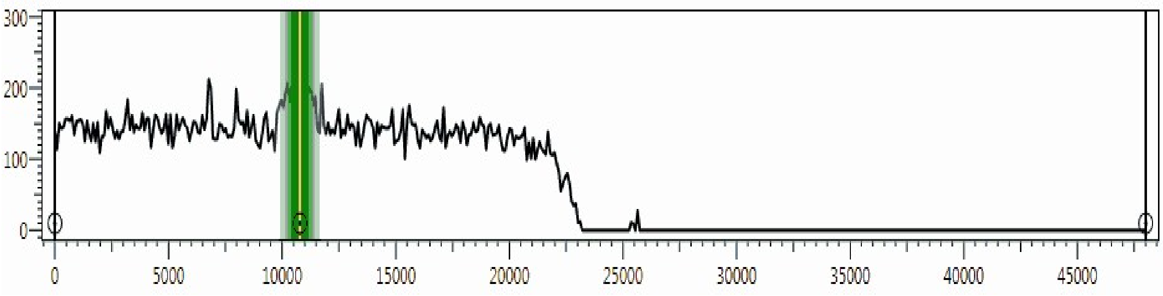
1. Open your favourite SDR receiver software such as SDR#, set your default audio piping method as the output and start the SDR.
2. Set the receive mode to USB, with a largish bandwidth of around 10-20 kHz if possible. Tip: SDR-Console is the best for this as it defaultly allows large USB bandwidths. Otherwise SDR# can be used with a modified minoutput sample rate.
3. Tune to a frequency around 145.950 - 145.970 MHz and look for the Funcube signal. Funcube telemetry data looks like this on the waterfall.



4. Open the Funcube Telemetry Dashboard software which can be downloaded from http://funcube.org.uk/working-documents/funcube-telemetry-dashboard/.

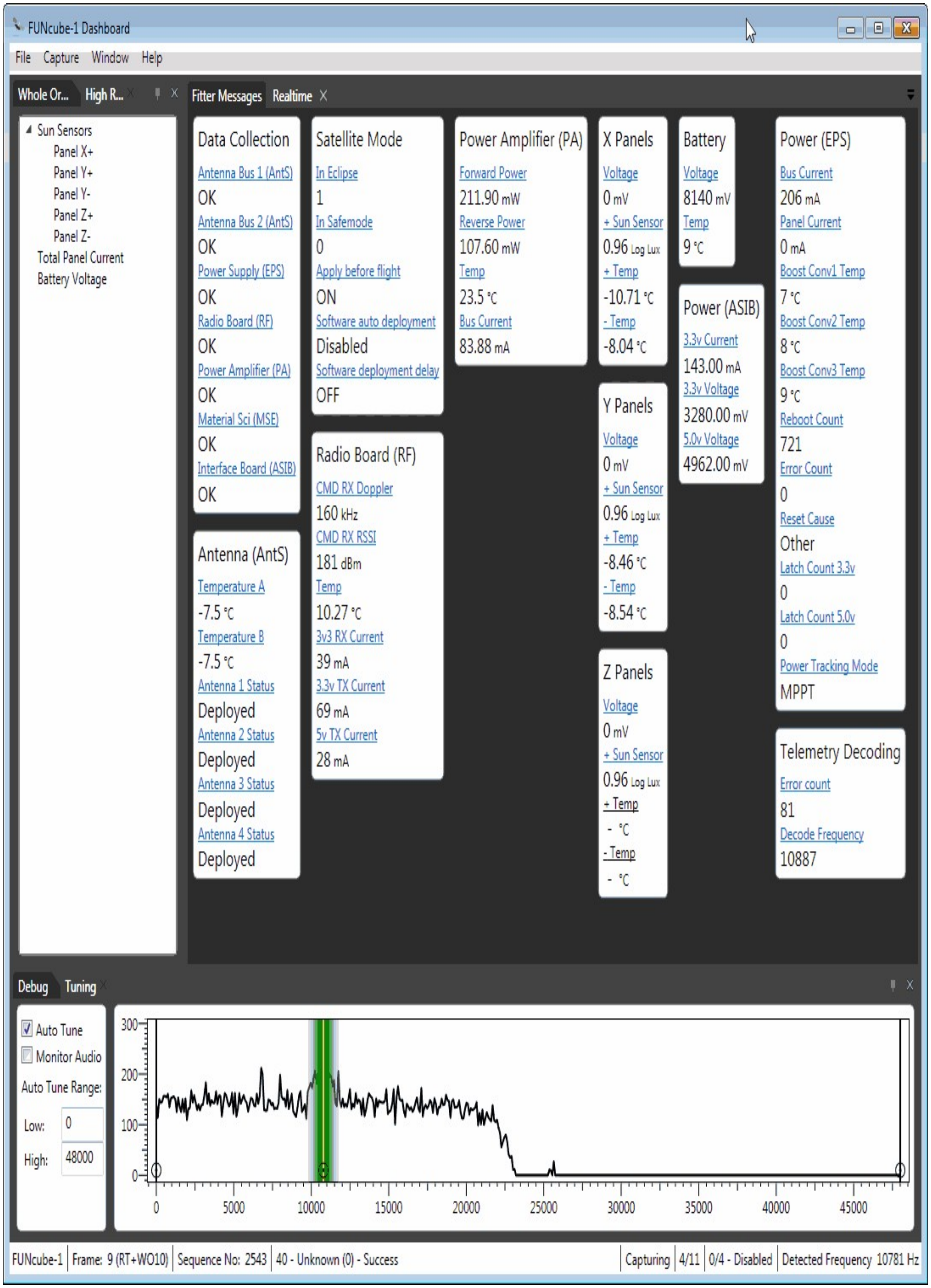
5. Go to **Capture -> Capture from soundcard** to begin capturing the data.

6. If it is working correctly, the audio graph at the bottom of the window will start graphing audio levels.



7. Ensure that “Auto Tune” is selected.

8. At this point if you are receiving strong Funcube-1 telemetry signals the dashboard should begin to show decoded telemetry.



Data Collection

Antenna Bus 1 (AntS)
OK
Antenna Bus 2 (AntS)
OK
Power Supply (EPS)
OK
Radio Board (RF)
OK
Power Amplifier (PA)
OK
Material Sci (MSE)
OK
Interface Board (ASIB)
OK

Antenna (AntS)

Temperature A
-7.5 °C
Temperature B
-7.5 °C
Antenna 1 Status
Deployed
Antenna 2 Status
Deployed
Antenna 3 Status
Deployed
Antenna 4 Status
Deployed

Satellite Mode

In Eclipse
1
In Safemode
0
Apply before flight
ON
Software auto deployment
Disabled
Software deployment delay
OFF

Radio Board (RF)

CMD RX Doppler
160 kHz
CMD RX RSSI
181 dBm
Temp
10.27 °C
3v3 RX Current
39 mA
3.3v TX Current
69 mA
5v TX Current
28 mA

Power Amplifier (PA)

Forward Power
211.90 mW
Reverse Power
107.60 mW
Temp
23.5 °C
Bus Current
83.88 mA

X Panels

Voltage
0 mV
+ Sun Sensor
0.96 Log Lux
+ Temp
-10.71 °C
- Temp
-8.04 °C

Y Panels

Voltage
0 mV
+ Sun Sensor
0.96 Log Lux
+ Temp
-8.46 °C
- Temp
-8.54 °C

Z Panels

Voltage
0 mV
+ Sun Sensor
0.96 Log Lux
+ Temp
- °C
- Temp
- °C

Battery

Voltage
8140 mV
Temp
9 °C

Power (ASIB)

3.3v Current
143.00 mA
3.3v Voltage
3280.00 mV
5.0v Voltage
4962.00 mV

Power (EPS)

Bus Current
206 mA
Panel Current
0 mA
Boost Conv1 Temp
7 °C
Boost Conv2 Temp
8 °C
Boost Conv3 Temp
9 °C
Reboot Count
721
Error Count
0
Reset Cause
Other
Latch Count 3.3v
0
Latch Count 5.0v
0
Power Tracking Mode
MPPT

Telemetry Decoding

Error count
81
Decode Frequency
10887

☑ Auto Tune
□ Monitor Audio
Auto Tune Range:
Low: 0
High: 48000

FUNcube-1 | Frame: 9 (RT+WO10) | Sequence No: 2543 | 40 - Unknown (0) - Success | Capturing | 4/11 | 0/4 - Disabled | Detected Frequency 10781 Hz

DECODING ORBCOMM SATELLITES

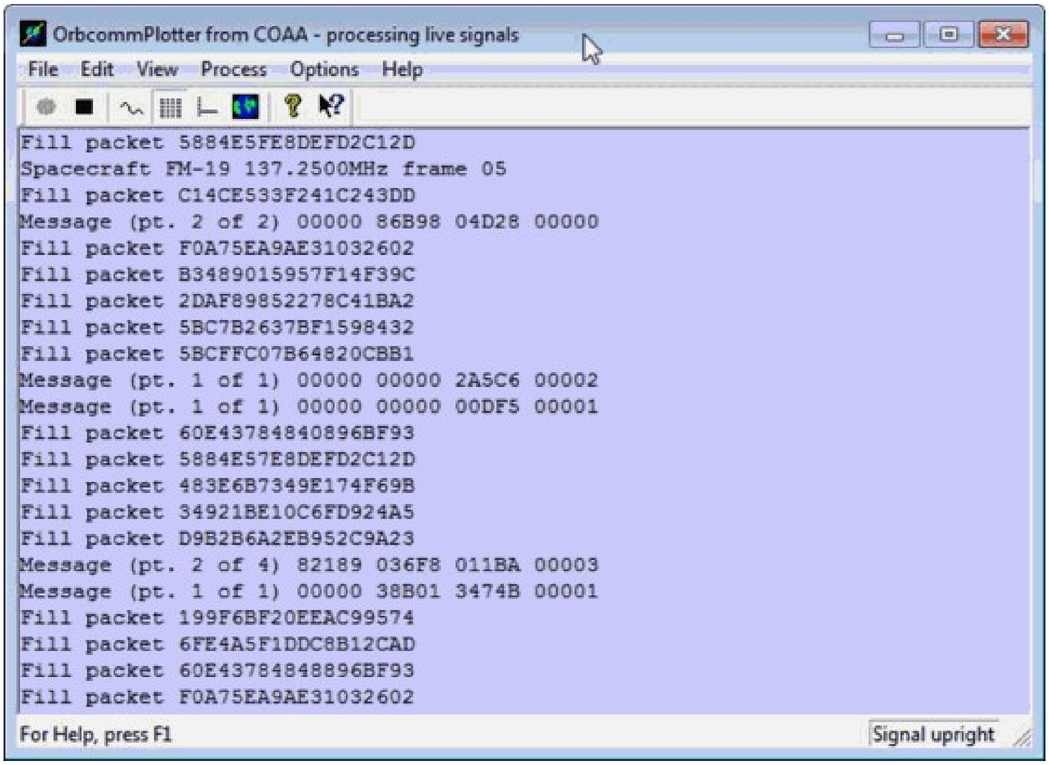
Orbcomm satellites are low earth orbit communications satellites that provide two way data services and also global positioning services to portable terminals. They transmit data at 137 - 138 MHz making them easy for scanners to receive with a good satellite antenna. They are also easy to receive as there are many orbcomm satellites in the sky positioned such that there will almost always be one available to receive at any point on the earth.

While Orbcomm signals can be received with a standard whip antenna tuned to the 137-138 MHz frequencies, a dedicated satellite antenna such as a turnstile or QFH antenna will provide much improved reception. See the Antenna Chapter for more information on suitable antennas.

Although you cannot decode actual Orbcomm data with a radio scanner like the RTL-SDR, you can still decode Orbcomm telemetry which will let you find out satellite positions and uplink/download frequencies. To do this a program called OrbcommPlotter can be used which can be obtained from http://www.coaa.co.uk/orbcommplotter.htm. Orbcomm plotter is not free, but there is a 21 day trial which can be used before paying 24 euros.

To use OrbcommPlotter follow the steps below.

1. Open your favourite SDR receiver such as SDR# or SDR-Radio.
2. Set the audio output device to your preferred audio piping method.
3. Tune to an Orbcomm frequency and set the receiver to NFM mode.
4. Open OrbcommPlotter.
5. Go to **Options->Audio->Source** and choose the audio piping method you used in your SDR receiver.
6. Go to the signal screen by pressing the signal button ∿. Ensure the signal level is loud enough, but not clipping.
7. Go to the Messages screen by pressing the messages button ▦. Orbcomm telemetry should be displayed in this window.



```
Fill packet 5884E5FE8DEFD2C12D
Spacecraft FM-19 137.2500MHz frame 05
Fill packet C14CE533F241C243DD
Message (pt. 2 of 2) 00000 86B98 04D28 00000
Fill packet F0A75EA9AE31032602
Fill packet B3489015957F14F39C
Fill packet 2DAF89852278C41BA2
Fill packet 5BC7B2637BF1598432
Fill packet 5BCFFC07B64820CBB1
Message (pt. 1 of 1) 00000 00000 2A5C6 00002
Message (pt. 1 of 1) 00000 00000 00DF5 00001
Fill packet 60E43784840896BF93
Fill packet 5884E57E8DEFD2C12D
Fill packet 483E6B7349E174F69B
Fill packet 34921BE10C6FD924A5
Fill packet D9B2B6A2EB952C9A23
Message (pt. 2 of 4) 82189 036F8 011BA 00003
Message (pt. 1 of 1) 00000 38B01 3474B 00001
Fill packet 199F6BF20EEAC99574
Fill packet 6FE4A5F1DDC8B12CAD
Fill packet 60E43784848896BF93
Fill packet F0A75EA9AE31032602
```

For Help, press F1

Signal upright

You may also wish to use Orbitron to automatically make SDR# tune to visible Orbcomm satellite frequencies. See the Orbitron tutorial in the NOAA weather satellite guide, but use Orbcomm satellites instead of NOAA weather satellites.

REFERENCES AND ADDITIONAL LINKS

https://www.youtube.com/watch?v=O-5MOMuH4v0

https://www.youtube.com/watch?v=XA5OzTH_dDA

ISS SATELLITE RECEPTION

The International Space Station sometimes transmits plain NFM audio transmissions during spacewalks. These can be found at around 121 MHz and 130 MHz. The ISS astronauts also sometimes communicate to schools using ham radio at a downlink frequency of around 145 MHz. The signal is usually quite strong so most satellite antennas tuned for the 137 MHz satellite band will work and even some non tuned or non satellite antennas may work.

The ISS also has an APRS packet radio repeater on 145.825 MHz which can be received when the satellite is overhead. See the APRS tutorial for more information on decoding APRS.

You may also wish to use Orbitron to automatically allow SDR# to tune to the ISS satellite frequency as soon as it is available. See the Orbitron tutorial in the NOAA weather satellite guide but use the ISS instead of the NOAA satellites.

WEATHER/TEMPERATURE SENSOR DECODING GUIDE

Many weather and temperature sensors transmit on the 433 MHz ISM band. A Linux command line program known as rtl_433 can be used to decode some of these transmissions. Rtl_433 can be downloaded from https://github.com/merbanan/rtl_433.

Rtl_433 can be installed with the following procedure.

```
git clone https://github.com/merbanan/rtl_433
cd rtl_433/
mkdir build
cd build
cmake ../
make
sudo make install
```

It can then be run with the following where PPM_OFFSET, FREQ and GAIN should be replaced by the required values.

```
rtl_433 -p PPM_OFFSET -f FREQ -g GAIN
```

If there is a sensor transmitting on a format that rtl_433 understands then you will be able to see sensor output data. Currently the Rubicson Temperature Sensor, Prologue Temperature Sensor, Silvercrest Remote Control, ELV EM 1000, ELV WS 2000, Waveman Switch Transmitter, Steffen Switch Transmitter and Acurite 5n1 Weather Station sensors are supported. If your sensor is not spported then you can look through the many forks of rtl_433 on Github for a fork that supports your sensor.

Rtl_433 can also run in an analysis mode by using the -a flag. In this mode rtl_433 will output raw binary data for any data that it recognizes. Using this raw binary data you could then reverse engineer the protocol for your device and decode the data.

There are also alternative rtl_433 decoders which are able to decode different sensors. There is a decoder for Oregon Scientific v1 sensors available at https://github.com/kevinmehall/rtlsdr-433m-sensor and one for v2.1 sensors at https://github.com/jaycedowell/rtl_osv21. Another Oregon Scientific decoder can be found here https://github.com/magellannh/rtl-wx. Here is another decoder that works with the Temperature Sensor S 300 IA, Temperature/Hygro sensor S 300 TH und ASH 2200 and Temperature/Hygro/Wind/Rain ("Kombi") sensor KS 200/300 https://github.com/skaringa/weather-sdr-decode.

DECODING SMART POWER METERS

Smart meters are RF enabled electricity/gas/water meters which allow meter reader personnel to collect consumption data simply by standing outside the house. Typically, this information is broadcast in the 902 - 928 MHz ISM band.

A Windows and Linux program called rtl_amr can be used to decode and monitor this information being sent. Rtl_amr should be compatible with any meter that uses the Encoder Receiver Transmitter (ERT) protocol. Here is a list of potentially compatible ERT meters that the author of rtl_amr has compiled https://github.com/bemasher/rtlamr/blob/master/meters.md. The ERT protocol is commonly used in the USA.

Rtl_amr can be obtained from https://github.com/bemasher/rtlamr. It will require installation of the Go language first. To install the Go lanuage and rtl_amr use the following instructions.

1. Download the latest version of Go from https://golang.org/dl/. Make sure you get the correct version for your OS, 32-bit or 64-bit.
2. Extract the contents of the tar file to the /usr/local folder.

```
sudo tar -C /usr/local -xzf go1.3.1.linux-amd64.tar.gz
```

3. Add the following lines to the ~/.bashrc text file.

```
export GOROOT=/usr/local/go
export GOPATH=$HOME/go
export PATH=$PATH:$GOROOT/bin:$GOPATH/bin
```

4. In a terminal type the following to install rtl_amr.

```
go get github.com/bemasher/rtlamr
```

5. Now rtlamr will have been downloaded in the $HOME/go/bin./rt folder.
6. Rtlamr can be used by running `rtl_tcp` in one terminal window and `rtl_amr` in another.

If you have an Efergy Energy Meter a Linux program called rtl-sdr-efergy can be used to decode it https://github.com/jdesbonnet/rtl-sdr-efergy. Energy Count 3000 meters can be decoded with the Linux program at https://github.com/EmbedME/ec3k_decoder.

To install rtl-sdr-efergy use the following commands at a Linux terminal.

```
git clone https://github.com/jdesbonnet/rtl-sdr-efergy
cd rtl-sdr-efergy
sudo chmod a+x build.sh
./build.sh
```

Then run it using the following, making sure to set the frequency appropriately.

```
rtl_fm -f 433.55M -W -s 200000 -r 96000 - | ./elite-decode -r 96000 > electricity.dat
```

EC3K-Decoder can be installed with the following.

```
git clone https://github.com/EmbedME/ec3k_decoder
cd ec3k_decoder/
gcc ec3k_decoder.c -o ec3k_decoder
```

Then it can be run with the following command, making sure to change the frequency where necessery.

```
sudo rtl_fm -f 868402000 -s 200000 -A fast - | ./ec3k_decoder
```

NRF24L01+ DECODING

NRF24L01+ is a wireless 2.4 GHz protocol used by many consumer devices such as keyboards, mice, remote controls, toys and appliances.

In order to receive NRF24L01+ signals you will need to first purchase a downconverter. A downconverter will move the 2.4 GHz signal down towards a frequency that is receivable by the RTL-SDR. Downconverters can be found cheaply from China on aliexpress.com. Search for “MMDS down converter”. You will want to purchase one for the 2.2 - 2.4 GHz range, with a local oscillator (LO) or 19998 MHz. Here is an example of one that has been successfully used with the RTL-SDR before http://www.aliexpress.com/item/2012-best-selling-L-O1998MHz-MMDS-down-converter/670265064.html. Note that to use these downconverters you will need to use a bias-t to power it. See the Tracking GPS section below for more information on bias-t’s.

For decoding the packets use the Linux based command line program nrf24-btle-decoder whose download and compile instructions can be found at https://github.com/omriiluz/NRF24-BTLE-Decoder. You will need to pipe the audio from rtl_fm into the decoder. Further instructions can be found at http://blog.cyberexplorer.me/2014/01/sniffing-and-decoding-nrf24l01-and.html.

ANALYSING UNKNOWN SIGNALS

REVERSE ENGINEERING ANALYSIS PROJECTS

Since the RTL-SDR has become popular, there have been various people who have used to to analyze and sometimes even reverse engineer unknown protocols.

Here someone was able to reverse engineer a wireless wall outlet http://www.embeddedrelated.com/showarticle/620.php. Another person who did something similar with wireless wall outlets can be found here http://leetupload.com/blagosphere/index.php/2014/02/24/non-return-to-zero-askook-signal-replay/.

This is a document showing how someone was able to decode an unknown 433 MHz signal from scratch. In the end he discovers that it is a wireless temperature sensor https://github.com/r-ohare/Amateur-SIGINT.

In this reverse engineering project the experimenter was able to reverse engineer the protocol on his remote controlled ceiling fan. Later he was able to use his BladeRF (a SDR that can transmit) to control the fan http://www.irrational.net/2014/03/22/reverse-engineering-a-ceiling-fan/.

Here someone was able to receive, capture and analyze a gate opening signal for a gated community http://mightydevices.com/?p=300. And here someone reverse engineered a garage door opener remote used a TI Chronos RF watch to replay the signal http://courses.cs.tau.ac.il/embedded/projects/fall2013/watch-garage-door-opener.pdf.

Here is a tutorial on analyzing 433 MHz transmitters http://arcanumx.com/?p=44.

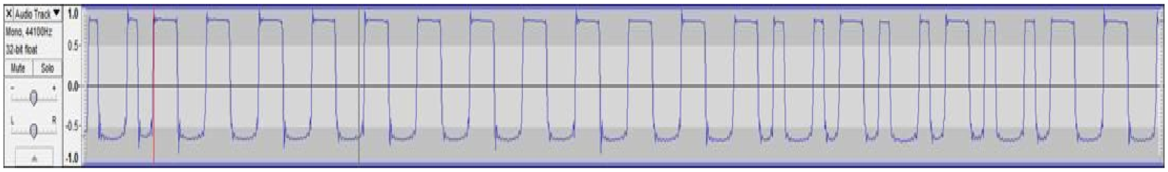
Finally, this person used her RTL-SDR to reverse engineer the protocol used by her local radio controlled bus stop display. In the end she was able to decode things like bus ID numbers and expected waiting times http://www.windytan.com/2013/11/decoding-radio-controlled-bus-stop.html.

After looking through most of these projects you should have a solid idea on how reverse engineering of a (simple) signal is done. Most of these signals were reverse engineered by recording a sample of the signal audio, looking at its audio waveform plot in a program like Audacity and then trying to determine the encoding used and the location of things like the preamble, checksum and data bits. This is fairly simple to do with AM signals that use modulation schemes like Amplitude Shift Keying (ASK) where the data is transmitted with On-Off-Keying (OOK). OOK is in a way similar to morse code (but faster), where the signal is a square wave which represents a binary string with high representing a 1 and a low representing a 0. Most simple short range wireless home devices like temperature sensors, wireless wall outlets and some remote controls use some sort of ASK or OOK encoding.

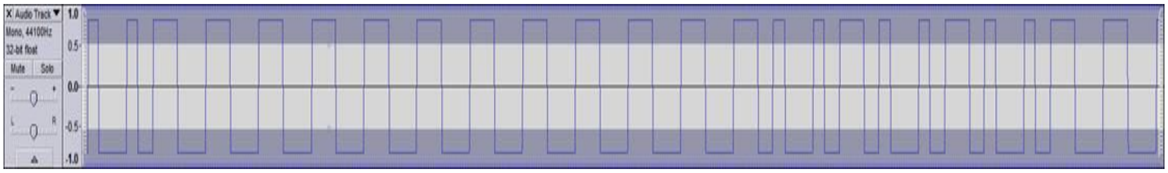
AUDACITY

Simple signal analysis can be performed with SDR# and an audio editor such as Audacity. Audacity is a free open source audio editor which can be downloaded from http://audacity.sourceforge.net/. The most important feature of audacity for signal analysis is that it can record and display audio waveforms. To use audacity to analyze simple signals follow these instructions.

1. Start SDR# or you favourite SDR receiver software and tune to a simple waveform that you'd like to analyze. Here we tune to a binary packet from a electricity energy logger on 433.94 MHz.
2. Open Audacity, and set your input device, which should be your stereo mix or preferred audio piping method Stereo Mix (VIA High De.
3. Set the input channels to Mono 1 (Mono) Inpu.
4. Click the record button to begin recording audio. You may need to adjust the volume to get good audio levels.
5. Wait for a packet to play or let your signal play for a few seconds and then click the stop button in audacity.
6. Now delete all the static in front and after the signal by using the mouse to select the noise part of the waveform and then pressing the delete key.
7. Zoom in on the captured waveform by holding the Ctrl key whilst scrolling the mouse wheel up.
8. To increase the signal amplitude go to **Effect -> Amplify**.
9. Now you should be clearly able to see the waveform and the bits that they represent.



10. To clean up the noise from a square wave signal go to **Effect -> Leveller**. Set the degree of levelling to Heaviest and press OK. Run this leveller a few times using Ctrl+R to get a clean looking digital signal.



MINIMODEM

Minimodem is a Linux command line program capable of generating and demodulating Frequency Shift Keying (FSK) signals. The MiniModem download and install instructions can be found at http://www.whence.com/minimodem/.

OOK-DECODER

OOK-Decoder is a free command line based On-Off Keying (OOK) demodulator that is compatible with the RTL-SDR. It is available at https://github.com/jimstudt/ook-decoder.

MEASURING THE CHARACTERISTICS OF RF FILTERS WITH AN RTL-SDR AND NOISE SOURCE

Every RF filter has a certain response or characteristic that shows what frequencies it blocks and what frequencies it passes. By using an RTL-SDR dongle together with a low cost noise source it is possible to measure the response of an RF filter.

These tutorials are based heavily on information learned from Adam Alicajic's (9A4QV) YouTube videos which can be found at https://www.youtube.com/watch?v=X_gd2gWyGi4, https://www.youtube.com/watch?v=UvdaURc01Ts, https://www.youtube.com/watch?v=SAAF2FvV9g4 and https://www.youtube.com/watch?v=WTdlDF7wpLg.

CHARACTERIZING FILTERS

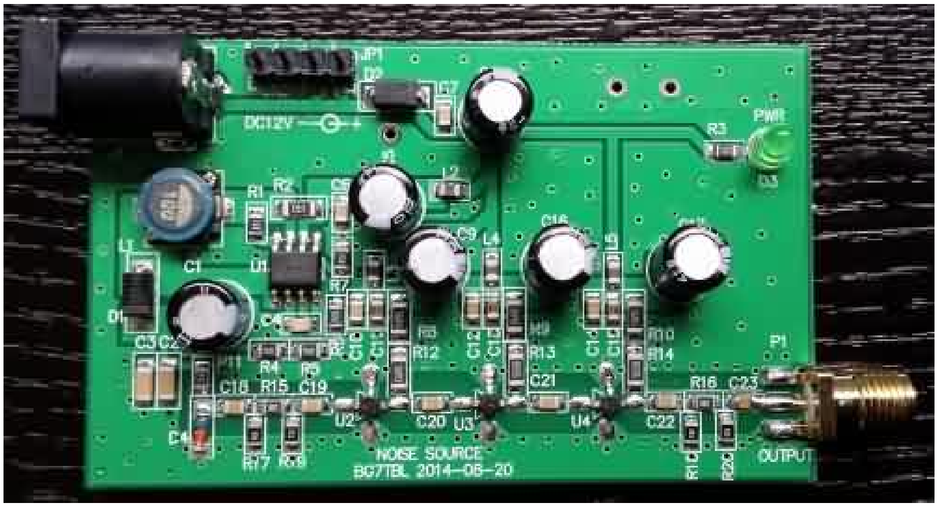
Using just a noise source and RTL-SDR dongle it is possible to determine the properties of an RF filter. A wideband noise source outputs RF noise over a wide range of frequencies, and can be thought of as transmitting noise on every frequency at once. Measuring the response of a filter can be very useful for those designing their own, or for those who just want to check the performance and characteristics of a filter they have purchased.

In our experiments we used the following equipment that we bought mostly from ebay:

EQUIPMENT

- BG7TBL Noise Source ($29.50 USD). Available on ebay and Aliexpress, just search for "DC12V 0.2A Noise Source for External Tracking Source"
- 12V Power Supply that can supply at least 0.2A for powering the noise source (~$5 USD)
- RTL-SDR Dongle (~$10-$25 USD)
- T-Junction SMA 3-Way Coax Splitter for testing coax stub notch filters ($2 USD)

The BG7TBL noise source is a wideband noise source that can provide strong noise over the entire frequency range of the RTL-SDR. It requires power from a 12V source which can be obtained from a common plug in power supply. It also uses an SMA female connector, so you may need some adapters to connect it to your filter under test (adapters can be found cheaply on ebay). Finally a quick warning: be careful when handling the circuit board after it has been powered for some time as some of the components can get very hot. Note that if the ebay store runs out of these noise sources then there is also a seller on Aliexpress with some available, just type “noise source” in the Aliexpress search bar. An image showing the 12V version of the BG7TBL noise source is shown below. There is also a 24V version available, but we prefer the 12V version because 12V power supplies are more common.



If you have a ham-it-up upconverter and are good at soldering small surface mount components you might instead consider purchasing the noise source kit add on that is sold by Nooelec. This provides the same function as the BG7TBL noise source. Here is a video showing how to build and test the ham-it-up noise source https://www.youtube.com/watch?v=HY1ijZbLfb8.

SOFTWARE

The software we use is as follows:

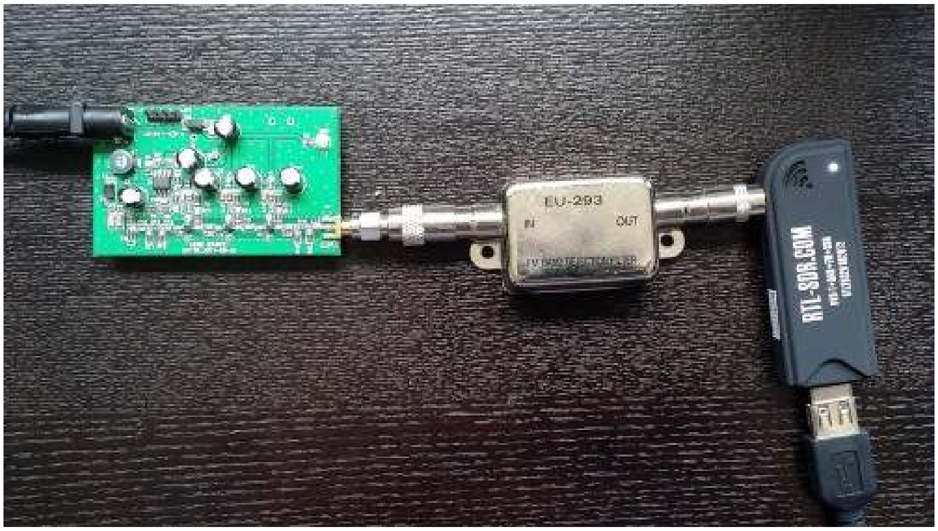
- rtl_power (keenerds build) - Available at http://igg.kmkeen.com/builds/.
- RTL-SDR Panorama (or any other rtl_power GUI or RTL-SDR spectrum analyzer) – (Optional) - Available at http://sourceforge.net/projects/guiforrtlpower/.

We will use rtl_power with the RTL-SDR Panorama GUI to record and graph a frequency sweep over the effective range of the filter. If you are experienced with command line use of rtl_power you may also use it without the GUI.

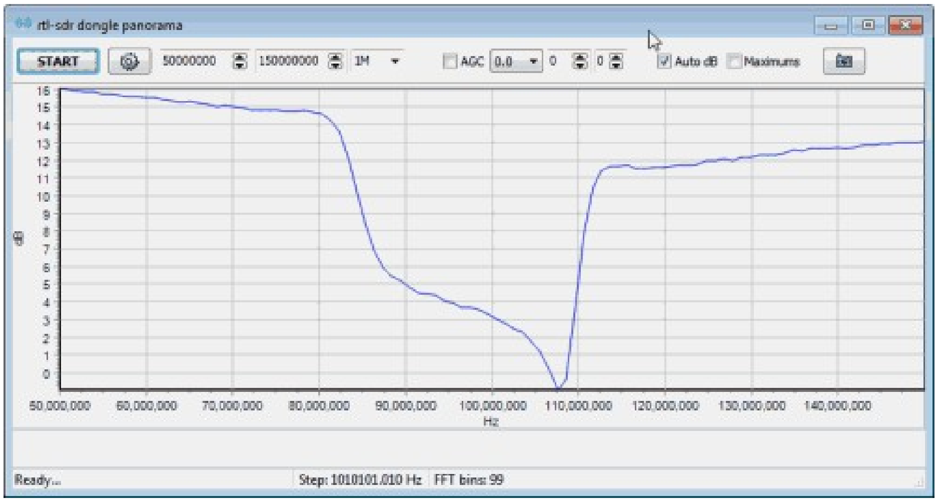
To set up RTL-SDR Panorama place the rtlpan.exe file into the same folder as rtl_power, which can be obtained from the official RTL-SDR release, or from one of keenerds builds (GitHub for Linux). We recommend using keenerds builds as his modified drivers seem to retune much faster making scans of large bandwidths significantly faster.

CHARACTERIZING FILTERS TUTORIAL

To characterize a filter simply connect the noise source to the input of the filter and the RTL-SDR to the output of the filter. Then with the RTL-SDR connected, open RTL-SDR Panorama and set the frequency range to the expected range of the filter and set the resolution to 1M. Also set the gain to 0 and enable Auto dB. Next click start and after a few seconds a graph of your filter response should be generated. The image below shows how the noise source and filters should be connected.



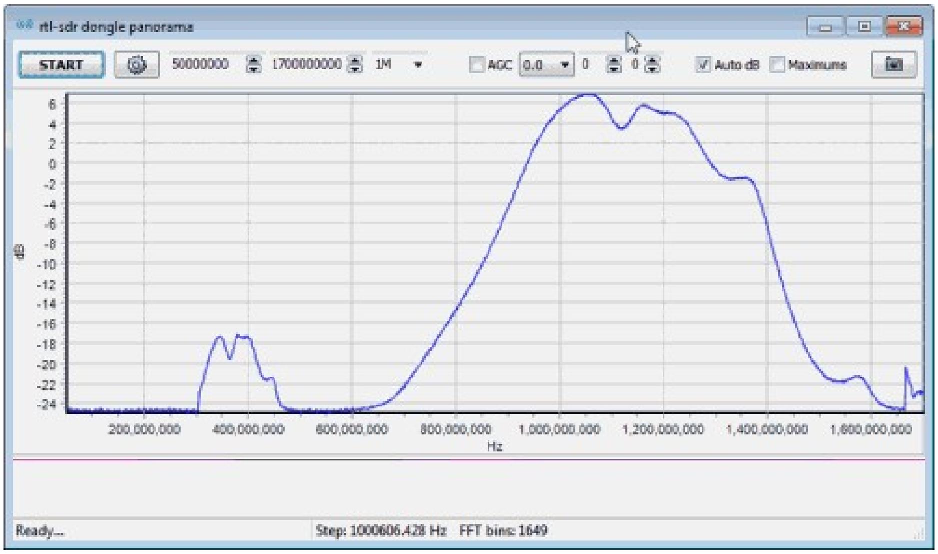
The image below shows the response of a cheap FM bandstop filter, such as the one available from MCM electronics. A band stop filter is designed to attenuate (reduce) the signal strength of overly strong FM broad cast stations. The graph is plotted between 50 MHz and 150 MHz. The response shows how the signal strength dips in the band stop region of 88 - 108 MHz, which is expected. A more expensive bandstop filter would have a more rectangular response shape, however for simply attenuating broadcast FM signals a rough shape like this is fine.



50000000 150000000 1M ☐ AGC 0.0 0 0 ☑ Auto dB ☐ Maximums

Ready... Step: 1010101.010 Hz FFT bins: 99

The image below shows the response of a 1090 MHz bandpass filter plotted between 100 MHz and 1.7 GHz. As expected the region around 1090 MHz has strong power, and neighbouring regions are tapered off.



50000000 1700000000 1M ☐ AGC 0.0 0 0 ☑ Auto dB ☐ Maximums

Ready... Step: 1000606.428 Hz FFT bins: 1649

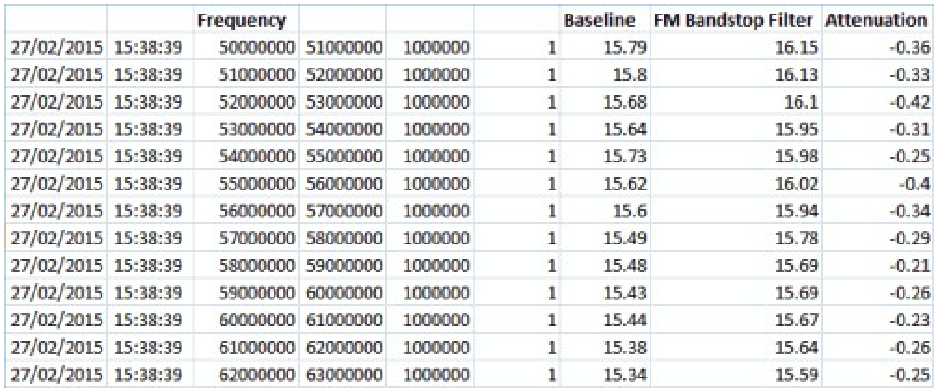
After viewing the response in RTL-SDR Panorama we recommend plotting the same graph against a baseline graph of just the noise source connected directly to the RTL-SDR. This can be done using Excel or any other similar plotting software. Each time RTL-SDR Panorama finishes a frequency sweep, the data is stored in a CSV file called scan.csv. It will be located in the same folder as rtlpan.exe.

First do a baseline sweep with the noise source connected directly to the RTL-SDR (no filters connected) and then stop RTL-SDR Panorama by pressing the stop button. Note that after you press the stop button the final sweep will still be running, so wait until it finishes first. Next rename the newly produced scan.csv to baseline.csv.

Now do a sweep with the filter connected – this will create a new scan.csv file. Next open baseline.csv in Excel. You should notice that the third column C contains the frequency values and the last column G contains the power data.

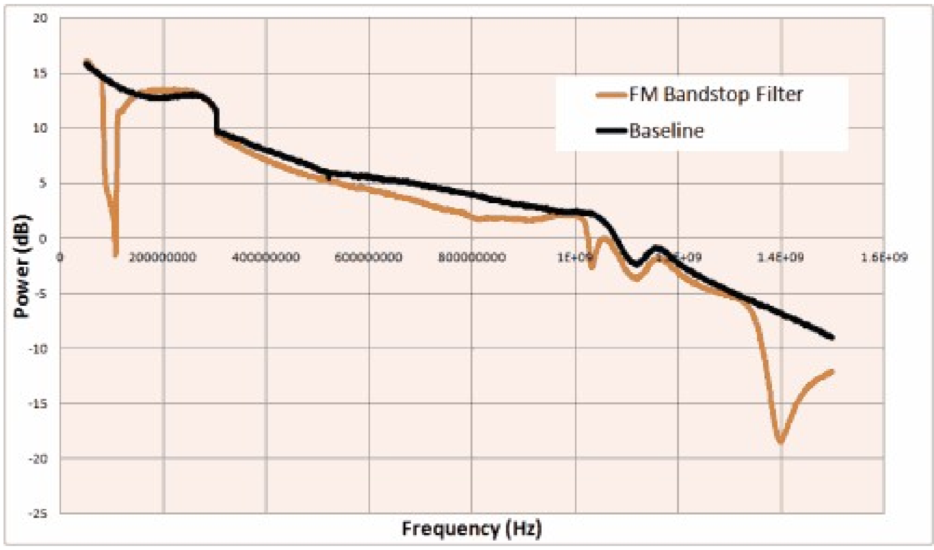
Use the Excel plotting tools to create a scatter plot in baseline.csv with frequency on the x-axis and power on the y-axis. Finally, open the scan.csv file which contains data for the scan with the filter attached. Copy the last column containing the power values (column G), and paste it into baseline.csv next to the baseline power values. Add a plot of the filtered power values to the same plot as the baseline plot and compare.

Next you can calculate the attenuation by simply subtracting the baseline column from the filter power column. Below we show an image of what your Excel sheet might look like after doing the above. There is the baseline power values next to the FM bandstop power values, and the attenuation column is simple the baseline value subtracted from the bandstop filter value.

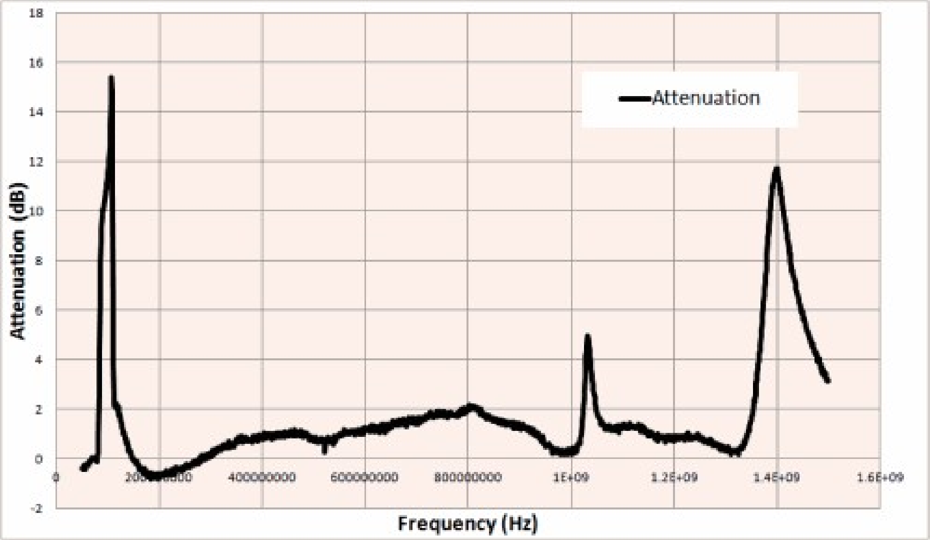


		Frequency				Baseline	FM Bandstop Filter	Attenuation
27/02/2015	15:38:39	50000000	51000000	1000000	1	15.79	16.15	-0.36
27/02/2015	15:38:39	51000000	52000000	1000000	1	15.8	16.13	-0.33
27/02/2015	15:38:39	52000000	53000000	1000000	1	15.68	16.1	-0.42
27/02/2015	15:38:39	53000000	54000000	1000000	1	15.64	15.95	-0.31
27/02/2015	15:38:39	54000000	55000000	1000000	1	15.73	15.98	-0.25
27/02/2015	15:38:39	55000000	56000000	1000000	1	15.62	16.02	-0.4
27/02/2015	15:38:39	56000000	57000000	1000000	1	15.6	15.94	-0.34
27/02/2015	15:38:39	57000000	58000000	1000000	1	15.49	15.78	-0.29
27/02/2015	15:38:39	58000000	59000000	1000000	1	15.48	15.69	-0.21
27/02/2015	15:38:39	59000000	60000000	1000000	1	15.43	15.69	-0.26
27/02/2015	15:38:39	60000000	61000000	1000000	1	15.44	15.67	-0.23
27/02/2015	15:38:39	61000000	62000000	1000000	1	15.38	15.64	-0.26
27/02/2015	15:38:39	62000000	63000000	1000000	1	15.34	15.59	-0.25

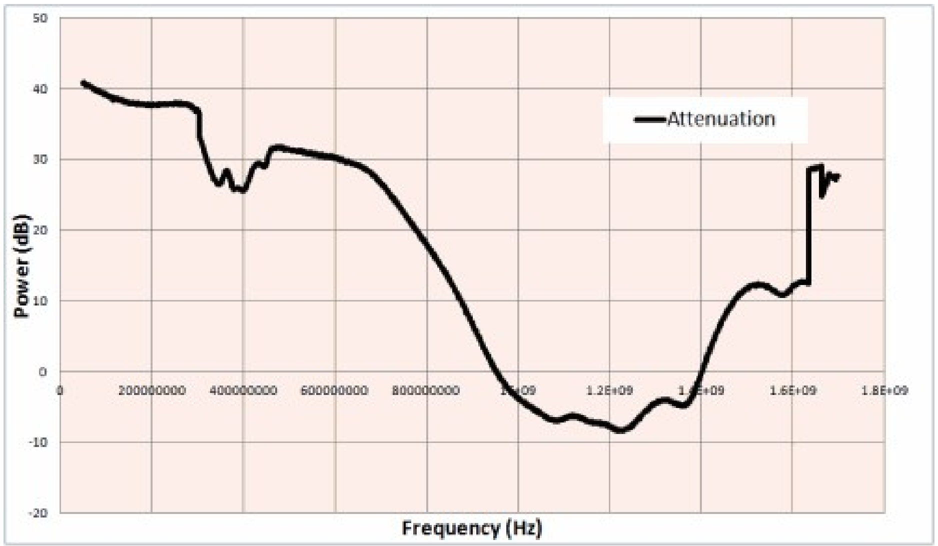
The plotted scatter graph below shows the difference between the baseline plot (no filter connected) and the FM bandstop filter plot. There is a clear dip in power in the FM band at 88 – 108 MHz. There is also another dip at around 1.4 GHz, and some slight loss over the rest of the band due to the filter or possibly due to the type of connectors used.



This next plot shows the attenuation, which is simply the baseline plot subtracted from the FM bandstop filter plot. The attenuation shows how strongly a signal at a particular frequency will be reduced in strength. In the graph we can see that the FM band has an attenuation of around 10 – 15 dB.



Below we also show the attenuation of the 1090 MHz ADS-B bandpass filter.



CHARACTERIZING COAX NOTCH/STUB FILTERS

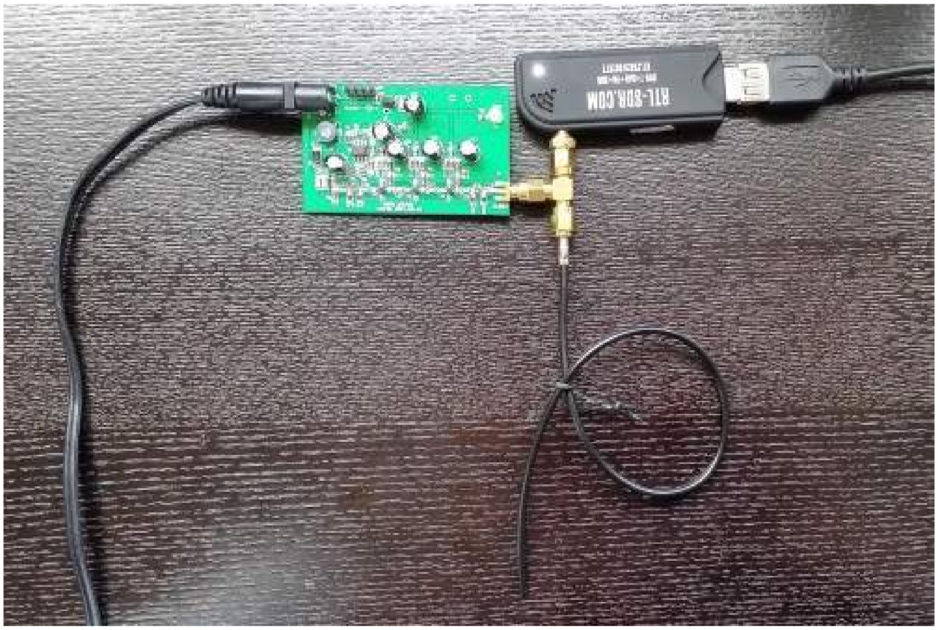
Another class of filters that can be tested with a noise source and RTL-SDR are coax notch filters, aka coax stub filters. Coax stub filters are simply a length of coax, cut to 1/4 wavelength * velocity factor of the desired notch frequency. They are usually used to reduce signal strength in a region with overly strong signals such as the FM broadcast band. Although they are called notch filters, the notch is usually quite wide. The required length can be calculated using an online calculator available at http://www.changpuak.ch/electronics/Coaxial_Stub_Filter_Designer.php, or just by using the equation below, where the Frequency is the frequency in MHz that you would like to notch out.

$$\text{Stub Length [m]} = \frac{75}{\text{Frequency [MHz]}} \times \text{Velocity Factor}$$

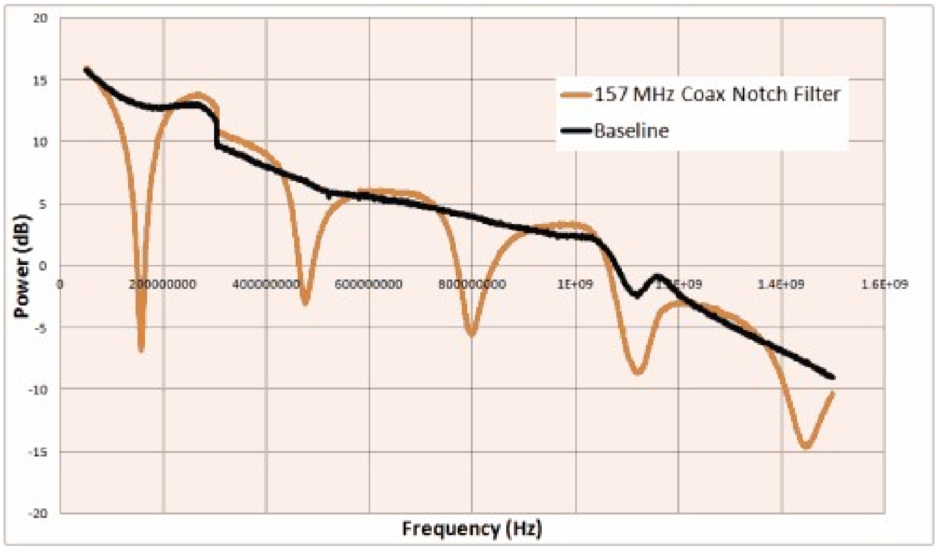
When cutting the coax we recommend cutting the coax a little longer and then using the noise source and RTL-SDR to fine tune the length.

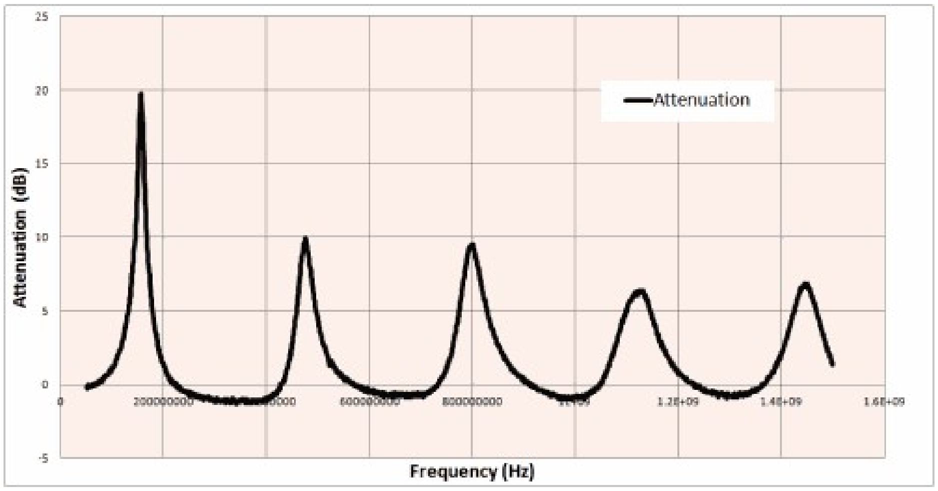
The velocity factor is a parameter that is unique to each brand and type of coax cable. If you are unsure what the velocity factor is, we recommend either using the calculator (it has pre-set velocity factor values for various coax types) or determining it through the instructions shown in next heading.

To test these filters you will need a T-junction (3-way) coax splitter. In this experiment we used a 3-way splitter with SMA connectors and used RG174U cabling with unknown velocity factor to create the coax stub filter. The image below shows how to connect everything together.



The excel plot below shows the response of a coax notch filter tuned for a notch at 157 MHz, in order to attenuate strong pager signals. The plot shows that at 157 MHz there is a deep notch as expected. Note that there are also notches present at higher frequencies as well and that the notches can have quite wide bandwidths. This is expected and a potential flaw of coax stub filters.





DETERMINING THE VELOCITY FACTOR OF COAX

Working backwards from the coax stub formula is it possible to determine the velocity factor of an unknown piece of coax. Simply create a coax stub filter of any known length with your suspect coax, then use the above method to measure where the main notch lies. Then use the equation:

$$\text{Velocity Factor} = \frac{\text{Frequency [MHz]} \times \text{Stub Length [m]}}{75}$$

In the previous experiment our coax stub measured 30.8 cm and had a main notch at 157 MHz. So we have a velocity factor given by the following calculation.

VF = 157 * 0.308 / 75= 0.645

This is quite close to the expected velocity factor value for most brands of RG174U which is 0.66.

DOWNLOADS

Below are some example spreadsheets in .xls and .ods formats that we used in the above tutorial. You can use them as a starting point for your own measurements.

Meauring Filter Attenuation:

http://www.rtl-sdr.com/wp-content/uploads/2015/03/Measure_Filter_Attenuation.xls
http://www.rtl-sdr.com/wp-content/uploads/2015/03/Measure_Filter_Attenuation.ods

MEASURING THE VSWR OF AN ANTENNA WITH A NOISE SOURCE AND RTL-SDR

With the same hardware needed to characterize filters shown in the tutorial above, and an additional piece of hardware called a directional coupler the standing wave ratio (SWR) of antennas can also be measured. The SWR of an antenna determines where the antenna is resonant and is important for tuning it for the frequency you are interested in listening to. See the Antenna Guide section in this book for more information on SWR.

EQUIPMENT

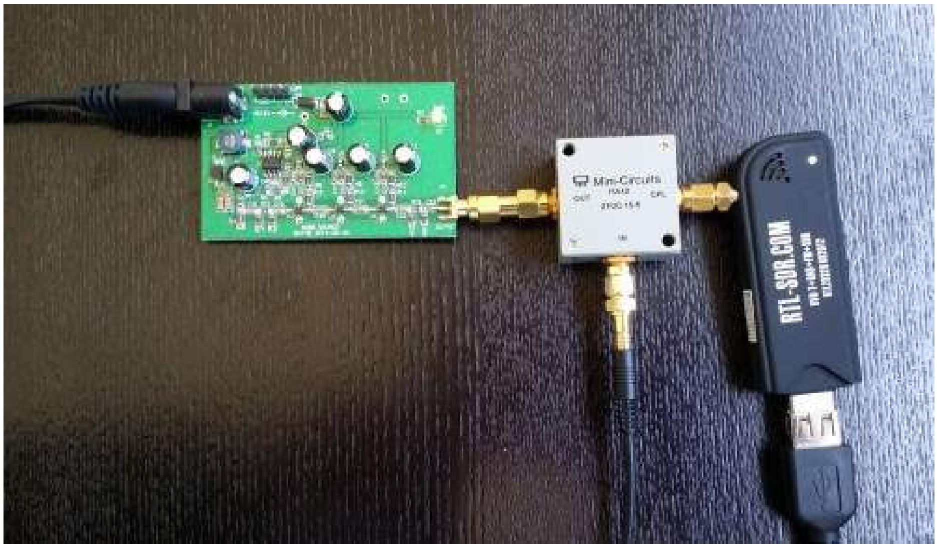
For measuring the VSWR of an antenna you will need some of the same equipment needed for characterizing filters, as well as a directional coupler.

- BG7TBL Noise Source ($29.50 USD). Available on ebay and Aliexpress, just search for “DC12V 0.2A Noise Source for External Tracking Source”
- 12V Power Supply that can supply at least 0.2A for powering the noise source (~$5 USD)
- RTL-SDR Dongle (~$10-$25 USD)
- MiniCircuits ZFDC-20-5 Directional Coupler (~$38 USD). You can usually find a Minicircuits directional coupler on ebay by searching for “Minicircuits Directional Coupler”.

MiniCircuits directional couplers can often be found used, but in good condition on ebay for about half price. In our experiments we actually used a MiniCircuits ZFDC-15-5 coupler with SMA connectors, but they appear to be sold out on ebay at the time this was written. The ZFDC-20-5 should work just as well however as will similarly specced directional couplers. Try and find a directional coupler that works over the frequencies that you are interested in. A good video showing use of a directional coupler in a similar experiment and explaining a bit about how one works can be found here https://www.youtube.com/watch?v=iBK9ZIx9YaY.

MEASURING THE VSWR OF AN ANTENNA TUTORIAL

To measure the VSWR of an antenna with the noise source and RTL-SDR we will use the directional coupler in reverse. Here we connect the noise source to the output of the coupler, the antenna to the input of the coupler and the RTL-SDR dongle to the coupling (CPL) port of the coupler. This way power from the noise source should be reflected from the antenna into the RTL-SDR. The image below shows how to connect the noise source, directional coupler and RTL-SDR togehter.



Open the RTL-SDR Panorama GUI and enter the frequency range of which you'd like to check the VSWR of the antenna. Also, set the resolution to 1M and gain to 0.

First do a sweep with the antenna disconnected from the coupler. After the sweep is done, stop RTL-SDR Panorama and rename the scan.csv file found in the same folder as rtlpan.exe to swr.csv. Note that after clicking on stop in RTL-SDR Panorama you will need to wait for the current sweep to stop before you can rename the csv file.

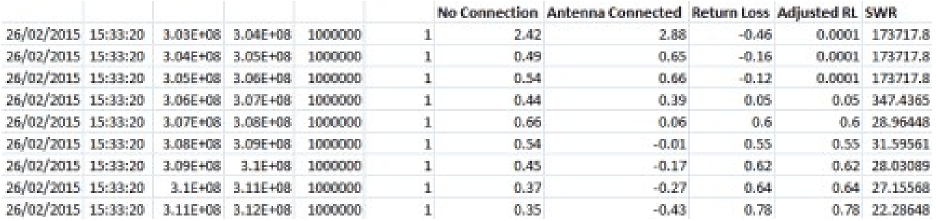
Secondly do the same sweep but with the antenna connected. Now open the swr.csv file and the latest scan.csv files in Excel or any similar program and copy the power data (the last column) from scan.csv into an empty column in swr.csv.

Now subtract the values in the first power column (power with antenna disconnected) from the values in the second power column (power with antenna connected) that you have just copied over. This will give you the return loss. If you get any negative values in the return loss column you may need to adjust the values so that the negative values are forced to be a small positive value like 0.0001. In Excel this can be done by creating a new column and using the max function on the return loss column, such as "=max(0.0001, J2)". Negative values sometimes appear because the measurement accuracy of this method is not perfect.

The VSWR can then be calculated from the return loss (RL) with the following equation.

$$VSWR = \frac{10^{\frac{RL[dB]}{20}} + 1}{10^{\frac{RL[dB]}{20}} - 1}$$

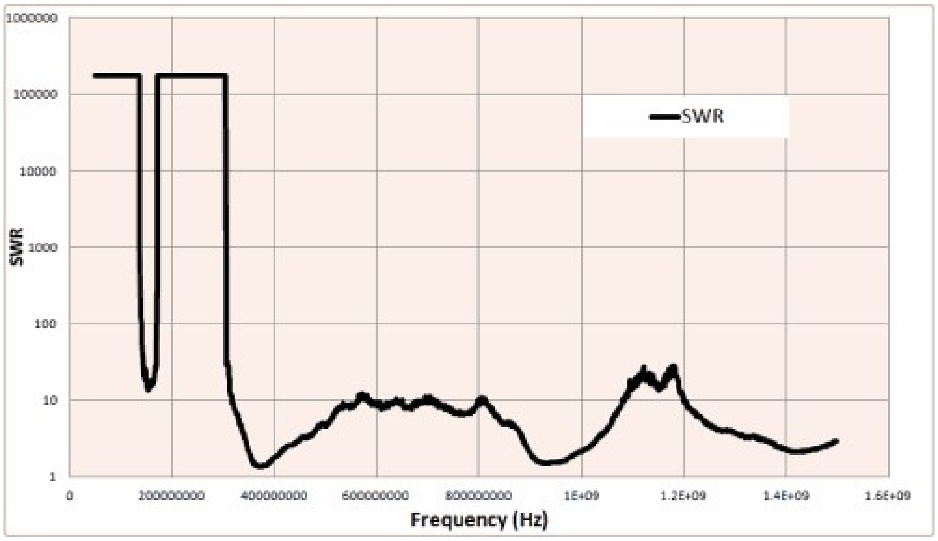
Calculate the VSWR in another column. The image below shows what your completed Excel sheet might look like.



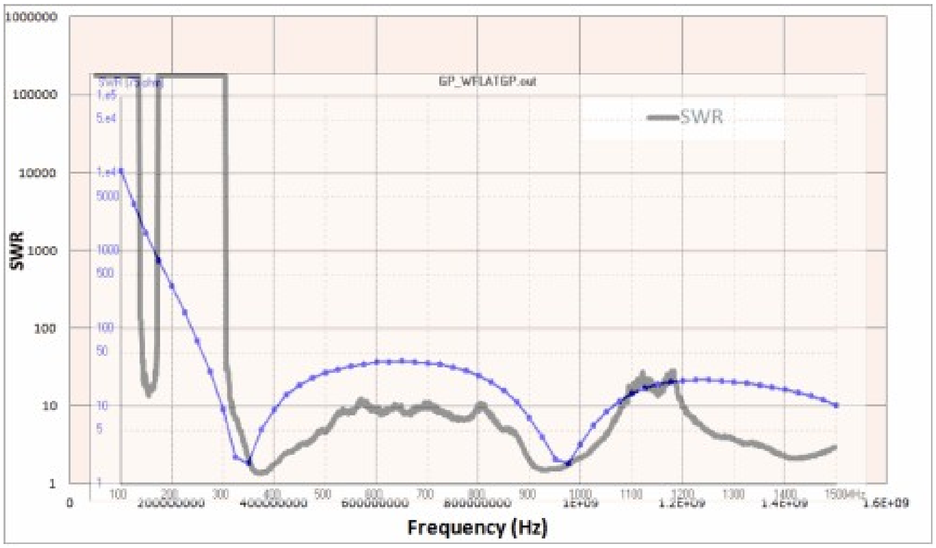
						No Connection	Antenna Connected	Return Loss	Adjusted RL	SWR
26/02/2015	15:33:20	3.03E+08	3.04E+08	1000000	1	2.42	2.88	-0.46	0.0001	173717.8
26/02/2015	15:33:20	3.04E+08	3.05E+08	1000000	1	0.49	0.65	-0.16	0.0001	173717.8
26/02/2015	15:33:20	3.05E+08	3.06E+08	1000000	1	0.54	0.66	-0.12	0.0001	173717.8
26/02/2015	15:33:20	3.06E+08	3.07E+08	1000000	1	0.44	0.39	0.05	0.05	347.4365
26/02/2015	15:33:20	3.07E+08	3.08E+08	1000000	1	0.66	0.06	0.6	0.6	28.96448
26/02/2015	15:33:20	3.08E+08	3.09E+08	1000000	1	0.54	-0.01	0.55	0.55	31.59561
26/02/2015	15:33:20	3.09E+08	3.1E+08	1000000	1	0.45	-0.17	0.62	0.62	28.03089
26/02/2015	15:33:20	3.1E+08	3.11E+08	1000000	1	0.37	-0.27	0.64	0.64	27.15568
26/02/2015	15:33:20	3.11E+08	3.12E+08	1000000	1	0.35	-0.43	0.78	0.78	22.28648

Now plot the SWR using a scatter plot with frequency on the x-axis (horizontal) and SWR on the y-axis (vertical). Change the vertical axis to a logarithmic scale for a better view.

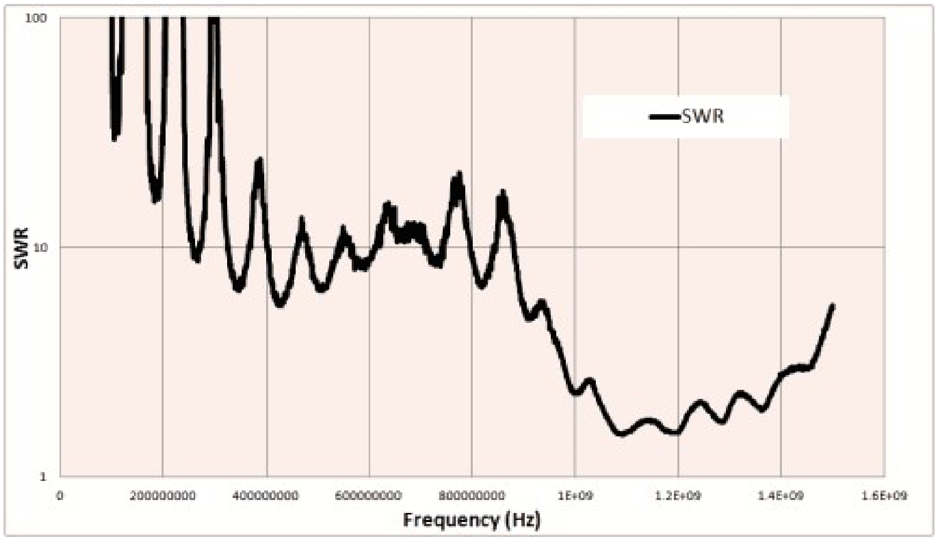
Below we show the SWR plot of a monopole antenna that is 23 cm long placed on a 20 cm diameter cookie tin that is acting as a ground plane. A perfect ground plane antenna with a 23 cm long whip should be resonant at about 310 MHz. However, because the ground plane for this antenna is only 20 cm, the SWR minimum point is pushed up to around 355 MHz. At the minimum point the SWR is about 1.15. Using a larger cookie tin would cause the minimum SWR point to approach 310 MHz.



Here we compare our results with a SWR simulation of our 23 cm monopole antenna and 20 cm ground plane made in the antenna modelling softwared called 4NEC2. See the [4NEC2 tutorial section]() for more information on this Antenna modelling software. We can see that the real world SWR plot approximately matches the simulation. Unmodelled factors such as coax cable, the type of connectors used as well as the RTL-SDR measurement accuracy can explain some of the discrepancies.



Below we show the SWR plot of a tuned quarter wave ground plane 1090 MHz antenna carefully made for ADS-B reception. The SWR at 1090 MHz is about 1.5 which is quite decent for this kind of antenna.



Many people incorrectly believe that the SWR value does not matter for receive only antennas. This misconception probably comes from the fact that a poor SWR value on a transmitting antenna could destroy the transmitter, but a poor SWR value on a receiving antenna poses no such threat. However, while it won't destroy your radio, a poor SWR value will still significantly impact reception. By using the calculator available at http://www.csgnetwork.com/vswrlosscalc.html, it is possible to determine the amount of signal loss you can expect with a particular SWR value. At a SWR of 1.5 we can expect a very small loss of about 0.177 dB (4% of the received power), whilst an SWR value of 10 gives a 4.807 dB loss (67% of the received power).

DOWNLOADS

Below are some example spreadsheets in .xls and .ods formats that we used in the above tutorial. You can use them as a starting point for your own measurements.

Measuring Antenna SWR:

http://www.rtl-sdr.com/wp-content/uploads/2015/03/Measure_SWR.xls
http://www.rtl-sdr.com/wp-content/uploads/2015/03/Measure_SWR.ods

RECEIVING AND TRACKING GPS WITH THE RTL-SDR

The RTL-SDR can be used to receive and track Global Positioning System (GPS) constellations in real time completely in software. To do this the RTL-SDR must be connected to a GPS antenna tuned at 1.57542 GHz. Most GPS antennas require power so you will need a bias-t to connect them to the RTL-SDR. Note that it is not possible to determine if the GPS signal is being receiving simply by looking at a waterfall. The reason is because GPS signals are spread spectrum and the signal strength is below the noise floor. Proper signal processing software is required to detect GPS signal such as the GNSS-SDR Linux software.

Note: We and several other testers of this software have never been able to get the GPS to get good locks using GNSS-SDR and the RTL-SDR. The only thing that has worked well is the calibration tool. It may or may not work for you. This tutorial is included for future purposes when the locking problem may be fixed and/or overcome, so please be aware that if you try this project you are unlikely to see good results.

To install the GNSS-SDR software use the following instructions.

Install GNU Radio 3.7. See the Installing GNU Radio tutorial section if you need to install it. Then install the remaining dependencies as follows:

```
sudo apt-get install libblas-dev liblapack-dev gfortran
```

```
wget http://sourceforge.net/projects/arma/files/armadillo-4.000.0.tar.gz
tar xvfz armadillo-4.000.0.tar.gz
cd armadillo-4.000.0
cmake .
make
sudo make install
```

```
wget http://gflags.googlecode.com/files/gflags-2.0.zip
unzip gflags-2.0.zip
cd gflags-2.0
./configure
make
sudo make install
```

```
wget http://google-glog.googlecode.com/files/glog-0.3.3.tar.gz
tar xvfz glog-0.3.3.tar.gz
cd glog-0.3.3
./configure
make
sudo make install
```

```
wget http://gperftools.googlecode.com/files/gperftools-2.1.tar.gz
tar xvfz gperftools-2.1.tar.gz
cd gperftools-2.1
./configure --enable-frame-pointers
make
sudo make install
```

```
wget http://googletest.googlecode.com/files/gtest-1.7.0.zip
unzip gtest-1.7.0.zip
cd gtest-1.7.0
./configure
make
```

In your $HOME/.bashrc file add the following making sure to change /home/username/gtest-1.7.0 by the actual directory where you downloaded gtest.

```
export GTEST_DIR=/home/username/gtest-1.7.0
```

Now install libosmosdr and gr-osmosdr

```
sudo apt-get install libssl-dev subversion
```

```
git clone git://git.osmocom.org/osmo-sdr.git
cd osmo-sdr/software/libosmosdr
mkdir build
cd build/
cmake ../
make
sudo make install
sudo ldconfig
cd ../../
git clone git://git.osmocom.org/gr-osmosdr
cd gr-osmosdr
mkdir build
cd build
cmake ../ -Wno-dev
make
sudo make install
sudo ldconfig
```

We are now ready to install GNSS-SDR. Before we do though we need to set some variables to tell it to install RTL-SDR support. Export the following variable with the following line.

```
export RTLSDR_DRIVER=1
```

Now also provide the path to your gr-osmosdr install using the following, making sure to replace the path directory accordingly.

```
export OSMOSDR_ROOT=/path/to/gr-osmosdr
```

Now install and build the GNSS software using the commands below.

```
svn co http://svn.code.sf.net/p/gnss-sdr/code/trunk gnss-sdr
cd gnss-sdr/build
cmake ../
make
make install
```

Before we can use the RTL-SDR with GNSS-SDR we need to run a calibration program to get around the RTL-SDR's crystal inaccuracies. But first we need to modify some configuration files to tell GNSS-SDR to use the RTL-SDR.

```
cd gnss-sdr/conf
```

Edit `front-end-cal.conf` and input your GPS receive latitude, longitude and elevation.

```
GNSS-SDR.init_latitude_deg=40.74846557442795
GNSS-SDR.init_longitude_deg=-73.98593961814200
GNSS-SDR.init_altitude_m=329.11968943169342
```

Ensure that supplemental GPS is turned on by setting the following line.

```
GNSS-SDR.SUPL_gps_enabled=true
```

Set the gains as follows (May require tweaking if you have bad reception).

```
SignalSource.AGC_enabled=true
SignalSource.gain=60
SignalSource.rf_gain=40
SignalSource.if_gain=30
```

You might also need to adjust the acquisition threshold if you have bad reception. By default the threshold is at 0.015, but it can be reduced slightly.

```
Acquisition.threshold=0.010
```

Now run the front end calibration tool by first navigating to the `gnss-sdr/install` folder and then running the following.

```
./front-end-rcvr
```

A successful run will detect a few satellites and calculate the sample rate offset. An example of a successful output is shown below.

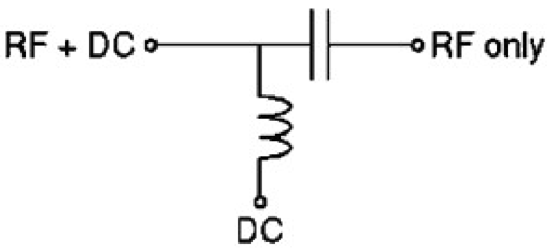
```
Initializing... Please wait.
Logging will be done at /tmp
Use front-end-cal --log_dir=/path/to/log to change that.
Trying to read ephemeris from SUPL server...
SUPL data received OK!
gr-osmosdr v0.1.0-44-g0d10f5e9 (0.1.1git) gnuradio 3.7.2git-57-g1aa07b36
built-in source types: file osmosdr fcd rtl rtl_tcp uhd hackrf netsdr
Using device #0 Realtek RTL2838UHIDIR SN: 00000001
Found Rafael Micro R820T tuner
Exact sample rate is: 2000000.052982 Hz
Actual RX Rate: 2000000.000000 [SPS]...
Actual RX Freq: 1575420000.000000 [Hz]...
PLL Frequency tune error 0.000000 [Hz]...Actual RX Gain: 40.200000 dB...
Front-end RAW samples captured
Using Volk machine: avx_32_mmx_orc
Searching for GPS Satellites in L1 band...
[ . . . . . . . . . . . . . 14 . . . 18 . . 21 22 .
. . . . . . . . . ]
Total signal acquisition run time 6.03541 [seconds]
Reference Time:
  GPS Week: 747
  GPS TOW:  208705 16696.400000
  ~ UTC:    Tue Dec 17 14:58:26 2013
Current TOW obtained from SUPL assistance = 208705
Reference location (defined in config file):
Latitude=33.5166 [�]
Longitude=73.1665 [�]
Altitude=518 [m]
Doppler analysis results:
SV ID  Measured [Hz]   Predicted [Hz]
  14   -74500.00   832.28
  18   -76500.00   -1141.35
  21   -78000.00   -2679.26
  22   -75000.00   219.07
Parameters estimation for Elonics E4000 Front-End:
Sampling frequency =2000095.73 [Hz]
IF bias present in baseband=-75410.22 [Hz]
Reference oscillator error =47.87 [ppm]
Corrected Doppler vs. Predicted
SV ID  Corrected [Hz]   Predicted [Hz]
  14   910.22   832.28
  18   -1089.78   -1141.35
  21   -2589.78   -2679.26
  22   410.22   219.07
GNSS-SDR Front-end calibration program ended.
```

Now record the sample rate and IF bias present in baseband and put it into the gnss-sdr_rtlsdr_realtime.conf file under the headings where it says to place the calibrated sample rates.

ACTIVE GPS ANTENNAS AND BIAS-T’S

Most GPS antennas are ‘active’ and require 5 volts of DC power to be sent down the coax cable. Unfortunately, we cannot directly connect a DC power source to the RTL-SDR antenna input as this would fry it.

To connect an active GPS antenna to the RTL-SDR you will need a simple electronic circuit called a bias-t. The bias-t sits between the dongle and active GPS antenna, allowing DC power to travel down the coax to the antenna whilst blocking the DC power from entering the dongle. A simple bias-t consists of a capacitor and an inductor. They can be bought relatively cheaply on Ebay, or easily constructed. A sample circuit diagram is shown below.



Attribution: Bias T by Matthias Buchmeier / CC 3.0

The capacitor value must have a small enough impedance (<<50 ohm) to allow 1.5 GHz to easily pass through and the inductor must have a high impedance (>>50ohm) to prevent the received GPS signal from shunting to ground through the battery.

To calculate bias-t values use the impedance formulas for capacitors and inductors. A good calculator can be found here http://www.qsl.net/pa2ohh/jslcimp.htm.

Impedance of a capacitor is calculated with the following equation.

X_c = 1 / (2 * pi * f * c)

And the impedance of an inductor can be calculated with the following equation.

X_l = 2 * pi * f * L

Where f is the frequency we are interested in (in this case ~1500 MHz), C is the capacitance value and L is the inductance value.

We need large capacitor and large inductor values. For example a 100 pF capacitor at 1.5 GHz (1500 MHz) has an impedance of about 1 Ohm, so it should be okay. Avoid capacitors that are too large as they can have parasitic impedance.

An inductor with a value of 0.1 uH has an impedance of about 940 Ohms at 1.5 GHz and should be okay. A homemade inductor can be wound using enamelled copper wire and a cylindrical object or a ferrite bar. To calculate windings for a home made air core inductor use this calculator http://www.66pacific.com/calculators/coil_calc.aspx. Using a ferrite core can increase inductance further.

PROJECTS FOR THE FUTURE AND BETTER SDR'S

Here we will talk about some applications that will work on current or future higher end devices.

HRPT

HRPT is an acronym for High Resolution Picture Transmission. It is similar to APT and LRPT used in the NOAA and Meteor-M2 weather satellites except that the images are much higher in resolution. HRPT is also transmitted by some NOAA weather satellites but at a frequency of around 1700 MHz. To receive this signal you will need a fairly advanced setup. It requires a high gain tracking satellite dish which is a computerized satellite dish that automatically follows the satellite in the sky. You can see the type of antenna required here http://pjm.uhf-satcom.com/twtr/antenna.jpg.

HRPT requires a bandwidth of about 3 MHz, which is just outside the RTL-SDRs stable maximum bandwidth. Newer SDRs like Airspy or the HackRF/BladeRF should be able to receive this no problem. Nevertheless, it seems as if at least one RTL-SDR user has been successful in receiving HRPT images. If you are an advanced user this GNURadio code allows a RTL-SDR to receive HRPT images https://github.com/poes-weather/gr-poes-weather/blob/master/apps/pw_rtlsdr_rx_noaa_hrpt.py.

More information at: http://www.satsignal.eu/software/hrpt.htm

HD RADIO

In the USA some WFM broadcast radio station signals are surrounded by two digital data signals. They look like two rectangles that surround the WFM signal and they carry the same audio but in a digital format. This is known as HD Radio and is a proprietary protocol that is only decodable by special radios. In the future someone may discover a way to decode these signals, or the owners of the HD Radio protocol may release information about the protocol allowing people to easily decode it.

SOFTWARE DVB-T DECODING

The RTL-SDR doesn't provide enough bandwidth for decoding DVB-T signals when in SDR mode. Future dongles with larger bandwidths may be able to decode DVB-T TV completely in software.

DECT CORDLESS PHONES

Modern cordless phones transmit in the 1.9 GHz band and higher using a protocol called DECT or Digital Enhanced Cordless Telecommunications. There is currently a GNU Radio decoder under development which may allow decoding of this protocol in the future https://www.cgran.org/wiki/GR_DECT.

OTHER SOFTWARE NOT MENTIONED YET

In this section we briefly mention a few software programs for special RTL-SDR projects that are not mentioned in the above tutorials but are worth checking out if find that you have an interest in them.

RTL-AIRBAND

This is a command line program for Windows, Linux and the Raspberry Pi that is intended for airband reception and streaming live communications to an online aggregation service like liveatc.net. It can simultaneously listen to up to 8 AM channels per dongle and can use multiple dongles to efficiently cover a wider bandwidth.

WEBSDR

WebSDR is a popular online streaming SDR service. People who have SDRs can set up the WebSDR server software which allows for many online users to publically listen to their SDR in real time. It is compatible with the RTL-SDR. The server software is free, but not publically available. You must first email the owner of WebSDR and request it. Examples of WebSDR radios and more information can be found at http://www.websdr.org/.

RTL ENTROPY

Radio noise received by a RTL-SDR dongle can be used as a high quality entropy source in Linux with a program called RTL-Entropy. Computers and their algorithms can not by themselves produce true randomness. By using a better source of random numbers, such as from the RTL-SDR receiving noise, better security can be obtained. RTL-Entropy is available from https://github.com/pwarren/rtl-entropy. A good tutorial on setting up RTL-Entropy as an entropy source on Linux can be found at http://blog.cros13.net/2014/08/cheap-entropy-using-your-rtl-sdr-as.html.

SWSCAN

This is a Linux based program which can be used to scan and listen to shortwave stations. It has a built in database of shortwave station frequencies as well as their broadcast schedules and it will even show you the stations power level and distance you are from the transmitter. Swscan is based on GNU Radio 3.7, so you will need to have that installed first. Information about SWScan can be found at http://www.reddit.com/r/RTLSDR/comments/2f6u21/swscan_a_rtlsdr_console_sw_broadcast_stations/.

OTHER INTERESTING PROJECTS PEOPLE HAVE DONE WITH THE RTL-SDR

- Received the Chinese Yutu moon rover telemetry.
  - http://www.rtl-sdr.com/receiving-chinese-yutu-moon-rover-rtl-sdr/
- Tracking of LEO satellites with an Inmarsat dish.
  - https://www.youtube.com/watch?v=ktnQ7nBCuqU
- Detected leaking FM radio from an HP laptop mic.
  - http://www.rtl-sdr.com/potential-major-security-flaw-hp-laptop-discovered-rtl-sdr/
- Attempted to reconstruct an LCD monitor image from RFI.
  - https://www.youtube.com/watch?v=5N1C3WB8c0o
- Built a dual passive radar system.
  - http://kaira.sgo.fi/2013/10/dual-antenna-passive-radar.html,
  - https://www.youtube.com/watch?v=V9gRjK2ZlzE#t=15,
  - http://www.rtl-sdr.com/passive-radar-dual-coherent-channel-rtl-sdr/
  - http://www.de8msh.blogspot.co.nz/2013/12/doppler-experiments.html
  - http://www.rtl-sdr.com/rtl-sdr-based-passive-multistatic-radar-used-track-aircraft/
- Decoded burger pager data.
  - http://www.windytan.com/2013/09/the-burger-pager.html
- Decoding tire pressure monitors.
  - http://www.rtl-sdr.com/receiving-decoding-tire-pressure-monitor-systems-using-rtl-sdr/
- ADS-B Augmented Reality on a AR-Drone.
  - http://www.youtube.com/watch?v=zeyCdk-OqG0
- RTL-SDR for radio direction finding - Finding an interfering beacon.
  - http://www.rtl-sdr.com/locating-interfering-signal-radio-direction-finding-rtl-sdr/
- Using the RTL-SDR to measure a filters frequency response.

  - http://www.rtl-sdr.com/measuring-frequency-response-bandpass-filter-rtl-sdr/

- Using the RTL-SDR to receive signals from the Lunar Reconnaissance Orbiter.
  - http://www.gat3way.eu/index.php?mact=News,cntnt01,detail,0&cntnt01articleid=210&cntnt01returnid=15

- Creating a low cost noise figure indicator.
  - http://www.canfi.eu/

UNKNOWN SIGNAL IDENTIFICATION

During your browses through the spectrum you will often encounter unknown signals. To identify them first we recommend looking at the signal identification guide over at www.sigidwiki.com and seeing if you can find a match there. If you do not see the signal there then you might try looking up the frequency on the RadioReference.com or qrg.globaltuners.com databases. Another option is to look up the signal in your countries spectrum management licence database. Most countries provide a searchable online database of all licensed frequencies. With the database you'll be able to look up a certain frequency and see who or what company it is licensed to and from there you can make an educated guess on what type of signal it is.

DVB-T HDTV ON LINUX

The RTL-SDR can still be used for its original purpose: DVB-T HD TV. Note that DVB-T TV does not exist in the USA.

DRIVERS

On Linux there are free open source drivers and software programs which can allow you to watch DVB-T TV. The easiest way to get it to work is to use a newer version of the Linux Kernel (3.6+), such as the one used by Ubuntu 13.10 or newer. On older versions you will need to install or update Video 4 Linux, or update your kernel.

Also, if you've already installed the SDR drivers on Linux then you've probably blacklisted the DVB-T drivers from loading at some point. You will need to unblacklist them to allow for TV watching or run `modprobe dvb_usb_rtl28xxu` to reload the drivers.

UPDATING VIDEO4LINUX

Note: Only do this if you are running an old version of Linux. Video4Linux can be updated by using the following commands in terminal. Note that this update may not work on all Linux versions and kernels.

```
sudo apt-get install git linux-headers-$(uname -r) build-essential patchutils libproc-processtable-perl
git clone git://linuxtv.org/media_build.git
cd media_build
./build
sudo make install
```

More info here https://help.ubuntu.com/community/DVB-T_(USB)

UPDATING THE KERNEL

Note: Only do this if you are running an old version of Linux. An alternative to updating the Video4Linux drivers is to update the kernel on your Linux distribution to a newer version which includes the DVB-T drivers. This can be done by first running the following.

```
sudo apt-get update
apt-cache search linux-image
```

Look at the output to find a newer kernel version for your system. Once you've found a newer kernel type the following where linux-image-newer_version should be replaced with the newer linux kernel.

```
sudo apt-get install linux-image-newer_version
```

For example according to the terminal output in this image you could type the following to update you to kernel 3.11 and automatically get the DVB-T drivers.

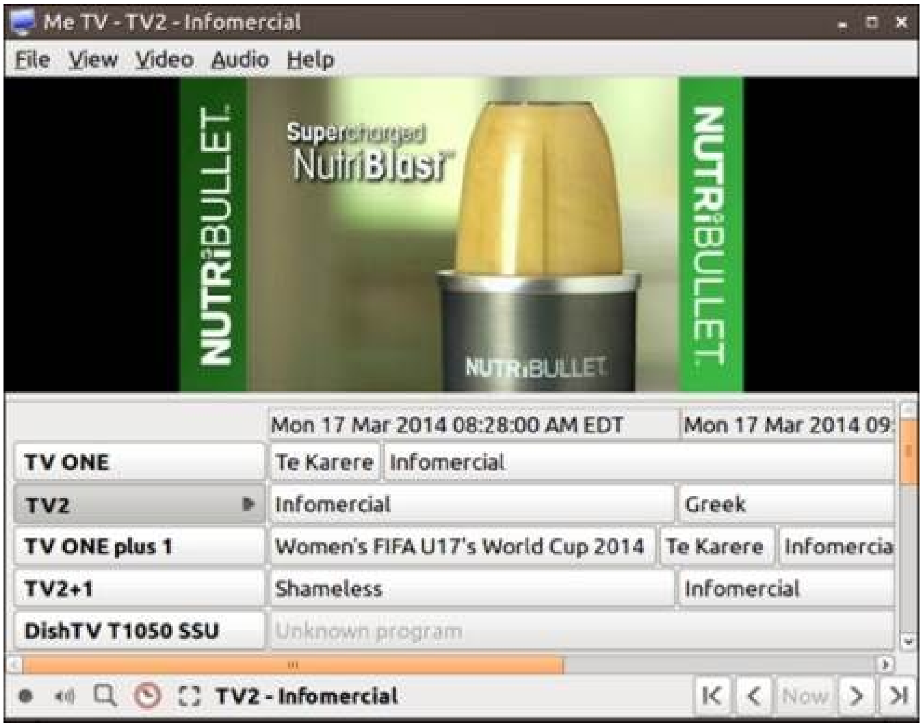
```
sudo apt-get install linux-image-3.11.0-19-generic
```

DVB-T TV WATCHING SOFTWARE

A program called Me-TV can be used to watch DVB-T TV and can be downloaded and installed with the following commands.

```
sudo apt-get update
sudo apt-get install me-tv
```

Once installed Me TV can be found in Ubuntu’s start menu under Sound & Video.



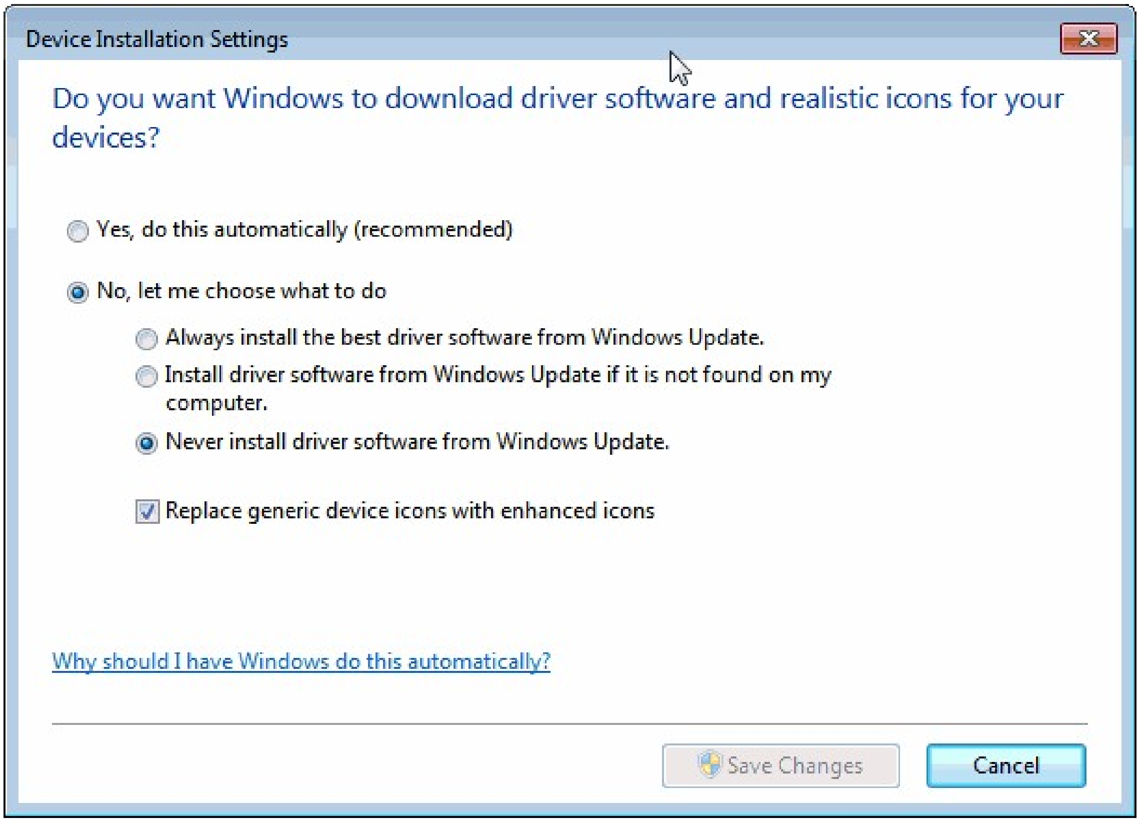
	Mon 17 Mar 2014 08:28:00 AM EDT	Mon 17 Mar 2014 09:
TV ONE	Te Karere Infomercial	
TV2	Infomercial	Greek
TV ONE plus 1	Women's FIFA U17's World Cup 2014	Te Karere Infomercia
TV2+1	Shameless	Infomercial
DishTV T1050 SSU	Unknown program	

Other more full featured software like MythTv can also be used.

DVB-T ON WINDOWS

Again, note that DVB-T TV does not exist in the USA. On Windows watching DVB-T TV is usually a simple matter of installing the CD software that comes with many dongles. However, this software is often pirated by the Chinese manufacturer and can sometimes stop working after a few weeks or months due to the software's anti piracy features.

There are also some issues with driver installation on newer versions of Windows. First for Windows 7 & 8 (perhaps vista too) you will need to turn off automatic driver installation otherwise Windows will continually replace the DVB-T driver with an older version which will not work. To do this go to the start menu and type in "Device Installation Settings" into the search bar. Click on "Change device installation settings". Change the settings to "No, let me choose what to do" and "Never install driver software from Windows Update".



Device Installation Settings

Do you want Windows to download driver software and realistic icons for your devices?

○ Yes, do this automatically (recommended)

● No, let me choose what to do

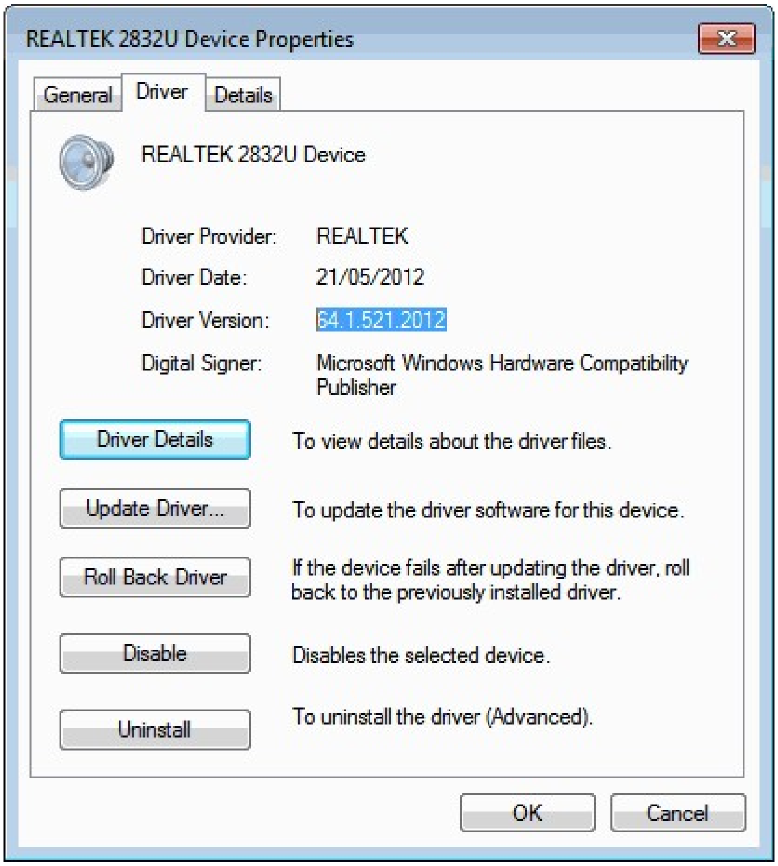
- ○ Always install the best driver software from Windows Update.
- ○ Install driver software from Windows Update if it is not found on my computer.
- ● Never install driver software from Windows Update.

☑ Replace generic device icons with enhanced icons

Why should I have Windows do this automatically?

Now you can install the DVB-T drivers from the CD. If your dongle did not come with the DVB-T CD, the DVB-T drivers can be downloaded from http://download.buytra.com/driver/SPC-0155-Driver.zip (Mirror: http://bit.ly/1igDacW). After unzipping the file, open the Drivers folder and run setup.exe to install the drivers. If Windows has already installed its own drivers (or you have the SDR drivers installed), you will need to uninstall those first and then reinstall them using the new drivers.

The correct drivers will have a Driver Version 64.1.521.2012. If an older version is shown, it probably won't work and you should reinstall the drivers. The older Windows drivers will recognize the dongle, but when searching for TV stations the dongle will simply never find any.



REALTEK 2832U Device

Driver Provider: REALTEK

Driver Date: 21/05/2012

Driver Version: 64.1.521.2012

Digital Signer: Microsoft Windows Hardware Compatibility Publisher

To view details about the driver files.

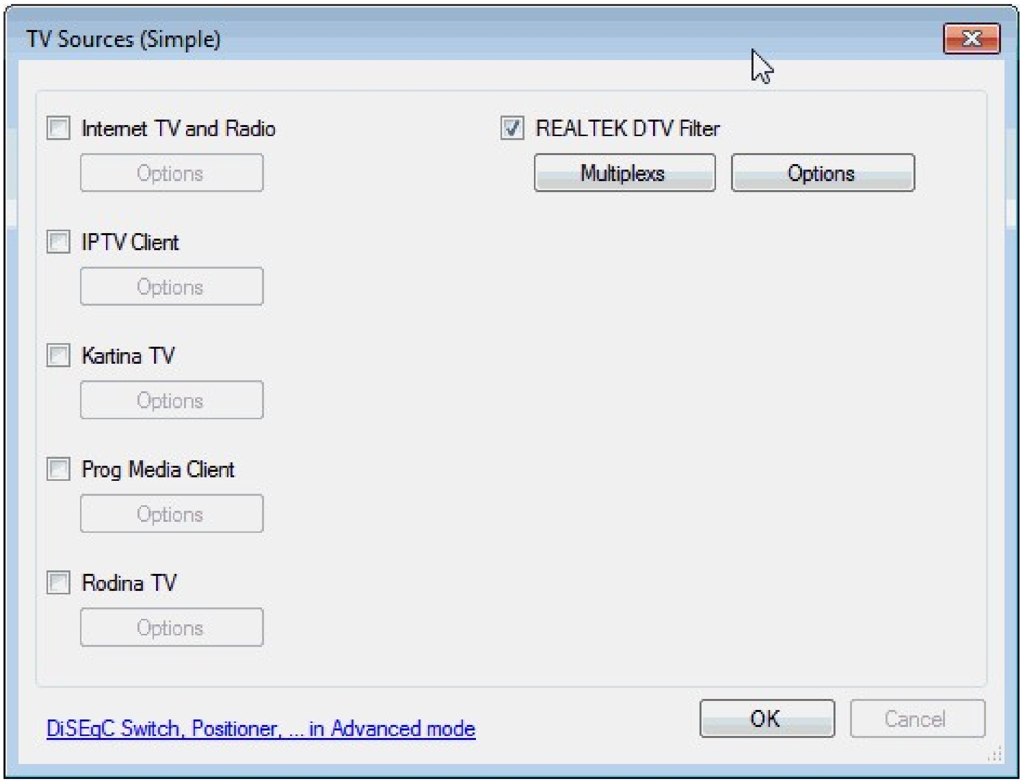
To update the driver software for this device.

If the device fails after updating the driver, roll back to the previously installed driver.

Disables the selected device.

To uninstall the driver (Advanced).

After you have installed the drivers, download the free version of Prog DVB from www.progdvb.com. After installing Prog DVB, open Prog DVB 7. Go to **Settings -> TV Sources** and check the box next to REALTEK DTV Filter.


TV Sources (Simple)

□ Internet TV and Radio
□ IPTV Client
□ Kartina TV
□ Prog Media Client
□ Rodina TV
☑ REALTEK DTV Filter

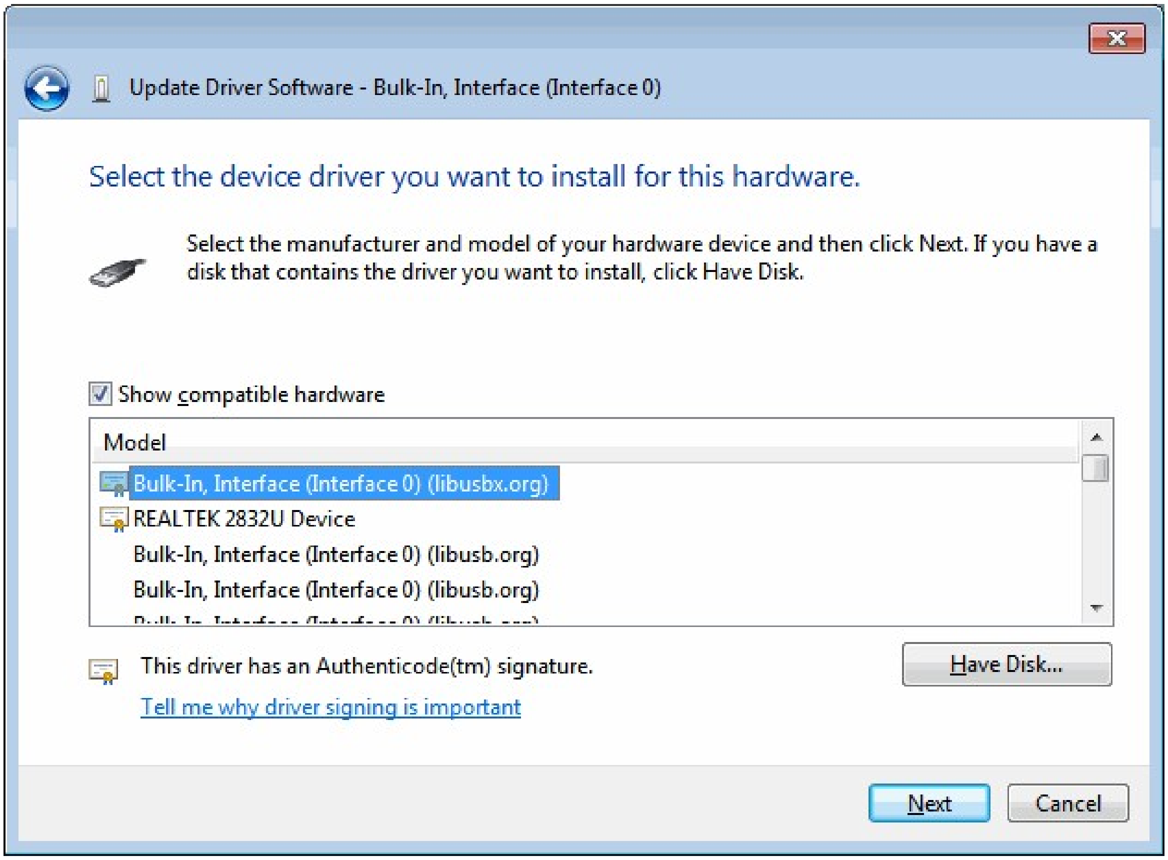
Now go click on Multiplexes and choose what Country you'd like to search for frequencies in. Press OK. Now go to **Channel List -> Channel Search -> Realtek DTV Filter**. From this point channels will be searched for and found if you have good enough reception.

QUICK WAY TO CHANGE BETWEEN DVB-T AND SDR DRIVERS ON WINDOWS VISTA/7/8

Once you have installed and uninstalled both drivers there is an easy way to switch between the DVB-T and SDR drivers within seconds. In device manager right click the RTL-SDR device and choose "Update Driver Software". (If currently installed as a DVB-T source the RTL-SDR will be under **"Sound, Video and Game Controllers -> REALTEK 2832U Device"**, and if currently installed as an SDR it will be under **"libusb (WinUSB) devices ->Bulk-In, Interface (Interface 0)".**

Sometimes the device won't show up in device manager. If you don't see the device in device manager, go to **Start -> Devices and Printers -> right click the RTL2832UHIDIR icon -> Properties -> Hardware tab -> Properties -> Driver Tab -> Update Driver**.

Click **Browse my computer for driver software -> Let me pick from a list of device drivers on my computer**. Here you can see the SDR Driver as “Bulk-In, Interface (Interface 0)” and the DVB-T driver as REALTEK 2832U Device Version 64.1.521.2012 [21/05/2012]. Choose the driver you want to change to then click next.



Update Driver Software - Bulk-In, Interface (Interface 0)

Select the device driver you want to install for this hardware.

Select the manufacturer and model of your hardware device and then click Next. If you have a disk that contains the driver you want to install, click Have Disk.

☑ Show compatible hardware

Model

Bulk-In, Interface (Interface 0) (libusbx.org)
REALTEK 2832U Device
Bulk-In, Interface (Interface 0) (libusb.org)
Bulk-In, Interface (Interface 0) (libusb.org)

This driver has an Authenticode(tm) signature.

Tell me why driver signing is important

ANTENNA GUIDE

In this section we will discuss antennas, cables and connectors. This section will not go in depth into antenna theory and will only provide a brief overview of some of the most common types of antenna. We try to provide a good starting point for further antenna research.

ANTENNA ESD SAFETY

Note that any antenna will pick up electrostatic charges over time. Large electrostatic discharges into the RTL-SDR dongle can damage it, even if it has an ESD diode in place. Be sure to properly ground your antennas and discharge any potential ESD build up by shorting the antenna to ground before using it.

For extra safety we recommend not using an outdoor antenna during a thunder storm and to disconnect the antenna from your RTL-SDR when not in use.

ANTENNA ADAPTER GUIDE

The most common RTL-SDR models use the relatively uncommon micro-coax (MCX) connectors on the dongles. These connectors are actually quite good as they work without much signal loss over a wide range of frequencies. We recommend buying dongles with the MCX connector due to their better passband. These will work better at higher frequencies for things like ADS-B. However, most connectors that fit onto coax cables and other antennas will not be MCX so you will need an adapter.

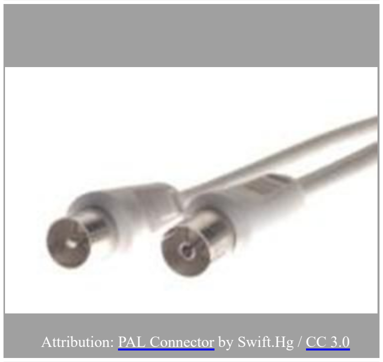
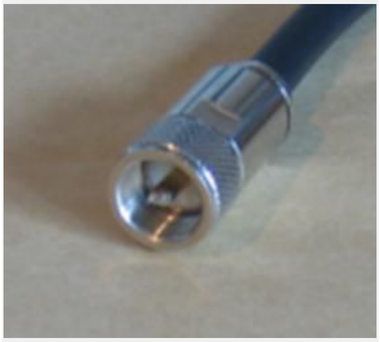
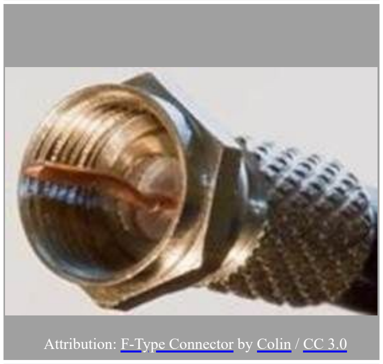
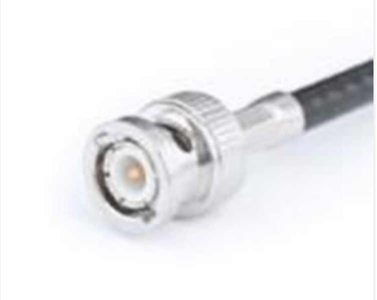
It is important to choose a good connector and adapter. Poor connectors will have a small pass band which won't work well at VHF and higher frequencies. Using an adapter to convert to another good connector such as SMA/BNC/N should introduce negligible losses.

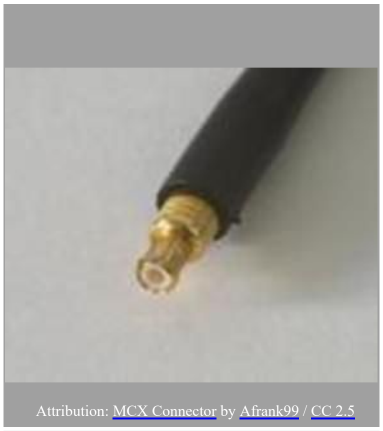
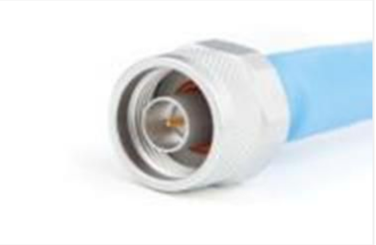
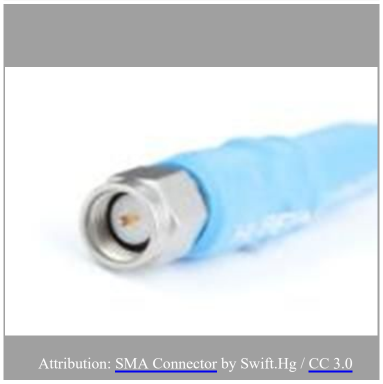
Dongles using the PAL connector will have poorer performance at high frequencies. Insertion loss expected at the ADS-B frequency (1090 MHz) is estimated to be about 1-3 dB (1.25x - 2x loss), but it can add up to more if you use PAL on both ends of the cable and start using adapters. But there is always the option of desoldering the MCX or PAL connector on the dongle and replacing it with a better one like SMA.

We also prefer the use of 'straight' adapters over 'pigtail' adapters. Pigtail adapters are the ones with a short length of coax between each end of the adapter. Pigtail adapters can be noisy and lossy if they use poor quality coax cable.

However, pigtail adapters tend to be more common and are easier to use if mounting the RTL-SDR in a box for instance. Straight adapters also add more mechanical stress to the antenna connector. If buying a pigtail adapter, we reccomend buying one with a right angled MCX connector in order to reduce mechanical strain. See below for an example of an MCX->SMA Pigtail Adapter (left) and MCX->SMA Straight Adapter (Right).

Below we show a table containing common RF connectors and their respective pass bands. The table is arranged from poorest to best.

Connector	Photo	Pass Band
PAL (a.k.a Belling Lee) (IEC 169-2)	 Attribution: PAL Connector by Swift.Hg / CC 3.0	0 - 100 MHz
UHF (SO-239/PL259)	 Attribution: UHF Connector by Peter Schwint / CC 2.0	0 - 300 MHz
F Type	 Attribution: F-Type Connector by Colin / CC 3.0	0 - 1 GHz Possibly up to 2 GHz on newer compression connectors.
BNC	 Attribution: BNC Connector by Swift.Hg / CC 3.0	0 - 4 GHz

MCX	 Attribution: MCX Connector by Afrank99 / CC 2.5	0 - 6 GHz
Type N	 Attribution: Type N Connector by Swift.Hg / CC 3.0	0 - 11 GHz. Possibly up to 18 GHz.
SMA	 Attribution: SMA Connector by Swift.Hg / CC 3.0	0 - 18 GHz Possibly up to 26.5 GHz

COAXIAL CABLE GUIDE

Coaxial cable is the cable you use to connect your radio to your antenna. Its job is to carry the signal from the antenna to the radio with as little interference and loss as possible. There exist various types of coaxial cable available for purchase. Different types of cables will have different losses, levels of shielding and impedances. Since the RTL-SDR can go into the GHz range, we recommend using at least RG6 coax cable which has a much lower loss in these high UHF frequencies compared to other common coaxial cables like RG58. RG6 is also a 75 Ohm cable which better matches the 75 Ohm input on the RTL-SDR. Most electrical stores will stock RG6 cable as it is commonly used for satellite TV installations. Good low loss 50 Ohm cabling will work just as well however, as the mismatch loss between 75 and 50 ohms is negligible at less than 0.2dB.

It is also recommended to try and reduce the overall length of coax runs. Instead, try to get the RTL-SDR dongle as close to the radio as possible using USB extension cables or remote computers. See the positioning your RTL-SDR dongle section for more information.

COAX LOSSES

In the following table we show estimated losses in dB (lower is better) for a range of common coaxial cable types over a length of 1 meter. Note that low loss cable is often much thicker and more difficult to work with.

MHz	RG 58 (50 Ω)	RG59 (75 Ω)	RG6 (75 Ω)	RG 213 (50 Ω)	RG-8 (50 Ω)	LMR-400 (50 Ω)
1	0.009	0.03	0.006	0.006	0.004	0.004
10	0.038	0.031	0.0196	0.02	0.013	0.013
100	0.133	0.102	0.056	0.069	0.042	0.04
500	0.346	0.241	0.162	0.175	0.099	0.093
1000	0.539	0.356	0.2	0.268	0.147	0.135
1500	0.708	0.449	0.216	0.348	0.186	0.168

VELOCITY FACTOR

Different coaxial cable types will also have a varying property called ‘velocity factor’. The velocity factor in the coax cable defines what the speed of light in the cable will be. This property is often used when designing coaxial based antennas as the velocity factor will change the required coax element lengths.

MATERIALS

To save costs some coax cables use aluminium braid rather than copper braid. This is fine, unless you need to solder to the shield. Aluminium is impossible to solder to, so if you are building a coax cable antenna, often it is better to spend more and get the copper braided cable.

ANTENNA GAIN/DIRECTIVITY

Antennas do not actually have ‘gain’ in the sense of an amplifier, however they can be more sensitive in certain directions (directivity). Antenna gain is measured relative to the ‘isotropic’ antenna which is a theoretical antenna that has exactly a gain of 0 dBi in a sphere around the antenna. This means that the isotropic antenna has the same sensitivity in all directions around the antenna.

As an analogy, think of a bare light bulb as an isotropic radiator - its light is not focused and can weakly illuminate a small room. Now add a reflector and a lens to the bulb and you have a torch which can focus the light strongly onto a spot much further away. Now you can strongly illuminate a spot across the room, but other spots especially ones behind the torch will be dark. A torch is similar to a directional antenna (like a Yagi) with high directivity/gain. Another example is a lantern which omnidirectionally focuses the light towards the horizon and avoids wasting light towards the sky and ground. A lantern is analogous to an antenna like a quarter wave ground plane/discone/collinear etc.

The unit dBi means decibels over the isotropic. The isotropic antenna can be said to have no directivity. All antenna gain is measured relative to the isotropic antenna. The measure of decibels is a measure of the ‘gain’ strength. Larger dBi numbers mean that your signal will be received stronger if it originates from a direction that the antenna is good at receiving.

As with most things, there are tradeoffs when choosing an antenna. You can either have a highly directional high gain antenna or an omni-directional but low gain antenna.

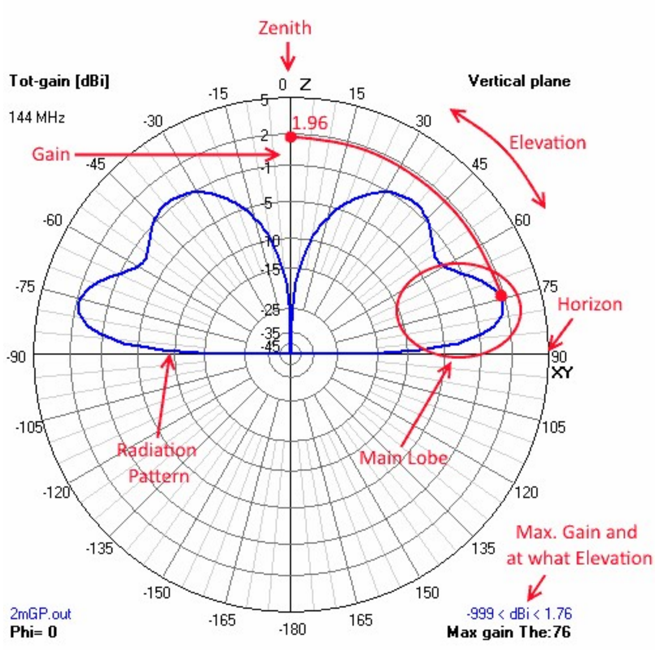
When choosing an antenna for a project, you need to think about where the signal is coming from. For example, if your signals are coming from the horizon in many directions, you may want to consider an antenna with large directivity directed towards the horizon like a quarter wave ground plane, discone, collinear or j-pole. If you want to receive a weak signal coming from one direction only, choose an antenna like a Yagi which has high gain in the direction it is pointing.

RADIATION PATTERNS

Radiation patterns show the antenna directivity, or in other words in what directions the antenna will receive best. In the example antennas shown below in the Example Antennas section we show 3D and 2D total power radiation patterns. The more red/pink the colour and the further out the pattern extends in the 3D images, the more gain in that particular direction.

Below we show a labelled example of a 2D radiation pattern plot drawn in the vertical plane. The values on the outside of the circle represent elevation. The horizon is defined to be at 90 degrees elevation and straight up into the sky (Zenith) is defined as 0 degrees elevation. The vertical axis is the gain in dBi. In the plot below we can read that this antenna has a maximum gain of 1.76 dBi at an elevation of 76 degrees. To read the gain at a particular elevation, follow the circular arc back to the vertical axis.

The ‘main lobe’ of an antenna radiation pattern is the lobe that contains the maximum gain. In the example plot below we have highlighted the main lobe for this radiation pattern.



ANTENNA DESIGN FREQUENCY

You also need to think about what frequency you are interested in receiving. Most antennas will receive well at the design frequency and near it, but poorly at any other frequency. An exception are wideband antennas like the discone or scantenna.

For most antennas the length of the elements determines the design frequency. Radio travels in waves that have lengths measured in meters. Low radio frequencies such as the LF/MF/HF bands (0 - 30 MHz) have long wave lengths. Very and Ultra high frequencies such as VHF/UHF (50 MHz+) have shorter wavelengths.

The size of the elements on an antenna are usually proportional to the wavelength. Therefore, HF antennas need to a large and VHF/UHF antennas can be much smaller. The wavelength in meters of any frequency measured in MHz can be easily calculated with the following formula.

wavelength (m) = 300 / frequency (MHz)

Broadband antennas are desirable as they receive all frequencies. However, in some cases it may be more desirable to have a narrowband antenna which only receives well the frequencies you are interested in. This is especially the case with a wideband device like the RTL-SDR. The reason is that the RTL-SDR is very sensitive to out of band interference. So, if your antenna attenuates out of band signals, performance of the in band signals can be much better.

STANDING WAVE RATIO (SWR)

The SWR of an antenna is a measure that says how well matched the antenna impedance and coax or radio impedance is for a particular frequency. Typically coax cable and radio inputs are standardized at 50 or 75 Ohms. The antenna impedance should be the same to get a good SWR. If an antenna has an impedance of say 450 Ohms, a matching 9:1 transformer should be used to bring its impedance down to 50 Ohms.

A poor impedance match and thus bad SWR means that most of the received signal by the antenna will not make it through the coax. Instead the signal will be reflected back causing interference to the signal, kind of like an echo. As an analogy, think about how listening to an outdoor speaker with an echo can be hard to understand. The same thing happens with mismatched impedances.

A good SWR means that most of the signal makes it through the cable without interference. A perfect SWR ratio is 1:1 and a really bad ratio might be 100:1 or higher. In practice a SWR under 10 is usually sufficient. SWR can be calculated by dividing the antenna impedance by the coax/radio impedance. For instance a 450 Ohm antenna connected to a 50 Ohm coax cable would have a SWR of 450:50 = 9:1.

Antenna impedance and therefore SWR can vary wildly with frequency. Non-broadband antennas will only likely have a small range of frequencies that have a good SWR. Broadband antennas are designed to have a low SWR over a large frequency range.

Note that SWR says nothing about antenna gain. The SWR may be good at a certain frequency, but the gain of the antenna may be poor at that frequency. Where SWR matters is the signal loss that the mismatch creates. This online calculator at http://www.csgnetwork.com/vswrlosscalc.html can be used to calculate the loss in dB for a particular SWR value. Here is a table of some SWR values to give you an idea of the loss you could be dealing with. From this table we can see that if we were to use 50 Ohm coax cabling with the 75 Ohm input RTL-SDR, we would have a SWR of 1.5 and so we would lose 0.177 dB worth of signal.

SWR	Loss (dB)
1.5	0.177
3	1.249
5	2.553
10	4.807
20	7.413
100	14.07
500	20.99
1000	23.99

Ham radio enthusiasts who transmit using high powers are particularly concerned about SWR. This is because a bad SWR means that large amounts of power used for transmitting will get reflected back into the radio causing it to break, or simply causing massive amounts of transmit power to be wasted. For reception a high SWR is not so dire as to break the radio, but a high SWR can still cause significant signal loss.

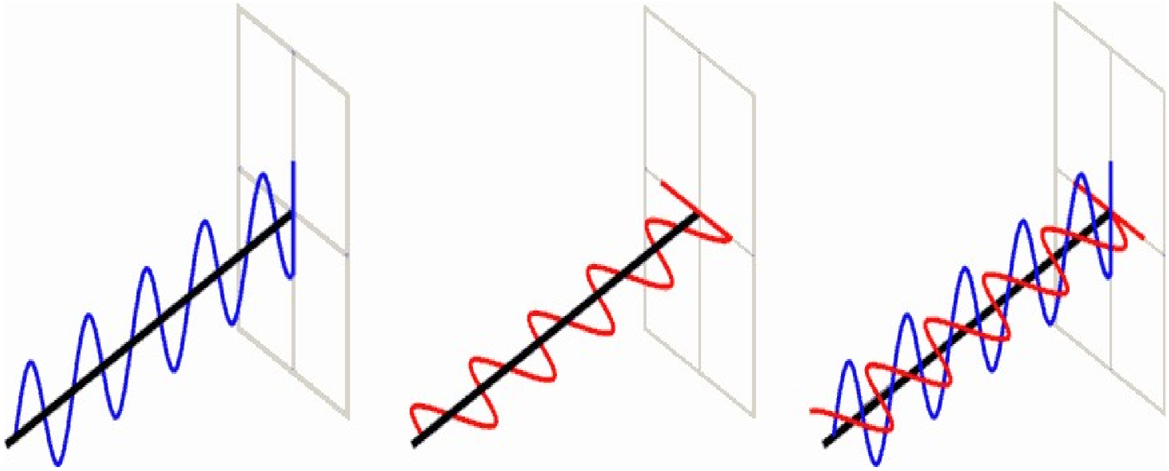
ANTENNA POLARIZATION

Radio signals can be transmitted at different polarizations (orientations). A vertically polarized signal must be received with a vertically polarized antenna and a horizontally polarized signal must be received with a horizontally polarized antenna. If you use the wrong polarization then the signal will be severely attenuated.

Below we show an image illustrating vertically, horizontally and circular polarized signals from left to right.

Notice how a circularly polarized signal is a vertically and horizontally polarized signal that are out of phase and added together. Depending on the phase of the signal, a circularly polarized signal can be Right Hand Circularly Polarized (RHCP) or Left Hand Circularly Polarized (LHCP). Circularly polarized antennas can receive both horizontal and vertically polarized signals.

A great tutorial on antenna polarization can be found here http://sv1bsx.50webs.com/antenna-pol/polarization.html.



Left: Vertically Polarized, **Middle:** Horizontally Polarized, **Right:** Circularly Polarized

ANTENNA RECIPROCITY

When looking at information about antennas, most articles and guides will refer to the transmission gain and SWR of the antenna. However, antennas have something called reciprocity which simply means that the properties will be the same for reception or transmission. So an antenna that has a low SWR at 144 MHz and high gain towards the horizon when transmitting will also have the same properties when receiving.

ANTENNA POSITIONING

Antennas normally work best when they are placed up as high as possible without obstructions. If an antenna is obstructed by for example a house, tree or hill, its reception from that direction will be greatly reduced. The usual place to mount an antenna is on the roof of your house using a mount. When mounting on the roof you must be weary of weatherproofing and grounding in case of lightning strikes.

Antennas can also be mounted in the attic or roof space of many houses without significant reception degradation. Mounting in the attic also has the advantage of weatherproofing and negating lightning strike risks. The most commonly used roof material in many countries is asphalt shingles which won't block RF signals significantly. However, metal roofs or roofs with concrete tiles with metal rebar or grids in them can degrade signals much more.

The height of the antenna will also affect the radiation pattern. Antennas placed up higher often have better gain directed towards the horizon.

COMMON MODE CHOKES

Almost any antenna that uses coaxial cable should use a common mode choke near the antenna. The common mode choke was mentioned previously in this book under the Improvements to the stock antenna section. A common mode choke will essentially ensure that the coax feedline does not become part of the antenna itself. Without the choke the coax cable can act as part of the antenna, drastically skewing the radiation pattern to something undesirable.

All that is needed for a common mode choke is to wind a bit of the coax cable near the antenna in a loop a few times around a cylindrical object. It will work even better if you use a ferrite core, but with larger coax cables big ferrites may be expensive and hard to find, so an air winding will have to do.

Additional Information

http://myantennas.com/wp/tech-info/about-cmc/

http://www.w8ji.com/ground_plane_verticals.htm

VHF/UHF EXAMPLE ANTENNAS

Below we describe some antennas that are commonly used with the RTL-SDR for radio scanning. Also shown are simulated radiation patterns and SWR plots which should give you some sort of idea on the performance of each antenna. The

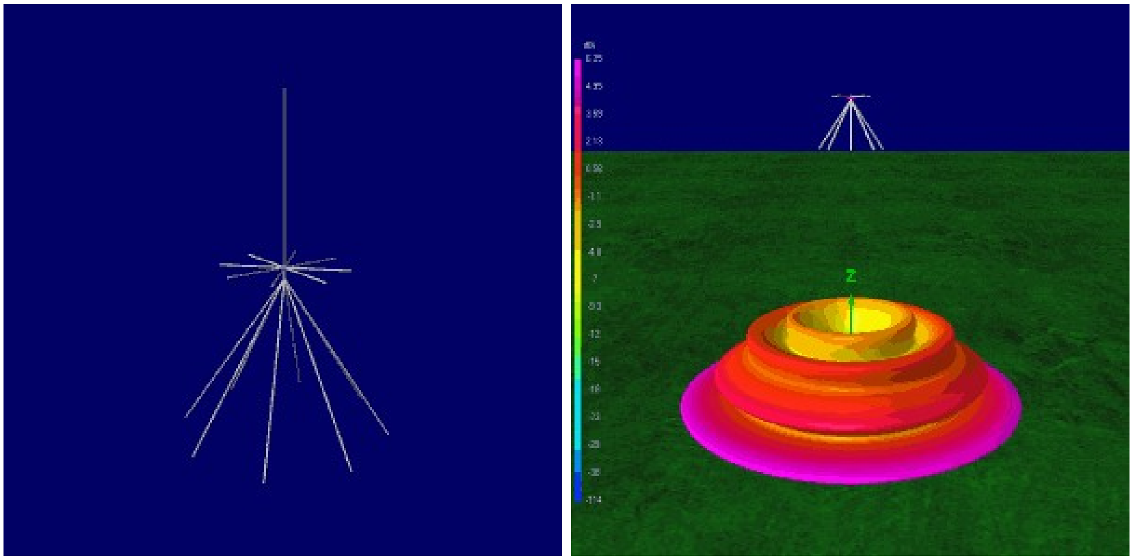
NEC simulation files used in the examples below can all be downloaded from http://bit.ly/1CfSNxT.

Most antennas shown below as easily constructed at home. Here is an excellent guide on the construction of some of the antennas mentioned in this section http://files.radioscanner.ru/files/download/file311/practical_antenna_design.pdf.

The radiation pattern and VSWR images used below were created using the free software 4NEC2 which can be downloaded from http://www.qsl.net/4nec2/. We have brief tutorial on using 4NEC2 after this section.

Note that the side by side images in this section may be easier to view in landscape mode, or with less columns on the Kindle e-reader.

DISCONE



Left: Image of a typical discone with whip, **Right:** 3D Radiation Pattern at 144 MHz and 5m high

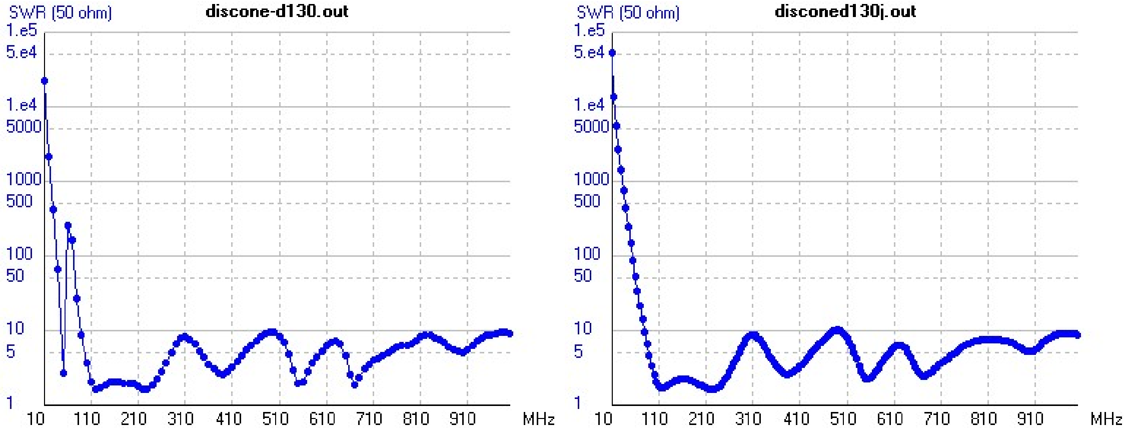
Name: Discone
Polarization: Vertical
Broadband: Yes
Uses: General purpose antenna good for almost all frequencies.

The discone is typically recommended as the overall most useful antenna for the RTL-SDR. This is because a discone has a very wide receivable bandwidth and can receive signals omnidirectionally with good gain/directivity towards the horizon. The larger the discone is, the lower the frequency it can receive, however some discone designs opt to place a whip on top to lower the SWR at some lower frequencies.

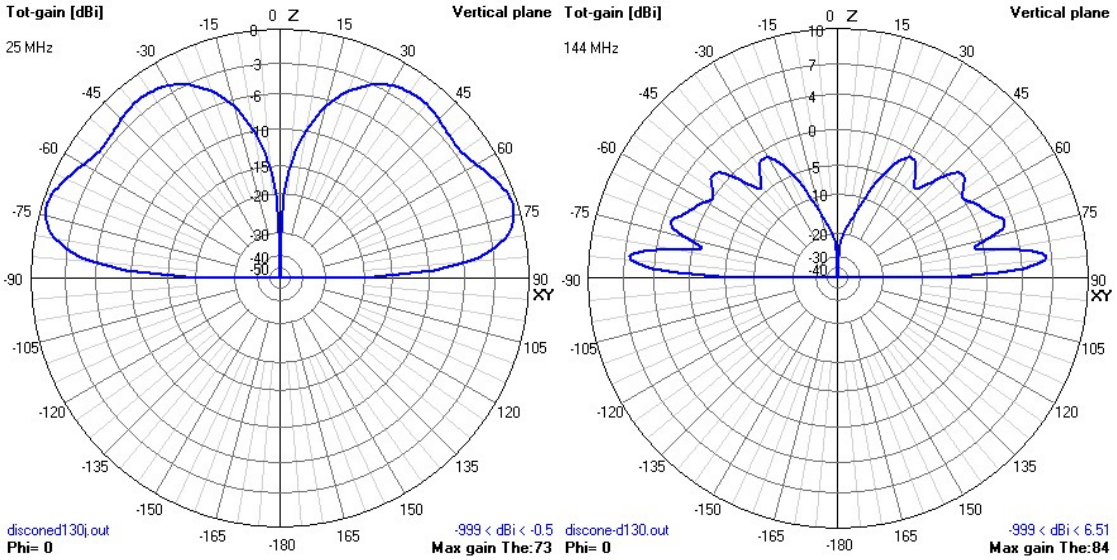
Commercially constructed discones can be found cheaply on Amazon (see our buy RTL-SDR page at http://www.rtl-sdr.com/buy-rtl-sdr-dvb-t-dongles/) for around $50 USD. Most local ham radio stores should also stock them. Higher priced models will be made of better more weather proof materials or will be larger and able to receive lower frequencies.

In the radiation pattern images below we show the patterns for a discone rated from 125 - 1300 MHz over various frequencies. The results show that the radiation pattern can vary significantly as the frequency changes. At higher frequencies there is significant ‘lobeing’, which just means that the radiation pattern becomes spiky or non smooth. In all cases, most of the radiation pattern is directed towards the horizon where most signals of interest originate from, but notice that as the frequency rises more and more radiation gets directed towards the sky, so discones tend to perform more poorly at higher frequencies.

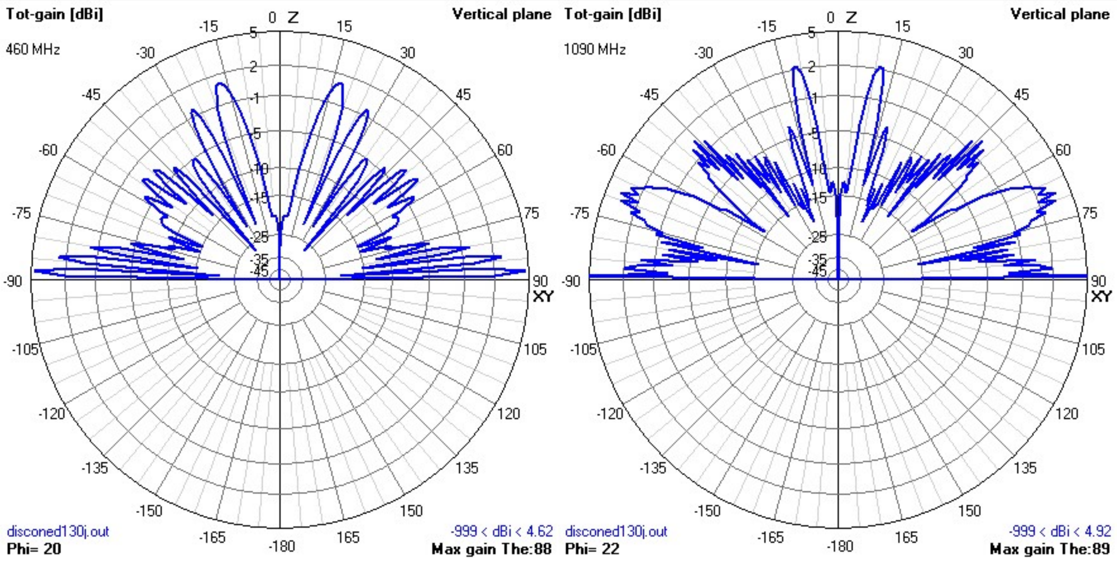
In the SWR graphs below we can also see that the SWR stays fairly constant, usually remaining below 10 for all frequencies above the lowest design frequency of 125 MHz. The effect of the whip can be seen in the SWR plot where the discone with the whip has a dip in SWR at lower frequencies.



Left: SWR with whip. **Right:** SWR without whip

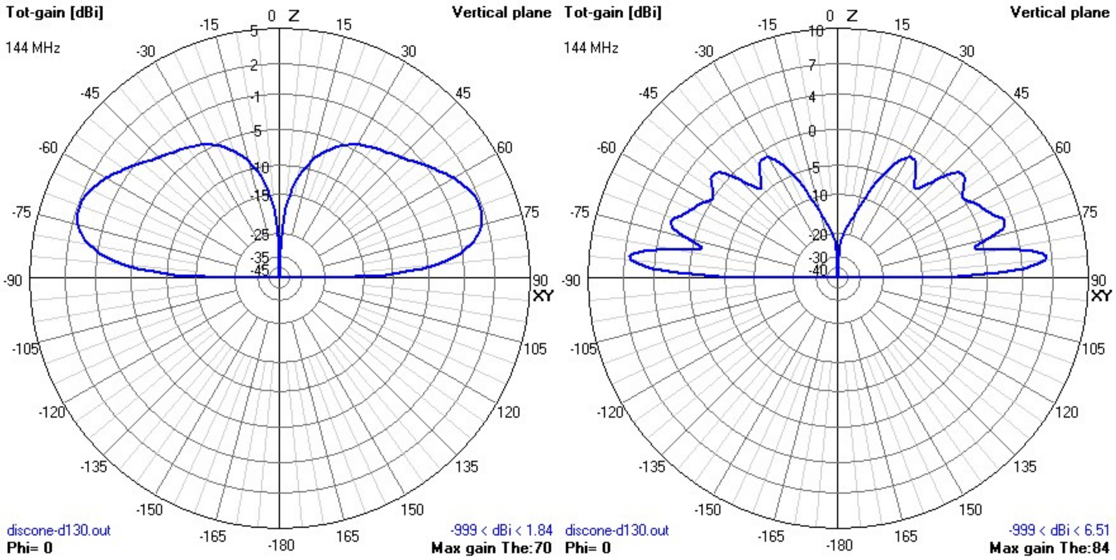


Left: 25 MHz. **Right:** 144 MHz

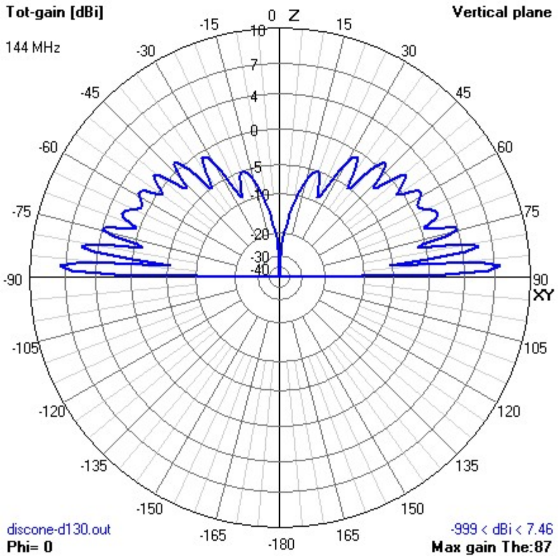


Left: 460 MHz. **Right:** 1090 MHz

Below the radiation patterns for a discone receiving 144 MHz and placed at 1m, 5m and 10m heights are shown. Increasing the height increases the gain and directivity towards the horizon.



Left: 1M Above Ground. **Right:** 5M Above Ground



Above: 10M Above Ground

Here is an example of a home made discone http://helix.air.net.au/index.php/d-i-y-discone-for-rtlsdr/ and here is a homemade build tutorial http://www.instructables.com/id/Discone/.

SCANTENNA

Name: Scantenna
Polarization: Vertical

Broadband: Yes
Uses: General purpose antenna good for almost all frequencies.

The scantenna is an antenna similar to the discone in that it has a very wide receivable bandwidth. Many people often state that the scantenna performs better than a discone. This is probably because the scantenna has greater gain out towards the horizon and is thus a better receiver for most common signal sources.

The most commonly bought scantenna has a frequency range of 30 - 1300 MHz. Beware that the scantenna is quite large measuring in at 144 x 19.5 x 8.6cm (56.8 x 7.7 x 3.4 inches).

See the RTL-SDR.com products page http://www.rtl-sdr.com/buy-rtl-sdr-dvb-t-dongles/ to see links on where to buy a Scantenna.

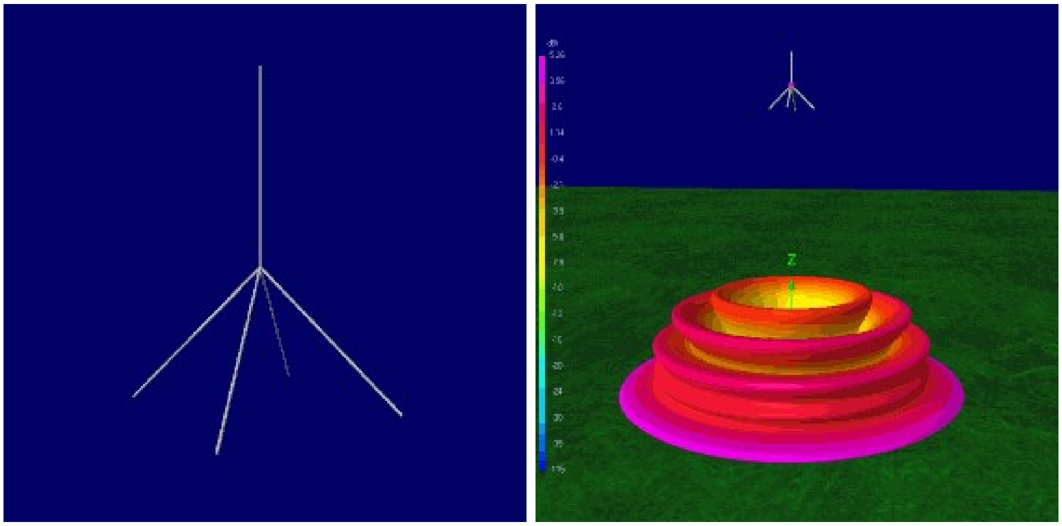
PLANAR DISK ANTENNA

Name: Planar Disk Antenna
Polarization: Vertical
Broadband: Yes
Uses: General purpose antenna good for almost all frequencies.

A planar disk antenna is another antenna with a very wide receivable frequency range. A planar disk antenna is very simple to build as it consists of only two circular metal discs placed next to each other with the coax cable soldered to each disk at the place the two disks touch. The lowest frequency receivable is determined by the 1/4 wave disk width. A disk with a 50cm wide width will receive down to around 140 MHz.

We don’t know of any planar disk antennas for sale, but they are very easy to build out of ordinary pizza pans. A nice tutorial can be found at http://www.wa5vjb.com/references/PlanarDiskAntennas.pdf.

QUARTER WAVE MONOPOLE GROUND PLANE



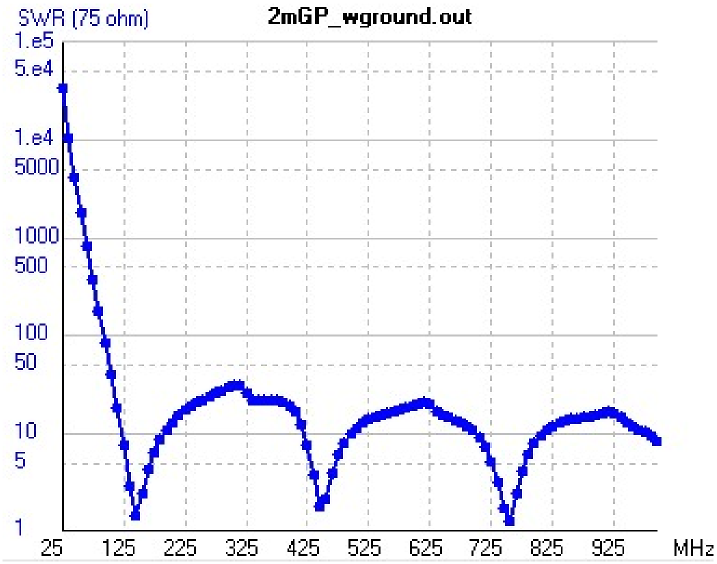
Left: Image of a typical 1/4 wave groundplane, **Right:** 3D Radiation Pattern at 144 MHz and 5m high

Name: Quarter Wave Ground Plane
Polarization: Vertical
Broadband: No
Uses: General narrow band antenna used in many situations. Good for ADS-B as antenna is small and has good directivity towards the horizon.

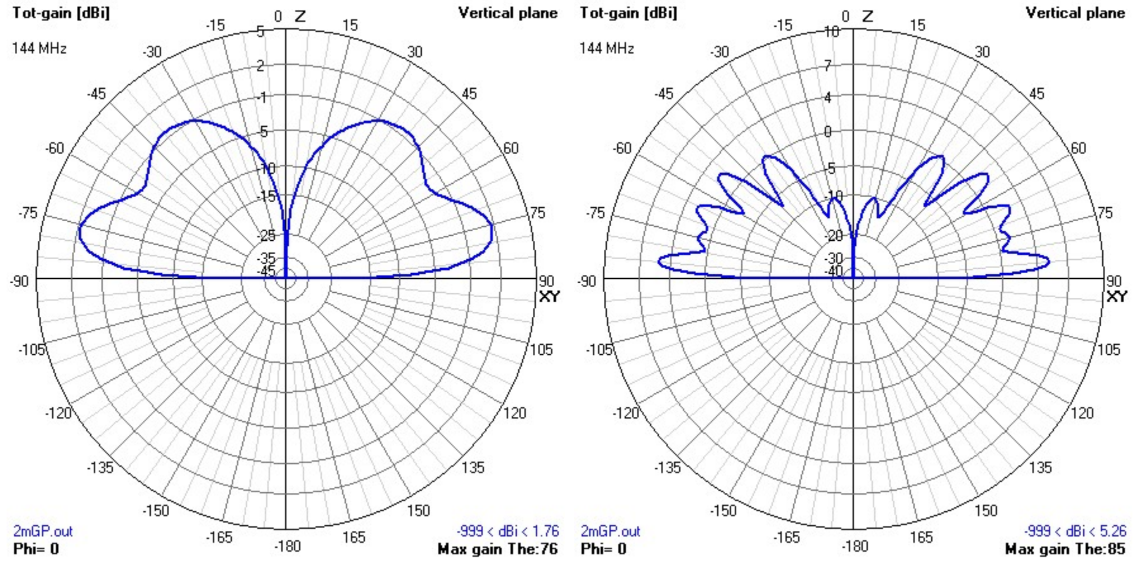
The quarter wave ground plane is essentially a straight wire/element cut to a quarter of the wave length of the design frequency with several spokes of the same length as the whip coming out the bottom. The straight wire forms the antenna and the spokes are the ground plane. It is a vertically polarized antenna. Theoretically, a quarter wave groundplane with good ground plane (enough radials) is equivalent to a dipole.

A calculator for calculating quarter wave ground plane element lengths for specific frequencies can be found at http://www.csgnetwork.com/antennagpcalc.html. Information on construction of an ADS-B quarter wave ground plane antenna can be found at http://www.atouk.com/wordpress/?wpdmact=process&did=Ni5ob3RsaW5r. Here is another page showing the design of this antenna http://ketil.com/la2dna/antenner/PMR446_ground_plane_antenna.htm.

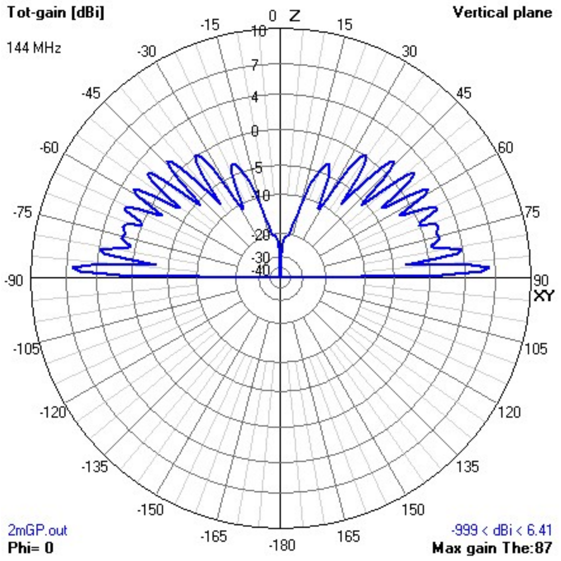
The SWR of a quarter wave ground plane is low at its design frequency and at a few other higher frequencies. In our example antenna with design frequency of 145 MHz, the SWR is minimum at 145 MHz.



The radiation pattern at the design frequency has a main lobe that directs most of its gain out towards the horizon. As the height of the antenna rises as shown in the radiation plots below, the gain towards the horizon also increases. At a low elevation most of the gain is directed a few degrees above the horizon.



Left: 1M Above Ground. **Right:** 5M Above Ground



Above: 10M Above Ground

WHIP

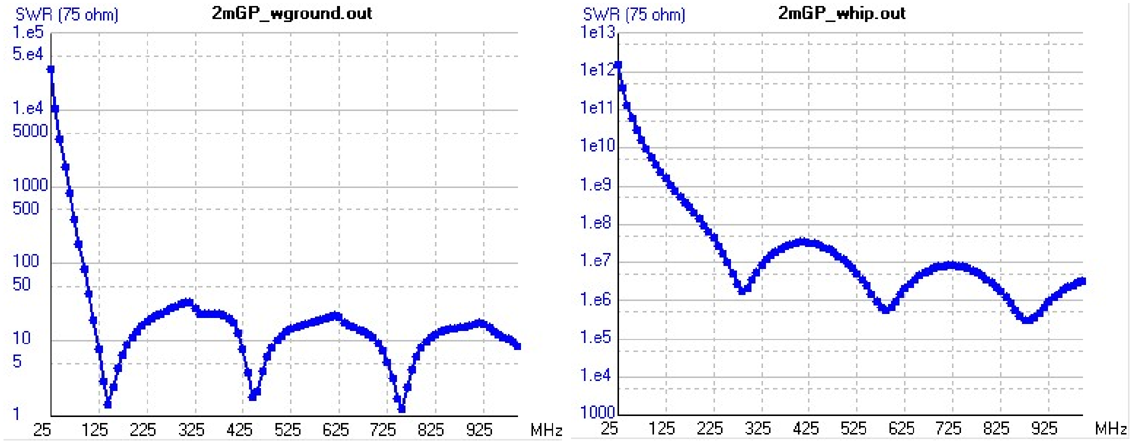
Name: Whip/Monopole
Polarization: Vertical
Broadband: No
Uses: Often used on portable radios.

The simplest whip antenna is just a vertical antenna like a quarter wave ground plane without a proper ground plane. It is then just essentially a vertical wire. Removing the ground plane of a quarter wave ground plane antenna will move the SWR minimum up to a higher frequency and cause the lowest SWR point to not be as low as with a proper ground plane. To compensate and move the minimum SWR point back down to a lower frequency the whip should be made longer than a quarter wave.

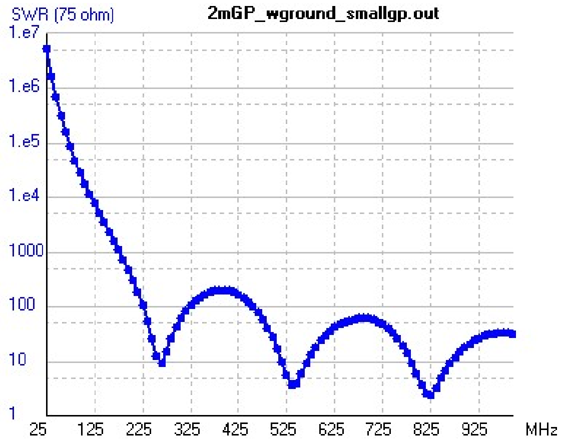
Whip antennas (such as the one shipped with the standard RTL-SDR) are usually designed with a magnetic base so that they can be placed on metallic objects such as a car roof to act as a ground plane. Often placing the magnetic mount on a metallic ground plane can make the difference between receiving and not receiving a signal. On hand held radios that have a whip, the radio chasis and your body touching the chasis act as the ground plane. The most versatile whip antenna is a telescopic whip, because it's length can be adjusted to optimise the SWR for the particular frequency you want to listen to. The larger the ground plane, the lower the SWR, and the more that the whip begins to approximate a 1/4 wave ground plane antenna, or dipole.

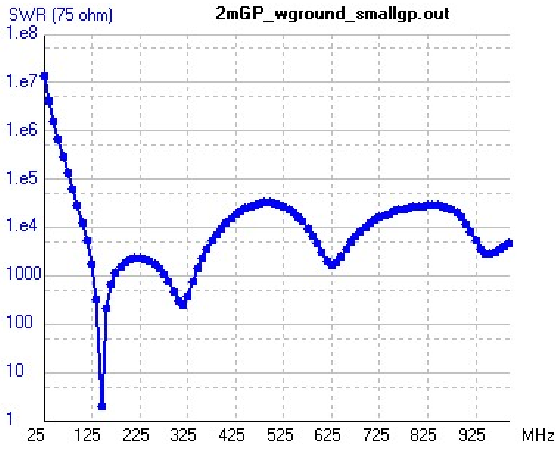
Below we compare the SWR of a quarter wave ground plane antenna vs one that has had its ground plane removed (a whip). In the whip the SWR at the design frequency of 145 MHz has significantly increased. But then even just by adding a small ground plane with a 1 cm radius as shown in the third graph, the SWR can be significantly reduced. Adding increasingly larger ground planes further reduces the SWR.

A trick used in many whip antenna designs is the inclusion of an inductive loading coil. This is the coil that is seen on the base or middle of some antennas. The loading coil is used "electrically lengthen" the antenna in order to reduce the SWR at the desired design frequency. By using loading coils, antennas that have good SWR and are much smaller than a quarter wavelength can be created. As shown in the SWR plot below, the inclusion of a loading coil of the correct value significantly reduces the SWR at the design frequency.



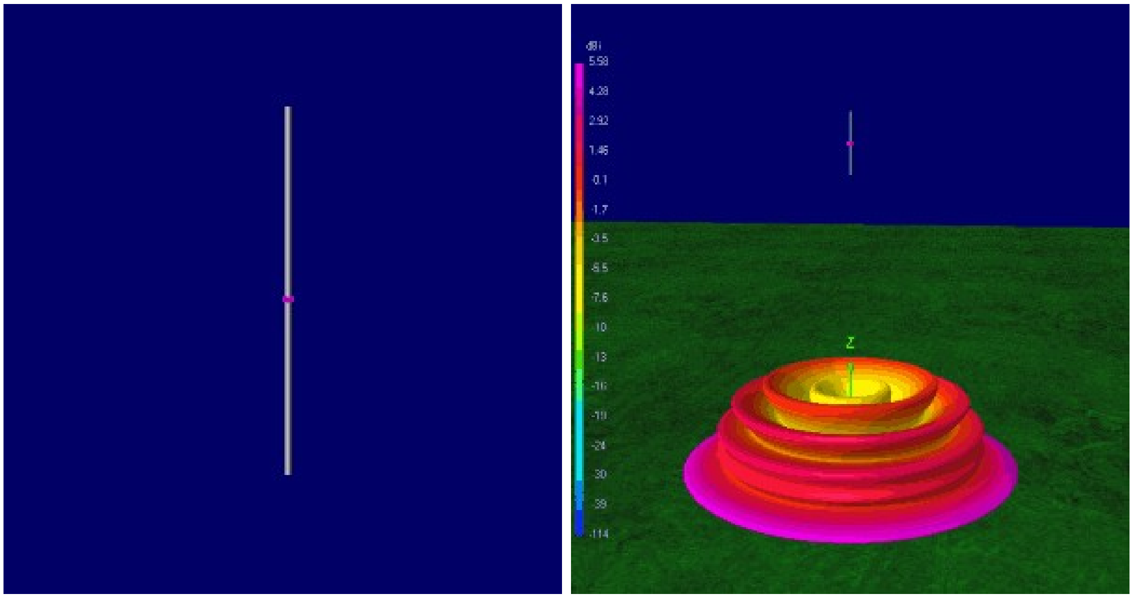
Left: SWR for quarter wave ground plane. **Right:** SWR for a whip with no ground plane.





Left: Whip with ground plane of 1cm. **Right:** Whip with ground plane of 1cm and 2.23 uH loading coil at the base.

DIPOLE



Left: Image of a typical dipole, **Right:** 3D Radiation Pattern at 144 MHz and 5m high

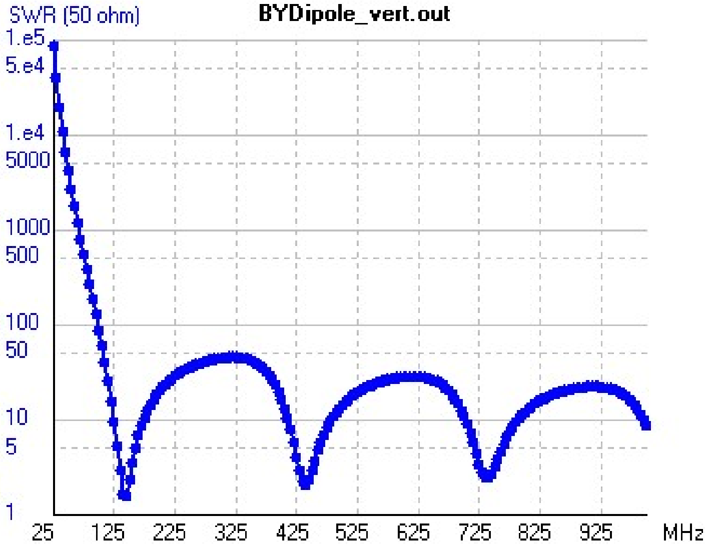
Name: Dipole
Polarization: Vertical/Horizontal (Depends on orientation)
Broadband: No
Uses: Used in bunny ear antennas for TVs, also good as a general antenna. Easiest to use in the horizontal orientation for horizontally polarized signals.

A dipole is essentially two wire elements pointing away from each other. The elements are not connected at the centre. One element connects to the centre of the coax and the other connects to the shield. It is a half wave length long, with each element being a quarter wavelength long.

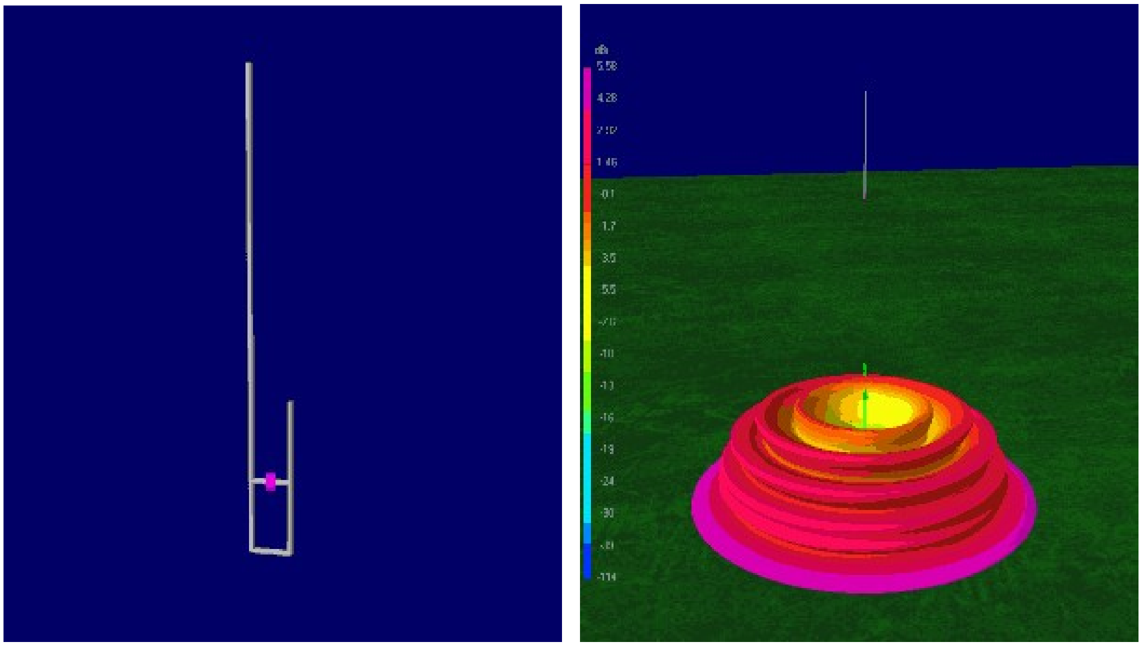
The dipole can be made horizontally or vertically polarized by placing it in the horizontal and vertical orientations respectively. The radiation pattern is greatest in the direction perpendicular to the elements. A commonly found VHF dipole in everyday life is the bunny ears antennas which are common on some older TVs. As TV transmissions are broadcast with horizontal polarization, bunny ears are arranged horizontally, but they can just as easily be oriented vertically for receiving other signals.

More information about dipoles can be found at http://www.radio-electronics.com/info/antennas/dipole/dipole.php. Here is a dipole length calculator http://www.angelfire.com/mb/amandx/dipole.html. Information on a very simple ADS-B dipole can be found at http://antirez.com/news/46.

We don't show the dipole radiation patterns here because compared to the quarter wave ground plane the dipole has a very similar, almost identical radiation pattern and SWR. A dipole is essentially equivalent to a quarter wave ground plane with a perfect ground plane. However, the disadvantage to a dipole is that it needs to be larger and it is more difficult to mount the feedline if using it vertically polarized. The feedline should run perpendicularly away from the dipole to avoid altering the radiation pattern.



J-POLE / SLIM JIM



Left: Image of a typical j-pole, **Right:** 3D Radiation Pattern at 144 MHz and 5m high

Name: J-Pole
Polarization: Vertical
Broadband: No
Uses: Good all rounder with an easy way to match SWR.

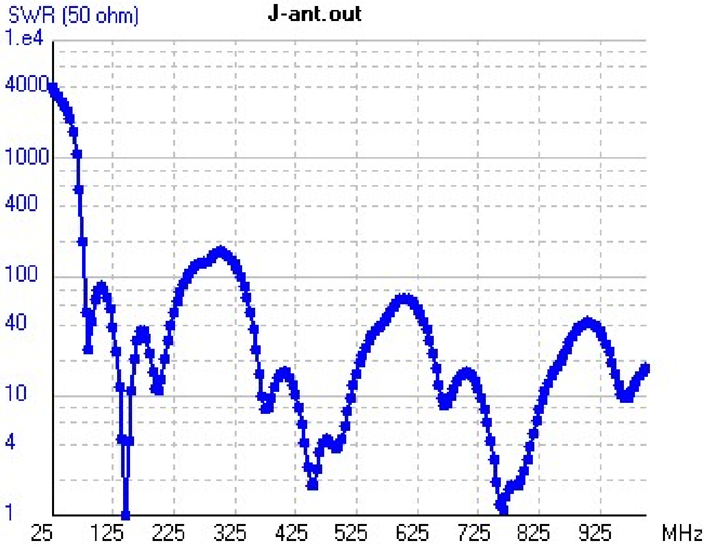
The j-pole antenna is named as such because it looks like the letter ‘J’. A j-pole is almost equivalent in terms of radiation pattern gain performance when compared to a dipole or quarter wave ground plane. The main difference is in the feed line connection. In a j-pole the feedline connects across the folded part and the height of the connection can be adjusted to tune the antennas SWR (you’ll need an SWR meter to be able to easily do this). J-poles can also be somewhat easier to handle compared to dipoles and can even be built out of twin lead wire for a collapsible portable version.

A j-pole schematic and calculator can be found at http://www.hamuniverse.com/jpole.html.

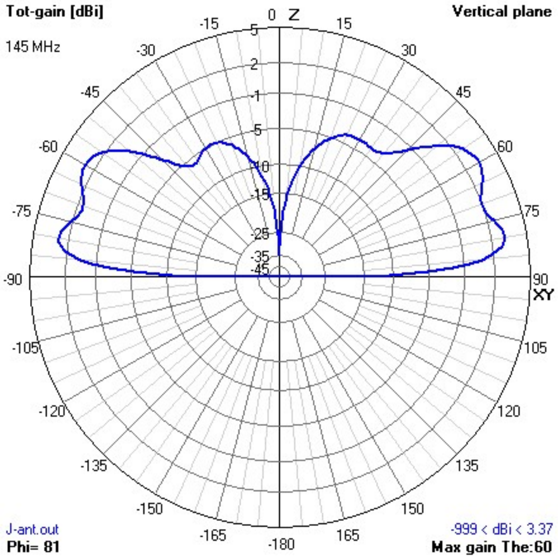
The slim jim is an antenna that is very similar to the j-pole, except that it has an extra fold on the topside. Slim jims have a radiation pattern that slightly favours signals coming from the horizon a little more and so are usually a better choice for most applications.

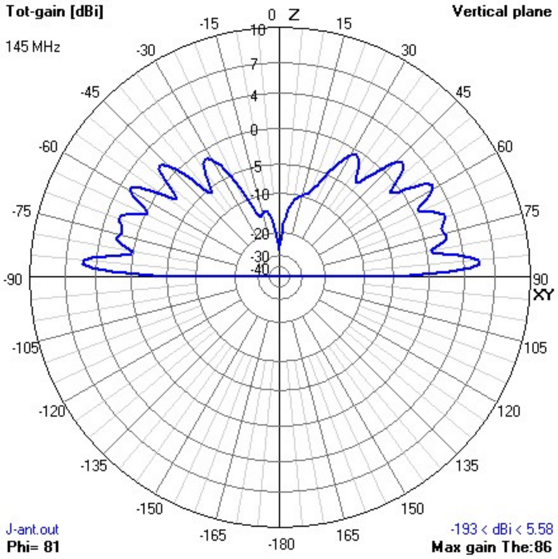
Slim jims and j-poles can easily be constructed out of twin lead wire. Here is a tutorial example http://www.g4aym.org.uk/projects/project3/page0.htm. Here is a slim jim calculator http://www.m0ukd.com/Calculators/Slim_Jim/. And here an example of a slim jim cut for AIS frequencies http://nmearouter.com/docs/ais/slimjim.jpg.

The j-pole in the example images shown above and in the graphs below is designed for 145 MHz. In the graphs below the SWR graph is somewhat similar to the quarter wave ground plane and dipole graphs, with the exception of some additional bumps.

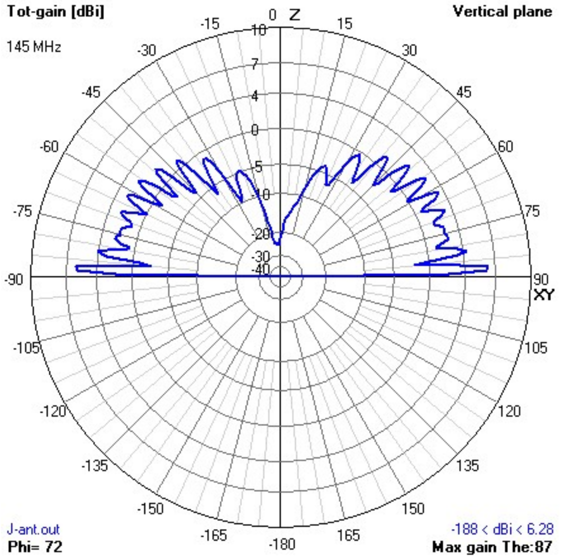


Here we show radiation patterns for a j-pole at 144 MHz for 1m, 5m and 10m for a frequency of 145 MHz. As you can see the radiation patterns are very similar to the dipole and quarter wave ground plane.



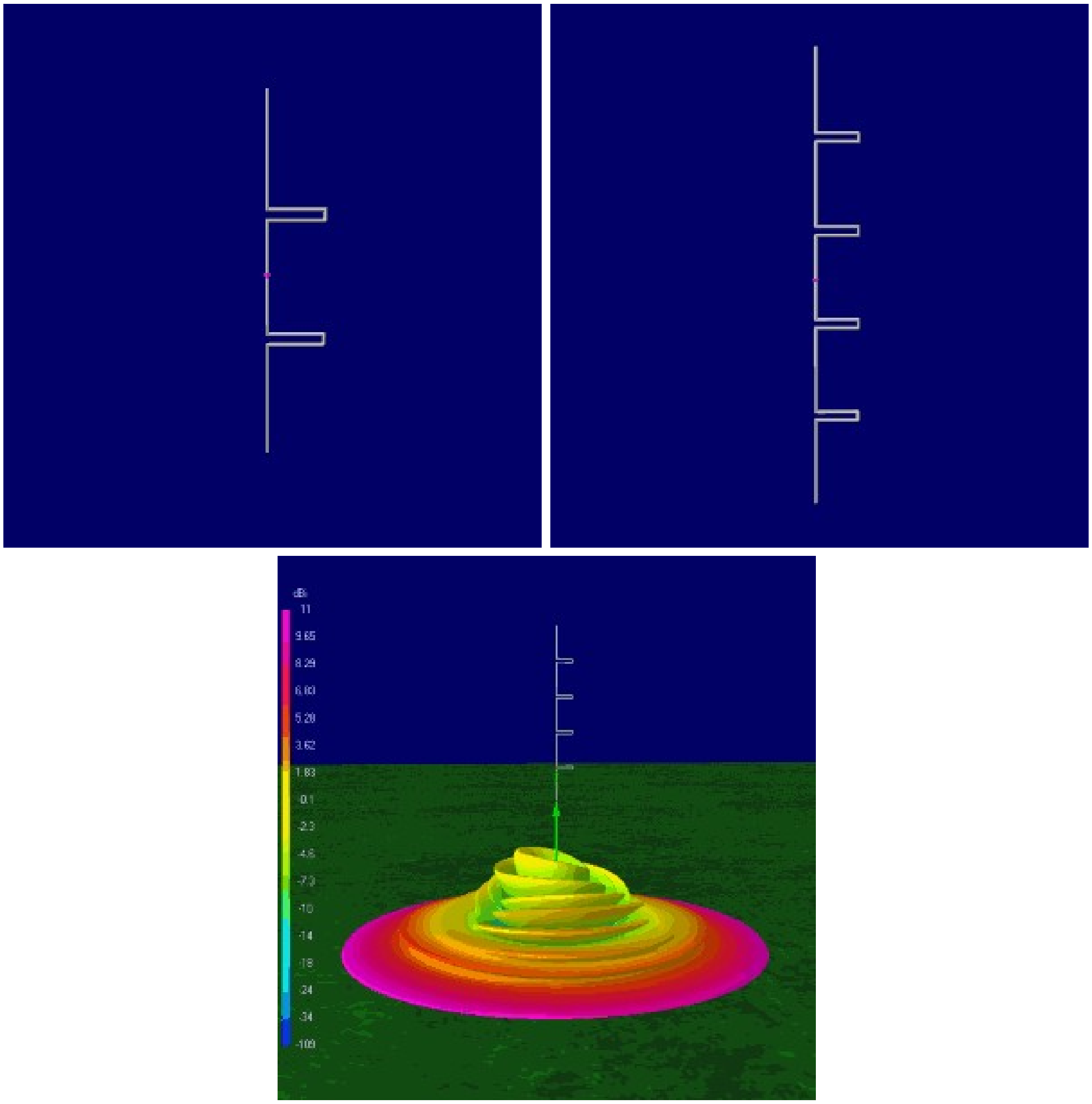


Left: 1M Above Ground. **Right:** 5M Above Ground



Above: 10M Above Ground

COLLINEAR



Left: Image of a collinear with one stack, **Middle:** Image of a collinear with two stacks, **Right:** 3D Radiation Pattern at 144 MHz and 5m high

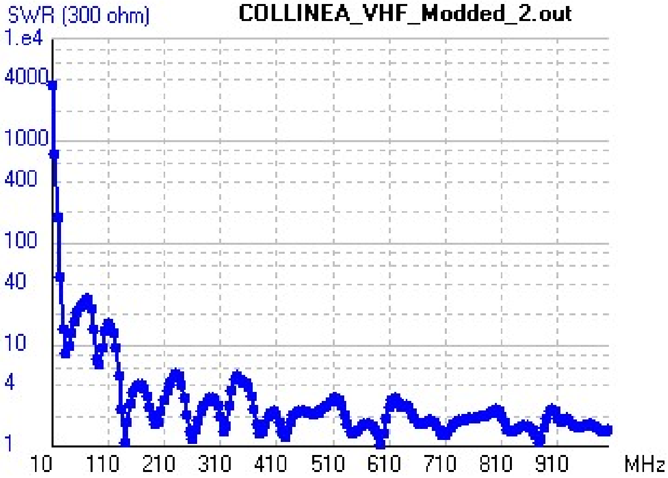
Name: Collinear
Polarization: Vertical
Broadband: No. Usually low SWR over most frequencies, but radiation pattern directs more energy towards the sky away from design frequency.
Uses: Great if you need extra gain out towards the horizon. Good for ADS-B, AIS etc.

Collinear antennas can have very high gain directed towards the horizon which make them ideal for receiving weak signals from long distances. They work very well for ADS-B and AIS reception as well as terrestrial voice communications frequencies. They are easily built out of coax cable or copper wire/pipe.

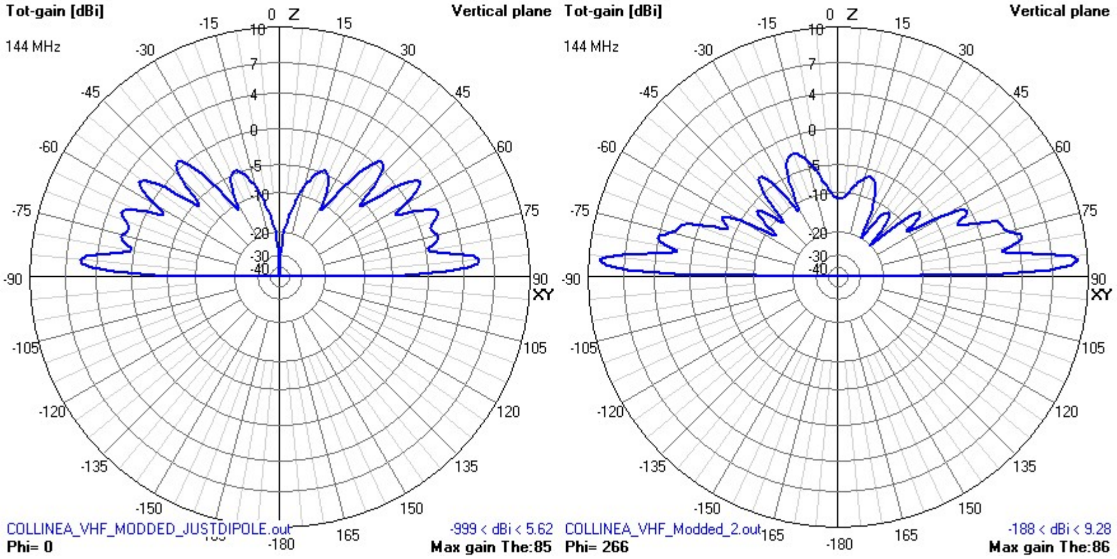
The more elements added to a collinear antenna, the greater the gain will be and the flatter and more directed towards the horizon the radiation pattern will be. With enough elements the radiation pattern becomes disc like. However note that a collinear antenna with too many elements will likely yield poorer performance for some applications like ADS-B compared to one with less. This is because the radiation pattern can become too flat and will miss signals coming in from higher elevations.

Collinear antennas with 50 Ohm impedances can be easily made out of coaxial cable. See here for information on constructing an ADS-B coaxial collinear http://www.balarad.net/. Coax collinears are also sometimes referred to as CoCo antennas. Here is a video showing the construction process of a CoCo antenna https://www.youtube.com/watch?v=TkUYdCPFXXs. One difference between a wire/pipe collinear and a coax collinear is that the coax collinear will be longer. See here for information on constructing a coaxial collinear for AIS http://nmearouter.com/docs/ais/aerial.html. Here is a wire collinear for WiFi http://martybugs.net/wireless/collinear.cgi, its element lengths could be adapted for other frequencies using the calculation they specify.

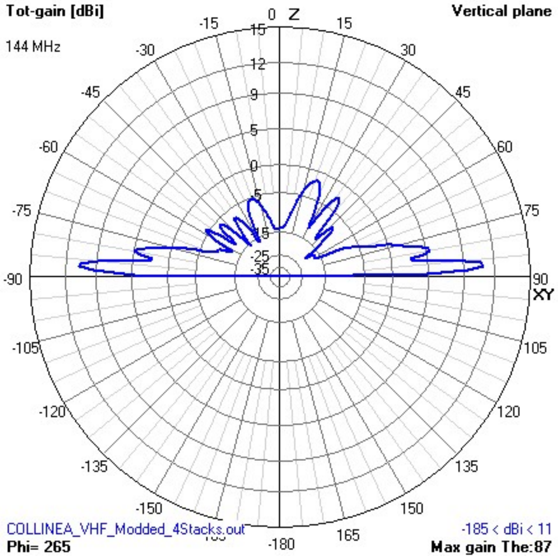
The antenna used in the images shown above is a collinear dipole and is designed for 144 MHz. It is a 300 Ohm antenna, so a matching device should be used with it. The SWR is very low across almost the entire band, but it cannot be classed as a broadband antenna as the radiation pattern becomes poor for frequencies other than the design frequency.



These graphs shown below show the effect of increasing elements on a collinear antenna. As the number of elements increase the radiation pattern flattens out and gain directivity is increased towards the horizon. Though with increasing elements there is a reduced return on investment. The first image shows the radiation pattern of a standard dipole, the second of a collinear with one extra stack and the last image is a collinear with two extra stacks.

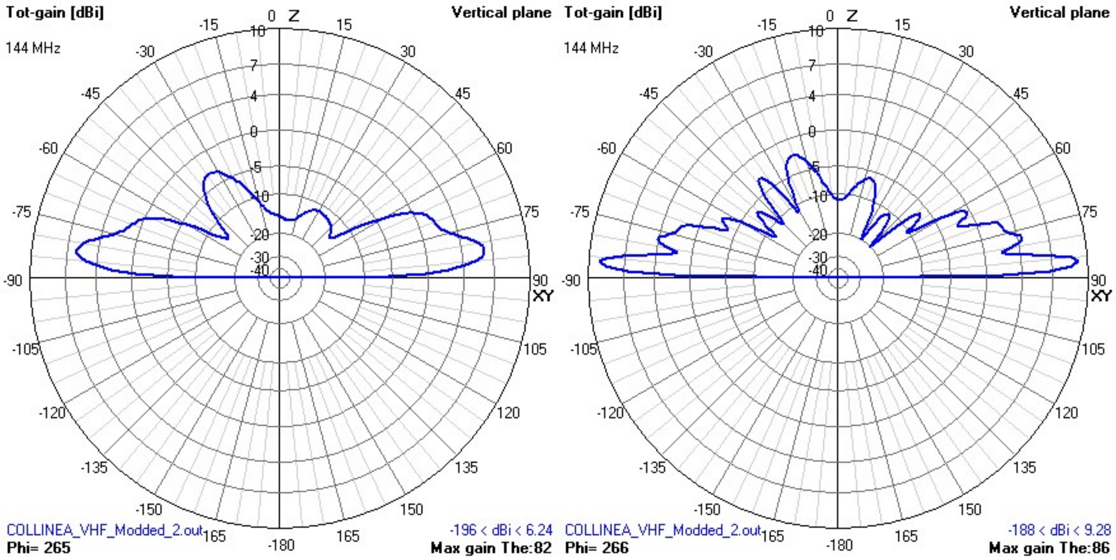


Left: Dipole. **Right:** One stack.

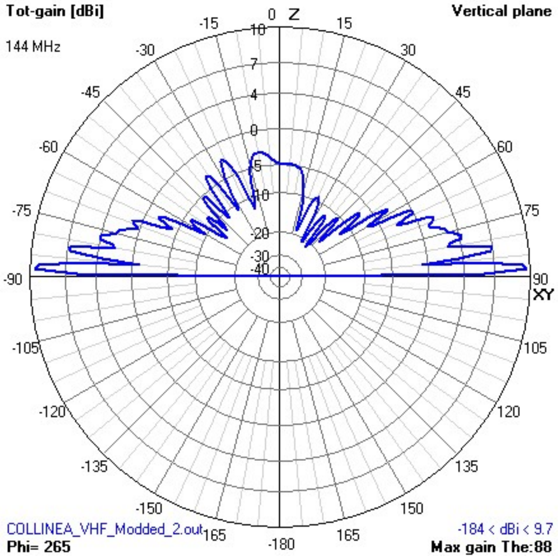


Above: Two stacks.

These images show the difference between 1m, 5m and 10m heights over ground for a collinear antenna with one stack tuned to 144 MHz.

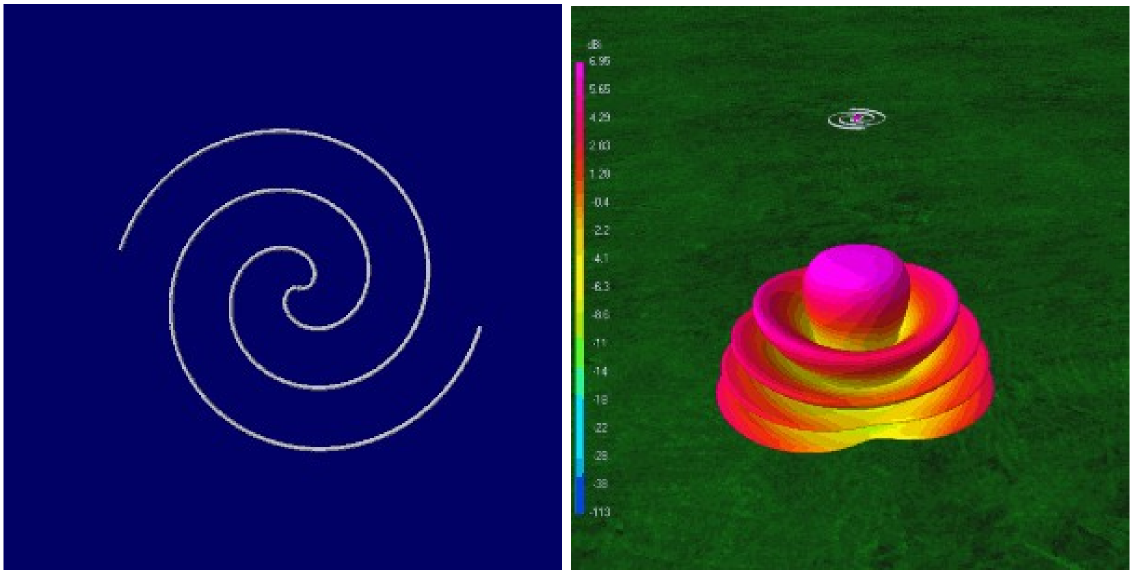


Left: 1M Above Ground. **Right:** 5M Above Ground.



Above: 10M Above Ground.

ARCHIMEDIAN SPIRAL



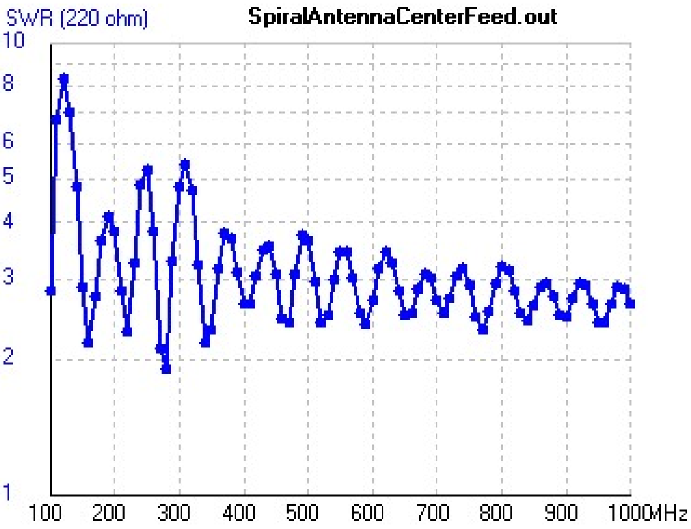
Left: Image of a typical archimedia spiral, **Right:** 3D Radiation Pattern at 137 MHz and 5m high

Name: Archimedian Spiral
Polarization: Circular
Broadband: Yes
Uses: Good for satellite reception.

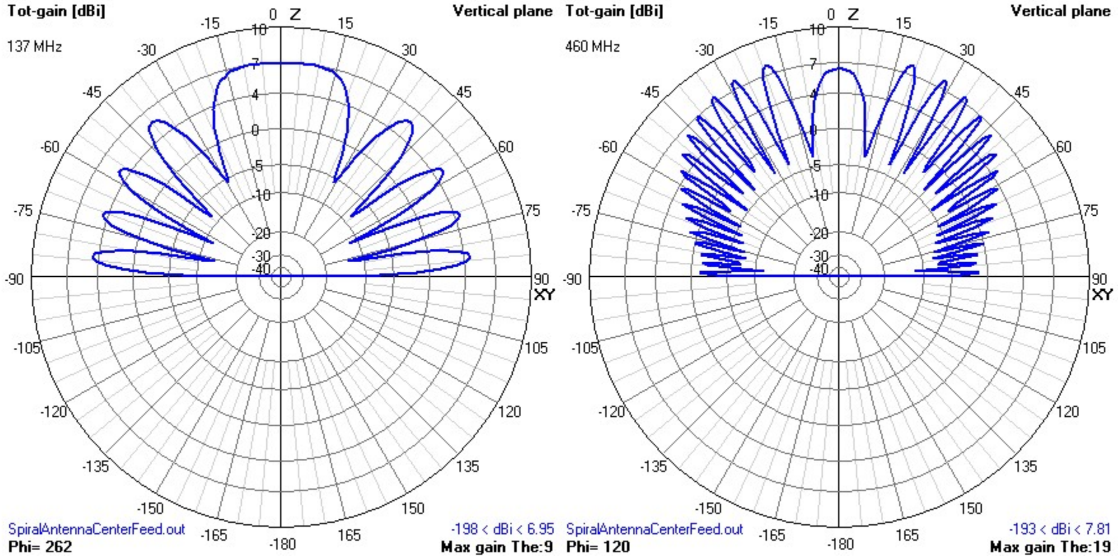
Spiral antennas are extremely wideband antennas that are particularly suited for satellite reception such as for GPS. They are circularly polarized and the direction of polarization can be altered by turning the spiral over.

These antennas can easily be built out of thin coaxial cable. More information about this type of antenna can be found at http://www.antenna-theory.com/antennas/travelling/spiral.php.

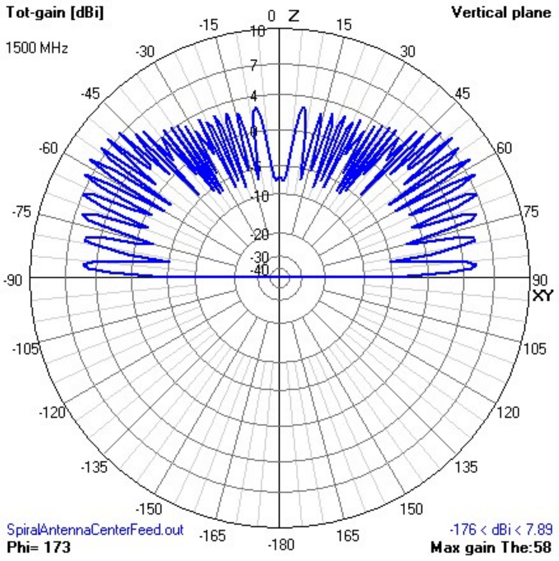
The antenna used above has a radius of 0.5m with 1.5 turns. The impedance of this spiral antenna turns out to be 220 Ohms. The SWR plot is plotted for a matching coax/radio impedance of 220 Ohms, but to connect to a standard 50 Ohm radio a 4:1 matching transformer should be used.



The graphs below show the radiation pattern over various frequencies. As the frequency changes the overall radiation pattern remains quite similar, just with an increasing number of lobes.

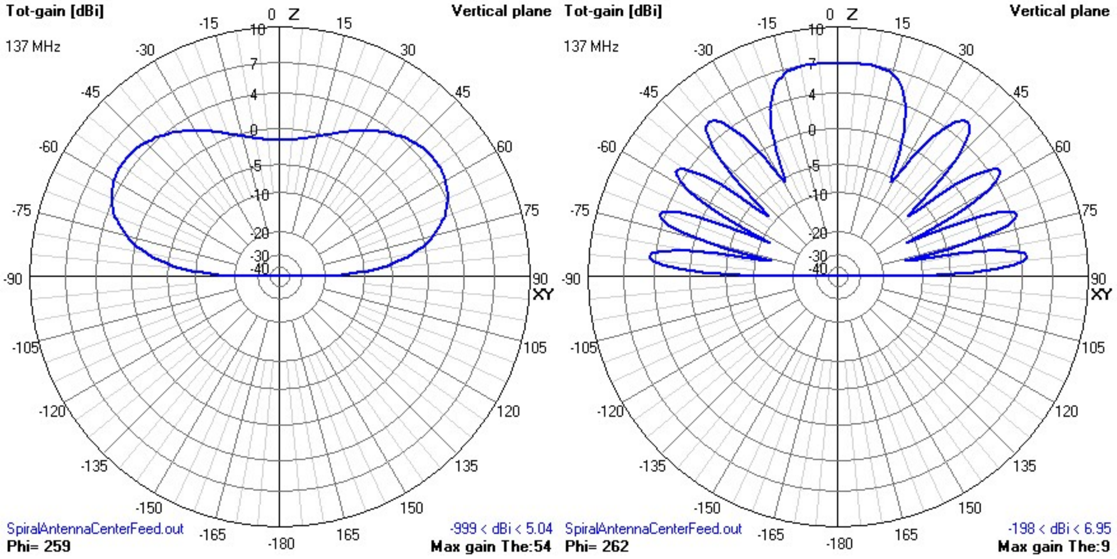


Left: 137 MHz. **Right:** 460 MHz.

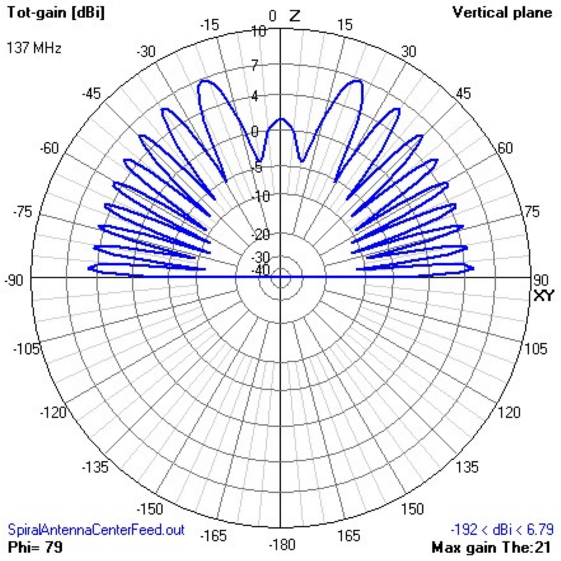


Above: 1500 MHz.

Below we show the Archimedian Spiral at heights of 1m, 5m and 10m at 137 MHz. The radiation pattern and gain changes with differing height.

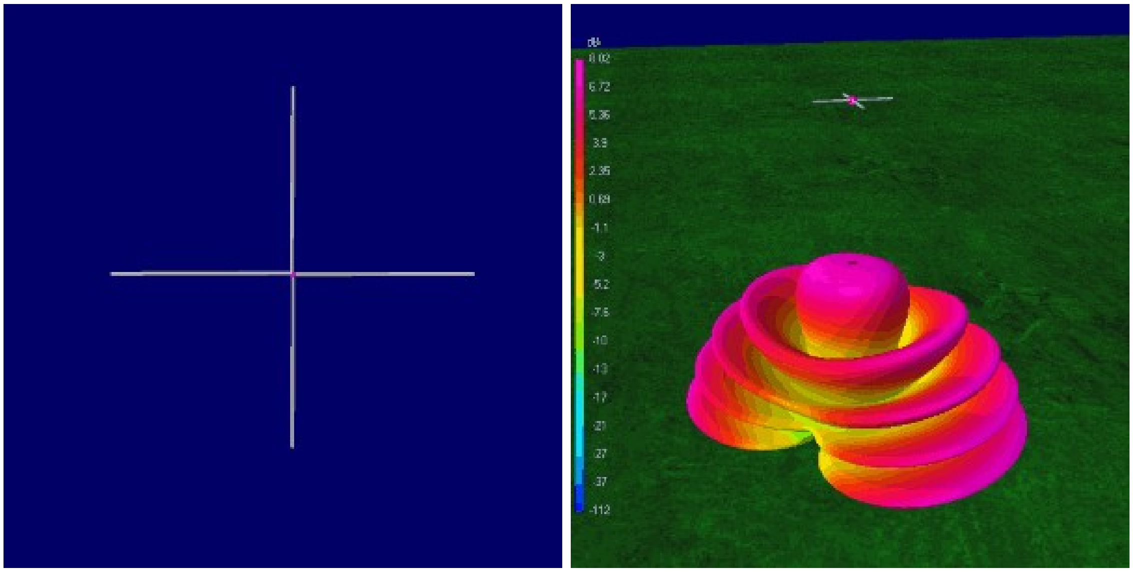


Left: 1M Above Ground. **Right:** 5M Above Ground.



Above: 10M Above Ground.

TURNSTILE / CROSS DIPOLE



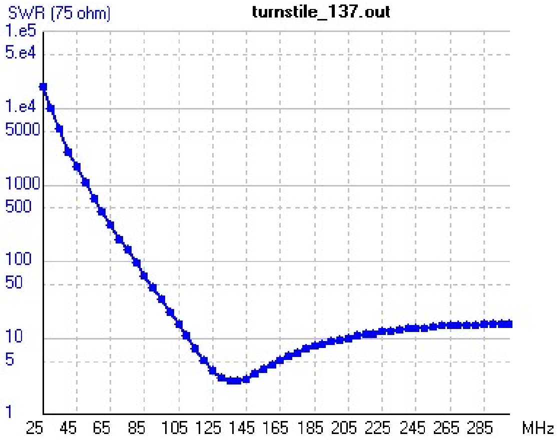
Left: Image of a typical turnstile, **Right:** 3D Radiation Pattern at 137 MHz and 5m high

Name: Turnstile
Polarization: Circular
Broadband: Somewhat. Fairly broadband above design frequency.
Uses: Good for satellite reception.

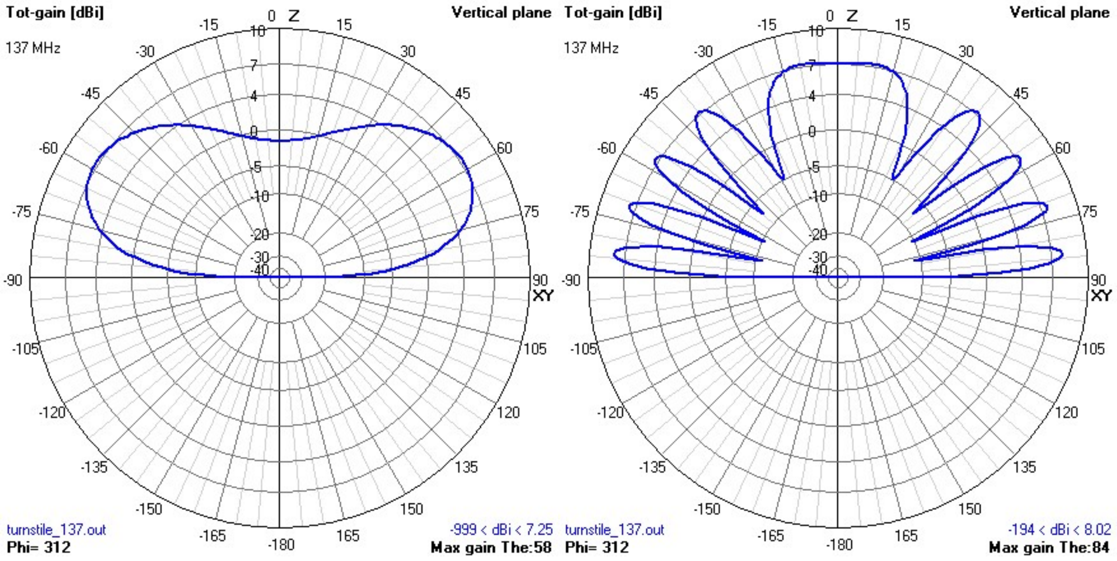
Turnstile antennas are circularly polarized antennas that are often used for FM reception, but can also be used for satellites. Turnstiles can operate in two modes, normal or axial. Normal mode creates an omnidirectional radiation pattern directed towards the horizon. In axial mode the turnstiles gain is directed towards the sky. The radiation pattern shown above and in the graphs below shows the turnstile in axial mode. For satellites the axial mode turnstile is required. If you are building a satellite antenna, make sure you connect the elements in axial mode or you will have poor performance. For satellite reception a reflector should also be used.

This turnstile in the above example images has a design frequency of 137 MHz and is an axial mode design for satellite reception. The VSWR is near 1 at this frequency and quickly ramps up to higher ratios at other frequencies.

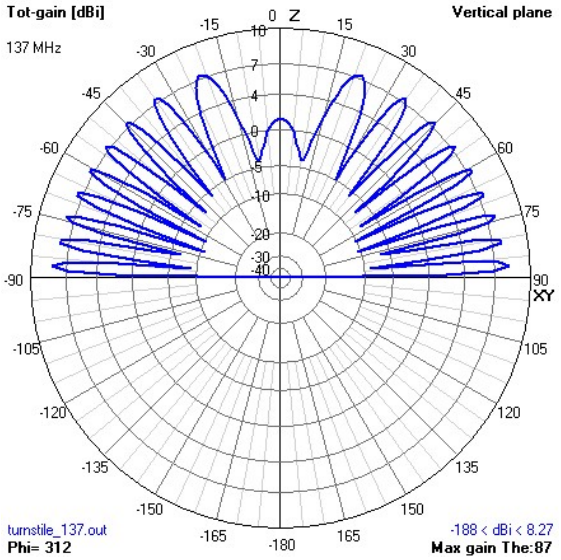
Here is a turnstile build for NOAA satellites http://umarca.blogspot.co.nz/2013/05/project-8-turnstile-antenna-for-weather.html. Here is a video tutorial showing how to build a simple turnstile out of wire for 433 MHz https://www.youtube.com/watch?v=sCypz0ZeDdo#t=201.



The images below show turnstile radiation patterns for 1m, 5m and 10m heights. The radiation patterns are very similar to a spiral antenna with similar dimensions.

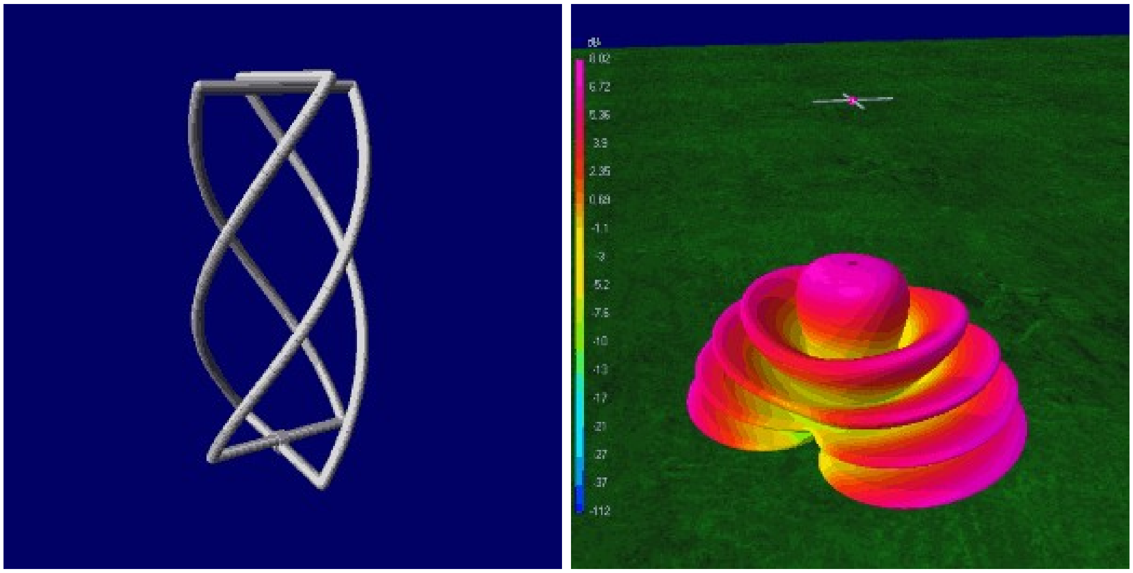


Left: 1M Above Ground. **Right:** 5M Above Ground.



Above: 10M Above Ground.

QUADRIFILAR HELIX (QFH)



Left: Image of a typical quadrifilar helix, **Right:** 3D radiation pattern at 137 MHz and 5m high

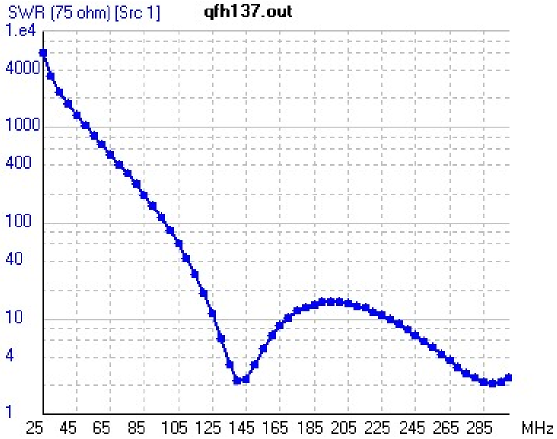
Name: Quadrifilar Helix
Polarization: Circular
Broadband: Somewhat. Fairly broadband above design frequency.
Uses: Good for satellite reception.

Quadrifilar helix (QFH) antennas are circularly polarized antennas that are best used for satellite reception. They tend to work better than turnstile and spiral antennas for this purpose and are especially good with NOAA satellites in the 137 MHz band. From the radiation pattern we can see that QFH antennas have fewer and less deep null areas when compared to a turnstile, making them better suited for good satellite reception.

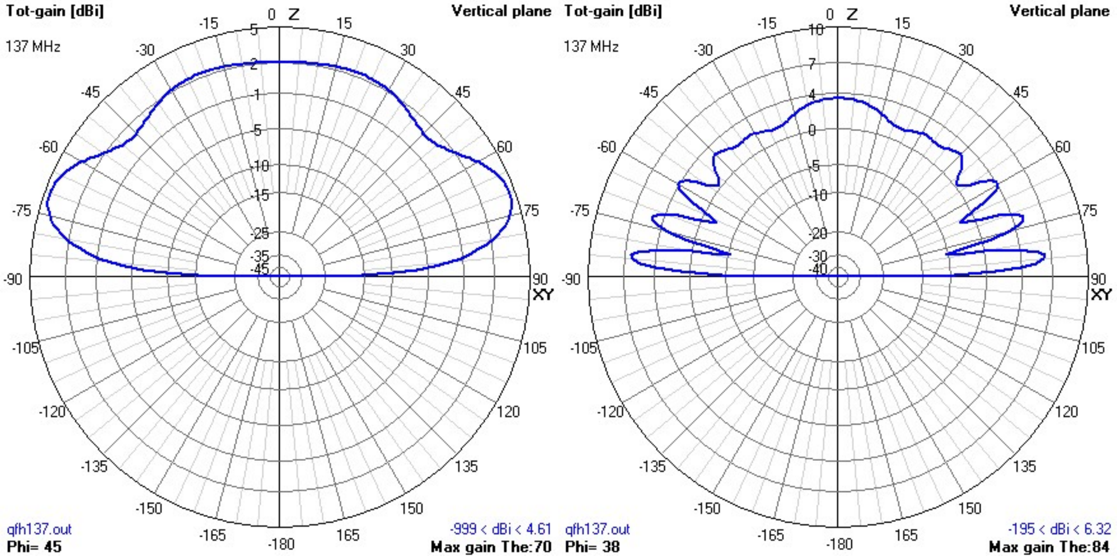
Here is a link which compares the Turnstile vs the QFH antenna http://www.satsignal.eu/wxsat/antennas/. His results show that the QFH works slightly better.

A QFH antenna can be easily constructed out of coaxial cable and some spare wood or PVC piping. A simple tutorial on creating a QFH out of flexible wire can be found here http://sdrformariners.blogspot.com/2013/08/weather-satellites-antennas.html. Here is another design that also uses coax cable which will probably work much better than the previous design http://www.g4ilo.com/qfh.html. If a more rigid structure is required, they can also be built from copper pipe. Here is one example build with copper pipe http://abdallah.hiof.no/QFH/.

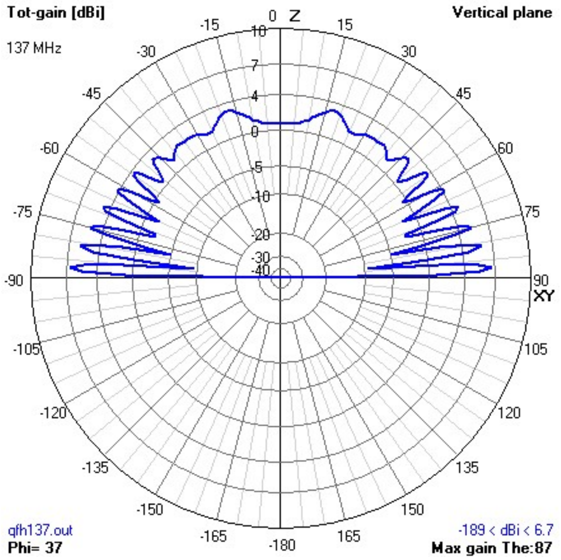
The QFH used in the graphs below has a design frequency of 137 MHz. At this frequency it has good SWR match for a 75 Ohm connection at 137 MHz.



The images below show the radiation patterns for a QFH antenna at 1m, 5m and 10m heights. The gain remains fairly similar at all heights. The main concern with a QFH antenna is getting it to an unobstructed view of the sky so that no nearby objects block its path.



Left: 1M Above Ground. **Right:** 5M Above Ground.



Above: 10M Above Ground.

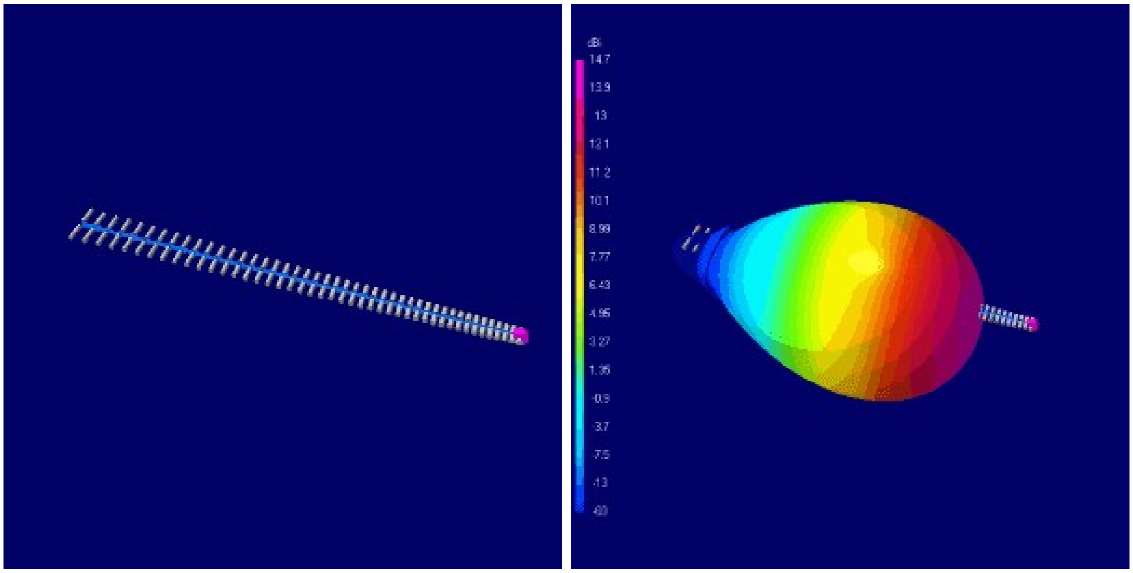
ACTIVE GPS PATCH

Name: Active GPS Patch Antenna
Polarization: Circular
Broadband: No
Uses: Only good for GPS satellite reception.

An active GPS antenna is a patch antenna with a built in LNA. Most active GPS antennas require DC power to be sent down the centre conductor of the coax. Be careful with this, DO NOT let DC power enter the RTL-SDR dongle or it will break.

To connect an active GPS antenna that needs in-line power, you will need to use a circuit called a bias-t. See the GPS tutorial section above for information about bias-t’s.

YAGI UDA



Left: Image of a typical yagi uda, **Right:** 3D radiation pattern at 475 MHz in free space.

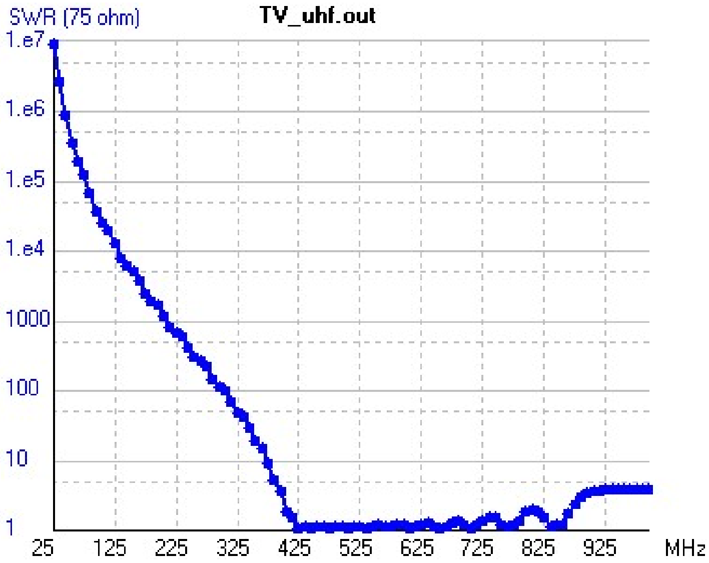
Name: Yagi Uda
Polarization: Horizontal/Vertical/Circular
Broadband: Somewhat. Will work well at frequencies above the design frequency.
Uses: High gain antenna for directional reception.

Yagi Uda, or just yagi antennas are highly directional antennas with very high directivity and gain. Their signal polarization is oriented in the same direction of the orientation of the elements. These antennas need to be pointed in the direction of the transmitting signal and will receive signals poorly from other directions. Yagi's in day to day life are most commonly found as TV antennas and WiFi range extender antennas.

Yagi's make good antennas for weak AIS signals. Here are some links to examples and tutorials for creating Yagi's for AIS. Remember that AIS signals are vertically polarized, so the Yagi must be oriented vertically. Here are some AIS Yagi construction webpages at http://www.vk6fh.com/vk6fh/162mhzyagi.htm and http://www.grafdxradiothings.blogspot.com/2012/03/on-air-home-made-realization-ais.html. They are also great for reaching a distant radio tower or tracking a distant weather balloon.

It is also possible to build circularly polarized Yagis for satellite reception. Since Yagi's are so directional, you will need to either track the satellite by hand as it passes over head or build an automated tracking system. It is usually not too difficult to track a ~137 MHz satellite by hand when combined with an augmented reality smartphone app that shows where the satellite currently is. Satellites at higher frequencies may need more accurate tracking with motorized mechanisms. Circularly polarized yagi antennas are also called Crossed-Yagi antennas.

The example antenna in the above images and graphs below is a high gain TV antenna designed for 475 MHz. From the SWR plot we can see that at its design frequency and above the SWR is near one. The 3D gain plot shown above shows a maximum gain of 14.7 dBi, which is much higher than any of the other antennas mentioned previously.



PCB ADS-B MONOPOLE ANTENNAS

There have been some ADS-B antennas designed on PCB boards with LNA's built into the PCB. These can be classed as 'Active Antennas'.

For example see the following links:

http://f5ann.pagesperso-orange.fr/PCBActiveAntenna/index.html

http://oshpark.com/shared_projects/5cOpKPP6

These antennas can actually be purchased premade and nicely packaged from http://h2204566.stratoserver.net/SmartStore.NET/de/diapason-antenna-series-for-1090-mhz.

HF ANTENNAS

LONG WIRE/RANDOM WIRE

A long wire antenna is simply a wire that is at least one wavelength long of the frequency you want to receive. For most HF frequencies, an antenna this long would be impossible to use practically as it would require too much space. The random wire antenna is an alternative and is quite literally a random length of insulated wire strung up as high and as long as you can physically get it. This is the simplest HF antenna and it can be quite effective for what it is.

The radiation pattern of a random wire antenna is quite unpredictable, though radiation patterns are not that critical for HF antennas, since the signal will usually come from multiple directions. However, this antenna may require a bit of experimentation with location and lengths to get the best performance.

To make this antenna simply connect a coax cable to the antenna input of your upconverter or direct sampling modded RTL-SDR. Run the coax to the location at which your random wire antenna will start. Now connect the random wire antenna to the centre wire of the coaxial cable.

To significantly decrease the amount of noise, connect the shield part of the coaxial cable to a ground. A good ground is a cold water pipe. Avoid using the electrical ground as it could be dangerous and it is also not the best ground due to possible electrical RF interference. If you don't have a good ground accessible, an alternative is to use a counterpoise, which is essentially just a long wire thrown on the ground.

To further improve performance you will need to make a 9:1 "unun" impedance transformer. This is because the impedance of a long wire antenna is 450 Ohms (note the impedance actually varies across different frequencies and 450 ohms is just an overall average). So to match it with the coax cable and RTL-SDRs antenna input impedance of 75 Ohms, the impedance of the long wire antenna need to be reduced by a factor of about 9, which will take it down to 50 Ohms. A 50 and 75 ohm mismatch is insignificant.

This website shows a good tutorial on creating a 9:1 unun http://www.m0ukd.com/Magnetic_Long_Wire_UnUn/ and this PDF file shows detailed construction information for a similar 9:1 unun http://www.earchi.org/92011endfedfiles/Endfed6_40.pdf. Here is another PDF that comprehensively discusses ununs http://setxac.com/wp-content/uploads/2013/12/9-to-1-Balun-for-End-Fed-Antenna.pdf. An even better improvement is to connect an antenna tuner after the 9:1 unun to accurately tune the antenna for the interested frequency, however this will be costly if you do not already own one.

MAGNETIC LOOP

The magnetic loop antenna is probably the best antenna for HF that can be used with the RTL-SDR. This is because the magnetic loop is not only an antenna, but it also acts as a preselector.

The advantage is that the magnetic loop will filter out any interference from signals you aren't tuned to. The disadvantage is that the magnetic loop can be very hard to tune as you need to tune to the frequency you want to listen to on the computer and then tune a variable capacitor on the magnetic loop to tune it to the correct frequency as well. The loop will also only work for a small band of frequencies depending on your variable capacitor and loop size. Also, the lower the frequency you are interested in, the larger and more unwieldy the loop will become.

When constructing a magnetic loop antenna for receiving only, large copper pipes are not required. Most internet tutorials for magnetic loops focus on transmitting loops which need thick pipe and large variable air gap capacitors to support the high transmit voltages. Building a receive only magnetic loop is the same as building one that transmits, but a receive only loop can be built out of thinner material such as coaxial cable and smaller cheaper variable capacitors.

Here is a useful magnetic loop calculator http://www.66pacific.com/calculators/small_tx_loop_calc.aspx. And here is a tutorial for magnetic loop construction http://www.cvarc.org/new-wp/download/technical/magnetic_loop_antenna.pdf. Here is an example of a very simple magnetic loop built out of coaxial cable. This one does not even require a tuning capacitor http://www.carc.org.uk/downloads/A%20Coaxial%20Magnetic%20Loop%20%20for%207MHz.pdf.

4NEC2 TUTORIAL

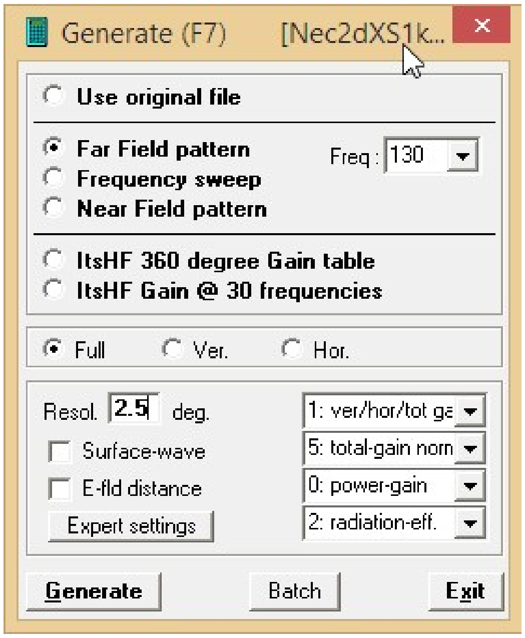
4NEC2 is a free Windows antenna modelling program available from http://www.qsl.net/4nec2/. This software was used to generate the radiation pattern plots and SWR plots for all the antennas shown above. Antenna models are specified as .NEC files. Writing an antenna model is out of the scope of this book, but there are many good example antenna .NEC files included with the 4NEC2 software. Various websites also have antennas models or real commercially made antennas available for download. Just Google for your antennas name + .nec file and you may find that it has already been modelled. The NEC file for the Discone we use in the examples below was found at http://www.qsl.net/kp4md/modelling.htm. If you would like to learn how to write .NEC files, refer to the official 4NEC2 documentation.

Here we show a brief tutorial that shows how to simulate the example .NEC files provided with the 4NEC2 software and any .NEC files you have downloaded. Note that the example files are stored in the 4NEC2 installation folder (likely in Program Files (x86)) under the folder models. The NEC simulation files used in the examples above can all be downloaded from http://bit.ly/1CfSNxT.

HOW TO SIMULATE AN ANTENNA RADIATION PATTERN

1. Open 4NEC2. Go to **File -> Open 4NEC2 in/out file**. Choose the .NEC file for the antenna you are interested in simulating.

2. Go to **Calculate -> NEC output-data**, or press F7. Choose Far Field Pattern. Here you can choose what frequency you would like to generate a radiation pattern for and what resolution you would like to use. The resolution specifies at what spacing of degrees should the gain be calculated. A larger resolution will give a smoother more accurate graph but will take longer to compute. Here we choose to simulate a radiation pattern for 130 MHz using a resolution of 2.5 degrees.



Generate (F7) [Nec2dXS1k...

○ Use original file

● Far Field pattern   Freq : 130
○ Frequency sweep
○ Near Field pattern

○ ItsHF 360 degree Gain table
○ ItsHF Gain @ 30 frequencies

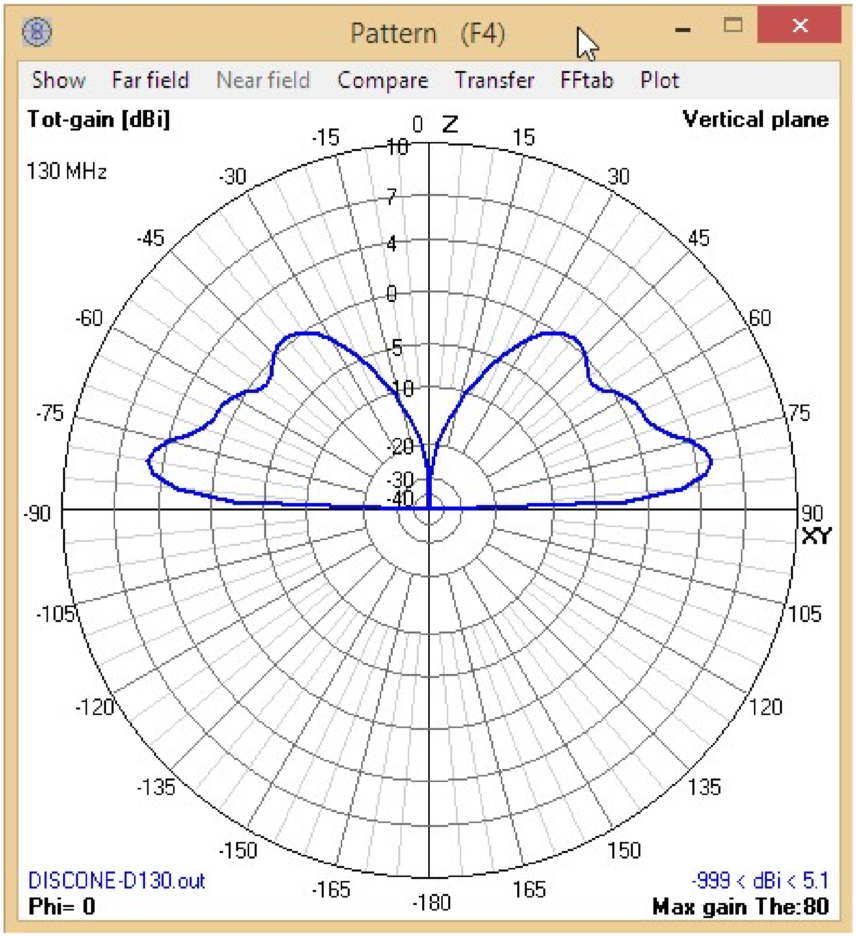
● Full ○ Ver. ○ Hor.

Resol. 2.5 deg.   1: ver/hor/tot ga
□ Surface-wave   5: total-gain norn
□ E-fld distance   0: power-gain
Expert settings   2: radiation-eff.

3. Click Generate to begin the calculations.

4. After the calculations are completed, you will see the gain pattern. If it does not show up go to **Window -> Pattern** or press F4. The gain pattern shows what signal gain you can expect for signals originating at various degrees of elevation. For example here we have simulated a Discone. The graph shows that from straight above (0 degree elevation) the gain is very poor, so we conclude that a discone is not a great receiver for say a satellite that is directly overhead. But at an elevation of 80 degrees (10 degrees above the horizon) the antenna has a high maximum gain of 5.1 dBi.

   By default you will see the radiation pattern for the vertical plane, but if you want to see the horizontal plane as well you can in the Pattern window go to **Far field -> Show both hor/ver**, or press 'D'.

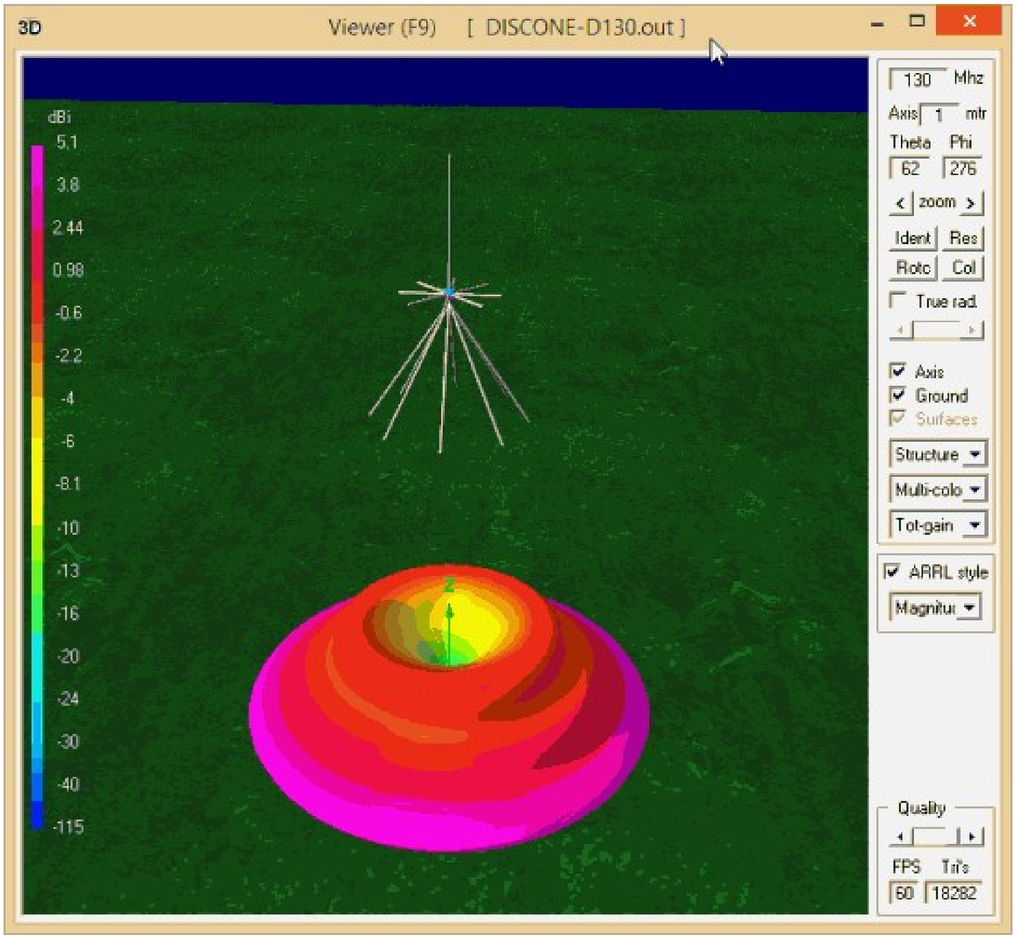


Phi= 0

Max gain The:80

5. To see the pattern in 3D go to **Window -> 3D Viewer**, or press F9. Then on the right of the 3D viewer Window in the second pull down box, change its selection from 'No Pattern' to 'Pattern', or 'Multi-color'. In the 3D Viewer

the the left mouse button rotates, the right mouse button pans and the middle mouse button zooms in and out.


3D Viewer (F9) [DISCONE-D130.out]

130 Mhz

Axis 1 mtr

Theta 62 Phi 276

☐ True rad

☑ Axis

☑ Ground

☑ Surfaces

Structure

Multi-colo

Tot-gain

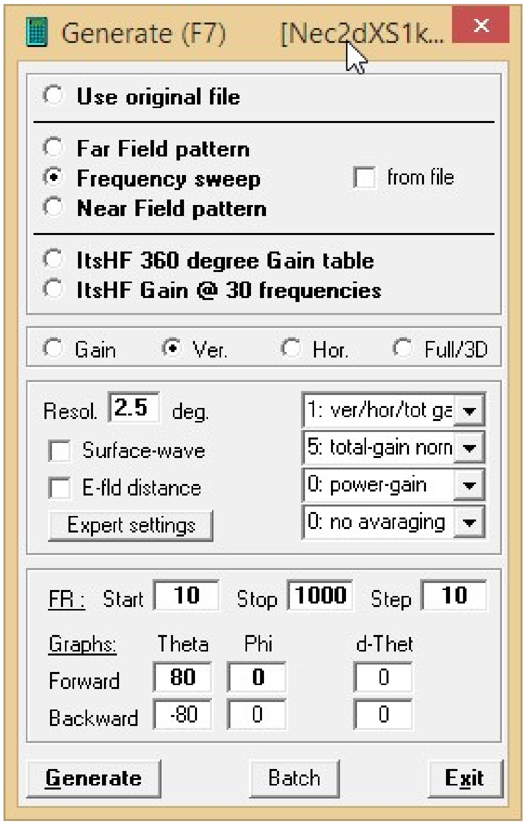
☑ ARRL style

Magnitu

FPS 60 Tri's 18282

HOW TO SIMULATE SWR

1. Open an antenna .NEC file as before if one is not already opened.
2. Set the impedance of your radio or coax connection at **Settings -> Char-impedance**. For the RTL-SDR this should be set at 75 Ohms.
3. Go to **Calculate -> NEC output-data** or press F7. This time choose Frequency Sweep. On the bottom you can choose your desired frequency range. Here we choose a sweep from 10 MHz to 1 GHz with a step size of 10 MHz. A step size of 10 MHz means that the SWR will be calculated at 10, 20, 30 MHz and so on. A smaller step size will give a more accurate graph, but will take longer to compute.



Generate (F7) [Nec2dXS1k...

○ Use original file

○ Far Field pattern
● Frequency sweep □ from file
○ Near Field pattern

○ ItsHF 360 degree Gain table
○ ItsHF Gain @ 30 frequencies

○ Gain ● Ver. ○ Hor. ○ Full/3D

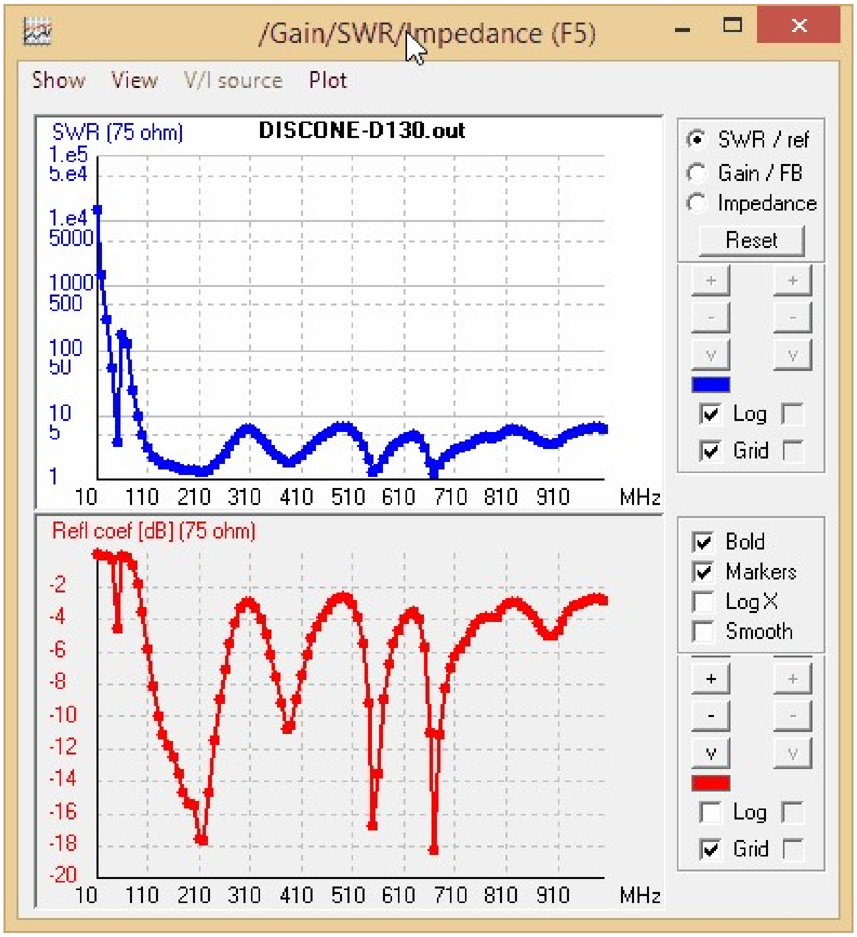
Resol. 2.5 deg. 1: ver/hor/tot ga
□ Surface-wave 5: total-gain norn
□ E-fld distance 0: power-gain
Expert settings 0: no avaraging

FR : Start 10 Stop 1000 Step 10

Graphs:	Theta	Phi	d-Thet
Forward	80	0	0
Backward	-80	0	0

4. Click Generate to begin the calculations.

5. When calculations are complete a SWR graph will show. If it does not show go to **Window -> Line Chart** or press F5. The top graph shows the SWR for various frequencies. The bottom graph shows the reflection coefficient which is similar to SWR as it shows how much wave is reflected. You can also view the total gain over frequency graph by going to **Show -> Gain / FB Ratio** and you can show the impedance over frequency graph by choosing **Show -> Imped./Phase**.



CHANGING THE SIMULATED ANTENNA GROUND

The type of simulated ground used when simulating an antenna can change the antennas modelled radiation pattern significantly. Some antennas NEC files are modelled in free space (i.e without a ground) to get the theoretical radiation pattern. To get a more accurate radiation pattern we should simulate a ground. To see what ground an antenna is using and to change it use these instructions.

1. Go **Settings -> NEC editor (new)** and make sure it is ticked.
2. Click on the Edit NEC Input-file button.
3. An editing window should have popped up. In this window open the the Freq./Ground tab.
4. Now under the Environment box you should be able to choose the type of ground you would like to simulate with. For most realistic situations we use the settings ‘Real Ground’ and set the ground type to ‘Average’. You can

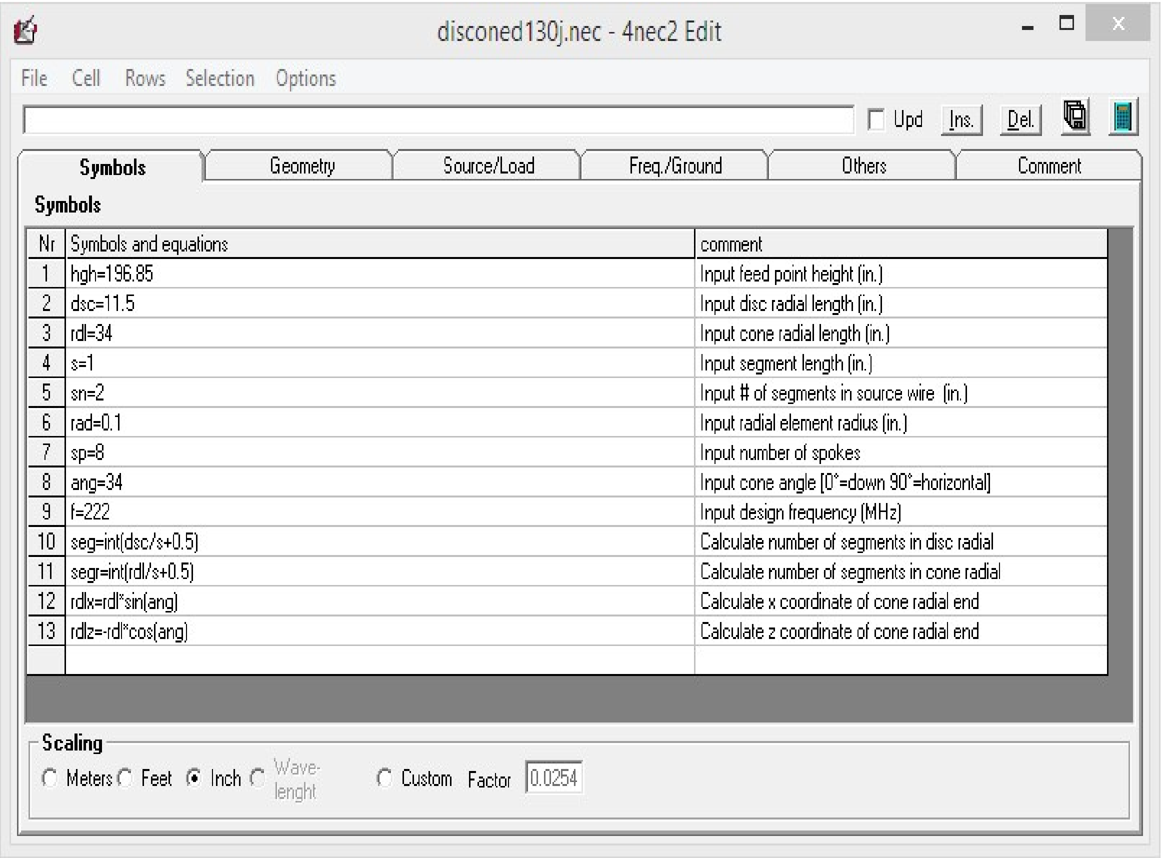
experiment with different types of grounds to see how they affect the radiation pattern.

5. To finish editing the ground, save the modified NEC file by going to **File -> Save**. Then re-simulate the radiation pattern of your antenna to see the changes.

CHANGING ANTENNA DIMENSIONS/HEIGHT

Most antenna .NEC files will come with several variables called ‘symbols’ that can be adjusted to change the dimensions, design frequency and height of the antenna. To see what symbols an antenna you are simulating has follow these instructions.

1. Go **Settings -> NEC editor (new)** and make sure that NEC editor (new) is ticked.
2. Click on the Edit NEC Input-file button .
3. The first tab that has been opened in the new window should be the symbols tab.



Symbols

Nr	Symbols and equations	comment
1	hgh=196.85	Input feed point height (in.)
2	dsc=11.5	Input disc radial length (in.)
3	rdl=34	Input cone radial length (in.)
4	s=1	Input segment length (in.)
5	sn=2	Input # of segments in source wire (in.)
6	rad=0.1	Input radial element radius (in.)
7	sp=8	Input number of spokes
8	ang=34	Input cone angle [0°=down 90°=horizontal]
9	f=222	Input design frequency (MHz)
10	seg=int(dsc/s+0.5)	Calculate number of segments in disc radial
11	segr=int(rdl/s+0.5)	Calculate number of segments in cone radial
12	rdlx=rdl*sin(ang)	Calculate x coordinate of cone radial end
13	rdlz=-rdl*cos(ang)	Calculate z coordinate of cone radial end

Scaling

○ Meters ○ Feet ● Inch ○ Wave-lenght ○ Custom Factor 0.0254

4. Before editing any values make sure to check what the scaling properties are set to in the bottom of the window. For example, if inches are selected than all variables such as height will be input in inches.

5. To save your changes and re-simulate go to **File->Save** and then click on the calculator icon in the top right of the editor window.

APPENDIX A: AUDIO PIPING

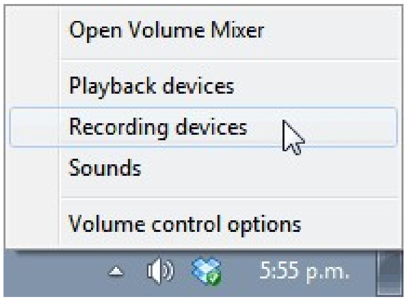
Many applications of the RTL-SDR require that the audio from SDR receiver software such as SDR# be “piped” into decoding software. Piping is simply a method that allows a decoding program to listen to the output audio that is produced by the SDR receiver software like SDR#.

There are several methods to pipe audio in Windows and below we show how to enable the Stereo Mix, VB Cable and Virtual Audio Cable audio piping methods. In most cases we recommend setting up VB Cable or Virtual Audio Cable.

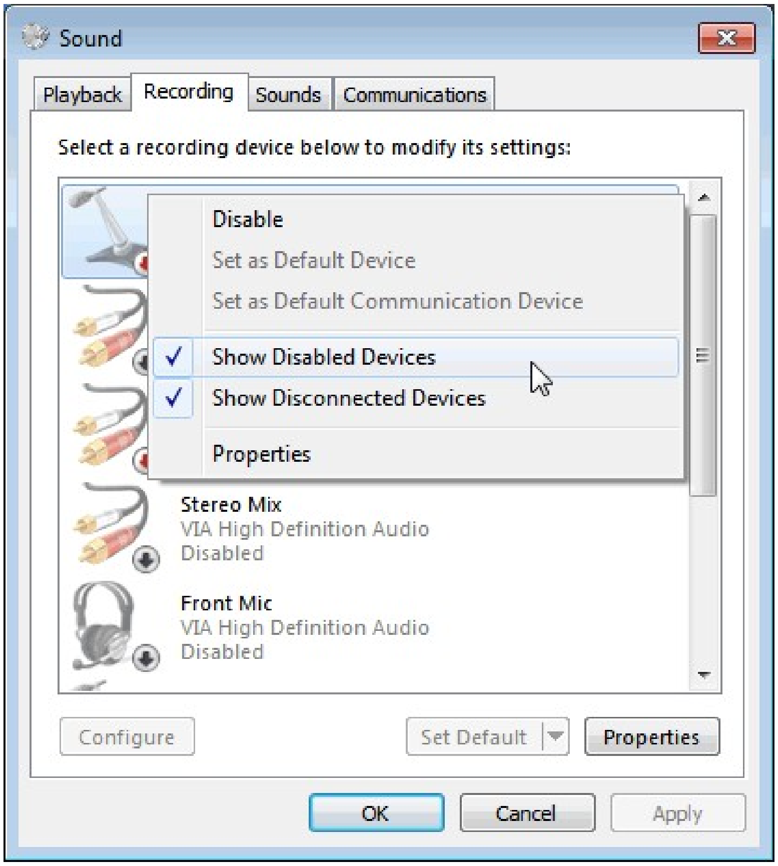
STEREO MIX

The simplest method is to use the Windows Stereo Mix option. With stereo mix, the decoding program will simply take the audio that is being sent to your speakers. You may not want to use this method if your decoding software outputs decoded audio to the speakers, as the raw digital audio will interfere. But it is fine for things like ACARS, weather satellite and fax reception. To enable stereo mix use the following instructions.

1. Right click the speaker icon in your task bar and click on “Recording devices”.



2. Right click anywhere in the window that pops up and ensure that “Show Disabled Devices” and “Show Disconnected Devices” are both checked.

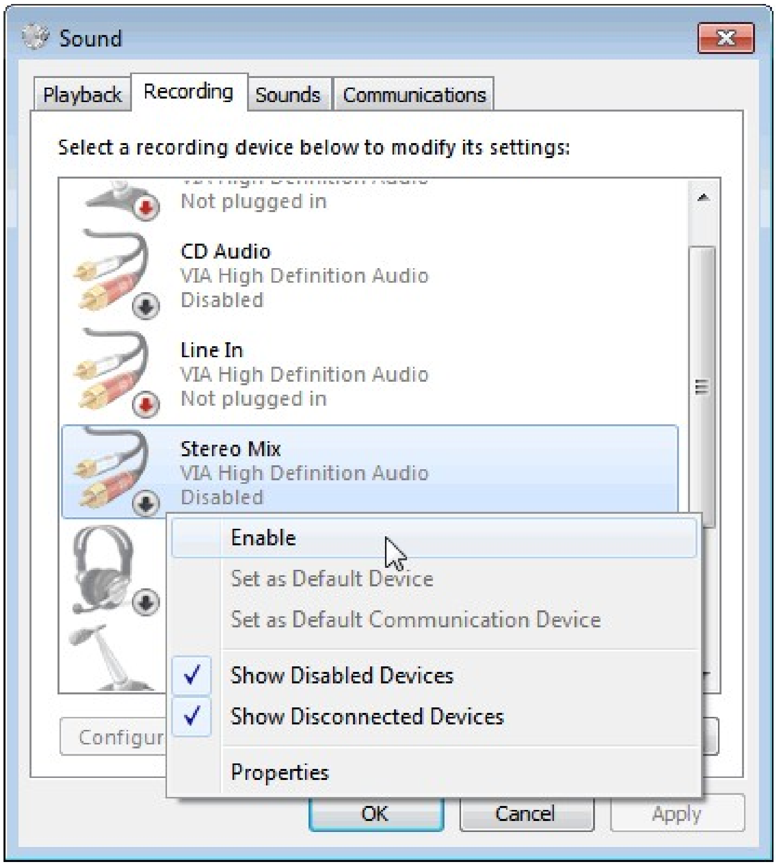


Select a recording device below to modify its settings:

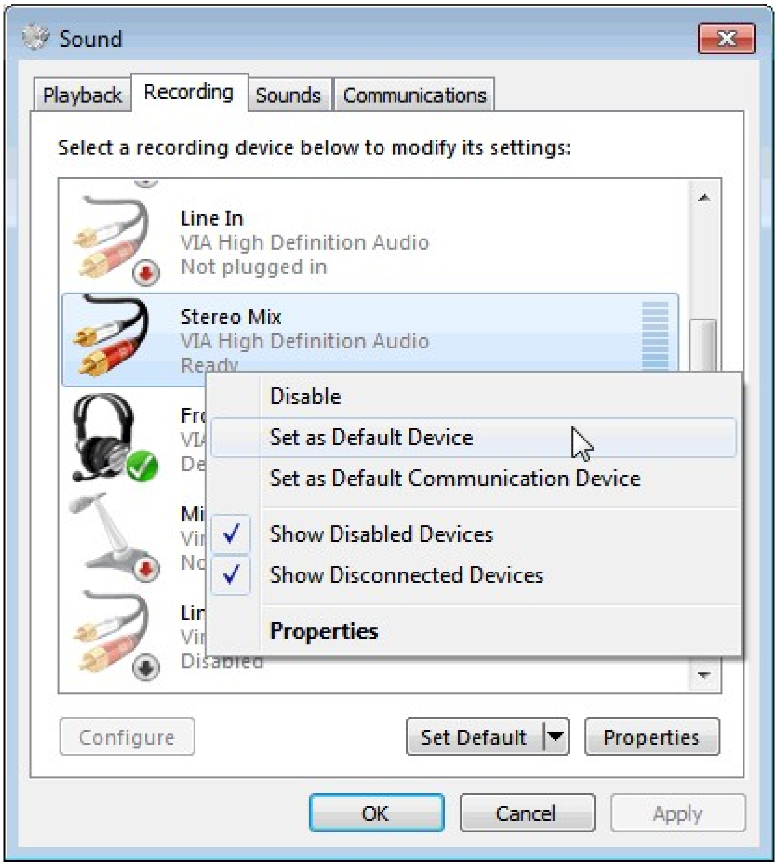
Stereo Mix
VIA High Definition Audio
Disabled

Front Mic
VIA High Definition Audio
Disabled

3. Find the “Stereo Mix” option, right click it and select Enable.


Sound
Playback
Recording
Sounds
Communications
Select a recording device below to modify its settings:
Not plugged in
CD Audio
VIA High Definition Audio
Disabled
Line In
VIA High Definition Audio
Not plugged in
Stereo Mix
VIA High Definition Audio
Disabled
Enable
Set as Default Device
Set as Default Communication Device
✓ Show Disabled Devices
✓ Show Disconnected Devices
Properties
Configur
OK
Cancel
Apply

4. If you have more than one recording device, such as a microphone, or you have also installed virtual audio cable or VB Cable make sure to set stereo mix to the default device. Many programs will select the default device as the audio pipe when looking for an audio source. Note that if you want to use another audio pipe such as virtual audio cable as the default device again, you will need to set that audio pipe back to the default device.



Select a recording device below to modify its settings:

Line In
VIA High Definition Audio
Not plugged in

Stereo Mix
VIA High Definition Audio
Ready

Fr
VI
De

Mi
Vi
No

Lin
Vi
Disabled

Now stereo mix is enabled and ready to use! Any audio coming through your speakers will also be piped to stereo mix which can be selected as an audio piping source in most programs that require one.

If stereo mix is not showing up in the recording properties window, try updating your sound card drivers.

VB CABLE

VB Cable is free software that can be downloaded from http://vb-audio.pagesperso-orange.fr/Cable/. VB Cable installs a new audio device that acts in a similar way to stereo mix. The advantage with a program like VB Cable over stereo mix is that the audio will not play through your speakers. This is desired and required in applications like digital voice decoding, where it is desired to listen to only the decoded voice and not the noise of the digital signal. Additionally, if stereo mix were used, the decoded voice would be piped back into the decoder, causing interference to the digital signal.

The disadvantage to VB Cable is that unlike VAC (shown below), there is no options screen or ability to create cable repeaters.

Note that when installing VB Cable you may need to run the installer as administrator.

To use VB Cable, use the same method as in the stereo mix tutorial shown above, but instead set VB Cable to the default audio device.

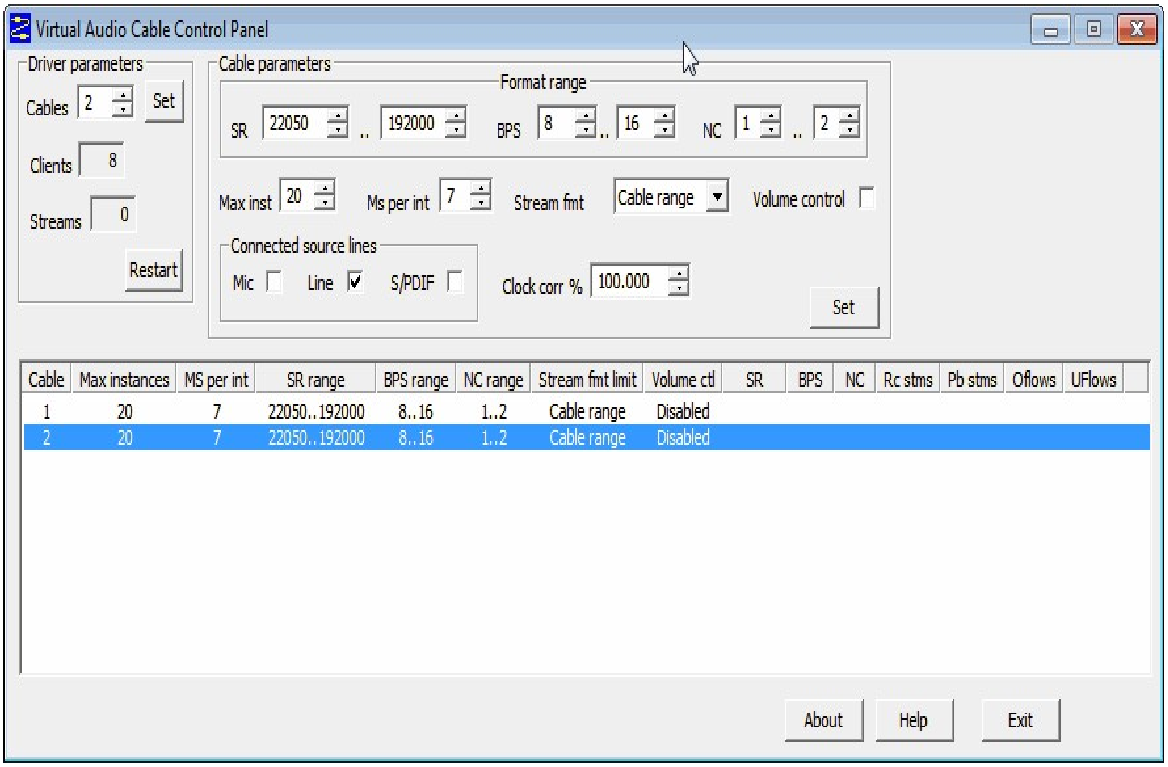
HI-FI CABLE

Hi-Fi Cable is made by the same programmers as VBCable. The difference is that it allows samples rates up to 384 kHz, whereas the standard VBCable only allows sample rates up to 96 kHz. Usually such high sample rates are not required, however for some tutorials, such as with the RDS and SCA tutorials, the higher sample rate is required. Hi-Fi Cable can be downloaded from http://vb-audio.pagesperso-orange.fr/Cable/.

VIRTUAL AUDIO CABLE

Virtual Audio Cable is another tool similar to VB Cable but has a cost of $25-$50. It can be downloaded from http://software.muzychenko.net/eng/vac.htm. They have a trial version which supports up to three cables, but it introduces a watermark voice that periodically says “Trial Version” which might corrupt some signals. However, most digital signals are tolerant to this kind of interference and should work fine with the watermark.

To use Virtual Audio Cable, use the same method as in the stereo mix tutorial shown above, but instead set Virtual Audio Cable to the default audio device.



Virtual Audio Cable Control Panel

Driver parameters

Cables 2 Set

Clients 8

Streams 0

Restart

Cable parameters

Format range

SR 22050 .. 192000 BPS 8 .. 16 NC 1 .. 2

Max inst 20 Ms per int 7 Stream fmt Cable range Volume control □

Connected source lines

Mic □ Line ☑ S/PDIF □

Clock corr % 100.000

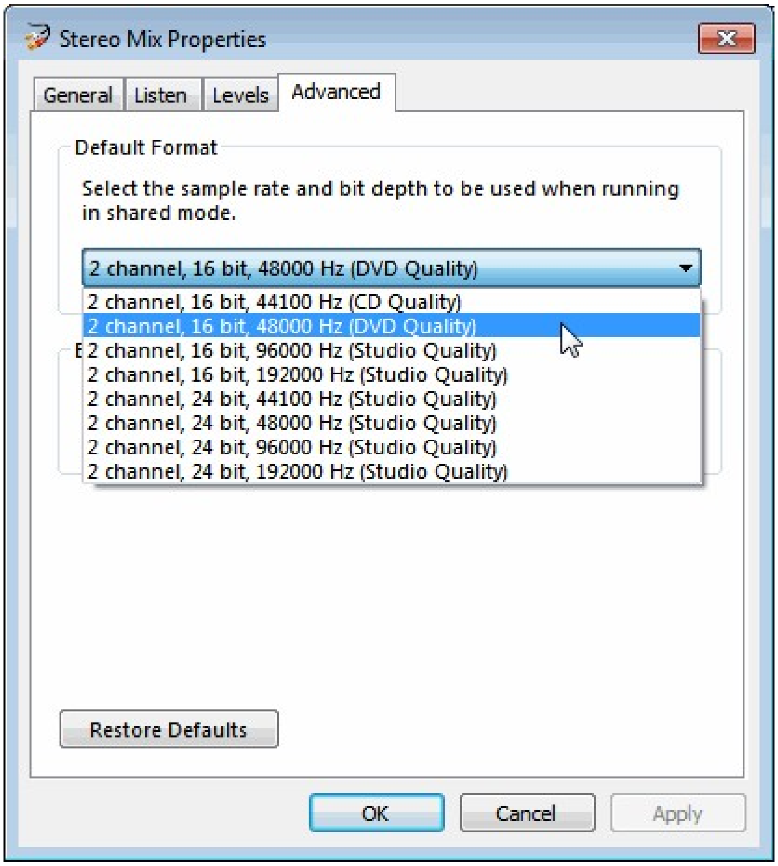
Set

Cable	Max instances	MS per int	SR range	BPS range	NC range	Stream fmt limit	Volume ctl	SR	BPS	NC	Rc stms	Pb stms	Oflows	UFlows
1	20	7	22050..192000	8..16	1..2	Cable range	Disabled							
2	20	7	22050..192000	8..16	1..2	Cable range	Disabled							

SETTING THE AUDIO SAMPLE RATE

For some applications and decoding programs it is essential to correctly set the audio sample rate. To do this use the following steps.

1. Right click the speaker icon in your task bar and click on “Recording devices”.
2. Right click on your audio piping device (eg stereo mix, VB Cable, VAC) and select properties.
3. Go to the Advanced Tab.
4. Choose the sample rate your decoding software requires. Do this for both the Playback and Recording devices if necessary.


Stereo Mix Properties
General
Listen
Levels
Advanced
2 channel, 16 bit, 44100 Hz (CD Quality)
2 channel, 16 bit, 48000 Hz (DVD Quality)
2 channel, 16 bit, 96000 Hz (Studio Quality)
2 channel, 16 bit, 192000 Hz (Studio Quality)
2 channel, 24 bit, 44100 Hz (Studio Quality)
2 channel, 24 bit, 48000 Hz (Studio Quality)
2 channel, 24 bit, 96000 Hz (Studio Quality)
2 channel, 24 bit, 192000 Hz (Studio Quality)
Restore Defaults
OK
Cancel
Apply

Default Format

Select the sample rate and bit depth to be used when running in shared mode.

2 channel, 16 bit, 48000 Hz (DVD Quality)

APPENDIX B: RADIO BASICS

This section will give a brief overview of radio for those just getting into the hobby with no scientific or engineering background in the subject.

In the simplest form, radio waves can be thought of as a type of invisible light. An analogy is as follows. Ships in the old days used to use signal lights to send messages to and from the land and other ships. Radio can be thought of as using an invisible light to do the same thing.

Various frequencies are analogous to different colours. For example, we could have two people using a signal light sending a message with a green colour and one sending a message with a red colour. Because the colors are different we would be able to tell that they are two different messages from two different people. Radio waves work in the same way. One radio station will transmit on one frequency and another will transmit on a different frequency.

Frequencies are measured in Hertz (Hz). Radio waves start in the kilohertz (kHz), and go up to the Megahertz (MHz) and Gigahertz (GHz) frequencies. The frequency ranges are divided up into groups known as low frequency (LF), medium frequency (MF), high frequency (HF), very high frequency (VHF) and ultra high frequency (UHF). Their ranges are shown below.

- LF: 0 kHz - 300 kHz
- MF: 300 kHz—3 MHz
- HF: 3 MHz—30 MHz
- VHF: 30 MHz—300 MHz
- UHF: 300 MHz—3 GHz

These frequency groupings are also sometimes referred to as bands (e.g. we might say that a signal is in the VHF band). The RTL-SDR receives mainly on the VHF and UHF frequencies, but some tuners like the R820T can receive the upper end of the HF range.

PROPAGATION

Radio waves travel in a straight line, just like light does. However, note that light bounces around. A light shone at a single wall will still illuminate the other walls a small room. Radio waves also bounce and tend to bounce around a lot more than light. This allows signals to be received indirectly.

The LF/MF and HF band radio waves tend to bounce around quite a lot and even bounce off the atmosphere itself. This allows these frequencies to travel very long distances allowing them to be received even across the globe. Sometimes ham radio enthusiasts talk about the radio propagation conditions. Depending on the state of the atmosphere and factors such as solar activity, HF frequencies can propagate better or worse.

VHF signals tend to bounce a lot less and they are known as 'line of sight' frequencies. This means the the receiving antenna must have a good unobstructed straight line between the transmitter and antenna. UHF signals are even more 'line of sight'.

BASEBAND SIGNAL

A signal being transmit over radio first starts as a baseband signal. This is a signal that is encoded in a frequency between 0 Hz and some higher cutoff frequency. For example a human ear operates from about 20 Hz to 20 kHz. So we might build a microphone that can record frequencies between 0 Hz and 10 kHz, the recorded audio is our baseband signal. To send this signal over radio frequencies, we need to shift this baseband signal up to an appropriate radio frequency such as 2 MHz, this is done by modulating it into a 2 MHz carrier wave.

BANDWIDTH

When we talk about signal bandwidth we are talking about the size of the chunk of frequency the signal takes up. For example, an AM broadcast signal takes up 12.5 kHz of bandwidth. So an AM signal at a frequency of 500 kHz would span between 493.75 kHz and 506.25 kHz. A broadcast FM station takes up about 192 kHz of bandwidth.

The larger the bandwidth, the more information that can be transmitted. For digital signals this means faster more robust data transmissions and for analogue voice this means higher quality voice.

MODES

There are several methods that can be used to transmit information over radio. The simplest is Amplitude Modulation (AM). In this mode information is encoded into the amplitude of the signal itself. In Frequency Modulation (FM) the information is encoded by varying the frequency of the signal. Broadcast radio uses Wideband FM (WFM), which is simply FM with a large bandwidth of about 200 kHz. Walkie talkies and other similar voice communications use narrowband FM (NFM), which is FM will a small bandwidth of about 12.5 kHz.

Single side band (SSB) modes such as Lower Sideband (LSB) and Upper Sideband (USB) are modes like AM, but only use half the bandwidth of an equivalent AM signal. Transmitting on SSB requires less bandwidth and less transmission power and this is why ham radio users prefer to use SSB modes. The disadvantage is that SSB requires finer tuning to get a signal that sounds right, if you are not tuned correctly the audio will sound either higher or lower pitched. Data modes on HF are often transmitted in the USB or LSB mode.

DECIBELS (DB) TO TIMES

Decibels can be a little confusing for those unaccustomed to them. Decibels (dB) are used to measure the intensity of a signal. The higher the dB value, the stronger the signal. Sometimes we say things like “the LNA (low noise amplifier) amplifies the signal by 20 dB”, or “the coax cable attenuates (reduces) the signal by 3 dB”. A 20 dB increase in signal means that the signal is increased by 100x. A 3 dB attenuation (loss) means that the signal is decreased by 2x.

Engineers like to use decibels instead of times because it allows them to represent massive ranges in smaller numbers that are much more manageable.

The formula to convert decibels to times is as follows, where X is the value in decibels.

$$\text{ratio} = 10 \text{ ^ } (X / 10)$$

To convert to decibels from a ratio or unit use the following.

$$\text{power (dB)} = 10 * \log(\text{ratio})$$

Here is a table converting decibels to ‘times’.

Decibels (dB)	Ratio
30	1000

20	100
10	10
9	8
8	6.31
7	5.01
6	4
5	3.16
4	2.51
3	2
2	1.58
1	1.26
0	1
-3	0.5
-10	0.1
-20	0.01
-30	0.001

-40	0.0001
-50	0.00001

Decibels must be measured with respect to a unit. It is useful to measure it with respect to power in watts (dBW) or milliwatts (dBm). As an example, we might have a signal that has a power of 0.00000000001 mW just before the antenna. This corresponds to -140 dBW or -110 dBm. Our example antenna has a 3 dBi gain in the direction that the signal came from. Then we pass it through a 20dB LNA, loose 5 dB in the coax and then amplify again at the dongle by a further 20dB. Our total input signal strength into the receiver is then given by:

-110 dBm + 3 dBi + 20 dB - 5 dB + 20 dB = -72 dBm

In amateur radio a signal said to be at S9 (good reception) is a signal that is -73 dBm strong, so our -72 dBm signal is quite strong. The RTL-SDR can receive signals down to about -130 dBm.

Note that from this calculation you might think that antenna gain is negligble overall. However, remember than an LNA amplifies any received noise as well as the signal. An antenna with good directivity towards the signal source will pick up more signal and less noise from other directions.

A good video for learning more about decibels can be found at http://www.youtube.com/watch?v=1mulRI-EZ80.

APPENDIX C: MULTIMODE DECODERS LIST

Sorcerer: CW, ACARS, ARQ-E, ASYNC FSK, AX.25 HF, AX.25 VHF, AX.25 9600Bd, CIS-11, Cosmos Navdata, EFR, GTOR, GlobeWireless FSK, IRA-ARQ, ITA2 FSK, IVSU, LINEA SITOR-A, LINEA SITOR-B, M823 Differential GPS, MD-674 ASYNC, Pactor I, Sitor-A, Sitor-B, STANAG 4481, SYNC FSK, CIS MFSK-16, CIS MFSK-20, Coquelet 8, Coquelet 13, Oivia, Piccolo MK6, ARINC 635, GlobeWireless PSK, HAM BPSK/QPSK Modes, MIL-STD 188-110A, MIL-STD 188-110A App. A, MIL-STD 188-110B, Pactor-II, Pactor-III, Stanag 4285, Stanag 4529, ALIS, BARRETT Selcall, Codan CALM ChirpCall, CODAN Selcall, CV-786, Datron S3, GMDSS HF DSC, GMDSS VHF DSC, HARRIS RF-3560, JENAL/SCHUEMPERLIN SC2 BARRETT, JENAL/SCHUEMPERLIN SC2 CODAN, JENAL/SCHUEMPERLIN SC3 BARRETT, JENAL/SCHUEMPERLIN SC3 CODAN, MIL-STD 188-141A ALE, Motorola MDC-1200, NECODE 321ARX Selcall, NECODE 322ARX Selcall, QMAC, RSX.25, SGC BARRETT, SGC CODAN, SPECTRATEK SR-3 BARRETT, THALES HF950, VERTEX, WA2 Selcall, Tone SELCALLS (EIA, EEA, ZVEI1, ZVEI2, ZVEI3, PZVEI, DZVEI, PDZVEI, PDZVEI, NATEL, EURO, MODAT, CCITT, VDEW, CCIR1, CCIR7, PCCIR, CTCSS, DTMF, ICAO), WeFax.

Rivet: CCIR493-4, CIS36-50 (BEE), CROWD36 Burst, CROWD36 Message, FSK200/500 Null Message, FSK200/500 with Message, FSK200/1000, GW FSK (100 Baud) Ship Transmission, GW FSK (100 Baud) Shore Side Transmission, FSK (75 baud 850 Hz shift) KG84, XPA, XPA2.

MultimonNG: POCSAG512, POCSAG1200, POSCAG2400, EAS, UFSK1200, CLIPFSK, AFSK1200, AFSK2400, AFSK2400_2, AFK2400_3, HAPN4800, FSK9600, DTMF, ZVEI1, ZVEI2, ZVEI3, DZVEI, PZVEI, EEA, EIA, CCIR, MORSE_CW.

Fldigi: Contestia, CW, DominoEX, Hellschreiber, MFSK, MT63, NAVTEX, SITOR-B, Olivia, PSK - BPSK, BPSKR, QPSK, Multi-Channel Modems, RTTY, Thor, Throb, TUNE, WEFAX.

MultiPSK: BPSK, QPSK, PSKAM, Packet + APRS, RTTY, ThrobX, Throb, PAX/PAX2, DTMF, FM HELL, PSK H, FEC31, CHIP, PSK220F, Amtor FEC-Navtex, LENTUS, MFSK + PIC, JT65, FELD HELL, PSK10, PSK63F, CW, CCW, Pactor1, MFSK8, OLIVIA, HELL 80, ALE400, FAX, MT63, DIGISSTV, QRSS, ASCII, DoF, THOR, DominoEX, Contestia, RTTYM, 141A (ALE), SSTV, SITORA, SELCAL. ARQ-E(3), POCSAG, FM/RDS, GMDSS, 110A, IEC 870-5, AIS, EPIRB, 1382, DGPS, 4285, HFDL, BIIS, VDL2, ACARS (VHF), SYNOP/SHIP, COQUELET, NWR (SAME), Amtor ARQ.

MixW: BPSK31, QPSK31, FSK31, RTTY, Packet, Pactor, Amtor, MFSK, Throb, MT63, Hellschreiber, Fax, SSTV.

Sigmira: HFDL, PSK31, FSK, SITOR-B, CW, STANAG 4285, Japanese Navy Slot Machine (JSM).

PDW: POCSAG/FLEX, ACARS, MOBITEX, ERMES

SondeMonitor:RS92SGP, Digital RS92AGP, Analog RS80, Analog 92KL, Meteomodem M2K2, Graw DFM-06, Meteolabor SRS-C32.

APPENDIX D: MISC

MANUAL INSTALLATION OF SDR#

Get around the problem: When I run install.bat a CMD/ DOS window flashes on the screen briefly then disappears. Nothing is installed.

1. Download and extract SDR# to a folder: http://sdrsharp.com/downloads/sdr-nightly.zip
2. Download and extract the RTL-SDR SDR# plugin to the same folder: http://sdrsharp.com/downloads/sdr-nightly-rtlsdr.zip
3. Copy and replace the SDRSharp.exe.Config file from config folder into the main SDR# folder.
4. Download the RTL-SDR Drivers, and extract the files from the x32 folder into the SDR# folder, replacing any files if it asks: http://sdr.osmocom.org/trac/attachment/wiki/rtl-sdr/RelWithDebInfo.zip
5. Download zadig from and place it into the SDR# folder: http://zadig.akeo.ie/downloads/zadig.exe (or if on XP download the XP version from http://zadig.akeo.ie)
6. Optional: Download ADSB# and extract it into the SDR# folder: http://sdrsharp.com/downloads/adsbsharp.zip

LAST WORDS

WHERE TO GET MORE HELP

The RTL-SDR community is growing larger every day and there are many people who are willing to help you out online. The largest community of RTL-SDR which is also frequented by experts is over on www.reddit.com/r/rtlsdr. There are also good communities on facebook, IRC and on our RTL-SDR.com blog.

Our RTL-SDR.com Forum: http://www.rtl-sdr.com/forum/

For discussion about SDR# there is an SDR# Yahoo! mailing list forum over on https://uk.groups.yahoo.com/neo/groups/SDRSharp/info.

The most active RTL-SDR Facebook Group: https://www.facebook.com/groups/rtlsdrdongle/

RTL-SDR Google Group: https://plus.google.com/communities/115515752664095602088

SDR# Wiki: http://sdrsharp.pbworks.com/w/page/62589136/FrontPage

SDR Hak5 Forum: https://forums.hak5.org/index.php?/forum/81-sdr-software-defined-radio/

RadioReference.com SDR Forum: http://forums.radioreference.com/software-defined-radio/

There is an active RTL-SDR chat channel on IRC that is frequented by several RF experts: http://webchat.freenode.net/?channels=##rtlsdr

RTL-SDR Forum for ADS-B: http://radarspotting.com/forum/index.php/board,51.0.html

ERRATA AND UPDATES

Note that if you aren’t receiving updates to this book automatically via the Kindle program then you may need to manually contact Kindle support to request an update. We will do our best to ensure that updates reach you, but due to the nature of the Kindle push updates are not always possible. You can check what the latest version is on the Amazon kindle description page.

Edition 4.1 - 15 June 2015

- Updated SDR# image.
- Added a little more info to SDR theory in introduction.
- Clarified info on the range and offset settings in SDR#.
- Clarified info on the FFT display settings in SDR#.

Edition 4 - 5 June 2015

- Added info on adding phantom power to the RTL-SDR
- Updated info on direct sampling to mention the newer RTL-SDR's that coming with pins 4 & 5 on the RTL2832U brought out onto easy to solder to pads.
- Updated info on shielding the RTL-SDR.
- Added info about the FlightAware FlightTracker Android app to the ADS-B tutorial and Android apps section.
- Updated info about modesdeco2.
- Added tutorial on using the RTL-SDR to characterise filters and how to measure the SWR of an antenna with an RTL-SDR.
- Added info on installing GQRX on OSX.
- Added info on the manual install of SDR#, just in case of issues with the install batch file.
- Added a tutorial on installing CubicSDR.
- Added a tutorial on decoding EAS.
- Minor edits to antenna guide.
- Updated RDS tutorial to show use of the MPX plugin in SDR#.
- Updated SCA tutorial to show use of the MPX plugin in SDR#.
- Added info about RTL-SDR sensitivity in the Misc. Information chapter.
- Added more info on optimizing tuning in the Misc. Information chapter.
- Improved SDR# beginners tutorial.
- Minor updates to radio basics appendix.
- Updated Appendix A: Audio Piping to include Hi-Fi cable.
- Improved and reduced book size by optimizing image sizes.
- Updated unitunker tutorial to mention priority and lockout features.
- Added a little info about fleetsync, more to come in later versions.
- Added a tutorial on decoding Meteor M2 in real time.

Edition 3 - 8 January 2015

- Added tutorial on interfacing RTL1090 with XHSI for a live ADS-B based cockpit instrument panel.

- Updated Quickstart guide with extra troubleshooting information.
- Updated information about the AISDecoder software.
- Added information about AISRec.
- Removed tutorial on how to use KG-VDL as the software is no longer available.
- Added tutorial on decoding and listening to TETRA signals.
- Added a tutorial on listening to SCA/SCMO broadcast FM subcarrier channels.
- Added a tutorial on using Acarsdeco2, a superior ACARS decoder for Windows/Linux/MacOS
- Updated info on meteor scatter.
- Added info on testing noise performance.
- Added a recommendation to use Keenerds newer RTL-SDR drivers and software where relevant.

Edition 2 - 1 October 2014

- Changed book cover image.
- Added 4NEC2 tutorial.
- Added a tutorial on receiving LRPT weather satellite images.
- Updated the Trunked Radio Tutorial to include information about new digital output options in Unitrunker.
- Showed how to enable direct sampling software other than SDR# in the Direct Sampling Section.
- Added a tutorial on compiling the latest version of DSD.
- Added USRP B200 information to other SDRs section. Thanks Richard!
- Added SDR Play to other SDRs section.
- Added information about RTL-Bridge and related software to Radio Astronomy Section.
- Updated radio astronomy information.
- Added PNAIS to AIS tutorial.
- Add MilAirComms1090 and FLARM to ADS-B information.
- Added a tutorial on using RTLSDR Scanner as a radio signal locator / heatmap generator.
- Added information about unknown signal identification.
- Updated APRS tutorial.
- Added info about the new HF drivers for the R820T.
- Updated antennas section with better graphs and clearer descriptions.
- Removed section on installing SDR# on Linux as this is no longer possible and will not be possible in the future.

- Updated information about DAB Player.
- Added a new tutorial about Analyzing GSM Signals with Airprobe.
- Fixed several typos.
- Updated GNU Radio installation tutorial.
- Added a TETRA analysis tutorial.

Edition 1.1 - 20 May 2014

- Added Errata and Updates section.
- Fixed typo in the SDR# Setup Guide Radio Tab. 9 kHz should be 9 MHz. Thanks David!
- Fixed CB Radio Frequency Range in General Frequency Guide. Thanks Daivd!
- Fixed ATIS Frequency Range in General Frequency Guide.
- Fixed grammar in title.

Edition 1 - 16 May 2014

- Release

LEGAL STUFF

The contents of this book are intended for educational purposes only. In some countries and regions reception of signals mentioned in this book may not be legal. We take no responsibility for any illegal actions performed using the instructions in this book. Please respect your local laws.

Furthermore, we do not condone the transmission of any radio signal with a transmit capable radio unless you are licensed to do so.

Table of Contents