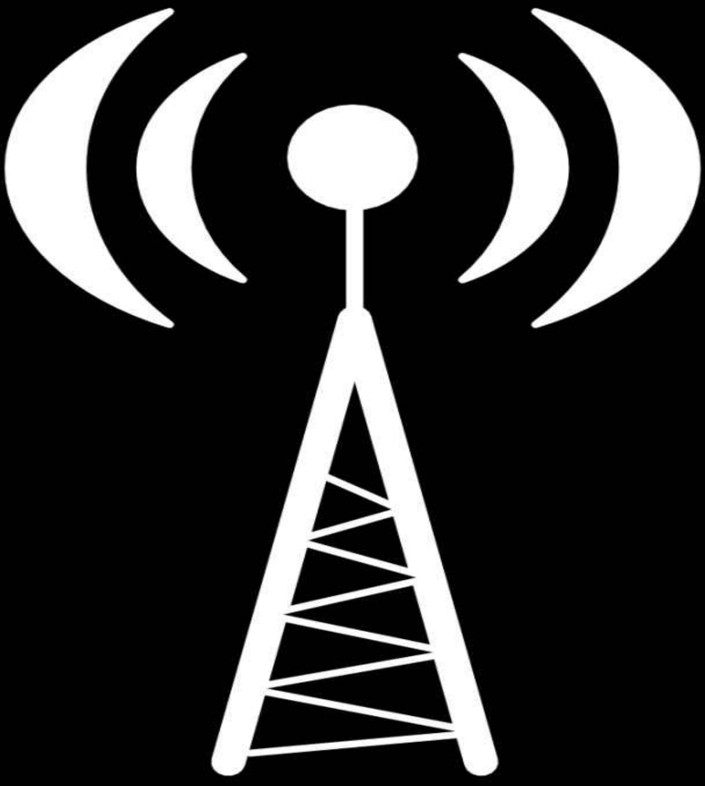
The Hobbyist's Guide to

RTL-SDR



Really Cheap Software Defined Radio

THE HOBBYIST’S GUIDE TO RTL-SDR:

REALLY CHEAP SOFTWARE DEFINED RADIO

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

TABLE OF CONTENTS

- The Hobbyists Guide to RTL-SDR
- Preface
- Introduction
  - What is Software Defined Radio (SDR)?
  - What is The RTL-SDR?
  - What Equipment do I need to get into RTL-SDR?
  - RTL-SDR Technical Specifications
    - RTL-SDR ADC
    - RTL-SDR Bandwidth
    - Input Impedance
    - RTL-SDR Current Usage
  - RTL-SDR Minimum PC Specifications
  - RTL-SDR Compatible Dongles
  - Tips for Buying RTL-SDR Dongles
    - R820T Packages
  - Other SDRs worth mentioning
  - Software Defined Radio Basic Theory
- Setting up and using your RTL-SDR
  - SDR# Setup Guide (Tested on Windows Vista/7 + XP)
    - Troubleshooting Guide
  - SDR Sharp Users Guide
    - Main Settings
    - Other Settings
    - Setting the RF Gain
    - Setting the PPM Correction
    - SDR Sharp Plugins
    - SDRSHarp Minoutput Sample rate
  - Compiling SDR# On Linux with Mono
  - Other General Purpose Scanning Software
    - HDSDR Setup Guide
    - SDR-RADIO.COM V2 Setup Guide
    - GQRX
    - LINRAD
  - Installing the RTL-SDR Drivers on Linux
  - RTL_FM
    - Examples of RTL_FM use
  - Linux with Preinstalled RTL-SDR and Live DVDs
    - KB1OIQ - Andy's Ham Radio Linux
    - Kali Linux
    - Porteus
    - GNU Radio Live DVD
- RTL-SDR Misc. Information
  - RTL-SDR Crystal Tolerance
  - RTL-SDR DC Spike
  - RTL-SDR Spurs
  - RTL-SDR Images
  - RTL-SDR Frequency Drift

- Using Multiple Dongles with the Same Serial Number
- Max Number of Dongles per USB Port

RTL-SDR Improvements and Modifications

- Low Noise Amplification
- Receiving LF/MF/HF (0 - 30 MHz)
  - 1) Direct Sampling Mode
  - 2) Upconverters
- Positioning your RTL-SDR Dongle
  - 1) Use an active USB repeater cable or hub
  - 2) Use a mini embedded or Remote computer to transmit the signal wirelessly to your PC
- Dongle Shielding and Noise Reduction
  - Common Sources of Noise
  - Dongle USB Cable Grounding Improvement
  - USB Cable Ferrites
- Preselector Filters
  - Coax Stub Filter
- Improving the Stock Antenna
- Calibrating the RTL-SDR
- Audio Piping
- General Frequency Guide
- ACARS Receiving Guide
  - Introduction to ACARS
  - ACARS Tutorial (Windows)
- ADS-B Receiving Guide
  - ADS-B Introduction
  - ADS-B Tutorial
  - Feeding Data to flightradar24.com
- NOAA Weather Satellite (APT) Guide
  - Introduction to NOAA Weather Satellites
  - NOAA Weather Satellite Receive Tutorial
- Weather Balloon (Radiosonde) Guide
  - Radiosonde Receiving Tutorial
- Marine Automatic Identification System (AIS) Guide
  - AIS Tutorial
  - AIS Antenna Guide
  - Software Tutorials
- Decoding VHF Data Link Mode 2 (VDL2)
  - Introduction to VDL2
  - Decoding VDL2 with MultiPSK
  - KG-VDL
- Air Traffic Control Scanner Guide
  - SDR# Frequency Scanner
- DAB Radio Guide
  - DAB Receive Tutorial
- Receiving Analogue TV (PAL/NTSC) Guide
- Guide to Listening to Trunked Radio: Analogue and Digital P25
  - Unitrunker Voice Trunking Following Tutorial
  - Decoding Digital Voice (P25/DMR/Mototrbo/NXDN/ProVoice)
  - Tuning DSD+

- CTCSS Trunked Radio

Pager Decoding Guide
- Tutorial
- Other Decoders

Radio Astronomy Guide
- Observing the Hydrogen Line and Galactic Plane
- Meteor Scatter Detection

HF Modes Decoding Guide
- Shortwave Broadcast Radio
- CW (Morse Code)
- RTTY
- STANAG 4285
- SSTV
- Digital SSTV
- WSPR
- Marine HF Modes
- Differential GPS (DGPS)
- Digital Radio Mondiale (DRM) Guide
- HFDL Decoding Guide

D-STAR

RTL-SDR for Android
- SDR Touch
- Wavesink
- USB ADSB RTL-SDR
- SDR Weather

Receiving 10 GHz Beacons

GNU Radio Introduction
- GNU Radio Program: FFT Display
- GNU Radio Program: WBFM Receiver
- GNU Radio Program: Compiling a WBFM-RDS Receiver

Compiling RTL-SDR Drivers from Source on Windows
- Compiling with MingW
- Compiling using Visual C++ 2010

Panadapter Guide

RDS Receiving Guide

APRS Guide
- APRSISCE/32
- AGW Packet Engine and Tracker

MDT Decoding Guide
- More Information and References:

Using multimonNG on Linux and Windows

Using RTL_TCP

Streaming MP3 FM With Linux: RTL_FM

Heat Map Band Scan
- RTL_POWER Flags

RTLSDR Scanner

Train Telemetry Data Decoding

Decoding Funcube Satellite Telemetry

Decoding Orbcomm Satellites

- ISS Satellite Reception
- Weather/Temperature Sensor Decoding Guide
- Decoding Smart Power Meters
- NRF24L01+ Decoding
- Analyzing Unknown Signals
  - Audacity
  - MiniModem
- Receiving and Tracking GPS with the RTL-SDR
  - Active GPS Antennas and Bias-T’s
- Projects For the Future and Better SDR’s
  - HRPT
  - Software DVB-T Decoding
  - DECT Cordless Phones
- Other Interesting Projects People Have Done with the RTL-SDR
- DVB-T HDTV on Linux
  - Drivers
  - DVB-T TV Watching Software
- DVB-T on Windows
- Quick Way to Change between DVB-T and SDR Drivers on Windows Vista/7/8

Antenna Guide
- Antenna ESD Safety
- Antenna Adapter Guide
- Coaxial Cable Guide
- Antenna Gain/Directivity
  - Radiation Patterns
- Antenna Design Frequency
- Standing Wave Ratio (SWR)
- Antenna Polarization
- Antenna Positioning
- VHF/UHF Example Antennas
  - Discone (Broadband Antenna)
  - Scantenna (Broadband Antenna)
  - ¼ Wave Monopole ground plane
  - Whip
  - Dipole
  - J-Pole / Slim Jim
  - Collinear
  - Spiral
  - Turnstile / Cross Dipole
  - Quadrifilar Helix (QFH)
  - Active GPS Patch
  - Yagi Uda
  - PCB ADS-B Antenna
- HF Antennas
  - Long Wire/Random Wire
  - Magnetic Loop

Appendix A: Audio Piping
- Stereo Mix
- VB Cable

- Virtual Audio Cable
- Setting the Sample Rate

Appendix B: Radio Basics

- Propagation
- Bandwidth
- Modes
- Decibels (dB) to Times

Appendix C: Multimode Decoders List

Last Words

- Where to Get More Help
- Errata Updates
- Legal Stuff

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 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.

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.

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 toolbars. The free Calibre software is another good alternative for reading this book.

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 advanced radios that previously required complicated analogue hardware now being able to be implemented easily in software. This has reduced the cost of advanced radio capabilities such as wideband tuning and waterfall displays.

WHAT IS THE RTL-SDR?

The RTL-SDR is an extremely cheap software defined radio which is based on DVB-T TV (Digital HD

TV) USB receiver dongles that have the RTL2832U chip in them. 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 mode which enabled SDR.

Today, by using custom software drivers, a commonly used cheap TV dongle (under $30 USD) can then be turned into a sophisticated SDR 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.

A wideband SDR opens up many interesting possible projects and avenues to explore. Some 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 trunking 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 wireless temperature sensors and wireless power meter sensors.
- Listening to VHF amateur radio.
- Decoding ham radio APRS packets.
- Watching analogue broadcast TV.
- Sniffing GSM signals.
- Using rtl-sdr on your Android device as a portable radio scanner.
- Receiving GPS signals and decoding them.
- Using rtl-sdr as a spectrum analyzer.
- Receiving NOAA weather satellite images.
- Listening to satellites and the ISS.
- Listening to unencrypted military communications.
- Radio astronomy.
- Monitoring meteor scatter.
- Listening to FM radio, and decoding RDS information.

- Listening to DAB broadcast radio.
- Use rtl-sdr as a panadapter for your traditional hardware radio.
- Decoding taxi mobile data terminal signals.
- Use rtl-sdr as a true random number generator.
- 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, and HAARP signals.

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

All one really needs is the following

- 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
- RTL-SDR software (most of which 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 filter noise.

Places to buy 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 giving ~50 dB 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 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 into values between -127 and +127, whereas a 12 bit ADC can scale values from -2047 to +2047.

So with a low bit ADC certain small details in the analogue input, such as weak signals may be lost during digitization. **The RTL-SDR has an 8-bit ADC**, which is fairly low, but large enough to give decent performance. 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 is usually slightly larger (~60 dB) due to the oversampling trick that can be performed in software.

Dynamic range is the range between the largest and smallest possible values.

RTL-SDR BANDWIDTH

The maximum bandwidth of the RTL-SDR is 3.2 MHz, though 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 giving choppy audio.

Most RTL-SDR compatible software will let you choose your bandwidth which is sometimes referred to as sample rate as well. Although 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 frequency spectrum that you can see at any one time.

INPUT IMPEDANCE

As RTL2832U dongles are intended for use with TV, 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 mismatch will be a problem, however the loss due to the mismatch is minimal, equating to less than 0.2 dB.

RTL-SDR CURRENT USAGE

The R820T uses approximately 300 mA of current whilst the E4000 needs approximately 170 mA.

Source: http://superkuh.com/rtlsdr.html#tuners

RTL-SDR MINIMUM PC SPECIFICATIONS

For 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.

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 attention to the tuner chip used in the dongle. The tuner chip defines the frequency range of the dongle.

There are two commonly used tuners. These are the **R820T and E4000** chips. There are also the less common FC0013 and FC0012. Recently there is also the R828D and FC2580, however these are even less common.

Tuner	Min Freq (MHz)	Max Freq (MHz)
R820T	24	1766
E4000	52	2200
FC0012	22	948.6
FC0013	22	1100
R828D	24	1766
FCI FC2580	146/308	438/924

The R820T is the most commonly purchased RTL-SDR. The E4000 used to be more popular but Elonics, the manufacturer has closed down making these chips rare and expensive. Other tuners are less commonly found.

For these reasons, currently RTL-SDR.com recommends the R820T tuner, unless you require the higher frequencies that the E4000 tuner gives.

TIPS FOR BUYING RTL-SDR DONGLES

There are several online places to buy RTL-SDRs. The one place that RTL-SDR.com recommends the most is from the Nooelec Amazon store. Nooelec is a company that knows their products and can provide installation support should you need it. When buying from Amazon you also get the extra Amazon buyers protection. If you are a first time buyer, we highly recommend the Nooelec Amazon store. See our Buy RTL-SDR Dongles page online for information on obtaining dongles http://www.rtl-sdr.com/buy-rtl-sdr-dvb-t-dongles/.

Dongles are also available on Ebay. However, you must be careful when buying on Ebay as sometimes sellers misrepresent their product. This is especially the case with the E4000 tuner dongles. Some sellers advertise E4000 dongles cheaply, when in fact they are actually R820T dongles.

Aliexpress.com is also another place to purchase these dongles. However, the same warnings as with EBay apply here, plus shipping times are much slower.

R820T PACKAGES

The commonly bought R820T 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 now which is very tiny.

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. Note that it has also been reported that the ‘nano’ models get hotter causing greater frequency instability.

OTHER SDRS WORTH MENTIONING

	Freq	Bandwidth	ADC (Bits)	Comments	Price

	Range (MHz)	(MHz)			(USD)
Funcube	0.150 - 260 410 - 2050	0.192	16	RX. Filters	200
HackRF	30 - 6000	20	8	RX/TX	300
BladeRF	300 - 3800	40	12	RX/TX	400, 650
USRP B210	70 - 6000	56	16	RX/TX	1100
SockRock Ensemble II	0.180 - 3	0.192	Soundcard	RX/TX	67 (TX) 89 (RX/TX)
AirSpy	24 - 1750	10	12	RX	~100
Mirics (Msi3101)	64 - 108 162 - 240 470 - 960	5	12	RX	85

In the cheaper end of the SDR spectrum is the AirSpy dongle which promises to have improved performance over the RTL-SDR. At the time of writing this book AirSpy has not yet been released. The Funcube dongle is another device similar to the RTL-SDR, but with improved RF design and some front end filters. The Softrock Ensemble II is a dedicated HF receiver which comes as a kitset. You will need good soldering and construction skills to complete the kit. It has excellent shortwave reception performance.

In the higher priced end of the SDR spectrum we have the HackRF/BladeRF and USRPs which are all capable of receive and transmit.

There is also the 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

An SDR simply works by receiving an analogue radio signal 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.

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 an IF 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).

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

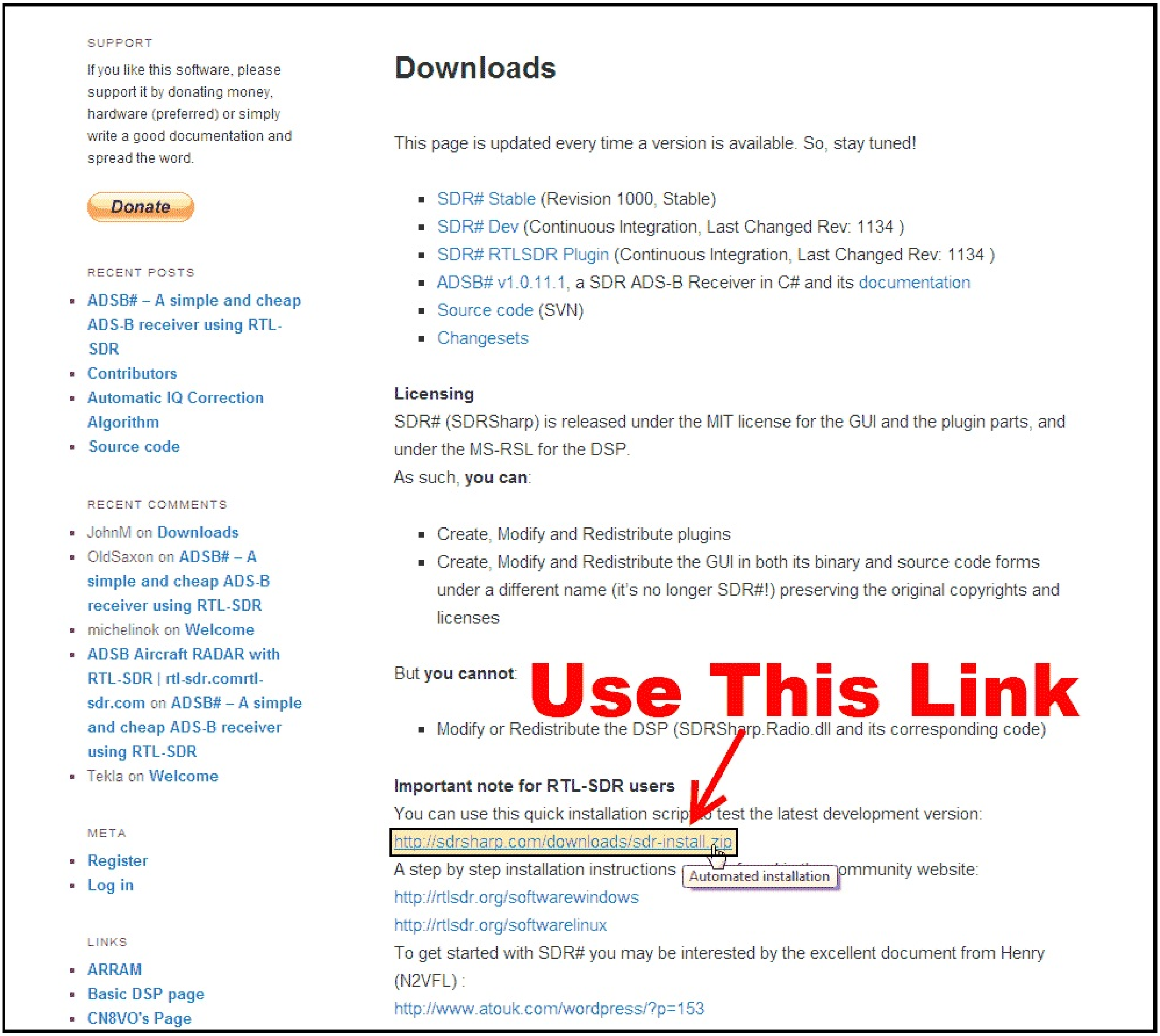
SETTING UP AND USING YOUR RTL-SDR

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

1. Purchase an RTL-SDR dongle. The cheapest and best for most applications is the R820T dongle, which can be purchased from Nooelec http://www.rtl-sdr.com/buy-rtl-sdr-dvb-t-dongles/.

2. SDRSharp is the easiest and most commonly used software defined radio software receiver for the RTL-SDR. Go to www.sdrsharp.com and head to the download page. Note that you must have the Microsoft .NET redistributable installed to use SDRSharp. Modern Windows PCs 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. Ignore the downloads on the top of the page. Scroll down to where it says “Important note for RTL-SDR users”. Download sdr_install.zip from that link. **NOTE: At the time of writing this book the SDR# downloads page has recently changed. The sdr_install.zip file download is currently available from the main page at sdrsharp.com but it’s location may change in the future.**



SUPPORT

If you like this software, please support it by donating money, hardware (preferred) or simply write a good documentation and spread the word.

Donate

RECENT POSTS

- ADSB# – A simple and cheap ADS-B receiver using RTL-SDR
- Contributors
- Automatic IQ Correction Algorithm
- Source code

RECENT COMMENTS

- JohnM on Downloads
- OldSaxon on ADSB# – A simple and cheap ADS-B receiver using RTL-SDR
- michelinok on Welcome
- ADSB Aircraft RADAR with RTL-SDR | rtl-sdr.comrtl-sdr.com on ADSB# – A simple and cheap ADS-B receiver using RTL-SDR
- Tekla on Welcome

META

- Register
- Log in

LINKS

- ARRAM
- Basic DSP page
- CN8VO's Page

Downloads

This page is updated every time a version is available. So, stay tuned!

- SDR# Stable (Revision 1000, Stable)
- SDR# Dev (Continuous Integration, Last Changed Rev: 1134)
- SDR# RTLSDR Plugin (Continuous Integration, Last Changed Rev: 1134)
- ADSB# v1.0.11.1, a SDR ADS-B Receiver in C# and its documentation
- Source code (SVN)
- Changesets

Licensing

SDR# (SDRSharp) is released under the MIT license for the GUI and the plugin parts, and under the MS-RSL for the DSP.

As such, **you can**:

- Create, Modify and Redistribute plugins
- Create, Modify and Redistribute the GUI in both its binary and source code forms under a different name (it's no longer SDR#!) preserving the original copyrights and licenses

But **you cannot**:

- Modify or Redistribute the DSP (SDRSharp.Radio.dll and its corresponding code)

Important note for RTL-SDR users

You can use this quick installation scrip o test the latest development version:

http://sdrsharp.com/downloads/sdr-install.zip

A step by step installation instructions ommunity website:

http://rtlsdr.org/softwarewindows

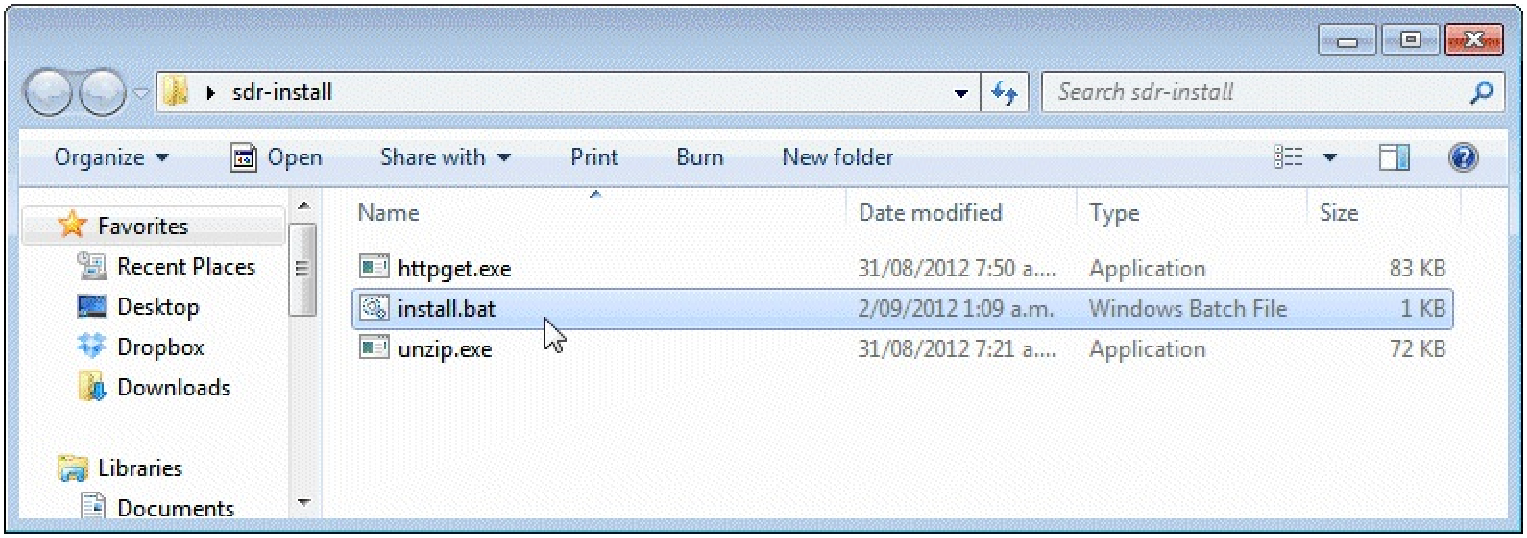
http://rtlsdr.org/softwarelinux

To get started with SDR# you may be interested by the excellent document from Henry (N2VFL) :

http://www.atouk.com/wordpress/?p=153

4. Extract the sdr-install folder from the zip file to a place on your computer.

5. Double click on install.bat from within the extracted folder. This will start a command prompt that will download SDRSharp and all the files required to make SDRSharp work with 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.

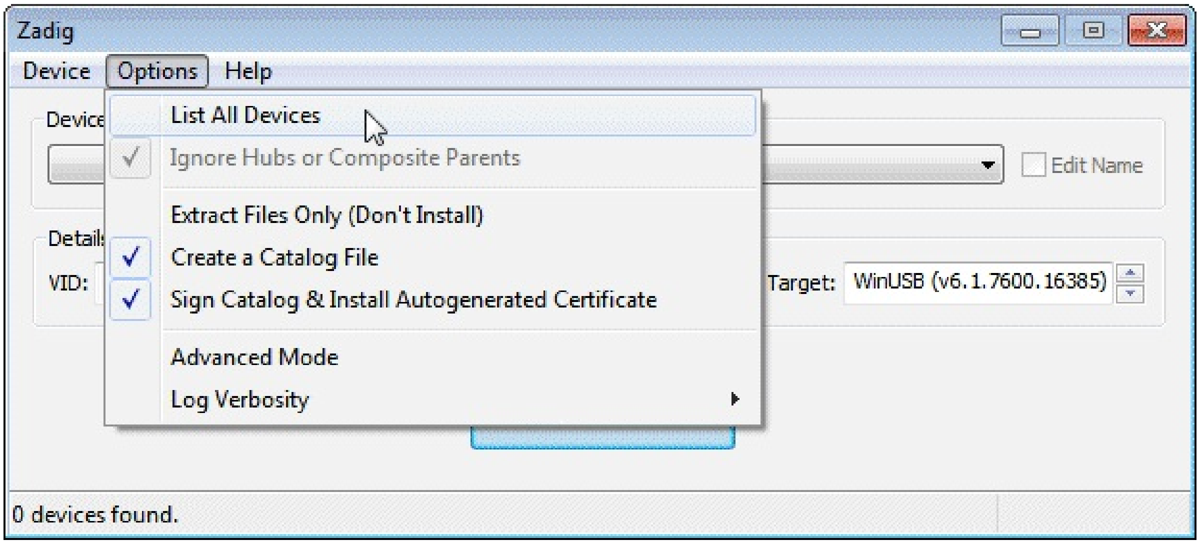


Name	Date modified	Type	Size
httpget.exe	31/08/2012 7:50 a....	Application	83 KB
install.bat	2/09/2012 1:09 a.m.	Windows Batch File	1 KB
unzip.exe	31/08/2012 7:21 a....	Application	72 KB

6. Plug in your dongle and do not install any of the software that it came with, but ensure you let plug and play finish trying to install it. If necessary, uninstall any DVB-T software drivers you’ve installed from the CD that some dongles come with.

7. Open the newly created sdrsharp folder. 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 from http://zadig.akeo.ie/. If you are on Windows 8 you may get unsigned driver issues when using zadig. To get around this use the newer version (v2.1+) of zadig from http://zadig.akeo.ie/.

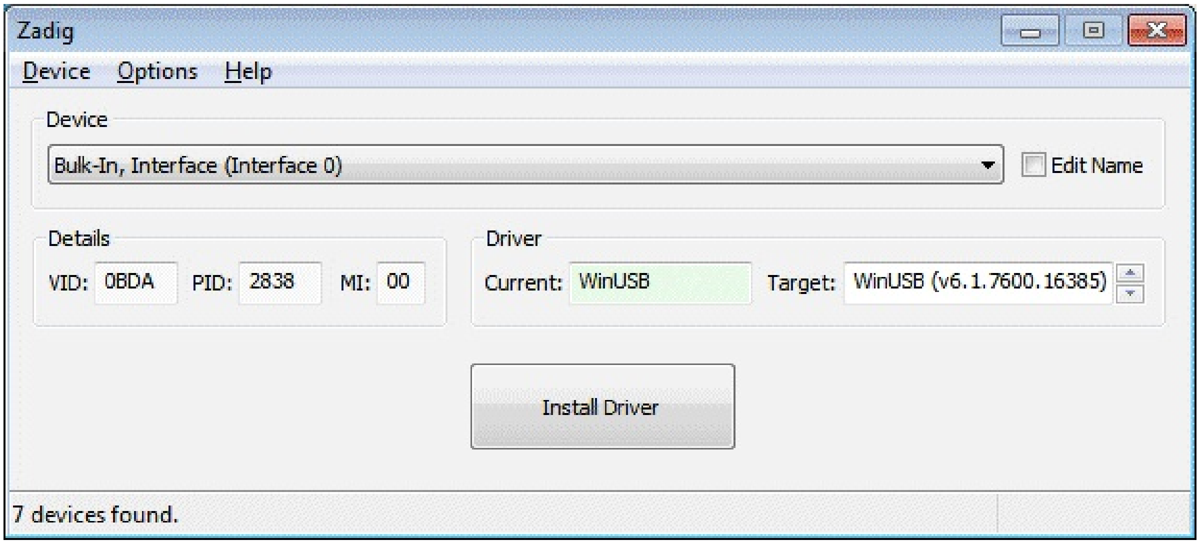
8. In Zadig, go to **Options -> List All Devices** and make sure this option is checked.


Zadig
Device
Options
Help
List All Devices
✓ Ignore Hubs or Composite Parents
Extract Files Only (Don't Install)
✓ Create a Catalog File
✓ Sign Catalog & Install Autogenerated Certificate
Advanced Mode
Log Verbosity
□ Edit Name
Device
Details
VID:

Target: WinUSB (v6.1.7600.16385)

0 devices found.

9. 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 instead of the bulk in interface. This is also a valid selection).



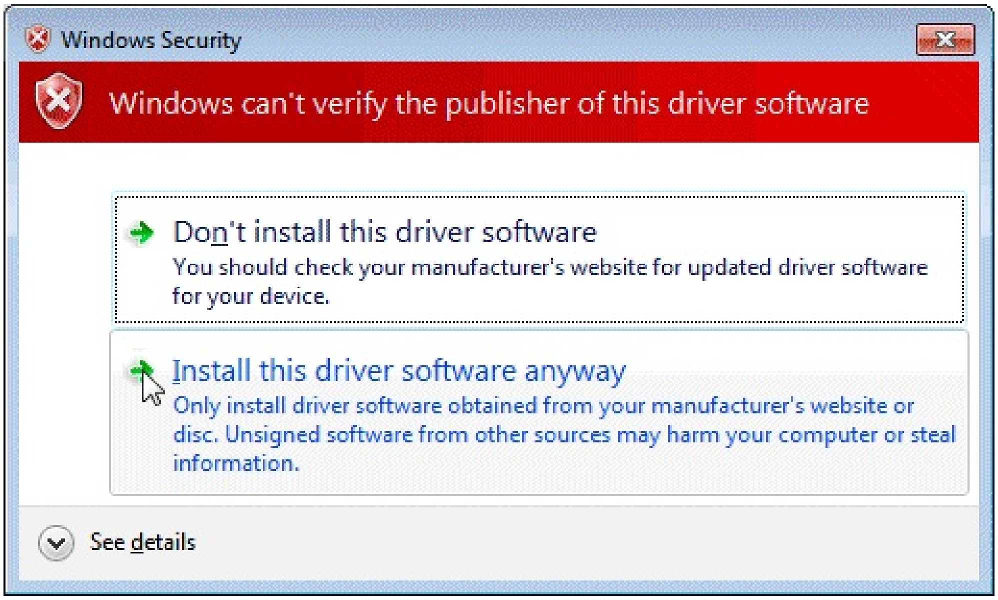
Device: Bulk-In, Interface (Interface 0)

Details — VID: 0BDA PID: 2838 MI: 00

Driver — Current: WinUSB Target: WinUSB (v6.1.7600.16385)

7 devices found.

10. Click Install Driver. You might get a warning 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.

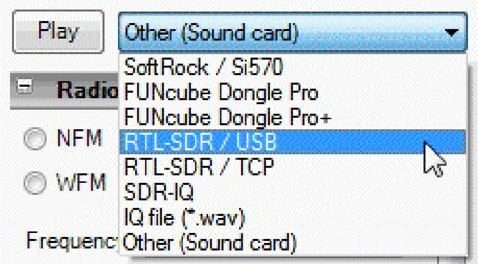


Windows can't verify the publisher of this driver software

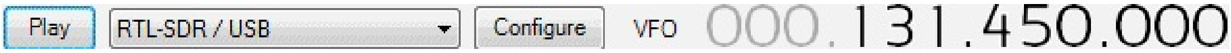
Don't install this driver software
You should check your manufacturer's website for updated driver software for your device.

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.

11. Open SDRSharp.exe. Set the drop down box at the top next to the Play button to **'RTLSDR / USB'**. Press Play. Your rtl-sdr software radio should now be playing some static and showing an RF spectrum and waterfall! If everything has worked you should be able to start tuning to frequencies using the numbers at the top of the program.



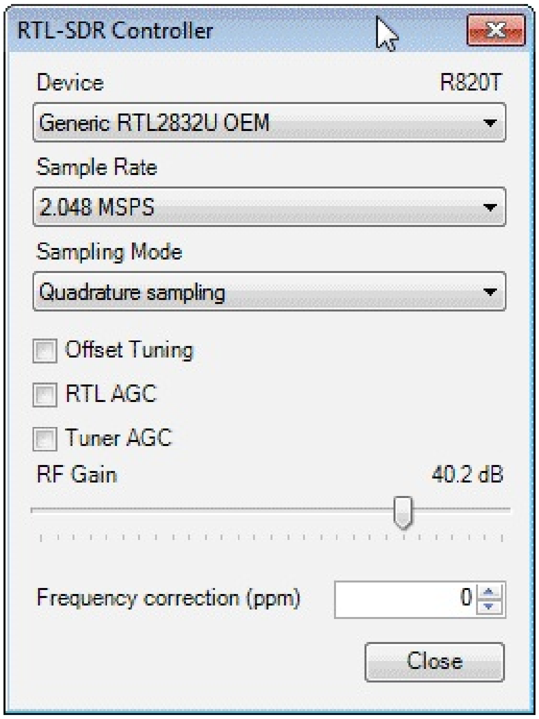
12. Click on the Configure button to bring up the configure menu.



RTL-SDR / USB

VFO 000.131.450.000

13. By default the RTL-SDR’s gain is set to zero. Increase this gain by moving the RF Gain slider up, or by turning on Tuner AGC.



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

TROUBLESHOOTING GUIDE

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. It may also show up as the brand of your dongle or something prefixed with “rtl”. This option should work too.

I DON’T SEE RTL-SDR/USB IN SDRSHARP

You may have downloaded a version without rtl-sdr support. Check that you followed the instructions in step 3 and downloaded the version under the heading “Important note for RTL-SDR users”.

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/.

RECEPTION IN SDR# SEEMS VERY POOR/RECEIVER IS INSENSITIVE

Make sure you have increased the 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.

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

This might mean that you do not have the .NET 3.5 Framework installed. 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 it out of range of the RTL-SDR. Tune back to a known supported frequency like 000.100.000.000 and press play.

WHAT IS BULK IN INTERFACE-1?

This is the IR interface for the remote control which is not required for SDR.

ZADIG WON'T RUN

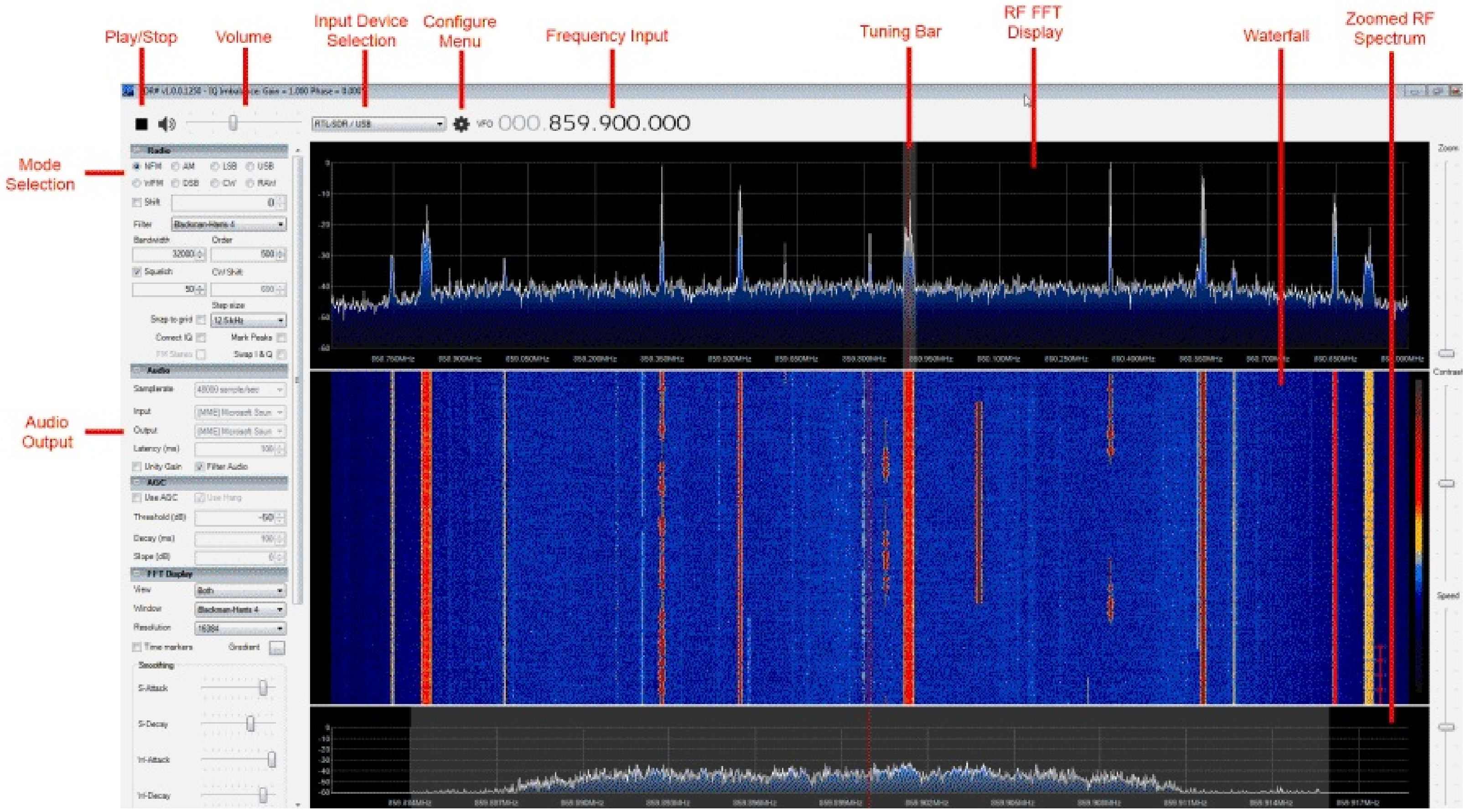
Some users report that Zadig crashes upon opening. An alternative driver installer can be found at http://visualgdb.com/UsbDriverTool/. Use this tool to install the libUSB - WinUSB drivers.

SDR SHARP USERS GUIDE

SDR# is currently the most popular SDR program used with the RTL-SDR. It is easy to setup and simple to use. To install SDR#, go through the RTL-SDR setup guide first.

Upon starting SDR#, we suggest you reduce the Range setting and increase the FFT resolution to 16384. See below for more information.

Open SDR#. You will be greeted with this screen shown below. Here we have highlighted the main parts of SDR#.



MAIN SETTINGS

PLAY BUTTON

This button is used to start and stop the SDR.

INPUT DEVICE SELECTION

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

Clicking this button opens up the configure menu. In here you can change settings like the sample rate (bandwidth) and RF gain.

FREQUENCY INPUT

Use the mouse to set the desired frequency in MHz 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.

MODE SELECTION

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

NFM - Narrowband Frequency Modulation. Common mode used by walkie talkie radios and digital signals.

WFM - Wide Band Frequency Modulation. Is the mode that broadcast FM stations use (eg the radio you listen to in your car).

AM - Amplitude Modulation. Used by broadcast AM stations that are receivable by normal radios and air band voice frequencies used by aircraft and air traffic control.

LSB/USB - Lower Side Band/Upper Sideband. Use in the HF band by ham radio to transmit voice efficiently.

CW - Continuous Wave. Used for listening to Morse Code.

DSB - Double Side Band. Not commonly used.

RAW - Raw signal. Almost never used.

VOLUME / AF GAIN

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

RF FFT SPECTRUM

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

RF FFT 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 shows where on the RF spectrum the RTL-SDR is currently tuned to. The tuning can be altered by simply using the mouse to click and drag the red line.

The shaded rectangular area around the red line shows the bandwidth of the tuned area. (don’t confuse this with the bandwidth/sample rate 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 by simply dragging the edges of the shaded area in or out.

OTHER SETTINGS

Other options which you may find useful are described below.

RADIO TAB

OFFSET

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 offset to be -100,000,000 (don’t forget the minus sign). Without the offset, to tune to a signal at 9 MHz you would need to actually tune to 100 + 9 = 109 MHz. With the offset 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 offset to -125,000,000.

FILTER BANDWIDTH

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

FILTER TYPE

Changes the filter type used. Different filters have different shapes. The filter is used to select the highlighted signal in the RF window. A good filter will select only the selected signal. Blackman-Harris is usually the best filter to choose.

FILTER ORDER

You may notice that with low filter orders signals outside of the tuned bandwidth can still be heard. Larger filter orders “tighten” or “sharpen” the band pass filter used in the tuned bandwidth 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.

SQUELCH

Mutes the audio when the signal level is below the specified value. Useful for when listening to speech as the static when no one is talking will be muted.

CW SHIFT

Mainly useful for when transmitting CW as it specifies the offset between CW transmit and receive frequencies.

SNAP TO GRID

In many bands signals are spaced a fixed distance apart. For instance in most countries air band signals are spaced 25 kHz apart (or 8.33 kHz in some countries). This setting can help with tuning by snapping 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.

CORRECT IQ

Should usually be selected as ON. This setting uses an algorithm to remove unwanted images (ghosts of strong signals) from showing up in the spectrum.

SWAP I & Q

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

FM STEREO

Will enable stereo output for broadcast radio WFM signals.

MARK PEAKS

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

AUDIO TAB

OUTPUT

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 cable to send the audio to.

AGC (AUTOMATIC GAIN CONTROL):

The AGC tab will only be activated in the AM, DSB, LSB, USB and CW demodulation modes.

USE AGC

Turns on the automatic gain control. The AGC will attempt control the volume level so that loud sounds are not too loud and quiet sounds are not too quiet. The default settings work well for audio signals.

FFT DISPLAY

VIEW

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.

RESOLUTION

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.

RANGE

Modifies the dB level range shown on the left axis of the RF spectrum window. You should adjust this so that the noise floor sits near the bottom of the RF spectrum window. As the RTL-SDR has a dynamic range of approximately 50 dB (plus a little more after oversampling), you will not need a range much higher than 70 dB.

DIGITAL NOISE REDUCTION

It is useful to turn this on when listening to noisy analogue voice signals. This setting will attempt to reduce the background 'hiss' sound. It is usually more useful to use the IF digital noise reduction.

NOISE BLANKER

The noise blanker is an algorithm that can be turned on to help reduce impulsive noise from sources like spark gaps. Examples of this type of noise may come from motors and electricity lines.

SETTING THE RF GAIN

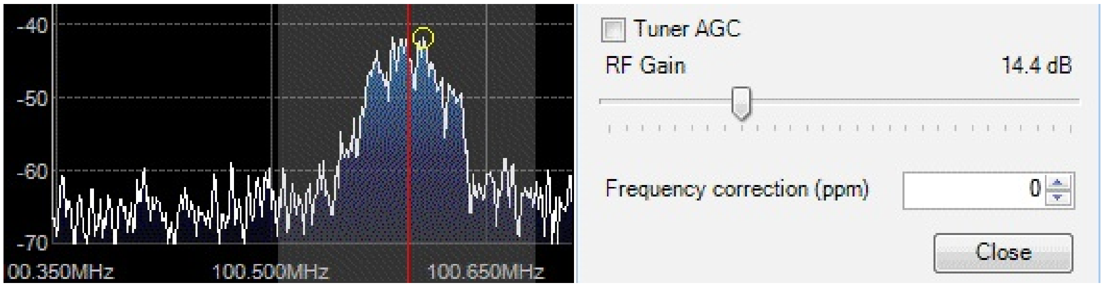
There are three RF gain settings that can be found by clicking on the Configure button. RTL AGC turns on the internal dongle automatic gain control algorithm. Tuner AGC uses the RTL-SDR drivers AGC settings and finally, the gain slider can be used to manually set the gain.

For casual browsing, Tuner AGC will likely suffice most of the time. When trying to optimize the gain of a signal, manual tuning will be required. RTL AGC is almost never used as it tends to introduce a lot of unwanted noise.

The goal when setting the gain is to try and get the signal to noise ratio 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 (SNR) by subtracting the peak signal height from the height of the noise floor.

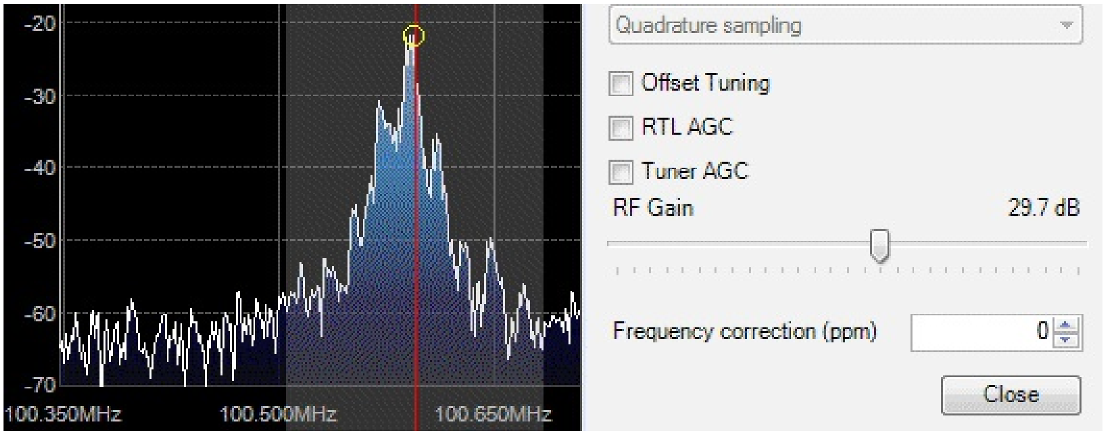


☐ Tuner AGC

RF Gain 14.4 dB

Frequency correction (ppm) 0

SNR: -41 - -65 = 24



Quadrature sampling

□ Offset Tuning

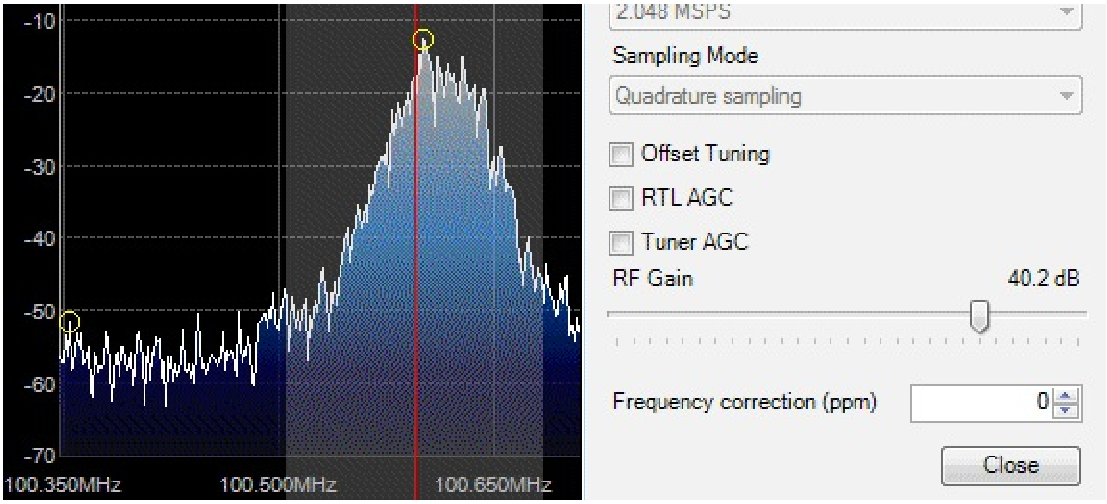
□ RTL AGC

□ Tuner AGC

RF Gain 29.7 dB

Frequency correction (ppm) 0

SNR: -21 - -65 = 44



2.048 MSPS

Sampling Mode

Quadrature sampling

□ Offset Tuning

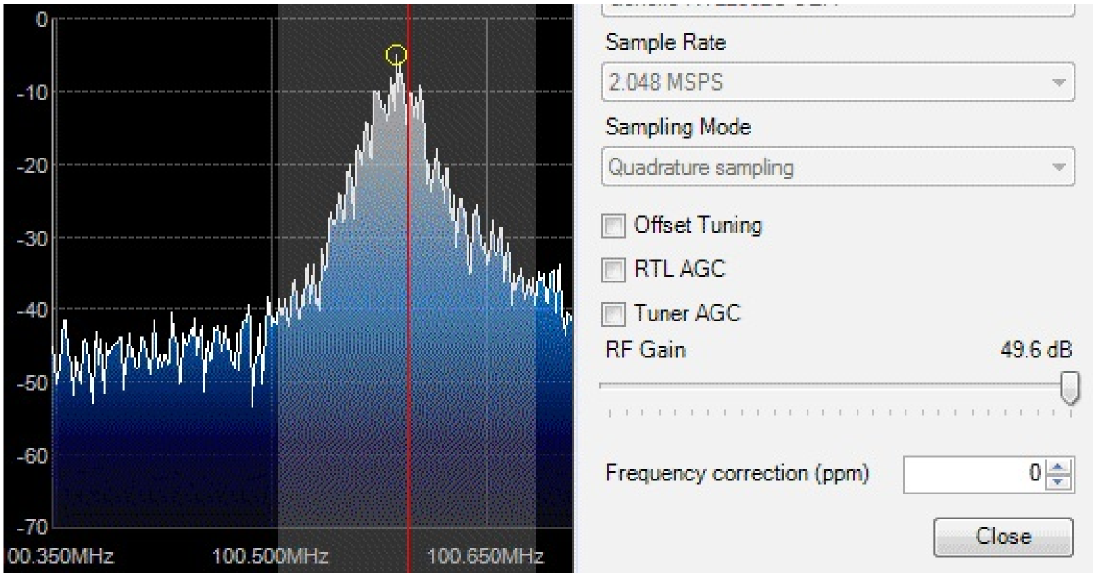
□ RTL AGC

□ Tuner AGC

RF Gain 40.2 dB

Frequency correction (ppm) 0

SNR: -12 - -57 = 45



Sample Rate

2.048 MSPS

Sampling Mode

Quadrature sampling

□ Offset Tuning

□ RTL AGC

□ Tuner AGC

RF Gain 49.6 dB

Frequency correction (ppm) 0

SNR: -5 - -45 = 40

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 that gain was optimal at around 40.2 dB.

SETTING THE PPM CORRECTION

PPM is a measurement which stands for parts per million. High quality 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 approximately +-100 PPM.

A large tolerance can mean that a known frequency you tune 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.

Fortunately, SDR# has an option to correct the frequency offset that you may encounter. To set the PPM offset, 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 signals, 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 slightly offset from the known signal. Click the Configure button. Now adjust the PPM offset value box accordingly so that the signal lines up exactly with the red tuning line.

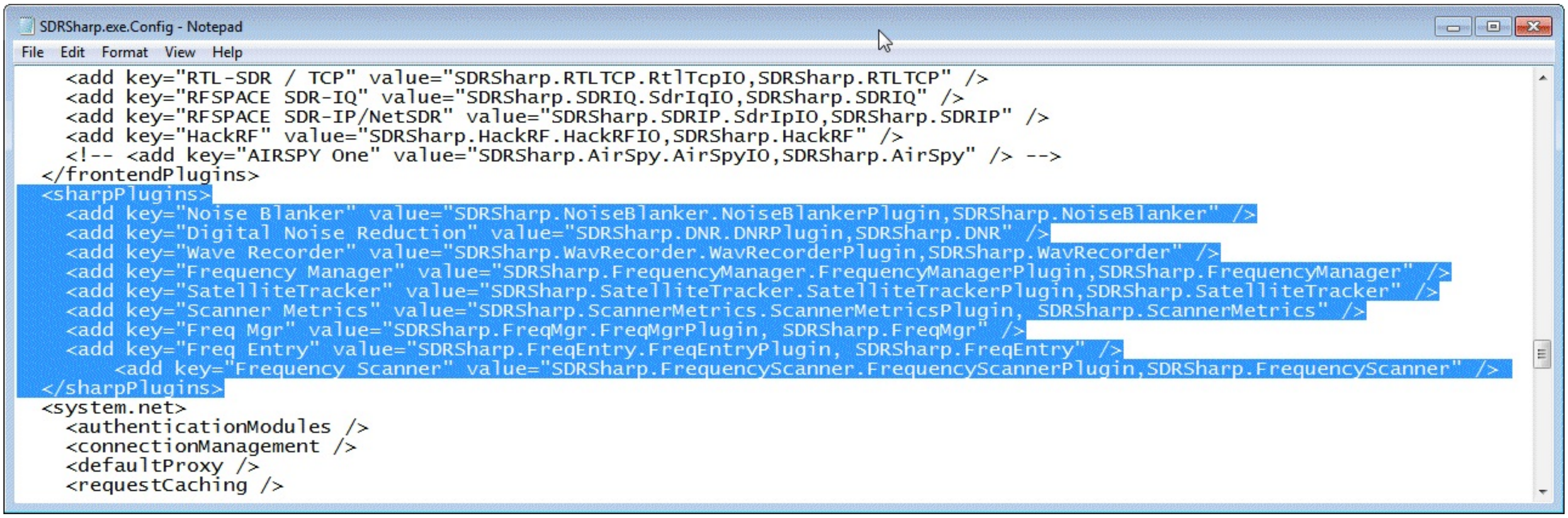
Another way to get an accurate PPM offset value is to use a Linux program called Kalibrate. This command line program uses mobile phone GSM signals to accurately detect the dongles offset. Obviously you will need GSM signals in your area and an antenna which can receive them for this to work. A tutorial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

SDR SHARP PLUGINS

There are many plugins available for SDR# that extend its functionality. We recommend visiting http://www.rtl-sdr.com/sdrsharp-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 SDRSharp.exe.Config 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 SDRSharp.exe.Config file.



```
    <add key="RTL-SDR / TCP" value="SDRSharp.RTLTCP.RtlTcpIO,SDRSharp.RTLTCP" />
    <add key="RFSPACE SDR-IQ" value="SDRSharp.SDRIQ.SdrIqIO,SDRSharp.SDRIQ" />
    <add key="RFSPACE SDR-IP/NetSDR" value="SDRSharp.SDRIP.SdrIpIO,SDRSharp.SDRIP" />
    <add key="HackRF" value="SDRSharp.HackRF.HackRFIO,SDRSharp.HackRF" />
    <!-- <add key="AIRSPY One" value="SDRSharp.AirSpy.AirSpyIO,SDRSharp.AirSpy" /> -->
  </frontendPlugins>
  <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="SatelliteTracker" value="SDRSharp.SatelliteTracker.SatelliteTrackerPlugin,SDRSharp.SatelliteTracker" />
    <add key="Scanner Metrics" value="SDRSharp.ScannerMetrics.ScannerMetricsPlugin, SDRSharp.ScannerMetrics" />
    <add key="Freq Mgr" value="SDRSharp.FreqMgr.FreqMgrPlugin, SDRSharp.FreqMgr" />
    <add key="Freq Entry" value="SDRSharp.FreqEntry.FreqEntryPlugin, SDRSharp.FreqEntry" />
        <add key="Frequency Scanner" value="SDRSharp.FrequencyScanner.FrequencyScannerPlugin,SDRSharp.FrequencyScanner" />
  </sharpPlugins>
  <system.net>
    <authenticationModules />
    <connectionManagement />
    <defaultProxy />
    <requestCaching />
```

SDRSHARP MINOUTPUT SAMPLE RATE

The maximum bandwidth of the tuned area for NFM in SDR# is restricted to 32 kHz. However, for many 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" />
```

COMPILING SDR# ON LINUX WITH MONO

Note: At this time this tutorial canot be used as the SDR# subversion repositories are down. They may come up again in the future.

First install RTL-SDR on Linux as shown in the Installing RTL-SDR Drivers on Linux section below. Next install Mono and the Mono Develop environment using the following

```
sudo apt-get install mono-complete
sudo apt-get install monodevelop
```

Also you will need to install libportaudio for sound

```
sudo apt-get install libportaudio-dev
sudo apt-get install libportaudio2
```

Next use svn to checkout the current SDRSharp trunk. If you don’t have svn installed use `sudo apt-get install subversion` to install subversion.

```
svn co https://subversion.assembla.com/svn/sdrsharp/trunk sdrsharp
cd sdrsharp
```

We first need to replace a version string in the SDR# solution file as Monodevelop currently does not recognize solution files created by Visual Studio 2012.

```
sed -i 's/Format Version 12\.00/Format Version 11\.00/' SDRSharp.sln
```

Open the solution file with monodevelop

```
monodevelop SDRSharp.sln
```

In Monodevelop set the build to Release|x86, then go to Build->Build All

Now we need to first symbolically link libportaudio to the Release folder where SDR# was just built.

```
cd Release
```

First we need to find out if you are running a 32-bit (x86/i686) OS or a 64-bit (x64/x86_64) OS. To do this type

```
uname -m
```

If you see the result as `i686`, you are running a 32-bit Linux so type in

```
ln -s /usr/lib/i386-linux-gnu/libportaudio.so.2 libportaudio.so
```

If you see the result as `x86_64` you are running a 64-bit kernel so type

```
ln -s /usr/lib/x86_64-linux-gnu/libportaudio.so.2
```

Add RTL-SDR support to the SDRSharp.exe.config file by using the command

```
sed -i 's/<!-- <add key="RTL-SDR \/ USB" value="SDRSharp.RTLSDR.RtlSdrIO,SDRSharp.RTLSDR" \/> -->/<add key="RTL-SDR \/ USB" value="SDRSharp.RTLSDR.RtlSdrIO,SDRSharp.RTLSDR" \/>/' SDRSharp.exe.config
```

Now you can run SDRSharp by typing

```
mono SDRSharp.exe
```

You may get an error about SDRIQ, but this can be safely ignored.

If you are interested in running SDR# on MacOS, it is possible, but requires a little work. A useful tutorial on this can be found at http://jasiek.me/2014/02/22/running-sdrsharp-on-mac-os-x.html.

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 more advanced programs as well.

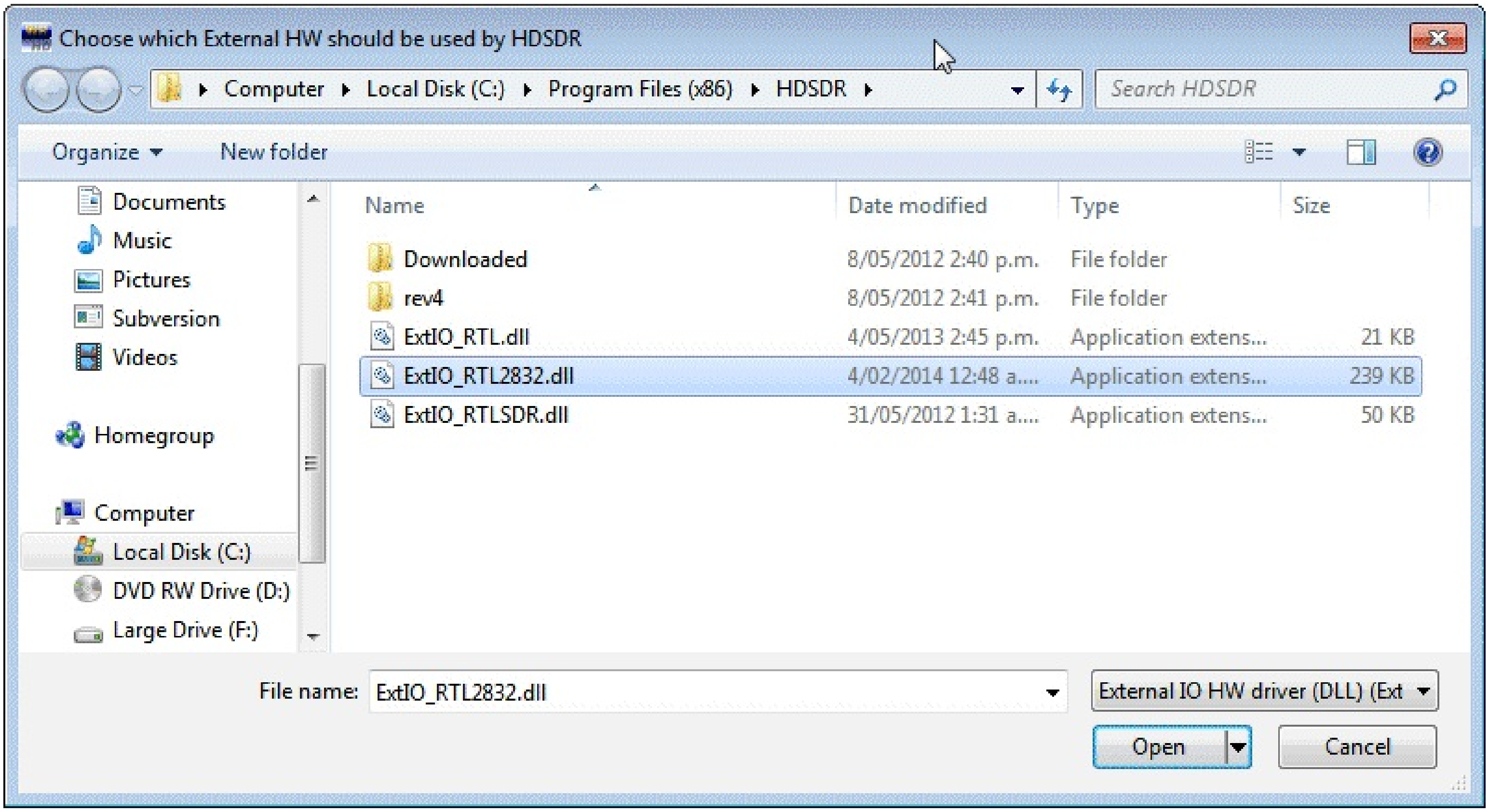
HDSDR SETUP GUIDE

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 advanced features. Users will also find an Audio FFT and waterfall display on the bottom of the screen. The output audio can also be bandpass filtered by dragging the filter borders on the 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 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.
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. If you are 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 ▸

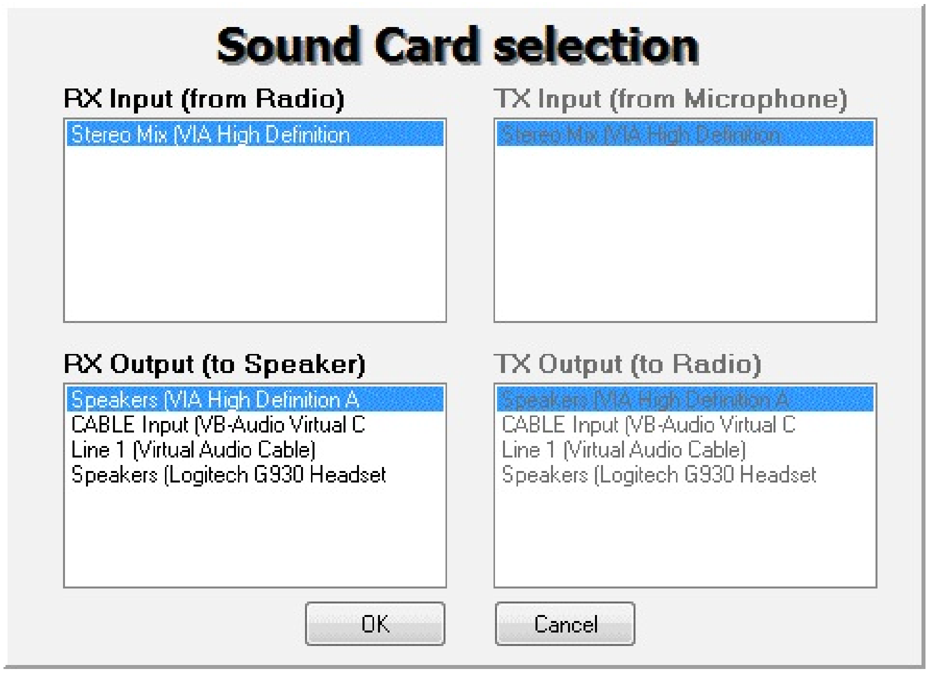
Name	Date modified	Type	Size
Downloaded	8/05/2012 2:40 p.m.	File folder	
rev4	8/05/2012 2:41 p.m.	File folder	
ExtIO_RTL.dll	4/05/2013 2:45 p.m.	Application extens...	21 KB
ExtIO_RTL2832.dll	4/02/2014 12:48 a....	Application extens...	239 KB
ExtIO_RTLSDR.dll	31/05/2012 1:31 a....	Application extens...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

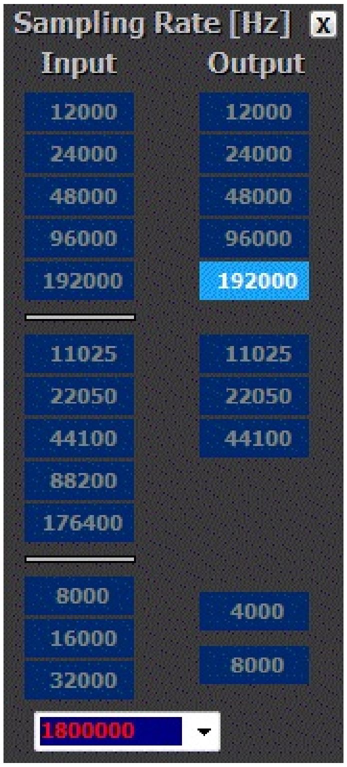
RX Input (from Radio)
Stereo Mix (VIA High Definition

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

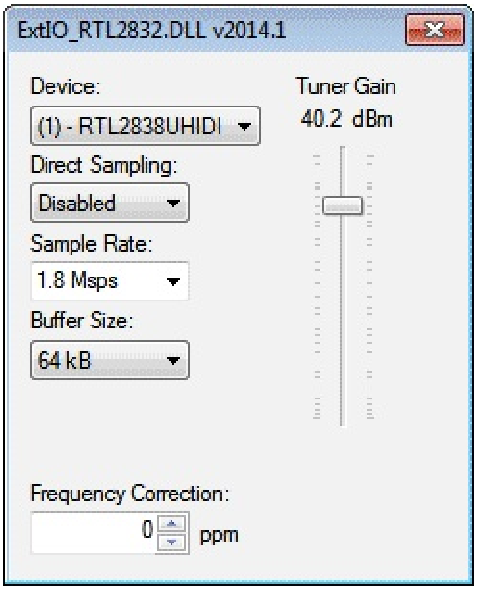
TX Output (to Radio)
Speakers (VIA High Definition A
CABLE Input (VB-Audio Virtual C
Line 1 (Virtual Audio Cable)
Speakers (Logitech G930 Headset

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

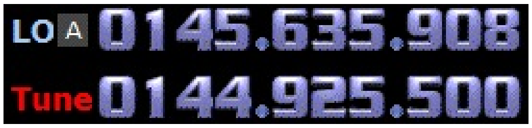
Sample Rate: 1.8 Msps

Buffer Size: 64 kB

Tuner Gain 40.2 dBm

Frequency Correction: 0 ppm

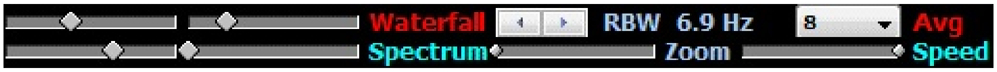
11. To tune to a station, change the Local Oscillator (LO) frequency to a frequency near the frequency you are interested in. Then tune to the desired frequency either by clicking in the RF spectrum, or using the numbers next to the red word Tune.



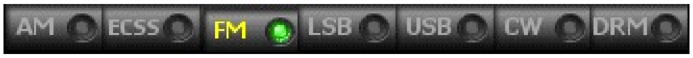
LO A 0145.635.908

Tune 0144.925.500

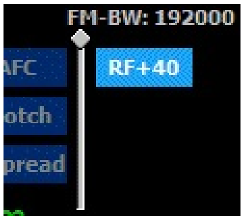
12. You can zoom in and out of the spectrum by using the Zoom slider which is to the left of the word zoom.



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

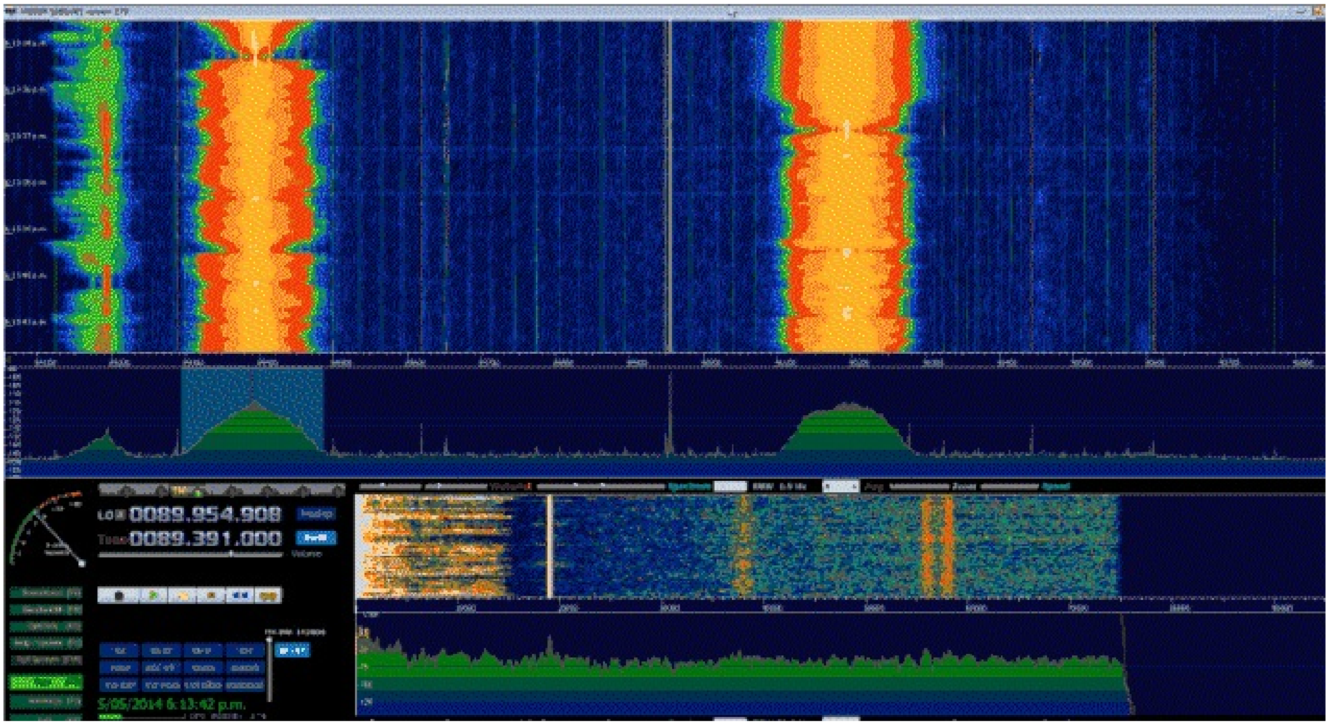


14. After clicking on the FM mode button, the FM bandwidth can be modified with the FM-BW slider. If this nslider is missing you may need to click on the FM mode button again.



FM-BW: 192000

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 pressing F6, and the selecting the output sampling rate as 192000 Hz and then using the FM-BW slider to increase the bandwidth.



LO 0089.954.908
Tune 0089.391.000

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.COM V2 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 can have six VFO’s.

To install SDR-RADIO for the RTL-SDR follow the steps below

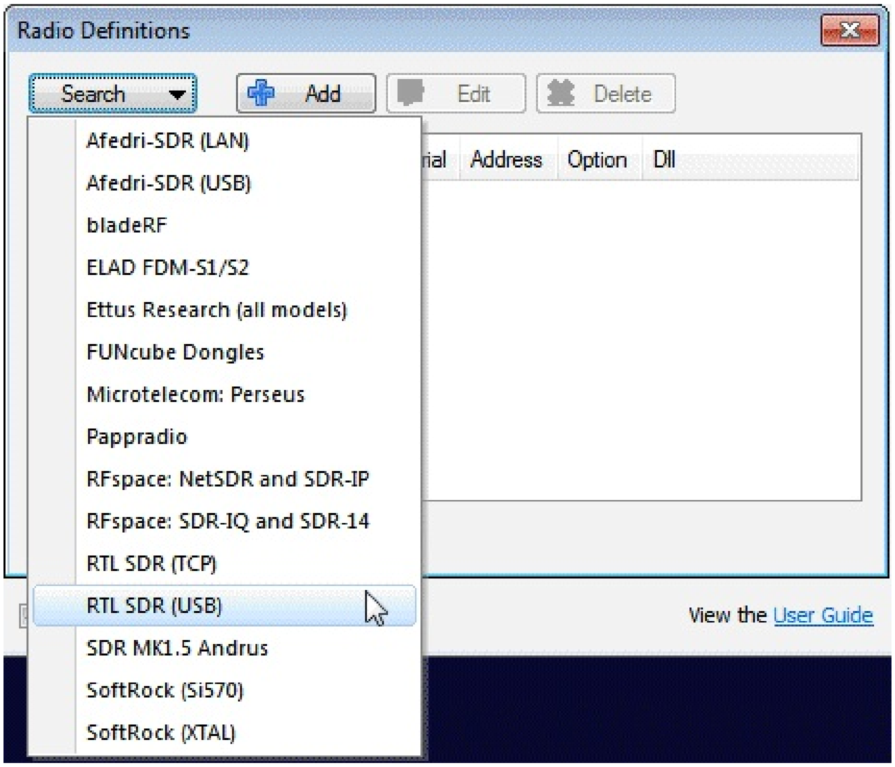
1. If you have not already, download and install zadig from http://zadig.akeo.ie/ and install the WinUSB drivers for RTL-SDR using the instructions in the SDR# setup guide above.
2. Download the SDR-RADIO installer from http://v2.sdr-radio.com/.

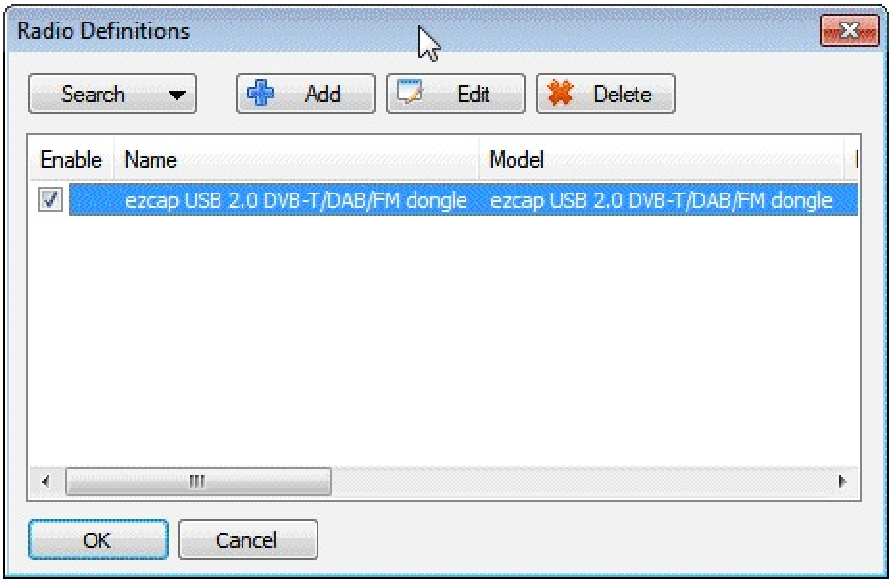
3. Use the installer to install SDR-RADIO.

4. Either compile yourself using the instructions from http://sdrnuke.wces.net/Support/RTLSDRs/BuildingDlls.aspx, or download from the bottom of http://www.aa5sh.com/?page_id=65 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.

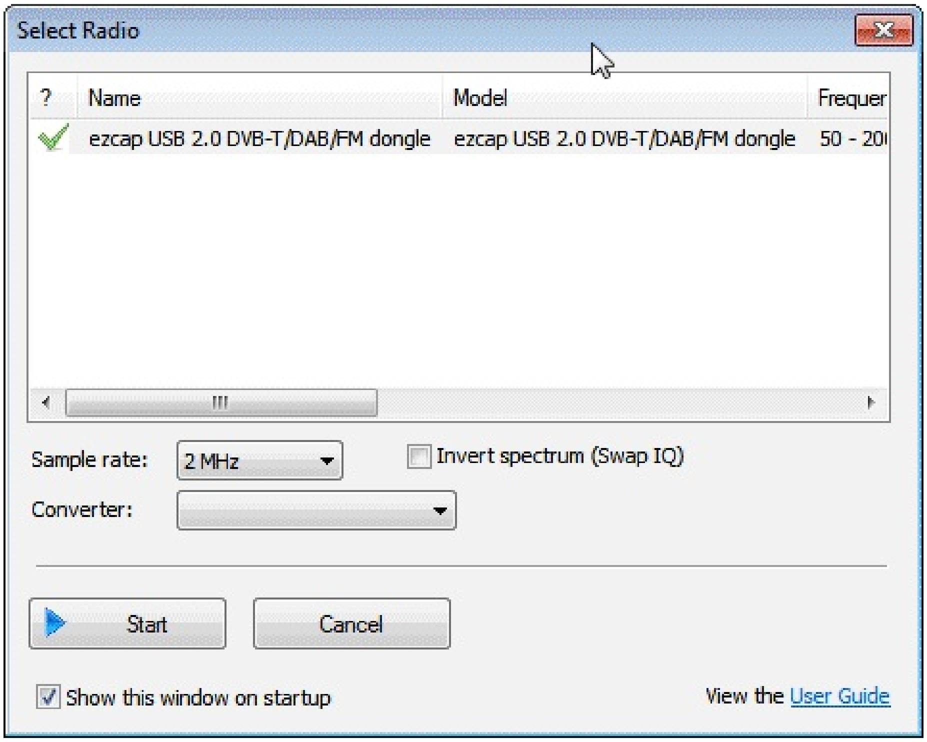
6. In the new window open the Search drop down menu and select RTL SDR (USB). After clicking it the RTL-SDR will be added to the Radio Definitions list. Click OK.





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 click to select it, choose your desired sample rate then click Start.



?	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

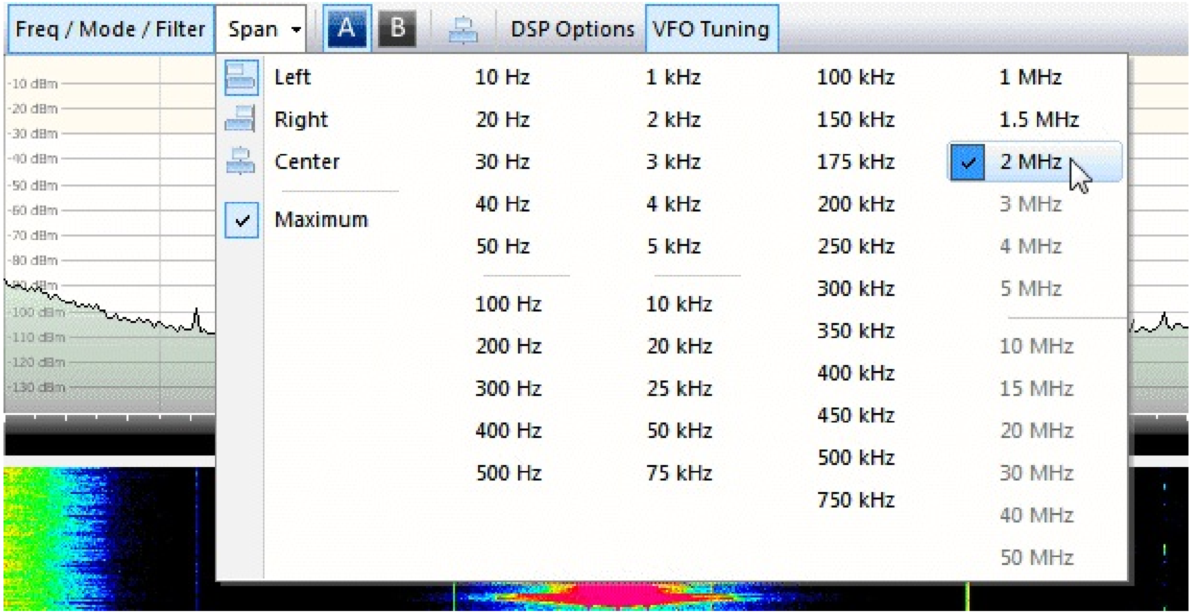
□ Invert spectrum (Swap IQ)

Converter:

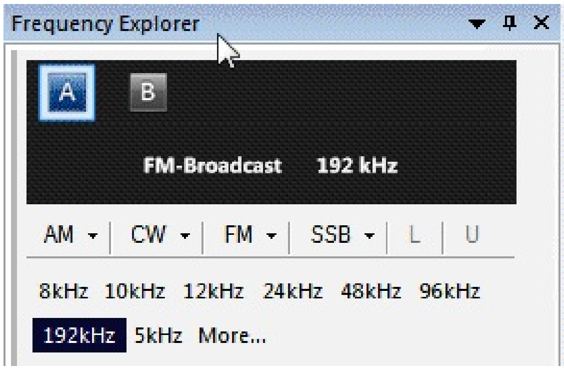
☑ Show this window on startup

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.

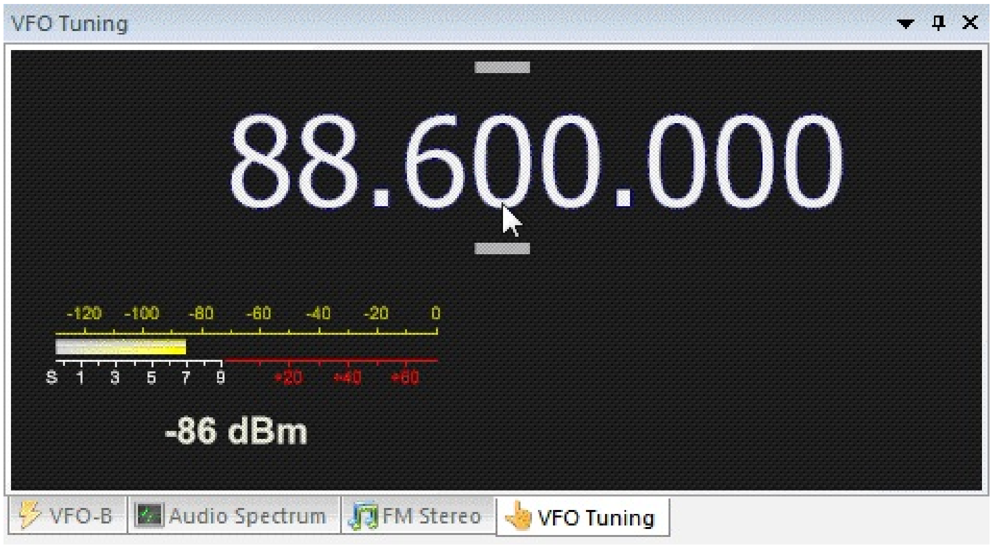


9. To change the receive mode use the left menu under the frequency tab.



FM-Broadcast 192 kHz

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



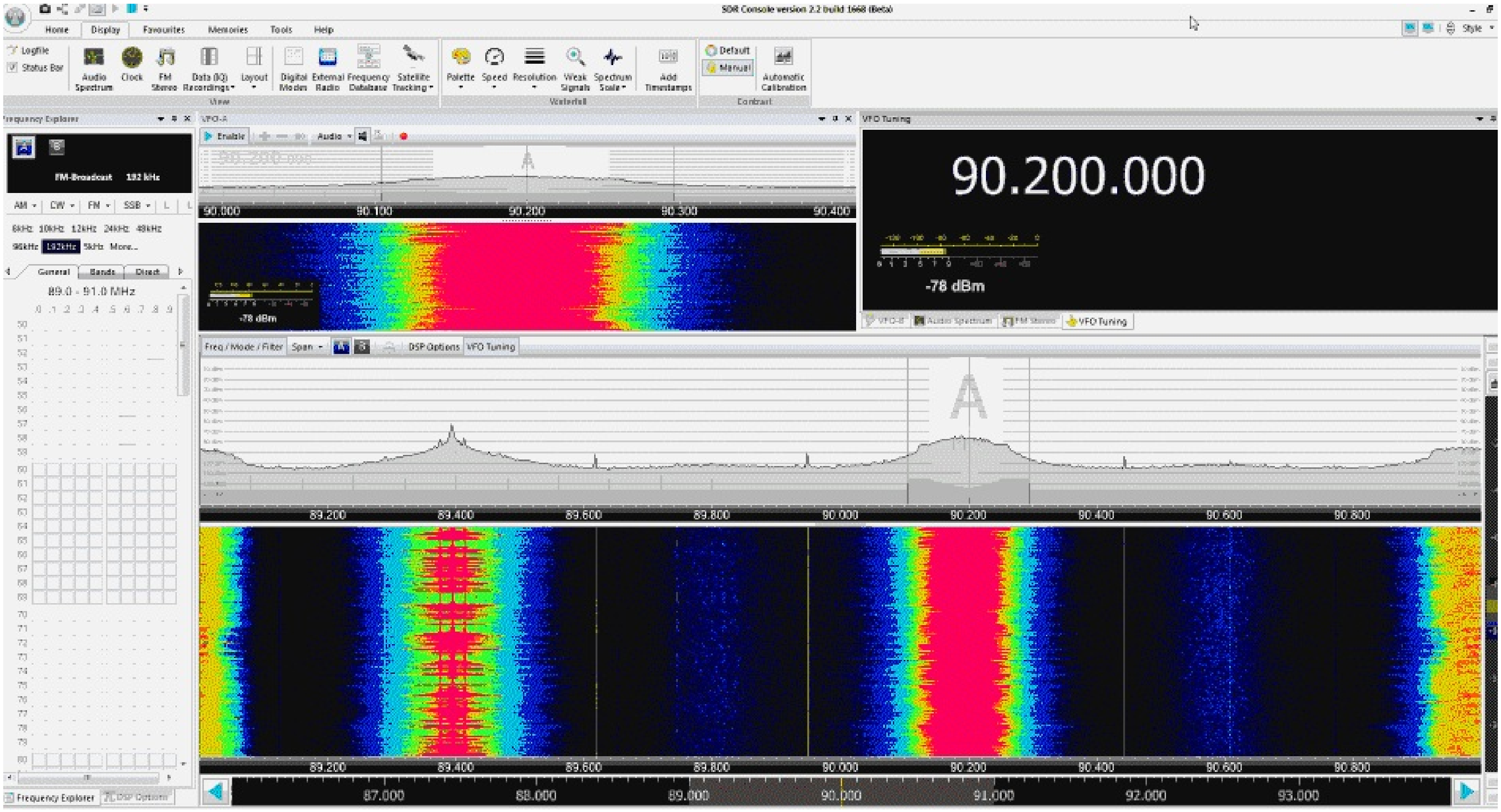
88.600.000

-86 dBm

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

12. Also to adjust the waterfall colors so that signals are 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 and features. To install GQRX follow these instructions:

To install GQRX on Linux 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
```

LINRAD

Linrad is an SDR receiver program that has some very advanced features that can run on older PCs

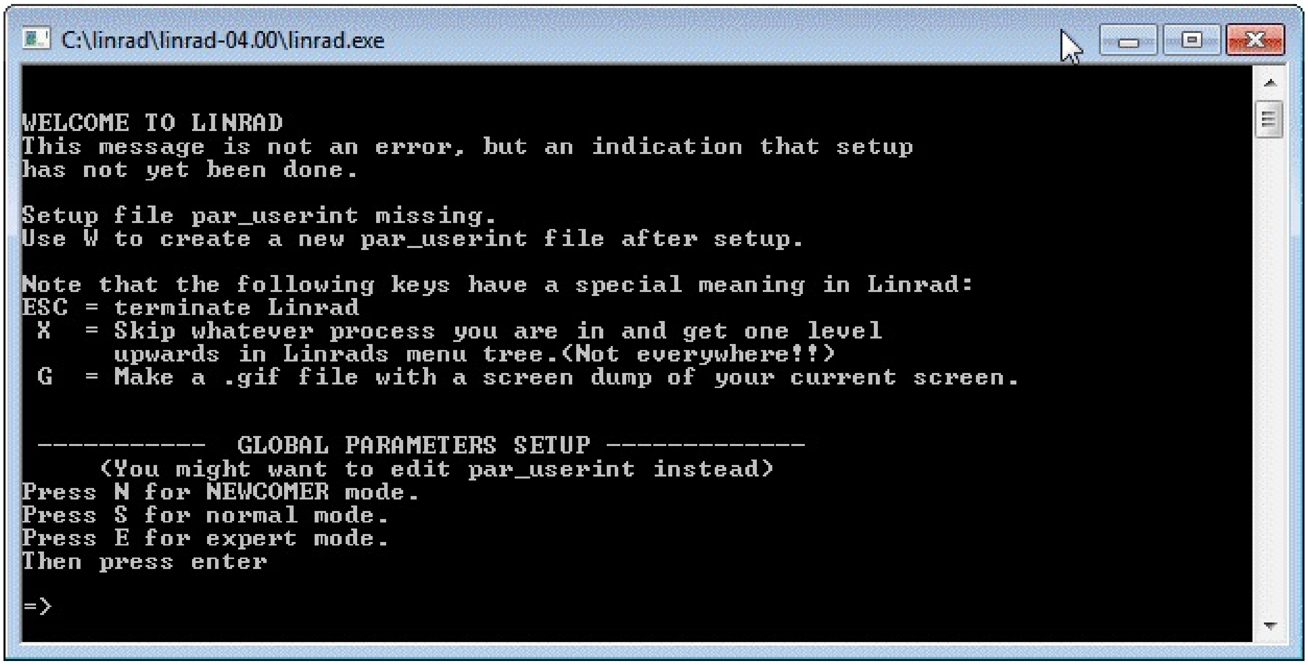
(Pentium 4’s). While Linrad has advanced features it also has a very difficult learning curve and is for advanced users only. This tutorial will show you how to set up Linrad and test it on Wideband FM radio. More advanced features require I/Q and filter calibration which requires a pulse generator.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

1. If you have not 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. 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. Then 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.

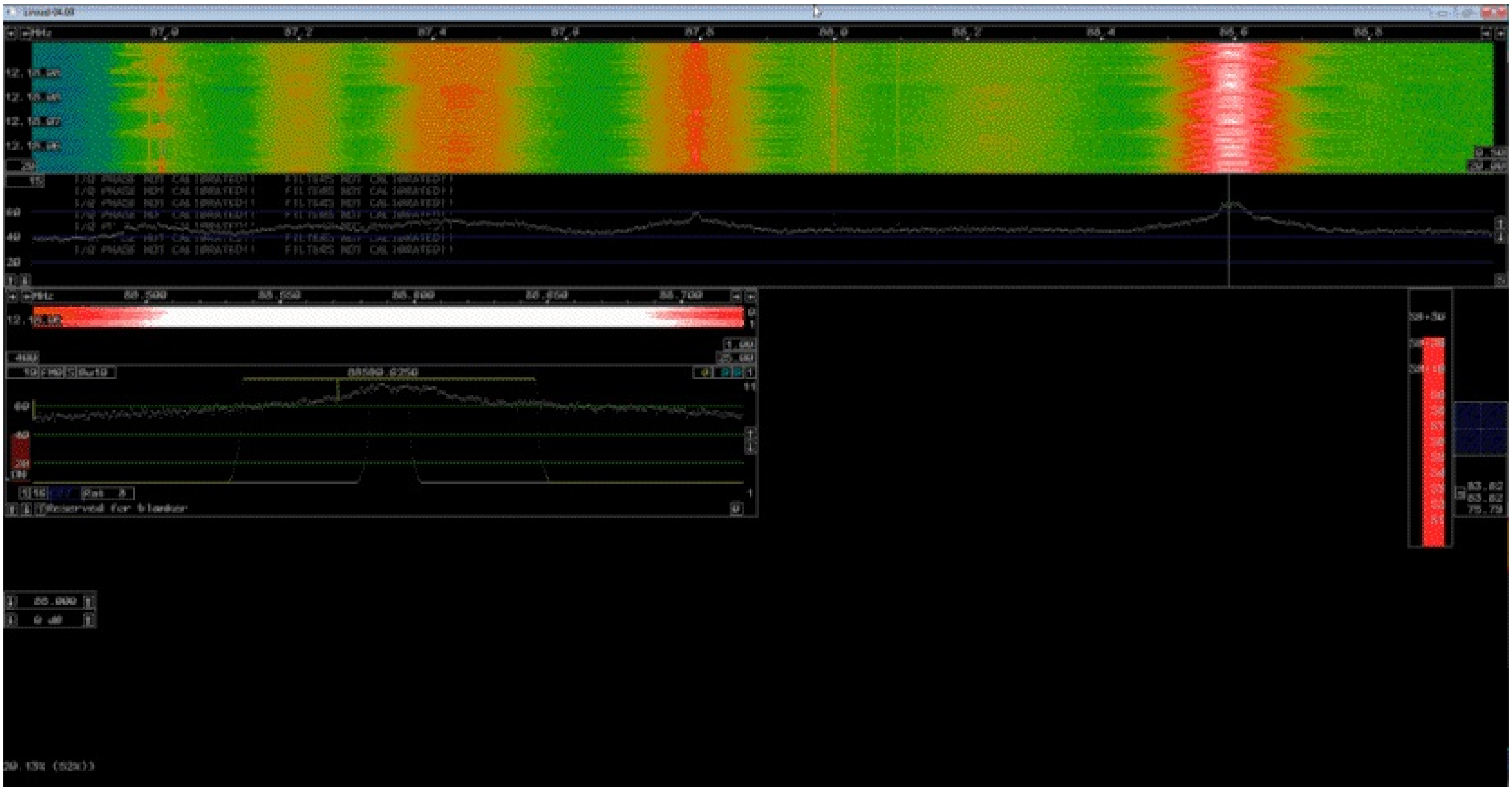
8. Next choose N to specify the window size in percentage. Then choose 100% width and around 90% for the height.

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 cable 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, go 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.



INSTALLING THE RTL-SDR DRIVERS ON LINUX

First update your distribution

```
sudo apt-get update
```

Now install git, cmake and libusb, the USB drivers required for RTL-SDR

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

Now download, compile and install the RTL2832U Osmocom drivers

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

Restart Linux.

Plug in the dongle and then run `sudo rtl_test -t` to test the dongle installation. If you get errors like

```
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.
```

This means that the RTL2832U DVB-T drivers are currently active. Disable them by running

```
sudo rmmod dvb_usb_rtl28xxu rtl2832
```

Running this command will only stop the DVB-T drivers once. When you remove and plug in the dongle again or restart, 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-rtl.conf" in /etc/modprobe.d. In that file put the text `blacklist dvb_usb_rtl28xxu`. Unplug and plug back in the dongle and it should now default to using the rtl-sdr drivers.

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

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.

To use the play command used in the examples below you will need to install sox. Sox can be obtained on Debian Linux with `sudo apt-get install sox`. On Windows it can be downloaded from http://sourceforge.net/projects/sox/. You may 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. Here 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 ouput sox command too in order to select the default audio device.

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. First we list all the possible RTL_FM flags and then show some examples.

RTL_FM FLAGS

```
-M set the receive mode (wbfm, fm, am, lsb, usb, raw)
-s bandwidth/sample rate
-g gain
-l squelch level
-t squelch delay
-p ppm offset
-d choose between multiple dongles attached to a PC
```

```
-F use better filters
-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)
```

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 -
```

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 -
```

You can increase the quality of the received signal by using the -F option at the expensive of increased computation time. 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.

Rtl_fm can also be used as a scanner. Simply add multiple -f frequencies of the channels that you wish to scan. Rtl_fm will scan until it finds an active signal.

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, 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 -
```

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

LINUX WITH PREINSTALLED RTL-SDR AND LIVE DVDS

There are several Linux distributions available now that contain the rtl-sdr drivers preintalled 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 install it. Most also come with a wide range of amateur radio and other RF tools.

KB1OIQ - ANDY'S HAM RADIO LINUX

Andy’s Ham Radio Linux is an amateur radio centric distribution. It contains 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/

KALI LINUX

Kali Linux is a security and penetration testing centric distribution. Of note Kali Linux comes with GNU Radio 3.6 (note 3.7 is the latest version).

http://www.kali.org/

PORTEUS

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

http://www.porteus.org/

GNU RADIO LIVE DVD

A Linux distribution base on GNU Radio programs. Contains 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

RTL-SDR MISC. INFORMATION

RTL-SDR CRYSTAL TOLERANCE

The RTL-SDR is a mass Chinese manufactured device with poor crystal tolerances. 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 for information on how to calibrate the RTL-SDR to remove the frequency offset.

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 dongle, the “Offset Tuning” option in the SDR# configure menu can be used. Offset tuning will do nothing for the R280T.

The R820T may also have a smaller 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, the frequency of the internal oscillator.

You may also see strong spurs at multiples of the USB clock frequencies (48 MHz).

There is unfortunately not much that can be done about this other to know where they are and ignore them. A spur looks like a signal thin solid line on the RF spectrum or waterfall. It will also likely sound like a single audio tone on the LSB/USB modes.

RTL-SDR IMAGES

At a bandwidth of 0.25 MHz there will be images of strong signals at every 250 kHz. At 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.

RTL-SDR FREQUENCY DRIFT

As the dongle warms up the crystal oscillator will so too causing the received frequency of signals to drift over time before stabilizing on a single offset value. Usually this is not a problem for most applications. The final frequency offset can be adjusted by setting the PPM offset value.

USING MULTIPLE DONGLES WITH THE SAME SERIAL NUMBER

Some dongles such as the hama nano may all have the same serial number. This means that you can only plug in one dongle at a time into your PC. You can adjust the serial number by using rtl_eeprom which is an 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.

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.

RTL-SDR IMPROVEMENTS AND MODIFICATIONS

LOW NOISE AMPLIFICATION

The amplifier used in the dongle is ‘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 LNA will introduce into the signal. By using a high quality LNA with a noise figure <1dB this noise can be reduced.

One good LNA that RTL-SDR.com recommends is the LNA4ALL. It’s link can be found on our Buy RTL-SDR Dongles page 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. Surface mount components are used because they produce the least amount of noise. A DIY LNA design for the RTL-SDR can be found here https://github.com/loxodes/rtl-sdr-lna.

When placing 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, to help reduce the effects of noise that can enter through the coax transmission cable and attenuation over long cable runs. In other words, it makes the desired signals bigger at the antenna so that any noise or losses introduced by the coax cable will be negligible.

An LNA will be of most use at frequencies above 50 MHz. At HF frequencies (<50 MHz) the ambient RF noise is so high that an 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.

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

No RTL-SDR dongle will tune to the entire LF/MF/HF band (0-30 MHz) without modifications or extensions. There are two methods that will allow your RTL-SDR to receive LF/MF/HF frequencies.

1. The direct sampling mode modification

2. An upconverter circuit

1) DIRECT SAMPLING MODE

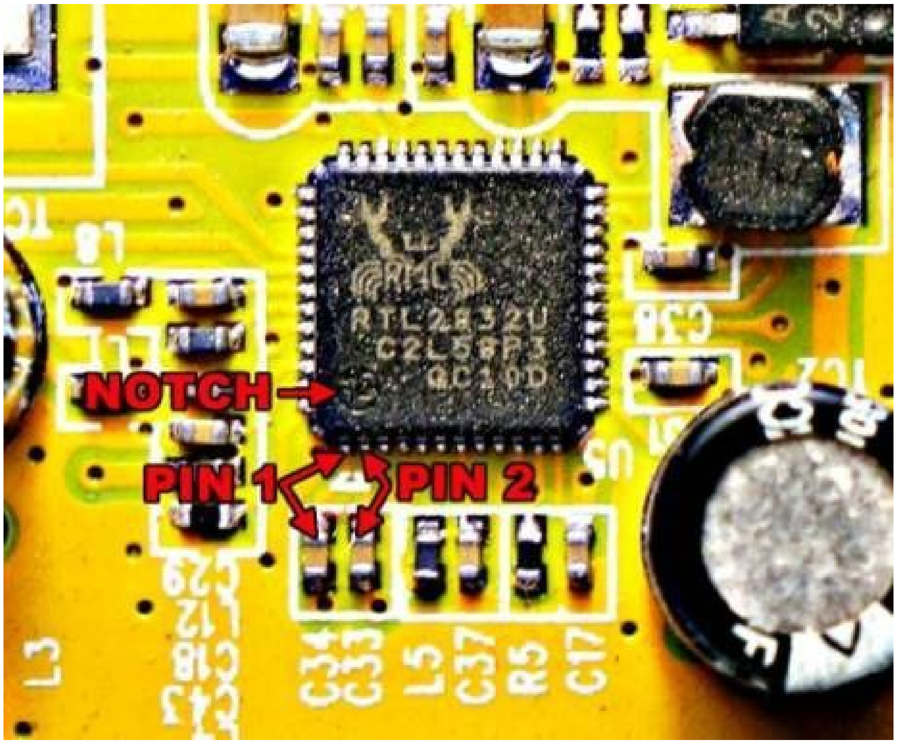
The rtl-sdr can be told to run in a mode called “direct sampling mode”, which with a small hardware mod allows the dongle to tune to the LF/MF/HF frequencies where ham radio and many other interesting signals are found. No upconverter circuit is required with this mod.

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. This is because an upconverter introduces some conversion losses.

This 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 bypassing any electrostatic protection the dongle has. However, in practice damaging the chip through static discharge is a rare occurrence if care is taken.

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 pin one/two or pin four/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.

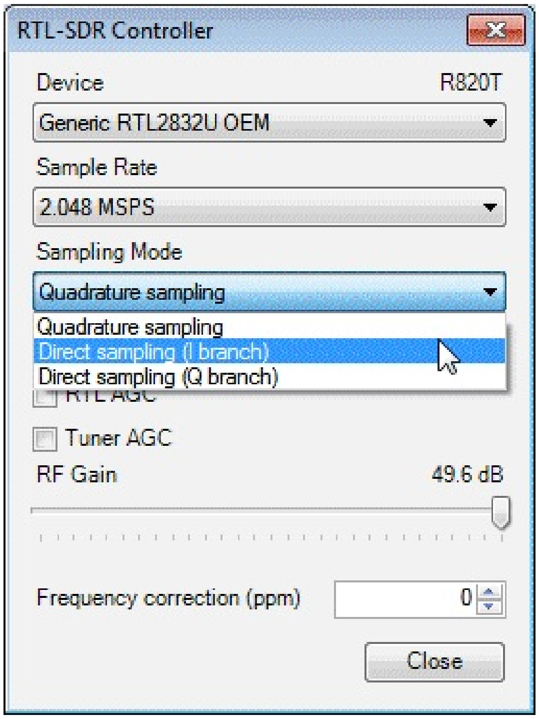
Pins one and two are located below the circle indentation on the RTL2832U chip. It might be easier to solder the wire to one of the two capacitors which are connected directly to pin one and two.



If you are using an R820T it may be best to use pins 4 or 5 for this mod, as due to its architecture these pins are actually unused. This will allow you to use normal Quadrature sampling for VHF/UHF and direct sampling as well for HF on the same dongle. The problem with these pins is that they do not have a capacitor connection, thus making them very hard to solder to.

To activate the mod, you will need to change some software settings. SDRSharp has this adjustment as a built in option. In SDRSharp, go to configure, and change the sampling mode to “Direct sampling (I branch)”, or ”Direct sampling (Q branch)”. The I branch corresponds to pin one/pin two, and the Q branch corresponds to pin four/pin five which can also be used. Pin three is not connected to anything.

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. However, frequencies between 14.4 and 28.8 MHz can be received with a band pass filter.



RTL-SDR Controller

Device R820T

Generic RTL2832U OEM

Sample Rate

2.048 MSPS

Sampling Mode

Quadrature sampling

□ Tuner AGC

RF Gain 49.6 dB

Frequency correction (ppm) 0

The mod described above can be improved on further. Since the RTL2832U uses two pins to create a differential input, a balun can be used to connect both pins one/two or four/five to an antenna. 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.

As the RTL2832U input impedance is unknown, it is difficult to match impedance with the antenna. In Dekars mod (http://mikikg.files.wordpress.com/2012/08/rtl2832u-dc-mods.pdf) , he assumed a 1:2 impedance conversion would help, but it actually made the signal worse. A 1:4 impedance conversion seems to be the accepted value though.

Most experimenters of this mod find that FM interference can be a problem and thus extra low pass filters might be necessary.

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. One good one is the one by KN0CK which can be found at http://www.easy-kits.com/.

DIRECT SAMPLING WITH NO HARDWARE MOD DISCOVERY

There is also a discovery found by Reddit 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. Unfortunately this was patched in the newer rtlsdr drivers.

However, there is still a way to test this. An RTL-SDR developer, tejeez has merged code into another developer, Keenerds experimental rtlsdr 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 instructions 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. This mode is no where near as sensitive as the true direct sampling hardware mod or with an upconverter, but it may be useful for those who just want to listen to AM.

2) UPCONVERTERS

Upconverters convert 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 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. Most upconverters use a local oscillator at around 100 MHz to 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, faint images can still show up. Theoretically, the low pass filter should prevent any interference from FM bands, but some very strong stations can overcome the filter. Most upconverters now use a frequency of 125 MHz, which avoids FM broadcast stations.

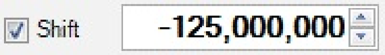
There are now several commercial upconverters available. One that we recommend is the Nooelec Ham-it-up upconverter. This upconverter is based on an opensource design and is of very high quality production and decent price. It increases the frequency of all signals by 125 MHz, and has a low pass filter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

There are also modified RTL-SDR dongles with built in upconverters. One can be found at http://janielectronics.com/ and another at http://www.easy-kits.com/.

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 setting in the receiver. The offset setting will allow you to tune directly to your desired frequency. In SDR# you can set the Offset value to -125,000,000 for the ham-it-up (don’t forget the minus sign).



☑ Shift -125,000,000

POSITIONING YOUR RTL-SDR DONGLE

Traditionally, 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 signal quality. Therefore the rtl-sdr dongle should be situated right next to the antenna, 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 USB cable that will work is about 5 meters (16 feet), and for USB 3.0 about 3 meters. This limitation is due to timing (although poor quality cable may also not work due to voltage drops). USB 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 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 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 runs of cable 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 running a Linux server is placed next to the dongle. The embedded PC is plugged into the rtl-sdr dongle, which then can be to used to transmit the radio data via TCP/IP (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 MS/s of data, 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 MS/s.

To do this you will need a little Linux and networking experience. But if you are good with computers it will be easy to figure out. Embedded computers such as the Raspberry Pi and Beaglebone have been reported to work well with this solution.

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

RTL-SDR ON A RASPBERRY PI/BEAGLEBONE OR OTHER EMBEDDED COMPUTER

Note that there appears to be a bug in newer Linux kernels that can cause stuttering problems 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”. To install RTL-SDR on a Raspberry Pi or other Embedded PC follow these instructions.

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

```
git clone git://git.osmocom.org/rtl-sdr.git
cd rtl-sdr/
mkdir build
cd build
cmake ../
make
sudo make install
sudo ldconfig
```

Embedded computers make great RTL-SDR servers as they can be placed in remote locations near an antenna and be used to broadcast a signal over a network or internet to a receiving station. To use as a server your embedded computer will need to have some form on Linux installed. Installing RTL-SDR on an embedded computer uses the same method as installing it on a regular Linux PC.

Note for Raspberry Pi users: Since the Rpi uses a USB based Ethernet controller, its Ethernet speed may not be sufficient to keep up with the data rate of rtl_tcp. We recommend using the streaming fm guide in the tutorials section to stream FM.

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 rtl_fm which allows the frequency and modulation tuned in rtl_fm to be modified remotely via a simple web interface. This is very useful for remote embedded servers.

See the section Using RTL_TCP for information on remotely transmitting I/Q data from the Raspberry Pi RTL-SDR server to a remote PC.

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. If 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 enter the other components.

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. 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 a waterfall.

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.

Noise Reduction Tip: *Never plug the dongle directly into the PC. Always use an USB extension cord with multiple ferrite beads.*

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.
- Ethernet over power.

DONGLE USB CABLE GROUNDING IMPROVEMENT

The rtl-sdr dongle has an electrical grounding design flaw. In USB devices the USB cable shielding is usually not meant to be connected to the electrical ground on the device, but rather to a separate ground. However, in the rtl-sdr, the USB cable shield connection *is* connected to the electrical 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 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.

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.

References:

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

USB CABLE FERRITES

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

Clip on ferrite's 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 wideband SDR radios has no preselector at the front end. A preselector is an electronic hardware circuit which filters out any frequency that is not in the filters passband. It may also be called a bandpass/lowpass/highpass filter. A bandpass filter will only allow signals to pass through within a certain designed band, for example between 1 GHz and 1.1 GHz for ADS-B. A low pass filter will only allow signals below a certain frequency and a high pass filter will only allow signals above a certain frequency.

Using a preselector on a wide band device like the RTL-SDR would mean that only a small area of the 22-2200 MHz receivable range would be usable.

However, preselectors will drastically reduce interference in the tuned frequencies. Most analogue radios 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 circuits for each chunk of frequencies you'd want to look at. This is because you would need different values for the analogue components depending on the frequencies you are interested in.

But still, if you are building a system for a single purpose, like only for receiving at ADS-B signals for instance, you will want to use a bandpass 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/second to 1200 messages/second with a bandpass preselector enabled. Most good upconverters will also use a lowpass preselector filter to filter out interfering signals above 30 MHz.

An alternative to a band pass filter for the preselector is a band stop filter. Instead of just passing a specific band, a band stop filter will block or "stop" a certain range of frequencies. Typically, you'd want a band stop filter to block the broadcast FM and/or pager bands, as these strong local signals are what cause most interference. These FM bandstop filters are sometimes referred to as "FM Traps".

You can buy good band stop FM filters from Miniciruits, search for the NSBP-108+. Miniciruits also has the BHP-400 which is a high pass filter which will block all frequencies below 395 MHz. These filters are unfortunately very expensive. Cheaper alternatives can be found on Amazon 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 bandpass/bandstop 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.

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

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.

To make a coax stub filter you will need a "T" junction connector. The coax 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:

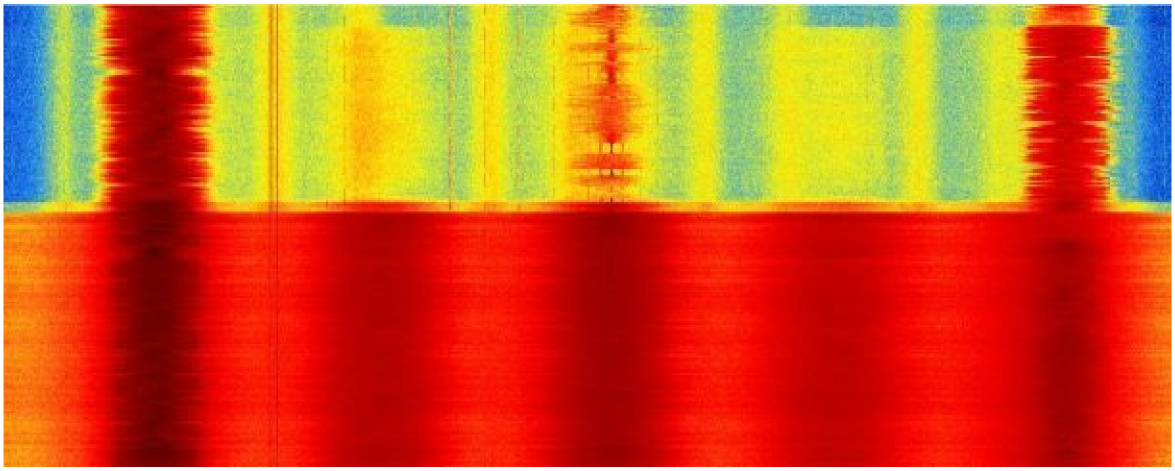
```
stub length = 300/band stop freq * coax velocity factor * ¼.
```

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

IMPROVING THE STOCK ANTENNA

The performance of the stock antenna that comes with many RTL-SDR units can be improved simply by placing it on a metallic surface that has a radius similar to the length of the whip, which for the standard antenna is around 12 cm.

Placing it on a metallic surface completes the antenna as a ¼ wave ground plane. Below we show an image comparing the antenna without (bottom) and with (top) a metallic ground plane. The noise floor is drastically reduced with the ground plane.



CALIBRATING 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 can be adjusted from the “Configure” menu.

It is also possible to calibrate the RTL-SDR automatically using a Linux command line program called Kalibrate. 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 to get it to work. To compile and install on Linux use kalibrate use the following instructions.

1. Download the kalibrate-rtl repository from the GitHub either by using clone or downloading and extracting the zip file from the page.

2. In a terminal navigate to the kalibrate folder. Type the following into the terminal.

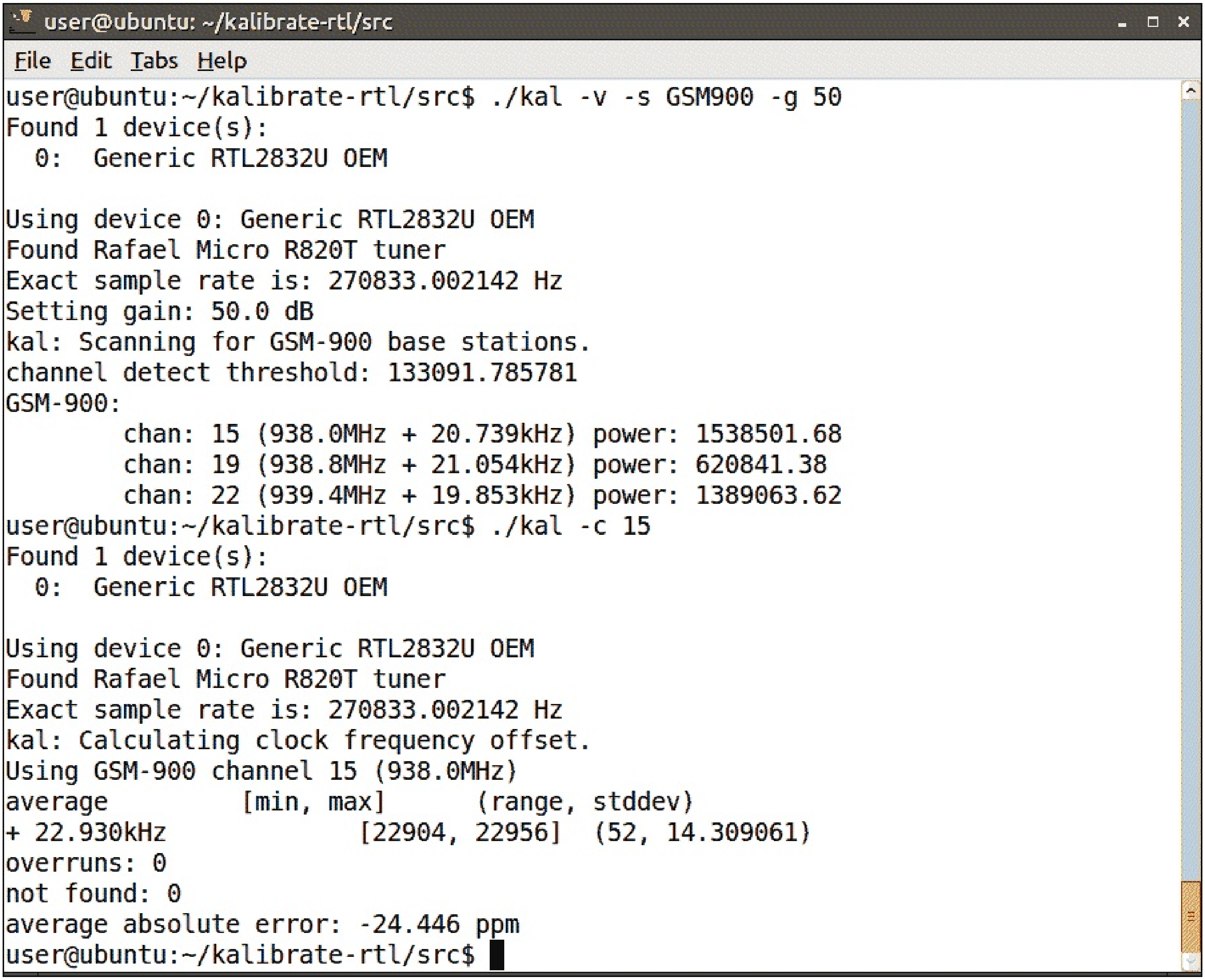
```
./bootstrap && CXXFLAGS='-W -Wall -O3'
./configure
make
make install
```

3. Now kal is ready to use by typing the following where GSM900 can be replaced by GSM850, GSM-R, GSM900, EGSM, DCS, PCS depending what GSM systems are in use in your particular country. Kalibrate will run for a while searching for GSM stations.

```
./kal -s GSM900
```

4. Once it has found some GSM channels, choose a channel from the output displayed in the terminal and then run the following to determine the frequency offset in PPM.

```
./kal -c channel
```



```
user@ubuntu:~/kalibrate-rtl/src$ ./kal -v -s GSM900 -g 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: 133091.785781
GSM-900:
        chan: 15 (938.0MHz + 20.739kHz) power: 1538501.68
        chan: 19 (938.8MHz + 21.054kHz) power: 620841.38
        chan: 22 (939.4MHz + 19.853kHz) power: 1389063.62
user@ubuntu:~/kalibrate-rtl/src$ ./kal -c 15
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 15 (938.0MHz)
average         [min, max]      (range, stddev)
+ 22.930kHz             [22904, 22956]  (52, 14.309061)
overruns: 0
not found: 0
average absolute error: -24.446 ppm
user@ubuntu:~/kalibrate-rtl/src$ 
```

RTL-SDR PROJECT TUTORIALS

These tutorials are written with the RTL-SDR in mind but 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 from your SDR receiver (e.g SDR#, SDR-Console) to the decoding program. See the Audio Piping Appendix 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.
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	
Radio Controlled Toys	72 - 76	Digital Data	
Broadcast FM	88 - 108	WFM Voice	
Air Traffic Control	108 - 136	AM Voice	
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 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.
Trunking Radios	~860	NFM Voice/Data	
GSM	850 900 1800 1900	Digital Data	Cell Phones.
Cordless Phones	900	Digital Voice	
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 flight crew. Some ACARS messages will also contain aircraft GPS coordinates, but if your goal is aircraft radar, the ADS-B project 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. ACARS uses vertically polarized signals. We recommend a discone, collinear, ¼ wave ground plane or J pole antenna. See the antenna guide for more information on these antennas. An LNA and bandpass filter may also help but are not necessary, 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.

Here are 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 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.

You will also need to have installed a Windows audio piping program. See Appendix A for more information.

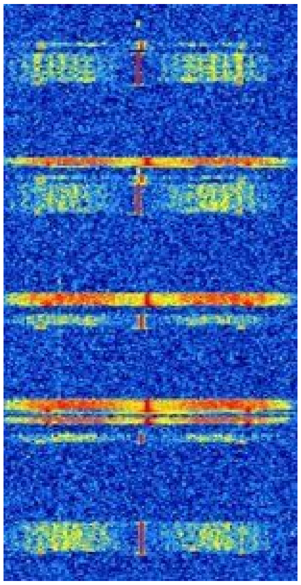
The first steps are listed below.

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 as you can output up to 6 potential ACARS channels to the same sound device. For SDR-RADIO set up as many VFO’s as ACARS channels you have transmitting and tune to all of them.

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

3. 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.

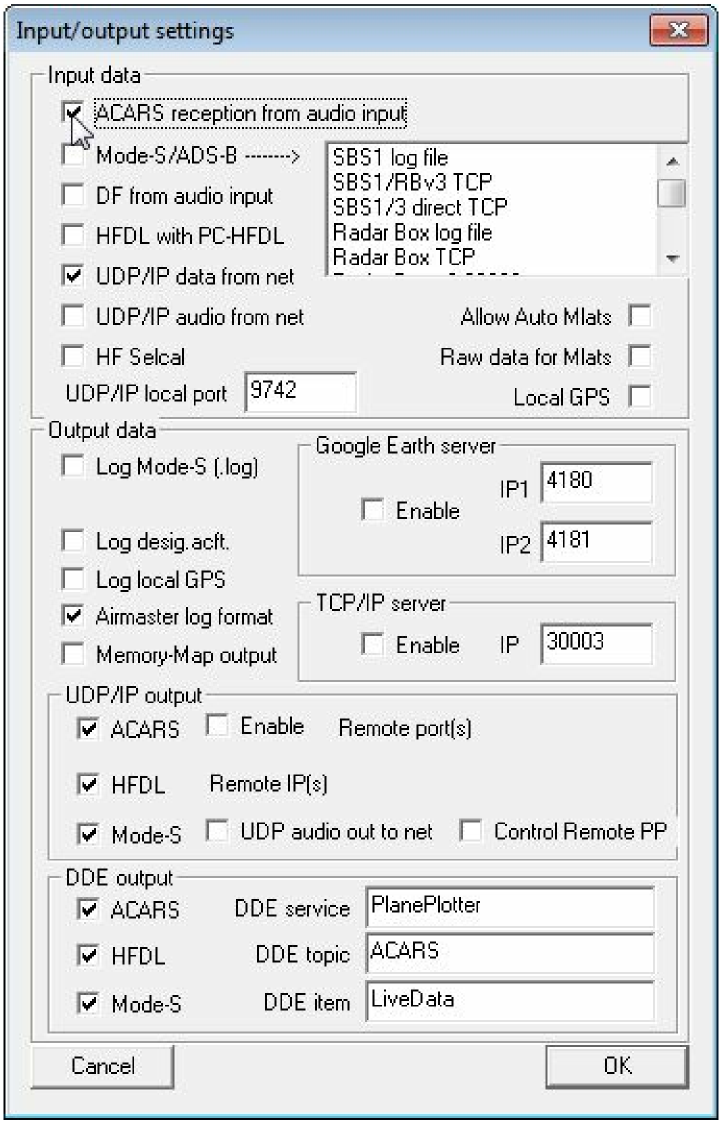


There are four commonly used ACARS decoder software programs, PlanePlotter, AirNav ACARS 2, ACARSD and KG-ACARS. Below we present tutorials on each software.

PLANEPLOTTER

The first one is PlanePlotter, which is a paid software package that costs 25 Euros. However, there is also a 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, 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)

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

☐ Log desig.acft.

☐ Log local GPS

☑ Airmaster log format

☐ Memory-Map output

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

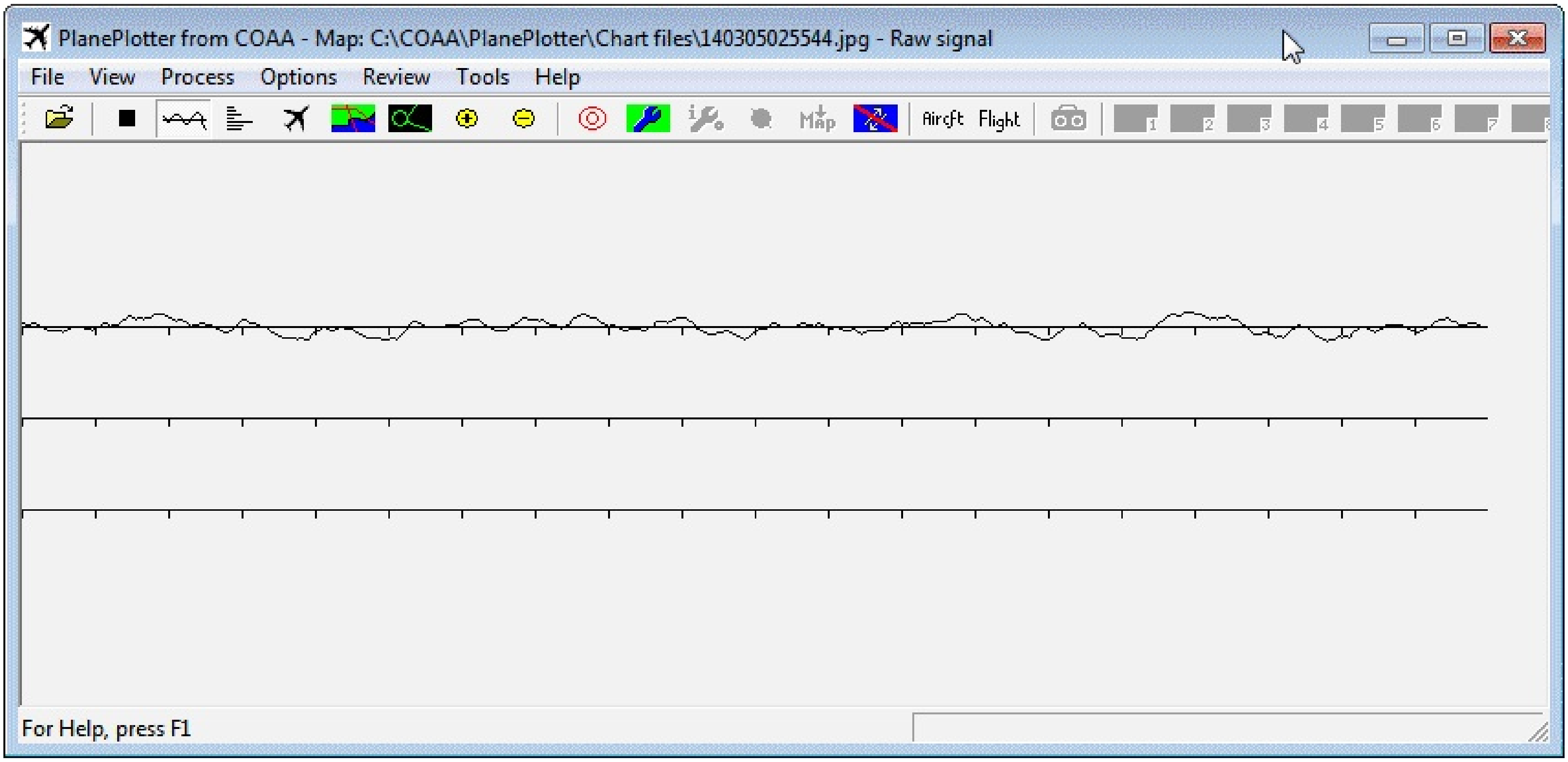
☑ Mode-S — DDE item LiveData

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

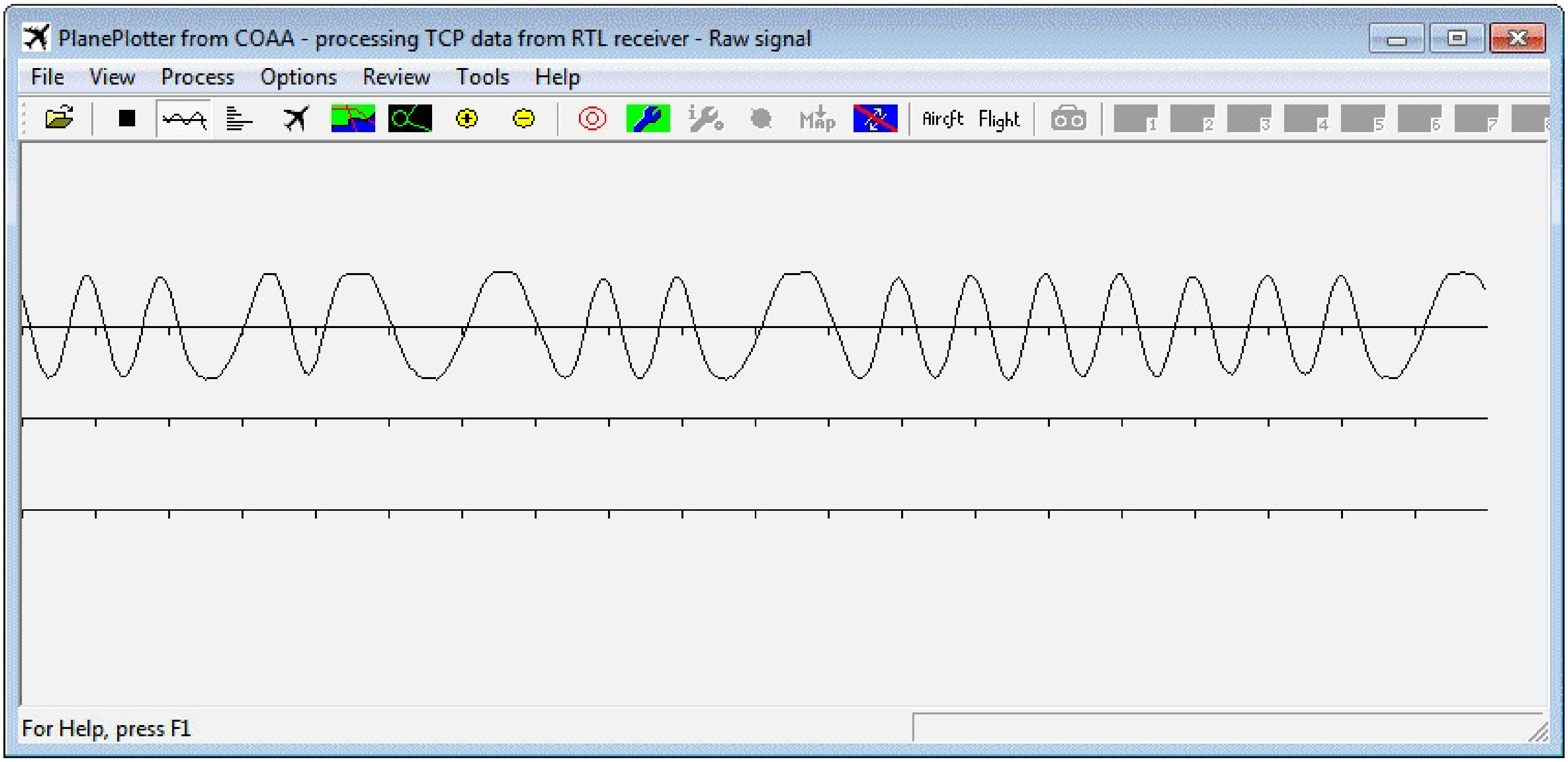
6. Click on the "Start" icon ●.

7. Press the "Signal" icon ∿. This shows a graph of signal 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).

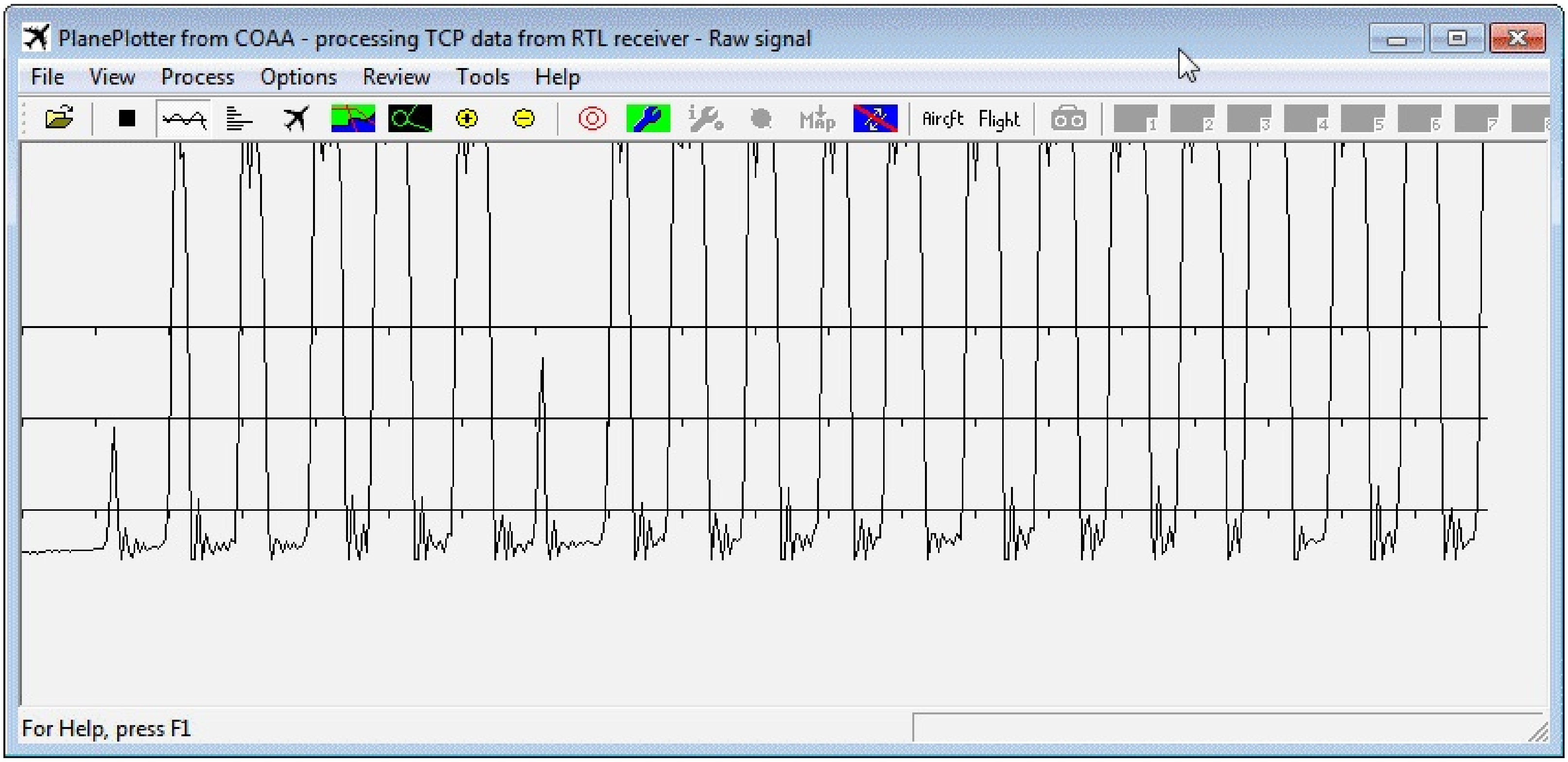
Here is an example of an ideal idle volume.


PlanePlotter from COAA - Map: C:\COAA\PlanePlotter\Chart files\140305025544.jpg - Raw signal
File
View
Process
Options
Review
Tools
Help
Aircft
Flight
For Help, press F1

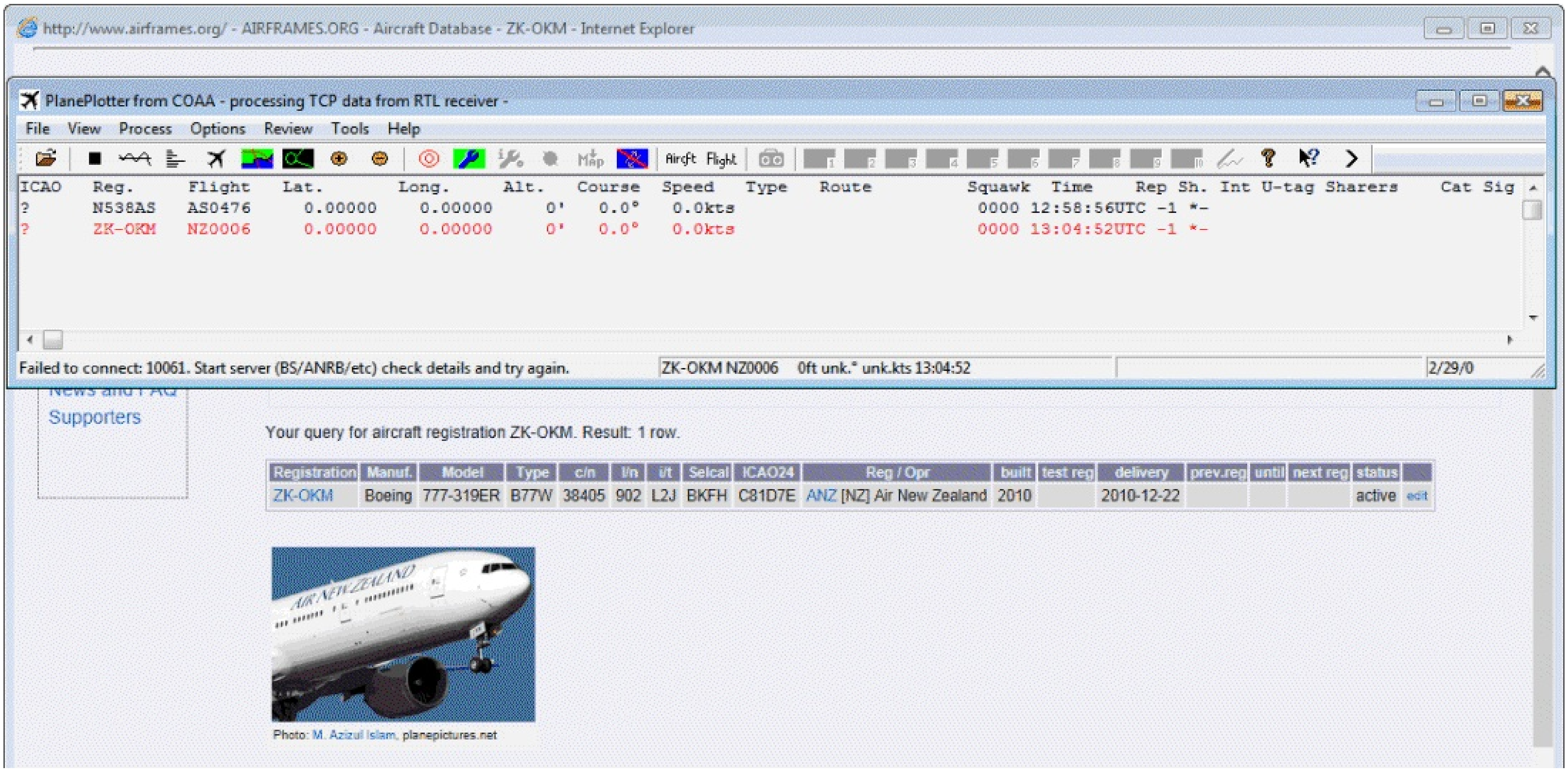
Here is an example of ideal volume when an ACARS packet is sounding.


PlanePlotter from COAA - processing TCP data from RTL receiver - Raw signal
File
View
Process
Options
Review
Tools
Help
Aircft
Flight
For Help, press F1

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


PlanePlotter from COAA - processing TCP data from RTL receiver - Raw signal
File View Process Options Review Tools Help
Aircft Flight
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
Supporters
Photo: M. Azizul Islam, planepictures.net

```
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

AIRNAV ACARS

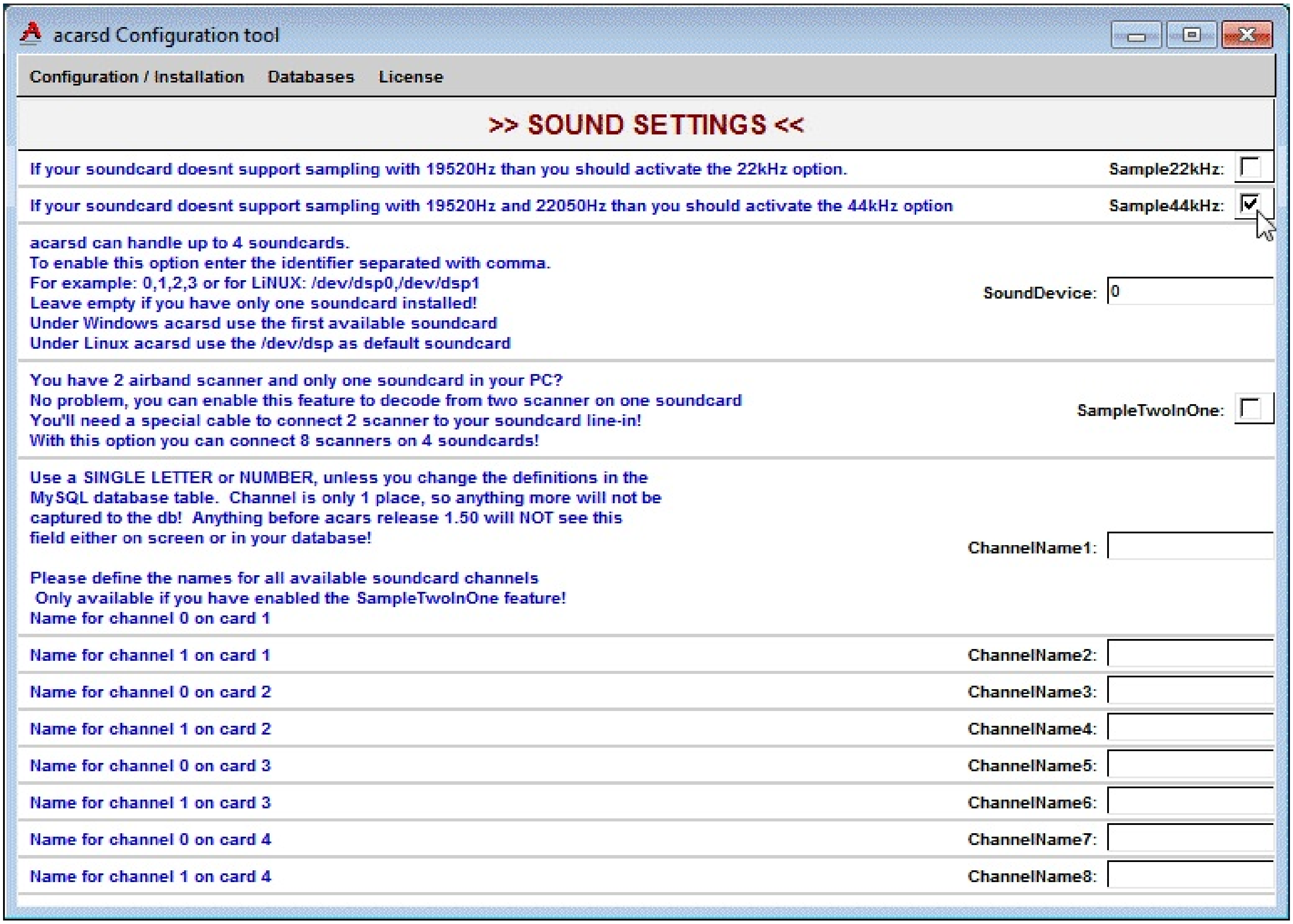
Another commercially available ACARS decoder is the one by AirNav, known as AirNav ACARS decoder 2. This product comes with a 30 day trial but costs $39.95 USD every six months, which makes this the priciest option available. It is available at http://www.airnavsystems.com/ACARS/. The advantage with 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 works 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.



>> 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.

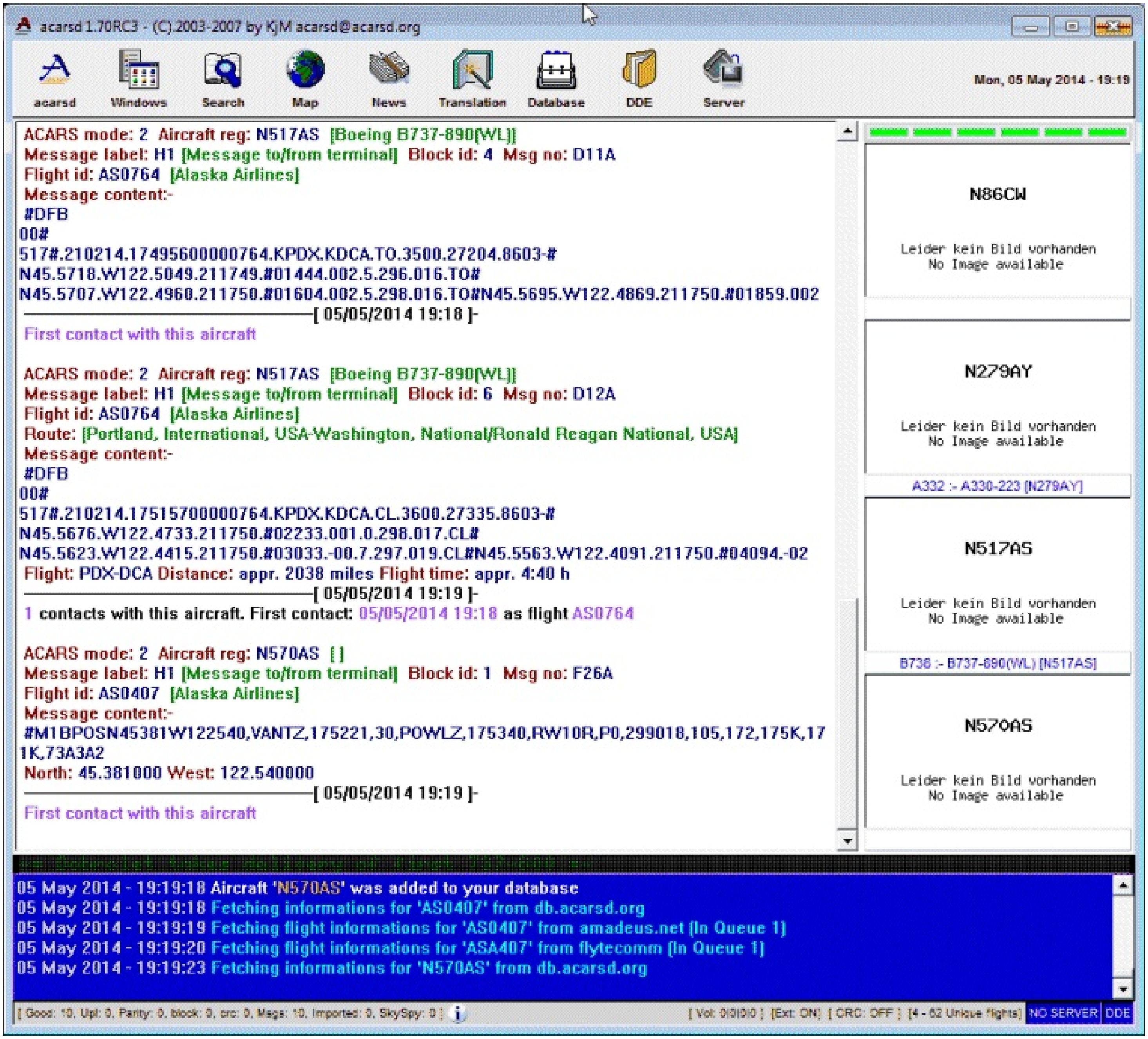
6. In your SDR receiver, set it to output to the audio piping method you have set as your default audio device.

7. Run acarsd.exe in your acarsd folder to open it.

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 for when there is no signal. 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.



```
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
```

```
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
```

```
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]  NO SERVER  DDE
```

LINUX 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 more than one ACARS channel. They should both have similar performance as they are based on the same code base. However, I was personally unable to get these Linux ACARS program to decode as well as the Windows programs.

To run acarsdec, download and extract tar package from http://sourceforge.net/projects/acarsdec/.

Install the following dependencies as well as the librtlsdr libraries if you haven't already. See the Installing RTL-SDR Drivers on Linux section.

```
apt-get install libsndfile1-dev
apt-get install libasound2
```

Then run `make` in the acarsdec folder.

To run rtl_acars_ng simply download the zip file from https://github.com/gat3way/rtl_acars_ng, extract it then run `make`. To get help run `./rtl_acars_ng -h`. To start decoding type

```
./new_rtl_acars -f 131550000
```

ADS-B RECEIVING GUIDE

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 devised. ADS-B reads an onboard GPS and then broadcasts the data via radio giving more accurate and stable 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. 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 your own ADS-B signals.

ADS-B is based on something called Mode S which provides 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 are also receivable by some decoders.

ADS-B signals can be found at a frequency of 1090 MHz or less commonly in some places at 978 MHz. It has been discovered by the RTL-SDR community, that the R820T 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 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. Aside from this you will need

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

ADS-B ANTENNAS

With a good antenna the RTL-SDR could be capable of receiving aircraft from over 250nm away, that is about 460km 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. This will depend on how far away you are from the aircraft and what your local RF interference is like. However, if you want to increase your ADS-B radars 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 transmitter. Since ADS-B signals are generally quite strong and come from aircraft up high, 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.

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 installations, such as RG-6 or RG-8, as these cables are designed for gigahertz frequencies. See the Coaxial Cable Guide section for more information.

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

A low noise amplifier (LNA) placed at the antenna such as the LNA4ALL 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 Raspberry Pi running RTL-SDR at the antenna and then used WiFi or an Ethernet cable to send decoded data to a PC.

For ADS-B we recommend either a ¼ Wave Ground Plane, J-Pole, Dipole or Collinear antenna. See the Antenna Guide for more information about these antennas.

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 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. Minicircuits also sells an 1090 MHz surface mount bandpass filter at http://www.minicircuits.com/MCLStore/ModelInfoDisplay?13979749855620.039050499120180104. 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.

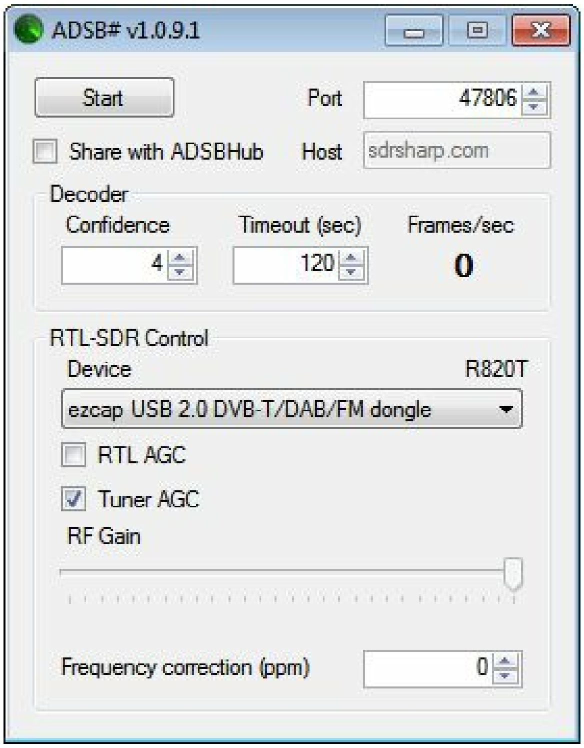
SOFTWARE GUIDE

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 and dump1090.

ADSB# (WINDOWS)

ADSB# is a Windows based 1090 MHz ADS-B decoder developed by the programmer of SDR Sharp. It has a simple graphic user interface and is probably the easiest decoder to use overall. To use ADSB#, download it from http://sdrsharp.com/downloads/adsbsharp.zip.



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 install and run ADSB#, all you need to do is extract the zip file into a folder, and then 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 first time testing, but manual gain settings work best after tweaking. Maximum gain with no AGC 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.

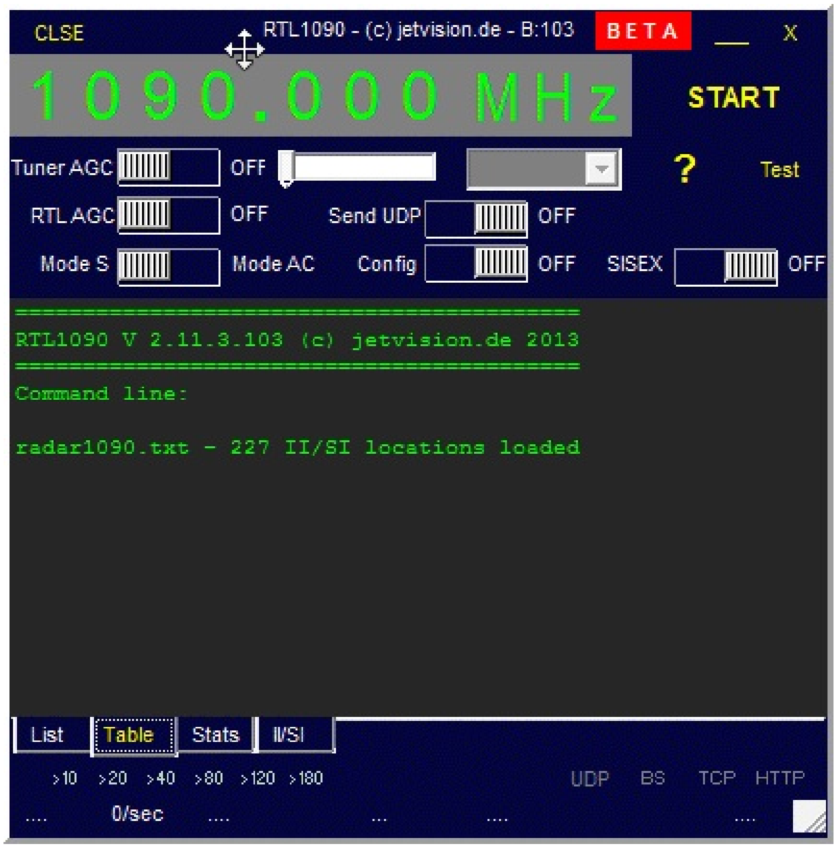
ADSB# decodes the ADS-B data and then sends location data via TCP/IP. Another program such as Virtual Radar Server, ADSBScope or PlanePlotter (explained later) is needed to visualize the data. If you have several antennas in different locations, you can combine your data in this way using the adsbhub.exe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

The frequency correction (ppm) setting is the same as the setting in SDRSharp 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.

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 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. The timeout setting defines how long the decoder will wait before checking the confidence of the aircraft again.

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 guides you through installing the drivers through zadig. Note if you’ve already installed the drivers through zadig via the Quickstart guide, you can ignore the zadig installation shown in the IMU.



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

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

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 one that has the location information. RTL1090 also has a built in spectrum viewer which can be activated by flipping the SISEX switch. Finally, RTL1090 also has a 1-bit checksum error checker built in. The error checker can be activated in the Config menu. It is disabled by default because it can be taxing on older slower PCs. You can access the options screen by clicking the OPEN button in the top left corner of the program GUI. Most people report that RTL1090 has better receive performance compared to ADSB#.

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 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. Very useful for Linux based 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 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, simply download the zip file from the git repository and unzip the files into a folder or do a git clone on https://github.com/MalcolmRobb/dump1090. Then assuming you have the librtlsdr libraries preinstalled, run ‘make’ at a terminal in that folder. This 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 Google Maps web server that can display received aircraft. In a web browser go to http://localhost:8080 to see the map.

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 is mainly useful for low traffic areas as it may cause many false positives under high traffic conditions. 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.

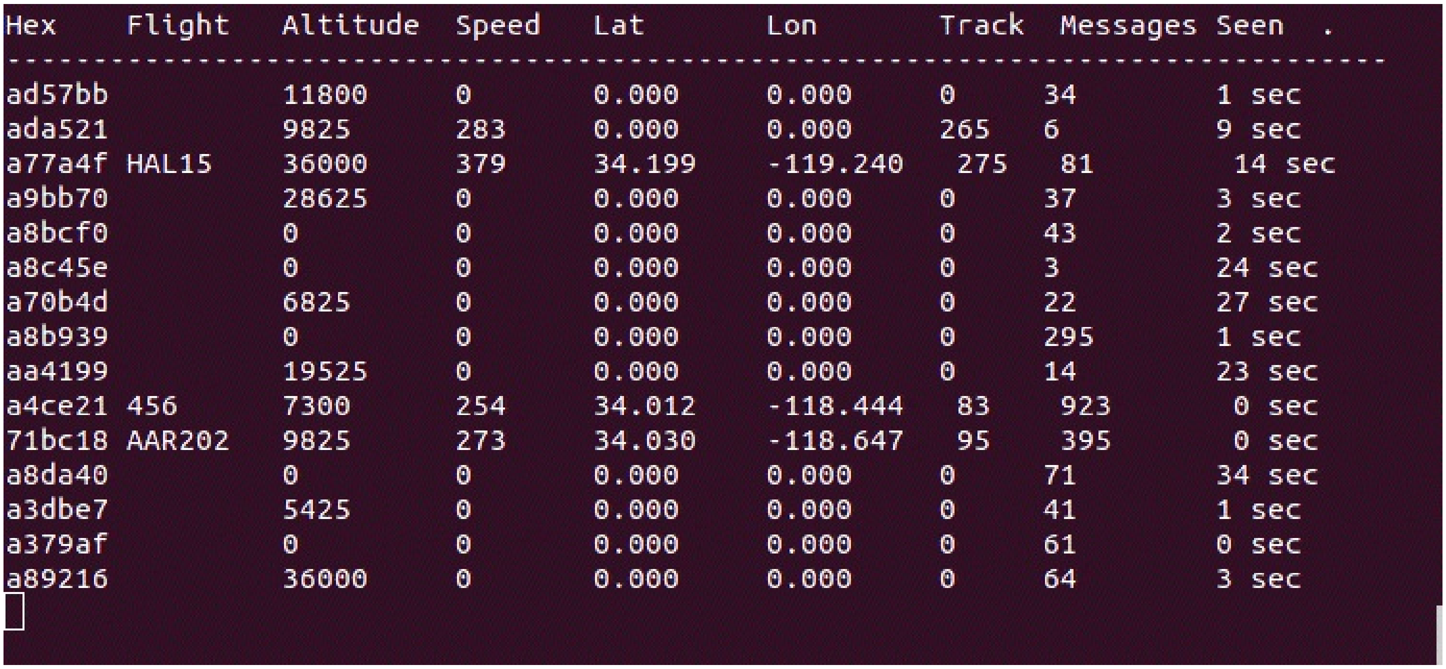
Dump1090 can also receive Mode AC messages together with Mode S by using the flag --modeac

To install dump1090 on Windows follow these steps:

1. Download the dump1090 zip file from the GitHub download link.
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
```

MODESDECO

ModesDeco is a Windows/Linux/OSX/Rpi command line based decoder. It can receive Mode S and Mode A/C data simultaneously. 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 modesdeco, open a command line prompt from windows, navigate to the directory you’ve extracted modesdeco to and type the following.

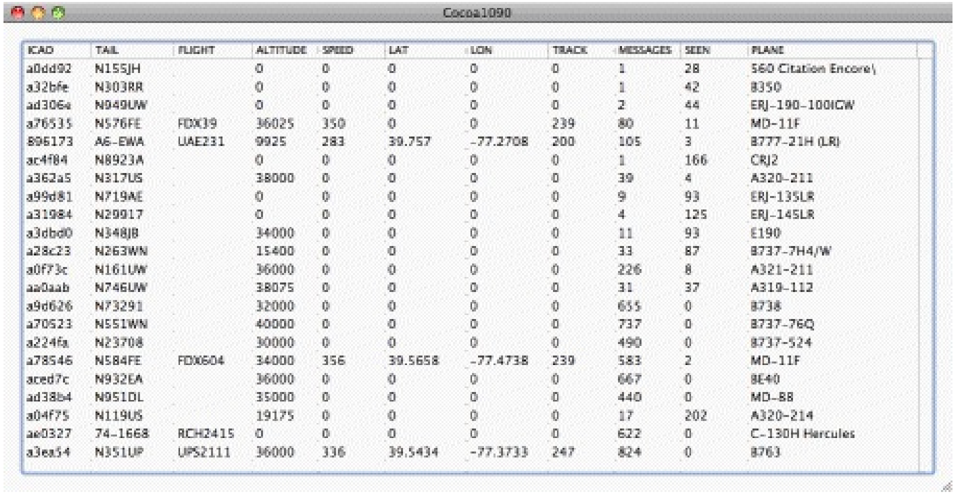
```
modesdeco2.exe --gain 49.6 --net 10001 --sbs10001 --location lat:lon
```

This mode outputs in the base station format. To output in other formats such as AVR usable by other graphical radar displays use

```
./modesdeco2 --gain 49.6 --freq-correction 68 --sbs10001 10001 --avr 44044 --msg 30003  --location 43.25:39.33 --filter-nocountry --rbs --verbose
```

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.



ICAO	TAIL	FLIGHT	ALTITUDE	SPEED	LAT	LON	TRACK	MESSAGES	SEEN	PLANE
a0dd92	N155JH		0	0	0	0	0	1	28	560 Citation Encore\
a32bfe	N303RR		0	0	0	0	0	1	42	B350
ad306e	N949UW		0	0	0	0	0	2	44	ERJ-190-100IGW
a76535	N576FE	FDX39	36025	350	0	0	239	80	11	MD-11F
896173	A6-EWA	UAE231	9925	283	39.757	-77.2708	200	105	3	B777-21H (LR)
ac4f84	N8923A		0	0	0	0	0	1	166	CRJ2
a362a5	N317US		38000	0	0	0	0	39	4	A320-211
a99d81	N719AE		0	0	0	0	0	9	93	ERJ-135LR
a31984	N29917		0	0	0	0	0	4	125	ERJ-145LR
a3dbd0	N348JB		34000	0	0	0	0	11	93	E190
a28c23	N263WN		15400	0	0	0	0	33	87	B737-7H4/W
a0f73c	N161UW		36000	0	0	0	0	226	8	A321-211
aa0aab	N746UW		38075	0	0	0	0	31	37	A319-112
a9d626	N73291		32000	0	0	0	0	655	0	B738
a70523	N551WN		40000	0	0	0	0	737	0	B737-76Q
a224fa	N23708		30000	0	0	0	0	490	0	B737-524
a78546	N584FE	FDX604	34000	356	39.5658	-77.4738	239	583	2	MD-11F
aced7c	N932EA		36000	0	0	0	0	667	0	BE40
ad38b4	N951DL		35000	0	0	0	0	440	0	MD-88
a04f75	N119US		19175	0	0	0	0	17	202	A320-214
ae0327	74-1668	RCH2415	0	0	0	0	0	622	0	C-130H Hercules
a3ea54	N351UP	UPS2111	36000	336	39.5434	-77.3733	247	824	0	B763

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://www.cgran.org/wiki/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.

GRAPHICAL RADAR DISPLAY SOFTWARE FOR WINDOWS

The Windows decoders mentioned above all 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.

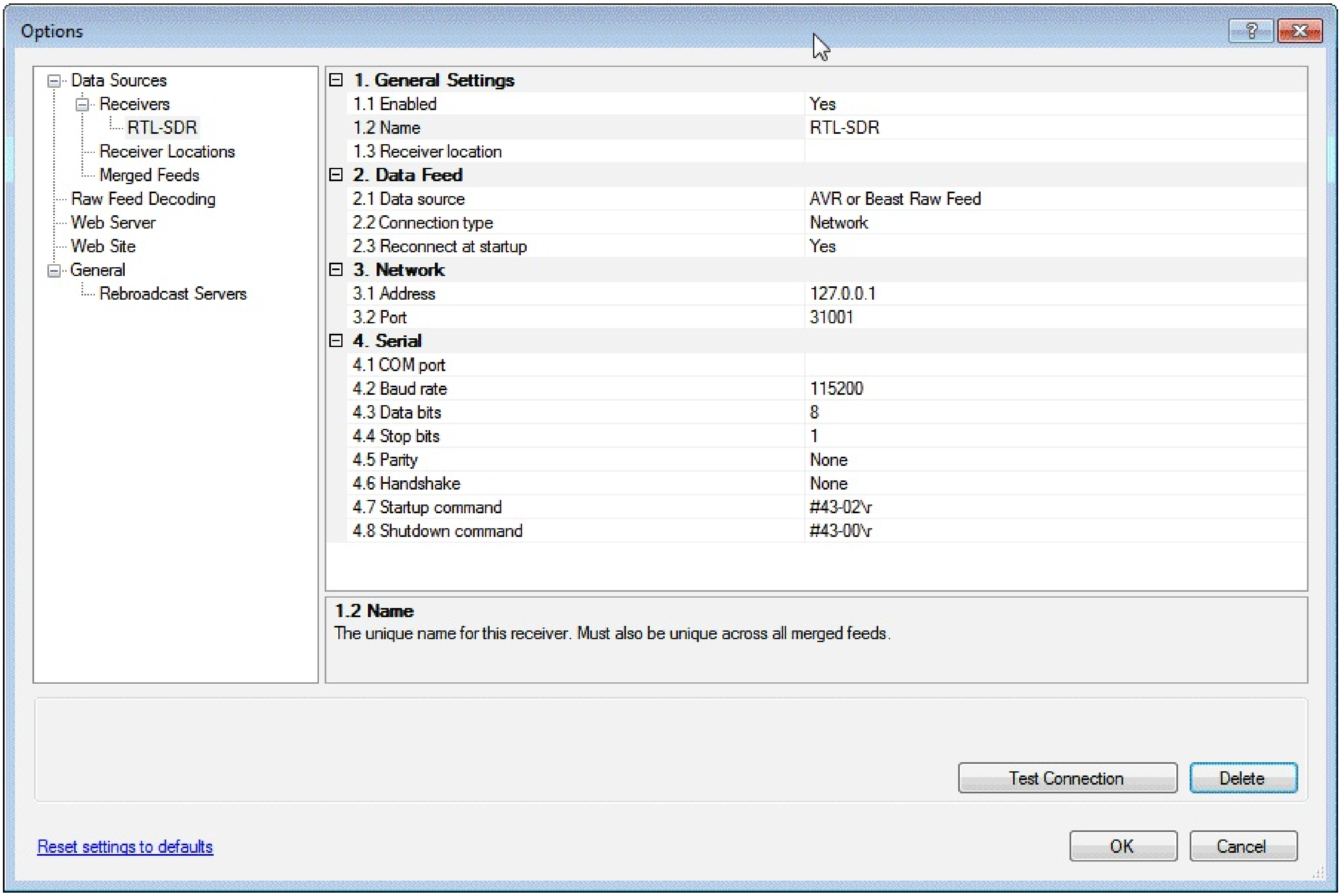
VIRTUAL RADAR SERVER (WINDOWS)

Virtual Radar Server (VRS) is Windows and Linux Mono compatible based free software which displays plane positions in a browser window 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 in 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

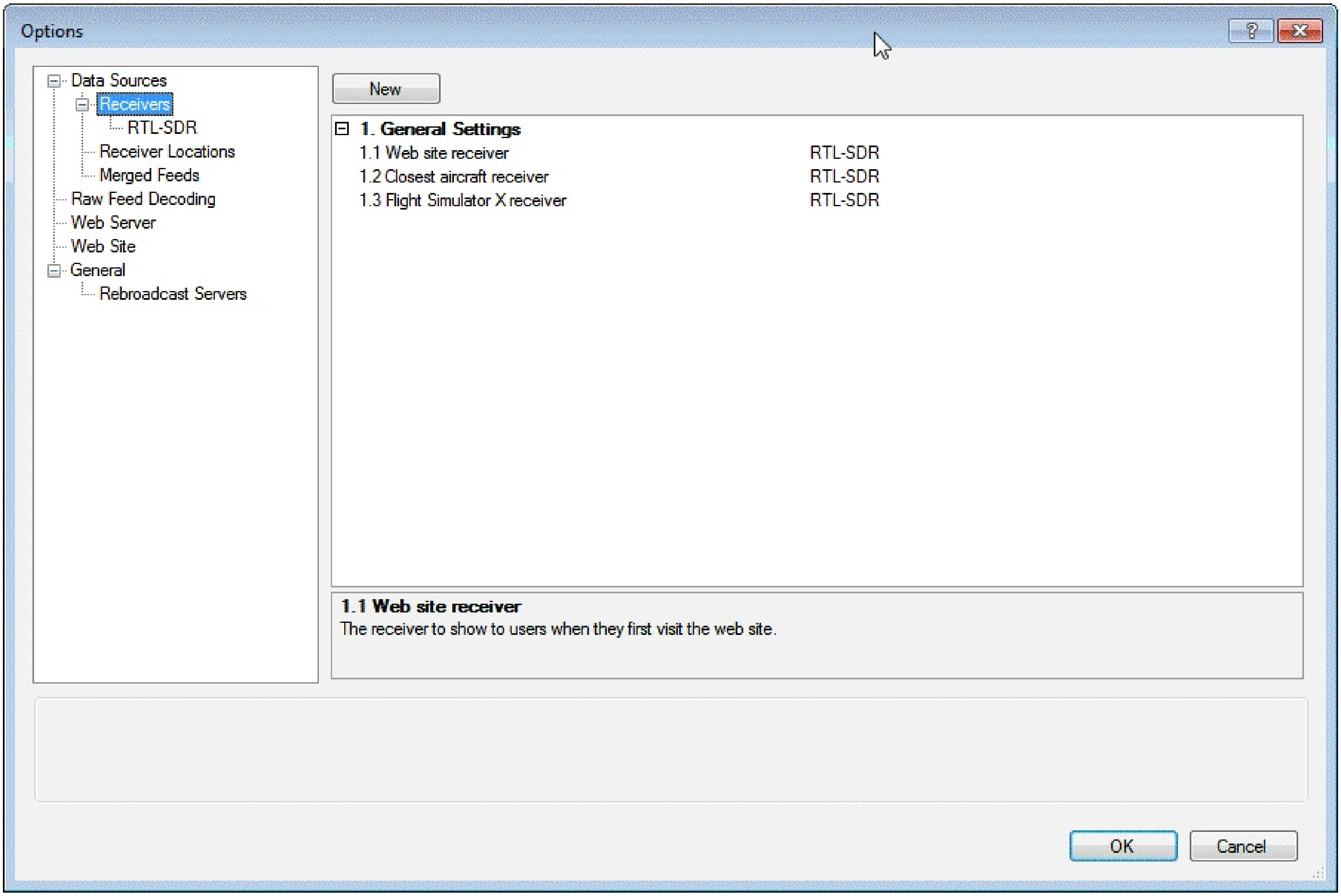
number to 47806 if using ADSB#, or 31001 if using RTL1090. (Or set the port to a custom value in ADSB#). The following picture shows the settings you should be using with port 31001 used.



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 created. Press OK.



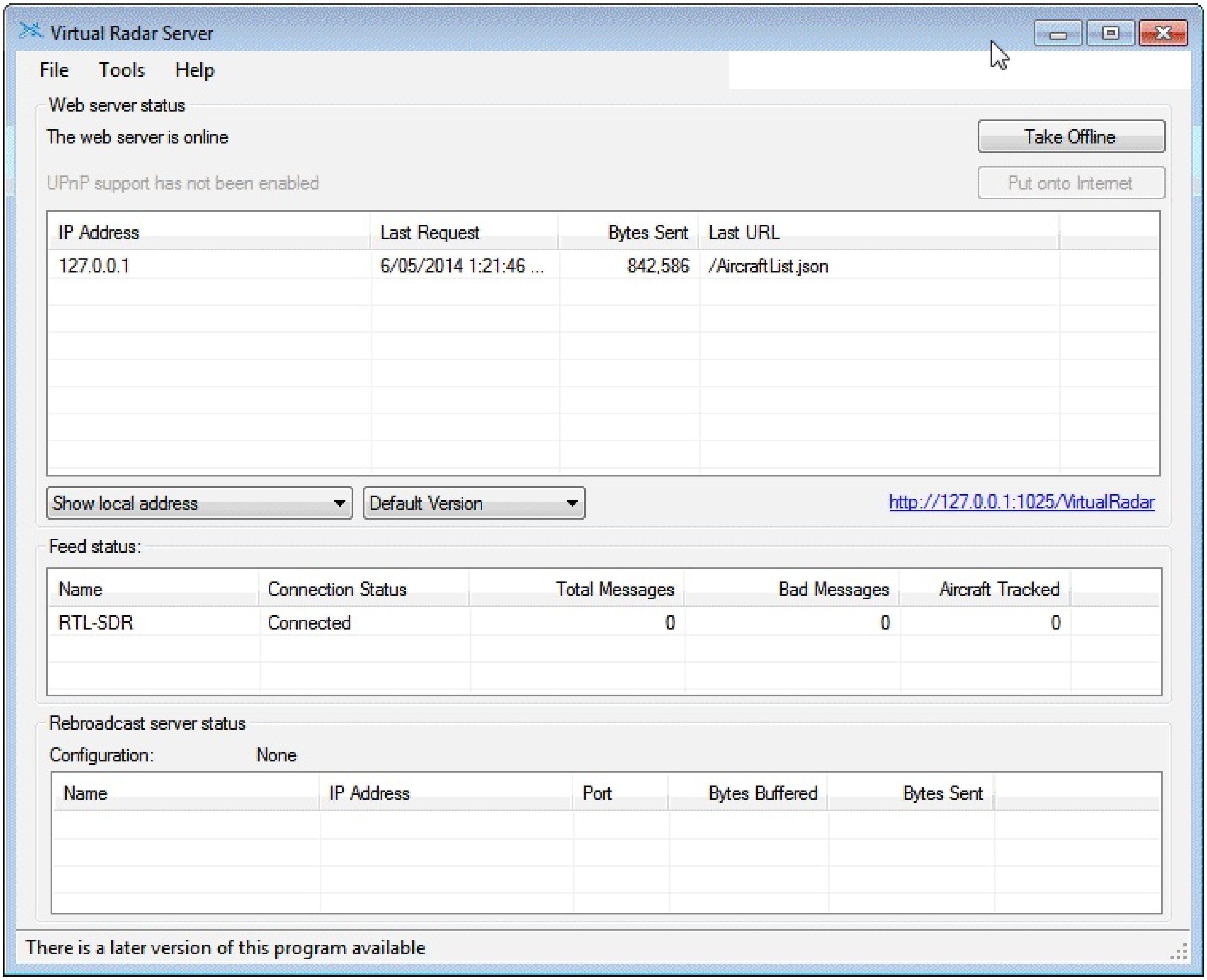
1. General Settings

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 of the VRS main window to be taken to a web browser with Google maps and VRS.



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

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, you should see planes appear on the map if you are getting 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. (An alternative basestation.sqb can be found here http://pp-sqb.mantma.co.uk/)

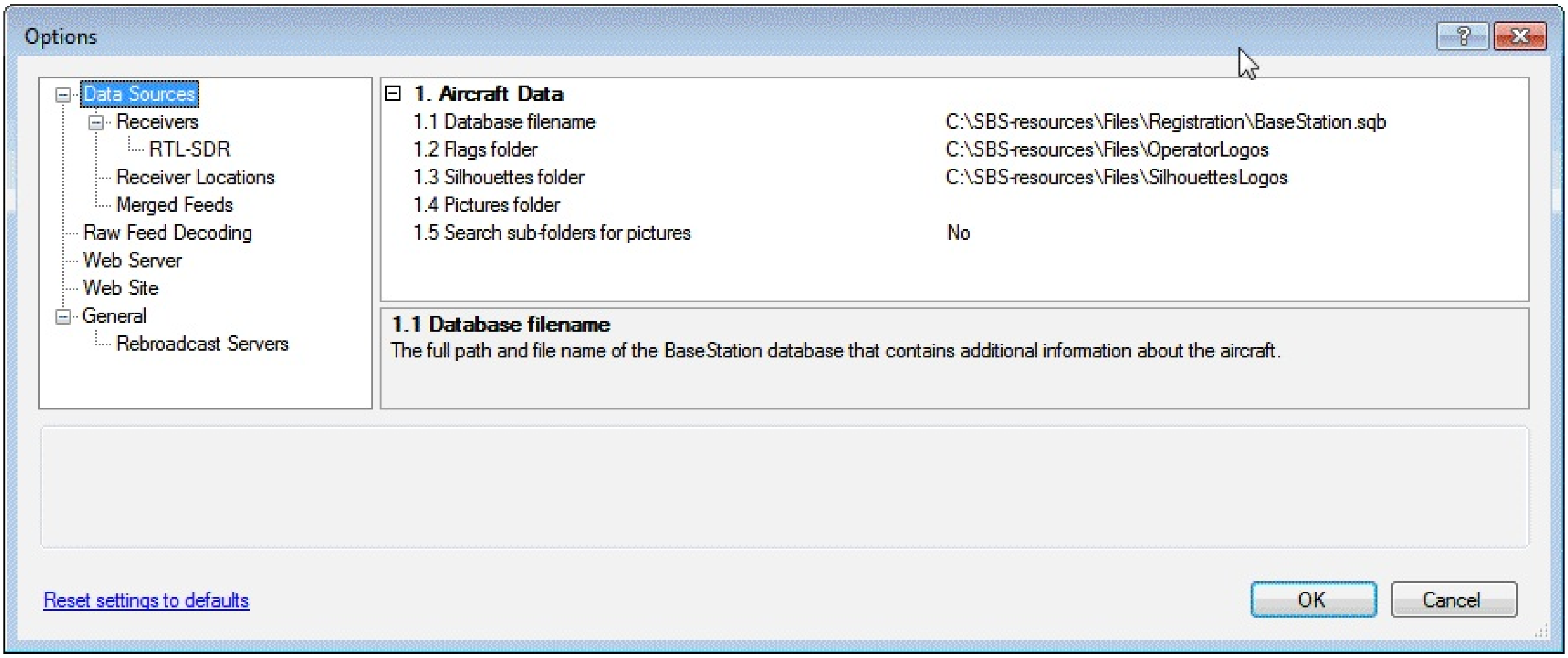
2. Now in Virtual Radar Server go to **Tools->Options** and click on the Data Sources tab.

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

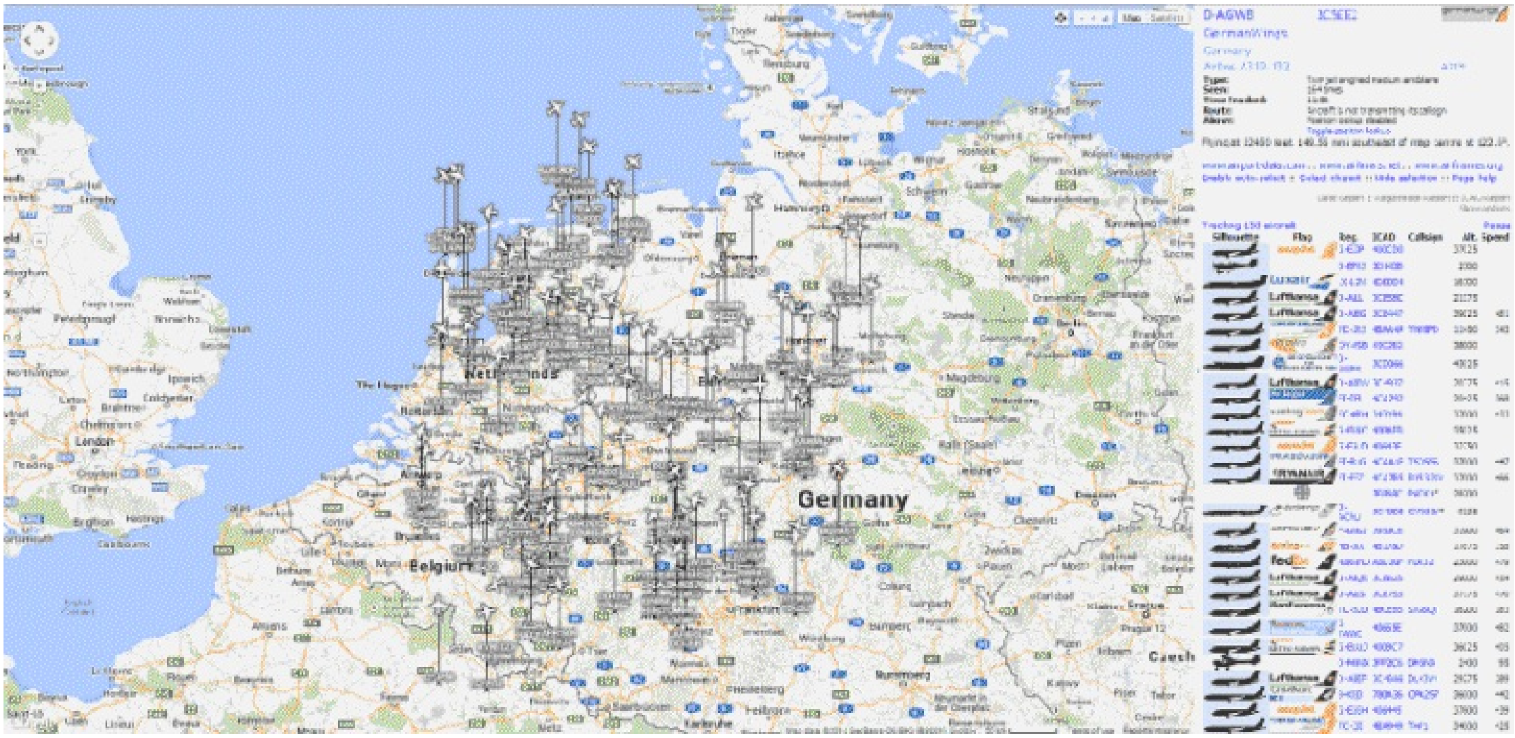
6. Click OK. The database is now set up.



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.



Virtual Radar Server also has various plugins available on the website.

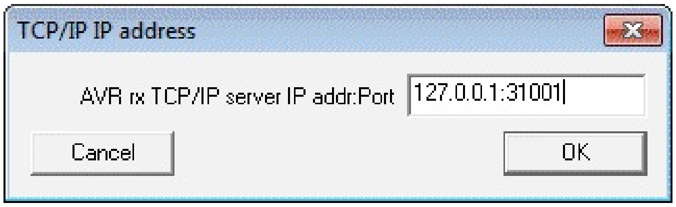
PLANEPLOTTER

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.

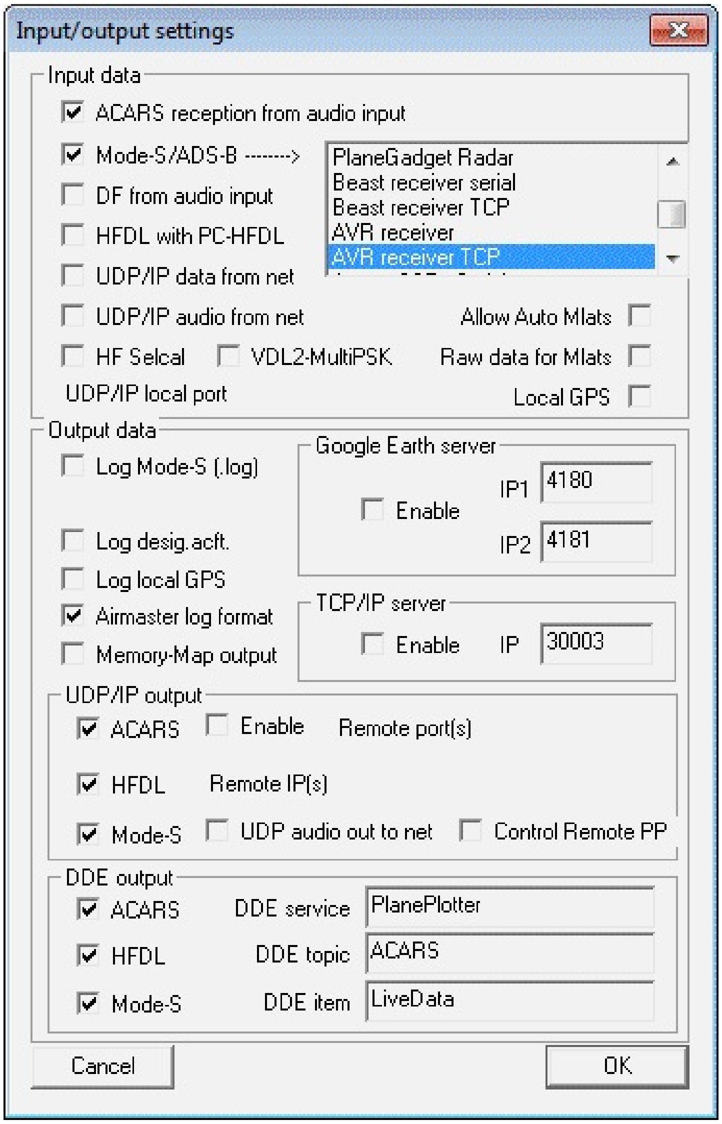
1. Open PlanePlotter
2. Go to **Options->Mode-S receiver -> AVR Receiver -> TCP/IP Client** . Set the address and port to 127.0.0.1:31001.



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 -------> AVR receiver TCP
☐ DF from audio input
☐ HFDL with PC-HFDL
☐ UDP/IP data from net
☐ UDP/IP audio from net
☐ HF Selcal ☐ VDL2-MultiPSK
UDP/IP local port
Allow Auto Mlats ☐
Raw data for Mlats ☐
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

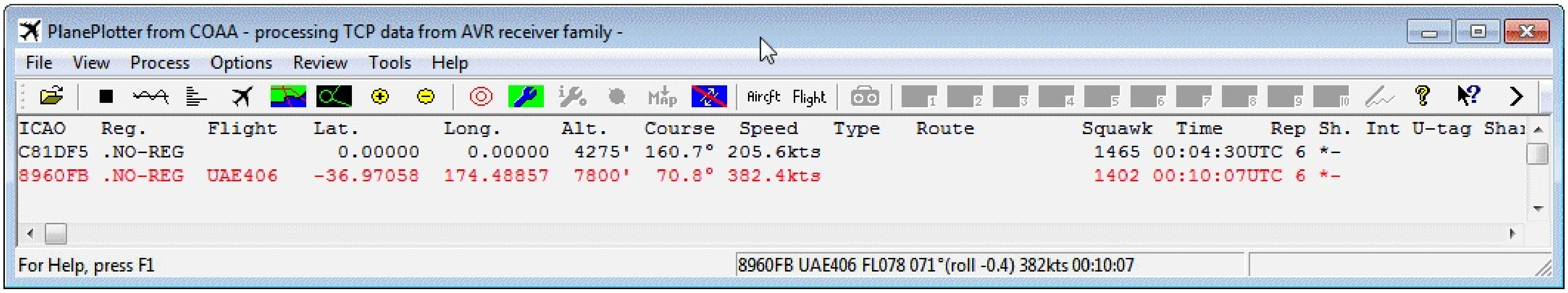
UDP/IP output
☑ ACARS ☐ Enable Remote port(s)
☑ HFDL Remote IP(s)
☑ Mode-S ☐ UDP audio out to net ☐ Control Remote PP

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 RTL1090 which has the default TCP/IP port set at 31001. Turn on ADSB# or RTL1090.

5. Click the green start button ●.

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

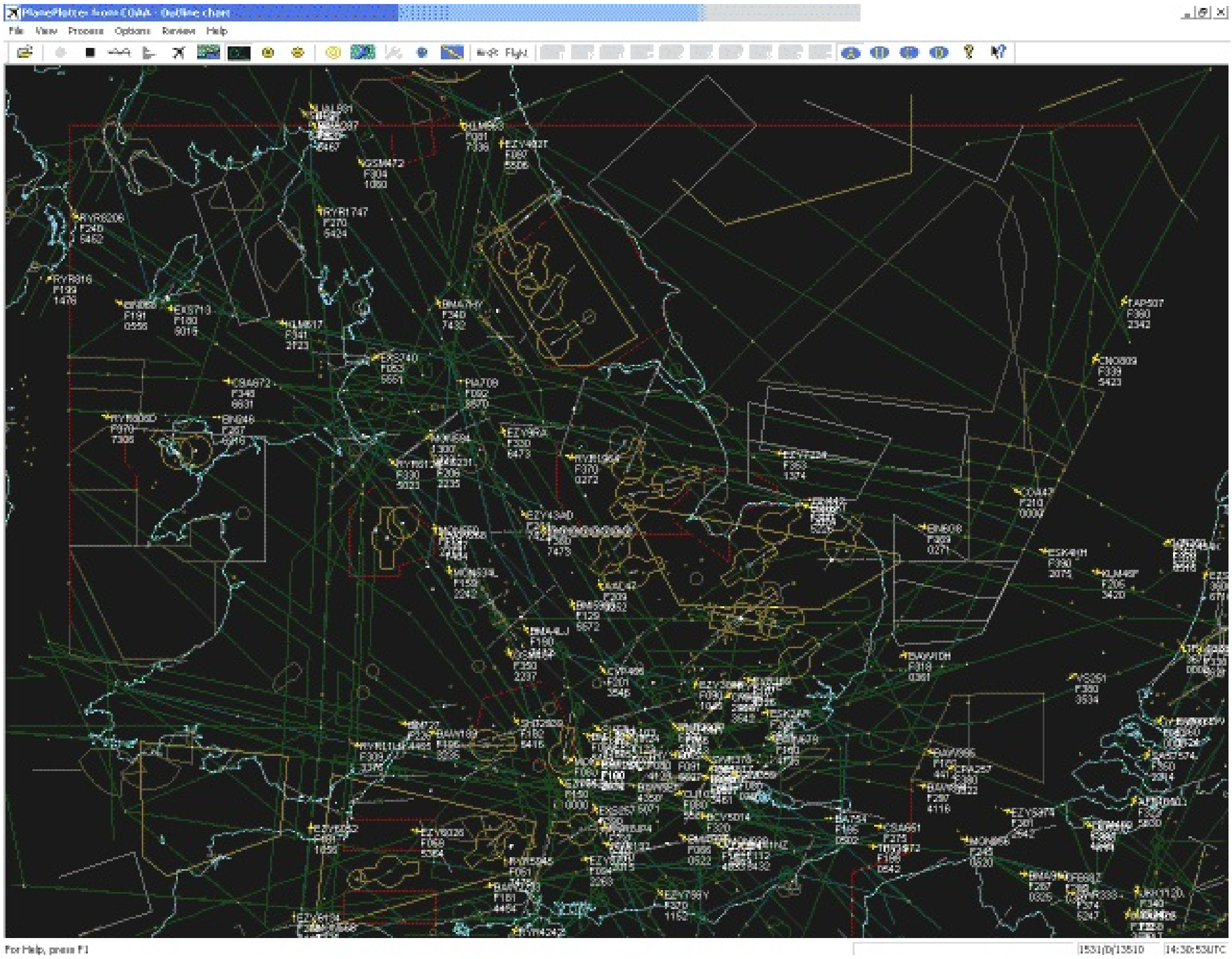


PlanePlotter from COAA - processing TCP data from AVR receiver family -
File View Process Options Review Tools Help
Airft Flight

```
ICAO    Reg.     Flight    Lat.       Long.      Alt.   Course  Speed     Type   Route          Squawk  Time          Rep Sh.  Int  U-tag  Sha
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 Maps 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
Flght

For Help, press F1 | 1531/0/13510 | 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.

1. Go to **Options->I/O Settings** and click the Enable checkbox under the Google Earth Server, and press OK. Note to access I/O settings you'll need to stop PlanePlotter decoding first 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 installed PlanePlotter to. The default installation path is C:\Program Files (x86)\COAA\PlanePlotter. In this folder locate the google_aircraft.kml, 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.

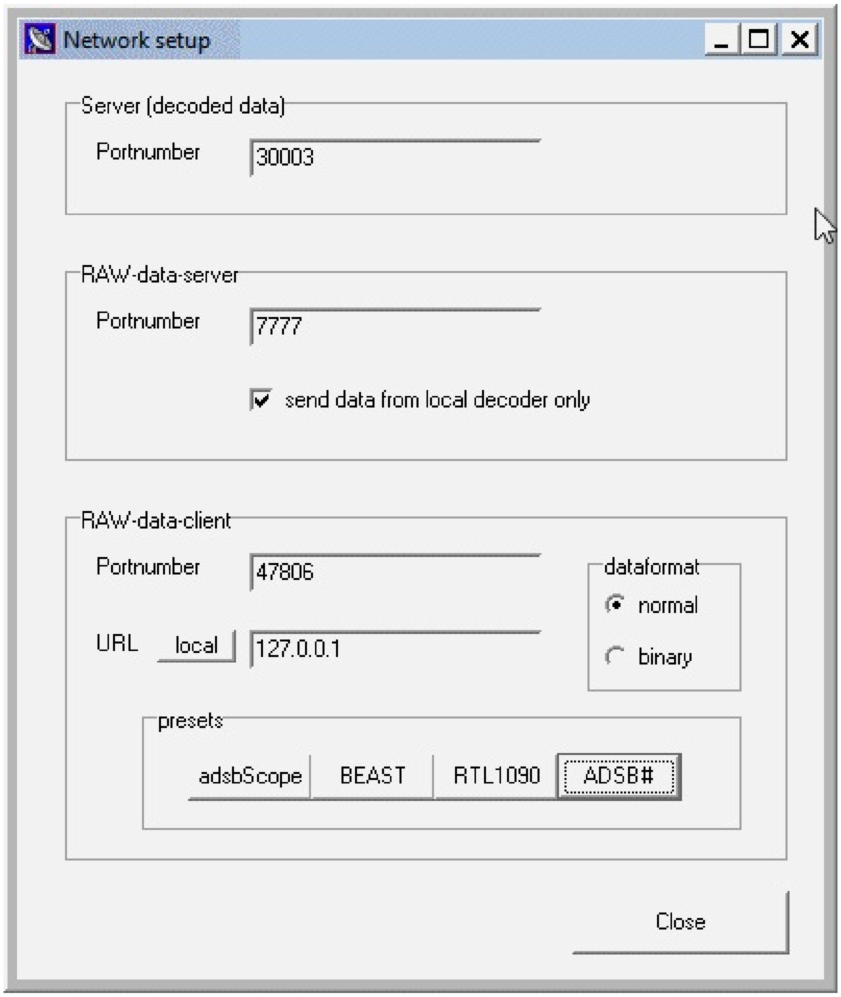
ADSBSCOPE

adsbSCOPE is a free ADS-B radar program. adsbSCOPE 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 options 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 following 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.



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

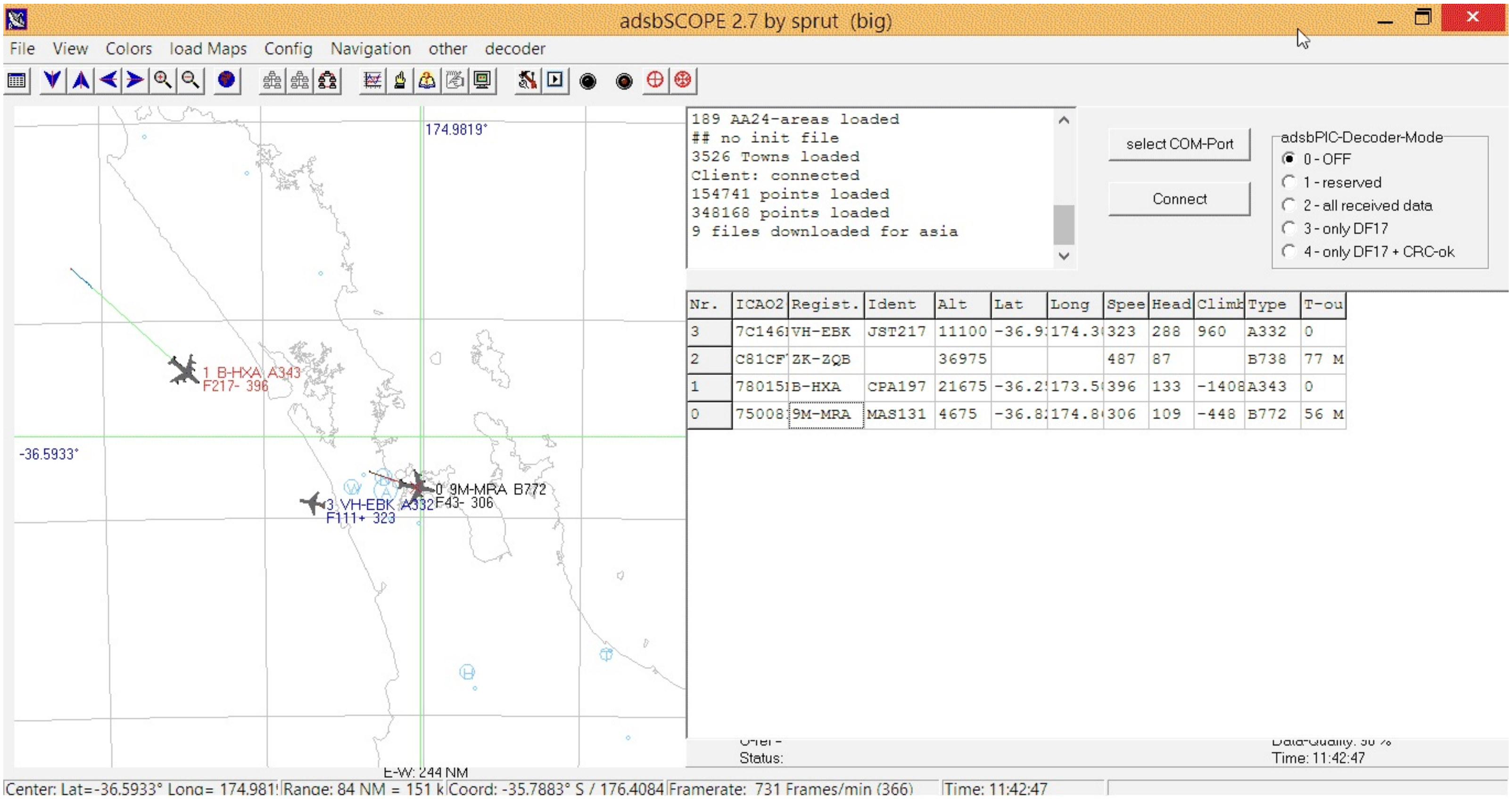
4. Now turn on RTL1090 or ADSB#.

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**.



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

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

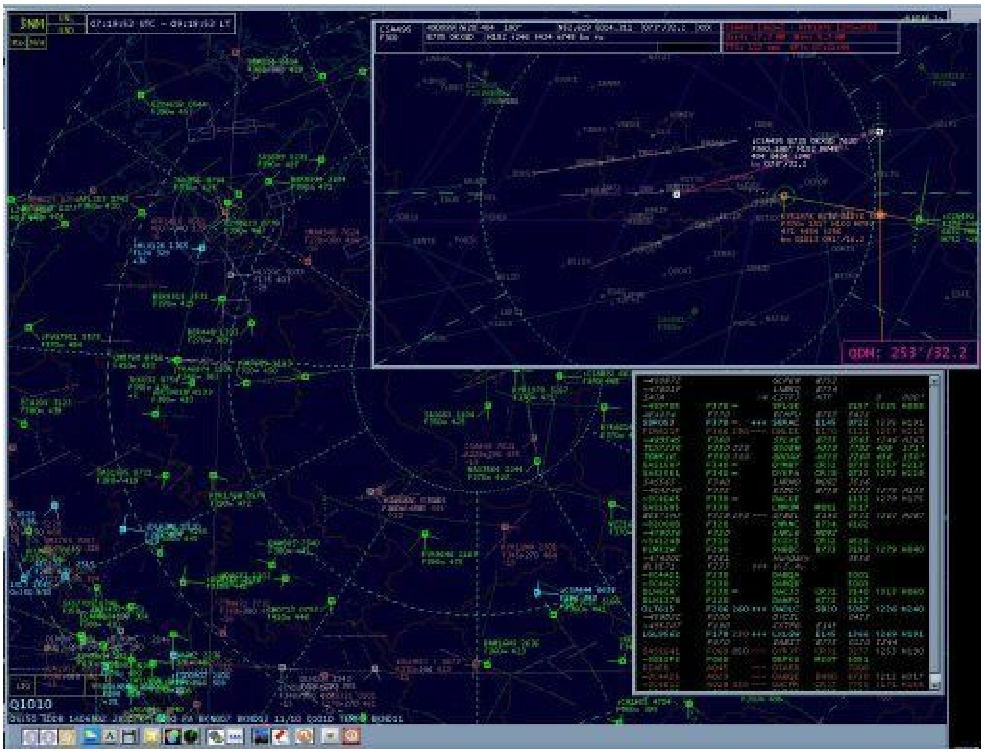
ADSBScope can also create a max range plot which can over a period of time show how well your antenna is working. This will draw a line to each of the furthest aircraft seen over time. To turn it on simply follow these steps.

1. Go to **Config->Maximum Range** and enable ‘show all altitudes’.

2. Ensure that **View->Maximum Range** is checked.

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=10424

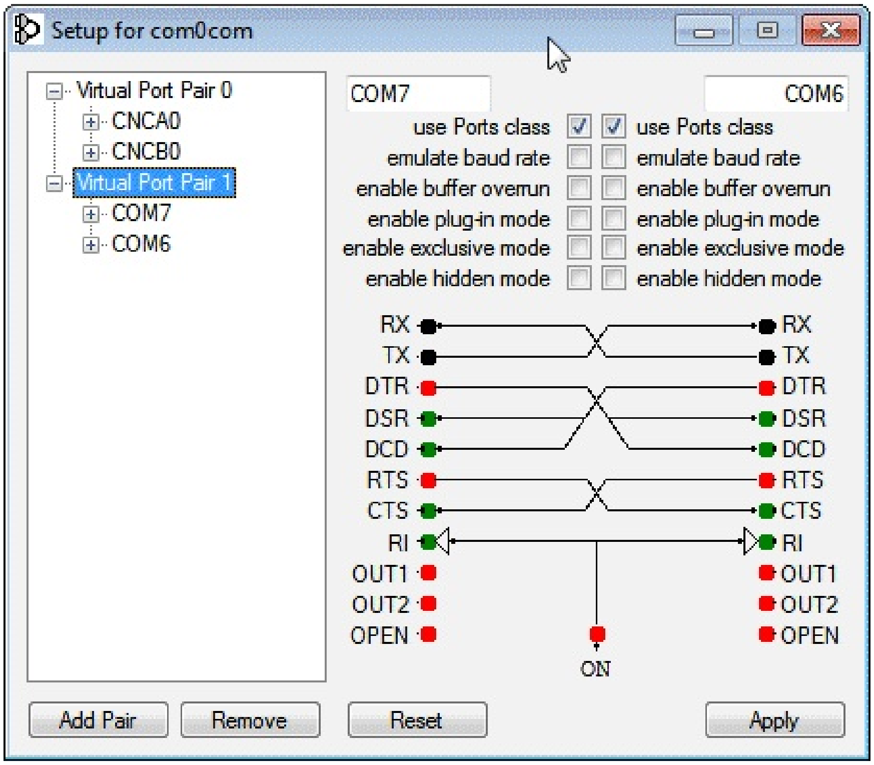


BASESTATION

BaseStation is one of the original ADS-B display software packages. It was around when there were only dedicated ADS-B hardware receivers available. 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.

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`.

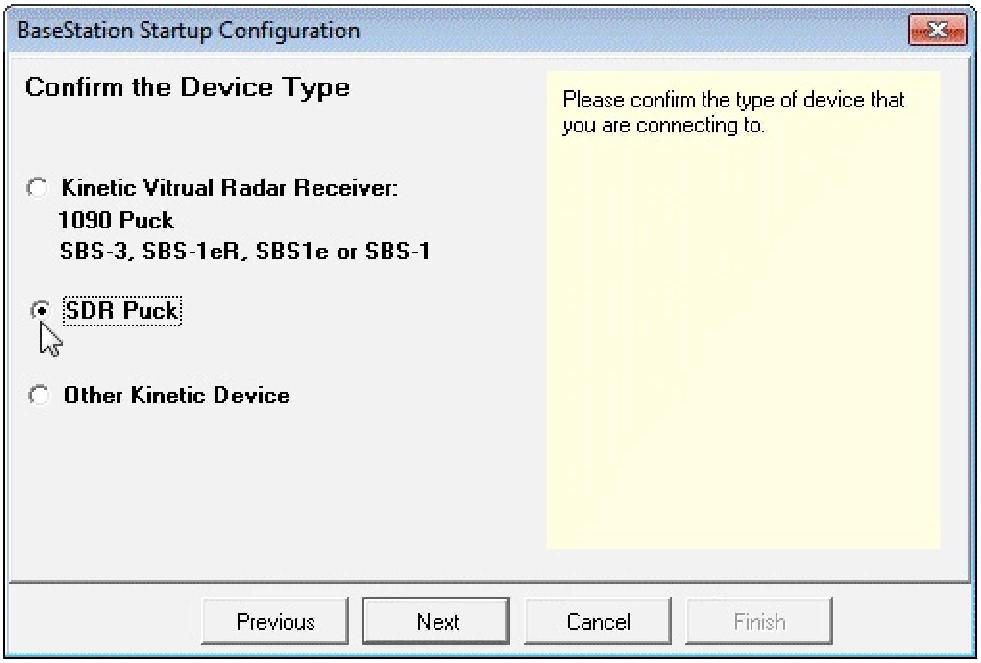
```
com2tcp --baud 3000000 --parity o \\.\COM6 127.0.0.1 31001
```

```
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.



Confirm the Device Type

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

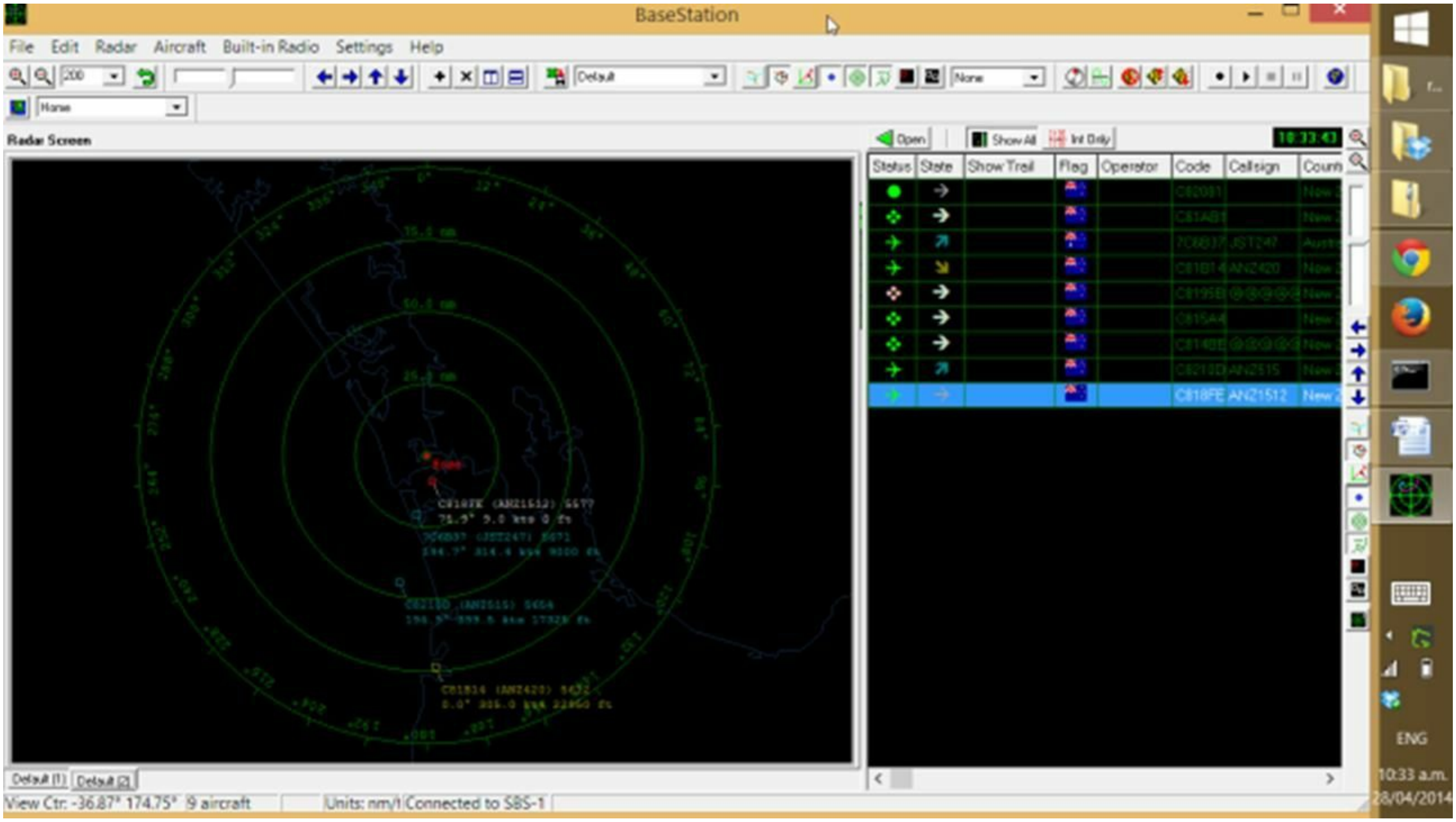
○ Kinetic Vitrual Radar Receiver:
1090 Puck
SBS-3, SBS-1eR, SBS1e or SBS-1

● 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.

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



Status	State	Show Trail	Flag	Operator	Code	Callsign	Coun
●	→				C82081		New Z
	→				C81AB1		New Z
	↗				7C6B37	JST247	Austr
	↘				C81B14	ANZ420	New Z
	→				C8195B	@@@@@	New Z
	→				C815A4		New Z
	→				C814BE	@@@@@	New Z
	↗				C8210D	ANZ515	New Z
	→				C818FE	ANZ1512	New Z

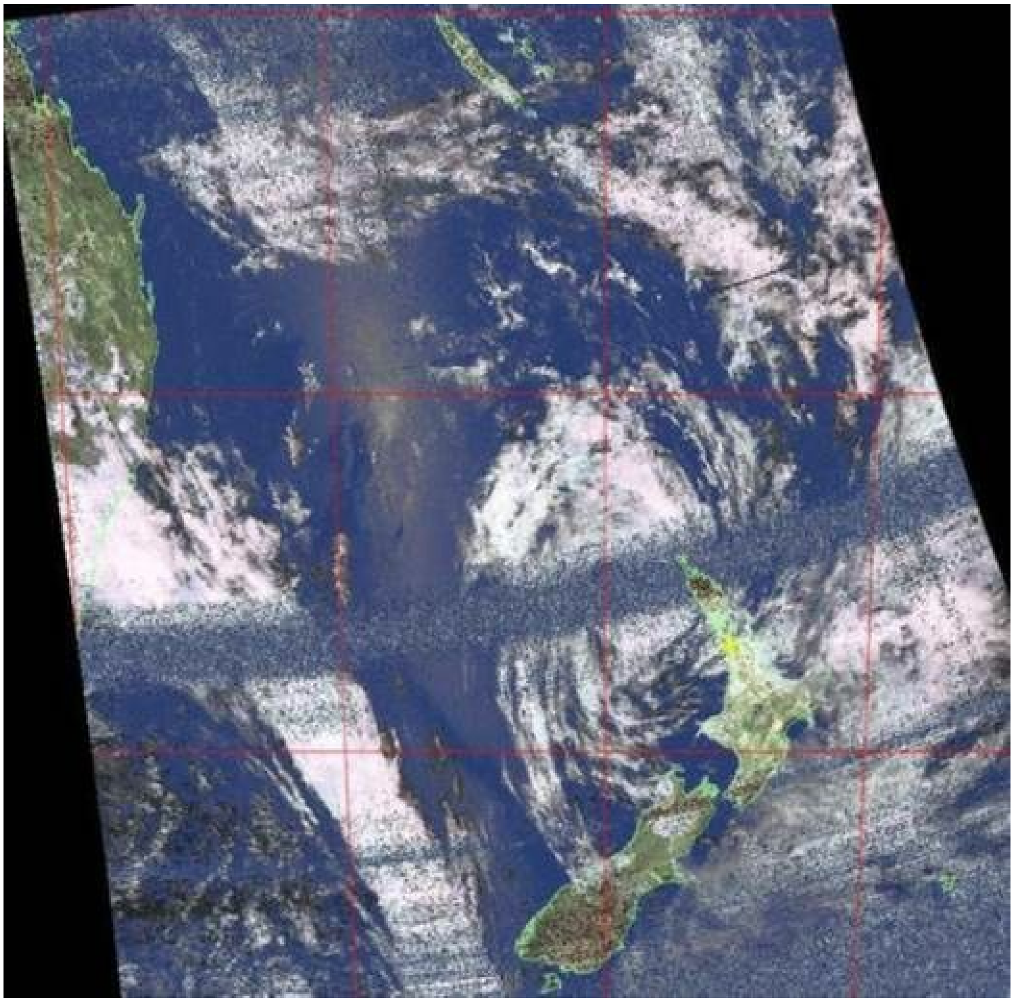
View Ctr: -36.87° 174.75° 9 aircraft | Units: nm/ft Connected to SBS-1

FEEDING DATA TO FLIGHTRADAR24.COM

flightradar24.com is a service which collects ADS-B flight information from numerous sources around the world and shares the aggregated data on their website. You can use your RTL-SDR ADS-B radar set up to contribute to this service by either using PlanePlotter, or downloading the software from their website. If you are a contributor, you will be rewarded with access to flightradar24's premium features.

Instructions on setting this up can be found at http://www.flightradar24.com/dvbt-stick.

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 is passes 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 SDRSharp 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 signal is circularly polarized because the NOAA satellite spins as it transmits, causing the signal to 'spin' in a circle as well.

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 colourize the image using an infrared image also obtained from the transmission.

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 SDR's and even some hardware radio scanners can also work with this tutorial, provided the radio 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.
- 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 quadrafilar 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 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.

A Quadrifilar Helix (QFH) is a circularly polarized antenna that can be constructed out of PVC pipe and coax cable. 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.

See the antenna guide for more information about these 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.

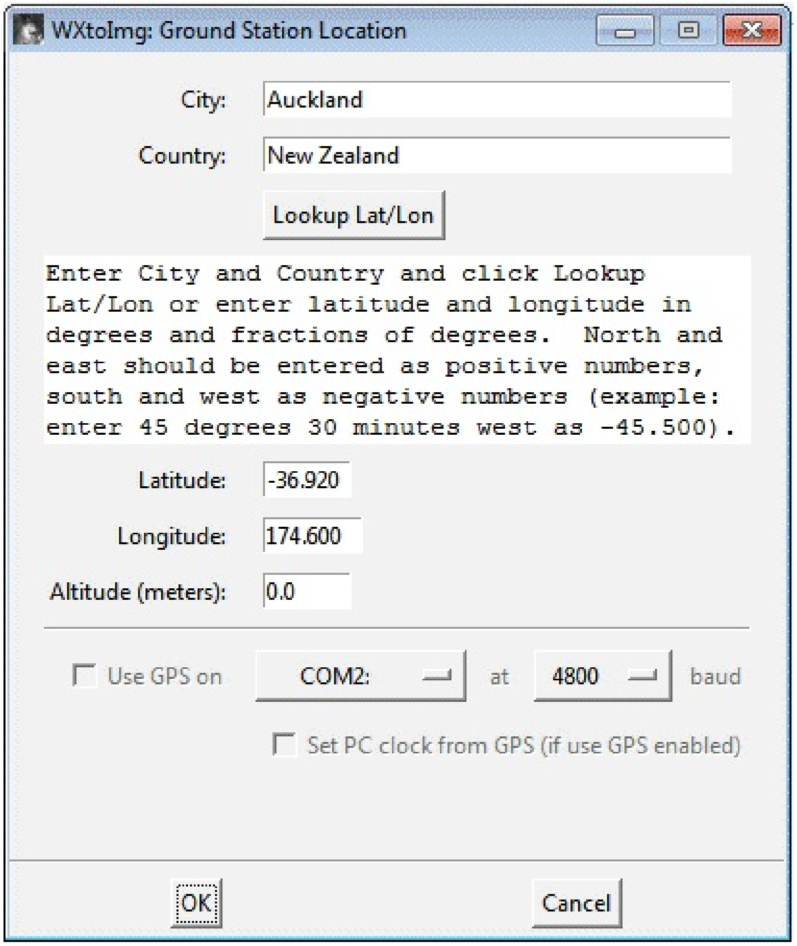
The NOAA satellites only pass overhead at certain times of the day, broadcasting a signal. These signals appear at around ~137 MHz, and only when a satellite is passing overhead. Each satellite uses a different frequency. Their frequencies are shown below.

- NOAA 15 - 137.6200 MHz
- NOAA 18 - 137.9125 MHz
- NOAA 19 - 137.1000 MHz

WXTOIMG SOFTWARE TUTORIAL

WXtoImg is a free weather satellite decoding program, which can 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 SDRSharp 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 height if you want.



City: Auckland

Country: New Zealand

```
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

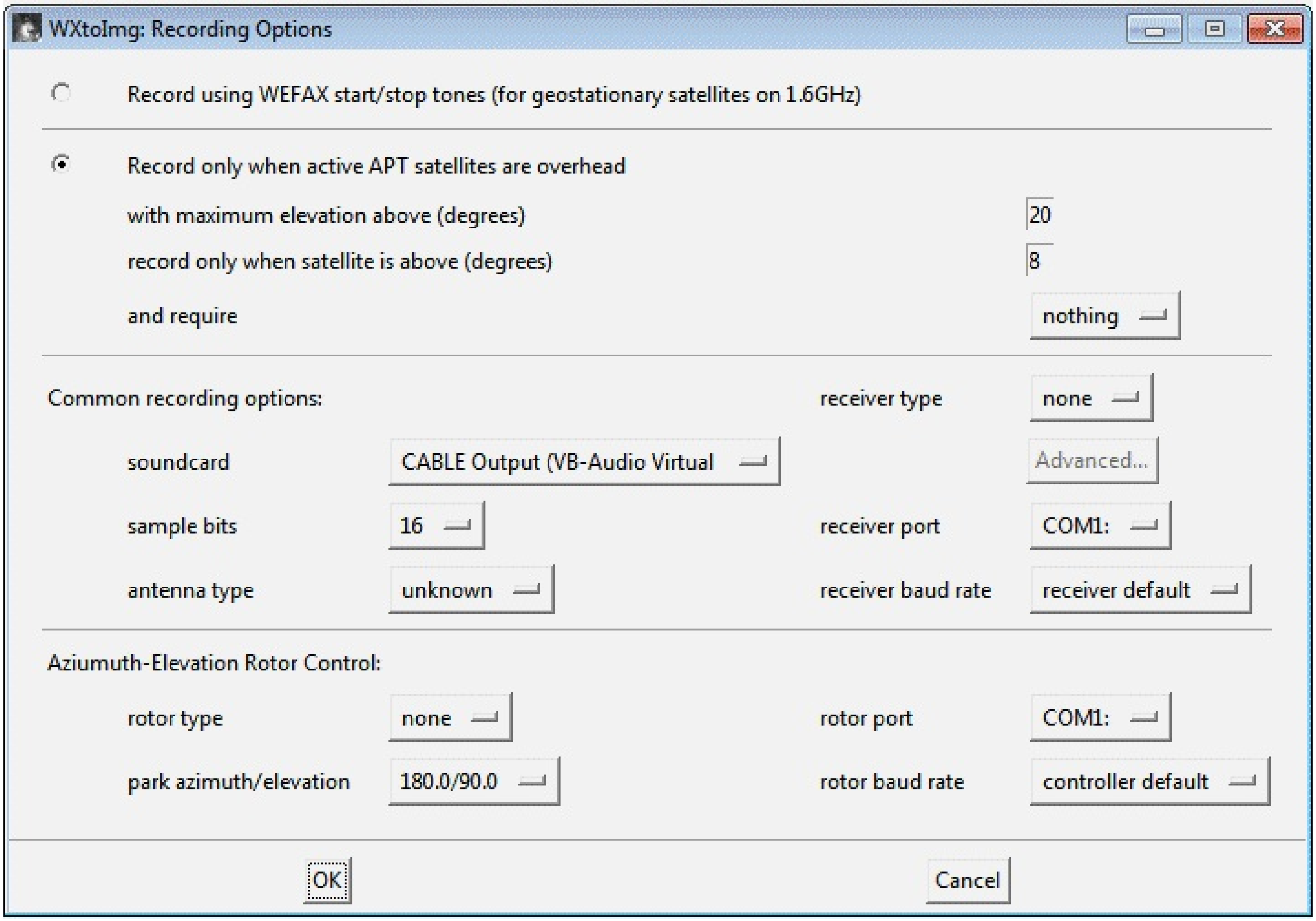
Altitude (meters): 0.0

☐ 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 device is selected under the soundcard option.

4. Also, here 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 SDRSharp.



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:

soundcard CABLE Output (VB-Audio Virtual

sample bits 16

antenna type unknown

receiver type none

receiver port COM1:

receiver baud rate receiver default

Aziumuth-Elevation Rotor Control:

rotor type none

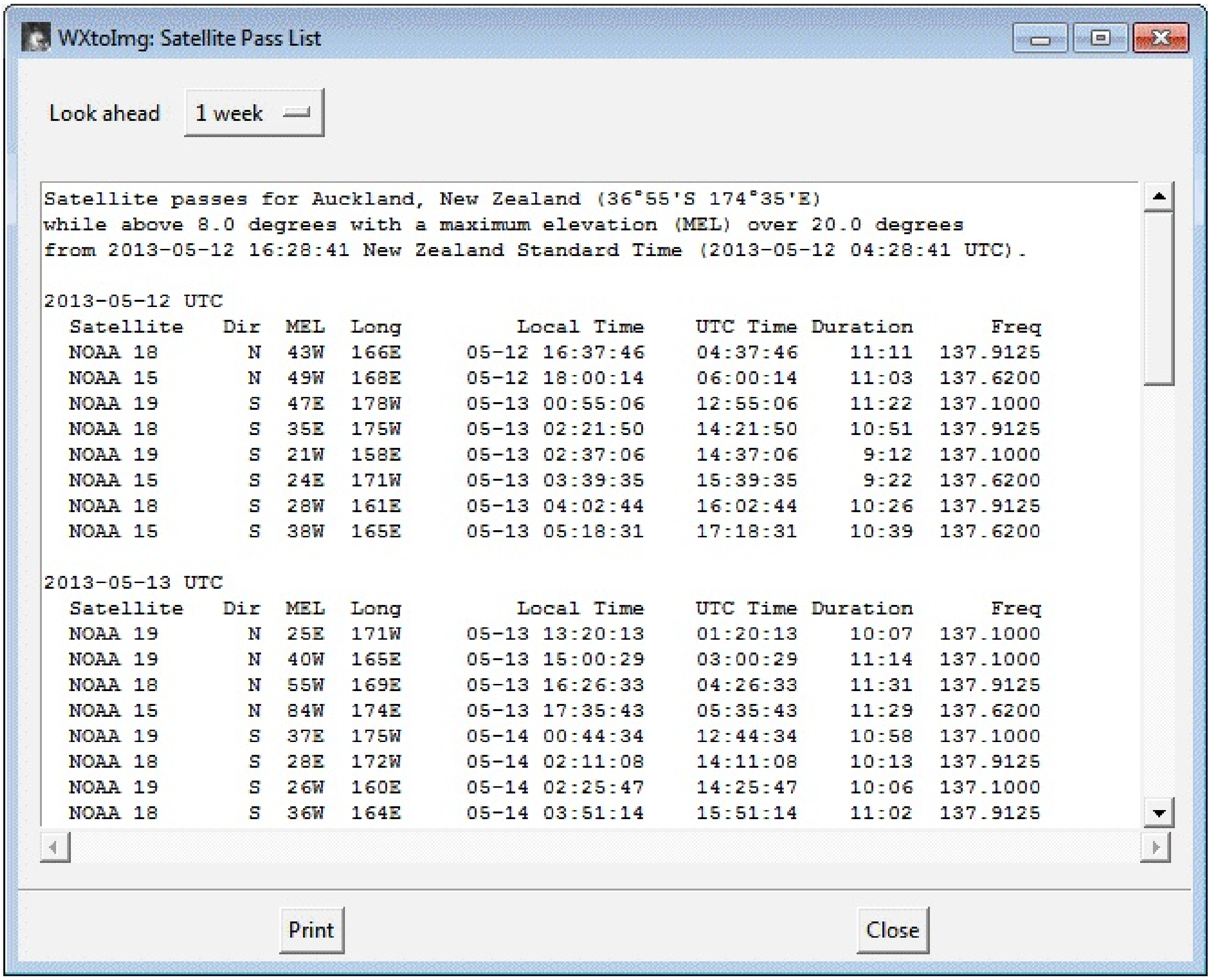
park azimuth/elevation 180.0/90.0

rotor port COM1:

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.

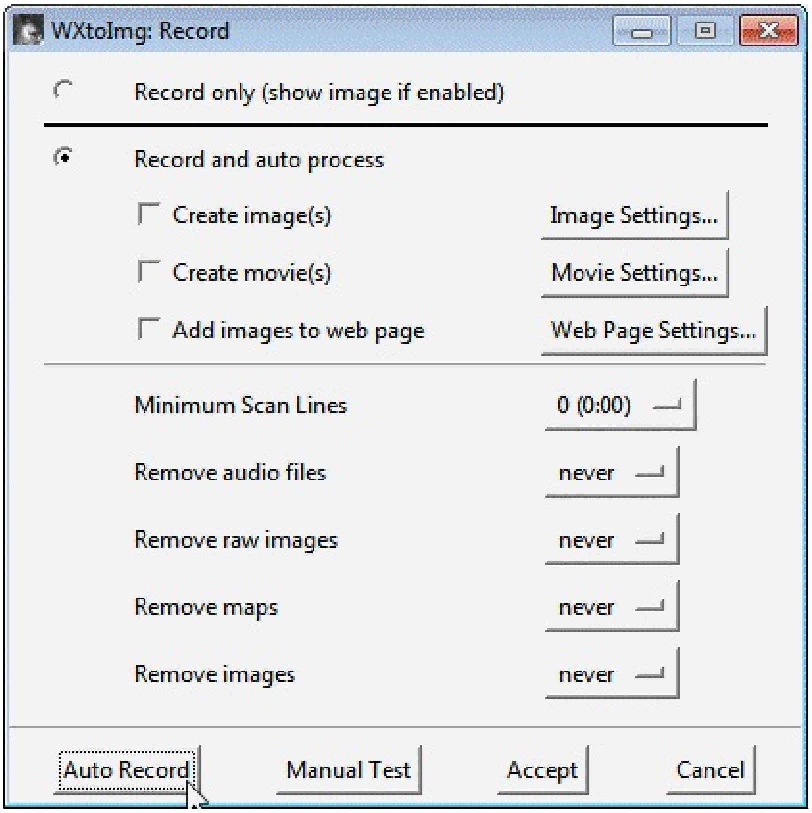
6. Now you can go to **File -> Satellite Pass List**, and find a time when a satellite will be passing overhead. Take note of the frequency of the pass as well.



Look ahead 1 week

```
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)

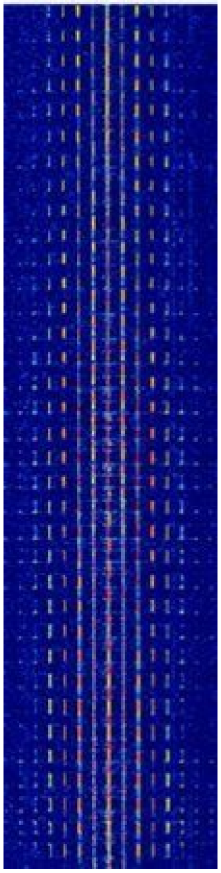
Remove audio files never

Remove raw images never

Remove maps never

Remove images never

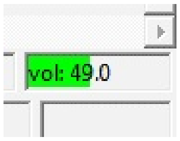
8. Open SDRSharp select the audio piping method you are using under the Audio Output drop down box, and then tune to the frequency that the satellite will be broadcasting at. Adjust the gain settings in SDRSharp under the Configure button so that you get good reception of the signal. Set the receive 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 SDRSharp until it is centred on the satellite signal.

10. Adjust the AF Gain in SDRSharp, and/or Windows volume settings so that the volume bar in the bottom right hand corner of WXtoImg shows a green colour.



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 satellites as they appear on your horizon. Also, although not entirely necessary for the RTL-SDR Orbitron will help to compensate for satellite frequency drift due to the doppler effect.

1. Download and install Orbiton from http://www.stoff.pl/.

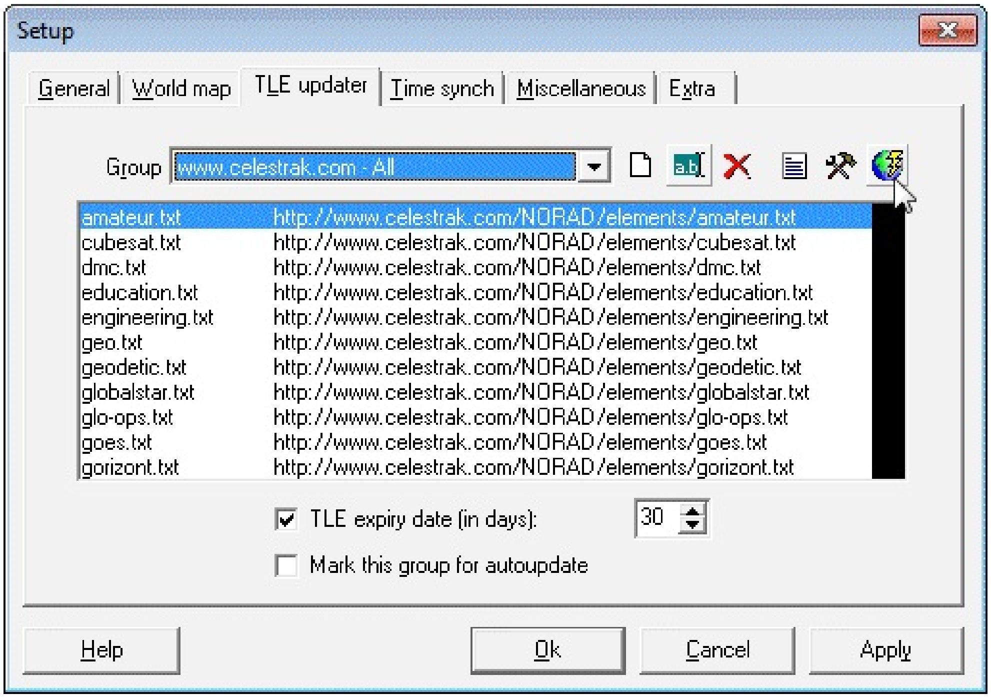
2. Download the Orbitron DDE drivers from http://www.stoff.pl/orbitron/files/mydde.zip. Extract the files to the Orbitron installation directory (likely in Program Files (x86)/Orbitron)

3. Download the SDRSharp Orbitron plugin from http://public-xrp.s3.amazonaws.com/SatelliteTracker2.zip. The readme file comes with installation instructions, but some of the instructions did not work for me. You can try follow their instructions or the ones below.

4. Extract the SDRSharp orbitron plugin files to the SDRSharp directory. With notepad or another text editor open SDRSharp.exe.config. Scroll down to the section where it says <sharpPlugins>. 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 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

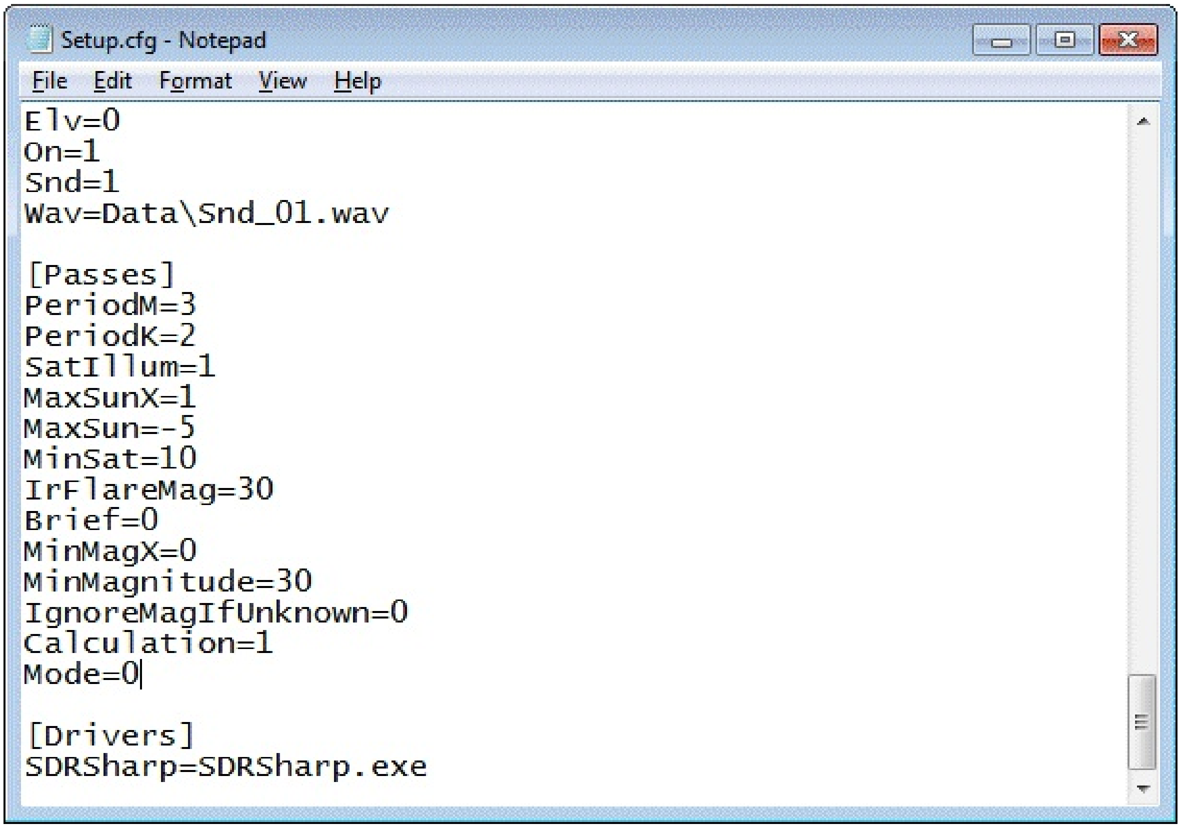
□ Mark this group for autoupdate

6. Close Orbiton. Now open Notepad in Administrator mode, by right clicking it’s shortcut in the Start Menu, and clicking on Run as Administrator.

7. 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.

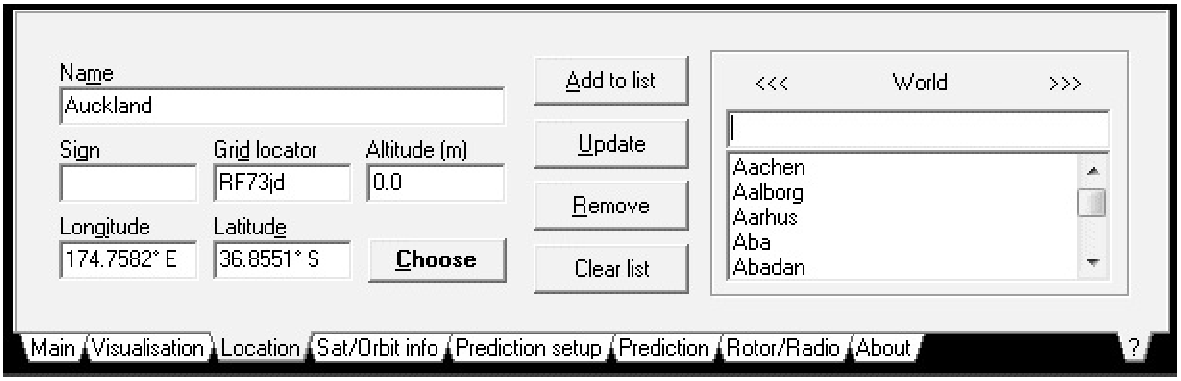
8. At the bottom of the Setup.cfg text file, add these two lines.

```
[Drivers]
SDRSharp=SDRSharp.exe
```

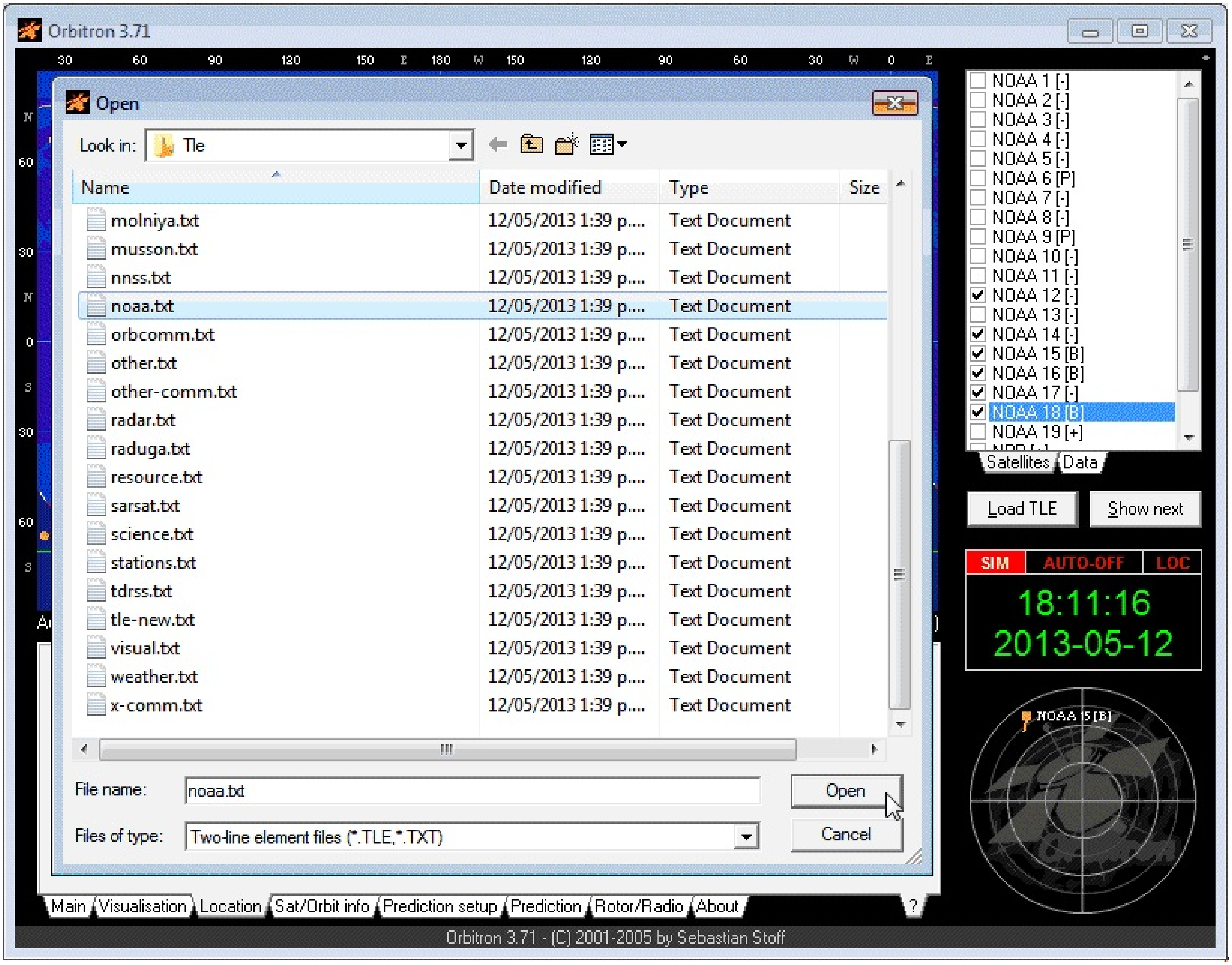


```
Elv=0
On=1
Snd=1
Wav=Data\Snd_01.wav

[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
```

9. 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

10. Next click on Load TLE, and load the noaa.txt file.



Look in: Tle

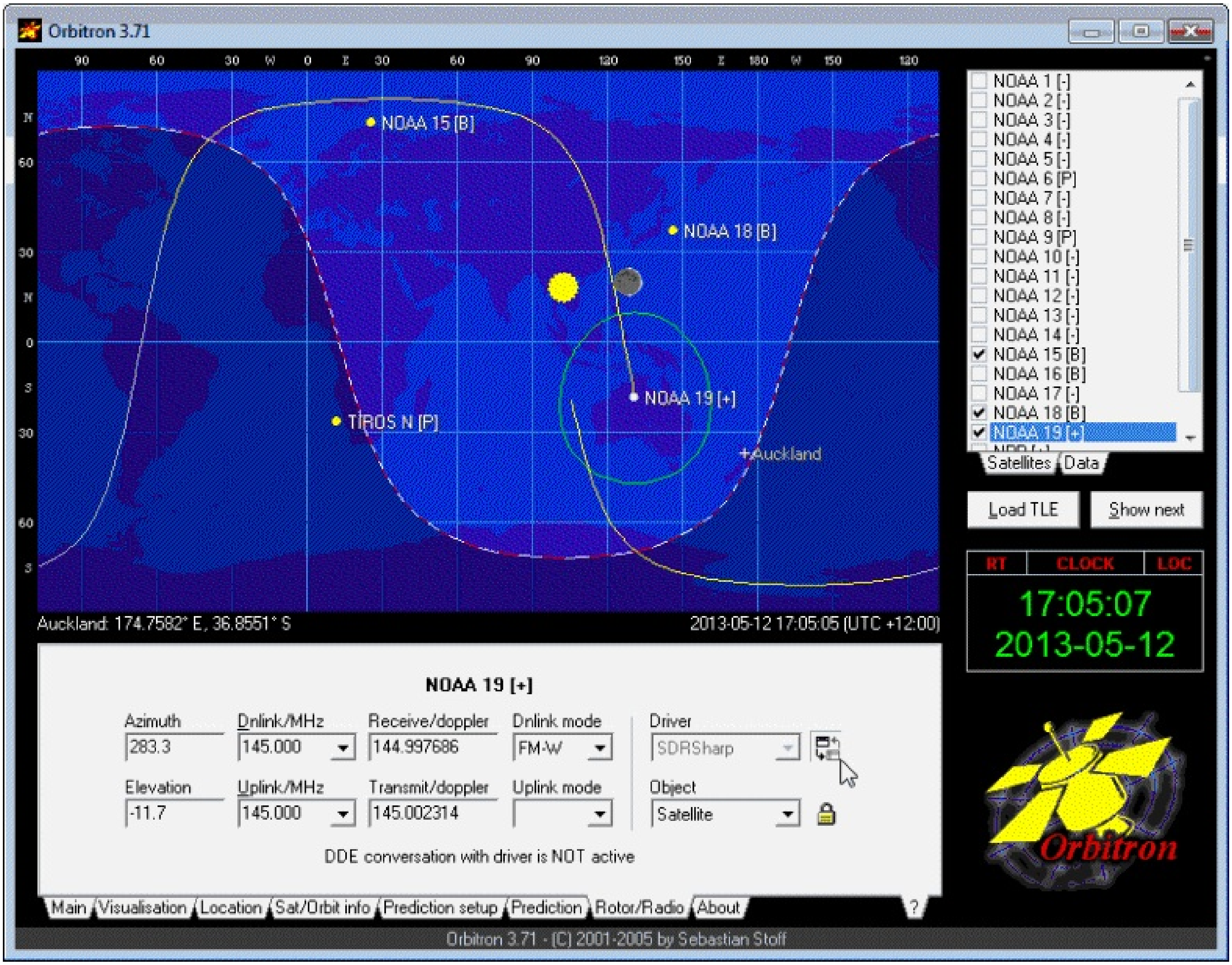
Name	Date modified	Type	Size
molniya.txt	12/05/2013 1:39 p....	Text Document	
musson.txt	12/05/2013 1:39 p....	Text Document	
nnss.txt	12/05/2013 1:39 p....	Text Document	
noaa.txt	12/05/2013 1:39 p....	Text Document	
orbcomm.txt	12/05/2013 1:39 p....	Text Document	
other.txt	12/05/2013 1:39 p....	Text 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)

☐ 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 [+]

11. 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 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 [+]

17:05:07
2013-05-12

NOAA 19 [+]

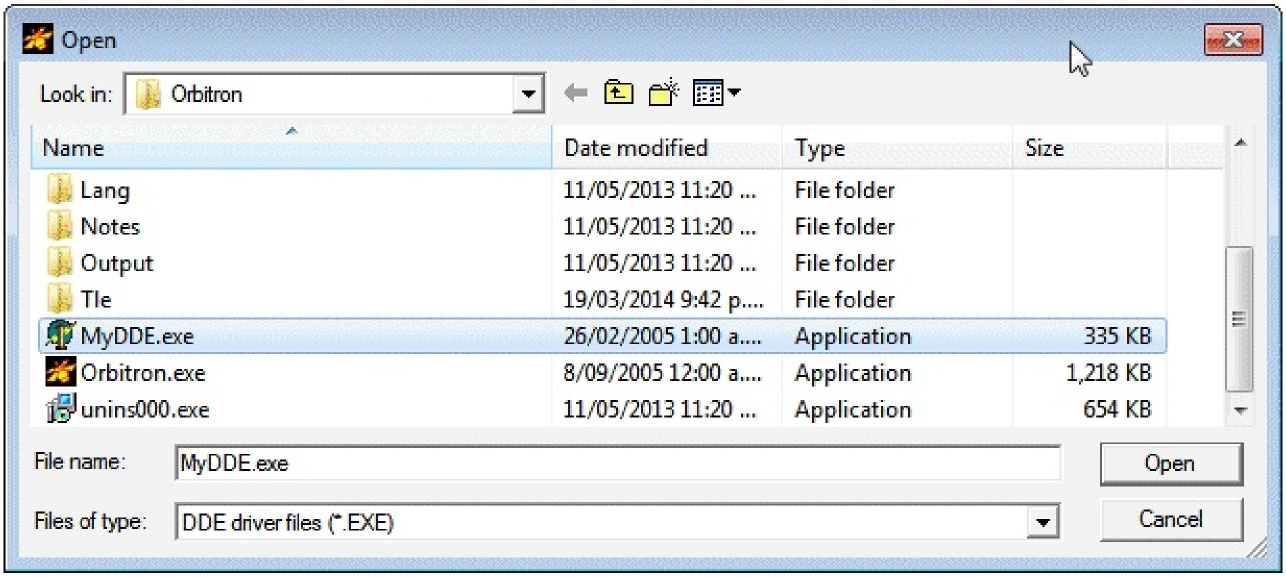
Azimuth	Dnlink/MHz	Receive/doppler	Dnlink mode	Driver
283.3	145.000	144.997686	FM-W	SDRSharp
Elevation	Uplink/MHz	Transmit/doppler	Uplink mode	Object
-11.7	145.000	145.002314		Satellite

DDE conversation with driver is NOT active

Main | Visualisation | Location | Sat/Orbit info | Prediction setup | Prediction | Rotor/Radio | About | ?

Orbitron 3.71 - (C) 2001-2005 by Sebastian Stoff

12. 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.



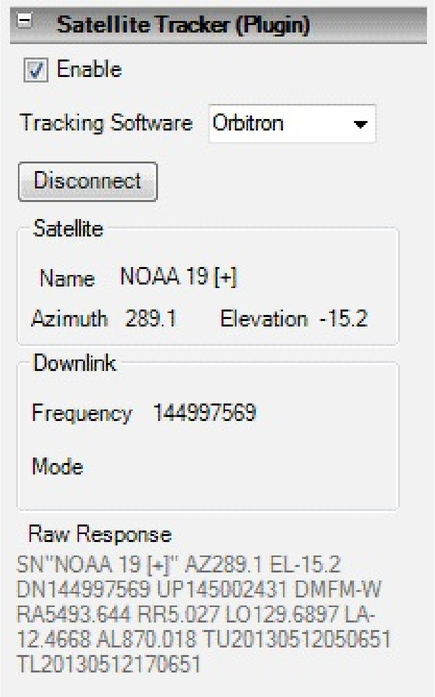
Look in: Orbitron

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)

13. Open SDRSharp, press Play and then head to the Orbitron Plugin. Put a check next to Enable, set the Tracking Software to Orbitron and then click connect. Now double clicking on a satellite name in Orbitron should set the correct Doppler corrected frequency in SDRSharp.



Satellite Tracker (Plugin)

☑ Enable

Tracking Software: Orbitron

Satellite

Name: NOAA 19 [+]

Azimuth: 289.1

Elevation: -15.2

Downlink

Frequency: 144997569

Mode

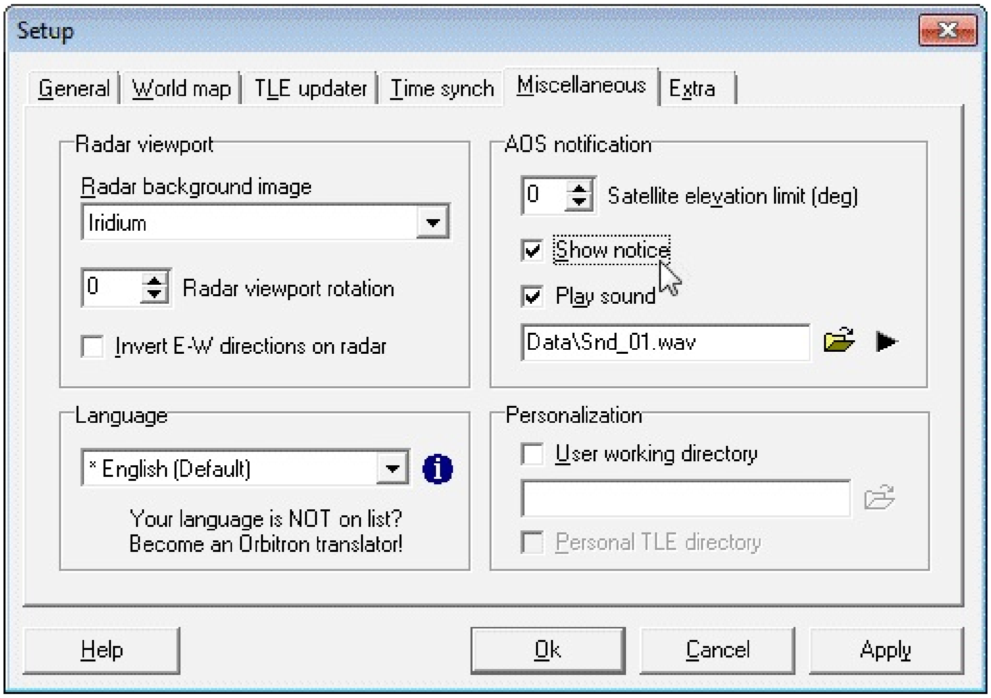
Raw Response

```
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
```

14. SDRSharp 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 SDRSharp configure box to ensure the signal is centred.

15. 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).

16. 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)

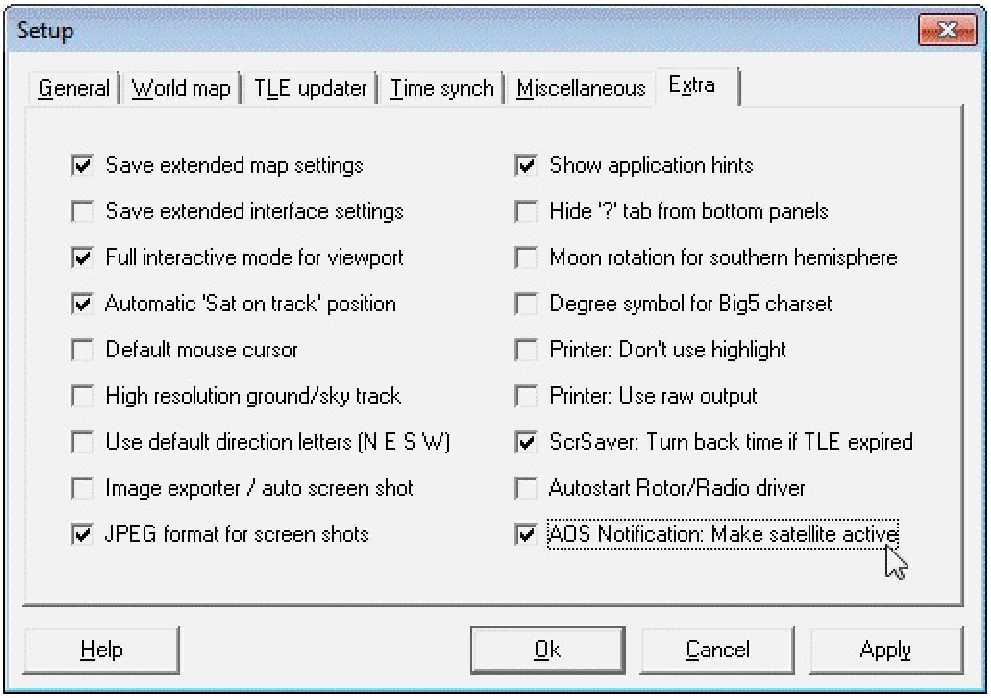
Your language is NOT on list?
Become an Orbitron translator!

Personalization

□ User working directory

□ Personal TLE directory

17. 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

18. 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**.

WEATHER BALLOON (RADIOSONDE) GUIDE

Around the world meteorological weather balloons are launched twice daily, and 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 it's ascent.

Note that currently, this won't work in many places in the USA, as there are no decoding programs available for use with the telemetry protocol used in the states. The USA uses the Mark II Sippican radiosondes. Most of the rest of the world use Graw, Vaisala, Meteomodem or Meteolabor radiosondes, all of which are decodable. However, it seems that the USA is undergoing a Radiosonde Replacement System (RRS), where they are switching to Vaisala RS92SGP radiosondes which are decodable. Unfortunately, they appear to be using the more difficult to receive frequency of 1680 MHz. See http://www.ua.nws.noaa.gov/RRS.htm for more information. SondeMonitor is able to decode Digital RS92SGP, Analogue RS92AGP, Analogue RS80, Analogue 92KL, Graw DFM-06, MeteoModem M2K2/M10 and Meteolabor SRS-C34.

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

Most radiosondes transmit at a frequency between 403 to 406 MHz, so you will need an antenna capable of receiving these frequencies. Antennas good for this purpose can be a quarter wave groundplane, dipole, j-pole, turnstile or QFH antenna. A Yagi can work if you require more gain from a weak signal and are willing to track the radiosonde by hand. See the antenna guide for more information about these antennas. Some countries use radiosondes that transmit at 1680 MHz. For this frequency you will likely need a directional Yagi antenna pointed by hand or other means in the direction of the radiosonde.

An LNA and preselector for the 403-406 MHz frequencies may also aid reception. See the LNA section and preselector section for more information.

SOFTWARE TUTORIAL

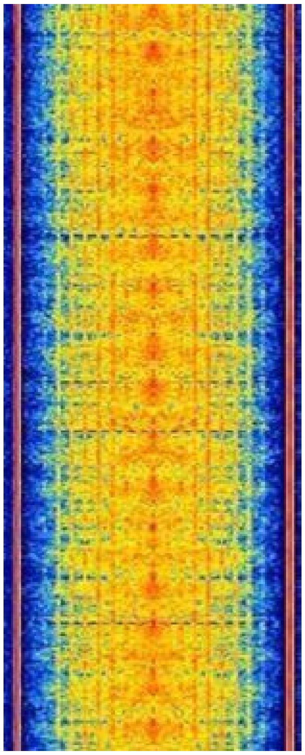
In order to receive and decode weather balloon radiosondes you will need four things.

- 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 isused in your area (can be guessed with trial and error).

Weather balloons most commonly transmit at around 403 MHz and 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

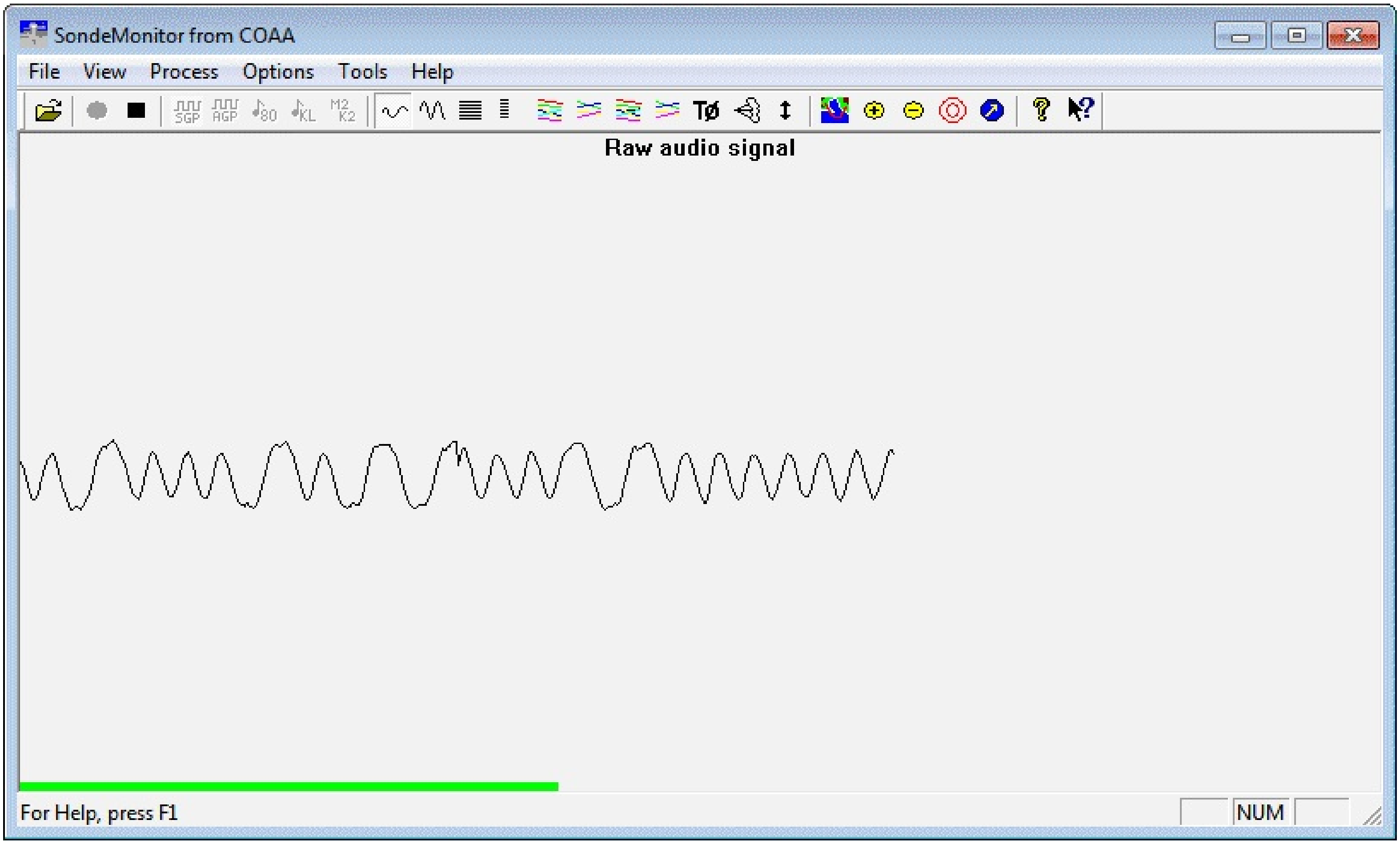
1. Download and install SondeMonitor. SondeMonitor is paid software with a 25 euro price tag, but comes with a 21 day trial. Download SondeMonitor from http://www.coaa.co.uk/sondemonitor.htm.
2. When a weather balloon is due to be launched, open SDRSharp, set your audio piping method under the Audio Output drop down box and tune to a frequency between 403 and 406 MHz. Scan

around and look for a radiosonde signal, tune to it and adjust the RF gain settings for best performance. Set the receive mode to NFM, filter order to 10, filter audio to OFF. Adjust the filter bandwidth so that it just covers the signal.

3. Open SondeMonitor. Go to **Options -> Audio -> Audio Source** and choose the audio piping method you are using.

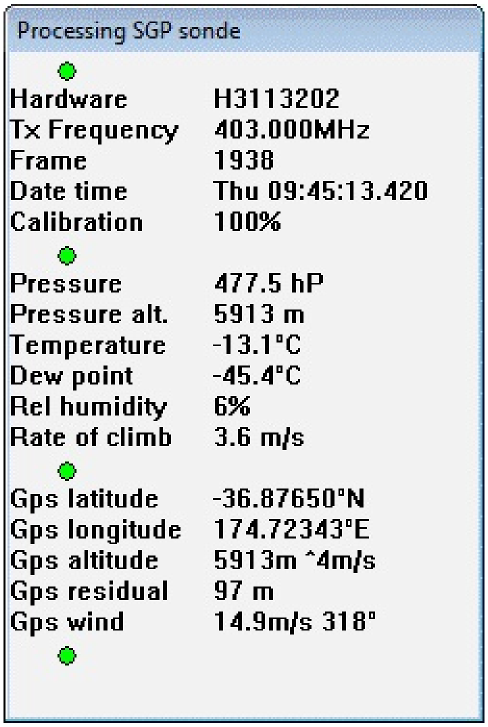
4. Select the radiosonde decoding protocol used in your area. These can be selected with these icons SGP AGP 80 KL M2 K2 which are to the right of the start and stop icons. Hovering over the icons will show which protocol they represent.

5. Now start the decoding by clicking the green circle ●. A telemetry data window will pop up. Next, click on the raw signal icon ∿. Adjust the audio volume using the audio gain in SDRSharp 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 completed amount of the calibration process 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

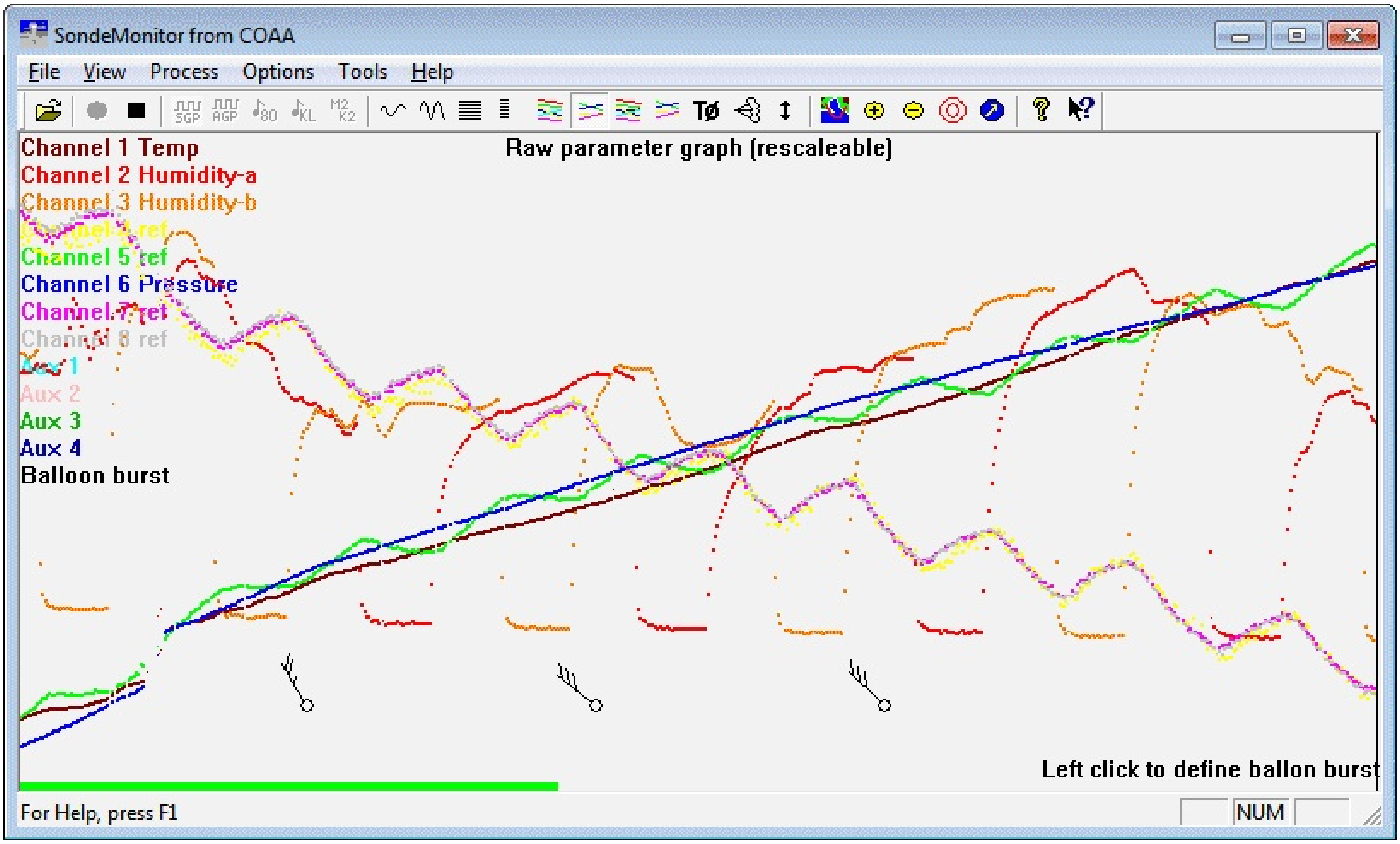
calibration does not complete, or takes a very long time, it means your signal strength is not good enough.



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.



Left click to define ballon burst

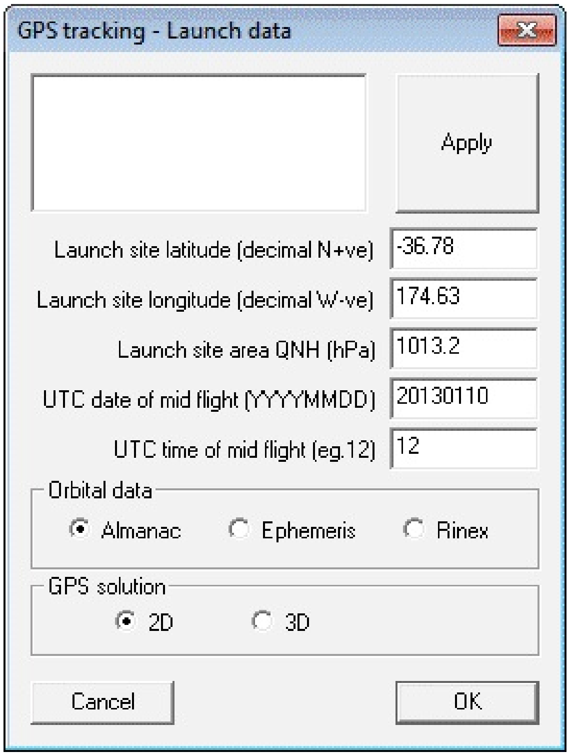
For Help, press F1

For most launches, the weather balloon's 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 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.



GPS tracking - Launch data

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

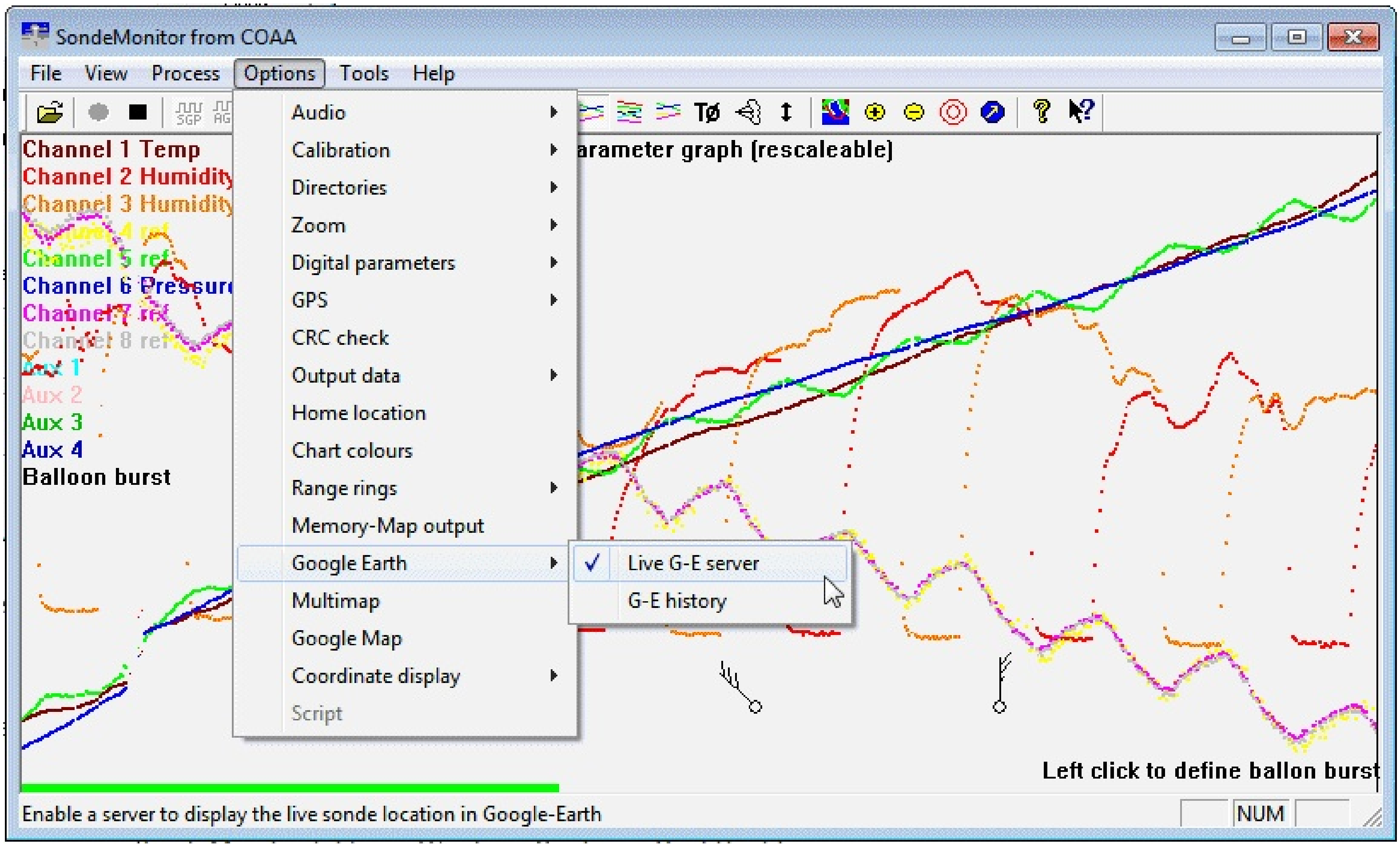
Orbital data: ● Almanac ○ Ephemeris ○ Rinex

GPS solution: ● 2D ○ 3D

3. Once you press OK, a file selection dialog box will pop up. Browse to the folder your current .al3 Almanac file is stored in, and open it. Now, if you have set the Almanac, start coordinates, and date and time 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

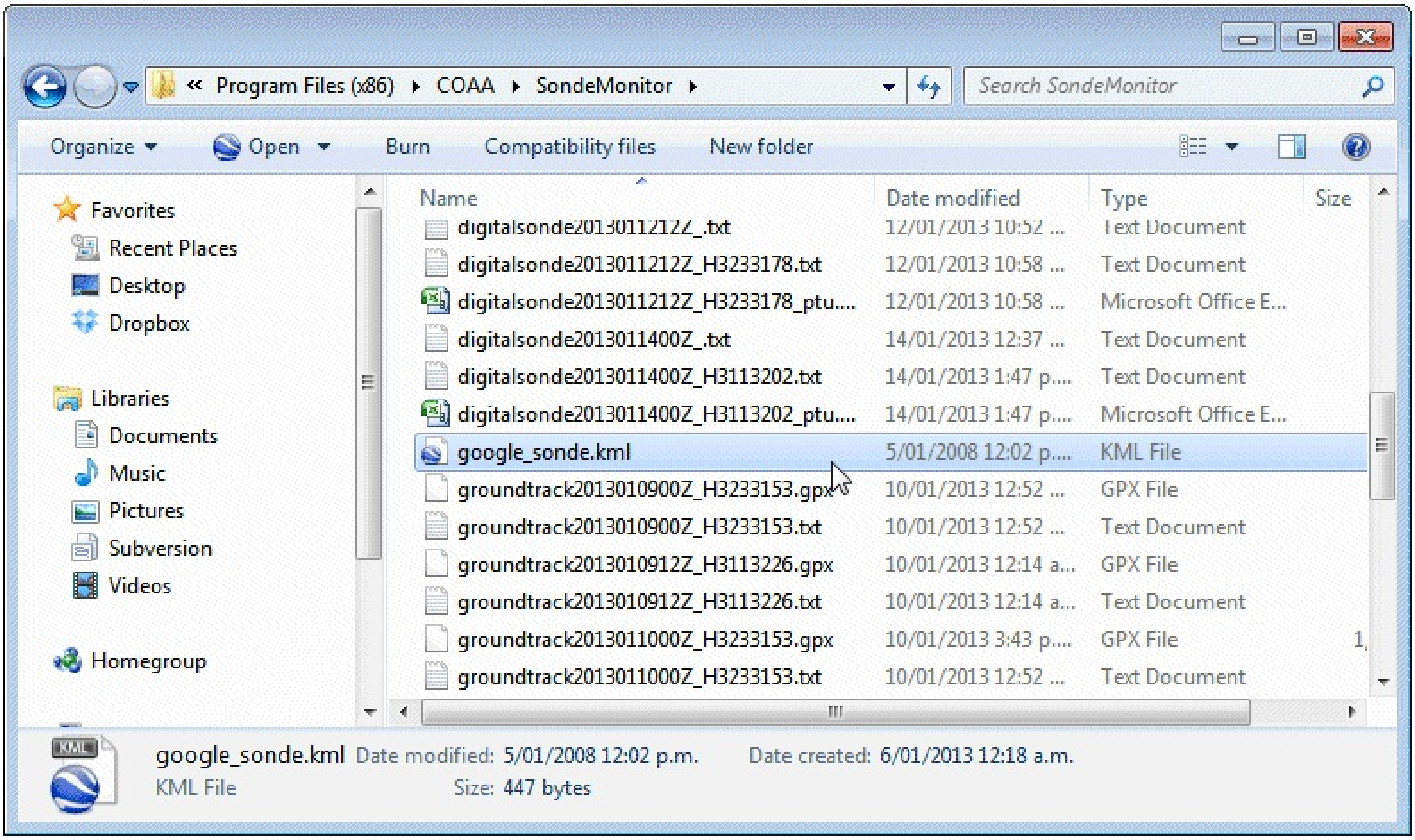
balloon location is. If you have gotten any parameters wrong, the residual will be very large.

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.



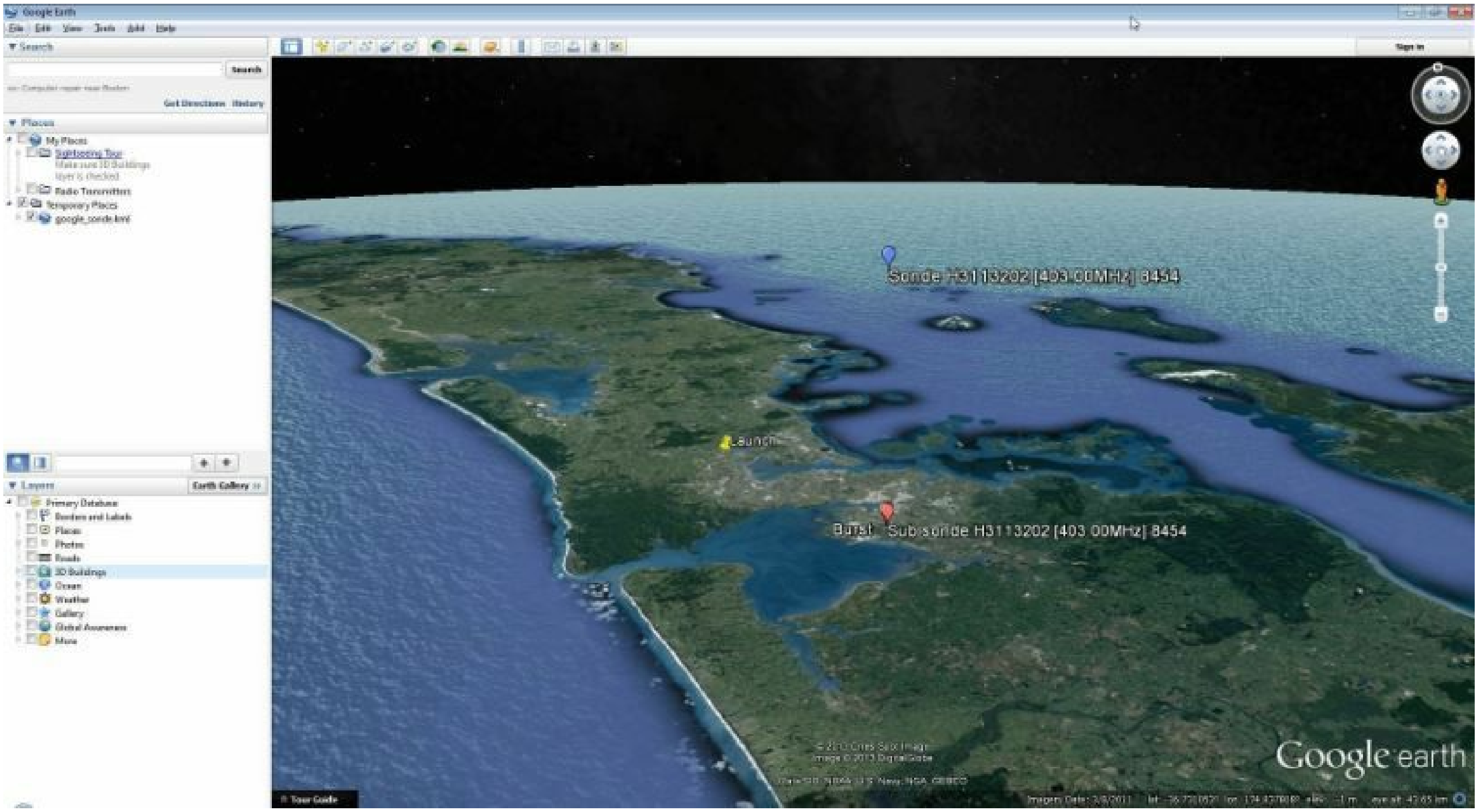
Enable a server to display the live sonde location in Google-Earth

5. Now open your SondeMonitor 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

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 ADS-B 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.
- 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 quickstart guide for an easy setup routine with SDRSharp. You will also need to have an audio piping method installed and set up. Audio piping will allow the audio from SDRSharp 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 74 kilometres. So if your radio setup is more than 74 kilometres away from any boats, you will probably not be able to receive AIS signals. 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.

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.

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 to 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 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 SDRSharp or SDR-CONSOLE 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.

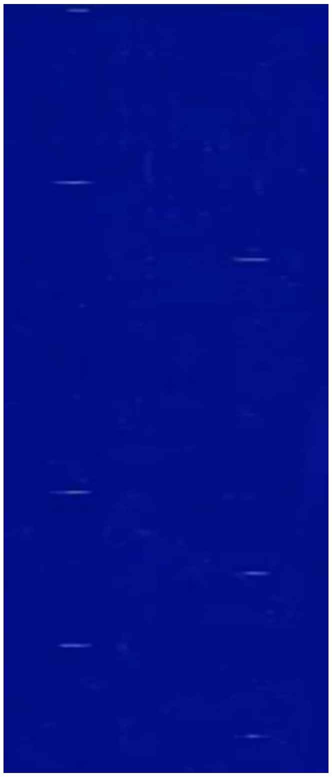
3. For AIS we highly recommend using SDR-CONSOLE as it will allow us to listen to both AIS channels simultaneously. If using SDR-CONSOLE, set two VFOs to listen to both AIS channels. Ensure that each one outputs to a separate audio piping device.

4. Adjust the RF gain in the 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.

5. 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 the Filter Order to 10 and Filter Audio to OFF. Setting IF digital noise reduction could possibly help.

6. AIS signals look like small horizontal lines on the waterfall, as is shown on the image below. 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 is that we can open two instances of the program and use them to decode the two AIS channels simultaneously.

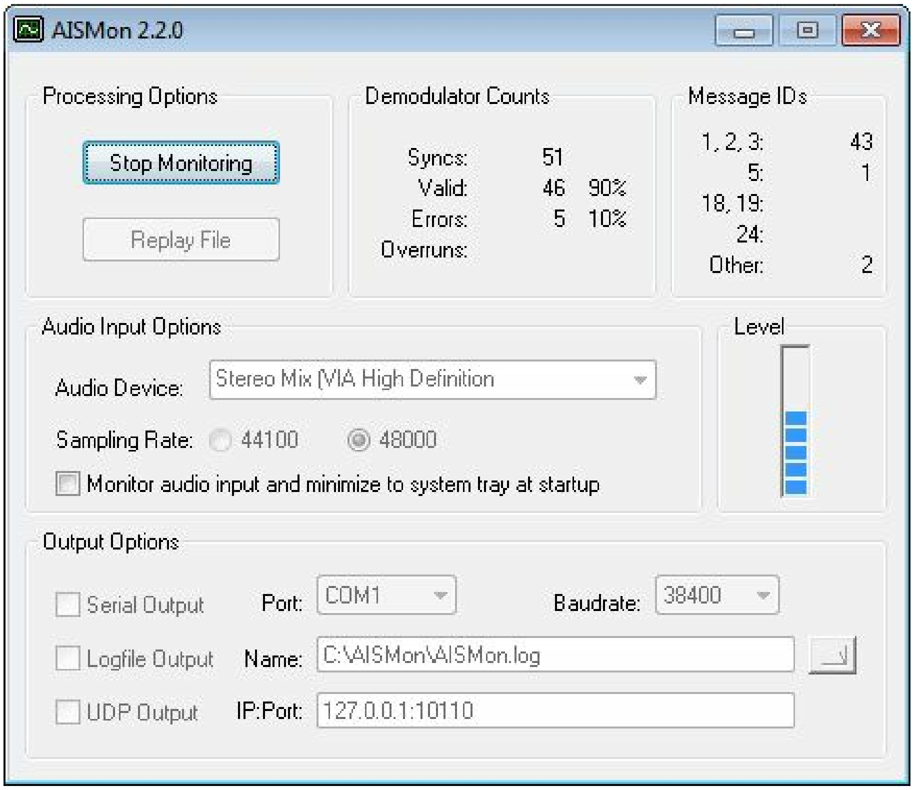
1. Open AISMon, set the audio piping device to the one you have chosen and set the sampling rate

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 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.

4. Open a second instance of AISMon and repeat steps 1-3 for the second audio pipe if you are using SDR-CONSOLE.



AISMon 2.2.0

Processing Options

Stop Monitoring

Replay File

Demodulator Counts

Syncs: 51
Valid: 46 90%
Errors: 5 10%
Overruns:

Message IDs

1, 2, 3: 43
5: 1
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

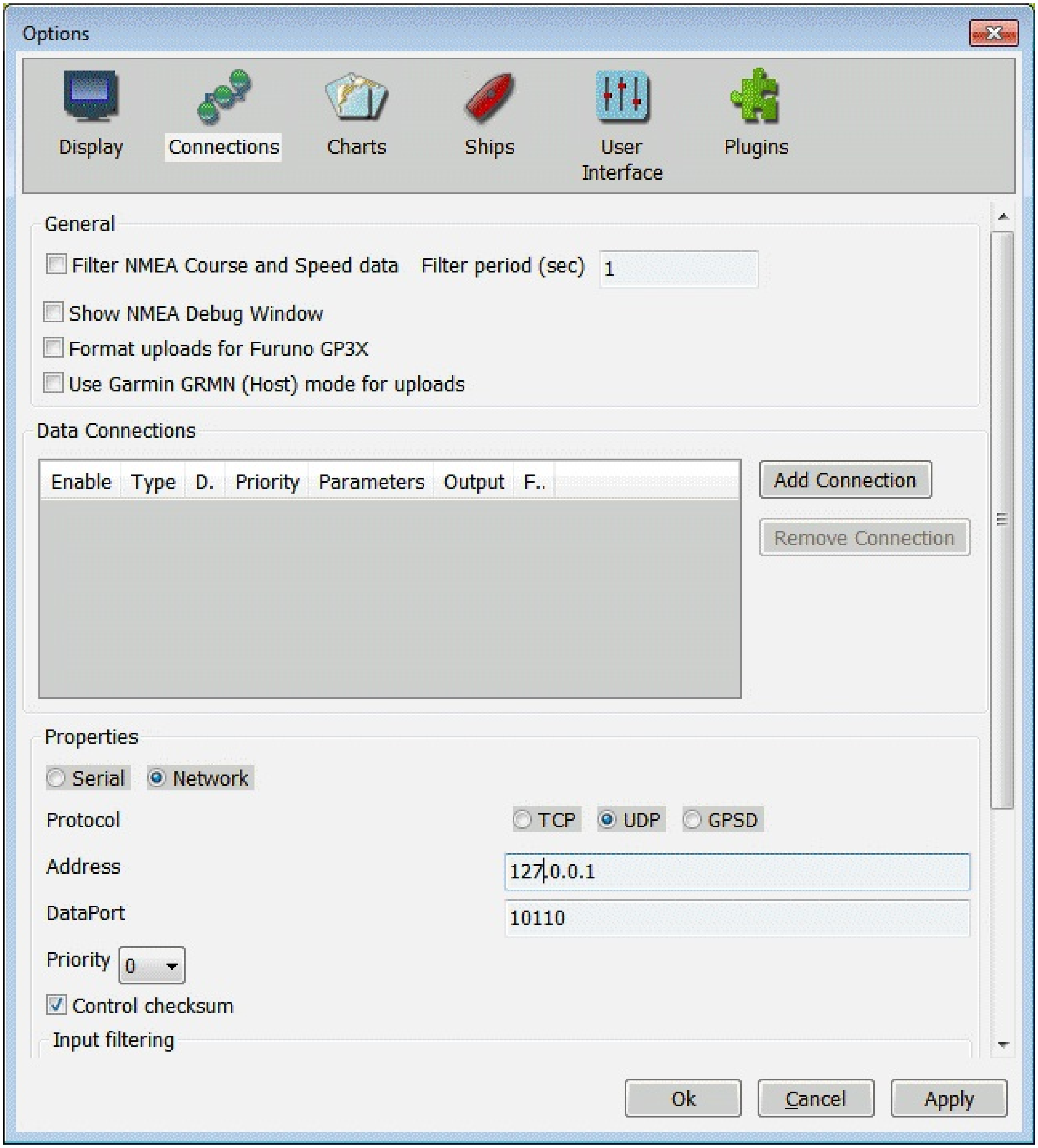
□ UDP Output IP:Port: 127.0.0.1:10110

By itself, AISMon does not display any ship information. To view ship information you will need another free opensource program called OpenCPN which can be downloaded from http://opencpn.org/ocpn/. OpenCPN is a chart plotting and navigation software which can read the NMEA information that is output by AISMon, and plot the ship positions on a map.

1. Download and install OpenCPN.

2. Click on the Options button 🔧 in OpenCPN 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-CONSOLE and are using two audio outputs, add a second UDP connection identical to the first one, but with a port number of 10120.



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

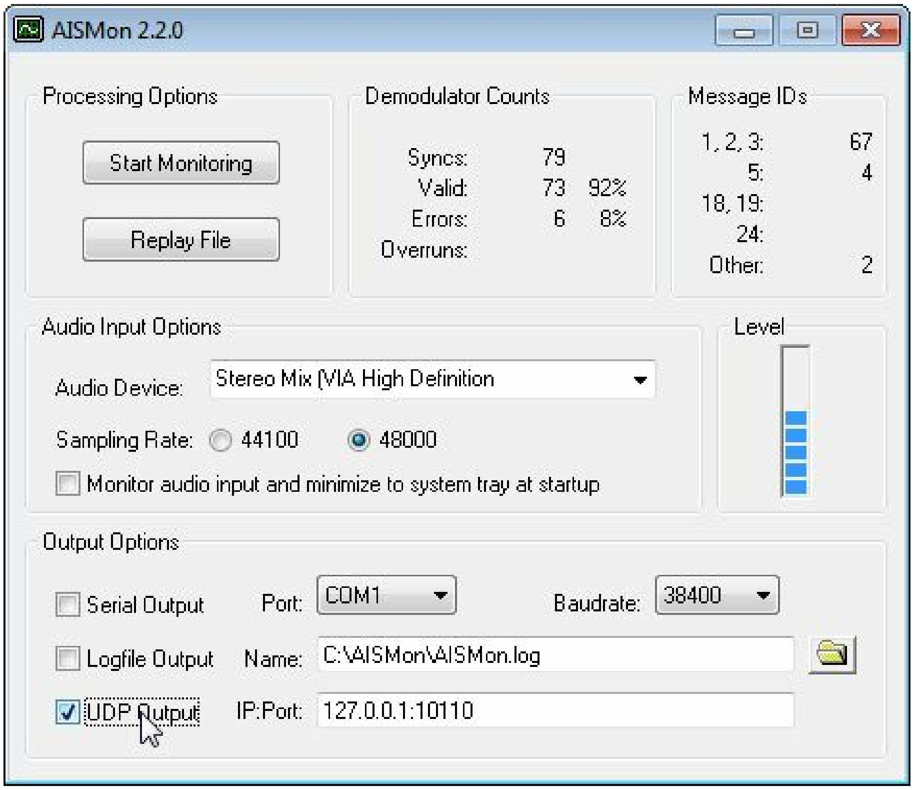
Priority 0

☑ 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 if monitoring is currently active).

5. If you are running SDR-CONSOLE 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.



AISMon 2.2.0

Processing Options

Start Monitoring

Replay File

Demodulator Counts

Syncs:	79	
Valid:	73	92%
Errors:	6	8%
Overruns:		

Message IDs

1, 2, 3:	67
5:	4
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

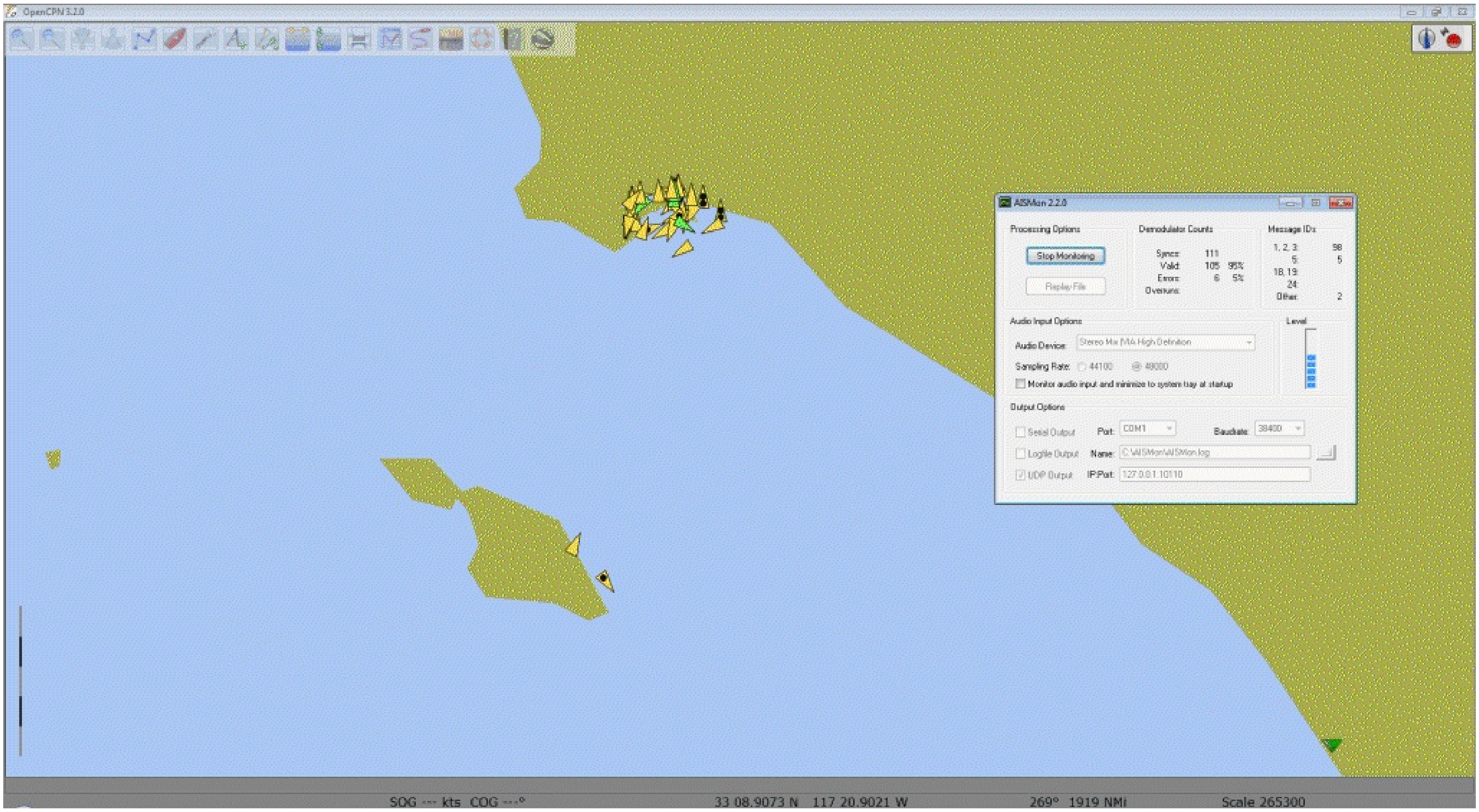
Output Options

□ Serial Output Port: COM1 Baudrate: 38400

□ Logfile Output Name: C:\AISMon\AISMon.log

☑ UDP Output IP:Port: 127.0.0.1:10110

6. You can now click Start Monitoring again. If everything has been set up correctly, ships will begin to appear in the OpenCPN map.



OpenCPN 3.2.0

AISMon 2.2.0

Processing Options

Stop Monitoring

Replay File

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

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.

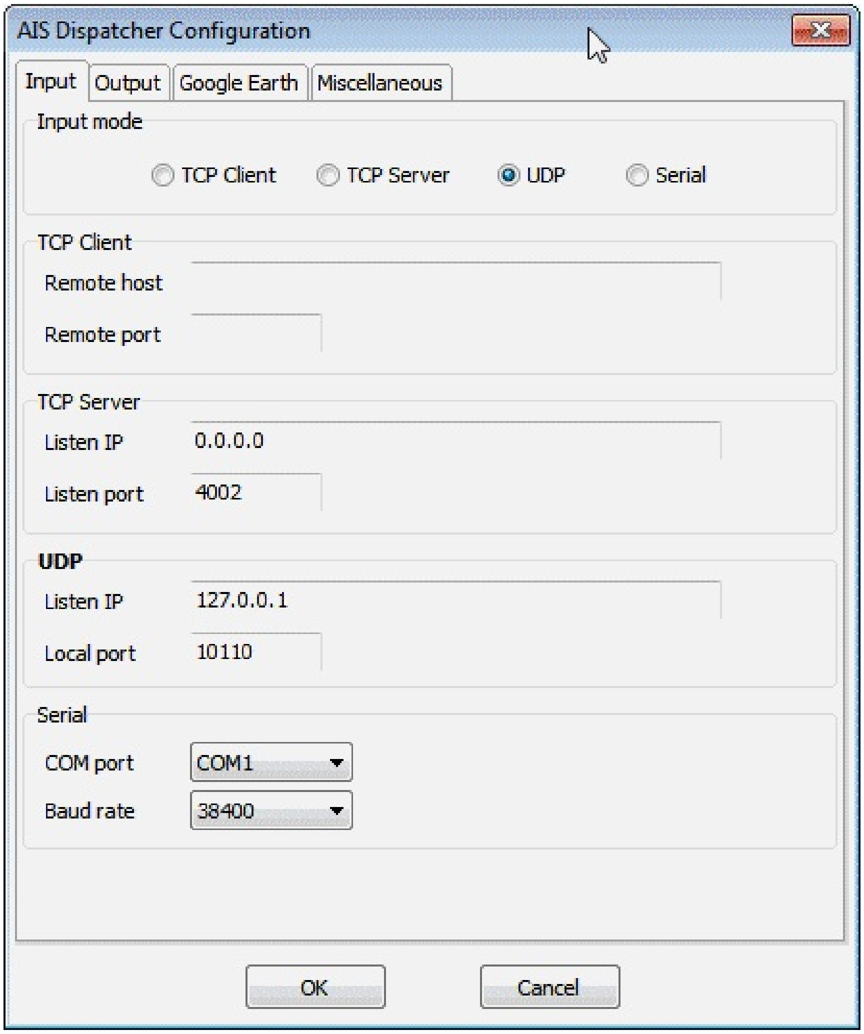
AIS DISPATCHER + AISMON TUTORIAL

A simple but 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

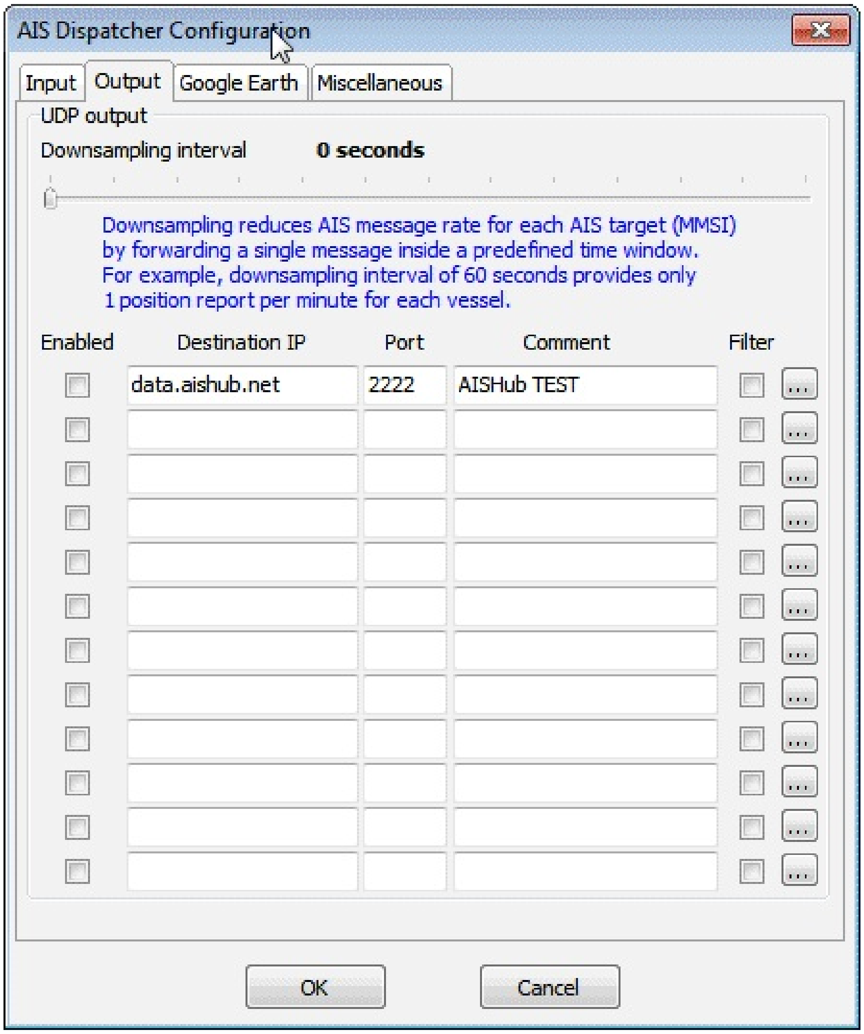
Local port 10110

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.



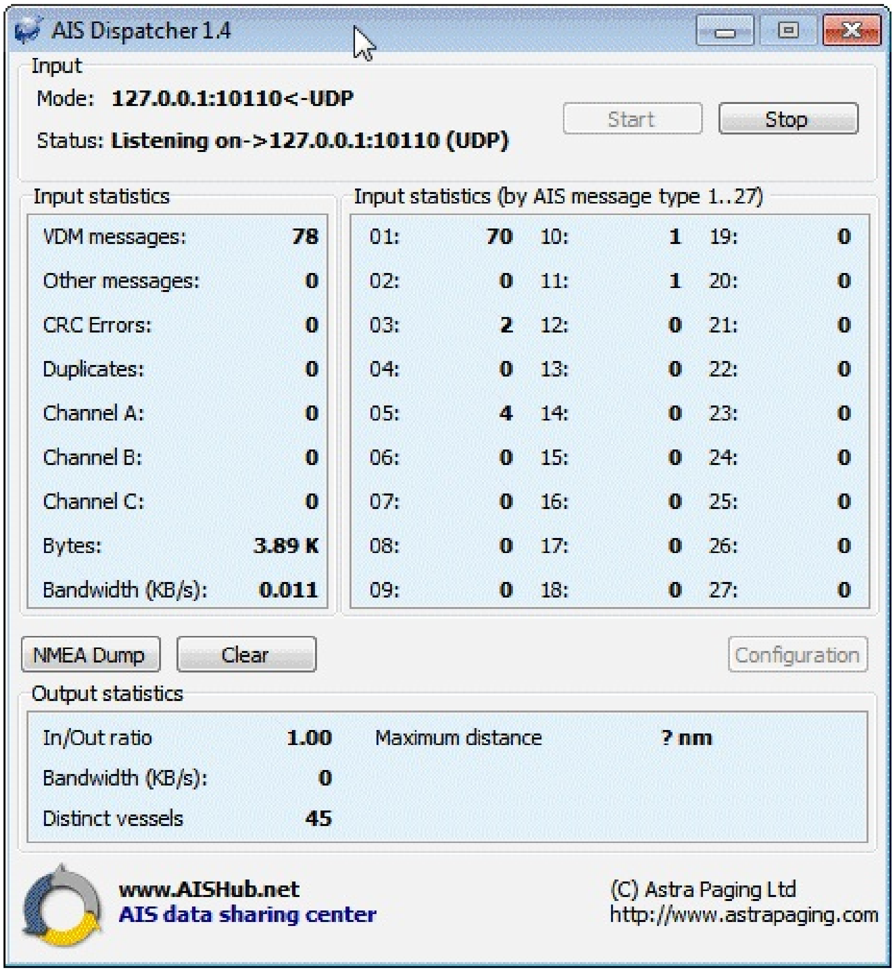
UDP output

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.



AIS Dispatcher 1.4

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

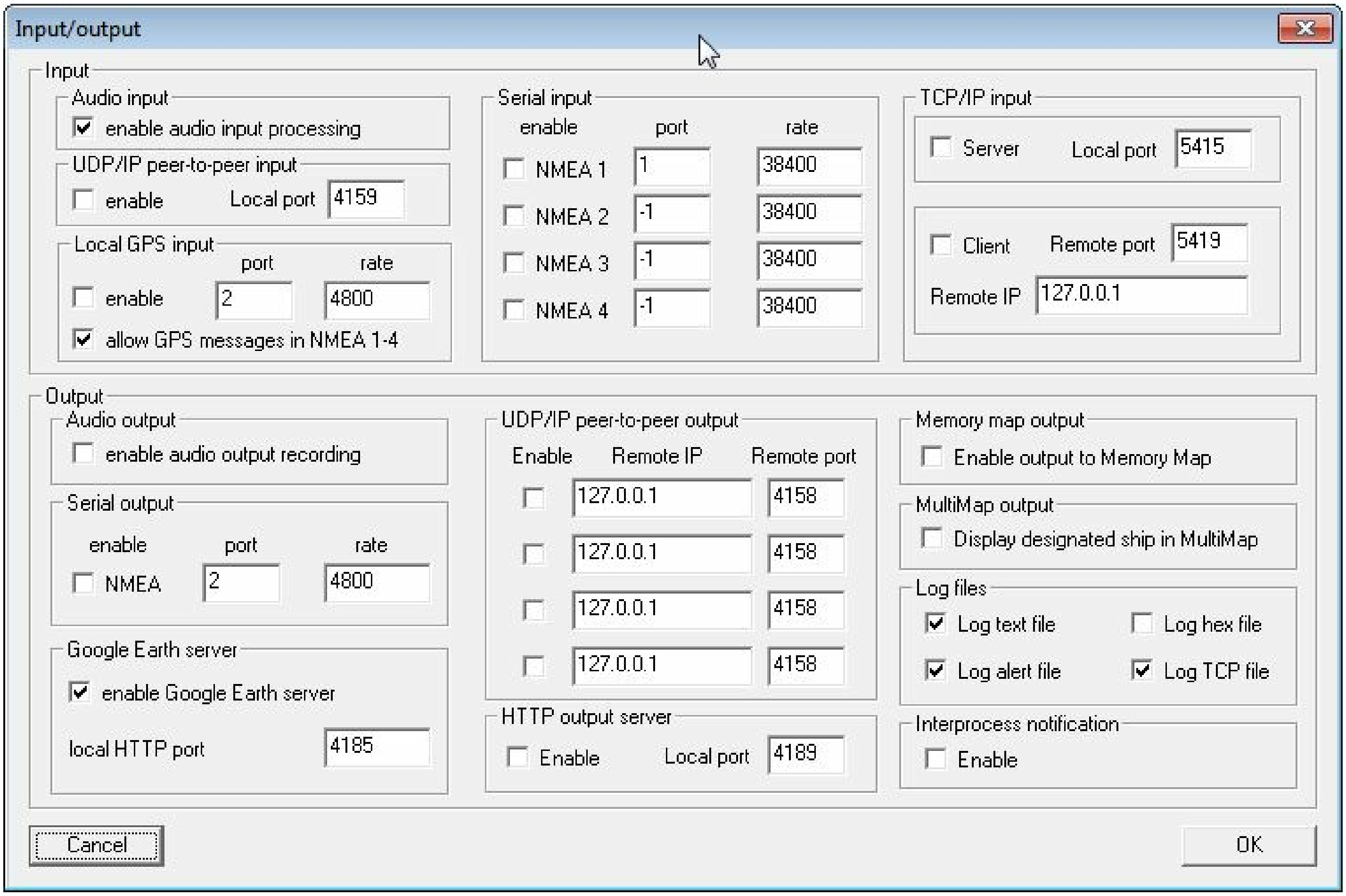
(C) Astra Paging Ltd
http://www.astrapaging.com

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. 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

	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

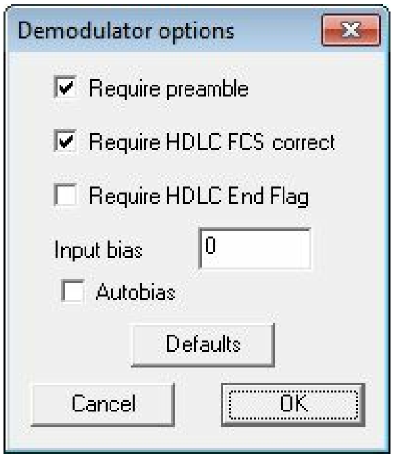
☑ Log text file □ Log hex file

☑ Log alert file ☑ Log TCP file

Interprocess notification

□ Enable

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

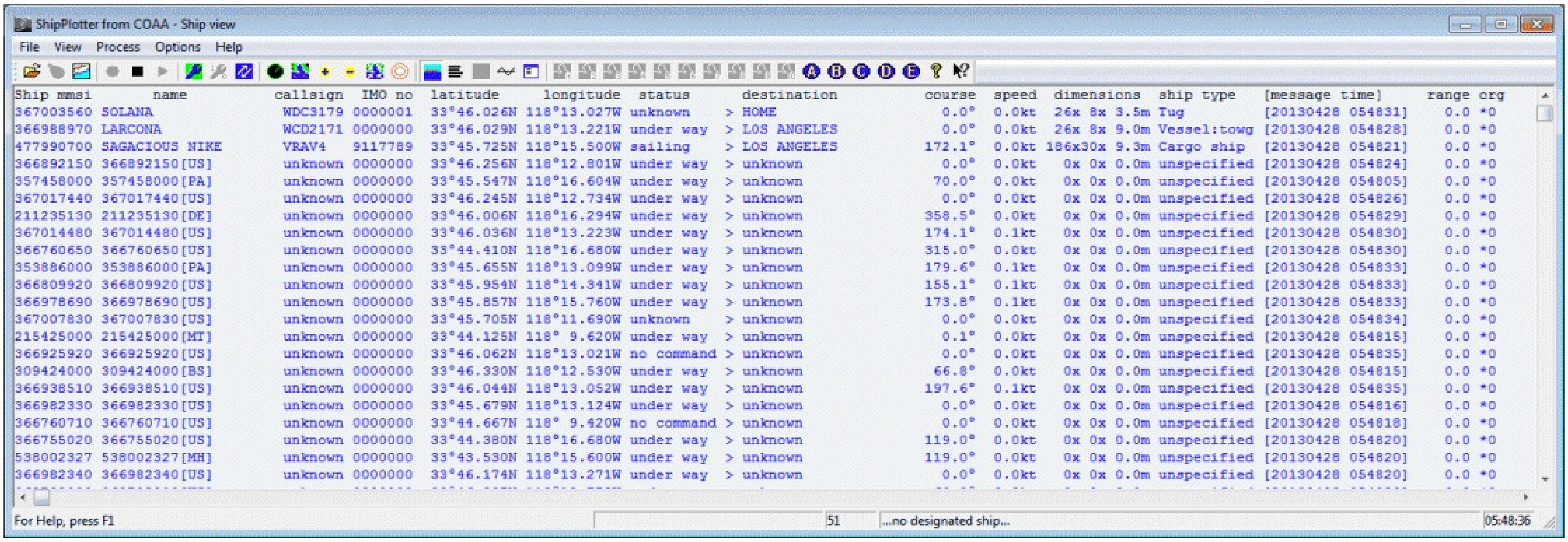
□ Require HDLC End Flag

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 (sine wave) 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 SDRSharp, or the windows volume settings.

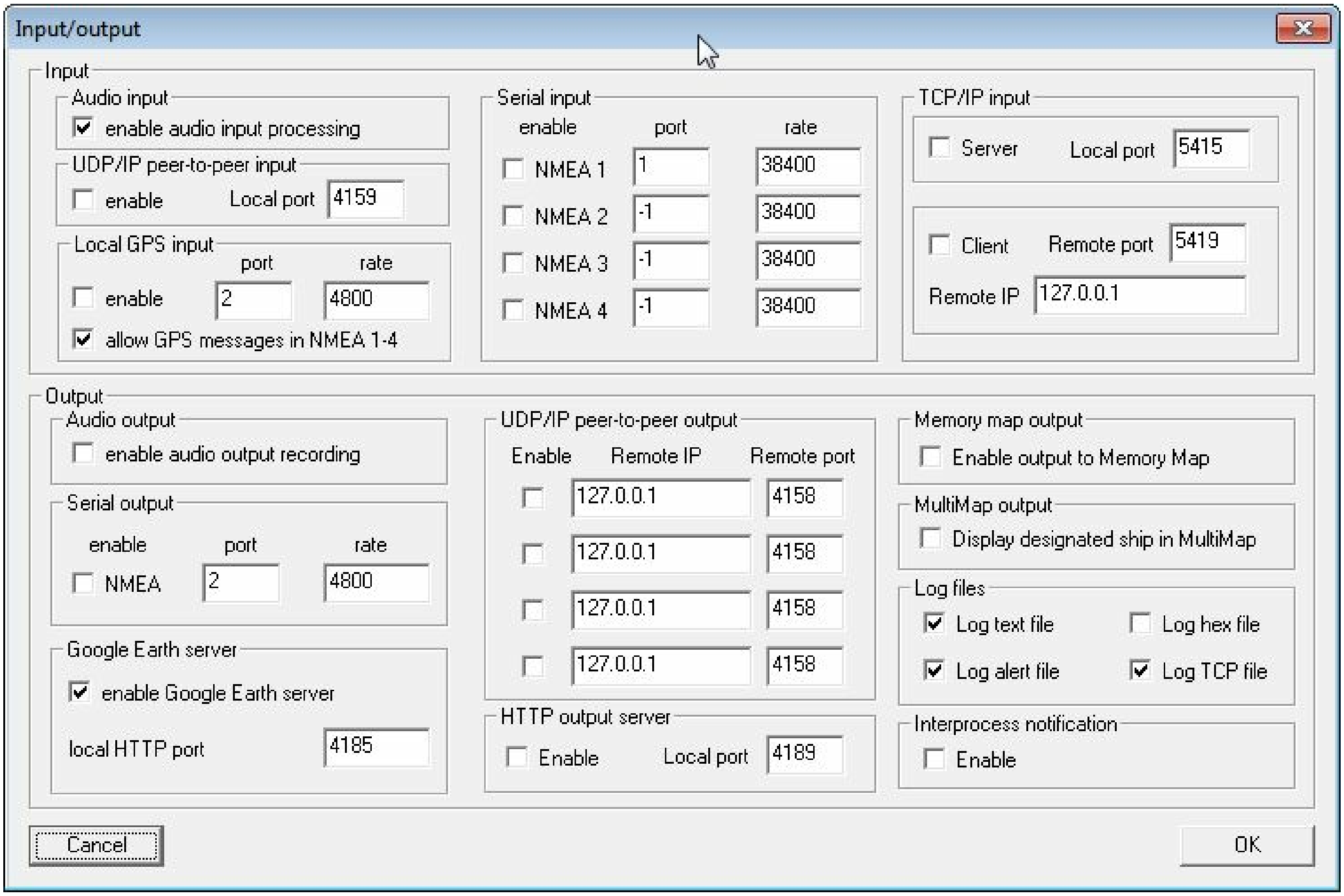
6. Now you should be able to click on the ‘Ships’ and ‘Messages’ icons on the toolbar to see the decoded AIS information. To see ships visually, you will need to follow the ShipPlotter instructions for downloading charts.


ShipPlotter from COAA - Ship view
File View Process Options Help

```
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
```

For Help, press F1 | 51 | ...no designated ship... | 05:48:36

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” is checked. The HTTP port can be left as default.



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

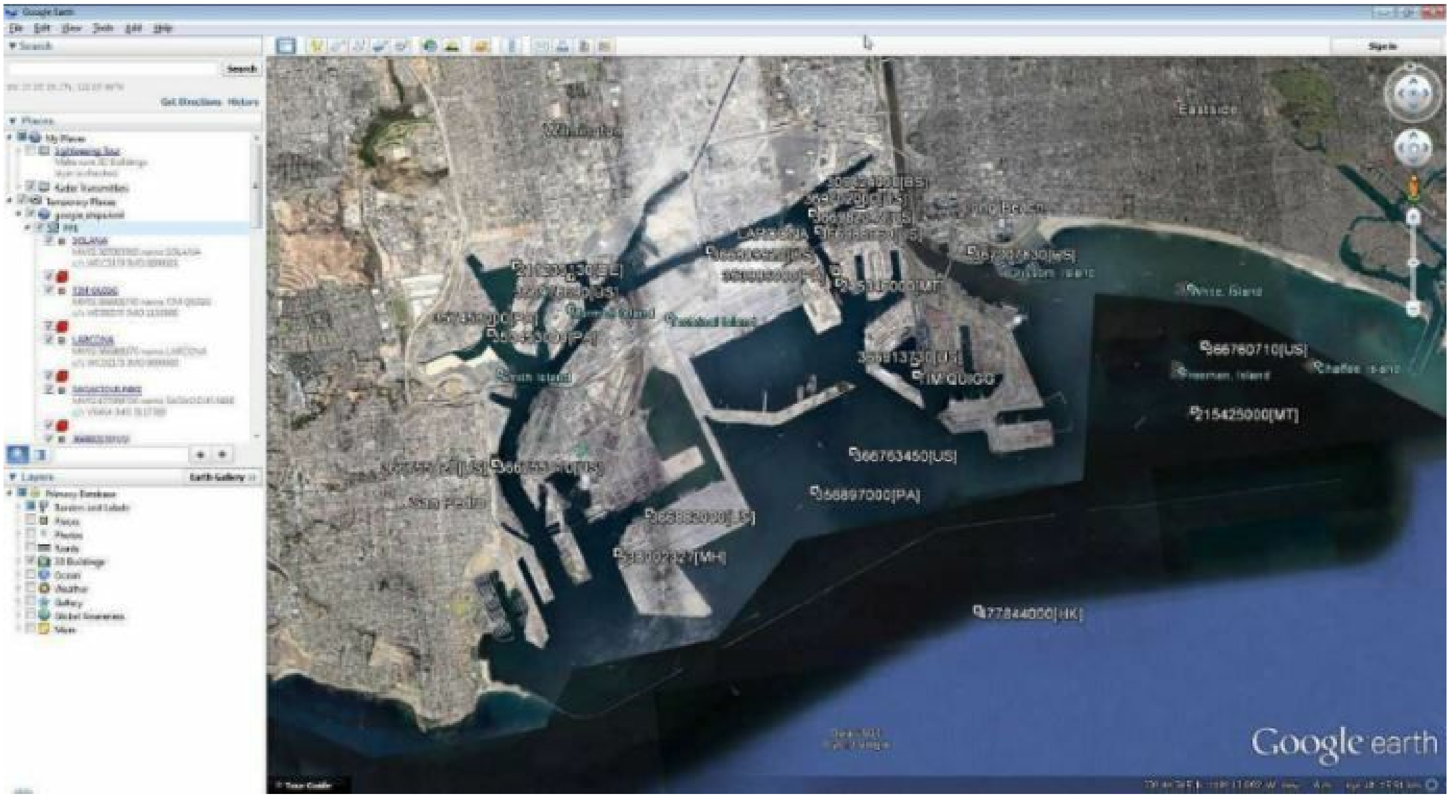
Memory map output
☐ Enable output to Memory Map

MultiMap output
☐ Display designated ship in MultiMap

Log files
☑ Log text file ☐ Log hex file
☑ Log alert file ☑ Log TCP file

Interprocess notification
☐ Enable

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 based AIS decoder which can output NMEA positions for use in OpenCPN. Download AISDecoder from http://forum.aishub.net/ais-decoder/ais-decoder-beta-release/.

Compile the program with

```
tar zxvf aisdecoder.tar.gz
cd aisdecoder
mkdir build
cd build
cmake ../ -DCMAKE_BUILD_TYPE=Release
make
```

Now open two terminal windows. 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.

```
./aisdecoder -h 127.0.0.1 -p 10110 -apulse -c mono -d -l
```

Aisdecoder can be compiled on Windows using MingW. Brief instructions follow. First extract the tar file to a folder on Windows, for example C:\aisdecoder. Open cmake-gui and then set the source code path to C:\aisdecoder. Set the build folder to something like C:\aisdecoder\build. Press Configure, and select MingW32 as the compiler. Once configuring is complete press Generate. Now open a cmd prompt and then browse to C:\aisdecoder\build. Type mingw32-make to compile the program to an exe.

See the steps for building rtl_sdr drivers for a similar more instructive tutorial on compiling with cmake-gui and Mingw.

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.

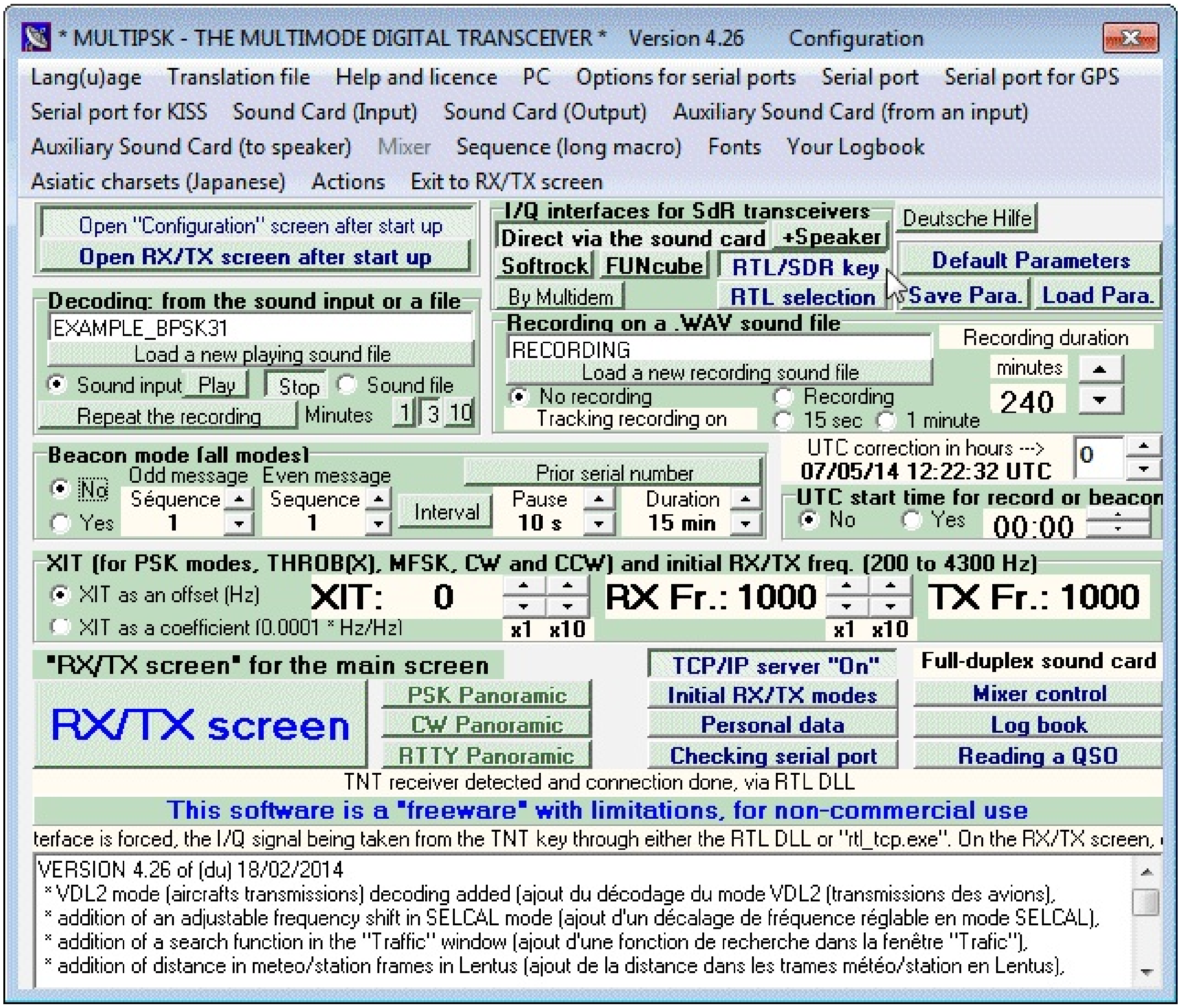
DECODING VDL2 WITH MULTIPSK

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.



TNT receiver detected and connection done, via RTL DLL

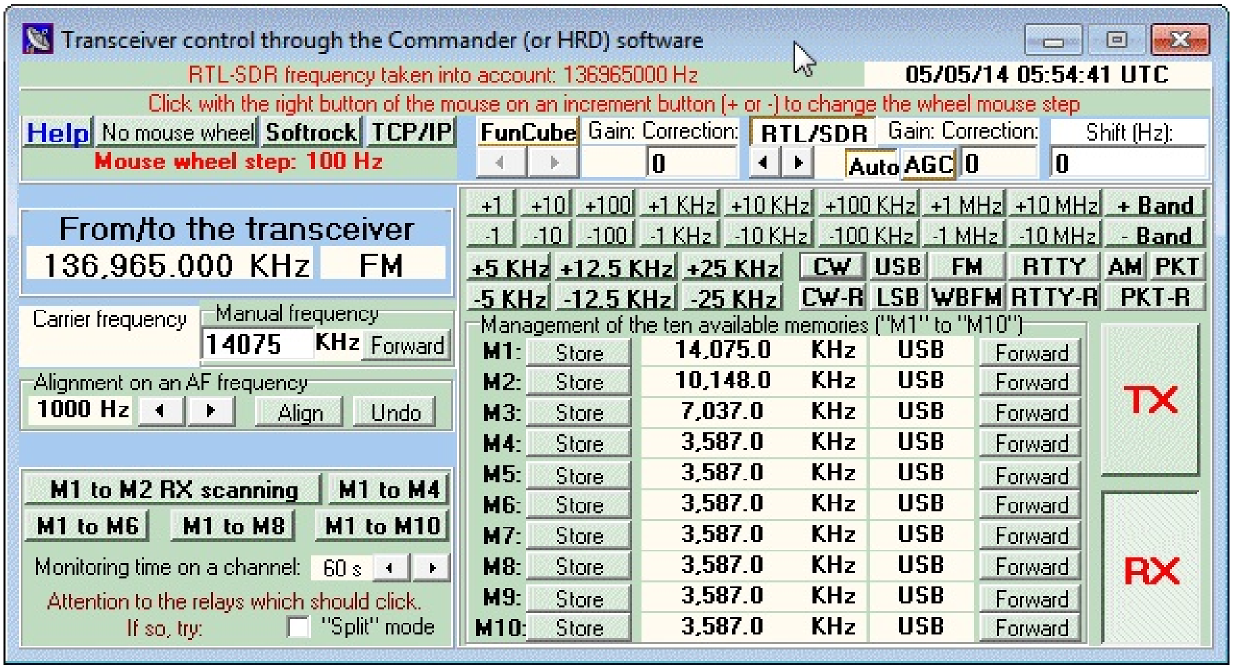
This software is a "freeware" with limitations, for non-commercial use

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

From/to the transceiver

136.965.000 KHz FM

Carrier frequency

Manual frequency: 14075 KHz

Alignment on an AF frequency: 1000 Hz

Monitoring time on a channel: 60 s

Attention to the relays which should click.

If so, try: □ "Split" mode

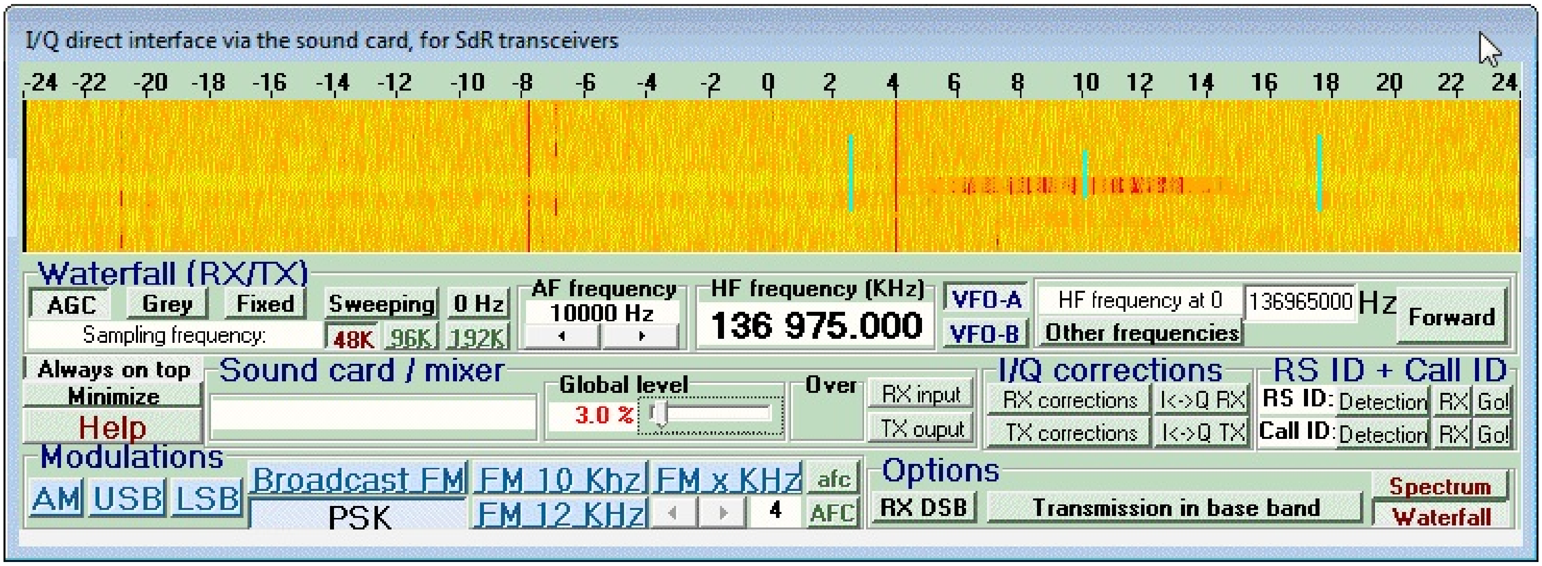
Management of the ten available memories ("M1" to "M10")

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

Help
No mouse wheel
Softrock
TCP/IP
FunCube
Gain: Correction: 0
RTL/SDR
Gain: Correction:
Auto AGC 0
Shift (Hz): 0
Forward
Align
Undo
M1 to M2 RX scanning
M1 to M4
M1 to M6
M1 to M8
M1 to M10
+1 +10 +100 +1 KHz +10 KHz +100 KHz +1 MHz +10 MHz + Band
-1 -10 -100 -1 KHz -10 KHz -100 KHz -1 MHz -10 MHz - Band
+5 KHz +12.5 KHz +25 KHz CW USB FM RTTY AM PKT
-5 KHz -12.5 KHz -25 KHz CW-R LSB WBFM RTTY-R PKT-R
TX
RX

6. In the floating window with the waterfall/spectrum change the HF frequency at 0 to 136965000 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

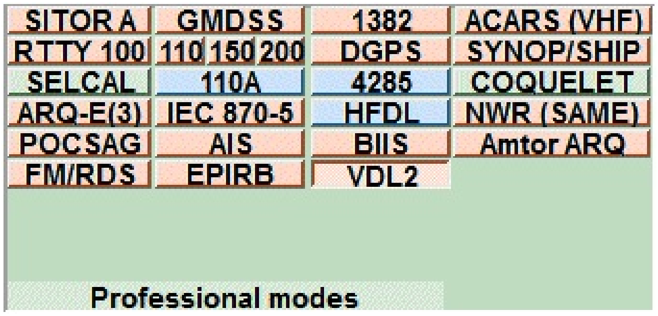
AF frequency: 10000 Hz

HF frequency (KHz): 136 975.000

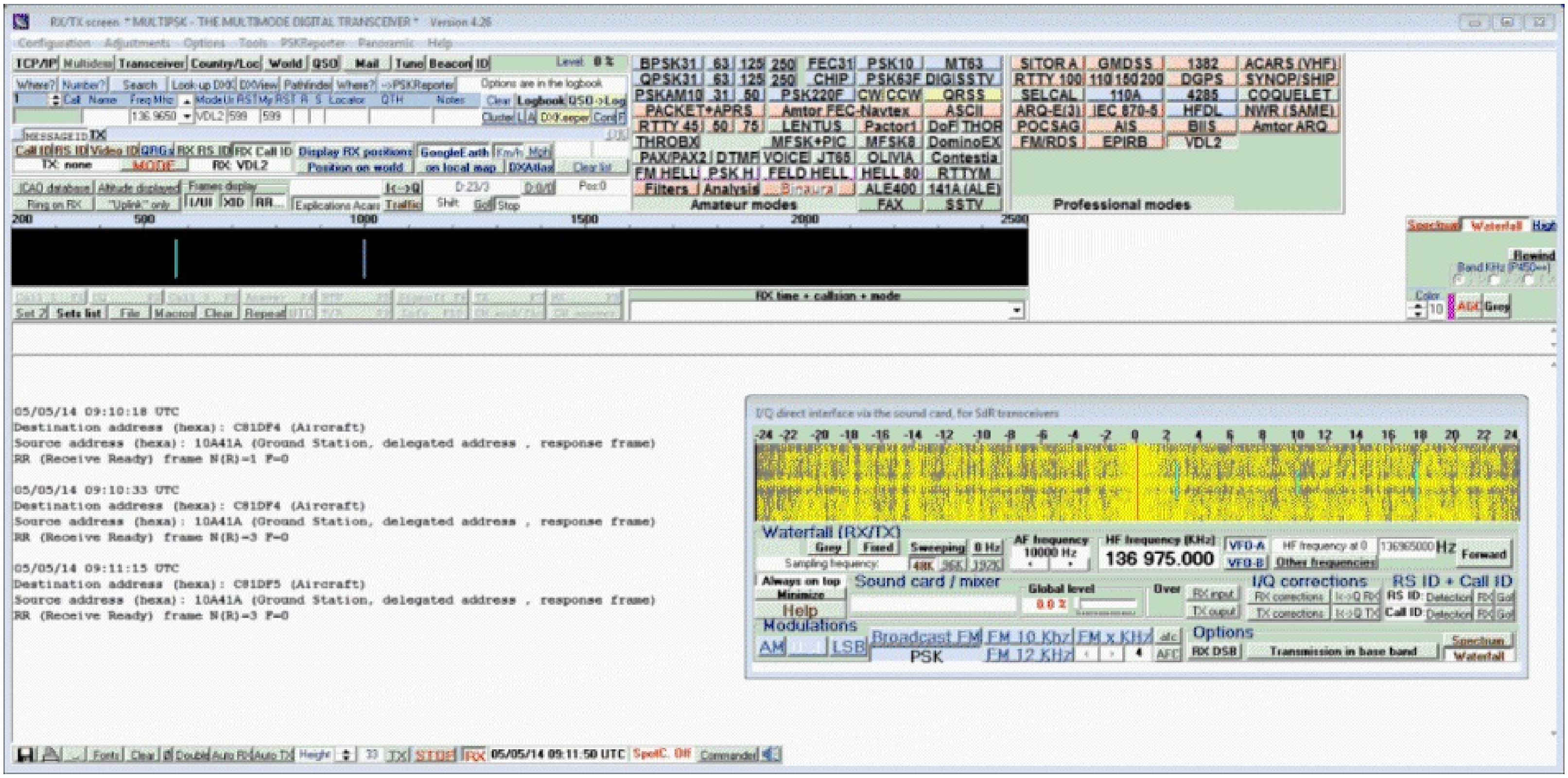
HF frequency at 0: 136965000 Hz

Global level: 3.0 %

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.



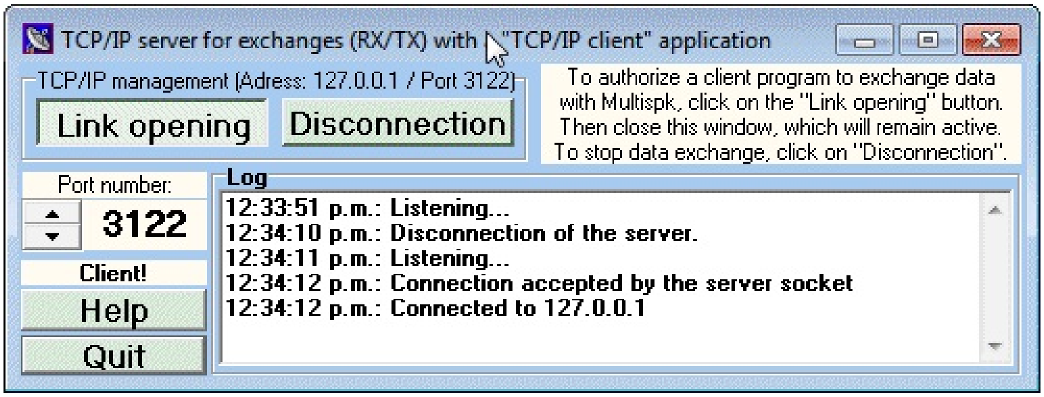
9. Now VDL2 decoding will begin, and you will see VDL2 messages in the main window.


RX/TX screen * MULTIPSK - THE MULTIMODE DIGITAL TRANSCEIVER * Version 4.26
TCP/IP
Amateur modes
Professional modes
VDL2
I/Q direct interface via the sound card, for SdR transceivers
Waterfall (RX/TX)
136 975.000
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. To do this follow these instructions.

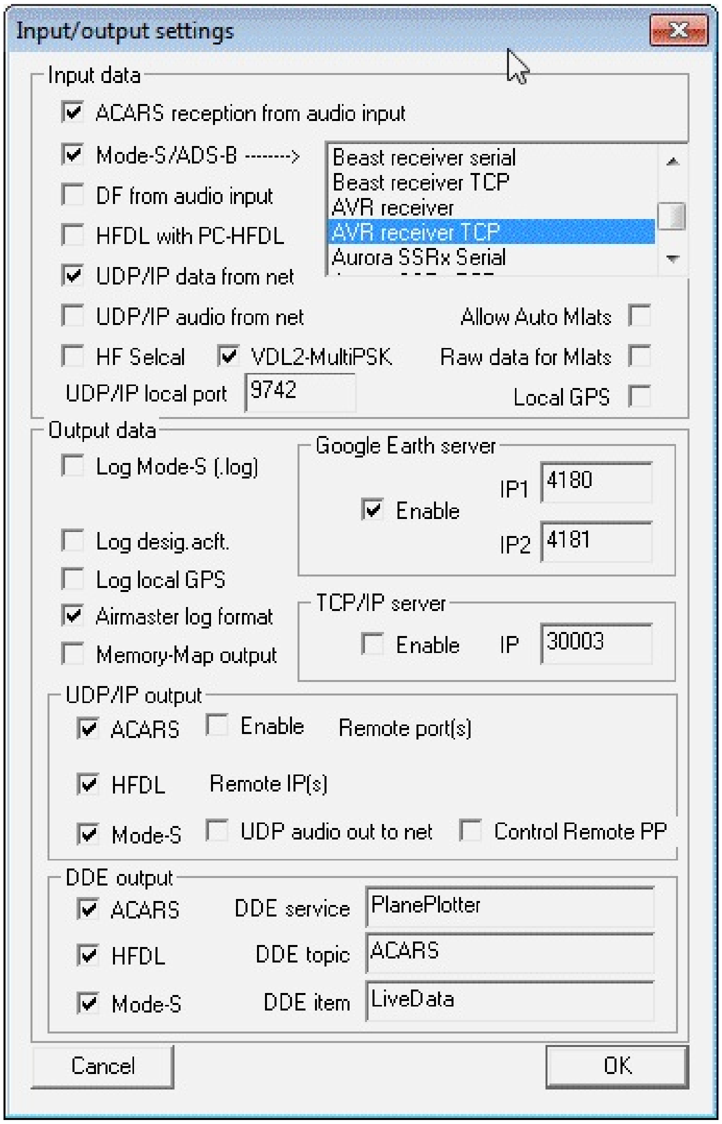
1. In the main MultiPSK window click the TCP/IP button TCP/IP in the top left corner. Make sure the port number is set to 3122.


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 (selected)
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

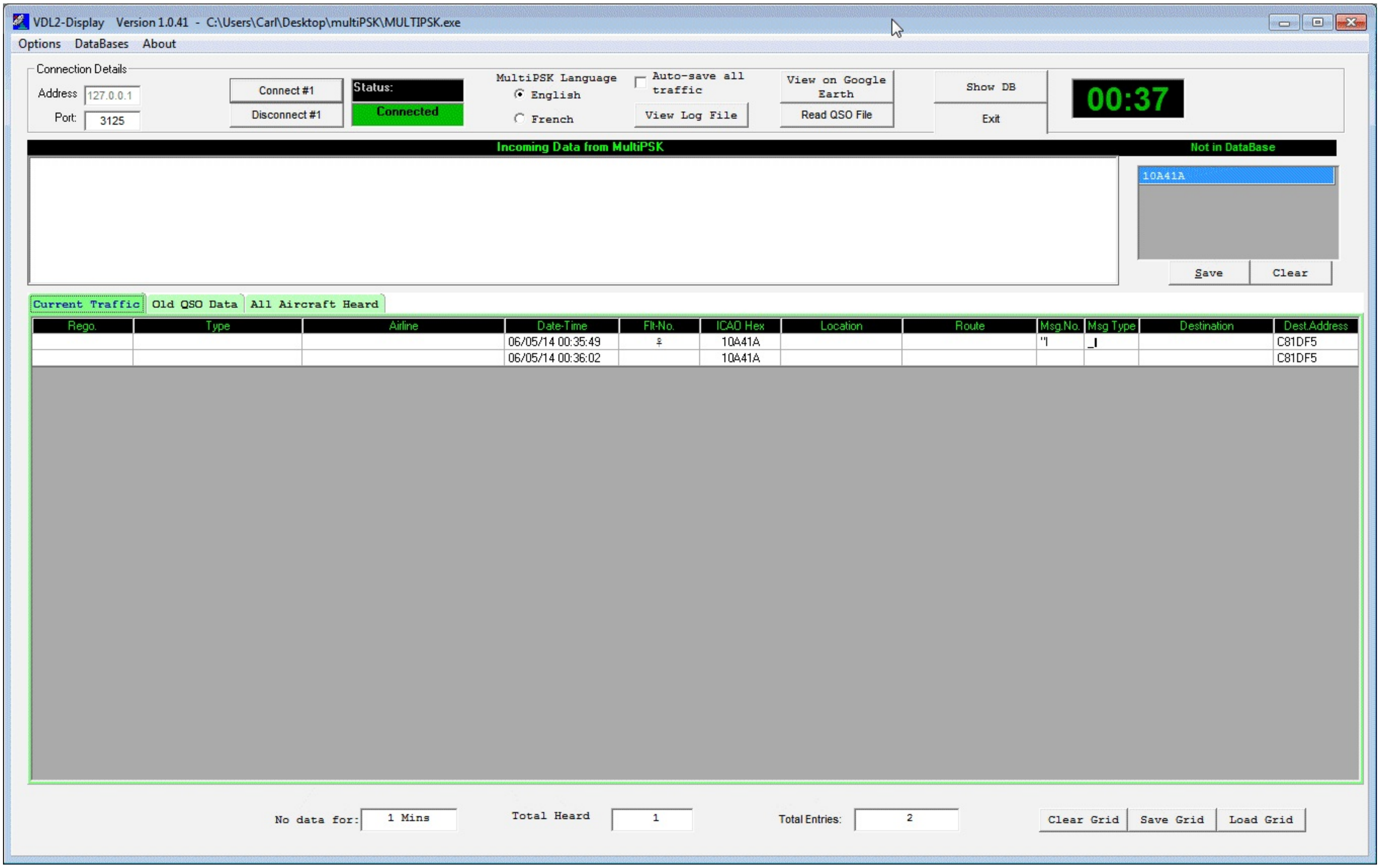
3. Start PlanePlotter by pressing the green start button. Now any messages received by MultiPSK will appear in PlanePlotter as well.

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 more clear 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 port.

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.



Address 127.0.0.1
Port: 3125
Status: Connected
● English
○ French
□ Auto-save all traffic
00:37

10A41A

Rego.	Type	Airline	Date-Time	Flt-No.	ICAO Hex	Location	Route	Msg.No.	Msg Type	Destination	Dest.Address
			06/05/14 00:35:49	♀	10A41A			"I	_I		C81DF5
			06/05/14 00:36:02		10A41A						C81DF5

No data for: 1 Mins
Total Heard 1
Total Entries: 2

KG-VDL

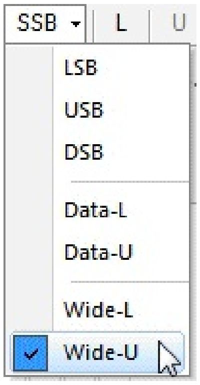
KG-VDL is a VDL2 decoder by a Japanese developer. It is a paid product, but has a 10 minute trial which can be used for testing. It can be downloaded from http://www2.plala.or.jp/hikokibiyori/soft/kgvdl/. Unfortunately, it seems that KG-VDL cannot be purchased anymore and only the trial version is available. To use KG-VDL you must use SDR-RADIO v2 as the audio source as it is the only SDR receiver with a wide USB setting. To decode VDL2 using KG-VDL follow these steps

1. Open KG-VDL and set the input audio device by clicking the in-button ⇥IN to select the audio piping method you prefer.

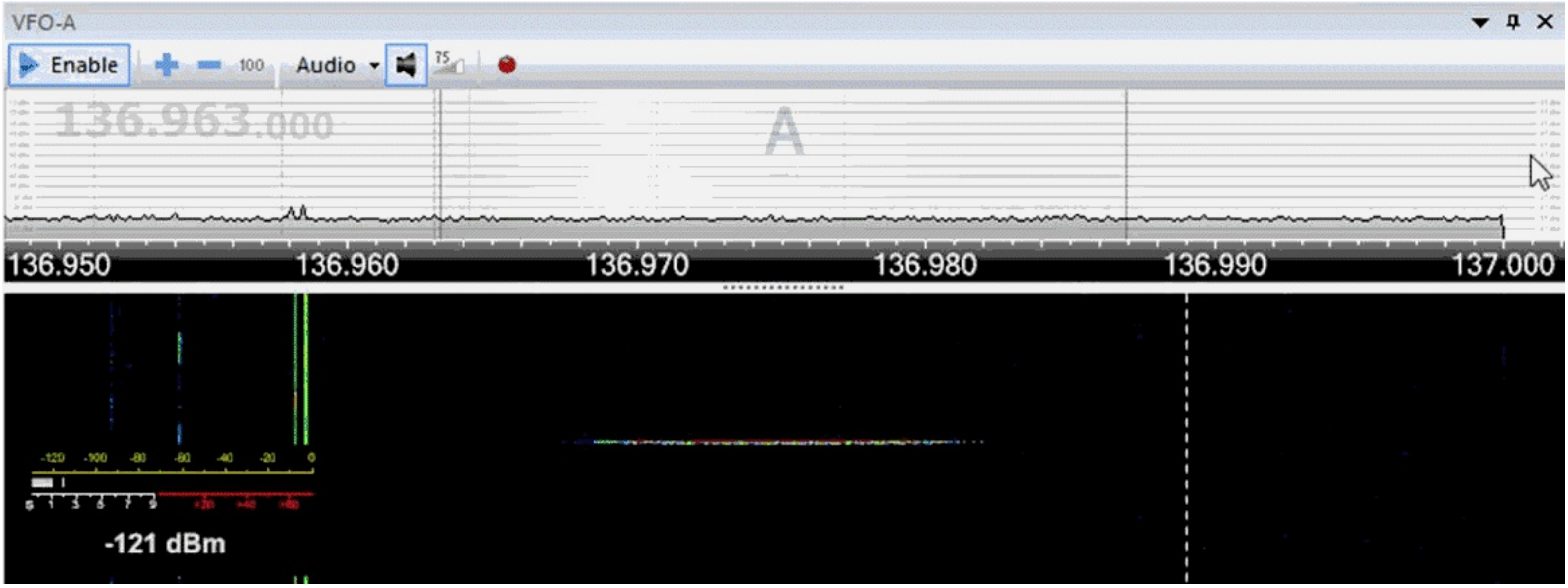
2. Now open SDR-CONSOLE, and select the audio piping method you are using for the output audio.

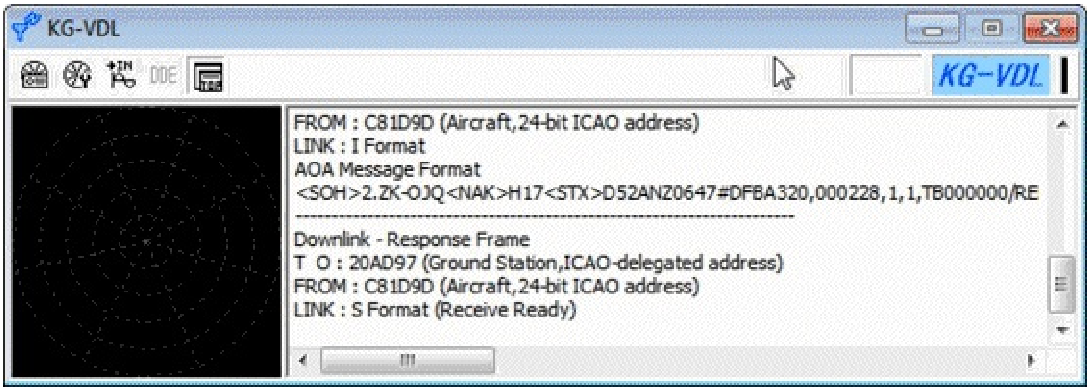
3. Tune to 136.963 MHz.

4. Choose Wide-U as the receive type.



5. Drag the bandwidth to be as large as possible. When a VDL packet is received, the window should look like this.





```
FROM : C81D9D (Aircraft,24-bit ICAO address)
LINK : I Format
AOA Message Format
<SOH>2.ZK-OJQ<NAK>H17<STX>D52ANZ0647#DFBA320,000228,1,1,TB000000/RE
--------------------------------------------------------------------
Downlink - Response Frame
T O : 20AD97 (Ground Station,ICAO-delegated address)
FROM : C81D9D (Aircraft,24-bit ICAO address)
LINK : S Format (Receive Ready)
```

REFERENCES AND ADDITIONAL LINKS

PlanePlotter MultiPSK VDL2 Tutorial:
http://planeplotter.pbworks.com/w/page/75894179/VDL%20mode%202

AIR TRAFFIC CONTROL SCANNER GUIDE

Air traffic control 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 system that requires scanning.

SDR# FREQUENCY SCANNER

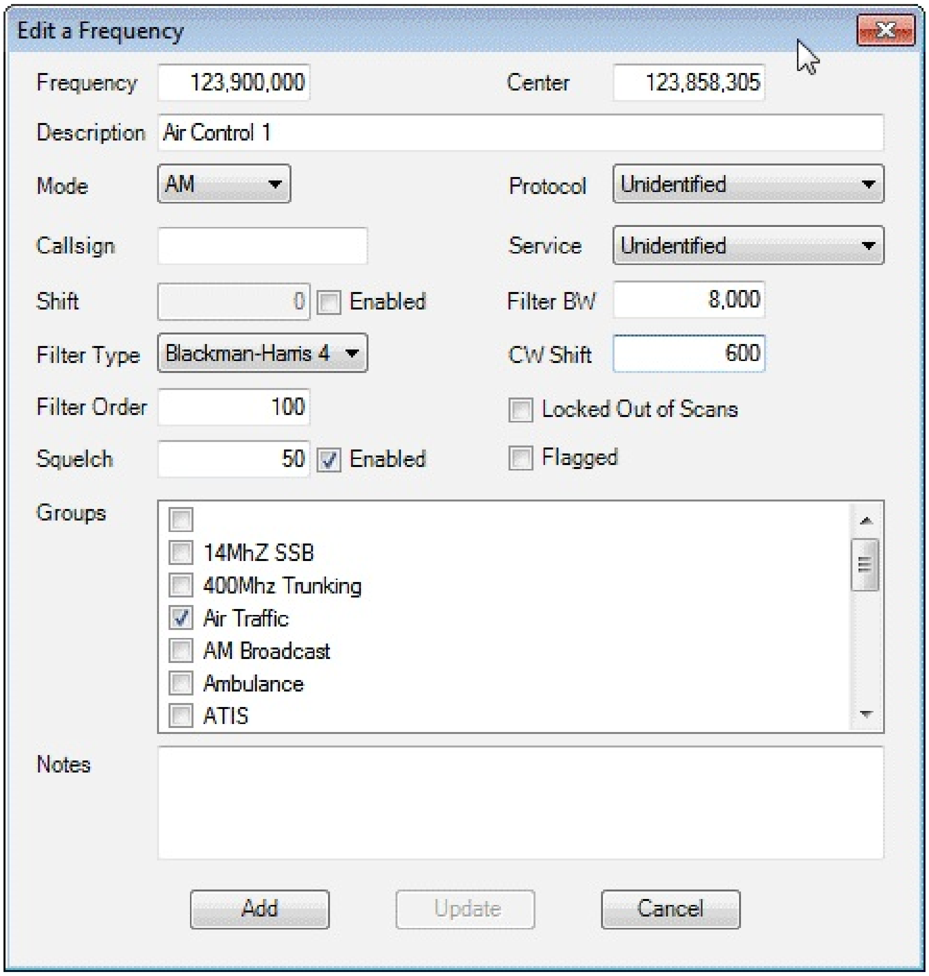
The SDR# frequency scanner plugin we will use can be downloaded from www.sdrsharpplugins.com. It has an automatic installer so there is no need to edit any config files. Simply extract the zip file to a folder and then run PluginsSetup.exe following the on screen instructions.

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 record active frequencies you see 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.



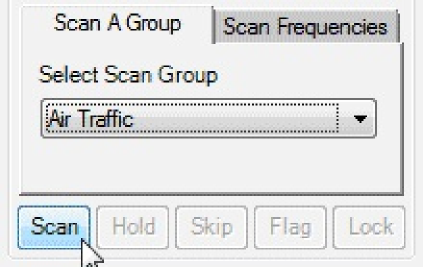
Edit a Frequency

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

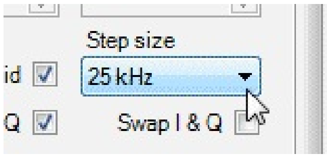
You can edit the scan speed by clicking on the spanner icon 🔧 and then adjusting the “Radio Settle Time in Milliseconds” slider under the Performance tab.

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 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

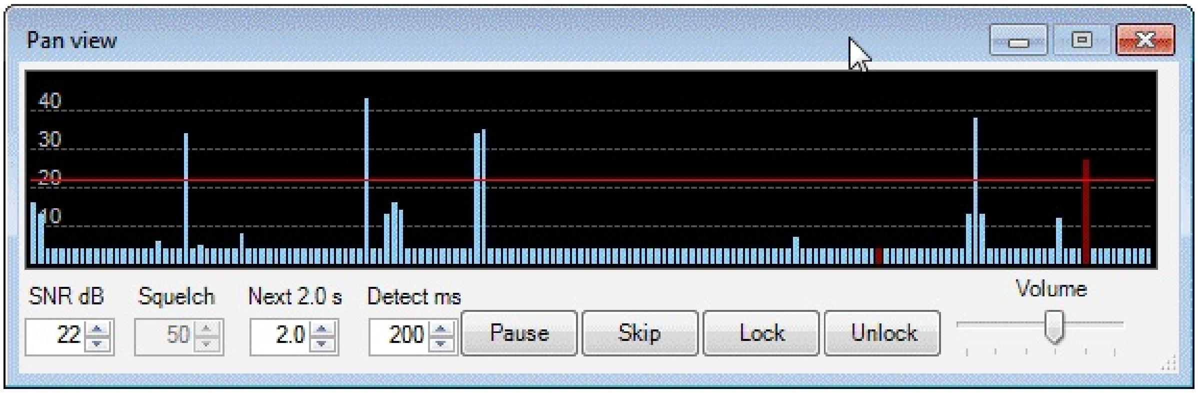
2. Go to the fast scanner frequency plugin window. Choose either "screen" or "manual" mode in the drop down menu. Screen mode will scan for active frequencies that are currently displayed on the screen and manual mode will let you scan between an arbitrary minimum and maximum frequency.

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 scanner frequency spectrum, or by dragging the mouse in the scanner 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.


Pan view

SNR dB: 22
Squelch: 50
Next 2.0 s: 2.0
Detect ms: 200

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.

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 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.

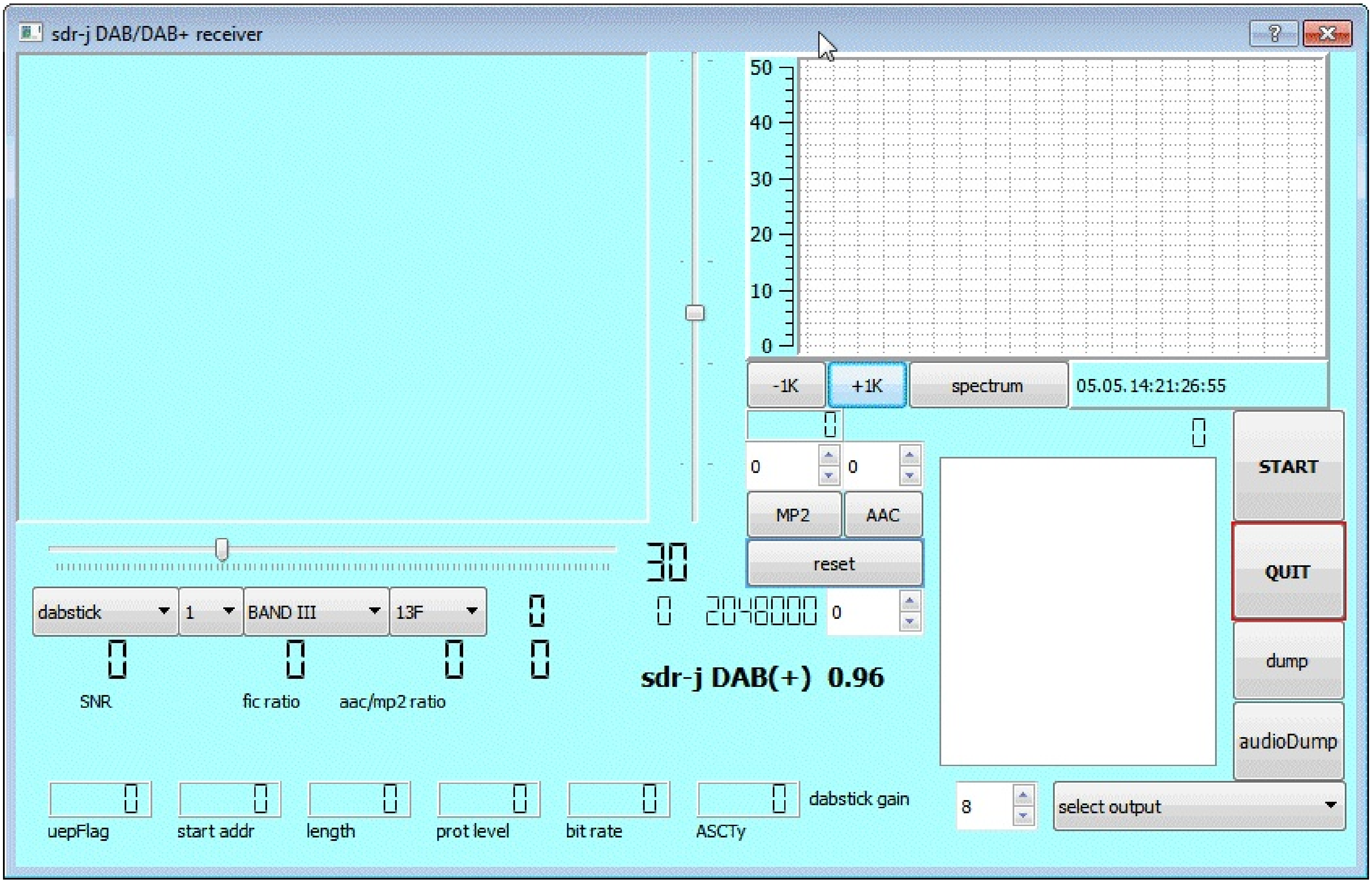
DAB RECEIVE TUTORIAL

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. It can be downloaded from http://www.sdr-j.tk/ making sure to download the dabstick-radio zip file.

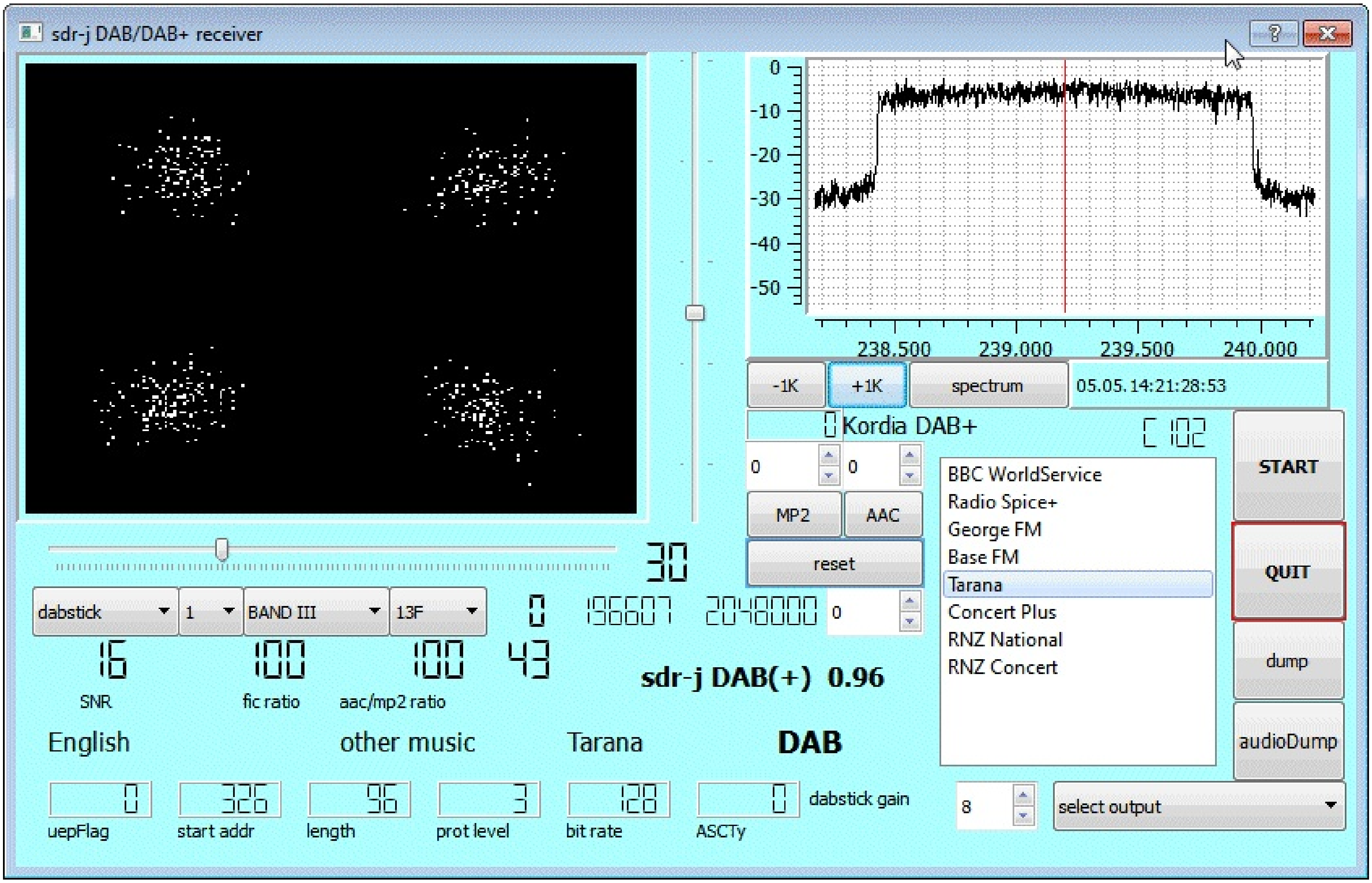
1. Download SDR-J and extract the zip file to a folder.

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.



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 spectrum 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.
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.



05.05.14:21:28:53

Kordia DAB+ C 102

dabstick | 1 | BAND III | 13F

30 0 196607 2048000 0

SNR 16 · fic ratio 100 · aac/mp2 ratio 100 · 43

English · other music · Tarana · DAB

uepFlag 0 · start addr 326 · length 96 · prot level 3 · bit rate 128 · ASCTy 0

dabstick gain 8 · select output

BBC WorldService
Radio Spice+
George FM
Base FM
Tarana
Concert Plus
RNZ National
RNZ Concert

An alternative DAB player that uses the official RTL-SDR drivers can be downloaded from http://www.ukwtv.de/cms/downloads-aside/281-dab-player-von-andreas-gsinn.html. Note that the language of this program is in German. To use this program you will need to install the provided drivers first (for the R820T download Treiber2.zip) which are different to the SDR drivers.

RECEIVING ANALOGUE TV (PAL/NTSC) GUIDE

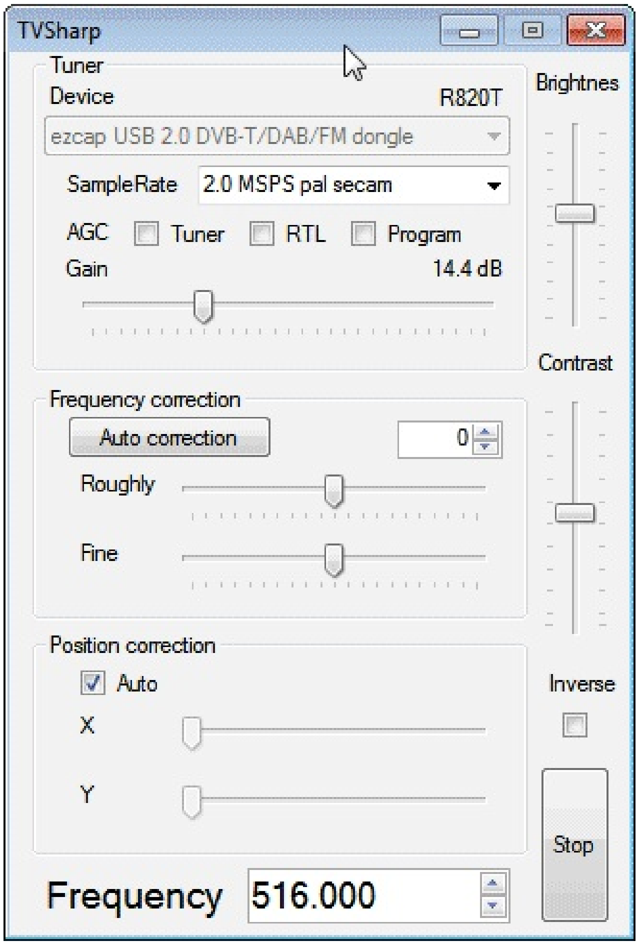
Most countries have begun phasing out transmission of Analogue TV instead favouring the switch to Digital transmission standards like ATSC (USA) or DVB-T (most of world). In the countries that still transmit Analogue TV the RTL-SDR can be used to receive it.

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 PAL or NTSC.

A program which does this 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 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. Then tune to a Analogue TV frequency. Using a higher sample rate will increase the bandwidth sampled, improving picture quality. But setting it too high can seems to cause reception problems. You can then use the frequency text box to tune to a known TV station.

The program also has automatic frequency and position correction options built in which will help to prevent image scrolling and non centred images.



TVSharp

Tuner

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

Frequency correction

0

Position correction

☑ Auto

Inverse □

Frequency 516.000

GUIDE TO LISTENING TO TRUNKED RADIO: ANALOGUE AND DIGITAL P25

In an inefficient radio system, one company (or talkgroup) 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. In a trunked radio system, a small set number of frequencies are shared between a large number of talkgroups. Each radio receives a special computer controlled control channel. The control channel determines a vacant frequency that a particular talkgroup should use on the fly. This helps to make radio frequency allocations more efficient.

Because a talkgroup might switch between various frequencies often, it can make listening to a conversation difficult for radio scanners that cannot decode the control channel. The 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 with just Unitrunker. 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

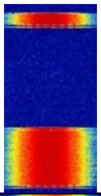
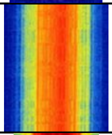
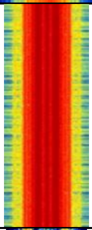
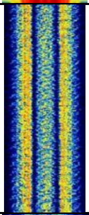
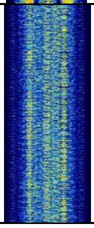
- 2x R820T RTL2832U RTL-SDR dongles. See the Buy RTL-SDR dongles page at http://www.rtl-sdr.com/buy-rtl-sdr-dvb-t-dongles/ if you don’t already have two. You will need one dongle to receive the control channel and one dongle to receive the voice channel. Only the R820T appears to be supported at the moment.
- The Unitrunker software (Download the latest version from http://www.unitrunker.com/download)
- One antenna that splits into two connections for the two dongles, or two 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

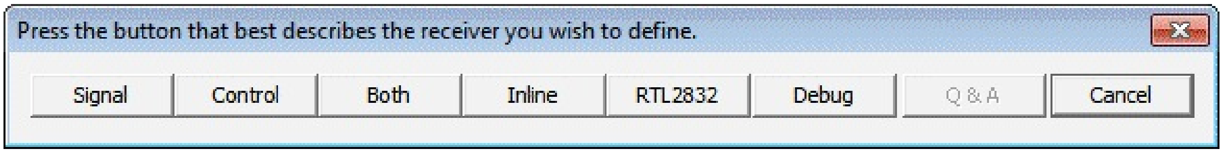
First determine the frequency of a trunking control channel that you are interested in monitoring. You can use SDR# or a similar program to search for these first. They will be signals that are continuously transmitting with no breaks. Here are some examples of some common trunking modes. The are commonly found in the 400 and 800 MHz bands.

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.

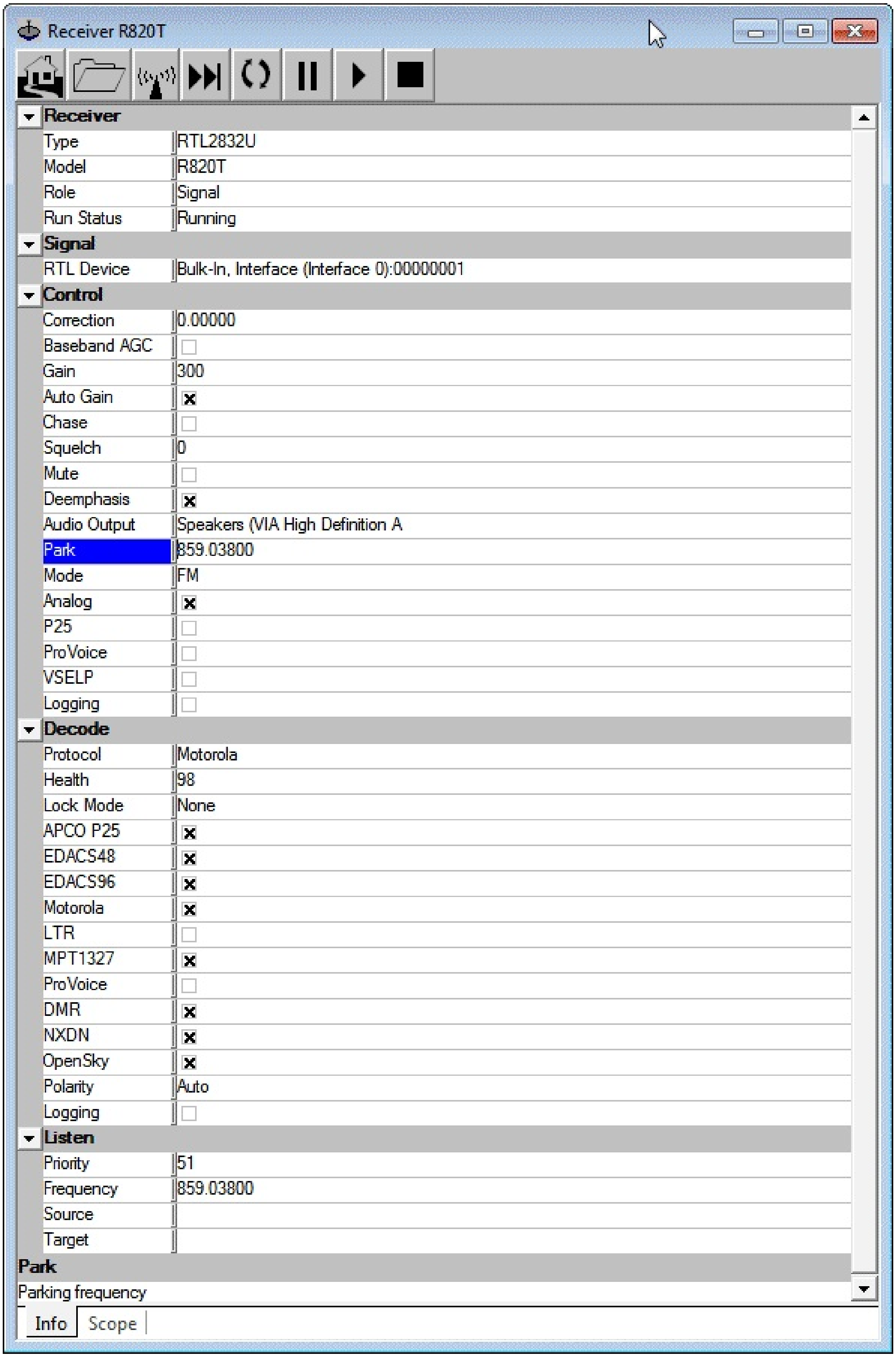


Press the button that best describes the receiver you wish to define.

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 offset in PPM in Correction.
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 ▶. A new window should soon pop up and the value of the Protocol setting under the Decode heading should change to the trunking protocol detected.

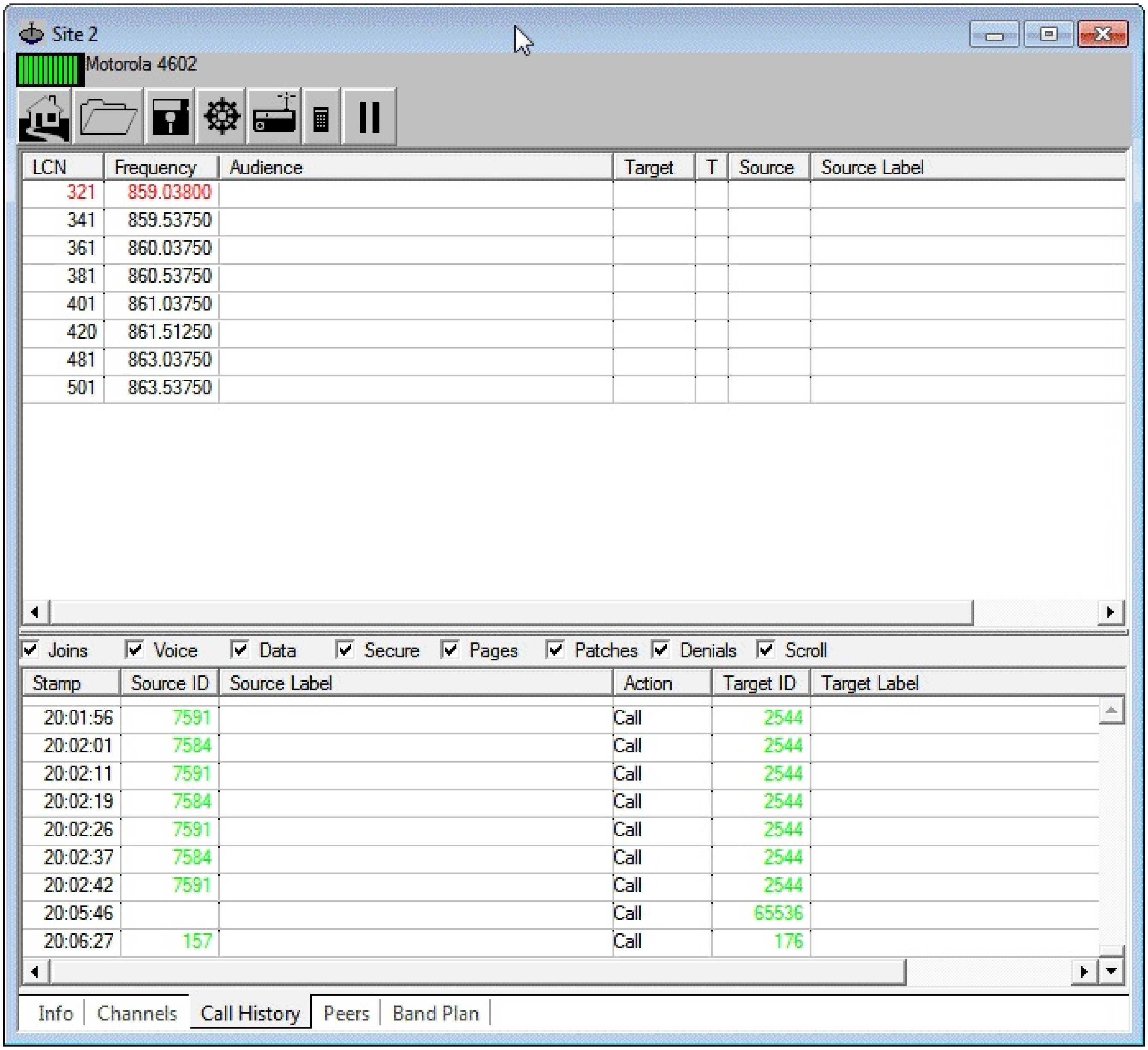
10. If the window does not pop up, either increase the Gain value or turn on the Auto Gain setting. If no window still pops up, set the Audio Output setting to your speakers and uncheck the mute button. Listen to see if the trunking channel is being correctly played through your speakers. If not, you may need to fine tune the frequency.



Receiver	
Type	RTL2832U
Model	R820T
Role	Signal
Run Status	Running
Signal	
RTL Device	Bulk-In, Interface (Interface 0):00000001
Control	
Correction	0.00000
Baseband AGC	☐
Gain	300
Auto Gain	☒
Chase	☐
Squelch	0
Mute	☐
Deemphasis	☒
Audio Output	Speakers (VIA High Definition A
Park	859.03800
Mode	FM
Analog	☒
P25	☐
ProVoice	☐
VSELP	☐
Logging	☐
Decode	
Protocol	Motorola
Health	98
Lock Mode	None
APCO P25	☒
EDACS48	☒
EDACS96	☒
Motorola	☒
LTR	☐
MPT1327	☒
ProVoice	☐
DMR	☒
NXDN	☒
OpenSky	☒
Polarity	Auto
Logging	☐
Listen	
Priority	51
Frequency	859.03800
Source	
Target	

Park
Parking frequency

11. If the signal is strong enough, a new 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

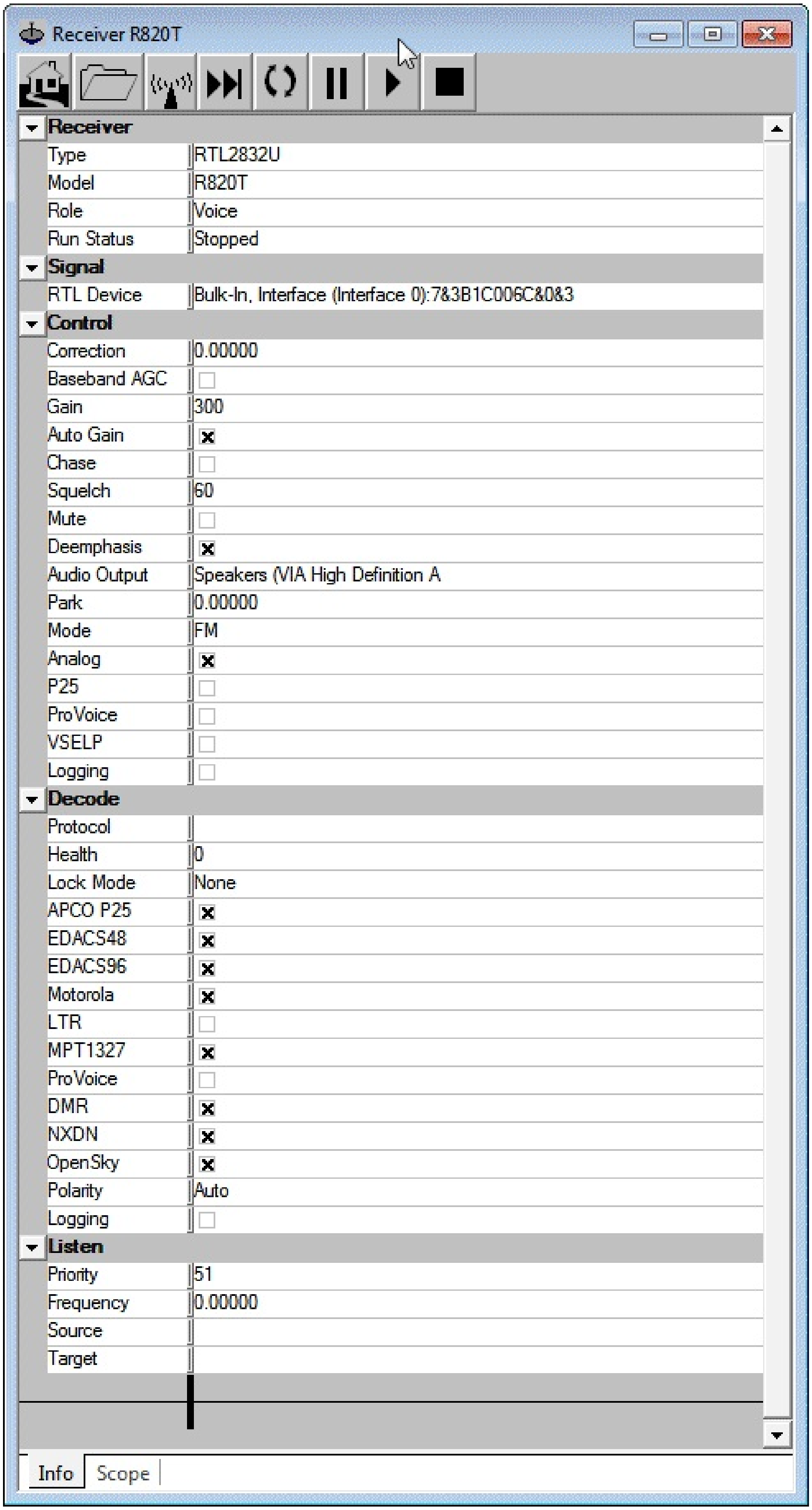
Stamp	Source ID	Source Label	Action	Target ID	Target Label
20:01:56	7591		Call	2544	
20:02:01	7584		Call	2544	
20:02:11	7591		Call	2544	
20:02:19	7584		Call	2544	
20:02:26	7591		Call	2544	
20:02:37	7584		Call	2544	
20:02:42	7591		Call	2544	
20:05:46			Call	65536	
20:06:27	157		Call	176	

12. 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.
6. Choose the Audio Output as your speakers and make sure that Mute is *not* checked.
7. Set the ‘Park’ frequency to zero.

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 Status	Stopped
Signal	
RTL Device	Bulk-In, Interface (Interface 0):7&3B1C006C&0&3
Control	
Correction	0.00000
Baseband AGC	□
Gain	300
Auto Gain	☒
Chase	□
Squelch	60
Mute	□
Deemphasis	☒
Audio Output	Speakers (VIA High Definition A
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	51
Frequency	0.00000
Source	
Target	

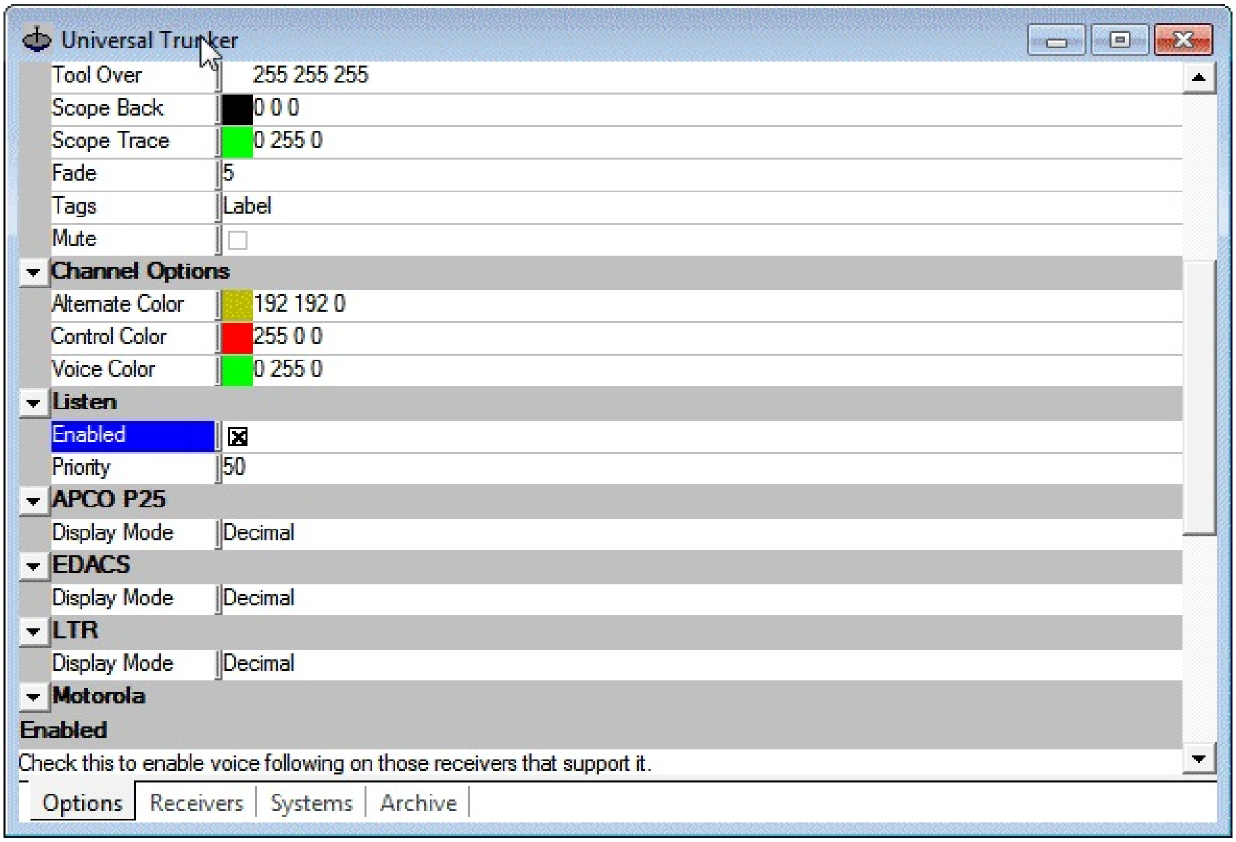
9. Press the Play button ▶.

ENABLING UNITRUNKER FOLLOWING

1. Now we will make sure that Unitrunker is set to allow the voice receiver to follow calls from the signal channel.

2. Go back to the main Unitrunker window and click on the "Options" tab at the bottom of the window.

3. 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!

DECODING DIGITAL VOICE (P25/DMR/MOTOTRBO/NXDN/PROVOICE)

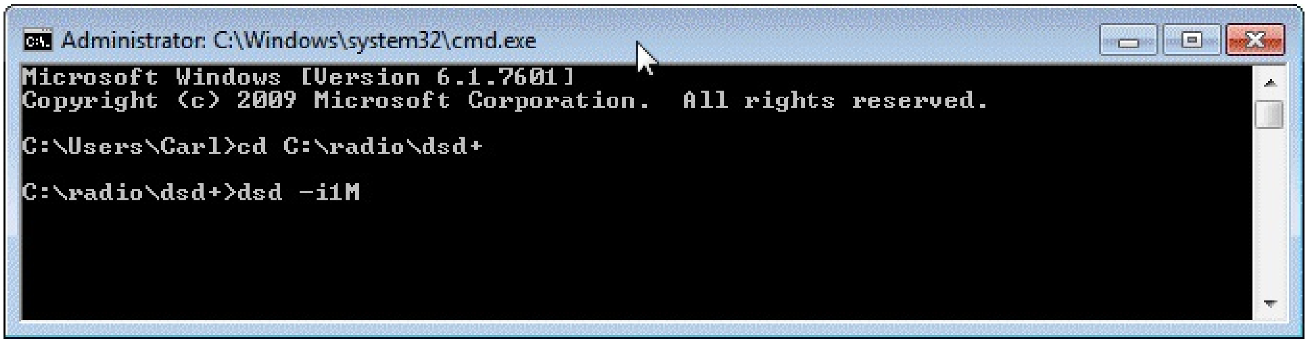
Decoding of unencrypted digital voice signals like P25/DMR/Mototrbo can be set up in the same way as the analogue system shown above. However, instead of outputting audio to the speakers, audio should be output to a virtual audio pipe such as VAC or VB Cable. Then DSD+ should be used to listen to the audio pipe. See information in Appendix A: Audio Piping for more information if you don't have any audio pipes set up.

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 DSD version 1.6, but it has vastly improved decoding in terms of voice quality. You can find the download for DSD+ here http://www.rtl-sdr.com/improved-digital-voice-p25-decoding-dsd/.

To run DSD+. you will need to place an MP3 encoder file lame_enc.dll into the same folder as the dsd.exe executable. This file is not included with DSD+ due to licensing. For Windows, lame_enc.dll can be downloaded from http://lame1.buanzo.com.ar/#lamewindl (Mirror: http://bit.ly/1subqre). Download the ZIP option, and then copy the lame_enc.dll file into the same folder as DSD+.

To run DSD+ open a command prompt from **Start->All Programs->Accessories->Command Prompt**, and navigate to the folder where DSD+ is located. For those who are not familiar with the command prompt, use the change directory 'cd' command to change directories within command prompt. For example, if your DSD+ folder is located in c:\Radio\dsd+, you would type in command prompt 'cd c:\Radio\dsd+'.

Type 'dir' to see a list of all files in the current directory. Next in command prompt type 'dsd -i1M', where the number after the -i is the number of your audio pipe. If your audio pipe is set as your default sound device in the Windows audio recording properties it will be device 1, otherwise it may be device 2 or higher. You can use 'dsd -h' to see more options for DSD+.



```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\Carl>cd C:\radio\dsd+

C:\radio\dsd+>dsd -i1M
```

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 when a digital voice signal is active. The text 'voice' will show on the right when a voice signal is playing. To see the help screen and get more options for DSD+, type 'dsd -h' into terminal. Some signals may require you to set the decoder options described in the help.

To stop dsd running at any time press "Ctrl + C".

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, 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.

TUNING 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.

DSDTune 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.

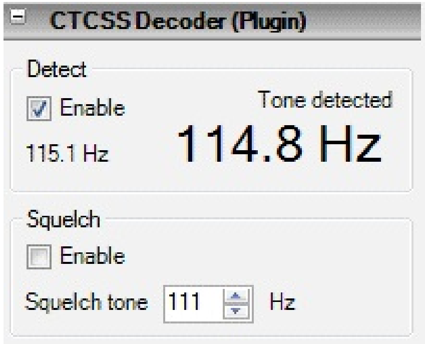
To compile DSDTune, download or git clone it from https://github.com/dreinhold/dsdtune. In command prompt run `gcc -Wall -o dsdtune.exe utils.c dsdtune.c` assuming you have MingW installed. This will generate dsdtune.exe.

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 discriminator 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).

CTCSS TRUNKED RADIO

CTCSS is an acronym for **C**ontinuous **T**one-**C**oded **S**quelch **S**ystem and is a 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 to identify each group of users, and 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. 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

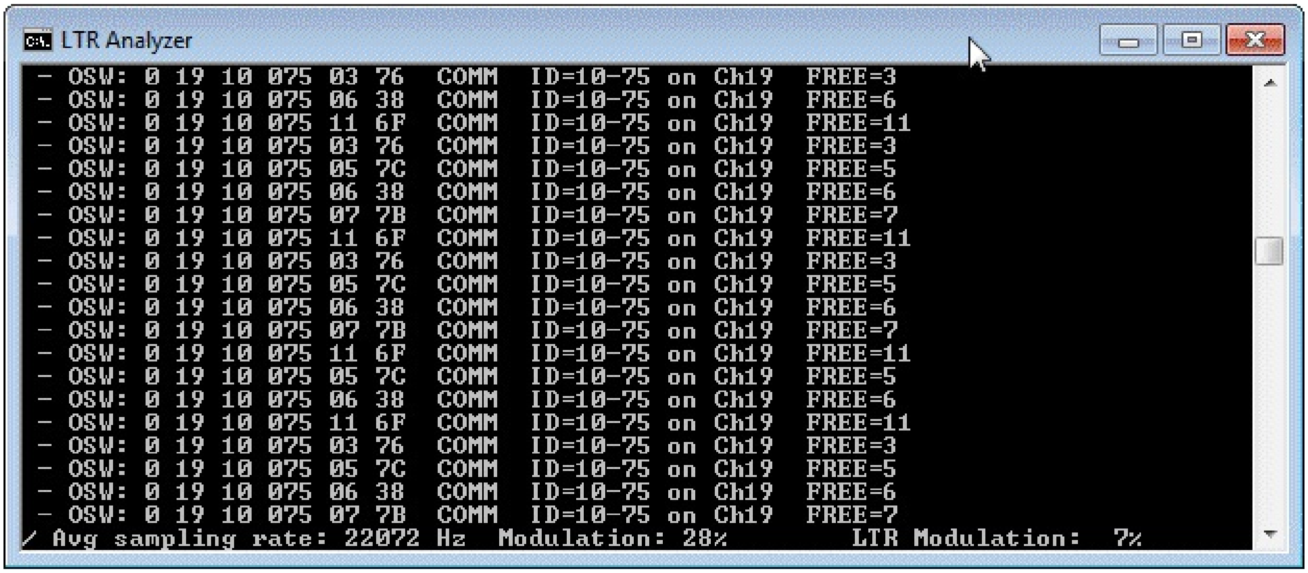
Squelch tone 111 Hz

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 decoded 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. The program will not do much more than showing information about the LTR signal however.

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. When listening to LTR audio, the output will be similar to the example screenshot shown below.



```
- OSW: 0 19 10 075 03 76   COMM   ID=10-75 on Ch19   FREE=3
- OSW: 0 19 10 075 06 38   COMM   ID=10-75 on Ch19   FREE=6
- OSW: 0 19 10 075 11 6F   COMM   ID=10-75 on Ch19   FREE=11
- OSW: 0 19 10 075 03 76   COMM   ID=10-75 on Ch19   FREE=3
- OSW: 0 19 10 075 05 7C   COMM   ID=10-75 on Ch19   FREE=5
- OSW: 0 19 10 075 06 38   COMM   ID=10-75 on Ch19   FREE=6
- OSW: 0 19 10 075 07 7B   COMM   ID=10-75 on Ch19   FREE=7
- OSW: 0 19 10 075 11 6F   COMM   ID=10-75 on Ch19   FREE=11
- OSW: 0 19 10 075 03 76   COMM   ID=10-75 on Ch19   FREE=3
- OSW: 0 19 10 075 05 7C   COMM   ID=10-75 on Ch19   FREE=5
- OSW: 0 19 10 075 06 38   COMM   ID=10-75 on Ch19   FREE=6
- OSW: 0 19 10 075 07 7B   COMM   ID=10-75 on Ch19   FREE=7
- OSW: 0 19 10 075 11 6F   COMM   ID=10-75 on Ch19   FREE=11
- OSW: 0 19 10 075 05 7C   COMM   ID=10-75 on Ch19   FREE=5
- OSW: 0 19 10 075 06 38   COMM   ID=10-75 on Ch19   FREE=6
- OSW: 0 19 10 075 11 6F   COMM   ID=10-75 on Ch19   FREE=11
- OSW: 0 19 10 075 03 76   COMM   ID=10-75 on Ch19   FREE=3
- OSW: 0 19 10 075 05 7C   COMM   ID=10-75 on Ch19   FREE=5
- OSW: 0 19 10 075 06 38   COMM   ID=10-75 on Ch19   FREE=6
- OSW: 0 19 10 075 07 7B   COMM   ID=10-75 on Ch19   FREE=7
/ Avg sampling rate: 22072 Hz  Modulation: 28%       LTR Modulation:  7%
```

PAGER DECODING GUIDE

The rtl-sdr combined with SDRSharp and a POCSAG/Flex capable decoding 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 used more, but pagers are still popular with doctors, some fire and ambulance agencies and various service companies. Pagers tend to be more reliable and have greater coverage.

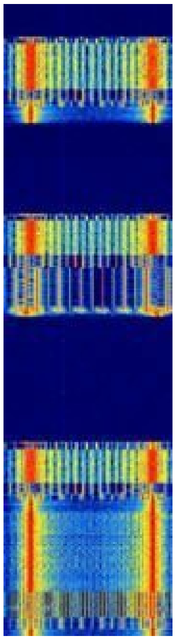
In most countries it is perfectly legal to receive messages from pagers, as they are plain text unencrypted. BUT it is illegal to act on the information received. Please respect your local laws.

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, 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 signals waterfall will look something like the signal shown below. They also have a distinctive sound on NFM. For a sound example see the signal identification guide on rtl-sdr.com.



For this tutorial, you will need to have an rtl-sdr dongle set up and working with SDRSharp or another SDR receiver program. We will assume you have this much done already. If you do 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.

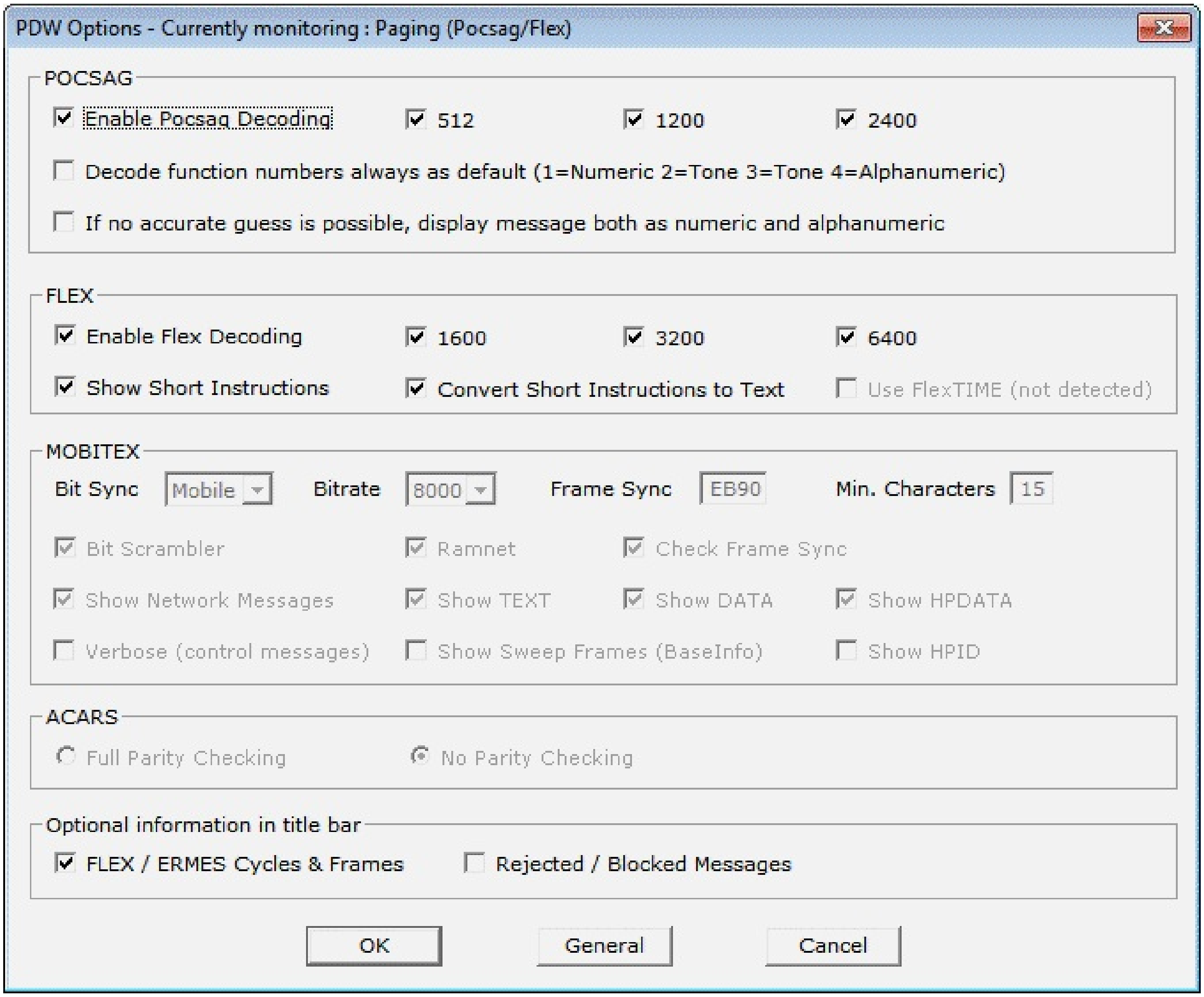
You will also need to have an audio piping method installed and set up. Audio piping will allow the audio from SDRSharp to be passed to a decoding program. Either stereo mix, or Virtual Audio Cable, or VBCable may can be used. See Appendix A: Audio Piping if you need more information. Also before starting ensure you audio piping method is set to a sample rate of 44100 Hz. See the setting the 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 and set the audio piping method to the one you will use 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.

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

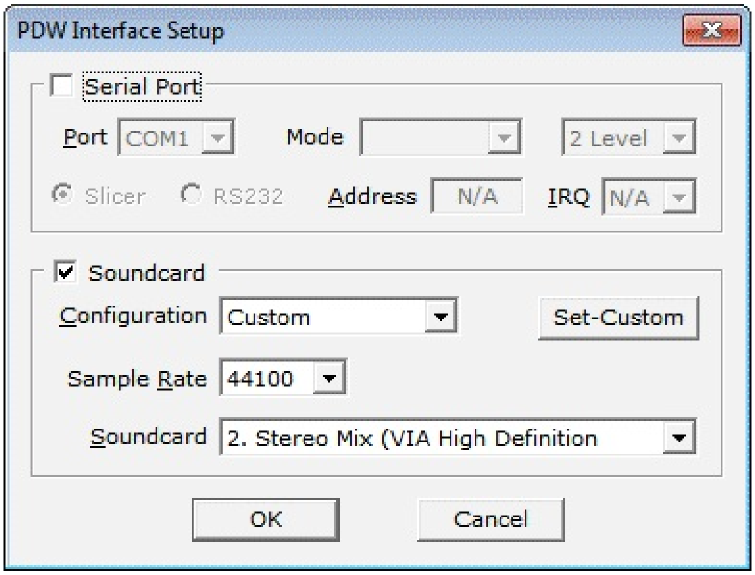
ACARS

○ 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.



PDW Interface Setup

□ Serial Port

Port: COM1; Mode: ; 2 Level

● Slicer ○ RS232; Address: N/A; IRQ: N/A

☑ Soundcard

Configuration: Custom

Sample Rate: 44100

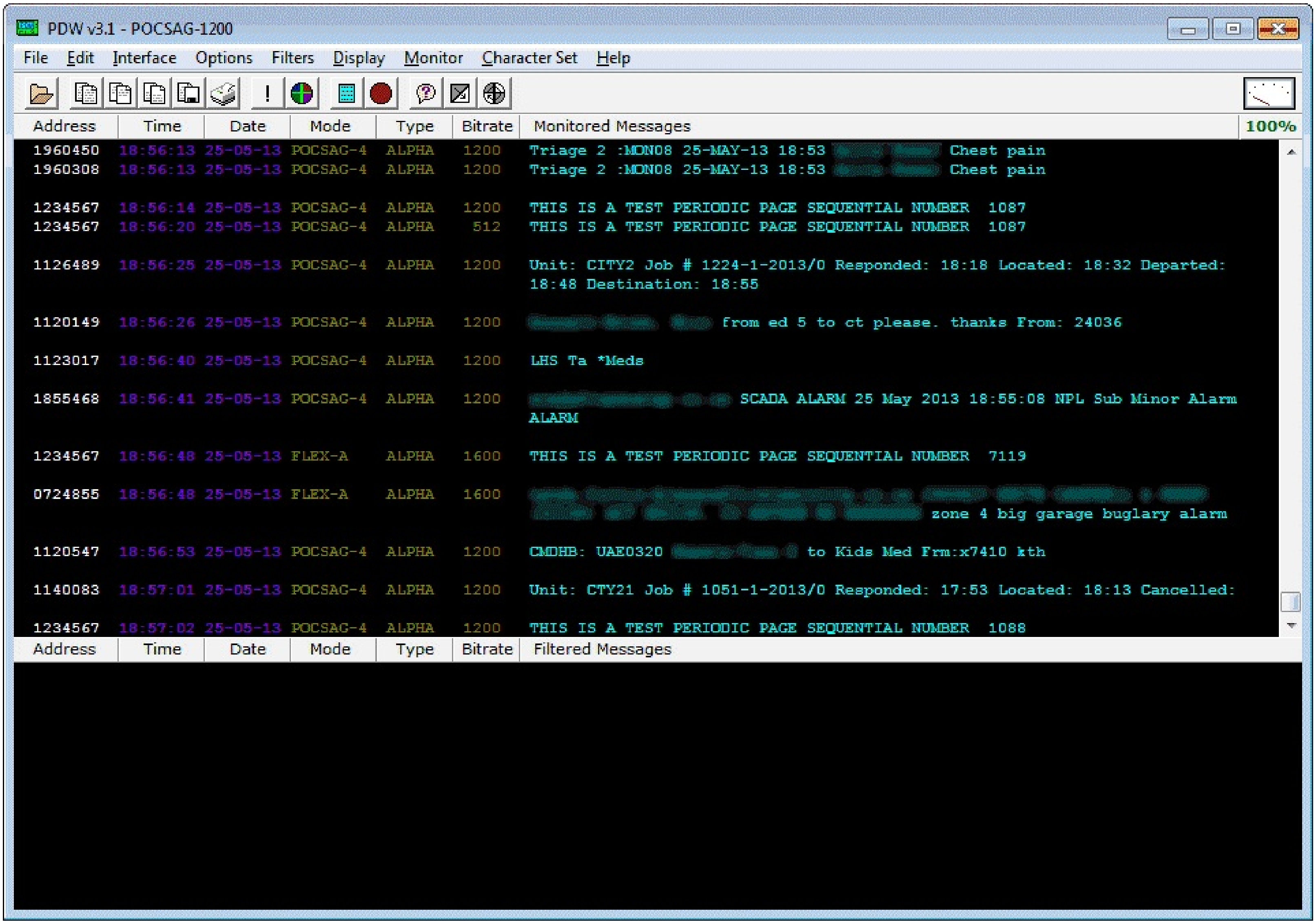
Soundcard: 2. Stereo Mix (VIA High Definition

5. Go to Monitor, and ensure POCSAG/FLEX is ticked.

Now, if everything is set up correctly, the pager audio from SDRSharp 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 SDRSharp, 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.



```
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]
                                                                                  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 the pager frequency in your area.

See the rtl_fm guide if you need more information.

```
./rtl_fm -f 158.1M -s 22050 -p 49 | multimon-ng -t raw -a POCSAG512 -a POCSAG1200 -a POCSAG2400 -f alpha /dev/stdin
```

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 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.

Items you will need to observe the hydrogen line:

- Large dish (>1m) or high gain Yagi antenna or Helical antenna or Horn antenna.
- 1-2 LNA Preamps.
- 1420 MHz bandpass filter.

Marcus Leech was one of the first people to experiment with Radio Astronomy and the RTL-SDR, 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. However, another experimenter has had success with a 22 element Yagi http://www.y1pwe.co.uk/RAProgs/HLRrtl.pdf which may be cheaper and easier to build. 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.

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.

The Hydrogen line is observed by pointing the antenna to the sky and tuning the radio to 1420 MHz. Then 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.

The result will be a frequency spectrum graph showing a distinct peak in power at the 1420 MHz frequency.

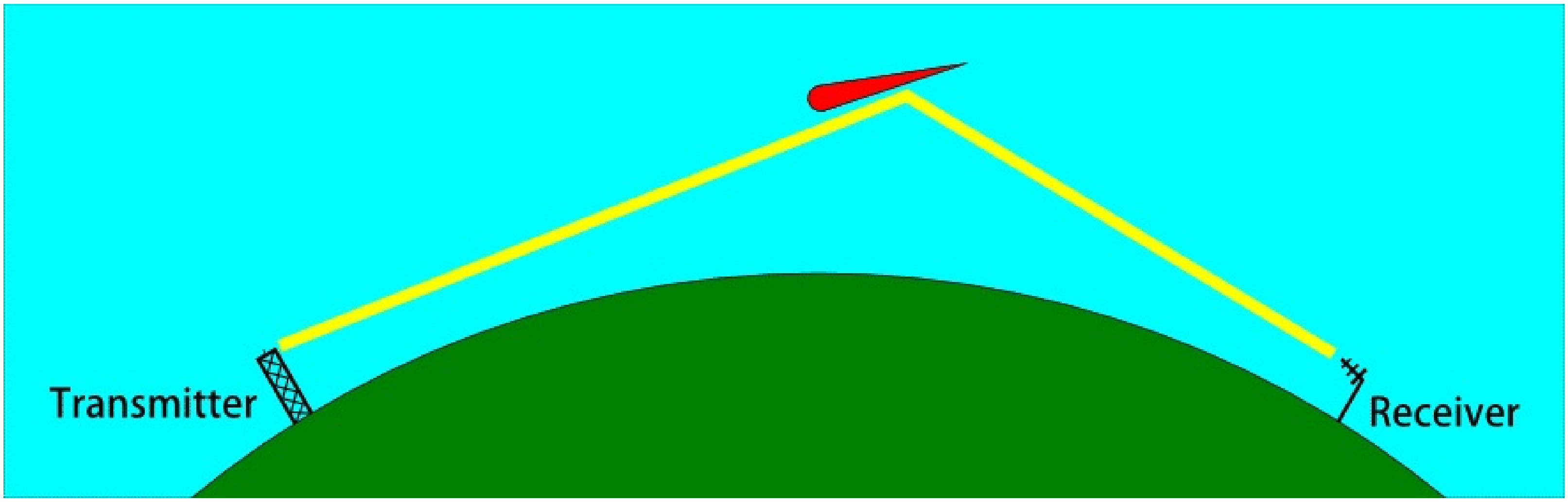
Some more technical papers:

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

Meteor scatter detection works by detecting signals from strong but distant broadcast radio stations using signal reflections from meteors as they enter the atmosphere. As a meteor enters the atmosphere, it ionizes the air behind it leaving a trail of RF reflective ionized air. 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.

To do meteor detection there needs to be a strong broadcasting station that is between 300 to 1200 km away from your receiving antenna. This is far enough so that you cannot receive the signal directly, but close enough so that a meteor in the sky can reflect the signal down to your receiver.



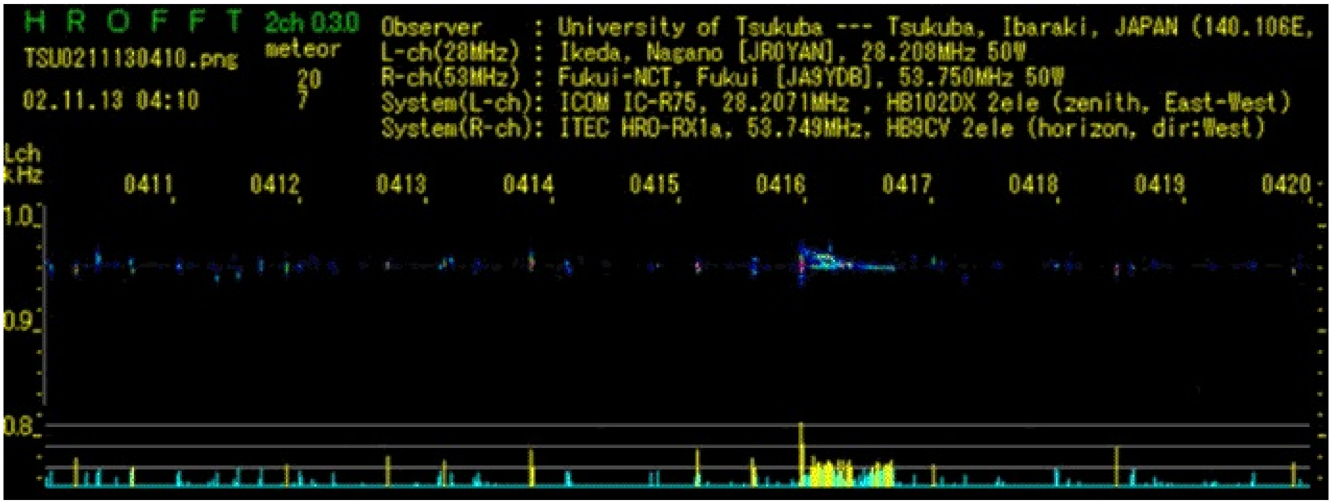
As meteor scatter signals can be weak you will also need a high gain Yagi-Uda antenna tuned to the band you are monitoring with it pointed towards the sky in the direction of the broadcasting station. Non directional antennas can work, but may not have enough gain for good performance. Some people have reported success with magnetic loop antennas. An optional LNA is also desirable.

A good paper describing meteor scatter detection with the RTL-SDR can be found here http://www.sbrac.org/files/meteor_forward_scatter.pdf.

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. Set the received mode to USB and increase its 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.



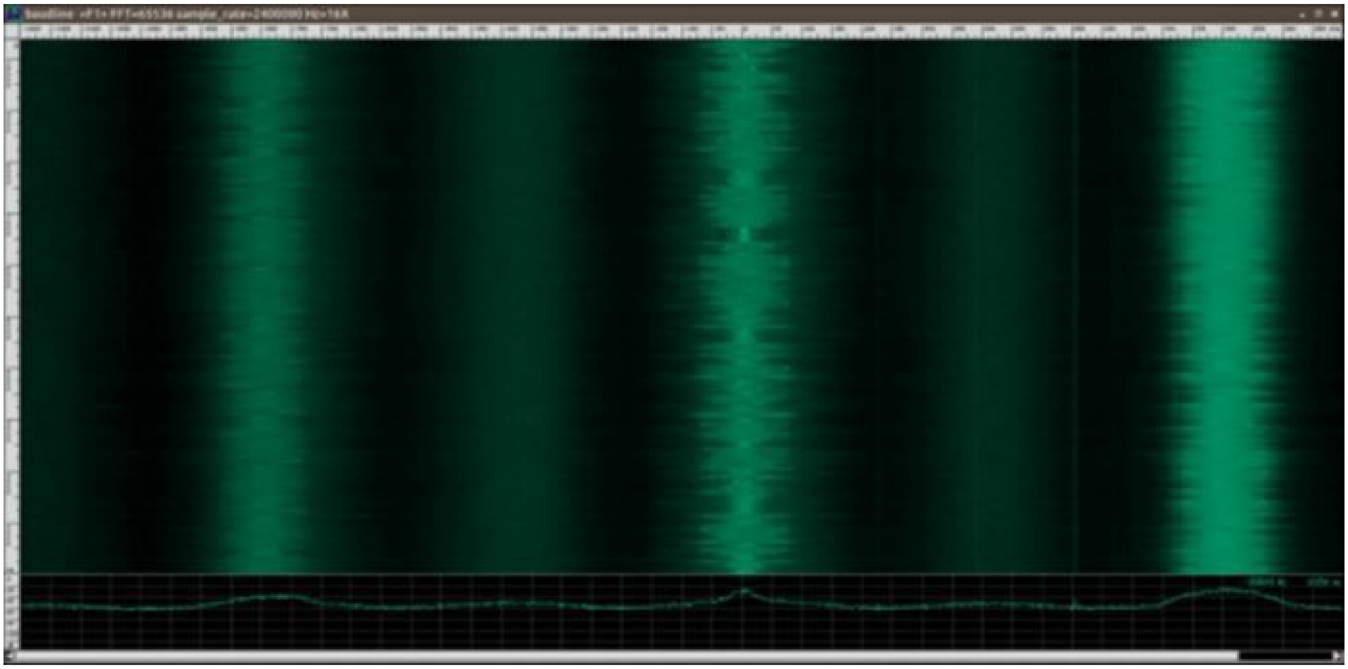
```
H R O F F T        2ch 0.3.0  Observer     : University of Tsukuba --- Tsukuba, Ibaraki, JAPAN (140.106E,
TSU0211130410.png  meteor     L-ch(28MHz)  : Ikeda, Nagano [JR0YAN], 28.208MHz 50W
                     20       R-ch(53MHz)  : Fukui-NCT, Fukui [JA9YDB], 53.750MHz 50W
02.11.13 04:10        7       System(L-ch): ICOM IC-R75, 28.2071MHz , HB102DX 2ele (zenith, East-West)
                              System(R-ch): ITEC HRO-RX1a, 53.749MHz, HB9CV 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
```



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.

HF MODES DECODING GUIDE

To receive HF signals you will need an RTL-SDR dongle with an upconverter or direct sampling mod enabled. Or, you will need a HF capable SDR radio such as the Funcube Dongle or SoftRock SDR.

SHORTWAVE BROADCAST RADIO

Shortwave broadcast radio is simply international AM radio that is sent over HF. All you need to receive this is an upconverter a decent HF antenna and an 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.

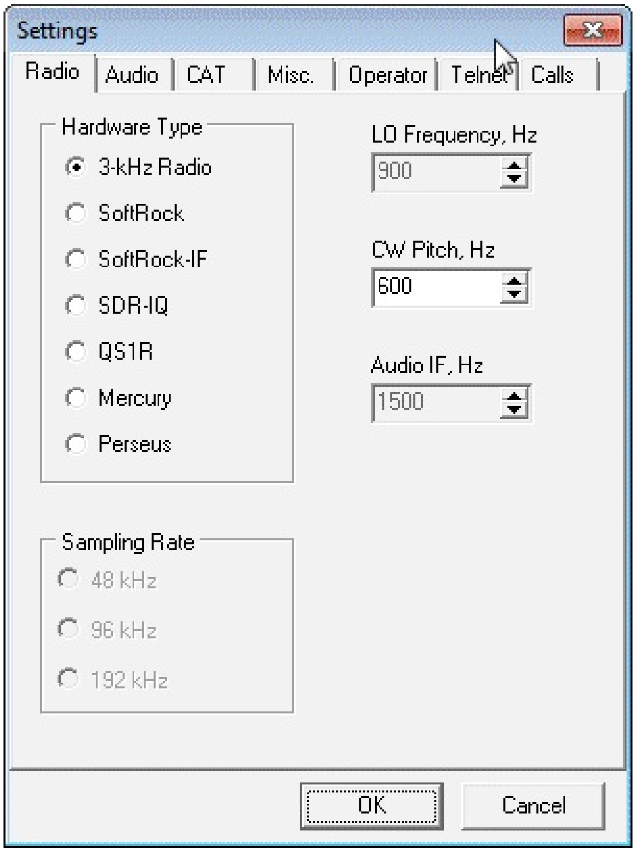
CW (MORSE CODE)

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. Morse code is easy for humans who have learnt it to decode, but it actually quite difficult for computers to decode accurately. Nevertheless, there are several CW decoders available.

All decoders require the CW 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 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).



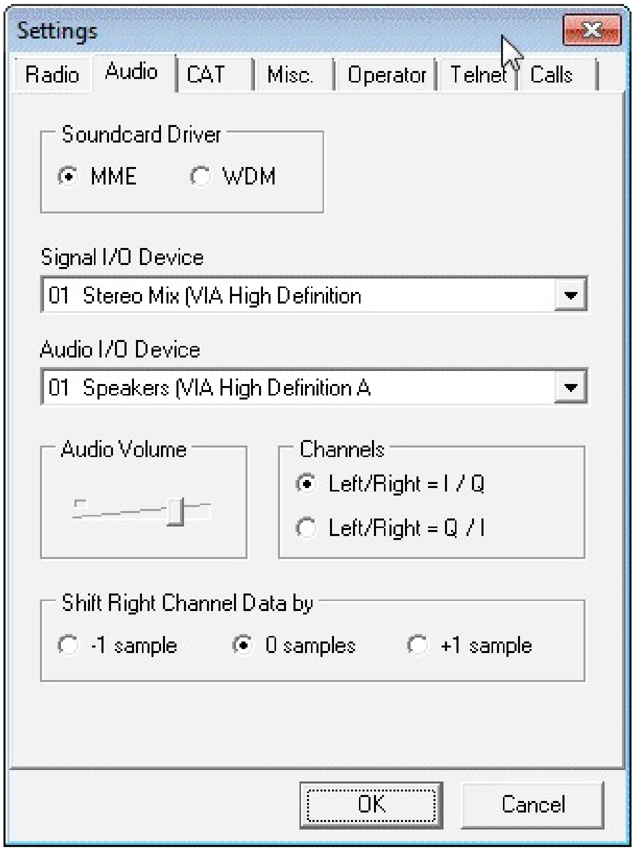
Hardware Type
- ● 3-kHz Radio
- ○ SoftRock
- ○ SoftRock-IF
- ○ SDR-IQ
- ○ QS1R
- ○ Mercury
- ○ Perseus

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 alternatives with good decoders 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 get CW decoding.

Fldigi 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

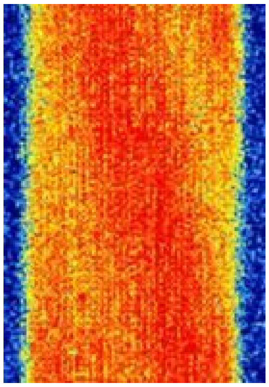
RTTY is an acronym for Radioteletype and is a way for radio amateurs to send text messages over radio, similar to CW. A list of RTTY frequencies can be found at http://www.arrl.org/band-plan.

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 RTL-SDR.com signal identification guide page and play the example STANAG 4285 signal using Windows stereo mix as your default audio piping method. The programs should be able to decode from this example audio.

SORCERER

Sorcerer is a decoder program that used to be 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 the audio piping method used.

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.

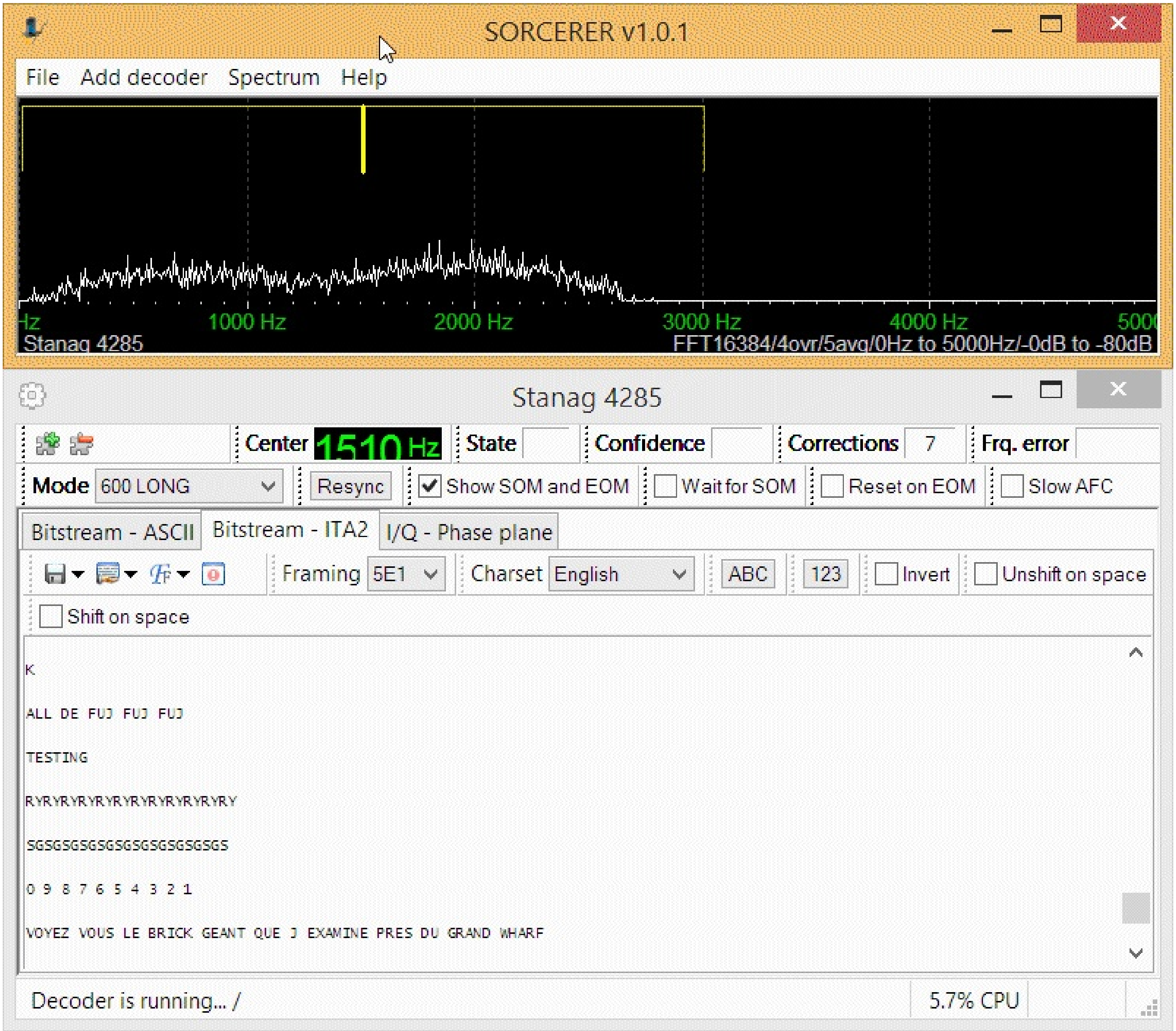
4. Go to **Add Decoder ->PSK->Stanag 4285**. Double click on Stanag 4285 to start that decoder.

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. 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.

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

SGSGSGSGSGSGSGSGSGSGSGSG

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... /

5.7% CPU

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 folder described also on the website. If this isn’t done Sigmira will fail to open. If Sigmira later 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

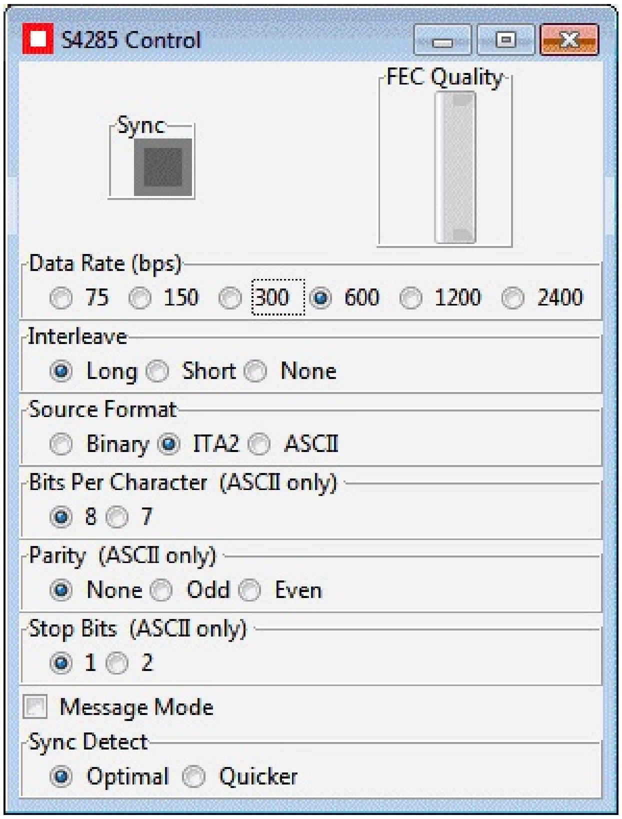
Windows sound properties for the audio pipe to unmute it if this happens.

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

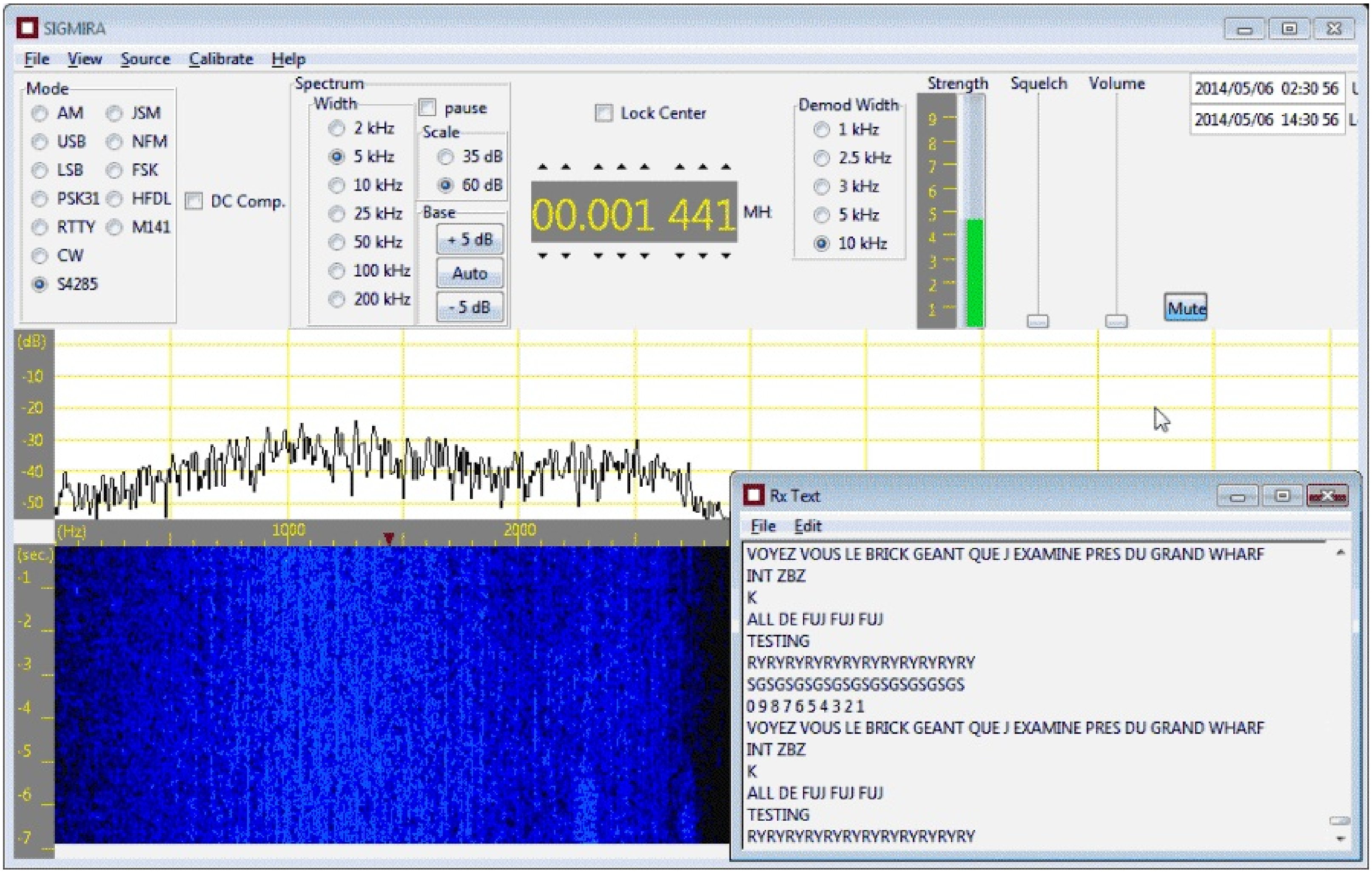
Stop Bits (ASCII only)
● 1 ○ 2

□ 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 shortly. It may take some trial and error to correctly centre the red arrow.



Mode: ○ AM ○ JSM ○ USB ○ NFM ○ LSB ○ FSK ○ PSK31 ○ HFDL ○ RTTY ○ M141 ○ CW ● S4285

□ DC Comp.

Width: ○ 2 kHz ● 5 kHz ○ 10 kHz ○ 25 kHz ○ 50 kHz ○ 100 kHz ○ 200 kHz

□ pause

Scale: ○ 35 dB ● 60 dB

Base: + 5 dB | Auto | - 5 dB

□ Lock Center

00.001 441 MH

Demod Width: ○ 1 kHz ○ 2.5 kHz ○ 3 kHz ○ 5 kHz ● 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
RYRYRYRYRYRYRYRYRYRYRYRY
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
RYRYRYRYRYRYRYRYRYRYRYRY
```

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

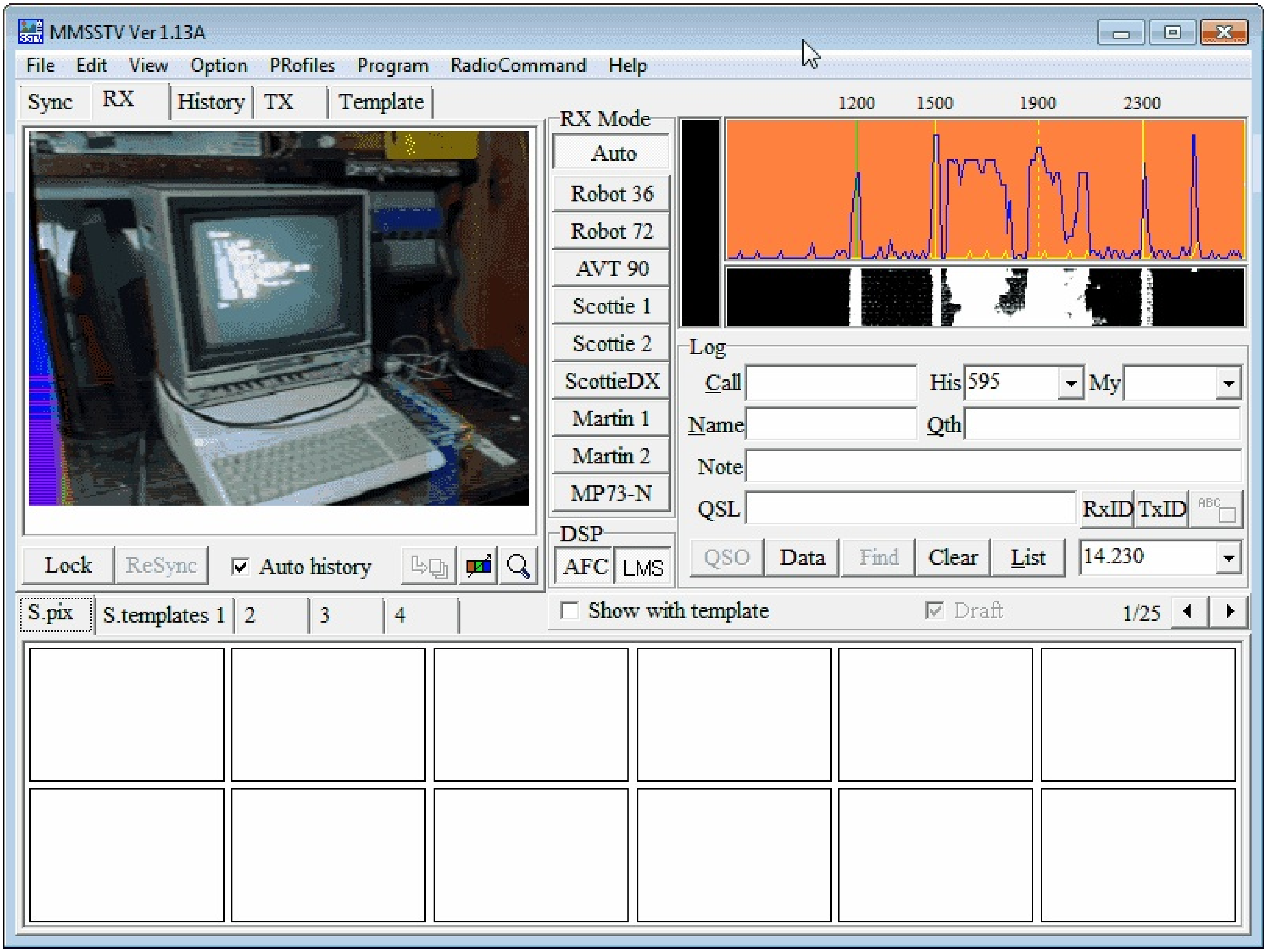
SSTV is an acronym for Slow Scan TV and is a mode used by amateur radio hobbyists to send small calling card images mainly on HF. There are several SSTV formats available, with Scottie S1 and Martin S1 being the most common. SSTV is most commonly found on or near the following frequencies in MHz.

- 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 from military satellites by Brazilian and Mexican radio pirates. The pirates hijack a military satellite transponder to transmit SSTV images and other signals. More information here http://www.qsl.net/py4zbz/ec/ec.htm.

A good SSTV decoder is MMSSTV from http://hamsoft.ca/pages/mmsstv.php. MMSSTV will automatically detect the SSTV mode used. Simply pipe the audio from a general SDR receiver such as SDR#.

Some SSTV audio samples can be found at https://www.youtube.com/watch?v=Ke5UB2HLaZ4.


MMSSTV Ver 1.13A

☑ Auto history

His 595

14.230

☐ Show with template

☑ Draft

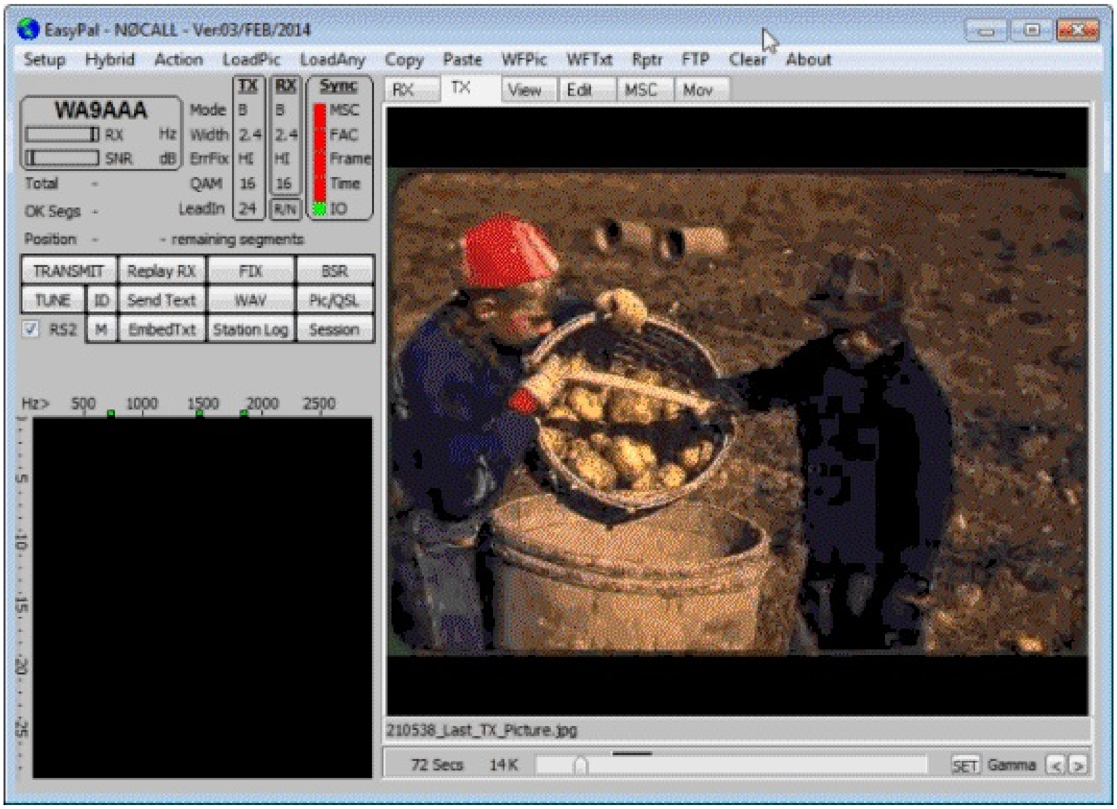
1/25

DIGITAL SSTV

Digital SSTV is a modern SSTV mode which 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 digital SSTV decoder is EasyPal which can be downloaded from http://www.kc1cs.com/.


EasyPal - NØCALL - Ver03/FEB/2014
Setup Hybrid Action LoadPic LoadAny Copy Paste WFPic WFTxt Rptr FTP Clear About
RX TX View Edit MSC Mov
WA9AAA
RX Hz
SNR dB
Total -
OK Segs -
Position - - remaining segments
TX RX Sync
MSC FAC Frame Time IO
TRANSMIT Replay RX FIX BSR
TUNE ID Send Text WAV Pic/QSL
RS2 M EmbedTxt Station Log Session
Hz> 500 1000 1500 2000 2500
5 10 15 20 25
SET Gamma

	TX	RX
Mode	B	B
Width	2.4	2.4
ErrFix	HI	HI
QAM	16	16
LeadIn	24	R/N

☑ RS2

210538_Last_TX_Picture.jpg

72 Secs 14K

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

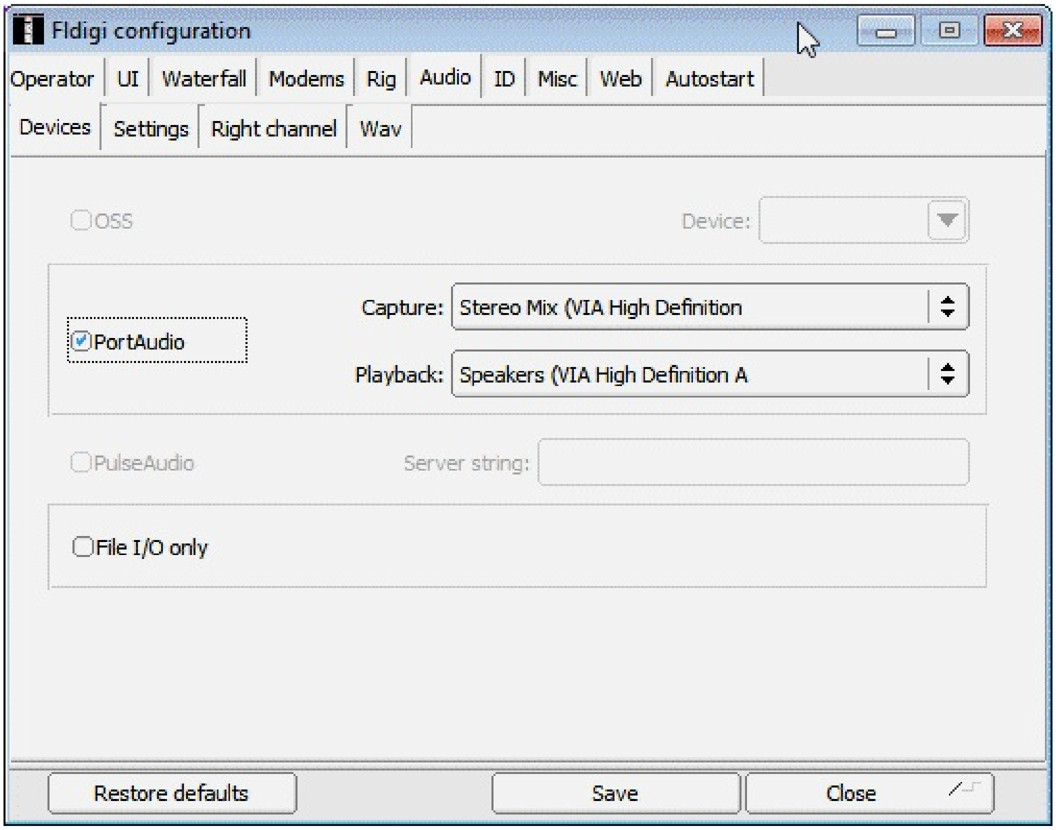
WEATHERFAX (HFFAX) DECODING GUIDE

WeatherFax is also known as HFFAX and is a HF mode used to transmit pictures of weather information charts. It is mainly used by mariners at sea. There is also a Japanese newspaper known as “Kyodo News” which transmits entire pages of its newspapers via HFFAX. There is a large document of radio fax frequencies for many countries that can be found from NOAA here http://www.nws.noaa.gov/om/marine/rfax.pdf.

To receive HFFAX with the RTL-SDR you will need an upconverter or a dongle modified with the direct sampling mode.

To decode HF was use a general SDR receiver like SDRSharp or SDR-RADIO, and pipe the audio to 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 weather fax follow these instructions

1. Open your SDR receiver and set the audio output method to your preferred audio piping method.
2. Tune to a weather fax 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

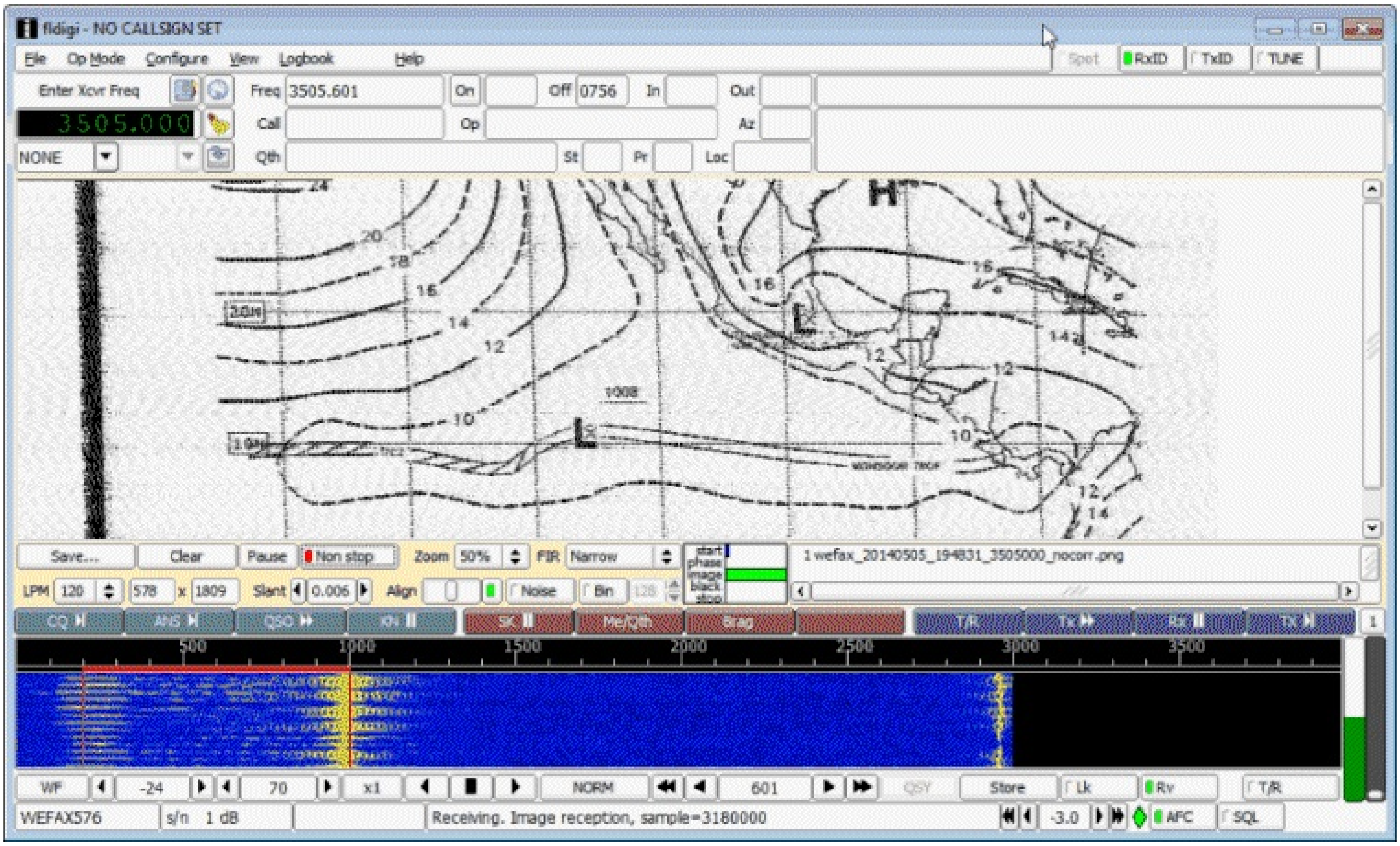
Capture: Stereo Mix (VIA High Definition

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.



3505.000

NONE

Freq 3505.601
On
Off 0756

Non stop
Zoom 50%
FIR Narrow
1 wefax_20140505_194831_3505000_nocorr.png

LPM 120
578 x 1809
Slant 0.006
Align

-24
70
x1
601

WEFAX576
s/n 1 dB
Receiving. Image reception, sample=3180000
-3.0

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 REMAINING MODES WE WILL INTRODUCE THEM FIRST AND THEN SHOW HOW TO DECODE THEM IN MULTIPSK IN A LATER SECTION.

SYNOP

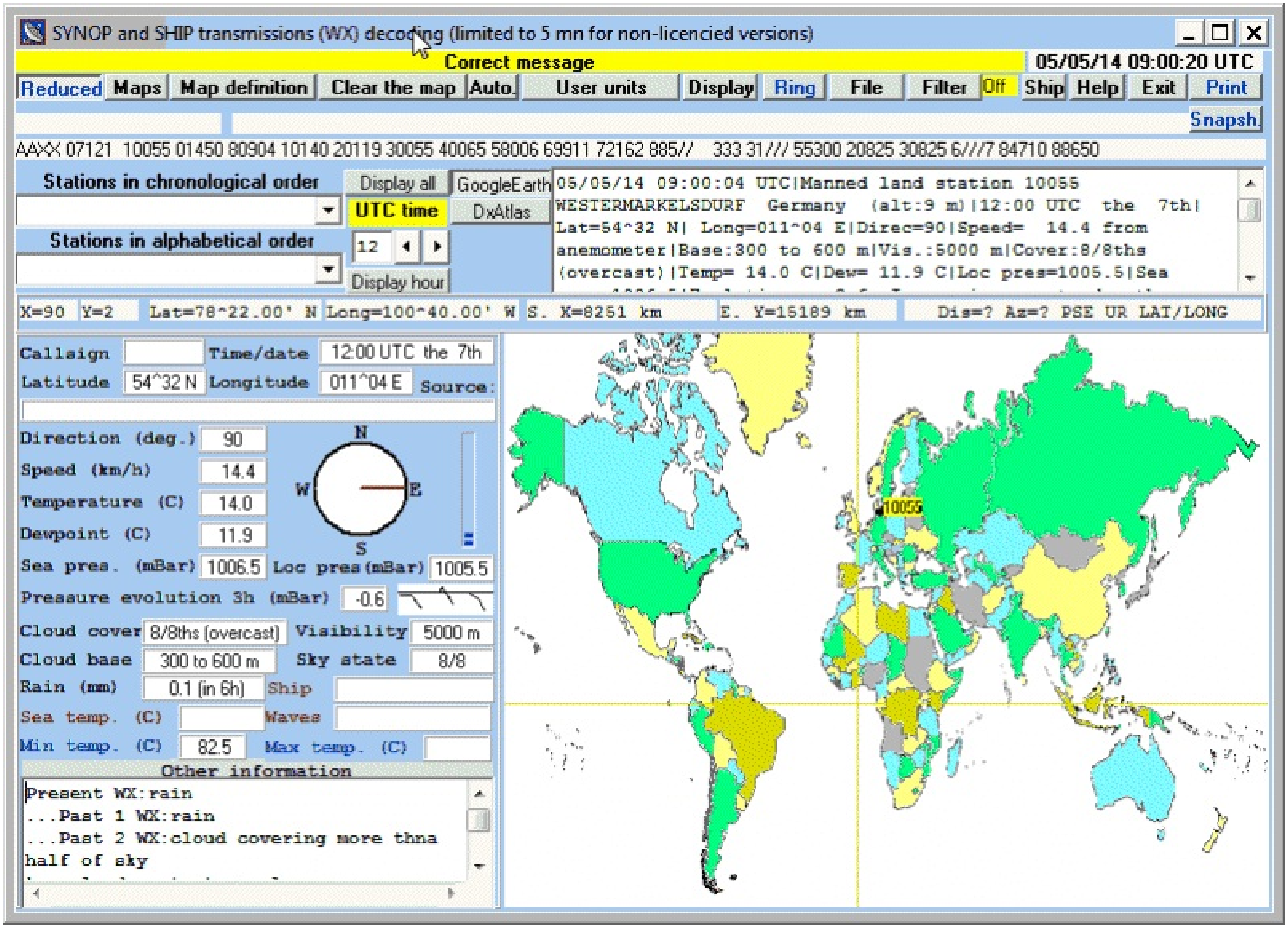
Synop (Surface Synoptic Observations) is a type of RTTY signal used on HF by the German meteorological service Deutscher Wetterdienst for sending out weather reports. The reports are generally intended for ships at sea. SYNOP message use 5 digit codes to encode a set number of weather parameters. Synop uses LSB mode. It can usually be received around Europe. 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.



Correct message

05/05/14 09:00:20 UTC

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 on a Navtex machine. The messages usually contain information about maritime weather. Uses USB mode. Navtex 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 error correction to correct errors. SITOR-B is also known as Navtex.

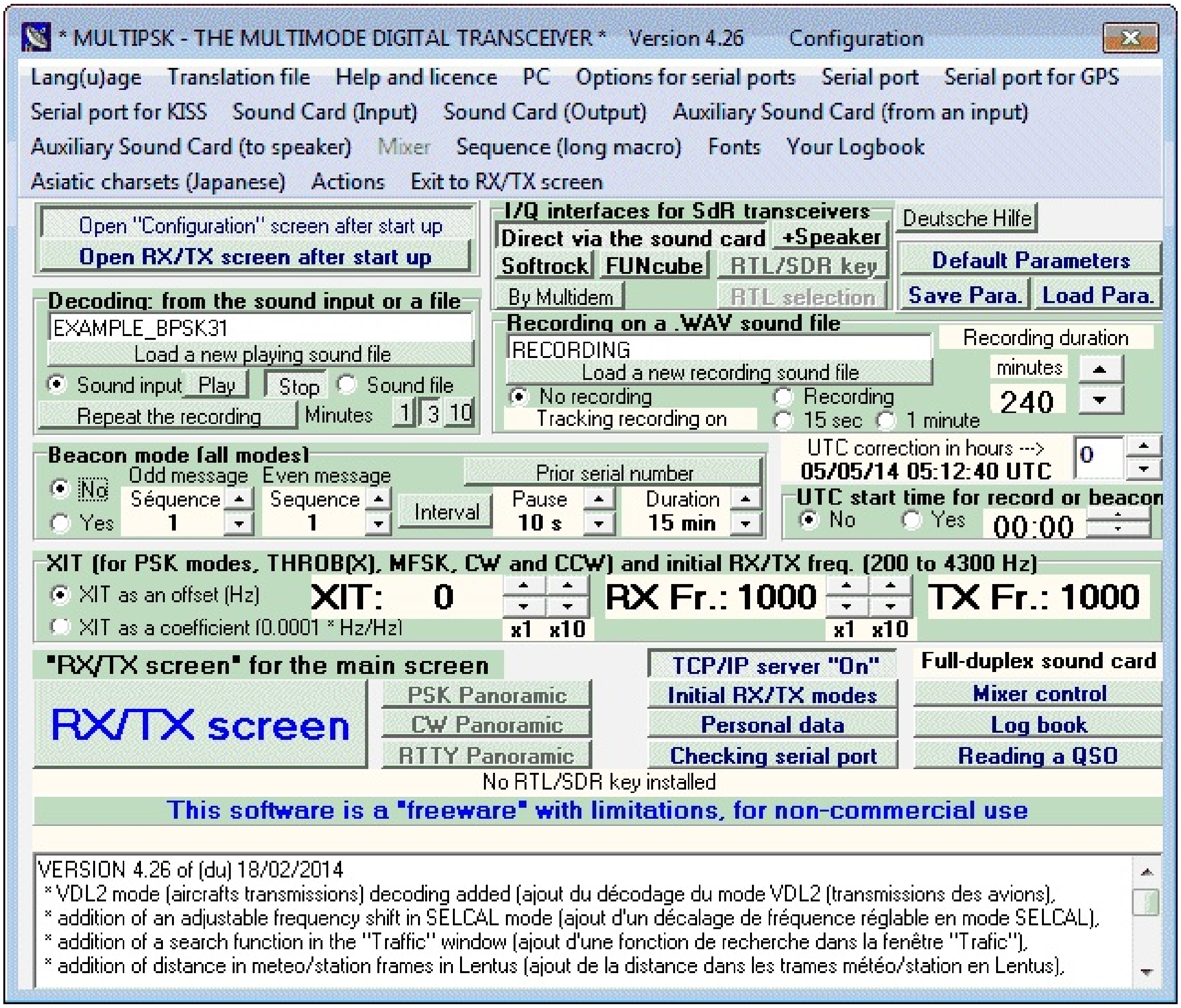
DECODING MARINE MODES WITH MULTIPSK

MultiPSK is 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-Console.
2. Set the SDR receiver to use your favourite audio piping method.
3. Open MultiPSK. You'll first be greeted with the following screen.



EXAMPLE_BPSK31

RECORDING

240

● Sound input

● No recording

● No

Séquence 1 Sequence 1

Pause 10 s Duration 15 min

0

05/05/14 05:12:40 UTC

● No 00:00

● XIT as an offset (Hz)

XIT: 0 RX Fr.: 1000 TX Fr.: 1000

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.
6. Click on the big RX/TX screen button in the bottom left.

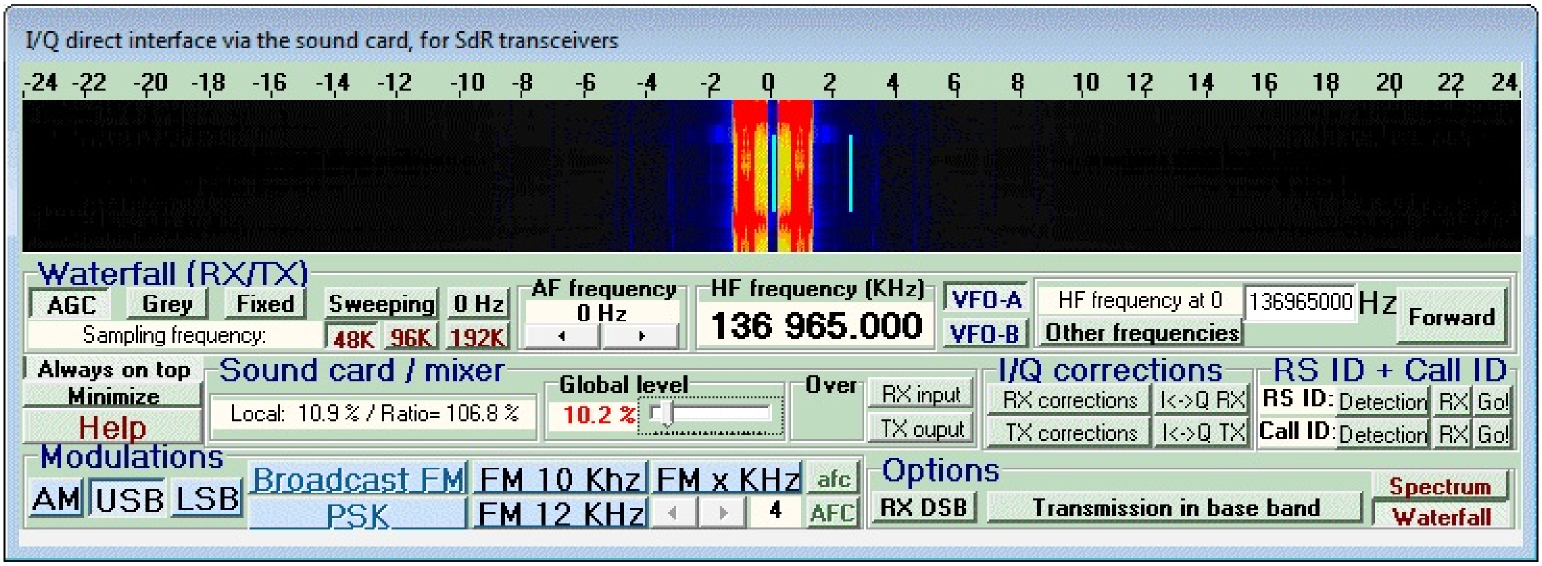
Now you are ready to begin decoding signals. But first for modes like Synop, we need to set up the maps.

1. Download maps.zip from the multiPSK webpage (http://f6cte.free.fr/index_anglais.htm). 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. SYNOP/SHIP SITOR A GMDSS Amtor FEC-Navtex

2. In the “I/Q direct Interface via the sound card, for SdR transceivers” window set the AF frequency to zero.



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:

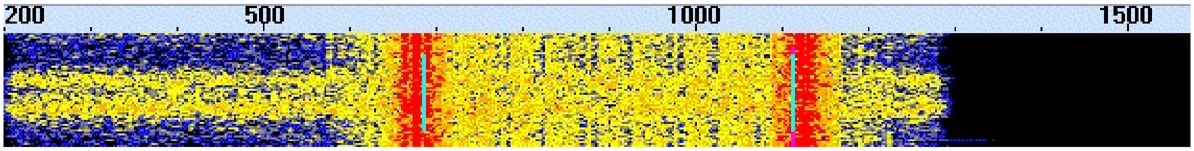
Local: 10.9 % / Ratio= 106.8 %

Global level 10.2 %

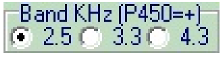
RS ID: Detection

Call ID: Detection

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. Spectrum Waterfall



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

DIFFERENTIAL GPS (DGPS)

GPS accuracy used to be purposely degraded by the US military to prevent enemies using it against them. 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
```

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), good antenna and decoding software the RTL-SDR can 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.

TUTORIAL: HOW TO RECEIVE AND DECODE DRM SIGNALS

To receive DRM with rtl-sdr, you will need four things

- An HF upconverter or dongle modified with the direct sampling mod.
- 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

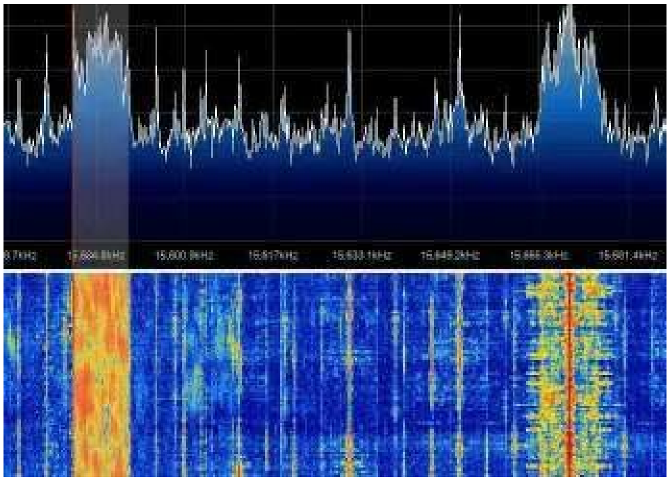
faad2_drm.dll file into your DREAM folder.

3. Now you can open SDRSharp 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 as well. AGC can be left on, but it may need to be experimented with in order to get the best decoding performance.

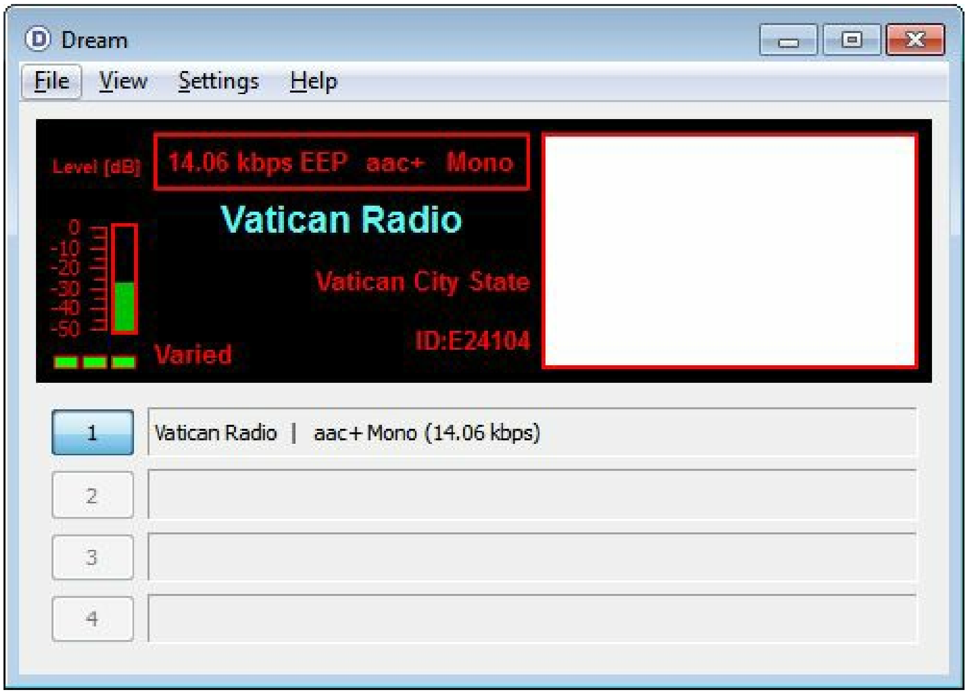
5. You should also experiment with the filter order. Usually a large filter order of 100+ works well especially if there are strong signals nearby.

6. A DRM signal looks like this (left) on the waterfall, placed next to a normal shortwave AM signal (right).



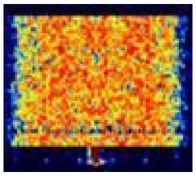
7. Now, open DREAM and then go to **Settings->Sound Card->Signal Input->Device** and set your audio pipe as the input device. Also, ensure that **Settings->Sound Card->Signal Input->Sample Rate** is set to 48000Hz.

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. 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.



HFDL DECODING GUIDE

HFDL is an acronym for High Frequency Data Link. It is essentially ACARS 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 waterfall image to the below.



To decode HFDL we can use a program known as PC-HFDL. 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 general purpose SDR receiver and set the audio output to your preferred audio piping method.

2. Tune to a HFDL station using Upper Side Band (USB) mode.

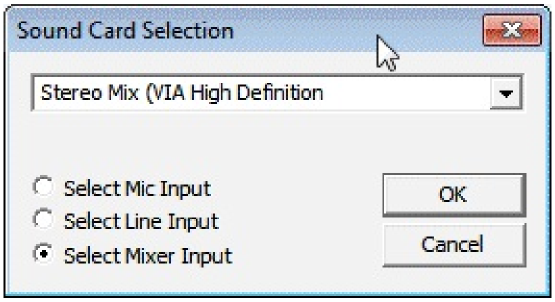
3. Ensure that the signal is tuned to 1440Hz below the centre of the HFDL packets. This is important.

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.

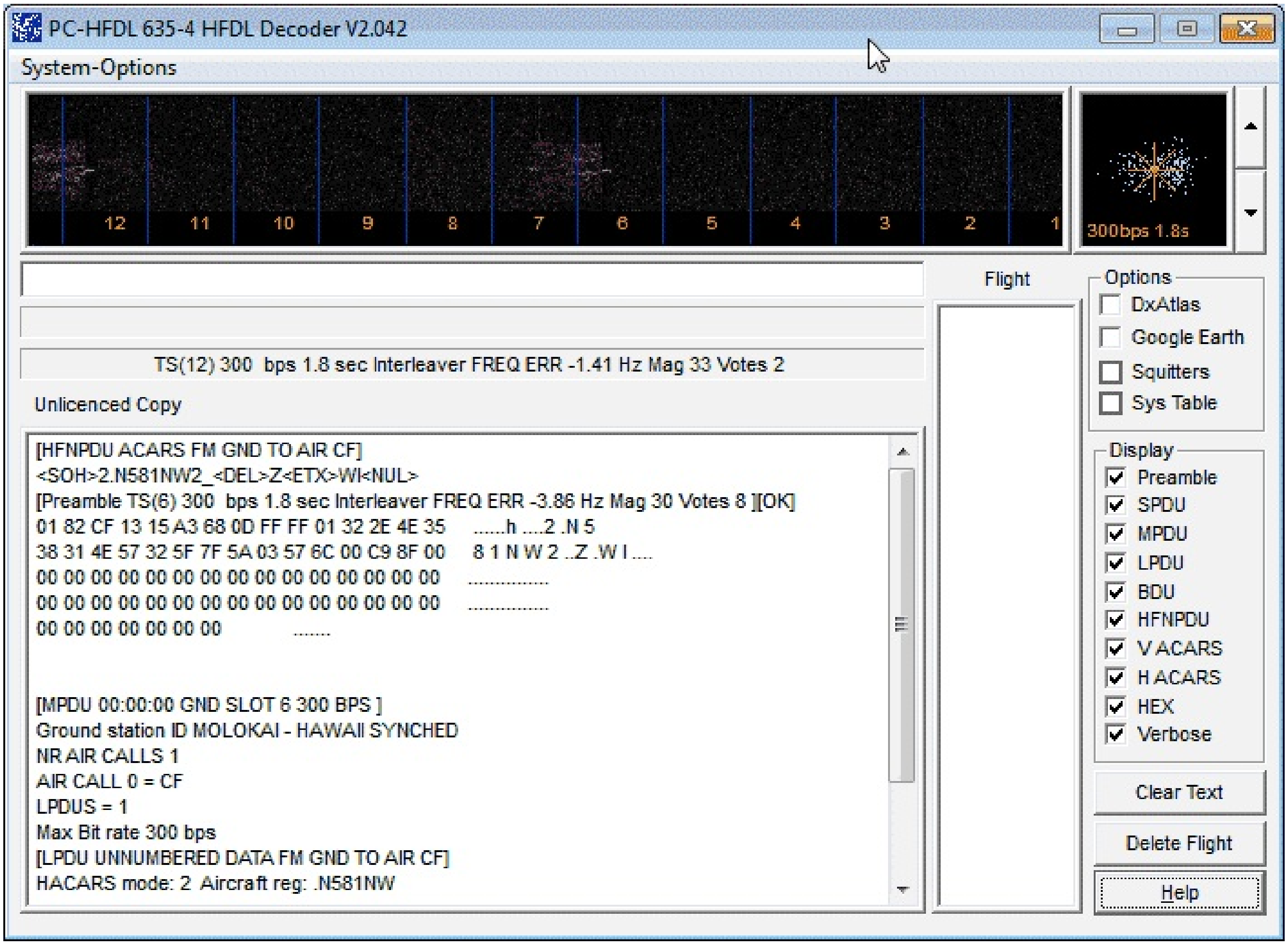


Sound Card Selection

Stereo Mix (VIA High Definition

○ Select Mic Input
○ Select Line Input
● Select Mixer Input

8. When an HFDL packet is received, PC-HFDL will decode it and display it in its main window.



PC-HFDL 635-4 HFDL Decoder V2.042

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

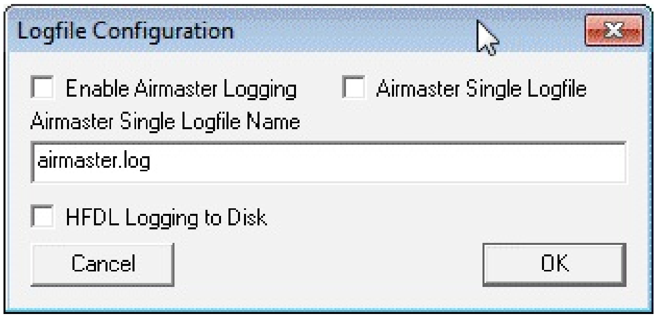
You can further extend the information received by PC-HFDL by using a second program called PC-HFDL-Display. This program has a large database of aircraft and will resolve aircraft registration numbers and show you the aircraft type and airline they fly for.

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

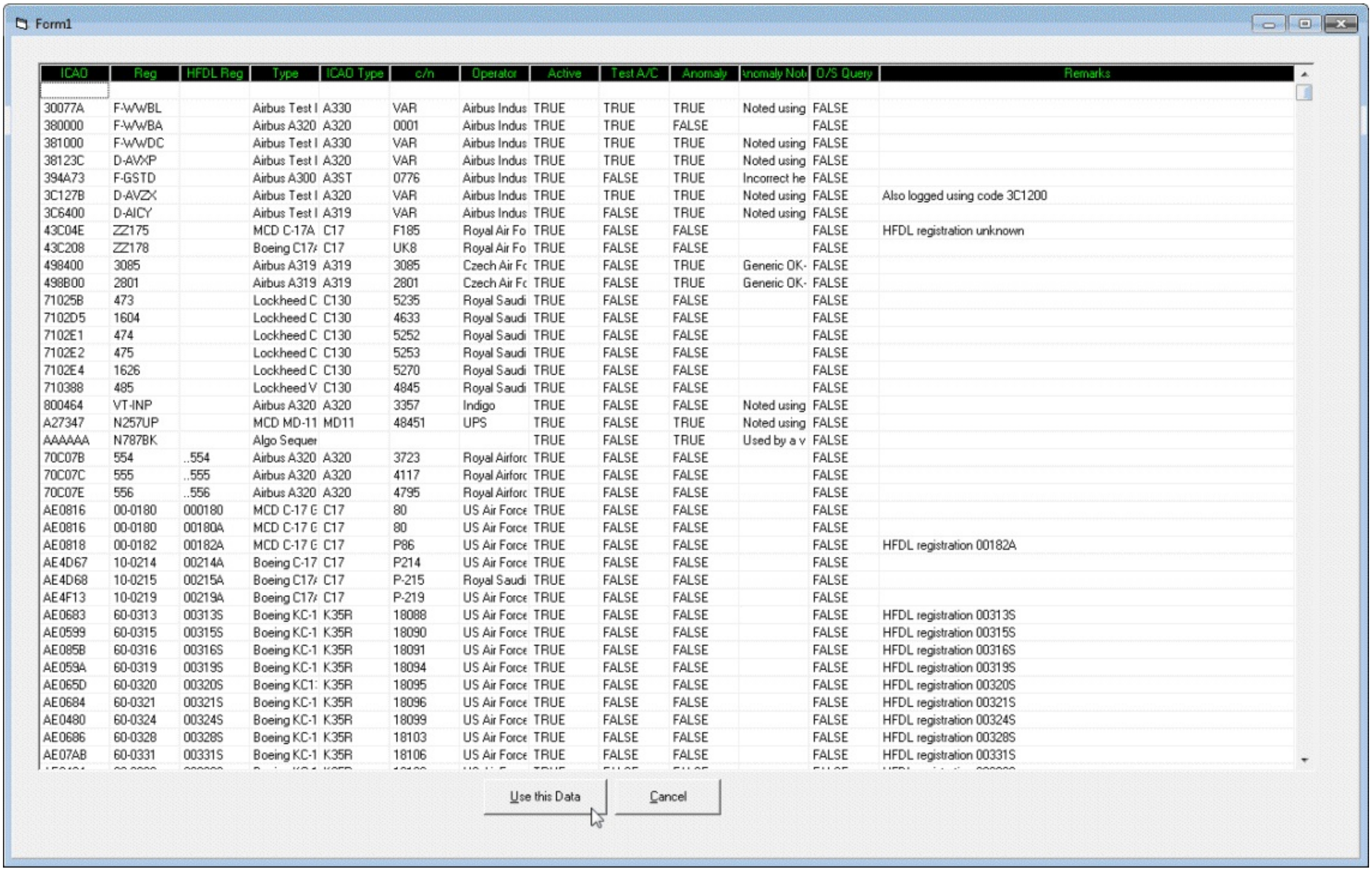
Airmaster Single Logfile Name

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. Click 'Use this Data' on the screen that pops up. You may also get a warning about not having a log path setup, you can ignore this for now.



Form1

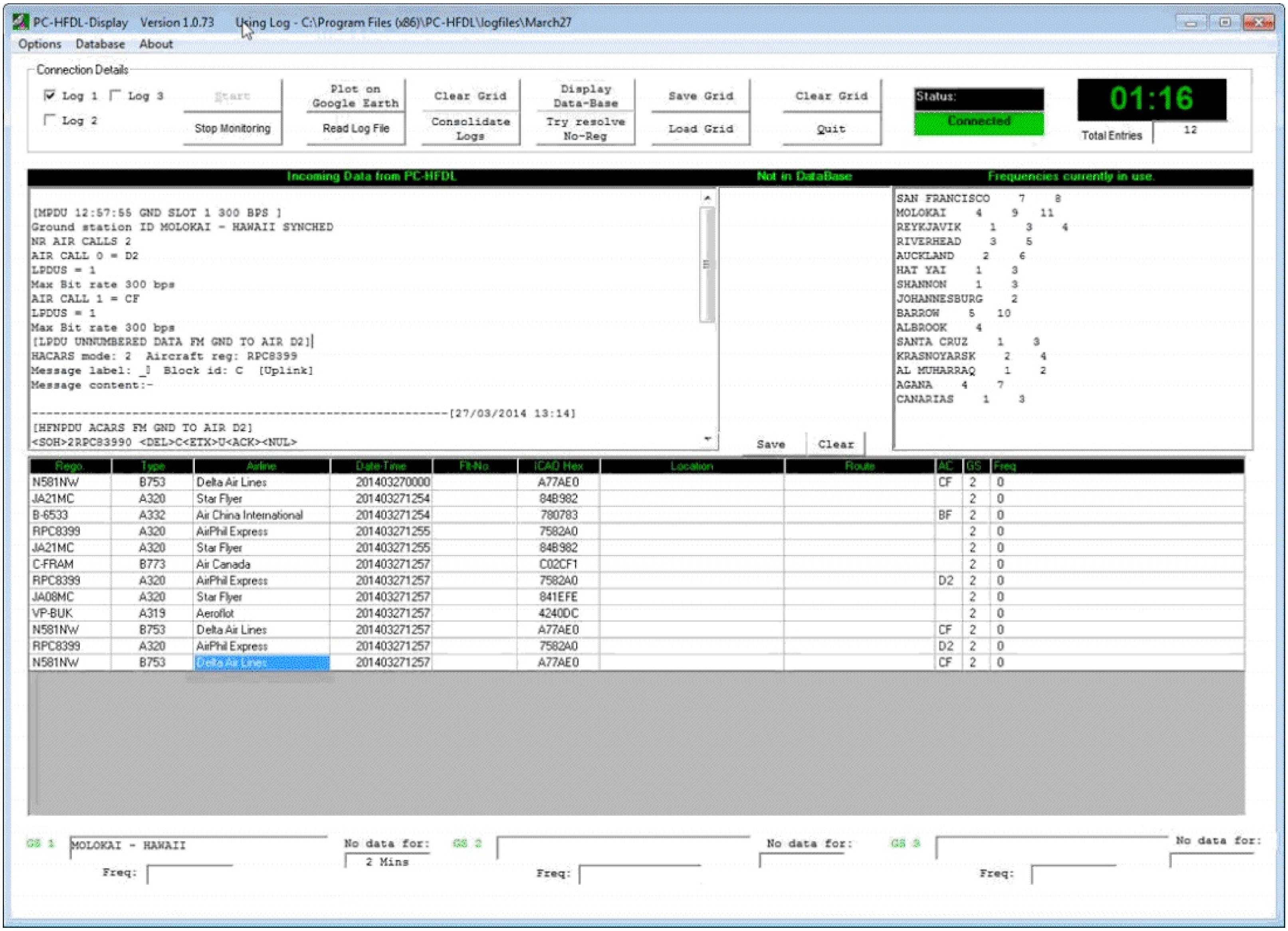
ICAO	Reg	HFDL Reg	Type	ICAO Type	c/n	Operator	Active	Test A/C	Anomaly	Anomaly Not	O/S Query	Remarks
30077A	F-WWBL		Airbus Test I	A330	VAR	Airbus Indus	TRUE	TRUE	TRUE	Noted using	FALSE	
380000	F-WWBA		Airbus A320	A320	0001	Airbus Indus	TRUE	TRUE	FALSE		FALSE	
381000	F-WWDC		Airbus Test I	A330	VAR	Airbus Indus	TRUE	TRUE	TRUE	Noted using	FALSE	
38123C	D-AVXP		Airbus Test I	A320	VAR	Airbus Indus	TRUE	TRUE	TRUE	Noted using	FALSE	
394A73	F-GSTD		Airbus A300	A3ST	0776	Airbus Indus	TRUE	FALSE	TRUE	Incorrect he	FALSE	
3C127B	D-AVZX		Airbus Test I	A320	VAR	Airbus Indus	TRUE	TRUE	TRUE	Noted using	FALSE	Also logged using code 3C1200
3C6400	D-AICY		Airbus Test I	A319	VAR	Airbus Indus	TRUE	FALSE	TRUE	Noted using	FALSE	
43C04E	ZZ175		MCD C-17A	C17	F185	Royal Air Fo	TRUE	FALSE	FALSE		FALSE	HFDL registration unknown
43C208	ZZ178		Boeing C17/	C17	UK8	Royal Air Fo	TRUE	FALSE	FALSE		FALSE	
498400	3085		Airbus A319	A319	3085	Czech Air Fc	TRUE	FALSE	TRUE	Generic OK-	FALSE	
498B00	2801		Airbus A319	A319	2801	Czech Air Fc	TRUE	FALSE	TRUE	Generic OK-	FALSE	
71025B	473		Lockheed C	C130	5235	Royal Saudi	TRUE	FALSE	FALSE		FALSE	
7102D5	1604		Lockheed C	C130	4633	Royal Saudi	TRUE	FALSE	FALSE		FALSE	
7102E1	474		Lockheed C	C130	5252	Royal Saudi	TRUE	FALSE	FALSE		FALSE	
7102E2	475		Lockheed C	C130	5253	Royal Saudi	TRUE	FALSE	FALSE		FALSE	
7102E4	1626		Lockheed C	C130	5270	Royal Saudi	TRUE	FALSE	FALSE		FALSE	
710388	485		Lockheed V	C130	4845	Royal Saudi	TRUE	FALSE	FALSE		FALSE	
800464	VT-INP		Airbus A320	A320	3357	Indigo	TRUE	FALSE	FALSE	Noted using	FALSE	
A27347	N257UP		MCD MD-11	MD11	48451	UPS	TRUE	FALSE	TRUE	Noted using	FALSE	
AAAAAA	N787BK		Algo Sequer				TRUE	FALSE	TRUE	Used by a v	FALSE	
70C07B	554	..554	Airbus A320	A320	3723	Royal Airforc	TRUE	FALSE	FALSE		FALSE	
70C07C	555	..555	Airbus A320	A320	4117	Royal Airforc	TRUE	FALSE	FALSE		FALSE	
70C07E	556	..556	Airbus A320	A320	4795	Royal Airforc	TRUE	FALSE	FALSE		FALSE	
AE0816	00-0180	000180	MCD C-17 G	C17	80	US Air Force	TRUE	FALSE	FALSE		FALSE	
AE0816	00-0180	00180A	MCD C-17 G	C17	80	US Air Force	TRUE	FALSE	FALSE		FALSE	
AE0818	00-0182	00182A	MCD C-17 G	C17	P86	US Air Force	TRUE	FALSE	FALSE		FALSE	HFDL registration 00182A
AE4D67	10-0214	00214A	Boeing C-17	C17	P214	US Air Force	TRUE	FALSE	FALSE		FALSE	
AE4D68	10-0215	00215A	Boeing C17/	C17	P-215	Royal Saudi	TRUE	FALSE	FALSE		FALSE	
AE4F13	10-0219	00219A	Boeing C17/	C17	P-219	US Air Force	TRUE	FALSE	FALSE		FALSE	
AE0683	60-0313	00313S	Boeing KC-1	K35R	18088	US Air Force	TRUE	FALSE	FALSE		FALSE	HFDL registration 00313S
AE0599	60-0315	00315S	Boeing KC-1	K35R	18090	US Air Force	TRUE	FALSE	FALSE		FALSE	HFDL registration 00315S
AE085B	60-0316	00316S	Boeing KC-1	K35R	18091	US Air Force	TRUE	FALSE	FALSE		FALSE	HFDL registration 00316S
AE059A	60-0319	00319S	Boeing KC-1	K35R	18094	US Air Force	TRUE	FALSE	FALSE		FALSE	HFDL registration 00319S
AE065D	60-0320	00320S	Boeing KC1:	K35R	18095	US Air Force	TRUE	FALSE	FALSE		FALSE	HFDL registration 00320S
AE0684	60-0321	00321S	Boeing KC-1	K35R	18096	US Air Force	TRUE	FALSE	FALSE		FALSE	HFDL registration 00321S
AE0480	60-0324	00324S	Boeing KC-1	K35R	18099	US Air Force	TRUE	FALSE	FALSE		FALSE	HFDL registration 00324S
AE0686	60-0328	00328S	Boeing KC-1	K35R	18103	US Air Force	TRUE	FALSE	FALSE		FALSE	HFDL registration 00328S
AE07AB	60-0331	00331S	Boeing KC-1	K35R	18106	US Air Force	TRUE	FALSE	FALSE		FALSE	HFDL registration 00331S

7. Go to **Options->PC_HFDL Log Path 1**.

8. Navigate to the folder where you 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 decode 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 12:57:55 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    3
KRASNOYARSK    2    4
AL MUHARRAQ    1    2
AGANA    4    7
CANARIAS    1    3
```

Rego.	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		84B982				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		84B982				2	0
C-FRAM	B773	Air Canada	201403271257		C02CF1				2	0
RPC8399	A320	AirPhil Express	201403271257		7582A0			D2	2	0
JA08MC	A320	Star Flyer	201403271257		841EFE				2	0
VP-BUK	A319	Aeroflot	201403271257		4240DC				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

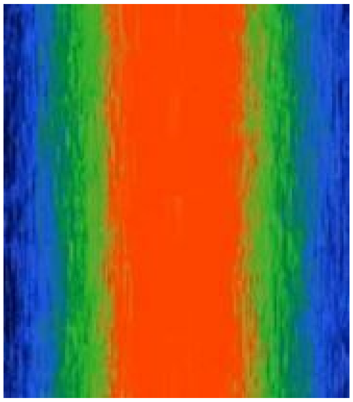
GS 1 MOLOKAI - HAWAII No data for: 2 Mins
Freq:
GS 2 No data for:
Freq:
GS 3 No data for:
Freq:

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 and connect to MultiPSK via TCP/IP as shown in other MultiPSK tutorials above.

D-STAR

D-STAR is a VHF digital amateur radio mode that is usually heard around 145.670 MHz with a bandwidth of 6.25 kHz. D-STAR uses repeaters that go through the internet which allow for global communications. A zoomed in D-STAR waterfall image looks like the image shown below.



DSD Version 1.7 (or newer) 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 audio piping device to be the default Windows recording device in the Windows sound recording properties. We recommend using either VBCable or Virtual Audio Cable for D-STAR.

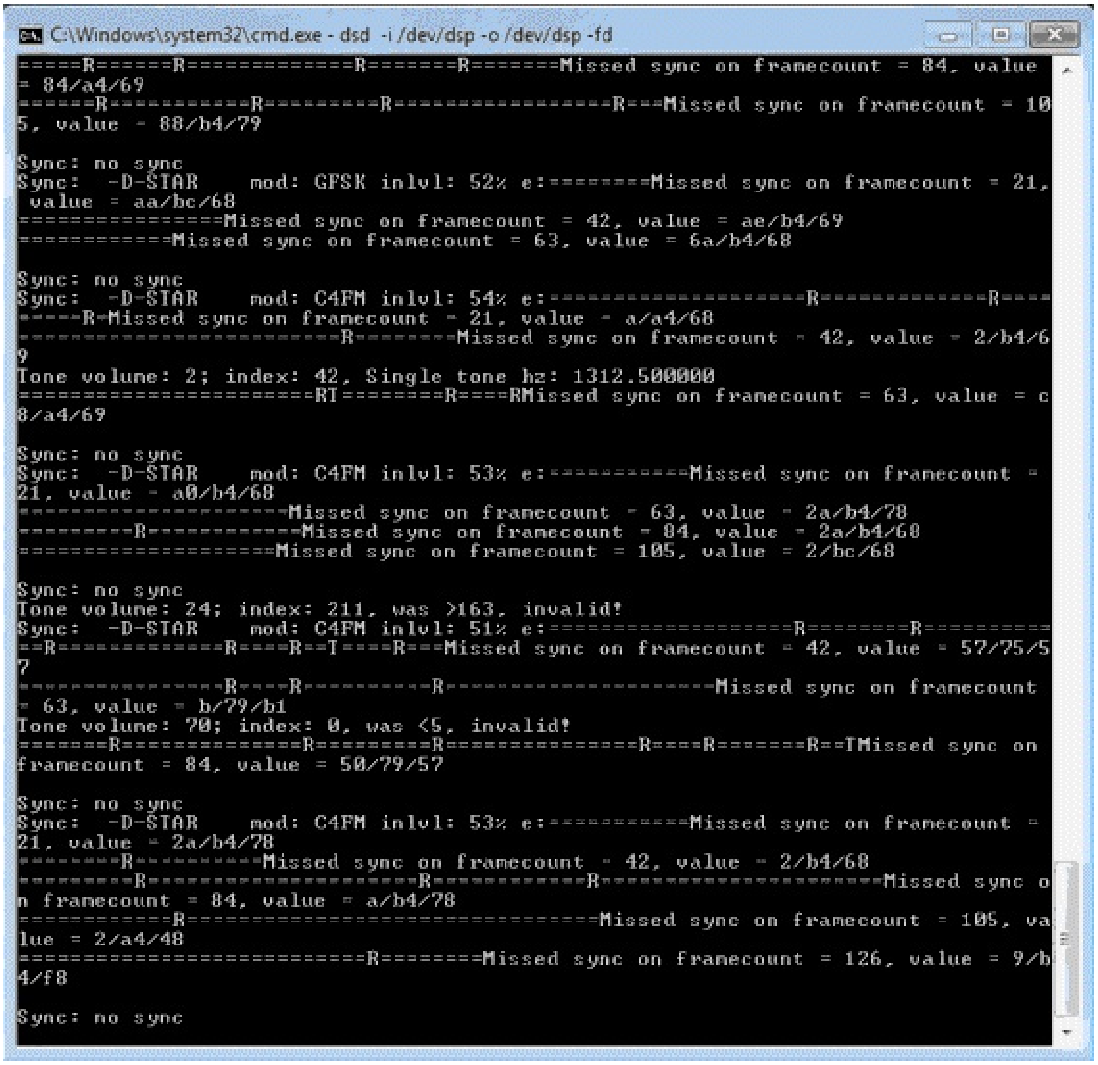
2. Open your favourite SDR receiver software like SDR# or SDR-Console and set your audio piping device. Tune to the D-STAR frequency 145.670 MHz and set the mode to NFM and the bandwidth to 6.25 kHz.

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 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.


C:\Windows\system32\cmd.exe - dsd -i /dev/dsp -o /dev/dsp -fd

```
=====R======R=============R=======R=======Missed sync on framecount = 84, value
= 84/a4/69
======R===========R=========R=================R===Missed sync on framecount = 10
5, value = 88/b4/79

Sync: no sync
Sync:  -D-STAR    mod: GFSK inlvl: 52% e:========Missed sync on framecount = 21,
 value = aa/bc/68
================Missed sync on framecount = 42, value = ae/b4/69
============Missed sync on framecount = 63, value = 6a/b4/68

Sync: no sync
Sync:  -D-STAR    mod: C4FM inlvl: 54% e:====================R=============R====
=====R=Missed sync on framecount = 21, value = a/a4/68
=========================R========Missed sync on framecount = 42, value = 2/b4/6
9
Tone volume: 2; index: 42, Single tone hz: 1312.500000
=======================RT========R====RMissed sync on framecount = 63, value = c
8/a4/69

Sync: no sync
Sync:  -D-STAR    mod: C4FM inlvl: 53% e:===========Missed sync on framecount =
21, value = a0/b4/68
=====================Missed sync on framecount = 63, value = 2a/b4/78
=========R============Missed sync on framecount = 84, value = 2a/b4/68
====================Missed sync on framecount = 105, value = 2/bc/68

Sync: no sync
Tone volume: 24; index: 211, was >163, invalid!
Sync:  -D-STAR    mod: C4FM inlvl: 51% e:===================R========R==========
==R=============R====R==T===R===Missed sync on framecount = 42, value = 57/75/5
7
================R====R==========R=====================Missed sync on framecount
= 63, value = b/79/b1
Tone volume: 70; index: 0, was <5, invalid!
=======R==============R=========R===============R====R=======R==TMissed sync on
framecount = 84, value = 50/79/57

Sync: no sync
Sync:  -D-STAR    mod: C4FM inlvl: 53% e:===========Missed sync on framecount =
21, value = 2a/b4/78
========R==========Missed sync on framecount = 42, value = 2/b4/68
=========R=====================R============R======================Missed sync o
n framecount = 84, value = a/b4/78
============R================================Missed sync on framecount = 105, va
lue = 2/a4/48
===========================R========Missed sync on framecount = 126, value = 9/b
4/f8

Sync: no sync
```

RTL-SDR FOR ANDROID

At the time of writing this book there are currently four Android RTL-SDR apps available on the Google Play marketplace.

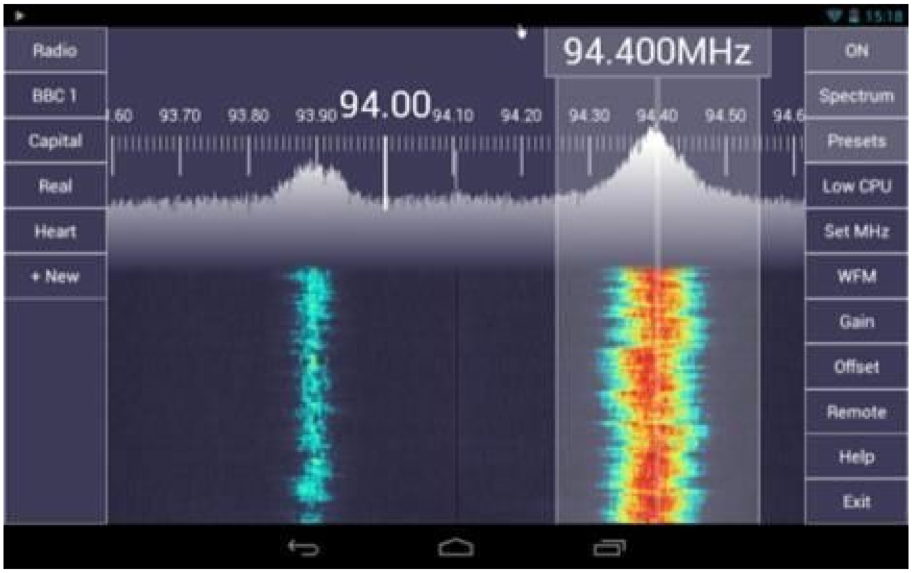
To use RTL-SDR on Android, you need a fairly powerful device that runs at least Android 4.0. You also need a special USB OTG (on-the-go) cable adapter. See the Buy RTL-SDR dongles page for a link to places where you can buy OTG cables http://www.rtl-sdr.com/buy-rtl-sdr-dvb-t-dongles/. Note that some 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. 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#. It works very well and has an RF spectrum and waterfall display. SDR Touch can be downloaded from the Google Play store by searching for “SDR Touch”. There is a trial version and a paid version. More information can be found at http://www.sdrtouch.com/.

SDR Touch can tune in WFM, NFM, AM, SSB (USB/LSB) and CW signals. It cannot decode digital signals like DRM or DAB/DAB+.

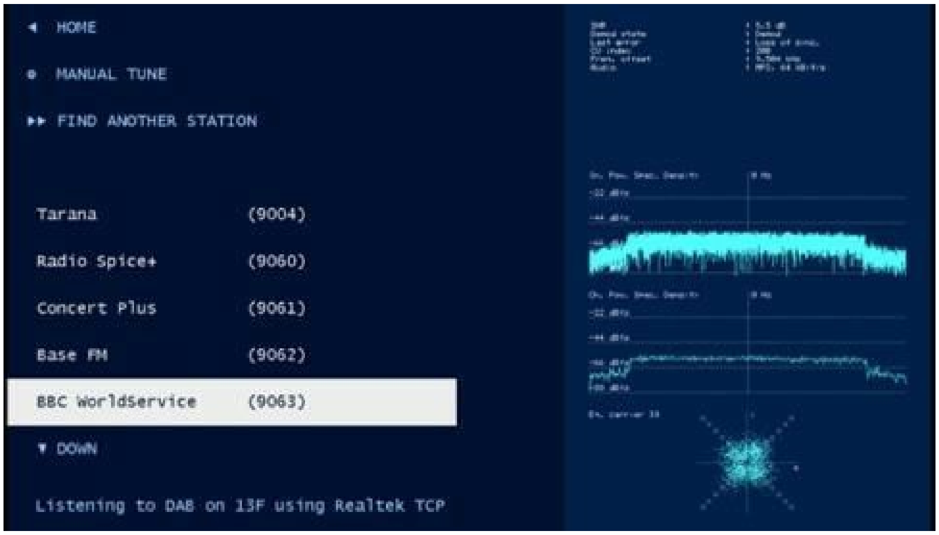
https://play.google.com/store/apps/details?id=marto.androsdr2



WAVESINK

Wavesink is an Android program similar to SDR Touch, but it is restricted to decoding Broadcast FM, DRM and DAB/DAB+ audio only. Audio quality on either mode is excellent.

https://play.google.com/store/apps/details?id=de.ses.wavesink

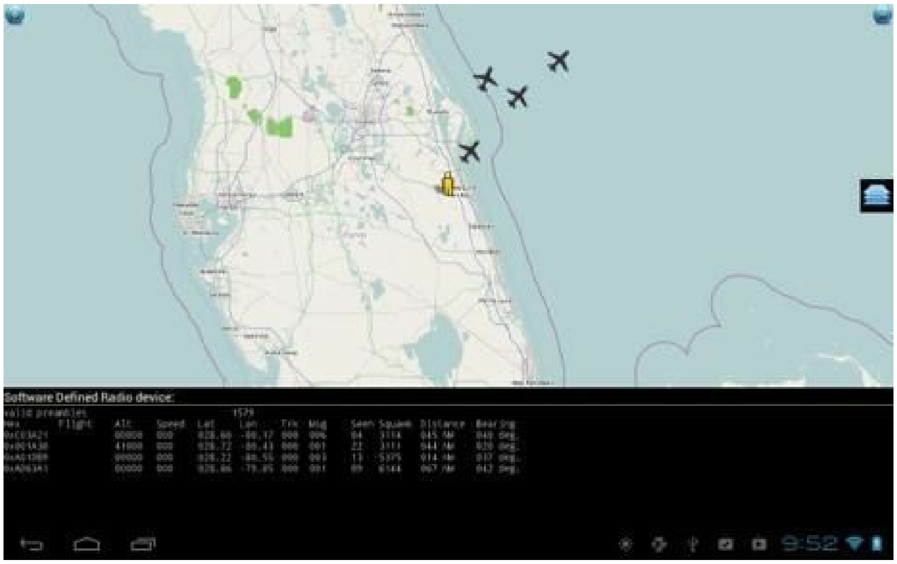


```
Tarana              (9004)
Radio Spice+        (9060)
Concert Plus        (9061)
Base FM             (9062)
BBC WorldService    (9063)

Listening to DAB on 13F using Realtek TCP
```

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.

https://play.google.com/store/apps/details?id=com.wilsonae.android.usbserial



```
Software Defined Radio device:
valid preambles:                 1579
Hex       Flight    Alt    Speed  Lat     Lon     Trk  Mag    Seen Squawk  Distance  Bearing
0xC03A21            00000  000    028.60  -80.17  000  006    04   3114    045 NM    040 deg.
0x001A38            41000  000    028.72  -80.43  000  001    22   3111    044 NM    028 deg.
0xAC1D89            00000  000    028.22  -80.55  000  003    13   5375    014 NM    037 deg.
0xA0634I            00000  000    028.86  -79.95  000  001    09   6144    067 NM    042 deg.
```

SDR WEATHER

This app receives weather radio and Emergency Alert System (EAS) alerts using the RTL-SDR.

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.

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.

GNU RADIO INTRODUCTION

GNU Radio is an advanced graphical programming tool for digital signal processing (DSP). It can interface to the RTL-SDR and perform real time decoding.

This guide is intended to teach the basics of GNU Radio by helping you install GNU Radio and create your first GNU Radio RTL-SDR program. Note that the RTL-SDR Linux drivers need to be installed for GNU Radio to work with the RTL-SDR. See Installing RTL-SDR Drivers on Linux. There are several Linux distributions and Live DVDs that now have GNU Radio preinstalled. See the GNU Radio Live DVDs section for more information. To install GNU Radio on a version of Linux that does not have it preinstalled follow these instructions.

```
sudo apt-get update
sudo apt-get install git
sudo apt-get install libasound2-dev
git clone git://github.com/pybombs/pybombs
cd pybombs
sudo ./pybombs install gnuradio
sudo ./pybombs install gr-osmosdr
sudo ldconfig
```

After the install gnuradio line you will be presented with a few pre-installation options. Unless you know what you are doing you can use the default options for all the options except the prefix, which we highly recommend setting as `Prefix=/usr/local`. To use the default options simply press enter at each line.

The installation process will take a while (1-2 hours) to install depending on your PC and internet connection speeds. Important Note: Do not run this installation process on an embedded device running Linux on an SDCard. The amount of writes GNU Radio does during the installation process can destroy the card.

Start GNU Radio Companion by typing `gnuradio-companion` at a terminal window.

If for some reason you just can’t get compilation from source to work, you can always use “apt-get install gnuradio”, however this will usually install a very old version of GNU Radio which may not be compatible with some newer GNU Radio programs.

If you happen to be running GNU Radio in a 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.

GNU RADIO PROGRAM: FFT DISPLAY

The first program we’ll make is a simple FFT RF spectrum display using GNU Radio.

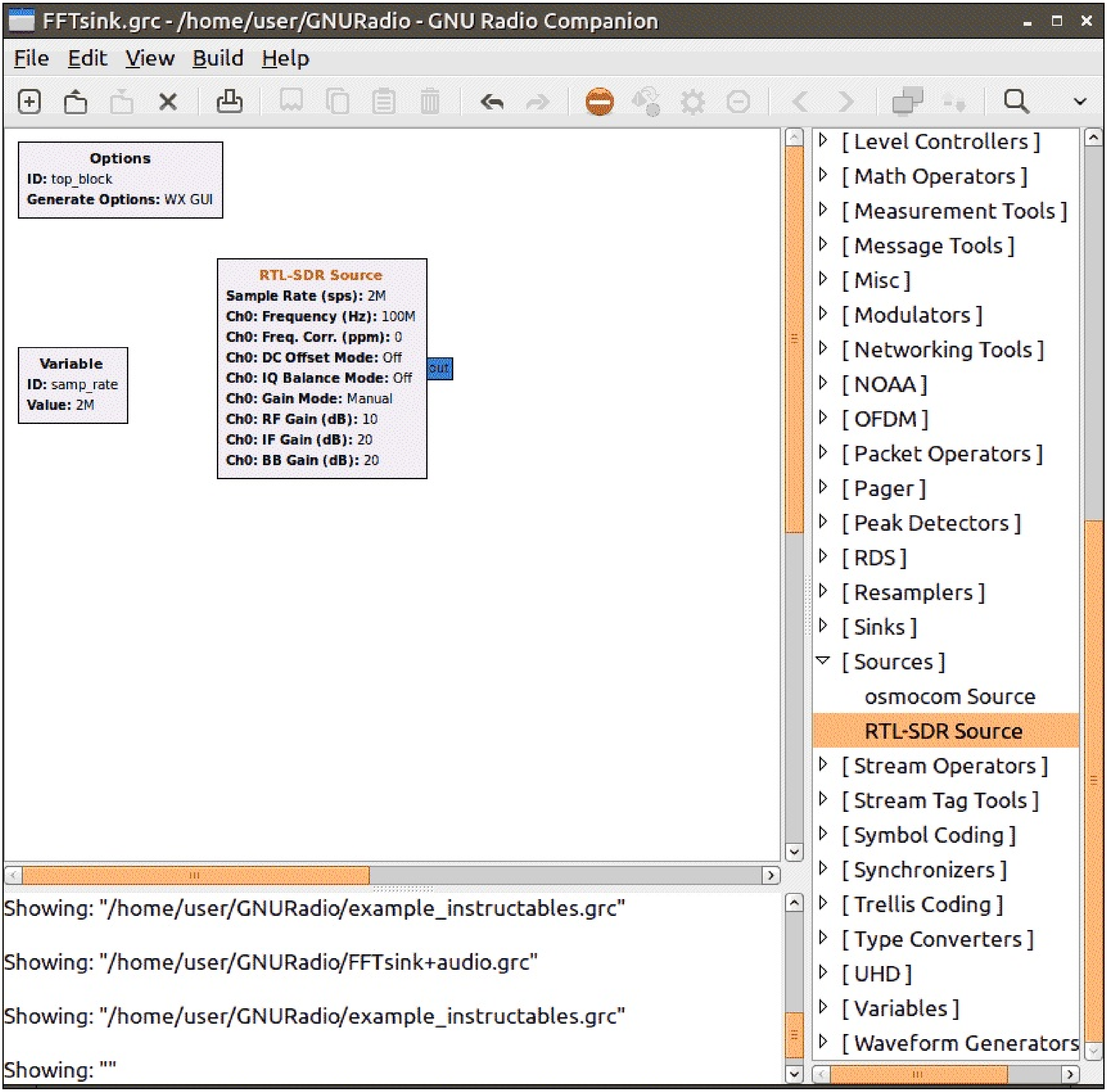
1. First open GNU-Radio Companion by typing into a terminal

```
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.

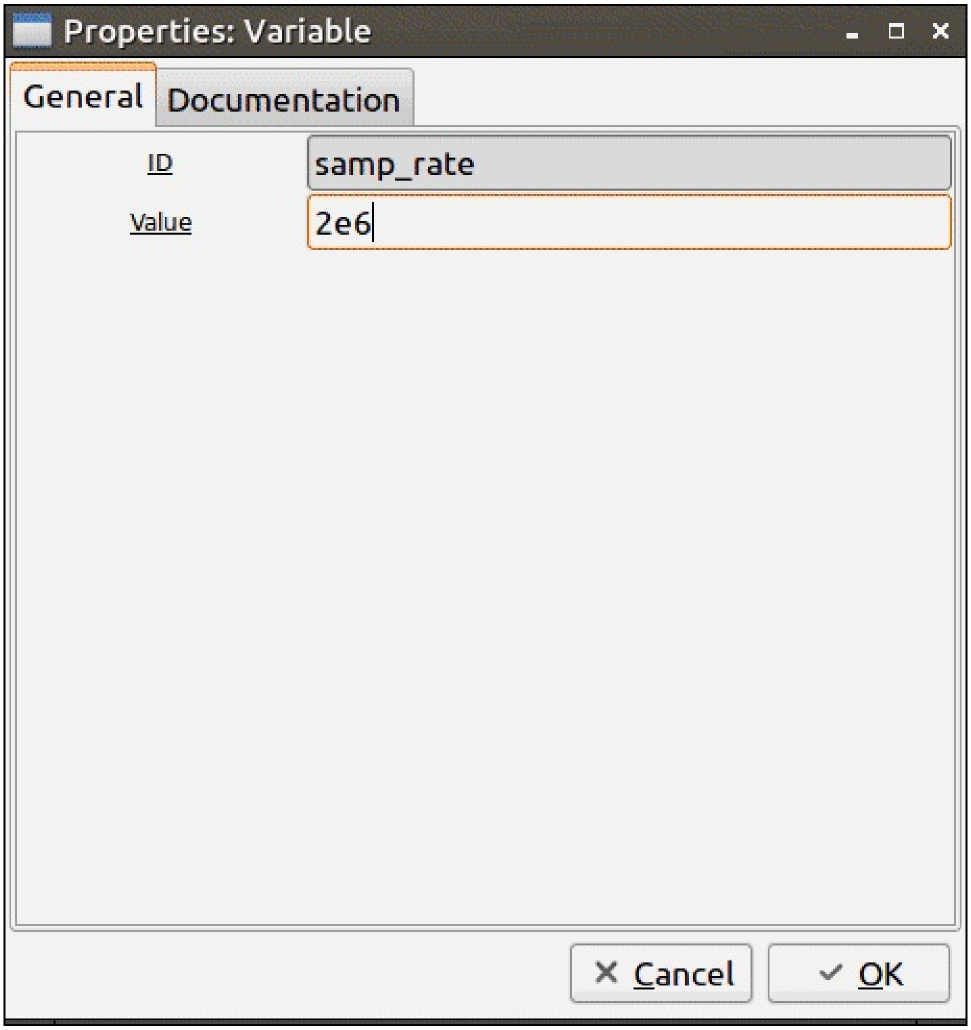


Options
ID: top_block
Generate Options: WX GUI

Variable
ID: samp_rate
Value: 2M

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

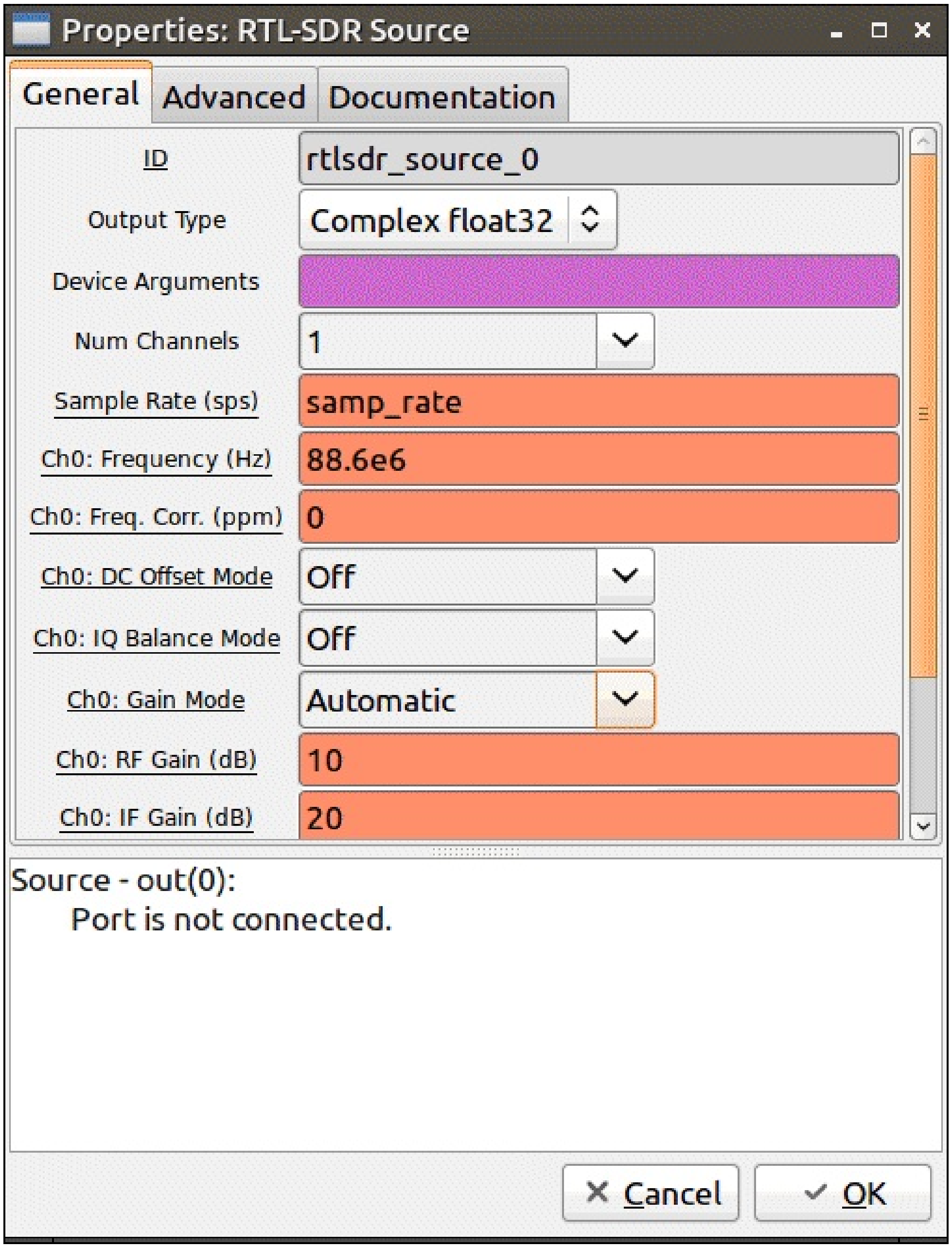
```
Showing: "/home/user/GNURadio/example_instructables.grc"

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 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) if required.



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

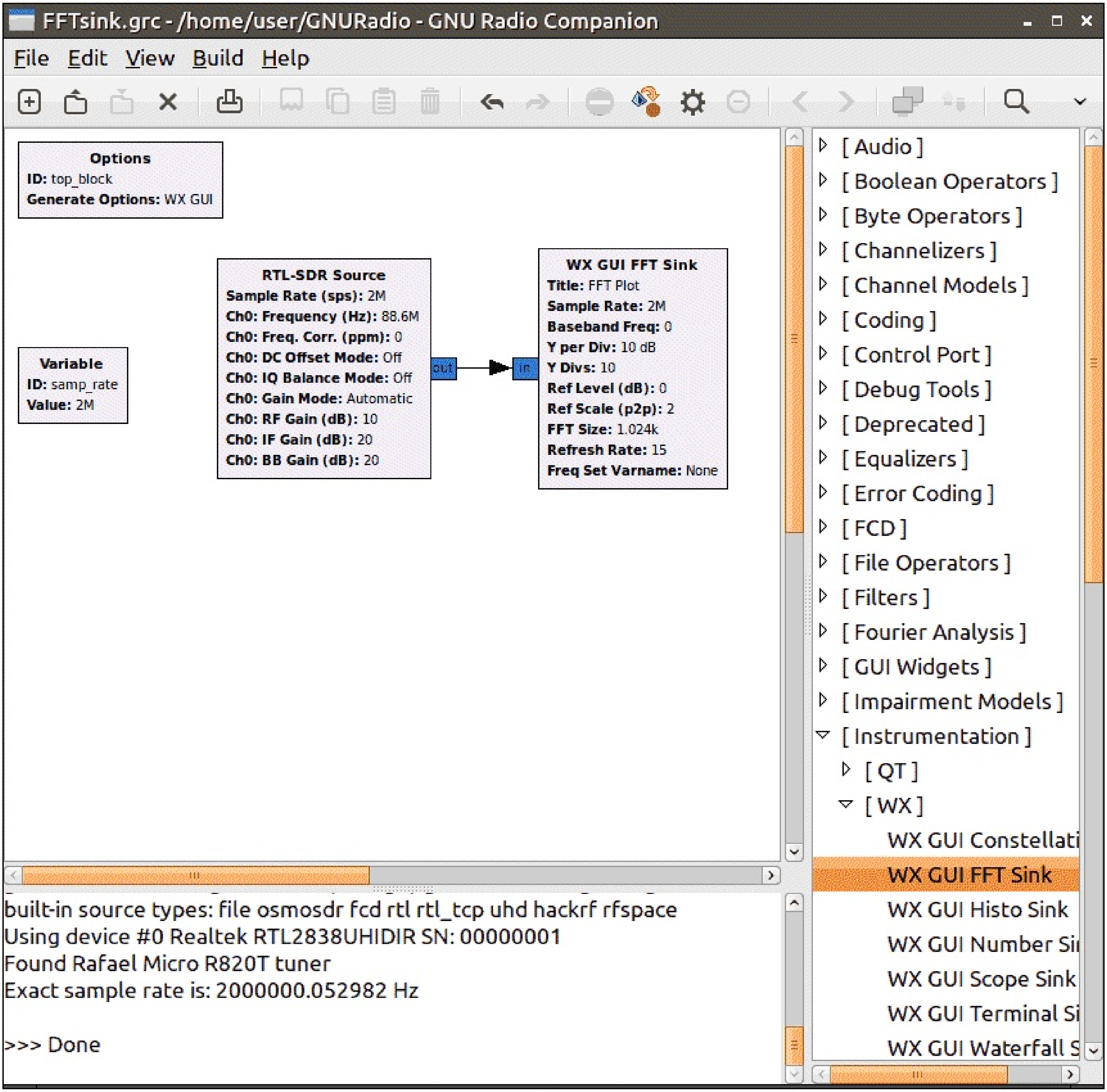
Ch0: RF Gain (dB): 10

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

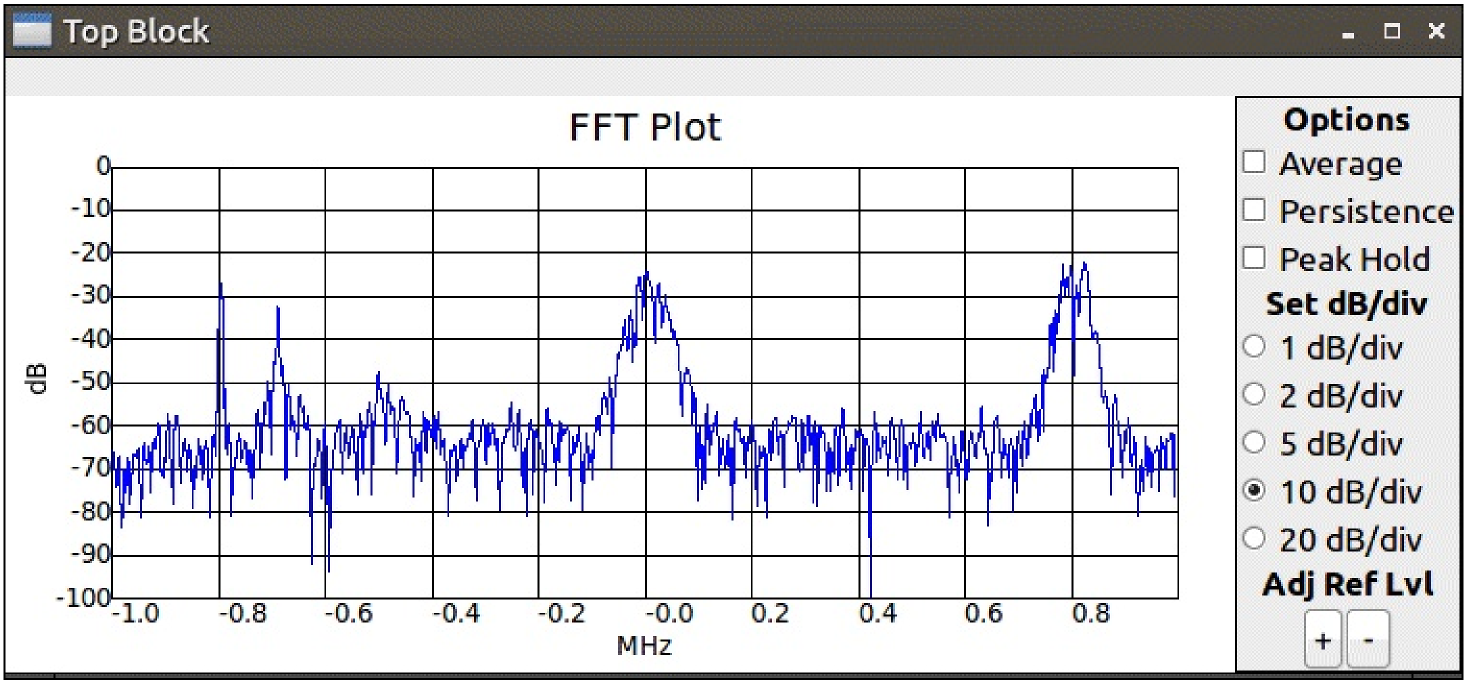
RTL-SDR Source
Sample Rate (sps): 2M
Ch0: Frequency (Hz): 88.6M
Ch0: Freq. Corr. (ppm): 0
Ch0: DC Offset Mode: Off
Ch0: IQ Balance Mode: Off
Ch0: Gain Mode: Automatic
Ch0: RF Gain (dB): 10
Ch0: IF Gain (dB): 20
Ch0: BB Gain (dB): 20

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 .

The result will be a live RF spectrum plot with your chosen frequency in the centre.



Options

□ Average
□ Persistence
□ Peak Hold

Set dB/div

○ 1 dB/div
○ 2 dB/div
○ 5 dB/div
● 10 dB/div
○ 20 dB/div

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. Go to file save as and save the new file as 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

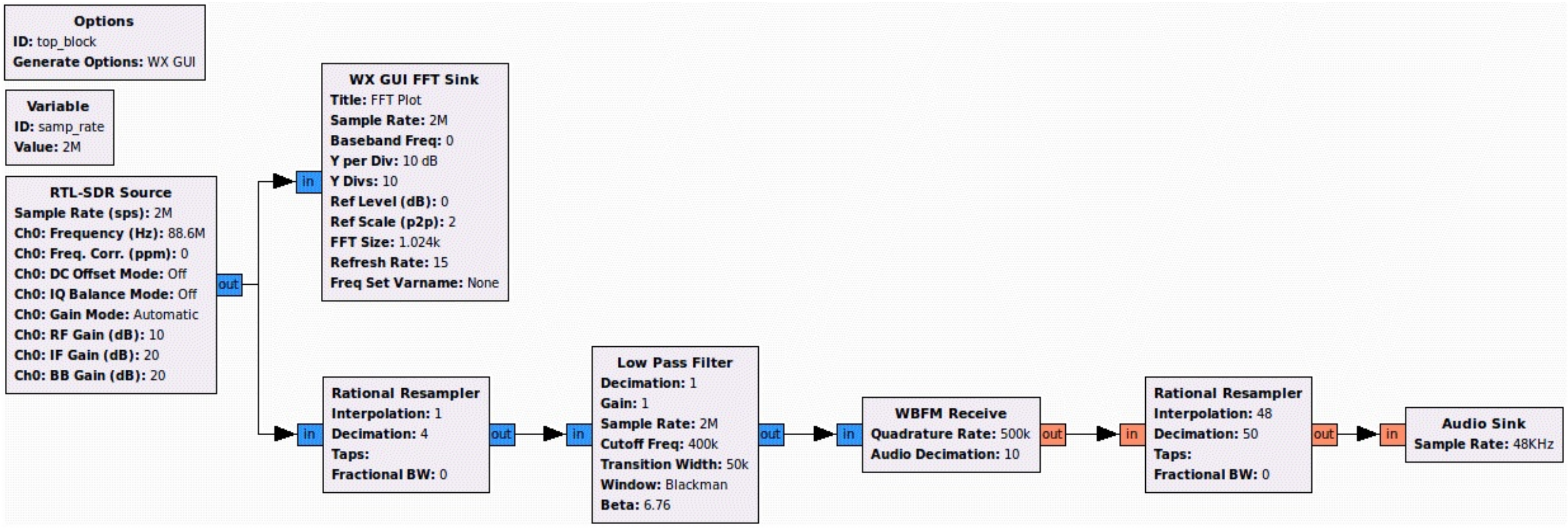
4. Add in a 'WBFM Receive' Block. Connect it to the Low Pass Filter. Set the parameters as follows.
   - Quadrature Rate = 500e6
   - Audio Decimation = 10

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

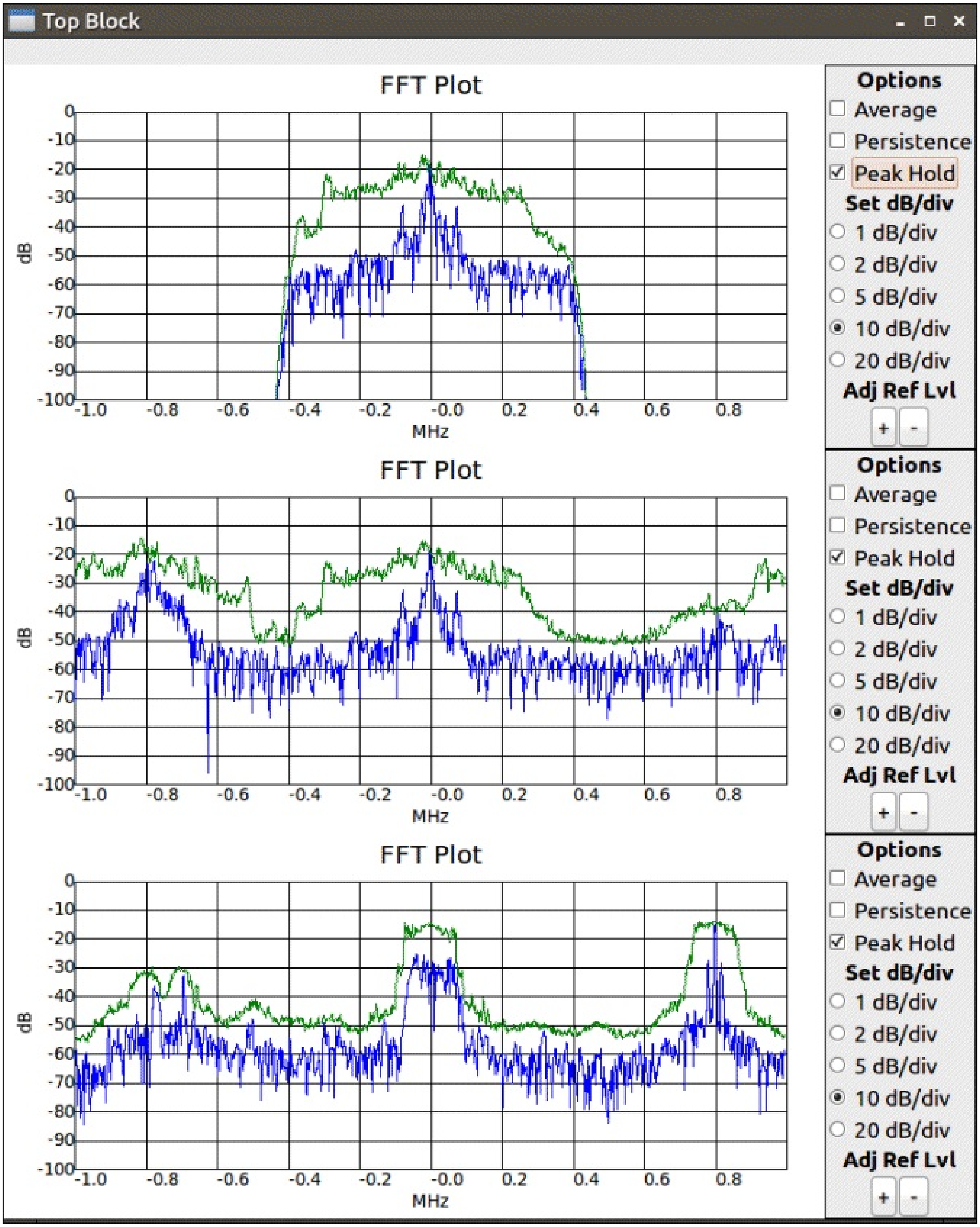
The first thing that happens in the flow graph is that the signal from the RTL-SDR is decimated by four times. Decimation is a way to get more bits out of an ADC. Essentially, by decimating by four, we reduce our 2 Msps sample rate to 0.5 Msps, but gain one extra bit on our ADC. We need to be careful 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 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 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

Options
☐ Average
☐ Persistence
☑ Peak Hold
Set dB/div
○ 1 dB/div
○ 2 dB/div
○ 5 dB/div
● 10 dB/div
○ 20 dB/div

Options
☐ Average
☐ Persistence
☑ Peak Hold
Set dB/div
○ 1 dB/div
○ 2 dB/div
○ 5 dB/div
● 10 dB/div
○ 20 dB/div

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 GitHub using

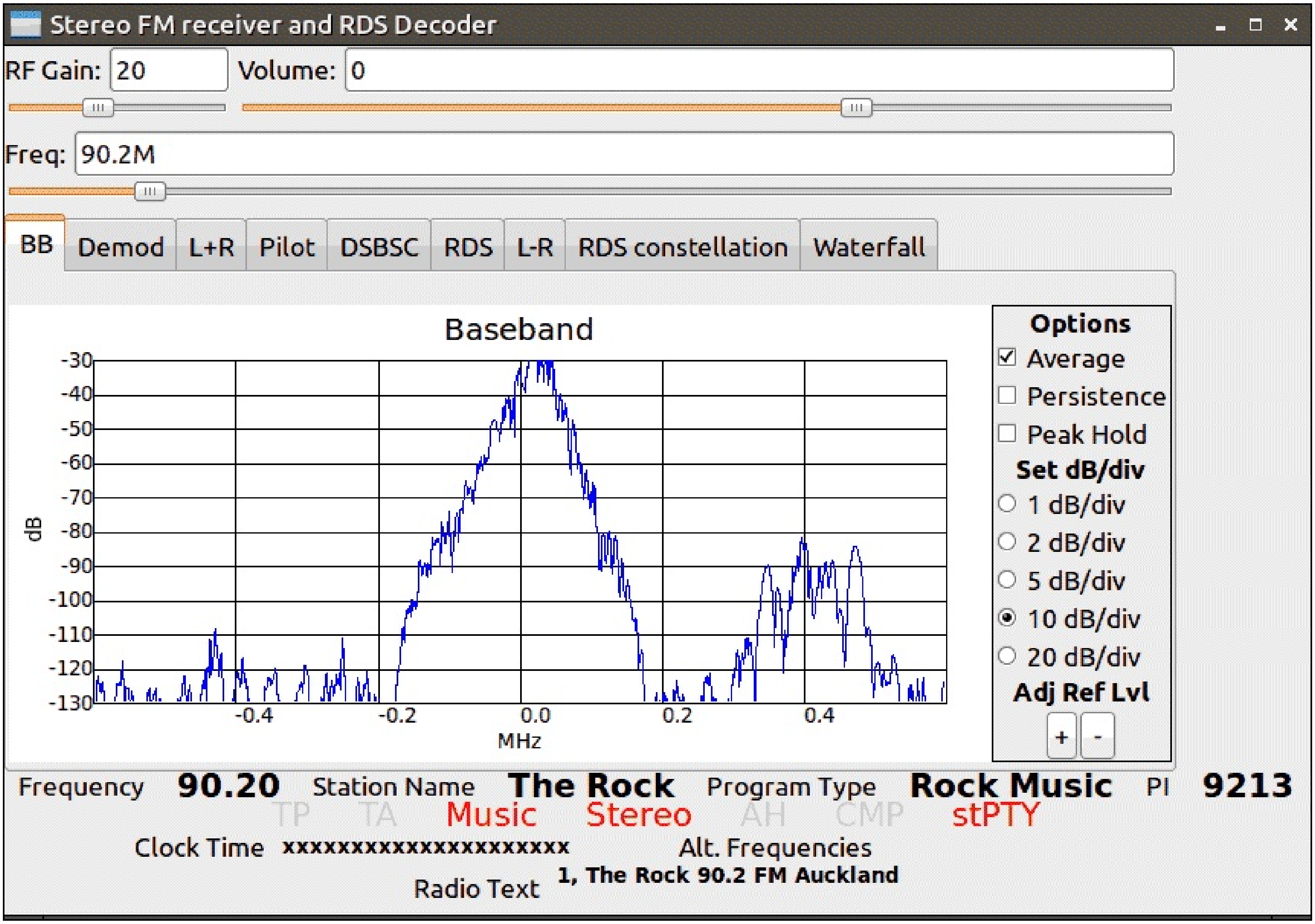
```
git clone git://git.github.com/bastibl/gr-rds
```

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 or running one of the following files

```
cd ..
cd apps
gnuradio-companion rds_rx.grc --OR-- python rds_rx.py
```



Stereo FM receiver and RDS Decoder

RF Gain: 20 Volume: 0

Freq: 90.2M

Options
☑ Average
☐ Persistence
☐ Peak Hold
Set dB/div
○ 1 dB/div
○ 2 dB/div
○ 5 dB/div
● 10 dB/div
○ 20 dB/div

Frequency **90.20** Station Name **The Rock** Program Type **Rock Music** PI **9213**

TP TA Music Stereo AH CMP stPTY

Clock Time xxxxxxxxxxxxxxxxxxxx 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

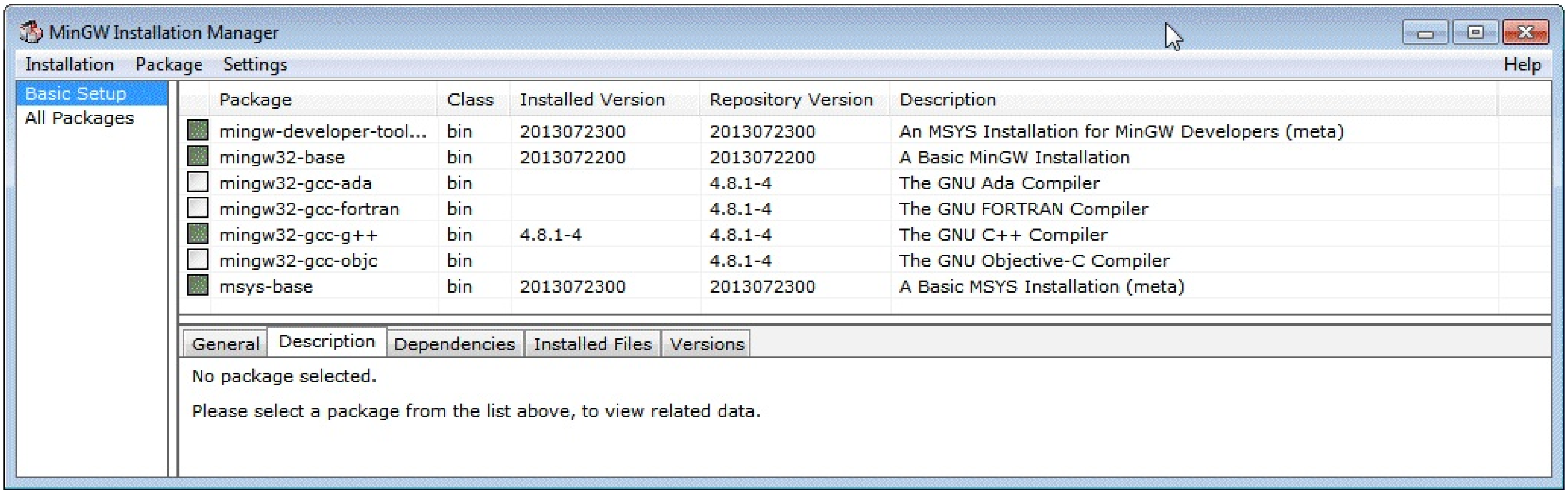
COMPILING RTL-SDR DRIVERS FROM SOURCE ON WINDOWS

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 if you desire. 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 GitHub page. Here is a short tutorial on compiling the RTL-SDR drivers on Windows.

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 have MingW set up on their system as well. MingW can be downloaded from http://www.mingw.org/. Make sure you set the MingW path correctly if it does not set automatically.

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.

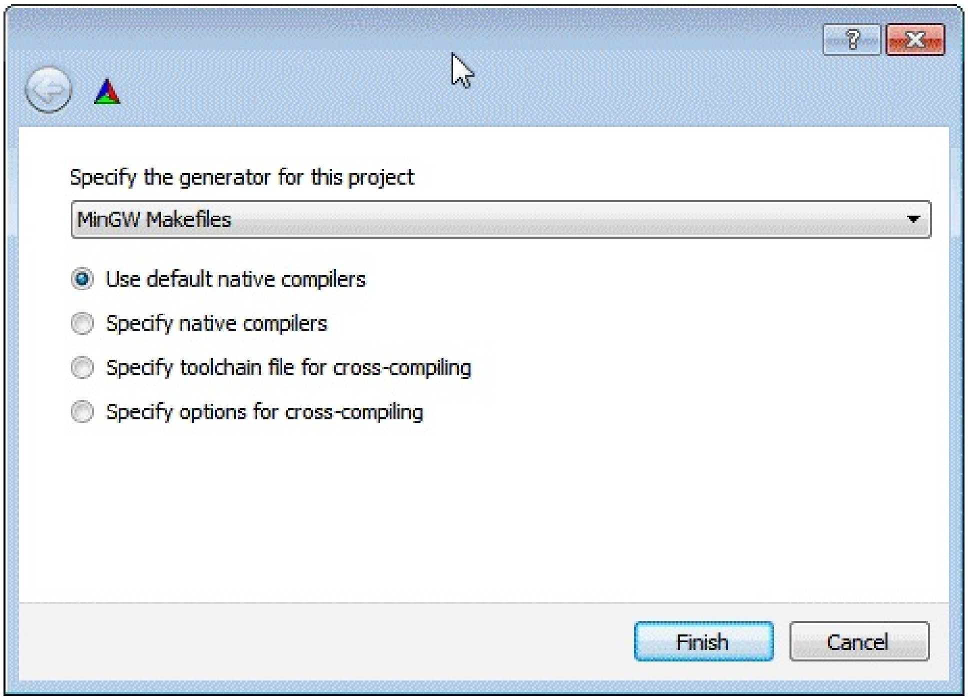
6. Download the rtl-sdr source from git at git://git.osmocom.org/rtl-sdr or https://github.com/steve-m/librtlsdr either by downloading the zip file or doing a git clone. Extract to a directory like C:\rtl-sdr.

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.

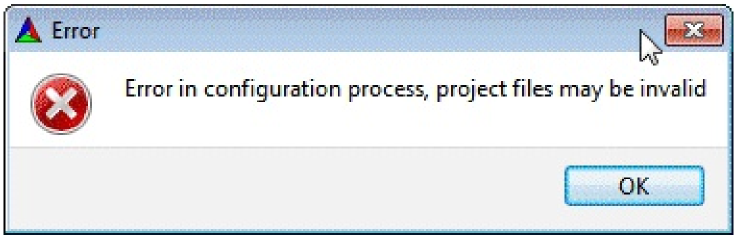


Specify the generator for this project

MinGW Makefiles

- ● Use default native compilers
- ○ Specify native compilers
- ○ Specify toolchain file for cross-compiling
- ○ Specify options for cross-compiling

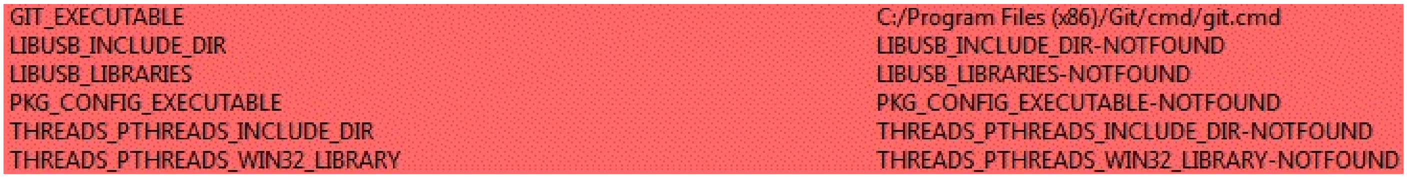
11. You may receive an error, but don't worry about it yet.



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

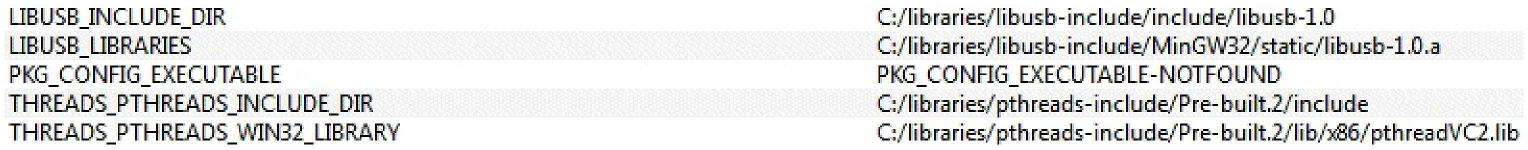
13. Set LIBUSB_INCLUDE_DIR to c:\libraries\libusb-include\include\libusb-1.0\

14. Set LIBUSB_LIBRARIES to c:\libraries\libusb-include\MinGW32\static\libusb-1.0.a

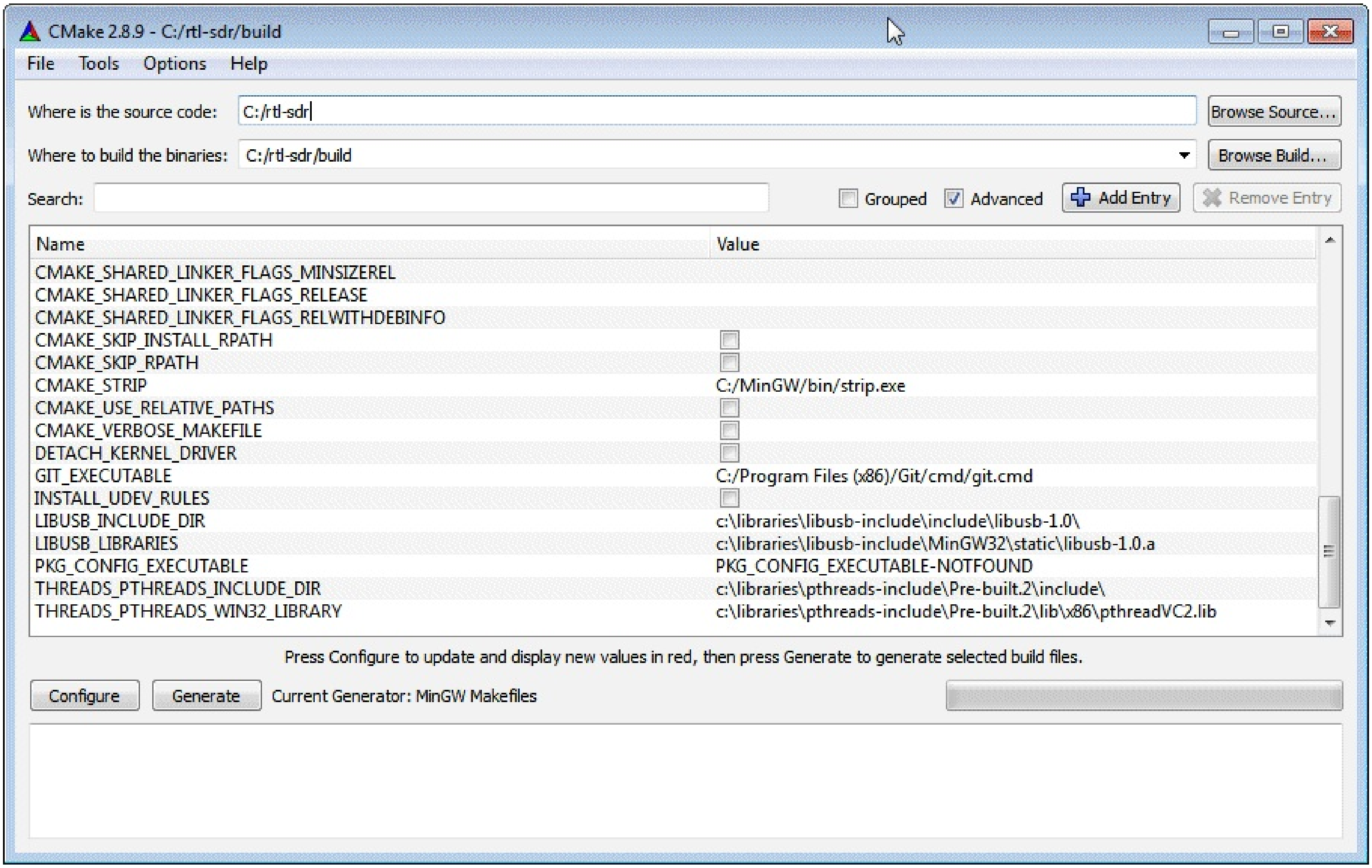
15. Set THREADS_PTHREADS_INCLUDE_DIR to c:\libraries\pthreads-include\Pre-

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

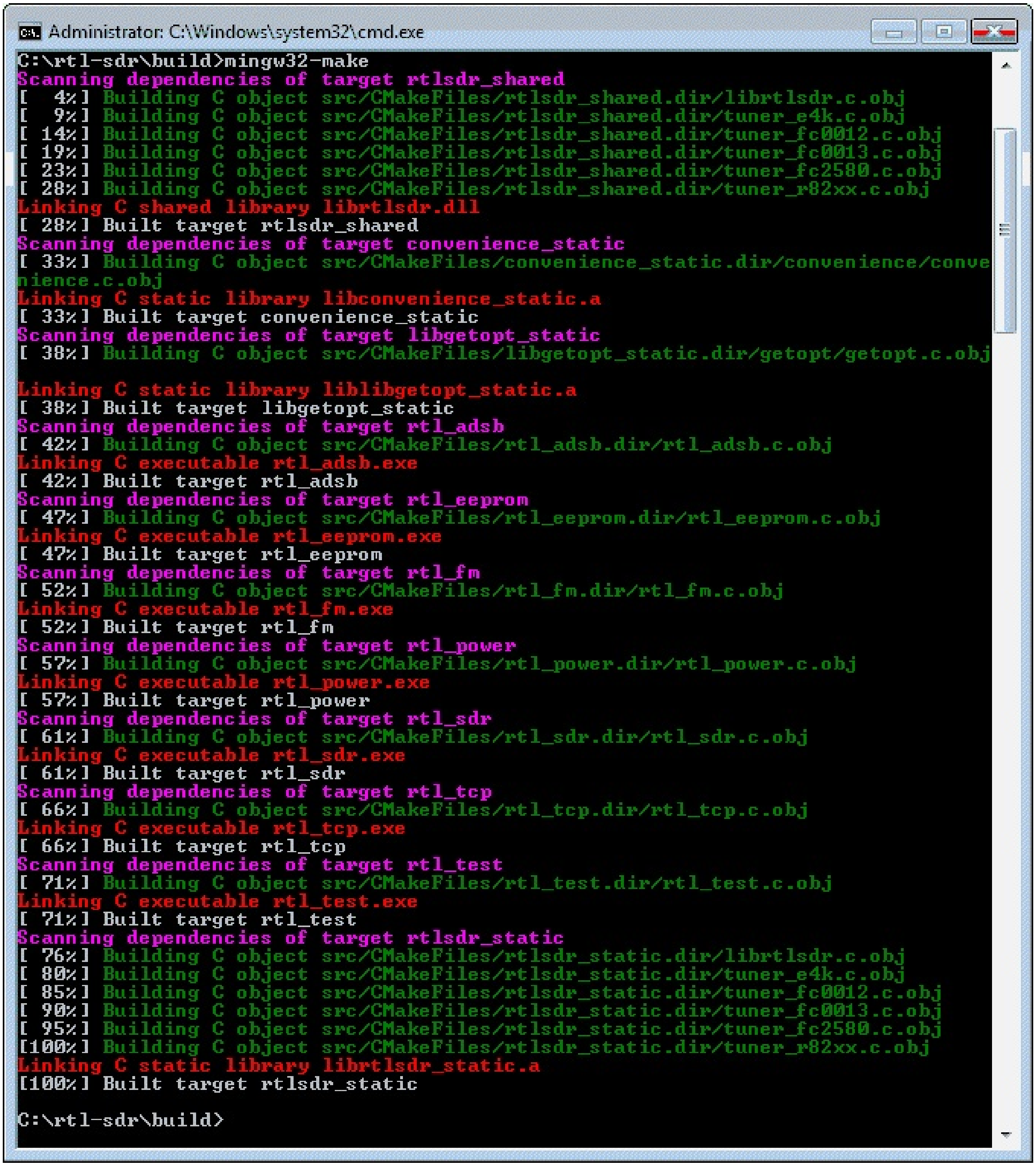
Name	Value
CMAKE_SHARED_LINKER_FLAGS_MINSIZEREL	
CMAKE_SHARED_LINKER_FLAGS_RELEASE	
CMAKE_SHARED_LINKER_FLAGS_RELWITHDEBINFO	
CMAKE_SKIP_INSTALL_RPATH	☐
CMAKE_SKIP_RPATH	☐
CMAKE_STRIP	C:/MinGW/bin/strip.exe
CMAKE_USE_RELATIVE_PATHS	☐
CMAKE_VERBOSE_MAKEFILE	☐
DETACH_KERNEL_DRIVER	☐
GIT_EXECUTABLE	C:/Program 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>
```

librtlsdr.dll will be found in C:\rtl-sdr\build\src

COMPILING USING VISUAL C++ 2010

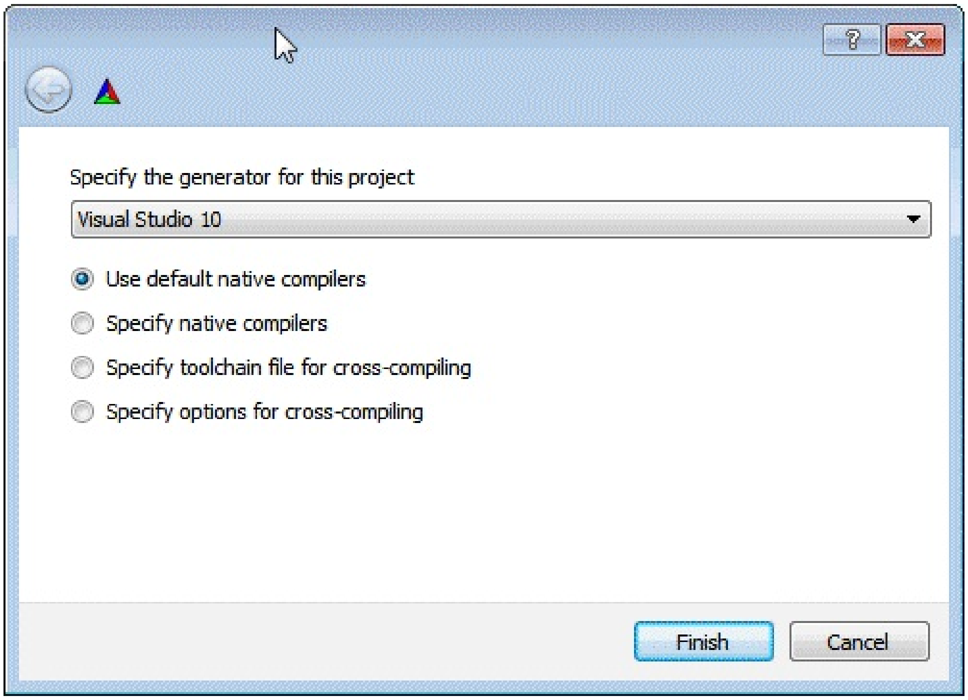
The RTL-SDR drivers can also be compiled using Visual Studio. Compiling this way is more portable as more people will be likely to have the required runtimes as they are usually included with windows.

1. Install 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 combining 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 instead.

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. 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 be found in C:\rtl-sdr\build\src\Debug

PANADAPTER GUIDE

A panadapter is a device that connects to a traditional analogue hardware radio and displays a visualization of the spectrum through an RF spectrum and waterfall display. Most RTL-SDR GUI receivers show 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 radio and the visual RF display common to software defined radios.

Creating an RTL-SDR panadapter is 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 output and some will have it as an available out port. Most radios will also allow you to control it via GUI software like Omnirig HDSDR using 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.

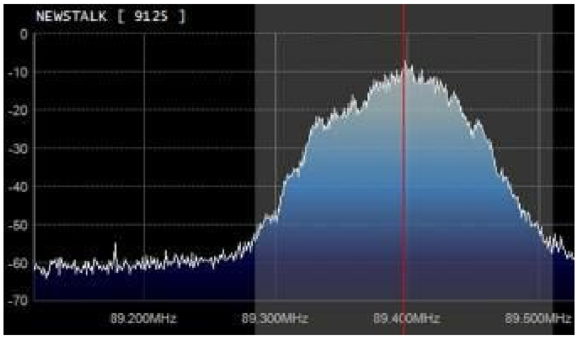
A good tutorial on doing this for the Yaesu FT-857 can be found on this YouTube video https://www.youtube.com/watch?v=tshlXgarnBQ#t=164. Another video showing the performance of the FT-857 with RTL-SDR panadapter can be found here https://www.youtube.com/watch?v=wqcGACGPtrM**.** An excellent pdf showing the modifications required for many Yaesu radios is here http://www.radioamatoripeligni.it/i6ibe/pdf/panadapter.pdf (note in Italian).

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.

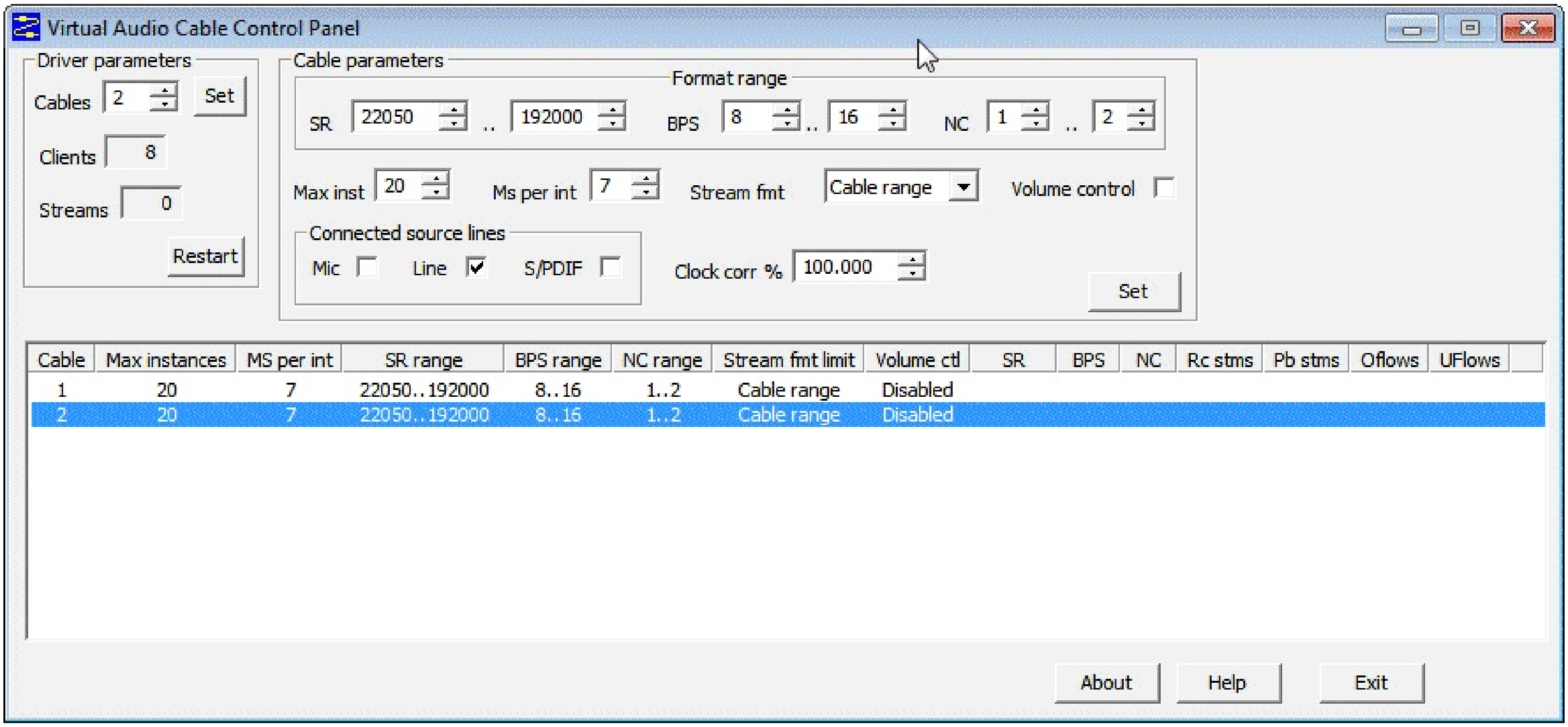


Another program often used for RDS decoding is RDS Spy. RDS Spy has better sensitivity and analysis options so it is better for dxing (distance receiving - very long range reception). RDS Spy can be downloaded from http://rdsspy.com/. Unfortunately RDS Spy will not work with SDR# due to SDR#'s implementation. To use RDS Spy, it must be used with HDSDR and you must use the Virtual Audio Cable audio pipe, not VB Cable. The setup process is also a little involved and is shown below.

1. You will need to have downloaded and installed Virtual Audio Cable first. Since the trial version supports up to three cables it will be sufficient for this task. The watermark audio played will not pose a major problem. You will also need to have installed HDSDR and have set it up working with the RTL-SDR.

2. Create two virtual audio cables. 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. Be sure to click on the Set button after changing the options sample rate to make it permanent.



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

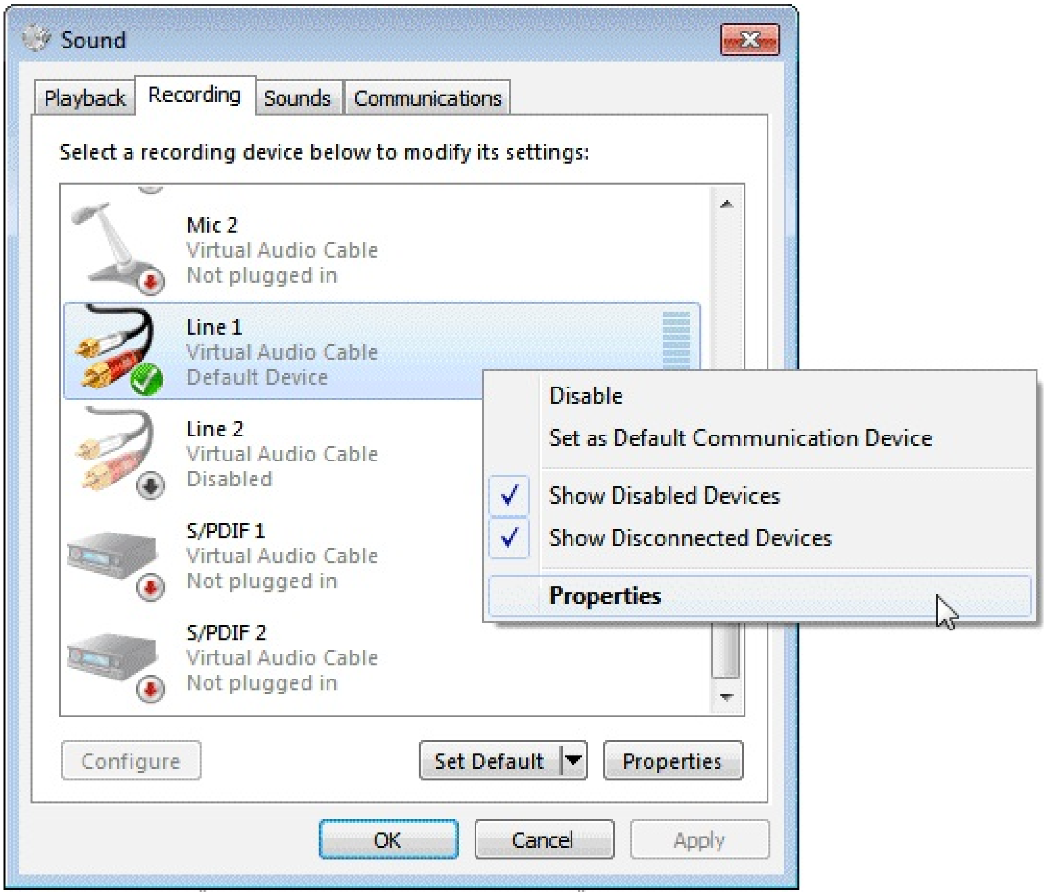
Mic ☐ Line ☑ S/PDIF ☐

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 and 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 the Recording **AND** Playback tabs.



Select a recording device below to modify its settings:

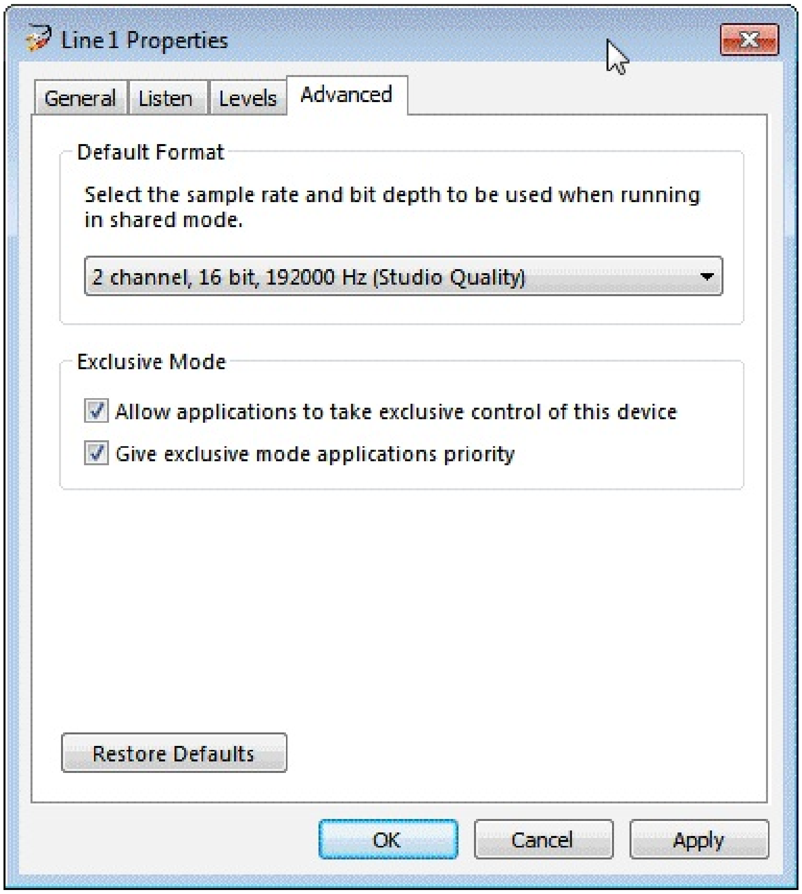
Mic 2
Virtual Audio Cable
Not plugged in

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.

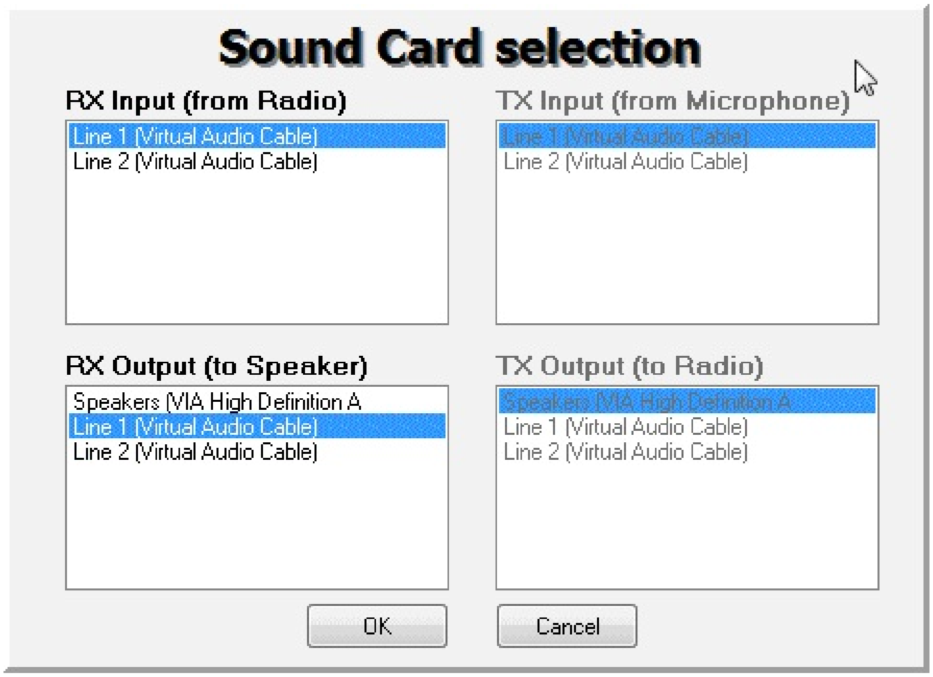
2 channel, 16 bit, 192000 Hz (Studio Quality)

Exclusive Mode

☑ Allow applications to take exclusive control of this device

☑ Give exclusive mode applications priority

5. Open HDSDR.
6. Click the Soundcard button in the lower left corner and then select the first virtual audio cable device under RX Output (to Speaker).



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)

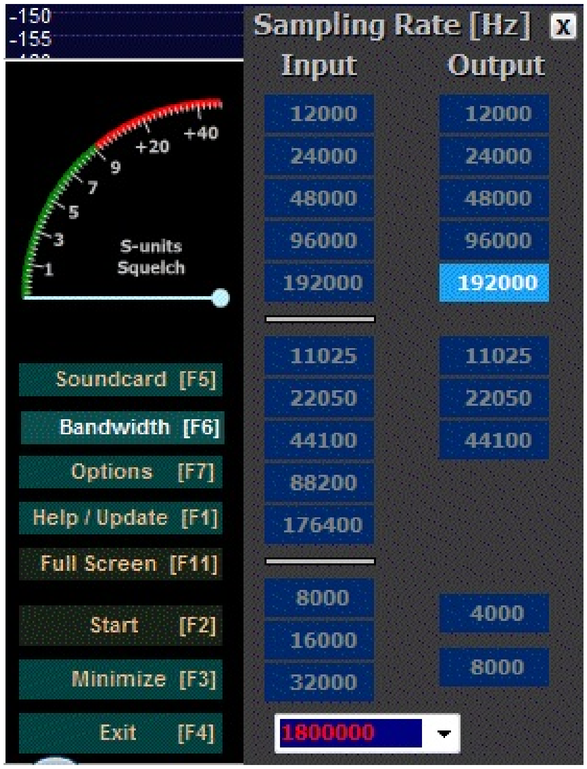
RX Output (to Speaker)

- Speakers (VIA High Definition A
- Line 1 (Virtual Audio Cable)
- Line 2 (Virtual Audio Cable)

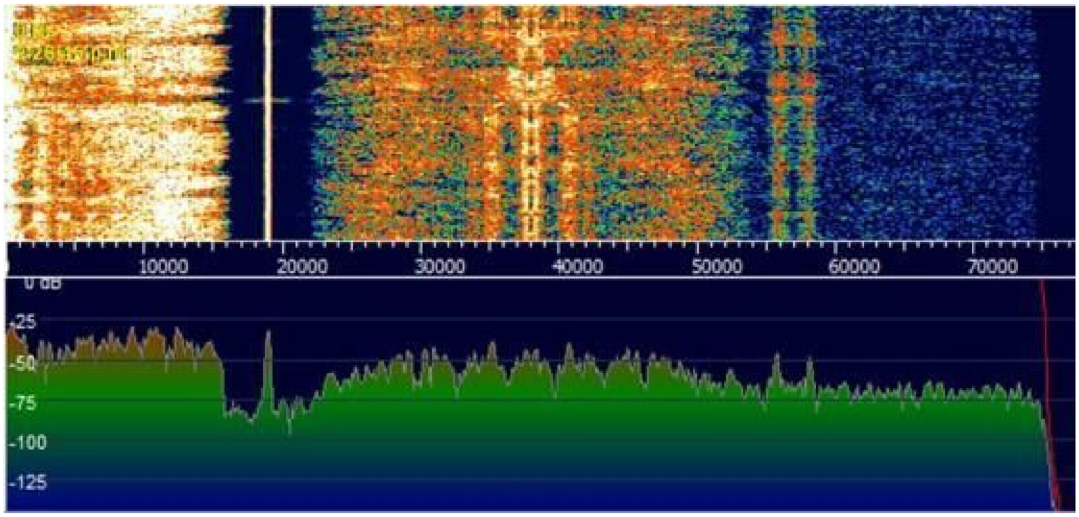
TX Output (to Radio)

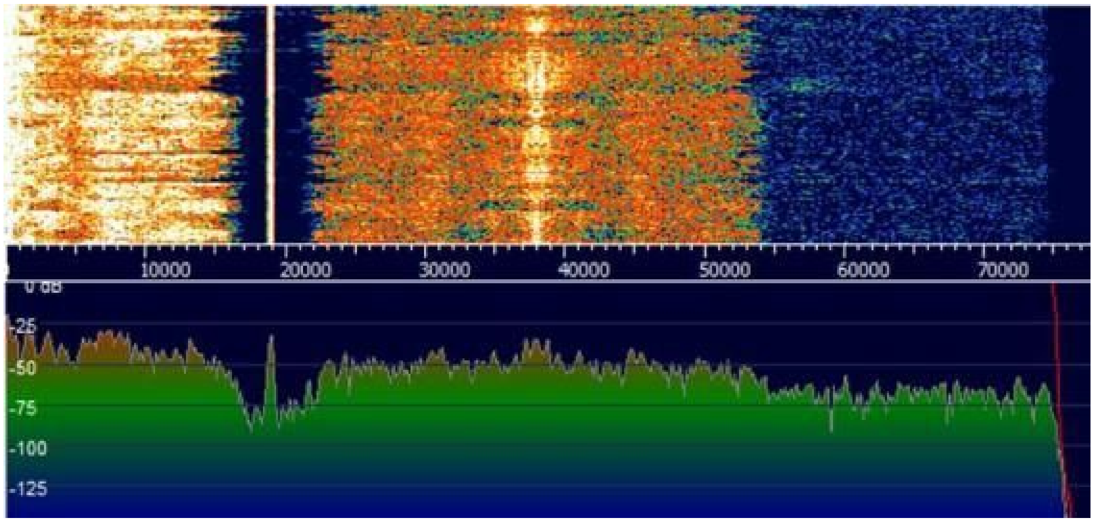
- Speakers (VIA High Definition A
- Line 1 (Virtual Audio Cable)
- Line 2 (Virtual Audio Cable)

7. Click the Bandwidth button in the lower left corner and then set the Output to 192000.



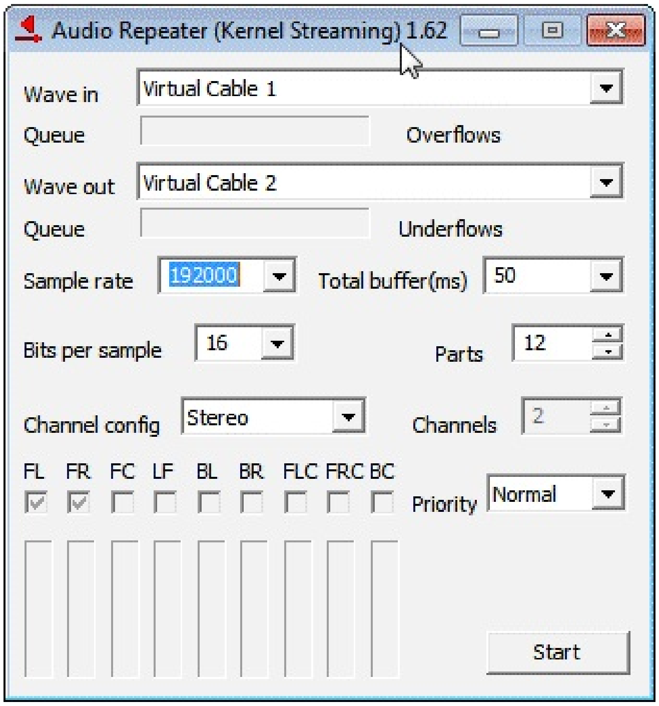
8. Now click Start in HDSDR and tune to a broadcast FM station that has RDS. You can check visually in HDSDRs audio spectrum to see if the station you are tuned to has an RDS signal. In the example screenshot below there is a thin signal on the right of the spectrum, this is the RDS signal. An example of an FM station without RDS is shown below it.





9. Go to **Start Menu -> Virtual Audio Cable -> Audio Repeater (Kernel Streaming).**

10. 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

Wave out Virtual Cable 2

Sample rate 192000

Total buffer(ms) 50

Bits per sample 16

Parts 12

Channel config Stereo

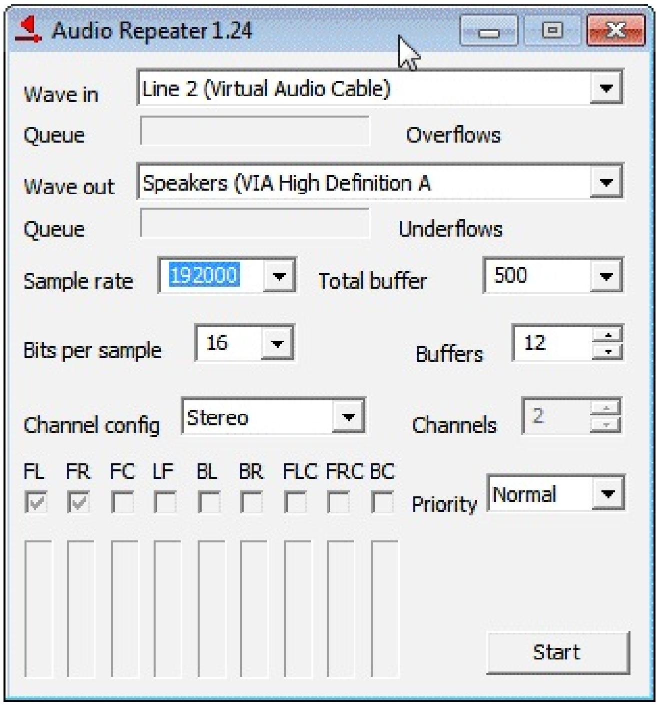
Channels 2

FL ☑ FR ☑ FC □ LF □ BL □ BR □ FLC □ FRC □ BC □

Priority Normal

11. OPTIONAL: Go to **Start Menu -> Virtual Audio Cable -> Audio Repeater (MME).**

12. 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 listen to the FM station audio.



Audio Repeater 1.24

Wave in Line 2 (Virtual Audio Cable)

Wave out Speakers (VIA High Definition A

Sample rate 192000

Total buffer 500

Bits per sample 16

Buffers 12

Channel config Stereo

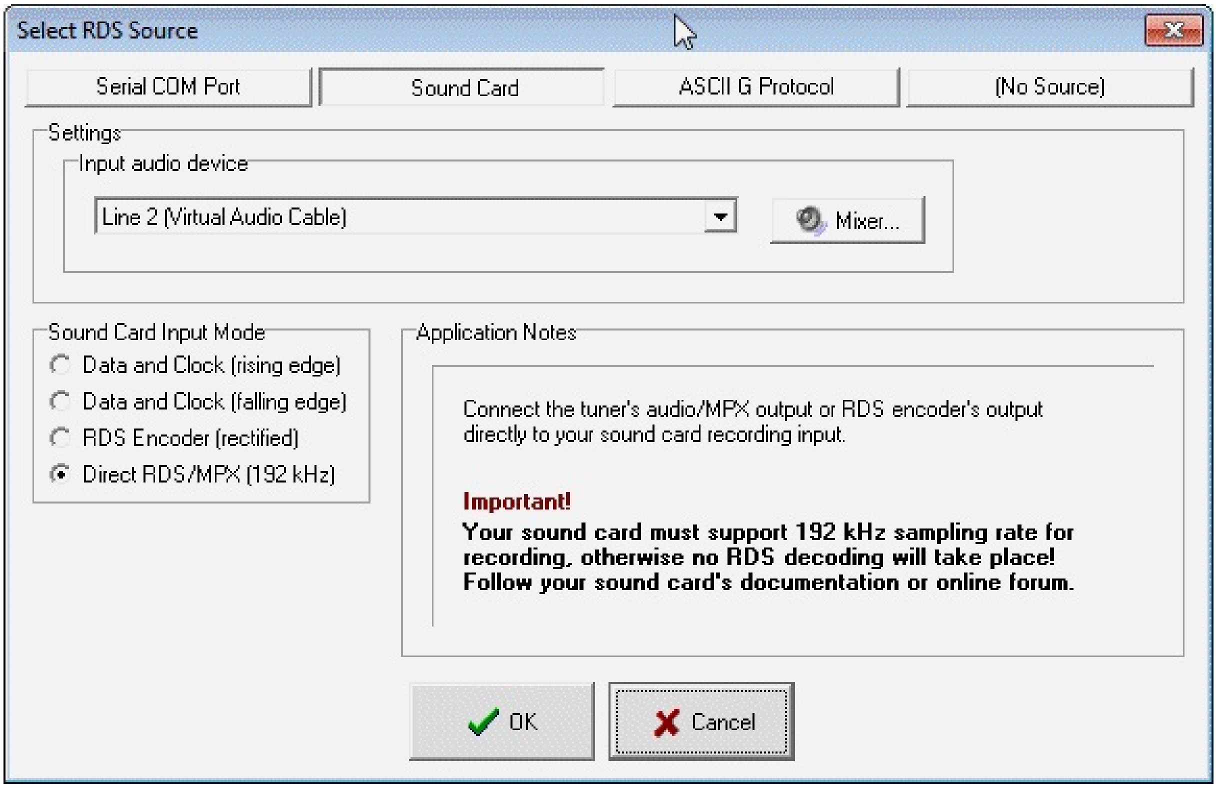
Channels 2

FL ☑ FR ☑ FC □ LF □ BL □ BR □ FLC □ FRC □ BC □

Priority Normal

13. Open RDS Spy and go to **Configure -> Select RDS Source**.

14. 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.



Input audio device: Line 2 (Virtual Audio Cable)

Sound Card Input Mode

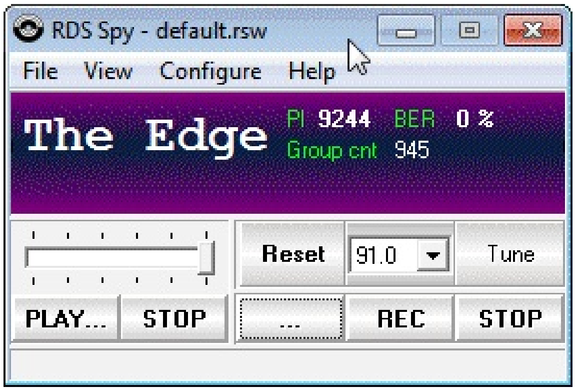
○ Data and Clock (rising edge)
○ Data and Clock (falling edge)
○ RDS Encoder (rectified)
● Direct RDS/MPX (192 kHz)

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.

15. RDS Spy should now be decoding RDS signals.



The Edge PI 9244 BER 0 %
Group cnt 945
91.0

For those interested there is also a GNU Radio RDS decoder implementation which is also extremely useful for learning how RDS is decoded. https://www.cgran.org/wiki/RDS.

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 like 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	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

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. AFSK1200 can be downloaded from http://sourceforge.net/projects/qtmm/. To use AFSK1200 follow the instructions below

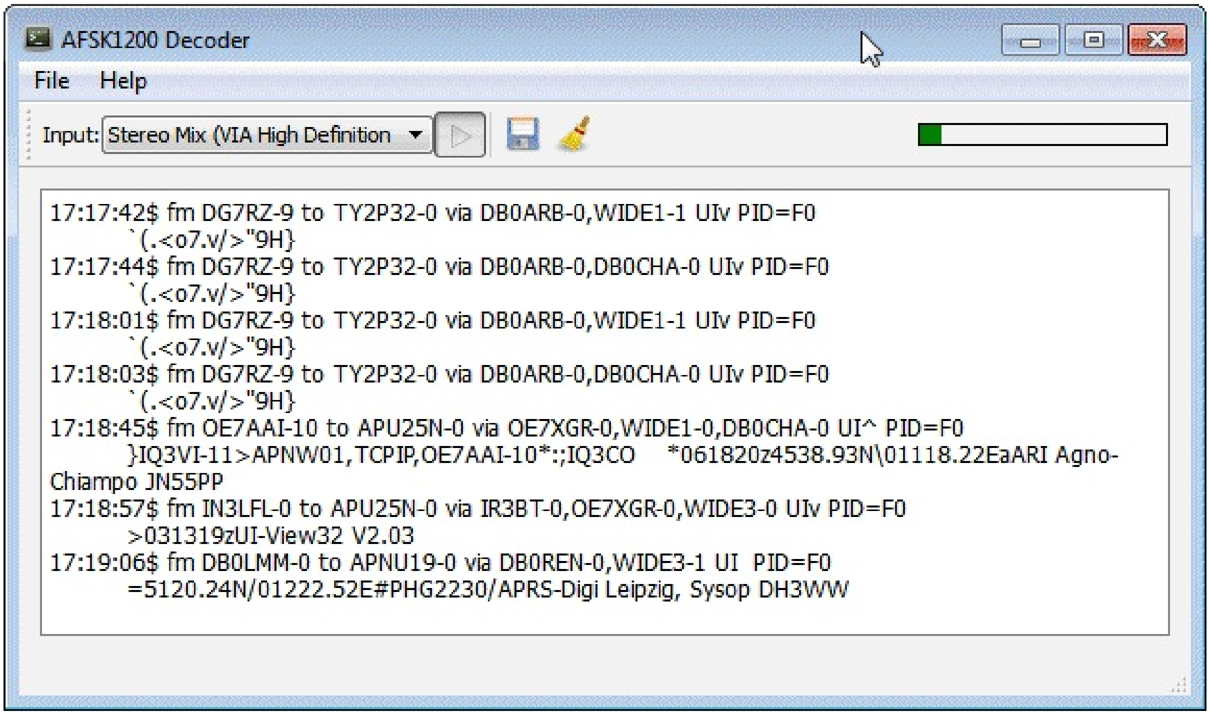
1. Open SDR# or your favourite general purpose SDR receiver and set the audio output to your preferred audio piping method.

2. Tune to a local APRS frequency at 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.

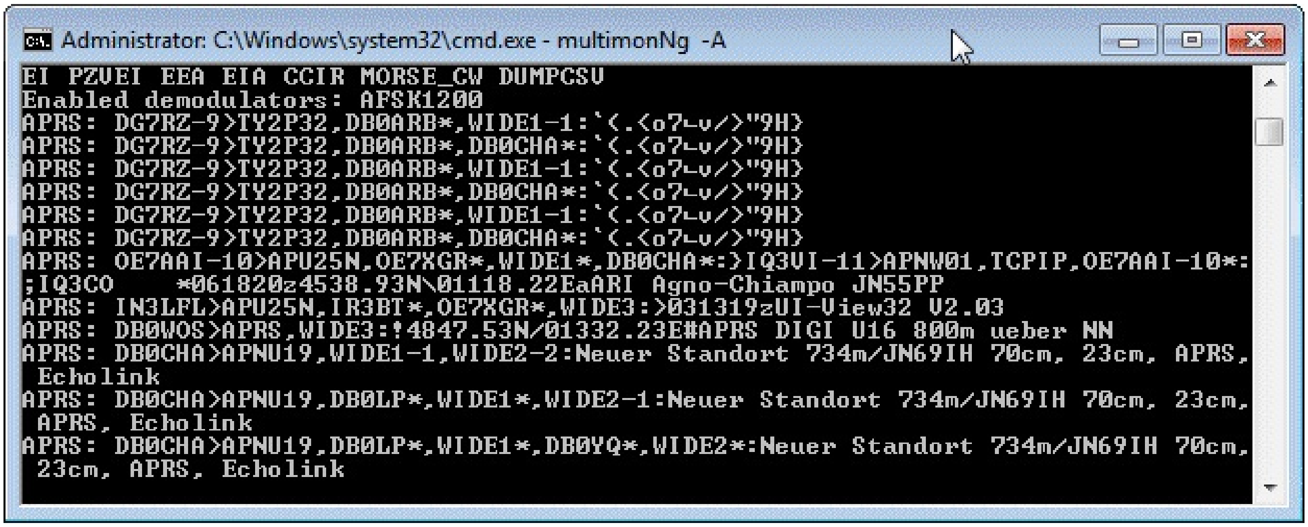
4. Press the Play button ▷ to begin listening.

Decoded message will show up in the main window.



```
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 can also be used to decode 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
```

For amateur radio hobbyists, the RTL-SDR can also be set up as an 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.

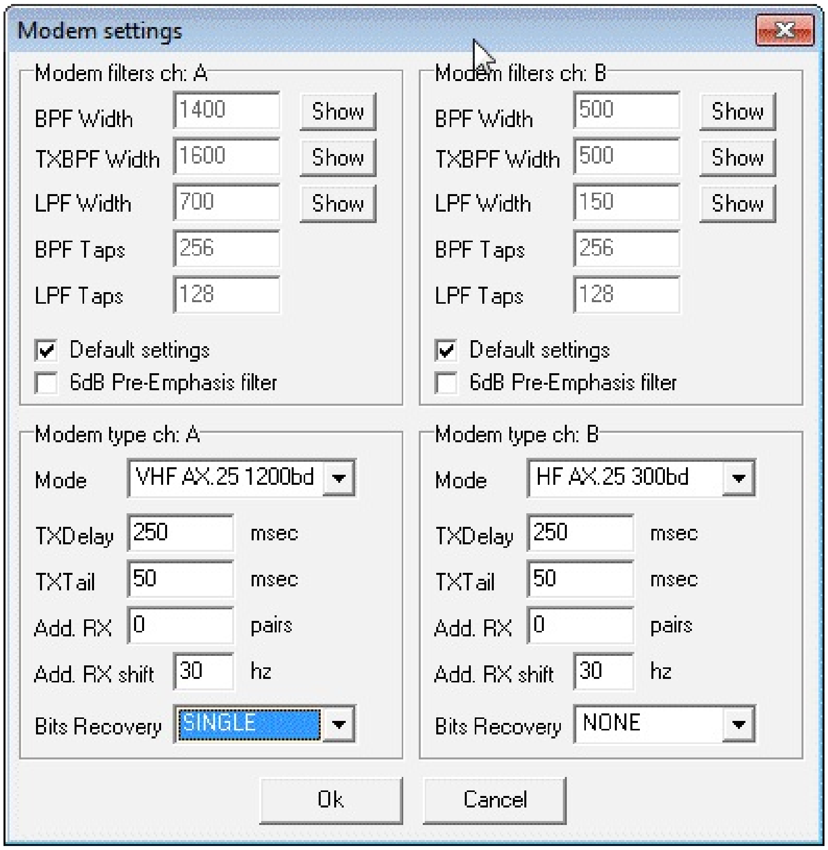
APRSISCE/32

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 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 general purpose 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

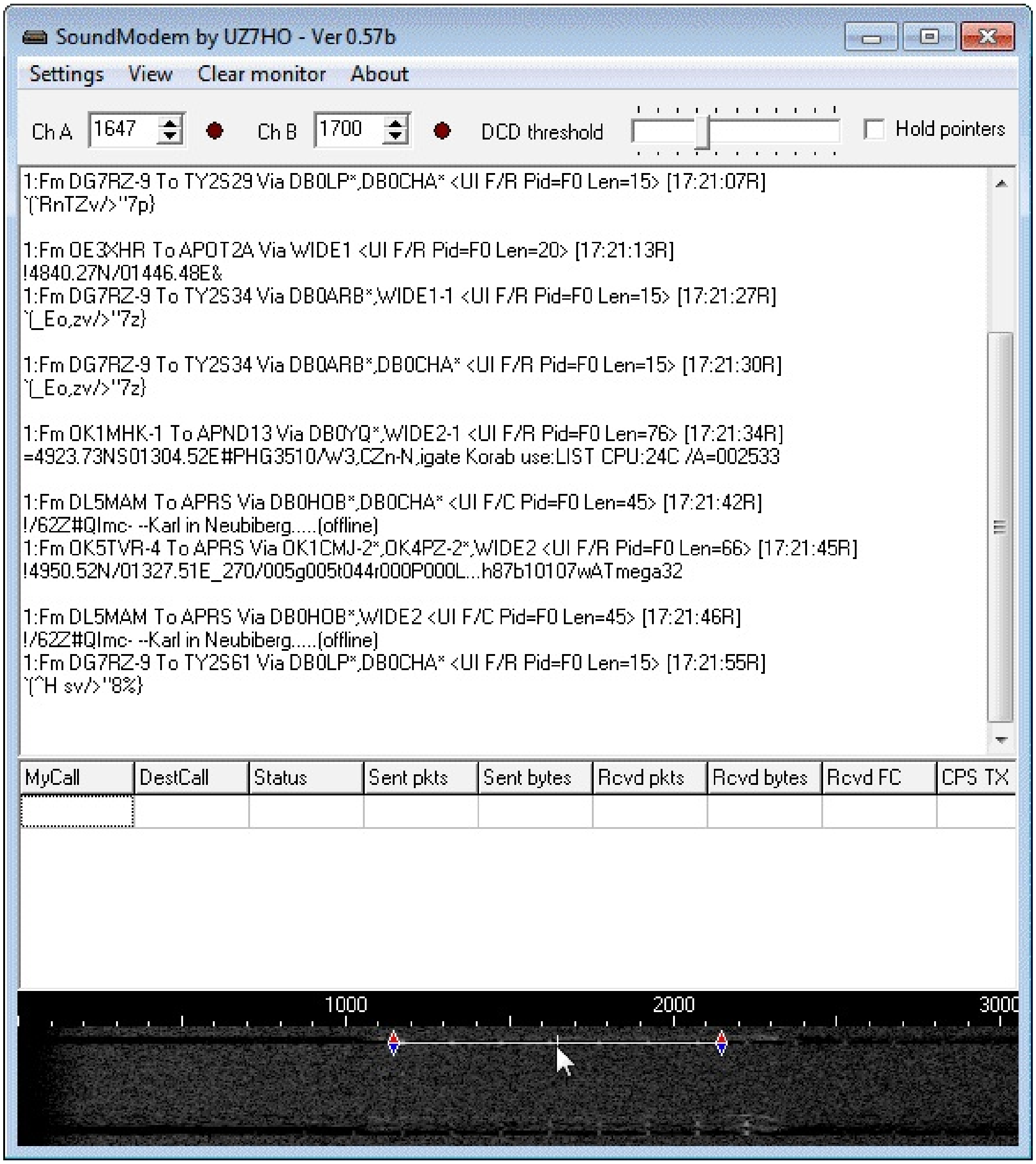
Mode VHF AX.25 1200bd
TXDelay 250 msec
TXTail 50 msec
Add. RX 0 pairs
Add. RX shift 30 hz
Bits Recovery SINGLE

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 screen 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%}
```

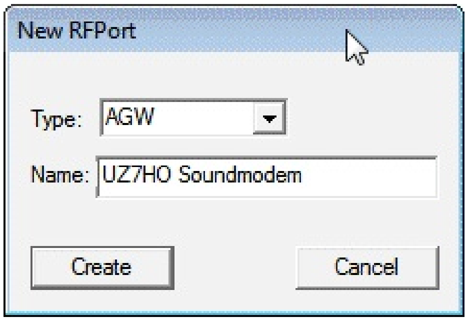
MyCall	DestCall	Status	Sent pkts	Sent bytes	Rcvd pkts	Rcvd bytes	Rcvd FC	CPS TX

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 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.**

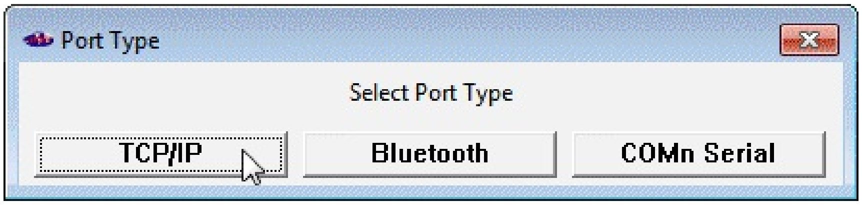
9. Change the type to AGW, set the name as UZ7HO soundmodem.


New RFPort

Type: AGW

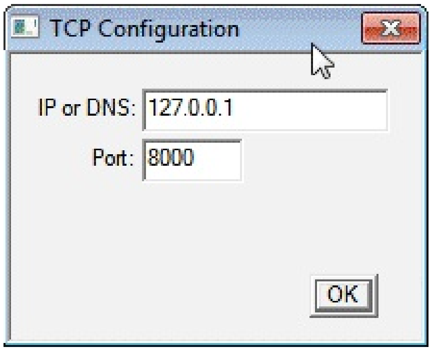
Name: UZ7HO Soundmodem

10. For port type choose TCP/IP.


Port Type

Select Port Type

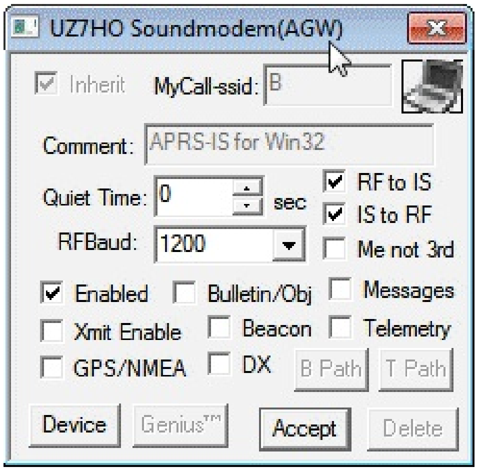
11. Set the IP to 127.0.0.1 and the port to 8000.


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 ☑ RF to IS

☑ IS to RF

RFBaud: 1200 □ 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 program explained 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 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

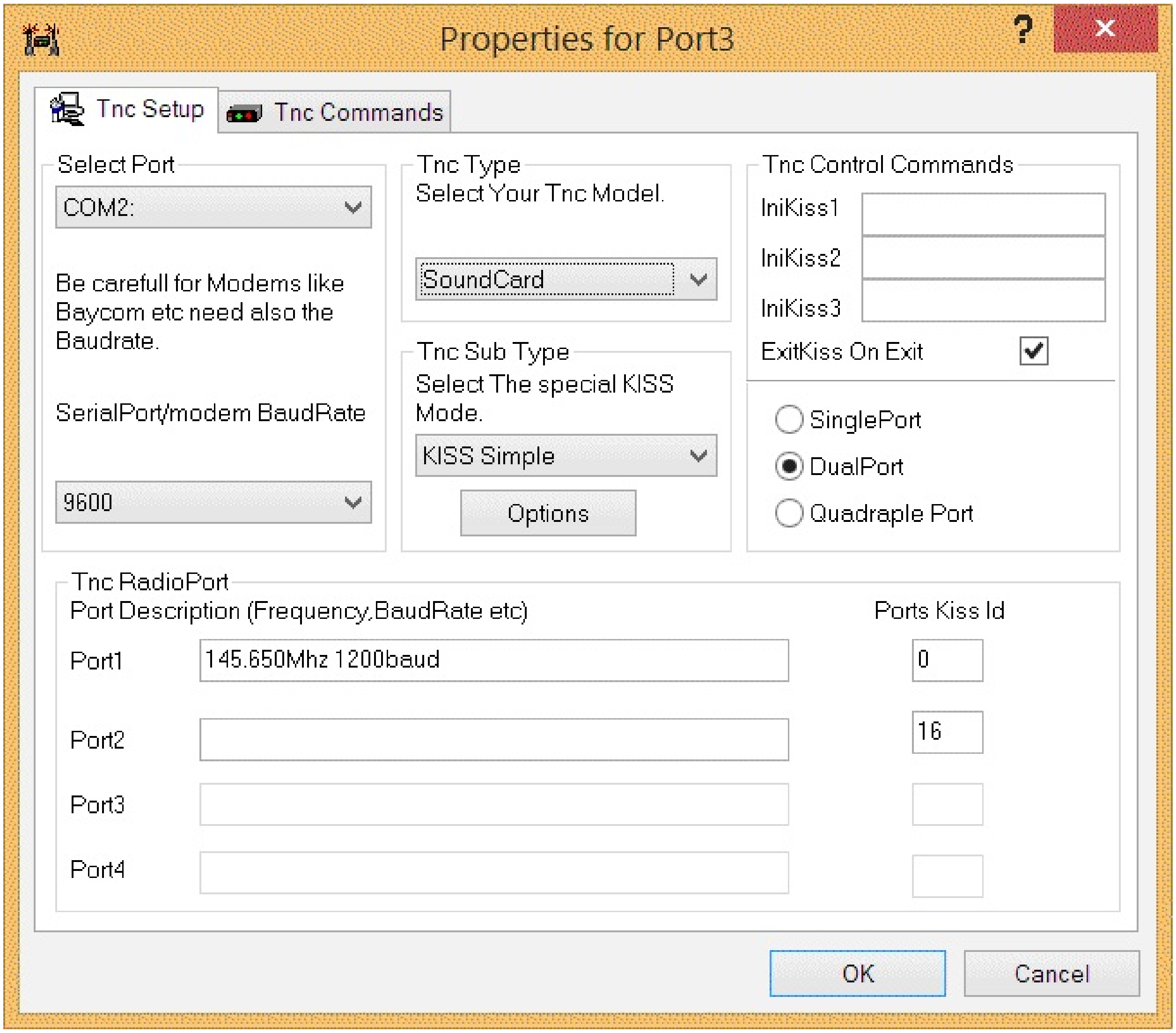
1. Open SDR# or your favourite general purpose SDR receiver and set the audio output to your preferred audio piping method. Tune to an APRS frequency.

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

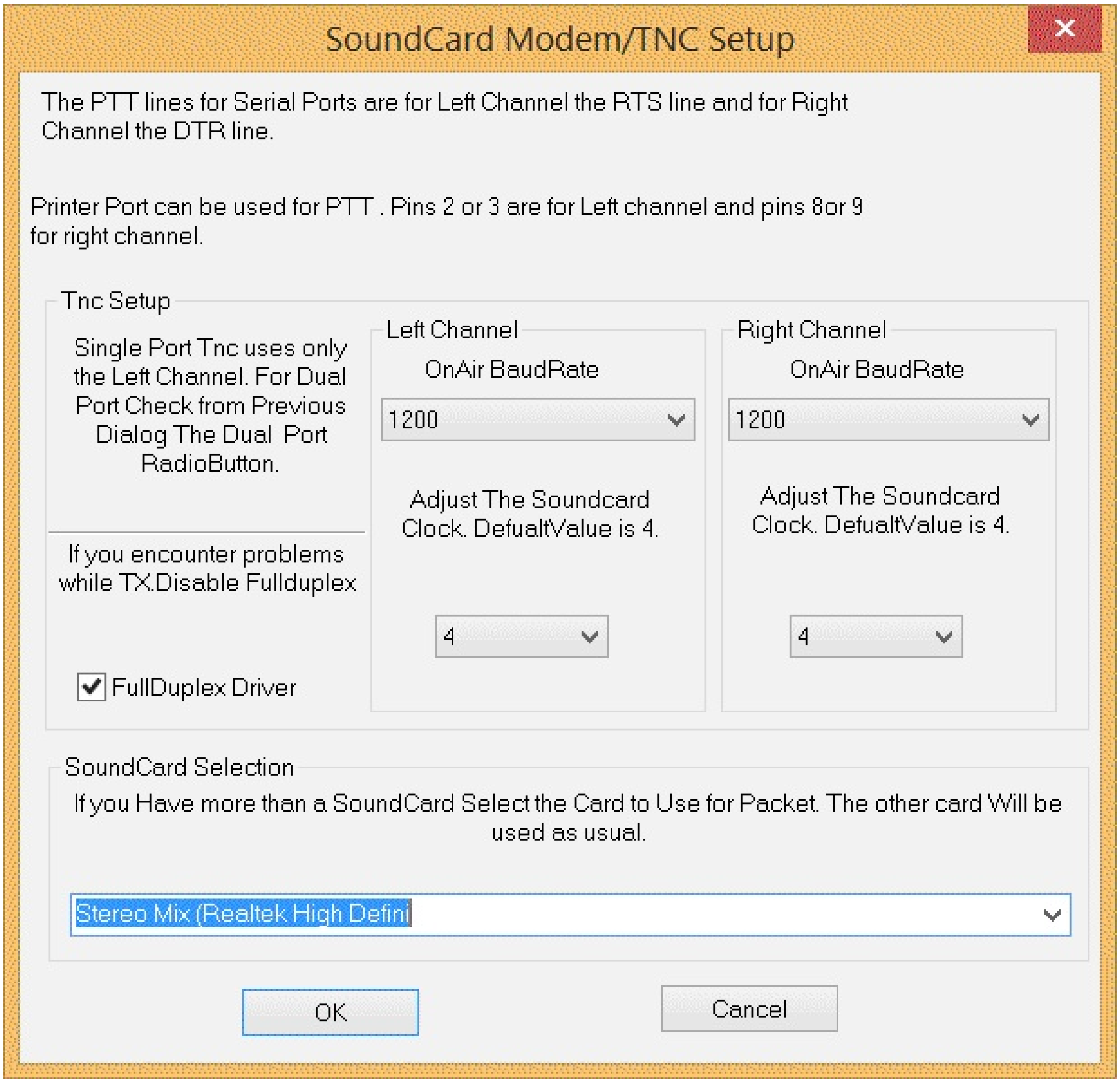
Tnc Control Commands
IniKiss1
IniKiss2
IniKiss3
ExitKiss On Exit ☑

○ SinglePort
● DualPort
○ Quadraple Port

Tnc RadioPort

Port Description (Frequency,BaudRate etc)		Ports Kiss Id
Port1	145.650Mhz 1200baud	0
Port2		16
Port3		
Port4		

6. Now another window will pop up. In this window 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.

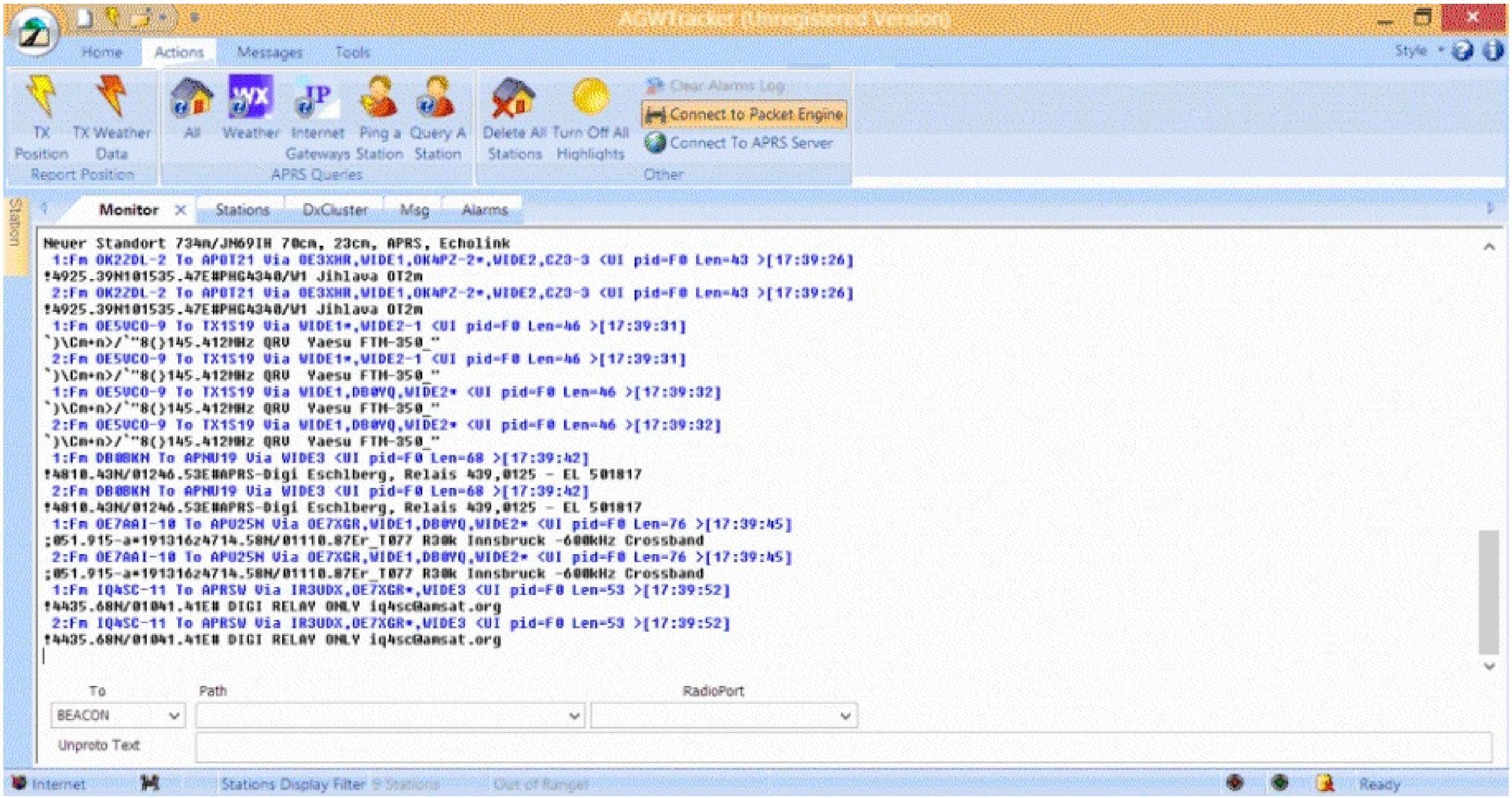
Stereo Mix (Realtek High Defini

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.

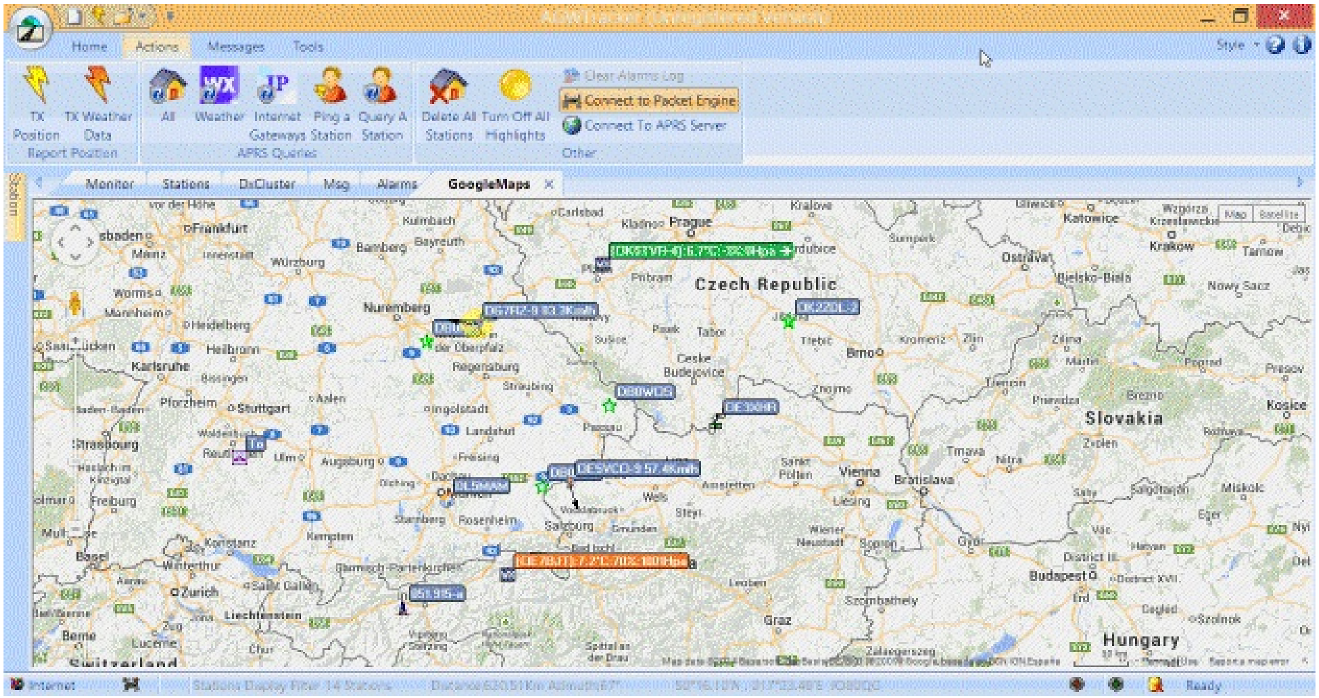
9. Go to the actions tab and ensure that the Connect to packet engine button Connect to Packet Engine is pressed in. Also ensure Connect to APRS Server is not pressed in if you don't want to receive APRS packets via the internet.

10. As 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,DB0VQ,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,DB0VQ,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,DB0VQ,WIDE2* <UI pid=F0 Len=76 >[17:39:45]
;051.915-a*191316z4714.50N/01110.07Er_T077 R30k Innsbruck -600kHz Crossband
 2:Fm OE7AAI-10 To APU25N Via OE7XGR,WIDE1,DB0VQ,WIDE2* <UI pid=F0 Len=76 >[17:39:45]
;051.915-a*191316z4714.50N/01110.07Er_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
```

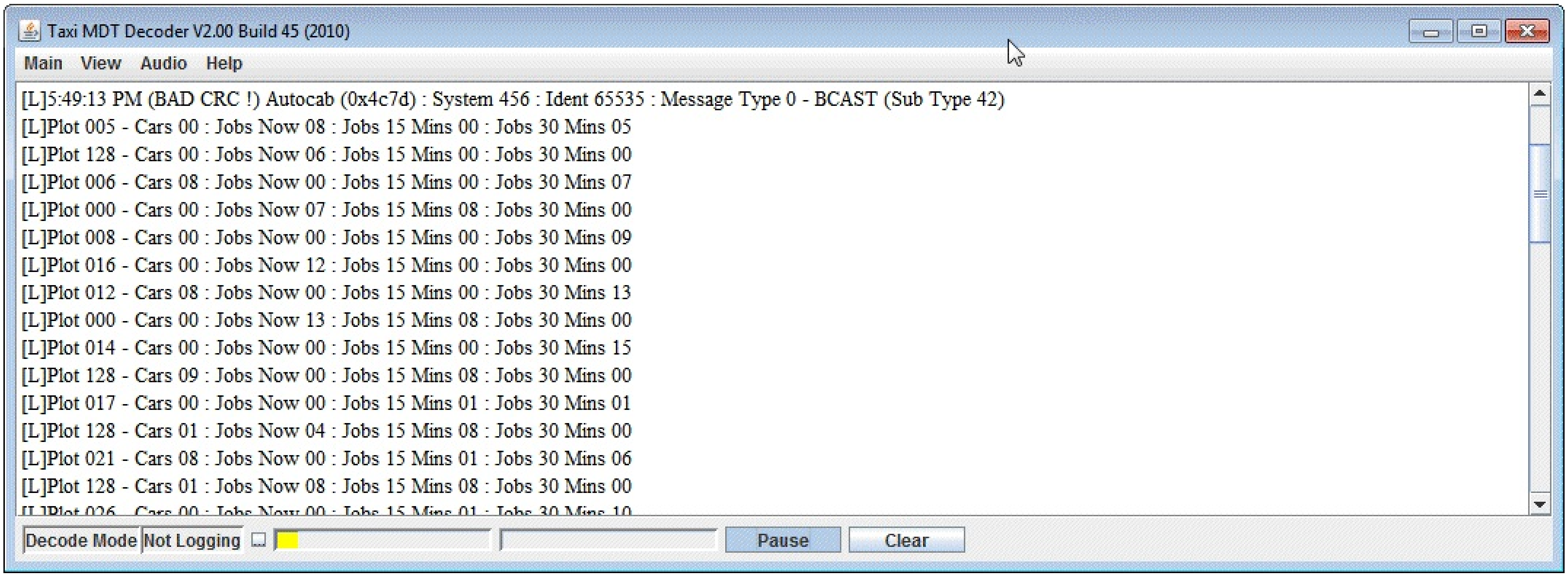
11. To view received stations on a map, go to the Stations Tab, then right click on a station and select locate and show on map. In the window that pops up choose your preferred map (Google, Bing or Yahoo! Maps). You can then zoom out to see all the other stations.



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 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 and pipe the audio into taxi decoder. Go to **Audio->Audio Devices** and choose your audio piping method. 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. 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 first download and extract the zip file to a folder on your computer. Then follow these instructions for Windows.

1. Open your favourite SDR receiver such as SDR#.
2. Set your audio output to your favourite audio pipe.
3. Open a Windows command prompt. Go to **Start->All Programs>Accessories->Command Prompt**.
4. 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 Windows multimonng will by default listen to your default audio piping method set in Windows sound recording properties.

On Linux you will need to pipe in FM audio with a sample rate of 22050 into multimonNG. You can use rtl_fm to do this referring to 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 sample rate, 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 I/Q 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 bandwidth. To calculate the bandwidth required we need to multiply the chosen sample rate by the number of bits per sample. The number of bits in the raw 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 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 and 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.

To use rtl_tcp on Windows follow the instructions below.

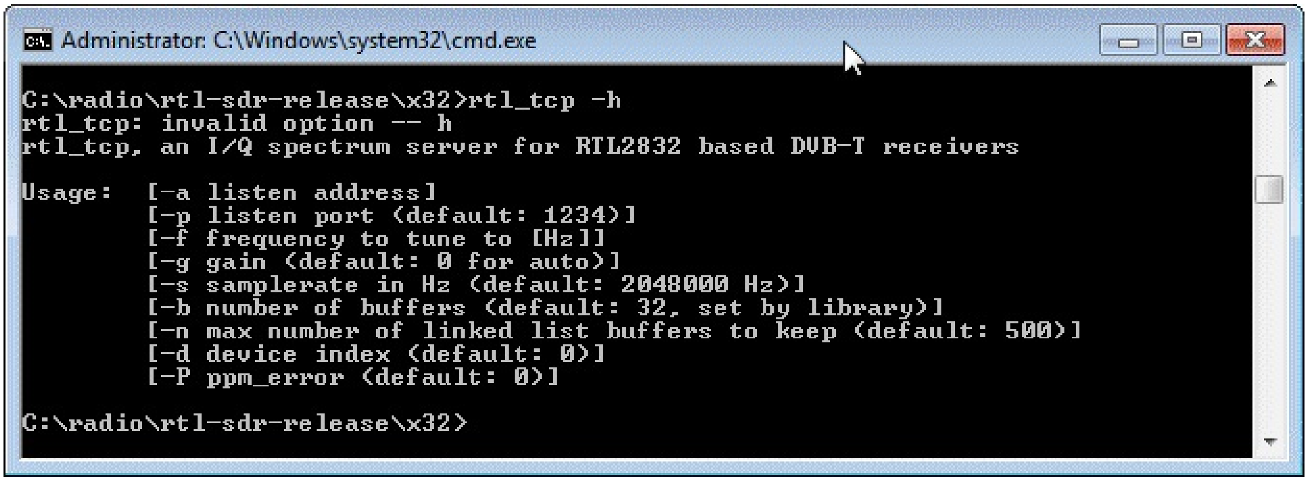
1. Download the rtl_sdr precompiled binaries for Windows from http://sdr.osmocom.org/trac/attachment/wiki/rtl-sdr/RelWithDebInfo.zip.

2. Extract the RelWithDebInfo.zip 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. 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
```

2. 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#.

3. Press Play.

To use SDR Touch with RTL_TCP

1. At the command prompt use the following (where -s sets the sample rate)

```
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.

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 only needing about 8kb/s at most which almost any network including the internet can handle.

On Ubuntu sox, socat and netcat can be installed with

```
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 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 first on a sample 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, 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

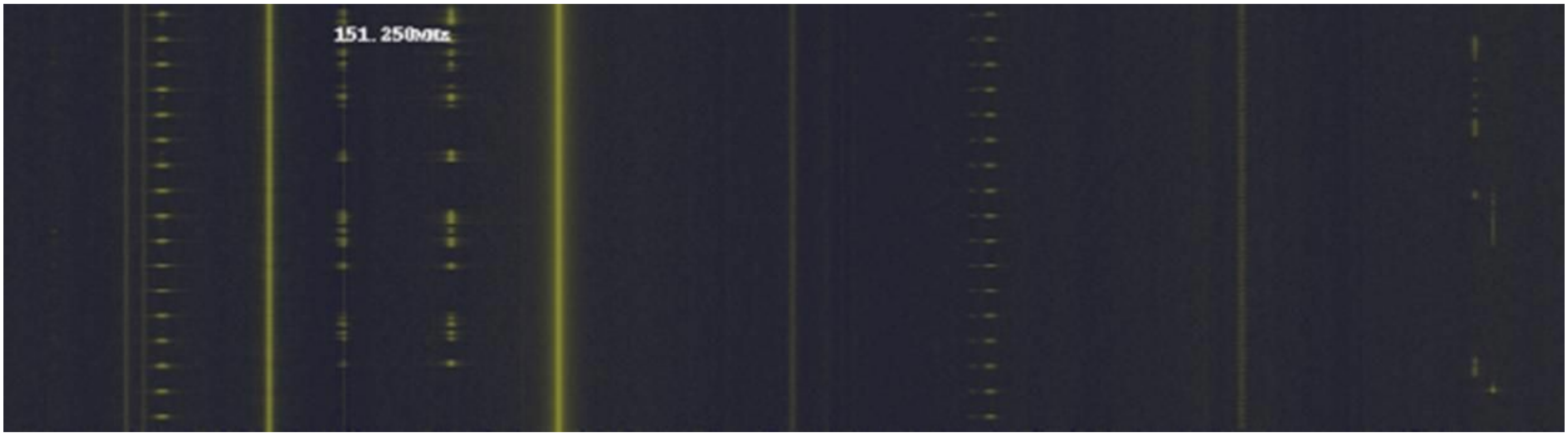
```
rtl_fm -M fm -f 161.649M -s 12k -g 50 -l 70 | sox -traw -r12k -es -b16 -c1 -V1 - -tmp3 - | socat -u - tcp-listen:8080
```

On the listener side use the command

```
netcat IP_ADDRESS 8080 | play -t mp3 -
```

Where you should change the IP_ADDRESS to the ip address of your sending server. You should 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 (http://sdr.osmocom.org/trac/attachment/wiki/rtl-sdr/RelWithDebInfo.zip) can be used. This tutorial will be based on Windows, but it can easily be adapted to work on Linux.

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 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.

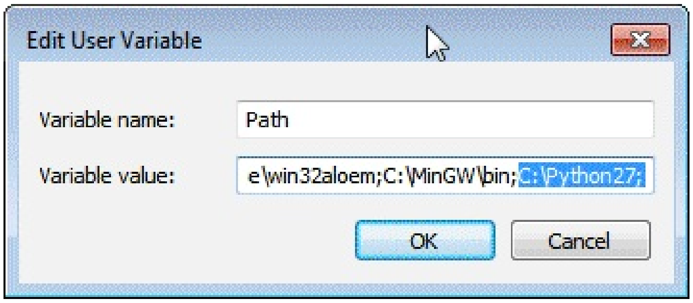
1. Download the official RTL-SDR release for Windows (http://sdr.osmocom.org/trac/attachment/wiki/rtl-sdr/RelWithDebInfo.zip) and unzip it to a folder.

2. Download 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.



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, making sure to keep the one decimal place.

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
-e exit_timer The length of time that the program should run
```

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.

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.

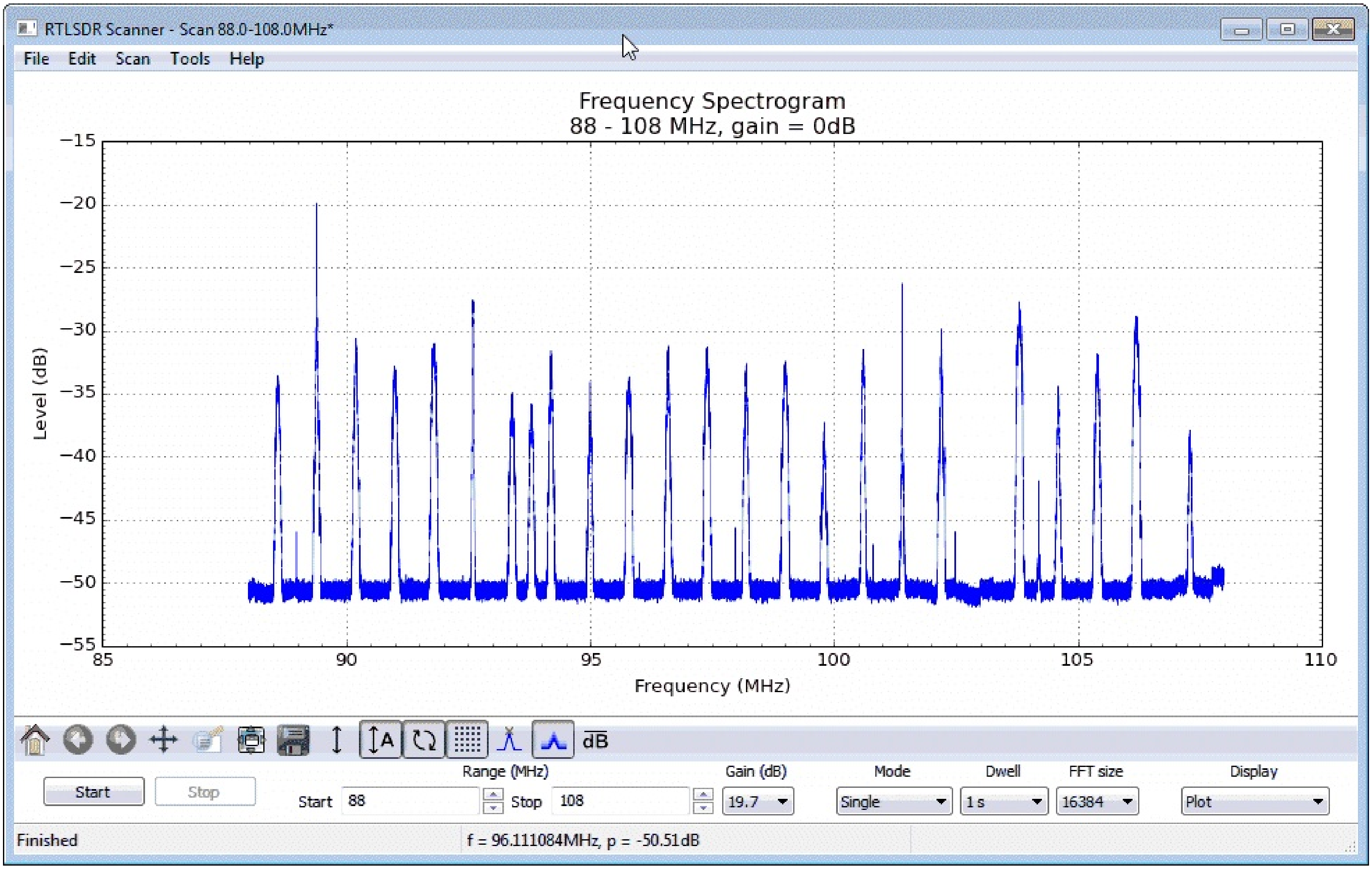
The Mode can be set to either Continuous or Single. In continuous mode the scanner will continuously refresh the spectrum every time it finishes one pass.

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.

The FFT size indicates the FFT resolution used during scanning. Higher values will give higher resolutions graphs which can see the structure of signals more clearly.

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.



Start 88 Stop 108

Gain (dB) 19.7

Mode Single

Dwell 1 s

FFT size 16384

Display Plot

Finished

f = 96.111084MHz, p = -50.51dB

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 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.

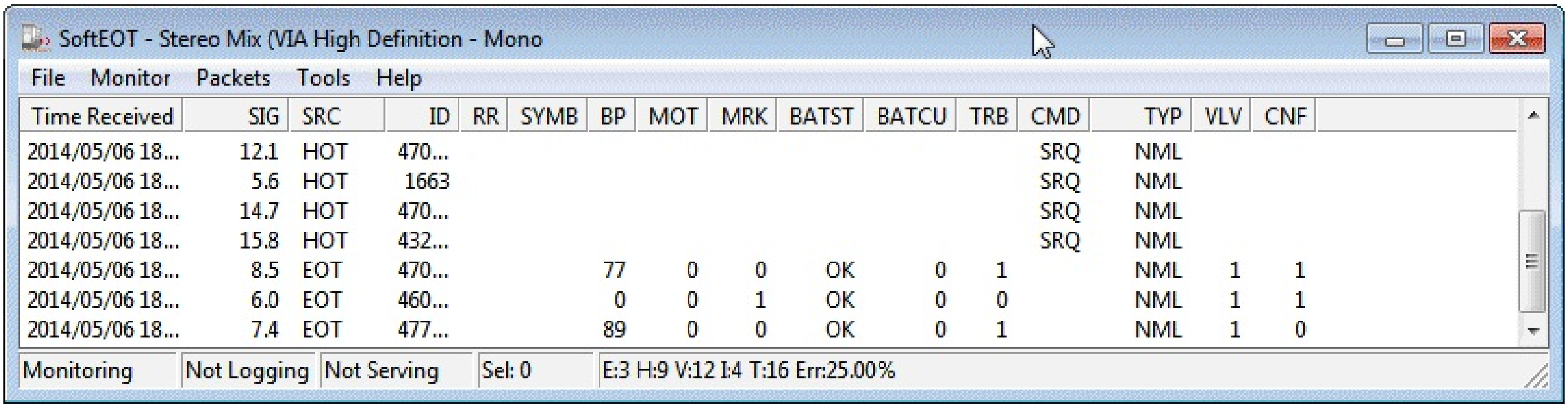
HOT, EOT and DPU signals can be decoded and monitored using the RTL-SDR, a general purpose receiver, audio piping and some special decoding software. The decoding software goes by the names of softEOT and softDPU and can only be downloaded after registering for a Yahoo! Group at https://groups.yahoo.com/neo/groups/SoftEOT/. Registering may take a few hours as you must wait for a moderator to approve your application.

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 a default audio device for audio piping in the Windows sound recording settings.
3. Open your favourite general purpose SDR GUI such as SDR# or SDR-Console.
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.

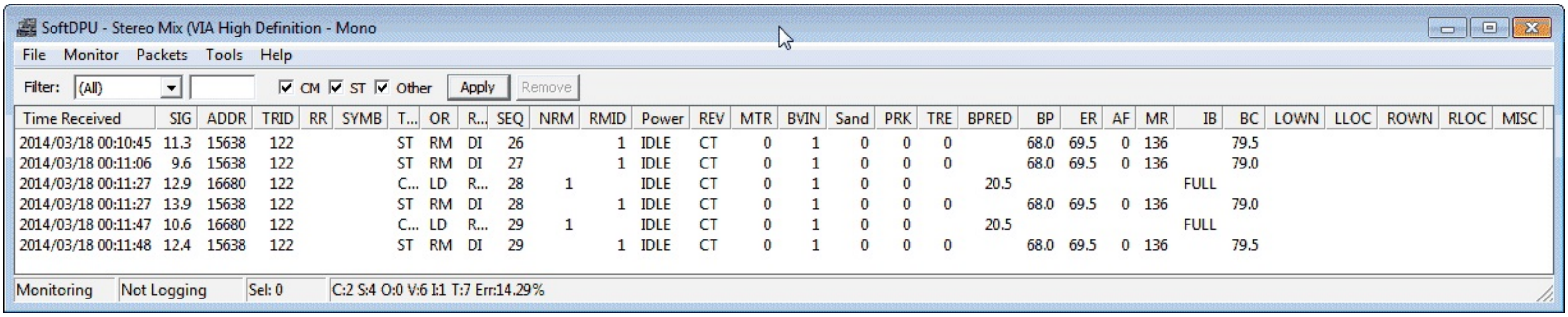
Decoded messages will begin to show.



SoftEOT - Stereo Mix (VIA High Definition - Mono

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%



SoftDPU - Stereo Mix (VIA High Definition - Mono

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 train 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 general purpose SDR receiver like SDR# using NFM mode and piping the audio into TrainPlotter.

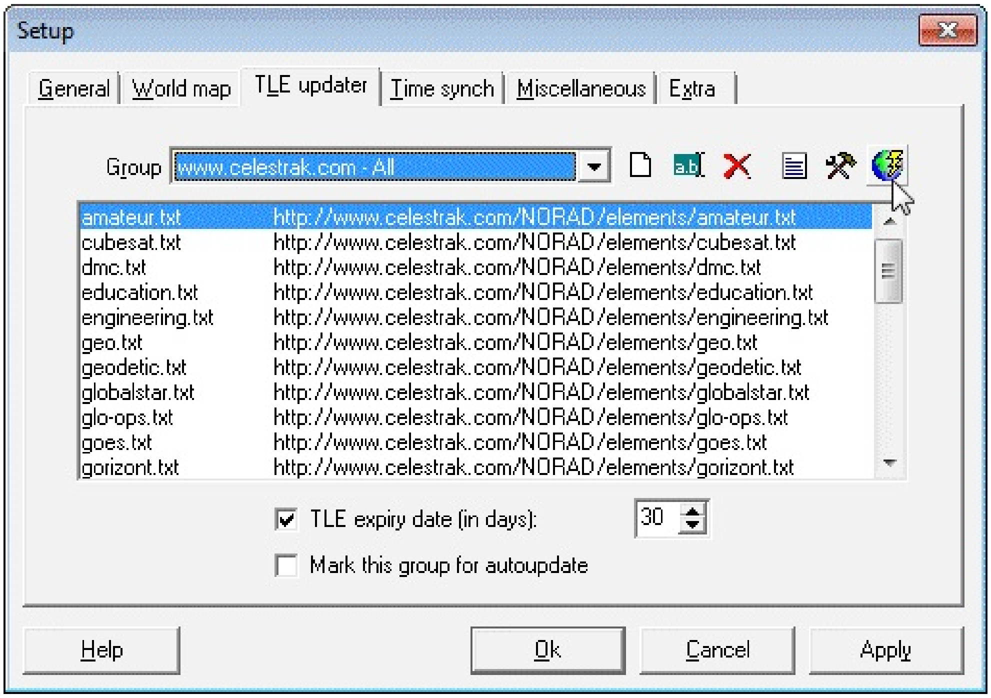
DECODING FUNCUBE SATELLITE TELEMETRY

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. 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.

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

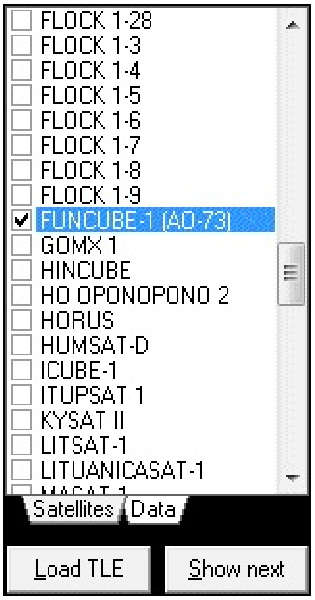
☐ Mark this group for autoupdate

3. Now click the Load TLE button Load TLE on the right of the Orbitron window.

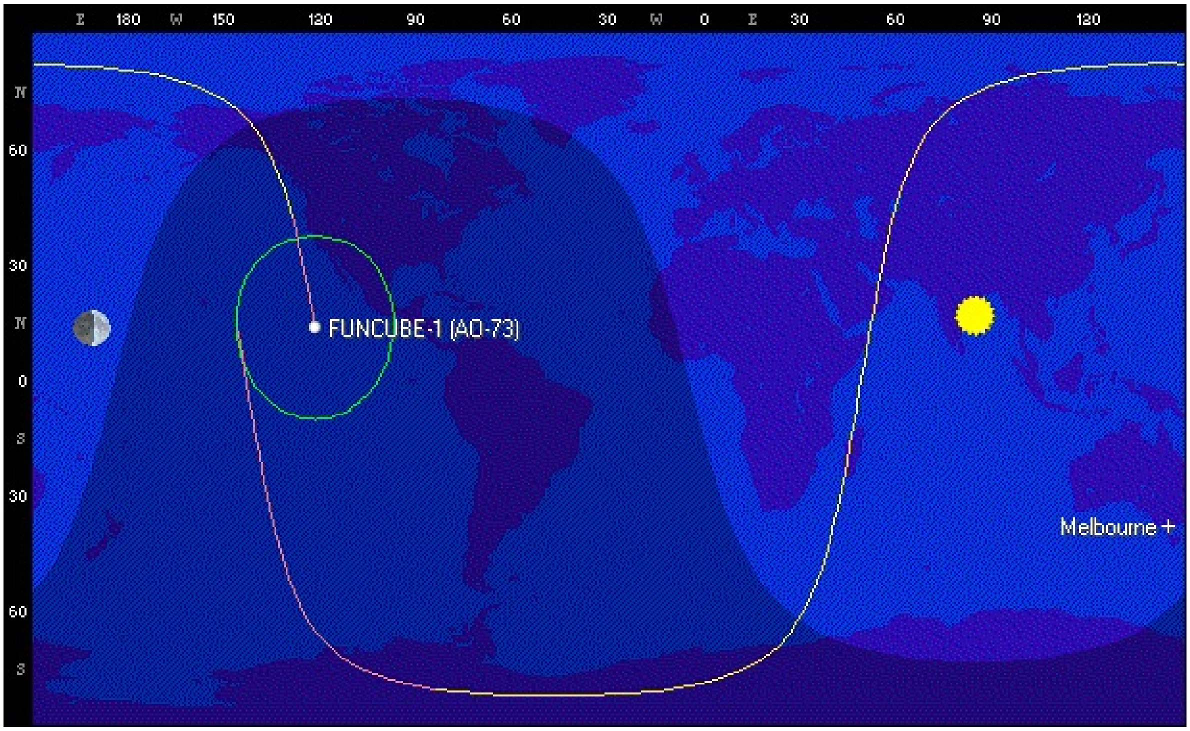
4. Open Cubesat.txt.

5. There will now be a list of satellites in the top right of the Orbitron window. Scroll down and

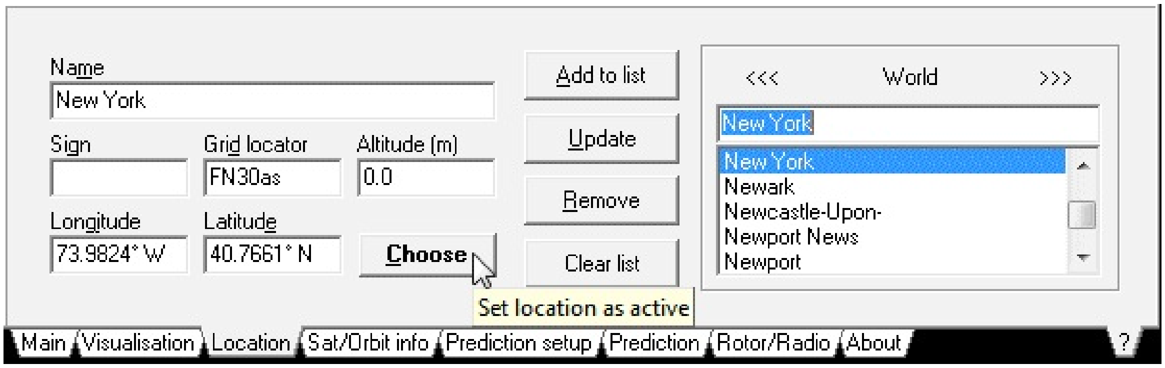
place a check next to Funcube-1 (AO-73). The current position of the satellite will show in the Orbitron map.



□ 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



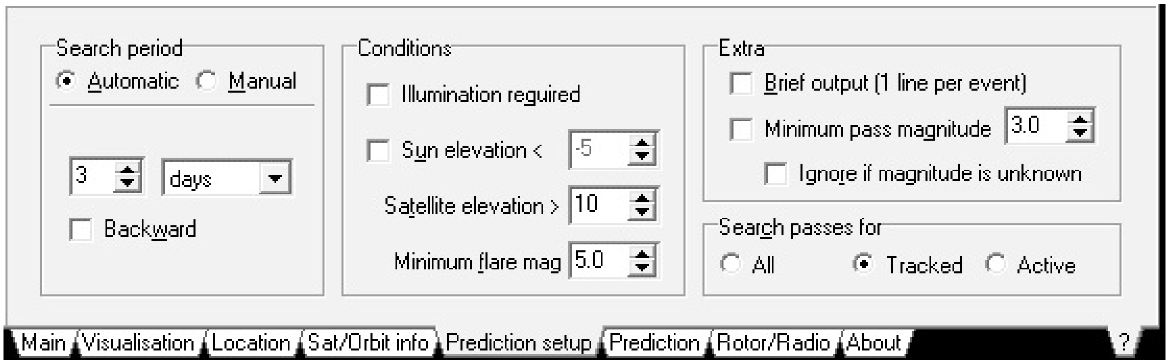
6. To predict when the satellite will be overhead, first set your location by clicking on the location tab at the bottom of the Orbitron window. Either enter your lat/long coordinates, or find your city in the countries list on the right. Click the Choose button when done.



Name: New York
Sign:
Grid locator: FN30as
Altitude (m): 0.0
Longitude: 73.9824° W
Latitude: 40.7661° N

New York
New York
Newark
Newcastle-Upon-
Newport News
Newport

7. Now click on the Prediction setup tab and uncheck “Illumination Required”.



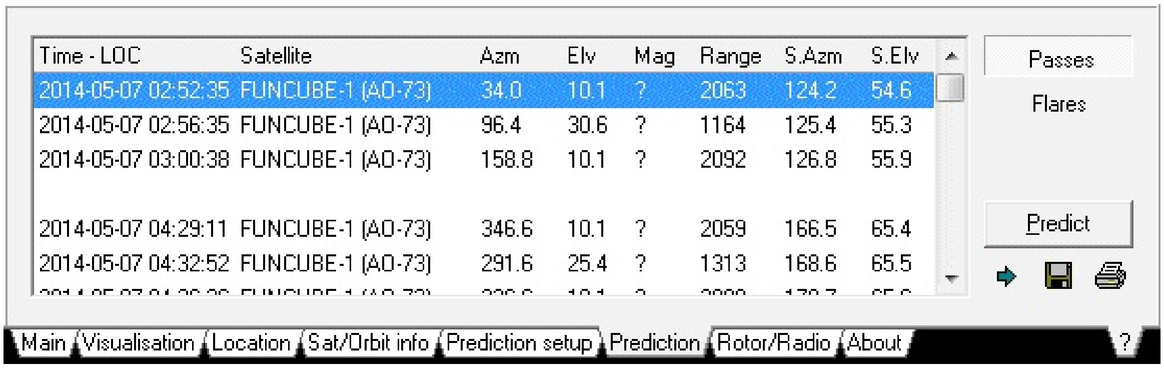
Search period
● Automatic ○ Manual
3 days
□ Backward

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

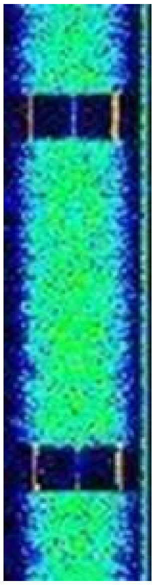
You may also want to look at the NOAA satellite Orbitron guide which can show you how to interface Orbitron with SDR# to automatically tune to the Funcube signal.

To decode the Funcube-1 telemetry follow these instructions

1. Open your favourite general purpose SDR receiver, 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.

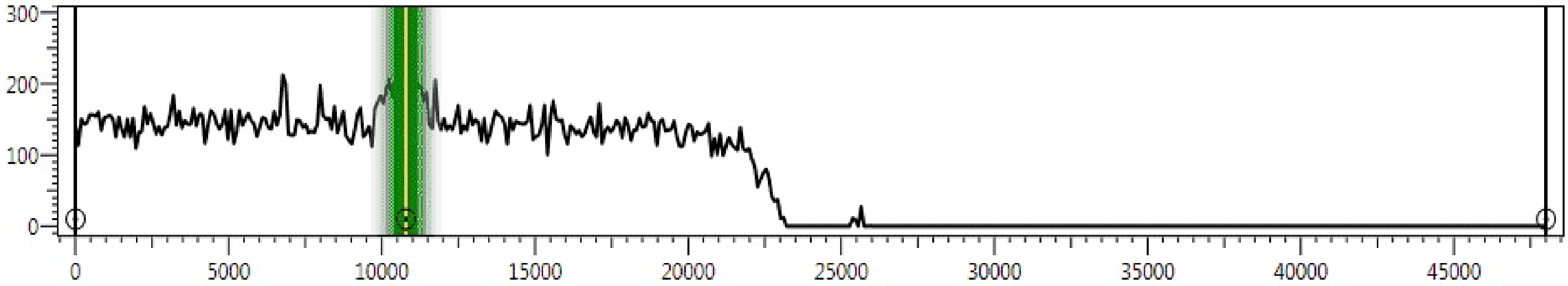
3. Tune to a frequency around 145.950 - 145.970 MHz and look for the Funcube signal. Funcube telemetry data looks like this on the waterfall.



4. Open the Funcube Telemetry Dashboard software which can be downloaded from http://funcube.org.uk/working-documents/funcube-telemetry-dashboard/.

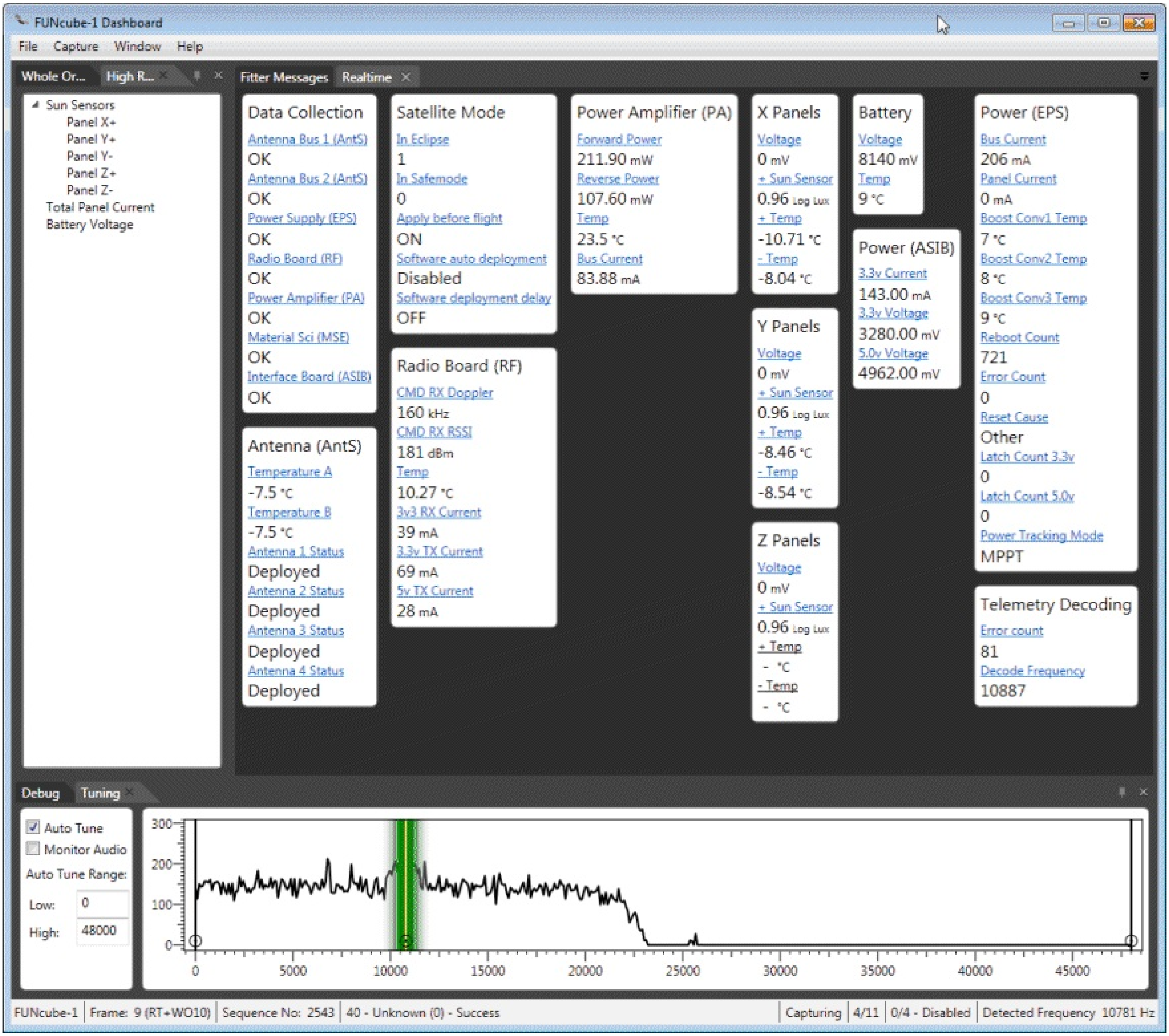
5. Go to **Capture -> Capture from soundcard** to begin capturing the data.

6. If it is working correctly, the audio graph at the bottom of the window will start showing audio.



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

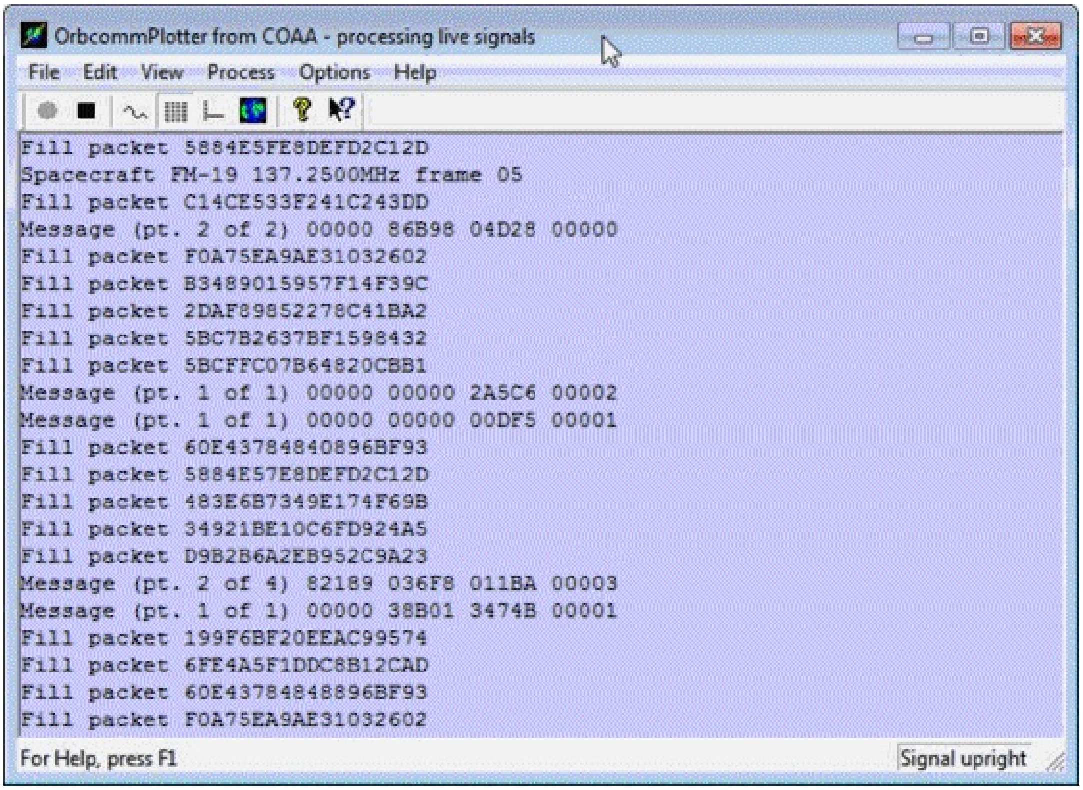
Orbcomm satellites are low earth orbit communications satellites that can provide two way data services and also global positioning services to portable terminals. They transmit data at 137 to 138 MHz making them easy for scanners to receive with a good antenna. They are also easy to receive as there are many orbcomm satellites in the sky positioned such that there will almost always be one available to receive.

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.

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-Console.
2. Set your 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 your audio piping method 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.


OrbcommPlotter from COAA - processing live signals
File Edit View Process Options Help
For Help, press F1
Signal upright

```
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
```

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 using plain NFM audio transmissions during spacewalks. These can be found at around 121 MHz and 130 MHz. The signal is usually quite strong so any antenna built for around 143 MHz should work.

The ISS astronauts also sometimes communicate to schools using ham radio at a downlink frequency of around 145 MHz.

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 make SDR# 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. Build instructions are on the git hub page.

If there is a sensor transmitting on a format that rtl_433 understands you will be able to see sensor output data. Currently on a few sensors are supported.

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.

DECODING SMART POWER METERS

Smart Meters are RF enabled electricity meters which allow electricity meter reader personnel to collect consumption data simply by standing outside the house. Typically, this information is broadcast in the 900 MHz ISM band.

A Windows and Linux program called rtl_amr can be used to decode and monitor this information being sent. Rtl_amr can be obtained as a source file from https://github.com/bemasher/rtlamr. It will require installation of the Go language first.

If you have an Efergy Energy Meter the following Linux program 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.

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.

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. Further instructions can be found at http://blog.cyberexplorer.me/2014/01/sniffing-and-decoding-nrf24l01-and.html.

ANALYZING UNKNOWN SIGNALS

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# and tune to a simple waveform 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.

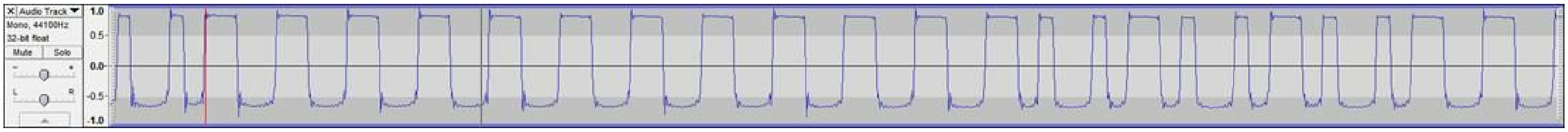
5. Wait for a packet to play or let your signal play for a few seconds and then click the stop button in audacity.

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.

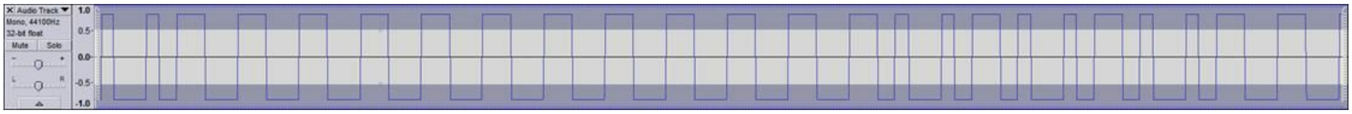
7. Zoom in on the captured waveform by holding the Ctrl key whilst scrolling the mouse wheel up.

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. MiniModem and install instructions can be found at http://www.whence.com/minimodem/.

RECEIVING AND TRACKING GPS WITH THE RTL-SDR

The RTL-SDR can be used to receive and track 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 their strength is below the noise floor. Proper signal processing software is required to detect GPS such as the GNSS-SDR Linux software.

Note: I and several other people 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 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.

To install this software use the following instructions.

Install GNU Radio 3.7. Follow the GNU Radio tutorial 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

```
export RTLSDR_DRIVER=1
```

Now also provide the path to your gr-osmosdr install using

```
export OSMOSDR_ROOT=/path/to/gr-osmosdr
```

Now install and build the GNSS software

```
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 it’s crystal inaccuracies.

Now 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

```
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

```
./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
```

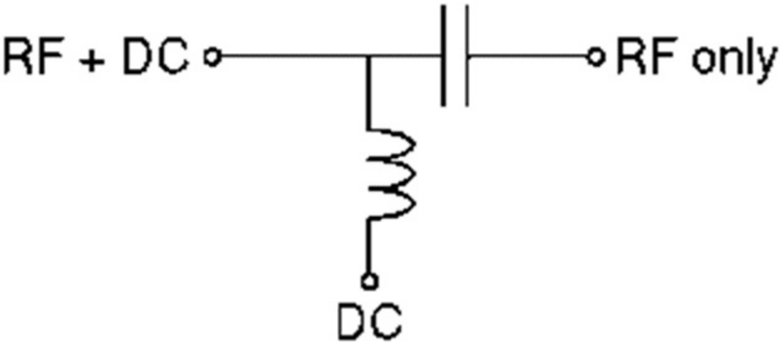
```
 18   -76500.00   -1141.35
 21   -78000.00   -2679.26
 22   -75000.00   219.07
Parameters estimation for Elonics E4000 Front-End:
Sampling frequency =2000095.73 [Hz]
IF bias present in baseband=-75410.22 [Hz]
Reference oscillator error =47.87 [ppm]
Corrected Doppler vs. Predicted
SV ID  Corrected [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 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 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. Don’t use too large capacitors 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 used in NOAA weather satellites except that the images are much higher in resolution. HRPT is also transmitted by 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. That is a computerized satellite dish which automatically follows the satellite in the sky. You can see the type of antenna required 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.

http://www.satsignal.eu/software/hrpt.htm

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 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 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/
- Decoding 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/

DVB-T HDTV ON LINUX

The RTL-SDR can still be used for its original purpose: DVB-T TV watching. 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 make DVB-T TV work. 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.

UPDATING VIDEO4LINUX

The 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)

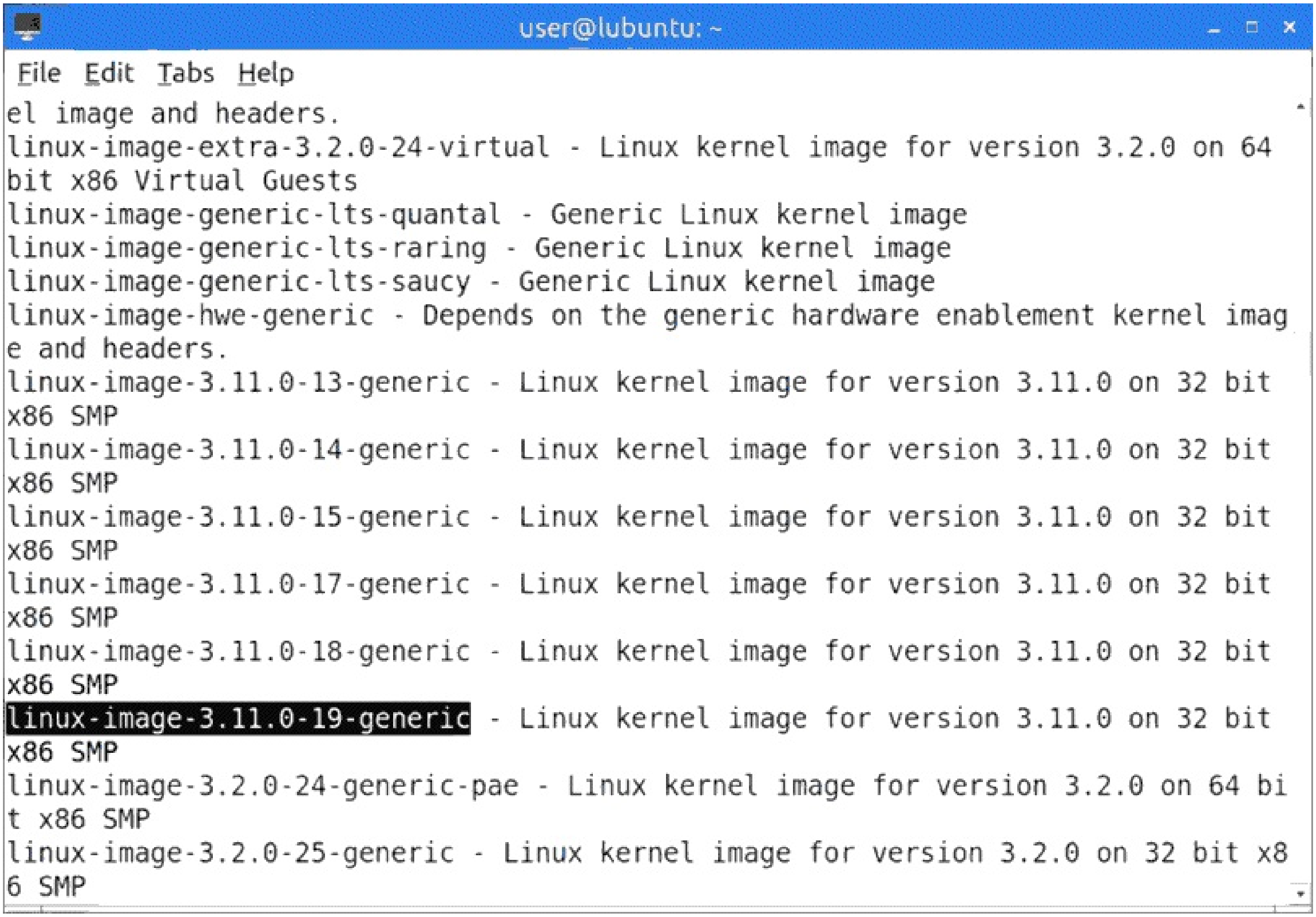
UPDATING THE KERNEL

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
```



```
el image and headers.
linux-image-extra-3.2.0-24-virtual - Linux kernel image for version 3.2.0 on 64
bit x86 Virtual Guests
linux-image-generic-lts-quantal - Generic Linux kernel image
linux-image-generic-lts-raring - Generic Linux kernel image
linux-image-generic-lts-saucy - Generic Linux kernel image
linux-image-hwe-generic - Depends on the generic hardware enablement kernel imag
e and headers.
linux-image-3.11.0-13-generic - Linux kernel image for version 3.11.0 on 32 bit
x86 SMP
linux-image-3.11.0-14-generic - Linux kernel image for version 3.11.0 on 32 bit
x86 SMP
linux-image-3.11.0-15-generic - Linux kernel image for version 3.11.0 on 32 bit
x86 SMP
linux-image-3.11.0-17-generic - Linux kernel image for version 3.11.0 on 32 bit
x86 SMP
linux-image-3.11.0-18-generic - Linux kernel image for version 3.11.0 on 32 bit
x86 SMP
linux-image-3.11.0-19-generic - Linux kernel image for version 3.11.0 on 32 bit
x86 SMP
linux-image-3.2.0-24-generic-pae - Linux kernel image for version 3.2.0 on 64 bi
t x86 SMP
linux-image-3.2.0-25-generic - Linux kernel image for version 3.2.0 on 32 bit x8
6 SMP
```

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.

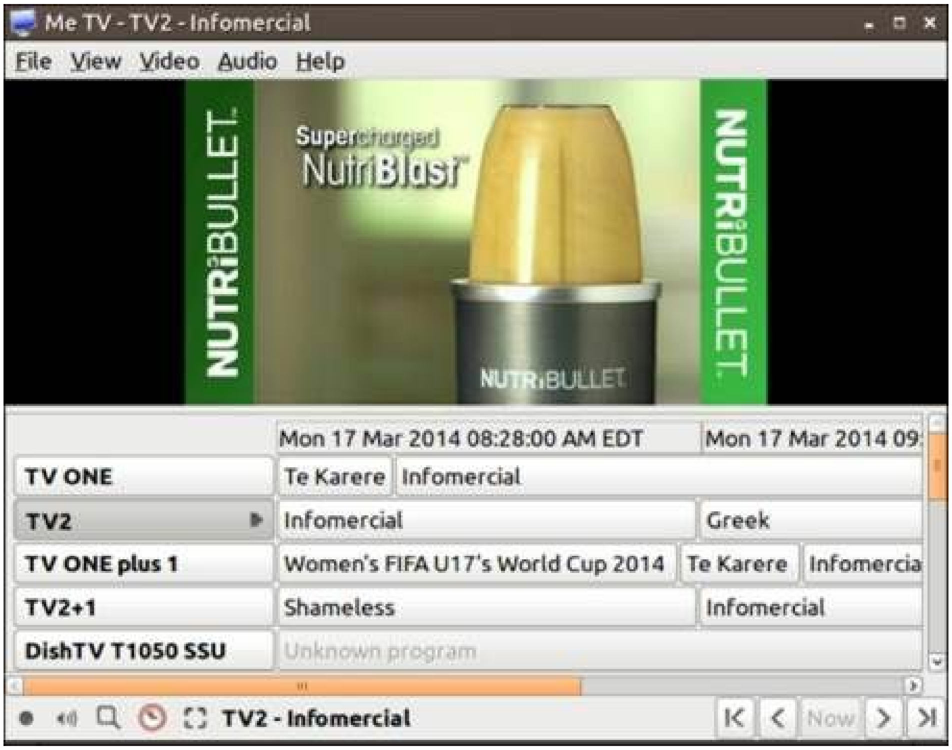
```
sudo apt-get install linux-image-3.11.0-19-generic
```

DVB-T TV WATCHING SOFTWARE

A program called Me-TV can be used to watch DVB-T TV and can be downloaded with

```
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.



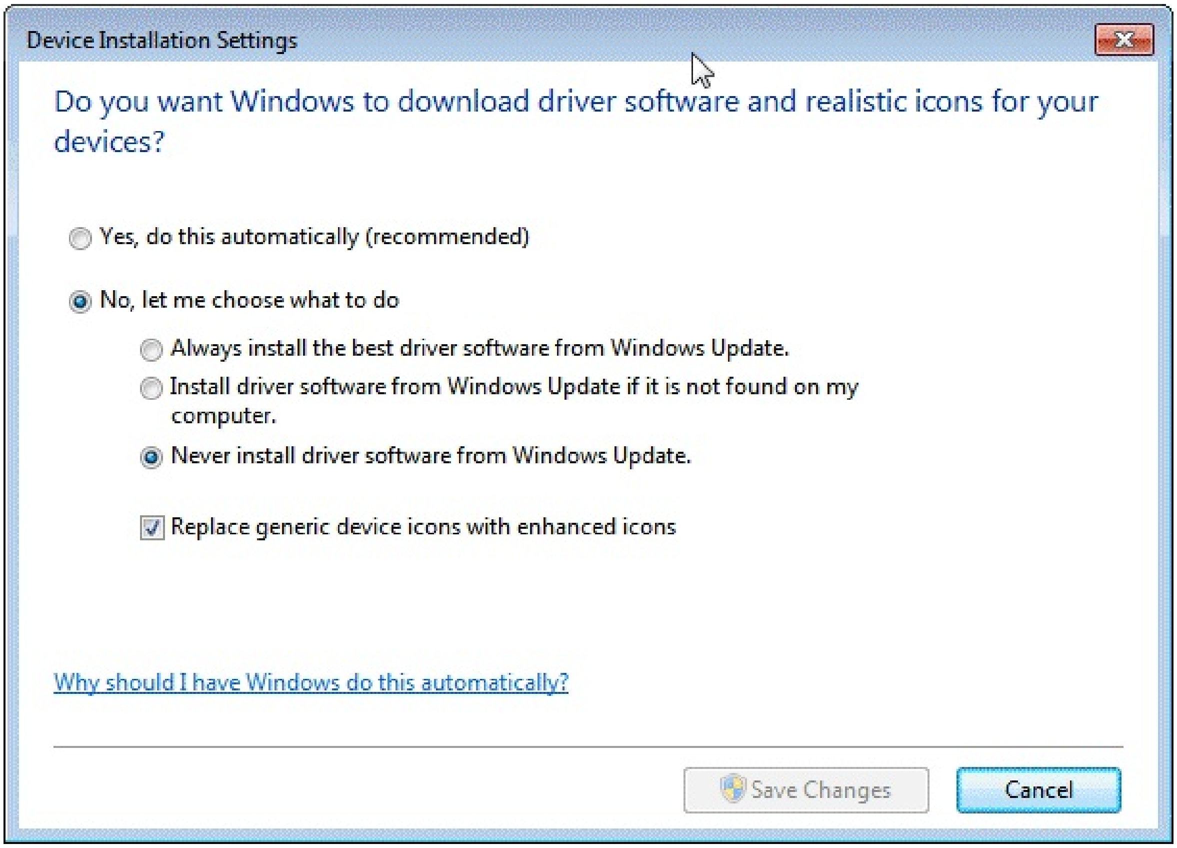
	Mon 17 Mar 2014 08:28:00 AM EDT		Mon 17 Mar 2014 09:
TV ONE	Te Karere	Infomercial	
TV2	Infomercial		Greek
TV ONE plus 1	Women's FIFA U17's World Cup 2014	Te Karere	Infomercia
TV2+1	Shameless		Infomercial
DishTV T1050 SSU	Unknown program		

Other software like MythTv can also be used.

DVB-T ON WINDOWS

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 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.

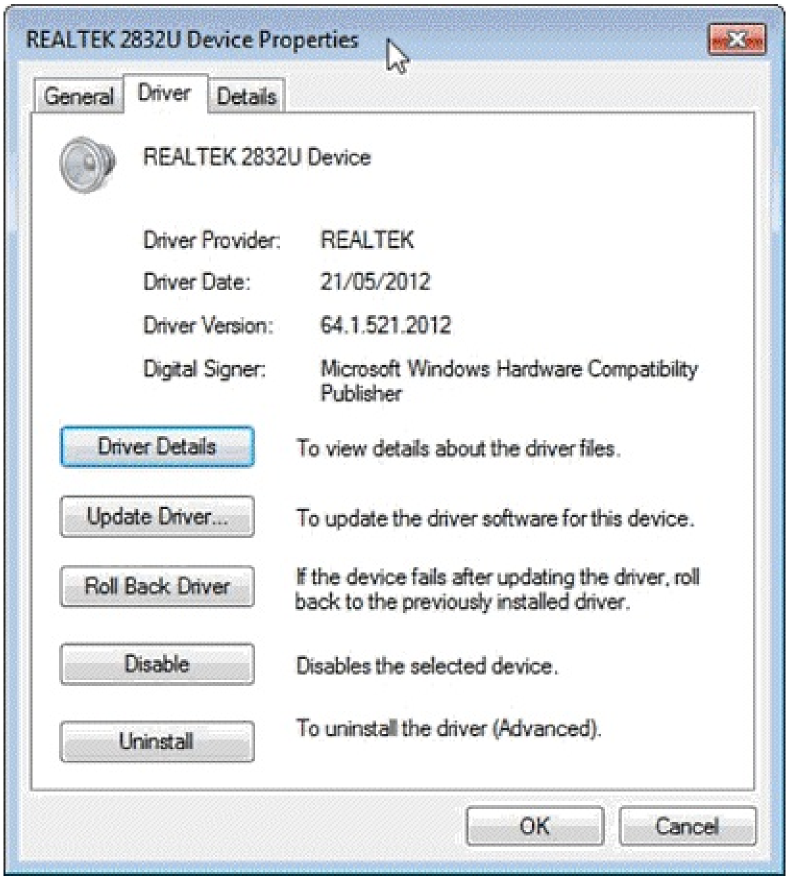
○ Install driver software from Windows Update if it is not found on my computer.

● Never install driver software from Windows Update.

☑ Replace generic device icons with enhanced icons

Now you can install the DVB-T drivers from the CD. If you're 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. 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 Properties

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.

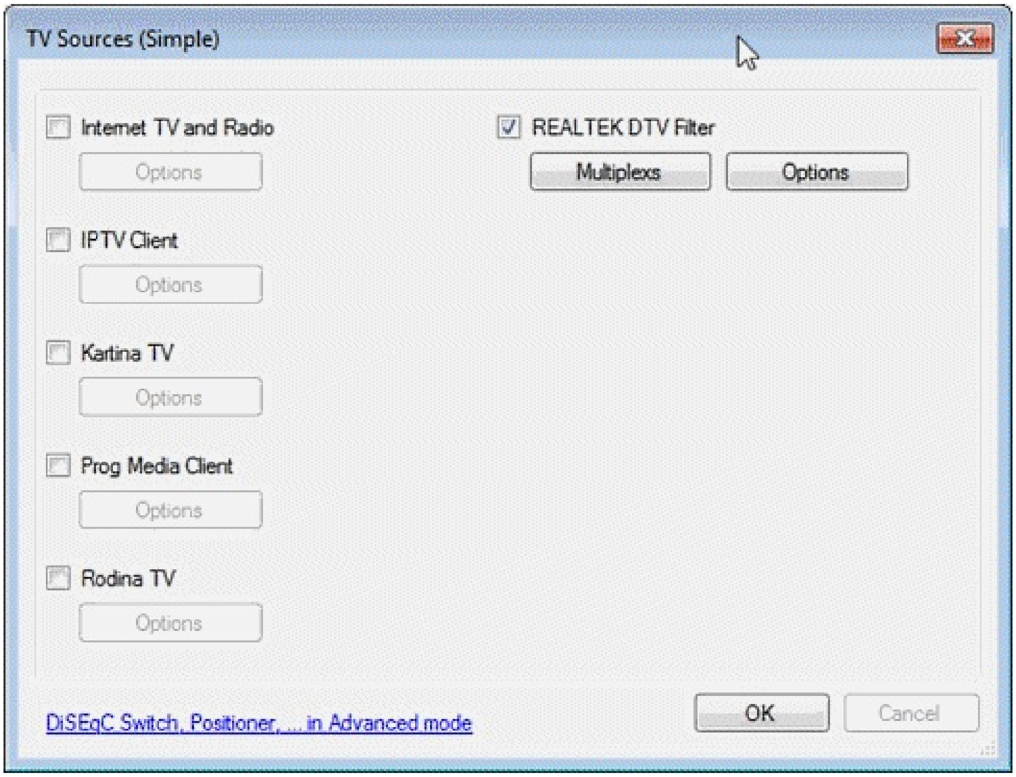
To update the driver software for this device.

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)

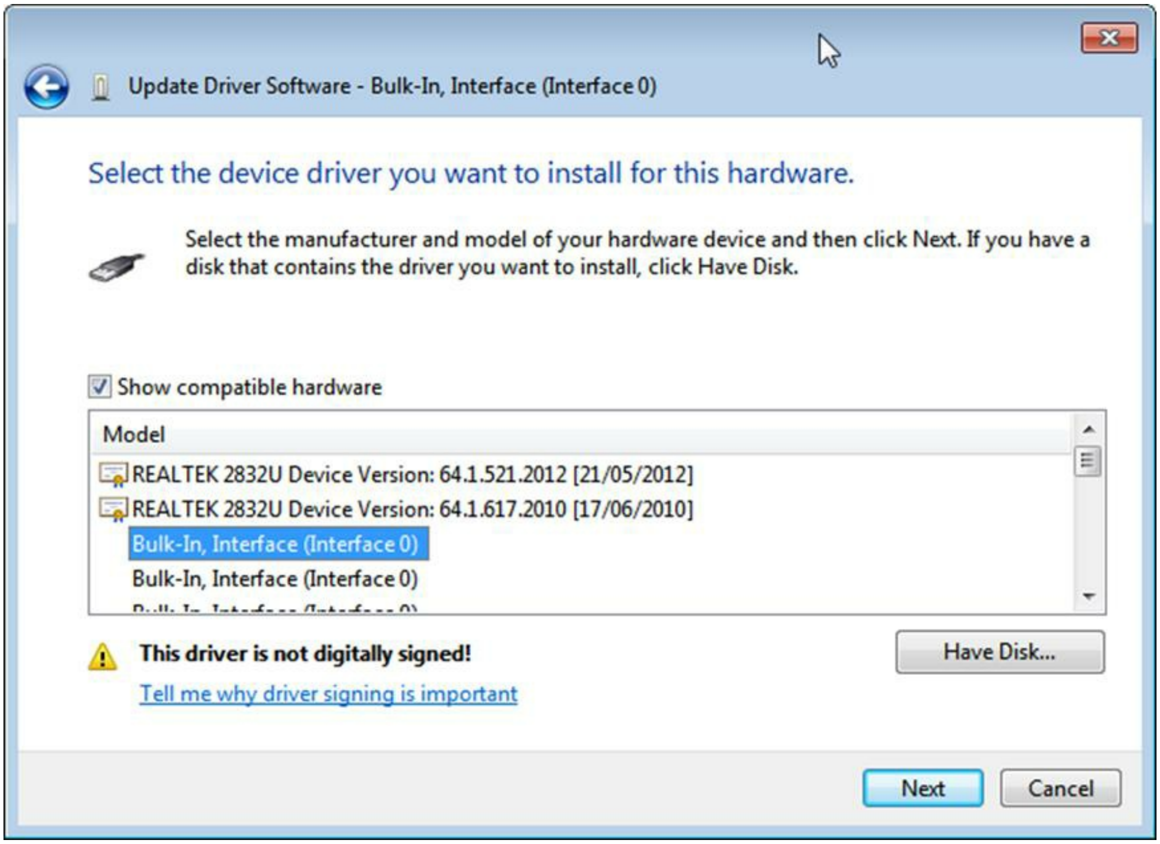
☐ Internet TV and Radio
☐ IPTV Client
☐ Kartina TV
☐ Prog Media Client
☐ Rodina TV
☑ REALTEK DTV Filter

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)”**

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
REALTEK 2832U Device Version: 64.1.521.2012 [21/05/2012]
REALTEK 2832U Device Version: 64.1.617.2010 [17/06/2010]
Bulk-In, Interface (Interface 0)
Bulk-In, Interface (Interface 0)
Bulk-In, Interface (Interface 0)

This driver is not digitally signed!

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 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 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 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 another good connector such as SMA/BNC/N should introduce negligible losses.

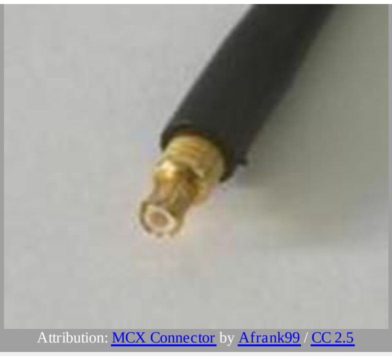
Dongles using the PAL connector will have poorer performance at high frequencies. Insertion loss expected at the ADS-B frequency (1090 MHz) is estimated to be about 1-3 dB (1.25x - 2x loss), but it can add up to more if you use PAL on both ends of the cable and start using adapters.

There is always the option of desoldering the MCX 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 if poorly constructed. However, pigtail adapters tend to be more common and are easier to use if mounting in a box for instance. 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 passbands. The table is arranged from worst 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		0 - 6 GHz

	 Attribution: MCX Connector by Afrank99 / CC 2.5	
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

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. RG6 is a 75 Ohm cable which matches the 75 Ohm input on the RTL-SDR. 50 Ohm cabling will work just as well however, as the mismatch loss between 75 and 50 ohms is negligible. Most electrical stores will stock RG6 cable as it is commonly used for satellite TV installations.

It is also recommended to try and reduce the 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.

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.

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

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 it is not more sensitive in any direction to another.

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 this isotropic antenna. The measure of decibels is a measure of the ‘gain’ strength. Larger dB numbers mean that your signal will be received stronger.

As with most things, there are tradeoffs when choosing an antenna. You can either have a highly directional high gain antenna for a specific frequency or an omni-directional but low gain antenna for a wide band of frequencies.

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 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. If you want to receive signals from all directions for general scanning of all frequencies, you will need an omnidirectional antenna like a Discone or Scantenna.

RADIATION PATTERNS

Radiation patterns show in what directions the antenna will receive best. In the example antennas shown below in the Example Antennas section we show 3D total power radiation patterns. The more red/pink the colour, the more gain in that particular direction. Remember that radio waves basically travel in straight lines from the source (ignoring reflections). Imagine straight lines coming from the signal source heading towards the antenna centre to see where on the radiation pattern the signal will hit.

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 is wideband antennas like the Discone or Scantenna, but they of course have less gain.

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 HF bands (0 - 30 MHz) have long wave lengths. High frequencies such as VHF/UHF (50 MHz+) have shorter wavelengths.

The size of the elements on an antenna are proportional to the wavelength. Therefore, HF antennas need to a large, and VHF/UHF antennas can be much smaller.

STANDING WAVE RATIO (SWR)

The SWR of an antenna is a measure that says how well matched the antenna impedance and coax 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.

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. For instance speaking in a room with an echo can make your voice harder to understand. The same thing happens with mismatched impedances.

A good SWR means that most of the signal makes it through the cable. A perfect SWR ratio is 1:1, a really bad ratio might be 100:1 or higher. Fortunately, SWR does not matter that badly when receiving, but it can still make a noticeable difference.

SWR can vary wildly with frequency. Non-broadband antennas will only likely have a small range of frequencies that can receive with a good SWR.

Note that SWR says nothing about the gain. The SWR may be good at a frequency, but the gain of the antenna may be poor at that frequency.

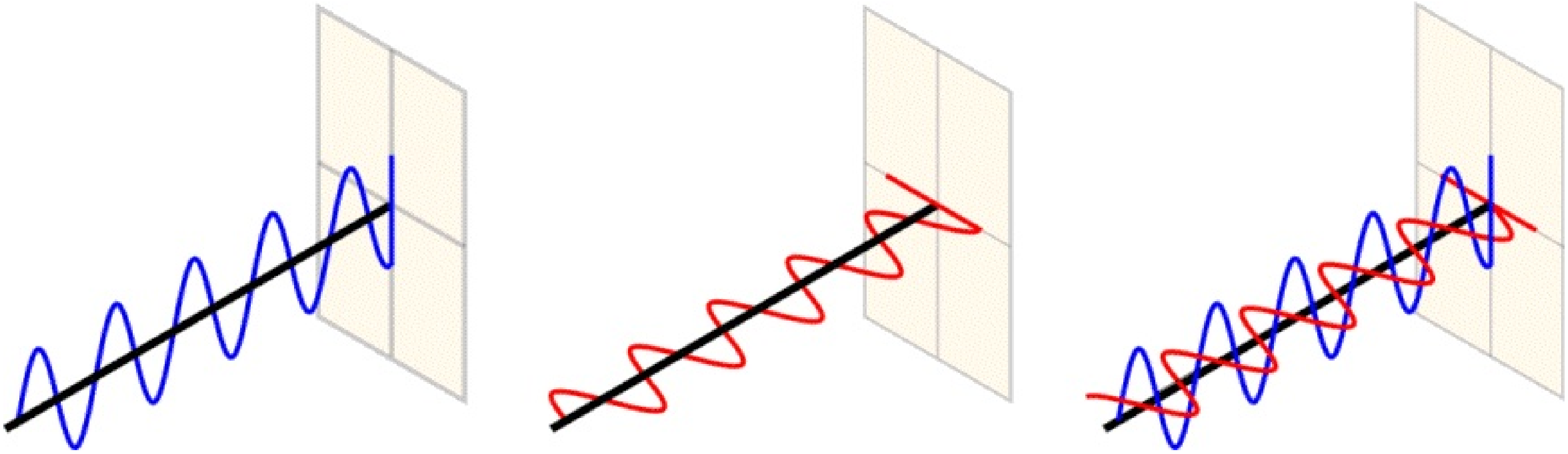
ANTENNA POLARIZATION

Radio signals can be transmit at different polarization 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.

Polarization refers to the orientation of the signal. 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 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.



ANTENNA POSITIONING

Antennas 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, it’s 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.

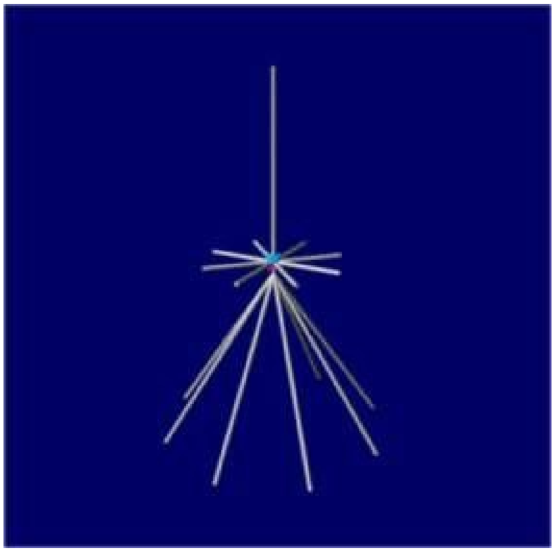
Antennas can also be mounted in the attic or roof space of many houses without significant reception degradation. Mounting in the attic also has the advantage of weatherproofing and negating lightning strike risks. The commonly used roof material asphalt shingles won’t block RF signals significantly. However, metal roofs or roofs with concrete tiles with metal rebar or grids in them can degrade signals much more.

VHF/UHF EXAMPLE ANTENNAS

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 in the free software 4NEC2 which can be downloaded from http://www.qsl.net/4nec2/.

DISCONE (BROADBAND ANTENNA)

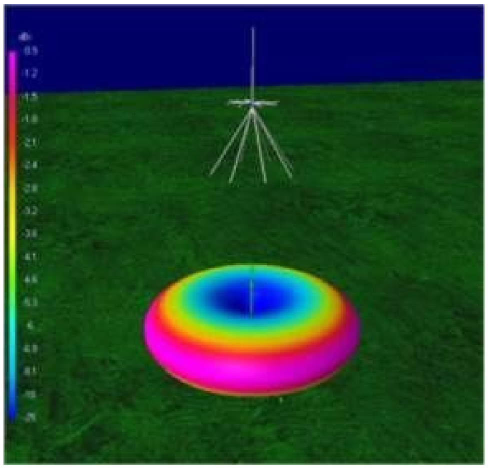


The Discone is typically recommended as the overall most useful antenna for the RTL-SDR. This is because a Discone has a very wide receivable bandwidth and can receive signals from almost all directions with directivity towards the horizon.

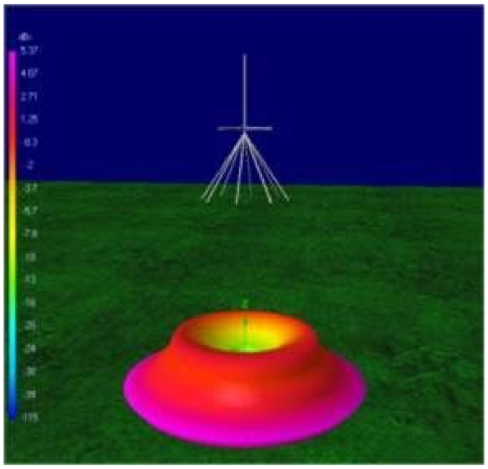
Commercially constructed discones can be found cheaply on Amazon (see our Buy RTL-SDR page at http://www.rtl-sdr.com/buy-rtl-sdr-dvb-t-dongles/) for around $50 USD.

In the 3D radiation pattern 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 over different frequencies. At 1090 MHz there is significant 'lobeing', which just means that the radiation pattern becomes spiky or non smooth.

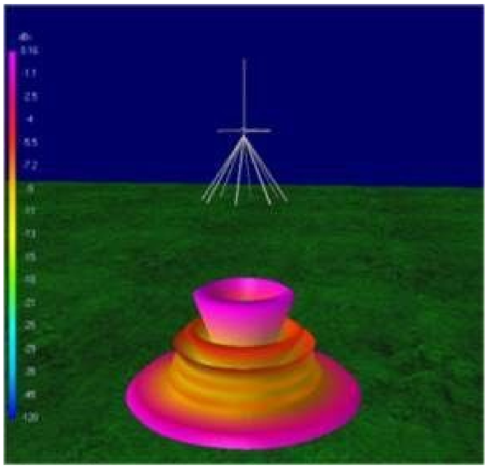
The VSWR also varies quite wildy, but remains usually below 10.



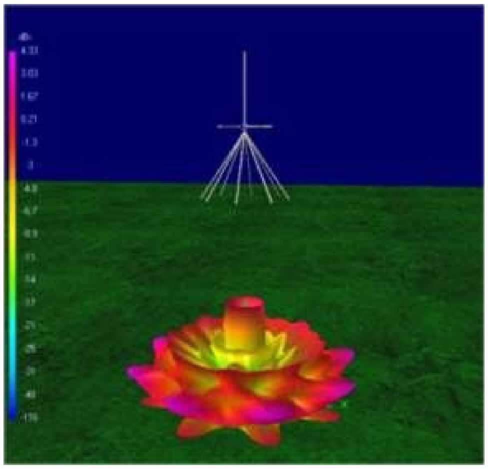
25 MHz



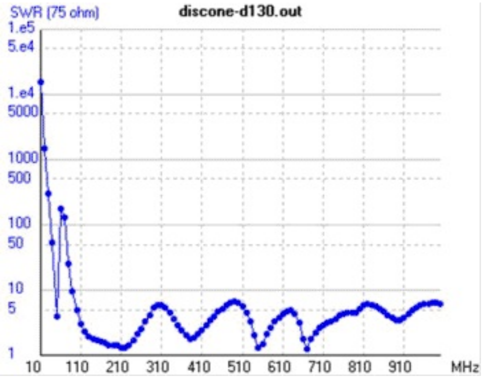
144 MHz



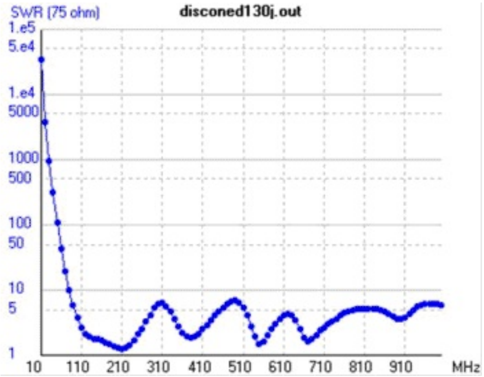
460 MHz



1090 MHz



With whip



Without whip

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/.

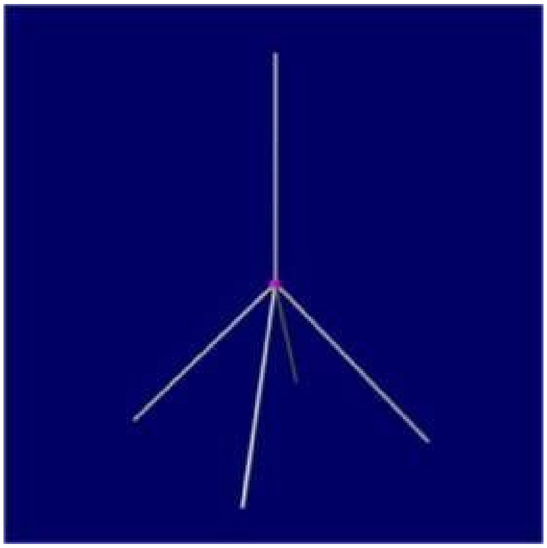
SCANTENNA (BROADBAND ANTENNA)

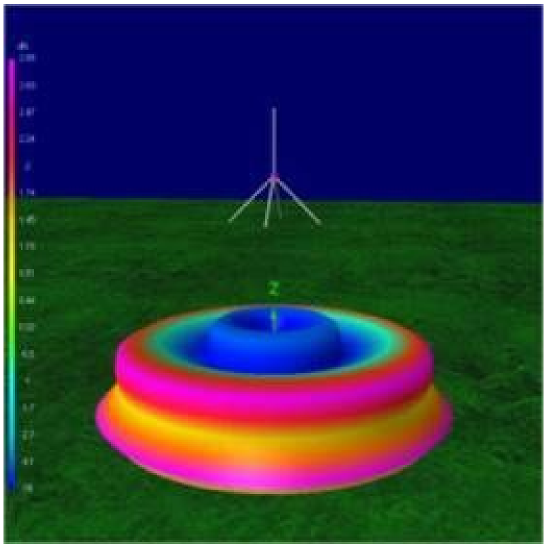
The scantenna is an antenna similar to the discone in that it has a very wide receivable bandwidth. Many people often state that the scantenna performs better than a discone. This is probably because the scantenna has greater gain out towards the horizon and is thus a better receiver for most common signal sources.

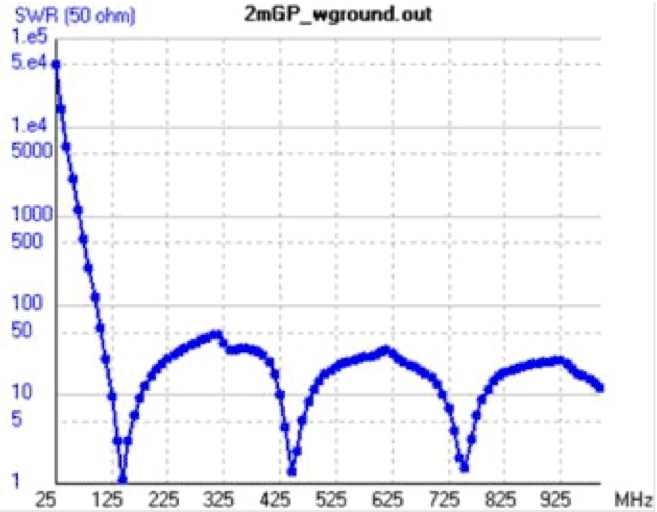
The most commonly bought scantenna has a frequency range of 30 - 1300 MHz. Beware that the scantenna is quite large measuring in at 144 x 19.5 x 8.6cm (56.8 x 7.7 x 3.4 inches).

See the RTL-SDR.com buy page http://www.rtl-sdr.com/buy-rtl-sdr-dvb-t-dongles/ to buy a Scantenna.

¼ WAVE MONOPOLE GROUND PLANE







The ¼ wave ground plane is essentially a straight wire cut to ¼ wave length of the design frequency with several spokes 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 ¼ wave groundplane with good ground (enough radials) is equivalent to a dipole.

The radiation pattern directs most of it's gain out towards the horizon.

The VSWR of a ¼ wave ground plane is good at it's design frequency and at a few other higher frequencies. In our example antenna with design frequency of 145 MHz, the VSWR is minimum at 145 MHz and also at 445 MHz and 755 MHz.

Information on construction of an ADS-B ¼ wave ground plane antenna can be found at http://www.atouk.com/wordpress/?wpdmact=process&did=Ni5ob3RsaW5r.

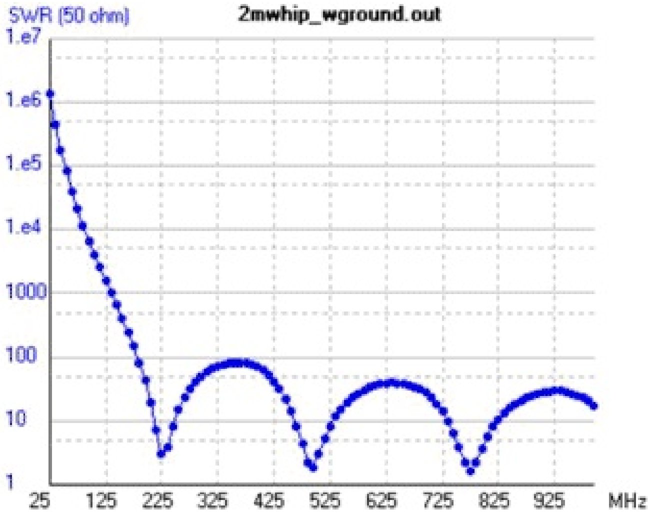
Here is another page showing the design of this antenna. http://ketil.com/la2dna/antenner/PMR446_ground_plane_antenna.htm

A calculator for calculating ¼ wave ground plane element lengths for specific frequencies can be found at http://www.csgnetwork.com/antennagpcalc.html.

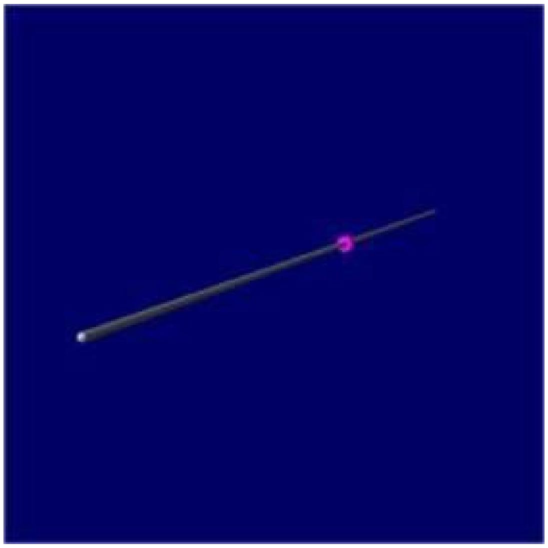
WHIP

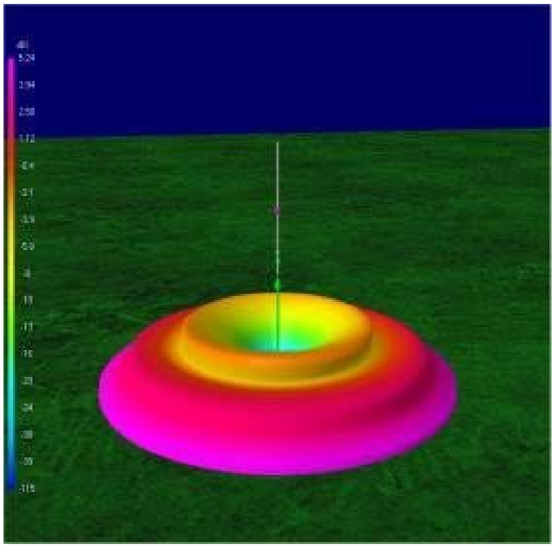
A whip antenna is simply a ¼ wave ground plane antenna without a proper ground plane. It is then just a wire that is ¼ the length of the design wavelength. A whip antenna will change the VSWR design frequency and cause the VSWR to not be as low as with a proper ground plane.

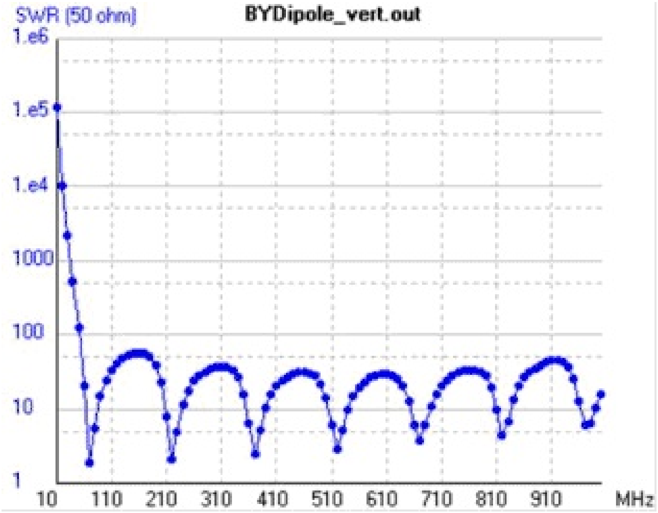
Whip antennas (such as the one shipped with the standard RTL-SDR) are usually designed with a magnetic base so that they can be placed on metallic objects such as car roofs to act as a ground plane.



DIPOLE







A dipole is essentially two wire elements pointing away from each other. The wires are not connected at the centre. One element connects to the centre coax and the other connects to the shield.

The dipole can be made horizontally or vertically polarized by placing it in the horizontal and vertical orientations respectively. The radiation pattern is 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.

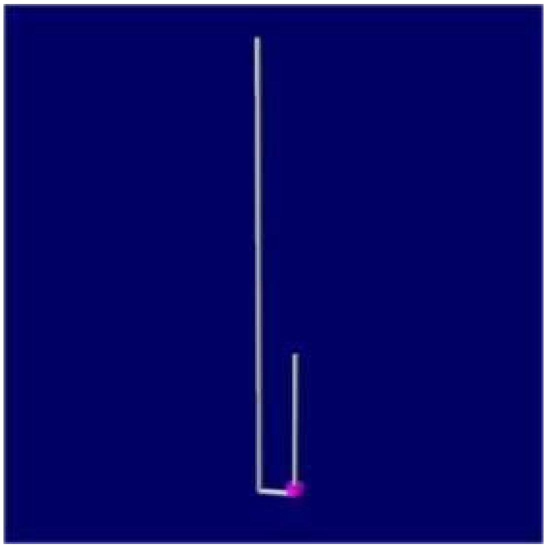
Compared to the ¼ wave groundplane the dipole has the exact same radiation pattern and gain. However, the disadvantage to a dipole is that it needs to be larger and it is more difficult to mount the feedline is using it vertically polarized. The feedline should run perpendicularly away from the dipole to avoid altering the radiation pattern.

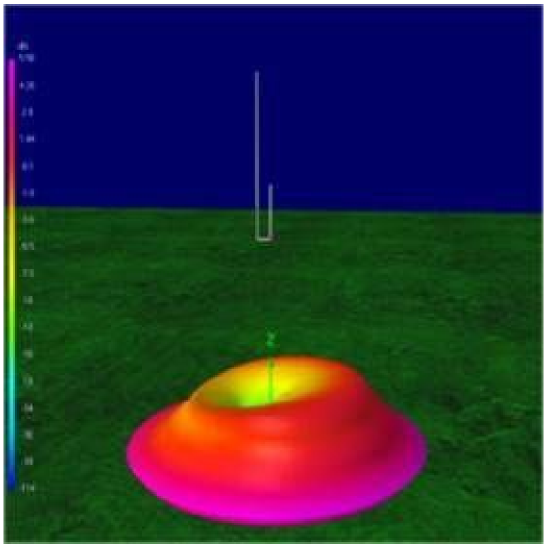
More information about dipoles can be found at http://www.radio-electronics.com/info/antennas/dipole/dipole.php.

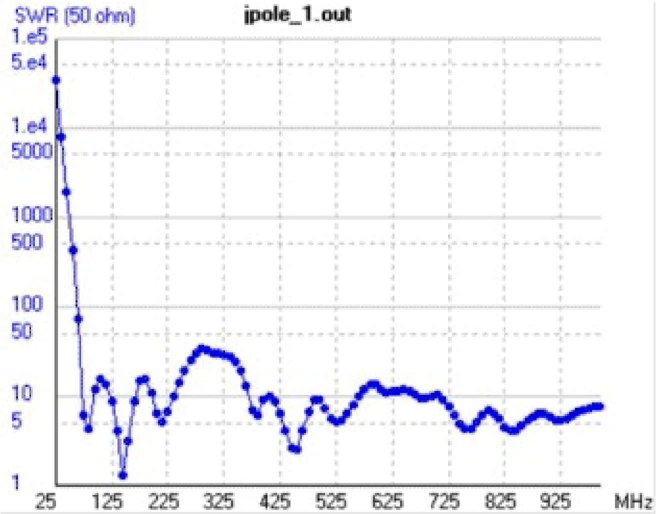
Here is a dipole length calculator http://www.angelfire.com/mb/amandx/dipole.html.

Information on a very simple ADS-B dipole can be found at http://antirez.com/news/46.

J-POLE / SLIM JIM







The J-pole antenna is named as such as it looks like the letter ‘J’. These antennas usually have good gain at their intended frequency. An advantage to J-poles is that their impedance can be matched to the coaxial cable easily by moving the connection points up and down. A J-pole can give 6.5 dBi gain or more.

The J-pole in the example images shown above is designed for 145 MHz.

A J-Pole schematic and calculator can be found at http://www.hamuniverse.com/jpole.html.

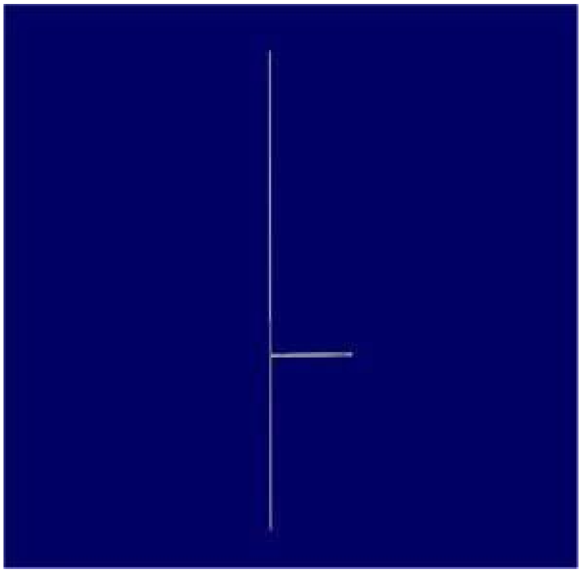
The slimjim is an antenna that is very similar to the J-Pole. Slim jims have a radiation pattern that favours signals coming from the horizon a little more and so are usually a better choice for most applications.

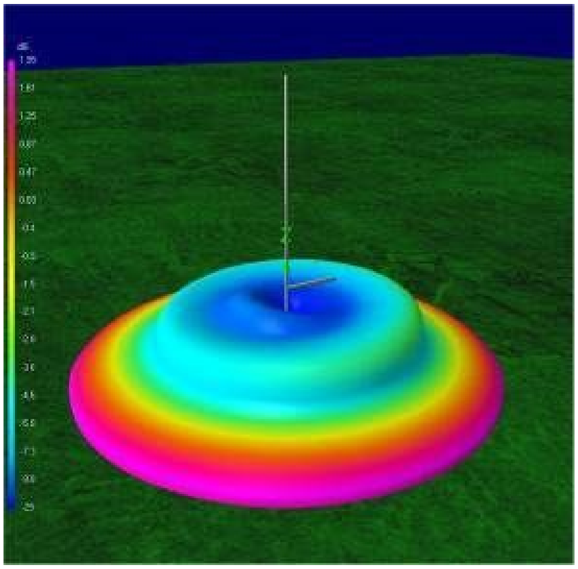
Slim jims can easily be constructed out of twin lead wire. Here is a tutorial example http://www.g4aym.org.uk/projects/project3/page0.htm

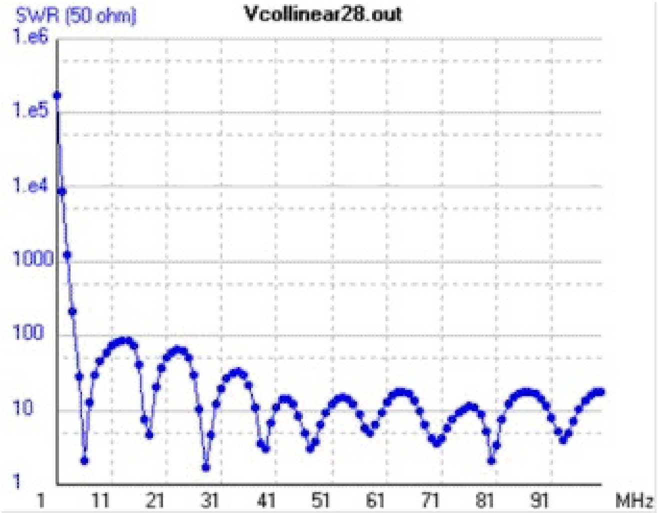
Here is a slim jim calculator http://www.m0ukd.com/Calculators/Slim_Jim/

And here an example of a Slim Jim cut for AIS http://nmearouter.com/docs/ais/slimjim.jpg

COLLINEAR







Collinear antennas can have very high gain directed towards the horizon. They are easily built out of coax cable, or copper wire. As they have high gain they also tend to have a theoretical narrow frequency range, though in practice they will receive well at most frequencies in the gain area. The more elements added to a collinear, the greater the gain will be and the flatter the radiation pattern will be. With enough elements the radiation pattern becomes disc like.

For ADS-B and most applications a collinear antenna with too many elements will likely yield poorer performance compared to one with less. This is because the radiation pattern will become far too flat.

Collinear antennas can easily be built out of coaxial cable. Coaxial collinears are sometimes referred to as CoCo antennas. For CoCo antennas we recommend around 4 to 6 elements as the maximum for ADS-B.

Collinear antennas work very well for ADS-B and AIS reception as well as terrestrial voice communications frequencies.

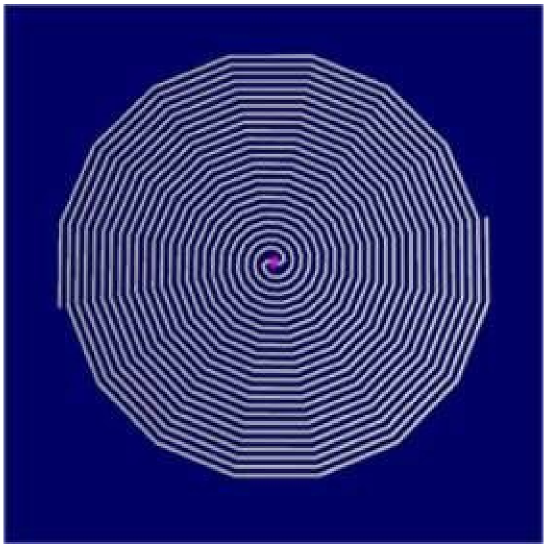
The antenna in the images shown above is designed for 28.8 MHz.

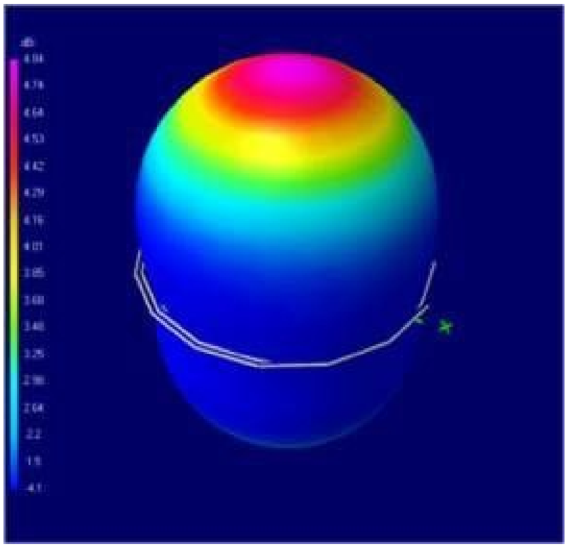
See here for information on constructing an ADS-B coaxial collinear http://www.balarad.net/. Here is a video showing the construction process of that antenna https://www.youtube.com/watch?v=TkUYdCPFXXs.

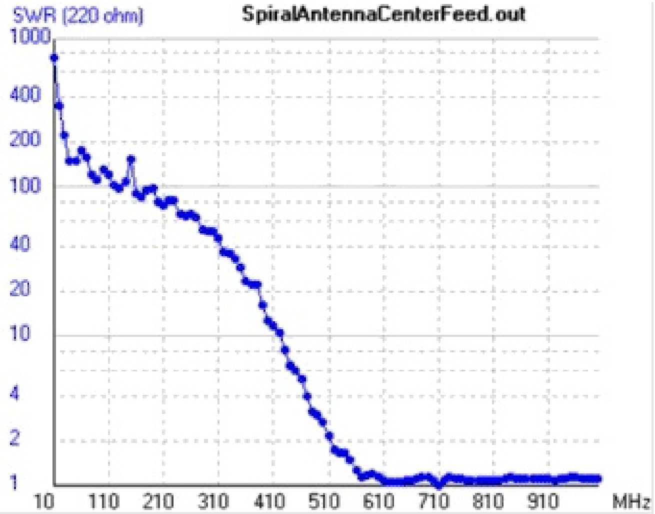
See here for information on constructing a coaxial collinear for AIS http://nmearouter.com/docs/ais/aerial.html

Here is a wire collinear for WiFi, but it's length's could be adapted for other frequencies http://martybugs.net/wireless/collinear.cgi using the calculation they specify.

SPIRAL







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.

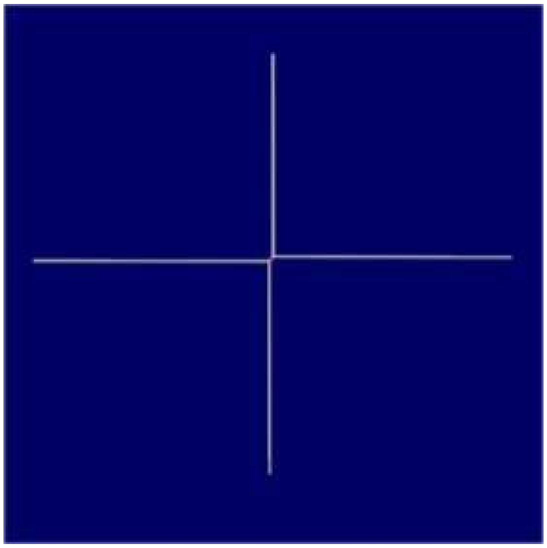
As these antennas are very wideband, their gain isn't that great.

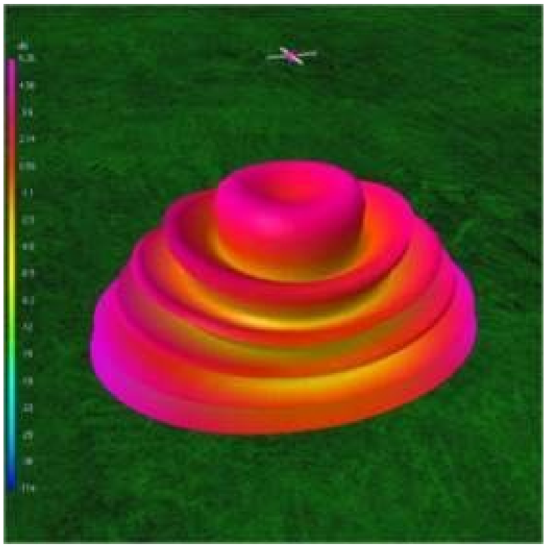
These antennas can easily be built out of coaxial cable.

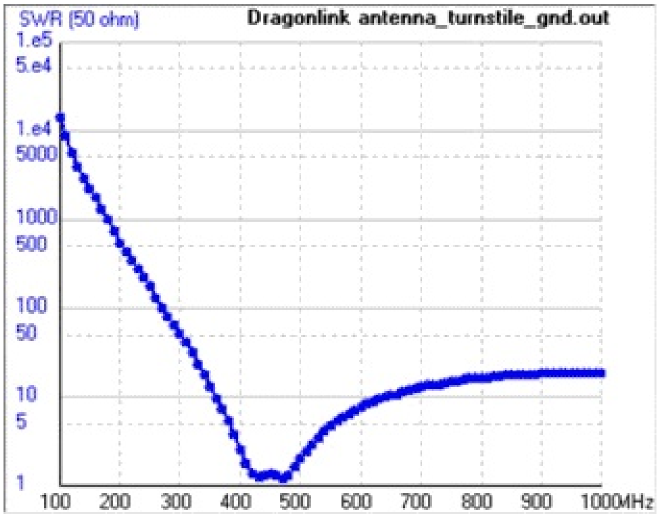
The antenna in the example images shown above is designed for 500 MHz and above.

More information about this type of antenna can be found at http://www.antenna-theory.com/antennas/travelling/spiral.php.

TURNSTILE / CROSS DIPOLE







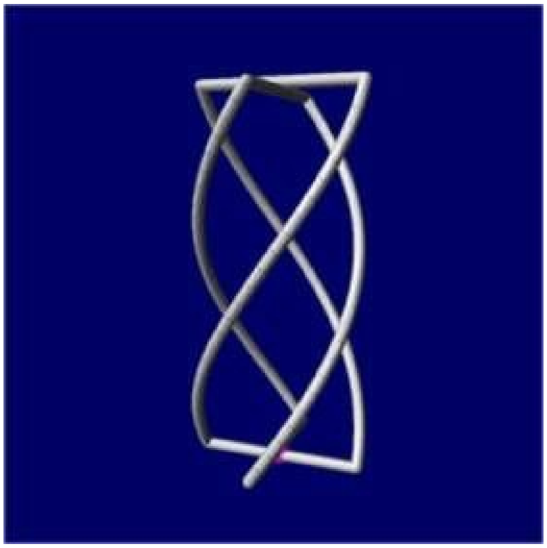
Turnstile antennas are circularly polarized antennas that are often used for FM reception, but can also be used for satellites due to their circular polarization. Turnstiles can operate in two modes, normal or axial. Normal mode creates a horizontally non-directional (omnidirectional) radiation pattern directed towards the horizon. In axial mode the turnstile gain is directed towards the sky. The radiation pattern shows the turnstile in axial mode. For satellites the axial mode turnstile is required.

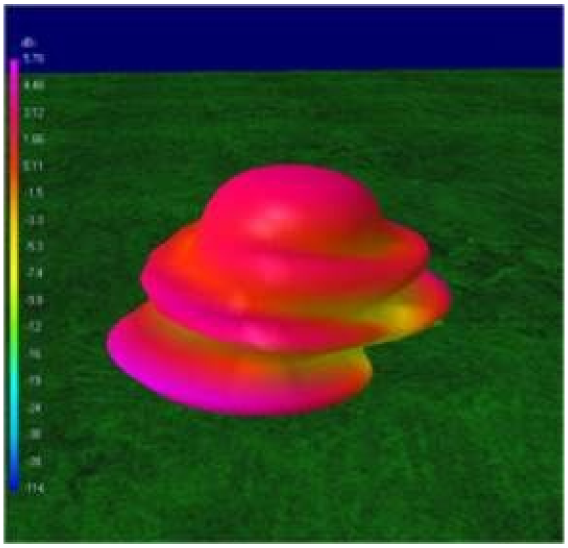
This turnstile in the above example images has a design frequency of 450 MHz and is an axial mode design. The VSWR is near 1 at this frequency and quickly ramps up to high ratios at other frequencies.

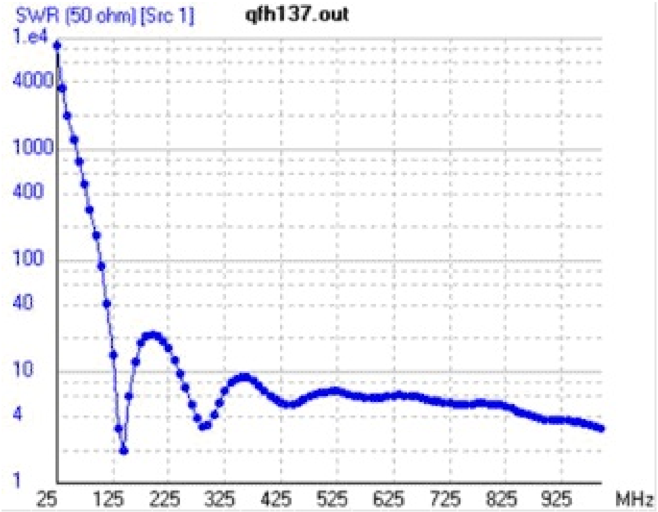
Here is a turnstile build for NOAA satellites http://umarca.blogspot.co.nz/2013/05/project-8-turnstile-antenna-for-weather.html.

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.

QUADRIFILAR HELIX (QFH)







Quadrangular antennas are circularly polarized antennas that are best used for satellite reception. They tend to work better than turnstile antennas for this purpose especially with NOAA satellites. From the radiation pattern we can see that they have fewer null areas compared to a turnstile.

This QFH has a design frequency of 145 MHz. At this frequency it has good VSWR, but the VSWR remains decent at higher frequencies as well.

Here is a link which compares the Turnstile vs the QFH antenna. His results show that the QFH is slightly better. http://www.satsignal.eu/wxsat/antennas/

Here is a link for a QFH antenna for NOAA satellites http://forum.allaboutcircuits.com/blog_attachment.php?attachmentid=325&d=1389045274

A simple tutorial on creating a coax cable QFH can be found here http://sdrformariners.blogspot.com/2013/08/weather-satellites-antennas.html.

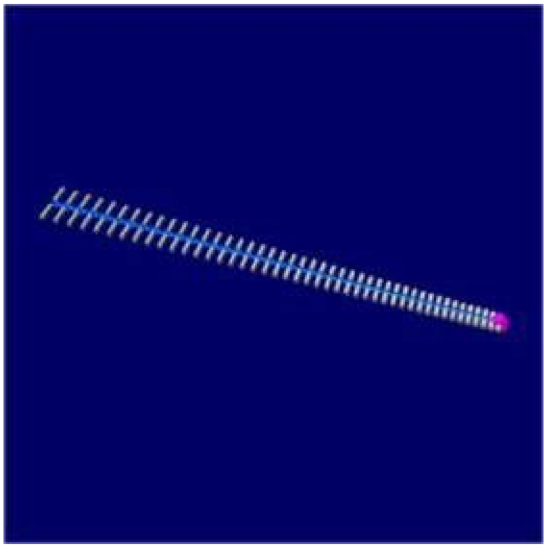
Here is another interesting build with copper pipe http://abdallah.hiof.no/QFH/.

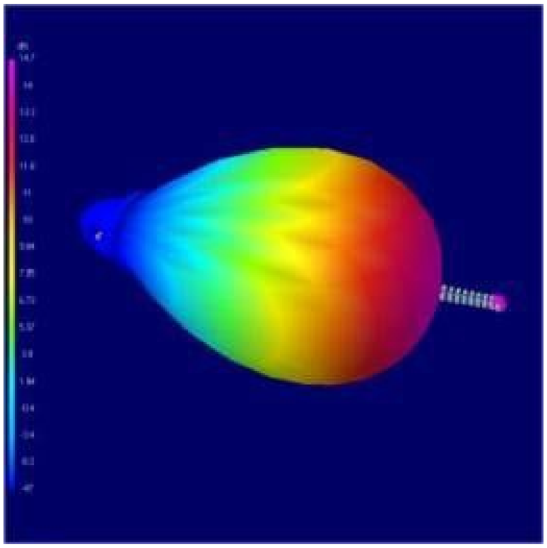
ACTIVE GPS PATCH

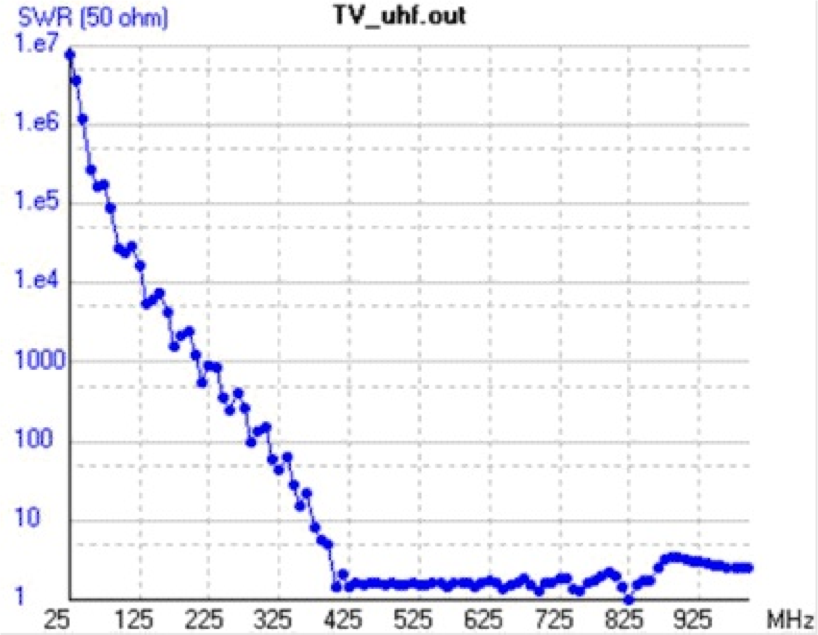
Most active GPS antennas require DC power to be sent down the centre conductor of the coax. Be careful with this, DO NOT let DC power enter the RTL-SDR dongle or it will break.

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







Yagi antennas are highly directional over a small bandwidth with very large potential gain. Their signal polarization is oriented in the same direction the orientation of the elements.

The example antenna in the above images is designed for 475 MHz.

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.

http://www.vk6fh.com/vk6fh/162mhzyagi.htm http://www.grafdxradiothings.blogspot.com/2012/03/on-air-home-made-realization-ais.html

It is also possible to build circularly polarized Yagi's 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. Circularly polarized antennas are also called Crossed-Yagi antennas.

PCB ADS-B ANTENNA

There have been some ADS-B antennas designed on PCB boards with LNA's built into the PCB.

For example see

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 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 antenna and is can be quite effective for what it is.

The radiation pattern of a random wire antenna is quite unpredictable. Therefore this antenna may require a bit of experimentation with location and lengths to get the best performance.

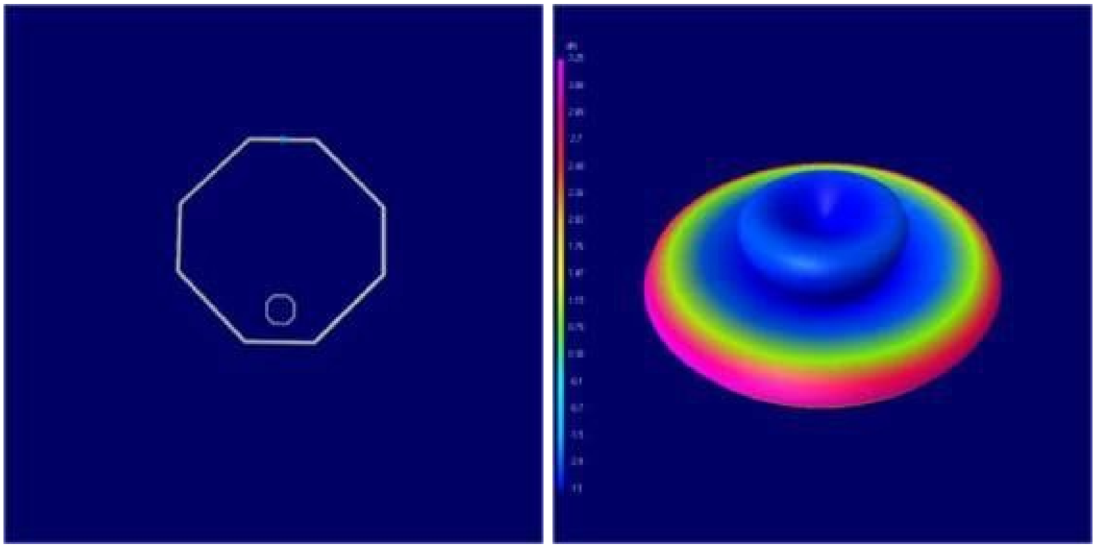
To make this antenna simply connect a coax cable to the antenna input of your upconverter or direct sampling modded RTL-SDR. Run the coax to the location at which your random wire antenna will start. Now connect the random wire antenna to the centre wire of the coaxial cable.

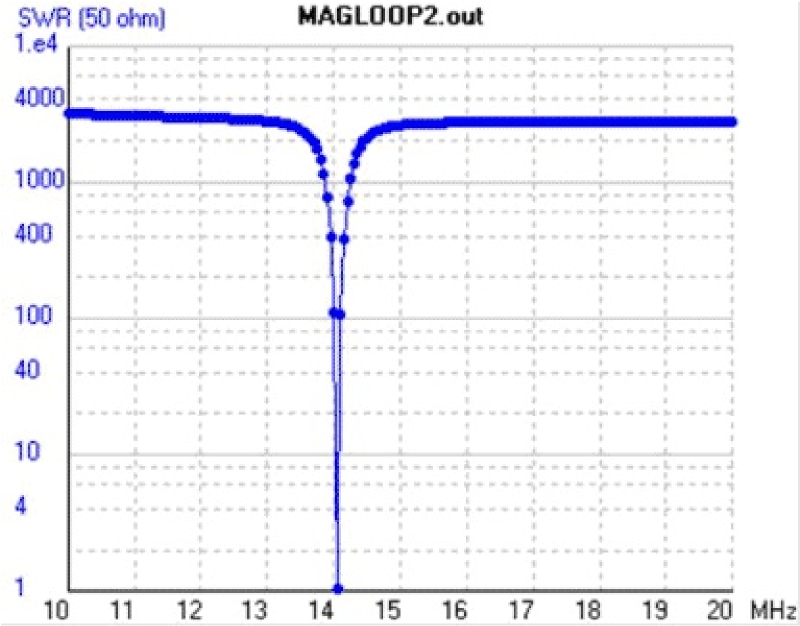
To significantly decrease the amount of noise, connect the shield part of the coaxial cable to a ground. A good ground is a cold water pipe. Avoid using the electrical ground as it 1) could be dangerous and 2) is not the best ground due to electrical interference. If you don't have a good ground accessible, 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 an antenna tuner.

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. This means 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 antenna itself to tune it to the correct frequency as well.

The advantage is that the magnetic loop will filter out any interference on signals you aren’t tuned to. The disadvantage is that the loop will 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.

In the radiation pattern pictures above the magnetic loop is tuned for 14 MHz. Notice how the SWR is very narrow around 14 MHz.

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.

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 simply 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.

APPENDIX A: AUDIO PIPING

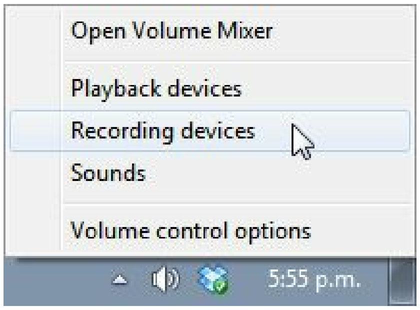
Many applications of the RTL-SDR require that the audio from an SDR receiver such as SDR# be "piped" into decoding software. Piping is simply a way to allow the decoding program to hear the output audio produced by the SDR receiver.

There are several methods to pipe audio in Windows, below we show how to enable the Stereo Mix, VB Cable and Virtual Audio Cable options.

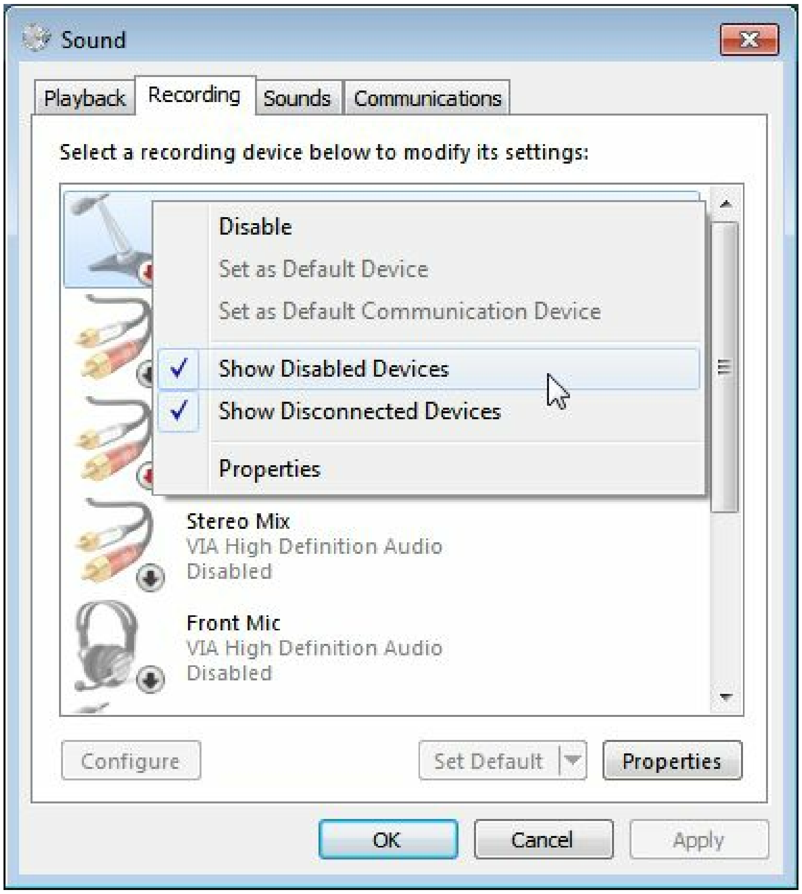
STEREO MIX

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.

1. Right click the speaker icon in your task bar and click on "Recording devices".

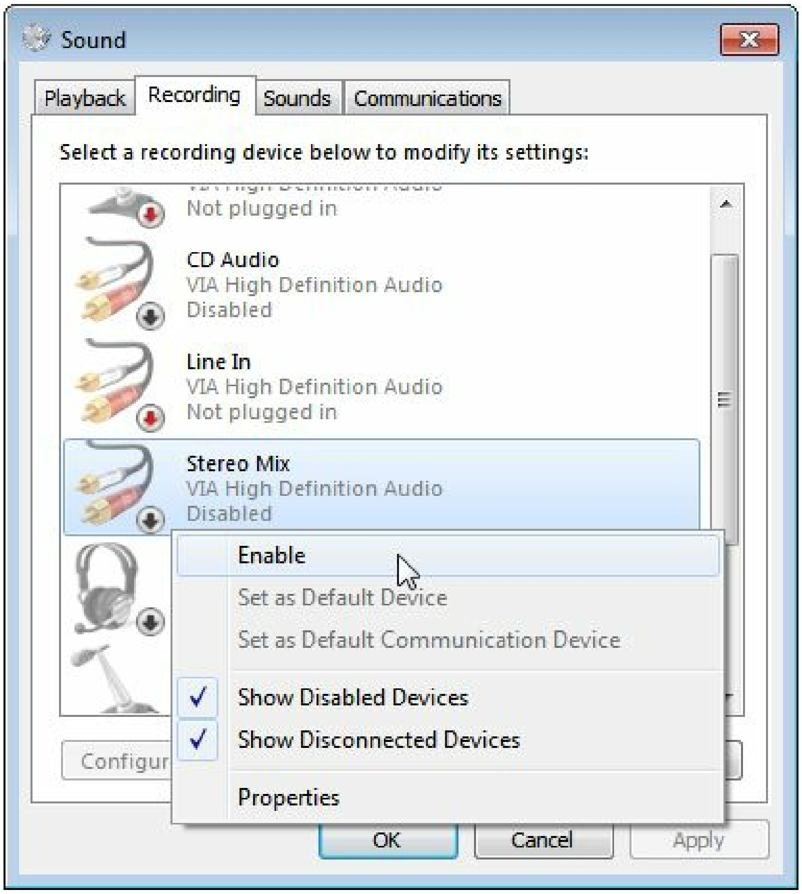


2. Right click anywhere in the window that pops up and ensure that "Show Disabled Devices" and "Show Disconnected Devices" are both checked.



Select a recording device below to modify its settings:

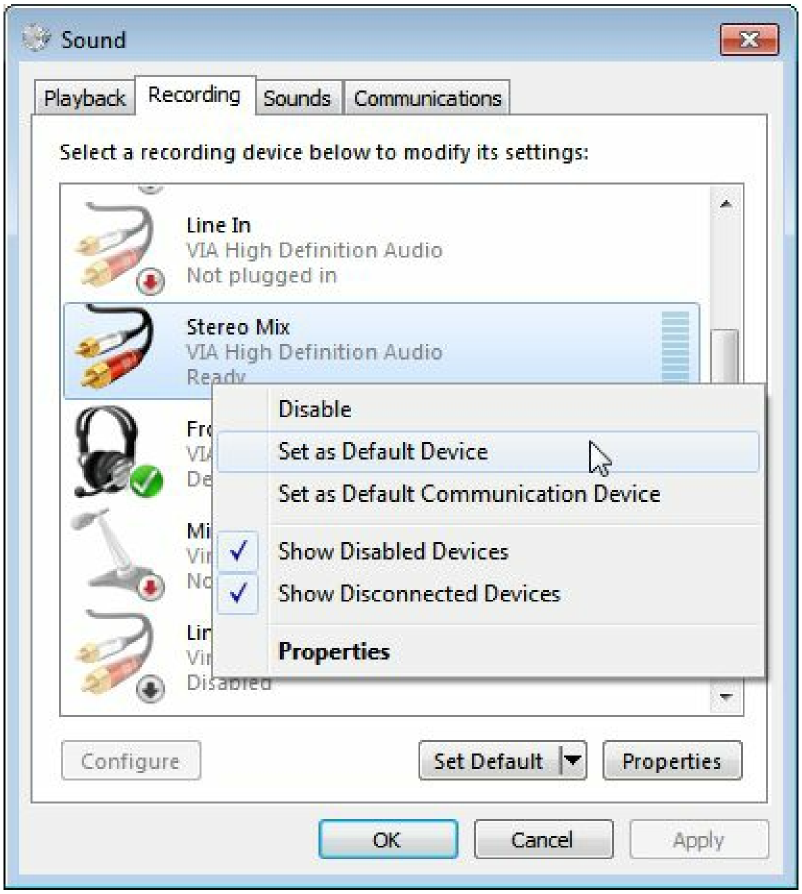
3. Find the “Stereo Mix” option, right click it and select Enable.



Select a recording device below to modify its settings:

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 only when looking for an audio source. Note that if you

want to use virtual audio cable as the default device again, you will need to set it 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

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

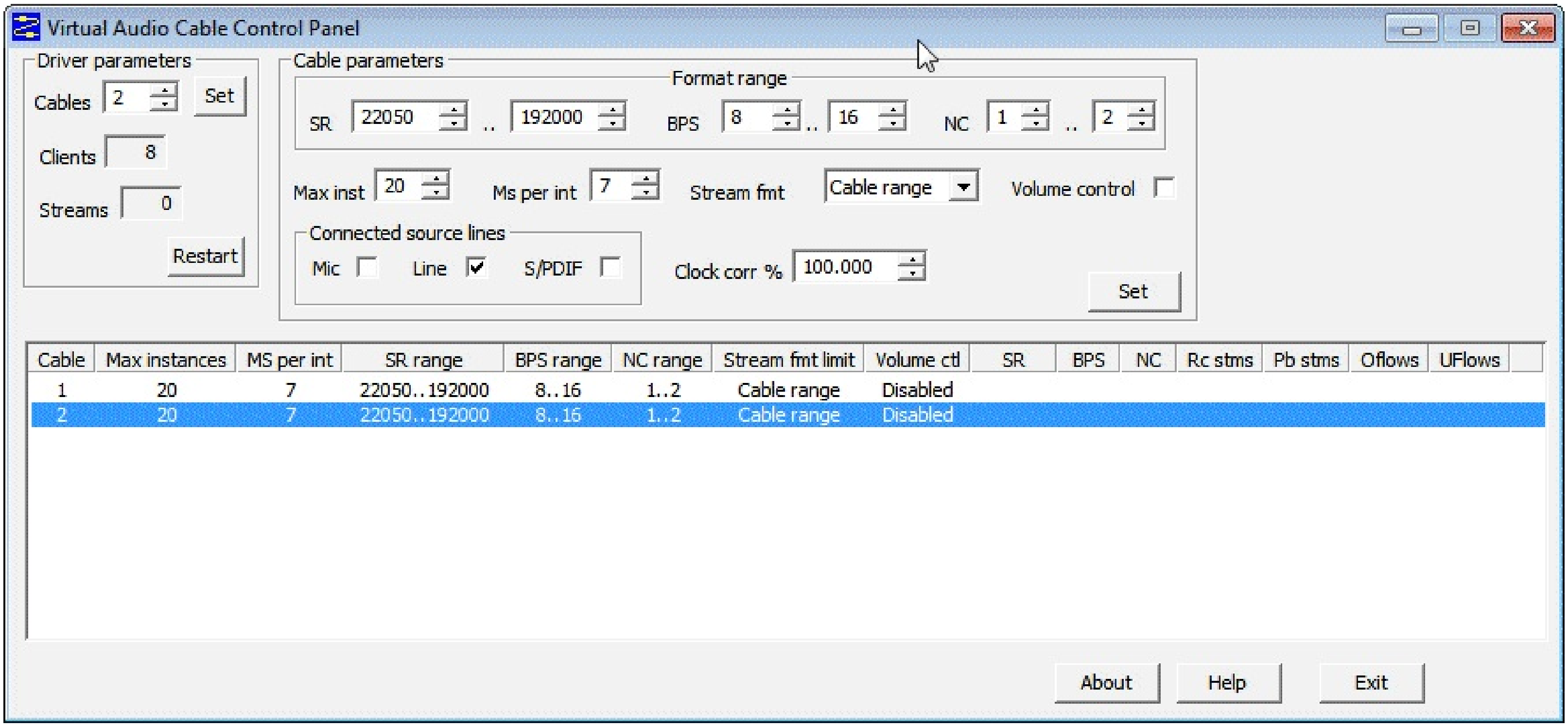
VB Cable is free software that can be downloaded from http://vb-audio.pagesperso-orange.fr/Cable/. VB Cable installs a new audio device that acts in a similar way to stereo mix. The advantage with a program like VB Cable over stereo mix is that the audio will not play through your speakers. This is desired 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 (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.

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 also 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.



Cables 2

Clients 8

Streams 0

SR 22050 .. 192000

BPS 8 .. 16

NC 1 .. 2

Max inst 20

Ms per int 7

Stream fmt Cable range

Volume control □

Mic □ Line ☑ S/PDIF □

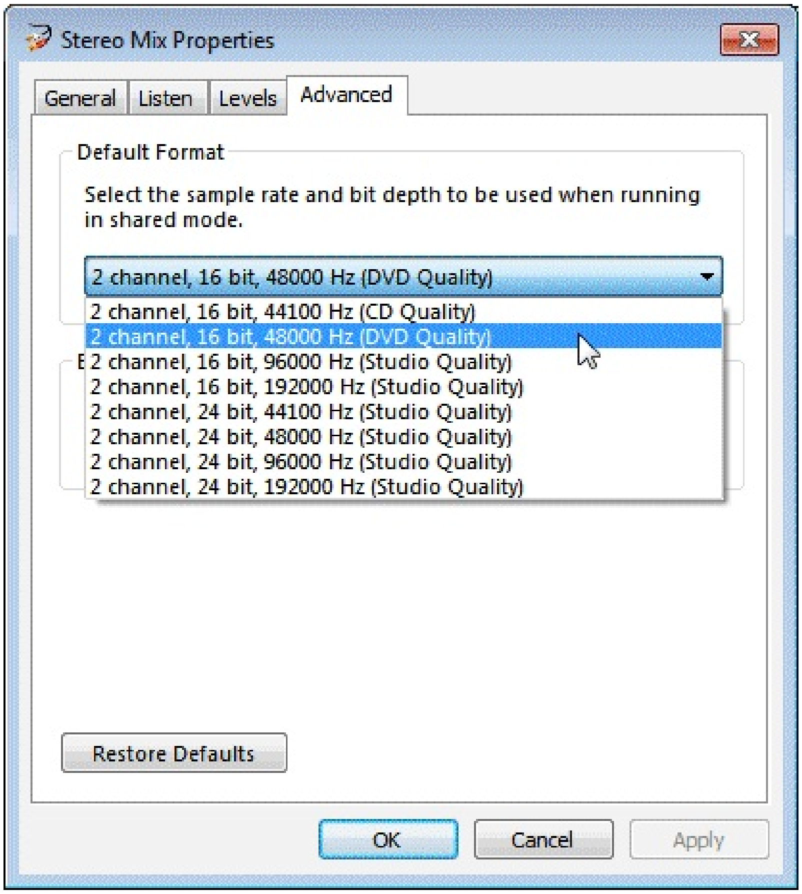
Clock corr % 100.000

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 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 a simple form, radio waves can be thought of as a form 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 can correspond 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 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

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 spectrum. Amateur radio occurs on the HF frequencies and AM broadcast radio is on the MF frequencies.

PROPAGATION

Radio waves travel in a straight line, just like light does. However, note that light does bounce around. A light shone at a single wall will still illuminate an entire 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 can 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’.

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.

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 into a varying frequency within the bandwidth. Broadcast radio uses Wideband FM (WFM), which is simply FM with a big 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 modes such as Lower Sideband (LSB) and Upper Sideband (USB) are modes like AM, but only use half the bandwidth of a corresponding AM signal. Transmitting on SSB requires less bandwidth and less transmission power. The disadvantage is that it requires finer tuning to get a signal that sounds right. Data modes on HF are often transmitted with USB signals.

DECIBELS (DB) TO TIMES

Decibels can be a little confusing for those unaccustomed to it. 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 amplifies the signal by 20 dB”, or “the coax cable attenuates the signal by 3 dB”. A 20 dB increase in signal means that the signal is increased by 100x. A 3 dB 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 numbers that are much more manageable.

The formula to convert decibels to times is as follows, where X is the value in decibels.

$$\text{ratio} = 10\ \hat{}\ (X / 10)$$

To convert to decibels from a ratio or unit use

$$\text{power (dB)} = 10 * \log(\text{ratio})$$

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 0.000001mW just before the antenna. This corresponds to -90 dBW or -60 dBm. Our example antenna has a 3 dBi gain in the direction that the signal came from. Then we pass it through an 20dB LNA, loose 5 dB in the coax and then amplify again at the dongle by a further 20dB. Our total signal strength is then given by.

$$-60\ \text{dBm} + 3\ \text{dBi} + 20\ \text{dB} - 5\ \text{dB} + 2\text{-}\ \text{dB} = -22\ \text{dBm}$$

In amateur radio a signal said to be at S9 (good reception) is a signal that is -73 dBm strong.

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.

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 general RTL-SDR users is over on www.reddit.com/r/rtlsdr. There are also good communities on facebook, check out our RTL-SDR Blog page and the RTL-SDR (Closed Group) group.

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: http://webchat.freenode.net/?channels=##rtlsdr

RTL-SDR Forum for ADS-B: http://radarspotting.com/forum/index.php/board,51.0.html

ERRATA UPDATES

Edition 1.1 - 20 May 2014

- Added Errata 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 transmission of any radio signal with a transmit capable radio unless you are licenced to do so.