**RTL-SDR TUTORIAL: CHEAP AIS SHIP TRACKING**

[**http://www.rtl-sdr.com/rtl-sdr-tutorial-cheap-ais-ship-tracking/**](http://www.rtl-sdr.com/rtl-sdr-tutorial-cheap-ais-ship-tracking/)

Large ships and passenger boats are required to broadcast an identification signal containing position, course, speed, destination, and vessel dimension information to help prevent sea collisions. This 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 software defined radio](http://www.rtl-sdr.com/buy-rtl-sdr-dvb-t-dongles/) (or a more advanced SDR such an [Airspy](https://www.itead.cc/spyverter.html?acc=cfcd208495d565ef66e7dff9f98764da" \t "_blank)) 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, Airspy 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.

Note, tracking ships with AIS is very similar to [tracking aircraft with ADS-B](http://www.rtl-sdr.com/adsb-aircraft-radar-with-rtl-sdr/), which is another project that may interest you.

**Examples of AIS received with RTL-SDR**

An AIS radar example is shown by YouTube user [Vinicius Lenci](https://www.youtube.com/user/viniasp21?feature=watch" \t "_blank) who uses an RTL-SDR, SDRSharp and ShipPlotter. This video also shows what a strong AIS signal sounds like.

Another example of AIS radar displayed on Google maps using an RTL-SDR, SDR Sharp and ShipPlotter is shown by YouTube user [Superphish](http://www.youtube.com/user/Superphish" \o "Superphish" \t "_blank).

[Second Attempt with RTL SDR (RTL2832), SDR Sharp and ter](http://www.youtube.com/watch?v=CThfM9tPKGM" \o "Play Video \"AIS Ship Position Second Attempt with RTL SDR (RTL2832), SDR Sharp and ShipPlotter\")

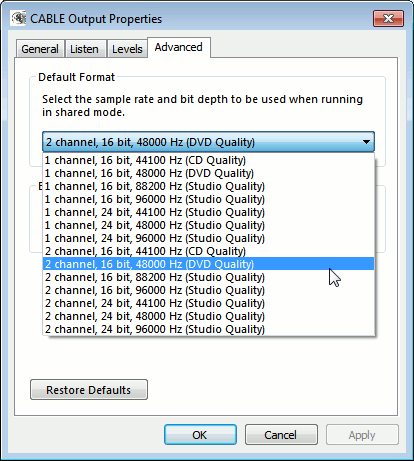
**Requirements and Setup**

To set up an AIS ship radar on a windows system you will need four things.

1. An [RTL-SDR dongle](http://www.rtl-sdr.com/buy-rtl-sdr-dvb-t-dongles/) working with SDRSharp. (Or other more advanced SDRs such an [Airspy](https://www.itead.cc/spyverter.html?acc=cfcd208495d565ef66e7dff9f98764da" \t "_blank))
2. An audio piping method.
3. A vertically polarized antenna tuned to 162MHz.
4. 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 the [Buy RTL-SDR](http://www.rtl-sdr.com/buy-rtl-sdr-dvb-t-dongles/) page for information, and the check out the [Quickstart Guide](http://www.rtl-sdr.com/rtl-sdr-quick-start-guide/" \o "Quick Start Guide" \t "_blank) 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. You can use either [windows stereo mix](http://www.rtl-sdr.com/tutorial-how-to-enable-stereo-mix-in-windows-7/), [VB-cable](http://www.rtl-sdr.com/free-virtual-audio-cable-alternative/) (free) or [virtual audio cable](http://software.muzychenko.net/eng/vac.htm) (paid with trial version).

The sampling rate of your audio piping method must be set to 48000 samples/sec. To set this in Windows, right click your device in the Windows sound recording tab, go to properties and under the advanced tab, set the sample rate to 48000Hz. Do the same to the same device under the Playback tab as well.

[](http://rtlsdrblog.rtlsdrblog.netdna-cdn.com/wp-content/uploads/2013/04/SoundPropertiesAdvanced1.png)

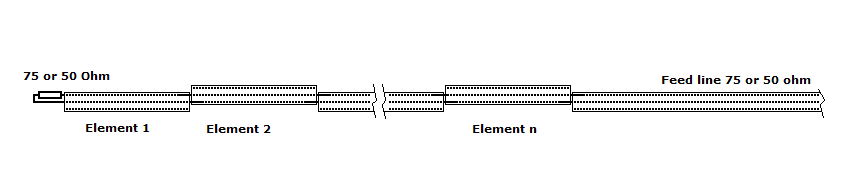
**AIS Antennas**

AIS signals are broadcast at both 161.975 MHz and 162.025 MHz and have a maximum range of approximately 74 kilometers. So if your radio set up is more than 74 kilometers away from any boats, you will probably not be able to receive AIS signals. AIS is also considered a line of sight signal, meaning that if there are large buildings or mountains in the way of your antenna and the boats, AIS signals could be blocked. Because of this reason it is important to put your antenna as high up as possible.

There are multiple commercial AIS antennas designed for marine use that will work. However, sometimes home made antennas work even better and they of course are cheaper. Some home made AIS antennas are shown below.

**Coax Collinear Antenna**

A good antenna for AIS is the coax 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.

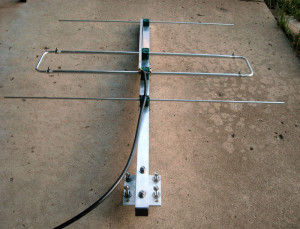
A collinear coax antenna is basically a length of multiple short coax cables, where the coax outer conductor is connected to the inner conductor in an alternating fashion. As it is made out of coax cable almost entirely, it is a very cheap antenna to build.[](http://rtlsdrblog.rtlsdrblog.netdna-cdn.com/wp-content/uploads/2013/04/CollinearAntenna.gif)

**Yagi-Uda (Yagi)**

If the majority of ships in your area are focused in one direction only, a high gain Yagi antenna may be a good choice. A Yagi is a very directional antenna, meaning that it will only pick up signals in the direction it is pointed. The advantage is that with directionality we can get a very large signal gain, which allows reception of further and weaker signals.

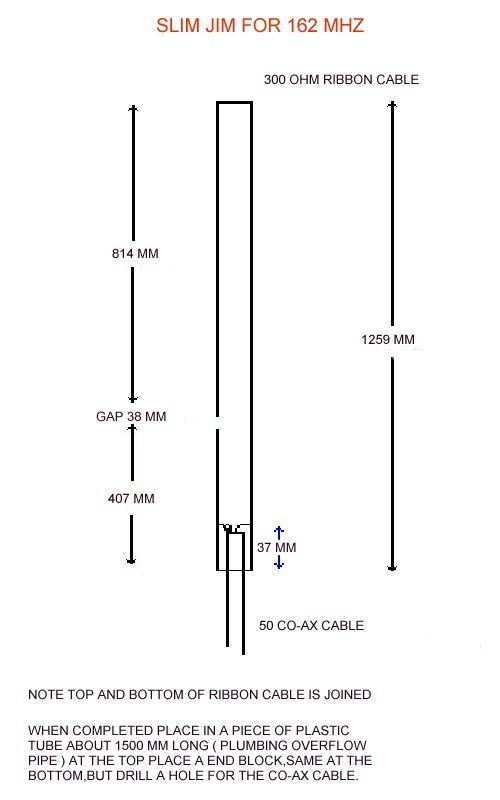
Building a Yagi is definitely a tougher job compared to the building a coax collinear antenna, but it is much more compact and can have a similar gain to a coax collinear. One page on a homemade AIS Yagi’ can [be found here](http://www.vk6fh.com/vk6fh/162mhzyagi.htm) and another [here](http://www.grafdxradiothings.blogspot.com/2012/03/on-air-home-made-realization-ais.html).

A commercial AIS Yagi can be [bought from here](http://www.innovantennas.com/antennas-a-accesories.html?page=shop.product_details&flypage=flypage.tpl&product_id=350&category_id=37) and a review of this antenna can be found [from here](http://continuouswave.com/whaler/reference/AIS_Antenna.html).

[](http://rtlsdrblog.rtlsdrblog.netdna-cdn.com/wp-content/uploads/2013/04/G0KSC_3el_LFA_Yagi_160.jpg)

**Slim-Jim**

The slim-jim is another high gain antenna that is reported to work well with AIS. A slim-jim antenna can be easily made on a budget using common [300 Ohm twin lead ribbon cable](http://en.wikipedia.org/wiki/Twin-lead). A page showing a twin lead slim-jim [is here](http://www.g4aym.org.uk/projects/project3/page0.htm), but that version is designed for 145 MHz. To calculate the correct lengths for AIS, [this calculator](http://www.m0ukd.com/Calculators/Slim_Jim/) can be used by inputting 162 Mhz into the frequency box. An image showing a slim-jim with AIS dimensions [is here](http://web.arundale.co.uk/docs/ais/slimjim.jpg).



**Software Tutorials**

First, use SDRSharp to tune to an AIS signal. Open SDRSharp, set your audio piping method in the Audio output drop down box, and tune to an AIS frequency (161.975MHz or 162.025MHz). In SDRSharp, the signals may not appear exactly on the AIS frequencies, since the RTL-SDR is not frequency accurate (unless you have a RTL-SDR with TCXO). Just tune manually until the signals are properly centred, or set the PPM offset correctly.

Play with the RF gain in the SDRSharp configure button 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. Take note of the optimal gain setting for later.

Set the receive mode to NFM, bandwidth to 12.5 kHz, Filter Audio to OFF and squelch to OFF. Be double sure that you have set Filter Audio to OFF or this will cause bad or no decodes.

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. An example audio snippet of a busy AIS signal is provided below. (Warning: Might be loud.)

The two AIS frequencies broadcast the same information and are used in commercial AIS receivers for redundancy and to help avoid transmission collisions. Some AIS decoder software written for the RTL-SDR such as AISdeco2 can monitor both channels at once, but most others cannot.

Once you have AIS reception set up you can then use a program for decoding. There are three decoders that we recommend. The most recommended decoder is the free AISdeco2. It is command line based, but can connect directly to the RTL-SDR and can monitor both AIS channels simultaneously. AISMon is another alternative free program, but is a little more clunky to use even though it has a GUI, as you must pipe audio from a program like SDR#. Both AISdeco2 and AISMon require the use of a third program such as OpenCPN to visualise the ship data on a map. ShipPlotter is another option that includes decoder and mapping, but is a commercial program with a 21-day trial.

**AISDeco2**

Currently the software we most recommend is AISdeco2 which is available for Windows, Linux and the Raspberry Pi. It can be downloaded from [xdeco.org](http://xdeco.org/). AISdeco2 directly connects to the RTL-SDR (requires no audio piping) and can listen to both AIS channels simultaneously. To use AISdeco2 on Windows follow these instructions.

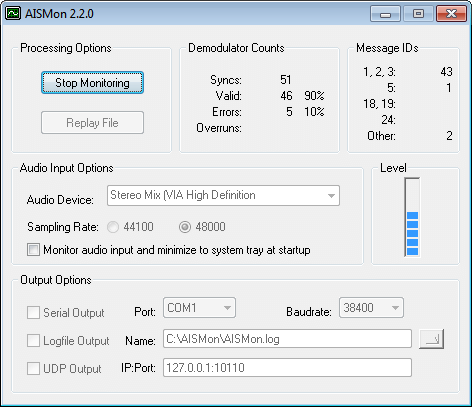
1. First ensure that you have accurately determined your dongle’s PPM offset using SDR# or another program (unless you are using a TCXO dongle).
2. Download the correct version of AISdeco2 for your OS from [xdeco.org](http://xdeco.org/) and extract the zip file into a folder on your PC.
3. In the extracted folder use Notepad to open aisdeco2.bat.
4. In this file set the value after the –freq-correction flag for your RTL-SDR dongles PPM offset value. Also set the –gain value to the optimal gain that you found when receiving AIS signals in SDR# or similar software. Save your changes and close Notepad.
5. Now double click the aisdeco2.bat file to start the command line software. Decoding will begin automatically.

Now to display the data on a map scroll down to the “Display AISDeco2 or AISMon data on a map” heading which will explain how to map the data on OpenCPN.

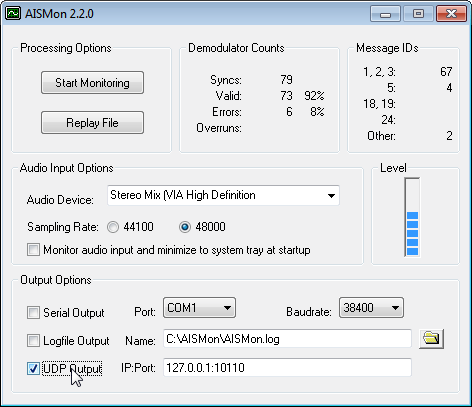
**AISMon Tutorial**

AISMon is a free AIS decoding program which can be used to decode one AIS channel. It is an alternative to using AISdeco2 which is the recommended software. AISMon can be downloaded from the [AISMon Yahoo group files section](http://groups.yahoo.com/group/aismon/" \o "AISMon Yahoo Group" \t "_blank). 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. AISMon does not connect directly to the RTL-SDR and so requires you to pipe the audio from your radio receiver software like SDR#.

1. Open AISMon, set the audio piping device to the one you have chosen, and set the sampling rate to 48000. Press start monitoring.
2. Tune to your AIS signal and adjust the volume in SDRSharp, and/or the Windows volume settings until the Level meter in AISMon reads at about halfway. If everything is working you should begin to see numbers appear in the Demodulator Counts section of AISMon.

[](http://rtlsdrblog.rtlsdrblog.netdna-cdn.com/wp-content/uploads/2013/04/AISMonSS.png)

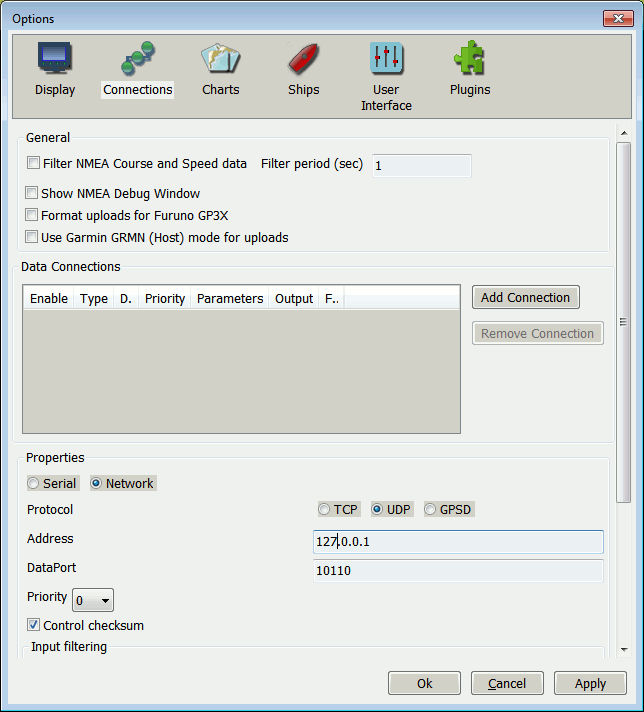
1. To output data to a mapping program (explained below), check UDP Output, and enter 127.0.0.1:10110 into the IP:Port box. (You will need to stop monitoring first if monitoring is still running).

[](http://rtlsdrblog.rtlsdrblog.netdna-cdn.com/wp-content/uploads/2013/04/AISMonUDPEnabled.png)

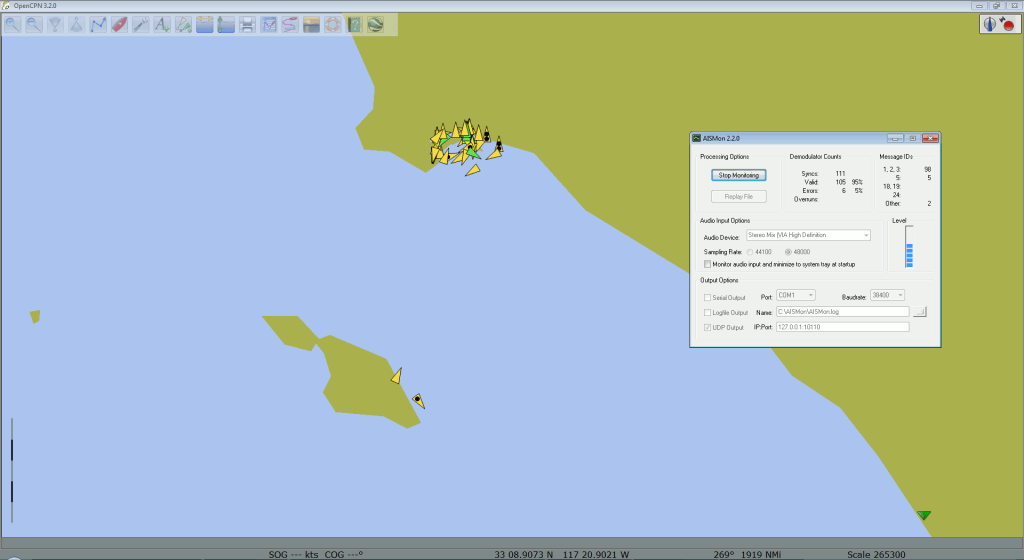
**Display AISDeco2 or AISMon data on a map**

By itself, AISDeco2 and AISMon do not display any ship information on a map. To view ship information you will need [another free opensource program called OpenCPN](http://opencpn.org/ocpn/), which is a chart plotting and navigation program. OpenCPN can read the “NMEA” information that is output by AISdeco2/AISMon, and plot the ship positions on a map.

1. Download and install OpenCPN [from here](http://opencpn.org/ocpn/download).
2. Click on the Options button (looks like a wrench) in OpenCPN, and go to the Connections tab.
3. Under Data Connections, click on Add Connection, and add a Network UDP connection, with address 127.0.0.1, and port 4159 is using AISDeco2 or 10110 if using AISMon. Click Apply and OK.

[](http://rtlsdrblog.rtlsdrblog.netdna-cdn.com/wp-content/uploads/2013/04/OpenCPNAddConnection1.png)

1. If everything has been set up correctly, ships will begin to appear in the OpenCPN map.

[](http://rtlsdrblog.rtlsdrblog.netdna-cdn.com/wp-content/uploads/2013/04/OpenCPNSS1.png)

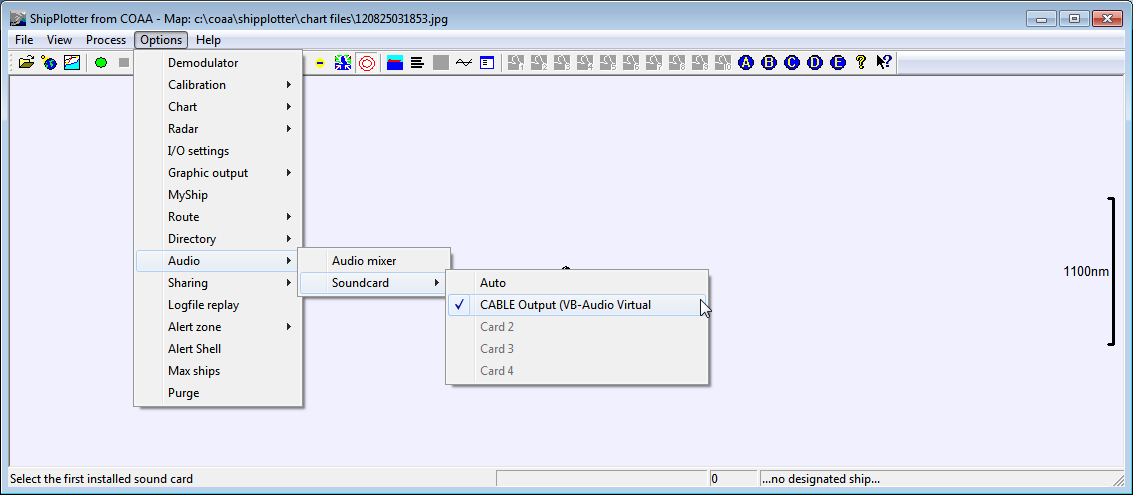
If you want more accurate maps or charts you will need to follow the [instructions on the OpenCPN website for downloading charts](http://opencpn.org/ocpn/installing_charts) for your particular location.

Another possible method for displaying AISMon data is to share your UDP data to the [marinetraffic.com](http://www.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. [Instructions are here](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.

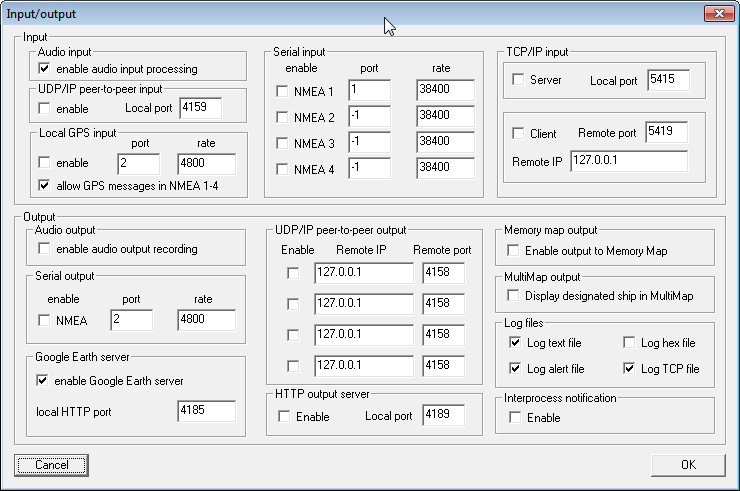
**ShipPlotter Tutorial**

[ShipPlotter](http://www.coaa.co.uk/shipplotter.htm) is another alternative to AISdeco2 and AISMon. It is a software tool that can decode and also 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 does not connect directly to the RTL-SDR and so requires you to pipe the audio from your radio receiver software like SDR#. Instructions on using ShipPlotter are shown below.

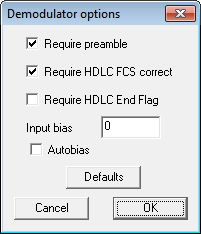
1. Download and install ShipPlotter from their [website here](http://www.coaa.co.uk/shipplotter.htm).
2. Open ShipPlotter. Go to Options -> Audio -> SoundCard and select your audio piping method.

[](http://rtlsdrblog.rtlsdrblog.netdna-cdn.com/wp-content/uploads/2013/04/ShipPlotterAudioSelection.png)

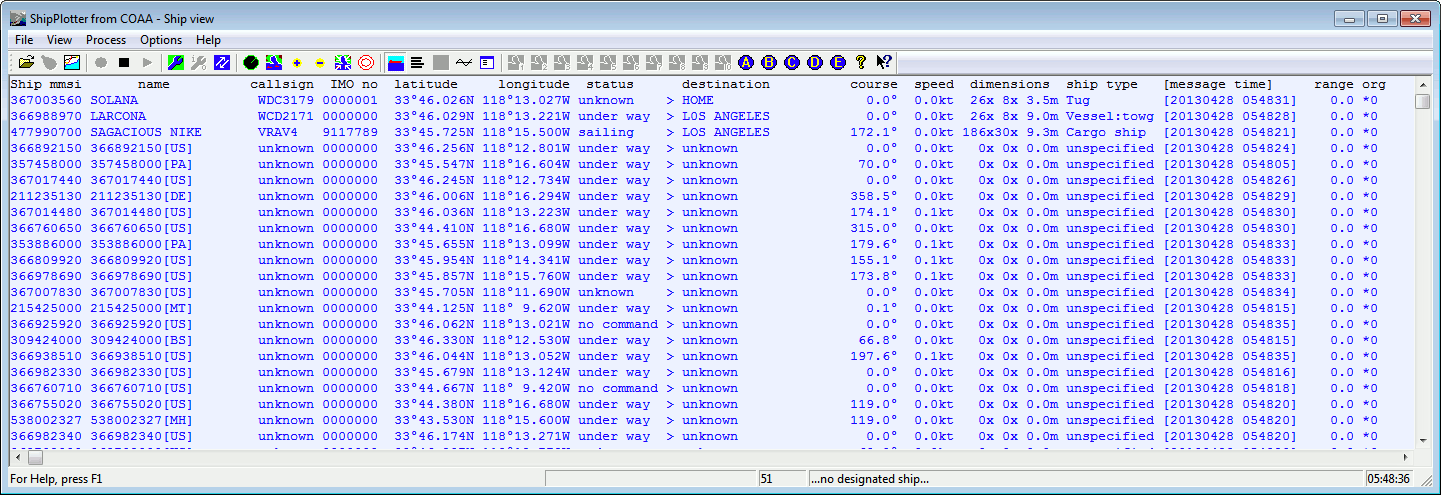
1. Under Options -> I/O Settings, ensure that enable audio input processing is checked.

[](http://rtlsdrblog.rtlsdrblog.netdna-cdn.com/wp-content/uploads/2013/04/ShipPlotterInputOutputOpts.png)

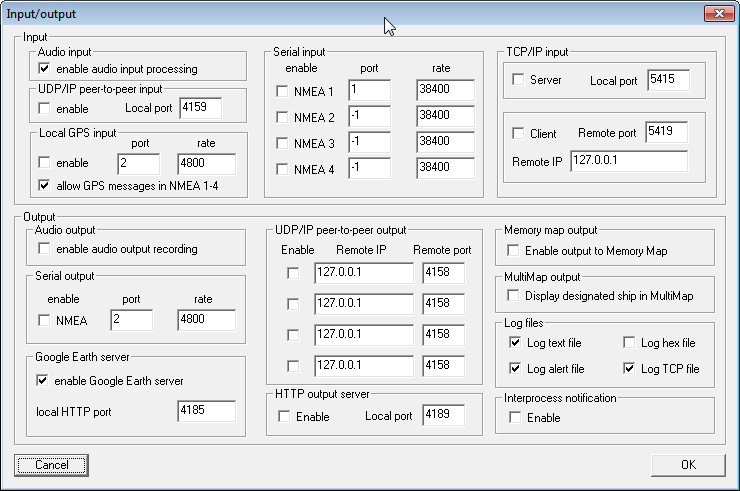
1. Ensure the Demodulator options in Options -> Demodulator are set to the default values, with ‘Require Preamble’ and ‘Require HDLC FCS correct’ both checked.

[](http://rtlsdrblog.rtlsdrblog.netdna-cdn.com/wp-content/uploads/2013/04/ShipPlotterDemodOpts.png)

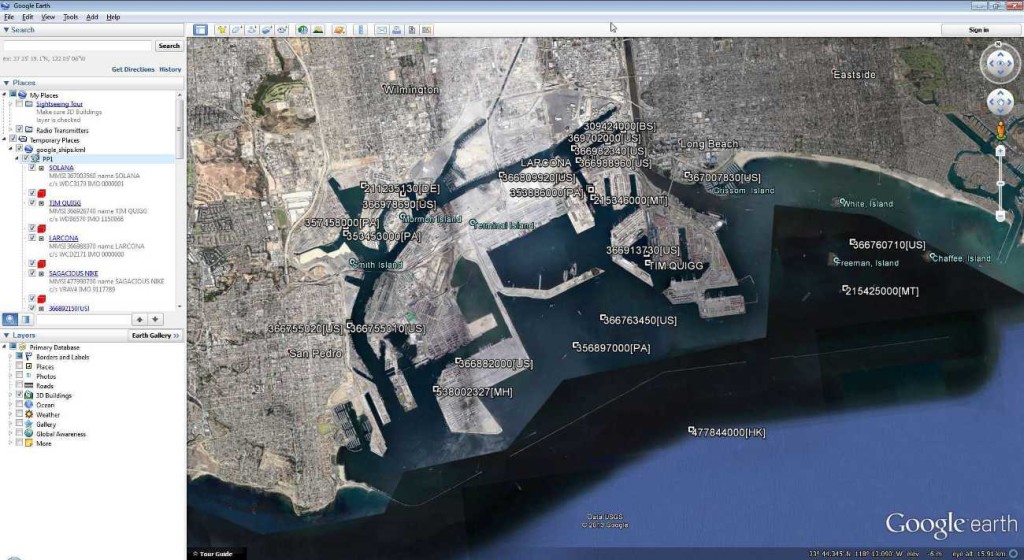
1. Tune to your AIS signal and push the green ‘Start’ button and then click on the ‘Raw’ icon which looks like a horizontal line with a squiggle (sine wave) going through it. This will show a waveform of the input audio. Ensure the audio levels are adequate and not clipping, by making sure the waveform peaks at about halfway on the graph by adjusting the volume settings in SDRSharp, or the windows volume settings.
2. Now you should be able to click on the ‘Ships’ and ‘Messages’ icons to see the decoded AIS information. To see ships visually, you will need to follow the ShipPlotter instructions for downloading charts.

[](http://rtlsdrblog.rtlsdrblog.netdna-cdn.com/wp-content/uploads/2013/04/ShipPlotterShipViewSS1.png)

An easier method than downloading charts is to display the ships in [Google Earth](http://www.google.com/earth/index.html). To get ShipPlotter to work with Google Earth, you must first enable the Google Earth server in Options->I/O Settings. The HTTP port can be left as default.

[](http://rtlsdrblog.rtlsdrblog.netdna-cdn.com/wp-content/uploads/2013/04/ShipPlotterInputOutputOpts.png)

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, to see the ships. Note that you will need to open ShipPlotter and begin decoding AIS signals by pressing the green button first BEFORE opening the google\_ships.kml file, otherwise ships will not show up.

[](http://rtlsdrblog.rtlsdrblog.netdna-cdn.com/wp-content/uploads/2013/04/GoogleEarthAISScreenShot1.jpg)

**Linux Software**

AISdeco2 is the currently recommended software to use in Linux and the tutorial is similar to the Windows one shown above.

**Some Tips**

* You may need to play with the bandwidth setting if using SDR# to pipe the AIS audio if your signals are weak in order to get a good decode. Just ensure the bandwidth covers the signal width adequately.
* A low noise amplifier (LNA) such as [this one](http://www.rtl-sdr.com/lna-for-all-wideband-low-noise-amplifier/) or [this one](http://rover.ebay.com/rover/1/711-53200-19255-0/1?icep_ff3=2&pub=5575102625&toolid=10001&campid=5337613909&customid=&icep_item=121502835316&ipn=psmain&icep_vectorid=229466&kwid=902099&mtid=824&kw=lg) may help you to pick up weak AIS signals better.
* Make sure you are using a good AIS tuned antenna for optimal reception.
* Antenna height is important, the higher and more unobstructed the better.
* You can test AIS decoding without an antenna using the example AIS .wav file in the AISMon Yahoo groups files section. Just use stereo mix as the audio piping method and play the audio in any audio player on your computer.

## High Gain Collinear AIS (162MHz) Receiving Aerial

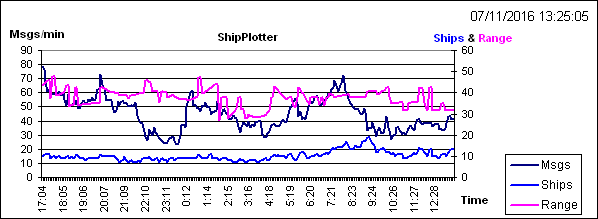
### by Neal Arundale - M1CHS

#### Background

Living by the sea and having an interest in both yachting and radio, I became interested in AIS as soon as the inexpensive Nasa AIS engine became available. I use the Nasa engine in conjunction with Seaclear software when sailing offshore on friends Yachts, and the SR162 with ShipPlotter at home. See [Comparative performance of Nasa, SR162 and Comar AIS-2-usb](http://arundale.com/docs/ais/sr162_nasa.html) Over the last year, I have been experimenting with different aerials to try and improve my reception distance at home. See [AIS Aerial Performance Comparisons](http://arundale.com/docs/ais/aerial_comparisons.html)  
I would be interested in any comments,  [click here to e-mail me](mailto:%6e%65%61%6c%5f%61%72%75%6e%64%61%6c%65%40%61%72%75%6e%64%61%6c%65%2e%63%6f%2e%75%6b" \t "_blank)

#### Location

My home is in Scarborough, North Yorkshire, UK about ½ mile from the sea and 250 feet above sea level facing the North Sea. I cannot actually see the sea from my home as the line of sight is obscured by houses, trees and the Castle headland. There are 400 feet cliffs 10 miles north and south. My aerial is currently located on top of the flat roof of my garage, giving easy access to the aerial. At some time in the future, I will probably locate the aerial on the chimney stack but for the present it is more useful to be able to gain access to the aerial easily and enables me to make a comparative judgment as to the effectiveness of the ideas I have been trying. I am fortunate in having daily ferries passing 30 miles offshore, Hull 30 miles across land as well as a number of rigs (normally having the same support vessels) 30+ miles offshore,and two receivers which help to judge the aerials' performance.

[](http://arundale.com/docs/ais/spxl.html)

#### [Software to produce this graph](http://arundale.com/docs/ais/spxl11.html)

#### http://arundale.com/docs/ais/spxl\_user\_guide.html

#### Increasing Reception Range

AIS signals are transmitted at VHF frequencies, which like terrestrial TV signals do not "bend" over the horizon. They are line of sight, typically 20 miles at reasonable aerial heights.  AIS signals are very sensitive to any distortion, a bit like digital terrestrial TV - the picture is good, very scrambled, or nothing - there's no in between. Increasing the sensitivity of the receiver by, for example, adding  a aerial pre-amplifier may make little or no difference to your ability to receive weak signals. This is because the receiver will amplify the signal, noise and distortion equally, so the information contained in the signal still cannot be decoded by the receiver or computer.The way round the problem is to improve the gain of the aerial and to reduce the noise introduced into the receiver by the downlead. The gain of the aerial is indicative of the amount by which the aerial will increase the signal relative to the background noise, and is why a good aerial is so important to the reception of weak signals. A shorter downlead or using better quality cable will introduce less noise into the receiver.  
I have tried to give some idea of the range of the various aerials I have tried and should be taken more as an indication of the relative performance of one aerial compared with another. The actual performance will depend on the aerials location and height, and, at over the horizon distance, [propagation conditions](http://www.mike-willis.com/Tutorial/PF6.htm) are important, but can on occasions result in reliable signals being received from over 100 miles away.  
Click [here](http://arundale.com/docs/ais/horizon.html) for a chart of horizon distance against aerial height.

#### Collinear Aerials

Aerials do not actually produce gain, they are more sensitive in a given direction. It's like putting a lens in front of the light in a light house, the light is concentrated in one direction, so at sea level the light is bright (where you want it) and 20° above the light will be much dimmer. Comparing with an aerial, the gain is the amount the light is brighter, looking at the light in the direction of the lens, than the light would be with no lens. A vertical collinear aerial is equally sensitive 360° around the aerial in the horizontal plane, but as the number of elements are increased, the increasing sensitivity in the horizontal plane is matched by a decrease in the sensitivity above and below the horizontal. By using 50ohm coax for the active elements, there is no impedance matching problem.

#### Mk1 plain wire

My first aerial was a plain wire attached to the end of the centre conductor of a length of RG58 coax, hung from the inside of my lounge window. This enabled me to receive AIS on the Nasa engine but only up to a range of around 5 miles. The length of the wire aerial proved to be very critical, the optimum appearing to be around 80 cms, this increased the range to around 10 miles.

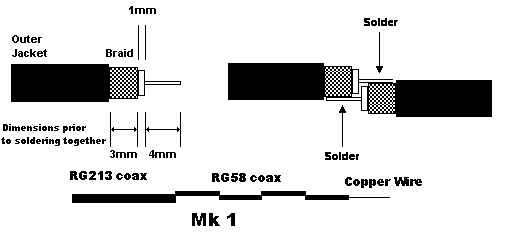
#### Silva Marine VHF aerial

The results from this were to say the least disappointing, although better than a plain wire, it was not much better, even though it was mounted on my garage roof.

#### Amateur 2m aerial

This was a [Diamond F-22 collinear](http://www.diamond-ant.co.jp/english/index.html) 144Mhz  (amateur 2m band). At 3.2 meters long was quite a large aerial. The aerial is tuned to 144Mhz whereas AIS is 162Mhz, thus it was some 10% off its centre frequency .It was a big improvement over the marine VHF aerial, more than doubling the minimum range to around 20-25 miles.

#### Mk1 -  5 element Collinear

After researching the Internet (see references/links below), my first home constructed collinear was a great success, with a range of 25-30 miles. It was very simple to construct being basically a 3 meter length of wire which I hung from the gutter of the roof above the flat roof of the garage.

The above dimensions are immediately prior to soldering and show how the ends solder together. Click the thumbnail to enlarge.

RG213 coax was used for the feed as I already had a wire in place. If you do not have any RG213, I would try using RG58, particularly if the cable run is quite small. Do not use RG213 for the aerial. Both coaxes are 50 ohm for matching purposes. 75 ohm TV or satellite coax theoretically should not work well, but I haven't actually tried it (if anyone wants to try you'll have to adjust the lengths by the velocity factor of the cable you use). 

Practice a joint or two first to ensure you get the length correct after soldering

|  |  |
| --- | --- |
| 1. Cut 4 lengths of RG58 coax 63.9 cms long 2. Strip  7 mm down to the center conductor 3. Strip a further 10 mm of outer cover 4. Gradually twist the screen back to to cover, leaving 3 mm of screen exposed 5. CuttingCarefully run solder round the 3mm of exposed braid to bind the braid 6. Cut back the dielectric to the centre conductor leaving 1 mm exposed 7. Tin the centre conductor with solder tinnedGutter 8. Match up the end with a similarly prepared end of the next section of coax 9. Cut back both centre conductors to just contact the screen of the next section the dimensions should now be as shown in the top line drawing. Most importantly the total length of the coax braid should be 61.1cm - this is the "active" length of  each section of the aerial. 10. Solder both centre conductors to both other screens 11. Connect all 4 sections of coax in the same manner 12. Attach the feed wire in the same manner, you may wish to add a connector on the feed wire. 13. Solder a 46.3cm length of plain uninsulated copper wire to the top end centre conductor 14. Check the continuity from top rod to centre of aerial plug also checking there is no short across the plug 15. Wrap PVC insulating tape round the joints 16. Insulate the end of the top wire from gutter, I used a "chock block" connector and hooked it onto the gutter using a wire coat hanger Gutter |  |

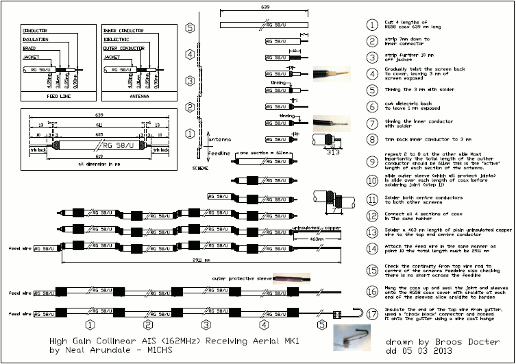
I constructed the aerial in an hour or so, total cost about £2. I also managed to hang it from the gutter using a long pole

##### **Modification tried**

Replacing the top ¼ wave top rod with ¼ wave shorted coax and ¼ wave top rod.  The reception distance was halved. Many marine VHF aerials would appear to be constructed in a similar way as they are designed to work with transmitters. They are often called DC grounded as a resistance check will indicated a short between the screen and the aerial.

**Technical Drawing**

Click the Image below for an [enlarged PDF version](http://arundale.com/docs/ais/aerial_mk1_technical.pdf), kindly drawn by Broos Docter (Netherlands)

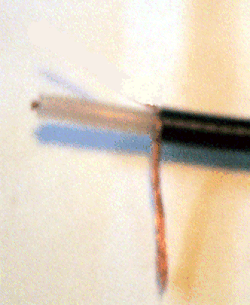
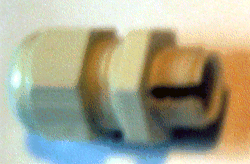
[](http://arundale.com/docs/ais/aerial_mk1_technical.pdf)

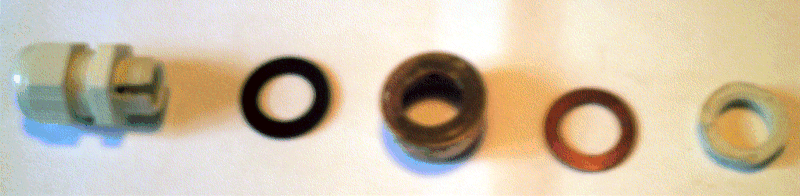
[See here for further pictures](http://arundale.com/docs/ais/kunkel.html) and reception details kindly supplied by Greg Kunkel, Long Island NY.

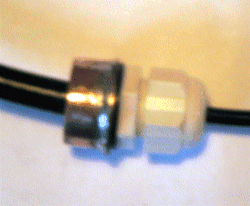
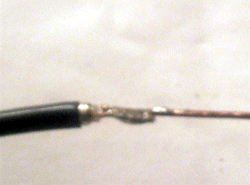
#### Mk2 - 9 element vertical collinear with ground plane

Self supporting 6 meters high, range 35-45 miles.  
Can be constructed without the ground plane with reduced range  
Unobtrusive to neighbours  
The aerial fits cleanly inside the [Sota](http://www.sotabeams.co.uk/sota-poles/) 7 meter fibreglass pole. This is tapered along its length and comes in seven telescopic sections (like a transistor radio aerial).

Construct 8 sections of RG58 coax the same manner as in the Mk 1 aerial above up to step 7

1. Slip a 2" (5 cm) sleeve over the coax  RG58 cover to strengthen the eventual joint. I used a length of sleeving removed from standard TV aerial coax. This was a fairly tight fit, but slipped on nicely when lubricated with a touch of washing up liquid
2. Continue with steps 8 to 10
3. When all 8 sections are soldered together check the continuity
4. Slip the sleeves over the joints
5. Again check the continuity
6. Hang the coax up and seal the sleeves onto the RG58 coax cover with araldite at each end of the sleeve
7. Leave the araldite overnight to dry
8. When dry, again check the continuity. Mine was OK so I didn't have to locate and fix a bad joint. I suspect if  your soldering is not up to scratch you'll probably have to start again.
9. Cut a length of about 1½ metres of RG213 coax
10. Cut about 25mm of sleeve off one end the RG213 coax
11. Unbraid the braid on the coax back to the sleeve. The coax tail will be required if you fit the ground plane  
    
12. Obtain a 16M plastic cable gland from an electrical wholesaler. It must be able to clamp the RG58 cable **and** allow the RG213 to enter through the threaded side **and** you must be able to tighten the nut from within the cup of the cap. I had to file the nut down a fraction.   
    
13. File a slot in the threaded (bolt) part of the cable gland just sufficient to slot the unbraided coax into when the RG213 coax is pushed through the bolt end of the gland towards the nut end. I used a Dremel.   
    
14. Put all the cable gland except the nut onto the RG58 coax before joining the RG58 & RG213 coaxes
15. Obtain a copper or brass washer that will just fit over the cable gland bolt  
    
16. Make a cup to hold the ground plane
    1. Cut a 46.3 cm (¼ wave) length of 28 mm copper tube for the ground plane
    2. Obtain a plain (not "Yorkshire") 28mm copper joint sleeve, file off the centre groove so that the sleeve will slide right over the tube, and is a quite tight fit
    3. Cut a further 1 cm ring of the same tube
    4. Saw through the ring to split it open
    5. By filing the saw cut open and squeezing it together, create an insert that just fits into to copper tube
    6.  Push the insert half way into the tube to hold it together and solder a 28mm copper or brass washer onto the top of the insert. The washer will eventually hold the 16m cable gland, so check you can fit the gland before you solder it. You may need to enlarge the centre of the washer after soldering.  
       When completed the cap should be a tight fit in the copper tube
17. This is how they assemble together



1. Push the RG213 right into the cable gland (leaving the braid tail through the slot), while allowing the nut and clamping washer to tighten firmly on the RG58 coax. When satisfied, clamp the cable gland firmly onto the RG58 coax by tightening the nut
2. Push the copper/brass washer tightly down over the gland bolt so that the plastic nut will tighten onto the washer and braid
3. Put the plastic nut on and tighten up. The whole joint should finish being mechanically & electrically sound, as well as being waterproof from the top
4. Solder the top rod on to the end of the RG85
5. Put a 10mm offcut of RG58 sleeve over the top joint & fill with araldite to strengthen the joint onto the cable
6. Assemble the top and second section of the fibreglass pole. Mark  the overlap on the pole so that when the pole is assembled you can accurately position the aerial alongside and outside the pole in order to determine where the bottom joint of the aerial will be in the pole
7. Assemble the rest of the [Sota](http://www.sotabeams.co.uk/sota-poles/) pole, position the aerial alongside and mark the outside of the pole exactly where the copper/brass cup washer will be located when inserted into the pole
8. Position the sleeve on the outside of the tube (see section 17.2) to act as a stop preventing the copper tube going further up the pole & kinking the aerial coax
9. Put the copper tube over the RG213 and push into the copper/brass cup
10. Assemble the bottom two sections of the pole and mark on the outside of the bottom section where the next section ends
11. Assemble the whole of the pole and lay the aerial alongside with the top of the aerial exactly where it will be when inserted in the pole
12. Move the sleeve on the outside of the copper tube to the position where it will act as a stop when inserted into the pole. This prevents the aerial being kinked inside the pole.
13. Cut the RG213 so that when a connector is added the whole of the connection will be within the bottom section of the pole
14. Put on the N type jack  
    
15. Obtain a length of 25mm plastic electrical conduit, this will just slide into the copper tube. This must be prevented from going inside the copper tube as it is used to support the weight of the copper tube. I obtained a conduit joining sleeve which just slipped a couple of mm into the copper tube, and would slip over the conduit. Note the RG213 socket must go pass through the centre of this connector. Alternatively you could glue a washer or sleeve onto the plastic conduit, as long as the socket will pass through and it will fit inside the pole.
16. Before you assemble the aerial into the fibre glass pole, obtain 30 cm of  38 mm aluminum tube (TV aerial mounting pole). This should slide about ½ way down the outside of the bottom section of the pole. With a hack saw, make a longitudinal cut through one side only of the tube, from top to bottom.
17. Slide the tube over the bottom section of the pole to nearly the bottom, prising the tube slightly open as required.  This is where the pole will be clamped in the aerial mounting bracket. Note that as you tighten the mounting bracket clamps this tube will fit the taper of the fibreglass pole exactly
18. Push the rubber centre bung out from the screw cap on the end of the bottom section of the pole and replace it with a plastic washer that just fits in the cap but will allow a N type plus through the centre. The hole must not allow a 25 mm plastic electric conduit pipe through. I manufactured the washer from an old plastic bottle. It must support the weight of the copper tube and RG213 cable.
19. Cut the conduit at the correct length to just support the copper tube to the mark on the pole, whilst the conduit itself is supported be the end cap of the pole when it is screwed on the bottom of the pole.   
    
20. Carefully assemble the whole aerial into the pole, starting with the top section, slide each section down from the top. I supported the cable at each pole joint with 15mm of foam cut from central heating pipe lagging.
21. When assembled clamp the pole into a TV aerial mounting bracket round the aluminum tube.

|  |  |
| --- | --- |
| **Comments** The mechanics of the construction is how I made the aerial, there are probably better and/or simpler ways of building the aerial. Some of the references below may give you other ideas. Note however the electrical dimensions must be adhered to. PVC pipe (in place of fibreglass) may alter the electrical dimensions, as it could affect the velocity factor of the aerial. Glass fibre is OK, carbon fibre is not. *18/11/08 a test with an antenna analyser indicated, when inserted in either a PVC or fibreglass tube, the resonant frequency was reduced by 2%. This is barely significant*.  I was also conscious of rain dripping into the joints as it would run down the pole, and if inside the pole, down the coax cable as well. This is why the bottom of the pole was not sealed, as it would help any moisture or condensation within the pole to dissipate. The aerial has been subject to a 60 mph gale with no problems. The ground plane appears to increase the range by about 20%, and also appears to reduce ships being "lost" under the high (400 feet) cliffs around Flamborough Head. |  |

[See here for further pictures](http://arundale.com/docs/ais/jeanpierre.html) and reception details kindly supplied by Jean Pierre, Perpignan, France.

Futher modification (January 08)

I wasn't totally happy with the way the aerial wire hung down the [Sota](http://www.sotabeams.co.uk/sota-poles/) pole as it tended to kink and really needed suspending from the top while still supporting the weight of the copper tube from the bottom.

 I replaced the top 1/2 inch of the top copper rod with a small (probably 8ba) screwed rod (again 1/2 inch), made from a brass bolt with the head chopped off, by soldering the screwed section to the top of the copper rod. It must be small enough to protrude through the fibreglass pole (see below).

 I cut down the top section of the Sota pole so that the top 1/4 inch of the screwed rod just projected through the fibreglass at the top of the pole when the whole aerial was re-assembled.

 I found a small plastic cup that just (and only just) slipped over the top of the top section of the fibreglass. Something like the top of a biro would do only a bit smaller - I actually used a hypodermic needle cover !!

I carefully araldited a couple of brass nuts into the plastic cup with a similar bolt in place (until the araldite was set) in such a way that I could use the cup to hold the top rod + the weight of the coax at the top of the pole.

 In this manner I could dissemble the aerial from the pole at a later date if required, adjust the length to just prevent the coax kinking, and maintain a waterproof cover over the top of the pole.

#### Equipment & Software

I use a [Smart Radio](http://www.diytrade.com/china/4/manufacturers/96914/products-list.html) SR162  dual channel receiver at home, purchased direct from Smart at $460 including air freight. Delivery took 5 days. I also have a [Nasa](http://www.nasamarine.com/products.php?cat=18) AIS engine. This enables me to compare the relative performance of both the aerials and the receivers. The sensitivity and response of both receivers can be judged from the comparison [here](http://arundale.com/docs/ais/sr162_nasa.html). Notice the Smart monitors both AIS channels simultaneously, whereas the Nasa switches between the channels.  
At home I use the ShipPlotter software because it makes it simple to download data to other servers. When sailing I use the Nasa engine with the SeaClear software because SeaClear has much better chart management and displays NMEA info alongside the chart.  
The ShipPlotter software will accept decoded NMEA input as well as raw audio. Both receivers output decoded NMEA . I put a [discriminator tap](http://www.discriminator.nl/index-en.html) on the IC inside the Nasa engine to see if the ShipPlotter program would make a better job of decoding the audio signal than the Nasa engine. It made no difference.  
  
**Comparison data on graph (September 08)**I'm using a coaxial relay to switch the aerials under control of the same PC running Shipplotter and plotting the graphs. It is synchronised to plot message rate (1) or (2) on the graph as the aerial is switched. Every alternate message rate is therefore recorded for a different aerial. The two aerials are located at the same height about 10 feet apart, using the same type and length of downlead and feeding the same AIS receiver.

#### Calculations

v=fλ   
where v=velocity, f=frequency and λ=wavelength  
In our case we want wavelength so λ= v/f  
V is the velocity of propagation which is the speed of light in free space  
AIS frequencies are 161.975 MHz and 162.025 MHz  
One wavelength λ in free space is 300/162 = 1.852 metres = 185.2 cm  
[Velocity Factor](http://home.swipnet.se/27mhz/rg.html)  for RG58U coax is 0.66  
One wavelength λ in RG58U coax is 185.2 x 0.66 =  122.2 cm

|  |  |  |  |
| --- | --- | --- | --- |
| Element | Length | Velocity factor | Size |
| Top Rod | ¼ wave | 1 | 46.3 cm |
| Middle Coax | ½ wave | 0.66 | 61.1 cm |

#### Cutting Length of each coax section

|  |  |
| --- | --- |
| *Braid* | 61.1 cm |
| *Dielectric* | 0.1 x 2 = 0.2 cm |
| Center Conductor | 1.3 x 2 =  2.6 cm |
| Total | 63.9 cm |

#### Approximate Total Length of Aerial

Top rod + number of coax sections x (length of braid of each section + 1mm joining allowance)

|  |  |  |
| --- | --- | --- |
| Elements | Length | Gain |
| 3 | 168.7 cm | 3db |
| 5 | 291.1 cm | 6db |
| 9 | 536.0 cm | 9db |

**References**

|  |  |
| --- | --- |
| [How to build a high gain Aerial for the UHF or CB bands](http://arundale.com/docs/ais/radioexperimenter.html) | Anon |
| [Build A 9 dB, 70cm, Collinear Antenna From Coax](http://www.rason.org/Projects/collant/collant.htm) | Mike Martell N1HFX |
| [Build a 2 metre, 5/4 Wave Antenna](http://www.rason.org/Projects/ant54/ant54.htm) | Mike Martell N1HFX |
| [Omni-Gain VerticalCollinear for VHF and UHF](http://www.repeater-builder.com/antenna/wa6svt.html) | Mike Collis WA6SVT |
| [Construction Notes on a WA6SVT Coaxial Collinear Antenna](http://www.repeater-builder.com/antenna/n1bug-construction.html) | Paul Kelly N1BUG |
| [Some variations and construction ideas on the WA6SVT Omnidirectional Coaxial Collinear](http://www.repeater-builder.com/antenna/uhfcoll.html) | Kevin Custer W3KKC |
| [Slim Jim](http://www.m0ukd.com/Calculators/Slim_Jim/index.php) | M0UKD |
| [Horizontal Bi-Directional Wires](http://www.cebik.com/) available from [Antennex](http://www.antennex.com/premium.html) (by subscription) | L.B.Cebik W4RNL |
| [The Case of the Curly Collinear](http://www.cebik.com/) available from [Antennex](http://www.antennex.com/premium.html) (by subscription) | L.B.Cebik W4RNL |
| [2 m. Band VHF Collinear with 2 Dipoles](http://www.qsl.net/sv1bsx/gammadip/collinear2.html) | sv1bsx |
| [The W7LPN 2 - 440 Vertical Collinear Antenna Project](http://www.hamuniverse.com/2m440collinearvertical.html) | W7LPN ~ Rick Frazier |
| [A 2.4Ghz Vertical Collinear](http://www.nodomainname.co.uk/Omnicolinear/2-4collinear.htm) | By Brian Oblivion and Capt.Kaboom modified by Richard A Wenner |
| [Coaxial Collinear hints](http://members.aol.com/K7ITM/) | K7ITM |
| [UHF/VHF Range Calculations](http://arundale.com/docs/ais/AppNote_UHF_VHF_Calc.pdf) | Pacific Crest |
| [Aerial Gain Explained](http://www.marcspages.co.uk/tech/antgain.htm) | Marc G Dekenah |
| [VHF and UHF Path Loss Calculations for Amateurs](http://arundale.com/docs/ais/PathlossCalculationsforAmateurs.pdf) | Leith Martin VK2EA |
| [ARRL Antenna Handbook Radio Wave Propagation](http://www.ebooksx.com/download2.php?id=The%20ARRL%20Handbook%20for%20Radio%20Communications%202007)- Link was broken - this is the complete book | Anon |
| [Everyday VHF, UHF, and SHF propagation](http://www.qsl.net/oz1rh/troposcatter99/troposcatter99.htm) | Palle Preben-Hansen, OZ1RH |
| [Beyond the Horizon Propagation](http://www.mike-willis.com/Tutorial/PF6.htm) | Mike Willis [G0MJW](http://www.mike-willis.com/g0mjw/g0mjw.html) |
| [Tropospheric Ducting Forecast](http://www.dxinfocentre.com/tropo_nwe.html#hour6) | William R Hepburn |
| [Radio Horizon](http://arundale.com/docs/ais/horizon.html) | Neal Arundale |
| [Sota Fibreglass Poles](http://www.sotabeams.co.uk/sota-poles/) |  |
| [Smart Radio $460](http://www.sr162.com/) |  |
| [Nasa AIS Engine](http://www.nasamarine.com/proddetail.php?prod=ais_engine) |  |
| [Diamond f-22 144MHz BASE STATION ANTENNA](http://www.rfparts.com/diamond/f22a.html) |  |
| [Cushcraft Antennas](http://www.cushcraftamateur.com/Product.php?productid=AR-2) |  |
| [Discriminator Taps](http://www.discriminator.nl/index-en.html) |  |
| [RG Series Cable Characteristics](http://home.swipnet.se/27mhz/rg.html) |  |
| [Ship Plotter](http://www.coaa.co.uk/shipplotter.htm) |  |
| [Sea Clear](http://www.sping.com/seaclear/) |  |
| [AIS currently being received in Scarborough](http://arundale.com/docs/ais/sp_map.html) | Neal Arundale |
| [Graphical Monitoring of ShipPlotter](http://arundale.com/docs/ais/spxl.html) | Neal Arundale |
| [Performance Comparision Nasa - SR162](http://arundale.com/docs/ais/sr162_nasa.html) | Neal Arundale |
| [AIS Aerial Performance Comparisons](http://arundale.com/docs/ais/aerial_comparisons.html) | Neal Arundale |

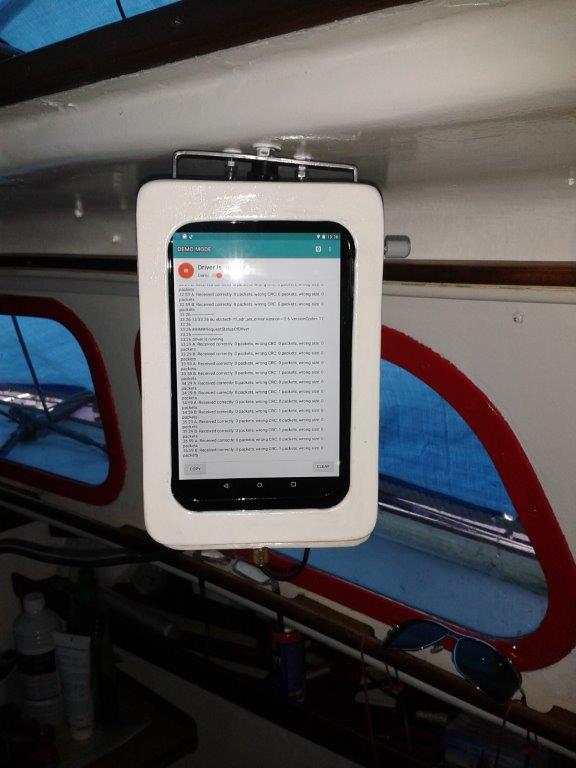
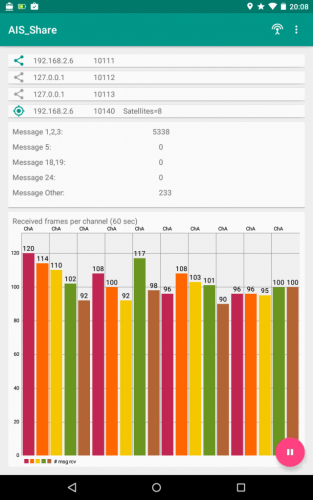
**USING AIS SHARE, OPENCPN AND AN RTL-SDR ON A SAILBOAT**

AIS Share is an [app for Android that allows you to turn an Android device into an AIS receiver](http://www.ebctech.eu/blog/ais-share-share-ais-ship-messages-via-android/) by using an RTL-SDR. AIS stands for Automatic Identification System and is used by ships to broadcast their GPS locations, to help avoid collisions and aid with rescues. An RTL-SDR with the right software can be used to receive and decode these signals, and plot ship positions on a map.

AIS Share is a dual channel decoder that outputs decoded NMEA messages via UDP, so that plotting software like OpenCPN can be used to display the ships on a map. AIS Share had been around before in another form known as [rtl\_ais\_android which we posted before](http://www.rtl-sdr.com/a-new-ais-decoder-for-the-rtl-sdr-on-android/" \t "_blank), but this version of AIS Share is a newly updated and improved version that now includes a very nice GUI. The app costs about $2 and [is available on the Google Play store](https://play.google.com/store/apps/details?id=eu.ebctech.ais_share), but there is [a demo](http://www.ebctech.eu/blog/rtl-sdr-ais-driver/) available that will work up until 1000 messages are received. You will need an RTL-SDR and a USB OTG cable to run the app.

Over on YouTube user [Tobias Härling](https://www.youtube.com/channel/UC2CDwfTelYPpfG1Aux5WTrQ) has uploaded a video showing how he used a Raspberry Pi and RTL-SDR dongle to set up an AIS receiver. AIS stands for Automatic Identification System and is a radio system similar to ADS-B which allows you to create a radar-like system for boats. For Windows we have a tutorial on AIS [reception here](http://www.rtl-sdr.com/rtl-sdr-tutorial-cheap-ais-ship-tracking/).

In his setup he uses [rtl\_ais](https://github.com/dgiardini/rtl-ais" \t "_blank) and the [kplex](http://www.stripydog.com/kplex/" \t "_blank) software and shows how to install everything from scratch. He also shows how to set the system up so that decoding automatically starts up and begins outputing NMEA data through the network when the Raspberry Pi is powered on. This way an a device like an iPad could be used to run OpenCPN to view the plotted ships.

[](http://rtlsdrblog.rtlsdrblog.netdna-cdn.com/wp-content/uploads/2016/02/Harmen_sailboat_aisreceiver_enclosure_backside_1.jpg)[](http://rtlsdrblog.rtlsdrblog.netdna-cdn.com/wp-content/uploads/2016/02/Harmen_sailboat_aisreceiver_enclosure_1.jpg)[](http://rtlsdrblog.rtlsdrblog.netdna-cdn.com/wp-content/uploads/2016/02/ais_share_main_screen_gps-640x1024.png)

Recently the author of the app received word from a user called Harmen [who has successfully been using his AIS Share app on his sailboat](http://www.ebctech.eu/blog/using-ais-share-and-opencpn-on-sail-boat/). Harmen uses the app on an Android tablet which is enclosed in a waterproof box. For an antenna he uses a coax collinear.

In the future the author writes that he’d like to update the app to support things like the ability to change more dongle settings like bandwidth/sample rate and add the possibility of using the internal phone/tablet GPS. He is also open to any community suggestions.AIS Share Receiver on the sailboat in a waterproof case.

The back of the Android Tablet, showing the RTL-SDR and the antenna connection.

The AIS Share main screen GUI.

# SETTING UP A RASPBERRY PI BASED AIS RECEIVER WITH AN RTL-SDR

# <https://youtu.be/3VBz4HE0bZA>

# <http://www.rtl-sdr.com/setting-up-a-raspberry-pi-based-ais-receiver-with-an-rtl-sdr/>

<https://github.com/dgiardini/rtl-ais>

<http://www.stripydog.com/kplex/>

Informative <http://opencpn.org/ocpn/Plugin_external_rtlsdr-usb>

<http://www.rtlsdr.com/2013/07/receiving-decoding-and-plotting-ais-using-a-rtl-sdr/>

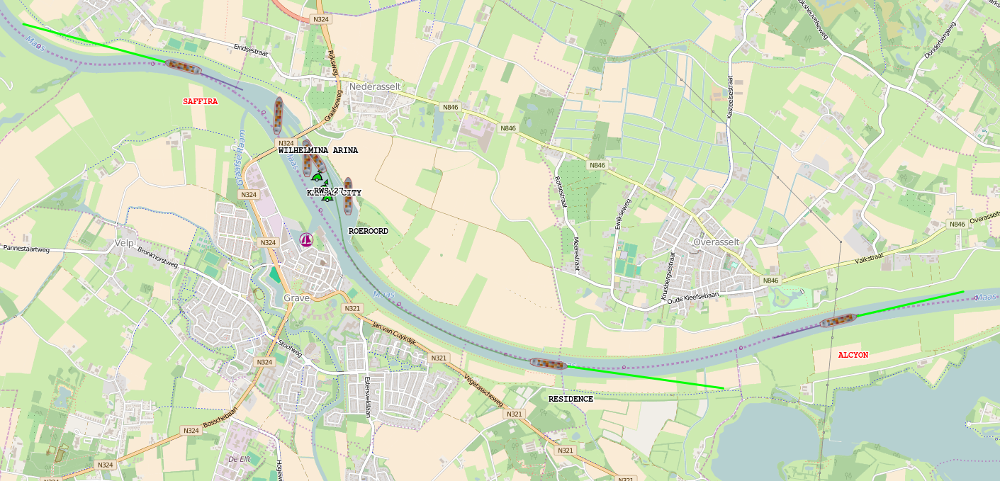
<https://publiclab.org/notes/ajawitz/06-11-2015/raspberry-pi-as-marine-traffic-radar>

<https://mvcesc.wordpress.com/2014/09/03/ais-on-the-cpn-pi-lot-project/>

<https://mvcesc.wordpress.com/2015/04/28/opencpn-on-the-raspberry-pi-2-with-ais/>

<http://lagunabeachcomputer.com/cheap-homemade-diy-ais-162mhz-sdr-antenna/>

<http://www.sailoog.com/en/blog-categories/openplotter-rpi>

**[](http://blog.videgro.net/wp-content/uploads/2015/12/screenshot_openstreetmap_20151229_1705.png)RASPBERRY PI: YOUR CHEAP AUTOMATIC IDENTIFICATION SYSTEM (AIS) RECEIVER**

<http://blog.videgro.net/2015/12/raspberry-pi-automatic-identification-system-ais-receiver/>

AIS tracking at December 29th 2015 – 17:05h – Saffira, Wilhelmina Arina, RWS 27,Kansas City, Roeroord, Residence, and Alcyon. **(Click image to enlarge)**

In this tutorial you will learn how to teach your Raspberry Pi to become a brave **Automatic Identification System (AIS)**receiver.

*What is AIS? The Automatic Identification System (AIS) is an automatic tracking system used on/for ships. It transmits information like identification, position, course and speed.*

**Ingredients**

* Raspberry Pi (2 model B)
* DVB-T-DAB-FM receiver
* Client computer running OpenCPN

1. **Buy DVB-T-DAB-FM receiver**

Buy a cheap RTL2832U / RTL2838UHIDIR based DVB-T-DAB-FM receiver USB stick at i.e. eBay, DealExtreme etc. I bought mine for < 10 euro. Take a delivery time of several weeks into account.

1. **Install Rasbian**

Meanwhile make sure the the latest Rasbian is running at your Raspberry Pi [(read this: How to install Rasbian at Raspberry Pi)](http://blog.videgro.net/2015/11/raspberry-pi-first-use/)

1. **Install extra packages**

Login to the console of your Raspberry Pi using SSH. Install software defined radio (SDR) package based on the Realtek **RTL2832U** chipset, version control system (GIT) and some extra development libraries needed as dependencies to compile RTL-AIS.



|  |  |
| --- | --- |
| 1  2 | sudo apt-get update && sudo apt-get upgrade  sudo apt-get install rtl-sdr git librtlsdr-dev libusb-1.0-0-dev |

1. **Creating working environment**

Make a working environment in your home directory at the Raspberry Pi and step into this new directory. In this directory we will download the source files and compile RTL-AIS.



|  |  |
| --- | --- |
| 1  2 | mkdir ~/ais  cd ~/ais |

1. **Build RTL-AIS**
   1. First clone the RTL-AIS GIT repository.



|  |  |
| --- | --- |
| 1 | git clone https://github.com/dgiardini/rtl-ais |

* 1. *Optional*: Check the version you cloned:



|  |  |
| --- | --- |
| 1 | git log -n 1 |

The HEAD of my cloned GIT repository is at commit: 5a03d3505639d2594f488ddc5391be1235c0130e, Mon Jul 27 10:37:08 2015 -0300.

* 1. Step into the cloned directory:



|  |  |
| --- | --- |
| 1 | cd rtl-ais |

* 1. Compile RTL-AIS



|  |  |
| --- | --- |
| 1 | make |

After a while, a new file exists in this directory: ‘**rtl\_ais**‘. This is the executable which listens for AIS messages at a certain frequency, translates them into NMEA messages and send them to a defined UDP port at a specific host.

**NMEA** (short for **National Marine Electronics Association**) is a specification for marine communication and includes for example GPS data and data of a lot of other instruments.

1. **Prepare client computer**

To be able to visualize the AIS/NMEA data, you have to install a decoder/client. I used at my Ubuntu (15.10) workstation OpenCPN as client. Besides of that ‘rtl\_ais’ tries to send the data to UDP port 10110 at the computer which is running the client software, so you have to open this port in the firewall at the workstation for incoming UDP traffic from the Raspberry Pi.

* 1. Install OpenCPN ([Install OpenCPN (4.1.1108) at Ubuntu (15.10)](http://opencpn.org/ocpn/adding_ppa))
  2. Open firewall – Make sure the UDP port 10110 at your workstation is open for incoming traffic. This depends highly on your firewall configuration! For example:



|  |  |
| --- | --- |
| 1 | iptables -A INPUT -p udp --dport 10110 -j ACCEPT |

1. **Connect hardware**

[](http://blog.videgro.net/wp-content/uploads/2015/12/raspberry-pi2-model-b_sdr-rtl_433MHz.jpg)

Now it’s time to insert the DVB-T-DAB-FM receiver USB stick into the Raspberry Pi. After inserting, check the output of the command ‘dmesg’. At the end there should be something like:



|  |  |
| --- | --- |
| 1  2  3  4  5  6 | [ 2609.827861] usb 1-1.5: new high-speed USB device number 6 using dwc\_otg  [ 2609.939903] usb 1-1.5: New USB device found, idVendor=0bda, idProduct=2838  [ 2609.939931] usb 1-1.5: New USB device strings: Mfr=1, Product=2, SerialNumber=3  [ 2609.939948] usb 1-1.5: Product: RTL2838UHIDIR  [ 2609.939965] usb 1-1.5: Manufacturer: Realtek  [ 2609.939981] usb 1-1.5: SerialNumber: 00000001 |

…indicating that the DVB-T-DAB-FM receiver has been recognised as RTL2838UHIDIR. Variations are possible.

1. **Start receiving AIS messages**

Now you can start RTL-AIS at the Raspberry Pi. For this we do need to know the IP address of the workstation which is running the AIS/NMEA client (OpenCPN). Besides of that I like it to see the decoded messages in my console, so I used parameter ‘-n’.



|  |  |
| --- | --- |
| 1 | sudo ./rtl\_ais -p 0 -R on -n -h 192.168.178.2 |

When you don’t see any messages after a certain time (let’s say 5 minutes), check the chapter ‘**Troubleshooting**‘. Stop RTL-AIS by pressing CTRL+C. Advance only to the next step when you are receiving messages indeed!

1. **Start AIS/NMEA client**

In this tutorial we decided to use **OpenCPN** as client and AIS data is send to port **10110** of the workstation.

* 1. Start ‘opencpn’
  2. Click the Options-icon
  3. Connections-tab
  4. Add Connection
     + Network
     + Protocol: UDP
     + Address: IP address of workstation (i.e. 192.168.178.2)
     + DataPort: 10110
     + Optional – Check “*Show NMEA Debug Window*” to see incoming messages.

*Now wait for data and enjoy following vessels in your neighbourhood.*

**Results**

**Example 1: Activity in canal lock Grave – The Netherlands**

Check this time-lapse as example (best viewed in full-screen). Starring: Noorderlicht (MMSI: 244650819), Roeroord (MMSI: 244750033), Serena (MMSI: 244630092), Danique-F (MMSI: 205228790), Heavenly (MMSI: 244700610), Staay (MMSI: 244780486), Heumen (MMSI: 244780915), and Hulder (MMSI: 244700789). The **MMSI** is the **Maritime Mobile Service Identity**which is a series of nine digits to uniquely identify ships. **(Best viewed full screen)**

|  |  |
| --- | --- |
| [AIS tracking at December 24th 2015 - 10:02h - Roeroord, Zomp, KVB Maranta, Kansas City, Serena, Markstroom, and Wilhelmina Arina.](http://blog.videgro.net/wp-content/uploads/2015/12/screenshot_opencpn_20151224_1002.png)AIS tracking at December 24th 2015 – 10:02h – Roeroord, Zomp, KVB Maranta, Kansas City, Serena, Markstroom, and Wilhelmina Arina. **(Click image to enlarge)** | [Serena - cruise ship - MMSI 244630092](http://www.marinetraffic.com/en/ais/details/ships/244630092)Serena – cruise ship – MMSI 244630092 |

**Example 2: [Roeroord‬](http://www.mvogroep.nl/nl/materieel/wat-hebben-we/drijvend-materieel/zandwielenponton/" \t "_blank) on the move (Martens en Van Oord)**

Starring: Roeroord (MMSI: 244750033), Iris (MMSI: 244700937), Serena (MMSI: 244630092), RWS 27 (MMSI: 245380000), KVB Maranta (MMSI: 244650982), and Kansas City (MMSI: 244750436). **(Best viewed full screen)**

**Troubleshooting**

* + **General**

*Preferable*: Put the antenna as high as possible, with a free line-of-sight (outside).

* + **Find the AIS frequency**

Here comes a tricky part, find the AIS frequency for your hardware. It is possible that some deviations arose during manufacturing of the receiver. In that case you have to tune to another frequency as the default AIS frequencies which are build into the rtl\_ais-executable. The AIS signal is transmitted to two channels (**161.975MHz** and **162.025MHz**) it is designed like this to avoid interference, and to allow channels to be shifted without communications loss from other ships. The data will be duplicated to both frequencies, so we only need to find the AIS signal at least at one of these two frequencies. Just find the frequency with the highest **signal/noise** ratio. For this I used the software defined radio receiver ‘**gqrx**‘ at my Ubuntu workstation. Make sure packages ‘rtl-sdr’ and ‘gqrx-sdr’ have been installed and of course the DVB-T-DAB-FM receiver must be inserted into this computer. Tune to a frequency near 162MHz, **narrow FM** and a filter of +/- 12.5kHz. Fine tune to the frequency with small beeps, make a note of this frequency. For me 162.017MHz worked.

Now use this command to start RTL-AIS use the frequency which worked in your case:



|  |  |
| --- | --- |
| 1 | sudo ./rtl\_ais -r 162.017M -p 0 -R on -n -h 192.168.178.2 |

**Bonus**

* **Start RTL-AIS automatically after reboot**

When everything works fine, you are maybe interested in how to start this AIS receiver automatically after plugging in the Raspberry Pi. The best way to do this, is to create an **AIS receiver service** at your Raspberry Pi. [Learn in this article how to start RTL-AIS automatically after reboot.](http://blog.videgro.net/2015/12/linux-let-your-program-run-as-background-service/)

* **Show the ships in a web browser**

[](http://blog.videgro.net/wp-content/uploads/2015/12/screenshot_openstreetmap_20151229_1705.png)

AIS data showed in an OpenStreetMap / OpenSeaMap. **(Click image to enlarge)**

I extended this example with my own application. My application is a *replacement*for the OpenCPN-client, it enables me to show the ships in my **web browser** (like: Firefox/Chrome) without the usage of an extra client.

The application consists of mainly three parts:

* 1. **Java 8** software which listens for *AIS/NMEA 0183* messages at a UDP port. Messages are received from ‘rtl\_ais’. The Java program translates them to *JSON* messages and will send them to another UDP port.
  2. A **Node.js** program which serves static *HTML*, *CSS* and *JavaScript* documents. It listens for *JSON messages* received from the Java program and retransmits them using **Socket.io**.
  3. The **HTML/JavaScript page** contains an *OpenStreetMap / OpenSeaMap* and is rendered in the web browser. The JavaScript part is listening for new *JSON messages* which are received using Socket.io. On receiving a new message,  a ship, name of ship, speed and trace will be plotted at the map. You can see additional information in a pop up by hovering over the ship.

This first two programs (Java- and Node.js) are running at my Raspberry Pi, part three is running in one or more (remote) web browser(s).

The ships are plotted in the OpenStreetMap / OpenSeaMap at their last received position. A trace at the back of the ship is showing the previous positions. The length of the line in front of the ship (green) is depicting the speed. It is possible to switch of one or more of the layers (Ships/Ship traces and OpenSeaMap). A red ship name indicates that we are not receiving any information from the ship any more, the ship and trace will be removed in a short time. On receiving information about a new ship, a bell sound and the name is pronounced using the **eSpeak** speech synthesizer / text-to-speech (TTS) engine.

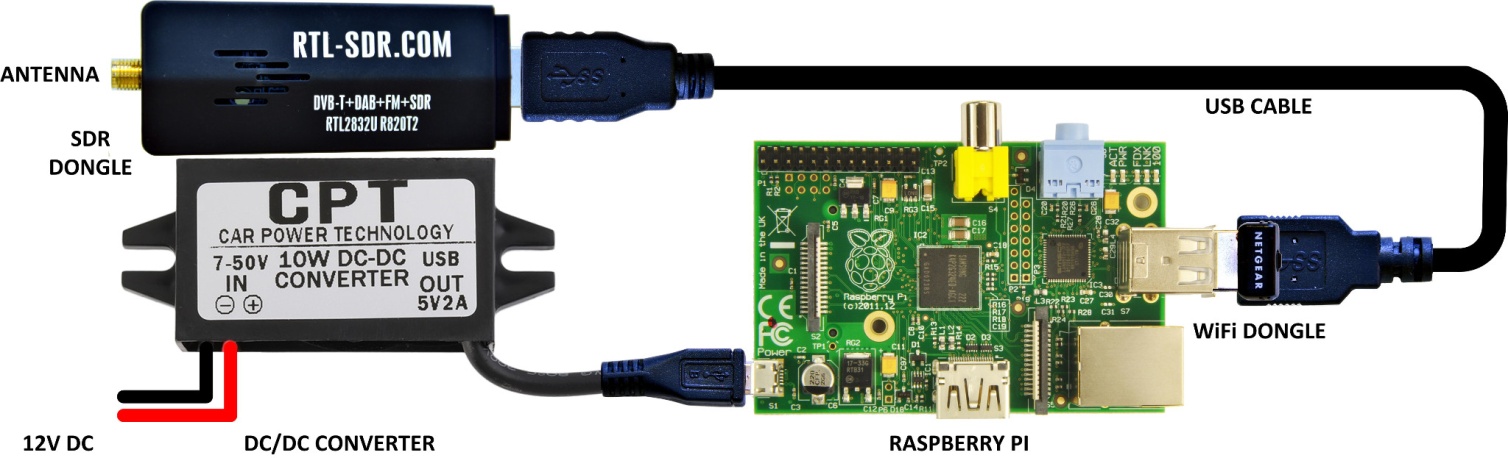
**More information**

* [RTL-SDR (RTL2832U) and software defined radio news and projects. Also featuring Airspy, HackRF, FCD, SDRplay and more.](http://www.rtl-sdr.com/)
* [Wikipedia – Automatic Identification System (AIS)](https://en.wikipedia.org/wiki/Automatic_Identification_System)
* [Wikipedia – NMEA 0183](https://en.wikipedia.org/wiki/NMEA_0183)

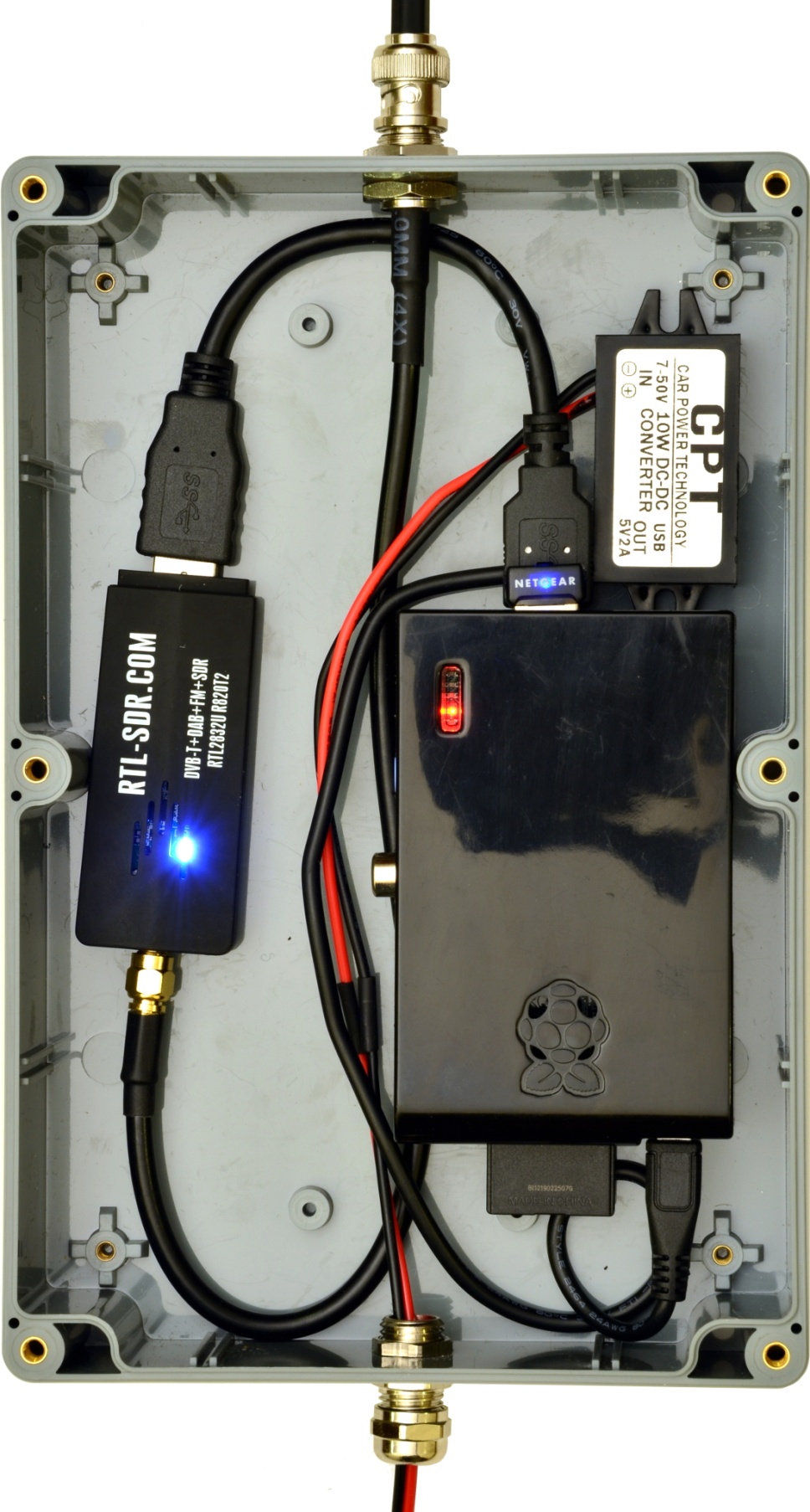
**AIS Receiver - Amateur Radio Magazine Article**

<http://www.sarcnet.org/projects/project_ais.html>

The draft article describing the inspiration and operation of the AIS receiver is [here](http://www.sarcnet.org/files/ais/AIS%20Saves%20Lives.pdf).



**AIS Receiver Pictorial Schematic**



**AIS Receiver Layout**

**Parts List**

The AIS receiver requires the following hardware and software:

* Antenna mast.
* 162MHz antenna.
* Weatherproof, non-metallic, enclosure and mounting hardware.
* Raspberry Pi single board computer version 1, 2 or 3, model B or B+.
* RTL-SDR USB dongle (TCXO 1ppm) with a 10cm USB extension cable.
* WiFi USB dongle to suit your home WiFi router (not required for Raspberry Pi version3).
* 5V 2A micro USB converter.
* 12V Plug Pack with two-core cable.
* 8GB SD or Micro SD card.

The RTL-SDR dongles are available from [www.RTL-SDR.com](http://www.rtl-sdr.com/). The Raspberry Pi and WiFi dongle from [www.element14.com](http://www.element14.com/). Other components and cables are readily available locally or search the name on eBay.

**Create a Raspberry Pi SD Card boot image**

Do the following on a Windows PC:

1. Download raspbian-jessie.zip from <https://www.raspberrypi.org/downloads/raspbian/>
2. Unzip raspbian-jessie.img
3. Download Win32DiskImager from <https://sourceforge.net/projects/win32diskimager/>
4. Install Win32DiskImager
5. Insert 8GB SD or microSD card (USB SD card adapter may be necessary)
6. Start Win32DiskImager
7. Select the image file and SD card above. Select Write.
8. Eject the SD card from the PC

**Set up the Raspberry Pi initial configuration using the desktop**

Do the following on a Raspberry Pi:

1. Insert the SD card into the Raspberry Pi
2. Connect Raspberry Pi version 1 temporarily to a USB hub. It can be removed later. This is not necessary for version 2 or 3.
3. Connect USB WiFi Dongle, USB keyboard and USB mouse. Do not connect USB RTL-SDR dongle yet.
4. Connect an HDMI monitor.
5. Power up the Raspberry Pi
6. Task Bar|Panel Settings|Panel Preferences|Edge: Bottom
7. Task Bar|WiFi|Select and connect to your WiFi Access Point
8. Task Bar|Menu|Preferences|Raspberry Pi Configuration
9. System|Filesystem|Expand Filesystem
10. System|Boot: To CLI
11. Localization|Set Locale
12. Localization|Set Timezone
13. Loalization|Set Wifi Country
14. Select OK
15. Reboot as suggested. (Say goodbye to the desktop)

**Upgrade, install and compile the AIS software**

In terminal mode now, type in the following commands to get and run our installation script:

1. wget http://www.sarcnet.org/files/ais/ais
2. chmod +x ais
3. ./ais
4. shutdown now

Note: Step 3 may take around 15-30 minutes. The software is distributed under the original GPL licences. No fitness for purpose is expressed or implied. The system must not be used for actual navigation. We decided to source copies of the original software from our web site because the original sources are subject to development and some have already been broken. We may update our copies from time to time.

**Check the RTL-SDR frequency offset in ppm**

This optional procedure uses kal and local GSM mobile base stations to check the frequency offset of the RTL SDR dongle.

Plug in the RTL-SDR dongle, connect a GSM antenna (or 80mm of wire) to it and cycle power to the Raspberry Pi.

At the prompt, press CTRL-C to exit starting up the AIS Receiver.

Type the following commands and note the results:

1. kal -s GSM900

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: Scanning for GSM-900 base stations.  
GSM-900:  
     chan: 8 (936.6MHz + 2.289kHz)     power: 40324.67  
     chan: 12 (937.4MHz + 1.677kHz)    power: 48893.99  
     **chan: 15 (938.0MHz + 2.250kHz)    power: 68151.80**  
     chan: 77 (950.4MHz + 1.764kHz)    power: 51462.37

Identify the channel with the strongest signal above (but not an absurd offset) then enter the channel number in the command below:

1. 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)  
   + 2.243kHz       [2218, 2267]     (50, 13.075513)  
   overruns: 0  
   not found: 0  
   average absolute error: **-2**.391 ppm

Round this number off to the nearest integer (e.g. -2) and remember it for the final configuration below.

**Set up a free on-line account with AIS Servers**

You will have to apply to AIS service organisations to get their IP address and port number. Only Pocketmariner publishes this data freely. Since AIS servers don't all share your data please register and upload to as many as possible. It will make the world a safer place for vessels. Here are some links:

1. **Pocketmariner**: <http://pocketmariner.com/ais-ship-tracking/cover-your-area/> **(IP Address/Port = 54.225.113.225:5322)**
2. **AISHub**: <http://www.aishub.net/ais-tracking-join-us.php>
3. **Marinetraffic**: <https://www.marinetraffic.com/en/users/register/1/12>
4. **Shipfinder**: <http://www.shipfinder.com/Login/reg>

**Complete the final startup configuration**

1. Cycle power to the AIS Receiver
2. At the prompt, press CTRL-C to exit starting up the AIS Receiver.
3. Type: **sudo nano /home/pi/.bash\_profile** to edit the terminal startup profile. It should looks like this:

echo "Starting AIS Receiver in 10 seconds. CTRL-C to exit"  
sleep 10  
setterm -blank 0 -powerdown 0  
sudo killall -w rtl\_ais  
sudo killall -w aisdispatcher  
#Please edit the following list of IP addresses AAA.BBB.CCC.DDD and port numbers PPPP for your AIS server(s). Separate by commas (no spaces).  
/usr/bin/aisdispatcher -u -G -h 127.0.0.1 -p 10110 -H **AAA.BBB.CCC.DDD:PPPP,AAA.BBB.CCC.DDD:PPPP** &  
#Please change the -p option to set the frequency offset in ppm of your RTL-SDR dongle  
/usr/bin/rtl\_ais -p **0** &

1. Use the keyboard to change the text in **bold**
   1. Enter your own list of AIS server IP addresses and port numbers
   2. Enter your own ppm offset (found using kal above) after -p. E.G. -p **-2**
   3. Type **CTRL-o <Enter>** to save your changes.
   4. Type **CTRL-x** to exit.
2. Type **passwd**to change the default password from "raspberry" to your own secure password.
3. Type **shutdown now**

**Reconfigure for stand-alone operation**

1. Disconnect USB keyboard and USB mouse
2. Disconnect the USB hub
3. Connect the USB WiFi dongle to the Raspberry Pi
4. Connect the USB RTL-SDR dongle to the Raspberry Pi via a USB extension cable
5. Connect the power. It should startup showing AIS sentences.
6. Disconnect the HDMI monitor.

**Log in remotely to your AIS Receiver**

1. You can log into the AIS receiver over your WiFi network to check its operation .
2. To do this you need to discover the IP address of your AIS receiver (check your WiFi router DHCP Leases. Look for hostname: **raspberrypi**)
3. Use an SSH terminal program (such as PuTTY) to log into that IP address at port 22.
4. Log in as user "**pi**" and use your password set above. You will see "Starting AIS Receiver in 10 seconds. CTRL-C to exit"
5. If you let the AIS receiver start you will see the received AIS sentences on your remote screen.
6. Before you logout of your remote session type: **sudo reboot** to return to stand-alone mode.

|  |  |
| --- | --- |
| **Re: OpenCPN on RPi3 Latest Raspbian**  Quote:   |  | | --- | | Originally Posted by **boat\_alexandra** [View Post](http://www.cruisersforum.com/forums/f134/opencpn-on-rpi3-latest-raspbian-166842-2.html#post2135015)  *Not using compression? It's going to suck at raster charts.  Can you post glxinfo output? I will try to download the image as well. Are you using raspbian?* |   Using raspbian.  Code:  name of display: :0.0  display: :0 [screen](http://www.cruisersforum.com/forums/tags/screen.html): 0  direct rendering: Yes  server glx vendor string: SGI  server glx version string: 1.4  server glx extensions:  GLX\_ARB\_create\_context, GLX\_ARB\_create\_context\_profile,  GLX\_ARB\_fbconfig\_float, GLX\_ARB\_framebuffer\_sRGB, GLX\_ARB\_multisample,  GLX\_EXT\_create\_context\_es2\_profile, GLX\_EXT\_framebuffer\_sRGB,  GLX\_EXT\_import\_context, GLX\_EXT\_texture\_from\_pixmap, GLX\_EXT\_visual\_info,  GLX\_EXT\_visual\_rating, GLX\_INTEL\_swap\_event, GLX\_MESA\_copy\_sub\_buffer,  GLX\_OML\_swap\_method, GLX\_SGIS\_multisample, GLX\_SGIX\_fbconfig,  GLX\_SGIX\_pbuffer, GLX\_SGIX\_visual\_select\_group, GLX\_SGI\_swap\_control  client glx vendor string: Mesa [Project](http://www.cruisersforum.com/forums/tags/project.html) and SGI  client glx version string: 1.4  client glx extensions:  GLX\_ARB\_create\_context, GLX\_ARB\_create\_context\_profile,  GLX\_ARB\_create\_context\_robustness, GLX\_ARB\_fbconfig\_float,  GLX\_ARB\_framebuffer\_sRGB, GLX\_ARB\_get\_proc\_address, GLX\_ARB\_multisample,  GLX\_EXT\_buffer\_age, GLX\_EXT\_create\_context\_es2\_profile,  GLX\_EXT\_fbconfig\_packed\_float, GLX\_EXT\_framebuffer\_sRGB,  GLX\_EXT\_import\_context, GLX\_EXT\_texture\_from\_pixmap, GLX\_EXT\_visual\_info,  GLX\_EXT\_visual\_rating, GLX\_INTEL\_swap\_event, GLX\_MESA\_copy\_sub\_buffer,  GLX\_MESA\_multithread\_makecurrent, GLX\_MESA\_query\_renderer,  GLX\_MESA\_swap\_control, GLX\_OML\_swap\_method, GLX\_OML\_sync\_control,  GLX\_SGIS\_multisample, GLX\_SGIX\_fbconfig, GLX\_SGIX\_pbuffer,  GLX\_SGIX\_visual\_select\_group, GLX\_SGI\_make\_current\_read,  GLX\_SGI\_swap\_control, GLX\_SGI\_video\_sync  GLX version: 1.4  GLX extensions:  GLX\_ARB\_create\_context, GLX\_ARB\_create\_context\_profile,  GLX\_ARB\_fbconfig\_float, GLX\_ARB\_framebuffer\_sRGB,  GLX\_ARB\_get\_proc\_address, GLX\_ARB\_multisample, GLX\_EXT\_buffer\_age,  GLX\_EXT\_create\_context\_es2\_profile, GLX\_EXT\_framebuffer\_sRGB,  GLX\_EXT\_import\_context, GLX\_EXT\_texture\_from\_pixmap, GLX\_EXT\_visual\_info,  GLX\_EXT\_visual\_rating, GLX\_INTEL\_swap\_event, GLX\_MESA\_copy\_sub\_buffer,  GLX\_MESA\_multithread\_makecurrent, GLX\_MESA\_query\_renderer,  GLX\_MESA\_swap\_control, GLX\_OML\_swap\_method, GLX\_OML\_sync\_control,  GLX\_SGIS\_multisample, GLX\_SGIX\_fbconfig, GLX\_SGIX\_pbuffer,  GLX\_SGIX\_visual\_select\_group, GLX\_SGI\_make\_current\_read,  GLX\_SGI\_swap\_control, GLX\_SGI\_video\_sync  OpenGL vendor string: Broadcom  OpenGL renderer string: Gallium 0.4 on VC4  OpenGL version string: 2.1 Mesa 11.1.0  OpenGL shading language version string: 1.20  OpenGL extensions:  GL\_AMD\_shader\_trinary\_minmax, GL\_APPLE\_packed\_pixels,  GL\_APPLE\_vertex\_array\_object, GL\_ARB\_ES2\_compatibility,  GL\_ARB\_buffer\_storage, GL\_ARB\_clear\_buffer\_object,  GL\_ARB\_color\_buffer\_float, GL\_ARB\_compressed\_texture\_pixel\_storage,  GL\_ARB\_copy\_buffer, GL\_ARB\_debug\_output, GL\_ARB\_depth\_texture,  GL\_ARB\_draw\_buffers, GL\_ARB\_draw\_elements\_base\_vertex,  GL\_ARB\_explicit\_attrib\_location, GL\_ARB\_explicit\_uniform\_location,  GL\_ARB\_fragment\_coord\_conventions, GL\_ARB\_fragment\_program,  GL\_ARB\_fragment\_program\_shadow, GL\_ARB\_fragment\_shader,  GL\_ARB\_framebuffer\_object, GL\_ARB\_framebuffer\_sRGB,  GL\_ARB\_get\_program\_binary, GL\_ARB\_get\_texture\_sub\_image,  GL\_ARB\_half\_float\_pixel, GL\_ARB\_half\_float\_vertex,  GL\_ARB\_internalformat\_query, GL\_ARB\_invalidate\_subdata,  GL\_ARB\_map\_buffer\_alignment, GL\_ARB\_map\_buffer\_range, GL\_ARB\_multi\_bind,  GL\_ARB\_multisample, GL\_ARB\_multitexture, GL\_ARB\_occlusion\_query,  GL\_ARB\_occlusion\_query2, GL\_ARB\_pixel\_buffer\_object,  GL\_ARB\_point\_parameters, GL\_ARB\_point\_sprite,  GL\_ARB\_program\_interface\_query, GL\_ARB\_provoking\_vertex,  GL\_ARB\_robustness, GL\_ARB\_sampler\_objects, GL\_ARB\_separate\_shader\_objects,  GL\_ARB\_shader\_objects, GL\_ARB\_shading\_language\_100, GL\_ARB\_shadow,  GL\_ARB\_sync, GL\_ARB\_texture\_border\_clamp, GL\_ARB\_texture\_compression,  GL\_ARB\_texture\_cube\_map, GL\_ARB\_texture\_env\_add,  GL\_ARB\_texture\_env\_combine, GL\_ARB\_texture\_env\_crossbar,  GL\_ARB\_texture\_env\_dot3, GL\_ARB\_texture\_mirrored\_repeat,  GL\_ARB\_texture\_multisample, GL\_ARB\_texture\_non\_power\_of\_two,  GL\_ARB\_texture\_rectangle, GL\_ARB\_texture\_storage,  GL\_ARB\_texture\_storage\_multisample, GL\_ARB\_transpose\_matrix,  GL\_ARB\_vertex\_array\_bgra, GL\_ARB\_vertex\_array\_object,  GL\_ARB\_vertex\_attrib\_binding, GL\_ARB\_vertex\_buffer\_object,  GL\_ARB\_vertex\_program, GL\_ARB\_vertex\_shader, GL\_ARB\_window\_pos,  GL\_ATI\_blend\_equation\_separate, GL\_ATI\_draw\_buffers,  GL\_ATI\_separate\_stencil, GL\_ATI\_texture\_env\_combine3, GL\_EXT\_abgr,  GL\_EXT\_bgra, GL\_EXT\_blend\_color, GL\_EXT\_blend\_equation\_separate,  GL\_EXT\_blend\_func\_separate, GL\_EXT\_blend\_minmax, GL\_EXT\_blend\_subtract,  GL\_EXT\_compiled\_vertex\_array, GL\_EXT\_copy\_texture,  GL\_EXT\_draw\_range\_elements, GL\_EXT\_fog\_coord, GL\_EXT\_framebuffer\_blit,  GL\_EXT\_framebuffer\_multisample, GL\_EXT\_framebuffer\_multisample\_blit\_scaled,  GL\_EXT\_framebuffer\_object, GL\_EXT\_framebuffer\_sRGB,  GL\_EXT\_gpu\_program\_parameters, GL\_EXT\_multi\_draw\_arrays,  GL\_EXT\_packed\_depth\_stencil, GL\_EXT\_packed\_pixels,  GL\_EXT\_pixel\_buffer\_object, GL\_EXT\_point\_parameters,  GL\_EXT\_polygon\_offset, GL\_EXT\_provoking\_vertex, GL\_EXT\_rescale\_normal,  GL\_EXT\_secondary\_color, GL\_EXT\_separate\_specular\_color,  GL\_EXT\_shader\_integer\_mix, GL\_EXT\_shadow\_funcs, GL\_EXT\_stencil\_two\_side,  GL\_EXT\_stencil\_wrap, GL\_EXT\_subtexture, GL\_EXT\_texture, GL\_EXT\_texture3D,  GL\_EXT\_texture\_cube\_map, GL\_EXT\_texture\_edge\_clamp,  GL\_EXT\_texture\_env\_add, GL\_EXT\_texture\_env\_combine,  GL\_EXT\_texture\_env\_dot3, GL\_EXT\_texture\_lod\_bias, GL\_EXT\_texture\_object,  GL\_EXT\_texture\_rectangle, GL\_EXT\_texture\_sRGB, GL\_EXT\_texture\_sRGB\_decode,  GL\_EXT\_vertex\_array, GL\_EXT\_vertex\_array\_bgra,  GL\_IBM\_multimode\_draw\_arrays, GL\_IBM\_rasterpos\_clip,  GL\_IBM\_texture\_mirrored\_repeat, GL\_INGR\_blend\_func\_separate,  GL\_KHR\_context\_flush\_control, GL\_KHR\_debug, GL\_MESA\_pack\_invert,  GL\_MESA\_window\_pos, GL\_NV\_blend\_square, GL\_NV\_fog\_distance,  GL\_NV\_light\_max\_exponent, GL\_NV\_packed\_depth\_stencil,  GL\_NV\_texgen\_reflection, GL\_NV\_texture\_env\_combine4,  GL\_NV\_texture\_rectangle, GL\_OES\_EGL\_image, GL\_OES\_read\_format,  GL\_SGIS\_generate\_mipmap, GL\_SGIS\_texture\_border\_clamp,  GL\_SGIS\_texture\_edge\_clamp, GL\_SGIS\_texture\_lod, GL\_SUN\_multi\_draw\_arrays  OpenGL ES profile version string: OpenGL ES 2.0 Mesa 11.1.0  OpenGL ES profile shading language version string: OpenGL ES GLSL ES 1.0.16  OpenGL ES profile extensions:  GL\_APPLE\_texture\_max\_level, GL\_EXT\_blend\_minmax,  GL\_EXT\_discard\_framebuffer, GL\_EXT\_draw\_buffers,  GL\_EXT\_draw\_elements\_base\_vertex, GL\_EXT\_map\_buffer\_range,  GL\_EXT\_multi\_draw\_arrays, GL\_EXT\_read\_format\_bgra,  GL\_EXT\_separate\_shader\_objects, GL\_EXT\_texture\_format\_BGRA8888,  GL\_EXT\_texture\_type\_2\_10\_10\_10\_REV, GL\_EXT\_unpack\_subimage,  GL\_KHR\_context\_flush\_control, GL\_KHR\_debug, GL\_NV\_draw\_buffers,  GL\_NV\_fbo\_color\_attachments, GL\_NV\_read\_buffer, GL\_NV\_read\_depth,  GL\_NV\_read\_depth\_stencil, GL\_NV\_read\_stencil, GL\_OES\_EGL\_image,  GL\_OES\_EGL\_image\_external, GL\_OES\_EGL\_sync,  GL\_OES\_compressed\_ETC1\_RGB8\_texture, GL\_OES\_depth24, GL\_OES\_depth\_texture,  GL\_OES\_draw\_elements\_base\_vertex, GL\_OES\_element\_index\_uint,  GL\_OES\_fbo\_render\_mipmap, GL\_OES\_get\_program\_binary, GL\_OES\_mapbuffer,  GL\_OES\_packed\_depth\_stencil, GL\_OES\_rgb8\_rgba8, GL\_OES\_stencil8,  GL\_OES\_surfaceless\_context, GL\_OES\_texture\_3D, GL\_OES\_texture\_npot,  GL\_OES\_vertex\_array\_object |