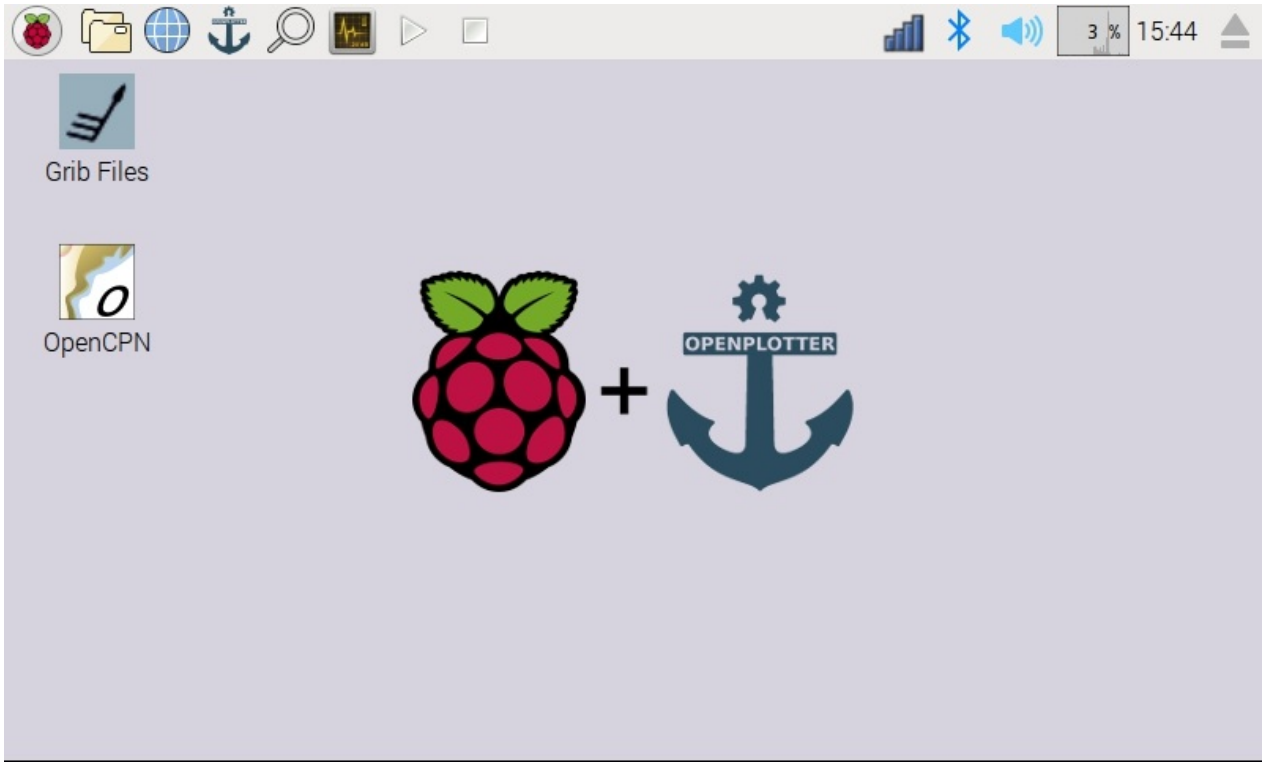
Table of Contents

What is OpenPlotter?

There are people who buy boats but there are also people who build them, why not build your own electronics too? OpenPlotter is a combination of software and hardware to be used as navigational aid on small and medium boats. It is also a complete home automation system onboard. It works on ARM computers like the Raspberry Pi[1] and is open-source, low-cost and low-consumption. Its design is modular, so you just have to implement what your boat needs. Do it yourself.

Features

- **OpenCPN**[2]. Concise and robust navigation software. Chart a course and track your position.
- **zyGrib**[3]. Weather data download and visualization of GRIB files format 1 and 2.
- **SDR-AIS**. Receive and decode AIS with cheap DVB-T dongles. Calibration tools Included.
- **NMEA 0183**. Receive, filter, multiplex and send NMEA data through serial and network inputs and outputs.
- **NMEA 2000**. Connect an USB adapter to your N2K network to receive and send data.
- **Signal K Server**[4]. Convert data from sensors to Signal K and exchange data between all formats.
- **WiFi Access Point**. Share NMEA and Signal K data with laptops, tablets and phones.
- **Headless**. Access to OpenPlotter desktop from the cockpit through your mobile devices.
- **Dashboards**. Build instrument panels to visualize data from your mobile devices.
- **Compass**. Heading and heel from an IMU sensor. Tilt compensated. Self-calibration.
- **1W sensors**. Connect multiple temperature sensors. Motor, exhaust, fridge, water ...
- **I2C sensors**. Connect temperature, pressure, humidity or light sensors.
- **SPI sensors**. Measure battery charge, tank level or any data from an analog sensor.
- **Digital inputs**. Connect common or specials switches. Detect smoke, opening doors, motion ...
- **Digital outputs**. Activate relays, LEDs, buzzers ...
- **Actions**. Use date, time or value of any data flowing through your system as a trigger to run multiple actions.
- **Share data**. Publish data from your boat on Twitter or send emails and SMS automatically.
- **Remote Monitoring**. Receive or send data to your boat while you are away.
- **Open tools**. Use multiple built in tools to interact with the system and create your own ones.



[1] https://www.raspberrypi.org [2] http://opencpn.org [3] http://www.zygrib.org [4] http://signalk.org

Frequently Asked Questions

This chapter needs to be written/updated/translated: http://forum.openmarine.net/forumdisplay.php?fid=16

OpenPlotter

Q: Where is OpenPlotter installed? **A**: OpenPlotter python scripts are installed in /home/pi/.config/openplotter. Openplotter private configuration is stored in /home/pi/.openplotter

Q: Why can't I see the directory .config under /home/pi in the file manager? **A**: In the file manager you have to activate "Show Hidden" (menu View->Show Hidden).

Q: Can I manually install OpenPlotter? **A**: There is no pip install for OpenPlotter. If you have a working OpenPlotter you can rename the openplotter folder "mv /home/pi/.config/openplotter /home/pi/.config/openplotter_bak". Then you can unzip a downloaded openplotter.zip file from github, or you can clone an openplotter version from github "cd /home/pi/.config" and then "git clone -b beta https://github.com/sailoog/openplotter.git". After that you have to make three files executable."cd /home/pi/.config/openplotter", "chmod 755 openplotter","chmod 755 startup", "chmod 755 keyword".

Linux

Q: How can I see if my 1wire sensors (DS18B20) are recognized? **A**: In the terminal enter command to change directory "cd /sys/bus/w1/devices" and then list the installed sensors with "ls". If they don't show up here OpenPlotter can't find them.

Q: How can I check which usb devices are connected/recognized? **A**: Use the "lsusb" command. It shows all connected usb devices. Three devices are integrated in the raspberry pi (you can't disconnect them).

Signal K

Q: How to connect from smartphone to SignalK Instrumentpanel?

A: Be sure you are connected to the openplotter access point. Open your browser. Type in 10.10.10.1:3000. Click on Instrumnetpanel. On the right side of the Instrumentpanel is a sign select it to open the menubar. Select *Connect* and enter in *New SignalK Server* 10.10.10.1:3000

Q: Can I create my own SignalK names?

A: You can but it is recommended that you don't (to develop more standard names look for specifications in Github SignalK). If you want to have an overview of already existing SignalK names open the *diagnostic SignalK input* window and click on *Unit Setting*. If there is a star in the SignalK name you should put in an easy understandable name.

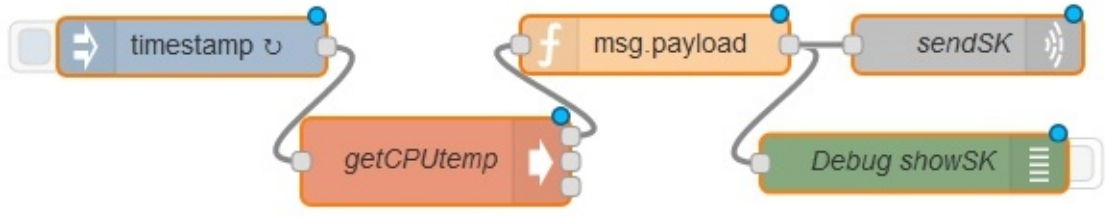
Node-Red

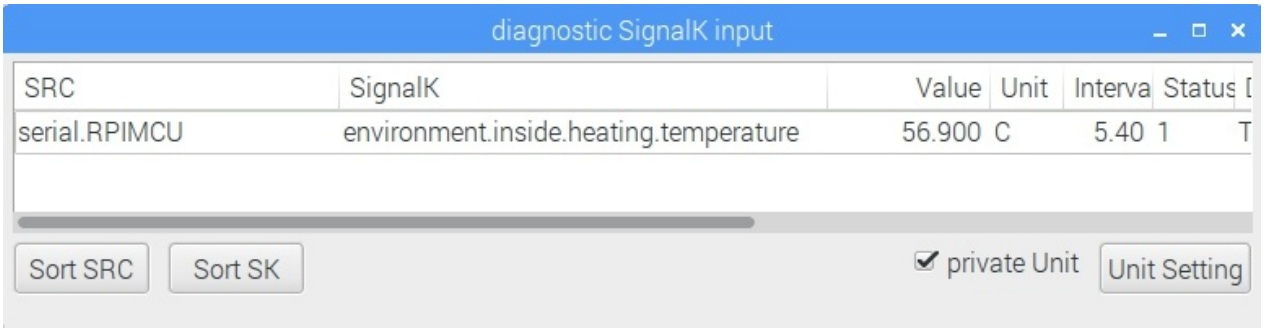
Q: Is there also a chance to send values to the SignalK server of OpenPlotter?

A: Yes you can send values to the server when you use the minimum SignalK JSON format. For example:

```
{
    "updates": [{
            "source": {
                "type": "ARMTEMP",
                "src": "RPIMCU"
            },
            "values": [{
                    "path": "environment.inside.heating.temperature",
                    "value": "323.15"
                }
            ]
        }
    ]
}
```

In this example we want to send the RPI cpu temperature to the SignalK Server.





We diidn't find a better SignalK name than environment.inside.heating.temperature.

You can import the code to test it on your own (the import does work when you copy this code from the internet html file).

```
[{"id":"f5fa759f.be894","type":"inject","z":"4deb498f.fbcfd8","name":"","topic":"","payload":"","payloadType":"date",
"repeat":"10","crontab":"","once":true,"x":131,"y":127,"wires":[["adb4b4a8.9c10c8"]]},{"id":"adb4b4a8.9c10c8","type":
"exec","z":"4deb498f.fbcfd8","command":"vcgencmd","addpay":false,"append":"measure_temp","useSpawn":"","timer":"","na
me":"getCPUtemp","x":258,"y":186.5,"wires":[["2705f134.792f86"],[],[]]},{"id":"7b5088a.c5f0af8","type":"debug","z":"4
deb498f.fbcfd8","name":"Debug showSK","active":false,"console":"false","complete":"payload","x":507,"y":187.5,"wires"
:[]},{"id":"2705f134.792f86","type":"function","z":"4deb498f.fbcfd8","name":"msg.payload","func":"cpu_temp = parseFlo
at(msg.payload.replace(\"temp=\",\"\").replace(\"'C\\n\",\"\"));\ncpu_temp = cpu_temp + 273.15\nmsg.payload = '{\\\"u
pdates\\\": [{\\\"source\\\": {\\\"type\\\": \\\"ARMTEMP\\\",\\\"src\\\" : \\\"RPIMCU\\\"},\\\"values\\\":[{\\\"path\
\\": \\\"environment.inside.heating.temperature\\\",\\\"value\\\":'+cpu_temp+'}]}]}\\n';\nreturn msg;","outputs":1,"n
oerr":0,"x":372,"y":126,"wires":[["7b5088a.c5f0af8","c17f4c75.737088"]]},{"id":"c17f4c75.737088","type":"udp out","z":
"4deb498f.fbcfd8","name":"sendSK","addr":"localhost","iface":"","port":"55559","ipv":"udp4","outport":"","base64":fal
se,"multicast":"false","x":535,"y":126,"wires":[]}]
```

How to collaborate

> Everything takes time, money, and monkeys. You need a lot from any two groups, and a little from the third. An increase in any one reduces the requirement for the other two. Change occurs when one of those three change.

Moe's Law, Navigatrix[1] project.

Time

Get a Raspberry Pi and the required elements, download OpenPlotter RPI and test and test and test ...

Report bugs and request new features on OpenMarine forums[2].

Spread the word among your friends in ports and forums.

Money

This project is financed by selling related and tested products or by voluntary contributions. You can buy or donate on the WebShop[3].

Monkeys

> Men wanted for hazardous journey. Low wages, bitter cold, long hours of complete darkness. Safe return doubtful. Honour and recognition in event of success.

Ernest Shackleton

If you have python skills, push your commits to the github repository[4].

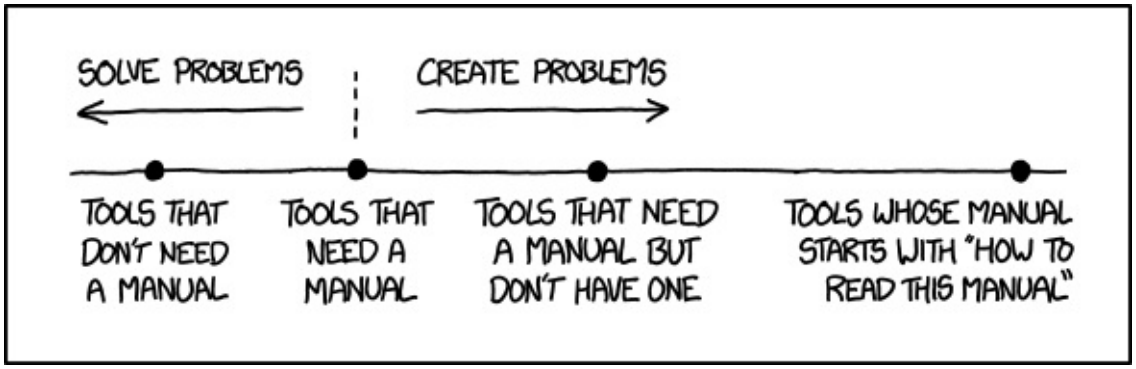
If you have electronics skills, share your work on OpenMarine forums[2].

If you have English language skills, helps us with documentation. Follow the steps in the next chapter Documentation.

---

[1] http://navigatrix.net [2] http://forum.openmarine.net [3] http://shop.sailoog.com [4] https://github.com/sailoog/openplotter

Documentation



Author: Randall Munroe[1], CC BY-NC 2.5 license.

We have a Github repository[2] for OpenPlotter documentation. This repository is connected to a Gitbook book[3] and we coordinate editors in a forum[4] on OpenMarine.

if you want to help us to write, update, correct or translate you have two options.

You are familiar with Github and Gitbook repositories edition:

- Create an account in **Github**[5]
- Send your Github account user name to the forum[4] and wait for you to be granted permissions to edit the repository.
- From the sticky threads in the forum[4], select a chapter marked in blue and notify your intentions in a new thread.
- Once you have permissions, login to **Gitbook**[6] with your Github account.
- Go to the book[3] and click on the *Edit* button upper right.

You are not familiar with Github and Gitbook repositories edition:

- From the sticky threads in the forum[4], select a chapter marked in blue and notify your intentions in a new thread.
- When you are done, publish text and images into a new thread in the forum[4] and someone will edit the book for you.

Writing

We write the documentation source in English to make easier translating to other languages.

Please do not modify the index or the structure without consulting developers. Once a chapter is done, please notify on the forum[4] to be marked in green.

To include images in the rest of languages from English folder, the path is `../en/xxx.png`

If you want to add images of wiring and connections consider using fritzing[7] application.

Do not modify or translate files names (xxx.md xxx.png xxx.jpg ...) because they are referenced in the text.

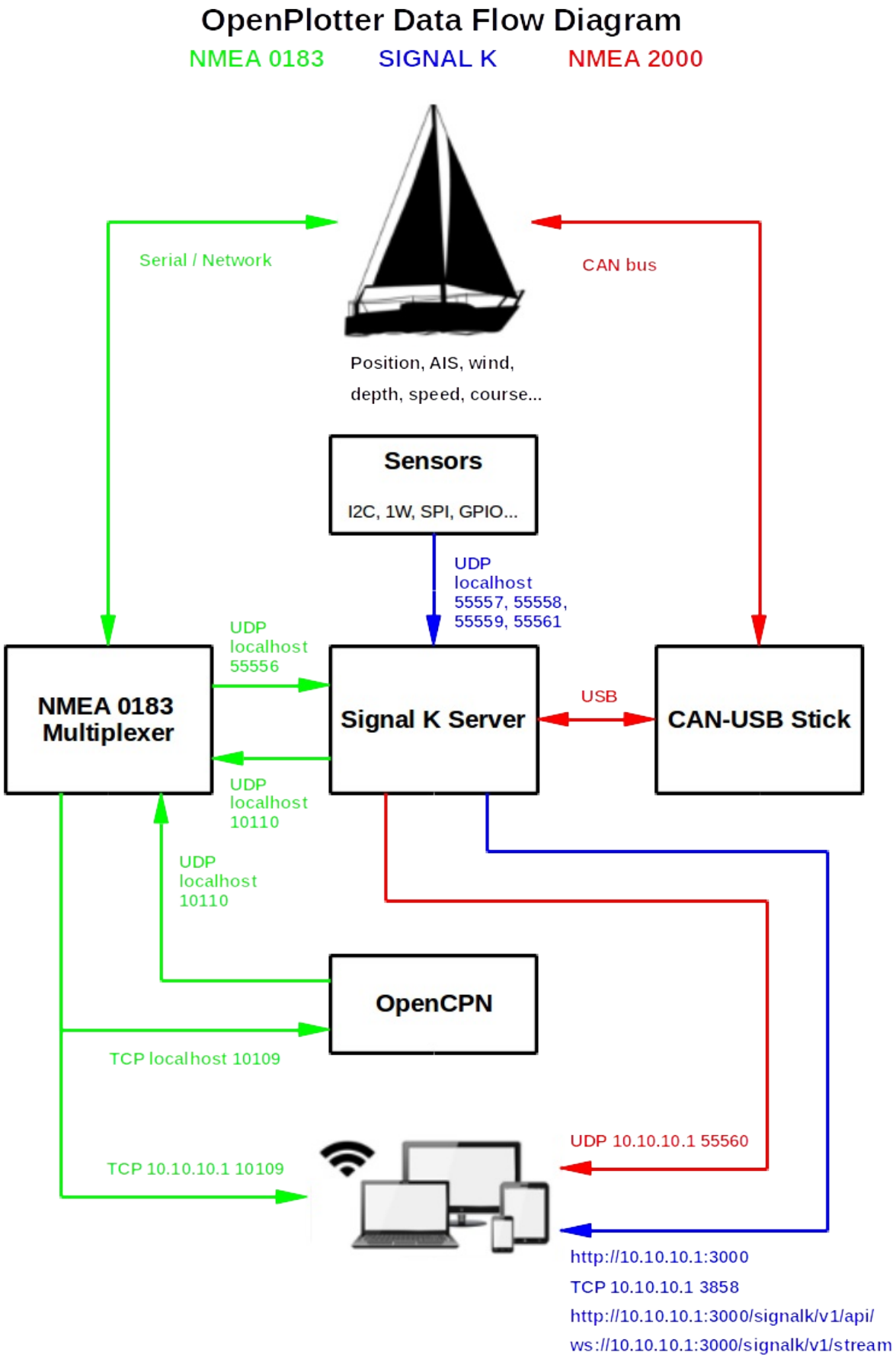
---

[1] https://xkcd.com [2] https://github.com/sailoog/openplotter-documentation [3] https://www.gitbook.com/book/sailoog/openplotter-documentation/details [4] http://forum.openmarine.net/forumdisplay.php?fid=16 [5] https://github.com/join [6] https://www.gitbook.com/login [7] http://fritzing.org

How does it work?

This chapter needs to be written/updated/translated

http://forum.openmarine.net/forumdisplay.php?fid=16



OpenPlotter can collect/serve data from/to different devices:

- Analog and digital sensors connected to GPIO port.
- Serial and other devices connected to USB port.

- Any computer or portable device connected to the same network.

Inputs

Most of these sources directly send data in the maritime formats called NMEA 0183 or NMEA 2000. The problem is that NMEA 0183 can not communicate with NMEA 2000 and there are non-strictly maritime data that are not defined in the NMEA schemas.

We use the free maritime communication Signal K schema to connect both NMEA formats. Other devices like SDR AIS, analog and digital sensors, switches.. are processed by OpenPlotter to convert raw data into Signal K too. Signal K has been the OpenPlotter base protocol since v0.9.x.

Outputs

Once we have all data stored in Signal K schema, we can convert data into both NMEA formats, process data to generate new Signal K data or set conditions to trigger defined actions. We can also server directly Signal K data to the connected devices.

Uses

Different pieces of software and hardware need different data streams.

The good old NMEA 0183 is a readable serial protocol used for:

- Most current PC, tablets and mobile software.
- The internal chartplotter, OpenCPN.
- With OpenPlotter tools, we can build obsolete, non-standard, proprietary and missing NMEA 0183 sentences from Signal K.

NMEA 2000 is special CAN-BUS protocol. It's advantage is the data security. It is popular in the automotive industrie. NMEA 2000, is used for:

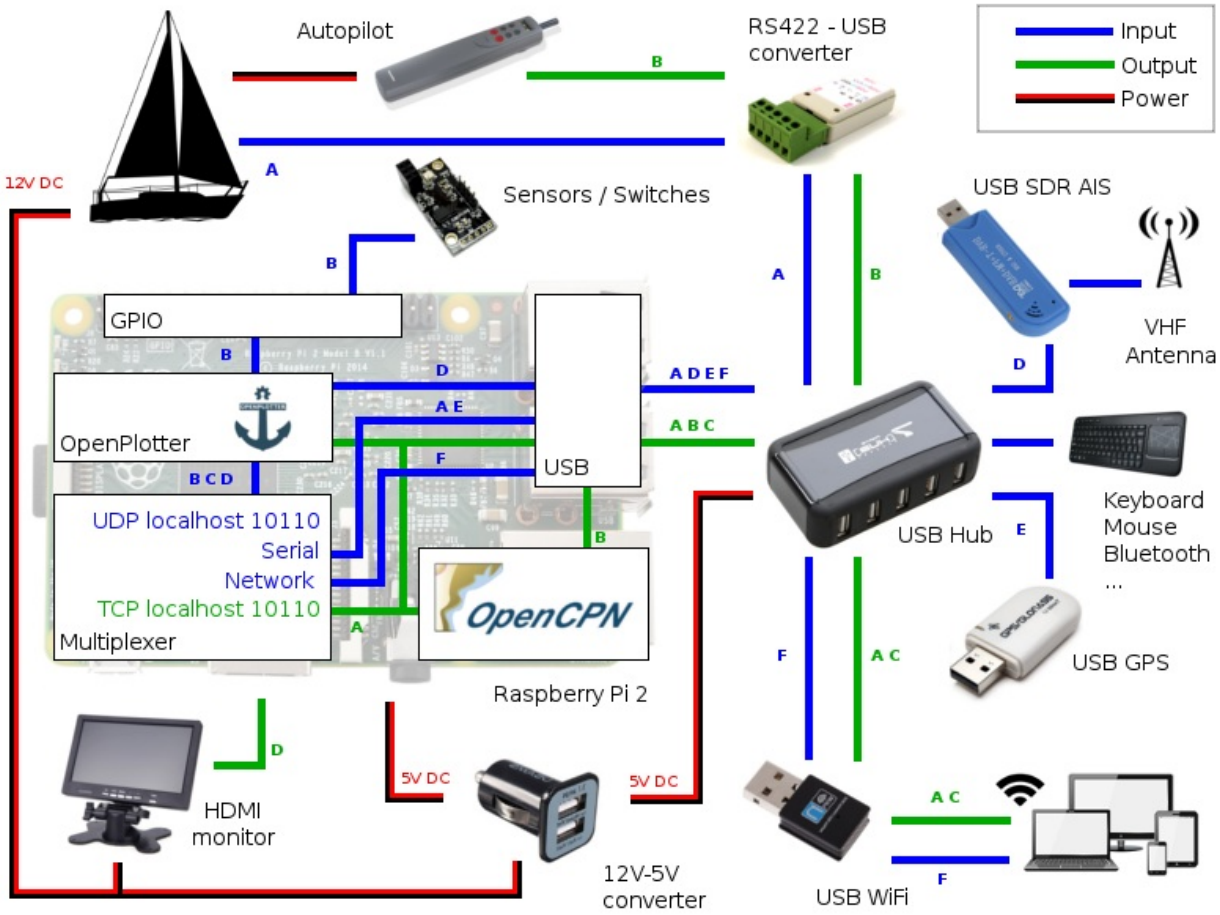
- Most modern boats (plotter, engine...).
- Very little software.

The browsers optimized protocol Signal K is used for:

- Future software.
- Virtual instrument panels.
- It has an open and readable protocol where additions can be done.

The non-strictly maritime datastream is used for:

- Security.
- Alarms.
- Energy savings.
- Remote automation.
- ...



Picture: another data flow example

What do you need?

You will need the required parts and some optional parts. It will depend on what kind of data you want to collect, process or display and what kind of equipment your boat already has.

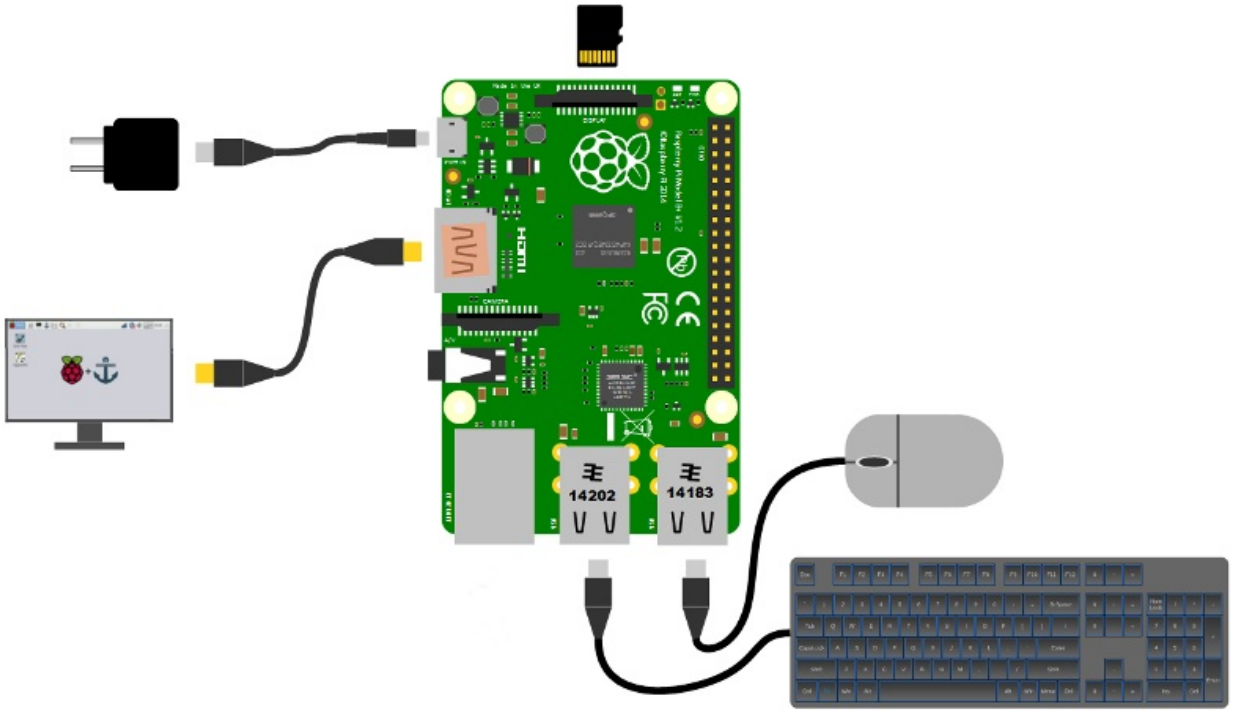
Required items

You need at least these items to run *the software* and you have two options: either **with monitor** or without monitor (**headless**).

If you need help with these basic parts see the official Raspberry Pi documentation[1].

Monitor option

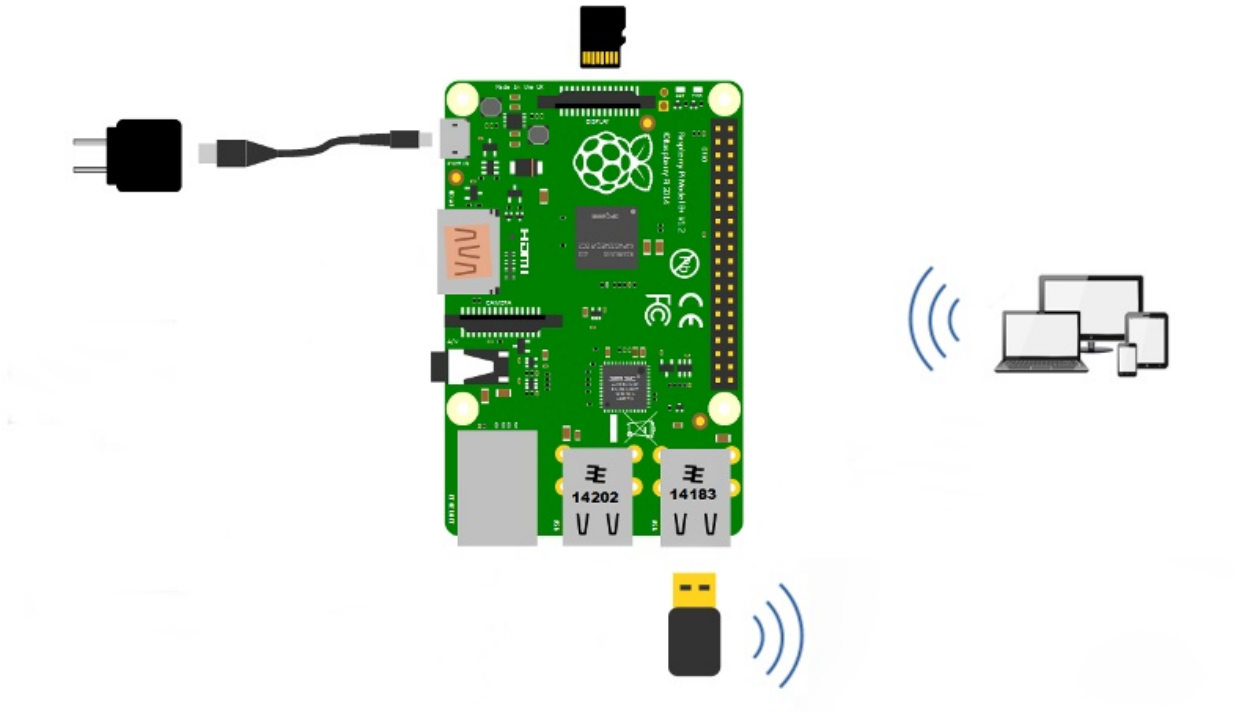
- **Raspberry Pi 3** or Raspberry Pi 2
- **Box**
- **Power supply**
- **HDMI monitor**
- **Keyboard and mouse**
- **SD card with OpenPlotter RPI** (*the software*)



Author: Malcolm Maclean[4], modified by Sailoog.

Headless option

- **Raspberry Pi 3** or Raspberry Pi 2 + WiFi dongle
- **Box**
- **Power supply**
- **SD card with OpenPlotter RPI** (*the software*)
- **Any laptop, tablet or smartphone**



Author: Malcolm Maclean[4], modified by Sailoog.

Where can you find them?

You can buy these basic parts from any official distributor[2] or any Raspberry store.

Occasionally we may have some of them on our store[3].

Optional items

Next chapters contain a list of supported devices to communicate with boat and sensors to collect data from environment.

Where can you find them?

Occasionally we may have some of them on our store[3].

---

[1] https://www.raspberrypi.org/learning/hardware-guide [2] https://www.raspberrypi.org/products [3] http://shop.sailoog.com
[4] https://leanpub.com/RPiMRE/read

Self powered USB Hub

If you are connecting devices which use more power than your Raspberry can provide, you will need a self powered USB Hub.

You can start connecting devices to the raspberry and switch to a self powered hub when you start to see strange behavior of the devices.

Hubs with **FE1.1S** chip work right, avoid hubs with MA8601 chip. Often the same model can contain any of them. And some of them do not contain chip at all! just a glob of black material to hide what is underneath ... nothing.

It would be a good idea to get a Hub with 5V input so that you could power it with the same source as your Raspberry.

You can also get a Hub with 12v input. Sometimes those devices have an output to charge phones or tablets which probably you could use to power your Raspberry.

USB WiFi dongle

You can set OpenPlotter either as access point to allow devices to connect to it or as client to connect to any access point. You can not do this at the same time with just one WiFI device so you will need an extra USB WiFi dongle for Raspberry 3 and two USB WiFi dongles for Raspberry 2 to do this.

Being connected to an access point as client to get Internet connection and at the same time running OpenPlotter as access point, you will be able to share Internet connection with all devices connected to OpenPlotter.

A good WiFi adapter will probably need more power than the Raspberry Pi USB port can provide, especially if there is a large distance from the WiFi adapter to the WiFi Access Point, or it is transferring large amounts of data. Therefore, you may need to plug the WiFi adapter into a powered USB hub.

Not all WiFi dongles can function as access point, only devices with the correct chipset will work. We recommend **RTL8192CU/CUS** chipset.

Settings

See WiFI AP chapter to configure OpenPlotter as client or access point.

USB GPS dongle

If you don't have any GPS on board or you want an extra positioning device, this is the cheapest and most effective way.

Connecting an USB GPS dongle to OpenPlotter will provide accurate position, date/time and speed/course over ground.

> This item is available in our store[1]

Settings

See NMEA 0183 chapter to configure USB GPS dongles.

---

[1] http://shop.sailoog.com

NMEA 0183 to USB converter

If you have electronics with NMEA 0183 outputs on board (depth, wind, heading...), you will need an USB converter to connect it to OpenPlotter. Additionally, if this converter is bi-directional, you will be able to talk to electronics with NMEA 0183 inputs like the autopilot.

The NMEA 0183 hardware standard uses **RS422** connectors but you may find some devices with **RS232** as well. Find out about what type of connection you need. In general, if you have a TX+ and a TX- and/or a RX- and RX+ you have a RS422 interface. If you ave a RX and a TX and a ground you have an RS232 interface.

> This item is available in our store[1]

Settings

See NMEA 0183 chapter to configure NMEA 0183 to USB converters.

---

[1] http://shop.sailoog.com

CAN-USB Stick

Project

The CAN-USB Stick project was done to analyse the data stream on a N2K network sending and receiving CAN messages. It is electrically isolated to avoid damages.

The program of the MCU has been re-engineered to work together with CANBOAT project[2], which is used by Signal K project[3]. Both packets are used in OpenPlotter project.

The CAN-USB Stick does also work with OpenSkipper project[4].

Not tested:

- MacENC[5]
- PolarView NS[6]

So it does use the command set which is used in the CANBOAT actisense-serial program. Sending and receiving data into the N2K network can be done directly from OpenPlotter too.

New PGNs are not blocked, as they are on other devices capable to work with CANBOAT. The transmission speed can be set higher than the CAN bus speed. Other devices capable to work with CANBOAT have a lower transferrate than N2K networks and they may suffer a bottleneck.

> This item is available in our store[9]

Hardware

The CAN-USB Stick V2 is based on a stm32 micro-controller (MCU) connected to an isolated CAN transceiver and an USB to serial converter.

Warning / Disclaimer

CAN-USB Stick is a research project on data communication on CAN bus and N2K networks in boats.

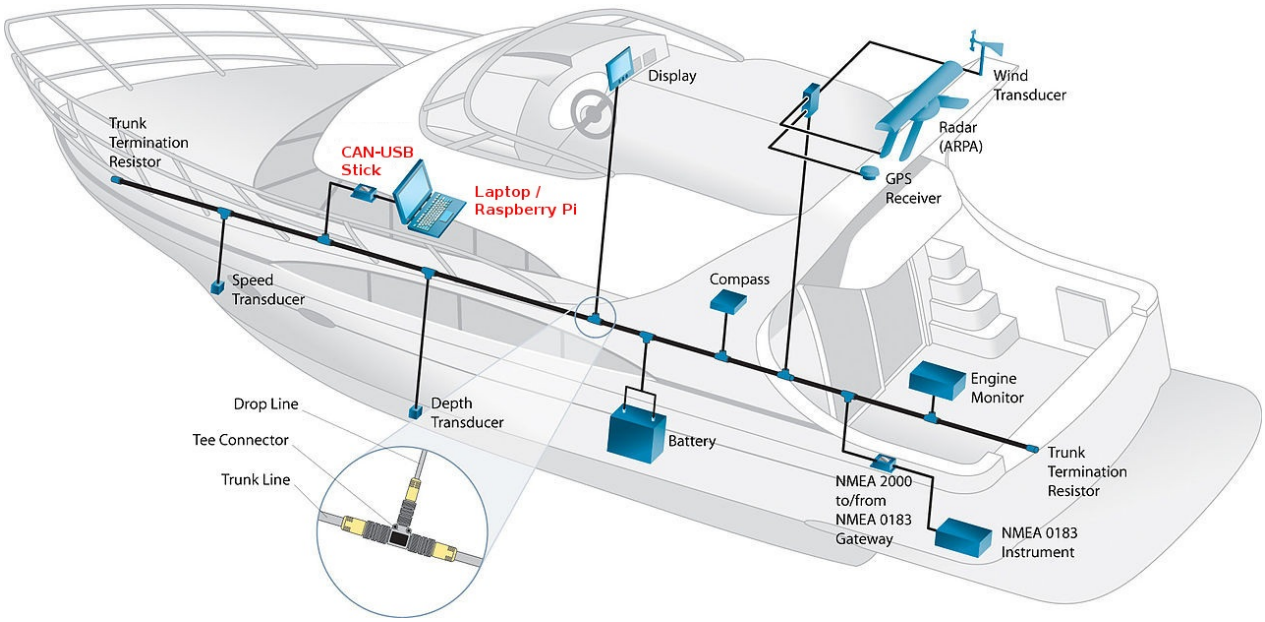
The software is still under development and has not been fully tested. Malfunctions of the CAN-USB Stick or any connected device might be possible at any time. Manipulating your N2K network could cause damage to connected devices.

Do not rely on data from this device and do not use it as primary source for navigation. Liability cannot be accepted for any damages, personal injuries or malfunctions caused by this device.

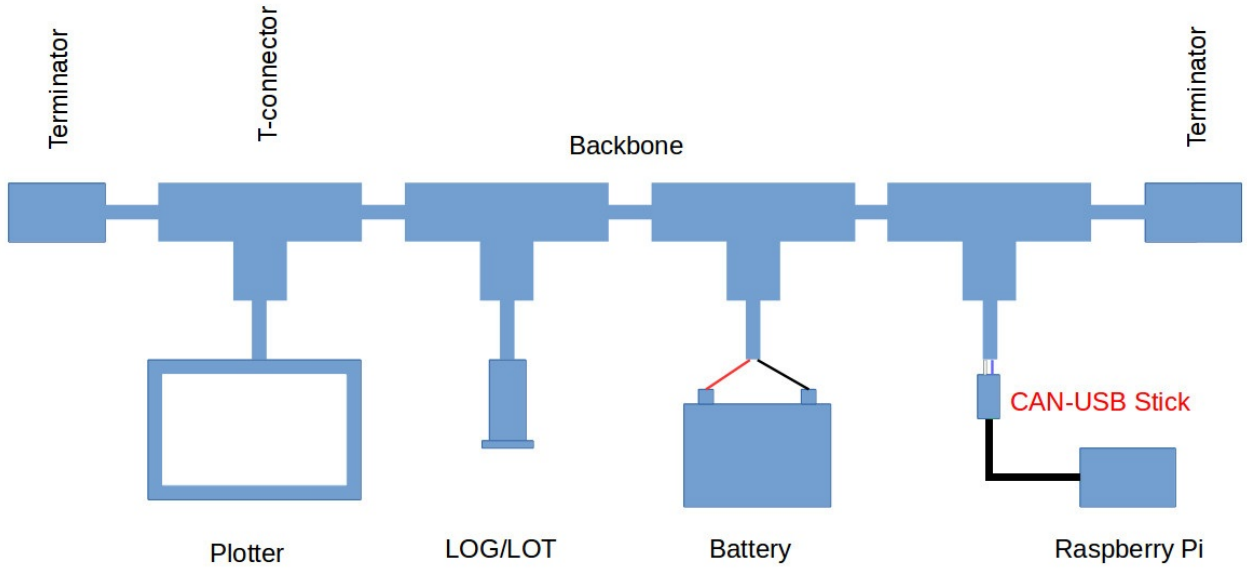
The CAN-USB Stick is not certified by NMEA®.

It is not allowed to use the Actisense® NMEA Reader software for the CAN-USB Stick.

N2K networks



Author: Femnett/Maretron[1], modified by Sailoog, CC BY 2.5 license.



Example of a small N2K Network.

N2K networks are described in Wikipedia[7]. The backbone (or trunk) starts with a 120Ω terminator and ends with a 120Ω terminator. Two resistors are working in parallel, so the resistance is 120Ω/2=60Ω. If there is a broken connection in the backbone you can measure only 120Ω or nothing but not 60Ω. That is a very easy way to check the bus.

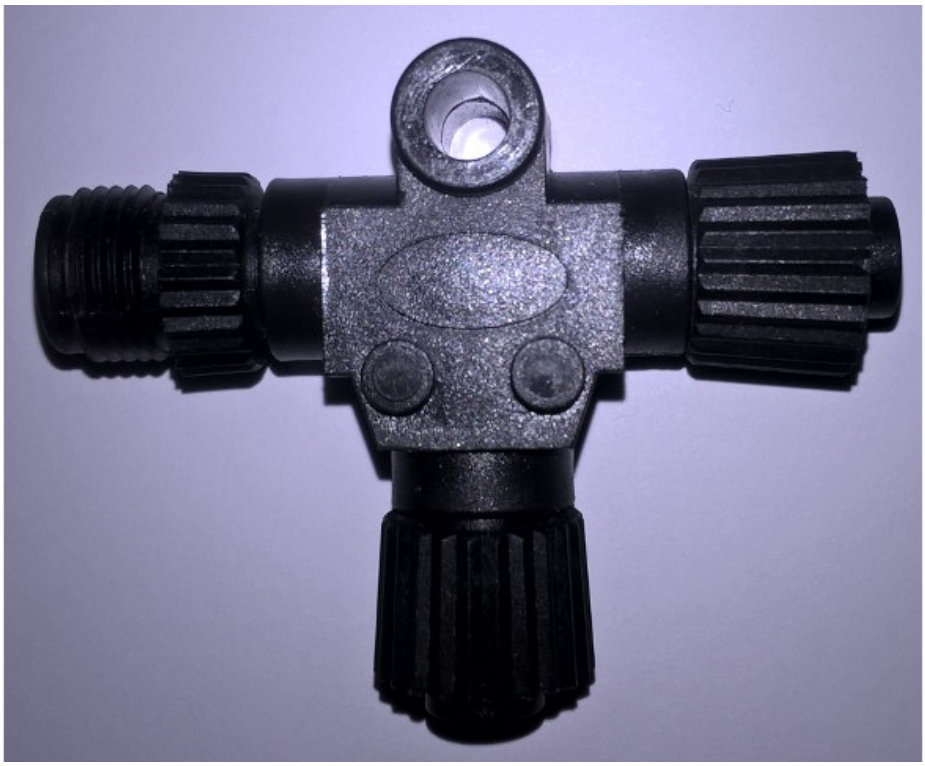
M12 male 120Ω terminator

The drop line to devices should not be longer than 6 m. The backbone can have 100m in length.

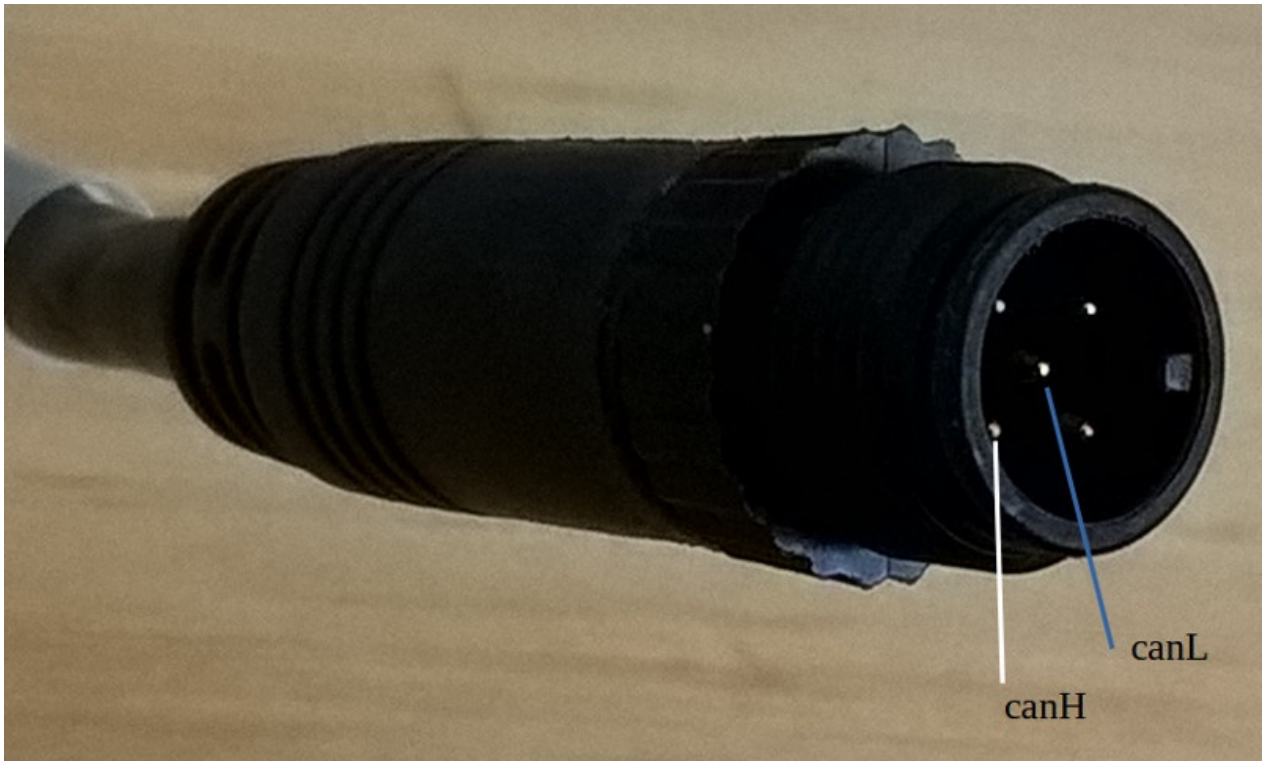
The CAN-USB Stick is electrically isolated so devices and your computer are protected even if they are powered by a diffent source than your N2K network.

Connection

To connect the CAN-USB Stick to the network you need a free T-connector on your backbone and a drop line. The drop line should have a M12 5 pin male connector in one side and 5 wires (but we only need 2) in the other side. The HIRSCHMANN ELST 5012 PG7 connector has a screw terminal.

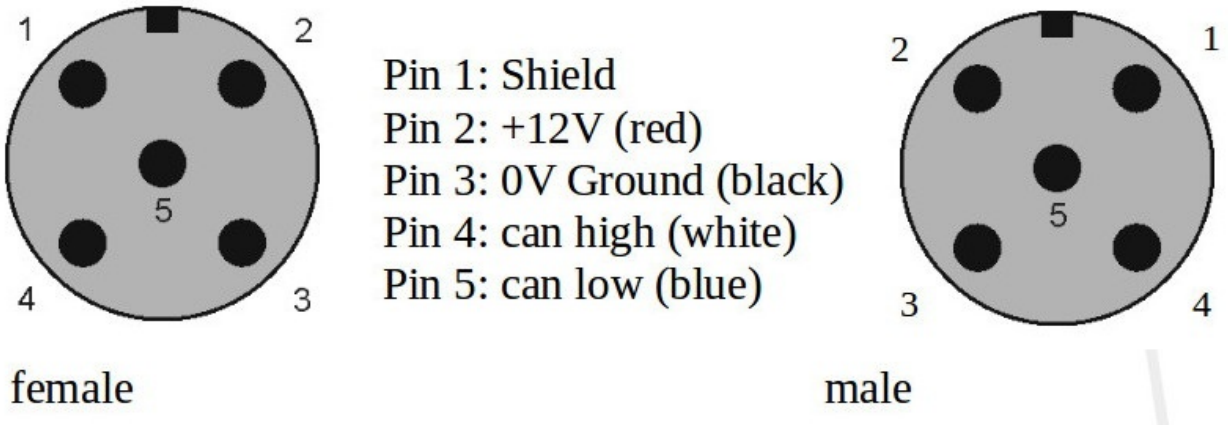


T-connector



Drop line M12 5 pins male connector side

Drop line wires side



- Pull out the green screw terminal of the stick.
- Connect the drop line blue wire from pin 5 (pin in the middle) to the green terminal on CANL.
- Connect the drop line white wire from pin 4 to the green terminal on CANH.
- Turn off the main power switch to be sure that there is no power on the network.
- Connect the drop line to the free T-connector on your backbone.
- Use a multimeter and measure the resistance between CANH and CANL (on the screws). The resistance should be around 60 Ohm.
- Connect the green screw terminal to the CAN-USB Stick.
- Check again the 60 Ohm between CANH and CANL.
- On the drop line there are three cables left. They have to be isolated.
- Turn on the main power.
- Switch on instrumentation.

To configure your CAN-USB Stick, see the chapter N2K. On Windows use OpenSkipper.

LED

The CAN-USB Stick LED will be **OFF** 10" during the boot secuence and then:

- Fixed **ON** if it is not connected to the network.
- **ON** for a second if it is connected to the network.
- Fixed **OFF** if there are not input data.
- **FLASHING** if there are input data.

Support

If you need support or you have any suggestion you can publish your questions on OpenMarine forum[8].

---

[1] https://commons.wikimedia.org/wiki/File:NMEA2000_Modified_motor_yacht.jpg [2] https://github.com/canboat/canboat [3] http://signalk.org [4] http://openskipper.org [5] http://macenc.com [6] http://www.polarnavy.com [7] https://en.wikipedia.org/wiki/NMEA_2000 [8] http://forum.openmarine.net/forumdisplay.php?fid=11 [9] http://shop.sailoog.com

USB DVB-T dongle

DVB-T dongles based on the Realtek **RTL2832U** chip can be used as cheap one channel AIS receptors. You can also receive the entire VHF marine band.

These devices are called Software Defined Radio (SDR) receivers.

SDR receivers will need more power than the Raspberry Pi USB port can provide. You need to plug the dongle into a self powered USB hub.

> This item is available in our store[4]

Antenna

The most important factor for good reception is the antenna. Any VHF antenna will work right. You can build some proficient homemade antennas[1][2][3].

Settings

Some SDR devices need calibration to receive AIS signal. See SDR receiver chapter to configure USB DVB-T dongles as AIS or marine radio receivers.

---

[1] http://www.radioforeveryone.com/p/ais-antennas.html [2] http://nmearouter.com/docs/ais/aerial.html [3] https://www.youtube.com/watch?v=SdEglNHyHB4 [4] http://shop.sailoog.com

IMU sensor

If you don't have a electronic compass on board you will need an IMU.

An Inertial Measurement Unit, or IMU, measures and reports on velocity, orientation and gravitational forces, using a combination of an accelerometer, gyroscope, and a magnetometer.

Connecting an IMU to OpenPlotter will provide magnetic heading which is needed to calculate true heading and true wind. You will have heel angle data as well.

> This item is available in our store[1]

Supported IMU sensors

- InvenSense MPU-9150 single chip IMU.
- InvenSense MPU-6050 plus HMC5883 magnetometer on MPU-6050's aux bus (handled by the MPU-9150 driver).
- InvenSense MPU-6050 gyros + acclerometers. Treated as MPU-9150 without magnetometers.
- InvenSense MPU-9255 single chip IMU (I2C and SPI).
- STM LSM9DS0 single chip IMU.
- STM LSM9DS1 single chip IMU.
- L3GD20H + LSM303D (optionally with the LPS25H) as used on the Pololu AltIMU-10 v4.
- L3GD20 + LSM303DLHC as used on the Adafruit 9-dof (older version with GD20 gyro) IMU.
- L3GD20H + LSM303DLHC (optionally with BMP180) as used on the new Adafruit 10-dof IMU.
- Bosch BMX055 (although magnetometer support is experimental currently).
- Bosch BNO055 IMU with onchip fusion. Note: will not work reliably with RaspberryPi/Pi2 due to clock-stretching issues.

Wiring

IMU sensors have to be connected by I2C interface. See chapter Wiring I2C sensors.

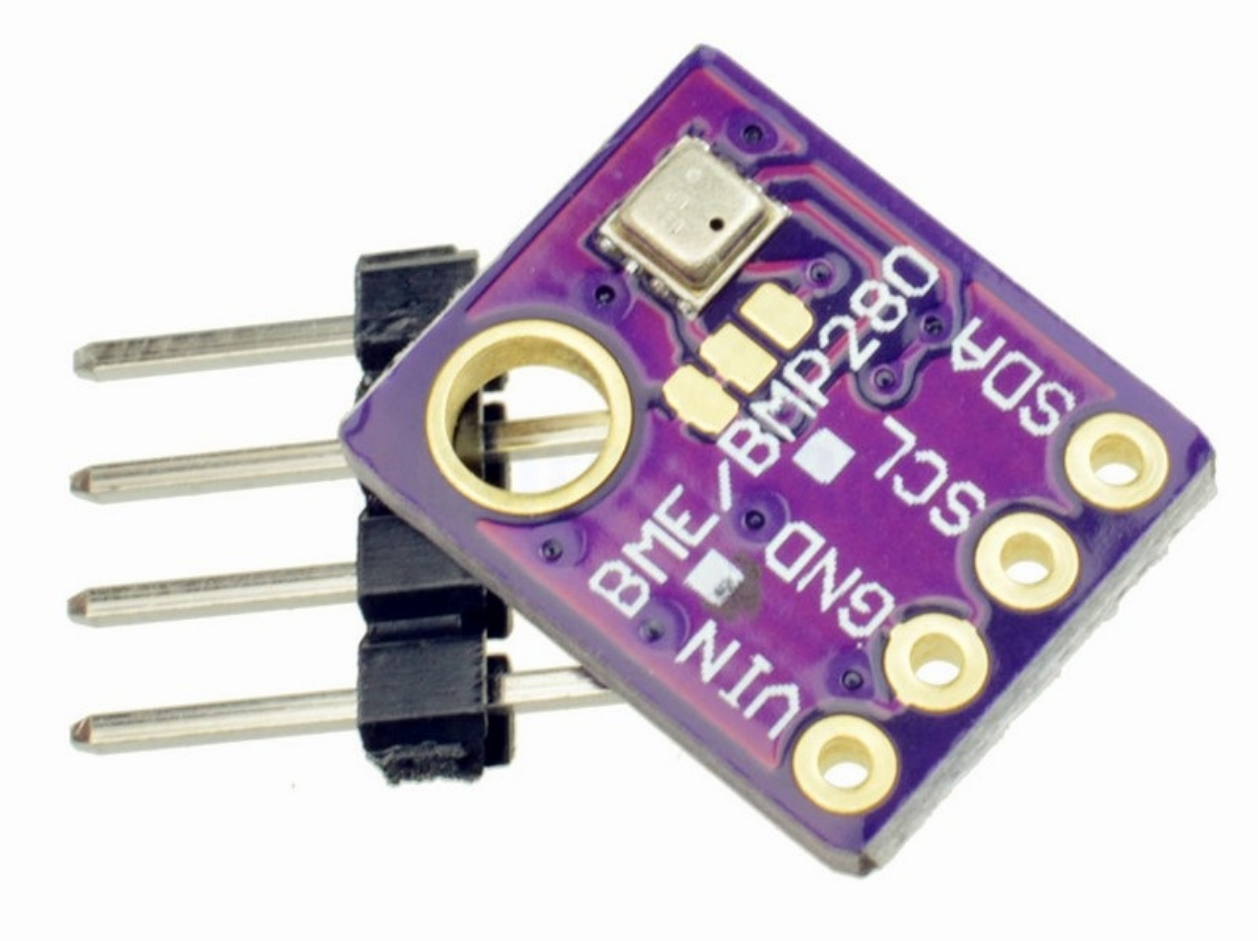
Settings

See Compass chapter to configure IMU sensors.

---

[1] http://shop.sailoog.com

Environment sensors



Connecting a little environment sensor to OpenPlotter will provide air pressure, temperature or humidity data to monitor weather.

> This item is available in our store[1]

Supported environment sensors

- BME280 (pressure, temperature, humidity)
- BMP180 (pressure, temperature)
- LPS25H (pressure, temperature)
- MS5611 (pressure, temperature)
- MS5637 (pressure, temperature)
- HTS221 (humidity, temperature)
- HTU21D (humidity, temperature)

Wiring

Environment sensors have to be connected by I2C interface. See chapter Wiring I2C sensors.

---

[1] http://shop.sailoog.com

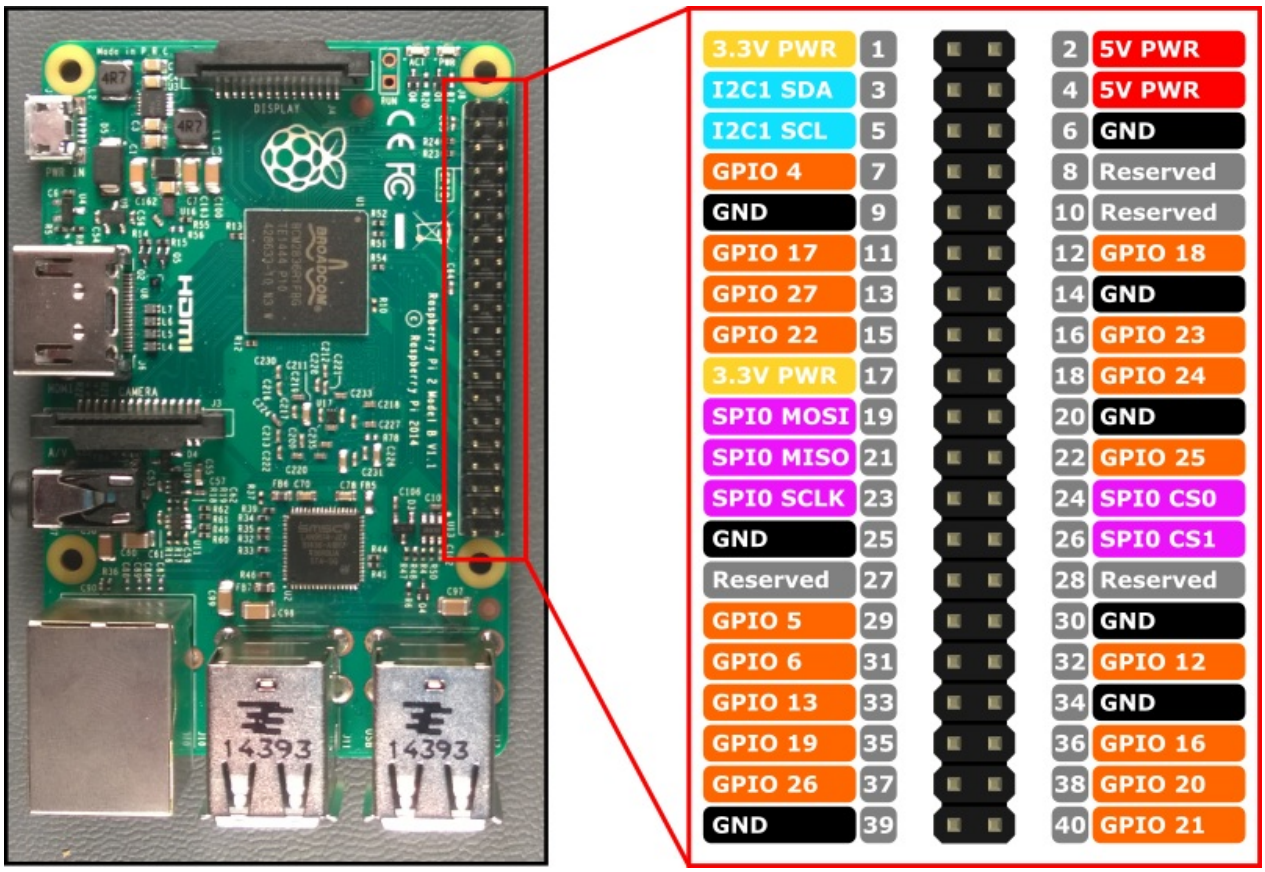
1W temperature sensor

You can connect one wire (1W) **DS18B20** sensors to OpenPlotter. This sensor is waterproof and can withstand high temperatures. Connecting multiple DS18B20 in parallel to the same pins, you will be able to get temperature data from coolant engine, exhaust, engine room, fridge, sea ...

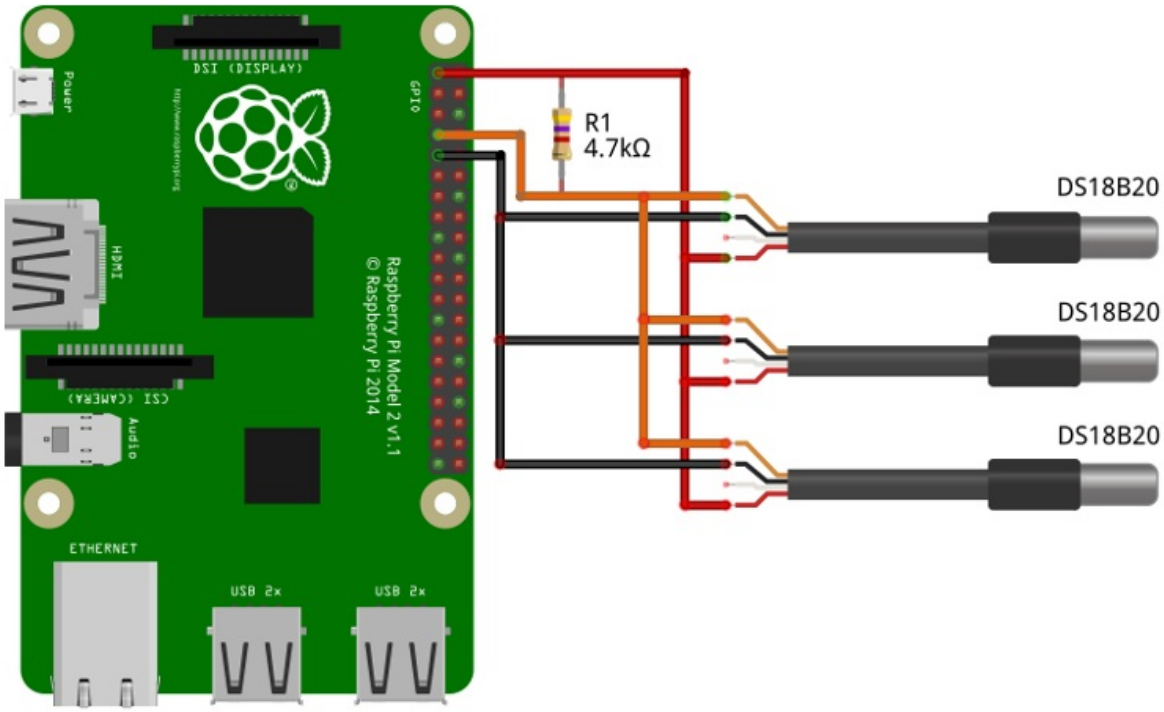
> This item is available in our store[1]

Wiring

Pins names are according to the diagram below.



You have to connect these sensors to **GPIO4** (aka GCLK), **GND** and **3.3V** pins. Some sensors may have a fourth wire which do not need to be connected. You need to use a pull-up resistor as shown in the image below. You can connect multiple sensors in paralel using just one resistor.



If you want to change the GPIO pin, edit the file config.txt typing in a terminal:

```
sudo nano /boot/config.txt
```

At the end of the file you should see a line like this

dtoverlay=w1-gpio

replace it by

dtoverlay=w1-gpio,gpiopin=x

where x is your desired GPIO pin. Save and reset.

Settings

See 1W chapter to configure DS18B20 sensors.

---

[1] http://shop.sailoog.com

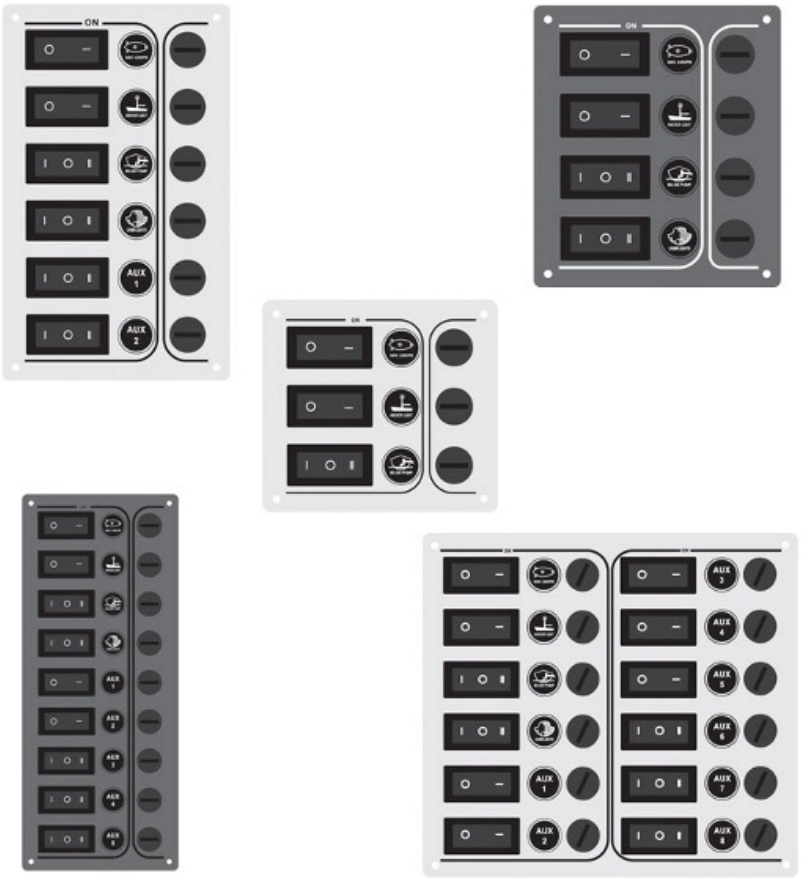
PIR motion sensor

Connecting this sensor you can detect if a human has moved in or out of the sensor range. They are small, inexpensive, low-power and easy to use. It is basically made of a pyroelectric sensor which can detect levels of infrared radiation. Everything emits some low level radiation and the hotter something is, the more radiation is emitted. If you connect OpenPlotter to internet you will be aware of what is happening in your boat when you are not there.

> This item is available in our store[1]

[1] http://shop.sailoog.com

Common switch



Almost all of the OpenPlotter features have a defined action. You can connect any type of external switch (momentary, toggle...) and assign an action to each state (on/off). So you can shutdown/reset the system, run a custom Linux command, play/stop services, trigger external devices, play/stop an audio ...

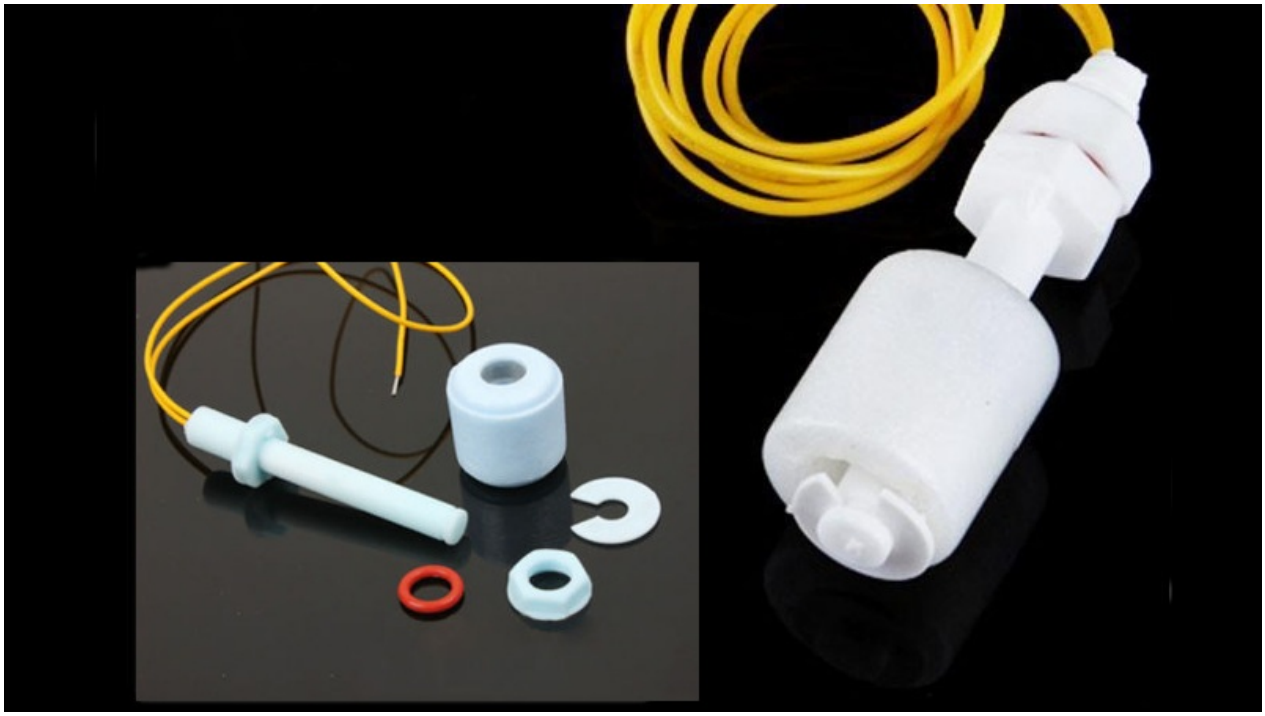
Door switch

Installing this little magnetic switches on doors and windows you will be able to detect any unauthorized opening. If you connect OpenPlotter to internet you will be aware of what is happening in your boat when you are not there.

> This item is available in our store[1]

[1] http://shop.sailoog.com

Float switch



Installing this little magnetic switches on tanks and bilges you will be able to detect when they are full or empty. OpenPlotter may actuate a pump, an indicator, an alarm or warn you by Twitter or email.

> This item is available in our store[1]

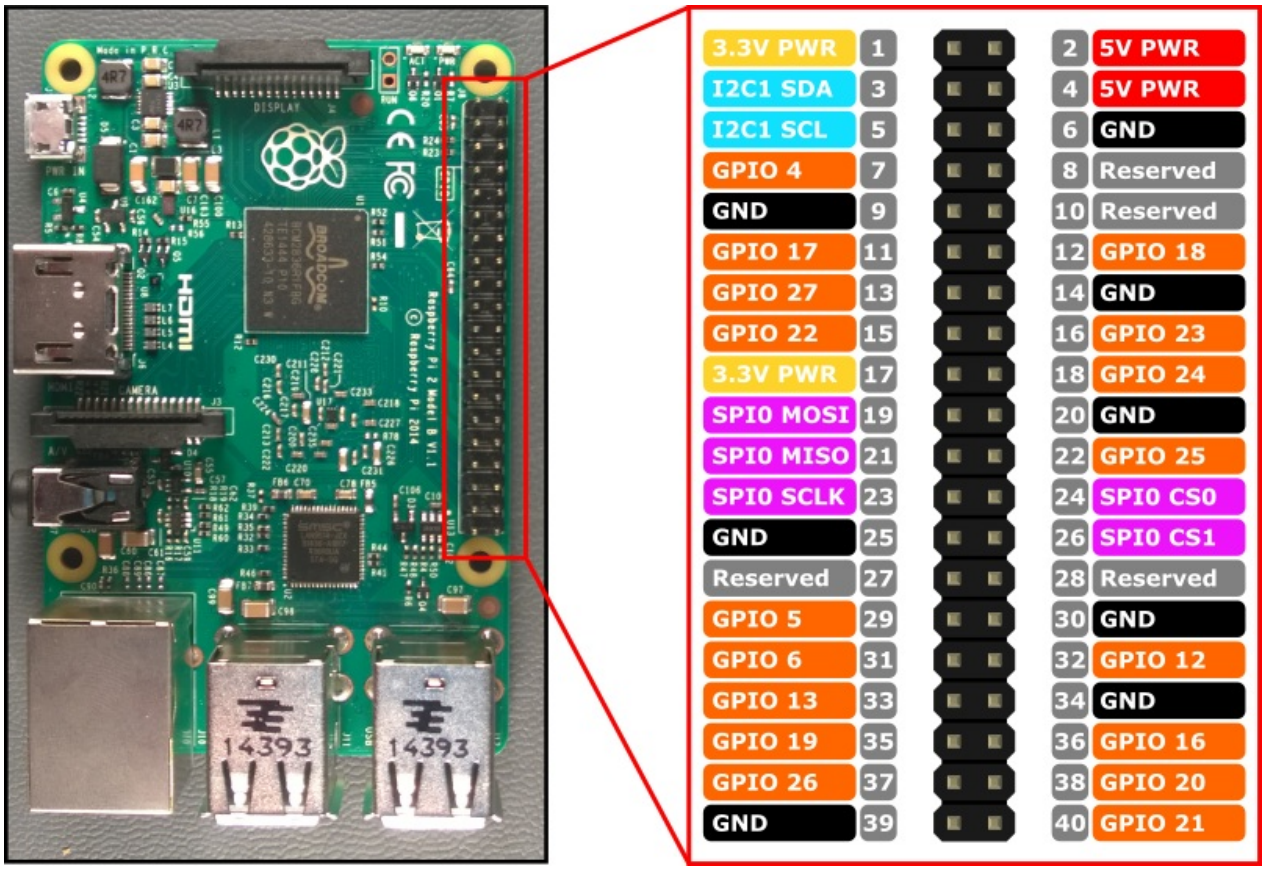
[1] http://shop.sailoog.com

Relay

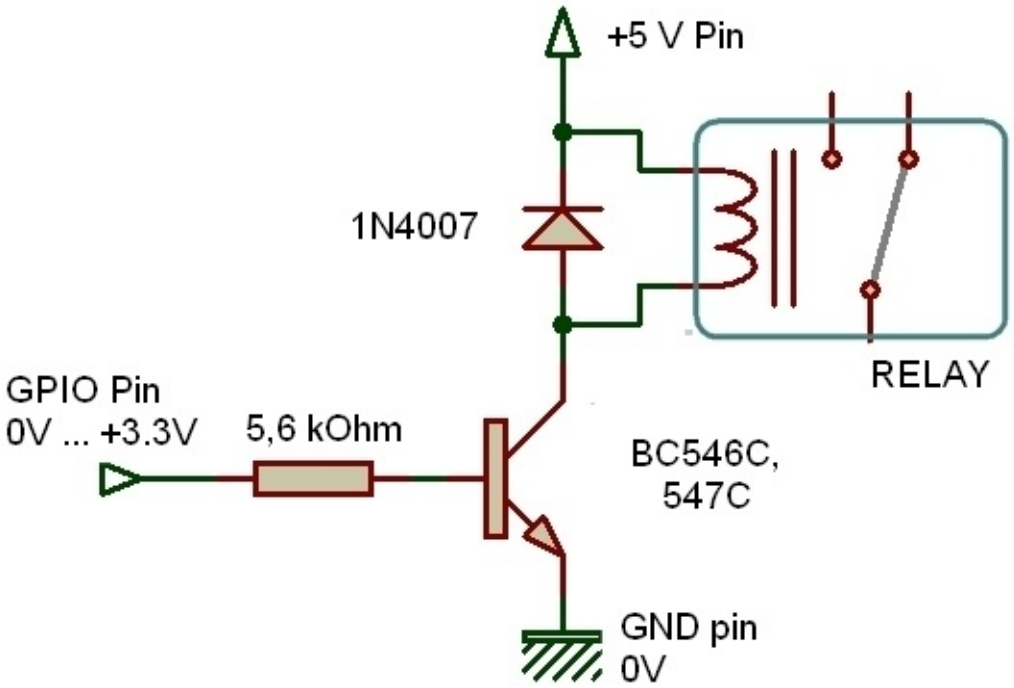
This chapter is under construction

Wiring

Pins names are according to the diagram below.



The GPIO pins can only deliver a few mA, and cannot drive directly a relay. Therefore a few electronics components are required :



- The NPN transistor BC54xC acts as a valve to control the current through the relay coil (switching mode). It is either passing when GPIO is HIGH, or blocked when GPIO is LOW. Its maximum collector current is 100 mA. The final "C" ensures than its high gain (>400) will minimize the current drawn from the GPIO.
- The 5,6 kOhm resistor limits the current drawn from the GPIO pin below 0,5 mA ;
- The diode, 1N4007 prevents damaging the transistor (and/or GPIO pins !) when the relay is commuted.
- The relay is a 5VDC relay, with an internal resistance of 60 Ohm. It requires less than 90mA, compatible with the transistor. (e.g. printed circuit relay Omron G5RL-1-E-HR 5 VDC 5 V/DC 16 A). It is available from Conrad on line shops.

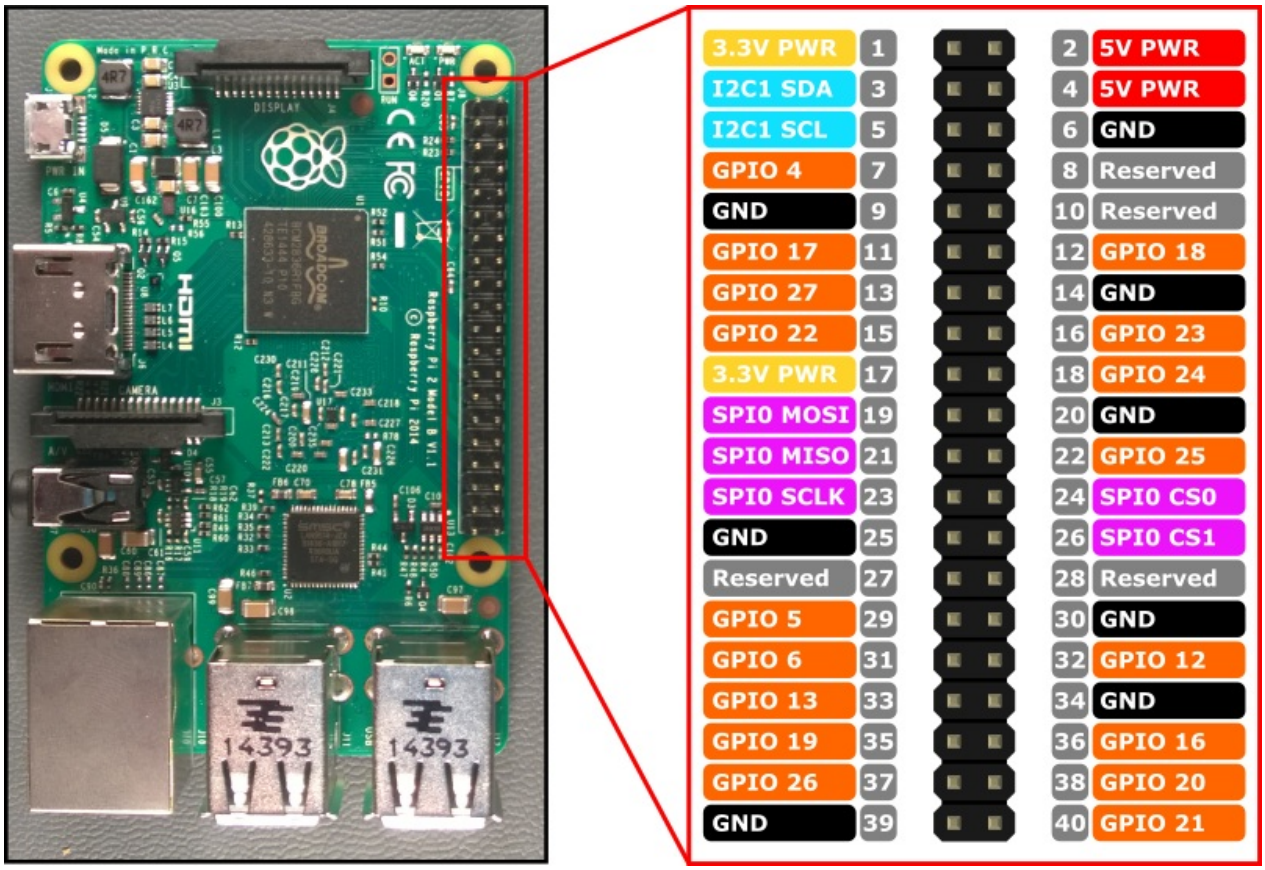
Relays used in automobile industry should be connected to + 12 VDC. They will draw a higher current (4 to 500 mA), and call for a more elaborated design (cascading two transistors).

LED

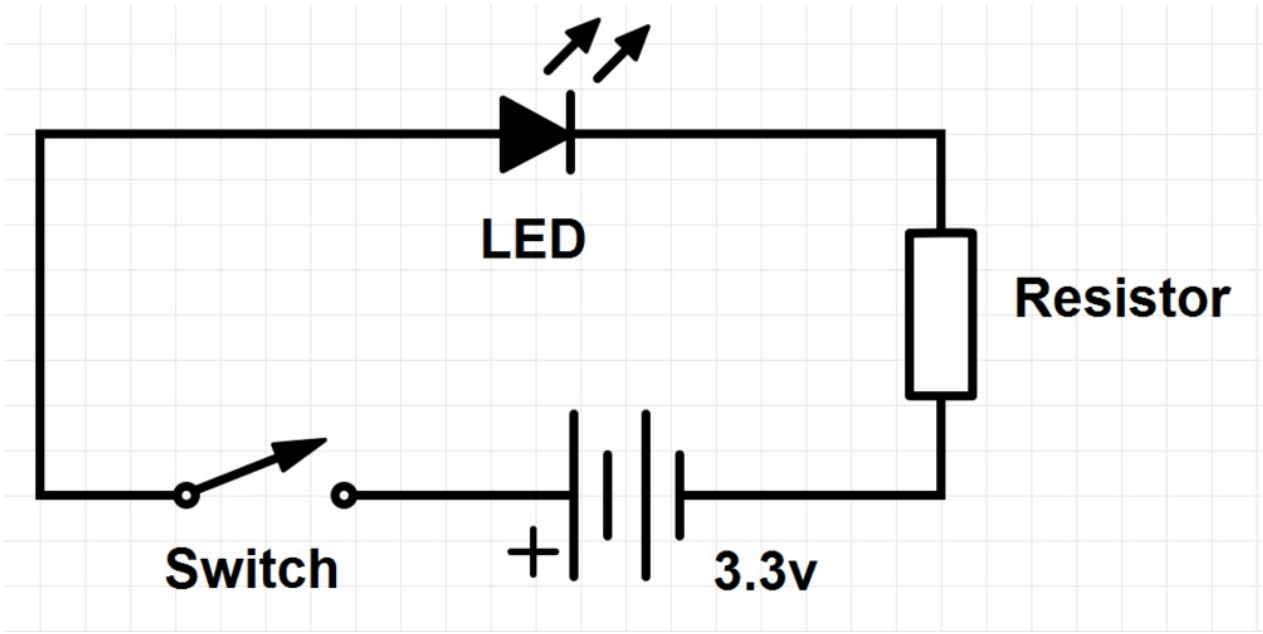
This chapter is under construction

Wiring

Pins names are according to the diagram below.

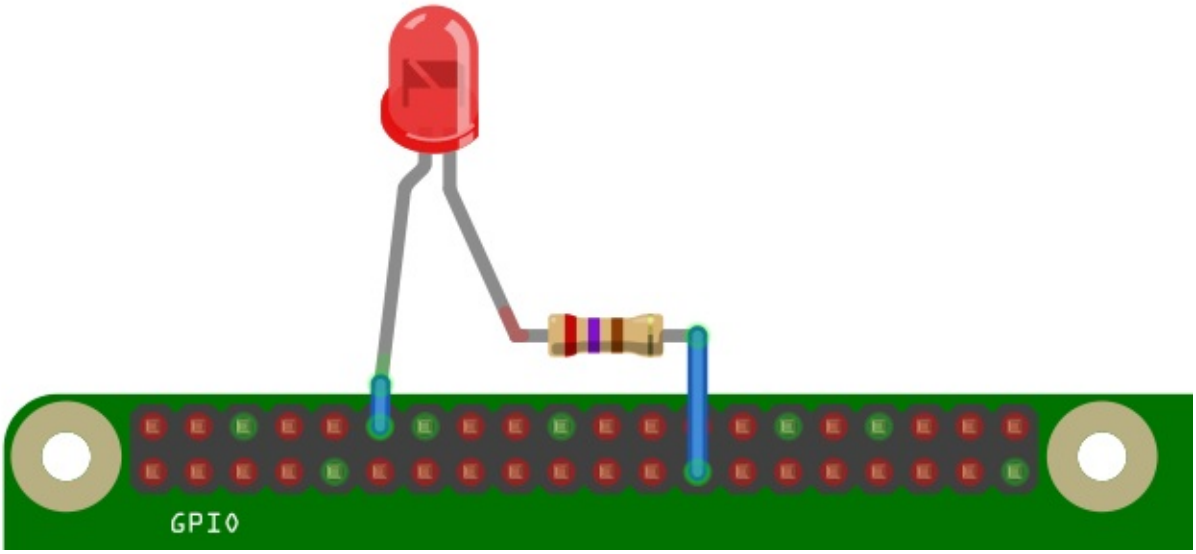


Ignoring the Pi for a moment, one of the simplest electrical circuits that you can build is a battery connected to a light source and a switch (the resistor is there to (limit the current flow) and protect the LED):



When we use a GPIO pin as an output, the Raspberry Pi replaces both the switch and the battery in the above diagram. Each pin can turn on or off,or go **HIGH** or **LOW** in computing terms. When the pin is **HIGH** it outputs **3.3 volts** (3v3); when the pin is LOW it is off.

Here's the same circuit using the Raspberry Pi. The LED is connected to a GPIO pin (which can output +3v3) and a ground pin which is 0v and acts like the negative terminal of the battery : (the resistor is there to limit the current flow and protect both the LED and the Raspberry CPU). A value of 1 kOhm will retrain the current around 1.5 mA



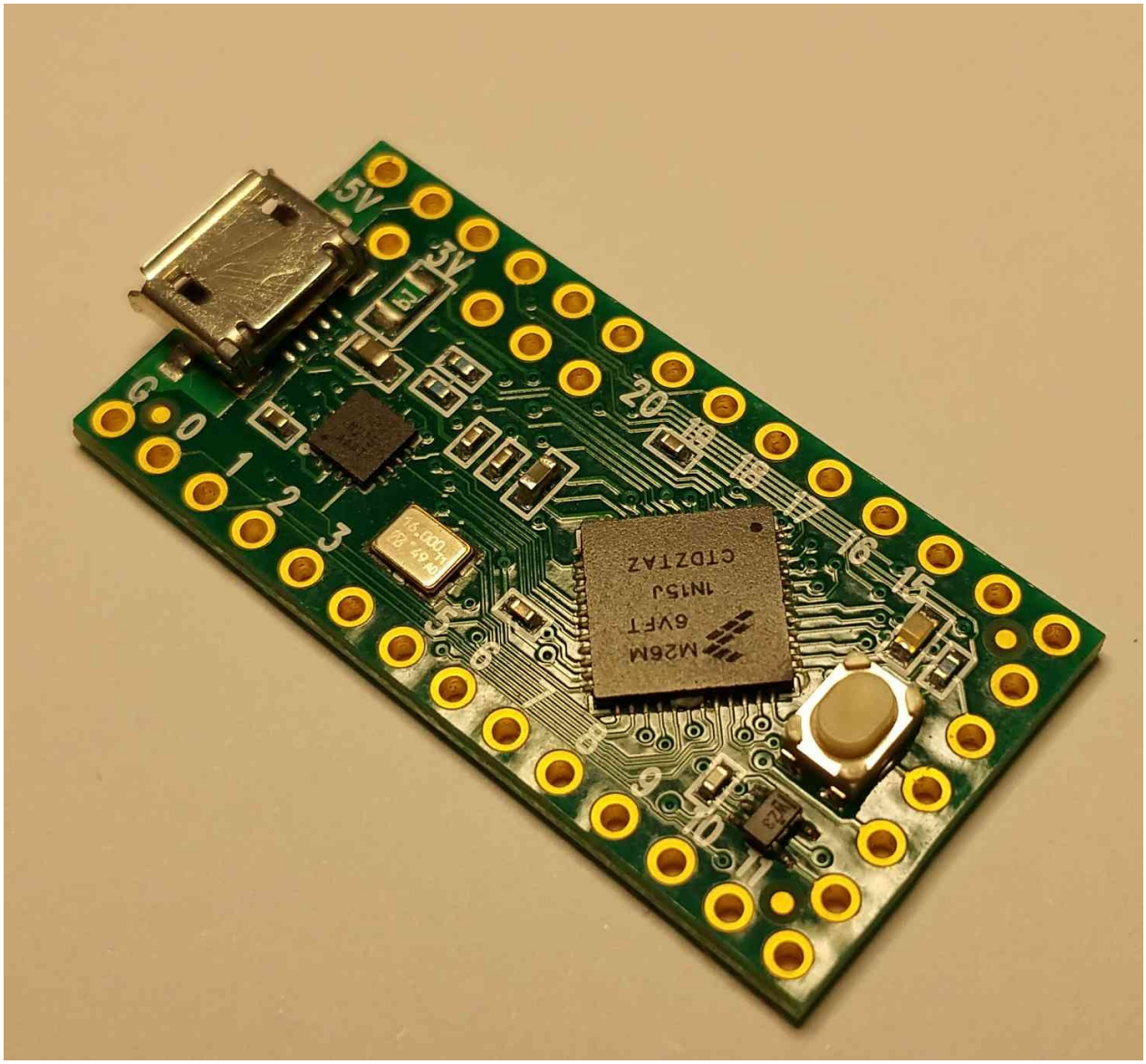
Buzzer

This chapter is under construction

Arduino

This chapter needs to be written/updated/translated

http://forum.openmarine.net/forumdisplay.php?fid=16



Picture: Teensy LC.

Arduino is a board with a MCU. Most of Arduinos and compatible have a MCU with build in adc. They have 8-16 bit adc.

The number of analog inputs channel depend on the MCU.

Connected to USB serialport.

In Openplotter we use the firmata script of Arduino IDE.

It communicates with the RPI over firmata protocol.

advantage:

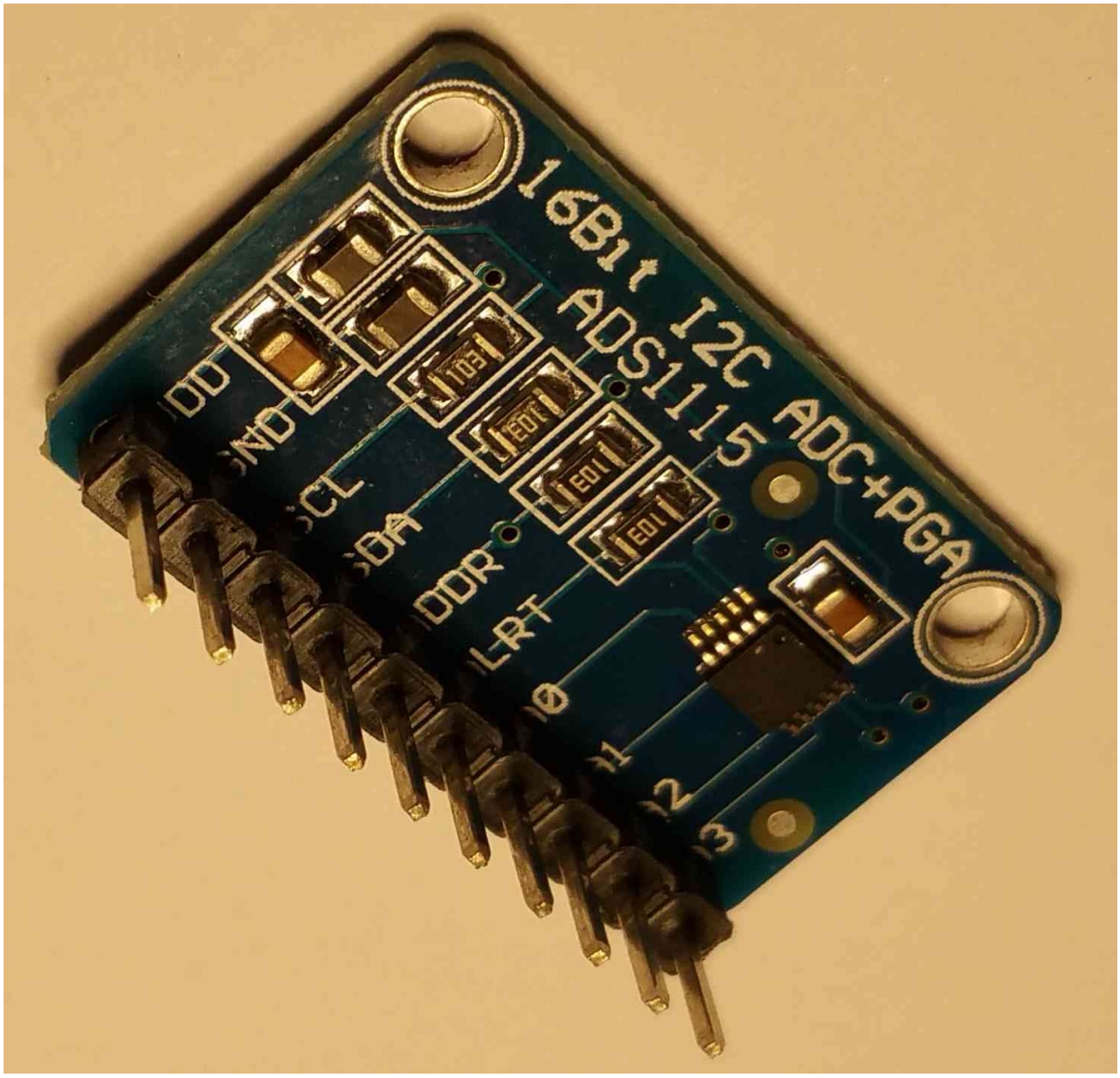
- can be isolated (by USB to USB isolator, or uart to USB isolator)

See Analog Firmata

ADS1115

This chapter needs to be written/updated/translated

http://forum.openmarine.net/forumdisplay.php?fid=16



It is a 16 bit adc with 4 analog inputs

Connected to gpio i2c

See Analog ads1115

ESP8266

The boards with the Espressif mcu are popular because they have an integrated wifi.

They can be programed with the Arduino ide to directly send signalk sentences to openplotter.

nrf24L01

Is a tranceiver module which works at 2.4 GHz.

There are some products which use this chip. For example a power monitor. The values can be received by openplotter.

Wireless thermometer 433MHz

House used wetherstations are often combined with an outdoor sensor. These sensors use often 433 MHz.

With the RTL2838 dongle and the rtl_433 library many different protocols can be translated.

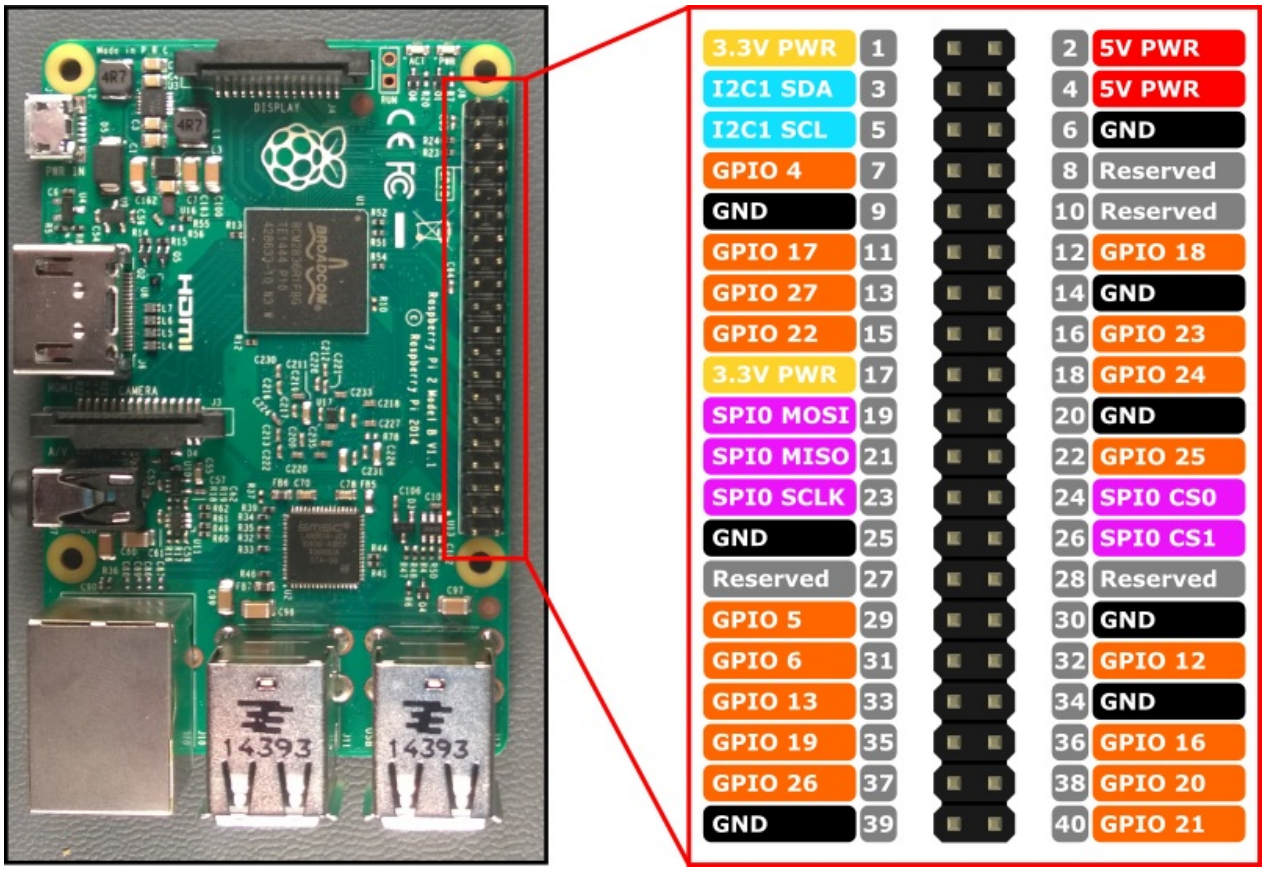
So its easy to get for example the temperature of the fridge into openplotter.

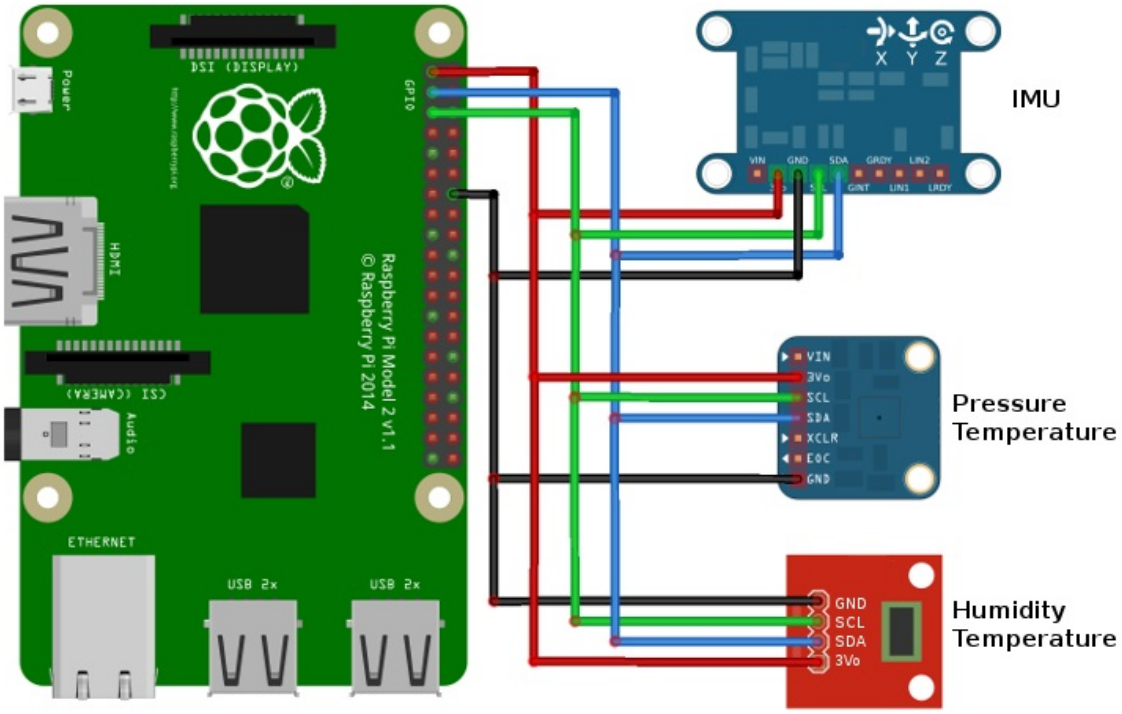
Wiring I2C sesnsors

This chapter needs to be written/updated/translated

http://forum.openmarine.net/forumdisplay.php?fid=16

Pins names are according to the diagram below.




DSI (DISPLAY)
Power
GPIO
HDMI
(CAMERA) CSI
Audio
ETHERNET
USB 2x
USB 2x
Raspberry Pi Model 2 v1.1
© Raspberry Pi 2014
X Y Z
IMU
VIN
GND
SDA
GRDY
LIN2
GINT
LIN1
LRDY
VIN
3Vo
SCL
SDA
XCLR
EOC
GND
Pressure
Temperature
GND
SCL
SDA
3Vo
Humidity
Temperature

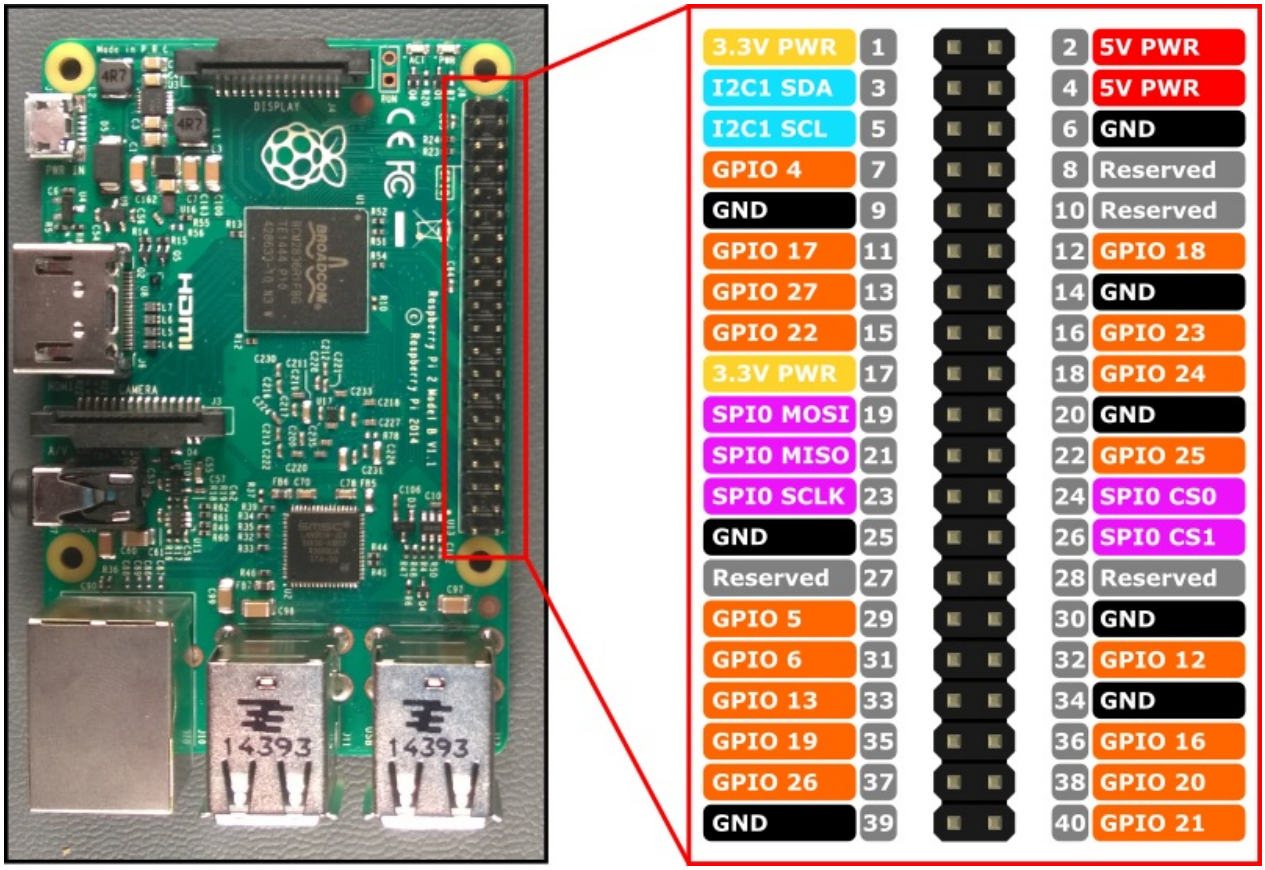
Wiring digital sensors

This chapter needs to be written/updated/translated

http://forum.openmarine.net/forumdisplay.php?fid=16

Digital sensors are driven through the **GPIO** (General Purpose Input/Output) pins. These pins are copper connected to the CPU chip. A short circuit with 0 V or +3.3 V is not final, but any direct contact with the + 5 V (or higher) can KILL the Raspberry Pi.

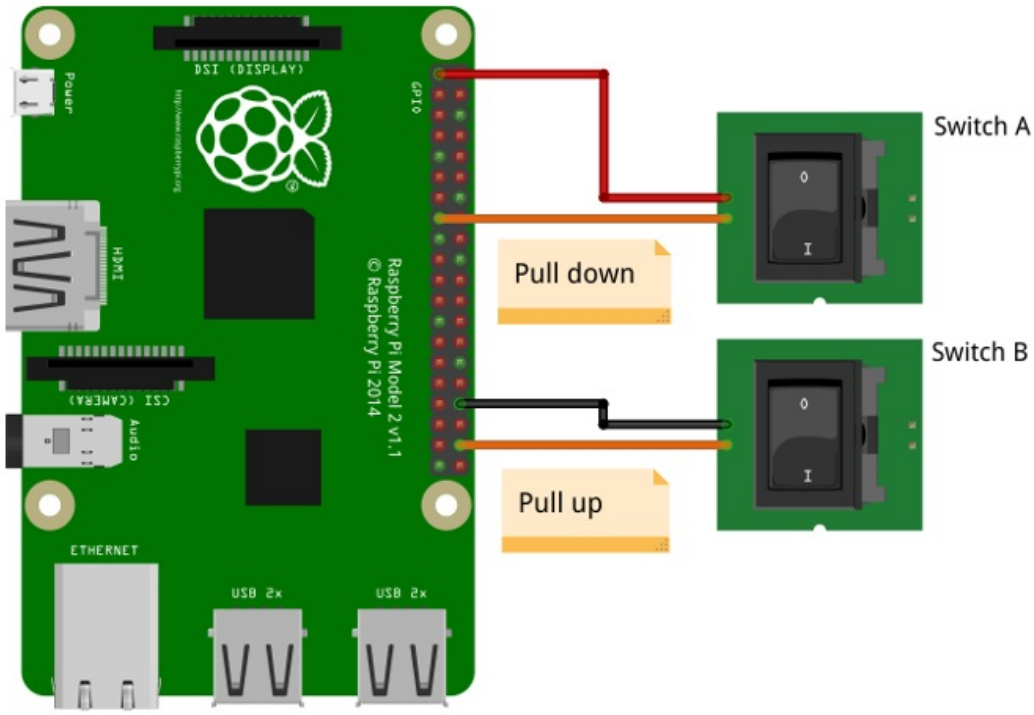
Pins names are according to the diagram below.



For a normal digital sensor (open by default), you have to configure it as "Pull down" and to connect it between selected GPIO pin and +3.3v pin (DANGER, NEVER TO +5v).

For a special digital sensor (closed by default), you have to configure it as "Pull up" and connect it between selected GPIO pin and GND pin.

It is not a problem if you make a mistake connecting to GND or +3.3v but be careful and avoid the +5v pin. It isn't a bad idea to add a 1 kOhm resitor into the circuit.



Available pins for digital sensors

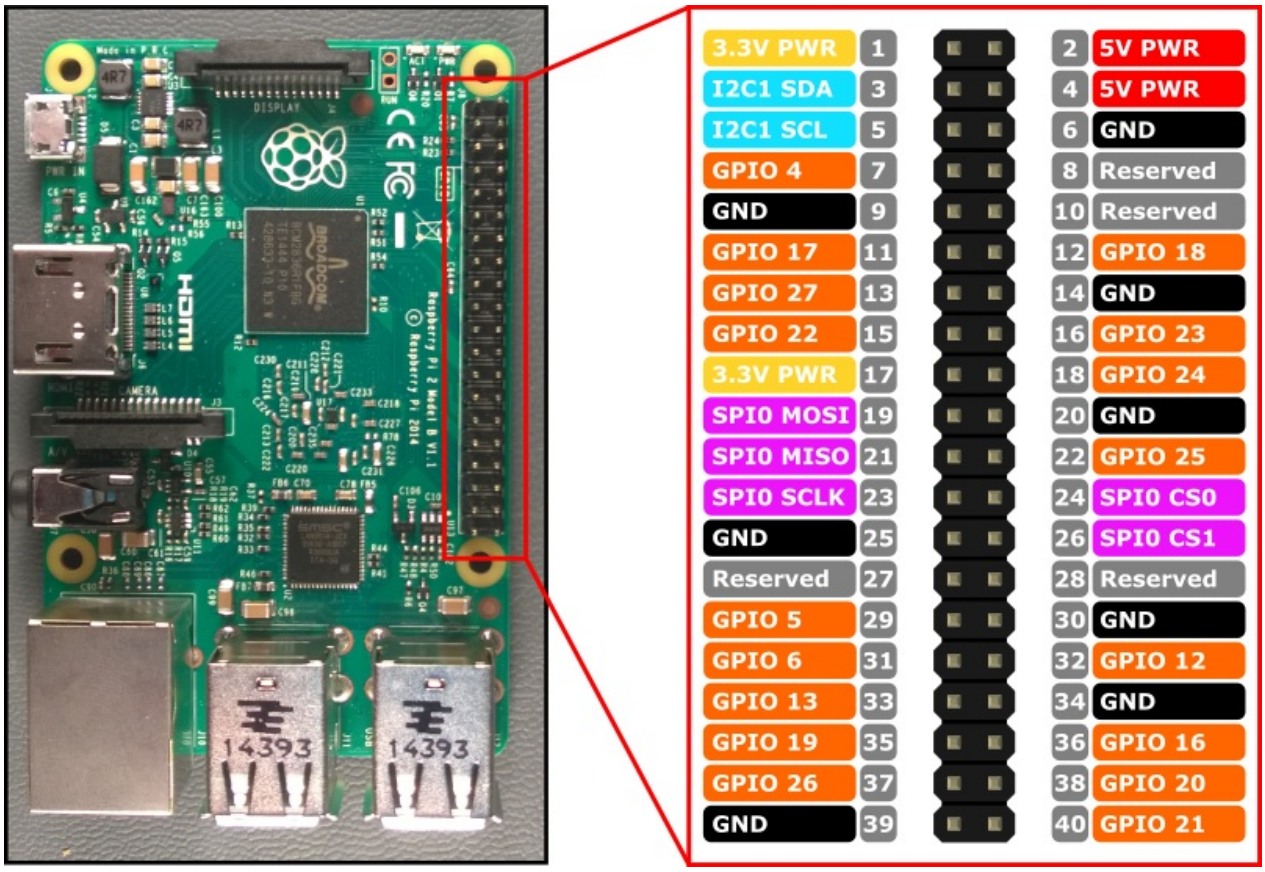
Powered digital sensors

Wiring output devices

This chapter needs to be written/updated/translated

http://forum.openmarine.net/forumdisplay.php?fid=16

Pins names are according to the diagram below.



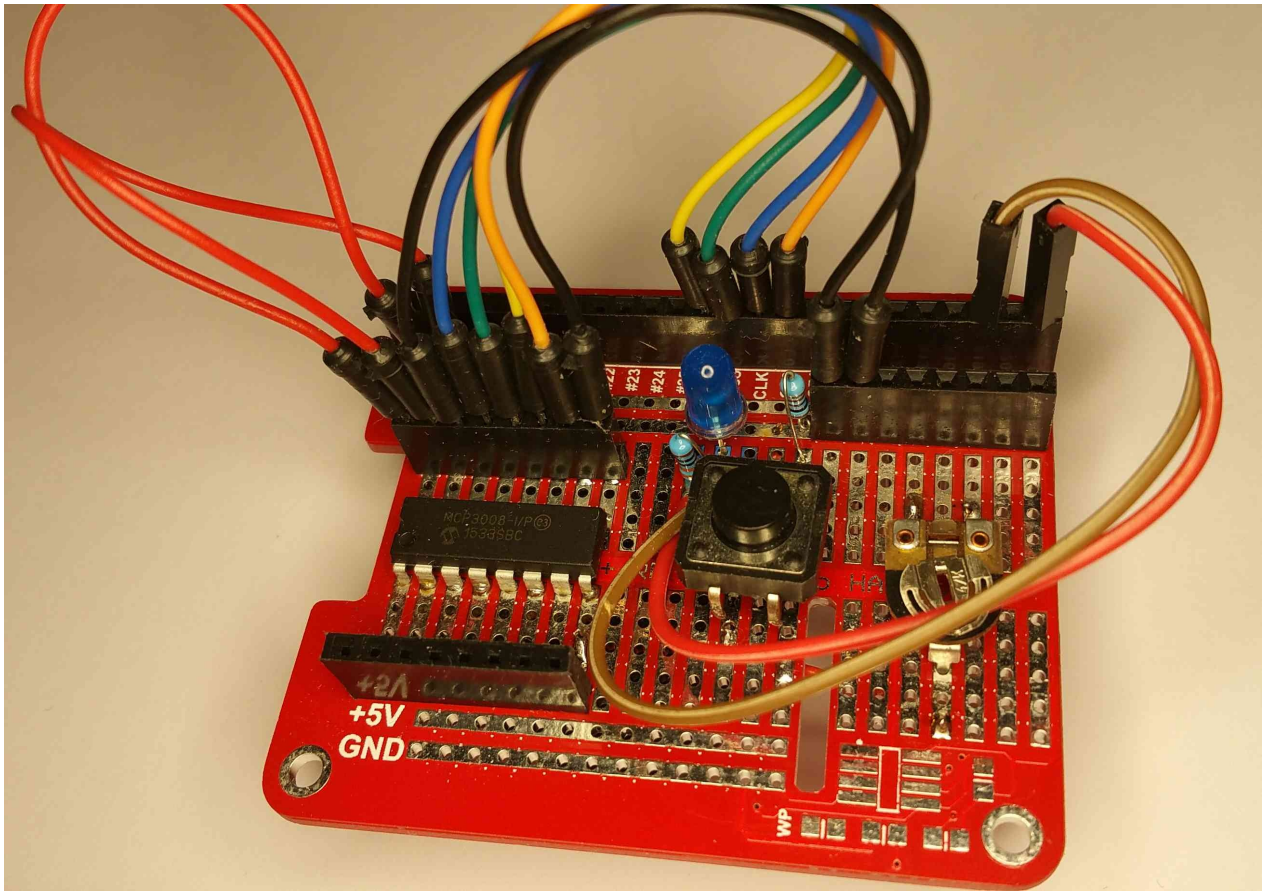
Available pins for digital sensors

Wiring SPI sensors

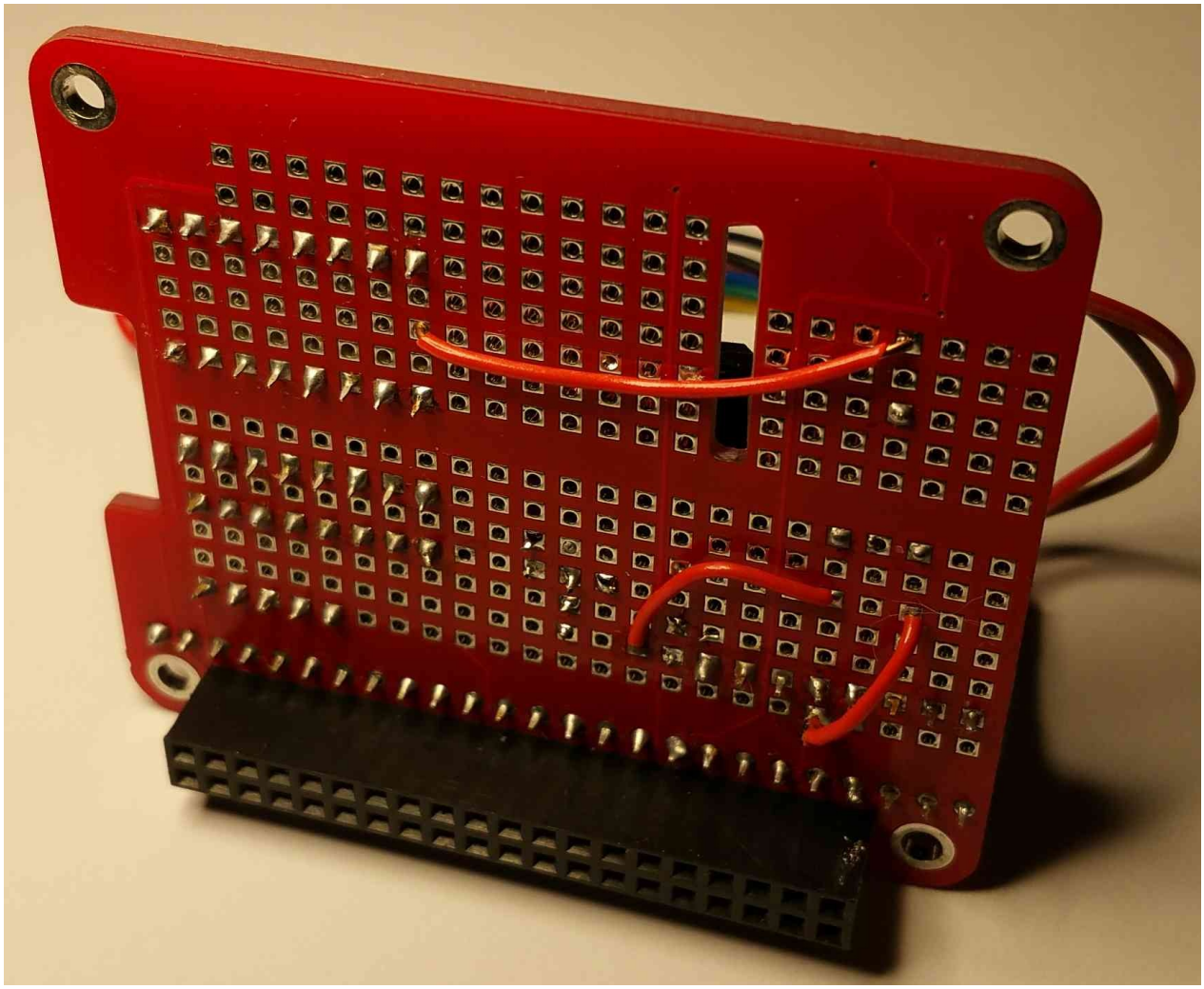
This chapter needs to be written/updated/translated

http://forum.openmarine.net/forumdisplay.php?fid=16

MCP3008



DIY development board front



DIY development board back

Getting started

First of all you have to put together at least all the required hardware parts.

Then, you have to run the software on your Raspberry. **OpenPlotter RPI** is a modified version of Raspbian, the official operating system for the Raspberry Pi. It contains all you need. OpenPlotter RPI is open-source and free.

Here you have two options, either buy our plug and play SD card or download and install the software on an SD card.

8GB/16GB SD card with OpenPlotter RPI ready to run.
(waiting for v0.9.xbeta to be released)
shop.sailoog.com.

Installing OpenPlotter RPI on an SD card

Any micro-SD-compatible card will work on your Raspberry. However, there are some guidelines that should be followed.

A minimum of 8GB is required but 16GB is recommended.

The card class determines the sustained write speed for the card; a class 4 card will be able to write at 4MB/s, whereas a class 10 should be able to attain 10 MB/s. However it should be noted that this does not mean a class 10 card will outperform a class 4 card for general usage, because often this write speed is achieved at the cost of read speed and increased seek times.

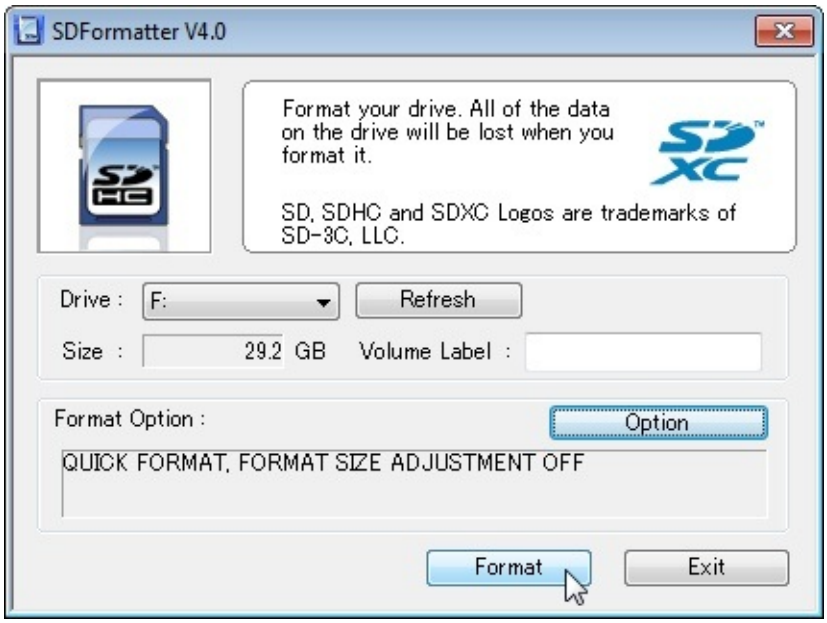
To begin with, it's always a good idea to make sure you have formatted your SD card. You'll need to make sure your computer has a built-in SD card reader, or you can use a USB SD card reader.

Visit the SD Association's website and download [SD Formatter 4.0] (https://www.sdcard.org/downloads/formatter_4/index.html) for either Windows or Mac.

Follow the instructions to install the software.

Insert your SD card into the computer or laptop's SD card reader and make a note of the drive letter allocated to it, e.g. F:/.

In SD Formatter, select the drive letter for your SD card and format it.



Download the latest NOOBS installer version of **OpenPlotter RPI** from

www.sailoog.com/en/blog-categories/openplotter-rpi

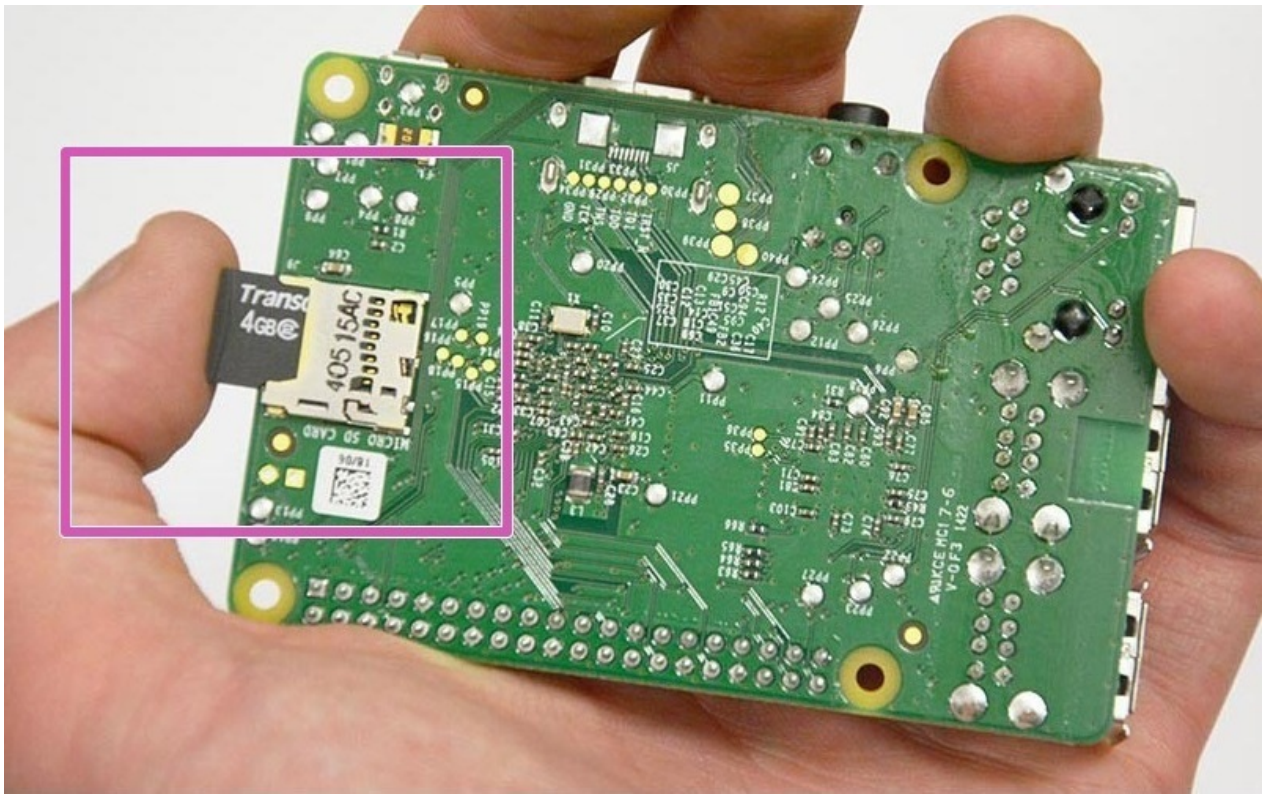
It is a compressed file of about 1GB so it will take a while.

Extract the files from the zip.

Once your SD card has been formatted, drag all the files in the extracted NOOBS folder and drop them onto the SD card drive.

The necessary files will then be transferred to your SD card.

When this process has finished, safely remove the SD card and insert it into your Raspberry Pi.



First boot

Connect power to the Raspberry Pi.

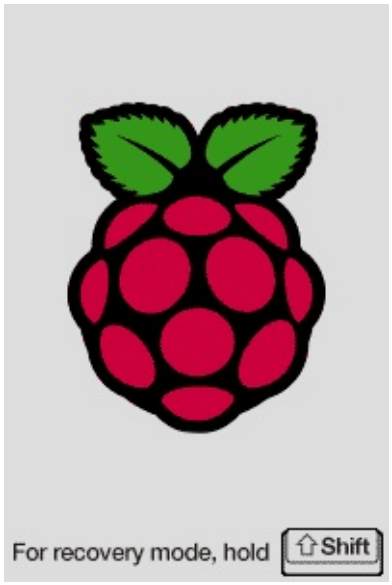
OpenPlotter NOOBS installer will make a silent install, this means that you do not have to do anything. It will take several minutes to format partitions and install the system.

If you are on an headless system you have to wait until a WiFi network named "OpenPlotter" apears on your client computer. The default password is "12345678", please change this as soon as possible on WiFi AP tab. For more information see Headless chapter.

Once the OpenPlotter NOOBS installer has installed OpenPlotter RPI, it will start directly every time we connect the Raspberry Pi.

Recovery system

If our system gets damaged or unstable, we can install OpenPlotter RPI again pressing the Shift key when we see this symbol at startup:



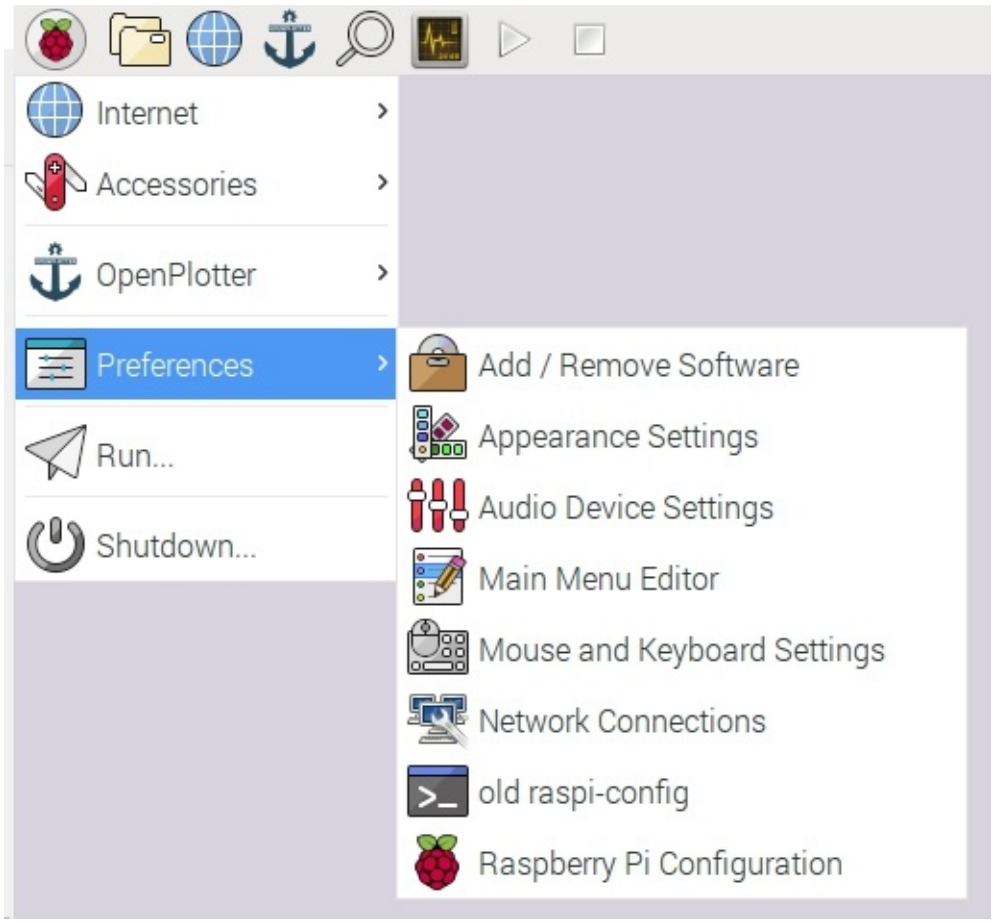
We will lose all data, manually installed programs and settings after installation.

Settings

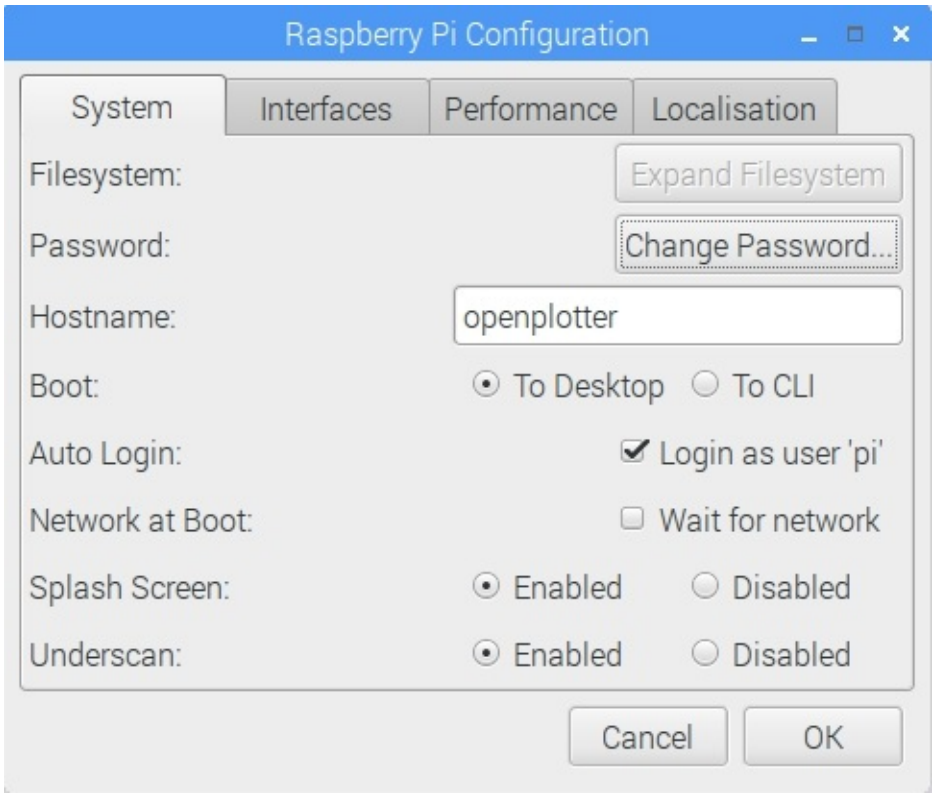
The nativ monitor resolution for 800x480 monitors will be auto detected. The right settings for it will work on the next boot! If we have such a monitor. We do a restart.

Auto Setup window will pop up to help us to configure the USB connected devices. We can skip or close it. It will appear on every boot until we uncheck setup autostart on every boot. For more information see Auto Setup USB Ports chapter.

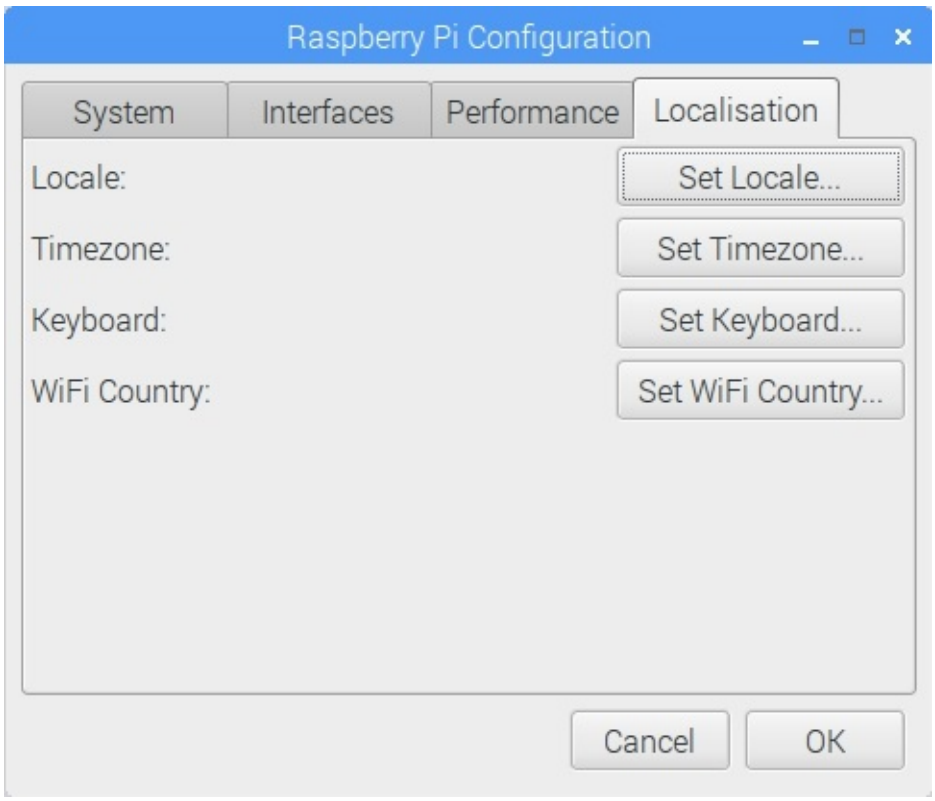
Go to Menu > Preferences and select *Raspberry Pi Configuration*.



A window will open where you can personalise your system. It is a good idea to change the Password to make OpenPlotter more secure. Click on *Change Password* (default password: raspberry).



If you need to set your system localisation, click on the *Localisation* tab and then on *Set Locale* (language), *Set Timezone, Set Keyboard, Set WiFi Country* buttons.



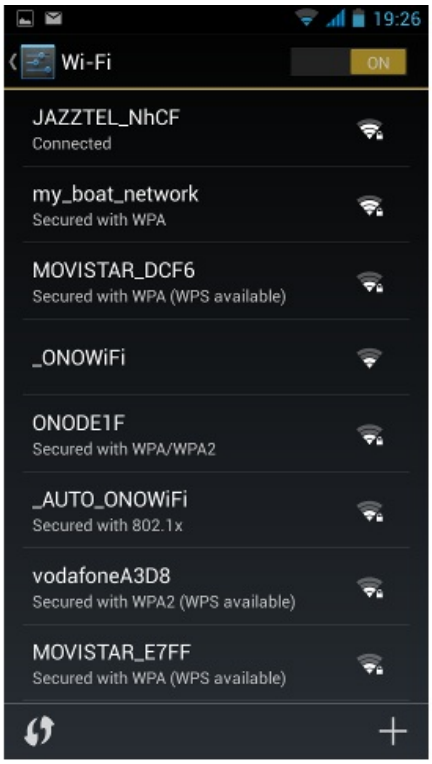
Headless

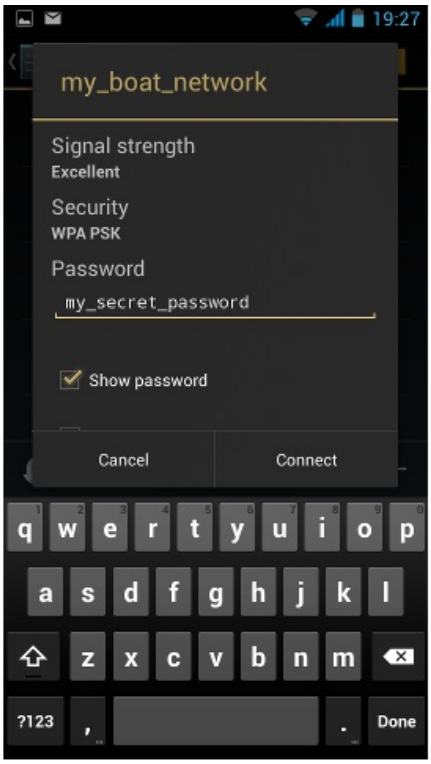
You can use the RPI in headless mode after the NOOBS installation is done.

There are three ways to work headless.

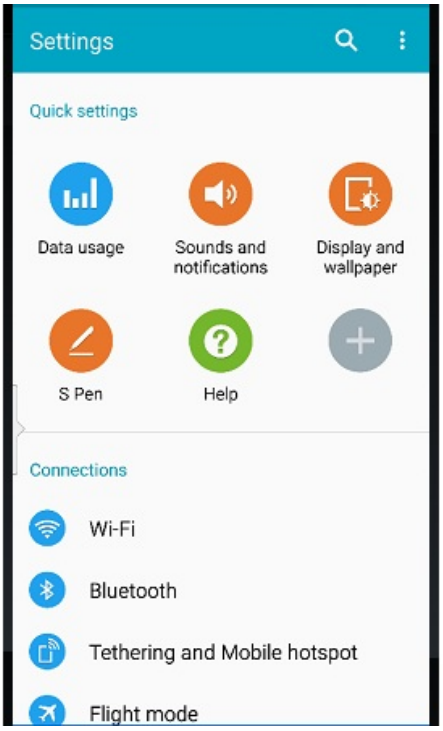
1. Over a WiFi Access Point. See WiFi AP chapter.
2. You can connect to the RPI by an Ethernet cable and use the bonjour protocol or the bridged hotspot.
3. You can connect to the RPI by an USB cable to your tablet or smartphone and use the USB tethering feature.

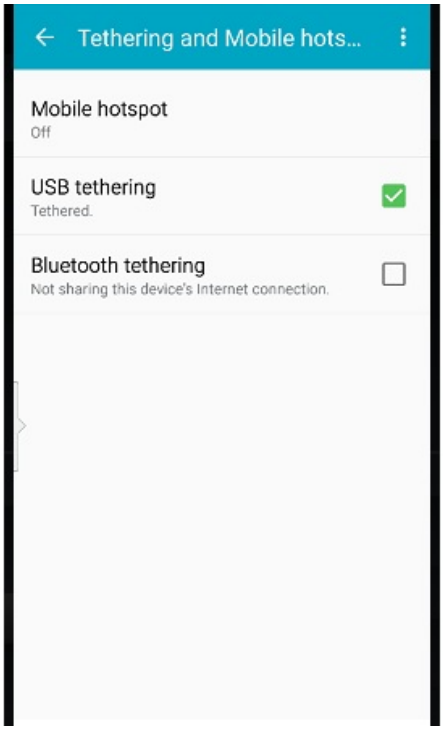
to 1) After it has finished booting up, there should be a new WiFi network called *OpenPlotter* available. Log onto this network using the password *my_secret_password*.





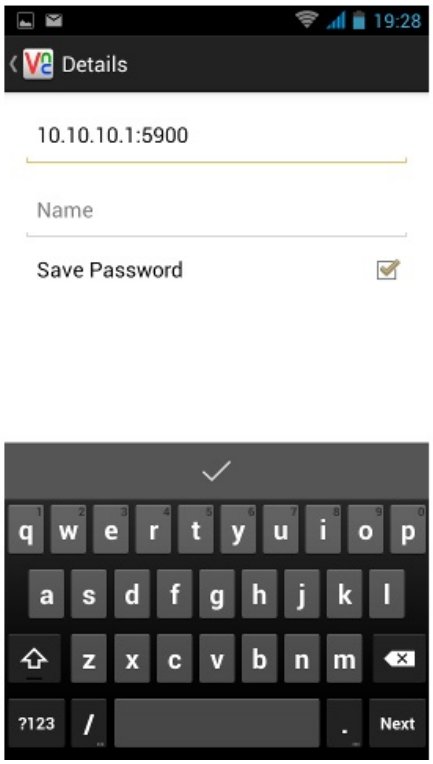
to 3) Selecting USB tethering

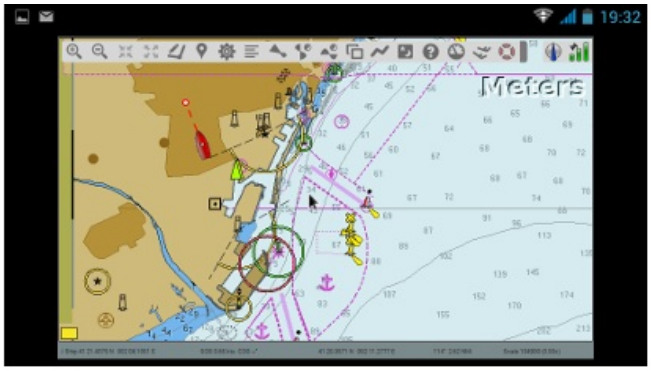


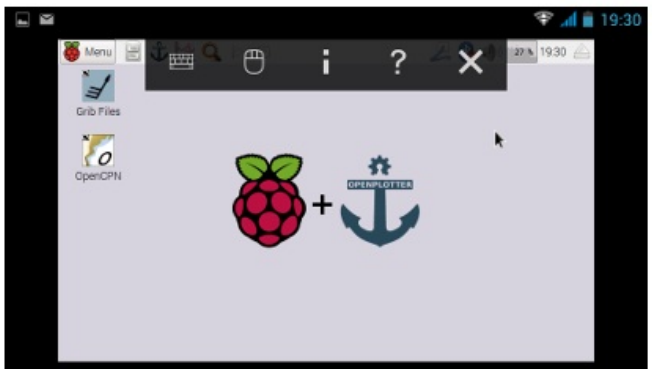


Finally, open your favourite VNC remote desktop client on your laptop, tablet or smartphone and make a new connection with the address/port combination: to 1) and 2) **10.10.10.1:5900**, or just the address **10.10.10.1** if you are using a RDP remote desktop client. to 3) **192.168.42.100:5900**., or just the address **192.168.42.100** if you are using a RDP remote desktop client.

Connect and enjoy!







See more information about tested remote desktop software in Remote desktop chapter.

Remote desktop

To connect to the raspberry pi openplotter can setup an AP (access point) or and can be connected by an ethernet cable.

It can be easily connected by bonjour protocol with "openplotter.local" as address or use the IP address.

The IP address of openplotter as WiFi access point is **10.10.10.1** as default.

In \/home\/pi\/.config\/openplotter\/openplotter.conf you can change the address manually.

If you connect the raspberry pi to a router by ethernet cable (and bridge mode isn't checked) the raspberry pi will get the IP address from the routers dhcp server. So connect to the router and look what IP address it sets for openplotter.

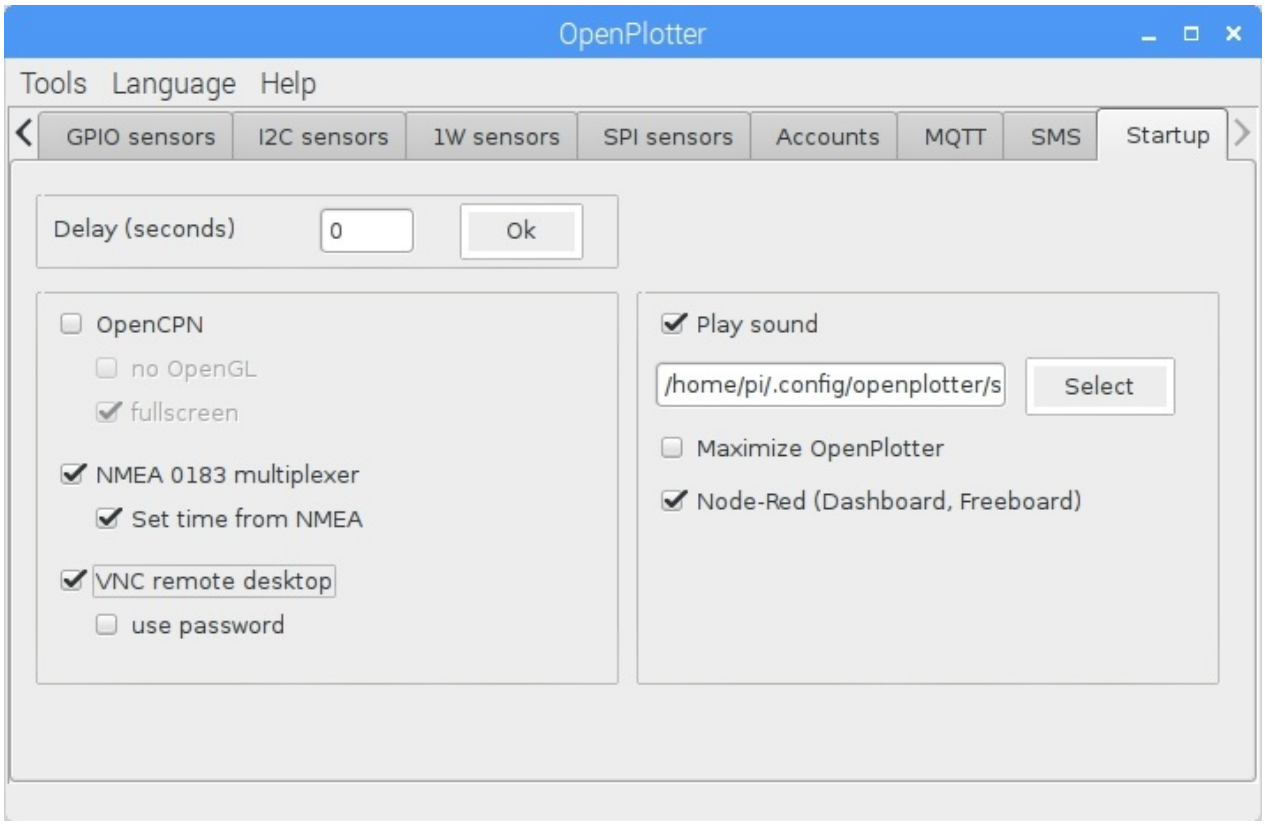
You have to install a remote desktop client on your remote device. There are two types, VNC and RDP, and some applications can use both of them.

VNC replicates the current desktop in OpenPlotter.

RDP can replicate the current desktop or create a new one with different resolution if you want. RDP is faster than VNC.

VNC

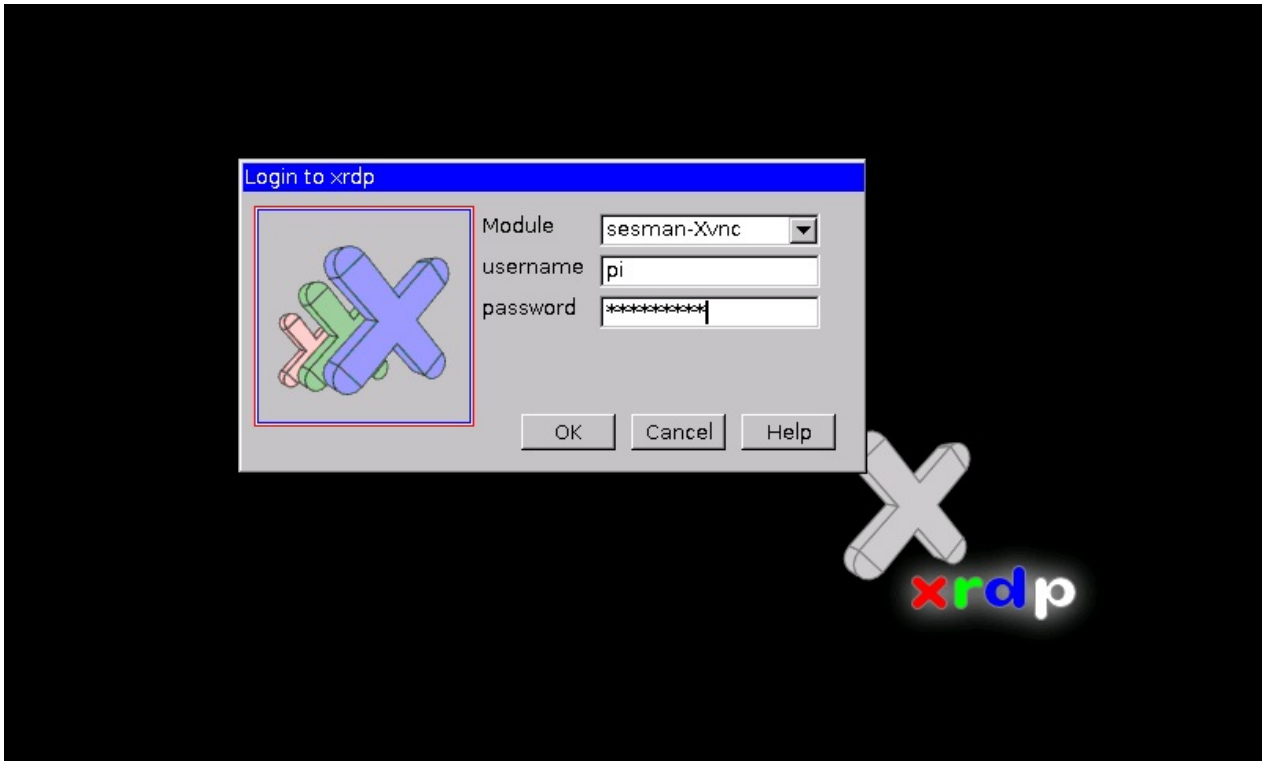
Be sure the checkbox *VNC remote desktop* is enabled in the *Startup* tab.



To connect by VNC you have to provide the IP of OpenPlotter and the port 5900.

RDP

To connect by RDP you have to provide just the IP or openplotter.local. Then you will be prompted for the password of user *pi*. If you have not changed it, it should be *raspberry*. When using Module sesman-Xvnc you get a new desktop. If you want to replicate the current desktop use Module console.



Tested remote desktop clients

VNC

Linux: Vinagre.

Windows: RealVNC Viewer, TightVNC.

Android: bVNC, RealVNC Viewer, VNC per Android.

IOS: RealVNC.

RDP

Linux: Vinagre.

Windows: Windows 10 Remote Desktop Client, Windows CE 5.0 Remote Desktop Client.

Android: load an old apk (microsoft-remote-desktop-8-0-5-24406-en-android.apk)

Auto Setup USB ports

On Linux serial USB ports are numbered in the sequence they are detected.

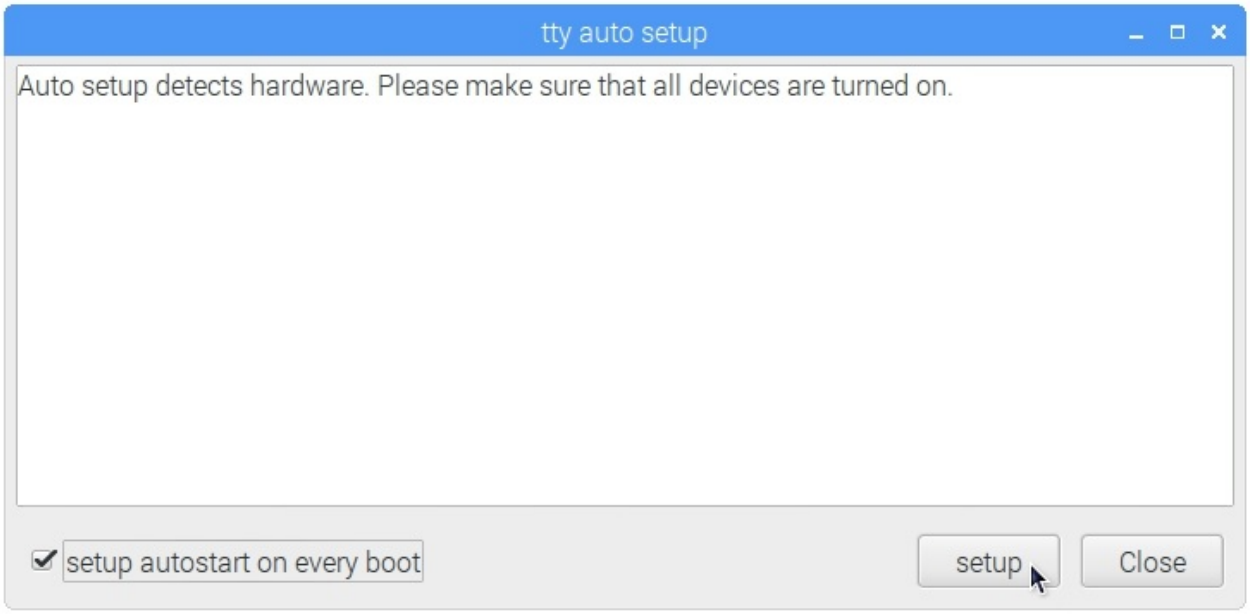
This could work some times. But it isn't reliable. If of any reason you loose a contact between the raspberry pi and an USB-serial port adapter. It could be that the entire configuration doesn't work anymore!

A more reliable way is, to give every serial port a name.

This is the reason why auto setup pop up on a fresh system.

Start tty auto setup

- all your devices should be connected to the internal \/ external USB-Hub (inclusive USB-gps-dongle and special CAN-BUS to USB adapter like actisense NGT-1 or !!link!!)
- turn on all your NMEA 0183 devices



- If setup autostart on every boot is checked uncheck it
- start the setup process by the setup buttom.

The tty auto setup can be started manually by Tools->Auto Setup Start

Auto setup is sometimes able to find

- NMEA 0183 ports

It auto detects the baud rate and checks if there are only NMEA 0183 sentences of one device-ID.

example1: $APxxx

- It will select the port name ttyOP_AP

example2: $GPxxx

- It will select the port name ttyOP_GP

If there are more device-IDs it will select ttyOP_MPX1, next tty_MPX2,...

If it detects an openplotter compatible CAN-USB adapter it selects ttyOP_N2K

To check if everything is done go to USB manager.

USB manager

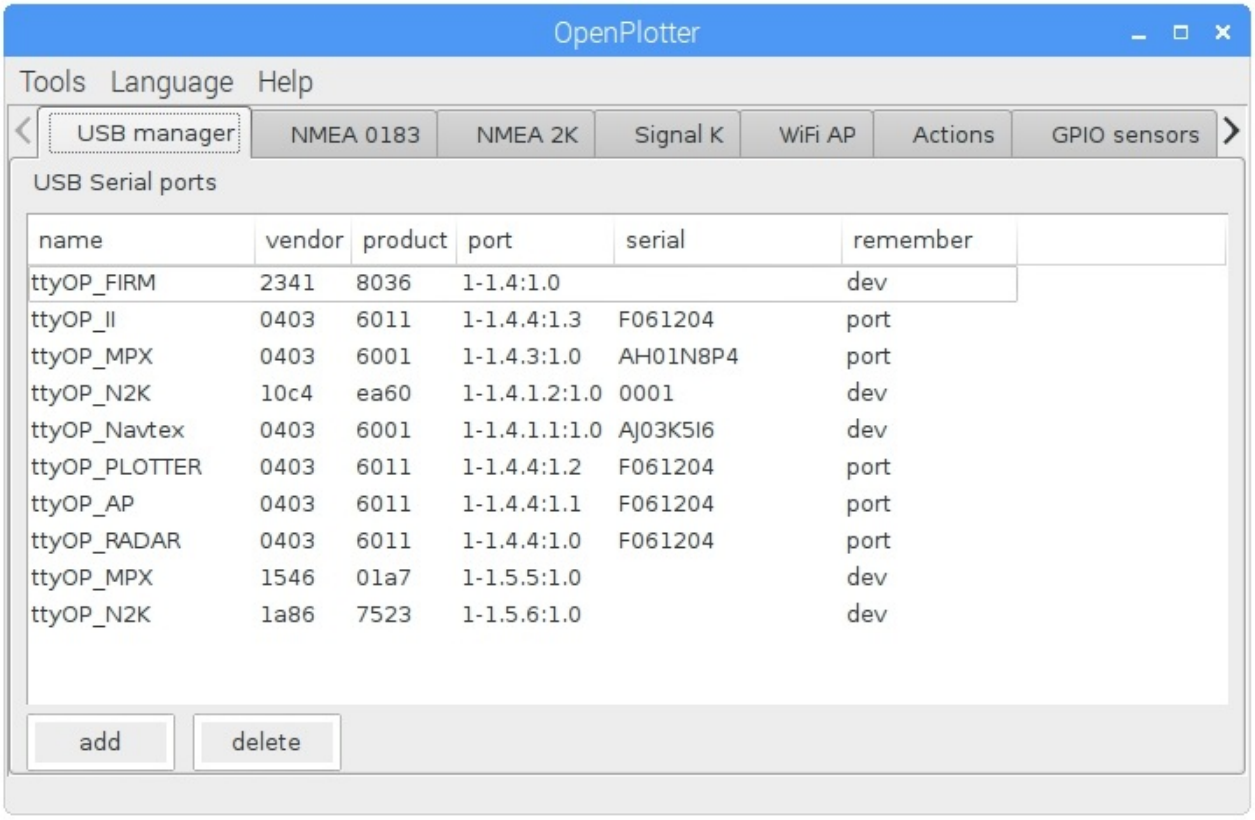
Why is it important to manage the serial ports?

Linux does use standard port names. They are named by the connection time.

If you can guarantee that every time all USB adapter are connected and that there is no chance of malfunction on one device and you never put another device to the system, then you don't need the USB manager.

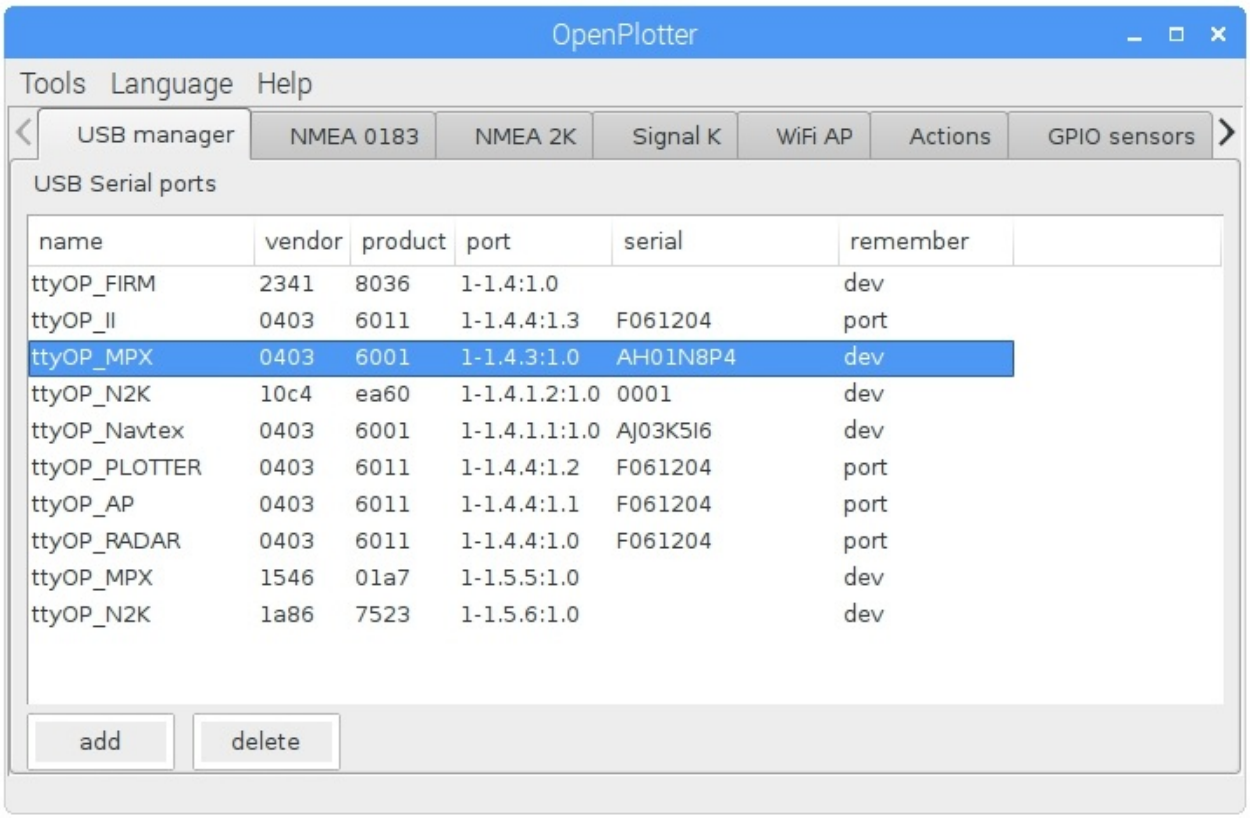
Otherwise the names ttyUSBx are changing (for example the AIS is on the autopilot port, the N2K Can-Bus is on the GPS port …). The multiplexer isn't able to do his job any more.

Here the USB-ports can be setup manually.



Picture 1: already configured USB ports

select add to open a window of not configured serial ports.

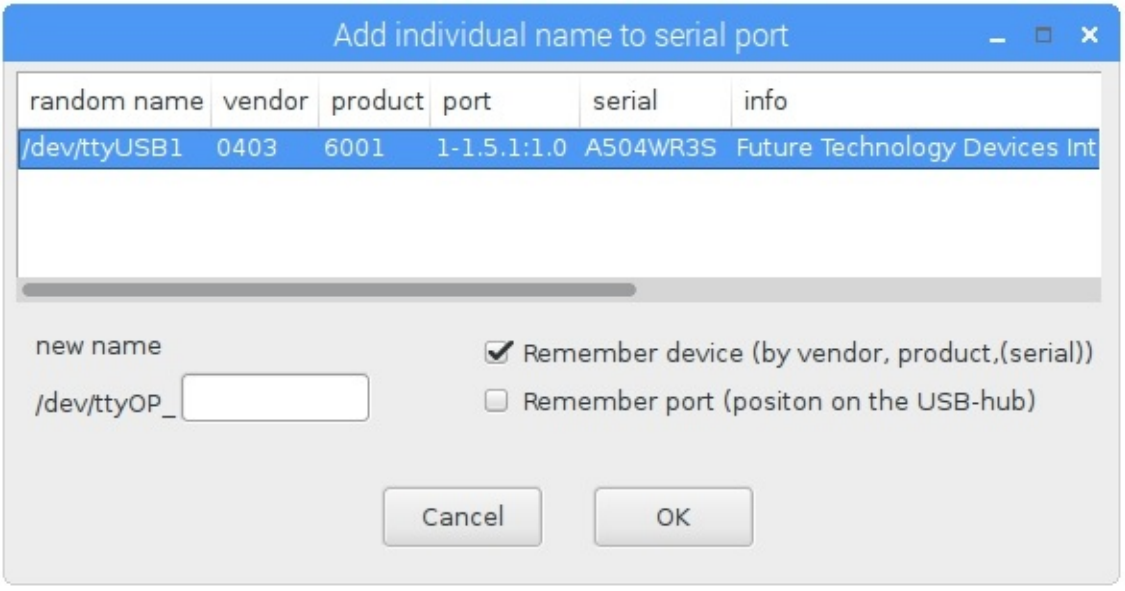


Picture 2: adding a FTDI port

random name: the numbered port name which is given by Linux

vendor product and serial: numbers and strings of the device

port: the USB port position



Picture 3: USB port names of the raspberry pi

If you add a new serial port you can select remember by device or remember by port.

Remember by port can be used every time. See picture 3 it shows the USB port names of the raspberry pi.

But normally you want the system to get the right name independent of the port it is connected to.

You can do this if there is not a second device with identical information. In picture 1 you see for example an 4 port FTDI hub. It has a vendor and a product number and a serial string. But you have 4 times identical data. So you can't use remember by device. There are cheaper adapter which haven't got a serial number. If you have more than 1 of them you also can't use remember by device.

Naming

The Linux names ttyUSB0, ttyUSB4 don't have any information what is connected to the port so it's better to use a meaningful name.

The USB manager uses ttyOP_xxxx.

Where xxxx belongs to personal decision. But we recommend to use

ttyOP_N2K: for actisense ngt-1

ttyOP_AP: for autopilot

ttyOP_GP: for GPS

ttyOP_II: for seatalk

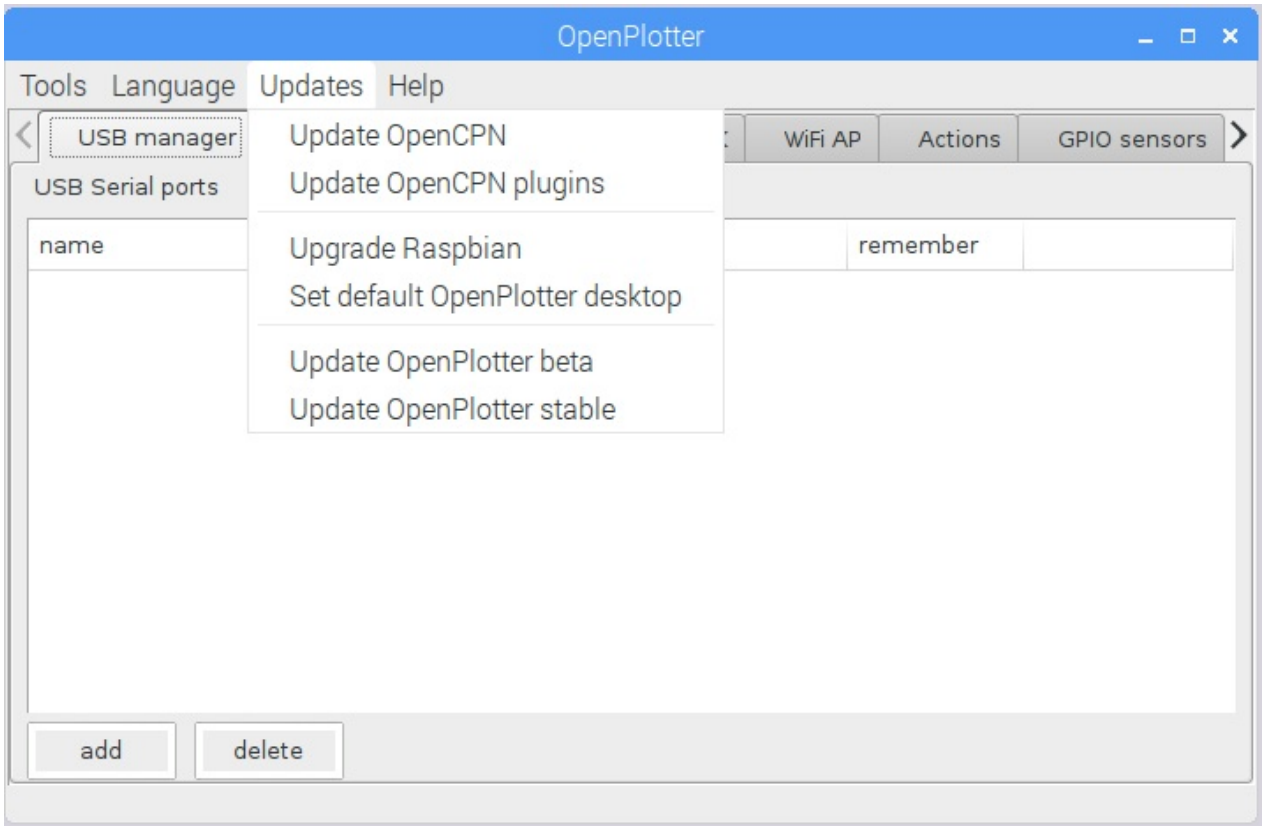
ttyOP_FIRM: for arduino with firmata

ttyOP_MPXx: is used for multiplexers but it could be better to rename it to a meaningful name like AIS or PLOTTER.

There are other devices which doesn't send anything they're only listening for sentences add a meaningful name to them.

To change a port name double click the line.

Updating OpenPlotter



From OpenPlotter version 0.10.0, you can update your installation without need of burning a new SD card.

Be sure you are connected to Internet and go to *Updates* in the OpenPlotter main menu.

You can update OpenCPN and plugins to the latest stable releases.

There you can update OpenPlotter to either a beta or to an stable version. Once you have updated from an stable version to a beta version you can not go back to the stable version and you have to keep updating the beta until a stable version is reached.

OpenPlotter will check if you need to do a minor or a major update and it will do all the work for you.

Version numbering

OpenPlotter releases have three numbers: **a**.**b**.**c** (v0.9.0, v0.10.0...) and a word (alpha, beta and stable).

When **c** increases, there is a minor change and means that only the OpenPlotter code has changed.

When **b** increases, there is a major change and means that other packages or dependencies need to be added or updated.

When **a** increases, there is an upgrade and means that Raspbian needs to be upgraded. In this case probably a new OpenPlotter image will be released and we recommend to burn a new SD card. You will be able to update OpenPlotter from the menu too but some parts may not work as expected.

Alpha means that some parts still need development.

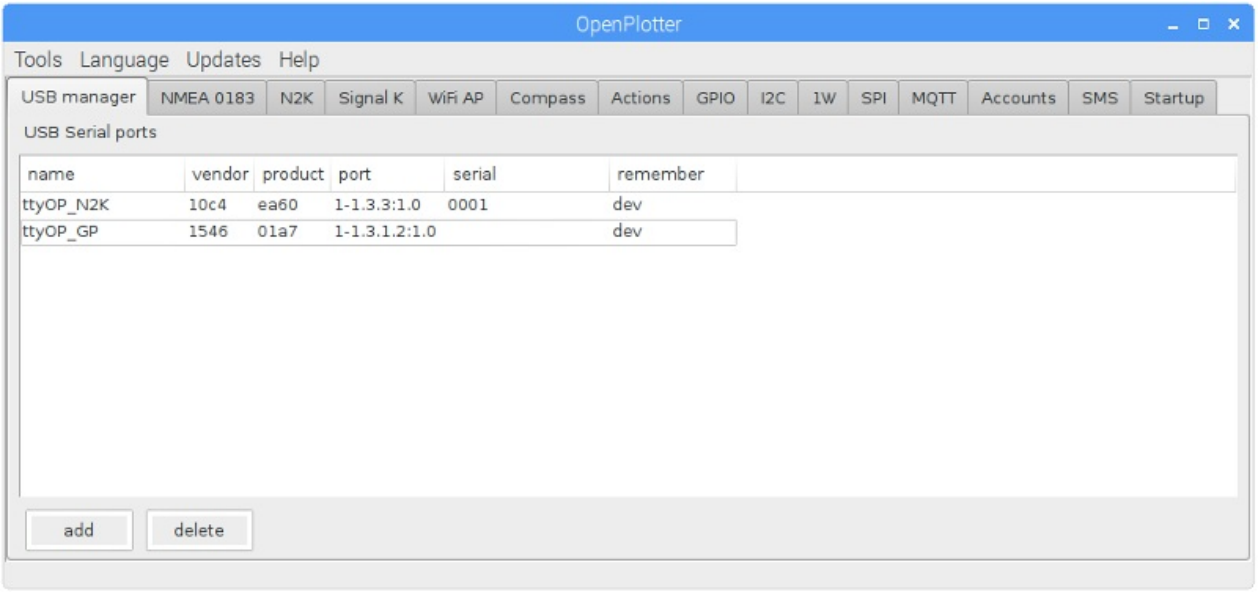
Beta means that all parts have been developed and there are not fatal errors but they need to be tested by users in different scenarios. Text will be translated from English into other languages on this stage.

Stable means that OpenPlotter code and dependencies have already been tested and there are not errors.

You can know what version you are running selecting the option *About* in *Help* menu.

OpenPlotter Tabs

Most of the core features of OpenPlotter can be configured from controls located on tabs.



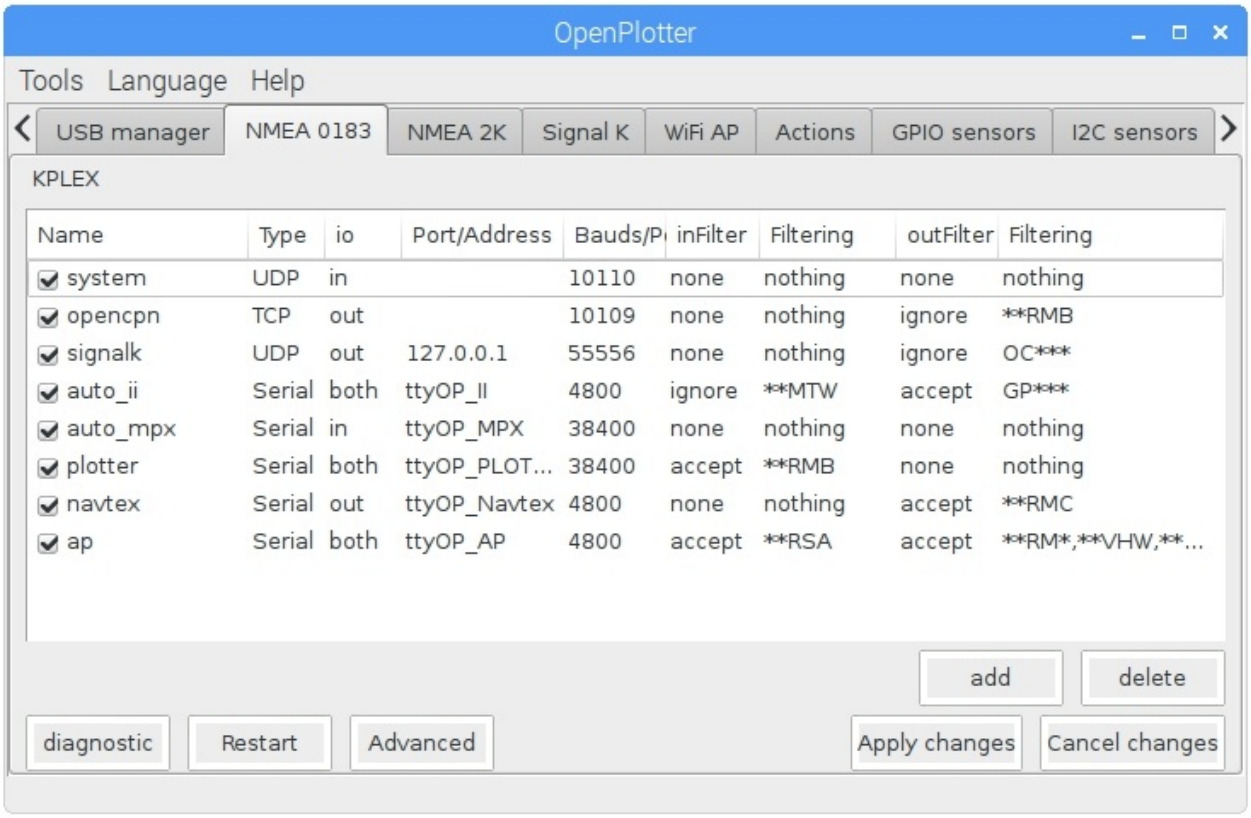
NMEA 0183 Multiplexer

It is important that you understand that OpenPlotter must drive all the NMEA data traffic to work properly, so you do not need to configure OpenCPN to get GPS signal, we will set this in OpenPlotter NMEA 0183 tab.

System defaults

There are three lines which couldn't be changed.

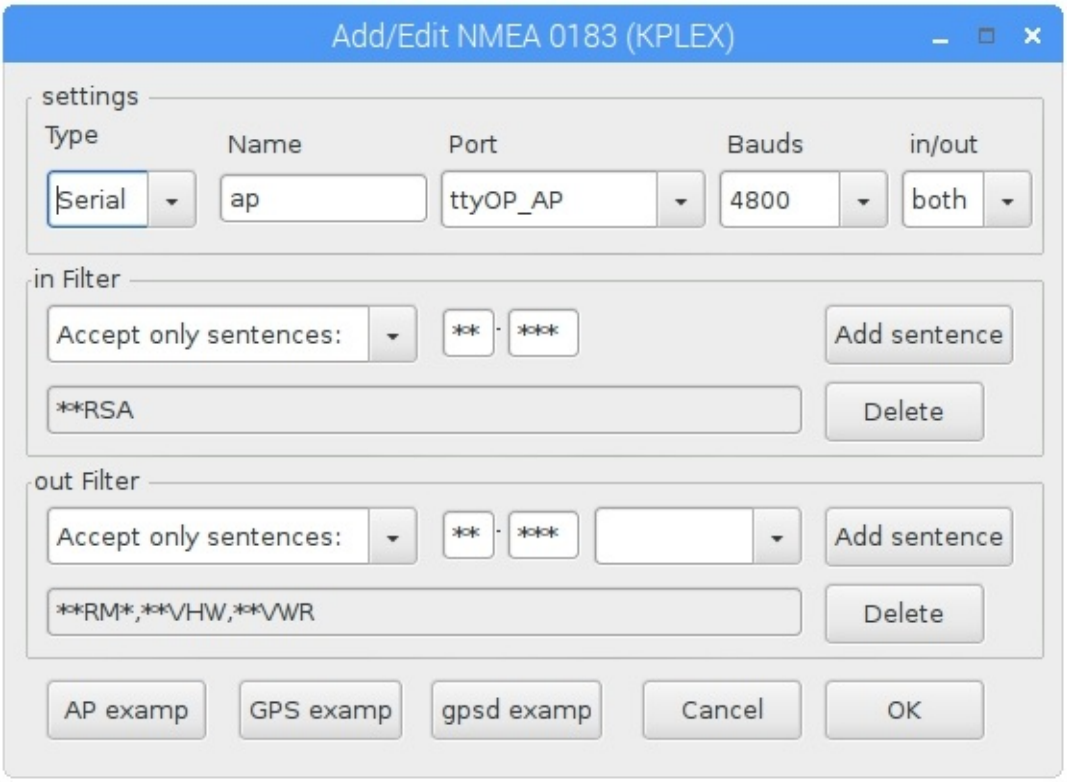
- system: Used by SDR-AIS, by Calculate and by NMEA 0183 generator
- opencpn: this is the datastream to opencpn
- signalk: this is the nmea 0183 stream to SignalK



Names with auto_xxx are made by Auto Setup USB ports to set the correct baudrate (the filters aren't set !!!).

Adding or changing a connection

- press add
- double click line to be edit



Kplex can work with Serial, TCP or UDP connection (Type).

Name: lower case letters should be meaningful

Port:

- Serial: The name of the port ttyxxxxx
- TCP / UDP: port number

Bauds: Baudrate for serial port

in / out:

- Serial: both, in, out
- TCP / UDP: in, out

in Filter: Data coming from the device

- none
- Accept only sentences
- Ignore sentences
    - **: device id
    - ***:type

  Use Add sentence to add a selection.

out Filter: Data send to the device. Same as above but filter can be connected to a given name

The three example buttons load default filter.

Diagnostics

On NMEA 0183 tab is the button diagnostic.

Select a line and click the button.

One or two windows pop up (two if setting is "both").

They show the transfered sentences

On the lower left side you can see the calculated transfer speed in baud.

! If the output speed is higher than the baudrate of the device some data will be cut. Select filters to send only needed data!

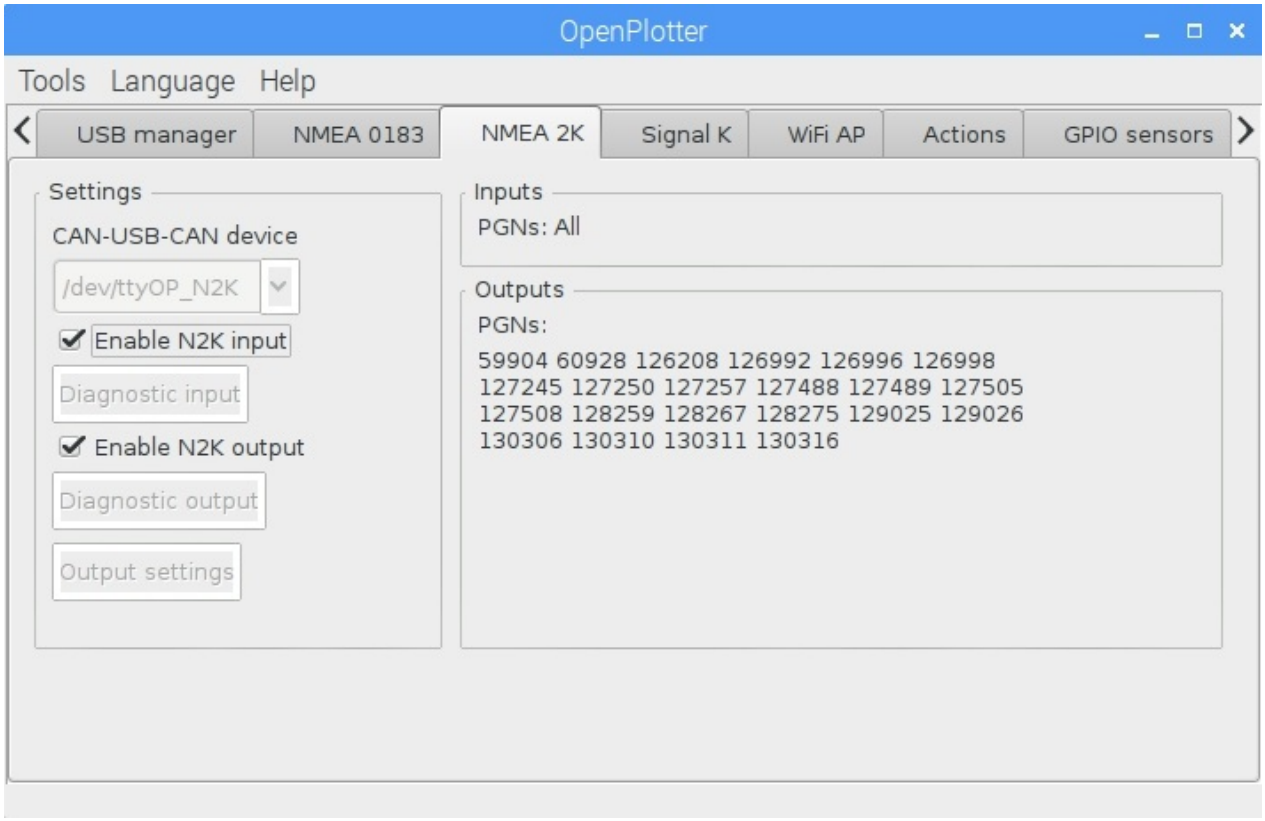
Advanced

There are many settings which could be done in OpenPlotter directly. But Kplex has advanced options which aren't used often. These settings or filter can be done between the lines

###end of OpenPlotter GUI settings

###Manual settings

N2K



Settings

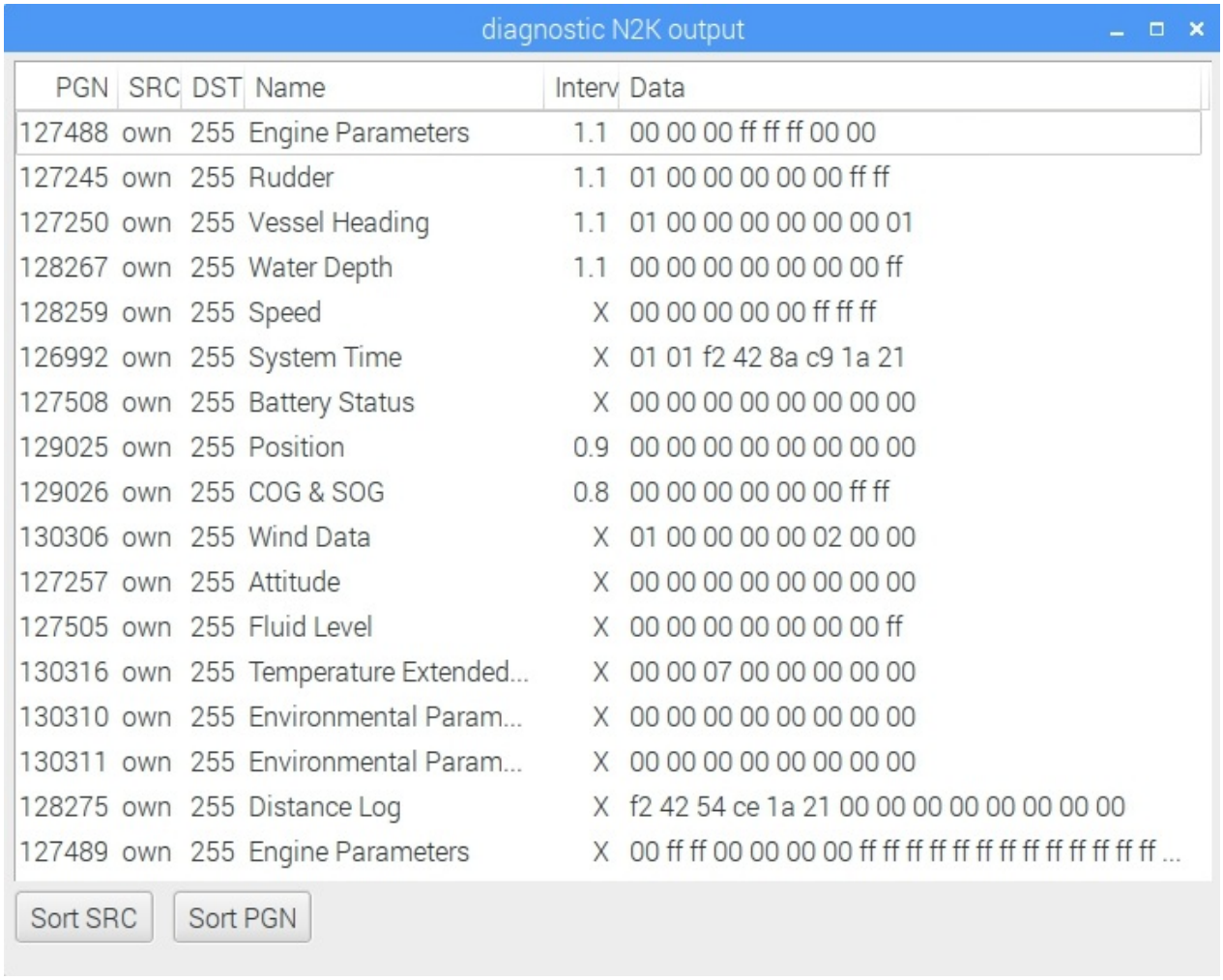
- select the N2K port
- activate Enable N2K input
- activate Enable N2K output

Diagnostic input

If button is gray, you have to deactivate Enable N2K input

Diagnostic output

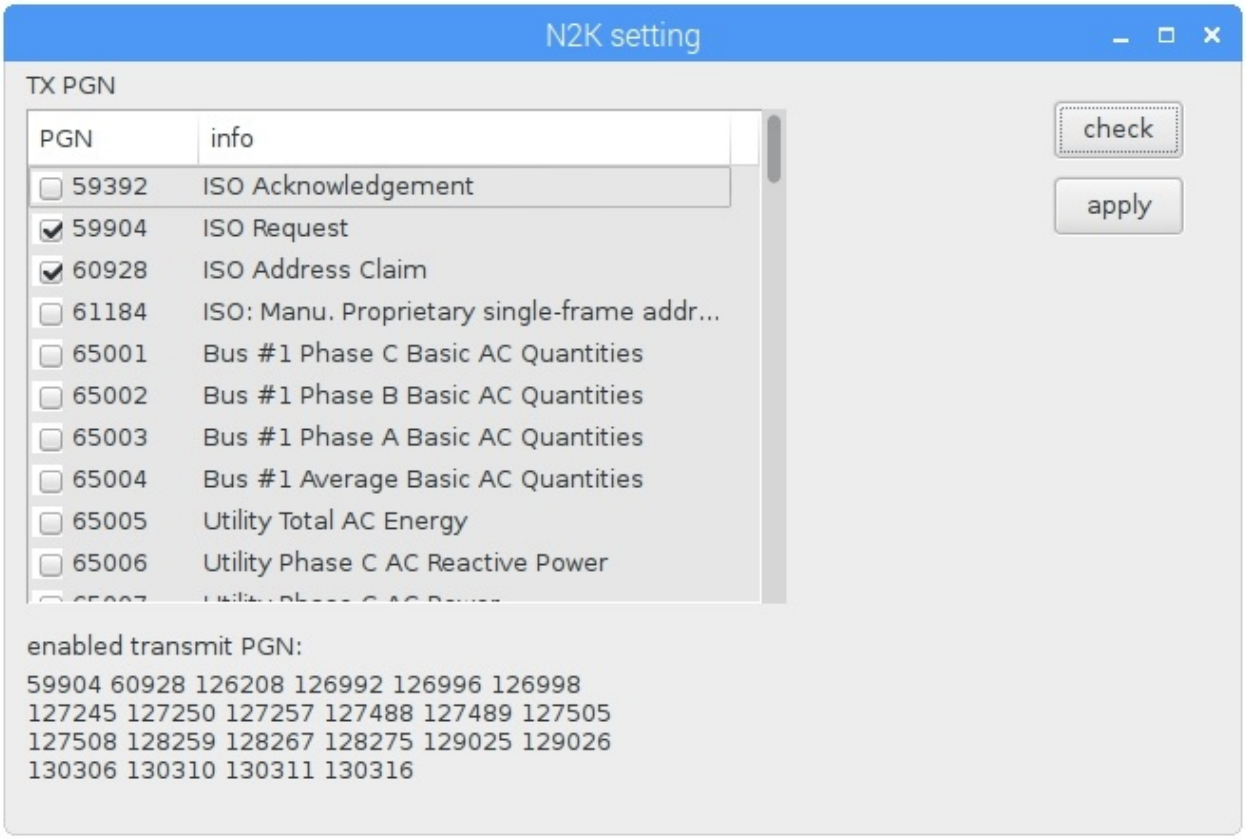
If button is gray, you have to deactivate Enable N2K output



Output settings

If button is gray, you have to deactivate Enable N2K input and Enable N2K output

These are hardware settings! If nothing is checked everything is blocked. You can't get anything out (security).

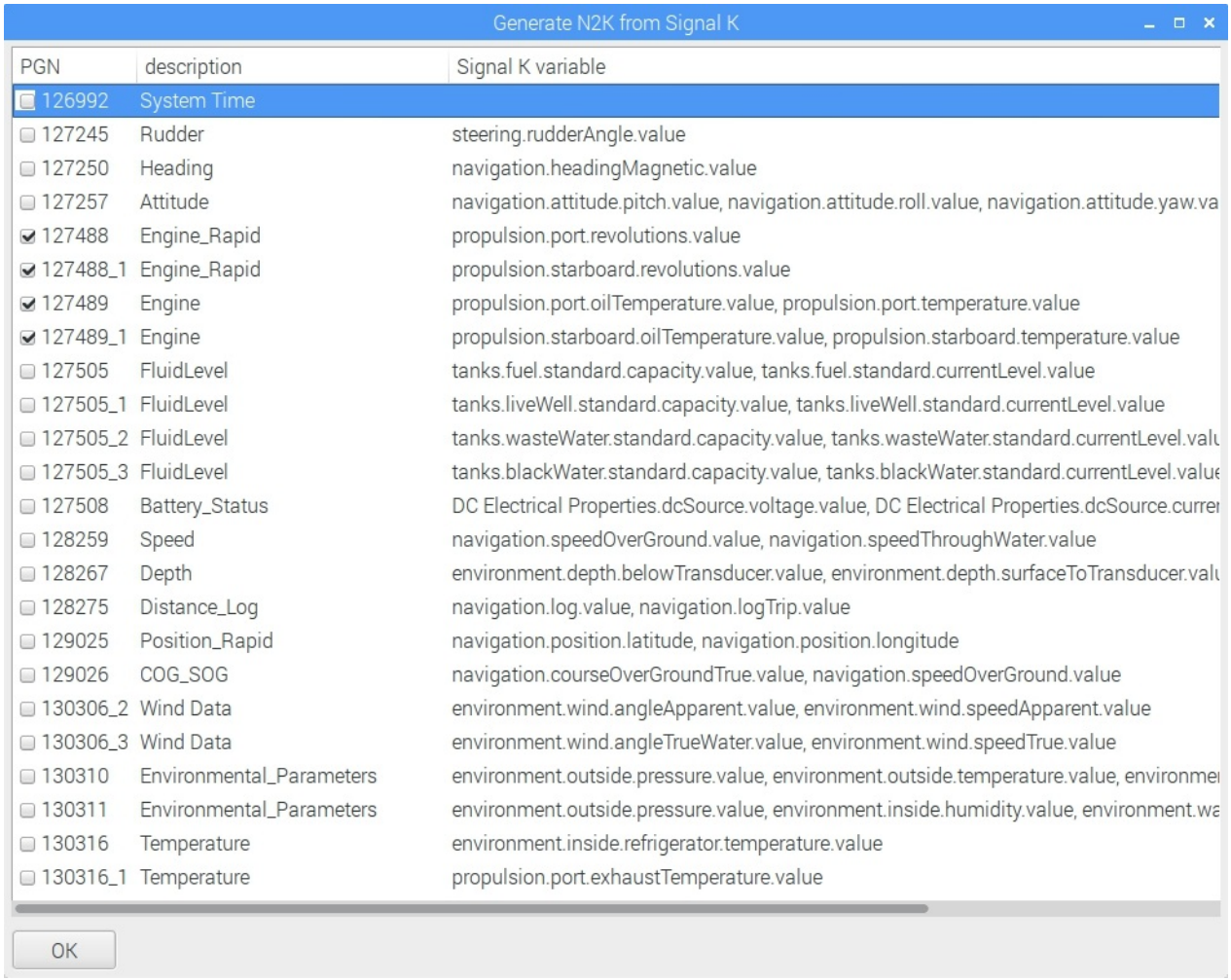


After opening the dialog box press two times check to read the settings from the CAN-BUS adapter.

First step is to enable needed PGNs for output. Then press apply and press check. Check if settings are done.

NMEA 2000 generator

Tools->NMEA 2000 generator



Check the PGNs you want to send onto the NMEA 2000 bus (it only works if the output settings for the PGNs are also set).

Signal K

Is the base protocol of OpenPlotter.

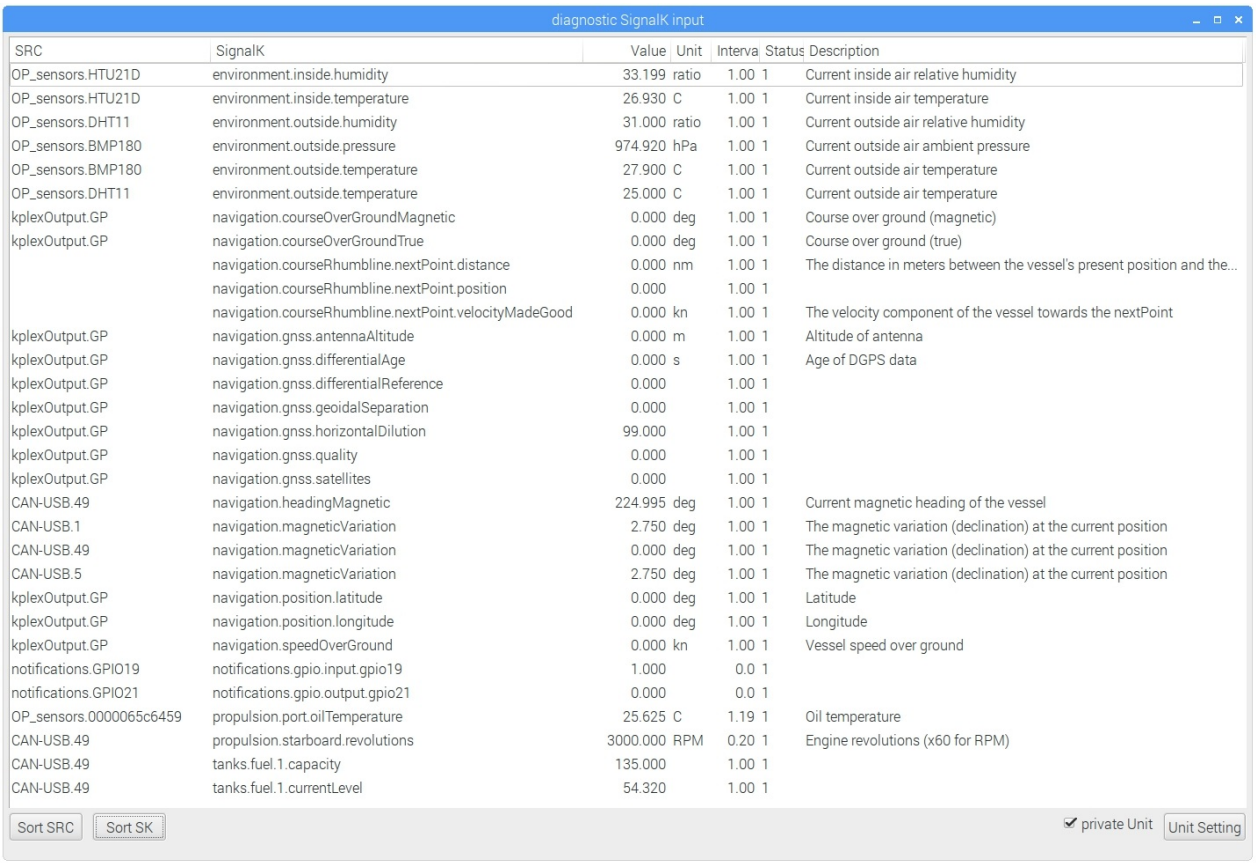
It is optimized for Internet browser. The data stream depend on json format.

both NMEA formats are converted to SK. The format is open and the values are readable. It uses strict SI-units.

On the SignalK tab you can enter the MMSI of the boat and "apply changes".

Diagnostic

SignalK diagnostic



SRC	SignalK	Value	Unit	Interva	Status	Description
OP_sensors.HTU21D	environment.inside.humidity	33.199	ratio	1.00	1	Current inside air relative humidity
OP_sensors.HTU21D	environment.inside.temperature	26.930	C	1.00	1	Current inside air temperature
OP_sensors.DHT11	environment.outside.humidity	31.000	ratio	1.00	1	Current outside air relative humidity
OP_sensors.BMP180	environment.outside.pressure	974.920	hPa	1.00	1	Current outside air ambient pressure
OP_sensors.BMP180	environment.outside.temperature	27.900	C	1.00	1	Current outside air temperature
OP_sensors.DHT11	environment.outside.temperature	25.000	C	1.00	1	Current outside air temperature
kplexOutput.GP	navigation.courseOverGroundMagnetic	0.000	deg	1.00	1	Course over ground (magnetic)
kplexOutput.GP	navigation.courseOverGroundTrue	0.000	deg	1.00	1	Course over ground (true)
	navigation.courseRhumbline.nextPoint.distance	0.000	nm	1.00	1	The distance in meters between the vessel's present position and the...
	navigation.courseRhumbline.nextPoint.position	0.000		1.00	1	
	navigation.courseRhumbline.nextPoint.velocityMadeGood	0.000	kn	1.00	1	The velocity component of the vessel towards the nextPoint
kplexOutput.GP	navigation.gnss.antennaAltitude	0.000	m	1.00	1	Altitude of antenna
kplexOutput.GP	navigation.gnss.differentialAge	0.000	s	1.00	1	Age of DGPS data
kplexOutput.GP	navigation.gnss.differentialReference	0.000		1.00	1	
kplexOutput.GP	navigation.gnss.geoidalSeparation	0.000		1.00	1	
kplexOutput.GP	navigation.gnss.horizontalDilution	99.000		1.00	1	
kplexOutput.GP	navigation.gnss.quality	0.000		1.00	1	
kplexOutput.GP	navigation.gnss.satellites	0.000		1.00	1	
CAN-USB.49	navigation.headingMagnetic	224.995	deg	1.00	1	Current magnetic heading of the vessel
CAN-USB.1	navigation.magneticVariation	2.750	deg	1.00	1	The magnetic variation (declination) at the current position
CAN-USB.49	navigation.magneticVariation	0.000	deg	1.00	1	The magnetic variation (declination) at the current position
CAN-USB.5	navigation.magneticVariation	2.750	deg	1.00	1	The magnetic variation (declination) at the current position
kplexOutput.GP	navigation.position.latitude	0.000	deg	1.00	1	Latitude
kplexOutput.GP	navigation.position.longitude	0.000	deg	1.00	1	Longitude
kplexOutput.GP	navigation.speedOverGround	0.000	kn	1.00	1	Vessel speed over ground
notifications.GPIO19	notifications.gpio.input.gpio19	1.000		0.0	1	
notifications.GPIO21	notifications.gpio.output.gpio21	0.000		0.0	1	
OP_sensors.0000065c6459	propulsion.port.oilTemperature	25.625	C	1.19	1	Oil temperature
CAN-USB.49	propulsion.starboard.revolutions	3000.000	RPM	0.20	1	Engine revolutions (x60 for RPM)
CAN-USB.49	tanks.fuel.1.capacity	135.000		1.00	1	
CAN-USB.49	tanks.fuel.1.currentLevel	54.320		1.00	1	

Data can be sorted by source or by SignalK.

To change the units click on Unit Setting

If private Unit is unchecked standard SI-units are shown.

SignalK Simulate

tools->SignalK Simulator

It's a tool which streams Data to the SignalK server. You can test it with Diagnostic.

There is a ".conf" file which is opened by the setting buttom

item_8 = [8, 'environment.wind.speedApparent', 4, 0, 40, 0.5144441, 0]

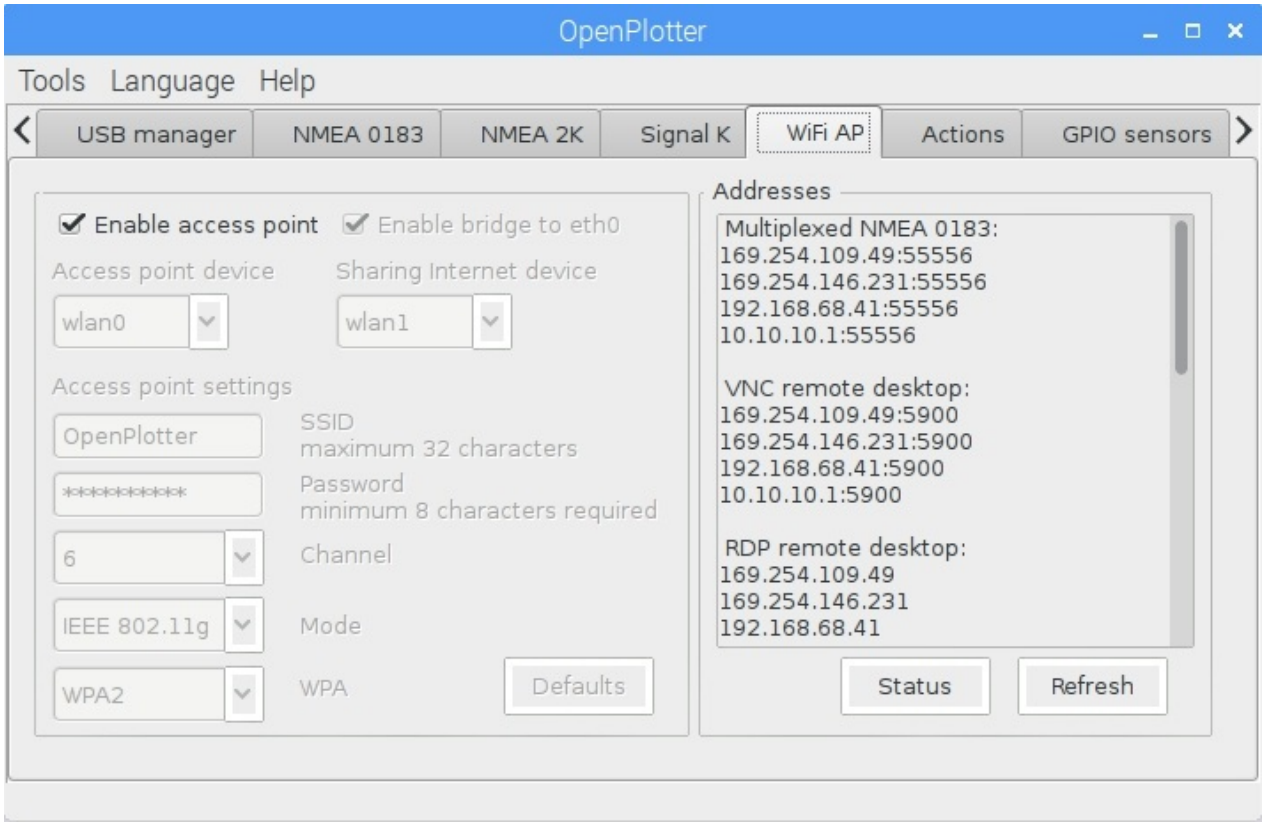
[Position in the GUI,SignalK-name,default value, min value, max value, multiplier, offset]

So it's easy to edit.

Start the GUI with the start buttom. The window is resizeable. Don't use more than 4 config lines on a small display.

!!! Never use this tool while sailing or boating !!!

WiFi AP



Picture: RPI active as AP with bridge to the Ethernet port. The internet connection comes over wlan1

If **Enable access point** is selected the AP should be active. To change settings you have to disable AP mode. **But be careful you will lose the connection to the RPI if you work headless.**

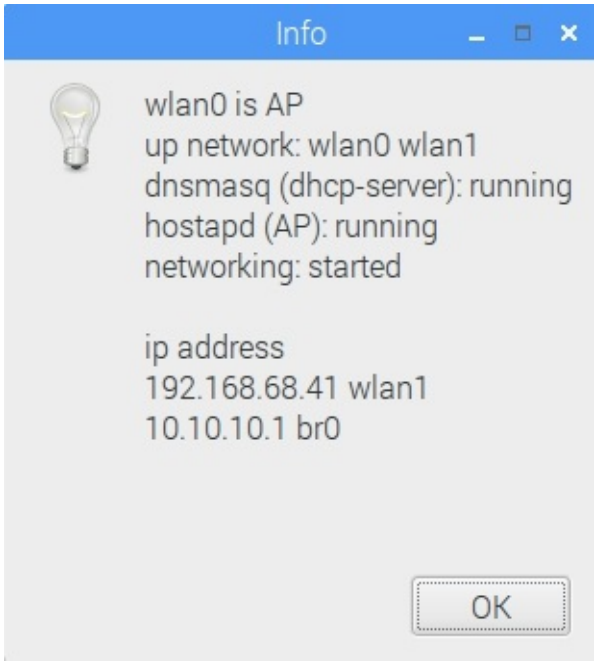
As **Access point device** you can only select AP mode able wlan (There were many changes in the RPI firmware and drivers only few wlan sticks work together. Sometimes if you add a non AP mode stick the AP mode for both is blocked.)

The second wlan is for connections to the internet (seaport wifi or mobile phone AP). If you do it this way all tablets mobile phones PCs haven't got to change the wlan when reaching a seaport. If the RPI has a connection to the internet (seaport wifi) all connected devices have internet access.

Sharing Internet can also be done by connecting to an internet router by Ethernet port. There is also a chances to connect to the internet with an gsm stick.(but it is easier to use mobile phone internet sharing over wifi)

The Ethernet port can be bridged to the AP to act like a normal router. This is important if you have a plotter with Ethernet port. Then the RPI acts like a gofree router.

When pressing the **Status** button



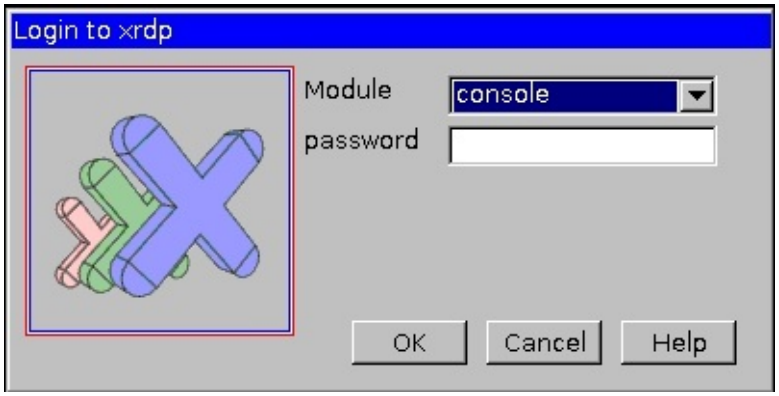
this window pops up and shows the network status.

RPI network manager

In the original RPI network-manager some devices show up as *device not managed*.

These devices are managed by OpenPlotter (in this example wlan0 is AP and bridge to eth0 is enabled).

In this example wlan1 can be managed by network-manager to connect to the internet.

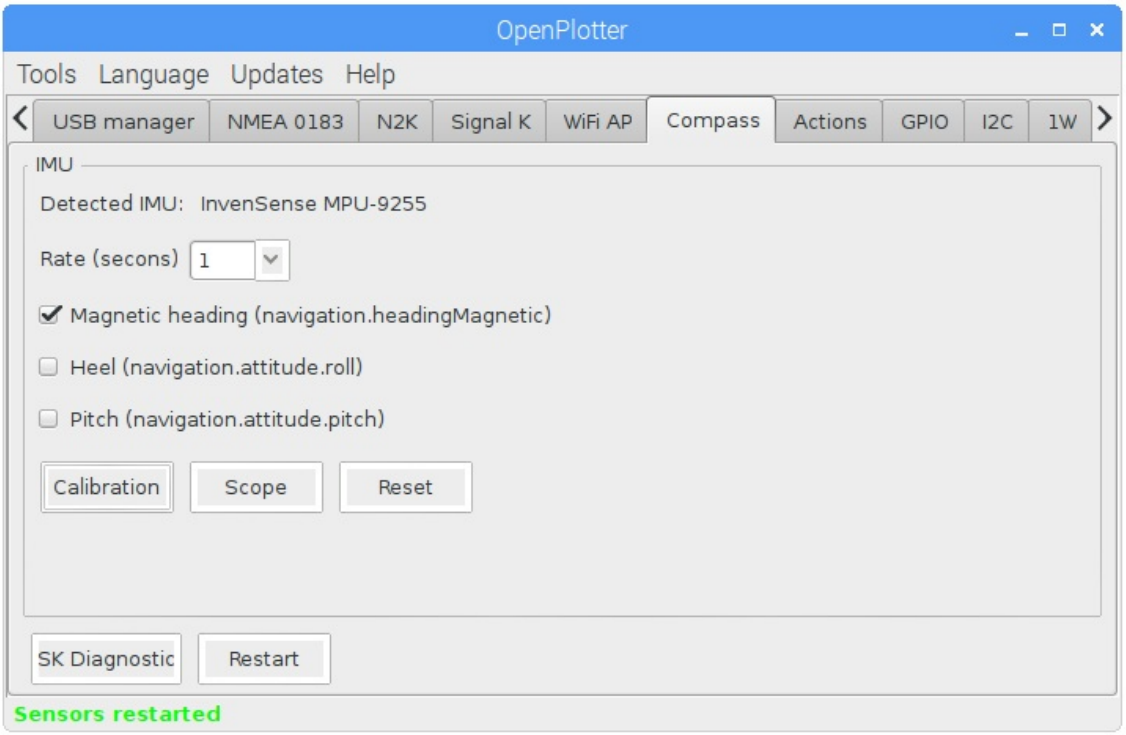


If you use remote desktop and want to connect to the internet by using the network-manager select **console**. Then you can connect to the internet otherwise linux security system blocks some settings.

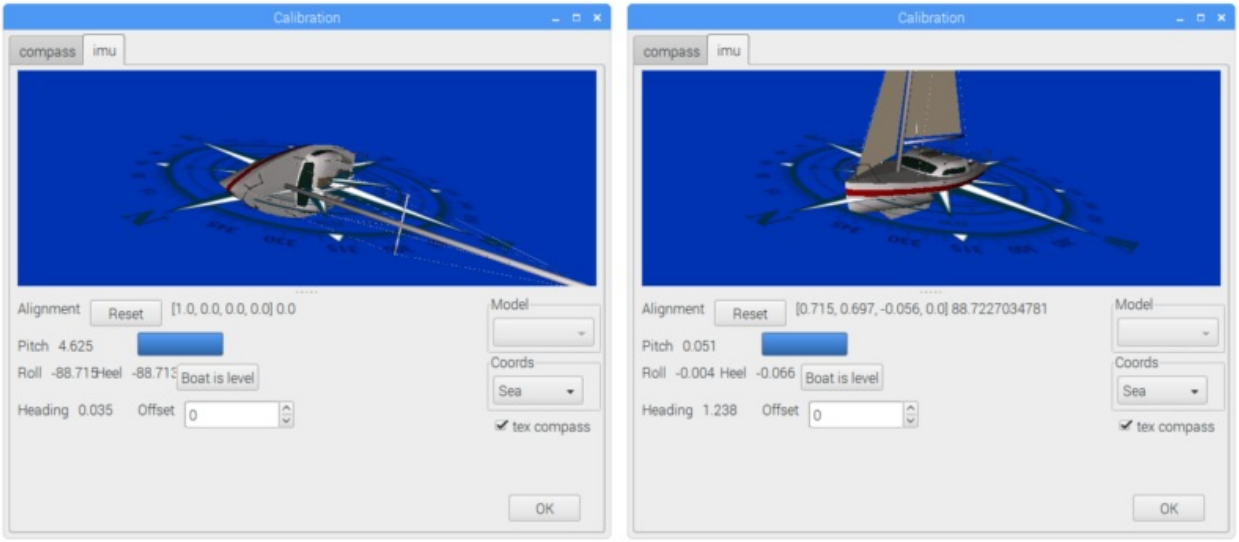
Compass

Place the IMU on his final position on board and away from possible metallic sources of interference, especially from speakers. After connecting the IMU, it should be detected on *Compass* tab.

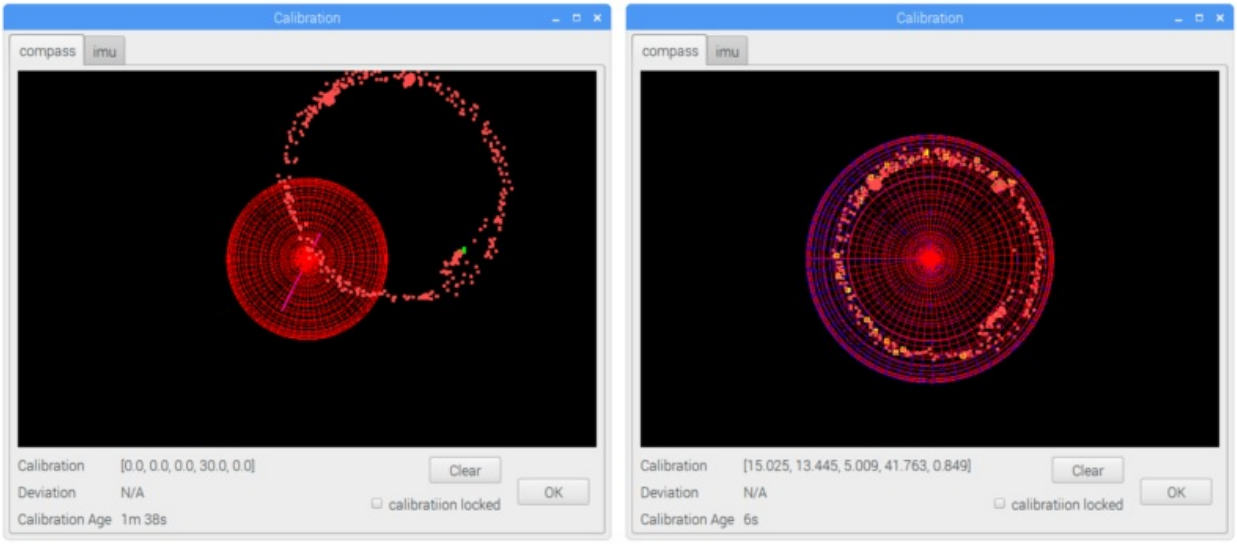
Enable reception of at least one of the 3 possible magnitude and press *Calibration*.



Being moored in port and the boat as more leveled as possible, select the *imu* tab and press *Boat is level*. After few seconds the 3D boat should be leveled.



Do not close the Calibration window, select the *compass* tab and go sailing normally. The system will collect data from the IMU during 2 minutes and if there is more than 60 degrees of heading variation and the collected data is valid, it will calibrate the IMU. A blue sphere should appear and the cloud of points should fit the red sphere.



The system will keep re-calibrating all the time every 2 minutes if reception of any of the IMU magnitudes is enabled and it has collected enough valid data. Every 10 minutes calibration and level data are saved to be used on next sesions.

Actions

Here you can activate reactions on events.

An event can be:

- SignalK data or timestamp or source
- Date

typical examples for events

If GPIO inputs are setup. They are displayed in SignalK (Diagnostic) when status changed.

An event can be:

- pushing a buttom which is connected to GPIO
- if a temperature rises over a level
- if the pressure drops under a level
- if a Signal K data wasn't updated in a time (lost)
- if the tank level drops under a level
- if battery voltage drops under a level
- if depth is lower then a level
- if speed is higher then a level
- ...

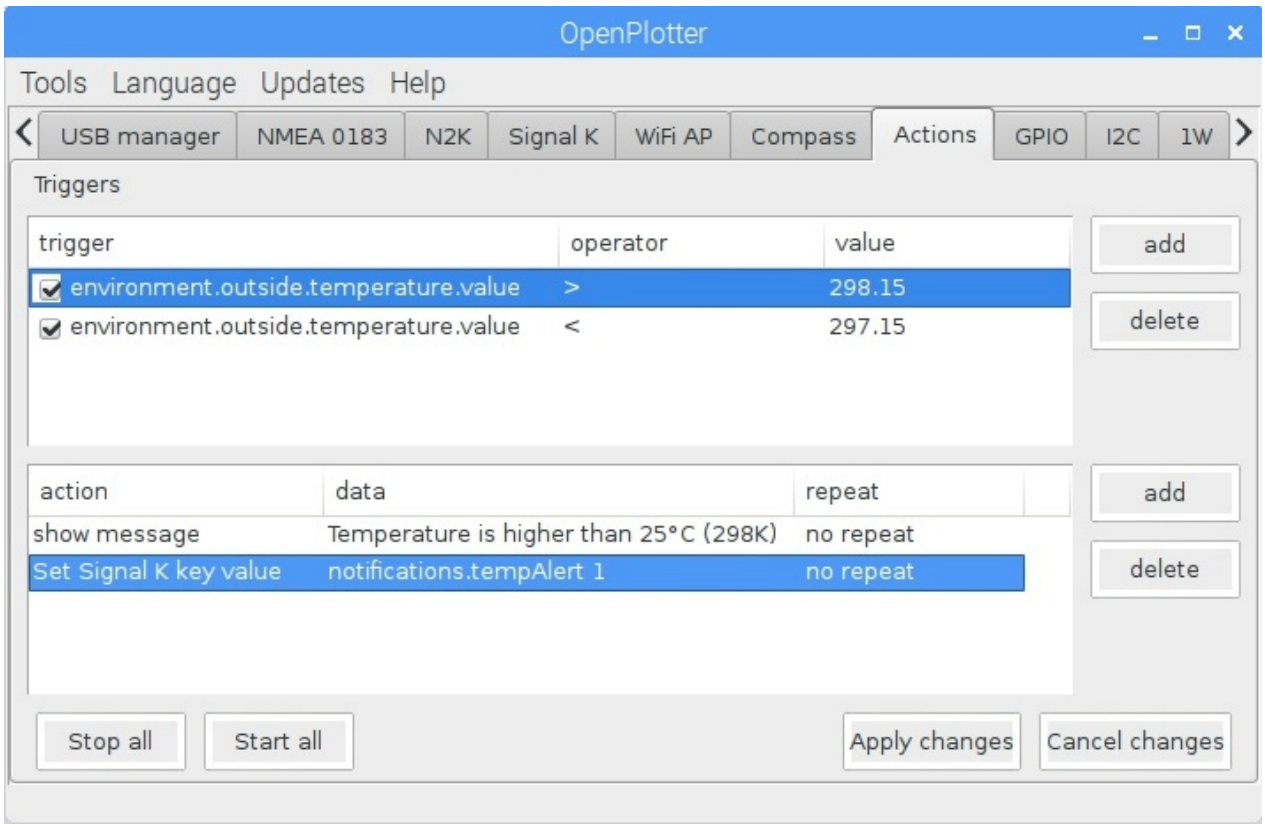
The action which is triggerd when the event occurs can be:

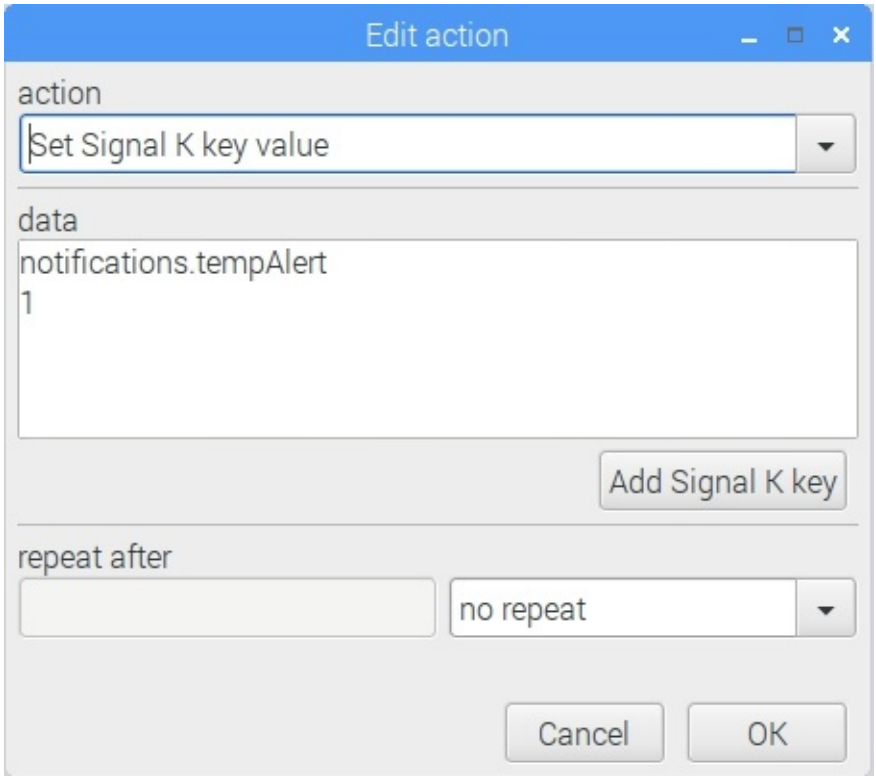
- play a sound (alarm, ...)
- Set Signal K key value (GPIO, alarm, ...)
- show message
- any command (some commands can be directly taken from the list. Command line commands must be entered)

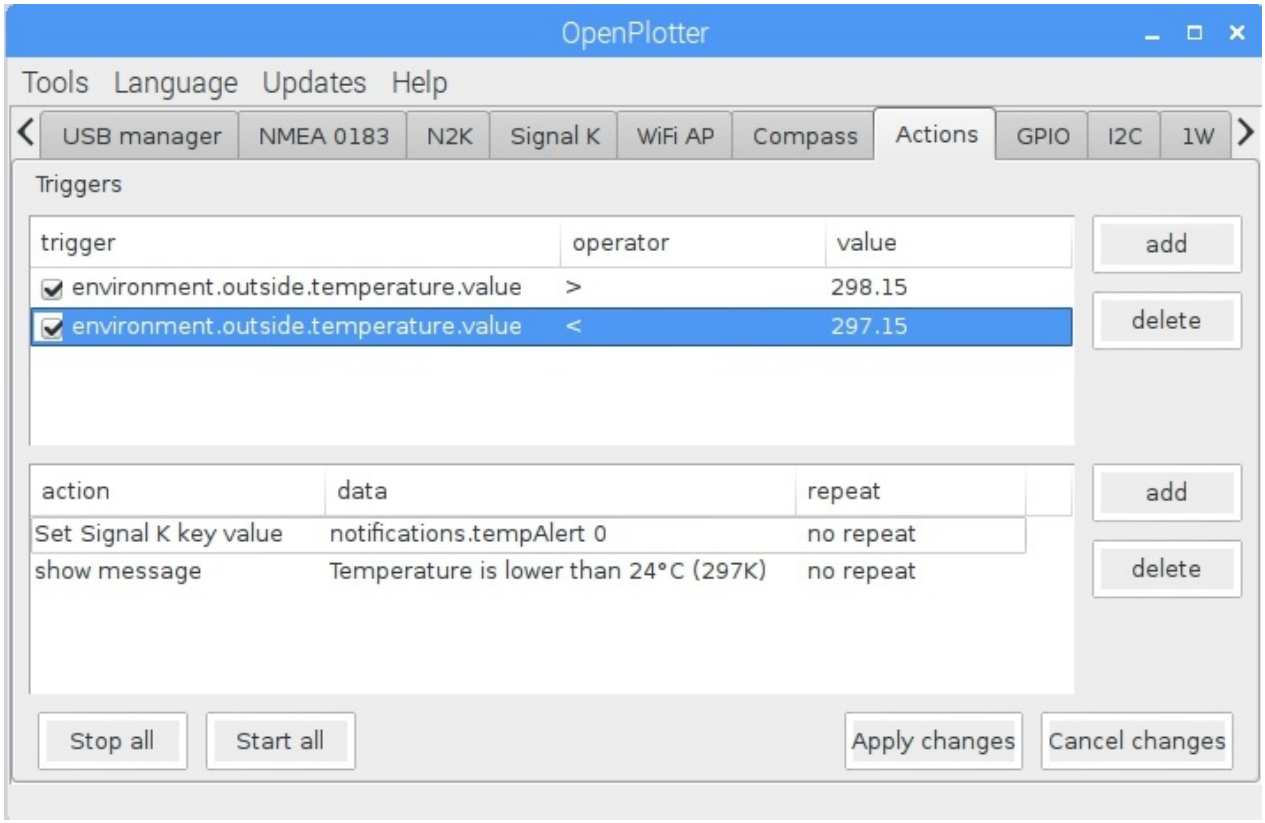
An example with temperature. When Temperature rises over 25 °C you want to be warned by a Message and a Signal K notification key should get the value 1.

If the temperature drops under 24 °C a warning message should pop up and the notification key should get value 0.

(don't forget that Signal K values are in SI unit in this case Kelvin)


OpenPlotter
Tools Language Updates Help
USB manager NMEA 0183 N2K Signal K WiFi AP Compass Actions GPIO I2C 1W
Triggers
trigger operator value
environment.outside.temperature.value > 298.15
environment.outside.temperature.value < 297.15
add
delete
action data repeat
show message Temperature is higher than 25°C (298K) no repeat
Set Signal K key value notifications.tempAlert 1 no repeat
add
delete
Stop all
Start all
Apply changes
Cancel changes


Edit action
action
Set Signal K key value
data
notifications.tempAlert
1
Add Signal K key
repeat after
no repeat
Cancel
OK



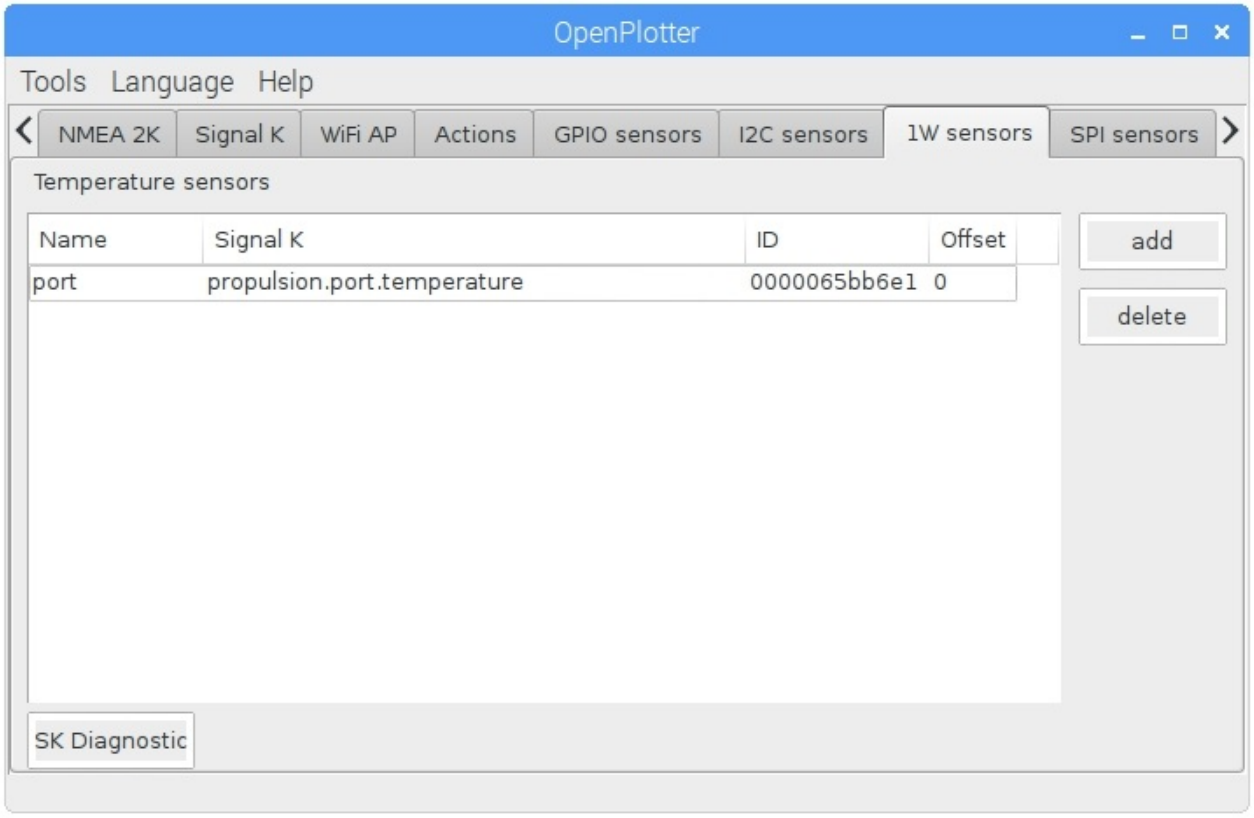
This example demonstrates the usage

GPIO sensors

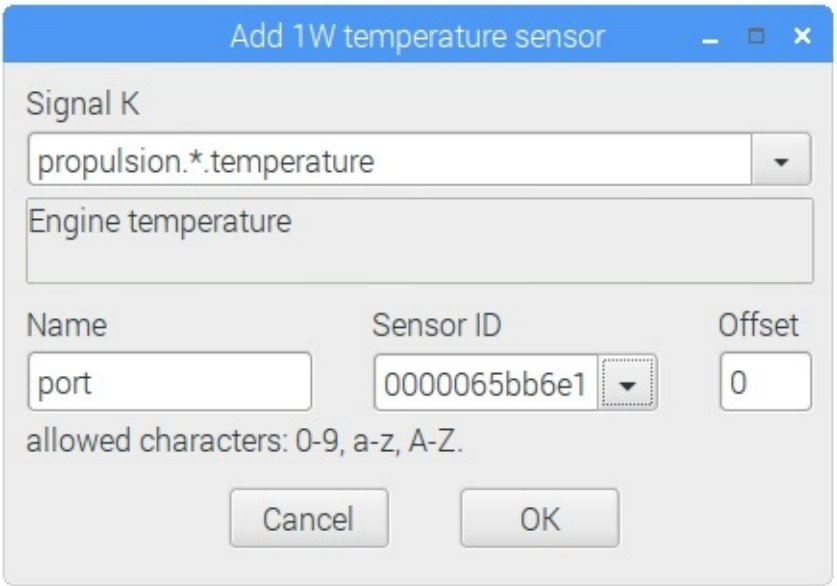
I2C

1W

- DS18B20
- One wire temperature sensor



Use add key to add another sensor. Double click the line to edit it.

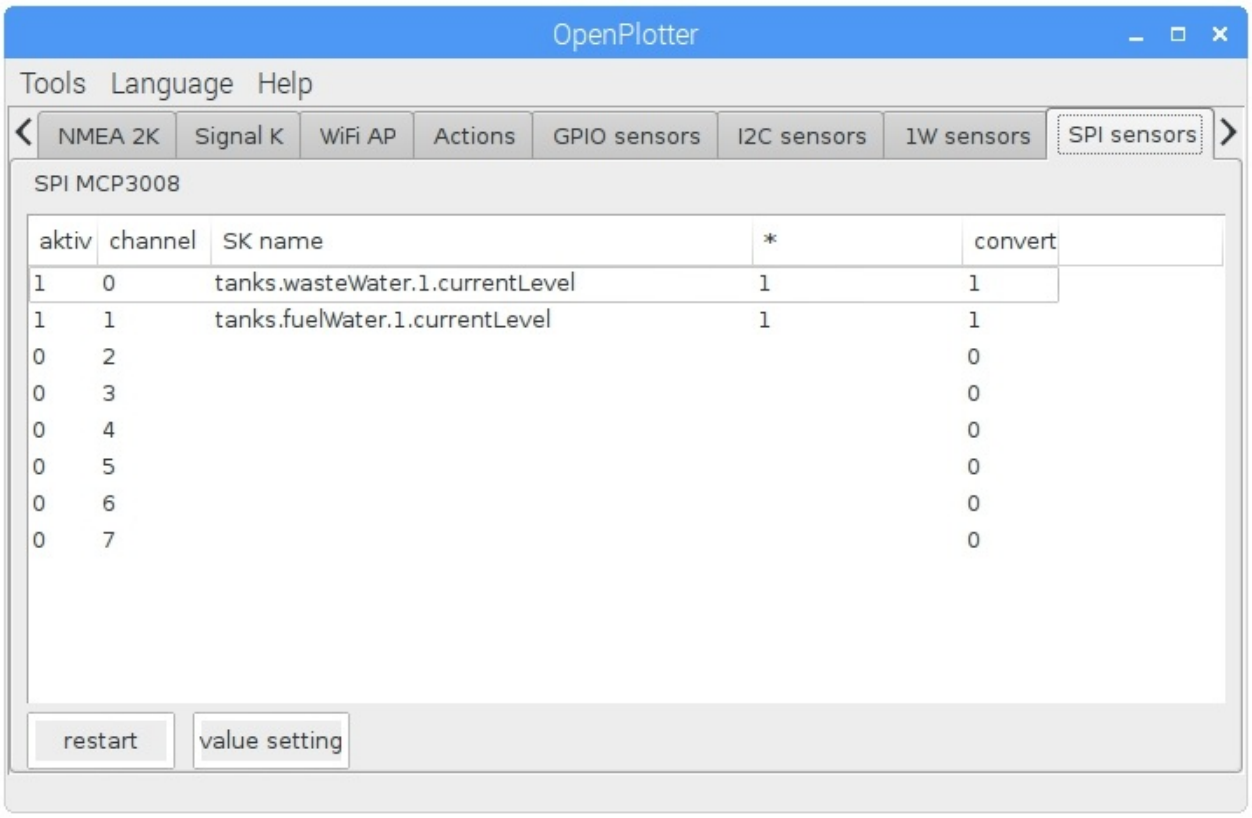


Use offset to correct value.

If there is a star in SignalK name. The text in field Name will replace the star.

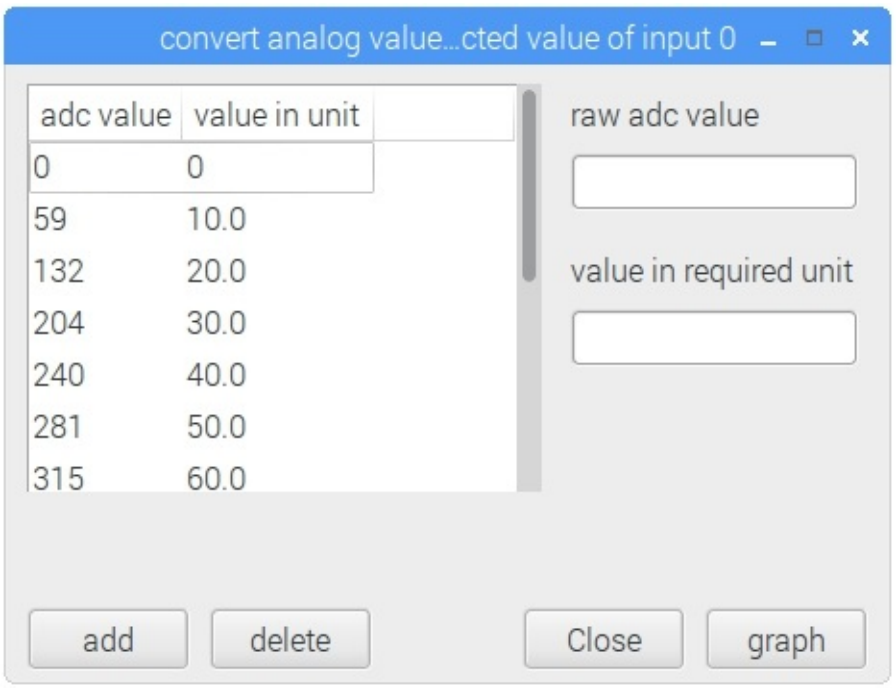
SPI

The adc (analog digital converter) MCP3008 is implemented in OpenPlotter.



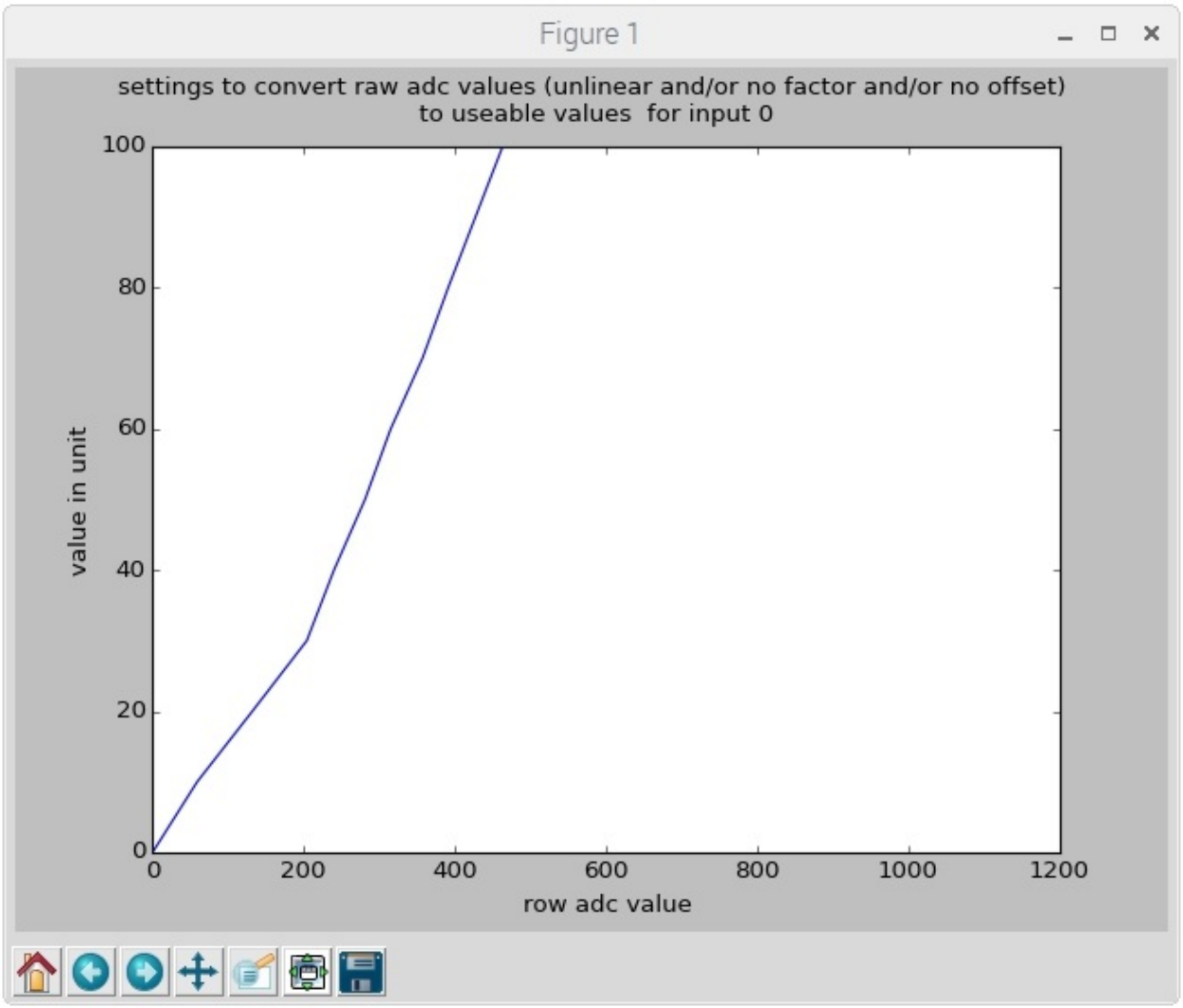
It does read voltage between 0 - 3.3 V. It is has a 10 bit resolution (0-1023).

Many signals aren't linear. To get a nearly linear result you can use the value setting.



The medium level in a tank on boats isn't linear to the height the sensor measures. To calibrate it empty it and insert the ADC value for 0%. fill in 10% of max volume and insert ADC value and 10.0. Go on until the tank is full.

Look at the graph you created by clicking on graph



Accounts

This chapter is under construction

To test remote monitoring by Twitter and Gmail feature, open new accounts. DO NOT use existing accounts or usual passwords because they can be exposed.

Open a new Gmail account and enable:

https://www.google.com/settings/security/lesssecureapps

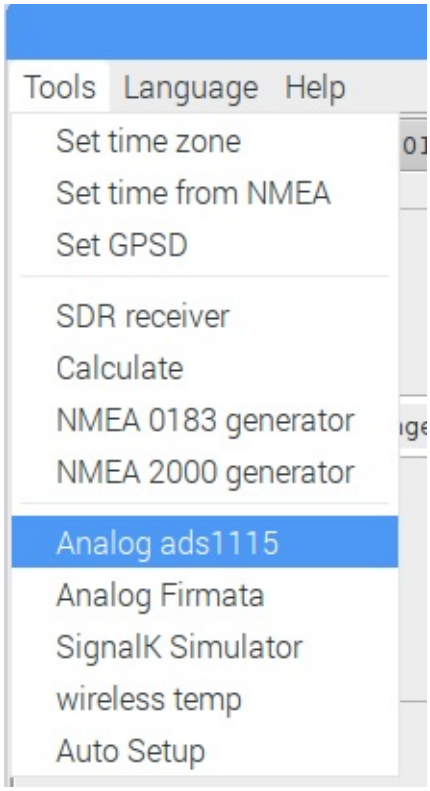
Using your new Gmail account, open a new Twitter account following this manual:

http://www.instructables.com/id/Raspberry-Pi-Twitterbot/step2/Create-ema

MQTT

Tools

In the tool menu are special modules.



The first three are:

- Set time zone

Opens a menu to choose a new time zone.

(It is the same as utc+x or utc-x)

- Set time from NMEA

The time from GPS (RMC-sentence) will be used to set the time of the RPI.

In future release it will read it from SignalK (then time can be taken not only from NMEA0183).

(If the RPI is connected to the internet it will get the time from a timeserver)

- Set GPSD

Opens the configuration file of GPSD in editor nano.

For details of GPSD google is your friend

- SDR receiver

SDR (Software Defined Radio) can be used to receive:

1. AIS
2. FM AM radio
3. VHF marine radio channel
4. weatherfax
5. 433/868 MHz remote controls or sensors
6. ...

Most cheap USB SDR can work from 30 Mhz - 1.7 GHz. So they can't receive navtex at 490 kHz or 518 kHz.

The antenna must be different for each frequency range to get good results.

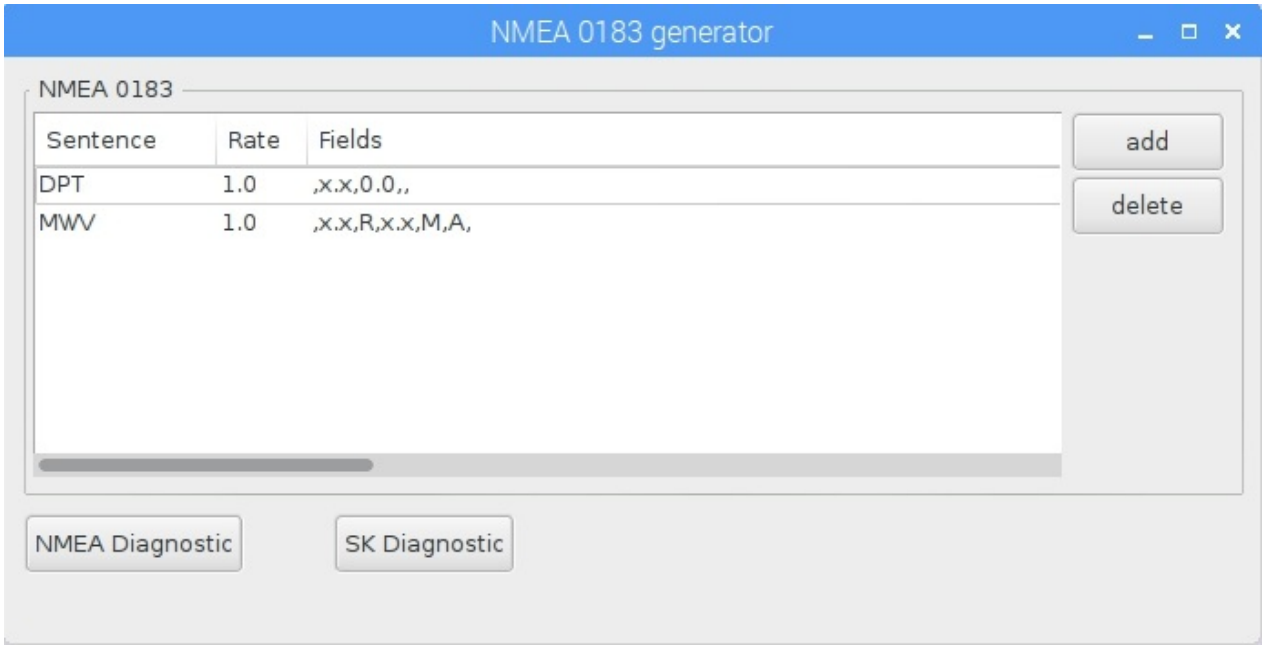
- Calculate

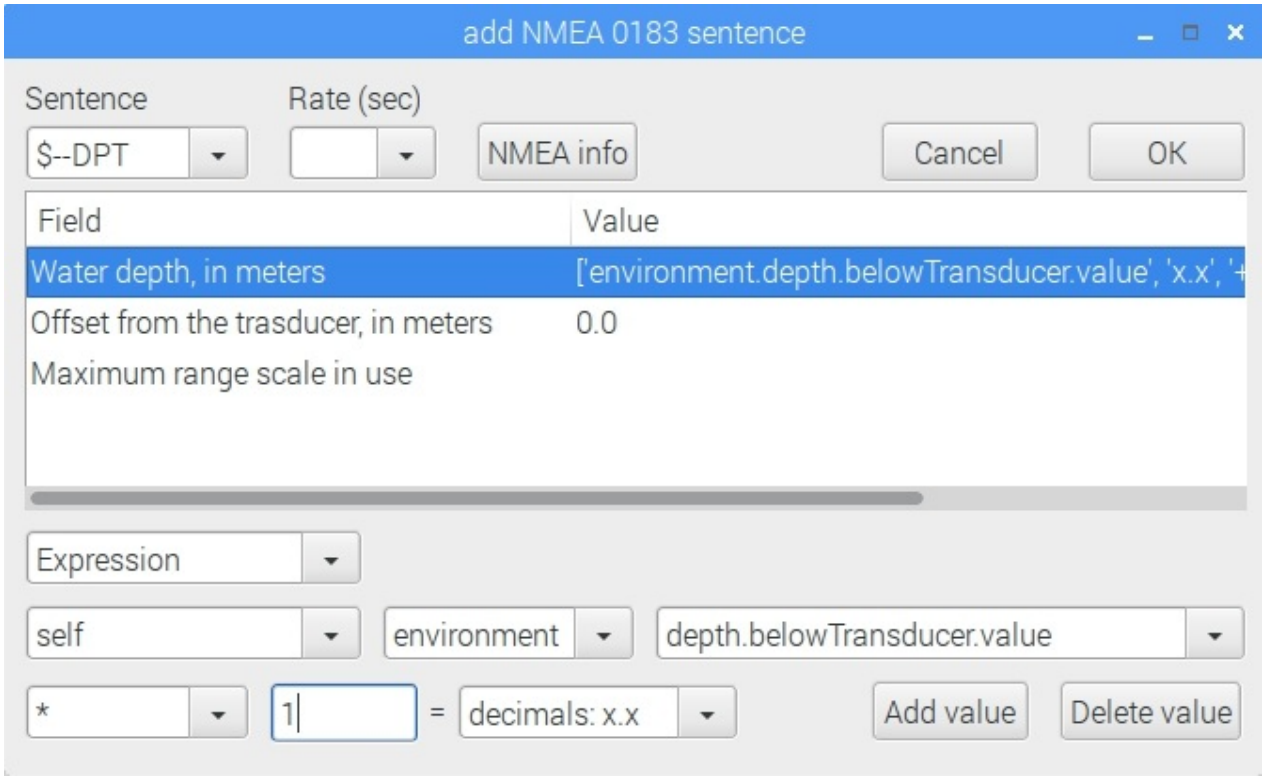
This is used to calculate the true wind from apparent wind

- NMEA 0183 generator

This tool allows you to create your own NMEA0183 sentences. You can select which SignalK value you want to take.

SignalK values are based on SI units (m,m²,m³,m/s,m³/s,J,rad,K,Pa,A,V,Hz,W) NMEA 0183 uses different units. That means you have to subtract for example 273.15 from a temperature value and to get radiant to degree you have to multiply by 57.29578.





To activate the changed configuration go to *SignalK* tab and click restart.

To check the result of the changes go to *NMEA 0183* tab -click on system -click on diagnostic.

- NMEA 2000 generator

- configureable tools

SDR receiver

DVB-T dongle (AIS)

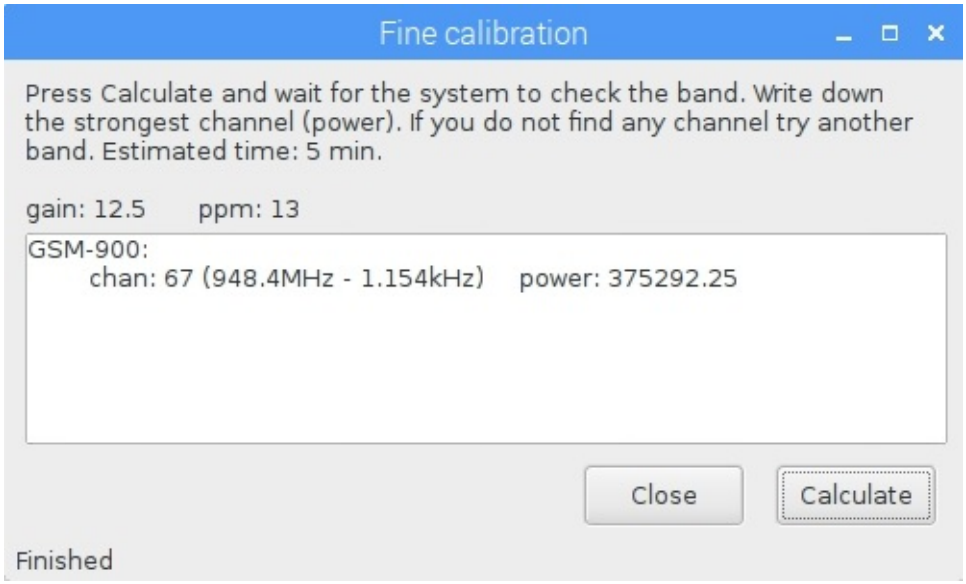
DVB-T dongles based on the Realtek RTL2832U chip and the new R820T2 tuner can work as a SDR AIS receiver.

A DVB-T dongle will need more power than the Raspberry Pi USB port can provide. You need to plug the dongle into a powered USB hub. Connecting and disconnecting can draw too much power and cause malfunction, try to do it when the system is off.

OpenPlotter is ready to get SDR AIS signal out of the box, you just have to calibrate to find **gain** and correction (**ppm**) values.

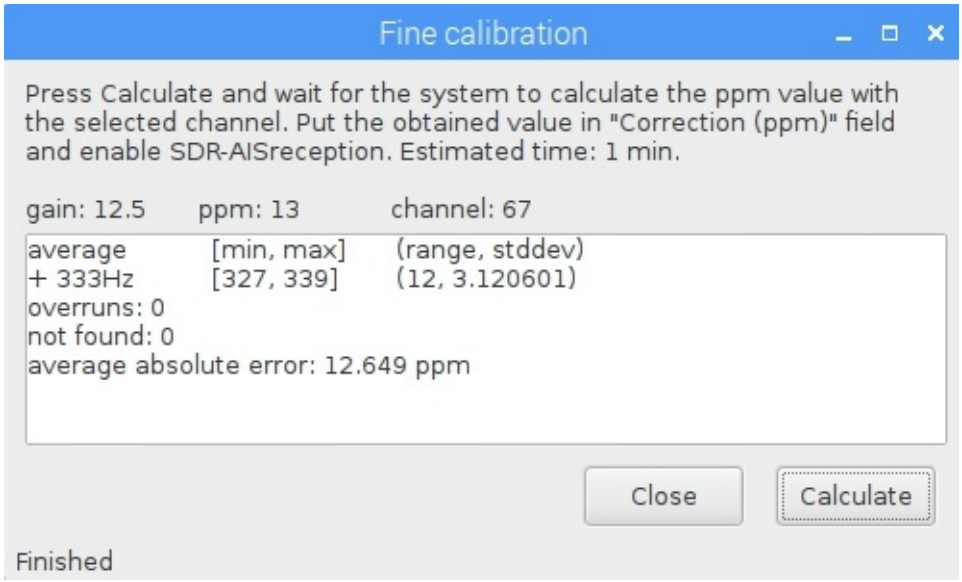
Fine calibration

It takes a few minutes after starting **Check band** and push **Calculate**.



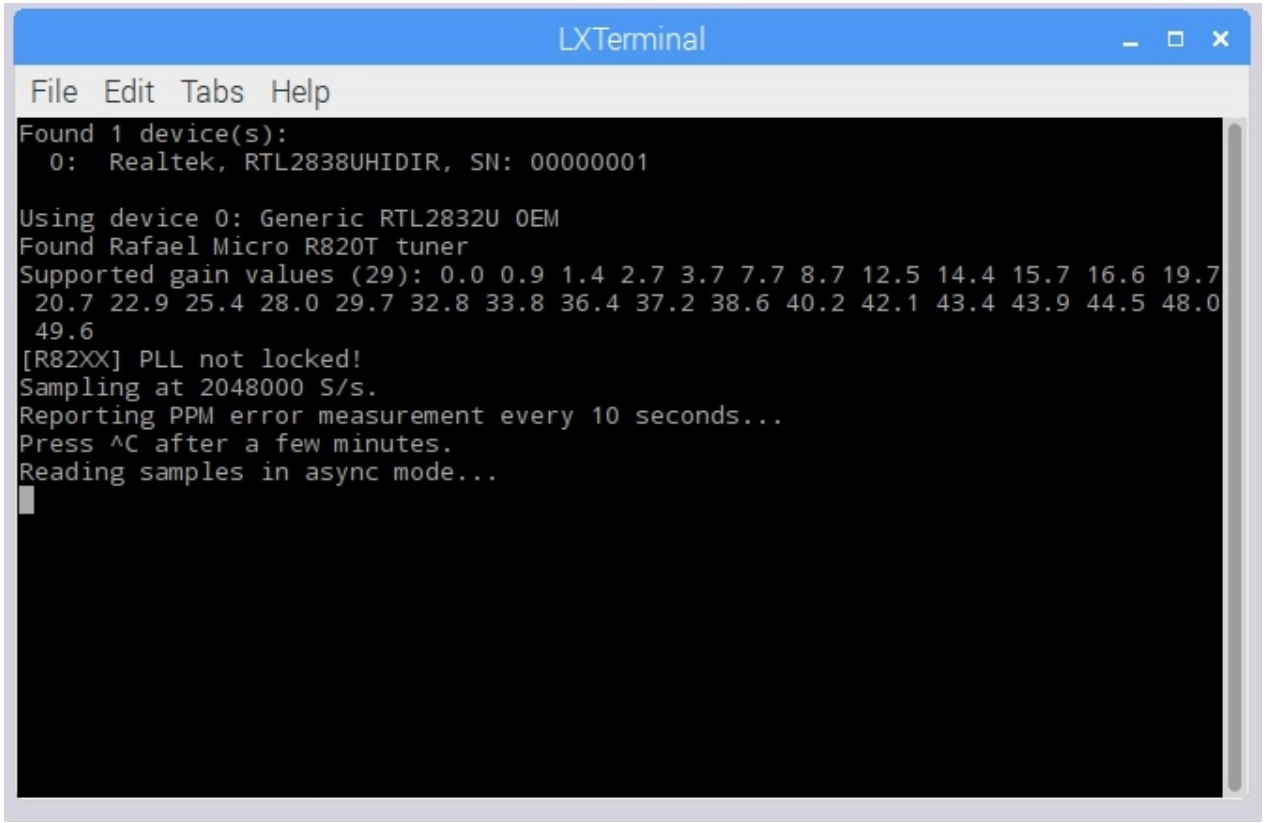
This is an example result.

If there are more "chan:" take the line with the highest power and put the "chan:" into **Channel** field and push **Fine calibration** and then **Calculate.**



For this example the field **Correction (ppm)** should be set to 13.

Now we look for the gain setting. Please push buttom **Calibration**.



The line **Supported gain values** is interesting for us. The max gain here is 49.6. Input this into the **Gain** field.

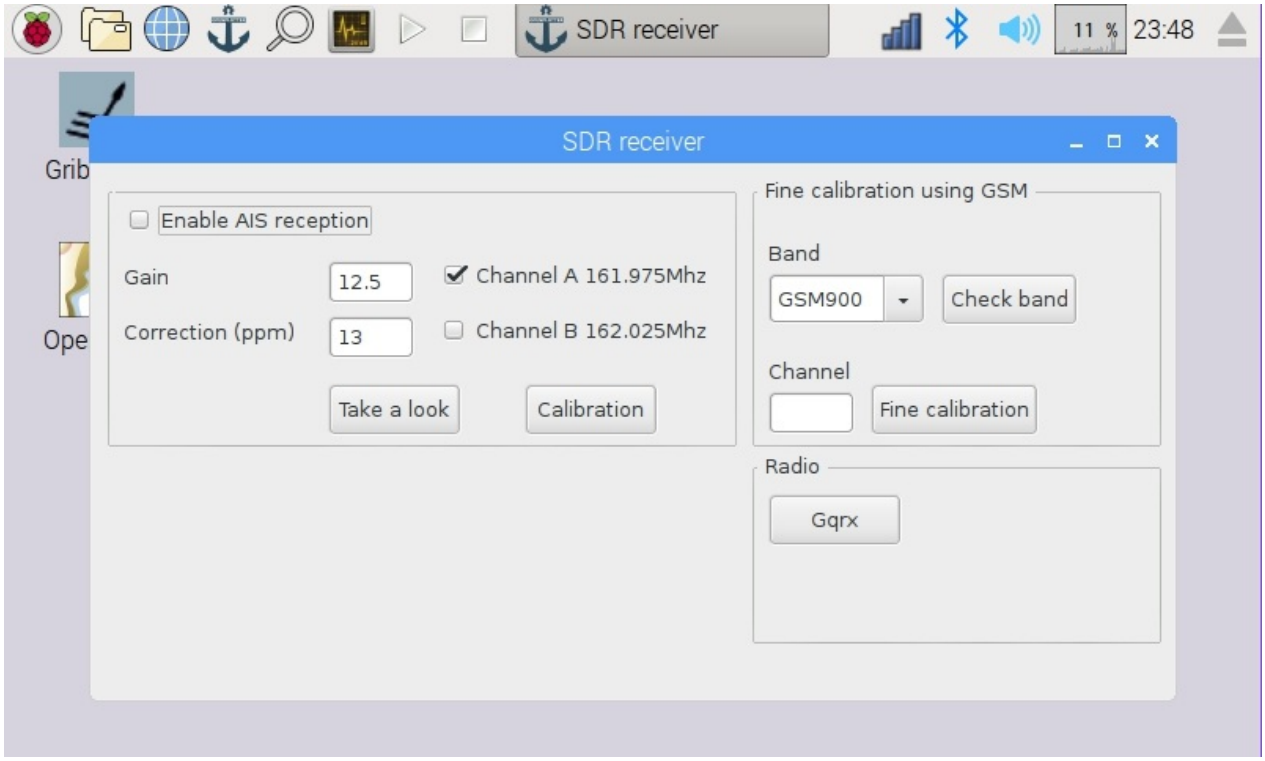
You can buy our DVB-T dongle and we can calibrate it for you and include a note with the gain and ppm values:

http://shop.sailoog.com/en/4-usb-sdr-ais-receiver.html

or you can follow this detailed guide:

http://sailoog.dozuki.com/Guide/Connecting+and+calibrating+SDR-AIS+dongles/3

AIS - receiving



Once you have found your **gain** and **ppm** value, select ***Enable AIS reception***.

You do not need to enable the rtlsdr plugin in OpenCPN. If you want to use that plugin you must disable SDR AIS reception in OpenPlotter.

Antenna

Although you can get to receive some boat with the supplied mini antenna, it is not enough for optimal reception of AIS frequencies. Any VHF antenna with the appropriate connector adapter will work fine. The antenna connector type of the dongle is female MCX.

Some home-made antennas:

http://sdrformariners.blogspot.com.es/p/blog-page.html

http://nmearouter.com/docs/ais/aerial.html

https://www.youtube.com/watch?v=SdEglNHyHB4

gqrx

gqrx is an application which also use the rtl2832u to receive something.

This gqrx-2.6-rpi3-1 version only supports RPI 3!

You can listen to:

- Radio in different waves
- vhf radio
- data

Be sure that the rtl2832u isn't used by openplotter or any other application. **It does consume much processor power. Other processes could be negative affected.**

Settings to begin with:

- configure I/O devices Input rate 240000
- Receiver Option Mode WFM (mono) FM radio
- Receiver Option Narrow FM FM marine radio
- FFT Settings Rate 5 fps 0 fps for low processor load
- Audio Gain 24.8 dB

wireless temperature sensors

advanced alpha features for experts

433 MHz temperature and humidity sensors known from wireless home weatherstations can be connected via rtl2832u to openplotter.

To translate this sensors to signalk use the python script tools\/rtl_433SK.py.

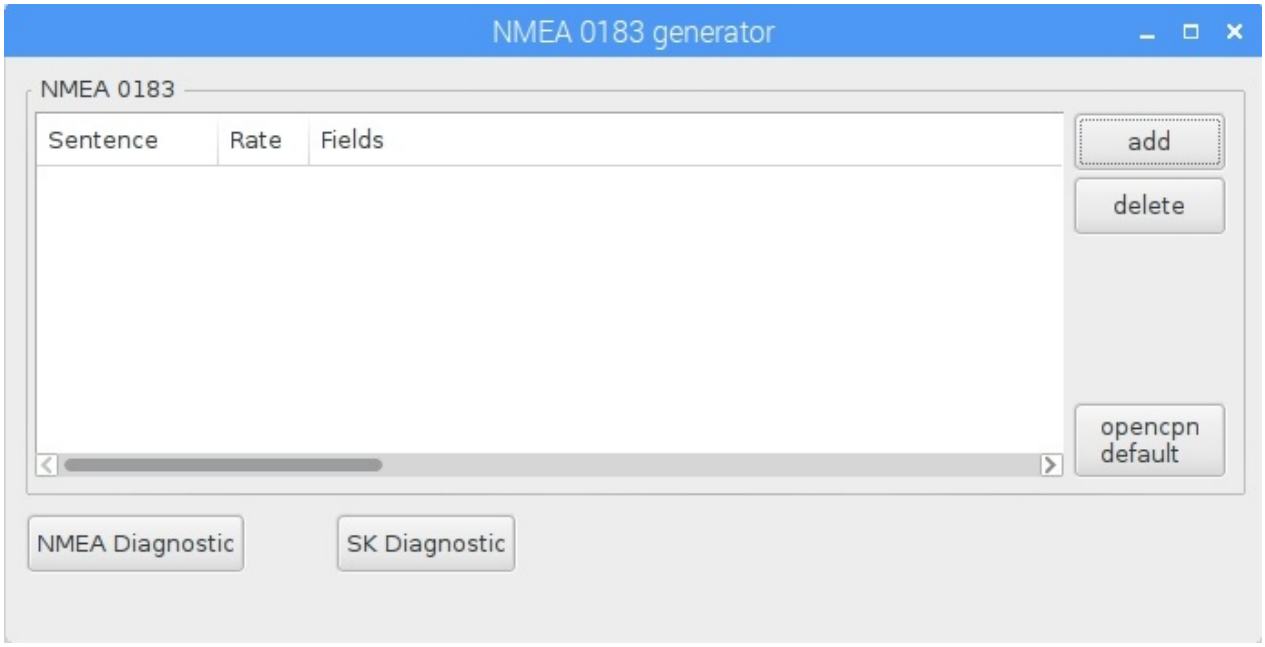
It can be integrated as tool. Add "[['433 Temp', 'get wireless temp', 'rtl_433SK.py', '0']]" this behind "py=" in openplotter.conf section [TOOLS].

NMEA 0183 generator

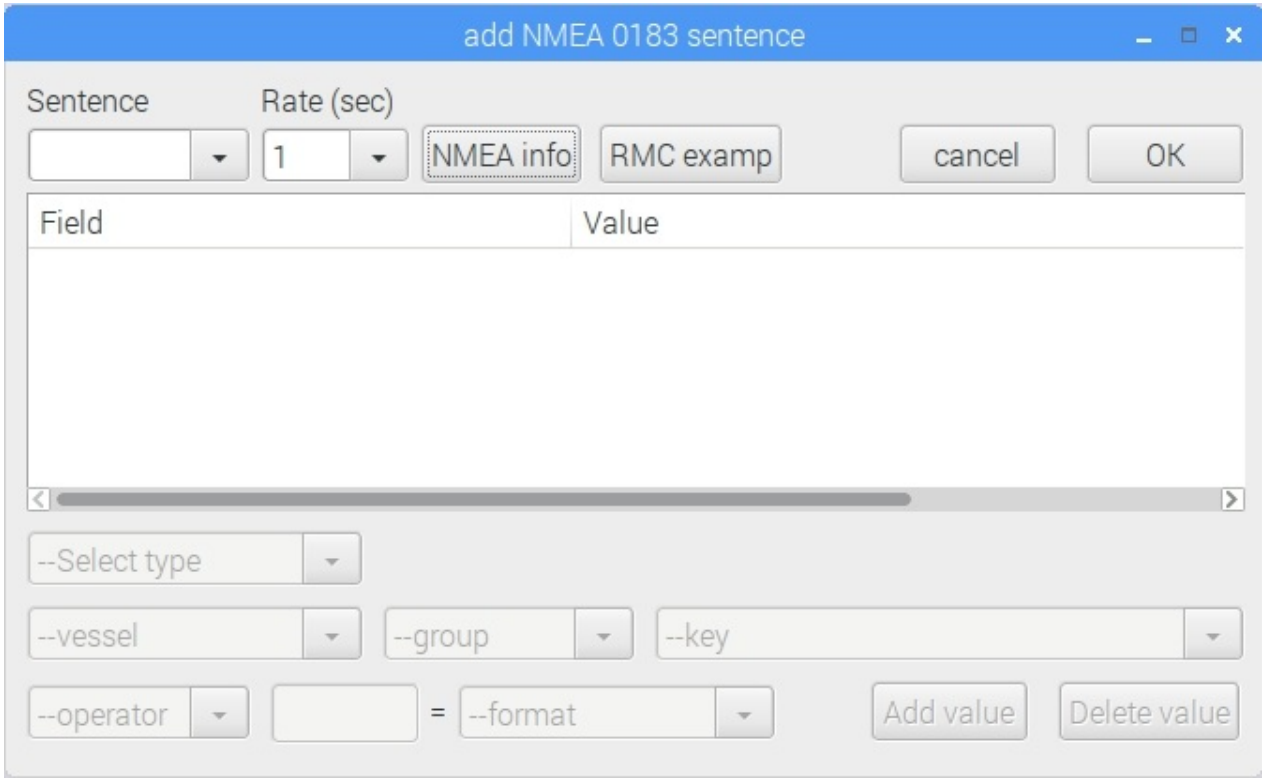
The generator is able to generate nmea sentences from SignalK values.

For example you have the value of the water temperature "environment.water.temperature" and you want to generate a NMEA 0183 sentence.

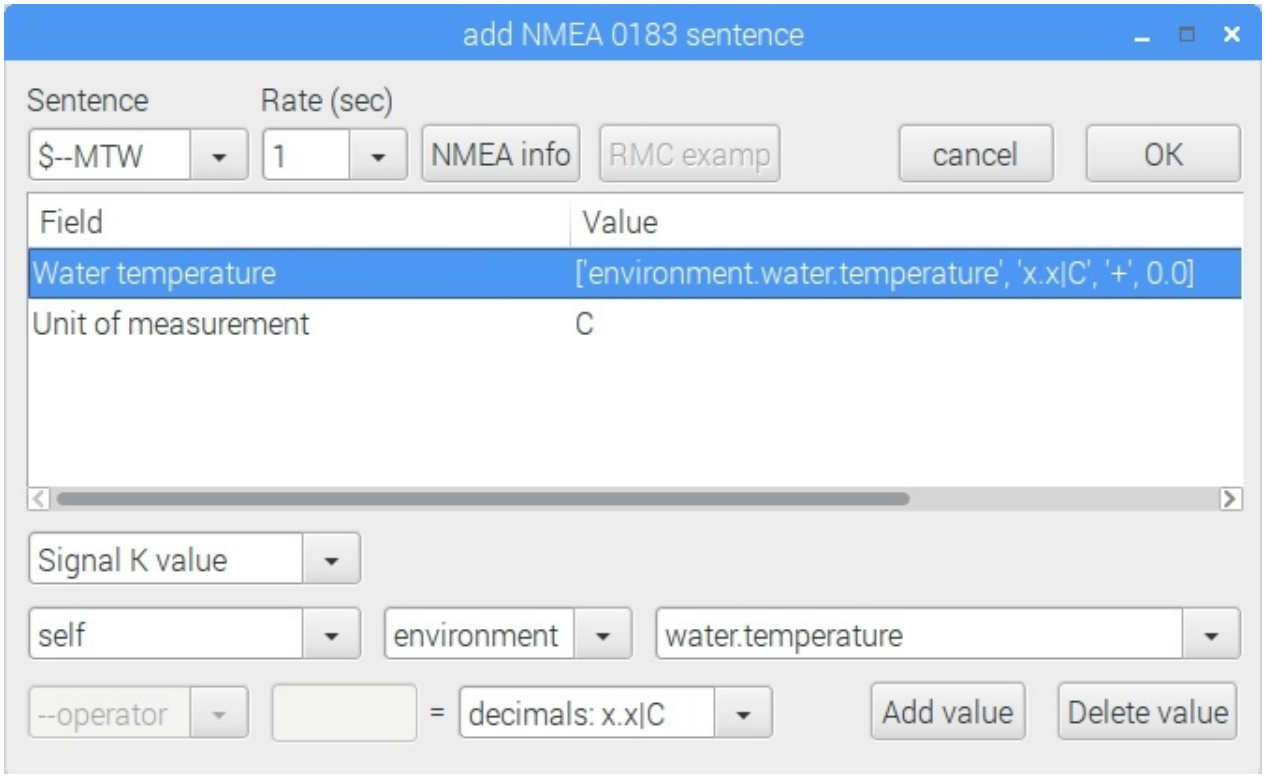
Open the window "NMEA 0183 generator" by selecting Tools->NMEA 0183 generator



Press "add"



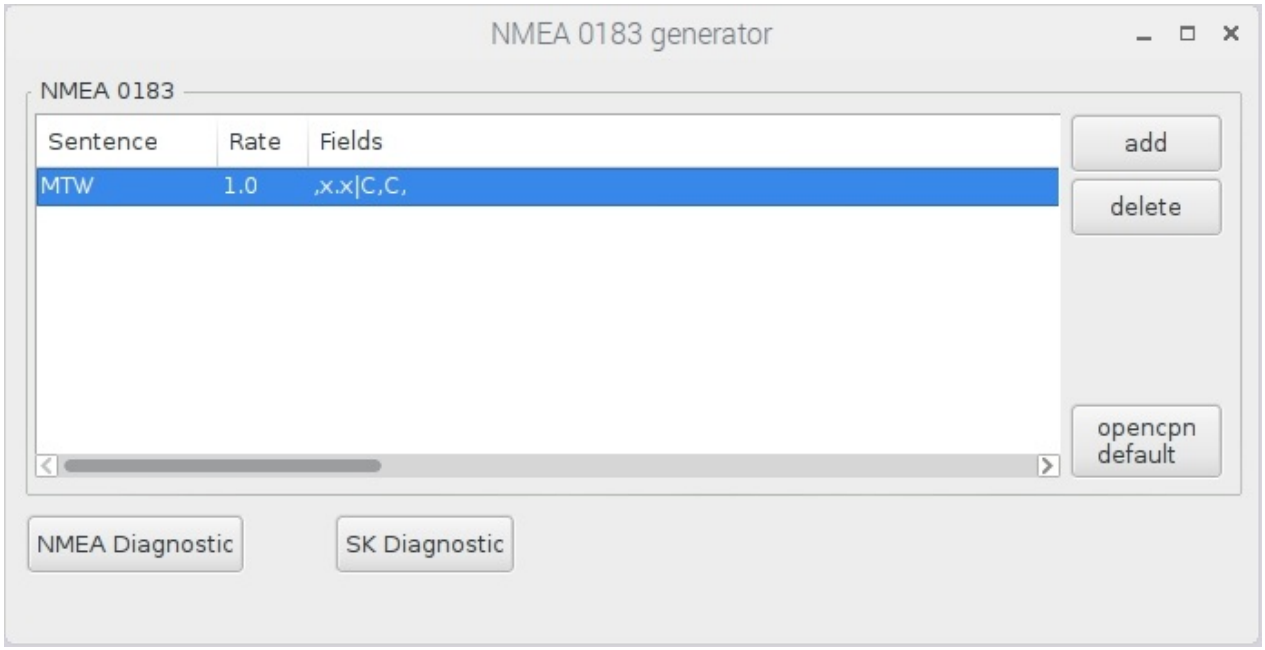
Select the type of NMEA "Sentence" you want to generate. For the example it is MTW. It should look like this:



Important: the SignalK value will be converted from Kelvin to Celsius.

Other values can also be converted from SignalK to NMEA 0183.

When you press OK, you get back to this window.



Opencpn sentences

To create the standard sentences for opencpn press "opencpn default" (It does only work if the list is empty!)

These SignalK values will be used:

"navigation.headingMagnetic"

"navigation.attitude.roll"

"navigation.attitude.pitch"

"environment.outside.pressure"

"environment.outside.temperature"

"environment.water.temperature"

Use Diagnostic

To see/debug the result of your generated sentence go to NMEA0183 tab.

Select the line system or opencpn then click the diagnostic button.

The generated sentence should be send in the time interval (Rate) you selected.

An alternative is to

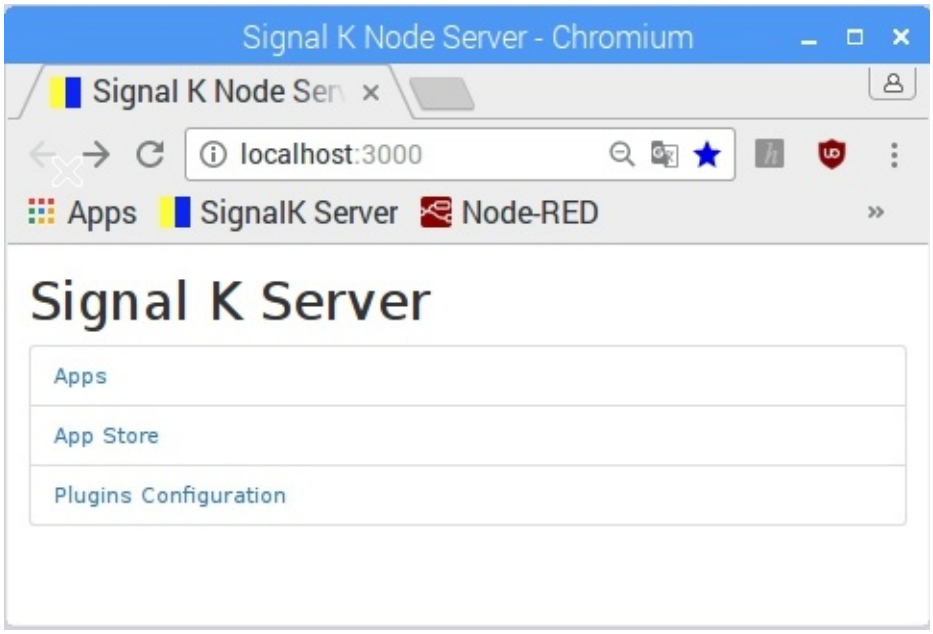
Generate NMEA0183 sentences from SignalK-server

Enter the SignalK Plugins from a browser by:

openplotter.local:3000

localhost:3000

10.10.10.1:3000



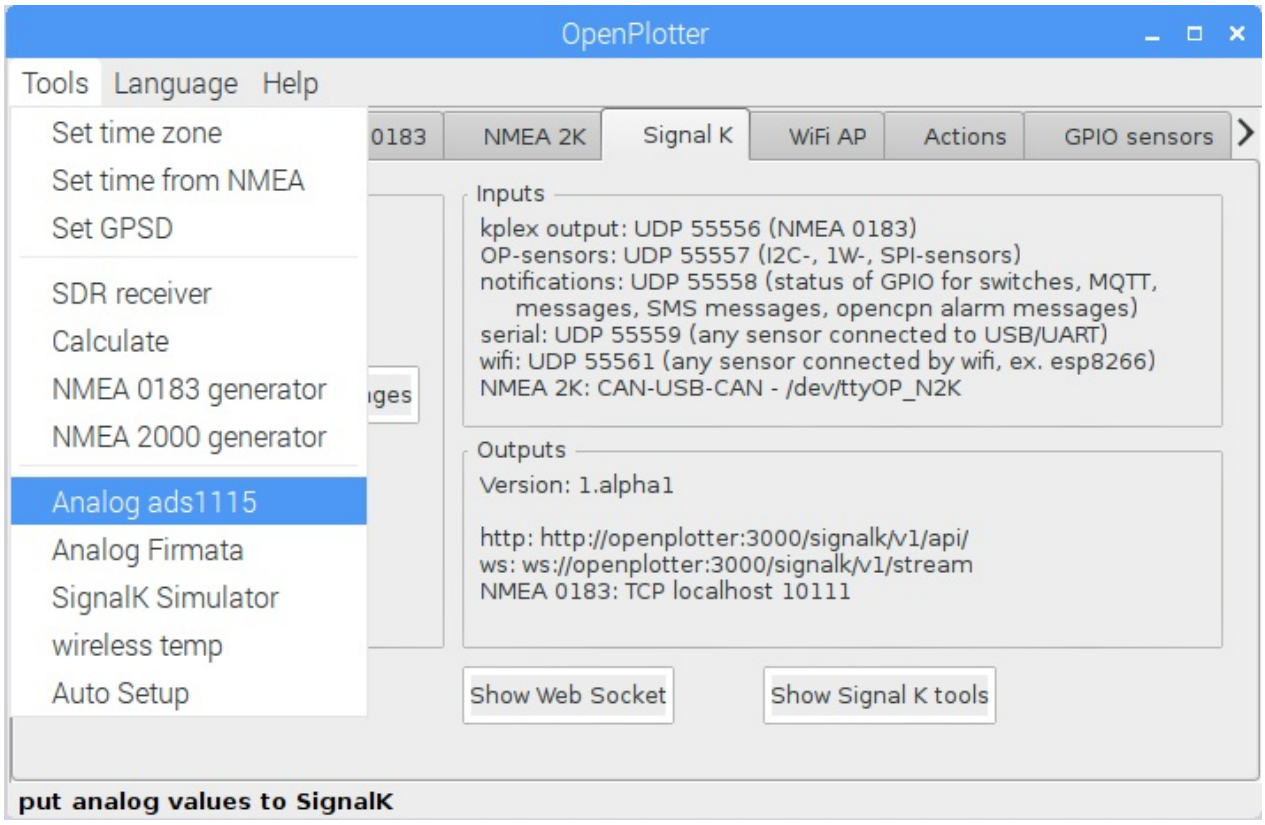
Signal K Server Plugin Configuration

- Ais Reporter
- Convert Signal K to NMEA0183
- Edit Zones
- Set System Time

Here you activate and select some NMEA0183 sentences. The performace is better than the NMEA 0183 generator from OpenPlotter but you have no chance to use other values, units, corrections or calculations.

Standalone tools

OpenPlotter has a chance to integrate independent programs which can share some of the OpenPlotter features. This was done to be open for add-ons done by advanced users and to keep OpenPlotter simple for newcomers.



There are implemented some standalone tools in OpenPlotter core maintaned by developers and a demo tool to make it easier to start. You can look at the code of these tools and build your own app/addon.

Adding a tool

Openplotter is able to work with external Python scripts.

There is a demo script in *~/.openplotter/tools/demo_tool.py*

To enable the demo tool open the OpenPlotter configuration file in *~/.openplotter/openplotter.conf*. Go to section [TOOLS] and you will see something like this:

```
[TOOLS]
py = [['Analog ads1115', 'put analog values to SignalK', 'analog_ads1115.py', '0'], ['Analog Firmata', 'put analog va
lues to SignalK', 'oppymata.py', '0'], ['SignalK Simulator', 'change values with sliders and send values to SignalK',
 'SK-simulator.py', '0'], ['Auto Setup', 'configure basic system', 'autosetup_tty.py', '0']]
```

The syntax of this array is:

```
[TOOLS]
py = [['title', 'description', 'file.py', 'startup'], [...], [...], [...]]
```

- **title** is the name you want to have in OpenPlotter Tools menu.
- **description** is a short sentence describing your app.
- **file.py** is the name of the file where your app lives.
- **startup** will be '1' if you want your app to start at startup or '0' if not.

To enable the demo tool you have to add this array to the array *py* in the [TOOLS] section of *~/.openplotter/openplotter.conf* file:

```
['Demo tool', 'You can use this app to start your own apps', 'demo_tool.py', '0']
```

Save the file and go to the OpenPlotter Tools menu. You should see now the option "Demo tool".

Once selected, a new window will be open with these options:

- **Settings**. If you have defined a config file for your app, it will be open with a text editor.
- **Start**. This option will start your app.
- **Stop**. This option will stop your app.
- **Cancel**. This option will close this window.

If you want to add your own tools you have just to create a python script in *~/.openplotter/tools* folder and edit the [TOOLS] section in *~/.openplotter/openplotter.conf* file just the same way you did for the demo tool.

See the source of the demo tool to know how to manage config files, manage the OpenPlotter config file, interact with Signal K server or access to OpenPlotter classes.

Let us know if you make any tool that we can add to OpenPlotter core.

Analog ads1115

Is an adc with:

- 4 inputs
- 16 bit
- selctable gain
- selectable sample rate

When selected settings the configuration file will be opened.

[ADS1115_0] active = 1
gain = 0
samples = 6
ohmmeter = 0
fixed_resistor = 0
high_voltage = 3000
voltage_divider = 0
upper_resistance = 1000
lower_resistance = 1000
adjust = 0
sk_name = tanks.fuel.left.currentLevel
adjust_points = [[100.0,0.0],[5000.0,90.0],[9000.0,100.0]]

If a channel is used active must be set to 1

The gain setting can be changed with gain.

gain = 0 +/- 6.144V

gain = 1 +/- 4.096V

gain = 2 +/- 2.048V

gain = 3 +/- 1.024V

gain = 4 +/- 0.512V

gain = 5 +/- 0.256V

samples setting can be changed with samples (see table).

samples 0 8 samples per second

samples 1 16 samples per second

samples 2 32 samples per second

samples 3 64 samples per second

samples 4 128 samples per second

samples 5 250 samples per second

samples 6 475 samples per second

samples 7 860 samples per second

ohmmeter can be set to 1 to measure the resistance of a sensor.

fixed_resistor must be set to the resistance of the fixed resistor and

high_voltage is the voltage you put on the two resistors in serial.

The resistance you want to measure is connected on one side to ground and

on the other side to the fixed resistor and analog input of the adc.

The other side of the fixed resistor is connected to + pol (for example 3.3V (high_voltage))

voltage_divider can be set to 1 if you want to have Volt values.

If you want to measure 12 V with a adc witch is only capable of max 3.3V,

you need a voltage divider in form of two resistors. (50k+10k would put the voltage from 12V to 2V)

upper_resistance has to be set to 50000 and lower resistance to 10000 for this example.

Look at the picture on http://forum.arduino.cc/index.php?topic=214930.0

adjust is only for adjusting the offset

The value is send to SignalK on the name you set in sk_name.

adjust_points

You can put in some points of a curve.

Between this point the value is calculated linear.

This is the easiest way to get good results.

It can be used in combination with voltage_devider or ohmmeter.

Analog Firmata

An Arduino, Teensy or STM32 boards which can be programed by Arduino IDE or which have a library for standard firmata are useable.

The communication is done with USB port or an USB to serial converter. We recommend to use an isolator when using analog signals (usb to usb isolator or uart to USB isolator or use an ADUM1201 and a simple uart to serial converter).

When firmata standard is installed. Ad the new USB port in USB manager. **The name of the port has to be ttyOP-FIRM!**

Next step is edditing the configuration file.

Analog Firmata can be started. Best way to see if connecting works is to start openplotter from a terminal session.

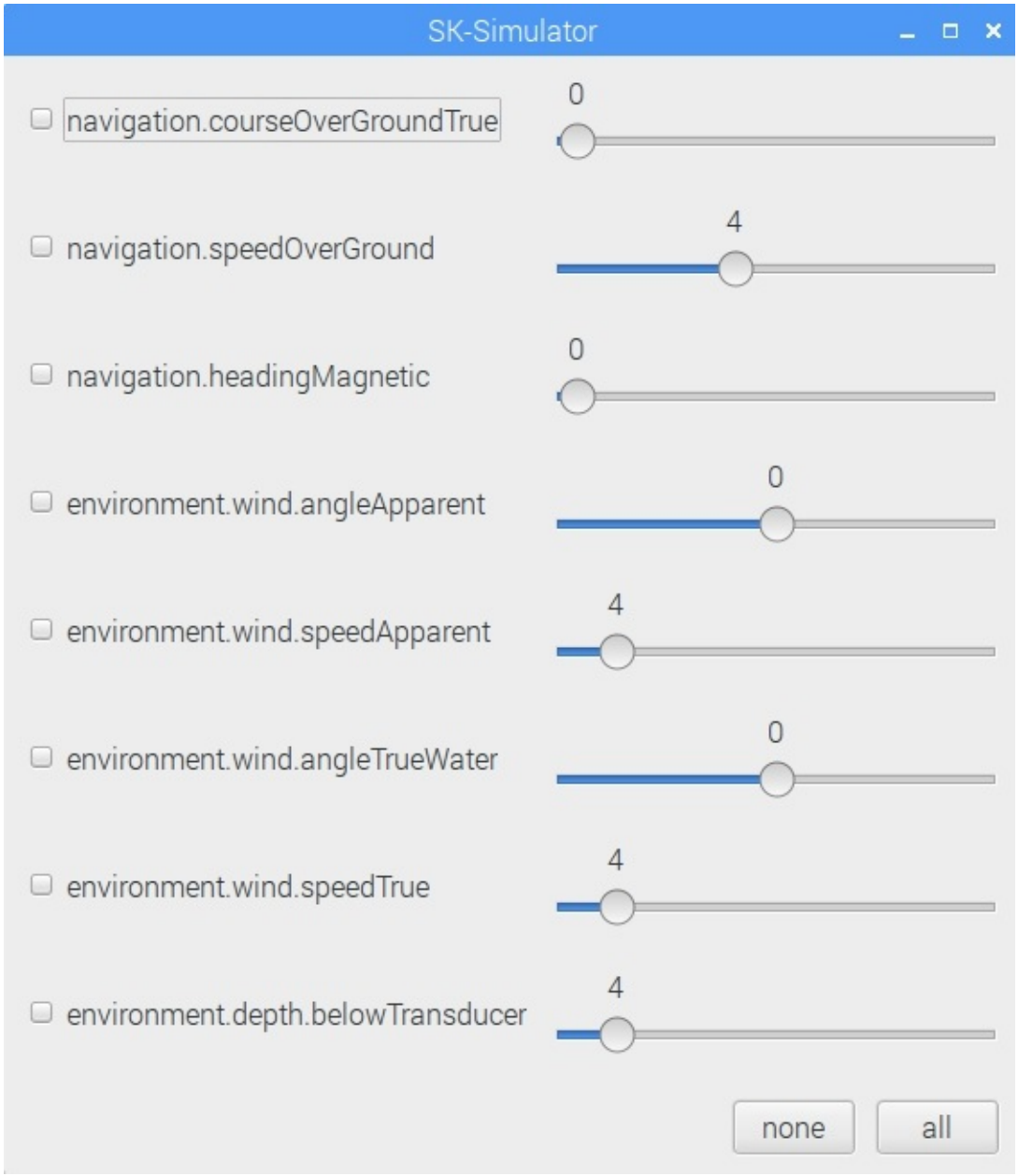
The start of pymata takes some seconds.

The configuration file is the same one which is used by ads1115.
There are up to 16 sections ([FIRMATA_0]..[FIRMATA_15])

Example:

[FIRMATA_1]
sk_name = tanks.fuel.right.currentLevel
adjust_points = [[52.0,0.0],[522.0,25.0],[708.0,50.0],[913.0,100],[1024.0,100.01]]

SignalK Simulator



Select the values you want. You can see your simulated values on SignalK **Diagnostic**.

You can use other SignalK values by editing the file _SK-simulator.conf. _It can be accessed by the setting buttom.

[main]

item_0 = [0, 'navigation.courseOverGroundTrue', 0, 0, 360, 0.0174533, 0]

item_1 = [1, 'navigation.speedOverGround', 4, 0, 10, 0.5144441, 0]

item_2 = [2, 'navigation.headingMagnetic', 0, 0, 360, 0.0174533, 0]

item_3 = [3, 'environment.wind.angleApparent', 0, -180, 180, 0.0174533, 0]

item_4 = [4, 'environment.wind.speedApparent', 4, 0, 40, 0.5144441, 0]

item_5 = [5, 'environment.wind.angleTrueWater', 0, -180, 180, 0.0174533, 0]

item_6 = [6, 'environment.wind.speedTrue', 4, 0, 40, 0.5144441, 0]

item_7 = [7, 'environment.depth.belowTransducer', 4, 0, 40, 1, 0]

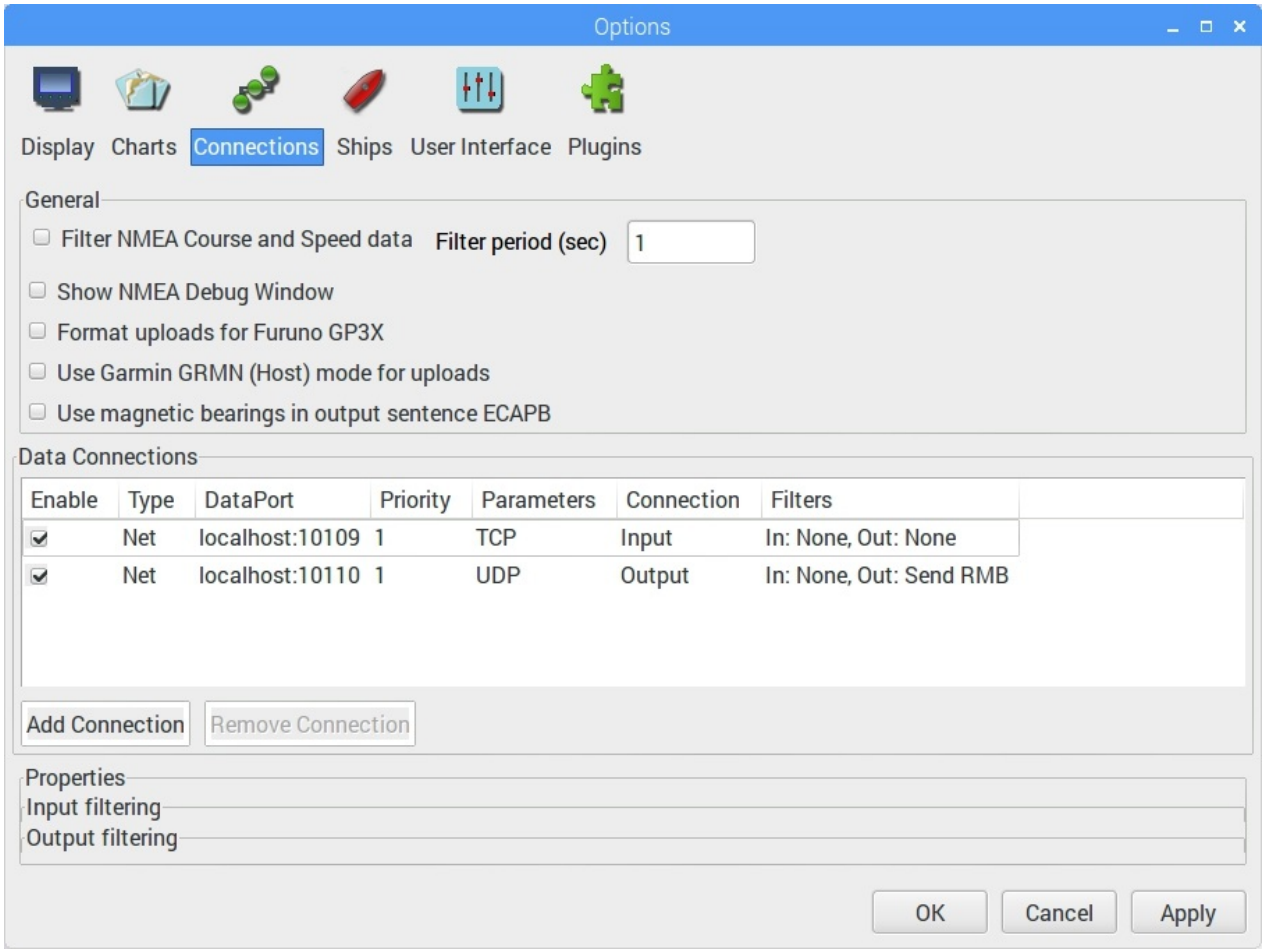
The numbers behind the SignalK string copied from first line (*0, 0, 360, 0.0174533, 0)*

are:

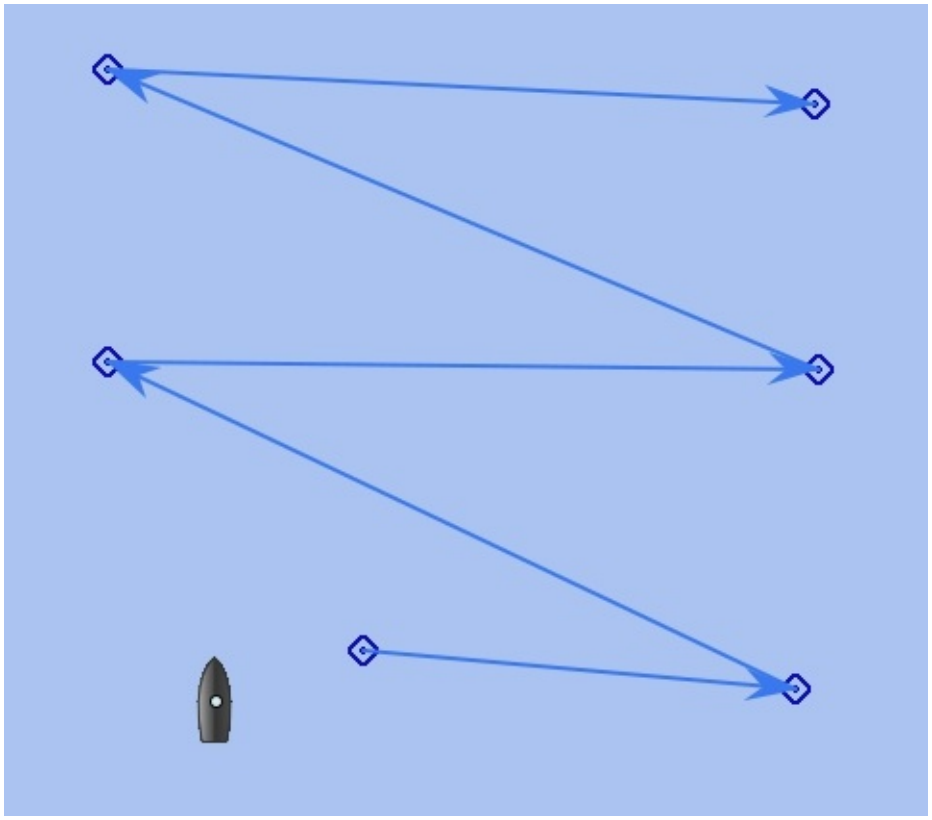
- default value: 0
- minimum value: 0
- maximum value: 360
- factor: 0.0174533 (PI/180 for converting deg to rad)
- offset: 0 (interesting for Fahrenheit)

under construction

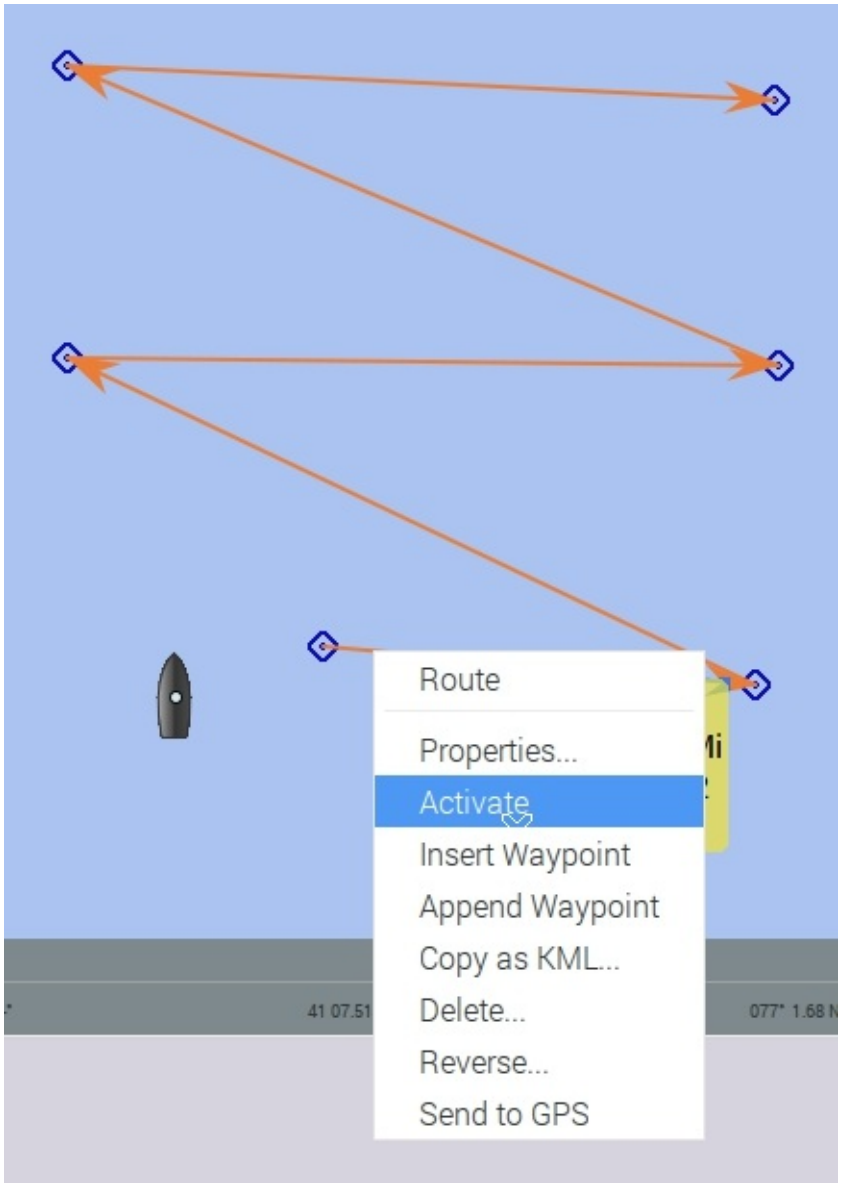
Chartplotter (OpenCPN)

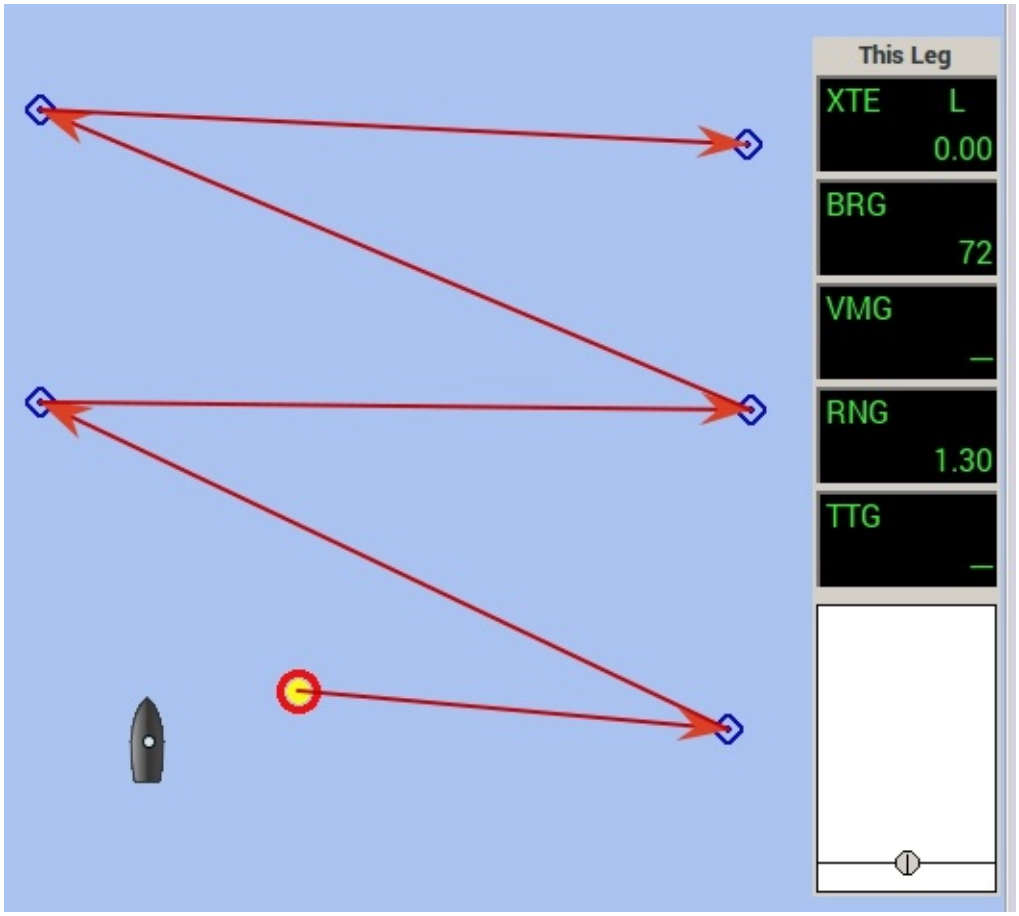


This are the standard connection settings of opencpn to connect to OpenPlotter.



You can activate the route with right mouse button.


Route
Properties...
Activate
Insert Waypoint
Append Waypoint
Copy as KML...
Delete...
Reverse...
Send to GPS



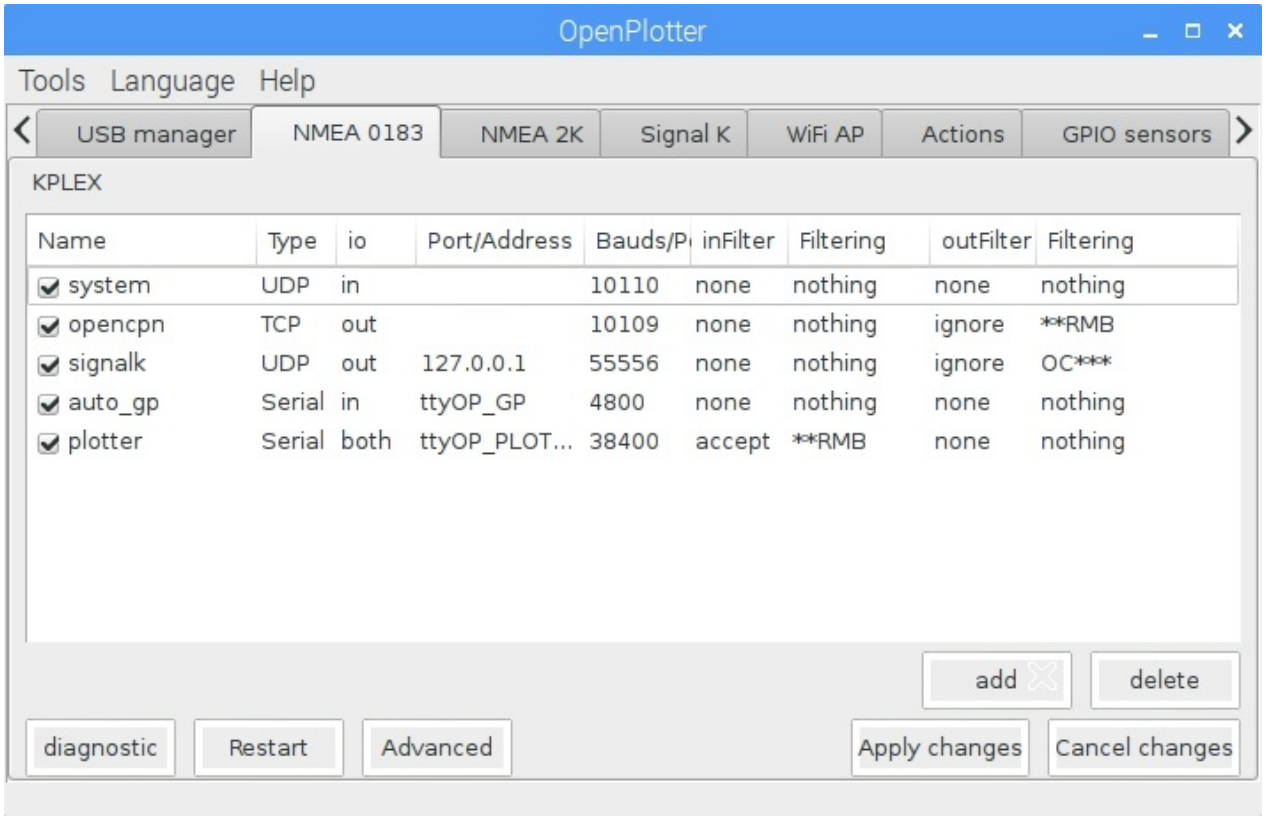
You see a yellow red point blinking. Opencpn sends an NMEA0183 RMB sentence.

If the autopilot has been setup correctly. You can start autopilot to navigate to next waypoint.

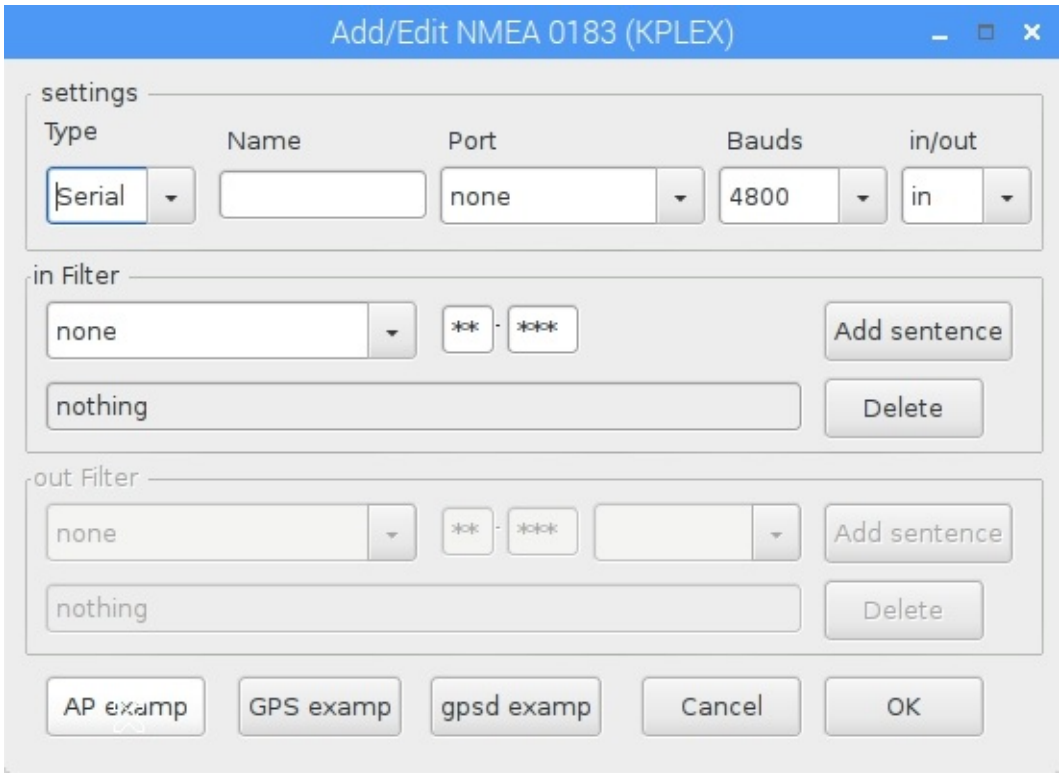
Go on with Sending data to autopilot.

Sending data to the autopilot

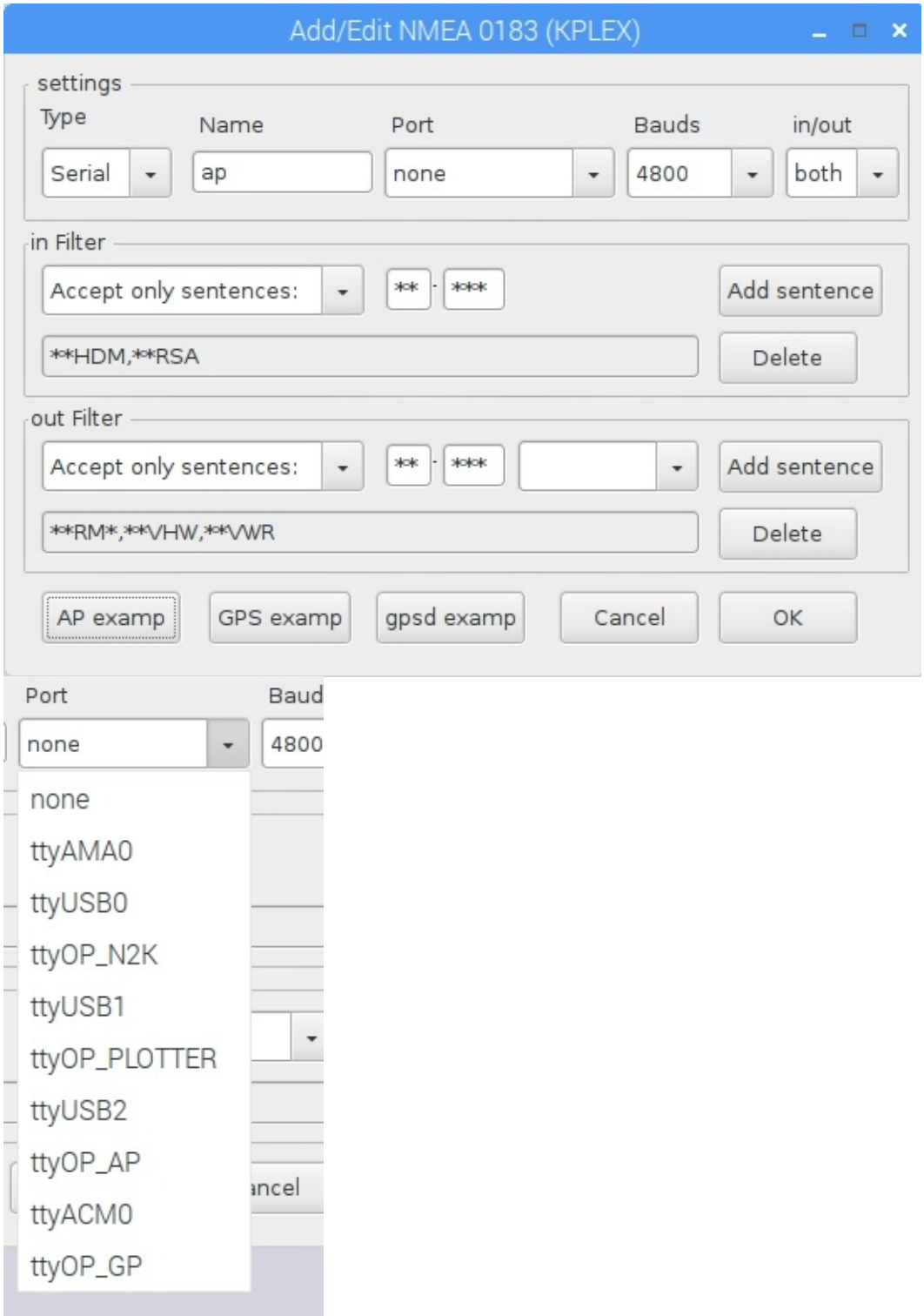
The autopilot has to be setup



Click on **add** (or double click on auto_ap).



Click on **AP examp** to load default settings for autopilot.



Select port ttyOP_AP

in Filter: What sentences are allowed to be received from the autopilot to OpenPlotter/kplex.

Most autopilots have there own fluxgate compass and rudder angle sensor. These data can be used from OpenPlotter.

More important is the

out Filter: What sentences are allowed to be send from OpenPlotter/kplex to the autopilot.

It is important to filter the sentences to reduce the transfer volume. NMEA0183 autopilots work normally with 4800 bauds. There's only space for a few sentences.

Which sentences are important?

- RMB (way to selected waypoint send from a chartplotter)
- RMC (minimum GPS data)
- VHW (waterspeed and heading)
- VWR (wind)
- sometimes APA and APB

License

OpenPlotter code

OpenPlotter is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 2 of the License, or any later version.

OpenPlotter is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with OpenPlotter. If not, see http://www.gnu.org/licenses/.

Documentation license

This Docuentation is licensed under a Creative Commons Attribution-ShareAlike 4.0 International License.

http://creativecommons.org/licenses/by-sa/4.0/

Credits

This chapter needs to be written/updated/translated

http://forum.openmarine.net/forumdisplay.php?fid=16

Documentation

Sailoog, e-sailing , Nicolas Janvier, Didier Barnouin, V36317, Dirk Wonning.

Software

OpenPlotter: http://www.sailoog.com

Chartplotter: http://opencpn.org/ocpn/

Weather forecast: http://www.zygrib.org

NMEA 0183 multiplexer: http://www.stripydog.com/kplex/index.html

NMEA 0183 Parser: http://github.com/Knio/pynmea2

IMU, pressure, temperature, humidity sensors: http://github.com/richards-tech/RTIMULib

RTL SDR: http://sdr.osmocom.org/trac/wiki/rtl-sdr

RTL SDR calibration: https://github.com/steve-m/kalibrate-rtl

AIS decoder: http://www.aishub.net/aisdecoder-via-sound-card.html

Python wrapper for librtlsdr: https://github.com/roger-/pyrtlsdr

Magnetic variation: http://github.com/cmweiss/geomag

Signal K: http://signalk.org

rtl_433: https://github.com/merbanan/rtl_433

canboat: https://github.com/canboat/canboat

gqrx: https://github.com/csete/gqrx

PyMata: https://github.com/MrYsLab/PyMata

node-red-contrib-freeboard: https://github.com/urbiworx/node-red-contrib-freeboard