<< Summary

NMEA multiplexing with the Raspberry PI

We will show a possibility to add data to an NMEA stream.
We will use the NMEA Console project as a starting point.

The idea is simple. It's relying on several points:

- You can use a Raspberry PI connected to your NMEA station to rebroadcast the data on HTTP, TCP, UDP, RMI, etc (see here).
- You can connect sensors on the Raspberry PI, and read from them data like air temperature, barometric pressure, etc. Those data are not always available from the NMEA station. And if they are - on can be - it is not always cheap...
- The NMEA Console now has the concept of user-exits, and the Console can run in headless mode.
- Turning the data read from the sensors into an NMEA string, and sending them to the already existing NMEA Listeners should not be a problem.

Turning the sensor data into NMEA string

No brainer. Done in the NMEA Parser (see project), class `ocss.nmea.parser.StringGenerator.java`, along with their reciprocal functions in `ocss.nmea.parser.StringParsers.java`.

Using the already existing NMEA Listeners

The Listeners we use are part of the NMEAContext, accessible from the Desktop. We need to use two different lists of listeners, `NMEAListener` and `NMEAReaderListener`, one is used to parse the incoming data and put them in the cache, the other to trigger the possible re-broadcasting (which we are specially interested in).

Flexibility

The number of sensors that can be plugged in the Raspberry PI is not limited, and growing.
A flexible approach is to use the concept of user-exits, already implemented in the Navigation Desktop. Those user-exits can be added on demand, removed, replaced, etc. Perfect for this kind of context.

The code

The user exit in made out of two classes.

- `olivsoftdesktopuserexits.Sensor2NMEA`, that implements the `DesktopUserExitInterface` interface.
- `olivsoftdesktopuserexits.rpisensor.AdafruitBMP180Reader`, that reads the BMP180 sensor.

Sensor2NMEA.java

```
1  package olivsoftdesktopuserexits;
2
3  import olivsoftdesktop.DesktopUserExitInterface;
4
5  import olivsoftdesktopuserexits.rpisensor.AdafruitBMP180Reader;
6
7  /**
```

```
 8   * Reads a sensor (BMP180) connected to the Raspberry PI
 9   * and turn the data into NMEA string to broadcast them.
```

Notice the lines 21 & 33.

AdafruitBMP180Reader

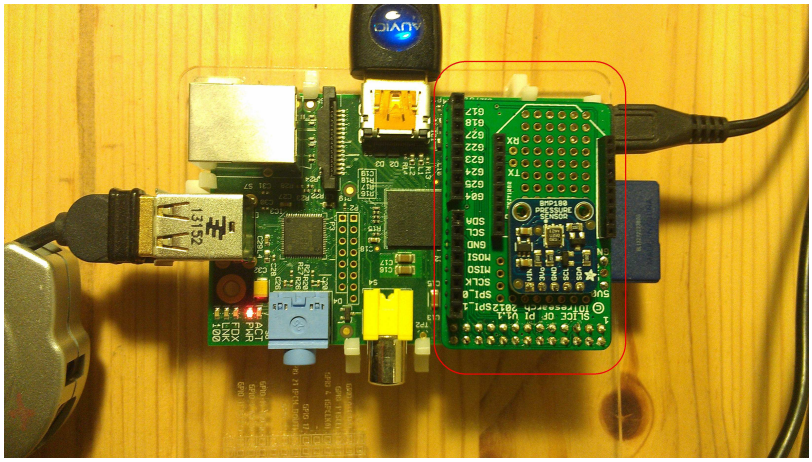
```
 1  package olivsoftdesktopuserexits.rpisensor;
 2
 3
 4  import com.pi4j.io.i2c.I2CBus;
 5  import com.pi4j.io.i2c.I2CDevice;
 6  import com.pi4j.io.i2c.I2CFactory;
 7
 8  import java.io.IOException;
 9
```

The way to deal with the sensor is the same as the one described in this document.
See how the data are turned into an NMEA Sentence on lines 416 & 417. The NMEA Strings are **MMB** & **MTA**. I know that **XDR** is recommended (and available in the NMEA Parser project), but still.
See how they are broadcasted to the listeners on lines 419 & 420, invoking the `broadcastNMEASentence` function, defined line 427.

To know how to compile, archive, and use the code as a user-exit in the NMEA Console, take a look at this document.

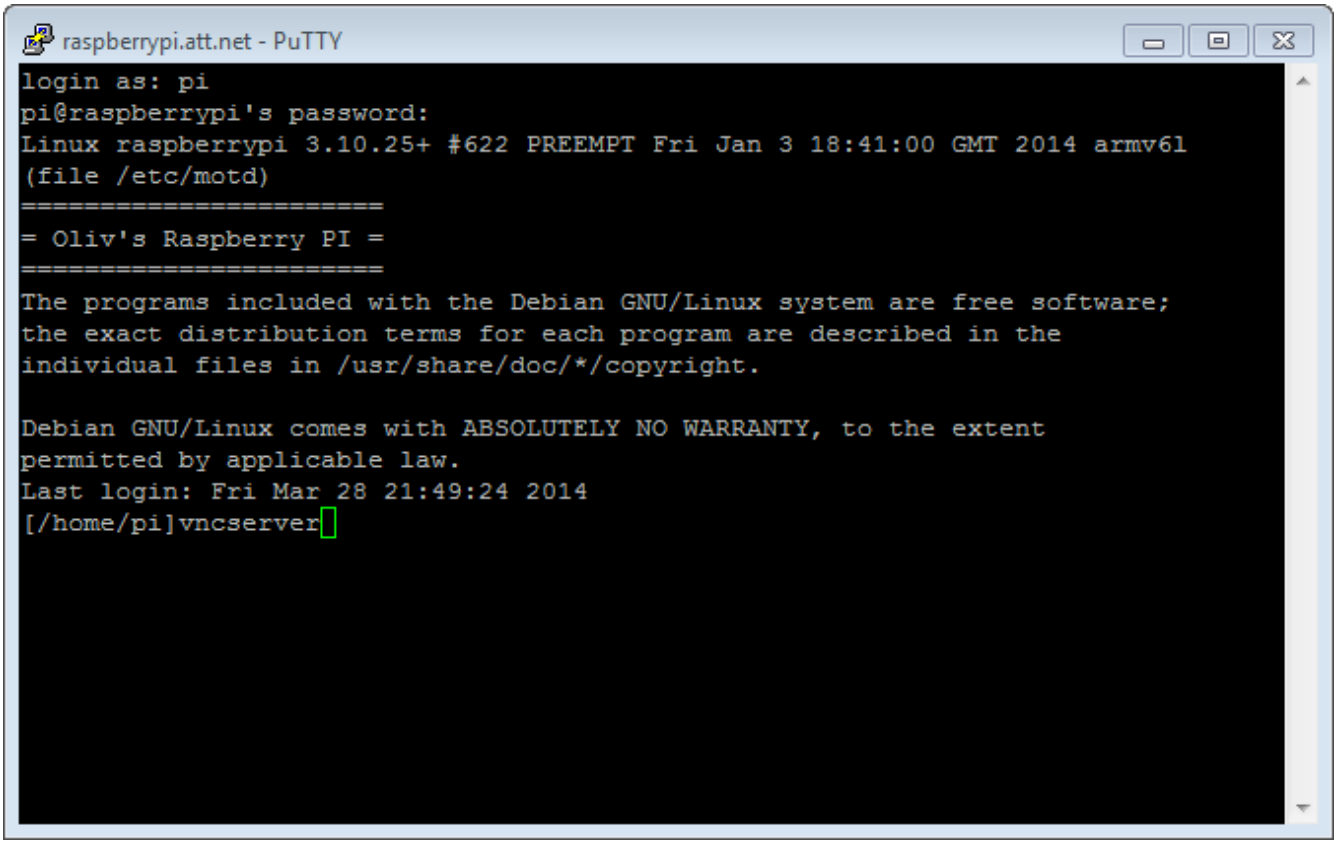
Demo

We show here how to integrate in the NMEA data flow the values coming from a BMP180 sensor, providing Air Temperature, Atmospheric Pressure, and Altitude (based on pressure), which we don't care about, for some reason...
We will turn the sensor data into NMEA **MBB** and **MTA** Strings.
To see how to process the signal from the BMP180, see here, for the code and details.

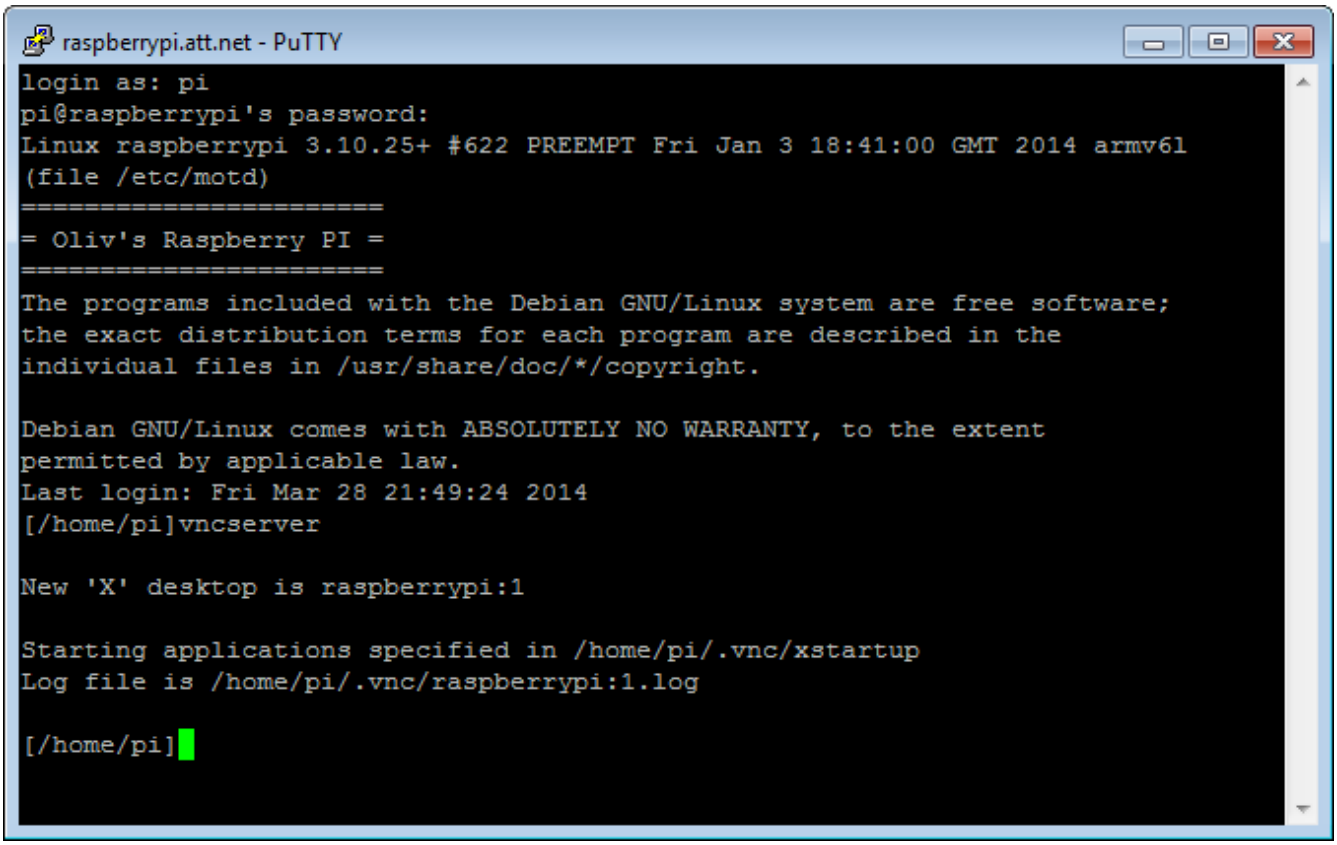


The Raspberry PI, with its BPM180 sensor, on a Slice of PI.

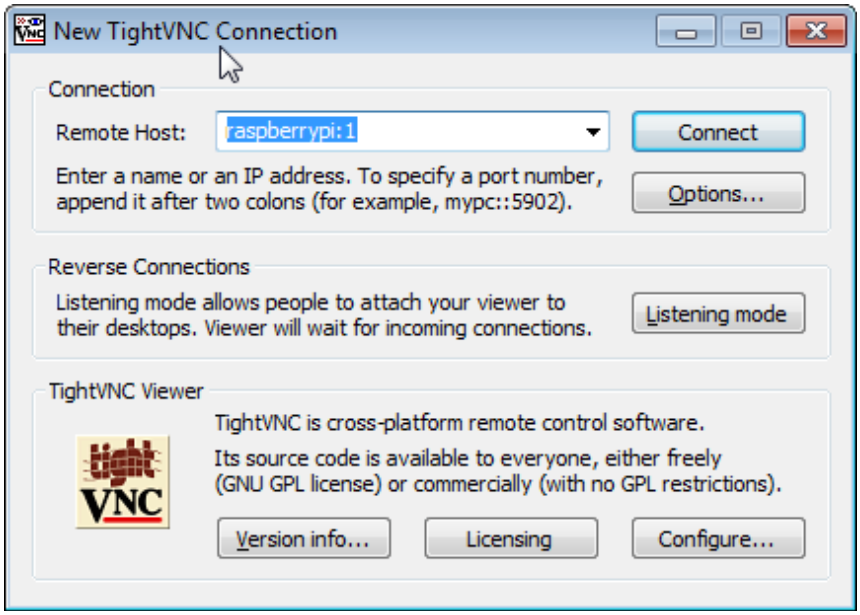
Note: After doing some tests, this is not the best location for the BMP180, it is too close to the CPU of the Raspberry PI, that generates heat. The temperature readings are impacted by the CPU temperature...



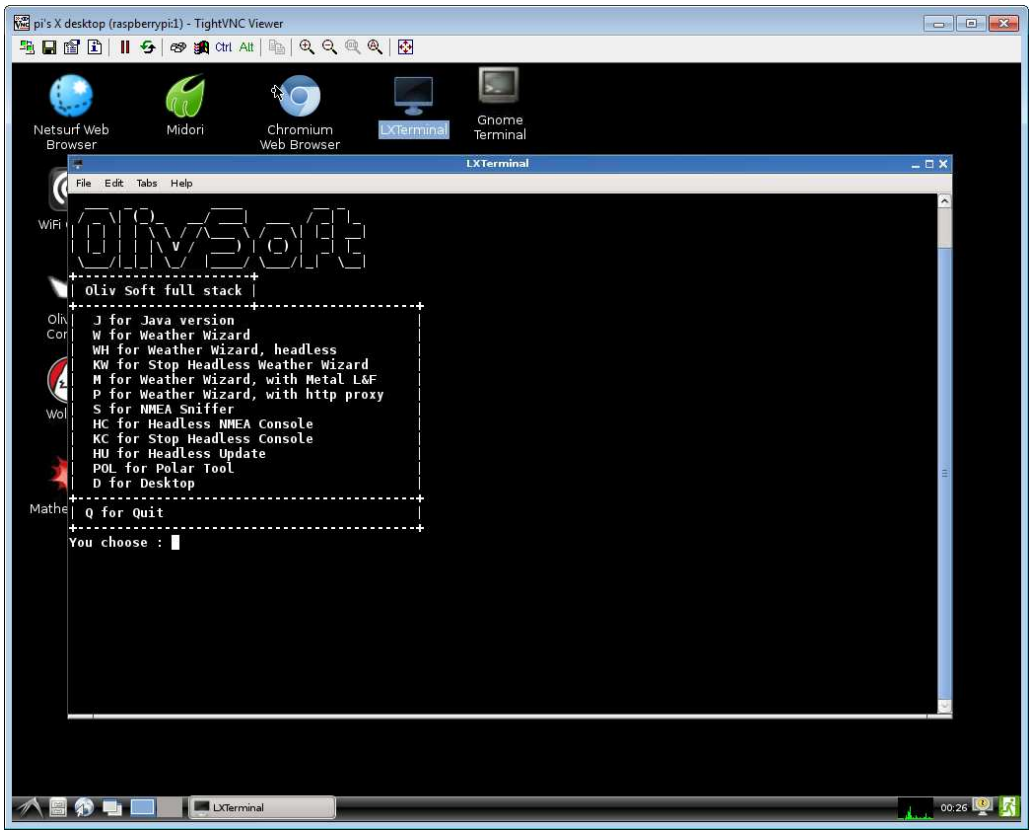
From another machine, connect on the RasPI using - for example - PuTTY, and start the vncserver.



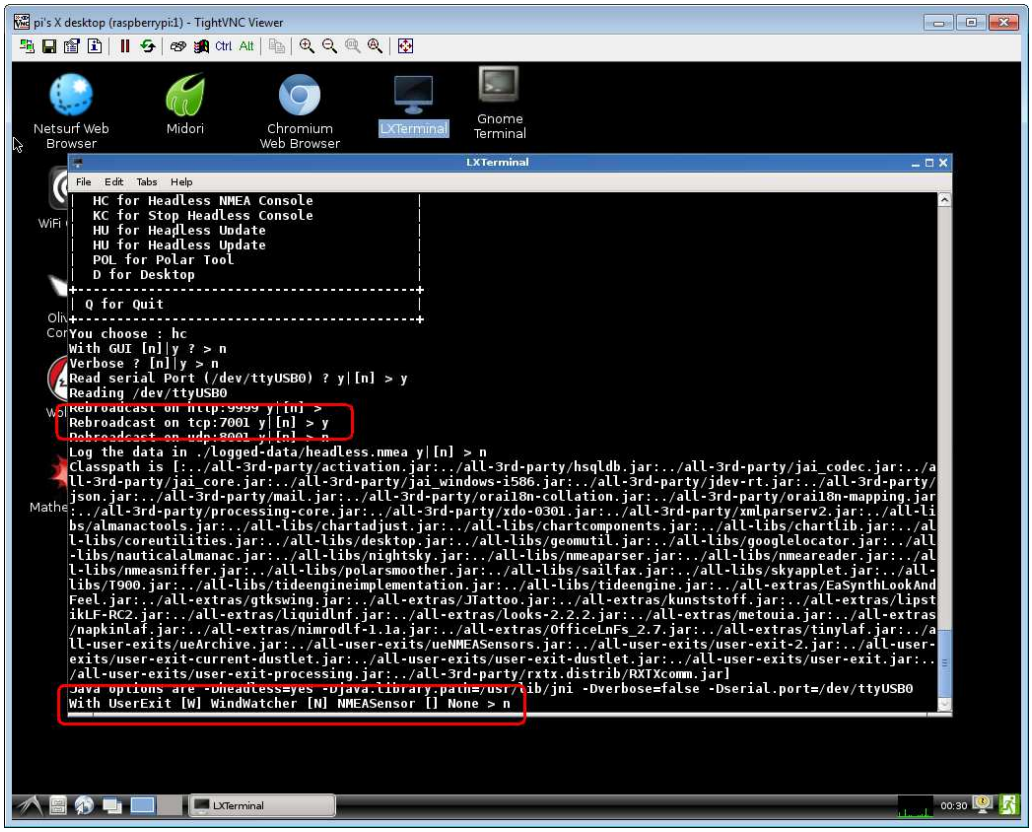
VNC is now started, we can connect.



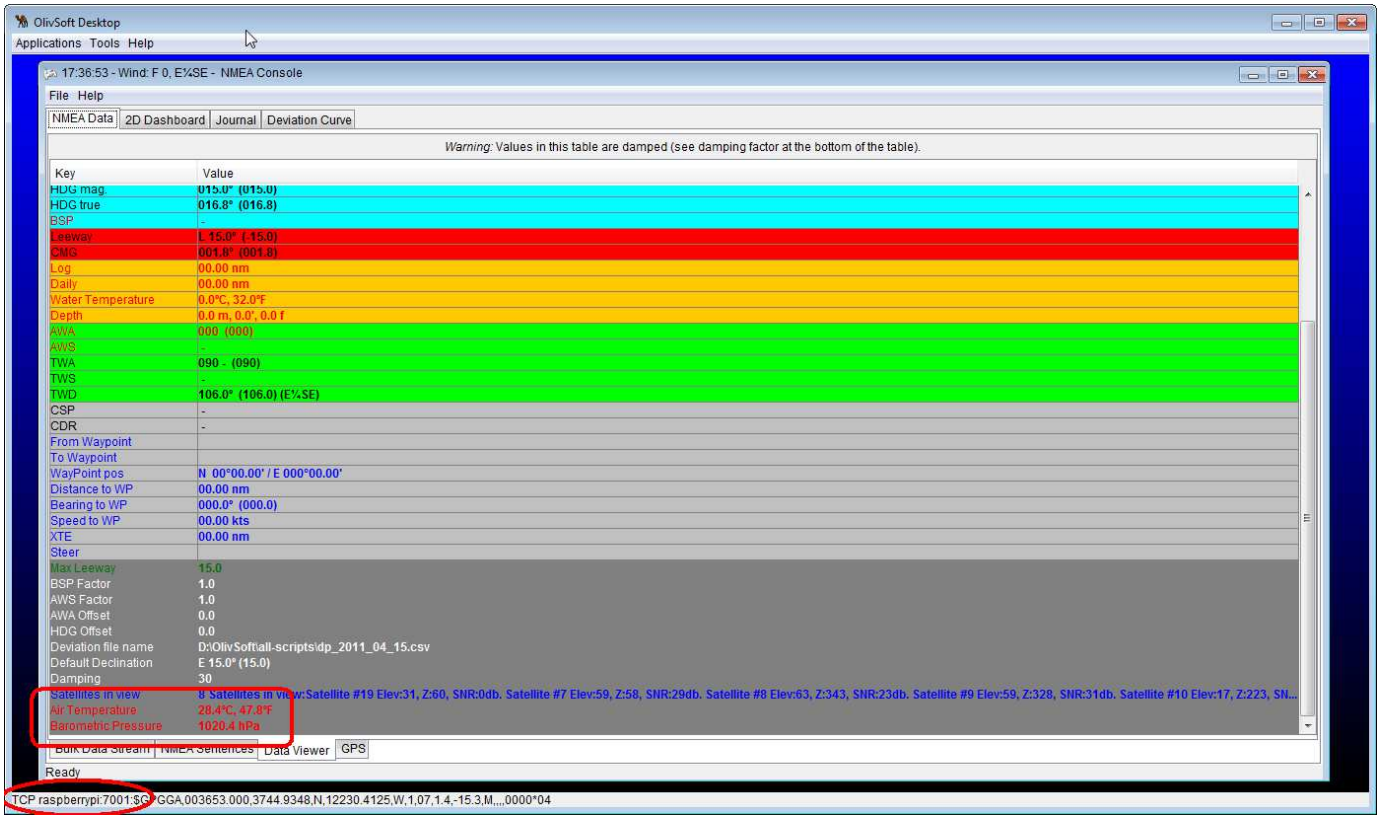
VNC connection...



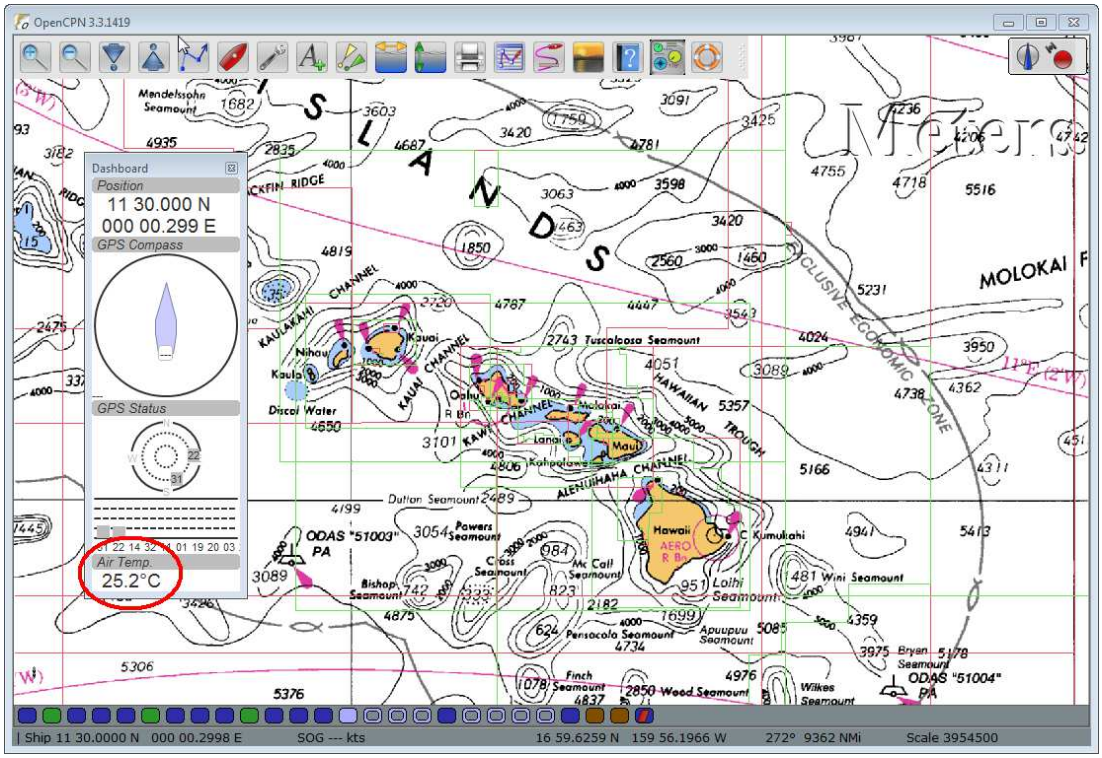
Start the "olivsoft" console, and choose HC, for Headless Console.



Option for the Headless Console...



That's it! From another machine, access the RasPI (TCP:7001 in this case), and check out the Data Viewer. The Air Temperature and Atmospheric Pressure - coming from the BMP180 connected on the RasPI - are available in the list, good for display, for logging, rendering..., whatever can be done with all the other data.



Whatever understands NMEA cannot tell the difference. Here is OpenCPN.

Here is a sample of the logging (*regular* logging) that can be done with the sensors injecting data in the NMEA stream:

```
$IIHDG,003,,,15,E*15
$IIMTW,+15.0,C*3C
$IIMWV,155,R,06.2,N,A*16
$IIMWV,155,T,06.6,N,A*14
$IIMMB,29.9870,I,1.0154,B*75
$IIMTA,25.6,C*04
$IIRMC,214009,A,3730.080,N,12228.857,W,00.0,080,090714,15,E,A*19
$IIXDR,P,1.0154,B,0*73
$IIVHW,,,003,M,00.0,N,,*67
$WIMDA,29.984,I,1.015,B,25.6,C,15.0,C,,,,,172.0,T,173.0,M,6.2,N,3.2,M*7D
$IIVLW,08195,N,000.0,N*56
$IIVWR,153,R,06.1,N,,,,*61
$IIGLL,3730.081,N,12228.856,W,214011,A,A*4B
$IIHDG,003,,,15,E*15
$IIMTW,+15.0,C*3C
$IIMWV,153,R,06.1,N,A*13
$IIMWV,155,T,06.2,N,A*10
$IIRMB,A,0.00,L,,HMB-3   ,,,,,001.20,184,,V,A*00
$IIMMB,29.9900,I,1.0155,B*72
$IIMTA,25.6,C*04
$IIXDR,P,1.0155,B,0*72
$WIMDA,29.987,I,1.015,B,25.6,C,15.0,C,,,,,170.0,T,171.0,M,6.1,N,3.1,M*7E
$IIRMC,214011,A,3730.081,N,12228.856,W,00.0,080,090714,15,E,A*10
$IIVHW,,,003,M,00.0,N,,*67
$IIVLW,08195,N,000.0,N*56
$IIVWR,150,R,05.9,N,,,,*69
$IIGLL,3730.081,N,12228.856,W,214011,A,A*4B
$IIHDG,003,,,15,E*15
$IIMTW,+15.0,C*3C
$IIMWV,150,R,05.9,N,A*1B
$IIMWV,153,T,06.1,N,A*15
$XXBAT,13.34,V,910,88*12
$IIMMB,29.9882,I,1.0154,B*78
$IIMTA,25.6,C*04
$IIRMB,A,0.00,L,,HMB-3   ,,,,,001.20,184,,V,A*00
$IIXDR,P,1.0154,B,0*73
$IIRMC,214013,A,3730.080,N,12228.856,W,00.0,080,090714,15,E,A*13
```

Source code

Available here.

A small detail: the BMP180 is only $9.95...
