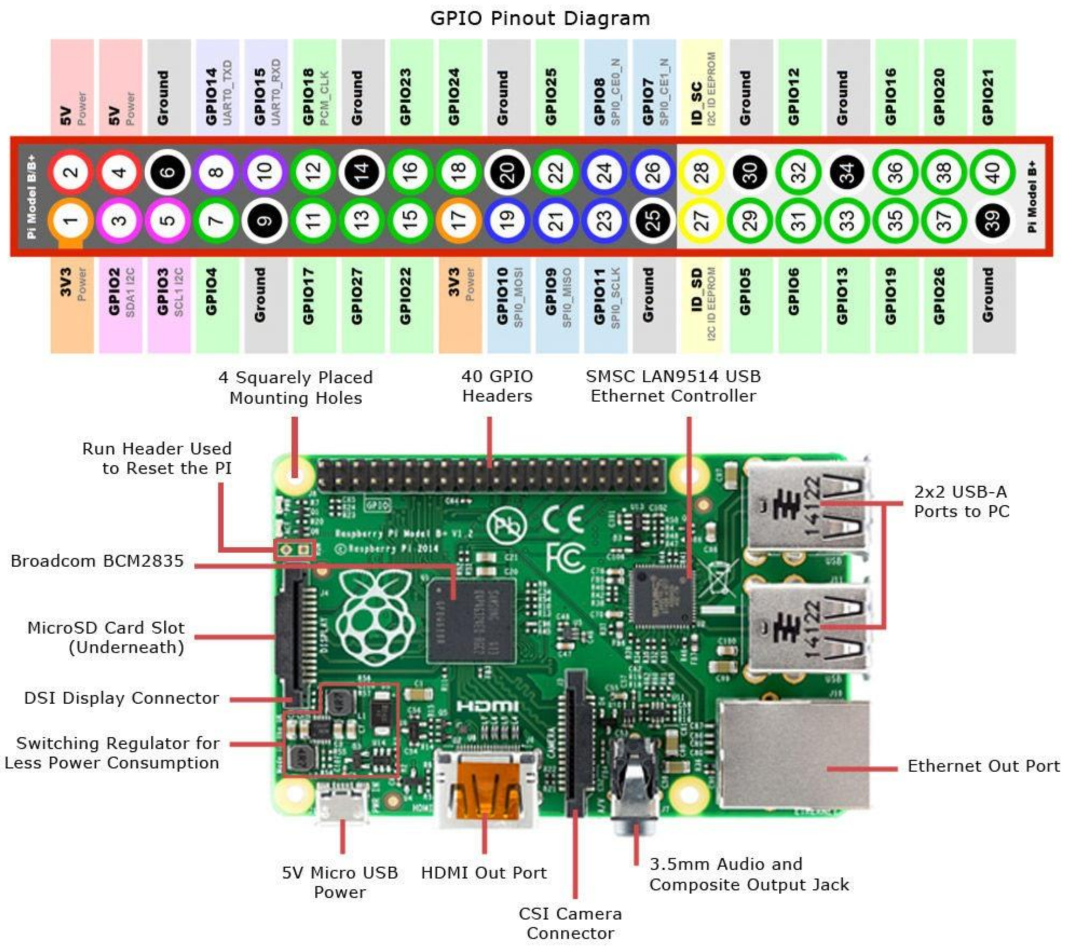
MANUAL OPENPLOTTER

SENSORS AND REPRESENTATIONS

NODE-RED

Content

1. INTRODUCTION OF THE RASPBERRY GPIO's



These GPIO pins are in the 40-pin connector side of Raspberry. In the picture we can see the legend of each pin.

As seen, pin 1 is the output of 3.3V.

Pins 2 and 4 function both as inlet or outlet voltage + 5V. If we feed the Raspberry by micro-USB connector we will have on these pins + 5V. But we could also connect a power supply, well stabilized 5V to one of these pins to work.

Blacks putting pins *Ground,* is the mass or negative power.

The GPIO pins are authentic green input / output general purpose. They are 17 in total.

There are other GPIOs with other colors. These, although can be used as general purpose, are reserved for specific functions such as serial port (gray GPIO 14 and 15), *I2C interface (* in pink GPIO3 and 5) which it serves to handle lmus *interface SPI (* Serial Parallel Interface clear blue GPIO10, 9, 11, 6 and 7).

The pins 27 and 28 are used to recognize yellow chips by I2C and thus install the appropriate driver automatically.

The purpose is to define the role of a GPIO that by definition and out.

¿ But what and means?

These pins are digital, or have the value 1 (on) or have the value 0 (disabled) and controlled by software. Is a program that can make use either as input or as output.

If you use outlet you have the function of a switch that is or is enabled or disabled. So that we can use to turn on or off devices, activate relays.

If we use it as input we can read the activation or deactivation of a sensor or switch. In this way we can to read sensors bilge open door, triggering an alarm (eg gas), even to read the pulses reel (count meters chain) or pulses of the slide (measuring speed on water). Also it used as GPIO output can activate another GPIO used as input.

I guess you are already imagining the possibilities offered by Raspberry. Not just a plotter but also the possibility of monitoring and control. Ie can turn the Raspberry in a central home automation.

Returning to the I2C and SPI interfaces, said interfaces are each developed by the industry.

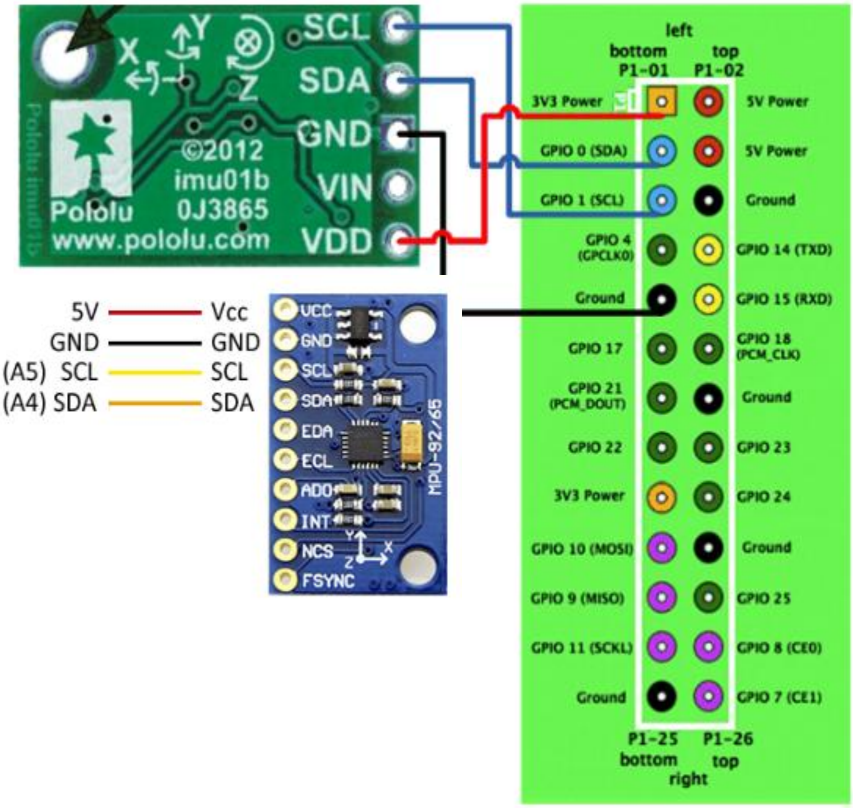
He **I2C** which it is I²C, English *Inter-Integrated Circuit,* It was developed by Philips in 1982. It is mainly used internally for communication between different parts of a circuit, for example, between a controller and peripheral circuits integrated. In our case Raspberry and IMU. The good thing about this interface is configured as a bus can connect multiple devices with a hierarchy to identify each of them.

At this point I clarify what a **IMU.** An IMU is a sensor measuring inertia (inertial measurement unit measurement unit- inertia). They are sensors that measure accelerations on different axes and are also known as gyros. Besides the gyroscope will usually add magnetic field sensors (use as compass) temperature, pressure and humidity.

He **Bus SPI (** *Serial Peripheral Interface)* is another communications standard, mainly used for transferring data between integrated circuits in electronic equipment. This bus uses more pins than the I2C but has the advantages, among others, to be faster than the I2C and be fully bidirectional (full duplex). This interface is used for example to connect to an analog-digital converter (AD). In this way we can read with the Raspberry analog signals, voltage, amperage, level of deposits, temperature, etc.

In short, the 40-pin connector of the Raspberry, opens the possibilities to connect the Raspberry with different electronic devices via the I2C and SPI bus and thus read data from sensors such as gyroscopes or magnetic field or analog-digital converters. The GPIO pin allows us to monitor and control both reading digital sensors and operating digital devices such as relays.

1.1. IMU connection



four cables are connected:

1. The power to pin 1 orange 3,3volt.
2. The SDA pin 3 violet.
3. The SCL goes to pin 5 violet.
4. The ground or negative to any pin GRND as 6 or 9 black.

Others are ignored. Some IMU support Power 3.3 and 5 volt. It is recommended to use 3.3V but if we use the 5 volt OJO not connect to
+ 3.3 but yours, to + 5V.

If they get more IMU's, they are placed in parallel. All together SDA and SCL all together. The raspi assigned an address on the bus to activate in OpenPlotter.

This is the command to display the I2C devices: -and i2cdetect 1

And this is the answer

0 1 2 3 Four. Five 6 July 8 9 to b c d and F

00: -- -- -- -- -- -- -- -- -- -- -- -- --

10: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --

twenty: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --

30: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --

40: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --

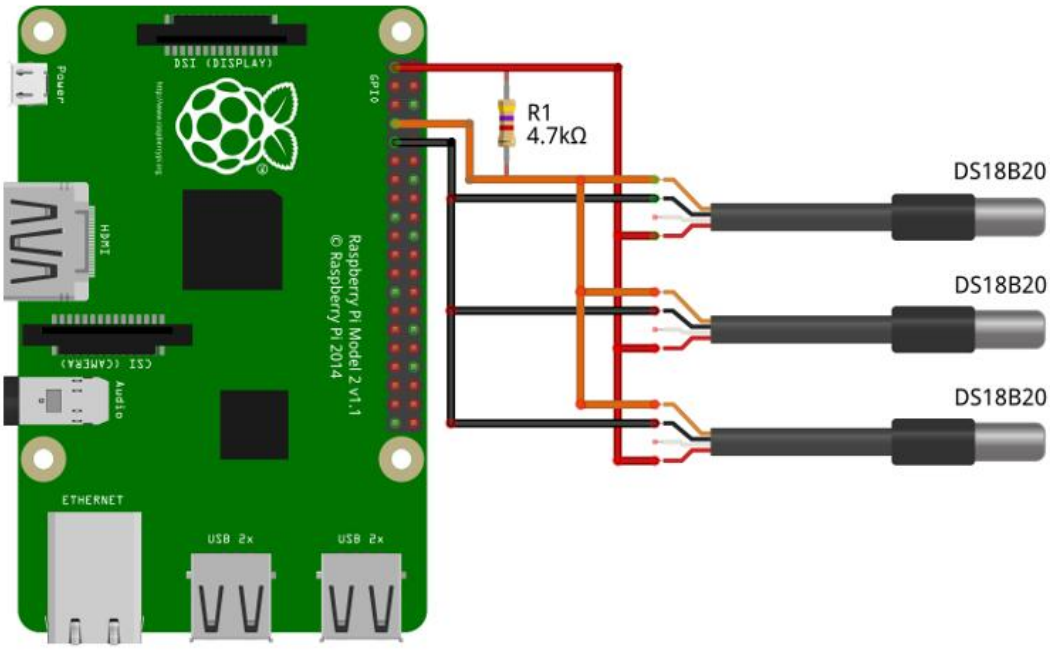
fifty: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --

60: 60 - -- -- -- -- -- -- -- -- -- -- -- -- -- --

70: -- -- -- -- -- -- -- --

The position is occupied 60/1 with the device 60.

1.2. Connection of temperature sensors DS18B20



These sensors are used to measure temperature and can be used to monitor ambient temperature, refrigerator, carter, exhaust, batteries, heater, etc.

They are water resistant so they can get wet without problems.

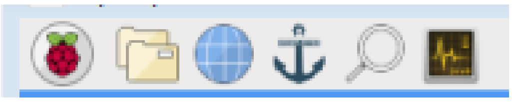
It is connected by three wires, and several sensors can be connected in series, as shown above.

1. Cable v 3.3
2. ground or negative to any pin GRND as 6 or 9 black.
3. amarrillo cable that connects to GPIO 4

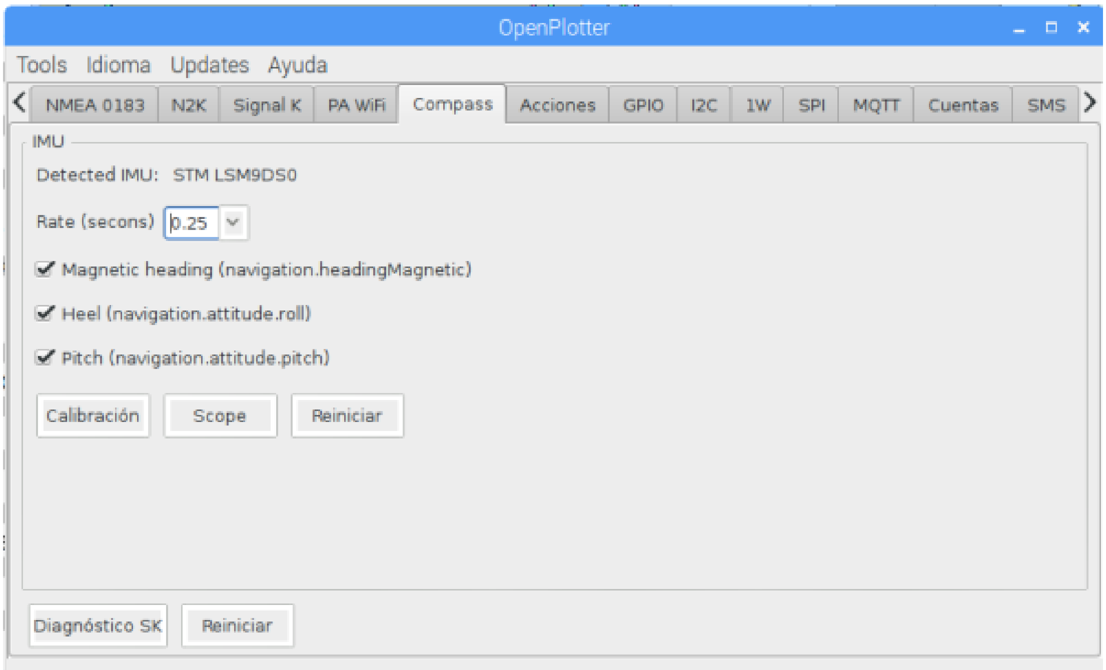
For correct operation must connect a resistor of 4.7 KQ between 3.3 vy cable connected to the GPIO 4 amarrillo cable.

2. CONFIGURATION SENSORS IN OPENPLOTTER

Once we connected the 1w sensors (which are temperature probes) and IMU (Gyro), must be configured in openplotter, for it opened openplotter with the anchor icon.



and opens openplotter



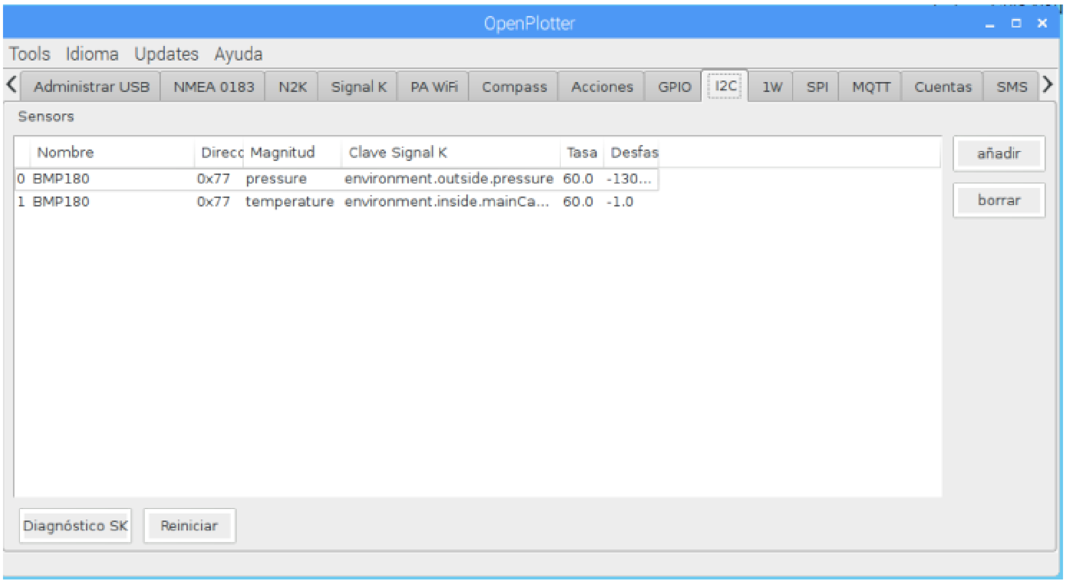
2.1. IMU configuration

If you have connected an IMU sensor, the tab **compass** we can see that the IMU has been detected, in this case the sensor is detected **STM LSM9050.** The system automatically detects / sensors, if they are connected.

We see that you can activate the magnetic heading, heeling and pitch. The sensor, which is performed by the icon calibration must be calibrated.

These three pieces of information, and are automatically set in Signal K, so you do not have to do it.

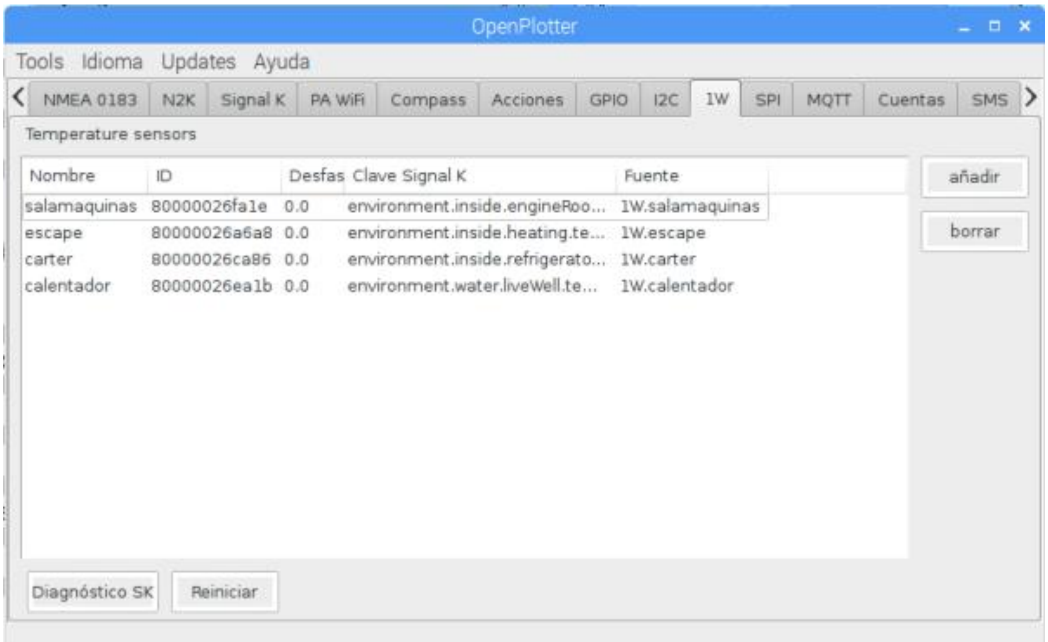
We no IMU, like this, they also give temperature and pressure data, and that in Openplotter we go to the tab I2C



We need to add data, which will be the two corresponding to the temperature and pressure sensor BMP180 name.

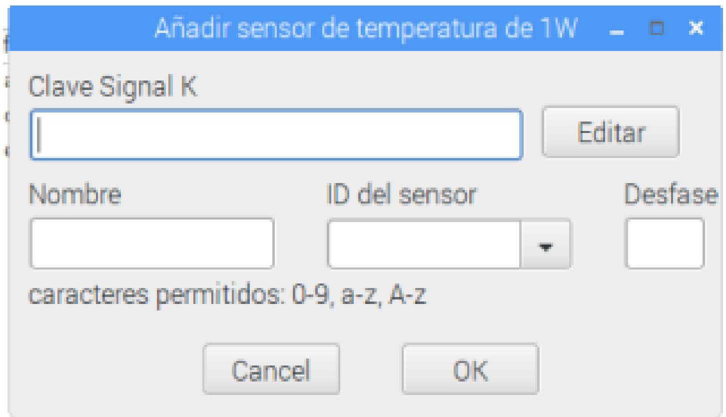
2.2. 1W configuration

Inside there is another tab called Openplotter **1W,** We entered it, and from there we will set the temperature sensors DSB180, we've connected in series.

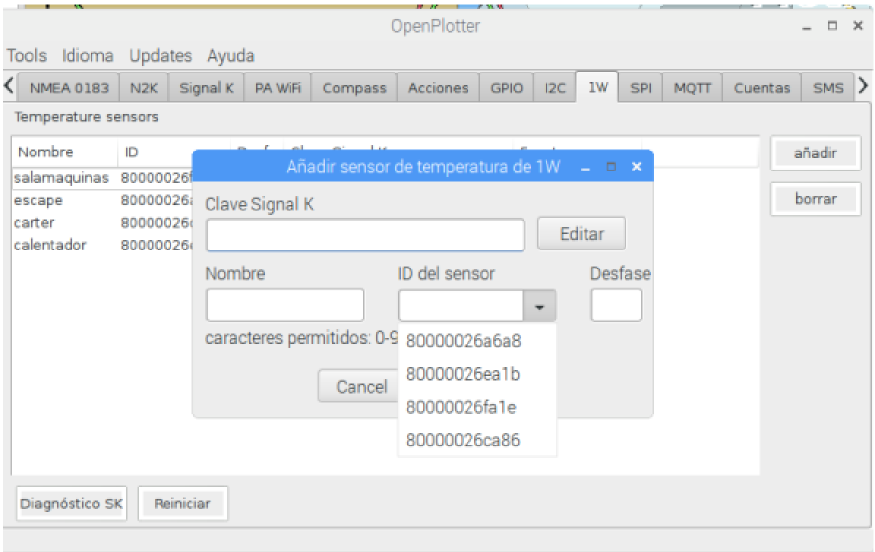


Here on this screen and appear set four sensors.

To add sensors have to click on the Add button and the following screen will appear.



To check if we properly connected sensors, we click where it says **ID sensor,** and we have connected sensors, appointed with a personal identifier each have to appear.



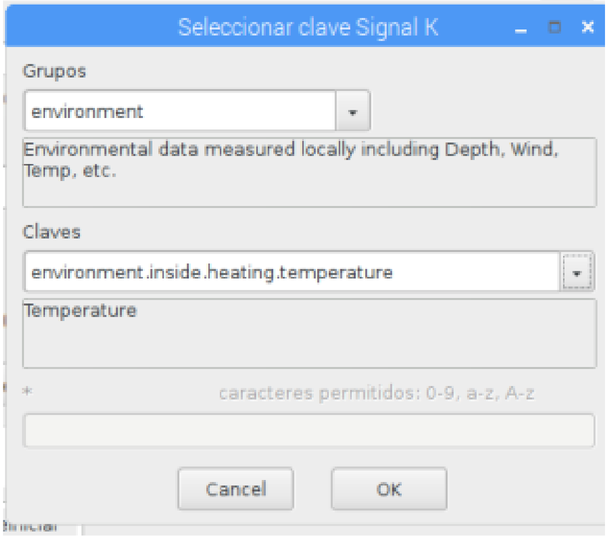
Select one of them, previously we have to know what it is, to then give it a name. To do this, how to do, it is select any, and to begin configuration.

We named it as we wish, in this case **escape,** and then we will edit where we're going to put a name to it has identified SignalK. This name is the one that later will use to generate 0183, NMEA200, judgments or set Node-network to display sensor data.

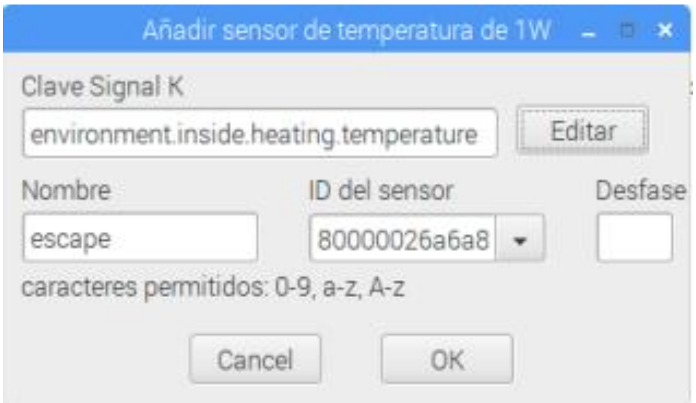
The name we use must be as close to reality, easily identify it later.

Pressing edit the following screen appears.

Here is the list of all keys SignalK, and we have to choose from the different groups that best suits us. For my case, I chose

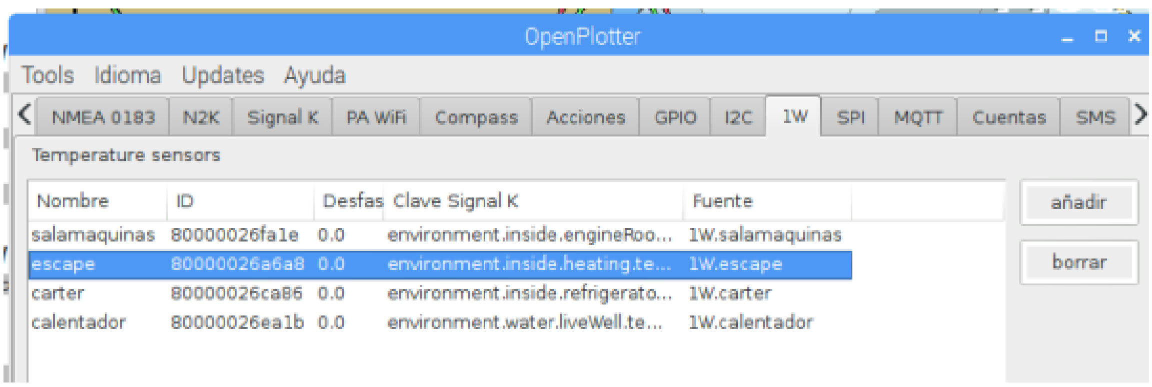


We give Ok



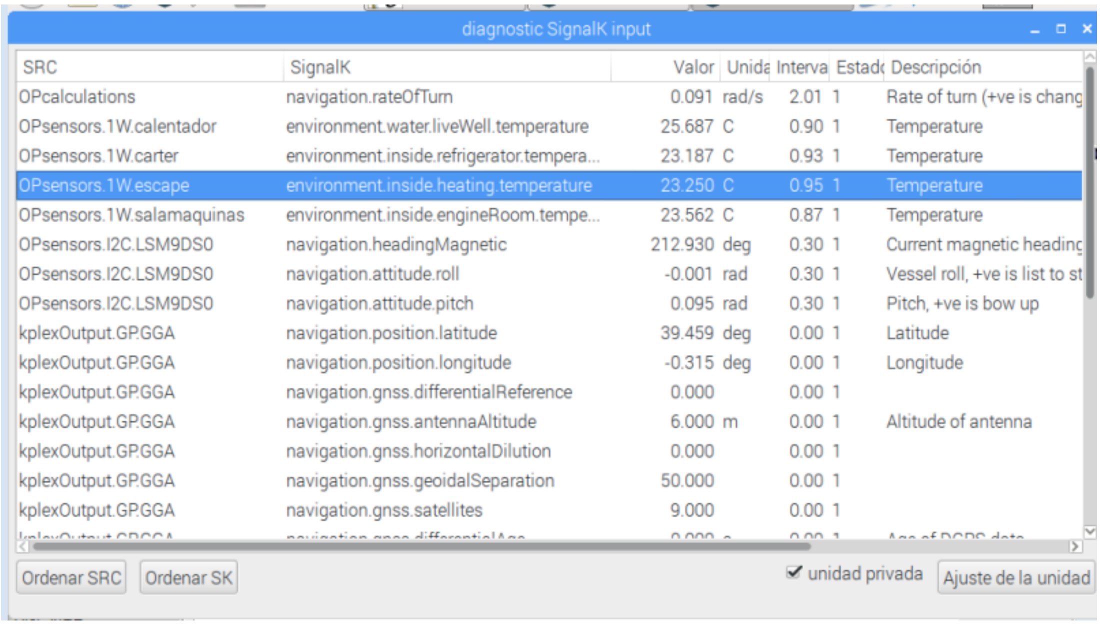
We can vary the measured sensor by **Gap,** if we found that the sensor measures more or less, here you can adjust the output value.

And as we accept the sensor appears set

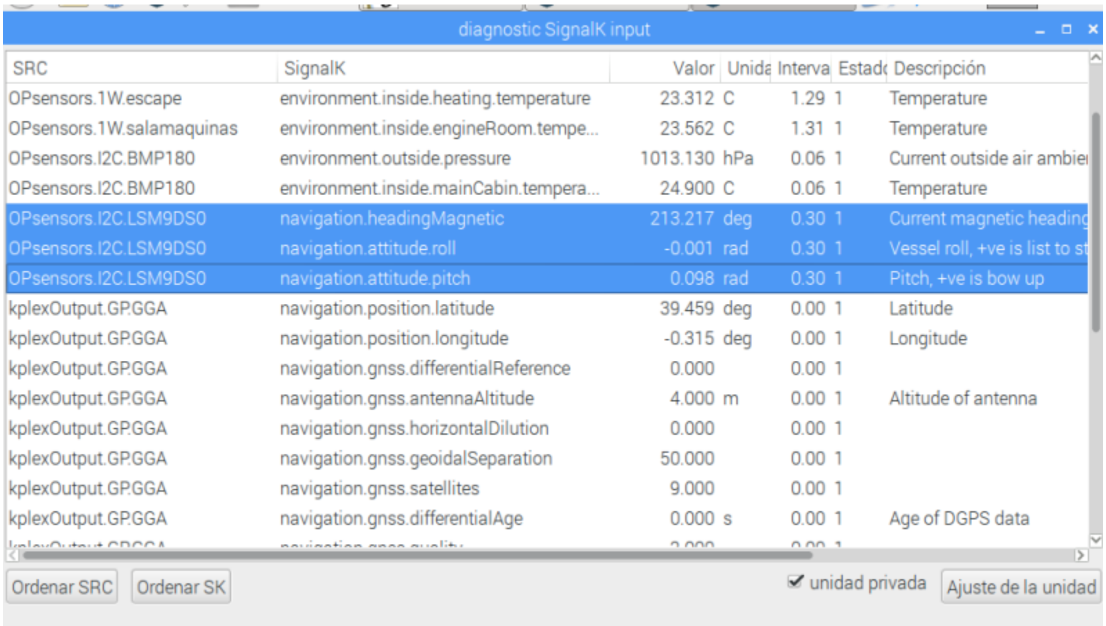


Now let's check whether it is properly set in **Signal K,** To do this click on the icon at the bottom left putting **SK diagnosis**

Here we see that the sensor is giving value in ° C.



Since **SignalK** we can see all the signs. Since we can see the signs of the IMU sensors, as can be seen



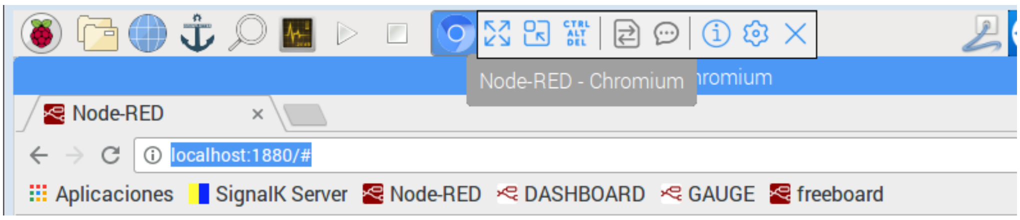
3. NODE DATA REPRESENTING-RED

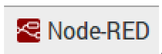
The representation of data obtained from the sensors (temperature, pressure, humidity), GPS antenna, data electronics boat as wind, depth, can be used to represent them, send them by e-mail, generate alarms, send them to an external server, or many more shares.

All this is done with the Node-network program, which Openplotter has been installed.

To enter the program we have to go to internet explorer.

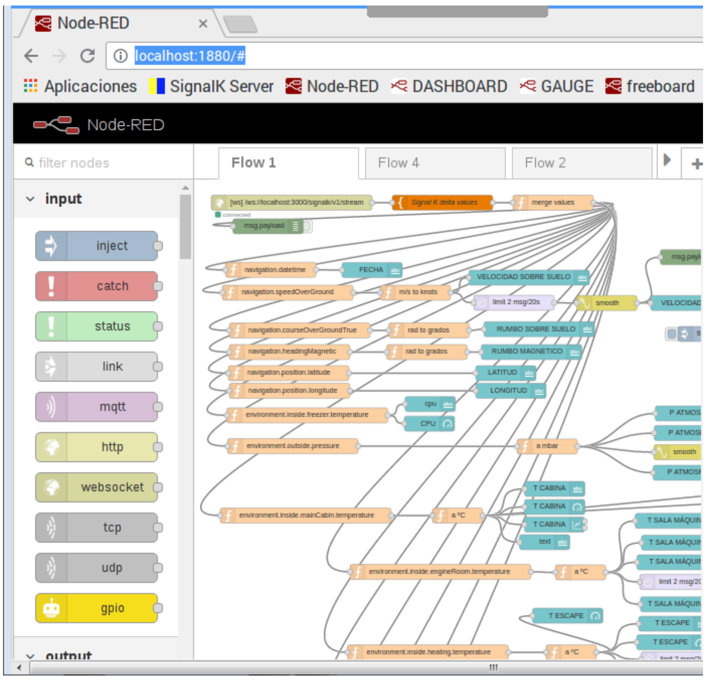
Opening the browser



In the bar **Markers** If we click on the icon 

And that leads us to the address can be observed **localhost: 1880 / #**

This opens the node-network program, and the following screen is observed.



Simply put, the program works by giving inputs, which in our case will be through the data provided to us through SignalK Openplotter, but can be other shapes.

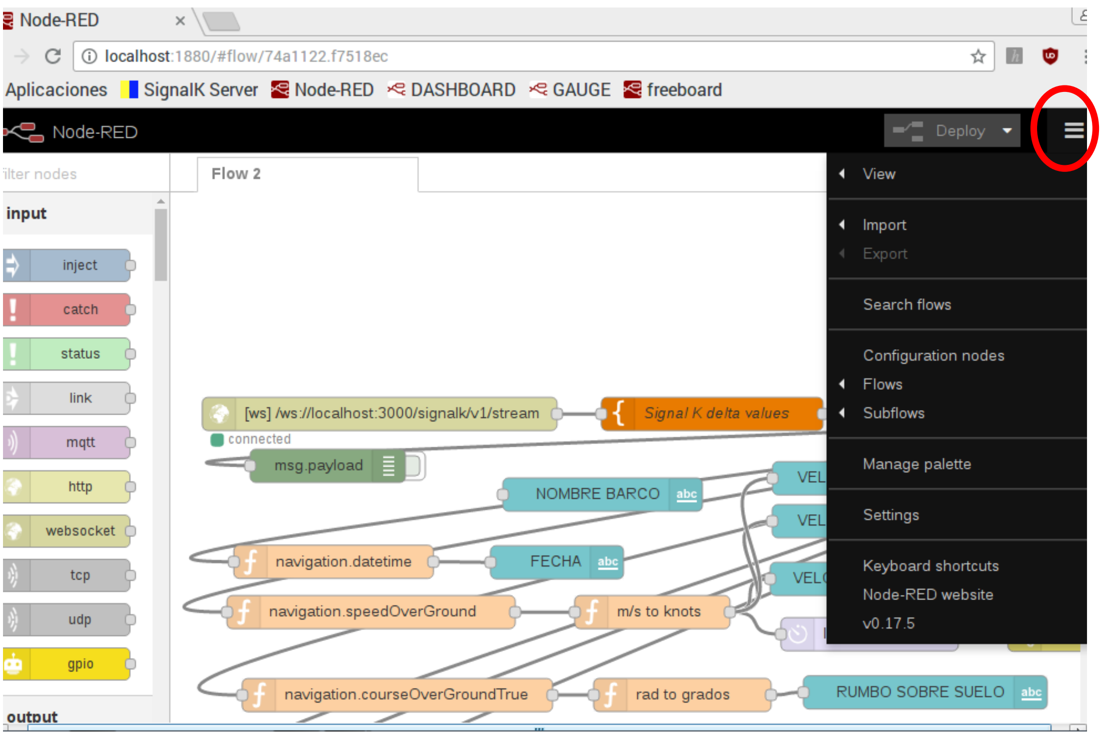
Once you have the data must be used in a function and then give an output, which can be a graphical, numerical data, a clock, sending by e-mail, etc. All this can be done without knowing programming, since programming can be copied and copied, and thus have represented data.

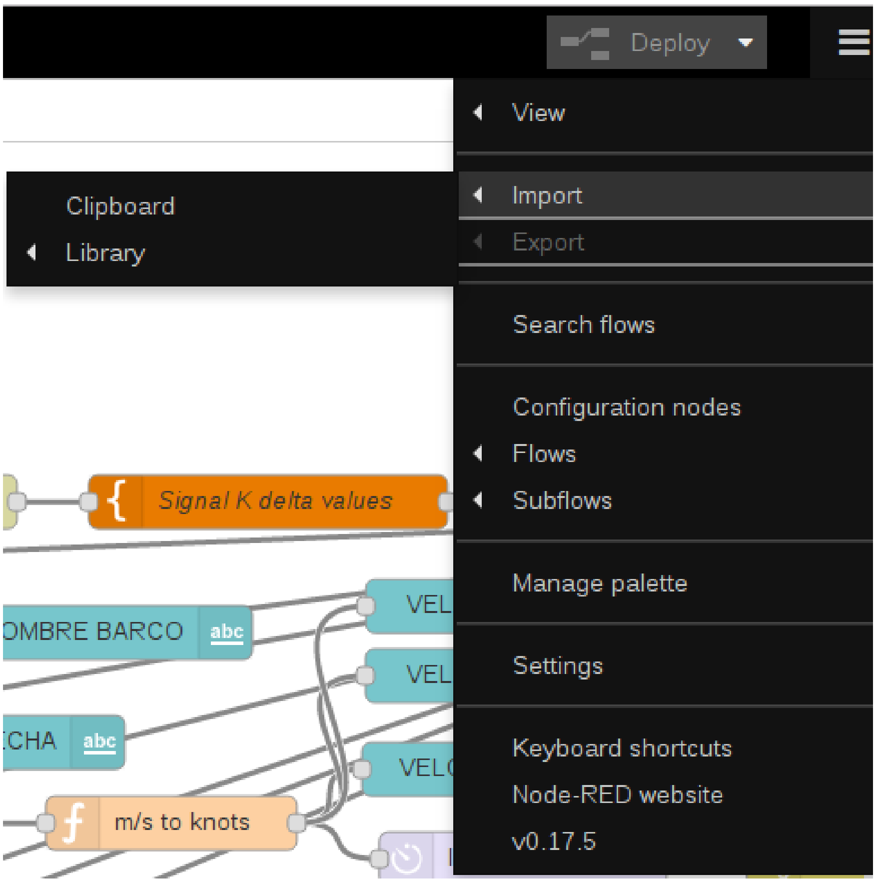
It can be seen as scheduled (by boxes, which have left) an example and copy the same to any other data that comes from SignalK and represent.

3.1. EXAMPLE FOR COPYING DATA RERESENTAR.

How to copy an example it is simple, to do what we have to do is copy the following code with Ctrl-C to copy it then go to network node.

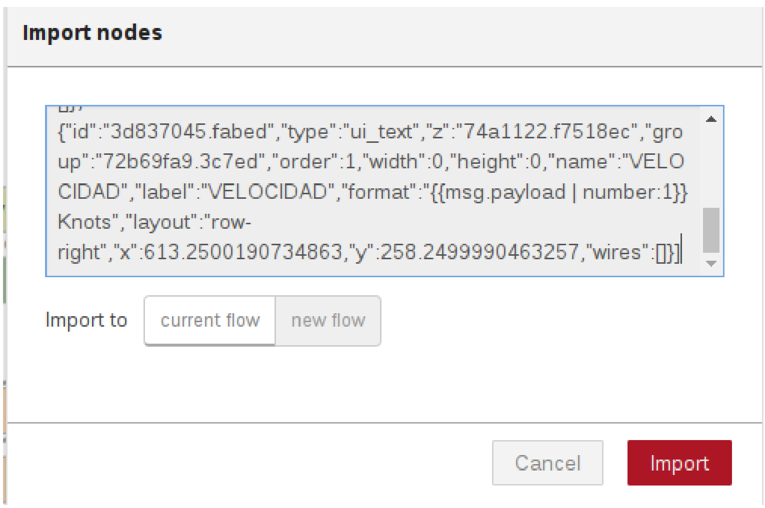
Once copied the code, we headed to node-network and in the upper right corner on the menu





We headed to Import / Clipboard

We will open the next screen where you will have to copy the code that is in the following pages. Once copied select **IMPORT**



[{ "Id": "74a1122.f7518ec", "type", "tab", "label": "Flow 2"}, { "id":
"bf7d28f7.f45188", "type": "websocket-
listener "," z "," "," path ":" ws: // localhost: 3000 / signalk / v1 / stream "," wholemsg ":" false "}, {" id ":" 9964cc9.930fa3 "
"type": "ui_group", "z": "", "name": "AUXILIARY", "tab": "74eebf13.f060e"
,"Order" 3, "disp" true, "width", "6"}, { "id": "247e3051.29c16", "type": "ui_group", "z", "", "name" "P

ATMOSPHERIC "," tab "," "," order "2," disp "false," width "," 15 "}, {" id ":" 821f2b6e.fc7458 "," typ e "" ui_base " , "theme":
{ "name": "theme-
custom "," lightTheme ": {" default "" # 0094CE "" baseColor "" # 0094CE "" BASEFONT "" Helveti ca

Neue "," edited ": true," reset "false}," darkTheme ": {" default "" # 097479 "," baseColor "" # 0974 79 "," BASEFONT ""
Helvetica
Neue "," edited ": true," reset "false}," customTheme ": {" name "," Black / white "," default "," # 4 B7930 "," baseColor ""
353079 ", "BASEFONT" "- apple-system, BlinkMacSystemFont, Segoe UI, Roboto, Oxygen-Sans, Ubuntu Cantarell
Helvetica Neue, sans
serif "," reset "false}," themeState ": {" base-
color ": {" default "" # 097479 "," value "," # 097479 "," edited ": false}," page-titlebarbackgroundColor ":
{" value "" # 097479 "," edited ": true }, "pagebackgroundColor": { "value" "# 111111", "edited": true},
"page-sidebarbackgroundColor": { "value": "# 000000", "edited": false}, "grouptextColor": { "value": "#
10cfd8", "edited": true}, "groupborderColor": { "value" "# 555555", "edited": true},
"groupbackgroundColor": { "value": "# 333333 "" edited ": false}," widgettextColor ": {" value ":" #
eeeeee "," edited ": true}," widgetbackgroundColor ": {" value ":"# 353079 "," edited ": false},"
widgetborderColor ": {" value "" # 333333 "," edited ": false}," base-font ": {" value ":" - applesystem,
BlinkMacSystemFont, Segoe

UI, Roboto, Oxygen-
Sans, Ubuntu, Cantarell, Helvetica Neue, sans-serif "}}}," site ": {" name ":" Node-RED
Dashboard "," hideToolbar ":" false "," allowSwipe ":" false "," dateFormat "" DD / MM / YYYY "," si zes ": {" sx ": 48," s ": 48,"
gx "6," g "6," cx "6," c "6," px "0," p ": 0}}}, {" id ":" 8bebf753.eb9e78 "," type ":" ui_tab "," z "," "," name ":" GAUGE "," icon ":"
dashboard "," order ": 3}, {" id ":" c168102c.ba 45c "," type ":" ui_group "," z "," "," name "," S

MACHINES "," tab "," "," order "1," disp "false," width "," 15 "}, {" id ":" a8fc798d.da22c8 "," type ":" ui_tab " "z": "", "name", "SPEED", "icon":
"dashboard", "order" 7}, { "id": "72b69fa9.3c7e d", "type": "ui_group" "z": "", "name", "SPEED", "tab", "a8fc798d.da22c8", "disp" false, "width",
"15"}, { "id": "74eebf13.f060e" , "type": "ui_tab", "z", "", "name", "GENERAL", "icon": "das hboard", "order": 1}, { "id": "6aabf213.b6880c" , "type":
"ui_group "," z "," "," name ":" T EXHAUST "," tab "," "," order "1," disp "false," width "," 15 "}, { id "" dcebd1d.9fff33 "," type ":" ui_gr oup "," z "," ","
name "," SPEED "," tab "," 8bebf753.eb9e78 "," order "1," disp "false," width "," 8 "}, {" id ":" 3023e014.44b19 "," type ":" ui_group "," z "," "," name
":" T"", "Name", "SPEED", "tab", "8bebf753.eb9e78", "order" 1, "disp" false, "width", "8"}, { "id": "3023e014. 44b19 "," type ":" ui_group "," z ":" ","
name ":" T"", "Name", "SPEED", "tab", "8bebf753.eb9e78", "order" 1, "disp" false, "width", "8"}, { "id": "3023e014. 44b19 "," type ":" ui_group "," z
":" "," name ":" T
CARTER "," tab "," "," order "1," disp "false," width "," 15 "}, {" id ":" 137084f7.28976b "," type ":" ui _group " , "z", "", "name": "Navigation", "tab",
"74eebf13.f060e", "order" 2, "disp" true, "wid th", "6"}, { " id ":" 1c3b3a6e.ec5616 "," type ":" ui_group "," z "," "," name ":" T HEATER "," tab "," ","
order "1," disp ": false, "width", "15"}, { "id": "1ce28a56.52d286" "ty pe" "ui_group", "z", "", "name", "DATE", "tab": "74eebf13.f060e", "order": 1,
"disp "false," wid th "," 13 "}, {" id ":" aa9e8706.028388 "," type ":" ui_group "," z "," "," name ":" SEA WATER " "tab", "", "order" 1, "disp" false,
"width", "15"}, { "id": "76ab3eec.2d1c", "type": "ui_group", "z" : "", "name", "HUMIDITY", "tab", "", "disp" false, "width", "15"}, { "id":
"f6243168.e1421", "t"Width", "15"}, { "id": "76ab3eec.2d1c", "type": "ui_group", "z", "", "name", "HUMIDITY", "tab", "", "disp" false, "width", "15"}, {
"id": "f6243168.e1421", "t"Width", "15"}, { "id": "76ab3eec.2d1c", "type": "ui_group", "z", "", "name", "HUMIDITY", "tab", "", "disp" false, "width",
"15"}, { "id": "f6243168.e1421", "t

pe "" ui_group "," z "," "," name "," TEMP
EXT "," tab "," "," disp "false," width "," 15 "}, {" id ":" a539de42.4b8c4 "," type ":" ui_group "," z ":" " , "n ame", "ENGINE", "tab", "", "order" 1, "disp"
false, "width", "6"}, { "id": "82ea316d.3a4d9", " type ":" ui_group "," z "," "," name "," AUXILIARY "," tab "," "," order "2," disp "false," width "," 6 "}, {
"id": "4fa 76d14.4c3a24", "type": "ui_group", "z", "", "name": "TEMPERATURE", "tab", "", "disp" false, "width "" 15 "}, {" id ":"98c39b56.332388 ","
type ":" ui_group "," z "," "," name ":" P ATM EXT & T "," tab "," "," order "1," disp ": true , "width", "15"}, { "id": "d3c64421.ceb768", "type": "ui_grou
p", "z", "", "name": "CARTER & ESCAPE & WATER", "tab", " "," order "1," disp "false," width "," 15 "}, {" id ":" 7290e28a.b805ac "," type ":" ui_group
"," z "," "," name ":" EFIwidth "," 15 "}, {" id ":" d3c64421.ceb768 "," type ":" ui_grou p "," z "," "," name ":" CARTER & ESCAPE & WATER "," tab ","
", "order" 1, "disp" false, "width", "15"}, { "id": "7290e28a.b805ac", "type": "ui_group", "z", "", "name" "EFIwidth "," 15 "}, {" id ":" d3c64421.ceb768 ","
type ":" ui_grou p "," z "," "," name ":" CARTER & ESCAPE & WATER "," tab "," ", "order" 1, "disp" false, "width", "15"}, { "id": "7290e28a.b805ac", "type": "ui_group", "z", "", "nar
INTERCHANGE "," tab "," "," order "1," disp "false," width "," 17 "}, {" id ":" d3f6900.2b56a7 "," type ":" or i_group " "z": "",
"name": "Tª
CPU "," tab "," 8bebf753.eb9e78 "," order "2," disp "false," width "," 8 "}, {" id ":" f79f8824.3c3fb8 "
,"Type": "debug", "z": "74a1122.f7518ec", "name", "", "active" false, "console", "false", "compl ete" "payload", "x
"142.00000762939453," and "246.99999523162842," wires "[]}, {" id ":" 493 ae04e.cc815 "," type ":" websocket

in "," z ":" 74a1122.f7518ec "," name "," "," server ":" bf7d28f7.f45188 "," client "," "," x ": 188.85714
721679688, "and" 200.14285278320312, "wires" [["67aefb02.b4b074"]]}, { "id": "34e13042.22 1D7", "type":
"UI_TEXT", "z": "74a1122. f7518ec "," group "," 137084f7.28976b "," order "3" width "0," height "0," name "," LENGTH
"," label "," LENGTH "," format " "{{msg.payload |
number: 2}} & Deg; "" layout ":" row-
spread "," x ": 624.8571472167969," and "574.1428527832031," wires "[]}, {" id ":" 67aefb02.b4b 074 "," type ":" jsonpath
"," z ":" 74a1122. f7518ec "" expression "" $ updates..values [*] "," split. "false," name ":" Signal
K delta
values "," x ": 488.8571472167969," and "200.14285278320312," wires "[[" 477cbfe4.a4f1e "]]}, {" id ":" 477cbfe4.a4f1e
"," type "," function "," z ":" 74a1122.f7518ec "," name ":" merge values "," func ":" var = {} paths; \ nvar arrayLength =
msg.payload.length; \ nfor (var i = 0; i <arrayLength ;
i ++) {\ N
paths [msg.payload [i] .Path] = msg.payload [i] .value; \ n} \ nmsg.payload = paths; \ Nreturn msg; "" outputs "1,"
noerr "0," x ": 708.8571472167969," and "200.14285278320312," wires "[[" 9187b1a5.59445 "" f79f8824.3c3fb8 ""
530adc5e.b244b4 "" cda1650a.6342d8 "," 1c2ac9
61.991ba7 "" 5444484f.daeed8 "" cde27078.49e7b "]]}, {" id ":" 9187b1a5.59445 "," type ":" fu nction "," z ":"
74a1122.f7518ec "," name ":" navigation.speedOverGround "" func ":" var signalk_key = \
"navigation.speedOverGround \"; \ n \ nif (msg.payload.hasOwnProperty (sign alk_key)) {\ n
msg.payload = msg.payload [signalk_key]; \ n return
msg; \ n} "," outputs "1," noerr "0," x ": 181.10714721679688," and "379.1428565979004," wires "[[" 5b83dd9c.6fca54
"]]}, {" id " "5b83dd9c.6fca54", "type", "function", "z": "74a1122.f7518ec", "name", "m / s to knots", "func" "msg.payload
= Number (msg. payload) * 1.94384 \ Nreturn msg; "" outputs "1," noerr "0," x ": 434.8571548461914," and
"379.1428565979004," wires "[[" bdbaac63.10573 "," 49a17d71.a46414 "" 829cae42.a7edc "" 3d837045.fabed "]]}, {" id
":" 2ac961.991ba7 1c "," type "," function "," z ":" 74a1122.f7518ec "," name " "navigation.position.lat itude", "func "" var

signalk_key = \ "navigation.position \"; \ n \ nif (msg.payload.hasOwnProperty (signalk_key)) {\ n
msg.payload = msg.payload [signalk_key] [\ "latitude \"] \ n return
msg; \ n} "," outputs "1," noerr "0," x ": 182.85714721679688," and "539.1428527832031," wires "[["
6e882d1c.0319a4 "]]}, {" id " "6e882d1c.0319a4", "type": "UI_TEXT", "z": "74a1122.f7518ec", "group",
"137084f7.28976b", "order" 4, "width" 0, "height" : 0, "name", "latitude", "label", "L atitude", "format": "{{msg.payload
| number: 2}} & Deg "," layout ":" row-
spread "," x ": 614.8571472167969," and "539.1428527832031," wires "[]}, {" id "," 42d8 cda1650a.63 "," type "," function
"," z ":" 74a1122. f7518ec "," name ":" navigation.position.longitude "," f unc ":" var

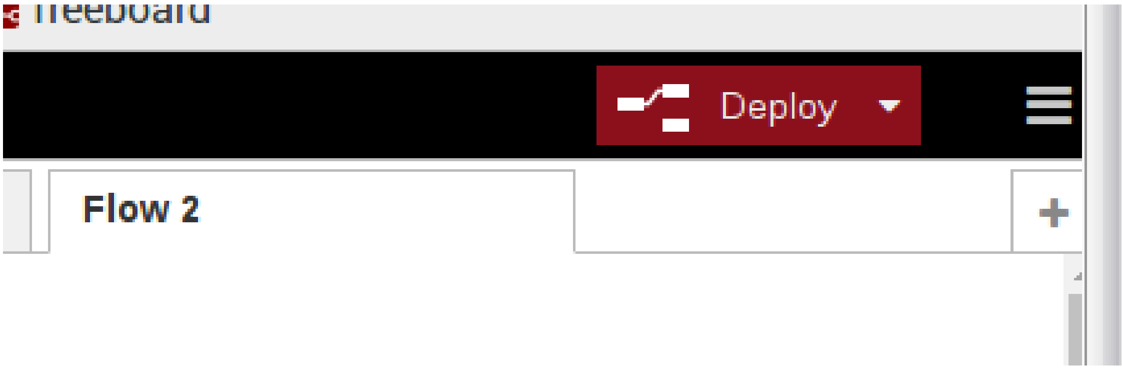
signalk_key = \ "navigation.position \"; \ n \ nif (msg.payload.hasOwnProperty (signalk_key)) {\ n
msg.payload = msg.payload [signalk_key] [\ "longitude \"] \ n return
msg; \ n} "," outputs "1," noerr "0," x ": 184.85714721679688," and "574.1428527832031," wires "[[" 34e13042.221d7 "]]}, {" id " "530adc5e.b244b4", "type": "function", "z", "74a1122.f7518ec", "name":
"navigation.courseOverGroundTrue", "func" "var
signalk_key = \ "navigation.courseOverGroundTrue \"; \ n \ nif (msg.payload.hasOwnProperty (signalk_key)) {\ n
msg.payload = msg.payload [signalk_key]; \ n return
msg; \ n} "," outputs "1," noerr "0," x ": 204.85714721679688," and "454.1428527832031," wires "[[" 3daced9b.52c2b2 "]]}, {" id " "49a17d71.a46414", "type": "delay", "z": "74a1122.f7518ec", "name", "", "pauseType", "rate", "timeout", "15", "timeoutUnits ":" seconds "," rate "," 2 "," nbRate Units "," 20 "," rateUnits ":" second "," randomFirst "," 1 "," randomLast "," 5 "," randomUnits " "sec onds", "drop" true, "x": 630.8571472167969, "and" 399.1428527832031, "wires" [["26332618. b1fb4a "]]}, {" id ":" d5957144.0a85 "

8.9405136108398, "and" 399.58731174468994, "wires": [[], []]}, { "id": "bdbaac63.10573", "type": "UI_TEXT", "z": "74a1122.f7518ec" "group": "137084f7.28976b", "order": 0, "width": 0, "height": 0, "name": "SPEED
ON SOIL "," label ":" SPEED ON
SOIL "," format ":" {{msg.payload | number: 1}} Knots "," layout ":" row-spread "," x ": 661.1071624755859," and "349.1428556442261," wires "[]}, {" id ":" 42480453.7e d1bc "," type ":" UI_TEXT "," z ":" 74a1122.f7518ec "," group ":" 137084f7.28976b "," order ": 0," widt h ": 0," height ": 0," name ":"
HEADING ON SOIL "," label ":" HEADING ON
SOIL "," format ":" {{msg.payload | number: 1}} "" Layout ":" row-spread "," x ": 681.1071548461914," and "451.6428565979004," wires "[]}, {" id ":" cde27078.49 E7B "," type "," function "," z ":" 74a1122. f7518ec "," name ":" navigation.datetime "" func ":" var signalk_key = \ "navigation.datetime \"; \ n \ nif (msg.payload.hasOwnProperty (signalk_key)) {\ n
msg.payload = msg.payload [signalk_key]; \ n return
msg; \ n} "," outputs "1," noerr "0," x ": 147.28570556640625," and "333.7142639160156," wires "[[" cbcda48b.a470b8 "]]}, {" id " "cbcda48b.a470b8", "type": "UI_TEXT", "z": "74a1122.f7518ec", "group", "1ce28a56.52d286", "order" 2, "width", "12", " height "," 2 "," name "," DATE "," label "," DATE "," format ":" {{}} msg.payload "," layout "," col-

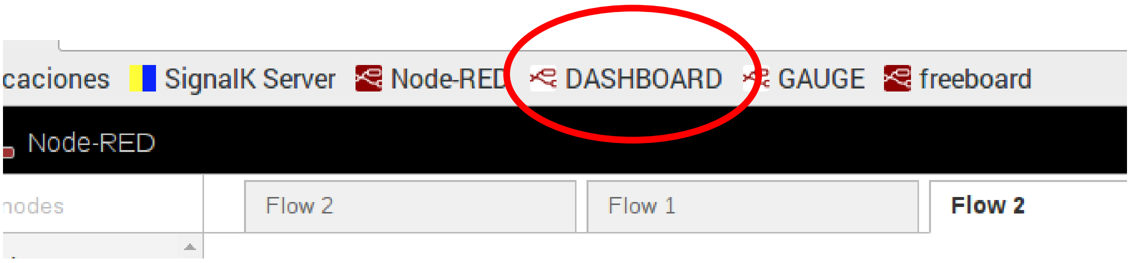
center "," x ": 348.8571472167969," and "334.1428527832031," wires "[]}, {" id ":" 26332618.b1f B4A "," type ":" smooth "," z ":" 74a1122. f7518ec "," name "," "," action "," mean "," count "," 15 "," rou nd ":" 1 "," mult ":" single "," x ": 815.6905364990234," and "399.67062759399414," wires ": [[" d59
57144.0a85 "" 829b5feb.b4e86 "]]}, {" id ":" 3daced9b.52c2b2 "," type "," function "," z ":" 74a112
2.f7518ec "," name ":" rad to degrees "," func ":" msg.payload =
Number (msg.payload) * 57.29 \ Nreturn
msg; "" outputs "1," noerr "0," x ": 461.94043731689453," and "454.0039415359497," wires "[[" 42480453.7ed1bc "]]}, {" id ":" 770a2f3d. 3f786 "," type "," inject "," z ":" 74a1122.f7518ec "" nam e "," "," topic "," "," payload ":" "," payloadType ":" date " , "repeat", "10", "crontab": "", "once": true, "x": 152.8570556640625, "and" 779.142822265625, "wires" [["d2bb92e9.c11e"]]}, { "id": "d2bb9 2e9.c11e", "type": "exec", "z", "74a1122.f7518ec", "command": "vcgencmd", "addpay": false, "append": "measure_temp" ,"UseSpawn": "", "timer", "", "oldrc" false, "name": "getCPUtemp", "x": 279.8570556640625, "and" 838.642822265625, "wires" [["292fb032.a2446 "], [], []]}, {" id ":" efe408d0.314968 "," type ":" debug "," z ":" 74a1122.f7518ec "," name ":" debug showSK "," active "false," console "," false "," complete "," payload "," x ": 528.8570556640625," and "839.642822265625," wires "[]}, {" id ":" 292fb032.a2446 "," type ":" function "," z "," 74a1122.8570556640625 "and" 838.642822265625, "wires" [["292fb032.a2446"], [], []]}, { "id": "efe408d0.314968", "type": "debug", "z "" 74a1122.f7518ec "," name ":" Debug showSK "," active ": false," console ":" false "," complete "," payload "," x ": 528.8570556640625," and "839.642822265625 , "wires": []}, { "id": "292fb032.a2446", "type", "function", "z": "74a1122.8570556640625 "and" 838.642822265625, "wires" [["292fb032.a2446"], [], []]}, { "id": "efe408d0.314968", "type": "debug", "z "" 74a1122.f7518ec "," name ":" Deb

f7518ec "," name ":" msg.payload "," func ":" cpu_temp =
parseFloat (msg.payload.replace (\ "temp = \" \ "\") replace (\ " 'C \\ n \" \ "\").) \ ncpu_temp =
cpu_temp \ nmsg.payload = '{\\\ "updates \\\": [{\\\ "source \\\": {\\\ "type \\\":
\\\ "ARMTEMP \\\" \\\ "\\\ src" : \\\ "RPIMCU \\\"}, \\\ "values \\\": [{\\\ "path \\\":
\\\ "environment.inside.freezer.temperature \\\" \\\ "value \\\" '+ cpu_temp +'}]}]} \\ n '; \ Nreturn msg; "" outputs ": 1,
"noerr" 0, "x": 382.8570556640625, "and" 780.142822265625, "wires" [["ef e408d0.314968" "6599324d.86aadc"]]}, {
"id": "6599324d. 86aadc "," type ":" udp out "," z "," 74a1122.f7518ec "," name ":" sendSK "," addr ":" localhost "," iface
":" "," port "," 55559 "
,"IPV" "udp4" "outport": "", "base64" false, "multicast": "false", "x": 556.8570556640625, "and" 7
78.142822265625, "wires" []}, { "id": "5444484f.daeed8", "type", "function", "z": "74a1122.f75 18ec", "name":
"environment.inside.freezer .temperature "" func ":" var signalk_key = \ "environment.inside.freezer.temperature \"; \
n \ nif (msg.payload.hasOwnPro perty (signalk_key)) {\ n
msg.payload = msg.payload [signalk_key]; \ n return
msg; \ n} \ n "," outputs "1," noerr "0," x ": 219.99603271484375," and "623.0317459106445," wir is "[[" a8e95994.5bbec8
"," dd095b1d.8f8658 "]]}, {" id ":" a8e95994.5bbec8 "," type ":" UI_TEXT "
,"Z": "74a1122.f7518ec", "group", "9964cc9.930fa3", "order" 5, "width" 0, "height" 0, "name": "cpu", "label": "CPU",
"format": "{{msg.payload | number: 1}} & Deg; "" layout ":" row-
spread "," x ": 464.4405288696289," and "600.5317449569702," wires "[]}, {" id ":" dd095b1d.8f8 658 "," type ":" ui_gauge
"," z ":" 74a1122. f7518ec "," name ":" CPU "," group ":" d3f6900.2b56a7 "
,"Order" 2, "width", "8", "height", "8", "GType" "gage", "title": "CPU", "label", "C", "format": "{{valu e |
number: 0}}
thC "," min "," 30 "," max ":" 75 "," colors ": [" # 00b500 "," # e6e600 "," # ca3838 "]," seg1 ":" 60 "," sec2 ":" 67 "," x ":
462.9127197265625," and "651.420654296875," wires "[]}, {" id ":" 829b5feb.b4e86 "," type ":" debug "," z ":"
74a1122.f7518ec "," name "," "," active "false," console "," false "," complet e "," false "," x ": 992.0000043596538," and
"308.4285670689174," wires " []}, { "id": "829cae42.a 7edc", "type": "ui_gauge", "z": "74a1122.f7518ec", "name", "",
"group", "dcebd1d.9fff33 "" ord er ":1, "width", "8", "height", "8", "GType" "gage", "title": "SPEED", "label", "Knots",
"format": "{{value |
number: 0}}
"" Min "0," max ":" 15 "," colors ": [" # 00b500 "," # e6e600 "," # ca3838 "]," seg1 "," 7 "," seg2 ":" 12 "," x ":
613.0000076293945," and "296.9999952316284," wires "[]}, {" id ":" a227ddb.91dc52 "," type ":" UI_TEXT "," z ":"
74a1122.f7518ec "" group "" 1ce28a56.52d286 "," order "1," width "," 12 "," hei ght "," 1 "," name ":" NAME
BARCO "," label ":" NAME
BOAT "," format ":" {{}} msg.payload "," layout "," row-
center "," x ": 390.00000762939453," and "272.9999952316284," wires "[]}, {" id ":" 3d837045.fa bed "," type ":" UI_TEXT ","
z ":" 74a1122. f7518ec "," group ":" 72b69fa9.3c7ed "," order ": 1," width ":
0, "height" 0, "name", "SPEED", "label", "SPEED", "format": "{{msg.payload |
number: 1}} Knots "," layout ":" row-
right "," x ": 613.2500190734863," and "258.2499990463257," wires "[]}]

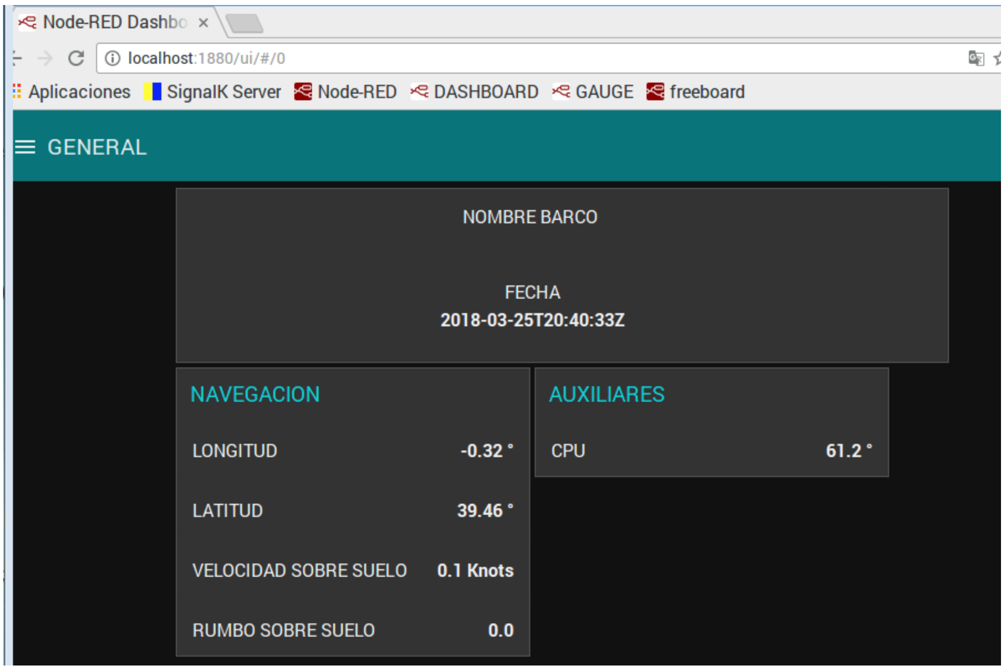
Once we have imported the code we have is to burn it, and this is done by selecting **Deploy**



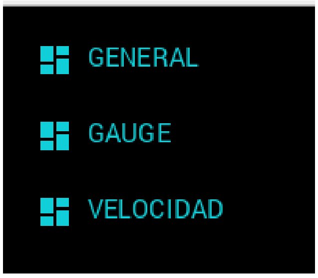
You are now recorded code, and what we have planned what we can see from

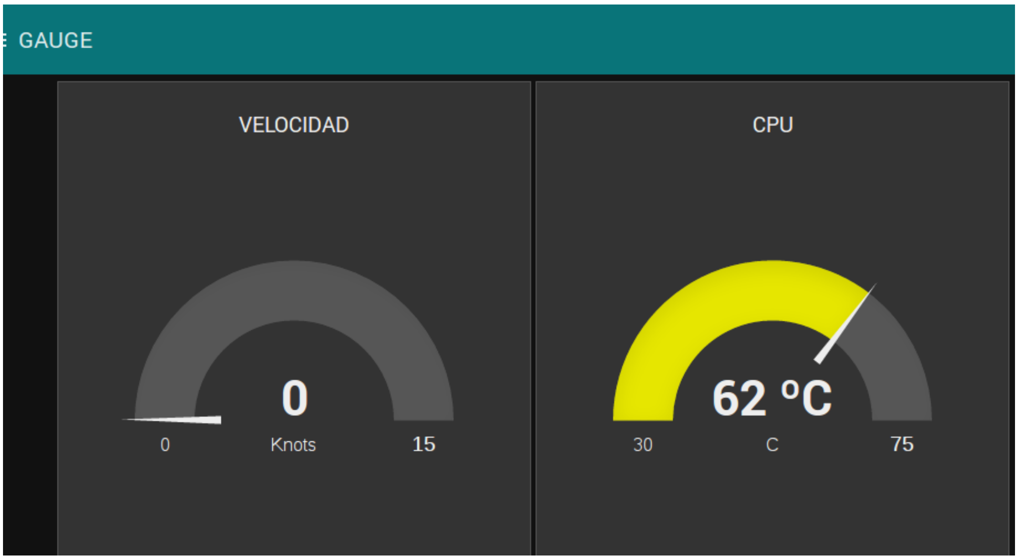


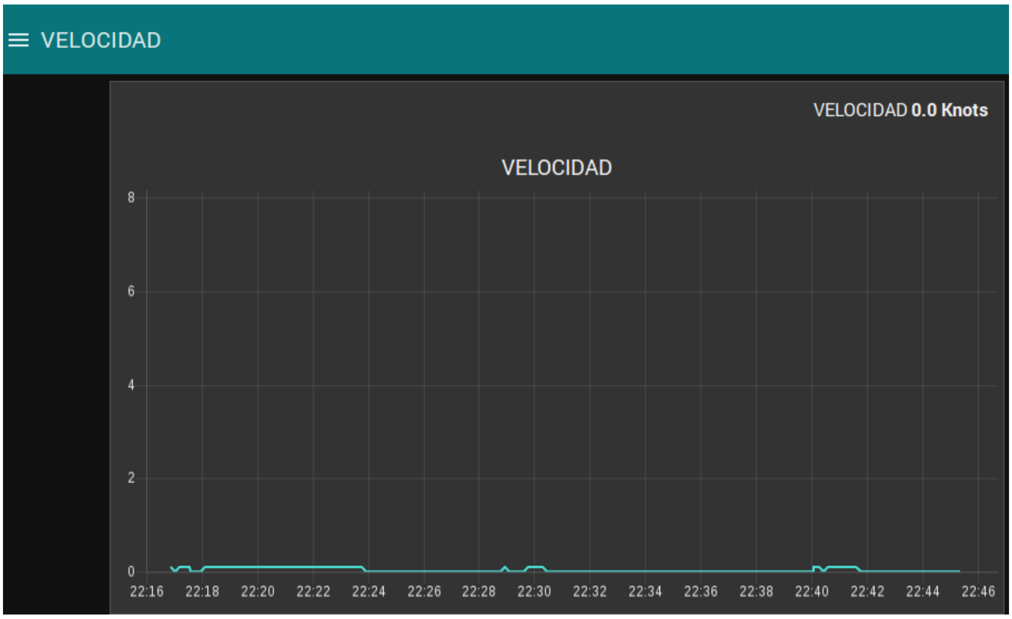
With what we will open,



If you select the menu at the top left deploy the pages.

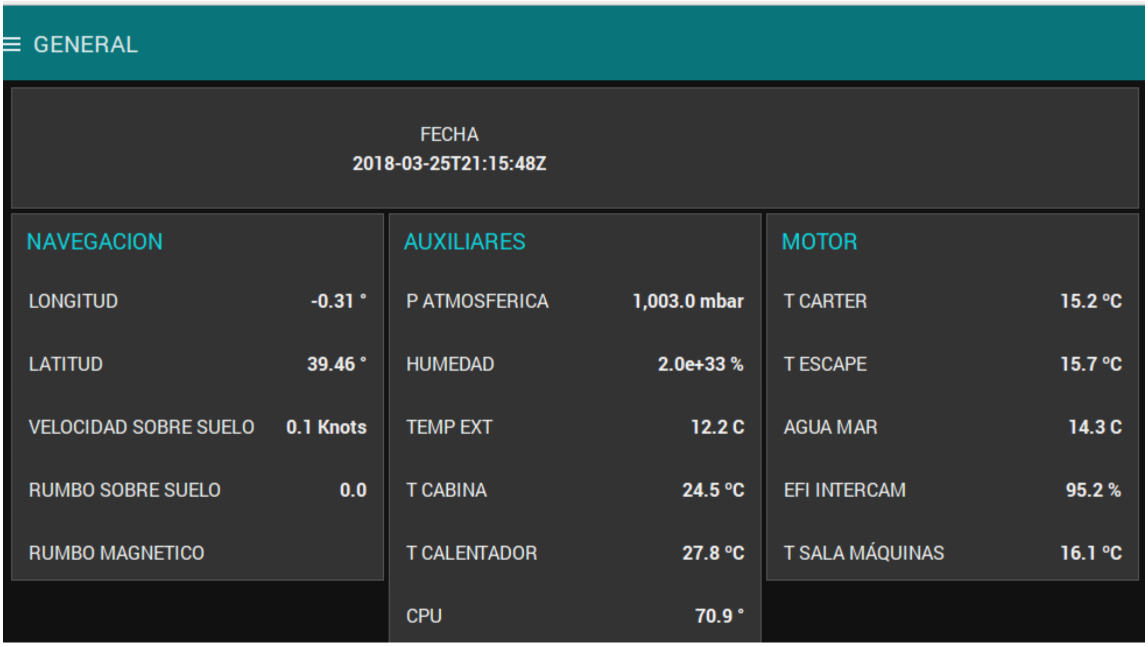


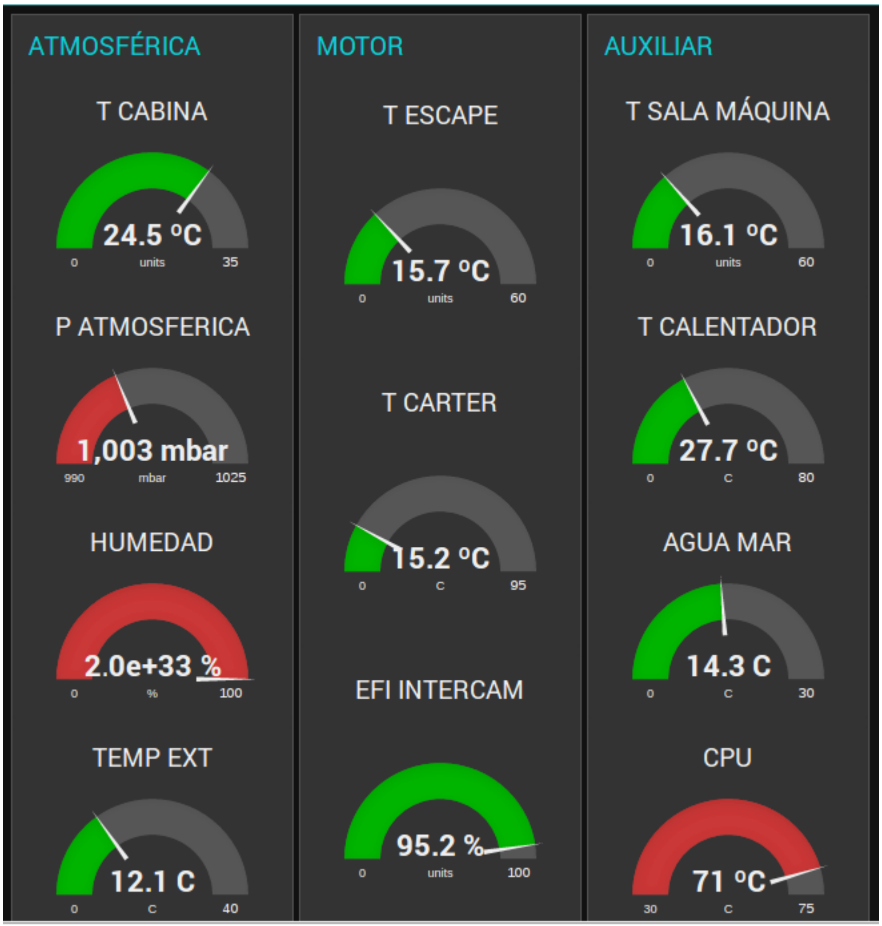


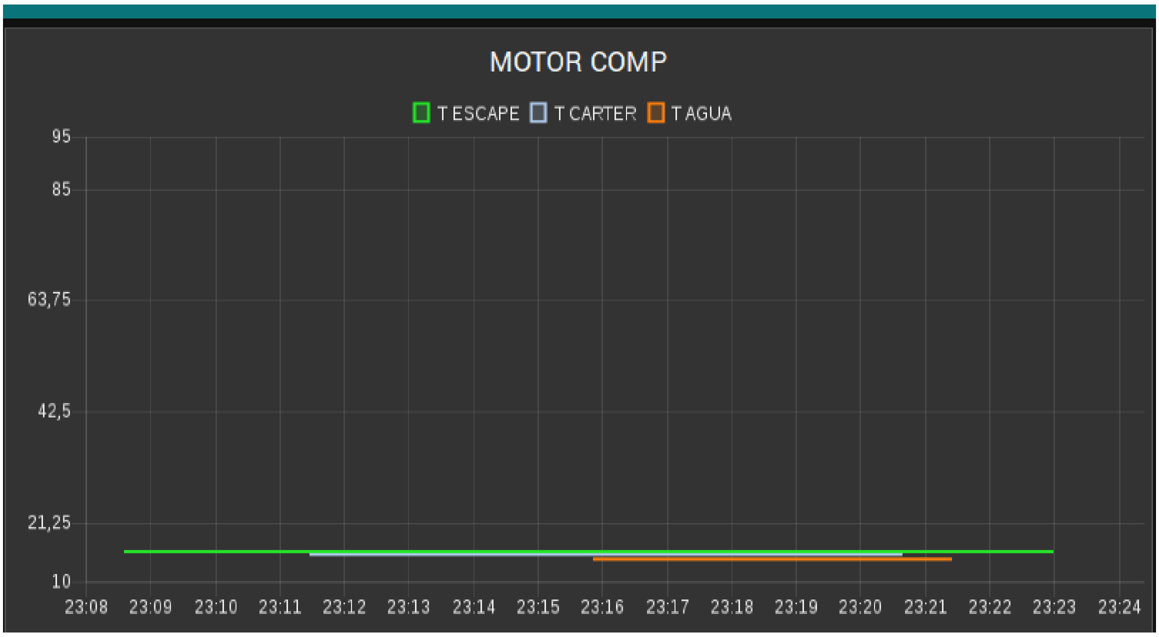


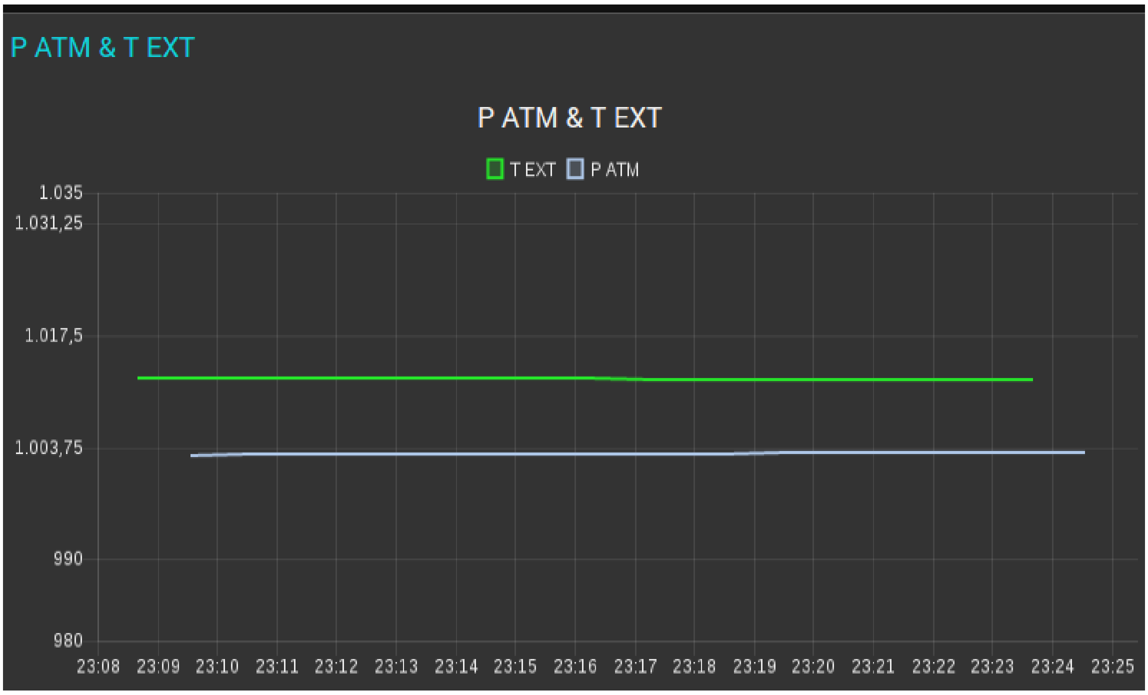
Here they are represented only data longitude, latitude, speed and temperature of the CPU, which are what have openplotter with a base configuration. If temperature sensors, pressure, humidity, battery voltage, etc. are added can represent the same way

As an example






MOTOR COMP
T ESCAPE
T CARTER
T AGUA
95
85
63,75
42,5
21,25
10
23:08 23:09 23:10 23:11 23:12 23:13 23:14 23:15 23:16 23:17 23:18 23:19 23:20 23:21 23:22 23:23 23:24


P ATM & T EXT
P ATM & T EXT
T EXT
P ATM
1.035
1.031,25
1.017,5
1.003,75
990
980
23:08 23:09 23:10 23:11 23:12 23:13 23:14 23:15 23:16 23:17 23:18 23:19 23:20 23:21 23:22 23:23 23:24 23:25