# Drop beer board computer

Open Plotter 2.0 setup guide

# Table of Contents

# Basic OP2 setup

1. Setup Arduino Nano as Drop Beer Board Computer control station
2. Install RPi imager on desktop
    a. https://www.raspberrypi.org/downloads/
    b. https://www.raspberrypi.org/blog/raspberry-pi-imager-imaging-utility/
3. Install non-NOOBS Raspberry Pi OS (32 bit) with desktop in an SD card
    a. https://www.raspberrypi.org/downloads/raspberry-pi-os/
4. Boot RPi
5. Copy SD card onto SSD card
6. Enable RPi to boot from SSD
7. In RPi config
    a. Change RPi password
    b. Hostname = DropBeer
    c. Set location/locale=english/us/ISO-8859
    d. Set timezone=america/los angelos
    e. Set keyboard=generic 105/english-us/english-us
    f. Set wifi=us
    g. Enable I2C
    h. Enable VNC

# OpenPlotter 2

8. Install OpenPlotter2
    a. https://github.com/openplotter/openplotter-settings/releases

   b. Rename menu item “Other” to “OpenPlotter”

9. Update RPi

   *a. sudo apt-get update*

   *b. sudo apt-get full-upgrade*

10. Install all OP2 packages, *except*

    a. Moitessier hat

    b. Kplex

    c. SDR VHF

## OpenCPN

11. Install OpenCPN

12. Open OpenCPN

13. Download maps

14. Prepare all charts

15. Load local race markers in separate layer

    a. Site: https://www.sequoiayc.org/sycracingmarkers

    b. File: https://www.sequoiayc.org/sites/default/files/SeqYC%20Racing%20Marks%202019-05.gpx_.txt

## SignalK

16. Install SignalK

17. Enable security by adding user/password

18. Install SignalK plugins:

    a. @signalk/udp-nmea-plugin

       i. BroadcastAddress=10.10.10.255

b. Nmea0183-to-nmea0183-plugin

c. Signalk-cloud

i. Obtain key from web site

d. signalk-raspberrypi-monitoring

e. signalk-tide-watch or signalk-tides-api

f. signalk-to-nmea2000

g. Signalk-wilhelmsk-plugin

i. Enable

h. @signalk/signal-node-red

i. signal-alarm-silencer

j. signal-notification-acker

k. influxdb

l. optional: sksim

19. Restart signalk

## Dashboards

20. Install Dashboards

a. Influxdb/Grafana

b. Node-red dashboard (optional)

## Setup networks: local and wide

21. Configure OP2: Network

a. AP and station (rpi3)

b. Sharing: auto

c. SSID: "GoFree-5670"

d. Password

22. Reboot

## Serial

## Can Bus

## PyPilot

## Setup UI panels

23. Add panel applet
    a. CPU Temperature Monitor
    b. CPU Usage Monitor
24. Setup applets as desired

## Install Drop Beer specifics

a. Insert prepared USB stick in RPi

b. On Rpi change directory into USB stick folder

c. Run ‘setup_dropbeer.sh’; this installs

   i. Power control interface

   ii. LCD screen configuration

        iii. Wifi stay alive

        iv. Artik fan controls

        v. Arduino development environment

        vi. ….

## Setup Artik fan

25. Artik fan specific setup

    a. 40º: 20%

    b. 45º: 40%

    c. 50º: 60%

    d. 55º: 100%

## Setup SignalK

26. Configure OP2: SignalK

    a. Vessel data

27. Configure OP2: Server plugins

    a. Convert SignalK to NMEA0183

        i. Enable

        ii. Enable pretty much everything

    b. Convert SignalK to NMEA2000

    c. Edit zones ???

    d. Set system time

    e.

28. Add signal sources

    a. Engine_sensor

        i. Server/connection/port:55557

## VNC

29. RPI config: enable VNC

30.

## Gmail

Setup GMAIL

Open browser

Setup user: dropbeerboardcomputer@gmail.com

Password: …

## Drop Beer node-red

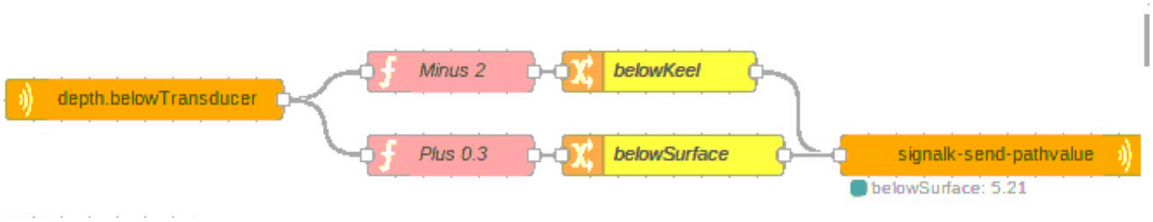
31. Add "e-mail" palette

32. Import flow by cut/paste from filevault

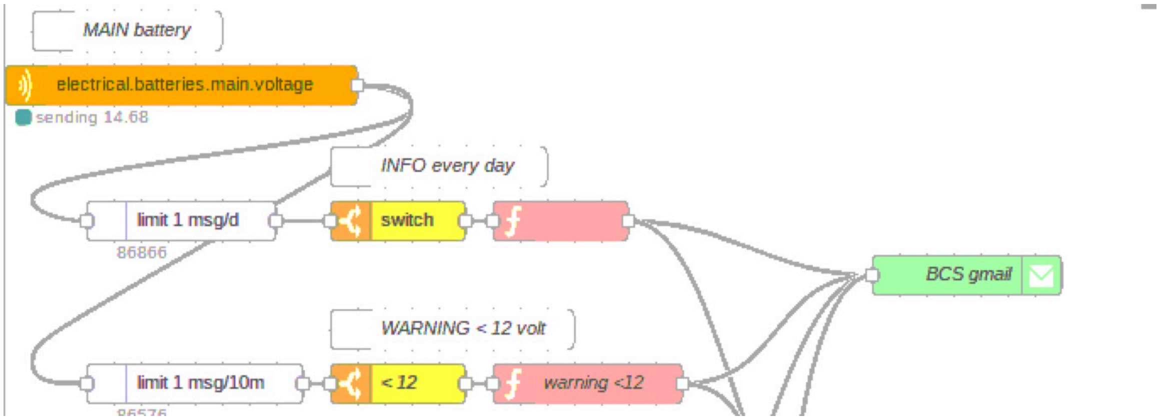
    a. Drop Beer node-red function

    *a.* Derive depth below keel
    *b.* Derive depth below surface
    *c.* Check MAIN and HOUSE batteries every 24 hours
    *d.* Check MAIN and HOUSE batteries for reaching 12 volts, email warning if so
    *e.* Setup emailer

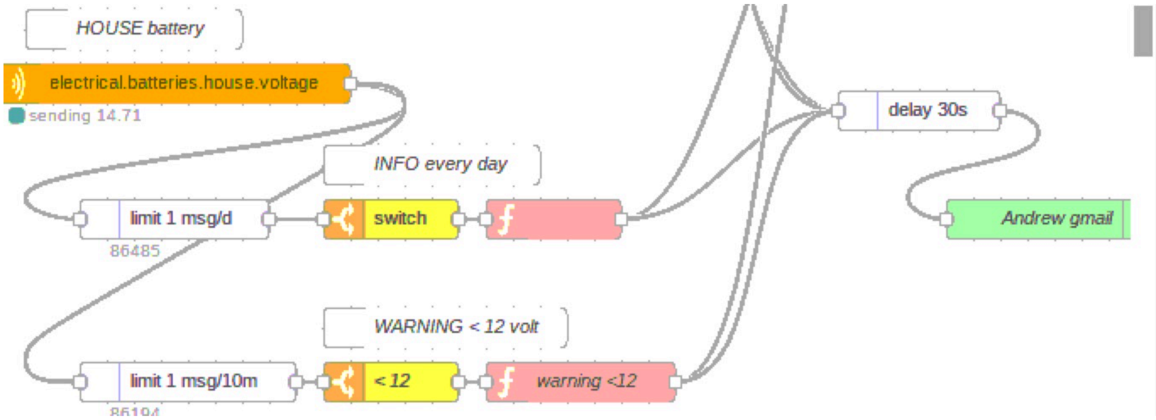
*i.* Open each email node, add < from email> and <from password>, redeploy. Check the messages pass



*f.*



*g.*



*h.*

## Setup influxdb

Assign database name: "dropbeerdb"

## Setup Grafana

Assure that when opening the database “dropbeerdb”, that i.s.o. “localhost” it references “127.0.0.1”. Otherwise it appears to work, but the database is empty.

## Mailer

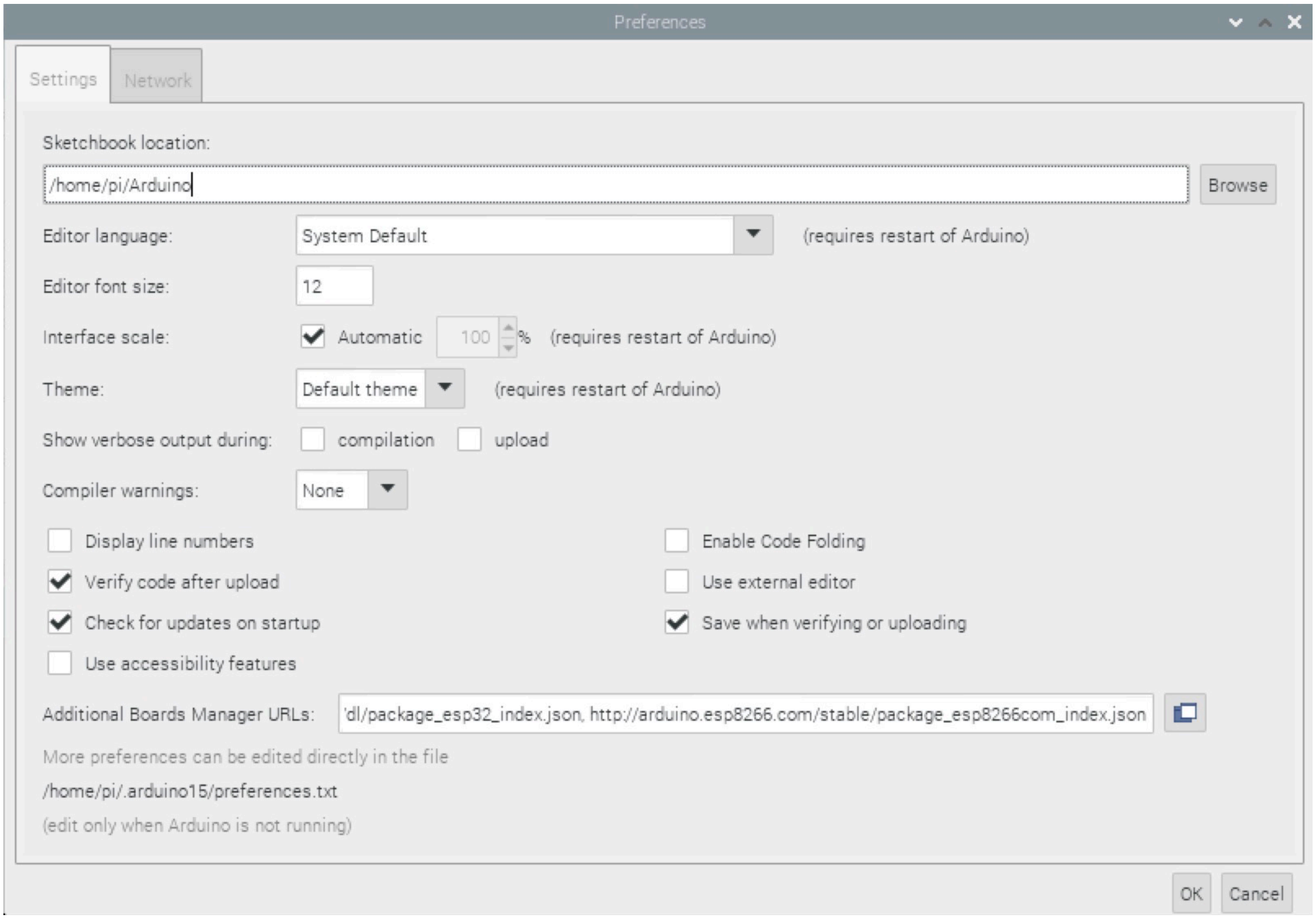
Not needed/working for GRIB file exchange, use the “XyGrib GRIB file viewer” for that purpose

## Arduino

### engine_sensor.ino

Add ESP8266 libraries:

> https://dl.espressif.com/dl/package_esp32_index.json ,
> http://arduino.esp8266.com/stable/package_esp8266com_index.json



Sketchbook location:
/home/pi/Arduino

Editor language: System Default (requires restart of Arduino)

Editor font size: 12

Interface scale: ☑ Automatic 100 % (requires restart of Arduino)

Theme: Default theme (requires restart of Arduino)

Show verbose output during: □ compilation □ upload

Compiler warnings: None

□ Display line numbers
☑ Verify code after upload
☑ Check for updates on startup
□ Use accessibility features

□ Enable Code Folding
□ Use external editor
☑ Save when verifying or uploading

Additional Boards Manager URLs: 'dl/package_esp32_index.json, http://arduino.esp8266.com/stable/package_esp8266com_index.json

More preferences can be edited directly in the file
/home/pi/.arduino15/preferences.txt
(edit only when Arduino is not running)

Arduino IDE Parameters:

* #ide_param:board=NodeMCU 1.0 (ESP-12E Module)

*

* Use compile, then export compiled library to local file location

* Find engine_sensor node using "sudo nmap –sn 10.10.10.1/24"

* This will show the engine_sensor node at 10.10.10.101, assuming the

* static IP address has not been modified

* Upload compiled sketch to: http://10.10.10.101:8080/webota

33. Copy all files from usb stick to ~/arduino

34. Compile/upload both nodes (wired)

35. Later, for updates, in order to find engine_sensor ip address

    a. sudo apt-get install nmap

    b. sudo nmap –sn 10.10.10.1/24

## DropBeer.ino (Power Control Unit)

```
/* Manually set the correct IDE parameters...
 *
 *  Arduino IDE Parameters:
 *  #ide_param:board=Arduino Nano
 *  #ide_param:processor=ATmega328P
 *  #ide_param:port=ttyUSB1 (or whatever ttyOP_ard points to)
*/
```

# Library customizations

## Adafruit_INA219.h

```
/**************************************************************************/
/*!
    @brief  Class that stores state and functions for interacting with INA219 current/power monitor IC
*/
/**************************************************************************/
class Adafruit_INA219{
 public:
  Adafruit_INA219(uint8_t addr = INA219_ADDRESS);
  void begin(void);
  void begin(TwoWire *theWire);
  void setCalibration_32V_2A(void);
  void setCalibration_32V_1A(void);
  void setCalibration_16V_400mA(void);
  void setCalibration_16V_160A(void);
  float getBusVoltage_V(void);
  float getShuntVoltage_mV(void);
  float getCurrent_mA(void);
  float getPower_mW(void);
```

## Adafruit_INA219.cpp

```
/**************************************************************************/
/*!
 @brief  Configures to INA219 to be able to measure up to 16V and 160A
 of current.

 @note   These calculations assume a 0.00075 ohm resistor is present
 */
/**************************************************************************/
void Adafruit_INA219::setCalibration_16V_160A(void) {

    ina219_calValue = 32768;
    ina219_currentDivider_mA = 320;    // Current LSB = 50uA per bit (1000/50 = 20)
    ina219_powerMultiplier_mW = 0.10f; // Power LSB = 1mW per bit

    wireWriteRegister(INA219_REG_CALIBRATION, ina219_calValue);

    uint16_t config = INA219_CONFIG_BVOLTAGERANGE_16V |
    INA219_CONFIG_GAIN_1_40MV |
    INA219_CONFIG_BADCRES_12BIT |
    INA219_CONFIG_SADCRES_12BIT_1S_532US |
    INA219_CONFIG_MODE_SANDBVOLT_CONTINUOUS;
    wireWriteRegister(INA219_REG_CONFIG, config);
}
```

# MD_MAX72XX.h

```
class MD_MAX72XX
{
public:
  /**
  * Module Type enumerated type.
  *
  * This enumerated type is used to defined the type of
  * modules being used in the application. The types of modules are
  * discussed in detail in the Hardware section of this documentation.
  */
  enum moduleType_t
  {
    PAROLA_HW,    ///< Use the Parola style hardware modules.
    GENERIC_HW,   ///< Use 'generic' style hardware modules commonly available.
    ICSTATION_HW, ///< Use ICStation style hardware module.
    FC16_HW,      ///< Use FC-16 style hardware module.
    DB_HW       // what I actually have
  };
```

# MD_MAX72XX.cpp

Seems I can just use ICSTATION, no need for customization.

```
void MD_MAX72XX::setModuleParameters(moduleType_t mod)
// Combinations not listed here have probably not been tested and may
// not operate correctly.
{
  _mod = mod;
  switch (_mod)
  {
    case PAROLA_HW:    _hwDigRows = true;  _hwRevCols = true;  _hwRevRows = false; break; // tested MC 8 March 2014
    case GENERIC_HW:   _hwDigRows = false; _hwRevCols = true;  _hwRevRows = false; break; // tested MC 9 March 2014
    case ICSTATION_HW: _hwDigRows = true;  _hwRevCols = true;  _hwRevRows = true;  break; // tested MC 9 March 2014
    case FC16_HW:      _hwDigRows = true;  _hwRevCols = false; _hwRevRows = false; break; // tested MC 23 Feb 2015
    case DB_HW:         _hwDigRows = true;  _hwRevCols = true;  _hwRevRows = true; break; // what I actually have
   default:          _hwDigRows = _hwRevRows = _hwRevCols = false; break;   // not a known board config
  }
}
```

## Backup copy

Use USB stick with SD card option

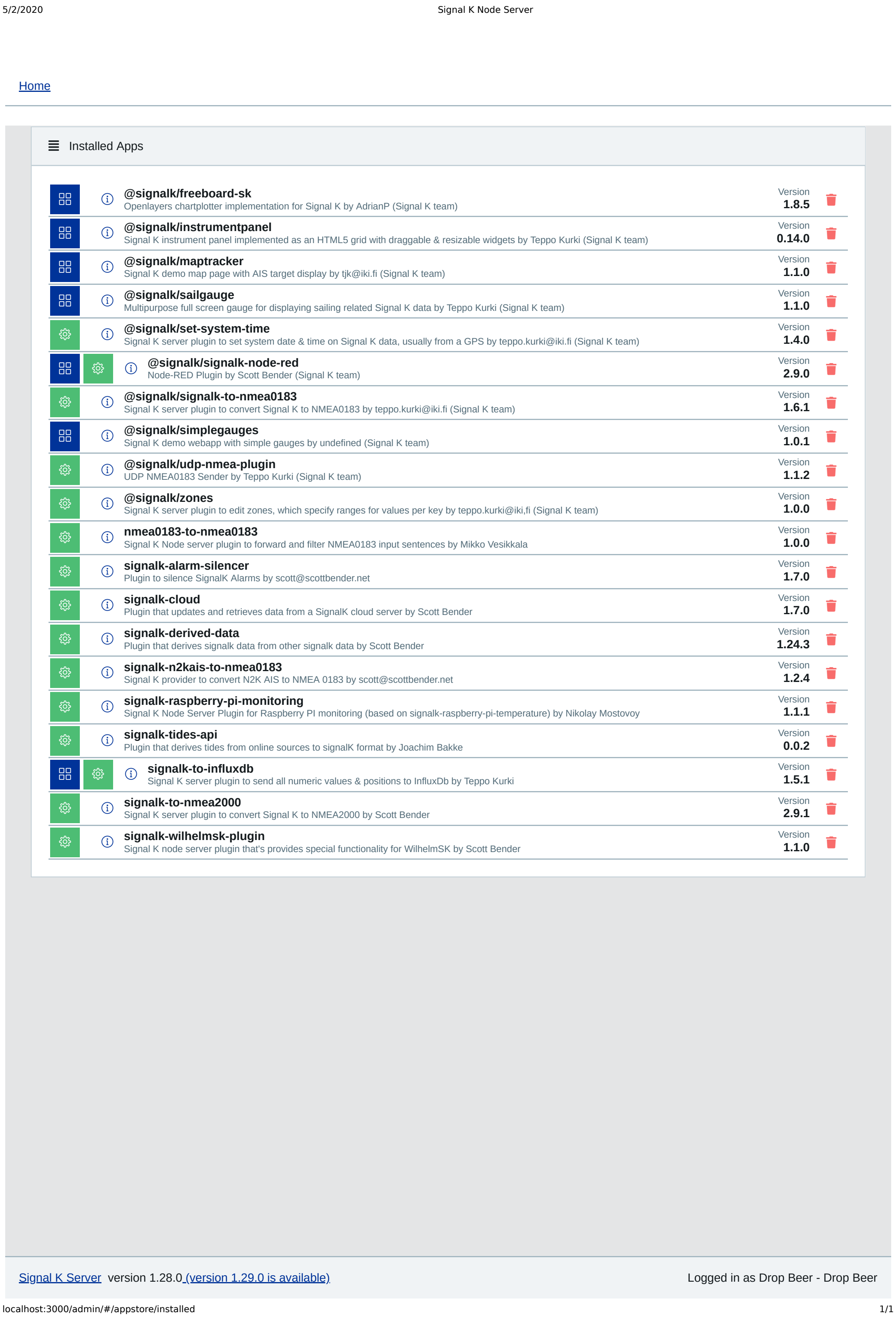
Run Accessories/SD Card Copier

## Passwords

| Application | Username | Password |
|---|---|---|
| **Raspberry pi** | pi | degroeten |
| **signalk** | Drop Beer | degroeten |
| **Grafana** | admin | degroeten |
| **SignalK cloud token** | | |
| **VNC** | beercansailor@gmail.com | |
| **WiFi to shore** | | |

## SignalK setup

### Installed apps



5/2/2020 Signal K Node Server

Home

Installed Apps

| | | |
|---|---|---|
| @signalk/freeboard-sk | Openlayers chartplotter implementation for Signal K by AdrianP (Signal K team) | Version<br>1.8.5 |
| @signalk/instrumentpanel | Signal K instrument panel implemented as an HTML5 grid with draggable & resizable widgets by Teppo Kurki (Signal K team) | Version<br>0.14.0 |
| @signalk/maptracker | Signal K demo map page with AIS target display by tjk@iki.fi (Signal K team) | Version<br>1.1.0 |
| @signalk/sailgauge | Multipurpose full screen gauge for displaying sailing related Signal K data by Teppo Kurki (Signal K team) | Version<br>1.1.0 |
| @signalk/set-system-time | Signal K server plugin to set system date & time on Signal K data, usually from a GPS by teppo.kurki@iki.fi (Signal K team) | Version<br>1.4.0 |
| @signalk/signalk-node-red | Node-RED Plugin by Scott Bender (Signal K team) | Version<br>2.9.0 |
| @signalk/signalk-to-nmea0183 | Signal K server plugin to convert Signal K to NMEA0183 by teppo.kurki@iki.fi (Signal K team) | Version<br>1.6.1 |
| @signalk/simplegauges | Signal K demo webapp with simple gauges by undefined (Signal K team) | Version<br>1.0.1 |
| @signalk/udp-nmea-plugin | UDP NMEA0183 Sender by Teppo Kurki (Signal K team) | Version<br>1.1.2 |
| @signalk/zones | Signal K server plugin to edit zones, which specify ranges for values per key by teppo.kurki@iki,fi (Signal K team) | Version<br>1.0.0 |
| nmea0183-to-nmea0183 | Signal K Node server plugin to forward and filter NMEA0183 input sentences by Mikko Vesikkala | Version<br>1.0.0 |
| signalk-alarm-silencer | Plugin to silence SignalK Alarms by scott@scottbender.net | Version<br>1.7.0 |
| signalk-cloud | Plugin that updates and retrieves data from a SignalK cloud server by Scott Bender | Version<br>1.7.0 |
| signalk-derived-data | Plugin that derives signalk data from other signalk data by Scott Bender | Version<br>1.24.3 |
| signalk-n2kais-to-nmea0183 | Signal K provider to convert N2K AIS to NMEA 0183 by scott@scottbender.net | Version<br>1.2.4 |
| signalk-raspberry-pi-monitoring | Signal K Node Server Plugin for Raspberry PI monitoring (based on signalk-raspberry-pi-temperature) by Nikolay Mostovoy | Version<br>1.1.1 |
| signalk-tides-api | Plugin that derives tides from online sources to signalK format by Joachim Bakke | Version<br>0.0.2 |
| signalk-to-influxdb | Signal K server plugin to send all numeric values & positions to InfluxDb by Teppo Kurki | Version<br>1.5.1 |
| signalk-to-nmea2000 | Signal K server plugin to convert Signal K to NMEA2000 by Scott Bender | Version<br>2.9.1 |
| signalk-wilhelmsk-plugin | Signal K node server plugin that's provides special functionality for WilhelmSK by Scott Bender | Version<br>1.1.0 |

Signal K Server version 1.28.0 (version 1.29.0 is available) Logged in as Drop Beer - Drop Beer

localhost:3000/admin/#/appstore/installed 1/1

## Server/Vessel data

## Server/Connections + engine_sensor

## Server/Plugin Config/SKtoNMEA0183

Home

› Alarm Silencer

⌄ Convert Signal K to NMEA0183

Package Name: @signalk/signalk-to-nmea0183
Status: Started

☑ Active

□ Enable Logging

□ Enable Debug

If there is SK data for the conversion generate the following NMEA0183 sentences from Signal K data:

□ APB - Autopilot info

☑ DBK - Depth Below Keel

☑ DBS - Depth Below Surface

☑ DBT - Depth Below Transducer

☑ DPT - Depth

☑ GGA - Time, position, and fix related data

☑ GLL - Geographical position, latitude and longitude

☑ HDG - Heading magnetic:.

☑ HDM - Heading Magnetic

☑ HDM - Heading Magnetic, calculated from True

☑ HDT - Heading True

☑ HDT - Heading True calculated from magnetic heading and variation

☑ MMB - Environment outside pressure

☑ MTA - Air temperature.

☑ MTW - Water Temperature

☑ MWD - Wind relative to North, speed might be ground speed.

☑ MWV - Aparent Wind heading and speed

☑ MWV - True Wind heading and speed

□ PNKEP,01 - Target Polar speed

□ PNKEP,02 - Course (COG) on other tack from 0 to 359°

□ PNKEP,03 - Polar and VMG, and optimum angle.

□ PNKEP,99 - Debug

□ PSILCD1 - Send polar speed and target wind angle to Silva/Nexus/Garmin displays

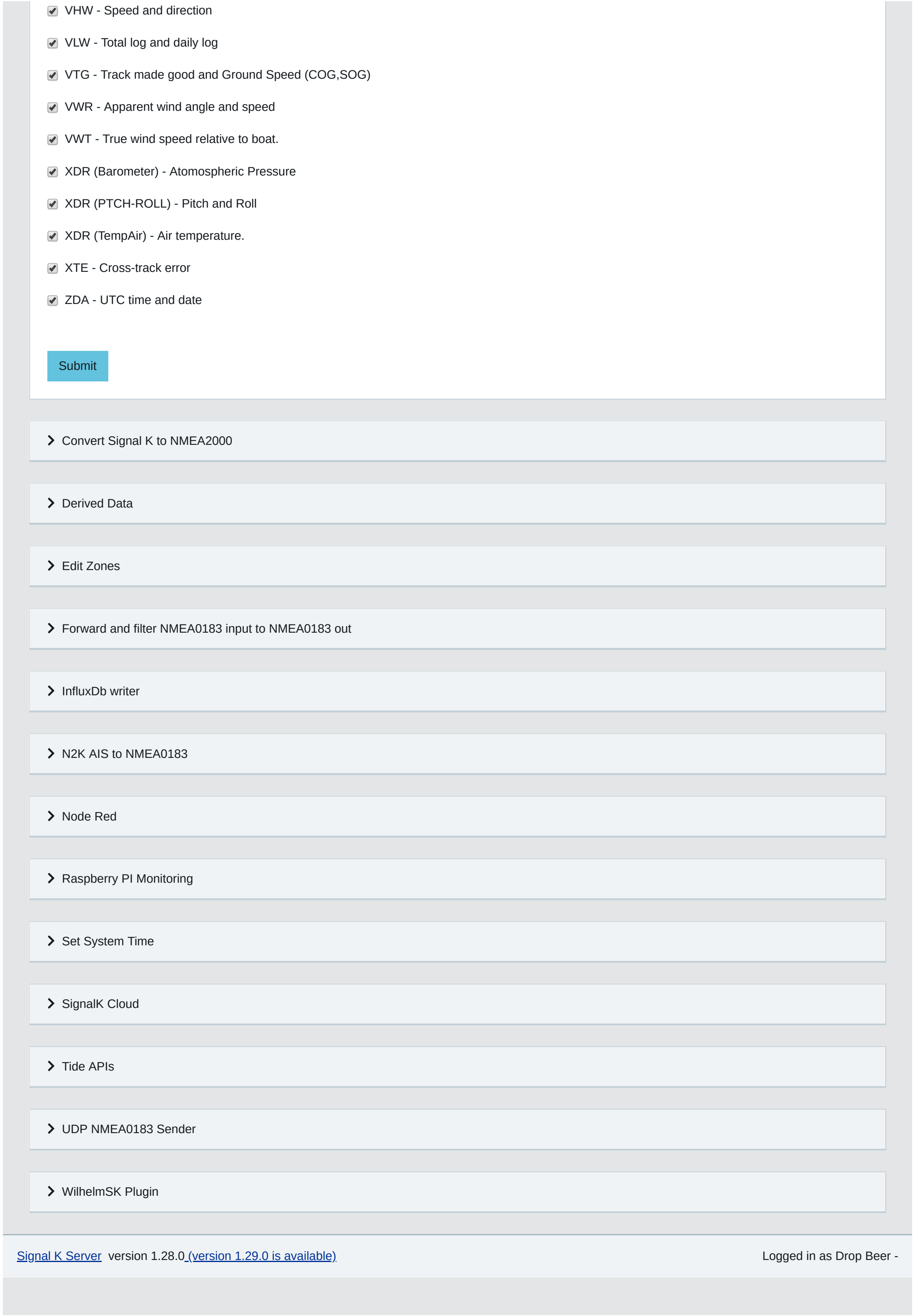
□ PSILTBS - Garmin proprietary target boat speed

☑ RMB - Heading and distance to waypoint

☑ RMC - GPS recommended minimum

☑ ROT - Rate of Turn

☑ RSA - Rudder Sensor Angle



☑ VHW - Speed and direction

☑ VLW - Total log and daily log

☑ VTG - Track made good and Ground Speed (COG,SOG)

☑ VWR - Apparent wind angle and speed

☑ VWT - True wind speed relative to boat.

☑ XDR (Barometer) - Atomospheric Pressure

☑ XDR (PTCH-ROLL) - Pitch and Roll

☑ XDR (TempAir) - Air temperature.

☑ XTE - Cross-track error

☑ ZDA - UTC time and date

› Convert Signal K to NMEA2000

› Derived Data

› Edit Zones

› Forward and filter NMEA0183 input to NMEA0183 out

› InfluxDb writer

› N2K AIS to NMEA0183

› Node Red

› Raspberry PI Monitoring

› Set System Time

› SignalK Cloud

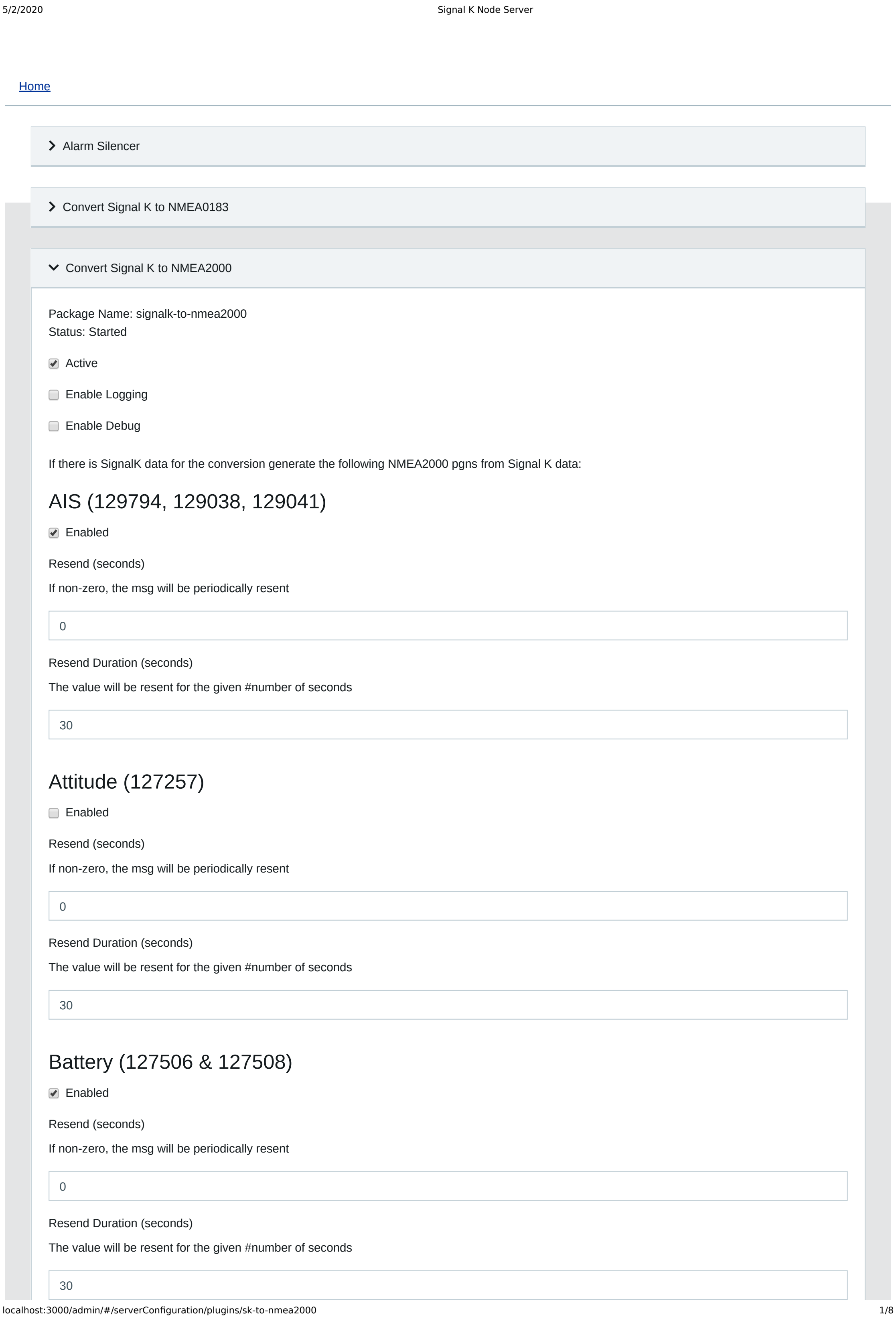
› Tide APIs

› UDP NMEA0183 Sender

› WilhelmSK Plugin

Signal K Server version 1.28.0 (version 1.29.0 is available) Logged in as Drop Beer -

## Server/Plugin Config/SKtoNMEA2000



Home

› Alarm Silencer

› Convert Signal K to NMEA0183

⌄ Convert Signal K to NMEA2000

Package Name: signalk-to-nmea2000
Status: Started

☑ Active

□ Enable Logging

□ Enable Debug

If there is SignalK data for the conversion generate the following NMEA2000 pgns from Signal K data:

## AIS (129794, 129038, 129041)

☑ Enabled

Resend (seconds)

If non-zero, the msg will be periodically resent

0

Resend Duration (seconds)

The value will be resent for the given #number of seconds

30

## Attitude (127257)

□ Enabled

Resend (seconds)

If non-zero, the msg will be periodically resent

0

Resend Duration (seconds)

The value will be resent for the given #number of seconds

30

## Battery (127506 & 127508)

☑ Enabled

Resend (seconds)

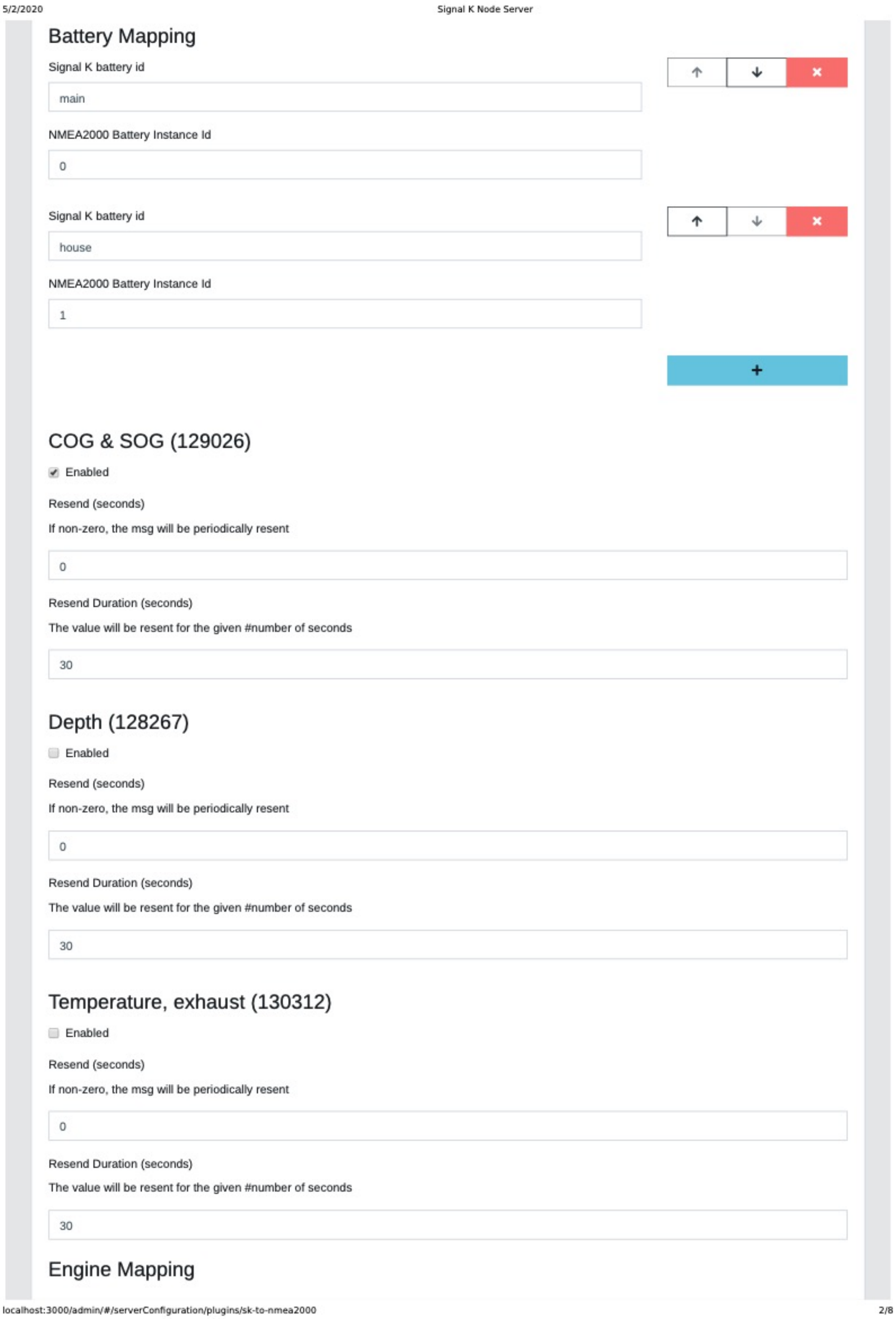
If non-zero, the msg will be periodically resent

0

Resend Duration (seconds)

The value will be resent for the given #number of seconds

30



## Battery Mapping

Signal K battery id

main

NMEA2000 Battery Instance Id

0

Signal K battery id

house

NMEA2000 Battery Instance Id

1

## COG & SOG (129026)

☑ Enabled

Resend (seconds)

If non-zero, the msg will be periodically resent

0

Resend Duration (seconds)

The value will be resent for the given #number of seconds

30

## Depth (128267)

☐ Enabled

Resend (seconds)

If non-zero, the msg will be periodically resent

0

Resend Duration (seconds)

The value will be resent for the given #number of seconds

30

## Temperature, exhaust (130312)

☐ Enabled

Resend (seconds)

If non-zero, the msg will be periodically resent

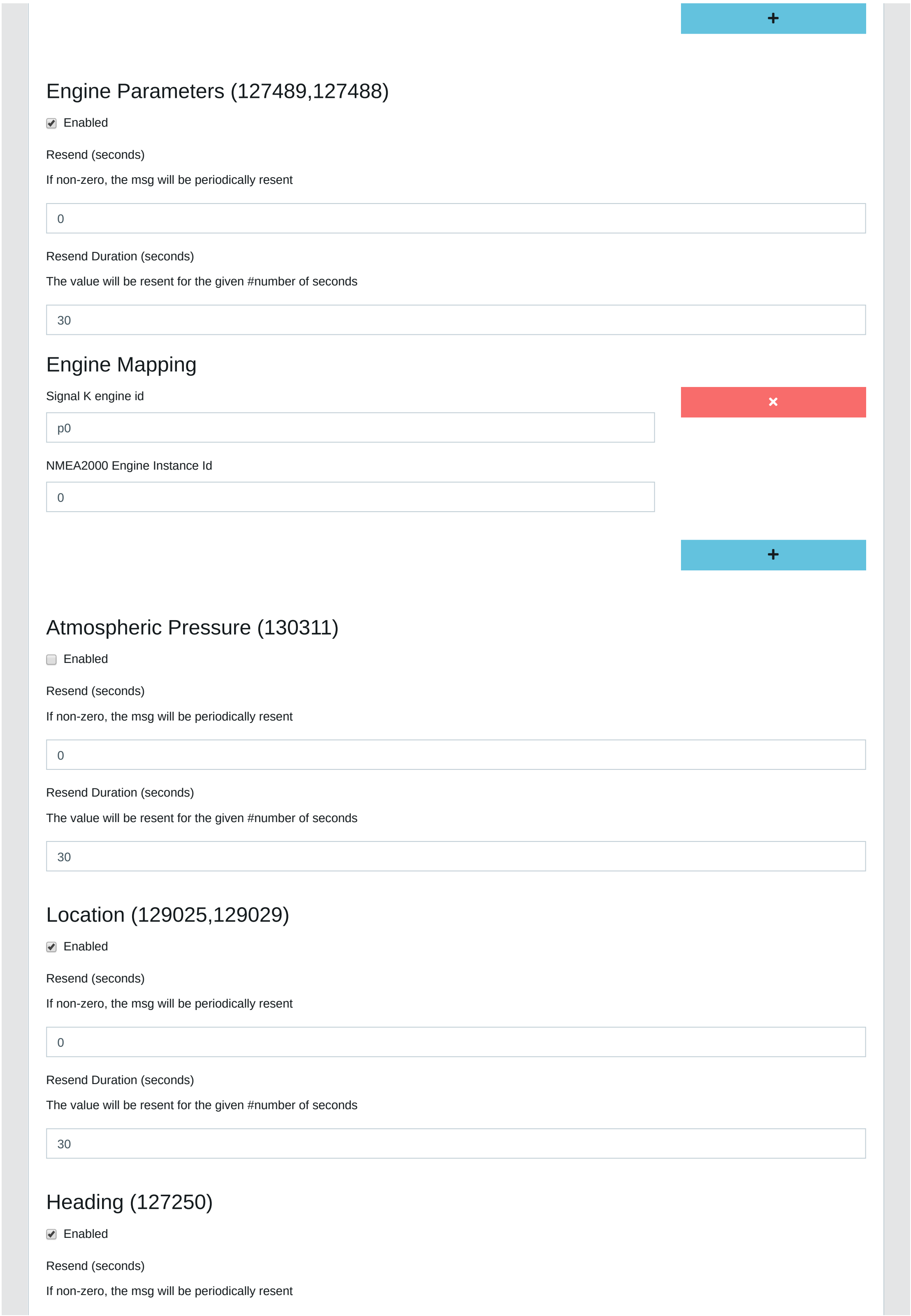
0

Resend Duration (seconds)

The value will be resent for the given #number of seconds

30

## Engine Mapping



## Engine Parameters (127489,127488)

☑ Enabled

Resend (seconds)

If non-zero, the msg will be periodically resent

0

Resend Duration (seconds)

The value will be resent for the given #number of seconds

30

## Engine Mapping

Signal K engine id

p0

NMEA2000 Engine Instance Id

0

## Atmospheric Pressure (130311)

☐ Enabled

Resend (seconds)

If non-zero, the msg will be periodically resent

0

Resend Duration (seconds)

The value will be resent for the given #number of seconds

30

## Location (129025,129029)

☑ Enabled

Resend (seconds)

If non-zero, the msg will be periodically resent

0

Resend Duration (seconds)

The value will be resent for the given #number of seconds

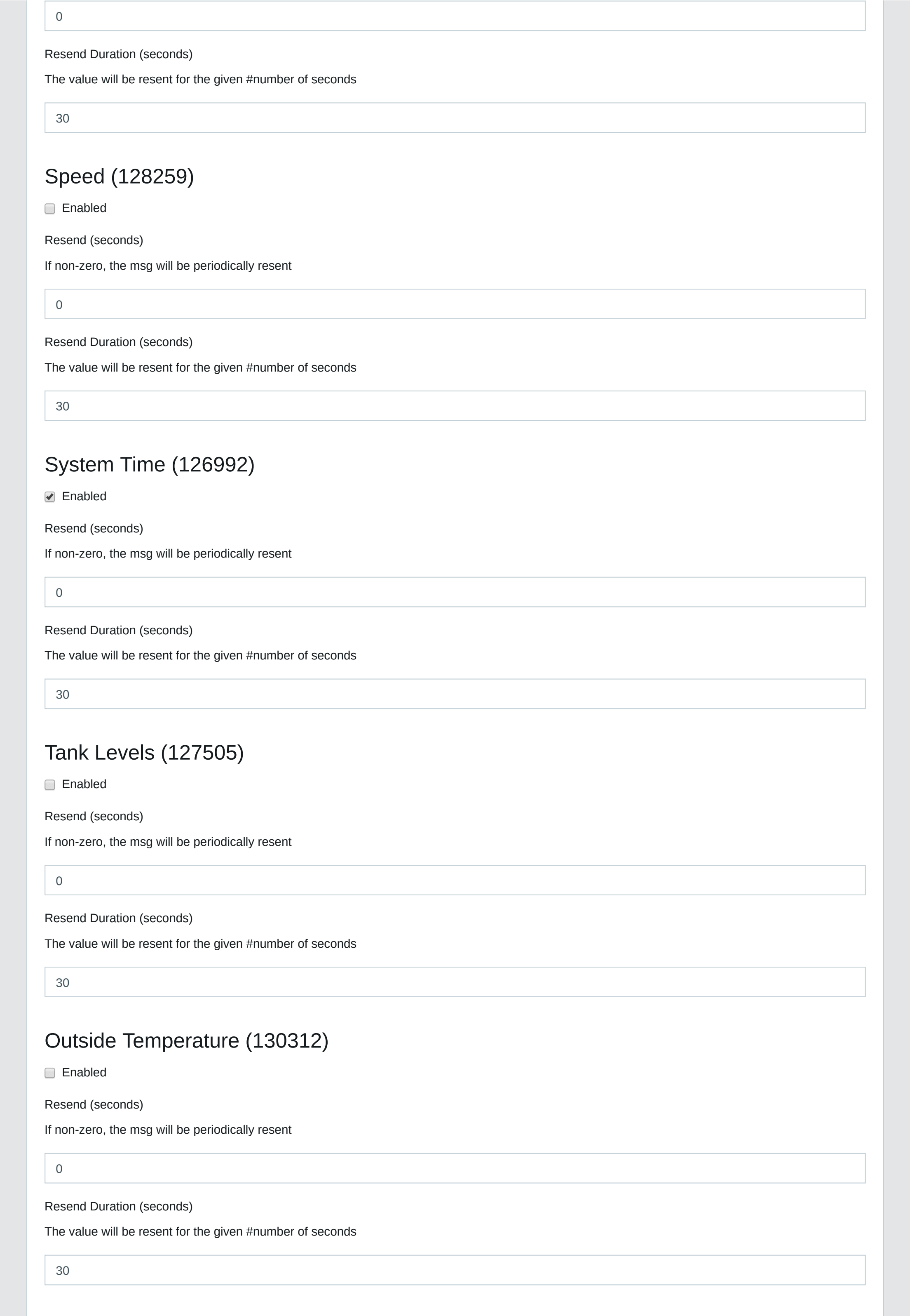
30

## Heading (127250)

☑ Enabled

Resend (seconds)

If non-zero, the msg will be periodically resent



0

Resend Duration (seconds)

The value will be resent for the given #number of seconds

30

### Speed (128259)

☐ Enabled

Resend (seconds)

If non-zero, the msg will be periodically resent

0

Resend Duration (seconds)

The value will be resent for the given #number of seconds

30

### System Time (126992)

☑ Enabled

Resend (seconds)

If non-zero, the msg will be periodically resent

0

Resend Duration (seconds)

The value will be resent for the given #number of seconds

30

### Tank Levels (127505)

☐ Enabled

Resend (seconds)

If non-zero, the msg will be periodically resent

0

Resend Duration (seconds)

The value will be resent for the given #number of seconds

30

### Outside Temperature (130312)

☐ Enabled

Resend (seconds)

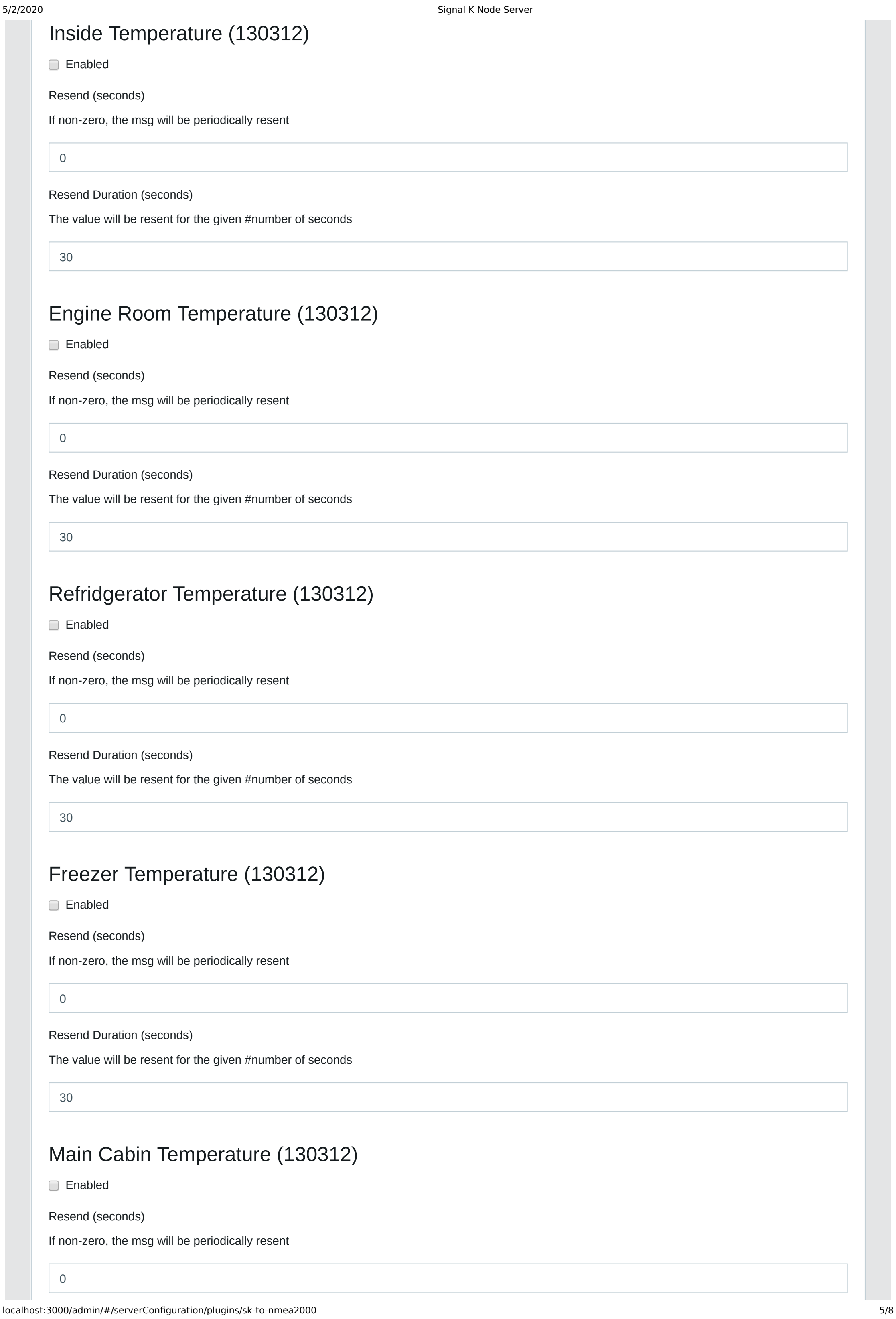
If non-zero, the msg will be periodically resent

0

Resend Duration (seconds)

The value will be resent for the given #number of seconds

30



## Inside Temperature (130312)

□ Enabled

Resend (seconds)

If non-zero, the msg will be periodically resent

0

Resend Duration (seconds)

The value will be resent for the given #number of seconds

30

## Engine Room Temperature (130312)

□ Enabled

Resend (seconds)

If non-zero, the msg will be periodically resent

0

Resend Duration (seconds)

The value will be resent for the given #number of seconds

30

## Refridgerator Temperature (130312)

□ Enabled

Resend (seconds)

If non-zero, the msg will be periodically resent

0

Resend Duration (seconds)

The value will be resent for the given #number of seconds

30

## Freezer Temperature (130312)

□ Enabled

Resend (seconds)

If non-zero, the msg will be periodically resent

0

Resend Duration (seconds)

The value will be resent for the given #number of seconds

30

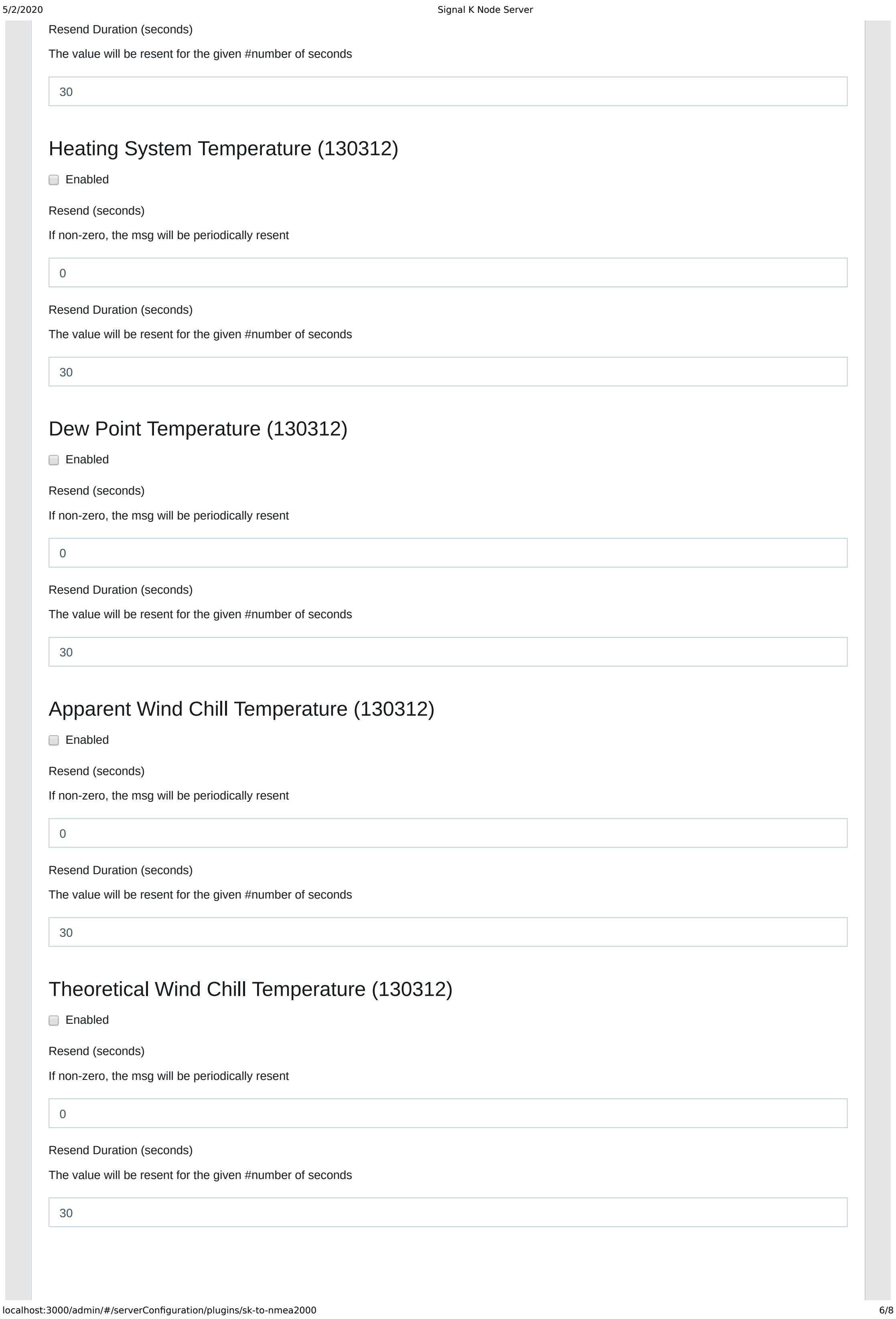
## Main Cabin Temperature (130312)

□ Enabled

Resend (seconds)

If non-zero, the msg will be periodically resent

0



Resend Duration (seconds)

The value will be resent for the given #number of seconds

30

## Heating System Temperature (130312)

☐ Enabled

Resend (seconds)

If non-zero, the msg will be periodically resent

0

Resend Duration (seconds)

The value will be resent for the given #number of seconds

30

## Dew Point Temperature (130312)

☐ Enabled

Resend (seconds)

If non-zero, the msg will be periodically resent

0

Resend Duration (seconds)

The value will be resent for the given #number of seconds

30

## Apparent Wind Chill Temperature (130312)

☐ Enabled

Resend (seconds)

If non-zero, the msg will be periodically resent

0

Resend Duration (seconds)

The value will be resent for the given #number of seconds

30

## Theoretical Wind Chill Temperature (130312)

☐ Enabled

Resend (seconds)

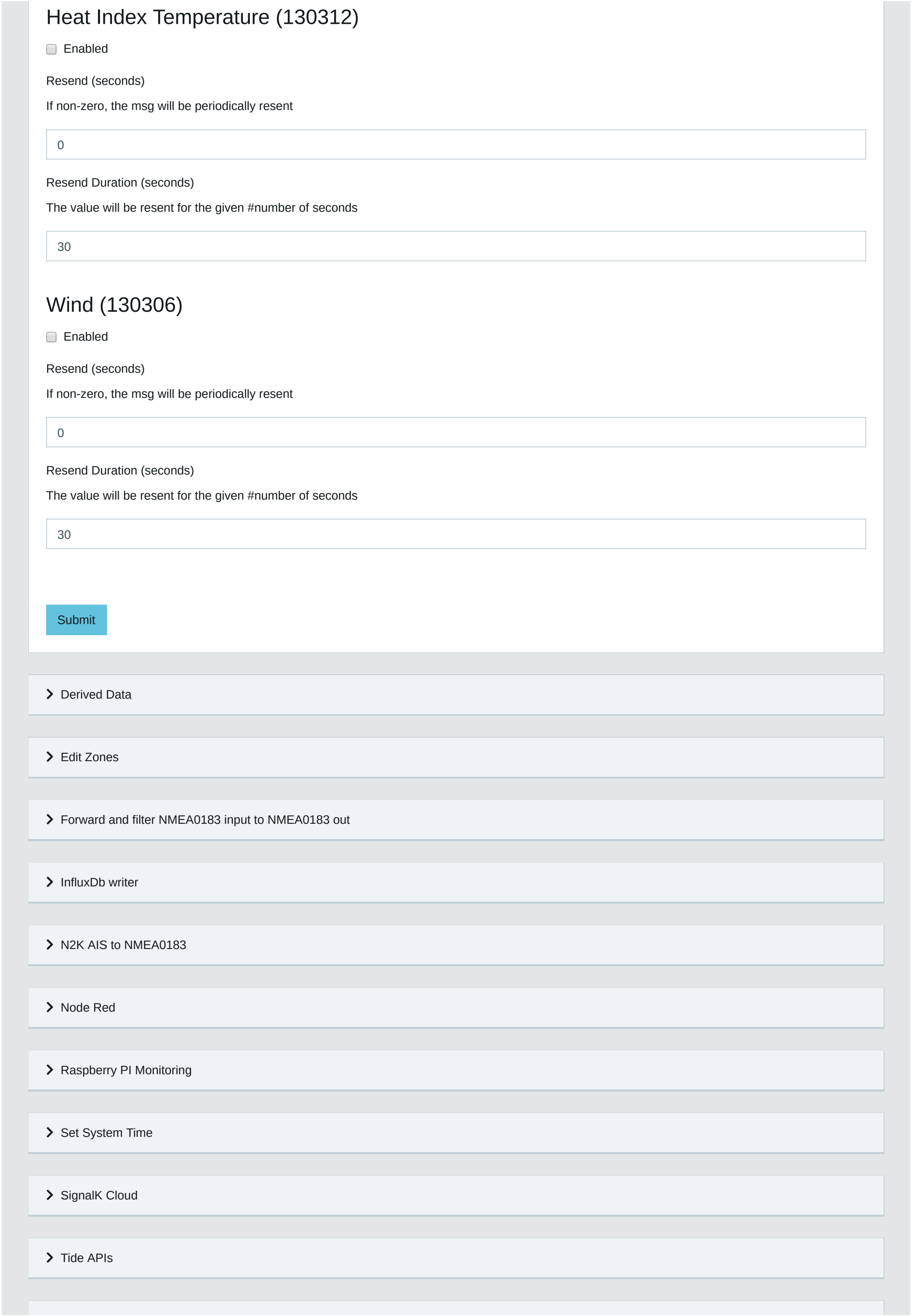
If non-zero, the msg will be periodically resent

0

Resend Duration (seconds)

The value will be resent for the given #number of seconds

30



### Heat Index Temperature (130312)

☐ Enabled

Resend (seconds)

If non-zero, the msg will be periodically resent

0

Resend Duration (seconds)

The value will be resent for the given #number of seconds

30

### Wind (130306)

☐ Enabled

Resend (seconds)

If non-zero, the msg will be periodically resent

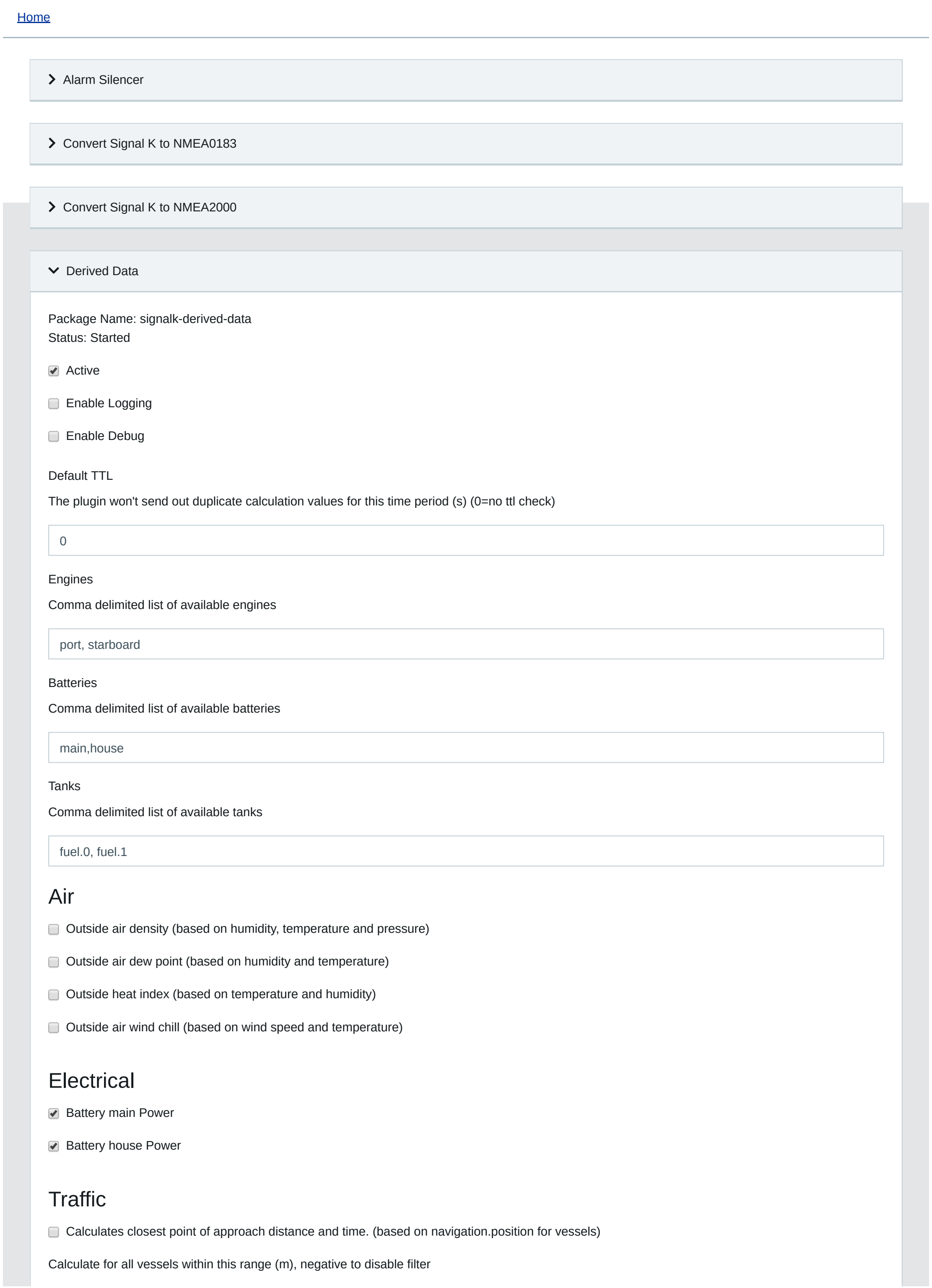
0

Resend Duration (seconds)

The value will be resent for the given #number of seconds

30

## Server/Plugin Config/Derived-data



Home

› Alarm Silencer

› Convert Signal K to NMEA0183

› Convert Signal K to NMEA2000

⌄ Derived Data

Package Name: signalk-derived-data
Status: Started

☑ Active

□ Enable Logging

□ Enable Debug

Default TTL

The plugin won't send out duplicate calculation values for this time period (s) (0=no ttl check)

0

Engines

Comma delimited list of available engines

port, starboard

Batteries

Comma delimited list of available batteries

main,house

Tanks

Comma delimited list of available tanks

fuel.0, fuel.1

## Air

□ Outside air density (based on humidity, temperature and pressure)

□ Outside air dew point (based on humidity and temperature)

□ Outside heat index (based on temperature and humidity)

□ Outside air wind chill (based on wind speed and temperature)

## Electrical

☑ Battery main Power

☑ Battery house Power

## Traffic

□ Calculates closest point of approach distance and time. (based on navigation.position for vessels)

Calculate for all vessels within this range (m), negative to disable filter

1852

Discard other vessel data if older than this (in seconds), negative to disable filter

30

☑ Global send dangerous targets notifications. You must also enable "Calculates closest point of approach distance and time..."

## Dangerous targets notification zone (CPA limit / TCPA limit => Notification level)

+

## Depth

□ Depth Below Keel (based on depth.belowSurface and design.draft.maximum)

□ Depth Below Keel (based on depth.belowTransducer and depth.transducerToKeel)

□ Depth Below Surface (based on depth.belowKeel and design.draft.maximum)

## Propulsion

□ port fuel economy (based on speed over ground, fuel rate)

□ starboard fuel economy (based on speed over ground, fuel rate)

□ propulsion.port.slip (based on RPM, propulsion.port.transmission.gearRatio and propulsion.port.drive.propeller.pitch)

□ propulsion.starboard.slip (based on RPM, propulsion.starboard.transmission.gearRatio and propulsion.starboard.drive.propeller.pitch)

## Wind

□ Ground Wind Angle and Speed (based on SOG, AWA and AWS)

□ True Wind Angle, Direction and Speed (based on speed through water, AWA and AWS)

□ True Wind Direction (based on AWA and headingTrue)

□ Magnetic Wind Direction (based on AWA and headingMagnetic)

□ Magnetic Wind Direction (based on wind.directionTrue and magneticVarition)

□ Wind Shift (experimental)

## Heading

□ True Heading (based on magnetic heading and magneticVarition)

□ Leeway (based on heel angle and STW)

Leeway correlation constant, typically from 9 to 16 (9 for super racer)

12

## Moon

☑ Sets environment.moon.* information such as phase, rise, and set (based on navigation.datetime or system time and navigation.position)

## Sun

☑ Sets environment.sun to dawn, sunrise, day, sunset, dusk or night. Sets environment.mode to day or night. (based on navigation.datetime or system time and navigation.position)

☑ Sets environment.sunlight.times.* to sunrise, sunset, etc (based on navigation.datetime or system time and navigation.position)

## Tanks

□ 'tanks.fuel.0' Tank Volume (based on currentLevel (requires calibration pairs (>2 for parallell sides, >3 for straight wedge and >4 for more complex shapes))

## Server/Plugin Config/UDP NMEA0183 Sender

# Server/Plugin Config/InfluxDB

Home

› Alarm Silencer

› Convert Signal K to NMEA0183

› Convert Signal K to NMEA2000

› Derived Data

› Edit Zones

› Forward and filter NMEA0183 input to NMEA0183 out

⌄ InfluxDb writer

Package Name: signalk-to-influxdb
Status: Started

☑ Active

☐ Enable Logging

☐ Enable Debug

Host

localhost

Port

8086

Database

dropbeerdb

Batch writes interval (in seconds, 0 means don't batch)

10

Resolution (ms)

200

When enabled the vessels position will be stored

☐ Record Track

By default the timestamp in the incoming data is used. Check this if you want log playback to simulate getting new data

☐ Override time with current timestamp

When enabled data from other vessels, atons and sar aircraft will be stored

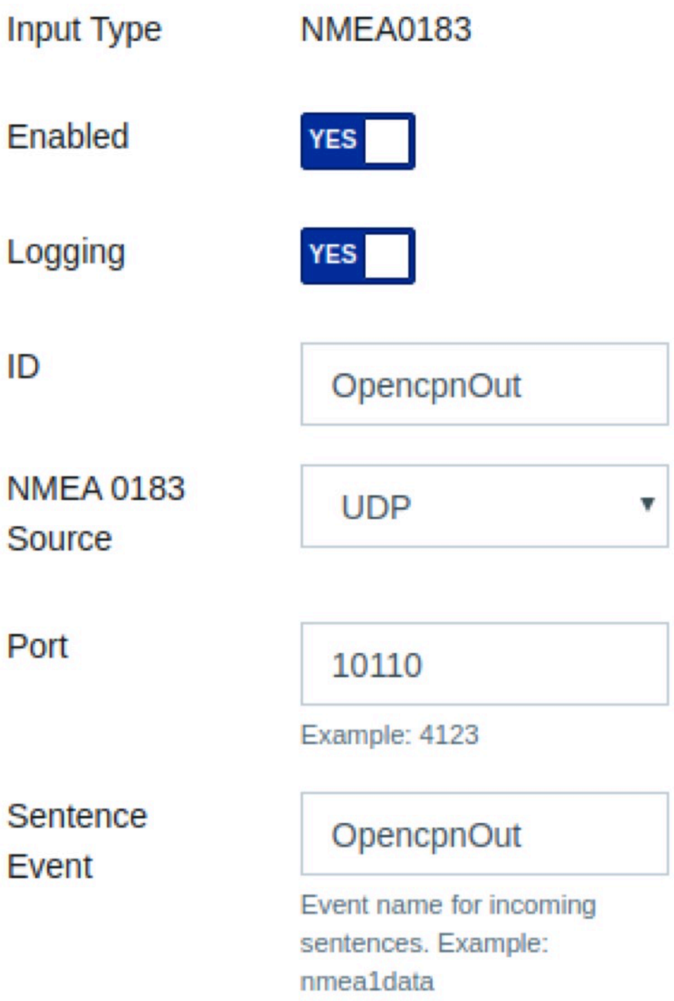
☐ Record Others

Type of List

With a blacklist, all numeric values except the ones in the list below will be stored in InfluxDB. With a whitelist, only the values in the list below will be stored.

# Connections

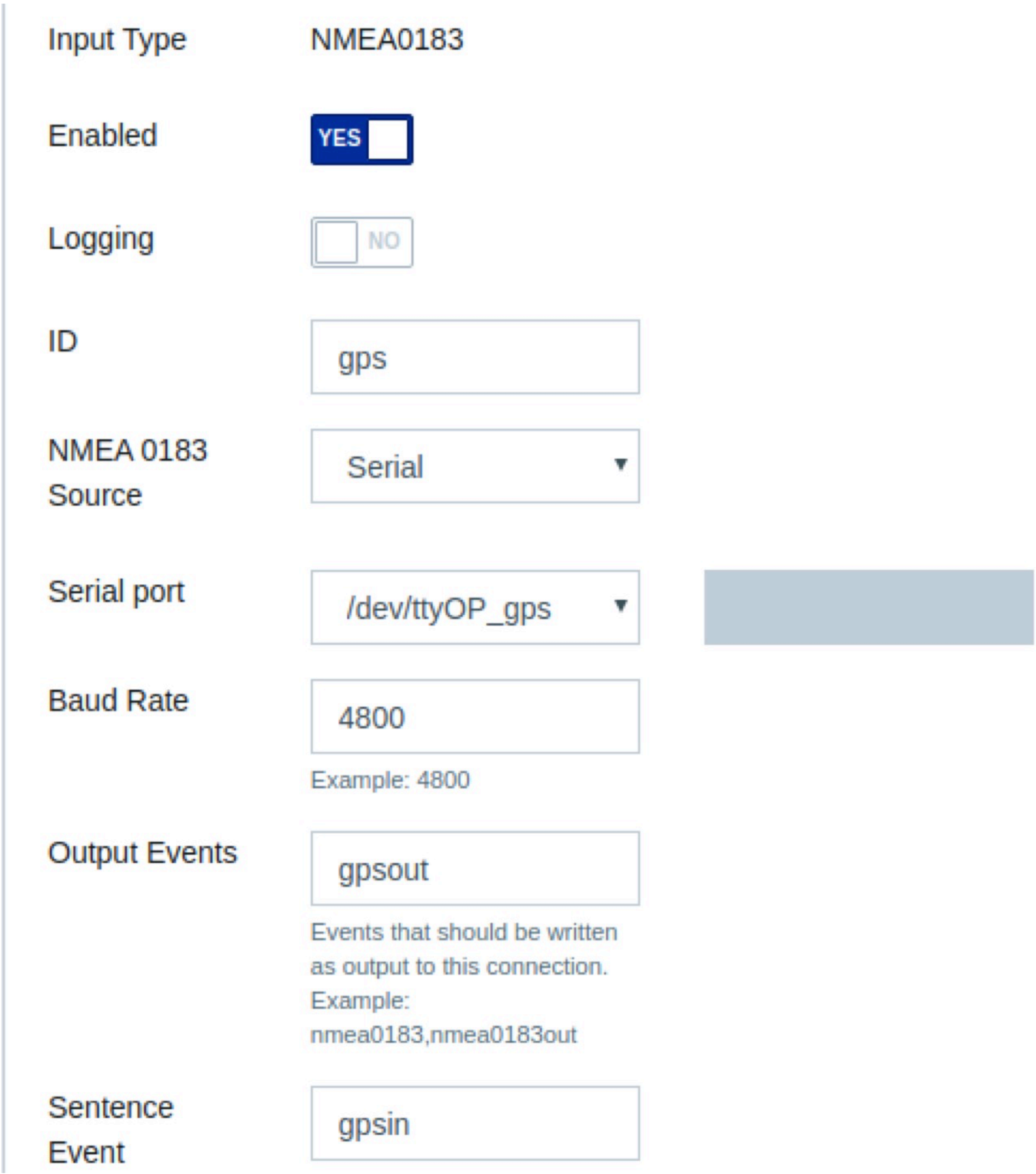
## OpenCPN



| | |
|---|---|
| Input Type | NMEA0183 |
| Enabled | YES |
| Logging | YES |
| ID | OpencpnOut |
| NMEA 0183 Source | UDP ▾ |
| Port | 10110<br>Example: 4123 |
| Sentence Event | OpencpnOut<br>Event name for incoming sentences. Example: nmea1data |

## GPS

Adapt to 38kbaud



| | |
|---|---|
| Input Type | NMEA0183 |
| Enabled | YES |
| Logging | NO |
| ID | gps |
| NMEA 0183 Source | Serial |
| Serial port | /dev/ttyOP_gps |
| Baud Rate | 4800<br>Example: 4800 |
| Output Events | gpsout<br>Events that should be written as output to this connection. Example: nmea0183,nmea0183out |
| Sentence Event | gpsin |

## Running Raspbian from USB Devices : Made Easy

This allows to boot from SD, and run from USB. Thus improving reliability of the system.

https://www.raspberrypi.org/forums/viewtopic.php?t=196778