
EXOCET USER MANUAL AND INSTALLATION GUIDE

Pixel Sur Mer



Revision 1.0

Contents

1	Safety	7
2	Introduction	8
3	EXOCET use cases	10
3.1	Common use cases	10
3.2	Exocet Blue	12
3.3	Exocet Silver	12
3.4	Exocet Gold	13
3.5	Exocet Red	13
4	Hardware specifications	14
4.1	Electrical	14
4.2	Mechanical & Environment	14
4.3	Inputs/Outputs	14
4.4	Connectors	15
4.5	Front LEDS	16
5	Quick start	17
5.1	Power on the Exocet	17
5.2	Check product integrity	17
5.3	Configuring your PC network for the initial connection	17
5.4	Change Exocet IP address	20
5.5	Synchronise time	21
5.6	User interface overview	23
6	Manta	24
6.1	Manta overview	24
6.2	Manta's <i>Hello world</i> graph	25
6.3	Manta boxes color code	26
6.4	Manta Flow	26
6.5	Manta box	28
6.6	Manta configuration box	32
6.7	The deployment	33
6.8	The messages	34
6.9	The message manipulations	35
6.10	Manta processing	40

6.11	Dataloging	40
6.12	Python scripting	41
6.13	Manta libraries	44
6.14	Manta good practices	45
7	Exocet web application	51
7.1	Exocet web app overview	51
7.2	Exocet Web app “Status” page	52
7.3	Exocet Web app “Records” page	54
7.4	Exocet Web app “Data Table” page	55
7.5	Exocet Web app “Alarms” page	56
7.6	Exocet Web app Dashboards	58
8	Communication with third party software	62
9	Tutorial : from sensor to cloud	63
9.1	Reading the analog data	63
9.2	Visualising the data in a probe	65
9.3	Renaming the input data	65
9.4	Calibrating input	67
9.5	Visualizing data in the data table	68
9.6	Visualising data in a dashboard	69
9.7	Setting an alert	72
9.8	Visualising the alert	73
9.9	Datalogging	74
9.10	Sending data to the cloud	75
9.11	Visualizing cloud data with Graphana	78
10	Manta boxes overview table	80
10.1	Input interfaces	80
10.2	Input devices	80
10.3	Input softwares	81
10.4	Input	81
10.5	Input User	82
10.6	Protocol Decode	82
10.7	Processing	84
10.8	Output Interface	85
10.9	Protocole Encode	86
10.10	Output Device	86

10.11 Output Software	87
10.12 Output	87
10.13 Debug	88
10.14 wind	88
11 Manta boxes detailed description	89
11.1 add_delay	89
11.2 adrena_xdr	90
11.3 ahrs	90
11.4 einstein	92
11.5 ais	93
11.6 alarm	93
11.7 analog	94
11.8 antiCapsize	95
11.9 ascii_out	98
11.10 blinkManager	99
11.11 bravo_in	100
11.12 bravo_out	101
11.13 bretronics	102
11.14 button	104
11.15 calibration	105
11.16 canbus_in	108
11.17 canbus_out	108
11.18 cycle	109
11.19 damping	110
11.20 datalogger	113
11.21 debounce	114
11.22 display_bandg	115
11.23 display_garmin	116
11.24 display_nke	121
11.25 e_telldes	122
11.26 empirbus	123
11.27 empirbus_out	125
11.28 enlight	126
11.29 eta	127
11.30 exocet_in	128
11.31 exocet_out	128
11.32 expression	129

11.33 fail_safe	138
11.34 freq_divider	142
11.35 gpio_in	143
11.36 gpio_out	144
11.37 h5000	145
11.38 height_estimation	146
11.39 hyperion	151
11.40 idatanet	152
11.41 ixblue	152
11.42 json_dec	158
11.43 json_enc	158
11.44 keel	159
11.45 kvh	161
11.46 lee_cur_tools	163
11.47 manta_status	169
11.48 modbus_in	170
11.49 modbus_out	171
11.50 moving_mean	173
11.51 net2000_in	175
11.52 net2000_out	175
11.53 nmea0183_dec	176
11.54 nortek	176
11.55 obstacleManager	177
11.56 on_board_assistant	179
11.57 opc_in	180
11.58 oscar	181
11.59 overlay_manager	182
11.60 pid	184
11.61 pilot_overlay_bandg	191
11.62 pilot_overlay_nke	194
11.63 pilot_settings_bandg	197
11.64 polar	199
11.65 probe	202
11.66 python_function	202
11.67 regulation_performance	212
11.68 ride_height	212
11.69 sailnet	214
11.70 sbg	215

11.71 scalar	216
11.72 senix	218
11.73 sentea	218
11.74 serial	220
11.75 serial_in	220
11.76 sirius	221
11.77 stats	222
11.78 string	223
11.79 system	224
11.80 table	226
11.81 tcp_in	228
11.82 tcp_out	228
11.83 tick	229
11.84 udp_in	229
11.85 udp_out	230
11.86 varfilter	231
11.87 varmux	231
11.88 vector_man	233
11.89 victron	234
11.90 wattandsea	236
11.91 williamson	238
11.92 wind_apparent	240
11.93 wind_ground	242
11.94 wind_mastrot	244
11.95 wind_motion	246
11.96 wind_processor_hr	249
11.97 wind_true	250
11.98 wind_upwash	252
11.99 wtp	257
11.100wtpSerialManager	258
11.101zeromq_in	259
11.102zeromq_out	260
12 Appendix	260
12.1 Recover the IP address of the Exocet	260
12.2 Configuring your PC to be a Time server	261

1 Safety

The device complies with Part 15 of the FCC Rules. Operation is subject to the following two conditions: (1) This device may not cause harmful interference, and (2) This device must accept any interference received, including interference that cause undesired operation.

The Company “Pixel Sur Mer” is not responsible for any changes or modifications not expressly approved by the party responsible for compliance. Such modifications could void the user’s authority to operate the equipment.

NOTE: This equipment has been tested and found to comply with the limits for a Class A digital device, pursuant to Part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference when the equipment is operated in a commercial environment. This equipment generates, uses, and can radiate radio frequency energy and, if not installed and used in accordance with the instruction manual, may cause harmful interference to radio communications. Operation of this equipment in a residential area is likely to cause harmful interference in witch case the user will be required to correct the interference at is own expense.

2 Introduction

Today's skippers and their design engineers are facing the great technological challenge of flying boats and high performance ships. In order to complement the existing panel of equipments and to fully respond to the new demands of its customers, PIXEL SUR MER has innovated and developed some of the most advanced systems on the market : **the EXOCET line**.

The Exocet line is made of 4 different products which have their own functions:

- **Exocet Blue:** Universal data acquisition and monitoring
- **Exocet Silver:** Central processor to improve safety and performances of latest generation of racing boats:
 - Advanced sensors management with failsafe support
 - Advanced true wind calculation
 - Autopilot safety and performance enhancement with pilot overlays rules
- **Exocet Gold:** Flight control process unit
- **Exocet Red:** High accuracy power servo controlled by the **Exocet Gold**



Figure 1: Exocet line

The Exocet devices are compatible and interface with greatest performing navigation systems on any boat. But, they are no restricted to be used in a boat system environment. In particular, the Exocet

blue is a universal datalogger fitting **industrial** and **aerospace** requirements and needs.



Exocet Red is a specific product not described in this document. Please refer to the Exocet Red User Manual for more information.

3 EXOCET use cases

The Exocet devices can be used for one or several use cases at the same time. Some are applicable to all devices, other are specific to the Blue, Silver or Gold.

3.1 Common use cases

Data Gateway

The Exocet devices can read data from miscellaneous hardware interfaces and protocols and then forward these data to other devices or software using another protocol and hardware interface.

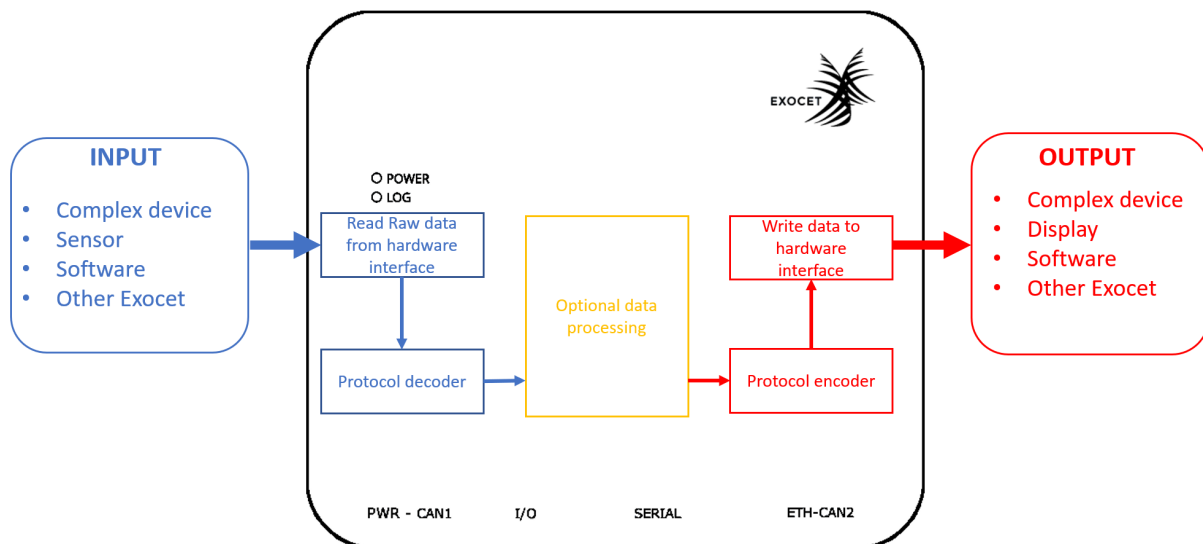


Figure 2: Data Gateway concept

- **Input hardware interfaces:** Analogue, GPIO, Serial, Ethernet/UDP/TCP, CAN bus
- **Standard input protocol decoder:** NMEA0183, NMEA2K, J1939, Modbus
- **Proprietary input protocol decoder:**
 - **Navigation Systems:** B&G H5000 CPU and WTP3, NKE ProcessorHR, Bravo 4 CPU
 - **Auto pilot status:** B&G H5000 pilot, NKE Gyropilot2
 - **Power monitoring:** Watt&Sea aero and hydro generator, Victron solar panel controller, Williamson and Bren-Tronics battery, ETA Powerplex, Garmin EmpirBus
 - **Optical fiber:** Luna and Pixel Sur Mer optical fiber interrogator, Luna Enlight software
 - **Attitude sensor:** IxBlue, SBG, KVH
 - **Exocet:** receive data from another Exocet device

- **Miscellaneous:** Nortek DVL, E-telltales, Sirius (Pixel sur Mer GPIO sensors bus), Oscar, Blink keypad...
- **Data processing:** damping, calibration, average, variance, min/max, median, delay, undersampling...
- **Output hardware interfaces:** GPIO, Serial, Ethernet/UDP/TCP, CAN bus
- **Standard output protocol encoder:** NMEA0183, NMEA2K, J1939, Modbus
- **Proprietary output protocol encoder:**
 - **Navigation System:** NKE ProcessorHR, Bravo 4 CPU
 - **Auto pilot control:** B&G H5000 pilot, NKE Gyropilot2
 - **Display:** B&G, NKE, Garmin...
 - **Exocet:** send data to another Exocet device
 - **Miscellaneous:** Adrena, Expedition, Idatanet, Garmin EmpirBus...



With **Python scripting support**, Exocet device can easily be extended with any custom protocol decoder/encoder or any custom data processing function.

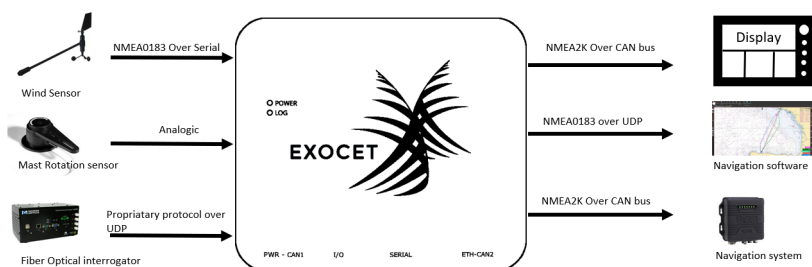


Figure 3: Data gateway exemple

System Monitoring

Custom dashboards and graphs can be created to monitor data on the fly. An editor is available to compose the live views best suited for each requirement. Alarms thresholds can be defined to be notified when a data is out of it's expected value range. The user interface is web based, so it is accessible from a PC, a tablet or a smartphone. If the connection is established from a smartphone, a web application adapted to small screen size is launched.

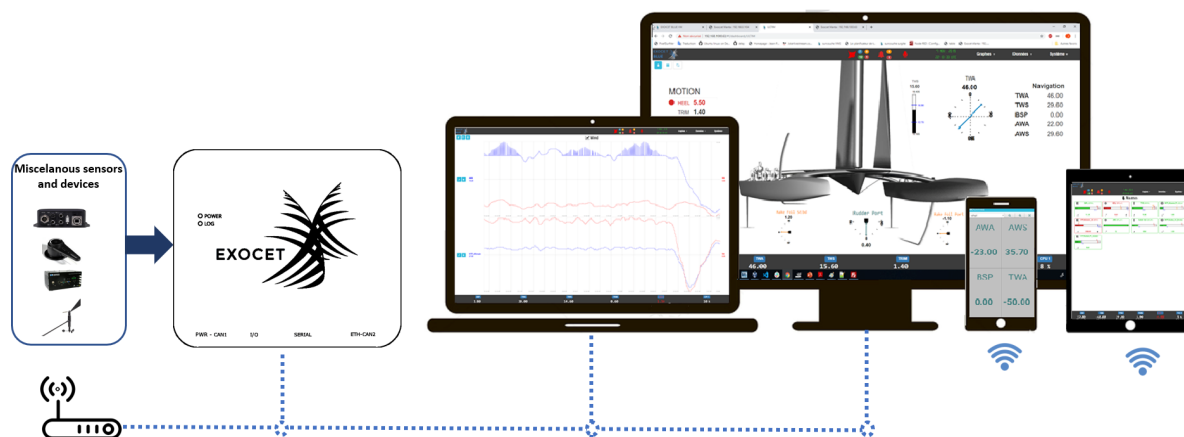


Figure 4: Live data monitoring

3.2 Exocet Blue

PIXEL's first innovation on the SEA, **Exocet Blue** is the state-of-the-art embedded data acquisition, storage and supervision system. With it's 250Gb storage capacity, it can record up to 100 variables, with an acquisition frequency of 100Hz during at least 6 months.

Black Box

Exocet Blue can be used as a **black box** for the purpose of facilitating the investigation of incidents. As it starts recording automatically when it is truned on, so it does not not need any user interaction. The data storage SD card is **IP68** rated to protected from immersion in water.

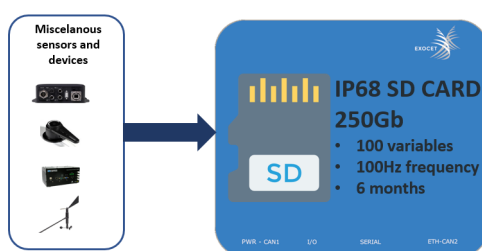


Figure 5: Black box

3.3 Exocet Silver

Central processor to improve safety and performances of lastest generation of racing boats.

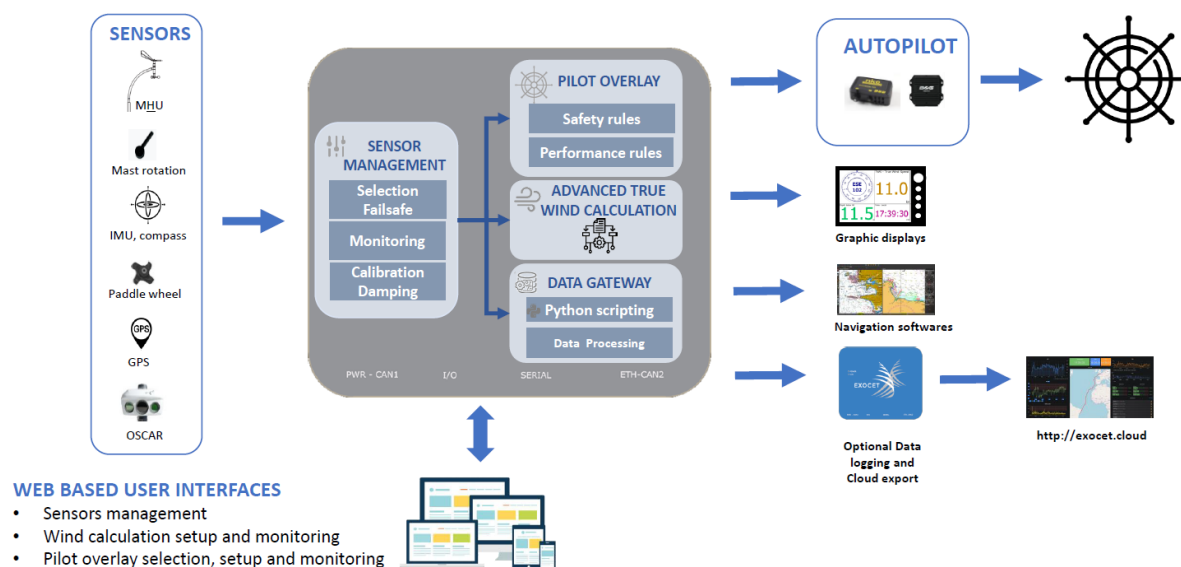


Figure 6: Exocet Silver overview

- Advanced sensors management with failsafe support
- Advanced true wind calculation
- Autopilot safety and performance enhancement with pilot overlay rules.
- Compliant with most marine protocol, devices and displays.
- User friendly graphical interfaces

3.4 Exocet Gold

EXOCET GOLD means:

- Central processor to:
 - Read and decode high-frequency sensors
 - Handle flight control algorithms
 - Controls one or several appendages via the EXOCET RED

3.5 Exocet Red

Fast and accurate servo command for local control of appendages via hydraulic pump motor or electric rams. Integrated safety mode.

4 Hardware specifications

4.1 Electrical

- **Input voltage** : 9-51V
- **Power consumption**: 4W

Exocet devices uses capacitors to prevent voltage glitches.

4.2 Mechanical & Environment

- **Weight**: 350 g
- **Material**: Anodized aluminum
- **Size**: 113 x 97 x 40mm
- **Specified temperature**: -10/+55°
- **IP Rating**: IP67
- **Persistent storage**: IP68 SD card (Exocet Blue only)

4.3 Inputs/Outputs

Interface	Number	Details
Ethernet	1	Gbe (Giga byte ethernet)
CAN 2.0	2	Up to 1 mbps
Analogic inputs	6	Isolated - 16 bits (0/5V, +/-5V, 0/10V, +/-10V). Sampling rate up to 50Hz
Digital inputs	2	Isolated
Digital outputs	3	Isolated, open collector. 1A max.
Serial	6	5 RS232, 1 RS232/RS422. Up to 115200 Bds
Power	2	Access to the input voltage to facilitate wiring

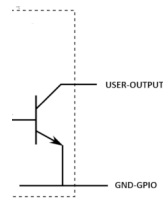


Figure 7: GPIO open collector

4.4 Connectors



Figure 8: Hardware interface

Exocet Blue, Silver and Gold have 4 connectors:

- Standard NMEA2000 male connector for power supply and CAN1
- I/O yellow connector for digital and analog inputs/outputs (see details below)
- Serial red connector for serial inputs/outputs (see details below)
- A small red connector linked to standard Ethernet and NMEA2000 connectors (provided)

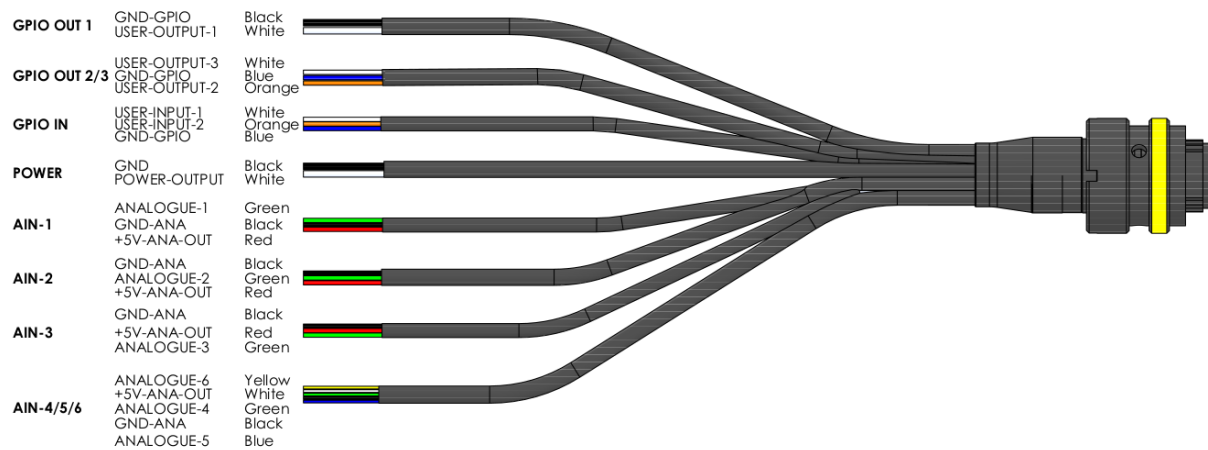


Figure 9: I/O cable

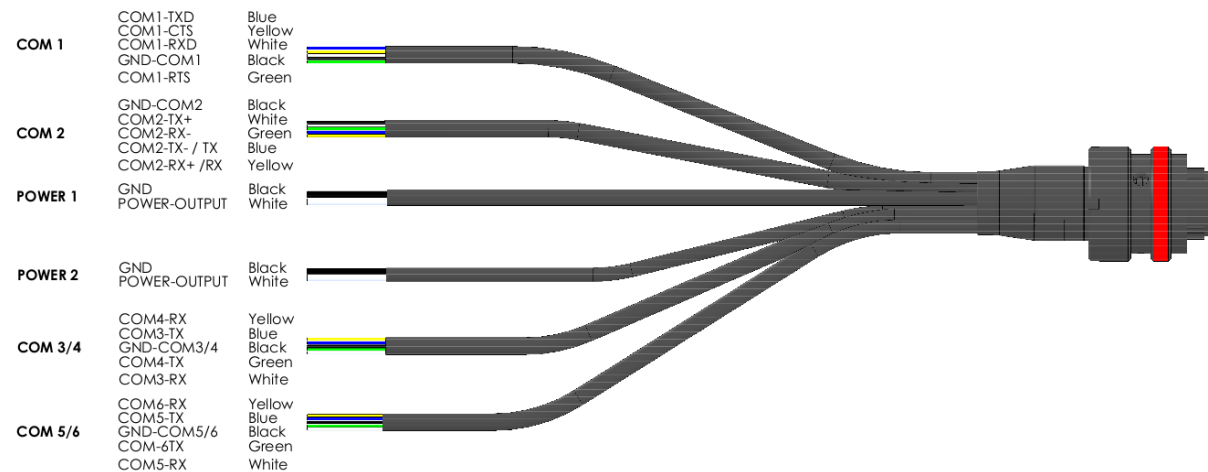


Figure 10: Serial cable

4.5 Front LEDS

Each Exocet has two leds, one green and one red.

Green led on indicates that the Exocet is running.

Red led on indicates that the datalogger is recording. Red led flashing indicates that a Manta box is in error state.

5 Quick start

5.1 Power on the Exocet

The first step is to power on the Exocet. As mentioned on the Hardware section, the power delivery is between 9 and 51 volts through the CAN-1 connector. It is a standard CAN M12 male connector. Here is a schematic of the pinout :

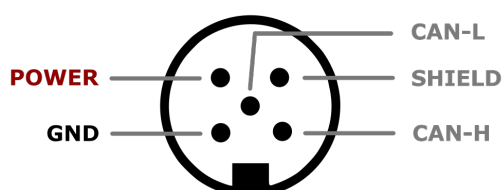


Figure 11: CAN-1 pinout

The power cable supplied with the Exocet can be used to connect the Exocet to a powered Can bus.

5.2 Check product integrity

On hardware side, verify that the sticky seal is present on one screw at the back of the Exocet. On the software side, check that the green light labelled POWER is on.

5.3 Configuring your PC network for the initial connection

Exocet are provided with IP address **192.168.100.251**.

To avoid any network trouble, it is recommended to connect directly the Exocet to a PC. To be able to communicate with Exocet the IP address of the PC must be set to 192.168.100.XXX with XXX a number from 1 to 250.

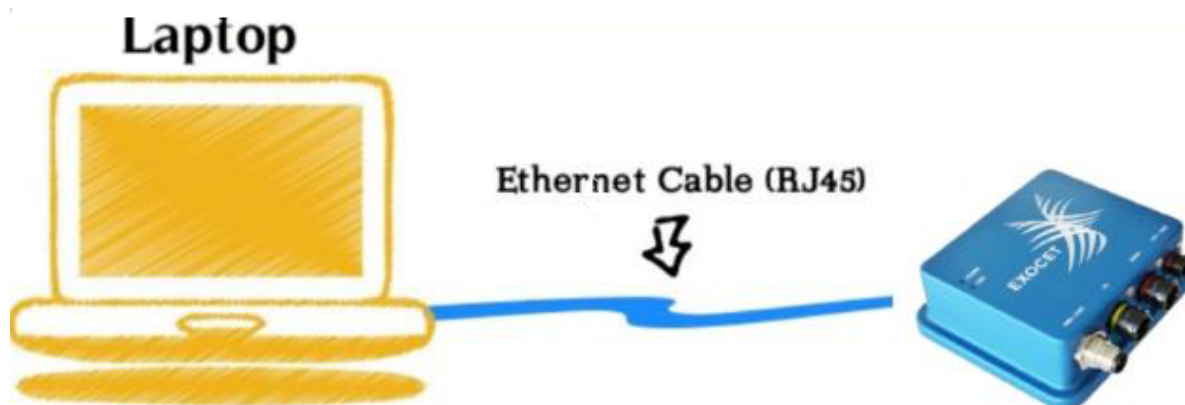


Figure 12: laptop direct connection

Set your PC with the static IP address 192.168.100.250

To set a static IP address in Windows 7, 8, and 10:

1. Click **Start Menu > Control Panel > Network and Sharing Center or Network and Internet > Network and Sharing Center.**
2. Click **Change adapter settings.**
3. Right-click on **Local Area Connection.**
4. Click **Properties.**
5. Select **Internet Protocol Version 4 (TCP/IPv4).**
6. Click **Properties.**
7. Select **Use the following IP address.**
8. Enter the **IP address 192.168.100.250, Subnet mask 255.255.255.0**
9. Click **OK.**

Your computer is now using 192.168.100.250 static IP address.

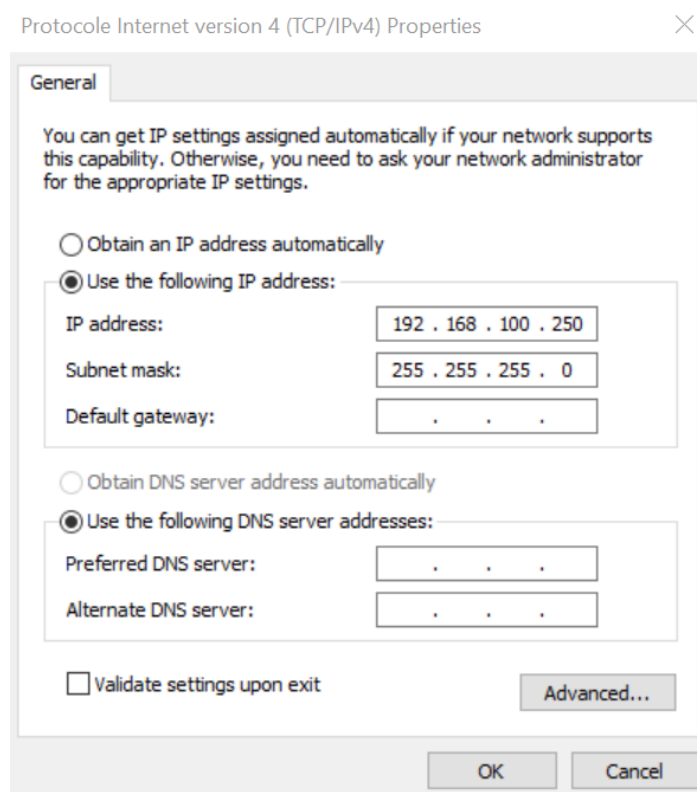
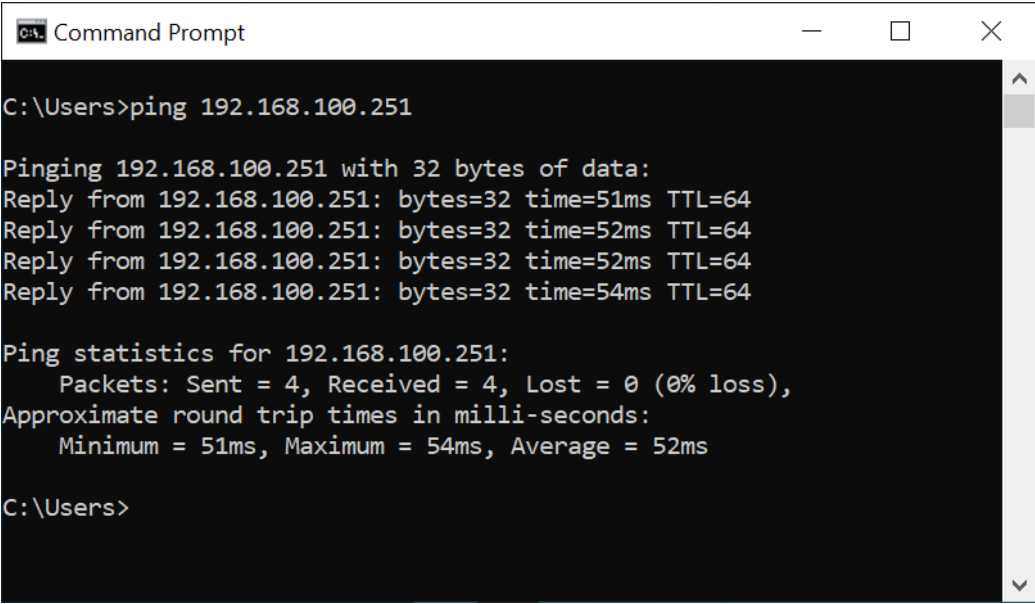


Figure 13: IPv4 setting

Check communication with EXOCET

1. Click **Start Menu > Command Prompt**
2. From the command prompt windows, enter the **ping 192.168.100.251** command and press **ENTER** key.



```
Command Prompt

C:\Users>ping 192.168.100.251

Pinging 192.168.100.251 with 32 bytes of data:
Reply from 192.168.100.251: bytes=32 time=51ms TTL=64
Reply from 192.168.100.251: bytes=32 time=52ms TTL=64
Reply from 192.168.100.251: bytes=32 time=52ms TTL=64
Reply from 192.168.100.251: bytes=32 time=54ms TTL=64

Ping statistics for 192.168.100.251:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 51ms, Maximum = 54ms, Average = 52ms

C:\Users>
```

Figure 14: ping command

If reply are received the PC and the Exocet are connected, else, check physical connections.

5.4 Change Exocet IP address

On your browser, search 192.168.100.251 and you should arrive on the Exocet web interface, on the page named "Status".

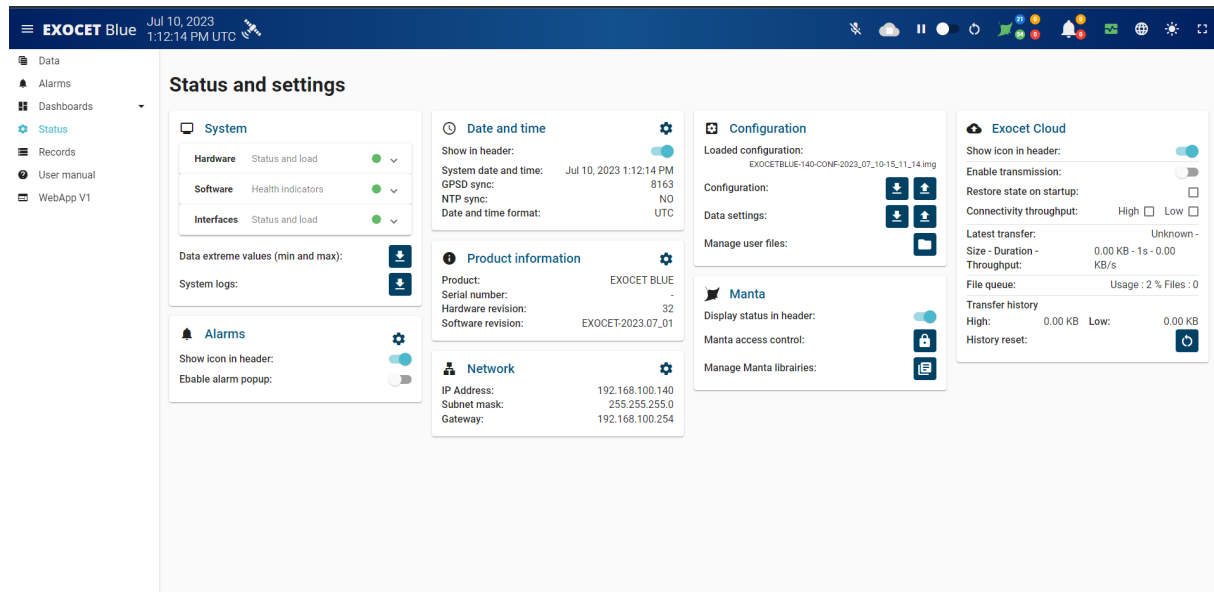


Figure 15: Exocet web app setting page

On the network section, click on the cogwheel to access IP settings. Change the IP to the desired one and click on “Ok” to validate. the Exocet will restart with the new network parameters.

5.5 Synchronise time

The Exocet has its own time management. The date and time displayed on the top of the web application may differ to the actual date and time. There are several possibilities to change the time of an Exocet.

Apply computer's time

On the same web page, go to the date and time settings by clicking on the cogwheel of the *Date and Time* card. As shown on the picture below, go to the *Synchronization* menu and click on the graphical button to apply PC time to Exocet (highlighted on green) and click on the *Apply* button.

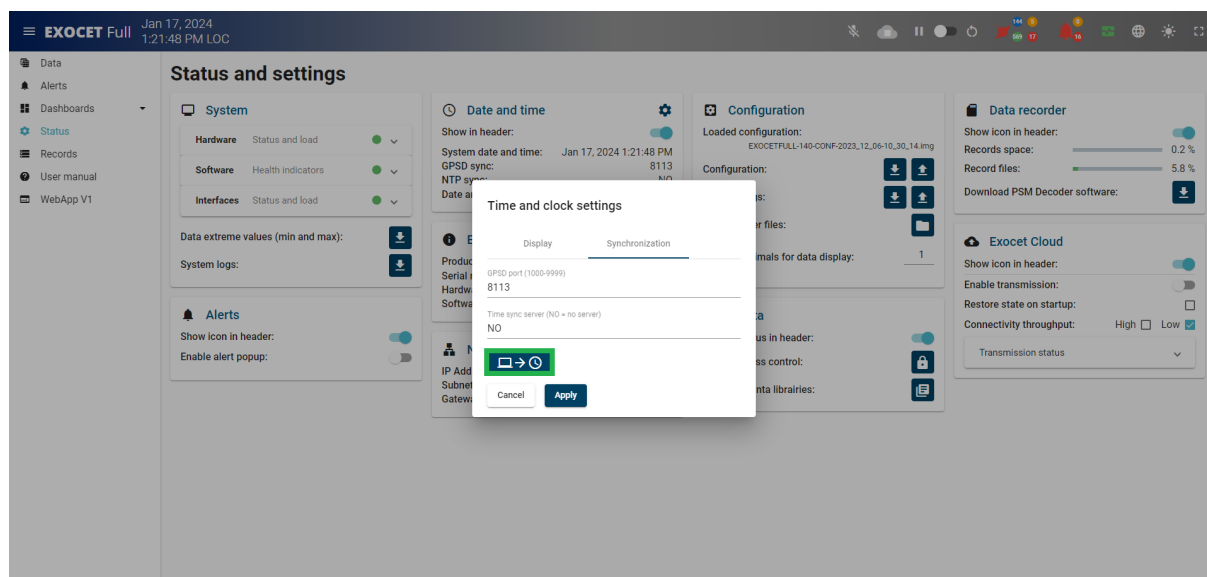


Figure 16: Synchronise time with compute

The time displayed on the top left of the web application has now the same time as your computer.

GPS synchronization

The time inside the Exocet may drift over time. It is recommended to use a dynamic time method to keep it synchronised. On the same menu as for the computer time, you can chose a *GPSD port* to send an NMEA0183 \$RMC sentence and *Apply* the changes. Therefore the Exocet time will be constantly following the high precision GPS time. The following icon will appear next to the date and time on the web application :



Figure 17: GPS synchronization

NTP server

An other way to keep up to date is to date is to synchronize to an NTP server. On the same menu as for the computer time, you can fill an ip address of a NTP server in the *Time sync server* field and *Apply* the changes. The Exocet time will be constantly following the server time. The following icon will appear next to the date and time on the web application :



Figure 18: NTP server

Note : Each Exocet has an internal NTP server. Once properly synchronised, an Exocet can be used as a time reference for other devices.

5.6 User interface overview

Exocet user interface is accessible using a web browser with Exocet IP address as URL. Default IP address is 192.168.100.251. This User Interface is made of two parts:

- *MANTA*, which is our user-friendly graphic configuration tool:
 - To define which and how incoming data are collected
 - To process collected data with a wide variety of processing functions available and by python scripting
 - To define which and how data are exported to third party device or software
 - To select data to log on SD card or to send to Exocet Cloud (Exocet Blue only)
- *Exocet Web App*, which is daily User Interface:
 - To monitor live data thanks to customizable dashboards, graphs and displays. An editor enables you to compose the live views best suited for each requirement.
 - To define and monitor alarms with several criticity levels
 - To manage data logs files (Exocet Blue only)
 - To configure and monitor internal data of the Exocet

6 Manta

6.1 Manta overview

Manta is the configuration editor of the Exocet line products. It comes in the form of a graphical data manipulation programming editor.

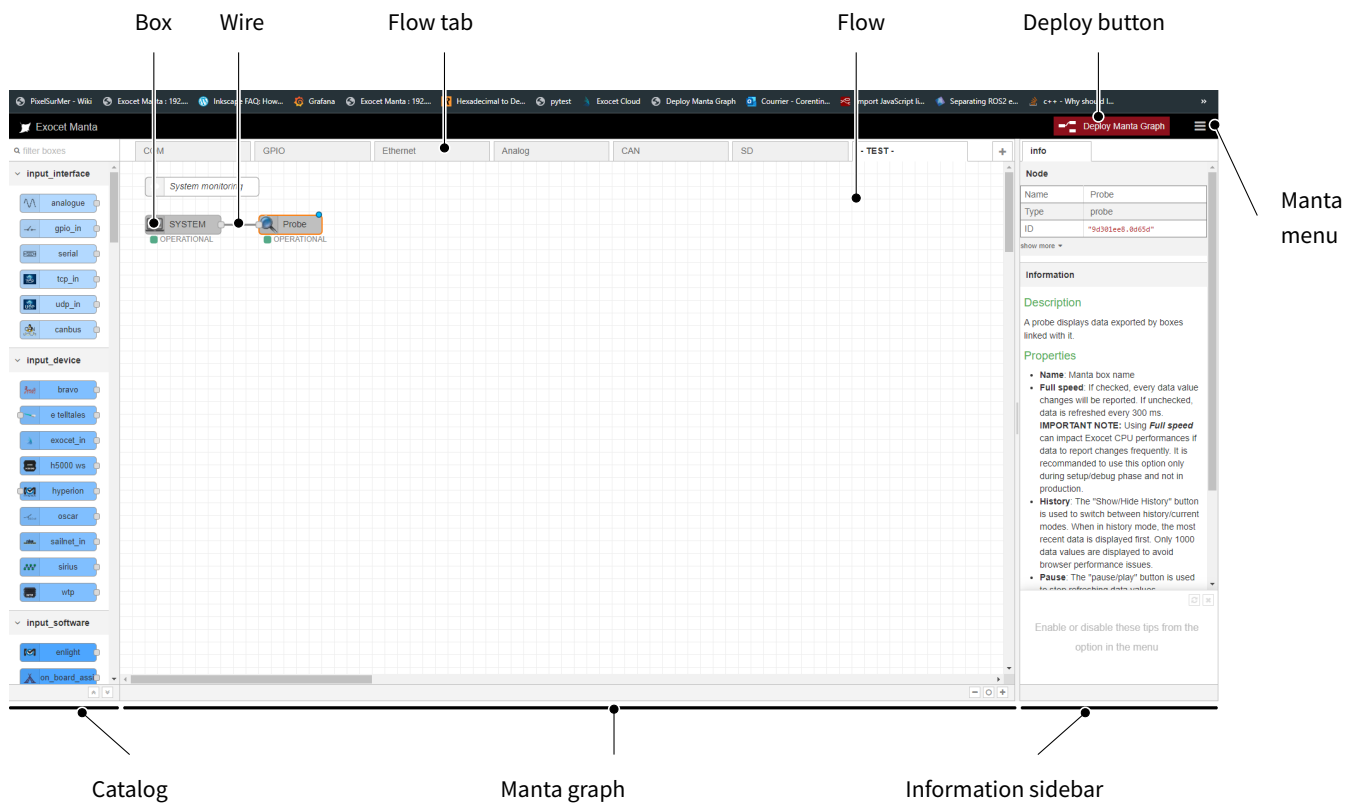


Figure 19: Overview of the Manta user interface.

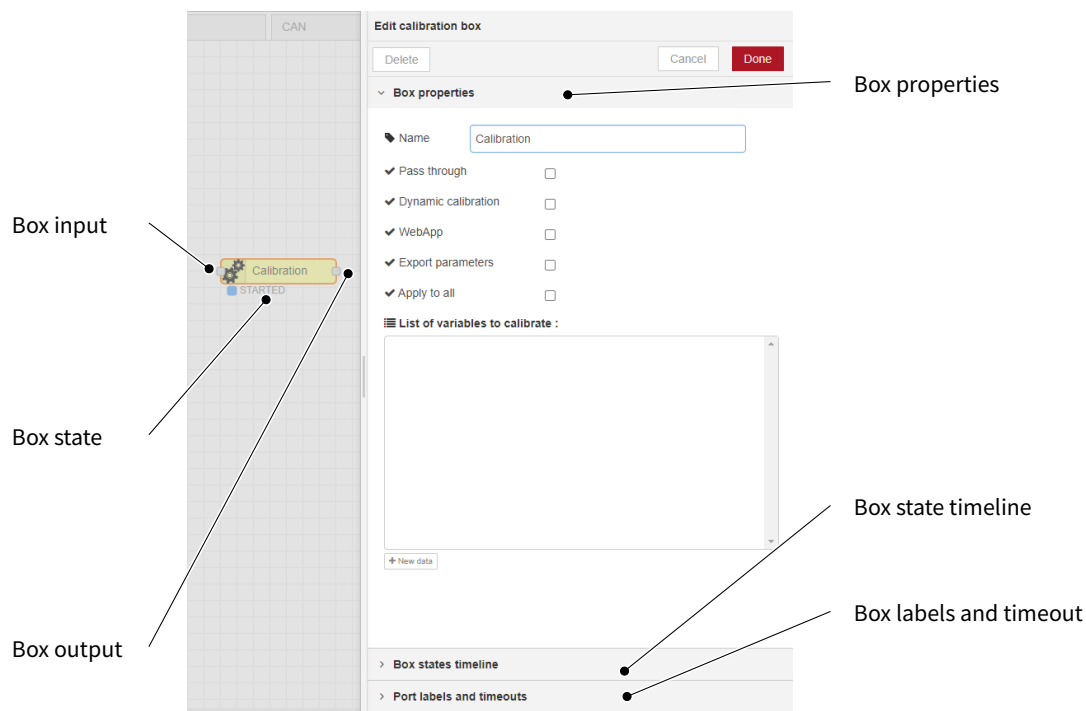


Figure 20: Manta box overview

To design a Manta Graph, and so configure Exocet data processing:


1. Drag and drop *boxes* from the *catalog* to the flow page. The search tool at the top of the catalog can help searching a specific box.
2. Configure each *box* by double clicking to open the *Edition sidebar*. A contextual information is available on the right margin.
3. Send “message” from a box to another by linking the *output port* of a box to the *input port* of another with a *wire*. It is done by pressing the left-mouse button on a box’s *output port*, dragging to the destination box’s *input port* and releasing the mouse button.
4. Press the “Deploy Manta Graph button” at the top right of the workspace.



Note: If a box is not well configured and prevent the deploy, a small orange triangle appears at the top right corner of the box. All boxes modified since the last deploy are notified by a blue spot at the top right corner of the boxes.

6.2 Manta’s Hello world graph

1. Drag and drop a *Tick* box from the *catalog* to the *flow*

2. Drag and drop a *Probe* box from the *catalog* to the *flow*
3. Link both by pressing the left-mouse button on the *Tick* box *output port*, dragging to the *Probe* box *input port* and releasing the mouse button.
4. Press the  button.
5. Double click on *Probe* box, *Tick* box data should appear on the box *properties panel*.

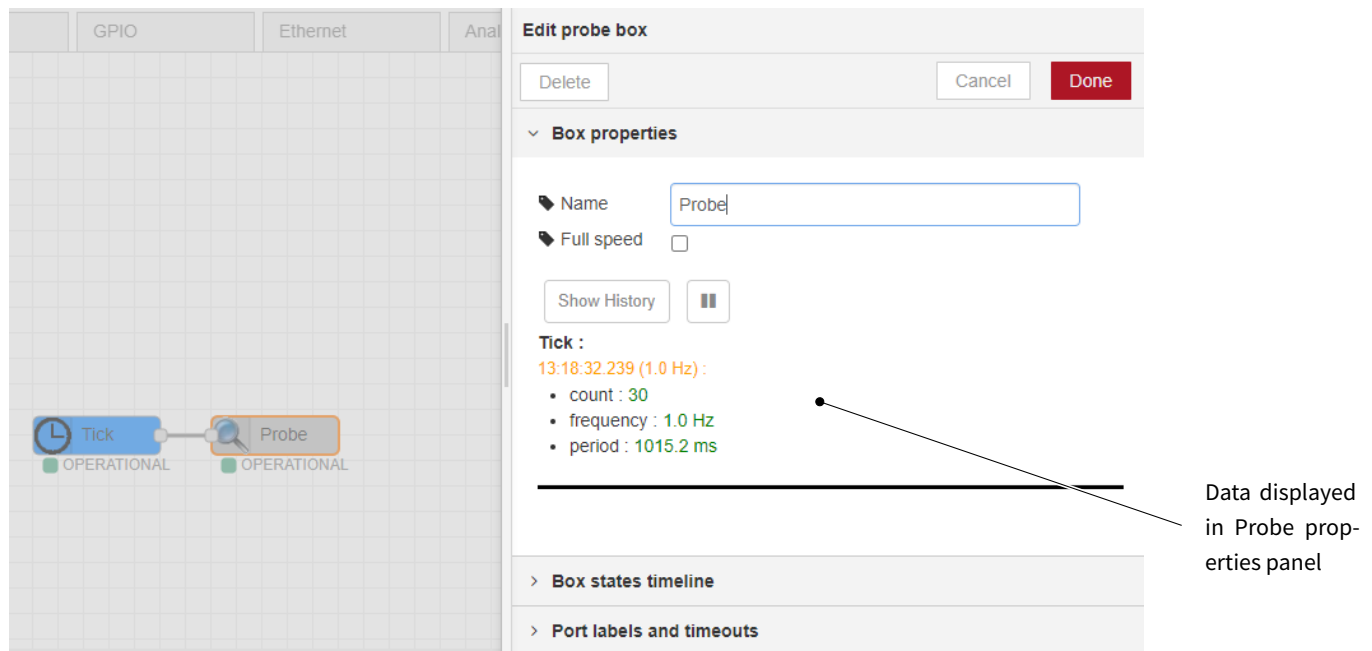


Figure 21: Live data display using a "Probe" box

6.3 Manta boxes color code

In the catalog, the boxes are organized with the following color code:

- **Input :** Acquisition of data from various equipment, software or Exocet inputs.
- **Protocol :** Decoding of various data from Ethernet protocol, CAN bus, serial link.
- **Processing :** Data processing. Calibration, mathematical calculations, filtering, programming, etc.
- **Output :** Data export to various equipment, software or Exocet outputs.
- **Debug :** Behaviour monitoring and visualisation.

6.4 Manta Flow

A flow is a page containing a set of boxes from the catalog and connected to each other by wires.

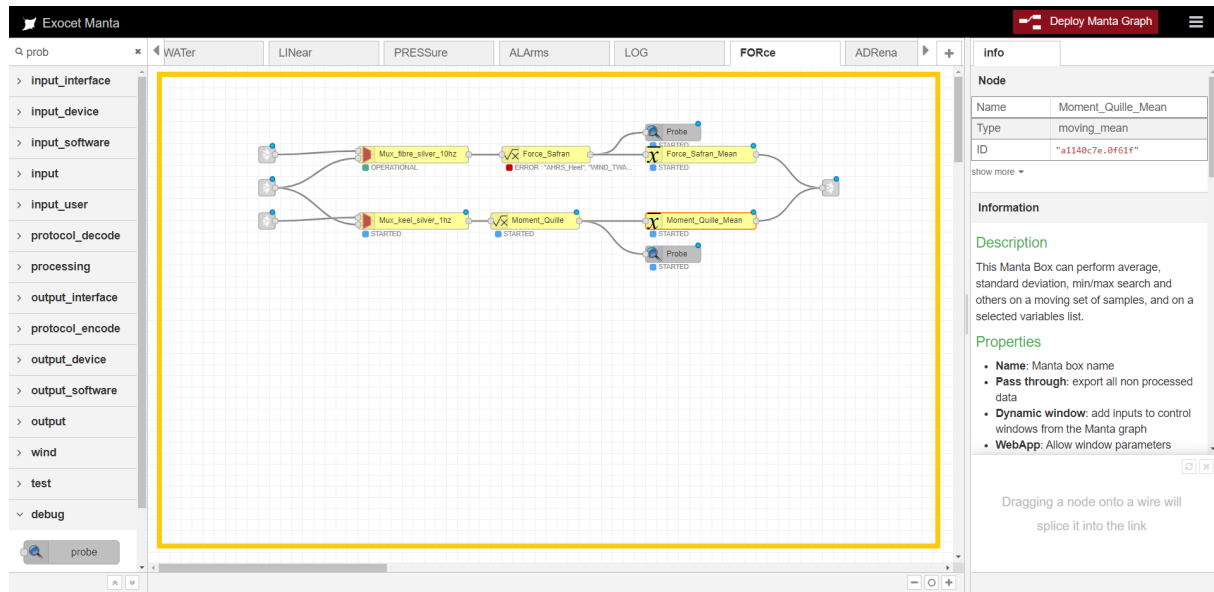


Figure 22: Manta Flow

A Flow is made of:

- A tab to select the flow to display and see its status
- A description visible in the information side panel
- A graph of boxes and wires
- A status: Enabled or Disabled

Parameters:

Accessible by double clicking on its tab

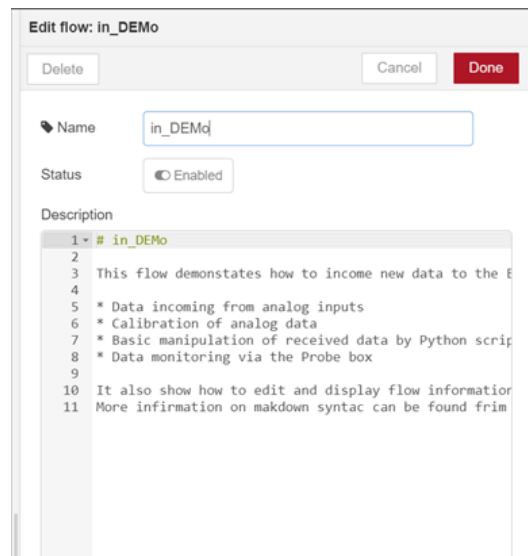




Figure 23: Manta flow parameters

- **Name** : Unique name of the flow
- **Status**: change the state of the flow. Boxes in a flow in the disabled state are not executed. Flows in the disabled state are identified by an icon  in their tab.
- **Description**: edit the flow description in markdown format
- The  button is used to delete the flow

Add a flow:

Click on the  button in the tab bar

Reordering the flows:

By dragging and dropping the tab. The arrangement of the flows has no impact on the execution order of the contained boxes.



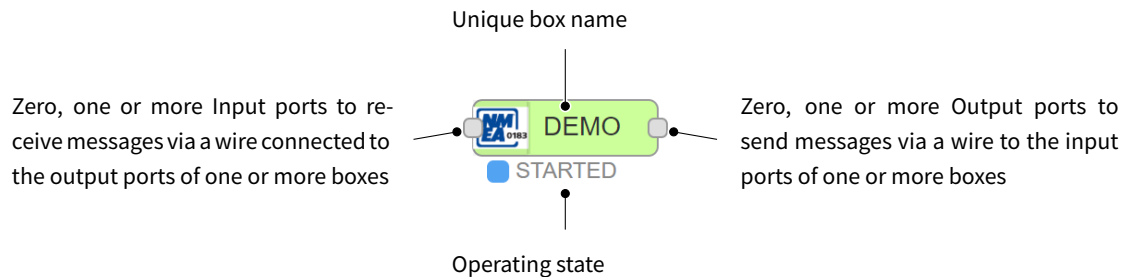
The **link boxes** are used to send messages from one flow to another.



The buttons  are available to change zoom.

6.5 Manta box

A box is an independent functional block covering a function of acquiring, processing or exporting data.

Overview:**Figure 24:** Overview of a Manta box

The *Show box status* option must be selected in the Manta menu to display the box status

A box can represent:

- A physical input/output (serial port, CAN, Ethernet ...)
- A device to communicate with (navigation system, battery, wind turbine...)
- A third-party software to communicate with (events, charts...)
- A protocol to encode/decode (NMEA0183, NMEA2000...)
- A mathematical processing (damping, calibration...)
- An internal feature (link, probe, datalogger...)
- A user specific algorithm written with Python 2.7 (including Numpy library)
- A user interaction associated with an Exocet WebApp dashboard widget (button, scalar...)
- ...

A box is made of :

- A *unique name* to identify it
- A set of *Input ports* to receive message from one or several boxes
- A set of *Output ports* to send message to one or several boxes
- A set of *Properties* which corresponds to box setting parameters
- A *State* to indicate the box internal status

Each port can be associated with:

- A *Label* to name or rename it
- A *Timeout* to switch to *warning* when no message is received or sent

Conditions of execution:

- Upon receipt of data via a physical I/O of the Exocet unit
- Upon receipt of a message by an ingress port of the box
- Periodically on timer

Documentation:

By clicking on a box, a contextual help is visible in the information panel. This help generally contains the headings:

- Description: general description of how the box works
- Properties: description of the parameters
- Inputs: description of data received via the input ports
- Outputs: description of data transmitted via the output ports



The *Show sidebar* option must be selected in the Manta menu to display the information panel

The parameters:

Figure 25: Manta box parameters

The edit panel can be accessed by double clicking on a box. The **Box properties** tab of this panel is used to change the name and settings of the box.

The list of parameters is specific to each box. If necessary, the information panel provides a description of each of the parameters.

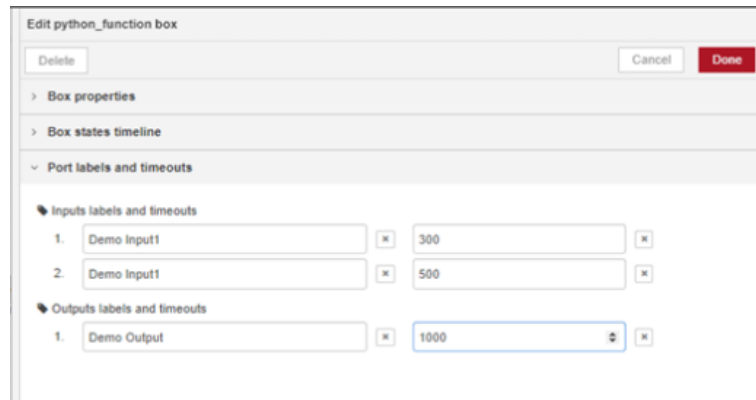


Figure 26: Ports labels and timeouts

The *Port labels and timeout* tab of the edit panel can be used to:

- Change the label associated with each port.
- Associate a timeout in milliseconds.
- For an input port: the box switches to the WARNING state if no message is received at the end of the timeout.
- For an output port: the box switches to the WARNING state if no message is transmitted at the end of the timeout.

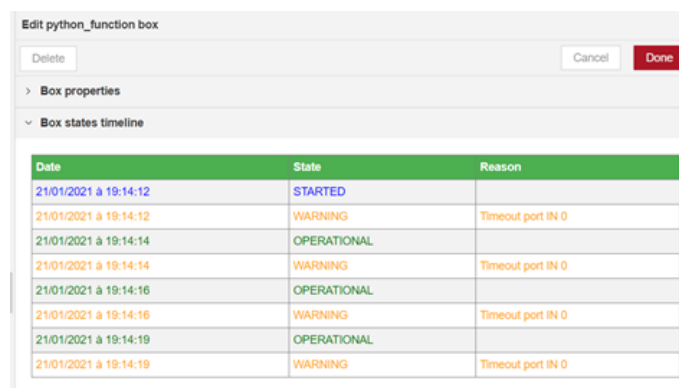
 The icon  is visible if a timeout has been associated with a port

The states:

During its operation, a box can take the following states:

- **STARTED** : The box is correctly initialized, but it has not yet received or sent any messages
- **OPERATIONAL** : The box has successfully received or transmitted at least one message
- **WARNING** : Timeout triggered on a port or minor internal malfunction. This state is always accompanied by a description of the trigger reason
- **ERROR** : Major dysfunction. This state is always accompanied by a description of the trigger reason. The revision mismatch, please replace this box reason indicates that the box should be replaced because its version is no longer compatible with the current firmware version.

The timeline of a box state changes is visible in the Box states timeline tab of the edit panel



Date	State	Reason
21/01/2021 à 19:14:12	STARTED	
21/01/2021 à 19:14:12	WARNING	Timeout port IN 0
21/01/2021 à 19:14:14	OPERATIONAL	
21/01/2021 à 19:14:14	WARNING	Timeout port IN 0
21/01/2021 à 19:14:16	OPERATIONAL	
21/01/2021 à 19:14:16	WARNING	Timeout port IN 0
21/01/2021 à 19:14:19	OPERATIONAL	
21/01/2021 à 19:14:19	WARNING	Timeout port IN 0

Figure 27: Manta box states timeline

6.6 Manta configuration box

This is a non-visible box that hosts a reusable configuration and will be shared by multiple boxes.

Used for the setting of:

- Serial ports
- CAN bus
- Handle NMEA2000

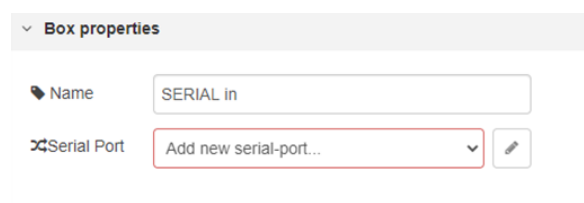




Figure 28: Serial box configuration panel

In the box properties tab of a box, the icon  indicates that the parameter corresponds to a configuration box. It is then possible to select an existing configuration from the drop-down list or to create a new configuration by clicking on the icon .

The configuration box management panel is accessible via the menu *Manta->Configuration boxes*.

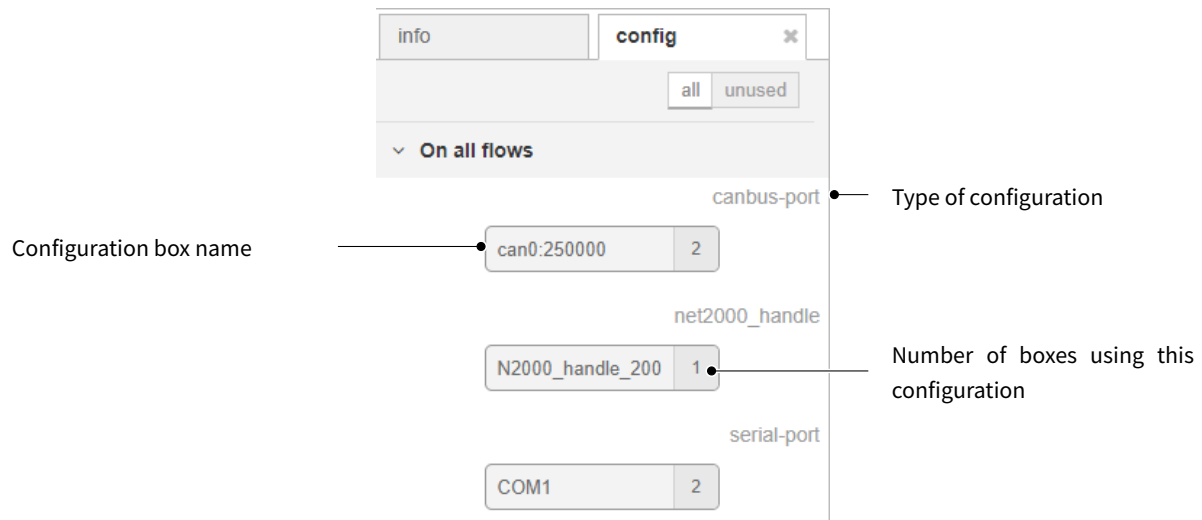


Figure 29: Configuration boxes



Unused configuration boxes generate unnecessary compute and memory load. They should therefore be deleted.

6.7 The deployment

Pressing the deploy button  triggers:

- Saving the Manta graph in Exocet
- Restarting the Exocet application

If the button is red, it means that at least one change has taken place since the last deployment. This may be due to:

- Adding / removing a box
- Adding / removing a wire
- Removing a configuration box
- Changing a Parameter of a Box
- Moving a box



Boxes that have been modified since the last deployment display a blue circle in their upper right



6.8 The messages

A **message** is information transmitted between two boxes via the wire connecting an output port and an input port.

The content of the message is a dictionary which associates the name of the variables with their value, type and other properties. The messages are not necessarily of fixed frequency.

Data types:

- *Boolean*: true or false
- *Integer* on 4 bytes
- *Floating* on 4 bytes
- *String*: max length of 32 characters
- *CanFrame*: id: 4 bytes, dlc: 1 byte, data: 8 bytes
- *Opaque*: 2048 single bytes buffer (used for serial, UDP or TCP data...)
- *FloatArray*: 1024 floating elements buffer
- *FloatTable*: 32x32 floating elements buffer (used for upwash or polar tables...)

Basic visualization with probe:

The probe box displays message sent by one or several output ports

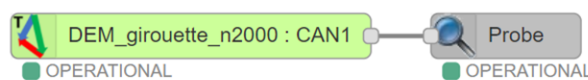


Figure 30: Manta probe box

The messages information are available by double clicking on the probe box:




Figure 31: Information on messages



A probe box can be connected to several output port

Advanced visualization with probe:

The mode  is used to display the history of received messages. In this mode, once the history is paused with the button, it is possible to download a CSV or JSON data log file.

In order not to penalize the performance of the web browser, the data refresh is limited to 300 ms (3.3 Hz).

If the data frequency is greater than 3.3 Hz, you must use the full speed option so that all the data is available in the CSV or JSON files.



Probes are the only boxes whose name does not have to be unique



A probe with the *full speed* option uses a lot of computing power. Its use must be reserved for the development phases. This option should not be used in a running system under any circumstances.

6.9 The message manipulations

Concatenation:

The *VarMux* box concatenates two messages

Message in 1 (F1 Hz)

- Data 10
- Data 11
- Data 12

Message in 2 (F2 Hz)

- Data 20
- Data 21



Message out (F3 Hz)

- Data 10
- Data 11
- Data 12
- Data 20
- Data 21

Figure 32: VarMux box

The box offers 3 options to control the frequency of the outgoing message:

- *Fastest*: message sent each time a message is received at the input port
- *Slowest*: message transmitted at the frequency of the slowest input message
- *Periodically*, from 0.1 to 50 Hz



The *fastest* option can generate a high message frequency due to the desynchronization of the received messages.



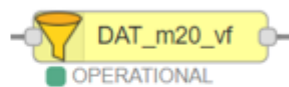
Make sure that there is no data with the same name in the dictionaries received by the VarMux box. Otherwise, some data will be lost after concatenation is complete.

Filter variables:

The *VarFilter* box removes variables from the dictionary

Message in (F Hz)

- Data 0
- Data 1
- Data 2
- Data 3



Message out (F Hz)

- Data 0
- Data 1

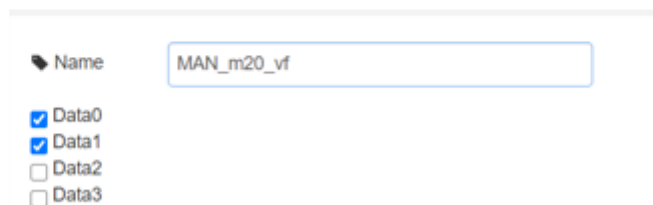


Figure 33: VarFilter box



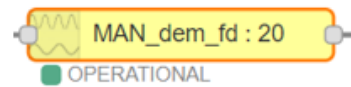
The frequency of the output message is the same as that of the input message

Undersampling:

The *freq_divider* box decreases the frequency of a message

Message in (FREQ Hz)

- Data 0
- Data 1
- Data 2



Name:
 Divider:
 Outputs:

Message out (FREQ/20 Hz)

- Data 0
- Data 1
- Data 2

Figure 34: freq_divider box

The *VarMux* box can also be used for frequency decrease, especially if the desired frequency is not a factor of the frequency of the input message.



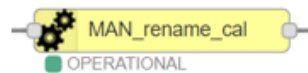
The box does not apply any interpolation function, it just selects a message from N.

Rename variables:

The *Calibration* box can be used to rename certain variables of a message.

Message in (FREQ Hz)

- count



List of variables to calibrate :
 Variable : Type:
 Output name : Unit:

Message out (FREQ Hz)

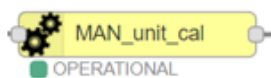
- Count_bis

Figure 35: Rename variables**Unit modification:**

The *Calibration* box can be used to specify or modify the unit of a variable.

Message in (FREQ Hz)

- Count: val



Message in (FREQ Hz)

- Count: val Kn

List of variables to calibrate :

Variable :	count	Type:	None
Output name :		Unit:	Nautical Knots

Figure 36: Modify unit

Selection of redundant messages:

The *FailSafe* box manages several redundant inputs.

In automatic mode, the highest priority valid port is selected. If no data is received at the timeout or if data is detected as invalid, the lower priority entry will be selected. The highest priority entry will be selected again if data reception is resumed or data retrieve a valid status.

At any time, the user has the flexibility to force the selection of a port.

If the *switch* option is selected in the settings panel, only manual port selection is available.

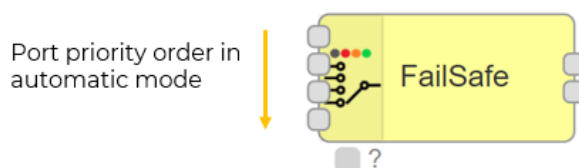


Figure 37: FailSafe box



Port expiration times are defined in the Port labels and timeouts tab

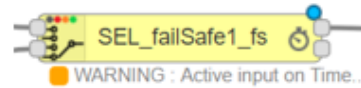
If the name of the variables of the messages received differ, it is then possible to configure the renaming of these variables in order to have an output dictionary with identical variable names, whatever the selected entry.

Message anglais (0.1 Hz)

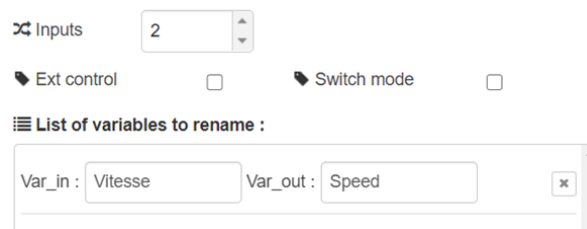
- Angle : 2°
- Speed : 1 Kn

Message francais (1 Hz)

- Angle : 200°
- Vitesse : 50 Kn

**Message (1 Hz)**

- Angle : 200°
- Speed : 50 Kn

Figure 38: FailSafe box variable rename**Figure 39:** Rename Vitesse to Speed**Exchange between flows:**

Link_in and *link_out* boxes are used to send a message from one flow to another. They can also be used to send a message inside a flow without making the graph heavy.



By clicking on a link, a label indicates in which flow the connected link is located. You can access the connected flow by clicking on the label.

Exchange between Exocets:

The boxes *exocet_in* and *exocet_out* are used to send messages from one Exocet to another. To overcome any data name conflicts, the data is renamed with the BOXNAME_VARIABLENAME format before transmission.

Maintenance messages:

The box *System* exports the following maintenance data:

- CAN buses Load
- CPU load

- RAM memory occupation
- SD card space occupation
- Duration of operation
- Processor temperature



A *System* box must be present in the Manta graph to have this data visible in the *status* page of the web application.

The box *manta_status* informs about the number and name of Manta boxes in the WARNING or ERROR state. This information can be logged data or for a user notification

6.10 Manta processing

Manta provides a number of data processing boxes:

- The *calibration* box is used to rename a variable, change its unit, or adjust the precision and accuracy of its value. To do this, different adjustment methods are available: by affine function or by correction table (LUT).
- The *moving_mean* box performs sliding calculations (average, variance, median, min / max...) for a list of variables.
- The *damping* box applies filtering.
- The *delay* box applies delay to precisely synchronize several messages.
- The *expression* box permits to apply all fundamental arithmetic operations and conditionnal branching.



The *calibration* box can be associated with a *curve* widget to view the adjustment function in a dashboard.

6.11 Dataloging

On receipt by the *datalogger* box, the data is stored on an SD card. Optionally, they can be transmitted to our data management application in the cloud <https://exocet.cloud/>

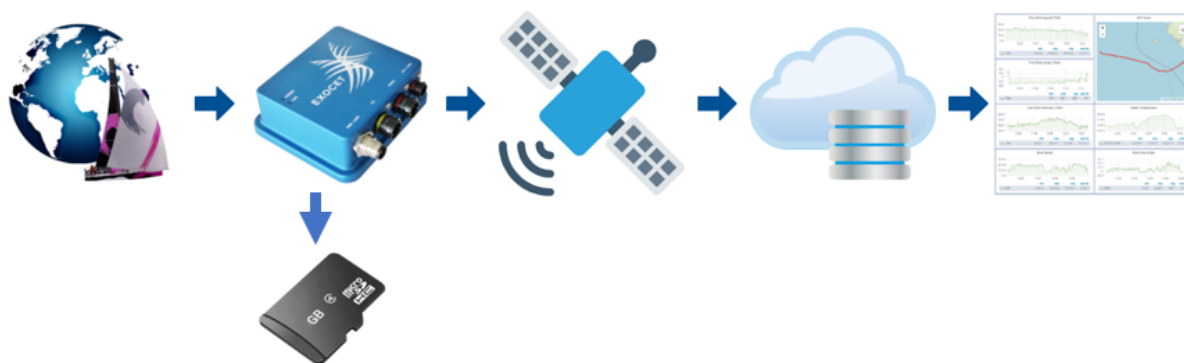


Figure 40: Data logging



This feature is specific to Exocet Blue.



To overcome any data name conflicts, the data is renamed with the BOXNAME_VARIABLENAME format before recording or transmission to the cloud..

6.12 Python scripting

The *python_function* box offers a Python 2.7 interpreter accompanied by a wide choice of libraries, including NumPy.

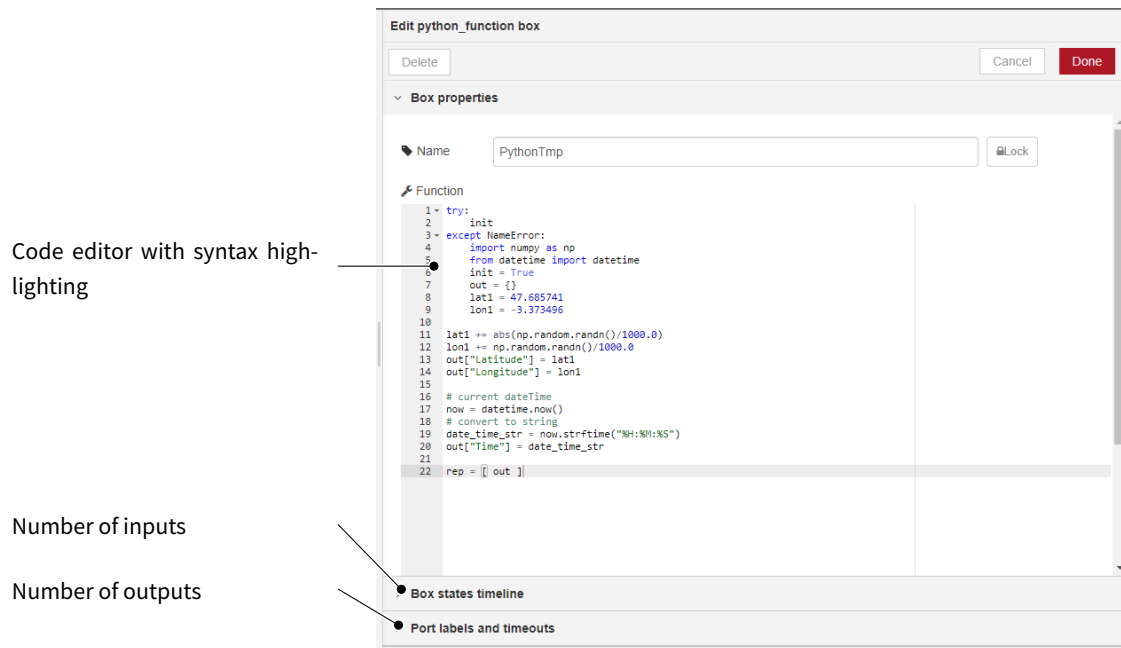


Figure 41: Python editor

General principle of operation:

The script is executed each time data is received by an input port. After executing the script, messages may be sent from the output ports.



The *tick* box can be used to clock the execution of a script that does not receive external data.

Special Python variables:

- *msg*: Dictionary containing the data of the port that triggered the call to the script
- *port*: Index of the port that triggered the call to the script
- *rep*: Table of dictionaries to export data out of the box. The size of the array = number of output ports. Each element of the array is either a dictionary or None. If the item is None, no data will be output from the corresponding port.

The stages of operation:

1. A message is received by one of the input ports of the *python_function* box
2. Waiting for the interpreter to be available
3. Loading the local context of the script
4. Creation of the *msg* variable from the received message. Manta variables are converted to Python variable.

5. Assignment of the index of the input port in the *port* variable
6. Running the Python script
7. Creation of the messages which will be sent by the output ports according to the variable *rep*.
Python variables are converted to Manta variables.
8. Saving the local context
9. Releasing access to the interpreter
10. Sending messages via the output ports

Hello World:

```
out={};
out["hello"] = "Hello World!";
rep = [ out ];
```

Manta to Python type conversion:

Manta	Python
Boolean, float, integer, string	Python equivalent
Opaque	ByteArray
Array, Table	FloatTable
CanFrame	pixel.CanFrame

pixel.CanFrame class is made of fields:

- Id: integer for CAN ID
- Bytes: ByteArray of data

Python to Manta type conversion:

The Python data is converted into Manta data when creating the messages to be sent to the output ports. The python array *rep* contains the list of messages to be transmitted.

Python	Manta
Boolean, float, integer	Manta equivalent
String	String truncated to 32 characters max
ByteArray	Opaque truncated to 2048 bytes
FloatTable	Array limited to 1024 float, or Table limited to 32x32 float

Python	Manta
pixel.CanFrame	CanFrame

Caution:

There is only one Python interpreter. When one box executes its script, the others are blocked, awaiting the availability of the interpreter. In order not to penalize the general performance of the system, it is therefore advisable to limit as much as possible:

- The complexity of scripts
- The frequency of messages sent to Python boxes. 10Hz maximum.


The use of Python boxes should be reserved for processing that cannot be performed by another box in the catalog (ex: Expression box).

6.13 Manta libraries

Libraries are used to import or export all or part of a flow.

Create a library:

From Manta:

1. Select a set of boxes with the mouse or all the boxes of a flow with CTRL + A. From menu, select *Export->Library*
2. Give a name to the library then press the button 

Instantiate a library in a flow:


From Manta:

1. From the menu, select Import-> Library-> Library name
2. Click in the flow at the desired location to place the content of the library



If the library contains Link boxes, these must be recreated.

Library management:

The library management interface is accessible via the button  on the *Settings* page of the web application.

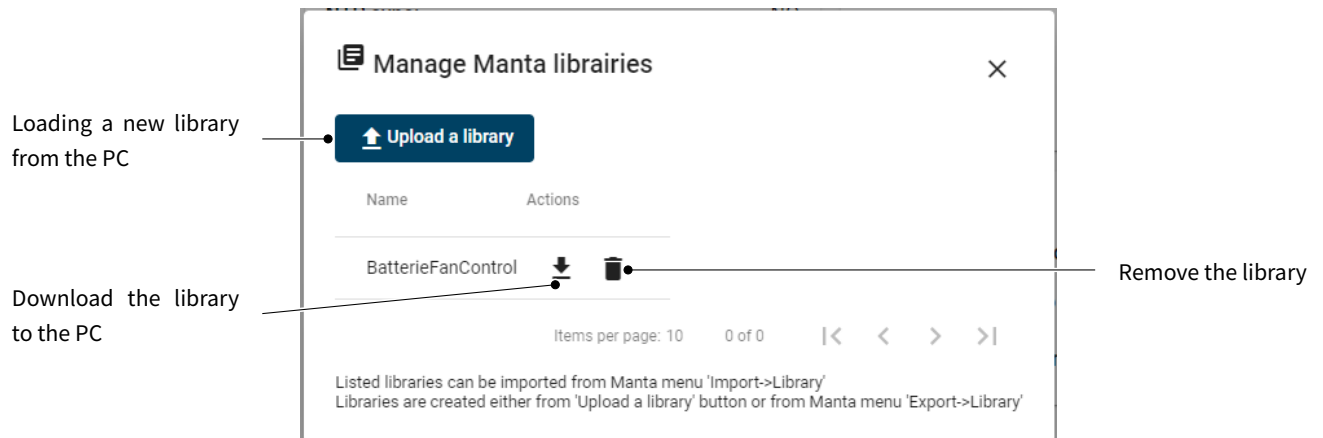


Figure 42: Manta library management panel

6.14 Manta good practices

To keep a clear Manta Graph which permit to facilitate maintenance and evolution, and to avoid use-less CPU load, some good practice have been identified:

Organize the Manta graph:

- Separate functional processes into relevant flows of modest size.
- Use *Link* boxes inside a flow to avoid wire crossings.
- Define and respect a naming convention for box and flow names.
- Give a name to the *Link* boxes for a better understanding of the connections between boxes.

Mistakes to avoid:



- Do not send angular variables without unit to the boxes *damping*, *moving_mean* and *calibration*.
- A box with periodic transmission option used before a *failsafe* box prevents the box from detecting loss of data reception.
- When instantiating a Manta library, the *Link* in and out boxes must be replaced.
- Avoid having several users modifying the Manta graph at the same time.
- Remember to save the configuration after an update, using the button *Export Configuration* from the Web application *settings* page.

To limit CPU load and memory occupation:

- Limit the frequency of too fast messages as early as possible in the graph, especially those which will be sent to a Python box.
- The frequency will be controlled by the *Probe* box
- The frequency limitation will be done with the *freq_divider* or *VarMux* boxes.

- Limit the complexity and the number of *Python* scripts.
- Filter variables as soon as possible to limit number of useless variables in messages with a *varfilter* box.
- Limit the number of *probes* as much as possible. No *probe* in full-speed mode outside the debugging phase.
- Remove configuration boxes not referenced by a box.

To check the correct operation:

- Check the status of boxes from the navigation bar of the web application .
- Use the timeline of boxes in the WARNING or ERROR state to get information on the timing and reason for the issue.
- Check the CPU load and memory occupation from the *Status page* of the web application. For a healthy system:
 - Total CPU and each CPU < 80%
 - RAM memory < 80%
 - Temperature < 80°
- Check the *uptime* from the *status/System* page. This time is reset to zero in event of an untimely restart.
- In case of an anomaly, save the system logs using the  button on the *status* page for analysis by Pixel Sur Mer.

Naming flows:

Here is a convention used to name flows : **[prefix]_[TRIGRAM]restofthename**

- Depending on the majority processing carried out by the flow, the prefix must be of the following type :

Prefix	Treatment
in	Sensor input or data reception
out	Output to actuators, bus or visualisation
pro	Computation or data processing
wnd	Wind calculation
ovl	pilot overlay
plt	Pilote interface
simu	Simulation

- The trigram is made up of 3 or 4 capital letters and must correspond to the name of the flow's main function. (e.g. INS, MAST, etc.). It is this trigram that will be used when naming flow elements.
- The rest of the name is in lower case. It should be as short as possible, and complete the flow name if necessary for greater clarity.

Splitting flows:

Pre-defined, standard reference flows providing an established data dictionary :

Name	Function
in_INS	Inertial data reception and processing
in_BSP	Receive and process boat speed data
in_GRE	Receive and process rigging load data
in_NRJ	Receive and process energy data: battery, hydrogenerator, solar panel, etc.
in_MAST	Reception and processing of mast data
in_MHU	Reception and processing of MHU data
in_GPS	GPS data reception and processing
in_out_HMI	Human machine interface (blink keyboard, display)
wnd_LEEWay	Drift calculations
wnd_WIND	Wind calculation
out_LOG	Log supply to recordings
out_ToH5K	Data supply to H5K equipment
plt_BAG	Dialogue with B and G pilot
plt_NKE	Dialogue with NKE driver
ovl_MNG	Overlay Manager calculation
ovl_HSafety	Heel safety overlay calculation
ovl_HRegulation	Heel regulation overlay calculation
ovl_AWARegulation	awa regulation overlay calculation
ovl_SPDRegulation	speed regulation overlay calculation
ovl_VMGPerf VMG	performance overlay calculation

For flows containing input boxes (e.g. Analog, CAN, Series, UDP, etc.): If the data concerned will be used by several functions, the processing of their reception must be put in a dedicated flow (prefixed with “in_”). This flow will then contain the processing of calibration, failsafe and any filtering. Otherwise, it is advisable to manage all processing (and associated processing) in the same flow. It will then be named according to the function performed.

Naming “out” links between flows

Here is a convention used to name out links : **ExplicitDictionaryName**

Use of Camel Case notation with a name representative of the exported data contained in its dictionary. Note: Its name cannot be taken from the destination for this box, as it may be linked to several recipient “in” boxes.

Naming “in” links between flows

Use of Camel Case notation, distinguishing between 2 cases:

- If the “In” box is linked in the flow with several functions, then its name must be representative of the imported data contained in its dictionary : **NameExplicitDictionary**
- If the “In” box is linked to a single function in the flow, then its name can be representative of the destination box : **NameExplicitDestination**

Naming function boxes

Here is a convention used to name function boxes : **[TRIGRAM flow]_[CamelCaseName]_[suffix]**

- Recall the trigram of the flow containing the box
- We associate the name of the data, treatment or material concerned by the box
- Add a suffix (optional) indicating the type of calculation performed, among the following types:

Suffix	Processing
ana	Analog reception
cal	Calibration
py	Python
damp	Temporal filtering
mov	Sliding window calculation (average...)
vf	Variable filtering in a dictionary
fs	FailSafe function
sel	Failsafe box used as selector

Suffix	Processing
pid	PID controller
btn	Button box
mux	Box used to multiplex several dictionaries
udpIn	Read UDP protocol port
udpOut	Write UDP protocol port
n2kIn	Read NMEA2000 protocol
canIn	Read CAN interface
canOut	Write CAN interface
serialIn	Read serial port
serialOut	Write serial port
tk	Tick box

Naming variables

Here is a convention used to name variables : **ExplicitCamelCaseName[_Suffix]**

- Use Camel Case notation with a name representative of the data concerned.
- Possibly suffixed for precision: _raw, etc...
- The following variables have standard names and correspond to the following data:

Name	Units	Description
AWA	°	Apparent wind angle / boat
AWS	kn	Apparent wind speed
TWA	°	True wind angle / boat
TWD	°	True wind direction (wind/water)
TWS	kn	True wind speed (wind / water)
MastRot	°	Mast angle along vertical axis / boat
MastRake	°	Mast angle in transverse axis / boat
MastCant	°	Mast angle in longitudinal axis / boat
BoatSpeed	kn	Longitudinal boat speed / water

Name	Units	Description
Leeway	°	Leeway angle / water
Course	°	Boat speed direction / water
SurgeVel	m/s	Longitudinal boat speed / land
SwayVel	m/s	Lateral boat speed / land
HeaveVel	m/s	Vertical speed boat / land
SurgeAcc	m/s ²	Longitudinal boat acceleration / land
SwayAcc	m/s ²	Lateral boat acceleration / land
SOG	kn	Boat speed / land
COG	°	Boat course / land
Lat	°	GPS position latitude
Long	°	GPS position longitude
Heading	°	DireBoat heading / north
Trim	°	Boat angle along transverse axis / land
Heel	°	Boat angle along longitudinal axis / land
YawRate	°/s	Boat rotation speed in vertical axis / land

7 Exocet web application

7.1 Exocet web app overview

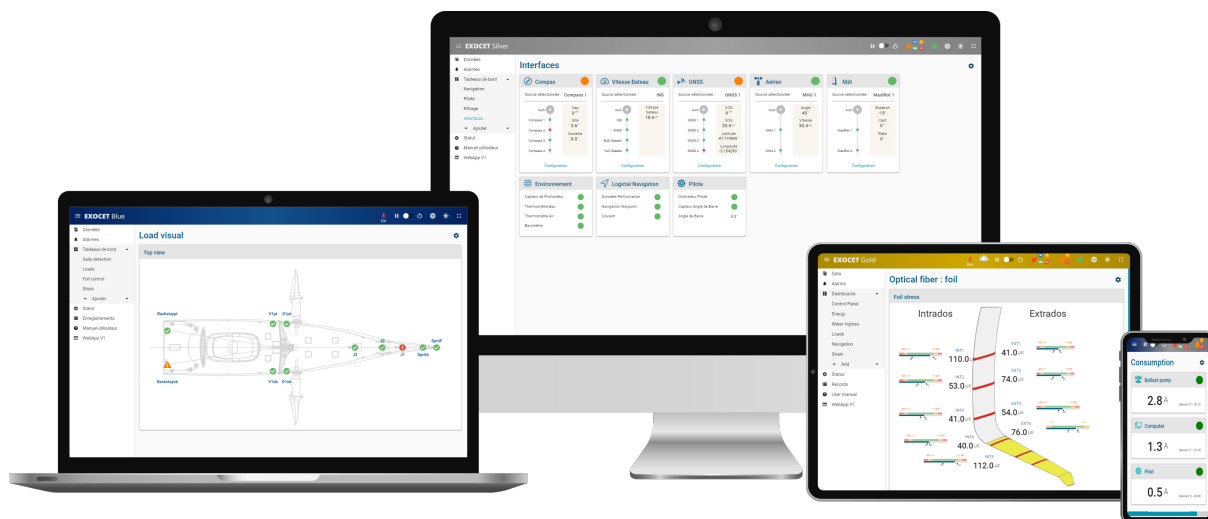


Figure 43: Multi screen UI

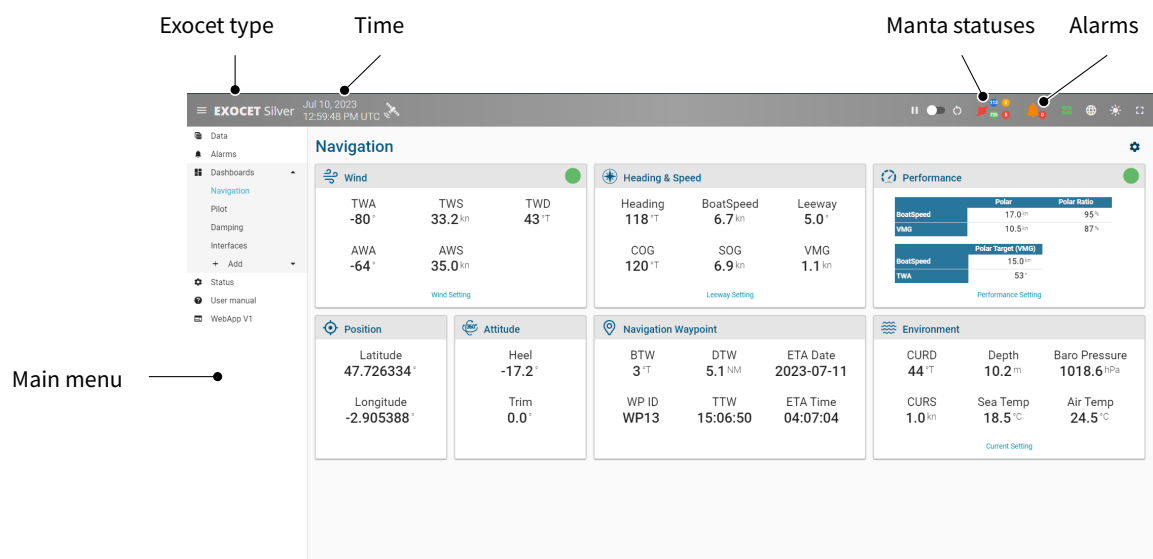


Figure 44: Exocet web page

Exocet Web App is a software user interface accessible via all mobile media and all web browsers. It is composed of a Navigation Bar at the top where can be found:

- Visual status of Manta boxes, with a link to the Manta workspace.
- Visual alarm status.
- A synchronised date/time indicator.

The main menu on the left offers several links to other pages:

- *Data* leads to the Data Table page.
- *Alarms* leads to the alarm management page.
- *Dashboards* provides a drop-down menu of the dashboards available for the product.
- *Status* leads to a general configuration page for the Exocet.
- *User manual* downloads the latest version of the user manual (this document).
- *WebApp V1* provides access to the previous web version of Exocet products.

In the center, contents depend on the selected page.

7.2 Exocet Web app “Status” page

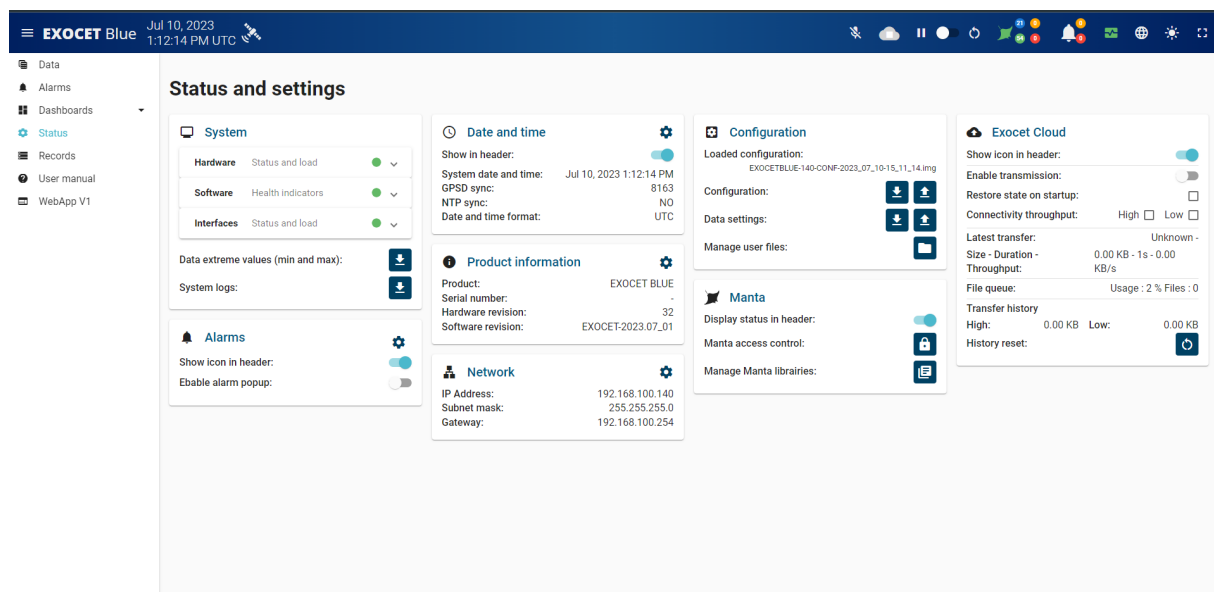


Figure 45: Exocet web app setting page

Settings page is used to configure:

- User interface: language, date/time, night/day display mode, the list of variables to display in page footer
- Data options: default decimal for displaying floating data. This setting can be overwritten for a given data from the “Data Table” page.

- System settings:
 - Time and Clock synchronization:
 - * Apply PC time to system
 - * Configure GPSD UDP port for time synchronization
 - * Configure remote NTP server for time synchronization. Note that any Exocet product can be used as Chrony server.
 - Export the current system configuration:
 - * Manta graph
 - * List of data to display on footer
 - * Data alarms, alias and decimals
 - * Graph list
 - * Dashboards
 - * Default decimals setting
 - * Mobile pages definition
 - Upload a system configuration
 - Reboot the entire system or restart the Manta application
 - Upload a new firmware (preserve the current configuration)
 - Create a Manta access control
 - Handle Manta librairies
 - Reset to factory state. It trashes the system configuration, but not the Network settings nor the records files.
- Record export (Exocet Blue only): Configure how data records are CSV format encoded, and download PSM Decoder software installer. This software is mandatory to export record files from Exocet to user PC.

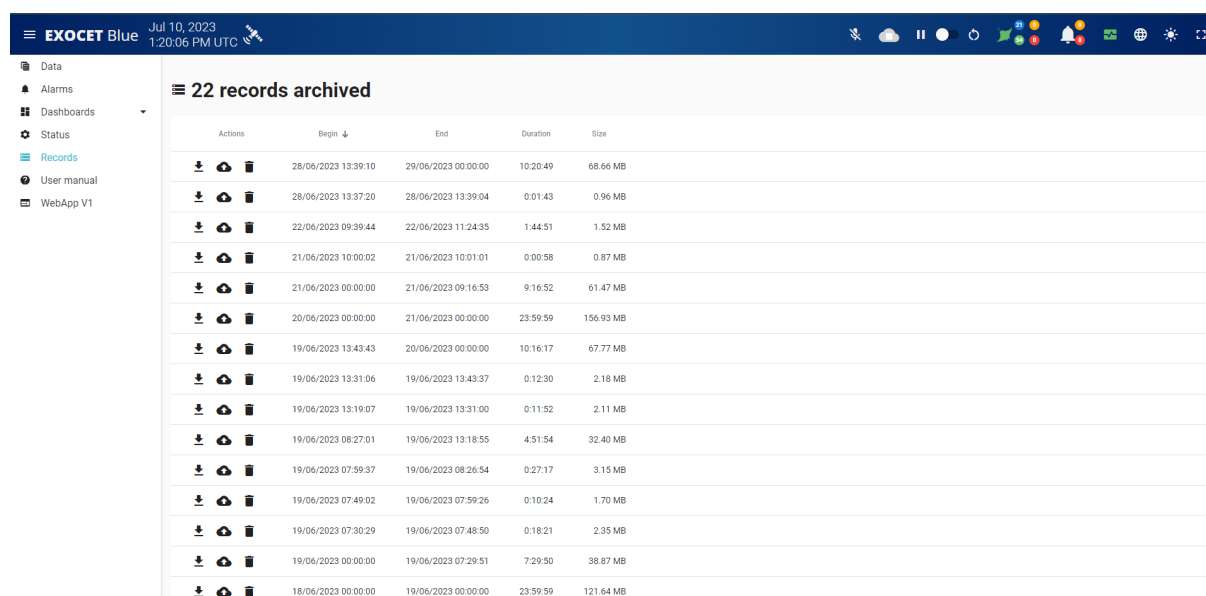
Some of the system states are also shown on this page :

- System date & time
- GPSD & NTP synchronization status
- System uptime
- Total and detailed CPU loads
- RAM memory usage
- System temperature
- Records space usage
- Number of record files usage
- Download system logs: to be provided to Pixel Sur Mer for problem analysis

An also some product informations:

- Product name
- Product serial number
- Hardware reference
- Software version
- Name of last configuration loaded

7.3 Exocet Web app “Records” page



22 records archived				
Actions	Begin ↓	End	Duration	Size
	28/06/2023 13:39:10	29/06/2023 00:00:00	10:20:49	68.66 MB
	28/06/2023 13:37:20	28/06/2023 13:39:04	0:01:43	0.96 MB
	22/06/2023 09:39:44	22/06/2023 11:24:35	1:44:51	1.52 MB
	21/06/2023 10:00:02	21/06/2023 10:01:01	0:00:58	0.87 MB
	21/06/2023 00:00:00	21/06/2023 09:16:53	9:16:52	61.47 MB
	20/06/2023 00:00:00	21/06/2023 00:00:00	23:59:59	156.93 MB
	19/06/2023 13:43:43	20/06/2023 00:00:00	10:16:17	67.77 MB
	19/06/2023 13:31:06	19/06/2023 13:43:37	0:12:30	2.18 MB
	19/06/2023 13:19:07	19/06/2023 13:31:00	0:11:52	2.11 MB
	19/06/2023 08:27:01	19/06/2023 13:18:55	4:51:54	32.40 MB
	19/06/2023 07:59:37	19/06/2023 08:26:54	0:27:17	3.15 MB
	19/06/2023 07:49:02	19/06/2023 07:59:26	0:10:24	1.70 MB
	19/06/2023 07:30:29	19/06/2023 07:48:50	0:18:21	2.35 MB
	19/06/2023 00:00:00	19/06/2023 07:29:51	7:29:50	38.87 MB
	18/06/2023 00:00:00	19/06/2023 00:00:00	23:59:59	121.64 MB

Figure 46: Exocet web app Records page

Records page, only available on Exocet Blue, allows to manage record files stored on the Exocet.

The two buttons on the left of each line permit to download the file with PSM format or CSV format. PSM format is an optimized format. PSM Decoder is required to decode a PSM format file.

A note can be attached to the file clicking on the right of the line.

The *Delete* button on the left permits to delete the selected files.

7.4 Exocet Web app “Data Table” page

Name	Valeur	Alarm
CPU 4	2 %	
CPU 3	1 %	
Status_Cpu2	1	
Status_Cpu1	1	
Status_Cpu4	1	
CPU 1	2 %	
CPU Total	1 %	
Status_Cpu3	1	
CPU 2	2 %	
Num CPU	4	
System uptime	884 s	
Manta loop detected	false	
Voltage too low detected	false	

Figure 47: Exocet web app Data Table page

Data Table page is used:

- to quickly display output data of any Manta box using the *Select* button on the left
- to access variable configuration. When pressing the pencil icon, the data configuration box is displayed.

Data settings for CPU 4

Settings
Alarm

Alias

Min value
Max value
0
100

☐ Min/Max calculation

Minimal number of digits displayed

Close Save

Data settings for CPU 4

Settings
Alarm

Create an alarm

Close Save

Figure 48: Data configuration box

For a given data, user can define:

- An alias: that will be used everywhere the data is displayed (page footer, graph, dashboards), but not on record files.
- A min/max value: that will be used by graphs and some dashboard widgets
- Number of decimals for display: to overwrite system default value for this data
- Number of digits displayed: force the minimum number of digits used to display this data
- An alarm: define with:
 - an enable option
 - a severity level
 - a high / low threshold to trigger an alarm
 - a high / low threshold to trigger a warning

If an alarm threshold is reached, an alarm or a warning will be raised. The “Alarm” page can be used to monitor the defined alarms.

7.5 Exocet Web app “Alarms” page

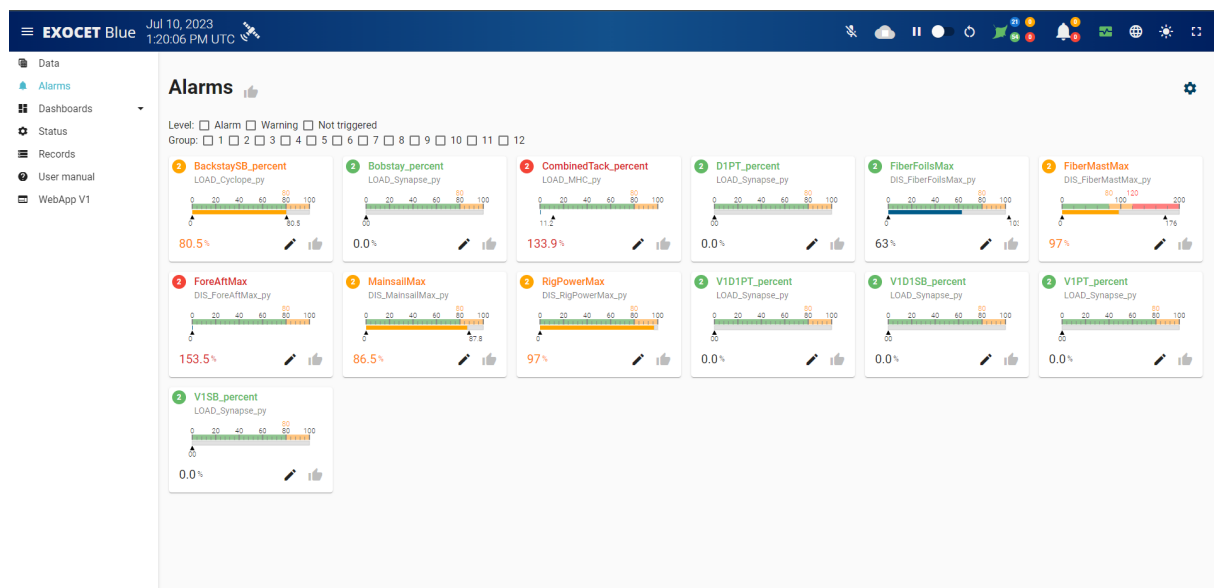


Figure 49: Exocet web app Alarm page

Alarm page displays linear gauges animation for each data with a warning and/or an alarm threshold defined.

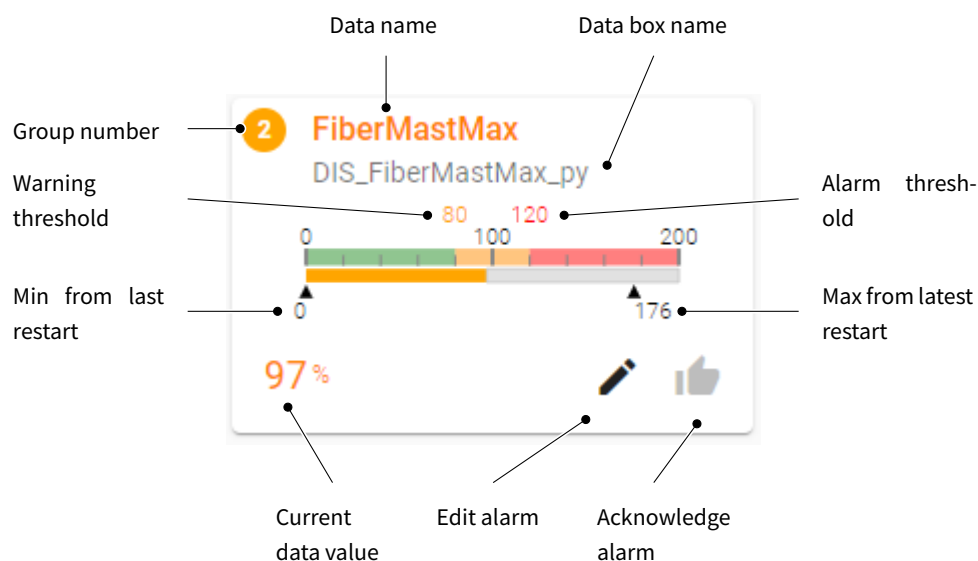


Figure 50: Une jauge linéaire d'alarme

Each gauge indicates:

- Severity level of the alarm
- Data and box name of the involved data
- Warning and /or alarm thresholds
- Max value reached since the system start
- Current value of the data
- An edit button to configure the alarm
- An acknowledge button to desactive the alarm, only available if required conditions are respected.

Active warning and alarm status are reported into the Navigation Bar at the top of the screen.

An *Alarm* box can be used into the Manta Graph to control a GPIO, and to ring a buzzer for example.

7.6 Exocet Web app Dashboards

Dashboards

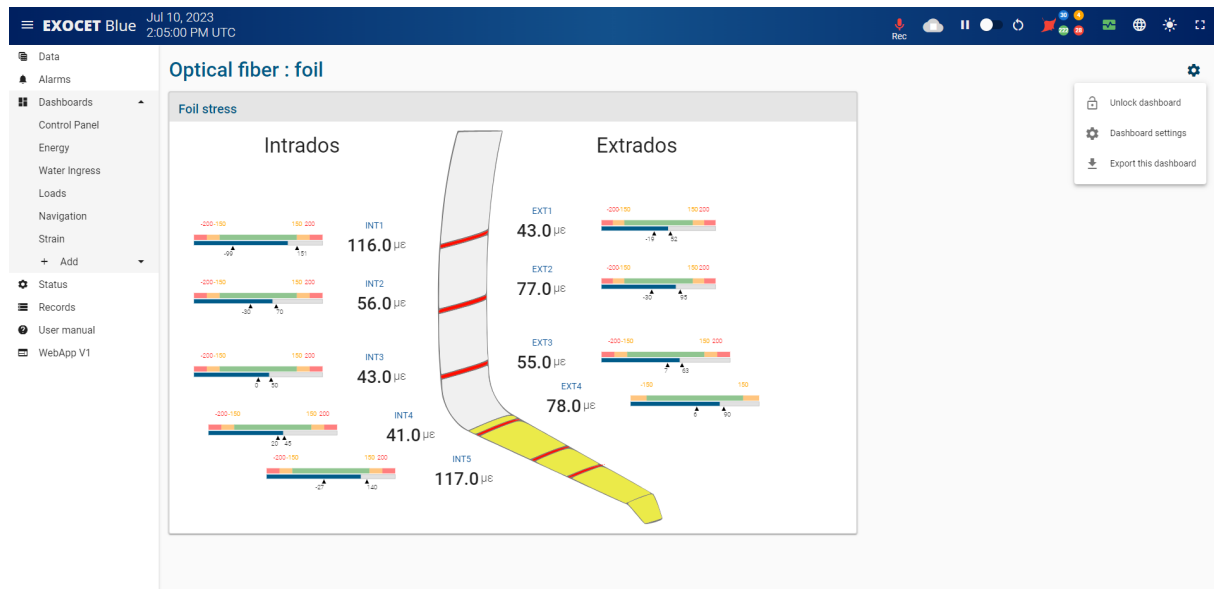


Figure 51: dashboard page overview

Dashboards pages are used to live display or control Manta Box data in a friendly environment.

A dashboard is a page that may display a background picture and a set of widgets. Widgets are animated graphical components that are associated with a Manta data. Widget location according to the background picture can be defined and saved. The graphical animation depends on the data.

By clicking on the cogwheel at the top right of the screen, the dashboard editing menu allows you to :

- Unlock the dashboard (allowing you to add or delete panels)
- Access the dashboard settings (name, English and French titles, menu display)
- Export a panel

Panel

Once the panel is unlocked, a notched wheel is available at the top right of each panel to access its editing menu, which includes :

- Unlocking the panel (to add a widget or freeze the widget geometry)

- Access to panel parameters
- Duplicate the panel
- Deleting the panel

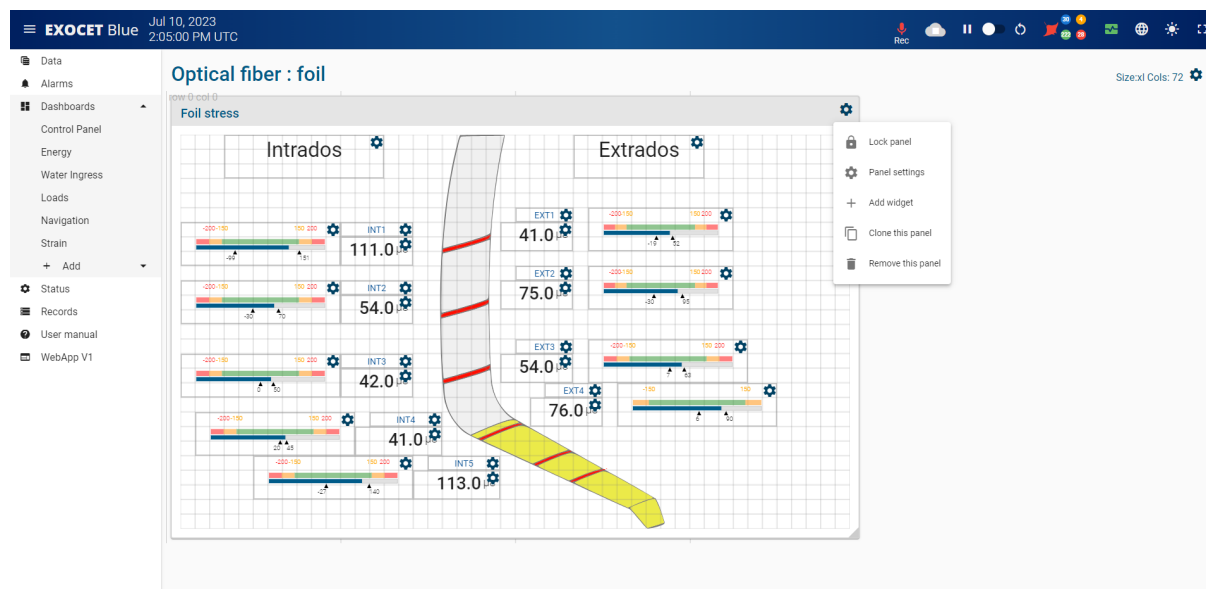


Figure 52: Panel modification

The panel parameters include :

- Name and Title
 - System name
 - English title
 - French title
- Display
 - Display mode (grid, free)
 - Layers (reserved for future use)
 - Widget size (changes the default size of panel widgets)
- Background
 - Background colour
 - Background image (the browser cache must be cleared and the page refreshed if an image with the same name as the previous one is selected)
 - Display size (Original, Adjust, Fill, Stretch)
 - Image repeat (No, Yes, Horizontal, Vertical)

- Horizontal position (Left, Centre, Right)
- Vertical position (Top, Centre, Bottom)
- Icon. Allows a small image to be placed to the left of the panel title
 - Icon
 - Hide icon
 - Icon colour
- Status. Adds a coloured indicator to show status.
 - Grey. *Bitfield* data at 0
 - Green. *Bitfield* data type of 1 or *Boolean* data type of True
 - Orange. *Bitfield* data type 3
 - Red. *Bitfield* data type 7 or *Boolean* tupe False
- Optional display. Used to set the panel layout.

Widget

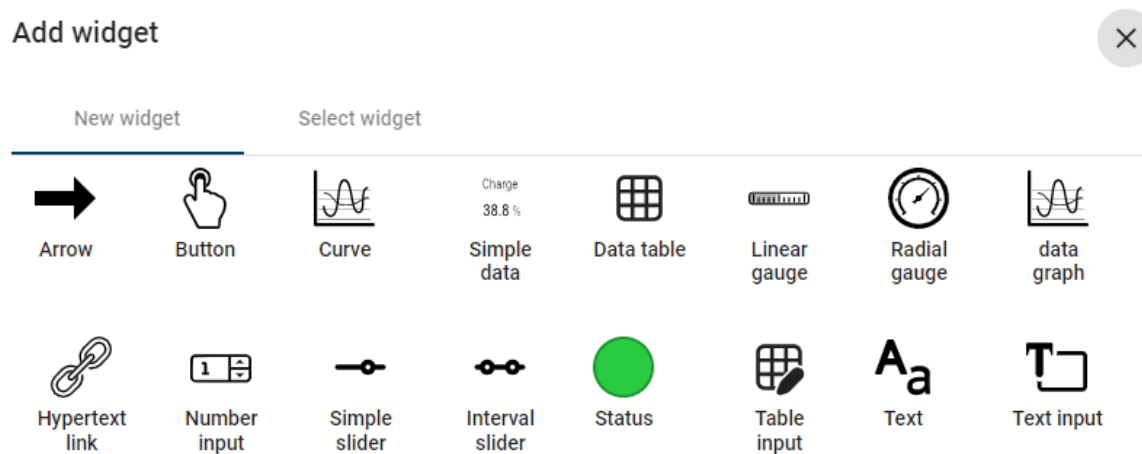


Figure 53: widget library

By clicking on the “Add a widget” button on a panel, the widget library appears.

Some widgets, such as *Simple Data* or *Gauge*, are used to display Manta box data. Others, such as *Button* or *Slider*, are used to control Manta box data using *Input_User* boxes (not available in all Exocet).

Clicking on the padlock button at the top right of a widget takes you to the widget editing menu, where you can :

- Configure the widget parameters
- Duplicate the widget
- Delete the widget

The parameters of each widget necessarily include the following three tabs with information to fill in or select:

- Name and Title
 - System name
 - English title
 - French Titer
- Size and colour
 - Size
 - Colour
 - Background colour
- Title display
 - Hide title
 - Use data name as title
 - Position (Front, Top)
 - Alignment (Default, Left, Centre, Right)
 - Text size (Smaller, Unchanged, Larger)
 - Bold
 - Title width in %.

Other tabs are available depending on the widget chosen. For example, you can select the data to be displayed, choose a visual gauge, change the appearance of a table, etc.

8 Communication with third party software

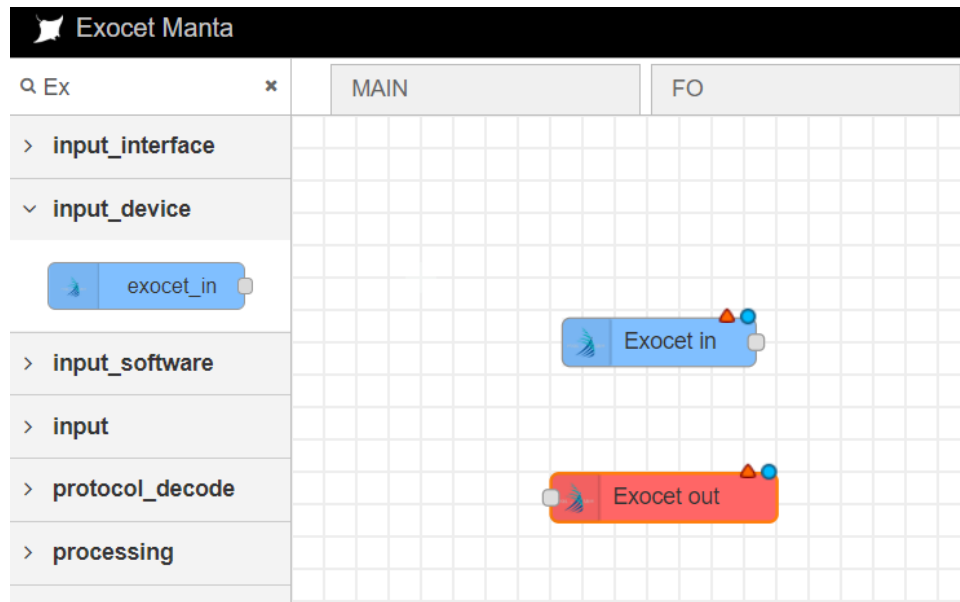


Figure 54: Exocet “in” and “out” boxes

“Exocet in” and “Exocet out” Manta boxes have been designed to propagate data between Exocet devices, mainly to send data from Exocet Silver or Gold to the Exocet blue to log these data.

These boxes are also the easiest way to communicate with third party software or devices. “Exocet out” box sends Exocet data to third party software or device. “Exocet in” box receive data from third party software or device.

Data are sent over Ethernet UDP, using a Pixel Sur Mer Proprietary protocol for data serialization. The protocol description can be provided on demand.

9 Tutorial : from sensor to cloud

The aim of this section is to give an overview of Exocet data management. It is recommended to read the **Manta** and **Exocet web application** section before trying this tutorial.

Here is a schematics of the data flow that will be explored in this tutorial.

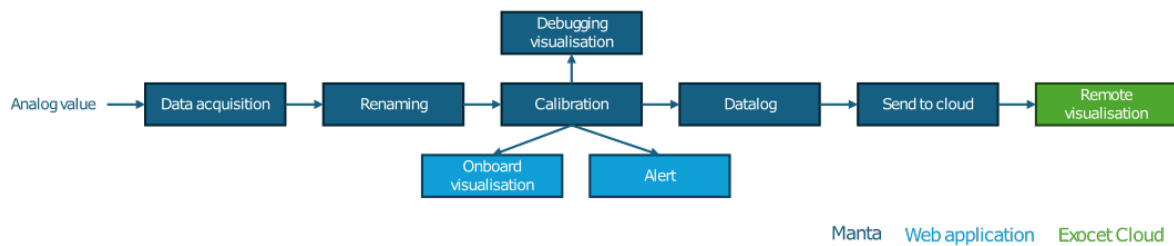


Figure 55: Data flow

The objective is to read an analog value, that describes a rudder angle, and send it to the cloud. Intermediate stages, such as calibration, onboard visualization and datalogging, will be presented to bring a step by step guide of Exocet capabilities.

9.1 Reading the analog data

The first step is to read the ananalog input data. To do so, open Manta, double click on the flow header (1), rename the flow to “TUTorial” (2) and validate the modification by pressing *Done* button (3). Throughout this tutorial, naming is based on Manta good practices. It is strongly recommended to rename each flow and box to keep the system organized.

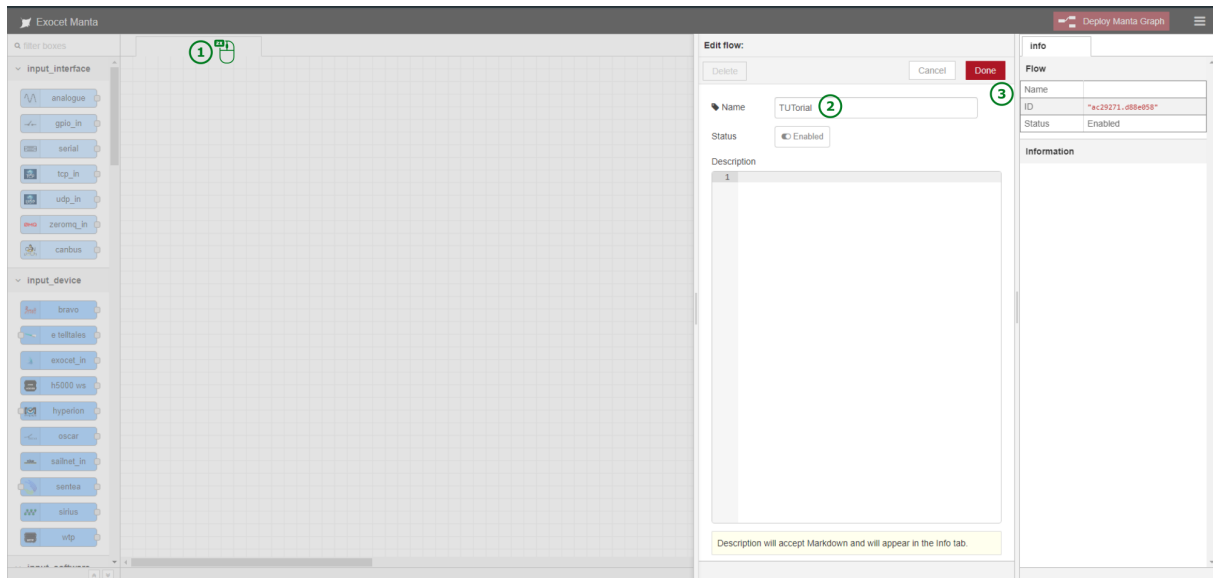


Figure 56: Rename manta flow

Now drag and drop an analog box (1), double click on it (2) and rename it to “TUT_RawRudderValue_ana” (3). You can validate this change by pressing *Done* (4) and deploy by pressing the dedicated button (5).

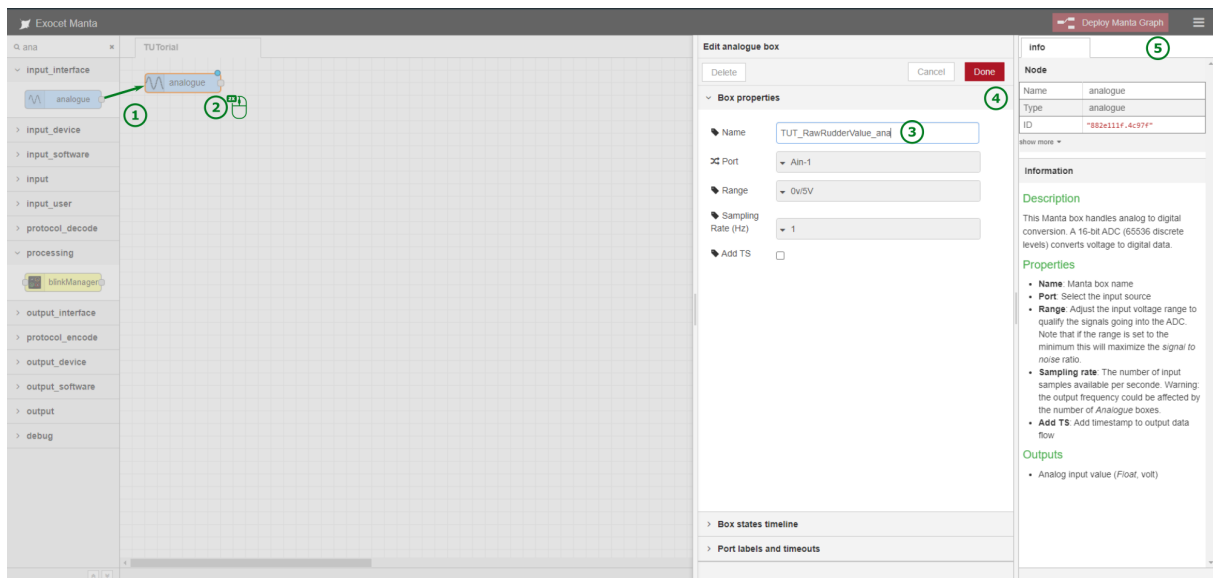


Figure 57: Read analog value

You have now a configuration that can read an analog value between zero and five volts at one hertz.

9.2 Visualising the data in a probe

Now that the system is acquiring data, it's time to visualize it. To visualize the input data, drag and drop a probe box (1) and link the *TUT_RawRudderValue_ana* output to the *Probe* input (2). To apply the created configuration, click on the *Deploy Manta Graph* (3). Below each box, a status appears, confirming that these boxes are now running. Open the probe box by double clicking on it (4). Data can be visualized on the edition area (5). The box *TUT_RawRudderValue_ana* is well publishing at one hertz a data named *Ain-1*, image of the analog1 voltage.

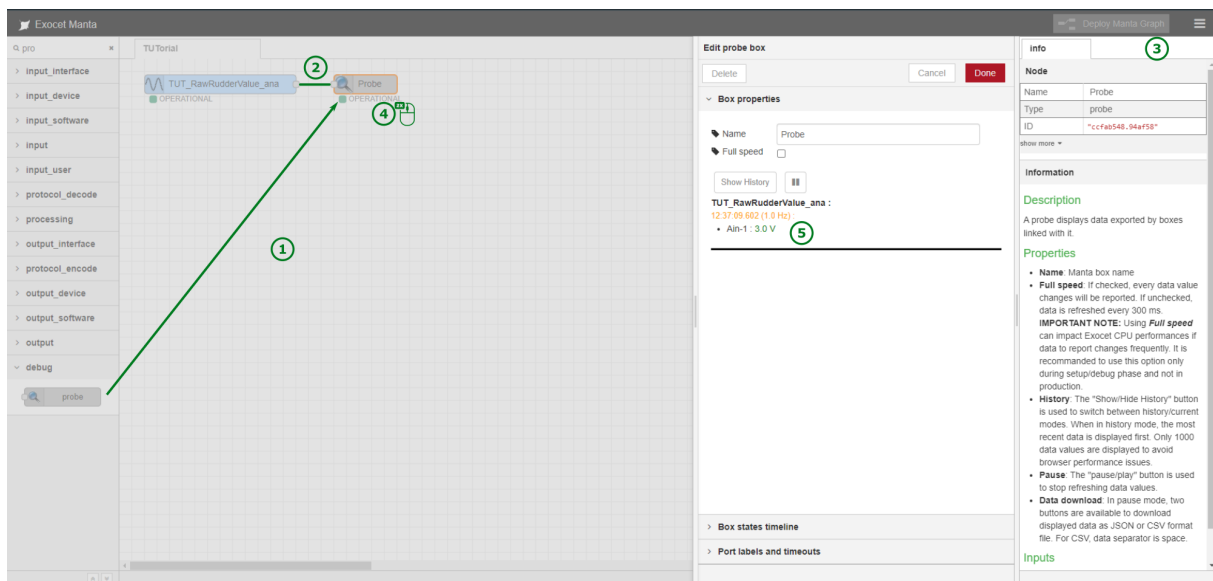


Figure 58: Probe visualization

9.3 Renaming the input data

As the data published by *TUT_RawRudderValue_ana* is named "Ain-1", the aim is to rename it with a relevant name. Drag and drop a Calibration box (1), open it (2) and rename it to *TUT_RawRudderValue_cal* (3). Then click on *New data* (4). On the *Variable* field, put the name of the input data, called "Ain-1" in this context. In this tutorial, the analog value is used to measure a rudder angle, rename it by writing "RawRudderValue" on the *output name* field (5) and validate by clicking *Done* (6).

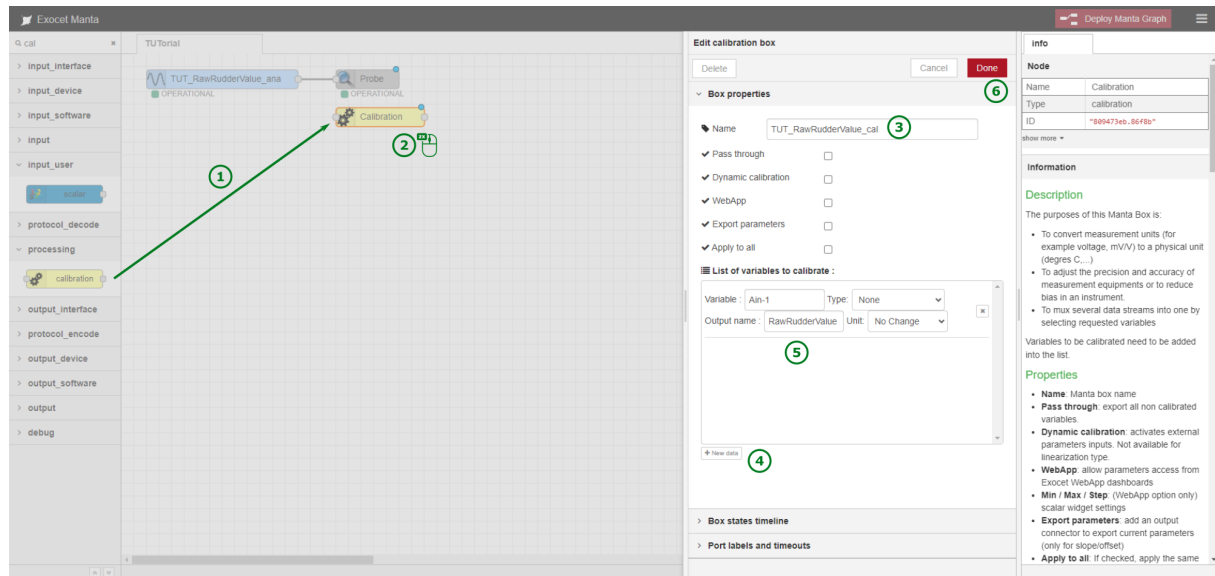


Figure 59: Rename input

You can now cable the *TUT_RawRudderValue_ana* box with the *TUT_RawRudderValue_cal* one (1) as the *TUT_RawRudderValue_cal* with the *Probe* (2). Deploy this schematics (3) and open the *Probe* (4). You can see both the raw data and the renamed one.

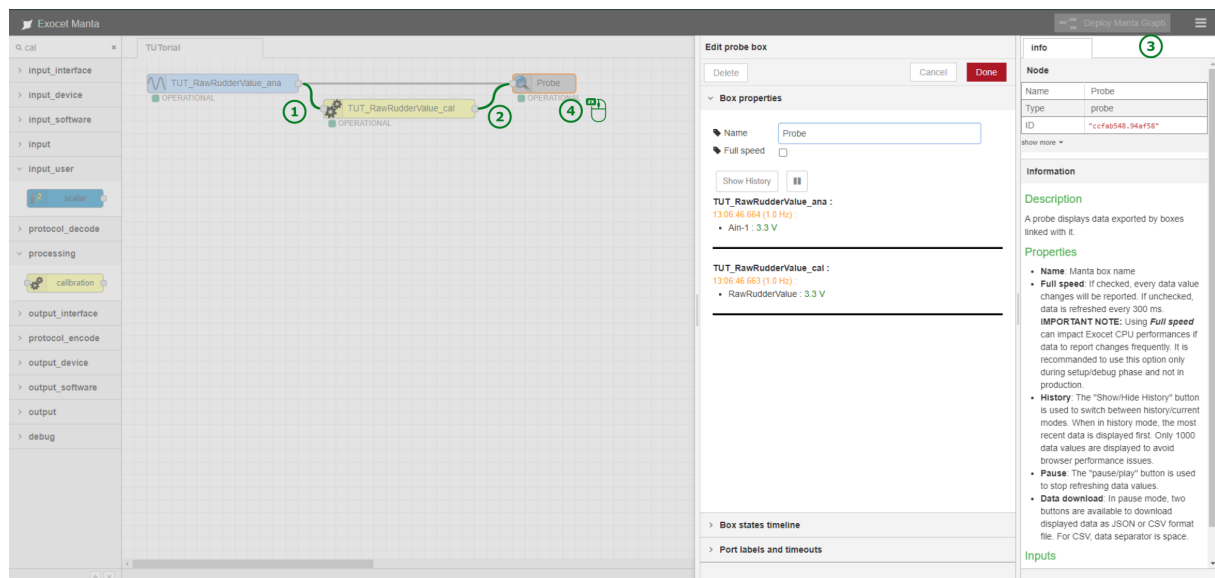


Figure 60: Visualise new data name

First step is done ! Let's now play with this data.

9.4 Calibrating input

As you may have noticed, a Calibration box has been previously used to rename a data. The calibration type was left to None to apply no correction, only renaming. This time, the data will be changed too. Reopen the *TUT_RawRudderValue_cal* box (1), change its name to *TUT_RudderAngle_cal* (2) and the output name to Rudder Angle (3). For the calibration part, set the calibration type to *Linearization* and the Unit to *ANGULAR_DEGREE_180* (4). Assuming that 0v corresponds to -180° and 5V to 180° , apply these values in the X1/X2/Y1/Y2 table (5). Then validate the changes by pressing the *Done* button (6) and deploy them by pressing the *Deploy Manta Graph* button (7).

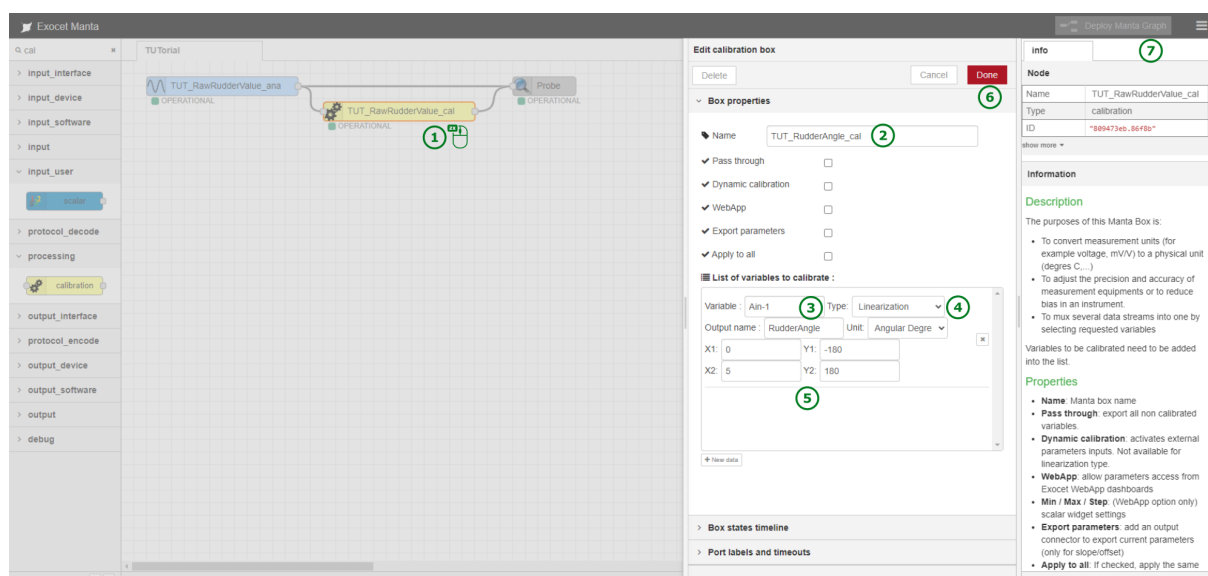


Figure 61: Calibration of input data

To have a better precision for displayed data in Manta, go to the Exocet graphical interface, on Status page and set *Default decimals for data display:* to 3. Then go back to the Manta and open the probe. You will see both raw and calibrated data with three decimals.

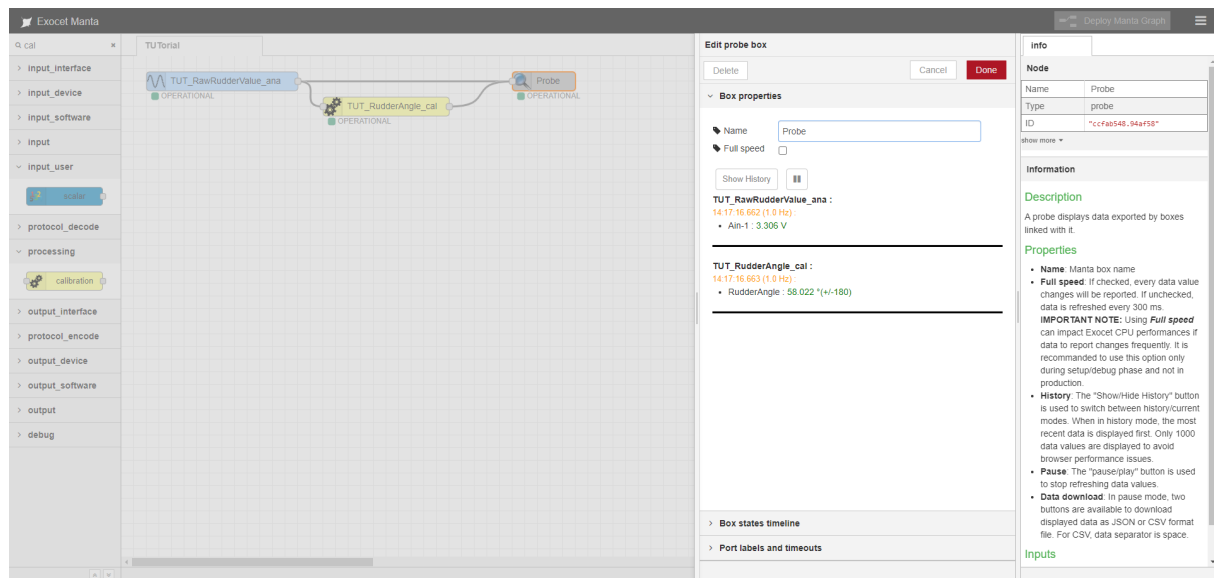


Figure 62: Calibrated data visualization

9.5 Visualizing data in the data table

Now that the logical part of our tiny code is done, we will focus on visualization tools offered by the Exocet. Go to the web application on the Data page. On the *Select port* field, indicate the name of a box output you want to monitor (“TUT_RudderAngle_cal” for this example). The list of data exported by the selected output connector is now accessible allowing to display current value and accessing data configuration. Note that if a box has a single output connector, has the same name as the box, else connectors have a #NUMBER postfix.

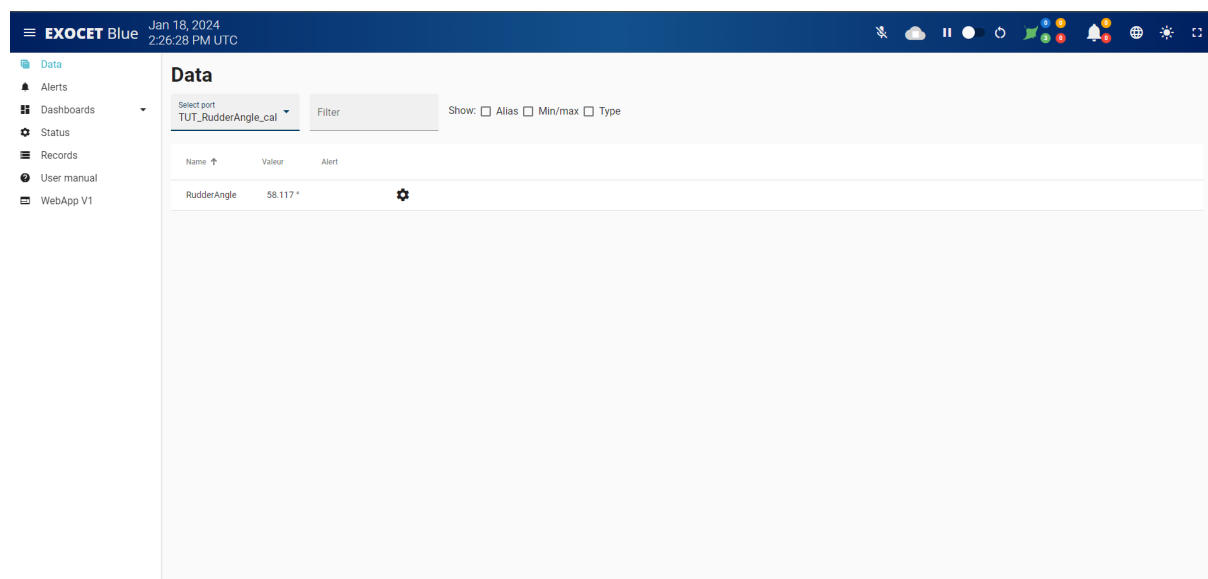


Figure 63: Data table visualization

9.6 Visualising data in a dashboard

The analog value is now converted to its true physical representation. It is time to display it on the graphical interface ! You first need to create a dashboard. On the web app, click on Dashboard/Add/New. On the opened menu, fill the *System name* with “my-first-dashboard” and the title by “My first dashboard”. Click *Save*. Wonderful, you have your first custom dashboard.

On the upper right of the dashboard area, click on the cogwheel and unlock dashboard. The dashboard is now editable. Click again on the cogwheel and choose the *Add panel* option. As done before, on the opened menu, fill the *System name* with “my-first-panel” and the title by “My first panel”. Click *Save*. Panels and dashboards allow to find the desired layout that suits your needs.

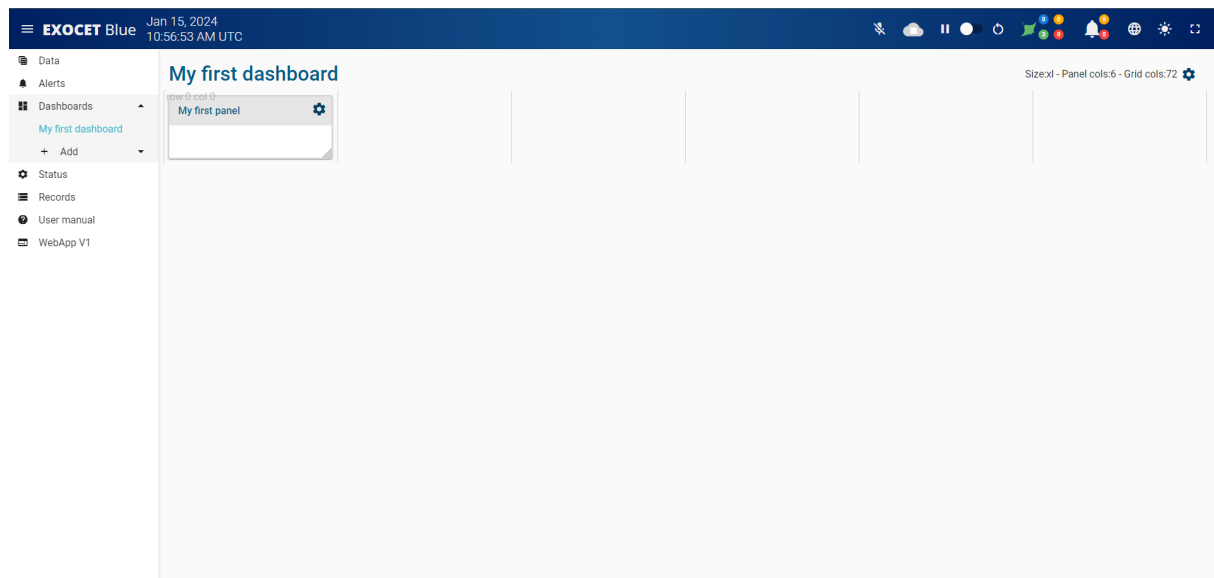


Figure 64: Panel creation

The editable area of a panel is composed by a grid. The bottom right of a panel can be clicked and dragged to resize the panel. Try it to have a 10x5 grid. As we have now more displaying space, click on the cogwheel of the panel and select *Add widget*. Multiples widgets are available. As we want to display a single numerical data, select *Simple data*.

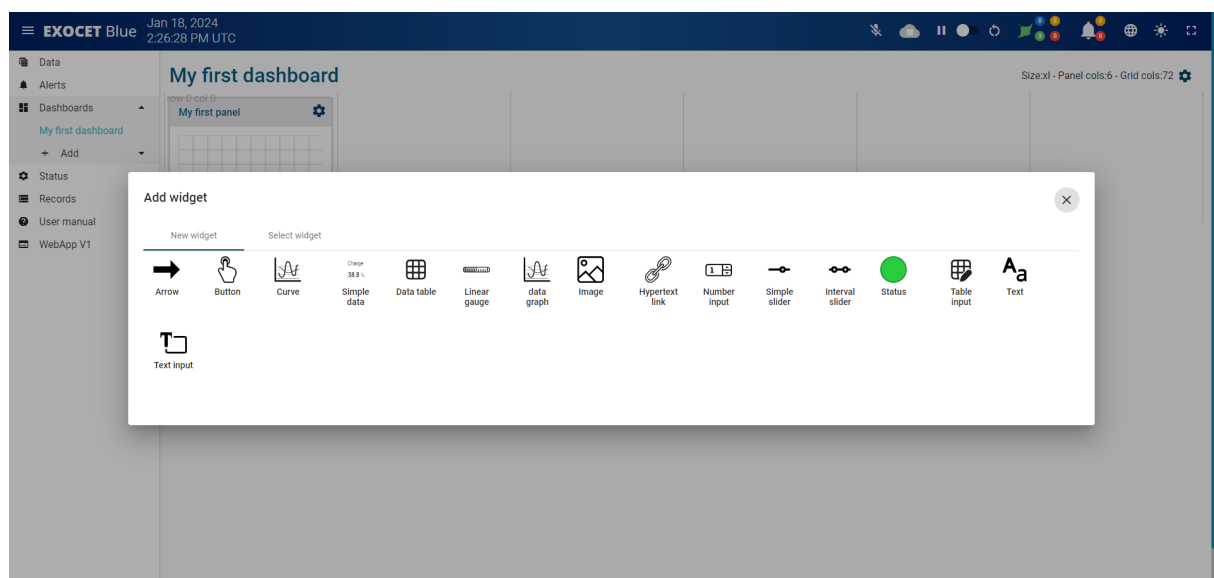


Figure 65: Available widgets

The data simple widget spawned with a tiny dimension. Resize it to match 5x3 cubes in the grid. To dis-

play the rudder angle on the widget, click on the cogwheel of the widget and click *Widget settings*. On the *Name and title* section, fill the system name with “myfirstpanel-rudder” and the title with “Rudder angle”.

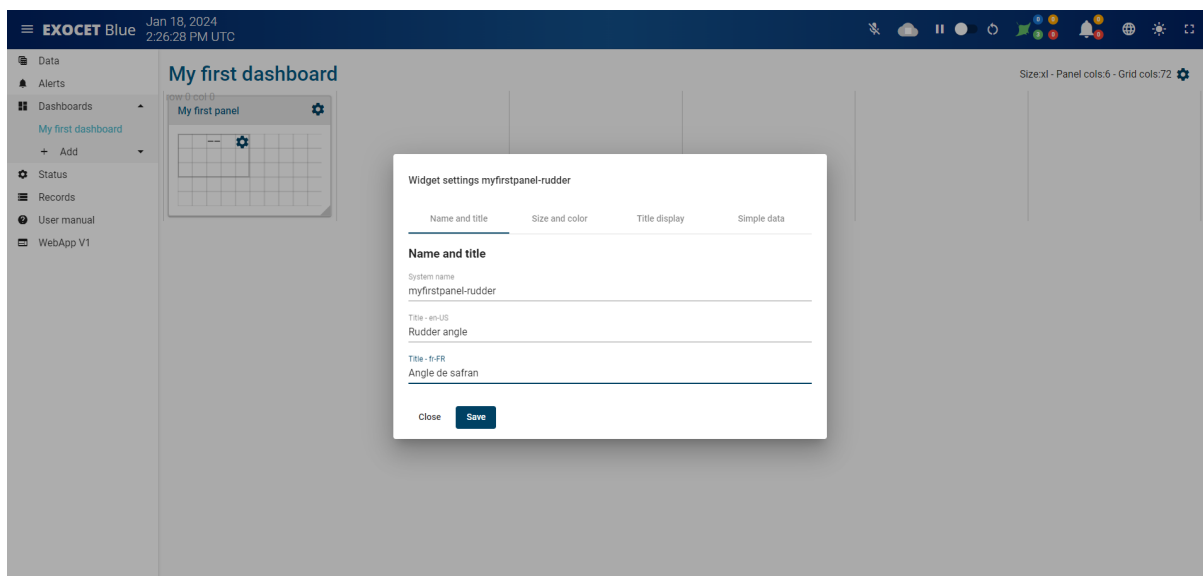


Figure 66: Set widget title

On the *Simple data* section, select the box that export the desired data (*TUT_RudderAngle_cal*). As “RudderAngle” is the only data exported by this box, it is automatically chosen. You can now save these changes.

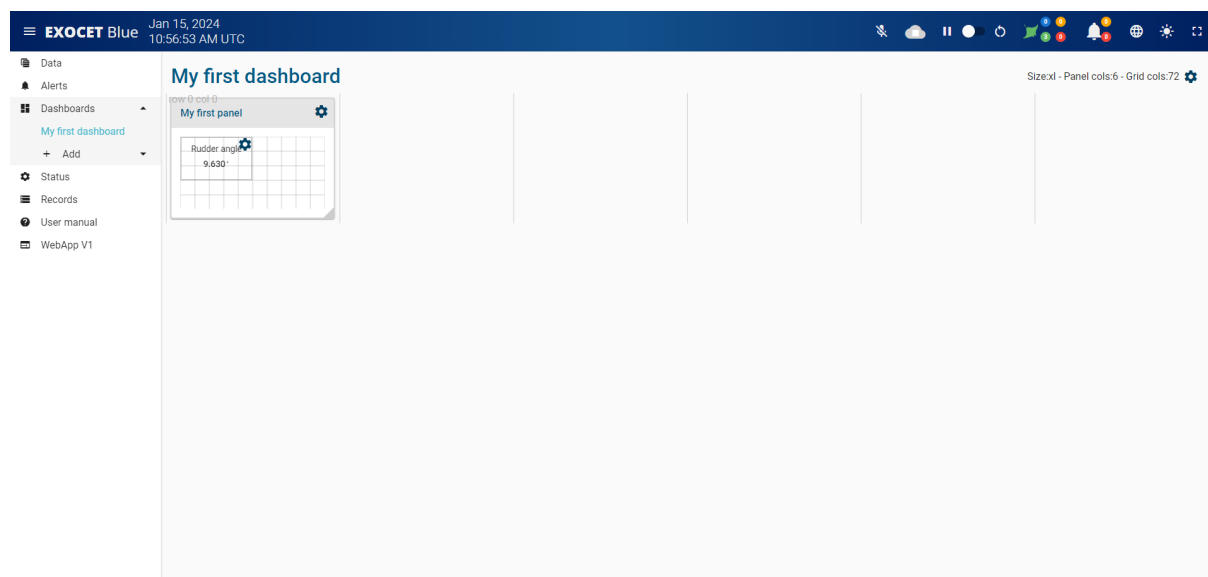


Figure 67: Widget data visualization

Congratulations, the rudder angle data is now displayed ! To close the edition mode, click on the cogwheel of the dashboard and lock dashboard. Feel free to play with the editor later.

9.7 Setting an alert

Imagine that we would place an alert on the rudder data. For this example, we can consider that the true angle of the rudder is between -45° and 45° . If we want to be notified that the sensor input is wrong, we will set an alert to alarm level if the rudder angle data is out of range. To do this, go to the web application's data page and view RudderAngle as described in the previous two subsections. Then access to the parameters of the data (black cogwheel on the same line). On the Alert section, click on "Create an alert". For this example, attribute the alert to the group 1 and set *Alarm threshold high* to 45 and *Alarm threshold low* to -45, then Save the changes by clicking on the Save button.

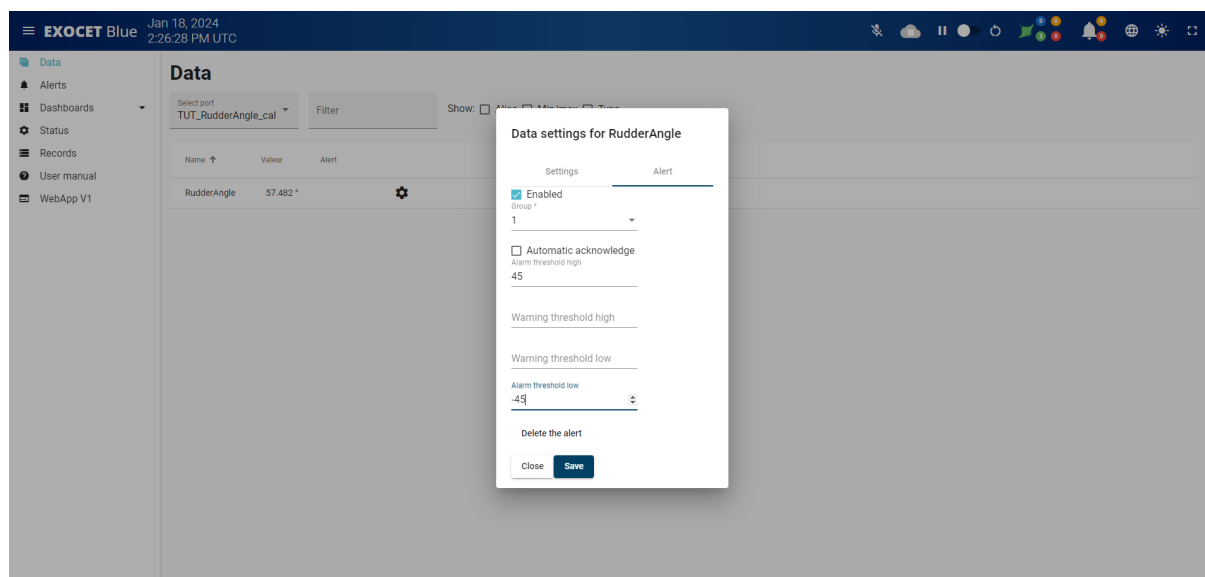


Figure 68: Set an alert

The alarm is now created, a status appeared to show the current status.

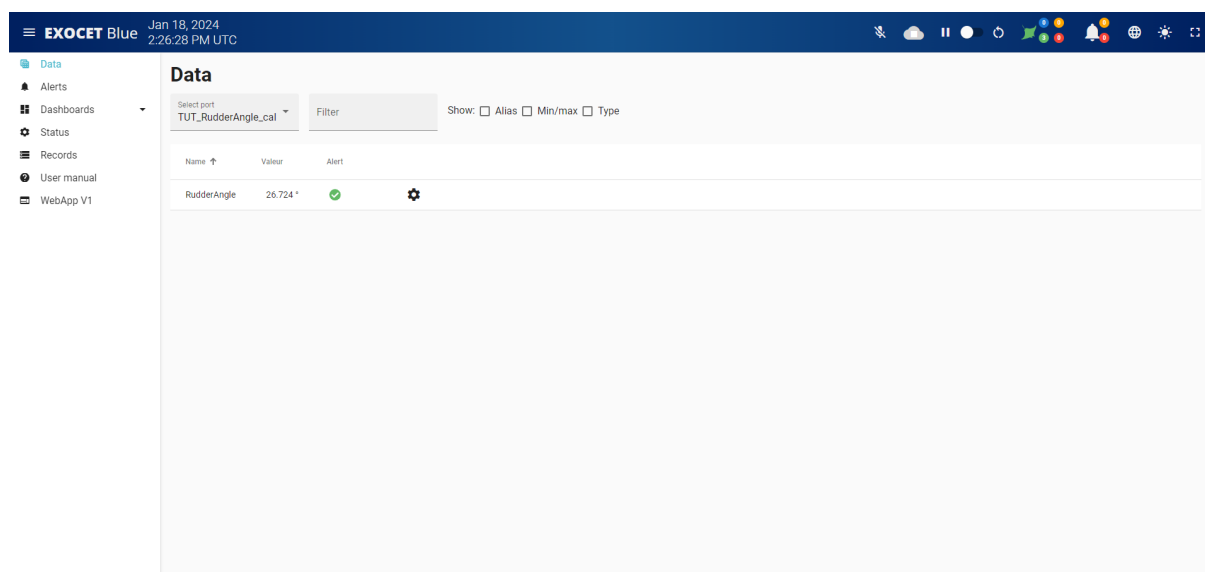


Figure 69: Alert check

9.8 Visualising the alert

Let's visualize the new alert on the dedicated page. Go to the alert page and you can see your alert showing green or red depending on the state of the RudderAngle value.

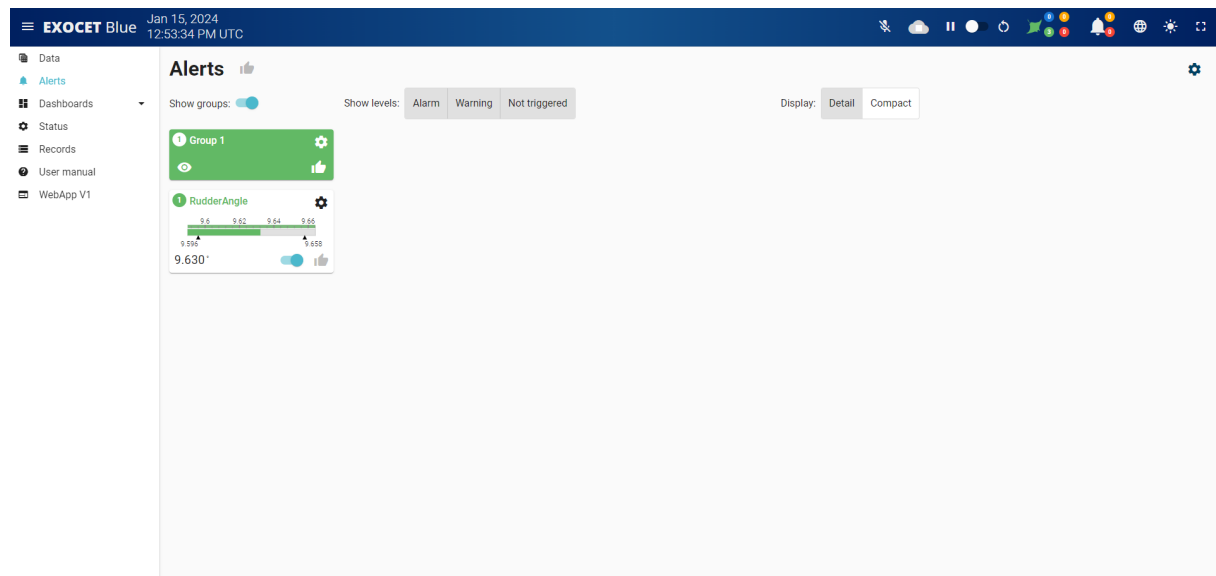


Figure 70: Web app alert visualization

9.9 Datalogging

To log the desired data in the SD card of the Exocet, we need to add a new box in the manta flow. Switch to the Manta page and add a *datalogger* box (1). Link the *TUT_RudderAngle_cal* box to the datalogger first input (2) and open the datalogger box (3). Rename the box to “TUT_DataLogger_dtl” (4). For the moment we will keep the default settings of the box. Read the description on the right to have more explanation on each parameter. You can validate the box settings by clicking on *Done* button (5) the box settings and deploy (6). The RudderAngle is now logged on the Exocet.

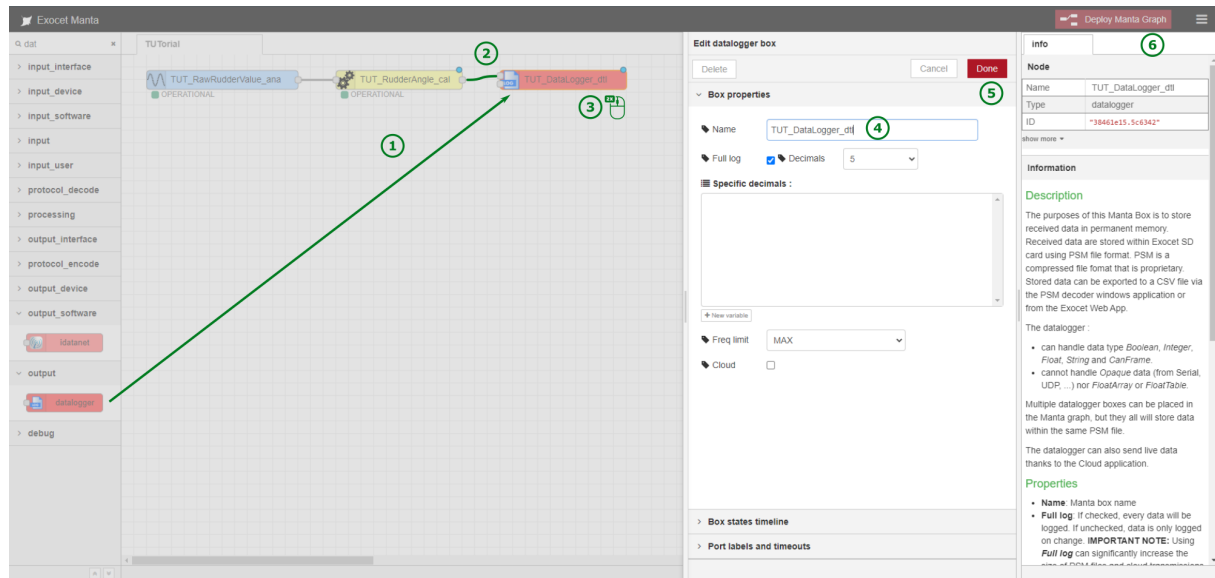


Figure 71: Datalogger setup

9.10 Sending data to the cloud

To send data to the cloud, you first need a cloud account. When this account is enabled, the Exocet can export data to the cloud while saving them in the SD card. On the Manta flow, open the *TUT_DataLogger_dtl* box (1) and tick the checkbox named *Cloud* (2). Create a new cloud configuration by clicking on the pen next to *Cloud conf* (3).

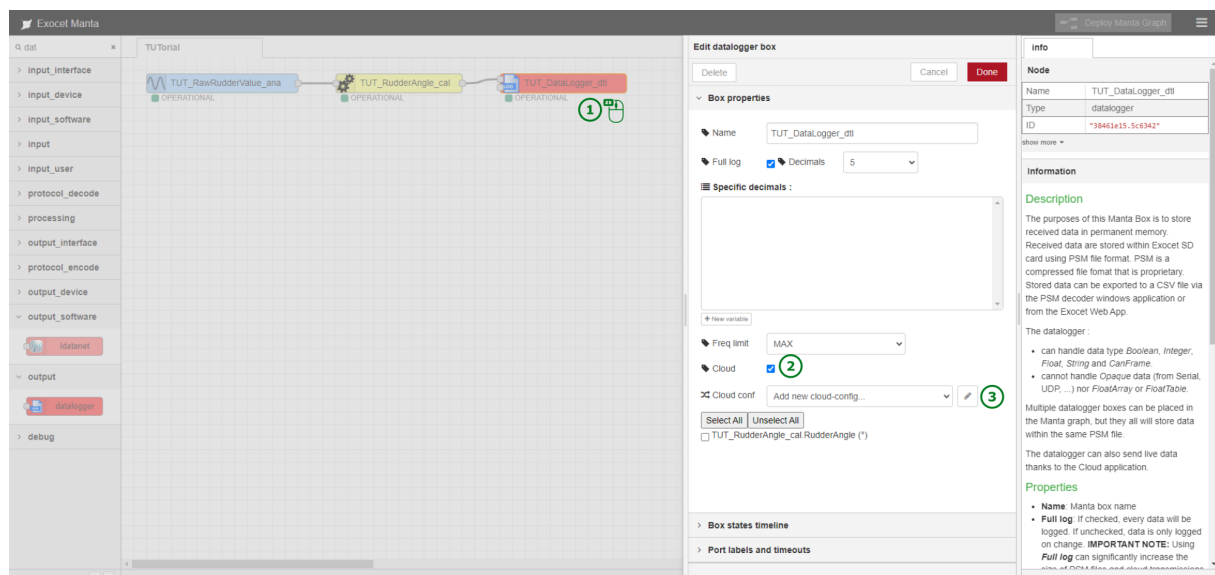


Figure 72: Add cloud config

On the cloud config menu, change the name to “MyCloudConf” (1). For this example (not recommended on boats) change the *Sending* setting to 10s (2). Be careful : the more you send data to the cloud, the more you will use internet data. Click on *Add* to add the new configuration (3).

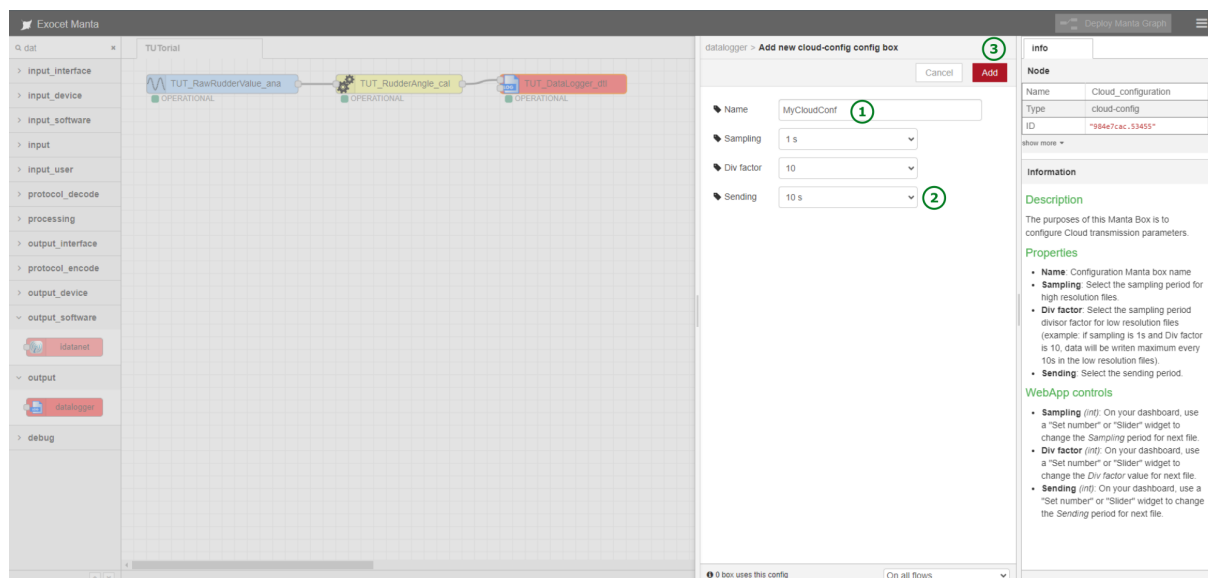


Figure 73: Change cloud config

Finally, all the available data on the *dataLogger* box are available to be sent to the cloud. Select *TUT_RudderAngle_cal.RudderAngle (°)*(1), validate those settings by clicking the *Done* button (2) and deploy the changes (3).

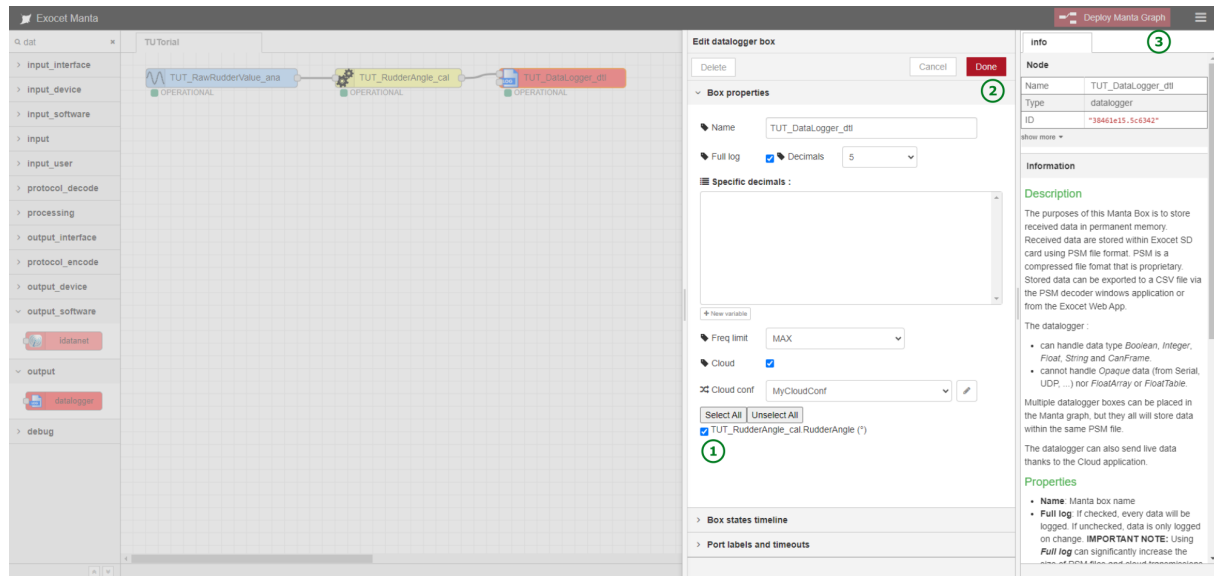


Figure 74: Valid cloud config

To enable cloud data transfer, switch to the Status page of web interface and activate *Enable transmission* on the *Exocet Cloud* section. You should see 10sec later that the *Latest transfer* label has been updated.

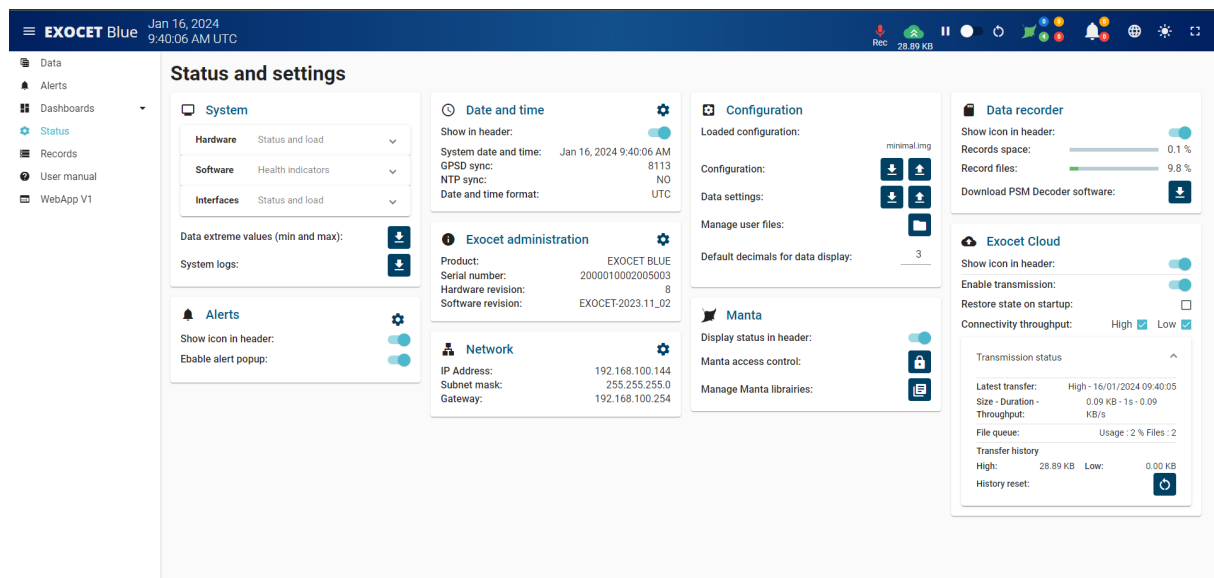


Figure 75: Cloud connectivity

To get more information on Exocet cloud, download and read the dedicated documentation : <https://exocet.cloud/documentation>.

9.11 Visualizing cloud data with Grafana

Login to your organisation at <https://exocet.cloud>. Create a dashboard and choose *Add a panel*.

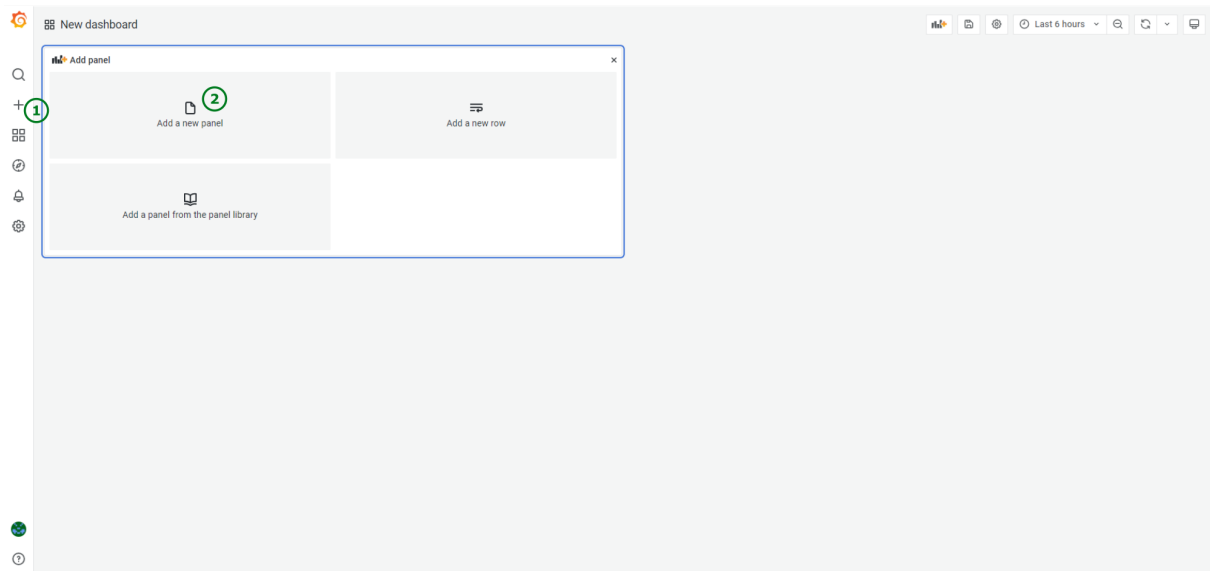
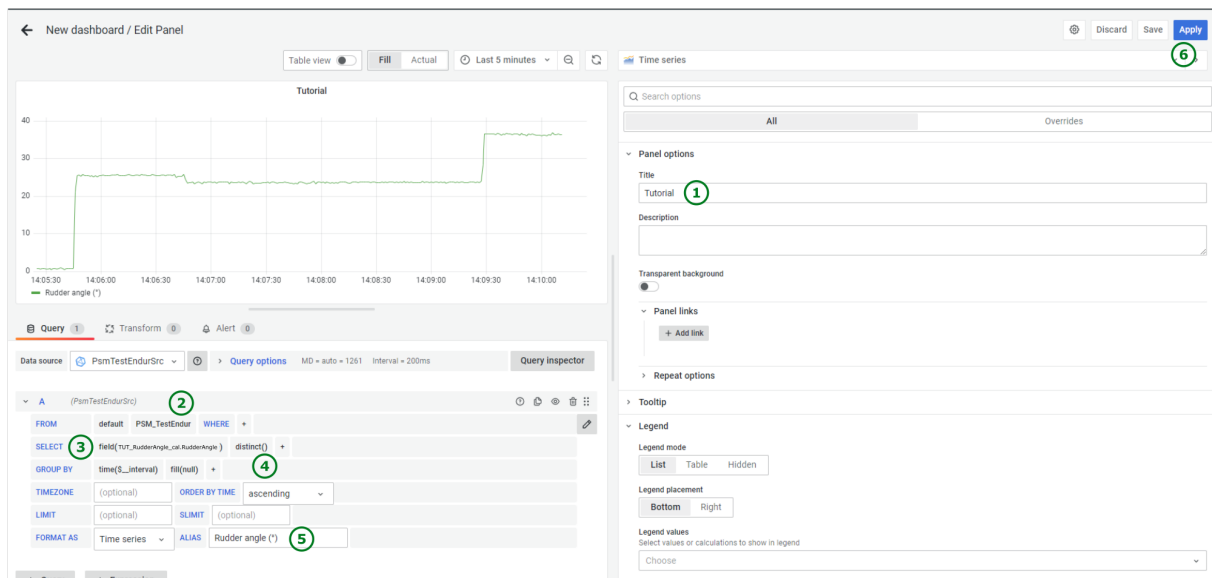
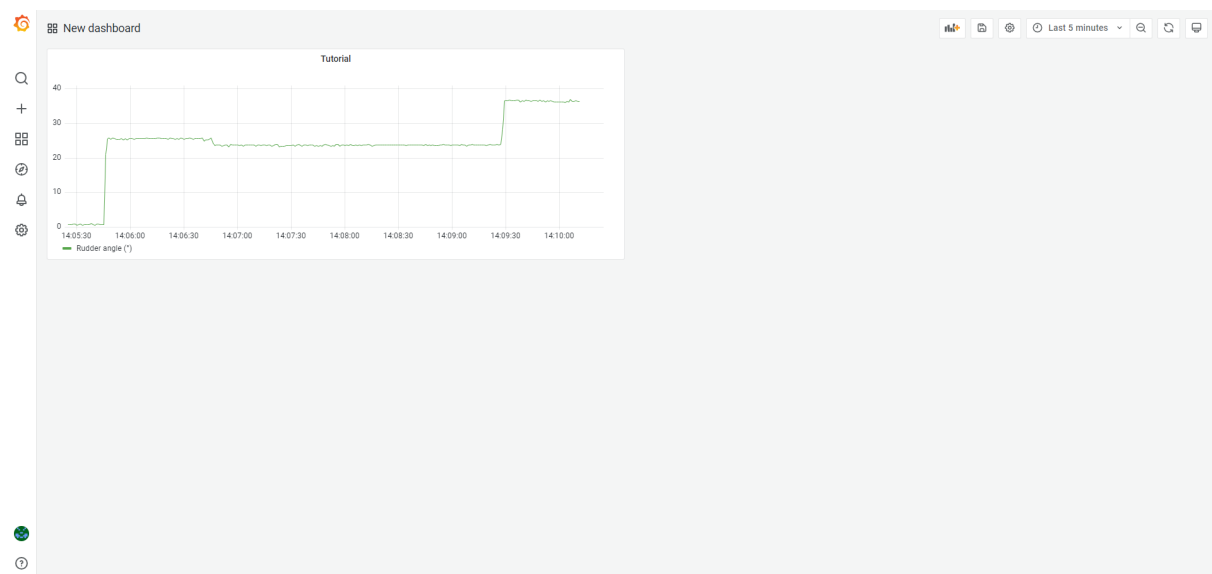


Figure 76: Grafana add panel

You are now on the edition view. On the right side, all parameters refers to the appearance of the panel. On the upper left is displayed a preview of the panel. On the lower left are listed the requests to select the variables to display. First set the title to “Tutorial” (1). Then go to the request layout and select you database by clicking on *select measurements* (2). The name will differ from this example. The next step is to select the rudder data. Click on *field(value)* and select *TUT_RudderAngle_cal.RudderAngle* (3). In order to display all the data rather than a local mean, click on the + next to *mean()* to choose *distinct()* (4). Finally, fill the *Alias* field with “Rudder angle (°)” (5). Verify in the preview that “Rudder angle (°)” is not unselected. Then click on *Apply* on the upper right (6).

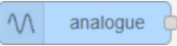



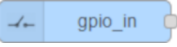



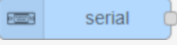







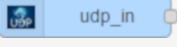



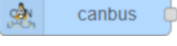



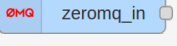



**Figure 77:** Grafana panel edition

You now have the complete data chain working !

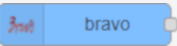







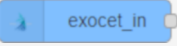



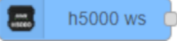



**Figure 78:** Visualization on the Cloud

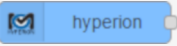

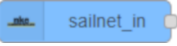

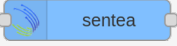

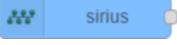

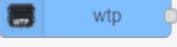

10 Manta boxes overview table

10.1 Input interfaces

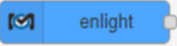

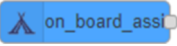

Box	Description	Exocet
 analogue	Convert an analog signal to a digital value	  
 gpio_in	Manage digital input	  
 serial	Manage RS232 or RS422 serial input	  
 tcp_in	Get data from a TCP interface	  
 udp_in	Get data from an UDP interface	  
 canbus	Get data from an CAN 2.0 bus interface	  
 zeromq_in	Get data from a ZeroMQ interface	  

10.2 Input devices

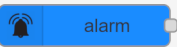





Box	Description	Exocet
 bravo	Import data from Bravo4 sailing instrument processor	  
 e telltales	Export angles for a set of a-telltales	  
 exocet_in	Import data from another Exocet Blue, Silver or Gold	  
 h5000 ws	Import B&G H5000 sailing instrument variables	  



Box	Description	Exocet
 hyperion	Read microstrains or temperatures from Hyperion, the Micron Optics optical fiber interrogator	
 sailnet_in	Imports NKE data from SailNet protocol over Ethernet	
 sentea	Read microstrains or temperatures from Sentea optical fiber interrogator	
 sirius	Import data from Pixel Sur Mer multiple sensors gateway	
 wtp	Import B&G WTP3 sailing instrument variables	

10.3 Input softwares



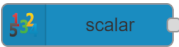





Box	Description	Exocet
 enlight	Read microstrains or temperatures from Enlight, the Micron Optics optical interrogator PC software	
 on_board_assi	Receive events from OnBoardAssistant Sailing performance software.	

10.4 Input





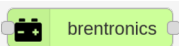

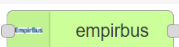

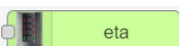

Box	Description	Exocet
 alarm	Inform if at least one alarm or warning is active	
 link	Virtual wires to import data from another flow	
 manta_status	Inform if at least one Manta box is on error state or on warning state	







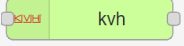

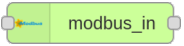

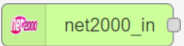

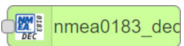

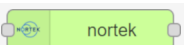

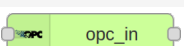

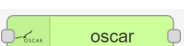

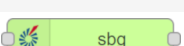

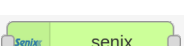





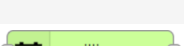

Box	Description	Exocet
 tick	Generate a periodic tick	

10.5 Input User

Box	Description	Exocet
 button	Inject a boolean value that can be updated by user from Web App dashboard	
 scalar	Inject an integer or decimal value that can be updated by user from Web App dashboard	
 string	Inject a text value that can be updated by user from Web App dashboard	
 table	Inject a table that can be updated by user from Web App dashboard	

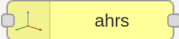



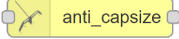




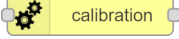



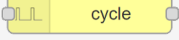


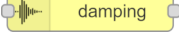



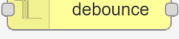



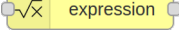



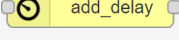



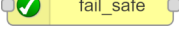



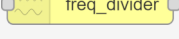



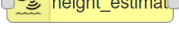

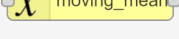



10.6 Protocol Decode





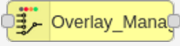

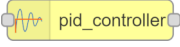





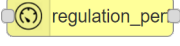

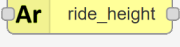





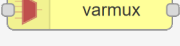



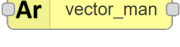

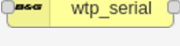



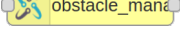

Box	Description	Exocet
 ainstein	Decode Ainstein altimeter UART & CAN protocol	
 ais	Decode AIS sentences payload	
 bretronics	Decode Brentronics battery status sent over CAN bus	
 empirbus	Decode EmpirBus user data sent over CAN bus	
 eta	Decode ETA Power Distribution Systems sent over CAN bus	

Box	Description	Exocet
 ixblue	Decode several IxBlue Inertial Navigation System protocols	
 JSON	Extract data from JSON	
 keel	Decode data of IMOCA keel management system	
 kvh	Decode some ANPP binary messages from KVH Inertial Navigation System.	
 modbus_in	Import data from remote Modbus device	
 net2000_in	Read all variables, parameters and events of a selected Net2000 source	
 nmea0183_dec	Decode NMEA0183 sentences sent over Ethernet/UDP or serial	
 nortek	Decode some Nortek DVL binary messages	
 opc_in	Import data from remote OPC-UA device	
 oscar	Decode some data from Oscar messages	
 sbg	Decode some SBG Inertial Navigation System binary messages	
 senix	Decode Senix ASCII protocol	
 victron	Decode Victron VE.Direct protocol	
 wattandsea	Decode Watt&Sea aero and hydro generator status sent via serial interface	
 williamson	Decode williamson battery status sent over CAN bus	









Box	Description	Exocet
-----	-------------	--------

















10.7 Processing

Box	Description	Exocet
 ahrs	Calibrate AHRS position and apply magnetic corrections	  
 anti_capsize	Angular piece of anti-capsize system	
 blinkManager	Manage Blink Powerkey Pro from NMEA2000	 
 calibration	Adjust data precision and convert units	  
 cycle	Execute a periodic digital signal when its input goes to True state.	 
 damping	Filtering for scalar or complex data	  
 debounce	Execute a debouncing filter on specified variables	  
 expression	Use C++ Mathematical Expression Toolkit to execute arithmetic operations, functions and processes	  
 add_delay	Delays a signal	  
 fail_safe	Manage several redundant inputs to ensure a safe output based on port timeouts	  
 freq_divider	Perform a sampling frequency division on incoming data	  
 height_estimat	compute the height above the water of points of interest	
 moving_mean	Perform statistics on selected variables	  



















Box	Description	Exocet
 stats	Perform statistics on data dictionary	  
 Overlay_Manage	Manages several pilot overlays	
 pid_controller	PID controller	
 python_function	A function block where you can write Python 2.7 code to do interesting things	  
 regulation_per	Pilot overlay platform for the implementation of multi-objective regulation performance rule	
 Ar ride_height	compute the height above the water of points of interest	
 varfilter	Extract one or more variables from an input data stream	  
 varmux	Mux several input data streams	  
 Ar vector_man	Rotate a 3D vector by providing 3 rotations angles for each axis	
 wtp_serial	Manage WTP serial module from NMEA2000	  
 obstacle_man	Manage obstacle detection	

10.8 Output Interface





Box	Description	Exocet
 gpio_out	Manage digital output	  
 serial	Send data to the specified RS232 or RS422 port	  





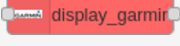



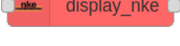



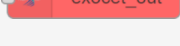









Box	Description	Exocet
 tcp_out	Send data to a TCP interface	  
 udp_out	Send data to an UDP interface	  
 canbus	Send data to a CAN 2.0 bus interface	  
 zeromq_out	Send data to a ZeroMQ interface	  

10.9 Protocole Encode

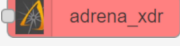







Box	Description	Exocet
 ascii_encoder	Generic ASCII protocol encoder	  
 empirbus_out	Encode EmpirBus user data to send over CAN bus	
 json_enc	Encode data to JSON	  
 modbus_out	Export data to remote Modbus device	  
 net2000_out	Send variable, parameter or event to Net2000	  

10.10 Output Device

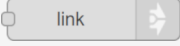



Box	Description	Exocet
 bravo	Export data to Bravo 4 Sailing Instrument Processor	  

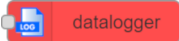

Box	Description	Exocet
 display_bandg	Send custom variables to B&G displays	  
 display_garmin	Send custom variables to Garmin displays	  
 display_nke	Send custom variables to NKE displays	  
 exocet_out	Export data to another Exocet Blue, Silver or Gold	  
 pilot_overlay_nke	Ease of use NKE pilot interface	
 pilot_overlay_bg	Ease of use B&G pilot interface	
 pilot_settings_bg	Ease of use B&G pilot interface	

10.11 Output Software

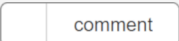



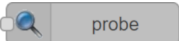



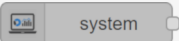



Box	Description	Exocet
 adrena_xdr	Export data to Adrena navigation software via XDR protocol	  
 idatanet	Export data to iDataNet iPhone/iPad app via Ethernet/UDP	  

10.12 Output

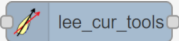

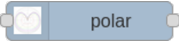

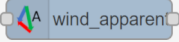

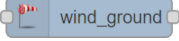

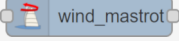

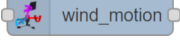

Box	Description	Exocet
 link	Virtual wires to export data to another flow	  

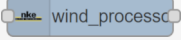

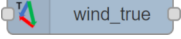

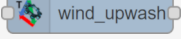

Box	Description	Exocet
 datalogger	Store data within Exocet persistent memory	

10.13 Debug

Box	Description	Exocet
 comment	A box you can use to add comments to your Manta graph	  
 probe	Display data exported by one or several boxes	  
 system	Export system usage data	  

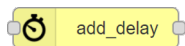
10.14 wind

Box	Description	Exocet
 lee_cur_tools	Leeway and current tools	
 polar	Calculation of VMG and other boat speed performance data	
 wind_apparent	Back-calculation of the undamped apparent wind in the horizontal plane	
 wind_ground	Calculation of the ground wind, relative to the sea floor	
 wind_mastrot	Compensate the selected Mast Head Unit angle from the mast rotation and twist	
 wind_motion	Boat motion compensation on measured wind	

Box	Description	Exocet
 wind_processc	Encode data to drive Processor HR with Pixel wind	
 wind_true	Calculate the original true wind in the horizontal plane	
 wind_upwash	Compensate the True Wind from various symmetricals errors	

11 Manta boxes detailed description

11.1 add_delay



Description

Delays data streams.

This box accumulates a history of input streams and replays it with a delay. The history length is bounded.

Properties

- **Name:** Manta box name
- **Delay (s):** Amount of time of the delay (s).
- **Outputs:** Number of data streams.
- **Dynamic delay:** Check this box to control the delay as an input.
- **WebApp:** Allow delay access from Exocet WebApp dashboards.
 - **Min / Max / Step:** Widget settings for window parameters.
- **Export parameters:** Outputs the current delay when it changes.

Inputs

- **Data i:** connect here the data streams containing variables to delay. Only one wire should be connected per Data input.

- **Delay:** connect here the delay parameter as input (s)
- **Reset:** reset dynamic and WebApp delay to initial value defined above

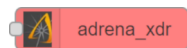
Outputs

- **Delayed data *i*:** Delayed data streams
- **Delay:** Delay parameter as output (s)

WebApp controls

- **Delay (s) (float):** If *WebApp* is selected, on your dashboard, use a “Set Number” widget to change the *Delay (s)* value.

11.2 adrena_xdr



Description

Export data to Adrena navigation software. All received data are forwarded to Adrena via the XDR protocol.

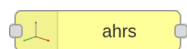
Properties

- **Name:** Manta box name
- **IP:** Enter IP address of PC running Adrena
- **Port:** Enter the Adrena Xdr port

Inputs

- Data to send to Adrena.

11.3 ahrs



Description

This Manta box performs a calibration of the AHRS (Attitude and Heading Reference System) position and, if needed, apply magnetic corrections.

Properties

- **Name:** Manta box name
- **Misroll:** Roll angle of sensor in boat frame
- **Mispitch:** Pitch angle of sensor in boat frame
- **Misyaw:** Yaw angle of sensor in boat frame
- **AHRS input mapping:** Fill Manta names corresponding to attitude, heading and magnetic deviation data
- **Magnetic variation input mapping:** Fill Manta name corresponding to magnetic variation

Inputs

AHRS raw data

Connect here one data stream containing raw sensor data (Euler angle of sensor in NED frame) * *Heel (float, deg +/-180)*: Roll angle. Sign convention (-) port side down; (+) port side up. * *Trim (float, deg +/-180)*: Pitch angle. Sign convention : (-) bow down; (+) bow up. * *Heading (float, deg)*: Heading (Compass, Magnetic or True) * *RollRate (float, deg/s)* * *PitchRate (float, deg/s)* * *YawRate (float, deg/s)* * *Deviation (float, deg +/-180)* : Magnetic deviation from boat magnetic field.

Magnetic variation

* *Magnetic variation (float, deg +/-180)* : Magnetic variation from geographic position (also named magnetic declination).

Outputs

AHRS calibrated data (Euler angle of boat in NED frame): * *Heel (float, deg +/-180)*: Roll angle. Sign convention (-) port side down; (+) port side up. * *Trim (float, deg +/-180)*: Pitch angle. Sign convention : (-) bow down; (+) bow up. * *Heading (float, deg T)*: True Hedging. * *RollRate (float, deg/s)* * *PitchRate (float, deg/s)* * *YawRate (float, deg/s)*

Misalignments (Euler angle of sensor in boat frame): * *Misroll (float, deg +/-180)*: Roll angle of sensor in boat frame * *Mispitch (float, deg +/-180)*: Pitch angle of sensor in boat frame * *Misyaw (float, deg +/-180)*: Yaw angle of sensor in boat frame

Corrections: corrections due to AHRS position calibration

Notes

Heading

$\text{Heading_Magnetic} = \text{Heading_Compass} + \text{Magnetic_Deviation}$

$\text{Heading_True} = \text{Heading_Magnetic} + \text{Magnetic_Variation}$

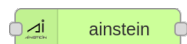
Misalignments

Most applications will only have low angles on roll and pitch misalignment. If large angles on roll and pitch are expected ($> 5^\circ$), user must consider the rotation composition order: roll, then pitch, then yaw.

NED frame

The North-East-Down (NED) coordinate system with origin defined relative to the Earth's reference ellipsoid (World Geodetic System, 1984). This is the coordinate system we refer to in our everyday life. It is usually defined as the tangent plane on the surface of the Earth moving with the boat, but with axes pointing in different directions than the body-fixed axes of the boat. For this system the x axis points towards true North, the y axis points towards East while the z axis points downwards normal to the Earth's surface.

11.4 einstein



Description

Decode Einstein altimeter UART & CAN protocol

Properties

- **Name:** Manta box name
- **Protocol:** Select UART or CAN protocol
- **CAN ID:** Only for CAN protocol, enter CAN frame ID (hexadecimal)
- **Version ID:** Only for UART protocol, enter UART protocol version ID (byte 2 of data packet)

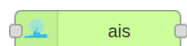
Inputs

- **data_in:** serial or can data stream

Outputs

- **data:** altitude data
- **status:** altitude data status (SNR...)

11.5 ais



Description

This box decodes and exports the AIS info from received NMEA0183 sentences. For now, only messages of type 1,2,3,5,18,19 and 24 are supported. The description of each message type is available [here](#).

Properties

- **Name:** Manta box name

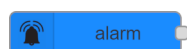
Inputs

- **NMEA0183 VDO/VDM:** connect here the VDO/VDM output of the NMEA0183 decoding box.

Outputs

- **Position report class A:** Return the full decoded class A position report (AIS message type 1,2,3)
- **Static and voyage related data:** Return the full decoded static and voyage data message (AIS message type 5)
- **Standard position report class B:** Return the full decoded standard position report class B (AIS message type 18)
- **Extended position report class B:** Return the full decoded extended position report class B (AIS message type 19)
- **Static data report:** Return the full decoded static data report (AIS message type 24)

11.6 alarm



Description

This box exports two variables to inform if at least one data is on alarm state or on warning state. These variables can be used as entry value of *GPIO output box* to ring a buzzer when a data is on alarm.

Outputs Frequency: 5 Hz

Properties

- **Name:** Manta box name
- **Group:** If a group is selected (from 1 to 12), only alarm/warning of the selected group are considered. If *Any* is selected, all alarm/warning are considered.

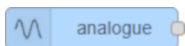
Inputs

- **Acknowledge** (boolean): acknowledge alarm/warning of the selected group on rising edge

Outputs

- **Alarm** (boolean): true if at least one data is on alarm with the selected group, else false.
- **Warning** (boolean): true if at least one data is on warning with the selected group, else false.

11.7 analog



Description

This Manta box handles analog to digital conversion. A 16-bit ADC (65536 discrete levels) converts voltage to digital data.

Properties

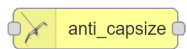
- **Name:** Manta box name
- **Port:** Select the input source
- **Range:** Adjust the input voltage range to qualify the signals going into the ADC. Note that if the range is set to the minimum this will maximize the *signal to noise* ratio.

- **Sampling rate:** The number of input samples available per seconde. Warning: the output frequency could be affected by the number of *Analogue* boxes.
- **Add TS:** Add timestamp to output data flow

Outputs

- Analog input value (*Float*, volt)

11.8 antiCapsize



Description

This box is the angular piece of anti-capsize system.

It monitors a data and takes care of triggering actuators when this data exceeds one or more trigger thresholds.

Multiple TWA Ranges can be defined to have a different trigger threshold per TWA Range for the same actuator. For each TWA Range, an actuator trigger threshold list is defined. In this list, an actuator is associated with one trigger threshold per tack. Therefore the triggered actuator can be different on port and starboard tack for the same threshold.

The TWA range and tack selection can be either automatic by using TWA data or manual by using sliders on dashboard. To be in automatic selection, sliders have to be set on “auto” position. If the wind calculation is in default (TWA input error), it is possible to use the system in degraded mode by manually selecting the TWA range and tack. When the TWA range and tack are set in automatic selection and an error occurs on the TWA, the anti-capsize box continues to operate with the last automatically determined TWA range and tack.

This box requires an activation key to be activated.

Properties

- **Name:** Manta box name.
- **Frequency** (*int*, *Hz*): Execution Frequency

Monitored data * Comparison Sign: Define if actuators are triggered when monitored data is “above or equal” thresholds or “below or equal” threshold. * **Absolute Value:** If checked, the absolute value of the monitored data is compared to thresholds.

TWA Ranges Definition * TWA Range List: Definition of the number of TWA ranges to consider and their names (max 4 TWA range). * **TWA Limit N - Initial Value** (*int, deg*): Initial value of the limit between 2 TWA range.

“TWA Limit N” = TWA limit between the “TWA Range N” and “TWA Range N+1”.

Actuator Trigger Thresholds * Number of Actuators (*int*): Define the number of actuator to manage (max 30). * **TWA Range Threshold lists:** Define the different trigger threshold to consider on each TWA rang (1 list by TWA range).

For each threshold the following parameters have to be set : * **Threshold for:** Threshold name (use to defined WebApp widget name). * **Initial Value** (*float*): Initial threshold value. * **Step Value** (*float*): Step applied when incrementing or decrementing threshold value with WebApp widget. * **Min Value** (*float*): Minimum threshold value. * **Max Value** (*float*): Maximum threshold value. * **Port Actuator:** Define the actuator to trigger on port tack. * **Starbord Actuator:** Define the actuator to trigger on starboard tack.

Input Variable Mapping: Map the name of input variables.

Inputs

Monitored Data (*float*): Data whose value is compared to thresholds. Default timeout = 1s.

TWA (*float, deg +/-180*): True Wind Angle (-: Port tack, +: Starboard tack). Default timeout = 1s.

External Control: Can be used to control WebApp Widgets from the manta graph. * **Anti-Capsizes On** (*bool*): Control “AntiCapsizesOn” widget (True: ON, False: OFF) * **TWA Range Selected** (*int*): Control “TwaRangeSelected” widget. * **Tack Selected** (*int*): Control “TackSelected” widget. * **TWA Limit N** (*int*): Control “Twa Limit N” widget.

Outputs

Status * **Status** (*int*): Anti-Capsizes status. See note at the end for the meaning. * **Status_display** (*string*): Anti-Capsizes status color * **ErrorStatus** (*int*): Anti-Capsizes error status (1 bit for each cause of error). See note at the end for the meaning. * **TwaRangeName_Status** (*int*): TWA Range status (1 variable by TWA range). See note at the end for the meaning. * **TwaRangeName_Status_display** (*string*): TWA Range status color (1 variable by TWA range)

Output frequency: 1 Hz or data change

Parameters * *TwaRangeSelected (int)*: TWA range selection slider position (0: automatique selection, 1: TWA Range 1, ...) * *TackSelected (int)*: Tack selection slider position (0: automatique selection, 1: Port, 2: Starboard) * *Twa Limit N (int)*: TWA Limit N value (1 variable by TWA Limit) * *TwaRange-Name_Threshold (float)*: Threshold value (1 variable by threshold)

Output frequency: 1 Hz or data change

Actuators Control N (1 port by actuator) * *Trigger Actuator N (bool)* : True if at least one threshold associated to the actuator N is exceeded, False otherwise.

Output frequency: Execution Frequency defined in parameter.

WebApp controls

- **AntiCapsizeOn (bool)**: On/Off bistable button.
- **TwaRangeSelected (int)**: TWA range selection slider
- **TackSelected (int)**: Tack selection slider
- **Twa Limit N (int)**: TWA limit values (1 widget by Twa Limit).
- **TwaRangeName_Threshold (float)**: Actuator threshold values (1 widget by threshold).

Notes

Anti-Capsize status

Status	Status color	Meaning
0	"dimgray"	OFF
1	"lime"	ON with TWA Range & Tack automatic switch available
3	"orange"	ON without TWA Range or Tack automatic switch (TWA input error)
7	"red"	ERROR. See ErrorStatus to know the reason.

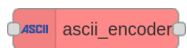
Anti-Capsize ErrorStatus

ErrorStatus Bit (little endian)	Meaning
Bit0 = 1	Monitored Data input error
Bit1 = 1	No TWA Range selected

ErrorStatus Bit (little endian)	Meaning
Bit2 = 1	No Tack selected

TWA Range status

Status	Status color	Meaning
0	“dimgray”	OFF
1	“lime”	ON in standby
3	“orange”	ON and at least one actuator triggered

11.9 ascii_out**Description**

Generic ASCII protocol encoder.

Properties

- **Name:** Manta box name
- **Dec sep:** Select decimal separator for floating point values
- **List of fields:**
 - **Nature:** define if the field is a constant, filled from a manta variable or a checksum
 - **Value:** only for constants, value of the constant
 - **Hexa:** only for constants, check if value is in hexadecimal format (ex: 0x0D0A = CRLF)
 - **Variable:** only for variables, name of the manta variable
 - **Digits:** 0 to ignore, otherwise force digits number of the whole part
 - **Decimals:** only used for float, decimals number to use
 - **Type:** only for checksums, type of the checksum
 - **Begin:** only for checksums, position of first character to consider
 - **End:** only for checksums, position from the end of last character to consider

Inputs

- **Variables** (*bool, scalar, string*). Only one wire should be connected.

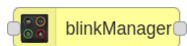
Outputs

- **Sentence**: Should be connected to a serial or UDP output box.

Notes

NMEA0183: To format an nmea0183 sentence, use constants to define header and comma separators between fields. The checksum must have the *LRC-8* type, and must begin at *1* and end at *-1*.

11.10 blinkManager



Description

Ease of use Blink Powerkey Pro interface: * Init Keypad * Decode button state from N2000 * Prepare led command to send through N2000

Properties

- **Name**: Manta box name
- **Address**: Keypad CAN source address. (Several keypad can have the same address)
- **Model**: Select the number of buttons
- **Periodic**: If checked, active periodic transmission of key states at 1Hz
- **Mult. Led cmd**: If checked, allow multiple command at LED command input using *LedColor_x* and *LedState_x* to control Led of button X. Otherwise, support historical protocol where only one LED can be addressed using *KeyCode*, *LedColor* and *LedState*.
- **Day Led level (%)**: Enter LED brightness level in day mode
- **Day backlight (%)**: Enter backlight level in day mode
- **Night Led level (%)**: Enter LED brightness level in night mode
- **Night backlight (%)**: Enter backlight level in night mode
- **Mapping**: Enter your variables names of Led commands

Inputs

- **N2k in:** Should be connected to port 61184 of a net2000 in box with type set as Powerkey
- **LED command:** Should be connected to your Led command management. Must contain:

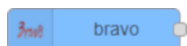
Variable	Type	Description
Key code	Int	Key code ID (from 1 to 15)
Led color	Int	0:nothing, 1:red, 2: green, 3: blue, 4: yellow, 5: cyan, 6: magenta, 7: white, 8: amber
Led state	Int	0: Off, 1: On, 2: Blink
Erase all	Bool	Optional. If true, erase all before sending the command.

- **Night mode** (*bool*): Connect a boolean to enable/disable night mode. Default is disabled.

Outputs

- **N2k Led command:** Must be connected to port 61184 of a net2000 out box with type set as Powerkey
- **Button state:** Should be connected to your button state management.
- **N2k Led brightness:** Must be connected to port 61184 of a net2000 out box with type set as Powerkey
- **N2k backlight:** Must be connected to port 61184 of a net2000 out box with type set as Powerkey
- **N2k enable periodic:** Must be connected to port 61184 of a net2000 out box with type set as Powerkey
- **N2k period:** Must be connected to port 61184 of a net2000 out box with type set as Powerkey

11.11 bravo_in



Description

This Manta box imports data from Bravo 4 Sailing Instrument Processor.

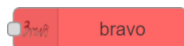
Properties

- **Name:** Manta box name
- **Network:** Select network configuration. Press the *pencil* icon to create a new one. Enter the IP address and port number used to communicate with Bravo processor. Once created, it can be shared with a *Bravo out* box.
- **Bravo variables to import:** Press *New data* button. Fill the Bravo data name and choose the decimals to be used by Bravo to format the imported data.

Outputs

- Specified Bravo variables to import

11.12 bravo_out



Description

This Manta box exports data to Bravo 4 Sailing Instrument Processor.

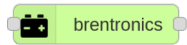
Properties

- **Name:** Manta box name
- **Network:** Select network configuration. Press the *pencil* icon to create a new one. Enter the IP address and port number used to communicate with Bravo processor. Once created, it can be shared with a *Bravo in* box.
- **Bravo variables to export:** Press *New data* button. Fill the Exocet data name, the Bravo data name and choose the decimals to be used by Exocet to format the exported data. If decimal parameter is set to “Auto”, then the number of decimals used to display the variable within Exocet Web App will be applied.

Inputs

- Exocet variables to export to Bravo. (Multiple wires can be connected).

11.13 bretronics



Description

Decode Brentronics battery status sent to CAN bus

Properties

- **Name:** Manta box name
- **Reference:** Select the battery reference
- **Serial Number:** Enter the battery serial number as returned on CAN bus. In case of trouble, first check this parameter.

Inputs

- **CAN_input:** Must be connected to a canbus in box configured at 250 kbps

Outputs

- **CAN_output:** Must be connected to a canbus out box configured at 250 kbps
- **Battery_Status:** Return the specified battery status:

Name	Units	Type
Firmware_version		float
Firmware_patch		integer
Firmware_build		integer
AtRate	Amperes	float
AtRateTimeToFull	Minutes	integer
AtRateTimeToEmpty	Minutes	integer
AtRateOK		bool
Temperature	Degrees Celcius	float
Voltage	Volts	float

Name	Units	Type
Current	Amperes	float
AvgCurrent	Amperes	float
MaxError	Percent	integer
RelStateOfCharge	Percent	integer
AbsStateOfCharge	Percent	integer
RemainingCapacity	Amperes Hours	float
FullChargeCapacity	Amperes Hours	float
RunTimeToEmpty	Minutes	integer
AvgTimeToEmpty	Minutes	integer
AvgTimeToFull	Minutes	integer
ChargingCurrent	Amperes	float
ChargingVoltage	Volts	float
Flags	<i>see below</i>	bit field
CycleCount		integer
DesignCapacity	Amperes Hours	float
DesignVoltage	Volts	float
ManufactureDate		integer
SerialNumber		integer

- **Flags** consolidates following Battery Status bit flags:

- *Flags_OverchargedAlarm*
- *Flags_TerminateChargeAlarm*
- *Flags_OverTemperatureAlarm*
- *Flags_TerminateDischargeAlarm*
- *Flags_RemainingCapacityAlarm*
- *Flags_RemainingTimeAlarm*
- *Flags_Initialization*
- *Flags_ChargeFETTest*
- *Flags_FullyCharged*
- *Flags_FullyDischarged*

- *Flags_ErrorCodes*: 0 ok, 1 busy, 2 reserved command, 3, unsupported command, 4 access denied, 5, overflow/underflow, 6 bad size, 7 unknown error

Notes

Bren-Tronics battery should not be connected on a CAN bus where PGN 61184 is already using because conflicts can occurred.

11.14 button



Description

User controlled box.

This Manta box is controlled from Exocet Web App dashboards via the “Button” widget. The boolean output of the box corresponds to the “Button” widget state.

Properties

- **Name**: Manta box name
- **Save**: If selected, the button state is saved to be restored when Exocet is rebooted or Manta graph deployed.
- **Output Name**: Exported data name. Manta box name is used if not defined.
- **Period**: to send button state periodically. State is also sent at any user change. If **None** is selected, only value changes are sent.
- **Initial Value**: initial data value.
- **Ext control**: If selected, add inputs to externally control the button value.

Inputs

- No Input connector, the input comes from Web App.
- **Bool**: If *Ext control* is selected, a boolean value can be received to set the data value.
- **Reset**: If *Ext control* is selected, a boolean rising edge can be received to return to the *Initial Value*.

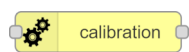
Outputs

- **Output Name:** Boolean data value

WebApp controls

- **Output Name** (*bool*): On your dashboard, use a “Button” widget to switch the value.

11.15 calibration



Description

The purposes of this Manta Box is:

- To convert measurement units (for example voltage, mV/V) to a physical unit (degrees C,...)
- To adjust the precision and accuracy of measurement equipments or to reduce bias in an instrument.
- To mux several data streams into one by selecting requested variables

Variables to be calibrated need to be added into the list.

Properties

- **Name:** Manta box name
- **Pass through:** export all non calibrated variables.
- **Dynamic calibration:** activates external parameters inputs. Not available for linearization type.
- **WebApp:** allow parameters access from Exocet WebApp dashboards
- **Min / Max / Step:** (WebApp option only) scalar widget settings
- **Export parameters:** add an output connector to export current parameters (only for slope/offset)
- **Apply to all:** If checked, apply the same process to all variables of the input data stream.
- **Unit:** (Apply to all option only) Modify or remove the unit of calibrated variables.
- **Prefix:** (Apply to all option only) Use this field to prefix all output variables.
- **Suffix:** (Apply to all option only) Use this field to suffix all output variables.
- **Replace/by:** (Apply to all option only) Use these fields to search and replace a substring into all output variables.

- **List of variables to calibrate** (maximum 250):
 - **Variable:** name of the variable to calibrate
 - **Type:** calibration type. See below for calibration types definition.
 - **Output Name:** Change the name of the calibrated variable.
 - **Unit:** Modify or remove the unit of the calibrated variable.
 - **Other properties depends on the calibration type.**

Inputs

- **Data:** Connect here the data stream containing variables to calibrate.
- **slope/offset:** Connect scalar to configure calibration parameters.
- **table:** Connect table to configure calibration LUT.
- **Reset:** Connect here a boolean to reset dynamic and WebApp parameters to initial values defined above

Outputs

- **Calibrated_data:** A list of calibrated data, and non calibrated data if *Pass through* option is selected.
- **Parameters:** For Slope/Offset type, the list of current slope/offset

WebApp controls

- **Output Name(slope) (float):** If *WebApp* is selected, on your dashboard, use a “Set Number” widget to change *slope* value.
- **Output Name(offset) (float):** If *WebApp* is selected, on your dashboard, use a “Set Number” widget to change *offset* value.
- **Output Name(X1) (float):** If *WebApp* is selected, on your dashboard, use a “Set Number” widget to change *X1* value.
- **Output Name(Y1) (float):** If *WebApp* is selected, on your dashboard, use a “Set Number” widget to change *Y1* value.
- **Output Name(X2) (float):** If *WebApp* is selected, on your dashboard, use a “Set Number” widget to change *X2* value.
- **Output Name(Y2) (float):** If *WebApp* is selected, on your dashboard, use a “Set Number” widget to change *Y2* value.
- **Output Name(LUT) (floatTable):** If *WebApp* is selected, on your dashboard, use a “Table input” widget to change *LUT* value.

Notes

Calibration types:

Slope/Offset: This type of calibration requires two parameters: the slope coefficient and the offset coefficient. Returns $y=ax+b$. Those coefficients are default values in case of dynamic calibration.

Linearization: This type of calibration requires 2 points to define a straight line. It calculates a slope and an offset, and then applies a standard slope/offset calibration.

1D-LUT: This type of calibration requires n points to define a transfert function (curve). These points are used as default values in case of dynamic calibration. Several outputs can be defined (see 1D-LUT format below). Interpolation could be linear, polynomial, trigonometric or spline. Be careful with some interpolation method, especially polynomial, which can create some unexpected behavior depending on defined points. Please, use the “Curves” widget in the WebApp to check the interpolation result.

2D-LUT: This type of calibration requires 2 input variables, n points for each, to define a transfert function (plane). These points are used as default values in case of dynamic calibration. Interpolation is linear.

1D-LUT format:

The transfert function (curve) is defined by n pairs of x,y coordinates. The first column define x (must be in ascending order), the second define y . The column separator could be space or tab. Example:

```
0.0 1.0
10.0 1.5
20.0 1.5
```

If extrapolation is selected, first and last segment are used outside the LUT limits (to use very carefully if interpolation method is not linear). Otherwise, limits are defined as follows :

```
x < x[0] => y = y[0]
x > x[n] => y = y[n]
```

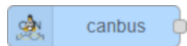
The 1D-LUT can define several curves for several outputs (number of y need to match with the *Nb output* field):

```
0.0 1.0 2.0
10.0 1.5 2.0
20.0 1.5 3.0
```

2D-LUT format:

The transfert function (plane) is defined by 2 input variables in a x,y,z coordinates. The first column define x (*Variable*, must be in ascending order), the first line define y (*Variable 2*, must be in ascending order), the intersections define results z. The column separator could be space or tab. The first line first column will be ignored but must be a valid number. Example:

```
0 45.0 90.0
0.0 1.0 1.0
10.0 1.0 1.5
20.0 1.5 2.0
```

11.16 canbus_in**Description**

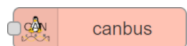
Get data from the specified CAN Bus.

Properties

- **Name:** Manta box name
- **CAN port:** CAN port configuration box. Press the pencil icon to create a new one. Fill the CAN port parameters, then press the *Add* button. Once created, it can be shared with other CAN in our out boxes.
- **Add TS:** Add timestamp to output data flow
- **Filter:** If selected, export only can frames with *CAN ID* header.
- **CAN IDs to filter out:** Only if *Filter* is selected, list of CAN frame IDs to export (hexadecimal)

Outputs

- **can** (*CanFrame*): received CAN data

11.17 canbus_out

Description

Send data to the specified CAN Bus.

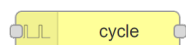
Properties

- **Name:** Manta box name
- **CAN port:** CAN port configuration box. Press the pencil icon to create a new one. Fill the CAN port parameters, then press the *Add* button. Once created, it can be shared with other CAN in our out boxes.

Inputs

- **can** (*CanFrame*): data to send

11.18 cycle



Description

Execute a periodic digital signal when its input goes to True state.

Properties

- **Name:** Manta box name
- **Output Name:** Exported data name. Manta box name is used if not defined.
- **Period:** Signal periodicity in seconds (from 1s to 15s)
- **Duty cycle:** Signal duty cycle (from 0% to 100%). Default step is 10%, use *high resolution* option to decrease step to 1%.
- **High resolution:** Permit to increase duty cycle resolution from 10% to 1%.
- **Max repetition:** Number of cycles to execute when input is triggered (from 0 to 10, 0 = infinite)
- **Complete:** If true, execute max repetition cycles even if input return to False state. If false, stop to execute the digital signal when input return to False state.
- **Ext control:** If selected, add inputs to externally control the signal parameters.

Inputs

- **Trigger:** A boolean to activate the periodic digital signal
- **Ext control:** If *Ext control* is selected, scalar can be received to set the signal period, duty cycle and max repetition

Name	Description
Period	Signal periodicity in seconds (from 1 to 15, step 1)
DutyCyle	Signal duty cycle (from 0 to 100, step 10 or 1)
MaxRepetition	Number of cycles to execute when input in triggered (from 0 to 10, 0 = infinite)

- **Reset:** If *Ext control* is selected, a boolean rising edge can be received to return to the initial signal parameters

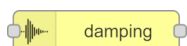
Outputs:

- **Signal:** The periodic digital signal when input in triggered. Data frequency is 0.1Hz in idle state, 10Hz or 100Hz when input is triggered depending on *high resolution* option.
- **Parameters:** Export the current signal parameters

WebApp controls

- **Period** (*int*): On your dashboard, use a “Set number” or “Slider” widget to change the *Period* value.
- **DutyCycle** (*int*): On your dashboard, use a “Set number” or “Slider” widget to change the *Duty cycle* value.
- **MaxRepetition** (*int*): On your dashboard, use a “Set number” or “Slider” widget to change the *Max repetition* value.

11.19 damping



Description

The purposes of this Manta Box is to apply a digital infinite-impulse-response (IIR) filter on all or selected variables.

Properties

- **Name:** Manta box name
- **Dynamic:** If selected, add inputs to control cutoff from the Manta graph
- **Export params:** add an output connector to export current parameters
- **WebApp:** Allow cutoff parameters access from Exocet WebApp dashboards
- **Min / Max / Step:** (*WebApp* option only) widget settings for cutoff parameters
- **Design:** Design of damping to apply (see *Notes* below)
- **Attenuation (dB):** (*Chebyshev* option only) Minimum stopband attenuation in dB
- **Freq div:** Divisor factor to reduce input sampling frequency to output sampling frequency (see *Notes* below)
- **Apply to all:** If checked, apply the same process to all variables of the input data stream
- **Pass through:** (Unchecked *Apply to all* option only) Export all non damped variables
- **Auto suffix:** (*Apply to all* option only) Check to add an automatic suffix to all variables, else use the input field to define a custom suffix
- **Type:** Type of damping to apply (see *Notes* below)
- **Setting type:** Type of the damping setting: *Frequency (Hz)* or *Time Constant (s)*. Both are linked with $\text{Time Constant} = 1 / (2 \cdot \pi \cdot \text{Cutoff Frequency})$. The cutoff frequency corresponds to a 3dB attenuation.
- **Setting value:** Value of the damping setting relative to the selected *Setting type* (see *Notes* below)
- **List of variables to damp** (maximum 250):
 - **Var type:** Type of variable : scalar or complex (see *Notes* below)
 - **Var_in:** Name of the scalar variable to damp
 - **Var_out:** Name of the damped scalar variable
 - **Mod_in:** Name of the modulus part of complex variable to damp
 - **Mod_out:** Name of the modulus part of damped complex variable
 - **Arg_in:** Name of the argument part of complex variable to damp (rad or °, default is °)
 - **Arg_out:** Name of the argument part of damped complex variable

Inputs

- **Data:** Connect here the data stream containing variables to damp. Only one wire must be connected on Data input.
- **cutoff:** Connect here one (or two) scalar to configure the “Setting Value(s)” dynamically. Unit depends on “Setting Type” selection.
- **Reset:** Connect here a boolean to reset dynamic and WebApp parameters to initial values defined above.

Outputs

- **Damped_data:** A list of damped data, and non damped data if *Pass through* option is selected.
- **Parameters:** The list of current “Setting Values”

WebApp controls

- **Output Name(Hz or s) (float):** If *WebApp* is selected, on your dashboard, use a “Set Number” widget to change the *Settings Value*.

Notes

Designs

Name	Delay for lowpass filter		Roll-off
		Max cut-off frequency	
Butterworth order 1	Equal to <i>Time Constant</i>	No limit	20 dB / decade
Butterworth order 2	Equal to 1.4 <i>Time Constant</i>	50 Hz	40 dB / decade
Chebyshev type II order 2	Less than 1.4 <i>Time Constant</i>	50 Hz	Slightly more than 40 dB / decade until <i>Attenuation</i> is reached

Types

- **Low Pass:** passes signals with frequency lower than selected cutoff frequency, and attenuates

signals with higher frequencies.

- **High Pass:** passes signals with frequency higher than selected cutoff frequency, and attenuates signals with lower frequencies.
- **Band Pass:** passes signals with frequency within range, and attenuates signals with outside frequencies.
- **Band stop:** attenuates signals with frequency within range, and passes signals with outside frequencies.

Frequency Divider

Used to re-sample output signal relative to input signal sampling frequency. If high frequencies have been attenuated, there is no reason to keep the input sampling frequency. Minimum output sampling frequency should be twice the max cutoff frequency (i.e. 2Hz in case of cutoff at 1Hz).

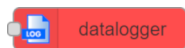
Setting value

For 2nd order filters, an internal sampling is done at 100Hz. The filter cutoff frequency must be lower than the internal sampling frequency divided by two. Hence the cutoff frequency cannot be beyond 50Hz. For the Butterworth order 1 filter, there is no such limitation.

Complex variable type

The box converts a vector defined by *Modulus* and *Argument* from polar coordinate to a complex number in a form $z=a+ib$. Damping is applied to the complex form and then convert back to a vector in polar coordinate vector.

11.20 datalogger



Description

The purposes of this Manta Box is to store received data in permanent memory. Received data are stored within Exocet SD card using PSM file format. PSM is a compressed file format that is proprietary. Stored data can be exported to a CSV file via the PSM decoder windows application or from the Exocet Web App.

The datalogger :

- can handle data type *Boolean*, *Integer*, *Float*, *String* and *CanFrame*.
- cannot handle *Opaque* data (from Serial, UDP, ...) nor *FloatArray* or *FloatTable*.

Multiple datalogger boxes can be placed in the Manta graph, but they all will store data within the same PSM file.

The datalogger can also send live data thanks to the Cloud application.

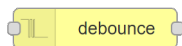
Properties

- **Name:** Manta box name
- **Full log:** If checked, every data will be logged. If unchecked, data is only logged on change. **IMPORTANT NOTE:** Using **Full log** can significantly increase the size of PSM files and cloud transmissions if data to log have a high frequency. It is recommended to use this option only during setup/debug phase and not in production.
- **Decimals:** Number of decimals used to determine if a float data has changed, and to print this data into CSV file when *Exocet* float format is selected in PSM Decoder.
- **Specific decimals:** List of variables that need a specific *Decimals* parameter
- **Freq limit:** Limit logged data frequency
- **Cloud:** Enable Cloud transmission
- **Cloud conf:** Cloud configuration box. Press the pencil icon to create. Only one configuration is allowed, it is shared with other datalogger boxes.
- **Variables:** Select variables to send to the Cloud application. Note that the Manta graph must have been deployed a first time after adding the *Datalogger* box and connecting it, to populate the variables list.

Inputs

- **DATA:** Connect here data to log. Multiple wires can be connected.
- **Enable** (*bool*): Connect a boolean to enable/disable the datalog. Default is enable.

11.21 debounce



Description

Execute a debouncing filter on specified variables. Rising and Falling edge could be separately configured.

Properties

- **Name:** Manta box name
- **List of variables to process:**
 - **Variable:** name of the input variable to process
 - **Output Name:** Exported data name. Input variable name is used if not defined
 - **Rising:** Delay in milliseconds to filter out any glitches on input rising edge (step is 50ms)
 - **Falling:** Delay in milliseconds to filter out any glitches on input falling edge (step is 50ms)
 - **Invert:** Invert the output digital signal

Input variable type can be boolean, a scalar or a string. Scalar (Integer or Float) with value equal or greather than 1 is considered as true, else it is considered as false. String containing either “Yes” or “On” or “Enabled” or “1” is considered as true, else it is considered as false.

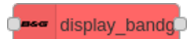
Inputs

- **Data :** connect here the data stream containing variables to process

Outputs:

- **Output:** processed data

11.22 display_bandg



Description

This box is used to send custom variables to B&G displays.

Properties

- **Name:** Manta box name.
- **Emission Period:** Data emission period (in seconds).
- **List of variables to display:**
 - **Variable:** Name of the input variable to display

- **Channel:** Select channel to use
 - **Long caption:** Data label used in display menu (max 16 characters). If empty, the name of the input variable truncated at 16 characters is used.
 - **Short caption:** Data label used on display screen (max 8 characters). If empty, the name of the input variable truncated at 8 characters is used.
 - **Decimals:** Select decimal number to display
- **Known User Vars on CAN bus:**
 - **CAN Port:** Select CAN bus port of net2000 out box
 - **User Vars:** Channel - Long caption - Short caption - Decimals

Inputs

Variables to display (*Integer or Decimal*). Multiple wires can be connected.

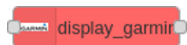
Outputs

- **PGN 130833:** Must be connected to port 130833 of a net2000 out box with type set as User var
- **PGN 130824_x:** Must be connected to port 130824 of a net2000 out box with type set as User var

Notes

Use input timeout to display dashes.

11.23 display_garmin



Description

This box is used to send custom variables to Garmin displays.

Properties

- **Name:** Manta box name.
- **Emission Period:** Data emission period (in seconds).
- **List of variables to display:**
 - **Variable:** Name of the input variable to display
 - **Channel number:** Select channel to use
 - **Format:** Select desired format
 - **Display name:** Data label used on display screen (max 10 characters). If empty, the name of the input variable truncated at 10 characters is used.
 - **Timeout:** Enter desired timeout in second. Default is 0 (no timeout).

Inputs

Variables to display (*Boolean, Scalar or Time*). Multiple wires can be connected.

Outputs

- **126720_X:** Must be connected to port 126720 of a net2000 out box with type set as User var

Notes

Use input timeout to send dashes.

Formats:

Format	Description
Invalid	Display dashes [--]
Unsigned	Display scalar as unsigned integer [0:4294967295]

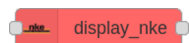
Format	Description
Signed	Display scalar as signed integer [-2147483648:2147483647]
Float	Display scalar as float [-XXXX.XXXX:XXXX.XXXX]
Boolean	Display boolean with translated string True/False or 1/0
Unsigned angle	Display scalar as unsigned angle, with degree symbol [0°:360°]
Signed angle	Display scalar as signed angle, with degree symbol [-180°:+180°]
3 digits angle	Display scalar as unsigned angle padded with leading zeros to 3 digits, with degree symbol [000°,001°:360°]

Format	Description
P/S angle	Display scalar as signed angle, with degree symbol and P/S indicator, P sent as negative [180°P:+180°S]
Temperature	Display scalar as float (convert from Celcius), with degree symbol [-12°,35°]
Length	Display scalar as float (convert from meters)
Depth	Display scalar as float (convert from meters)
Speed	Display scalar as float (convert from nautical knots)
Pressure	Display scalar as float (convert from millibar)
Volume	Display scalar as float (convert from meters cubed)

Format	Description
Flow	Display scalar as float (convert from meters cubed per hour)
Voltage	Display scalar as float (convert from Volts)
Percent	Display scalar as unsigned integer, with percent symbol
Port / Starboard	Display boolean with translated string Port/Starboard (Port is True)
On / Off	Display boolean with translated string On/Off (On is True)
Yes / No	Display boolean with translated string Yes/No (Yes is True)

Format	Description
Time	Display scalar as time HH:MM:SS (convert from Seconds) [00:00:00 to 99:59:59]

11.24 display_nke



Description

This box is used to send custom variables to NKE displays.

Properties

- **Name:** Manta box name.
- **Emission Period:** Data emission period (in seconds).
- **List of variables to display:**
 - **Variable:** Name of the input variable to display.
 - **Channel number:** 1 to 8 or 1 to 30 depending on NKE display.
 - **Format:** Display format. See below for the format definition.
 - **Display name:** Data label of the display (max 10 characters). If empty, the name of the input variable truncated at 10 characters is used.
 - **Overwrite unit:** Checkbox to overwrite the unit label of the display.
 - * Unchecked: The unit name of the input variable truncated at 7 characters is used.
 - * Checked: The *display unit* field is used.
 - **Display unit:** Unit label of the display (max 7 characters).

Inputs

Variable to display (*Integer or Decimal*). Multiple wires can be connected.

Outputs

- **data** (*Opaque*): NKE proprietary NMEA0183 sentence. The output connector must be connected to a serial or udp_out box.

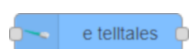
Notes

Output Frequency: It is defined by the emission period.

Formats:

Format label	Data value
XX_XX	00.00 to 99.99
XXXX	0000 to 9999
XX	00 to 99
XXX_X	000.0 to 999.9
XDIR	-179° to 180°
XANG	000° to 359°
X_XXX	0.000 to 9.999
XCHRONO	00:00 to 99:59
XX_X	00.0 to 99.9
SXXX	-999 to 999
SXX_X	-99.9 to 99.9
SX_XX	-9.99 to 9.99
XXPC	00% to 99%
XREND	0000 to 9999
XLONG	0L to 999L

11.25 e_telltales



Description

This box exports the angles for a set of e-telltales.

Properties

- **Name:** Manta box name
- **List of E-telltales to export:** define a list of e-telltales to export. For each, define:
- **ID:** identifier. *nmea0183_decode* Manta box with “PENON” prefix can be used to list all the e-telltales identifier available.
- **Ax:** Constant to calibrate angle data. Update this value on e-telltales manufacturer recommendation only, else keep default value.
- **B:** Constant to calibrate angle data. Update this value on e-telltales manufacturer recommendation only, else keep default value.

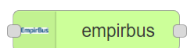
Inputs

- **nmea0183 data:** nmea0183 stream from serial or UDP.

Outputs

- One output per listed e-telltale. Each output exports the e-telltale angle.

11.26 empirbus



Description

Decode EmpirBus user data sent over CAN bus.

Properties

- **Name:** Manta box name
- **Instance:** unique instance to distinguish / route the data
- **Data Model:** Data model of Application Specific Data

- **List of data to import:**

- **Field:** EmpirBus user data field
- **Output name:** name of the imported variable
- **Unit:** unit of the imported variable

Inputs

- **PGN 65280:** Must be connected to port 65280 of a net2000_in box with type set as Energy

Outputs

- A list of decoded data according to the list defined in properties

Note

Application example: Below is an example of EmpirBus output configuration. It uses Data Model 2 to export a Dimmer value on User Data Word 1, and an AC fan status on User Data Bit 5:8.

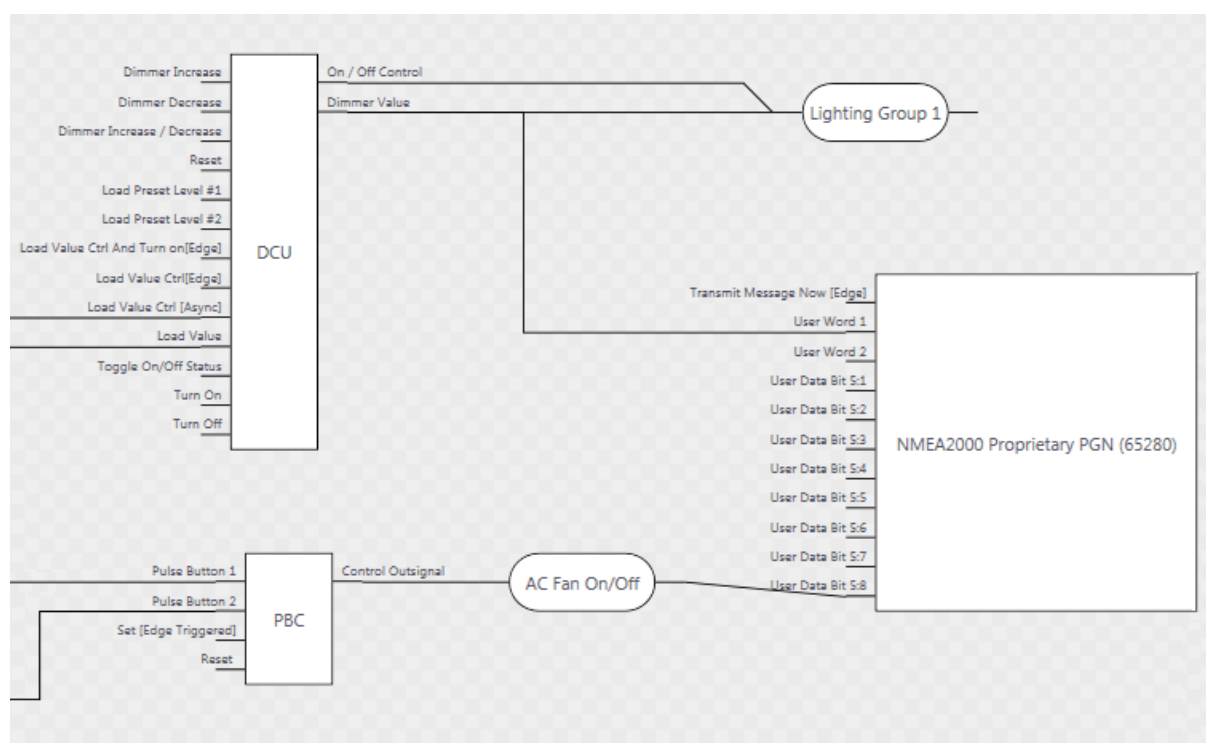


Figure 79

11.27 empirbus_out



Description

Encode EmpirBus user data to send over CAN bus.

Properties

- **Name:** Manta box name
- **Instance:** unique instance to distinguish / route the data
- **Data Model:** Data model of Application Specific Data
- **List of data to export:**
 - **Variable:** name of the variable to export
 - **Field:** EmpirBus user data field

Inputs

- Variables to export (*bool, int or float*). Multiple wires can be connected.

Outputs

- **PGN 65280:** Must be connected to port 65280 of a net2000_out box with type set as Energy

Note

Application example: Below is an example of EmpirBus input configuration. It uses Data Model 2 to import a Load value on User Data Word 1, and a boolean command on User Data Bit 5:8.

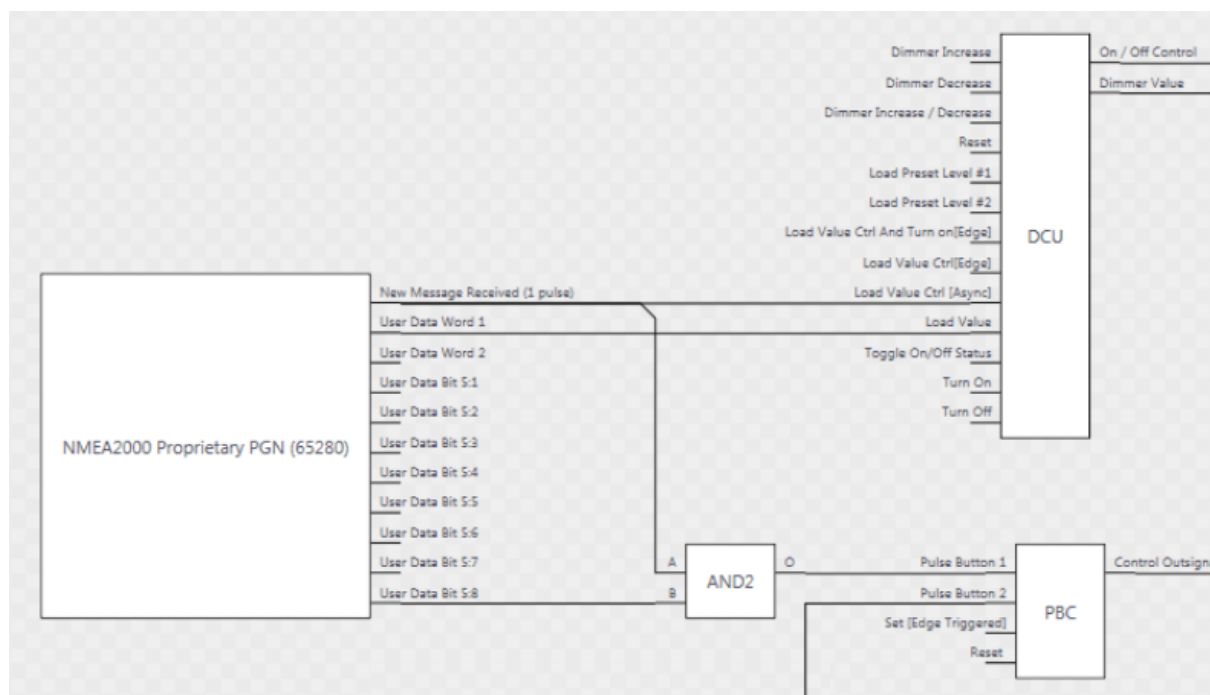
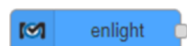


Figure 80

11.28 enlight



Description

Export variables such as microstrains or temperatures from Enlight, the Micron Optics optical interrogator PC software. It is compatible with all the Optics interrogators.

Notes:

- The grouping notion of Enlighth is not taken into account for now
- The variables created from the *Sensor* panel are exported without unit
- The *remote command interface* parameter must be checked within *settings* panel

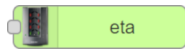
Properties

- **Name:** Manta box name
- **IP:** IP address of PC running **Enlight**

Outputs

- A set of variables such as microstrains or temperatures according to the configuration file

11.29 eta



Description

Decode ETA Power distribution Systems sent over CAN bus.

Notes:

- ETA system must first be configured using PowerPlex software to periodically transmit load outputs current and analog inputs voltage over CAN bus.
- Current value feature is not available for the 1 Ampere load outputs. Only the 8A and 25A load outputs are able to transmit it. To worlaround this, a constant current value can be associated with a memory flag ID. If the 1A current output is linked with a this memory flag, then the constant value will be exported when 1A output is switched on.

Properties

- **Name:** Manta box name
- **ID:** ETA system ID.
- **List of ETA data to export:** define a list of data to export. For each data, select:
- **Type:**
 - **Current Out 1A:** to send a constant value when output is switched on.
 - **Current Out 8A:** to get current transmitted by a 8 Amperes load output.
 - **Current Out 25A:** to get current transmitted by a 25 Amperes load output.
 - **Analog In 0-10V:** to get analogic input voltage.
- **ID:** load output ID or analogic input ID or memory flag ID
- **out name:** can be used to define a name for the output. If not set, a default name will be defined.
- **Value:** constant value to export when 1 A output is switched on. Else 0 ampere value is exported.

Inputs

- **canin:** can bus input to receive data from ETA system.

Outputs

- A list of *analogID* or *currentID*, respectively in Volts or Amperes according to ETA box configuration.

11.30 exocet_in



Description

This Manta box imports data from another Exocet Blue, Silver or Gold.

Properties

- **Name:** Manta box name
- **Local port:** Enter here the local input port to receive data. You need to customize the default value if you use several Exocet input boxes.
- **Remote IP:** Enter here the emitter IP address.
- **Remote port:** Enter here the emitter output port. You need to customize the default value if there is several Exocet output boxes.
- **WebApp:** Allow connexion parameters access from Exocet WebApp dashboards

Outputs

- Imported Exocet variables

Notes

Maximum number of variables is limited to 1024.

11.31 exocet_out



Description

This Manta box exports data to another Exocet Blue, Silver or Gold.

Properties

- **Name:** Manta box name
- **Local port:** Enter here the local output port to send data. You need to customize the default value if you use several Exocet output boxes.
- **Remote IP:** Enter here the receiver IP address.
- **Remote port:** Enter here the receiver input port. You need to customize the default value if there is several Exocet input boxes.
- **Frequency:** Select the data emission frequency. It should be enough to ensure a good performance, but slow enough to not overload the receiver.
- **WebApp:** Allow connexion parameters access from Exocet WebApp dashboards

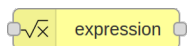
Inputs

- Exocet variables to export. (Multiple wires can be connected).

Notes

Maximum number of variables is limited to 1024.

11.32 expression



Description

This Manta Box uses the C++ Mathematical Expression Toolkit (ExprTk) to support fundamental arithmetic operations, functions and processes. Each expression must finish by a semicolon “;” (see *Notes* below for examples and more details), including loops that finish by a closing bracket “}”.

Properties

- **Name:** Manta box name
- **Expression:** write here the expression to process. See *Notes* below for help.
- **List of symbols (maximum 100):**
 - **Symbol:** symbol used in the expression
 - **Nature:** define if the symbol is a constant, an input/output variable, or an internal variable
 - **Value:** only for constants, value of the constant (scalar only)
 - **VarName:** name of the variable used in the Manta graph
 - **Direction:** input for variables to look for on input data stream, output for variables to export
 - **VarType:** type of the variable used in the Manta graph (note that inside this box Integer and Boolean are managed as floating point number, and CanFrame are managed as string)
 - **Init:** Check box to initialize input variable, and fill field with initial value to use. Otherwise, initialization of variable is checked before expression execution.
 - **Unit:** only for output direction, unit applied to the exported variable
 - **Type:** only for internal variables, select if the variable is a Float, a String or a Vector (of Float)
 - **Init:** only for internal variables, fill field with initial value to use. For a Vector, all elements are set to this value.
 - **Size:** only for internal variables of type Vector, size of the vector.

Inputs

- **Data:** connect here the data stream containing variables required to process the expression.

Outputs

- **Result:** export output variables defined in the list of symbols, depending on assignments of defined expression.

Notes

Example

```
var bsp := AWS * sin(AWA);  
if (bsp > 35.0, debug := 'Too fast',
```

```
if (bsp < 1.0, debug := 'Too slow',
    debug := 'cool');
```

With: * AWS, AWA: symbols of input variables of type double * bsp: symbol of a local variable * debug: symbol of an output variable of type string

Reserved symbols Due to the internal processing, the following words are reserved and so cannot be used as symbol: *abs, acos, acosh, and, asin, asinh, atan, atanh, atan2, avg, break, case, ceil, clamp, continue, cos, cosh, cot, csc, default, deg2grad, deg2rad, equal, erf, erfc, exp, expm1, false, floor, for, frac, grad2deg, hypot, iclamp, if, else, ilike, in, inrange, like, log, log10, log2, logn, log1p, mand, max, min, mod, mor, mul, ncdf, nand, nor, not, not_equal, null, or, pow, rad2deg, rep, repeat, return, root, round, roundn, sec, sgn, shl, shr, sin, sinc, sinh, sqrt, sum, swap, switch, tan, tanh, true, trunc, until, var, while, xnor, xor, &, |*

Disable output A special internal variable named “*rep*” can be used to disable data export on output. If *rep:=true*, data is exported (default), if *rep:=false* no data is exported.

```
if (bsp < 1.0, rep := false, rep := true);
```

Arithmetic & Assignment Operators

OPERATOR	DEFINITION
+	Addition between x and y. (eg: x + y)
-	Subtraction between x and y. (eg: x - y)
*	Multiplication between x and y. (eg: x * y)
/	Division between x and y. (eg: x / y)
%	Modulus of x with respect to y. (eg: x % y)
^	x to the power of y. (eg: x ^ y)
:=	Assign the value of x to y. (eg: y := x)
+=	Increment x by the value of the expression on the right hand side. (eg: x += y)
-=	Decrement x by the value of the expression on the right hand side. (eg: x -= y)

OPERATOR	DEFINITION
*=	Assign the multiplication of x by the value of the expression on the righthand side to x. (eg: x *= y) /= Assign the division of x by the value of the expression on the right-hand side to x. (eg: x /= y) %= Assign x modulo the value of the expression on the right hand side to x. (eg: x %= y)

Equalities & Inequalities

OPERATOR	DEFINITION
== or =	True only if x is strictly equal to y. (eg: x == y)
<> or !=	True only if x does not equal y. (eg: x <> y or x != y)
<	True only if x is less than y. (eg: x < y)
<=	True only if x is less than or equal to y. (eg: x <= y)
>	True only if x is greater than y. (eg: x > y)
>=	True only if x greater than or equal to y. (eg: x >= y)

Boolean Operations

OPERATOR	DEFINITION
true	True state or any value other than zero (typically 1).
false	False state, value of exactly zero.
and	Logical AND, True only if x and y are both true. (eg: x and y)
mand	Multi-input logical AND, True only if all inputs are true. Left to right short-circuiting of expressions. (eg: mand(x > y, z < w, u or v, w and x))
mor	Multi-input logical OR, True if at least one of the inputs are true. Left to right short-circuiting of expressions. (eg: mor(x > y, z < w, u or v, w and x))
nand	Logical NAND, True only if either x or y is false. (eg: x nand y)
nor	Logical NOR, True only if the result of x or y is false (eg: x nor y)

OPERATOR	DEFINITION
not	Logical NOT, Negate the logical sense of the input. (eg: $\text{not}(x \text{ and } y) == x \text{ nand } y$)
or	Logical OR, True if either x or y is true. (eg: $x \text{ or } y$)
xor	Logical XOR, True only if the logical states of x and y differ. (eg: $x \text{ xor } y$)
xnor	Logical XNOR, True iff the biconditional of x and y is satisfied. (eg: $x \text{ xnor } y$)

General Purpose Functions

FUNCTION	DEFINITION
abs	Absolute value of x. (eg: $\text{abs}(x)$)
avg	Average of all the inputs. (eg: $\text{avg}(x,y,z) == (x + y + z) / 3$)
ceil	Smallest integer that is greater than or equal to x. (eg: $\text{ceil}(x)$)
clamp	Clamp x in range between r0 and r1, where $r0 < r1$. (eg: $\text{clamp}(r0,x,r1)$)
equal	Equality test between x and y using normalised epsilon
erf	Error function of x. (eg: $\text{erf}(x)$)
erfc	Complimentary error function of x. (eg: $\text{erfc}(x)$)
exp	e to the power of x. (eg: $\text{exp}(x)$)
expm1	e to the power of x minus 1, where x is very small. (eg: $\text{expm1}(x)$)
floor	Largest integer that is less than or equal to x. (eg: $\text{floor}(x)$)
frac	Fractional portion of x. (eg: $\text{frac}(x)$)
hypot	Hypotenuse of x and y (eg: $\text{hypot}(x,y) = \sqrt{x^2 + y^2}$)
iclamp	Inverse-clamp x outside of the range r0 and r1. Where $r0 < r1$. If x is within the range it will snap to the closest bound. (eg: $\text{iclamp}(r0,x,r1)$)
inrange	In-range returns “true” when x is within the range r0 and r1. Where $r0 < r1$. (eg: $\text{inrange}(r0,x,r1)$)
log	Natural logarithm of x. (eg: $\text{log}(x)$)
log10	Base 10 logarithm of x. (eg: $\text{log10}(x)$)
log1p	Natural logarithm of 1 + x, where x is very small. (eg: $\text{log1p}(x)$)

FUNCTION	DEFINITION
log2	Base 2 logarithm of x. (eg: log2(x))
logn	Base N logarithm of x. where n is a positive integer. (eg: logn(x,8))
max	Largest value of all the inputs. (eg: max(x,y,z))
min	Smallest value of all the inputs. (eg: min(x,y,z))
mul	Product of all the inputs (eg: mul(x,y,z) == (x * y * z))
ncdf	Normal cumulative distribution function. (eg: ncdf(x))
nequal	Not-equal test between x and y using normalised epsilon
pow	x to the power of y. (eg: pow(x,y) == x ^ y)
root	Nth-Root of x. where n is a positive integer. (eg: root(x,3) == x^(1/3))
round	Round x to the nearest integer. (eg: round(x))
roundn	Round x to n decimal places (eg: roundn(x,3)) where n > 0 and is an integer. (eg: roundn(1.23456,2) == 1.23)
sgn	Sign of x, -1 where x < 0, +1 where x > 0, else zero. (eg: sgn(x))
sqrt	Square root of x, where x >= 0. (eg: sqrt(x))
sum	Sum of all the inputs. (eg: sum(x,y,z) == (x + y + z))
swap, <=>	Swap the values of the variables x and y and return the current value of y. (eg: swap(x,y) or x <=> y)
trunc	Integer portion of x. (eg: trunc(x))

Trigonometry Functions

FUNCTION	DEFINITION
acos	Arc cosine of x expressed in radians. Interval [-1,+1] (eg: acos(x))
acosh	Inverse hyperbolic cosine of x expressed in radians. (eg: acosh(x))
asin	Arc sine of x expressed in radians. Interval [-1,+1] (eg: asin(x))
asinh	Inverse hyperbolic sine of x expressed in radians. (eg: asinh(x))
atan	Arc tangent of x expressed in radians. Interval -1,+1
atan2	Arc tangent of (x / y) expressed in radians. [-pi,+pi] (eg: atan2(x,y))

FUNCTION	DEFINITION
atanh	Inverse hyperbolic tangent of x expressed in radians. (eg: atanh(x))
cos	Cosine of x. (eg: cos(x))
cosh	Hyperbolic cosine of x. (eg: cosh(x))
cot	Cotangent of x. (eg: cot(x))
csc	Cosecant of x. (eg: csc(x))
sec	Secant of x. (eg: sec(x))
sin	Sine of x. (eg: sin(x))
sinc	Sine cardinal of x. (eg: sinc(x))
sinh	Hyperbolic sine of x. (eg: sinh(x))
tan	Tangent of x. (eg: tan(x))
tanh	Hyperbolic tangent of x. (eg: tanh(x))
deg2rad	Convert x from degrees to radians. (eg: deg2rad(x))
deg2grad	Convert x from degrees to gradians. (eg: deg2grad(x))
rad2deg	Convert x from radians to degrees. (eg: rad2deg(x))
grad2deg	Convert x from gradians to degrees. (eg: grad2deg(x))

String Processing

FUNCTION	DEFINITION
=, ==, !=, <>, <=, >=, <, >	All common equality/inequality operators are applicable to strings and are applied in a case sensitive manner. In the following example x, y and z are of type string. (eg: (x <= "AbC") and (y != "aaa") and (z == x))
in	True only if x is a substring of y. (eg: "abc" in "abcdef")
like	True only if the string x matches the pattern y in a case sensitive manner. Available wildcard characters are "*" and "?" denoting zero or more and zero or one matches respectively. (eg: "abcdef" like "a?cf") <i>ilike</i> True only if the string x matches the pattern y in a case insensitive manner. Available wildcard characters are "*" and "?" denoting zero or more and zero or one matches respectively. (eg: 'ABCDEF' ilike 'a?cf')

FUNCTION	DEFINITION
[r0:r1]	The closed interval [r0,r1] of the specified string, where $r0 < r1$. (eg: with $x := \text{"abcdef"}$, $x[1:3] == \text{"bcd"}$, $x[:2] == \text{"abc"}$, $x[3:] == \text{"def"}$, $x[:] == \text{"abcdef"}$) Note: In case of fractional components, truncation will be performed. (eg: $1.67 \rightarrow 1$)
:=	Assign the value of x to y. (eg: $y := x$, $y := \text{"abc"}$)
+	Concatenation of x and y. (eg: $x + y$, $x + \text{"abc"}$)
+=	Add to x the value of the expression on the right hand side. (eg: $x += y$, $x += \text{"abc"}$)
<=>	Swap the values of x and y. Where x and y are mutable strings. (eg: $x <=> y$)
[]	The string size operator returns the size of the string being actioned. (eg: $x[] == 3$, $\text{"abc"}[] == 3$)

Control Structures

STRUCTURE	DEFINITION
if	If x is true then return y else return z. (eg: $\text{if } (x, y, z)$)
if-else	The if-else statement. Subject to the condition branch, the statement will return either the value of the consequent or the alternative branch. (eg: $\text{if } (x > y) z; \text{ else } w;$)
switch	The first true case condition that is encountered will determine the result of the switch. If none of the case conditions hold true, the default action is assumed as the final return value. (eg: $\text{switch } \{ \text{case } x > 5 : y; \text{ case } x < 3 : z; \text{ default} : w; \}$);
while	The structure will repeatedly evaluate the internal statements "while" the condition is true. (eg: $\text{while } (x > 0) \{ y += 2; x -= 1; \}$;
repeat/until	The structure will repeatedly evaluate the internal statements "until" the condition is true. (eg: $\text{repeat } y += 2; x -= 1; \text{ until } x > 0$)

STRUCTURE	DEFINITION
for	The structure will repeatedly evaluate the internal statements while the condition is true. On each loop iteration, an “incrementing” expression is evaluated. The conditional is mandatory whereas the initialiser and incrementing expressions are optional. (eg: for (var x := 0; x < n; x += 1) { y += 2; });)
break	Break terminates the execution of the nearest enclosed loop. (eg: while (y < 10) { if (x < 5) y += 2; else break; });)
continue	Continue results in the remaining portion of the nearest enclosing loop body to be skipped. (eg: for (var i := 0; i < 10; i += 1) { if (i < 5) continue; j += i; });)
return	Return immediately from within the current expression.
?	Ternary conditional statement, similar to that of the above denoted if-statement. (eg: x ? y : z)
[*]	Evaluate any consequent for which its case statement is true. (eg: [*] { case x > 2 : y += z; case x > 5 : y += w; });)

Comments

```
// This is a single line comment
### This is a single line comment
/* This is a multi-line comment */
```

Vector Processing Initialisation of a vector can be done in several ways:

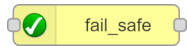
```
var x[3]; # Vector of size 3, all elements are zero
var x[3] := {}; # Vector of size 3, all elements are zero
var x[3] := [ 42 ]; # Vector of size 3, all elements are 42
var x[3] := { 1, 2, 3 }; # Vector of size 3, elements are 1, 2, 3
```

Access to vector elements is done using the following syntax:

```
x[0] := 42; # Assign 42 to the first element of the vector
y := x[0]; # Assign the first element of the vector to y
```

More Other examples are available on ExprTk homepage, and more details are available on Readme page.

11.33 fail_safe



Description

This Manta box uses several redundant sources and select one to give a safe and robust output. The box can filter out incorrect values. A source can be diagnosed as “invalid” based on several optional criteria: - the validity status of the source is false - the value has been frozen for too long - incorrect values are too frequent (NaN, zero, out-of-range, outliers, etc.) In auto position (see *WebApp controls*), the valid source with highest priority is selected, otherwise, you can manually select the source.

Properties

- **Name:** Manta box name
- **Inputs:** Number of sources
- **Ext control:** If selected, add an input to externally control the input to use from the Manta graph.
- **Switch mode:** If selected, FailSafe fonctionnality is disabled and the box behaves like a simple switch.
- **Save:** If selected, current enabled inputs and used input are saved to be restored when Exocet application is restarted.
- **Debug:** If selected, add an output to export useful variables for debug.
- **Failure filter ratio threshold (%):** It defines the maximum allowable ratio of incorrect values before considering a source as invalid. A value is considered incorrect depending on the checks selected (NaN, zero, empty and/or outlier). The threshold should be set to a level that minimizes false positives (declaring a valid source as invalid) while still capturing an unacceptable ratio of incorrect values. Recommendation: $threshold [\%] = (p + 5 \cdot \sqrt{p \cdot (1-p) / (2 \cdot N - 1)}) \times 100$, with p the probability of an incorrect data and N the **Failure filter coefficient**.
- **Failure filter coefficient:** Filtering coefficient N applied to make statistics for each source to determine the ratio of incorrect values. It determines how quickly the ratio is adjusted over time. A higher coefficient results in slower adjustments and makes the ratio determined over

a long period. It is comparable to the number of values used to compute the ratio of incorrect values. Recommendation: $N > (1-p) / 2p$, with p the probability of an incorrect data.

- **Data freeze time (ms):** Time (in milliseconds) to consider a source as frozen (see *Check Freeze*).
- **Reject incomplete data streams:** If selected, check if inputs contain all output data.
- **List of variables to rename** (optional):
 - **Var_in:** Enter an input variable name (must be unique).
 - **Var_out:** Enter name to use at the output.
- **List of variables checks** (optional):
 - **Var_out:** Enter output name of variable to check (must be unique).
 - **Check NaN:** To filter out NaNs and consider the value incorrect.
 - **Check Zero:** To filter out zeros and consider the value incorrect.
 - **Check Empty:** To filter out empty strings or arrays and consider the value incorrect.
 - **Check Freeze:** If selected, check if value is freezed using *Data freeze time (ms)*. If so, input is considered faulty. Only applied on numbers.
 - **Check OOR:** To filter out out-of-range values and consider the value incorrect. *Minimum* and *Maximum* parameters must be defined below.
 - **Minimum:** Enter minimum limit to *Check OOR* on this variable.
 - **Maximum:** Enter maximum limit to *Check OOR* on this variable.
 - **Check Outlier:** To filter out outlier values and consider the value incorrect.
 - **Max noise:** Enter maximum noise (typical 3 x standard deviation) to *Check Outlier* on this variable.
 - **Max evo:** Enter maximum evolution speed (per second) to *Check Outlier* on this variable.
 - **Check presence:** If selected, check if inputs contain these data.
 - **Validity variable:** If selected, this variable is used to check source validity. Must be a boolean.
 - **Invert validity:** If selected, invert boolean logic of this *Validity variable* (default: var=true is OK, inverted: var=false is OK)
 - **Smooth switching:** If selected, smooth an output data gap generated by a position change.
 - **Slope:** If *Smooth switching* selected, enter slope (absolute value per second) to reduce the output data gap.
- **Port labels and timeouts:** For each *Input-x*, enter a timeout (in ms) to consider the input as faulty. 0 is ignored, 1 is minimal value. You can also define a personal label instead of *Input-x*.

Inputs

Input_x * Connect your redundant inputs. The input order defines priority: *input-1* is the highest priority.

Ext control * **Select** (*int*): Define the input to use. If auto is selected (0), input is automatically switched in case of failure detection. * **Enable_Input_x** (*bool*): Enable or disable input X (enabled by default).

Outputs

Valid_out * Main output containing active input variables.

Status * *autoMode* (*bool*): true if auto position is selected, false otherwise. * *select* (*int*): selected input number. * *selectStr* (*str*): selected input label. * *alarm* (*int*): 1 if problem detected on input and no solution is found, 0 otherwise * *warning* (*int*): 1 if problem detected on input and an alternative solution is found, 0 otherwise * *status* (*int*): Box status. See *Notes* below. Only enabled inputs are considered. * *state_icon* (*uchar array, json*): Box state icon as JSON object. * *Input_X_status* (*int*): Input status. See *Notes* below. All inputs are considered. * *enabledInputListStatus* (*uchar array, json*): List of enabled input status as JSON array. * *filteringBadData* (*bool*): true if the box currently filter out data because an error has been detected and position is not yet changed

Debug * *Input_X_checkFlags* (*int*): bitfield indicating which errors are detected. See *Notes* below. * *Input_X_errRate* (*float*): failure filter output, indicates the rate of instantaneous errors detected on input data stream.

WebApp controls

- **Valid_out** (*int*): On your dashboard, use a “Slider” widget to define the input to use. If auto is selected (default), input is automatically switched in case of failure detection.
- **Enable_Input_x** (*bool*): On your dashboard, use Button widgets to enable or disable some inputs (enabled by default).

Notes

Status convention: * bit 0 : ON/OFF (0=off 1=on) * bit 1 : Problem detected (0=valid 1=error) * bit 2 : Problem severity (0=warning 1=error)

Box status

State	State icon color	Meaning with slider in auto position	Meaning with slider in manual position
0 (0b000)	“dimgray”	Not used	Not used
1 (0b001)	“lime”	OPERATIONAL. First input is valid and selected	OPERATIONAL. Selected input is valid
3 (0b011)	“orange”	WARNING. First input is invalid but one secondary input is valid	Not used
7 (0b111)	“red”	ERROR. All inputs are invalid	ERROR. Selected input is invalid

Input status

State	WebApp color	Meaning
0 (0b000)	“dimgray”	Input off
1 (0b001)	“lime”	Input is valid
3 (0b011)	“orange”	Data received but detected as invalid (future usage)
7 (0b111)	“red”	No data received (input on timeout)

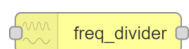
Check flags

Bit	Meaning
0x001	NAN value detected
0x002	Zero value detected
0x004	Empty value detected
0x008	Freeze value detected
0x010	Validity check is NOK

Bit	Meaning
0x020	Out Of Range value detected
0x040	Outlier value detected
0x080	Incomplete dictionary
0x100	Presence check is NOK

Outlier rejection The failsafe box can reject outliers and consider them as incorrect values if the box **Check Outlier** is checked. The logic to detect an outlier value is based on the following model. The value is considered as a measurement of a parameter. The parameter can evolve with a speed up to **Max eva**. The measurement is considered noisy and is the error of measurement is up to **Max noise**. Based on this model, if the value evolves slowly, it is considered as correct. If the value evolves too rapidly, and neither the parameter evolution speed, nor the measurement noise can explain such a change, the value is considered as incorrect.

11.34 freq_divider



Description

This Manta box performs a sampling frequency division on incoming data.

Properties

- **Name:** Manta box name
- **Divider:** Divisor factor to reduce output sampling frequency
- **Outputs:** Number of channels to divide

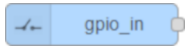
Inputs

- Connect the data streams containing variables to process. Only one wire must be connected per input.

Outputs

- Data with divided sampling frequency ($fs \rightarrow fs/divider$)

11.35 gpio_in



Description

This Manta box handles digital input.

Properties

- **Name:** Manta box name
- **GPIO line:** Select the digital input port.
- **Mode:** In “Edge” mode, return raw edges. In “Frequency” mode, return frequency between rising edges (pulse frequency computation)

Edge mode :

- **Edge:** Select when a signal changes from one state to another.
- **Debouncing time:** When a switch is pressed, it does not provide a clean edge. Adjust the debouncing time to filter out any glitches.
- **Invert:** Invert the digital signal output of the box.
- **Period (s):** Maximum period between data emission.
- **Add TS:** Add timestamp to output data flow

Frequency mode :

- **Output frequency:** Frequency of output message containing pulse frequency computation
- **Timeout:** If no rising edge is detected during the delay time, the output is set to zero.

Outputs

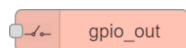
Edge mode :

- Digital input value (*Boolean*, no unit)

Frequency mode :

- EdgeFrequency

11.36 gpio_out



Description

This Manta box manage digital output.

Properties

- **Name:** Manta box name
- **GPIO line:** Select the digital output port
- **Type:** Select the output type. It could be STATE (on/off) or PWM (Pulse Width Modulation). The PWM option is only available on port 1. For STATE mode, port 2 & 3 must be preferred.
- **Init state:** Select the initial state of the output.
- **Invert:** Check to invert boolean logic of the output.
- **Frequency:** Enter PWM frequency (from 0.2Hz to 1000Hz). It is the interval of time between successive occurrences of the same state.
- **Duty cycle:** Enter PWM duty cycle (from 0% to 100%). It is the ratio of the high period to the total period of a pulse wave.

Inputs

- **State** (*Boolean*): In case of STATE mode, digital signal to generate. In case of PWM mode, permits to disable PWM output.
- **Duty cycle** (*Integer*): In case of PWM mode, value from 0% to 100% to modify PWM duty cycle.
- **Frequency** (*Float*): In case of PWM mode, value from 0.2Hz to 1000Hz to modify PWM frequency.

Notes

Exocet digital outputs are open collector outputs, so active state is near 0 and off state is high impedance. An external pull-up resistor or other electronic circuit is mandatory to raise the output voltage and drive the current when the output is turned off.

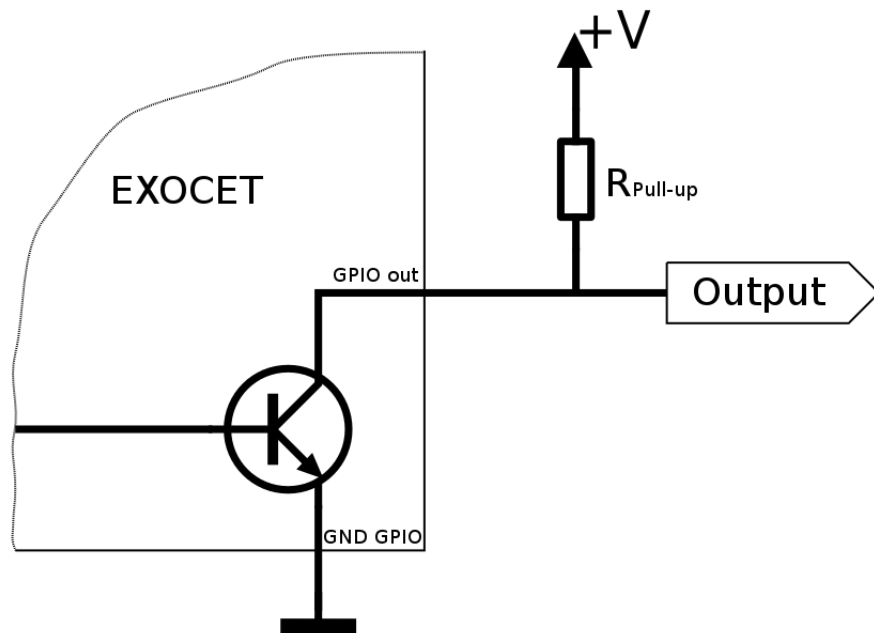
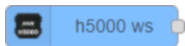


Figure 81

11.37 h5000



Description

The purpose of this Manta box is to import B&G H5000 sailing instrument variables.

Properties

- **Name:** Manta box name
- **Ip:** H5000 IP address
- **Port:** Port number used by H5000 to export data

Use the variable list to check/uncheck the H5000 variables to export. *Select All* and *Unselect All* buttons can be used to select/unselect all the variables.

Outputs

- The list of selected variables values.

11.38 height_estimation



Description

Estimates the height and height derivative at specified locations by fusing height and inertial measurements. The box also estimates water elevation and slopes.

The algorithm is a Kalman filter using a model for the inertial navigation system, for the height sensors and for the waves. The wave model is based on the state-of-the-art knowledge on ocean-wave spectra.

The filter is capable to reject false height measurements when they seem unlikely and is thus robust to spray, short-term height sensor failure and even long-term height sensor failure in the presence of height sensor redundancy. ### Properties

Box properties

- **Name** Name of the box.
- **Frequency** Output frequency of the box in Hz.

INS configuration

- **Inertial sensor gives**

raw accelerations: height estimation uses raw acceleration measurements. Raw acceleration measurements do not include gravity, i.e. it is $\sim 9.81 \text{ m/s}^2$ upward when the sensor is motionless.

inertial accelerations: height estimation uses inertial acceleration measurements. Inertial acceleration includes gravity, i.e. it is 0 m/s^2 when the sensor is motionless.

heave speed: height estimation uses heave speed.

- **Get heel and trim accuracy from input** Check this box to provide the 1σ accuracy of the pitch and roll angles as inputs of the box.
- **Heel angle 1σ accuracy** Accuracy on the heel angle ($\sim \text{max. error} / 3$).
- **Trim angle 1σ accuracy** Accuracy on the trim angle ($\sim \text{max. error} / 3$).
- **Gyros ARW** Angular random walk of the gyroscopes (i.e. white noise on the rotation rate measurements).
- **Accel. VRW** Velocity random walk of the accelerometers (i.e. white noise on the acceleration measurements).

- **Accel. bias uncertainty** 1σ accuracy of the uncertain bias on the accelerometers (\sim max. error / 3).
- **INS computation point** Coordinates of the point where the inertial navigation unit computes the heave speed or where the accelerations are measured.
- **INS Variable Mapping** Section to define input variable names of the INS data.

Altimeter configuration

- **Input height min/max** Minimum and maximum of valid height measurements. Measurements outside this range are considered invalid.
- **Pos.** Coordinates of the height sensor.
- **1σ accuracy** Accuracy of the height sensor measurement (\sim max. error / 3).
- **Enable update** Check this box to use the sensor for the estimation. Otherwise the validity test is still performed, but the measurement is not used.
- **Height** Input variable name of the height measurement.
- **Timestamp (opt.)** Input variable name of the measurement timestamp. Can be left blank if no timestamp is available.

Estimation options

- **Swell longitudinal slope** Check this box to estimate the longitudinal slope of the swell. Check this box only if at least two height sensors have different X positions. If unchecked, the longitudinal slope is considered as zero.
 - **Swell lateral slope** Check this box to estimate the lateral slope of the swell. Check this box only if at least two height sensors have different Y positions. If unchecked, the lateral slope is considered as zero.
- Check both boxes only if you have at least 3 unaligned height sensors.

Output configurations

- **Pos.** Coordinates where the height and height derivative is computed.
- **Height** Output height variable name.
- **Height deriv.** Output height derivative variable name.

Advanced options

- **Boat max. height** Maximum height of the boat with respect to the surface.
- **Max. significant wave height** Maximum significant wave heights that can be encountered.
- **Mahalanobis validity thd** Validity threshold of the χ^2 test for 1D samples. Default is 6.63.

Inputs

Nota: Default units are °, °/s and m.

Name	Data	Description
INS meas.	<i>Heel angle</i>	Roll angle
	<i>Trim angle</i>	Pitch angle
	<i>Body rot. rate X</i>	Angular speed around the boat x-axis
	<i>Body rot. rate Y</i>	Angular speed around the boat y-axis
	<i>Body rot. rate Z</i>	Angular speed around the boat z-axis
	<i>Heel angle 1σ accuracy</i>	Accuracy on the roll angle
	<i>Trim angle 1σ accuracy</i>	Accuracy on the pitch angle
	<i>Heave speed</i>	Heave speed (if selected)
	<i>Acc. X</i>	Measured acceleration along the boat x-axis (if selected)
	<i>Acc. Y</i>	Measured acceleration along the boat y-axis (if selected)
	<i>Acc. Z</i>	Measured acceleration along the boat z-axis (if selected)
	<i>Timestamp</i>	Timestamp of the INS measurement (opt.)
Height sensor #i	<i>Height</i>	Height measurement
	<i>Timestamp</i>	Height measurement timestamp (opt.)
	<i>FuseOrder</i>	0 to ignore the measurement (validity test is still done); 1 to fuse the measurement only if valid; 2 to force fusion even if χ^2 deviation is > 1 . FuseOrder is also available with a widget on a dashboard.

Outputs

Name	Data	Description
Outputs	Timestamp	Timestamp of the output
	<i>Height #i</i>	Height output at location #i (m)
	<i>Height derivative #i</i>	Height derivative output at location #i (m/s)
Status	Timestamp	Timestamp of the status
	Heave	Estimated heave (m)
	Heave_std	Estimated heave 1 σ accuracy (m)
	HeaveSpeed	Estimated heave speed (m/s)
	HeaveSpeed_std	Estimated heave speed 1 σ accuracy (m/s)
	Height	Height (m)
	Height_std	Height 1 σ accuracy (m)
	HeightDerivative	Height derivative (m/s)
	HeightDerivative_std	Height derivative 1 σ accuracy (m/s)
	WaveHeave	Wave heave (m)
	WaveHeave_std	Wave heave 1 σ accuracy (m)
	WaveHeaveSpeed	Wave heave speed (m/s)
	WaveHeaveSpeed_std	Wave heave speed 1 σ accuracy (m/s)
	WaveSlopeX	Wave longitudinal slope (°)
	WaveSlopeX_std	Wave longitudinal slope 1 σ accuracy (°)
	WaveSlopeY	Wave lateral slope (°)
	WaveSlopeY_std	Wave lateral slope 1 σ accuracy (°)
	AccBias	Vertical bias on measured accelerations (m/s ²)
	AccBias_std	Vertical bias 1 σ accuracy (m/s ²)

Name	Data	Description
	HeightSensor#i_InvalidRatio Ratio of invalid measurements HeightSen- sor#i_Chi2Dev χ^2 deviation (if >1 the measurement is considered invalid)	

Nota: Status outputs are expressed at the INS computation location

Notes

Glossary

- **INS** Inertial Navigation System
- **ARW** Angular Random Walk
- **VRW** Velocity Random Walk

Illustration Side view of boat

Front view of boat (starboard on the left)

Reference frame All coordinates (INS computation location, height sensors and height outputs) must have a common origin. The choice of the origin is at the convenience of the user.

The axis system is SNAME (+x forward/+y starboard/+z downward).

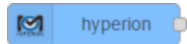
Heave and heave speeds are positive downward.

A positive longitudinal wave slope corresponds to an ascent forward. A positive lateral wave slope corresponds to a descent towards starboard.

Height measurements and height outputs are positive upward, i.e. when the sensor or output location is over water.

Timestamped measurements Input INS measurement and height measurements can be timestamped. The height estimation can process these measurements chronologically and interpolates from one to another to be insensitive to data delays. When the input timestamp name is not given (blank field), the algorithm processes the measurement as if it is not delayed.

11.39 hyperion



Description

Export variables such as microstrains or temperatures from Hyperion, the Micron Optics optical interrogator. It is compatible with si155 and si255 Micron Optics interrogators.

The Hyperion box is configured using a *moi* file. *moi* files are generated by Micron Optics *Enlighth* software. They contain interrogator connectivity parameters, sensors or variables definitions.

Note: The grouping notion of Enlighth is not taken into account for now.

Properties

- **Name:** Manta box name
- **Conf file:** Select one of the already downloaded *moi* file
- **Fout divider:** Enter a frequency divider to apply on data exported by the Hyperion interrogator
- **Upload new Conf:** Upload a new *moi* file from your PC to the Exocet
- **Delete or Download configuration files:** Use *Trash icon* to delete a configuration file and the *download icon* to download the configuration file from the Exocet to the PC
- **Select variables:** Check/Uncheck the variables to export out of the Manta box
- **Select channel:** Check/Uncheck the channel

Inputs

- **Enable** (*bool*): Connect a boolean to enable/disable the box. Default is enable.

Outputs

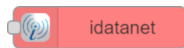
- A set of variables such as microstrains or temperatures according to the configuration file.

Notes

Control

- Use the button on the left side of the box to release the access to the interrogator. When released, the Hyperion interrogator can be accessed by *Enlighth Software*.

11.40 idatanet



Description

Export data to iDataNet iPhone/iPad app. Data are sent over UDP to the iOS device.

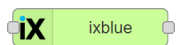
Properties

- **Name:** Manta box name
- **IP:** IP address of iOS device
- **Port:** UDP port to send data to
- **Decimals:** Number of decimals for exported float numbers
- **Specific decimals:** List of variables that need a specific *Decimals* parameter

Inputs

- Data to send (multiple wires can be connected).

11.41 ixblue



Description

This Manta box decode several IxBlue Inertial Navigation System Protocols.

Vessel reference frame: The data provided by the IxBlue Manta box use the standard SNAME's (1950) notation (forward/starboard/downward and North/East/Down frames).

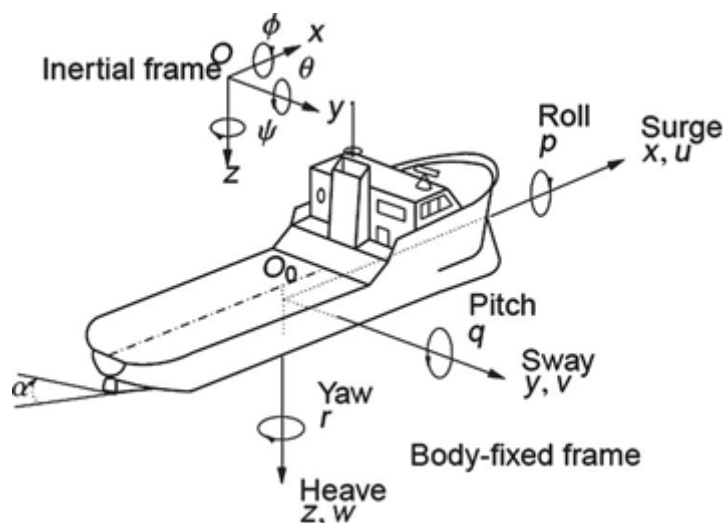


Figure 82

Properties

- **Name:** Manta box name
- **Protocol:** Select protocol to decode

Inputs

- UDP or serial stream to decode.

Outputs

Ifremer Victor:

Data Name	Description
Roll	roll in degrees -180/+180
Pitch	pitch in degrees -180/+180
Heading	heading in degrees 0/360
RollRate	roll rate in degrees per second
PitchRate	pitch rate in degrees per second
YawRate	heading rate in degrees per second

Data Name	Description
SurgeAccel	Surge acceleration in meters per second squared
SwayAccel	sway acceleration in meters per second squared
HeaveAccel	Heave acceleration in meters per second squared
Heave	heave in meters

INDYN:

Data Name	Description
Latitude	latitude in degrees -90/+90
Longitude	longitude in degrees -180/+180
Altitude	altitude in meters
Roll	roll in degrees -180/+180
Pitch	pitch in degrees -180/+180
Heading	heading in degrees 0/360
RollRate	roll rate in degrees per second
PitchRate	pitch rate in degrees per second
YawRate	heading rate in degrees per second
SurgeSpeed	surge speed in meters per second

PIXSE-SUBGP1:

Data Name	Description
Time	UTC time
Latitude	in degrees -90/+90, + for north
Longitude	in degrees -180/+180, + for east
RangeToBottom	in meters, if DVL bottom track data is valid
Altitude	altitude in meters
NorthSpeed	in meters per second, + northwards

Data Name	Description
EastSpeed	in meters per second, + eastwards
VerticalSpeed	in meters per second, + upwards
SpeedThWater	in meters per second
Roll	degrees -180/+180, + when port is up
Pitch	degrees -180/+180, + when bow is down
HeadingTrue	in degrees 0/360, 0 is true north
LongVelocity	in meters per second, + forwards
TransVelocity	in meters per second, + to starboard
InsAlgoStatus	INS algo status

PIXSE-SUBGP2:

Data Name	Description
LatitudeStdDev	latitude standard deviation
LongitudeStdDev	longitude standard deviation
RangeToBottomStdDev	range to bottom standard deviation
DepthStdDev	depth standard deviation
NorthSpeedStdDev	north speed standard deviation
EastSpeedStdDev	east speed standard deviation
VerticalSpeedStdDev	vertical speed standard deviation
SpeedThWaterStdDev	SOW standard deviation
HeadingStdDev	heading standard deviation
RollStdDev	roll standard deviation
PitchStdDev	pitch standard deviation
LongVelStdDev	longitudinal velocity standard deviation
TransVelStdDev	transverse velocity standard deviation

PIXSE-SUBGP3:

Data Name	Description
RollRate	in degrees per second, + when increase
PitchRate	in degrees per second, + when increase
YawRate	in degrees per second, + when increase
LongitudinalAcc	in degrees per second squared, + forwards
TransverseAcc	in degrees per second squared, + to starboard
VerticalAcc	in degrees per second squared, + upwards

CONTROL:

Data Name	Description
AccXV1	acceleration XV1 in meters per second squared
AccXV2	acceleration XV2 in meters per second squared
AccXV3	acceleration XV3 in meters per second squared
RotRateXV1	rotation rates XV1 in degrees per second
RotRateXV2	rotation rates XV2 in degrees per second
RotRateXV3	rotation rates XV3 in degrees per second

LONG BIN NAV HR:

Data Name	Description
Time	data time in seconds
Latitude	latitude in degrees -180/+180
Longitude	longitude in degrees -180/+180
Altitude	altitude in meters
Heave	heave in meters
VelocityNorth	north speed in meters per second
VelocityEast	east speed in meters per second
VelocityDown	down speed in meters per second

Data Name	Description
Roll	roll in degrees -180/+180
Pitch	pitch in degrees -180/+180
Heading	heading in degrees 0/360
RollRate	roll rate in degrees per second
PitchRate	pitch rate in degrees per second
YawRate	yaw rate in degrees per second
UserStatus	INS user status
LatitudeStdDev	latitude standard deviation in meters
LongitudeStdDev	longitude standard deviation in meters
VelocityNorthStdDev	north speed standard deviation in meters per second
VelocityEastStdDev	east speed standard deviation in meters per second
VelocityDownStdDev	down speed standard deviation in meters per second
RollStdDev	roll standard deviation in degrees
PitchStdDev	pitch standard deviation in degrees
HeadingStdDev	heading standard deviation in degrees

SEAPATH:

Data Name	Description
Time	data time in seconds
Latitude	latitude in degrees -180/+180
Longitude	longitude in degrees -180/+180
Altitude	altitude in meters
Heave	heave in meters
VelocityNorth	north speed in meters per second
VelocityEast	east speed in meters per second
VelocityDown	down speed in meters per second
Roll	roll in degrees -180/+180

Data Name	Description
Pitch	pitch in degrees -180/+180
Heading	heading in degrees 0/360
RollRate	roll rate in degrees per second
PitchRate	pitch rate in degrees per second
YawRate	yaw rate in degrees per second
Status	status: 0xAA=valid, 0x00=invalid

11.42 json_dec

0000

Description

Extract data from JSON.

Properties

- **Name:** Manta box name

Inputs

- **Json** (*Opaque*): Data to decode

Outputs

- **Data:** Decoded data

11.43 json_enc



Description

Encode data to JSON.

Properties

- **Name:** Manta box name
- **Output name:** Exported data name. Manta box name is used if not defined.

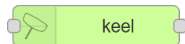
Inputs

- **Data:** Data to encode

Outputs

- **Json** (*Opaque*): Encoded data

11.44 keel



Description

Decode data of IMOCA keel management system.

Properties

- **Name:** Manta box name
- **System state CAN ID:** Enter CAN ID of System State can frame (hexadecimal)
- **Keel state CAN ID:** Enter CAN ID of Keel State can frame (hexadecimal)
- **Capa state CAN ID:** Enter CAN ID of Capa State can frame (hexadecimal)

Inputs

- **CAN_input:** Must be connected to a canbus in box

Outputs

- **System_state**: Export general system state data

Nom	Description
RollUp	Compteur de trame continue
NbFsc	Nombre de façade connecté [0..3]
PresenceCapa	0 : capa absente 1 : capa présente
PosSensor	0 : capteur absent 1 : capteur present
StrdPresSensor	0 : capteur absent 1 : capteur present
OVLO	Over Voltage Lock Out : 0 : tension batterie Ok 1 : tension batterie trop importante
UVLO	Under Voltage Lock Out : 0 : tension batterie Ok. 1 : tension batterie trop faible
CCEvRelPort	Court-circuit électrovanne release bâbord : 0 : Ev Ok. 1 : Ev en court-circuit.
CCEvRelStar	Court-circuit électrovanne release tribord : 0 : Ev Ok. 1 : Ev en court-circuit.
CCEvMouvPort	Court-circuit électrovanne mouvement bâbord : 0 : Ev Ok. 1 : Ev en court-circuit.
CCEvMouvStar	Court-circuit électrovanne mouvement tribord : 0 : Ev Ok. 1 : Ev en court-circuit.
CCCmdMotor	Court-circuit commande moteur : 0 : Sortie Ok. 1 : Sortie en court-circuit.

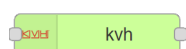
- **Keel_state:** Export keel state data

Nom	Description
InvPressure	Indique une inversion de poussé de la quille (la quille génère de la portance).0 : la quille génère du couple de redressement (normal).1 : la quille génère de la portance (inversion de poussée). Remarque : De part la géométrie du vérin l'inversion de poussé n'est pas directement lié à une inversion de pression.
KeelAngle	Angle de quille actuel [-38°..38°]. Un angle négatif indique la quille à bâbord, un angle positif indique la quille à tribord.
PortPressure	Pression dans la chambre bâbord (nez) du vérin.
StarboardPressure	Pression dans la chambre tribord (fond) du verin

- **Capa_state:** Export super capacitors state data

Nom	Description
RollUp	Compteur de trame continue
VBat	Tension batterie
VCapa	Tension supercondensateur
ICapa	Courant des supercondensateurs

11.45 kvh



Description

This Manta box decode some ANPP binary messages from KVH Inertial Navigation System.

Supported messages are: System State Packet (20), Raw Sensors Packet (28), Body Velocity Packet (36), External Body Velocity Packet (47), External Heading Packet (48), Heave Packet (58), North Seeking Initialisation Status Packet (71), Automotive Packet (73)

Vessel reference frame: The data provided by the KVH Manta box use the standard SNAME's (1950) notation (forward/starboard/downward and North/East/Down frames).

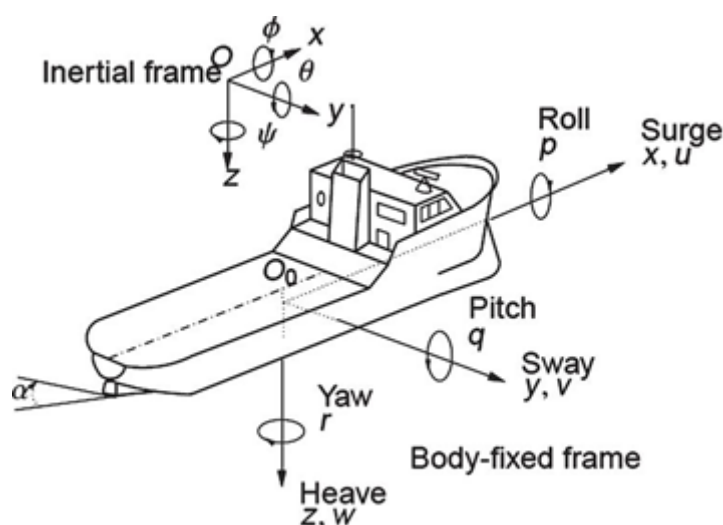


Figure 83

Properties

- **Name:** Manta box name
- **Packet ID xx:** If selected, decode this message

Inputs

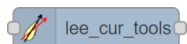
- **Input:** Serial stream to decode.

Outputs

- **20-System State:** Output data of System State Packet (20)
- **28-Raw Sensors:** Output data of Raw Sensors Packet (28)

- **36-Body Velocity:** Output data of Body Velocity Packet (36)
- **47-External Body Velocity:** Output data of External Body Velocity Packet (47)
- **48-External Heading:** Output data of External Heading Packet (48)
- **58-Heave:** Output data of Heave Packet (58)
- **71-North Seeking Init Status:** Output data of North Seeking Initialisation Status Packet (71)
- **73-Automotive:** Output data of Automotive Packet (73)

11.46 lee_cur_tools



Description



Figure 84

This box realizes calculations in the “SOG, SOW and CUR” Triangle. It contains several independants calculations which are activated on received data and/or optionals properties selected.

Availables Calculations are : - Calculate the course and the Speed Over Water (referenced to the course) from the selected sensors. - Get the leeway with current estimation as input. - Estimate the leeway with Hull shape factor formula. - Calculate the boatspeed and leeway using a 2D speedmeter sensor. - Estimate the water referenced Boatspeed for a flying boat without speedometer. - Get the current with the leeway estimation as input.

Properties

- **Name:** Manta box name
- **Calculate Course:** Enable output 0 calculation, see description below
- **Calculate COG-Heading Leeway:** Enable output 1 calculation, see description below
- **Calculate HSF-Leeway:** Enable output 2 calculation, see description below
- **Calculate BoatSpeed:** Enable output 4 calculation, see description below
- **Calculate Current:** Enable output 5 calculation, see description below
- **HullShapeFactor:** Hull Shape Factor ([0..20]), used to calculate output 2

- **Max Leeway** (°): Limit calculated leeway to +/- this maximum
- **Min SOG** (kn): Minimum speed over ground, no Leeway output at low speed (inaccurate COG)
- **Use Velocity North East** : Use North East Ground Velocity as input instead of COG, SOG

Inputs

Input 0

Default timeout = 2s.

Value Name	Units	Description	Calculations using this input
Leeway	(°)	Leeway angle	- Course, SOW (Output 0) - BSP (Output4) - Current (Output 5)

Note : For a leeway measured with a sensor, you can interface it with a net2000 “leeway” input or the NMEA0183 \$xxLWY sentence.

Input 1

Default timeout = 1s.

Value Name	Units	Description	Calculations using this input
Heel	(°)	Heel angle	- Hull Shape Factor Leeway calculation (Output 2)
Heading	(°T)	Heading angle	- Leeway (Output 0) or Current (Output 5) - Course (Output 1)

Input 2

Default timeout = 2s.

Value Name	Units	Description	Calculations using this input
BoatSpeed	(kn)	Longitudinal Boat Speed; water referenced	- SOW (Output 0) - HSF Leeway calculation (Output 2) - Current (Output 5)

Input 3 COG, SOG or North East Ground Velocity if *Use Velocity North East* is checked. Default timeout = 2s.

Value Name	Units	Description	Calculations using this input
COG	(°T)	CourseOver Ground	- Current (Output 5) or Leeway (Output 1) - SOW and BSP (Output 4)
SOG	(kn)	Speed Over Ground	- Current (Output 5) or Leeway (Output 1) - SOW and BSP (Output 4)

Value Name	Units	Description	Calculations using this input
VelocityNorth	(m/s) or (kn)	Ground Velocity North	- Current (Output 5) or Leeway (Output 1) - SOW and BSP (Output 4)
VelocityEast	(m/s) or (kn)	Ground Velocity East	- Current (Output 5) or Leeway (Output 1) - SOW and BSP (Output 4)

Input 4

Default timeout = 2s.

Value Name	Units	Description	Calculations using this input
LongitudinalSpeedWater	(kn)	Longitudinal Boat Speed, water referenced	- Leeway, BSP and SOW (Output 3)
TransverseSpeedWaterReferenced	(kn)	Transverse Boat Speed, water referenced	- Leeway, BSP and SOW (Output 3)

Note : You can connect this input to a net2000 “boat speed 2D” input, PGN 130578.

Input 5

Default timeout = 5s.

Value Name	Units	Description	Calculations using this input
CURD	(°T)	Tidal stream or ocean current direction	- BSP, SOW (Output 4) - Leeway (Output 1)
CURS	(kt)	Tidal stream or ocean current speed	- BSP, SOW (Output 4) - Leeway (Output 1)

Note : You can get a grib current from your navigation software using the NMEA0183 \$xxVDR sentence

Outputs

Output 0

Require *Calculate Course* properties to be ticked to be calculated. Calculated at the INS stream rate.

Value Name	Units	Description	Necessary inputs to calculate this data
Leeway	(°)	Leeway angle	Copied from Input 0
Course	(°T)	Course	- Leeway (Input 0) - Heading (Input 1)
SpeedOverWater	(kn)	Speed Over Water (Referenced in Course direction)	- Leeway (Input 0) - BSP (Input 2)

Output 1

Require *COG-Heading Leeway* properties to be ticked and *MaxLeeway* (°) and *MinSOG* (kt) to be filled to be calculated. Calculated at the GNSS stream rate.

Value Name	Units	Description	Necessary inputs to calculate this data
Leeway	(°)	Leeway angle	- Heading (Input 1) - COG, SOG (Input 3) - TideSet, TideRate (Input 5)
BoatSpeed	(kn)	Longitudinal Boat Speed, water referenced	- Heading (Input 1) - COG, SOG (Input 3) - TideSet, TideRate (Input 5)

Value Name	Units	Description	Necessary inputs to calculate this data
Course	(°T)	Course	- Heading (Input 1) - COG, SOG (Input 3) - TideSet, TideRate (Input 5)
SpeedOverWater	(kn)	Speed Over Water (Referenced in Course direction)	- Heading (Input 1) - COG, SOG (Input 3) - TideSet, TideRate (Input 5)
Leeway_orig	(°)	Leeway angle before current correction	- Heading (Input 1) - COG, SOG (Input 3) - TideSet, TideRate (Input 5)
BoatSpeed_orig	(kn)	Longitudinal Boat Speed, ground referenced (before current correction)	- Heading (Input 1) - COG, SOG (Input 3) - TideSet, TideRate (Input 5)

Output 2

Require *HSLeeway* properties to be ticked and *HullShapeFactor* to be filled to be calculated. Accurate only for a sailboat without moving appendage. Calculated at the INS stream rate.

Value Name	Units	Description	Necessary inputs to calculate this data
Leeway	(°)	Leeway angle estimated using the Hull Shape Factor method	- Heel (Input 1) - BSP (Input 2)

Output 3

Value Name	Units	Description	Necessary inputs to calculate this data
Leeway	(°)	Leeway angle calculated using 2 axes speedometer	LongitudinalSpeedWaterReferenced, TransverseSpeedWaterReferenced (Input 4)

Value Name	Units	Description	Necessary inputs to calculate this data
BoatSpeed	(kn)	Longitudinal Boat Speed, water referenced	2 axes speedometer (input 4)
SpeedOverWater	(kn)	Speed Over Water (Referenced in Course direction)	2 axes speedometer (input 4)

Output 4

Require *CalcBoatspeed* propertie to be ticked to be calculated. Calculated at the INS stream rate.

Value Name	Units	Description	Necessary inputs to calculate this data
BoatSpeed	(kn)	Longitudinal Boat Speed, water referenced,	- Leeway (Input 0) - COG, SOG (input 3) - TideRate, TideSet (Input 5)
SpeedOverWater	(kn)	Speed Over Water (Referenced in Course direction)	- COG, SOG (input 3) - TideRate, TideSet (Input 5)

Output 5

Require *CalcCurrent* propertie to be ticked to be calculated. Calculated at the BoatSpeed stream rate.

Value Name	Units	Description	Necessary inputs to calculate this data
CURD	(°T)	Tidal stream or ocean current direction	- Leeway (Input 0) - Heading (Input 1) - Boatspeed (Input 2) - COG, SOG (Input 3)
CURS	(kt)	Tidal stream or ocean current speed	- Leeway (Input 0) - Heading (Input 1) - Boatspeed (Input 2) - COG, SOG (Input 3)

WebApp controls

- **HullShapeFactor** (float): If *Calculate HSF-Leeway* is selected, on your dashboard, use a “Set Number” widget to change the *HullShapeFactor* value.
- **MaxLeeway** (float): If *Calculate COG-Heading Leeway* or *Calculate HSF-Leeway* is selected, on your dashboard, use a “Set Number” widget to change the *MaxLeeway* value.
- **CurrentCorrection** (bool): If *Calculate COG-Heading Leeway* or *Calculate BoatSpeed* is selected, on your dashboard, use a “button” widget to activate or deactivate current correction.

Notes

Leeway, Current, Course definitions:

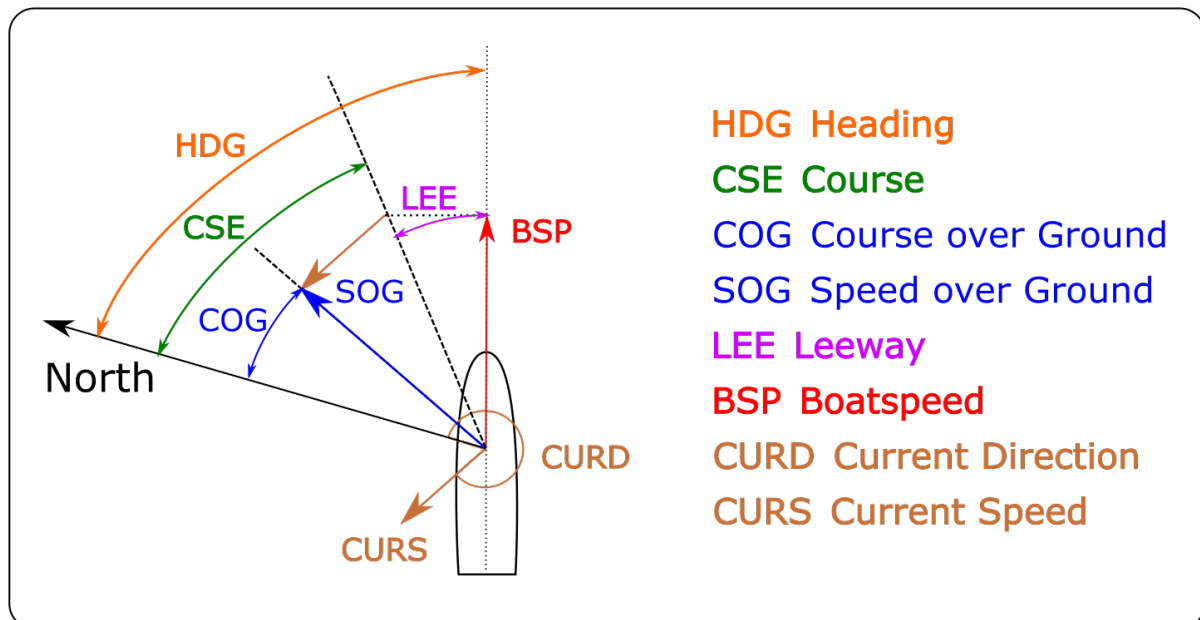
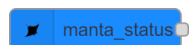


Figure 85

11.47 manta_status



Description

This box exports a status of Manta boxes at 5 Hz.

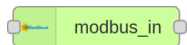
Properties

- **Name:** Manta box name
- **Out error names:** Select to output names of boxes seen at least one time in error state
- **Out warning names:** Select to output names of boxes seen at least one time in warning state

Outputs

- **error** (*boolean*): true if a box is on error, else false
- **error_nb** (*integer*): number of boxes on error
- **error_x** (*string*): name of boxes on error
- **warning** (*boolean*): true if a box is on warning, else false
- **warning_nb** (*integer*): number of boxes on warning
- **warning_x** (*string*): name of boxes on warning

11.48 modbus_in



Description

This Manta box import data from a remote Modbus device. It works as a master / client, the remote device should be a slave / server.

Properties

- **Name:** Manta box name
- **Context:** Select Modbus context: RTU or TCP
- **Master:** Check to work as a master. Otherwise, it works as a slave.
- **Serial port:** RTU context only. Serial port configuration box. Press the pencil icon to create a new one. Fill the serial port parameters, then press the *Add* button. Once created, it can be shared with other boxes.
- **Remote IP:** TCP context only. Enter IP address of the remote device to talk.
- **Remote port:** TCP context only. Enter used port of the remote device (default Modbus port is 502).
- **Slave ID:** Define the slave ID of the remote device to talk (from 1 to 247, broadcast=0, tcp=255).
- **List of periodic requests to send** (master mode only, maximum 100):

- **Function code:** Select the requested function code
- **Address:** Address to start read
- **Number:** Number of data to read (max 100)
- **Period:** Period of request in milliseconds
- **Mapping of data** (slave mode only, maximum 100):
 - **Type:** Select data type
 - **Address:** Address of data
 - **Variable:** Optional, enter name of variable. If not set, variable name is *Type_Address*.

Inputs

Inputs 0: Read request

Name	Type	Description
functionCode	Int	1: read bits (coils); 2: read input bits; 3: read registers (Holding); 4: read input registers
address	Int	Address to start read
number	Int	Number of data to read (max 100)

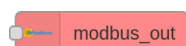
Outputs

- **data** (*Int*): read data. Variable names start with `bit_`, `inputBit_`, `register_` or `inputRegister_`, and are followed by the data address.

Notes

COM3 to COM6 use an USB extension and are so a little less effective. COM1 and COM2 must be preferred.

11.49 modbus_out



Description

This Manta box export data to a remote Modbus device. It could work as a master / client, in this case it sends request to write data to a remote device; or as a slave / server, in this case it make data available in a Modbus object dictionary.

Properties

- **Name:** Manta box name
- **Context:** Select Modbus context: RTU or TCP
- **Master:** Check to work as a master. Otherwise, it works as a slave.
- **Serial port:** RTU context only. Serial port configuration box. Press the pencil icon to create a new one. Fill the serial port parameters, then press the *Add* button. Once created, it can be shared with other boxes.
- **Remote IP:** TCP context, master mode only. Enter IP address of the remote device to talk.
- **Remote port:** TCP context, master mode only. Enter used port of the remote device (default Modbus port is 502).
- **Local port:** TCP context, slave mode only. Enter used local port (default Modbus port is 502).
- **Slave ID:** Define the slave ID of the remote device to talk in master mode, or define the local slave ID otherwise (from 1 to 247, broadcast=0, tcp=255).
- **Mapping of data** (slave mode only, maximum 100):
 - **Variable:** Enter name of variable to store (int or bool)
 - **Type:** Select data type
 - **Address:** Address of data

Inputs

Inputs 0: Write request (master mode)

Name	Type	Description
functionCode	Int	5: write bit (coil); 6: write register; 15: write bits (coils); 16: write registers; 23: write/read registers
address	Int	Address to write

Name	Type	Description
data	Bool, int, opaque register). or FloatAr ray	Data to write. Use opaque to write multiple bits (one byte = one bit). Use floatArray to write multiple registers (one float = one register).
read_addr	Int	Only used if <i>functionCode</i> =23, Address to start read
read_nb	Int	Only used if <i>functionCode</i> =23, Number of data to read (max 100)

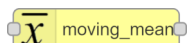
Inputs 0: Data to store (slave mode)

See *Mapping of data*

Notes

COM3 to COM6 use an USB extension and are so a little less effective. COM1 and COM2 must be preferred.

11.50 moving_mean



Description

This Manta Box can perform average, standard deviation, min/max search and others on a moving set of samples, and on a selected variables list.

Properties

- **Name:** Manta box name
- **Pass through:** export all non processed data
- **Dynamic window:** add inputs to control windows from the Manta graph
- **WebApp:** Allow window parameters access from Exocet WebApp dashboards

- **Min / Max / Step:** (WebApp option only) widget settings for window parameters
- **Export parameters:** add an output connector to export current parameters
- **Apply to all:** If checked, apply the same process to all variables of the input data stream. Variables are suffix with the process type.
- **Process type:** (Apply to all option only) Select the desired process type (see *Notes* below).
- **Percent:** (Apply to all option only) Only for percentile process type, set the threshold for the percentile process.
- **Window:** (Apply to all option only) Set the period of time to process (in seconds). This period can't be higher than 30min.
- **Auto suffix:** (Apply to all option only) Check to add an automatic suffix to all variables, else use the input field to define a custom suffix.
- **List of variables to process** (maximum 250):
 - **Process type:** Select the desired process type (see *Notes* below).
 - **Var_in:** Enter the name of variable to process
 - **Var_out:** Enter the output name of processed variable
 - **Window:** Set the period of time to process (in seconds). This period can't be higher than 30min.
 - **Percent:** Only for percentile process type, set the threshold for the percentile process.
- **Frequency divider:** Divisor factor to reduce output sampling frequency

Inputs

- **Data:** Connect here the data stream containing variables to process. Only one wire should be connected on Data input.
- **window:** Connect here a scalar to configure the “Window” dynamically.
- **Reset:** Connect here a boolean to reset dynamic and WebApp parameters to initial values defined above.

Outputs

- **Processed_data:** A list of processed data, and non processed data if *Pass through* option is selected.
- **Parameters:** The list of current processing windows.

WebApp controls

- **Output Name(s)** (*int*): If *WebApp* is selected, on your dashboard, use a “Set Number” widget to change the *Window* value.

Notes

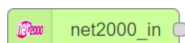
Max samples:

The max number of samples per box is 30000 (for example: 10 variables at 10Hz for 5min => $10 \times 60 \times 5 \times 10 = 30000$ samples).

Process type:

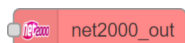
- **Average**: Get the sum of samples divided by the number of samples.
- **Variance**: Return how far the set of samples are spread out from their mean value.
- **Standard deviation**: Return the dispersion of samples from the mean value.
- **Mean deviation**: Return the average dispersion of samples from the mean value.
- **Min**: Get the min value from the current set of samples.
- **Max**: Get the max value from the current set of samples.
- **Median**: Get the middle value from the current set of samples, i.e. the value that divide the set in two equal parts.
- **Percentile**: Get the given percentile value from the current set of samples, i.e. the value below which the given percentage falls (0% gets the minimum, 100% gets the maximum, 50% gets the median).
- **Range**: Get the *Max* minus *Min* value.
- **Max of abs**: Get the max of absolute values of samples.

11.51 net2000_in



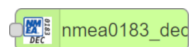
Select a source to see more help

11.52 net2000_out



Select a source to see more help

11.53 nmea0183_dec



Description

NMEA0183 protocol decoder.

Decoded sentences are : \$BWC, \$GGA, \$GLL, \$HDG, \$HDM, \$HDT, \$INDYN, \$LWY, \$MWV, \$PCBT, \$PENON, \$PHLIN, \$PHSPD, \$PHTRO, \$PIXEL, \$PRDC, \$PRDID, \$PRDM, \$PSBGB, \$PSBGI, \$PSXN, \$PTXT, \$RMB, \$RMC, \$VDR, \$VTG, \$VWR, \$XDR, \$XTE, \$ZDA, \$ZTG, !VDO !VDM.

Properties

- **Name:** Manta box name
- **Checksum Validation:** If checked, only sentences with a valid checksum are transmitted.
- **NMEA0183 sentences to decode:** Press *New NMEA0183 sentence* button. Select prefix of frames to decode.

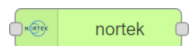
Inputs

- UDP or serial frame to decode.

Outputs

- Parsed sentence. One output per selected prefix.

11.54 nortek



Description

This Manta box decode some Nortek DVL binary messages.

Supported messages are: DF3 - Current Profile Data

Properties

- **Name:** Manta box name
- **Output convention:** Select Raw to have output as defined by Nortek

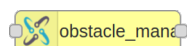
Inputs

- **Input:** UDP or serial stream to decode.

Outputs

- **Main Output:** Main single fields of received message
- **Second Output:** Other single fields of received message
- **Velocity:** Raw velocity data as a buffer + 5 first cells decoded velocity (X-Y-Za-Zb)
- **Amplitude:** Amplitude data as a buffer + 5 first cells decoded amplitude
- **Correlation:** Correlation data as a buffer + 5 first cells decoded correlation
- **AltimeterRaw:** Altimeter Raw data as a buffer
- **EchoSounder:** Echo Sounder data as a buffer
- **PercentGood:** Percent Good data as a buffer

11.55 obstacleManager



Description

This Manta box performs data fusion using data coming from the RADAR, the AIS, the OSCAR camera and the boat state.

Properties

- **Name:** Manta box name

AIS configuration

- **AIS activated:** If ticked the AIS measurements will be take into account in the data fusion
- **AIS Position noise:** white noise on the position measurement
- **AIS Speed noise:** white noise on the velocity measurements

- **AIS range threshold:** Maximum range from the position of the boat for which the measurements should be considered
- **AIS validation threshold:** Mahalanobis distance threshold for AIS measurement association. If the Mahalanobis distance to an already considered obstacle computed for the measurement is below this value, the measurement may be associated to the obstacle (unless another one has a shorter distance).

RADAR configuration

- **RADAR activated:** If ticked the RADAR measurements will be taken into account in the data fusion
- **RADAR Position noise:** white noise on the range measurements
- **RADAR bearing noise:** white noise on the bearing measurements
- **RADAR range threshold:** Maximum range for which the measurements should be considered for fusion
- **RADAR validation threshold:** Same as AIS for RADAR measurements

OSCAR configuration

- **OSCAR activated:** If ticked the OSCAR measurements will be taken into account in the data fusion
- **OSCAR Position noise:** white noise on the range measurements
- **OSCAR bearing noise:** white noise on the bearing measurements
- **OSCAR range threshold:** Maximum range for which the measurements should be considered for fusion
- **OSCAR validation threshold:** Same as AIS for OSCAR measurements

Prediction configuration

- **Prediction processing noise:** white noise applied on the position prediction
- **Prediction frequency:** Rate of obstacle position prediction computation

Miscellaneous

- **Forgetting threshold:** Threshold on the uncertainty of obstacle position. If the uncertainty reaches the threshold, the obstacle is considered lost
- **Obstacle default size:** Obstacle size if no information about it is available.
- **Web app:** If ticked, the previous parameters are accessible and can be modified from the dashboard

Inputs

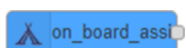
- **OSCAR:** oscar raw measurements coming from the 1st output of the OSCAR manta box.

- **RADAR**: RADAR raw measurements
- **AIS**: AIS measurement.
- **Boat_State**: Boat state info which is a mix of GNSS and AHRS information. Dictionnary must contain "Latitude", "Longitude,"SOG","COG" and "Heading"

Outputs

- **Main**: Returns info about the obstacles validated one by one
- **Auxilliary**: Returns info about all obstacles validated in one dictionnary
- **Sensors enabled** Returns which sensors info are used in the fusion algorithm

11.56 on_board_assistant



Description

Receive events from OnBoardAssistant Sailing performance software, using iDataNet, Adrena or Expedition protocol.

Properties

- **Name**: Manta box name
- **Port**: Port number to receive data from

Outputs

Output 1: Sails data

- **Main** (*String*) : main sail description
- **MainReefs** (*String*) : main reef description
- **HeadSail** (*String*) : Head sail description
- **StaySail** (*String*) : Stay sail description
- **Spinnaker** (*String*) : Spinnaker description

Output 2 : phases data

- **Subject** (*String*) : phase subject
- **StartDate** (*String*) : phase start date
- **Status** (*String*) : phase status
- **Comment** (*String*) : phase comment

11.57 opc_in



Description

This Manta box import data from a remote OPC-UA device as client.

Properties

- **Name:** Manta box name
- **URL:** URL of OPC-UA server to connect to
- **NsIndex:** OPC-UA *namespaceIndex*
- **Policy:** Select the desired policy. *OnRequest*, periodic read requests are sent. *OnDataChange*, a *subscription* is created on server side with a set of *MonitoredItems*.
- **Vars names:** Select the desired naming convention for output variables. *Custom*, use dedicated field to name Manta variables (*Id* value is used if empty). *Server display names*, use display names defined on server (note: if names are changed on server side, the Exocet operation could be affected).
- **List of data to import (max 100):**
 - **IdType:** OPC-UA *identifierType*
 - **Id:** OPC-UA *identifier*, that identify a Node in the AddressSpace
 - **Out name:** If *Custom* naming convention, name of output Manta variable
 - **Period:** If *OnRequest* policy, period of request in milliseconds

Inputs

Inputs 0: Read request

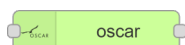
Name	Type	Description
IdType	Int	OPC-UA <i>identifierType</i> . 0: numeric; 1: string

Name	Type	Description
Id	Int or String	OPC-UA <i>identifier</i>
VarName	String	Manta variable name to use (<i>Id</i> value is used if empty or not present)

Outputs

- **data:** read data

11.58 oscar



Description

This Manta box imports data from Oscar sensor.

Properties

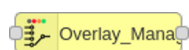
- **Name:** Manta box name
- **IP:** Enter the IP address of the OSCAR processing unit.
- **Port:** Enter the port of the OSCAR processing unit
- **Detection distance:** objects detected beyond this distance are ignored

Outputs

- **Main:** Decoded data, one object at a time
- **Auxiliary:** Decoded data, all objects simultaneously

Data Name	Type	Unit	Description
NbObjectsProc	Int		Number of objects detected by OSCAR for which the distance is below the threshold set by the user
MaxDetectionCount	Int		Number of frames for which the object has been tracked continuously
TrackerId	Int		ID of the target
Distance	Float	m	Euclidean distance measured from bow
Angle	Float	°	Bearing of the object from the bow
Size	Float	m	Estimated size of the object
Velocity_x	Float	m/s	Velocity X in SNAME (forward/starboard/downward)
Velocity_y	Float	m/s	Velocity Y in SNAME (forward/starboard/downward)
Velocity_z	Float	m/s	Velocity Z in SNAME (forward/starboard/downward)

11.59 overlay_manager



Description

This Manta box handles overlay rules cohabitation and communicates the active overlay rule offset to the PilotOverlay box.

Each overlay rule has a number for ID (*N*).

Two different types of overlay rules are managed: **Safety** and **Performance**

Safety rules

A safety rule prevents a monitored variable from exceeding a threshold value. Below the threshold, the rule is in standby mode and no action is done. When the threshold is exceeded, an emergency process

is triggered. When triggered, it preempts any lower priority safety rule and any active performance rule. Multiple safety rules can run at the same time. Overlay rules with the lowest ID have the highest priority.

When activated a safety rule is in one of these following states: * Standby: Monitoring variable below the threshold value. The overlay rule can be triggered. * Triggered (active): The overlay rule calculates an offset which is sent to the autopilot. * Preempted: Higher priority safety rule is triggered. The overlay rule can't be triggered.

Performance rules

A performance rule modifies autopilot behavior in parallel to normal guidance by applying a positive or negative offset to the user target. Only one performance rule can be active at a given time. When activated a performance rule is in one of these following states: * Active : The overlay rule sends offset to the autopilot. * Preempted : A safety rule is triggered. The overlay rule is paused and automatically switches to active state when the security rule is no longer triggered.

Properties

- **Name:** Manta box name
- **List of overlay rules to manage** (10 maximum):
 - **Type:** Overlay rule type (*Safety* or *Performance*)

Inputs

- **PilotOverlay_Status** : Must be connected to the “Status” port of the PilotOverlay box.
- **OverlayN_button** (*Boolean*): User activation/deactivation of overlay rule *N*. Any variable name is accepted. Can be connected to a “push button” or “gpio_in” manta box.
- **OverlayN_Control** : Must be connected to the “Control” port of the overlay rule *N* box.
 - **Offset** (*Float, Degree*): Overlay rule offset.
 - **RequiredState** (*String*): Internal state received from overlay rule.

Outputs

- **PilotOverlay_Control** : Must be connected to the “Control” port of the PilotOverlay box.
 - **EngagedOverlay** (*Bool*): To engage/disengage pilot overlay functionality.
 - **Offset** (*Float, Degree*): Active overlay rule offset.
- **Status** :

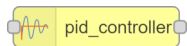
- **EngagedOverlayNumber** (*Integer*): Number of the current engaged overlay rule. It is the rule that sends offset to the pilot. “0” if no overlay rule is engaged.
 - **SafetyTriggered** (*Bool*): “True” if a Safety rule is triggered, “False” otherwise.
 - **ActivationMsg** (*String*): To notify that activation failed if activation condition not respected.
 - **OverlayNState** (*String*): Overlay rule *N* state indicator color.
 - **OverlayNActivationFailed** (*Bool*): “True” once if the activation of the overlay rule *N* failed.
- **OverlayN_State** : Must be connected to the “State” port of the overlay rule *N* box.
 - **ManagedState** (*String*): To control overlay rule state.

Notes

Overlay state indicator color convention :

- * **Safety rule:** * gray = Off * orange = Activated and preempted by a higher priority safety rule. * green = Activated in Standby. * red = Activated and triggered. It is the rule that sends offset to the pilot.
- * **Performance rule:** * gray = Off * orange = Activated and preempted by a safety rule. * green = Active. It is the rule that sends offset to the pilot.

11.60 pid



Description

Proportional Integral Derivative controller.

Properties

- **Name** Name of the box.
- **Frequency** Update frequency of the PID in Hertz.
- **Type** Chosen formulation of the PID. See section **PID Type** below.
- **P,I,D** Check the actions to enable
- **Derivative action** For each of derivative signal (reference and feedback), you can use either an external input (“*explicit input*”), or compute them internally from the reference and feedback using a first order numerical derivation.

- **Filter derivative feedback** Check this to filter the derivative feedback. Only if **Derivative action** is set to *Get derivative feedback from explicit input*. See section **Derivative feedback filter**. It is recommended to use this filter.
- **Derivative filter coefficient N** Ratio between the derivative filter time constant and τ_d . Only if **Filter derivative feedback** is enabled. It is recommended to have $3 < N < 15$.
- **Coefficients** Coefficients of the PID. See section **PID Type** below.

PID Type	Name	Description
Parallel PID	Kp	Gain applied to the error
	Ki	Gain applied to the integral of the error
	Kd	Gain applied to the derivative error
Standard PID	Ks	Gain applied to sum of all actions
	Ti	Inverse of the gain applied to the integral of the error
	Td	Gain applied to the derivative error
Zero-placement PID	Kz	Minimum gain of the PID transfer function when proportional action is enabled.
	τ_i	Time constant of the first zero
	τ_d	Time constant of the second zero

- **Enable deadband** Check this in order to cut-off PID updates when errors are small. See section **Deadband**.
- **Error deadband** Min/max of the deadband on error.
- **Derivative error deadband** Min/max of the deadband on derivative error.
- **Limit output** Check this to saturate the output. Can be disabled only if **Anti-windup method** is not *Clamping*.
- **Output bounds** Min/max of the output.
- **Anti-windup method** Chosen method for anti-windup. See section **Anti-windup**.
- **Autoset back-calculation** Check this to set automatically the back-calculation loop dynamics. See section **Anti-windup**.
- **Back-calculation gain Kb** Gain of the back-calculation loop.
- **Enable feedforward** Check this to input a feedforward signal to add to the PID output. The feedforward is taken into account by the PID to limit the output.
- **Control gains with**

- *raw values*: control gains with their raw values.
- *levels*: use an integer between 1 and 9 to control the intensity of the P, I and D actions. See section **Gain control** for more details.
- **Parameters from input** Check this to add an input to control the parameters.
- **WebApp** Check this to control the parameters from the dashboard.
- **Export parameters** Check this to output the parameters.

Inputs

Name	Data	Description
Reference	<i>Reference</i>	Reference value. Zero is assumed if not fed.
	<i>Derivative reference</i>	Derivative reference value. Zero is assumed if not fed. Only if Derivative action is set to <i>Get derivative reference from explicit input</i> .
Feedback	<i>Feedback</i>	Feedback value.
	<i>Derivative feedback</i>	Derivative feedback value. Only if Derivative action is set to <i>Get derivative feedback from explicit input</i> .
Command feedback	<i>Command feedback</i>	Feedback command from the actuator (e.g. angle of a rudder). The command feedback is used for backcalculation and to initialize the integral term when EnableI is set to true.
Parameters	EnableP	True to enable P action
	EnableI	True to enable I action
	EnabledD	True to enable D action
	SetIntegral	True to force integral action to IntegralValue
	IntegralValue	Value used to set the integral action if SetIntegral is true
	Kp	Gain applied to the error (Parallel PID only)
	Ki	Gain applied to the integral of the error (Parallel PID only)
	Kd	Gain applied to the derivative error (Parallel PID only)

Name	Data	Description
	Ks	Gain applied to sum of all actions (Standard PID only)
	Ti	Inverse of the gain applied to the integral of the error (Standard PID only)
	Td	Gain applied to the derivative error (Standard PID only)
	Kz	Minimum gain of the PID transfer function when proportional action is enabled. (Zero-placement PID only)
	Tau _i	Time constant of the first zero (Zero-placement PID only)
	Tau _d	Time constant of the second zero (Zero-placement PID only)
	PLevel	Proportional action level
	ILevel	Integral action level
	DLevel	Derivative action level
	N	Ratio between the derivative filter time constant and τ_d
	Kb	Gain of the back-calculation loop
	ErrMin	Minimum of the error deadband.
	ErrMax	Maximum of the error deadband.
	ErrDotMin	Minimum of the derivative error deadband.
	ErrDotMax	Maximum of the derivative error deadband.
	OutMin	Command minimum.
	OutMax	Command maximum.
Reset	Reset	Reset the parameters to their initial value.
Feedforward	<i>Feedforward</i>	Extra signal added to the PID output. The feedforward is considered to limit the output of the PID.

Outputs

Name	Data	Description
Command	<i>Command</i>	PID output. Command for the actuator position.
Status	Error	Error
	DerivativeError	Derivative error
	ProportionalAction	Proportional action
	DerivativeAction	Derivative action
	IntegralAction	Integral action
Parameters	EnableP	True to enable P action
	EnableI	True to enable I action
	EnableD	True to enable D action
	StopIntegration	True to stop integration
	Kp	Gain applied to the error (Parallel PID only)
	Ki	Gain applied to the integral of the error (Parallel PID only)
	Kd	Gain applied to the derivative error (Parallel PID only)
	Ks	Gain applied to sum of all actions (Standard PID only)
	Ti	Inverse of the gain applied to the integral of the error (Standard PID only)
	Td	Gain applied to the derivative error (Standard PID only)
	Kz	Minimum gain of the PID transfer function when proportional action is enabled. (Zero-placement PID only)
	TauI	Time constant of the first zero (Zero-placement PID only)
	TauD	Time constant of the second zero (Zero-placement PID only)
	PLevel	Proportional action level

Name	Data	Description
	ILevel	Integral action level
	DLevel	Derivative action level
	N	Ratio between the derivative filter time constant and τ_d
	Kb	Gain of the back-calculation loop
	ErrMin	Minimum of the error deadband.
	ErrMax	Maximum of the error deadband.
	ErrDotMin	Minimum of the derivative error deadband.
	ErrDotMax	Maximum of the derivative error deadband.
	OutMin	Command minimum.
	OutMax	Command maximum.

Documentation

PID Type

Parallel PID The parallel PID has the following equation:

$$u = K_p \cdot e_p + \int_0^t K_i \cdot e_p \cdot dt + K_d \cdot e_v$$

where:

- e_p is the error
- e_v is the derivative error
- u is the PID output

Standard PID The standard PID has the following equation:

$$u = K_s \cdot (e_p + \int_0^t \frac{1}{T_i} \cdot e_p \cdot dt + T_d \cdot e_v)$$

- T_i can be interpreted as the duration required to eliminate the mean of all past errors.
- T_v can be interpreted as the prediction time on future errors.

Zero-placement PID The zero-placement PID is defined by the value of its two zeros and its minimum gain. For the P, I, PI, and PD, the equation is as follows, removing the unused terms:

$$u = K_z(e_p + \int_0^t \frac{1}{\tau_i} \cdot e_p \cdot dt + \tau_d \cdot e_v)$$

For the PID, the equation is:

$$u = K_z(\frac{\tau_i + \tau_d}{\tau_i} e_p + \int_0^t \frac{1}{\tau_i} \cdot e_p \cdot dt + \tau_d \cdot e_v)$$

The corresponding frequency response is:

Zero-placement PID frequency response

- Errors having a frequency below $1/\tau_i$ rad/s are eliminated by the integral action.
- Errors having a frequency over $1/\tau_d$ rad/s are anticipated.

Deadband The deadband is defined as the intersection of a error between [pmin, pmax] and a derivative error between [vmin,vmax]. The control is disabled in the deadband, thus when the error and the derivative error **are both** in the deadband intervals. The deadband on derivative error is enabled only if the derivative action is enabled. The control output is continuous when entering and leaving the deadband.

Anti-windup If the PID output is bounded using **Limit output**, when the controller reaches the actuator limits, the integral term accumulates a significant error and becomes really large. This is called windup. A controller windup can cause a significant delay when the error changes its sign before the PID output leaves its bound. Using the anti-windup prevents the controller to windup. Two methods are proposed:

Clamping The clamping anti-windup method consists in stopping the integration of the error when the output has reached a bound. This option can be selectable only if **Limit output** is selected. This method is recommended when only an internal saturation is used.

Back-calculation The back-calculation method consists in unwinding the integral of the error when the output has reached a bound. The following block-diagram illustrates the back-calculation loop.

Back-calculation loop with external input

Back-calculation loop with internal input

Back-calculation gain can be automatically set using the option **Autoset back-calculation**.

Back-calculation can be convenient for dynamic systems with large delays. Back-calculation can also be used to track the actuator target and therefore ensure a bumpless controller switch.

Gain control Gain control can be done with levels using the option **Control gains with levels**. When this options is set. The input or dashboard variables PLevel, ILevel and DLevel can be set between 1 and 9. A coefficient corresponds to each level:

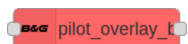
Level	1	2	3	4	5	6	7	8	9
Coefficient	0.41	0.51	0.64	0.80	1.00	1.25	1.56	1.95	2.44

A level increment increases the corresponding action (P, I or D) by 25%. These coefficients impact the following variables:

	PLevel +1	ILevel +1	DLevel +1
Parallel PID	Kp +25%	Ki +25%	Kd +25%
Standard PID	Ks +25%	Ti -25%	Td +25%
Zero-placement PID	Kz +25%	τ_i -25%	τ_d +25%

Input timeout management If the **Feedback** input or **Actuator position** input have timed out, the PID controller stops updating and its state switches to *WARNING*. Once the timed-out input is eventually refreshed, the PID resumes with no discontinuity.

11.61 pilot_overlay_bandg



Description

Ease of use B&G H5000 PILOT interface :

- Virtual Pilot Controller (on dashboard with button widgets)
- Pilot overlay for Safety or performance purposes :

- Add an external “Offset” to the pilot target (WindReference or HeadingToSteer), set by the skipper: $CorrectedTarget = UserTarget + Offset$
- Works in “TrueWind” mode and “Heading” mode. “ApparentWind”, “No Drift” and “Navigation” modes are not taking into account. This box must not be used with an H5000 PILOT in one of these modes.
- Disengages automatically when H5000 PILOT disengage, change mode, port/starboard 10° order, tack/gybe order and if lost communication with the autopilot.
- Disengages on “EngageOverlay” falling edges.
- When disengage, the H5000 PILOT keeps its last target value (“FeedbackTarget”)
- Engages on “EngageOverlay” rising edges if engagement conditions are respected.
- Engagement conditions :
 - * H5000 PILOT connected, in TrueWind or Heading mode
 - * “Offset” received

Properties

- **Name:** Manta box name
- **Offset:** Mapping name of the offset
- **EngageOverlay:** Mapping name of the engage control signal
- **Offset MaxVariation:** If the variation between two successive Offset is above, ignore the last Offset value.
- **UserTarget MaxVariation:** If the variation between two successive UserTarget is above, ignore the last UserTarget value.

Inputs

- **Control:** Pilot Overlay control inputs

Variable	Type	Unit	Description
Offset	Float	Degrees	Heading or TrueWind angle offset
EngageOverlay	Bool		Engage/Disengage Pilot Overlay functionality

- **65305-SIMRAD Modes/Heartbeat:** Must be connected to a “Pilot Controller net2000_in” Box with “Address Filter” to “All”
- **130850-SIMRAD Event Command:** Must be connected to a “Pilot Controller net2000_in” Box

with “Address Filter” to “All”

- **65341-SIMRAD Autopilot Sailing Mode:** Must be connected to an “Autopilot net2000_in” Box with “Address Filter” to “All”
- **127237-Heading Track Control:** Must be connected to an “Autopilot net2000_in” Box with “Address Filter” to “All”
- **Button <10:** May be connected to a button box (periodicity = None)
- **Button 10>:** May be connected to a button box (periodicity = None)
- **Button <1:** May be connected to a button box (periodicity = None)
- **Button 1>:** May be connected to a button box (periodicity = None)
- **Button STBY:** May be connected to a button box (periodicity = None)
- **Button AUTO:** May be connected to a button box (periodicity = None)
- **Button MODE:** May be connected to a button box (periodicity = None)

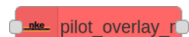
Outputs

- **65305-SIMRAD Modes:** Must be connected to a PilotControllerOut Box with “Address Filter” to “All”
- **130845-SIMRAD Parameter Handle:** Must be connected to a PilotControllerOut Box with “Address Filter” to “All”
- **130850-SIMRAD Event Mode:** Must be connected to a PilotControllerOut Box with “Address Filter” to “All”
- **130850-SIMRAD Event ChangeRef:** Must be connected to a PilotControllerOut Box with “Address Filter” to “All”
- **Status:**

Variable	Type	Unit	Description
PilotEngaged	Bool		“True” / “False”
Mode	Text		“Stop”, “Heading”, “TrueWind”
PilotMode	Int		0 = “Stop”, 1 = “Heading”, 2 = TrueWind”
PilotSuperModeOn	Bool		Always “False”, because B&G H5000 Pilot doesn’t have super modes.
UserTarget	Float	Degrees	Target set by the skipper. Can be change by +/-1° when pilot overlay engaged

Variable	Type	Unit	Description
CorrectedTarget	Float	Degrees	"UserTarget" + "Offset"
FeedbackTarget	Float	Degrees	Actual H5000 PILOT TrueWind or Heading target
OverlayEngaged	Bool		"True" / "False"
OverlayDisengageEvent	Text		"None", "TackGybe", "Port10", "Starboard10", "ChangeMode", "PilotDisengage", "EngageOverlayInput", "PilotComTimeout"
PilotComTimeout	Bool		"True" if one timeout on the input ports coming from "Autopilot In" box is triggered
InvalidOffsetCount	Int		Number of Offset discarded due to too large variation
InvalidWindTargetCount	Int		Number of WindTarget discarded due to too large variation
InvalidHeadingTargetCount	Int		Number of HeadingTarget discarded due to too large variation

11.62 pilot_overlay_nke



Description

Ease of use NKE PILOT interface :

- Virtual Pilot Controller (on dashboard with button widgets)
- Pilot overlay for safety or performance purposes :
 - Add an external "Offset" to the pilot target (Wind or Heading), set by the skipper: $CorrectedTarget = UserTarget + Offset$
 - Works in "TrueWind", "ApparentWind", and "Heading" mode. "Polar" and "Rudder" modes are not taking into account. This box must not be used with a PILOT in one of these modes.

- Disengages automatically when PILOT disengage, change mode, active super mode order, port/starboard 10° order and tack/gybe order.
- Disengages on “EngageOverlay” falling edges.
- When disengage, the PILOT keeps its last target value (“FeedbackTarget”)
- Engages on “EngageOverlay” rising edges if engagement conditions are respected.
- Engagement conditions :
 - * PILOT connected, in TrueWind, ApparentWind or Heading mode
 - * “Offset” received
 - * NKE “Super modes” Off

Properties

- **Name:** Manta box name
- **Offset:** Mapping name of the offset
- **EngageOverlay:** Mapping name of the engage control signal
- **Offset MaxVariation:** If the variation between two successive Offset is above, ignore the last Offset value.
- **UserTarget MaxVariation:** If the variation between two successive UserTarget is above, ignore the last UserTarget value.

Inputs

- **Control:** Pilot Overlay rules inputs (must be sent periodically)

Variable	Type	Unit	Description
Offset	Float	Degrees	Heading or Wind angle offset
EngageOverlay	Bool		Engage/Disengage Pilot Overlay functionality

- **SailNet:** Must be connected to a SailNet_in box correctly configured on Processor HR. Waiting for variables *PilotRef*, *ModePilHR* and *StatusPilHR*.
- **Feedback:** Must be connected to a udp_in box correctly configured to receive NKE Ethernet Box.
- **Button <10:** May be connected to a button box (periodicity = None)
- **Button 10>:** May be connected to a button box (periodicity = None)
- **Button <1:** May be connected to a button box (periodicity = None)

- **Button 1>**: May be connected to a button box (periodicity = None)
- **Button STOP**: May be connected to a button box (periodicity = None)
- **Button AUTO**: May be connected to a button box (periodicity = None)
- **MODE**: Pilot mode control input for update from the Manta graph (external control)

Outputs:

- **Serial out**: Must be connected to a udp_out box correctly configured to send to NKE Ethernet Box.
- **Status**:

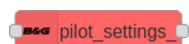
Variable	Type	Unit	Description
PilotEngaged	Bool		“True” / “False”
Mode	Text		“Stop”, “Heading”, “TrueWind”, “ApparentWind”
PilotMode	Int		1 = “Heading”, 2 = TrueWind“, 0 = pilot no engaged or other pilotMode PilotSuperModeOn Bool ”True” if at least on NKE pilot super mode is on. “False” otherwise.
UserTarget	Float	Degrees	Target set by the skipper. Can be change by +/-1° when pilot overlay engaged
CorrectedTarget	Float	Degrees	“UserTarget” + “Offset”
FeedbackTarget	Float	Degrees	Actual PILOT target (Wind or Heading)
OverlayEngaged	Bool		“True” / “False”
OverlayDisengageEvent	Text		“None”, “TackGybe”, “Port10”, “Starboard10”, “ChangeMode”, “PilotDisengage”, “EngageOverlayInput”, “PilotSuperModeOn”
PilotComTimeout	Bool		Always “false”. No communication timeout detection with NKE PILOT
InvalidOffsetCount	Int		Number of Offset discarded due to too large variation
InvalidUserTargetCount	Int		Number of UserTarget discarded due to too large variation

- **Super Modes:** State of the NKE pilot “Super modes” : Boost, Gust, PilotTable, Surf.

WebApp controls

- **PilotModeNKE** (*int*): On your dashboard, use a “Slider” widget to manually set the pilot mode. It is synchronised with the NKE Processor HR pilot mode.

11.63 pilot_settings_bandg



Description

Ease of use B&G H5000 PILOT interface

Properties

- **Name:** Manta box name

Inputs

- **PGN 127237:** Must be connected to port 127237 of a net2000 in box with type set as Autopilot
- **PGN 130845:** Must be connected to port 130845 of a net2000 in box with type set as Autopilot
- **Ext control:** Pilot Settings control input for update from the Manta graph

Variable	Type	Unit	Comment
RudderGain	Float		
RudderLimit	Float	deg	
CounterRudder	Float	s	
CruisingSpeed	Float	kn	
ManualSpeed	Float	kn	
BoatLength	Float	m	
PerformanceMode	Int		

Variable	Type	Unit	Comment
AutoTrim	Float	s	
TackAngle	Float	deg	
TackTime	Float	s	
WindMode	Int		1=Apparent, 2=True, 4=Auto
HeelCompensation	Bool		
GustResponse	Bool		
TwsResponse	Bool		
SteeringAutomaticResponse	Int		0=Off, 1=Economy, 2=Normal, 3=Sport
SteeringRecovery	Int		0=Off, 1=Narrow, 2=Medium, 3=Wide
Adapt	Bool		
OffCourse	Float	deg	
LowBoatSpeed	Float	kn	
RudToRateRatio	Float		
RudderFeedback	Int		1=InternalFrequency, 4=External
DriveVoltage	Float	V	
MotorOutput	Int	%	
ManualDeadband	Float	deg	

Outputs

- **STATUS:** Export current Pilot settings
- **Other outputs:** Must be connected to port 130845 of a net2000 out box with type set as Autopilot

WebApp controls

On your dashboard, use “Slider”, “Set number” or “Button” widgets to change parameter values.

11.64 polar



Description

Use polar table to calculate some performance data.

Properties

- **Name:** Manta box name
- **Ext control:** If checked, add inputs to dynamically manage the performance table
- **Save:** If checked, save dynamic update on the file. It is strongly recommended to not periodically send a dynamic table if this option is enabled.
- **Interpolation:** Select desired interpolation method
- **File format:** Select used file format
- **List of polar tables:**
 - **Name:** Enter a table name
 - **File:** Select a performance polar table (format, see note below)

Inputs

Input 0: True Wind

Name	Units	Description
TWA	°T	True wind angle
TWS	kn	True wind speed

Input 1: Boat speed

Name	Units	Description
BoatSpeed	kn	Longitudinal boat speed, water referenced

Input 2: Ext polar table

Name	Type	Description
PolarTable	FloatTable	Dynamic polar table. Variable name is free.

Input 3: Ext table selection

Name	Type	Description
TableSelector	Scalar/String	Dynamic table selector. Variable name is free.

Outputs**Output 0: Performances**

Name	Units	Description
VMG	kn	Velocity Made Good (VMG) is the component of Boat Speed in the direction of the True Wind.
BSP_Polar	kn	Expected BoatSpeed from the polar table, at current TWA and TWS.
BSP_PolarRatio	%	Ratio of current BoatSpeed on BSP_Polar.
VMG_Polar	kn	Optimum VMG of the polar table, for current TWS and current TWA sector.
VMG_PolarRatio	%	Ratio of current VMG on VMG_Polar.
TWA_PolarTgVmg	°(+/-180)	TWA target to get the optimum VMG of the polar table, for current TWS and current TWA sector.
BSP_PolarTgVmg	kn	BoatSpeed target to get the optimum VMG of the polar table, for current TWS and current TWA sector.

Notes**VMG:**

Velocity Made Good is defined as the projection of boat speed on the wind axis.

Tables requirements:

- B&G WTP/Deckman, B&G H5000 or NKE Processor Regatta compatible format
- TWA must be $> 0^\circ$ and $\leq 180^\circ$
- TWS must be $\geq 0\text{kn}$ and $\leq 100\text{kn}$
- BSP must be $\geq 0\text{kn}$ and $\leq 100\text{kn}$

Calculations above boundaries:

- The spline surface interpolation is constrained in order to have a null boat speed at $\text{TWA}=0^\circ$
- The spline surface interpolation is constrained in order to continue with the same boat speed at $\text{TWA}=180^\circ$
- For TWS below the first TWS : return values corresponding to the first TWS
- For TWS above the last TWS : return values corresponding to the last TWS

Spline surface limitations:

- The 3D spline interpolation can give bumps if 2 different points are too close and if the TWS/TWA are poorly distributed.
- The spline surface is smoother with only few points per TWS.

B&G WTP/Deckman file format

Size: maximum 16 pairs of BSP/TWA, maximum 20 TWS lines.

```

v1  a1  v2  a2  v3  a3  v4  a4  v5  a5
TWS BSP TWA BSP TWA BSP TWA BSP TWA BSP TWA
TWS BSP TWA BSP TWA BSP TWA BSP TWA BSP TWA
```

B&G H5000 file format

Size: 17 TWA columns, maximum 20 TWS lines.

```

TWS\TWA 20 30 40 50 60 70 80 90 100 110 120 130 140 150 160 170 180
TWS    BSP BSP BSP BSP BSP BSP BSP BSP BSP BSP BSP BSP BSP BSP BSP BSP
TWS    BSP BSP BSP BSP BSP BSP BSP BSP BSP BSP BSP BSP BSP BSP BSP BSP
```

NKE Processor Regatta file format

Size: 10 TWS columns, 14 TWA lines.

```

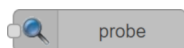
TWS TWS TWS TWS TWS TWS TWS TWS TWS TWS
TWA BSP BSP BSP BSP BSP BSP BSP BSP BSP BSP
TWA BSP BSP BSP BSP BSP BSP BSP BSP BSP BSP
```

Expedition / SailingPerformance file format

Size: maximum 16 pairs of TWA/BSP, maximum 20 TWS lines.

```
TWA BSP TWA BSP TWA BSP TWA BSP TWA BSP
TWS TWA BSP TWA BSP TWA BSP TWA BSP TWA BSP
TWS TWA BSP TWA BSP TWA BSP TWA BSP TWA BSP
```

11.65 probe



Description

A probe displays data exported by boxes linked with it.

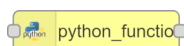
Properties

- **Name:** Manta box name
- **Full speed:** If checked, every data value changes will be reported. If unchecked, data is refreshed every 300 ms. **IMPORTANT NOTE:** Using **Full speed** can impact Exocet CPU performances if data to report changes frequently. It is recommended to use this option only during setup/debug phase and not in production.
- **History:** The “Show/Hide History” button is used to switch between history/current modes. When in history mode, the most recent data is displayed first. Only 1000 data values are displayed to avoid browser performance issues.
- **Pause:** The “pause/play” button is used to stop refreshing data values.
- **Data download:** In pause mode, two buttons are available to download displayed data as JSON or CSV format file. For CSV, data separator is space.

Inputs

- Data stream to monitor. Multiple wires can be connected.

11.66 python_function



Description

A function block where you can write python code to do interesting things.

- Python 2.7 code is supported and the NumPy library as well.
- The Python code is run each time a data is received by an input connector. The *Tick* Manta box can be used to periodically execute the Python code.
- Each Box holds its own local context. This local context is preserved accross each script invocation.
- Python variables can be shared accross boxes by using the *global* keyword, but its use is strongly discouraged since all python boxes of Manta could be impacted.
- To use an external variable inside a function, prefer to use parameters

Inputs

- Multiple inputs can be defined by using the *Number of Inputs* widget at the bottom.
- The incoming data is accessible from the *msg* Python object. *msg* is a dictionary with variable names as keys.
- The *port* Python variable holds the input connector index on which the data arrives. It is used to know which input connector has triggered the Python script.

Outputs

- Multiple outputs can be defined by using the *Number of Outputs* widget at the bottom.
- The Python script must define a *rep* Python array to export data out of the box. The *rep* array size must match the number of output connectors.
- Each *rep* element is either a dictionary, either *None*. Each element represents the set of data to export to a specific output connector on script completion. If an element is a dictionary, variables will be exported on the connector with same index as the element. If an element is *None*, no data will be exported on the corresponding output connector.
- Depending on the Python variable type, *Outputs* data type can be: Boolean, Integer, Float, String, CanFrame, Opaque (max size 2048), FloatArray (max size 1024) or FloatTable (max size 32 x 32).

Notes

Initialize a variable at first script invocation

In order to initialize or keep a variable value between all the script invocations, you can use that:

```
try:
    X
except NameError:
    X = 0.0
...
```

Hello World exemple

In this exemple, a data named “hello” with String value “Hello World!” is sent to output connector 0 at each script invocation;

```
out={}
out["hello"] = "Hello World!"
rep = [ out ]
```

Handling Input connectors

In this exemple:

- The Python box is assumed to be configured with several input connectors and a single output.
- If script is triggered by input connector 0, incoming data are forwarded to output connector.
- If script is triggered by another input connector than 0, no data is exported.

```
if port==0:
    out = dict(msg)
else:
    out = None
```

```
rep = [ out ]
```

Handling Input data

In the example below:

- The function block is assumed to be configured with 2 output connectors.
- The incoming data *msg* is assumed to have three fields: *TWA*, *TWS*, and *BSP*.
- The *rep* variable will output the *out1* object to connector 1, and the *out2* object to connector 2. Thus connector 2 will pass a variable with a single *Debug* field, whereas connector 1 will have the newly calculated *TWS* and *BSP*.

```
###reads incoming data fields of msg
```

```
trueWindAngle = msg["TWA"]
trueWindSpeed = msg["TWS"]
```



```
boatSpeed = msg["BSP"]

apparentWindAngle = msg["AWA"]
apparentWindSpeed = msg["AWS"]

out1={}
out2={}

bsp = math.sin(apparentWindAngle)*apparentWindSpeed

out1["TWS"] = math.cos(trueWindAngle)*0.6*trueWindSpeed
out1["BSP"] = bsp

if bsp > 35:
    out2["Debug"] = "Too fast"
elif (bsp < 1.0):
    out2["Debug"] = "Too slow"
else:
    out2["Debug"] = "Cool"

rep = [ out1 , out2 ]
```

Handling Units

A data can be published with a unit. The data must be encapsulated in a list with its unit in a string format (upper case) at first position.

```
out["MyConsumptionMessage"] = ["WATT", 1.3]

rep = [ out ]
```

Current available units are: UNSET, AMPEREHOURS, AMPERES, ANGULAR_DEGREES, ANGULAR_DEGREES_180, ANGULAR_DEGREES_360, ANGULAR_DEGREES_PER_SECOND, ANGULAR_MAGNETIC_DEGREES, ANGULAR_TRUE_DEGREES, BAR, BITFIELD, CANDELA, CELSIUS, CUBIC_METERS, DATE, DECIBELS, FARENHEIT, FEET, GRAVITY_ACCELERATION, HERTZ, HOURS, HPA, HUMIDITY, JOULE, KELVIN, KILOBYTES, KILOBYTES_PER_SECOND, KILOGRAMS, KILOGRAMS_PER_CUBIC_METER, KILOMETERS_PER_HOUR, KILOWATTS, LITERS, LITERS_PER_SECOND, MAGNETICFIELD, MEGABYTES, MEGABYTES_PER_SECOND, METERS, METERS_PER_SECOND, METERS_PER_SQUARE_SECOND, MICROSTRAIN, MILLIAMPERES, MILLIBAR, MILLIMETERS, MILLISECONDS, MILLIVOLTS, MILLIVOLTS_PER_VOLT, MILLIWATTS, MINUTES, MOLE, NANOMETERS, NAUTICAL_KNOTS, NAUTI-

CAL_MILES, NEWTON, NEWTONMETERS, PASCAL, PERCENT, PPM, RADIANS, RADIANS_PER_SECOND, RPM, SALINITY, SECONDS, TESLA, TIME, TONNES, VOLTS, WATT.

Handling ByteArray

bytearrays are a kind of data, that can come from *UDP*, *Serial*, or other Python box. Bytearray data outgoing from the Python Box is converted to the Opaque data type.

```
out = {}
```

```
s = "Very interesting !"
b=bytearray(s)
out["data"] = b
rep = [ out ]
```

That leads to an output message containing the string “Very interesting” as an array.

Display HTML or icons in dashboards

Output data must be in JSON format, converted to the Opaque data type. JSON keys are : * type : *icon* or *text* * icon : string containing icons class names. Font-Awesome is used. The first fa class is added automatically. Default-icon is fa-circle. * text : HTML text * color (optional) : color of text or icon. Color can be HTML color name or HEX value with # prefix. See Web colors.

```
out = {}
```

```
value = msg['Scalar']
icon = 'fa-circle'
if value == 0:
    color = '#c0c0c0'
    text = '<h3>Stopped</h3>'
elif value <7:
    color = 'green'
    text = '<h3>Running</h3>'
elif value < 9:
    color = 'orange'
    text = '<h2>Warning</h2>'
else:
    color = 'red'
    icon = 'fa-times-circle'
    text = '<h1>STOP !</h1>'
```

```
### export values in byte arrays
```

```
out["icon"] = bytearray('{"type":"icon","icon":"' + icon + ' fa-2x","color":"' +  
out["text"] = bytearray('{"type":"text","text":"' + text + '"}')
```

```
rep = [ out ]
```

Outputs are automatically displayed as icons or HTML in dashboard data widgets.

Handling CAN Frames

When a CAN box is connected to an input of the python box, the incoming `msg["can"]` is special. It is of `pixel.CanFrame` type

The `pixel.CanFrame` class has 2 attributes:

- `id` : the CAN Id
- `bytes`: a `ByteArray` of the CAN data (up to 8 bytes)

Example:

In this example, regardless of the incoming CAN frame ID, the 2 first bytes of the frame are modified, and frame with ID “0x457” is emitted.

```
### Handle input message
```

```
canmsg = msg["can"]
```

```
###Is CAN frame?
```

```
try:
```

```
    if type(canmsg) is not pixel.CanFrame:
```

```
        raise TypeError('no canframe')
```

```
    isCAN = True
```

```
except:
```

```
    isCAN = False
```

```
###Process
```

```
if isCAN:
```

```
    expectedId = 0x08FF0DFC
```

```
    # Test on 29 bits identifier (can 2.0b)
```

```
    if ( (canmsg.id & 0x1FFFFFFF) == expectedId):
```

```
        MSG = pixel.CanFrame(canmsg.id, canmsg.bytes)
```

```
    else:
```

```
        MSG = pixel.CanFrame(0x08FFFF00, bytearray([0x00, 0x00]))
```

```
else:
```

```
    MSG = None
```

```
### emits the result on the output connector
```

```
rep = [ MSG ]
```

Playing with bytearray and deque FIFO

In the following example, all incoming bytes are first stored in a deque FIFO (First In, First Out) buffer, and are post-processed later. Notice that the deque buffer is declared as a persistent variable.

The data is added to the deque at the right end through *append* and popped from the left end through *popleft*.

```
from collections import deque
```

```
try:
```

```
    buf
```

```
    height
```

```
except NameError:
```

```
    buf = deque()
```

```
    height = 0.0
```

```
def isBeginOfFrame(b):
```

```
    if ((b & 0x80) == 0x00 and (b & 0x7f) == 0x48):
```

```
        return True
```

```
    return False
```

```
### First, store incoming data at end of deque buffer
```

```
data = msg["data"]
```

```
for val in data:
```

```
    buf.append(val)
```

```
### Process
```

```
while True:
```

```
    if (len(buf) < 3):
```

```
        break
```

```
    b = buf.popleft()
```

```
    if (isBeginOfFrame(b)):
```

```
        b2=buf.popleft()
```

```

        b3=buf.popleft()
        index = b2 & 0x7f + ((b3 & 0x7f) << 7)
        height = index * 0.05 * 1000

out = {}
out["height"] = height
rep = [ out ]

```

Advanced numerical computation : The kalman filter

Use manta, python and graphs to simulate the response of your filters and design advanced low latency estimators like :

- Dynamic filter (True Wind damping)
- Sensors hybridation (Surface speed, Ride height)
- Estimate a constant, a position

Simulate a noisy signal:

The simulated measurement is generated with the uncertainty knowledge.

```

### Noisy sinus generator with outliers - may 9, 2019 - v1.0
###
### IN0 : Must be connected to a tick (Typ 25Hz)
### OUT0 : 'Signal' : Simulate a measurement (A noisy sinus with glitches)
###       'Covariance' : Covariance of the measurement
###
### Copyright © 2019 Pixel Sur Mer

import random

try:
    init
except NameError:
    init = True
    t=0.0
    cmpt=0
    out = {}

t=t+0.04
cmpt=cmpt+1

```

```

if cmpt>50:
    cmpt=0
    offset=random.uniform(-50.0, 50.0)
else:
    offset=0.0
noise=random.gauss(0, 2)+offset          # gaussian noise + glitches
out["Signal"] = 10.0*math.sin(t)+noise
out["Covariance"] = noise**2              # In this case noise mean=0 -> covariance
rep = [ out ]

```

Dynamic Kalman filter

The uncertainty of the measurement is used to reject noises. This is a generic implementation of a Kalman filter, you can use this template for another use of the kalman filter.

```

### Implementation of a dynamic Kalman Filter - may 9, 2019 - v1.0
###
### Matrices must be set up for your system
### Notation : https://en.wikipedia.org/wiki/Kalman\_filter
### Basic tutorial : https://www.kalmanfilter.net/default.aspx
###
### IN0 : 'Signal' :      Measurement (fix frequency)
###       'Covariance' : Optional covariance of this measurement
### OUT0 : 'KF_Predict' : Kalman predicted output
###
### Copyright © 2019 Pixel Sur Mer

```

```
import numpy as np
```

```
global np
```

```

class Kalman(object):
    def __init__(self, F = None, B = None, H = None, Q = None, R = None, P = None):
        if (F is None or H is None):
            raise ValueError("Set proper system dynamics.")
        self.n = F.shape[1]
        self.m = H.shape[1]
        self.F = F
        self.H = H
        self.B = 0 if B is None else B
        self.Q = np.eye(self.n) if Q is None else Q

```

```

self.R = np.eye(self.n) if R is None else R
self.P = np.eye(self.n) if P is None else P
self.x = np.zeros((self.n, 1)) if x0 is None else x0

def predict(self, u = 0):
    self.x = np.dot(self.F, self.x) + np.dot(self.B, u)
    self.P = np.dot(np.dot(self.F, self.P), self.F.T) + self.Q
    return self.x

def update(self, z, R = None):
    self.R = np.eye(self.n) if R is None else R
    y = z - np.dot(self.H, self.x)
    S = self.R + np.dot(self.H, np.dot(self.P, self.H.T))
    K = np.dot(np.dot(self.P, self.H.T), np.linalg.inv(S))
    self.x = self.x + np.dot(K, y)
    I = np.eye(self.n)
    self.P = np.dot(np.dot(I - np.dot(K, self.H), self.P),
                    (I - np.dot(K, self.H)).T) + np.dot(np.dot(K, self.R), K.T)

try:
    init
except NameError:
    init = True
    out = {}
    dt = 1.0/60 # Time step
    cov = 0.5 # Default observation covariance
    P = [] # Prediction
    F = np.array([[1, dt, 0], [0, 1, dt], [0, 0, 1]]) # State-transition matrix
    H = np.array([1, 0, 0]).reshape(1, 3) # Observation matrix
    Q = np.array([[0.05, 0.05, 0.0], [0.05, 0.05, 0.0], [0.0, 0.0, 0.0]]) # Covariance of process noise
    R = np.array([cov]).reshape(1, 1) # Covariance of observation noise
    kf = Kalman(F = F, H = H, Q = Q, R = R)
    if "Covariance" in msg:
        cov=msg["Covariance"]
    if "Signal" in msg:
        pred=np.dot(H, kf.predict())[0] # Predicted state
        P.append(pred)
        kf.update(msg["Signal"], R=np.array([cov]).reshape(1, 1)) # Calc kalman
        out["KF_Predict"] = pred[0]

```

```
else:  
    raise ValueError("No data 'Signal'")
```

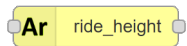
11.67 regulation_performance



Description

This box is a pilot overlay platform for implementation of multi-objective regulation/performance rule.

11.68 ride_height



Description

The purposes of this Manta Box is:

- To compute the height above the water (z axis) of points of interest, computed from some look down range sensor(s)
- The range sensor(s) shall measure a distance to the water which is pitch/roll independant => shortest distance (typical for sonic sensor)
- Up to 6 range measurement sensors can be configured. For outputs as many points can be configured where the height above water will be computed
- This Manta box is designed to work with SI unit only. (meters & deg)
- The heights outputs will have two values
 - One with extension : .csys => height in the specified coordinate system (see Properties)
 - One with extension : .disp => height always positive when point is above water plan (Z axis up)
- Compute height dynamic state to understand if height is increasing, decreasing or stable

Properties

- **global properties:** Used for each sensors:

- **Name:** Manta box name
- **Filter duration:** Sensors filter duration in time unit
- **Min range:** Sensor measurement below this value will be rejected.
- **Max range:** Sensor measurement above this value will be rejected.
- **Coord. System:** Right Hand or SNAME (Right hand pointing down).
- **Slope Lag Duration:** Backward duration to define height dynamic state
- **Height Deadband:** Deadband in wich no need to compute height dynamic state
- **Sensor Count:** Number of sensor to compute
- **Sensor configuration:**
 - Sensor variable name
 - XYZ location in meter
 - * RH : X+ => Fwd, Y+ => to Port, Z+ => Up
 - * SNAME : X+ => Fwd, Y+ => to Stbd, Z+ => Down
- **Input mapping**
 - Heel variable (RH & SNAME : positive => heel to stbd)
 - Trim variable (RH : positive => bow down. SNAME : positive => bow up)
- **Output mapping**
 - Height name: nmea of the output height
 - Height Derivative Sign name:
 - XYZ: Distance in meter of point of interest

Inputs

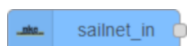
- **Input 1 :**
 - Heel variable (RH & SNAME : positive => heel to stbd)
 - Trim variable (RH : positive => bow down. SNAME : positive => bow up)
- **Other input:**
 - Sensors 1, 2, ...

Outputs

- **Output 1** As many as required output heights
 - RH : X+ => Fwd, Y+ => to Port, Z+ => Up

- SNAME : X+ => Fwd, Y+ => to Stbd, Z+ => Down
- **Output 2** height dynamic state:
 - Positive: Height increasing
 - Negative: Height decreasing
 - Zero: no change

11.69 sailnet



Description

This Manta box imports NKE systems data by using SailNet protocol. To grab data from NKE *Processor HR* or *Processor Regatta*, the SaiNet protocol must be initiaized from NKE system web interface.

From “Outils” -> “Parcourir le répertoire courant du Processor”, open “/mnt/flash/processor/instal/Instal.ini”:

- *ValidSailNet = Y*
- *SailNetOutIp = Exocet IP address*
- *SailNetOutPort = Manta box Local port*
- *SailNetInPort = Manta box Remote port*
- *Save File when done*

From “Développement à façon” -> “Paramétrage de SailNet”:

- *SailNetOutIp = Exocet IP address*
- *SailNetOutPort = Manta box Local port*
- *SailNetInPort = Manta box Remote port*
- *Save File when done*

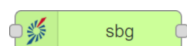
Properties

- **Name:** Manta box name
- **NKE system IP address:** Enter IP address of the NKE device.
- **Local port:** UDP port to receive data. It corresonds to *SailNetOutPort*.
- **Remote port:** NKE UDP port to send query. It corresonds to *SailNetInPort*.
- **NKE variables to export:** Select from the list the NKE variables to import. Note that the Manta graph must have been deployed a first time after adding the *NKE* box to populate the variables list.

Outputs

- The list of selected NKE variables (Integer or Decimal)

11.70 sbg



Description

This Manta box decode some SBG Inertial Navigation System binary messages.

Supported messages are: LOG_STATUS (01), LOG_UTC_TIME (02), LOG_IMU_DATA (03), LOG_EKF_EULER (06), LOG_EKF_NAV (08), LOG_SHIP_MOTION (09), LOG_GPS1_VEL (13), LOG_GPS1_POS (14), LOG_GPS1_HDT (15)

Vessel reference frame: The data provided by the Sbg Manta box use the standard SNAME's (1950) notation (forward/starboard/downward and North/East/Down frames).

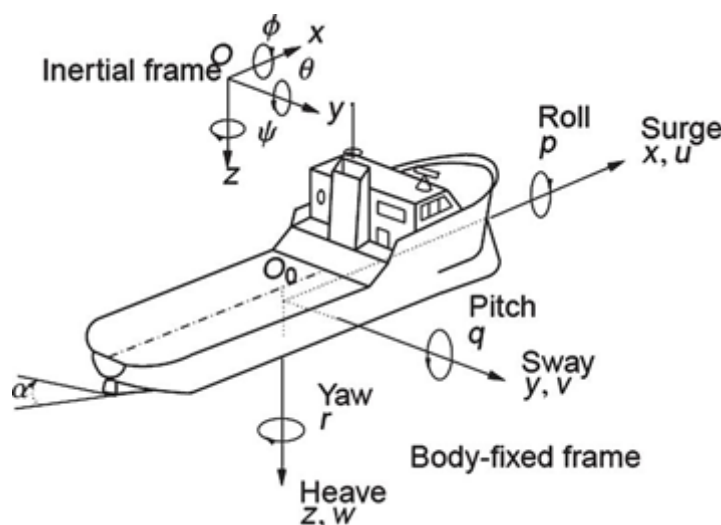


Figure 86

Properties

- **Name:** Manta box name
- **Msg ID xx:** If selected, decode this message

Inputs

- **Input:** UDP or serial stream to decode.

Outputs

- **01-LOG_STATUS:** Output data of SBG_ECOM_LOG_STATUS (01)
- **02-LOG_UTC_TIME:** Output data of SBG_ECOM_LOG_UTC_TIME (02)
- **03-LOG_IMU_DATA:** Output data of SBG_ECOM_LOG_IMU_DATA (03)
- **06-LOG_EKF_EULER:** Output data of SBG_ECOM_LOG_EKF_EULER (06)
- **08-LOG_EKF_NAV:** Output data of SBG_ECOM_LOG_EKF_NAV (08)
- **09-LOG_SHIP_MOTION:** Output data of SBG_ECOM_LOG_SHIP_MOTION (09)
- **13-LOG_GPS1_VEL:** Output data of SBG_ECOM_LOG_GPS1_VEL (13)
- **14-LOG_GPS1_POS:** Output data of SBG_ECOM_LOG_GPS1_POS (14)
- **15-LOG_GPS1_HDT:** Output data of SBG_ECOM_LOG_GPS1_HDT (15)

11.71 scalar



Description

User controlled box.

This Manta box exports a scalar data that can be changed from Exocet Web App (*Integer or Decimal*).

Properties

- **Name:** Manta box name
- **Save:** If selected, the data value is saved to be restored when Exocet is rebooted or Manta graph deployed.
- **Output Name:** Exported data name. Manta box name is used if not defined.
- **Period:** to send data value periodically. Data value is also sent at any user change. If *None* is selected, only data changes are sent.
- **Type:**
 - **Integer Range:** an Integer value within the range bordered by *min* and *max*. Any value within the range is possible.

- **List of defined Integer values:** an integer value that can takes only a set of pre defined values.
 - **Decimal Range:** a Decimal value within the range bordered by *min* and *max*. Any value within the range is possible.
 - **List of defined Decimal values:** a decimal value that can takes only a set of pre defined values.
- **Decimals:** number of decimals to display the value.
 - **Initial Value:** initial data value.
 - **Step Value:** step applied when incrementing or decrementing data from web application.
 - **Min Value:** minimum value for the data
 - **Max Value:** maximum value for the data
 - **List of defined data values:** Press *New predefined value* button to fill a new value. For each value, a *Label* can be defined to help web application support. The initial value is the first defined value.
 - **Ext control:** If selected, add inputs to externally control the scalar value.
 - **Unit:** select unit of the exported data

Inputs

- No Input connector, the input comes from Web App.
- **Scalar:** If *Ext control* is selected, a scalar value can be received to set the data value.
- **Minus:** If *Ext control* is selected, a boolean rising edge can be received to decrement the data value.
- **Plus:** If *Ext control* is selected, a boolean rising edge can be received to increment the data value.
- **Reset:** If *Ext control* is selected, a boolean rising edge can be received to return to the *Initial Value*.

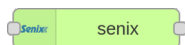
Outputs

- **Output Name:** Integer or decimal data value

WebApp controls

- **Output Name** (*int or float*): On your dashboard, use a “Set number” or “Slider” widget to change value.

11.72 senix



Description

Decode Senix ASCII protocol data sent via serial interface.

Properties

- **Name:** Manta box name
- **Sensor type:** Select product type
- **Output name:** Exported altitude data name
- **Add status:** If selected, check min/max optimum & extremum validity. Export result as a bitfield (see *Notes* below) and as a status icon.

Inputs

- **serialin:** serial data stream

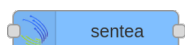
Outputs

- **data:** altitude data + optional data status (see note below)

Notes

Status convention: * bit 0 : ON/OFF (0=off 1=on) * bit 1 : Problem detected (0=valid 1=error) * bit 2 : Problem severity (0=warning 1=error)

11.73 sentea



Description

Export variables such as microstrains or temperatures from Sentea optical interrogator using UDP mode, format 202. It is compatible with DM-4120 and DM-8120 interrogators.

The Sentea box is configured using files generated by Sentea *Peakviewer* software. This software must be used to configure the optical interrogator (see *Notes* below).

Properties

- **Name:** Manta box name
- **Local port:** Enter the UDP input local port
- **Remote IP:** Enter interrogator IP address
- **CSV file:** Select one of the already downloaded .csv file, containing sensor definitions
- **INI file:** Select one of the already downloaded .ini file, containing sensor equations
- **Export peaks:** Check to export peaks on Debug output
- **Upload new Conf:** Upload a new *moi* file from your PC to the Exocet
- **Delete or Download configuration files:** Use *Trash icon* to delete a configuration file and the *download icon* to download the configuration file from the Exocet to the PC
- **Select variables:** Check/Uncheck the variables to export out of the Manta box

Inputs

- **Enable** (*bool*): Connect a boolean to enable/disable the box. Default is enable.

Outputs

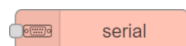
- **Data:** A set of variables such as microstrains or temperatures according to the configuration file.
- **Debug:** Some useful data.

Notes

The Sentea *Peakviewer* software must be used to configure the optical interrogator and to define sensors: * Configure IP address to be in the same subnet. * Write and load a *python script* to configure UDP output communication (format 202), capture and peakfind parameters. Use the *auto start* option to automatically execute the script at startup of the interrogator. * Use the *Sensor definition editor* to configure each FBG parameters (name, wavelength min/ref/max). Export these definitions to a csv file, and upload it directly on the Exocet. * Use the *Sensor equation editor* to define each constants and

sensor equations. FBG and reference names (`_0`) come from *sensor definitions*. All constants and variables are shared and can be used in any equation. The *Unit* field should be one the following: *Strain*, *Temperature*, *Pressure*, *Acceleration*, *Displacement*, *Wavelength*, *Angle*. Export these equations to the *ini* file, and upload it directly on the Exocet.

11.74 serial



Description

Send data to serial port. Exocet sends data to the specified RS232 or RS422 port.

Properties

- **Name:** Manta box name
- **Serial port:** Serial port configuration box. Press the pencil icon to create a new one. Fill the serial line parameters, then press the *Add* button. Once created, it can be shared with other Serial in or out boxes.
- **WebApp:** Allow port selection from Exocet WebApp dashboards

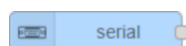
Inputs

- Data to send. Must be of *Opaque* or *String* type. Note that string variables are limited to 32 characters.

Notes

COM3 to COM6 use an USB extension, behaviour can differ from COM1 and COM2.

11.75 serial_in



Description

Get data from a serial port. Exocet received data from the specified RS232 or RS422 port.

Properties

- **Name:** Manta box name
- **Serial port:** Serial port configuration box. Press the pencil icon to create a new one. Fill the serial port parameters, then press the *Add* button. Once created, it can be shared with other Serial in or out boxes.
- **Add TS:** Add timestamp to output data flow
- **WebApp:** Allow port selection from Exocet WebApp dashboards

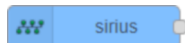
Outputs

- **data** (*Opaque*): received serial data. Depending of use, a buffering may be necessary.

Notes

COM3 to COM6 use an USB extension, behaviour can differ from COM1 and COM2.

11.76 sirius



Description

Get Multiple Sensors from Pixel Sur Mer Sirius gateway.

Sirius is a GPIO sensors gateway used for different purpose: water intrusion detection, sail and rudder position detection, hook position ... It is based on 1 wire technology. A Sirius Gateway can handle up to 3 sensor buses. Each bus can handle multiple sensors.

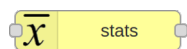
Properties

- **Name:** Manta box name
- **Local port:** Port to received data from

Outputs

- A list of sensors *Boolean* values.

11.77 stats



Description

This Manta Box can perform average, standard deviation, min/max search and others on a data dictionary.

Properties

- **Name:** Manta box name
- **List of processes:**
 - **Process type:** Select the desired process type (see *Notes* below).
 - **Output name:** Exported data name (optional).
 - **Percent:** Only for percentile process type, set the threshold for the percentile process.

Inputs

- **Data:** Connect here the data dictionary to process. Only one wire should be connected on Data input.

Outputs

- **Processed_data:** List of results. *Float* type, except for names.

Notes

Process type:

- **Average:** Get the sum of data divided by the number of data.
- **Variance:** Return how far the set of data are spread out from their mean value.
- **Standard deviation:** Return the dispersion of data from the mean value.
- **Mean deviation:** Return the average dispersion of data from the mean value.
- **Min:** Get the min value from the current set of data, and its name.
- **Max:** Get the max value from the current set of data, and its name.

- **Median:** Get the middle value from the current set of data, i.e. the value that divide the set in two equal parts.
- **Percentile:** Get the given percentile value from the current set of data, i.e. the value below which the given percentage falls (0% gets the minimum, 100% gets the maximum, 50% gets the median).
- **Range:** Get the *Max* minus *Min* value.
- **Max of abs:** Get the max of absolute values of data, and its name.
- **Sum:** Get the sum of data

11.78 string



Description

User controlled box.

This Manta box exports a string data that can be changed from Exocet Web App.

Properties

- **Name:** Manta box name
- **Save:** If selected, the string data is saved to be restored when Exocet is rebooted or Manta graph deployed.
- **Ext control:** If selected, add inputs to externally control the string value.
- **Output Name:** Exported data name. Manta box name is used if not defined.
- **Period:** to send data string periodically. String is also sent at any user change. If *None* is selected, only value changes are sent.
- **Type:** select *single* or *list of string*
- **Initial Value:** (Single type only) initial data value
- **List of strings:** (List of string type only) Press *New predefined value* button to fill a new value. The initial value is the first defined value. Maximum is 50.

Inputs

- No Input connector, the input comes from web app Dashbord widget
- **String:** If *Ext control* is selected, a string value can be received to set the data value.

- **Reset:** If *Ext control* is selected, a boolean rising edge can be received to return to the *Initial Value* or the initial *List of strings*.
- **Add list option:** If *Ext control* and *list of string* is selected, a string value can be received to add a choice in the list of strings
- **Remove list option:** If *Ext control* and *list of string* is selected, a string value can be received to remove a choice in the list of strings

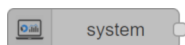
Outputs

- **Output Name:** String data value

WebApp controls

- **Output Name** (*string*): On your dashboard, use a “Set Text” widget to change value.

11.79 system



Description

This Manta box exports system usage data. Some of these data are only exported when used.

Properties

There is no customizable properties.

Outputs

Data Name	Unit	Description
Bus Load Can X	%	Estimated Can X bus load
COMX Rx Load	%	Estimated COM X load of the reception
COMX Tx Load	%	Estimated COM X load of the transmission

Data Name	Unit	Description
Conf usage	%	Estimated configuration memory usage (Manta graph, dashboards, user files...)
CPU X	%	Estimated load of CPU X
CPU Total	%	Estimated load of all CPU
CPU Temperature	°C	System CPU temperature
Cycle usage	%	Max time load of main periodic thread
Ethernet	kB/s	Ethernet data transfert rate
Manta loop detected	-	Data loop detected in the Manta graph
Mem Available	Mb	Estimated available RAM memory
Mem Total	Mb	Total amount of usable RAM memory
Mem usage	%	Ratio of available memory on total memory
Node input overflow	-	Total number of Node input buffer overflow from start-up
Num CPU	-	CPU number
Num variables usage	%	Overall number of variables on maximum number
Output Connectors usage	%	Overall number of box outputs on maximum number
Record disk usage	%	Estimated record memory usage
Record files usage	%	Number of record files on maximum number
Slowdown detected	-	System slowdown detected
Status_CanX	-	CAN bus overload detected
Status_ComXRx	-	Reception COM X overload detected
Status_ComXTx	-	Transmission COM X overload detected
Status_ConfUsage	-	Configuration memory usage too high detected
Status_CpuX	-	CPU X overload detected
Status_CpuTotal	-	CPU total overload detected
Status_CpuTemperature	-	CPU temperature too high detected

Data Name	Unit	Description
Status_CycleUsage	-	Cycle usage too high detected
Status_MemUsage	-	RAM memory usage too high detected
Status_NumVariablesUsage	-	Too many variables detected
Status_OutConnectorsUsage	-	Too many output connectors detected
Status_RecordsDiskUsage	-	Record memory usage too high detected
Status_RecordsFilesUsage	-	Record files usage too high detected
Status_WebRefresh	-	Too long web refresh period detected
System status	-	Overall system status (see note below)
System uptime	s	Exocet application uptime
Voltage too low	-	Total number of low power detected
Voltage too low detected	-	Low power detected
Web refresh	ms	Max refresh delay of WebApp

Output frequency: 0.5 Hz

Notes

CPU Load is the ratio of active time on active plus idle time. It is calculated on 2s. Total is the average of all cores.

Variables Max allowed variables is 20,000. Max allowed output connectors is 2048.

Status Status convention: * bit 0 : ON/OFF (0=off 1=on) * bit 1 : Problem detected (0=valid 1=error) * bit 2 : Problem severity (0=warning 1=error)

11.80 table



Description

User controlled box.

This Manta box exports a table data that can be changed from Exocet Web Application. Maximum size is 32 x 32.

Properties

- **Name:** Manta box name
- **Save:** If selected, the data value is saved to be restored when Exocet is rebooted or Manta graph deployed.
- **Output Name:** Exported data name. Manta box name is used if not defined.
- **Period:** to send data value periodically. Data value is also sent at any user change. If *None* is selected, only data changes are sent.
- **Ext control:** If selected, add inputs to externally control the table value.
- **Use text file:** If selected, use selected *Table file* as initial table. Then, all changes in table are saved in the file.
- **Initial table:** Enter initial values of the table (integer or float format). Column separator must be space or tab.
- **Table file:** Select one of the already downloaded Table files. It must be text file with extension .txt. Column separator must be space or tab.

Inputs

- No Input connector, the input comes from Web App.
- **Table:** If *Ext control* is selected, a table can be received to set the table.
- **Reset:** If *Ext control* is selected, a boolean rising edge can be received to return to the *Initial table*.

Outputs

- **Output Name:** Table data value

WebApp controls

- **Output Name** (*float table*): On your dashboard, use a “Table input” widget to change values.

11.81 tcp_in



Description

Get data from a TCP interface.

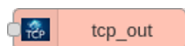
Properties

- **Name:** Manta box name
- **Client:** Check to work as a client, otherwise it works as a server.
- **Local port:** Enter the input local port
- **Remote IP:** Enter the emitter IP address
- **Remote port:** Enter the emitter output port
- **WebApp:** Allow connexion parameters access from Exocet WebApp dashboards
- **Add TS:** Add timestamp to output data flow

Outputs

- **data** (*Opaque*): TCP received data

11.82 tcp_out



Description

Send data to a TCP interface.

Properties

- **Name:** Manta box name
- **Client:** Check to work as a client, otherwise it works as a server.
- **Local port:** Enter the output local port
- **Remote IP:** Enter the receiver IP address

- **Remote port:** Enter the receiver input port
- **WebApp:** Allow connexion parameters access from Exocet WebApp dashboards

Inputs

- Data to send. Must be of *Opaque* or *String* type. Note that string variables are limited to 32 characters.

11.83 tick



Description

This box generates a periodic tick. It can for example be used to periodically run a Python box.

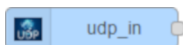
Properties

- **Name:** Manta box name
- **Frequency:** Select the tick event frequency. **IMPORTANT NOTE:** Selecting a frequency up to 50Hz may impact Exocet CPU performances.

Outputs

- **count** (*Integer*): Tick counter
- **period** (*Float*): Period measurement in ms
- **frequency** (*Float*): Inverse of period measurement in Hz

11.84 udp_in



Description

Get data from an UDP interface.

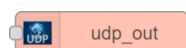
Properties

- **Name:** Manta box name
- **Local port:** Enter the input local port
- **Add TS:** Add timestamp to output data flow
- **Filter:** If selected, filter incoming data on local port using *Filter IP*
- **Filter IP:** Only if *Filter* is selected, IP to filter out
- **WebApp:** Allow connexion parameters access from Exocet WebApp dashboards

Outputs

- **data** (*Opaque*): UDP received data

11.85 udp_out



Description

Send data to an UDP interface.

Properties

- **Name:** Manta box name
- **Local port:** Enter the output local port (optional)
- **Remote IP:** Enter the receiver IP address
- **Remote port:** Enter the receiver input port
- **Broadcast:** Allow data to be sent to a broadcast address
- **WebApp:** Allow connexion parameters access from Exocet WebApp dashboards

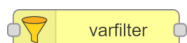
Inputs

- Data to send. Must be of *Opaque* or *String* type. Note that string variables are limited to 32 characters.

WebApp controls

- **RemoteIp** (*string*): If *WebApp* is selected, on your dashboard, use a “Set Text” widget to change value.
- **RemotePort** (*int*): If *WebApp* is selected, on your dashboard, use a “Set Number” widget to change value.

11.86 varfilter



Description

The purposes of this Manta Box is to extract one or more variables from an input data stream.

Properties

- **Name:** Manta box name
- **Splitter:** Add an output to export rejected variables
- **Variables:** Select variables to extract. Note that the Manta graph must have been deployed a first time after adding the *VarFilter* box and connecting it, to populate the variables list.

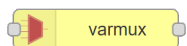
Inputs

- **Data stream:** Data stream to analyse

Outputs

- **Filter_in:** Extracted variables
- **Filter_out:** Rejected variables, only if *Splitter* option has been checked

11.87 varmux



Description

The purpose of this Manta Box is to multiplex several input data streams.

Properties

- **Name:** Manta box name
- **Frequency:** Select output data frequency:
 - Value of the desired output data frequency.
 - *slowest* to synchronise with the slowest input data stream received on *Main input* (see *Notes* below).
 - *fastest* to send each time an input data is received on *Main input*.
- **Policy:** Select the desired sending policy (only applied to data streams from the *Main input*) :
 - *Standard*: Send data according to the selected frequency.
 - *Send only if refreshed*: Send data according to the selected frequency if at least one variable from the *Main input* has been updated.
 - *Send immediately if refreshed*: Send data according to the selected frequency and when at least one variable from the *Main input* has been updated.
 - *Send only if changed*: Send data according to the selected frequency if at least one variable from the *Main input* has changed.
 - *Send immediately if changed*: Send data according to the selected frequency and when at least one variable from the *Main input* has changed.
- **Pass through:** Export all non renamed variables, permit to filter out all non renamed variables
- **Disable checks:** Disable configuration check at runtime to save CPU
- **List of variables to rename** (optional):
 - **Var_in**: Enter the input variable name
 - **Var_out**: Enter the name to use at output

Inputs

- **Main input:** Data streams to be multiplexed (max 64) and be used by the selected *policy* and *frequency*
- **Secondary input:** Other data streams to be multiplexed (max 64)

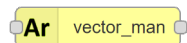
Outputs

- Data stream containing all data of input data streams, and sent at the specified *frequency* and *policy*.

Notes

- Maximum number of variables is limited to 1024.
- The principle of *slowest* frequency is to update the output when all data streams connected to the *Main input* have been refreshed. This can block the output if one data stream is no longer refreshed. To avoid this, a monitoring algorithm update the output if the delay is too long compare to the last update periods. To disable this algorithm you can select the *Send only if refreshed* policy.
- You can also use the *Main input* timeout to stop sending if main input is not refreshed.

11.88 vector_man



Description

The purposes of this Manta Box is:

- To rotate a 3D vector by providing 3 rotations angles for each axis
- You can scale each axis component with individual factors
- You can add an offset to each axis component

Properties

- **Name:** Manta box name
- **Pass through:** output vector = input vector
- **Frequency:** Box processing frequency
- **Rotation order:** RPY => roll pitch yaw, YRP => yaw pitch roll
- **Mult. factors:** x y z multiplying factors
- **Offsets:** x y z offsets
- **Inputs mapping**

- **X**: X vector name.
- **Y**: Y vector name.
- **Z**: Z vector name.
- **Heel**: Trim name.
- **Trim**: Trim name.
- **Heading**: Heading name.
- **Outputs mapping**
 - **New vector**: A generic name for the new vector.

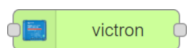
Inputs

- Input 1 : Vector to be manipulated (XYZ).
- Input 2 : Heel, Trim and Heading. If no data is provided no rotation will be applied to that axis

Outputs

- 3 outputs data will be generated with extension .x .y .z

11.89 victron



Description

Decode Victron VE.Direct-Protocol data sent via serial interface.

Properties

- **Name**: Manta box name

Inputs

- **serialin**: serial input frame

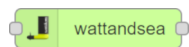
Outputs

- **data:** device data

Data are exported each second.

Data Name	Description	Type	Unit
vbat	Main battery voltage	Float	Volts
vaux	Auxiliary (battery) voltage	Float	Volts
vpan	Panel voltage	Float	Volts
wpan	Panel power	Integer	Watt
ibat	Battery current	Float	Amperes
iload	Load current	Float	Amperes
temp	Battery temperature	Integer	°C
pow	Instantaneous power	Integer	Watt
consumedAmpHours	Consumed amp hours	Float	Amperes Hour
soc	State of charge	Float	%
timeToGo	Time to go	Integer	Minutes
alarm	Alarm condition active	Boolean	N/A
relay	Relay state	Boolean	N/A
alarmreason	Alarm reason	Integer	N/A
deepestDischarge	Depth of the deepest discharge	Float	Amperes Hour
nbChargeCycles	Number of charge cycles	Integer	N/A
vbatMin	Minimum main (battery) voltage	Float	Volts
vbatMax	Maximum main (battery) voltage	Float	Volts
vauxMin	Minimum auxiliary (battery) voltage	Float	Volts
vauxMax	Maximum auxiliary (battery) voltage	Float	Volts
maxpowtoday	Maximum power today	Integer	Watt
errorCode	Error code	Integer	N/A
stateOfOperation	State of operation	Integer	N/A

11.90 wattandsea



Description

Decode Watt&Sea status sent via serial interface

Properties

- **Name:** Manta box name
- **Protocol:** Select the desired protocol

Inputs

- **serial_in:** serial frame to decode

Outputs

Aero Generator:

- **Data:**

Data Name	Description	Type	Unit
RPM	Generator speed	Integer	RPM
VInDC	Input voltage	Float	Volts
VCharge	Battery charge Voltage	Float	Volts
IchBat1	Battery 1 current	Float	Amperes
IchBat2	Battery 2 current	Float	Amperes

- **System:**

Data Name	Type	Unit
VBus	Float	Volts

Data Name	Type	Unit
IBus	Float	Amperes
PBus	Float	Watts
VchBat1	Float	Volts
VchBat2	Float	Volts
Temperature	Float	°C

Aero-v2 Generator:• **Converter:**

Data Name	Description	Type	Unit
Vgene	Generator voltage	Float	Volts
Vbatt	Battery voltage	Float	Volts
Ibatt	Battery current	Float	Amperes
RPM	Generator speed	Integer	RPM
DefBatt	Battery default	Integer	-
EtatCharge	Charge state	Integer	-
Alarme	Alarms	Integer	-

• **Servo:**

Data Name	Type	Unit
Current	Float	Amperes
Velocity	Integer	-
Position	Float	Deg
Voltage	Float	Volts
Temperature	Integer	°C

Hydro Generator:

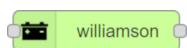
- **Converter:**

Data Name	Description	Type	Unit
Uin	Generator voltage	Float	Volts
Uout	Output voltage	Float	Volts
Iout	Output current	Float	Amperes
Temp	Converter internal temperature	Float	°C
F1	Generator #1 speed	Integer	RPM
F2	Generator #2 speed	Integer	RPM

- **Pump:**

Data Name	Type	Unit
Pressure	Float	Bars
ModeAuto	Bool	-
Pitch	String	-
Active	Bool	-

11.91 williamson



Description

Decode williamson battery status sent to CAN bus

Properties

- **Name:** Manta box name
- **Reference:** select battery reference (see note below)
- **Refresh:** data refresh period in second. From 1 to 3600.
- **Battery ID:** select the ID of the battery. From 1 to 25.

- **Use BatID as prefix:** If selected, all battery data are prefixed with “BatX_”. * **

Inputs

- **canin:** can bus input to receive battery data

Outputs

- **data:** battery data
- **canout:** can bus output to send battery status query

Each battery sends the following data:

Data Name	Description
fault	Fault code. See below
ich	Charge current
iequil	Balancing current
iload	Load current
levelch	Charge level
nbcycles	Cycle numbers
tbat	Internal temperature
tmos	Switches temperature
vbat	Battery voltage
vch	Charge voltage

Fault codes:

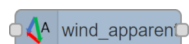
- **0** : Balancing current hight
- **2** : Discharge current hight
- **4** : User overvoltage
- **20**: Charging current hight
- **64** :User short-circuit
- **68**: overvoltage or temperature hight

Notes

Battery Reference

- **WILPA1741, WILPA2044, WILPA2533A** : with SAFT element of type MP176065 G5 and G6
- **WILPA2795A** : with SAFT element of type MP176065 xlr
- **WILPA2935A** : with Panasonic element of type NCR 18650A

11.92 wind_apparent



Description



Figure 87

Back-calculation of the undamped apparent wind in the horizontal plane using True wind and navigation data. Only one calculation is available. All inputs are necessary.

Properties

- **Name:** Manta box name
- **Ref. to Course:** The Apparent Wind Angle is referenced to the Course axis. Unchecked : Referenced to the longitudinal boat axis
- **TW ref. to Course:** The True Wind Angle is referenced to the Course axis. Unchecked : Referenced to the longitudinal boat axis

Inputs

Input 0

Name	Units	Description
TWA_undamp	(°T)	Undamped true wind angle, referenced on the horizontal plane

Name	Units	Description
TWS_undamp	(kn)	Undamped apparent wind speed, referenced on the horizontal plane

Input 1

Name	Units	Description
Leeway	(°)	Undamped Leeway Angle

Input 2

Name	Units	Description
BoatSpeed	(kn)	Undamped BoatSpeed relative to the surface of water

Outputs

Name	Units	Description
AWA_undamp	(°)	Undamped Apparant wind angle, referenced on the horizontal plane
AWS_undamp	(°)	Undamped Apparant wind speed, referenced on the horizontal plane

Notes**Apparent Wind definition:**

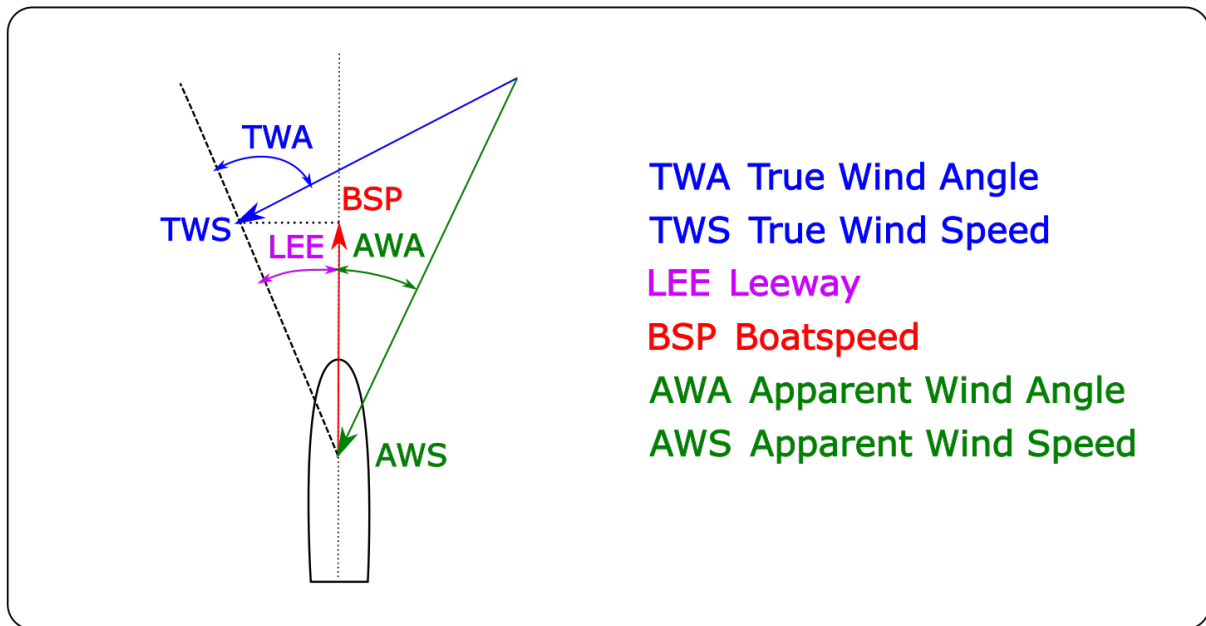
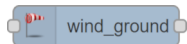


Figure 88

11.93 wind_ground



Description



Figure 89

Calculation of the ground wind using True Wind data and current, relative to the sea floor (referenced at 10m altitude if weather forecast comparison is required). Only one calculation is available. All inputs are necessary. No options are required.

Properties

- **Name:** Manta box name

Inputs

Input 0

Name	Units	Description
TWD_disp	(°T)	Damped True Wind Direction plane, relative to the sea surface and referenced on the horizontal plane
TWS_disp	(kn)	Damped True Wind Speed, relative to the sea surface and referenced on the horizontal plane

Input 1

Name	Units	Description
TideSet	(°T)	Damped Surface Current Direction
TideRate	(kn)	Damped Surface Current Speed

Outputs

Name	Units	Description
GW_direction	(°T)	Damped Ground Wind Direction, relative to the sea floor and referenced on the horizontal plane
GW_speed	(kn)	Damped Ground Wind Speed, relative to the sea floor and referenced on the horizontal plane

Notes

Ground Wind definition:

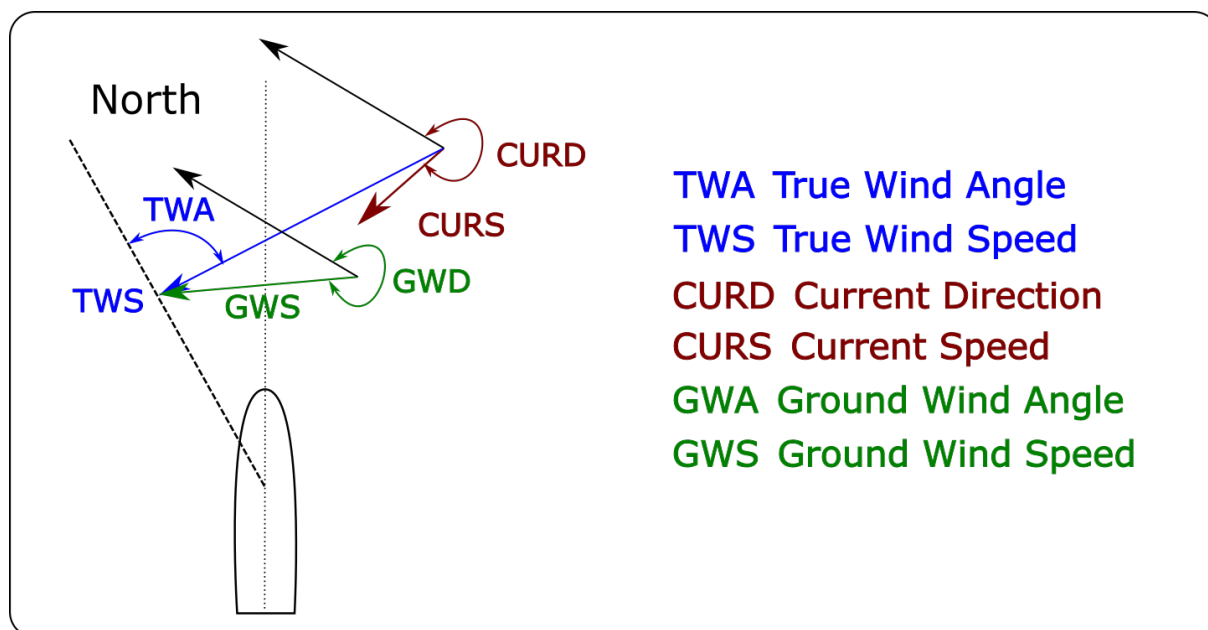


Figure 90

11.94 wind_mastrot



Description



Figure 91

This Manta box compensate the selected Mast Head Unit “MHU_angle” from the mast rotation and twist.

Properties

- **Name:** Manta box name
- **Bypass:** Checked = No rotating mast in the installation
- **UseTwist:** Checked = Use twist correction

Inputs

Input 0 : Calibrated Selected Mast Head Unit

Value Name	Units	Description	Calculations using this input
MHU_angle	(°)	Measured Wind Angle in 2D plane at the top of the mast	Measured Wind Angle (Output 0)
MHU_speed	(°)	Measured Wind Speed in 2D plane at the top of the mast	Measured Wind Speed (Output 0)

Input 1 : Optional Mast Rotation

Value Name	Units	Description	Calculations using this input
RotatingMastAngle	(°)	Mast Rotation Angle around the mast longitudinal axis and measured at the bottom of the mast, positive in clockwise direction	Measured Wind Angle (Output 0)

Note : Add a time out in this port if you use a mast angle sensor

Input 2 : Optional Mast Twist

Value Name	Units	Description	Calculations using this input
MastTwist	(°)	Mast Twist Angle measured between the bottom and the top of the mast, positive in clockwise direction	Measured Wind Angle (Output 0)

Note : Add a time out in this port if you use a mast twist solution

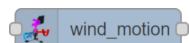
Outputs

Output : Measured Wind

Value Name	Units	Description	Necessary inputs to calculate this data
MW_angle	(°)	Measured Wind Angle in 2D plane at the top of the mast	- MHU_angle (Input 0) - RotatingMastAngle (Input 1) - MastTwist (Input 2)
MW_speed	(°)	Measured Wind Speed in 2D plane at the top of the mast	- MHU_speed (Input 0)

Note : Output frequency is the same as Input 0 frequency

11.95 wind_motion



Description



Figure 92

Calculate the Uncorrected Apparent Wind defined in the horizontal plane and corrected from:

- Hull and mast attitude effect (geometric tilt error on the 2D measured wind angle, and quasi cosine tilt response of the 2D anemometer sensor).
- The motion of the Mast Head Unit (due to the boat motion in the waves and to the big lever arm of the mast).

The Apparent Wind can be then back calculated from the upwash and wind shear corrected true wind.

Properties

- **Name:** Manta box name
- **Correction:** define the level of wind denoising:
 - **Attitude:** Only hull and mast attitude (heel, trim, MastRake, MastCant) are applied to the measured wind projection on the horizontal plane. No dynamic motion compensation.
 - **Dual rotation:** Use pitch & roll angular velocities (p, q) and Mast Head Unit height (-MHUz) for motion correction.
 - **Attitude & dual rotations:** Use hull and mast attitude (heel, trim, MastRake, MastCant) for geometric correction, pitch & roll angular velocities (p, q) and Mast Head Unit height (-MHUz) for motion correction.
 - **Attitude & triple rotations:** Use hull and mast attitude (heel, trim, MastRake, MastCant) for geometric correction, pitch & roll & yaw angular velocities (p, q, r) and Mast Head Unit full position (MHUx, MHUy, -MHUz) for motion correction. Useful in fast tacking or if the Mast Head Unit is not installed above the center of buoyancy.
 - **Full Motion:** Use hull and mast attitude (heel, trim, MastRake, MastCant) for geometric correction, pitch/roll/yaw angular velocities (p, q, r) and surge/sway linear velocities (u, v) and Mast Head Unit full position (MHUx, MHUy, -MHUz) for motion correction. Useful if the position of the center of buoyancy can change dynamically (multihull).
- **MHU x,y,z:** Position of the mast head unit from the center of rotation (note: MHUz = -MHU attitude)
- **WebApp:** Allow MHU position access from Exocet WebApp dashboards

Inputs

Input 0

Value Name	Units	Description	Calculations using this input
MW_angle	(°)	Measured Wind Angle in 2D plane at the top of the mast	CMW_angle, CMW_speed (output 0)
MW_speed	(kn)	Measured Wind Speed in 2D plane at the top of the mast	CMW_angle, CMW_speed (output 0)

Inputs 1

Name	Units	Description
Heel	(° or rad)	Boat attitude relative to the horizontal plane
Trim	(° or rad)	Boat attitude relative to the horizontal plane
p, q, r	(°/s or rad/s)	Angular velocities around x,y,z boat axis
u, v, w	(m/s)	Linear velocities in x,y,z boat axis

Inputs 2

Name	Units	Description
MastRake	(°)	Mast attitude relative to the hull axis
MastCant	(°)	Mast attitude relative to the lateral axis

Output

Name	Units	Description
CMW_angle	(°)	Corrected Measured wind angle (referenced on the horizontal plane passing through the MHU position)
CMW_speed	(°)	Corrected Measured wind speed (referenced on the horizontal plane passing through the MHU position)
CMW_altitude	(m)	Altitude of the wind measurement
CMW_altRatio	(%)	Equal to MHU_Height / Altitude (depending of the hull & mast altitude)

WebApp controls

- **MHU x,y,z** (*float*): If *WebApp* is selected, on your dashboard, use a “Set Number” widget to change MHU position.

Note

Reference: The standard SNAME's (1950) notation (forward/starboard/downward and North,East,Down frames) is used internally and for sensor position. The origin of the position must be close to the center of rotation (top of the keel).

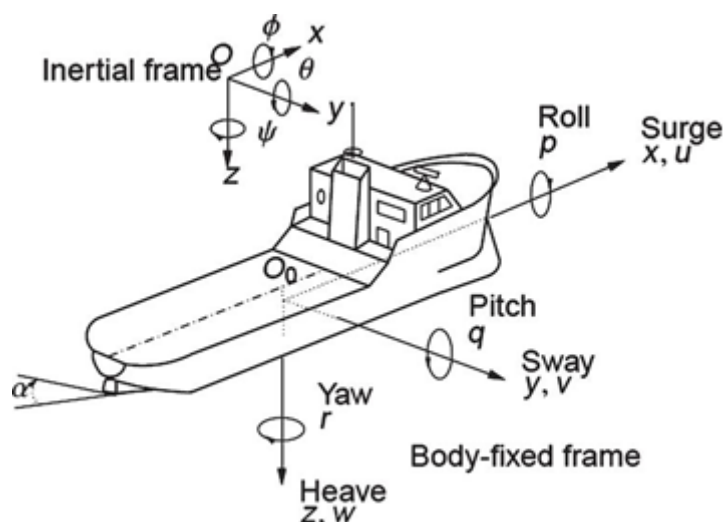
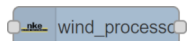


Figure 93

11.96 wind_processor_hr**Description**

Encode data to drive Processor HR with Pixel wind.

Inputs

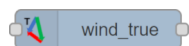
Name	Units	Description
Yaw	(°)	Heading (magnetic, 0/360)
Pitch	(°)	Negative with boat goes down (-180/180)
Roll	(°)	Negative with boat goes starboard (-180/180)

Name	Units	Description
YawRate	(°/s)	Positive clockwise
PitchRate	(°/s)	Negative with boat goes down
RollRate	(°/s)	Negative with boat goes startboard
AWA	(°)	Apparent wind angle, positive with wind from startboard (-180/180)
AWS	(Kn)	Apparent wind speed
TWA	(°)	True wind angle, positive with wind from startboard (-180/180)
TWS	(Kn)	True wind speed
SOW	(Kn)	Boat speed (pilot)
Declination	(°)	

Outputs

- **Serial out:** Prepare 25Hz frames to sent to NKE system

11.97 wind_true



Description



Figure 94

Calculate the original true wind (TWA, TWS) in the horizontal plane using corrected measured wind and navigation data. Calculate TWD if course is available. Only one straight forward calculation is available, no option are required.

Properties

- **Name:** Manta box name
- **Ref. to Course:** The True Wind Angle is referenced to the Course axis. Unchecked : Referenced to the longitudinal boat axis

Inputs

Input 0 : Corrected Measured Wind

Value Name	Units	Description	Calculations using this input
CMW_angle	(°)	Corrected Measured wind angle	TWA, TWS and TWD (Output 0)
CMW_speed	(kn)	Corrected Measured wind speed	TWA, TWS and TWD (Output 0)

Input 1

Value Name	Units	Description	Calculations using this input
Course	(°T)	Course	TWD (output 0)
Leeway	(°)	Leeway Angle	TWA, TWS and TWD (Output 0)

Input 2

Value Name	Units	Description	Calculations using this input
BoatSpeed	(kn)	BoatSpeed relative to the surface of water	TWA, TWS and TWD (Output 0)

Output

Name	Units	Description
TWA_orig	(°)	Original true wind angle (referenced on the horizontal plane passing through the MHU position)

Name	Units	Description
TWS_orig	(kn)	Original true wind speed (referenced on the horizontal plane passing through the MHU position)
TWD_orig	(°T)	Original true wind direction (referenced on the horizontal plane passing through the MHU position)

Notes

True Wind definition:

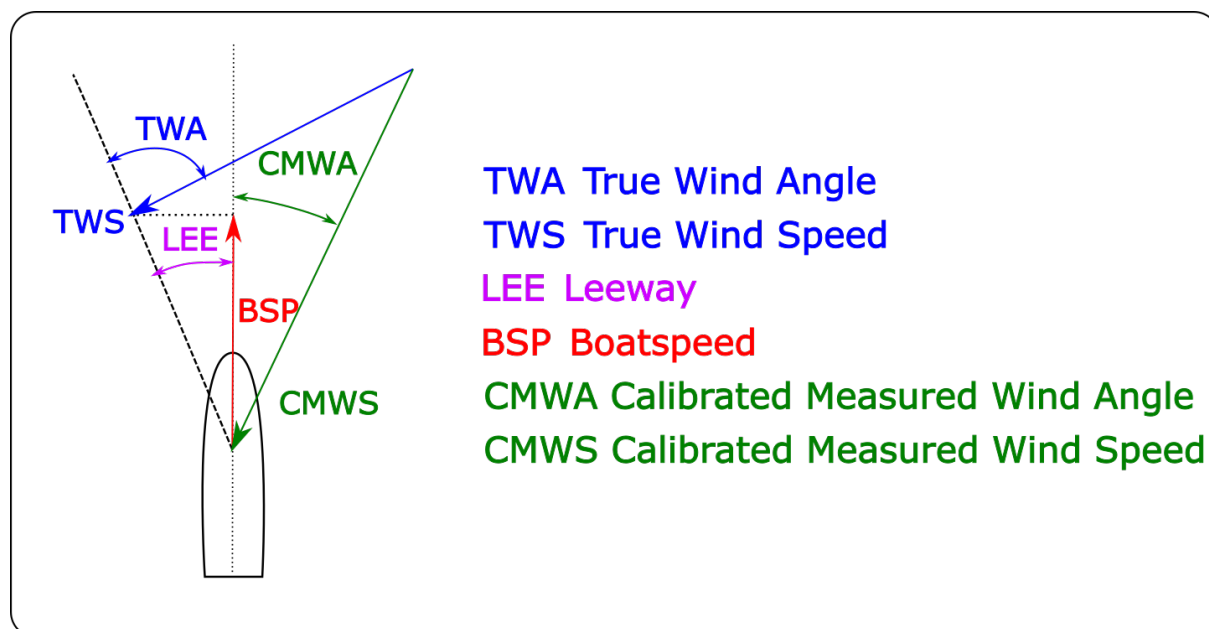
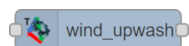


Figure 95

11.98 wind_upwash



Description



Figure 96

Use correction tables to compensate the True Wind from various symmetrical errors (identical on port and starboard tack) :

- Downwind Upwash : Wind perturbations at the Mast Head Unit position from the tip vortices (mast profile, the MHU wand, spinnaker, mainsail...)
- Mast attitude : The mast twist is function of reefing, running backstay tension...
- Residual errors : Not perfect sensors calibration

Properties

- **Name:** Manta box name
- **Bypass:** If checked, no upwash correction is applied
- **Ext control:** If checked, add inputs to dynamically manage the correction tables
- **Save:** If checked, save dynamic update on the file. It is strongly recommended to not periodically send a dynamic table if this option is enabled.
- **Interpolation:** Select desired interpolation method
- **File format:** Select used file format
- **List of correction tables:**
 - **Set name:** Enter a name for the set of correction tables
 - **File TWA:** Select a true wind angle correction table (format, see note below)
 - **File TWS:** Select a true wind speed correction table (format, see note below)

Inputs

Input 0: True Wind

Name	Units	Description
TWA_orig	(°)	Original true wind angle
TWS_orig	(kn)	Original true wind speed
TWD_orig	(°T)	Original true wind direction

Input 1/2: Ext correction table

Name	Type	Description
CorrecTable	FloatTable	Dynamic correction table. Variable name is free.

Input 3: Ext table selection

Name	Type	Description
TableSelector	Scalar/String	Dynamic table selector. Variable name is free. Use “Set name” or table set number as value. Number of table set starts at 0.

Outputs**Output 0: Corrected True Wind**

Name	Units	Description
TWA_upwash	(°)	Upwash corrected true wind angle (referenced on the horizontal plane passing through the MHU position)
TWS_upwash	(kn)	Upwash corrected true wind speed (referenced on the horizontal plane passing through the MHU position)
TWD_upwash	(°T)	Upwash corrected true wind direction (referenced on the horizontal plane passing through the MHU position)

Output 1: Status

Name	Units	Description
AngleCorrection	(°)	Current TWA and TWD correction
SpeedCorrection	(kn)	Current TWS correction
TableSetSelected		Number of selected table set
TableSetSelectedStr		Name of selected table set

Notes

Tables requirements:

- B&G WTP/Deckman, B&G H5000 or NKE Processor Regatta compatible format
- TWA must be $> 0^\circ$ and $\leq 180^\circ$
- TWS must be $\geq 0\text{kn}$ and $\leq 100\text{kn}$
- TWA corrections must be $\geq -90^\circ$ and $\leq 90^\circ$
- TWS corrections must be $\geq -90\text{kn}$ and $\leq 90\text{kn}$

Calculations above boundaries:

TWA correction (adjwa.d):

- The spline surface interpolation is constrained in order to have a null correction at $\text{TWA}=0^\circ$ and $\text{TWA}=180^\circ$
- For TWS below the first TWS : the same TWA correction as the first TWS is applied
- For TWS above the last TWS : the same TWA correction as the last TWS is applied

TWS correction (adjvt.d):

- The spline surface interpolation is constrained in order to have a null correction at $\text{TWA}=0^\circ$
- The spline surface interpolation is constrained in order to continue with the same correction at $\text{TWA}=180^\circ$
- For TWS below the first TWS : the same TWS correction as the first TWS is applied
- For TWS above the last TWS : the same TWS correction as the last TWS is applied

Spline surface limitations and multiple upwash tables:

- The 3D spline interpolation can give bumps if 2 different corrections are too close and if the TWS/TWA are poorly distributed.
- The spline surface is smoother with only few corrections per TWS.

- If you require a step in correction function of TWA at angles that you need to change sails/reefs or if you can't rise the optimal sail set, you can prepare multiple upwash tables and select a specific sail set.

B&G WTP/Deckman & NKE Processor Regatta file format

Size: maximum 6 pairs of Co/TWA, maximum 20 TWS lines.

	v1	a1	v2	a2	v3	a3	v4	a4	v5	a5
TWS	Co	TWA	Co	TWA	Co	TWA	Co	TWA	Co	TWA
TWS	Co	TWA	Co	TWA	Co	TWA	Co	TWA	Co	TWA

B&G H5000 file format

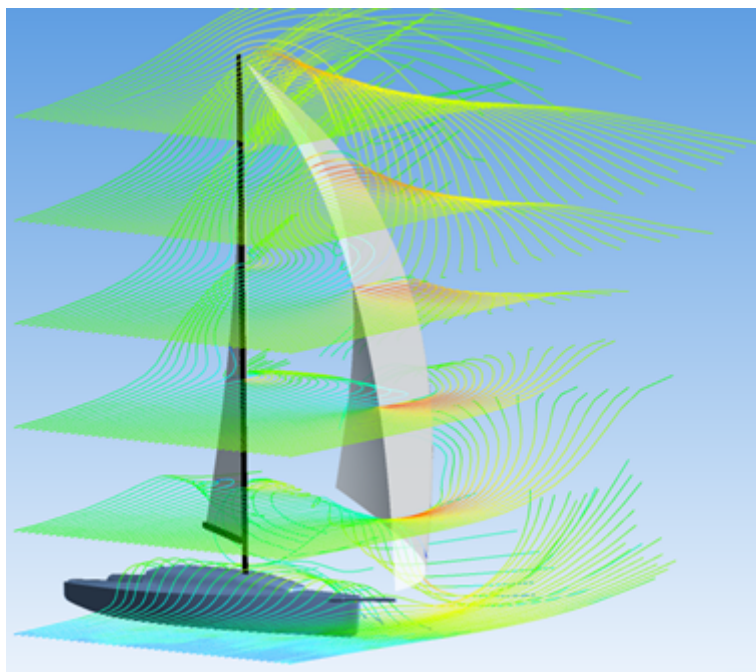
Size: 7 TWS columns, maximum 6 TWA lines.

	TWS	TWS	TWS	TWS	TWS	TWS	TWS
TWA	Co	Co	Co	Co	Co	Co	Co
TWA	Co	Co	Co	Co	Co	Co	Co

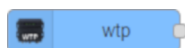
Upwash modelisation:

In a regular wind, the Ground Wind Speed and Direction must be similar on average when going from tack to tack or from upwind to downwind. This error is name *wiggle*. The angle and speed wiggle will be used to improve the upwash tables calibration (see the calibration help manual). You can use the True Wind Speed and Direction if the current is negligible.

The upwash and mast twist is physically applied on the apparent wind, we prefer calibrate their errors on the true wind and back calculate the apparent wind in order to make a mental calculation calibration possible.

**Figure 97**

11.99 wtp



Description

Import B&G WTP3 sailing instrument variables.

Properties

- **Name:** Manta box name
- **Multicast IP address:** Enter UDP multicast IP address
- **Multicast port:** Enter the UDP multicast port number

Use the variable list to check/uncheck the WTP3 variables to export. *Select All* and *Unselect All* buttons can be used to select/unselect all the variables.

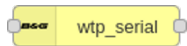
Outputs

- The list of selected WTP variables

Notes

Output Frequency: 10 Hz Variables must be declared in the *bg_vars.d*, where the EOL symbol is “CRLF”.

11.100 wtpSerialManager



Description

Ease of use WTP3 Serial interface: * Configure WTP3 Serial module * Prepare data to send through N2000 * Extract received data from N2000

Properties

- **Name:** Manta box name
- **CAN port:** Select CAN port
- **Device selection:** Select WTP Serial module
- **Instance:** WTP Serial module instance
- **Port:** WTP Serial module connector (COM1 = connector A)
- **Direction:** WTP Serial module usage
- **Baudrate:** Serial communication baudrate
- **Data bits:** Number of data bits to use
- **Stop bits:** Number of stop bits to use
- **Parity:** Parity configuration to use
- **Protocol:** Serial communication protocol to use (according to WTP Serial module wiring)
- **Output mode:** Output mode to use

Inputs

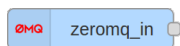
- **PGN 65280:** Must be connected to port 65280 of a net2000 in box with type set as WTP3 Serial
- **PGN 130822:** Must be connected to port 130822 of a net2000 in box with type set as WTP3 Serial

- **PGN 130823:** Must be connected to port 130823 of a net2000 in box with type set as WTP3 Serial
- **Data** (*opaque*): Serial data to send (only if module is used as output).

Outputs

- **PGN 65280:** Must be connected to port 65280 of a net2000 out box with type set as WTP3 Serial
- **PGN 130822:** Must be connected to port 130822 of a net2000 out box with type set as WTP3 Serial
- **PGN 130823:** Must be connected to port 130823 of a net2000 out box with type set as WTP3 Serial
- **Received data** (*opaque*): Received serial data (only if module is used as input).

11.101 zeromq_in



Description

Get data from a ZeroMQ interface.

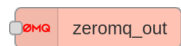
Properties

- **Name:** Manta box name
- **Client:** Check to work as a client, otherwise it works as a server.
- **Local port:** Only for server, enter the local port to receive external connection
- **Remote IP:** Only for client, enter the remote IP address to connect to
- **Remote port:** Only for client, enter the remote port to connect to
- **Type:** Select the type corresponding to the desired communication model (Req/Rep, Pub/Sub, Push/Pull...)
- **Topic:** Only for *Subscribe* type, enter an optionnal *topic* to accept only incoming messages beginning with this specified prefix
- **Add TS:** Add timestamp to output data flow

Outputs

- **data** (*Opaque*): ZeroMQ received data

11.102 zeromq_out



Description

Send data to a ZeroMQ interface.

Properties

- **Name:** Manta box name
- **Client:** Check to work as a client, otherwise it works as a server.
- **Local port:** Only for server, enter the local port to receive external connection
- **Remote IP:** Only for client, enter the remote IP address to connect to
- **Remote port:** Only for client, enter the remote port to connect to
- **Type:** Select the type corresponding to the desired communication model (Req/Rep, Pub/Sub, Push/Pull...)

Inputs

- Data to send. Must be of *Opaque* or *String* type. Note that string variables are limited to 32 characters.

12 Appendix

12.1 Recover the IP address of the Exocet

In case the IP address of the Exocet has been forgotten, a recovery IP address is available. For that, connect directly the Exocet to a computer. Fix the IP address of the computer to 10.0.0.2 (mask : 255.255.255.0)

Then on a web browser connect to 10.0.0.1. The web interface if the Exocet is now accessible. Change the common IP address to the desired one.

12.2 Configuring your PC to be a Time server

Refer to the Microsoft support at :

<https://support.microsoft.com/en-us/help/816042/how-to-configure-an-authoritative-time-server-in-windows-server>