MaxAir Technical – Communication with an eBUS enabled Boiler

eBUS Background

eBUS (energy bus) is a 2-wire digital serial data-bus communication interface used in heating and solar energy appliances, by mainly German manufacturers. It was originally proposed by the Karl Dungs company, and has since been adopted by several other manufacturers. The eBUS interface has also been used by home-automation enthusiasts to connect their domestic solar or heating system to a networked PC for monitoring or remote control.

The eBUS 2-wire interface is an asynchronous serial port with active-low voltage that exchanges 8-bit bytes with start and (single) stop bits (no parity bit), at a symbol rate of 2400 baud, and can be implemented with a standard UART plus a voltage converter. It differs from the RS-232 interface, from which it is derived, in that the voltage levels were chosen to allow the bus also to supply power to bus participants, that can use a voltage stabilizer to derive an internal 5 V supply:

logical 0 = 9–12 volt

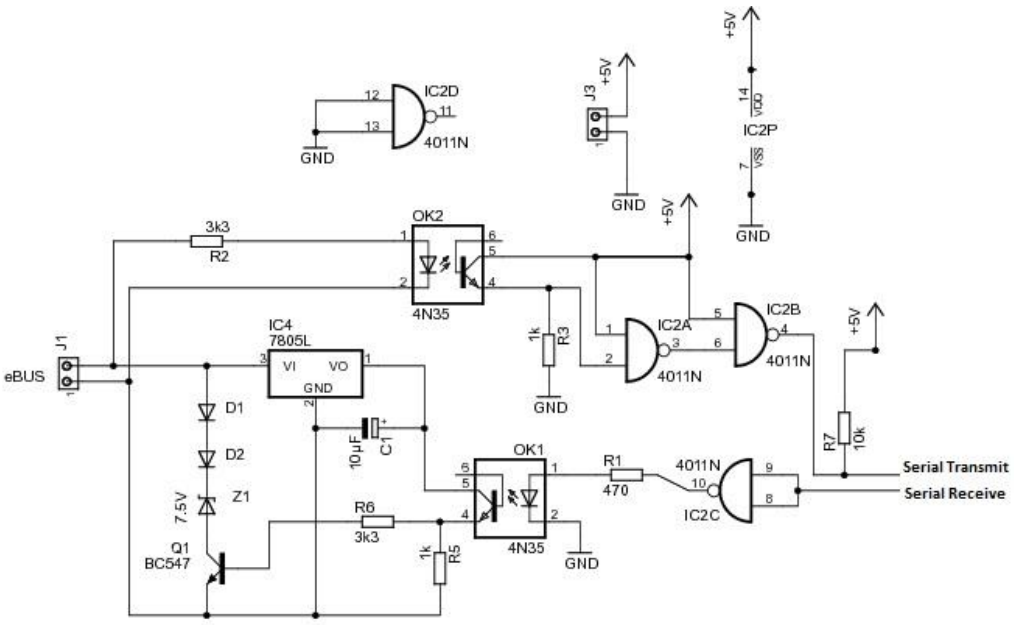
logical 1 = 15–24 volt (typical: 20 V)

Interfacing to MaxAir

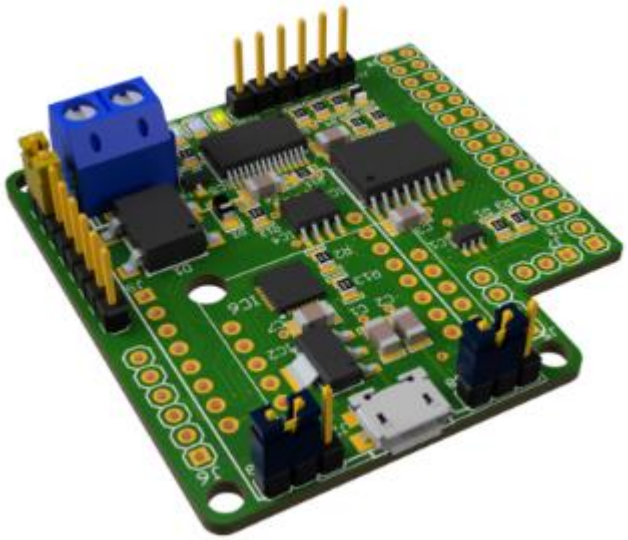
MaxAir need to be able to communicate with eBUS at both the hardware and software levels.

Hardware

It is possible to construct a relatively simple interface to connect an eBUS interface to the MaxAir controller's serial interface:



An alternative is to purchase a ready built and slightly more sophisticated version from [Welcome to eBUS Adapter 3! (ebusd.eu)]



Software

It is possible to construct code to send and receive eBUS messages but by far the simplest approach is to use 'ebusd' which is available via [Home · john30/ebusd Wiki (github.com)]

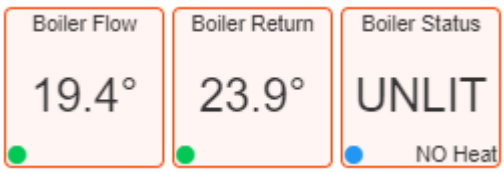
Once this daemon has be installed and configured, it can be added as a background service and can be interacted with by using the 'ebusctl' command.

MaxAir Integration

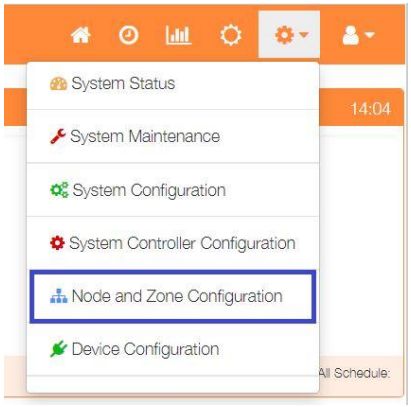
- A new Python script '/var/www/cron/ebus/ebus.py' has been added to interface between 'ebusd' and the MaxAir 'messages_in' queue.
- A new table 'ebus_messages' has been added for the 'ebusd' formatted messages.
- The existing 'Message Sensors' together with the 'Dummy Node' capabilities are used to display the eBUS captured data on the MaxAir Home screen.

Example Implementation

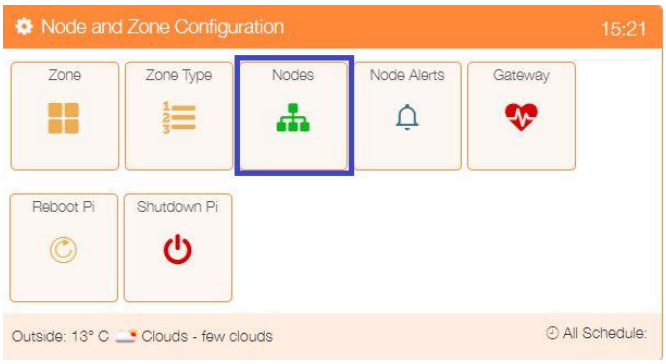
This example will display three tiles on the Home screen, to show Flow Temperature, Return Temperature and Boiler Status.

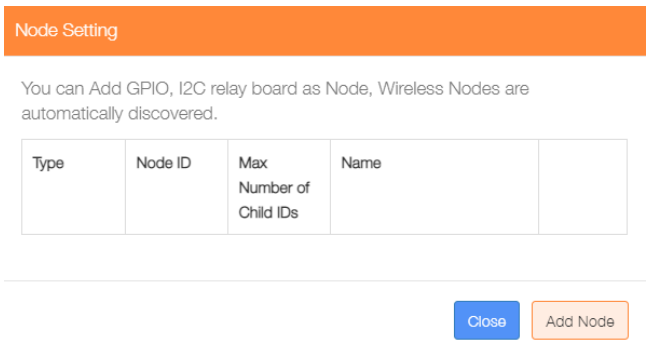


Step 1 – Create a Dummy Node



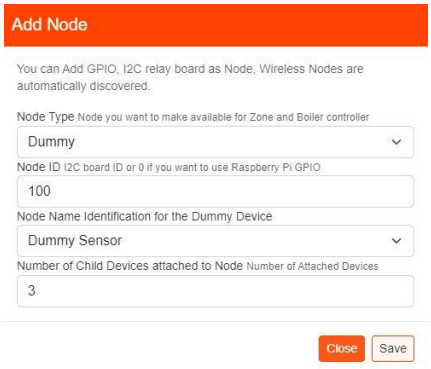
Select ‘Node and Zone Configuration’ from the Settings dropdown list, then click the ‘Sensors’ button.



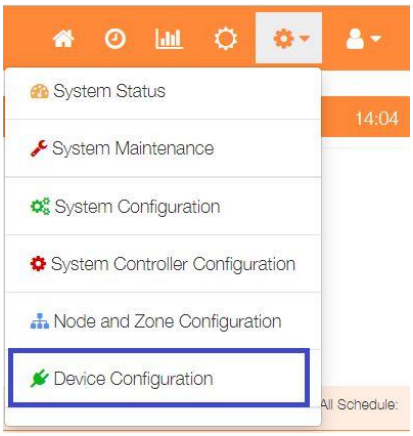


Click on ‘Add Node’.

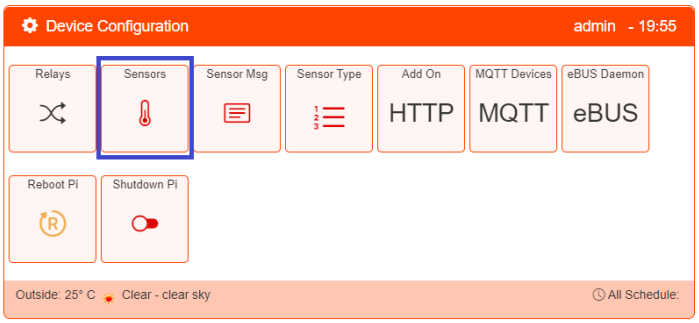
Add a ‘Dummy’ node type, the ‘Node ID’ can be any value not currently in use, select ‘Dummy Sensor’ for the ‘Node Name’ and for this example the ‘Number of Child Devices attached to Node’ will be 3.

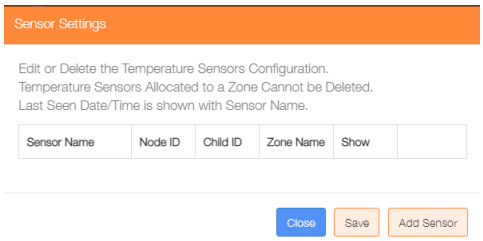


Step 2 – Create Sensors



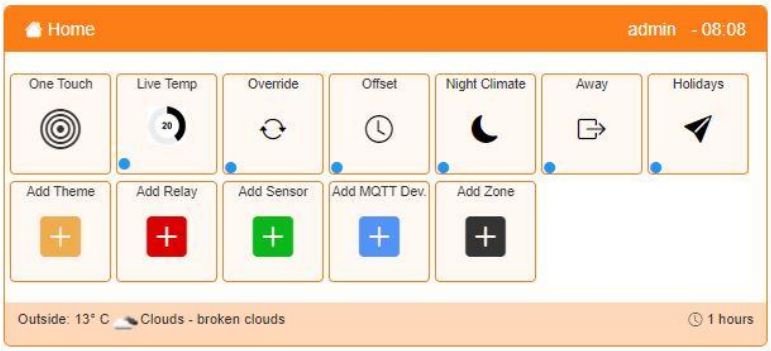
Select ‘Device Configuration’ from the Settings dropdown list, then click the ‘Sensors’ button.



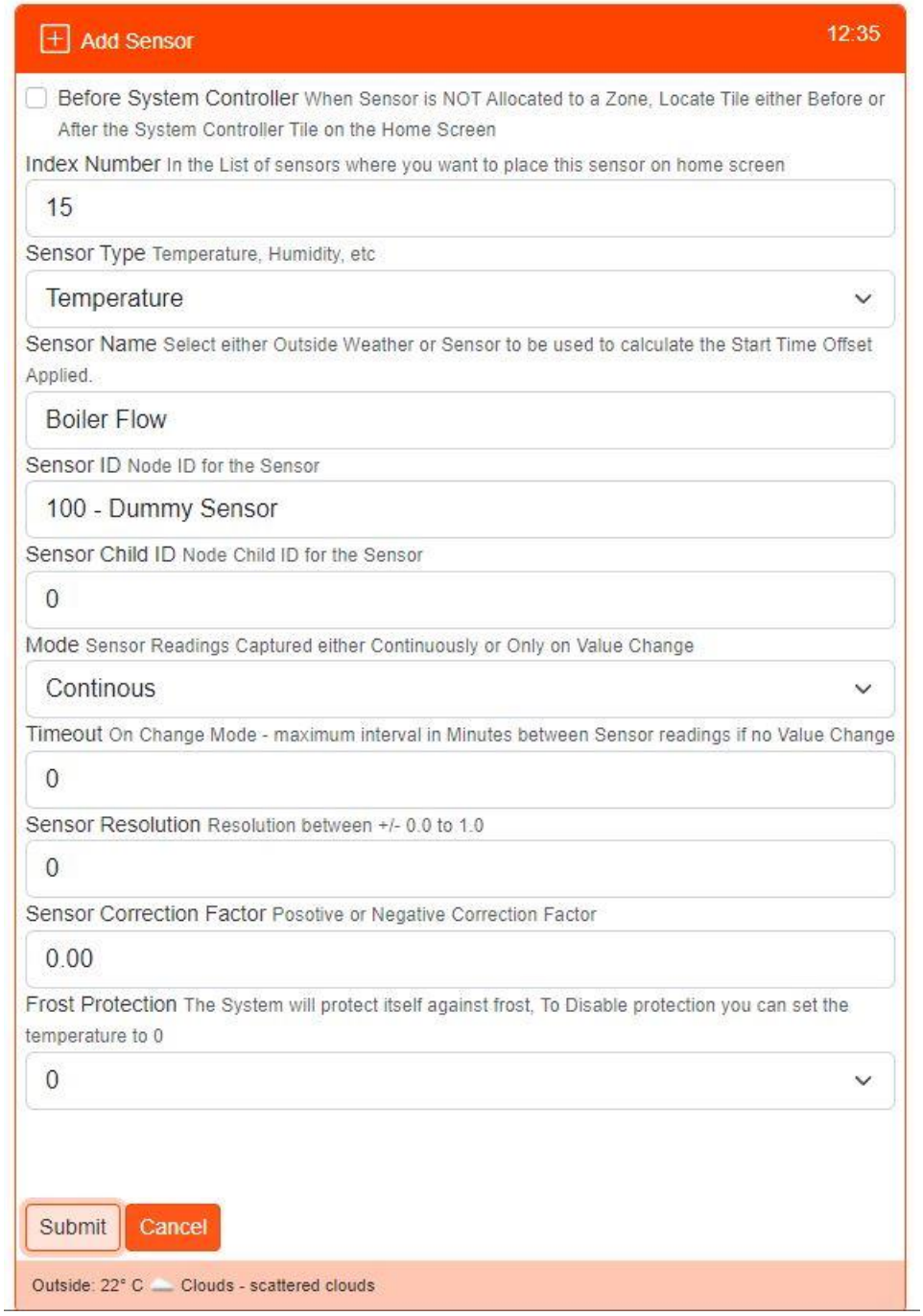


Click on the ‘Add Sensor’ button to configure the first sensor

An alternative method to go directly to the Add Sensor dialogue, is from the Home screen click on the ‘One Touch’ button then select the ‘Add Sensor’ menu item.

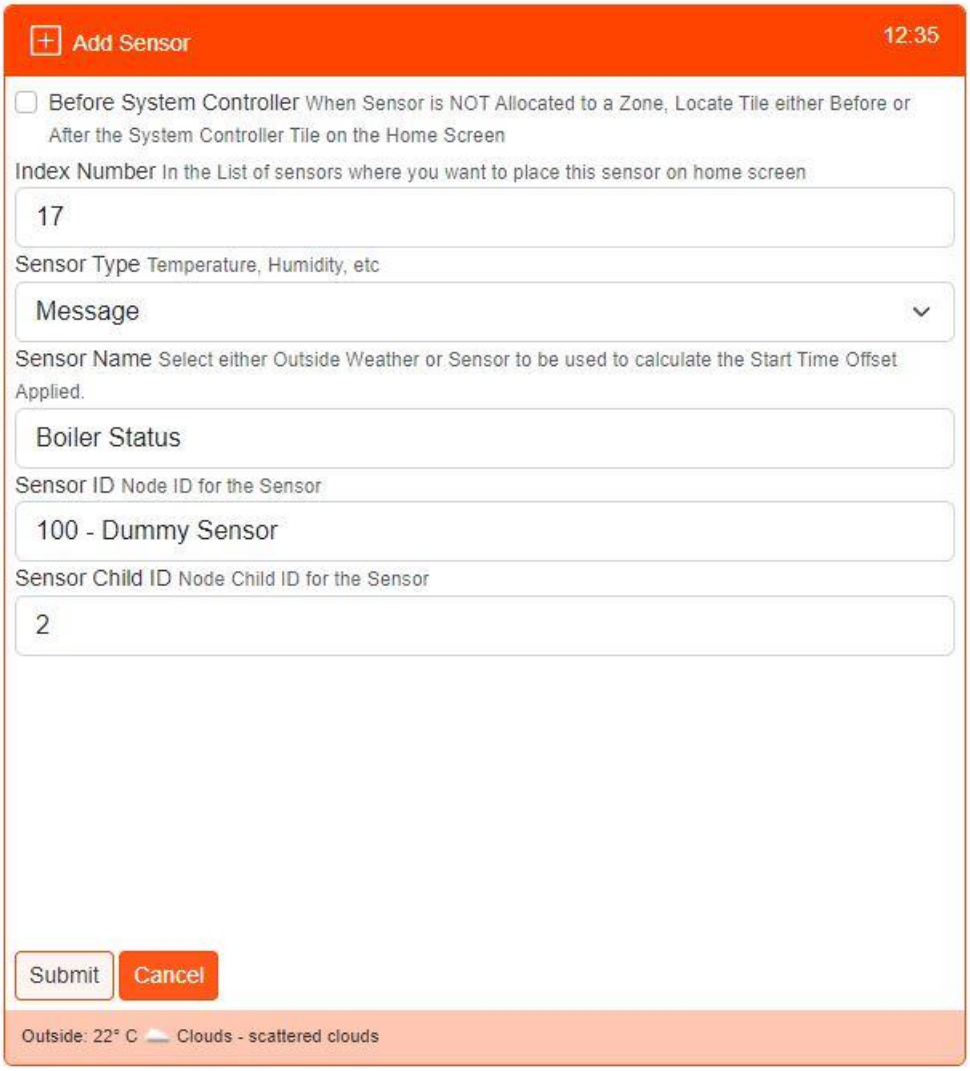


Add a new Sensor for the ‘Boiler Flow’ temperature using the ‘Dummy’ node created above and ‘Sensor Child ID’ set to 0.

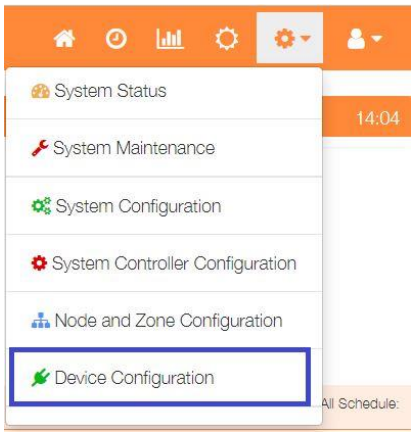


Repeat the procedure for the ‘Boiler Return’, but this time use ‘Sensor Child ID’ set to 1.

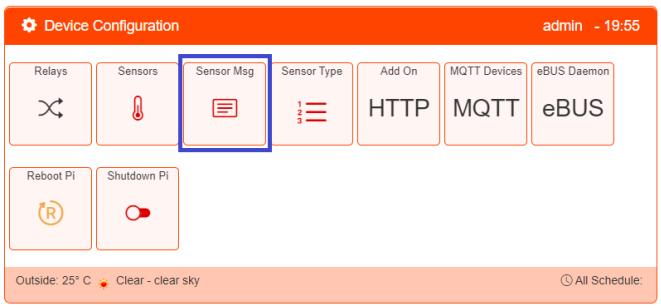
In order to display the Boiler Status we need to create a Message Sensor using the same ‘Dummy’ node created above, in this case use ‘Sensor Child ID’ set to 2.

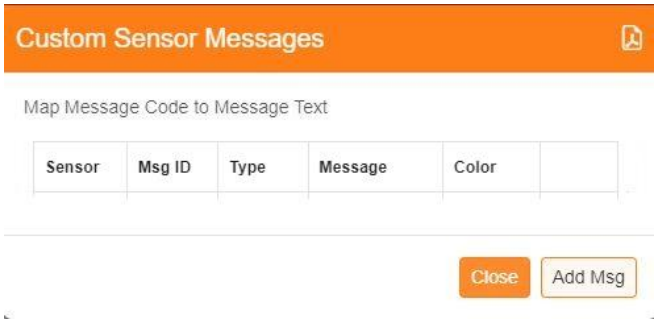


Step 4 – Create Sensors Messages



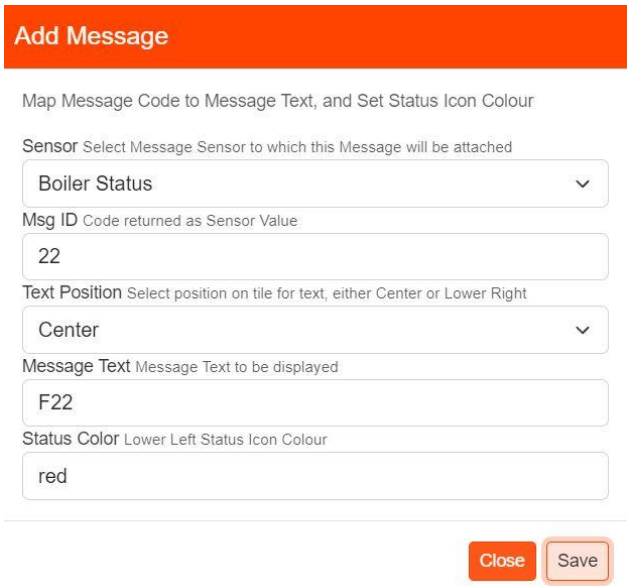
Select ‘Device Configuration’ from the Settings dropdown list, then click the ‘Sensors Msg’ button.





To start building the message mapping, click on the 'Add Msg' button.

For a centre message and associated status icon color:



Select the Message Sensor

Add the Message numeric code

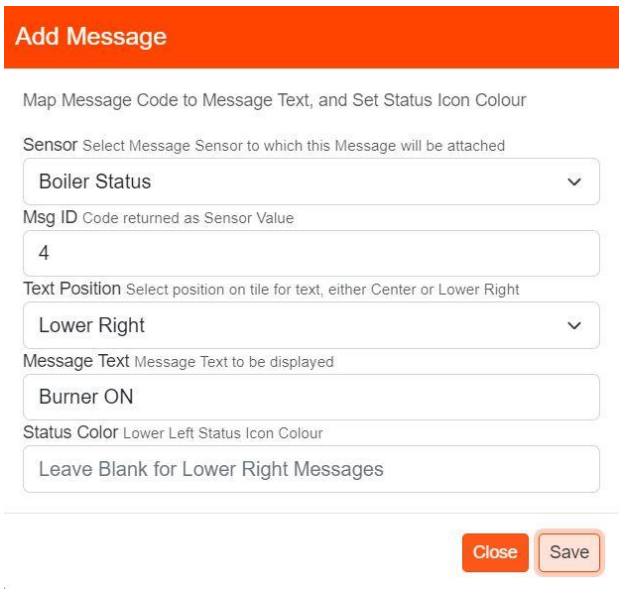
Select message position

Enter the text to be displayed

Set the associated status icon colour (HEX color codes can be used)

Click 'Save' when completed.

For a lower right message:



Select the Message Sensor

Add the Message numeric code

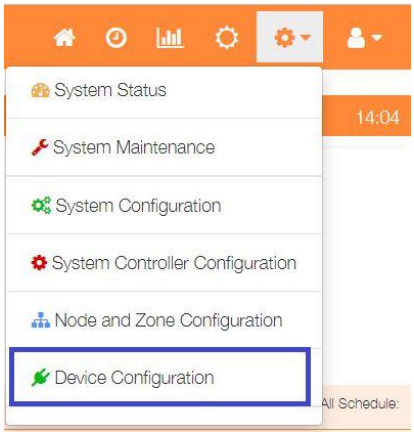
Select '1' for lower right message

Enter the text to be displayed

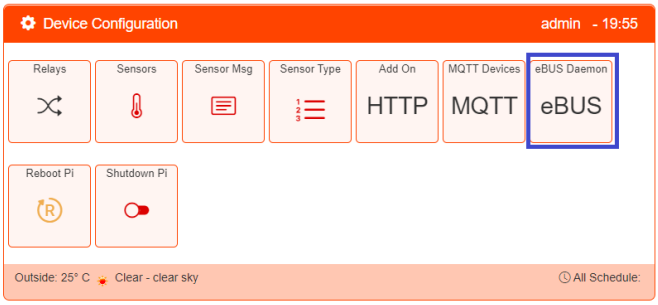
Leave Blank

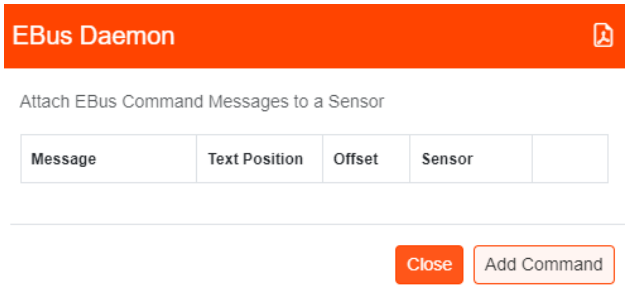
Click 'Save' when completed.

Step 5 – Create eBUS Command Messages

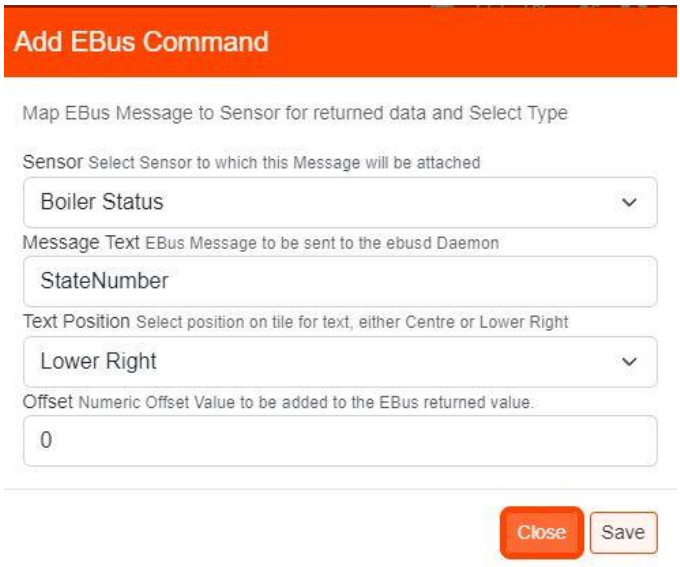


Select 'Device Configuration' from the Settings dropdown list, then click the 'eBUS Daemon' button.





Click on the 'Add Command' button.



Select the Sensor created above

Enter the message to be sent to the 'ebusd' daemon

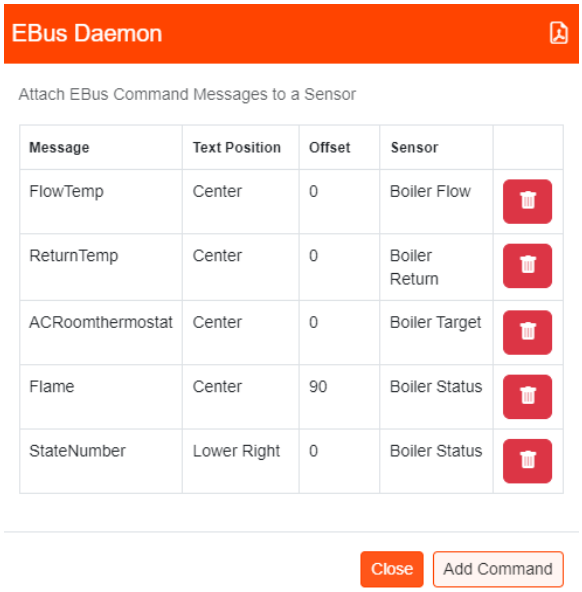
Select which position the response data will be positioned

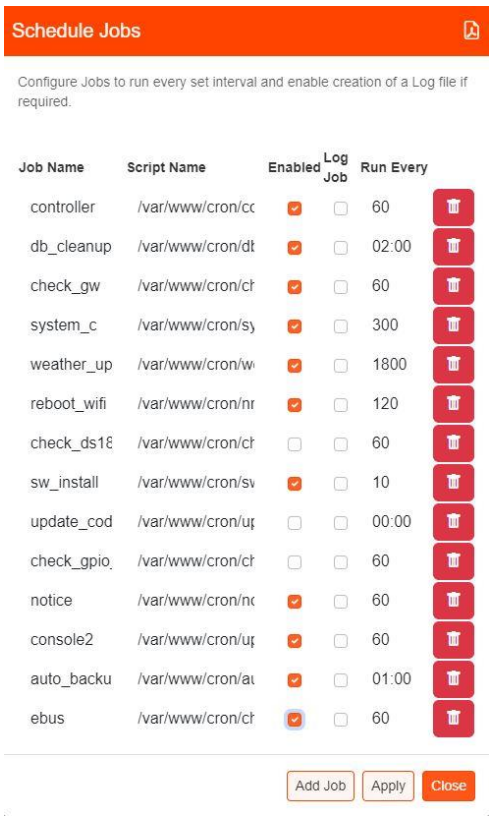
Add any numeric offset to be applied to the response data

Click on 'Save' to store new message

Repeat for any other eBUS messages to be used.

Below is an example of typical eBUS and Custom Sensor message entries.





Step 6 – Enable the 'ebus' Job in the scheduler.