

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

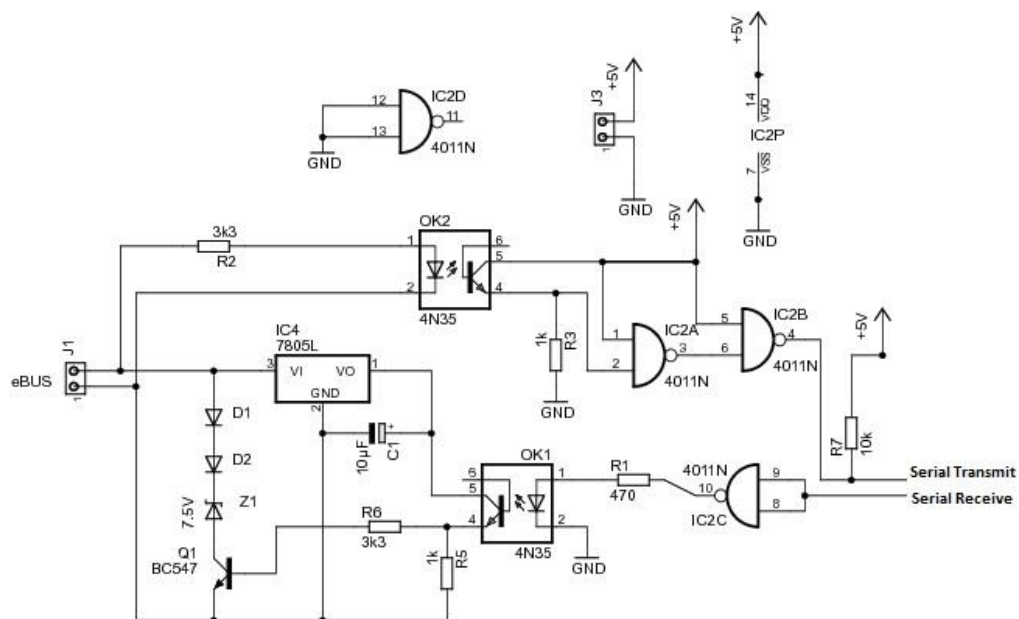
logical 1 = 15–24 volt (typical: 20 V)

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! \(eBUSd.eu\)](http://Welcome%20to%20eBUS%20Adapter%203%21%20%28eBUSd.eu%29)



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 \(github.com\)](https://github.com/john30/eBUSd/wiki)

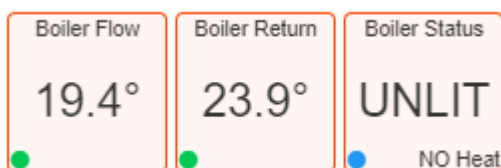
Once this daemon has been installed and configured, it can be added as a background service and can be interacted with by using the 'ebusctl' command.

MaxAir Integration

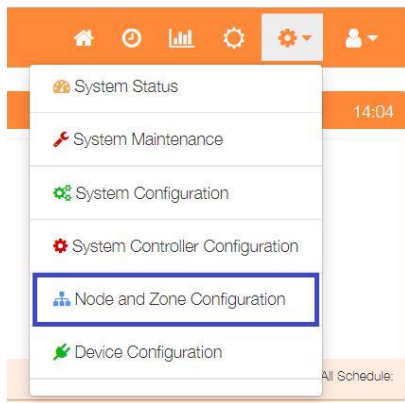
- A new Python script '/var/www/cron/ebus/ebus.py' has been added to interface between 'ebusd' and the MaxAir 'messages_in' queue.
- A new table 'ebus_messages' has been added for the 'ebusd' formatted messages.
- The existing 'Message Sensors' together with the 'Dummy Node' capabilities are used to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

Example Implementation

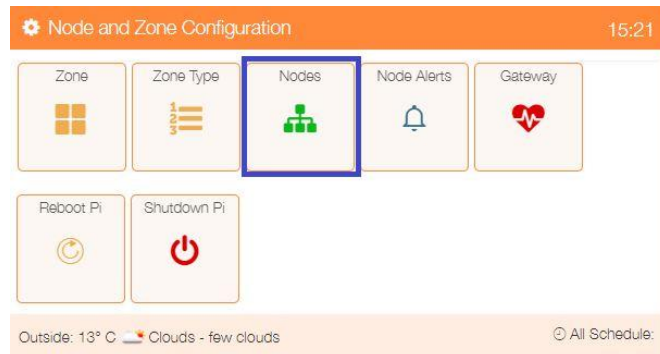
This example will display three tiles on the Home screen, to show Flow Temperature, Return Temperature and Boiler Status.



Step 1 – Create a Dummy Node



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



Node Setting

You can Add GPIO, I2C relay board as Node, Wireless Nodes are automatically discovered.

Type	Node ID	Max Number of Child IDs	Name	

CloseAdd Node

Click on 'Add Node'.

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

Add Node

You can Add GPIO, I2C relay board as Node, Wireless Nodes are automatically discovered.

Node Type Node you want to make available for Zone and Boiler controller:
Dummy

Node ID I2C board ID or 0 if you want to use Raspberry Pi GPIO:
100

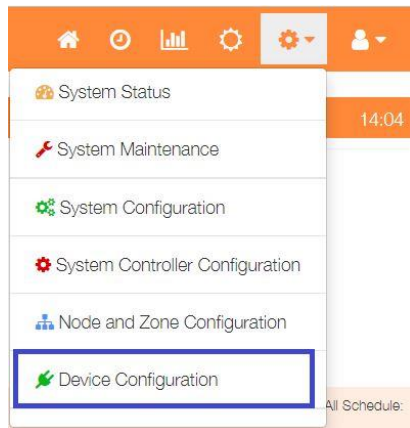
Node Name Identification for the Dummy Device:
Dummy Sensor

Number of Child Devices attached to Node Number of Attached Devices:
3

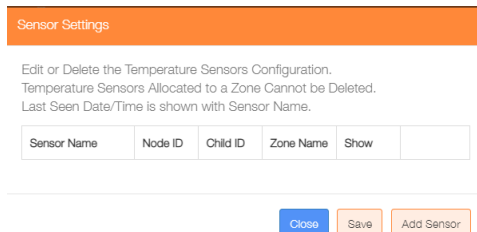
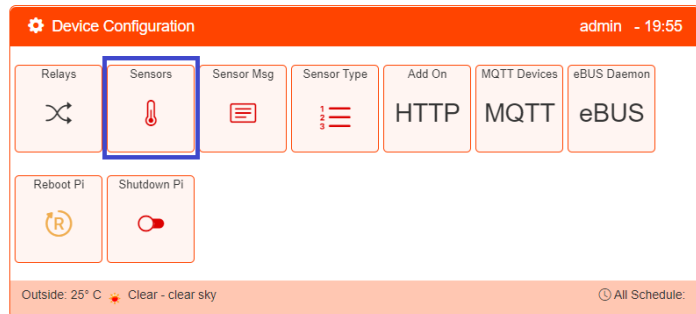
CloseSave

value
Name
Child

Step 2 – Create Sensors

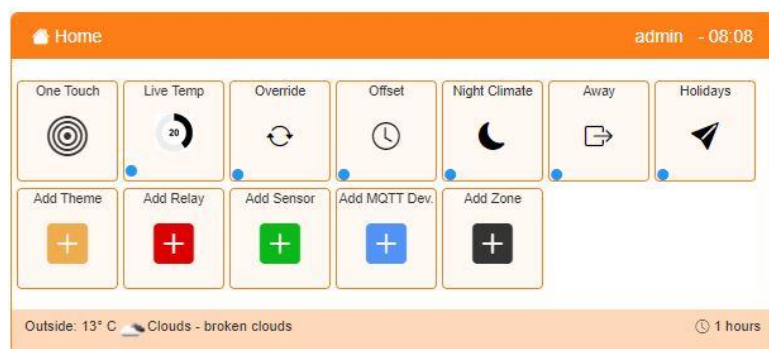


Select 'Device Configuration' from the Settings dropdown list, then click the 'Sensors' button.



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.

Add Sensor

12:35

☐ Before System Controller When Sensor is NOT Allocated to a Zone, Locate Tile either Before or After the System Controller Tile on the Home Screen

Index Number In the List of sensors where you want to place this sensor on home screen

15

Sensor Type Temperature, Humidity, etc

Temperature

Sensor Name Select either Outside Weather or Sensor to be used to calculate the Start Time Offset Applied.

Boiler Flow

Sensor ID Node ID for the Sensor

100 - Dummy Sensor

Sensor Child ID Node Child ID for the Sensor

0

Mode Sensor Readings Captured either Continuously or Only on Value Change

Continuous

Timeout On Change Mode - maximum interval in Minutes between Sensor readings if no Value Change

0

Sensor Resolution Resolution between +/- 0.0 to 1.0

0

Sensor Correction Factor Positive or Negative Correction Factor

0.00

Frost Protection The System will protect itself against frost, To Disable protection you can set the temperature to 0

0

Submit

Cancel

Outside: 22° C ☁ Clouds - scattered clouds

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

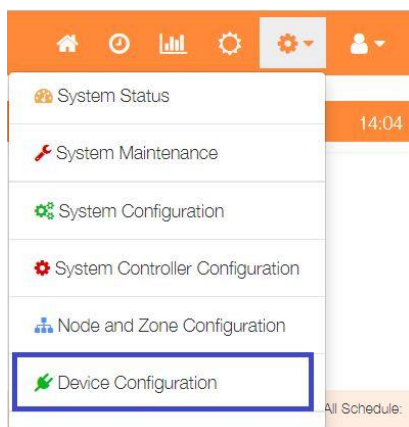
In order to display the Boiler Status we need to create a Message Sensor using the same 'Dummy' node created above, in this case use 'Sensor Child ID' set to 2.

The screenshot shows the 'Add Sensor' form with the following fields and values:

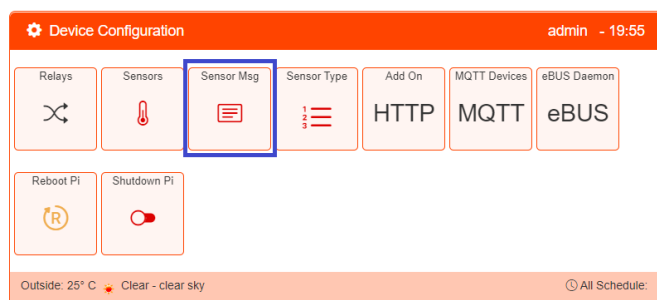
- Index Number:** 17
- Sensor Type:** Message
- Sensor Name:** Boiler Status
- Sensor ID:** 100 - Dummy Sensor
- Sensor Child ID:** 2


Buttons at the bottom: Submit, Cancel. Status bar at the bottom: Outside: 22° C Clouds - scattered clouds.

Step 4 – Create Sensors Messages



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



Custom Sensor Messages


Map Message Code to Message Text

Sensor	Msg ID	Type	Message	Color	

Close
Add Msg

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

For a centre message and associated status icon color:

Add Message

Map Message Code to Message Text, and Set Status Icon Colour

Sensor Select Message Sensor to which this Message will be attached
Boiler Status

Msg ID Code returned as Sensor Value
22

Text Position Select position on tile for text, either Center or Lower Right
Center

Message Text Message Text to be displayed
F22

Status Color Lower Left Status Icon Colour
red

Close
Save

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:

Add Message

Map Message Code to Message Text, and Set Status Icon Colour

Sensor Select Message Sensor to which this Message will be attached
Boiler Status

Msg ID Code returned as Sensor Value
4

Text Position Select position on tile for text, either Center or Lower Right
Lower Right

Message Text Message Text to be displayed
Burner ON

Status Color Lower Left Status Icon Colour
Leave Blank for Lower Right Messages

Close
Save

Select the Message Sensor

Add the Message numeric code

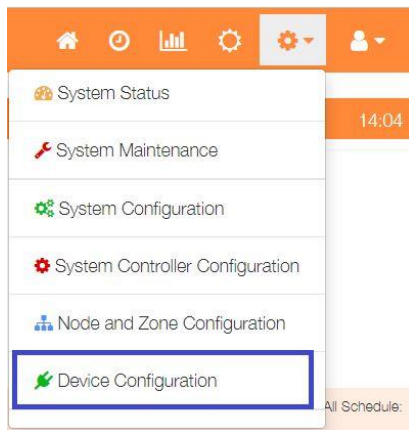
Select '1' for lower right message

Enter the text to be displayed

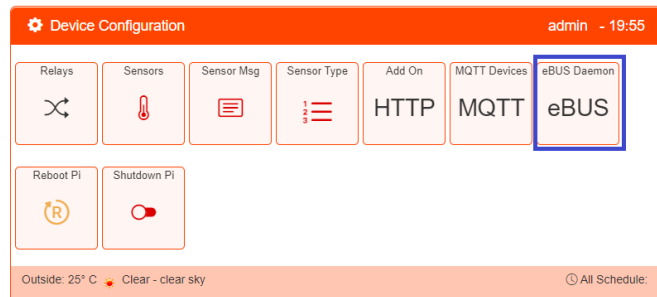
Leave Blank

Click 'Save' when completed.

Step 5 – Create eBUS Command Messages



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



EBus Daemon

Attach EBus Command Messages to a Sensor

Message	Text Position	Offset	Sensor

Close Add Command

Click on the 'Add Command' button.

Add EBus Command

Map EBus Message to Sensor for returned data and Select Type

Sensor Select Sensor to which this Message will be attached

Boiler Status

Message Text EBus Message to be sent to the ebusd Daemon

StateNumber

Text Position Select position on tile for text, either Centre or Lower Right

Lower Right

Offset Numeric Offset Value to be added to the EBus returned value.

0

Close Save

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.

EBus Daemon

Attach EBus Command Messages to a Sensor

Message	Text Position	Offset	Sensor	
FlowTemp	Center	0	Boiler Flow	
ReturnTemp	Center	0	Boiler Return	
ACRoomthermostat	Center	0	Boiler Target	
Flame	Center	90	Boiler Status	
StateNumber	Lower Right	0	Boiler Status	

Close
Add Command

Custom Sensor Messages

Map Message Code to Message Text

Sensor	Msg ID	Type	Message	Color	
Boiler Status	22.00	0	F22	red	
Boiler Status	0.00	0	CLEAR	#00C853	
Boiler Status	72.00	0	F11	red	
Boiler Status	0.00	1	NO Heat		
Boiler Status	1.00	1	Fan Pre		
Boiler Status	2.00	1	Pump Pre		
Boiler Status	3.00	1	Ignition		
Boiler Status	4.00	1	Burner ON		
Boiler Status	5.00	1	Overrun		
Boiler Status	6.00	1	Fan Over		
Boiler Status	7.00	1	Pump Over		
Boiler Status	8.00	1	Anti-Cycle		
Boiler Status	30.00	1	NO Heat		
Boiler Status	53.00	1	Waiting		
Boiler Status	98.00	1	Over Heat		
Boiler Status	90.00	0	UNLIT	#2198F3	
Boiler Status	91.00	0	LIT	red	

Close
Add Msg

Schedule Jobs

Configure Jobs to run every set interval and enable creation of a Log file if required.

Job Name	Script Name	Enabled	Log Job	Run Every	
controller	/var/www/cron/cc	<input checked="" type="checkbox"/>	<input type="checkbox"/>	60	
db_cleanup	/var/www/cron/dt	<input checked="" type="checkbox"/>	<input type="checkbox"/>	02:00	
check_gw	/var/www/cron/ct	<input checked="" type="checkbox"/>	<input type="checkbox"/>	60	
system_c	/var/www/cron/sy	<input checked="" type="checkbox"/>	<input type="checkbox"/>	300	
weather_up	/var/www/cron/w	<input checked="" type="checkbox"/>	<input type="checkbox"/>	1800	
reboot_wifi	/var/www/cron/nr	<input checked="" type="checkbox"/>	<input type="checkbox"/>	120	
check_ds18	/var/www/cron/ct	<input type="checkbox"/>	<input type="checkbox"/>	60	
sw_install	/var/www/cron/sv	<input checked="" type="checkbox"/>	<input type="checkbox"/>	10	
update_cod	/var/www/cron/uq	<input type="checkbox"/>	<input type="checkbox"/>	00:00	
check_gpio	/var/www/cron/ct	<input type="checkbox"/>	<input type="checkbox"/>	60	
notice	/var/www/cron/hc	<input checked="" type="checkbox"/>	<input type="checkbox"/>	60	
console2	/var/www/cron/uq	<input checked="" type="checkbox"/>	<input type="checkbox"/>	60	
auto_backu	/var/www/cron/at	<input checked="" type="checkbox"/>	<input type="checkbox"/>	01:00	
ebus	/var/www/cron/ct	<input checked="" type="checkbox"/>	<input type="checkbox"/>	60	

Add Job
Apply
Close

Step 6 – Enable the ‘ebus’ Job in the scheduler.