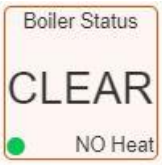
MaxAir Technical – Custom Message Sensors

'Message Sensors' provide the ability to display external text information on a Home Screen sensor tile.



For example, it is possible to display external status information captured from an interface to the boiler. Due to the format of the 'messages_in' table, this information is passed as a numeric code, which must be converted to the message to be displayed on the tile'

There are four areas on the tile that accept data:

1. The Tile Name.
2. The centre text area.
3. The lower left status icon colour.
4. The lower right text area.

The same technique could be used for displaying data from other external sources.

Implementation

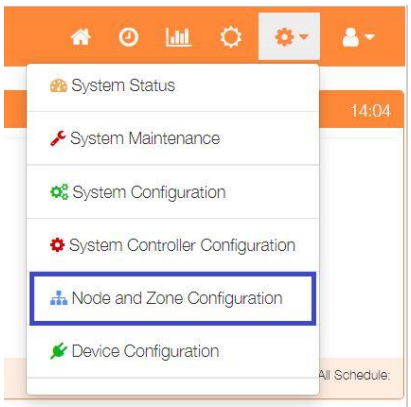
MaxAir

1. A 'Dummy' node will be created.
2. A 'Message Sensor' device will be created and allocated to the 'Dummy' node.
3. Mapping information will be created to place the required information on the sensor tile..

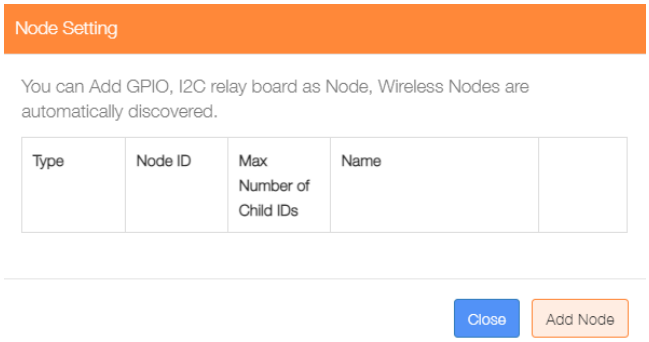
External System

1. The external system will be able to access the MaxAir database from its Python script.
2. The required data in the form of a message code will be captured and used to add an entry to the MaxAir 'messages_in' table, using the 'Dummy' node IDs created above.

MaxAir Configuration

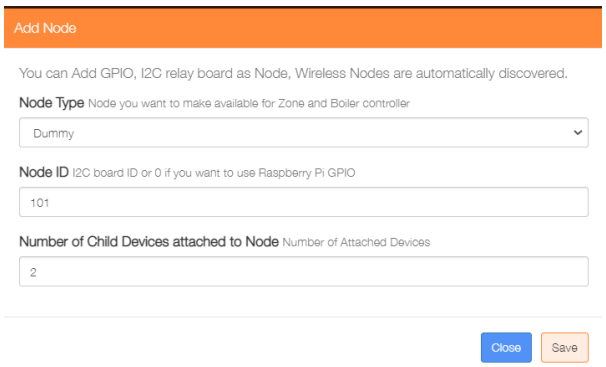


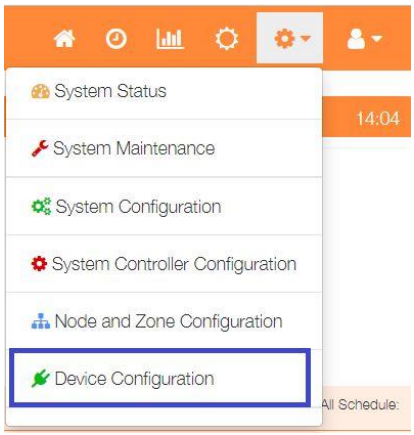
Select 'Node and Zone Configuration' from the Settings dropdown list, then click the 'Sensors' button.



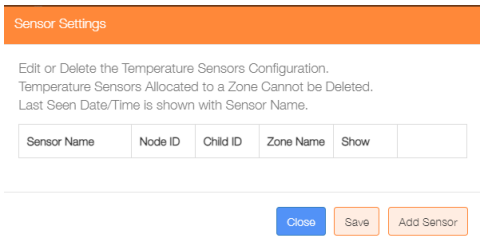
Click on ‘Add Node’.

Add a ‘Dummy’ node type, the ‘Node ID’ can be any value not currently in use, and for this example the ‘Number of Child Devices attached to Node’ will be 1.



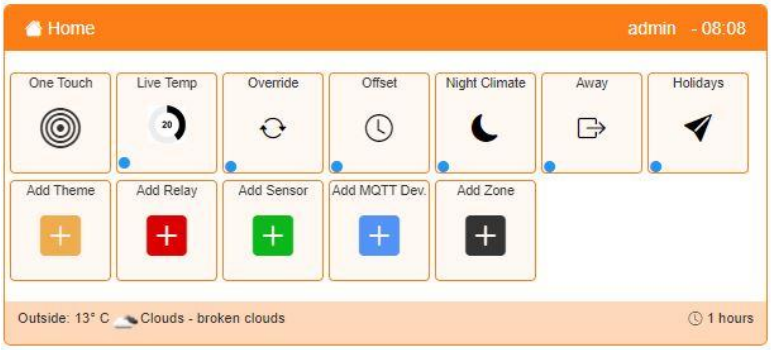


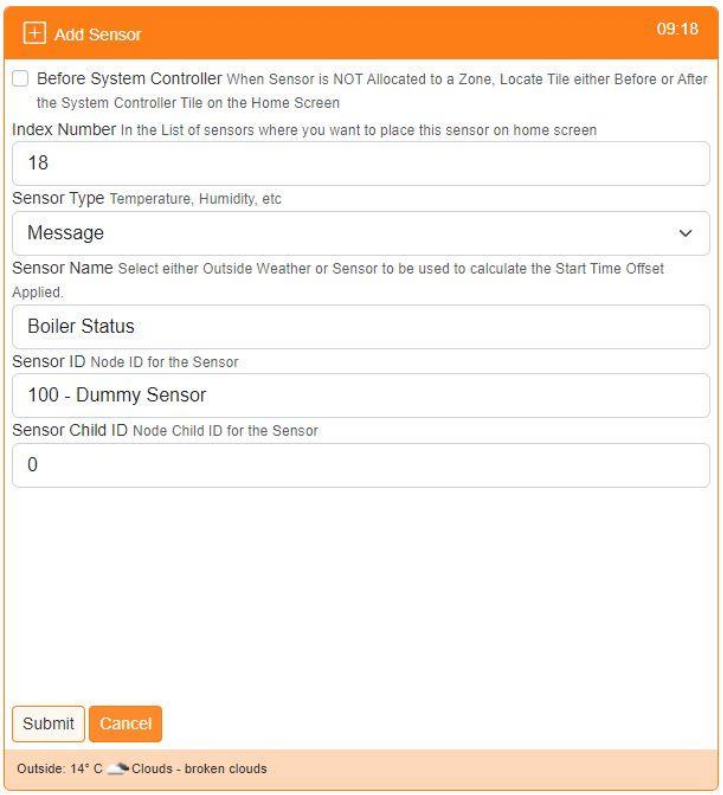
Select ‘Device Configuration’ from the Settings dropdown list, then click the ‘Sensors’ button.



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.





Show before or after the system controller on the Home screen

Used to order where on the Home screen the sensor is displayed

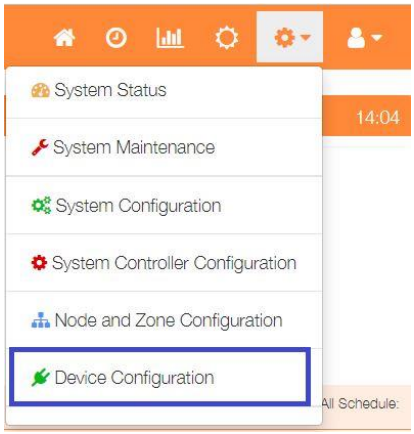
Select 'Message' type

Provide a name for this sensor device

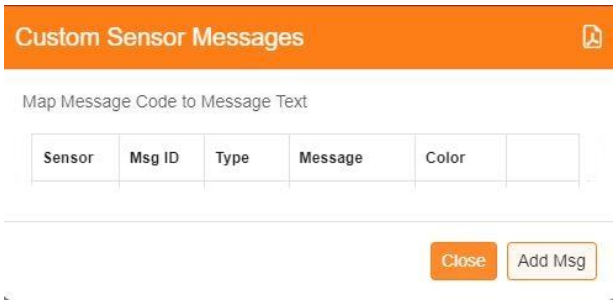
Select the Sensor ID from the dropdown list of available Nodes

Choose the Child ID from the dropdown list, for nodes with only 1 sensor, this will be 0

Click on 'Submit' to add the device.

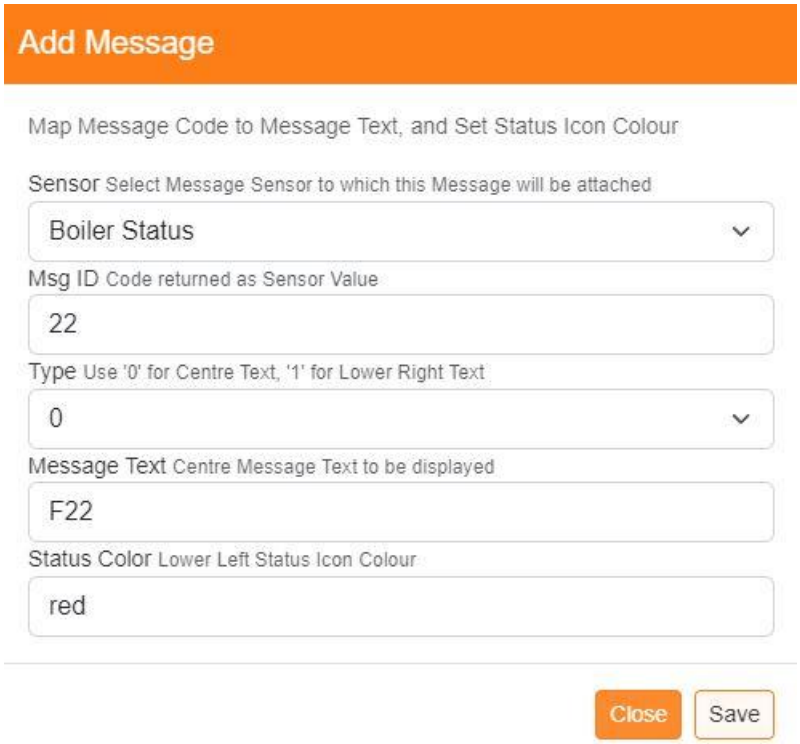


Select 'Device Configuration' from the Settings dropdown list, then click the 'Sensors Msg' button.



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

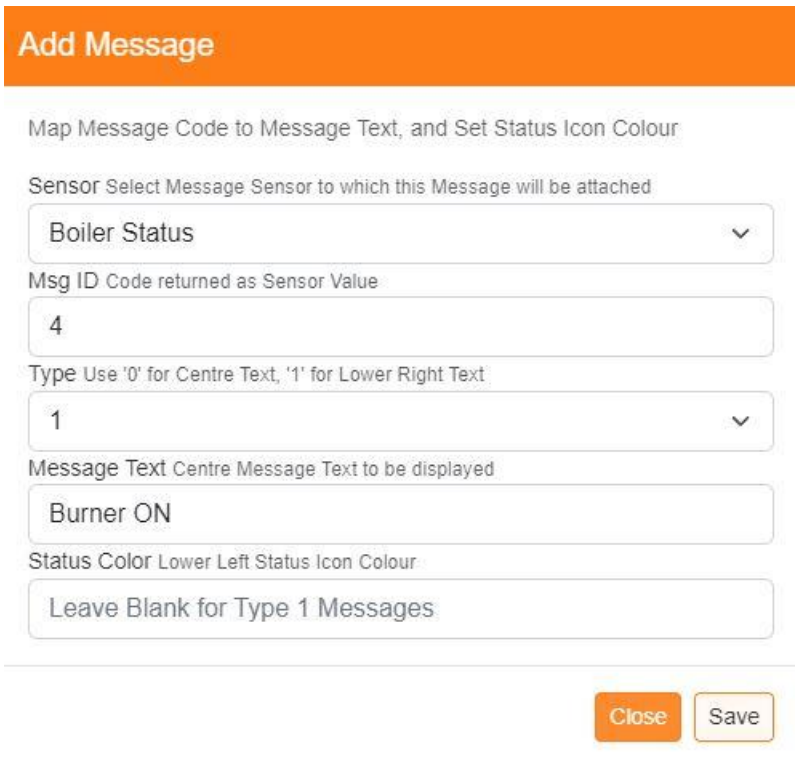
Select ‘0’ for centre message

Enter the text to be displayed

Set the associated status icon colour

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

Set the associated status icon colour

Click ‘Save’ when completed.

Example Python Script to Update the MaxAir Database

```
#!/usr/bin/env python
import time, datetime, MySQLdb
from configparser import ConfigParser

########## Initialise the database access varables ##########
config = ConfigParser()
config.read('/var/www/st_inc/db_config.ini')
servername = config.get('db', 'hostname')
username = config.get('db', 'dbusername')
password = config.get('db', 'dbpassword')
dbname = config.get('db', 'dbname')
nodeID = config.get('db', 'kitchen_node_id')

########## Initialise the database connection ##########
cnx = MySQLdb.connect(host=servername, user=username, passwd=password, db=dbname)

########## Find the node and child ids for the dummy sensors used to pass data back to the PiHome database ##########
query = ("SELECT * FROM temperature_sensors WHERE name = 'Boiler Status' LIMIT 1;")
cursorselect.execute(query)
results =cursorselect.fetchone()
if cursorselect.rowcount > 0 :
  status_id = int(results[0])
  status_sensor_id = int(results[4])
  status_sensor_child_id = int(results[5])
  cursorselect.execute('SELECT node_id FROM nodes WHERE id = (%s)', (status_sensor_id, ))
  results =cursorselect.fetchone()
  if cursorselect.rowcount > 0 :
    status_node_id = int(results[0])

Loop reading status from boiler and send to MaxAir
while True:
  # Add Error and Current Status to the messages_in table
  e_code = ..... # code here to get error status from boiler
  s_code = ..... # code here to get current state from boiler
  try :
    cursorinsert = cnx.cursor()
    cursorinsert.execute('INSERT INTO messages_in(`sync`, `purge`, `node_id`, `child_id`, `sub_type`, `payload`) VALUES(%s,%s,%s,%s,%s,%s)', (0,0,status_node_id,status_sensor_child_id,0,e_code))
    cursorinsert.close()
    cnx.commit()
  except :
    pass

  # Add Current Status to the messages_in table
  try :
    cursorinsert = cnx.cursor()
    cursorinsert.execute('INSERT INTO messages_in(`sync`, `purge`, `node_id`, `child_id`, `sub_type`, `payload`) VALUES(%s,%s,%s,%s,%s,%s)', (0,0,status_node_id,status_sensor_child_id,1,s_code))
    cursorinsert.close()
    cnx.commit()
  except :
    pass
  time.sleep(1)
```