

Implementing Warehouse Sensors

Difficulty Level: Easy

Objective

Implement simulated temperature and humidity sensors for the warehouse in a smart factory scenario.

Achievements

The skills to be acquired at the end of this module:

- Using the *inject* and *function* nodes in Node-RED to create virtual sensors
- Generating random numbers to emulate sensor data
- Using an MQTT-out node to send the sensor data

1. Preparing the Flow for Parts Warehouse Sensors

In the previous module, we have seen that “flows” in Node-Red are separate tabs to organize our nodes and application functionality. We have also seen that we can give the flows a descriptive name by double clicking the flow tab name.

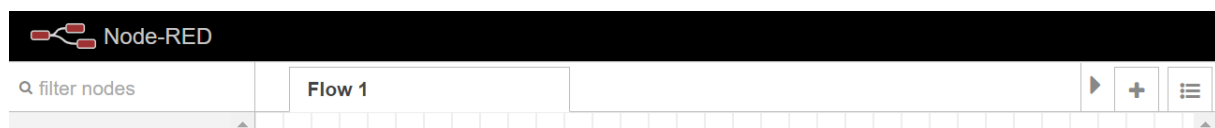


Figure 1. Flow tabs in Node-RED

- Assuming that we start with a clean NodeRED instance, we should have a single flow with the default title “Flow 1”, as shown in Figure 1.
- Double click on the flow tab title, enter “Warehouse Sensors” in the Name field, and click on “Done” button.

We will now implement two virtual sensors (one for temperature and one for humidity) under this NodeRED flow “Warehouse Sensors”.

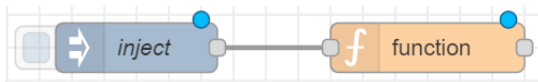
2. Temperature sensor implementation

We start with the temperature sensor.

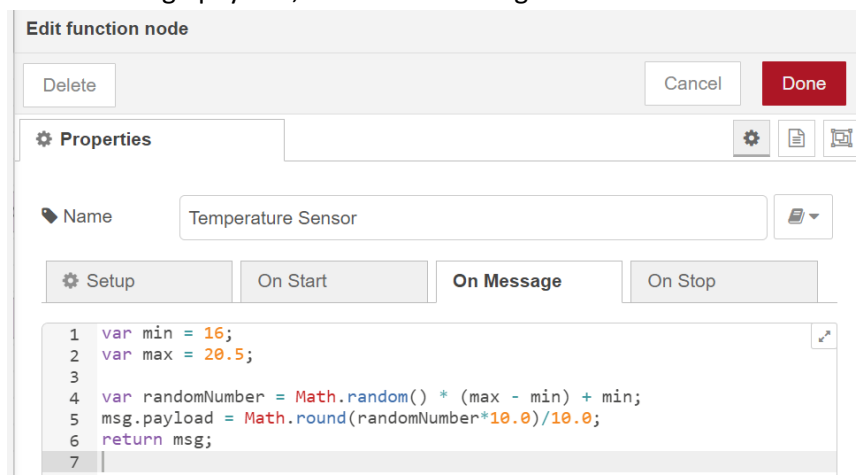
Our emulated temperature sensor will periodically generate and transmit (every 5 sec.) a random value as the current sensor reading. The generated temperature values will be in the range [16°C, 20.5°C] and published to an MQTT broker¹ under the topic ***warehouse/temperature***.

The first step is similar to the simple sensor implementation in the previous module:

- Drag and drop an “inject” node from the node palette.
- Drag and drop a “function” node from the node palette.
- Connect the output of the `inject` node to the input of the `function` node.



- Double click on the “inject” node to configure its frequency. In the settings window that pops up, set the repeat option to “Interval” and set the interval value to 5 sec.
- Double click on the “function” node to open the window for editing its JavaScript code.
- Edit the code to generate a number between 16 and 20.5, with up to one decimal, and add it to our message payload, as shown in the figure below.




- We can also change the node name to “Temperature Sensor” for better clarity of our flow.
- Finally click on “Done”.

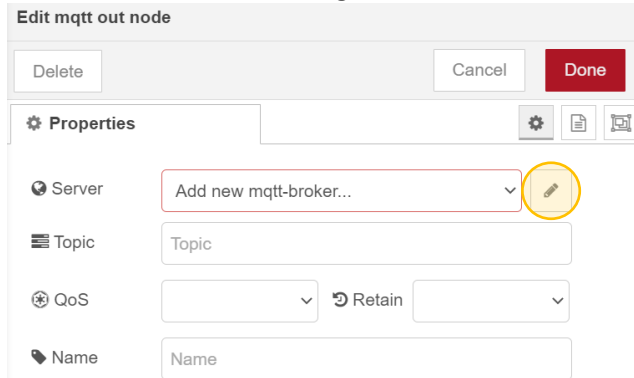
3. Sending the temperature sensor data via MQTT

Now we need to send the messages to the MQTT broker. Node-RED provides an MQTT client in the form of a simple node.

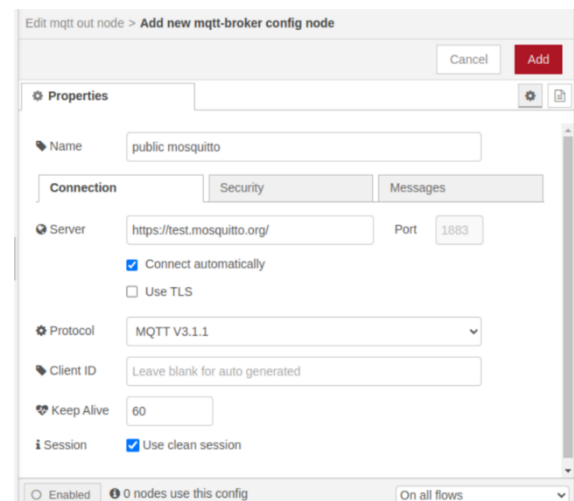
- Add an “mqtt out” node following the “Temperature Sensor” node. Wire them together.

¹ We will use the public MQTT broker at test.mosquitto.org, which will be explained later.

- The “mqtt out” node requires a *configuration* node, a special kind of node that enables some functionality. In this case, it handles the connection to the MQTT broker. To add one, press the edit button  to the right of the Server field inside the node's properties.



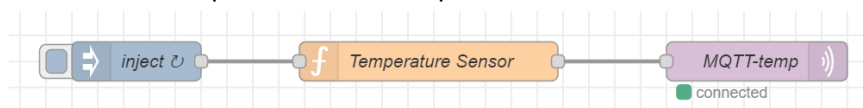
- Add a new MQTT broker connection. A new properties tab with the MQTT connection details should open. Set up the connection with the MQTT broker and click Add/Update. In our case, we will use the publicly available MQTT broker *test.mosquitto.org*. Other parameters are:
 - Server: test.mosquitto.org
 - Port: 1883
 - Use clean session: set
 - Enable SSL/TLS: unset
- After pressing *add*, the “mqtt out” node's properties will open again.
- Set the topic name and the QoS. In our case the parameters are:
 - Topic: warehouse/temperature
 - QoS: 0
 - Retain: false
- Finally, press "Done" to save the configuration.



We now have a virtual sensor that sends MQTT messages with temperature readings. Before proceeding, we can check that the flow works as it should.

- Deploy the changes using the top right button "Deploy".
- Check that the MQTT node connects successfully to the broker. A green square should appear below the node.

The finished temperature sensor implementation should look like something like the following.



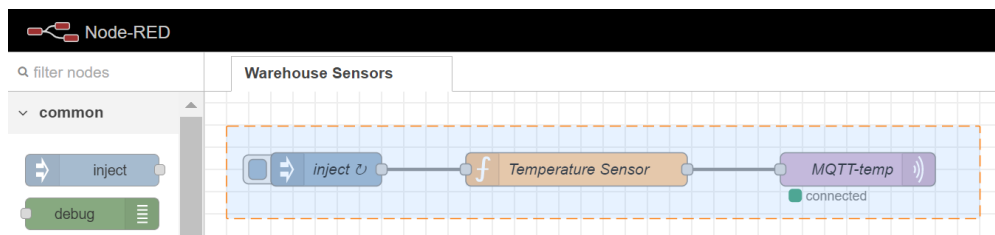
4. Humidity sensor implementation

We will now create a *humidity sensor* under the same “Warehouse Sensors” tab.

Our emulated humidity sensor will periodically generate and transmit (every 10 sec.) a random value as the current sensor reading. The generated temperature values will be in the range [40%, 65%] and published to the MQTT broker under the topic ***warehouse/humidity***.

In this step, we could follow the same steps as in Steps 2-3 above, but instead of repeating all those actions one by one, we can use the shortcut of copying and pasting all nodes and wires for our “Temperature” sensor and edit the new nodes to create a “Humidity” sensor.

- For this, click on an empty area on the NodeRED editor and drag the cursor to cover all nodes in our flow, as shown below:



- Press <Ctrl + C> on your keyboard to copy the selected nodes. You should see the message “3 nodes copied” on the screen.
- Then press <Ctrl + V> to paste the copied nodes.
- The duplicates of the three connected nodes will appear next to your cursor. Move your cursor to “drop” them on a suitable place in the flow. (You can reselect all nodes and move them around again if needed.)

Now we just need to make some changes in the node settings to configure the humidity sensor as needed:

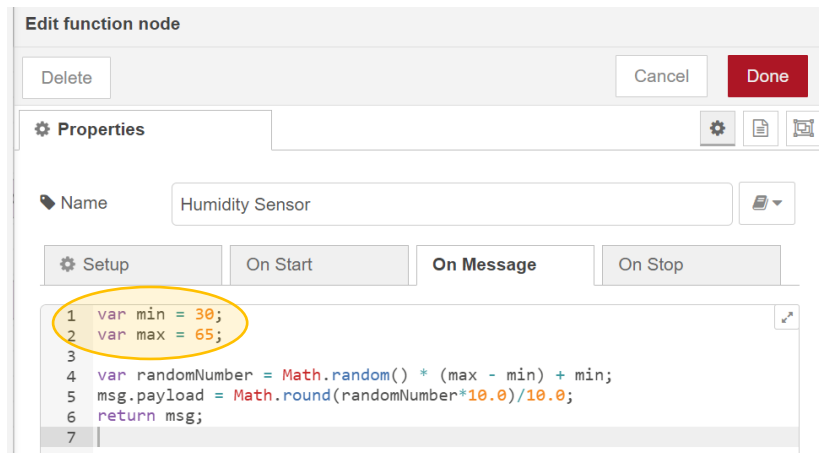
- Inject node:** We would like to avoid the temperature and humidity sensors to simultaneously send their data. Therefore, we add 3 seconds of delay here:

☒ Inject once after seconds, then

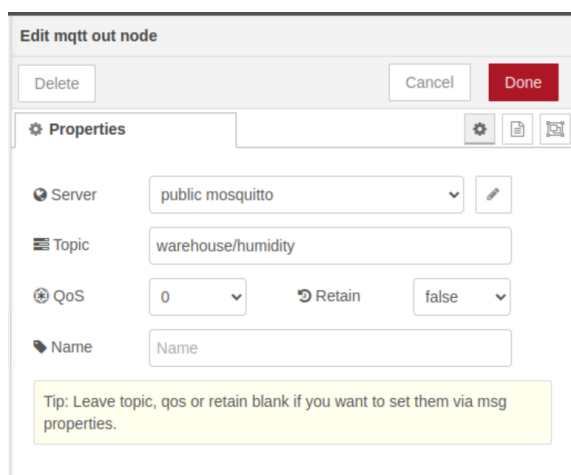
☐ Repeat

every

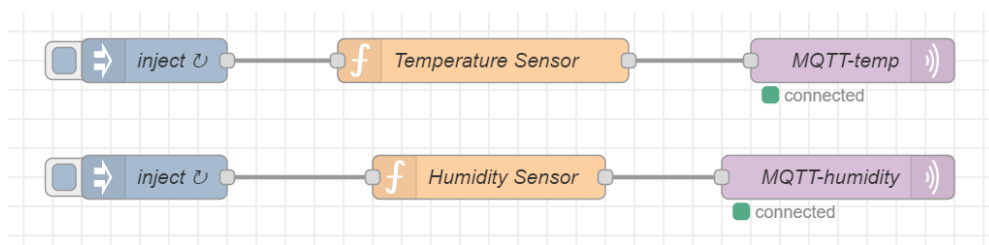
- Sensor emulator node:** Our humidity sensor readings should represent percentage values in the range of 40% to 65%. So we only need to update the min and max variables in the function code accordingly. The rest of the code for random number generation remains the same:



- **MQTT out node:** The only change we need to make here is to publish the sensor readings under the 'humidity' topic instead of 'temperature' topic. So the complete MQTT topic for this virtual sensor becomes "warehouse/humidity":



We have finished implementing the "Humidity" sensor for the parts warehouse. The completed flow should look like the following.



Once we **deploy** the changes in Node-RED, both virtual sensors should start sending data to the MQTT broker.