|  |
| --- |
| Monnit sensors using LuvitRED |
|  |





**Franco Arboleda**

**21-Jun-16**

Table of Contents

1 Introduction 3

2 Hardware configuration 4

3 LuvitRED configuration. 5

3.1 Receiving data from the Monnit sensors. 6

# Introduction

This document explains how to connect Monnit sensors to the CloudGate and use them on LuvitRED.

This document was written using CloudGate’s firmware version 2.62.4 and LuvitRED version 2.8.0. Although older versions of firmware and LuvitRED might work in the same way, we strongly recommend to upgrade to the above mentioned versions or newer in order to ensure the same results.

The following are the different variants of Monnit back slot expansion cards from Option:

|  |  |  |
| --- | --- | --- |
| Option Part # | Frequency Band | WiFi Support? |
| CG2103 | 868 MHz | Yes |
| CG2104 | 900 MHz | Yes |
| CG2105 | 433 MHz | Yes |
| CG2111 | 868 MHz | No |
| CG2112 | 900 MHz | No |
| CG2113 | 433 MHz | No |

Table : Monnit expansion card variants.

The Option card and Monnit’s sensor operate on the ISM 900 MHz (902-928 MHz) band as well as 868 MHz and 433 MHz bands.

The Monnit expansion is shown on Figure 1 below:

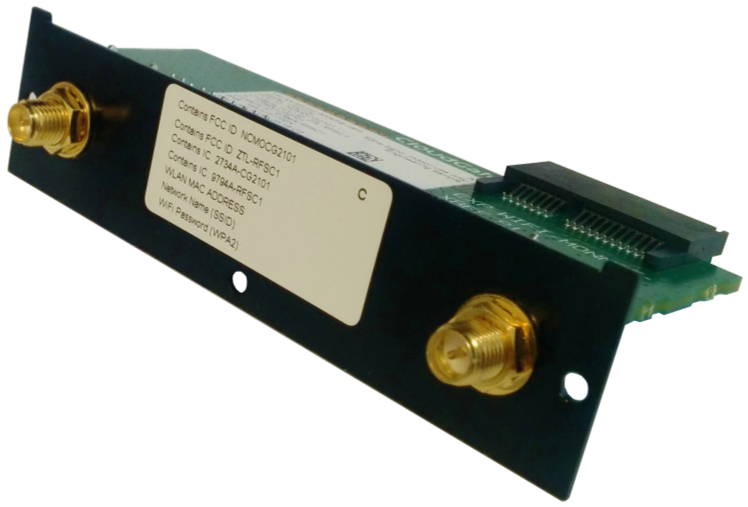


Figure 1: Monnit expansion card with Wi-Fi (CG2104).

For these configurations we are going to use the "Advanced Editor" of LuvitRED.

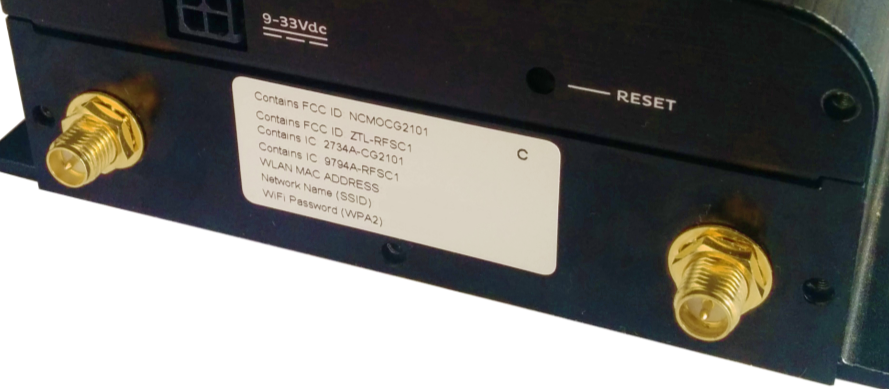
# Hardware configuration

Insert the Monnit card on the back slot of the CloudGate (See Figure 2):



Figure 2: Monnit expansion card with Wi-Fi (CG2104) installed on CloudGate.

Install the correct antenna on the SMA-Male connectors (See Figure 3):



**Monnit antenna** **Wi-Fi antenna**

Figure 3: Correct antenna position.

**Note:** Make sure your sensors do not have the battery inserted in them yet.

# LuvitRED configuration.

Go to the web interface of the CloudGate and then to the "Plugin" tab and sub-tab called "LuvitRED"



Figure 4: Plugin tab, LuvitRED.

**NOTE:** Do not focus on the SNMP (Simple Network Management Protocol) sub-tab. This tab is not going to be used on this document.

Under the “Advanced Editor” of LuvitRED, there are two nodes that are related to Monnit (See Figure 5):

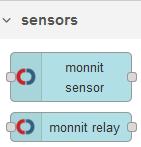


Figure 5: Monnit nodes.

In this document we only focused on the “monnit sensor” node. The “monnit relay” node is used to control the Monnit relay/control sensors, which are not going to be covered in this document.

## Receiving data from the Monnit sensors.

1. Drag and drop a “monnit sensor” node:



Figure 6: monnit sensor node.

1. Configure the monnit sensor node in the following way:

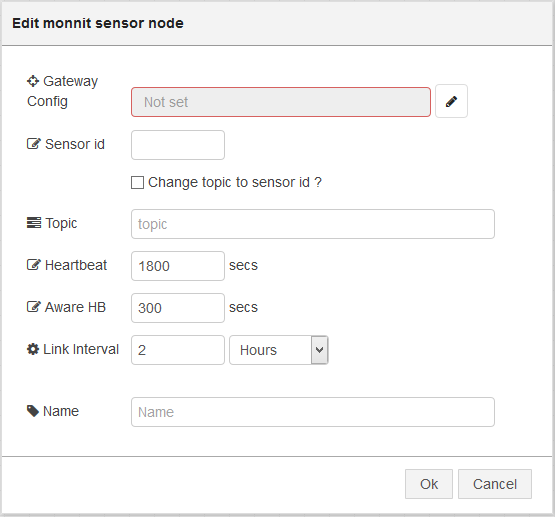


Figure 7: Default monnit sensor node.

* 1. Add a **new** Gateway configuration by clicking on the pencil icon and configure the “Board type” for the **Option Monnit Gateway Back Card** and click on “Add”:

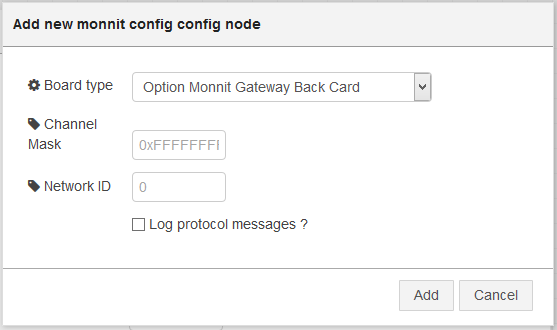


Figure 8: Selecting the back slot card.

* + 1. The “Channel Mask” is by default 0xFFFFFFFF, this means that the CloudGate will allow the 32 available channels in order to find sensors.
    2. The “Network ID” is by default 0, this means that the ID is randomly generated. Change this value manually to make sure there are no conflics when using more than one Gateway. The maximum value for a network ID is **255**.
  1. On the node configuration type the “Sensor id” of the monnit sensor we are going to monitor (See Figure 9 for an example on where to find the sensor information). We are going to use a temperature sensor for this example, because of this, we will type **temperature** as the “Topic”. We are going to change the “Heartbeat” to 10 seconds and the “Aware HB” to 5 seconds (these heartbeats are just for our testing). Finally, we are going to change the “Name” of the node to **Temp\_Sensor**. Click on “OK” when finished (See Figure 10):



Figure 9: Monnit sensor information example (sensor bottom label).

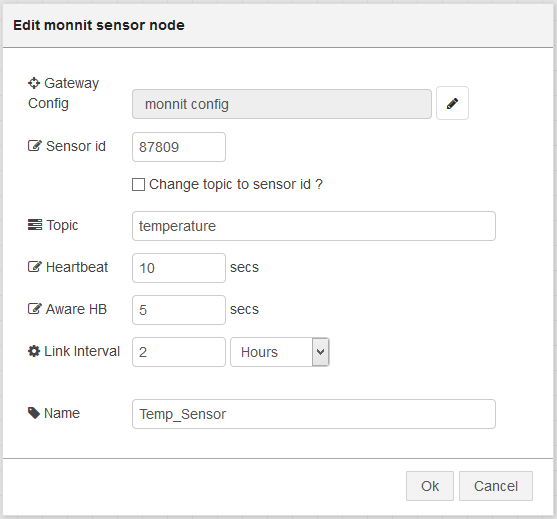


Figure 10: Final node configuration.

1. The “Hearbeat” is the normal reporting interval.
2. The “Aware HB” is a special reporting interval when the sensor enters a certain range. At this moment the range is the one programmed at factory and it could be changed using the iMonnit platform (not via LuvitRED)
3. “Link interval” is the interval to maintain the link between the sensor and the Gateway.

This node is now configured to send out the information from that specific temperature sensor. Now we need to display the data:

1. Drag and drop two debug nodes and configure one to output a “complete msg object” (See Figure 12 - do not change the other debug node):

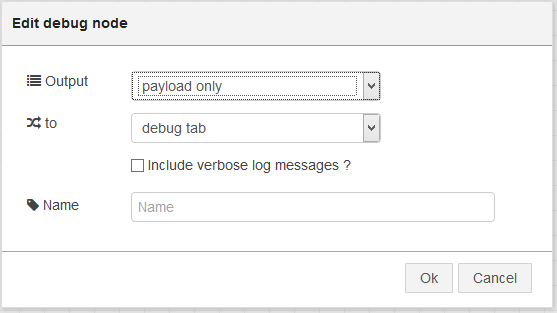


Figure 11: Standard debug node configuration.

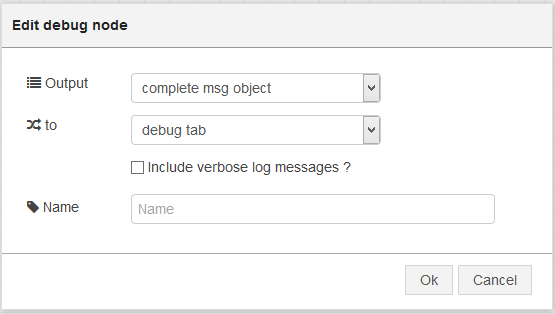


Figure 12: complete msg object debug node configuration.

**Note:** We actually only need one debug node to show the values coming from the sensor, but for this example we are using two debug nodes to compare the information shown by the standard one and by the “complete msg object” one (the information is always there, but the standard debug configuration does not show it all).

1. Let’s connect the nodes together in the following way:

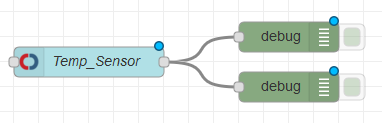


Figure 13: Final Configuration.

1. Click on “Deploy” and wait for the “Temp\_Sensor” node status to say “waiting for sensor”:

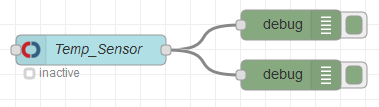


Figure 14: Temp\_Sensor node status "inactive".

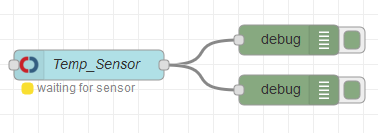


Figure 15: Temp\_Sensor node status "waiting for sensor".

Now, it is time to insert the battery into the sensor and wait for it to register on our configuration (node status becomes “active”):

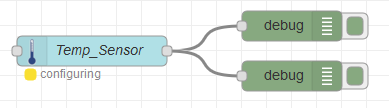


Figure 16: Temp\_Sensor node status "configuring".

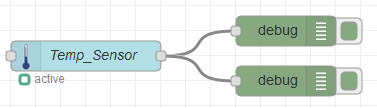


Figure 17: Temp\_Sensor node status "active".

We should now be seeing the values being reported by the sensor under the debug tab:

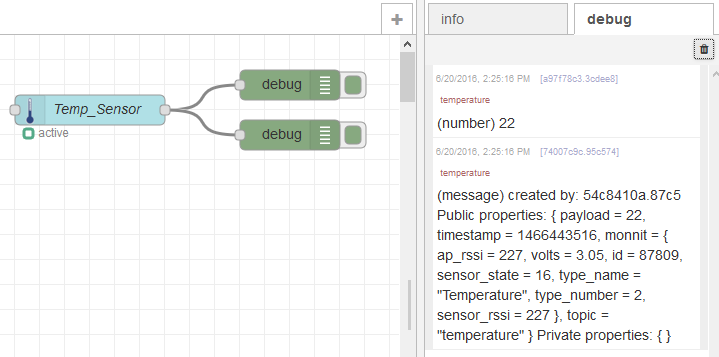


Figure 18: Values coming from the temperature sensor.

The first line is coming from the default debug node as it only shows the payload which in the temperature sensor is just the actual temperature (in Degrees Celsius).

The second line is coming from the debug node configured to output the “complete msg object”, on that line we can see the payload (the value as on the first line), but we can also see in deep information about the sensor inside the monnit section of the message:

(message)created by: 54c8410a.87c5 Public properties: { **payload = 22**, timestamp = 1466443516, **monnit = { ap\_rssi = 227, volts = 3.05, id = 87809, sensor\_state = 16, type\_name = "Temperature", type\_number = 2, sensor\_rssi = 227 }**, topic = "temperature" } Private properties: { }

The information included is:

* ap\_rssi: AP RSSI
* volts: battery voltage
* id: sensor ID
* sensor\_state: sensor state (numeric)
* type\_name: type of sensor (text)
* type\_number: type of sensor (numeric)
* sensor\_rssi: sensor RSSI

More sensors can be included into the configuration using the same steps explained above.