POLITECNICO
MILANO

Internet of Things
2015 – 2016

Project title:

Log sensor data on Node-red

Mohamad Ghaderi
Mohammad Ala Amjadi

The goal is to develop a free, stand-alone and reliable system to manage data coming from a wireless sensor network (WSN) using new tool node-RED. Wiring together hardware devices, APIs and online services for internet of things devices were possible with a visual tool named Node-RED. Node-RED provides a browser-based flow editor that makes it easy to wire together flows using the wide range of nodes in the palette.

Introduction on the project:

The project is made in TinyOS and simulated using Cooja simulator. There are WSN with one sink node and two sensor nodes equipped with a temperature and humidity sensor. Sensed data will be send through the wireless module included in the sensor nodes.

The project is about sensing temperature and humidity through sensors that are attached to the motes connected to the sink mote. Sink mote will receive packets from other motes and send it to the node-red. Then we can monitor live, collect data and process the collected data.

Scenario:

There is a WSN constructed of one Server/Sink node (PAN coordinator) and 2 motes/nodes which each of them is connected to a sensor.

1. At the beginning, motes will boot and try to start their radio interfaces. Each node from which sensor they are connected to, will get data and make a packet with a type ID in order to differ sensors.
2. Sink mote, will wait and listen to the channel for any receiving packets coming from other 2 motes.
3. Other 2 motes will sense from sensors and make a packet which includes the sensing value in the payload and type ID.
4. After making the packet, motes will send their packets through their radio interface.
5. When sink mote receives any packet, it will read the type ID that is included in the packet structure and put the payload in an ID related variable.
6. Sink mote will wait till it gets the other sensor data from other mote and do the same thing as explained in number 5.
7. When sink mote receives from all the sensor motes, it will put the IDs and variables in another packet structure.
8. Sink mote will send the packets via serial interface to the PC.
9. In Cooja apart from simulating nodes to simulate a serial socket server through which we can have TCP connection on local host ports.
10. In the same time we have node-red platform running on the local machine which is listening on the same port as Cooja is sending data.
11. Data receiving in Node-RED are packets in TCP packet format and we can access the payload by parsing the packet payload.
12. Received data are stored in a file called ‘sensorlog.txt’ in the local machine which is accessible both through file browser and Node-RED.
13. In the same time we can see live charts from different cloud base platforms. Real time data can be seen either in a web app connected to Google Drive or another cloud base platform called Dweet, which lets us to visualize our data in dashboard.
14. The real time data are monitors for any exceeding threshold which than can trigger an email alarm in case of Temperature increase or Twitter message in case of high humidity.
15. And at the end we callback our data log or plot them using function provided in Node-RED.

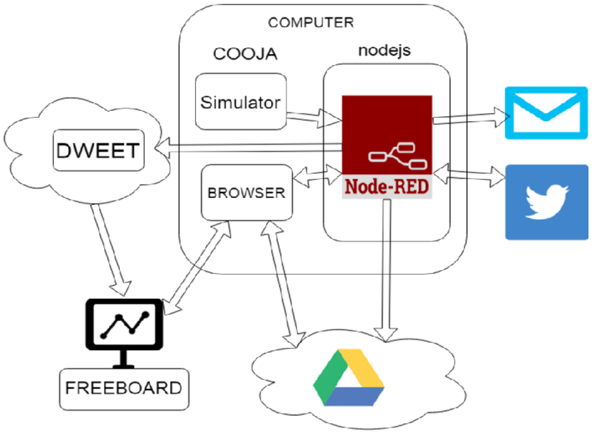
The project file contains:

Files for each node including sink and sensor nodes are in their specific named folder (Sink, Temperature and Humidity).

Each node have header, module and configuration files in their folder. Sensor nodes have specific SensorC and SensorP that provides all the interfaces and commands needed to read from sensors.

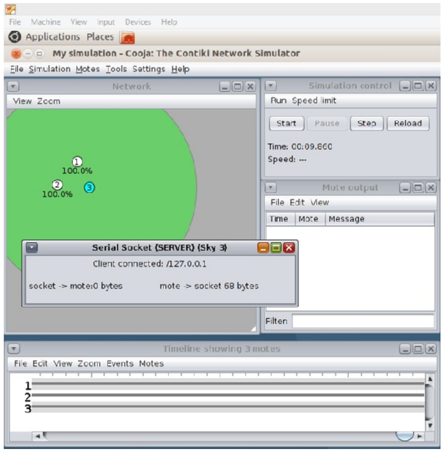
In our case, we don't have real sensors so we are reading data from a header file included in sensors' data folder.

The project can be seen as following diagram which is showing a schematic on how different parts are communicating with each other.



SIMULATION:

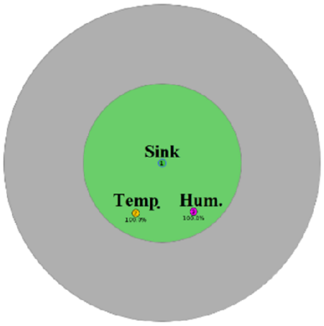
1. At the beginning each sensor node sends its data through radio channel to the sink and the sink sends them through serial channel to computer. This is done by Cooja's serial socket to the Node-RED:


File Machine View Input Devices Help
Applications Places
My simulation - Cooja: The Contiki Network Simulator
File Simulation Motes Tools Settings Help
Network
View Zoom
1
100.0%
2
100.0%
3
Simulation control
Run Speed limit
Start
Pause
Step
Reload
Mote output
File Edit View
Time
Mote
Message
Serial Socket (SERVER) (Sky 3)
Filter:
Timeline showing 3 motes
File Edit View Zoom Events Motes
1
2
3

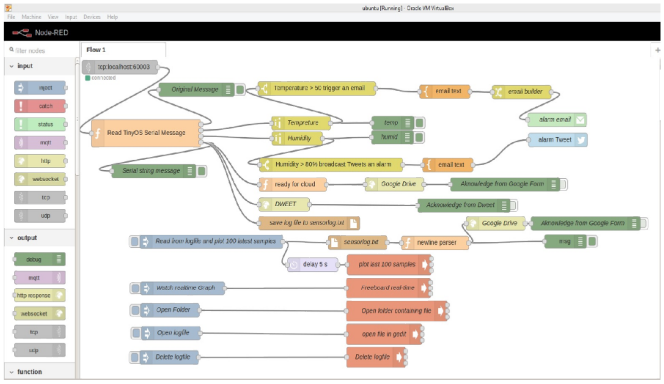
Time: 00:09.860
Speed: ---

Client connected: /127.0.0.1

socket -> mote:0 bytes mote -> socket 68 bytes



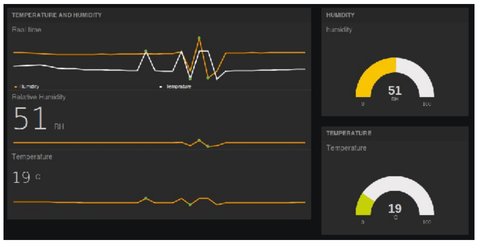
2. Node-RED receives data with listening to the TCP port (60003 in our case) in the same port of Cooja sending data and send data for further usages.



3. Providing lots of functionalities in Node-RED we can Monitor, Visualize, set alarms, store and analyze data and perform different commands to connect our things and automate a lot of tasks and save huge time!

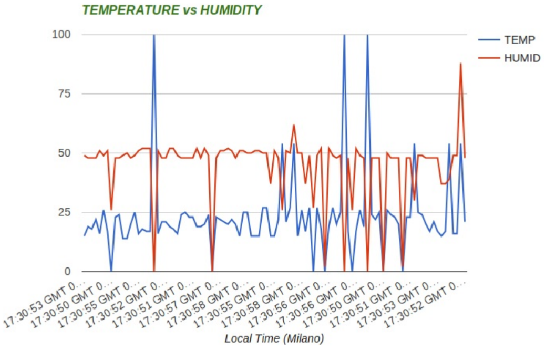
4. Here are visualizing data with two different platforms which are both cloud based:

a. With an HTTP POST we can send our data to DWEET.
Data is sent from Node-RED to the cloud by with a HAPI web API. To send data a URL https://dweet.io/dweet/for/IOT_EXAPLE?temp={{payload.Temperature}}&humid={{payload.Humid}} is called. All query parameters we add to the request will be added as key-value pairs to the content of the dweet. From there we make a dashboard in freeboard.io which is a public dashboard and can be called from Node-RED.



b. At the same time we POST our data to Google drive to make a copy of them in the cloud and save them in Google Drive. This is possible with posting data as an HTTP REQUEST to google forms and make a copy of them.

5. We also save our data to a text file which stores sensor samples and a local time which we generate in Node-RED to be able to retrieve our data later. It is also possible to plot the saved data in visualize it as a web page.
6. Alarms are provided through a threshold which in case of Humidity more than 80% or Temperature more than 50 degrees will send a Tweet or an email respectively.


TEMPERATURE vs HUMIDITY
TEMP
HUMID
17:30:53 GMT 0...
17:30:50 GMT 0...
17:30:55 GMT 0...
17:30:52 GMT 0...
17:30:51 GMT 0...
17:30:57 GMT 0...
17:30:58 GMT 0...
17:30:55 GMT 0...
17:30:58 GMT 0...
17:30:56 GMT 0...
17:30:56 GMT 0...
17:30:50 GMT 0...
17:30:51 GMT 0...
17:30:53 GMT 0...
17:30:52 GMT 0...
Local Time (Milano)