# ACTIVITY5 - IoT

## By Stefano Agresti, Gianluca Guarro

## PERSONAL INFO

**Name**: Stefano Agresti

**Person code**: 10685203

**Matricola**: 913079

-

**Name**: Gianluca Guarro

**Person code**: 10658177

**Matricola**: 918696

-

**Channel number:** 1070801

**Channel link:** https://thingspeak.com/channels/1070801

## ASSIGNMENT

*Create a Cooja simulation with 3 TinyOS (sky) motes, called 1, 2, 3.The motes communicate over the radio. Mote #2 and #3 sends a message every 5 seconds to mote #1. The message contains a random value (between 0 an 100) and a static topic. Mote #1 receives messages and ''forwards'' them to node-red. The node-red dashboard removes the values > 70 and publishes the message to thingspeak via MQTT into two different charts.*

## EXECUTION

### 1. CREATING THE MOTES

The code for this task, contained in "*Activity5.h*", "*Activity5AppC.nc*" and "*Activity5.nc*", closely resembles the one used in Activity #1. The main changes are in the *Receive* and *Send* function: the former now contains a *printf* call, to send messages to *Node-Red*, while the latter uses the *Random* interface to create a random number between 0 and 100. The selection on values greater than 70 is performed in the *Receive* function.

The rest of the code is pretty straightforward and doesn't require a deeper explanation.

## 2. SIMULATING WITH COOJA

Using the *Cooja* interface, we created three motes using the code previously written and opened a server connection on mote #1. The results can be seen in *Figure 1*.
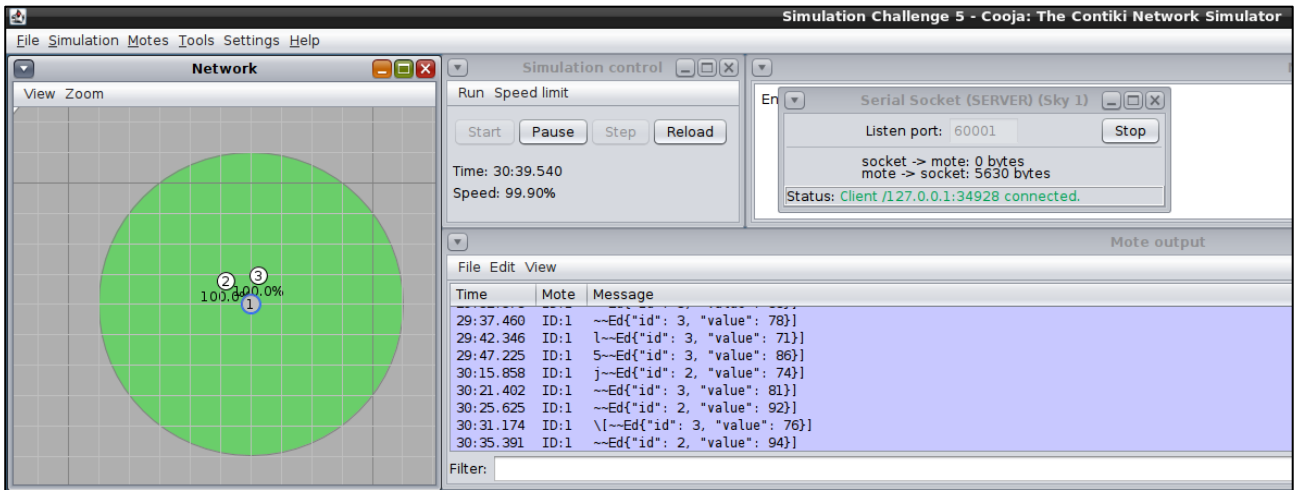


*Figure 1: Cooja simulation*

## 3. SETTING UP NODE-RED

The execution was once again pretty straightforward and can be observed in *Figure 2*. The blocks are built as follows:

- *TCP* node: connected to the server associated to mote #1
- *Function* node #1: removes the bad characters resulting from the compatibility issues between *Cooja* and *Node-Red*
- *Function* node #2: parses the incoming messages and prepares an *MQTT* publish request
- *Delay* node: just created to avoid issues with *Thingspeak*
- *MQTT* node: to communicate with *Thingspeak*

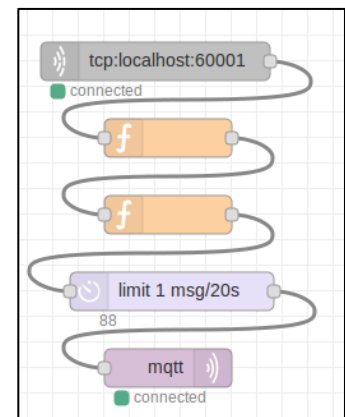No particular issue was raised while doing this part of the challenge.



*Figure 2: Node-Red flow*

## 4. SETTING UP THINGSPEAK CHANNEL

Using the standard procedure, we set up a new channel (1070801) which contains two charts, one for each of the two motes sending messages. Using the link (https://thingspeak.com/channels/1070801) it's possible to observe the values created in the simulation.