

HOME ASSIGNMENT #5

IoT 19-20

Name Surname	Person Code	Id Number
Andrea Crivellin	10491856	928320
Gabriele Guelfi	10491169	916207

Link repo git:

https://github.com/GabrieleGuelfi/IoT_assignments2020.git

Link ThingSpeak channel:

<https://thingspeak.com/channels/1070239>

IMPLEMENTATION

TinyOs

We started from *pipeline.h*, defining the message struct that includes:

- **msg_value**: the random number between 0 and 100 generated at runtime
- **msg_topic**: an integer defining the mote that sent the message, so it can be 2 or 3

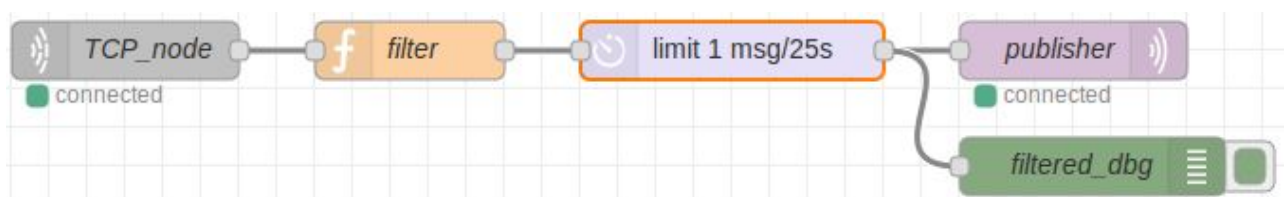
In the *pipelineAppC.nc* file we defined and linked the components we needed:

- AMSenderC, AMReceiverC and ActiveMessageC for communication
- TimerMilliC to start the periodic timer of mote #1
- RandomC to generate random numbers
- SerialPrintfC to print strings through serial interface

Now we'll describe the *pipelineC.nc* and the most relevant functions and events:

- **SplitControl.startDone**: mote#2 and mote#3 start their periodic timer(0.2Hz)
- **MilliTimer.fired**: every time the timer fires, mote#2 and mote#3 send a message to mote#1 with a random generated integer and their *TOS_NODE_ID* as topic
- **Receive.receive**: mote#1 reads the payload of the received packet, then it prints topic and value through serial interface in a well formatted JSON string

Node-RED



- **TCP_node:** it's connected to the serial socket of mote#1 and receives data from it (HOST: localhost, PORT: 60001, delimiter: \n)
- **filter:** it parses the message string as a JSON and return as output only the ones that have a value less or equal than 70. The payload of the output message is a string in the form: "field"+(obj.topic-1)+"="+obj.value+"&status=MQTTPUBLISH". So in ThingSpeak field1 contains data of mote#2 and field2 of mote#3.
- **limit** node: the only function is to limit the message rate to one message every 25 seconds to avoid ThingSpeak restrictions.
- **publisher:** this node's function is to publish the mqtt messages to the thingspeak channel at the topic: channels/1070239/publish/I3DJGO4YYLP9RZEK

SIMULATION

The simulation has to be started in **Cooja**, adding three sky motes and starting a Serial Socket on the port 60001 only on the mote#1. Then the flows continues through **Node-RED** where we left a debug node to check the correctness of data. The final output can be seen in the **ThingSpeak** channel into cool and smooth charts.