Project Report: Challenge 4 - IOT

*Alberto Latino - **10600138**, Lorenzo Mainetti - **10622242***

Goal:
Simulate a TinyOS system made of two nodes communicating with each other. Node 1 performs requests to Node2, where requests are identified by a counter. For each request, Node2 takes a value from a fake sensor and sends it back to Node1 inserting the counter in the response. This way the response can be associated with a request. Nodes require acks to make sure the receiver received the message.

1. **TinyOS App implementation**

   a. **SendAckC.nc**
      In this module we define all the events that define the behavior of the nodes when the timer starts, a message is received/sent. Each method checks what is the node ID in order to perform different behaviors.

   b. **SendAckAppC.nc**
      Our application implements the following TinyOS components
      components MainC, sendAckC as App;
      components new TimerMilliC() as MilliTimer;
      components new AMSenderC(AM_MY_MSG);
      components new AMReceiverC(AM_MY_MSG);
      components new FakeSensorC();
      components ActiveMessageC;
      In this file we declared them and then we wired them accordingly.

   c. **Message Structure**
      The message is represented by a struct containing three fields.
      i. msg_type: indicating whether the message is a request or a response
      ii. counter: indicating the id of the request/response
      iii. value: contains the value generated by the fake sensor in Node2. Request messages do not use this field, therefore its default value is zero.

   d. **Flow (X = 9, Y = 00)**
      i. Every 1000 ms Node1 starts the process for a specific counter value, up to the value of X = 9. Node 2 boots at time Y = 0 (we set it in the RunSimulationScript.py).
      ii. Node 1 sends a request to Node2. At the same time Node1 is waiting for an ack from Node 2.
      iii. Node2 processes the request and takes the value generated by the fake sensor. Generate the response inserting the value and the same counter value and send it to Node1, waiting for an ack.
      iv. Receives the response for Node2 and send ack to node 2
      v. Node2 receives the ack.

2. **TOSSIM simulation**
   We compiled our TinyOS implementation adding the '*sim*' keyword and we simulated the interaction between the two nodes using the RunSimulationScript.py.