Raffaele Cicellini 10626081

Francesca Forbicini 10628756

**Challenge 04 – Internet of Things**

First, in the “sendAckC.nc” we wrote all the interfaces we needed to implement the sensors and in “sendAckAppC.nc” we wrote all the needed components and we wired them together. We also specified in “sendAck.h” the structure of the messages sent by the motes with the needed fields (msg\_type=0 for response, 1 for request, msg\_counter, value) and in “RunSimulationScript.py” we modified the time when mote 2 boots up fixing it at time y=26s. Then, we implemented the logic of the application modifying the events that were already present in “sendAckC.nc”.

* sendReq(): function called by mote 1 every time it wants to send a request message (so every time the timer is fired). We prepare the packet setting msg\_type=1 (=request) and msg\_counter to the current counter value. Then we request an Ack for this message and we send it using AMSend and locking the transmission
* sendResp(): function used by mote 2 when it sends a response to mote 1. It simply reads a random value from a fake sensor.
* Event Boot.booted(): event fired when application starts. It simply starts the process.
* Event AMControl.startDone(): called when startup phase is completed. If we are in mote 1 we start the periodic timer, otherwise we do nothing. If the startup phase was not completed correctly we retry.
* Event AMControl.stopDone(): not used.
* Event MilliTimer.fired(): called when timer is fired. It increments the counter and call sendReq().
* Event AMSend.sendDone(): called when a packet is sent. If it was correctly sent we unlock the transmission. If the packet was acked and we are on mote 1 we increment the counter of received acks and if it is equal to X (2 in our case) we call MilliTimer.stop to stop execution of mote 1.
* Event Receive.receive(): called when we receive a packet. If it is a request message we send a response with sendResp(), otherwise we simply print the content of the message received.
* Event Read.readDone(): called when we read a value from the fake sensor on mote 2. If everything went fine we prepare the response message, ask for Ack and then we send it locking up the transmission, otherwise we call again sendResp().