HOME ASSIGNMENT #1

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

IMPLEMENTATION

We used the class example RadioCountToLeds as skeleton for our implementation. We started from RadioLeds.h adding the sender_id field in the message struct. Then in the RadioLedsC.nc file we defined the behaviour of the motes, below the most relevant events are described:

- **AMControl.startDone**: each mote checks its 1own TOS_NODE_ID and sets its own inner timer to the correct frequency (id = 1 -> 1 Hz; id = 2 -> 3 Hz; id = 3 -> 5 Hz)
- **MilliTimer.fired**: when timer fires each mote increase its own counter and sends a broadcast message containing the counter and his TOS_NODE_ID and sets its status as locked until the event AMSend.sendDone triggers
- **Receive.receive:** when a packet is received by a mote, it checks if the counter field is a multiple of 10 and if it's true it turns off all *its* leds. Otherwise the mote checks the sender_id field and toggles the corresponding led (id 1 -> led 0; id 2 -> led 1; id 3 -> led 2)

In the RadioLedsAppC.nc file we described which components to use and wired them through interfaces.

SIMULATION

We created a new Cooja simulation with three sky motes. We observed that when the simulation starts the motes are not booted at the same moment. This leads to a delay that causes motes' timers to not be synchronized. We also used Cooja's simulation tool to debug our implementation.