#03: TinyOS

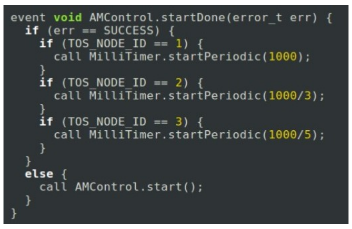
Andrea Calici, C. Persona 10490117, Matricola 944717

Starting from the *RadioCountToLeds* project seen in class, I implemented the *RadioLeds* project for the challenge. In particular:

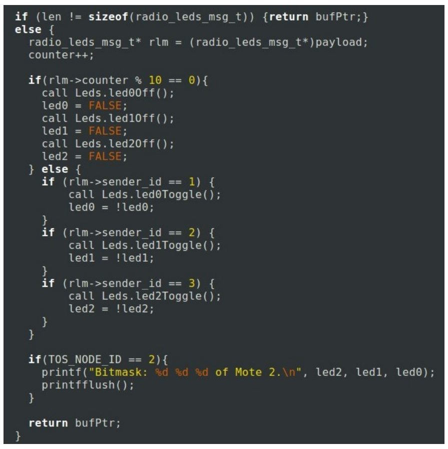
- `RadioLeds.h`: I added the `nx_uint8_t sender_id` variable in order to store the Cooja ID of the sender;

- `RadioLedsC.nc`: I set the frequency to send all the messages (Mote 1: 1 Hz, Mote 2: 3 Hz, Mote 3: 5 Hz), then I defined when the Leds have to be toggled or turned off:
  - messages sent by mote 1 toggle led0;
  - messages sent by mote 2 toggle led1;
  - messages sent by mote 3 toggle led2;
  - messages received with `counter mod 10 == 0` turn off all the LEDs;

  The counter is incremented when a message is received. Finally, if node ID is equal to 2 I printed the status of all the Leds in order to compile the form for the challenge, using 3 boolean values that follow the behaviour of the Leds.



```
event void AMControl.startDone(error_t err) {
  if (err == SUCCESS) {
    if (TOS_NODE_ID == 1) {
      call MilliTimer.startPeriodic(1000);
    }
    if (TOS_NODE_ID == 2) {
      call MilliTimer.startPeriodic(1000/3);
    }
    if (TOS_NODE_ID == 3) {
      call MilliTimer.startPeriodic(1000/5);
    }
  }
  else {
    call AMControl.start();
  }
}
```



```
if (len != sizeof(radio_leds_msg_t)) {return bufPtr;}
else {
  radio_leds_msg_t* rlm = (radio_leds_msg_t*)payload;
  counter++;

  if(rlm->counter % 10 == 0){
    call Leds.led0Off();
    led0 = FALSE;
    call Leds.led1Off();
    led1 = FALSE;
    call Leds.led2Off();
    led2 = FALSE;
  } else {
    if (rlm->sender_id == 1) {
        call Leds.led0Toggle();
        led0 = !led0;
    }
    if (rlm->sender_id == 2) {
        call Leds.led1Toggle();
        led1 = !led1;
    }
    if (rlm->sender_id == 3) {
        call Leds.led2Toggle();
        led2 = !led2;
    }
  }

  if(TOS_NODE_ID == 2){
    printf("Bitmask: %d %d %d of Mote 2.\n", led2, led1, led0);
    printfflush();
  }

  return bufPtr;
}
```

- `RadioLedsAppC.nc`: I added the `printf` library and wired components and interfaces.