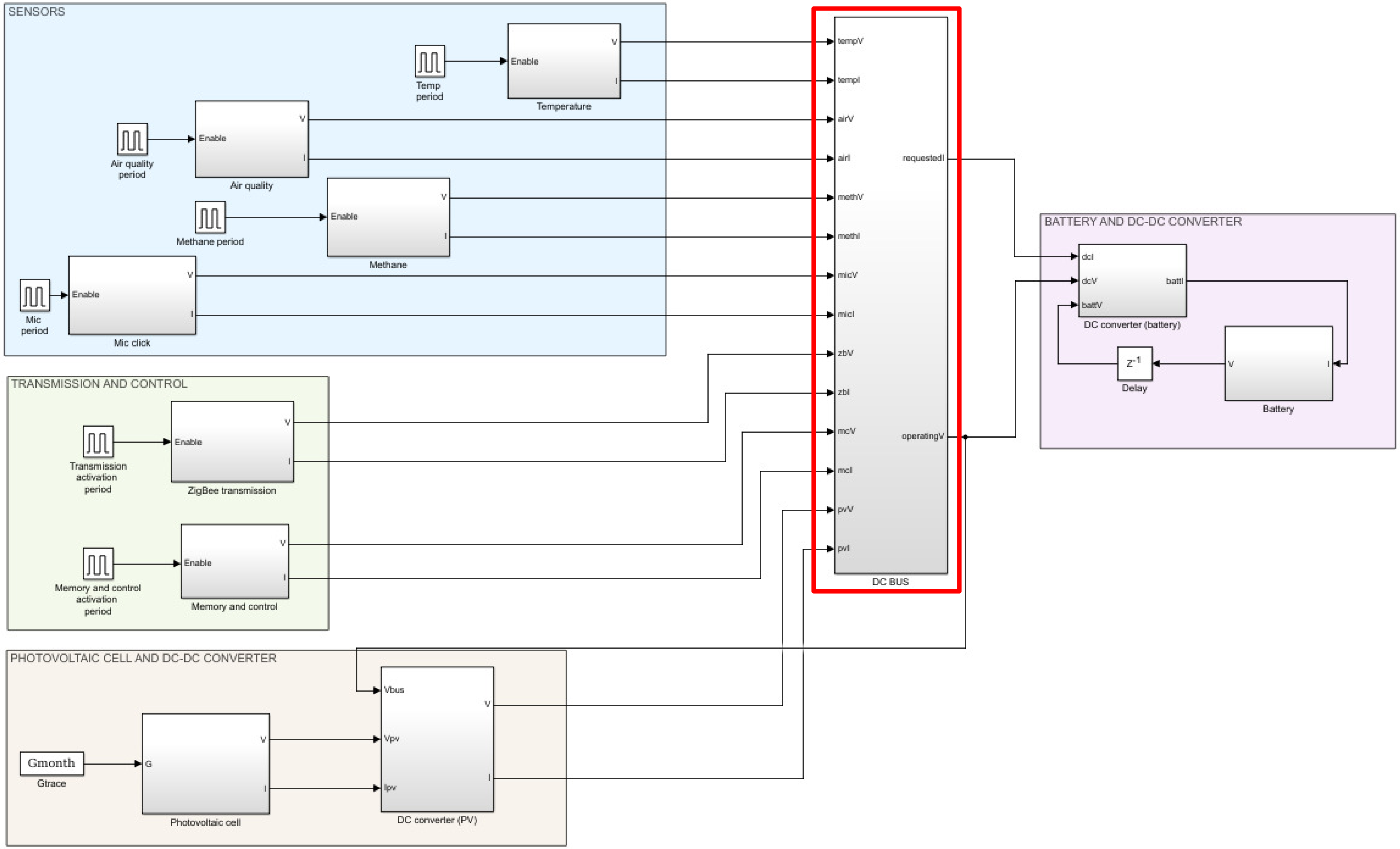
Lab 3 – Day 3
Load modeling and scheduling

Simulink implementation



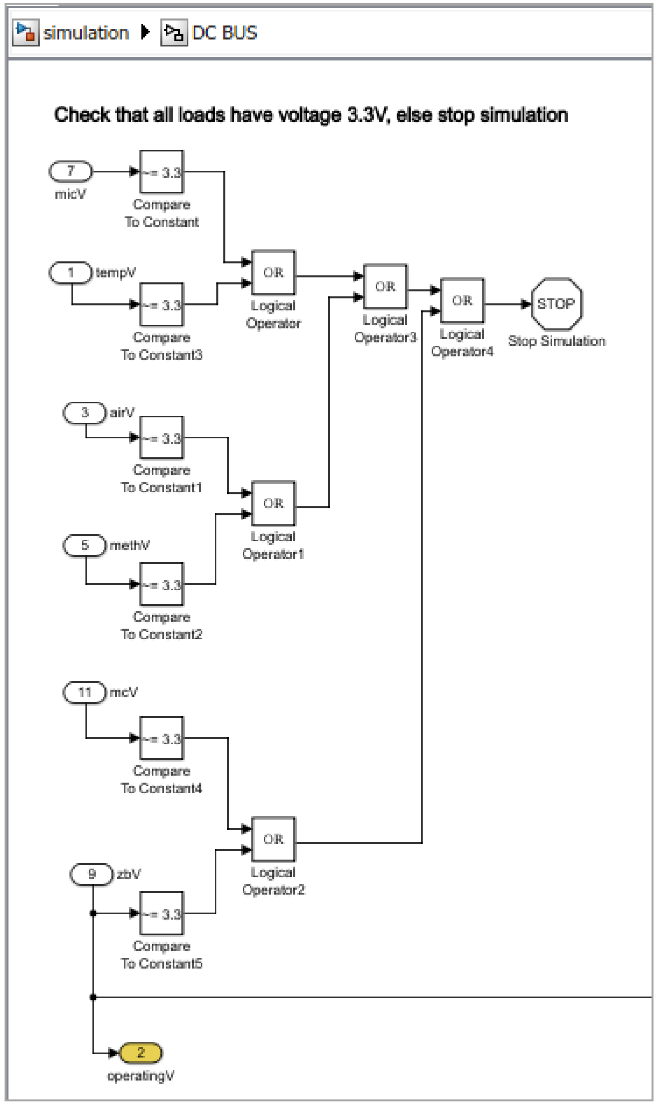
DC bus

- Two-fold goal:
  - Provide a **reference voltage** for the system (3.3V)
  - **Rule the energy flow** in the system
    - Determine what is the total power consumption of loads
    - Collect the power production of the photovoltaic module
    - Decide how to use the battery:
      - If the photovoltaic power is higher or equal to total load demand, no need to use the battery
        - » I can even charge the battery!
      - Else, estimate how much current must be drained from the battery

- **How long will my IoT system survive?**

DC bus

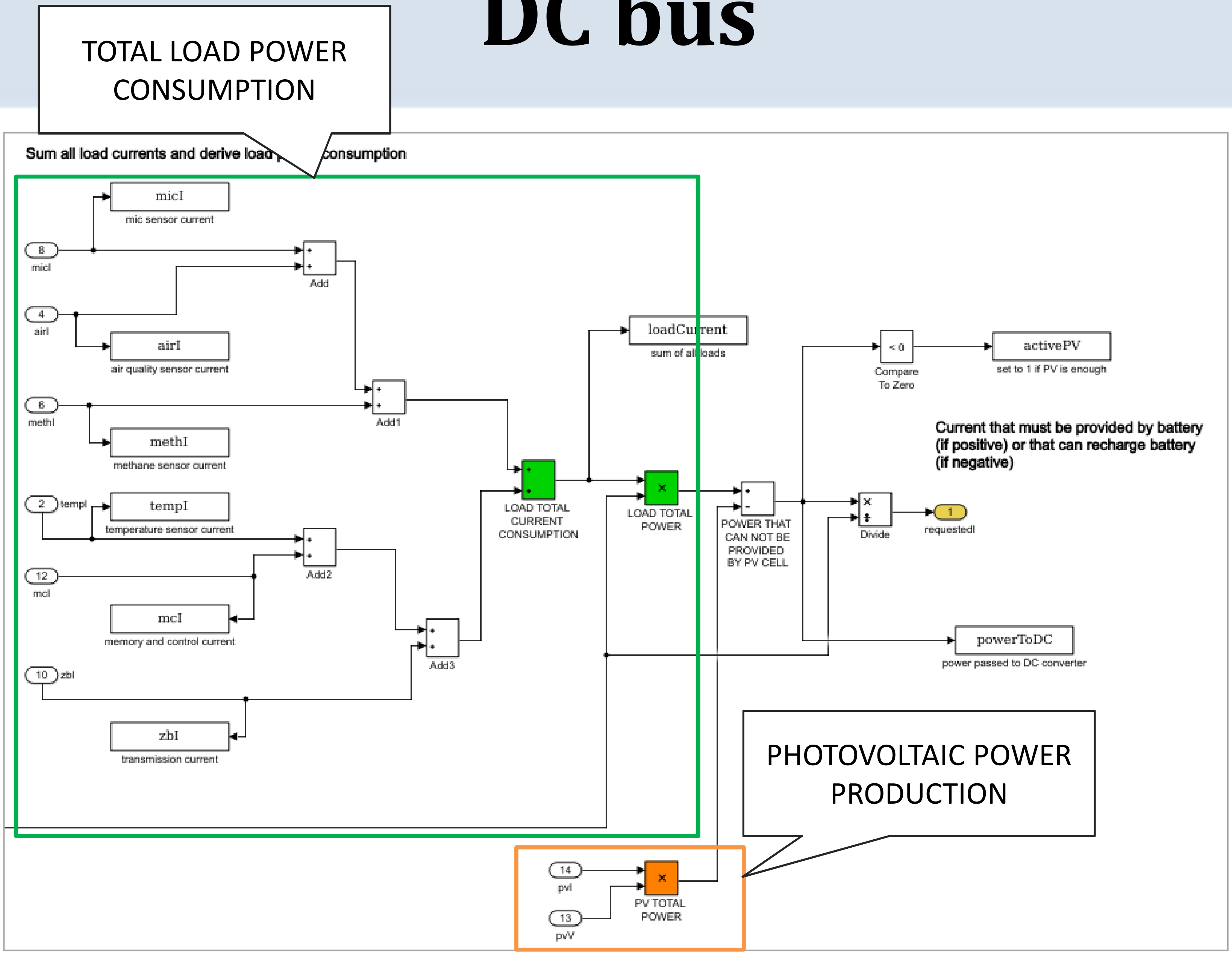
- **Part 1:**
  - Guarantee that all current requests happen at the same voltage
  - If any voltage is different from 3.3V, stop the simulation!
    - Maybe someDC-DC converter is missing?



DC bus

- **Part 2:**
  - Estimate the energy flow
    1. Estimate total load power consumption
    2. Derive photovoltaic power production
    3. Calculate the difference: $P_{LOAD} - P_{PV}$
       - If the difference is positive, that power must be provided by the battery (photovoltaic module is not enough!)
       - If the difference is negative, that power can be used to charge the battery (unused power!)
    4. Battery current is thus: $\frac{P_{LOAD} - P_{PV}}{3.3V}$

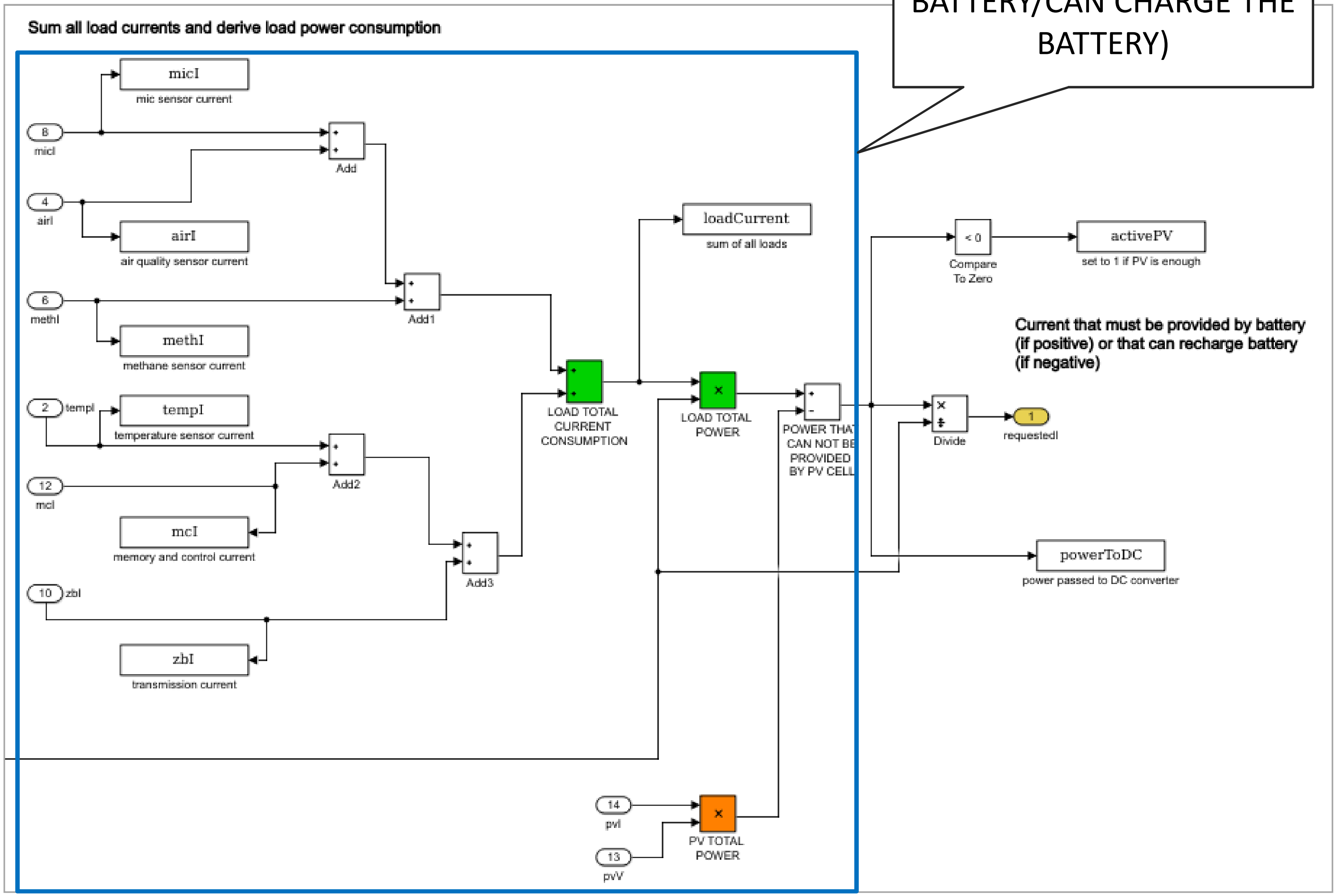
DC bus



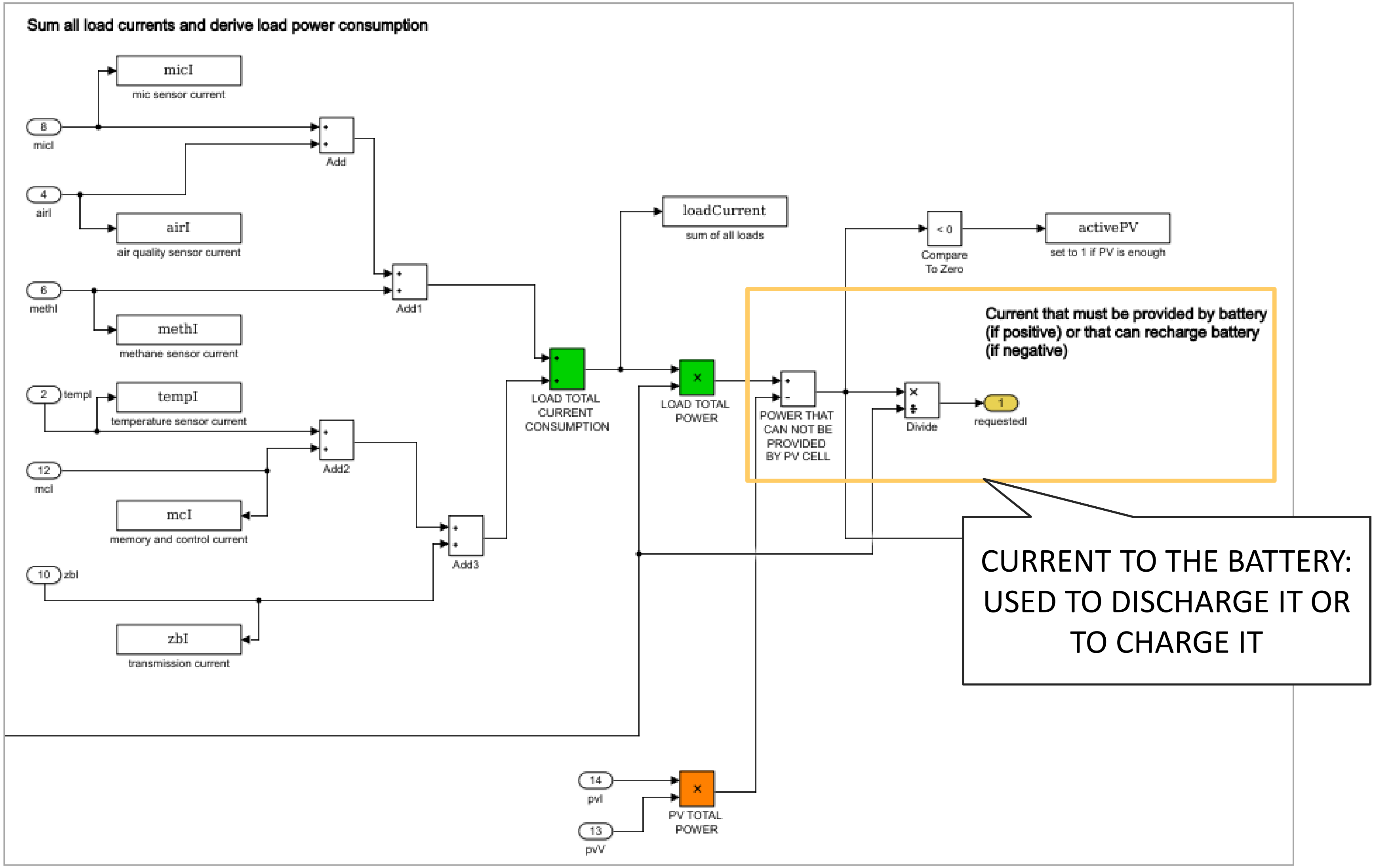
TOTAL LOAD POWER CONSUMPTION
Sum all load currents and derive load power consumption
micI
mic sensor current
8
micI
Add
4
airI
airI
air quality sensor current
6
methI
methI
methane sensor current
Add1
2
tempI
tempI
temperature sensor current
Add2
12
mcI
mcI
memory and control current
Add3
10
zbI
zbI
transmission current
loadCurrent
sum of all loads
LOAD TOTAL CURRENT CONSUMPTION
LOAD TOTAL POWER
POWER THAT CAN NOT BE PROVIDED BY PV CELL
< 0
Compare To Zero
activePV
set to 1 if PV is enough
Current that must be provided by battery (if positive) or that can recharge battery (if negative)
Divide
1
requestedI
powerToDC
power passed to DC converter
PHOTOVOLTAIC POWER PRODUCTION
14
pvI
13
pvV
PV TOTAL POWER

DC bus

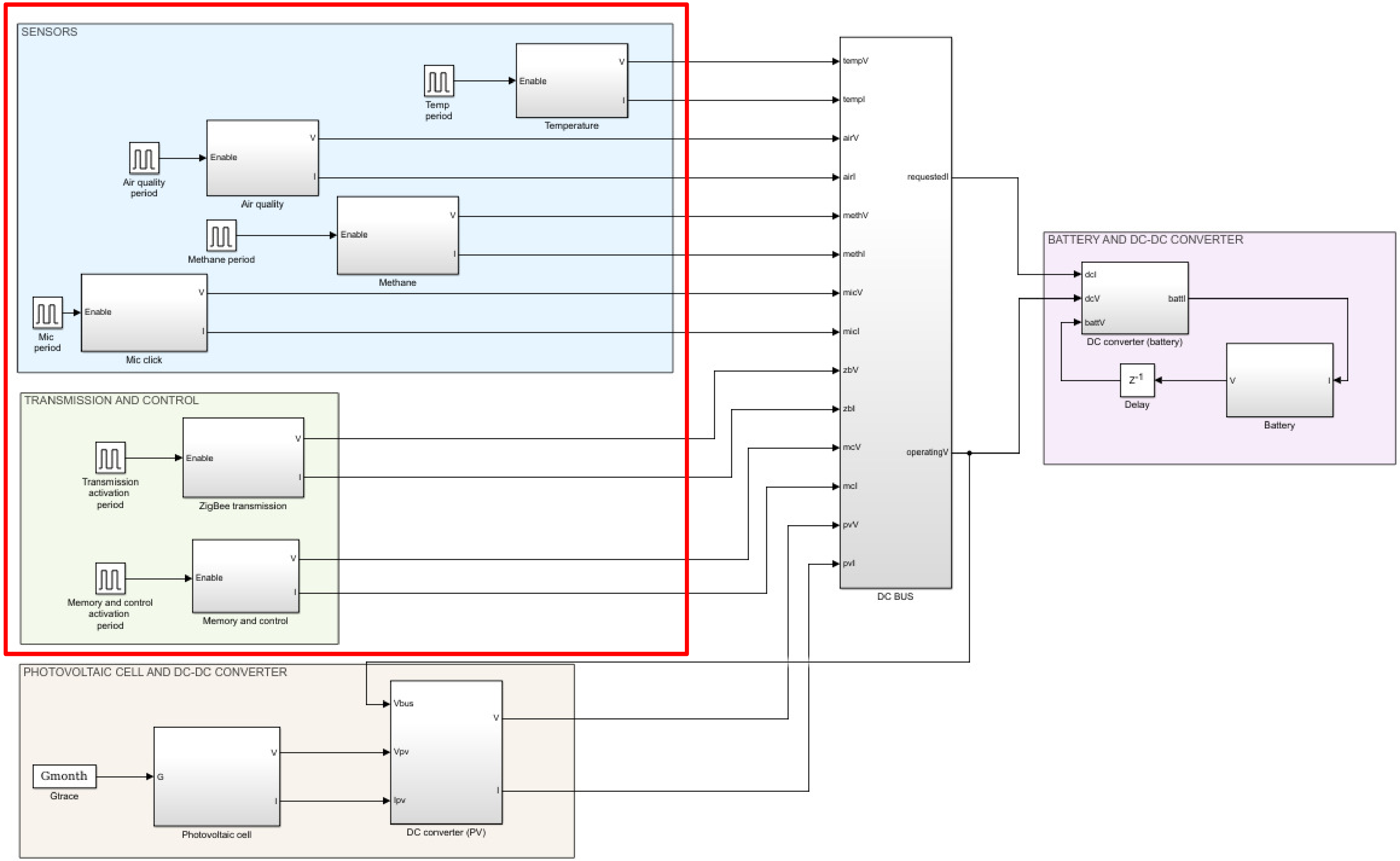
POWER MISSING/STILL AVAILABLE
(MUST DISCHARGE THE BATTERY/CAN CHARGE THE BATTERY)



DC bus


Sum all load currents and derive load power consumption
micI
mic sensor current
8
micI
Add
4
airI
airI
air quality sensor current
6
methI
Add1
methI
methane sensor current
2
tempI
tempI
temperature sensor current
Add2
12
mcI
mcI
memory and control current
Add3
10
zbI
zbI
transmission current
LOAD TOTAL CURRENT CONSUMPTION
loadCurrent
sum of all loads
LOAD TOTAL POWER
POWER THAT CAN NOT BE PROVIDED BY PV CELL
< 0
Compare To Zero
activePV
set to 1 if PV is enough
Current that must be provided by battery (if positive) or that can recharge battery (if negative)
Divide
1
requestedI
14
pvI
13
pvV
PV TOTAL POWER
CURRENT TO THE BATTERY: USED TO DISCHARGE IT OR TO CHARGE IT

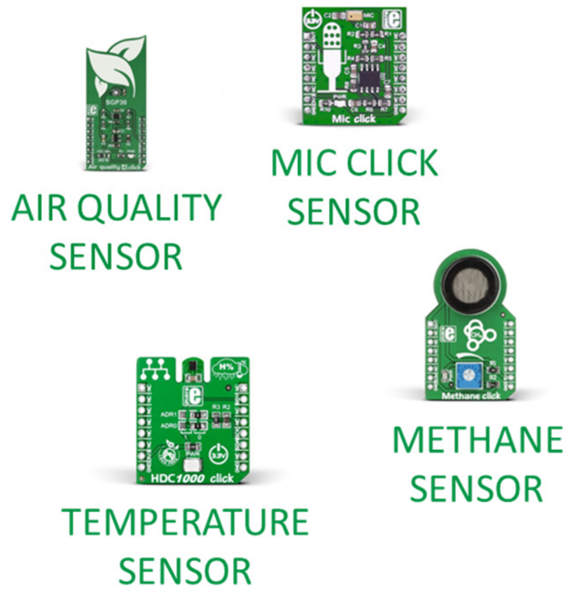
Simulink implementation



Loads

- 6 loads
  - Temperature sensor
    - Humidity and temperature sensor
  - Methane sensor
    - CH_4 sensor
  - Air quality sensor
    - Metal-oxide, H_2 and Ethanol sensing
  - Mic click sensor
    - Silicon microphone
  - Memory and control unit
  - A module to transmit data over ZigBee

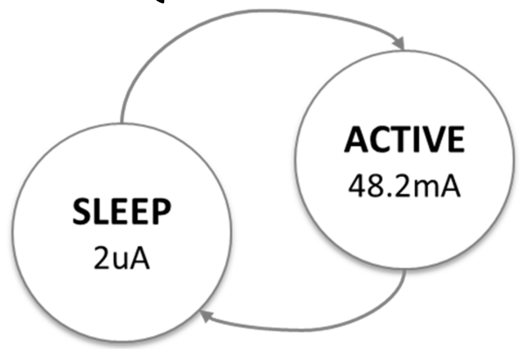
MIKROE
be on time



Loads

- Loads are implemented as simple PSM
  - Fixed voltage, same as DC bus (3.3V)
  - Varying current
    - One active state (higher consumption)
    - One sleep state (lower consumption)
- Data taken from load datasheets
  - Typical current consumption depending on activity

AIR QUALITY SENSOR



LOAD	ACTIVE (mA)	SLEEP (mA)
Air quality sensor	48.2	0.002
Methane sensor	18	0.002
Temperature sensor	3	0.002
Mic click sensor	0.15	0.002
ZigBee transmission	0.1	0.001
Memory and control	13	0.002

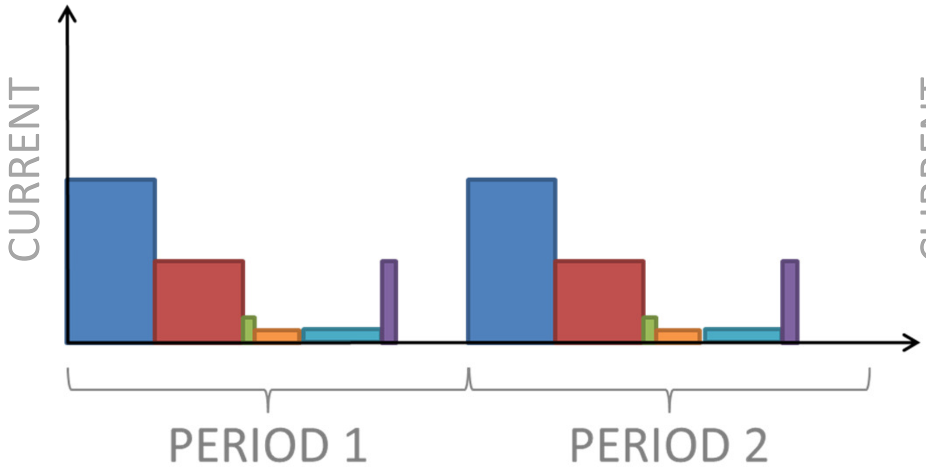
Loads

- Loads behavior is periodic
  - Active time is fixed for each load
    - Depends on the requirements of the component
    - E.g., how long it takes for the sensor to make the measurements
    - Given as input specification

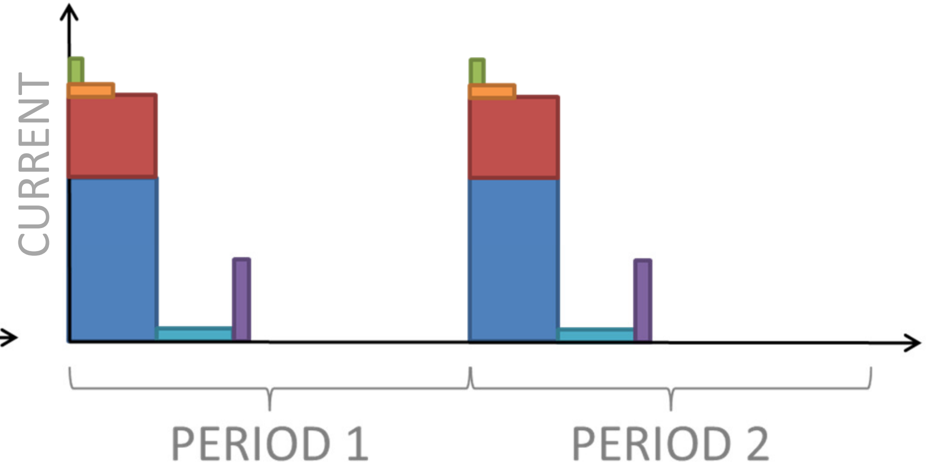
LOAD	TIME (s)
Air quality sensor	30
Methane sensor	30
Temperature sensor	6
Mic click sensor	12
ZigBee transmission	24
Memory and control	6

Loads

- Loads behavior is periodic
  - Load activation depends on a schedule
    - Fixed order: first sensors, then memory and control, then transmission
    - Sensors can be activated sequentially or in parallel



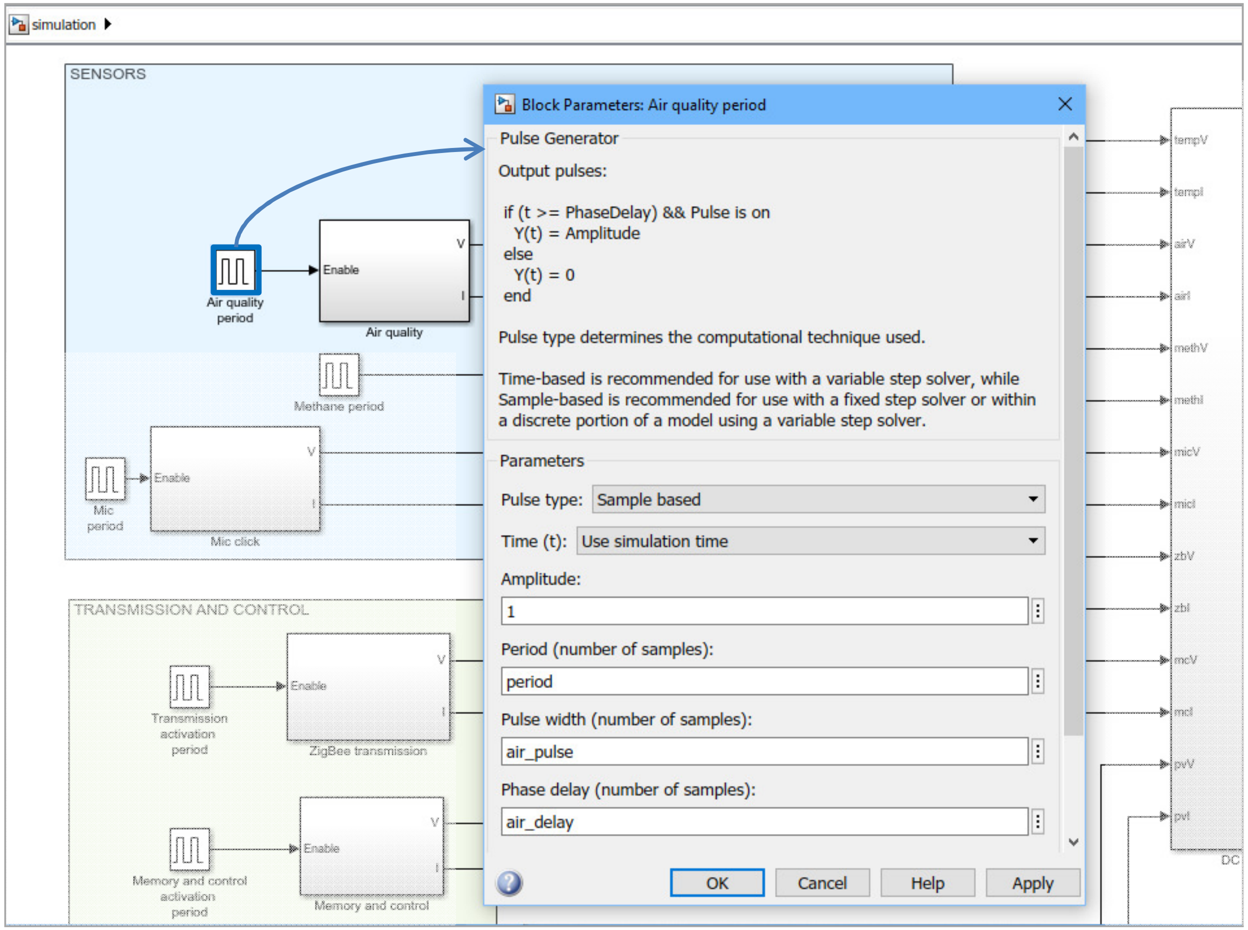
SEQUENTIAL ACTIVATION OF SENSORS



PARALLEL ACTIVATION OF SENSORS

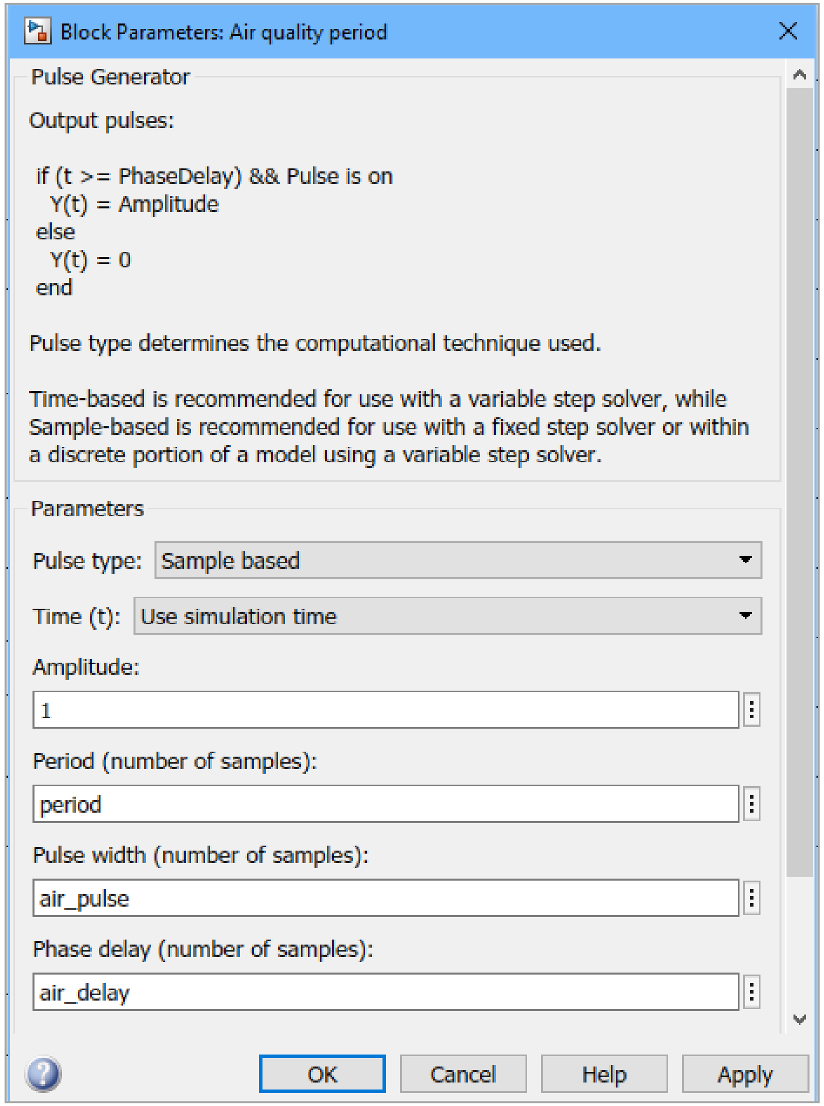
Loads

- Who controls load activation?



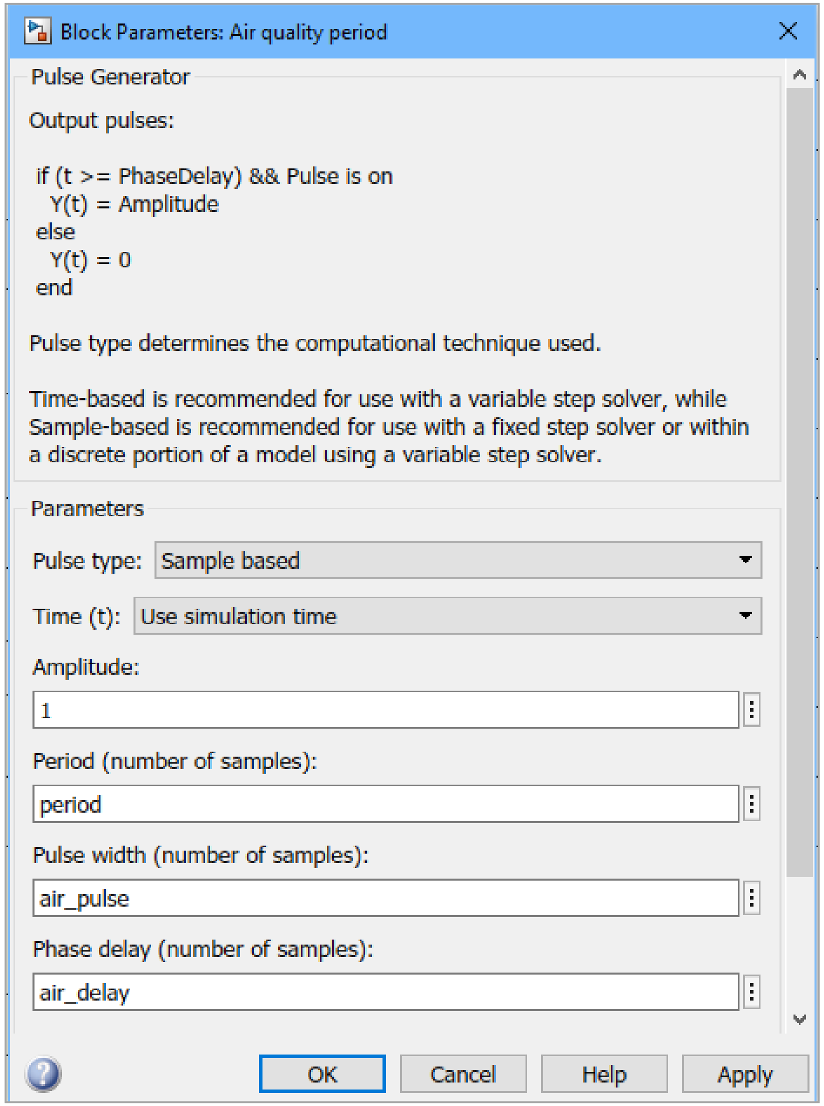
Loads

- **Pulse generator:**
  - Define the *period*
    - Equal for all loads
  - Define the *delay*
    - 0 if load starts at the beginning of each period
    - Else equal to the computation time of former loads
  - Define *pulse width*
    - % of period in which the load is active
    - $\frac{\text{ACTIVE TIME} \cdot 100}{\text{PERIOD}}$



Loads

- **Pulse generator:**
  - Loads variables from the workspace
    - E.g., for the air quality sensor:
      - Period = `period` variable
      - Delay = `air_delay` variable
      - Pulse width = `air_pulse` variable
  - Starting values provided in the **config.m** file
    - <u>Must be executed before you run the simulation!</u>



config.m

LOAD IRRADIANCE TRACE

```
load('gmonth.mat')
```

LENGTH OF ACTIVE TIME FOR EACH LOAD

```
% sensor activation durantion
air_time = 30;
methane_time = 30;
temp_time = 6;
mic_time = 12;
transmit_time = 24;
mc_time = 6;
```

ACTIVATION DELAY FOR EACH LOAD

(0 MEANS THAT LOAD IS ACTIVATED AT THE BEGINNING OF EACH PERIOD) HERE SENSORS ARE ACTIVATED IN PARALLEL

```
% activation delay
% parallel exec
air_delay = 0;
methane_delay = 0;
temp_delay = 0;
mic_delay = 0;
mc_delay = 30;
transmit_delay = mc_delay + mc_time;
```

ACTIVATION PERIOD

(HERE 120s)

```
% period
% setting 1: 120 (2 minutes)
% setting 2: 60*10 (10 minutes)
period = 120;
```

PULSE WIDTH FOR EACH LOAD

$$\left(\frac{\text{ACTIVE TIME} \cdot 100}{\text{PERIOD}}\right)$$

```
% pulse width: % of period when sensor is active
air_pulse = (air_time * 100)/period;
methane_pulse = (methane_time * 100)/period;
temp_pulse = (temp_time *100)/period;
mic_pulse = (mic_time * 100)/period;
mc_pulse = (mc_time*100)/period;
transmit_pulse = (transmit_time * 100)/period;
```

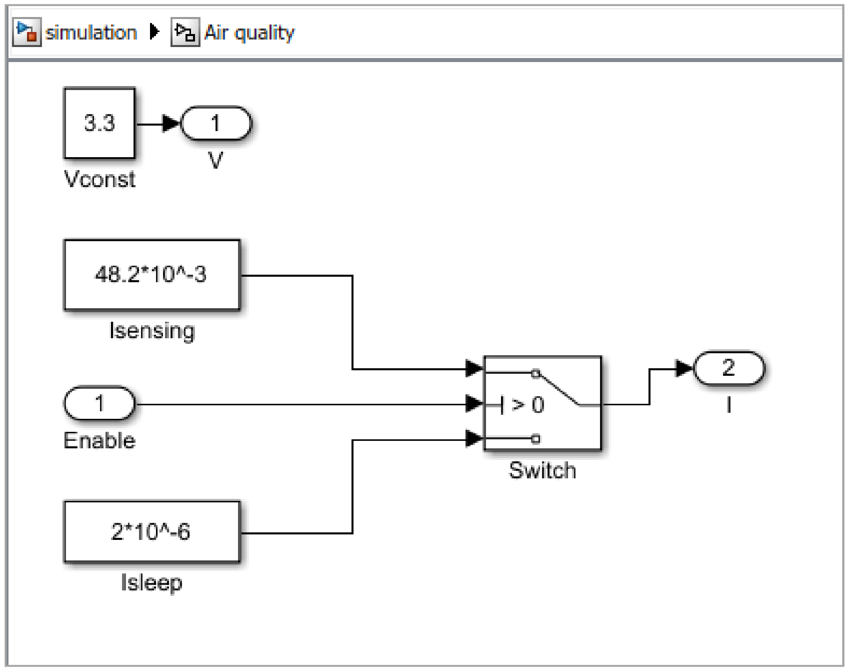
SIMULATION LENGTH

(HERE 30 days)

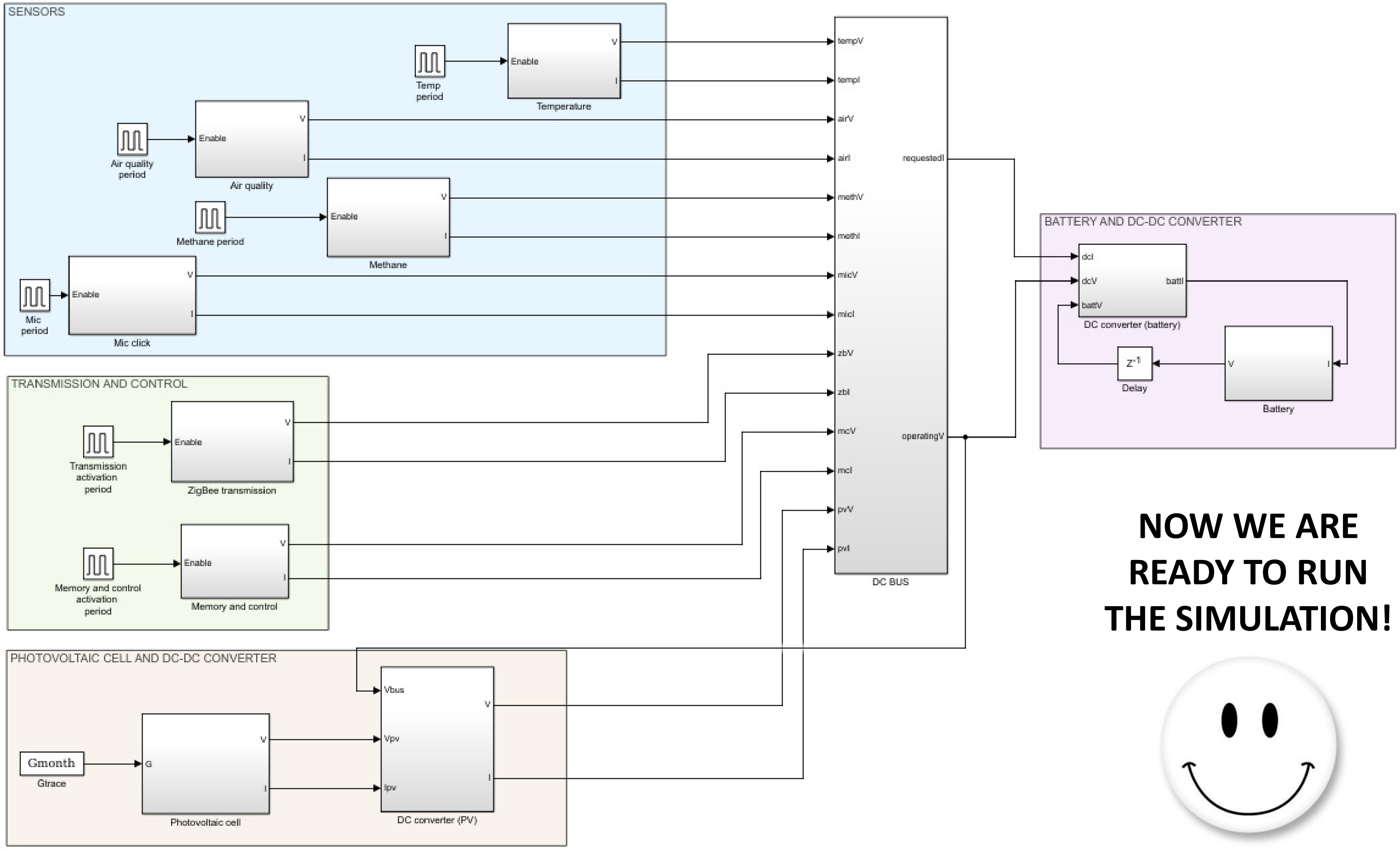
```
% simulation length
% max simulation length = 1 month (30 days)
length = 30*24*60*60;
```

Loads

- Loads react to the pulse:
  - Pulse is 1 → higer current value
    - Active load
  - Pulse is 0 → lower current value
    - Non active load
  - Voltage is fixed
    - Always 3.3V



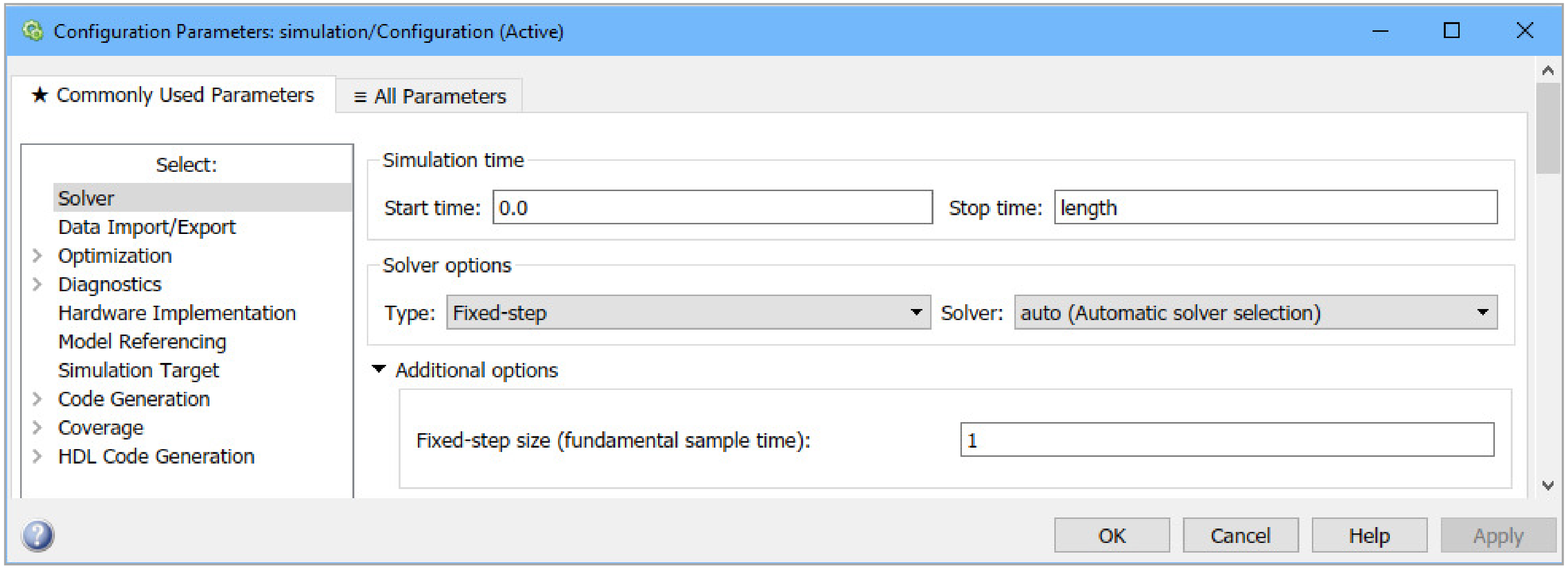
Simulink implementation



NOW WE ARE READY TO RUN THE SIMULATION!

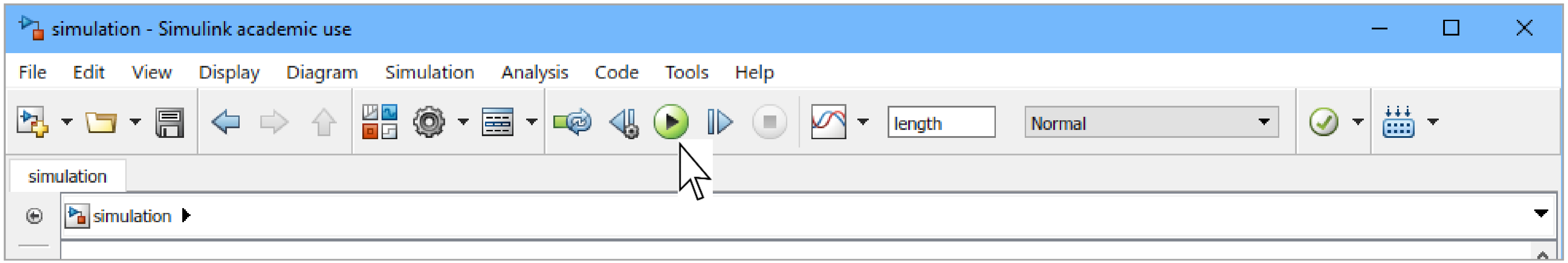
Simulation settings

- **Already set – no need to change them!**
- Simulation length: length variable
  - Set to 30 days
- Timestep: 1 second



Simulation settings

- To run the simulation:
  - Click on the green play button
  - Simulation stops when:
    - You reached the defined simulation length (30 days)
    - Any stop has been activated
      - E.g., battery discharged



First analysis of the Lab:

- In the report, no need to comment on the construction of the components

- **Discuss the main features of the simulation** given the provided schedule (parallel sensors)
  1. Exhibit one example trace of the simulation, showing meaningful qualtities
     - E.g., loads power, photovoltaic power, battery power, and battery SOC to show that the system behaves correctly
  2. Efficiency of the converters, to see if we fall in a good area (high efficiency) or not (low efficiency = a lot of wasted power)
  3. How often the battery has to be used to supply power
     - I.e., photovoltaic power is not enough

Second analysis of the Lab:

- **Determine the best scheduling**
  - Allows **longer lifetime** of the system

  - Compare:
    - Different periods, from 2 minutes to 10 minutes
    - Different scheduling
      - All sensors in parallel
      - Sequential activation of the sensor
      - Any impact on the evolution battery?
    - Choose (motivating) the best schedule

Second analysis of the Lab:

- What the analysis should contain:
  - What different parameters have been explored
    - E.g., different period lengths
  - What schedule orders have been explored
  - What lifetime has been achieved with each configuration
    - Document simulation with useful plots, e.g., of SOC
  - Which configuration is considered better and why
    - With value of maximum achieved lifetime