

sensor:

- platform: template

sensors:

Template sensor for values of power import (active_power > 0)

power_import:

friendly_name: "Power Import"

unit_of_measurement: 'W'

value_template: >-

```
{% if (states('sensor.keller_3em_channel_a_power')|float +
states('sensor.keller_3em_channel_b_power')|float +
states('sensor.keller_3em_channel_c_power')|float) > 0 %}
```

```
  {{ states('sensor.keller_3em_channel_a_power')|float +
states('sensor.keller_3em_channel_b_power')|float +
states('sensor.keller_3em_channel_c_power')|float }}
```

```
{% else %}
```

```
  {{ 0 }}
```

```
{% endif %}
```

availability_template: "{{

```
  [ states('sensor.keller_3em_channel_a_power'),
```

```
    states('sensor.keller_3em_channel_b_power'),
```

```
    states('sensor.keller_3em_channel_c_power')
```

```
  ] | map('is_number') | min
```

```
  }}"
```

Template sensor for values of power export (active_power < 0)

power_export:

friendly_name: "Power Export"

unit_of_measurement: 'W'

value_template: >-

```
{% if (states('sensor.keller_3em_channel_a_power')|float +
states('sensor.keller_3em_channel_b_power')|float +
states('sensor.keller_3em_channel_c_power')|float) < 0 %}
```

```
  {{ (states('sensor.keller_3em_channel_a_power')|float +
states('sensor.keller_3em_channel_b_power')|float +
states('sensor.keller_3em_channel_c_power')|float) * -1 }}
```

```
{% else %}
```

```
  {{ 0 }}
```

```
{% endif %}
```

```
availability_template: "{{
```

```
  [ states('sensor.keller_3em_channel_a_power'),
```

```
    states('sensor.keller_3em_channel_b_power'),
```

```
    states('sensor.keller_3em_channel_c_power')
```

```
  ] | map('is_number') | min
```

```
}}"
```

```
# Template sensor for values of power consumption
```

```
power_consumption:
```

```
  friendly_name: "Power Consumption"
```

```
  unit_of_measurement: 'W'
```

```
  value_template: >-
```

```
    {% if (states('sensor.power_export')|float(0)) > 0 and (states('sensor.power_solargen')|float(0) -  
states('sensor.power_export')|float(0)) < 0 %}
```

```
      {{ 0 }}
```

```
    {% elif (states('sensor.power_export')|float(0)) > 0 and (states('sensor.power_solargen')|float(0)  
- states('sensor.power_export')|float(0)) > 0 %}
```

```
      {{ (states('sensor.power_solargen')|float(0)) - states('sensor.power_export')|float(0) }}
```

```
    {% else %}
```

```
      {{ states('sensor.power_import')|float(0) + states('sensor.power_solargen')|float(0) }}
```

```
    {% endif %}
```

```
# Template sensor for values of energy Solar generation (solar_power > 0)
```

```
# YOU NEED TO UPDATE THE NAMES BELOW!!!
```

```
power_solargen:
```

```
  friendly_name: "Power Solar Generation"
```

```
  unit_of_measurement: 'W'
```

```
  value_template: >-
```

```
    {% if (states('sensor.bkw.power')|float(0) + states('sensor.bkw_power')|float(0)) > 0 %}
```

```
    {{ (states('sensor.bkw.power')|float(0) + states('sensor.bkw_power')|float(0)) }}  
    {% else %}  
    {{ 0 }}  
    {% endif %}
```

Sensor for Riemann sum of energy import (W -> Wh)

```
- platform: integration  
source: sensor.power_import  
name: energy_import_sum  
unit_prefix: k  
round: 2  
method: left
```

Sensor for Riemann sum of energy export (W -> Wh)

```
- platform: integration  
source: sensor.power_export  
name: energy_export_sum  
unit_prefix: k  
round: 2  
method: left
```

Sensor for Riemann sum of energy consumption (W -> Wh)

```
- platform: integration  
source: sensor.power_consumption  
name: energy_consumption_sum  
unit_prefix: k  
round: 2  
method: left
```

utility_meter:

```
energy_import_daily:  
source: sensor.energy_import_sum
```

name: Energy Import Daily

cycle: daily

energy_import_monthly:

source: sensor.energy_import_sum

name: Energy Import Monthly

cycle: monthly

energy_export_daily:

source: sensor.energy_export_sum

name: Energy Export Daily

cycle: daily

energy_export_monthly:

source: sensor.energy_export_sum

name: Energy Export Monthly

cycle: monthly

energy_consumption_daily:

source: sensor.energy_consumption_sum

name: Energy Consumption Daily

cycle: daily

energy_consumption_monthly:

source: sensor.energy_consumption_sum

name: Energy Consumption Monthly

cycle: monthly