# Home Assistant

- Fronius PV einbinden
- Xiaomi Smart Devices
- Clients einbinden
- Touch PC
- KWL Pluggit
- Beschattungsautomatik

# Fronius PV einbinden

```
# in die Datei /homeassistant/configuration.yaml einbauen
...
template:
  - sensor:
      - name: "Battery Power Charging"
        unit_of_measurement: W
        device_class: power
        state: "{{ max(0, 0 -  states('sensor.solarnet_power_battery') | float(default=0)) }}"
      - name: "Battery Power Discharging"
        unit_of_measurement: W
        device_class: power
        state: "{{ max(0, states('sensor.solarnet_power_battery') | float(default=0)) }}"
      - name: "Power Photovoltaics"
        unit_of_measurement: W
        device_class: power
        state: "{{ states('sensor.solarnet_power_photovoltaics') | float(default=0) }}"
      - name: "Netzbezug"
        unit_of_measurement: W
        device_class: power
        state: "{{ states('sensor.solarnet_leistung_netzbezug') | float(default=0) }}"
      - name: "Netzeinspeisung"
        unit_of_measurement: W
        device_class: power
        state: "{{ states('sensor.solarnet_leistung_netzeinspeisung') | float(default=0) }}"

sensor:
  - platform: integration
    source: sensor.battery_power_charging
    name: "Total Battery Energy Charged"
    unique_id: "myuuid_1234"
    unit_prefix: k
    method: left
  - platform: integration
    source: sensor.battery_power_discharging
    name: "Total Battery Energy Discharged"
```

```
    unique_id: "myuuid_1235"
    unit_prefix: k
    method: left
  - platform: integration
    source: sensor.power_photovoltaics
    name: "Total Photovoltaics Energy"
    unique_id: "myuuid_1236"
    unit_prefix: k
    method: left
  - platform: integration
    source: sensor.netzbezug
    name: "Totaler Netzbezug"
    unique_id: "myuuid_1237"
    unit_prefix: k
    method: left
  - platform: integration
    source: sensor.netzeinspeisung
    name: "Totale Netzeinspeisung"
    unique_id: "myuuid_1238"
    unit_prefix: k
    method: left
```

Quelle: https://kofler.info/fronius-wechselrichter-in-home-assistant-einbinden/

# Xiaomi Smart Devices

```
---------
  NAME:     Bad Eg
  ID:       77726756
  MAC:      78:11:DC:AB:C3:DB
  IP:       192.168.0.142
  TOKEN:    68471cab5e8f5d92f9cb64ce4694387f
  MODEL:    yeelink.light.ceiling1
  ---------
  NAME:     Bad OG (Farbe)
  ID:       131153750
  MAC:      04:CF:8C:84:AE:84
  IP:       192.168.0.135
  TOKEN:    095ccb1783aa83cb87d63381650d8bc1
  MODEL:    yeelink.light.color2
  ---------
  NAME:     Viola
  ID:       131399194
  MAC:      04:CF:8C:88:6D:48
  IP:       192.168.0.139
  TOKEN:    9f190efb06b2ad0f8fc395f570f7c937
  MODEL:    yeelink.light.ct2
  ---------
  NAME:     Schlafzimmer (weiß)
  ID:       131408275
  MAC:      04:CF:8C:88:90:C1
  IP:       192.168.0.140
  TOKEN:    fe8557b3f5dbd2fe8ea953278bdb03f3
  MODEL:    yeelink.light.ct2
  ---------
  NAME:     Roborock
  ID:       262063414
  MAC:      50:EC:50:1E:0A:0F
  IP:       192.168.0.42
  TOKEN:    4e4b4942326f4e4351427372374e554c
  MODEL:    roborock.vacuum.s5
```

```
---------
NAME:     Viomi
ID:       428598805
MAC:      C0:E7:BF:58:36:60
IP:       192.168.0.43
TOKEN:    73717a4562337a6a6c47433446666c4f
MODEL:    viomi.vacuum.v6
---------
```

# Clients einbinden

Linux: lnx link

Windows: IOT Link besser: Hass.Client

# Touch PC

Unified Remote:

https://www.unifiedremote.com/tutorials/how-to-install-unified-remote-server-deb-via-terminal

# KWL Pluggit

```
//
//□@author□□□Helge Klug
//□@copyright□□Copyright (c) 2018 Helge Klug
//□@version□□1.10
//
//□@see□□□
http://www.pluggit.com/fileserver/files/1413/609560454939420/21_10_2013_modbus_addresses.pdf

//□□Register□:□40025 | prmFWVersion□□□□□| UINT□| Read□|□FW Version: Major(8bits) and Minor (8bi
//□□Register□:□40101 | prmHALTaho1□□□□□□| Float□| Read□|□Fan1 rpm
//□□Register□:□40103 | prmHALTaho2□□□□□□| Float□| Read□|□Fan2 rpm
//□□Register□:□40109 | prmDateTime  □□□□□| UINT□| Read□|□Current Date/time in Unix time
//□□Register□:□40109 | prmDateTimeSet  □□□□| UINT□| Write□|□New date/time in Unix time
//□□Register□:□40133 | prmRamIdxT1  □□□□□| Float□| Read□|□Outdoor temperature T1, °C
//□□Register□:□40135 | prmRamIdxT2 □□□□□| Float□| Read□|□Supply  temperature T2, °C
//□□Register□:□40137 | prmRamIdxT3  □□□□□| Float□| Read□|□Extract temperature T3, °C
//□□Register□:□40139 | prmRamIdxT4  □□□□□| Float□| Read□|□Exhaust temperature T4, °C
//□□Register□:□40161 | prmPreheaterDutyCycle  □□□| UINT□| Read□|□Power of Preheater in %
//□□Register□:□40169 | prmRamIdxUnitMode□□□□| UINT□| Write□|□Active Unit mode:
//□□□□□□□□□□□□□□□□□□□□□□□□| Read□|□□Demand Mode□□□□0x0002□□2
//□□□□□□□□□□□□□□□□□□□□□□□□□□□□Manual Mode□□□□0x0004□□4
//□□□□□□□□□□□□□□□□□□□□□□□□□□□□WeekProgram Mode□□0x0008□□8
//
//□□□□□□□□□□□□□□□□□□□□□□□□□□□□Away Mode Start□□□0x0010□□16
//□□□□□□□□□□□□□□□□□□□□□□□□□□□□Away Mode End□□□0x8010□□40169
//
//□□□□□□□□□□□□□□□□□□□□□□□□□□□□Fireplace Mode Start□0x0040□□64
//□□□□□□□□□□□□□□□□□□□□□□□□□□□□Fireplace Mode End□□0x8040□□32832
//
//□□□□□□□□□□□□□□□□□□□□□□□□□□□□Summer Mode Start□□0x0800□□2048
//□□□□□□□□□□□□□□□□□□□□□□□□□□□□Summer Mode End□□□0x8800□□34816
//
//□□□□□□□□□□□□□□□□□□□□□□□□□□□□Select manual bypass□0x0080□□128
//□□□□□□□□□□□□□□□□□□□□□□□□□□□□Deselect Manual bypass□0x8080□□32896
//
```

```
//□□Register□:□40199 | prmRamIdxBypassActualState□□| UINT□| Read□|□Bypass state:
//□□□□□□□□□□□□□□□□□□□□□□□□□□□□□0□Closed□□0x0000
//□□□□□□□□□□□□□□□□□□□□□□□□□□□□□1□In process□0x0001
//□□□□□□□□□□□□□□□□□□□□□□□□□□□□□32□Closing□□0x0020
//□□□□□□□□□□□□□□□□□□□□□□□□□□□□□64□Opening□□0x0040
//□□□□□□□□□□□□□□□□□□□□□□□□□□□□□255□Opened□□0x00FF
//
//□□Register□:□40265 | prmRamIdxBypassManualTimeout□| UINT□| Read□|□Manual bypass duration in m
//
//□□Register□:□40325 | prmRomIdxSpeedLevel□□□□| UINT□| Write□|□Speed level of Fans ->  Manual m
step can be set
//□□□□□□40325 | prmRomIdxSpeedLevel□□□□| UINT□| Read□|□Speed level of Fans ->  Other modes: Fan
read.
//□□Register□:□40445 | prmBypassTmin □□□□□| Float□| Read□|□Min temperature for outdoor air (T1)
//□□Register□:□40447 | prmBypassTmax □□□□□| Float□| Read□|□Max temperature for extract air (T3)
//□□Register□:□40473 | prmCurrentBLState □□□□| UINT□| Read□|□Current unit mode:
//□□□□□□□□□□□□□□□□□□□□□□□□□□□□□0□Standby
//□□□□□□□□□□□□□□□□□□□□□□□□□□□□□1□Manual
//□□□□□□□□□□□□□□□□□□□□□□□□□□□□□2□Demand
//□□□□□□□□□□□□□□□□□□□□□□□□□□□□□3□Week program
//□□□□□□□□□□□□□□□□□□□□□□□□□□□□□4□Servo-flow
//□□□□□□□□□□□□□□□□□□□□□□□□□□□□□5□Away
//□□□□□□□□□□□□□□□□□□□□□□□□□□□□□6□Summer
//□□□□□□□□□□□□□□□□□□□□□□□□□□□□□7□DI Override
//□□□□□□□□□□□□□□□□□□□□□□□□□□□□□8□Hygrostat override
//□□□□□□□□□□□□□□□□□□□□□□□□□□□□□9□Fireplace
//□□□□□□□□□□□□□□□□□□□□□□□□□□□□□10□Installer
//□□□□□□□□□□□□□□□□□□□□□□□□□□□□□11□Fail Safe 1
//□□□□□□□□□□□□□□□□□□□□□□□□□□□□□12□Fail Safe 2
//□□□□□□□□□□□□□□□□□□□□□□□□□□□□□13□Fail Off
//□□□□□□□□□□□□□□□□□□□□□□□□□□□□□14□Defrost Off
//□□□□□□□□□□□□□□□□□□□□□□□□□□□□□15□Defrost
//
//□□Register□:□40555 | prmFilterRemainingTime  □□| UINT□| Read□|□Remaining time of the Filter L
(Days)
//□□Register□:□40557 | prmFilterDefaultTime  □□□| UINT□| Write□|□Filter Lifetime (Days)
//□□□□□□□□□□□□□□□□□□□□□□| Read□|
//□□Register□:□40559 | prmFilterReset  □□□□| UINT□| Write□|□Reset filter timer
//□□□□□□□□□□□□□□□□□□□□□□□□□□□|□Reset filter timer
//□□Register□:□40625 | prmWorkTime  □□□□□| UINT□| Read□|□Work time of system, in hours
```

```
//□□Register□:□40669 | prmStartExploitationDateStamp□| UINT□| Read□|□Date Stamp of the system s
time in Unix time
//      Register□:□40431 | prmVOC □□□□□□□□| UINT□| Read□|□VOC sensor value (read from VOC); ppm.
not installed, then 0.
//□□Register□:□40563 | prmPPM1Unit □□□□□□| UINT□| Read□|□Low Treshold of VOC (ppm)
//□□Register□:□40565 | prmPPM2Unit □□□□□□| UINT□| Read□|□Low Treshold of VOC (ppm)
//□□Register□:□40567 | prmPPM3Unit □□□□□□| UINT□| Read□|□Low Treshold of VOC (ppm)
//

Bridge modbus:tcp:PluggitAP310 [ host="192.168.0.15", port=502, id=2 ] {

□//---------------------------------------------------------------------------
□//□Status□□:□OK
□//
□Bridge poller prmFWVersion□□□□□[ start=25,□□length=4, refresh=5000,□□type="holding" ] {
□□Thing data register□□□□□□□[ readStart="25",□readValueType="uint32" ]
□}
□
□//---------------------------------------------------------------------------
□//□Status□□:□OK
□//
□Bridge poller prmHALTaho1□□□□□[ start=101,□□length=4, refresh=5000,□□type="holding" ] {
□□Thing data register□□□□□□□[ readStart="101",□readValueType="float32" ]
□}

□//---------------------------------------------------------------------------
□//□Status□□:□OK
□//
□Bridge poller prmHALTaho2□□□□□[ start=103,□□length=4,□refresh=5000,□type="holding" ] {
□□Thing data register□□□□□□□[ readStart="103",□readValueType="float32" ]
□}

□//---------------------------------------------------------------------------
□//□Status□□:□OK
□//
□Bridge poller prmDateTime□□□□□[ start=108,□□length=4,□refresh=5000,□type="holding" ] {
□□Thing data register□□□□□□□[ readStart="108",□readValueType="int64" ]
□□Thing data registerDateTime□□□□[ readStart="108",□readValueType="int64",
readTransform="JS(EpocheToDateTime.js)" ]
□}
```

```
    //---------------------------------------------------------------------
    //  Status  :  ???
    //
    Bridge poller prmDateTimeSet      [ start=108,   length=4,  refresh=5000,  type="holding" ] {
      Thing data register       [ writeStart="108", writeValueType="int64", writeType="holding" ]
    }

    //---------------------------------------------------------------------
    //  Status  :  OK
    //
    Bridge poller prmRamIdxT1      [ start=133,   length=4,  refresh=5000,  type="holding" ] {
      Thing data register       [ readStart="133",  readValueType="float32" ]
    }

    //---------------------------------------------------------------------
    //  Status  :  OK
    //
    Bridge poller prmRamIdxT2      [ start=135,   length=4,  refresh=5000,  type="holding" ] {
      Thing data register       [ readStart="135",  readValueType="float32" ]
    }

    //---------------------------------------------------------------------
    //  Status  :  OK
    //
    Bridge poller prmRamIdxT3      [ start=137,   length=4,  refresh=5000,  type="holding" ] {
      Thing data register       [ readStart="137",  readValueType="float32" ]
    }

    //---------------------------------------------------------------------
    //  Status  :  OK
    //
    Bridge poller prmRamIdxT4      [ start=139,   length=4,  refresh=5000,  type="holding" ] {
      Thing data register       [ readStart="139",  readValueType="float32" ]
    }

    //---------------------------------------------------------------------
    //  Status  :  ???
    //
    Bridge poller prmPreheaterDutyCycle    [ start=160,   length=2,  refresh=5000,  type="holding" ] {
```

□□Thing data register□□□□□□[ readStart="160",□readValueType="uint32_swap" ]
□}

□//------------------------------------------------------------------------
□//□Status□□:□???
□//
□Bridge poller prmRamIdxUnitMode□□□□[ start=168,□□length=4,□refresh=5000,□type="holding" ] {
□□Thing data register□□□□□□[ readStart="168",□readValueType="int64_swap", writeStart="169",□
writeValueType="int64_swap", writeType="holding" ]
□}

□//------------------------------------------------------------------------
□//□Status□□:□OK
□//
□Bridge poller prmRamIdxBypassActualState□[ start=198,□□length=4,□refresh=5000,□type="holding"
□□Thing data register□□□□□□[ readStart="198",□readValueType="uint32_swap" ]
□}

□//------------------------------------------------------------------------
□//□Status□□:□OK
□//
□Bridge poller prmRamIdxBypassManualTimeout□[ start=264,□□length=4,□refresh=5000,□type="holding
□□Thing data register□□□□□□[ readStart="264",□readValueType="uint32_swap" ]
□}

□//------------------------------------------------------------------------
□//
□//□Status□□:□OK
□//
□Bridge poller prmRomIdxSpeedLevel□□□[ start=324,□□length=4,□refresh=5000,□type="holding" ] {
□□Thing data□register□□□□□[ readStart="324",□readValueType="int64_swap", writeStart="324",□
writeValueType="int64_swap", writeType="holding"  ] //uint32
□}

□//------------------------------------------------------------------------
□//Anpassung Christoph!
□//Status: ???

□□Bridge poller prmVOC □□□□□[ start=430,□□length=4, refresh=5000,□□type="holding" ] {
□□Thing data register□□□□□□[ readStart="430",□readValueType="uint32_swap" ]

□}
□
□//------------------------------------------------------------------------
□//
□//□Status□□:□OK
□//
□Bridge poller prmBypassTmin□□□□□[ start=445,□□length=4,□refresh=5000,□type="holding" ] {
□□Thing data register□□□□□□[ readStart="445",□readValueType="float32" ]
□}

□//------------------------------------------------------------------------
□//□Status□□:□OK
□//
□Bridge poller prmBypassTmax□□□□□[ start=447,□□length=4,□refresh=5000,□type="holding" ] {
□□Thing data register□□□□□□[ readStart="447",□readValueType="float32" ]
□}

□//------------------------------------------------------------------------
□//□Status□□:□OK
□//
□Bridge poller prmCurrentBLState□□□□[ start=472,□□length=4,□refresh=5000,□type="holding" ] {
□□Thing data register□□□□□□[ readStart="472",□readValueType="uint32_swap" ]
□}

□//------------------------------------------------------------------------
□//□Status□□:□OK
□//
□Bridge poller prmFilterRemainingTime□□[ start=554,□□length=4,□refresh=5000,□type="holding" ] {
□□Thing data register□□□□□□[ readStart="554", readValueType="uint32_swap" ]
□}

□//------------------------------------------------------------------------
□//□Status□□:□OK
□//
□Bridge poller prmFilterDefaultTime□□□[ start=556,□□length=4,□refresh=5000,□type="holding" ] {
□□Thing data register□□□□□□[ readStart="556", readValueType="int64", writeStart="556",
writeValueType="int64", writeType="holding" ]
□}

```
 //------------------------------------------------------------------------
 //Status:
 //
 Bridge poller prmFilterReset[ start=558, length=4, refresh=5000, type="holding" ] {
  Thing data register[ readStart="558", readValueType="int64", writeStart="558",
writeValueType="int64", writeType="holding" ]
 }

 //------------------------------------------------------------------------
 //
 //Status:
 //
 Bridge poller prmWorkTime[ start=624, length=4, refresh=5000, type="holding" ] {
  Thing data register[ readStart="624", readValueType="uint32_swap" ]
 }

 //------------------------------------------------------------------------
 //
 //Status:OK
 Bridge poller prmStartExploitationDateStamp[ start=668, length=4, refresh=5000, type="holdin
{
  Thing data register[ readStart="668", readValueType="uint32_swap" ]
  Thing data registerDateTime[ readStart="668", readValueType="uint32_swap",
readTransform="JS(EpocheToDateTime.js)" ]
 }

}
```

# Beschattungsautomatik

Ausrichtung Haus überprüfen: [Sonnenatlas Bayern]

```
alias: Sonnenstand Fassade
description: ""
triggers:
  - entity_id: sensor.sun_solar_azimuth
    trigger: state
conditions: []
actions:
  - choose:
      - conditions:
          - condition: template
            value_template: "{{ 0 < states('sensor.sun_solar_azimuth') | float < 73 }}"
        sequence:
          - target:
              entity_id: input_boolean.sonne_ostseite
            action: input_boolean.turn_on
            data: {}
          - target:
              entity_id:
                - input_boolean.sonne_sudseite
                - input_boolean.sonne_westseite
            action: input_boolean.turn_off
            data: {}
      - conditions:
          - condition: template
            value_template: "{{ 73 <= states('sensor.sun_solar_azimuth') | float < 162 }}"
        sequence:
          - target:
              entity_id:
                - input_boolean.sonne_ostseite
                - input_boolean.sonne_sudseite
            action: input_boolean.turn_on
```

```
            data: {}
          - target:
              entity_id: input_boolean.sonne_westseite
            action: input_boolean.turn_off
            data: {}
      - conditions:
          - condition: template
            value_template: "{{ 162 <= states('sensor.sun_solar_azimuth') | float < 253 }}"
        sequence:
          - target:
              entity_id:
                - input_boolean.sonne_sudseite
                - input_boolean.sonne_westseite
            action: input_boolean.turn_on
            data: {}
          - target:
              entity_id: input_boolean.sonne_ostseite
            action: input_boolean.turn_off
            data: {}
      - conditions:
          - condition: template
            value_template: "{{ states('sensor.sun_solar_azimuth') | float >= 253 }}"
        sequence:
          - target:
              entity_id: input_boolean.sonne_westseite
            action: input_boolean.turn_on
            data: {}
          - target:
              entity_id:
                - input_boolean.sonne_ostseite
                - input_boolean.sonne_sudseite
            action: input_boolean.turn_off
            data: {}
mode: single
```