ioBroker.ems-esp Adapter Documentation – February 2024

ioBroker.ems-esp Adapter - Introduction

The adapter supports an interface to the Bosch Group's heating systems with EMS or EMS+ bus. (Buderus/Junkers/Netfit, etc.) The adapter can be used for both

- with the original LAN interfaces of the Bosch Group heaters (IP-inside, km200, km100, km50, MB-LAN2 etc.)
- as well as to the EMS bus gateway with ESP32 chip. (https://github.com/emsesp/EMS-ESP32). The EMS Bus Gateway can be ordered from BBQKees: BBQKees Electronics – EMS bus to Home Automation interfaces (bbqkees-electronics.nl)

The newer Bosch interfaces (e.g. MX200/MX300 etc.) no longer support local LAN access and are not supported by the adapter.

The ems-esp gateway is a small box that is connected to the service port or directly to the EMS bus of the heating system / heat pump and which then establishes the connection between the heating system and the home automation system via WLAN/LAN and MQTT/WEB-API.

Together with the software developers of the EMS-ESP firmware, the WEB API has been adapted so that this ems-esp ioBroker adapter can be seamlessly integrated.

API-Calls

The adapter can read and write data on both gateways via WEB API to control all heating components. It can be used either for the original gateways of the Bosch Group or the ems-esp gateway or both in parallel.

The WEB-API communication to the km200 gateway is encrypted, the communication to the EMS-ESP gateway is not.
Write operations are ensured by a generated access token in the ems-esp gateway.

	ioBroker Adapter			
regular polling	< ----- API ---------- > website LAN en- / decrypted	**km200 Gateway**	< -------- >	**E**
regular polling	< ----- API /V3 ------- > web LAN / WLAN	**ems-esp Gateway**	< -------- >	**M** **BUS**
Energy polling & Calculation additional functions	< ----- Adapter Logic ------ >	**ems-esp/km200**	< -------- >	**S**

Differences Original Bosch Gateways (km200) vs. Ems-Esp Gateway

The original Bosch gateways ensure that the heating system can be operated via the Internet via the corresponding apps of the Bosch Group. The range of functions is very limited. Communication takes place via the Bosch cloud. Essentially, the switching times, vacation times and temperature specifications can be adjusted. A few plant data are readable.

Bosch does not have a documented API. Communication with the original gateways is encrypted – even in the local LAN (see below). The km200 gateway does not support any control or system parameters in the LAN. API accesses must be made separately for each data field. This means that the reading process (polling) is slow and the polling cycle is limited by the adapter to a minimum of 90 seconds.

The Ems-ESP Gateway is a separate piece of hardware that directly accesses the telegram traffic in the EMS bus. Since these telegrams are not documented, it has been a lot of detective work to decipher and implement them for the different heating systems. Many heating parameters and settings are supported, but other user settings are not (yet) available. A variety of different (including older) heating systems without internet gateway and brands are supported.

The data is read out very quickly, which means that short polling cycles of 15 seconds are also possible.

Adapter Sequence Logic

In the adapter it is possible to select whether a KM200 and/or an EMS-ESP gateway should be read. If both are to be present, it is recommended to display the EMS-ESP data structure in the KM200 logic as well.

In addition, the "Parameters" tab allows you to select whether statistics are to be created or whether the boiler efficiency is to be determined as a function of the temperatures. (calorific effect). In addition, there is a heating demand control system (see below)

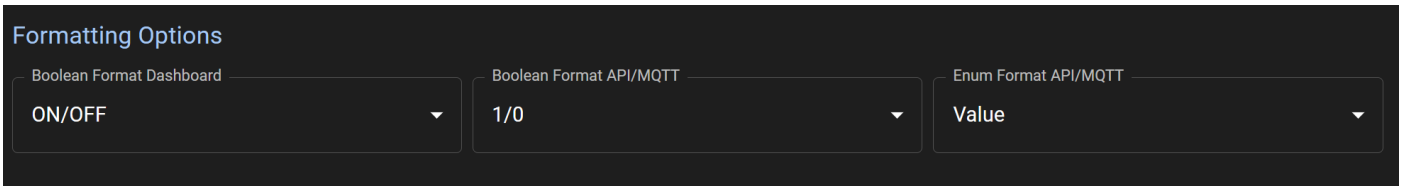
When the adapter is started, all previous states can be deleted (see parameters). This is only recommended if data structures are changed.

After starting, all details of the states are read once via API and the corresponding ioBroker objects are generated. After that, the current values are read and the states are updated in a regular polling cycle.

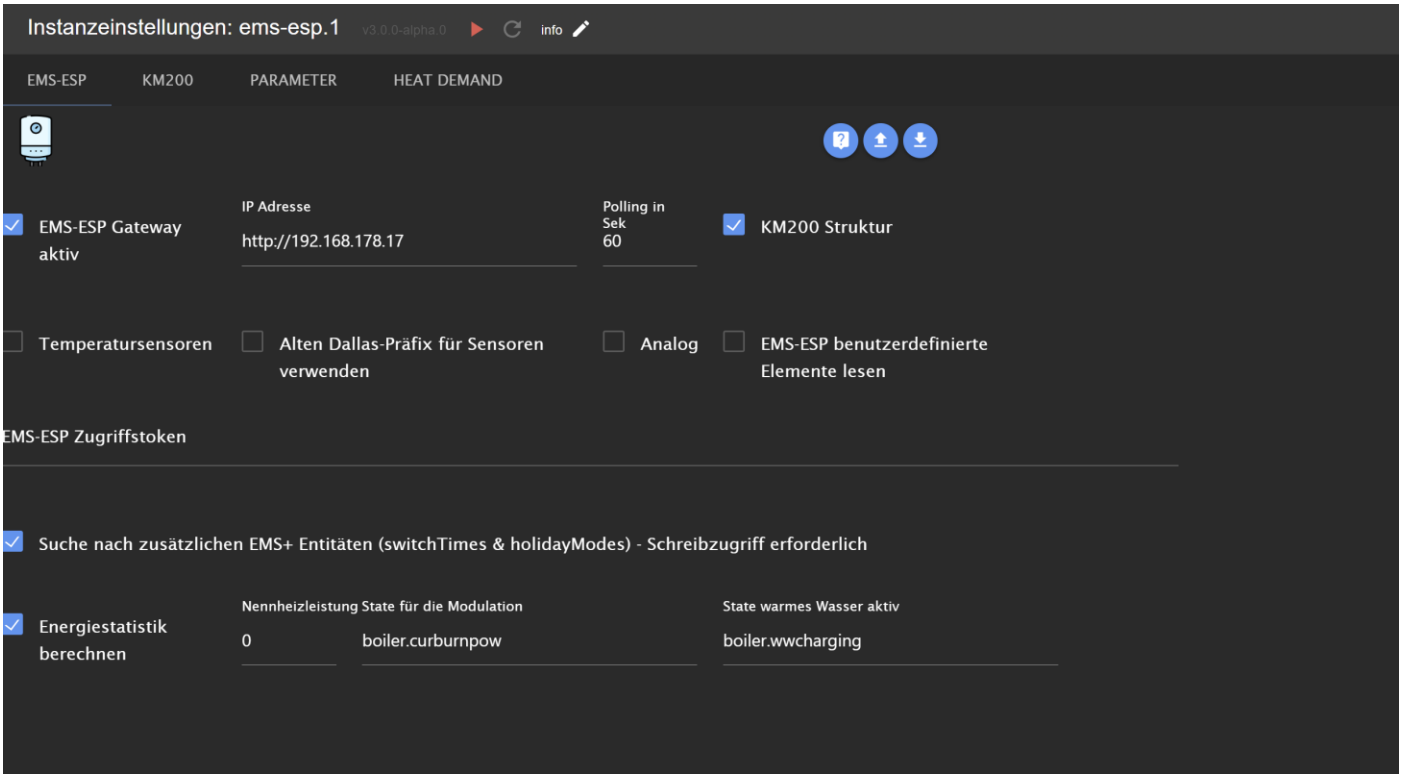
All other processing is carried out independently in its own processing cycles.

EMS-ESP settings:

In the EMS-ESP Gateway, the "Formatting Options" for the API interface must be set under Settings.
Formatting options must be 1/0 for the Boolean format and Index or Value for the enum format.



In order for the ioBroker adapter to be able to write values via API, this must be allowed in the settings. I recommend beginners to start with "Bypass Access Token authorization on API calls". and later generate the token for access protection and enter it in the adapter configuration. ioBroker Instance Settings:



EMS-ESP Settings:

The km structure checkbox either uses the km200-like device structure for EMS ESP data fields or keeps the original EMS ESP device view: boiler, thermostat, mixer, etc. Otherwise, the IP address of the gateway, the polling time and the access token must be entered. In addition, the reading of connected Dallas and analog sensors can be enabled. It is now also possible to activate the creation of energy consumption statistics (see chapter Energy).

NEW: The current ems-esp firmware does not support switchPrograms and holidayModes for EMS+ thermostats (RC300 / RC310 or similar). These objects are present in the original KM200 gateway.

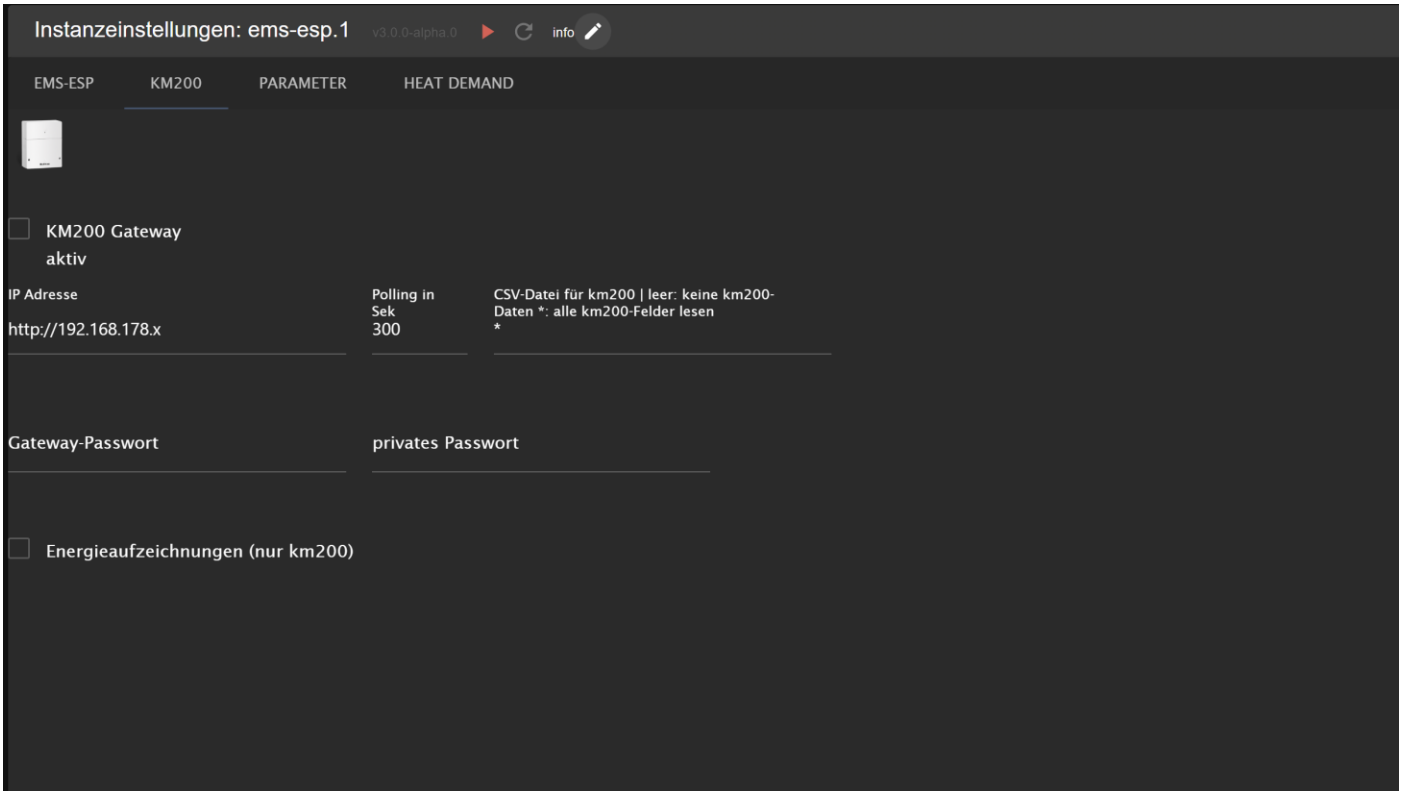
With the search for additional EMS+ entities, adapter version 3.0.0 can be used to check whether these values are present with RAW telegrams. If so, the corresponding objects/states are created in the same format as the KM200 gateway. In this case, the KM200 Gateway must not be selected, as the objects are created under the same name.

Settings KM200

Web API calls to/from the km200 gateway are encrypted. Two passwords are required for encryption/decryption:

- The gateway password on a label on the gateway in the form: xxxx-xxxx-xxxx-xxxx
  (case sensitive and include the hyphenaries)
- The private password is the one set with the Buderus **MyDevice** app!
- (**do not use myBuderus** or similar cloud apps!)

ioBroker Instance Settings:



You have to enter the IP address of the gateway, the polling time and the two passwords.

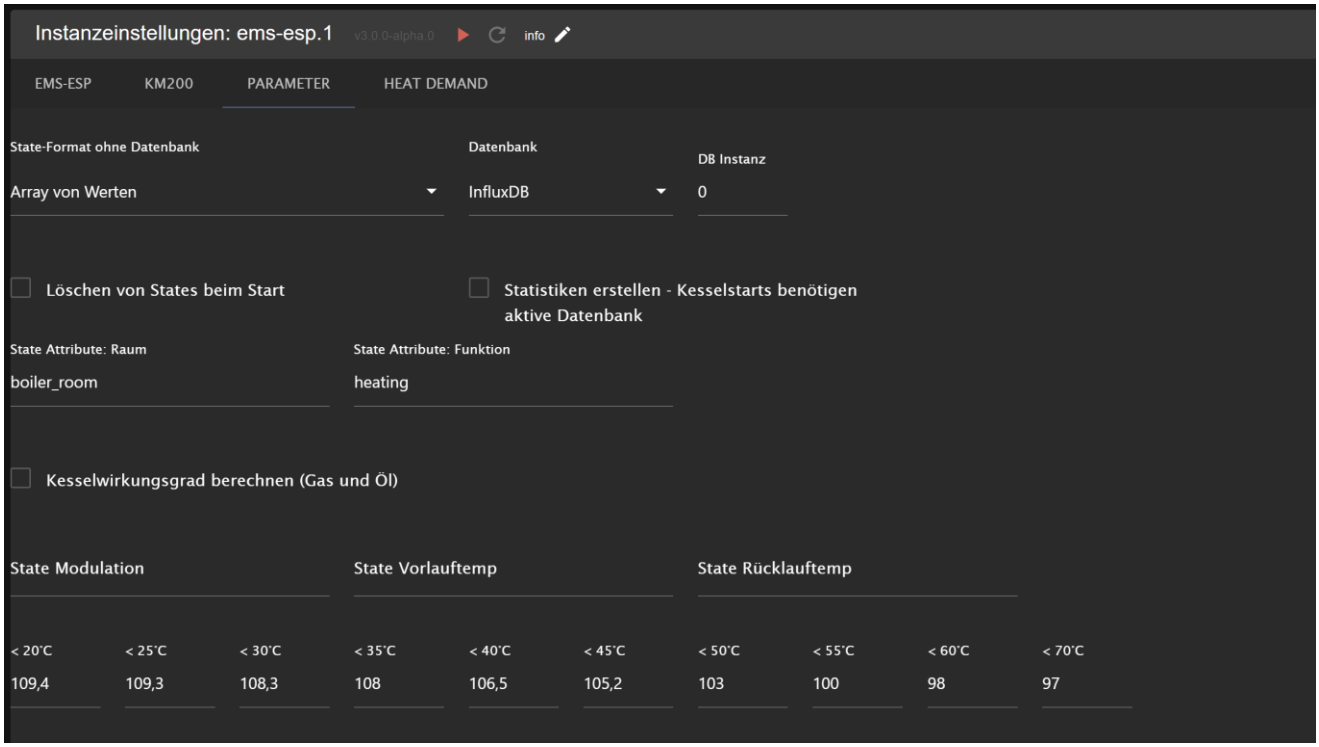
At the 1st adapter start, it is recommended to select all km200 data fields with a "*". The adapter then creates a km200.csv file in the .. /iobroker-data/ems-esp/{instance}.

This file can be used the next time you start the adapter instance. Unneeded rows (fields) can be deleted to reduce the number of km200 fields to be read. (Make a copy and rename file). With File, the staes must not be deleted at the start of the adapter, as the objects are not recreated.

The km200 Web API requires querying each individual value with its own http-get command. Even with systems with 2 heating circuits, this can be about 150 individual queries. A single query cycle takes a correspondingly long time (20-60 seconds).

- KM200 polling is also a parameter (default 300 seconds) and the minimum adjustable value is 90 seconds. (see above duration of the query cycle)
- km200-recordings (energy consumption and temperature statistics) are updated hourly

Parameter



In the Parameters tab, select the recordings formats without database and the database. mySQL / MariaDB, History and InfluxDB v1 and V2 are supported.
InfluxDB V2 requires the InfluxDB adapter version >= 4.0.2 for proper operation!!

For InfluxDB V2, the retention policy should be set to at least 60 weeks. (Change retention policy globally for ioBroker duration 60w;) to be able to save all values. In the current adapter version > 2.8.0, the value of the retention period is determined and only data within this time is updated.

The databases are also needed for the statistics. (see statistics)

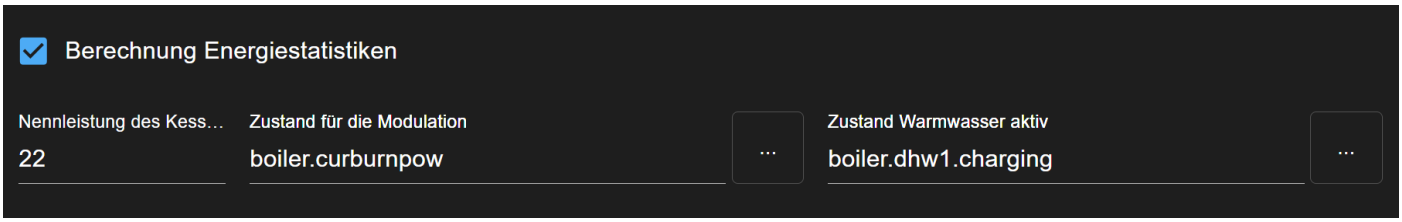
At the instance start, the states can be reinitialized (deleted). However, this only makes sense if the data structure is changed. E.g. if the EMS-ESP gateway structure is changed to KM200 structure. To be on the safe side, entries with history / DB entries are retained and may have to be deleted manually.

In addition, the boiler data for the boiler efficiency can be entered. These must be determined from the boiler's data sheet if this function is to be activated.

Energy consumption statistics for EMS-ESP

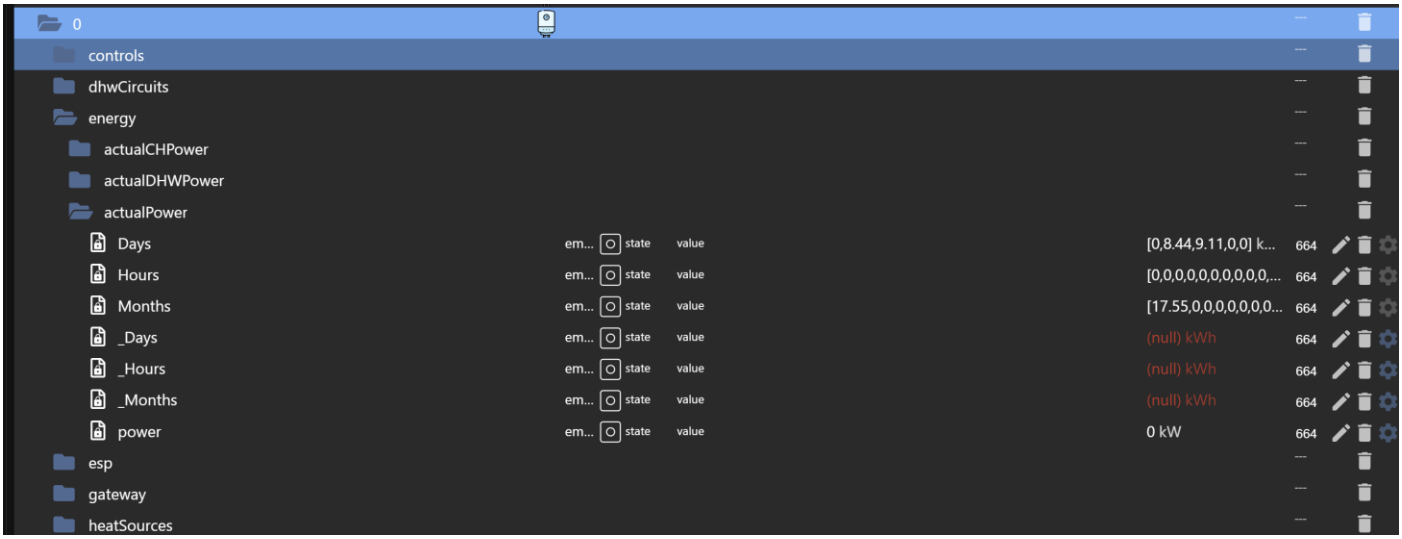
The EMS-ESP Gateway does not calculate the consumption values in the firmware. This calculation is now implemented in the ioBroker Adapter. Historical values are not readable. For the EMS-ESP Gateway, values are determined and updated every 15 seconds.
The energy statistics can be activated in the instance and require an active database instance. (History, mySQL, InfluxDB).

EMS-ESP:



You have to enter the nominal heating capacity of the boiler and the state names for the current modulation and for WW-Aktiv. On this basis, the current burner, heating (CH) and hot water output (DHW) are then determined.

In the object structure, three substructures are then created under energy: actualCHPower (heating), actualDHWPower (hot water) and actualPower (total).



The power states are updated every 15 seconds and then the energy consumption values are determined every 10 minutes. The consumption values can then be displayed graphically with e.g. Flot.

- hourly values below _Hours
- the daily values below _Days
- the monthly values below _Months

with database direct access saved/updated. (That's why zero). The JSON array values are updated depending on the format selection (current values first):

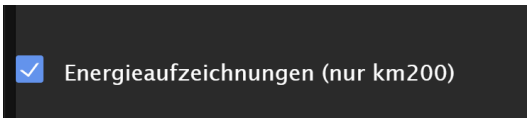
- Hourly values under Hours
- the Daily Values under Days

- the monthly values under Months

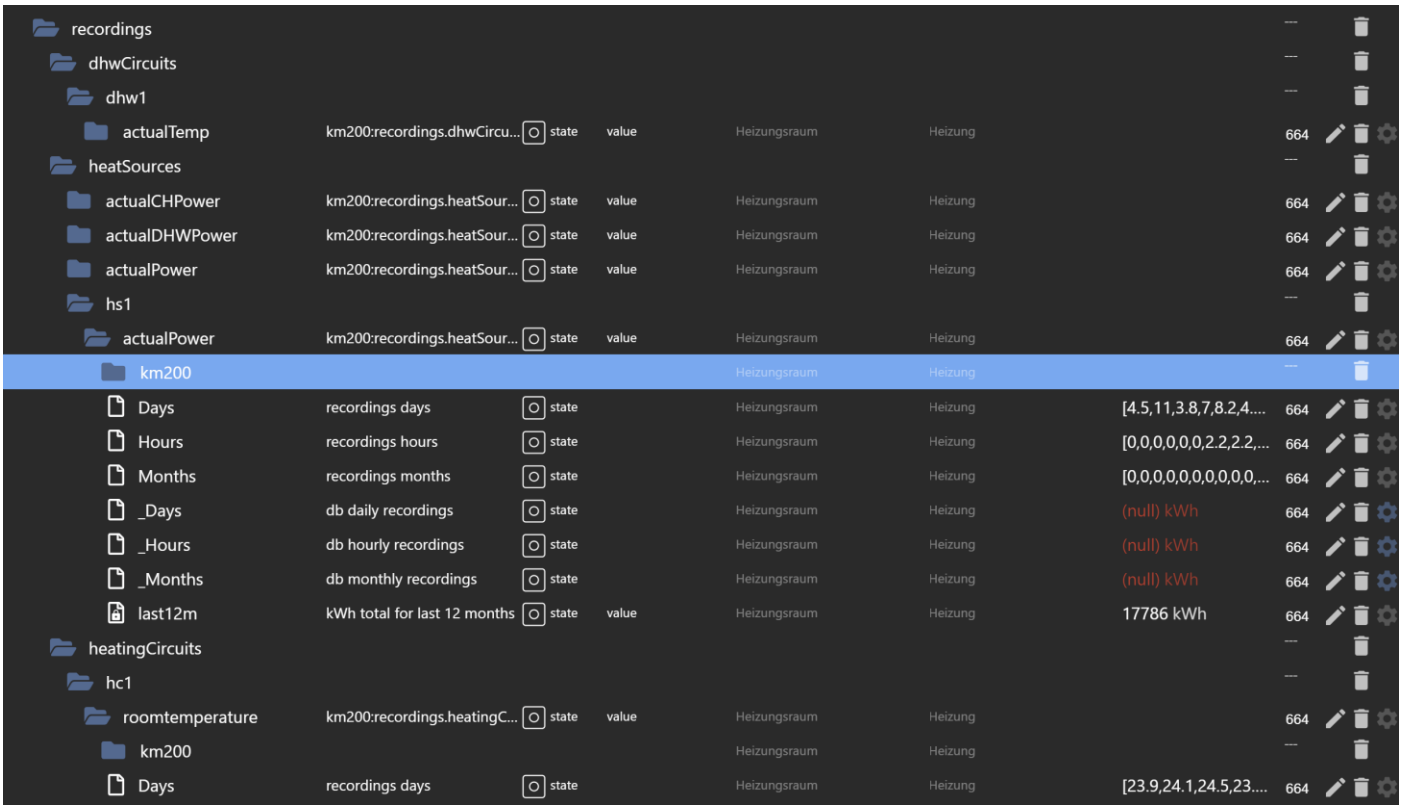
Energy Consumption Statistics for KM200 (Recordings)

Most modern heating systems have an IP-Inside or KMxxx gateway and support energy and temperature statistics. These are calculated in the gateway. As a rule, consumption and temperature values for the last 12 months can be retrieved. The Bosch gateways calculate the consumption values by means of "samples" every 60 seconds. Hourly, daily and monthly values can be read. The adapter reads these values every hour.

KM200:



The checkbox "Energy recordings" must be activated and the database instance (History, mySQL or InfluxDB) must be defined under Parameters and of course also be active.



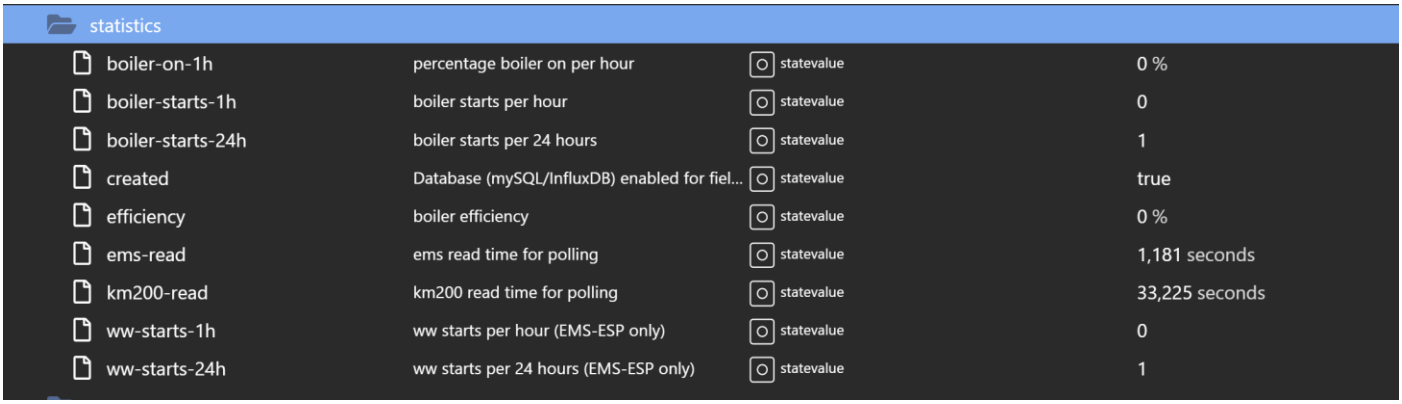
The object structure is specified by the Bosch API. There are consumption values and temperatures.

As with the EMS-ESP, it is divided into hours, days and months as arrays of values and in the states _Hours, _Days and _Months directly as database entries for graphical representation.

IMPORTANT: For databases, the data is written directly to the "_" states using database commands. Under Objects, the value (null) is displayed. (That's right!!). See previous chapter.

Statistics

Burner statistics can be enabled and show:



- boiler-on-1h: What percentage (0-100%) was the boiler active during the last hour?
- boiler-starts-1h and boiler-starts-24h: Number of boiler starts in the period (1 / 24 hours)
- created: Indicator that the statistics structure has been created
- efficiency: current boiler efficiency when activated (condensing boiler benefit for gas and oil boilers)
- ems-read: The query cycle processing time for EMS ESP gateway reads
- km200-read: ... analog for KM200
- ww-starts-1h and ww-starts-24h (only with active EMS-ESP gateway) – boiler starts for hot water preparation

To calculate the statistics, an active database instance (see above) is required.

Condensing Value Benefits – Burner Efficiency

The boiler efficiency can be calculated when the parameters are filled in.
(gas and oil boilers only)

Efficiency (condensing value) is calculated based on the average boiler temperature: (boiler temperature + return temperature) / 2.

Check your boiler's data sheet to adjust the efficiency chart accordingly.

The states for modulation, flow and return temperature must be entered.

Heat demand control in the ems-esp adapter

In the current version of the ems-esp adapter, the heating system can be controlled as a function of a calculated heat demand.

There is a separate configuration page "Heat Requirements" in which there are 2 input lists:(New entries with the + symbol)

In the first block, the following entries are defined for each room (name of your choice):

- *Settemp*: State for the set temperature of the radiator / room
- *Actual temp*: State for the actual temperature of the room
- *Minimum delta* – difference between settemp – actualtemp from which the heating requirement exists:Example: Target 21° - Is 20→° delta 1°. If delta >= minimum → delta then heating demand.Minimum delta = 0 means heating demand if the current temperature is equal to or less than the target temperature.
- *Hc* : Assignment to the heating circuit (hc1 ... hc4)
- *Weight*: Weighting of the radiator / room (What is the heating capacity of the radiator or the underfloor heating?).

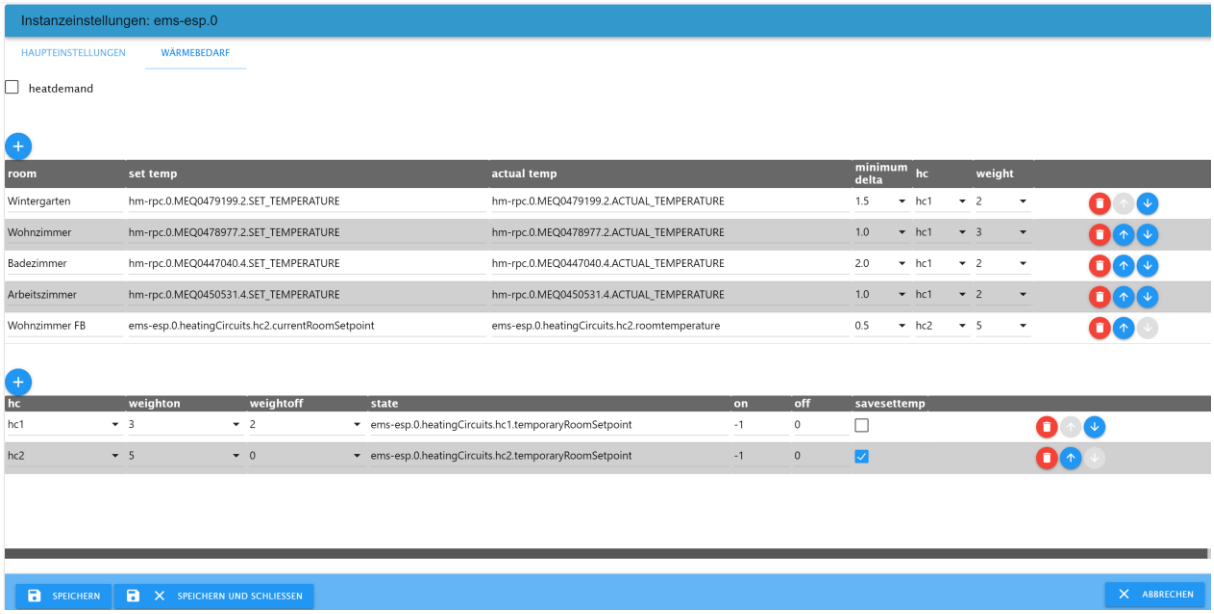
The current heat demand per room / thermostat is determined with the following hysteresis. Once Actual Temp < (Settemp – Delta) → then heating demand is active with the weighting as specified. Once Actual Temp > Settemp → then the heating requirement is deactivated

In the second block, the following are defined for each heating circuit:

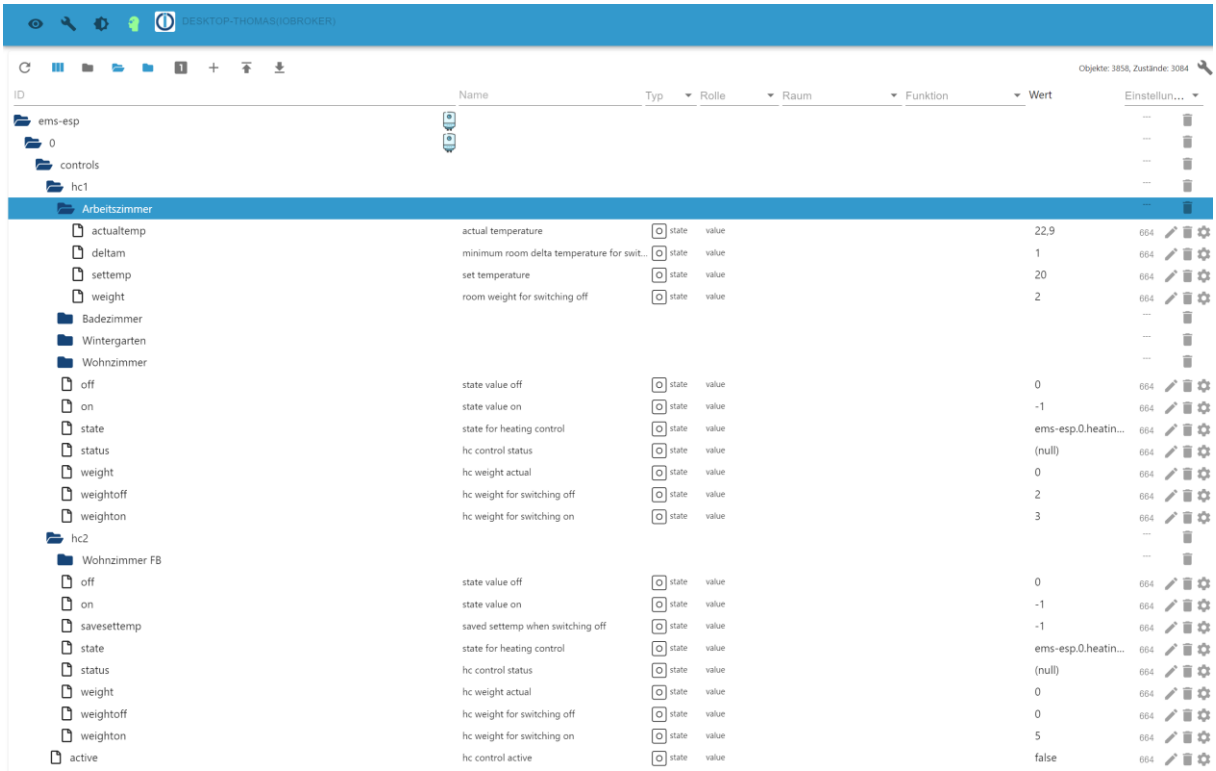
- *Weighton*: **HK on** at the sum of the weighting values of the heating circuit >= weighton
- *Weightoff*: **HK off** at sum of the weighting values of the heating circuit <= weightoff
- *State*: state to be switched
- *On*: Value of the state for HK.
- *Off*: Value of the state for HK.
- *Savetemp*: If enabled, the current setpoint is stored and this value is taken as a reference when the heating circuit is switched off. This is necessary because when the underfloor heating circuit is switched off, the setpoint is set to 0. If the heating demand is calculated from your own thermostats, then savsettemp must be switched off !

The *heatdemand* switch turns the automatic heat demand control on or off. This state can be changed during an active instance – e.g. via VIS.

Here is an example configuration with Homematic thermostats (hc1) and km200 mixer-controlled underfloor heating circuit (hc2).



In the object structure of the ems adapter, the following object states are created under controls after the instance has been started :



The last state active is preset with the value of *heatdemand* at the start of the adapter and controls whether the heat demand-dependent control is active (true) or inactive (false). The value can then be set, for example, via VIS. For example, when the adapter is started, the control system is inactive for the time being without the *heatdemand* value set and can be activated later in VIS.

It is important to choose the state to be switched carefully. It would be possible, for example, to switch the heating circuit off or on during summer / winter operation (e.g. km200: heatingCircuits.hc1.suWiSwitchMode)This has the disadvantage that in case of adapter stop or network problems (km200 not available) the heating circuit may remain permanently off or on and must be manually reset on the thermostat. According to Murphy's Law, this usually happens during vacation/absence....

That's why I prefer the "temporary setpoint" (e.g. heatingCircuits.hc1.temporaryRoomSetpoint).
With my RC310, these temporary adjustment options are available for each heating circuit.

This state has the advantage that value changes are only valid temporarily until the next switching time of the heating program. The value "0" turns the HK off, the value "-1" turns it back on to automatic mode. (But it would also be possible to set a fixed temperature: e.g. "21" degrees.)

I use the automatic mode so that an automatic heating circuit shutdown continues to work after reaching the outside temperature threshold of the heating circuit, despite the active heat demand control.

The weighting of the radiators, the switch-on threshold and the "*minimum delta" of the* heating circuit should be determined in such a way that the heat output of the heat generator can be absorbed for a sufficient period of time with the smallest modulation. With a *"minimum delta*" of 1°, my Homematic thermostats ensure that the valves are open. If all rooms / thermostats are weighted equally (e.g. with 1) and the switch-on threshold of the heating circuit is also 1, then every heating demand of a room will turn on the burner.