

ebusd installieren:

<https://github.com/john30/ebusd/wiki/1.-Build-and-install>

bus koppler von esera:

<https://www.esera.de/produkte/ebus/135/1-wire-hub-platine>

ich habe einen von esera und einen von der sammelbestellung auf einem raspi laufen.
alternativ kann man wie man möchte oder wie die gegenheiten sind aufteilen.

2 koppler braucht man auf jeden fall weil beide geräte (wärmepumpe und lüftung) auf derselben adresse laufen.

Jeder koppler kann daher nur ein signal auffangen.

ttyebus treiber installieren falls platine vorhanden:

(<https://forum.fhem.de/index.php?topic=104268.0>)

<https://github.com/eBUS/ttyebus>

ebusd kopieren falls man 2 koppler am laufen hat:

```
sudo cp /lib/systemd/system/ebusd.service /lib/systemd/system/ebusd2.service
```

```
sudo nano /lib/systemd/system/ebusd2.service
```

```
systemctl daemon-reload
```

```
sudo reboot
```

autostart für den ebusd2 aktivieren:

```
sudo systemctl enable ebusd2
```

```
sudo nano /lib/systemd/system/ebusd2.service
```

folgende zeilen anpassen:

```
PIDFile=/var/run/ebusd2.pid
```

```
ExecStart=/usr/bin/ebusd $EBUSD_OPTS1
```

```
sudo nano /etc/default/ebusd
```

so sehen die beiden bei mir aus: alles mit xxx ist mit den passenden daten zu ersetzen

```
EBUSD_OPTS="--pidfile=/var/run/ebusd.pid -c /etc/ebusd -d /dev/ttyebus -p 8888 -l  
/var/log/ebusd.log --enabledefine --enablehex --mqtttopic=ebusd/%circuit/%name --  
mqttthost=xxx --mqttpport=1883 --mqttuser=xxx --mqttpass=xxx --accesslevel=* --  
scanconfig=full --mqttjson"
```

```
EBUSD_OPTS1="--pidfile=/var/run/ebusd2.pid -c /etc/ebusd -d /dev/ttyUSB0 -p 8889 -l  
/var/log/ebusd2.log --enabledefine --enablehex --mqtttopic=ebusd2/%circuit/%name --  
mqttthost=xxx --mqttpport=1883 --mqttuser=xxx --mqttpass=xxx --accesslevel=* --  
scanconfig=full --mqttjson"
```

```
sudo service ebusd status
```

bzw.

```
sudo service ebusd2 status
```

sollten jetzt ohne fehlermeldungen funktionieren

```

[pi@raspberrypi:~ $ sudo service ebusd status
● ebusd.service - ebusd, the daemon for communication with eBUS heating systems.
   Loaded: loaded (/lib/systemd/system/ebusd.service; enabled; vendor preset: enabled)
   Active: active (running) since Sat 2020-10-31 12:32:28 CET; 20min ago
     Process: 694 ExecStart=/usr/bin/ebusd $EBUSD_OPTS (code=exited, status=0/SUCCESS)
    Main PID: 695 (ebusd)
      Tasks: 5 (limit: 2200)
     Memory: 1.4M
    CGroup: /system.slice/ebusd.service
            └─695 /usr/bin/ebusd --lograwdatafile=/var/log/ebusraw.log --pidfile=/var/run/ebusd.pid -c /etc/

Oct 31 12:32:28 raspberrypi systemd[1]: Starting ebusd, the daemon for communication with eBUS heating systems.
Oct 31 12:32:28 raspberrypi systemd[1]: Started ebusd, the daemon for communication with eBUS heating systems.
[pi@raspberrypi:~ $ sudo service ebusd2 status
● ebusd2.service - ebusd, the daemon for communication with eBUS heating systems.
   Loaded: loaded (/lib/systemd/system/ebusd2.service; enabled; vendor preset: enabled)
   Active: active (running) since Sat 2020-10-31 12:31:58 CET; 21min ago
     Process: 671 ExecStart=/usr/bin/ebusd $EBUSD_OPTS1 (code=exited, status=0/SUCCESS)
    Main PID: 672 (ebusd)
      Tasks: 5 (limit: 2200)
     Memory: 1.1M
    CGroup: /system.slice/ebusd2.service
            └─672 /usr/bin/ebusd --pidfile=/var/run/ebusd2.pid -c /etc/ebusd -d /dev/ttyUSB0 -p 8889 -l /var/

Oct 31 12:31:58 raspberrypi systemd[1]: Starting ebusd, the daemon for communication with eBUS heating systems.
Oct 31 12:31:58 raspberrypi systemd[1]: Started ebusd, the daemon for communication with eBUS heating systems.
[pi@raspberrypi:~ $ █

```

ebusctl -p 8888 info

bzw.

ebusctl -p 8889 info

ebenfalls

```

pi@raspberrypi:~ $ ebusctl -p 8888 info
version: ebusd 3.4.v3.3-51-g57eae05
update check: revision v3.4 available, broadcast.csv: different version available, vaillant/08.hmu.csv: dif
access: *
signal: acquired
symbol rate: 48
max symbol rate: 127
min arbitration micros: 8
max arbitration micros: 46
min symbol latency: 4
max symbol latency: 4
reconnects: 0
masters: 4
messages: 70
conditional: 0
poll: 0
update: 11
address 03: master #11
address 08: slave #11, scanned "MF=Vaillant;ID=HMU01;SW=0300;HW=8802", loaded "vaillant/08.hmu.csv"
address 10: master #2
address 31: master #8, ebusd
address 36: slave #8, ebusd
address 71: master #9
address 76: slave #9, scanned "MF=Vaillant;ID=VWZIO;SW=0107;HW=0103"
address e3: slave, scanned "MF=Vaillant;ID=FMU00;SW=0202;HW=6502"
address e8: slave, scanned "MF=Vaillant;ID=FMU00;SW=0202;HW=6502"

pi@raspberrypi:~ $ ebusctl -p 8889 info
version: ebusd 3.4.v3.3-51-g57eae05
update check: revision v3.4 available, broadcast.csv: different version available, vaillant/broadcast.csv:
access: *
signal: acquired
symbol rate: 23
max symbol rate: 58
min arbitration micros: 730
max arbitration micros: 1029
min symbol latency: 5
max symbol latency: 10
reconnects: 0
masters: 3
messages: 49
conditional: 0
poll: 0
update: 9
address 03: master #11
address 08: slave #11, scanned "MF=Vaillant;ID=recoV;SW=0223;HW=0203", loaded "vaillant/08.recov.csv"
address 31: master #8, ebusd
address 33: master #13
address 36: slave #8, ebusd
address 38: slave #13, scanned "MF=Vaillant;ID=V32;SW=0117;HW=9802"

```

recovair file per hand erstellen falls noch nicht automatisch im system:

```
cd /etc/ebusd
```

```
cd vaillant
```

```
dir
```

```
sudo nano 08.recov.csv
```

folgenden inhalt kopieren:

```

# type (r[1-9];w;u)",circuit,name,[comment],[QQ],ZZ,PBSB,[ID],field1,part
(m/s),datatypes/templates/divider/values/unit,comment
#,recov,recovAir 360/4,,,,,,,,,,,,,
# ##### Generell #####,,,,,,,,,,,,,
*r,,,,,B513,,,,IGN:2,,

```

```

*w,,,,,B509,,,,,
r,recov,IntensTag,Lueftungsintensitaet Tag auslesen,,,,048a03,HEX:15,,UIN,,Werte 1-6
r,recov,IntensNacht,Lueftungsintensitaet Nacht auslesen,,,,048b03,HEX:15,,UIN,,Werte 1-6
r,recov,WaermeRueck,Waermerueckgewinnung
auslesen,,,,048c03,HEX:15,,UIN,0=auto;1=on;2=off,,Werte auto/on/off
w,recov,IntensTag,Lueftungsintensitaet Tag setzen,,,,0e8a03,,,UCH,,Werte 1-6
w,recov,IntensNacht,Lueftungsintensitaet Nacht setzen,,,,0e8b03,,,UCH,,Werte 1-6
w,recov,HeatRecovery,Waermerueckgewinnung,,,,0e8c03,,,UCH,,Werte 0=auto;1=on;2=off
w,recov,SetTag,Lueftung Tag setzen,,,B522,000200FFFF,,,,,
w,recov,SetNacht,Lueftung Nacht setzen,,,B522,000100FFFF,,,,,
w,recov,SetIntensiv,Stosslueften aktivieren,,,B522,000300FFFF,,,,,
w,recov,SetIntensivAb,Stosslueften aktivieren,,,B522,000000FFFF,,,,,
w,recov,SetWunschTemp,Wunschtemperatur,,,,0e140e,,,temp,,,Wert ändert sich erst nach
restart von EBUSD?

```

```
# ##### Read #####,,,,,,,,,,,,,
```

```

*r,,,,,B509,,,,,IGN:2,,
r,recov,BypassPosition,Position der Bypass-Klappe,,,,290b0e,,,UIN,10,%,
r,recov,Frostschutz,Position der Bypass-Klappe,,,,29f10c,,,onoff,,,
r,recov,WunschTemp,Wunschtemperatur,,,,29140e,,,temp,,°C,
r,recov,Filterwechsel,Drehzahl Zuluft,,,,29360e,,,UIN,,Tage,
r,recov,Volumenstrom,Drehzahl Zuluft,,,,29030e,,,UIN,,m3/h,
r,recov,Zuluft,Drehzahl,,,,29FC0D,,,UIN,,rpm,
r,recov,TempZuluft,Temperatur Zuluft,,,,29b30d,,,temp,,°C,
r,recov,TempAbluft,Temperatur Abluft,,,,29b40d,,,temp,,°C,
r,recov,TempAussenluft,Temperatur Aussenluft,,,,290c0e,,,temp,,°C,
r,recov,TempFortluft,Temperatur Fortluft,,,,290d0e,,,temp,,°C,
r,recov,FeuchtigkeitAbluft,Feuchtigkeit Abluft,,,,298503,,,percent,,%,
r,recov,ErtragHeute,Ertrag Heute,,,,29560E,,,UIN,,kw/h,
r,recov,ErtragVortag,Ertrag Vortag,,,,29570E,,,UIN,,kw/h,
r,recov,ErtragMonat,Ertrag Monat,,,,29580E,,,UIN,,kw/h,
r,recov,ErtragJahr,Ertrag Jahr,,,,29590E,,,UIN,,kw/h,
r,recov,ErtragGesamt,Ertrag Gesamt,,,,295A0E,,,UIN,,kw/h,

```

php server installieren:

```
sudo apt-get install php
```

wieder ins root verzeichnis wechseln

```
cd /var/www/html/
```

php file erstellen (fungiert als schnittstelle zu loxone)

```
sudo nano ebusd.php
```

```
<?php
```

```
# Adresse der Recovair im eBus - Standard: 08
```

```
# IP des Rechners mit der eBus-Installation - 10.0.1.24 ersetzen!
```

```
# Gewählter Port in eBus für Telnet - Standard: 8888
```

```
$ebusIP = "10.0.1.24";
```

```

$port = $_GET["port"];
$mode = $_GET["mode"];
$value = $_GET["value"];
$set = $_GET["set"];

$fp = fsockopen($sebusIP, $port, $errno, $errstr, 30);
if (!$fp) {
    echo "$errstr ($errno)<br />\n"; }
else {
    switch($mode)
    {

case ("read"):
    $out = "r -f ".$value."\r\n"; fwrite($fp, $out);
    $result = fgets($fp, 128);
    //($result == 00)
    echo "value:". $result;
    break;

case ("hex"):
    $out = "hex ".$value."\r\n"; fwrite($fp, $out);
    $result = fgets($fp, 128);
    echo "HEX:", substr($result,2,1);
    break;

case ("write700"):
    $out= "w -c 700 ".$value." ".$set."\r\n"; fwrite($fp, $out);
    $result = fgets($fp, 128);
    //if ($result == 00)
    echo "700:", $result;
    break;

case ("writehmu"):
    $out= "w -c hmu ".$value." ".$set."\r\n"; fwrite($fp, $out);
    $result = fgets($fp, 128);
    //if ($result == 00)
    echo "hmu:", $result;
    break;

case ("writerecov"):
    $out= "w -c recov ".$value." ".$set."\r\n"; fwrite($fp, $out);
    $result = fgets($fp, 128);
    //if ($result == 00)
    echo "recov:", $result;
    break;

case ("setmode"):
    $out= "w -c hmu SetModeOverride ". "0;11.0;-;-;1;1;1;0;0;1"."\r\n"; fwrite($fp, $out);

```

```
$result = fgets($fp, 128);
//if ($result == 00)
echo "setmode:", $result;
break;

}
fclose($fp);
}

?>
```

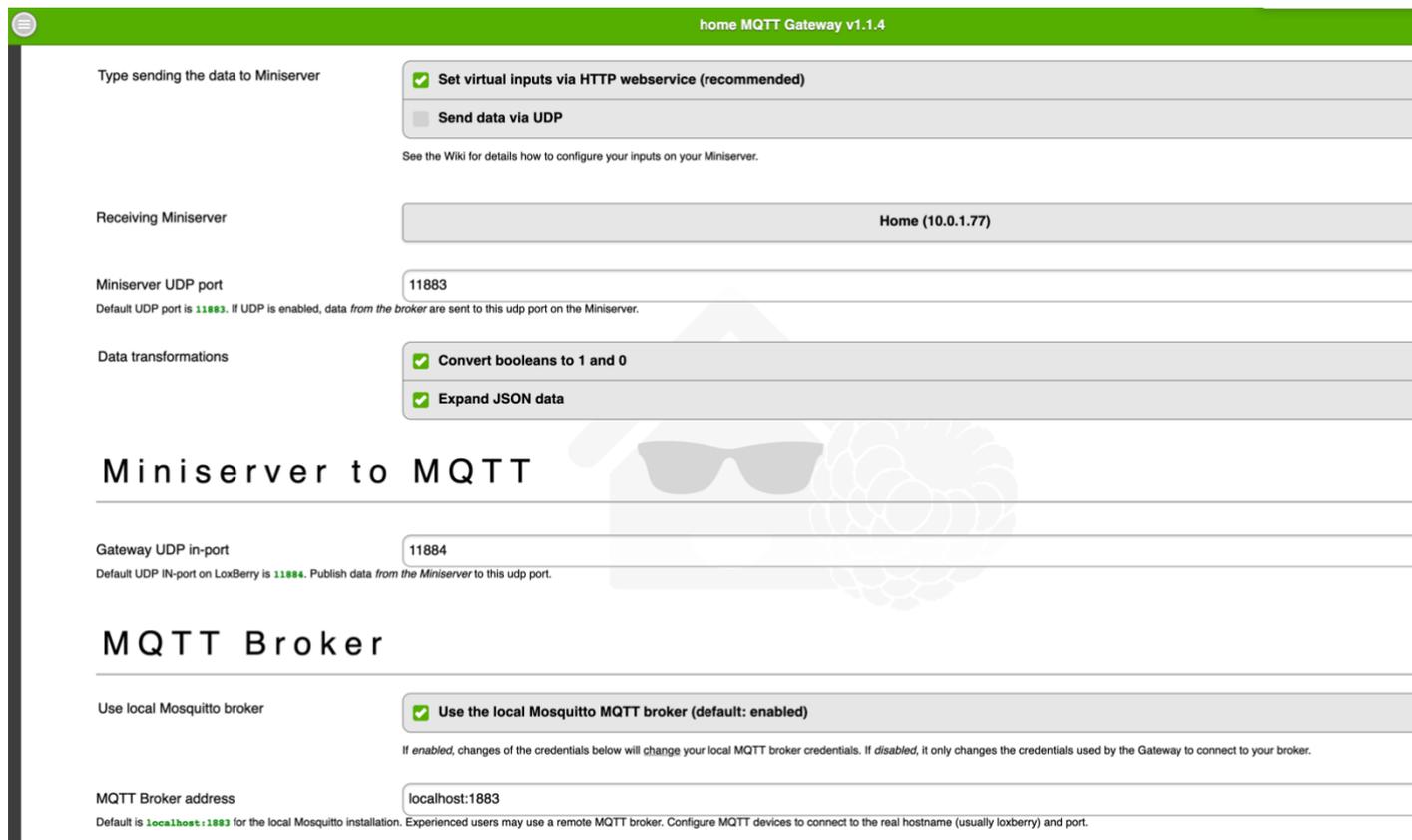
Jetzt kann man das schon mal im browser austesten:

<http://10.0.1.24/ebusd.php?port=8888&mode=read&value=status01>

um das ganze via mqtt verwenden zu können verwende ich loxberry

<https://www.loxwiki.eu/display/LOXBERRY/LoxBerry>

meine einstellungen siehe bilder:



home MQTT Gateway v1.1.4

Type sending the data to Miniserver

- Set virtual inputs via HTTP webservice (recommended)
- Send data via UDP

See the Wiki for details how to configure your inputs on your Miniserver.

Receiving Miniserver

Home (10.0.1.77)

Miniserver UDP port

11883

Default UDP port is 11883. If UDP is enabled, data from the broker are sent to this udp port on the Miniserver.

Data transformations

- Convert booleans to 1 and 0
- Expand JSON data

Miniserver to MQTT

Gateway UDP in-port

11884

Default UDP in-port on LoxBerry is 11884. Publish data from the Miniserver to this udp port.

MQTT Broker

Use local Mosquitto broker

- Use the local Mosquitto MQTT broker (default: enabled)

If enabled, changes of the credentials below will change your local MQTT broker credentials. If disabled, it only changes the credentials used by the Gateway to connect to your broker.

MQTT Broker address

localhost:1883

Default is localhost:1883 for the local Mosquitto installation. Experienced users may use a remote MQTT broker. Configure MQTT devices to connect to the real hostname (usually loxberry) and port.

home MQTT Gateway v1.1.4

Settings Subscriptions Conversions Incoming overview

MQTT Subscriptions

Subscriptions are the topics of MQTT devices that are relayed to the Miniserver. A # sign symbols a joker in MQTT. Use exactly the topic(s) provided by the device vendor.

Advanced feature: Send specific subscriptions to specific Miniservers - after the subscription, use a pipe | symbol followed by a list of Miniserver numbers (from the Miniserver widget), e.g. `shellics/#|2` or `shellics/#|2,3`. Without pipe, the default Miniserver from the page is used.

One subscription topic per line. For Shelly devices, use `shellics/#`

```
shellics/#
nuki/#
ebusd/#
08/#
hex/#
rcvt/#
roomba/#
recov/#
ebusd2/#
```

Subscriptions defined by other plugins

Save and Apply Restart

MQTT Gateway running (PID1877) | Mosquitto running (PID812)

home MQTT Gateway v1.1.4

Settings Subscriptions Conversions Incoming overview

Text-to-Value conversions

MQTT devices often are sending strings that Loxone unfortunately cannot handle. In this field you can create text-to-value conversions. You can assign values to incoming strings, that values are sent to the Miniserver can be evaluated e.g. with a Status block.

Booleans (on, off etc.) are already converted, if you have 'Boolean conversion' enabled.

Text-to-Value conversion

One conversion per line. Use `Text=Value`

```
off=0
an=1
auto=2
load=6
```

Save and Apply Restart

MQTT Gateway running (PID1877) | Mosquitto running (PID812)

home MQTT Gateway v1.1.4

Settings
 Subscriptions
 Conversions
Incoming overview
Logfiles

Last transmissions to Miniserver

This page lists the transmissions via the MQTT Gateway to the Miniserver in the last 24 hours. It is refreshed *automatically on-the-fly* (as long as you have text selected, updating stops).
 You can copy and paste the 'Virtual Input' names for HTTP transmissions, or create command recognitions from the UDP messages.

Show advanced table information

HTTP Virtual Inputs (122 entries)

🔍 ebusd

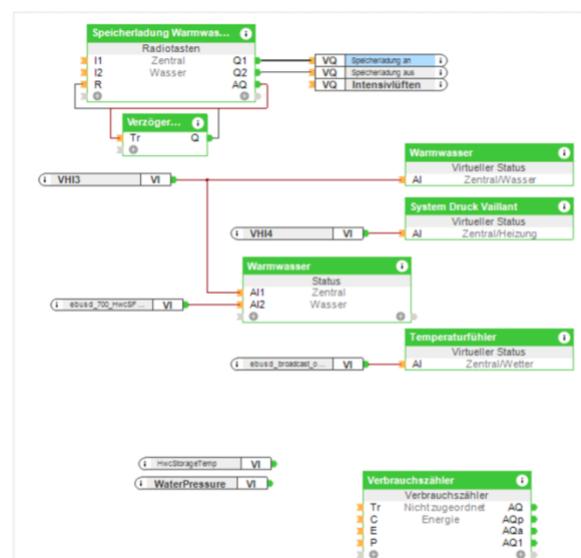
Miniserver Virtual Input	Last value	Last submission
ebusd2_global_running	1	31.10.2020 13:...
ebusd2_global_signal	1	31.10.2020 13:...
ebusd2_global_updatecheck	"revision v3.4 available, broadcast.csv: different version available, vaillant/broadcast.csv: different version available"	31.10.2020 13:...
ebusd2_global_uptime	3696	31.10.2020 14:...
ebusd2_global_version	"ebusd 3.4.v3.3-51-g57eae05"	31.10.2020 13:...
ebusd_700_HwcSFMode_sfmode_value	6	31.10.2020 14:...
ebusd_700_HwcStorageTemp_tempv_value	52.5	31.10.2020 14:...
ebusd_700_HwcTempDesired_tempv_value	60	31.10.2020 14:...
ebusd_700_WaterPressure_pressv_value	1.6	31.10.2020 14:...
ebusd_broadcast_outsidetemp_temp2_value	15.25	31.10.2020 14:...
ebusd_broadcast_vdatetime_date_value	31.10.2020	31.10.2020 14:...
ebusd broadcast vdatetime time value	14:12:02	31.10.2020 14:...

das ganz in loxone einbinden:
 einen virtuellen eingang bzw. ausgang anlegen mit der richtigen ip adresse vom raspi

Eigenschaft	Wert
Allgemein	
Bezeichnung	ebusd set
Beschreibung	
Hinweis-Text	Bearbeiten...
Objekttyp	Virtueller Ausgang
Anschluss	VQ9
Raum	Nicht zugeordnet
Einstellungen	
Adresse	http://10.0.1.24
<input checked="" type="checkbox"/> Verbindung nach Senden sch...	
Trennzeichen	;
Befehl bei Verbindungsaufbau	
Logging/Mail/Call/Track	

den passenden befehl erstellen:
in meinem fall die speicherladung warmwasser
dasselbe prinzip auch bei der lüftung anwenden (port 8889)

Eigenschaft	Wert
Allgemein	
Hinweis-Text	Bearbeiten...
Objekttyp	Ausgangsreferenz
Objektfarbe	<input type="checkbox"/> eeeee
Einstellungen	
Ausgang	Speicherladung an (Home)
Referenziertes Objekt	
Allgemein	
Bezeichnung	Speicherladung an
Beschreibung	
Hinweis-Text	Bearbeiten...
Objekttyp	Virtueller Ausgang Befehl
<input type="checkbox"/> Statistik	
Kategorie	Nicht zugeordnet
Raum	Nicht zugeordnet
Visualisierung	
<input type="checkbox"/> Verwenden	
<input type="checkbox"/> Visualisierungskennwort	
Bewertung	★★★★★★★★
Berechtigungen	
Berechtigungen verwalten	Bearbeiten...
Berechtigte Benutzer / Gruppen	Bearbeiten...
Einstellungen	
Befehl bei EIN	/ebusd.php?port=8888&mode=write700&value=hwcSFMode&set=0v
HTTP-Erweiterung bei EIN	
HTTP-Post-Befehl bei EIN	
HTTP Methode bei EIN	GET
Befehl bei AUS	
HTTP-Erweiterung bei AUS	
HTTP-Post-Befehl bei AUS	
HTTP Methode bei AUS	GET
HTTP-Antwort speichern	
Erste Wiederholung [s]	0
Abstand Wiederholung [s]	0
<input checked="" type="checkbox"/> Als Diagnostics verwenden	



Der baustein sieht bei mir so aus:

Status bearbeiten

I	V1	Wert	I	V2	Wert	I	V3	Wert	I	V4	Wert	Symbol	Status text	Stat	
AI2		6	-	==	0	-	==	0	-	==	0	🔥	Heiß	<v1.1> °C	0
AI1	>	50	-	==	0	-	==	0	-	==	0	🟢	Voll	<v1.1> °C	0
AI1	<=	50	AI1	>=	25	-	==	0	-	==	0	🟡	Mitte	<v1.1> °C	0
AI1	<	25	AI1	!=	0	-	==	0	-	==	0	🔴	Leer	<v1.1> °C	0
AI1	==	0	-	==	0	-	==	0	-	==	0	⚠️	Warnung	<v1.1> °C	0
-	==	0	-	==	0	-	==	0	-	==	0		Nicht zugeordnet		0

Sobald eine Bedingung zutrifft, wird sie verwendet (Reihenfolge der Texte ist wichtig). Wenn keine Bedingung angegeben ist, dann trifft sie immer zu, und sollte folglich ganz unten sein. Alle Bedingungen sind UND-verknüpft.

Zahl-Vergleiche: == (gleich), > (größer), >= (größer gleich), < (kleiner), <= (kleiner gleich), != (ungleich)
 Text-Vergleiche: == (gleich), != (ungleich), *= (enthält), !* (enthält nicht), := (beginnt mit), !: (beginnt nicht mit)

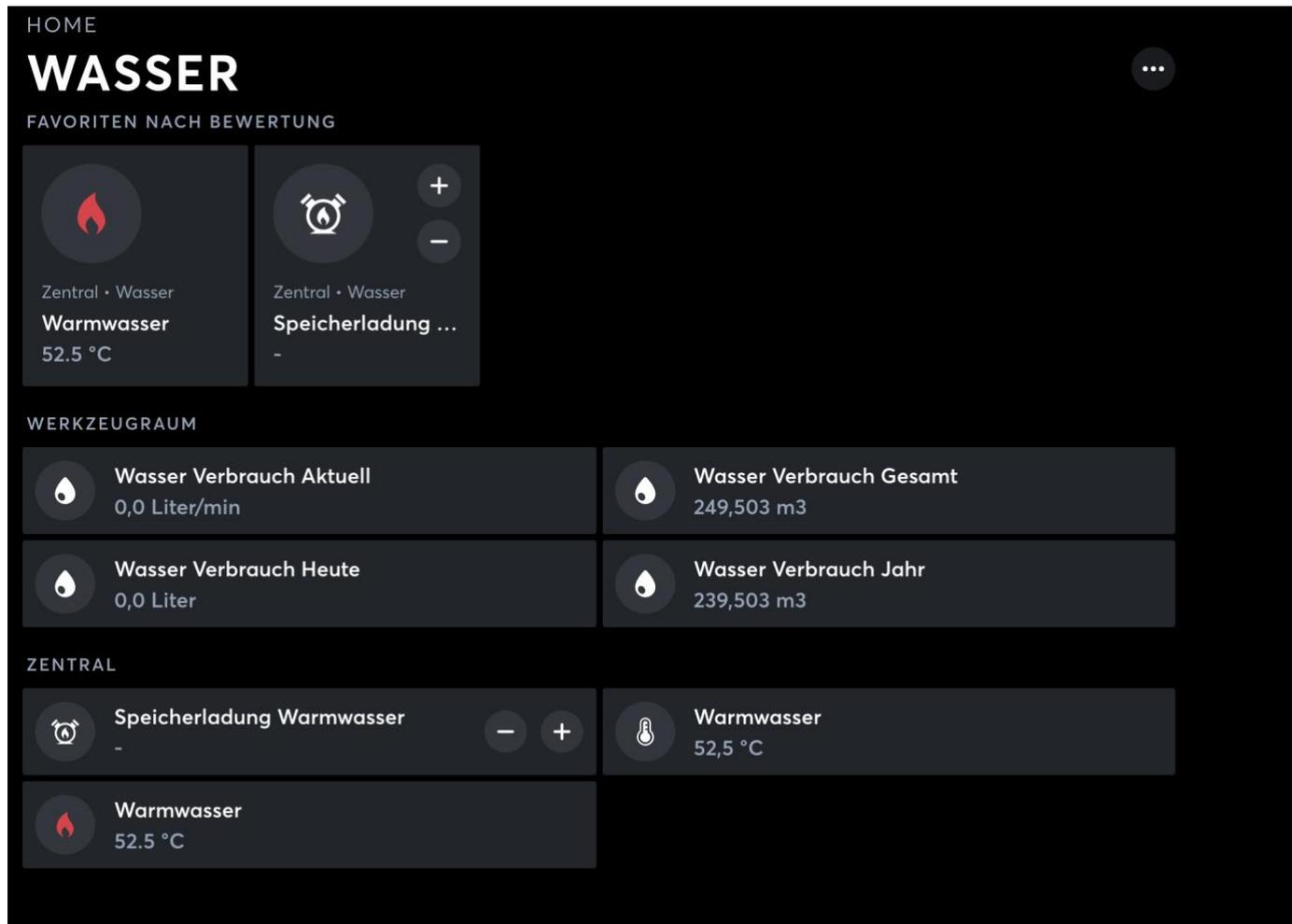
Textfelder: <v1> bis <v4> entsprechen den Werten von AI1 bis AI4
 <v1> = Wert AI1 ohne Nachkommastellen oder Text
 <v1.2> = Wert AI1 mit 2 Nachkommastellen
 <v2.2*100> = Wert AI2 mit 2 Nachkommastellen mal 100
 <v2.2/100> = Wert AI2 mit 2 Nachkommastellen durch 100
 <vn> = eigener Objektname
 <v1.t> = Zeitpunkt der Änderung von AI1, <v1.d> = EIB Datum AI1, <v1.m> = EIB Monat AI1
 Fehlerhafte Texte werden rot angezeigt!

Jalousie-Beispiel laden Werte-Beispiel laden Text-Beispiel laden

OK

Abbr

Loxone visualisierung:



Die grafische umsetzung bei der lüftung habe ich noch nicht umgesetzt.
Wird nachgereicht ... 😊

ps:

sämtliche geräte im netzwerk müssen immer eine fixe adresse zugewiesen bekommen.
Am besten die ip adresse im router über die mac adresse fix zuweisen.

viel spaß bei der umstzung!!

**unbedingt ein backup der sd karte machen sobald das system am laufen ist.
mir ist eine nach einem stromausfall schon kaputt gegangen. 😞**