

# HomeHub 4.1

Stand: April 2024

## Übersicht

1	Einleitung, HomeHub Übersicht.....	2
1.1	Allgemeines.....	2
1.2	Ressourcen.....	3
2	Installation, Update & Backup.....	4
2.1	Voraussetzungen.....	4
2.2	Installation.....	4
2.2.1	Neuinstallation unter Debian.....	4
2.2.2	Neuinstallation auf Synology.....	5
2.3	Update.....	6
2.4	Backup.....	6
3	Konfiguration.....	7
3.1	Grundeinstellung, erster Start, Konfiguration.....	7
3.1.1	Dateien/Verzeichnisse.....	7
3.1.2	Grundeinstellung (config.php).....	7
3.1.3	Erster Start & Geräteimport.....	7
3.2	Navigationsleiste - Kategorien bearbeiten (categories.json).....	8
3.2.1	Home-Funktion.....	9
3.3	Benutzerspezifische Seiten (custom.json).....	10
3.4.2	Homematic Geräte/Variablen einbinden.....	10
3.4.2.1	Indicator.....	12
3.4.2.2	showtime.....	12
3.4.3	ioBroker Integration.....	13
3.4.4	Custom Components.....	16
3.4.4.1	Abfallkalender.....	16
3.4.4.2	Audio.....	17
3.4.4.3	diagramm.....	17
3.4.4.4	ExtLink.....	19
3.4.4.5	Fritzbox und Fritzbox2.....	19
3.4.4.6	Iframe.....	21
3.4.4.7	Meteoblue.....	22
3.4.4.8	Multiview.....	23
3.4.4.9	sysvarpin.....	23
3.4.4.10	Tagesschau.....	24
3.4.4.11	Tankerking.....	24
3.4.4.12	weatherunderground.....	25
3.4.4.13	Sonnenstand.....	25
3.4.4.14	Mondphase.....	27
3.4.4.15	Sonnenaufuntergang.....	29
3.4.4.16	VIR-LG-RGBW-DIM, VIR-LG-WHITE-DIM.....	30
3.4.4.17	Webcam.....	30
3.4.5	Sonstige Customizations.....	30
3.4.5.1	Stylesheets.....	30
3.4.5.2	Angezeigter Text bei Logik-/Werteliste-Variablen durch Icons ersetzen.....	31
3.4.5.3	Icons für Geräte mit Statusmeldungen "state_icons".....	32
3.4.5.4	Einbindung LED-Display HM-OU-LED16.....	32
4	Tips & Tricks.....	34
4.1	Änderungen an custom.json oder categories.json.....	34
4.2	Umlaute & Sonderzeichen.....	34

4.3 ISE-IDs.....	34
4.4 Neue / fehlende Geräte.....	35
.....	35
Anhang.....	36
A1 Changelog.....	36
A2 Krimskrams.....	36

## 1 Einleitung, HomeHub Übersicht

### 1.1 Allgemeines

Homehub ist eine Visualisierung, um HM-Geräte, Variablen, Programme aus einer CCU sowie verschiedener OCCU -Derrivate wie RaspberryMatic, pivCCU, Debmatic anzuzeigen, bzw. zu bedienen.

Homehub wird seit 2015 kontinuierlich weiter entwickelt wird.

Haupt-Augenmerk lag und liegt auf der Einfachheit und Übersichtlichkeit.

Eingebunden werden Geräte, Heizungsgruppen, virtuelle Taster, Systemvariablen und Programme.

Unterstützt werden z.Z. >200 Homematic Geräte (HM, HMW, HmIP, HmIPW) sowie ausgewählte CuxD Geräte.

Mit Homehub 4 wurde erstmalig auch die Möglichkeit geschaffen, Geräte/Objekte aus ioBroker zu nutzen. Fokus war auf den wichtigsten Objekten (Schalter, Taster, Dimmer, Werte/Texte, Programme triggern). Es können jetzt z.B. viele Zigbee-, Tasmota-, native Sonoff/Shelly/Tuja-Geräte oder Info-Adapter mit integriert werden. MQTT i.d.R. nur über den Umweg von Skript oder Node-Red (Ausnahme: Shelly Adapter).

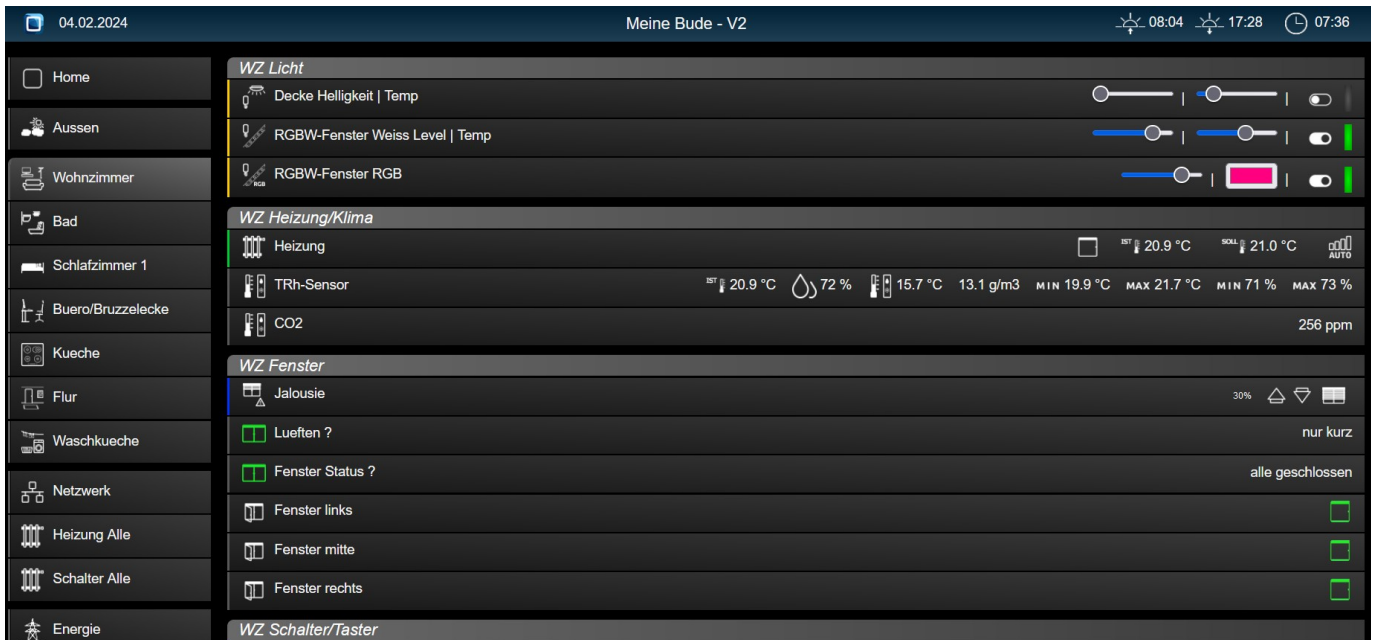
Virtuelle CCU-Jack Geräte werden nicht unterstützt (Einbindung ggfls. über den Umweg virtueller Taster/Systemvariablen oder CuxD exec Schalter/Taster).

Dieses Dokument soll helfen, den Einstieg zu erleichtern - insbesondere auch für jene, bei denen z.B. PHP-Programmierung, json-Syntax, o.ä. nicht Bestandteil der Muttermilch waren (so wie bei mir).

Eine eigenes, funktionales Homehub kann aber mit recht wenig Aufwand und etwas Achtsamkeit bei der Syntax in den Konfigurations-Dateien schnell erreicht werden.

Für Fragen, empfiehlt sich auch das Homematic-Forum und hier insbesondere die Haupt-Threads zu der jeweils aktuellen Version, z.Z: <https://homematic-forum.de/forum/viewtopic.php?f=41&t=79208>

Bilder



## 1.2 Ressourcen

Für den Einstieg empfehlen sich verschiedenste Quellen. Dort finden sich auch Beispiele, sowie viele nützliche Infos. Vereinzelte Info's Lösungen sind nicht mehr ganz up-to-date, deshalb u.U. von hinten nach vorne durcharbeiten.

Dateien innerhalb des Homehub-ZIPs – hier finden sich verschiedene Hinweise zur Verwendung, insbesondere in den PHP-Dateien der componenten in /homehub/custom/componenten

HM-Forum Threads:

- Version 4.0 <https://homematic-forum.de/forum/viewtopic.php?f=41&t=79208>
- Version 3.3 <https://homematic-forum.de/forum/viewtopic.php?f=41&t=76034>
- Version 3.2 <https://homematic-forum.de/forum/viewtopic.php?f=41&t=66530#p653085>
- Version 3.11 <https://homematic-forum.de/forum/viewtopic.php?f=41&t=50538&hilit=homehub>
- Version 2.02 <https://homematic-forum.de/forum/viewtopic.php?f=41&t=27975&hilit=homehub#p248217>
- Version „1“ <https://homematic-forum.de/forum/viewtopic.php?f=18&t=25840&hilit=homehub>

Sonstige

- V3.1 Video von Gerti [https://www.youtube.com/watch?v=CLq\\_ubm33pM](https://www.youtube.com/watch?v=CLq_ubm33pM)
- V3.1 Video verdrahtet <https://www.youtube.com/watch?v=8njvl4UDhF0>  
incl. Installation auf Synology Diskstation
- V3.0 Homepage von Gerti <https://github.com/Gerti1972/homehub>
- V1 Github Seite von braindead <https://github.com/braindead1/HomeHub-WebUI>

## Icons

Eine Übersicht der Icons kann aufgerufen werden mit - <http://IPdesWebservers/homehub/icon/index.php>

Github Seite: <https://github.com/OpenAutomationProject/knx-uf-iconse> (CC BY-SA 3.0)

## 2 Installation, Update & Backup

Mit Version 4.1 wurde der Zugriff auf die CCU von XML-API auf Homematic Script umgestellt, was ein Geschwindigkeitsvorteil bringt.

### 2.1 Voraussetzungen

Für die Verwendung von Homehub gelten folgende Voraussetzungen:

1. Webserver mit mindestens PHP 8.2 (aktuell 8.3)
2. Module php-xml (mit simplexml) und php-curl müssen installiert und aktiviert sein.
3. ~~installierte XML-API 1.2 oder XML-API 2\* auf der CCU.~~  
\* ~~Token erforderlich, siehe 2.2 Punkt 4~~ **nicht mehr erforderlich für Homehub**
4. ioBroker adapter simpleapi (<https://www.iobroker.net/#de/adapters/adapterref/iobroker.simple-api/README.md>), falls auch Geräte/Objekte aus ioBroker genutzt werden sollen

### 2.2 Installation

1. das Verzeichnis "Homehub" aus der ZIP entpacken und z.B. über WINSCP in das Basisverzeichnis des Webserver kopieren.
2. Den Ordnern "config" und "cache" Schreibberechtigungen erteilen (777).
3. im Verzeichnis [/VerzeichnisDesWebserver/homehub/config/](#) folgende Dateien umbenennen:
  - "categories.template.php" in "categories.php"
  - "config.template.php" in "config.php"
  - "custom.template.php" in "custom.php"
4. In der Datei "config.php" dann folgende Variablen anpassen, siehe 3.1.2
5. Aufruf von Homehub dann mit [IpdesWebserver/homehub/](#)
6. Geräteimport & Konfiguration der Menüleiste/Seiten, siehe Abschnitte 3.3 und 3.4

#### 2.2.1 Neuinstallation unter Debian

Diese Anleitung ist beispielhaft für Debian 11 (Bullseye), 12 (Bookworm) oder entsprechender RaspberryPi OS-Versionen.

Als Webserver wird hier Apache 2 samt Modulen verwendet.

Für andere Distributionen und/oder Webserver wird auf die entsprechende Dokumentation verwiesen.

##### 1. WebServer & PHP installieren.

```
sudo wget -qO /etc/apt/trusted.gpg.d/php.gpg https://packages.sury.org/php/apt.gpg
sudo echo "deb https://packages.sury.org/php/ $(lsb_release -sc) main" | sudo tee /etc/apt/sources.list.d/php.list
```

```
sudo apt update  
sudo apt install php
```

# Apache2 wird automatisch mitinstalliert, falls noch nicht vorhanden

Kontrolle der PHP-Version, sollte  $\geq 8.2$  anzeigen

```
php -v
```

## 2. Module installieren & Neustart

```
sudo apt install php-curl php-xml  
sudo reboot
```

Kontrolle installierter Module, curl und simplexml müssen in der Liste auftauchen

```
php -m
```

Erweiterte Kontrolle/Konfigurationsdetails

im Basisverzeichnis des Apache2-Webservers (</var/www/html/>) eine Datei z.B. "info.php" mit folgendem Inhalt erstellen

```
<?PHP  
    echo phpinfo();  
?>
```

Aufruf mit [IpdesWebservers/info.php](http://ipdesWebservers/info.php)

**Hinweis:** falls der Webserver unter der gleichen IP wie die CCU installiert wird, muss der Webserver-Port geändert werden, da CCU und Webserver sonst gleichzeitig Port 80 nutzen wollen. Vorgehen:

1. in 'ports.conf' unter 'Listen 80' neuen (freien) Port setzen, z.B. in 90 -> speichern

```
sudo nano /etc/apache2/ports.conf
```

2. in '000-default.conf' unter 'VirtualHost: \*:80' auf neue Portnummer anpassen

```
sudo nano /etc/apache2/sites-enabled/000-default.conf
```

3. Rechner oder Apache2 neu starten

```
sudo systemctl restart apache2 #Systemd  
sudo service apache2 restart #SysVInit
```

4. Aufruf HomeHub erfolgt dann immer mit [IpdesWebservers/homehub:neuerPort/homehub](http://ipdesWebservers/homehub:neuerPort/homehub)

## 2.2.2 Neuinstallation auf Synology

WIP

Eine aktuelle Beispielkonfiguration hier <https://homematic-forum.de/forum/viewtopic.php?f=41&t=79208&start=110#p789851>

Eine grundsätzliche Beschreibung ist hier zu finden:

[https://kb.synology.com/de-de/DSM/tutorial/How\\_to\\_host\\_a\\_website\\_on\\_Synology\\_NAS](https://kb.synology.com/de-de/DSM/tutorial/How_to_host_a_website_on_Synology_NAS)

Ein etwas älteres Video von verdrahtet <https://www.youtube.com/watch?v=8njvl4UDhF0> ist schon 3 Jahre alt aber es beschreibt die grundsätzliche Vorgehensweise recht gut.

## 2.3 Update

Ein update gestaltet sich 'straight forward' und besteht aus 4 Schritten:

1. update.txt in der ZIP Datei lesen und Hinweise beachten
2. Vor dem update mindestens die folgenden Dateien aus dem Verzeichnis /homehub/config/ sichern
  - categories.json
  - custom.json
  - config.php
  - export.json

hat man darüber hinaus selber noch andere Dateien geändert (z.B. die mapping.json), muß man für den Abgleich mit einer evtl. modifizierten Version im Update selber Hand anlegen.

3. Neue Homehub-ZIP entpacken und über die alte Installation drüber kopieren.
4. Gesicherte Dateien aus Punkt 2 zurückspielen

## 2.4 Backup

Ein Backup erfolgt ganz simpel, indem man das komplette /homehub/ Verzeichnis an einen Ort seiner Wahl kopiert.

## 3 Konfiguration

### 3.1 Grundeinstellung, erster Start, Konfiguration

#### 3.1.1 Dateien/Verzeichnisse

WIP

#### 3.1.2 Grundeinstellung (config.php)

Die globalen Basis-Parameter für Homehub werden in der Datei /config/config.php vorgenommen.

\$homematicIp = 'IpderCCU' oder 'DNS Name der CCU' (mandatory)

\$title = 'beliebiger Name' (optional) – wird in der GUI Oben angezeigt

\$latitude und \$longitude (optional) – für Bestimmung des Sonnenauf-/ untergangs

\$navi\_fix = true/false (optional) – einfrieren der Menüleiste links, nur sinnvoll, wenn man nicht mehr Menüeinträge hat, als auf der Seite angezeigt werden kann.

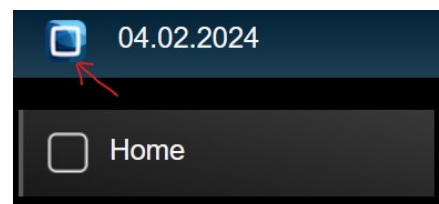
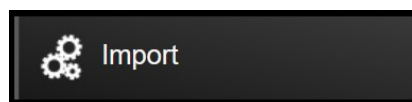
\$viewport = 'device-width' oder '#pixel' (optional)

\$responsive = true/false bei true muss \$viewport auf 'device-width' stehen

#### 3.1.3 Erster Start & Geräteimport

Nach dem ersten Start müssen zuerst die Geräte, Programme und Variablen aus der CCU importiert werden.

1. Dazu wählt man entweder in der Navigationsleiste ganz unten "Import" oder macht einen Doppelklick auf das Homematic-Symbol in der Kopfzeile oben links.



[Dann einen Moment warten](#)

3. In der Folge sollten folgende Kategorien auftauchen. Diese sind schon automatisch befüllt. Diese Zuordnung wird über die /homehub/config/mapping.json gesteuert





Hinweis: mindestens ein Zentralenprogramm auf der CCU muß vorhanden sein. Ggfls. Ist ein "Dummy"-Programm zu erstellen. Dieses benötigt keine DANN-Bedingung.

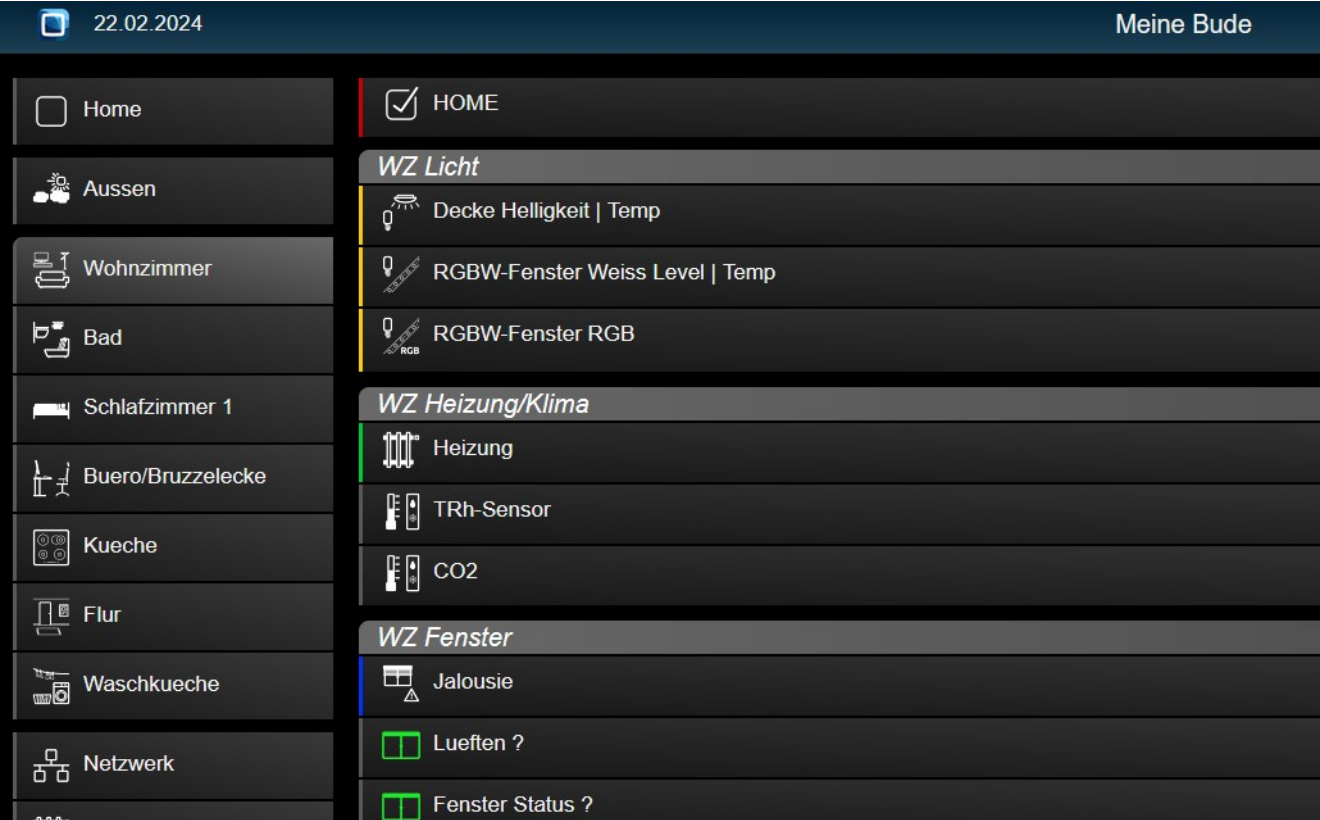
### 3.2 Navigationsleiste - Kategorien bearbeiten (categories.json)

Die Navigationsleiste wird in der Datei /config/categories.json konfiguriert. Es können Unterkategorien definiert werden.

Bei der Namensvergabe sollte darauf geachtet werden, dass man nicht schon existierende Namen von Geräten, CCU- Variablen, Default-Kategorien (wie unter 3.1.3 gelistet) verwendet.  
Ggfls. kann z.B. ein Leerzeichen an das Ende gesetzt werden.

```
{
  "categories":[
    {
      "name":"Home",
      "icon":"control_home.png",
      "subcategories":[],
      "append_divider":"true"
    },
    {
      "name":"Aussen",
      "icon":"weather_cloudy_light.png",
      "subcategories":["Wetterinfo", "GA Sonstiges"],
      "append_divider":"true"
    },
    {
      "name":"Wohnzimmer",
      "icon":"scene_livingroom.png",
      "subcategories":["WZ Licht", "WZ Heizung/Klima", "WZ Fenster", "WZ Schalter/Taster", "WZ Sonstiges"]
    },
    :
    :
    :
    {
      "name":"Programme",
      "icon":"audio_play.png"
    },
    {
      "name":"Systemvariablen",
      "icon":"control_clear.png",
      "append_divider":"true"
    },
    {
      "name":"Import",
```

```
"icon": "edit_settings.png",
"append_divider": "true"
}
}
```



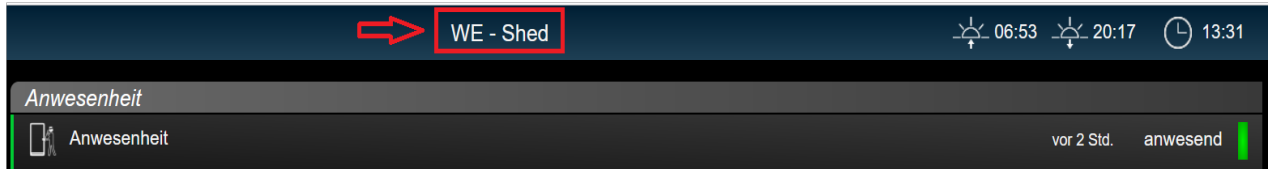
Parameter für categories.json

Parameter	Inhalt/Zustände	Anwendung & Hinweise
"name"	"beliebig"	Angezeigter Name im Menü
"icon"	"name der png-Datei"	Siehe Abschnitt 1.2
"subcategories"	"beliebig"	
"visible"	"true   false"	Kategorie wird angezeigt Ja/Nein
"append_divider"	"true   false"	Schmalere Spacer unter diesem Eintrag

**Hinweis:** Editor mit UTF-8 encoding verwenden

3.2.1 Home-Funktion

Mit Hoemhub 4.1 wurde auch eine neue „Home“-Funktion eingeführt. Diese ist standardmäßig aktiv. Dazu braucht man jetzt nur den eigenen Namen seines Homehub’s anwählen. Eine Kategorie „Home“ muß existieren. Diese wird beim ersten Start automatisch miterzeugt. Hat man diese gelöscht oder den Namen geändert, wird eine leere Seite angezeigt.



### 3.3 Benutzerspezifische Seiten (custom.json)

Die in der categories.json definierten Seiten können individuell in der /config/custom.json konfiguriert werden. Geräte, Variablen, etc. könne so beliebig oft auf der gewünschten Seite gelistet werden. Die Namen der CCU Geräte/Variablen kann man ja gut den Default-Kategorien entnehmen oder aus der xml-api.

Für IoBroker basierte Objekte oder custom components siehe Abschnitte 3.4.3 und 3.4.4ff

Generell gilt auch hier: Bei der Namensvergabe sollte darauf geachtet werden, dass man nicht schon existierende Namen von Geräten, CCU-Variablen, Default-Kategorien (wie unter 3.1.3 gelistet) verwendet. Ggfls. kann z.B. ein Leerzeichen an das Ende gesetzt werden.

**Hinweis:** Editor mit UTF-8 encoding verwenden

### 3.4.2 Homematic Geräte/Variablen einbinden

**Hinweis:** Geräte/Variablen/etc. der CCU werden über ihren Namen und nicht über ihre ID eingebunden. Falls man also Namen in der CCU ändern sollte, kann das auch die Notwendigkeit einer entsprechenden Anpassung in Homehub nach sich ziehen.

HM Geräte/Variablen in die eigenen categories einbinden

```
"WZ Heizung/Klima":[
  {
    "name":"WZ-Heizgruppe:1",
    "icon":"sani_heating.png",
    "display_name":"Heizung"
  },

  {
    "name":"CUxD-WZ-Regler:1",
    "icon":"temp_dew_point.png",
    "display_name":"TRh-Sensor"
  },

  {
    "name":"CO2",
    "icon":"temp_dew_point.png",
    "display_name":"CO2"
  }
],

"WZ Fenster":[
  {
    "name":"WZ-Jalousie Bedienung",
    "icon":"fts_shutter_attention.png",
    "display_name":"Jalousie"
  },

  {
    "name":"Lueften Test",
    "icon":"fts_window_2w_gn.png",
    "display_name":"Lueften ?"
  }
]
```

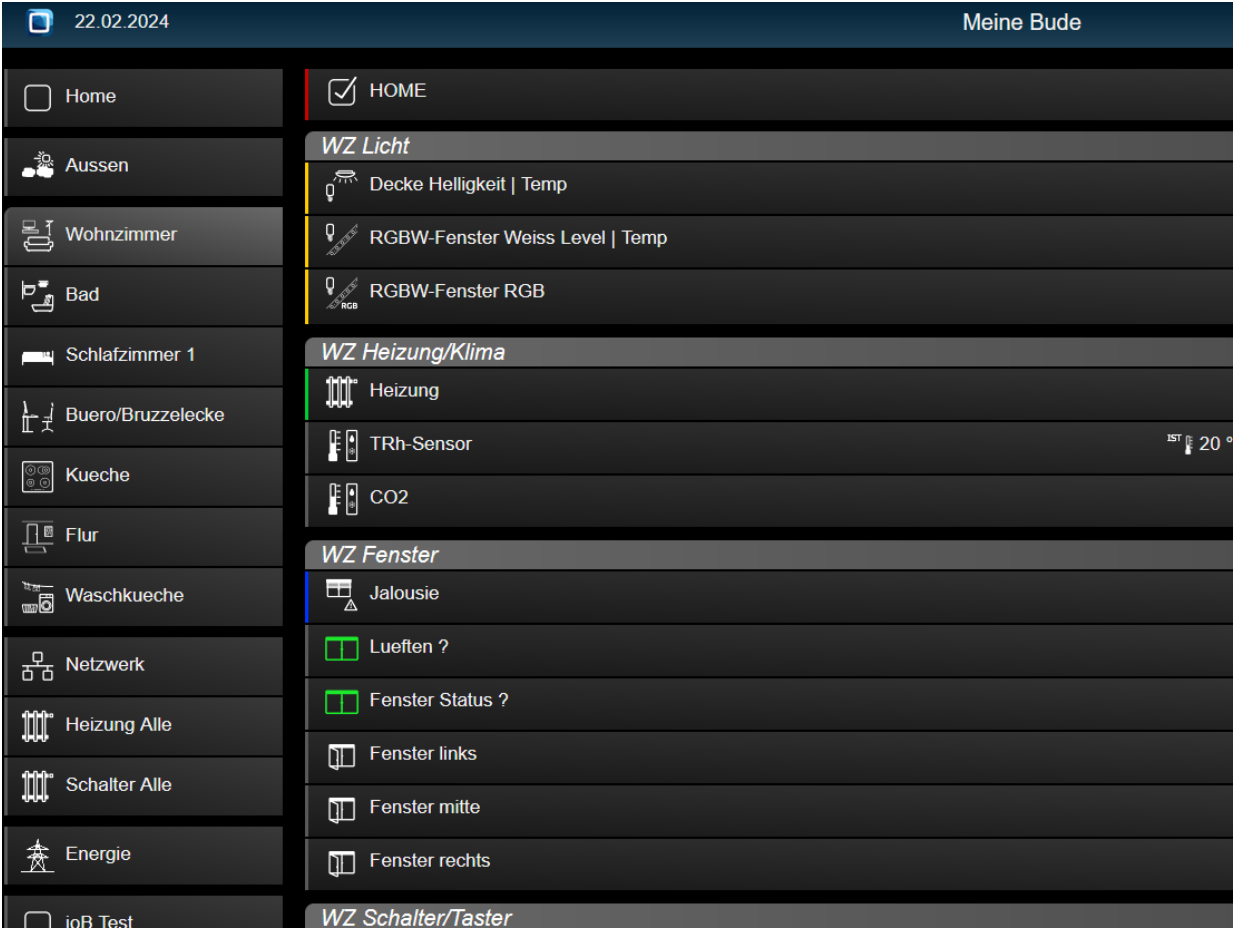
```
    },

    {
      "name": "Status Fenster",
      "icon": "fts_window_2w_gn.png",
      "display_name": "Fenster Status ?",
    },

    {
      "name": "WZ-Fenster-links:1",
      "icon": "fts_window_2w_open_l.png",
      "display_name": "Fenster links"
    },

    {
      "name": "WZ-Fenster-mitte:1",
      "icon": "fts_window_2w_open_l.png",
      "display_name": "Fenster mitte"
    },

    {
      "name": "WZ-Fenster-rechts:1",
      "icon": "fts_window_2w_open_l.png",
      "display_name": "Fenster rechts"
    }
  ],
```



Parameter für config.json

Parameter	Inhalt/Zustände	Anwendung & Hinweise
"name"	"Name in CCU"	EXAKTER Name des Gerätekanals, Variable
"icon"	"name der png-Datei"	Siehe Abschnitt 1.2
"display_name"	"beliebig"	Angezeigter Name
"append_divider"	"true   false"	Schmalere Spacer unter diesem Eintrag
"operate"	"true   false"	Steuern der Bedienbarkeit von CCU Variablen,

		default=true
"indicator"	siehe 3.4.2.1	Siehe 3.4.2.1
"color"	"HEX-Wert"	Farbe des Farbindikatorks links in der Zeile. Experte: in der PHP Datei von dem Gerätetyp den "color" Wert ändern, wirkt global!
"Label"	"beliebig"	Nur für CCU-Programme, ändert Bezeichnung des Buttons
"showtime"	"true   false"	Nur bestimmte Geräte, siehe 3.4.2.2

### 3.4.2.1 Indicator

Mit "indicator" kann die Farbe des Status-Indikators beeinflusst werden. Die Farbzuoordnung ist:

false: Aus (grau)  
true: Ein (grün)  
warn: Warnung (gelb)  
alarm: Alarm (rot)

Verwendung ..

1. für CCU-Variable des Typs Logik oder Werteliste

Beispiel:

„indicator“: "1,true;0,warn;3,alarm" – CCU-Werteliste mit 5 Werten,  
Wert 2 und 5 werden hier automatisch auf false (=grau) gesetzt.

2. Für CCU-Variable des Typs Zahl

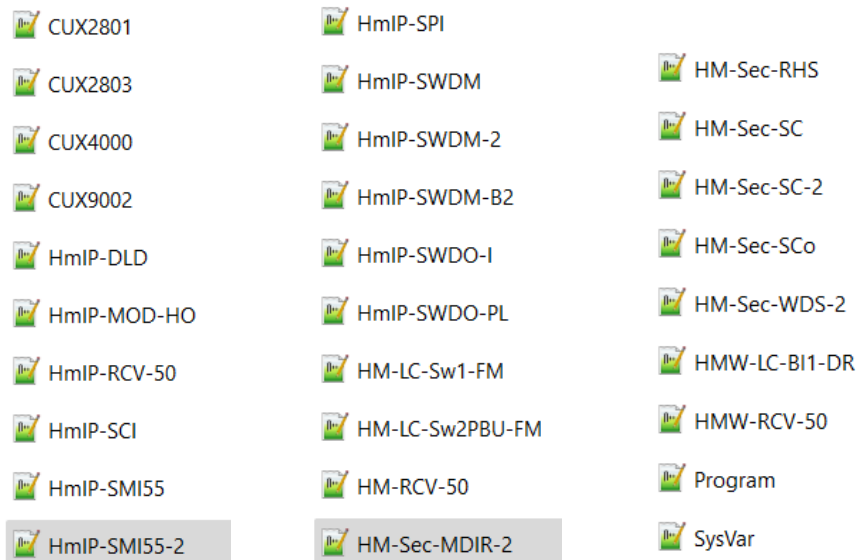
Beispiel:

"indicator": "40,true;60,warn;100,alarm" – Vergleichsoperanden werden nicht gebraucht, Hier wird bei  
<=40 grün, bei <=60 gelb und <=100 rot angezeigt.



### 3.4.2.2 showtime

Mit "showtime" kann man für ausgewählte Geräte sowie Systemvariablen und CCU Programme den verkürzten Timestamp der letzten Änderung anzeigen lassen. Z.Z. unterstützt werden:



### Einbindung in custom.json

```
{
  "name": "Anwesenheit",
  "icon": "secur_burglary.png",
  "display_name": "Anwesenheit",
  "indicator": "0,false;1,true;2,warn;3,alarm",
  "color": "#00CC33",
  "showtime": "true"
}
```



### 3.4.3 ioBroker Integration


Im folgenden wird beschrieben, wie man Objekte aus ioBroker über die custom.json in Homehub integrieren kann.

Die Vorgehensweise ist grundsätzlich identisch zur Einbindung von Homematic Geräten, man muss aber beachten, dass verschiedene Parameter hier nicht verfügbar sind, bzw. etwas anders verwendet werden (speziell: "name", "color")

Vorraussetzung ist ein aktivierter SimpleAPI-Adapter in ioBroker.

Die Standardeinstellungen können belassen werden.

(<https://www.iobroker.net/#de/adapters/adapterref/iobroker.simple-api/README.md>)


Instanzeinstellungen: simple-api.0
v2.7.2
info

ALLGEMEINE EINSTELLUNGEN
LET'S ENCRYPT SSL

Als Web-Adapter-Erweiterung nutzen
kein

IP

[IPv4] 0.0.0.0 - Zugriff von allen IPs zulassen

Port

8087

☐ Verschlüsselung (HTTPS)

☐ Authentifizierung

Ausführen als Benutzer
admin

☐ Nur erlauben wenn Benutzer auch Besitzer ist

Datenquelle auswählen

kein

## Parameter für config.json

Parameter	Inhalt/Zustände	Anwendung & Hinweise
"component"	"ioBroker"	fix
"api"	"http://IpdesioBrokers:8087"	Port lt. config des SimpleAPI-Adapters, Standard 8087
"name"	"beliebig"	Achtung bei Sonderzeichen
"icon"	"NameIconDatei.png"	Verzeichnis /homehub/icon/
"objekt"	"ObjektIDioBroker"	ID aus ioBroker Objektbaum
"modus"	"toggle", "program", "text", "dimmer", "color"	Siehe Beispiele
"unit"	1. "beliebig"; 2. "Zahl"	1. für Parameter = "text" ← zeigt Text, z.B. Einheit 2. für Parameter = "dimmer" ← max. Wert in ioB*
"label"	"beliebig"	Nur für "program", ändert Bezeichnung des Buttons
„showtime“	„true   false“	Zeigt letzten Timestamp des Objektes. Gültig für toggle, program, text, dimmer, color, siehe Beispiel 3

\* für z.B. Farbtemperatur kann der notwendige Wertebereich, um den vollen Einstellbereich verwenden zu können, nicht wie sonst 0-100, sondern z.B. 0-550 sein.

## Beispiele

Modus = "toggle": Zigbee Schalter

```
{
  "component": "ioBroker",
  "api": "http://192.168.178.66:8087",
  "name": "ZB#7 Wohnzimmer",
  "icon": "message_socket.png",
  "objekt": "zigbee.0.7cb03eaa0a09311e.state",
  "modus": "toggle"
}
```



ZB#7 Wohnzimmer



Modus = "program": Zigbee Taster

```
{
  "component": "ioBroker",
  "api": "http://192.168.178.66:8087",
  "name": "Xiaomi Zigbee Switch left",
  "icon": "control_central_arrow_right.png",
  "objekt": "zigbee.0.90fd9ffffdae70c.arrow_left_click",
  "modus": "program",
  "label": "Klick"
}
```



Xiaomi Zigbee Switch left

Klick

Modus = "text": Linux-Adapter Abfrage is-online (zeigt Ergebnis der letzten Abfrage mit Zeitpunkt)

```
{
  "component": "ioBroker",
  "api": "http://192.168.178.66:8087",
  "name": "linux-control.0.DatschaServer.info.is_online",
  "icon": "cyan_dot.png",
  "objekt": "linux-control.0.DatschaServer.info.is_online",
  "modus": "text",
  "color": "#30D5C8",
  "showtime": "true"
}
```



Reboot

Reboot

linux-control.0.DatschaServer.info.is\_online

vor 3 Min. true

Modus = "text": Linux-Adapter Batterie Ladezustand

```
{
  "component": "ioBroker",
  "api": "http://192.168.178.66:8087",
  "name": "Batterie",
  "icon": "measure_battery_75.png",
  "objekt": "linux-control.0.HomeServer.Batterie",
  "modus": "text",
  "unit": "%"
}
```



Batterie

100 %

Modus = "text": Shelly pro 3EM Spannung

```
{
  "component": "ioBroker",
  "api": "http://192.168.178.66:8087",
  "name": "ioB Uservariable (Zahl)",
  "icon": "measure_current.png",
  "objekt": "0_userdata.0.Shelly_3EM.L1_Spannung",
  "modus": "text",
  "unit": "V"
}
```



3EM Spannung

231.77 V



Modus = "color" (kombiniert): RGB-LED-Strip

```
{
  "component": "ioBroker",
  "api": "http://192.168.178.66:8087",
  "name": "RGBW-Fenster RGB",
  "icon": "light_led_stripe_rgb.png",
  "objekt": "tuya.0.bfc82de5df4eba95a0uswb.22,tuya.0.bfc82de5df4eba95a0uswb.24-rgb,tuya.0.bfc82de5df4eba95a0uswb.20",
  "modus": "dimmer,color,toggle"
}
```



Modus = "dimmer" (kombiniert): Zigbee-Lampe kalt-/warmweiss (Einstellbereich 0-550)

```
{
  "component": "ioBroker",
  "api": "http://192.168.178.66:8087",
  "name": "Decke Helligkeit | Temp",
  "icon": "light_ceiling_light.png",
  "objekt": "zigbee.0.d0cf5efffe28381c.brightness,zigbee.0.d0cf5efffe28381c.colortemp,zigbee.0.d0cf5efffe28381c.state",
  "modus": "dimmer,dimmer,toggle",
  "unit": "°",
  "550",
}
```



### 3.4.4 Custom Components

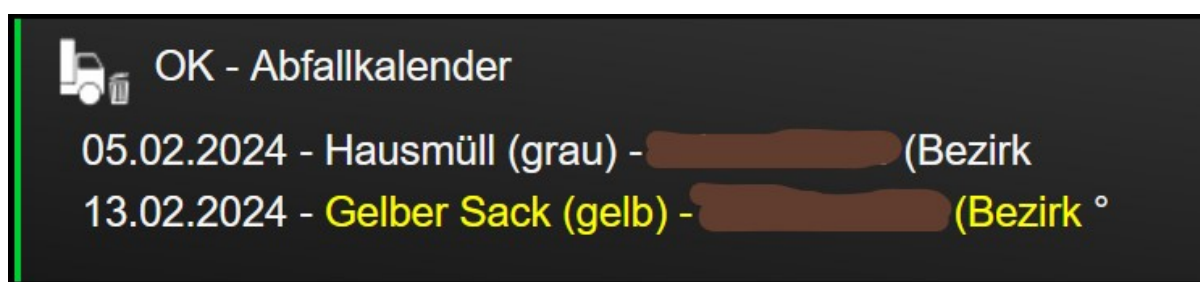
Die verfügbaren Custom components findet man im Verzeichnis /custom/components/

Gemeinsame Parameter

Parameter	Inhalt	Hinweise
"component"	"name"	Name der Komponente, case-sensitiv
"name"	"beliebig"	Wählbarer Text
"icon"	"name_des-icons"	Frei wählbar, siehe /homehub/icon/ oder IPdesWebservers/homehub/icon/index.php

#### 3.4.4.1 Abfallkalender

Vorschau der Müllabfuhrtermine aus einer ics-Datei. Wenn die Termine veraltet sind (Vorjahr), wird ein Warnhinweis ausgegeben.



Die .ics-Datei des lokalen Versorgers unter den Namen "abfallkalender.ics" im Verzeichnis /homehub/custom/components ablegen.

Zum testen kann auch die schon vorhandene Beispiel-.ics genutzt werden

Einbinden in der custom.json

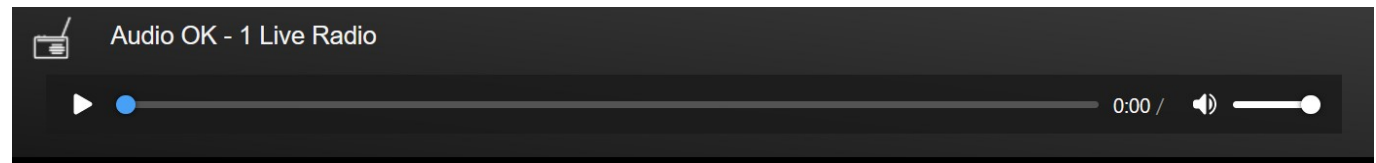
```
{
  "component": "abfallkalender",
  "name": "Abfallkalender",
  "icon": "message_garbage_collection.png",
  "tage": "10",
  "color": "#00CC33"
}
```

Parameter	Inhalt	Hinweise
"tage"	"Zahl"	Anzahl Tage für Vorschau

Hinweis: Sollten nach upload der eigenen ics-Datei noch Einträge aus der vorinstallierten abfallkalender.ics erscheinen, mal die Datei abfallkalender.cache im Verzeichnis /VerzeichnisDesWebservers/homehub/cache löschen und Cache löschen (strg+F5).

3.4.4.2 Audio

Einfach Abspielfunktion für Webradio-Streams im mp3-Format. Das abspielen stoppt, wenn man auf eine andere Seite wechselt



Für verschiedene Sender hier zwei beispielhafte Such-Seiten  
<https://rrradio.de/radiosender/>  
<http://fmstream.org/index.php?c=FT>

Einbinden in der custom.json

```
{
  "component": "Audio",
  "name": "Radio",
  "icon": "it_radio.png",
}
```

```
"file": "https://wdr-1live-live.icecastssl.wdr.de/wdr/1live/live/mp3/128/stream.mp3"
}
```

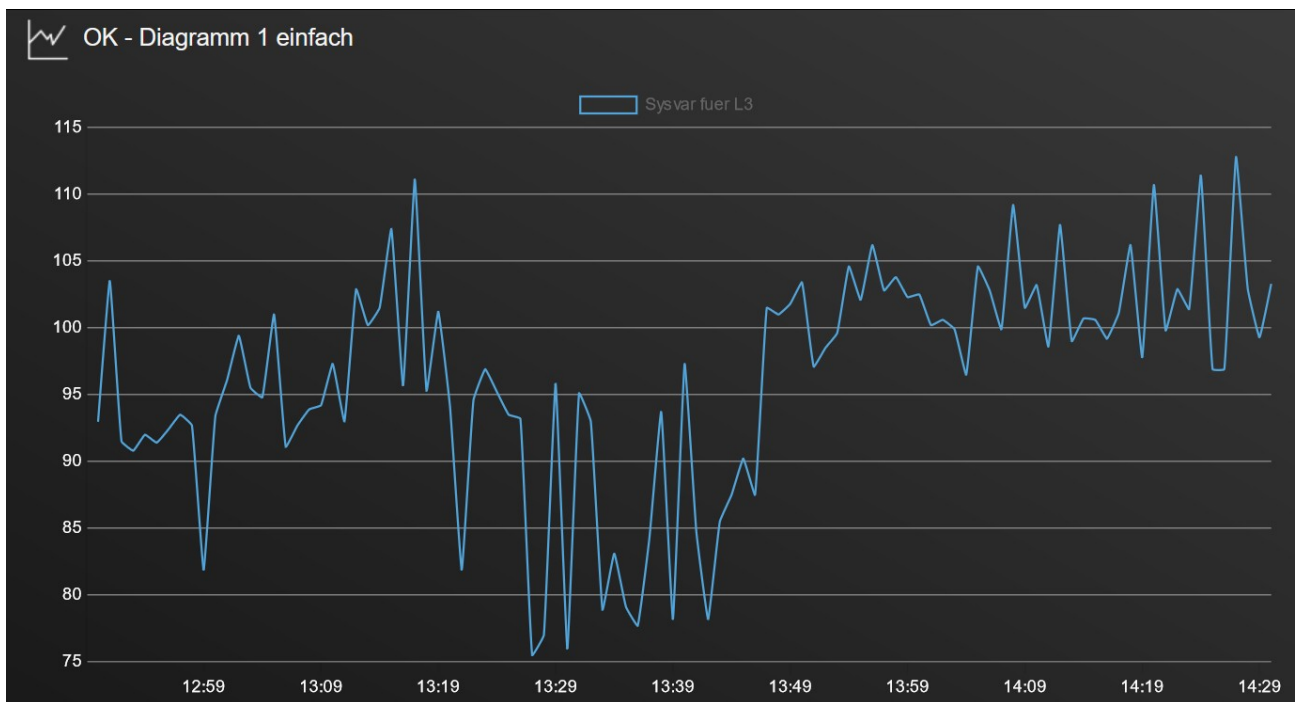
Parameter	Inhalt	Hinweise
"file"	"URL"	URL zur mp3-Stream Datei

### 3.4.4.3 diagramm

#### WIP

Eine Diagramm-Funktion für Homehub, die ohne Datenbank auskommt

Alternativ kann man Charts aus externen Programmen/Quellen auch über Iframe einbinden,



Benötigt wird/werden immer:

1. die ise-id(s) des/der anzuzeigenden Werte(s) oder Variablen

2. ein cronjob auf dem Webserver für "diagramm\_collect.php" (Pfade ggfls anpassen)

```
*1 * * * * cd /var/www/html/homehub/ && /usr/bin/php /var/www/html/homehub/diagramm_collect.php >/dev/null 2>&1
```

Bei der Synology kann man eine Aufgabe erstellen. (täglich wiederholen, Start 0:00, jede Minute, Ende 23:59)

Skript ausführen (IP/Pfad ggf. anpassen.):

```
curl http://192.168.178.117/homehub4/diagramm_collect.php
```

### **Option 1 - Einfach**

Einbinden in der custom.json

```
{
  "component": "diagramm",
  "name": "Grafik einfach",
  "icon": "time_graph.svg",
  "ise_id": "19467",
  "history": "10",
  "collect": "3"
}
```

Parameter	Inhalt	Hinweise
"ise_id"	"ise-id"	URL zur mp3-Stream Datei
"collect"	"Zahl"	Entspricht Minuten, wo Zustand der ise-id geloggt wird
"history"	"Zahl"	Anzahl der Werte, die in der CSV gespeichert werden
"legend"	"text1;text2; ... ;text5"	Texte, die in der legende angezeigt werden

### **Hinweise**

- Werte werden in einer csv-Datei in /VerzeichnisDesWebservers/homehub/cache abgelegt (z.B. diagramm\_19465\_20.csv)

- Ein Klick auf den Namen in der Homehub UI öffnet einen Speichern-Dialog für die CSV-Datei

- wenn das Diagramm angezeigt wird, aber noch keine Daten, mal mit der Frau reden, mit dem Hund um die Ecke gehen oder was anderes schönes machen - dann sollte was drin sein.

#### **3.4.4.4 ExtLink**

Komponente, um externe URLs auf einer neuen Seite aufzurufen



### **Beispiel 1** ,

Um in der Navigationsleiste ggf. die ein oder andere Schaltfläche sparen zu können, kann man ExtLink als Element einbinden und darüber z.B. eine Kategorie über die index.php anzuzeigen (hier die Liste der Systemvariablen)

#### Einbinden in der custom.json

```
{
  "component": "ExtLink",
  "name": "Systemvariablen",
  "icon": "control_clear.png",
  "url": "/index.php/Systemvariablen",
  "color": "#CC0000"
}
```

Parameter	Inhalt	Hinweise
"url"	"url"	URL zur externen Seite oder einer Kategorie in der index.php

#### Beispiel 2 - Einbinden einer externen Seite

#### Einbinden in der custom.json

```
{
  "component": "ExtLink",
  "name": "HM-Forum HH4",
  "icon": "control_clear.png",
  "url": "https://homematic-forum.de/forum/viewtopic.php?f=41&t=79208"
}
```

**Hinweis:** der Link wird im gleichen Fenster geöffnet

#### 3.4.4.5 Fritzbox und Fritzbox2

Listenansicht der ein-/ausgehenden Anrufe aus einer Fritzbox.  
Es existieren 2 Versionen.

Fritzbox-Anruferliste					
Datum	Typ	Anrufer	Angerufene	Gerät	Dauer
03.02.24 09:29		0633 5443 12	514 1 570	Mobilteil 1	0:12
03.02.24 06:57		0176 298 12 999	92 56 799	Anrufbeantworter55	0:01
03.02.24 06:47		0176 298 12 999	92 56 799	Anrufbeantworter55	0:01
03.02.24 06:47		0176 298 12 999	92 56 799		0:00
03.02.24 06:44		tert	92 56 799		0:00
02.02.24 16:25		Schappert Christian	92 56 799	Anrufbeantworter55	0:01
02.02.24 13:10		Schappert Christian	92 56 799	Anrufbeantworter55	0:01
02.02.24 11:49		514 1 570	0633 5443 12	Mobilteil 1	0:01
02.02.24 11:47		514 1 570	13 12	Mobilteil 1	0:02

## Vorbereitung:

1. Fritzbox Benutzer anlegen

2. Berechtigungen setzen unter "Heimnetz" - "Netzwerk" - "Netzwerkeinstellungen":

**Heimnetzfreigaben**

- ☒ Zugriff für Anwendungen zulassen

Diese Einstellung ermöglicht Anwendungen, Einstellungen der FRITZ!Box zu lesen  
Diensteanbieters. Die Übertragung erfolgt nach dem Standard TR-064.  
Für den Zugriff muss die Anwendung sich an der FRITZ!Box anmelden. Unter "System-  
Einstellungen" muss die Anwendung mit Erlaubnis für den Zugang aus dem Internet genutzt werden.
- ☒ Statusinformationen über UPnP übertragen

Über Universal Plug & Play (UPnP) werden im Heimnetz Statusinformationen der F

**Option 1:** "Fritzbox" : ursprüngliche Version von slice

Einrichtung siehe <https://homematic-forum.de/forum/viewtopic.php?f=41&t=76034&start=10#p741718>

**Option 2:** "Fritzbox2" : erweiterte Version

<https://homematic-forum.de/forum/viewtopic.php?p=796580#p795518>

Einbinden in der custom.json

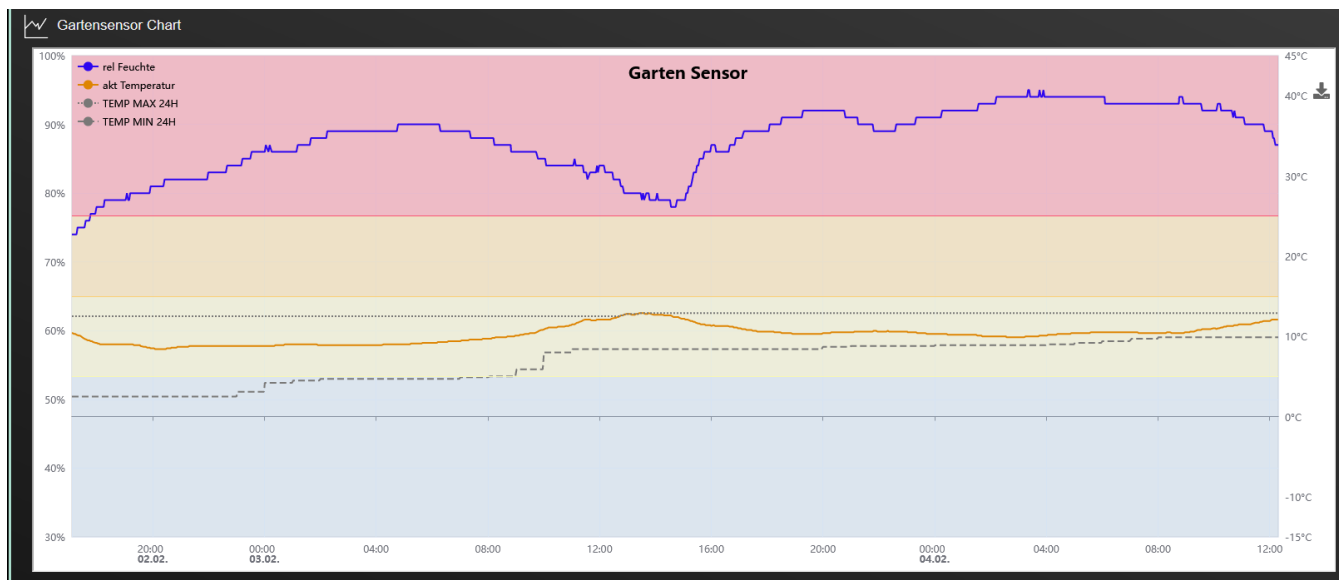
```
{
  "component": "Fritzbox2",
  "name": "Fritzbox-Anruferliste",
  "icon": "it_telephone.png",
  "fritz_url": "192.x.x.x",
  "fritz_pwd": "daskennwortdesusers",
  "fritz_user": "deruser",
  "anzahl": "20",
  "ausgehend": "#2E2EFE",
  "verpasst": "#FF0040",
  "eingehend": "#01DF01",
  "blockiert": "#BDBDBD",
  "ab": "#fc4103"
}
```

Parameter	Inhalt	Hinweise
"fritz_url"	"url_zur_Fritz"	
"fritz_pw"	"pw_des_users"	Passwort des Fritz-Benutzers
"fritz_user"	"name_des_users"	Name des Fritzbox-Users

"anzahl"	"Zahl"	Anzahl der anzuzeigenden, letzten Einträge aus der Fritte
"ausgehend", "verpasst", "eingehend", "blockiert", "ab"	"HEX-code"	HEX-Code der jeweiligen Farbe in der Liste
"aufgeklappt": "1"	"0 1"	Liste aufgeklappt anzeigen nein   ja

### 3.4.4.6 Iframe

Einbinden einer URL, welche dann in einem Iframe innerhalb von Homehub angezeigt wird.  
 Gut geeignet, um z.B. Charts aus Grafana, ioBroker echarts, oder andere externe Seiten einzubinden.  
 Bedingt geeignet, um z.B. Radio-Seiten einzubinden.



#### Hinweise:

- Manche Seiten blockieren das einbinden als iFrame. Eine Seite, um zu prüfen, ob dies der Fall ist, ist <https://www.iframe-generator.com/>

- für ioBroker eCharts muss der Port beim Aufruf geändert werden (8082 anstelle von 8081), um das blockieren zu umgehen. Siehe auch Beispiel unten

- Besonderheit zum einbinden von Grafana-Charts <https://homematic-forum.de/forum/viewtopic.php?f=41&t=50538&hilit=homehub&start=140#p585661>

Einbinden in der custom.json

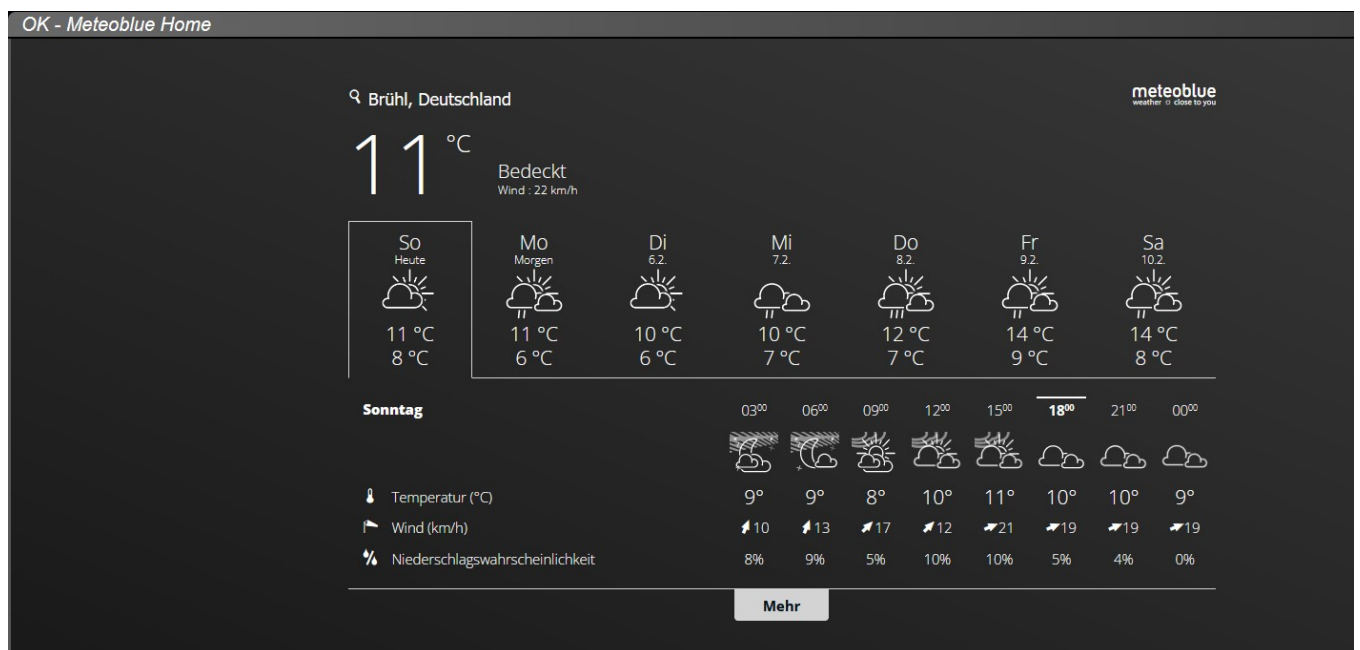
```
{
  "component": "Iframe",
  "name": "Gartensensor Chart",
  "icon": "time_graph.png",
  "aufgeklappt": "0",
  "url": "http://192.168.178.66:8082/adapter/echarts/chart/index.html?preset=echarts.0.Garten%20Sensor",
  "height": "70%"
}
```

Parameter	Inhalt	Hinweise
"url"	"url"	URL zur externen Seite
"aufgeklappt"	"0   1"	Liste aufgeklappt anzeigen nein   ja, Standard =1
"height"	"0-100%" oder "0-630px"	Höhe des Frames in % oder Pixel, Breite unverändert

### 3.4.4.7 Meteoblue

Funktionales Wetter-Widget, eingebunden über ein Iframe

Zulässigkeit der Nutzung unter Berücksichtigung der Freigaben von Meteoblue bitte selbst prüfen.





Den eigenen Ort kann man wie folgt herausfinden:

- meteoblue.com aufrufen und eigenen Ort wählen.

- In der Adressleiste des Browsers erscheint der Ort

[https://www.meteoblue.com/de/wetter/woc ... nd\\_2897216](https://www.meteoblue.com/de/wetter/woc...nd_2897216)

Den hinteren Abschnitt (also z.B. h%c3%bcrth\_deutschland\_2897216) als location in der custom.json hinterlegen.

Einbinden in der custom.json

```
{
  "component": "Meteoblue",
  "name": "Wettervorhersage",
  "icon": "weather_wind.png",
  "color": "#595959",
  "location": "h%c3%bcrth_deutschland_2897216"
}
```

Parameter	Inhalt	Hinweise
"location"	"string_aus_url"	Siehe oben
"height"	"0-100%" oder "0-630px"	Höhe des Frames in % oder Pixel, Breite unverändert

#### 3.4.4.8 Multiview

WIP

Anzeige von Werten/Inhalten für verschiedene Geräte/Variablen in einer kombinierten Zeile



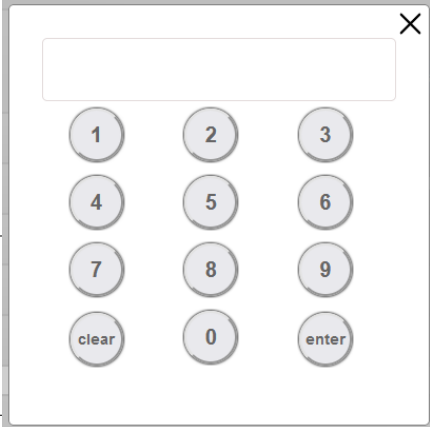
Einbinden in der custom.json

```
{
  "component": "MultiView",
  "name": "NOK MultiView L1-L3",
  "icon": "time_graph.png",
  "ise_id": "19465,19466,19467",
  "ise_unit": "W L1,W L2,W L3",
  "ise_component": "3EM_Watt_L1,3EM_Watt_L2,3EM_Watt_L3",
  "ise_datapoint": ",",
}
```

Parameter	Inhalt	Hinweise
"ise_id"	"ise-id"	Der CCU Systemvariablen
"ise_unit"	"Text"	Optionale Bezeichnung der Unit für die Werte
"ise_component"	?	?
"ise_datapoint"	?	?
"ise_datavaluelist"	?	?

### 3.4.4.9 sysvarpin

Änderungssperre für Variablen über 4-stellige PIN



**Beispiel 1:** CCU Variable Typ Zahl  
Einbinden in der custom.json

```
{
  "component": "sysvarpin",
  "icon": "secur_burglary.png",
  "display_name": "sysvarpin DC 0815",
  "code": "0815",
  "ise_id": "3000"
}
```

Parameter	Inhalt	Hinweise
"display_name"	"Text"	Anzeigetext, beliebig
"code"	"0000"	4-stellige PIN
"ise_id"	"ise-id"	Der CCU Variablen, "http://[IP-der-CCU]/addons/xmlapi/sysvarlist.cgi"
"pinvalue"	"value1,value2, ...,value_n"	Wertebezeichnungen aus der CCU-Variable, case sensitiv

**Beispiel 2:** CCU Variable Typ Logik- oder Werteliste  
Einbinden in der custom.json

```
{
  "component": "sysvarpin",
  "icon": "secur_burglary.png",
  "display_name": "sysvarpin Werteliste truefalse 4711",
  "code": "4711",
  "ise_id": "2512",
  "pinvalue": "reduziert,normal,aus"
}
```

- Bei einer Werteliste von mit den Werten A B C D, kann man mit "pinvalue": "A,B,C" die drei Werte schützen. "pinvalue": "A" würde nur bei A eine Pin-Abfrage erzeugen.
- Bei wahr/falsch (true/false) ist dies identisch: "pinvalue": "true,false" schütze jede Änderung. "pinvalue": "true" lediglich das aktivieren der Variable.

### **Hinweise:**

- PIN-Eingabefenster muß geschlossen werden, kurz warten bis geänderter Wert sichtbar ist
- Forum: <https://homematic-forum.de/forum/viewtopic.php?f=41&t=76034&start=60#p762063>

#### **3.4.4.10 Tagesschau**

Anzeige von Tagesschau in 100 Sekunden im Iframe

Einbinden in der custom.json

```
{
  "component": "Tagesschau",
  "name": "OK - Tagesschau in 100 Sekunden",
  "icon": "it_radio.png"
}
```

#### **3.4.4.11 Tankerkönig**

Aktuelle Spritpreise. Je Tankstelle eine Zeile



- eigenen API key holen <https://creativecommons.tankerkoenig.de/> (rechts oben Tools auswählen)
- Tankstellen ID herausfinden: <https://creativecommons.tankerkoenig.de/TankstellenFinder/index.html>

Einbinden in der custom.json

```
{
  "component": "Tankerkoenig",
  "name": "Aral",
  "icon": "scene_gas_station.png",
  "api_key": "aaaaaaa-bbbb-cccc-dddd-eeeeeeeeeeee",
  "station_id": "51e3908d-b5ee-4cf6-89b1-2cbc4b6c2710",
  "fuel_types": ["diesel", "e10", "e5"]
}
```

Parameter	Inhalt	Hinweise
"api_key"	"api-Key"	Eigener API-Key, <a href="https://creativecommons.tankerkoenig.de/">https://creativecommons.tankerkoenig.de/</a>
"station_id"	"ID"	ID der Tankstelle <a href="https://creativecommons.tankerkoenig.de/TankstellenFinder/index.html">https://creativecommons.tankerkoenig.de/TankstellenFinder/index.html</a>
"fuel_types"	["diesel", "e10", "e5"]	Name der Sorten, ACHTUNG eckige Klammern [ ]

#### **3.4.4.12 weatherunderground**

Anzeige für Weatherunderground. Erfordert eigenen API-Key sowie eigene Wetterstation (pws).

Ungetestet

Einbinden in der custom.json (Beispiel) mit

```
{
  "component": "WeatherUnderground",
  "name": "Wetter",
  "icon": "weather_sunrise.png",
  "api_key": "123456789123456789123456789aaaaa",
  "station": "pws:IBRHL82"
}
```

### 3.4.4.13 Sonnenstand

Anzeige der in der CCU berechneten Systemvariablen von Azimuth und Elevation der Sonne.  
Die Datei Sonnenstand.php ist nicht Bestandteil der standard Homehub-ZIP Datei und muss erst erstellt werden.

OK - Sonne Sonne Azimut: 241.7 ° Sonne Elevation: 0.1 °

1. Falls nicht schon vorhanden, erstellen von zwei CCU Systemvariablen

SV_Sonne_Azimut	Horizontalwinkel	Zahl	Minimalwert: -360 Maximalwert: 360
SV_Sonne_Elevation	Höhenwinkel	Zahl	Minimalwert: -180 Maximalwert: 180

2. Erstellen eines CCU-Programms mit zeitgesteuertem, wiederkehrendem Trigger (z.B. alle 5min)

Name	Beschreibung	Bedingung
Sonnenstand	Astro und rel zu Solarpanel	Zeit: Periodisch Ganztägig beginnend

**Bedingung: Wenn...**  

Zeitsteuerung **Periodisch Ganztägig beginnend am 28.01.2024** zu Zeitpunkten auslösen

UND

ODER

**Aktivität: Dann...** ☒ Vor dem Ausführen alle laufenden Verzögerungen für diese Aktivitäten beenden (z.B. Retriggern).  

Skript **! Azimut und Elevation aus Systemvariablen holen ! Name Az...** sofort

~~Skript **! Einfallswinkel der Sonne für Solarkollektor ! https://www....** verzögert um 5 Sekunden~~

**Aktivität: Sonst...** ☐ Vor dem Ausführen alle laufenden Verzögerungen für diese Aktivitäten beenden (z.B. Retriggern).

## Skript

```
! Azimut und Elevation aus Systemvariablen holen
! Name Azimut Systemvariable : SV_Sonne_Azimut / Typ: Zahl / Wert: -360 bis 360 / Horizontalwinkel
! Name Altitude Systemvariable: SV_Sonne_Elevation / Typ: Zahl / Wert: -180 bis 180 / Höhenwinkel

! Auslesen der neuen Werte in die o.g. Systemvariablen
dom.GetObject("SV_Sonne_Azimut").State(system.SunAzimuth().Round(1));
dom.GetObject("SV_Sonne_Elevation").State(system.SunAltitude().Round(1));
```

Hinweis: ich habe auf eine Nachkommastelle gerundet .Round(1), das kann ja geändert werden.

## 3. Erstellen einer Datei Sonnenstand.php im Verzeichnis

/VerzeichnisDesWebserver/homehub/custom/components sowie die zwei data-id's in zeile 7+8 durch die eigenen ise-id's ersetzen

## Inhalt

```
<?php
function Sonnenstand($component) {
    return '<div class="hh">'
        . '<div class="pull-left">' . $component["name"] . '</div>'
        . '<div class="pull-right">'
        . '<span class="info">Sonne Azimut:<span class="info" data-id="16598" data-component="SysVar" data-datapoint="4"
data-unit="&deg"></span></span>'
        . '<span class="info">Sonne Elevation:<span class="info" data-id="16599" data-component="SysVar" data-datapoint="4"
data-unit="&deg"></span></span>'
        . '</div>'
        . '<div class="clearfix"></div>'
        . '</div>';
}
```

## 4. Einbinden in der custom.json mit


```
{
  "component": "Sonnenstand",
  "name": "OK - Sonne",
  "icon": "weather_sun.png"
}
```

Eine hilfreiche Seite mit Solar-, Geo- und Photovoltiakdaten

<https://www.sonnenverlauf.de/#/50.9383,6.9586,9/2024.01.29/08:45/1/3>

3.4.4.14 Mondphase

Anzeige der in der CCU berechneten Systemvariablen für Mondphase, -stand, -tag, sowie nächstem Vollmond.  
Die Datei Mondphase.php ist nicht Bestandteil der standard Homehub-ZIP Datei und muss erst erstellt werden.

 OK - Mond

abnehmender Vollmond    Mondstand: -67 %    Montag: Tag 5    Vollmond am: 24.02.2024

1. Falls nicht schon vorhanden, erstellen von vier CCU Systemvariablen

Mondphase		Zeichenkette		
Mondstand		Zeichenkette		%
Montag		Zeichenkette		
naechster_Vollmond		Zeichenkette		

2. Erstellen eines CCU-Programms mit zeitgesteuertem Trigger 1x pro Tag (z.B. um 0:15)

Name	Beschreibung	Bedingung
Sonne_Mond	Sonnenauf_untergang Mondphase	Zeit: Täglich um 00:15 Uhr beginnend a

Bedingung: Wenn...

Zeitsteuerung

Täglich um 00:15 Uhr beginnend am 27.01.2024

zu Zeitpunkten auslösen

UND

ODER

Aktivität: Dann... ☒ Vor dem Ausführen alle laufenden Verzögerungen für diese Aktivitäten beenden (z.B. Retriggern).

Skript

!Testversion V2.1 !Script zur Mondphasenberechnung von Dako ...

sofort

Skript

~~object sysvarSA = (dom.GetObject(ID\_SYSTEM\_VARIABLES)).Get("...~~

~~verzögert um~~ 5 ~~Sekunden~~

Sonst...

☐ Vor dem Ausführen alle laufenden Verzögerungen für diese Aktivitäten beenden (z.B. Retriggern).

## Skript aus HM-Forum

```
!Testversion V2.1
!Script zur Mondphasenberechnung von Dako V1.1
!Erweiterung Mondphase zu- oder abnehmend von hanan V2.0
!Erweiterung Mondtag für Sera LED Moonlight Kit V2.1
!-----
!Hinweise:
!Die Berechnung erfolgt mit dem durchschnittlichen synodischen Mondmonat, weshalb keine
!Stundengenaue Berechnung erfolgen kann
!-----
!Systemvariablen:
!naechster_Vollmond (Zeichenkette): naechster Vollmond
!Mondstand (Zeichenkette): Mondphase in % (Neumond = 0%)
!Mondphase (Zeichenkette): Mondphase in Textform
!Mondtag (Zeichenkette): Mond-Kalender in Textform
!-----
!Variablen definieren
real a_1 = 1.0*(system.Date("%F %X").ToTime().ToInteger()); !aktueller UNIX-Timestamp (1374688453)
real a_2 = 1106652740.0;!referenz vollmond
real a_3 = 2551443.0;!synodischer mondmonat
real a_4 = 1105362167.0;!referenz neumond

!Aktuelle Mondphase
real x2 = ((a_1-a_4)/a_3);
integer x3 = x2.ToInteger();
real x4 = (x2-x3)*200;
integer x5 = x4.ToInteger();

string z3 = "Fehler!";
string z4 = "Fehler!";

if(x5 <=200){real x6 = x5-200;}
if(x5 <=100){real x6 = x5;}

if(x5 <= 205){z3 = "Neumond";}
if(x5 <= 195){z3 = "abnehmender Halbmond";}
if(x5 <= 150){z3 = "abnehmender Vollmond";}
if(x5 <= 105){z3 = "Vollmond";}
if(x5 <= 95){z3 = "zunehmender Halbmond";}
if(x5 <= 50){z3 = "zunehmender Neumond";}
if(x5 <= 5){z3 = "Neumond";}

if(x5 <= 198) {z4 = "Tag 14";}
if(x5 <= 191) {z4 = "Tag 13";}
if(x5 <= 184) {z4 = "Tag 12";}
if(x5 <= 177) {z4 = "Tag 11";}
if(x5 <= 170) {z4 = "Tag 10";}
if(x5 <= 163) {z4 = "Tag 9";}
if(x5 <= 156) {z4 = "Tag 8";}
if(x5 <= 149) {z4 = "Tag 7";}
if(x5 <= 142) {z4 = "Tag 6";}
if(x5 <= 135) {z4 = "Tag 5";}
if(x5 <= 128) {z4 = "Tag 4";}
if(x5 <= 121) {z4 = "Tag 3";}
if(x5 <= 114) {z4 = "Tag 2";}
if(x5 <= 107) {z4 = "Tag 1";}
if(x5 <= 100) {z4 = "Tag 29";}
if(x5 <= 93) {z4 = "Tag 28";}
if(x5 <= 86) {z4 = "Tag 27";}
if(x5 <= 79) {z4 = "Tag 26";}
if(x5 <= 72) {z4 = "Tag 25";}
if(x5 <= 65) {z4 = "Tag 24";}
if(x5 <= 58) {z4 = "Tag 23";}
if(x5 <= 51) {z4 = "Tag 22";}
if(x5 <= 46) {z4 = "Tag 21";}
if(x5 <= 39) {z4 = "Tag 20";}
if(x5 <= 32) {z4 = "Tag 18";}
if(x5 <= 25) {z4 = "Tag 18";}
if(x5 <= 18) {z4 = "Tag 17";}
if(x5 <= 11) {z4 = "Tag 16";}
if(x5 <= 4) {z4 = "Tag 15";}

dom.GetObject("Mondstand").State(x6.ToString(0)#" %");!Ausgabe als Prozentzahl
dom.GetObject("Mondphase").State(z3);!Ausgabe als Text
dom.GetObject("Mondtag").State(z4);!Ausgabe als Text

!Naechster Vollmond am:
while(a_2 <= a_1){a_2 = a_2 + a_3;}
real x1 = a_2.ToTime();
string y1 = x1.Format("%d.%m.%Y");
dom.GetObject("naechster_Vollmond").State(y1);
```

### 3. Erstellen einer Datei Mondphase.php im Verzeichnis

/VerzeichnisDesWebserver/homehub/custom/components sowie die zwei data-id's in zeile 7-10 durch die eigenen ise-id's ersetzen.

#### Inhalt

```
<?php
function Mondphase($component) {
    return '<div class="hh">'
        . '<div class="pull-left">' . $component["name"] . '</div>'
        . '<div class="pull-right">'
        . '<span class="info"><span class="info" data-id="23742" data-component="SysVar" data-datapoint="" data-unit=""></span>'
        . '<span class="info">Mondstand:<span class="info" data-id="23741" data-component="SysVar" data-datapoint="" data-unit=""></span>'
        . '<span class="info">Mondtag:<span class="info" data-id="23743" data-component="SysVar" data-datapoint=""></span>'
        . '<span class="info">Vollmond am:<span class="info" data-id="23740" data-component="SysVar" data-datapoint="" data-unit=""></span>'
        . '</div>'
        . '<div class="clearfix"></div>'
        . '</div>';
}
```

### 4. Einbinden in der custom.json mit

```
{
  "component": "Mondphase",
  "name": "OK - Mond",
  "icon": "weather_moon_phases_7_half.png"
}
```

#### 3.4.4.15 Sonnenaufuntergang

Homehub bring von Hause aus schon Sonnenauf- und untergangszeiten mit und zeigt diese ober rechts neben der Uhr.

Für diejenigen, welche aus - wie auch immer gearteten Gründen - diese aber als Zeile in der View haben möchten, können wie u.a. vorgehen.

Die Datei Sonnenaufuntergang.php ist nicht Bestandteil der standard Homehub-ZIP Datei und muss erst erstellt werden.



#### 1. Falls nicht schon vorhanden, erstellen von zwei CCU Systemvariablen

Sonnenaufgang		Zeichenkette		
Sonnenuntergang		Zeichenkette		



- Erstellen eines CCU-Programms mit zeitgesteuertem Trigger 1x pro Tag (z.B. um 0:15)  
(ich habe das zusammen mit der Mondphase in einem Programm)

Name	Beschreibung	Bedingung (V)
Sonne_Mond	Sonnenauf_untergang Mondphase	Zeit: Täglich um 00:15 Uhr beginnend am :

**Bedingung: Wenn...**

Zeitsteuerung ☐ Täglich um 00:15 Uhr beginnend am 27.01.2024 ☐ zu Zeitpunkten auslösen ☐

☐ UND ☐

☐ ODER ☐

**Aktivität: Dann...** ☒ Vor dem Ausführen alle laufenden Verzögerungen für diese Aktivitäten beenden (z.B. Retriggern).

☐ Skript ☐ Testversion V2.1 !Skript zur Mondphasenberechnung von Dako... ☐ sofort ☐

☐ Skript ☐ object sysvarSA = (dom.GetObject(ID\_SYSTEM\_VARIABLES)).Get("... verzögert um   ☐

**Aktivität: Sonst...** ☐ Vor dem Ausführen alle laufenden Verzögerungen für diese Aktivitäten beenden (z.B. Retriggern).

## Skript

```
object sysvarSA = (dom.GetObject(ID_SYSTEM_VARIABLES)).Get("Sonnenaufgang");
object sysvarSU = (dom.GetObject(ID_SYSTEM_VARIABLES)).Get("Sonnenuntergang");
if ( (sysvarSA) && (sysvarSU) && (sysvarSA.ValueSubType() == istChar8859) && (sysvarSU.ValueSubType() == istChar8859) ){
sysvarSA.State(system.SunriseTime("%H:%M"));
sysvarSU.State(system.SunsetTime("%H:%M"));
```

- Erstellen einer Datei Sonnenaufuntergang.php  
im Verzeichnis /VerzeichnisDesWebservers/homehub/custom/components  
sowie die zwei data-id's in zeile 7+8 sind durch die eigenen ise-id's ersetzen

```
<?php
function Sonnenaufuntergang($component) {
    return '<div class="hh">'
        . '<div class="pull-left">' . $component["name"] . '</div>'
        . '<div class="pull-right">'
        . '<span class="info">Sonnenaufgang:<span class="info" data-id="23841" data-component="SysVar" data-datapoint="" data-unit=""></span></span>'
        . '<span class="info">Sonnenuntergang:<span class="info" data-id="23842" data-component="SysVar" data-datapoint="" data-unit=""></span></span>'
        . '</div>'
        . '<div class="clearfix"></div>'
        . '</div>';
}
```

- Einbinden in der custom.json mit

```
{
```

```
"component": "Sonnenaufuntergang",
"name": "OK - Sonne A/U",
"icon": "weather_sunrise.png"
}
```

#### 3.4.4.16 VIR-LG-RGBW-DIM, VIR-LG-WHITE-DIM

WIP

#### 3.4.4.17 Webcam

Ungetestet, WIP

Einbinden in der custom.json mit

Parameter	Inhalt	Hinweise
name	angezeigter Name	
url	URL zum Bild	
Icon	Icon	
autorefresh	0, 1	1 = aktualisiert selbstständig, 0 = statisch
aufgeklappt	0, 1	0 zugeklappt, 1 aufgeklappt - standard =1

### 3.4.5 Sonstige Customizations

#### 3.4.5.1 Stylesheets

Das Aussehen von HomeHub kann mit der Datei assets/css/custom.css angepasst werden. Das Stylesheet wird nach dem HomeHub Stylesheet aufgerufen. Es ist auch möglich, das komplette Design für einen einzelnen Menüeintrag zu verändern. Dazu muss eine HTML Datei für diese Kategorie angelegt werden und im Ordner app/Views/lowercase\_category\_name.html abgelegt werden.

#### 3.4.5.2 Angezeigter Text bei Logik-/Werteliste-Variablen durch Icons ersetzen

Standardmäßig wird bei CCU Variablen der Typen Logik oder Werteliste der Variableninhalt als Text angezeigt.



Um diesen Text durch bedingungsabhängige Icons zu ersetzen, kann folgendermaßen vorgegangen werden:

1. Eine CCU-Variable des Typs Zeichenkette anlegen, z.B.

HH_Test_IconChange	html-Tag Icon	Zeichenkette		
--------------------	---------------	--------------	--	--

2. Ein CCU-Programm anlegen, das über Skripts, den jeweiligen Verweis zu den gewünschten Icons in diese Variable schreibt.

Im Beispiel wurden virtuelle Taster als Trigger genutzt. Kann aber z.B. ein Tür-/Fensterkontakt mit dem entsprechenden Zustand (geschlossen, gekippt, geöffnet) sein, o.ä.

Name	Beschreibung	
HH_Test_IconChange_Prog	Icon wechseln ueber string	Kanalzustand

**Bedingung: Wenn...**

Geräteauswahl **HH4 Test VT0** bei Tastendruck kurz ☒ ☐

☒ UND ☐

☒ ODER ☐

**Aktivität: Dann...** ☒ Vor dem Ausführen alle laufenden Verzögerungen für diese Aktivitäten beenden (z.B. Retriggern).

Skript **string icon = "<img src=icon/edit\_numeric\_0.png>"; dom.GetObject("HH\_Test\_IconChange").State(icon);** sofort ☒

☒

**Bedingung: Sonst, wenn...**

Geräteauswahl **HH4 Test VT1** bei Tastendruck kurz ☒ ☐

☒ UND ☐

☒ ODER ☐

**Aktivität: Dann...** ☒ Vor dem Ausführen alle laufenden Verzögerungen für diese Aktivitäten beenden (z.B. Retriggern).

Skript **string icon = "<img src=icon/edit\_numeric\_1.png>"; dom.GetObject("HH\_Test\_IconChange").State(icon);** sofort ☒

☒

**Bedingung: Sonst, wenn...**

Geräteauswahl **HH4 Test VT2** bei Tastendruck kurz ☒ ☐

☒ UND ☐

☒ ODER ☐

**Aktivität: Dann...** ☒ Vor dem Ausführen alle laufenden Verzögerungen für diese Aktivitäten beenden (z.B. Retriggern).

Skript **string icon = "<img src=icon/edit\_numeric\_2.png>"; dom.GetObject("HH\_Test\_IconChange").State(icon);** sofort ☒

☒

### Skript für HH4 Test VT1

```
string icon = "<img src=icon/edit_numeric_0.png>";
dom.GetObject("HH_Test_IconChange").State(icon);
```

### Skript für HH4 Test VT2

```
string icon = "<img src=icon/edit_numeric_1.png>";
dom.GetObject("HH_Test_IconChange").State(icon);
```

3. In der custom.json dann die neue CCU-Variable einbinden

```
{
  "name": "HH_Test_IconChange",
  "icon": "edit_numeric_0.png",
  "display_name": "HH_Test_IconChange-String",
  "color": "#00CC33"
}
```

Angezeigt wird das dann so (hier wenn Taster "HH4 Test VT2" getriggert hat):

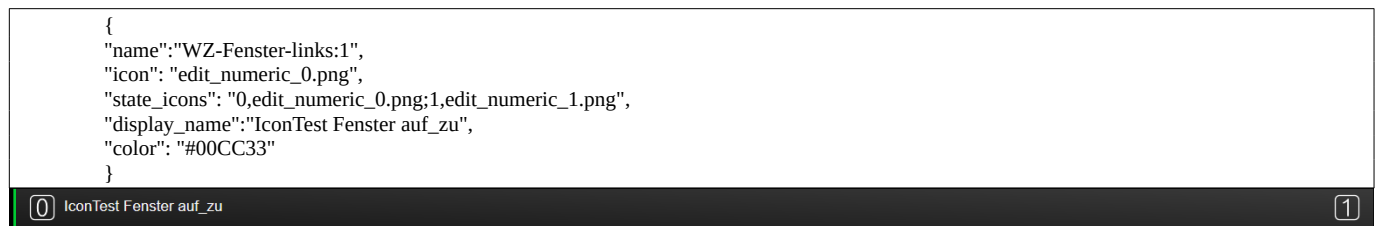


Diskussion zu dem Thema hier <https://homematic-forum.de/forum/viewtopic.php?f=41&t=79208&p=797898#p797680>

### 3.4.5.3 Icons für Geräte mit Statusmeldungen "state\_icons"

Für Geräte, die Statusmeldungen wie 0,1,2 oder false,true zurück geben (z.B. Fenster-/Türkontakte, Kontaktschnittstelle,...) kann man die in HH vordefinierten, statusabhängigen Icons verändern.

Beispiel



Wird kein Icon angezeigt, statt 0 und 1 mal false und true probieren oder umgekehrt.

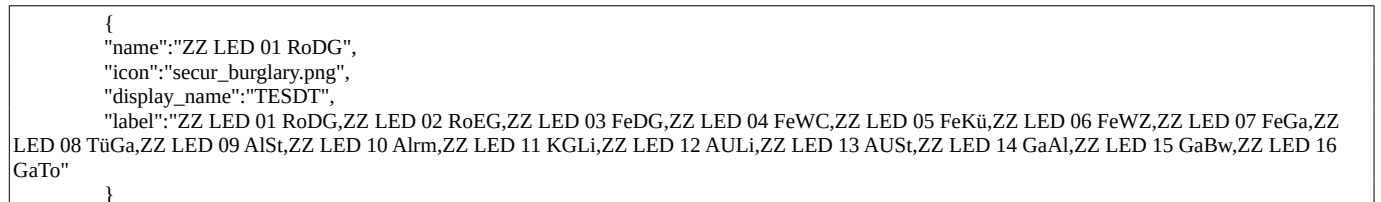
Wird das alte Icon angezeigt, mal einen Browser-Reload (mit gehaltener Shift-Taste) durchführen.

















### 3.4.5.4 Einbindung LED-Display HM-OU-LED16

Farbindikatoren für HM-OU-LED16

Als Erweiterung lässt sich das komplette Display wie in real abbilden (siehe Beschreibung in components\hm-ou-led16.php)

Einbindung in custom.json



	ZZ LED 01 RoDG	ZZ LED 02 RoEG	
	ZZ LED 03 FeDG	ZZ LED 04 FeWC	
	ZZ LED 05 FeKü	ZZ LED 06 FeWZ	
	ZZ LED 07 FeGa	ZZ LED 08 TüGa	
	ZZ LED 09 AlSt	ZZ LED 10 Alrm	
	ZZ LED 11 KGLi	ZZ LED 12 AULi	
	ZZ LED 13 AUSt	ZZ LED 14 GaAl	
	ZZ LED 15 GaBw	ZZ LED 16 GaTo	

Siehe auch <https://homematic-forum.de/forum/viewtopic.php?f=41&t=79208&start=310#p798181>

## 4 Tips & Tricks

### 4.1 Änderungen an custom.json oder categories.json

Homehub reagiert „sensibel“ auf Syntax-Fehler in der custom.json oder categories.json, insbesondere setzen von Klammern oder Kommata.

Auswirkungen bei Fehlern gehen im einfachsten Fall von der nicht-Anzeige einer einzelnen Zeile in der UI, über eine komplett leere Kategorie-Seite, bis zu komplett leeren Kategorien.

Falls man also in der json-Syntax noch nicht so erfahren ist, empfiehlt es sich, immer erst nur einzelne Blöcke oder Zeilen zu ändern → speichern → kontrollieren.

Die Suche nach solchen Fehlern in größeren Blöcken kann mitunter ziemlich zeitraubend werden.

Ein Editor im UTF-8 Format sollte verwendet werden.

### 4.2 Umlaute & Sonderzeichen

In der CCU sollte die Verwendung von Umlauten und Sonderzeichen bei der Namensvergabe von Geräten, Systemvariablen oder Programme generell vermieden werden.

In HomeHub führt die direkte Verwendung von z.B. Umlauten im Parameter „name“ zu einer leeren Seite im UI

Umlaute können:

- in Parameter wie „name“ in Unicode, oder
- in url's HTML – kodiert eingefügt werden (ohne das Semikolon)

Beispiele:

Zeichen	Beschreibung	Name in HTML	Unicode
Ä	Deutscher Umlaut Ä	&Auml;	&#196;
ä	Deutscher Umlaut ä	&auml;	&#228;
Ü	Deutscher Umlaut Ü	&Uuml;	&#220;
ü	Deutscher Umlaut ü	&uuml;	&#252;
Ö	Deutscher Umlaut Ö	&Ouml;	&#214;
ö	Deutscher Umlaut ö	&ouml;	&#246;
ß	Scharfes S	&szlig;	&#223;

Quelle: [https://html-php.de/index.php?seite=h\\_zeichen](https://html-php.de/index.php?seite=h_zeichen)

<http://www.sonderzeichen.de/Latein-1/Unicode-00DC.html>

### 4.3 ISE-IDs

Diese werden an verschiedener Stelle gebraucht. Möglichkeiten, diese für das jeweilige Gerät/Systemvariable zu finden sind:

1. xml-api

Geräte: devicelist.cgi oder statelist.cgi

Variablen: sysvarlist.cgi

Beispiel [http://\[IP-der-CCU\]/addons/xmlapi/sysvarlist.cgi](http://[IP-der-CCU]/addons/xmlapi/sysvarlist.cgi)

2. ioBroker – Objekte - hm-rega.0 – ID

3. SDV von Black

#### **4.4 Neue / fehlende Geräte**

WIP

# Anhang

## A1 Changelog

04082024

- 3.2.1 „Home“ Funktion - neu
- 3.4.3 Parameter showtime für ioBroker-Objekte

04072024

- 2 ff – geänderte Voraussetzungen/Installation mit HH4.1
- 3.4.2.2 neuer Parameter showtime hinzugefügt
- 3.4.5.4 Update für LED-Display HM-OU-LED16 hinzugefügt
- 3.4.2 Parameter "Label" für CCU-Programme hinzugefügt
- 3.4.4.13 Sonnenstand Skript Fehler behoben, Unit geändert
- diverse kleinere Korrekturen

02282024

- erste public version