

# Vorgehen für Geräte

Umsetzung von konkreten Problemstellungen und Anbindung von Geräten/Sensoren an Casa Corrently.

- EBus Geräte (Heiztechnik)
- TP-Link Kasa WLAN Steckdose HS110 mit Stromaufzeichnung
- AVM FRITZ!DECT 200 Steckdose
- Anbindung IP-Symcon Gebäudeautomation

# EBus Geräte (Heiztechnik)



eBus ist ein im Bereich der Heizungstechnik verwendetes Schnittstellenprotokoll, das auf der seriellen Schnittstellentechnik RS232 in Twisted-Pair-Technik mit zwei Drähten aufbaut. ([Wikipedia](#))

Zur Verwendung von eBus innerhalb von Casa Corrently hat sich die Installation der Software [ebusd](#) als nützlich erwiesen, welcher in den aktuellen Versionen eine direkte MQTT Unterstützung besitzt. Es empfiehlt sich den Daemon direkt auf der Casa Corrently Box zu installieren und wie folgt aufzurufen:

```
/usr/bin/ebusd --scanconfig --mqttclientid=ebusd --mqttthost=127.0.0.1 --mqttport=1883
```

## Beispiel: Fahrplanmanagement Wärmepumpe



In diesem Beispiel kommt eine Vaillant Wärmepumpe zum Einsatz, deren Warmwasserzubereitung über Casa Corrently optimiert wird. Zunächst wird der kombinierte GrünstromIndex ([eXgsi](#)) abgefragt, um dann die beste Zeit für die Warmwasserzubereitung zu definieren und im Programm (Timer) des Gerätes zu hinterlegen.

Funktion GSI integrieren

```
let gsi = global.get("eXgsi");

let hours = [];
let hwset = false;

for(let i=0;i<gsi.length;i++) {
  if(flow.get("temp_avg")>15) {
    if(gsi[i].fields.s2 > 0) {
      node.status({text: "In "+i+" Stunden"});
      if(!hwset) {
        flow.set('hwstart', Math.round(gsi[i].timestamp/1000000));
        hwset=true;
      }
    }
  }
}
```

```
        hours.push(gsi[i].fields.s2 * 1000);  
    } else {  
        hours.push(gsi[i].fields.s20 * 450);  
    }  
}  
msg.payload = hours;  
  
return msg;
```

Je nach durchschnittlicher Temperatur der folgenden Tage wird entweder der Zweistundenschalter (Zeile 16) oder der 20 Stunden Schalter (Zeile 18) als Basis verwendet.

Über dein `MQTT Out Node` werden im Anschluss die Zeitschaltuhren mittels des jeweiligen MQTT Topics gesetzt. Beim Empfang einer neuen Nachricht wird der ebusd diese Nachricht direkt an die Wärmepumpe senden.

Bei der eingesetzten Wärmepumpe existiert ein spezieller "Schnellschalter" für eine einmalige Speicherladung. Wird dieser gesetzt, so wird die Warmwassertemperatur auf dem höchstmöglichen Wert gehalten (unabhängig vom Zeitplan). Sobald ein Überschuss an Strom vorhanden ist und in das Netz eingespeist werden müsste, wird dieser Schalter gesetzt.

# TP-Link Kasa WLAN Steckdose HS110 mit Stromaufzeichnung

Diese Smart-Home Steckdose zeichnet dadurch aus, dass sie keine weiteren Controller benötigt und direkt in das WLAN eingebunden werden kann. Für die Verwendung innerhalb von Casa Corrently kommt die Strommessung zum Einsatz zum Aufbau von flexiblen Lasten bei Geräten, die dies eigentlich nicht unterstützen. (vergleiche [Ebene 1](#))

Vorinstallierte Komponente: [node-red-contrib-tplink-smarthome](#)

[hs110\\_kasa\\_smartplug.png](#)  
Image not found or type is unknown

Damit die Leistung von der Steckdose empfangen werden kann, muss eine Nachricht "getMeterInfo" Nachricht gesendet werden.

Funktion: SmartPlug:getMeterInfo

```
msg.payload="getMeterInfo";  
return msg;
```

Die Rückgabe des Leistungswertes steht dann im Feld `power_mw` zur Verfügung und kann weiter verarbeitet werden.

Funktion: Wirkleistung

```
msg.payload = Math.round(msg.payload.power_mw / 1000);  
node.status({text: "P: "+msg.payload+" W"});  
// msg.payload = msg.payload;  
const subSUM='Consumption';  
  
if(isNaN(msg.payload)) msg.payload = 0;  
  
let sum = flow.get(subSUM) * 1;  
if(isNaN(sum)) sum = 0;
```

```

let previous = context.get("previous") * 1;
if(isNaN(previous)) previous = 0;

if(flow.get("SaldoID") != context.get("SaldoID")) {
    context.set("SaldoID", flow.get("SaldoID"));
} else {
    sum -= previous;
}
sum += msg.payload;

context.set("previous", msg.payload);

flow.set(subSUM, sum);

return msg;

```

## Intelligentes Schalten (Last Flexibilität)

Das hier gezeigte Beispiel nutzt lediglich einen Einschalter, der das angeschlossene Gerät für eine Stunde aktiviert, wenn innerhalb dieser Stunde nicht ein weiterer Einschalt Impuls kommt (gesetzt über die Funktion "Timer" der TP-Link Kasa App).

[flexen.png](#) file not found or type unknown

In diesem Fall wird ein Webhook innerhalb von Node-RED genutzt, um von einer anderen Anwendung/Dienst die Steckdose zu aktivieren.

Funktion: Turn On

```

msg.payload=true;
return msg;

```

# AVM FRITZ!DECT 200

## Steckdose

Die FRITZ!DECT Steckdose zeichnet sich durch ihre nahtlose Anbindung an ein auf die Fritz Box basierendes System aus. Zusätzlich hat diese Steckdose eine Energieaufzeichnung, die sich innerhalb von Casa Corrently nutzen lässt.

Vorinstallierte Komponente: [node-red-contrib-fritzapi](#)

### Beispiel: Erzeugungsmessung Micro PV Anlage

[balkonpv\\_micropv\\_fritz.png](#)

Die Mini-PV Anlage hat keine eigene Messung der Erzeugung, weshalb diese von der FRITZ!DECT gemessen wird. Damit die angesprochene Fritz Box die richtige Steckdose auswählen und als Payload-Information an die Funktion Wirkleistung übergeben kann, ist die AIN der Steckdose zu setzen.

Funktion: BalkonPV

```
msg.payload.ain = "087610221618";  
return msg;
```

# Anbindung IP-Symcon

## Gebäudeautomation

Die **Software IP-Symcon** ist eine Gesamtlösung für die Gebäudeautomation, welche in ihren aktuellen Versionen problemlos via MQTT in Casa Corrently integriert werden kann. Es wird empfohlen, dass IP-Symcon nicht auf demselben Raspberry Pi installiert wird, da der Ressourcenbedarf ansonsten sehr schnell zu hoch werden kann.

Da es Überschneidungen zwischen Casa Corrently und IP-Symcon hinsichtlich der Aufgaben/Funktionen gibt, wird folgende Aufteilung empfohlen:

- Automatisierung => IP Symcon
- Energie Management => Casa Corrently

 unknown

Innerhalb von IP-Symcon wird eine neue Instanz des **MQTT Configurator** erstellt. Dadurch können Variablen innerhalb von IP-Symcon als MQTT Nachricht genutzt werden (Beschreiben mit RequestAction):

```
RequestAction( 37917, $p2);
```

Innerhalb von Casa Corrently wird der durch IP-Symcon bereitgestellt MQTT Broker eingebunden:

 unknown

Der Wert der Variable **\$p2** von Symcon steht dann als **msg.payload** innerhalb von Node-RED zur Verfügung.