

# Grundlagen der Verwendung

**Node-RED** ist eine von der IBM entwickelte Lösung zur Verknüpfung von Hardware Geräten mit Logik und Steuerung. Die Verwendung von Node-RED zeichnet sich durch eine hohe Effizienz und einfache Bedienung des Systems aus. Theoretisch kann man mit diesem System die unterschiedlichsten Aufgabenstellungen, die sich rund um ein Haus/Unternehmen ergeben abbilden und automatisieren. Innerhalb von Casa Corrently werden jedoch lediglich die Elemente, welche für ein Energie-Management relevant sind betrachtet.

- Anbindung von Sensoren (Messgeräte, Stromzähler, IoT Devices, Smart-Home, Appliances)
- Ansteuerung von OpenEMS
- Erstellung von Last- und Erzeugungsprognosen
- Interaktion der verschiedenen Systemkomponenten
- Einfache Anpassung an die jeweiligen Gegebenheiten

## Definitionen

Innerhalb von Node-RED kann im Prinzip mit jeder Art von Daten und Einheit gearbeitet werden. Zur Vereinheitlichung wurden jedoch für Casa Corrently einige Vorgaben gesetzt.

## Verwendung von Watt (Auflösung, Einheit, Dimension)

Obwohl sehr viele Prozesse in der Energiewirtschaft auf Watt-Stunden (bzw. Kilowatt-Stunden) festgelegt sind, wird innerhalb von Casa Corrently versucht immer mit Watt zu arbeiten. Dies ist möglich, da im Hintergrund mit InfluxDB eine zeitreihenbasierte Datenbank zum Einsatz kommt.

Der große Vorteil einer Verwendung von Watt als Einheit ist, dass bei unterschiedlichen Sensoren/Messgeräte diese die Daten asynchron zur Verfügung stellen. Durch Bildung einer Differenz der Zeitstempel zwischen den einzelnen Messwerten kann aus Watt (Leistung) recht leicht in Watt-Stunden (Arbeit) umgerechnet werden.

## Verwendung von WattStunden (nur bei Speicher SoC)

Lediglich bei der Definition von Speichern wird mit Watt-Stunden gerechnet. Hier wird von der sonst üblichen Darstellung des State-Of-Charge (SoC) in Prozentwerten abgewichen, da die Verarbeitung einfacher und nachvollziehbarer ist.

## Verwendung von Ganzzahlen (keine Kommawerte)

Durch die gewählte Auflösung in Watt ist eine ausreichende Genauigkeit vorhanden, um nicht jedem Elektron hinterherzurennen und gleichzeitig den gesamten Stromfluss einer Großstadt

abbilden zu können. Es wird daher auf die Verwendung von Fließkommazahlen verzichtet (Big Integer anstelle von Double oder Float). Dies hat bei modernen 64 Bit Prozessoren zudem den Vorteil, dass mehrere Berechnungen parallel in einem Takt durchgeführt werden können.

## Flows ohne Promise

Zwar erlaubt Node-RED, dass man mit Promise (Asynchronen Funktionen) arbeiten könnte, jedoch steigert dies die Komplexität von Prozessen sehr stark. Bei Casa Corrently kommen daher keine Promise in den Funktionen zum Einsatz. Stattdessen werden die Flos so definiert, dass sie entweder durch einen Sensor getriggert oder durch einen `Inject Node` (Zeitschalter) von Anfang bis zum Ende ein Ereignis abarbeiten.

## Dashboards in Grafana

Zwar bietet Node-RED mit `UI` eine Möglichkeit zur Visualisierung von Daten, jedoch wird diese bei Casa Corrently nicht verwendet. Sobald Daten visualisiert werden sollen, werden diese in die InfluxDB gespeichert, wovon sie mittels Grafana visualisiert werden können.

Eine Ausnahme sind hier die Einstellungen, welche in den einzelnen Flows verwendet werden. Diese vom Anwender zu setzenden Informationen für die Anpassung des Systems werden mit der Node-RED UI umgesetzt.

### Beispiel: Setzen der Postleitzahl für den GrünstromIndex

Im Flow

`settings.png` or type unknown

Innerhalb der UI

`settings_ui.png` or type unknown

## Nutzung des Node Status

Da innerhalb von Node-RED auf die Verwendung der Visualisierung (Dashboard) verzichtet wird, jedoch das Ergebnis von Anpassungen und Konfigurationen möglichst schnell sichtbar sein soll, werden die Status Elemente (grauer Text unter den Funktionen) genutzt.

`node_status.png` or type unknown

## MQTT für Sensoren

Auf einem Casa Corrently System läuft ein eigener MQTT Broker (`Mosquitto`) an Port 1883. Geräte und Sensoren, welches dieses Protokoll unterstützen können diesen für Nachrichten nutzen.

`mqtt.png` or type unknown

Im Bild wird innerhalb des Flows direkt auf den lokalen MQTT Broker zugegriffen, sobald eine neue Meldung des Stromspeichers vorhanden ist wird der Speicherfüllstand in Watt-Stunden umgerechnet.

## Flows für Ebenen und Fahrplanmanagement

Auf einem Flow wird entweder eine **Ebene** oder ein Fahrplan (Gerät) abgebildet. Auf Sub-Flows wird bislang verzichtet.

[ebenen\\_tabs\\_flows.png](#)  
image not found of type unknown

Zusätzlich existiert ein Flow/Tab, welcher zur Gesamtsteuerung dient.

---

Revision #3

Created 20 July 2020 00:36:03 by Thorsten Zoerner

Updated 20 July 2020 01:28:19 by Thorsten Zoerner