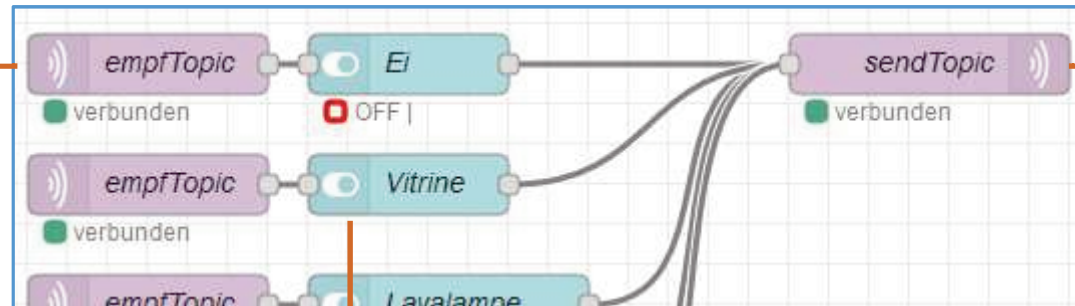


# NodeRed – MQTT Aufbau für die NodeRed Gui



Empfange Status des Gerätes.

NodeRed Button erzeugt Topic und Payload und zeigt den Status des Gerätes an.

Sendet den entsprechenden Topic und Payload des angeklickten Buttons

mqtt in Node bearbeiten

Löschen Abbrechen Fertig

Properties

Server: alles empfangen

Topic: stat/light/straussenei/livingroom/POWER

QoS: 2

Output: auto-detect (string or buffer)

Name: empfTopic

switch Node bearbeiten

Löschen Abbrechen Fertig

Properties

Size: 6 x 2

Label: Ei

Tooltip: optional tooltip

Icon: Custom

On Icon: lightbulb\_outline Colour: yellow

Off Icon: lightbulb\_outline Colour: grey

Pass through msg if payload matches new state:

Indicator: Switch icon shows state of the input

When clicked, send:

On Payload: ON

Off Payload: OFF

Topic: cmd/light/straussenei/livingroom/POWER

mqtt out Node bearbeiten

Löschen Abbrechen Fertig

Properties

Server: alles empfangen

Topic: Topic

QoS: Retain

Name: sendTopic

Ei

Lavalampe

Salzlampe

Leselampe

Vitrine

HOME

Ausschnitt NodeRed mit Rückmeldung, siehe auch Folgeseite

## MQTT Einstellung im Gerät (hier Sonoff Steckdose mit Tasmota)

### Arten der Prefixe in Tasmota fest eingestellt

<b>cmnd</b>	Geräte (Laptop, Handy) steuern
<b>stat</b>	Status- oder Konfigurationsmeldung zurückmelden
<b>tele</b>	unaufgeforderte Telemetrie-Informationen in regelmäßigen Abständen melden. (bspw. Temp., ...)

Das MQTT-Format kann bei jedem unterschiedlich sein, je nachdem wie seine Logik (Gruppierung) aufgebaut ist und was man damit machen will. Bsp.:

- prefix/Ort/Typ/Gerät „cmnd/wz/licht/leselampe“ „ON“ man kann es dann erweitern nach dem Prinzip „cmnd/wz/\*“ „OFF“ schaltet alle Geräte im Wohnzimmer aus, oder eine andere Logik wäre
- prefix/Typ/Gerät/Ort „cmnd/licht/leselampe/wz“ „ON“, hier ist die erweiterte Logik dann „cmnd/licht/\*“ „OFF“ schalte das Licht in allen Räumen aus.

Ausschnitt der Tasmota-Einstellung.

The screenshot shows a configuration window for MQTT settings. It contains the following text:

```
Topic = %topic% (sonoff)  
light/straussenei/livingroom  
Full Topic (%prefix%/%topic%/  
%prefix%/%topic%/
```

An orange box highlights the value 'light/straussenei/livingroom' in the Topic field. An arrow points from this box to the 'Full Topic' field, which contains the template '%prefix%/%topic%/'. A second arrow points from the left side of the page to the 'Full Topic' field.

# MQTT Einstellung im Gerät (hier Sonoff Steckdose mit Tasmota)

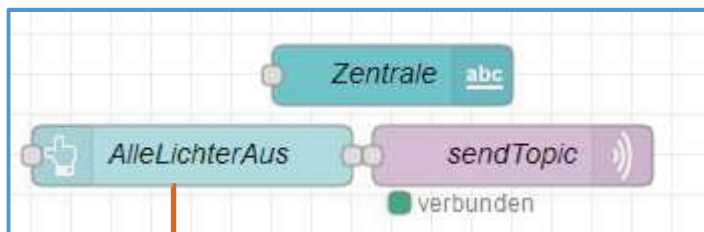
## Der GroupTopic

Mit dem Group Topic kann ich, wie der Name schon sagt 😊, ganze Gruppen mit einem Befehl ein-/ausschalten.

Hier meine Group Topic Einstellung in allen „Licht“-Geräten. (die Sonoff's)

<b>MQTT Topic</b>	light/straussenei/livingroom
<b>MQTT Group Topic</b>	grplight/POWER
<b>MQTT Full Topic</b>	cmnd/light/straussenei/livingroom/
<b>MQTT Full Topic</b>	cmnd/light/straussenei/livingroom/

Auf der NodeRed-Seite sieht es so aus. Ein Button für alle Lichter aus.



Das sind die Werte, die der Button sendet

When clicked, send:

Payload

Topic

## Sepia Varianten

Ich nehme mal an, dass der MQTT-Integration in Sepia ähnlich dem sein wird, wie das Beispiel beim MQTT Sepia-Service. Daher nehme ich auch den Input, den ich auf der NodeRed Seite erhalte als Ausgangspunkt.

Zuerst einmal zu Deiner Frage mit den weiteren Infos im Objekt.

- Ja, es sollte mindestens der Raum mit drin sein, sonst weiß NodeRed nicht welches Gerät angesprochen werden soll. NodeRed weiß ja im Prinzip nichts. Es bekommt einen String und einen Wert (Topic und value) und soll damit irgendetwas machen. Dein Aufbau momentan ist:

*sepia/smart-devices/ + [selbst definiert]*)

[selbst definiert] = Licht?? :

„Licht im Wohnzimmer auf 70%“ und „Licht im Schlafzimmer auf 70%“ ergibt bei beiden

*sepia/smart-devices/Licht { „state“: 70}* nur welches Licht soll NodeRed nehmen/steuern?

Dann gibt es, zumindest beim Service, folgende Varianten die in NodeRed ankommen.

Ist schon relevant. Beim 1. Bsp. ist *type* wichtig bei den anderen beiden Beispielen *valueType* und *value* zusammen. Beachte: 70% <> 70 Grad bzw. 23% <> 23 Grad ;-)  
Was ich meine ist, nur den *value*-Wert zu nehmen, kann zu Fehlinterpretation führen.

„Schalte das Licht im Wohnzimmer ein“

```
▼ action: object
  valueType: "number_plain"
  type: "on"
  value: ""
```

„Licht im Wohnzimmer auf 70%“

```
▼ action: object
  valueType: "number_percent"
  type: "set"
  value: "70"
```

„Stelle die Heizung im Wohnzimmer auf 23 Grad“

```
▼ action: object
  valueType: "number_temperature"
  type: "set"
  value: "23"
```