

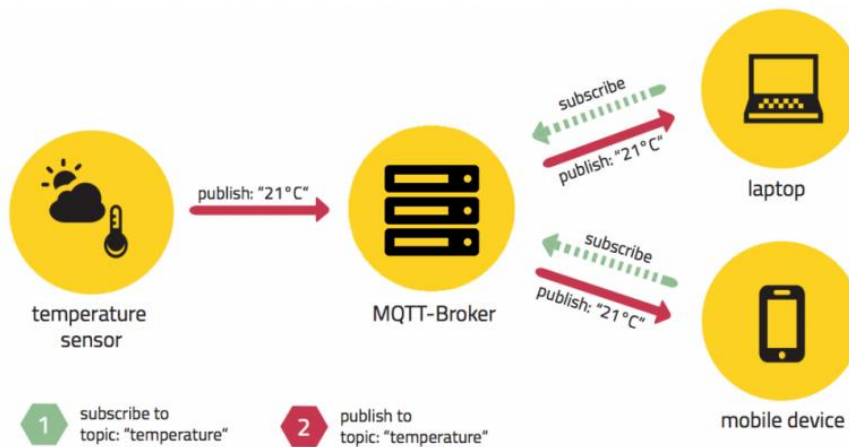


Verwendung von MQTT mit Thingworx

Das **Message Queue Telemetry Transport** Protokoll, kurz MQTT (früher MQ Telemetry Transport), ist ein ursprünglich von IBM entwickeltes leichtgewichtiges Kommunikationsprotokoll für M2M-Kommunikation im Internet of Things auf Anwendungsebene.

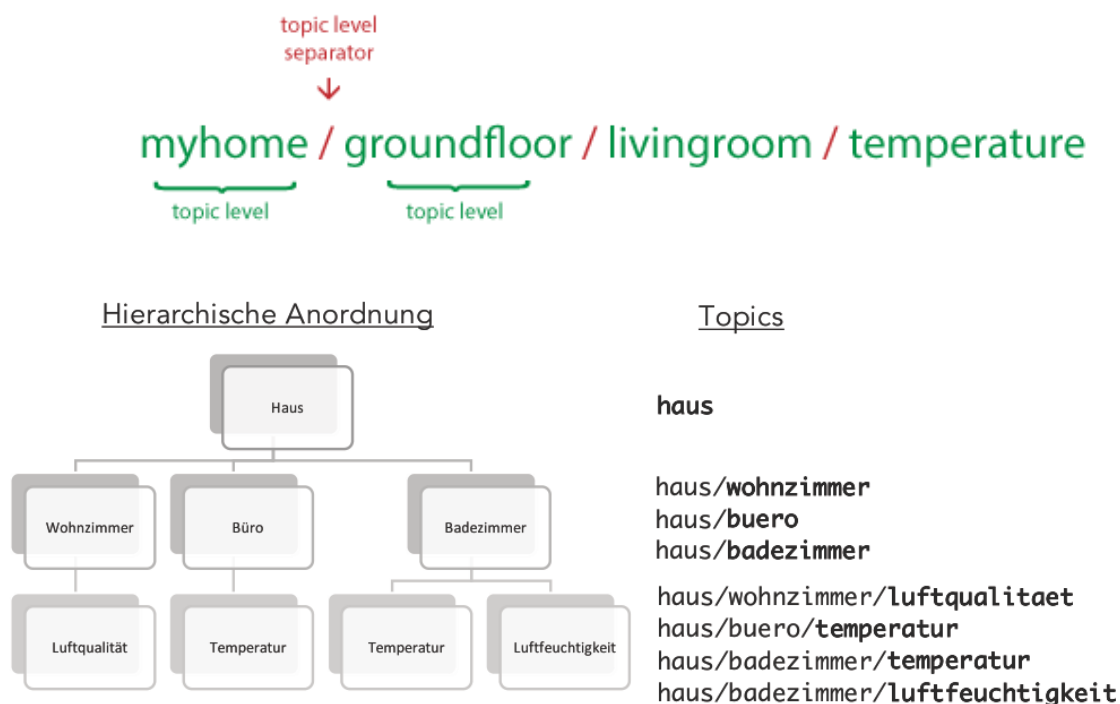
MQTT wurde bereits im Jahr 1999 von IBM-erfunden, um zu der damaligen Zeit Öl-Pipelines über eine Satelliten-Verbindung miteinander zu vernetzen.

Um mit MQTT arbeiten zu können ist ein sogenannter **MQTT-Server** bzw. **MQTT-Broker** erforderlich.



Im gesamten System gibt es nur einen **Broker(Server)** alle anderen sind dann Clients. Die **Clients** können sowohl Nachrichten senden (**publish**) als auch Nachrichten empfangen (**subscribe**).

Die verschiedenen Nachrichten (z.B. Temperaturwerte) werden mit sogenannten **Topics** eingeordnet:



NÜTZLICHE SOFTWARE

CLIENTS

- **PC:** MQTT.fx. Letzte Gratisversion ist 1.7.1!
- **IOS:** MQTTTool <https://apps.apple.com/de/app/mqtttool/id1085976398>
- **Android:** MQTT Client <https://play.google.com/store/apps/details?id=in.dc297.mqttclpro>

Bei allen Clients muss unter anderem auch eine **Client_ID** angegeben werden. Diese darf **nicht doppelt** verwendet werden! (Beim RC-Car wird als Client_ID die MAC-Adresse des WLAN-Moduls verwendet)

BROKER

(Diese Einstellungen werden in Thingworx und in den Client-Apps benötigt)

MQTT Server in Wiener Neustadt:

URL: letto.htlwrn.ac.at

Ports:

- 5003 (SSL verschlüsselt)
- 1883 (unverschlüsselt)

User: iot2021

Pass: iot2021

Dieser User hat nur Schreib- und Leserechte auf alle SubTopics von **edu/iot2021/**

HiveMQ Server (Public MQTT Broker)

URL: broker.hivemq.com

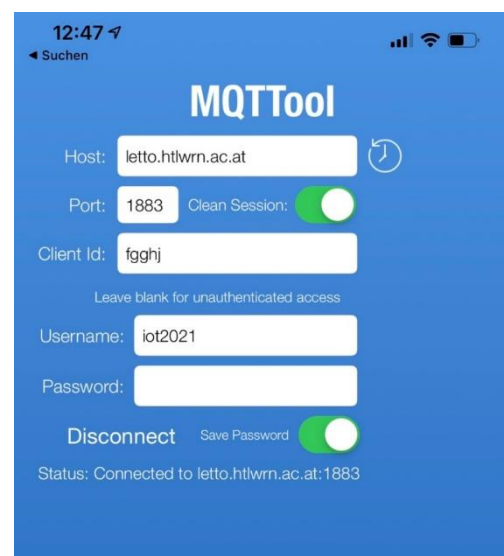
Ports:

- 1883 (unverschlüsselt)

User:

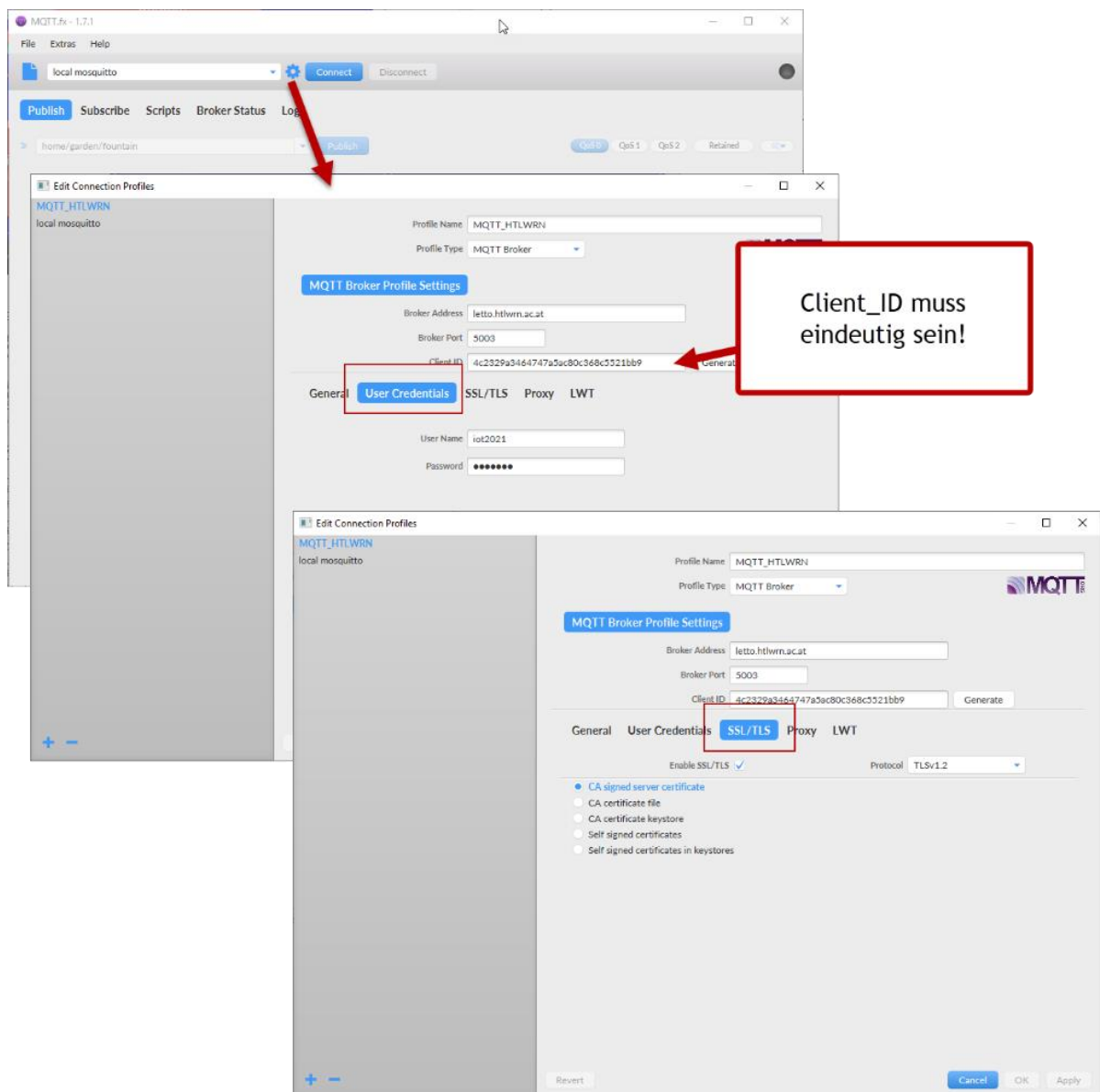
Pass:

Alle User haben überall Schreib- und Leserechte!



EINSTELLUNGEN BEI MQTT.FX

(mit SSL)



ERSTE VERSUCHE MIT MQTT

Publish: edu/iot2021/**SKZ**/Temperatur

SKZ = z.B. **304417**

Subscribe: edu/iot2021/#

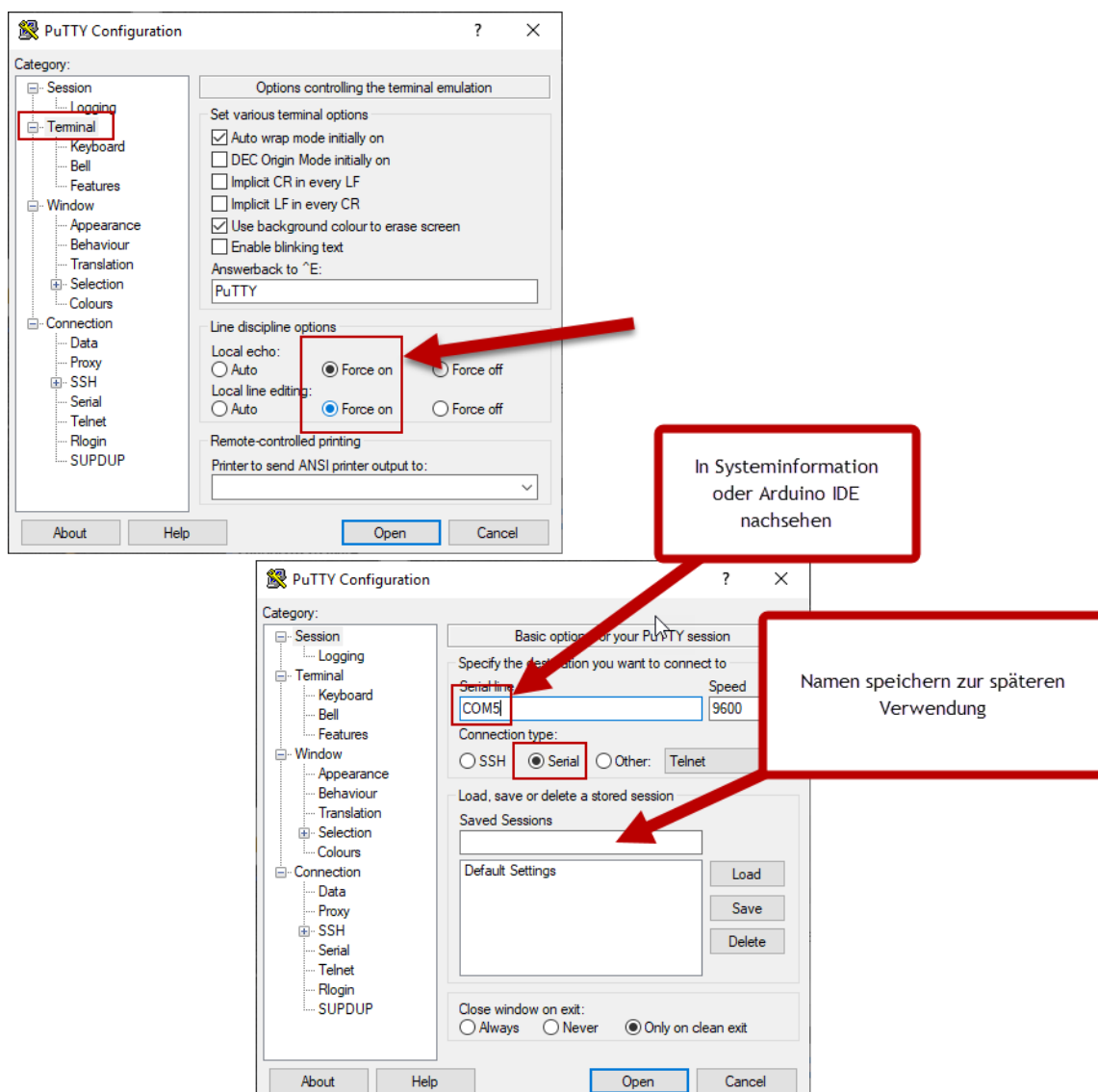
- ist ein Wildcard für alle untergeordneten Topics.

Man sieht somit die Nachrichten aller Teilnehmer.

VORBEREITUNG RC-CAR

Aufspielen der neuen Firmware auf den MKR1010 (aktuelle Version ist 1.1) über die Arduino IDE.

Mittels Putty die Einstellungen für die Verbindung vornehmen:



```

PuTTY (inactive)
*****
*
* DigiPro RC-Car MQTT Connection Setup (c) 2021 FIASUM
*
* ATTENTION use PUTTY or KITTY
* --> to see entered chars turn on echo!!!
* --> confirm every value with <ENTER>
* --> backspace and del ar not supported
*****

* Topic for subscription will get */cmd/#
* Topic for publish will get */data
* --> Example: Topic entered is edu/iot2021
* --> subscription will be edu/iot2021/cmd/#
* --> publish will be edu/iot2021/data
*****

Insert SSID:
Schubi

Insert Wifi Password:
[redacted]

Insert Broker:
letto.htlwrn.ac.at

Insert Port:
1883

Insert MQTT-Topic:
edu/iot2021

Insert MQTT-User:
iot2021

Insert MQTT-Pass:
iot2021

Settings stored !

SSID: |Schubi|
WifiPWD: |[redacted]|
Broker: |letto.htlwrn.ac.at|
Port: |1883|
Topic: |edu/iot2021|
MQTT-Topic: |iot2021|
MQTT-Pass: |iot2021|
Attempting to connect to SSID: Schubi
SSID: Schubi
IP Address: 192.168.0.99
Signal strength (RSSI):-58 dBm
Attempting MQTT connection...connected
536903540
[ACS712]
|   v_value = 1559.60
|   Current = -10.35

[HC-020K]
|   RPM = 0

```

Pub und Sub wird ergänzt!
(Hardcoded)

Vertikale Linien
Kennzeichen Stringende -
keine Leerzeichen!

Client_ID wird
automatisch erzeugt

m.schubert/a.fischer 2021

Einstellungen am TWX-Server:


MQTT-Thing erzeugen:

Thing: 304417_IoTSeminar_2021_Mqtt

To Do Save Cancel More

General Information Properties and Alerts Services Events Subscriptions Configuration Permissions Change History View Relationships

General Information

 **Name**
304417_IoTSeminar_2021_Mqtt

[Change](#)

Description

Project (required)
304417_IoTSeminar_2021 [Set as project context](#)

Tags
Search Model Tags

Base Thing Template
MQTT

Implemented Shapes
Search Thing Shapes

Value Stream
Search Value Streams

☒ **Active**

☐ **Published**

Identifier

Thing: 304417_IoTSeminar_2021_Mqtt

To Do Save Cancel More

General Information Properties and Alerts Services Events Subscriptions Configuration Permissions Change History View Relationships

Configuration

Mappings

| Actions | subscribe | publish | name | topic |
|---------|-----------|---------|------|-------|
| No data | | | | |

MQTT Settings

clientIdFormat
304417_SUM_01

☒ **use SSL**

serverName
lets.tdum.ac.at

serverPort
5003

uriPath
/MQTT

password

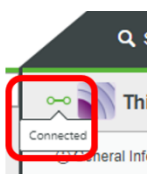
timeout
5000

connectTimeout
10000

retryInterval
30000

Muss eindeutig sein !

Save!



← Erfolgreiche Verbindung zum MQTT-Broker

TOPIC SUBSCRIBE

Configuration → Mappings → Add+

304417_IoTSeminar_2021_HiveMQTT:
Mappings

Mappings (3)
☒ subscribe
☐ publish

name ⓘ

topic

Set Cancel

Property erstellen - Name muss gleich lauten wie in Mappings!

Properties → My Properties → Add+

Properties Alerts
Filter
Choose category

My Properties
Add Duplicate Delete Manage Bindings Refresh

| Name | Actions | Source | Default Value | Value | Alerts | Category | Additional Info |
|-------------------------------------|---------|--------|---------------|-----------------------------|--------|----------|-----------------|
| <input type="checkbox"/> json_rccar | | | | { "Hum": 46.4, ... more + } | 0 | | |

Save! – Wenn Daten gesendet werden sollte nun das JSON lesbar sein:

Mit more kann der gesamte Inhalt angezeigt werden.

View Content

```

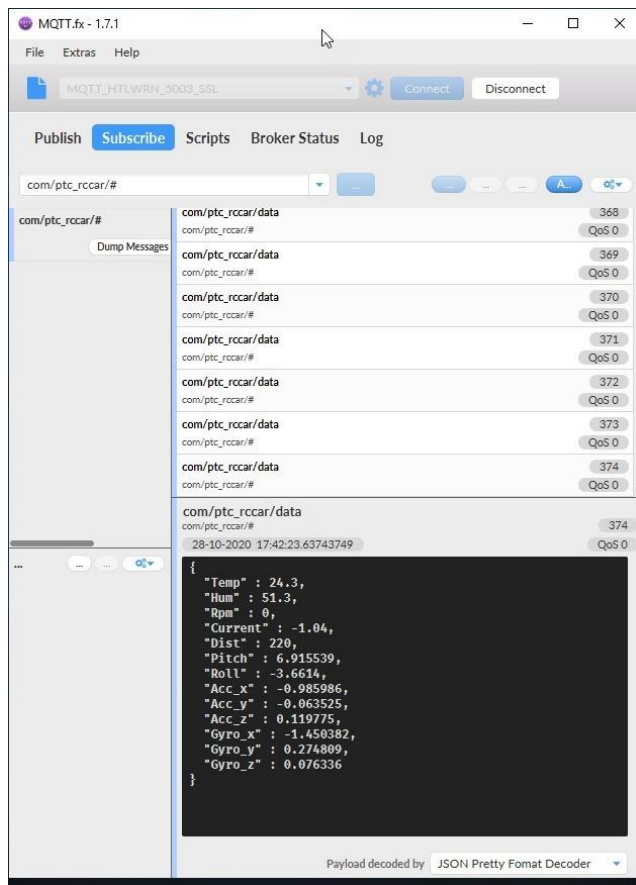
{
  "Hum": 46.4,
  "Dist": 81,
  "Gyro_z": 6.442748,
  "Rpm": 1125,
  "Gyro_x": -58,
  "Gyro_y": -0.473282,
  "Temp": 24.5,
  "Pitch": 19.72529,
  "Acc_x": -1.12287,
  "Acc_y": -0.067871,
  "Roll": -3.259972,
  "Current": -10.62,
  "Acc_z": 0.402832
}

```

Done

m.schubert/a.fischer 2021

Datenanzeige unter MQTT.fx



WEITERVERARBEITUNG DER DATEN

Aus dem JSON können nun die relevanten Daten in entsprechende Propertyts kopiert werden. Die professionelle Methode wäre über Datashapes und Infotables. Ich habe mich für die Maschinenbauvariante entschieden 😊

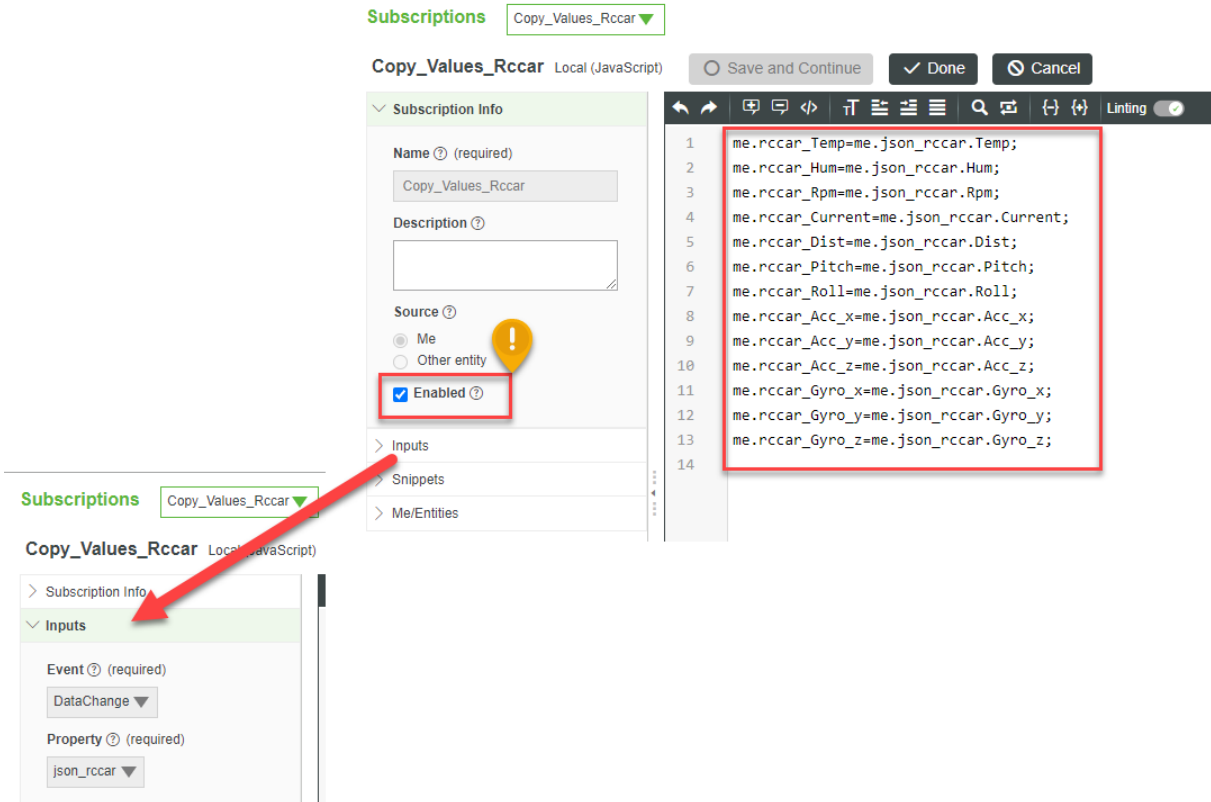
Anlegen der Propertyts:

Auch gleich anlegen, werden für publish verwendet!

| Name | Actions | Source | Default Value | Value | Alerts | Category | Additional Info |
|--|---------|--------|---------------|-------------------------|--------|----------|-----------------|
| <input type="checkbox"/> json_rccar | | | | { "Hum": 46.4, ... more | 0 | | |
| <input type="checkbox"/> # rccar_Acc_x | | | | -1.12207 | + | 0 | |
| <input type="checkbox"/> # rccar_Acc_y | | | | -0.067871 | + | 0 | |
| <input type="checkbox"/> # rccar_Acc_z | | | | 0.402832 | + | 0 | |
| <input type="checkbox"/> # rccar_Current | | | | -10.62 | + | 0 | |
| <input type="checkbox"/> # rccar_Dist | | | | 81 | + | 0 | |
| <input type="checkbox"/> # rccar_Gyro_x | | | | -50 | + | 0 | |
| <input type="checkbox"/> # rccar_Gyro_y | | | | -0.473282 | + | 0 | |
| <input type="checkbox"/> # rccar_Gyro_z | | | | 6.442748 | + | 0 | |
| <input type="checkbox"/> # rccar_Hum | | | | 46.4 | + | 0 | |
| <input type="checkbox"/> # rccar_Pitch | | | | 19.72529 | + | 0 | |
| <input type="checkbox"/> 123 rccar_pwm_fan | | | | 1 | + | 0 | 0 to 1 |
| <input type="checkbox"/> 123 rccar_pwm_led | | | | 0 | + | 0 | 0 to 1 |
| <input type="checkbox"/> # rccar_Roll | | | | -3.259972 | + | 0 | |
| <input type="checkbox"/> # rccar_Rpm | | | | 1125 | + | 0 | |
| <input type="checkbox"/> # rccar_Temp | | | | 24.5 | + | 0 | |

m.schubert/a.fischer 2021

Danach eine Subscription erzeugen:



Subscriptions Copy_Values_Rccar ▼

Copy_Values_Rccar Local (JavaScript) Save and Continue Done Cancel

Subscription Info

Name (required)
Copy_Values_Rccar

Description

Source

☐ Me

☐ Other entity

☒ Enabled

Inputs

Event (required)
DataChange ▼

Property (required)
json_rccar ▼

```

1 me.rccar_Temp=me.json_rccar.Temp;
2 me.rccar_Hum=me.json_rccar.Hum;
3 me.rccar_Rpm=me.json_rccar.Rpm;
4 me.rccar_Current=me.json_rccar.Current;
5 me.rccar_Dist=me.json_rccar.Dist;
6 me.rccar_Pitch=me.json_rccar.Pitch;
7 me.rccar_Roll=me.json_rccar.Roll;
8 me.rccar_Acc_x=me.json_rccar.Acc_x;
9 me.rccar_Acc_y=me.json_rccar.Acc_y;
10 me.rccar_Acc_z=me.json_rccar.Acc_z;
11 me.rccar_Gyro_x=me.json_rccar.Gyro_x;
12 me.rccar_Gyro_y=me.json_rccar.Gyro_y;
13 me.rccar_Gyro_z=me.json_rccar.Gyro_z;
14

```

Die Bezeichnungen auf der rechten Seite richtet sich dabei nach dem Aufbau des JSON:

```

{
  "Temp" : 24.3,
  "Hum" : 51.3,
  "Rpm" : 0,
  "Current" : -1.04,
  "Dist" : 220,
  "Pitch" : 6.915539,
  "Roll" : -3.6614,
  "Acc_x" : -0.985986,
  "Acc_y" : -0.063525,
  "Acc_z" : 0.119775,
  "Gyro_x" : -1.450382,
  "Gyro_y" : 0.274809,
  "Gyro_z" : 0.076336
}

```