easy MQTT

MQTT bietet viele Möglichkeiten, die gerade am Anfang nicht einfach zu durchschauen sind.

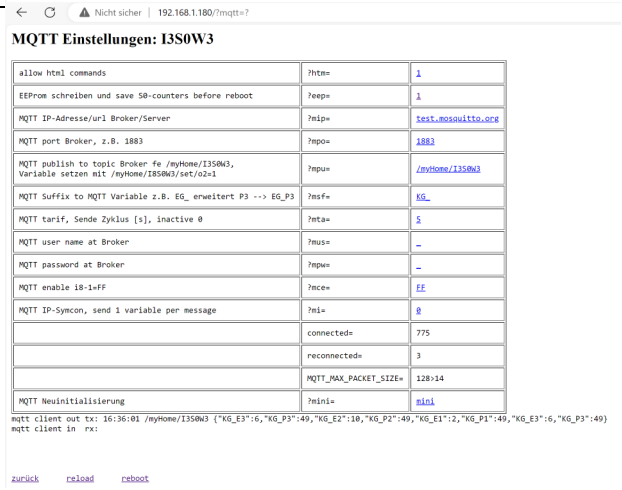
Immer wird benötigt:

a) ein MQTT-Client, das ist ein IPswitch, der seine Messwerte an einen
b) MQTT-Server (wird MQTT-Broker genannt) sendet und
c) ein weiterer MQTT-Client, wie der IPswitch „MQTT-Diagramm“ oder eine App, wie das IoT MQTT-Panel oder MQTT-Dashboard unter Android oder mit dem MQTT-Explorer unter WIN. Der Client abonniert beim Broker ein topic (Directory) und bekommt die Messdaten daraus gesendet.

Im einfachsten Fall nutzt man einen öffentlichen MQTT-Broker ohne Registrierung, wie http://test.mosquitto.org . Soll nun ein IPswitch Verbrauchswerte senden (?im=1000 und ?na=), wäre im IPswitch-3xS0-WiFi-3 unter ?mqtt=? einzutragen:

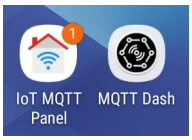
allow html commands	?html=1
EEProm schreiben und save S0-counters before reboot	?eep=1
MQTT Broker ip/url	?mip=test.mosquitto.org
MQTT Broker Port	?mpo=1883
MQTT publish to topic Broker	?mpu=/myHome/I3S0W3
MQTT Suffix to MQTT Variable z.B. EG_ erweitert P3 --> EG_P3	?msf=KG_
MQTT Broker Tarif (Sendezyklus[s])	?mta=5
MQTT user name at Broker	?mus=
MQTT password at Broker	?mpw=
MQTT enable i8-1	?mce=7

mit einem ?mini=1 wird das mqtt im IPswitch neu initialisiert und in der Fußleiste können die vom IPswitch an den Broker gesendeten Telegramme beobachtet werden.

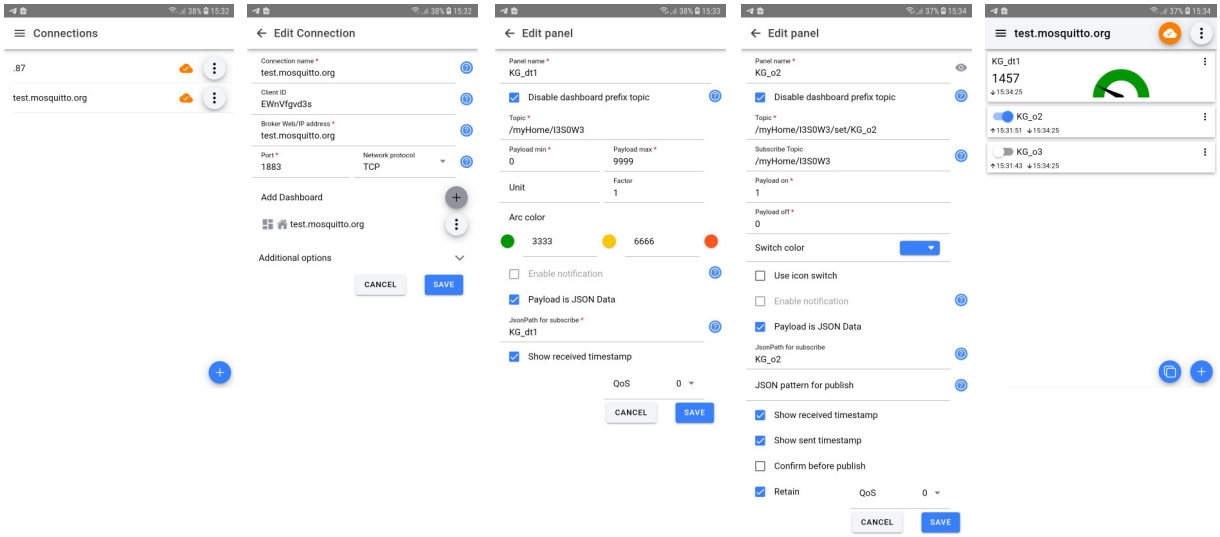


Wenn sich nun ein anderer MQTT-Client beim Broker mit der mpu anmeldet, dann empfängt er die gesendeten Telegramme des IPswitch-3xS0-WiFi-3.

Folgend zeigen wir dies mit dem „IoT MQTT-Panel“ und dem „MQTT-Dash“ unter Android. Wir wollen mit dem Smartphone empfangene Daten darstellen und auch einen Ausgang am IPswitch schalten, dazu ist am I3S0W3 einzugeben ?im=-1 und ?na=

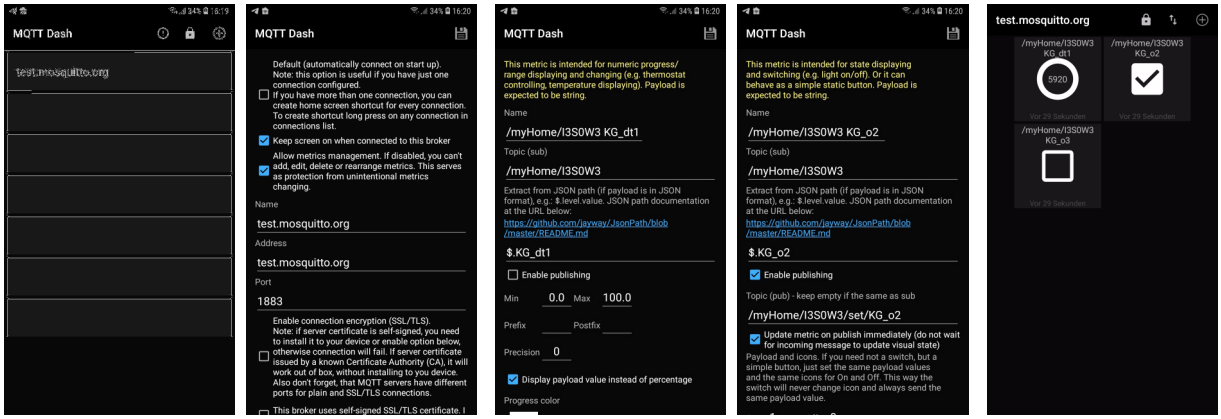


Das „IoT MQTT-Panel“ ist zu installieren und folgend zu konfigurieren:



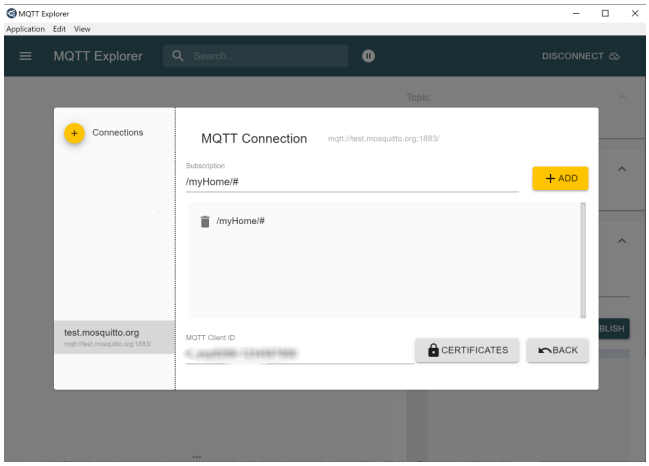
Der Ausgang am I3S0W3 lässt sich nun mit dem Handy schalten.

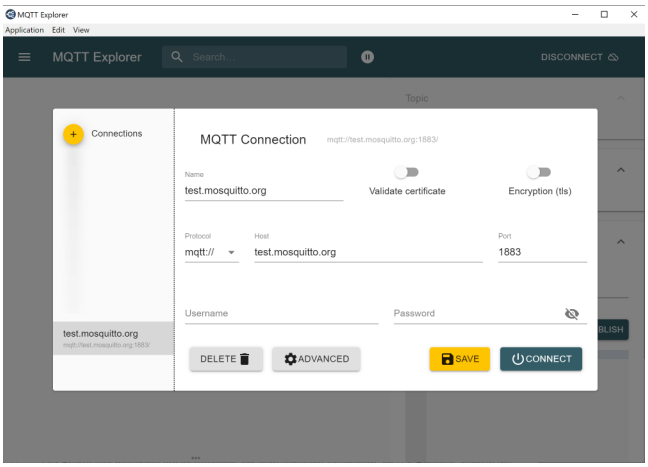
Das „MQTT-Dash“ ist zu installieren und folgend zu konfigurieren:

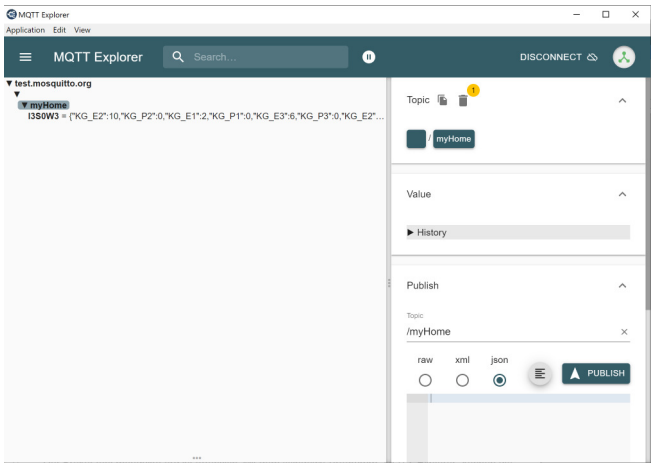


Auch hier lässt sich nun der Ausgang am I3S0W3 mit dem Handy schalten.

Anstelle der Apps oder auch zusätzlich können die Daten mit dem Windows-Programm „MQTT-Explorer“ vom Broker mit dem topic /myHome/# dargestellt werden.







Bitte beachten, alle Nutzer des Brokers test.mosquitto.org können die Daten unter /myHome/# einsehen und mit der Publish-Funktion auch schalten.

Mit dem topic /# werden alle (!!) Telegramme von test.mosquitto.org gelistet. Deshalb sollte ein weniger öffentlicher Broker für einen Dauerbetrieb genutzt werden.

Alternativ kann auch unser lokaler MQTT-Broker anstelle eines externen genutzt werden:

https://www.sms-guard.org/downloads/easy-MQTT-Broker-Anleitung.pdf

Der Vorteil ist, die Daten bleiben im lokalen Netz und die Einrichtung und Pflege eines eigenen Brokers entfällt. Außerdem verbraucht der easy-MQTT-Broker sehr viel weniger Strom als ein Server.

Whom ever may concern: zum Aufbau des topics vom 29.05.25:

die mpu (mqtt publish == topic) sollte mit Bedacht gewählt werden und ist auch eine Frage der Struktur:

1. Zeichen immer /
wenn man mehrere Immobilien verwaltet die Ortskennung, z.B. /S25 für Stinthorn 25 der Installationsort, z.B. /S25/KG für Kellergeschoß der Typ des IPswitch, z.B. /S25/KG/I8S0L

Für die Inbetriebnahme einen mta=10 (mqtt tarif ==Sendeabstand[s]) nehmen und für den Dauerbetrieb 300, da sonst riesige Datenmengen in der Datenbank entstehen.

Wir verwenden als Broker mosquitto auf einem Raspberry Pi mit node-red, influxdb und grafana. Am RaPi können Sie alle Telegramme sehen mit:

mosquitto_sub -v -t /# | ts

oder hier mit mi=0 (mqtt ipsymcon, 0: mehrere Variablen in einem Telegramm, 1: eine Variable pro Te-

legramm):

mosquitto_sub -v -t /KG/I8S0L/# | ts

May 29 07:32:16 /KG/I8S0L
{"E1":0,"P1":0,"E2":0,"P2":0,"E3":0,"P3":0,"E4":0,"P4":0,"i5":0,"cn5":0,"dt5":1189,"i6":0,"cn6":0,"dt6":1189,"o7":1,"cn7":4,"dt7":1121,"o8":1,"cn8":4,"dt8":1123}

Die Telegramme werden in Abhängigkeit der Variablenlängen zusammen gesetzt (E bis zu 20 Stellen, P bis zu 10) mit den vorgegebenen Bezeichnungen, wie E1, da sonst in den nachgeschalteten Systemen der Wartungsaufwand unnötig steigt. Die Namensgebung mit alias erfolgt hier erst in grafana.

Mit:
mosquitto_sub -v -t /+/I8S0L/# | ts

May 29 07:38:39 /KG/I8S0L
{"E1":0,"P1":0,"E2":0,"P2":0,"E3":0,"P3":0,"E4":0,"P4":0,"i5":0,"cn5":0,"dt5":1574,"i6":0,"cn6":0,"dt6":1574,"o7":1,"cn7":4,"dt7":1506,"o8":1,"cn8":4,"dt8":1508}
May 29 07:38:39 /poweron/I8S0L/I8S0L {"name":"I8S0L","topic out":"/KG/I8S0L","topic in":"/KG/I8S0L/set","model":"m4-14a000,May 28
2025 16:11:50,FC:F5:C4:3E:5A:50","myIP":"192.168.1.171"}

kommt auch das poweron Telegramm mit vielen Infos zum IPswitch. Zu Wartungszwecken rufen wir das immer zu erst auf und sehen so alle maßgeblichen Einstellungen.