



## easy MQTT

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

### Immer wird benötigt:

- ein MQTT-Client, das ist ein IPswitch, der seine Messwerte an einen
- MQTT-Server (wird MQTT-Broker genannt) sendet und
- 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.

MQTT Einstellungen: I3S0W3

allow html commands	?html=	1
EEProm schreiben und save S0-counters before reboot	?eep=	1
MQTT IP-Adresse/url Broker/Server	?mip=	test.mosquitto.org
MQTT port Broker, z.B. 1883	?mpo=	1883
MQTT publish to topic Broker fe /myHome/I3S0W3, Variable setzen mit /myHome/I3S0W3/set/02-1	?mpu=	/myHome/I3S0W3
MQTT Suffix to MQTT Variable z.B. EG_ erweitert P3 --> EG_P3	?msf=	KG_
MQTT tarif, Sende Zyklus [s], inactive 0	?mta=	5
MQTT user name at Broker	?mus=	-
MQTT password at Broker	?mpw=	-
MQTT enable i8-1+FF	?mce=	EE
MQTT IP-Symcon, send 1 variable per message	?mi=	0
	connected=	775
	reconnected=	3
	MQTT_MAX_PACKET_SIZE=	128*14
MQTT Neuinitialisierung	?mini=	1

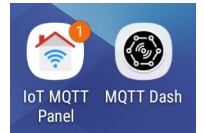
mqtt client out tx: 16:36:01 /myHome/I3S0W3 ("KG\_E3":6,"KG\_P3":49,"KG\_E2":10,"KG\_P2":49,"KG\_E1":2,"KG\_P1":49,"KG\_E3":6,"KG\_P3":49)  
mqtt client in rx:

[zurück](#) [reload](#) [reboot](#)

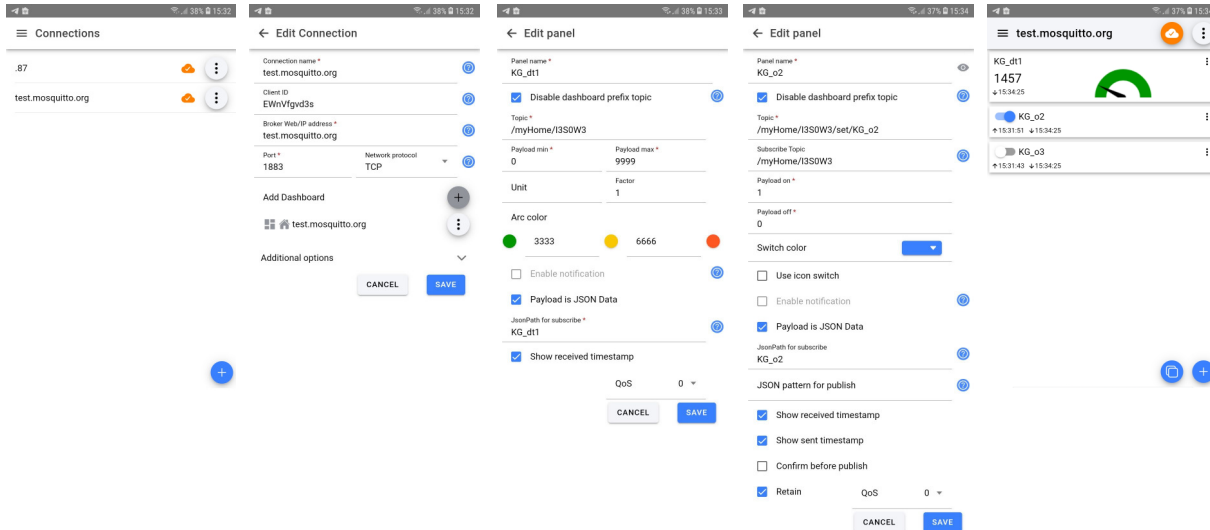
Wenn sich nun ein anderer MQTT-Client beim Broker mit der mpu anmeldet, dann empfängt er die gesendeten Telegramme des IPswitch-xS0-WiFi-3.



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=

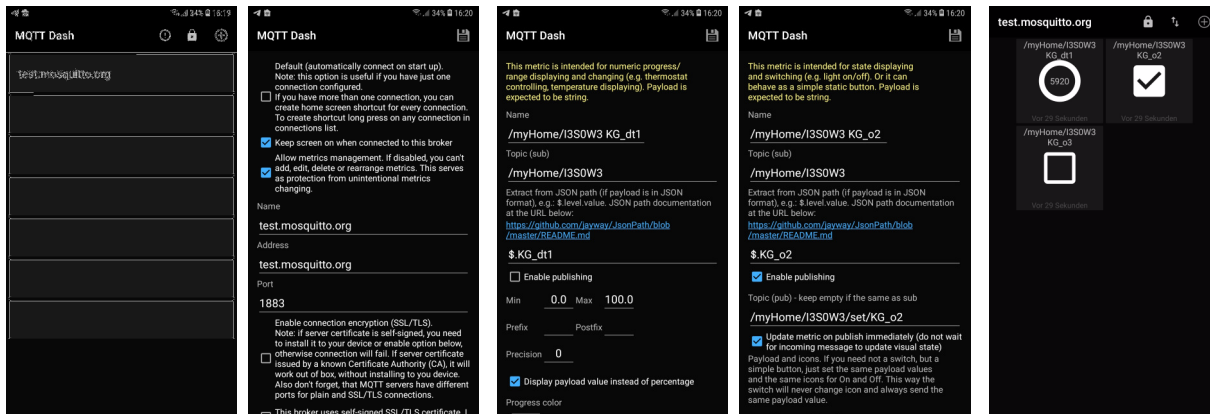


Das „IoT MQTT-Panel“ ist zu installieren und folgend zu konfigurieren:



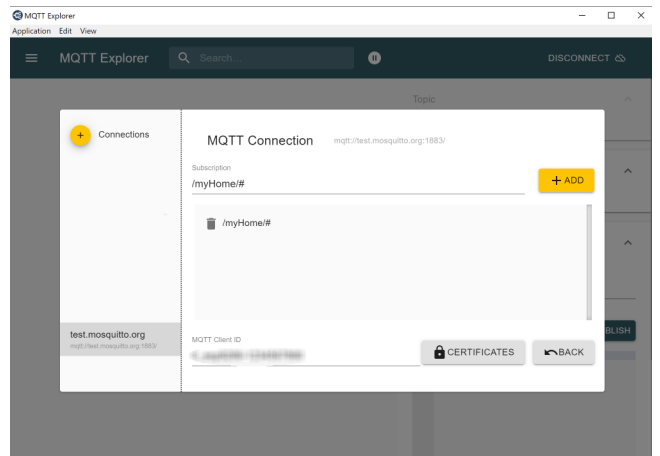
Der Ausgang am I3S0W3 lässt sich nun mit dem Handy schalten.

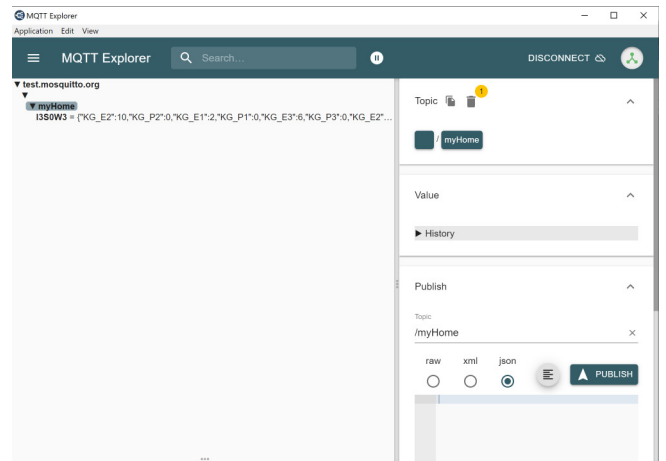
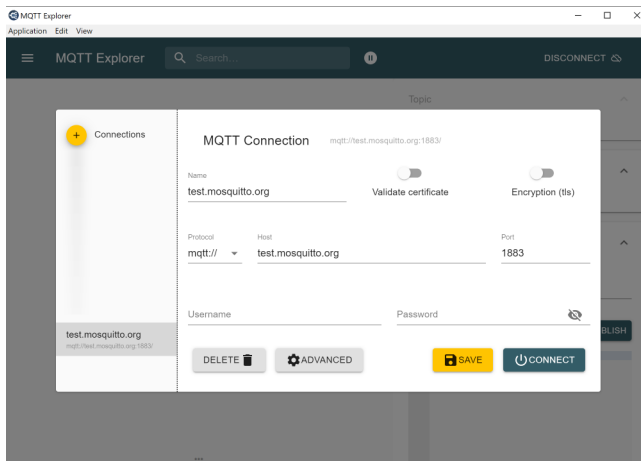
Das „MQTT-Dash“ ist zu installieren und folgend zu konfigurieren:



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.