

## Svxlink via IoT MQTT Panel (Android) steuern



Wer die SvxLink-Steuerung im Dashboard des DJSpot zu kompliziert findet, oder auf einem Smartphone zu fummelig, oder wer den Konfigurationsaufwand beim Anlegen neuer Buttons zum Wechseln der TalkGroups (TG) scheut, der sollte mal einen Blick auf MQTT werfen. Sozusagen als Ergänzung und Bonus zum [MQTT Tutorial für den SVXCube](#) findet ihr nachfolgend eine Anleitung für die Konfiguration eines MQTT Clients für iOS. Eine Anleitung für Android gibt es [hier](#).

### Voraussetzungen:

- DJSpot mit installiertem und gestarteten MQTT-Broker ([Anleitung hier](#))
- DJSpot mit installiertem und gestarteten Node-RED ([Anleitung hier](#))

ODER

- das Image von Jens (hier ist ein MQTT-Broker und Node-RED bereits installiert!)
- Android Gerät mit installierter IoT MQTT-Panel

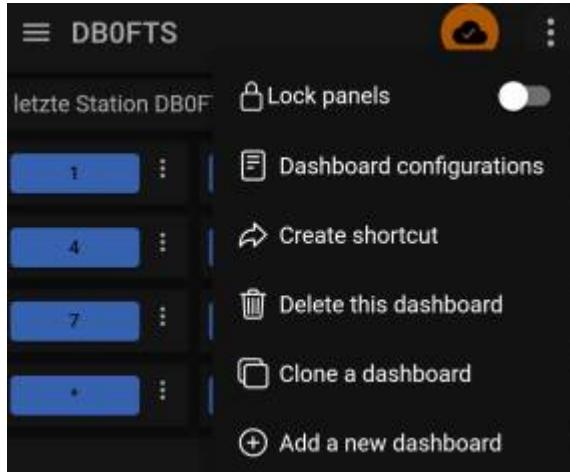
Die nachfolgende Anleitung beschreibt die MQTT-Konfiguration in der App IoT-MQTT (findet ihr im PlayStore). Die kostenlose Variante funktioniert tadellos für erste Gehversuche.

Noch ein Hinweis: Wir sind weder verwandt noch verschwägert mit dem Entwickler von IoT MQTT Panel, auch haben wir keine Aktien im Depot liegen...(leider). Es gibt alternative Apps!

## Konfiguration

### MQTT-Broker verbinden

#### 1.Add New Dashboard



## 2. Add New Dashboard

Name Frei wählbar, Client ID leer lassen wird Automatisch erzeugt, BrokerWeb/IP eintragen zb Djspot oder Broker nach Wahl

The screenshot shows the 'Edit Connection' screen with the following fields:

- Connection name: V3
- Client ID: sLMpyqOTYS
- Broker Web/IP address: fm-funknetz.de
- Port: 1883
- Network protocol: TCP

Falls Notwendig User und Passwort eintragen

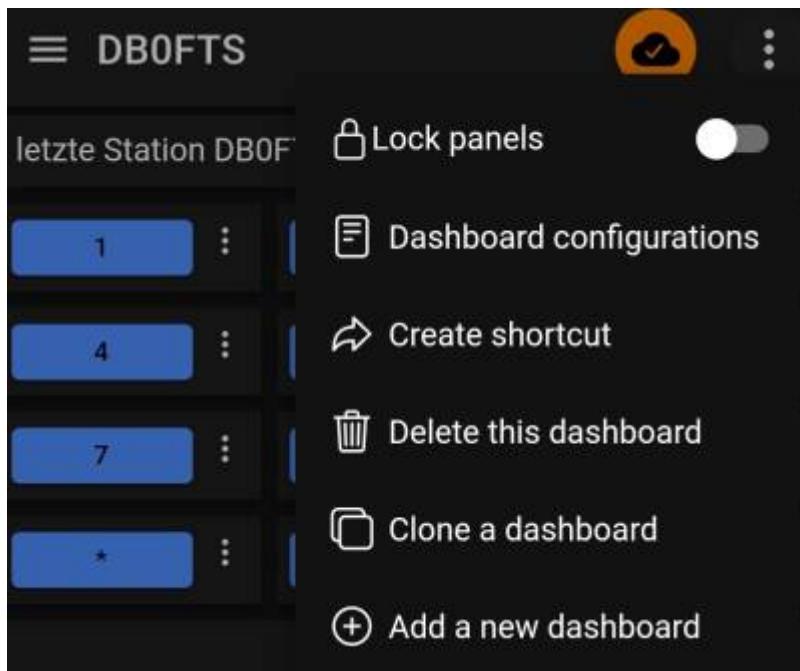
The screenshot shows the 'Additional options' screen with the following settings:

- Connection timeout: 30
- Keep alive: 600
- Username: oneplus9pro
- Password: (redacted)
- Enable will message:
- Notify on disconnect:
- Connect automatically:

At the bottom are 'CANCEL' and 'SAVE' buttons.

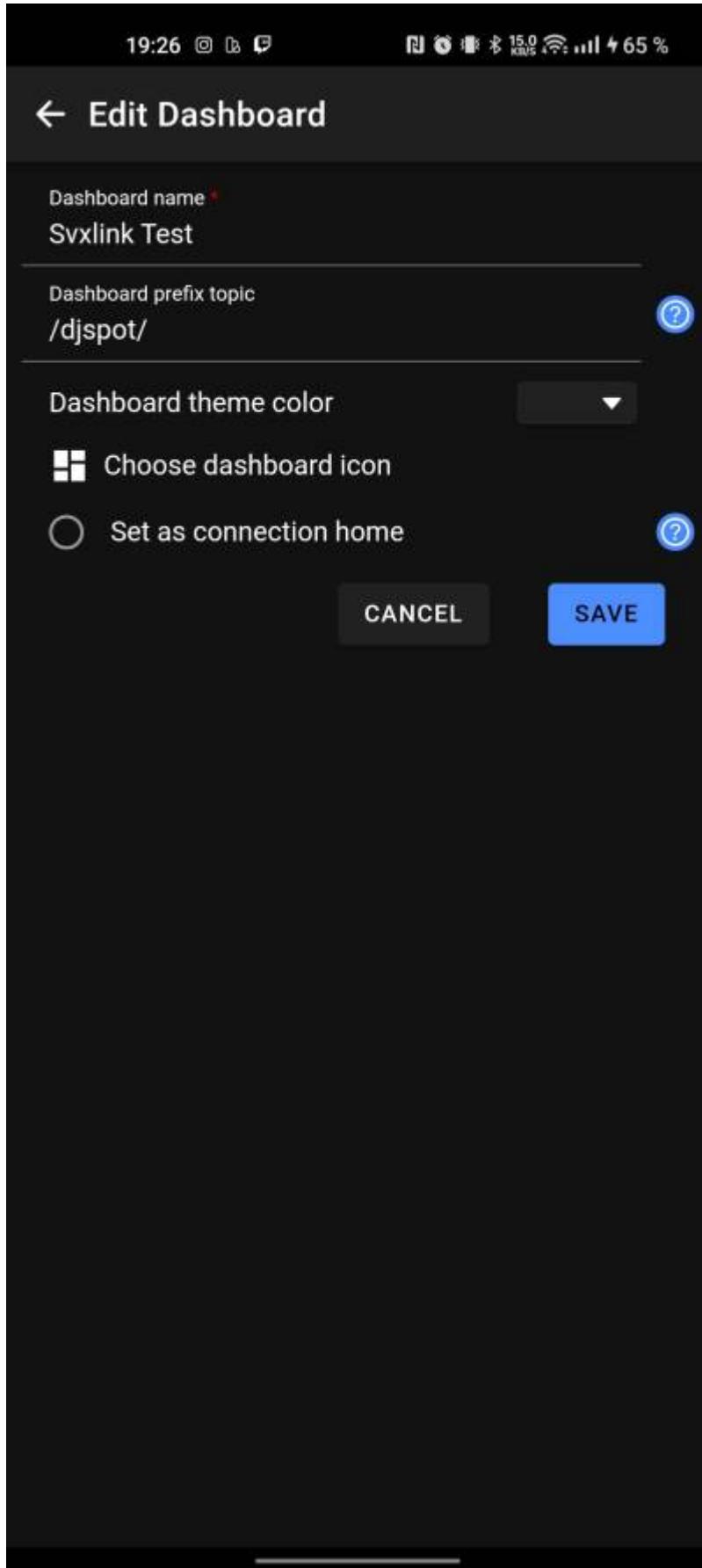
## 3. Dashboard anlegen

Oben Rechts 3 Punkte drücken und ADD NEW DASHBOARD auswählen



#### 4. Dashboard Config

Freie Namenswahl, und Topic vergeben !! Ich vergabe hier vorab das Topic für das Dashboard zb /djspot/



5. Dashboard bauen / Button / Slider etc.

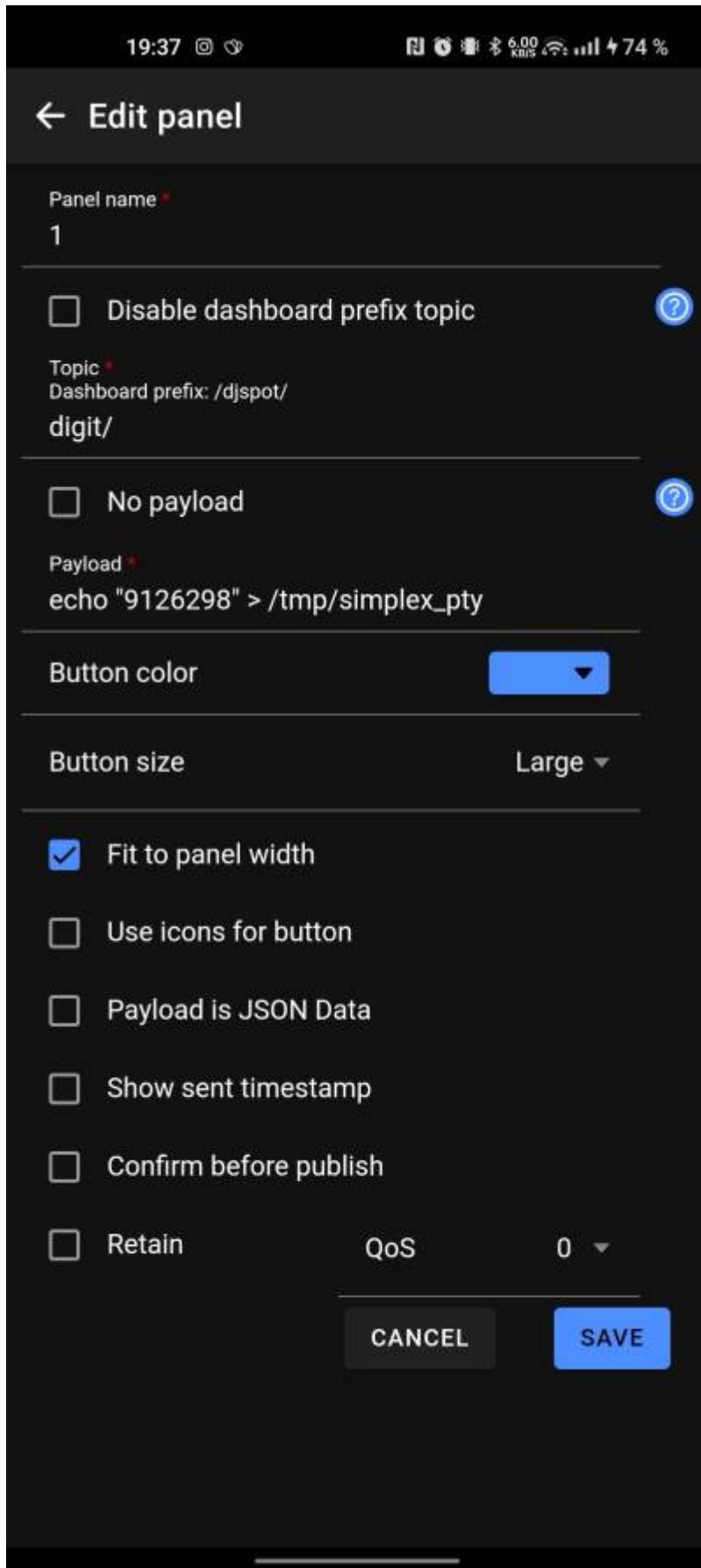
Add Button 3Punkte oben rechts

Panel Name was im Dashboard angezeigt werden soll

Topic was nach dem schon vorgegeben Dashboard topic kommen soll „digit/“ > wäre bei dem Befehl folgendes Topic „/djspot/digit/“ !! Hinweis „/djspot/digit“ ist eine andere Ebene und kann unter umständen keine Aktion Auslösen.

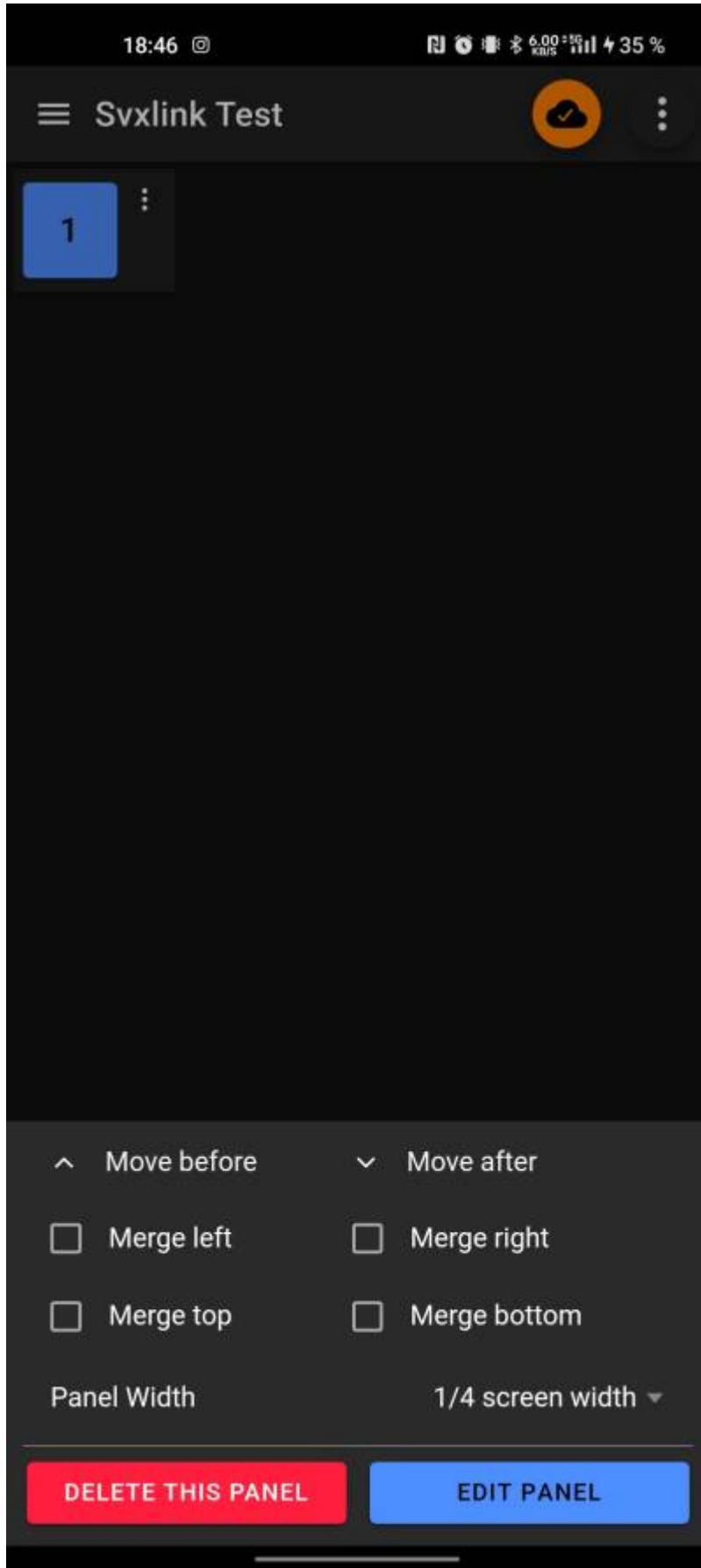
Payload: hier kommt wie man sieht der Befehl rein was man am Ende machen möchte siehe Bild

Abschliesend „Save“

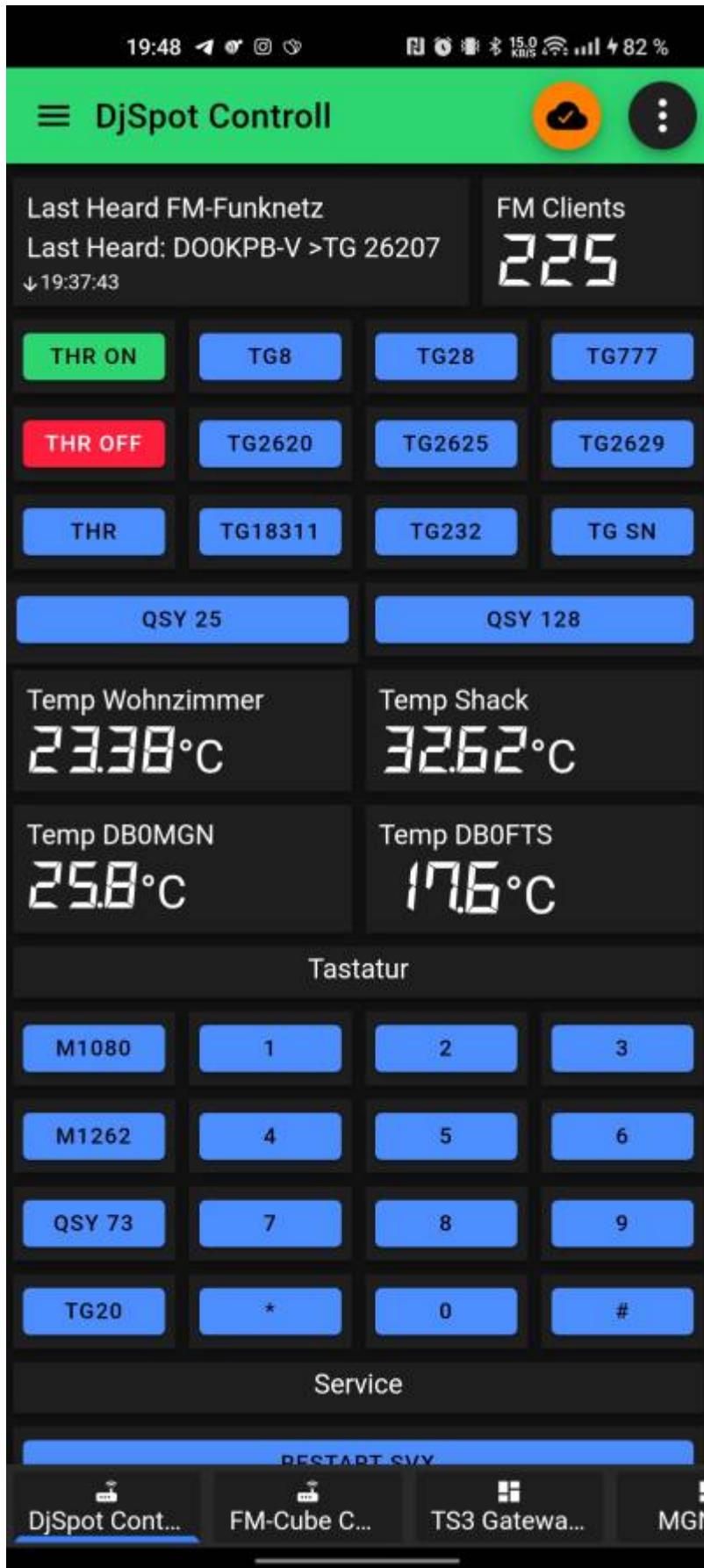


erstes Ergebnis ansehen / Button anpassen

3Punkte am Button drücken und Panel Width einstellen je nach wünschen des entstehenden Dashboards.



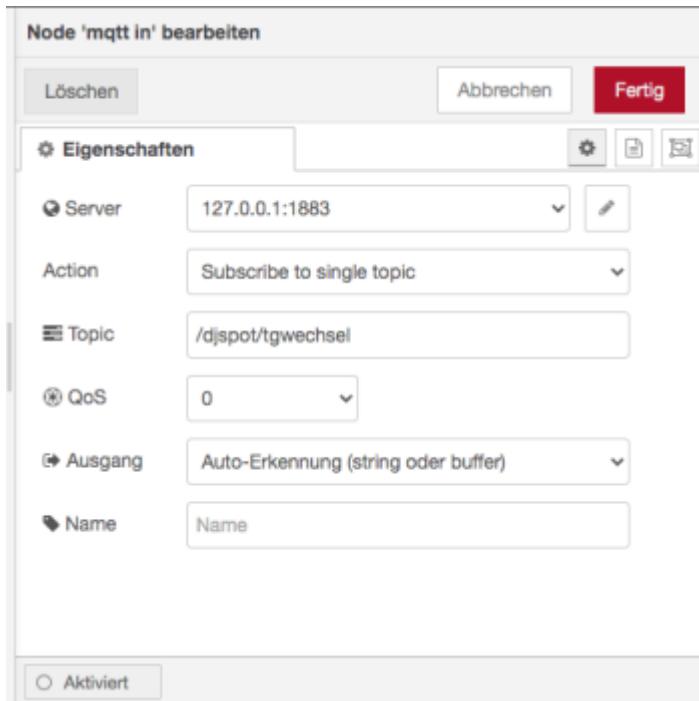
Das Button anlegen wiederholt man solange bis man das erwünschte Ergebnis hat. Zum Schluss könnte es zb. so oder so Aussehen.



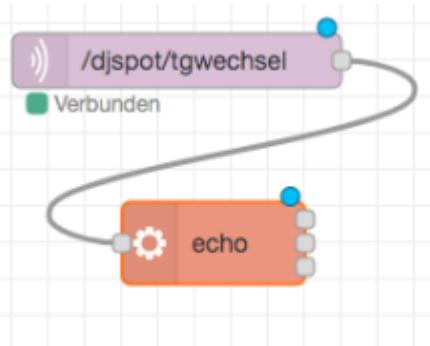


Es ist am Anfang etwas Aufwendig dies alles zu erstellen aber hat man es einmal geschafft, so hat mein ein Mega Tool an der Hand. Man benötigt kein DYNDNS, da sich jeder Client mit dem MQTT Broker verbindet. Einfach Genial ✅

## Node-RED konfigurieren



Damit die vom MQTT-Broker empfangenen Nachrichten auch zu einer Aktion auf dem DJSpot führen, muss in Node-RED ein sogenannter Flow erstellt werden. Hört sich vielleicht kompliziert an, ist es aber nicht. Öffnet zunächst die Konfigurationsoberfläche von Node-RED. Das geht im Browser (Chrome oder Firefox) über die IP-Adresse des DJSpots mit der Portnummer 1880. Wenn die Oberfläche von Node-RED gestartet ist, sieht ihr zentral eine Zeichnungsfläche für die typischen Nodes. Falls ihr das Image von Jens verwendet, ist hier bereits ein Flow vorbereitet. Andernfalls zieht euch zwei Nodes aus dem Werkzeugkasten (links) auf die Zeichnungsfläche. Ihr benötigt: „mqtt in“ und „exec“. Verbindet beide Nodes miteinander.



Ein Doppelklick auf einen der Nodes öffnet die Eigenschaften. Tragt im Node „mqtt in“ folgendes ein:

- **Server:** 127.0.0.1:1883 (evtl Anpassen an euren MQTT Broker)
- **Topic:** /djspot/digit/ (oder was ihr euch ausgedacht habt)

Im Node „exec“ sind diese Einstellungen vorzunehmen:

- **Befehl:** echo
- **zusätzliche Überabeparameter:** > /tmp/simplex\_pty

Beide Einstellungen mit „Fertig“ abschließen und dann nicht vergessen, auf den Button „Übernahme (deploy)“ (rechts oben) klicken, damit die Einstellungen wirksam werden. Nun ist es an der Zeit, eine

Nachricht aus der IoT MQTT-App zu verschicken! Versucht den vorbereiteten Button zu drücken und schaut ob im Log von Svxlink etwas ankommt.

**Noch ein Hinweis: Das hier gezeigte Konfigurationsbeispiel beschreibt recht speziell einen TG-Wechsel. Natürlich können auch andere Befehle via MQTT auf dem Raspberry ausgeführt werden, z.B. systemctl restart svxlink oder ein shutdown, kill oder was auch immer. Dafür würde ich eigene Aktionen (Flows) in Node-RED anlegen, oder aber den bestehenden Flow verallgemeinern (also ohne den jetzt definierten Übergabeparameter > /tmp/simplex\_pty). In diesem Fall müssten die kompletten Anweisungen (Commands) via MQTT als Payload mitgeschickt werden.**

---

Viel Spaß mit dem FM-Funknetz!

73 Jens, DJ1JAY

Sept. 2023

From:  
[./](#) - Wiki FM-Funknetz

Permanent link:  
[./doku.php?id=fm-funknetz:technik:iotmqtt-android](#)



Last update: **16.09.2023 17:55**