

Siri & Co. steuert DJ-Spot



Wer die SvxDLink-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 gestartetem MQTT-Broker ([Anleitung hier](#))
- DJSpot mit installiertem und gestartetem Node-RED ([Anleitung hier](#))

ODER

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

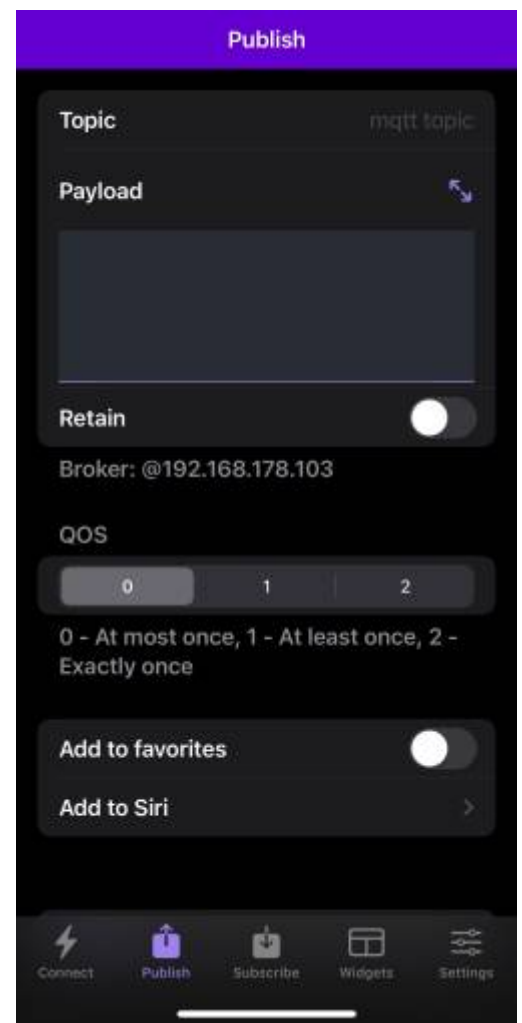
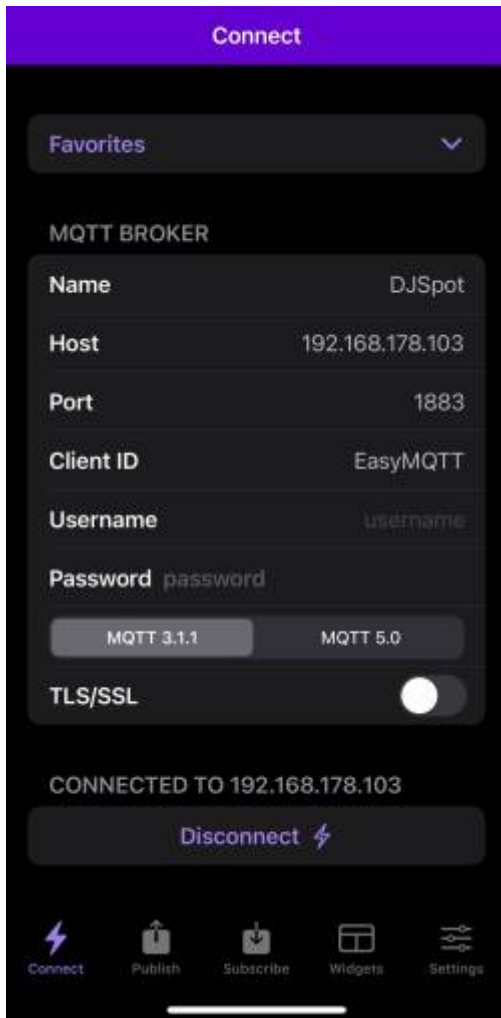


Die nachfolgende Anleitung beschreibt die MQTT-Konfiguration in der App EasyMQTT (findet ihr im AppStore). Die kostenlose Variante funktioniert tadellos für erste Gehversuche, ist aber auf das Anlegen nur eines Favoriten beschränkt und Siri wird auch nicht unterstützt. Wer die Sprachsteuerung sowieso zu nerdig findet, der kann die TG-Wechsel auch über iOS-Kurzbefehle auslösen. Dazu später mehr. Hierfür sollte die kostenlose App-Version ausreichen, denn bei Verwendung der iOS-Kurzbefehle müssen in EasyMQTT keine Favoriten angelegt werden.

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

Konfiguration

MQTT-Broker verbinden



Wechselt in der EasyMQTT-App zunächst auf den Tab „Connect“ . Hier könnt ihr einen Namen für den MQTT-Broker vergeben, z.B. DJSpot. Unter „Host“ tragt ihr die IP-Adresse eures Spots ein. Das ist in der Regel eine lokale IP-Adresse, die der DJSpot vom DHCP-Dienst eures Routers erhält. Wer sein iPhone unterwegs als Hotspot betreibt und den DJSpot damit gekoppelt hat, der kann auch die Adresse „djspot.local“ probieren, falls die Namensauflösung funktioniert.

Dann auf verbinden klicken - der Client verbindet sich jetzt mit dem Raspberry MQTT Broker.

Sofern der Verbindungsaufbau geklappt hat, erscheint die Meldung:

„CONNECTED TO ...“

Nachricht verschicken

Wechselt nun auf den Tab „Publish“. Wenn wir das Anlegen von Favoriten fürs Erste ausser Acht lassen (mehrere Favoriten anlegen geht nur in der Bezahlversion!), können wir eine Testnachricht an den DJSpot verschicken. Unter Topic tragt ihr dazu Folgendes ein:

```
/djspot/tgwechsel
```

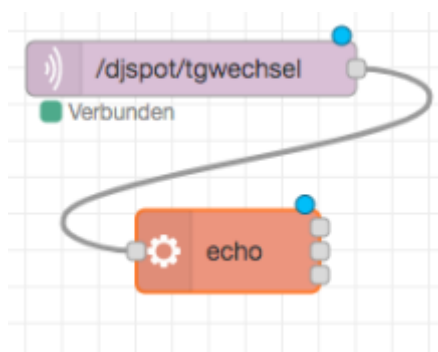
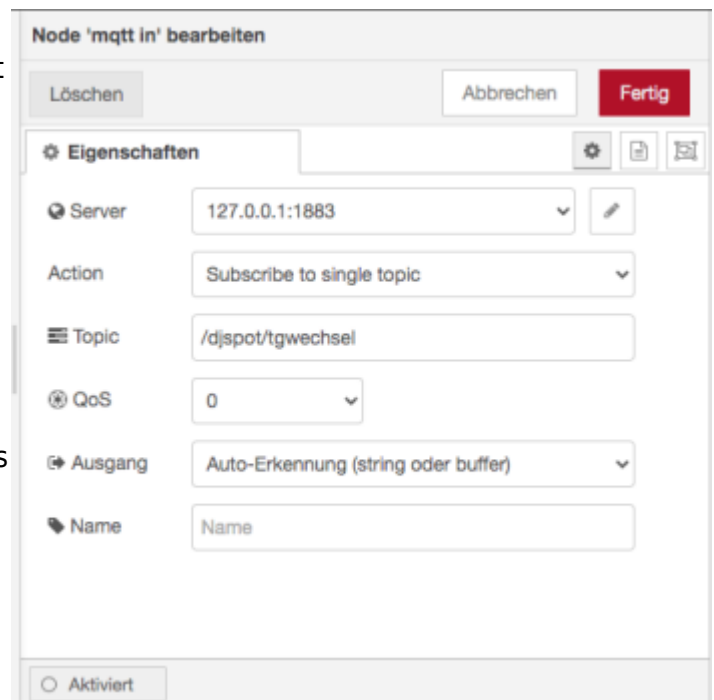
Unter Payload könnt ihr beispielsweise Folgendes eintragen:

9120#

Im Screenshot rechts ist der Button „Publish“ nicht zu sehen. Wenn ihr in der App nach unten scrollt (je nach Displaygröße eures Smartphones), wird dieser Button sichtbar. Ein Verschicken der eingestellten Nachricht macht jedoch zum gegenwärtigen Zeitpunkt noch keinen Sinn. **Dazu folgender Hinweis:** Damit eure Nachrichten auf dem DJSpot empfangen und korrekt ausgewertet werden können, muss eine Node-RED-Aktion angelegt worden sein. Wie ihr das vorbereitet, zeigt der nachfolgende Abschnitt.

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, seht 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
- **Topic:** /djspot/tgwechsel

Im Node „exec“ sind diese Einstellungen vorzunehmen:

- **Befehl:** echo
- **zusätzliche Übergabeparameter:** > /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 EasyMQTT-App zu verschicken! Versucht die vorbereitete Nachricht „9120#“ aus der App zu versenden, es sollte funktionieren.

Favoriten einrichten (nur in der Bezahlversion von EasyMQTT möglich)

Jetzt auf den „Favoriten“ klicken und nach beliebigen Eure Talk Groups eintragen und speichern. Auf Siri klicken und die Siri Kommandos eingeben – dann könnt Ihr Siri aufrufen und mit dem vorher vereinbarten Kommando die Talkgroup wechseln....

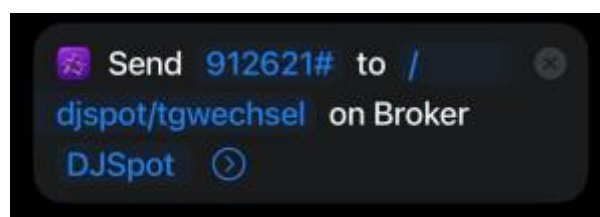
iOS-Kurzbefehle einrichten

Eleganter als die Verwendung der Favoriten in der EasyMQTT-App finde ich es, Kurzbefehle in iOS festzulegen. Sieht schick aus und kann ggf. auch mit Siri ausgelöst werden. Kurzbefehle funktionieren darüber hinaus problemlos mit der kostenlosen Version der MQTT-App. Voraussetzung ist, dass EasyMQTT bereits installiert, konfiguriert und mit dem MQTT-Server (Broker) verbunden ist. Ein Kurzbefehl an EasyMQTT setzt sich wie folgt zusammen:



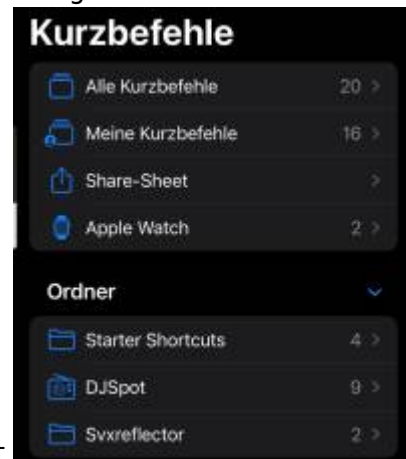
1. dem **Payload**: das ist in unserem Fall die aus den DTMF-Befehlen bekannte Ziffernfolge für den TG-Wechsel, also 912621# für einen Wechsel in TG 2621.
2. dem **Topic**: das ist sozusagen der „Betreff“ der Zustellung an den Broker. In unserem Fall ist das „/djspot/tgwechsel“ oder das was ihr unter MQTT-Einrichten konfiguriert habt.
3. dem **Broker**: das ist der Adressat, oder Empfänger. In unserem Fall haben wir ihn „djspot“ genannt.

Der fertige Kurzbefehl für einen Wechsel in die TG 2621 sieht so aus:



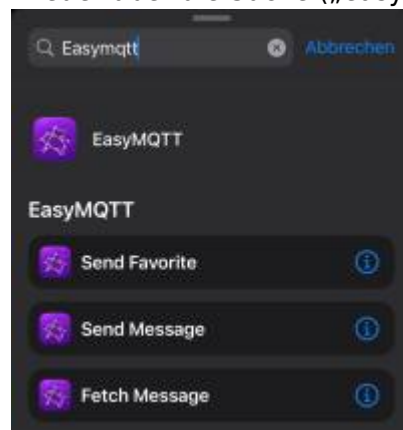
Schritt für Schritt:

1. Kurzbefehle-App auf dem iPhone starten. 
2. In der Übersicht der Kurzbefehle lassen sich neue Ordner anlegen und Kurzbefehle damit



gruppieren. Ich habe mir einen Ordner „DJSpot“ angelegt.

3. In den angelegten Ordner wechseln. Über „+“ einen neuen Kurzbefehl hinzufügen.
4. Aktion hinzufügen
5. App auswählen. Entweder über die Suche („easymqtt“) oder unter „Apps“ und dann EasyMQTT



→ „Send Message.“

6. Payload = „9120#“ (z.B. für einen Wechsel in die TG 20); to Topic „/djspot/tgwechsel,“; Broker = „djspot“ (ALLE Eingaben ohne „ „ !)

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_ptty`). In diesem Fall müssten die kompletten Anweisungen (Commands) via MQTT als Payload mitgeschickt werden. Entsprechend anders sind die Kurzbefehle in iOS zu konfigurieren.

Viel Spaß mit dem FM-Funknetz!

73 Peter, DL4ZAQ und Michael, DG6IMF

April 2023

From:

[./](#) - **Wiki FM-Funknetz**

Permanent link:

[./doku.php?id=fm-funknetz:technik:easymqtt_ios](#)

Last update: **02.05.2023 16:22**

