

SVXcube Display: DJ-Spot TG Wechsel per MQTT mit SVXcube Display steuern.

Konfiguration SVXcube mit DJ-Spot Hotspot (**Image von DJ1JAY Jens installiert**)

Vorraussetzungen:

1. Linux Grundkenntnisse – Schwierigkeitsgrad: Mittel – Anfängern rate ich zum „einlesen“ der u.a. Links zum Thema und sofern möglich einen Funkfreund mit Linux Kenntnissen um Unterstützung zu bitten.
2. DJ-Spot mit dem Image von Jens DJ1JAY ist installiert (Download und Anleitung fm-funknetz.de) und alles funktioniert.

An dieser Stelle ein Hinweis: Ich kann Euch leider nicht mit dem Shari (Ali-Spot) oder dem Daniel (Poland) Spot helfen. Es könnte nach Installation von MQTT und Konfiguration von Node Red ähnlich funktionieren. Hier muss dann auch noch die SVXlink.conf angefasst werden. Supportanfragen kann ich leider nicht bearbeiten.

3. SVXcube mit der Firmware 1.2 installiert und konfiguriert. SVXcube ist verbunden und TG, Datum Uhrzeit wird im Display angezeigt. TG Auswahl und Wechsel (MQTT) funktioniert noch nicht.
4. DJ-Spot und SVXcube sind mit dem gleichen WLAN verbunden (Handy Hotspot habe ich nicht getestet)

Vorgehensweise:

Ich empfehle die Lektüre der folgenden Informationen zum Thema MQTT und NodeRed um die Grundprinzipien zu verstehen und dieser Anleitung gut folgen zu können. Der Lerneffekt ist enorm!

1. <https://www.elektronik-kompendium.de/sites/raspberry-pi/2709041.htm>
2. <https://randomnerdtutorials.com/getting-started-node-red-raspberry-pi/>

Jetzt kann es losgehen! Seit Ihr bereit?

Anpassen der SVXcube.ini

```
[mqtt]
host = IP des MQTT Broker (Djspot/Svxlink Client)
port = 1883
user =
(bleibt leer mit DJ1JAY DJSpot Image / ansonsten Daten des konfigurierten
MQTT Broker setzen - falls User und PW gesetzt)
pass =
module = 9
topic = /djspot/svx/digit
pty = /tmp/simplex_pty
type = 1
```

Das DJ-Spot Image von Jens hat bereits einen lauffähigen MQTT Broker sowie eine lauffähige Version von Node Red installiert. Seht gut!

Die SVXlink Konfiguration (Image von Jens!!) **muss nicht angepasst werden!** Das ist doch erneut gut!

Im ersten Schritt überprüfen wir ob die MQTT Kommunikation auf dem Raspberry (Server) d.h. lokal funktioniert:

Mosquitto lokal testen Mit folgendem Kommando starten wir den MQTT-Subscriber-Client und „subscriben“ das Topic „test“.

```
mosquitto_sub -t test
```

Der Subscriber-Client kann mit STRG + C beendet werden. Um den Broker zu testen, benötigen wir ein zweites Terminal (Fenster/Reiter), in dem wir Daten an den Broker schicken können. In dem zweiten Terminal schicken wir mit dem MQTT-Publisher-Client eine Nachricht an den Broker.

```
mosquitto_pub -t test -m "Hallo Welt"
```

Wenn im ersten Terminal die Nachricht „Hallo Welt“ angezeigt wird, dann war die MQTT-Kommunikation und damit auch die Installation erfolgreich.

Hinweis: An dieser Stelle könnte man sagen, dass die Installation und Konfiguration erfolgreich war und abgeschlossen ist. Dem ist aber nicht so. Denn bis zu diesem Punkt funktioniert die Kommunikation nur lokal auf dem Raspberry Pi. Aber die MQTT-Kommunikation über das Netzwerk funktioniert standardmäßig noch nicht. Die muss noch konfiguriert werden.

Hat alles funktioniert? Nur dann macht es Sinn, hier weiterzumachen...

Der Einfachheit halber werden wir ohne Authentifizierung arbeiten. Es ist jetzt eine Konfigurationsdatei für MQTT anzulegen.

Konfiguration: Kommunikation ohne Authentifizierung

Vorsicht, die folgende Konfiguration ermöglicht es ohne Authentifizierung mit dem MQTT-Broker über das Netzwerk zu kommunizieren.

Zuerst öffnen wir die Konfigurationsdatei von Mosquitto

```
sudo nano /etc/mosquitto/conf.d/local.conf
```

In die Datei fügen wir folgende Zeilen ein:

```
listener 1883
allow_anonymous true
```

Anschließend die Datei speichern und schließen: Strg + O, Return, Strg + X. Damit die Konfiguration wirksam wird, muss Mosquitto neu gestartet werden.

```
sudo systemctl restart mosquitto
```

Hinweis: Wenn der Mosquitto neu gestartet wird, werden dabei alle MQTT-Verbindungen beendet und müssen neu hergestellt werden.

Jetzt überprüfen wir, ob der SVXcube seine MQTT Messages beim Betätigen der entsprechenden Tasten (TG Wechsel) aussendet und ob der MQTT Broker auf unserem Raspberry die Nachrichten auch empfängt und uns anzeigt.

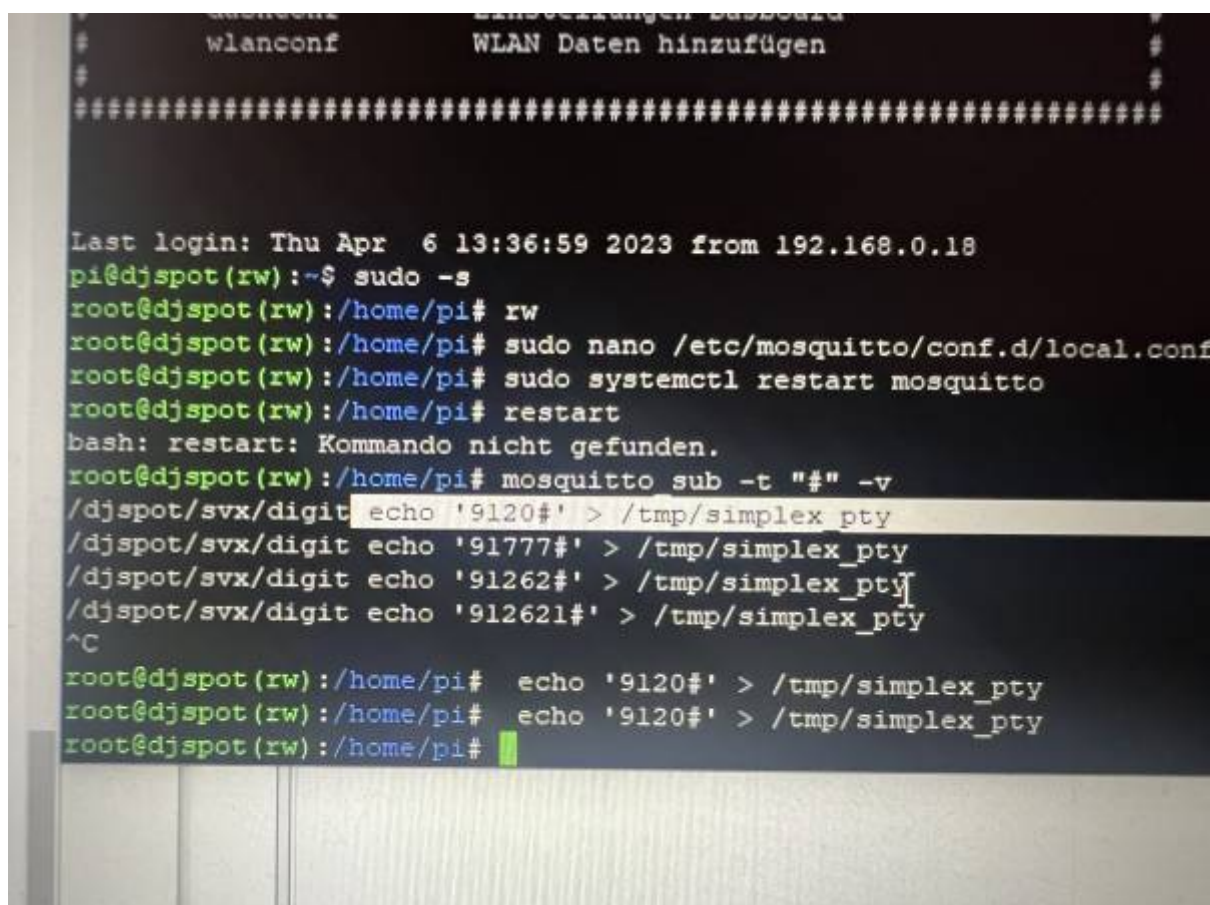
Kommunikation übers Netzwerk prüfen Die Frage ist jetzt, funktioniert die Kommunikation über das Netzwerk mit oder ohne Authentifizierung?

Hierzu startet man einen MQTT-Subscriber-Client mit folgendem Kommando (ohne Authentifizierung am MQTT-Broker):

```
mosquitto_sub -t "#" -v
```

Dieses Kommando hört auf jedes Topic (#) und meldet sobald neue Daten empfangen wurden. Man kann dieses Kommando in einer Kommandozeile als Logging-Anzeige laufen lassen, wenn man wissen will, ob ein MQTT-Publisher Daten gesendet hat.

Der Subscriber-Client kann mit STRG + C beendet werden. Im Terminalfenster seht ihr jetzt die vom SVXcube empfangenen MQTT Meldungen. Und ihr könnt über den Echo Befehl die TG manuell umschalten.



```
#####
Last login: Thu Apr  6 13:36:59 2023 from 192.168.0.18
pi@djspot(rw) ~$ sudo -s
root@djspot(rw) ~/home/pi# rw
root@djspot(rw) ~/home/pi# sudo nano /etc/mosquitto/conf.d/local.conf
root@djspot(rw) ~/home/pi# sudo systemctl restart mosquitto
root@djspot(rw) ~/home/pi# restart
bash: restart: Kommando nicht gefunden.
root@djspot(rw) ~/home/pi# mosquitto_sub -t "#" -v
/djspot/svx/digit echo '9120#' > /tmp/simplex_pty
/djspot/svx/digit echo '91777#' > /tmp/simplex_pty
/djspot/svx/digit echo '91262#' > /tmp/simplex_pty
/djspot/svx/digit echo '912621#' > /tmp/simplex_pty
^C
root@djspot(rw) ~/home/pi# echo '9120#' > /tmp/simplex_pty
root@djspot(rw) ~/home/pi# echo '9120#' > /tmp/simplex_pty
root@djspot(rw) ~/home/pi#
```

Alles gut? Kommen die MQTT Messages bei Euch an? Prima!

Jetzt der nächste Test:

Funkgerät einschalten, Lautstärke aufdrehen und über das Terminal folgenden Befehl senden:

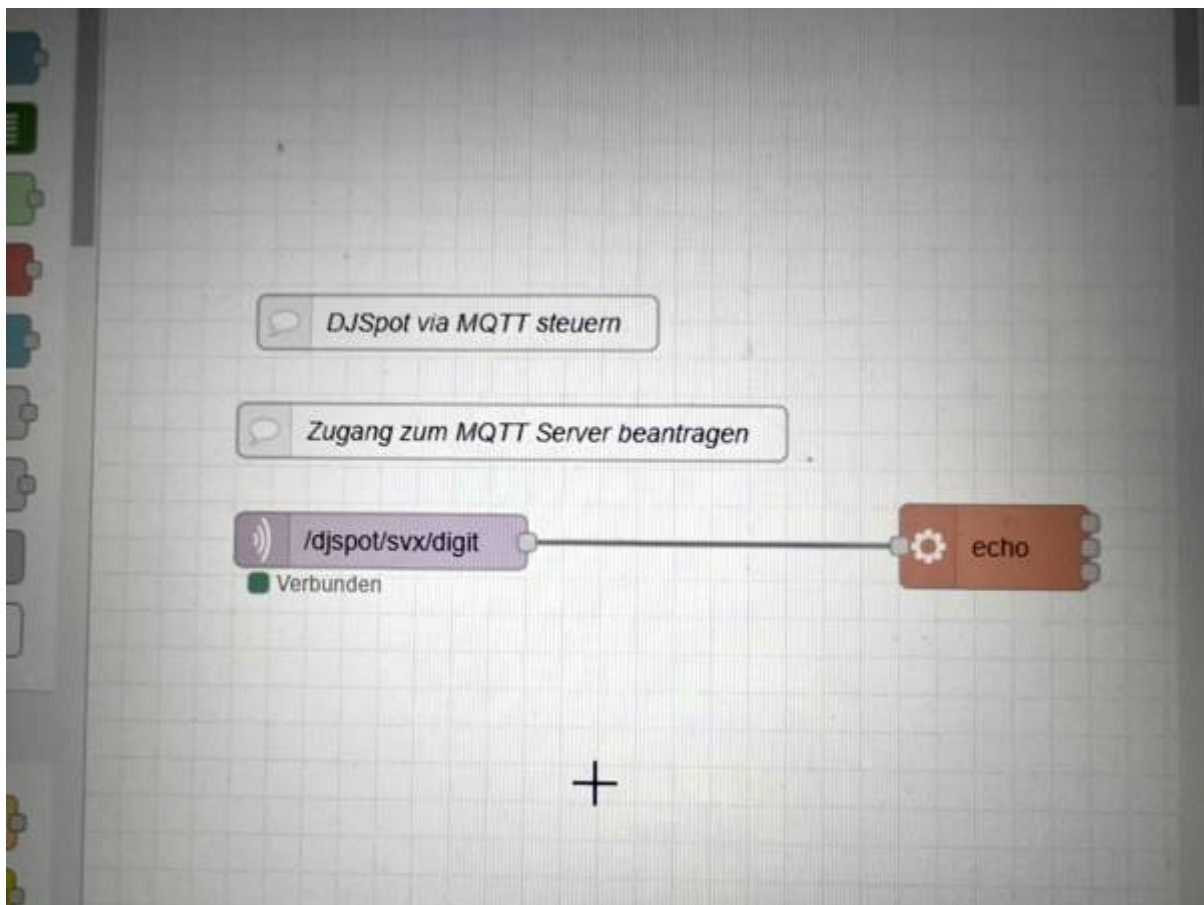
```
echo „9120#“> /tmp/simplex_pty
```

Und der DJ-Spot sollte jetzt auf die TG20 wechseln. Jetzt kommt der letzte Schritt: Konfiguration von Node-Red (Habt Ihr den Link dazu gelesen? Das ist eine wirklich spannende Software mit der sich einiges automatisieren lässt... nicht nur Euer DJ-Spot bzw. svxlink Node-Red ist bereits auf dem Raspberry installiert. Es muss gestartet werden: Also Terminal auf und los geht's:

```
node-red-start
```

Jetzt müssen wir den „Flow“ anpassen. **Dazu öffnet Ihr einen Browser und gebt die folgende Adresse ein: <http://IP> Adresse Eures Raspberry Pi's:1880**

Es öffnet sich ein Browserfenster mit der Oberfläche von Node-Red. Dort wird Euch bereits ein „Flow“ angezeigt den ihr wie folgt ändern müsst:



Die Änderungen nehmt ihr vor, indem ihr auf die „Bausteine“ im Flow klickt und in dem dann erscheinenden Menü die Einträge vornehmt bzw. die nicht benötigten Module löscht.

Jetzt werden die zwei Module angepasst - d.h. anklicken und die Änderungen wie abgebildet vornehmen und speichern nicht vergessen!

Node 'mqtt in' bearbeiten

Löschen Abbrechen Fe

Eigenschaften

Server 127.0.0.1:1883

Action Subscribe to single topic

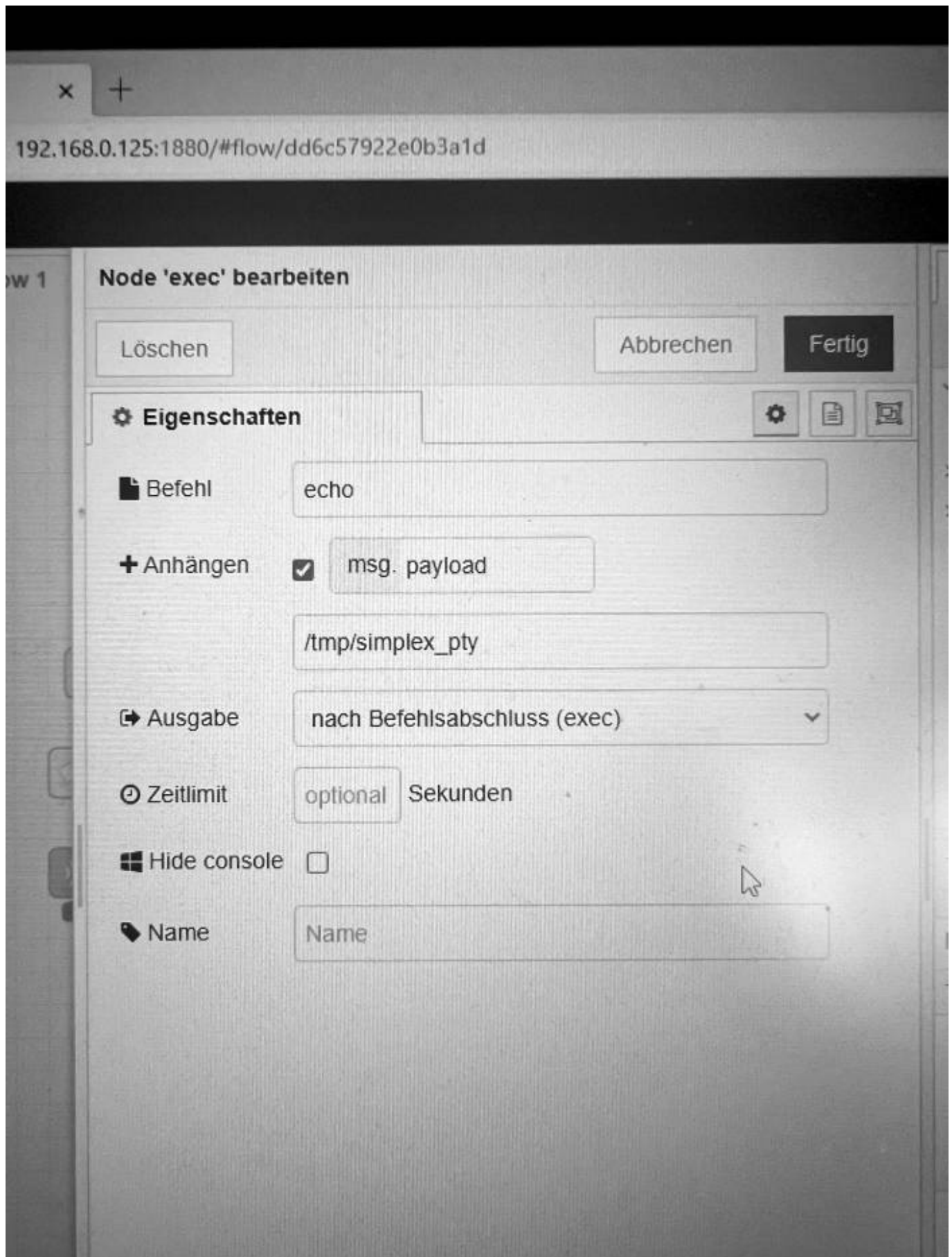
Topic /djspot/svx/digit

QoS 0

Ausgang Auto-Erkennung (string oder buffer)
This option is depreciated. Please use the new auto-detect mode.

Name Name

or-tab-properties Aktiviert



Den Flow noch aktualisieren indem ihr auf **DEPLOY** klickt (rechts oben) und

Ab sofort sollte der DJ-Spot auf den SVXcube reagieren und ihr habt es geschafft. Herzlichen Glückwunsch!

Node-Red und der MQTT Broker starten beim Systemstart automatisch.

Viel Freude mit dem SVXcube!

Peter, DL4ZQA im April 2023

From:

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

Permanent link:

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

Last update: **22.04.2023 05:15**

