

## Installation evSoc Version 2 auf Raspberry Pi

Diese Anleitung richtet sich an Windows Benutzer und wenn evSoc auf einem separaten Raspberry installiert werden soll. Die Installation ist auch direkt auf dem Raspberry in der OpenWB möglich, dabei wären dann einige Schritte abweichend von dieser Anleitung!

Alle Befehle oder Ausgabemeldungen des Raspberry sind grün dargestellt. Die Befehle können einfach aus dem Dokument kopiert und mit Rechtsklick auf der Maus in das Putty Fenster eingefügt werden.

Erforderlich:

- Raspberry Pi (in meinem Fall Raspberry Pi 2)
- SD Karte mit mind. 16GB
- USB Bluetooth Dongle (kompatibel zu Raspberry) - SOFERN der RPi kein eingebautes BT hat
- OBDII Bluetooth Dongle ELM327 (es funktioniert leider nicht jeder ELM327 Dongle, derzeit als funktionierend bekannt sind „TONWON OBD2 Bluetooth Mini OBD2“ für ca. 15€, und ich nutze OBDLink LX für ca. 55€)

Zuerst muss das Betriebssystem Raspbian auf den Raspberry. Dazu die SD-Karte in den SD-Reader am PC stecken und mit dem Programm Raspberry Pi Imager das aktuelle Image auf die SD-Karte schreiben. Nun sollte der Zugriff per SSH auf des Raspberry eingerichtet werden, weil man sonst nicht aus dem Netzwerk darauf zugreifen kann. Dazu einfach die SD-Karte einmal kurz abziehen und wieder einstecken und dann auf dem Laufwerk „Boot“ eine Datei (ohne Dateiendung!) mit dem Namen „ssh“ erstellen. (Mehr dazu z.B. [hier](#)).

Der Raspberry sollte mit dem Netzwerk verbunden sein. In meinem Fall per LAN, wenn man den Raspberry per WLAN einbinden will, muss man dies entsprechend vorab einrichten (z.B. [hier](#) beschrieben). Dabei muss eine Datei mit Namen „wpa\_supplicant.conf“ ebenfalls direkt auf der SD-Karte angelegt werden mit folgendem Inhalt:

```
# Datei wpa_supplicant.conf in der Boot-Partition (Raspbian Stretch)
country=DE
ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=netdev
update_config=1
network={
    ssid="wlan-bezeichnung"
    psk="passwort"
    key_mgmt=WPA-PSK
}
```

Nun kann die SD Karte in den Raspberry.

Dann die IP Adresse des Raspberry ermitteln (unter Windows z.B. mit „Ausführen“->“CMD“->“ping raspberrypi“

Nun brauchen wir einen SSH Client (z.B. Putty). Hier nach dem Start den Host name oder die eben ermittelte IP eingeben, Port 22, Connection Type „SSH“. Kommt ggfs. Ne Meldung mit dem SSH Key, da auf „Ja“ klicken. Dann kann man sich auf dem Raspberry einloggen mit „pi“ und Passwort „raspberrypi“.

Nun sollte als erstes das Passwort geändert werden (kommt auch ein Hinweis). Dazu einfach „passwd“ eingeben und dann einmal das aktuelle Passwort (raspberrypi) eingeben und dann wie angewiesen zweimal das neue Passwort.

Nun die folgenden Befehle:

```
sudo apt update
sudo apt upgrade
sudo apt install python3-
{pip,rpi.gpio,serial,requests,sdnotify,pyroute2,smbus,yaml,gevent} gpsd git
watchdog rsyslog-
sudo pip3 install paho-mqtt
```

Jetzt müssen die Dateien von evSoc auf den Raspberry kopiert werden. Dafür nutze ich WinSCP. Man kann aber auch ein anderes SFTP Programm nutzen. Man verbindet sich auf den Raspberry mit dem Benutzernamen pi und dem eben geänderten Passwort.

Nun geht man auf das Verzeichnis /home/pi und kopiert die evSoc-2.1.tar dort hin.

Nun zurück in Putty kann man die Datei in das Verzeichnis /opt/ kopieren:

```
sudo cp evSoc-2.1.tar /opt/
```

Wechseln ins Verzeichnis /opt:

```
cd /opt
```

Datei entpacken:

```
sudo tar xf evSoc-2.1.tar
```

Nun muss die config angepasst werden auf den Fahrzeugtyp und die IP der openWB:

```
sudo nano /opt/evSoc/config.yaml
```

Es öffnet sich die Datei config.yaml im Editor namens „nano“. Dort kann man einfach mit den Pfeiltasten der Tastatur an die richtige Stelle.

Die Raute am Anfang einer Zeile bedeutet, dass diese auskommentiert ist (nicht aktiv). Hier muss also die Raute vor der Zeile mit eurem Fahrzeug weg und stattdessen in die Zeile vor „KONA\_EV“. Es dürfen nicht zwei Zeilen mit „type“ gleichzeitig aktiv sein!

Dann muss noch unten bei „mqtt“ hinter „server:“ localhost durch die IP eurer openWB ersetzt werden.

Nun das Ganze speichern mit „STRG“ und „o“ gleichzeitig gedrückt, einmal Enter um den Namen zu bestätigen und dann „STRG“ und „x“ zum beenden des Editors.

Jetzt kommt die Bluetooth Verbindung:

```
sudo bluetoothctl
power on
agent on
default-agent
scan on
```

Bei den Ergebnissen sollte euer OBDII Dongle auftauchen mit seiner MAC Adresse xx:xx:xx:xx:xx:xx, diese müsst ihr immer dort eintragen wo diese xx hier stehen

```
pair xx:xx:xx:xx:xx:xx
trust xx:xx:xx:xx:xx:xx
scan off
quit
```

```
sudo rfcomm bind rfcomm0 xx:xx:xx:xx:xx:xx
```

Jetzt würde ich empfehlen das Ganze mal zu testen, indem evSoc per Hand gestartet wird. Das hat den Vorteil, dass man sehen kann ob alles funktioniert wie es soll.

```
cd /opt/evSoc
./evSoc.py
```

Dauert jetzt einen Moment, dann sollte eine Info Meldung kommen in der Art:

```
INFO:EVNotiPi/AtBaseDongle:Initializing PiOBD2Hat
INFO:evSoc:MQTT subscribed
INFO:evSoc:Charging started
INFO:evSoc:Plugged
INFO:evSoc:SoC=61.0
```

Mit „STRG“ und „c“ wird das Programm wieder gestoppt.

Am besten ist es dazu, wenn das Auto am Ladekabel hängt, weil man dann gleich den Ladestand mitbekommt (wie in meinem Beispiel).

Sollte es hier zu irgendwelchen Fehlermeldungen kommen kann man in der config.yaml den Log Level auf Debug stellen (wie oben beschrieben die # vor der Zeile weg machen).

Ansonsten kann man jetzt alles „dauerhaft“ einrichten, indem man das Binding des Bluetooth Dongle auf rfcomm0 in den Autostart legt:

```
sudo nano /etc/rc.local
```

In eine leer Zeile folgendes eintragen:

```
sudo rfcomm bind rfcomm0 xx:xx:xx:xx:xx:xx
```

Wie oben beschrieben wieder mit STRG + o, Enter, STRG+x beenden.

Jetzt noch evSoc als Dienst registrieren und aktivieren:

```
sudo systemctl link /opt/evSoc/evSoc.service  
sudo systemctl enable evSoc.service
```

Ich würde den Raspberry nun vorsichtshalber einmal neu starten:

```
sudo reboot
```

Auf der openWB muss für den Ladepunkt 1 noch mqtt als SOC Modul aktiviert werden.

Dann am besten Fahrzeug mal weiter laden lassen und beobachten, ob der SOC in der openWB sauber angezeigt wird.

Viel Erfolg!

Gruß,  
Jürgen