

Inhaltsverzeichnis

MQTT SoC

MQTT SoC allgemein

MQTT SoC mit WiCAN-OBd2

2

2

3

MQTT SoC

Die Einrichtung erfolgt in openWB software2 im Webinterface unter Einstellungen - Konfiguration - Fahrzeuge im jeweiligen Fahrzeug. Das MQTT-Modul für die Übergabe von SoC-Werten über das MQTT-Protokoll stammt von openWB. Für die Einrichtung der externen, übergebenden Software (z.B. HomeAssistant oder meatPi) bitte in [openWB-Forum](#) beraten lassen, das ist keine Support-Leistung von openWB.

MQTT SoC allgemein

Das MQTT SoC Modul sieht vor, dass externe Software bestimmte Werte an die openWB übergibt. Die Pfade hierbei sind wie folgt, wobei die Nummer **/0/** für die ID des Fahrzeugs in der openWB steht (einzusehen unter Status):

The screenshot shows the 'Status' page of the openWB web interface. The top navigation bar includes links for openWB, Status, Auswertungen, Einstellungen, Ladeeinstellungen, Konfiguration, System, and Wiki. The main content area displays several status cards:

- Alle Ladepunkte** (All charging points) - kW
- LP1** (Charging point 1) - kW
- LP2** (Charging point 2) - kW
- EVU Zähler** (Grid meter) - 1,083 kW
- Wechselrichter** (Inverter) - 1,519 kW
- Speicher** (Storage) - kW / 100 %
- Standard-Fahrzeug** (Standard vehicle) - ID: 0
- Fahrzeug 2** (Vehicle 2) - ID: 1

Each vehicle card shows 'Fahrzeugdaten' (Vehicle data) with fields for Ladestand (Charging status), Reichweite (Range), and Letzter Zeitstempel (Last timestamp). Red arrows point to the 'ID' field in the vehicle cards, indicating the vehicle ID used in the MQTT paths.

Die Pfade im Einzelnen:

- openWB/set/mqtt/vehicle/0/get/soc
- für Fahrzeug-SoC mit Nachkommastellen (Float) oder Ganzzahl
- Beispiel: 89.5
- openWB/set/mqtt/vehicle/0/get/soc_timestamp
- für Zeitstempel des SoCs in s als Unix-Zeitstempel
- Diese Info ist optional. Wird kein Wert für das Topic veröffentlicht, wird bei der Abfrage automatisch der aktuelle Zeitstempel gesetzt.
- Beispiel: 1748339524

- openWB/set/mqtt/vehicle/0/get/range
- für Reichweite des Fahrzeugs in km mit Nachkommastellen (Float) oder Ganzzahl
- Diese Info ist optional
- Beispiel: 356.5

Die Pfade sind auch noch mal direkt im MQTT-Modul in der Oberfläche beschrieben:

SoC-Modul des Fahrzeugs ?
MQTT

SoC direkt aus Fahrzeug auslesen ?

Nein ✓

Ja

Aktualisierung der Fahrzeugdaten ?

Während der Ladung ?

5

Min.

Ohne laufende Ladung ?

720

Min.

Nur aktualisieren wenn angesteckt ?

Nein ✓

Ja

Das ausgewählte Fahrzeug-Modul "MQTT" wird von openWB gepflegt.

Einstellungen für SoC-Modul "MQTT"

- openWB/set/mqtt/vehicle/0/get/soc

Fahrzeug-SoC mit Nachkommastellen (Float) oder Ganzzahl

Beispiel: 89.5
- openWB/set/mqtt/vehicle/0/get/soc_timestamp

Zeitstempel des SoCs in s als Unix-Zeitstempel

Diese Info ist optional. Wird kein Wert für das Topic veröffentlicht, wird bei der Abfrage automatisch der aktuelle Zeitstempel gesetzt.

Beispiel: 1748339524
- openWB/set/mqtt/vehicle/0/get/range

Reichweite des Fahrzeugs in km mit Nachkommastellen (Float) oder Ganzzahl

Diese Info ist optional.

Beispiel: 356.5

SoC während der Ladung berechnen ?

Nein ✓

Ja

MQTT SoC mit WiCAN-OBD2

Mit Hilfe des WiCAN OBD2 Dongles von meatPi Electronics können die Werte des Fahrzeugs über die OBD2-Schnittstelle ausgelesen und im WLAN per MQTT an die openWB Wallbox gesendet werden. In Verbindung mit dem manuellen SoC Modul der openWB ist hiermit die Nutzung ohne Cloud-Dienste der Fahrzeughersteller möglich. <https://github.com/meatpiHQ/wican-fw>

Die erste Umsetzung erfolgte von User „zut“ im openWB Forum, der hierfür auch eine Unterstützung von Spritmonitor umgesetzt hat. Hierfür ist ein weiteres Gerät im Netzwerk nötig, auf dem ein kleines Programm (soc_helper) laufen kann: <https://forum.openwb.de/viewtopic.php?t=7451>

Im Laufe der weiteren Entwicklung wurde die Möglichkeit des automatischen Auslesen (AutoPID) in die Firmware des Gerätes integriert. Hierfür ist nun keine zusätzliche Hardware mehr nötig, da der Dongle die Daten direkt per MQTT an die openWB sendet. Die Unterstützung von Spritmonitor ist hiermit jedoch nicht möglich. Die Einrichtung dieser Funktion soll hier näher beschrieben werden.

1. Voraussetzungen

WiCAN Dongle mit Firmware 3.48, 4.03 oder neuer (4.00 ist fehlerhaft). Konfiguration des Manuellen SoC Moduls im Fahrzeug der openWB. WLAN Empfang im Bereich der openWB bzw. des Fahrzeugs. Ggf. Beachtung einer Alarmanlage, die den OBD2 Port überwacht.

2. WiCAN Dongle

Die Beschaffung der Hardware ist zunächst die größte Hürde, da die Firma meatPi Electronics Ihren Sitz in Australien hat. Es sind aktuell 2 Bezugsquellen bekannt:

<https://github.com/meatpiHQ/wican-fw?tab=readme-ov-file#order-on-mouser-or-crowd-supply> Es ist hierbei darauf zu achten, dass nur der WiCAN-OB2 oder der WiCAN PRO (Coming soon) verwendet werden kann. Bei Bestellung über Mouser erfolgt der Versand ab 50€ Versandkostenfrei, so dass sich idealerweise 2 oder mehr Besteller zu einer Sammelbestellung zusammentun können. Achtung: Die angegebenen Preise sind Nettopreise.

3. Konfiguration im MQTT SoC-Modul der openWB

In der Fahrzeugkonfiguration der openWB muss das MQTT SoC Modul eingerichtet werden. Für die spätere Konfiguration der MQTT-Topics wird die Fahrzeug-ID benötigt. Diese kann man am besten auf der Status Seite der openWB auslesen:

The screenshot shows the 'Status' page of the openWB interface. It features a navigation bar at the top with links like 'openWB', 'Status', 'Auswertungen', 'Einstellungen', 'Ladeeinstellungen', 'Konfiguration', 'System', and 'Wiki'. Below the navigation bar, the 'Status' section displays various system components and their current status. Two vehicle configurations are shown: 'Standard-Fahrzeug' (ID: 0) and 'Fahrzeug 2' (ID: 1). Each vehicle configuration includes a table of 'Fahrzeugdaten' (Vehicle Data) with columns for 'Ladestand' (Charge Status), 'Reichweite' (Range), and 'Letzter Zeitstempel' (Last Timestamp). Red arrows point to the 'ID' field for each vehicle, indicating the vehicle ID used for MQTT configuration.

Component	Status
Alle Ladepunkte	- kW
LP1	- kW
LP2	- kW
EVU Zähler	-1,083 kW
Wechselrichter	-1,519 kW
Speicher	- kW / 100 %
Standard-Fahrzeug	ID: 0
Fahrzeug 2	ID: 1

Die im MQTT SoC-Modul angezeigten Pfade für SoC und Reichweite müssen später in die „Send_to“-Felder der „Automate - Custom PIDs“-Konfiguration der meatPi-Oberfläche eingetragen werden (siehe Punkt 6):

SoC-Modul des Fahrzeugs ? MQTT

SoC direkt aus Fahrzeug auslesen ? Nein ✓ Ja

Aktualisierung der Fahrzeugdaten ?

Während der Ladung ? Min.

Ohne laufende Ladung ? Min.

Nur aktualisieren wenn angesteckt ? Nein ✓ Ja

⚙ Das ausgewählte Fahrzeug-Modul "MQTT" wird von openWB gepflegt.

Einstellungen für SoC-Modul "MQTT"

- `openWB/set/mqtt/vehicle/0/get/soc`
Fahrzeug-SoC mit Nachkommastellen (Float) oder Ganzzahl
Beispiel: 89.5
- `openWB/set/mqtt/vehicle/0/get/soc_timestamp`
Zeitstempel des SoCs in s als Unix-Zeitstempel
Diese Info ist optional. Wird kein Wert für das Topic veröffentlicht, wird bei der Abfrage automatisch der aktuelle Zeitstempel gesetzt.
Beispiel: 1748339524
- `openWB/set/mqtt/vehicle/0/get/range`
Reichweite des Fahrzeugs in km mit Nachkommastellen (Float) oder Ganzzahl
Diese Info ist optional.
Beispiel: 356.5

SoC während der Ladung berechnen ? Nein ✓ Ja

4. Konfiguration WiCAN Settings

- Zunächst muss aber der WiCAN ins Heim-WLAN geholt werden:
<https://meatpihq.github.io/wican-fw/config/wifi>
- Nun werden die weiteren Parameter konfiguriert:

meatPi electronics

Status

Settings

Automate

Monitor

Power Saving

System

About

CAN

CAN Bitrate: 500K

CAN Mode: Normal

Port Type: TCP

TCP/UDP Port: 3333

Protocol: AutoPID

MQTT: Enable

BLE

Passkey: 123456

BLE Status: Disable

MQTT

MQTT URL: 192.168.1.31

MQTT Port: 1883

MQTT User:

MQTT Pass:

TX Topic: others/wican/Auto/can/tx ☒

RX Topic: others/wican/Auto/can/rx ☒

Status Topic: others/wican/Auto/status

openWB IP-Adresse

- Protocol AutoPID wird zur automatischen Abfrage der PIDs benötigt (Bereich Automate)
- MQTT zur openWB erfolgt mit leerem User und Passwort
- RX und Status Topic müssen mit others/ beginnen, damit sie von openWB beachtet werden. Diese dienen nur der Fehleranalyse und Überwachung mit MQTT-Explorer. Für die eigentliche Funktion werden sie nicht benötigt.
- Sleep Mode (Bereich Power Saving): Bei Unterschreiten der Sleep Voltage schaltet sich der WiCAN ab. Hierdurch wird ein Entladen der 12V Batterie verhindert.

5. Ermittlung der PID Parameter aus dem Vehicle Profile bei meatPi

Es werden bei meatPi die benötigten Parameter verschiedener Fahrzeuge in Vehicle-Profilen gesammelt. Die Informationen aus diesen Profilen können wir verwenden, um die Parameter, die wir für die openWB benötigen, zu ermitteln.

Die Profile sind hier einzeln verfügbar:

https://github.com/meatpiHQ/wican-fw/tree/main/vehicle_profiles

Anhand des Beispiels eines VW:ID, die wichtigen Informationen (fettgedruckt). Wir benötigen zwingend den SoC, optional können wir auch die Reichweite (Range) gebrauchen:

```
{ „car_model“:„VW: ID“, „init“:„ATST96;ATFCSD300000;ATFCSM1;“, „pids“:[ { „pid“:„22028C1“,
„pid_init“:„ATSP7;ATCP17;ATSHFC007B;ATFCSH17FC007B;ATCRA17FE007B;“, „parameters“:[ {
„class“:„battery“, „expression“:„B4*0.4425-6.1947“, „name“:„soc“, „unit“:„%“ } ] }, {
„pid“:„222AB62“, „pid_init“:„ATSP6;ATCP18;ATSH710;ATFCSH710;ATCRA77A;“, „parameters“:[ {
„class“:„none“, „expression“:„[B5:B6]“, „name“:„range“, „unit“:„km“ } ] } }
```

6. Übernahme der Werte für die Custom PIDs (Automate-Tab)

Text in Arbeit

From:
<https://wiki.openwb.de/> - **openWB GmbH & Co. KG**

Permanent link:
https://wiki.openwb.de/doku.php?id=openwb:vc:2.3.x:software:fahrzeug-infos:mqtt_soc

Last update: **2026/10/28 21:53**

