

# Inhaltsverzeichnis

- WiCAN-OBd2 SoC** ..... 2
  - WiCAN Dongle ..... 2
  - WiCAN Grundeinstellungen ..... 2
  - openWB MQTT SoC-Modul ..... 5

# WiCAN-OBd2 SoC

Mit Hilfe des WiCAN OBd2 Dongles von meatPi Electronics können die Werte des Fahrzeugs über die OBd2-Schnittstelle ausgelesen und im WLAN an die openWB Wallbox gesendet werden. Hiermit ist die Nutzung ohne Cloud-Dienste der Fahrzeughersteller möglich.

Die erste Umsetzung erfolgte vom 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>

Inzwischen ist es auch möglich, dass der Dongle die Daten direkt per MQTT an die openWB senden oder über HomeAssistant bereitstellen kann. Diese Funktion soll hier näher beschrieben werden.

## WiCAN Dongle

Die Firma meatPi Electronics hat Ihren Sitz in Australien. Für die Beschaffung der Hardware 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 PRO** oder der **WiCAN-OBd** verwendet werden kann.

Für Neubestellungen empfiehlt sich der WiCAN PRO, da er einen besseren Funktionsumfang und Hardwareausstattung bietet. Bei Bestellung über Mouser erfolgt der Versand ab 50€ Versandkostenfrei. Achtung: Die angegebenen Preise sind Nettopreise.

## WiCAN Grundeinstellungen

- Zunächst muss der WiCAN ins Heim-WLAN geholt werden:  
<https://meatpihq.github.io/wican-fw/config/wifi>

Dann erfolgt die Konfiguration:

- Settings:
  - Als Protocol wird AutoPID ausgewählt.
  - MQTT aktivieren.
  - Die MQTT-Verbindung zur openWB erfolgt mit leerem User und Passwort.
  - RX und Status Topics müssen mit others/ beginnen, sonst werden sie vom openWB Broker verworfen. Diese dienen nur der Fehleranalyse und Überwachung mit MQTT-Explorer. Für die eigentliche Funktion werden sie nicht benötigt.

The screenshot shows the 'meatPi electronics' settings interface. On the left is a navigation menu with 'Settings' selected. The main content area is divided into sections: CAN, BLE, and MQTT. The CAN section includes fields for CAN Bitrate (500K), CAN Mode (Normal), Port Type (TCP), and TCP/UDP Port (3333). The MQTT section includes Protocol (AutoPID) and MQTT (Enable), both highlighted with a red box. The BLE section includes Passkey (123456) and BLE Status (Disable). The MQTT section includes MQTT URL (192.168.1.31), MQTT Port (1883), and MQTT User/Pass fields. The TX Topic, RX Topic, and Status Topic fields are also highlighted with a red box. A blue callout bubble points to the MQTT URL field with the text 'openWB IP-Adresse'.

- Power Saving / Sleep Mode: Bei Unterschreiten der Sleep Voltage schaltet sich der WiCAN ab. Hierdurch wird ein Entladen der 12V Batterie verhindert.

The screenshot shows the 'meatPi electronics' settings interface with 'Automate' selected in the navigation menu. The 'Sleep Mode' section is visible, containing: Sleep Mode (Enable), Sleep Voltage (slider at 13.2 V), Sleep After (slider at 5 min), and Battery Alert (Disable). A 'Submit Changes' button is located at the bottom of the section.

- Automate / Fahrzeugspezifische Einstellungen:

Unter Automate ein Fahrzeugprofil (Vehicle Profile) suchen, welches dem eigenen Fahrzeug möglichst ähnelt. Der WiCAN PRO bietet hier gleich die Möglichkeit, die Parameter zu testen:

### Automate Parameters

Standard PIDs   **Vehicle Specific**   User Custom

#### Vehicle Specific PIDs

Vehicle Specific PIDs:

Vehicle Model:

Vehicle Profiles:  Keine ausgewählt

Specific Initialisation:

---

▲ SOC 85 %   Enabled

Name:	<input type="text" value="SOC"/>
PID:	<input type="text" value="2201019"/>
PID Init:	<input type="text" value="Init"/>
Expression:	<input type="text" value="B10/2"/>
Unit:	<input type="text" value="%"/>
Class:	<input type="text" value="battery"/>
Min Value:	<input type="text" value="0"/>
Max Value:	<input type="text" value="100"/>
Period(ms):	<input type="text" value="5000"/>
Destination Type:	<input type="text" value="MQTT_Topic"/>
Destination:	<input type="text" value="openWB/set/mqtt/vehicle/4/get/soc"/>

Die gefundenen Parameter (SOC, ggf. RANGE und ODOMETER) werden nun unter Custom neu eingetragen, damit sie bei einer Änderung im Fahrzeugprofil nicht verloren gehen:



Status

Settings

**Automate**

Monitor

Power Saving

System

About

### Custom PIDs

Custom Initialisation:

ATAT1:ATST96;

#### ▲ SOC Delete

Name:

Init:

PID:

Expression:

Min Value:

Max Value:

Period(ms):

Destination Type:  ▼

Send\_to:

#### ▲ RANGE Delete

Name:

Init:

PID:

Expression:

Min Value:

Max Value:

Period(ms):

Destination Type:  ▼

Send\_to:

## openWB MQTT SoC-Modul

In der Fahrzeugkonfiguration der openWB wird das MQTT SoC Modul eingerichtet. Die im MQTT SoC-Modul angezeigten Pfade für SoC (und ggf. Reichweite und Kilometerstand) werden in die „Send\_to“-Felder der „Automate - Custom“-Konfiguration des WiCAN eingetragen. Als Destination-Type wird MQTT\_Wallbox ausgewählt:

The screenshot shows the openWB interface with the following elements:

- SoC-Modul des Fahrzeugs:** Set to MQTT. A toggle switch for "SoC direkt aus Fahrzeug auslesen" is set to "Nein".
- Aktualisierung der Fahrzeugdaten:** A section with a red box around the title. It includes:
  - "Während der Ladung": 5 Min.
  - "Ohne laufende Ladung": 15 Min.
  - A toggle switch for "Nur aktualisieren wenn angesteckt" is set to "Nein".
- Einstellungen für SoC-Modul "MQTT":** A yellow warning box states that user management must be active. Below it, a list of MQTT topics is shown:
  - `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
  - `openWB/set/mqtt/vehicle/0/get/odometer`: Aktueller Km-Stand des Fahrzeugs in km mit Nachkommastellen (float) oder Ganzzahl. Diese Info ist optional. Beispiel: 20356.5
- Custom PIDs:** A sidebar on the right showing configuration for two PIDs:
  - soc:** Name: soc, Init: ATSP7.ATCP17.ATSHFC007B, PID: 22028C1, Expression: B4\*0.4425-6.27, Min Value: 1, Max Value: 100, Period(ms): 60000. Destination Type: MQTT\_WallBox, Send\_to: openWB/set/mqtt/vehicle/0/get/.
  - range:** Name: range, Init: ATSP6.ATCP18.ATSH710.ATF, PID: 222AB62, Expression: [B5:B6], Min Value: 1, Max Value: 9999, Period(ms): 60000. Destination Type: MQTT\_WallBox, Send\_to: openWB/set/mqtt/vehicle/0/get/.

### Wichtige Hinweise:

- Die Daten werden im Fahrzeugprofil erst im nächsten Aktualisierungsintervall (Während der Ladung / Ohne laufende Ladung) berücksichtigt.
- Da der WiCAN keinen Timestamp senden kann, wird der letzte Zeitstempel im Status in jedem Aktualisierungsintervall erneuert.
- Für Fahrzeuge, bei denen der SoC während der Ladung nicht zuverlässig gesendet wird, muss die Option „SoC während der Ladung berechnen“ aktiviert werden, dann erfolgt bei der Ladung nur eine Berechnung anhand der eingepflegten Fahrzeugdaten im Fahrzeugprofil.

Fragen und Probleme können hier im Forum gestellt werden:

<https://forum.openwb.de/viewtopic.php?t=9397>

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

Permanent link: <https://wiki.openwb.de/doku.php?id=openwb:vc:2.1.9:software:fahrzeug-infos:wican-obd2-soc>

Last update: 2026/04/15 00:25

