

BLANKOVATION

simple & smart solutions



Integrationsanleitung

MQTT

Inhalt

1	FUNKTIONSPRINZIP	2
2	VORAUSSETZUNGEN	2
3	EINRICHTUNG IM TANKBUDDY-WEBINTERFACE	2
4	ÜBERSICHT ALLER VERÖFFENTLICHTEN TOPICS	3
5	MÖGLICHE MQTT-BROKER	3
5.1	EINRICHTUNG IN IOBROKER	3
5.2	EINRICHTUNG IN HOME ASSISTANT.....	4
5.3	EINRICHTUNG MIT EINEM EIGENSTÄNDIGEN BROKER (MOSQUITTO)	5
6	FEHLERBEHEBUNG	6

1 Funktionsprinzip

Der TankBuddy verbindet sich als MQTT-Client zu einem einzigen, vorhandenen MQTT-Broker und veröffentlicht dort in regelmäßigen Abständen seine Messwerte und Statusinformationen. Ein Rückkanal (Steuerung von TankBuddy per MQTT-Befehl) ist nicht vorgesehen – die Integration ist reines Publish-only.

Wichtige technische Rahmenbedingungen:

- Es wird ausschließlich eine unverschlüsselte MQTT-Verbindung unterstützt (Standardport 1883). Broker, die ausschließlich TLS/SSL (üblicherweise Port 8883) verlangen, funktionieren nicht.
- Benutzername/Passwort sind optional – bei einem Broker ohne Authentifizierung einfach leer lassen.
- Jedes TankBuddy-Gerät erzeugt automatisch eine eigene Geräte-ID nach dem Muster esp32-XXXXXX, wobei XXXXXX die letzten 6 Zeichen der WLAN-MAC-Adresse des Geräts sind (Großbuchstaben, ohne Doppelpunkte). Diese ID ist fest und hat nichts mit dem unter WLAN einstellbaren Hostnamen zu tun. Sie lässt sich einfach ablesen: Im Reiter WLAN steht unter dem Hostnamen-Feld der Hinweis „Aufruf im Webbrowser über <http://esp32-XXXXXX>“ – genau dieses esp32-XXXXXX ist die Geräte-ID.
- Alle Topics beginnen mit dem Basis-Topic tankbuddy/esp32-XXXXXX/ (Beispiel: tankbuddy/esp32-61EEA4/tank/fill_percent).
- Werte werden nur bei tatsächlicher Änderung gesendet, maximal alle 5 Sekunden. Direkt nach jedem (Wieder-)Verbindungsaufbau werden zusätzlich einmalig alle Werte komplett gesendet.

2 Voraussetzungen

- Ein erreichbarer MQTT-Broker im Netzwerk (z. B. Mosquitto, der Broker/die Adapter-Funktion von ioBroker, der Mosquitto-Add-on von Home Assistant, oder ein beliebiger anderer Broker).
- TankBuddy (WLAN) muss den Broker per IP und Port erreichen können (keine Client-Isolation im WLAN, keine Firewall-Sperre).
- Empfohlen: feste IP-Adresse bzw. DHCP-Reservierung für den Broker, da der TankBuddy die Adresse fest in den Einstellungen speichert.

3 Einrichtung im TankBuddy-Webinterface

MQTT

☐

Broker-Adresse:

Port:

Benutzername:

Passwort:

1. Im Webinterface den Abschnitt „MQTT“ öffnen (Übersichtskachel „MQTT“ anklicken).
2. Schalter oben aktivieren.
3. Broker-Adresse eintragen (IP-Adresse des Brokers/Servers).
4. Port eintragen (Standard: 1883).
5. Benutzername und Passwort eintragen, falls der Broker Authentifizierung verlangt – sonst leer lassen.
6. Speichern klicken. TankBuddy trennt eine eventuell bestehende Verbindung und baut sofort im Anschluss eine neue auf.

Die Übersichtskachel „MQTT“ zeigt den Status farbig an: grau = deaktiviert, rot = aktiviert, aber nicht verbunden, grün = verbunden.

4 Übersicht aller veröffentlichten Topics

Alle Topics unterhalb von tankbuddy/esp32-XXXXXX/ :

Sensorwerte:

- sensor/distance_cm – gemessene Distanz Sensor–Oberfläche in cm
- sensor/temperature_c – Temperatur in °C

Tankwerte:

- tank/fill_percent – Füllstand in %
- tank/fill_height_cm – Füllhöhe in cm (wird nur bei Tanktyp „Zylinder“, nicht bei „Peiltabelle“, gesendet)
- tank/fill_volume_l – aktuelle Menge in Litern
- tank/max_volume_l – maximales Tankvolumen in Litern

Status:

- status/code – Statuscode (0 = OK, 1 = Warnung, 2 = Fehler)
- status/text – Statustext im Klartext
- status/config_valid – „true“/„false“, ob die Tank-Konfiguration plausibel ist
- status/sensor_valid – „true“/„false“, ob der Sensormesswert gültig ist
- status/calculation_valid – „true“/„false“, ob die Berechnung plausibel ist

Geräteinformationen:

- device/firmware – Firmware-Version
- device/uptime_s – Laufzeit seit Neustart in Sekunden
- device/ip – aktuelle IP-Adresse
- device/wifi_rssi – WLAN-Signalstärke in dBm
- device/restart_reason – Grund des letzten Neustarts

Alle Zahlenwerte werden als Text mit Dezimalpunkt (z. B. „42.3“) übertragen.

5 Mögliche MQTT-Broker

5.1 Einrichtung in ioBroker

Variante A – ioBroker als eigener Broker (einfachster Fall, kein zusätzlicher Broker nötig):

1. Adapter „mqtt“ installieren, falls noch nicht vorhanden.
2. Neue Instanz anlegen, Modus „Server/Broker“ wählen.
3. Port festlegen (Standard 1883), optional Benutzername/Passwort im Adapter aktivieren, wenn Zugangsschutz gewünscht ist.
4. Instanz starten.
5. Im TankBuddy-Webinterface unter MQTT: Broker-Adresse = IP-Adresse des ioBroker-Servers, Port wie im Adapter eingestellt, Benutzername/Passwort nur ausfüllen, wenn im Adapter aktiviert.
6. Speichern.
7. Sobald TankBuddy die erste Nachricht gesendet hat, legt der Adapter automatisch einen Objektbaum unter mqtt.0.tankbuddy.esp32-XXXXXX.* an – keine Discovery oder Zusatzkonfiguration nötig.

Variante B – vorhandener externer Broker (z. B. Mosquitto):

- Adapterinstanz im Modus „Client“ anlegen, IP/Port des externen Brokers sowie ggf. Zugangsdaten eintragen.
- TankBuddy im Webinterface auf denselben Broker (gleiche IP/Port/Zugangsdaten) konfigurieren.
- Ergebnis wie oben: Objektbaum entsteht automatisch.

Hinweis

Der Adapter legt neue Datenpunkte beim ersten Empfang meist als Typ „string“ an. Für numerische Werte (z. B. fill_percent) kann der Datentyp bei Bedarf über „Objekte bearbeiten“ auf „number“ umgestellt werden, damit Diagramme/VIS-Widgets korrekt damit rechnen.

5.2 Einrichtung in Home Assistant

Broker-Verbindung

Falls noch nicht vorhanden: MQTT-Broker einrichten (z. B. das „Mosquitto broker“-Add-on bei Home Assistant OS, oder ein externer Broker), danach unter Einstellungen → Geräte & Dienste → Integration hinzufügen → MQTT die Verbindung zu diesem Broker herstellen. Beim Mosquitto-Add-on empfiehlt es sich, für TankBuddy einen eigenen, nicht-administrativen Home-Assistant-Benutzer anzulegen und dessen Zugangsdaten im TankBuddy-Webinterface unter MQTT einzutragen (der Add-on-Broker authentifiziert über die Home-Assistant-Benutzerverwaltung).

Topics manuell anlegen

Die Topics müssen einmalig per YAML in *configuration.yaml* angelegt werden. Die folgende Struktur kann kopiert und eingefügt werden - XXXXXX jeweils durch die eigene Geräte-ID ersetzen:

```
mqtt:
  sensor:
    - name: "TankBuddy Füllstand"
      unique_id: tankbuddy_XXXXXX_fill_percent
      state_topic: "tankbuddy/esp32-XXXXXX/tank/fill_percent"
      unit_of_measurement: "%"
      state_class: measurement
    - name: "TankBuddy Menge"
      unique_id: tankbuddy_XXXXXX_fill_volume
      state_topic: "tankbuddy/esp32-XXXXXX/tank/fill_volume_l"
      unit_of_measurement: "l"
      state_class: measurement
    - name: "TankBuddy Distanz"
      unique_id: tankbuddy_XXXXXX_distance
      state_topic: "tankbuddy/esp32-XXXXXX/sensor/distance_cm"
      unit_of_measurement: "cm"
      state_class: measurement
    - name: "TankBuddy Temperatur"
      unique_id: tankbuddy_XXXXXX_temperature
      state_topic: "tankbuddy/esp32-XXXXXX/sensor/temperature_c"
      unit_of_measurement: "°C"
      device_class: temperature
      state_class: measurement
    - name: "TankBuddy Status"
      unique_id: tankbuddy_XXXXXX_status_text
      state_topic: "tankbuddy/esp32-XXXXXX/status/text"
    - name: "TankBuddy WLAN-Signal"
      unique_id: tankbuddy_XXXXXX_rssi
      state_topic: "tankbuddy/esp32-XXXXXX/device/wifi_rssi"
      unit_of_measurement: "dBm"
      device_class: signal_strength
      state_class: measurement
      entity_category: diagnostic
  binary_sensor:
    - name: "TankBuddy Konfiguration gültig"
      unique_id: tankbuddy_XXXXXX_config_valid
      state_topic: "tankbuddy/esp32-XXXXXX/status/config_valid"
      payload_on: "true"
      payload_off: "false"
      entity_category: diagnostic
```

Danach die Konfiguration übernehmen: Entwicklerwerkzeuge → YAML → „MQTT“ neu laden (oder bei Unsicherheit Home Assistant komplett neu starten).

Profi-Tipp: Fügt man jedem Sensor denselben device-Block hinzu, werden alle Werte in Home Assistant zu einem gemeinsamen Gerät zusammengefasst statt als einzelne, unverbundene Entitäten zu erscheinen:

```
device:
  identifiers: "tankbuddy_XXXXXX"
  name: "TankBuddy Zisterne"
  manufacturer: "TankBuddy"
```

5.3 Einrichtung mit einem eigenständigen Broker (Mosquitto)

Wer weder ioBroker noch Home Assistant einsetzt – etwa bei Nutzung von Node-RED, FHEM, openHAB oder eigenen Skripten – kann stattdessen einen eigenständigen Mosquitto-Broker betreiben. Mosquitto ist Open Source, sehr ressourcenschonend und läuft z. B. auf einem Raspberry Pi, in einem Docker-Container (NAS, Server) oder unter Windows/Linux.

Installation (Beispiel Raspberry Pi / Debian-basiertes Linux):

1. `sudo apt update`
2. `sudo apt install mosquitto mosquitto-clients`
3. Mosquitto startet danach automatisch als Dienst und lauscht standardmäßig auf Port 1883.

Installation als Docker-Container (z. B. auf einer NAS):

1. Image „eclipse-mosquitto“ verwenden, Port 1883 auf den Host durchreichen.
2. Eine minimale `mosquitto.conf` bereitstellen, da Mosquitto ab Version 2 ohne Konfigurationsdatei keine Verbindungen von außen mehr zulässt: `listener 1883`
`allow_anonymous true` (`allow_anonymous true` nur für den ersten Test bzw. in abgesicherten Netzen verwenden – siehe nächster Punkt)

Zugangsschutz einrichten (empfohlen):

1. Benutzer anlegen: `sudo mosquitto_passwd -c /etc/mosquitto/passwd tankbuddy`
2. In der `mosquitto.conf` ergänzen: `allow_anonymous false password_file /etc/mosquitto/passwd`
3. Dienst neu starten: `sudo systemctl restart mosquitto`

Einrichtung im TankBuddy:

1. Im Webinterface unter MQTT: Broker-Adresse = IP-Adresse des Geräts, auf dem Mosquitto läuft.
2. Port = 1883 (bzw. abweichend, falls geändert).
3. Benutzername/Passwort wie oben angelegt eintragen, bzw. leer lassen, falls `allow_anonymous true` verwendet wird.
4. Speichern.

Verbindung testen: Auf dem Rechner mit Mosquitto (oder einem anderen Gerät im Netzwerk mit installierten `mosquitto-clients`) folgenden Befehl ausführen, um alle TankBuddy-Topics live mitzulesen: `mosquitto_sub -h <Broker-IP> -u <Benutzer> -P <Passwort> -t "tankbuddy/#" -v`. Erscheinen dort laufend Zeilen wie „`tankbuddy/esp32-XXXXXX/tank/fill_percent 42.3`“, ist die Verbindung korrekt eingerichtet. Die Geräte-ID `esp32-XXXXXX` lässt sich wie in Abschnitt 1 beschrieben im Reiter WLAN des TankBuddy-Webinterface ablesen.

Hinweis

Ein separat betriebener Mosquitto-Broker lässt sich jederzeit nachträglich auch an Home Assistant (MQTT-Integration) oder ioBroker (MQTT-Adapter im Modus „Client“) anbinden, ohne dass an TankBuddy etwas geändert werden muss – der TankBuddy kennt nur den Broker, nicht die dahinterliegende Plattform.

6 Fehlerbehebung

Status bleibt dauerhaft rot:

- Broker-Adresse/Port falsch, oder Broker läuft nicht.
- Firewall/Client-Isolation verhindert Erreichbarkeit zwischen WLAN und Broker.
- Falsche Zugangsdaten, oder Broker verlangt TLS (wird nicht unterstützt).

Status ist grün, aber es kommen keine Werte an:

- Falscher Basis-Topic bzw. falsche Geräte-ID verwendet – die Groß-/Kleinschreibung und die 6-stellige MAC-Endung genau prüfen.
- In Home Assistant: Sensor zeigt „Unbekannt“, weil seit dem letzten Neustart von Home Assistant noch keine Wertänderung gesendet wurde – MQTT in TankBuddy kurz aus- und wieder einschalten, um einen sofortigen Vollständig-Publish zu erzwingen.

Füllhöhe/Menge fehlen oder sind unplausibel:

- Tank-Geometrie im Webinterface noch nicht korrekt konfiguriert (unabhängig von MQTT).
fill_height_cm wird zudem bei Tanktyp „Peiltabelle“ grundsätzlich nicht gesendet.