MQTT-Handbuch
für eviateg-Geräte

Copyright © 2026 eviateg GmbH

Alle Angaben in diesem Buch entsprechen dem technischen Stand bei Drucklegung, gelten jedoch nicht als Zusicherung von Produkteigenschaften. Die eviateg GmbH haftet in dem Umfang, der in den Allgemeinen Geschäftsbedingungen festgelegt ist.

Die eviateg GmbH übernimmt keine Gewähr für technische Ungenauigkeiten und behält sich vor, Änderungen zum Zwecke des technischen Fortschritts vorzunehmen.

Die neueste Version dieses Handbuches ist über die Download-Seite unserer Website abrufbar. Alle Erweiterungen gegenüber älteren Versionen dieses Handbuches befinden sich in der Historie auf Seite 30.

Handbuch Version 1.7

Firmwareversion 6.08

Henstedt-Ulzburg, 16.01.2026

eviateg GmbH

Kleistring 8

D-24558 Henstedt-Ulzburg

Internet: https://www.eviateg.de

E-Mail: info@eviateg.de

Warenzeichen: eviateg™ ist eingetragenes Warenzeichen der eviateg GmbH.
Windows™ ist eingetragenes Warenzeichen der Microsoft Corporation
IBM ist eingetragenes Warenzeichen der International Business Machines Corporation.
Shelly ist ein Warenzeichen der ALLTERCO ROBOTICS EOOD.
Alle anderen Warenzeichen sind Warenzeichen der jeweiligen Eigentümer

Inhaltsverzeichnis

1 MQTT - das Protokoll für das "Internet der Dinge" 5
1.1 Einleitung 5
2 MQTT-Clients 5
2.1 Bereitstellen von Daten 5
2.2 Abonnieren von Daten 6
3 MQTT-Broker 6
3.1 Punkt-zu-Punkt-Verbindung 6
3.2 Einer-zu-Vielen 7
3.3 Viele-zu-Einem 7
3.4 Aufbau von Broker und Clients 7
3.5 Verbindungen zwischen Clients und Broker 7
3.6 Bridges 7
3.7 Datensicherheit 8
4 Begriffe aus dem MQTT-Protokoll 9
4.1 Dienstqualität 9
4.2 Das Retain-Flag 9
4.3 Das "Testament" (der "letzte Wille") 9
4.4 Topic-Namen 9
4.5 Wildcards 10
4.5.1 Single level wildcard 10
4.5.2 Multi level wildcard 10
4.6 Verbindungsüberwachung 10
5 Der MQTT-Client in den eviateg-Geräten 12
5.1 Struktur der Topics 12
5.2 Struktur der Topic-Namen bei MBus-Slaves 13
5.3 Abonnierte Topics 13
5.3.1 Steuerbefehle 13
5.3.2 Steuerung von Ausgängen 14
5.4 Publizierte Topics 14
5.4.1 Topics mit digitalen Zuständen 14
5.4.2 Topics mit numerischen Werten 14
5.4.3 Zähler 15
5.4.4 MBus-Datensätze 15
5.5 Parametrierung von Eingängen und Wächtern 15

5.6 Parametrierung für numerische Werte....16
6 Programmgesteuertes Publizieren....17
7 Anwendungsbeispiele....18
7.1 Eingänge eines Gerätes schalten Ausgänge eines anderen Gerätes....18
7.2 Schalten von Lampen bei Sonnenuntergang und Sonnenaufgang....18
7.3 Anruf soll Relais einschalten....19
7.4 Steuerung durch Verknüpfungen....19
7.4.1 Verknüpfung von Eingangszuständen....19
7.4.2 Verknüpfungen von Flags....20
7.4.3 Weitere logische Verknüpfungen....20
7.5 Alarmierungen und Steuerbefehle durch eine SPS....20
8 Einrichten eines eigenen MQTT-Brokers....22
8.1 Einrichten eines MQTT-Brokers auf einem Windows-Rechner....22
8.1.1 Installation....22
8.1.2 Einrichten von Benutzern und Passwörtern (empfohlen)....22
8.1.3 Starten des Brokers als Dienst....23
8.1.4 Betrieb hinter einem Router....23
8.1.5 Testen des Brokers....23
8.2 Einrichten eines MQTT-Brokers auf einem Raspberry Pi....23
8.2.1 Installation....23
8.2.2 Einrichten von Benutzern und Passwörtern (empfohlen)....23
8.2.3 Betrieb hinter einem Router....24
8.2.4 Testen des Brokers....24
9 Empfehlenswerte Zusatzprogramme....25
9.1 MQTT-Explorer....25
9.1.1 Eintragen der Zugangsdaten zum Broker....25
9.1.2 Abonnieren von Topics....25
9.1.3 Publizieren von Topics....25
10 Beispiele für MQTT-Topics bei eviateg-Geräten....26
10.1 IT180 (mit zwei MBus-Slaves)....26
10.2 IT70....27
10.3 IT251....28
10.4 IT35G....29
11 Historie....30
Stichwortverzeichnis....31

1 MQTT - das Protokoll für das "Internet der Dinge"

1.1 Einleitung

MQTT ist ein offenes Netzwerkprotokoll für die intelligente Vernetzung von zwei oder mehreren Geräten ("M2M", Machine-to-Machine").

Der Begriff leitet sich aus "MQ Telemetry Transport" ab, wobei "MQ" eine Geräteserie von IBM bezeichnet.

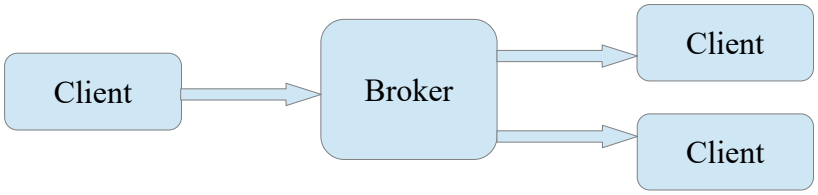
Vielfach wird MQTT auch mit "Message Queue Telemetry Transport" übersetzt.

MQTT-fähige Geräte umfassen u.a.

- **Sensoren** wie Schalter, Temperaturfühler, Fenster-Kontakte, Bewegungsmelder etc.
- **Aktoren** wie Relais, schaltbare Netzsteckdosen, dimmbare Glühlampen etc.

MQTT-fähige Geräte kommunizieren nicht direkt miteinander, sondern über einen Server, den sogenannten "**Broker**" (Makler, Verteiler).

Die Geräte werden (unabhängig davon, ob sie die Sensor- oder die Aktor-Rolle einnehmen) unter dem Begriff "**MQTT-Client**" zusammengefasst.



2 MQTT-Clients

2.1 Bereitstellen von Daten

Der Vorgang, dass ein Client Daten eines Sensors an den Broker sendet, wird mit dem Begriff "**Publish**" (Publizieren) bezeichnet.

Dazu kennzeichnet der publizierende Client die Daten mit einer Bezeichnung des Sensors, dem sogenannten "**Topic**" oder "Thema", und dem jeweiligen Zustand des Sensors (meistens als "**Dateninhalt**", "**Message**" oder "**Payload**" bezeichnet).

Ein Topic kann (ähnlich der Unterverzeichnisstruktur eines Rechners) **strukturiert** aufgebaut werden. Dazu werden die einzelnen Ebenen durch einen Schrägstrich '/' voneinander getrennt, z.B. `Pumpwerk_123/Eingaenge/Pumpe_3/Laufzeit`.

Bei komplexeren zu publizierenden Daten kann für den Dateninhalt die **JSON**-Notation verwendet, z.B.

```
{
    "ison": false,
    "mode": "color",
    "red": 0,
    "green": 0,
```

```
    "blue": 255,
    "white": 0,
    "gain": 100,
    "temp": 4750,
    "brightness": 100,
    "effect": 0
}
```

2.2 Abonnieren von Daten

Der umgekehrte Vorgang, dass ein Client auf diese Daten zugreifen möchte (also quasi abonnieren möchte), wird mit dem Begriff "**Subscribe**" bezeichnet.

Dazu gibt er die Topics der Sensoren an, die er weiterverarbeiten möchte.

Dabei kann er einen oder mehrere "**Joker**" ("**Wildcards**") in die Subscribe-Anfragen einfügen, z.B.

- `Pumpwerk_123/Eingänge/+/Laufzeit`, um die Laufzeit aller Pumpen zu abonnieren
- `+/Eingänge/Pumpe_3/Zustand`, um die Zustände der Pumpe_3 bei allen Anlagen zu abonnieren
- `Pumpwerk_123/#`, um alle Meldungen dieser Anlage zu abonnieren

Wenn ein Client die Daten eines oder mehrerer andere Clients nicht mehr abonnieren möchte, gibt es auch den umgekehrten Vorgang ("**Unsubscribe**").

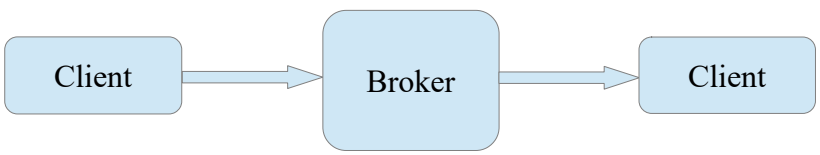
3 MQTT-Broker

Die Aufgabe des Brokers ist es, die Daten entsprechend zwischen den beteiligten Clients zu verteilen.

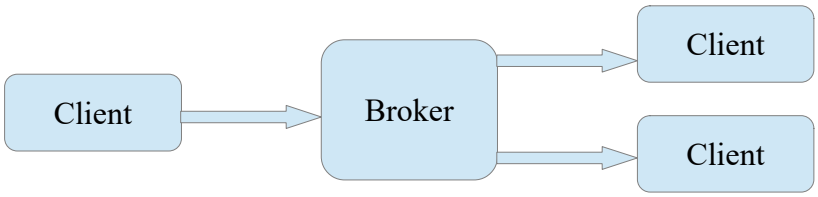
Dadurch ergeben sich folgende Möglichkeiten an Kommunikationsformen:

1. **Punkt-zu-Punkt**: ein Publish-Client übermittelt Daten an einen Subscribe-Client
2. **Einer-zu-Vielen**: ein Publish-Client übermitteln Daten an mehrere Subscribe-Clients
3. **Viele-zu-Einem**: mehrere Publish-Clients übermitteln Daten an einen Subscribe-Client

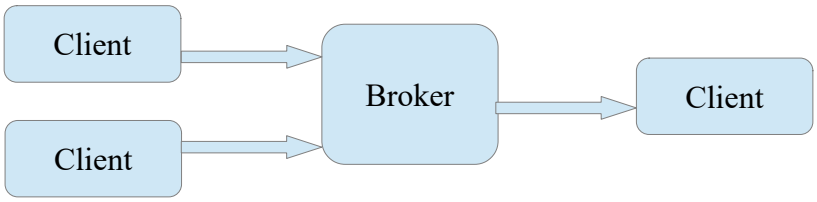
3.1 Punkt-zu-Punkt-Verbindung



3.2 Einer-zu-Vielen



3.3 Viele-zu-Einem



3.4 Aufbau von Broker und Clients

Broker gibt es in vielen Größen und Bauformen - vom Raspberry Pi im Hutschienen-Gehäuse bis hin zum dedizierten Windows-Server.

Client-Lösungen gibt es nicht nur PC-basierend, sondern mittlerweile auch als Apps unter **iOS** und **Android**, so dass z.B. Schaltvorgänge auch von unterwegs aus ausgelöst werden können.

So bietet die iOS-App "**EasyMQTT**" die Möglichkeit, Topics per Kurzbefehl und per Siri-Sprachsteuerung zu senden oder den Verlauf von Messwerten als Grafik dargestellt zu bekommen.

3.5 Verbindungen zwischen Clients und Broker

Die Verbindungen zwischen Client und Broker werden typischerweise über TCP/IP-Netze (LAN, **GPRS**) als ständige Verbindungen aufgebaut.

So gibt es z.B. schaltbare Steckdosen und Leuchtmittel, die sich per WLAN mit dem Broker verbinden und per entsprechender Topics ein- und ausgeschaltet bzw. gedimmt werden können.

3.6 Bridges

Als Interface zu preiswerten **ZigBee**-kompatiblen Endgeräten gibt es ZigBee-Bridges (z.B. "zigbee2mqtt"), die sich dem MQTT-Broker gegenüber wie ein Client verhalten und dem ZigBee-Netz gegenüber wie ein ZigBee-Koordinator.

3.7 Datensicherheit

Die Anmeldung eines Clients beim Broker kann optional mittels Benutzernamen und Passwort erfolgen.

Außerdem kann die Datenverbindung mittels **TLS** verschlüsselt werden.

4 Begriffe aus dem MQTT-Protokoll

4.1 Dienstqualität

MQTT-Nachrichten werden mit einer definierbaren Dienstqualität (Quality of Service, QoS) versendet:

QoS 0	"at most once"	die Nachricht wird einmal gesendet und kommt bei Verbindungsunterbrechung möglicherweise nicht an
QoS 1	"at least once"	die Nachricht wird so lange gesendet, bis der Empfang bestätigt wird, und kann beim Empfänger mehrfach ankommen
QoS 2	"exactly once"	hierbei wird sichergestellt, dass die Nachricht auch bei Verbindungsunterbrechung genau einmal ankommt

4.2 Das Retain-Flag

Mit dem sogenannten Retain-Flag kann der Broker angewiesen werden, die Nachricht zu diesem Topic zwischenzuspeichern.

Clients, die dieses Thema neu abonnieren, bekommen als erstes die zwischengespeicherte Nachricht zugestellt.

4.3 Das "Testament" (der "letzte Wille")

Damit ein Broker alle abonnierenden Clients darüber informieren kann, dass die Verbindung zu einem anderen, publizierenden Client unvorhergesehenerweise unterbrochen wurde, kann jeder Client beim Verbindungsaufbau eine Art "Testament" ("**Last Will**") mit angeben.

Wenn sich ein Client nach einer einstellbaren Zeit nicht mehr meldet, publiziert der Broker im Namen dieses Clients dessen "letzten Willen", z.B. das Topic `Pumpwerk_123/Verbindung` mit dem Dateninhalt `offline`.

Logischerweise sollte dieser Client direkt nach erfolgreichem Verbindungsaufbau zum Broker das Topic `Pumpwerk_123/Verbindung` mit dem Dateninhalt `online` publizieren.

4.4 Topic-Namen

Die Namen der Topics werden im **UTF8**-Format übertragen. Damit sind Namen wie z.B. `Eingänge` und `Analog-Ausgänge` möglich.

Manche Clients stellen allerdings Sonderzeichen nicht richtig dar. Deshalb sollte in solchen Fällen besser `Eingaenge` statt `Eingänge` verwendet werden.

Ebenso sollten unbedingt Leertasten im Topic-Namen vemieden werden.

Bei Topic-Namen wird zwischen Groß- und Kleinscheibung unterschieden.
`Out` und `out` sind also nicht das Gleiche !

Ein Topic-Name wird typischerweise durch das Zeichen '/' in mehrere Ebenen unterteilt (ähnlich der Pfadangaben beim PC), z.B. `eviateg/RZ_5B/Ausgänge/Out03`.

Auf ein '/' am Anfang eines Topic-Namens sollte unbedingt verzichtet werden, weil dadurch quasi eine "leere" Ebene entstehen würde.

4.5 Wildcards

Beim Abonnieren eines Topics kann ein Client Joker-Zeichen ("Wildcards") für eine oder mehrere Ebenen einsetzen:

4.5.1 Single level wildcard

Mit dem Zeichen '**+**' in der Topic-Bezeichnung wird der Broker angewiesen, alle Nachrichten ohne Filterung auf der entsprechenden Ebene an den abonnierenden Client zu schicken:

Beispiel:

Ein Client abonniert `eviateg/+/Alarme/Spannung/Pwr24`.

Der Broker schickt diesem Client die Nachrichten
`eviateg/IT180_20BAC1/Alarme/Spannung/Pwr24`
und
`eviateg/PW_Meierstrasse/Alarme/Spannung/Pwr24`

4.5.2 Multi level wildcard

Mit dem Zeichen '**#**' in der Topic-Bezeichnung wird der Broker angewiesen, alle Nachrichten ohne Filterung auf der entsprechenden Ebene **und allen darunterliegenden Ebenen** an den abonnierenden Client zu schicken.

Das Zeichen '**#**' kann in der Topic-Bezeichnung **nur an der letzten Stelle** stehen !

Beispiel 1:

Ein Client abonniert `eviateg/Halle3/Inputs/#`.

Der Broker schickt diesem Client die Nachrichten aller Eingänge von Halle3, z.B.
`eviateg/Halle3/Inputs/IN00`
und
`eviateg/Halle3/Inputs/IN03`

Beispiel 2:

Ein Client abonniert `eviateg/+/Alarme/#`.

Der Broker schickt ihm daraufhin alle Alarme aller Geräte.

4.6 Verbindungsüberwachung

Ein Client gibt beim Verbindungsaufbau zum Broker eine Zeit für die Verbindungsüberwachung an ("Keep Alive time").

Der Wert wird in Sekunden bis zu einer Maximaldauer von 18 Stunden, 12 Minuten und 15 Sekunden angegeben.

Wenn ein Broker innerhalb des Eineinhalbfachen dieser Zeit keine Nachricht vom Client erhalten hat, trennt er die Verbindung.

Ein Client muss also innerhalb dieser Zeit ein Paket senden (ggf. ein PING-Request), um die Verbindung aufrechtzuerhalten.

Ein Wert von 60 Sekunden dürfte für die meisten Fälle angemessen sein.

5 Der MQTT-Client in den eviateg-Geräten

Ab der Firmware-Version 5.16 ist auf folgenden Geräten ein MQTT-Client verfügbar:

- IT35G
- IT70
- IT180
- IT181
- IT251GI
- IT251LI

Der Client stellt folgende Funktionalitäten bereit:

- Publizieren von **Zustandsmeldungen** für Eingänge, Temperaturwächter etc.
- Publizieren von **Alarmen** für Eingänge, Temperaturwächter etc.
- Publizieren von **Messwerten** für Temperatursensoren, Analog-Eingänge etc.
- **Programmgesteuertes Publizieren** von Nachrichten als **Reaktion auf ein Ereignis**, z.B. bei Ablauf eines Timers, bei einem Anrufe etc. - auch in Verbindung mit der Prüfung von Bedingungen und Flag
- Abonnieren eines Topics zum **Schalten von digitalen Ausgängen** und zum Einstellen von Spannungen an **analogen Ausgängen**
- Abonnieren eines Topics zur **Auswertung von Befehlen**, z.B. zum Ändern von Rufnummern für Sprachalarmierungen
- Anmeldung beim Broker mit einem "Last Will" zum Überprüfen des Verbindungszustandes dieses Clients
- Optionale Verschlüsselung mit **TLS** (ohne Verwendung von Zertifikaten)

Die Parametrierung des Clients erfolgt mit der Konfigurationssoftware **CONNY**, die kostenfrei von der eviateg-Download-Seite heruntergeladen werden kann.

Eine Parametrierung mit der einfachen Konfigurationssoftware QuickSetup ist nicht möglich.

5.1 Struktur der Topics

Der MQTT-Client verwendet Topic-Namen mit der folgenden Struktur:

- einem Präfix (Werkseinstellung: "eviateg")
- dem Gerätenamen (Werkseinstellung: Gerätetyp und Seriennummer)
- einer weiteren Ebene, z.B. "Eingänge"
- dem Namen des Eingangs, Ausgangs etc., z.B. "IN00"

Beispiel:

```
eviateg/IT251GI_0020A384/Ausgänge/OUT05
```

Der **Präfix**, der **Gerätename** und die **Namen** der verschiedenen **Topic-Ebenen** können mit CONNY auf der Eigenschaftenseite "**MQTT**" angepasst werden.

Die **Namen** von Eingängen, Ausgängen, Temperaturwächter etc. können auf den jeweiligen Eigenschaftenseiten, z.B. "Ein-/Ausgänge / Digital-Eingänge", angepasst werden.

5.2 Struktur der Topic-Namen bei MBus-Slaves

Bei Geräten mit MBus-Master-Funktion haben die Topic-Namen folgende Struktur:

- einem Präfix (Werkseinstellung: "eviateg")
- dem Gerätenamen (Werkseinstellung: Gerätetyp und Seriennummer)
- der Ebene "MBus"
- dem Namen des MBus-Slaves, z.B. "FIN_23007990"
- dem Namen des Datensatzes auf dem MBus-Slave, z.B. "Netzspannung"

Die Namen der MBus-Slaves ergeben sich in der Werkseinstellung aus der dreistelligen Herstellerkennung und der Seriennummer. Sie können mit **CONNY** auf der Eigenschaftenseite "M-Bus" angepasst werden.

Die Namen der Datensätze des MBus-Slaves werden in der Werkseinstellung durchnummeriert und können mit **CONNY** auf der Eigenschaftenseite "M-Bus" angepasst werden.

Beispiel:

`eviateg/IT180_00209F0F/MBus/FIN_23007990/Netzspannung`

5.3 Abonnierte Topics

Der MQTT-Client der eviateg-Geräte abonniert folgende Topics:

- ein Topic für **Steuerbefehle**, z.B. `eviateg/IT70_20B123/cmd`
- ein Topic zur **Steuerung von Ausgängen**, z.B. `eviateg/PW_Nord/out/#`

`cmd` und `out` können mit CONNY auf der Eigenschaftenseite "**MQTT**" angepasst werden.

Beispiele für die abonnierten Topics sind im Abschnitt 10, Seite 26ff, aufgeführt.

5.3.1 Steuerbefehle

Steuerbefehle an `cmd` werden mit einer Antwort an das Topic für Befehlsantworten (z.B. `eviateg/PW_Nord/Ergebnis`) mit `OK` oder `ERROR` beantwortet.

`Ergebnis` kann mit CONNY auf der Eigenschaftenseite "**MQTT**" angepasst werden.

Wenn der Steuerbefehl eine Textausgabe bewirkt, werden zwei Nachrichten vom Gerät publiziert: die Textausgabe sowie die Antwort `OK`.

Beispiel:

Eine Nachricht `Number 3 ?` an `eviateg/Halle3/cmd` führt zum Publizieren von `017123...` sowie (als zweite Nachricht) von `OK` an `eviateg/Halle3/Ergebnis`.

5.3.2 Steuerung von Ausgängen

Der Client empfängt alle Nachrichten an `.../out/#` zum Schalten von Ausgängen.

Der Ausgang kann entweder numerisch oder anhand seines Namens angegeben werden.

Der Schaltzustand kann mit `0` bzw. `1`, `true` oder `false`, `on` oder `off` angegeben werden.

Beispiel:

`eviateg/PW_Nord/out/1` mit der Payload `1` schaltet das Relais "OUT01" ein.

Als "Rückmeldung", dass der Ausgang geschaltet wurde, publiziert das Gerät den Zustand mit dem entsprechenden Topic, z.B. mit `eviateg/PW_Nord/Ausgänge/OUT01` und `ein`.

5.4 Publizierte Topics

Die Bezeichnungen für Topic-Ebenen wie "Eingänge" oder "Zähler" können mit CONNY auf der Eigenschaftenseite "**MQTT**" in der Tabelle "**Publizierte Topics**" in der Spalte "**Name der Topic-Ebene**" geändert werden.

5.4.1 Topics mit digitalen Zuständen

Für alle **digitalen Eingänge** und **Wächter-Funktionen** (Temperatur-Wächter, Spannungs-Wächter) können der jeweilige digitale **Zustand** und / oder die Info, ob ein **Alarm- oder Ruhezustand** vorliegt, publiziert werden.

Beispiel:

IN03 ist als Öffner (NC, normally closed) beschaltet.

Beim **Öffnen des Kontaktes** kann

- eine **Zustandsmeldung** `eviateg/Halle3/Eingänge/IN03` und der Payload `geöffnet` **und / oder**
- eine **Alarmmeldung** `eviateg/Halle3/Alarme/IN03` und der Payload `ausgelöst`

publiziert werden.

Beim **Schließen des Kontaktes** kann

- eine **Zustandsmeldung** `eviateg/Halle3/Eingänge/IN03` und der Payload `geschlossen` **und / oder**
- eine **Rücksetzmeldung** `eviateg/Halle3/Alarme/IN03` und der Payload `beendet`

publiziert werden.

Die Möglichkeit, Alarm- und Rücksetzmeldungen zu publizieren, wurde wegen der Möglichkeit geschaffen, solche Meldungen mit `eviateg/+/Alarme/#` zu abonnieren.

5.4.2 Topics mit numerischen Werten

Folgende Topics werden einmal pro Minute mit dem jeweiligen Mittelwert publiziert:

- **Temperaturwerte**, z.B. `eviateg/Halle3/Temperatursensoren/Temp3`
- **Spannungen**, z.B. `eviateg/IT70_20B123/Spannungswerte/Akku`

– **GSM-Feldstärke**, z.B. `eviateg/IT70_20B123/Feldstärke`

Als Payload wird der Wert publiziert.

Mit CONNY kann auf der Eigenschaftenseite "**MQTT**" in der Tabelle "**Publizierte Topics**" in der Spalte "**Einheiten**" gewählt werden, ob **zusätzlich** zum Wert auch die **Einheit** mit publiziert werden soll, z.B.

– `°C` für Temperaturwerte

– `V` für Spannungen oder `mA` für Ströme

Diese Möglichkeit unterstützt die Auswertung von Werten durch abonnierende Clients, z.B. für eine grafische Verlaufsanzeige, die ausschließlich numerische Werte verarbeiten kann.

5.4.3 Zähler

Zähler (Counter) werden im Moment der Veränderung publiziert; eine Ausnahme bilden Zähler in der Betriebsart **Betriebsdauer-Zähler**, deren Wert einmal pro Minute publiziert werden, weil die sekündliche Änderung zu viel Datenverkehr erzeugen würde.

Als Payload wird der **Zählerwert** bzw. die **Zeitdauer** in Wochen / Tagen / Stunden / Minuten / Sekunden übertragen, z.B. `eviateg/Halle3/Zähler/Counter00`.

Wenn ein Zähler in einer Betriebsart betrieben wird, die **Ereignisse** erzeugt (z.B. beim Erreichen von Ober- oder Untergrenzen), können für diesen Zähler auch **Alarm- und Rücksetzmeldungen** publiziert werden, z.B. `eviateg/Halle3/Alarme/Counter00`.

5.4.4 MBus-Datensätze

Damit die Werte von Datensätzen eines MBus-Slaves publiziert werden, müssen sie mit **CONNY** auf der Eigenschaftenseite "M-Bus" freigegeben werden.

Erweitern Sie den Baum des jeweiligen MBus-Slaves und des jeweiligen Datensatzes, so dass der Zweig "Meldungen" sichtbar wird. Markieren Sie unterhalb dieses Zweiges den Eintrag so, dass "Änderungen werden auf dem MQTT-Broker publiziert" in Fettschrift dargestellt wird.

5.4.5 Nicht benötigte Topic-Ebenen

Wenn bestimmte Topic-Ebenen (z.B. die Spannungen von Analog-Eingängen oder die GSM-Feldstärke) nicht publiziert werden sollen, kann mit mit CONNY auf der Eigenschaftenseite "**MQTT**" in der Tabelle "**Publizierte Topics**" die Zelle mit dem entsprechenden Namen (z.B. "Analogeingänge") gelöscht werden.

Dieses gilt insbesondere für die Topics, für die keine anderweitigen Einstellmöglichkeiten vorhanden sind.

Das Filtern nicht benötigter Topic-Ebenen wird ab Firmware-Version 6.08 unterstützt.

5.5 Parametrierung von Eingängen und Wächtern

Das Verhalten beim Publizieren von Topics mit digitalen Werten kann mit CONNY auf der betreffenden Eigenschaftenseite (z.B. "Ein-/Ausgänge / Temperatur-Wächter") beeinflusst werden:

Die Dienstequalität mit einer Auswahlliste zwischen QoS 0 und QoS 2 eingestellt werden.

Das Retain-Flag kann mit einem Auswahlkästchen gesteuert werden.

Mit jeweils einer Auswahlliste kann für **Alarm-/Rücksetzmeldungen** und für **Statusmeldungen** der Payload-Text vorgegeben werden, z.B.

- 0 / 1
- ausgelöst / zurückgesetzt
- offen / geschlossen

Wenn die jeweilige Meldung nicht benötigt wird, kann in der Auswahlliste "nicht publiziert" ausgewählt werden.

Hinweis:

Die Namen von Eingängen und Wächtern kann durch einen Klick mit der rechten Maustaste auf die Reiter mit den Namen geändert werden.

5.6 Parametrierung für numerische Werte

Für die zyklisch publizierten Werte (Analog-Eingänge, Temperatur-Sensoren etc.) kann mit CONNY auf der Eigenschaftenseite "**MQTT**" in der Tabelle "**Publizierte Topics**" in der Spalte "**Einheiten**" gewählt werden, ob zusätzlich zum Wert auch die Einheit mit publiziert werden soll.

Mit einem Klick auf die jeweilige Zelle wird ein Häkchen ein- oder ausgeblendet.

6 Programmgesteuertes Publizieren

Mit Hilfe des Befehles **SendMQTT** können ereignisgesteuert MQTT-Nachrichten über den Broker versendet werden.

Anwendungsbeispiele dafür sind:

- Eingänge eines Clients, die Ausgänge eines anderen Clients schalten
- uhrzeitgesteuertes Schalten von Lampen
- Öffnen einer Schranke durch Anrufe

Syntax:

SendMQTT [QoS <QoS>] [Retain] <Topic> <Payload>

Die Parameter **QoS** und **Retain** sind optional.
Die Parameter-Bezeichner können bis auf '**q**' und '**r**' abgekürzt werden.

Der **Topic-Name** muss eindeutig angegeben werden,

Der **Payload-Text** kann **Text-Makros** enthalten, die zum Zeitpunkt des Sendens dynamisch erweitert werden, z.B. "&DT" für Datum und Uhrzeit oder "&CN" für die Rufnummer eines Anrufers.

Der SendMQTT-Befehl kann mit CONNY in den entsprechenden Textfeldern für die Reaktionen auf Ereignisse oder Aktionsmakros eingegeben werden.

Die verwendeten Topics werden von CONNY **in einer Auswahlliste gespeichert**, so dass bei für der nächsten Eingabe eines SendMQTT-Befehls wieder zur Verfügung stehen.

Beispiele:

```
SendMQTT QoS 1 Retain Delock/Switch1/cmnd/POWER off
SendMQTT q 2 r shellies/Shelly/light/0/command on
```

7 Anwendungsbeispiele

7.1 Eingänge eines Gerätes schalten Ausgänge eines anderen Gerätes

Eine der häufigsten Anforderungen ist die Kopplung von Anlagen, bei der der Zustand von Eingängen am Gerät A auf entsprechende Ausgänge des Gerätes B ´gespiegelt werden soll.

Wenn z.B. zwei IT70 miteinander verbunden werden sollen, um zwei Eingangssignale auf die Relais des Partnergerätes abzubilden, müssen mit CONNY in die jeweiligen Eingangsreaktionen mit Hilfe des Assistenten entsprechende SendMQTT-Befehle eingetragen werden:

In die Reaktion beim Schließen von IN00 müsste der SendMQTT-Befehl wie folgt eingetragen werden:

```
SendMQTT eviateg/GerätB/out/0 1
```

Der Befehl für die Reaktion beim Öffnen von IN00 müsste dann lauten:

```
SendMQTT eviateg/GerätB/out/0 0
```

Das Schließen von IN01 müsste folgendermaßen weitergeleitet werden:

```
SendMQTT eviateg/GerätB/out/1 1
```

und das Öffnen von IN01 entsprechend mit

```
SendMQTT eviateg/GerätB/out/1 0
```

Das Prinzip lässt natürlich auf bis zu sechs Eingänge/Ausgänge beim IT251 erweitern.

7.2 Schalten von Lampen bei Sonnenuntergang und Sonnenaufgang

In diesem Beispiel kommt die Fähigkeit der eviateg-Geräte zur Anwendung, Ereignisse beim Sonnenauf- bzw. -untergang auszulösen.

In der Werkseinstellung werden die Geräte mit einer Tabelle ausgeliefert, deren Zeiten sich auf den Standort der eviateg GmbH beziehen.

Sie können mit CONNY sehr leicht eine Tabelle mit den Zeiten erzeugen, die am Standort des Gerätes gelten. Klicken Sie dazu im Navigationsbaum auf "**Zeitsteuerung**", um diese Eigenschaftenseite zu öffnen.

Im zweiten Rahmen von oben ("Sonnenauf- und -untergangszeiten") öffnet sich nach Klick auf die Schaltlfäche "**Tabelle erzeugen**" ein Dialog, in den Sie zu einer Stadt (und ggf. Straße) die Längen- und Breitengrade ermitteln können.

Zur Unterscheidung zwischen Schaltjahren und allen anderen können Sie auch das Jahr für diese Tabelle vorgeben.

Durch Klick auf die Schaltfläche "**Tabelle erzeugen**" öffnet sich der Windws-Dialog zum Speichern der Tabelle. Als Vorgabe wird der Dateiname aus dem Ortnamen und dem Jahr verwendet.

Durch Klick auf die Schaltfläche "Tabelle übertragen" kann die neu erzeugte Tabelle in das Gerät übertragen werden.

In die Felder für die Reaktionen bei Sonnenaufgang und Sonnenuntergang können nun die Befehle zum Ansteuern der Lampen eingetragen werden, z.B. beim Sonnenaufgang

```
SendMQTT q 2 r shellies/Shelly/Etage1/0/command off | SendMQTT q 2 r shellies/Shelly/Etage2/0/command off
```

und bei Sonnenuntergang

```
SendMQTT q 2 r shellies/Shelly/Etage1/0/command on | SendMQTT q 2 r shellies/Shelly/Etage2/0/command on
```

Der senkrechte Strich ' | ' (mit Leertasten davor und dahinter !) dient der Trennung der Befehle.

7.3 Anruf soll Relais einschalten

Im dritten Beispiel soll die Schranke zu einem Industriehof durch den Anruf berechtigter Teilnehmer geöffnet werden.

Dazu muss mit CONNY die Rufnummer des Teilnehmers im Gerät eingetragen werden. Bei nur **wenigen Teilnehmern** kann dies über die Eigenschaftenseite "**Anrufer**" erfolgen.

Der Befehl zum Öffnen der Schranke muss in diesem Fall auf der Eigenschaftenseite "Reaktionen / Anruf-Reaktionen" eingetragen werden. Das Reaktionsfeld ergibt sich aus dem Gerätetyp (Geräte mit ISDN-Anschluss wie das IT251GI bieten bis zu drei Rufnummern mit differenzierten Reaktionen).

Das Schalten kann z.B. mit

```
SendMQTT Hof/Schranke/Relais on
```

geschehen.

Bei vielen Teilnehmern empfiehlt es sich, deren Rufnummern auf der Eigenschaftenseite "**Adressbuch**" zu erfassen und die Reaktion unter "**Reaktionen / Anrufer-Reaktionen / Anrufer aus dem Adressbuch**" einzutragen (diese sogenannte "WhiteList-Caller-Reaktion wird aber nicht von allen Geräten unterstützt).

7.4 Steuerung durch Verknüpfungen

Selbstverständlich lassen sich auch bedingte SendMQTT-Befehle durch logische Verknüpfungen, Verknüpfungen von Eingangszuständen und Verknüpfungen von Flags einsetzen.

7.4.1 Verknüpfung von Eingangszuständen

Beispiel:

Wenn die Eingänge 0, 2 und 5 geschlossen sind, soll eine schaltbare Steckdose eingeschaltet und anderenfalls ausgeschaltet werden. Dazu stellt CONNY die Assistenten "**Logische Verknüpfungen von Ein-/Ausgängen**" und "**MQTT-Topic publizieren**" zur Verfügung:

```
IO 0 2 5 And |+ SendMQTT Delock/Switch1/cmnd/POWER on | IO 0 2 5 And |- SendMQTT Delock/Switch1/cmnd/POWER off
```

Hinweis: das Senden eines MQTT-Befehls setzt das interne Flag "Aktion war erfolgreich". Der Ausschalt-Befehl kann also nicht direkt mit " `|-` " eingeleitet werden, sondern

erfordert ein erneutes Testen der Eingänge.

Alternativ kann auch ein Textspeicherplatz für den on-/off-Parameter verwendet werden, wobei der gemerkte Inhalt dann dynamisch per Text-Makro eingesetzt wird:

```
IO 0 2 5 And |+ Text 9 on |- Text 9 off | SendMQTT
Delock/Switch1/cmnd/POWER &T9
```

Da das Speichern des Textes ("Text 9...") das interne Flag "Aktion war erfolgreich" nicht verändert, kann der SendMQTT-Befehl ohne Bedingung mit " `|` " eingeleitet werden.

7.4.2 Verknüpfungen von Flags

Ebenso können mehrere der 32 Flags UND- bzw. ODER-verknüpft werden.

Auch hierbei wird mit einer Test-Anweisung das interne Flag für die bedingte Ausführung einer Aktion verwendet.

Dabei hilft der **Assistent "Flags setzen, zurücksetzen oder prüfen"** bei der Eingabe der Test-Anweisung und der Assistent "**MQTT-Topic publizieren**" bei der Eingabe des SendMQTT-Befehls:

Beispiel:

Wenn eines der Flags 2, 18 oder 28 gesetzt ist, soll eine Pumpe eingeschaltet werden:

```
Flag 2 18 28 Or |+ SendMQTT Pumpwerk3/Pumpe1 on
```

Die Flag-OR-Anweisung setzt in diesem Fall das Flag "Aktion war erfolgreich", das dann mit " `|+` " ausgewertet wird.

7.4.3 Weitere logische Verknüpfungen

Der Assistent "**Flags setzen, zurücksetzen oder prüfen**" bietet noch eine Besonderheit: das Prüfen, ob eine neue Stunde, ein neuer Tag, eine neue Woche oder ein neuer Monat angefangen hat.

Weitere Abfragen dieser Bedingung würden bis zum Auftreten einer neuen Stunde etc. das interne Flag für die bedingte Ausführung auf "Vorige Aktion nicht erfolgreich" setzen.

Beispiel:

An jedem ersten Montag im Monat soll um 8 Uhr die Funktionsfähigkeit einer Anlage geprüft werden.

Auf der Eigenschaftenseite "Zeitsteuerung" muss dazu in der Zeile "Montag" in der Spalte "8 Uhr" ein Häkchen gesetzt werden, damit die Reaktion für "Zeitsteuerung aktiv" ausgeführt wird.

Mit den Assistenten für "**Flags setzen, zurücksetzen oder prüfen**" sowie "**MQTT-Topic publizieren**" im Feld für das Aktivieren der Zeitsteuerung kann dann z.B. eingesetzt werden:

```
Flag Month Test |+ SendMQTT QoS 1 Delock/Switch1/cmnd/POWER on
```

7.5 Alarmierungen und Steuerbefehle durch eine SPS

Befehle, z.B. zum Alarmieren, können auf das vom Gerät abonnierte Befehls-Topic `cmd` (siehe Abschnitt 5.3.1, Seite 13) publiziert werden.

Die Rückmeldungen des Gerätes werden über ein entsprechendes Topic (Werkseinstellung `Ergebnis`) publiziert.

Damit die Ablaufsteuerung in der SPS zwischen einem vorangegangenem Ergebnis und dem aktuellen Ergebnis unterscheiden kann, sollte das Ergebnis-Topic vor dem Steuerbefehl **gelöscht** oder **eindeutig vorbesetzt** werden (durch Publizieren einer leeren Paylod oder eines "?" an das Ergebnis-Topic) !

8 Einrichten eines eigenen MQTT-Brokers

An zwei Beispielen soll aufgezeigt werden, wie einfach das Einrichten eines eigenen MQTT-Brokers ist:

8.1 Einrichten eines MQTT-Brokers auf einem Windows-Rechner

Wer einen Windows-Rechner zur Verfügung hat, der rund um die Uhr läuft, kann darauf eine Broker-Software installieren.

Eine der bekanntesten MQTT-Broker ist das Open-Source-Projekt **Mosquitto** der unter https://mosquitto.org/download/ als 64-Bit- Version und 32-Bit-Version heruntergeladen werden kann.

Beachten Sie bitte in jedem Fall die rechtlichen Aspekte wie Lizenzbedingungen etc.

8.1.1 Installation

Die heruntergeladene Datei ist allerdings nicht elektronisch signiert, so dass u.U. eine entsprechende Warnung vom Windows Defender angezeigt wird.

Bei den zu installierenden Komponenten sollte auf jeden Fall "Service" markiert sein.

Die Installation erfolgt typischerweise unter "**C:\Program Files\mosquitto**" und dauert nur wenige Minuten.

8.1.2 Einrichten von Benutzern und Passwörtern (empfohlen)

Um den Zugang zum Broker mit einem oder mehreren Benutzernamen und Passwort / Passwörtern zu schützen, legen Sie mit einem einfachen Text-Editor (z.B. NotePad) eine **leere Textdatei** in einem Unterverzeichnis an, das nicht durch die Benutzerkontensteuerung beschränkt ist, z.B. **c:\temp\mqtt\Zugangsdaten.txt** .

Starten Sie ein Befehlsfenster, wechseln in das Installationsverzeichnis (s.o.) und rufen Sie das Programm **mosquitto_passwd** wie folgt einmal oder mehrmals auf:

```
mosquitto_passwd.exe -b "c:\temp\mqtt\Zugangsdaten.txt"
Benutzername Passwort
```

Die Benutzernamen sind in der Datei im Klartext und das Passwort verschlüsselt abgelegt.

Für den Fall, dass das Passwort nicht verschlüsselt ist, geben Sie

```
mosquitto_passwd.exe -U "c:\temp\mqtt\Zugangsdaten.txt"
```

ein.

Hinweise: Um einen Benutzer auszutragen, geben Sie den Befehl

```
mosquitto_passwd.exe -D "c:\temp\mqtt\Zugangsdaten.txt"
Benutzername
```

ein.

Anschließend muss in der Datei **C:\Program Files\mosquitto\mosquitto.conf** der Eintrag `allow_anonymous false` aktiviert werden (Kommentarzeichen '`#`' entfernen und `true` in false ändern).

Außerdem muss der Eintrag `password_file` um den Verweis auf die Passwort-Datei `"c:\temp\mqtt\Zugangsdaten.txt"` erweitert und das Kommentarzeichen '`#`' entfernt werden.

8.1.3 Starten des Brokers als Dienst

Nach der Installation ist der Broker noch nicht als Dienst gestartet.

Dies kann am Einfachsten über die Eingabe des Befehls `net start mosquitto` in einem Befehlsfenster erfolgen.

Umgekehrt kann der Dienstm mit `net stop mosquitto` wieder beendet werden.

8.1.4 Betrieb hinter einem Router

Wenn der Broker vom Internet aus erreichbar sein soll, empfiehlt es sich aus Sicherheitsgründen, den üblicherweise benutzten Port 1883 im Router auf einen "unüblichen" Port, z.B. 58317, zu ändern und in der NAT-Tabelle (Network Address Translation) eine Weiterleitung auf den Broker-PC und dessen Port 1883 einzurichten.

8.1.5 Testen des Brokers

Ob die Einrichtung des Brokers erfolgreich war, kann z.B. mit dem Tool "**MQTT-Explorer**" (siehe Abschnitt 9, Seite 25, getestet werden.

8.2 Einrichten eines MQTT-Brokers auf einem Raspberry Pi

Ein MQTT-Broker erfordert keinen dauernd durchlaufenden PC, sondern kann sogar auf einem Raspberry Pi installiert werden.

Im Elektronikhandel gibt es fertig zusammengestellte **Bundle**, bestehend aus einem Raspberry Pi im Hutschienengehäuse, einem Netzteil, Stromversorgungskabel, Kühlkörper-Satz und Micro-SD-Karte mit Betriebssystem für **unter 70 € brutto**.

Der **Raspberry Pi 3 B+** bietet neben einem LAN-Port auch einen WLAN-Port, so dass z.B. ein Betrieb mit WLAN-gesteuerten Steckdosen und Lampen in einem gemeinsamen WLAN ermöglicht wird.

Nach der Installation des Betriebssystems auf dem Raspberry Pi (die fast selbsterklärend ist, für die es aber auch genügend Tutorials im Internet gibt) muss der Broker installiert werden.

Nachfolgend wird dies wieder am Beispiel des MQTT-Brokers Mosquitto beschrieben.

8.2.1 Installation

Der Befehl

```
sudo apt-get install -y mosquitto mosquitto-clients
```

installiert und startet den Broker.

8.2.2 Einrichten von Benutzern und Passwörtern (empfohlen)

Da der Broker im Rahmen der Installation automatisch gestartet wurde, muss er zur

Übernahme der Zugangsdaten vorerst mit

```
sudo systemctl stop mosquitto.service
```

gestoppt werden.

Mit dem Befehl

```
sudo mosquitto_passwd -c /etc/mosquitto/passwd <Benutzername>
```

kann eine Passwortdatei mit dem angegeben Benutzernamen angelegt werden. Das zugehörige Passwort wird anschließend abgefragt und verifiziert.

Bearbeiten Sie anschließend die Konfiguationsdatei mit dem Befehl

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

und fügen Sie die beiden Einträge

```
password_file /etc/mosquitto/passwd
```

und

```
allow_anonymous false
```

hinzu.

Danach muss der Broker mit dem Befehl

```
sudo systemctl start mosquitto.service
```

wieder gestartet werden.

8.2.3 Betrieb hinter einem Router

Wenn der Broker vom Internet aus erreichbar sein soll, empfiehlt es sich aus Sicherheitsgründen, den üblicherweise benutzten Port 1883 im Router auf einen "unüblichen" Port, z.B. 58317, zu ändern und in der NAT-Tabelle (Network Address Translation) eine Weiterleitung auf den Raspberry Pi und dessen Port 1883 einzurichten.

8.2.4 Testen des Brokers

Ob die Einrichtung des Brokers erfolgreich war, kann z.B. mit dem Tool "**MQTT-Explorer**" (siehe Abschnitt 9, Seite 25, getestet werden.

9 Empfehlenswerte Zusatzprogramme

9.1 MQTT-Explorer

In vielen Fällen möchte man die Daten eines MQTT-Brokers ansehen können, die von einem oder mehreren Clients übertragen worden sind.

Eines der flexibelsten Programme dafür ist der **MQTT-Explorer**, der unter http://mqtt-explorer.com/ kostenlos heruntergeladen werden kann.

Beachten Sie bitte in jedem Fall die rechtlichen Aspekte wie Lizenzbedingungen etc.

9.1.1 Eintragen der Zugangsdaten zum Broker

Fügen Sie einen neuen Broker-Eintrag durch einen Klick auf die gelbe Schaltfläche "+" in der linken oberen Ecke hinzu.

Geben Sie unter "**Name**" eine Bezeichnung des Brokers ein.

Unter "**Host**" geben Sie die IP-Adresse des Brokers oder eine URL ein.

Wenn Broker und MQTT-Explorer auf dem selben Rechner betrieben werden, tragen Sie hier "127.0.0.1" ein.

Tragen Sie unter "**Username**" und "**Password**" die vorher zugewiesenen Daten ein.

9.1.2 Abonnieren von Topics

Als Voreinstellung (die mit der Schaltfläche "**Advanced**" geändert werden kann) abonniert der MQTT-Explorer alle Topics (mit der Angabe "`#`") sowie alle Statistikdaten des Brokers (mit "`$SYS/#`").

Klicken Sie zum Verbinden mit dem Broker auf die Schaltfläche "**Connect**".

Die von den Clients publizierten werden in einer Baumdarstellung dargestellt, die durch einen Klick auf die Dreiecksflächen erweitert oder eingeklappt werden kann.

Eine interessante Eigenschaft vom MQTT-Explorer ist, dass Topics eines Clients wieder gelöscht werden können, die z.B. als Überbleibsel nach dem Umbenennen von Topics auftreten können. Markieren Sie dazu den entsprechenden Zweig der Baumdarstellung und klicken Sie anschließend auf das Mülleimer-Symbol.

Eine weitere sehr interessante Eigenschaft vom MQTT-Explorer ist die Darstellung des zeitlichen Verlaufes von numerischen Werten in einer Grafik, sofern die Daten ohne Einheiten publiziert worden sind.

Erweitern Sie bei solchen Topics auf der rechten Seite die Überschrift "History".

Um den Names eines Topics in die Zwischenablage zu kopieren, genügt ein Klick auf das Symbol neben "Topic" am oberen Rand.

9.1.3 Publizieren von Topics

In der rechten unteren Hälfte besteht darüber hinaus die Möglichkeit, MQTT-Topics mit einer Payload unter Angabe der QoS und des Retain-Flags zu publizieren.

10 Beispiele für MQTT-Topics bei eviateg-Geräten

Die vom Gerät abonnierten und publizierten Topics können mit CONNY auf der Eigenschaftseite "**MQTT**" mit der Schaltfläche "**Topic-Namen speichern**" exportiert und editiert werden.

Die nachfolgenden Beispiele entsprechen der Werkseinstellung; Ebenenbezeichnungen und Namen wurden nicht geändert.

10.1 IT180 (mit zwei MBus-Slaves)

```
Vom IT180 publizierte Topics:
eviateg/IT180-00209F0F/Eingänge/In00
eviateg/IT180-00209F0F/Eingänge/In01
eviateg/IT180-00209F0F/Eingänge/In02
eviateg/IT180-00209F0F/Eingänge/In03
eviateg/IT180-00209F0F/Eingänge/In04
eviateg/IT180-00209F0F/Eingänge/In05
eviateg/IT180-00209F0F/Ausgänge/Out00
eviateg/IT180-00209F0F/Spannungswächter/Pwr00
eviateg/IT180-00209F0F/Spannungswächter/Pwr01
eviateg/IT180-00209F0F/Temperaturwächter/TMPGrd00
eviateg/IT180-00209F0F/Temperaturwächter/TMPGrd01
eviateg/IT180-00209F0F/Temperaturwächter/TMPGrd02
eviateg/IT180-00209F0F/Analogwertwächter/ADCGrd00
eviateg/IT180-00209F0F/Analogwertwächter/ADCGrd01
eviateg/IT180-00209F0F/Analogwertwächter/ADCGrd02
eviateg/IT180-00209F0F/Analogwertwächter/ADCGrd03
eviateg/IT180-00209F0F/Analogwertwächter/ADCGrd04
eviateg/IT180-00209F0F/Analogwertwächter/ADCGrd05
eviateg/IT180-00209F0F/Alarme/In00
eviateg/IT180-00209F0F/Alarme/In01
eviateg/IT180-00209F0F/Alarme/In02
eviateg/IT180-00209F0F/Alarme/In03
eviateg/IT180-00209F0F/Alarme/In04
eviateg/IT180-00209F0F/Alarme/In05
eviateg/IT180-00209F0F/Alarme/Out00
eviateg/IT180-00209F0F/Alarme/Pwr00
eviateg/IT180-00209F0F/Alarme/Pwr01
eviateg/IT180-00209F0F/Alarme/TMPGrd00
eviateg/IT180-00209F0F/Alarme/TMPGrd01
eviateg/IT180-00209F0F/Alarme/TMPGrd02
eviateg/IT180-00209F0F/Alarme/ADCGrd00
eviateg/IT180-00209F0F/Alarme/ADCGrd01
eviateg/IT180-00209F0F/Alarme/ADCGrd02
eviateg/IT180-00209F0F/Alarme/ADCGrd03
eviateg/IT180-00209F0F/Alarme/ADCGrd04
eviateg/IT180-00209F0F/Alarme/ADCGrd05
eviateg/IT180-00209F0F/Temperatursensoren/Temp0
eviateg/IT180-00209F0F/Temperatursensoren/Temp1
eviateg/IT180-00209F0F/Spannungswerte/Stromversorgung
eviateg/IT180-00209F0F/Spannungswerte/Akku
eviateg/IT180-00209F0F/Analogeingänge/InA
eviateg/IT180-00209F0F/Analogeingänge/InB
eviateg/IT180-00209F0F/GSM-Feldstärke
eviateg/IT180-00209F0F/MBus/SVM_105158/Datensatz_00
eviateg/IT180-00209F0F/MBus/SVM_105158/Datensatz_01
eviateg/IT180-00209F0F/MBus/SVM_105158/Datensatz_02
eviateg/IT180-00209F0F/MBus/SVM_105158/Datensatz_03
eviateg/IT180-00209F0F/MBus/SVM_105158/Datensatz_04
eviateg/IT180-00209F0F/MBus/SVM_105158/Datensatz_05
eviateg/IT180-00209F0F/MBus/SVM_105158/Datensatz_06
eviateg/IT180-00209F0F/MBus/SVM_105158/Datensatz_07
eviateg/IT180-00209F0F/MBus/SVM_105158/Datensatz_08
eviateg/IT180-00209F0F/MBus/SVM_105158/Datensatz_09
eviateg/IT180-00209F0F/MBus/SVM_105158/Datensatz_10
eviateg/IT180-00209F0F/MBus/SVM_105158/Datensatz_11
eviateg/IT180-00209F0F/MBus/SVM_105158/Datensatz_12
eviateg/IT180-00209F0F/MBus/SVM_105158/Datensatz_13
eviateg/IT180-00209F0F/MBus/FIN_23007990/Datensatz_14
```

```
eviateg/IT180-00209F0F/MBus/FIN_23007990/Datensatz_15
eviateg/IT180-00209F0F/MBus/FIN_23007990/Datensatz_16
eviateg/IT180-00209F0F/MBus/FIN_23007990/Datensatz_17
eviateg/IT180-00209F0F/MBus/FIN_23007990/Datensatz_18

Vom IT180 abonnierte Topics:
eviateg/IT180-00209F0F/cmd
eviateg/IT180-00209F0F/out/#
```

10.2 IT70

```
Vom IT70 publizierte Topics:
eviateg/IT70-0020B60F/Eingänge/In00
eviateg/IT70-0020B60F/Eingänge/In01
eviateg/IT70-0020B60F/Eingänge/In02
eviateg/IT70-0020B60F/Eingänge/In03
eviateg/IT70-0020B60F/Ausgänge/Out00
eviateg/IT70-0020B60F/Ausgänge/Out01
eviateg/IT70-0020B60F/Spannungswächter/Pwr00
eviateg/IT70-0020B60F/Spannungswächter/Pwr01
eviateg/IT70-0020B60F/Temperaturwächter/TMPGrd00
eviateg/IT70-0020B60F/Temperaturwächter/TMPGrd01
eviateg/IT70-0020B60F/Temperaturwächter/TMPGrd02
eviateg/IT70-0020B60F/Temperaturwächter/TMPGrd03
eviateg/IT70-0020B60F/Temperaturwächter/TMPGrd04
eviateg/IT70-0020B60F/Temperaturwächter/TMPGrd05
eviateg/IT70-0020B60F/Temperaturwächter/TMPGrd06
eviateg/IT70-0020B60F/Temperaturwächter/TMPGrd07
eviateg/IT70-0020B60F/Temperaturwächter/TMPGrd08
eviateg/IT70-0020B60F/Analogwertwächter/ADCGrd00
eviateg/IT70-0020B60F/Analogwertwächter/ADCGrd01
eviateg/IT70-0020B60F/Analogwertwächter/ADCGrd02
eviateg/IT70-0020B60F/Analogwertwächter/ADCGrd03
eviateg/IT70-0020B60F/Analogwertwächter/ADCGrd04
eviateg/IT70-0020B60F/Analogwertwächter/ADCGrd05
eviateg/IT70-0020B60F/Analogwertwächter/ADCGrd06
eviateg/IT70-0020B60F/Analogwertwächter/ADCGrd07
eviateg/IT70-0020B60F/Analogwertwächter/ADCGrd08
eviateg/IT70-0020B60F/Analogwertwächter/ADCGrd09
eviateg/IT70-0020B60F/Analogwertwächter/ADCGrd10
eviateg/IT70-0020B60F/Analogwertwächter/ADCGrd11
eviateg/IT70-0020B60F/Alarme/In00
eviateg/IT70-0020B60F/Alarme/In01
eviateg/IT70-0020B60F/Alarme/In02
eviateg/IT70-0020B60F/Alarme/In03
eviateg/IT70-0020B60F/Alarme/Out00
eviateg/IT70-0020B60F/Alarme/Out01
eviateg/IT70-0020B60F/Alarme/Pwr00
eviateg/IT70-0020B60F/Alarme/Pwr01
eviateg/IT70-0020B60F/Alarme/TMPGrd00
eviateg/IT70-0020B60F/Alarme/TMPGrd01
eviateg/IT70-0020B60F/Alarme/TMPGrd02
eviateg/IT70-0020B60F/Alarme/TMPGrd03
eviateg/IT70-0020B60F/Alarme/TMPGrd04
eviateg/IT70-0020B60F/Alarme/TMPGrd05
eviateg/IT70-0020B60F/Alarme/TMPGrd06
eviateg/IT70-0020B60F/Alarme/TMPGrd07
eviateg/IT70-0020B60F/Alarme/TMPGrd08
eviateg/IT70-0020B60F/Alarme/ADCGrd00
eviateg/IT70-0020B60F/Alarme/ADCGrd01
eviateg/IT70-0020B60F/Alarme/ADCGrd02
eviateg/IT70-0020B60F/Alarme/ADCGrd03
eviateg/IT70-0020B60F/Alarme/ADCGrd04
eviateg/IT70-0020B60F/Alarme/ADCGrd05
eviateg/IT70-0020B60F/Alarme/ADCGrd06
eviateg/IT70-0020B60F/Alarme/ADCGrd07
eviateg/IT70-0020B60F/Alarme/ADCGrd08
eviateg/IT70-0020B60F/Alarme/ADCGrd09
eviateg/IT70-0020B60F/Alarme/ADCGrd10
eviateg/IT70-0020B60F/Alarme/ADCGrd11
eviateg/IT70-0020B60F/Temperatursensoren/Temp0
eviateg/IT70-0020B60F/Temperatursensoren/Temp1
eviateg/IT70-0020B60F/Temperatursensoren/Temp2
eviateg/IT70-0020B60F/Spannungswerte/Stromversorgung
eviateg/IT70-0020B60F/Spannungswerte/Akku
eviateg/IT70-0020B60F/Spannungswerte/Ladestrom
```

```
eviateg/IT70-0020B60F/Analogeingänge/InA
eviateg/IT70-0020B60F/Analogeingänge/InB
eviateg/IT70-0020B60F/Analogeingänge/InC
eviateg/IT70-0020B60F/Analogeingänge/InD
eviateg/IT70-0020B60F/GSM-Feldstärke

Vom IT70 abonnierte Topics:
eviateg/IT70-0020B60F/cmd
eviateg/IT70-0020B60F/out/
```

10.3 IT251

```
Vom IT251GI publizierte Topics:
eviateg/IT251GI-00208153/Eingänge/In00
eviateg/IT251GI-00208153/Eingänge/In01
eviateg/IT251GI-00208153/Eingänge/In02
eviateg/IT251GI-00208153/Eingänge/In03
eviateg/IT251GI-00208153/Eingänge/In04
eviateg/IT251GI-00208153/Eingänge/In05
eviateg/IT251GI-00208153/Eingänge/In06
eviateg/IT251GI-00208153/Eingänge/In07
eviateg/IT251GI-00208153/Eingänge/In08
eviateg/IT251GI-00208153/Eingänge/In09
eviateg/IT251GI-00208153/Eingänge/In10
eviateg/IT251GI-00208153/Eingänge/In11
eviateg/IT251GI-00208153/Eingänge/In12
eviateg/IT251GI-00208153/Eingänge/In13
eviateg/IT251GI-00208153/Eingänge/In14
eviateg/IT251GI-00208153/Eingänge/In15
eviateg/IT251GI-00208153/Eingänge/In16
eviateg/IT251GI-00208153/Eingänge/In17
eviateg/IT251GI-00208153/Eingänge/In18
eviateg/IT251GI-00208153/Eingänge/In19
eviateg/IT251GI-00208153/Ausgänge/Out00
eviateg/IT251GI-00208153/Ausgänge/Out01
eviateg/IT251GI-00208153/Ausgänge/Out02
eviateg/IT251GI-00208153/Ausgänge/Out03
eviateg/IT251GI-00208153/Ausgänge/Out04
eviateg/IT251GI-00208153/Ausgänge/Out05
eviateg/IT251GI-00208153/Spannungswächter/Pwr00
eviateg/IT251GI-00208153/Spannungswächter/Pwr01
eviateg/IT251GI-00208153/Spannungswächter/Pwr02
eviateg/IT251GI-00208153/Spannungswächter/Pwr03
eviateg/IT251GI-00208153/Spannungswächter/Pwr04
eviateg/IT251GI-00208153/Temperaturwächter/TMPGrd00
eviateg/IT251GI-00208153/Temperaturwächter/TMPGrd01
eviateg/IT251GI-00208153/Temperaturwächter/TMPGrd02
eviateg/IT251GI-00208153/Temperaturwächter/TMPGrd03
eviateg/IT251GI-00208153/Temperaturwächter/TMPGrd04
eviateg/IT251GI-00208153/Temperaturwächter/TMPGrd05
eviateg/IT251GI-00208153/Temperaturwächter/TMPGrd06
eviateg/IT251GI-00208153/Temperaturwächter/TMPGrd07
eviateg/IT251GI-00208153/Temperaturwächter/TMPGrd08
eviateg/IT251GI-00208153/Alarme/In00
eviateg/IT251GI-00208153/Alarme/In01
eviateg/IT251GI-00208153/Alarme/In02
eviateg/IT251GI-00208153/Alarme/In03
eviateg/IT251GI-00208153/Alarme/In04
eviateg/IT251GI-00208153/Alarme/In05
eviateg/IT251GI-00208153/Alarme/In06
eviateg/IT251GI-00208153/Alarme/In07
eviateg/IT251GI-00208153/Alarme/In08
eviateg/IT251GI-00208153/Alarme/In09
eviateg/IT251GI-00208153/Alarme/In10
eviateg/IT251GI-00208153/Alarme/In11
eviateg/IT251GI-00208153/Alarme/In12
eviateg/IT251GI-00208153/Alarme/In13
eviateg/IT251GI-00208153/Alarme/In14
eviateg/IT251GI-00208153/Alarme/In15
eviateg/IT251GI-00208153/Alarme/In16
eviateg/IT251GI-00208153/Alarme/In17
eviateg/IT251GI-00208153/Alarme/In18
eviateg/IT251GI-00208153/Alarme/In19
eviateg/IT251GI-00208153/Alarme/Out00
eviateg/IT251GI-00208153/Alarme/Out01
eviateg/IT251GI-00208153/Alarme/Out02
```

```
eviateg/IT251GI-00208153/Alarme/Out03
eviateg/IT251GI-00208153/Alarme/Out04
eviateg/IT251GI-00208153/Alarme/Out05
eviateg/IT251GI-00208153/Alarme/Pwr00
eviateg/IT251GI-00208153/Alarme/Pwr01
eviateg/IT251GI-00208153/Alarme/Pwr02
eviateg/IT251GI-00208153/Alarme/Pwr03
eviateg/IT251GI-00208153/Alarme/Pwr04
eviateg/IT251GI-00208153/Alarme/TMPGrd00
eviateg/IT251GI-00208153/Alarme/TMPGrd01
eviateg/IT251GI-00208153/Alarme/TMPGrd02
eviateg/IT251GI-00208153/Alarme/TMPGrd03
eviateg/IT251GI-00208153/Alarme/TMPGrd04
eviateg/IT251GI-00208153/Alarme/TMPGrd05
eviateg/IT251GI-00208153/Alarme/TMPGrd06
eviateg/IT251GI-00208153/Alarme/TMPGrd07
eviateg/IT251GI-00208153/Alarme/TMPGrd08
eviateg/IT251GI-00208153/Temperatursensoren/Temp0
eviateg/IT251GI-00208153/Temperatursensoren/Temp1
eviateg/IT251GI-00208153/Temperatursensoren/Temp2
eviateg/IT251GI-00208153/Spannungswerte/Stromversorgung
eviateg/IT251GI-00208153/Spannungswerte/Akku
eviateg/IT251GI-00208153/Spannungswerte/Ladestrom
eviateg/IT251GI-00208153/GSM-Feldstärke

Vom IT251GI abonnierte Topics:
eviateg/IT251GI-00208153/cmd
eviateg/IT251GI-00208153/out/#
```

10.4 IT35G

```
Vom IT35G publizierte Topics:
eviateg/IT35G-00208C2B/Eingänge/In00
eviateg/IT35G-00208C2B/Eingänge/In01
eviateg/IT35G-00208C2B/Ausgänge/Out00
eviateg/IT35G-00208C2B/Alarme/In00
eviateg/IT35G-00208C2B/Alarme/In01
eviateg/IT35G-00208C2B/Alarme/Out00
eviateg/IT35G-00208C2B/GSM-Feldstärke

Vom IT35G abonnierte Topics:
eviateg/IT35G-00208C2B/cmd
eviateg/IT35G-00208C2B/out/#
```

11 Historie

Version	Datum	Änderungen
1.0	18.09.2020	Erste Version
1.1	27.10.2020	Erweiterungen für MBus
1.2	28.10.2020	Beispiele für MQTT-Topics erweitert
1.3	01.02.2021	Hinweise auf Client-Apps und ZigBee-Bridges
1.4	10.09.2021	Hinweise zur Befehlssteuerung durch eine SPS
1.5	20.07.2022	Erwähnung des Stichwortverzeichnisses im Inhaltsvereichnis; Hinweise auf Verschlüsselung mit TLS herausgestellt
1.6	21.08.2024	Anpassung Firmenanschrift
1.7	16.01.2026	Neuer Abschnitt 5.4.5 für nicht benötigte Topic-Ebenen

Stichwortverzeichnis

abonnieren....**6, 9f., 14f.**
Abonnieren....**6, 10, 12**
Adressbuch....**19**
Alarme....**10, 12, 14f.**
Android....7
Anruf....**12, 17, 19**
Assistent....**18ff.**
Ausgang....**12, 14**
Befehls-Topic....20
Benutzername....**8, 22, 24**
Broker....**5ff., 12, 17, 22ff.**
Client....**5ff., 17, 25**
CONNY....**12ff., 26**
Dienstqualität....**9**
EasyMQTT....7
Ebene....**5, 9f., 12ff.**
Ebene "MBus"....**13**
Eingang....**12, 18f.**
Ergebnis-Topic....21
Flags....**19f., 25**
Gerätename....**12f.**
GPRS....**7**
iOS....7
IT180....**26**
IT251....**28**
IT35G....**29**
IT70....**27**
Joker....**6**
JSON....**5**
Keep Alive....**10**
LAN....**7, 23**
Last Will....**9, 12**
MBus....**13, 15**
MBus-Slaves....26

Messwerte....**12**
Mosquitto....**22f.**
MQTT-Explorer....**23ff.**
Passwort....**8, 22ff.**
Payload....**5, 14f., 17, 25**
Präfix....**12f.**
Programmgesteuertes Publizieren....**12, 17**
Publish....**5f.**
QoS....**9, 15, 17, 20, 25**
Quality of Service....**9**
QuickSetup....**12**
Raspberry Pi....**7, 23f.**
Retain....**9, 15, 17, 25**
Router....**23f.**
SendMQTT....**17ff.**
Siri....7
SPS....20
Steuerbefehle....**13**
Subscribe....**6**
TCP/IP....**7**
Temperaturwächter....**12f.**
Testament....**9**
TLS....8, 12
Topic....**5f., 9f., 12ff., 19f., 25f.**
Topic-Namen....**9**
uhrzeitgesteuert....**17**
Unsubscribe....**6**
UTF8....**9**
Verbindungsüberwachung....**10**
Verknüpfung....**19f.**
Werkseinstellung....12f., 18, 21, 26
wildcard....**10**
Wildcard....**6, 10**
WLAN....**7, 23**
Zähler....**14f.**
Zeitsteuerung....**18, 20**
ZigBee....7

ZigBee-Bridge.....7
zigbee2mqtt.....7
Zustandsmeldungen.....**12**