



HowTo:

Das MQTT-Plugin an der Homematic IP Home Control Unit



homematic 

Inhaltsverzeichnis

1. Das MQTT-Plugin im Überblick	3
1.1 Was ist MQTT?	3
1.2 Ziel und Nutzen.....	3
1.3 Unterstützte Funktionen	3
1.4 Aktueller Funktionsumfang	4
1.5 Unterstützte Geräteklassen.....	5
2. Inbetriebnahme.....	6
2.1 Voraussetzungen	6
2.2 Aufbau der Plugin-Konfiguration.....	6
2.3 Broker anlegen	7
2.4 Broker konfigurieren	7
2.5 Gerät anlegen und konfigurieren.....	9
2.6 Feature-Felder konfigurieren	9
2.7 Payload-Verarbeitung	11
2.8 Steuerung über Control-Topics.....	13
2.9 Konfigurationsbeispiele.....	15

1. Das MQTT-Plugin im Überblick

Mit dem MQTT-Plugin können Sie kompatible Geräte und Dienste über einen MQTT-Broker in Ihr Homematic IP System einbinden. Zustände, Messwerte und Steuerfunktionen stehen anschließend für Visualisierungen, Automatisierungen, Gruppen und unterstützte Energiemanagementfunktionen zur Verfügung.

1.1 Was ist MQTT?

MQTT (Message Queuing Telemetry Transport) ist ein etabliertes Kommunikationsprotokoll für die Vernetzung von Geräten und Systemen. Es wird insbesondere in Smart Home, Gebäudeautomation, Energiemanagement und Internet of Things (IoT) eingesetzt.

Geräte und Dienste unterschiedlicher Hersteller tauschen Informationen über einen zentralen MQTT-Broker aus. Nachrichten werden auf sogenannten Topics veröffentlicht und von interessierten Teilnehmern abonniert.

1.2 Ziel und Nutzen

Die Integration richtet sich insbesondere an energie- und automationsrelevante Geräte, die bisher nicht direkt von Homematic IP unterstützt werden. Dazu zählen beispielsweise Wechselrichter, Batteriespeicher, Wallboxen und Wärmepumpen. Herstellerübergreifend angebundene Geräte stehen anschließend wie andere kompatible Geräte in der Homematic IP App sowie für Gruppen, Automatisierungen und weitere Systemfunktionen zur Verfügung.

1.3 Unterstützte Funktionen

- Verbindungen zu MQTT-Brokern über TCP oder WebSocket
- Optional verschlüsselte Verbindung über TLS
- Authentifizierung per Benutzername und Passwort oder Client-Zertifikat
- Empfang von Status- und Messwerten über frei konfigurierbare Topics
- Versand von Sollwerten und Steuerbefehlen auf separate Control-Topics
- Verarbeitung von Text-, Zahlen-, Boolean- und JSON-Payloads
- Extraktion, Mapping, Skalierung und Offset für eingehende Werte
- Zuordnung von MQTT-Datenpunkten zu Homematic IP Gerätetypen und Features
- Nutzung integrierter Geräte in App, Gruppen, Automatisierungen und Systemfunktionen

Beispiel: MQTT-fähige Leuchte

Eine über MQTT angebundene Leuchte kann in der Homematic IP App geschaltet, gemeinsam mit Homematic IP Tastern bedient und in Automatisierungen oder Gruppen eingebunden werden.

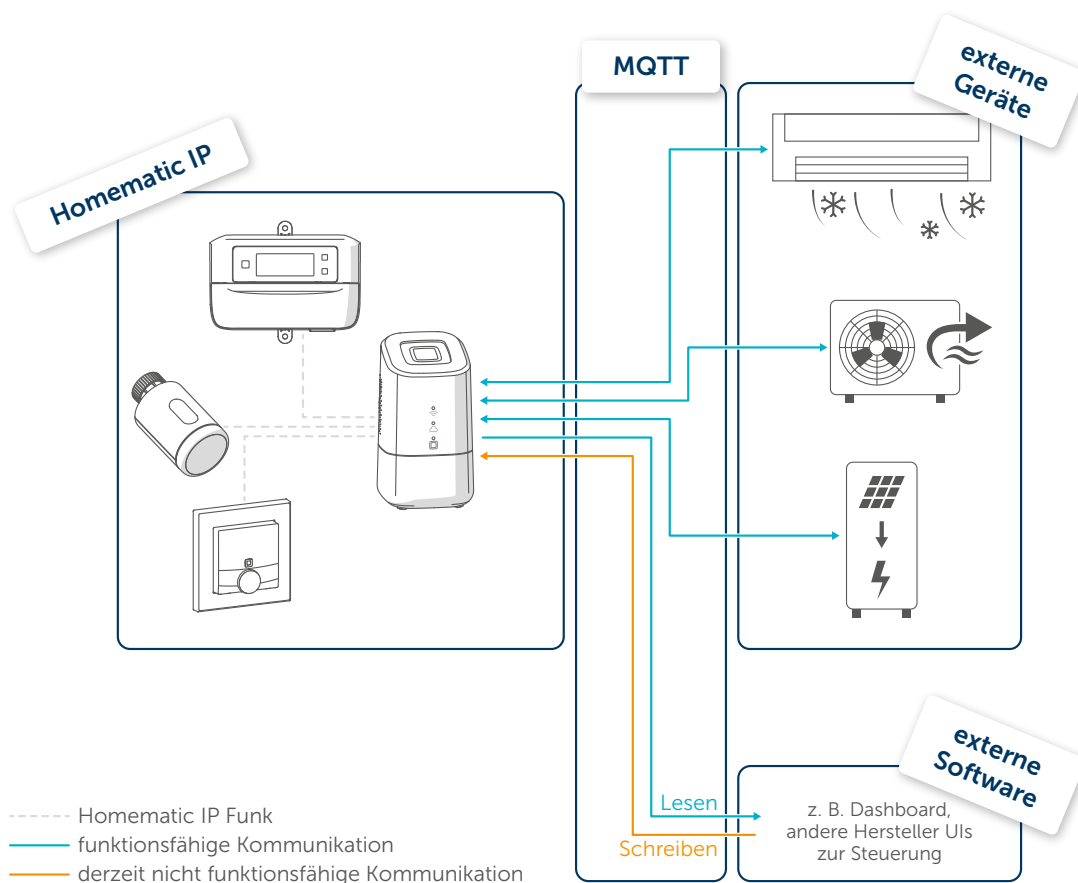
1.4 Aktueller Funktionsumfang

Die aktuelle Version ermöglicht die lesende und schreibende Kommunikation mit externen Geräten über MQTT. Das Schreiben erfolgt über ein zuvor registriertes Topic. Externe Geräte können so mit Homematic IP kommunizieren.

Beispiel: Eine externe Wärmepumpe übermittelt ihren Status über MQTT an die Home Control Unit. Anschließend kann die Wärmepumpe mit Homematic IP gesteuert werden.

Abgrenzung des Funktionsumfangs

Beispiel: Der Homematic IP Heizkörperthermostat kann über MQTT nicht von einer externen Software direkt gesteuert werden. Über die externe Software können jedoch alle verfügbaren Statusinformationen aus dem Homematic IP System ausgelesen werden.



Unterstützt	Derzeit nicht Bestandteil
Anlegen und Verwalten externer MQTT-Geräte	Veröffentlichen von Mess- und Sensordaten bestehender Homematic IP Geräte an externe MQTT-Systeme
Empfang und Verarbeitung von Statuswerten	Automatische Erkennung beliebiger Topic-Strukturen
Steuerung bidirektionaler Features	Beliebige Rücktransformationen, z. B. true/false zu ON/OFF, außerhalb des Control-Patterns
TCP, WebSocket, TLS und mTLS	

Daten wie Temperatur-, Luftfeuchtigkeits- oder Wetterinformationen vorhandener Homematic IP Geräte werden derzeit nicht über MQTT an externe Systeme veröffentlicht.

1.5 Unterstützte Geräteklassen

Bereich	Geräteklassen
Schalten & Licht	Schalter, Leuchten, Schalteingänge, Kontaktsensoren
Beschattung	Rollläden und Jalousien
Heizen, Klima & Lüften	Thermostate, HVAC-Systeme, Wärmepumpen, Klimasensoren
Sicherheit	Rauchwarnmelder, Wassersensoren, Kontakt-, Präsenz- und Belegungssensoren
Energie & Leistung	Energiezähler, Netzanschlusspunkte, Batteriespeicher, Wallboxen, Wechselrichter
Umwelt	Feinstaubsensoren
Mobilität	Fahrzeuge

2. Inbetriebnahme

2.1 Voraussetzungen

Für die Einrichtung benötigen Sie:

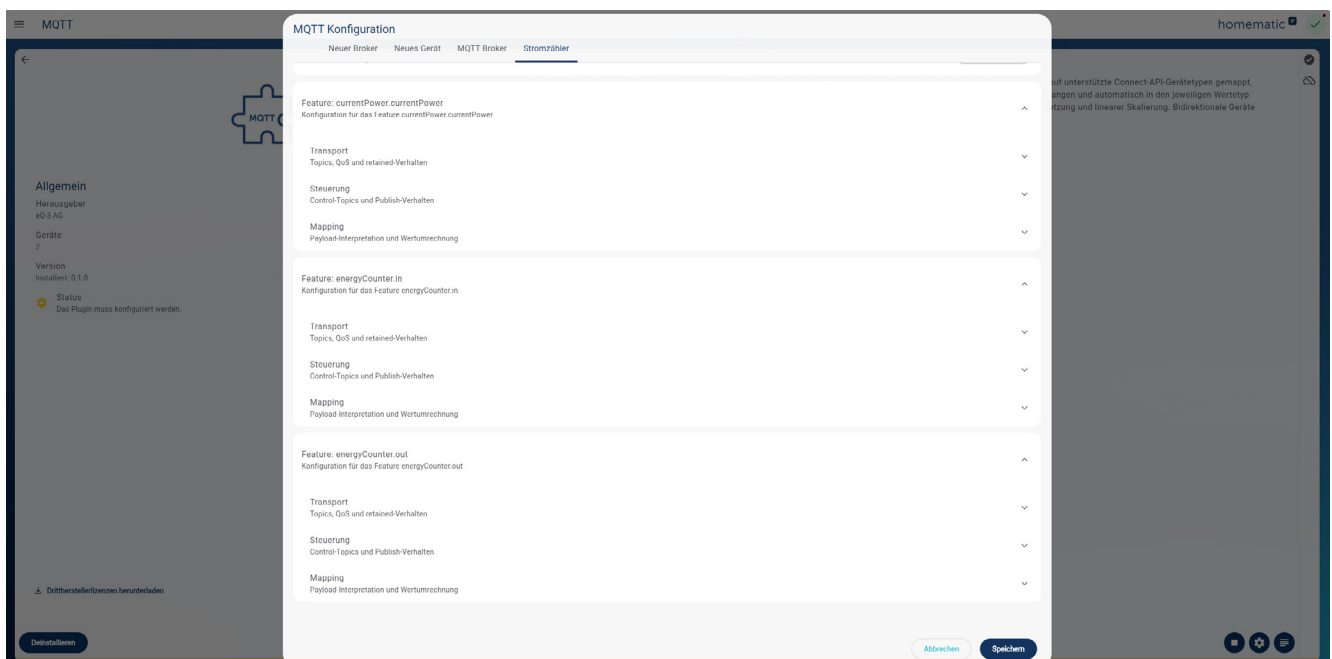
- eine Homematic IP Home Control Unit mit Zugriff auf die HCUweb
- ein installiertes MQTT-Plugin
- einen erreichbaren MQTT-Broker
- eine Netzwerkverbindung zwischen Home Control Unit und Broker
- die Verbindungsdaten des Brokers: Host, Port, Protokoll und gegebenenfalls Zugangsdaten oder Zertifikate
- die Topic- und Payload-Dokumentation der einzubindenden Geräte

2.2 Aufbau der Plugin-Konfiguration

Die Konfiguration des MQTT-Plugins ist hierarchisch in Gruppen, Untergruppen und Bereiche gegliedert:

- In der Gruppe „Neuer Broker“ legen Sie eine neue Broker-Verbindung an.
- In der Gruppe „Neues Gerät“ legen Sie ein neues MQTT-Gerät an.
- Bereits angelegte Broker und Geräte werden jeweils als eigene Gruppe angezeigt.
- Broker-Gruppen enthalten die Bereiche Verbindung, Sitzung, Sicherheit und Authentifizierung.
- Geräte-Gruppen enthalten den Bereich Allgemein sowie
- für jedes verfügbare Feature eine eigene Untergruppe.
- Die einzelnen Feature-Untergruppen sind in die Bereiche Transport, Steuerung und Mapping gegliedert.

Abhängig vom gewählten Gerätetyp und Feature werden nur die jeweils verfügbaren Einstellungen angezeigt.



2.3 Broker anlegen

- Öffnen Sie in der HCUweb die Konfiguration des MQTT-Plugins.



Zum Anlegen eines Brokers steht die feste Gruppe „Neuer Broker“ zur Verfügung.

- Öffnen Sie die Gruppe „Neuer Broker“.
- Tragen Sie die erforderlichen Angaben in den Bereichen Verbindung, Sitzung, Sicherheit und Authentifizierung ein.
- Klicken Sie im Bereich Verbindung auf „Broker hinzufügen“.
- Der Broker wird sofort angelegt und erscheint als eigene Gruppe.
- Speichern Sie spätere Änderungen an den Broker-Einstellungen regulär über „Speichern“.

Die Gruppe „Neuer Broker“ dient ausschließlich zum Anlegen einer neuen Broker-Verbindung. Änderungen an einem bereits angelegten Broker nehmen Sie anschließend in dessen eigener Gruppe vor.

Nur das Hinzufügen und Entfernen von Brokern und Geräten wirkt unmittelbar. Änderungen an Host, Sicherheit oder Feature-Feldern werden erst beim Speichern übernommen.

2.4 Broker konfigurieren

Feld	Typ	Standard	Beschreibung
Name	Text	–	Frei wählbarer Anzeigename
Host	Text	–	IP-Adresse oder Hostname des Brokers
Port	Zahl	1883 (TCP), 80 (WS)	Port des Brokers
Protokoll	Auswahl	TCP	TCP oder MQTT über WebSocket (WS)
TLS aktiviert	Ja/Nein	false	Verschlüsselte Verbindung über TLS/SSL
Clean Session	Ja/Nein	false	Verwirft bei true gespeicherte Subscriptions und wartende Nachrichten
WebSocket-Pfad	Text	/	Pfad des WebSocket-Endpunkts; nur bei WS
Authentifizierungstyp	Auswahl	none	none, username_password oder client_certificate

Authentifizierungstypen

Typ	Beschreibung
none	Keine Authentifizierung – Verbindung ohne Zugangsdaten.
username_password	Authentifizierung per Benutzername und Passwort.
client_certificate	Mutual TLS (mTLS) über Client-Zertifikat und privaten Schlüssel.

Beim Typ username_password müssen Sie den Authentifizierungstyp „username_password“ ausdrücklich auswählen. Nur eingetragene Zugangsdaten ohne passende Typ-Auswahl werden ignoriert.

Feld	Beschreibung
Benutzername	Benutzername für die Broker-Authentifizierung
Passwort	Passwort für die Broker-Authentifizierung

Client-Zertifikate

Mutual TLS (mTLS) – Authentifizierung über Client-Zertifikate. Alle Felder sind optional; es werden nur die gesetzten Werte an den TLS-Stack übergeben.

Feld	Dateitypen	Beschreibung
Client-Zertifikat	.crt, .pem	Zertifikat des Clients
Client-Key	.key, .pem	Privater Schlüssel des Clients
CA-Zertifikat	.crt, .pem	CA-Zertifikat zur Validierung des Brokers
Key-Passwort	–	Passwort für einen verschlüsselten Client-Key

HMIP System Events

In der Untergruppe System Events kann jeder Broker unabhängig Homematic IP Systemereignisse an weitere MQTT-Systeme weiterleiten. Die Weiterleitung kann als vollständige Event-Transaktion (Raw) oder als auf einzelne Events aufgeteilte Nachrichten (Split) erfolgen. Beide Modi können unabhängig voneinander oder gleichzeitig aktiviert werden. Ein leeres Topic deaktiviert den jeweiligen Modus.

Feld	Typ	Standard	Beschreibung
Raw Topic	Text	leer	Veröffentlicht die vollständige eventTransaction aus dem HMIP_SYSTEM_EVENT unverändert als JSON auf diesem festen Topic
Split Topic Prefix	Text	leer	Teilt die Transaktion in einzelne OutputEvent-Payloads und veröffentlicht sie unter <Prefix>/<EVENT_TYPE>/<Origin-ID>
QoS	0 / 1 / 2	0	Gemeinsame MQTT-QoS-Stufe für Raw- und Split-Nachrichten
Retained	Ja/Nein	false	Veröffentlicht Raw- und Split-Nachrichten als retained MQTT-Nachrichten

Beispiel für den Prefix `hmip/changed`:

Eine Geräteänderung mit der ID `3014F7110000000000000000` wird unter `hmip/changed/DEVICE/3014F7110000000000000000` veröffentlicht. Nachgestellte `/` im konfigurierten Prefix werden entfernt.

Der Split-Payload enthält `eventType`, die `originId` der Transaktion, `changedId` und das ursprüngliche einzelne Event. Es findet keine sendeseitige Filterung nach Eventtyp oder ID statt. Empfänger können über MQTT-Topic-Wildcards nur die benötigten Events abonnieren, beispielsweise `hmip/changed/DEVICE/#`.

2.5 Gerät anlegen und konfigurieren

- Wählen Sie in der Gruppe „Neues Gerät“ im Bereich „Allgemein“ den Gerätetyp und den zuvor angelegten Broker aus.
- Klicken Sie auf „Neues Gerät“.
- Das Gerät wird sofort angelegt und erscheint als eigene Gruppe.
- Tragen Sie einen geeigneten Gerätenamen ein.
- Konfigurieren Sie die benötigten Feature-Felder und speichern Sie die Änderungen.

Feld	Beschreibung
Name	Frei wählbarer Anzeigename des Geräts
Broker	Auswahl des zuvor angelegten MQTT-Brokers
Gerätetyp	Bestimmt, welche Feature-Felder angezeigt werden

Für jedes Feature des gewählten Gerätetyps wird eine eigene Untergruppe mit den Bereichen Transport, Steuerung und Mapping angezeigt. Ein Feature bleibt unberücksichtigt, wenn kein Status-Topic eingetragen ist.

Gerät entfernen

- Klicken Sie im Bereich „Allgemein“ auf „Gerät entfernen“, um das Gerät sofort vollständig zu löschen.

2.6 Feature-Felder konfigurieren

Jedes Feature-Feld, zum Beispiel `dimmerState.dimLevel` oder `switchState.on`, enthält Einstellungen für Status-Empfang, Steuerung und Payload-Verarbeitung.

Status-Empfang

Feld	Typ	Beschreibung
Topic	Text	MQTT-Topic für Statuswerte; leer lassen, wenn das Feature nicht verwendet wird.
QoS	0 / 1 / 2	0 = höchstens einmal, 1 = mindestens einmal, 2 = genau einmal.
Retained	Ja/Nein	Bei true wird die zuletzt gespeicherte Retained-Nachricht beim Verbinden verarbeitet. Bei false werden Retained-Nachrichten ignoriert.

Bei false werden Retained-Nachrichten ignoriert.

Steuerung

Die folgenden Felder erscheinen nur bei bidirektionalen Features, die gelesen und gesteuert werden können.

Feld	Typ	Standard	Beschreibung
Control-Topic	Text	–	Topic für ausgehende Steuerbefehle
Control-Pattern	Text	{{value}}	Vorlage für den gesendeten Payload
Publish-Retained	Ja/Nein	false	Sendet Steuerbefehle als Retained-Nachricht

Control-Topic und Status-Topic müssen unterschiedlich sein. Andernfalls kann das Plugin den eigenen Steuerbefehl wieder als Statuswert empfangen und eine Echo-Schleife verursachen.

Bei Publish-Retained speichert der Broker den letzten Steuerbefehl und sendet ihn neuen Subscribern unmittelbar nach dem Verbinden.

Payload-Einstellungen

Feld	Typ	Standard	Beschreibung
Payload-Typ	TEXT / NUMBER / BOOLEAN / JSON	NUMBER	Legt fest, wie der eingehende Payload grundsätzlich interpretiert wird.
JSON-Pointer	Text	–	Nur bei JSON: RFC-6901-Pointer auf den eigentlichen Wert, z. B. /device/power. Ohne Angabe wird das gesamte JSON-Dokument als Wert verwendet.
Input-Extractor	Regex	–	Optional erster Verarbeitungsschritt: extrahiert einen Teilstring aus dem Rohpayload (vor dem Text-Mapping). Nützlich um Präfixe, Suffixe oder Einheiten zu entfernen.
Text-Mapping	JSON-Objekt	{}	Ersetzt Textwerte nach dem Input-Extractor. Format: {"Quellwert": "Zielwert"}. Case-insensitiver Fallback vorhanden.
Number-Pattern	Regex	[-+]?[0-9]+(?:[.][0-9]+)?	Regex zur Zahlenextraktion aus dem (ggf. bereits vorverarbeiteten) Text. Erst das Ergebnis dieses Patterns wird als Zahl geparkt.
Skalierungsfaktor	Zahl	1.0	Multiplikator für den ermittelten Zahlenwert.
Offset	Zahl	0.0	Additivkonstante nach der Skalierung. Formel: Ergebnis = Wert × Skalierungsfaktor + Offset

2.7 Payload-Verarbeitung

Das Plugin verarbeitet eingehende Payloads nach einem mehrstufigen Pipeline-Prinzip. Die genaue Abfolge hängt vom konfigurierten Payload-Typ ab.

Allgemeine Verarbeitungspipeline

Rohpayload (String)

↓

[1] Input-Extractor (Regex) → Teilstring extrahieren, z. B. Zahl aus „Temp: 23.5 °C“

↓

[2] Text-Mapping → Textwerte ersetzen, z. B. „ein“ → „true“

↓

[3] Typ-spezifische Konvertierung → abhängig vom Payload-Typ (s. u.)

↓

[4] Skalierung & Offset → nur bei Zahlenwerten: $\text{Wert} \times \text{scaleFactor} + \text{offset}$



Beim Typ JSON wird vor Schritt 1 zunächst das JSON geparkt und ggf. der JSON-Pointer angewendet. Nur wenn der extrahierte JSON-Knoten ein Textwert ist, läuft er danach durch Schritt 1–4.

Payload-Typ NUMBER

Der Payload wird als Zahl interpretiert.

- Input-Extractor und Text-Mapping werden optional angewendet.
- Das Number-Pattern extrahiert die Zahl.
- Komma und Punkt werden als Dezimaltrennzeichen akzeptiert.
- Bei gemischter Schreibweise gilt das letzte Trennzeichen als Dezimaltrennzeichen.
- Zum Schluss werden Skalierungsfaktor und Offset angewendet.

Formel: Ergebnis = Wert × Skalierungsfaktor + Offset

Typischer Anwendungsfall: Sensor, der Werte wie 23.5 oder 23,5 oder Temp: 23.5 °C sendet.

Payload-Typ BOOLEAN

Der Payload wird als Wahrheitswert interpretiert.

1. Input-Extractor – optionaler Regex
2. Text-Mapping – optionale Textersetzung (z. B. gerätespezifische Texte auf true/false mappen)
3. Boolean-Parse – folgende Werte werden erkannt (case-insensitiv):

Wert → true	Wert → false
true, on, 1, yes	false, off, 0, no

Andere Werte führen zu einem Verarbeitungsfehler.

Payload-Typ TEXT

Nach Input-Extractor und Text-Mapping wird der Text abhängig vom Zieltyp in einen Boolean-, Zahlen- oder Enum-Wert konvertiert. Bei einem String-Ziel wird der Text unverändert weitergegeben.

1. Input-Extractor – optionaler Regex
2. Text-Mapping – optionale Textersetzung
3. Typ-Konvertierung – abhängig vom Zieltyp des Feature-Felds:
 - Zieltyp Boolean → Boolean-Parse (gleiche Regeln wie Typ BOOLEAN)
 - Zieltyp Zahl → Number-Pattern → Skalierung & Offset
 - Zieltyp Enum → Enum-Matching (case-insensitiv)
 - Sonstiges → Wert wird als String weitergegeben

Payload-Typ JSON

Der Payload wird als JSON-Dokument interpretiert.

1. JSON-Parse – der Rohpayload muss valides JSON sein
2. JSON-Pointer – optional; extrahiert einen bestimmten Knoten aus dem Dokument (RFC 6901, z. B. /data/temperature). Ohne Pointer wird das gesamte Dokument weiterverarbeitet.
3. Knotentyp-Auswertung:
 - JSON-Zahl → direkt Skalierung & Offset
 - JSON-Boolean → direkt als Wahrheitswert
 - JSON-String → weiter durch Input-Extractor → Text-Mapping → Typ-Konvertierung (wie bei TEXT)
 - JSON-Objekt/Array → nur möglich wenn Zieltyp String oder generisch ist

Text-Mapping im Detail

Das Text-Mapping ermöglicht die Übersetzung gerätespezifischer Textwerte in Standardwerte.

Format: JSON-Objekt als String, z. B.:

```
{„ein“: „true“, „aus“: „false“}
```

oder

```
{„OPEN“: „1“, „CLOSED“: „0“}
```

Verhalten:

1. Exakter Treffer (case-sensitiv) wird zuerst gesucht.
2. Kein exakter Treffer → case-insensitiver Vergleich als Fallback.
3. Kein Treffer → der ursprüngliche Wert wird unverändert weitergegeben (kein Fehler).

Ein leeres Objekt {} deaktiviert das Mapping effektiv.

2.8 Steuerung über Control-Topics

Bidirektionale Feature-Felder können nicht nur Statuswerte empfangen, sondern auch Steuerbefehle senden. Ein Steuerbefehl wird ausgelöst, wenn das übergeordnete System einen neuen Sollwert für das Gerät setzt.

Control-Topic

Das Control-Topic ist das MQTT-Topic, auf dem der Steuerbefehl veröffentlicht wird.



Control-Topic und Status-Topic müssen unterschiedlich sein. Wenn beide identisch sind, empfängt das Plugin seinen eigenen Steuerbefehl als Statuswert (Echo-Loop).

Wertaufbereitung vor dem Control-Pattern

1. Typ-spezifische Rückrechnung des HCU-Wertes.
 - Zahl → $(\text{Wert} - \text{Offset}) \div \text{Skalierungsfaktor}$
 - Boolean → true / false
 - Text → unverändert
 - Enum → Enum-Name
 - JSON → String-Darstellung
2. Umkehr-Text-Mapping
 - Sucht im Text-Mapping den Eintrag, dessen Zielwert dem zuvor berechneten Wert entspricht.
 - Verwendet dann den zugehörigen Quellwert.
 - Ohne Treffer bleibt der Wert unverändert.
3. Einsetzen des Ergebnisses in {{value}}.

Beispiel Zahl mit Skalierung:

Skalierungsfaktor `0.1`, Offset `0`, Steuerwert `23.5` → Rückrechnung $23.5 \div 0.1 = 235$ → kein Text-Mapping konfiguriert → Payload `235`.

Beispiel mit Text-Mapping:

Text-Mapping `{„ON“: „true“, „OFF“: „false“}`, Steuerwert `true` (Boolean) → Schritt 1 liefert `„true“` → Schritt 2 findet den Eintrag mit Zielwert `„true“` und liefert `ON`.

Control-Pattern

Das Control-Pattern bestimmt den Payload des Steuerbefehls. Der Platzhalter {{value}} wird durch den aufbereiteten Wert ersetzt.

Beispiele:

Control-Pattern	Gesendeter Payload (Wert: true)
{{value}}	true
{"state": "{{value}}"}"	{"state": "true"}
SET_{{value}}	SET_true
1 (ohne Platzhalter)	1 (statischer Wert)

Wenn kein Platzhalter angegeben ist, wird immer der gleiche statische Payload gesendet.

Publish-Retained

Bei aktiviertem Publish-Retained wird der Steuerbefehl als Retained-Nachricht veröffentlicht. Der Broker speichert dann den letzten Befehl und sendet ihn an neue Subscriber sofort beim Verbinden.

2.9 Konfigurationsbeispiele

2.9.1 Temperatursensor (NUMBER mit Skalierung)

Ein Sensor sendet Temperaturwerte als Ganzzahl in Zehntelgrad, zum Beispiel 235 für 23,5 °C.

Feld	Wert
Topic	sensor/temperature/raw
QoS	0
Retained	true
Payload-Typ	NUMBER
Skalierungsfaktor	0.1
Offset	0

Eingehender Payload 235 → $235 \times 0.1 + 0 = 23.5$

2.9.2 Schalter mit Text-Payload (TEXT + Text-Mapping)

Ein Schalter sendet ein oder aus statt true/false.

Feld	Wert
Topic	home/switch/status
QoS	1
Retained	true
Payload-Typ	TEXT
Text-Mapping	{"ein":"true","aus":"false"}

Eingehender Payload „ein“ → Text-Mapping → true → Boolean true

2.9.3 JSON-Gerät mit verschachteltem Wert (JSON + JSON-Pointer)

Ein Gerät sendet strukturierte Statusdaten als JSON:

```
{„device“: {„power“: 1850, „unit“: „W“}}
```

Feld	Wert
Topic	device/energy
QoS	1
Retained	false
Payload-Typ	JSON
JSON-Pointer	/device/power

Eingehender Payload → JSON-Parse → Pointer /device/power → Wert 1850 → **1850**

2.9.4 Sensor mit Einheit im Payload (NUMBER + Input-Extractor)

Feld	Wert
Topic	sensor/temp
Payload-Typ	NUMBER
Input-Extractor	<code>[-+]? \d+ (?: [.,] \d+)?</code>

Der Input-Extractor extrahiert 23.5, das Number-Pattern parst daraus die Zahl, das Ergebnis ist 23.5.



In diesem Fall übernimmt der Input-Extractor und das Number-Pattern dieselbe Arbeit. Es reicht, nur das Number-Pattern zu setzen; der Input-Extractor ist zusätzlich nützlich, wenn vor der Zahlenextraktion noch eine Textersetzung (Text-Mapping) stattfinden soll.

2.9.5 Bidirektionale Steuerung (Lampe ein-/ausschalten)

Eine Lampe empfängt den Status auf einem Topic und akzeptiert Steuerbefehle auf einem separaten Topic. Das Steuersystem sendet true oder false; der MQTT-Broker erwartet ON bzw. OFF.

Status-Empfang:

Feld	Wert
Topic	home/light/status
QoS	1
Retained	true
Payload-Typ	TEXT
Text-Mapping	<code>{"ON": "true", "OFF": "false"}</code>

Steuerung:

Feld	Wert
Control-Topic	home/light/set
Control-Pattern	<code>{{value}}</code>
Publish-Retained	false

Steuerwert true → Control-Pattern → gesendeter Payload: true

Wenn der Broker stattdessen ON/OFF erwartet:

Control-Pattern	Steuerwert true → Payload
<code>{{value}}</code>	true

2.9.6 Prozentwert aus JSON extrahieren und skalieren

Ein Gerät sendet den Dimmwert in Prozent als JSON:

```
{„state“: „ON“, „brightness_pct“: 75}
```

Das Feature erwartet einen Wert zwischen 0 und 1.

Feld	Wert
Topic	zigbee2mqtt/lamp/state
Payload-Typ	JSON
JSON-Pointer	/brightness_pct
Skalierungsfaktor	0.01
Offset	0

Eingehender Payload → Pointer /brightness_pct → 75 → $75 \times 0.01 = 0.75$

homematic

Homematic IP ist eine Marke der eQ-3 AG.

eQ-3 AG
Maiburger Straße 29
26789 Leer
www.eq-3.de
support@eq-3.de

Tel.: +49 491 600 8 600
Fax: +49 491 600 899 600

Änderungen und Irrtümer vorbehalten.