

# MQTT vs. HTTP - Vorteile und Nachteile

Eine häufig gestellte Frage im LoxForum ist: "Was bringt mir die Anbindung per MQTT im Vergleich zu HTTP?"

Diese Frage liest man meist im Zusammenhang mit Shelly-Geräten, aber auch mit anderen sowohl HTTP- als auch MQTT-fähigen Geräten, wenn es darum geht, diese Geräte mittels HTTP anzusteuern, oder mittels [MQTT Gateway](#) Plugin.

Hier gehen wir der Sache, am Beispiel eines Shelly 1 PM (Shelly 16A Aktor mit Strommessung) auf den Grund.

Die Wertungen sind ein gewisser Anhaltspunkt. Es kann natürlich jeder für sich entscheiden, wie wichtig ein Thema beim eigenen Betrieb ist.

## Daten senden (Miniserver → Gerät)

In Bezug auf Shelly-Geräte könnte das Senden von Daten zum Beispiel das Ein- und Ausschalten eines Relais sein, oder das Einstellen des Dimming-Levels, oder die RGB-Farbe eines LED-Bandes.

| HTTP-Verbindung  | MQTT-Verbindung  |
|--|--|
| Pro Gerät ein Virtueller HTTP-Ausgang; Virtuelle Ausgangsbefehle | Einmalig ein Virtueller UDP-Ausgang; Virtuelle Ausgangsbefehle |

Es handelt sich um eine DIREKTE Verbindung vom Miniserver zum Gerät - kein zusätzliches Gerät (LoxBerry) dazwischen. Damit ist diese Lösung ausfallsicherer. - Wenn der LoxBerry ausfällt, kann das Gerät nicht mehr gesteuert werden.

\* Eine Überwachung des Zustands kann direkt als virtueller Eingang eingerichtet werden (keepaliveepoch) | - Es gibt kein direktes Feedback, ob die Schaltaktion ausgeführt wurde (Loxone kann die Antwort des HTTP-Befehls nicht auswerten). | + Eine Statusänderung des Geräts wird sofort per MQTT zurück übertragen. | + Am Gerät muss keine besondere Einrichtung erfolgen (außer evt. Zugangsdaten definieren). Alles wird in der Loxone Config definiert. | - Am Gerät müssen einmalig die Zugangsdaten zum MQTT Broker angegeben werden. | + Die Schaltaktion oder Wertänderung wird möglicherweise schneller ausgeführt. | - Durch das Processing am LoxBerry kann - je nach Raspberry-Modell - mit einer Verzögerung von 5 bis 10 ms gerechnet werden, bis der Befehl am Gerät ankommt. | - Die HTTP-Übertragung vom Miniserver hat größeren Protokoll-Overhead, der vom Miniserver und dem Empfängergerät verarbeitet werden muss. | + Die UDP-Verbindung vom Miniserver zum MQTT Gateway ist sehr schlank. UDP erfordert keine Rückmeldung an den Miniserver. Die MQTT-Verbindung selbst ist eine offene Verbindung und wird binär übertragen, kein eigener Verbindungsaufbau erforderlich. | - Für mehrere Geräte müssen mehrere Virtuelle Ausgänge und Virtuelle Ausgangs-Befehle eingerichtet werden. | + Es wird nur ein Virtueller Ausgang zum MQTT Gateway eingerichtet. Alle Geräte sind unter diesem virtuellen Ausgang mit verschiedenen Befehlen eingerichtet. | - Die Netzwerkinfrastruktur (Hostnamen, IP-Adressen) muss korrekt gepflegt werden.

Wird mit Hostnamen gearbeitet, muss der DNS-Server für alle Geräte immer korrekt funktionieren.

\* Wird mit IP-Adressen gearbeitet, müssen an den Geräten feste IP-Adressen, oder feste DHCP-Leases vergeben werden.

\* Je nach Infrastruktur und Netzwerk-Erfahrung könnte ein Wechsel des Internet Routers schon erhebliche Sorgen verursachen. | + Ausschließlich der MQTT Broker (also der LoxBerry) muss IMMER erreichbar sein (egal ob mit Hostname oder fester IP)

\* Alle Geräte (der Miniserver und alle MQTT-Geräte) verbinden immer zum MQTT Broker.

\* Da der Broker selbst nie eine Verbindung zu den Geräten aufbaut, sondern die Geräte zum Broker, benötigt der Broker weder Hostname noch IP-Adresse der Geräte. | + Shelly verwendet eine klare HTTP API als REST Interface (GET-Requests)

- Andere Geräte oder Software verwendet möglicherweise ganz andere Aufruf-Syntaxen, mit Query-Parametern, POST- oder PUT-Requests, Digest Authentication,... Jeder Hersteller hat hier eigene Anforderungen. | + MQTT hat einen standardisierte, klaren Aufbau (Topics), wie Parameter an ein bestimmtes Gerät übertragen werden. Die Parameter jedes Gerätes sind individuell, folgen aber der klaren Struktur von MQTT.

+ Geräte oder Software anderer Hersteller, die MQTT-fähig sind, haben zwar andere Parameter, aber die gleiche Struktur. | ===== Daten empfangen (Gerät->Miniserver) ===== In Bezug auf Shelly-Geräte könnte das der Relais-Status des Geräts sein, oder eingestellten Farbwerte, bei Geräte mit Sensoren die Temperatur oder Luftfeuchte, ein Tastendruck, Fensterkontakte oder bei Geräten mit Strommessung die aktuelle Leistungsaufnahme in Watt. Hier unterscheidet sich HTTP von MQTT **wesentlich**: Bei MQTT werden Daten direkt geliefert, während bei HTTP der Miniserver die Daten 24/7 in einem starren Intervall abfragen muss. | HTTP-Verbindung

Pro Gerät URL ein Virtueller HTTP-Eingang; Virtuelle Eingangsbefehle mit Befehlskennungen | MQTT-Verbindung

Für jeden Wert ein Virtueller Eingang | - Das minimale Abfrageintervall von Loxone ist 10 Sekunden. D.h. egal ob eine Änderung passiert ist oder nicht, wird mit dem von Loxone vorgegebenen Minimalintervall 14.400 mal pro Tag eine URL abgerufen.

\* Dies multipliziert sich weiter, wenn ein Gerät mehrere URLs benötigt, um alle Daten einzusammeln.

\* Dies multipliziert sich nochmals für jedes weitere Gerät.

\* Loxone erlaubt keine Abfrage von Daten "on demand" (z.B. nur bei Anwesenheit, nur wenn eine Logik eine Abfrage auslöst,...)

\* Beispielsweise die Abfrage eines Tastendrucks oder Doppelklicks kann damit überhaupt nicht realisiert werden. | MQTT ist eine offene Verbindung zwischen Gerät und Broker. Das Gerät sendet bei einer Änderung sofort diese Daten an den Broker.

\* Damit gibt es faktisch keine Wartezeit, bis die Daten übertragen werden.

\* Wenn hingegen keine Daten zu melden sind, passiert einfach nichts (kein sinnloses Pollen)

\* Die Strommessdaten der Shellys werden beispielsweise immer aktualisiert (auch mehrmals pro Sekunde, wenn erforderlich) - bleibt der Verbrauch dann aber konstant, müssen keine neuen Daten übertragen werden.

\* Beispielsweise wird der Tastendruck oder Doppelklick auf einen Taster "on-the-fly" übertragen und kann verwendet werden.

|   |  |
|---|--|
| <p>- Jede einzelne Befehlskennung innerhalb eines Virtuellen HTTP-Eingangs löst eine Suche innerhalb der HTTP-Antwort aus und belastet damit RAM und CPU des Miniservers</p>  | <p>+ Jeder einzelne Datensatz wird genau in seinen fest definierten Virtuellen Eingang geschrieben. Am Miniserver ist keine Suche in einem HTML oder dergleichen erforderlich.</p>   |
| <p>- Loxone unterstützt nur GET-Requests (keinen POST- oder PUT-, keine Request Header usw.)</p>  | <p>+ Die Übertragung von MQTT ist standardisiert.<br/>* Die Übertragung vom Gerät zum MQTT Gateway erfolgt standardisiert, es gibt hier keine unterschiedlichen Methoden.</p>  |
| <p>- Die Erstellung von Befehlskennungen kann - je nach Art der Antwort - sehr mühsam sein, oder das Finden des Datensatzes ist überhaupt nicht möglich.</p>  | <p>+ Daten werden bei MQTT in zwei Arten übertragen:<br/>* Der "pure" Wert → wird direkt übermittelt<br/>* Ein JSON-Datensatz (auch verschachtelt) → Das MQTT Gateway zerlegt diese Daten, sodass am Miniserver die eindeutigen Werte einzeln übernommen werden können.</p>  |
| <p>- Der Virtuelle HTTP-Eingang unterstützt keine Texte. Texte können nicht dargestellt werden.</p>   | <p>+ Wird in Loxone statt eines "Virtuellen Eingangs" ein "Virtueller Texteingang" erzeugt, wird der per MQTT übermittelte Text für die Miniserver-Visualisierung übermittelt.</p>   |
| <p>- Der Virtuelle HTTP-Eingang unterstützt keine textuellen Stati ("ein", "aus", "true", "false", "open", "closed" usw.)<br/><br/>* Diese Texte müssen mit komplizierten Befehlskennungen (\1 um einen ASCII-Code eines Buchstabens zu erhalten, dann wieder zusammensetzen mittels Statusbaustein) erfolgen.</p>  | <p>Für textuelle Stati gibt es im MQTT Gateway zwei Verfahren:\\ * Booleans (z.B. "on", "off", "true", "false",...) werden vom MQTT Gateway vollkommen automatisch in die numerischen Entsprechungen 1 oder 0 übersetzt.\\ * Spezielle Texte, wie den Status von Fenstern ("offen", "gekippt", "geschlossen") können per Conversion in selbst definierbare Werte (z.B. 1, 2, 3) übersetzt werden, und damit kann im Miniserver direkt weitergearbeitet werden.\\ * Per Conversion können auch Texte in andere Texte konvertiert werden: z.B. "undefined" nach "Keine Information verfügbar", um dies in einem Virtuellen Texteingang anzuzeigen.\\ * Fertige Conversions können von LoxBerry Plugins mitgeliefert werden, die MQTT unterstützen.   - Gleiche Stati werden immer und immer wieder abgefragt. Auch Daten, die man in der HTTP-Antwort nicht benötigt, müssen durchsucht werden.   Auch bei MQTT ist es möglich, dass ein Gerät regelmäßig seinen aktuellen Status übermittelt (beispielsweise Shelly übermittelt per MQTT zusätzlich zu direkten Statusänderungen auch alle 120 Sekunden den aktuellen Status).<br/><br/>* Das MQTT Gateway erkennt gleiche Daten, und übermittelt nur Änderungen an den Miniserver.<br/>* Dies entlastet den Miniserver wesentlich, da die Übertragung vieler gleicher Werte einfach wegfällt.<br/>* Daten, die man nie am Miniserver benötigt, kann man explizit wegschalten</p> |
| <p>- Der Ausfall eines Geräts kann frühestens erkannt werden, wenn es das nächste Mal abgefragt wird, und das auch nicht eindeutig. Der "Fehlerausgang" von Loxone ist fehlerhaft und unausgereift:<br/><br/>* Die Wertvalidierung setzt einen Standardwert, der zu Fehlverhalten in der Logik führen kann, und daher selbst wieder extra behandelt werden muss.<br/>* Liefert der HTTP-Abfrage ein unerwartetes Ergebnis (z.B. eine Fehlermeldung), sodass die Befehlskennung den Wert nicht mehr findet, geht der "Fehlerausgang" nicht auf 1 und der letzte Wert bleibt erhalten.<br/><br/>Damit ist keine sichere Fehlerhandlung möglich.</p> | <p>+ Die Technik des "Last Will And Testament" (LWT) von MQTT ermöglicht es, dass ein ausgefallenes Gerät automatisch als "Offline" gemeldet wird, wenn es ausgefallen ist.<br/><br/>* Dieser Status kann genauso wie andere Virtuelle Eingänge benutzt und für eigene Validierungszwecke weiterverarbeitet werden.<br/>* MQTT-Geräte gehen automatisch auf "offline", ohne dass dies extra abgefragt werden müsste.<br/>* Die Aktivität des MQTT Gateways selbst kann mit dem "keepaliveepoch"-Topic überwacht werden.</p>  |

# Allgemein

- Durchaus kann es Sinn machen, den Kompromiss zu wählen, und die Steuerung per HTTP durchzuführen, aber die Statusrückmeldung per MQTT zu empfangen (diesen Ansatz verwendet beispielsweise das [NUKI SmartLock Plugin](#)).
- Hast du es mittels MQTT Gateway einmal geschafft, die Vorgehensweise zur Einrichtung in beide Richtungen hinzubekommen, ist die Systematik bei allen weiteren Geräten immer die gleiche.
- Es steht auf der Roadmap von LoxBerry, dass das [MQTT Gateway](#) Plugin in den LoxBerry-Core übernommen wird und somit jedem LoxBerry ohne zusätzliche Installation zur Verfügung steht.
- Die Verwendung von MQTT zur Übertragung von Daten von einem Plugin an den Miniserver ist die Empfehlung für neue LoxBerry-Plugins. Für Benutzer bringt das eine einheitliche Einrichtung, für Entwickler den Wegfall der komplexen UIs zur Definition der Übertragungsweise und die Vereinfachung der Datensammlung/Übertragung im Plugin.

From:

<https://wiki.loxberry.de/> - **LoxBerry Wiki - BEYOND THE LIMITS**

Permanent link:

[https://wiki.loxberry.de/haufig\\_gestellte\\_fragen\\_faq/mqtt\\_vs\\_http\\_vorteile\\_und\\_nachteile](https://wiki.loxberry.de/haufig_gestellte_fragen_faq/mqtt_vs_http_vorteile_und_nachteile)

Last update: **2022/09/10 12:18**