


MQTT Gateway

Autor	Christian Fenzi
Logo	
Status	RELEASE
Version	1.1.4
Min. LB Version	LB 1.4.0
Download	Release: https://github.com/christianTF/LoxBerry-Plugin-MQTT-Gateway/archive/1.1.4.zip Pre-Release: -
Zweck	Installiert den MQTT-Broker Mosquitto und stellt ein MQTT-Gateway für den Miniserver bereit.
Languages	EN
Diskussion	https://www.loxforum.com/forum/projektforen/loxberry/plugins/176025-loxberry-mqtt

Version 0.1.1

- DEVELOPER PREVIEW - Nur einsetzen mit MQTT und Linux-Kenntnissen - Fehler ausschließlich per Issue bei GitHub
- Stellt ein MQTTLoxone Gateway bereit

Version 0.1.2

- DEVELOPER PREVIEW - Nur einsetzen mit MQTT und Linux-Kenntnissen - Fehler ausschließlich per Issue bei GitHub
- Implementiert ein UDP-Interface für LoxoneMQTT

Version 0.1.3

- DEVELOPER PREVIEW - Nur einsetzen mit MQTT und Linux-Kenntnissen - Fehler ausschließlich per Issue bei GitHub
- Eigene UI-Sektion für MQTT Broker
- Authentifizierung für MQTT Broker
- Lokaler Mosquitto abschaltbar
- Retry für den Mosquitto Service, wenn er beim Systemstart nicht automatisch gestartet wurde
- MQTT Gateway wird direkt nach der Installation gestartet
- Parameter-Update-Routine zum Hinzufügen von neuen Default-Parametern während der Installation

Version 0.1.4

- DEVELOPER PREVIEW - Nur einsetzen mit MQTT und Linux-Kenntnissen - Fehler ausschließlich per Issue bei GitHub
- UI: Expand JSON Option hinzugefügt
- Gateway: JSON wird bei HTTP expandiert (nur erster Level) und beim HTTP-Namen angehängt - EXPERIMENTELL
- Gateway: Alle übermittelten Topics werden im Speicher behalten (max. 24 Stunden)
- UI: Funktion zum Abfragen der gespeicherten Gateway-Topics (wird aber noch nicht dargestellt)

Version 0.1.5

- DEVELOPER PREVIEW - Nur einsetzen mit MQTT und Linux-Kenntnissen - Fehler ausschließlich per Issue bei GitHub
- UI: Größerer Umbau des UI (Settings, Subscriptions, Conversions, Incoming Overview, Logfiles) als Navbar
- UI: Subscriptions jetzt als eigene Seite (leichter anzupassen, man muss nicht immer auf- und abscrollen für Apply)
- UI: Conversions: Angabe von Strings, die zu Werten konvertiert werden sollen (Loxone kann mit Strings nichts anfangen)
- UI: Incoming Overview: Anzeige der an den Miniserver übermittelten Werte
- UI: Logs: Inline-Anzeige der Logfile-Übersicht
- Gateway: Benutzerdefinierte Conversion (eingehende Nachrichten können in Werte konvertiert werden)
- Gateway: Logfile: Parsing der Conversion wird geloggt, inklusive "komischer" Werte und Duplikate
- Gateway: Logfile wird beim Beenden des Gateways sauber abgeschlossen

Version 0.2.0 PRE-RELEASE

- Erster Pre-Release
- Allgemein: Neues Logo (MQTT-Logo statt Mosquitto-Logo)
- Gateway: Besseres Fehlerhandling beim Restart des Gateways
- UI: Filter bei den übermittelten Nachrichten (erleichtert die Suche)
- Donation: Erster Versuch, per Spende einen eigenen Test-Miniserver zusammen zu bekommen

Version 0.3 PRE-RELEASE

- Lokale Mosquitto-Authentifizierung als Standard (Benutzer: loxberry, Passwort: wird generiert)
- Broker-Anmeldedaten werden nur dann geladen, wenn der SecurePIN angegeben wird
- Ermöglicht direkt, die lokalen Mosquitto-Anmeldedaten zu ändern oder zu deaktivieren
- Änderung der Anmeldedaten eines nicht lokalen Brokers ändert nur die Anmeldung des Gateways
- Änderung der Konfiguration übernimmt diese sofort für das MQTT Gateway und/oder Mosquitto

Version 0.3.1 PRE-RELEASE

- Fix: Bei anonymer Verwendung von Mosquitto startet der Mosquitto-Dienst nach Reboot nicht mehr
- Aktivierung von Plugin AutoUpdate (ab dieser Version können Versions-Notifications oder automatisches Update aktiviert werden)

Version 0.4.1 PRE-RELEASE

- Fix: Nach Konfigurationsänderungen und "Apply" wurden diese nicht übernommen
- Neu: Per UDP eingehende Nachrichten werden vollständig weitergegeben (so können beispielsweise auch json Nachrichten gesendet werden)
- Neu: Das Gateway published seinen Connection-Status selbst unter dem Topic <hostname>/mqttgateway/status
- Neu: Das Gateway published im Minutentakt den aktuellen Epoch-Timestamp.
- Neu: Das Gateway subscribed sich selbst automatisch - so kann der Timestamp als Prüfung verwendet werden, ob das Gateway noch funktioniert
- Neu: Unterstützt Last Will and Testament (d.h. Broker setzt den Status auf Disconnected, wenn das Gateway stirbt)
- Neu: UDP-Nachrichten können mit publish oder retain gesendet werden
- Neu: UDP-Nachricht "reconnect" forciert das Neu-Verbinden und Übertragen an den Miniserver
- Neu: Default-UDP-In-Port am Gateway: 11884 (bisher 11883)

Version 0.5.0 PRE-RELEASE

- Fix: Evt. falsche Auswertung der Checkboxes im UI (#5)
- Erweitert: JSON Expansion: Vollständig generische Expandierung ohne Einschränkung auf die Hierarchie
- Erweitert: JSON Expansion für HTTP *und* UDP

Version 0.5.1 PRE-RELEASE

- Fix: JSON expansion not working

Version 0.6.0 PRE-RELEASE

- FIX: Anonymous connection was still possible with password auth (after installation, please save once to fix)
- Enhanced: Incoming Overview: Now is a live visualisation of the topic messages (updated once a second, without browser refresh)
- Enhanced: Incoming Overview: Enable advanced infos to show the original topics, and to delete messages on the broker
- Info for delete event: The data in the overview may not disappear, as the last transmission with "last sent to Miniserver" is still valid

Version 0.7.1 RELEASE

- FIX: Json expansion for UDP only worked when HTTP was enabled
- Added: Overview now shows ready-to-use command recognition for UDP inputs
- Added: Subscription topics are now validated on-the-fly (errors are displayed on the left)

Version 0.8.0 PRE-RELEASE

- FIX: Password was re-created on every update
- New: mosquitto.log on the Logfiles tab
- New: TLS-PSK for encrypted connections, listening on port 8883
- New: In the Incoming Overview, with Advanced Table Info enabled, you can **disable caching for selected data**, and **enable Reset-After-Send**
 - **Disable Cache:** Every incoming value from MQTT will be sent again, also if it has not changed.
 - **Reset-After-Send:** This is especially useful to generate impulses/triggers, if your device e.g. on a keypress does not send "pressed" and "released" events, but only one "pressed" impulse. Reset-After-Send will send a 0 after each incoming event.
 - UI: Enabling and disabling the checkbox responses with some seconds delay - the checkbox toggles back and after some seconds, shows the currently set value.
- Requires LoxBerry V1.4.0+

Version 0.8.1 RELEASE

- Incoming values are trimmed (removed leading and terminating blanks) to work with the Text-To-Value Conversion.

Version 0.8.2 PRE-RELEASE

- Updated Perl Net::MQTT::Simple lib to 1.23
- Corrects a connection issue of the lib to Mosquitto with version higher V1.5.8

Version 0.8.3 RELEASE

- Performance improvements for the Incoming Overview
 - Close HTTP collapsible if UDP is opened, and vice-versa
 - Only render opened collapsible
 - Optimized updating HTML table
 - Update interval of the Overview is calculated dependent to the number of elements
 - "Advanced Table View" checkboxes now are viewed in classic design instead of jQuery Mobile style

Version 0.9.0.1 PRE-RELEASE

- Incoming Overview: Button to purge Mosquitto's retain database (local Mosquitto installation only)
- Incoming Overview: Button to retransmit all data to Miniserver without cache (for testing of Miniserver configuration)
- Incoming Overview: Incoming data are shown also if no Miniserver is configured
- Restart button also restarts Mosquitto (local Mosquitto installation only). With external broker, still only the Gateway is restarted
- Mosquitto:
 - MQTT Websocket protocol. Websocket is available on port 9001. Use the same credentials. Websocket TLS currently not supported. (after update, Mosquitto may need to be restarted)
 - Changed Mosquitto database auto-save interval to 1 day (before: 30 min) to reduce sd writes. Shutdown/Restart still triggers saving of the db on sd card.
- New installations (not update) changed default configuration:
 - HTTP Transmission : enabled
 - UDP Transmission: disabled
 - Expand JSON data: enabled
- Gateway: Fixed wrong encoding in topic names (e.g. with umlauts)
- Plugin developers:
 - Plugin's UDP interface now also supports json as data format (compared to the simple udp interface, this also supports blanks in topic names) [MQTT Gateway - HTTP- und UDP-Interface](#)
 - Your own plugin can inject subscriptions, conversions to the MQTT Gateway plugin during installation and runtime [MQTT Gateway for plugin developers](#). Your injections are applied from the Gateway on-the-fly without restart.

Version 0.9.1 PRE-RELEASE

- Generic POST-GET-JSON Receiver - Generic Callback URL for non-MQTT devices: [MQTT-Gateway - Generischer POST / GET / JSON Empfänger](#)
- Fixed: JavaScript error

Version 0.9.2 PRE-RELEASE

- Fixed an error with Mosquitto during update of the plugin. Mosquitto did not update properly during the installation process.

Version 1.0 RELEASE

- Uninstall will uninstall Mosquitto

Version 1.0.1 RELEASE

- Enhanced: Implemented a delay of 10ms on "reset-after-send" topics between sending the value and sending the 0 (Miniserver network processing was too slow in some situations). The delay is adjustable in the mqtt.json config file (Main.resetaftersendms)
- Enhanced: Boolean Conversion does not translate an empty string to 0 anymore. Empty now stays empty.
- Enhanced: Deleting a value in the webif (X button) now also deletes the line in the webif.
- Enhanced: Removed logfile entries about Incoming Overview state requests for more clearness in the log
- Fixed: Corrected encoding (utf8) inconsistencies via the different interfaces (mqtt, udp, http, webif), e.g. with German umlauts.

Version 1.1 PRE-RELEASE

- New: Publish test messages from the UI: Open the right-side help flyout and open "Quick Publisher"
- New: Send different subscription data to different/multiple Miniservers
- Enhanced: Daemon creates a logfile on startup
- Fixed: Overview: Delete button was hidden in some resolutions

Version 1.1.1 RELEASE

- Updated Net::MQTT::Simple library to 1.24, with LoxBerry/Mosquitto rapid public patch (see <https://github.com/Juerd/Net-MQTT-Simple/issues/11>)

Version 1.1.2 PRE-RELEASE

- Allows to skip forwarding of selected data to the Miniserver from UI (to save Miniserver resources)
- Fixes topic matching for redirection of data to other Miniservers (pipe in the subscriptions)

Version 1.1.3 PRE-RELEASE

- Introducing healthcheck for LoxBerry Healthcheck (LB2.2+)
- Fixed another issue with topic matching for redirection of data to other Miniservers

Version 1.1.4 RELEASE

- Update for Net::MQTT::Simple library (adjusted send delay for Raspberry Pi 4 to 17ms)

Inhalt

- [Download](#)

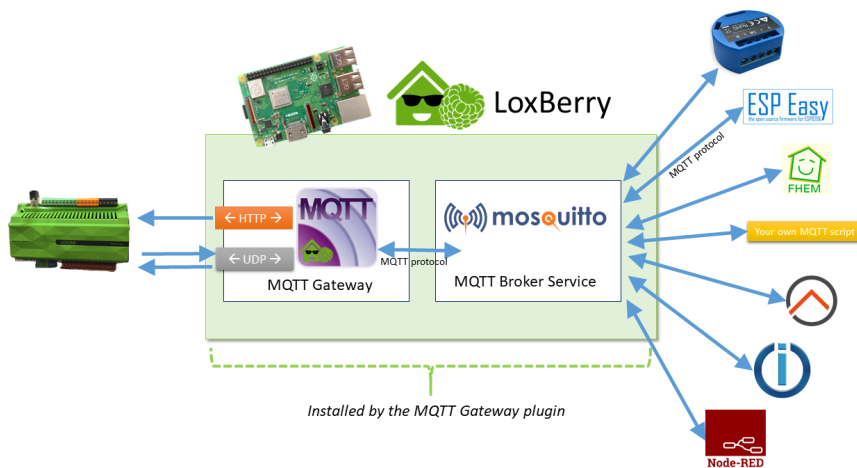
- Kommunikationsdiagramm
- Begriffserklärung
- Funktion des Plugins
- Schnell-Konfiguration am Beispiel eines Shelly 2
- MQTT an Loxone Miniserver
 - Einstellungen
 - Siehe auch: MQTT - Schritt für Schritt: MQTT -> Loxone
- Miniserver an MQTT
 - Siehe auch: MQTT - Schritt für Schritt: Loxone -> MQTT
- MQTT Broker-Einstellungen
- MQTT Subscriptions
 - Verteilung der Daten an verschiedene Miniserver (ab V1.1)
- Text-to-Value Conversions
- Incoming Overview
 - Show advanced table information
- Bekannte MQTT-Geräte und Subscription-Topics
- Weitere Hinweise
- Roadmap
- Fragen stellen und Fehler melden
 - Fehlerbeschreibungen
- Siehe auch

Download

Der direkte Download des Releases befindet sich in der Tabelle oben.

Repository: <https://github.com/christianTF/LoxBerry-Plugin-MQTT-Gateway>

Kommunikationsdiagramm



(Anklicken zum Vergrößern)

Sieh dir links im Navigationsbaum weiterführende Artikel zu MQTT an, um die Funktionsweise besser zu verstehen.

Begriffserklärung

Ein **"Broker"** ist ein Dienst, der als Vermittler von Nachrichten dient. Der Broker speichert auch den aktuellen Zustand aller angeschlossenen Geräte. Das Plugin installiert automatisch am LoxBerry den Mosquitto MQTT Broker als Vermittler. Über die Konfiguration kann auch ein Broker auf einem anderen Server angegeben werden. Das Plugin unterstützt Authentifizierung, jedoch keine Verschlüsselung zum Broker.

Ein **"Server"** ist in der Begrifflichkeit von MQTT jedes Gerät, das Daten einliefert. Beispielsweise sind die Shelly's MQTT-Server, die Daten an den Broker übermitteln.

Ein **"Client"** ist jede Software, die sich mit dem Broker verbindet und dort Stati abfragt, Befehle sendet bzw. auf Statusänderungen lauscht. Dieses Plugin ist so ein Client, der zum Broker verbindet und sich Statusaktualisierungen abonniert (subscribed). Ist vom Client ein MQTT-Topic abonniert, werden Daten dieses Topics vom Broker an diesen Client zur Verarbeitung weitergeleitet.

Funktion des Plugins

Das Plugin installiert zuerst automatisch den Mosquitto MQTT Broker (Hostname "localhost"). Erfahrene Benutzer können auch eine Verbindung zu einem anderen Broker einrichten. Das Plugin unterstützt kein TLS zum MQTT-Broker.

Das Plugin fungiert dann als Client an diesem MQTT-Broker, und leitet MQTT-Nachrichten an den Miniserver weiter, sowie empfängt Nachrichten vom Miniserver und leitet diese an den MQTT-Broker. Über das Webinterface können Topics abonniert werden. Die abonnierten Topics werden wahlweise per UDP, oder per HTTP Webservice an den eingestellten Miniserver übermittelt.

Das Senden von Nachrichten vom Miniserver an MQTT geht am besten per UDP.

Schnell-Konfiguration am Beispiel eines Shelly 2

MQTT an Loxone Miniserver

Einstellungen

Einstellung	Standard	Beschreibung
Miniserver	Erster Miniserver	Die abonnierten Daten werden an diesen Miniserver gesendet.
Miniserver-UDP-Port	11883	Ist UDP aktiviert, werden Nachrichten an diesen UDP-Port des Miniservers übermittelt.
Per REST Webservice übermitteln		<p>Ressourcenschonender als UDP!</p> <p>Die Nachricht wird direkt per HTTP-REST Webservice an den Miniserver übertragen. Dafür müssen selbst entsprechende Virtuelle Eingänge mit exakt dem hier dokumentierten Namen verwendet werden. Das Plugin verwendet die Nachricht als Bezeichnung für den Eingang, wobei / in _ konvertiert werden:</p> <p>shellies/shelly-12345/relay/1 wird zu shellies_shelly-12345_relay_1</p> <p>Schau ins Logfile, dort werden die Bezeichnungen angezeigt.</p>
Per UDP übermitteln		<p>Nachrichtenformat (für die Befehlsenerkennung):</p> <p>MQTT: shellies/shelly-12345/relay/1=off (oder =0)</p> <p>Befehlsenerkennung: MQTT: shellies/shelly-12345/relay/1=\v</p>
Booleans konvertieren	Ein	In MQTT werden häufig Strings bei der Nachricht verwendet, beispielsweise "off" oder "on". Das Plugin konvertiert als Booleans identifizierte Strings in die Werte 0 oder 1. Strings, die erkannt werden, siehe LoxBerry::System::is_enabled bzw. LoxBerry::System::is_disabled
JSON expandieren		<p>Werden JSON-Daten übertragen, wird die JSON-Struktur expandiert und an den http-Namen angehängt.</p> <p>Topic/abcde={"action":"on","linkquality":39} wird zu Topic_abcde_action = on (oder 1) Topic_abcde_linkquality = 39</p> <p>Auch Objekte und Arrays in beliebiger Hierarchie werden aufgelöst. Am besten in der Incoming Overview anschauen, wie die Daten nach der Expandierung aussehen.</p>
TAB Subscriptions Abonnierte Topics		<p>Die Angabe erfolgt mit einer Zeile pro Topic. Die Topics müssen vom MQTT-Device dokumentiert sein. Eine # definiert dabei einen "Joker" (wie man sonst den * verwendet), z.B. shellies/# abonniert alle Meldungen innerhalb des Topics shellies/.</p> <p>Siehe unten die Kurzübersicht bekannter MQTT-Geräte und Topics.</p>
TAB Conversions Eigene Wert-Zu-Text Konvertierung		<p>Da der Miniserver keine Texte weiterverarbeiten kann, können eingehende Nachrichten von einem String zu einem Wert konvertiert werden.</p> <p>Beispiel:</p> <p>button-up=1 button-down=2</p> <p>Sendet ein Device Topic_abcde_status=button-down, wird dies als Topic_abcde_status=2 an den Miniserver übertragen.</p> <p>Beispielsweise in einem Status-Baustein kann der Wert weiter verarbeitet werden.</p>

TAB Incoming Overview Anzeige der übermittelten Daten	Hier werden die per HTTP und UDP übermittelten Daten angezeigt. Die Aktualisierung erfolgt automatisch im Sekundentakt, ohne dass ein Browser-Refresh notwendig ist. In diesem Fenster können mit Copy&Paste virtuelle Eingänge/Texteingänge, oder für UDP die entsprechenden Befehlserkennungen kopiert werden.
---	--

Den Button zum Speichern ganz unten nicht vergessen. Die neue Konfiguration wird beim Speichern automatisch übernommen.

Siehe auch: [MQTT - Schritt für Schritt: MQTT -> Loxone](#)

Miniserver an MQTT

Einstellung	Standard	Beschreibung
Gateway UDP IN-Port	11884	Eingangsport am LoxBerry für UDP-Nachrichten, die an MQTT-Geräte weitergeleitet werden. Folgende Befehle werden unterstützt: <ul style="list-style-type: none"> • <code><topic> <value></code> löst eine Publish-Nachricht aus • <code>publish <topic> <value></code> löst eine Publish-Nachricht aus • <code>retain <topic> <value></code> löst eine Retain-Nachricht aus (wird am Broker gespeichert) • <code>reconnect</code> Löst eine Neuverbindung und ein forciertes Übermitteln der Daten an den Miniserver aus Beispiel 1: <code>shellies/shellyswitch-32BA7F/relay/1/command on</code> Beispiel 2: <code>retain wohnzimmer/temperature 12.5</code> Beispiel 3: <code>reconnect</code>

Wenn der Miniserver neu startet, empfiehlt es sich, mittels der Loxone-Zeit "Startimpuls" ein `reconnect` per UDP-Nachricht an das Gateway zu übermitteln. Beispiel im "Siehe auch"-Link.

Siehe auch: [MQTT - Schritt für Schritt: Loxone -> MQTT](#)

MQTT Broker-Einstellungen

Einstellung	Standard	Beschreibung
Lokalen Mosquitto-Broker verwenden	Ja	Ist dies aktiviert, prüft das Plugin beim Reboot, ob Mosquitto läuft, und startet diesen gegebenenfalls. Aktiv: Lokale Mosquitto-Instanz wird vom Plugin verwaltet (inkl. Änderung von Benutzername und Passwort) Inaktiv: Du kümmerst dich selbst um den Broker
MQTT Broker Adresse	localhost	Hier kann der Hostname angegeben werden, mit dem das Gateway verbinden soll. Wird beim Hostnamen kein Port angegeben, wird der Standardport 1883 verwendet. Eine Portangabe erfolgt per <code>hostname:port</code>
MQTT Broker Benutzername	loxberry	Der automatisch installierte Mosquitto-Broker wird mit dem Benutzer <code>loxberry</code> und einem generierten Passwort installiert. Diese Informationen kannst du bei der Einrichtung von Devices von hier kopieren. Wenn du Benutzername und Kennwort nachträglich änderst, musst du das auch bei allen bereits konfigurierten Geräten machen!
MQTT Broker Passwort	<generiert>	Leerer Benutzer und leeres Kennwort deaktivieren die Authentifizierung am lokalen Broker. Bei externen Brokern wird hier deren Benutzername und Kennwort eingegeben.

MQTT Subscriptions

Hier definierst du, welche Daten du abonnieren möchtest. Jede Subscription steht in einer Zeile. Unter welchem Topic die Daten gesendet werden, bestimmt das Absender-Gerät (bzw. das, was du dort einstellst). Siehe dazu den ausführlichen Artikel, wie Subscriptions und Topics funktionieren, unter [MQTT Gateway - Subscriptions und Topics](#).

Beispiel:

```
shellies/#
nuki/#
```

Um dir die Einrichtung zu erleichtern, können andere Plugins direkt im MQTT Gateway Subscriptions definieren. Wenn Plugin-Subscriptions vorhanden sind, findest du diese auf der *Subscriptions*-Seite unten. Du brauchst dann diese Subscriptions nicht nochmals oben eingeben.

Subscriptions defined by other plugins			
Alexa->Lox alexa2lox/#	Any Plugin meine nuki subscription/ noch eine subscription/#	NUKI SmartLock nuki/#	SqueezeJitter Player squeezejitter/#

Verteilung der Daten an verschiedene Miniserver (ab V1.1)

Wenn du mehrere Miniserver hast (z.B. Gateway-Konzentrator-Betrieb, oder mehrere Wohneinheiten mit unabhängigen Miniservern), möchtest du vielleicht unterschiedliche Daten an unterschiedliche Miniserver senden.

Wenn du eine Subscription ganz normal eingibst (z.B. `shellies/#`), werden deren Daten immer an den Standard-Miniserver gesendet, den du auf der ersten Seite angeben hast.

Mit einer Pipe hinter der Subscription und der oder den Miniserver-Nummern (aus dem Miniserver-Widget) kannst du das Datenziel ändern:

```
shellies/#|2
shellies/shellyem3-DC4F227649B8/#|3
nuki/#|1,2,3
fhem/#
```

- Zeile 1: Alle `shellies/-`Daten werden an den Miniserver 2 gesendet (außer `shellies/shellyem3-DC4F227649B8/`)
- Zeile 2: `shellies/shellyem3-DC4F227649B8/`-Daten gehen an Miniserver 3
- Zeile 3: Daten von `nuki/` werden sowohl an MS1, MS2 als auch MS3 gesendet.
- Zeile 4: Keine Pipe, daher gehen die Daten von `fhem/#` an den Standard-Miniserver

Dazu noch folgende Hinweise:

- Die Priorisierung der Reihung erfolgt entsprechend der Selektivität des Topics, d.h. `shellies/shellyem3-DC4F227649B8/#` ist spezifischer als `shellies/#`, deswegen "gewinnt" in unserem Beispiel bei Daten von `shellyem3-DC4F227649B8` der selektivere Eintrag (Daten gehen an MS3, die restlichen `shellies/#` Daten an MS2). Die Reihenfolge im Textfeld hat keinen Einfluss darauf!
- Die Topic-Joker `+` und `#` werden vollständig unterstützt.
- Du kannst eine Plugin-Subscription auch an einen anderen Miniserver senden lassen, indem du die gleiche Subscription oben im Textfeld mit einer Pipe definierst. Hat das Plugin beispielsweise `nuki/#` registriert, kannst du mit einer Zeile `nuki/#|2` die Daten an einen anderen Miniserver senden lassen.
- Alle Einstellungen auf der *Settings*-Seite und in der Incoming Overview (z.B. HTTP oder UDP, Boolean Conversion, Reset-After-Send,...) treffen für alle Miniserver zu. Das Datencaching von LoxBerry wird pro Miniserver angewandt.
- Fehler bei der Übertragung zu einem Miniserver führen in der Regel zu einer Warnung oder Fehlermeldung im Log, nicht aber zum Abbruch /Absturz des Programms.

Text-to-Value Conversions

Viele Geräte senden Text für ihren Status, z.B. "up" und "down", oder "open" oder "closed". Loxone ist nicht sehr geschickt, was den Umgang und Weiterverarbeitung von Texten angeht. Um diese Texte in Zahlen umzuwandeln (mit denen der Miniserver mehr anfangen kann), kannst du hier Text-to-Value Conversions, also Konvertierungen durchführen.

In jede Zeile eine Konvertierung.

Text-to-Value conversions

MQTT devices often are sending strings that Loxone unfortunately cannot handle. In this field you can create e.g. with a Status block.

Booleans (on, off etc.) are already converted, if you have 'Boolean conversion' enabled.

Text-to-Value conversion

One conversion per line. Use **Text=Value**

```
open=3
tilted=2
open_from_tilted=3
closed=1
```

Dieses Beispiel zeigt den Status von Enocean Fenstergriffen. Sendet der Fenstergriff "open", wird 3 an den Miniserver weitergesendet, ist gekippt ("tilted"), wird 2 weitergeleitet. Mit den Zahlen kann man Logiken abbilden, oder diese mit einem Status-Baustein wieder als Text darstellen lassen.

Diese Konvertierungen gelten für sämtliche Daten, die eingehen (unabhängig vom Topic). Es kann also nur eine Konvertierung pro Text geben. Duplikate werden ausgefiltert und im Log als Warnung angezeigt.

Incoming Overview

Miniserver Virtual Input	Last value	Last submission
loxberry-dev.brunnenweg_lan_mqttgateway_keepaliveepoch	1558891909	26.05.2019 19:31:50
loxberry-dev.brunnenweg_lan_mqttgateway_status	Connected	26.05.2019 19:31:50
loxberry-dev.mqttgateway_keepaliveepoch	1555756105	26.05.2019 19:31:50
loxberry-dev.mqttgateway_status	Connected	26.05.2019 19:31:50

In der "Incoming Overview" werden alle eingehenden Daten der letzten 24 Stunden angezeigt. Diese wurden - sofern das Caching die Daten nicht unterdrückt hat - so an den Miniserver weitergeleitet.

Die Übersicht wird automatisch aktualisiert. Wenn du Text markierst, wird die Aktualisierung unterdrückt, damit du Bezeichnungen mit der Zwischenablage kopieren kannst.

Show advanced table information

shellies_shelly1-5BACB1_input_0
(Topic shellies/shelly1-5BACB1/input/0)

0

26.05.2019 19:31:50

Disable cache Reset after send

Damit werden zusätzliche Schaltflächen und Optionen angezeigt.

- **Topic:** Zeigt bei HTTP die originale Topic-Bezeichnung an, auch bei JSON-extrahierten Daten.
- **Disable Cache:** Dieser Wert übergeht das Caching von LoxBerry, und sendet den Wert immer, auch wenn er gleich bleibt.
- **Reset after send:** Nachdem der Wert gesendet wurde, wird automatisch eine 0 nachgesendet. Das eignet sich für Geräte (z.B. Taster), die lediglich einen Wert (Impuls) senden und dann auf diesem Wert "stehen bleiben". Ein weiterer Impuls würde verursachen, dass wieder der gleiche Wert gesendet würde, und der Miniserver erkennt keine Änderung.
- **Do not forward:** Mit aktivierter Checkbox wird dieser Datensatz nicht an den Miniserver weitergeleitet. Das schont Ressourcen am Miniserver. Die Daten von diesen Topics werden in der Incoming Overview durchgestrichen dargestellt.
- **Lösch-Symbol:** Damit wird der Wert am Broker und in der Incoming Overview gelöscht.

Bekannte MQTT-Geräte und Subscription-Topics

Liebe Plugin-Benutzer! Bitte pflegt diese Tabelle mit euren eingesetzten Geräten, MQTT-Infos und Links!

Die Liste ist jetzt in diesem Unter-Artikel: [MQTT-Gateway - Bekannte MQTT-Geräte und Subscription-Topics](#)

Weitere Hinweise

- Die Deinstallation des MQTT-Gateway Plugins **deinstalliert** den Mosquitto-Broker. Bei der Deinstallation werden die zuletzt verwendeten Mosquitto-Zugangsdaten als Benachrichtigung in die Plugin-Verwaltung gesendet.
- Das Plugin integriert sich in den [LoxBerry Selbsttest / LoxBerry Healthcheck](#). Geprüft wird, ob das MQTT Gateway Service läuft, ob es Meldungen bei der Öffnung der Netzverbindungen gab, und ob die Broker-Verbindung korrekt funktioniert.

P4 LoxBerry: Plugin Management

15.01.2020 08:19: MQTT Gateway uninstalled. Your last broker credentials: USER: "loxberry" PASS: "NY2x2pdSrvdaVejc". Copy this NOW, if you need to recreate broker settings with same credentials!

Installation

Plugin Archive

Roadmap

- Fehlerkorrekturen
- Gateway: Eigenes Topic zur Steuerung des MS
- Integration in LoxBerry-Core

Fragen stellen und Fehler melden

[MQTT Gateway - Troubleshooting Guide](#)

GitHub: <https://github.com/christianTF/LoxBerry-Plugin-MQTT-Gateway/issues>

Forum: <https://www.loxforum.com/forum/projektforen/loxberry/plugins/176025-loxberry-mqtt>

Fehlerbeschreibungen

Für jeglichen Kontakt bzw. Probleme **immer** mitliefern:

- das Logfile
- die MQTT-Spezifikation des angebundenen Devices (z.B. direkter Link zur MQTT-Dokumentation des Geräts)
- Screenshots bzw. Beschreibung der durchgeführten Einstellungen usw.

Siehe auch

- [Einrichtung am MQTT-Device](#)
- [FHEM Geräte per MQTT steuern](#)
- [HTTP-Eingänge am Miniserver debuggen](#)
- [MQTT Gateway - Alle FHEM-Readings weitergeben](#)
- [MQTT-Gateway - Bekannte MQTT-Geräte und Subscription-Topics](#)
- [MQTT Gateway for plugin developers](#)
- [MQTT-Gateway - Generischer POST / GET / JSON Empfänger](#)
- [MQTT Gateway - HTTP- und UDP-Interface](#)
- [MQTT Gateway - Konfiguration sichern und auf anderem/neuen LoxBerry wiederherstellen](#)
- [MQTT Gateway - Subscriptions und Topics](#)
- [MQTT Gateway - Troubleshooting Guide](#)
- [MQTT - Schritt für Schritt: Loxone -> MQTT](#)
- [MQTT - Schritt für Schritt: MQTT -> Loxone](#)
- [Using multiple LoxBerry's with a single MQTT broker](#)
- [Was ist MQTT? Warum MQTT?](#)

- Informationen und Best Practices zu Topics und Subscriptions: <https://www.hivemq.com/blog/mqtt-essentials-part-5-mqtt-topics-best-practices/>
- Miele Home MQTT Gateway: <https://www.loxforum.com/forum/projektforen/loxberry/plugins/176967-miele-mqtt-loxberry-mqtt>
- Verschiedene MQTT-Interfaces von Devices und Diensten: <https://github.com/hobbyquaker/awesome-mqtt#interfaces>
- Fronius Hybrid und *go-eCharger* Bridge: <https://github.com/akleber/mqtt-connectors>