

Integrazione MQTT

MQTT è un protocollo di messaggistica open standard per l'Internet of Things (IoT). È progettato come un sistema di trasporto **publish/subscribe** estremamente leggero, ideale per connettere dispositivi remoti con un footprint di codice ridotto e un minimo utilizzo della banda di rete (fonte: <https://mqtt.org>).

MQTT significa Message Queuing Telemetry Transport.

Gli elementi fondamentali di un sistema MQTT sono **client** e **server** (noti anche come **broker**).

I client si sottoscrivono a **topic** per pubblicare e ricevere messaggi. Ciò significa che un client può svolgere sia il ruolo di **sottoscrittore** che quello di **publisher**.

I server gestiscono i topic: ricevono le **sottoscrizioni** dai client, ricevono i messaggi dai client e li inoltrano agli altri client interessati in base alle loro sottoscrizioni.

Un server Thinknx può essere configurato come client MQTT, collegandosi a un broker/server esterno e sottoscrivendo uno o più topic, oppure come server MQTT, gestendo direttamente sottoscrizioni e distribuzione dei topic.

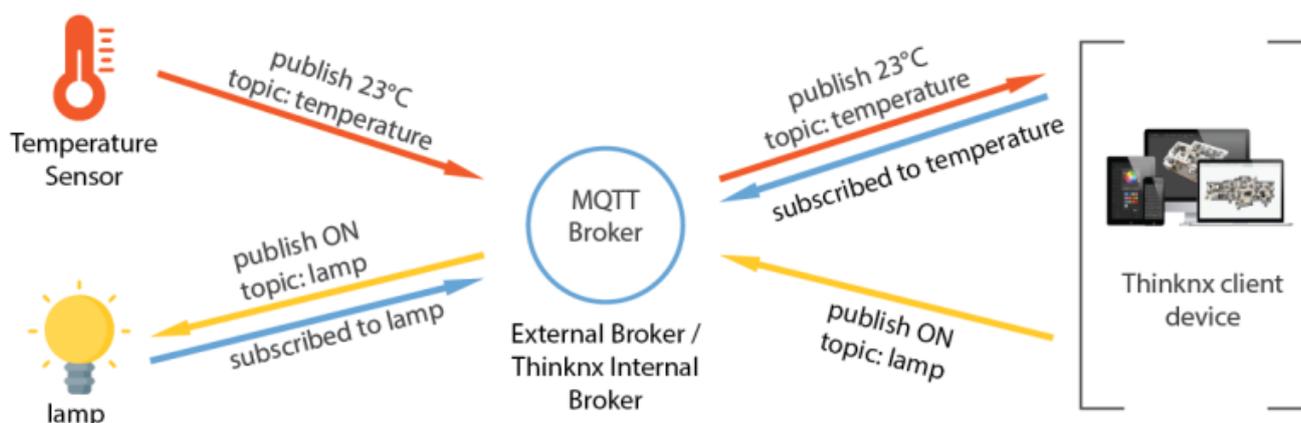


Figura: Panoramica MQTT

Client MQTT

Permette la configurazione del dispositivo Thinknx come client MQTT verso un broker esterno o verso quello interno. Per gestire messaggi MQTT, Thinknx dev'essere configurato come client, consentendo di leggere (input virtuale) e scrivere (output virtuale) valori associati a un topic, da usare nell'interfaccia utente o in altri oggetti del sistema. Per applicazioni più avanzate, i dati MQTT possono essere elaborati tramite un Data Format Editor. Thinknx supporta comunicazioni non criptate o criptate tramite TLS. La comunicazione criptata potrebbe non funzionare correttamente su modelli più vecchi di server Thinknx o con alcuni broker esterni.

- **Local Broker** Se True, utilizza il server MQTT interno. Se False, richiede la connessione a un broker MQTT esterno.
- **IP Port** Porta del broker MQTT. Porta predefinita: 1883.
- **IP Address/Hostname** Indirizzo IP del broker MQTT.
- **Use Credentials** Se True, usa username/password per la comunicazione client-server.
- **Username** Nome utente per la connessione MQTT.
- **Password** Password per la connessione MQTT.
- **Secure TLS** Se True, usa il protocollo crittografico TLS.
- **Client Certificate** Percorso del certificato digitale (*.crt).
- **Client Key** Percorso della chiave client (*.key).
- **Virtual Inputs** Subscribers MQTT.
- **Virtual Outputs** Publishers MQTT.

Input Virtuali (Subscribers)

Permette di sottoscrivere un topic e ottenere valori aggiornati associati ad esso. Il valore ricevuto può aggiornare l'interfaccia utente, essere utilizzato dal sistema o trasmesso al bus KNX.

- **Topic** Descrizione del topic da sottoscrivere.
- **Use Credentials** Se True, usa credenziali specifiche per la sottoscrizione.
- **Username** Username per autenticarsi.
- **Password** Password per autenticarsi.
- **Data Format Input** Editor per personalizzare il formato dei dati in ingresso.
- **Send Trigger** Specifica l'azione che avvia la comunicazione con il servizio esterno.
- **Use KNX gateway** Se abilitato, invia dati al bus KNX.
- **KNX value feedback group** Gruppo KNX a cui inviare il valore.
- **KNX datatype** Tipo di dato KNX da usare per l'invio.

Data Format Editor

Permette di configurare il formato dei dati in ingresso e la risposta.

1. Seleziona il tipo di formato dati ricevuto (XML, Stringa o JSON).
2. Configura il percorso corretto della risposta.
3. Converte il valore ricevuto nel formato corretto (Bit, Int, Double, Enum, String).

Output Virtuali (Publishers)

Permette di pubblicare valori dal server Thinknx verso un topic MQTT, valori che possono essere utilizzati da altri client MQTT. I valori possono provenire da gruppi KNX specifici.

- **Topic** Descrizione del topic da pubblicare.
- **Send Trigger** Specifica l'azione che avvia la comunicazione.
- **Data Format Input** Editor per personalizzare il formato dei dati inviati.
- **Use KNX gateway** Se abilitato, inoltra dati ricevuti al bus KNX.
- **KNX value group** Gruppo KNX che invoca il publisher.
- **KNX datatype** Tipo di dato KNX per l'invio.

Data Format Editor

Permette di configurare il formato dei dati inviati dal publisher.

1. Seleziona il tipo di dati inviati (XML, Stringa, Parameters, JSON).
2. Converte il valore inviato nel formato corretto (Bit, Int, Double, Enum, String).

Esempio 1: Thinknx Server come Client MQTT

In questo esempio viene usato un sensore di temperatura e umidità Shelly H&T che pubblica dati su un server MQTT esterno come Mosquitto, CloudMQTT o HiveMQ. Il server Thinknx, configurato come client MQTT, si sottoscrive ai topic temperatura e umidità e riceve i valori dal broker MQTT esterno.

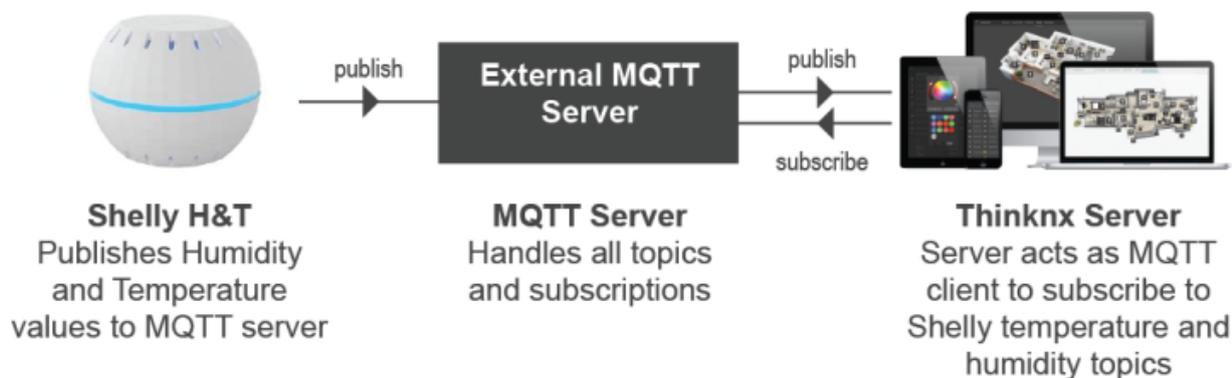
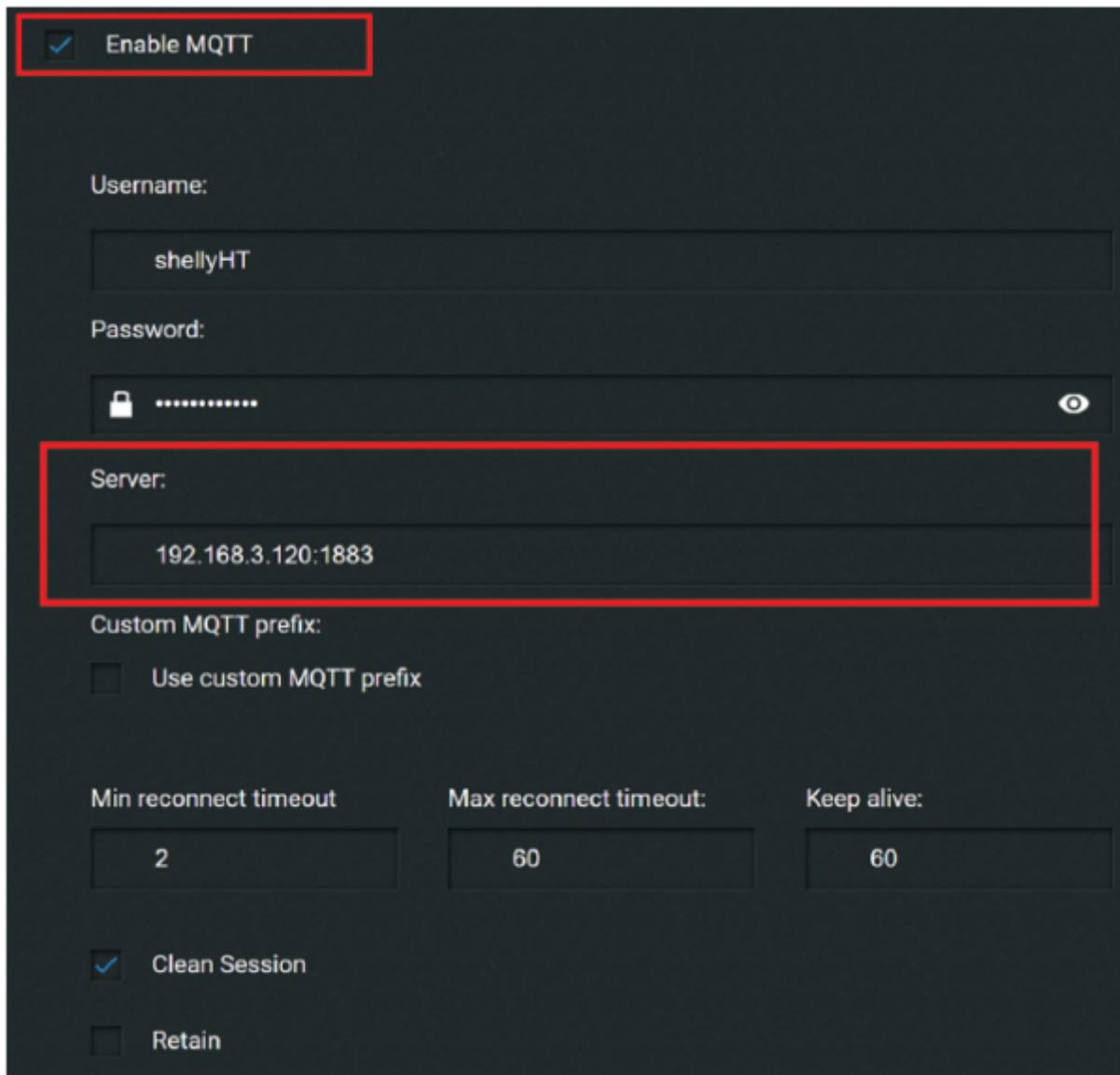


Figura: Thinknx server come client MQTT

Configurazione Shelly H&T

- Connettiti all'hotspot Shelly e accedi all'IP predefinito 192.168.33.1.
- Attiva il Client Mode WiFi e inserisci le credenziali di rete WiFi.
- Vai in Advanced Developer Settings e abilita "Enable action execution via MQTT". Inserisci l'indirizzo IP del broker MQTT e la porta (predefinita: 1883).
- Inserisci eventuali credenziali di accesso al broker MQTT.



Enable MQTT

Username:
shellyHT

Password:
.....

Server:
192.168.3.120:1883

Custom MQTT prefix:
 Use custom MQTT prefix

Min reconnect timeout: 2 Max reconnect timeout: 60 Keep alive: 60

Clean Session
 Retain

Figura: Configurazione Shelly H&T

Configurazione Thinknx Configurator

 Per visualizzare i topic Shelly, visita:
<https://shelly-api-docs.shelly.cloud/gen1/#shelly-h-amp-t>

- Vai in Sistema e aggiungi l'oggetto MQTT client.
- Imposta Local Broker su False, inserisci le informazioni del broker MQTT esterno.
- Aggiungi un Virtual Input per la temperatura. Topic: **shellies/shellyht-
<deviceid>/sensor/temperature**.
- Apri Data Format Input, scegli "Stringa", convertila in "Double".
- Aggiungi un Valore Analogico, scegli "Oggetto di sistema", seleziona l'input virtuale creato.

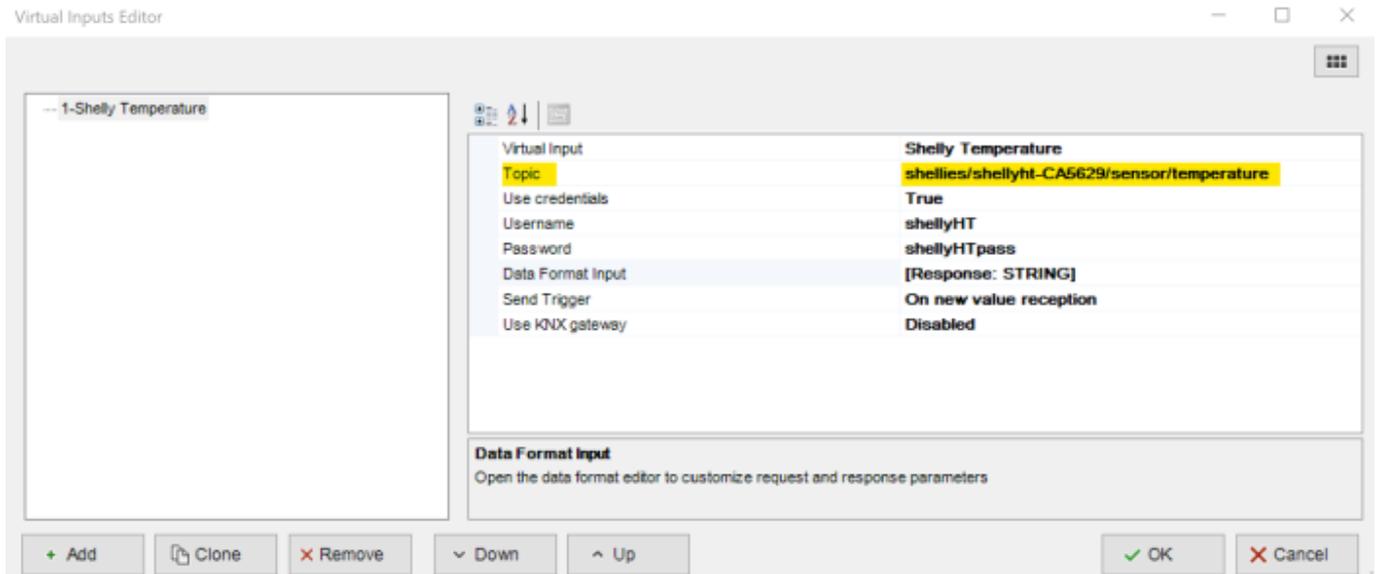
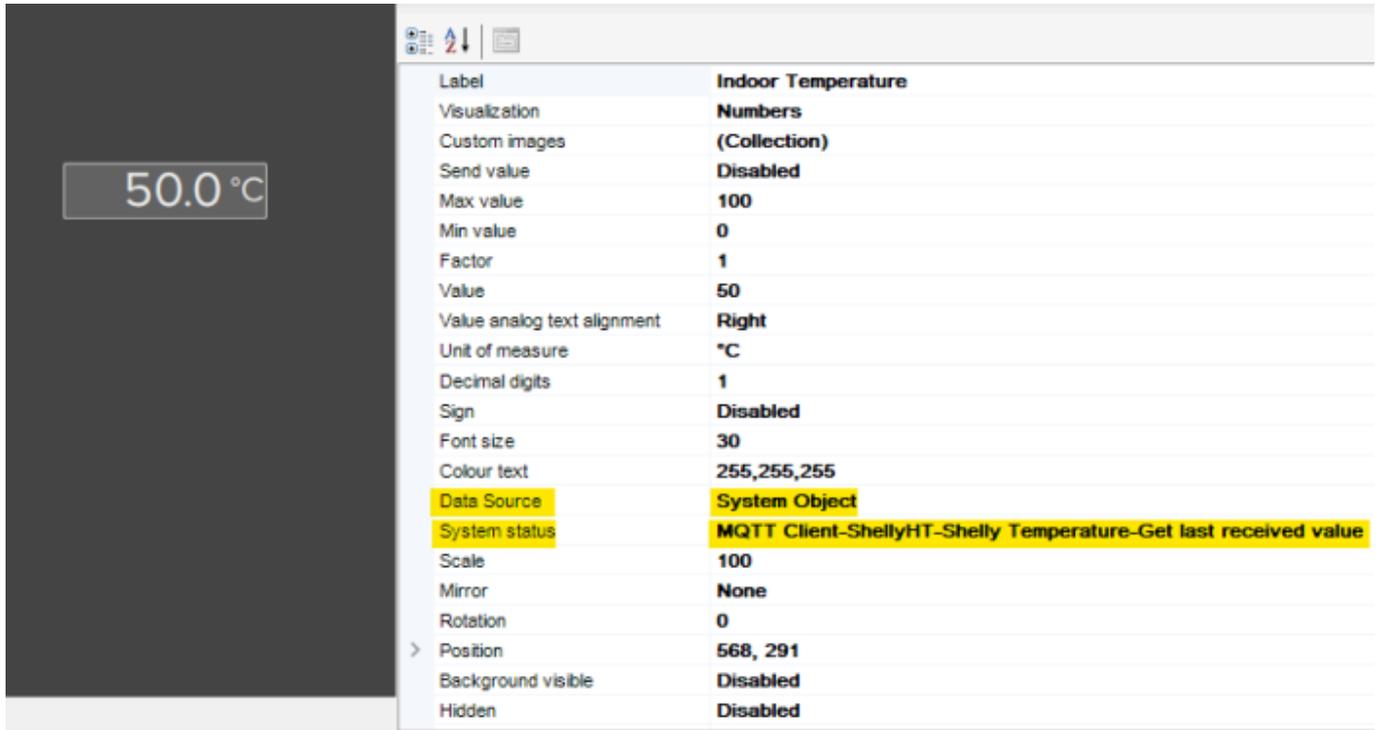


Figura: Input Virtuale MQTT Client



Figura: Data Input Format



The image shows a configuration interface for a temperature widget. On the left is a preview of the widget displaying '50.0 °C'. On the right is a configuration table with the following settings:

Label	Indoor Temperature
Visualization	Numbers
Custom images	(Collection)
Send value	Disabled
Max value	100
Min value	0
Factor	1
Value	50
Value analog text alignment	Right
Unit of measure	°C
Decimal digits	1
Sign	Disabled
Font size	30
Colour text	255,255,255
Data Source	System Object
System status	MQTT Client-ShellyHT-Shelly Temperature-Get last received value
Scale	100
Mirror	None
Rotation	0
Position	568, 291
Background visible	Disabled
Hidden	Disabled

Figura: Visualizzazione temperatura Shelly

From: <http://www.thinknx.com/wiki/> - **Learning Thinknx**

Permanent link: <http://www.thinknx.com/wiki/doku.php?id=prova4>

Last update: **2025/02/24 15:29**

