

# MQTT Integration

MQTT is an open standard messaging protocol for the Internet of Things (IoT). It is designed as an extremely lightweight **publish/subscribe** messaging transport that is ideal for connecting remote devices with a small code footprint and minimal network bandwidth (source: <https://mqtt.org>).

MQTT stands for Message Queuing Telemetry Transport.

The core elements of an MQTT system are **clients** and **servers** (also known as **brokers**).

Clients subscribe to **topics** to publish and receive messages. This means that a client handles the roles of **subscriber** and **publisher**.

Servers run topics, for example receive **subscriptions** from clients on topics, or messages from clients and forward them based on client's subscriptions, to interested clients.

A Thinknx server can be configured to work as an MQTT client to connect to an external broker/server and subscribe to one or multiple topics, or as an MQTT server itself to handle subscriptions and topics distribution.

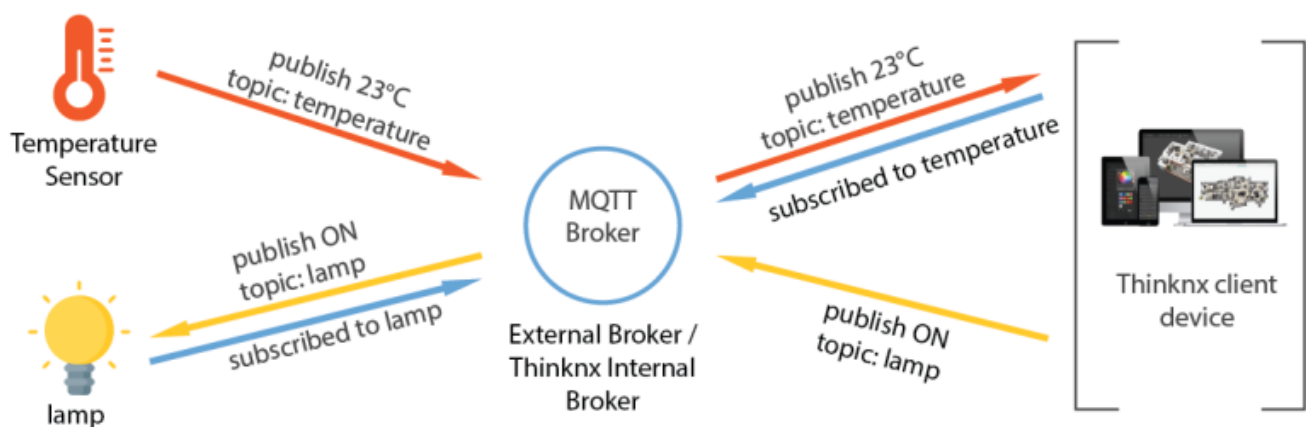


Figure: MQTT Overview

## MQTT Client

Permits the configuration of the Thinknx device as MQTT client either to an external broker or to the internal one. To handle messages from MQTT Thinknx needs to be configured as client. This permits to read (as virtual input) and write (as virtual outputs) values associated to a topic and to use them inside the UI or other system objects. To permits a wide application range, values coming or going to MQTT can be pre/post processed in a Data Format Editor. Thinknx supports plain or encrypted (TLS) communication with brokers. The encrypted communication may not work properly on all the circumstances depending on the server models (very old Thinknx servers may have some difficulties) and the capabilities of the external brokers.

- **Local Broker** If True, the internal MQTT server is used. If False, a connection to an external MQTT server is required.
- **IP Port** IP port used by the broker. Default port for MQTT communication is 1883.
- **IP Address/Hostname** IP address of the MQTT broker.
- **Use Credentials** If true, the communication between client and server requires username and password.
- **Username** Username required to connect to MQTT server.
- **Password** Password required to connect to MQTT server.
- **Secure TLS** if True, TLS cryptographic protocol is required during communication.
- **Client Certificate** Path to select for the Digital Certificate (\*.crt) used by clients to make authenticated requests to the server.
- **Client Key** Path to select for the Client Key (\*.key extension).
- **Virtual Inputs** MQTT subscribers
- **Virtual Outputs** MQTT publishers

## Virtual Inputs (Subscribers)

This section permits to subscribe for a topic and to get the updated values associated with it. The value can then be used to update UI, for the internal system objects or can be transmitted to KNX.

- **Topic** Description of the topic to be hooked to.
- **Use Credentials** if True, it permits to use specific credentials to authenticate with the service and get the value of the input.
- **Username** Username used to authenticate with the service.
- **Password** Password used to authenticate with the service.
- **Data Format Input** Opens the data format editor to customize the request and response parameters.
- **Send Trigger** Specify which action will trigger communication with the external service.
- **Use KNX gateway** If enabled, it will forward the data from the external service to the KNX bus.
- **KNX value feedback group** KNX group to use to send the value coming from the external service.
- **KNX datatype** KNX datatype to use to transmit the value on the KNX bus.

## Data Format Editor

This window permits to configure the format of the Incoming Data and Response.

1. First, select the incoming data format type. The data received can be XML, String, or JSON.
  2. Configure the correct response path.
1. Cast the value received to the appropriate representation (Bit - Int - Double - Enum - String).

## Virtual Outputs (Publishers)

This section permits to publish values coming from the server to a specific topic. The value can then be used by other MQTT client that will react to them. The value for these topics can come from specific KNX group addresses.

- **Topic** Description of the topic to be hooked to.
- **Send Trigger** Specify which action will trigger communication with the external service.
- **Data Format Input** Opens the data format editor to customize the format of the sent data.
- **Use KNX gateway** If enabled, it will forward the data from the external service to the KNX bus.
- **KNX value group** KNX group address that invokes the publisher.
- **KNX datatype** KNX datatype.

## Data Format Editor

This window permits to configure the format of the sent data by the publisher.

1. First, select the type of the Sent Data. The data sent can be XML, String, Parameters, or JSON.
1. Cast the value sent to the appropriate representation (Bit - Int - Double - Enum - String).

## Example 1: Thinknx Server as MQTT Client

A Shelly H&T humidity and temperature sensor is used in this example. This device will publish its data to an external MQTT server such as Mosquitto (<https://mosquitto.org/>), CloudMQTT (<https://www.cloudmqtt.com/>) or HiveMQ (<https://www.hivemq.com/>). The Thinknx server acting as an MQTT client, will subscribe to the temperature and humidity topics, and the external MQTT server will forward the values to the server.

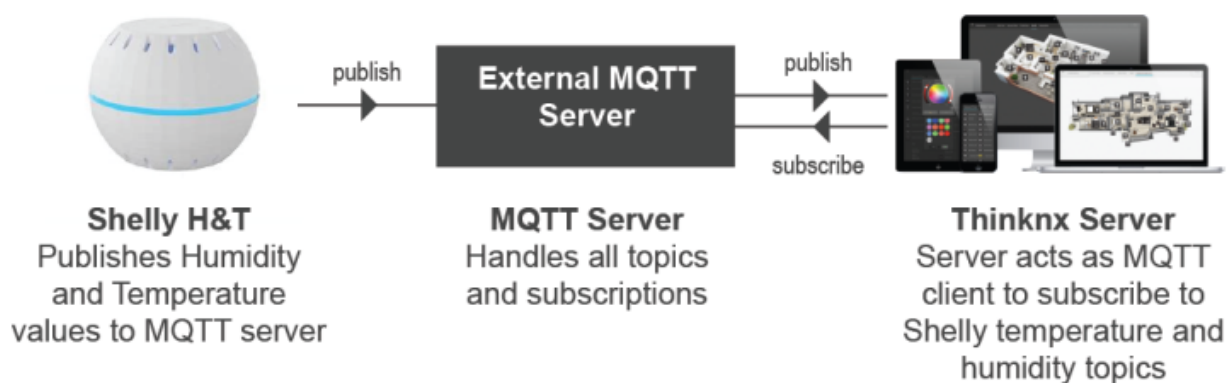
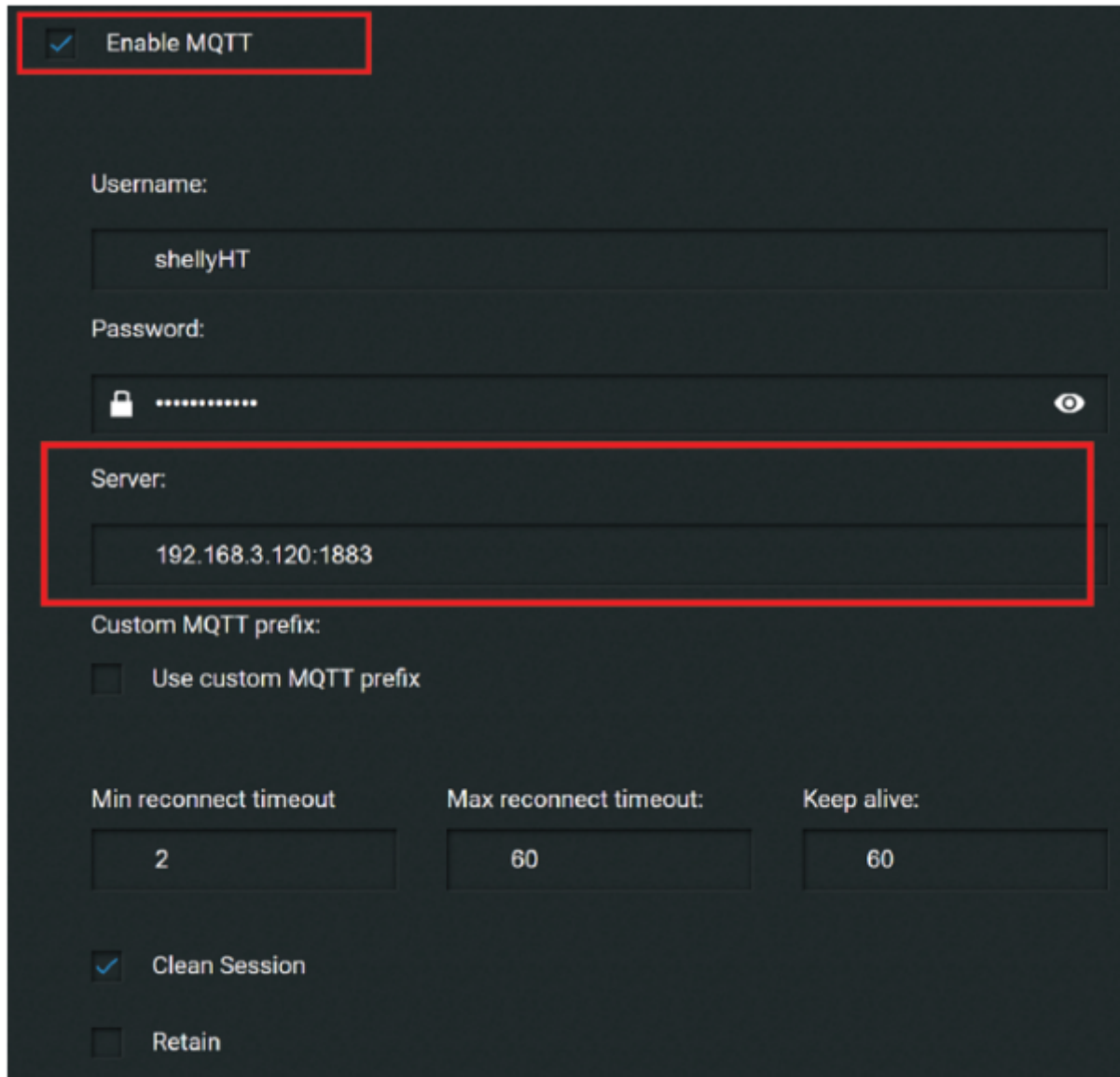


Figure: Thinknx server as MQTT client

## Configuration on Shelly H&T

- Connect to the Shelly device's hotspot and access its web interface by typing its default IP address 192.168.33.1.
- Go to Internet and Security Menu and enable the Wifi - Client Mode on your device and enter your Wifi credentials to include it in your network.
- Go into Advanced Developer Settings and check the box "Enable action execution via MQTT". Then, enter the IP address of your MQTT server and the proper port (default 1883).
- Enter credentials to connect to the broker if required by the broker.



☒ Enable MQTT

Username:  
shellyHT

Password:  
[masked]

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

Figure: Configuration Shelly H&amp;T

## Configuration in Thinknx configurator



To view all topics available for Shelly devices, visit their API documentation on this link: <https://shelly-api-docs.shelly.cloud/gen1/#shelly-h-amp-t>

- Go to System and add the object MQTT client.
- Set “Local Broker” as false, and enter the information of the external broker (IP address/port/credentials if applicable).
- Open the Virtual Input window, and add a new subscriber to read the temperature.
- Consult the Shelly API documentation link above to retrieve the correct topic for the Shelly H&T Temperature. According to the documentation, the Shelly device publishes the temperature on the following topic: **shellies/shellyht-<deviceid>/sensor/temperature**.
- Enter the proper credentials to connect if applicable.
- Open the “Data Format Input” window to confirm the response type as “string”, and cast it into a “double” and visualize it with the analog input.

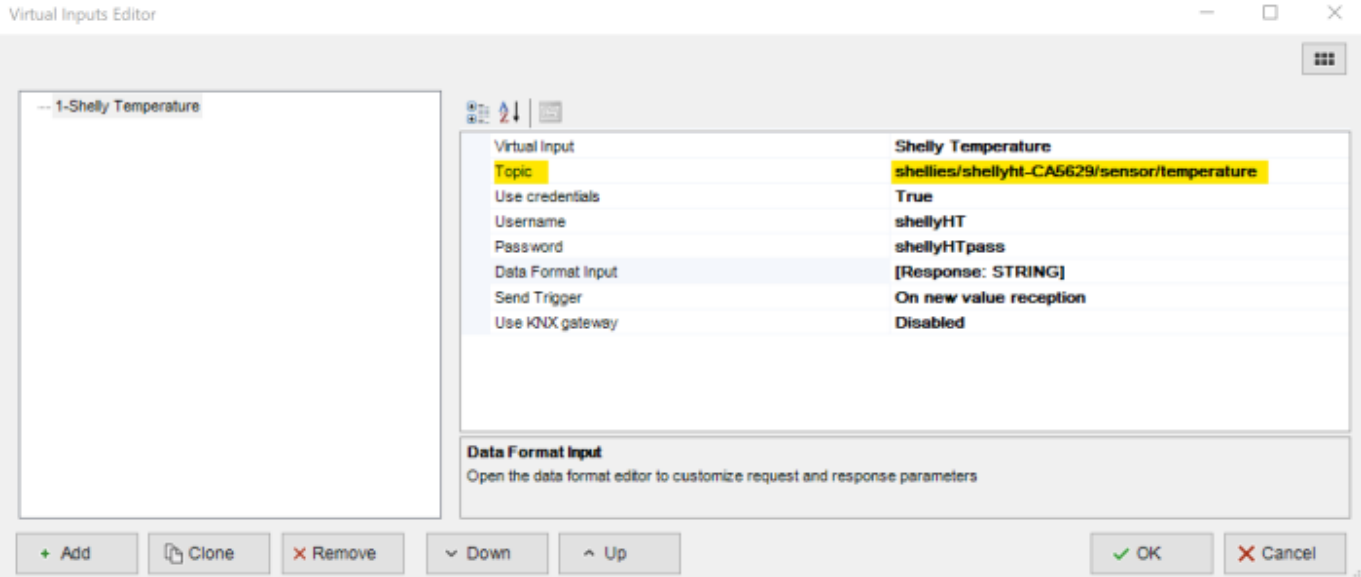


Figure: Virtual Input MQTT Client

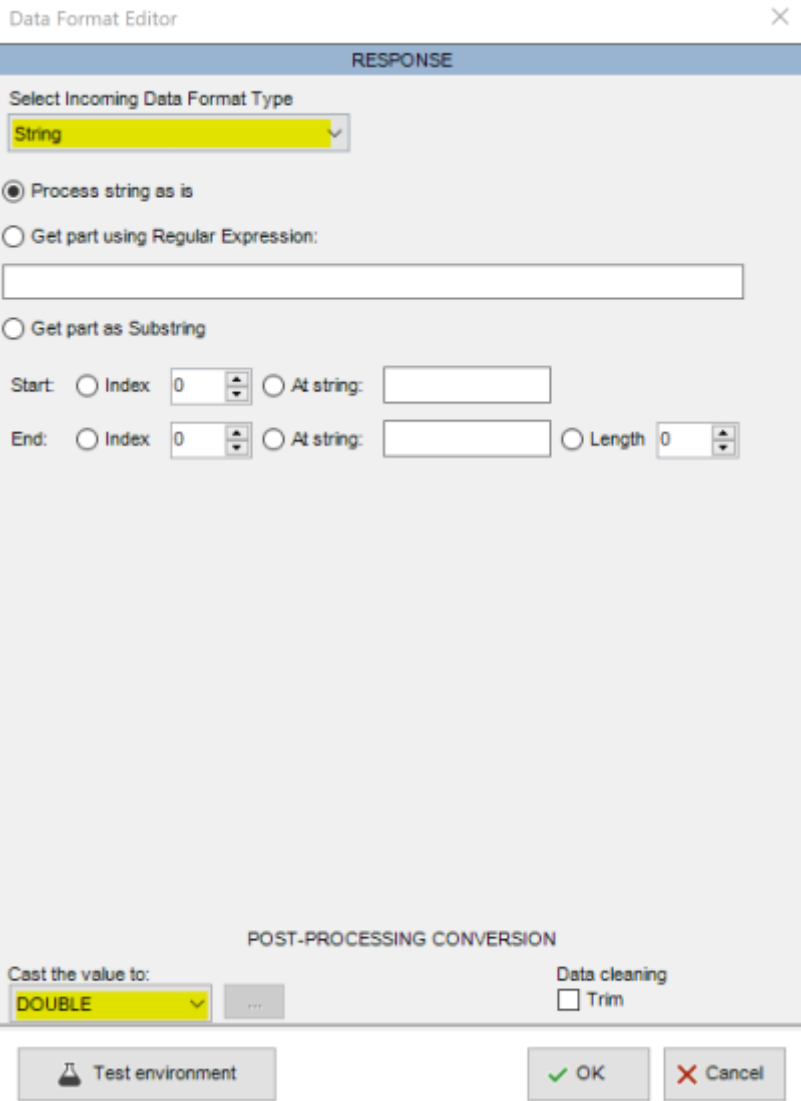
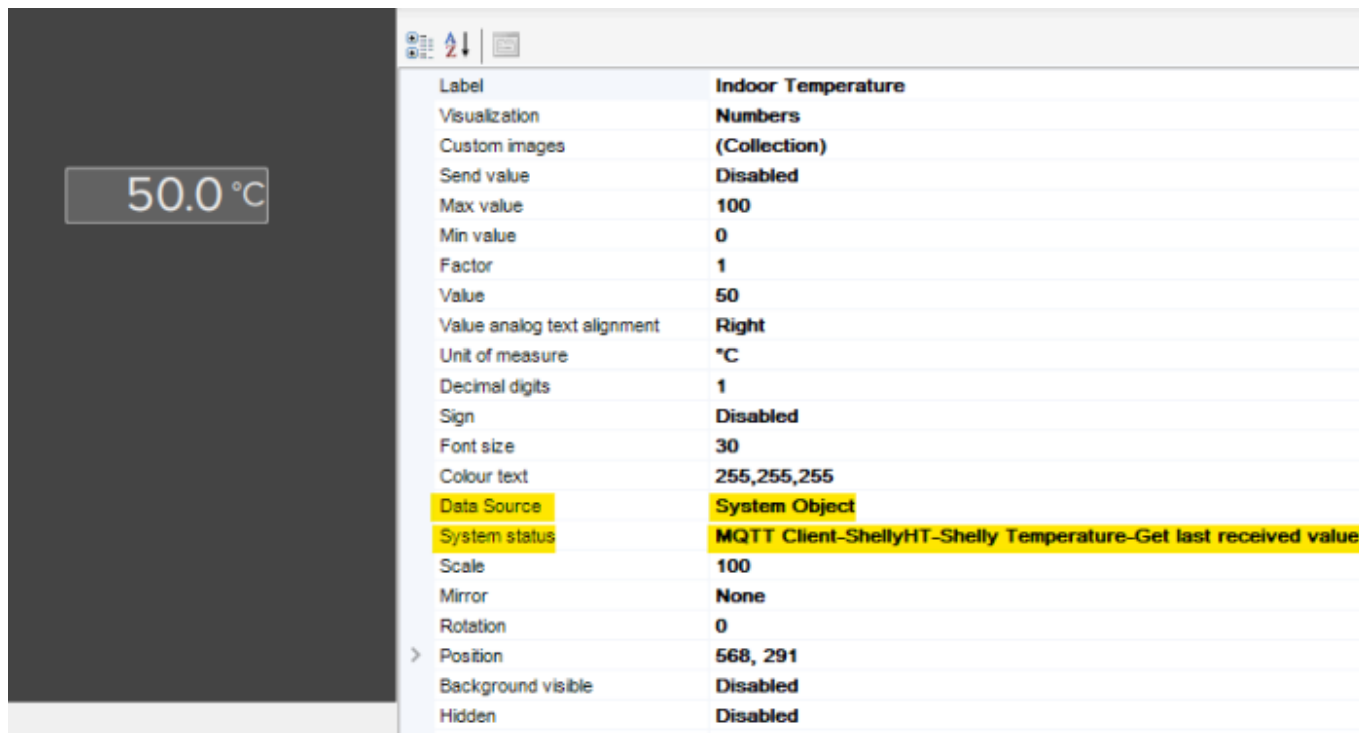


Figure: Data Input Format

To visualize the temperature value received by the Shelly device on a specific page:

- Add an Analog Value on the page.
- Make sure to configure the data source as “System object”.
- Under System Status, go to the virtual input created for the temperature and select “Get last received value”.



The screenshot displays a dashboard with a dark grey background. On the left, a white rectangular box shows the temperature '50.0 °C'. To the right, a configuration panel lists various settings for the visualization. The 'Data Source' is set to 'System Object' and the 'System status' is set to 'MQTT Client-ShellyHT-Shelly Temperature-Get last received value'. Other settings include 'Label' as 'Indoor Temperature', 'Visualization' as 'Numbers', 'Unit of measure' as '°C', and 'Font size' as '30'.

Property	Value
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

Figure: Visualizing Shelly Temperature

From:

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

Permanent link:

<https://www.thinknx.com/wiki/doku.php?id=mqtt>

Last update: **2022/08/23 08:58**

