

Festo AX Energy Insights Node-RED: energy data from Sentron/MA770 with Modbus



GASA-ENI-CTR-*

This document describes how to get electric data from the Siemens Sentron PAC 2200/3200/4200 with the Modbus protocol.

Title Festo AX Energy Insights Node-RED: energy data from Sentron/MA770 with Modbus
Version 1.20
Document no. 100793
Originalen
Author Festo SE & Co. KG

Last saved 27.07.2026

General information:

This document is intended for qualified, trained and instructed professionals. The data provided in this document are no guaranteed specifications, in particular with regard to functionality, condition or quality in the legal sense. The information in this document is intended only as simple indications for the implementation of a specific, hypothetical application, and in no way as a substitute for the operating instructions of the respective manufacturers or the design and testing of the respective application by the user. The respective operating instructions for Festo products can be found under www.festo.com. The user of this document must ensure that each function described herein works properly in his application. The user remains solely responsible for his or her own use of this document, even by studying this document and using the information mentioned therein. This also applies to any software (in source code and/or object code) that is made available to the user as an appendix to this document.

Rights of use of the software:

If software is made available to the user as an appendix to this document, the user is granted a simple and unlimited right of use hereto. This right also includes the processing and distribution of the software to third parties in edited or unedited form. The User is not permitted to use the name "Festo" to endorse or promote products derived from the Software without express prior written permission.

Software is provided to the user "as is". Festo does not assume any warranty or guarantee with regard to software. Festo's liability for damages of any kind arising from the use of the software is limited to intent and gross negligence.

Legal Notices:

©Festo SE & Co. KG, all rights reserved. A change in content and form is only permitted with the express written consent of Festo SE & Co. KG. Festo grants the user the right to reproduce this document in a form that remains unchanged in terms of content and form and to pass it on to third parties.



Need **support** with your **application**?
Get in touch with your local **Technical Support**:
www.festo.com/contact

Table of contents

1	Overview	5
1.1	Software and hardware used.....	5
1.2	Topology	5
1.3	Recommended manuals	5
2	Integration of Node-RED in Docker environment	6
2.1	Add Node-RED to the standard docker-compose.yml.....	6
3	Set up the Node-RED	7
3.1	Add palette to Node-RED.....	7
3.2	Import the Node-RED workflow for energy meter	9
4	Node-RED workflow for Siemens Sentron PAC 2200/3200/4200 energy meter	11
4.1	Introduction to the Energy Insights electric Node-RED workflow.....	11
4.1.1	Modbus Flex Getter node	11
4.1.2	Configure subflow “AX ENI el - Modbus TCP to MQTT - Sentron PAC2200/3200/4200”	12
5	Node-RED workflow for Phoenix Contact energy meter EEM MA700	14
5.1	Introduction to the Energy Insights electric Node-RED workflow.....	14
5.1.1	Modbus Flex Getter node	14
5.1.2	Configure subflow “AX ENI el - Modbus TCP to MQTT – Phoenix Contact EEM MA770”	15
6	MQTT connection to Festo AX broker	17

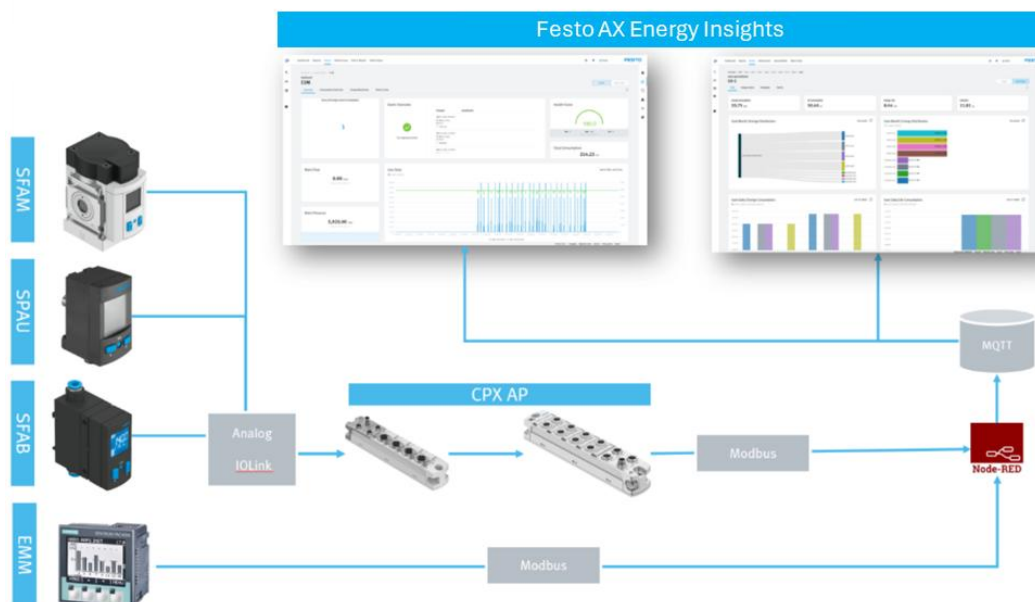
1 Overview

1.1 Software and hardware used

Type / name	Version software / firmware
Node-RED docker image	4.1.0
Siemens Sentron PAC 2200/3200/4200	-
Phoenix Contact MA770	-

Table 1.1: Software and hardware used

1.2 Topology



1.3 Recommended manuals

Title	Author	URL
“node-red-contrib-modbus”	Klaus Landsdorf and Community Driven	node-red-contrib-modbus (node) - Node-RED
“Power Monitoring Device Sentron PAC 2200”	Siemens	Power Monitoring Device Sentron PAC 2200
“Power Monitoring Device Sentron PAC 3200”	Siemens	Power Monitoring Device Sentron PAC3200
“Power Monitoring Device Sentron PAC 4200”	Siemens	Power Monitoring Device SENTRON PAC4200
“EEM-MA770”	Phoenix Contact	Phoenix Contact EEM MA770

Table 1.2: Manuals

2 Integration of Node-RED in Docker environment

2.1 Add Node-RED to the standard docker-compose.yml

Node-RED is not included in the delivered version of the standard docker-compose.yml.

To add Node-RED to the standard docker-compose.yml, follow these steps:

1. **Stop and remove the current containers**

Run the following command in the directory containing your docker-compose.yml file to stop and remove existing containers:

```
docker compose down
```

2. **Add Node-RED to your docker-compose.yml file**

Insert the following snippet under the services section in your docker-compose.yml file.

This will add a Node-RED container that restarts automatically unless stopped manually and persists data using a Docker volume.

```
# add a nodered service
nodered:
  image: nodered/node-red:latest
  container_name: nodered
  restart: unless-stopped
  logging: *default-logging
  ports:
    - '1880:1880'
  volumes:
    - noderedaedata:/data
  networks:
    - network_intern
    - network_extern
```

At the bottom of your docker-compose.yml, add the volume definition to persist Node-RED data:

```
# add the nodered volume in the volume section:
volumes:
  noderedaedata:
```

3. **Install and start the containers**

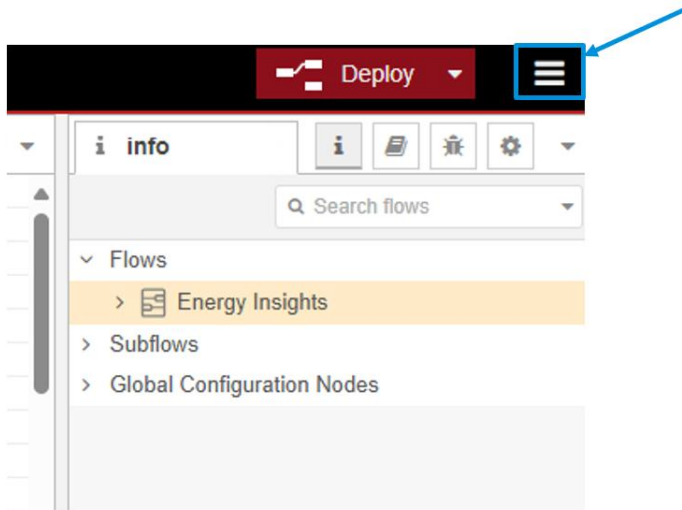
After saving your changes, start the containers again by running:

```
docker compose up
```

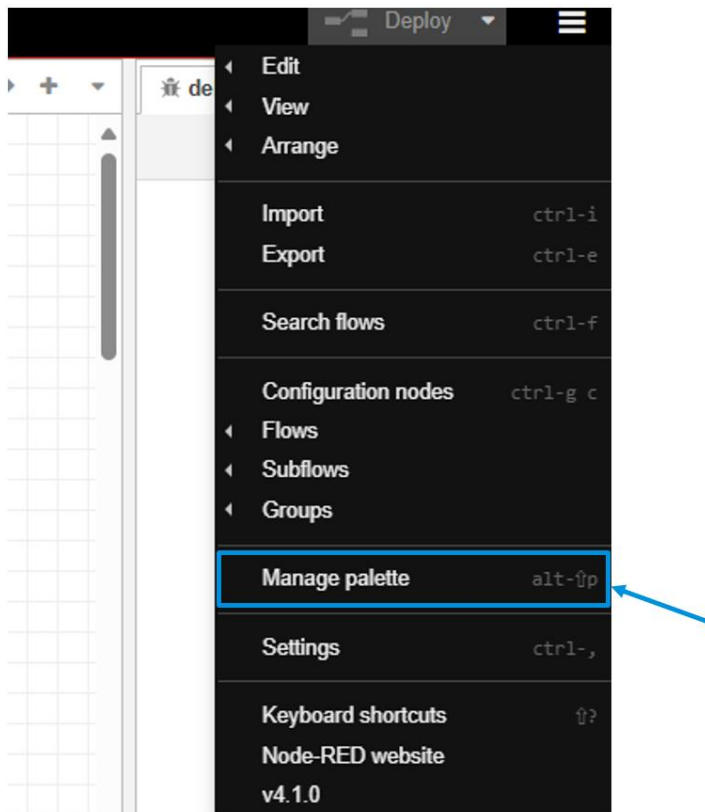
3 Set up the Node-RED

3.1 Add palette to Node-RED

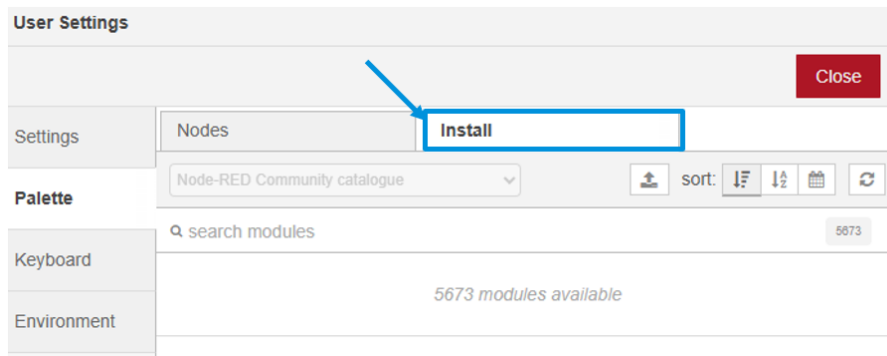
1. Open the Node-RED web interface by navigating to <http://localhost:1880> (or the appropriate address if running remotely).
2. Click on the menu icon (three horizontal lines) in the top right corner.



3. Select Manage palette from the dropdown menu.



4. In the palette manager window, go to the Install tab.



5. To enable communication via the Modbus protocol, install the following palette by searching for their name and clicking Install:

node-red-contrib-modbus

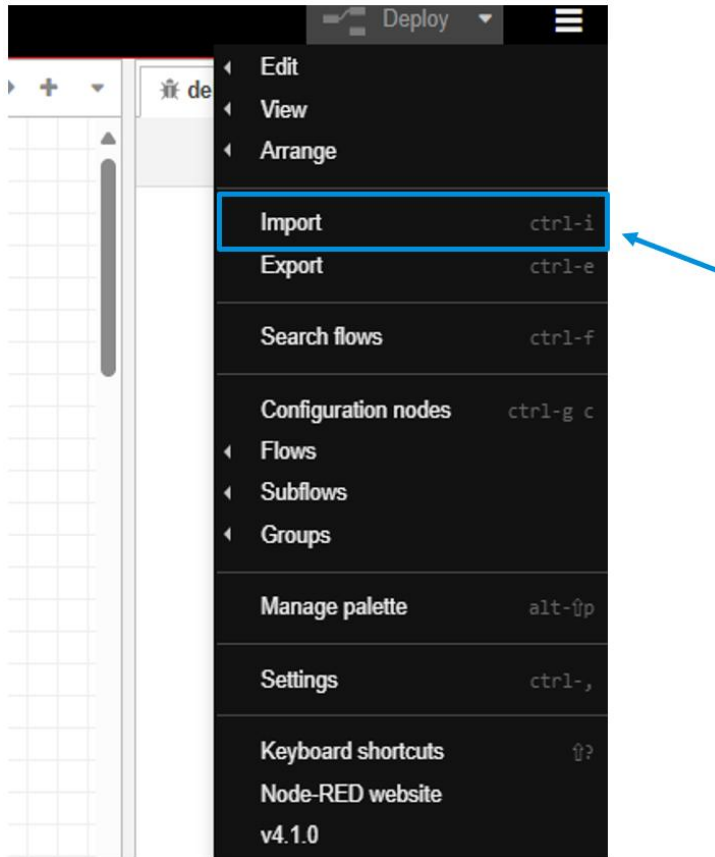


Note

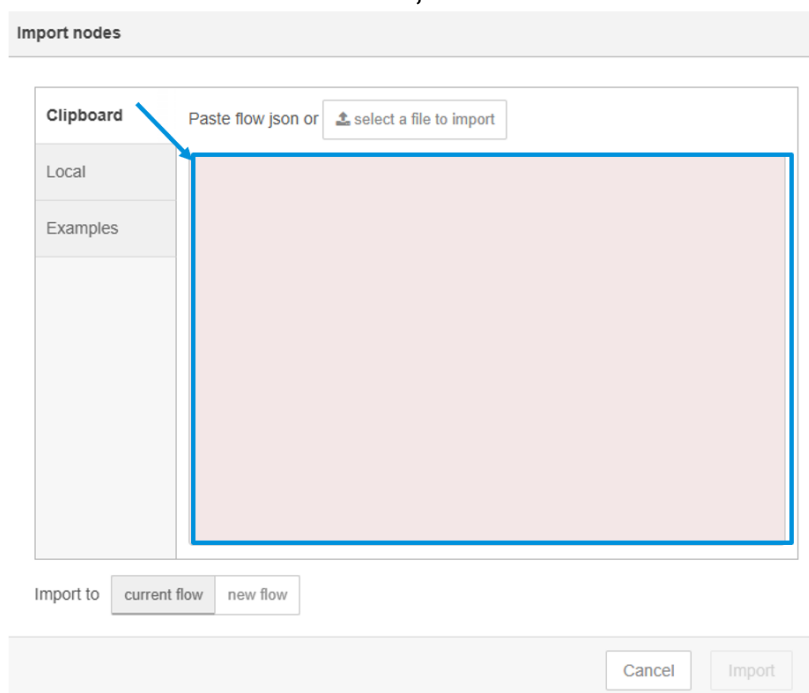
- An active internet connection is required to install Node-RED palettes.

3.2 Import the Node-RED workflow for energy meter

1. Get the .json flow file for specific Energy Meter from the folder.
2. Navigate to <http://localhost:1880> (or the appropriate address if running remotely) in your web browser.
3. Click on menu icon (three horizontal lines) in the top right corner and select Import.



4. In the Import dialog, either:
 - a. Paste the entire content of the *.json file into the text area.



Set up the Node-RED

- b. Use the 'select a file to import' option to upload the *flow.json file.

Import nodes

Clipboard

Paste flow json or [select a file to import](#)

Local

Examples

Import to current flow new flow

Cancel Import

5. Click Import to add the flow to your workspace.

Import nodes

Clipboard

Paste flow json or [select a file to import](#)

Local

Examples

```
[
  {
    "id": "27580427044ce8c5",
    "type": "group",
    "z": "f67f981d2d05372d",
    "style": {
      "stroke": "#999999",
      "stroke-opacity": "1",
      "fill": "none",
      "fill-opacity": "1",
      "label": true,
      "label-position": "nw",
      "color": "#a4a4a4"
    },
    "nodes": [
      "6a96cb006d5be5b3",
      "8474be0c54b35bfc",
      "5aad3bd1e23335a3",
      "022cf0f2c495cb90",
      "022cf0f2c495cb90"
    ]
  }
]
```

Import to current flow new flow

Cancel Import

4 Node-RED workflow for Siemens Sentron PAC 2200/3200/4200 energy meter

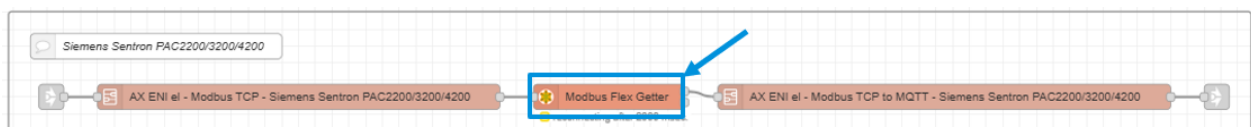
4.1 Introduction to the Energy Insights electric Node-RED workflow

The ENI Electric Siemens Sentron PAC Node-RED flow reads data via the Modbus protocol from the Siemens Sentron PAC 2200/3200/4200 energy meter. It collects essential electrical parameters including voltage, current, power measurements, and energy meter data. Each data point includes a timestamp (ts) indicating when the measurement was taken, and a topic that specifies the data path within the system for proper routing and identification.

For each sensor there is one line in the flow composed of 3 main nodes: a subflow to define the registers to read out, the Modbus TCP client node (Modbus Flex Getter) and a subflow to transform the read data into the required MQTT format. This line needs to be copied and configured for each sensor that needs to be connected.

4.1.1 Modbus Flex Getter node

Adjust the Modbus Flex Getter node.



1. Adjust the Server in the Modbus-Flex-Getter.

The screenshot shows the 'Edit Modbus-Flex-Getter node' dialog box. At the top, there are buttons for 'Delete', 'Cancel', and 'Done'. Below these is a 'Properties' section with tabs for 'Settings' and 'Optionals'. The 'Settings' tab is active. It contains a 'Name' field with the placeholder text 'Name'. Below that is a 'Server' field with the value 'modbus-tcp@192.168.0.30:502'. To the right of the 'Server' field is a blue square button with a pencil icon, which is highlighted by a blue arrow. There is also a '+' button to the right of the pencil icon. Below the 'Server' field, there are two checkboxes: 'Delay to activate input' and 'Allow oversized messages', both of which are currently unchecked.

Edit Modbus-Flex-Getter node > Edit modbus-client node

Delete Cancel Update

Properties

Settings Queues Optionals

Name

Type

Host

Port

TCP Type

Unit-Id

Timeout (ms)

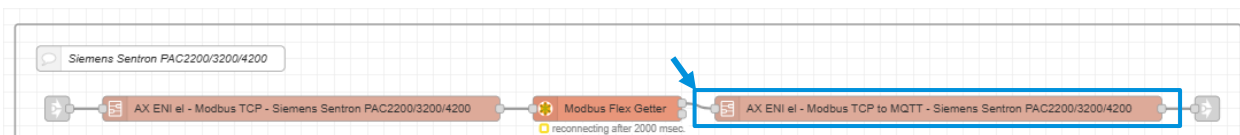
☒ Reconnect on timeout

Reconnect timeout (ms)

- Set Type to TCP.
- Set the Host to the IP address from your master module CPX-AP-I-EP.
- Enter the Port number (default is 502).

4.1.2 Configure subflow “AX ENI el - Modbus TCP to MQTT - Sentron PAC2200/3200/4200”

Each sensor or measurement unit uses a distinct topic and corresponds to a unique asset in Festo AX. This separation ensures clear identification and organization of data streams, enabling precise monitoring and analysis of each measurement point.



- Define if you want to use the consumption counter of the device or just the power values in ENI.
- If you want to suppress negative values, set the option to true. If you don't want to suppress negative values, set it to false.
- Set the asset name to a meaningful name that identifies the sensor or measurement point.
- Update the asset path according to your system's hierarchy, using the 9 different levels.

Edit subflow instance: AX ENI el - Modbus TCP to MQTT - Siemens Sentron PAC2200/3200/4200

Edit subflow template Delete Cancel Done

Properties

Name

Use consumption counter false

Negative power/energy value supression true

Asset name a_z electric sensor

Asset path - Level 1 a_z Country

Asset path - Level 2 a_z Site

Asset path - Level 3 a_z Hall

Asset path - Level 4 a_z Floor

Asset path - Level 5 a_z Segment

Asset path - Level 6 a_z Workgroup

Asset path - Level 7 a_z Line

Asset path - Level 8 a_z Station

Asset path - Level 9 a_z Substation

5 Node-RED workflow for Phoenix Contact energy meter EEM MA700

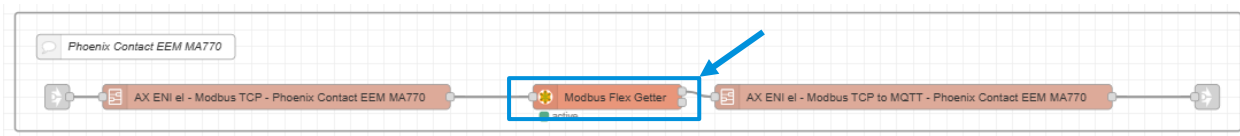
5.1 Introduction to the Energy Insights electric Node-RED workflow

The ENI Electric Phoenix Contact Node-RED Flow reads data via the Modbus protocol from the EEM MA 770 energy meter. It collects essential electrical parameters including voltage, current, power measurements, and energy meter data.. Each data point includes a timestamp (ts) indicating when the measurement was taken, and a topic that specifies the data path within the system for proper routing and identification.

For each sensor there is one line in the flow composed of 3 main nodes: a subflow to define the registers to read out, the Modbus TCP client node (Modbus Flex Getter) and a subflow to transform the read data into the required MQTT format. This line needs to be copied and configured for each sensor that needs to be connected.

5.1.1 Modbus Flex Getter node

Adjust the Modbus Flex Getter node



1. Adjust the Server in the Modbus-Flex-Getter

The screenshot shows the 'Edit Modbus-Flex-Getter node' dialog. At the top are 'Delete', 'Cancel', and 'Done' buttons. Below is a 'Properties' section with 'Settings' and 'Optionals' tabs. The 'Settings' tab is active, showing a 'Name' field with the placeholder 'Name', a 'Server' dropdown menu with the value 'modbus-tcp@192.168.0.30:502', and a blue rectangle highlighting the edit icon (pencil) next to the dropdown. Below the 'Server' field are two checkboxes: 'Delay to activate input' and 'Allow oversized messages', both of which are currently unchecked.

Edit Modbus-Flex-Getter node > Edit modbus-client node

Delete Cancel Update

Properties

Settings Queues Optionals

Name

Type

Host

Port

TCP Type

Unit-Id

Timeout (ms)

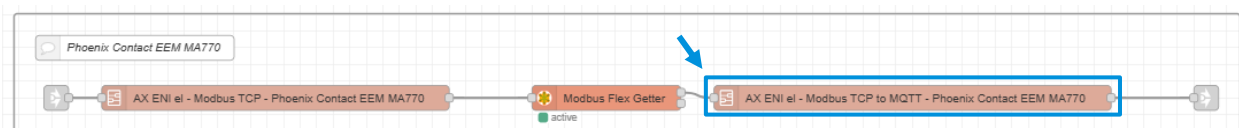
☒ Reconnect on timeout

Reconnect timeout (ms)

- Set Type to TCP
- Set the Host to the IP Address from your Master Module CPX-AP-I-EP
- Enter the Port Number (default is 502)

5.1.2 Configure subflow “AX ENI el - Modbus TCP to MQTT – Phoenix Contact EEM MA770”

Each sensor or measurement unit uses a distinct topic and corresponds to a unique asset in Festo AX. This separation ensures clear identification and organization of data streams, enabling precise monitoring and analysis of each measurement point.



- Define if you want to use the consumption counter of the device or just the power values in ENI.
- If you want to suppress negative values, set the option to true. If you don't want to suppress negative values, set it to false.
- Set the asset name to a meaningful name that identifies the sensor or measurement point.
- Update the asset path according to your system's hierarchy, using the 9 different levels.

Edit subflow instance: AX ENI el - Modbus TCP to MQTT - Phoenix Contact EEM MA770

Edit subflow template
Delete
Cancel
Done

Properties

Name	Name
Use consumption counter	☐ false
Negative power/energy value supression	☒ true
Asset name	electric sensor
Asset path - Level 1	Country
Asset path - Level 2	Site
Asset path - Level 3	Hall
Asset path - Level 4	Floor
Asset path - Level 5	Segment
Asset path - Level 6	Workgroup
Asset path - Level 7	Line
Asset path - Level 8	Station
Asset path - Level 9	Substation

6 MQTT connection to Festo AX broker

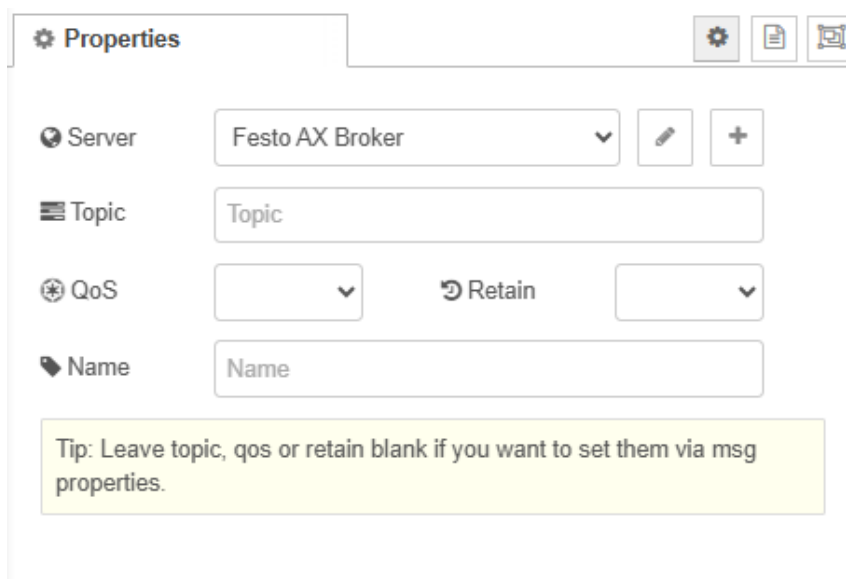
To onboard the electric or pneumatic measurement points, a mapping of the sensor values to MQTT messages following the specified data structure must be performed. Once the MQTT data, organized according to the defined MQTT topic hierarchy, is published to the MQTT broker of Festo AX, the asset auto-onboarding process is triggered automatically.

- MQTT connection to Festo AX MQTT broker

Server: mosquitto-extern

Port: 1883

Security: none



Properties

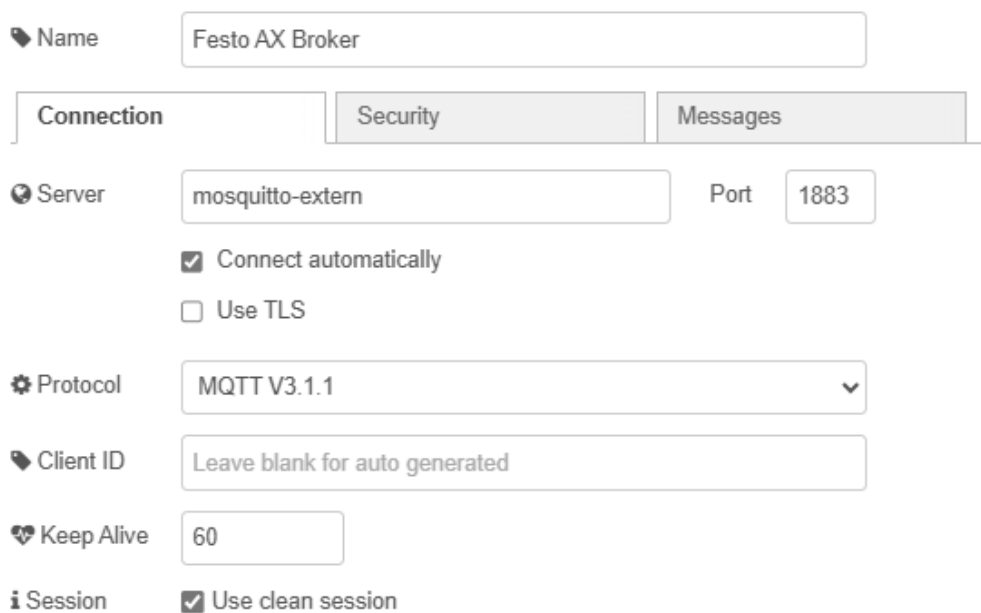
Server: Festo AX Broker

Topic:

QoS: 0 Retain: ☒

Name:

Tip: Leave topic, qos or retain blank if you want to set them via msg properties.



Name: Festo AX Broker

Connection | Security | Messages

Server: mosquitto-extern Port: 1883

☒ Connect automatically

☐ Use TLS

Protocol: MQTT V3.1.1

Client ID: Leave blank for auto generated

Keep Alive: 60

Session: ☒ Use clean session