



MQTT - Internal Broker service

Technical manual

NOTE:

The Information in this manual is in a “draft” state, meaning, that the contents may be incomplete or not completely up to date with the myGEKKO OS Version you may be using.

Table of Contents

Safety and Warranty.....	2
Safety instructions	2
Warranty.....	2
Internal MQTT Broker service document	3
Purpose.....	3
Configuration and connection.....	4
Basic configuration	4
Configuration test.....	7
Basic connection of software clients.....	8
Testing and monitoring the MQTT internal broker	12
Testing the message publishing	12
Monitoring the messages	13
Simulating MQTT-based devices	17
Creating a simulated device on MQTTX	17
Connecting real MQTT-based devices.....	19
Annex 1: checking the network configuration of myGEKKO controller	20
Annex 2: how to download and install MQTTX	22



Safety and Warranty

The devices are manufactured in a safe and workmanlike manner in compliance with current regulations and are operationally reliable. They have been tested and have left the factory in perfect condition.

However, there are residual risks. Read and comply with safety instructions to avoid risks.

Ekon GmbH assumes no liability for damages caused by non compliance with the safety instructions.

Safety instructions

Safety instructions about the product described in this document are listed below. Please observe them carefully while using the product.

Electric Voltage

Danger to life and fire hazard from electrical voltage

Inside the device are unprotected live component parts. Observe the VDE regulations.

Disconnect all cables to be installed and take safety precautions against unintentional switching on. Do not operate the device if damaged or not functioning properly. Decommission the device or system and secure it against accidental operation, if any operation without risk may supposedly not be guaranteed.

Device damage from external influences!

Moisture and dirt on the equipment can lead to its destruction.

Protect devices from moisture, dirt and damage during transport, storage and operation.

Warranty

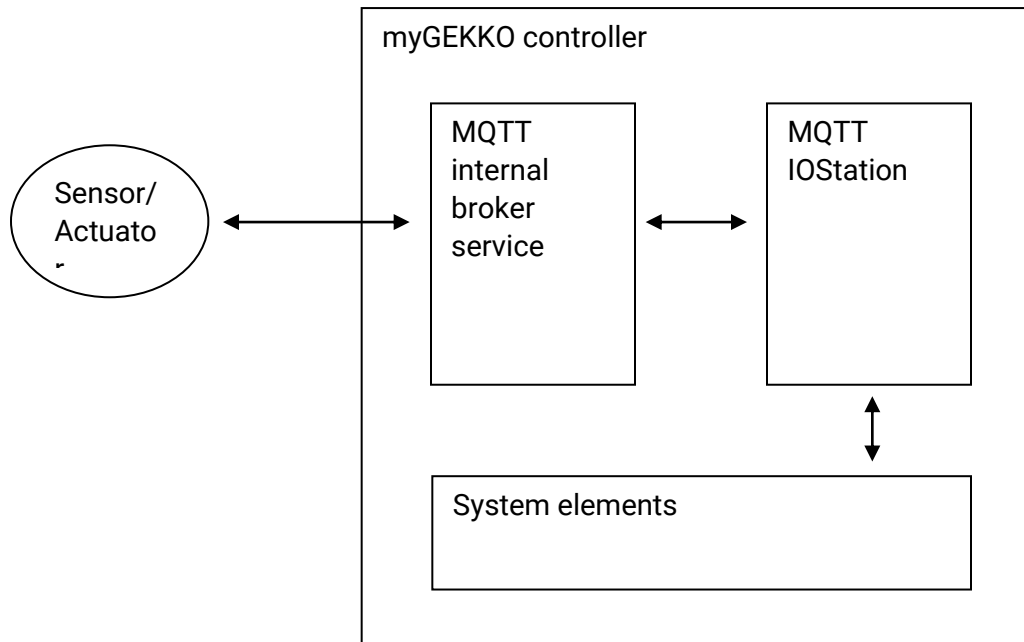
The device is intended solely for proper use. Any inappropriate change or non compliance with the operating instructions invalidates any warranty or guarantee. After unpacking, check the device immediately for any mechanical damage. In case of damage caused during transportation please immediately inform the supplier. The device must only be operated as a fixed installation, i.e. only when installed and after all of the installation and commissioning work has been completed, and exclusively in the environment it is intended for. Ekon GmbH shall not be liable for any modifications to standards and norms after publication of the operating instructions.



Internal MQTT Broker service document

Purpose

The MQTT internal broker service enables the communication of the myGEKKO building controller with any MQTT-based device to collect/send information from/to them. The information can then be read by the MQTT IO-Station and linked to any system element, such as light, load, or room temperature control.



The MQTT internal broker service can be initiated using any of the IP addresses available in the myGEKKO controller, as the controller could be connected to multiple networks at the same time (e.g. myGEKKO BASE in a Technical network and the private network). In addition, a custom user and password can be set to manage the rights of the devices that can be connected to the broker.

Since the purpose of the MQTT internal broker service is to act as a bridge between the sensors and the IO-Stations (more than one MQTT IO-Station can connect to the local broker), all the topic structures and payload formats are allowed. The processing of both topic and payload will be done in the IO-Stations according to the profiles file, making this architecture flexible and scalable.

The Mosquitto open-source message broker (<https://mosquitto.org/>) was used within the myGEKKO building controller as MQTT internal broker due to its extended community of users and usability. For more information, please visit the official manual website (<https://mosquitto.org/documentation/>).



Configuration and connection

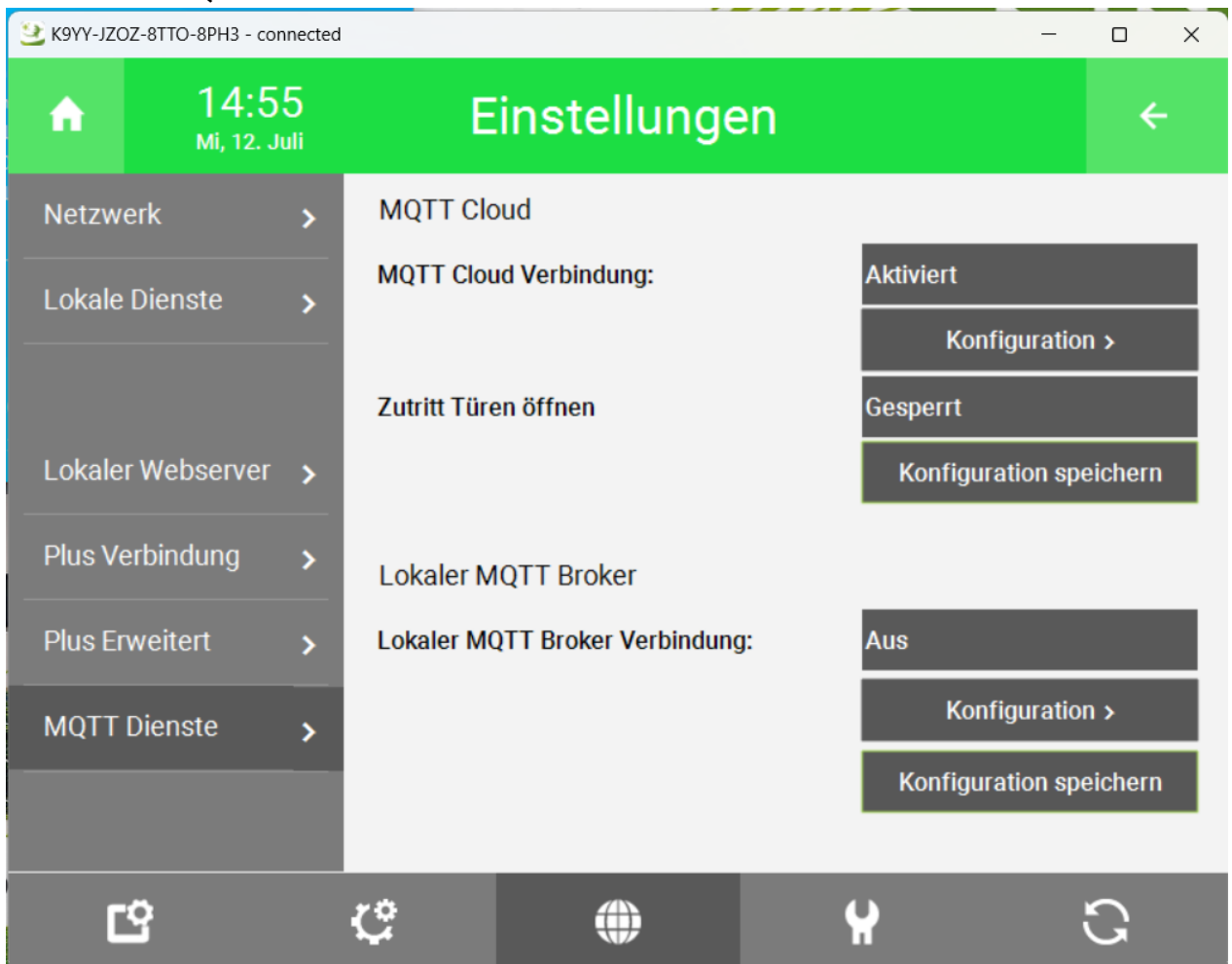
This section shows, in general, how to configure the myGEKKO internal MQTT broker service.

WARNING: Since the myGEKKO building controller can have multiple network interfaces, please carefully read the information about the networks to properly configure the MQTT local broker.

Basic configuration

With the following instructions you will be able to configure the MQTT Broker in the myGEKKO OS.

- 1) Open the “Einstellungen” menu
- 2) Select the network settings (globe icon) on the bottom
- 3) Select “MQTT Dienste” on the left. There you can see the “MQTT Broker Verbindung” and the “Interner MQTT Broker Dienst” menus.



- 4) Click by “Lokaler MQTT Broker Verbindung” on „Konfiguration >“. The configuration screen will be shown.



5) How to fill the boxes is described below.

K9YY-JZOZ-8TTO-8PH3 - connected

Lokaler MQTT Broker Konfiguration

Broker Adresse Port

Benutzername Passwort

Status prüfen

Nicht verbunden

Zurück

The configuration screen has 4 text boxes:

- **Broker address:** the IPv4 address where the broker will initialize the service and to where the devices should connect to. By default, the broker address is set to localhost (127.0.0.1)
WARNING: this IP address must be one of the IP addresses available within the myGEKKO controller and accessible for the MQTT devices. Please check the Annex 1 at the end of the document to learn how to check the available IP address of your myGEKKO controller.
- **Port:** the port where the connection of the broker will be open. If the connection does not require a secure connection, port 1883 is used. However, if a secure connection is needed, port 8883 is mandatory. By default, the port is set to 1883.
- **Username:** the username to allow devices to connect to the local broker when configuring a new device. By default, the username is set to "mqtt_mygekko_admin".



- **Password**: the password to allow devices to connect to the local broker when configuring a new device (along with the username). By default, the password is set to "mqtt_mygekko_admin".

NOTE: All the values can be changed by the user according to its needs, under its own responsibility. Some advanced knowledge may be needed. Therefore, if you are not sure about what you are doing, please contact the support service.

There are also two buttons:

- **Status prüfen**: to check the current status of the local broker process. The two possible statuses are:
 - o **Nicht verbunden**: the process is not running.
 - o **OK**: the process is running.
- **Zurück**: to go back to the MQTT Dienste menu.

There are two ways to configure the local broker.

The first one (1) is just to go to the "MQTT Dienste" menu, section "**Interner MQTT Broker Dienst**", and left click once on the button "**Konfiguration Speichern**". With this, the default configuration will be used when generating the configuration file of the MQTT broker.

(picture1: the menu MQTT Dienste screenshot adding a circle on the button **Konfiguration Speichern** of the **Interner MQTT Broker Dienst** section)

(picture2: a screenshot of the **Interner MQTT Broker Dienst** configuration menu showing the by-default values added automatically after clicking on the save button)

The second one (2) is to fill up the text boxes manually with the desired configuration. For example:

-After checking the network configuration of the myGEKKO controller (Annex 1), the IP address of the controller is 192.169.2.34. I decide to use this IP address as the address of the MQTT local broker. Then, I also set the port to 1883 since I do not want to set a secure connection (certificate-based connection) to make the connection of the devices easier. Finally, I decide to set a new username: *myHouse*, and a new password: *!my!nH0us3*.

The local MQTT broker configuration screen would look like this:

(image: a screenshot of the **Interner MQTT Broker Dienst** configuration menu showing the textfields filled with this information)



Then, the user must go back to the **“MQTT Dienste”** menu (by clicking on the **“Zurück”** button) and click on the **“Konfiguration speichern”** button to save the new configuration.

NOTE 1: when clicking in the **“Konfiguration speichern”** button, it turns and stays green for a few seconds. Please wait patiently until the button turns back to the grey color, meaning that the configuration file was successfully updated.

NOTE 2: if a text box is empty, the system will fill up automatically with the default value. Therefore, it is recommended to set up every text box or check the final configuration after clicking on **“Konfiguration speichern”** button.

Configuration test

After setting up the configuration of the MQTT internal broker, you can test the MQTT internal broker to check if it is working properly.

To do that, in the **“MQTT Dienste”** menu, **“Interner MQTT Broker Dienst”** section, go to **“Interner MQTT Broker”**. Click on the right side to change it to **“Aktiviert”**. This will start the broker service with the given configuration.

(picture1: the menu MQTT Dienste screenshot adding a circle on the button **Interner MQTT Broker** of the **Interner MQTT Broker Dienst** section)

Then, go again into the configuration, and click once on the button **“Status prüfen”**. Wait for a few seconds. If the status does not change to OK, wait 5 seconds, and try it again. Sometimes, the new processes need a few seconds to start and get listed in the processes list.

(image: screenshot of the menu showing that the status is OK!)

If after the second time the status remains in **“Nicht verbunden”**, it means that something went wrong when configuring the broker. Please check the broker’s configuration and make sure that both the IP address and the port are correct.

(image: a screenshot of the menu showing the error status!)

On the other hand, if the status shows **“OK”**, it means that the MQTT internal broker is running correctly and is ready for use.



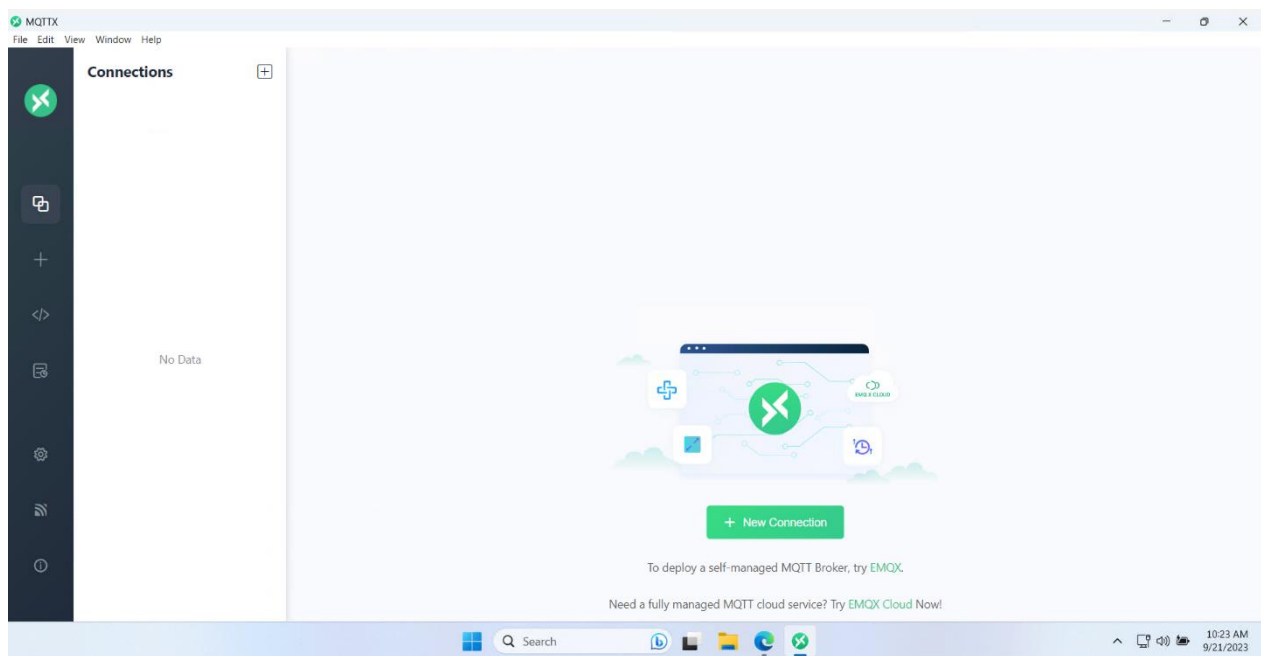
Basic connection of software clients

At this point, our MQTT internal broker is ready to receive messages from clients. A client can be defined as any program or device that uses the MQTT client library. A client can connect to the broker, publish messages, and subscribe to topics to receive messages.

Therefore, we could connect to our local MQTT broker to publish messages or to check the messages that other clients are publishing to the MQTT internal broker.

Among the large list of MQTT client tools that can be found on internet, we used MQTXX, an open-source tool which offers a cross-platform GUI that can run on Windows, Linux, and MacOS. Please go to Annex 2 to read how to download and install MQTXX from scratch.

After the installation, the main screen is shown.

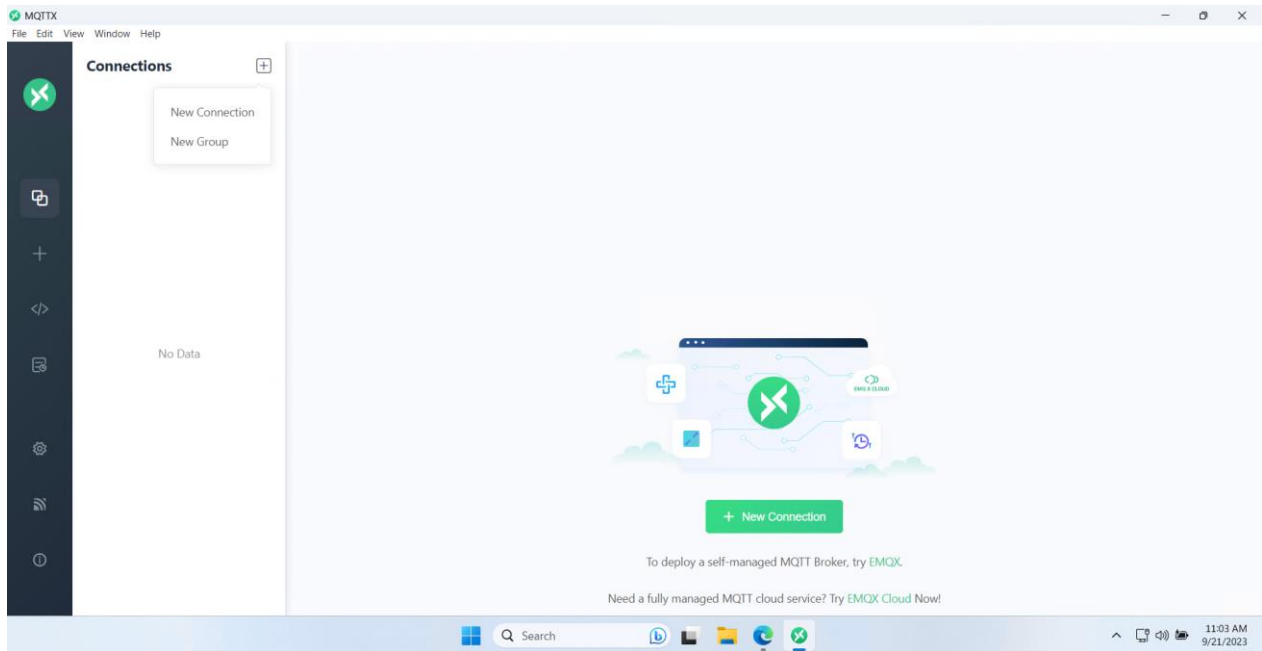


To add a new connection to a MQTT broker, go to the top left corner, and click on the “+”-symbol close to the word “Connections”. A small pop-up menu is shown with two options:

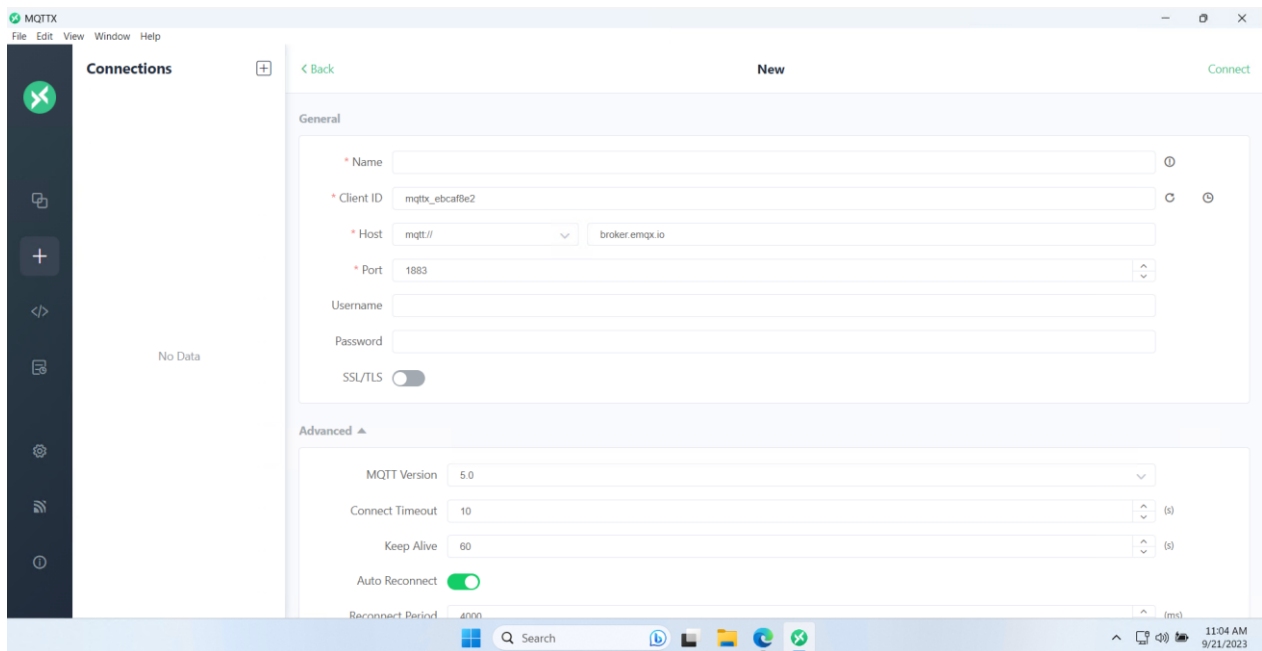
- 1) New Connection.
- 2) New Group.

Click on “New Connection”





The configuration screen is then shown, as follows:

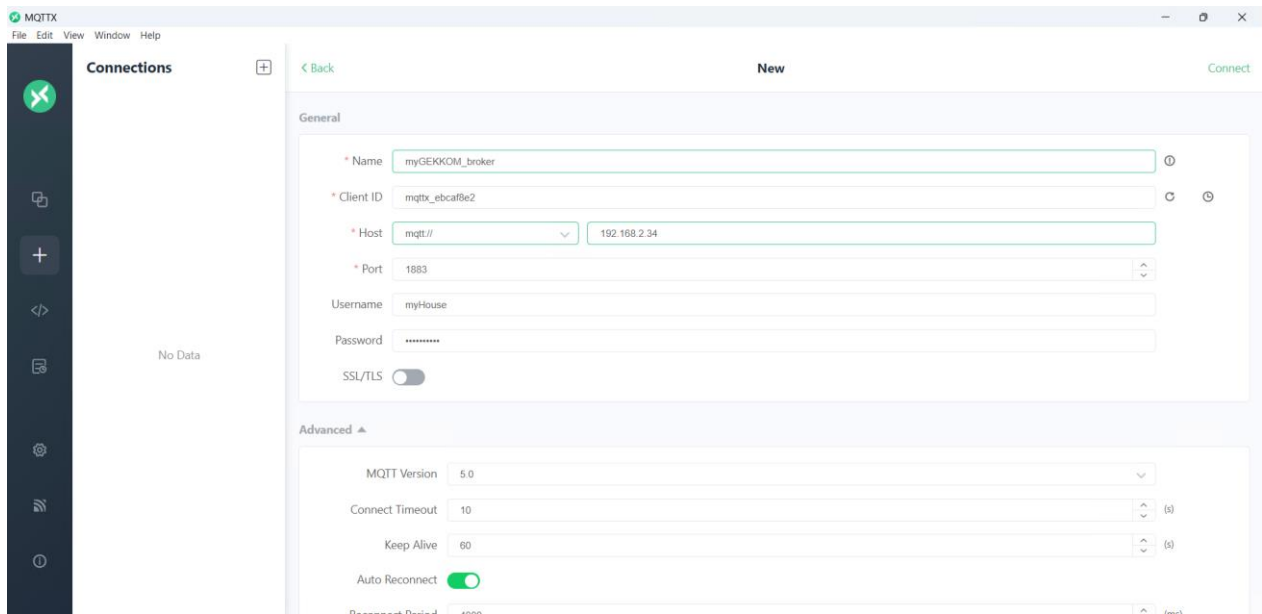


To configure the connection:

- **Name:** a custom name to identify the connection inside the MQTTX application. For example, "myGEKKO_broker".
- **Client ID:** a unique identifier for the client inside the broker. Although the user can modify this field, it is better to leave it with the default value that the tool selected.

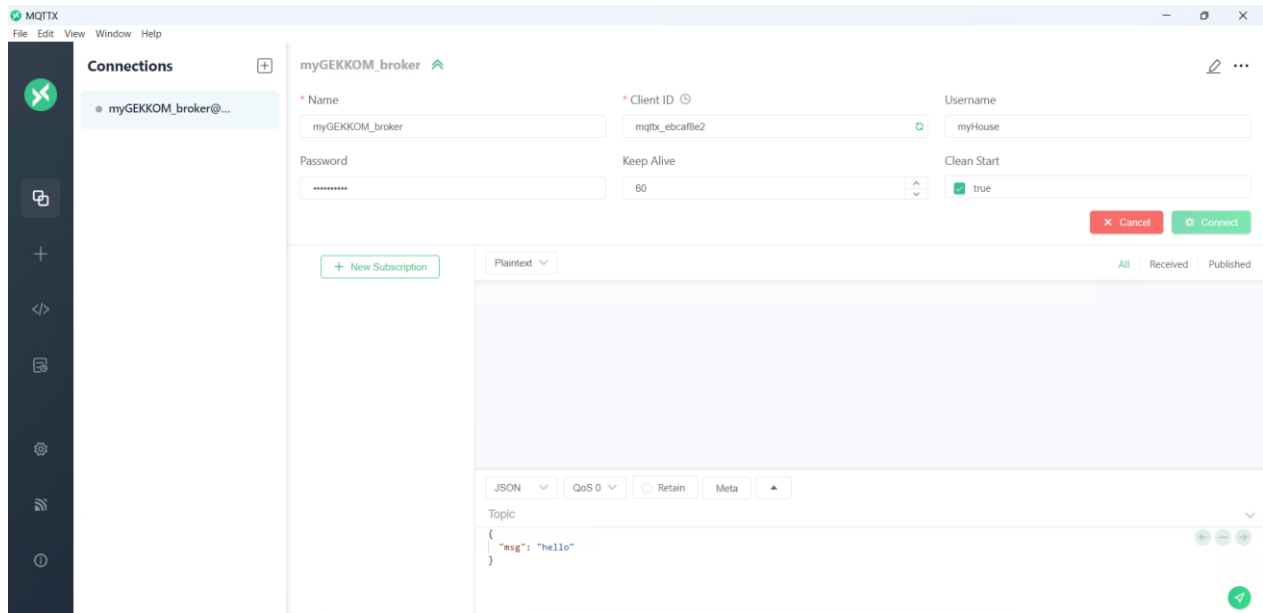


- **Host:** the type of the server that is hosting the broker. Since in the myGEKKO controller we set a basic MQTT broker, we need to select the option “mqtt://”. Then, in the text box, we will write the IP address that we set in the internal MQTT broker configuration. In our case, and following the example above, 192.168.2.34.
- **Port:** the port to set the connection with the host indicated before. As with the host, we need to write the same port number as we set in the configuration. In our case, 1883.
- **Username:** the username of the credentials to connect to the broker. It must match with the username set in the configuration. In our example, myHouse.
- **Password:** the password associated to the username to connect to the broker. It must match with the password set in the configuration. In our example, !my!nH0us3.
- **SSL/TLS:** We don't use it. Turn it off if not already set so.



Then, click on the “Connect” button located at the top right corner. The connecting process will start, showing the message “Connected!” if the connection was established. If not, a message of “Reconnecting” or “Disconnected” will appear.





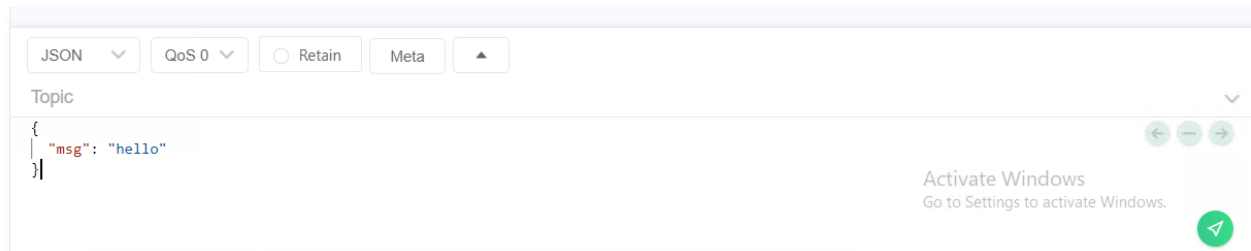
Now you are ready to subscribe to topics and publish messages.



Testing and monitoring the MQTT internal broker

Testing the message publishing

First test we can do is to publish a message. To do that, take a look to the bottom part of the MQTTX window:



We can send a test message like, for example:

-Topic: "myproject/test/dummy_message"

-Message: {"msg": "hello"}.

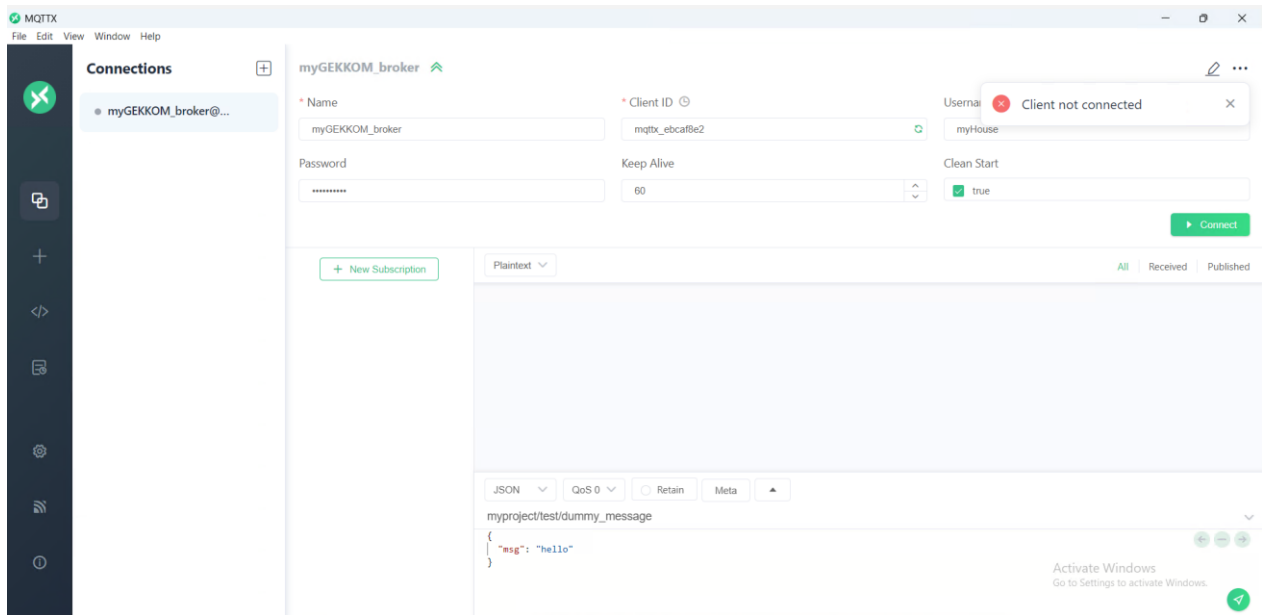
Note that the message is in JSON format, as indicated on the top left box. To send the message, left click once on the rounded green button placed on the bottom right corner.

If the message was sent successfully, you should see something like this:

(image: screenshot of the MQTTX app showing the message)

On the other hand, if there was any problem, an error or warning message will be shown indicating the source of the error. For example, if the client is not connected to the broker but we tried to send the message, we will get the error "Client not connected", meaning that either there was a problem on the broker side or in the MQTTX side:

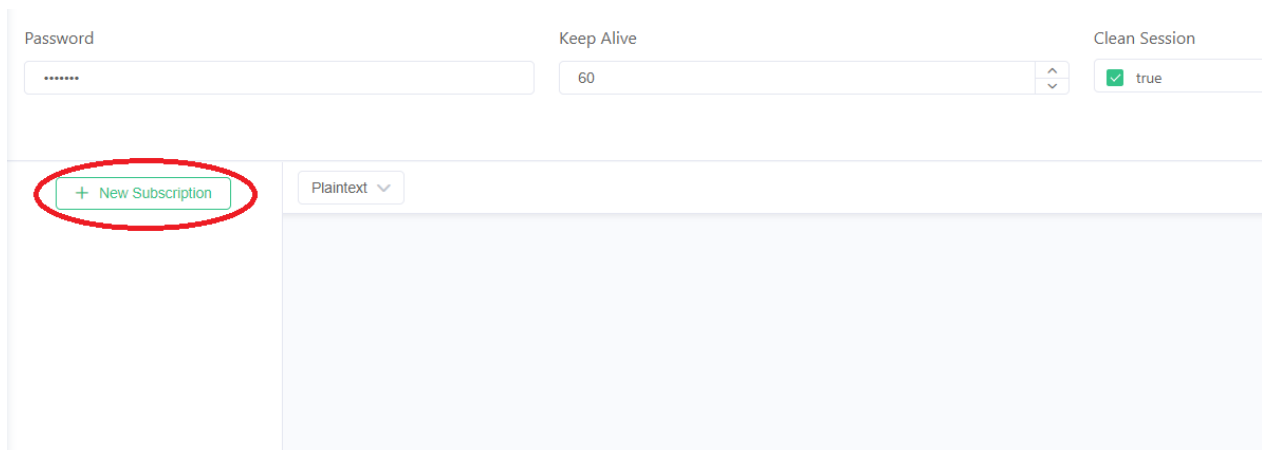




An easy way to test the local broker is to connect the **myGEKKO Broker Connection service** (**please read the manual XXXX**) to the **myGEKKO MQTT internal broker**. To do that, configure the **Verbindung zu MQTT Broker** service by using the configuration that you set in the local broker. If you do not know how to do it, please read the manual of the myGEKKO **Verbindung zu MQTT Broker**.

Monitoring the messages

After configuring and enabling the **myGEKKO Broker Connection service**, the controller will start publishing messages to the **myGEKKO MQTT internal broker**. To subscribe to all the messages, click left once on the “New Subscription” button located on the top left corner.



Then, a pop-up window should appear:

The image shows a 'New Subscription' dialog box with the following fields and values:

- Topic:** testtopic/#
- QoS:** 0 (At most once)
- Color:** #348A8C
- Alias:** (empty)
- Subscription Identifier:** (empty)
- No Local flag:** false
- Retain as Published flag:** false
- Retain Handling:** 0

Buttons: Cancel, Confirm

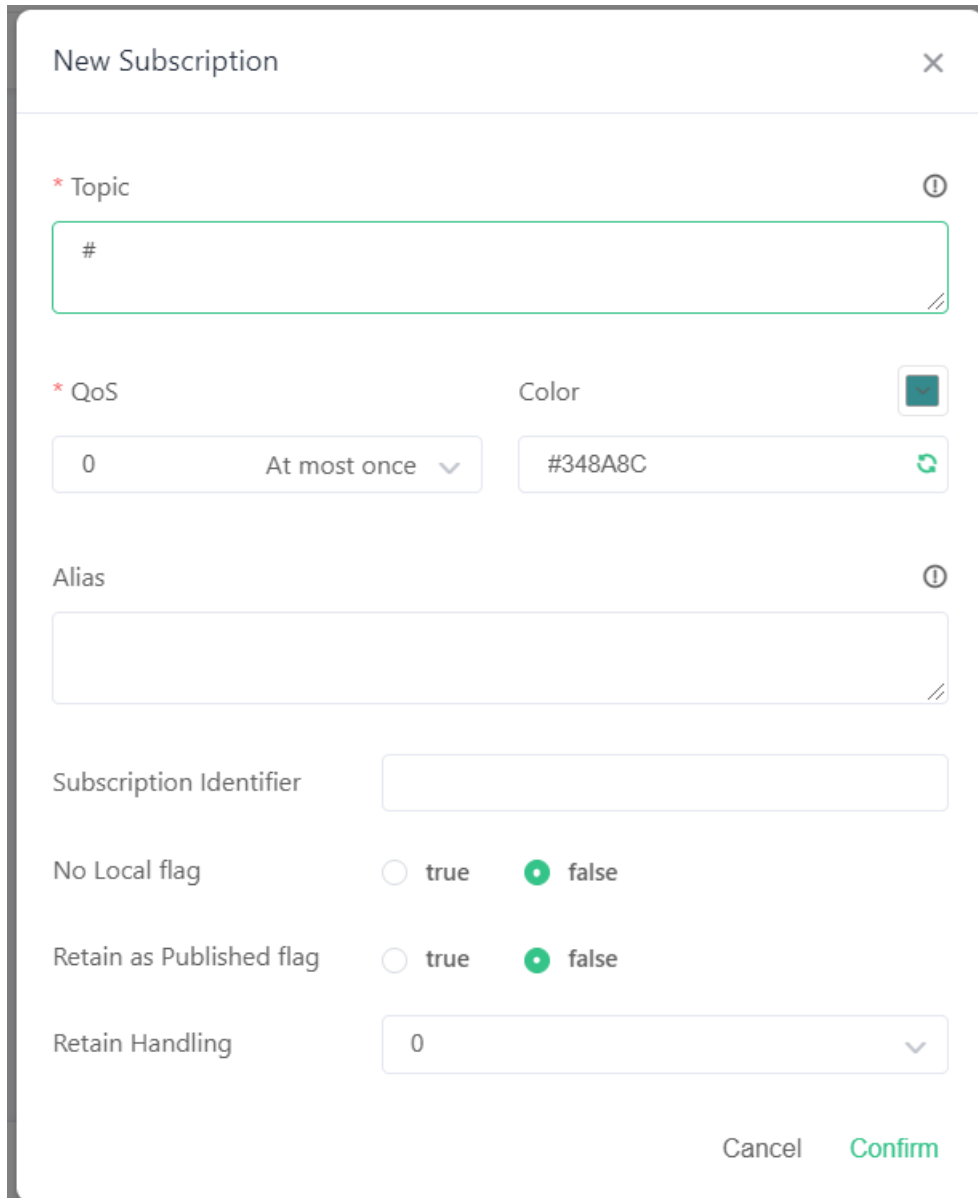
In the field "**Topic**", the user should write the topic to get subscribed. For example, and following the manual of the MQTT Broker connection service, valid topics would be:

- J9Y8-X9NZ-8TUD-5EG3/lights/item0
→ to receive messages from only the item0 of the lights.
- J9Y8-X9NZ-8TUD-5EG3/lights/#
→ to receive messages from all the lights.



- J9Y8-X9NZ-8TUD-5EG3/#
→ to receive messages from all the system elements of the myGEKKO controller.
- #
→ to subscribe to all the messages.

To make it easy, let's subscribe to all (#). The rest of the fields can be left as default:



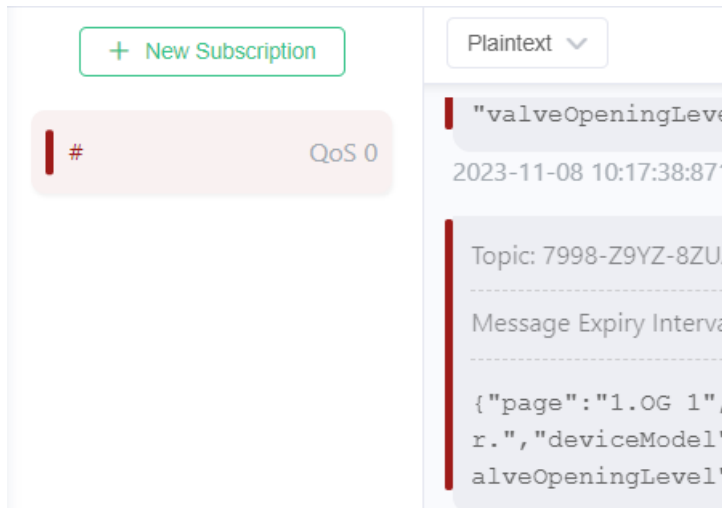
The screenshot shows a 'New Subscription' dialog box with the following fields and options:

- Topic:** A text input field containing '#'. It has an information icon (i) to its right.
- QoS:** A dropdown menu set to '0' with the text 'At most once' and a downward arrow.
- Color:** A color selection field showing '#348A8C' with a color swatch and a refresh icon.
- Alias:** An empty text input field with an information icon (i) to its right.
- Subscription Identifier:** An empty text input field.
- No Local flag:** Radio buttons for 'true' (unselected) and 'false' (selected).
- Retain as Published flag:** Radio buttons for 'true' (unselected) and 'false' (selected).
- Retain Handling:** A dropdown menu set to '0' with a downward arrow.

At the bottom right of the dialog are two buttons: 'Cancel' and 'Confirm'.

Finally, click on the **“Confirm”** button to save the subscription. Then, again in the main screen, a new subscription entry will be shown on the left side right under the “New subscription” button.





Then, the client will start receiving all the messages that are being posted into the myGEKKO MQTT internal broker.

If you want to monitor a specific topic (e.g., a specific sensor), then you will need to remove the “#” subscription and add the specific topic (e.g., shellies/shelly_plug_5234/#).

WARNING: please do not use this tool to send packages manually to the local broker since it requires technical knowledge. It may cause malfunctioning in the myGEKKO controller and, therefore, in your own house.



Simulating MQTT-based devices

The main purpose of the MQTT internal broker is to provide a connectivity bridge between MQTT-based sensors and the MQTT IO stations of the myGEKKO building controller to connect physical sensors to system elements, such as plugs or lights.

Creating a simulated device on MQTTX

MQTT-based devices can be simulated if we know the details of the topic structure and the payload content of a package. By using the MQTTX tool, we can create a package with the topic and payload looking exactly the same as the original sensor for testing or proof-of-concept purposes.

Let's take as reference the plug from Shelly (Shelly Plug S, <https://www.shelly.com/en/products/shop/shelly-plus-plug-s>). If we check the MQTT documentation of the Shelly Plug S (<https://shelly-api-docs.shelly.cloud/gen1/#shelly-plug-plugs-mqtt>), we can see that the topic structure is:

```
shellies/<model>-<deviceid>/relay/0
```

Followed by the message:

on

For example, a package that the plug S would publish into the MQTT broker to indicate its current status (if the plug is ON or OFF) could be:

- Topic: shellies/shellyplug-s- 083A8DC14ECE/relay/0
- Payload: on

NOTE: attention! In this case, the payload is a plain text, not a JSON object.

To simulate this device, a few steps are needed.

1. Open MQTTX tool.
2. Create a new connection with the name "simulated_ShellyPlugS"
3. Use the same broker address as before since we want to publish to our myGEKKO MQTT internal broker.
4. After creating the connection and connecting to the broker, check that everything is ok.
5. In the bottom part, at the top left corner, change the message type from JSON to plaintext.
6. In the topic textbox, write: shellies/shellyplug-s- 083A8DC14ECE/relay/0
7. In the payload textbox, write: on
8. Click on send.

At this point, you should see the message in the local broker connection:



The screenshot displays a MQTT message broker interface. At the top left, there is a '+ New Subscription' button. Below it, a subscription filter is set to 'shellys/shellyplu...' with a QoS of 0. The main area shows two messages received on the topic 'shellys/shellyplug-s- 083A8DC14ECE/relay/0'. The first message, received at 2023-11-08 10:22:41:104, has a QoS of 0 and contains the text 'on'. The second message, received at 2023-11-08 10:22:41:105, has a QoS of 1 and also contains the text 'on'. The interface includes a 'Plaintext' dropdown menu, a 'Retain' checkbox, and a 'Meta' button. At the bottom, there are navigation arrows and a green checkmark icon.

You can simulate as many devices as desired by following those steps. **For devices of a different manufacturer, please carefully read their documentation. Some messages could turn the device unusable.**



Connecting real MQTT-based devices

This step completely depends on each manufacturer, thus needing a manual for each device of each manufacturer.

In general, the devices need to be configured by using the phone app of the manufacturer or through a web app. Since these devices need WiFi connection to connect to the MQTT broker, they normally enable an Access Point (AP) to their local networks, from where they can be configured.

Then, the WiFi connection details can be set, as well as the MQTT broker address, username, and password. Please use the same broker address, username, and password as you used to simulate the MQTT device, along with the manufacturer manual, and it should work.

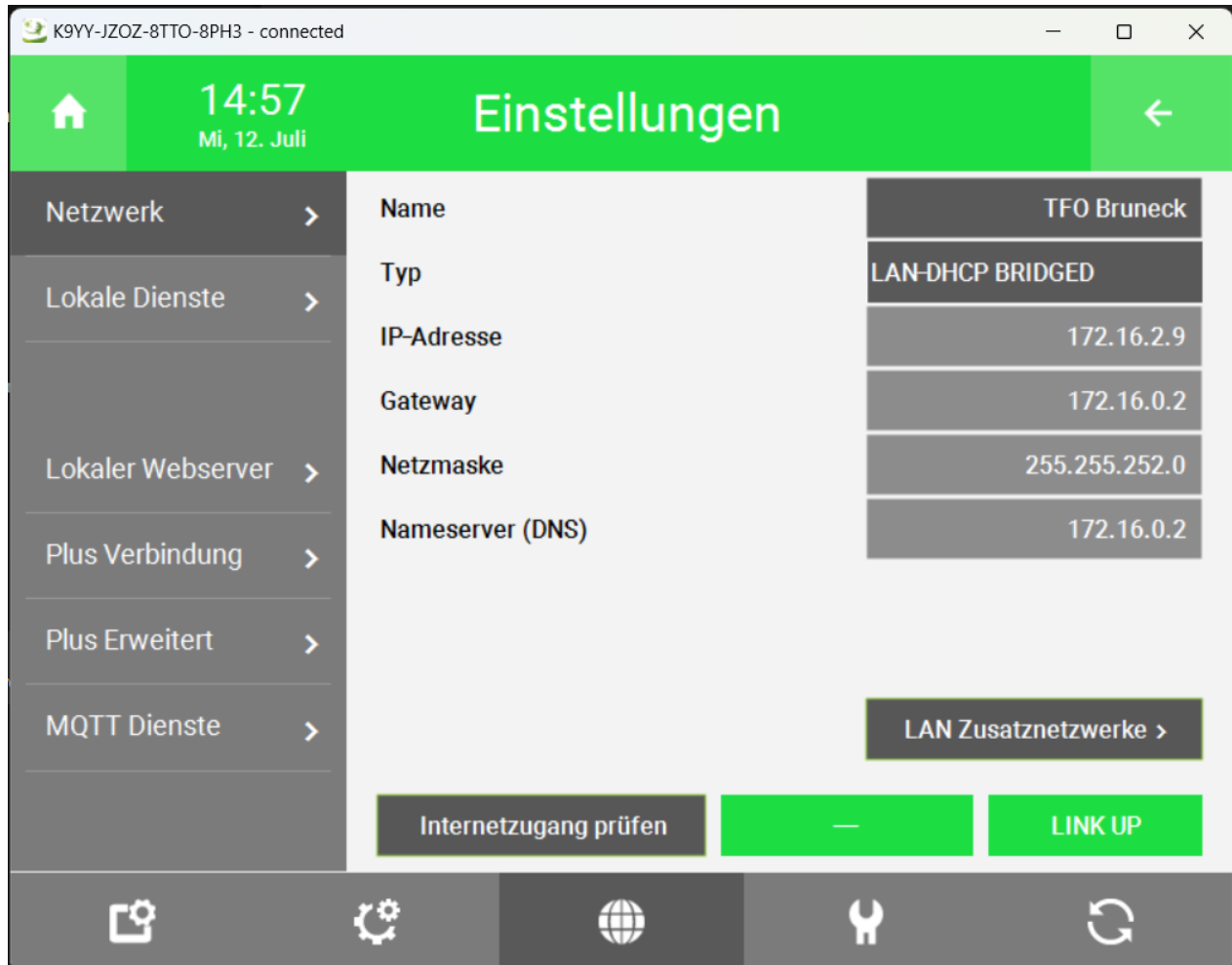
myGEKKO offers manuals for the most common devices to set up the connection to the myGEKKO controller. Please check the list of supported devices and look for the manual.

After connecting the device to the MQTT local broker, please read the MQTT IO Station manual to know how to add a new device, get the data from the MQTT local broker, and link it to the myGEKKO system elements.



Annex 1: checking the network configuration of myGEKKO controller

Go to “Einstellungen” (cog icon on top right), and then go to the network settings (globe icon at the bottom). Then, in the “Netzwerk” section, the network configuration of the myGEKKO Controller is shown:



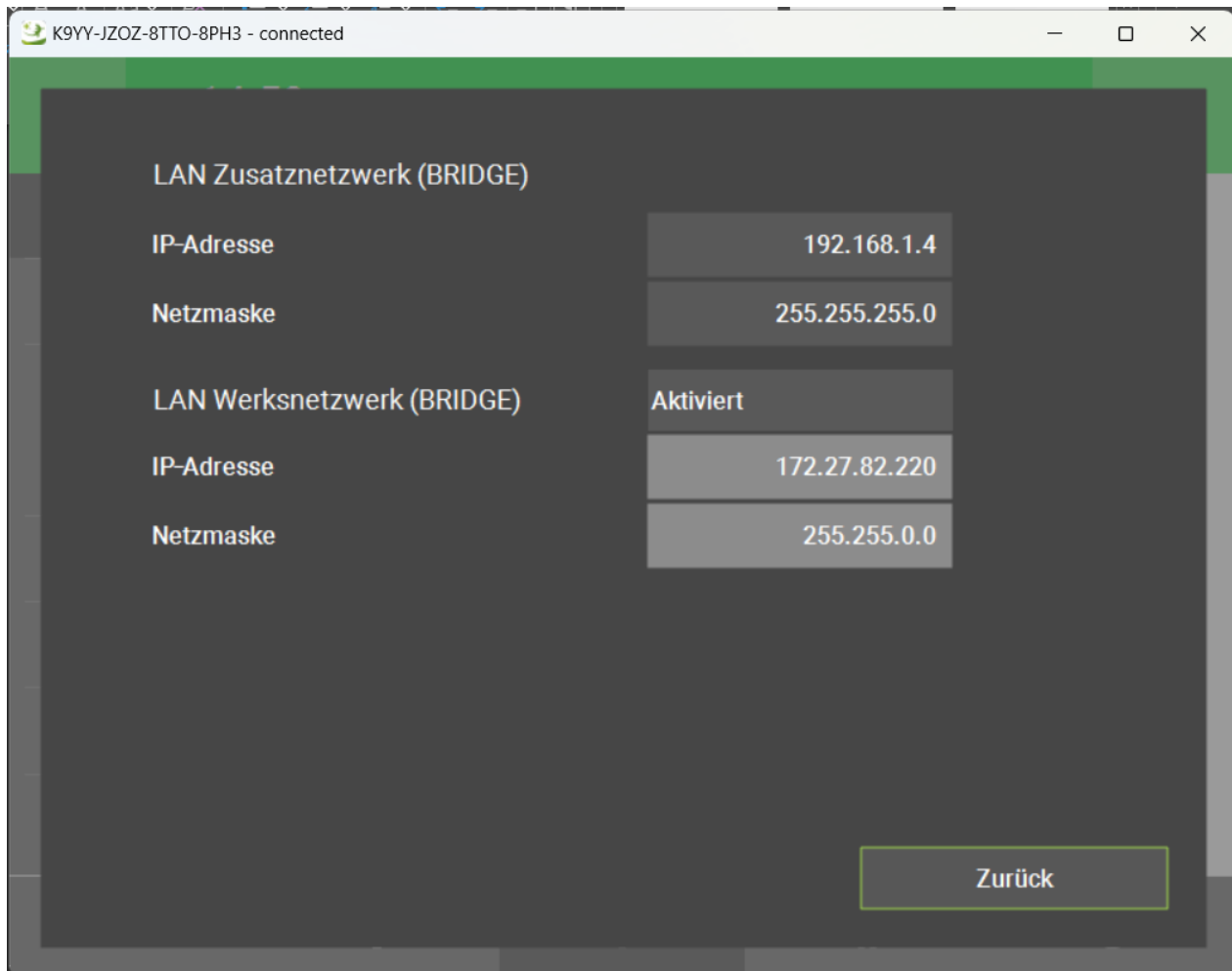
The field “IP-Adresse” shows the current IP address assigned to the myGEKKO controller.

NOTE: this address could be used as the main address of the MQTT internal broker configuration. This means that it will be the address where the MQTT internal broker will listen to the messages.

NOTE 2: we recommend to use this address as the main address for the MQTT internal broker.

To check the local additional networks, click on the “**LAN Zusatznetzwerke**” button. A pop-up window will appear:





Two more networks are shown:

- LAN Zusatznetzwerk, or additional LAN network. It could be used as well for the local broker, but only sensors within this local network could reach it.
- LAN Werksnetzwerk, or factory LAN network. This network is only available if the bridge option is enabled. It could be used as well for the local broker, but normally this network is reserved for other devices and not LoRa gateways.

Summarizing:

The devices that want to connect to the MQTT internal broker must to be in the same network as the myGEKKO controller. This can be: (1) the main network, (2) the additional network or (3) the factory network.

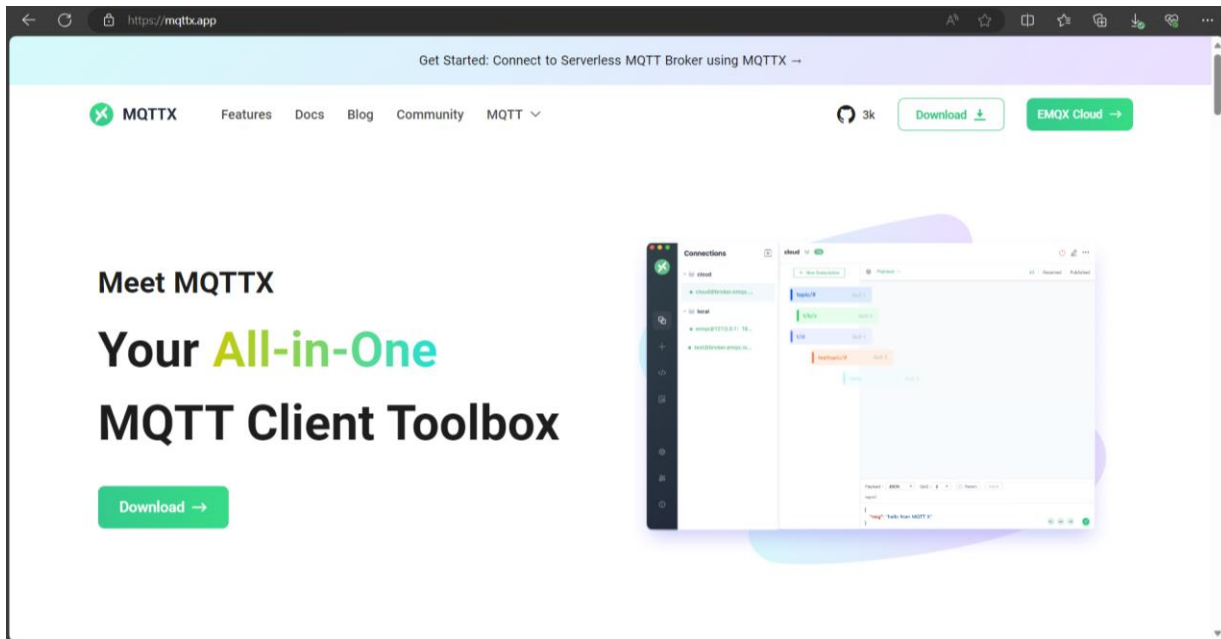


Annex 2: how to download and install MQTTX

Open a web browser and go to the website:

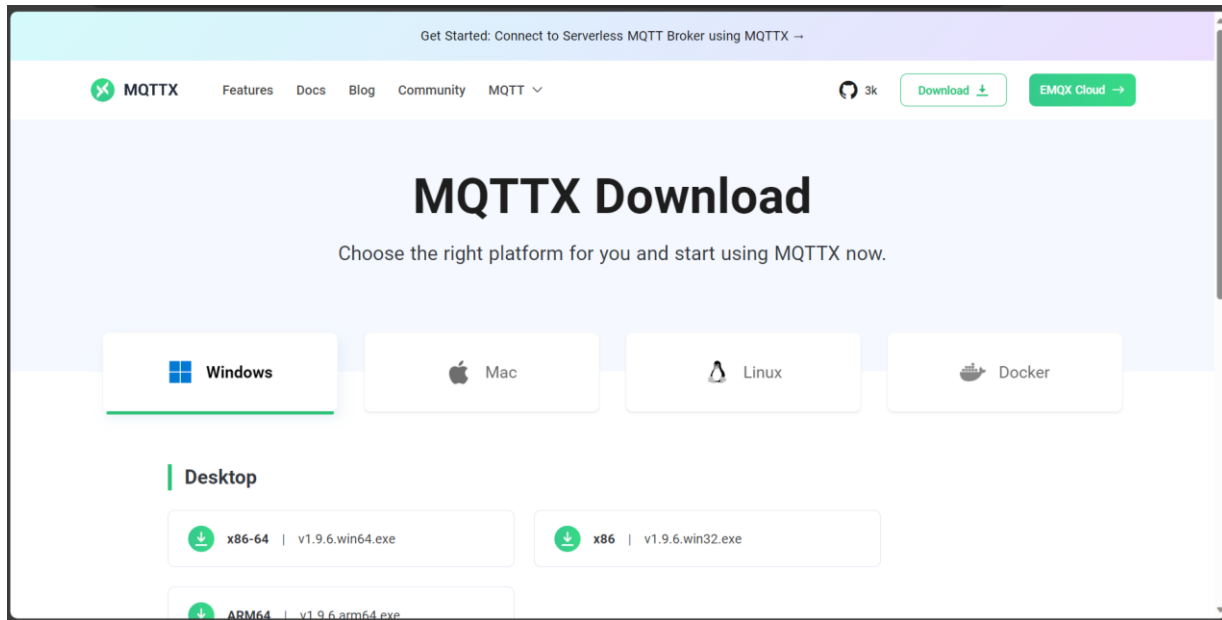
<https://mqttx.app/>

Once there, it should look like this:

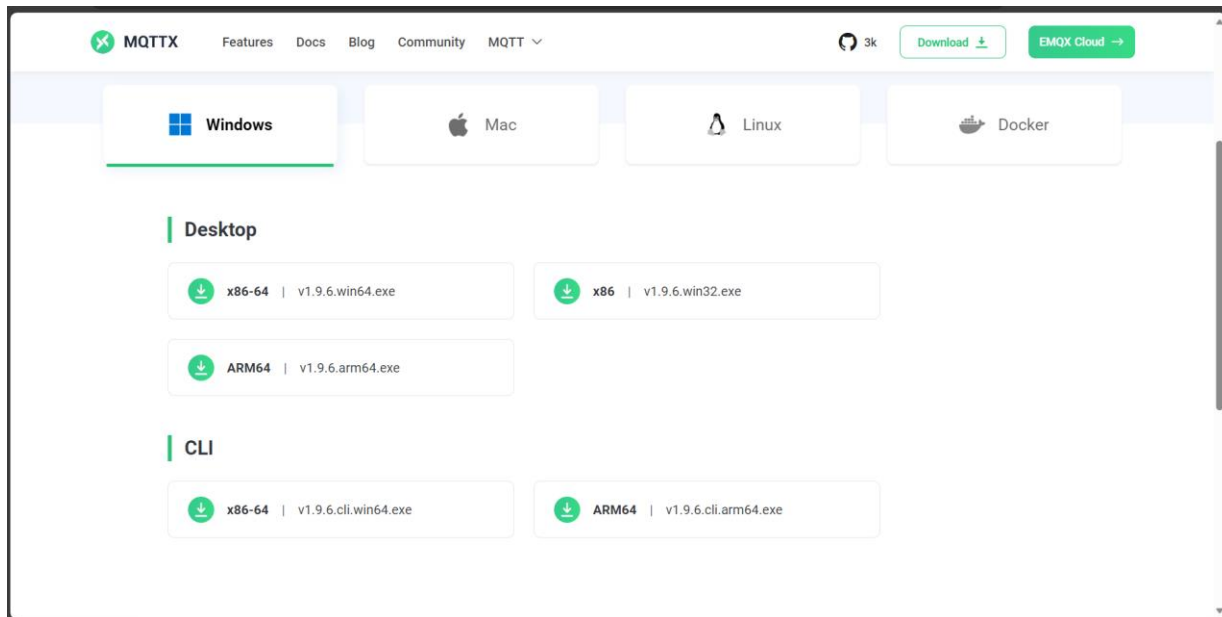


Then, left click once in the “Download” button located at the top right corner of the website. The “Downloads” section should look like this:



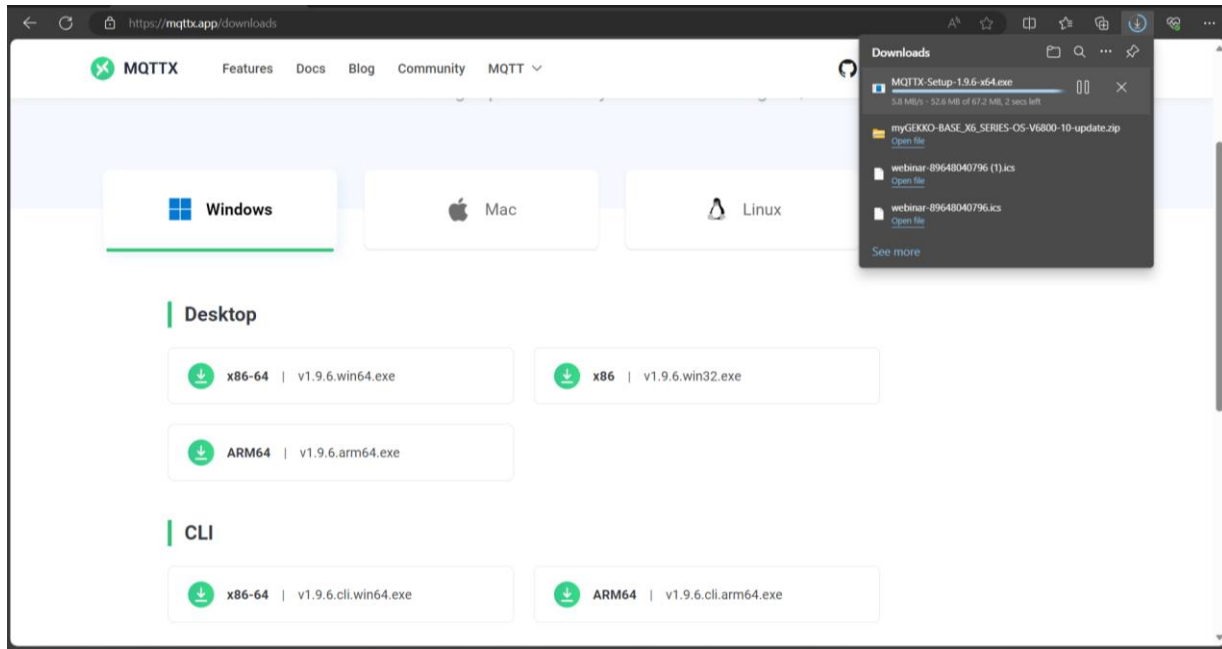


Select your Operating System (OS); in our case, Windows. Scroll down a bit to the section “Desktop” (meaning that we will download the desktop version of the MQTTX software), and select the architecture of your computer: 32 bit (x86) or 64 bit (x86-64). In our case, 64 bit.

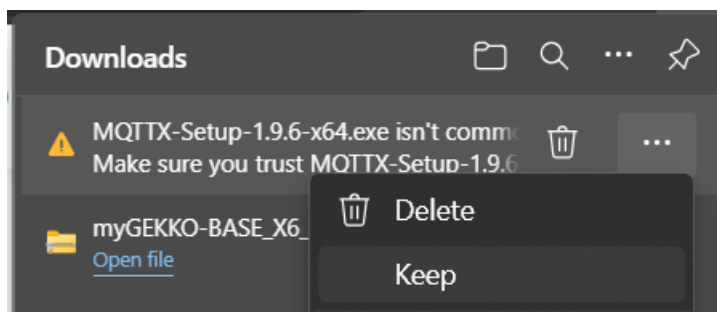
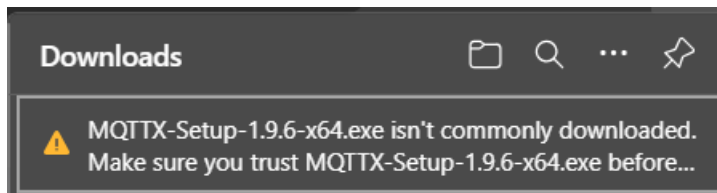


Then, click on the “download” icon of the corresponding architecture. The download process will start automatically.



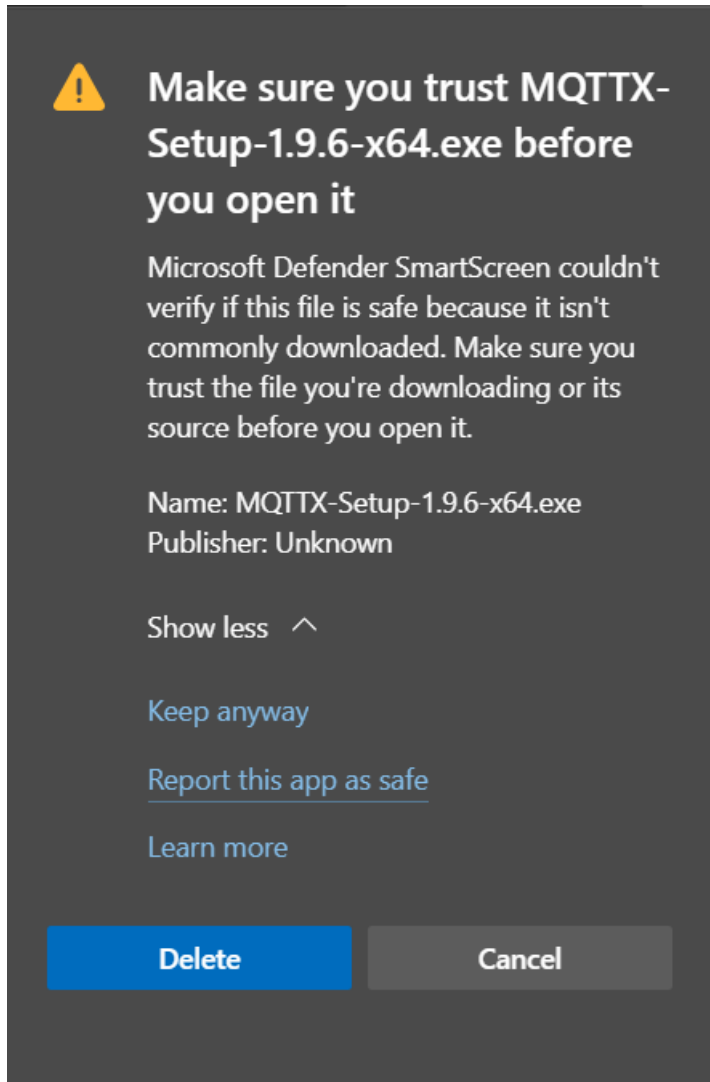


It may happen that the web browser determines that this software is not commonly used. If this happen, the user has to tell the browser that the file must be kept.

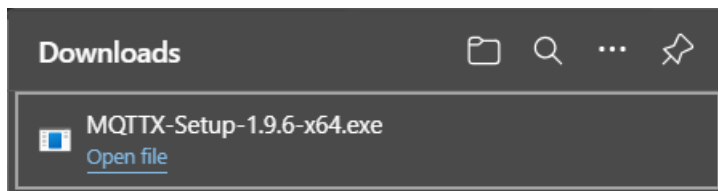


And then to indicate that again with "Keep anyway":





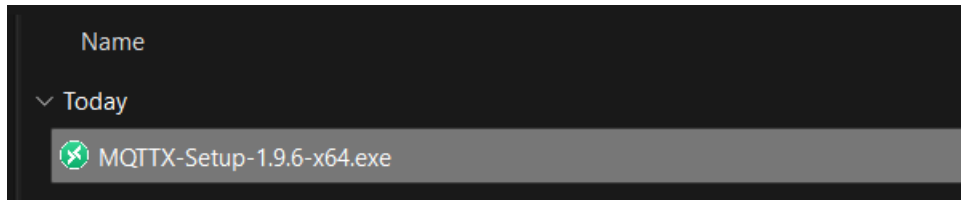
After clicking on the “Keep anyway” option, the installable file is ready to be executed.



The following section explains the installation of in a Windows System. If another System is used, install is as done on those systems and skip these steps.

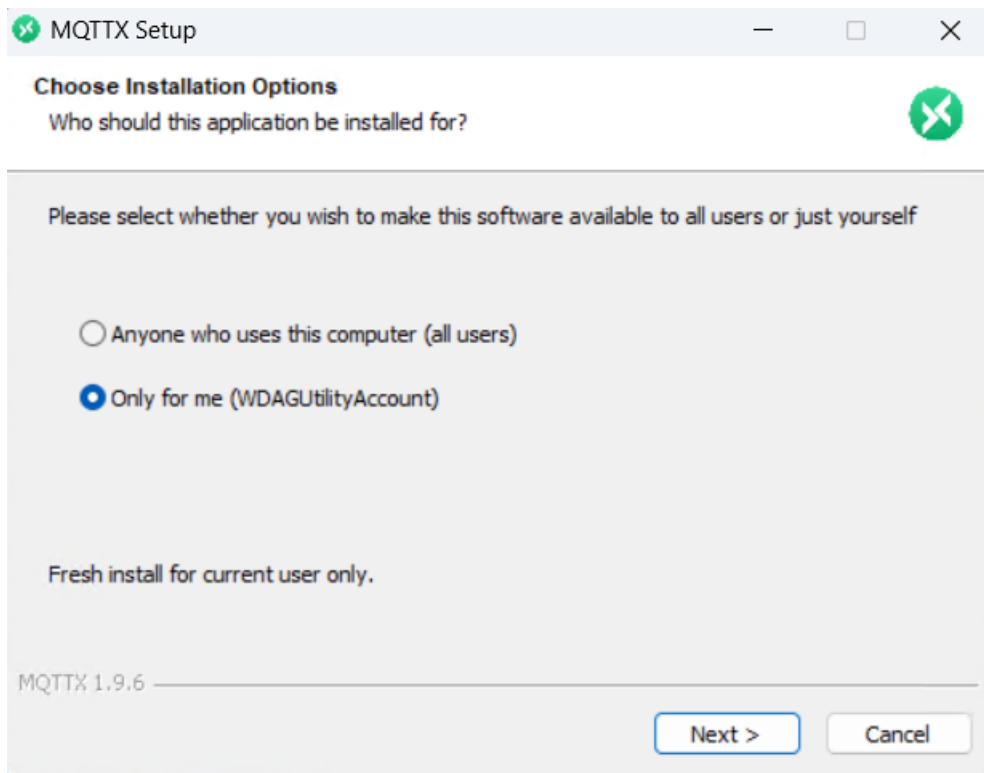
The user can directly click on the executable file from the browser or from the Downloads folder, where it will appear with this name (or similar, depending on the current version):





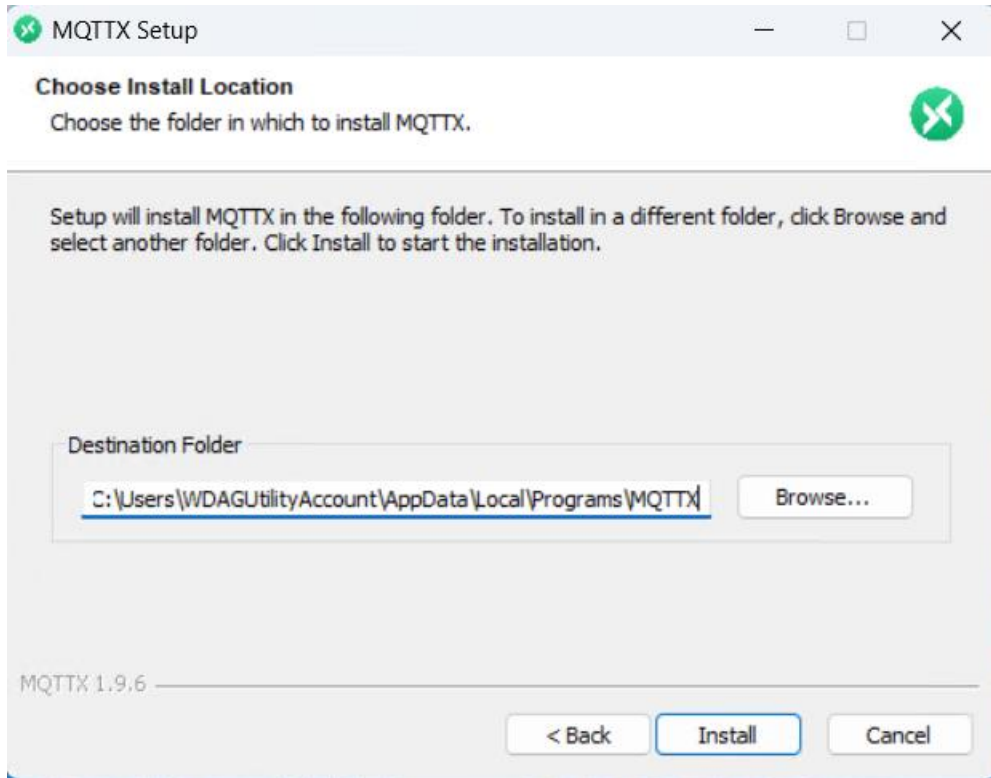
Double left click on the executable to start the installation process.

The first screen is the installation options. The user is free to choose this option.

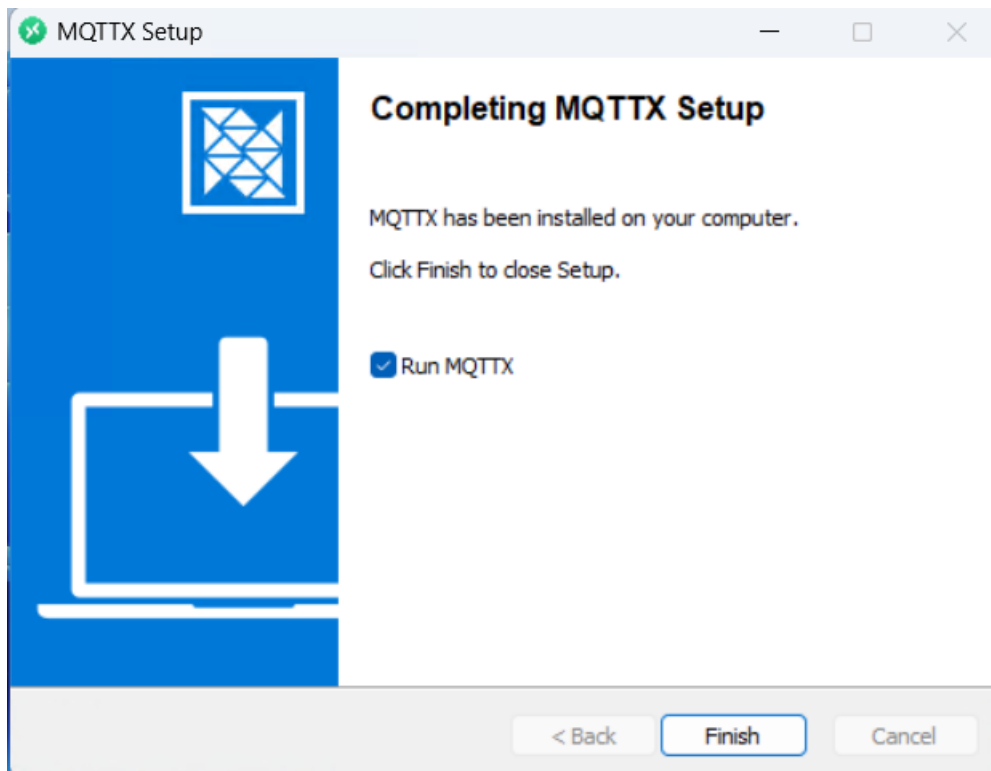


Then, the installation's destination folder can be selected. In our case, we left it as default, but again the user is free to change this.

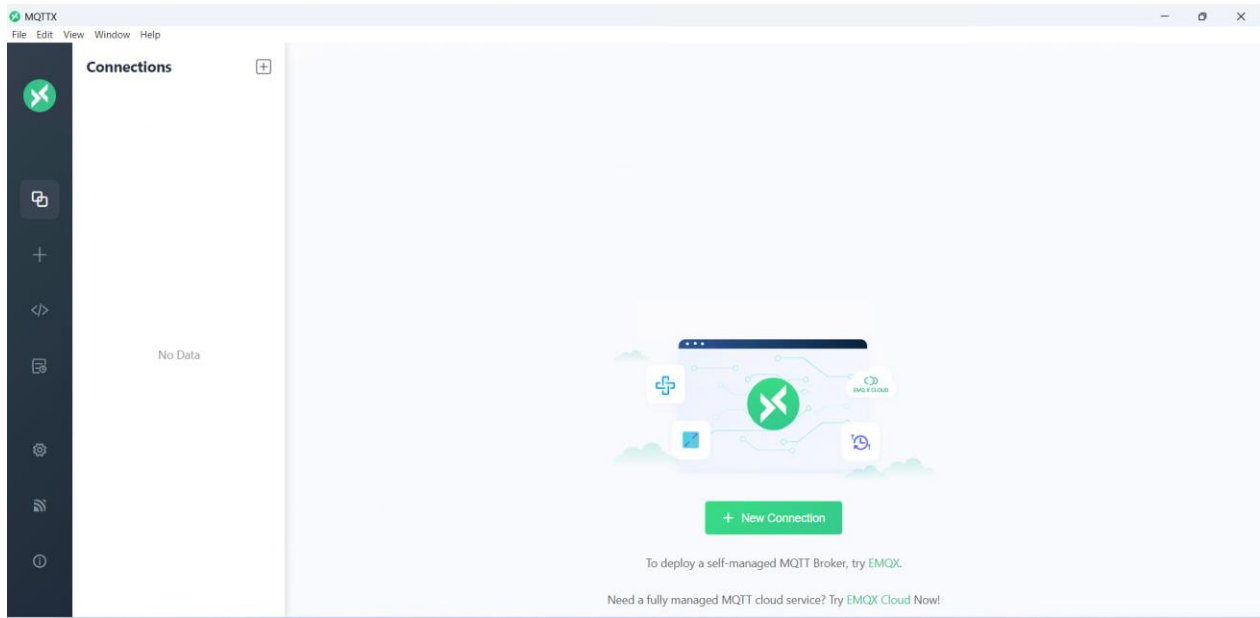




At this point, the installation can start by clicking on the button "Install".



When the installation is finished, MQTTX will be opened and the main screen will be shown, meaning that the tool is ready to use.



myGEKKO | Ekon GmbH

St. Lorenznerstraße 2
I-39031 Bruneck (BZ)

Tel. +039 0474 551820

info@my.gekko.com

www.my-gekko.com

Stand: 21.12.2023

