

Template:RutOS configuration example MQTT Gateway Demo LV



Contents

- [1 Summary](#)
- [2 Configuration overview & prerequisites](#)
- [3 Configuration using flespi.io as MQTT Broker](#)
 - [3.1 Configuring Modbus TCP Server](#)
 - [3.2 Configuring Modbus TCP Client](#)
 - [3.3 Configuring Flespi.io MQTT Broker](#)
 - [3.4 Configuring MQTT Modbus Gateway](#)
 - [3.5 Message format for MQTT Publisher](#)
 - [3.5.1 Examples](#)
 - [3.6 Testing MQTT Publisher and Subscriber on flespi.io](#)
- [4 Configuration using RUT241 router as MQTT Broker](#)
 - [4.1 Configuring MQTT Modbus Gateway settings](#)
 - [4.2 Configuring MQTT Broker settings on RUT241 device](#)
 - [4.3 Configuring MQTT Publisher/Subscriber on PC & testing MQTT Gateway](#)
- [5 See Also](#)

Summary

This guide will demonstrate how to configure the MQTT Gateway on a Modbus TCP Client using two different types of MQTT Brokers. The first configuration involves using a third-party MQTT Broker service "**Flespi.io**". The second setup uses the RUT241 router as the MQTT Broker. In this scenario, two routers will serve as the Modbus TCP Client and Server, while a PC will act as both the MQTT Publisher and Subscriber.

Configuration overview & prerequisites


- Two routers - **RUT1** as Modbus TCP Server (Slave) and **RUT2** as Modbus TCP Client (Master)
- Flespi.io account to act as an MQTT Broker/Publisher/Subscriber (for first configuration example)
- RUT241 with a Public IP address to act as MQTT Broker (for second configuration example)
- An end device (PC, Laptop) to act as MQTT Subscriber/Publisher (for second configuration example)

Configuration using flespi.io as MQTT Broker



Configuring Modbus TCP Server

Open RUT1 router's WebUI and navigate to **Services → Modbus → Modbus TCP Server**

1. Check **Enable**
2. Enter port that will be used, in this example 502 will be used
3. Enter device ID
4. Press 



**In this configuration LAN port is used hence "Allow Remote Access" is not needed*

Modbus TCP Server is now configured.

Configuring Modbus TCP Client


Open RUT2 router's WebUI and navigate to **Services → Modbus → Modbus TCP Client**

1. Press 



2. Check **Enable**
3. Enter the name for the Server device
4. Server ID must match with the previously configured Modbus TCP Server device ID
5. Enter the IP address of the Modbus TCP Server device
6. Chose same Port as in Server device - 502
7. Enter Period in seconds, how often requests will be sent to the Server device




2. Enter new request name and press  in Requests configurations




2. Choose "Data type" - **ASCII**, "Function" - **Read holding register (3)**, enter "First register number" - **72** and "Register count/Values" - **3** You can name each individual configuration, and then select enable on configurations that you want to use. In this example, register to get Routers name is used.
3. Press the "**Test**" button in Request Configuration Testing to see if the Server device responds to requests, a response similar to the image below should be shown.




2. Press  to save the configuration

List of available Modbus parameters can be found [\[here\]](#)

Configuring Flespi.io MQTT Broker

Log in or create an account on <https://flespi.io> 


Once logged in:

1. Navigate to MQTT Board on the left side of the screen and press it.
2. On the right-hand panel, top right corner, next to the name of the MQTT board, press the cogwheel-looking  icon to open *Connection Settings*
3. In the opened window, press "Get flespi token" to generate a username
4. Enter Client name
5. Copy Host address
6. Copy Username
7. Create a password
8. Press Save



Configuring MQTT Modbus Gateway

Open RUT2 router's WebUI and navigate to **Services → Modbus → MQTT Modbus Gateway**

1. Select Enable
2. Enter Host (copied from flespi connection settings without 'wss://' and port)
3. Select QoS (Quality of Service) level
4. Enter Username (Copied from flespi Connection settings generated token)
5. Enter Password
6. Press  to save the configuration



You can change Request and Response topics that you will have to publish and subscribe to get information from Modbus TCP Client through MQTT Gateway, but for this example, they are left on default topics

Message format for MQTT Publisher

The format is in the text - heavier and slower, but less difficult to edit.

1. Format version **0**
2. Cookie from **0** to **2⁶⁴ -1**
3. IP Type **0** - IPv4; **1** - IPv6; **2** - hostname
4. IP IPv6 must be in full format (for example:
2001:0db8:0000:0000:0000:8a2e:0370:7334)
5. Port Port number (for example: **502**)

6. Timeout in seconds (time to wait for response) from **1** to **999**
7. Server ID - Indicates to which Server request is sent from **1** to **255**
8. Function **3** - read registers; **6** - write single register; **16** - write multiple registers
9. Number of the first register from which information will be read or written from **1** to **65535**
- If function is 3 - from **1** to **123** (first register + registry value can not go out of range);
 If function is 6 - from **0** to **65535**
10. Registry value If function is 16 - from **1** to **123** (first register + registry value can not go out of range), Registry values separated by commas without spaces. Example.: **1,2,3,654,21,789**. There has to be as many values, as specified number of registers, and each value must be between **0** and **65535**. If number of registries is 0, there should be **no registry values**



Examples


Setting relay (on) (Relay address is 202, which means 'Number of first register will be 203) **0 65432 0 192.168.1.1 502 5 1 6 203 1**

Getting uptime **0 65432 0 192.168.1.1 502 5 1 3 2 2**

Testing MQTT Publisher and Subscriber on flespi.io


Log in and navigate to MQTT Board on <https://flespi.io>

Add a Subscriber:

1. Press  button on the top right corner
2. Select '**Subscriber**'
3. In the topic field enter '**response**'
4. Press '**Subscribe**' button



Add a Publisher:

1. Press  button on the top right corner
2. Select '**Publisher**'
3. In the topic field enter '**request**'
4. In the message field enter message, for this example '**Getting uptime**' is used
5. Press '**Publish**' button



Check the response in the '**Subscriber**' tab, you should receive a message similar to the one below.



When the value climbs over 65535 the counter resets and the value held by the first register increases by **1**. So one way to interpret the results would be to multiply the value in the first register by 2^{16} and add it to the value of the second register. In this example: $0 * 65536 + 31539 = 31539s$ or **0 days 8 hours 45 minutes and 39 seconds**.

This Means that MQTT Gateway on Modbus TCP Client router is working correctly and Modbus TCP Server receives requests.

Configuration using RUT241 router as MQTT Broker



The same configuration will be used for Modbus TCP Client and Server routers as in the previous example, only settings in Modbus TCP Client router will be changed to match MQTT Broker on RUT241 router

Configuring MQTT Modbus Gateway settings

Open RUT2 router's WebUI and navigate to **Services → MQTT Modbus Gateway**

1. **Enable**
2. Host: **Enter Public IP address of the new MQTT Broker (RUT241)**
3. Username: **N/A**
4. Password: **N/A**
5. Press



Configuring MQTT Broker settings on RUT241 device


Open RUT241 router's WebUI and navigate to **Services → MQTT → Broker**

1. Select **Enable**
2. Check **Enable Remote Access**



2. Scroll down to "**Broker settings**" and select "**Miscellaneous**" tab
3. Enable "**Allow anonymous**"



2. Press  to save the configuration

Configuring MQTT Publisher/Subscriber on PC & testing MQTT Gateway

For testing purposes, two terminal windows will be used on the same PC.

To get Ubuntu terminal on Windows 10/11, refer to the following link for instructions:
<https://tutorials.ubuntu.com/tutorial/tutorial-ubuntu-on-windows#0>

- Open first terminal, which will act as **Subscriber**. Type in:

```
mosquito_sub -h [Host_address] -p [Port] -t [Topic]
```



- Open second terminal, which will act as **Publisher**. Type in:

```
mosquito_pub -h [Host] -h [Host_address] -p [Port] -m ['Message'] -t [Topic]
```



- On the first window - **Subscriber**, a response should appear



See Also

- [Modbus Master RutOS configuration example](#)
- [Monitoring via Modbus](#)