https://wiki.teltonika-networks.com/view/Monitoring_via_MQTT

Monitoring via MQTT

Main Page > General Information > Configuration Examples > Router control and monitoring > **Monitoring via MQTT**

Router monitoring via MQTT Linux guide applies to TRB devices and to RUT, TCR routers.

□

Contents

Introduction

MQTT (MQ Telemetry Transport or Message Queue Telemetry Transport) is an ISO standard (ISO/IEC PRF 20922) publish-subscribe-based messaging protocol. It works on top of the TCP/IP protocol. It is designed for connections with remote locations where a "small code footprint" is required or the network bandwidth is limited. The publish-subscribe messaging pattern requires a message broker.

This article provides a guide on how to configure and use a basic MQTT setup on TRB devices and on RUT, TCR routers.

How MQTT works

In general an MQTT connection takes place between two Clients and a Broker. A TRB device and RUT/TCR router can be Broker, Client or both. The MQTT Publisher(Client) present in TRB devices and RUT/TCR routers subscribes to two topics by default: **<TYPE>/get** and **get/<SERIAL>/command**. Where **<TYPE>** is device type, for RUT/TCR routers type is **router** and for TRB devices type is **device**. Parameter **<SERIAL>** is the router's serial number. When a third party client connects to the Broker, it sends the message **id** to the the topic **<TYPE>/get**. The publisher then sends a response containing its serial number to the topic **<TYPE>/id**. Now that the Client knows the router's or device's serial number it can ask for values of various parameters by sending requests to the topic **<TYPE>/<SERIAL>/parameter_name**. The MQTT Publisher can send responses containing values of these system parameters, if device supports that parameter:

Parameter name	Parameter description	Supported devices
id	Modem IMEI	RUT2, RUT9, RUTX, RUT3, TCR1, TRB2, TRB5, OTD, RUT1, RUTM
temperature	Temperature of the module in 0.1 degrees Celsius	RUT2, RUT9, RUTX, RUT3, TCR1, TRB2, TRB5, OTD, RUT1, RUTM

operator	Current operator’s name	RUT2, RUT9, RUTX, RUT3, TCR1, TRB1, TRB2, TRB5, OTD, RUTM
signal	Signal strength in dBm	RUT2, RUT9, RUTX, RUT3, TCR1, TRB1, TRB2, TRB5, OTD, RUTM
network	Network state	RUT2, RUT9, RUTX, RUT3, TCR1, TRB1, TRB2, TRB5, OTD, RUTM
connection	Current connection type (2G, 3G, 4G)	RUT2, RUT9, RUTX, RUT3, TCR1, TRB1, TRB2, TRB5, OTD, RUTM
wan	WAN IP address	RUT2, RUT9, RUTX, RUT3, TCR1, TRB1, TRB2, TRB5, OTD, RUT1, RUTM
uptime	System uptime in seconds	RUT2, RUT9, RUTX, RUT3, TCR1, TRB1, TRB2, TRB5, OTD, RUT1, RUTM
name	Device's device code	RUT2, RUT9, RUTX, RUT3, TCR1, TRB1, TRB2, TRB5, OTD, RUT1, RUTM
digital1	Value of digital input no. 1	RUT9
digital2	Value of digital input no. 2	RUT9
analog	Value of analog	RUT9, TRB2, TRB141
pin2	Value of 2's pin state	TRB2
pin3	Value of 3's pin state	RUT2, RUT9, RUTX, RUT3, TRB1, TRB2, TRB5, RUTM
pin4	Value of 4's pin state	RUT2, RUT9, RUTX, RUT3, TRB1, TRB2, TRB5, RUTM

After the client sends a message containing of these parameters, the Publisher will send a response message containing the value of the requested parameter to the topic **<TYPE>/<SERIAL>/parameter_name**, where **parameter_name** is the name of the requested parameter. The scheme below shows example of how to get parameters on RUT routers.

It should also be noted that, according to the MQTT protocol, topic names are case-sensitive. For instance, topic *router* is not the same as topic *RoUtEr*.

Configuring the device

This section will provide an explanation on how to configure an MQTT Broker and MQTT Publisher on a TRB devices and on RUT/TCR routers.

Broker

Basic (not counting Security, Bridge and Miscellaneous Broker settings) Broker configuration only contains three fields. Make sure to **Enable** the Broker and specify a **Port** for connection to the Broker. **Enable Remote Access** if you wish to connect to your router remotely (via WAN). Although in this case your router would need to have a Public IP address. In image below is presented how to configure RUT9xx routers:

Configuration for a TRB device and a RUT/TCR router looks like this:

Publisher

Publisher configuration is similar in its simplicity to the Broker configuration. Make sure to **Enable** the Publisher and specify a **Port** for connection to a Broker. If you plan on using the router's Broker, specify the same port as the one in the Broker Settings section. **Hostname** is the Broker's host name or IP address. If you're using the router's Broker, specify your router's LAN IP address. **Username** and **Password** are optional and depend on the type of security (if any) that the Broker uses. In image below is presented how configure RUT9xx routers:

Configuration for a TRB device and a RUT/TCR router looks like this:

NOTE: If you are configuring a TRB device, default hostname is 192.168.2.1.

Configuring the PC

Once the Broker is up, you'll need to install **Mosquitto** and **Mosquitto Clients** on your PC. To do so, open the Linux **Terminal** app and enter this command:

```
$ sudo apt-get install mosquitto mosquitto-clients
```

NOTE: this software was chosen because it suits the needs of this example and is easily accessible to most users. It is not the only application compatible with RUT MQTT, so if you're using your own software for your solution, it is not mandatory to install this.

Subscribing and Publishing

Now you can use set up Brokers and Clients on your PC. In order to Publish commands and Subscribe to topics on a TRB series devices and RUT/TCR routers, you'll need to know the device's **Serial Number** and **TYPE**. For TRB series device's you'll need to use type : **device** and for RUT/TCR router's you'll need to use type: **router**. Router's serial number you can find in the **Status → Device** section of the router's WebUI. For TRB series devices serial number you can find in the **Status → System** section of the device WebUI. Or you can subscribe to the topic **<TYPE>/id** and publish the message **id** to the topic **<TYPE>/get**. To do so, enter these commands in separate Terminal windows:

Subscribe:

```
$ mosquitto_sub -h 192.168.1.1 -p 1833 -u user -P pass -t <TYPE>/id
```

Publish:

```
$ mosquitto_pub -h 192.168.1.1 -p 1833 -u user -P pass -t <TYPE>/get -m id
```

In image below is presented example of how to get RUT router serial number:

mosquitto_sub command is used to subscribe to a certain topic(s) and **mosquitto_pub** command is used to publish messages to specified topics. **-m** defines the message that you're publishing

-h indicates the host (the Broker's IP address) which, continuing from the example above, is **192.168.1.1**. Replace this value with your router's LAN IP address or, if you're using MQTT remotely, your WAN IP address.

-p indicates the port used for connection to the Broker. Replace it with the port number that you specified in the Broker configuration.

-u and **-P** specify the username and password. If the Broker doesn't require authentication, these values aren't mandatory. Else replace them with the log in information of your Broker.

-t specifies the **topics** that you subscribe or publish to.

Now that we have the using device's serial number we can start publishing messages and receiving responses containing the device's system parameter values. Lets say we want to monitor the device's **signal strength** and **up time** values. In order to do that we'll need to subscribe to the topics **router/<SERIAL>/signal** and **router/<SERIAL>/uptime** using the **mosquitto_sub** command. Again, replace the values given in the commands below with the ones in your configuration:

Subscribe:

```
$ mosquitto_sub -h 192.168.1.1 -p 1833 -u user -P pass -t
router/<SERIAL>/signal -t router/<SERIAL>/uptime
```

Publish:

```
$ mosquitto_pub -h 192.168.1.1 -p 1833 -u user -P pass -t router/get -m
signal
$ mosquitto_pub -h 192.168.1.1 -p 1833 -u user -P pass -t router/get -m
uptime
```

In example below you can see how to subscribe and publish to the topics **signal strength** and **uptime** on RUT/TCR routers:

External links

https://mosquitto.org/ - Eclipse Mosquitto official website