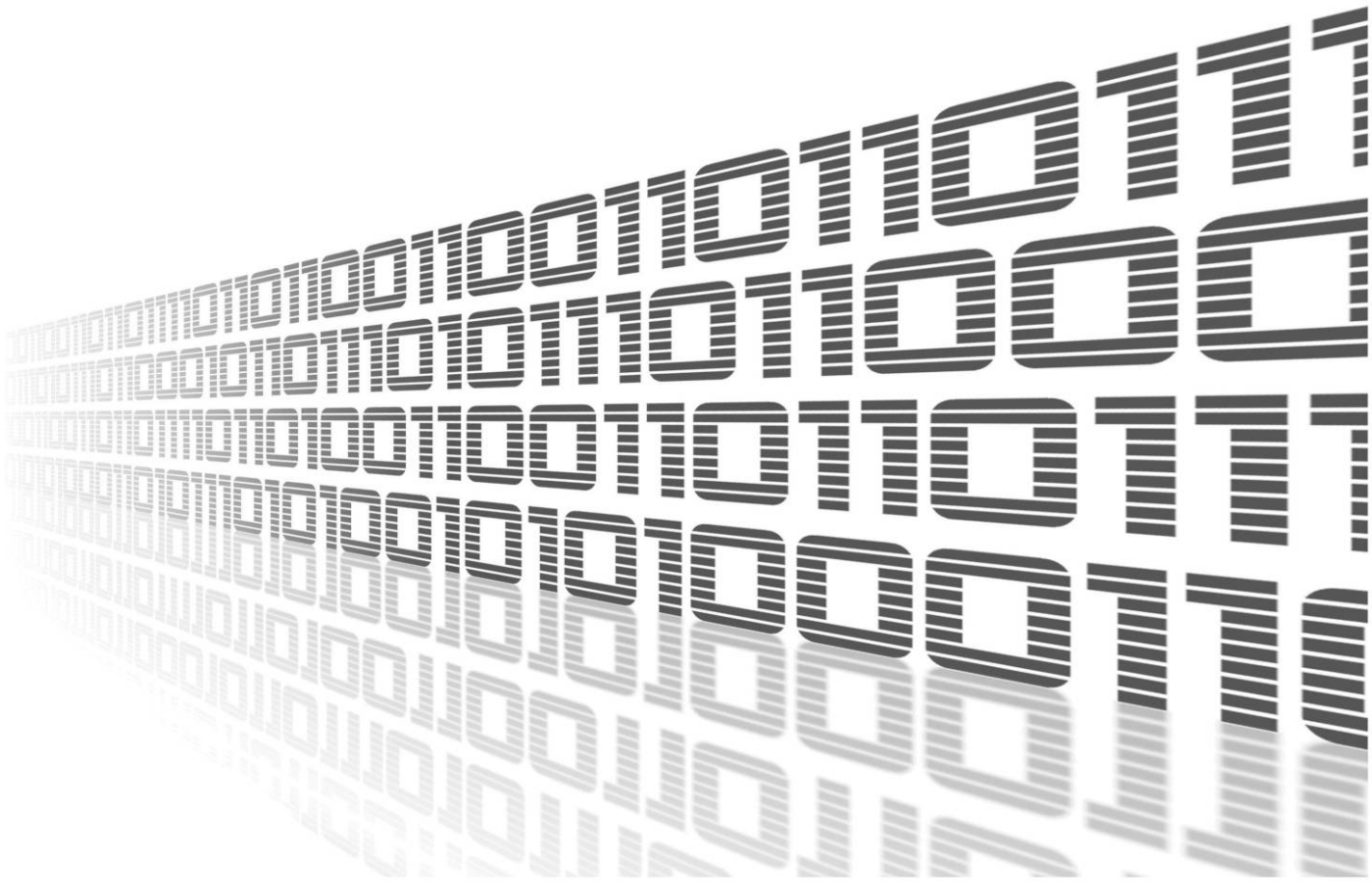
ADVANTECH

RouterApp

MQTT Manager



Advantech Czech s.r.o., Sokolska 71, 562 04 Usti nad Orlici, Czech Republic
Document No. APP-0110-EN, revised on April 3, 2025.

Used symbols

Danger – Information regarding user safety or potential damage to the router.

Attention – Problems that can arise in specific situations.

Information – Useful tips or information of special interest.

Contents

List of Figures

List of Tables

1. Description of the module

This Router App is not included in the standard router firmware. Uploading of this router app is described in the *Configuration manual* (see Chapter Related Documents).

The *MQTT Manager* is a Router App designed for controlling and monitoring Advantech routers via the MQTT protocol. Users can issue commands through specific MQTT messages to, for instance, modify binary outputs. Additionally, users can track the status of the router by subscribing to a specific topic.

2. Web Interface

Once the installation of the module is complete, the module's GUI can be invoked by clicking the module name on the Router apps page of router's web interface.

The left part of this GUI contains the menu with General menu section. Return menu item switches back from the module's web page to the router's web configuration pages. The main menu of the module's GUI is shown on Figure 1.

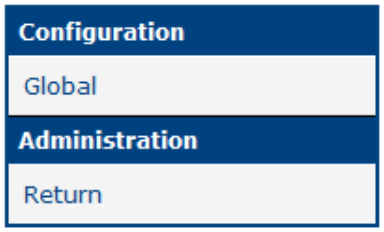


Figure 1: Menu

2.1 Global Configuration

Configuration of this router app can be done on the Settings page, under the Router menu section. All configuration items for the Settings configuration page are described in the table below.

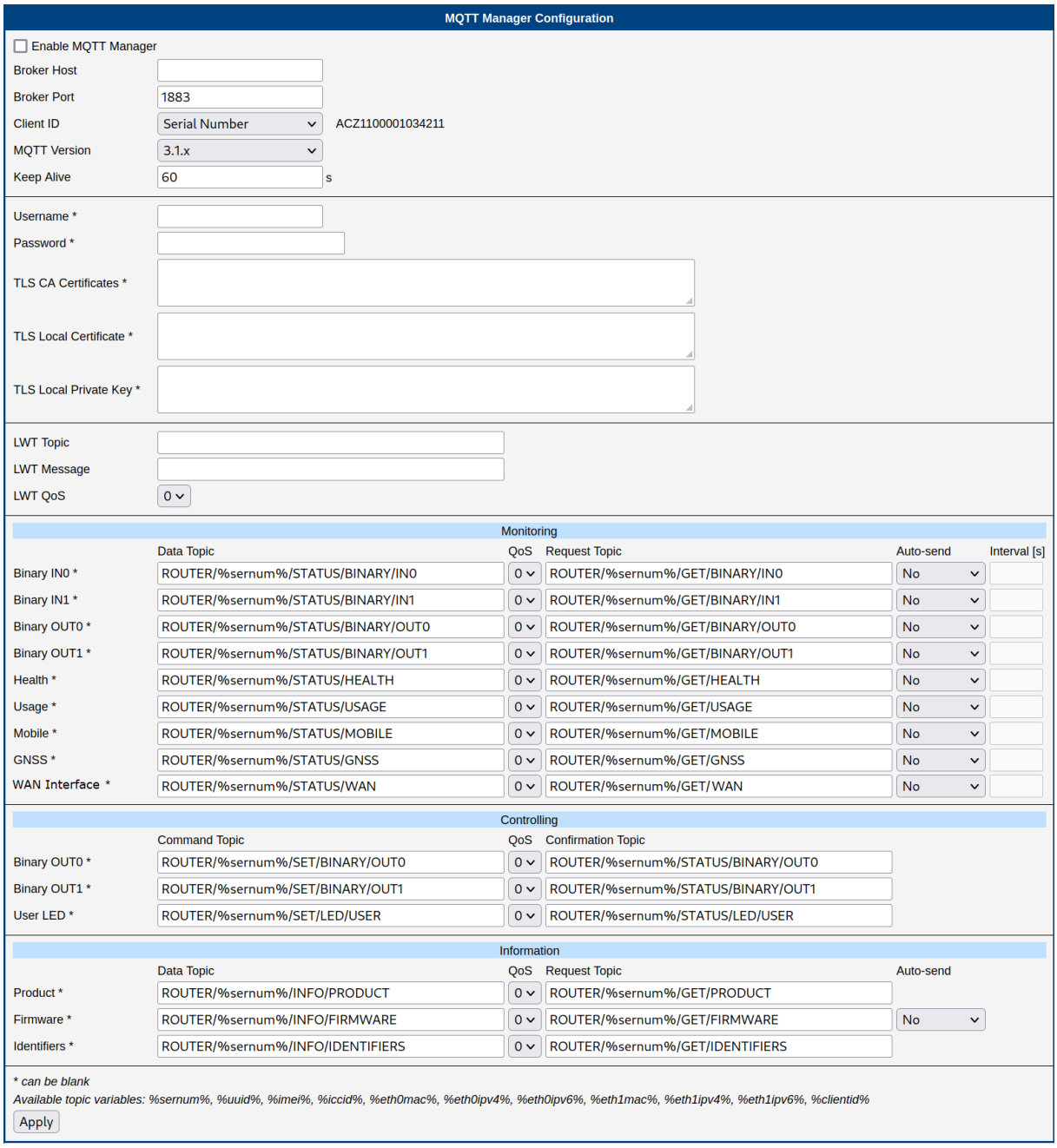


Figure 2: Configuration

Item	Description
Enable MQTT Manager	Enabled, MQTT Manager functionality of the module is turned on.
Broker Host	Enter IP address or domain name of MQTT broker.
Broker Port	Enter Broker Server Port Number (1-65535).
Client ID	String used as the client identifier to the broker. You can choose from the router unique IDs or insert your own.
MQTT Version	Select MQTT version.
Keep Alive	Enter MQTT Keep-Alive interval (1-3600).
Username	Enter username to connect to the broker if it is required.
Password	Enter password to connect to the broker if it is required.
TLS CA Certificates	If you use TLS connection, enter Certificate Authority certificate.
TLS Local Certificate	If you use TLS connection, enter the router local certificate.
TLS Local Private Key	If you use TLS connection, enter the router local private key.
LWT Topic	Enter topic of the message that will broker send to connected subscribers, when one of the routers will disconnect or stop to communicate. (so called Last Will and Testament feature)
LWT Message	Enter content of the message that will broker send to connected subscribers, when one of the routers will disconnect or stop to communicate. (so called Last Will and Testament feature)
LWT QoS	Select Quality of Service level of the message that will broker send to connected subscribers, when one of the routers will disconnect or stop to communicate. (so called Last Will and Testament feature)
Monitoring	See Chapter 2.1.1
Controlling	See Chapter 2.1.2
Information	See Chapter 2.1.3

Table 1: Settings Example Items Description

2.1.1 Monitoring

These messages encapsulate the value of some data change. Such a message will be dispatched to the user upon request using a *special message*, refer to Chapter 3 for more details. Additionally, it can be sent upon any change in value or at predetermined intervals. When dispatched due to a data change, it is important to recognize that the state is sampled at a specific frequency, meaning very rapid changes may go undetected (for example, if the state of a binary input fluctuates for only a millisecond). Furthermore, users have the ability to customize both the topic of the message being sent (Data Topic) and the topic of the message they wish to request (Request Topic) for each piece of monitored data.

By using the following command, you can receive all data sent by the manager (you must be connected to the local broker):

```
mosquitto_sub -t "ROUTER/#"
```

2.1.2 Controlling

These messages enable the user to control the router. For instance, the user LED can be toggled on or off by dispatching the appropriate message. "Command Topic" refers to the topic of the message the user sends to the router, while "Confirm Topic" is the topic of the message the router sends back to confirm the execution of the command.

2.1.3 Information

These messages are akin to Monitoring messages but pertain to static data that does not change (e.g., product name), eliminating the need for repeated transmission. They are dispatched solely upon request. An exception is the firmware version, which alters during updates and thus can be configured to be sent automatically. Request is an empty MQTT message sent to a given Request Topic.

For all topics, messages can be disabled by leaving the topic field empty. In such cases, the corresponding message will not be sent. For instance, if the user leaves the "Data Topic" field blank for "Binary IN0," this information will not be transmitted, and the user cannot request it; it is essentially blocked. Similarly, if "Confirmation Topic" is left unfilled in the Controlling section, the user can issue a command that will be executed, but they will not receive a confirmation of the successful outcome.

Here is an example of how to retrieve the value of IN0 (the first binary input) on a router:

```
mosquitto_pub -t "ROUTER/ACZ1100001019000/GET/BINARY/IN0" -m ""
```

Variable	Example
%sernum%	Will be replaced by device serial number. Example: "ACZ1199000000652"
%suuid%	Will be replaced by router UUID. Example: "a13cf7db-810f-4cec-afa8-bcbda3c285a8"
%imei%	Will be replaced by IMEI number. Example: "861536030196001"
%iccid%	Will be replaced by identification number of SIM card. Example: "89442001023880434 68"
%eth0mac%	Will be replaced by eth0 MAC address. Example: "02:AD:FF:00:00:65"
%eth0ipv4%	Will be replaced by eth0 IPv4 address. Example: "192.168.1.1"
%eth0ipv6%	Will be replaced by eth0 IPv6 address. Example: "fd00:a40::65"
%eht1mac%	Will be replaced by eth1 MAC address. Example: "02:AD:FF:01:00:65"
%eth1ipv4%	Will be replaced by eth1 IPv4 address. Example: "192.168.1.2"
%eth1ipv6%	Will be replaced by eth1 IPv6 address. Example: "fd00:a41::65"
%clientid%	Will be replaced by client ID of the router.

Table 2: Variables Description

3. Format of Messages

This section describes and includes examples of the MQTT messages used in communication between the MQTT Manager and customer's MQTT client.

3.1 Monitoring Examples

Binary IN0

- *Data message:*
  `{"binary": {"in0": B}}`

  Where *B* is binary value 0 or 1.

  Example:
  `{"binary": {"in0": 1}}`

- *Request message:*
  Empty message, choose the correct Request Topic as defined in the settings.

Binary IN1

- *Data message:*
  `{"binary": {"in1": B}}`

  Where *B* is binary value 0 or 1.

  Example:
  `{"binary": {"in1": 1}}`

- *Request message:*
  Empty message, choose the correct Request Topic as defined in the settings.

Binary OUT0

- *Data message:*
  `{"binary": {"out0": B}}`

  Where *B* is binary value 0 or 1.

  Example:
  `{"binary": {"out0": 1}}`

- *Request message:*
  Empty message, choose the correct Request Topic as defined in the settings.

Binary OUT1

- *Data message:*
  `{"binary": {"out1": B}}`

  Where *B* is binary value 0 or 1.

  Example:
  `{"binary": {"out1": 1}}`

- *Request message:*
  Empty message, choose the correct Request Topic as defined in the settings.

Health

- *Data message:*
  `{"health": {"temperature": T, "voltage": V, "battery": B}}`

  Where *T* is the temperature in Celsius degrees as an integer, *V* is the current supply voltage in Volts as a decimal number, and *B* is the RTC battery status as "OK" or "Empty".

  Not all routers have all the information available, for example v2i type of routers does not communicate the temperature. In this case, the relevant information is skipped.

  Example:
  `{"health": {"temperature": 36, "voltage": 11.7, "battery": "OK"}}`

- *Request message:*
  Empty message, choose the correct Request Topic as defined in the settings.

Usage

- *Data message:*
  `{"usage": {"cpu": C, "ram": {"total": T, "used": U, "percentage": P}}}`

  Where *C* is the CPU usage in percentage as a decimal number, *T* is the total amount of RAM in bytes as an integer, U is the used RAM in bytes as an integer, and *P* is the used RAM in percent as a decimal number.

  Example:
  `{"usage": {"cpu": 10.3, "ram": {"total": 521654272, "used": 28209152, "percentage": 5.4}}}`

- *Request message:*
  Empty message, choose the correct Request Topic as defined in the settings.

Mobile

- *Data message:*
  ```
  {"mobile": [{"sim": S, "technology": T, "operator": O, "plmn": P,
  "cell": C, "lac": L, "channel": H, "signal": {"strength": R, "quality": Q},
  "uptime": U, "ipv4: I4," ipv6 ", I6}, ...]}
  ```
  Where *S* is the number of the currently selected SIM cards 1 or 2, *T* is mobile technology as a string, *O* is operator as a string, *P* is PLMN as an integer, *C* is a cell number as an integer, *H* is a channel number as an integer, *R* is a signal strength in dBm as an integer, *Q* is the signal quality in dB as an integer, *U* is the uptime connection as a string, *I4* is the IPv4 address as a string, and *I6* is the IPv6 address as a string. "*Mobile*" is an array due to multiple cellular modules. With Smart Motion, there will be two items in the field. The "*sim*" data should always be present, other data only when the connection is active. IP addresses do not have to be set both.

  Example:
  ```
  {"mobile": [{"sim": 1, "technology": "LTE", "operator": "T-Mobile",
  "plmn": 23003,"cell": 10804, "lac": 947, "channel": 1849, "signal": {"strength": -91,
  "quality": -6}, "uptime": "0 days, 11 hours, 18 minutes","ipv4": "10.80.0.27"}]}
  ```

- *Request message:*
  Empty message, choose the correct Request Topic as defined in the settings.

3.2 Controlling Examples

Binary OUT0

- *Controlling message:*
  `{"binary": {"out0": B}}`

  Where *B* is binary value 0 or 1.

  Example:
  `{"binary": {"out0": 1}}`

- *Confirmation message:*
  It's the same, but with a different topic.

Binary OUT1

- *Controlling message:*
  `{"binary": {"out1": B}}`

  Where *B* is binary value 0 or 1.

  Example:
  `{"binary": {"out1": 1}}`

- *Confirmation message:*
  It's the same, but with a different topic.

User LED

- *Controlling message:*
  `{"led": {"user": B}}`

  Where *B* is binary value 0 or 1.

  Example:
  `{"led": {"user": 1}}`

- *Confirmation message:*
  It's the same, but with a different topic.

3.3 Information Examples

Product

- *Data message:*
  `{"info": {"series": S, "partnum": P}}`

  Where *S* is the model line as a string and *P* is the product number as a string.

  Example:
  `{"info": {"series": "ICR-324x", "partnum": "ICR-3241W"}}`

- *Request message:*
  Empty message, choose the correct Request Topic as defined in the settings.

Firmware

- *Data message:*
  `{"info": {"firmware": F}}`

  Where *F* is firmware version.

  Example:
  `{"info": {"firmware": "6.3.2 (2021-09-30)"}}`

- *Request message:*
  Empty message, choose the correct Request Topic as defined in the settings.

Identifiers

- *Data message:*
  `{"info": {"sernum": S, "uuid": U, "module": [{"imei": I, "iccid": C, "esn": E, "meid": M}, ...]}}`

  Where *S* is the serial number and *U* is the universally unique identifier. *I*, *C*, *E*, and *M* are identifiers used in mobile networks. All items are strings. If the router has two cellular modules, the module array will contain two entries, if it has no cellular module, module information will be omitted. Also *UUID* may not be presented.

  Example:
  `{"info": {"sernum": "ACZ1199000000736", "module": [{"imei": "863305040213851", "iccid": "89420310201221227 79" }]}}`

- *Request message:*
  Empty message, choose the correct Request Topic as defined in the settings.

4. Related Documents

You can obtain product-related documents on *Engineering Portal* at *icr.advantech.com* address.

To get your router's *Quick Start Guide*, *User Manual*, *Configuration Manual*, or *Firmware* go to the *Router Models* page, find the required model, and switch to the *Manuals* or *Firmware* tab, respectively.

The *Router Apps* installation packages and manuals are available on the *Router Apps* page.

For the *Development Documents*, go to the *Development* page.