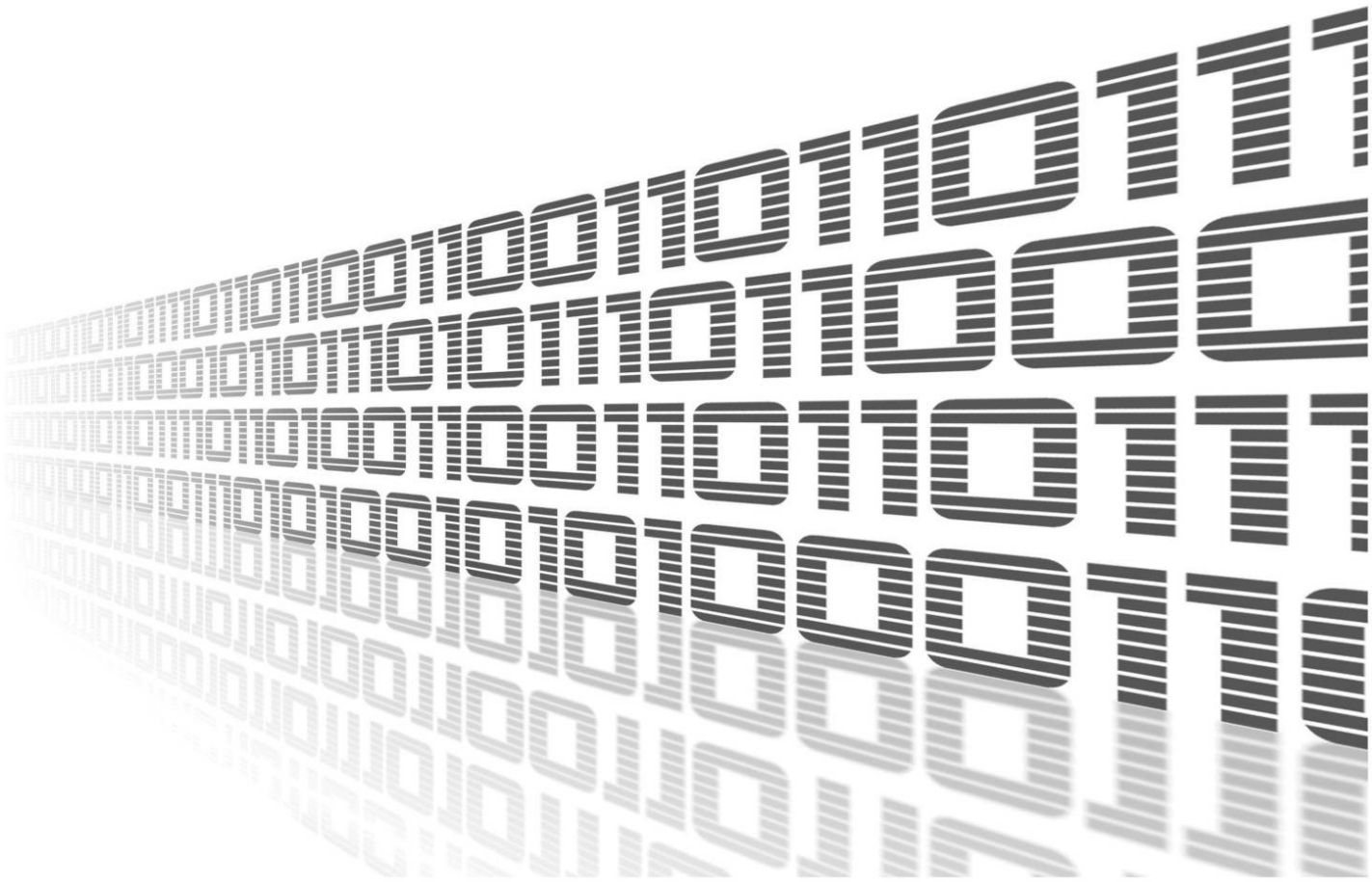
ADVANTECH

RouterApp

Modbus to MQTT



Advantech Czech s.r.o., Sokolska 71, 562 04 Usti nad Orlici, Czech Republic
Document No. APP-0087-EN, revised on April 16, 2026.

Used symbols

Important

Important — Indicates a risk to personal safety or potential damage to the router. Follow these instructions precisely to prevent injury or equipment damage.

Warning

Warning — Highlights conditions that may cause malfunction, loss of data, or unexpected behavior in specific situations. Read carefully before proceeding.

Info

Info — Provides helpful tips, context, or references that improve understanding but are not strictly required to complete the task.

Contents

List of Figures

List of Tables

1. Router App Introduction

Warning

- This router app is not included in the standard router firmware. Instructions for uploading it can be found in the *Configuration Manual*.
- This router app is compatible only with firmware version 6.4.x or higher.

1.1 Function and Scope

The *Modbus to MQTT* router app facilitates seamless communication between Modbus TCP devices and an MQTT broker. The app operates in two modes:

- **Modbus TCP Master:** Communicates with field devices (PLCs, sensors).
- **MQTT Client:** Acts as a publisher or subscriber to exchange data with a broker.

This architecture enables bidirectional data exchange between industrial field devices and cloud or IT systems using the MQTT protocol.

1.2 Web Interface

After installing the module, its graphical user interface (GUI) can be accessed by selecting the module name on the *Router Apps* page within the router's web interface.

The left side of the GUI contains a navigation menu with three main sections:

- **Status** – Includes *Log* and *Mapping Table*.
- **Configuration** – Includes the *Global Settings* and *Direct Methods Settings*.
- **Customization** – Includes the *Return* item to switch back to the main router configuration.

The main menu of the module's GUI is illustrated on the figure below.

Figure 1: Navigation menu

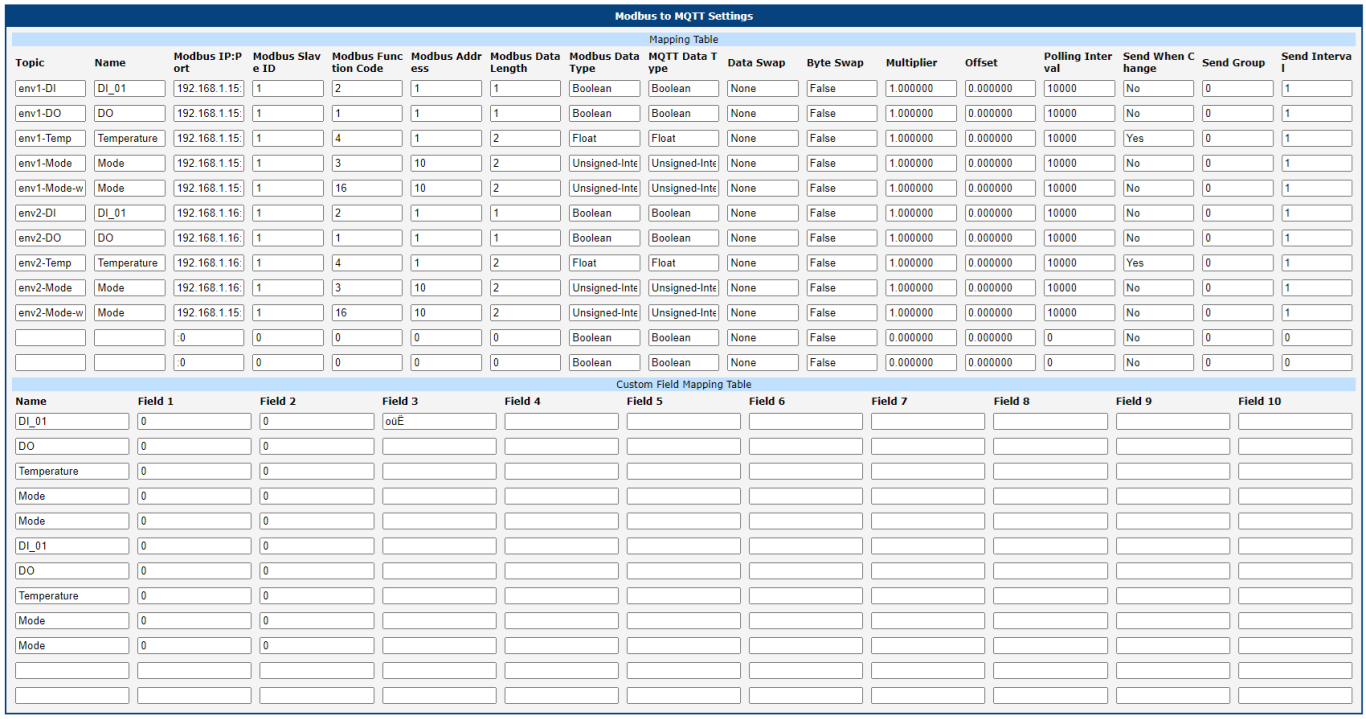
2. Status

2.1 Log

This section displays log messages generated by the router app, including connection status and error reports.

2.2 Mapping Table

Once configured, the active Modbus-to-MQTT mapping rules are displayed in the *Mapping Table*.



Modbus to MQTT Settings

Mapping Table

Topic	Name	Modbus IP:Port	Modbus Slave ID	Modbus Function Code	Modbus Address	Modbus Data Length	Modbus Data Type	MQTT Data Type	Data Swap	Byte Swap	Multiplier	Offset	Polling Interval	Send When Change	Send Group	Send Interval
env1-DI	DI_01	192.168.1.15:	1	2	1	1	Boolean	Boolean	None	False	1.000000	0.000000	10000	No	0	1
env1-DO	DO	192.168.1.15:	1	1	1	1	Boolean	Boolean	None	False	1.000000	0.000000	10000	No	0	1
env1-Temp	Temperature	192.168.1.15:	1	4	1	2	Float	Float	None	False	1.000000	0.000000	10000	Yes	0	1
env1-Mode	Mode	192.168.1.15:	1	3	10	2	Unsigned-Inte	Unsigned-Inte	None	False	1.000000	0.000000	10000	No	0	1
env1-Mode-w	Mode	192.168.1.15:	1	16	10	2	Unsigned-Inte	Unsigned-Inte	None	False	1.000000	0.000000	10000	No	0	1
env2-DI	DI_01	192.168.1.16:	1	2	1	1	Boolean	Boolean	None	False	1.000000	0.000000	10000	No	0	1
env2-DO	DO	192.168.1.16:	1	1	1	1	Boolean	Boolean	None	False	1.000000	0.000000	10000	No	0	1
env2-Temp	Temperature	192.168.1.16:	1	4	1	2	Float	Float	None	False	1.000000	0.000000	10000	Yes	0	1
env2-Mode	Mode	192.168.1.16:	1	3	10	2	Unsigned-Inte	Unsigned-Inte	None	False	1.000000	0.000000	10000	No	0	1
env2-Mode-w	Mode	192.168.1.15:	1	16	10	2	Unsigned-Inte	Unsigned-Inte	None	False	1.000000	0.000000	10000	No	0	1
		0	0	0	0	0	Boolean	Boolean	None	False	0.000000	0.000000	0	No	0	0
		0	0	0	0	0	Boolean	Boolean	None	False	0.000000	0.000000	0	No	0	0

Custom Field Mapping Table

Name	Field 1	Field 2	Field 3	Field 4	Field 5	Field 6	Field 7	Field 8	Field 9	Field 10
DI_01	0	0	oúÊ							
DO	0	0								
Temperature	0	0								
Mode	0	0								
Mode	0	0								
DI_01	0	0								
DO	0	0								
Temperature	0	0								
Mode	0	0								
Mode	0	0								

Figure 2: Mapping table

3. Configuration

3.1 Global Settings

The main configuration for this Router App is performed on the *Global Settings* page, located under the *Configuration* menu section. All configuration parameters available on this page are shown in the figure and described in the tables below.

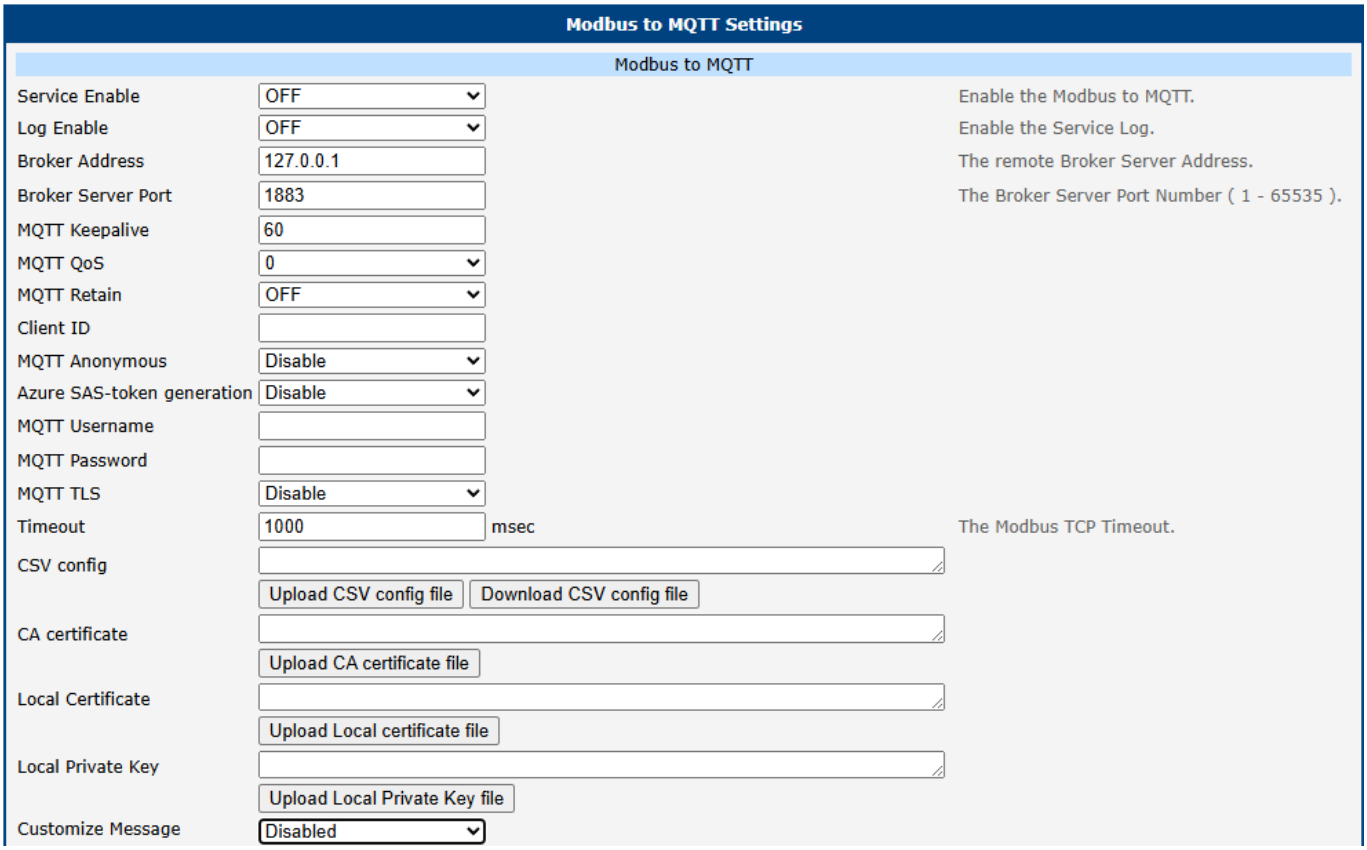


Figure 3: Global settings part 1

Item	Description
Service Enable	Enables or disables the *Modbus to MQTT* service.
Log Enable	Enables logging for this router app.
Broker Address	IP address or hostname of the remote MQTT broker.
Broker Server Port	MQTT broker TCP port number (standard is `1883`).
MQTT Keepalive	Keepalive interval in seconds (`1` – `3600`).
MQTT QoS	MQTT Quality of Service level (`0`, `1`, or `2`).
MQTT Retain	Enables retained messages on the MQTT broker.
Client ID	MQTT client identifier used by the router.
MQTT Anonymous	Enables anonymous connection without authentication.
MQTT Username	Username for authentication.
MQTT Password	Password for authentication.

Table 1: Global settings items description

Item	Description
MQTT TLS	Enables encrypted communication using TLS.
Interval (ms)	Modbus TCP polling interval in milliseconds.
Timeout (ms)	Modbus TCP communication timeout in milliseconds.
CSV Config	Upload field for the CSV configuration file. **Note:** To ensure the imported CSV configuration file is displayed correctly, a router restart may be required.
CA Certificate	Upload the CA certificate for TLS validation.
Local Certificate	Upload the local client certificate for mutual TLS.
Local Private Key	Upload the private key for the local certificate.
Customize Messages	Enables user-defined customization of the MQTT payload structure.

Table 1: Global settings items description (continued)

The second part of the settings allows configuration of the MQTT payload. The default payload includes variables such as **Topic**, **Name**, **Value**, **Time**, **IP**, **Port**, **ID**, **FC**, **Address**, and **Data Length**.

Additionally, up to ten custom fields can be defined to include static metadata (e.g., location or device type) in every JSON message.

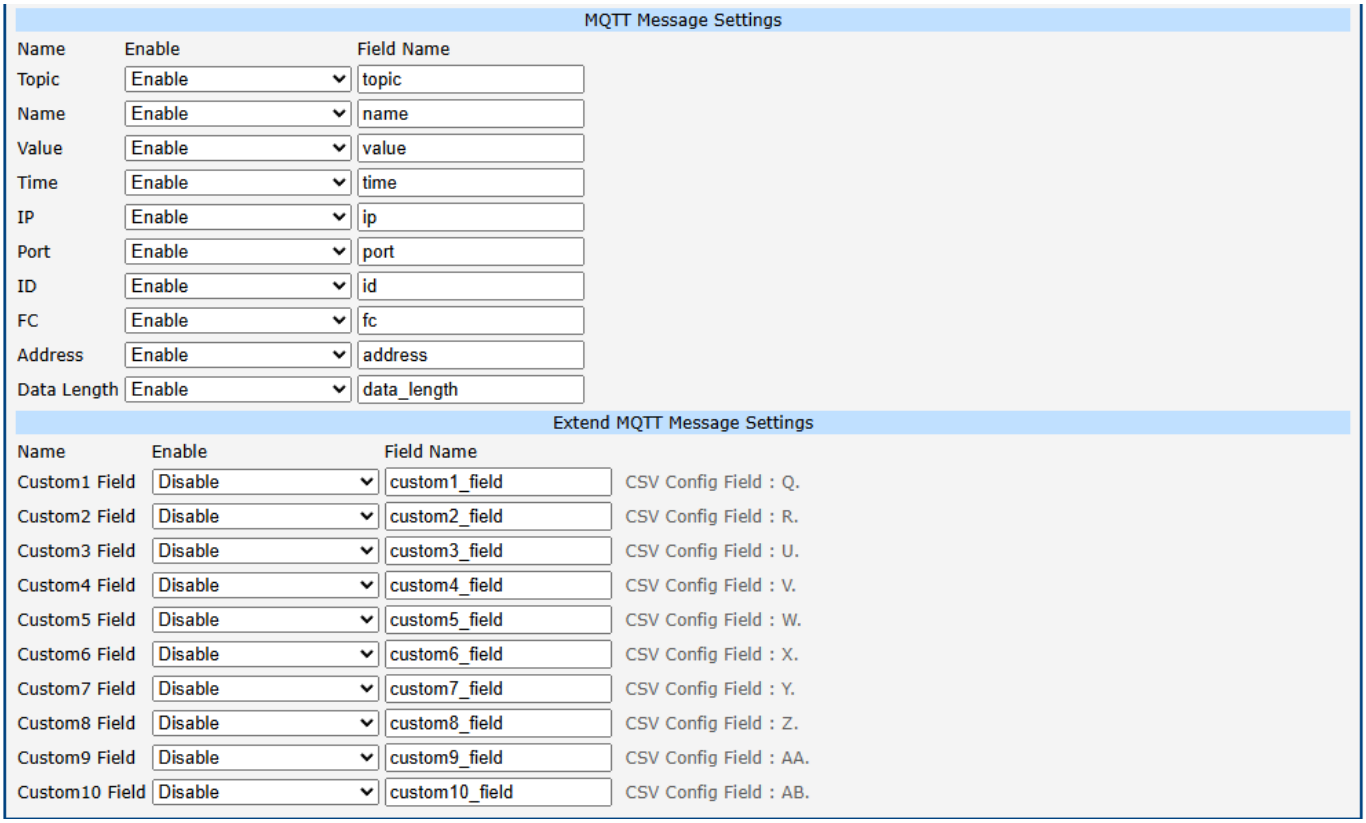


Figure 4: Global settings part 2

3.2 Direct Methods Settings

The *Direct Methods Settings* page configures how the *Modbus to MQTT* Router App processes commands received via MQTT and forwards them as Modbus write operations to connected Modbus/TCP devices.

This function is typically used when an external system (for example, an IoT platform, SCADA system, or cloud application) sends control commands via MQTT. The Router App receives the MQTT message, processes it using a user-defined script, and then translates it into a corresponding Modbus write request (typically using function codes 5, 6, 15, or 16).

Each configuration is defined as an independent rule, identified by a *Rule Index*, which allows multiple Direct Method behaviors to be configured in parallel.

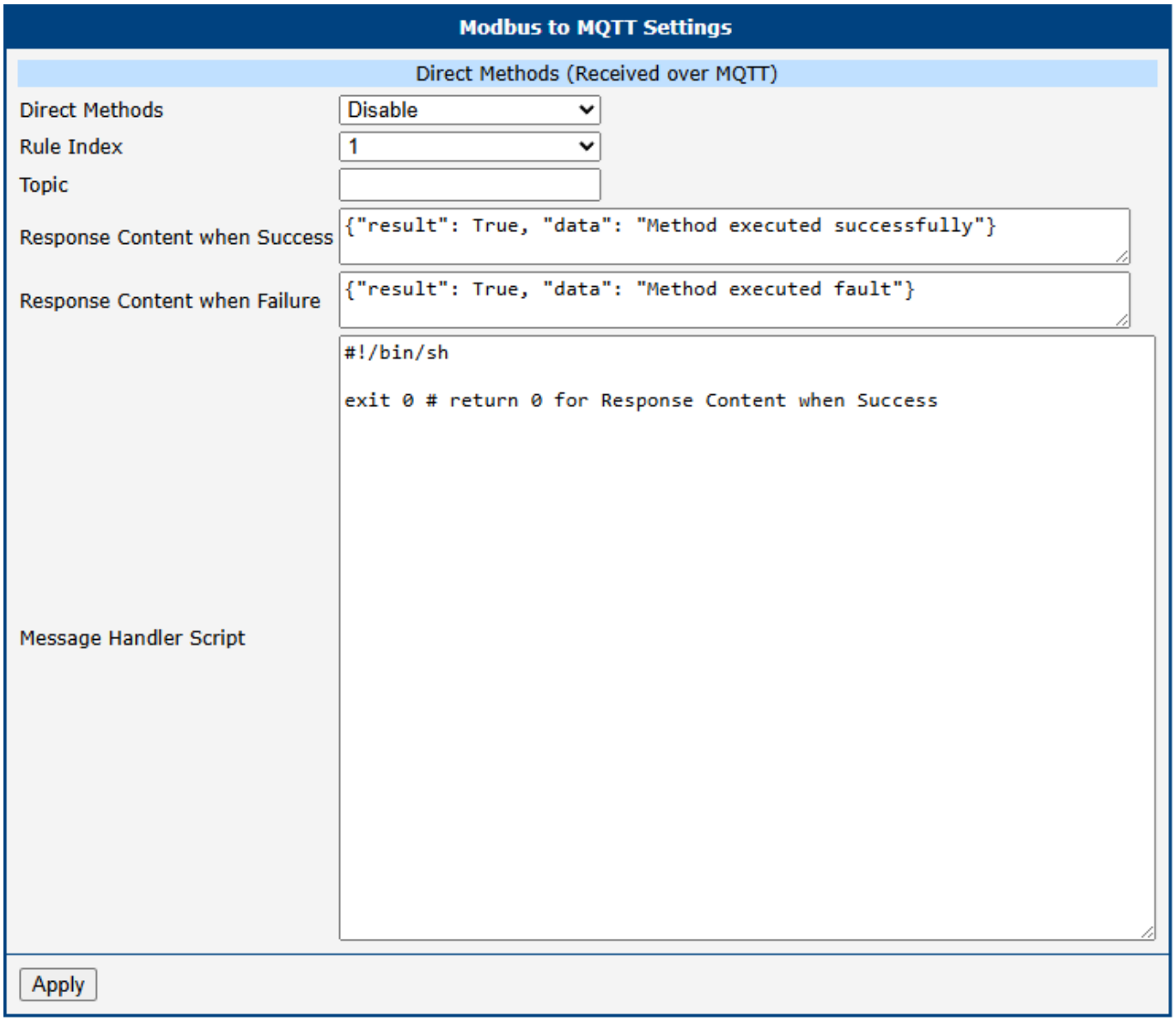


Figure 5: Direct methods settings

The Direct Methods Settings page contains the following configuration items:

Item	Description
Direct Methods	Enables or disables the selected Direct Method rule. When disabled, the rule is ignored by the Router App.
Rule Index	Selects which Direct Method rule is currently being configured. Multiple rules can be defined to handle different MQTT topics or control behaviors.
Topic	Specifies the MQTT topic that the Router App subscribes to for receiving control messages. Any message published to this topic is processed by the corresponding rule.
Response Content when Success	Defines the JSON message returned to the MQTT broker when the command is processed successfully. This response is sent when the handler script exits with code 0.
Response Content when Failure	Defines the JSON message returned to the MQTT broker when the command processing fails. This response is sent when the handler script exits with a non-zero code.
Message Handler Script	A user-editable shell script that processes incoming MQTT messages and performs the corresponding Modbus actions. The script can parse the MQTT payload, extract parameters, execute Modbus write commands, and determine success or failure based on its exit code.

Table 2: Direct Methods Settings Items Description

Processing Workflow

1. An external system publishes a command to the configured *Topic*.
2. The router passes the payload to the *Message Handler Script*.
3. The script interprets the data and executes a Modbus write command.
4. Based on the script's exit code, the router returns the *Success* or *Failure* response to the broker.

3.3 Peripheral Ports

Info

The configuration available on this page depends on the specific hardware model of the router.

This page allows configuration of physical interfaces, including serial ports and USB ports.

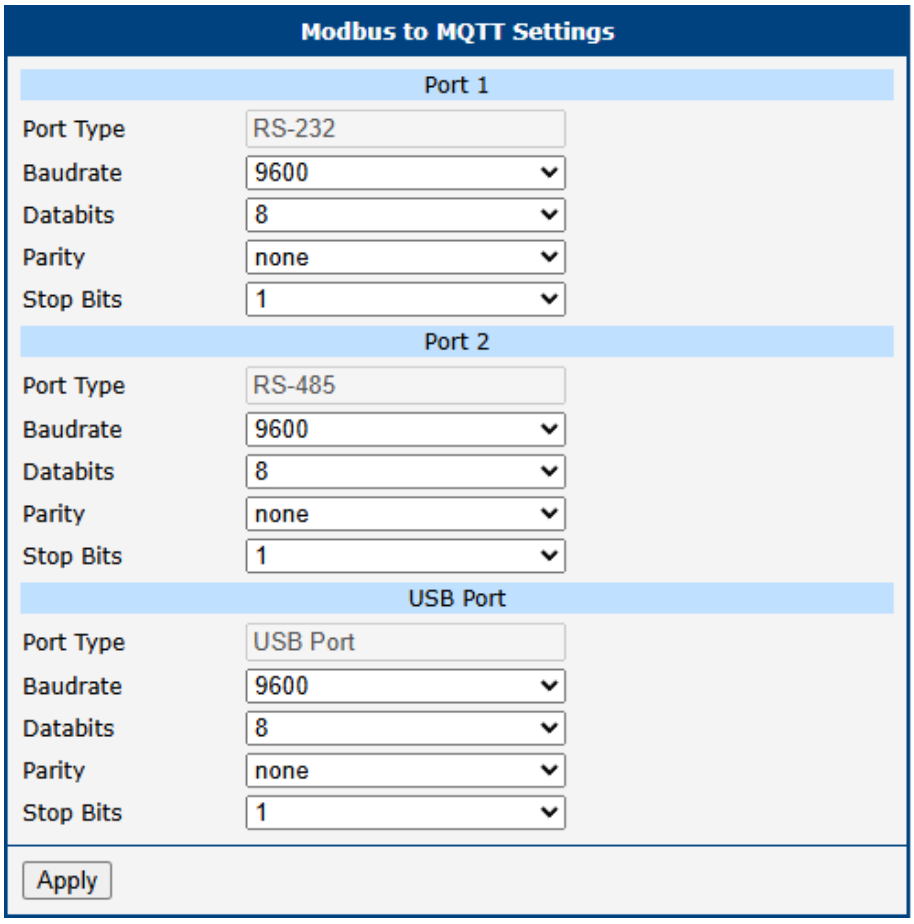


Figure 6: Peripheral ports

4. Data Mapping and Messaging

4.1 CSV Configuration File

The mapping between Modbus TCP and MQTT is defined in a CSV file (comma-delimited).

Topic	Name	IP	Port	Device ID	Function Code	Address	Data length	Modbus Data type	Data Swap	Byte Swap
env1-DI	DI_01	192.168.1.15	502	1	2	1	1	Boolean	None	False
env1-DO	DO	192.168.1.15	502	1	1	1	1	Boolean	None	False
env1-Temp	Temperature	192.168.1.15	502	1	4	1	2	Float	None	False
env1-Mode	Mode	192.168.1.15	502	1	3	10	2	Unsigned Integer	None	False
env1-Mode-w	Mode	192.168.1.15	502	1	16	10	2	Unsigned Integer	None	False
env2-DI	DI_01	192.168.1.16	502	1	2	1	1	Boolean	None	False
env2-DO	DO	192.168.1.16	502	1	1	1	1	Boolean	None	False
env2-Temp	Temperature	192.168.1.16	502	1	4	1	2	Float	None	False
env2-Mode	Mode	192.168.1.16	502	1	3	10	2	Unsigned Integer	None	False
env2-Mode-w	Mode	192.168.1.15	502	1	16	10	2	Unsigned Integer	None	False

MQTT Data Type	Multiplier	Offset	Polling Interval (ms)	Send When Change	Custom Field	custom2 field	Send Group	Send Interval
Boolean	1	0	10000	No	0	0	0	1
Boolean	1	0	10000	No	0	0	0	1
Float	1	0	10000	Yes	0	0	0	1
Unsigned Integer	1	0	10000	No	0	0	0	1
Unsigned Integer	1	0	10000	No	0	0	0	1
Boolean	1	0	10000	No	0	0	0	1
Boolean	1	0	10000	No	0	0	0	1
Float	1	0	10000	Yes	0	0	0	1
Unsigned Integer	1	0	10000	No	0	0	0	1
Unsigned Integer	1	0	10000	No	0	0	0	1

Figure 7: CSV file structure

The table below describes all supported columns in the CSV file (columns A through AB).

Column	Item	Description
A (1st)	*Topic*	MQTT topic string.
B (2nd)	*Name*	Identifier for the mapping rule.
C (3rd)	*IP Address or Peripheral Interface Selection*	Specifies the network IP address of the target Modbus device when using Modbus TCP communication. When Modbus RTU or Modbus ASCII mode is selected, this field is used to specify the physical communication interface instead of an IP address. The supported interface options are: • Expansion Port 1 • Expansion Port 2 • USB Port
D (4th)	*Port Number / Communication Mode*	For Modbus TCP communication, this parameter specifies the TCP port number of the remote Modbus slave device. When Modbus RTU or Modbus ASCII mode is selected, this field defines the serial communication protocol format. The supported options are: • RTU (default) • ASCII
E (5th)	*Device ID*	Modbus Unit ID (Slave ID).

Table 3: CSV configuration columns description

Column	Item	Description
F (6th)	*Function Code*	Supported Modbus Function Codes: **01**: Read coils **02**: Read discrete inputs **03**: Read holding registers **04**: Read input registers **05**: Write single coil **06**: Write single register **15**: Write multiple coils **16**: Write multiple registers
G (7th)	*Address*	Starting Modbus register address.
H (8th)	*Data Length*	Number of bits (FC 1, 2, 5, 15) or words (FC 3, 4, 6, 16).
I (9th)	*Modbus Data Type*	`Boolean`, `Integer`, `Unsigned Integer`, `Float`.
J (10th)	*Data Swap*	Byte order handling: **None**: No swap. **Word**: 0x0102 → 0x0201. **Double Word**: 0x01020304 → 0x04030201. **Quad Word**: Swaps 8 bytes fully.
K (11th)	*Byte Swap*	`TRUE` or `FALSE`. Swaps bytes within words.
L (12th)	*MQTT Data Type*	JSON payload format: `Boolean`, `Integer`, `Unsigned Integer`, `Float`, `Long Integer`, `Unsigned Long Integer`, `String`.
M (13th)	*Multiplier*	Factor to multiply the value by.
N (14th)	*Offset*	Value to add/subtract from the data.
O (15th)	*Polling Interval*	Polling rate in milliseconds (Range: `1` to `10000000`).
P (16th)	*Send When Change*	`Yes`: Send only on change. `No`: Send every poll.
Q (17th)	*Custom1 Field*	Definition value for Custom Field 1.
R (18th)	*Custom2 Field*	Definition value for Custom Field 2.
S (19th)	*Send Group*	Group ID (`0`–`500`) to bundle multiple Modbus points into one MQTT message. `0` = disabled.
T (20th)	*Send Interval*	Interval for grouped messages in seconds (`1`–`10000`).
U (21st)	*Custom3 Field*	Definition value for Custom Field 3.
V (22nd)	*Custom4 Field*	Definition value for Custom Field 4.
W (23rd)	*Custom5 Field*	Definition value for Custom Field 5.
X (24th)	*Custom6 Field*	Definition value for Custom Field 6.
Y (25th)	*Custom7 Field*	Definition value for Custom Field 7.
Z (26th)	*Custom8 Field*	Definition value for Custom Field 8.
AA (27th)	*Custom9 Field*	Definition value for Custom Field 9.
AB (28th)	*Custom10 Field*	Definition value for Custom Field 10.

Table 3: CSV configuration columns description (continued)

CSV Example

Below is a copy-pasteable example from the section above, which you can save directly as a plain-text CSV file:

```
env1-DI,DI_01,192.168.1.15,502,1,2,1,1,Boolean,None,FALSE,Boolean,1,0,10000,No,0,0,0,1
env1-DO,DO,192.168.1.15,502,1,1,1,1,Boolean,None,FALSE,Boolean,1,0,10000,No,0,0,0,1
env1-Temp,Temperature,192.168.1.15,502,1,4,1,2,Float,None,FALSE,Float,1,0,10000,Yes,0,0,0,1
env1-Mode,Mode,192.168.1.15,502,1,3,10,2,Unsigned Integer,None,FALSE,Unsigned Integer,1,0,10000,No,0,0,0,1
env1-Mode-w,Mode,192.168.1.15,502,1,16,10,2,Unsigned Integer,None,FALSE,Unsigned Integer,1,0,10000,No,0,0,0,1
env2-DI,DI_01,192.168.1.16,502,1,2,1,1,Boolean,None,FALSE,Boolean,1,0,10000,No,0,0,0,1
env2-DO,DO,192.168.1.16,502,1,1,1,1,Boolean,None,FALSE,Boolean,1,0,10000,No,0,0,0,1
env2-Temp,Temperature,192.168.1.16,502,1,4,1,2,Float,None,FALSE,Float,1,0,10000,Yes,0,0,0,1
env2-Mode,Mode,192.168.1.16,502,1,3,10,2,Unsigned Integer,None,FALSE,Unsigned Integer,1,0,10000,No,0,0,0,1
env2-Mode-w,Mode,192.168.1.15,502,1,16,10,2,Unsigned Integer,None,FALSE,Unsigned Integer,1,0,10000,No,0,0,0,1
```

4.2 MQTT Data Format

When Modbus/TCP function codes 1, 2, 3, or 4 are used, Modbus to MQTT acts as an MQTT publisher and sends Modbus data in JSON format to the MQTT broker.

When function codes 5, 6, 15, or 16 are used, Modbus to MQTT acts as an MQTT subscriber, receives control messages from the broker, and forwards them to the Modbus device.

ini Code

```
”time”: ”2026-06-09 15:25:06.667”,
”topic”: ”env1-DI”,
”name”: ”DI_01”,
”value”: true,
”ip”: ”192.168.1.15”,
”port”: ”502”,
”id”: ”1”,
”fc”: ”1”,
”address”: ”1”,
”data length”: ”1”
```

Note that Modbus to MQTT validates only the **topic**, **name**, and **value** fields of received subscription messages.

ini Code

```
”topic”: ”env1-Mode-w”,
”name”: ”Mode”,
”value”: ”1234”
```

5. Related Resources

You can obtain all product-related documents, software updates, and supplementary materials on the Advantech *Engineering Portal* at *icr.advantech.com*.

For easy access to specific resources, please refer to the following sections of the portal:

- **Router Support Materials:** To access your router's supporting documents (such as the *Hardware Manual* and *Configuration Manual*), the latest firmware, or other technical resources, navigate to *Support* → *Router Models*. Locate your specific model and select the appropriate tab under the *Documents to download* section. Available tabs include *Brochures*, *Manuals*, *Certificates*, *Firmware*, *Images/3D Models*, *PCN/SA*, and *Others*.
- **Router Apps:** To extend your router's functionality, installation packages and comprehensive manuals for various extension modules are available by navigating to *Download* → *Router Apps*.
- **Application Notes:** For detailed guides, configuration examples, and step-by-step instructions for implementing specific networking features and use cases, navigate to *Download* → *Application Notes*.
- **Development Documents:** If you are interested in custom scripting, programming your own applications, or compiling custom modules, navigate to *Development* page.