
MQTT

NETIO M2M API protocols docs

Protocol version: MQTT-JSON Version 2.0

Short summary

MQTT is implemented as M2M API protocol, where NETIO device is publisher providing output status and subscriber allowing control of NETIO power outputs (power sockets 230V or power outlets IEC-320 110/230V). A MQTT payload has JSON structure and is prepared for JSON parsers.

- For NETIO 4All, the protocol also includes power consumption values.
- The MQTT protocol must be enabled first in the WEB configuration of the respective device. For details, see the “NETIO WEB configuration” chapter.
- Username and password to access the MQTT broker have to be filled on the web GUI.

Supported devices

- NETIO 4All
- NETIO 4 (Energy metering not supported)
- NETIO 4C (Energy metering not supported)

Note: NETIO 4x means all NETIO 4 devices (NETIO 4 / 4All / 4C)

Supported devices and firmware

NETIO 4x firmware – 3.0.1 and later

General protocol info

MQTT is a machine-to-machine (M2M) / "Internet of Things" connectivity protocol. It was designed as an extremely lightweight publish/subscribe messaging transport. It is useful for connections with remote locations where a small code footprint is required and/or network bandwidth is at a premium.

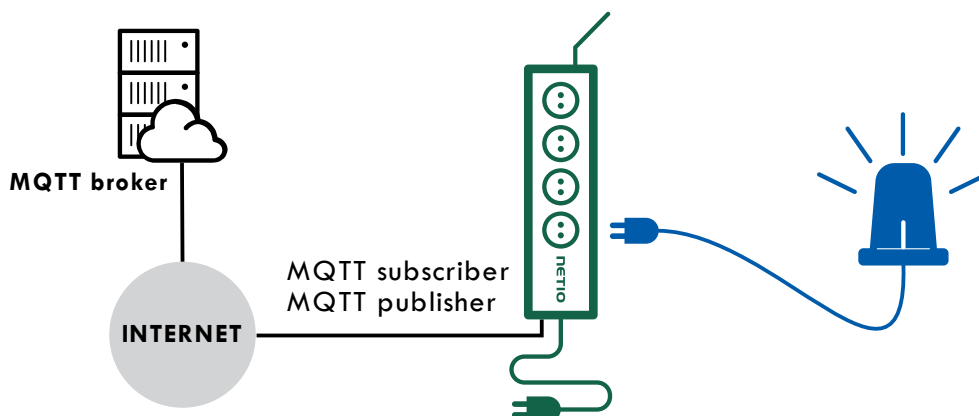
Sources:

<http://mqtt.org/>

<https://en.wikipedia.org/wiki/MQTT>

NOTE: This document provides basic info about the M2M API protocol. Other device functions are described in the product manual.

Quick start with MQTT & NETIO



Topics

- subscribe to **devices/<Client Id>/messages/events/**
- push commands to **devices/<Client Id>/messages/devicebound/**

ID - number of output

A payload has JSON structure and is prepared for JSON parsers.

Push command - Switch Power output 1 to ON and output 3 to OFF.

```
{"Operation": "SetOutputs", "Outputs": [{"ID": 1, "Action": 1}, {"ID": 3, "Action": 0}]}
```

ID - number of output

Action - Supported actions: 0 – off, 1 – on, 2 – short off, 3 – short on, 4 – toggle, 5 – no change

General NETIO 4x output functions

Output status – “read” function

- **0** – Power **OFF**
- **1** – Power **ON**

Output actions – “write” function

- **0** – Turn **OFF**
- **1** – Turn **ON**
- **2** – Short OFF delay (restart)
- **3** – Short ON delay
- **4** – **Toggle** (invert the state)
- **5** – No change
- **6** – Ignored (return value from reading the tag)
Current output value is in “State” tag (0 / 1).

Short ON / OFF delay

This command switches a power output On / Off for a defined time. It is useful for example to power-cycle a server with a defined switch-off time, or to switch on a pump for a defined time.

This “short” delay is protected: the power output will remain in the defined state regardless of any other M2M requests received. During this time, the output state can only be changed by pressing the button on the NETIO device and this action cancel M2M short ON/OFF command for the particular output. Other requests to control the particular output are simply ignored and an ERROR logged with reason rejected in a device Log.

The short ON / OFF delay interval can be defined in the device web administration. It is specified in ms (milliseconds) and rounded up to hundreds of milliseconds (0,1s).

This interval can be also defined using some M2M API protocol commands. In that case, it is valid only for a single protocol session (the following short ON / Short OFF command). When the connection is closed or restarted, the interval is reset to the device default value (defined in the web administration for each output).

Security issues

Do not use default usernames and passwords! Keep your Ethernet and WiFi networks secured.

Power-Up outputs state

All outputs are Off during the first 25 to 30 seconds after power-up.

After this time, all outputs are set to the selected state:

- **Last Output state**
After a power outage, the NETIO device sets each power output to the last stored state of this one output. The current state of each power output (socket/power outlet) is internally stored every 8 seconds.

Note: **Function Scheduler** is checked in Power-Up initialization. When enabled, it can affect one or more power output stated based on current time and date.

Custom based **Lua scripts** can affect output stated too.

Energy metering variables

Since NETIO fw 3.0.0 and later, there are 23 variables available for NETIO energy metering.

Parameters for each power output:

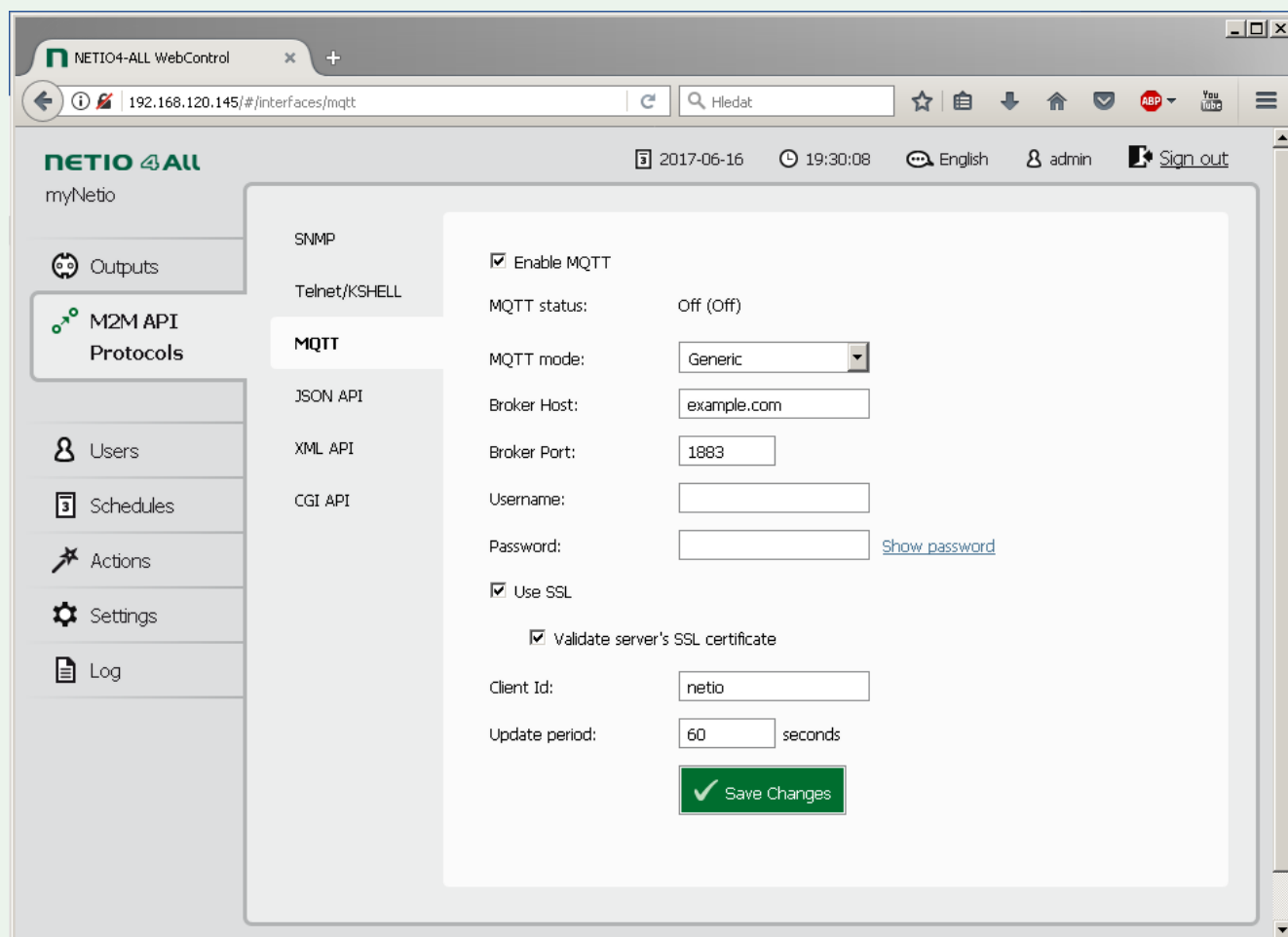
Variable	Unit	Description
4x Current	mA	Instantaneous current for the specific power output
4x PowerFactor	-	Instantaneous Power Factor for the specific power output
4x Load	W	Instantaneous load for the specific power output
4x Energy	Wh	Instantaneous Energy counter value for the specific power output

Parameters for the whole NETIO device:

Variable	Unit	Description
1x Voltage	V	Instantaneous voltage
1x Frequency	Hz	Instantaneous frequency
1x TotalCurrent	mA	Instantaneous total current through all power outputs
1x OverallPowerFactor	-	Instantaneous Power Factor – weighted average from all meters
1x TotalLoad	W	Total Load of all power outputs (device's own internal consumption is not included)
1x TotalEnergy	Wh	Instantaneous value of the Total Energy counter
1x EnergyStart	-	Date and time of the last reset of all energy counters

NETIO WEB configuration

M2M API protocols can be enabled and configured only over the web administration – select “M2M API Protocols” in the left-hand side menu and then select the “MQTT” tab.



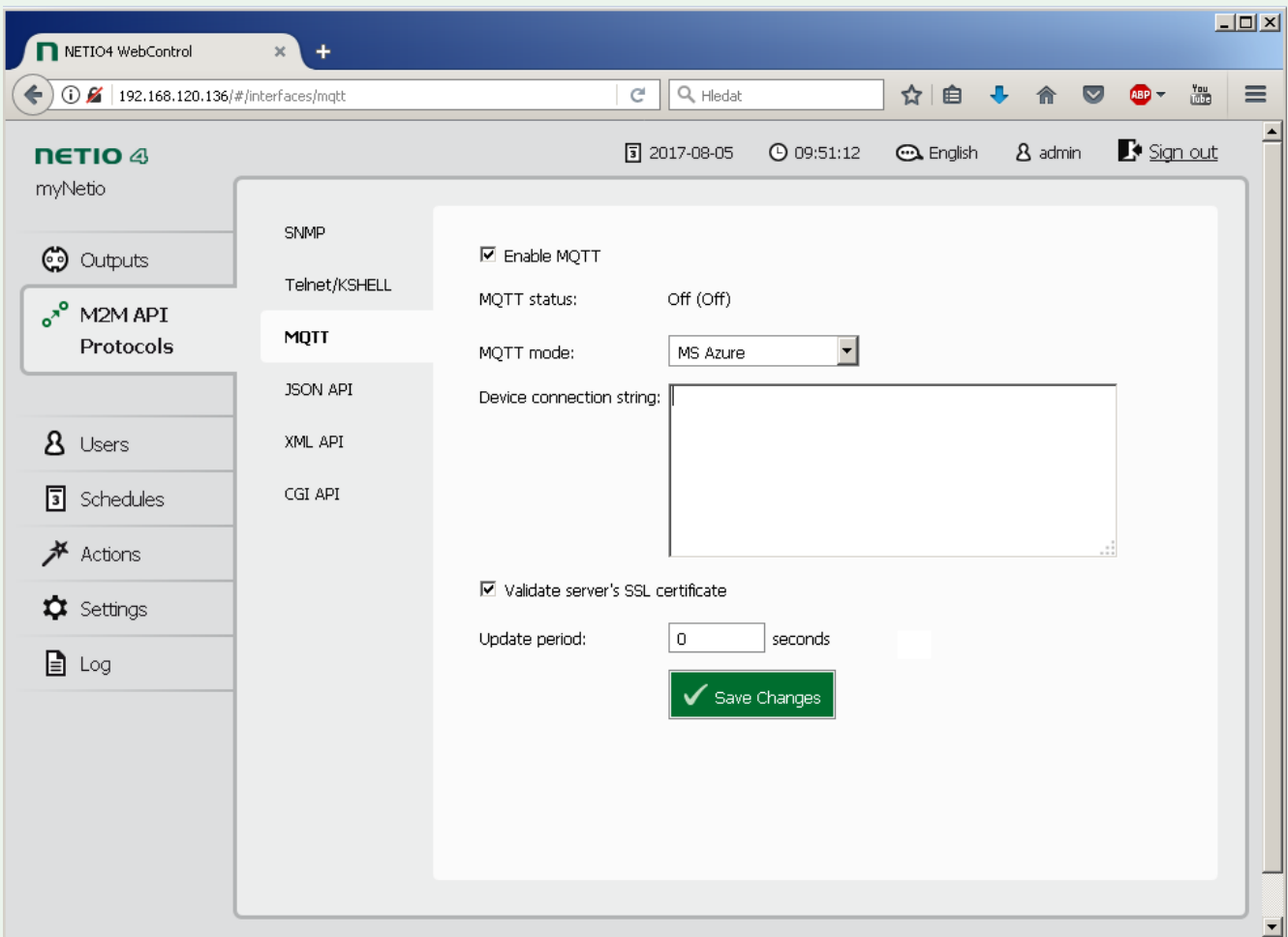
Picture 1 – M2M API Protocols / MQTT settings GUI – Generic mode

- **Enable MQTT** - Enable/disable M2M API protocol
- **MQTT status** - only notification MQTT connection status (Off, Connected, Error)
- **MQTT mode**
 - **Generic** - for general MQTT broker
 - **MS Azure** - for connection to MS Azure IoT Hub. Device Explorer from MS is necessary.

Generic mode (Picture 1):

- **Broker Host** - domain name or IP address of MQTT broker
- **Broker Port** - socket port where MQTT broker is served
- **Username** - user credential to login client to broker

- **Password** - password credential to login client to broker
- **Use SSL** - check if you want to use secure connection
- **Validate server's SSL certificate** - Check if you want to validate served authentication certificate against trusted certification authority.
- **Client Id** - Unique device id to distinguish many devices connected to one broker (Client Id is a part of MQTT topics)
- **Update period** - Outlets update message is repeatedly sent in this period. When 0 seconds is set - no period is applied and status is sent only if output status or measurements changed.



Picture 2 – M2M API Protocols / MQTT settings GUI – MS Azure mode

MS Azure mode (Picture 2):

- **Device connection string** - Device-specific connection string generated in Device Explorer.
- **Validate server's SSL certificate** - Check if you want to validate served authentication certificate against trusted certification authority.
- **Update period** - Outlets update message is repeatedly sent in this period. When 0 seconds is set - no period is applied and status is sent only if output status or measurements changed.

NETIO MQTT protocol structure

Generic mode

Topics

- subscribe to **devices/<Client Id>/messages/events/**
- push commands to **devices/<Client Id>/messages/devicebound/**

QOS must be set to 0.

A payload has JSON structure and is prepared for JSON parsers.

MQTT messages payloads (reading, subscribe to)

Topic subscription: **devices/<Client Id>/messages/events/**

Outlets update message: Sent according to parameter "Update period"

```
{
  "GlobalMeasure": {
    "Voltage": 238.1,
    "Frequency": 49.9,
    "TotalCurrent": 180,
    "OverallPowerFactor": 0.49,
    "TotalLoad": 21,
    "TotalEnergy": 1345,
    "EnergyStart": "2017-06-23T16:47:53+01:00" },
  "Outputs": [
    {
      "ID": 1,
      "Name": "output_1",
      "State": 0,
      "Action": 6,
      "Delay": 5000,
      "Current": 0,
      "PowerFactor": 0.00,
      "Load": 0,
      "Energy": 497
    },
    {
```

```

    "ID":2,
    "Name":"output_2",
    "State":1,
    "Action":6,
    "Delay":5000,
    "Current":0,
    "PowerFactor":0,
    "Load":0,
    "Energy":148
  },
  {
    "ID":3,
    "Name":"output_3",
    "State":1,
    "Action":6,
    "Delay":5000,
    "Current":180,
    "PowerFactor":0.49,
    "Load":21,
    "Energy":196
  },
  {
    "ID":4,
    "Name":"output_4",
    "State":0,
    "Action":6,
    "Delay":5000,
    "Current":0,
    "PowerFactor":0.00,
    "Load":0,
    "Energy":502
  }
]
}

```

Note:

1. Items/values related to metering (Voltage, Frequency, Current, PowerFactor, Load and Energy, etc.) are available only for the NETIO 4All model.

2. Returned message contains always "Action" with value "6" for all outputs. This value means "ignore" and works as a placeholder. Output state 0 / 1 is in the **State** value.

Outlet change message: Triggered by specific outlet status/metering change

```
{
  "Outputs": [
    {
      "ID": 1,
      "Name": "output_1",
      "State": 1,
      "Current": 0,
      "PowerFactor": 0.00,
      "Load": 0,
      "Energy": 497
    }
  ]
}
```

Note: Items/values related to metering (Voltage, Frequency, Current, PowerFactor, Load and Energy, etc.) are available only for the NETIO 4All model.

Detail status message: Can be invoked by Get description command message (described below)

```
{
  "Agent": {
    "Model": "NETIO 4All",
    "Version": "3.0.1",
    "JSONVer": "2.0",
    "DeviceName": "myNetio_10",
    "VendorID": 0,
    "OemID": 0,
    "MAC": "24:A4:2C:33:25:E1",
    "Uptime": 1708,
    "Time": "2017-11-21T10:23:34+00:00",
    "NumOutputs": 4
  },
  "GlobalMeasure": {
    "Voltage": 238.4,
    "Frequency": 49.9,
    "TotalCurrent": 0,
    "OverallPowerFactor": 0,
    "TotalLoad": 0,
    "TotalEnergy": 0,
  }
}
```

```
"EnergyStart": " 2017-06-23T16:47:53+01:00"  
},  
"Outputs": [  
  {  
    "ID": 1,  
    "Name": "output_1",  
    "State": 0,  
    "Action":6,  
    "Delay":5000,  
    "Current": 0,  
    "PowerFactor": 0,  
    "Load": 0,  
    "Energy": 0  
  },  
  {  
    "ID": 2,  
    "Name": "output_2",  
    "State": 0,  
    "Action":6,  
    "Delay":5000,  
    "Current": 0,  
    "PowerFactor": 0,  
    "Load": 0,  
    "Energy": 0  
  },  
  {  
    "ID": 3,  
    "Name": "output_3",  
    "State": 0,  
    "Action":6,  
    "Delay":5000,  
    "Current": 0,  
    "PowerFactor": 0,  
    "Load": 0,  
    "Energy": 0  
  },  
  {  
    "ID": 4,  
    "Name": "output_4",  
    "State": 0,  
    "Action":6,
```

```
    "Delay":5000,
    "Current": 0,
    "PowerFactor": 0,
    "Load": 0,
    "Energy": 0
  }
]
```

Note: Items/values related to metering (Voltage, Frequency, Current, PowerFactor, Load and Energy, etc.) are available only for the NETIO 4All model.

MQTT messages payloads (command, publish to push commands topic)

Topic publication to: **devices/<Client Id>/messages/devicebound/**

Set output command – single output control:

```
{"Operation":"SetOutputs","Outputs": [{"ID":<X>, "Action":<Z>}]}
```

Set output command – multiple outputs control:

```
{"Operation":"SetOutputs","Outputs": [{"ID":<X>, "Action":<Z>}, {"ID":<X>, "Action":<Z>}]}
```

ID: **X** = number of output

Action: **Z** = Supported actions: 0 – off, 1 – on, 2 – short off, 3 – short on, 4 – toggle, 5 – no change, 6 - Ignored value, use State tag)

Outputs can be controlled by two options:

1. **Action:** 0 – off, 1 – on, 2 – short off, 3 – short on, 4 – toggle, 5 – no change, (6 – ignore)
2. **State:** 0 – off, 1 – on (Action = 6 required)

Note: **Action** with other value than 6 has higher priority than the **State** tag.
State value is not reflected in case Action = 1 to 5.
If you wish to use **State** tag to control an output, **Action = 6 is required.**

Power output 1 = 1

```
{"Operation":"SetOutputs","Outputs": [{"ID":1, "Action":6, "State":1,}]}
```

Toggle Power output 1

```
{"Operation":"SetOutputs","Outputs": [{"ID":1, "Action":4}]}
```

Power output 1 = 1 + Power output 3 = 0

```
{"Operation":"SetOutputs","Outputs": [{"ID":1, "Action":1},{ "ID":3, "Action":0}]}
```

Switch Power output 1 to ON for 2 second, then switch it OFF.

```
{"Operation":"SetOutputs","Outputs": [{"ID":1, "Action":3, "Delay":2000}]}
```

Switch Power output 1 to ON for period set at web administration, then switch it OFF.

```
{"Operation":"SetOutputs","Outputs": [{"ID":1, "Action":3}]}
```

Note: If a Delay is not specified for Actions 2 or 3 then the Delay value defined at web administration for individual outputs is used.

Get description command:

This is used to get complete device status and information (Detail status message)

```
{"Operation": "GetDescription" }
```

MS Azure mode

Message payloads are the same as in Generic mode. DeviceId is parsed from ConnectionString so it is the same as device name in MS Azure.

NETIO 4 – listing of messages

Note: No metering values available for NETIO 4.

NETIO 4 - Outlets update message

```
{"Outputs": [{"ID": 1, "Name": "output_1", "State": 0}, {"ID": 2, "Name": "output_2", "State": 0}, {"ID": 3, "Name": "output_3", "State": 0}, {"ID": 4, "Name": "output_4", "State": 0}]}
```

NETIO 4 - Detail status message

```
{"Agent": {"Model": "NETIO 4", "Version": "3.0.1", "JSONVer": "2.0", "DeviceName": "myNetio", "VendorID": 0, "OemID": 0, "SerialNumber": "24:A4:2C:39:02:B8", "Uptime": 299, "Time": "2017-11-21T12:42:02+00:00", "NumOutputs": 4}, "Outputs": [{"ID": 1, "Name": "output_1", "State": 0, "Action": 6, "Delay": 5000}, {"ID": 2, "Name": "output_2", "State": 0, "Action": 6, "Delay": 5000}, {"ID": 3, "Name": "output_3", "State": 0, "Action": 6, "Delay": 5000}, {"ID": 4, "Name": "output_4", "State": 0, "Action": 6, "Delay": 5000}]}
```

NETIO 4 - Outlet change message

```
{"Outputs": [{"ID":1, "Name": "output_1", "State":1}]}
```

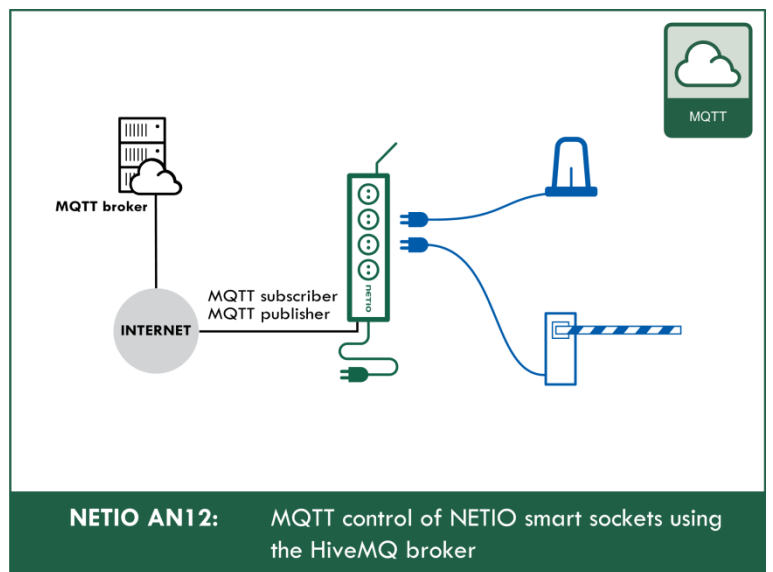
Notes

- The “Uptime” value is in seconds [s]

NETIO AN (Application Note)

AN12 MQTT control of NETIO smart sockets using the HiveMQ broker

NETIO 4x power sockets (4x 230V, LAN and WiFi) can be controlled over the Web and with various M2M protocols. Supported protocols include MQTT. Application Note AN12 describes the use of the MQTT protocol in the “Generic” mode with the HiveMQ public MQTT broker. This public broker is intended for testing; for live use, we recommend to use your own installation of this or another MQTT broker.



>> Read the AN12 on www.netio-products.com

Document history

Document Revision	Publication Date	Description
1.0	16.4.2018	Initial release
1.1	16.10.2018	AN20 description added
1.2	23.11.2018	Detailed description about the Action tag implemented