

---

**sml2mqtt**

**spacemanspiff2007**

**Apr 26, 2024**



## CONTENTS:

<b>1</b>	<b>Installation</b>	<b>3</b>
1.1	Virtual environment . . . . .	3
1.2	Docker . . . . .	5
<b>2</b>	<b>Configuration</b>	<b>7</b>
2.1	Example . . . . .	7
2.2	Example input Tibber bridge . . . . .	9
2.3	Example mqtt config . . . . .	9
2.4	Configuration Reference . . . . .	9
<b>3</b>	<b>Command Line Interface</b>	<b>15</b>
<b>4</b>	<b>Getting started</b>	<b>17</b>
4.1	1. Installation . . . . .	17
4.2	2. Create default configuration . . . . .	17
4.3	3. Edit inputs and mqtt . . . . .	18
4.4	4. Run with analyze . . . . .	20
4.5	5. Edit device settings . . . . .	22
4.6	6. Edit value settings . . . . .	23
<b>5</b>	<b>Operations</b>	<b>25</b>
5.1	Filters . . . . .	25
5.2	Actions . . . . .	27
5.3	Math . . . . .	27
5.4	Workarounds . . . . .	28
5.5	Date time based . . . . .	29
5.6	Time series . . . . .	30
5.7	Operations . . . . .	32
5.8	Examples . . . . .	32
<b>6</b>	<b>Indices and tables</b>	<b>35</b>
	<b>Index</b>	<b>37</b>



sml2mqtt is a asyncio application that can read multiple sml (Smart Message Language) streams from energy meters and report the values through mqtt. The meters can be read through serial ports or through http (e.g. Tibber) and the values that will be reported can be processed in various ways with operations.



## INSTALLATION

### 1.1 Virtual environment

#### 1.1.1 Installation

---

**Hint:** On Windows use the `python` command instead of `python3`

---

Navigate to the folder in which the virtual environment shall be created (e.g.):

```
cd /opt/sml2mqtt
```

If the folder does not exist yet you can create it with the `mkdir` command:

```
mkdir /opt/sml2mqtt
```

Create virtual environment (this will create a new subfolder “venv”):

```
python3 -m venv venv
```

Go into folder of virtual environment:

```
cd venv
```

1. Activate the virtual environment

Linux:

```
source bin/activate
```

Windows:

```
Scripts\activate
```

2. Upgrade pip and setuptools:

```
python3 -m pip install --upgrade pip setuptools
```

Install `sml2mqtt`:

```
python3 -m pip install sml2mqtt
```

1. Run sml2mqtt:

```
sml2mqtt --config PATH_TO_CONFIGURATION_FILE
```

### 1.1.2 Upgrading

1. Stop sml2mqtt
2. Activate the virtual environment

Navigate to the folder where sml2mqtt is installed:

```
cd /opt/sml2mqtt
```

Activate the virtual environment

Linux:

```
source bin/activate
```

Windows:

```
Scripts\activate
```

3. Run the following command in your activated virtual environment:

```
python3 -m pip install --upgrade sml2mqtt
```

4. Start sml2mqtt
5. Observe the log for errors in case there were changes

### 1.1.3 Autostart after reboot

To automatically start the sml2mqtt from the virtual environment after a reboot call:

```
nano /etc/systemd/system/sml2mqtt.service
```

and copy paste the following contents. If the user/group which is running sml2mqtt is not “openhav” replace accordingly. If your installation is not done in “/opt/sml2mqtt/venv/bin” replace accordingly as well:

```
[Unit]
Description=sml2mqtt
Documentation=https://github.com/spacemanspiff2007/sml2mqtt
After=network-online.target

[Service]
Type=simple
User=openhav
Group=openhav
Restart=on-failure
RestartSec=10min
ExecStart=/opt/sml2mqtt/venv/bin/sml2mqtt -c PATH_TO_CONFIGURATION_FILE
```

(continues on next page)



(continued from previous page)

```
[Install]
WantedBy=multi-user.target
```

Now execute the following commands to enable autostart:

```
sudo systemctl --system daemon-reload
sudo systemctl enable sml2mqtt.service
```

It is now possible to start, stop, restart and check the status of sml2mqtt with:

```
sudo systemctl start sml2mqtt.service
sudo systemctl stop sml2mqtt.service
sudo systemctl restart sml2mqtt.service
sudo systemctl status sml2mqtt.service
```

## 1.2 Docker

Installation through [docker](#) is available:

```
docker pull spacemanspiff2007/sml2mqtt:latest
```

The docker image has one volume `/sml2mqtt` which has to be mounted. There the `config.yml` will be used or a new `config.yml` will be created

The analyze option can also be set through an environment variable (see [command line interface](#)).



## CONFIGURATION

Configuration of sml2mqtt is done through a yaml file. The path to the file can be specified with `-c PATH` or `--config PATH`. If nothing is specified a file with the name `config.yml` is searched in the subdirectory `sml2mqtt` in

- the current working directory
- the venv directory
- the user home

If a config file is specified and it does not yet exist a default configuration file will be created.

### 2.1 Example

```
logging:
  level: INFO          # Log level
  file: sml2mqtt.log   # Log file path (absolute or relative to config file) or stdout

mqtt:
  connection:
    identifier: sml2mqtt-ZqlFvhSBdDGvJ
    host: localhost
    port: 1883
    user: ''
    password: ''
  topic prefix: sml2mqtt
  defaults:
    qos: 0             # Default value for QOS if no other QOS value in the config entry is
    ↪ set
    retain: false      # Default value for retain if no other retain value in the config
    ↪ entry is set
    last will:
      topic: status    # Topic fragment for building this topic with the parent topic

general:
  Wh in kWh: true      # Automatically convert Wh to kWh
  republish after: 120 # Republish automatically after this time (if no other filter
    ↪ configured)

inputs:
- type: serial
```

(continues on next page)

```

url: COM1    # Device path
timeout: 3   # Seconds after which a timeout will be detected (default=3)
- type: serial
  url: /dev/ttyS0    # Device path
  timeout: 3         # Seconds after which a timeout will be detected (default=3)

devices:
  # Device configuration by reported id
  device_id_hex:

    mqtt:                                     # Optional MQTT configuration for this meter.
      topic: DEVICE_BASE_TOPIC               # Topic fragment for building this topic with the
      ↪parent topic

    status:                                  # Optional MQTT status topic configuration for this
      ↪meter
      topic: status                          # Topic fragment for building this topic with the
      ↪parent topic

    skip:                                    # OBIS codes (HEX) of values that will not be
      ↪published (optional)
      - '00112233445566'

    # Configurations for each of the values (optional)
    values:

      - obis: '00112233445566'               # Obis code for this value
        mqtt:                               # Mqtt config for this value (optional)
          topic: OBIS_VALUE_TOPIC           # Topic fragment for building this topic with the
          ↪topic prefix
          # A sequence of operations that will be evaluated one after another.
          # If one operation blocks nothing will be reported for this frame
          operations:
            - negative on energy meter status: true  # Make value negative based on an energy
            ↪meter status. Set to "true" to enable or to "false" to disable workaround. If the
            ↪default obis code for the energy meter is wrong set to the appropriate meter obis code
            ↪instead
            - factor: 3                        # Factor with which the value gets multiplied
            - offset: 100                      # Offset that gets added on the value
            - round: 2                         # Round to the specified digits
            - refresh action: 300              # Republish value every 300s

```

## 2.2 Example input Tibber bridge

These input settings can be used to poll data from a Tibber bridge:

```
inputs:
- type: http
  url: http://IP_OR_HOSTNAME_OF_TIBBER_BRIDGE/data.json?node_id=1
  interval: 3 # Poll interval secs
  timeout: 10 # After which time the input will change to TIMEOUT
  user: "admin"
  password: "printed on bridge socket"
```

## 2.3 Example mqtt config

MQTT topics can be configured either by providing a full topic or a topic fragment. With a topic fragment the resulting topic is build with the parent topic. The structure is `topic prefix / device / value`. Providing a full topic will ignore the fragments. The entries for qos and retain are optional.

```
full topic: my/full/topic
qos: 1
```

## 2.4 Configuration Reference

All possible configuration options are described here. Not all entries are created by default in the config file and one should take extra care when changing those entries.

### settings Settings

```
field logging: LoggingSettings [Optional]
field mqtt: MqttConfig [Optional]
field general: GeneralSettings [Optional]
field inputs: list[HttpSourceSettings | SerialSourceSettings] = []
field devices: dict[Annotated[str], SmlDeviceConfig] = {}
    Device configuration by ID or url
```

### 2.4.1 logging

#### settings LoggingSettings

```
field level: str = 'INFO'
    Log level
field file: str = 'sml2mqtt.log'
    Log file path (absolute or relative to config file) or "stdout"
```

## 2.4.2 general

### settings GeneralSettings

**field Wh in kWh:** `bool = True`  
Automatically convert Wh to kWh

**field republish after:** `int = 120`  
Republish automatically after this time (if no other filter configured)

**field report blank energy meters:** `bool = False`  
Report blank energy meters (where the value is 0kwh)

**field report device id:** `bool = False`  
Report the device id even though it does never change

**field device id obis:** `list[Annotated[str]] = ['0100000009ff', '0100600100ff']`  
Additional OBIS fields for the serial number to configuration matching

## 2.4.3 inputs

### settings SerialSourceSettings

**field type:** `Literal['serial'] [Required]`

**field url:** `Annotated[str] [Required]`  
Device path

**Constraints**

- `strip_whitespace = True`
- `strict = True`
- `min_length = 1`

**field timeout:** `Union[Annotated[int], Annotated[float]] = 6`  
Seconds after which a timeout will be detected (default=6)

**field baudrate:** `int = 9600`

**field parity:** `str = 'None'`

**field stop bits:** `Union[Annotated[int], Annotated[float]] = 1`

**field byte size:** `int = 8`

Example:

```
type: serial
url: COM3
```

### settings HttpSourceSettings

**field type:** `Literal['http'] [Required]`

**field url:** `Annotated[Url] [Required]`

Url

**Constraints**

- `allowed_schemes` = ['http', 'https']

**field timeout:** `Union[Annotated[int], Annotated[float]] = 6`

Seconds after which a timeout will be detected (default=6)

**field interval:** `Union[Annotated[int], Annotated[float]] = 2`

Delay between requests

**Constraints**

- `ge` = 0.1

**field user:** `str = ''`

User (if needed)

**field password:** `str = ''`

Password (if needed)

**field request timeout:** `Union[Annotated[int], Annotated[float], None] = None`

Dedicated timeout for the http request

Example:

```
type: http
url: http://localhost:8080/sml
interval: 3
timeout: 10
```

## 2.4.4 mqtt

settings `MqttConfig`

**field connection:** `MqttConnection [Optional]`

**field topic prefix:** `Annotated[str] = 'sml2mqtt'`

Prefix for all topics. Set to empty string to disable

**Constraints**

- `strict` = True
- `strip_whitespace` = True

**field defaults:** `MqttDefaultPublishConfig [Optional]`

**field last will:** `OptionalMqttPublishConfig [Optional]`

settings `MqttConnection`

**field identifier:** `Annotated[str] = 'sml2mqtt-tNeMNZrhuwJSQ'`

**Constraints**

- `strict` = True
- `strip_whitespace` = True

**field host:** `Annotated[str] = 'localhost'`

**Constraints**

- `strict = True`
- `strip_whitespace = True`

**field port:** `int = 1883`

**Constraints**

- `ge = 0`

**field user:** `Annotated[str] = ''`

**Constraints**

- `strict = True`
- `strip_whitespace = True`

**field password:** `Annotated[str] = ''`

**Constraints**

- `strict = True`
- `strip_whitespace = True`

**field tls:** `MqttTlsOptions | None = None`

**settings OptionalMqttPublishConfig**

**field topic:** `Optional[Annotated[str]] = None`

Topic fragment for building this topic with the parent topic

**field full topic:** `Optional[Annotated[str]] = None`

Full topic - will ignore the parent topic parts

**field qos:** `Optional[Literal[0, 1, 2]] = None`

QoS for publishing this value (if set - otherwise use parent)

**field retain:** `Optional[Annotated[bool]] = None`

Retain for publishing this value (if set - otherwise use parent)

**settings MqttDefaultPublishConfig**

**field qos:** `Literal[0, 1, 2] = 0`

Default value for QoS if no other QoS value in the config entry is set

**field retain:** `Annotated[bool] = False`

Default value for retain if no other retain value in the config entry is set

**Constraints**

- `strict = True`

**settings MqttTlsOptions**

**field insecure:** `bool | None = None`

Enable/disable server hostname verification when using SSL/TLS.



**field ca certificates:** `str | None = None`  
 Path to Certificate Authority (CA) certificate file in PEM or DER format

**field cert file:** `str | None = None`  
 Path to PEM encoded client certificate file

**field key file:** `str | None = None`  
 Path to PEM encoded private keys file

**field file password:** `str | None = None`  
 Password to encrypt the cert file or the key file if needed

**field certificate requirement:** `Optional[Literal['NONE', 'OPTIONAL', 'REQUIRED']] = None`  
 Certificate requirement that the client imposes on the broker.

**field tls version:** `Optional[Literal['SSLv23', 'TLS', 'TLS_CLIENT', 'TLS_SERVER', 'TLSv1', 'TLSv1_1', 'TLSv1_2']] = None`  
 The version of the SSL/TLS protocol to be used.

**field ciphers:** `str | None = None`  
 Which encryption ciphers are allowable for the connection

## 2.4.5 devices

### settings SmlDeviceConfig

Configuration for a sml device

**field mqtt:** `OptionalMqttPublishConfig | None = None`  
 Optional MQTT configuration for this meter.

**field status:** `OptionalMqttPublishConfig = OptionalMqttPublishConfig(topic='status', full_topic=None, qos=None, retain=None)`  
 Optional MQTT status topic configuration for this meter

**field skip:** `set[Annotated[str]] [Optional]`  
 OBIS codes (HEX) of values that will not be published (optional)

**field values:** `list[SmlValueConfig] = []`  
 Configurations for each of the values (optional)

### settings SmlValueConfig

**field obis:** `Annotated[str] [Required]`  
 Obis code for this value

**Constraints**

- `strip_whitespace` = True
- `to_lower` = True
- `strict` = True
- `pattern` = `[0-9a-fA-F]{12}`

**field mqtt:** `OptionalMqttPublishConfig | None = None`  
 Mqtt config for this value (optional)

field operations: `Annotated[list[Annotated[Union[Annotated[OnChangeFilter], Annotated[DeltaFilter], Annotated[HeartbeatAction], Annotated[RangeFilter], Annotated[RefreshAction], Annotated[ThrottleFilter], Annotated[Factor], Annotated[Offset], Annotated[Round], Annotated[NegativeOnEnergyMeterWorkaround], Annotated[Or], Annotated[Sequence], Annotated[VirtualMeter], Annotated[MaxValue], Annotated[MinValue], Annotated[MaxOfInterval], Annotated[MinOfInterval], Annotated[MeanOfInterval]]]]]] = []`

A sequence of operations that will be evaluated one after another. If one operation blocks this will return nothing.

#### Constraints

- `min_length = 1`

## COMMAND LINE INTERFACE

```
usage: -c [-h] [-c CONFIG] [-a]
```

SML to MQTT bridge

options:

-h, --help	show this help message and exit
-c CONFIG, --config CONFIG	Path to configuration file
-a, --analyze	Process exactly one sml message, shows the values of the message and what will be reported. Can also be set by setting the environment variable "SML2MQTT_ANALYZE" to an arbitrary value



## GETTING STARTED

### 4.1 1. Installation

First install sml2mqtt e.g in a *virtual environment*.

### 4.2 2. Create default configuration

Run sml2mqtt with a path to a configuration file. A new default configuration file will be created.

```
logging:
  level: INFO          # Log level
  file: sml2mqtt.log   # Log file path (absolute or relative to config file) or stdout

mqtt:
  connection:
    identifier: sml2mqtt-Zq1FvhSBdDGvJ
    host: localhost
    port: 1883
    user: ''
    password: ''
  topic prefix: sml2mqtt
  defaults:
    qos: 0             # Default value for QOS if no other QOS value in the config entry is
    ↪ set
    retain: false      # Default value for retain if no other retain value in the config
    ↪ entry is set
  last will:
    topic: status      # Topic fragment for building this topic with the parent topic

general:
  Wh in kWh: true      # Automatically convert Wh to kWh
  republish after: 120 # Republish automatically after this time (if no other filter
    ↪ configured)

inputs:
- type: serial
  url: COM1            # Device path
  timeout: 3           # Seconds after which a timeout will be detected (default=3)
- type: serial
```

(continues on next page)

(continued from previous page)

```

url: /dev/ttyS0    # Device path
timeout: 3         # Seconds after which a timeout will be detected (default=3)

devices:
  # Device configuration by reported id
  device_id_hex:

    mqtt:          # Optional MQTT configuration for this meter.
      topic: DEVICE_BASE_TOPIC # Topic fragment for building this topic with the
    ↪parent topic

    status:        # Optional MQTT status topic configuration for this
    ↪meter
      topic: status  # Topic fragment for building this topic with the
    ↪parent topic

    skip:          # OBIS codes (HEX) of values that will not be
    ↪published (optional)
      - '00112233445566'

    # Configurations for each of the values (optional)
    values:
      - obis: '00112233445566' # Obis code for this value
        mqtt:                 # Mqtt config for this value (optional)
          topic: OBIS_VALUE_TOPIC # Topic fragment for building this topic with the
    ↪topic prefix
          # A sequence of operations that will be evaluated one after another.
          # If one operation blocks nothing will be reported for this frame
          operations:
            - negative on energy meter status: true # Make value negative based on an energy
    ↪meter status. Set to "true" to enable or to "false" to disable workaround. If the
    ↪default obis code for the energy meter is wrong set to the appropriate meter obis code
    ↪instead
            - factor: 3 # Factor with which the value gets multiplied
            - offset: 100 # Offset that gets added on the value
            - round: 2 # Round to the specified digits
            - refresh action: 300 # Republish value every 300s

```

### 4.3 3. Edit inputs and mqtt

Edit the configuration file and configure the appropriate *inputs* for serial or http (e.g. for tibber) and edit the mqtt settings.

```

1 logging:
2   level: INFO      # Log level
3   file: sml2mqtt.log # Log file path (absolute or relative to config file) or stdout
4
5 mqtt:
6   connection:

```

(continues on next page)

(continued from previous page)

```

7   identifier: sml2mqtt-ZqlFvhSBdDGvJ
8   host: localhost
9   port: 1883
10  user: ''
11  password: ''
12  topic prefix: sml2mqtt
13  defaults:
14      qos: 0          # Default value for QOS if no other QOS value in the config entry is_
↪set
15      retain: false  # Default value for retain if no other retain value in the config_
↪entry is set
16      last will:
17          topic: status  # Topic fragment for building this topic with the parent topic
18
19  general:
20      Wh in kWh: true    # Automatically convert Wh to kWh
21      republish after: 120 # Republish automatically after this time (if no other filter_
↪configured)
22
23  inputs:
24      - type: serial
25        url: COM1      # Device path
26        timeout: 3     # Seconds after which a timeout will be detected (default=3)
27      - type: serial
28        url: /dev/ttyS0 # Device path
29        timeout: 3     # Seconds after which a timeout will be detected (default=3)
30
31  devices:
32      # Device configuration by reported id
33      device_id_hex:
34
35          mqtt:          # Optional MQTT configuration for this meter.
36              topic: DEVICE_BASE_TOPIC # Topic fragment for building this topic with the_
↪parent topic
37
38          status:        # Optional MQTT status topic configuration for this_
↪meter
39              topic: status # Topic fragment for building this topic with the_
↪parent topic
40
41          skip:          # OBIS codes (HEX) of values that will not be_
↪published (optional)
42              - '00112233445566'
43
44          # Configurations for each of the values (optional)
45          values:
46
47              - obis: '00112233445566' # Obis code for this value
48                mqtt:                  # Mqtt config for this value (optional)
49                    topic: OBIS_VALUE_TOPIC # Topic fragment for building this topic with the_
↪topic prefix
50                # A sequence of operations that will be evaluated one after another.

```

(continues on next page)

(continued from previous page)

```

51      # If one operation blocks nothing will be reported for this frame
52      operations:
53      - negative on energy meter status: true    # Make value negative based on an
↪energy meter status. Set to "true" to enable or to "false" to disable workaround. If
↪the default obis code for the energy meter is wrong set to the appropriate meter obis
↪code instead
54      - factor: 3                                # Factor with which the value gets multiplied
55      - offset: 100                              # Offset that gets added on the value
56      - round: 2                                  # Round to the specified digits
57      - refresh action: 300                      # Republish value every 300s

```

## 4.4 4. Run with analyze

Now run `sml2mqtt` with the path to the configuration file and the `--analyze` option. (see [command line interface](#)). This will process one sml frame from the meter and report the output. It's a convenient way to check what values will be reported. It will also show how the configuration changes the reported values when you add an operation.

Check if the meter reports the serial number under obis `0100000009ff`. Example output for the meter data:

```

SmlMessage
  transaction_id: 17c77d6b
  group_no      : 0
  abort_on_error: 0
  message_body <SmlOpenResponse>
    codepage : None
    client_id : None
    req_file_id: 07ed29cd
    server_id : 11111111111111111111
    ref_time  : None
    sml_version: None
  crc16 : 25375
SmlMessage
  transaction_id: 17c77d6c
  group_no      : 0
  abort_on_error: 0
  message_body <SmlGetListResponse>
    client_id : None
    sever_id  : 11111111111111111111
    list_name  : 0100620affff
    act_sensor_time : 226361515
    val_list: list
      <SmlListEntry>
        obis      : 8181c78203ff
        status    : None
        val_time  : None
        unit      : None
        scaler    : None
        value     : ISK
        value_signature: None
      -> (Hersteller-Identifikation)
    <SmlListEntry>

```

(continues on next page)



(continued from previous page)

```

      obis      : 0100000009ff
      status    : None
      val_time  : None
      unit      : None
      scaler    : None
      value     : 111111111111111111
      value_signature: None
      -> (Geräteeinzelidentifikation)
    <SmlListEntry>
      obis      : 0100010800ff
      status    : 386
      val_time  : None
      unit      : 30
      scaler    : -1
      value     : 123456789
      value_signature: None
      -> 12345678.9Wh (Zählerstand Total)
    <SmlListEntry>
      obis      : 0100010801ff
      status    : None
      val_time  : None
      unit      : 30
      scaler    : -1
      value     : 123456789
      value_signature: None
      -> 12345678.9Wh (Zählerstand Tarif 1)
    <SmlListEntry>
      obis      : 0100010802ff
      status    : None
      val_time  : None
      unit      : 30
      scaler    : -1
      value     : 0
      value_signature: None
      -> 0.0Wh (Zählerstand Tarif 2)
    <SmlListEntry>
      obis      : 0100100700ff
      status    : None
      val_time  : None
      unit      : 27
      scaler    : 0
      value     : 555
      value_signature: None
      -> 555W (aktuelle Wirkleistung)
    <SmlListEntry>
      obis      : 8181c78205ff
      status    : None
      val_time  : None
      unit      : None
      scaler    : None
      value     :
    XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

```

(continues on next page)

(continued from previous page)

```

        value_signature: None
        -> (Öffentlicher Schlüssel)
    list_signature : None
    act_gateway_time: None
    crc16          : 22117
SmlMessage
    transaction_id: 17c77d6d
    group_no      : 0
    abort_on_error: 0
    message_body <SmlCloseResponse>
        global_signature: None
    crc16          : 56696

```

If the meter does not report `01000000009ff` it's possible to configure another number (of even multiple ones) for configuration matching (see *general under config*).

## 4.5 5. Edit device settings

Replace `device_id_hex` in the dummy configuration with the reported number (here `11111111111111111111`). Edit the mqtt settings or remove them to use the default. Add the obis code of values that should not be reported to the skip section. Run the analyze command again to see how the reported values change.

```

1  # ...
2
3  inputs:
4  - type: serial
5    url: COM1 # Device path
6    timeout: 3 # Seconds after which a timeout will be detected (default=3)
7  - type: serial
8    url: /dev/ttyS0 # Device path
9    timeout: 3 # Seconds after which a timeout will be detected (default=3)
10
11 devices:
12   # Device configuration by reported id
13   '11111111111111111111':
14
15     mqtt: # Optional MQTT configuration for this meter.
16       topic: meter_light # Topic fragment for building this topic with the
17   ↪parent topic
18
19     status: # Optional MQTT status topic configuration for this
20   ↪meter
21     topic: status # Topic fragment for building this topic with the
22   ↪parent topic
23
24     skip: # OBIS codes (HEX) of values that will not be
25   ↪published (optional)
26     - '8181c78205ff'
27
28   # Configurations for each of the values (optional)
29   values:

```

(continues on next page)

(continued from previous page)

```

26 - obis: '00112233445566'      # Obis code for this value
27   mqtt:                      # Mqtt config for this value (optional)
28     topic: OBIS_VALUE_TOPIC   # Topic fragment for building this topic with the
29 ↪topic prefix
30     # A sequence of operations that will be evaluated one after another.
31     # If one operation blocks nothing will be reported for this frame
32     operations:
33       - negative on energy meter status: true  # Make value negative based on an
34 ↪energy meter status. Set to "true" to enable or to "false" to disable workaround. If
35 ↪the default obis code for the energy meter is wrong set to the appropriate meter obis
36 ↪code instead
37       - factor: 3              # Factor with which the value gets multiplied
38       - offset: 100            # Offset that gets added on the value
39       - round: 2               # Round to the specified digits
40       - refresh action: 300    # Republish value every 300s

```

## 4.6 6. Edit value settings

It's possible to further configure how values will be reported. For every value there are multiple operations that can be applied. Each sml value can also be processed multiple times.

Run the analyze command again to see how the reported values change.

```

1  # ...
2
3  inputs:
4  - type: serial
5    url: COM1      # Device path
6    timeout: 3     # Seconds after which a timeout will be detected (default=3)
7  - type: serial
8    url: /dev/ttyS0 # Device path
9    timeout: 3     # Seconds after which a timeout will be detected (default=3)
10
11 devices:
12   # Device configuration by reported id
13   '11111111111111111111':
14
15     mqtt:                      # Optional MQTT configuration for this meter.
16     topic: meter_light         # Topic fragment for building this topic with the
17 ↪parent topic
18
19     status:                   # Optional MQTT status topic configuration for this
20 ↪meter
21     topic: status             # Topic fragment for building this topic with the
22 ↪parent topic
23
24     skip:                     # OBIS codes (HEX) of values that will not be
25 ↪published (optional)
26     - '8181c78205ff'

```

(continues on next page)

(continued from previous page)

```

24  # Configurations for each of the values (optional)
25  values:
26
27  - obis: '0100010800ff' # Obis code for the energy value
28    mqtt:
29      topic: energy_today
30    operations:
31      - type: meter # A virtual meter
32        start now: true # Start immediately
33        reset times: # Reset at midnight
34          - 00:00
35      - round: 1
36      - type: change filter # Only report on changes
37      - refresh action: 01:00 # ... but refresh every hour
38
39  - obis: '0100010800ff' # Obis code for the energy value
40    mqtt:
41      topic: energy_total
42    operations:
43      - round: 1
44      - type: change filter
45      - refresh action: 01:00
46
47  - obis: '0100100700ff' # Obis code for the power value
48    mqtt:
49      topic: power
50    operations:
51      - type: delta filter
52        min: 10
53        min %: 5
54      - refresh action: 01:00

```

Output from the analyze command that shows what values will be reported

```

...
sml2mqtt/meter_light/energy_today: 0 (QOS: 0, retain: False)
sml2mqtt/meter_light/energy_total: 12345.7 (QOS: 0, retain: False)
sml2mqtt/meter_light/power: 555 (QOS: 0, retain: False)
sml2mqtt/meter_light/status: OK (QOS: 0, retain: False)
...

```

## OPERATIONS

It's possible to define operations which are used to process the received value

### 5.1 Filters

#### 5.1.1 Change Filter

**settings OnChangeFilter**

A filter which lets the value only pass when it's different from the value that was passed the last time

**field type:** `Literal['change filter']` [Required]

Filter which passes only changes

**Example**

```
type: change filter
```

#### 5.1.2 Range filter

**settings RangeFilter**

Filters or limits to values that are in a certain range

**field min:** `float | None = None`

minimum value that will pass

**field max:** `float | None = None`

maximum value that will pass

**field limit:** `bool = False`

Instead of ignoring the values they will be limited to min/max

### Example

```
type: range filter
min: 0
```

## 5.1.3 Delta Filter

### settings DeltaFilter

A filter which lets the value only pass if the incoming value is different enough from value that was passed the last time. The delta can an absolute value or as a percentage. If multiple deltas are specified they are all checked.

```
field type: Literal['delta filter'] [Required]
field min: StrictInt | StrictFloat | None = None
field min %: StrictInt | StrictFloat | None = None
```

### Example

```
type: delta filter
min: 5
min %: 10
```

```
type: delta filter
min: 5
```

```
type: delta filter
min %: 10
```

## 5.1.4 Throttle Filter

### settings ThrottleFilter

Filter which only lets one value pass in the defined period. If the last passed value is not at least `period` old any new value will not be forwarded.

```
field throttle filter: DurationType [Required]
    Throttle period
```

### Example

```
throttle filter: 60
```

## 5.2 Actions

### 5.2.1 Refresh Action

#### settings RefreshAction

Action which lets every value pass. When no value is received (e.g. because an earlier filter blocks) this action will produce the last received value every interval.

**field refresh action:** **DurationType** [Required]

Refresh interval

#### Example

```
refresh action: 01:30:00
```

### 5.2.2 Heartbeat Action

#### settings HeartbeatAction

Action which lets a value pass periodically every specified interval. When no value is received (e.g. because an earlier filter blocks) this action will produce the last received value every interval.

**field heartbeat action:** **DurationType** [Required]

Interval

#### Example

```
heartbeat action: 30
```

## 5.3 Math

### 5.3.1 Factor

#### settings Factor

**field factor:** **Number** [Required]

Factor with which the value gets multiplied

### Example

```
factor: -1
```

## 5.3.2 Offset

### settings Offset

**field offset:** Number [Required]

Offset that gets added on the value

### Example

```
offset: 10
```

## 5.3.3 Round

### settings Round

**field round:** int [Required]

Round to the specified digits

#### Constraints

- ge = 0
- le = 6

### Example

```
round: 2
```

## 5.4 Workarounds

### 5.4.1 Negative On Energy Meter Status

#### settings NegativeOnEnergyMeterWorkaround

Make value negative based on an energy meter status.

**field negative on energy meter status:** StrictBool | ObisHex [Required]

Set to “true” to enable or to “false” to disable workaround. If the default obis code for the energy meter is wrong set to the appropriate meter obis code instead



### Example

```
negative on energy meter status: true
```

## 5.5 Date time based

### 5.5.1 Virtual Meter

#### settings VirtualMeter

A virtual meter. It will output the difference from the last reset

**field start now: bool [Required]**

Immediately start instead of starting after the next reset

**field reset times: list[time] = []**

Time(s) of day when a reset will occur

**field reset days: list[DayOfMonth | DayOfWeekStr] = []**

Days of month or weekdays where the time(s) will be checked

### Example

```
type: meter
start now: False
reset times:
  - 02:00
reset days:
  - 1
  - monday
```

### 5.5.2 Max Value

#### settings MaxValue

Maximum value since last reset

**field start now: bool [Required]**

Immediately start instead of starting after the next reset

**field reset times: list[time] = []**

Time(s) of day when a reset will occur

**field reset days: list[DayOfMonth | DayOfWeekStr] = []**

Days of month or weekdays where the time(s) will be checked

### Example

```
type: max value
start now: True
reset times:
  - 02:00
```

## 5.5.3 Min Value

### settings MinValue

Minimum value since last reset

**field start now: bool [Required]**

Immediately start instead of starting after the next reset

**field reset times: list[time] = []**

Time(s) of day when a reset will occur

**field reset days: list[DayOfMonth | DayOfWeekStr] = []**

Days of month or weekdays where the time(s) will be checked

### Example

```
type: min value
start now: True
reset times:
  - 02:00
```

## 5.6 Time series

### 5.6.1 Max Value

#### settings MaxOfInterval

Maximum value in a sliding interval

**field interval: timedelta [Required]**

Interval duration

**field wait for data: bool [Required]**

Only produce a value when data for the whole interval is available

**field reset after value: bool = False**

Clear all data as soon as a value has been produced

### Example

```

type: max interval
interval: 3600
wait for data: False

```

## 5.6.2 Min Value

### settings MinOfInterval

Minimum value in a sliding interval

**field interval:** `timedelta` [Required]

Interval duration

**field wait for data:** `bool` [Required]

Only produce a value when data for the whole interval is available

**field reset after value:** `bool = False`

Clear all data as soon as a value has been produced

### Example

```

type: min interval
interval: 3600
wait for data: False

```

## 5.6.3 Mean Value

### settings MeanOfInterval

Weighted mean in a sliding interval

**field interval:** `timedelta` [Required]

Interval duration

**field wait for data:** `bool` [Required]

Only produce a value when data for the whole interval is available

**field reset after value:** `bool = False`

Clear all data as soon as a value has been produced

### Example

```

type: mean interval
interval: 3600
wait for data: False

```

## 5.7 Operations

### 5.7.1 Or

#### settings Or

A sequence of operations that will be evaluated one after another. The first value that gets returned by an operation will be used.

**field or:** `OperationsListType` [Required]

##### Constraints

- `min_length` = 1

#### Example

```
or:  
- type: change filter  
- heartbeat action: 60
```

### 5.7.2 Sequence

#### settings Sequence

A sequence of operations that will be evaluated one after another. If one operation blocks this will return nothing.

**field sequence:** `OperationsListType` [Required]

##### Constraints

- `min_length` = 1

#### Example

```
sequence:  
- factor: 0.1  
- offset: -50
```

## 5.8 Examples

These are some examples for sml value configurations

### 5.8.1 Energy consumption today

This will report the power consumption of today. The first reported value every day will be 0 and then it will increase for every day.

```
obis: '0100010800ff'    # Obis code for the energy meter
mqtt:
  topic: energy_today    # MQTT topic for the meter
operations:
- type: meter
  start now: true        # Start immediately
  reset times:           # Reset at midnight
    - 00:00
- round: 1
- type: change filter    # Only report on changes
- refresh action: 01:00  # ... but refresh every hour
```

### 5.8.2 Downsample current power

This will report a power value every max every 30s. The reported value will be the weighted mean value of the last 30s.

```
obis: '0100100700ff'    # Obis code for the energy meter
mqtt:
  topic: power           # MQTT topic for the meter
operations:
- type: mean interval    # Calculate weighted mean over 30s
  interval: 30
  wait for data: False
- throttle filter: 30    # Let a value pass every 30s
- round: 0               # Round the mean value to the full number
- type: delta filter     # Only report when the value changes at least 10W or 5%
  min: 10
  min %: 5
- refresh action: 01:00  # ... but refresh every hour
```



## INDICES AND TABLES

- `genindex`
- `modindex`
- `search`





## B

baudrate (*SerialSourceSettings* attribute), 10  
 bytesize (*SerialSourceSettings* attribute), 10

## C

ca\_certs (*MqttTlsOptions* attribute), 12  
 cert\_file (*MqttTlsOptions* attribute), 13  
 cert\_reqs (*MqttTlsOptions* attribute), 13  
 ciphers (*MqttTlsOptions* attribute), 13  
 connection (*MqttConfig* attribute), 11

## D

defaults (*MqttConfig* attribute), 11  
 device\_id\_obis (*GeneralSettings* attribute), 10  
 devices (*Settings* attribute), 9  
 digits (*Round* attribute), 28

## E

enabled\_or\_obis (*NegativeOnEnergyMeter-Workaround* attribute), 28  
 every (*HeartbeatAction* attribute), 27  
 every (*RefreshAction* attribute), 27

## F

factor (*Factor* attribute), 27  
 file (*LoggingSettings* attribute), 9  
 file\_password (*MqttTlsOptions* attribute), 13  
 full\_topic (*OptionalMqttPublishConfig* attribute), 12

## G

general (*Settings* attribute), 9

## H

host (*MqttConnection* attribute), 11

## I

identifier (*MqttConnection* attribute), 11  
 inputs (*Settings* attribute), 9  
 insecure (*MqttTlsOptions* attribute), 12  
 interval (*HttpSourceSettings* attribute), 11  
 interval (*MaxOfInterval* attribute), 30

interval (*MeanOfInterval* attribute), 31  
 interval (*MinOfInterval* attribute), 31

## K

key\_file (*MqttTlsOptions* attribute), 13

## L

last\_will (*MqttConfig* attribute), 11  
 level (*LoggingSettings* attribute), 9  
 limit\_values (*RangeFilter* attribute), 25  
 logging (*Settings* attribute), 9

## M

max\_value (*RangeFilter* attribute), 25  
 min\_percent (*DeltaFilter* attribute), 26  
 min\_value (*DeltaFilter* attribute), 26  
 min\_value (*RangeFilter* attribute), 25  
 mqtt (*Settings* attribute), 9  
 mqtt (*SmlDeviceConfig* attribute), 13  
 mqtt (*SmlValueConfig* attribute), 13

## O

obis (*SmlValueConfig* attribute), 13  
 offset (*Offset* attribute), 28  
 operations (*Or* attribute), 32  
 operations (*Sequence* attribute), 32  
 operations (*SmlValueConfig* attribute), 13

## P

parity (*SerialSourceSettings* attribute), 10  
 password (*HttpSourceSettings* attribute), 11  
 password (*MqttConnection* attribute), 12  
 period (*ThrottleFilter* attribute), 26  
 port (*MqttConnection* attribute), 12

## Q

qos (*MqttDefaultPublishConfig* attribute), 12  
 qos (*OptionalMqttPublishConfig* attribute), 12

## R

report\_blank\_energy\_meters (*GeneralSettings* attribute), 10

report\_device\_id (*GeneralSettings* attribute), 10  
republish\_after (*GeneralSettings* attribute), 10  
request\_timeout (*HttpSourceSettings* attribute), 11  
reset\_after\_value (*MaxOfInterval* attribute), 30  
reset\_after\_value (*MeanOfInterval* attribute), 31  
reset\_after\_value (*MinOfInterval* attribute), 31  
reset\_days (*MaxValue* attribute), 29  
reset\_days (*MinValue* attribute), 30  
reset\_days (*VirtualMeter* attribute), 29  
reset\_times (*MaxValue* attribute), 29  
reset\_times (*MinValue* attribute), 30  
reset\_times (*VirtualMeter* attribute), 29  
retain (*MqttDefaultPublishConfig* attribute), 12  
retain (*OptionalMqttPublishConfig* attribute), 12

## S

skip (*SmlDeviceConfig* attribute), 13  
start\_now (*MaxValue* attribute), 29  
start\_now (*MinValue* attribute), 30  
start\_now (*VirtualMeter* attribute), 29  
status (*SmlDeviceConfig* attribute), 13  
stopbits (*SerialSourceSettings* attribute), 10

## T

timeout (*HttpSourceSettings* attribute), 11  
timeout (*SerialSourceSettings* attribute), 10  
tls (*MqttConnection* attribute), 12  
tls\_version (*MqttTlsOptions* attribute), 13  
topic (*MqttConfig* attribute), 11  
topic (*OptionalMqttPublishConfig* attribute), 12  
type (*DeltaFilter* attribute), 26  
type (*HttpSourceSettings* attribute), 10  
type (*OnChangeFilter* attribute), 25  
type (*SerialSourceSettings* attribute), 10

## U

url (*HttpSourceSettings* attribute), 10  
url (*SerialSourceSettings* attribute), 10  
user (*HttpSourceSettings* attribute), 11  
user (*MqttConnection* attribute), 12

## V

values (*SmlDeviceConfig* attribute), 13

## W

wait\_for\_data (*MaxOfInterval* attribute), 30  
wait\_for\_data (*MeanOfInterval* attribute), 31  
wait\_for\_data (*MinOfInterval* attribute), 31  
wh\_in\_kwh (*GeneralSettings* attribute), 10