Informatik Werkstatt
Analyse | Design | Entwicklung
In Cooperation with
KIPP & ZONEN

Operating Manual

UpCom®

Version 3.4.0 (04/2026)

Content

1 General

UpCom® is a software plugin for the Option Cloudgate (https://www.option.com/) router.

It retrieves measure values from various sensors, optionally stores this data on the device (on a micro SD card), and makes this data available through different "uplink" protocols.

Currently, following protocols are supported:

- ViewMondo® API
- OPC-UA (as specified by Swiss "Federal Roads Office FEDRO" on federal highways / Bundesamt für Strassen ASTRA)
- NTCIP ESS V4
- TLSoIP (as specified by German "Bundesanstalt für Straßen- und Verkehrswesen"
- CSV text file export (and transfer via (s)ftp(s)

1.1 Hardware requirements

The hardware required for this plugin is

- Option Cloudgate Router. Following models are supported/tested:
  - Option Cloudgate LTE (WW Rev 4) router (CG0124-12135) with additional Industrial Serial Card (CG1102-11920)
  - Option Cloudgate "Mini Bundle" (CM0125-12142) with build in Multi Purpose I/O Card (CM1123)
  - Option Cloudgate "Nano" (CM0126)
- „Microfit" 4-pin power cable (see Cloudgate documentation)
- DC Power supply (9-33V)

- sensor(s) that support UMB (Lufft), Modbus, or SDI-12 protocol
- And/or for SDI-12 sensors: an RS232/SDI-12 adapter (e.g. "TekBox TBS06-DR")

Depending on the specific installation and use case, following optional items might be needed:

- A micro SD card (**microSD/microSDHC**) up to 32GB – with FAT filesystem. Note: a „industrial" SD Card is recommended, e.g. SanDisk Industrial MLC, or Advantech Industrial microSD
- **NOTE: microSDXC cards are NOT supported!**
- A sim card
- Antenna for cellular network (possibly combined with GPS antenna)
- DIN-Rail mounting kit

2 Connecting UMB devices

The UMB bus is connected via the RS485 interface on the CloudGate devices "industrial serial card" (green connector on the left side) using the pins:

TX+ = A (green)

TX- = B (yellow)

Switch settings:

Wires: 2W

Termination: On

Note: the connector on the CloudGate Nano is "upside down", i.e. turned by 180°

3 Connecting to the world

The "uplink" protocols are all TCP/IP based. The connection to the "world" (internet or private network) can be either through the cellular network or one of the RJ45 ethernet connectors on the Cloudgate.

To connect to the cellular network, a sim card needs to be installed

The sim card slot is located at the back of the CloudGate – in the middle next to the power connector

Please note that an antenna needs to be connected to the Cloudgate for cellular network connections. The antenna is connected to the respective connector on the front of the CloudGate device.

For Ethernet connection, the WAN and/or LAN RJ45 slot can be used. These are also located at the front of the device.

4 CloudGate Setup

For detailed information about the CloudGate hardware please refer to the Cloudgate Documentation at

https://cloudgateuniverse.com/docs/cloudgate-user-guide

https://cloudgateuniverse.com/docs/cloudgate-lte-ww-rev-4-cg0124

The CloudGate user interface is web based and can always be accessed through the LAN Ethernet interface.

The initial IP address for the LAN interface is 192.168.1.1

4.1 User and Password

Factory settings for user and password are

User: admin

Password: admin

NOTE: if your Cloudgate is delivered pre-configured with the UpCom software already installed, the password will not be the default password anymore. Please contact our support for more information.

If you setup the Cloudgate device “from scratch”, you will need to change the password via “System” “Username & Password”

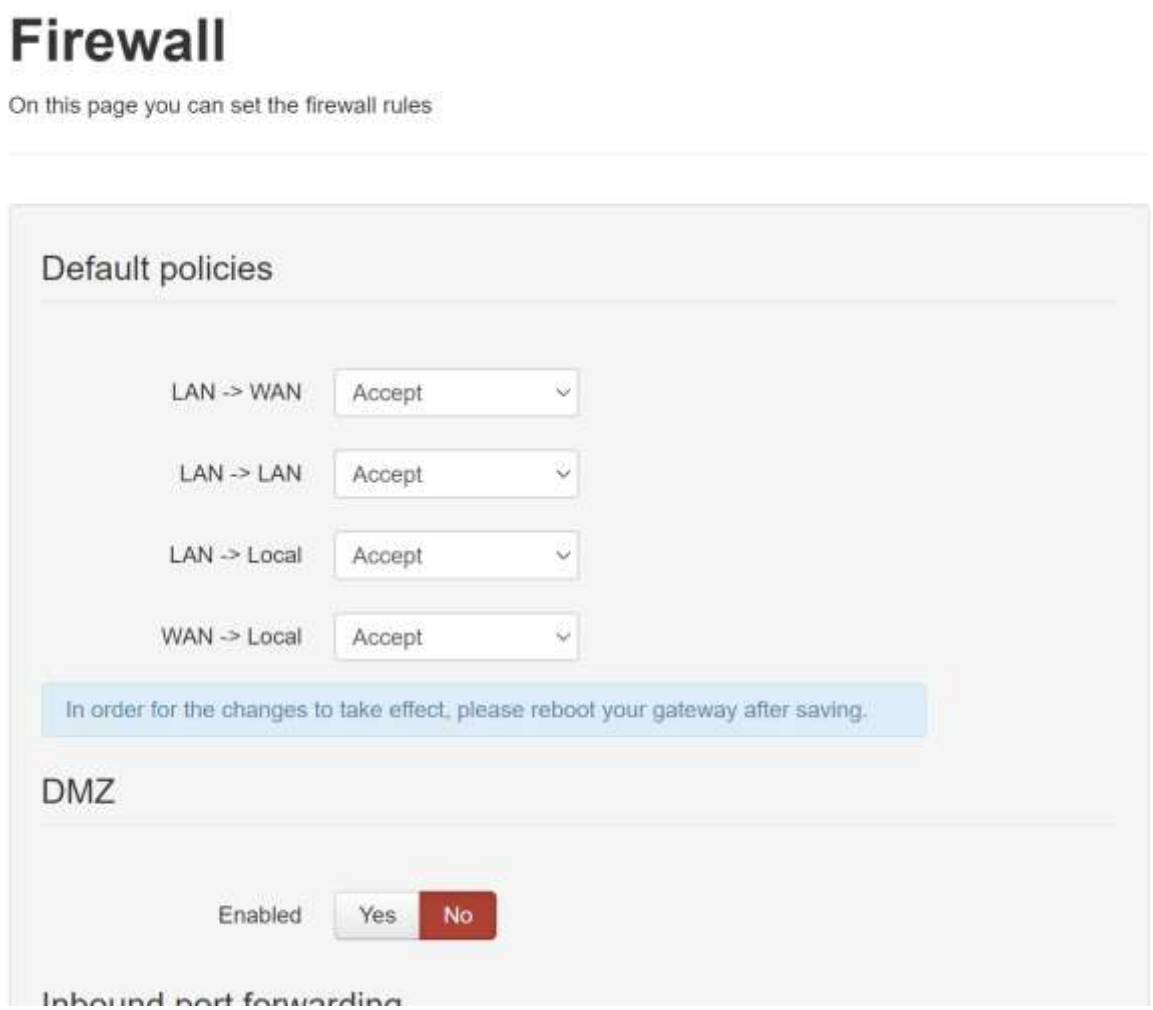
4.2 Interface settings

The cellular and/or ethernet adapter settings are configured in the “Interfaces” tab:

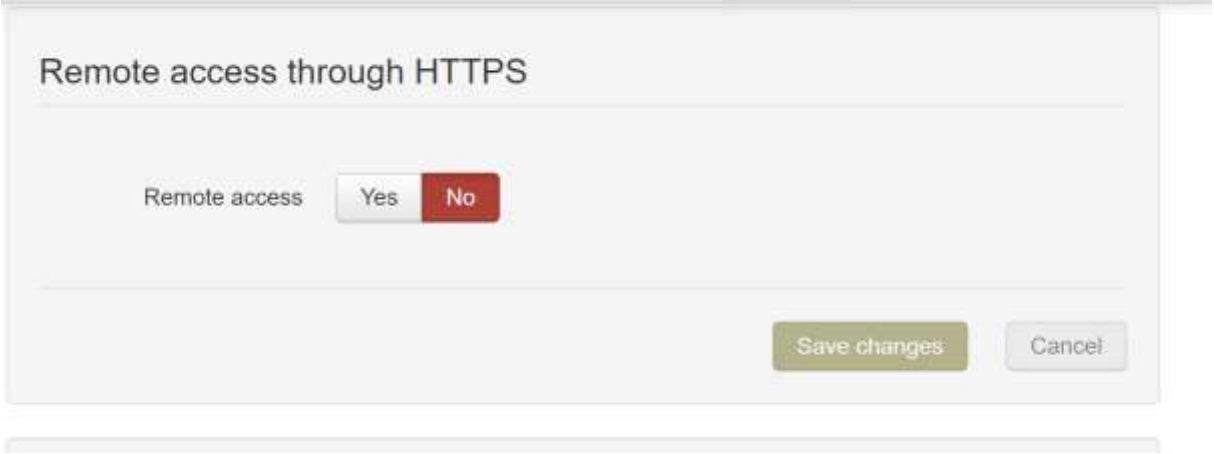
Both Ethernet interfaces can be set up as “LAN/WAN or PPOE”, and the usual IP settings can be configured for the interface (static/dynamic IP address, DHCP server etc.).

4.3 Firewall

Depending on your physical connection and the protocol used, and your security requirements, firewall settings might need to be changed to allow incoming IP connections to the CloudGate, e.g. set “WAN->Local” to “Accept”



It might also be necessary to allow "remote access through HTTPS" in the "system" tab:

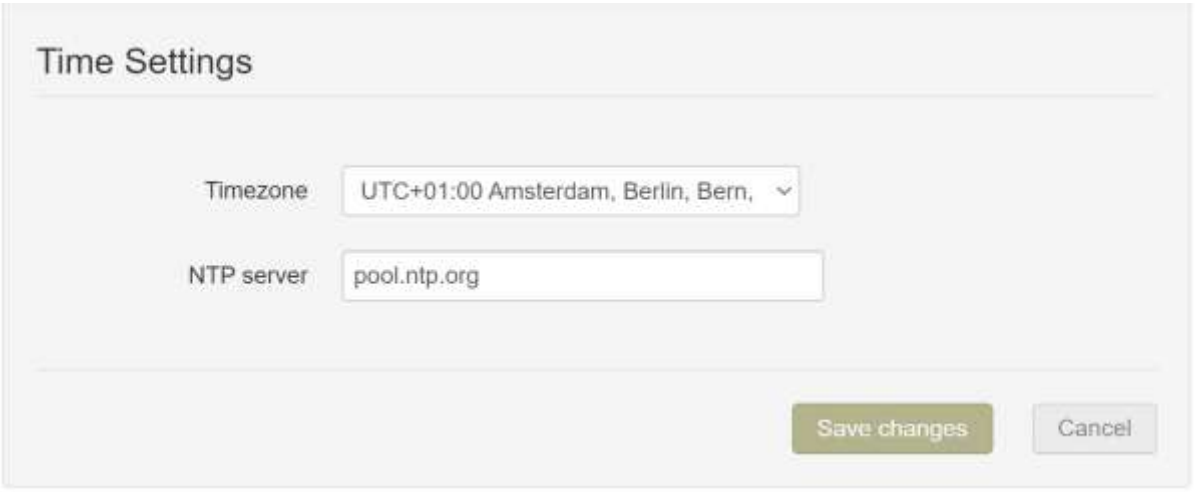


4.4 NTP Server

It is essential that the CloudGate clock is synchronized so that timestamps associated with the measure values are correct.

If the cloudgate is operated in a private network, the NTP server might have to be configured. Configuring the time zone is not essential, but also recommended.

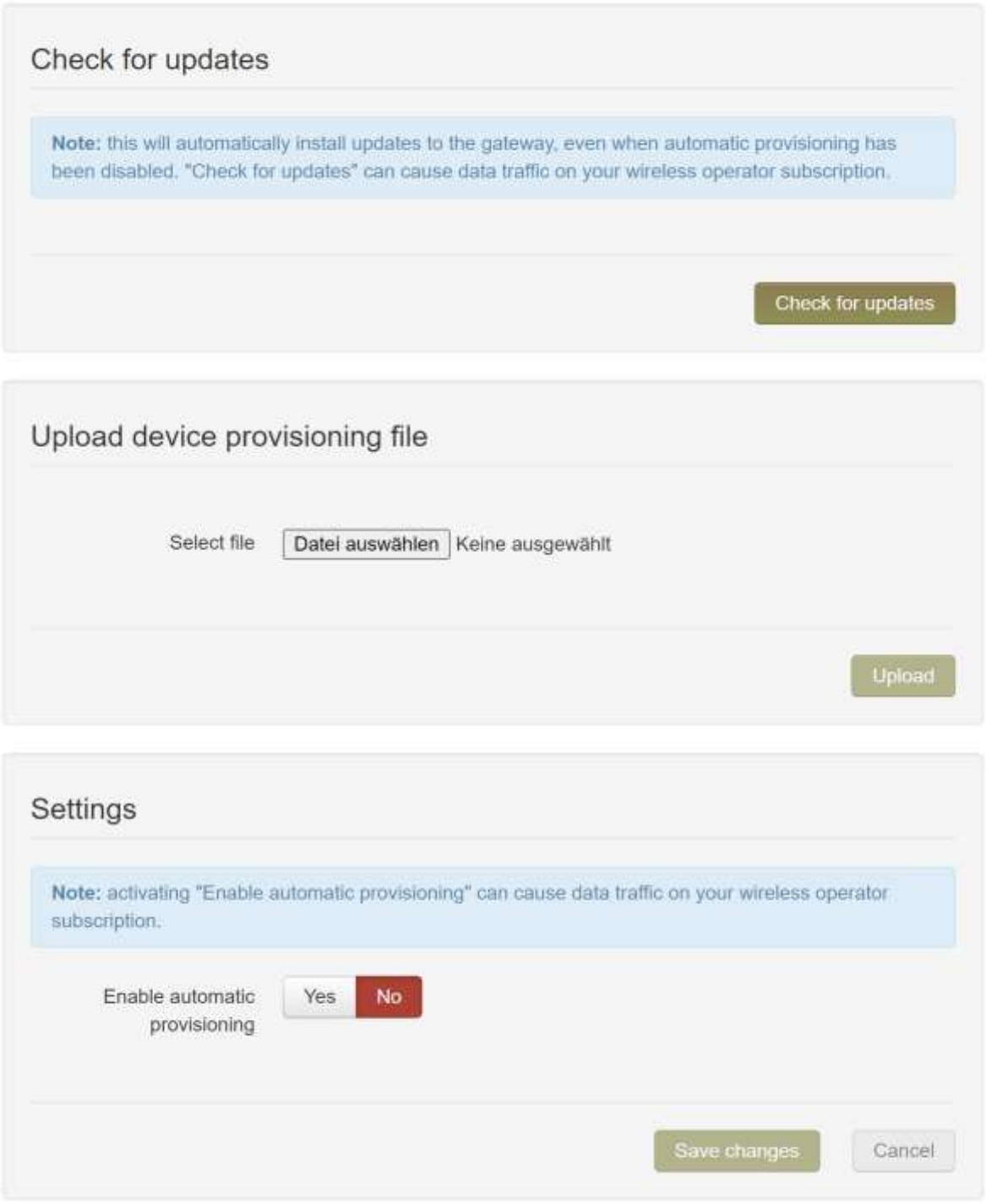
This can be set in the “System” tab – Time Settings



5 Installing/Updating UpCom Plugin

The “Provisioning” of software for the Cloudgate is configured in the “Provisioning” tab

Device Provisioning



Note: if your device is not managed via “CloudgateUniverse” (see below), “automatic provisioning” should be disabled to avoid accidental updates/resets via CloudGateUniverse, and “check for updates” should NOT be performed.

5.1 Cloudgate Universe

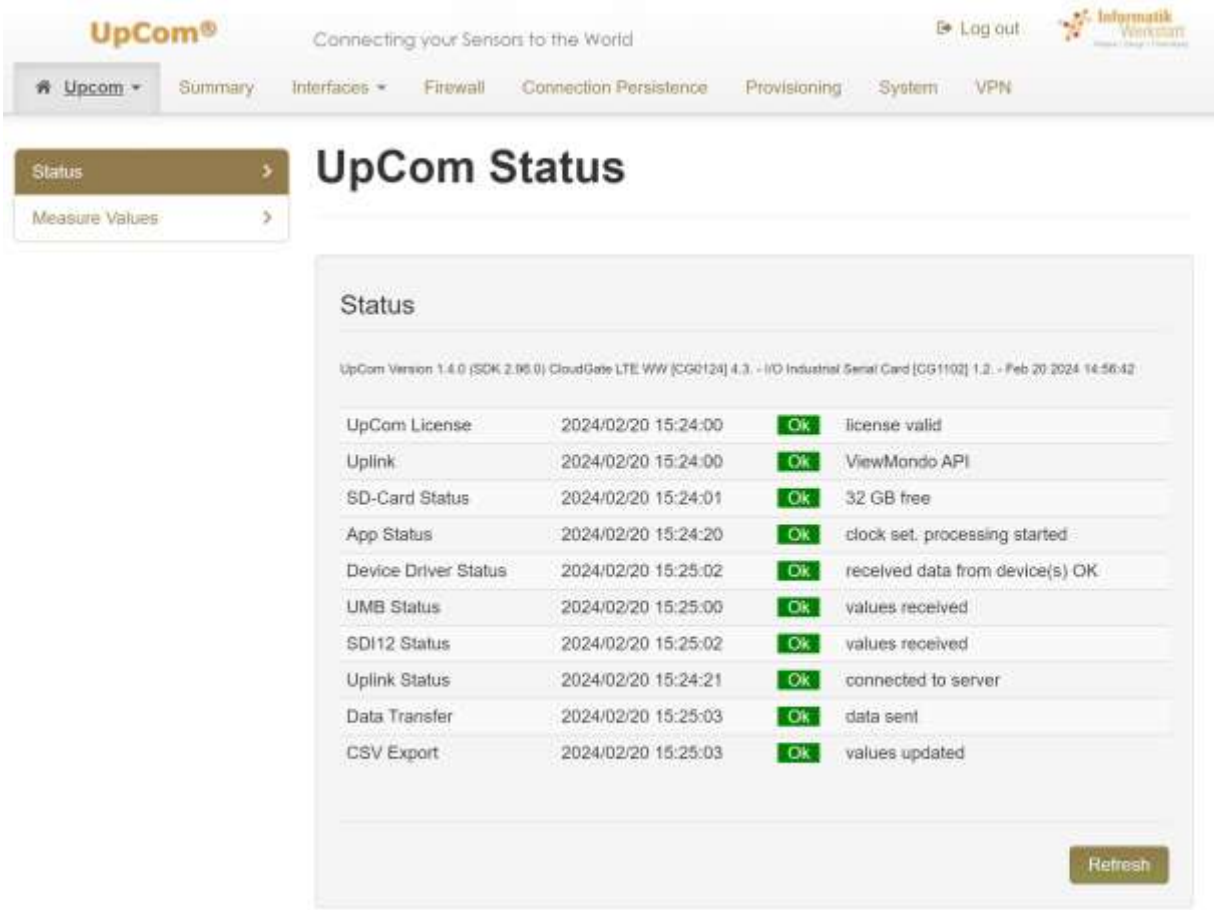
If your device is connected to the internet, it can be registered (or might already be registered) at "CloudgateUniverse". In this case, the UpCom plugin is configured for your device in the CloudGateUniverse settings and can be updated "automatically" if "automatic provisioning" is enabled or manually by "checking for updates"

5.2 Manual provisioning

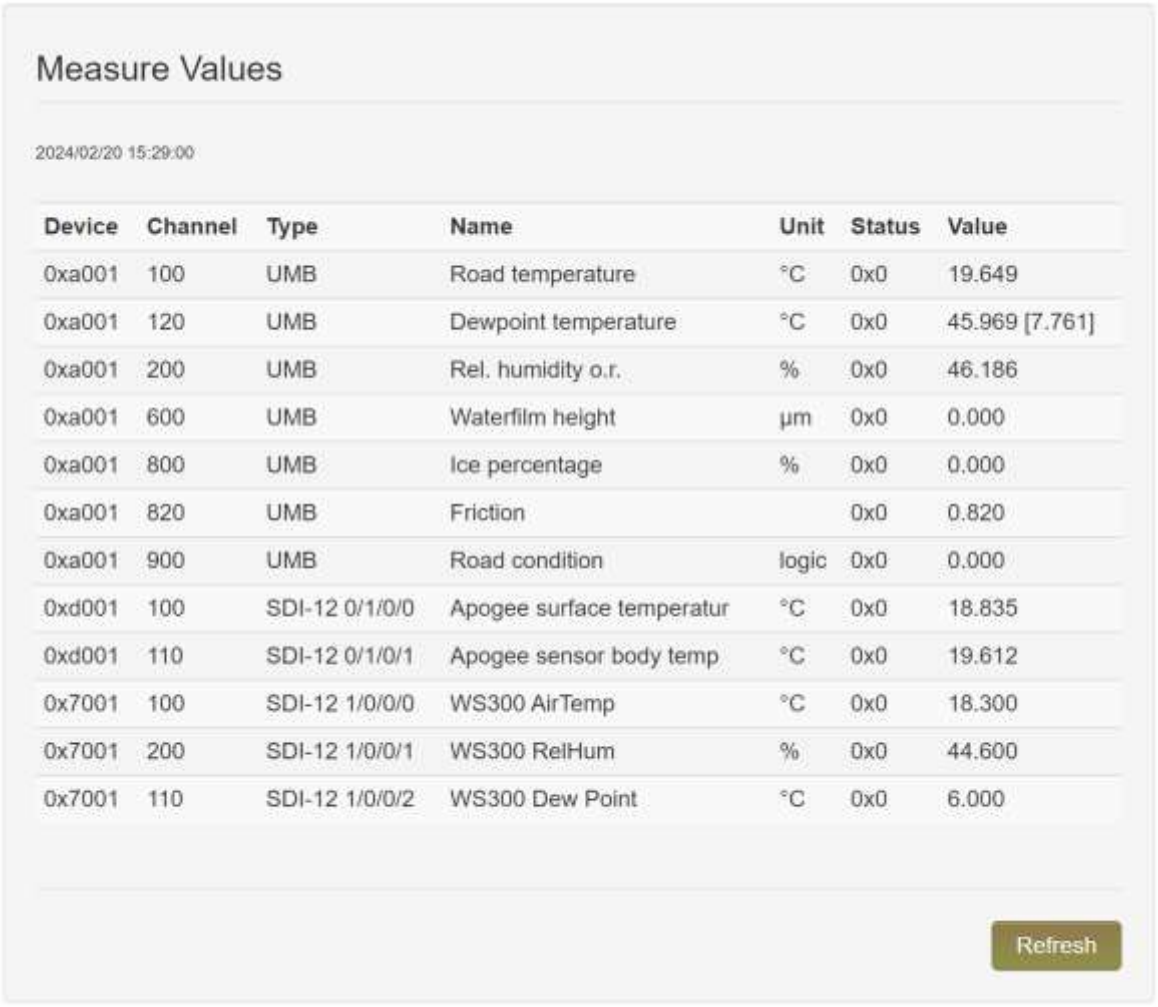
For manual provisioning, click the "select file" button and upload the UpCom plugin image provided by our support team to the device.

6 UpCom Status

This is shown right after logging on to the CloudGate and shows the overall status for the UpCom software



As well as the last measure values read from the configured UMB and SDI-12 device channels (see below) or the error status for the respective sensor channel



Measure Values

2024/02/20 15:29:00

Device	Channel	Type	Name	Unit	Status	Value
0xa001	100	UMB	Road temperature	°C	0x0	19.649
0xa001	120	UMB	Dewpoint temperature	°C	0x0	45.969 [7.761]
0xa001	200	UMB	Rel. humidity o.r.	%	0x0	46.186
0xa001	600	UMB	Waterfilm height	µm	0x0	0.000
0xa001	800	UMB	Ice percentage	%	0x0	0.000
0xa001	820	UMB	Friction		0x0	0.820
0xa001	900	UMB	Road condition	logic	0x0	0.000
0xd001	100	SDI-12 0/1/0/0	Apogee surface temperatur	°C	0x0	18.835
0xd001	110	SDI-12 0/1/0/1	Apogee sensor body temp	°C	0x0	19.612
0x7001	100	SDI-12 1/0/0/0	WS300 AirTemp	°C	0x0	18.300
0x7001	200	SDI-12 1/0/0/1	WS300 RelHum	%	0x0	44.600
0x7001	110	SDI-12 1/0/0/2	WS300 Dew Point	°C	0x0	6.000

Refresh

If there is a value mapping configured for a sensor channel, the original (raw) value is shown in square brackets.

For SDI-12 sensor channels, the SDI-12 configuration data (device id/measurement block/data block/value offset) is shown in addition to the mapped UMB device and channel number (see SDI-12 sensor configuration below).

7 Settings

7.1 Set Clock

Usually, the clock of the CloudGate router is set via NTP server when connecting to the internet, or can be set by many of the supported transmission protocols (NTCIP, TLSoIP, ViewMondo API).
In case the clock is not set by any of these ways, it can be set here. The time is taken from the browser – i.e. the device on which the browser is running and that is used to view the user interface.

The time shown here is the time as reported by the browser (i.e. the device the browser is running on) – not the CloudGate router! It is updated every second and can not be entered manually.

Clicking "set clock" will set the clock and restart the UpCom plugin.

Please note that the clock will still be set via NTP server and/or any uplink protocol if applicable.

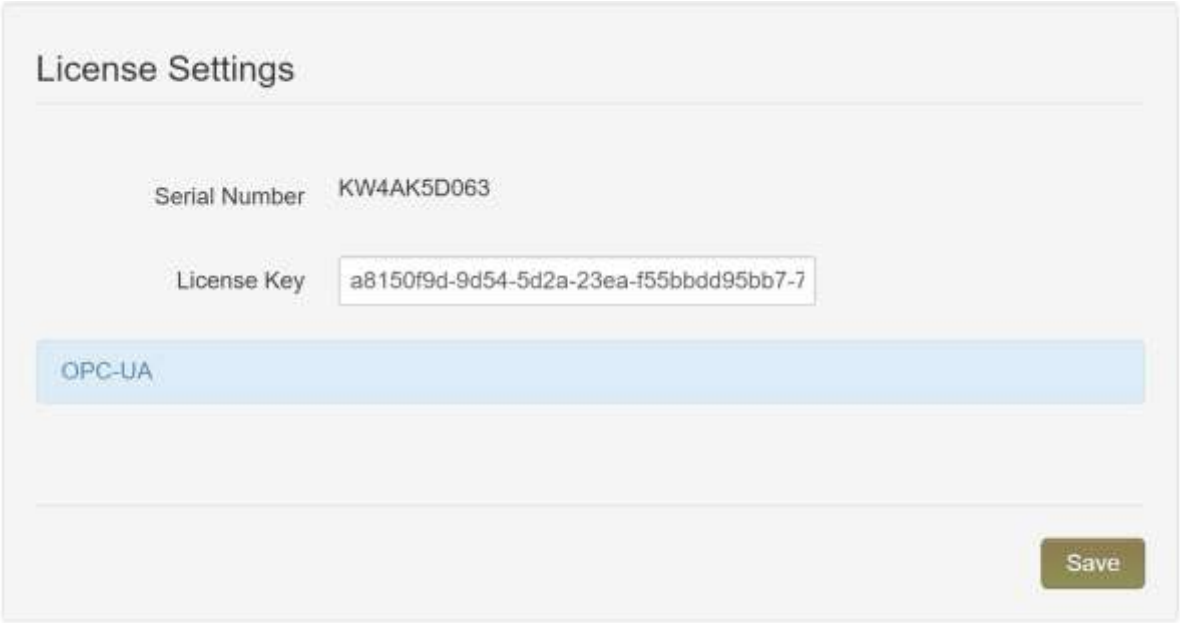
7.2 UpCom License

The features for the UpCom software are enabled by a license key. If no license key has been configured yet, please check with our support to receive a license key for your device.

The License key is configured via "Settings" in the "UpCom" tab on the main menu

Please provide the CloudGate serial number when requesting a license key.

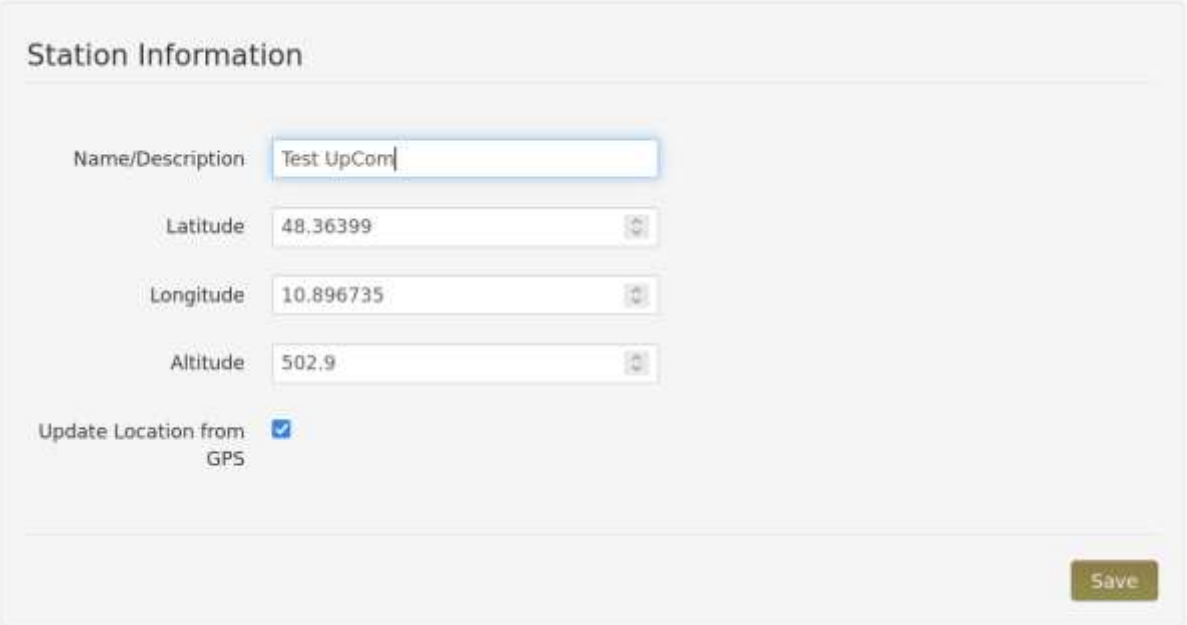
After entering a valid License key, the uplink protocol(s) included in the license are shown in the "License Settings"



7.3 Station Information

Here, general station information/meta data can be configured.

This data might be used by an uplink protocol (see below, e.g. NTCIP) if the protocol includes transmitting this kind of data



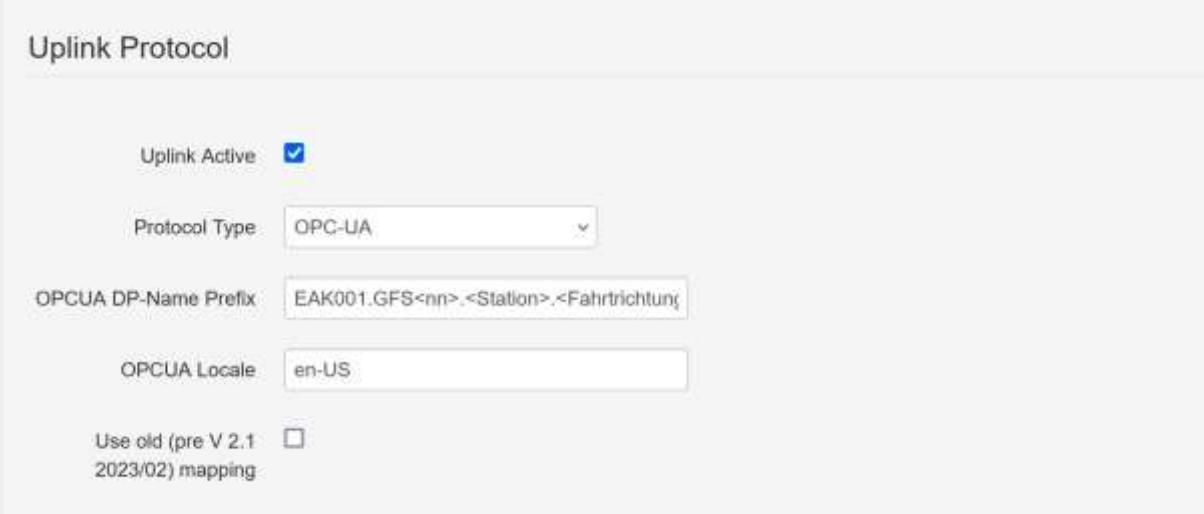
- Name/Description: a name/description for this station
- Latitude: the latitude
- Longitude: the longitude
- Altitude: the altitude
- Update Location from GPS: if GPS is supported and active (see below), this controls weather the location data (latitude/longitude/altitude) is set/updated form the data received by the GPS receiver.

7.4 Uplink Protocol

Here, the uplink protocol can be selected.

Depending on the uplink protocol type, other parameters are configured.

7.4.1.1 OPCUA



For OPCUA, a prefix for the data point name, and the OPCUA-Locale can be set. This prefix is used for any newly created data point – note that changing the prefix will NOT change the names for already configured data points.

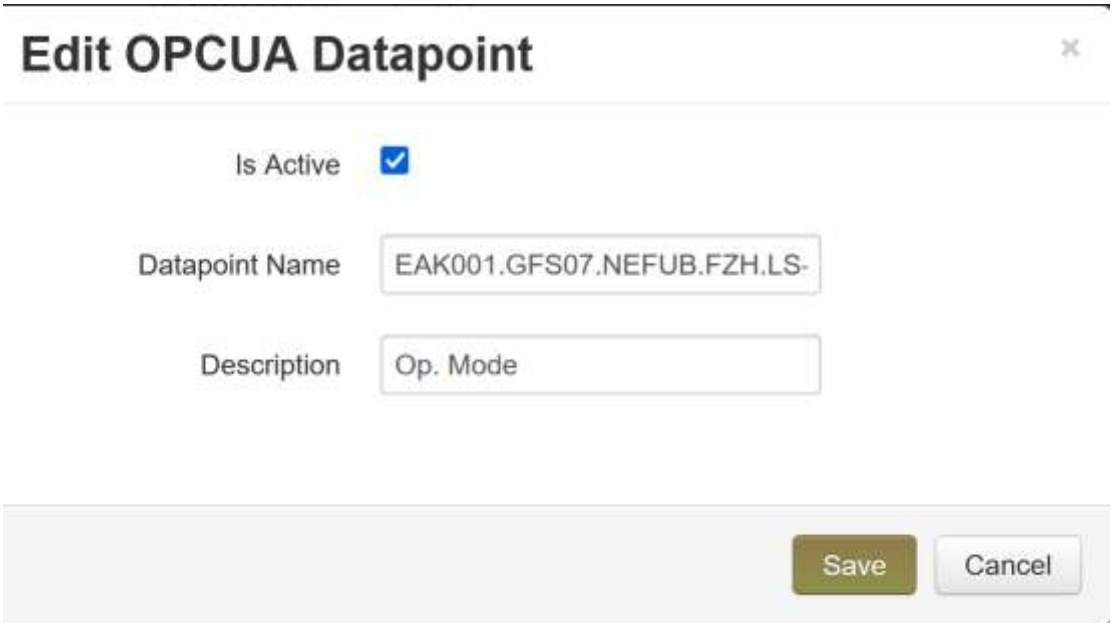
The OPCUA "data points" are automatically generated, based on the requirements as defined by the Swiss "Federal Roads Office FEDRO" on federal highways / Bundesamt für Strassen ASTRA.

The option "Use old (pre V 2.1 2023/02) mapping refers to the change in the encoding for road condition and precipitation type. In older versions of the ASTRA specification, a specific encoding for precipitation type and road condition was requested. From Version 2.1 2023/02 on, precipitation type and road condition are supposed to be encoded according to the German TLS standard (DE Type 70 and 71)

There are 6 "system" data points:

Type	Name	Description	Sensor Name	Action
☑	Reset	EAK001.GFS07.NEFUB.FZH.CMD-SET	Reset	
☑	Uptime Minutes	EAK001.GFS07.NEFUB.FZH.LS-RTM	Uptime Minutes	
☑	Com Error	EAK001.GFS07.NEFUB.FZH.LS-COM	Com Error	
☑	Op. Mode	EAK001.GFS07.NEFUB.FZH.LS-CTM	Op. Mode	
☑	System Status	EAK001.GFS07.NEFUB.FZH.LS-ERR	System Status	
☑	System Type	EAK001.GFS07.NEFUB.FZH.LS-OPM	System Type	

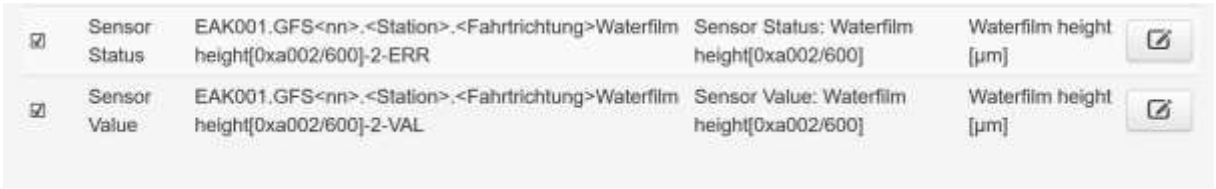
These data points can be edited:



The datapoint can be set active/inactive. Inactive data points will not be published/supported by the OPCUA-Server.

The datapoint name can be freely configured. Please note that for OPCUA a datapoint name must be unique within the respective device (server).

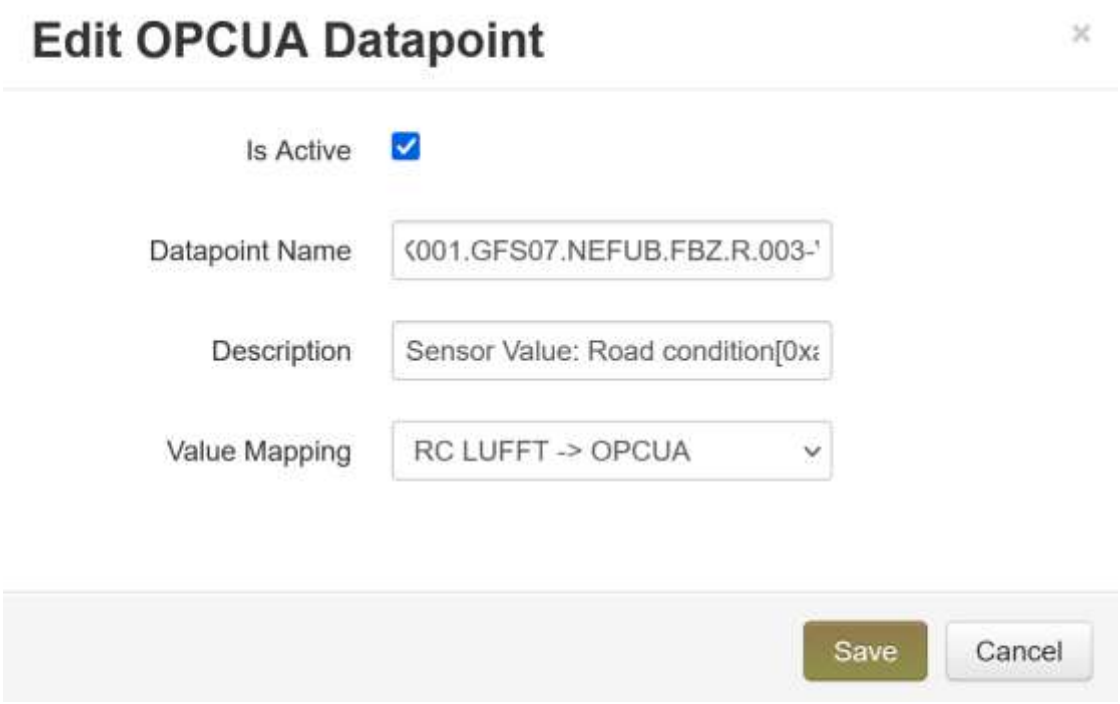
For each active UMB sensor channel 2 data points for status and value for the respective umb sensor channel:



The respective data point name is created by using the above data point prefix,appended with the sensor name and UMB-Device / Channel Nr.

The status datapoint ends with “-ERR”, the data point that actually supports the measure value ends in “-VAL”

For “value” data points, a value mapping can be set.

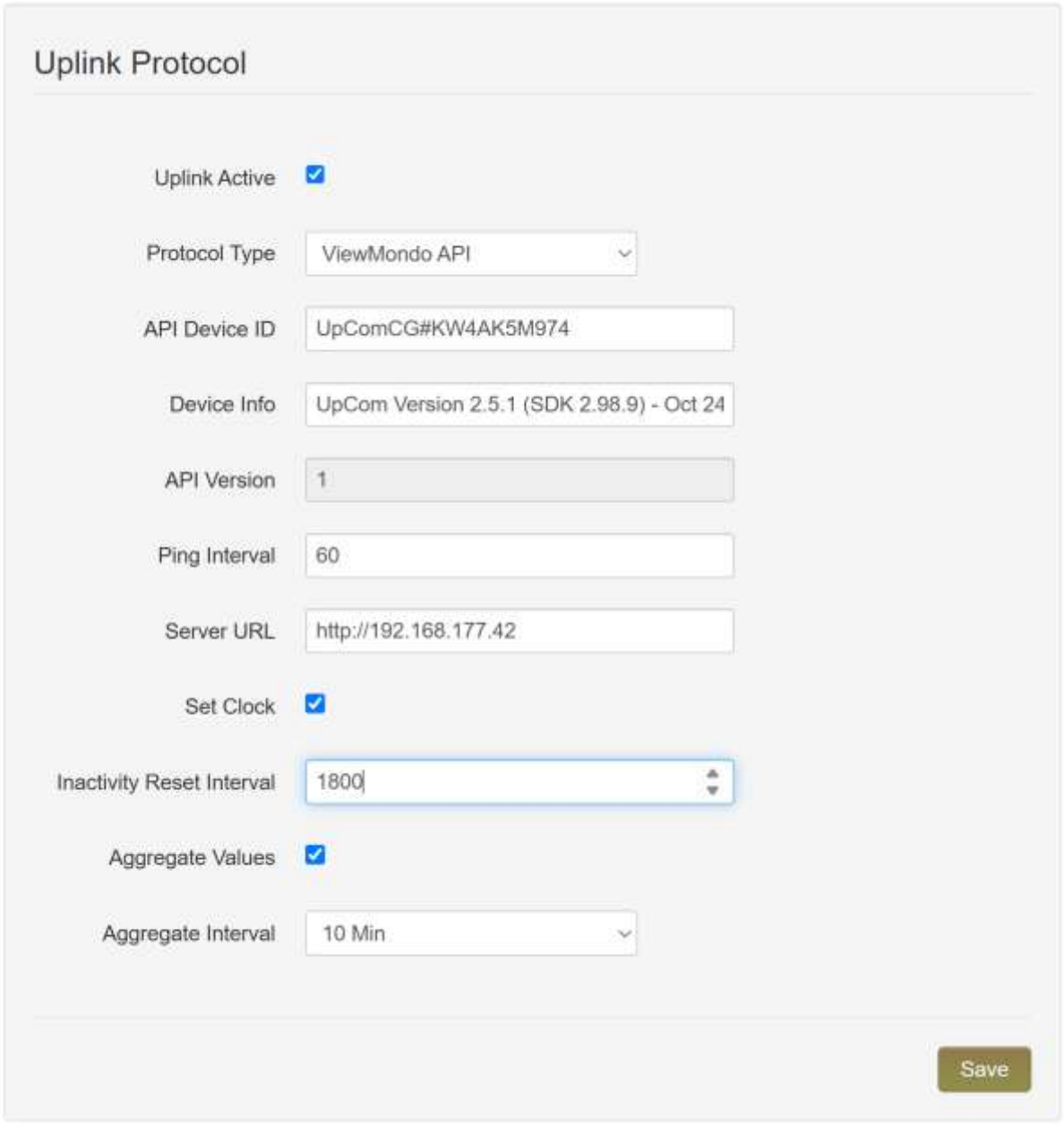


This might to be configured for precipitation type and road condition sensor channels. The mapping will be automatically pre-configured based on the requirements as defined by the Swiss "Federal Roads Office FEDRO" on federal highways / Bundesamt für Strassen ASTRA.

Note: if the OPC-UA Uplink is used for projects outside the the Swiss "Federal Roads Office FEDRO" on federal highways / Bundesamt für Strassen ASTRA – the data point names and value mapping might have to be configured differently, and you might want to deactive the "system status" datapoints and the "sensor status" (-ERR) datapoints to only present the actual measure values via OPC-UA

Note2: the encoding for precipitation type and road condition was changed in the respective document versions from ASTRA. Depending on which version of the protocol should be used, the "Value mapping" to OPCUA values might or might not be necessary. From specification Version 2.1 (2023/02) on, the encoding is supposed to be according to the German TLS protocol – DE Types 70 and 71, which is natively supported by the OttHydromet UMB sensors on the respective UMB sensor channels (1070 and 1071).

7.4.1.2 ViewMondo API



ViewMondo API is a REST/JSon based "push" protocol to push data and camera images to a ViewMondo server. Since this is a "push" protocol, no access to the CloudGate device from the internet is needed – i.e. the SIM card/contract does not have to support a public accessible IP address for the device.

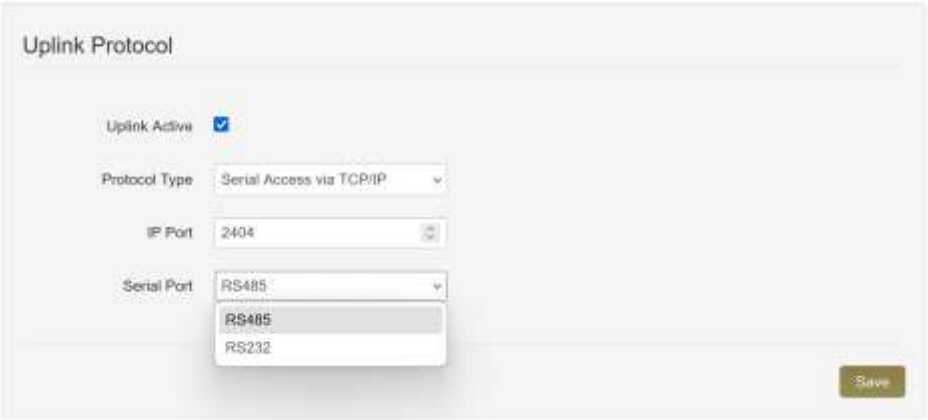
- API Device id: the (unique) API Device ID. The default value is created by including the CloudGate serial number, so it is unique by default. However, it can be edited on the device (and is independent from the actual CloudGate serial number), e.g. if the cloudgate hardware for a existing installation needs to be replaced (the API device id from the original device can be configured so no new station is created in ViewMondo for the device)
- Device Info: general information about the device
- API Version: the API version supported by this device

- Ping Interval: the interval the device connects to the server (independent of any data or cam picture being available for transfer)
- Server URL: the URL for the ViewMondo server
- Set Clock: synchronize clock with server time (on "ping"). Note: if the clock of the device is not yet set at all when connecting to the server, the clock is always set to the server time. This parameter only influences the behavior after the clock was set initially.
- Inactivity Reset Interval: inactivity interval after which UpCom plugin will reset if there is no activity on the API connection. 0= do not reset.
- Aggregate Values: aggregate the measure values for transmission via API – i.e. if the device poll interval is lower than the aggregate interval, the measure values are aggregated (min/max/average/sum/modal…) according to the settings for the respective sensor channel (see below).

7.4.1.3 Serial Access via TCP/IP

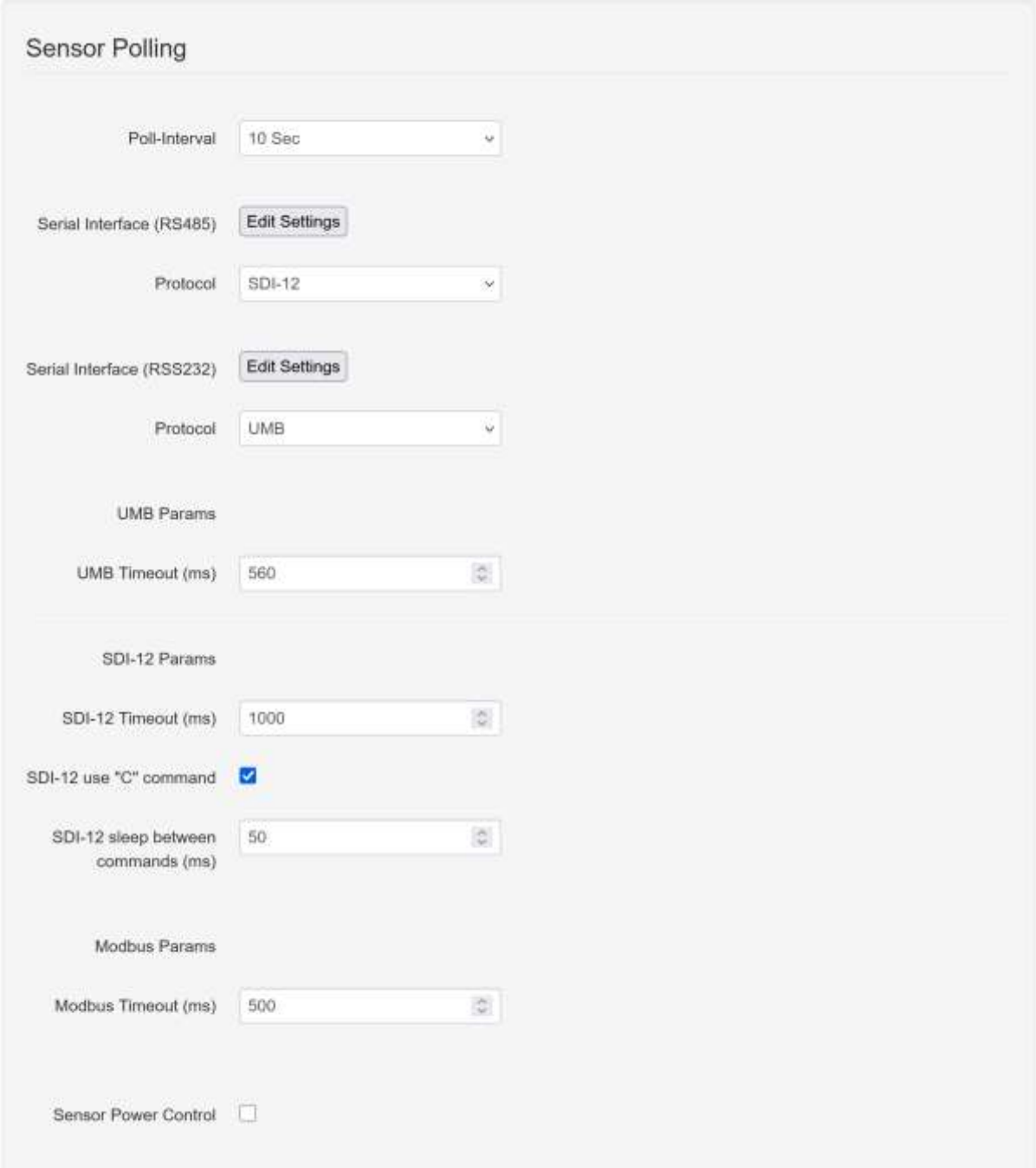
Direct Access via TCP/IP provides a "pass through" TCP/IP Port to the serial interface on the configured IP Port.

For the CloudGate with industrial serial card, the serial interface can be selected



Note: this can be used to access the UMB sensors with the Config Tool .net, or the SDI-12 sensor e.g. with PuTTY (configure connection type "other" and select "Raw") to enter SDI-12 commands directly.

7.5 Sensor Polling



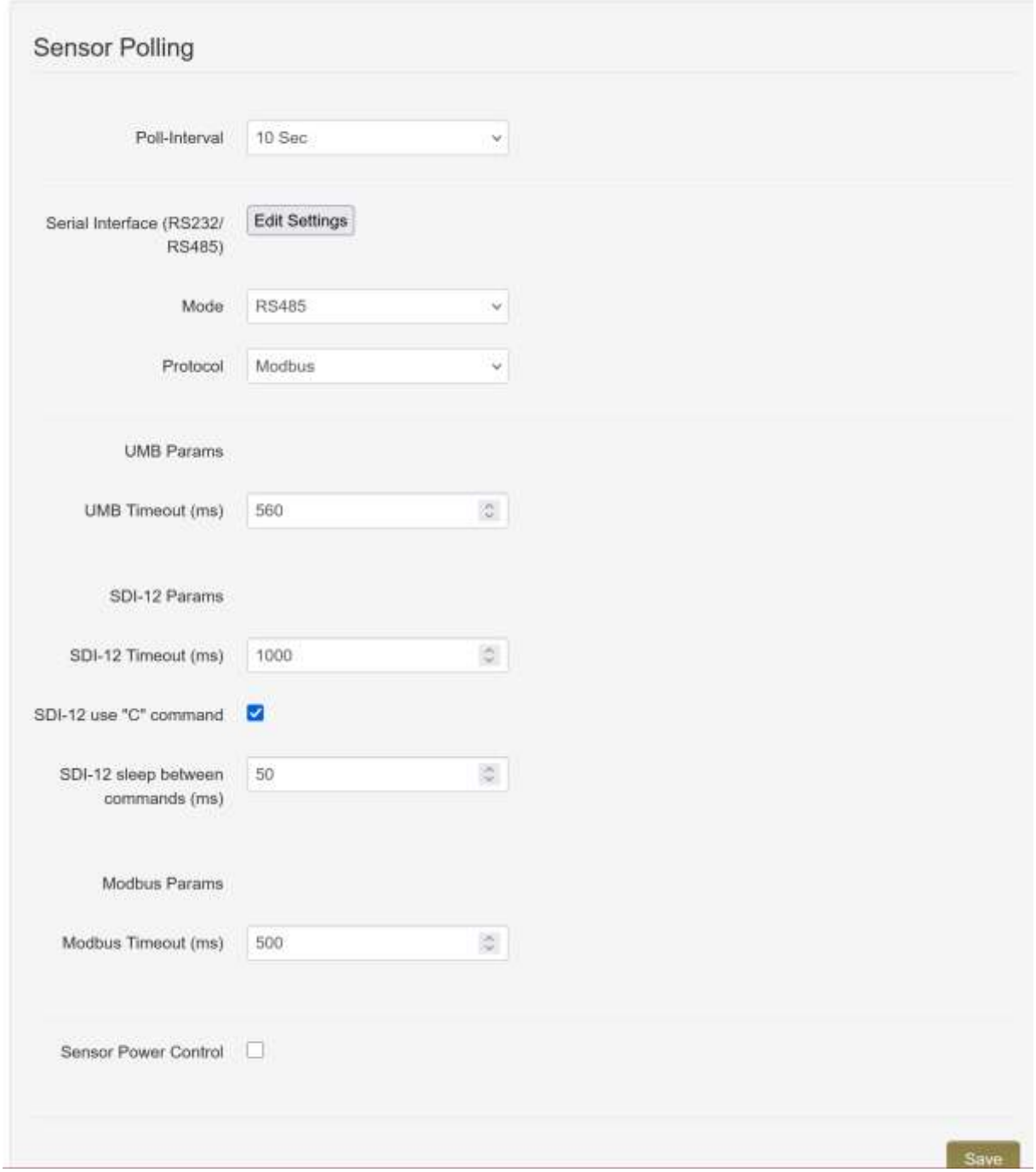
Here, the "Poll Interval" and the "Timeout" for communication with the sensor devices is configured.

The devices are polled in the configured interval, and (if enabled) the data is stored in the local database (see below).

NOTE: very small poll intervals (below 5 seconds) might not always get a result within the configured poll interval, e.g. if a device does not respond and timeouts are exceeded, also the SDI-12 protocol is rather

slow – poll intervals below 2 seconds are not possible to archive if SDI-12 sensors are used, even without any timeouts or retries on the device communication.

Depending on the CloudGate hardware options, settings for one or 2 serial interfaces can be selected.



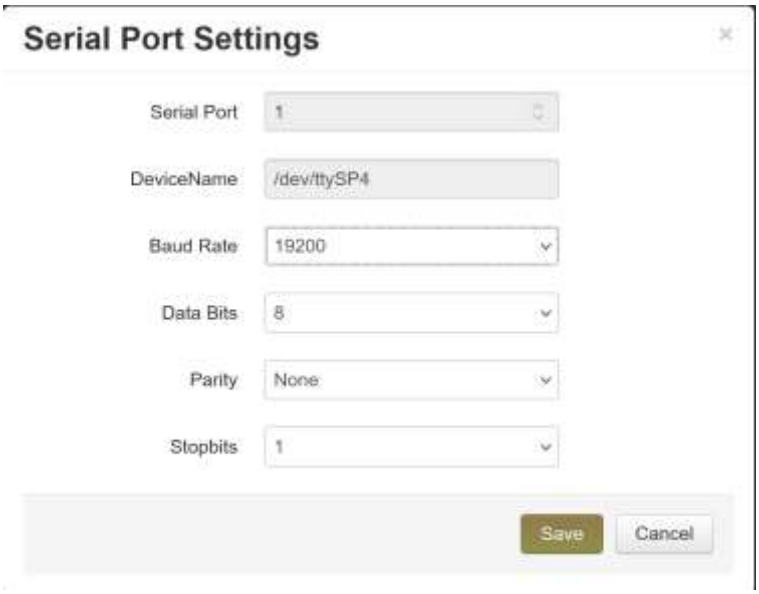
7.5.1 Mode: RS232 or RS485

On a Cloudgate with "industrial serial card", 2 serial interfaces – one fixed "RS232" and one fixed "RS485" – are available. The RS485 interface is on the left hand side on the green 5 pin connector. The RS232 interface is on the right hand side on the standard 9 pin connector

On a Cloudgate "Mini" or "Nano", only 1 serial interface is available – and the mode can be changed from "RS232" to "RS485". The serial interface is always the green 5 pin connector, and the parameter controls whether this is treated as an RS232 interface or as an RS485 interface.

NOTE: on the CloudGate Mini or Nano, changing the settings for the serial interface will cause the device to reboot.

7.5.2 Serial Port Settings



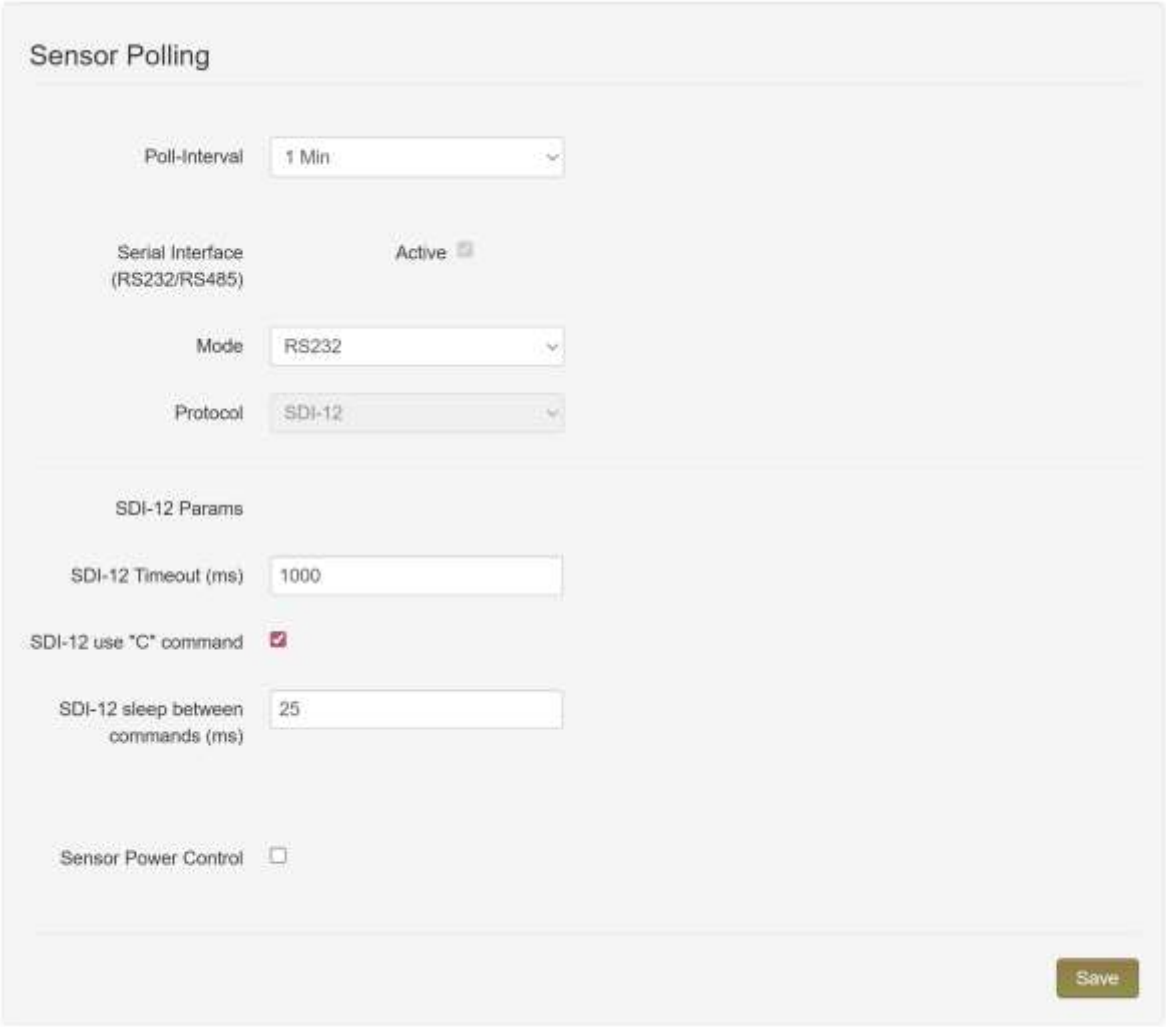
The parameters for the serial port (RS232/RS485) can be configured here.

7.5.3 Protocol: SDI-12 / UMB / Modbus

Sensor devices can be either "Lufft UMB", "SDI-12" or "Modbus" devices.

Note: only one sensor protocol can be used on a serial interface – so devices with only one serial interface (Cloudgate Mini/Nano) can drive either "UMB" or "SDI-12" sensor devices.

Once at least one sensor channel is configured, the protocol type setting can not be changed anymore:



For the Cloudgate with industrial serial port, two types of sensors can be attached at the same time.

SDI-12: to use the SDI-12 protocol, an RS232/SDI-12 (or RS485/SDI12) adapter (e.g. "TekBox TBS06-DR") needs to be connected to the respective serial port.

SDI-12 Timeout (ms): the timeout (in milliseconds)

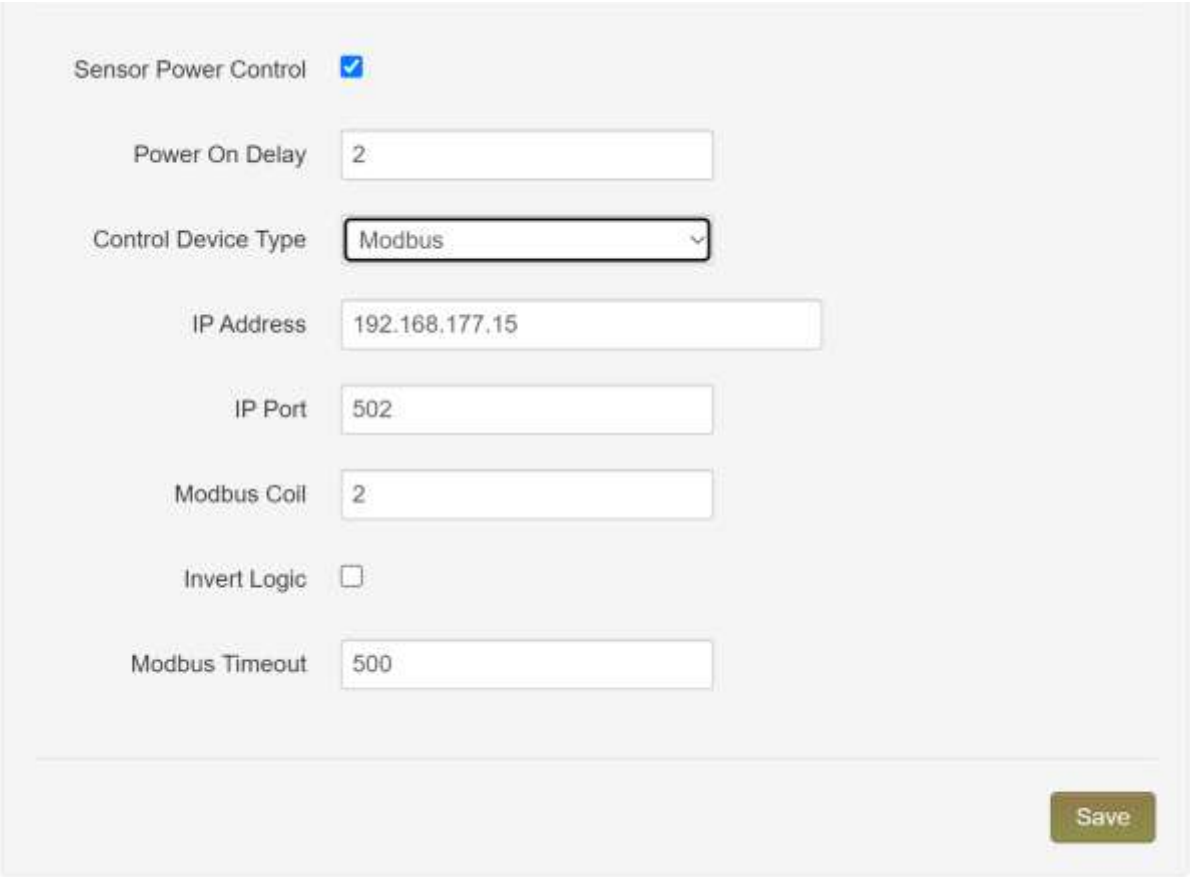
SDI-12 use "C" command: use the "C" command to start the measurement (if disabled, the "M" command is used

SDI-12 sleep between commands (ms): wait time between measurement and data commands

7.5.4 Sensor Power Control

If the hardware setup includes a digital I/O device that can be controlled via Modbus (e.g. Acromag, ControlByWeb) with relay and corresponding cabling, the

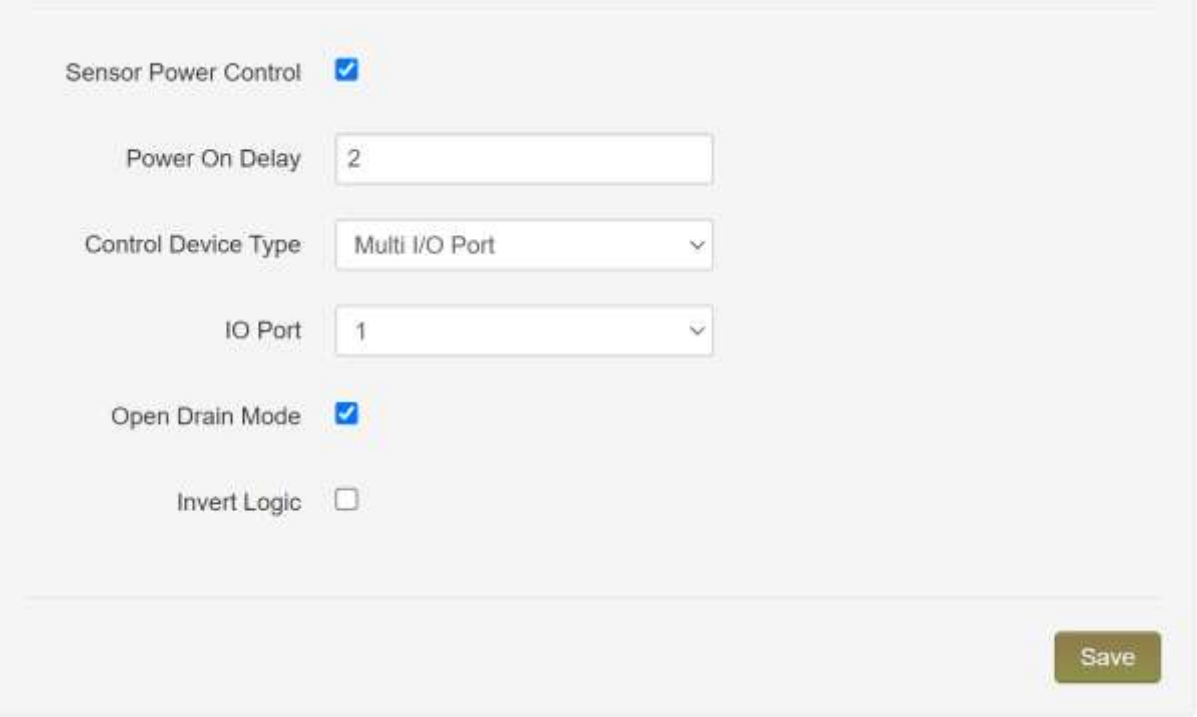
operating power for the UMB sensor can be controlled by the UpCom plugin for power saving (e.g. solar powered stations).



The parameters are:

- Power On Delay: the time (in seconds) the sensors need to power up. The power will be switched on the configured quantity of seconds before the sensors are due to be polled.
- Control Device Type: Modbus or "Multi IO Port" (Cloudgate Mini only, see below)
- IP Address: the IP address or DNS host name for the Modbus IO device
- IP Port: the IP port for the Modbus IO device
- Modbus Coil: the Modbus coil address where the power relay for the UMB sensors is connected.
- Invert Logic: invert the output logic of the Modbus output port.
- Modbus timeout: timeout for communication with the Modbus device.

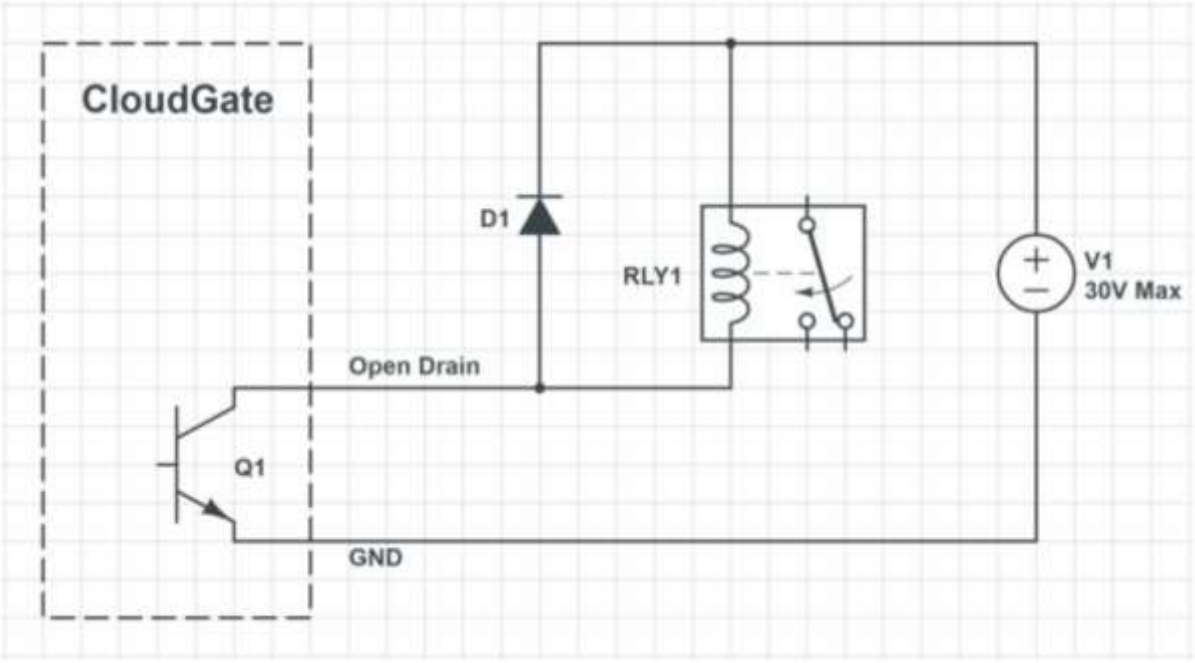
On "Cloudgate Mini" with Multi Purpose IO Card, one of the 3 "GPIO" ports on the IO card can be used to control the power for sensors:



Note: The output mode for the GPIO port can be either "push/pull" signal, or "open drain".

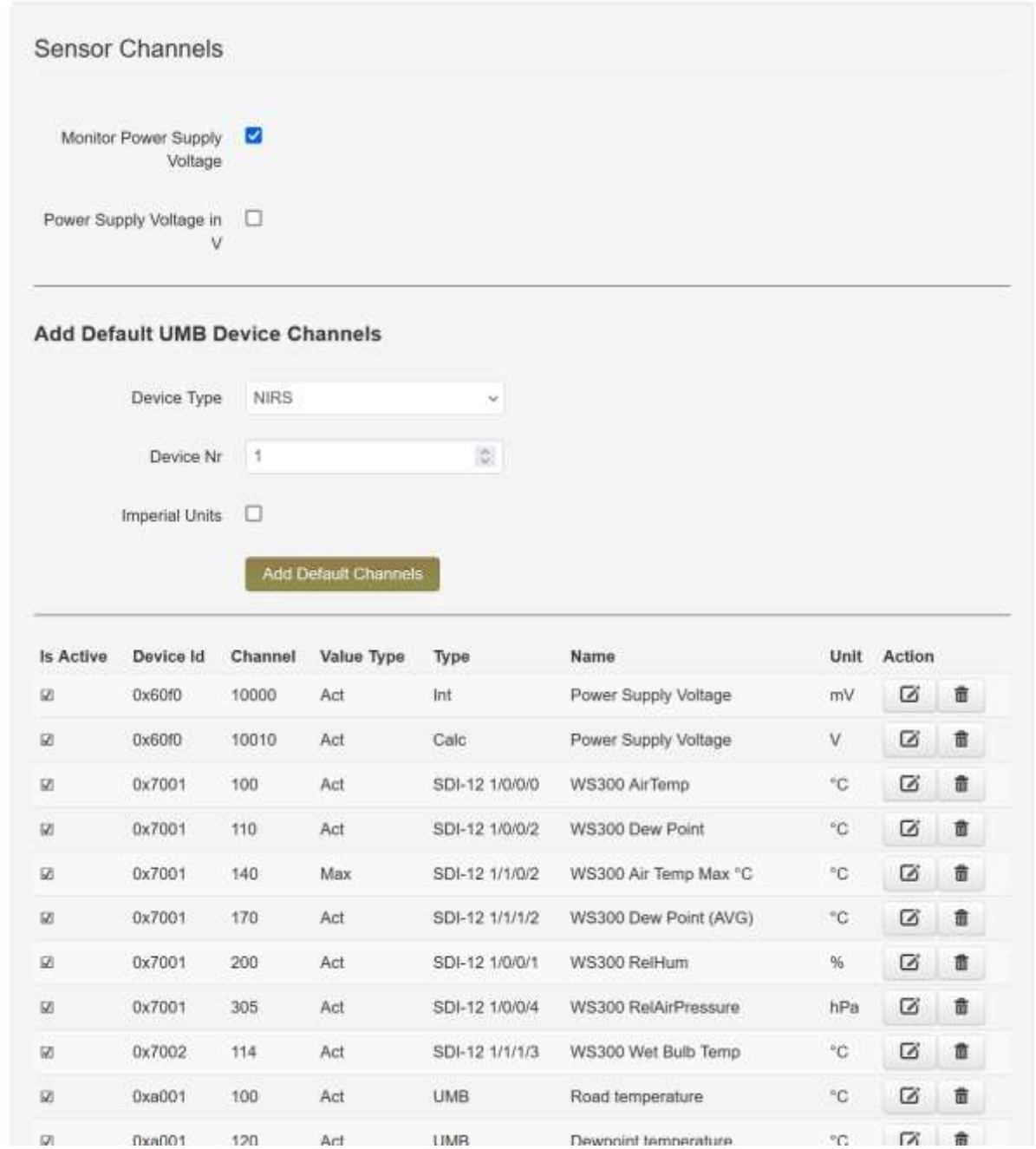
In "push/pull" mode, the signal level for the GPIO Ports is 3.2 Volts – and maximimum 20mA.

When connecting a relay, the best choice is to set your pin as an 'open drain' output. This means that the pin will sink the current from an external power supply when the pin is 'high' (current will flow) and that no current will flow when the pin is 'low'.



Please note, that a flyback diode must be included in the circuit, and the maximum current must be limited to 250mA to prevent damage to the CloudGate hardware!

7.6 Sensor Channels



Is Active	Device Id	Channel	Value Type	Type	Name	Unit	Action
☑	0x60f0	10000	Act	Int	Power Supply Voltage	mV	
☑	0x60f0	10010	Act	Calc	Power Supply Voltage	V	
☑	0x7001	100	Act	SDI-12 1/0/0/0	WS300 AirTemp	°C	
☑	0x7001	110	Act	SDI-12 1/0/0/2	WS300 Dew Point	°C	
☑	0x7001	140	Max	SDI-12 1/1/0/2	WS300 Air Temp Max °C	°C	
☑	0x7001	170	Act	SDI-12 1/1/1/2	WS300 Dew Point (AVG)	°C	
☑	0x7001	200	Act	SDI-12 1/0/0/1	WS300 RelHum	%	
☑	0x7001	305	Act	SDI-12 1/0/0/4	WS300 RelAirPressure	hPa	
☑	0x7002	114	Act	SDI-12 1/1/1/3	WS300 Wet Bulb Temp	°C	
☑	0xa001	100	Act	UMB	Road temperature	°C	
☑	0xa001	120	Act	UMB	Dewpoint temperature	°C	

Here, the UMB or SDI-12 sensor channels to be polled are configured.

Channels can be added by clicking on “Add”.

NOTE: the UpCom plugin needs to be restarted after making changes to the sensor configuration in order for the configuration change to take effect!

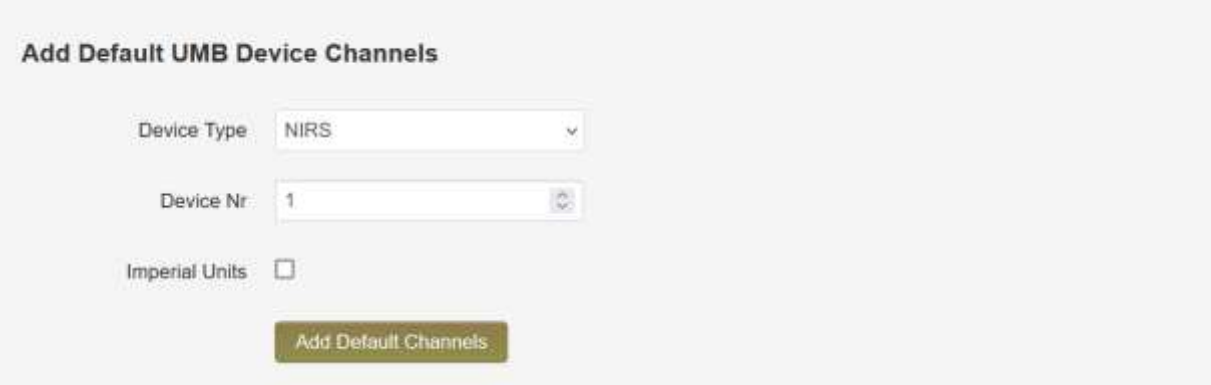
7.6.1 Monitor power supply voltage

If "**Monitor Power Supply Voltage**" is activated, the power supply voltage as reported by the CloudGate device is added as a "virtual sensor channel" (UMB Device 0x6f0 channel 10000)

NOTE: this "virtual" sensor channel is added automatically by the software as type "Int" (internal). Please do not manually configure an UMB sensor channel with this device id/channel nr.

If "Power supply voltage in V" is selected, the power supply voltage will be converted to V (instead of mV).

7.6.2 Add default channels for UMB devices



This can be used to easily add the standard channels for the mostly used UMB devices like NIRS, WSx weather stations and IRS31Pro road sensors

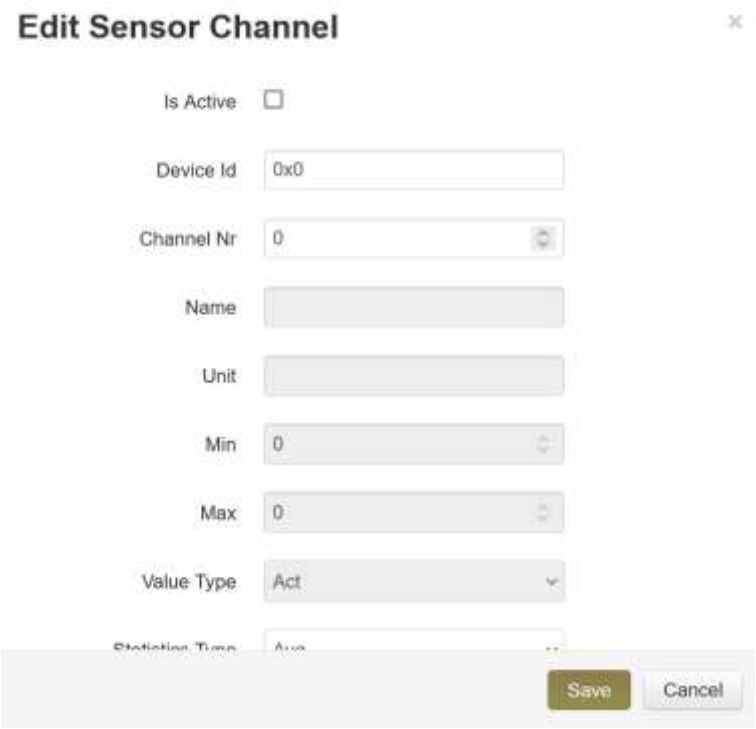
Device Type: select the device type you would like to add

Device Nr: select the device number within the UMB device type group you would like to add

Imperial units: check this box if you would like to add the channels for imperial units (°F/mil/inches/mph..) instead of the standard SI units (°C,µm,mm, m/s…)

Click "Add Default channels" to add the channels for the selected device type.

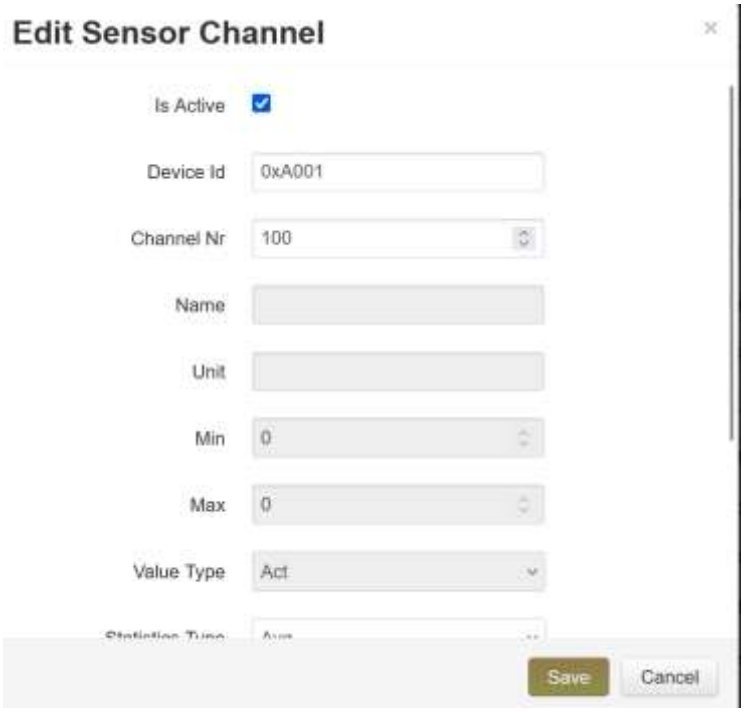
7.6.3 Add a sensor channel



"Edit Sensor Channel" is shown.

For all (UMB /SDI-12…) sensor channels, the UMB device id and sensor channel nr have to be configured. Set "is active" in order for the channel to be polled from the device.

Note: the device id can be configured as a "hexadecimal" string (as well as decimal). E.g.



Device ID 0xA001 – a device in device class 0xA (see UMB device manual) and sensor channel 100.

7.6.4 UMB Sensor channels:

Meta data for the UMB sensor channel – like name and unit – are (for newly configured sensor channels) retrieved from the device and will show as soon as the information could be read out from the respective device.

The "Statistics Type" is used when values are aggregated for API Uplink, Database storage or CSV Export

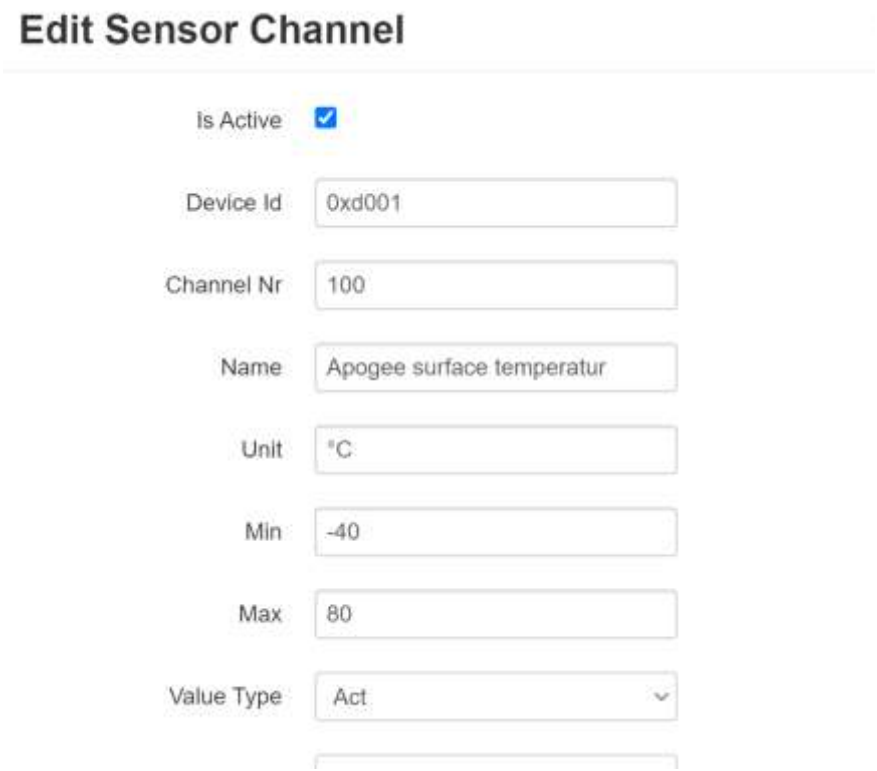
Value Mapping can also be configured for each sensor channel.

7.6.5 SDI-12 Sensor channels:

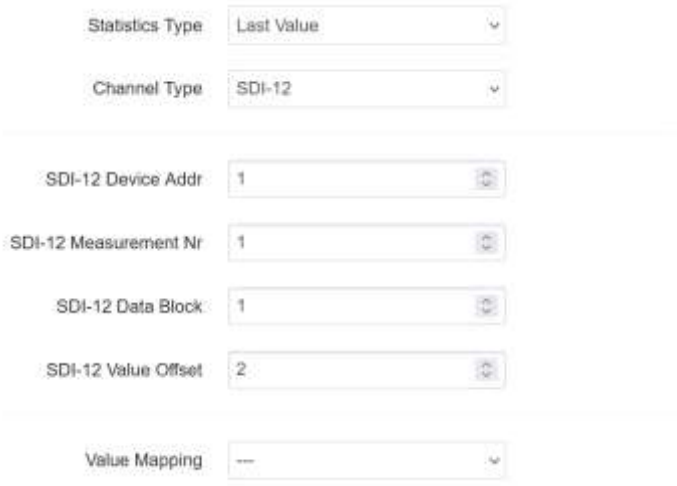
SDI-12 sensor channels are configured with UMB device ID and channel nr – just like "native" UMB channels, to map the values to an UMB channel. So after polling the sensor channels for the respective device, there is no difference between how UMB or SDI-12 sensors are handled by the software.

Even though the device id and channel number can be freely configured, choosing a meaningful UMB device id and channel nr is recommended, especially if the ViewMondo API uplink protocol is used (sensor channel type assignments in ViewMondo are automatically processed if the UMB device class and channel number are "well known")

In addition to the UMB device address and channel number, the sensor meta data like name, unit, minimum and maximum value have to be manually configured, since this information can not be obtained via SDI-12.



Please consult your SDI-12 device documentation about SDI-12 protocol details like the "measurement block/number", "data block" and "value offset" parameters.



The actual SDI-12 specific configuration items are:

- SDI-12 Device Addr: the SDI-12 device address (0…9)
- SDI-12 Measurement Nr: the SDI-12 measurement number. 0 = "no additional measurement" – i.e. the "start measurement" command would be "aM!" (basic measurement).
- SDI-12 Data Block: the data block within the measurement number (0..9)
- SDI-12 Value Offset: the value "offset" within the value result string from the device, starting at offset 0 for the first measure value in the string.

7.6.5.1 Example SDI-12 configuration – Apogee road temperature:

The Apogee road temperature sensor supports a "basic" measurement ("aM!") with 1 value (the road temperature) or an extended measurement ("aM1!") with 2 values (road temperature and body temperature) – or measurement 2 with millivolts and body temperature. There is only one "data block" with number 0 supported.

Command	Response	Response to 0D0!
aM! or aM0!	a0011<cr><lf>	Target temperature
aM1!	a0012<cr><lf>	Target temperature and sensor body temperature
aM2!	a0012<cr><lf>	Target millivolts and sensor body temperature

The "basic" measurement command is "aM!" ("a" being the SDI-12 device id, defaults to 0).

So – if only the road temperature should be polled, the SDI sensor configuration in UpCom would be

If both – surface temperature and body temperature are to be polled, the sensor channels are configured for measurement nr 1 and value offset 0 (road temperature) and 1 (body temperature).

7.6.5.2 Example SDI-12 configuration - OttHydromet WS300

The OttHydromet WSx family supports multiple measurement commands, depending on the device subtype

aM!	Measurement basic minimal data set	WS 600/700/800, 500, 400, 300, 200, 100, 501-510, 301-310, 601, 401, PSM 2 all
aM1!	Measurement temperatures	all
aM2!	Measurement humidity	all
aM3!	Measurement air pressure	all
aM4!	Measurement wind	all
aM5!	Measurement compass	all
aM6!	Measurement precipitation	all
aM7!	Measurement global radiation	all
aM8!	Measurement external temperature	all

The “data” buffer assignment (i.e. which values are reported in the respective command) depend on the subtype. For the WS-300, the manual lists following assignment:

19.6.3.4 Buffer Assignment Basic Data Set WS300-UMB

Device configured for measurement in metric units:

Measurement Value	UMB Channel	Min	Max	Unit
Buffer '0'				
Air Temperature (act)	100	-50.0	60.0	°C
Rel. Humidity (act)	200	0.0	100.0	%
Dew Point (act)	110	-50.0	60.0	°C
Abs. Air Pressure(act)	300	300.0	1200.0	hPa
Rel. Air Pressure (act)	305	300.0	1200.0	hPa
Buffer '1'				
Air Temperature (min)	120	-50.0	60.0	°C
Air Temperature (max)	140	-50.0	60.0	°C
Air Temperature (avg)	160	-50.0	60.0	°C
Rel. Humidity (avg)	260	0.0	100.0	%
Buffer '2'				
Rel. Humidity (min)	220	0.0	100,0	%
Rel. Humidity (max)	240	0.0	100,0	%
Rel. Air Pressure (min)	325	300.0	1200.0	hPa
Rel. Air Pressure (max)	345	300.0	1200.0	hPa
Rel. Air Pressure (avg)	365	300.0	1200.0	hPa
Buffer '3'				
Abs. Humidity (min)	225	0.0	1000.0	g/m³
Abs. Humidity (max)	245	0.0	1000.0	g/m³
Abs. Humidity (avg)	265	0.0	1000.0	g/m³
Buffer '4'				
Wet Bulb Temperature (act)	114	-50.0	60.0	°C
Specific Enthalpy (act)	215	-100.0	1000.0	kJ/kg

Example: Request buffer '0'

0D0!

`0+13.5+85.7+11.2+1017.0+1001.0`

Air temperature 13.5°C, rel. humidity 85.7%, dew point 11.2°C, rel. air pressure 1017.0hPa, abs. pressure 1001.0hPa

Assuming the SDI-12 device id is set to 2, the SDI-Parameters for Air Temp (Act) would be:

and for rel. Air Pressure (act) it would be

The "Wet Bulb Temperature (act)" could be polled using the "additional measurement command 1"

19.6.4.1 Buffer Assignment Additional Measurement Commands M1 / C1: Temperature

Device configured for measurement values in metric units:

Measurement Value	UMB Channel	Min	Max	Unit
Buffer '0'				
Air Temperature (act)	100	-50.0	60.0	°C
Air Temperature (min)	120	-50.0	60.0	°C
Air Temperature (max)	140	-50.0	60.0	°C
Air Temperature (avg)	160	-50.0	60.0	°C
Dew Point (act)	110	-50.0	60.0	°C
Buffer '1'				
Dew Point (min)	130	-50.0	60.0	°C
Dew Point (max)	150	-50.0	60.0	°C
Dew Point (avg)	170	-50.0	60.0	°C
Wet Bulb Temperature (act)	114	-50.0	60.0	°C

Example: Request with M command

0M1!

00009<CR><LF>

0D0!

0+12.5+10.7+13.5+11.8+5.3<CR><LF>

0D1!

0+4.2+5.9+5.6+9.8<CR><LF>

As

7.6.6 Modbus Sensor Channels

Modbus sensor channels (like SDI-12 sensor channels, see above) are configured with UMB device ID and channel nr – just like "native" UMB channels, to map the

values to an UMB channel. So after polling the sensor channels for the respective device, there is no difference between how UMB, SDI-12 or Modbus sensors are handled by the software.

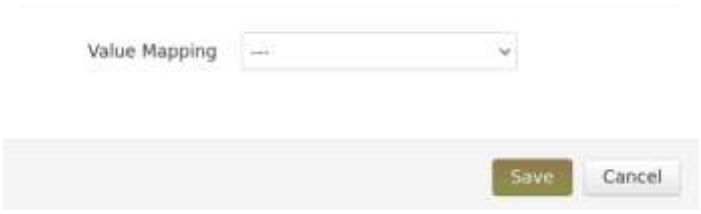
Even though the device id and channel number can be freely configured, choosing a meaningful UMB device id and channel nr is recommended, especially if the ViewMondo API uplink protocol is used (sensor channel type assignments in ViewMondo are automatically processed if the UMB device class and channel number are "well known")

In addition to the UMB device address and channel number, the sensor meta data like name, unit, minimum and maximum value have to be manually configured, since this information cannot be obtained via Modbus protocol.

Following parameters are configured for a Modbus sensor channel:

- Modbus Device ID: the device identifier
- PDU Address: the (start) address for the respective sensor value
- PDU Type: the PDU type (Holding Register (0x03), Register (0x04) or Coil (0x01))
- Data Type: the Data Type for the value (Unsigned Int16, Signed Int16, Unsigned Int32, Signed Int32, Float32, Byte)
- Byte Order: the byte order for Float32 values

7.6.7 Value mapping



Value mapping can be assigned to a sensor channel. Note: the value mapping will be applied when the data is read from the device and will override the "raw" value as reported by the sensor, i.e. all subsequent processing (e.g. storing the data in the database, transmitting the data via a protocol etc.) will use the mapped value as input.

NOTE: Be aware that for some protocols – like OPC-UA or NTCIP– a specific value mapping might be (pre-)configured for transmission of the data. The value mapping configured on the UMB sensor channel is independent of any further value mapping configured for a protocol sensor channel, so multiple value mappings might be processed!

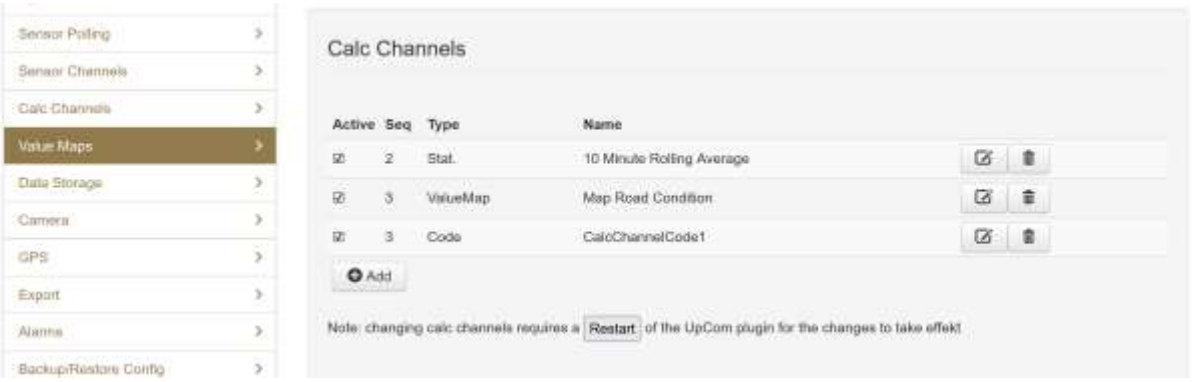
7.7 Calc Channels

Calc channels can be created based on value mapping (see below) or statistics aggregation – or by providing a custom Python script

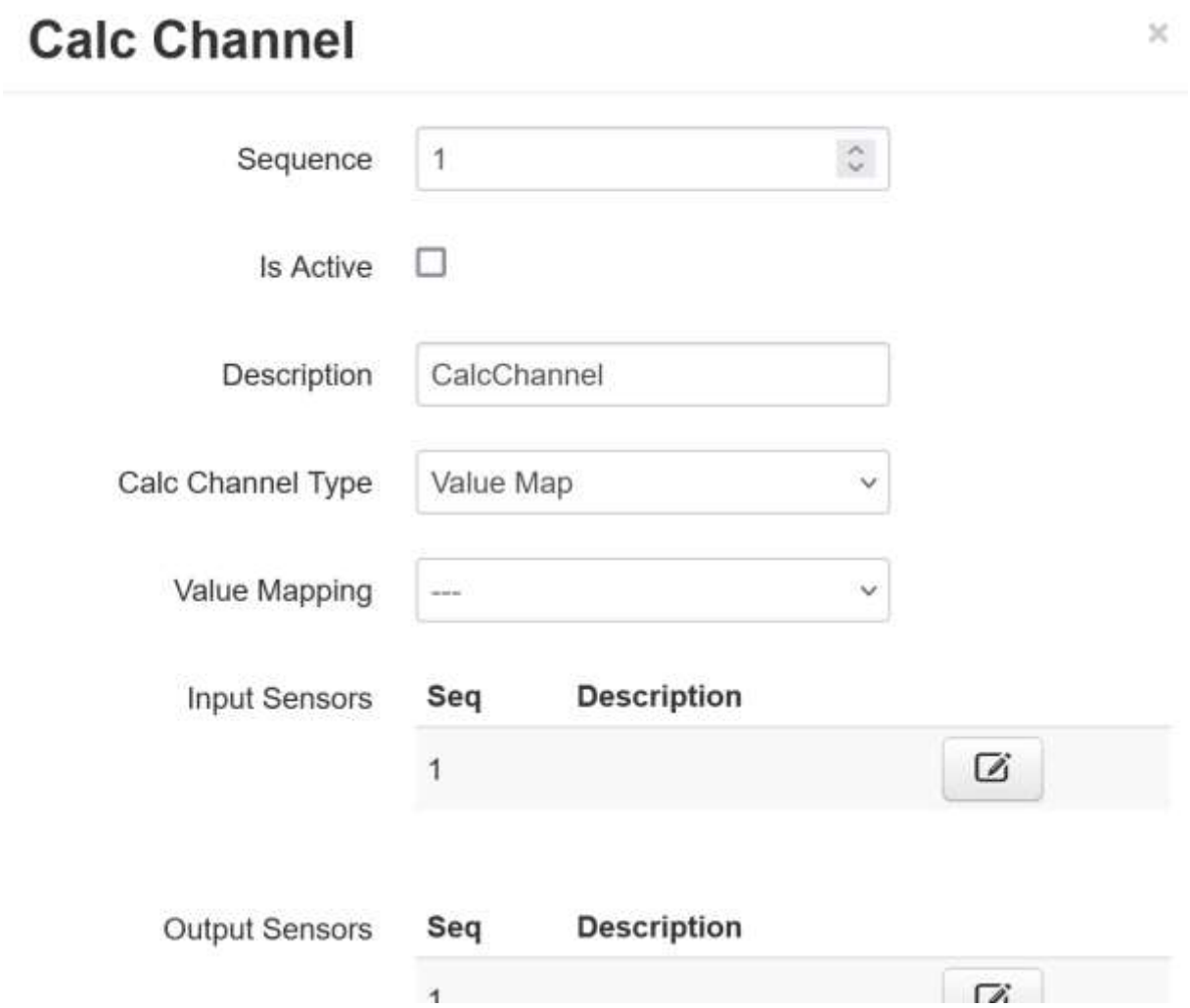
Complex calc channels might be available in later releases.

The simple calc channels always operate on one input and one output sensor channel. The input sensor channel can be any sensor channel (calc channel or UMB/SDI-12 sensor channel). The output sensor channel has to be a sensor channel of type "calc"

To set up a calc channel calculation, "add" a calc channel to the configuration:



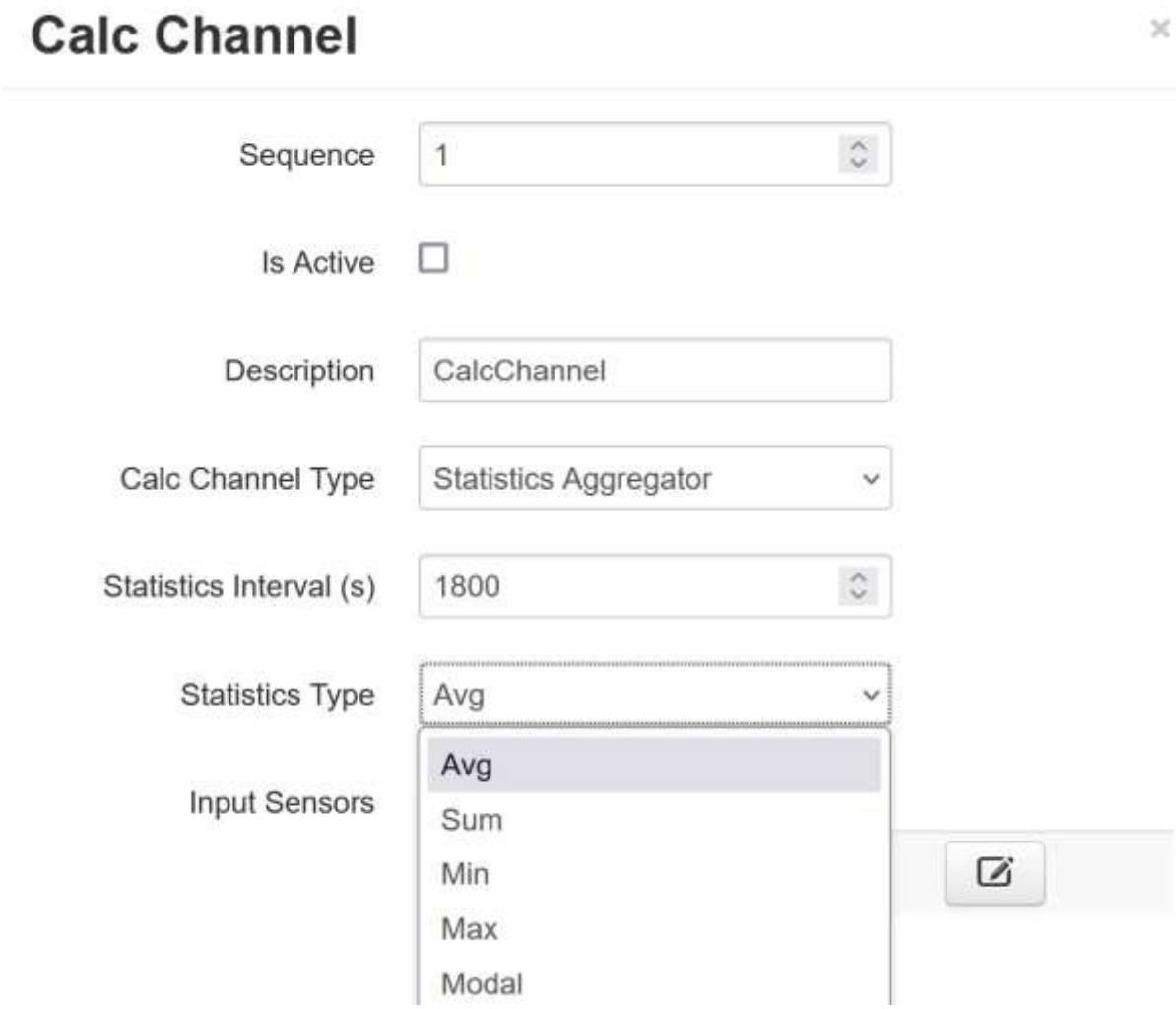
NOTE: the UpCom plugin needs to be restarted after making changes to the calc sensor configuration in order for the configuration change to take effect!



The "sequence" controls the order, in which multiple calc channels are calculated. This is important if you plan to use a calc channel result as input to another calc channel.

- Sequence: the sequence of the calc channel calculation
- Is Active: calc channel calculation is active
- Description: a description for this calculation setup
- Calc Channel Type: "Value Map" ,"Statistics Aggregator" or "Code"
- Value Mapping: only for type "Value Map" – select the value map (see below) from the dropdown list

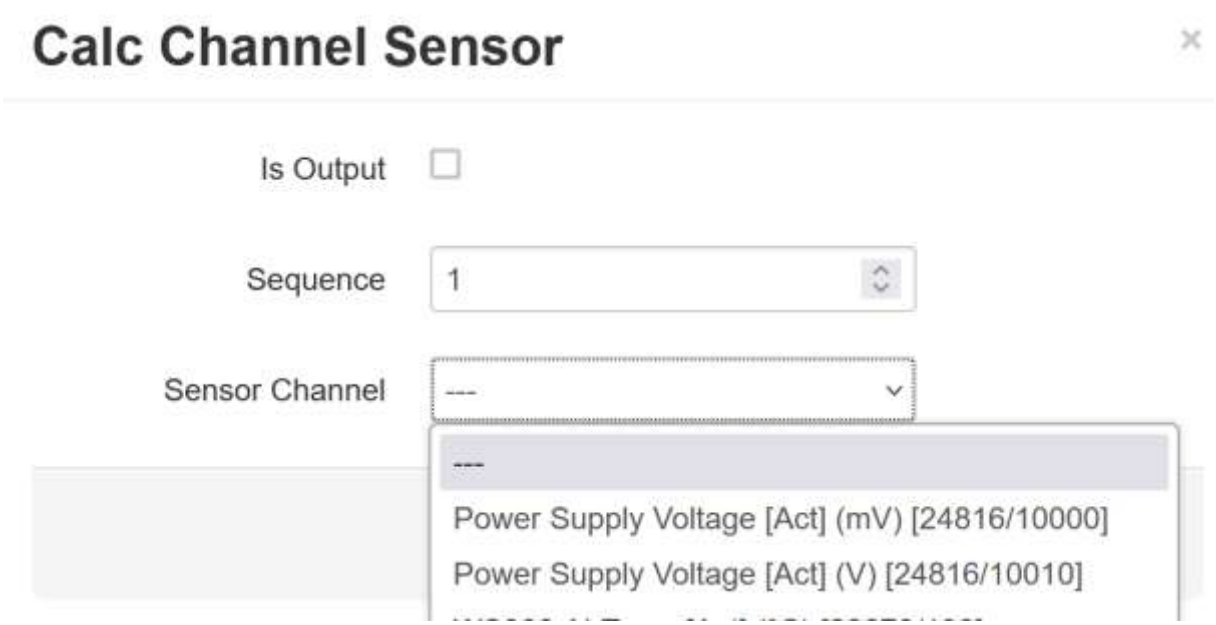
7.7.1 Type “Statistics Aggregator”



For “Statistics Aggregator”, the interval for the “rolling calculation” (in seconds) and the statistics type are configured.

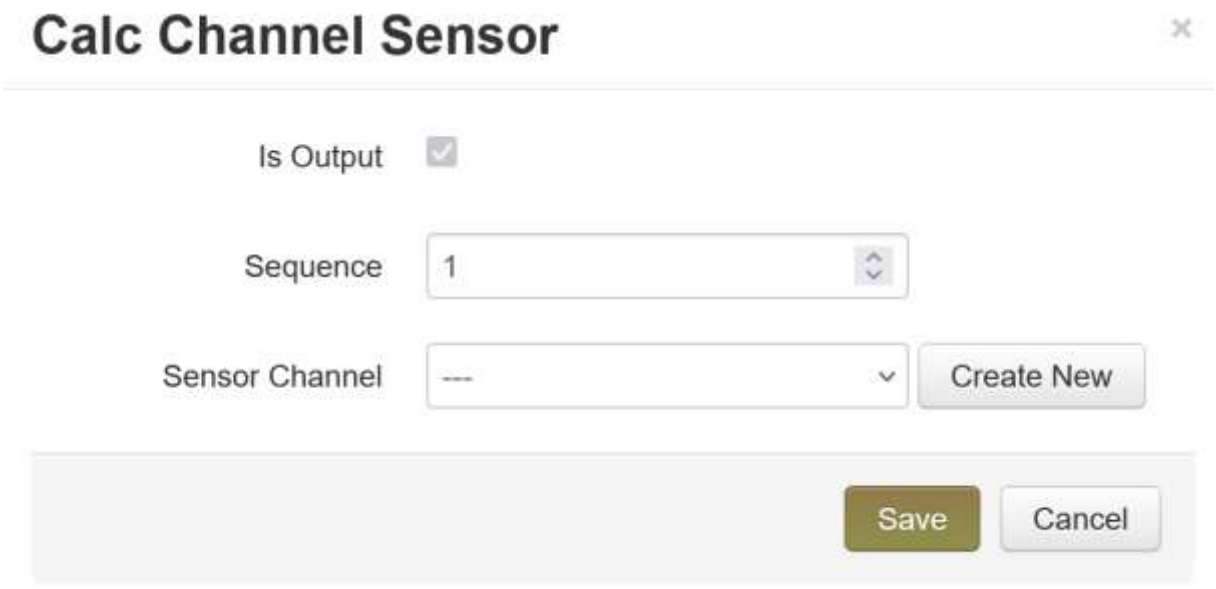
- Input Sensors: the input sensor channel(s) for this calculation. There is only one input sensor channel for the simple calc channels (Value Map/Statistics).
- Output Sensors: the output sensor channel(s) for this calculation. There is only one output sensor channel for the simple calc channels (Value Map/Statistics Aggregator).

For “input” sensors, any configured sensor channel can be selected:



Note: the "Is Input" checkbox is information only

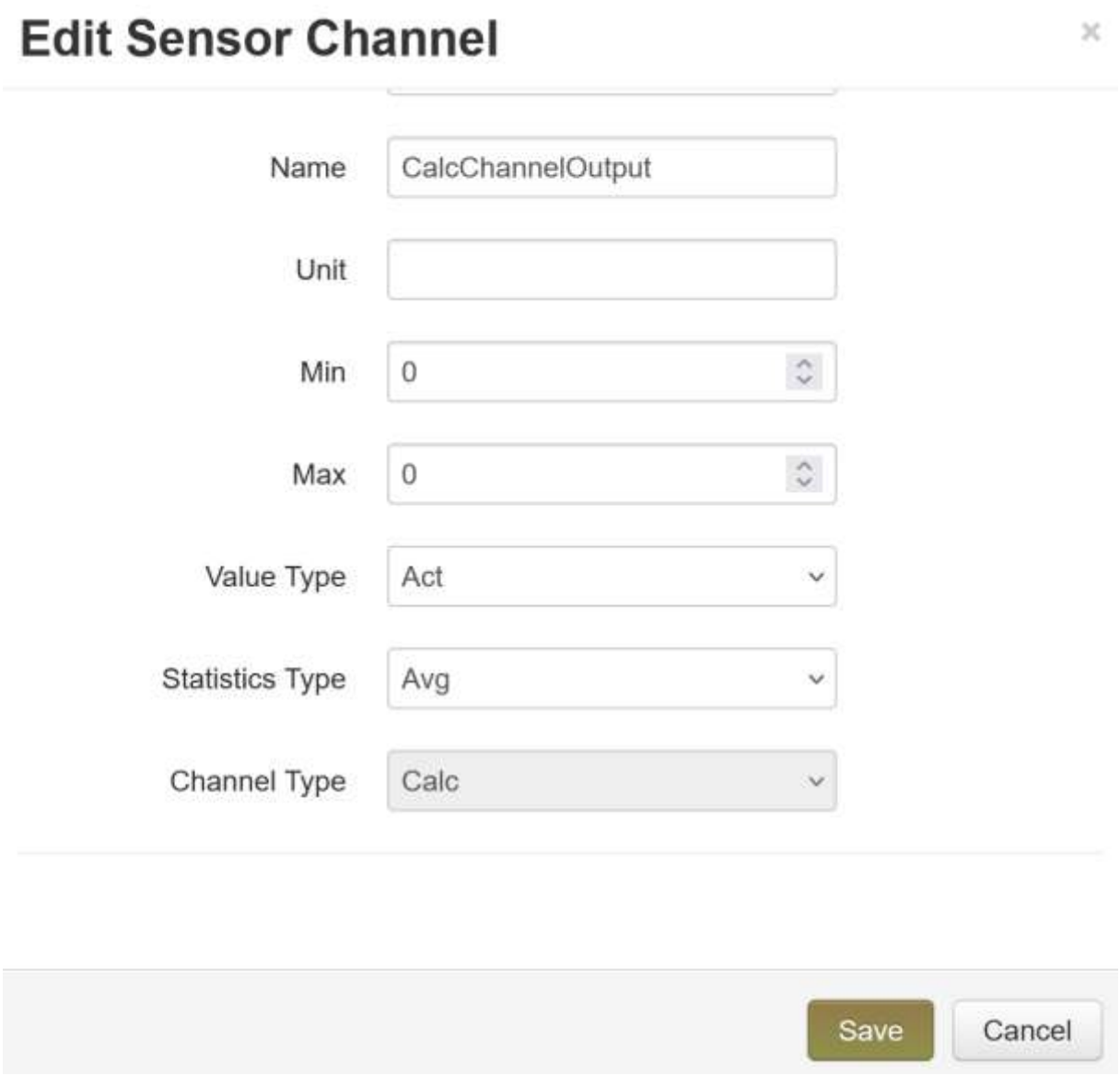
For "output" sensors, a "calc" channel needs to be selected:



7.7.1.1 Create Result Sensor

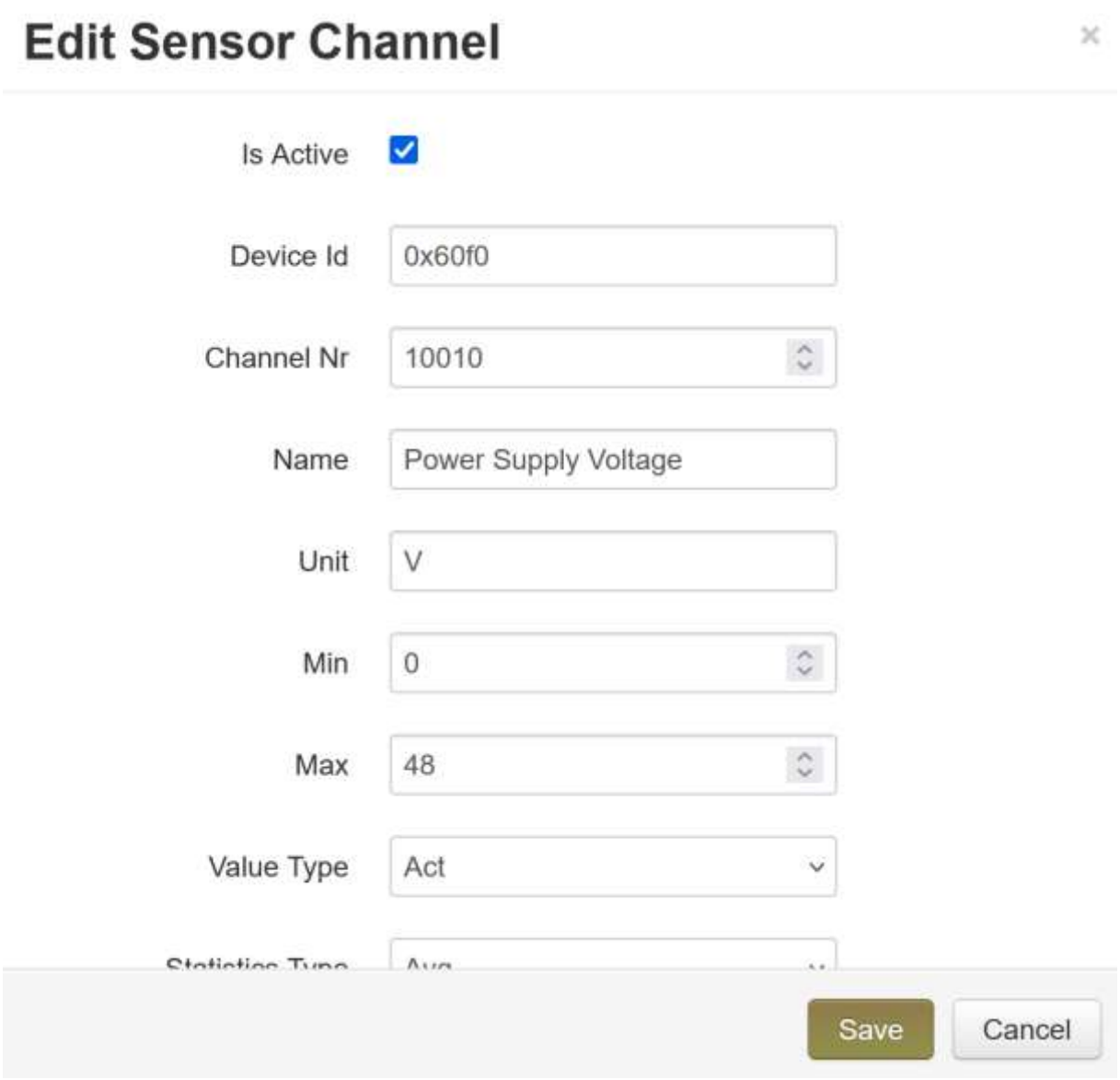
This usually means, that a new "output" sensor channel needs to be created for this calc channel calculation by clicking on "Create New"

This opens a "edit sensor channel" dialog for a calc channel:

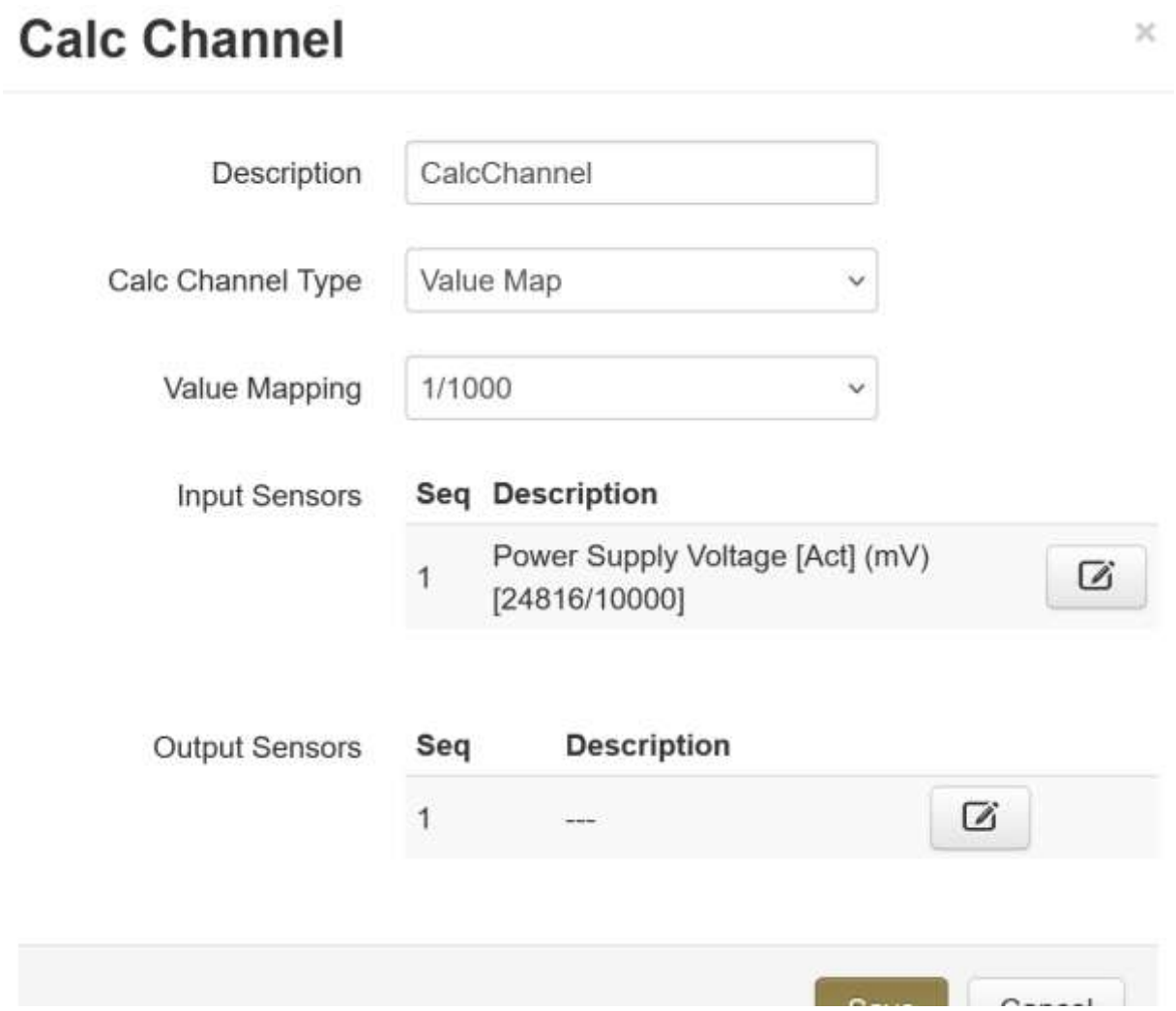


Select a unique Device Id and channel nr for this sensor channel. Note: since this is internally marked as channel type “Calc”, the device id could also be a real UMB device id without causing any conflict, e.g. if the result value matches a standard UMB sensor channel. If you use a “standard” UMB device id, please make sure to select a device id and sensor channel that will not conflict with a real sensor you might want to attach.

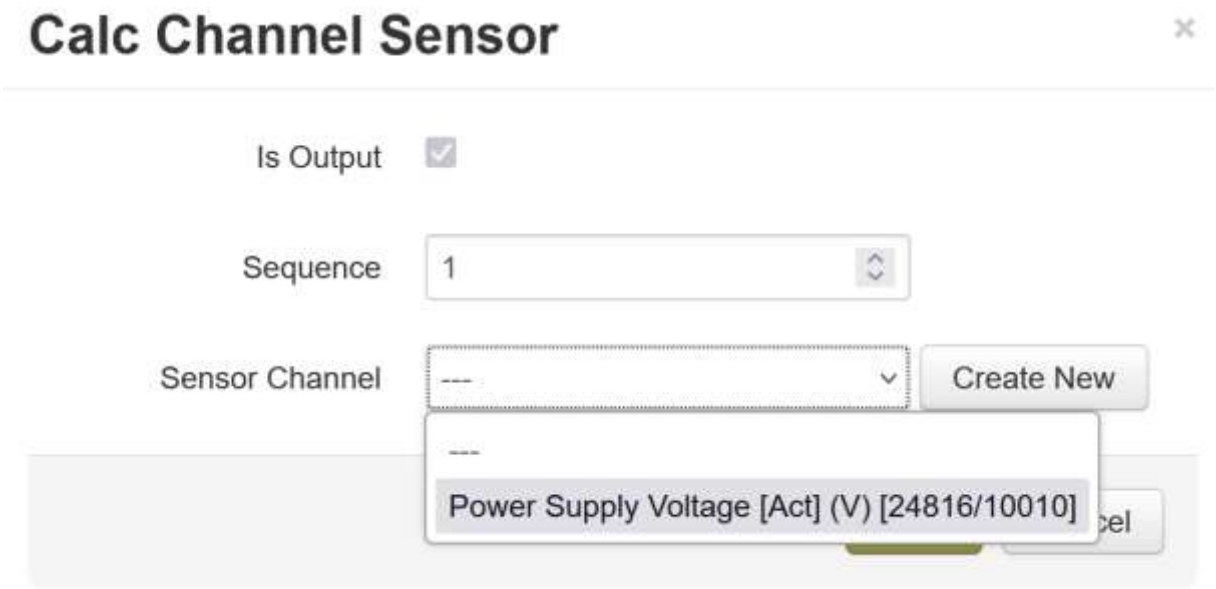
Also set a name, unit, min and max value and value type that matches the calculation result. E.g.:



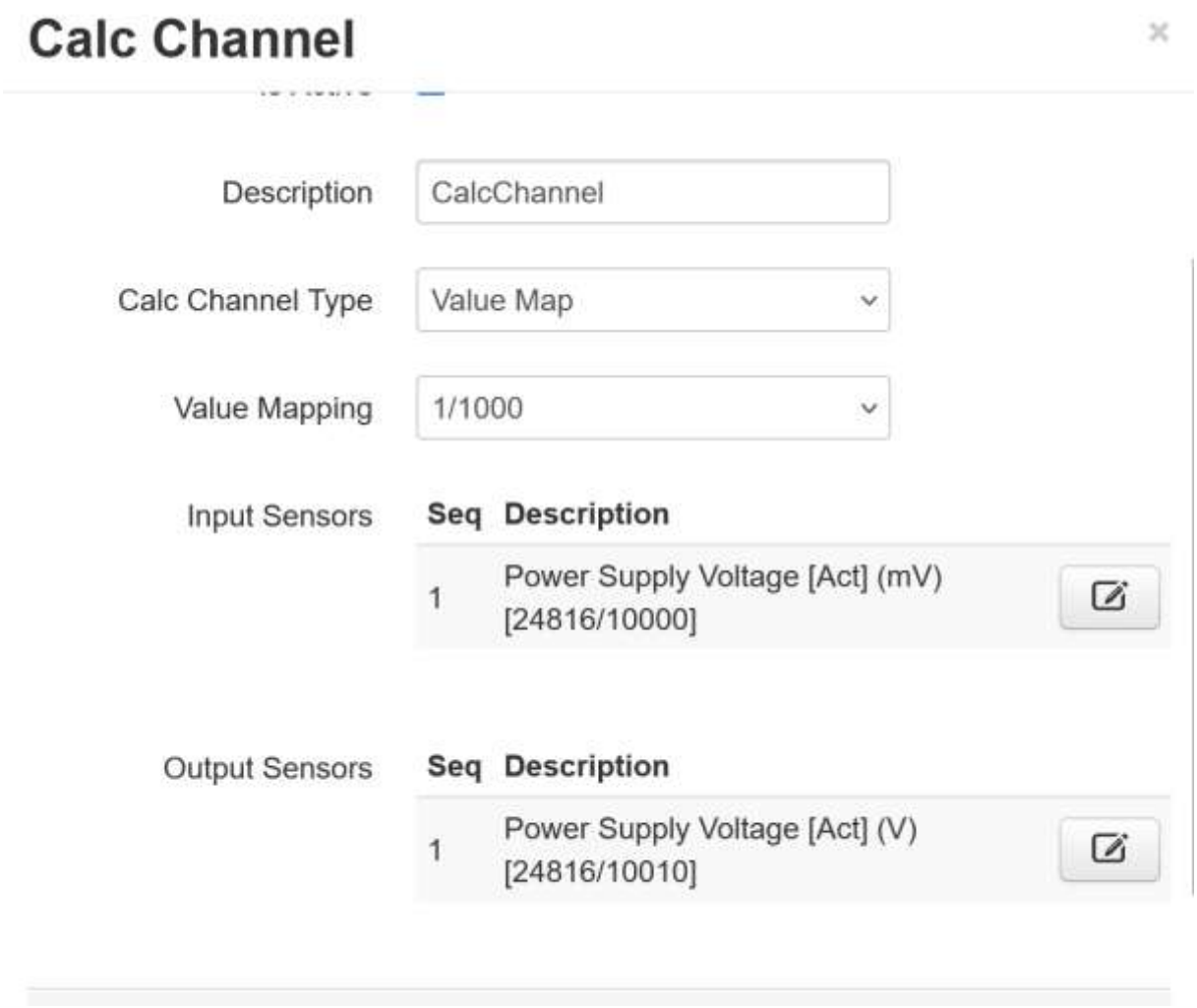
After saving the newly created sensor channel, you are taken back to the "edit calc channel" dialog



Now you can select the output sensor



And finalize the calc channel setup



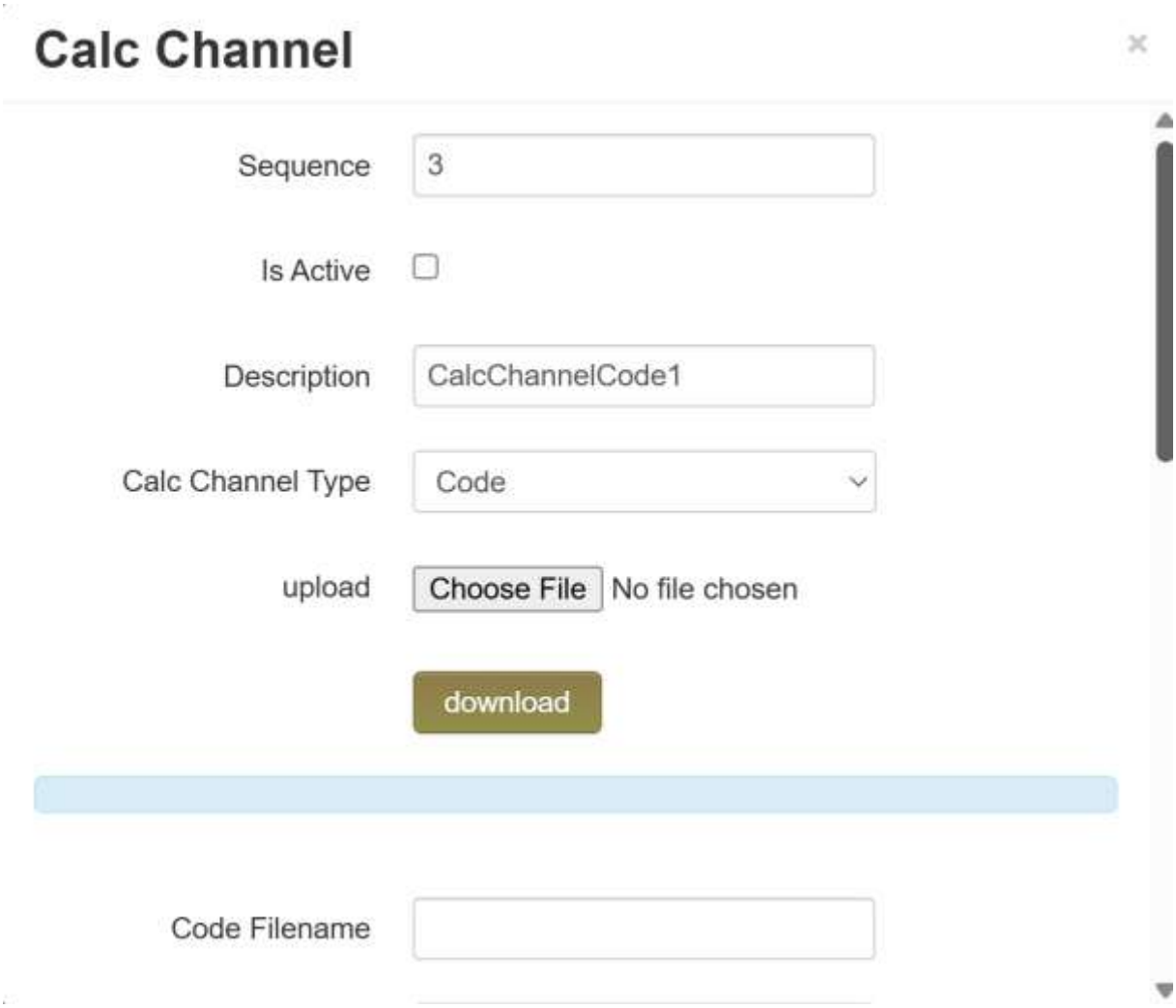
7.7.2 Type "Value Map"

The result is calculated using a "Value Map" – see below 7.8 Value Maps for details on how to configure a value map.

7.7.3 Type "Code"

The result is calculated using a user defined python script.

The python script version supported is python2.7.



Note: this status text also shows some error information on last execution of the script, when this dialog is opened.

7.7.3.1 Upload python script

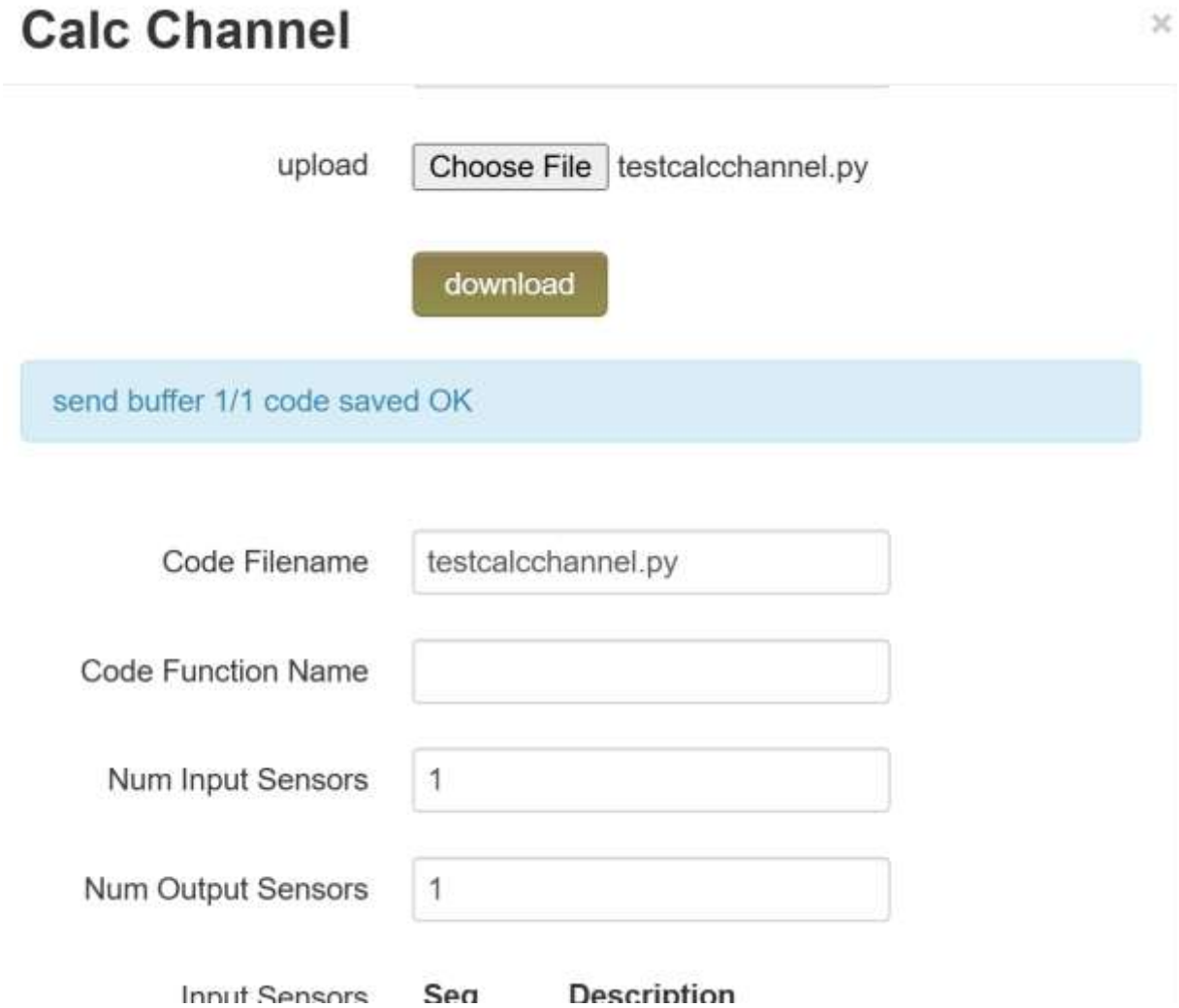
To upload the python script, click "Choose File" and select the script file from your hard drive.

Depending on the file size, the script file is split in multiple packages for uploading. Progress/Result of the transfer is shown on the blue status text.

7.7.3.2 Download python script

Clicking the "download" button downloads the code stored for this calc channel;

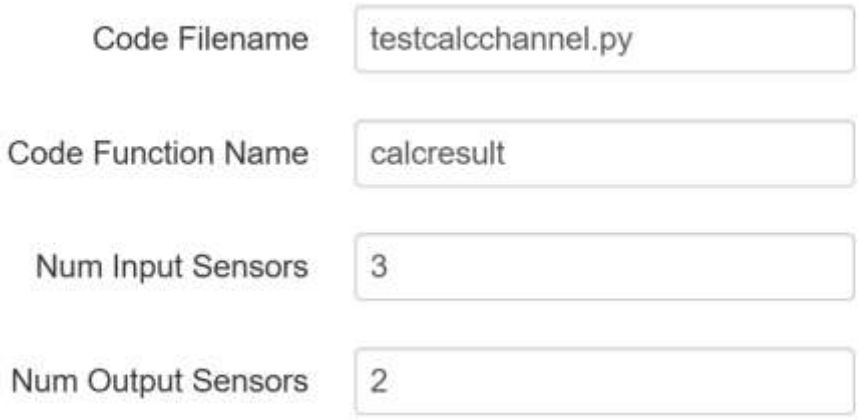
7.7.3.3 Code Parameter



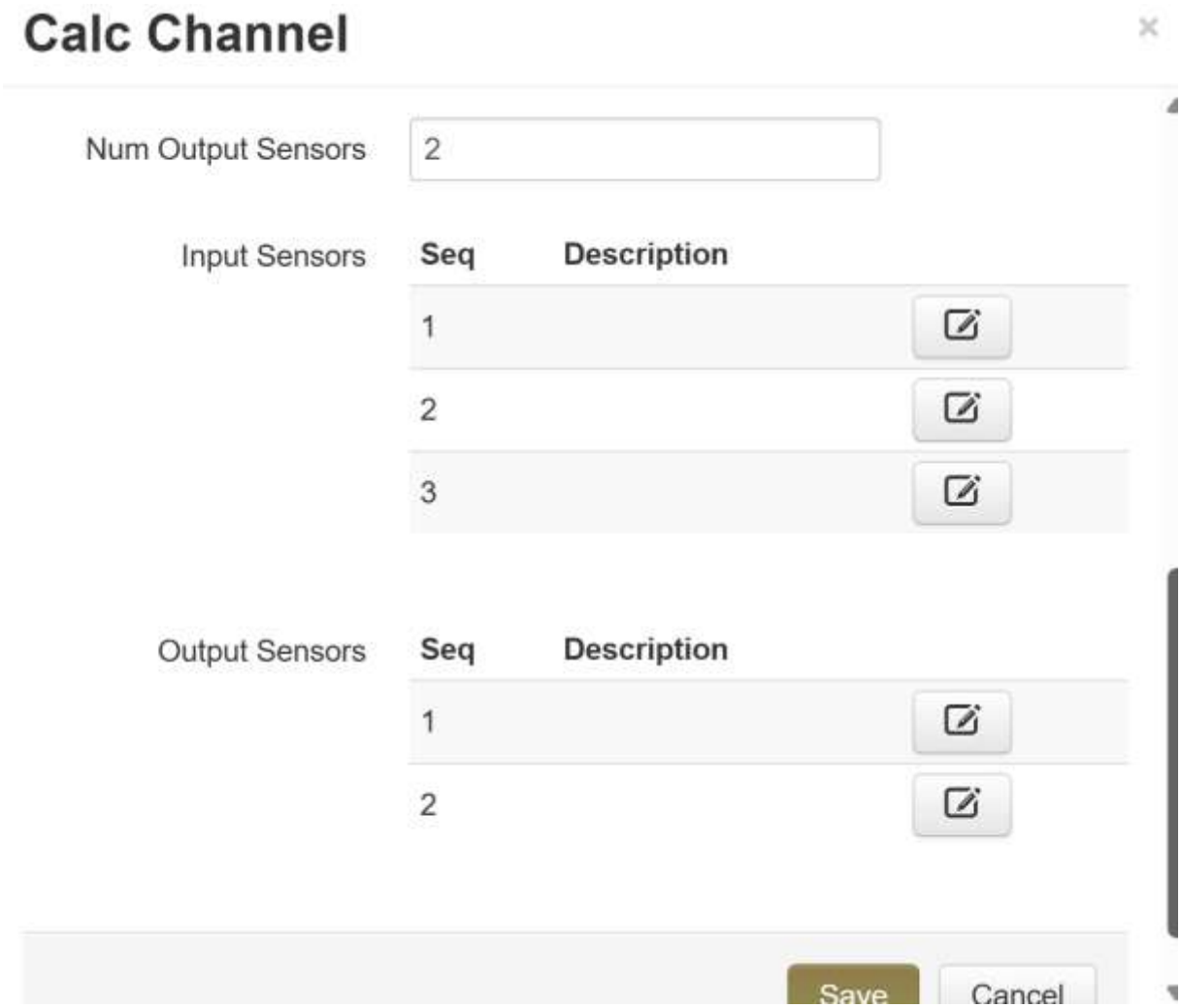
Folllowing parameters are configured here:

- **Code Filename**: the filename for this script. Note: should be unique for all configured script files. If you use the same python script for multiple calculations (with different input/output sensor channel), please adjust the Code Filename for each calc channels (e.g. testcalcchannel_1.py, testcalcchannel_2.py etc). This does not need to match the filename on your local drive when you uploaded the code, since the code is written from the configuration database to a temporary file.
- **Code Func Name**: the function to be called
- **Num Input Sensors**: the number of input sensor channels
- **Num Output Sensors**: the number of output sensors

Change the values to match the script file you are configuring here, e.g.:



Save the configuration to create the slots for the number of input/output sensors



Configure the input and output sensor channels as described above (7.7.1.1 Create Result Sensor)

7.7.3.4 Python Code – UpCom Interface

Note: Python Version is Python2.7

A simple example for a script is as follows

```python
import upcom

def dummyCalculator1(val1, val2):
    result = val1 + val2
    ## print (result)
    return result

def dummyCalculator2(val1, val2):
    result = val1 * val2
    ## print (result)
    return result
def dummyCalculator3(val1, val2):
    result = val2/val1
    ## print (result)
    return result

def calcresult():
    input1 = upcom.getvalue(1) # type: tuple
    input2 = upcom.getvalue(2) # type: tuple
    input3 = upcom.getvalue(3) # type: tuple


    if input1[0] > 0:
        print ("value 1 is error!")
        upcom.setresult( index: 1, status: 1,  value: 0.0)
        return False


    if input2[0] > 0:
        print ("value 2 is error!")
        upcom.setresult( index: 1, status: 1,  value: 0.0)
        return False

    val1 = dummyCalculator1(input1[1], input2[1])
    val2 = dummyCalculator2(input1[1], input3[1])
    val3 = dummyCalculator3(input3[1], input2[1])

    upcom.setresult( index: 1, status: 0, val1)
    upcom.setresult( index: 2, status: 0, val2)
    upcom.setresult( index: 3, status: 0, val3)

    return True

if __name__ == '__main__':
    print ("main")
    calcresult()
```

The script file needs to contain a function that does the calculation (which of course can call other functions…) – in this example "calcresult" – that is called by UpCom to do the processing.

7.7.3.4.1 Getting input values

The "`upcom.getvalue(index)`" function returns the current measure value for the corresponding input sensor channel.

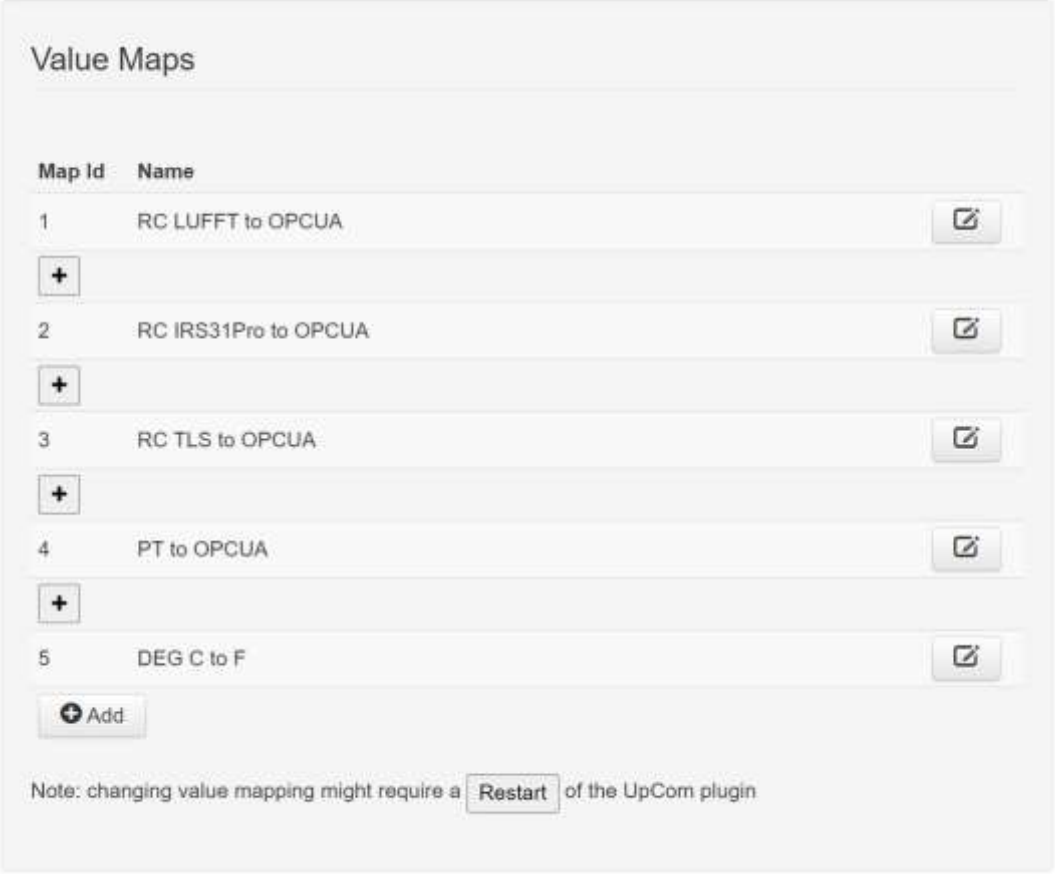
This is a python tuple with following values:

```
Status, value, measure time, device id, sensor channel
```

7.7.3.4.2 Setting result values

The "`upcom.setresult(index, status, value)`" function sets the measure value status (0=ok, 1..0xFF UMB Error codes, > 0xFF extended error codes).

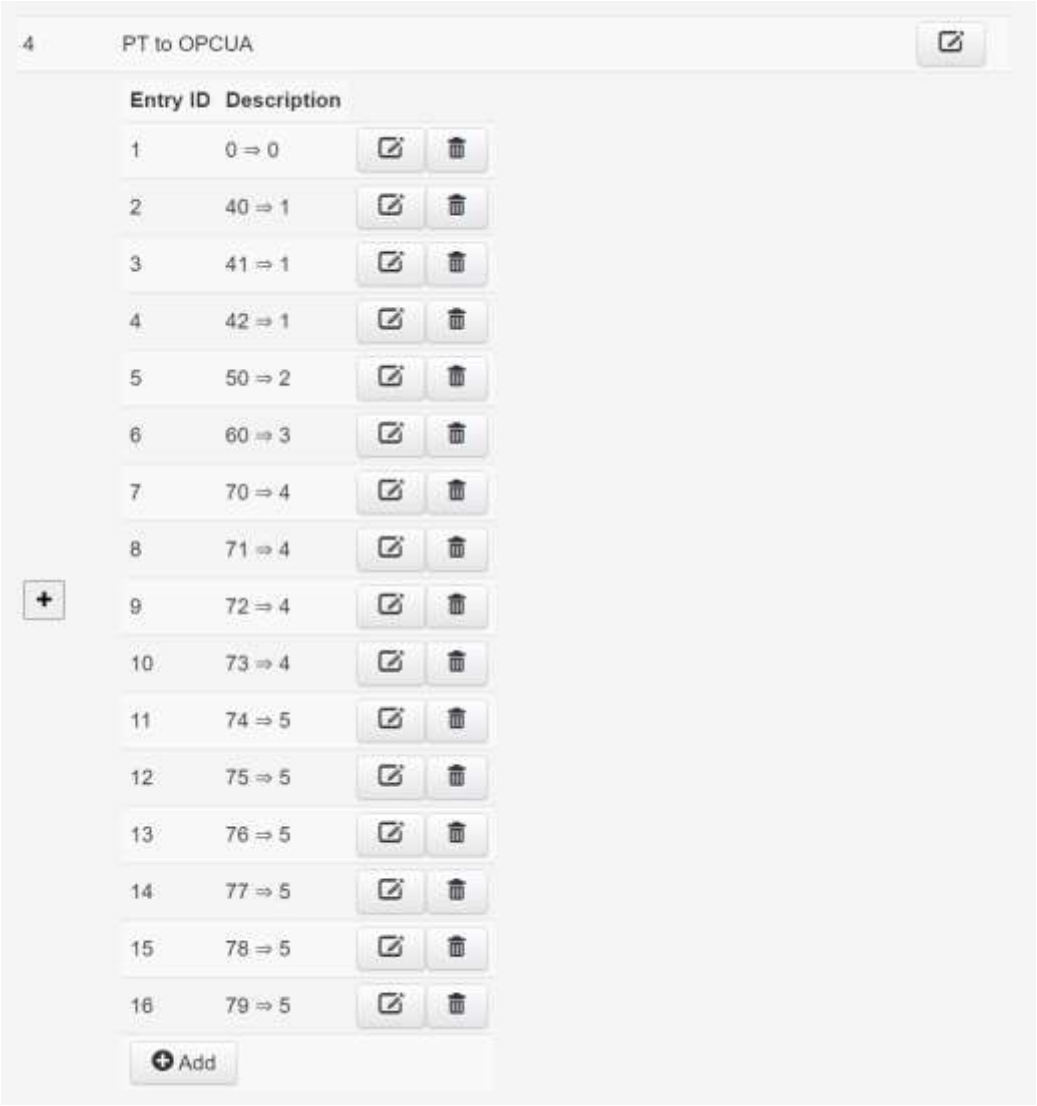
7.8 Value Maps



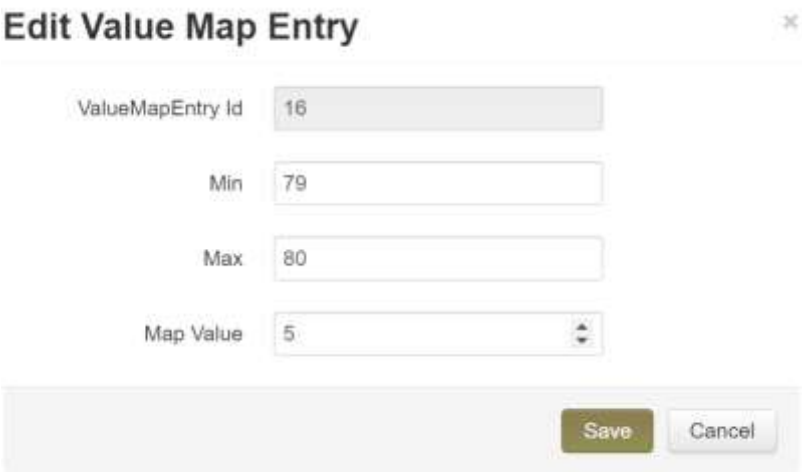
Value mapping can be used to convert a measure value. It might be configured on the UMB sensor channel (see above) and/or on a specific protocol channel (e.g. for OPC-UA, see below)

- Desciption: a description
- Type: the type of value mapping. Possible values are:
  - Scale and Offset: the result is calculated by applying scale and offset factors to the input value – result = ((input * scale1) + offset) * scale2
  - Table: the result is calculated by mapping the input value to a table entry (see below)
  - Scale and Table: both – scaling/offset and table mapping – are applied. The intermediate value is calculated by using the scale/offset parameters, which is then used to look up the result value in the table
- Is Range Type: for tables, this parameter controls weather the input value is compared to a single value or to a value range
- Default value: for tables, this parameter controls which value is applied if no matching table entry can be found.

For "Table" (or "scale and table" value map types, the table mapping entries are shown if the ("+") button is clicked:



A value map entry is configured as follows:

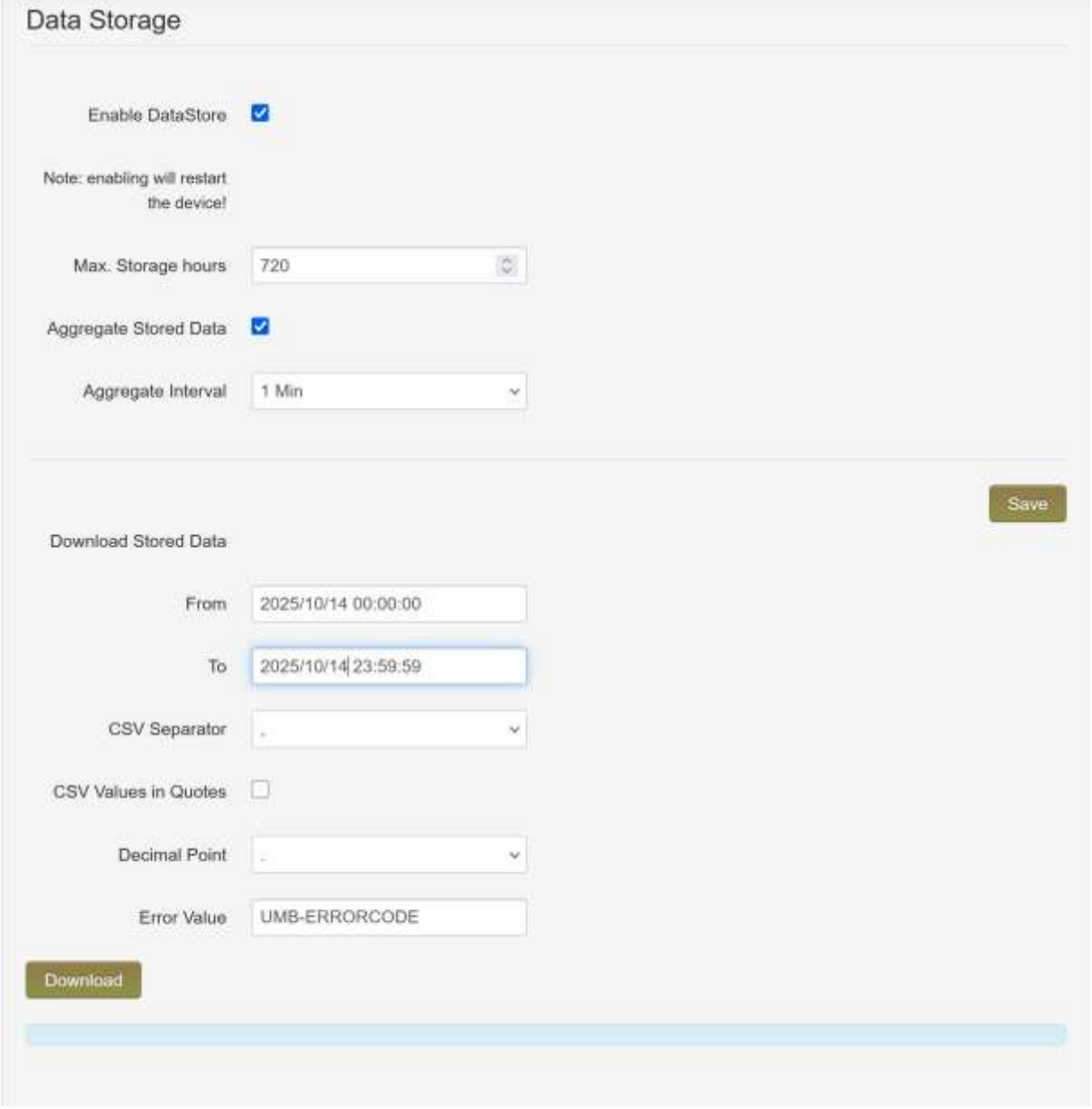


- Min: the minimum value (for a "range" table) or the to be converted input value (for a "not range" table)
- Max: the maximum value (for a "range" table)
- Map Value: the result value

For "range" table types, the entry is matched if the raw value is equal or greater than the min value, and less than the max value

For "not range" table types, the entry is matched if the input is equal to the min value.

7.9 Data Storage



Data storage can be activated (if a microSD card is installed), and the maximum storage hours can be set (default 720 hours = 30 days)

Values can be aggregated when stored to the database, i.e. if the aggregate interval is greater than the device poll interval, the values are aggregated and stored in the aggregate interval instead of the poll interval.

Download Stored Data:

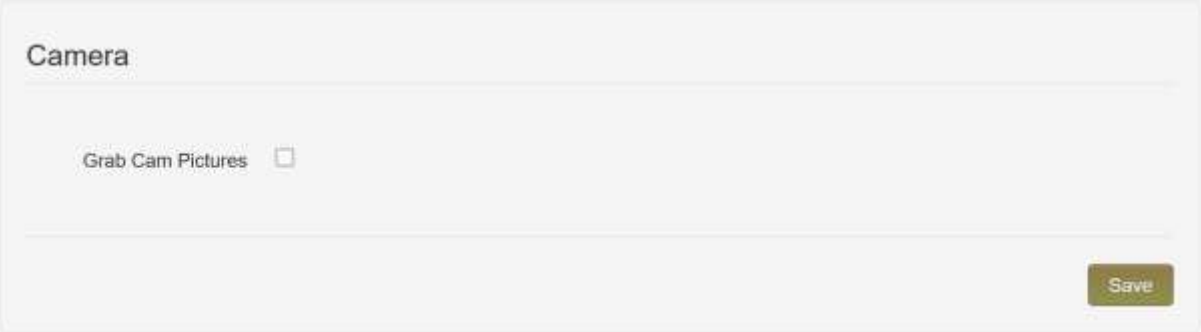
If data storage is enabled, the stored data can be downloaded as CSV text file.

The time interval is pre-set to the current day. Timestamps need to be configured in the format (yyyy/MM/dd HH:mm:ss)!

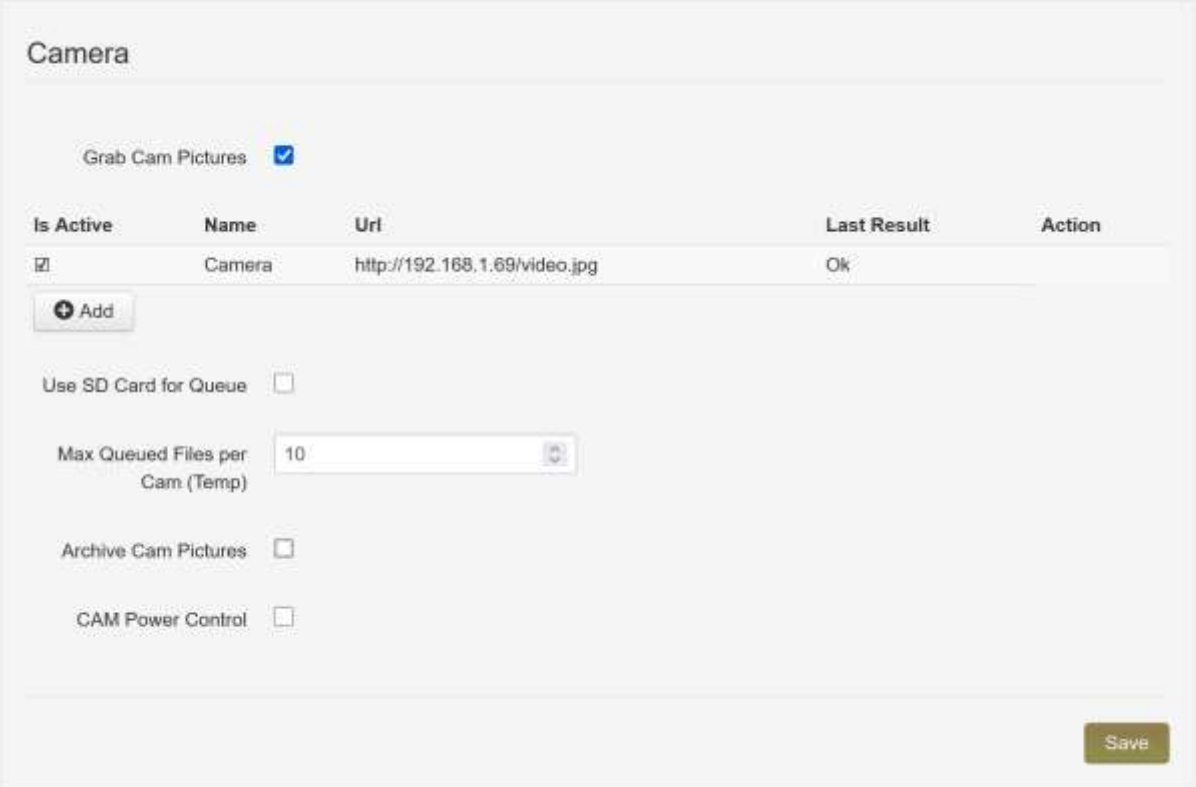
The timestamps are handled in the local time as configured for the device.

Note that – depending on the time settings – extracting and formatting the data might take some time.

7.10 Camera



Here, grabbing a camera picture and transmitting it via uplink protocol to the server can be configured if the uplink protocol supports transmission of camera pictures.



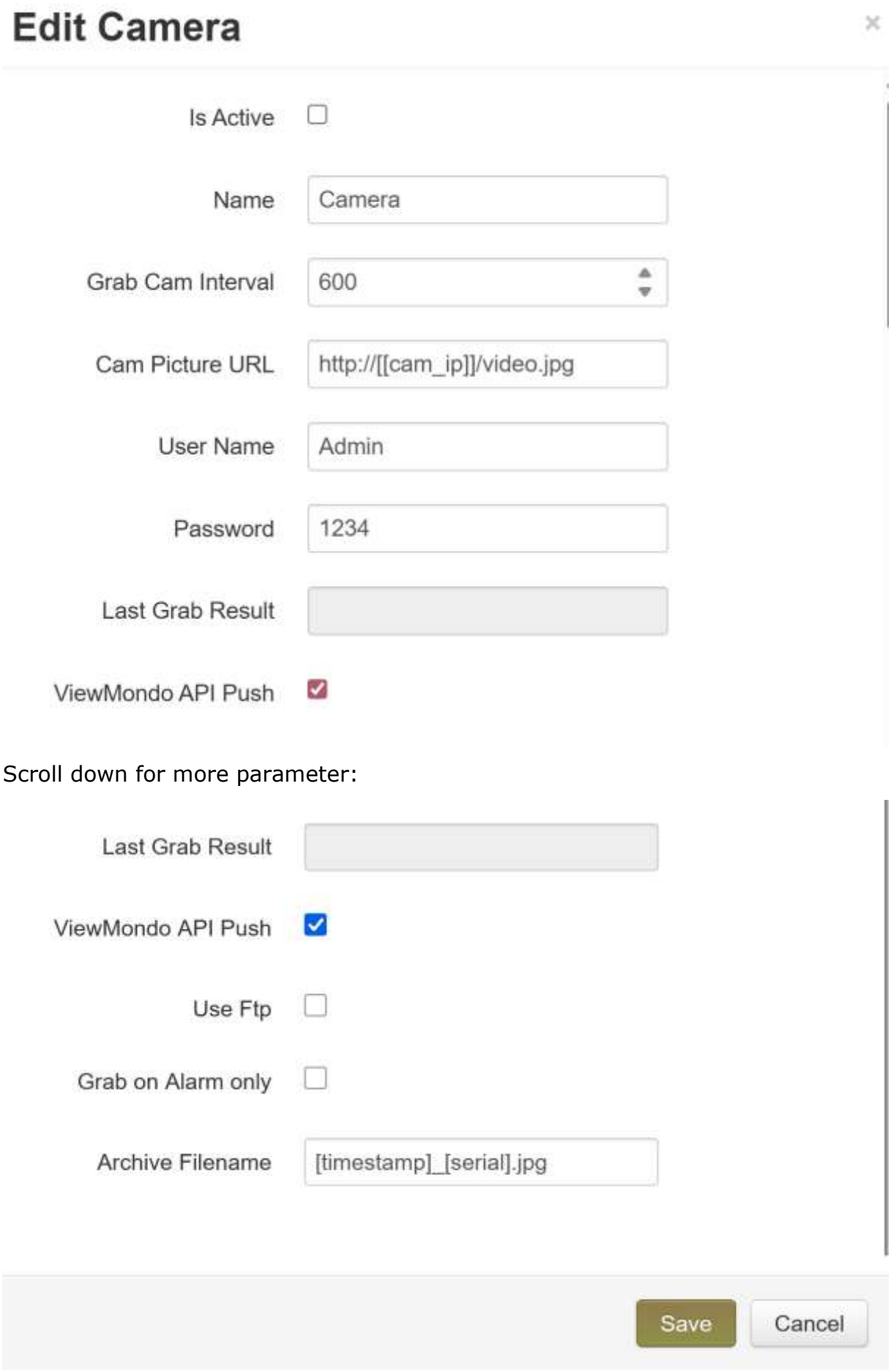
Multiple Cameras can be configured, and the maximum number of files (per camera) to be queued/stored in ram for transmission can be set.

Optional: if the device is equipped with an SD-Card, the SD-Card can be used to store/queue the camera images, making this a non-volatile transfer buffer in

case of a power outage while the network connection is not available.

If the device is equipped with a SD-Card, the camera pictures can also be archived on the SD-Card:

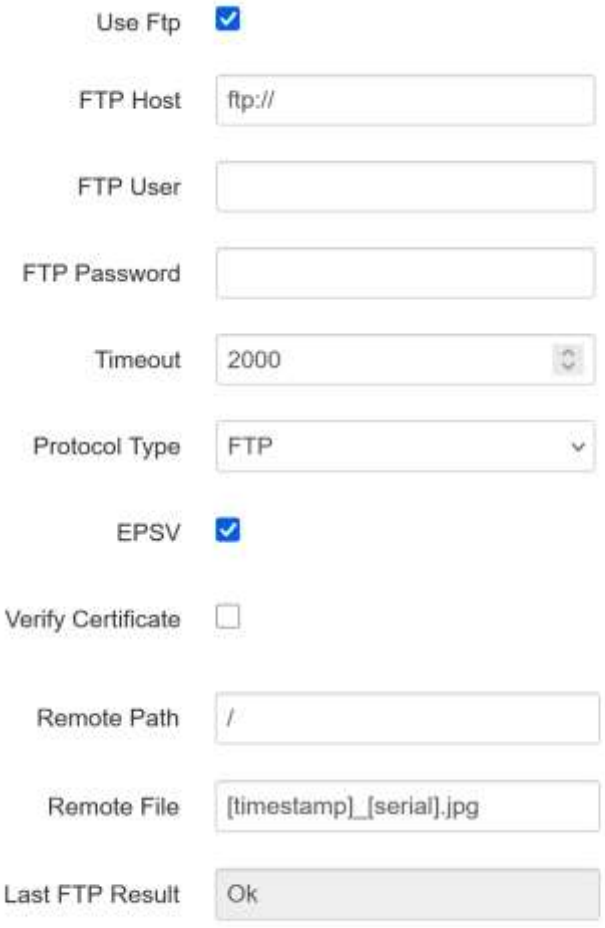
The archive can be accessed via network using “scp” (e.g. WinSCP). The camera images will be stored in the folder /mnt/sda1/archive/cam/{index}/



Scroll down for more parameter:

For each camera, following parameters can be configured:

- Is Active: camera is active / pictures are grabbed
- Name: a name for the camera
- Grab Cam Interval: interval in which camera pictures are grabbed
- Curl options: options for Curl command line (see Curl documentation)
- Cam Picture URL: the url to grab the cam picture from. Note: both http and https are supported
- User Name: the user name to access the camera
- Password: The password to access the camera
- ViewMondo API Push: camera picture is pushed to server via ViewMondo API (if ViewMondo API is active)
- Use Ftp: transfer camera picture using ftp



  - FTP Host: the ftp host url. Note: needs to start with
    ftp:// (plain FTP)
    ftps:// (FTPs)
    sftp:// (sFTP)
  - FTP User: the username
  - FTP Password: the password
  - Timeout: the timeout
  - Protocol type: FTP, FTPs, or sFTP

  - EPSV: passive ftp
  - Verify Certificate (for ftps)
  - Remote Path (needs to exist)
  - Remote File: may contain following tags:
    - [timestamp] the timestamp – format YYYYMMDDhhmmss (e.g. 20131229163250)
    - [date] the date – format YYYYMMDD (e.g. 20131229)
    - [time] the time – format hhmmss (e.g. 163250)
    - [year] the year – format YYYY (e.g. 2013)
    - [month] the month – format MM (e.g. 12)
    - [day] the day – format DD (e.g. 29)
    - [hour] the hour – format HH (e.g. 16)
    - [minute] or [min]: the minute – format mm (e.g. 32)
    - [second] or [sec]: the second format ss (e.g. 50)
    - [utctime] – seconds since 01.01.1970
    - [serial] – device serial string

      NOTE: Time is always formatted in UTC!
  - Last FTP Result: last FTP transfer
- Grab on Alarm only:

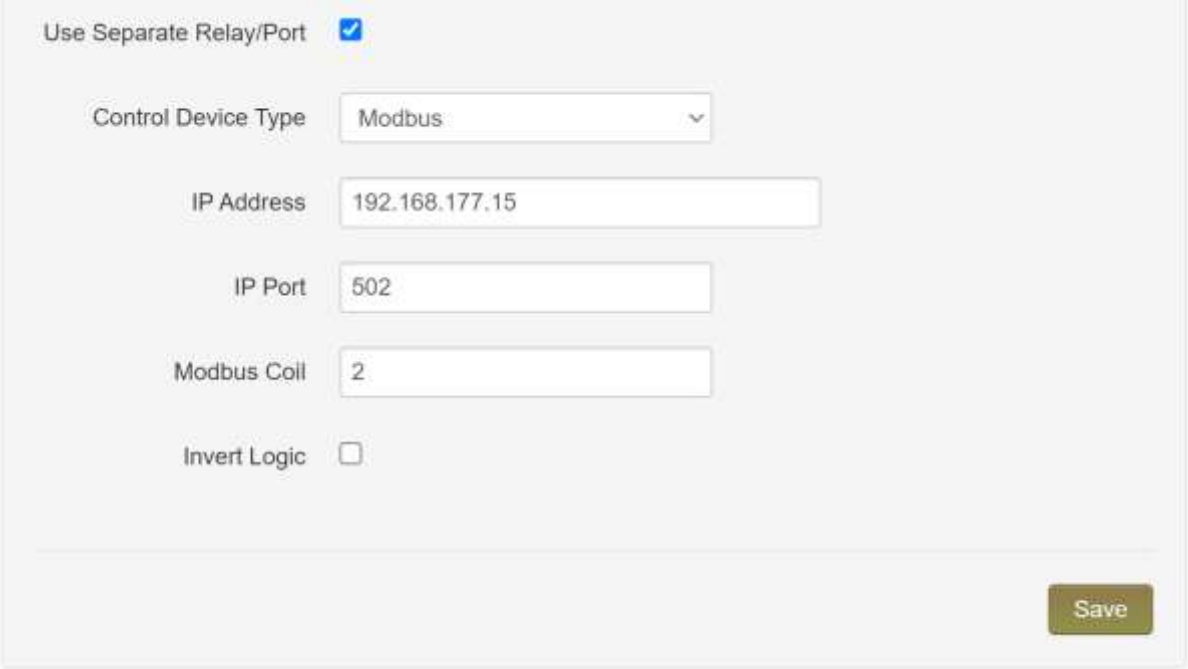
  Select a sensor channel and possibly a value mapping. The picture is only grabbed and forwarded if the resulting value is not 0.

- Archive Filename: a filename template for the archive (if archiving cam pictures is active). Tags see above.

7.10.1 CAM Power Control

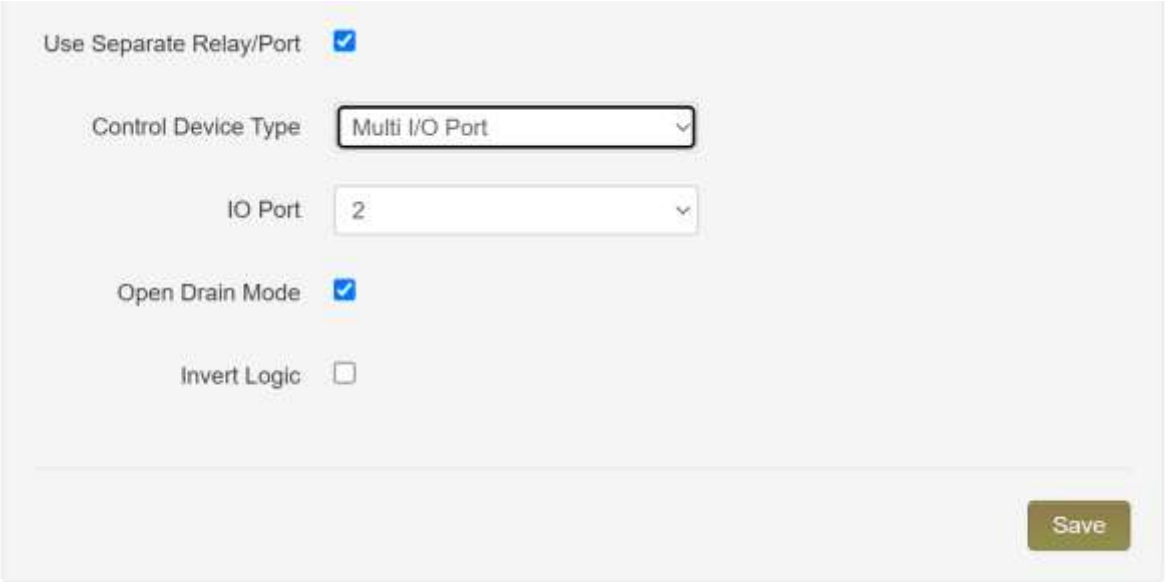
- Power On Delay: the time (in seconds) the camera(s) need to power up. The power will be switched on the configured quantity of seconds before the camera picture(s) are due to be grabbed.
- Use Separate Relay: use a separate Modbus device or coil to control the power for the camera(s).

If no separate relay/Modbus port is configured, the Modbus port configured for UMB power control (see above) will be used for the CAM power control. The power on delay will be calculated from both the UMB and Cam settings, considering the respective poll intervals.



- Control Device Type: Modbus or "Multi IO Port" (Cloudgate Mini only)
- IP Address: the IP address or DNS host name for the Modbus IO device
- IP Port: the IP port for the Modbus IO device
- Modbus Coil: the Modbus coil address where the power relay for the UMB sensors is connected.
- Invert Logic: invert the output logic of the Modbus output port.

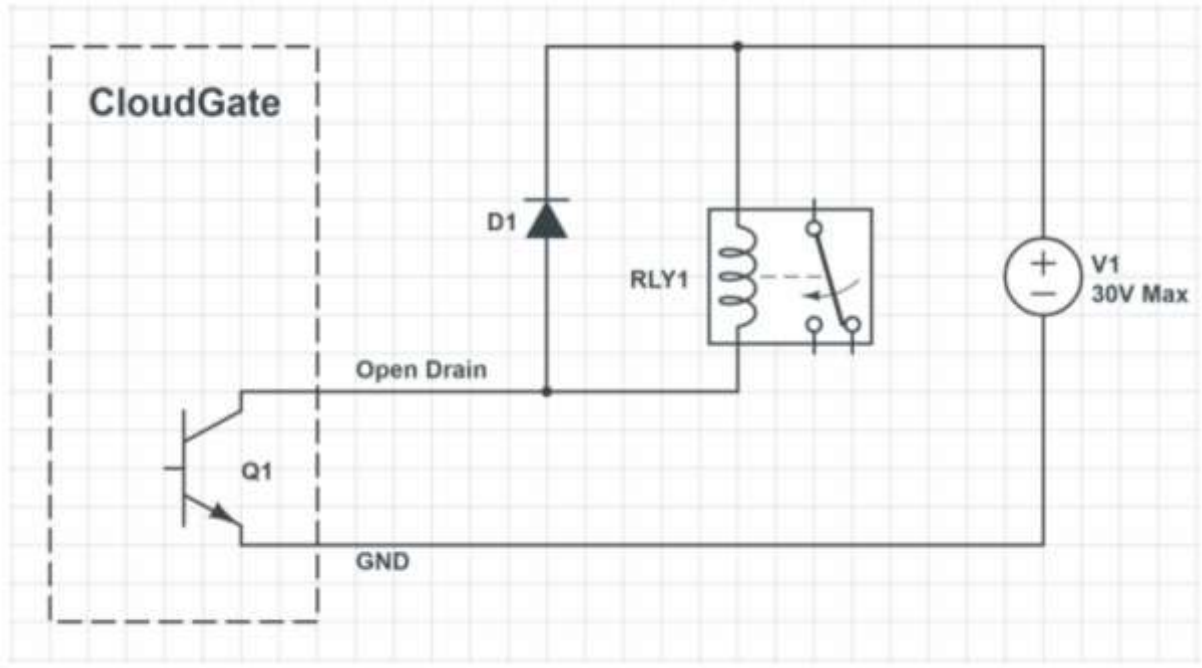
On the Cloudgate MINI with Multip Purpose IO Card, one of the 3 GPIO Ports can be used to control the power for the camera(s)



Note: The output mode for the GPIO port can be either "push/pull" signal, or "open drain".

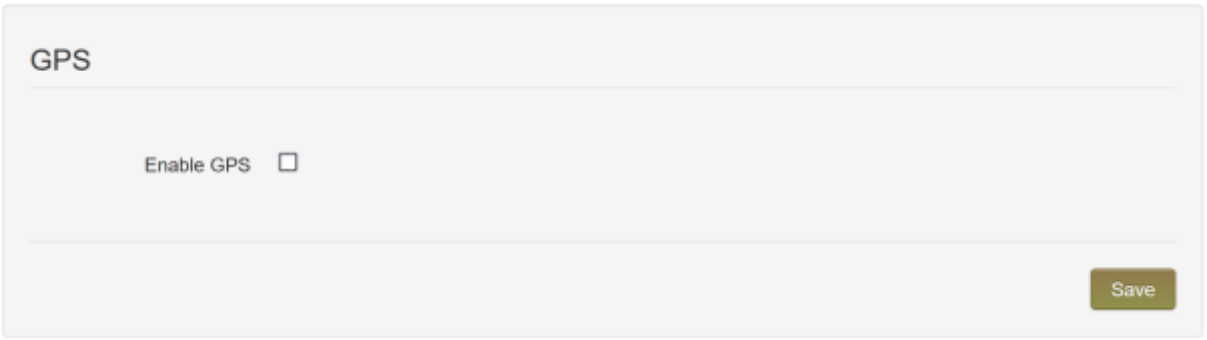
In "push/pull" mode, the signal level for the GPIO Ports is 3.2 Volts – and maximimum 20mA.

When connecting a relay, the best choice is to set your pin as an 'open drain' output. This means that the pin will sink the current from an external power supply when the pin is 'high' (current wil flow) and that no current will flow when the pin is 'low'.



Please note, that a flyback diode must be included in the circuit, and the maximum current must be limited to 250mA to prevent damage to the CloudGate hardware!

7.11 GPS

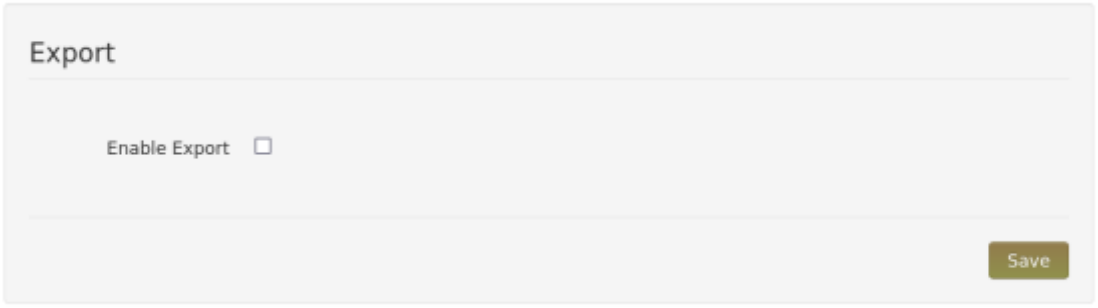


Here, GPS position processing can be enabled if needed/supported by the uplink protocol. Note that this also requires to use an antenna for GPS, and the respective settings for the antenna type must be configured in the "LTE Connection/General" settings (Active GPS Antenna yes/no etc.).

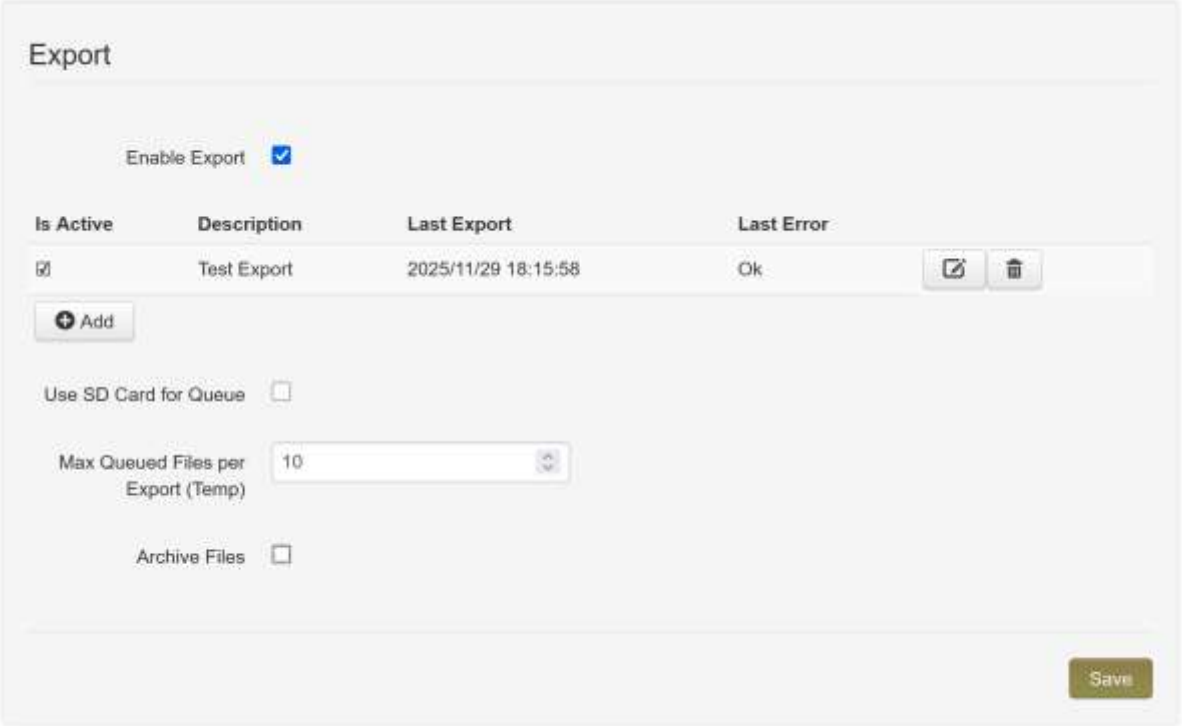
7.12 Export

If Export is part of the UpCom License, export can be enabled, and one or more export "jobs" can be configured.

Note: export can be configured independently from any other "Uplink Protocol", and will run in parallel to the primary Uplink Protocol



If enabled, the configured export jobs are shown (if any), and exports can be added or edited:



General Parameters:

- Max Queued Filer per Export (Temp): maximum number of files to be held in the transfer queue if no connection to the server can be established, and the files are kept in temporary storage (volatile, is lost in case of a power failure)
- Use SD Card: if the device is equipped with an SD card, the files to be transferred can be kept on the SD card, which is non-volatile, so the files will not be lost in case of a power failure.

- Max Queued Files per Export (SD): the maximum number of files to be held in the transfer queue if no connection to the server can be established, and the files are kept on the (non volatile) SD card.

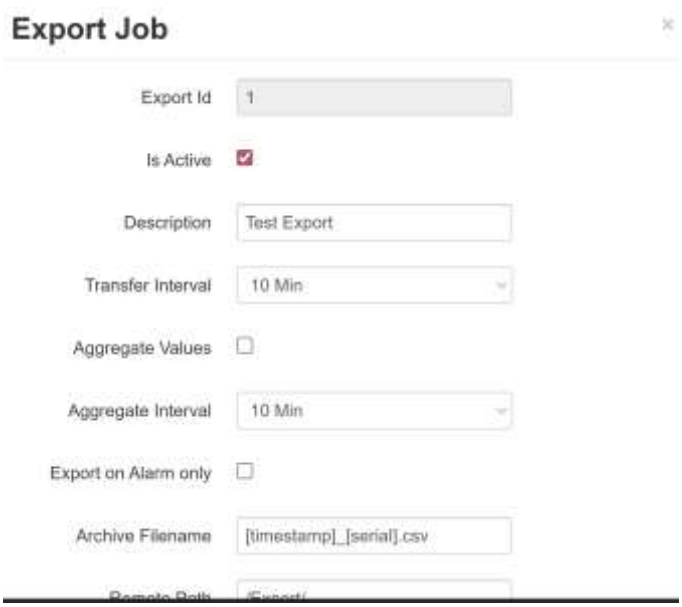
Archive Files: keep an archive of the configured amount of export files – per export – on the SD-Card

The archive can be accessed via network using “scp” (e.g. WinSCP). The export files images will be stored in the folder /mnt/sda1/archive/export/{index}/

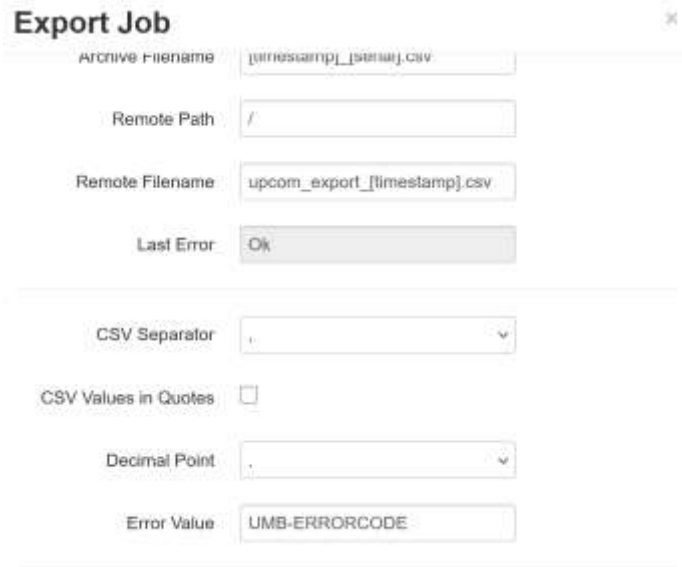
7.12.1 CSV Text Export

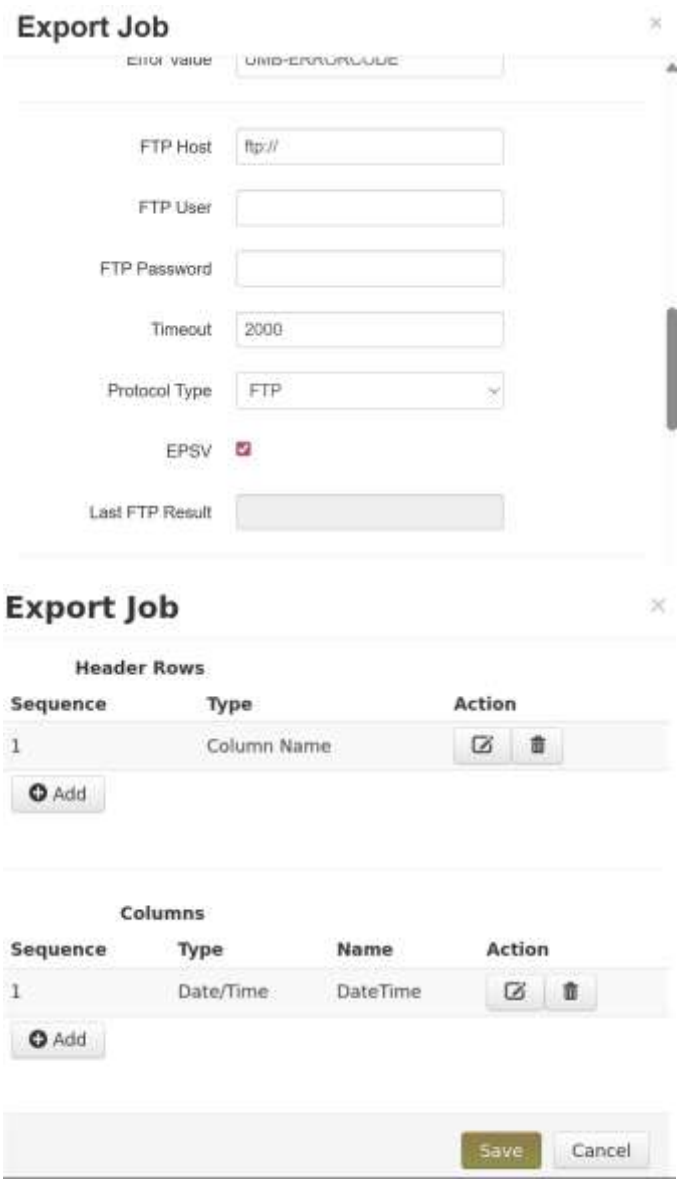
The standard export format is CSV (comma separated values) text.

A CSV text file can contain one or more header lines, and will contain one row for each measurement time that is included in the file.



Scroll down to see other parameters:





The parameters are:

- Export Id: the (internal) id for the export job
- Is Active: is active (or not)
- Description: a description of the export job
- Transfer Interval: interval for the export/ftp transfer. Note: transfers are only created/performed if there is data to export within this time period.
- Aggregate Values: aggregate the values for export
- Aggregate Interval: the interval in which the values are to be aggregated for export.
- Export on Alarm only: only export data if there is a alarm condition

Select a sensor channel and possibly a value mapping. The data is only

exported if the resulting value is not 0.

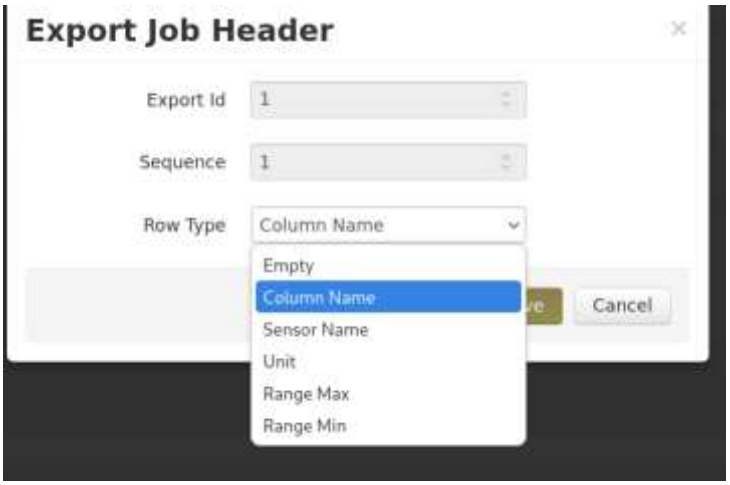
- Archive Filename: a filename template for the archive (if archiving export jobs is active). Tags see below.
- Remote Path: the remote FTP path
- Filename: the filename. Note: the filename might contain following tags that are replaced when the file transfer takes place:
  - [timestamp] the timestamp – format YYYYMMDDhhmmss (e.g. 20131229163250)
  - [date] the date – format YYYYMMDD (e.g. 20131229)
  - [time] the time – format hhmmss (e.g. 163250)
  - [year] the year – format YYYY (e.g. 2013)
  - [month] the month – format MM (e.g. 12)
  - [day] the day – format DD (e.g. 29)
  - [hour] the hour – format HH (e.g. 16)
  - [minute] or [min]: the minute – format mm (e.g. 32)
  - [second] or [sec]: the second format ss (e.g. 50)
  - [utctime] – seconds since 01.01.1970
  - [serial] – device serial string

  NOTE: Time is always formatted in UTC!
- CSV Separator: the CSV Separator
- CSV Values in quotes: if selected, the CSV values (cells) are formatted in quotes
- Decimal Point: the decimal point character
- Error Value: the string used to indicate an error value. Note "UMB-ERRORCODE" will create an error value formatted like `#Error 0x<nn>` where `<nn>` is replaced by the UMB error code in hex
- FTP Host: the ftp host/url. E.g. "`ftp://ftp.viewmondo.com/`". NOTE: the URL prefix is always "ftp://", independent of the protocol type selection (ftp/ftps/sftp)
- FTP user: the ftp user
- FTP password: the password
- Timeout: the timeout value in ms
- Protocol type: FTP, FTPs or sFTP are supported
- EPSV: use epsv mode

For sFTP and FTPs protocol, an additional option to control whether the ssl certificate should be verified can be selected:

7.12.1.1 Export Header Rows

After creating a new job, one header row with type "column name" is added by default. This row can be deleted if no header row should be included in the CSV Export file.

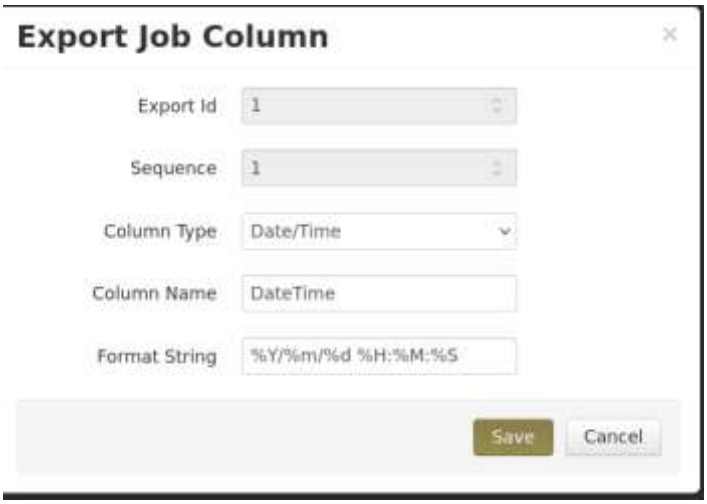


Following header row types can be configured:

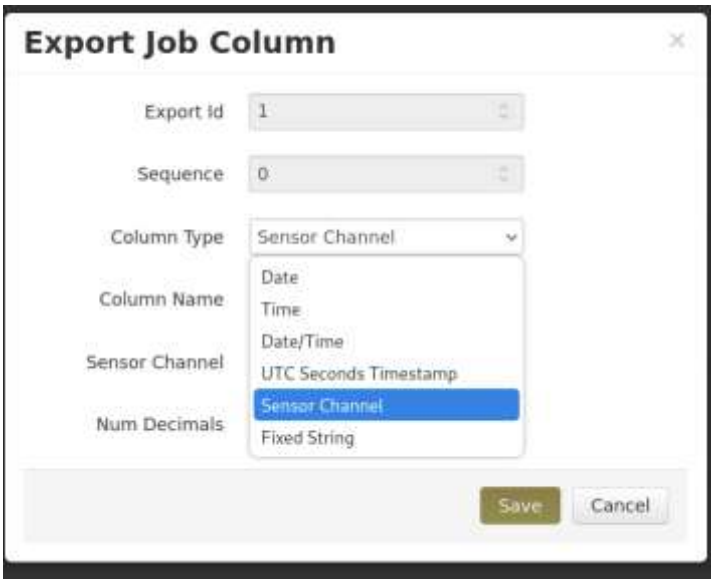
- Empty: an empty row/line
- Column Name: the column name (as configured for the respective column, see below)
- Sensor Name: (for columns of type "Sensor Channel") the respective sensor name
- Unit: (for columns of type "Sensor Channel") the respective Unit
- Range Max: (for columns of type "Sensor Channel") the respective maximum value
- Range Min: (for columns of type "Sensor Channel") the respective minimum value

7.12.1.2 Export Columns

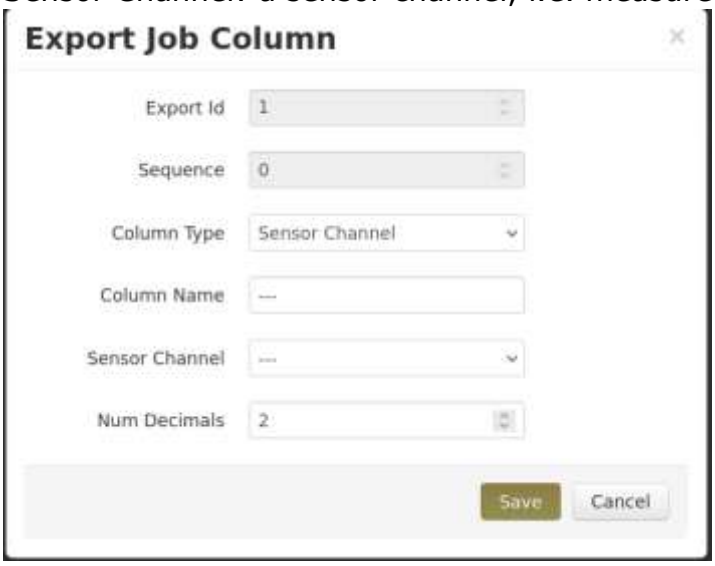
After creating a new job, one default column of type "Date/Time" is configured:



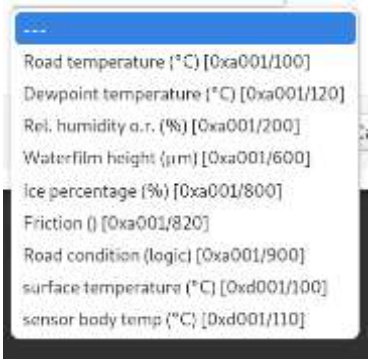
Following column types can be configured:



- Date: the date. Default format: "%Y/%m/%d" for year/month/day.
- Time: the time. Default format: "%H:%M:%S" for hour/minute/second
- Date/Time: the date and time, default format "%Y/%m/%d %H:%M:%S"
- UTC Seconds Timestamp: the date and time in seconds since 01.01.1970
- Sensor Channel: a sensor channel, i.e. measure value for a sensor channel



Here, the actual UMB Sensor channel can be selected from the "Sensor Channel" select box. The select box shows the sensor channel name and unit as reported by the UMB device, and the UMB device id and channel number



Selecting a sensor channel will also pre-set the "Column Name" to the

name of the sensor channel.
The number of decimal places can be configured for this column type (default is 2)

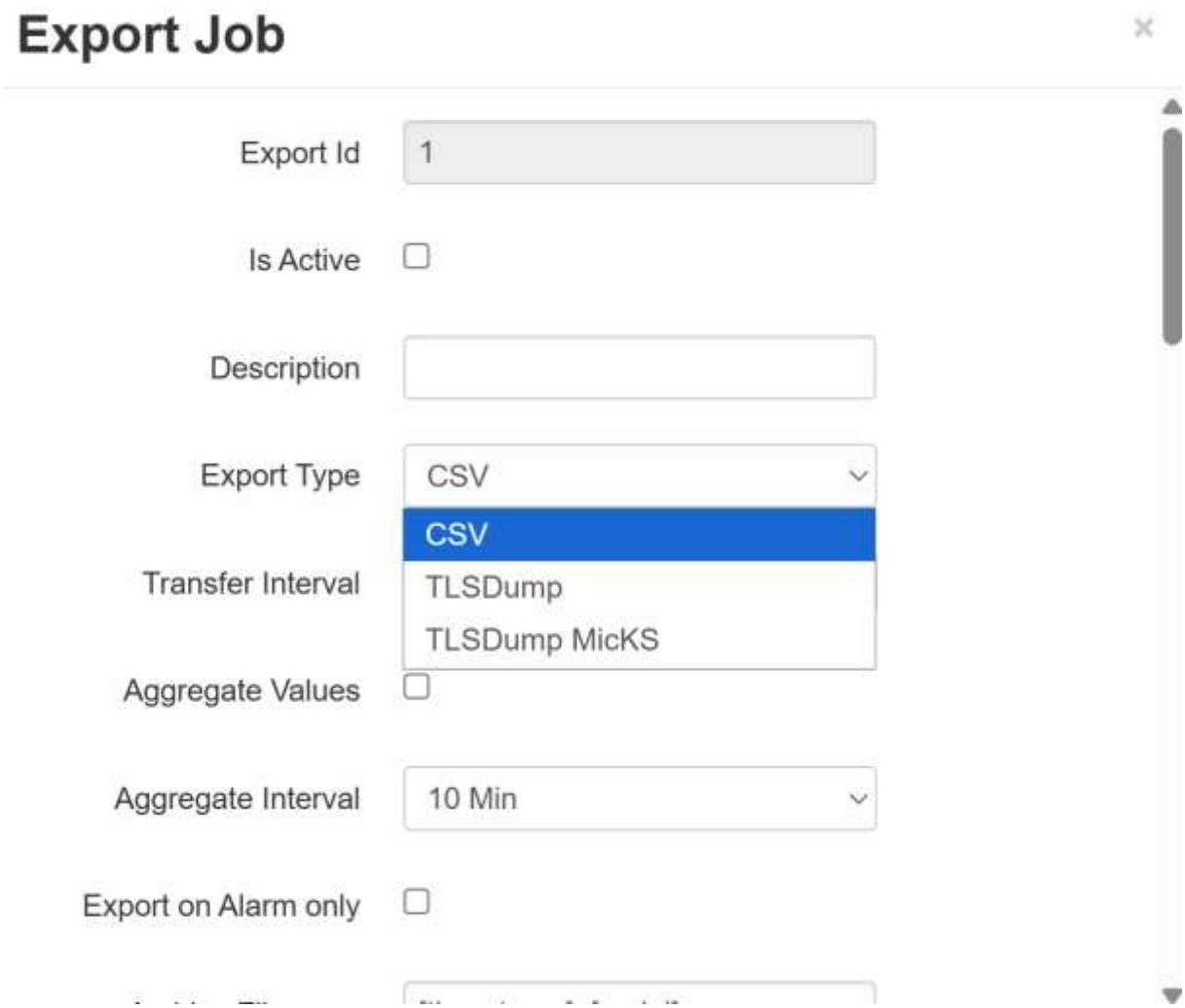
- Fixed String: a fixed string / empty column

7.12.2 TLS Dump / TLS Dump Micks

If the license for the UpCom plugin also contains the German "TLS" Protocol (see 9TLS (Technische Lieferbedingungen für Streckenstationen) Protocol (Germany)), the TLS data can also be exported as a "dump" file in 2 different versions.

Please note that this is actually not part of the official TLS protocol specification, but the format has been used in the past (mainly before TLSoIP was added to the protocol specification) to send data over TCP/IP based networks.

If TLS is active, the additional option "Export Type" is shown on the export job configuration and can be selected:

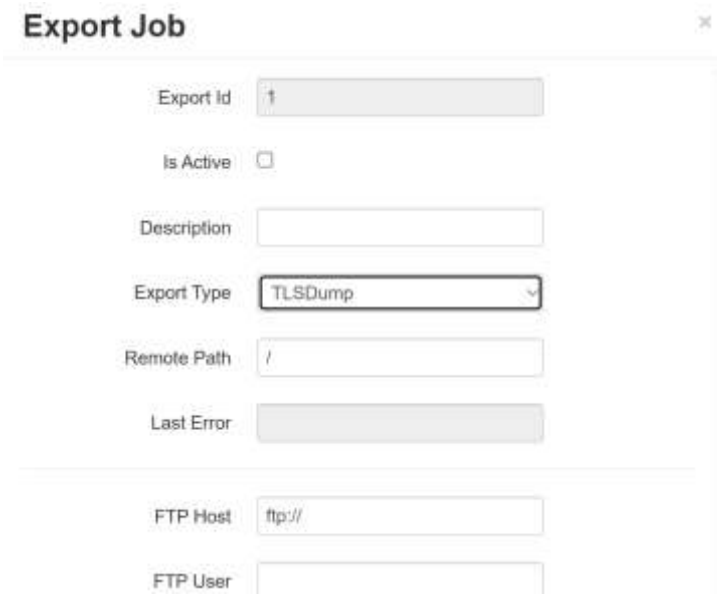


TLSDump: each file contains a TLS "Inselbus" (TC57) telegram with OSI7 content

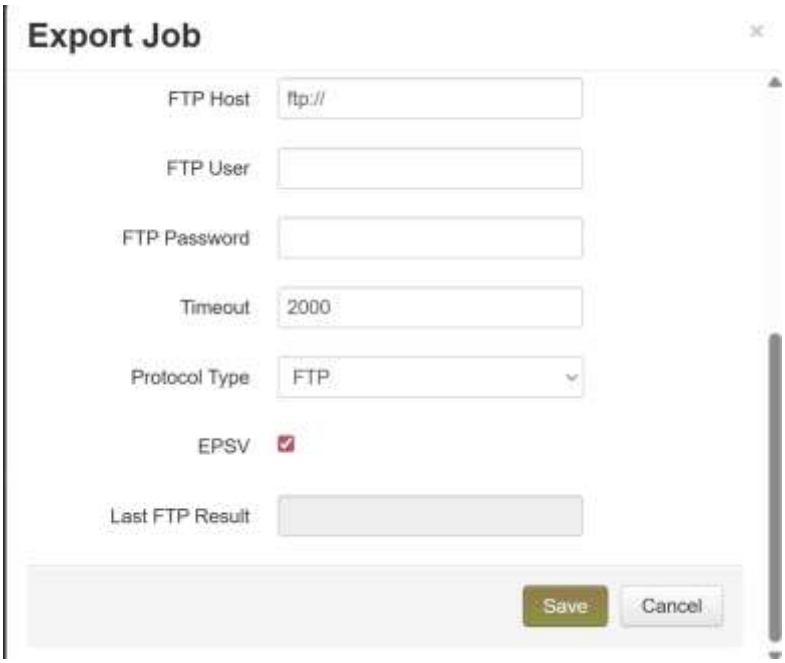
TLSDump (MicKS) : this file also contains a TLS “Inselbus” (TC57) telegramm with OSI7 content – but uses a special header and some characters in the binary data are masked

In both cases, the filename is fixed and will be “KN<osi7_node>_<timestamp>” without a file extension.

If one of these formats is selected, the options for the export job are reduced as follows.



- Export ID: the internal id for the export job
- Is Active: export is active
- Description: a description for the export job
- Export Type: TLSDump or TLSDump (MicKS)
- Remote Path: the path/folder on the ftp server. The actual filename is fixed (see above)



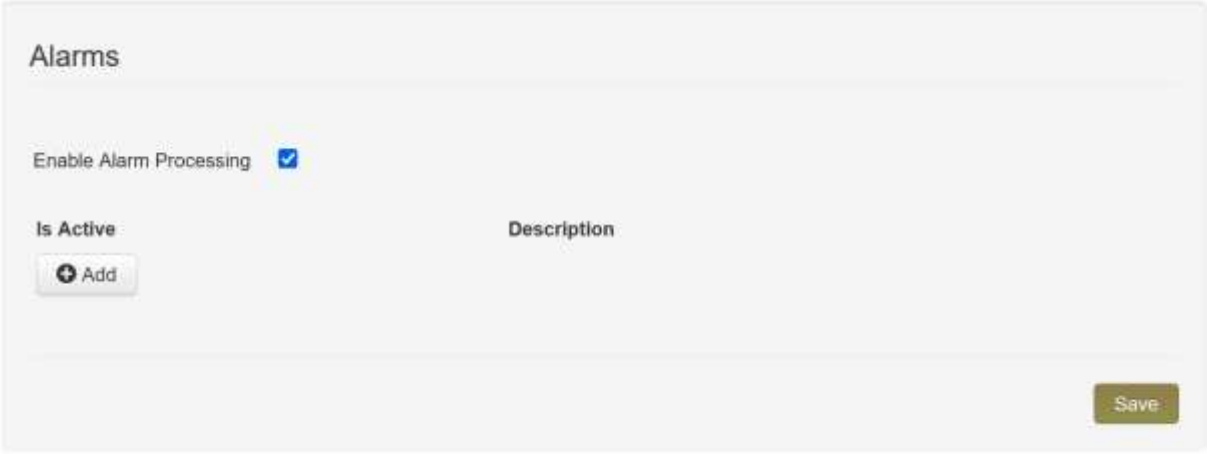
The FTP parameter are the same as for CSV text export, i.e.

- FTP Host: the ftp host (as an URL, always starting with ftp:// sftp:// or ftps://)
- FTP User: the user
- FTP Password: the password
- Timeout: the timeout in ms
- Protocol Type: the ftp protocol type
- EPSV: use passive mode

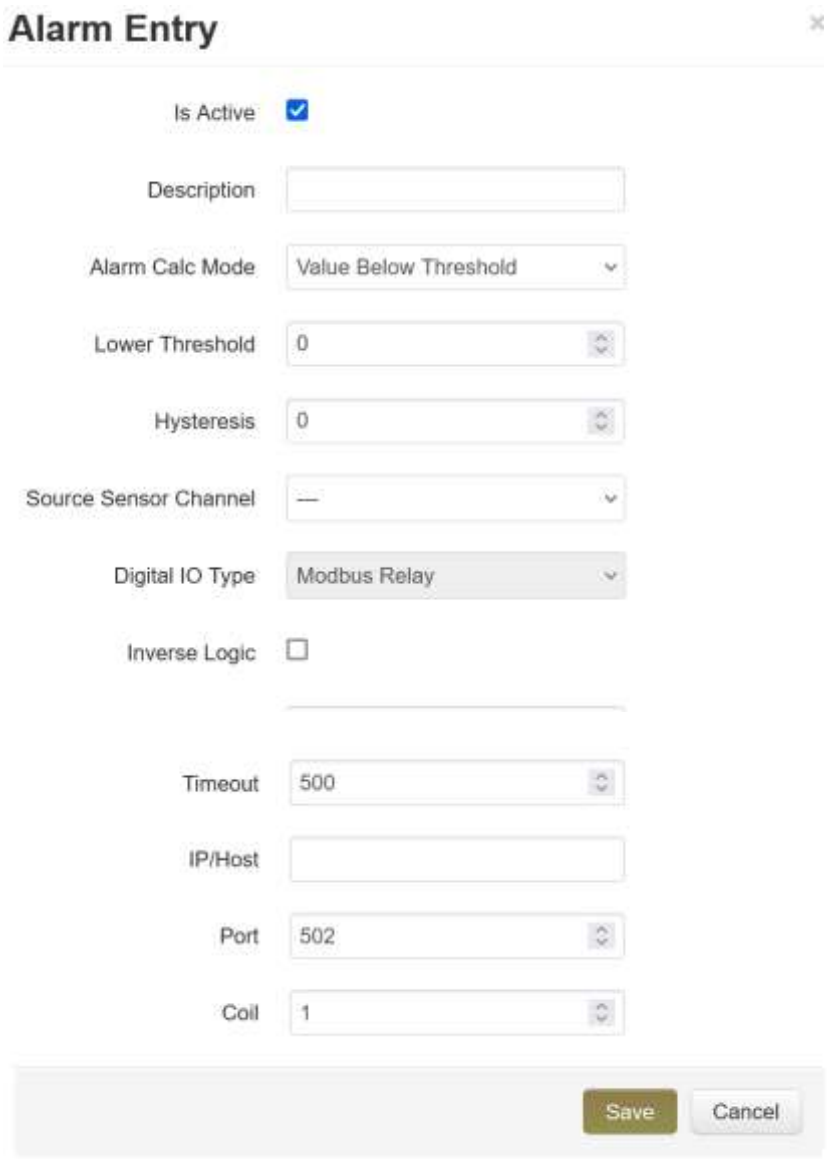
Transmission interval etc. are based on the general TLS parameters configured.

7.13 Alarms

Alarm processing can be used to calculate alarm stati, and control external devices via modbus relais or (on Cloudgate Mini bundle) GPIO ports (see sensor power control/cam power control for details on GPIO).



Multiple alarms can be configured



- Description: a description for the alarm calculation
- Alarm Calc Mode: the calc mode:
  - Value below threshold: alarm is activated if the value is below the configured threshold
  - Value above threshold: alarm is activated if the value is above the configured threshold
  - Value Map: alarm is activated if the value as calculated by the value map is not 0

- Hysteresis: for below/above threshold, the hysteresis is used as a buffer to switch off the alarm status – i.e. the alarm is de-activated if the alarm is active and the value is above/below the threshold minus/plus the configured hysteresis value. For value mapping alarms, the configured

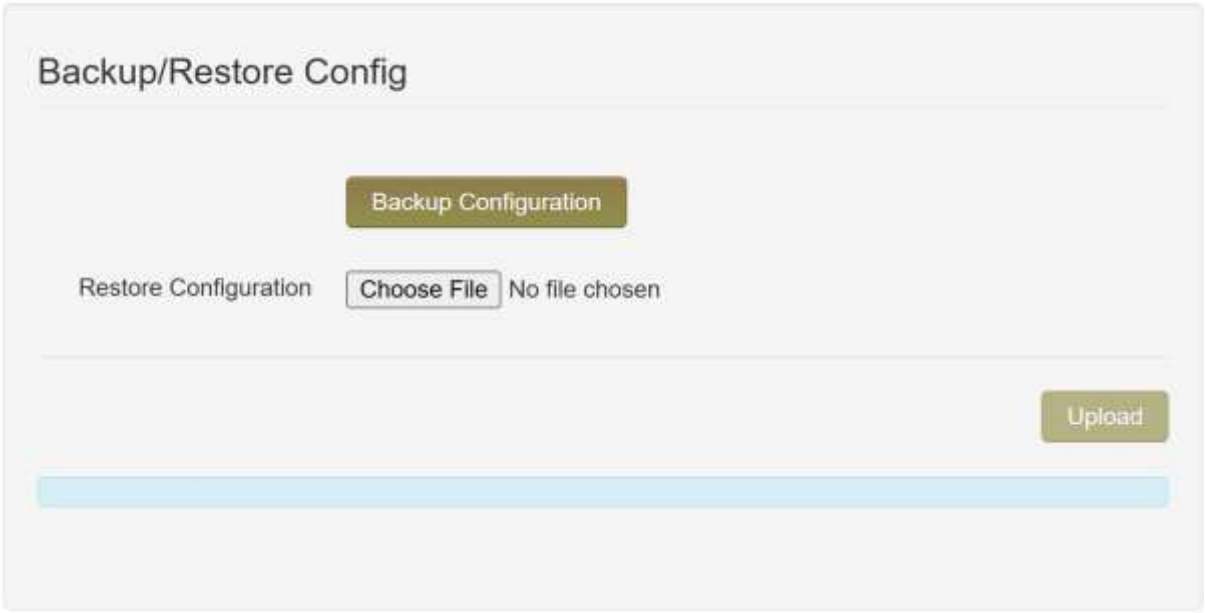
counter is used instead – i.e. the mapped value has to be at least "counter" times 0 for the alarm status to be switched off

- Digital IO Type: Modbus relay or GPIO (Cloudgate Mini only)
- Inverse Logic: inverse output logic
- Timeout: communication timeout with the device
- IP/Host: the IP address or DNS host name for the Modbus device
- Port: the IP port for the Modbus device
- Coil: the Modbus coil number

7.14 Backup/Restore Configuration

Here, the complete device configuration (i.e. sensor channels, calc channels, value mappings, uplink protocol settings, ntcip settings etc) can be backed up in a file, and restored.

If it is restored on the same device (same serial number), the license key is also restored (if there is no valid license key on the device)

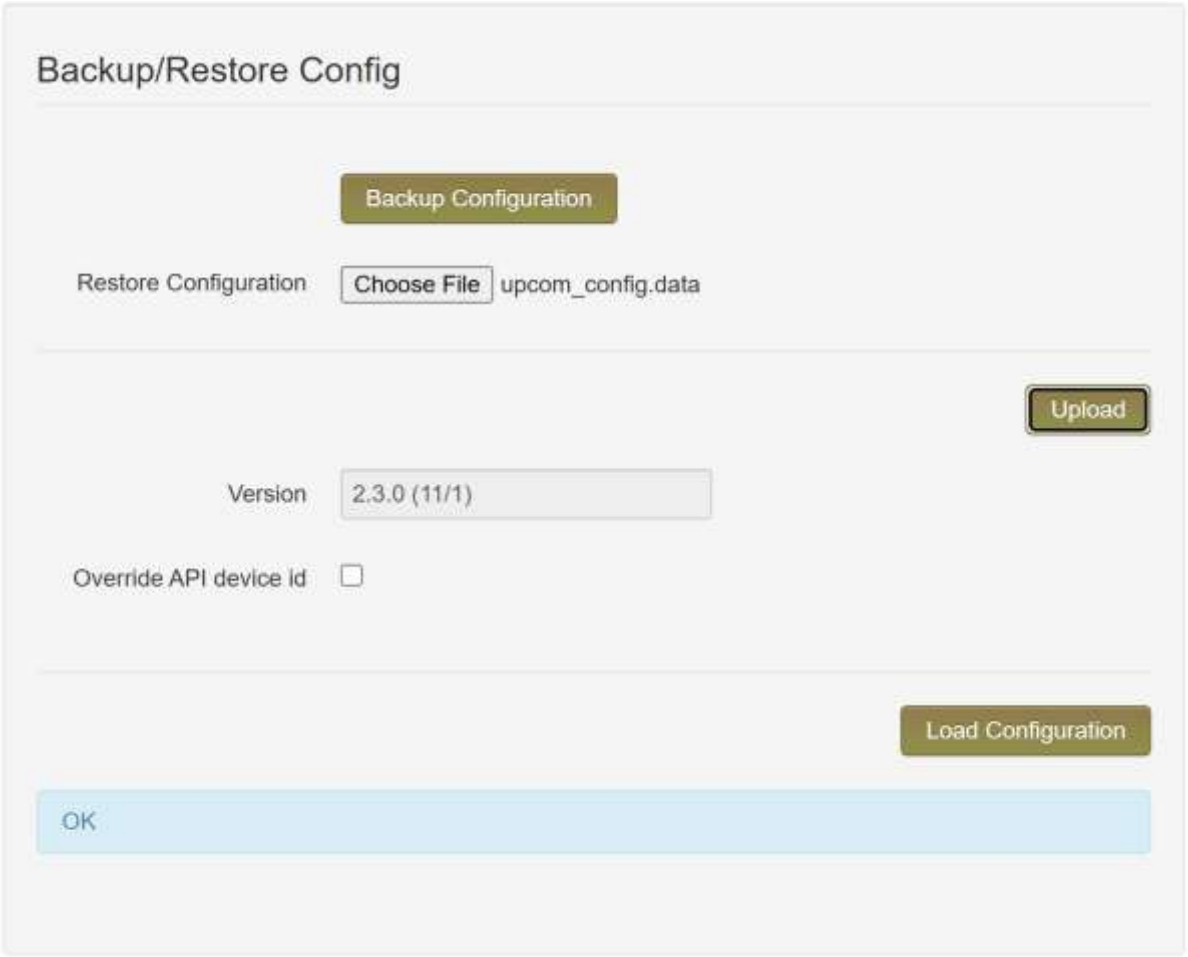


Clicking on "Backup Configuration" will download the configuration data to a file "upcom_config.data".

To restore the configuration, click "Choose File"

"Upload" will be activated if a file has been chosen. Click "Upload" to upload the file.

If the file is a valid UpCom configuration file, the version information is shown, and the configuration can be loaded:

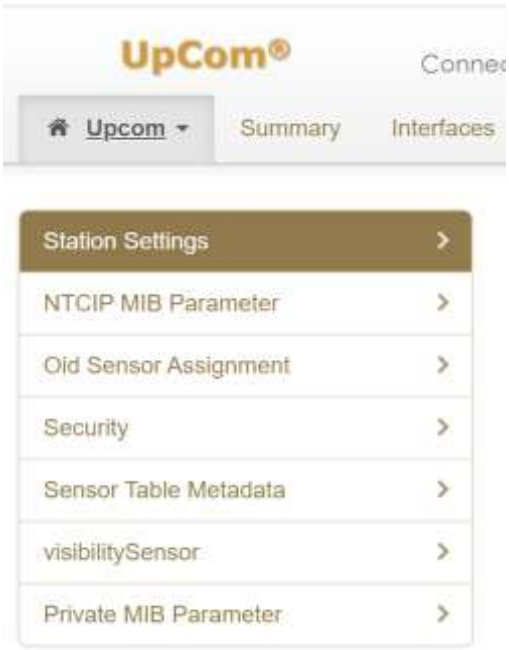


Override API device id: if active, the API device id (which per default contains the serial number) is overridden. This can be used if the CloudGate is used to replace an existing device. Note that the new device (which has a different serial number) requires a new license key.

8 NTCIP

If the UpCom license includes NTCIP, a separate menu entry for NTCIP settings is available.

This page allows configuration of all parameters related to NTCIP.

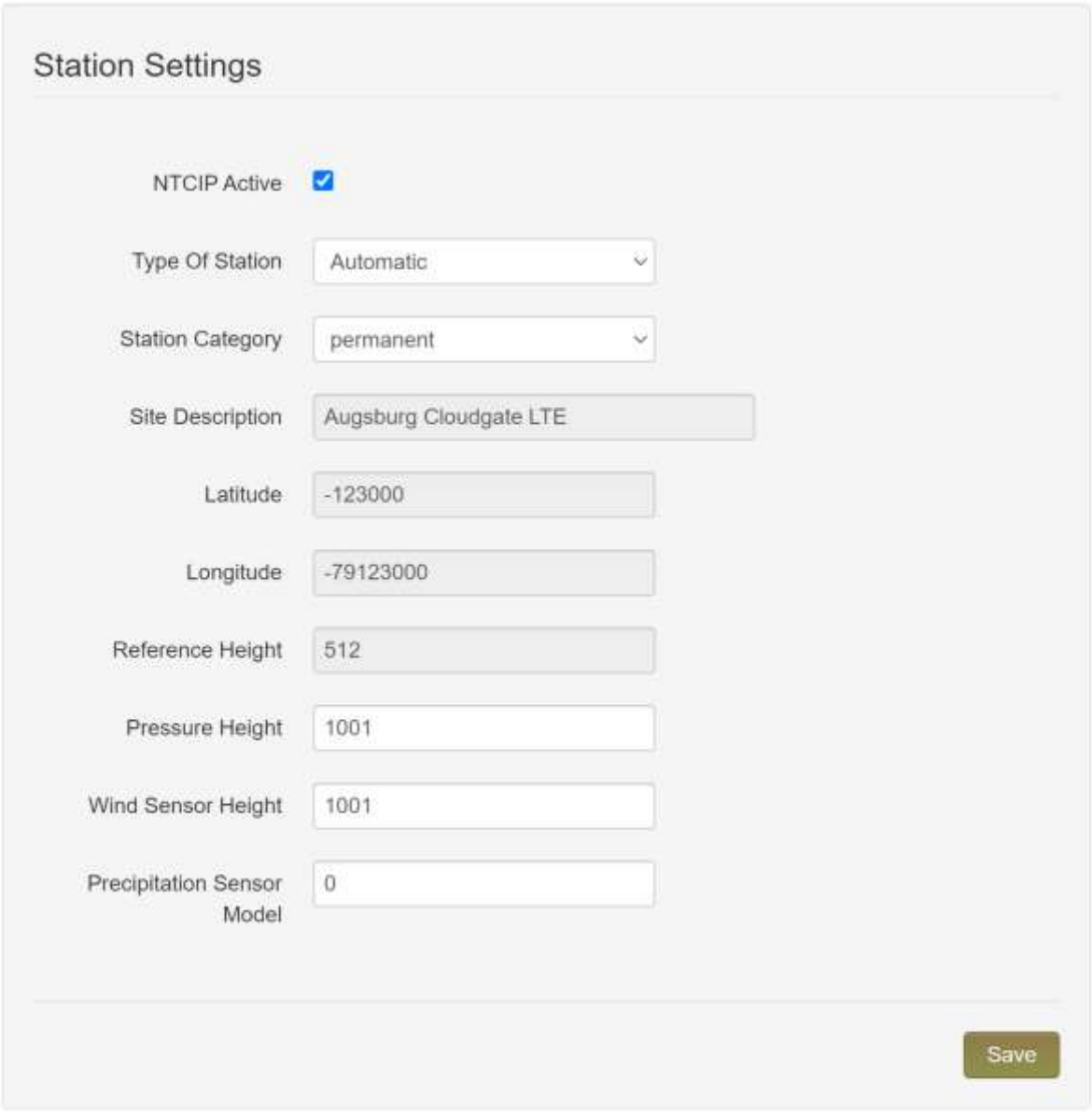


NTCIP can be activated independently from other "Uplink Protocols" like ViewMondo API, and will run in addition to the "main" Uplink Protocol.

Note: if "UMB Direct Access" is configured as primary Uplink Protocol, no measure values are collected, and will not be available for NTCIP either.

8.1 Station Settings

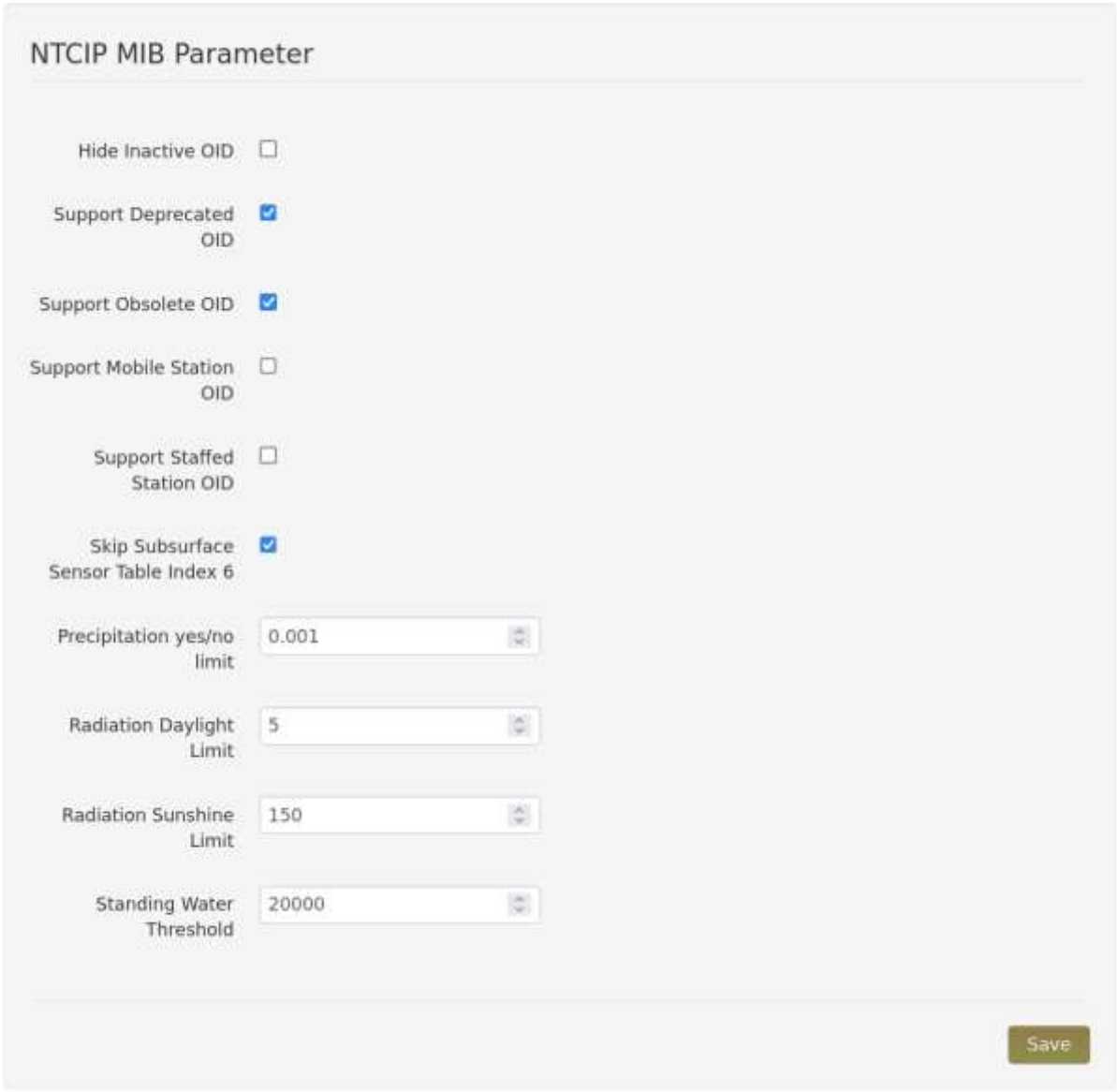
Here, some basic NTCIP station settings can be configured:



Site Description, Latitude, Longitude and reference height are configured on the main “Settings/Station Information” screen – see above.

8.2 NTCIP MIB Parameter

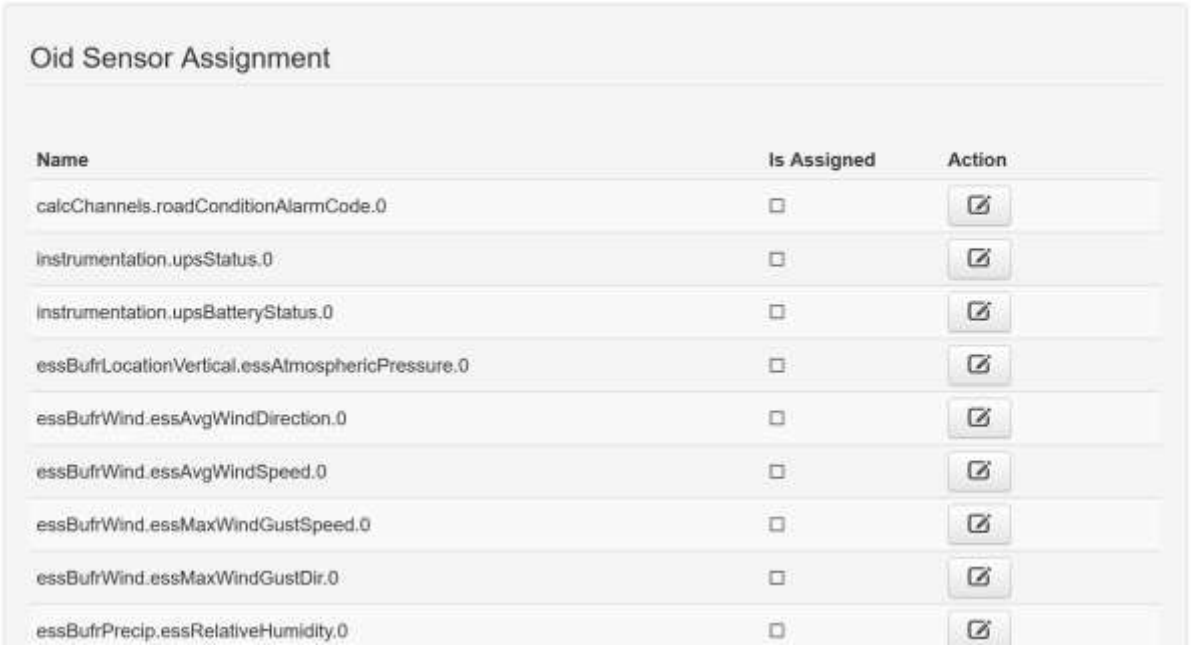
Here some parameters for the NTCIP are configured:



- Hide Inactive OID: OID that are not active will not be reported
- Support Deprecated OID: include OID that are marked as "deprecated" in the NTCIP specification
- Support Obsolete OID: include OID that are marked as "obsolete" in the NTCIP specification
- Support Mobile Station OID: include OID for mobile stations
- Support Staffed Station OID: include OID for staffed stations
- Skip Subsurface Sensor Table Index 6: skip or do not skip index 6 in the subsurface sensor table
- Precipitation yes/no limit: limit for precipitation detection
- Radiation daylight limit: limit for daylight
- Radiation sunshine limit: limit for sunshine
- Standing water threshold: limit for "standing water"

8.3 OID Sensor Assignment

Here, the sensor channels are assigned to the respective NTCIP OIDs. Only "Sensor" OIDs are shown in this list.



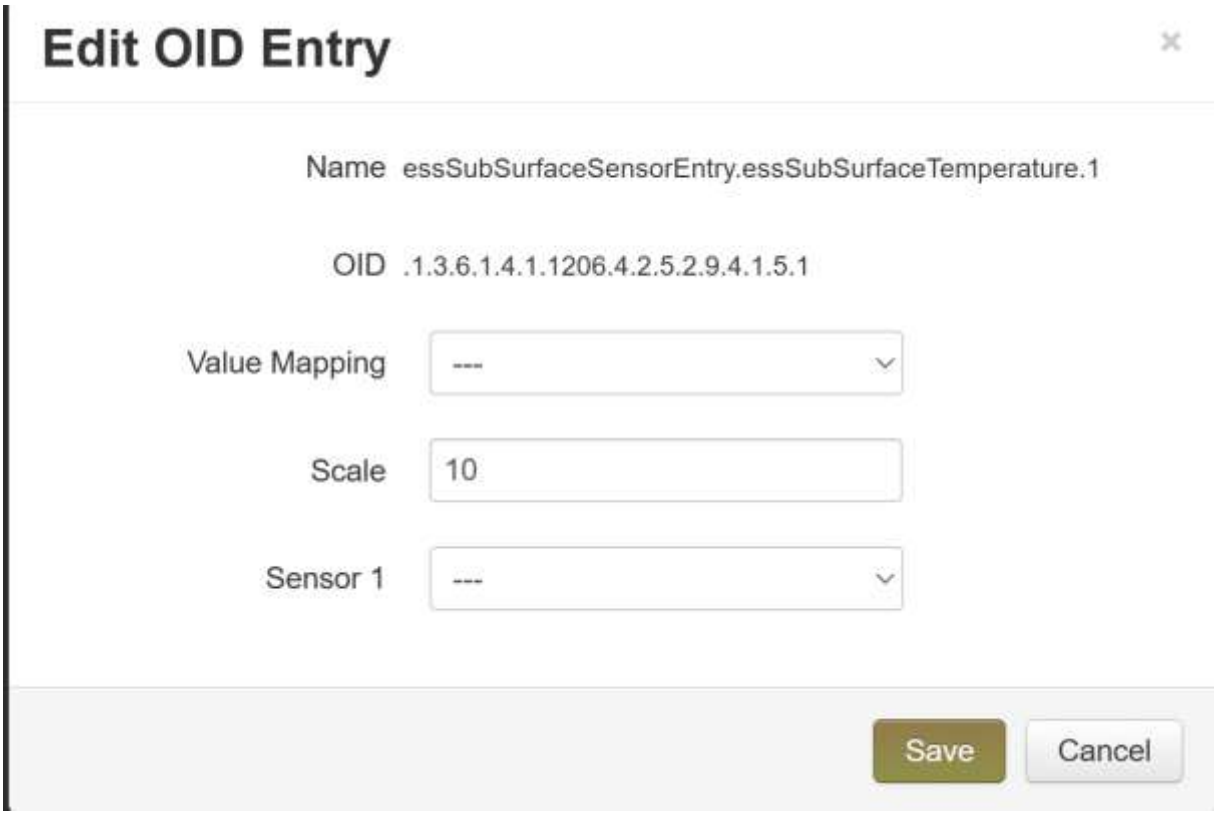
The OIDs shown depend on the MIB Parameters (see above) and other settings (see below), especially the number of entries in the various sensor tables.

Note: if – after changing the MIB settings or sensor table settings – the OID list doesn't show the expected entries, please re-load the page in the browser.

Below the OID list, the current OID-Sensor assignment can be downloaded as a CSV text file:

If an OID has a sensor assignment, the assignment can also be removed:

Depending on the OID type, one or more input sensors can be configured for the respective OID:

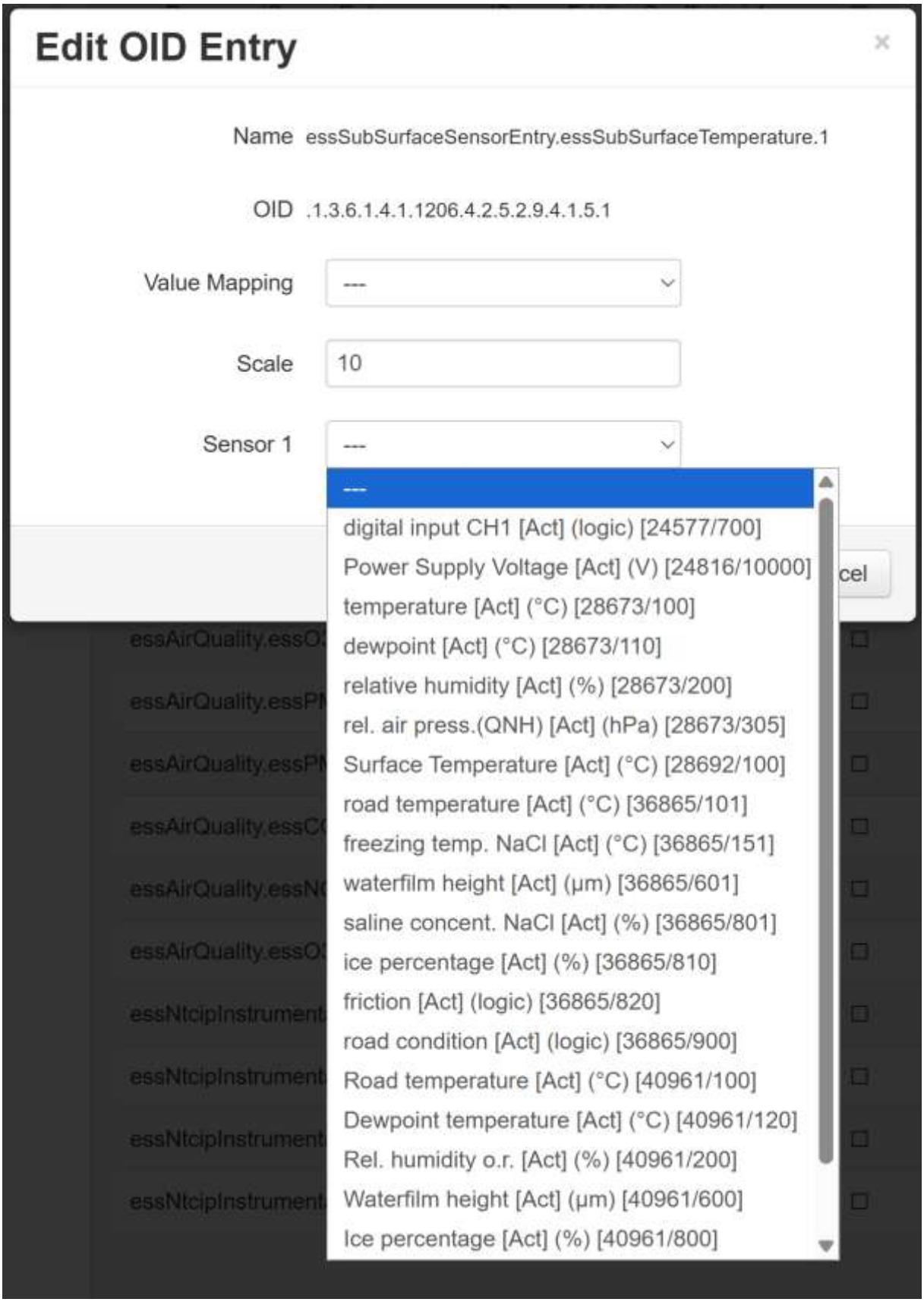


Value Mapping: the value mapping to be applied to the sensor value. NOTE: This value mapping is independent of the value mapping that might be configured on the sensor channel itself. It is only applied for the NTCIP value, and calculated after any value mapping that might be configured on the sensor channel

Scale: the scaling factor to be applied to the sensor value. NOTE: this scaling factor is independent of any scaling that might be configured on the sensor channel itself. It is only applied for the NTCIP value, and calculated after any scaling that might be configured on the sensor channel

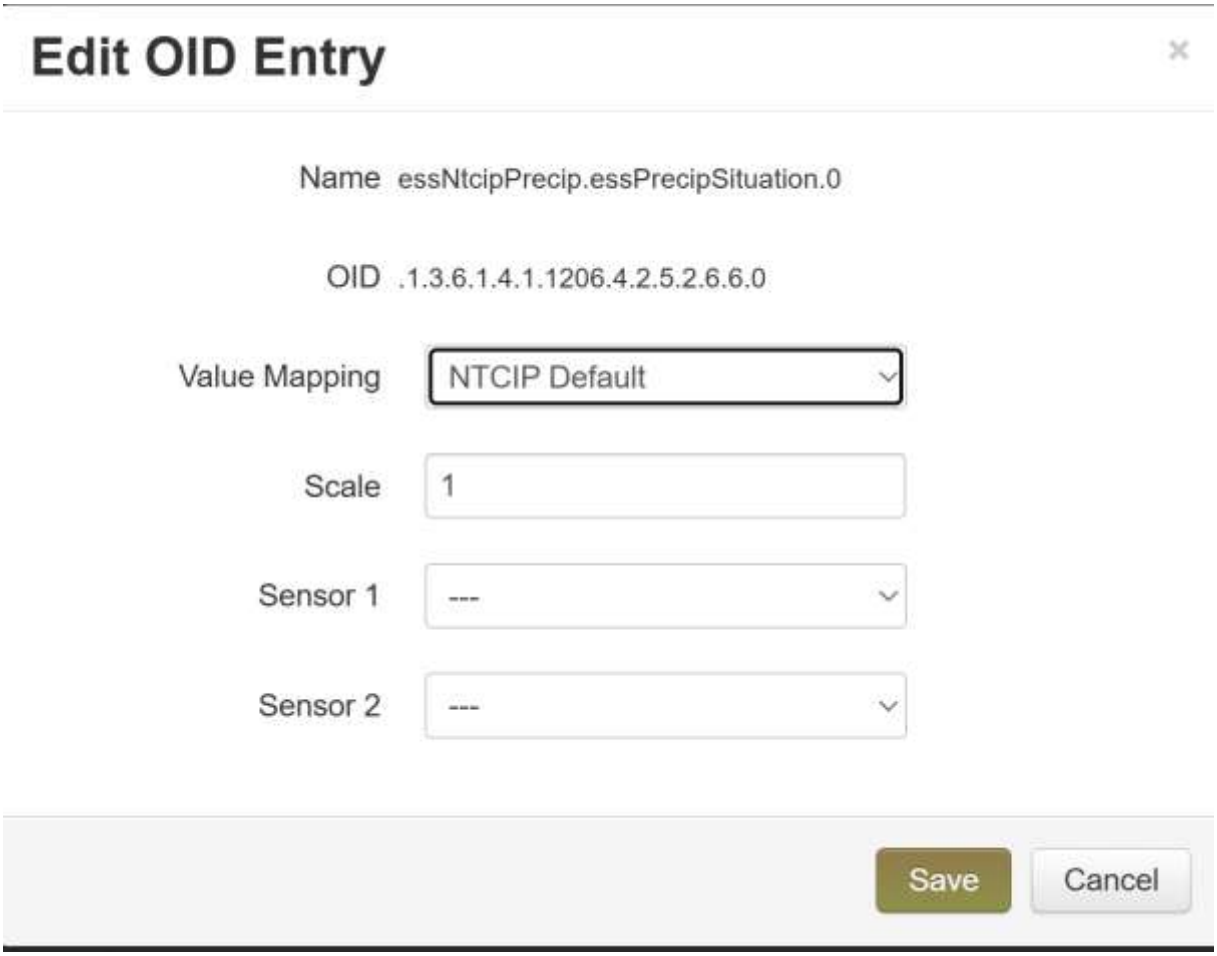
Sensor1: the input sensor for this OID

Depending on the OID, there might also be a Sensor 2 and Sensor 3 input channel configuration



The configured sensor channels can be selected from the select box.

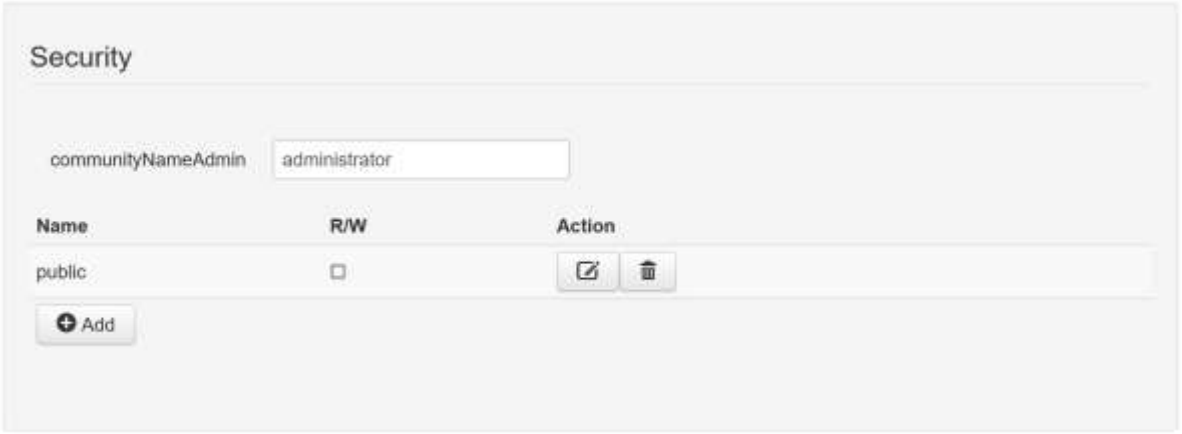
For some OID (e.g. essPrecipSituation) a default value mapping from UMB encoded values to NTCIP encoded values is pre-configured



This can be overridden by selecting a different value mapping from the select box.

8.4 Security

Here, the community names for SNMP access to the device can be configured:



As per NTCIP standard, at least an "administrator" community name (with read/write access) and one "restricted" community name (with read only access) should be configured.

Additional community names and the access type can be configured

8.5 Sensor Table Meta Data

Here, the meta data for various NTCIP ESS sensor tables can be configured. By configuring the respective sensor table entries, the number of entries in the respective table is also set

Sensor Table Metadata

windSensorTable

Index	Location	Action

Add

temperatureSensorTable

Index	Location	Action

Add

waterlevelSensorTable

Index	Location	Action

Add

pavementSensorTable

Index	Location	Action

Add

subsurfaceSensorTable

Index	Location	Action

Add

precipitationSensorTable

Index	Location	Action

Add

humiditySensorTable

Index	Location	Action

Add

radiationSensorTable

Index	Location	Action

Add

pressureSensorTable

Index	Location	Action

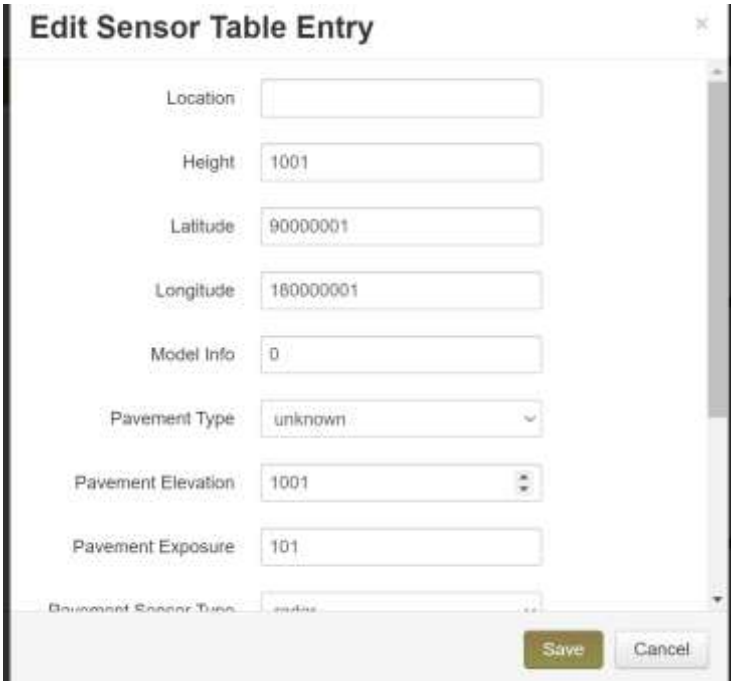
Add

airQualitySensorTable

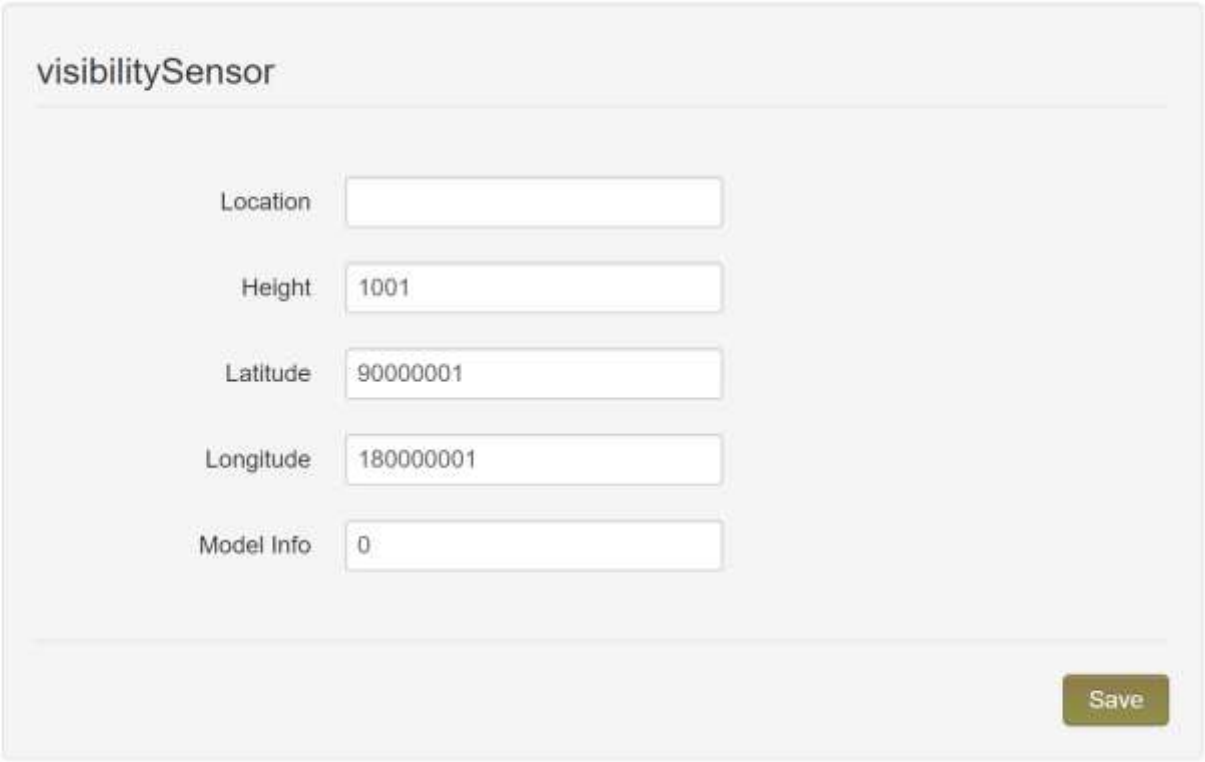
Index	Location	Action

Add

Sensor table entry details depend on the sensor table type – there is a common set of meta data like location information for all sensor table entries, but some entry types contain more attributes. For details, please refer to the NTCIP ESS 1204 documentation/MIB file



8.6 visibilitySensor



Here the meta data for the visibility sensor is configured.

8.7 Private MIB Parameter

Here, the parameters controlling the entries in the “Private MIB” are configured – specifically the number of entries in various sensor tables and the meta data for some of the tables.

Private MIB Parameter

General Settings

Sensor Tables

Number of Battery Status Entries	0
Number of Door Status Entries	0
Number of Radar Rain Sensor Entries	0
Number of All In One (WSx) Sensor Entries	0
Number of V Spectro Entries	0
Number of HSE IceSight Sensor Entries	0

activeRoadSensorTable

Index	Location	Action

Add

nonInvasiveRoadSensorTable

Index	Location	Action

Add

passiveRoadSensorTable

Index	Location	Action

Add

subSurfaceRoadSensorTable

Index	Location	Action

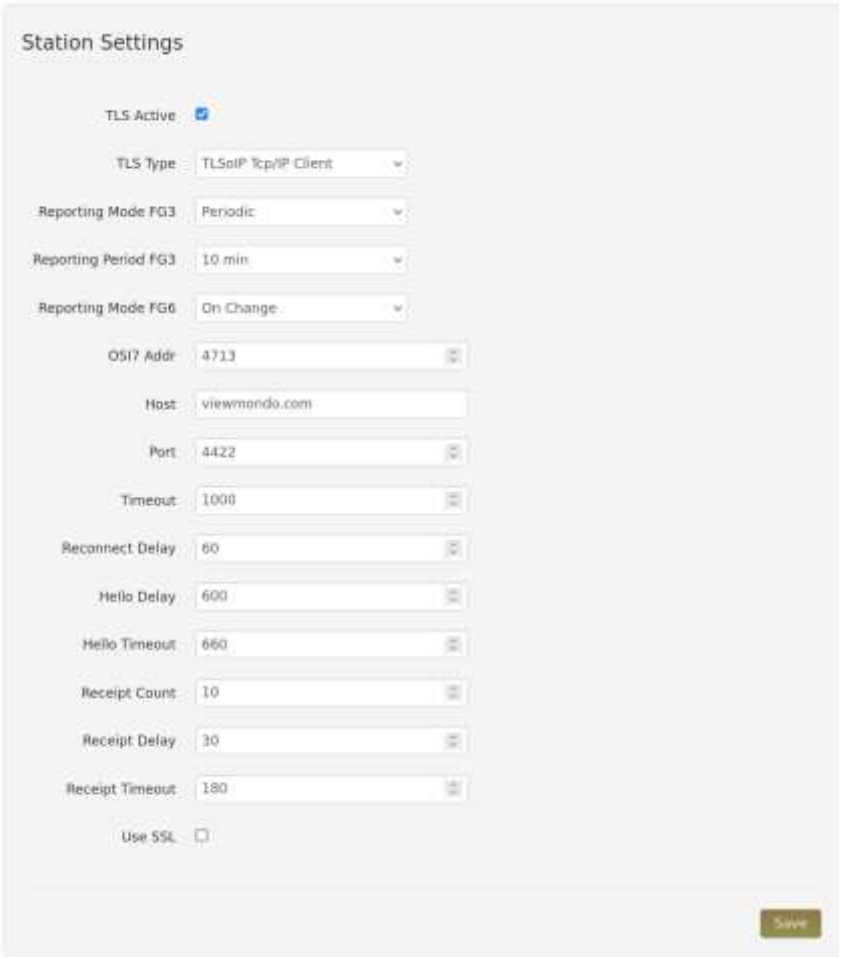
Add

Save

9 TLS (Technische Lieferbedingungen für Streckenstationen) Protocol (Germany)

If included with the license, the TLS protocol can be activated in parallel to the "Uplink" protocol (ViewMondo API/UMB Direct Access), and is configured on a separate configuration screen.

Here various parameters for TLS can be configured:



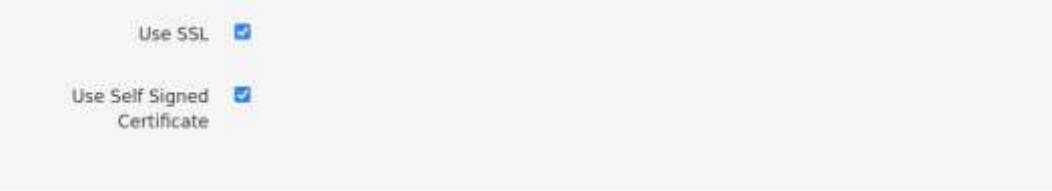
- TLS Active: TLS protocol driver active/inactive

- TLS Type: TLSoIP Tcp/IP Client (station = TLS slave connects to server = TLS master) or : TLSoIP Tcp/IP Server (server = TLS master connects to station = TLS slave)
- Reporting Mode FG3: "On Request" or "Periodic"
- Reporting Period FG3: the period in which FG3 (measure) values are transmitted
- Reporting Mode FG6: "On Request", "Periodic" or "On Change"
- Reporting Period FG6: the period in which FG6 (status) values are transmitted, if FG6 reporting mode is "periodic"
- OSI7 Addr: the OSI7 address for this station
- Host: for TLSoIP Tcp/IP Client – the server (master) IP address or DNS name to connect to
- Timeout: communication timeout
- Reconnect Delay: delay (in seconds) between connection attempts
- Hello Delay: delay (in seconds) between "keep alive" messages (if no data was sent in the meantime)
- Hello Timeout: timeout (in seconds) for receiving data or keep alive messages
- Receipt Count: count for data messages after which a receipt (ACK) is sent
- Receipt Timeout: timeout (in seconds) after receiving a data message after which a receipt (ACK) is sent
- Use SSL: use SSL encrypted communication

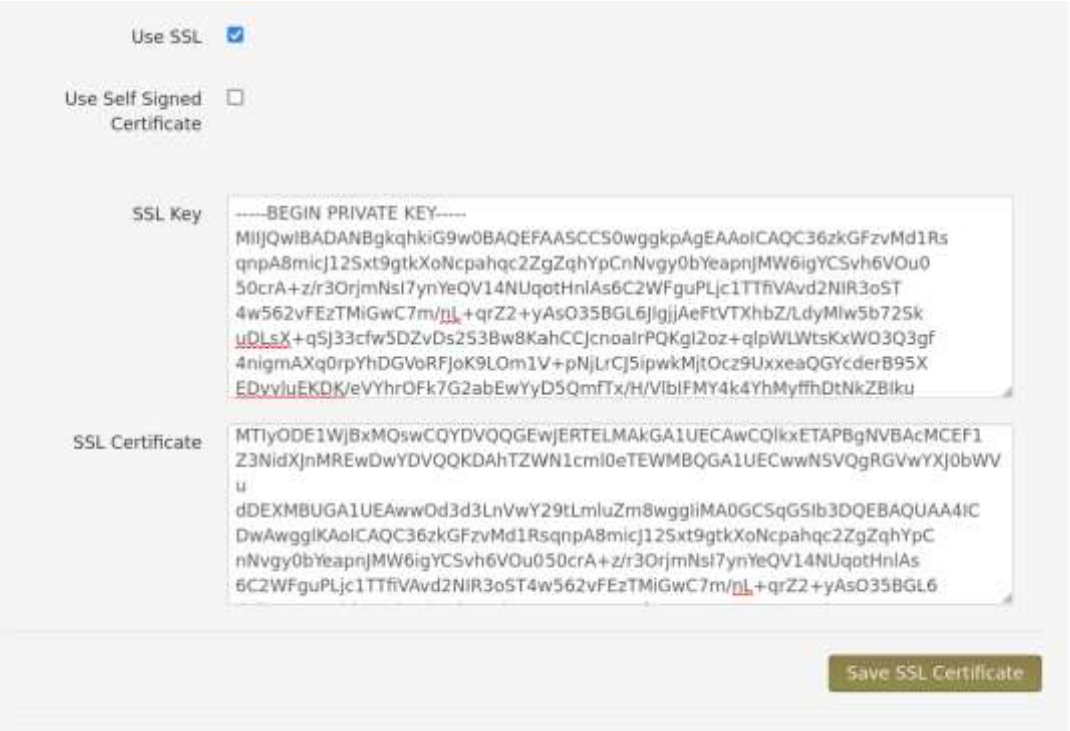
9.1 SSL Certificate (TCP/IP Server)

For TCP/IP Server, a TLS certificate needs to be used.

This can either be a "self signed" certificate (which is pre-generated and stored on the device), or a certificate supplied by the user



If the "self signed" certificate is not used, the certificate data (key and certificate data) needs to be supplied:

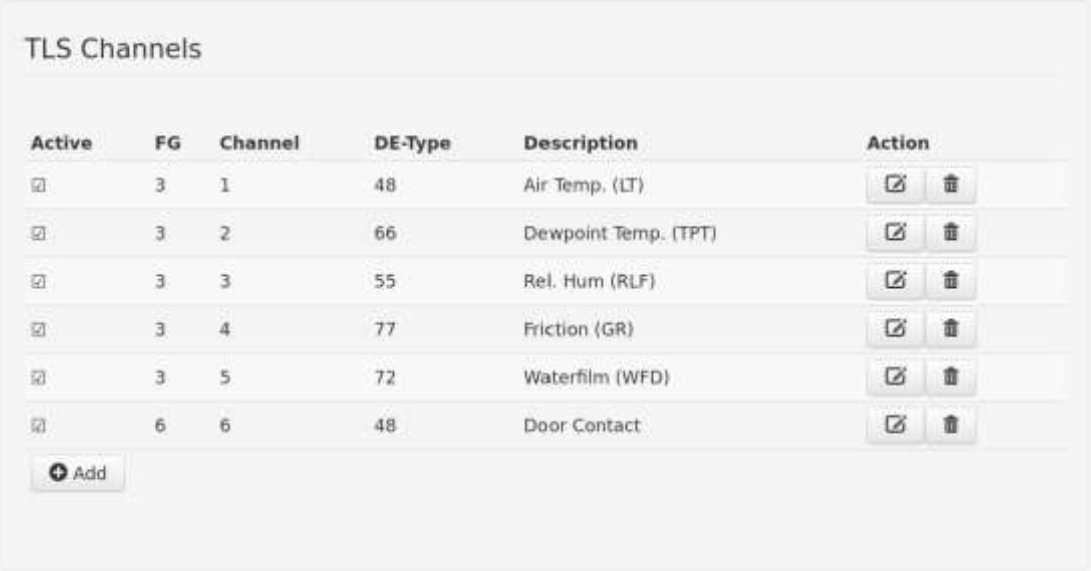


Note: the input fields for SSL key and SSL Certificate can be enlarged to view the complete data.

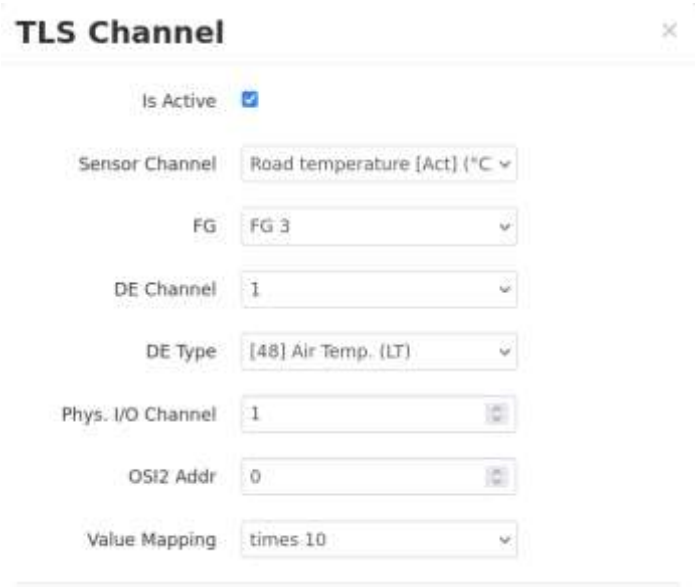
Click “Save SSL Certificate” to upload the certificate data.

9.2 TLS Channels

TLS channels are configured here

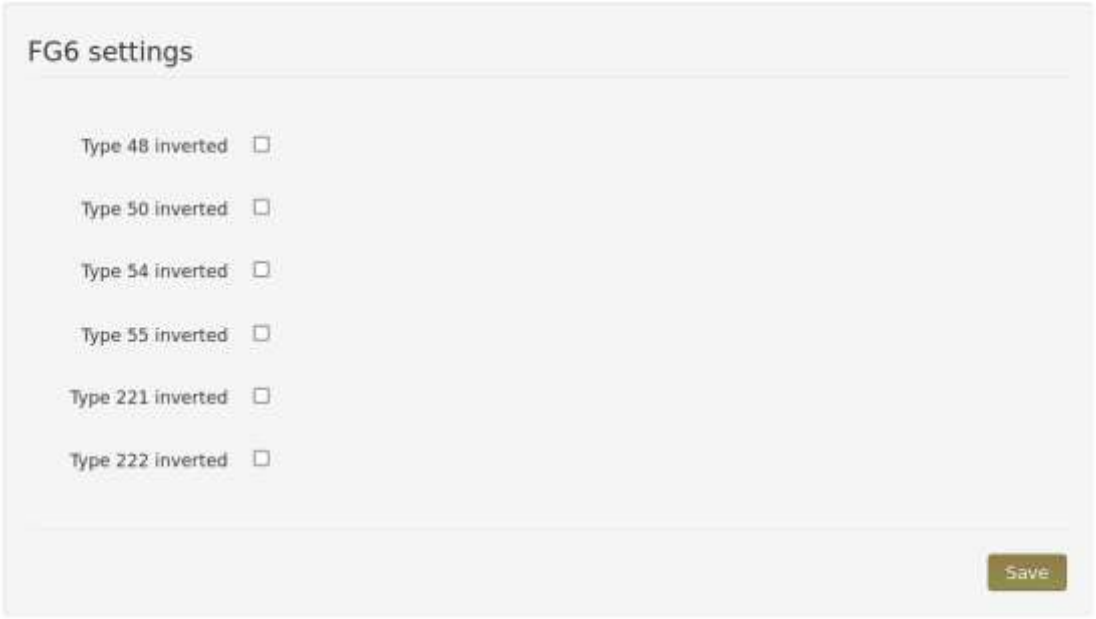


TLS channels are configured by assigning an sensor channel as input, and configuring the appropriate TLS channel settings:



- FG: FG 3 (measure values) or FG6 (status information)
- DE-Channel: the (within the FG unique) DE channel
- DE Type: the sensor type (select from dropdown list)
- Phys. I/O Channel: the physical IO channel for TLS (note: not important for TLSoIP but supplied in case we might support TC57)
- OSI2 Addr: the OSI2 Address for the TLS channel (note: not important for TLSoIP but supplied in case we might support TC57)
- Value Mapping: if needed, the sensor value can be mapped/scaled to match the TLS specifications. Note: for UMB sensor devices, the TLS encoded sensor channels supplied by the device already supply the values TLS encoded

9.3 FG6 Settings

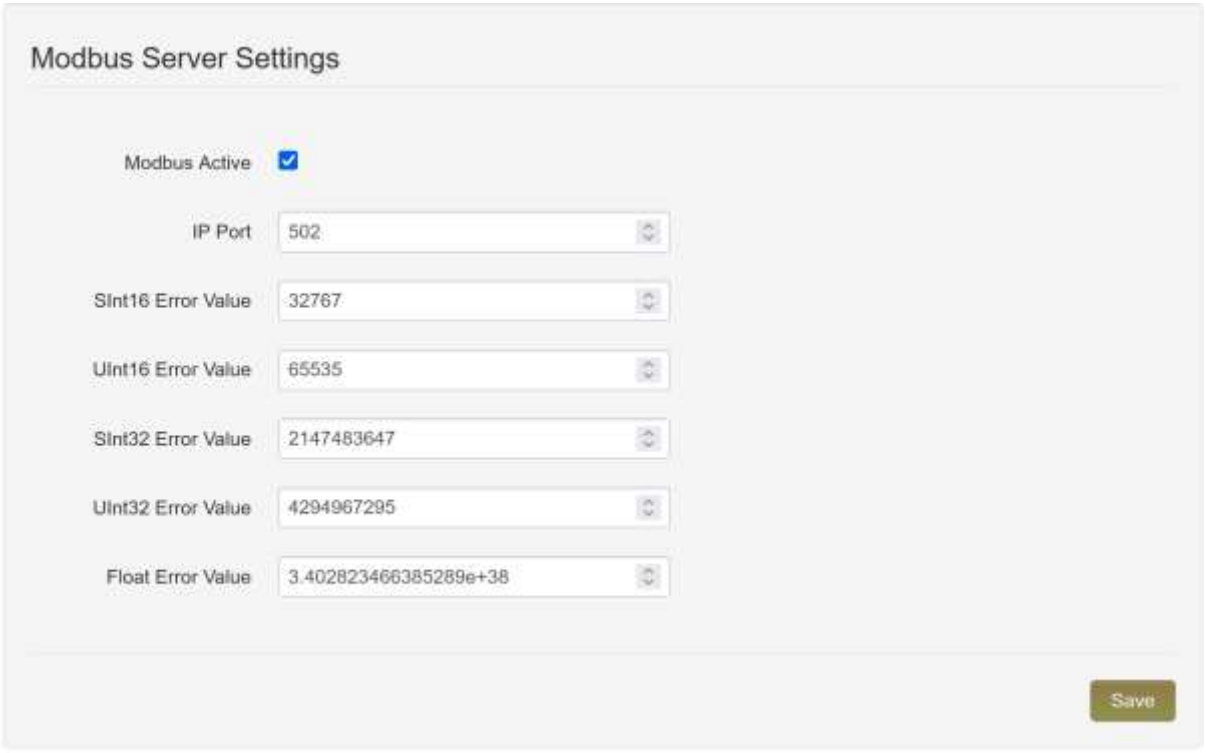


Here the status for some of the FG6 status values can be inverted

10 Modbus Server

If included in the license, measure values can be presented as Modbus input register on a Modbus Server interface.

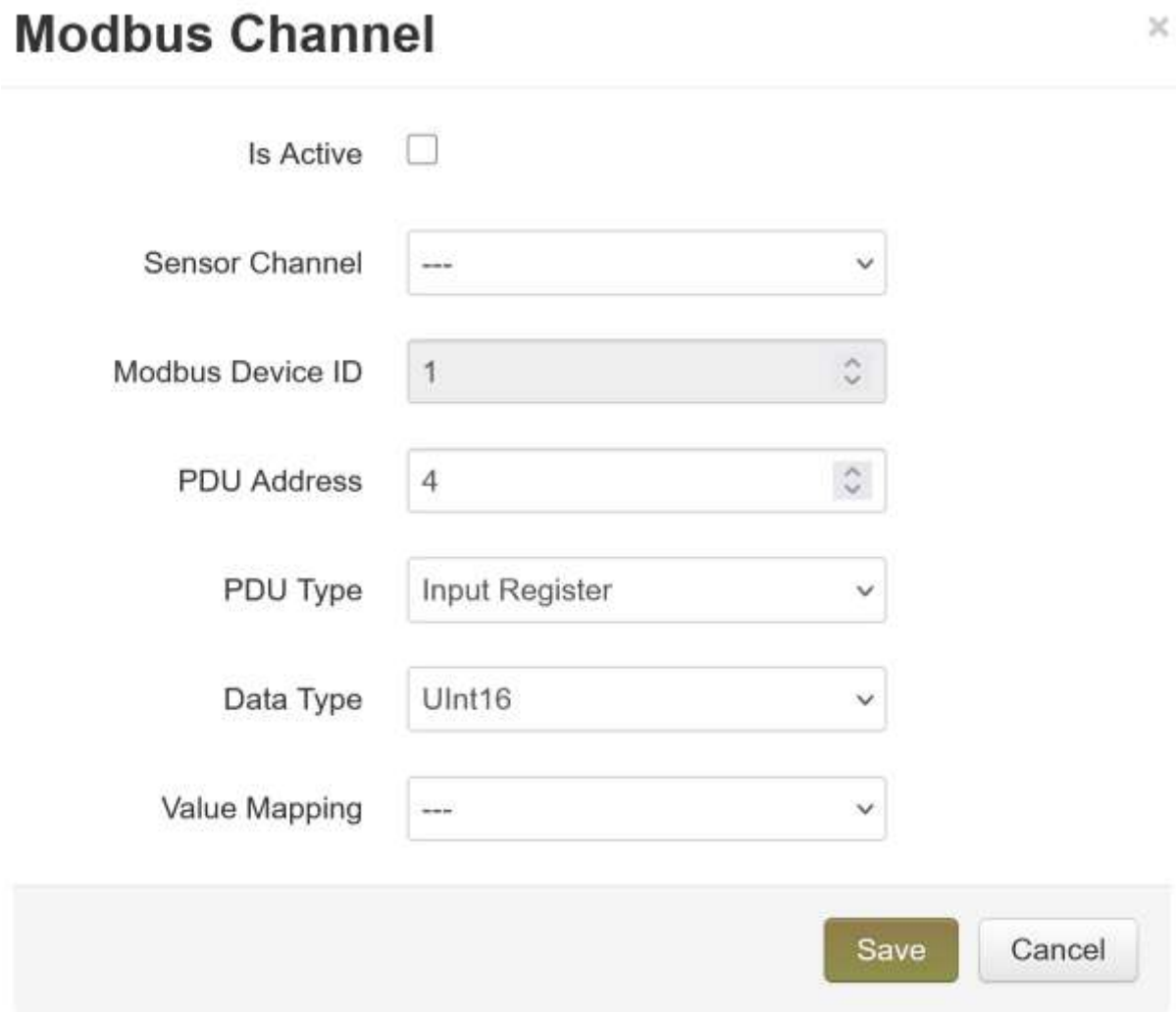
10.1 Modbus Server Settings



- Modbus Active: Modbus server is active
- IP Port: the IP Port on which the Modbus server is active
- Sint16 Error Value: value reported for signed int16 pdu types if the respective sensor channel reports an error or the value can not be obtained from the sensor channel
- UInt16 Error Value: value reported for unsigned int16 pdu types if the respective sensor channel reports an error or the value can not be obtained from the sensor channel
- Sint32 Error Value: value reported for signed int32 pdu types if the respective sensor channel reports an error or the value can not be obtained from the sensor channel
- Uint32 Error Value: value reported for unsigned int32 pdu types if the respective sensor channel reports an error or the value can not be obtained from the sensor channel
- Float Error Value: value reported for Float32 pdu types if the respective sensor channel reports an error or the value can not be obtained from the sensor channel

10.2 Modbus Channel

Following parameters can be configured to represent a sensor value via Modbus RPU:



- Is Active: the channel is active
- Sensor Channel: the sensor channel to be used
- PDU Address: the PDU Address
- PDU Type: the Modbus PDU type.
- Data Type: the data type.
- Value Mapping: value mapping to be applied to the sensor value

Note: for Float32, the byte order can also be configured:

11 Appendix

11.1 Third party software components and credits

Following is a list of third party libraries used in the UpCom software. The components are all based on the SDK and repository provided by Option/OpenWRT.

Current SDK Version is: 2.98.9

Note: for license and copyright statements regarding the Option Cloudgate operating system and SDK, please refer to the relevant documentation of the manufacturer

Library	**Credentials / Copyright notice**
Open62541	https://www.open62541.org/ Mozilla Public License https://www.mozilla.org/en-US/MPL/2.0/
Json-c	MIT Licence https://github.com/mesonbuild/json-c/blob/master/LICENSE.build
Curl	https://curl.se/ https://github.com/curl/curl/blob/master/README https://github.com/curl/curl/blob/master/COPYING
Net-snmp	http://www.net-snmp.org/about/license.html CMU/UCD copyright notice: (BSD like)
Libuuid	https://sourceforge.net/projects/libuuid/ https://sourceforge.net/p/libuuid/code/ci/master/tree/COPYING Modified BSD License
Sqlite3	https://www.sqlite.org/copyright.html Public Domain
Libssl	This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit (http://www.openssl.org/) https://github.com/jmhodges/libssl/blob/master/src/LICENSE
Libcrypto	https://github.com/jrgleason/libcrypto/blob/master/LICENSE GNU GPL
Python	https://www.python.org/download/releases/3.4.0/license/ PSF
Libmodbus	ProModbus commercial license https://libmodbus.org/promodbus-license/

12 Change History

October 2022	P. Rau	Initial Release 1.0.0
December 2022	P. Rau	Version 1.0.1 • OPC-UA datapoints “float” instead of “double” • Bug fix config dialog not working if too many sensor channels configured

January 2023	P. Rau	Version 1.0.2 • Improved initialization for OPCUA-Datapoints (to error status)
February 2023	P. Rau	Version 1.1.0 • ViewMondo API • Support for multiple Cameras • Value Mapping
April 2023	P. Rau	Version 1.1.1 • Bug fix handling config change via ViewMondo API • Fix reading value mapping list via Json
June 2023	P. Rau	Version 1.1.2 • Bug Fix GPS driver • Remove tabs from cam picture url
September 2023	P. Rau	Version 1.2.0 • Power Saving mode for UMB sensors and camera • Improved "sensor config readout"
November 2023	P. Rau	Version 1.2.1 • Support for CloudGate Nano and Cloudgate Mini • "UMB Direct Access" mode
January 2024	P. Rau	Version 1.3.0 • CSV Export • RS485 or RS232 interface for connection to UMB sensors
February 2024	P. Rau	Version 1.4.0 • SDI-12 sensors
March 2024	P. Rau	Version 1.4.1 • Updated SDK Version 2.98.5 • Optional Power Supply Voltage monitoring
March 2024	P. Rau	Version 1.5.0 • Use Cloudgate Mini "GPIO" ports for sensor/cam power saving • Fix "inverse logic" for power saving
March 2024	P. Rau	Version 1.5.1 • Optional GPIO "open drain" output mode
April 2024	P. Rau	Version 1.5.2 • Improved handling "inverse logic" for cam power control

		• power on sensors if sensor power control is active (in ViewMondo API mode) and uplink is set to "UMB Access via TCP/IP"
April 2024	P. Rau	Version 1.5.3 • bug fix combined power control sensors/cam with inverse settings
April 2024	P. Rau	Version 1.5.4 • SDK Version 2.98.7 • Fix switching between "UMB Access via TCP/IP" and "other uplink"
June 2024	P. Rau	Version 2.0.0 • FIX GPS handling and "GPS Enabled" vs "Required" • Cleanup javascript • Status page wirthout needing to logon (<cloudgate-ip>/UpCom) • Fix "poll time" if newly configured sensor channels do not respond • NTCIP Protocol
August 2024	P. Rau	Version 2.1.0 • Improved plugin startup – waiting for clock to be set in device/cam driver instead of general startup procedure • Calc Channels • New Parameter for OPC-UA – use old value mapping
September 2024	P. Rau	Version 2.2.0 • Alarm Processing
Oktober 2024	P. Rau	Version 2.3.0 • Backup/Restore Configuration • Aggregate Values for API Uplink/Export/Data Storage • Add default UMB device channels • Support SD-Card on CloudGate Mini/Nano
November 2024	P. Rau	Version 2.3.1 • Fix: UMB or SDI-12 Sensor Config on Mini/Nano
March 2025	P. Rau	Version 2.4.0 • TLS Protocol (Germany) • Improved status display • general improvements / memory usage • Fix: handle "NaN" value from UMB device (-> map to Error 0x55)

April 2025	P. Rau	Version 2.4.1 • SDK Version 2.98.9 • Fix "scale" input for NTCIP OID • Fix default scale for some NTCIP OIDs
April 2025	P. Rau	Version 2.4.2 • Fix SNMPD Startup
May 2025	P. Rau	Version 2.4.3 • Show sensor error description on status • Don't show UMB error code 0x55 as sensor error (show N/A value instead) • Improved status display • Poll sensors before clock is set • Set clock from browser time • Set clock via ViewMondo API
October 2025	P. Rau	Version 2.4.4 • Improved SDI-12 error handling • Improved error handling in value aggregator • Improved thread safety for value aggregator and app params • Fixed bug with quotes (and other "bad" characters) in camera config
Ooctober 2025	P. Rau	Version 2.5.0 • Download stored data via user interface • Bug fix CSV – decimal point • Buf fix "usleep"
October 2025	P. Rau	Version 2.5.1 • Bug Fix ValueAggregator • Monitor uplink with reset • Param to set clock via API
October 2025	P. Rau	Version 2.5.2 • Inactivity Reset Interval for API • Improved API clock sync (millisecond resolution, adjtime) • Improved locking on data storage (longer timeout)
November 2025	P. Rau	Version 3.0.0 • Bug Fix Cam Picture Queue Files • Disabled internal reload of sensor channels after config change because of problems in calc channel. User needs to restart plugin after configuration changes. • Calc channels type "Code" -> Python 2.7 scripting integration

		• Cam Picture Queue optional on SD Card (non volatile) • CSV Export Queue optional on SD Card (non volatile) • Optimizations
November 2025	P. Rau	Version 3.0.1 • Restart plugin after uploading backup • OttHydromet Logo
December 2025	P. Rau	Version 3.1.0 • New tags [utctime] and [serial] for export filenames • Improved camera grabbing using native libcurl calls • Improved camera status/error information • (s)FTP(s) push camera pictures • Optional keep export and camera queues on SD card (non volatile) instead of ram drive • Archive camera pictures and Export files on SD card • Alarm condition for camera pictures and export jobs (grab pictures/export data only if alarm condition is set) • Handle "Housekeeping" (database, archived files) in own thread • Handle FTP Transfers in own thread • Handle measure value database write operations in own thread
December 2025	P. Rau	Version 3.1.1 • Bug fix: allow value mapping etc. if no "primary" Uplink Protocol is selected
December 2025	P. Rau	Version 3.1.2 • SDI-12 – use "C" or "M" command • Serial Access via TCP/IP – select port for "WW Rev4 with industrial serial card".
December 2025	P. Rau	Version 3.1.3 • Bug fix activity monitoring
January 2026	P. Rau	Version 3.2.0 • Improved SDI-12 error handling • N(D)DOT Gate Controller (see separate documentation)
January 2026	P. Rau	Version 3.2.1 • Additional parameter for SDI-12 (delay between commands)

January 2026	P. Rau	Version 3.2.2 • Improved error handling for Gate Controller
February 2026	P. Rau	Version 3.2.3 • Improved error handling for file transfers (cam/export)
February 2026	P. Rau	Version 3.2.4 • Immediately close any additional client connections on TLSoIP Server and gate controller
February 2026	P. Rau	Version 3.3.0 • Support “TLSDump” and “TLSDump Micks” data export
February 2026	P. Rau	Version 3.3.1 • Bug fix NTCIP “SubSurfaceSensorError” if “skip index 6” is not active • Layout for “Edit OID Assignment” improved • Download (active) NTCIP OID Sensor assignment
April 2026	P. Rau	Version 3.4.0 • Support Modbus Sensors via RS485 • Configurable serial interface parameter • TLS FG6 DE Type 225 (Battery Status) • Modbus Server (Uplink)