MBUS-GSLE
Data concentrator and M-Bus hub

Version: 1.1
Date: 21 November 2024

Firmware Version 1.36

Authors:
Remo Reichel, Frank Richter
solvimus GmbH
Ratsteichstr. 5
98693 Ilmenau
Germany

Page intentionally left blank

Table of Contents

1 Notes and conventions

1.1 About this document

This manual provides guidance and procedures for a fast and efficient installation and start-up of the units described in this manual. It is imperative to read and carefully follow the safety guidelines.

1.2 Legal basis

1.2.1 Placing on the market

Manufacturer of the MBUS-GSLE is the solvimus GmbH, Ratsteichstraße 5, 98693 Ilmenau, Germany.

1.2.2 Copyright protection

1.2.3 Personnel qualification

The product use described in this documentation is intended exclusively for qualified electricians or persons instructed by these. They must all have good knowledge in the following areas:

- Applicable standards
- Use of electronic devices

1.2.4 Intended use

If necessary, the components or assemblies are delivered ex works with a fixed hardware and software configuration for the respective application. Modifications are only permitted within the scope of the possibilities shown in the documentation. All other changes to the hardware or software as well as the non-intended use of the components result in the exclusion of liability on the part of solvimus GmbH. Please send any requests for a modified or new hardware or software configuration to solvimus GmbH.

1.2.5 Exclusion of liability

Study this manual and all instructions thoroughly prior to the first use of this product and respect all safety warnings, even if you are familiar with handling and operating electronic devices.

The solvimus GmbH accepts no liability for damage to objects and persons caused by erroneous operation, inappropriate handling, improper or non-intended use or disregard for this manual, especially the safety guidelines, and any warranty is void.

1.2.6 Disclaimer

All products, company names, trademarks and brands are the property of their respective holders. Their use serves only to describe and identify the respective company, product or service. Use of them does not imply any affiliation with, commercial relationship with or endorsement by them.

1.3 Symbols

- Danger: It is essential to observe this information in order to protect persons from injury.
- Caution: It is essential to observe this information in order to prevent damage to the device.
- Notice: Boundary conditions that must always be observed to ensure smooth and efficient operation.
- ESD (Electrostatic Discharge): Warning of danger to components due to electrostatic discharge. Observe precautionary measures when handling components at risk of electrostatic discharge.
- Note: Routines or advice for efficient equipment use.
- Further information: References to additional literature, manuals, data sheets and internet pages.

1.4 Font conventions

Names of paths and files are marked in italics. According to the system the notation is using slash or backslash.
e. g.: *D: \Data*

Menu items or tabs are marked in bold italics.
e. g.: ***Save***

An arrow between two menu items or tabs indicates the selection of a sub-menu item from a menu or a navigation process in the web browser.
e. g.: ***File*** → ***New***

Buttons and input fields are shown in bold letters.
e. g.: **Input**

Key labels are enclosed in angle brackets and shown in bold with capital letters.
e. g.: ⟨**F5**⟩

Programme codes are printed in Courier font.
e. g.: `ENDVAR`

Variable names, identifiers and parameter entries are marked in italics.
e. g.: *Value*

1.5 Number notation

Numbers a noted according to this table:

Numbering system	Example	Comments
Decimal	100	Normal notation
Hexadecimal	0x64	C-like notation
Binary	'100' '0110.0100'	In apostrophes Nibbles separated by dots

Table 1: Numbering systems

1.6 Safety guidelines

- Observe the recognized rules of technology and the legal requirements, standards and norms, and other recommendations.
- Study the instructions for the extinction of fire in electrical installations.
- The power supply must be switched off before replacing components and modules.

If the contacts are deformed, the affected module or connector must be replaced, as the function is not guaranteed in the long term.

The components are not resistant to substances that have creeping and insulating properties. These include e.g. aerosols, silicones, triglycerides (ingredient of some hand creams). If the presence of these substances in the vicinity of the components cannot be excluded, additional measures must be taken:

- Install the components in an appropriate casing.
- Handle components with clean tools and materials only.

- Only use a soft, wet cloth for cleaning. Soapy water is allowed. Pay attention to ESD.
- Do not use solvents like alcohol, acetone etc. for cleaning.
- Do not use a contact spray, because in an extreme case the function of the contact point is impaired and may lead to short circuits.
- Assemblies, especially OEM modules, are designed for installation in electronic housings. Do not touch the assembly when it is live. In each case, the valid standards and directives applicable to the construction of control cabinets must be observed.
- The components are populated with electronic parts which can be destroyed by an electrostatic discharge. When handling the components, ensure that everything in the vicinity is well earthed (personnel, workplace and packaging). Do not touch electrically conductive components, e.g. data contacts.

1.7 Scope

This documentation describes the device manufactured by solvimus GmbH, Ilmenau, and stated on the title page.

1.8 Abbreviations

Abbreviation	Meaning
2G	Mobile radio standard, synonym for GSM or GPRS
3G	Mobile radio standard, synonym for UMTS
4G	Mobile radio standard, synonym for LTE
ACK	Acknowledge
AES	Advanced Encryption Standard
AFL	Authentication and Fragmentation Layer
AI	Analog Input
ANSI	American National Standards Institute
AO	Analog Output
APN	Access Point Name
ASCII	American Standard Code for Information Interchange
ASHRAE	American Society of Heating, Refrigerating and Air-Conditioning Engineers
BACnet	Building Automation and Control networks
BBMD	BACnet Broadcast Management Device
BCD	Binary-coded decimal numbers
BDT	Broadcast Distribution Table
BMS	Building Management System
CA	Certification Authority
CHAP	Challenge Handshake Authentication Protocol
CI	Control Information
CLI	Command line interface
COSEM	COmpanion Specification for Energy Metering
CPU	Central processing unit
CRC	Cyclic redundancy check
CSV	Character-Separated Values

Continued on next page

Table 2 – *Continued from previous page*

Abbreviation	Meaning
CTS	Clear to send
D0	D0 interface (optical interface, IEC 62056-21)
DDC	Direct Digital Control
DHCP	Dynamic Host Configuration Protocol
DI	Digital Input, digital input terminal
DIF	Data information field
DIFE	Data information field extensions
DIN	Deutsches Institut für Normung, German Institute for Standardization
DLDE	Direct Local Data Exchange (EN 62056-21, IEC 1107)
DLDERS	DLDE communication via RS-232 or RS-485
DLMS	Device Language Message Specification
DNS	Domain Name System
DO	Digital Output, digital output terminal
EEG	German Renewable Energy Sources Act
EIA/TIA	Electronic Industries Alliance/Telecommunications Industry Association
ELL	Extended Link Layer
EMC	Electromagnetic compatibility
EN	European norm
ESD	Electrostatic Discharge
FCB	Frame Count Bit
FCV	Frame Count Valid Bit
FNN	Forum Netztechnik/Netzbetrieb, subgroup of VDE
FSK	Frequency Shift Keying
FTP	File Transfer Protocol
FTPS	FTP via TLS
GB	Gigabyte
GMT	Greenwich Mean Time
GPRS	General Packet Radio Service
GSM	Global System for Mobile Communications
HCA	Heat cost allocator
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
I2C	Inter-Integrated Circuit
I/O	Input/Output
ICCID	Integrated Circuit Card Identifier
ICMP	Internet Control Message Protocol
ID	Identification, Identifier, unique marking
IEC	International Electrotechnical Commission
IEEE	Institute of Electrical and Electronics Engineers
IoT	Internet of Things
IP	Internet Protocol or IP address
ISO	International Organization for Standardization
JSON	JavaScript Object Notation
LAN	Local area network
LCD	Liquid-crystal display
LED	Light-Emitting Diode
LSB	Least significant byte
LSW	Least significant word
LTE	Long Term Evolution
M2M	Machine-to-Machine
M-Bus	Meter-Bus (EN 13757, part 2, 3 and 7)
MAC	Medium Access Control or MAC-Adresse
MB	Megabyte
MCR	Multi Channel Reporting
MCS	Modulation and Coding Scheme
MDM	Meter Data Management
MEI	Modbus Encapsulated Interface
MHz	Megahertz
MQTT	Message Queuing Telemetry Transport
MSB	Most Significant Byte
MSW	Most Significant Word
MUC	Multi Utility Communication, MUC controller
NB-IoT	Narrow Band Internet of Things
OBIS	Object Identification System
OEM	Original Equipment Manufacturer
OMS	Open Metering System
PAP	Password Authentication Protocol
PEM	Privacy Enhanced Mail
PID	Product ID
PIN	Personal Identification Number
PKI	Public Key Infrastructure

Continued on next page

Table 2 – *Continued from previous page*

Abbreviation	Meaning
PLC	Programmable Logic Controller
PLMN	Public Land Mobile Network
PPP	Point-to-Point Protocol
PPPoE	Point-to-Point Protocol over Ethernet
PTC	Polymer with positive temperature coefficient
PUK	Personal Unblocking Key
RAM	Random Access Memory
REQ_UD	Request User Data (Class 1 or 2)
RFC	Requests For Comments
RSP_UD	Respond User Data
RSRP	Reference Signal Received Power
RSRQ	Reference Signal Received Quality
RSSI	Received Signal Strength Indicator
RTC	Real-Time Clock
RTOS	Real-Time Operating System
RTS	Request to send
RTU	Remote Terminal Unit
S0	S0 interface (pulse interface, EN 62053-31)
SCADA	Supervisory Control and Data Acquisition
SCP	Secure Copy
SFTP	SSH File Transfer Protocol
SIM	Subscriber Identity Module
SML	Smart Message Language
SMTP	Simple Mail Transfer Protocol
SND_NKE	Send Link Reset
SND_UD	Send User Data to slave
SNTP	Simple Network Time Protocol
SPST	Single Pole Single Throw Relay (closing switch)
SRD	Short Range Device
SSH	Secure Shell
SSID	Service Set Identifier
SSL	Secure Sockets Layer
TCP	Transmission Control Protocol
THT	Through-Hole Technology
TLS	Transport Layer Security
U	Unit width of the housing (1 U = 18 mm)
UART	Universal Asynchronous Receiver Transmitter
UDP	User Datagram Protocol
UL	Unit load for M-Bus
UMTS	Universal Mobile Telecommunications System
UTC	Universal Time Coordinated
VCP	Virtual COM port
VDE	Verband der Elektrotechnik Elektronik Informationstechnik e.V., German Association for Electrical, Electronic & Information Technologies
VHF	Very high frequency
VID	Vendor ID
VIF	Value information field
VIFE	Value information field extensions
VLAN	Virtual Local Area Network
VPN	Virtual Private Network
WAN	Wide Area Network
WLAN	Wireless Local Area Network
wM-Bus	Wireless Meter-Bus (EN 13757, part 3, 4 and 7)
XML	eXtensible Markup Language
XSLT	eXtensible Stylesheet Language Transformation

Table 2: Abbreviations

2 Introducing the device

2.1 General information

The M-Bus (Meter-Bus) is an established and well-known interface for automated meter reading. Especially the ease of installation (simple two-wire system with powering by the bus) and the robustness are important features. These are also special attributes that are of interest for use in industrial environments.

The M-Bus is defined in the standard EN 13757. It establishes an own physics as well as an own protocol. For connecting it to other systems, a translation is necessary. In the field of building automation, Modbus TCP is one of the most common communication standards, and in the field of building automation it is BACnet IP.

The MBUS-GSLE links the world of the wired M-Bus with Modbus TCP, respectively with BACnet IP, permitting the transmission of meter data via Modbus TCP to a control system (host system, e.g. PLC, DDC and others) respectively via BACnet IP to a building management system (BMS).

The device supports operating 125, 250 respectively 500 unit loads (UL, mostly equivalent to the number of meters). A powerful protocol stack is implemented in the MBUS-GSLE for data handling compliant to the standard. All the meters available on the market can be read out and their data processed without further configuration. The MBUS-GSLE has an Ethernet communication interface. A downstream control system can thus request the meter data directly via a network connection.

When reading out meters via M-Bus, the data are always read out by a bus master. Thus, it disposes of all the data. Depending on the type of master, it may be difficult or even impossible to render the read data usable for another system. Physics rules out a second master in the bus, and additionally problems may arise when switching the bus by a relay.

The MBUS-GSLE was conceived for precisely this purpose. It can be integrated in existing M-Bus installations by simple insertion between the master and the slaves (meters). After the configuration of the MBUS-GSLE, meter data are read out, processed and transferred, optionally also stored by the integrated master. The data are transferred to the existing, original bus master via a slave interface. Thus, it receives as before the requested meter data.

The MBUS-GSLE comes in a housing 3 U (modules) wide and is intended for top hat rail mounting (DIN rail 35 mm).

The serial number of the devices of the solvimus GmbH can be read from the housing.

2.2 Delivery variants and scope of delivery

The MBUS-GSLE is offered in a range of variants, and so can easily be adapted to the requirements of the particular property.

Variant	Article number	M-Bus
MBUS-GSLE 125	500366	X (125 UL)
MBUS-GSLE 250	500371	X (250 UL)
MBUS-GSLE 500	500372	X (500 UL)

Table 3: Delivery variants

The scope of delivery contains the device and a Quick Start Guide.

2.3 Connectors

The various interfaces of the MBUS-GSLE are on different sides of the device.

The following figure shows the device. All variants are similar in outward appearance.

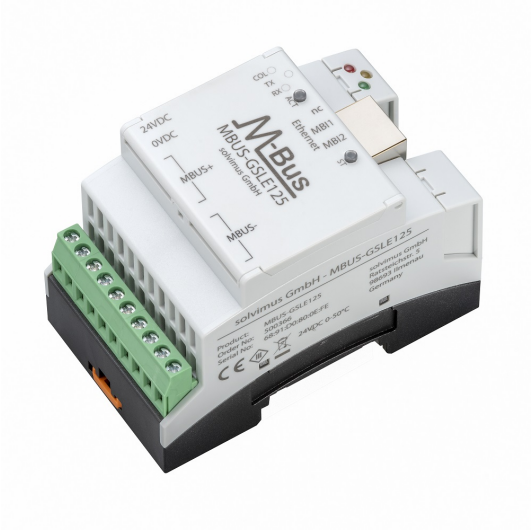


Figure 1: MBUS-GSLE

The following connectors are available at the MBUS-GSLE:

Connector	Designation	Pin assignment	Comments
Power supply	24 VDC, 0 VDC	24 VDC: positive power supply 0 VDC: negative power supply	12..36 VDC screw terminal cross section 2.5 mm^2
Ethernet connector	Ethernet	1: TX+ 2: TX- 3: RX+ 4: 5: 6: RX- 7: 8:	according to EIA/TIA 568A/B
M-Bus connector (meter)	MBUS+, MBUS-	MBUS+: positive bus line MBUS-: negative bus line	screw terminal cross section 2.5 mm^2 MBUS+ and MBUS- each joined internally
M-Bus connector (existing master)	MBI1, MBI2	MBI1: first bus line MBI2: second bus line	screw terminal cross section 2.5 mm^2

Table 4: Pin assignment

2.4 Status LEDs

The MBUS-GSLE is equipped with 5 status LEDs. These indicate the following states:

LED	Colour	Description
Active (ACT)	off orange (blinking) green (flashing)	Inactive, idle state Searching meters (scanning) meter reading
State (ST)	off green orange red	Software is not started Main programme is running Initialisation Error
COL	off red (blinking approx. 0.5 Hz)	Collision respectively too large capacitive load on the M-Bus Overload of the M-Bus
TX	yellow	Sending of data from the M-Bus master to the bus
RX	green	Reception in the M-Bus master of data from the M-Bus slaves

Table 5: Status LEDs

In the operating state, the *State-LED* is green and the *Active-LED* flashes green briefly during the readout.

2.5 First steps

2.5.1 Power supply

The MBUS-GSLE is supplied with an external voltage in the range 12-36 VDC (wide input voltage range). The MBUS-GSLE starts automatically after connection to the supply voltage.

By default, the following calls are made on system startup:

- Configuration of the network interface (Ethernet) via DHCP or static configuration
- One-time generation of SSL device keys (needs some time)
- Obtaining the system time via SNTP
- Start of system services
- Start of the main programme

The main programme then provides the entire functionality, including the web-based front end of the MBUS-GSLE.

2.5.2 Network configuration and first steps

The MBUS-GSLE can be completely configured via the network interface. Therefore, it has to be configured according to your network. If necessary, ask your administrator.

✔ The MBUS-GSLE is set by default to the static IP address 192.168.1.101 (subnet mask: 255.255.255.0, gateway: 192.168.1.254).

For intuitive operation, a configuration website is available on the device, which can be accessed via the IP address of the MBUS-GSLE called in a web browser.

➔ Website on the MBUS-GSLE, e.g.: http://192.168.1.101/

ℹ When handling multiple devices under same IP (e.g. commissioning) or with different software versions (e.g. update), you should always clear the cache of the browser (e.g. ⟨**CTRL+F5**⟩) to prevent an inconsistent display of the website.

The following page opens in the browser:

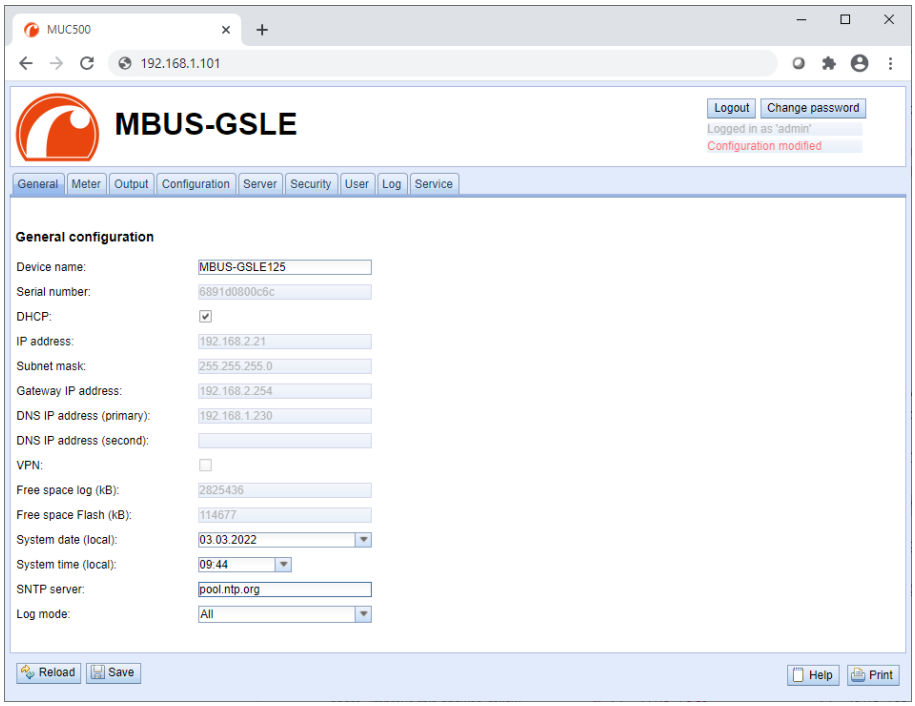


Figure 2: Website of the MBUS-GSLE

The web-based front end is described separately in Chapter 4. There you will find a detailed overview of the functionalities of the web-based front end.

In addition, access via SFTP, SCP, FTPS (file transfer) or via SSH (console) is also possible by default (see Chapter 3):

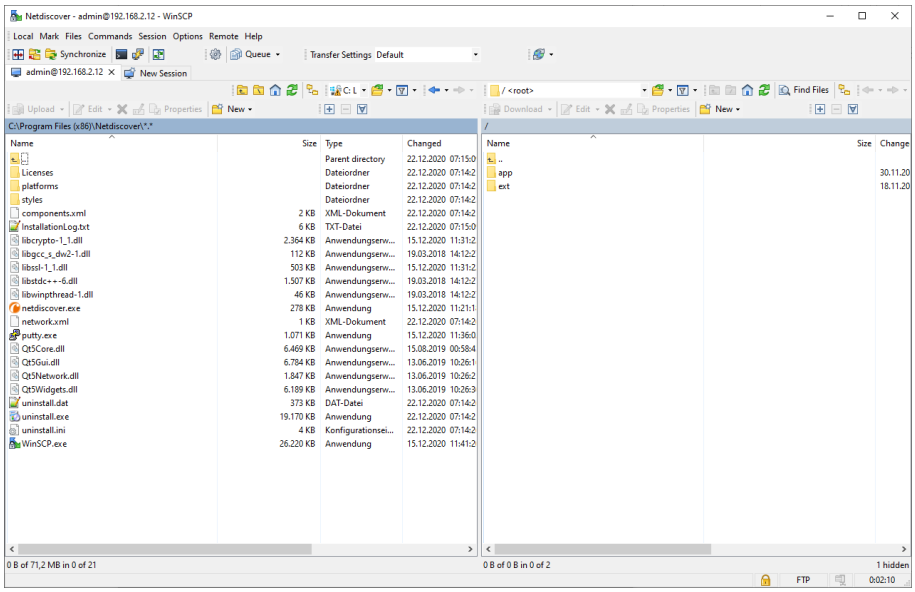


Figure 3: WinSCP main window after establishing the connection

2.6 Specific troubleshooting

In case the MBUS-GSLE does not work as described in this document, it is useful to locate the malfunction in order to resolve the issue and to recover the full functionality again.

2.6.1 All LEDs remain off, the device does not respond.

⚠ Only trained and appropriately qualified personnel are allowed to check the electric power supply (see Section 1.2.3).

Switch off the power supply and remove the device. Disconnect all cables. Check the external power supply and the MBUS-GSLE under laboratory conditions, that means at an isolated and separate measurement installation. Switch on the power supply at that measurement installation. It must adhere to the requirements given in Section 2.8.2.

If the problem persists, ensure that there are no faults in the power supply caused by the infrastructure, circuit breakers or residual current devices.

If errors could not be eliminated, please contact our customer support:
E-Mail: support@solvimus.de
Phone: +49 3677 7613065

2.6.2 The Power LED is blinking green.

⚠ Only trained and appropriately qualified personnel are allowed to check the electric power supply (see Section 1.2.3).

Switch off the power supply. Remove all cables except the power supply. Now switch on the power supply and check whether the *Power-LED* is now permanently on.

Now reconnect all cables one by one and check after each step whether the *Power-LED* remains permanently lit.

If the error occurs when connecting a specific cable, proceed to check this one more thoroughly. The error may reside in the external wiring, e.g. a short circuit or overload. Replace the faulty cable if necessary. Check

the external power supply.

If errors could not be eliminated, please contact our customer support:
E-Mail: support@solvimus.de
Phone: +49 3677 7613065

2.7 Typical application scenarios

Below some examples are given how the MBUS-GSLE can be used.

For using the MBUS-GSLE, the network and meter interfaces must be parameterised according to your application and your facility (see Chapter 4).

2.7.1 Local application without control system

The MBUS-GSLE can be used for local meter reading.

There is no control system (host system) required to collect and store meter data. Remote communication can therefore be deactivated. Only the local storage of CSV files (see Section 8.10) has to be set up in the tab ***Server*** (see Section 4.8).

In this case, the MBUS-GSLE is accessed via a PC that is located in the same network. The current meter values can thus be monitored via the web-based front end in the tab ***Meter***. The CSV files can be accessed via FTP access, provided logging is active. In order to do this, connect to the MBUS-GSLE with an FTP client (see Section 8.7).

Users can be configured in the user management with the corresponding access rights to allow read access to the meter list (see Section 4.10).

2.7.2 Remote monitoring without control system

This use case is largely equivalent to the example in Section 2.7.1. The only difference is the network infrastructure that is set up between a PC and the MBUS-GSLE (Internet). The PC and the MBUS-GSLE are not located in a physical but in a logical network.

- ✔ As a rule, routers and firewalls must be parameterised here to allow access to the MBUS-GSLE in the internal network from an external network (PC in the Internet). Please ask your administrator about setting up routings, port forwarding, packet filters and firewalls for the individual services of the MBUS-GSLE, such as FTP, HTTP and SSH.

If the network is parameterised correctly, you can access the MBUS-GSLE in the same way as in the local application.

2.7.3 Remote monitoring with email dispatch

The MBUS-GSLE can send the meter data as emails to any email address. The meter data is stored e.g. in XML format and can be processed as required (see Section 8.8).

- ✔ In order to send emails, the internal network has to be set up correspondingly (z. B. firewall, router). Ask your administrator about this.

2.7.4 Remote monitoring with FTP upload

The MBUS-GSLE can also actively upload CSV data to an FTP server (see Section 8.7) instead of manually downloading this data by the user. This makes it possible to access and process the files automatically.

- ✔ For the FTP upload, on the one hand the internal network (e.g. firewall, router) and on the other hand the receiving FTP server must be correctly configured. Ask your administrator about this.

2.7.5 Remote monitoring with SFTP upload

The transfer of files to a server can also be secured via encrypted communication. For example, it is possible to encrypt the data using Secure Shell (SSH).

For using SFTP, so-called finger prints need to be provided on the device. For more information see Section 8.7.

Subsequently, an encrypted cyclic upload of meter data can be performed via SFTP.

2.7.6 Remote monitoring with TCP/HTTP transmission

The transmission of XML data via TCP or HTTP (see Section 8.5) is suitable for the direct connection of database systems. The database servers thus receive the data directly (XML format see Section 8.4.1).

- ✔ For TCP/HTTP transmission, on the one hand the internal system network (e.g. firewall, router) and on the other hand the database server must be correctly configured. Ask your administrator about this.

2.7.7 Remote monitoring with JSON/MQTT transmission

The transmission of JSON data (see Section 8.4.3) via MQTT (see Section 8.9) is suitable for the direct connection of cloud services in the IoT field.

- ✔ In order to send emails, the internal network has to be set up correspondingly (z. B. firewall, router). Ask your administrator about this.

2.8 Technical data

2.8.1 General specifications

Dimensions/Mass

The devices have the following dimensions and the following mass:

- Width: 54 mm
- Height: 90 mm
- Depth: 60 mm
- Mass: approx. 150 g

Mounting

The device is intended for mounting in a control cabinet or a distribution board:

- Temperature range for operation: 0..50 °C (daily average)
- Temperature range for transport and storage: -20..70 °C (short-time)
- Air humidity: 0..95 % relH, non-condensing
- Degree of protection: IP20 (IEC 60529)
- Top hat rail mounting (DIN rail 35 mm, IEC 60715)

2.8.2 Electrical specifications

Power supply

The device is supplied by direct current (pin assignment see Section 2.3):

- Voltage: 12..36 VDC, screw terminals ($\leq$2.5 mm^2, tightening torque 0.5..0.6 Nm)
- Power consumption: 2 W (idle state), max. 40 W
- Safety: reverse polarity protected M-Bus, overvoltage protection (transients), protection class III (IEC 61140), electronic resettable fuse
- Peak inrush-current: approx. 4 A

Meter interfaces

The device has various meter interfaces (pin assignment see Section 2.3):

- M-Bus master: compliant to EN 13757-2/-3/-7, Umark=40 V, Uspace=27 V, screw terminals ($\leq$2.5 mm^2, tightening torque 0.5..0.6 Nm)
  - max. 125 unit loads (UL) for MBUS-GSLE125
  - max. 250 unit loads (UL) for MBUS-GSLE250
  - max. 500 unit loads (UL) for MBUS-GSLE500
  - Max. current rating permanent: 1500 mA
- M-Bus slave: compliant to EN 13757-2/-3/-7, current consumption approx. 3 mA (2 UL), screw terminals ($\leq$1.5 mm^2, tightening torque 0.5..0.6 Nm)
- Baud rate: 300..9600 bps

Communication interfaces

The device has an Ethernet communication interface (pin assignment see Section 2.3):

- Ethernet: compliant to IEEE 802.3, 10/100-Base-TX, RJ45 connector incl. status LEDs, Auto-MDIX

2.8.3 Further specifications

Galvanic isolation

M-Bus master and M-Bus slave are galvanically isolated:

- Galvanic isolation: 1000 V

Processing unit

The central unit is a microprocessor system:

- CPU: ARM9 architecture, 454 MHz clock frequency
- Memory: 128 MB RAM, 4 GB internal eMMC Flash
- Operating system: Linux
- Integrated RTC: backed-up for up to 7 days

3 Tool Netdiscover

3.1 General information

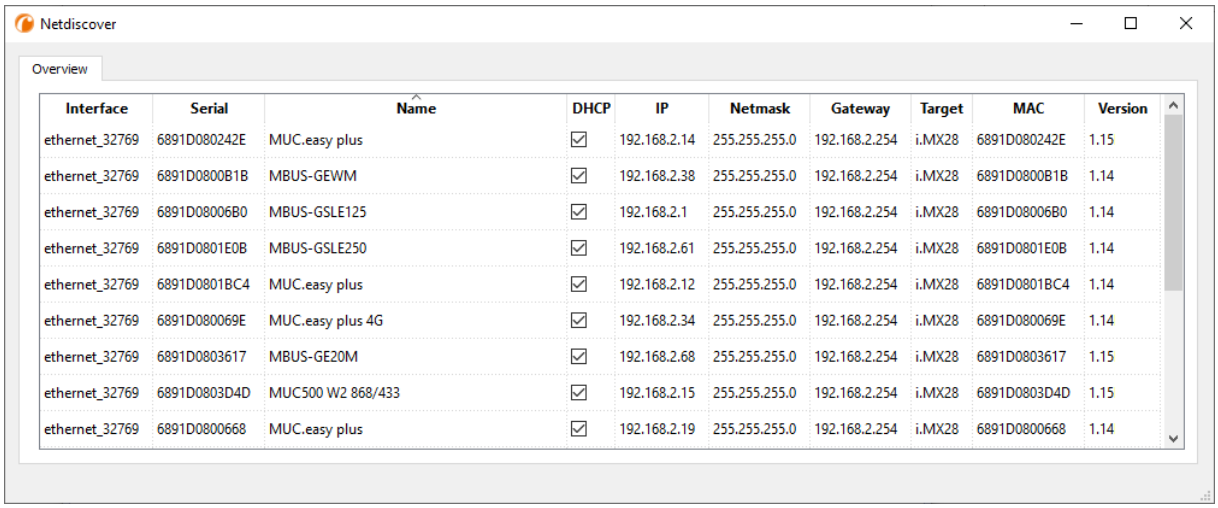
The solvimus GmbH provides its customers with the tool Netdiscover for easier management of products in the customer network. This tool, available for Windows and Linux, allows you to find devices of solvimus GmbH in the local network and to manage them.

Depending on the product and thus on the hardware resp. the software installed on your device, not all the functions and parameters referred to in the text, in tables and figures are available. The screenshots are intended to show examples and depict, as a rule, views from a data concentrator MUC.easy^{plus}. A gateway for instance does not have a report interface for data push or a cellular modem.

The installation comes with two additional programmes. The applications *Putty* and *WinSCP* are utilities for SSH and (S)FTP access. The integration into the tool Netdiscover enables the easy access to the devices from a central location.

3.2 Discovering and accessing devices

After the tool is started, it uses UDP broadcast via UDP port 8001 to discover all devices from solvimus GmbH accessible in the local network and displays them in the main window.



Interface	Serial	Name	DHCP	IP	Netmask	Gateway	Target	MAC	Version
ethernet_32769	6891D080242E	MUC.easy plus	☑	192.168.2.14	255.255.255.0	192.168.2.254	i.MX28	6891D080242E	1.15
ethernet_32769	6891D0800B1B	MBUS-GEWM	☑	192.168.2.38	255.255.255.0	192.168.2.254	i.MX28	6891D0800B1B	1.14
ethernet_32769	6891D08006B0	MBUS-GSLE125	☑	192.168.2.1	255.255.255.0	192.168.2.254	i.MX28	6891D08006B0	1.14
ethernet_32769	6891D0801E0B	MBUS-GSLE250	☑	192.168.2.61	255.255.255.0	192.168.2.254	i.MX28	6891D0801E0B	1.14
ethernet_32769	6891D0801BC4	MUC.easy plus	☑	192.168.2.12	255.255.255.0	192.168.2.254	i.MX28	6891D0801BC4	1.14
ethernet_32769	6891D080069E	MUC.easy plus 4G	☑	192.168.2.34	255.255.255.0	192.168.2.254	i.MX28	6891D080069E	1.14
ethernet_32769	6891D0803617	MBUS-GE20M	☑	192.168.2.68	255.255.255.0	192.168.2.254	i.MX28	6891D0803617	1.15
ethernet_32769	6891D0803D4D	MUC500 W2 868/433	☑	192.168.2.15	255.255.255.0	192.168.2.254	i.MX28	6891D0803D4D	1.15
ethernet_32769	6891D0800668	MUC.easy plus	☑	192.168.2.19	255.255.255.0	192.168.2.254	i.MX28	6891D0800668	1.14

Figure 4: Main window of the tool Netdiscover

The UDP broadcast finds all devices in the local network, regardless of IP settings and subnet masks. Therefore, this function is initially recommended.

The UDP broadcast is usually not forwarded by routers. Therefore, this tool will only find all devices in the local network, in front of the router.

In addition to the MAC address of the devices and their network configuration, the names of the devices and also the version of the operating system are shown. Thus, all devices to be managed can be clearly identified and matched.

The name of the devices corresponds to the **Device name** in ***General*** tab (see Section 4.3).

Various functions can be called using the context menu that appears by right-clicking on one of the devices:

- ***Ping***: starts the ping via ICMP to the device in a separate tab. So, testing of connectivity via TCP is possible.
- ***Web***: opens the default browser with the IP of the device. The web-based front end should open (see Chapter 4).

- ***FTP***: starts *WinSCP* with the IP of the device or blank. The login data or also the IP must be entered before connecting to the FTP/SFTP server of the device.
- ***FTP (default)***: starts *WinSCP* with the IP of the device and connects via FTPS with default login information of the user *admin*.
- ***SSH***: startet *Putty* with the IP of the device. The login data must be entered to connect to the SSH console.
- ***Deploy***: starts the mass deployment for devices in a separate tab.
- ***Import device list***: imports a device list into the main window.
- ***Net configuration***: starts a separate tab for changing the network configuration of the device via UDP broadcast.
- ***Version***: information about the version of the tool Netdiscover (displayed only if no device is selected).

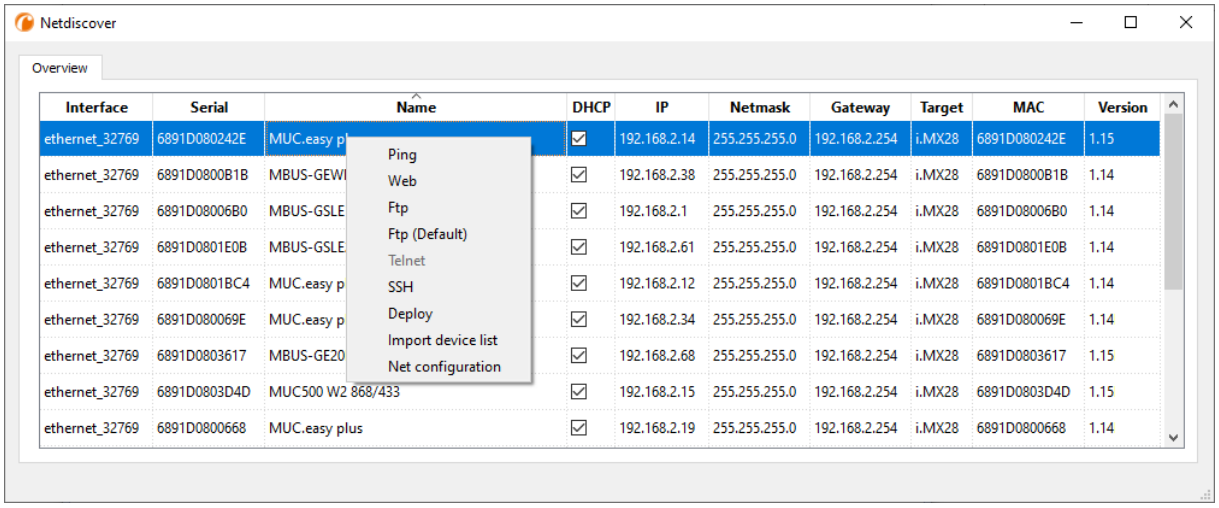


Figure 5: Context menu in the tool Netdiscover

- Depending on the network settings of your PC or your general network infrastructure, the UDP port 8001 may be blocked. Then calls of the tool are blocked and the main window remains empty.
- If a firewall is used in your network (also directly on the PC), there has to be an appropriate firewall rule. This rule should unblock this port to be able to list the devices.
- Ask your administrator about the firewall and network configuration.
- If access via UDP broadcast is denied, a list can be imported with the ***Import device list*** function in order to still be able to use all other functions via TCP.

Some important functions are described more in detail in the following subsections.

3.3 Network configuration

It is often necessary to adjust the network settings of the devices for further work, especially when commissioning devices.

The command ***Net configuration*** from the context menu in the tool Netdiscover opens another tab for the network configuration. Thus, IP address, subnet mask or gateway address can be changed to static or DHCP can be activated for obtaining these settings automatically from a DHCP server.

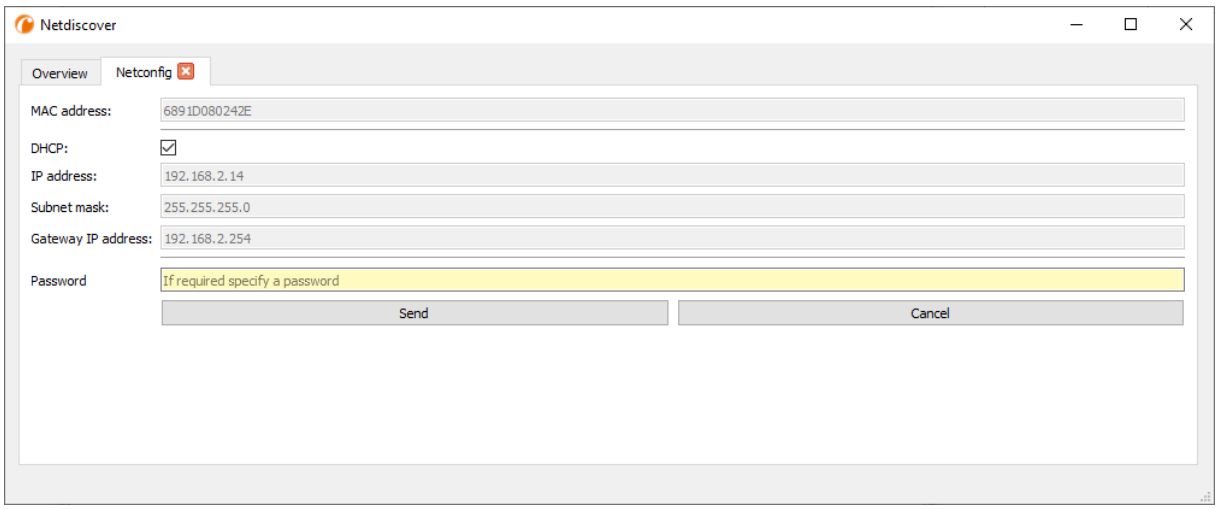


Figure 6: Network configuration via the tool Netdiscover

Modified configurations can be committed pressing the button **Send**. Modifications are only accepted with the password of the user *admin*, the admin password must be inserted in the field **Password**.

If automatic network configuration (DHCP) is selected, all parameters (**IP address**, **Subnet mask** and **Gateway IP address**) will be read from a DHCP server. The respective fields are deactivated then.

The assigned IP address can be identified at the DHCP server from the unique MAC address of the MBUS-GSLE. This address is displayed in the field **MAC address** in the main window of the tool Netdiscover as well as in the tab ***General*** (see Section 4.3) in the field **Serial number**.

Is the automatic configuration not possible in your network (no DHCP server available), the device will pick a standard address (169.254.xxx.xxx) according to RFC3927.

- The standard password in the default factory setting is described in the tab ***User*** (see Section 4.10).
- Changing the network parameters of the device can affect the accessibility. If the network parameters have already been set correctly by an administrator, they should not be changed.

3.4 Access to the web-based front end via HTTP

A web server is integrated on the devices from solvimus GmbH. This enables the configuration of the devices via an integrated, web-based front end (see Chapter 4).

Use the command ***Web*** from the context menu in the tool Netdiscover to quickly and easily call it in the default browser.

- If the web-based front end does not open, please follow the instructions in Section 4.14.

3.5 Access to the file system via FTP

The devices from solvimus GmbH can be accessed via FTP to work directly on the file system level. This enables updates, special configurations and extended functionality (see Chapter 10). The integrated FTP server of the devices supports both FTP and SFTP.

- If access via FTP or SFTP is not possible, check especially the IP settings and the opened ports, 21 for FTP and 22 for SFTP.
- In case of access issues, ask your administrator.

The commands ***FTP*** and ***FTP (default)*** from the context menu in the tool Netdiscover start the *WinSCP* programme and use the IP address of the selected device. Calling the command with a selected device, *WinSCP* always accesses the device via FTP. To use an SFTP connection, the context menu must be called without selecting a device before. Then, only the command ***FTP*** is available, opening a not pre-configured

WinSCP window. So, there is the choice between FTP, SFTP or SCP.

The mode ***FTP (default)*** tries to log in with the default login information of the user *admin*, while in the mode ***FTP*** any access data can be entered.

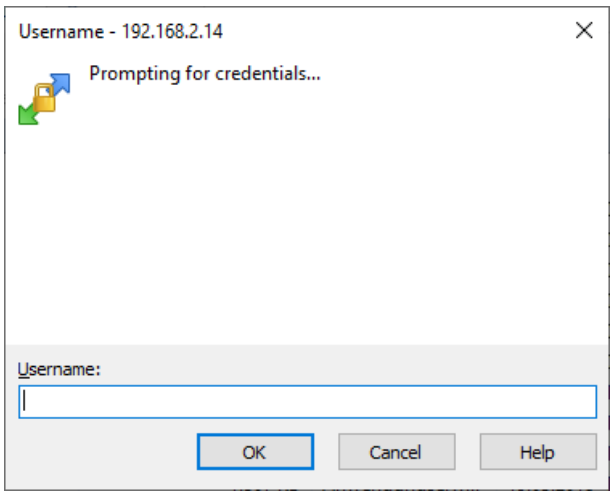


Figure 7: Entering user name when logging in via SFTP

- If the login information of the user *admin* is modified, the command ***FTP (default)*** can not be used.

WinSCP now establishes an SFTP or unsecure/secured FTP connection. When a connection is established to a specific device with SFTP, its authenticity is checked using stored certificates. Normally, the devices from solvimus GmbH are coming with an individual, self-signed certificate upon delivery. This certificate is usually classified as untrusted by your PC. Therefore, a security prompt with information about the device's certificate is displayed. The user must verify the validity of the certificate and then approve it to establish a secure connection. The confirmed certificate is stored in the PC for future connections.

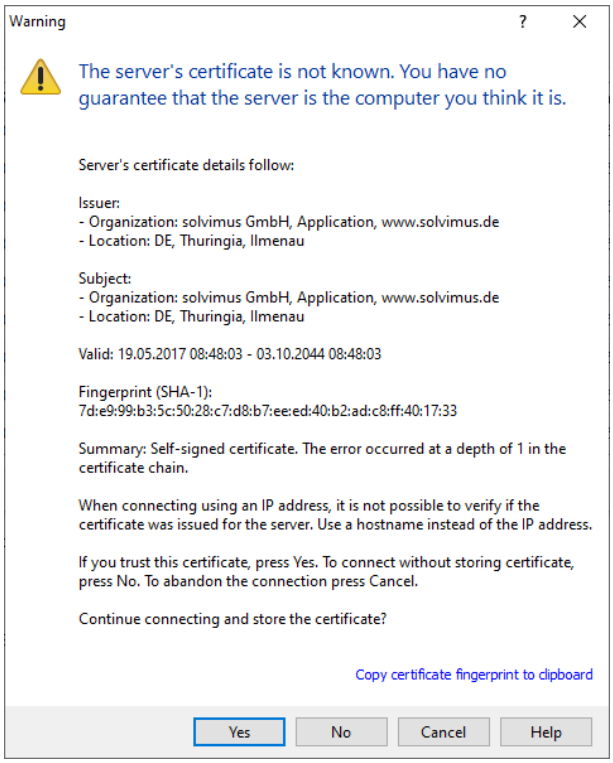


Figure 8: Security prompt for the certificate of the device for FTP access

WinSCP offers a dual-pane file manager after logging in successfully. This allows files to be uploaded to or downloaded from the device. File commands can be executed via a context menu, e. g. copying, renaming or editing. Drag&Drop for uploading and downloading is also supported.

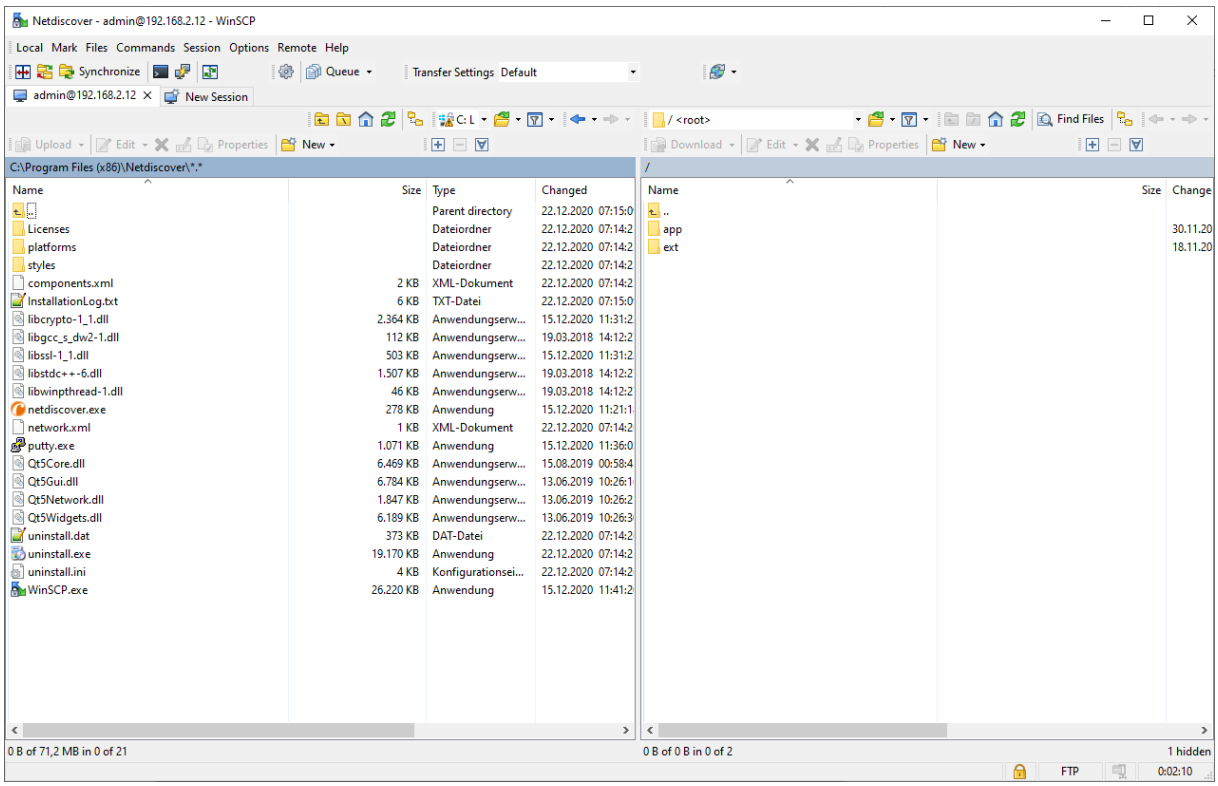


Figure 9: File manager view in *WinSCP*

- Changing files or the file system can affect the functionality of the system.
- The default login information, as delivered, is contained in Section 4.10.

3.6 Access to the command line via SSH

Access to the command line interface (CLI) of the device is suitable for maintenance purposes.

The command ***SSH*** from the context menu in the tool Netdiscover opens the integrated *Putty* client and establishes a connection to the device

When a connection is established to a specific device with SSH, its authenticity is checked using stored certificates. Normally, the devices from solvimus GmbH are coming with an individual, self-signed certificate upon delivery. This certificate is usually classified as untrusted by your PC. Therefore, a security prompt with information about the device's certificate is displayed. The user must verify the validity of the certificate and then approve it to establish a secure connection. The confirmed certificate is stored in the PC for future connections.

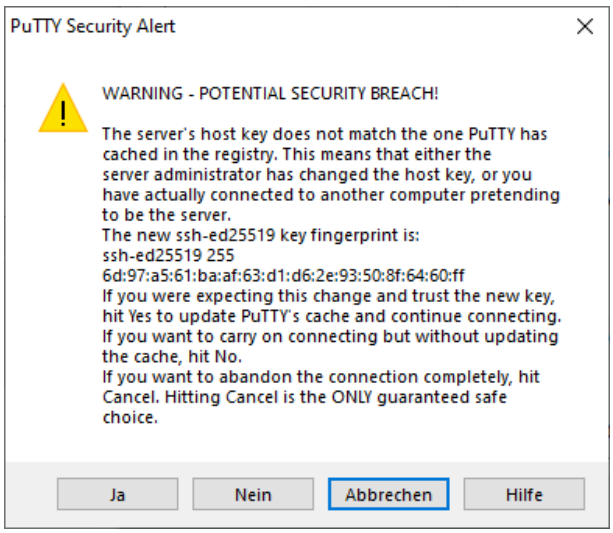


Figure 10: Security prompt for the certificate of the device for SSH access

Now the *Putty* client opens and the login information for the user *admin* has to be entered. Then, the command line is ready for input via SSH. The password is not displayed on the screen.

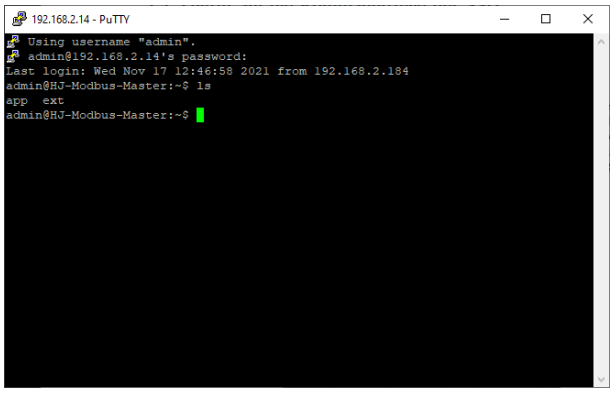


Figure 11: Command line in the *Putty* client

- Inputs on the command line can affect the functionality of the system.
- The default login information, as delivered, is contained in Section 4.10.

3.7 Mass deployment

This function allows performing certain device configurations or firmware updates in parallel for all devices displayed in Netdiscover. For example, is is possible to import an previosly exported device configuration to multiple other devices at the same time. Another example would be importing certificate files needed on multiple devices to export meter data. A third and final example would be updating the application software on multiple devices in parallel.

- The configuration or update should explicitly only be deployed on similar devices.

In this case mark the devices in the tool Netdiscover on which you want to perform a configuration or firmware update in parallel.

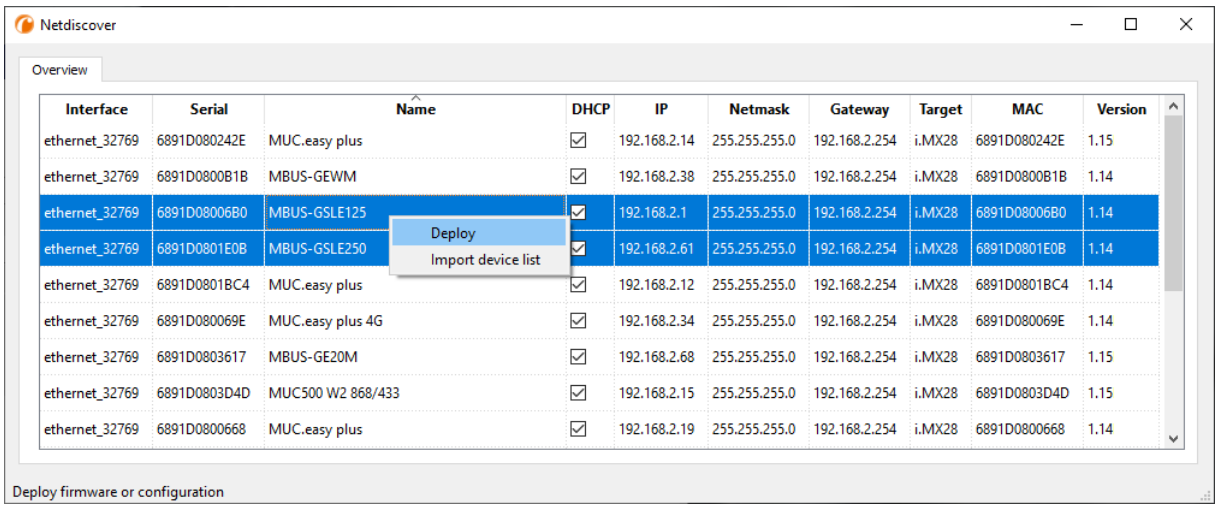


Figure 12: Selection of devices and initiation of the mass deployment

The command ***Deploy*** from the context menu in the tool Netdiscover opens another tab for mass deployment.

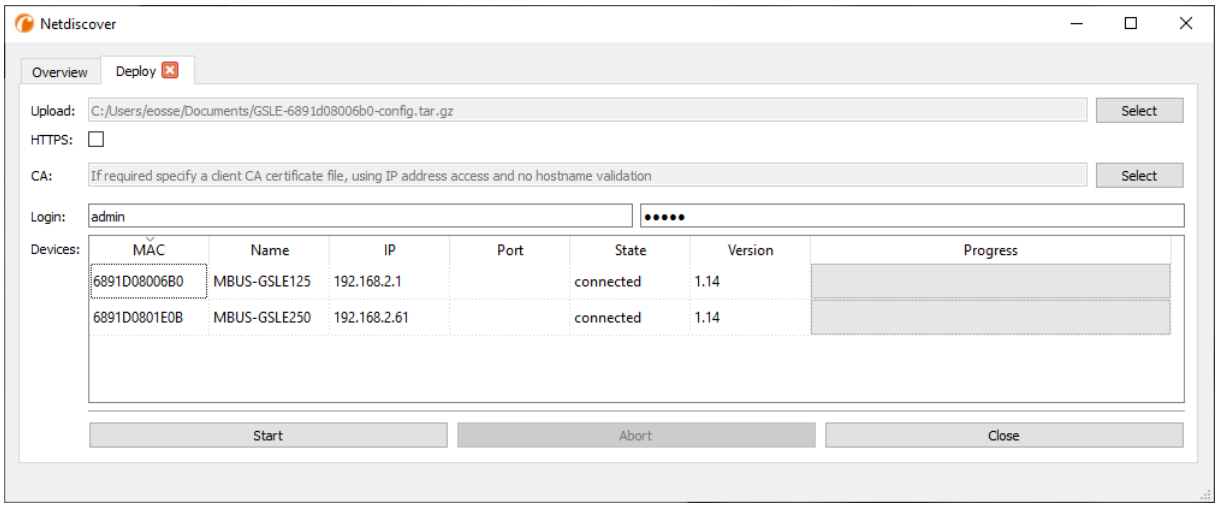


Figure 13: Mass deployment via the tool Netdiscover

The following input fields and buttons are available here:

- **Upload**: the configuration or update to be uploaded.
- **HTTPS**: selection field whether HTTP or HTTPS should be used.
- **CA**: the CA certificate to verify the client certificate of the devices for HTTPS-based work.
- **Login**: user name and password for the user *admin*.
- **Start**: starts the process.
- **Abort**: aborts the process.
- **Close**: closes the mass deployment tab.

In the central part, there is a list view with information about the devices and the status/progress of the process.

- Exclusively *.tar.gz archives are intended for the import of a device configuration or a certificate file.
- The generation of a *.tar.gz file with the device configuration is described in Section 4.12.
- Exclusively *.enc files are intended for the update of the firmware.
- An update of the firmware is also possible via the web site as described in Section 4.12.

The file is unpacked on the device after the upload, and processed. The device is then restarted.

3.8 Import of a device list

Devices cannot always be discovered automatically. Firewalls, routing settings or even the deactivation of the function **Network discovery active** in the ***Security*** tab (see Section 4.9) are possible reasons.

Therefore, a device list can be imported. This enables managing devices via the tool Netdiscover even without automatic dicovery.

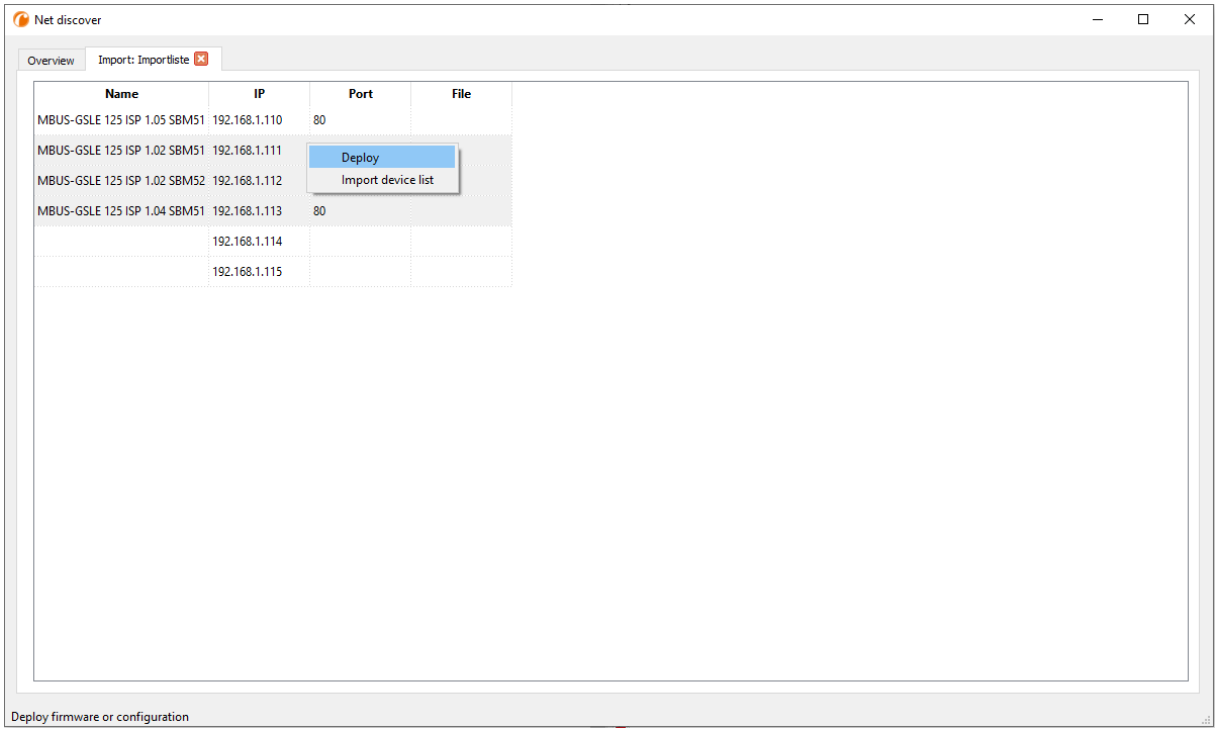


Figure 14: Viewing and using an imported list in the tool Netdiscover tool

First, a suitable CSV file has to be created before the actual import. In the CSV file, a comma or a semicolon can be used as a separator. The device data is entered here according to the following example to obtain the above list in the tool Netdiscover:

```
Port;Name;Password;Username;IP;File
80;MBUS-GSLE 125 ISP 1.05 SBM51;admin;admin;192.168.1.110;
80;MBUS-GSLE 125 ISP 1.02 SBM51;admin;admin;192.168.1.111;
80;MBUS-GSLE 125 ISP 1.02 SBM52;admin;admin;192.168.1.112;
80;MBUS-GSLE 125 ISP 1.04 SBM51;admin;admin;192.168.1.113;
;;admin;;192.168.1.114;
;;;;192.168.1.115;
```

- The header of the CSV file has to be identical to the one above.
- Only the *IP* column is mandatory. The other columns can be left empty and are set to default for special functions (*Port*: 80, *Password*: admin, *Username*: admin).

3.9 Troubleshooting network

3.9.1 No network connection

If no network connection to the device can be established, make a Ping connectivity test first (see Section 3.2).

If a Ping response is not detected, test the device via a direct network connection with a PC, provided the device is connected via a bigger network. Depending on the functions, a cross-over cable may need to be employed in case of a direct connection between PC and device.

Check the physical network connection between the device and the PC, if the cable is correctly joined and inserted.

✔ The network connection must be inserted in the connector for Ethernet.

At the network connection the *hyperlink-LED* must be lit yellow and the *Active-LED* must flash green from time to time. Check also the corresponding LEDs at the remote station (PC, hub etc.). If need be, repeat the connectivity test with switched cables.

If all LEDs are lit correctly, check if the device is detected in the tool Netdiscover (see Section 3.2). A prerequisite is that the device is connected to the PC via a local area network.

If the device being searched is not contained in the list (allocation via serial number), ensure that the communication is not prevented by a firewall.

If the device is in the list, configure it with a unique IP address available in the local network (see Section 3.3). Ask your administrator about this.

For a direct connection between PC and network, the following example configuration can be employed, provided no other participant is connected to the network with these addresses:

PC	
IP	192.168.1.10
Network mask	255.255.255.0
Device	
IP	192.168.1.101
Network mask	255.255.255.0

Table 6: Example IP addresses

If errors could not be eliminated, please contact our customer support:
E-Mail: support@solvimus.de
Phone: +49 3677 7613065

3.9.2 The device can not be accessed via website respectively FTP(S)

If the device can not be accessed via a browser, make a Ping connectivity test first (see Section 3.2) or log on tentatively via FTPS (see Section 3.5). If a network communication with the device is not possible in general, follow the instructions in section Section 3.9.1. If a single service is not available, check the passwords and the firewall configuration at the PC respectively in the network.

Is the web page displayed whereas a login is not possible, check if you can log on with the *admin* account. Clear the cache in the browser and reload the website (e. g. key ⟨**F5**⟩ respectively ⟨**CTRL+F5**⟩).

If errors could not be eliminated, please contact our customer support:
E-Mail: support@solvimus.de
Phone: +49 3677 7613065

4 Web-based front end

4.1 General information

Many products from solvimus GmbH, especially data concentrators and gateways for smart metering, are coming with an integrated web server and provide a website for the configuration. The devices can be configured easily and in a user-friendly manner via this website. Device parameters, meter configuration as well as services can be displayed or changed on this website.

This chapter gives an overview on how to use the web-based front end.

- Depending on the product and thus on the hardware resp. the software installed on your device, not all the functions and parameters referred to in the text, in tables and figures are available. The screenshots are intended to show examples and depict, as a rule, views from a data concentrator MUC.easy^{plus}. A gateway for instance does not have a report interface for data push or a cellular modem.

The web-based front end can easily be opened in the browser by entering the device's IP address. Alternatively, right-click on the device in our tool Netdiscover (see Chapter 3) and select the command ***Web*** in the context menu to launch the browser.

- We are testing the web-based front end in different browsers. We recommend using Chrome™ and Firefox browsers for optimal user experience. For the legally secure and data protection compliant setting of your browser, please ask your administrator.

The browser automatically tries to log in the user to the website using the default login information. The user „web“ with the password „web“ is used for this purpose. This user has initially full access to the website. This facilitates the initial commissioning.

When the default user „web“ has been modified in the configuration via the ***User*** tab, for example by changing the password, the automatic login is not possible anymore. Only entering the new, correct login information will allow accessing the front end. A login dialogue will then always appear:

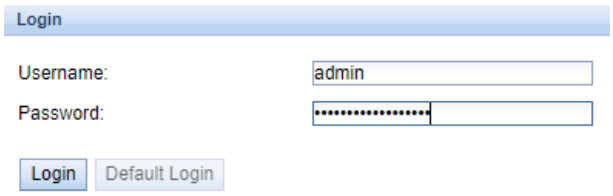


Figure 15: Login dialogue

- For switching to another user (e. g. from the default user), the **Logout** button at the top right of the web-based front end can be clicked.
- The default login information, as delivered, is contained in Section 4.10.

If the logged-in user has write access, the user has to log out after the configuration is finished. If the connection remains active, no other write access to the web-based front end is available. Only one session with write access is possible at a time.

- When a session is terminated without logging out previously, e. g. by closing the browser window, it remains active for approx. 1 min. Afterwards it is automatically closed and write access is possible again.

On the website of the device (see Figure 16), the functions are grouped into different tabs. So, the clarity can be maintained despite the large number of parameters. All modifications in one of the tabs must be saved before changing tabs, otherwise the modifications will be lost. The functions and parameters of the individual tabs are described below.

The **Print** button (see Figure 16, bottom right) can be used for getting an entire overview of the configuration or for exporting it via the clipboard. Details are given in Section 4.13.

The solvimus GmbH provides a manual in PDF format directly on the device. Click the **Help** button (see Figure 16, bottom right) to open the included PDF file.

4.2 Access via HTTPS

Normally, the web-based front end is accessible via HTTP (port 80) as well as via HTTPS (port 443). Depending on the requirements, one of the services can be deactivated (see Section 4.12).

Compared to HTTP, HTTPS offers both encryption and authentication methods and thus enables secure access to the devices in insecure networks.

The devices from solvimus GmbH are delivered with certificates and keys for preparing HTTPS access:

- *app/keys/http_host_cert*: self-generated certificate of the device to verify the identity of the device, server-side authentication
- *app/keys/http_host_key*: private key of the device

The user can upload another certificate to the device to fully secure the communication and for mutual authentication.

- *app/keys/http_host_ca*: root certificate to check the client certificate of the browser and thus the identity of the client, client-side authentication

Based on these files, the communication partners can securely identify and authenticate each other before a symmetric session key is negotiated.

- Access to the web-based front end via HTTPS can be blocked by installing incorrect or invalid certificates.
- Deactivating HTTPS or HTTP is only available via the respectively other access variant.
- Optionally, customer-specific certificates can be uploaded during production.

4.3 Tab General

The ***General*** tab displays general properties of the device and its network configuration.

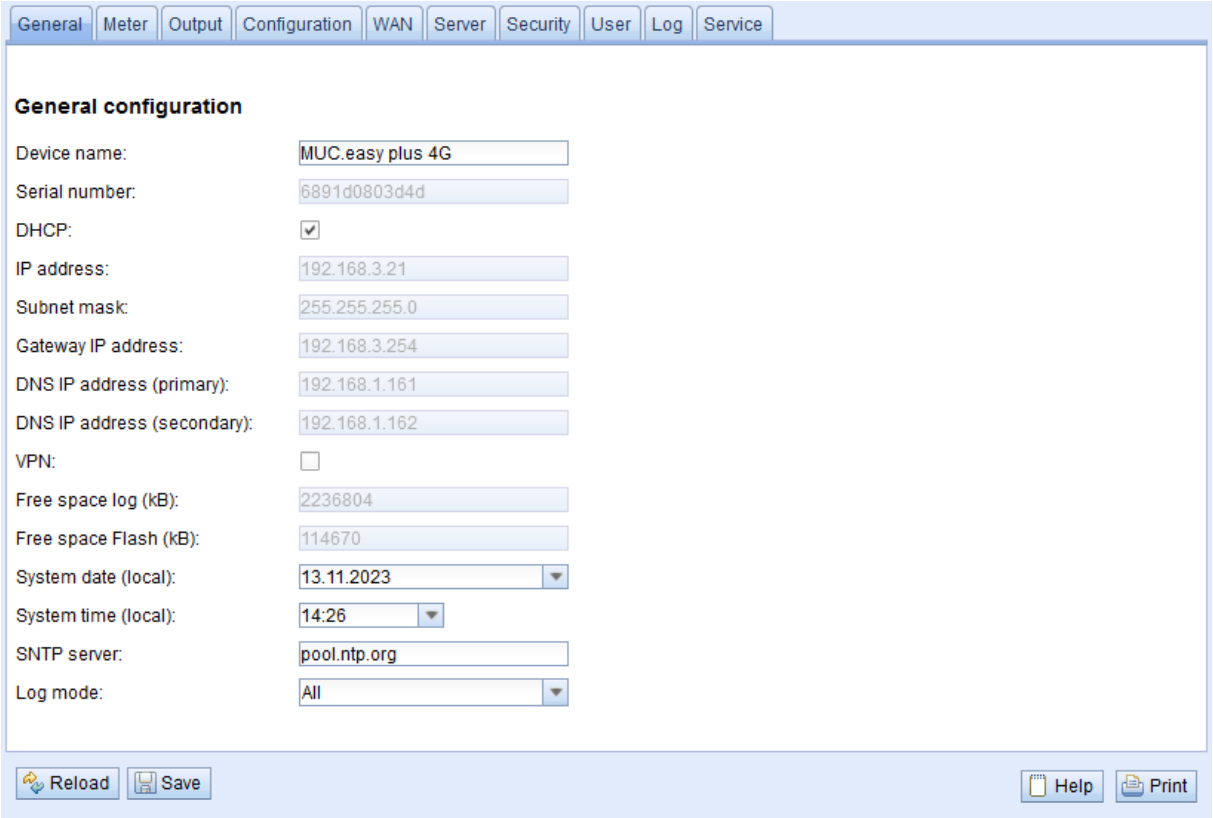


Figure 16: Tab General

The following parameters are shown and can be changed here:

Column name	Description
Device name	Name of the device (as assigned in the tool Netdiscover, max. 50 characters)
Serial number	Serial number of the device (MAC address), not editable
DHCP	Enable automatic network configuration. If no DHCP-server is available for the network configuration, the tick is set to inactive and the network interface can be configured using a free IP in the address space 169.254.0.0/16 (Zeroconf).
IP address	IP address of the device, not configurable when using DHCP
Subnet mask	Subnet mask of the device, not configurable when using DHCP
Gateway IP address	IP address of the standard gateway, not configurable when using DHCP
DNS IP address (primary)	IP address of the primary DNS server, not configurable when using DHCP
DNS IP address (secondary)	IP address of the secondary DNS server, not configurable when using DHCP
VPN	Activates the OpenVPN client functionality
Free space log (kB)	Free disk space for logging, not editable
Free space Flash (kB)	Free disk space for applications, not editable
System date (local)	Current, localized system date
System time (local)	Current, localized system time
SNTP Server	Address of the time server
Log mode	Level of detail of the log entries of the application ▪ *None*: The application does not generate any log entries. ▪ *Standard*: The application generates log entries for errors and warnings. ▪ *All*: The application generates log entries for all events.

Table 7: Fields in the General tab

The **Save** button is used to save the configuration. The **Reload** command loads the last saved parameters and resets current changes.

If the network configuration is changed, the device will be available under the new IP right after processing the changes. All active sessions will be closed and users will be logged out automatically then.

- Changing the network parameters of the device can affect the accessibility. If the network parameters have already been set correctly by an administrator, they should not be changed.
- The device is automatically reinitialized by accepting the parameters via the **Save** button.

Date and time are always processed as UTC time (without time zone shift). When shown on the website, the browser converts it according to the time zone of the respective computer. In Central Europe, for example, this is Central European Time or Central European Summer Time. If a different time zone is used here, the time shown on the website will be displayed accordingly.

The use of OpenVPN is described in the Section 10.5.

4.4 Tab Meter

The ***Meter*** tab displays an overview of the connected meters. It offers further possibilities to the user: searching meters automatically, adding meters manually and configuring meters that are already present. The meter list can additionally be exported through it.

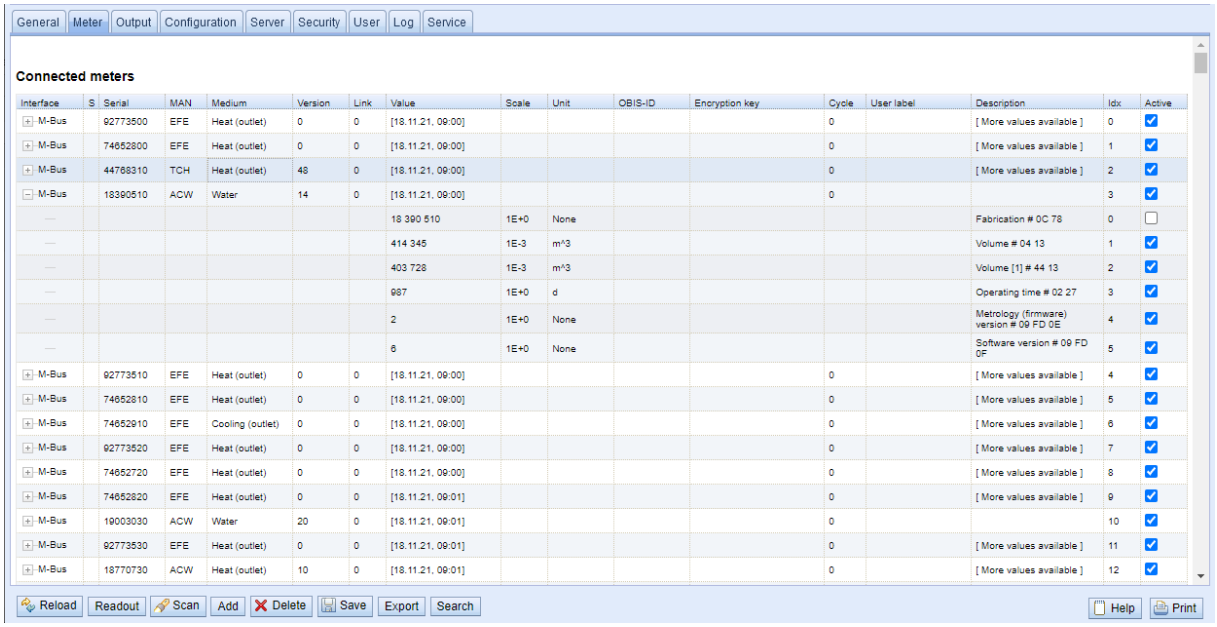


Figure 17: Tab Meter

The meter list is displayed in tabular format. Meter entries and the corresponding meter value entries are displayed one below the other. The individual columns have the following meaning:

Column name	Description
Interface	Interface to the meter ▪ *M-Bus*: wired M-Bus according to EN 13757-2/-3/-7 and OMS ▪ *wM-Bus*: wireless M-Bus according to EN 13757-4/-3/-7 and OMS ▪ *DLDE*: wired serial interface according to IEC 62056-21 or IEC 1107/61107 ▪ *Modbus*: interface via RS-485 (Modbus RTU) or Ethernet (Modbus TCP, according to IEC 61158) ▪ *S0*: wired counting/pulse input interface according to IEC 62053-31 or for simple contact outputs ▪ *System*: Monitoring of internally measured values from the device
S (Status)	Shows the status of the meter or the meter value ▪ *!*: meter or meter values cannot be read, meter values are not up-to-date. ▪ *E*: meter/meter value edited ▪ *A*: meter/meter value added ▪ ***: Meter value list of that meter is limited (see *Maximum value count* parameter in ***Configuration*** tab)
Serial	Serial number of the meter (meter number, secondary ID)
MAN	Manufacturer of the meter (abbreviation), DLMS Flag-ID
Medium	Meter medium, see second column in Table 26
Version	Version number of the meter
Link	Primary address of the meter for M-Bus resp. reception quality (RSSI, in steps of -0.5 dBm) for wM-Bus
Value	Meter reading or measured value (unscaled)

Continued on next page

Table 8 – *Continued from previous page*

Column name	Description
Scale	Scaling factor (scientific notation). The value is defined by $Value \rightarrow Value \cdot Scale$
User Scale	Scaling factor (scientific notation). It complements the *Scale* provided or set by the meter, but does not replace it. It is suitable if an additional scaling is necessary. The value is defined by $Value \rightarrow Value \cdot Scale \cdot User\ Scale$ A column for *User Scale* is displayed only if *User Scale* deviates from the default value of *1e+0* (see Table 25).
Unit	Unit, see second column in Table 28
OBIS-ID	OBIS code in the format X-X:X.X.X*X (X=0..255)
Encryption key	Key for encrypted wM-Bus meters. Supported modes: 5 and 7
Cycle	Readout interval in seconds (with 0, the general readout cycle is used, see ***Configuration*** tab)
User label	User-defined description of the meter value, this allows an application-specific mapping. Allowed characters are: A-Z, a-z, 0-9, !, §, $, %, &, /, (,), =, ?, + and *. A comma is also allowed. Illegal characters are: ⟨, ⟩ and ". If using the CSV format, the semicolon (or the corresponding separator) should not be used.
Description	Description of the meter value according to the second column in Table 27. The display of storage number, tariff, value type and raw data can be configured via the *Description mode* parameter in the ***Configuration*** tab.
Idx	Index/position of meter/meter value in the meter list
Register	Offset of the register set belonging to the value when using the Modbus server *
BACnet	Object number of the value when using the BACnet server *
Active	Activates a meter or meter value for reporting to a server or logging.

*if device is equipped with this interface/function

Table 8: Columns in Meter tab

The meter configuration can be changed with the buttons at the bottom or via the context menu. According to the limitations of the interface used (M-Bus, wM-Bus etc.), individual meters or meter values can be automatically scanned or manually created, deleted or changed.

The meters or meter values in the list can be selected by a simple mouse click. A range can be selected with the ⟨**SHIFT**⟩ key held down, or multiple meters can be selected (individually) with the ⟨**CTRL**⟩ key held down.

Duplicates of the serial number are marked yellow for easier checking of the meter list. Using the **Search** button, the complete meter list can be searched for a text. The search comprises as well meter values hidden by closing the symbol in front of the interface type.

Reload loads the last saved parameters, resets current changes, and correspondingly updates the meter values.

Upon delivery, the device has an empty meter list. If meters are connected via the external interfaces of the device, the **Scan** button can be used to start an M-Bus scan. The scan mode *M-Bus mode* is configured in the ***Configuration*** tab. More information on this can be found in Section 4.6.

- ✔ Depending on the mode and the number of connected meters, this may take a very long time.

The process can be interrupted using the **Cancel** button, whereby the meters already found are saved in the meter configuration. After the scan, the meter configuration is immediately applied, and only needs to be saved again after further changes. The scan procedure is only adding meters to the existing list, it is not deleting or changing already configured meters. Newly found M-Bus meters and their values are automatically activated after the scan and are assigned to a Modbus address or a BACnet number. The scan also permanently adds newly received wM-Bus meters to the configuration, provided that the parameter *wM-Bus listen* in the ***Configuration*** tab is activated. Since wM-Bus meters are not necessarily your own, they are not automatically activated, unlike M-Bus meters. The listen mode initially only lists all received meters without permanently saving them to the list.

- ✔ The meter values of M-Bus and wM-Bus meters are arranged in the same order as the data is present in the protocol. So, the meaning of the values can be directly compared with the data sheet of the relevant meter. Alternatively, the raw data of the meter values (see parameter *Description mode* in the ***Configuration*** tab, see Section 4.6) can be used for mapping the values.
- ✔ The timestamps transmitted in the M-Bus or wM-Bus protocol are automatically assigned to the individual measured values, and therefore not listed in the meter list by default. The configuration parameter

MUC_SHOWTIMESTAMPENTRIES in the configuration file *app/chip.ini* allows to manually activate the explicit representation of all timestamps (see Section 10.3).

- Newly received wM-Bus meters are deactivated by default, and have to be manually activated and saved in order to be integrated into the reports and log data. Unsaved wM-Bus meters are lost after a restart.

Meters which cannot be found as well as meters connected to interfaces which do not enable automated scanning can be added manually using the **Add** button or using the ***Add meter*** item in the context menu. The number of meters is limited. The button **Add** and ***Add meter*** in the context menu are automatically deactivated once the maximum number of meters is attained.

For configuring individual meters or meter values, double click an entry or call the editing dialogue with the ***Edit*** context menu item. The naming of the input fields corresponds to the columns of the meter list (see Table 8). Individual fields are activated or deactivated according to the interface.

Among other things, a *User label* can be assigned to all entries here, so the meter or meter value can be mapped to a specific application. The individual readout interval of the meters can be set via the parameter *Cycle* as well. The key required for decoding can also be set for wM-Bus meters in the Meter editing dialogue.

- S0 meters are internally processed with the number of pulses. The representation on the website in the *Value* column is nevertheless scaled to provide better readability. The *Scale* column contains the pulse value and, in contrast to other meter interfaces, does not have to be additionally multiplied. If a value of 280.09 and a scaling of 1e-4 is displayed in the ***Meter*** tab, 2800900 pulses are recorded internally. However, this unscaled meter value (280.09) appears in the report data analogously to those of other meters, such as the CSV or the XML files.
- Meter values of S0 meters can only be set in the Add or Edit dialogue if the *Set value* checkbox is activated. The *Set value* checkbox must be deactivated if a configuration is not meant to change or overwrite the current meter value (e. g. change of the user label). The input of a meter value needs to be scaled.
- Before saving the entered value of a S0 meter value, it is calculated back to the pulse count and rounded to whole pulses. Inaccuracies can result from the floating point data types.

The configuration can be finished with the **Ok** button or cancelled with the **Cancel** button.

For reporting and logging, individual meters and meter values can be directly activated or deactivated with the checkbox in the *Active* column. The meter values are automatically activated or deactivated by the configuration of a meter corresponding to the hierarchy. In the same way, an inactive meter is automatically activated if one of its meter values is activated. Multiple selected meters or meter values can be set with the context menu items ***Activate*** and ***Deactivate***.

All selected meters and meter values can be deleted by using the **Delete** button or the context menu item with the same name. Deleted wM-Bus meters are then created again if the parameter *wM-Bus listen* in the ***Configuration*** tab is activated.

- Individual meter values of an M-Bus or wM-Bus meter cannot be deleted.

The meter list is saved by using the **Save** button.

- Saving a meter configuration creates a new internal database file for logging the meter values aligned to this updated configuration.

The **Export** button can be used to export the meter list as a CSV file in the mode *Meter list* or to export the data pertaining to a particular instant as CSV, XML, JSON or User file in the mode *Log data (all meters)* resp. *Log data (selected meters)*, if Reporting is active in the ***Server*** tab with the settings defined therein. The time frame for the export of the meter data stretches from **Date (local)** and **Time (local)** to **End date (local)** and **End time (local)**.

- Logged meter data can only be exported if data was recorded for the specified period, i. e. at least one report was active during this period (see Section 4.8).

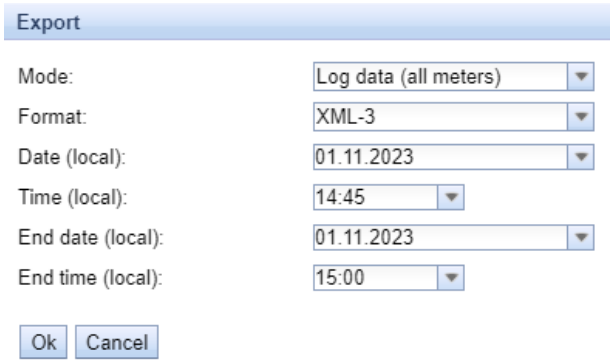


Figure 18: Exporting log data in the Meter tab

4.4.1 System meter

The system meter is a special function for providing device-specific operating parameters. These parameters are displayed via the system meter like normal meter values and can thus be monitored and analysed. The system meters must be added manually in the tab ***Meter*** using the **Add** button or using the ***Add meter*** item in the context menu.

Depending on the device, the parameters in the following table are available. Here, x denotes the S0 inputs (pulse inputs) and y the digital outputs.

Entry	Description
Digital input <x>	State of the digital input, channel x (S0 inputs)
Digital output <y>	State of the digital output, channel y
Operating time	Operating time counter, in seconds
Reset counter	Power outage counter
Temperature	Board temperature, uncalibrated
Ampere	Bus load on M-Bus
On time	Time counter since last power outage, in seconds
CPU	CPU load
Memory	Free RAM
Memory <1>	Free memory of the application partition
Memory <2>	Free memory of the database partition
RSSI	Field strength of the cellular network in dBm (-113 to -51 dBm, -114 corresponds to be not connected)

Table 9: Values of the system meter

System	D0803D4D	SLV	Communication controller	135	0	[11.05.22, 16:31]			0			2	☑
—						1	1E+0	None			Digital Input	0	☑
—						1	1E+0	None			Digital Input	1	☑
—						1	1E+0	None			Digital Input	2	☑
—						0	1E+0	None			Digital output	3	☑
—						19 364 133	1E+0	s			Operating time	4	☑
—						32	1E+0	None			Reset counter	5	☑
—						38	1E+0	Degree C			Temperature	6	☑
—						4	1E-3	A			Ampere	7	☑
—						1 141	1E+0	s			On time	8	☑
—						17	1E+0	%			CPU	9	☑
—						27 832	1E+0	kBytes			Memory	10	☑
—						111 950	1E+0	kBytes			Memory	11	☑
—						2 442 596	1E+0	kBytes			Memory	12	☑
—						-104	1E+0	dBm			RSSI	13	☑

Figure 19: System meter in Meter tab

➔ The system meter can be extended by further meter values via scripts. This is described in Section 10.7.3.

4.5 Tab Output

The tab ***Output*** lists, independent from the interface, an overview of the switchable digital outputs of all connected meters from the tab ***Meter***. These digital outputs can be switched via a checkbox.

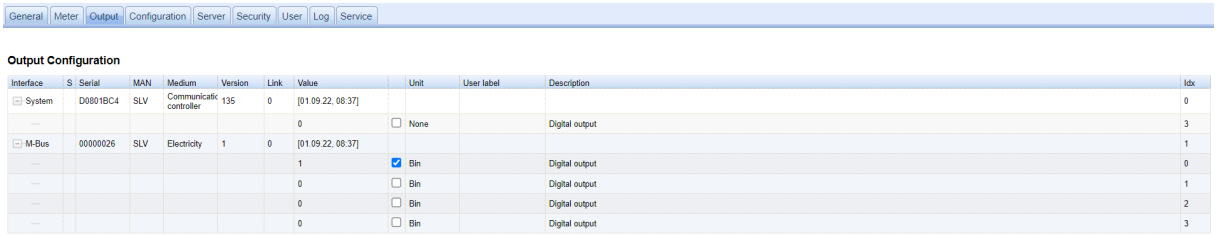


Figure 20: Tab Output

By default, only the S0 inputs and the digital output of the system meter can be switched. Information on the system meter is given in Section 4.4.1. If need be, the settings can be extended via the configuration file *chip. ini* (see Section 10.3). In the **Group [SOLVIMUS]**, the parameter *MUC_SETDEVICES* must be set.

4.6 Tab Configuration

The ***Configuration*** tab allows the parametrization of the meter interfaces of the device.

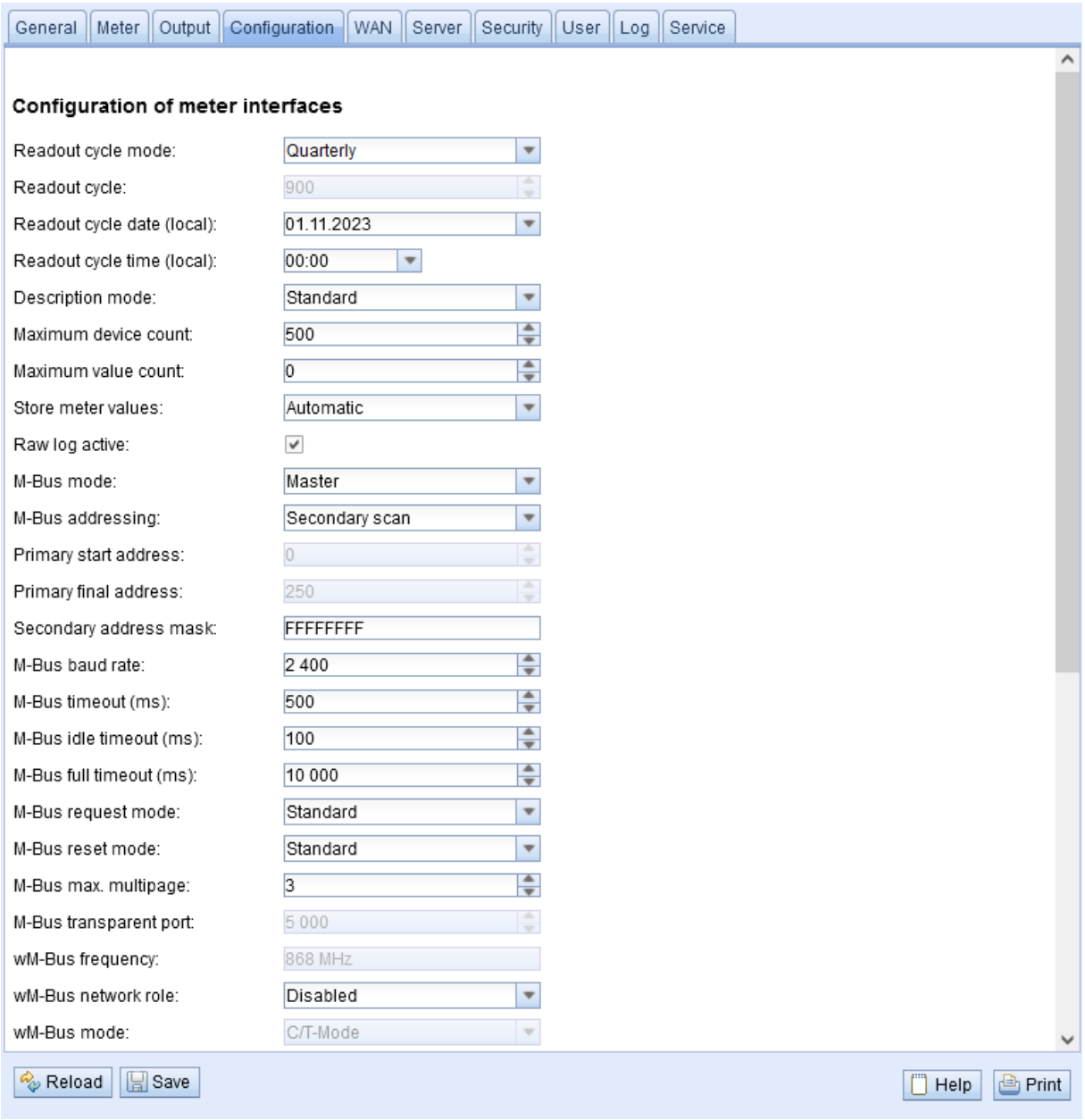


Figure 21: Tab Configuration

The following parameters are available:

Column name	Description
General readout and display parameters	
Readout cycle mode	Format for specifying the standard readout cycle (for all meters, unless otherwise specified for individual meters in the ***Meter*** tab via the parameter *Cycle*). ▪ *Second*: Readout cycle is specified in seconds ▪ *Minute*: Readout cycle is specified in minutes ▪ *Hour*: Readout cycle is specified in hours ▪ *Daily*: daily readout at the specified time ▪ *Weekly*: weekly readout on the specified weekday and at the specified time ▪ *Monthly*: monthly readout on the specified day of the month and at the specified time ▪ *Quarterly*: quarterly readout on the specified day and month of the quarter and at the specified time (month 1..3 per quarter) ▪ *Yearly*: yearly readout on the specified day and month and at the specified time
Readout cycle	Standard readout cycle of the meters (unit according to *Readout cycle mode* in seconds, minutes or hours; only for *Readout cycle mode* in *Second*, *Minute*, *Hour*)
Readout cycle date (local)	First readout day in case of daily to yearly specification of the standard readout cycle, depending on the interval format the entered month is used, the year is not relevant
Readout cycle time (local)	Readout time for daily to annual specification of the standard readout cycle
Description mode	Mode for displaying the meter value description on the website: ▪ *None*: empty meter value description ▪ *Standard*: simple meter value description (see Table 27) ▪ *Extended*: extended meter value description (parameters are only shown if not zero): Notation: description [storage number] ⟨tariff⟩ {value type} Example: Energy [2] ⟨1⟩ {max} ▪ *Extended with DIF/VIF*: extended meter value description added by raw DIF/VIF data: Notation: description [storage number] ⟨tariff⟩ {value type} # XX XX XX ... Example: Energy [2] ⟨1⟩ # 8C 11 04 ▪ *Extended with raw data*: extended meter value description added by complete raw data for this entry. Notation corresponds to *Extended with DIF/VIF*: Example: Energy [2] ⟨1⟩ # 8C 11 04 96 47 06 00 ▪ *DIF/VIF*: raw DIF/VIF data in description field ▪ *Raw data*: complete raw data for this entry in description field
Maximum device count	Limits the number of meters being added upon scanning (0: no limit). Already configured meters are included by this parameter.
Maximum value count	Limits the number of meter values for a meter during a readout process (0: no limit). Already configured meters keep their original configuration after initial scan or saving.
Store meter values	Setting if the read out values are to be written into the database when no report is active. ▪ *Automatic*: storage only if a report is active ▪ *On*: always storage This selection is only offered if the device supports reports and database storage.
Raw log active	Activating the logging of raw data from the interfaces
Specific parameters of the M-Bus-Master*	
M-Bus mode	Configuration of the communication. The following modes are available: ▪ *Disabled*: The M-Bus interface is deactivated. ▪ *Master*: The device is M-Bus master and can read out meters. ▪ *Transparent/TCP*: The M-Bus interface is available for a transparent communication via TCP. ▪ *Transparent/UDP*: The M-Bus interface is available for a transparent communication via UDP. ▪ *Master & Transparent/TCP*: The device is M-Bus master and can read out meters. The interface is at the same time available for a transparent communication via TCP.
M-Bus addressing	Configuration how the device searches meters during an M-Bus scan and how these meters are addressed (details see Section 5.3.2). The following modes are available: ▪ *Primary Scan*: Search for primary address ▪ *Secondary scan*: Search for secondary address ▪ *Secondary scan reverse*: Search for secondary address in inverted order
Primary start address	Sets the start address for the primary search.
Primary final address	Sets the final address for the primary search.
Secondary address mask	Sets the address mask for the secondary search, 8 digits; wildcards are indicated by the letter "F"; missing characters are filled up with leading 0 from the left.
M-Bus baud rate	M-Bus communication baud rate
M-Bus timeout	M-Bus timeout until first data is received (in ms)

Continued on next page

Table 10 – *Continued from previous page*

Column name	Description
M-Bus idle timeout	M-Bus timeout for detecting the end of communication (in ms)
M-Bus full timeout	M-Bus timeout (total) for the reception of a data telegram (in ms)
M-Bus request mode	Mode of the M-Bus readout process (REQ_UD2): ▪ *Standard*: Readout process using REQ_UD2 ▪ *Extended 1*: Readout process using Get-All-Data (DIF/VIF 0x7F 0x7E) and REQ_UD2 ▪ *Extended 2*: Readout process using Get-All-Data (DIF 0x7F) and REQ_UD2
M-Bus reset mode	Mode of the M-Bus reset (before scan and readout process): ▪ *None*: No reset ▪ *Standard*: SND_NKE to the primary address of the meter or to the broadcast address 0xFF in case of secondary addressing ▪ *Extended 1*: SND_NKE to the primary address 0xFD, followed by a SND_NKE to the primary address of the meter or to the broadcast address 0xFF in case of secondary addressing ▪ *Extended 2*: SND_NKE to the primary address 0xFD, followed by an application reset to the broadcast address 0xFF, followed by a SND_NKE to the primary address of the meter or to the broadcast address 0xFF in case of secondary addressing
M-Bus max. multipage	Limits the number of multipage requests
M-Bus transparent port	Network port of the transparent M-Bus mode
Specific parameters of the M-Bus-Slave*	
M-Bus slave mode	Sets the mode of the M-Bus slave (M-Bus, TCP or UDP) or deactivates the interface.
M-Bus slave baud rate	Sets the baud rate of the outer M-Bus network
M-Bus slave port	Network port of the M-Bus slave slave in case of TCP or UDP
M-Bus slave mode (2nd)	Sets the mode of the M-Bus slave (instance 2; TCP or UDP only) or deactivates the interface.
M-Bus slave port (2nd)	Network port of the M-Bus slave (instance 2)
Specific parameters of the wM-Bus*	
wM-Bus frequency	Frequency band for the communication with the wM-Bus meters
wM-Bus network role	Function of the wM-Bus interface. The following mode are available: ▪ *Disabled*: The wM-Bus interface is deactivated. ▪ *Master (Concentrator)*: The wM-Bus interface is used to read out meters. ▪ *Slave (Meter)*: The wM-Bus interface is used to transmit meter data.
wM-Bus mode	Sets the wM-Bus communication mode of the OMS interface (T, S, C or C/T-Mode) or deactivates the interface.
wM-Bus transparent mode	Activates and sets the transparent mode of the wM-Bus communication (Transparent/TCP or Transparent/UDP or Disabled).
wM-Bus transparent port	Network port of the transparent wM-Bus mode
wM-Bus listen	Activates the processing and listing of unconfigured and newly received wM-Bus devices
Show encryption keys	Displays the keys in plain text after saving the list.
Specific parameters of the wM-Bus (channel 2)*	
wM-Bus2 frequency	Frequency band for the communication with the wM-Bus meters (channel 2)
wM-Bus2 mode	Sets the wM-Bus communication mode of the OMS interface (T, S, C or C/T-Mode) or deactivates the interface (channel 2).
wM-Bus2 transparent mode	Activates and sets the transparent mode of the wM-Bus communication (Transparent/TCP or Transparent/UDP or Disabled, channel 2)
wM-Bus2 transparent port	Network port of the transparent wM-Bus mode (channel 2)
Specific parameters of the pulse inputs*	
S0 mode	Sets absolute or relative pulse counting or deactivates the interface.
Specific parameters of the serial interface*	
Serial mode	Sets the operating mode of the serial interface (DLDE, Modbus Slave RTU, Modbus Master RTU, Transparent/TCP or Transparent/UDP, DLMS) or deactivates the interface.
Serial baud rate	Serial communication baud rate
Serial data bits	Serial communication data bits
Serial stop bits	Serial communication stop bits
Serial parity	Serial communication parity
Serial first timeout	Serial communication timeout until first data is received (in ms). In push mode the meter has to be silent for this configured timeout (corresponds to idle time)
Serial idle timeout	Serial communication timeout for detecting the end of communication (in ms)
Serial full timeout	Serial communication timeout (total) for the reception of a data telegram (in ms)
Serial transparent port	Network port for the transparent serial communication

Continued on next page

Table 10 – *Continued from previous page*

Column name	Description
DLDE mode	Procedure of serial DLDE communication: ▪ *Request*: request according to mode A or mode B defined in IEC 62056-21 (static baud rate) ▪ *Request (C-Mode)*: request and handshake according to mode C defined in IEC 62056-21 (static baud rate) ▪ *Push*: reception of cyclically pushed data from the meter
Reply timeout (ms):	Timeout for a response of the meter
Silent interval (ms):	Idle interval between Modbus transmissions
DLMS transparent mode:	Modus for the transparent DLMS proxy
DLMS transparent port:	Network port for the transparent communication via DLMS

*if device is equipped with this interface/function

Table 10: Fields in the Configuration tab

The **Save** button is used to save the configuration. The **Reload** command loads the last saved parameters and resets current changes.

The device is automatically reinitialized by accepting the parameters via the **Save** button.

4.7 Tab WAN

The ***WAN*** tab allows the parametrization of the WAN connection for devices with integrated cellular modem. This is permanently set up when the device is restarted and is kept permanently active.

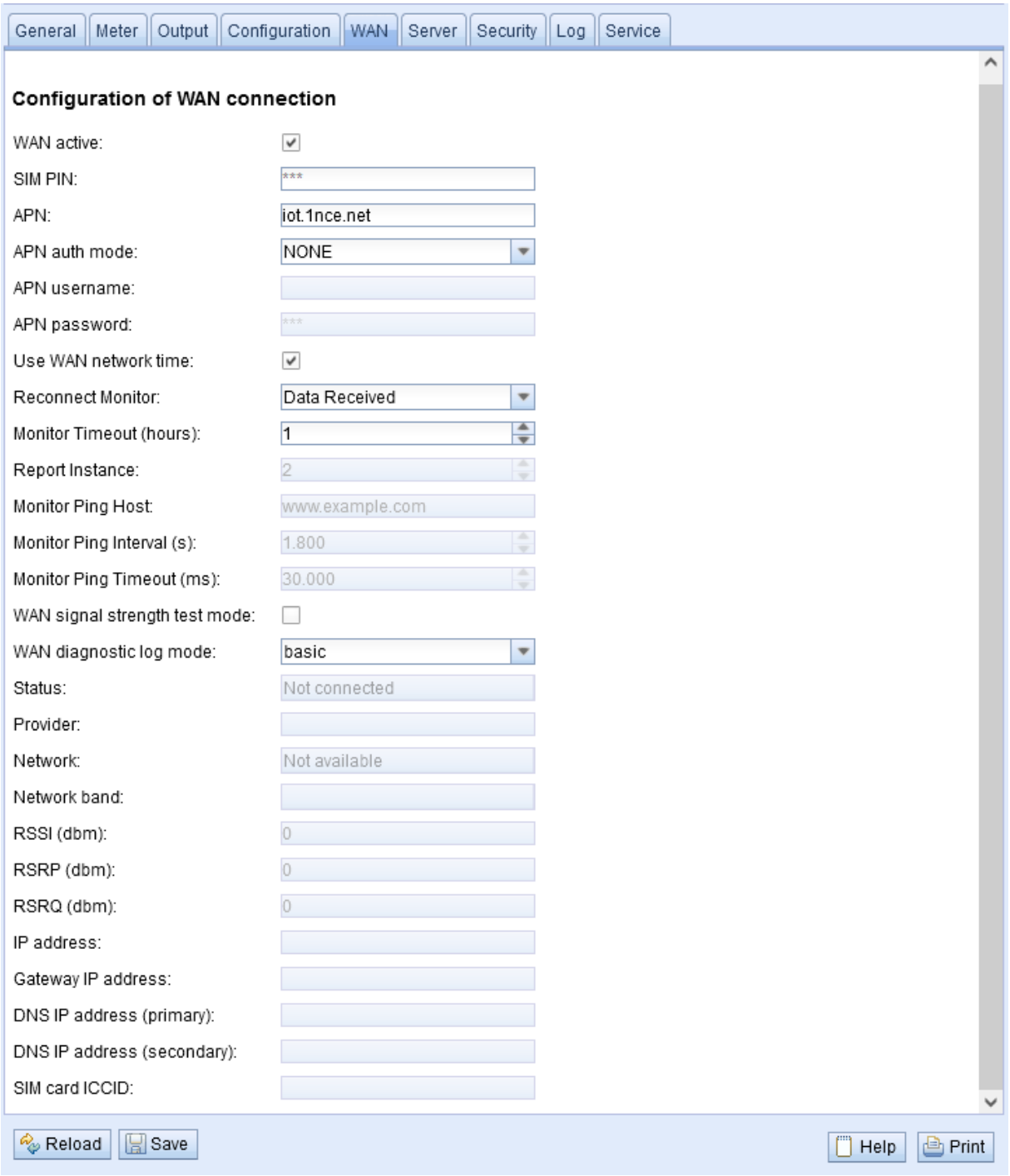


Figure 22: Tab WAN

The following parameters are available:

Column name	Description
WAN active	Activation of the WAN module
SIM PIN	PIN of the SIM card
APN	Name of the access point (APN)
APN auth mode	Authentication mode of the APN
APN username	User name for authentication at the APN
APN password	Password for authentication at the APN
Use WAN network time	Updates the system time when connecting with the radio network. This time is not updated regularly. SNTP (see Table 7) can be used for regular updating.

Continued on next page

Table 11 – *Continued from previous page*

Column name	Description
Reconnect Monitor	Additional monitoring of the radio connection and forced disconnection as well as renewal of the radio connection if the condition is not met. The following modes are available: ▪ *off*: no additional monitoring ▪ *Data Received*: data were received by radio in the indicated time frame ▪ *Any report successful*: an arbitrary report was at least once successful in the indicated time frame ▪ *All reports successful*: all reports were at least once successful in the indicated time frame ▪ *Selected report successful*: the selected report was at least once successful in the indicated time frame ▪ *Test Ping*: the ping host was reached at least once in the indicated time frame. Mind that: – A single echo request is sent. – *Monitor Ping Timeout* can block a readout. Therefore, *Test Ping* should not be used at very high readout frequencies. – The echo requests are sent with a payload of 0 byte, the function requires 28 bytes data volume each for in and out per interval. – The pings are logged in the tab ***General*** if the Log Mode *All* is selected; as successful or as warning if failed due to timeout.
Monitor Timeout (hours)	Interval in hours which is monitored. If the condition of the Reconnect Monitor is not met within this time frame, the WAN connection will be reinitialised. Rationale numbers are also valid here, e. g.: *0.25*.
Report Instance	Report Instance which is monitored if the mode *Selected report successful* is used (otherwise greyed out).
Monitor Ping Host	Host/IP-address to be monitored. An IP address should be configured for the test, not a DNS name. If a DNS name is given, it will be resolved to an IP address during startup and after modifications in the tab ***Configuration*** and, if successful, will only be resolved again after 24 hours. This avoids the consumption of additional data volume by repeated resolution of the DNS name.
Monitor Ping Interval (s)	Interval in which a ping is sent (in s).
Monitor Ping Timeout (ms)	Timeout for the reception of a response (in ms).
WAN signal strength test mode	Sets the WAN interface in a mode to monitor the signal strength to optimize the antenna positions. In this mode, the parameters Provider, Network and the signal indicators (RSSI, RSSQ, RSRQ) are updated at high frequency for all devices. In devices with just one modem channel (see note underneath this table), no data connection exists via the WAN interface in this mode.
WAN diagnostic log mode	Activation of raw data output for the WAN communication in the system log
Status	Status of the WAN connection (connected / not connected)
Provider	Diplays, with WAN connected, the PLMN code or the name of the provider with whom the device is connected. See note underneath this table.
Network	Network technology of the radio connection. See note underneath this table.
Network band	Displays the mobile radio band (frequency band) in use. See note underneath this table.
RSSI (dbm)	Field strength of the cellular network in dBm (-113 to -51 dBm, -114 corresponds to be not connected). See note underneath this table.
RSRP (dbm)	Reference Signal Received Power. See note underneath this table.
RSRQ (dbm)	Reference Signal Received Quality. See note underneath this table.
IP address	IP address in the WAN
Gateway IP address	Remote station in the WAN
DNS IP address (primary)	Primary DNS server for the name resolution
DNS IP address (secondary)	Secondary DNS server for the name resolution
SIM card ICCID	Displays the number/ICCID of the inserted SIM card with active WAN connection

Table 11: Fields in the WAN tab

- Hint with respect to *WAN signal strength test mode*:
  - Updates of the fields Provider, Network, Network band, RSSI, RSSP, RSSQ depend on the device hardware. They are regularly updated in devices with several channels to the modem (MUC.easy^{plus} 4G/NB-IoT). In devices with just one channel to the modem, the values are read only when establishing the connection (MUC.easy^{plus} 2G/3G, MUC.one). For these devices, the test mode can be used to benefit from regular values when the antenna position is to be optimized. This mode should only be activated in case of local connection as there is no data connection in these devices for this mode.
  - Only RSSI, RSSP and RSSQ are updated automatically in the web-based front end. The button **Reload** can be used for updating the remaining parameters.

The necessary parameters for the WAN connection should be provided by the cellular network provider of your SIM card.

- Please check whether the cellular network contract includes the expected quantity of data, otherwise increased costs or a blocking of the SIM card may follow.
- Please check whether the parameters are correct. Incorrect parameters can lead to increased costs or blocking of the SIM card.
- If an invalid PIN is entered, it will be used only once per software startup. Thus, the remaining attempts for entering the PIN are not depleted and a new PIN can be entered via the website.
- Changing the WAN configuration via an active cellular network connection is not recommended, as the device may no longer be accessible after a changed or invalid configuration.

The **Save** button is used to save the configuration. The **Reload** command loads the last saved parameters and resets current changes.

- The device is automatically reinitialized by accepting the parameters via the **Save** button. An existing WAN connection is terminated and re-established.

4.8 Tab Server

The ***Server*** tab allows the parametrization of the data reports to third-party systems. In some data concentrators, the function „Multi Channel Reporting“ (MCR) permits to send reports with meter data to up to 10 different and independent instances (configurations) that can be executed in parallel (siehe Chapter 8).

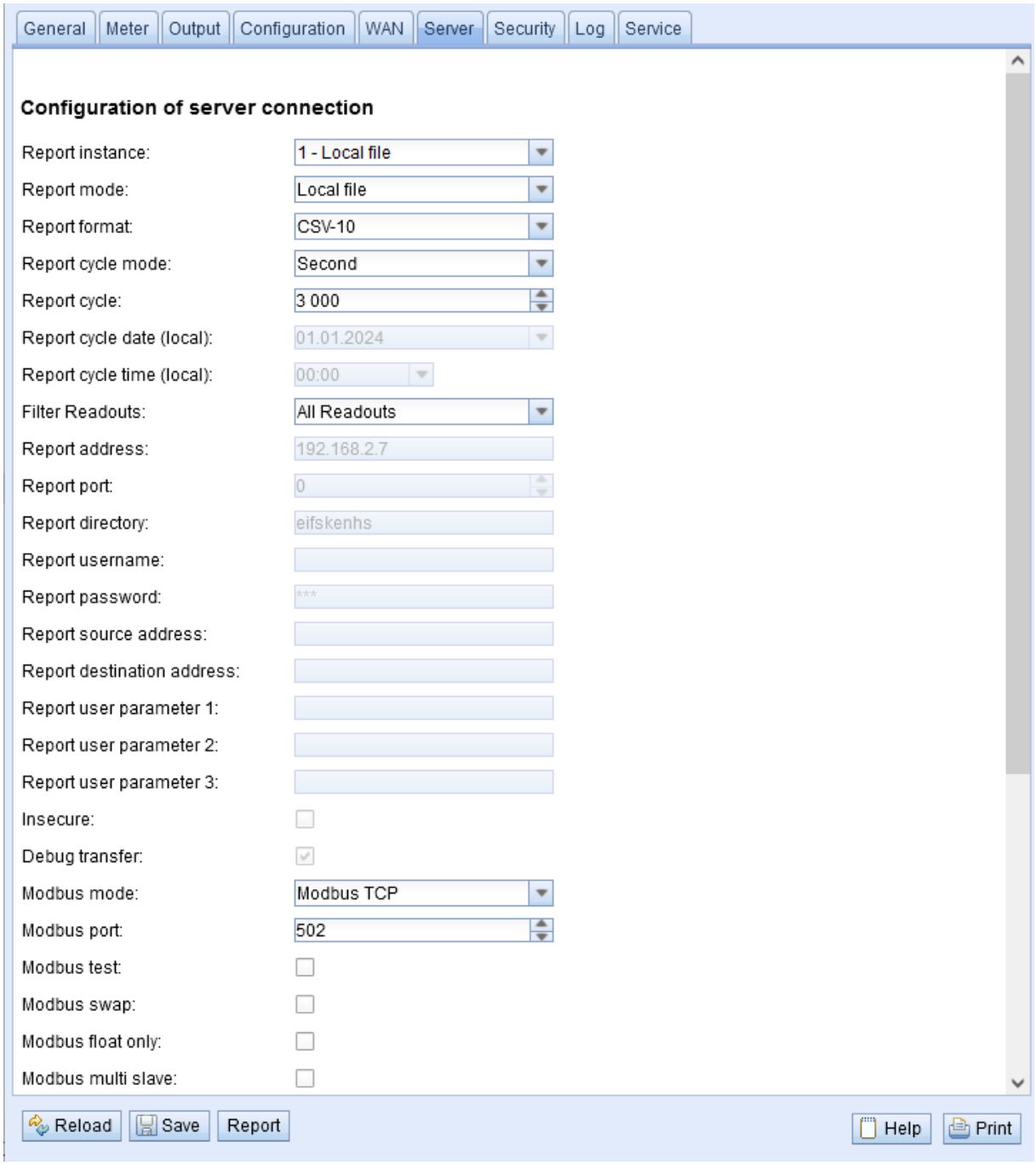


Figure 23: Tab Server

The following parameters are available:

Column name	Description
Parameters for data concentrators with Report functionality	
Report instance	Selection of the respective instance

Continued on next page

Table 12 – *Continued from previous page*

Column name	Description
Report mode	Sets the operating mode of the respective instance or deactivates it. The following modes are available: ▪ *TLS*: active data push via encrypted TCP channel to the specified server ▪ *TCP*: active data push via unencrypted TCP channel to the specified server ▪ *SMTP*: active data push via email to the specified address. The report is in the text of the email. ▪ *SMTP with Attachment*: active data push via email to the specified address. The report is in the attachment to the email, the text of the email is void. ▪ *FTP (client active)*: active file transfer via FTP to the specified server (encrypted or unencrypted). In case of unencrypted FTP, the data connection is established from the server. The files are stored in a specified directory on the server. For a MUC.easy^{plus} results: – File name: *<target path>/MUC_Easy_ID_<ID>_TS_<timestamp>.csv* – Example: */upload/MUC_Easy_ID_6891d0800d89_TS_1372759627.csv* The parameters in angle brackets denote respectively the configured target path, the serial number (ID) of the device and the timestamp (Unix timestamp) at the instant of data transmission. The meter data are transmitted in the CSV format, see Section 8.4.2. ▪ *FTP (client passive)*: active file transfer via FTP to the specified server (encrypted or unencrypted). In case of unencrypted FTP, the data connection is established from the device. The storage location and the naming convention of the files is identical to *FTP (client active)*. ▪ *MQTT*: active data push via MQTT client to the specified server/broker (encrypted or unencrypted) ▪ *Local File*: writing local files to internal memory for later data pull by third party systems (e. g. via FTP, see Section 8.10) ▪ *User*: user-specific report mechanism based on a BASH script (see Section 10.7.2)
Report format	Sets the data format used for the transmission of the respective instance. Several predefined formats are available (see Section 8.4). Further, the format *User* (see Section 8.4.4) can be selected in order to define an own format of the data using a XSLT script (see Section 10.7.1). The format *Systemlog* causes the systemlogs to be transmitted in text form compatible with syslog. The logs can then be transmitted e. g. to a Graylog server monitoring the logs (e. g. from many devices).
Report cycle mode	Format for specifying the report cycle of the respective instance ▪ *Second*: Report cycle is specified in seconds ▪ *Minute*: Report cycle is specified in minutes ▪ *Hour*: Report cycle is specified in hours ▪ *Daily*: daily report at the specified time ▪ *Weekly*: weekly report on the specified weekday and at the specified time ▪ *Monthly*: monthly report on the specified day of the month and at the specified time ▪ *Quarterly*: quarterly report on the specified day and month of the quarter and at the specified time (month 1..3 per quarter) ▪ *Yearly*: yearly report on the specified day and month and at the specified time ▪ *On Readout*: Report will be sent directly after readout. The report interval is identical to the readout interval.
Report cycle	Report cycle of the respective instance (unit according to *Report cycle mode* in seconds, minutes or hours; only for *Report cycle mode* in *Second*, *Minute*, *Hour*). Not active if Report cycle mode is *On Readout*.
Report cycle date (local)	First report day of the respective instance in case of daily to yearly specification of the report cycle, depending on the interval format the entered month is used, the year is not relevant. Not active if Report cycle mode is *On Readout*.
Report cycle time (local)	Report time of the respective instance for daily to annual specification of the report cycle. Not active if Report cycle mode is *On Readout*.
Filter Readouts	Selection if all values, or only the newest, or only the oldest value from a particular time span should to be transmitted in a cyclic report. This is beneficial for frequent readout if a report is requested at short intervals or if the values should also be available for Modbus. The following modes are available: ▪ *All readouts*: all values ▪ *Only newest readout*: only the newest value ▪ *Only oldest readout*: only the oldest value
Report address	Host address of the remote station or mail server (outgoing mail server)
Report port	Network port of the remote station to connect to
Report directory	Path on the remote station
Report username	User name for server access

Continued on next page

Table 12 – *Continued from previous page*

Column name	Description
Report password	Password for server access
Report source address	Address of the sender (Email)
Report destination address	Address of the recipient (Email)
Report user parameter 1	User-specific parameter 1 (parameter for user-specific Report scripts)
Report user parameter 2	User-specific parameter 2 (parameter for user-specific Report scripts)
Report user parameter 3	User-specific parameter 3 (parameter for user-specific Report scripts)
Insecure	Allow insecure encrypted communication by disabling certificate and hostname verification
Debug transfer	Additional logging for transmitting reports in order to investigate more thoroughly problems in the communication with the server.
Parameters for Modbus-Server*	
Modbus mode	Sets the operating mode to Modbus TCP, Modbus UDP or deactivates the service. In operating mode *Modbus TCP*, up to 5 parallel connections from different Modbus TCP masters are accepted.
Modbus port	Network port on which the service is waiting for incoming connections from a remote station (the Modbus TCP client)
Modbus test	Dummy mode for representing the test process data via Modbus
Modbus swap	Changes the word order from MSW first (default) to LSW first (option checked)
Modbus float only	Reduces the Modbus register layout from 10 registers per value to 2 registers per value by only representing the serial number of the meter and the floating point value of the corresponding meter value
Modbus multi slave	Activates the multi-slave feature, where the data of a meter can be accessed as individual virtual Modbus slave using a unique Modbus address
Parameters for BACnet server*	
BACnet active	Activates the BACnet functionality
BACnet config network	Activates a second virtual network interface for the BACnet service
BACnet IP	IP address of the second virtual network interface for BACnet
BACnet netmask	Subnet mask of the second virtual network interface for BACnet
BACnet broadcast	Broadcast address of the second virtual network interface for BACnet
BACnet BBMD	IP address of a BACnet Broadcast Management Device (BBMD) for routing across local network boundaries
BACnet port	UDP port number of the BACnet service (default port: 47808)
BACnet device ID	ID number of the BACnet device
BACnet device name	Device name of the BACnet device
BACnet location	Location information of the BACnet device

*if device is equipped with this interface/function

Table 12: Fields in the Server tab

Depending on the operating mode of the server interface, individual parameters required for the configuration are enabled.

- When using encrypted connections (TLS, MQTTS, SMTPS, FTPS), the server certificate or the Root CA certificate for the server must be saved on the device. This is achieved by **Config Import** of the certificates in PEM format in the tab ***Service***.

The **Save** button is used to save the configuration. The **Reload** command loads the last saved parameters and resets current changes. The **Report** button allows immediate transmission of the data previously read out.

- Setting the parameters via the button **Save** causes a reinitialization of the device.
- Mind a correct system time before activating the report if Report cycle mode is not *On Readout*. If the system time is synchronized later, e. g. via a SNTP service, gaps may occur in the log. These gaps may cause empty files to be transmitted to the server.

4.9 Tab Security

The ***Security*** tab allows the parametrization of the network services by the device.

Figure 24: Tab Security

The following parameters are available:

Column name	Description
HTTP server active	Activation of the internal HTTP server of the device. Deactivation is possible only by selecting HTTPS.
HTTPS server active	Activation of the internal HTTPS server of the device. Deactivation is possible only by selecting HTTP.
FTP server active	Activates the internal HTTP server of the device. If deactivated, there is no FTP access to the device.
SSH server active	Activates the internal SSH server of the device (administrative access).
Network discovery active	Activates the internal discovery server of the device. If deactivated, the device is no longer displayed in the Netdiscover tool (see Chapter 3)
Network discovery password	Password for setting the network parameters via the Netdiscover tool
Modbus server active	Modbus server active, read-only, depending on the ***Server*** tab
BACnet server active	BACnet server active, read-only, depending on the ***Server*** tab

Table 13: Fields in the Security tab

The **Save** button is used to save the configuration. The **Reload** command loads the last saved parameters and resets current changes.

The device is automatically reinitialized by accepting the parameters via the **Save** button. An existing WAN connection is terminated and re-established.

4.10 Tab User

The ***User*** tab allows the parametrization of different users and their permissions for the website.

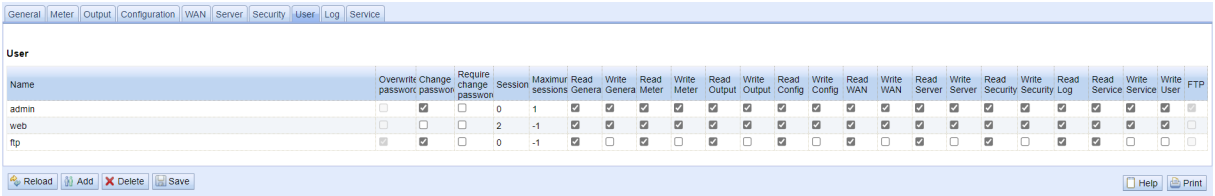


Figure 25: Tab User

The following users are preconfigured upon delivery:

User name	Password	Comments
admin	admin	Administrative user with full access to all services of the device (HTTP, FTP, SSH, IP configuration).

Continued on next page

Table 14 – *Continued from previous page*

User name	Password	Comments
web	web	Default user for the website. If a user with this name and password exists, the web interface automatically logs in using these credentials. Otherwise, the user is prompted to enter individual credentials. Per default, this user has full access to the website of the device.
ftp	ftp	User for unencrypted FTP access restricted to the log in path */ext/Log*

Table 14: User accounts upon delivery

On the website, the existing configuration is shown in a table and can be changed respectively:

Column name	Description
Name	User name
Overwrite password	It is set if a (new) password has been set for the user in the editing dialogue.
Change Password	Setting whether the user is allowed to change his password
Require change Password	Setting whether the user has to change his password at the next login
Sessions	Number of currently active sessions of this user
Maximum sessions	Setting how often the user may be logged in at the same time in parallel (-1=unlimited)
Read General	Read permission to the General tab
Write General	Write permission to the General tab
Read Meter	Read permission to the Meter tab
Write Meter	Write permission to the Meter tab
Read Output	Read permission to the Output tab
Write Output	Write permission to the Output tab
Read Config	Read permission to the Configuration tab
Write Config	Write permission to the Configuration tab
Read WAN	Read permission to the WAN tab
Write WAN	Write permission to the WAN tab
Read Server	Read permission to the Server tab
Write Server	Write permission to the Server tab
Read Security	Read permission to the Security tab
Write Security	Write permission to the Security tab
Read Log	Read permission to the Log tab
Read Service	Read permission to the Service tab
Write Service	Write permission to the Service tab
Write User	Read and write permission to the User tab
FTP	Permission of the user to log in via FTP (maximum 2 users)

Table 15: Fields in the User tab

The user configuration can be changed with the buttons at the bottom or via the context menu. Except from the *admin* user, other users can be created, deleted or changed.

The users in the list can be selected by a simple mouse click. A range can be selected with the ⟨**SHIFT**⟩ key held down, or multiple users can be selected (individually) with the ⟨**CTRL**⟩ key held down.

The **Reload** command loads the last saved parameters and resets current changes.

When write permission to a tab is granted, read permission is also granted automatically.

- The *admin* user cannot be changed or deleted in the user configuration. The administrator password can only be changed by using the ***Change password*** button when the *admin* user is logged in.
- If the administrator password is lost, the device can only be reset to factory defaults by solvimus GmbH as file access on the device is limited for safety reasons. When resetting, all configuration data and meter data are lost.
- Only the *admin* user has full access to the file system of the device via encrypted FTP (SFTP). The second FTP user can access only the path */ext/Log*, even without encryption.

New users can be added via the **Add** button or via the context menu item with the same name. The following dialogue will open:

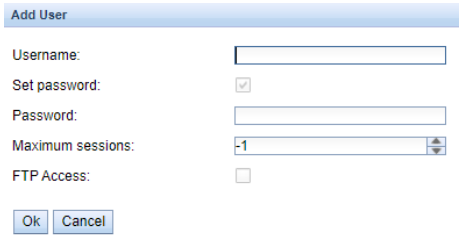


Figure 26: Input dialogue for adding new users

In addition to the user name and password, you can specify how often a user may log in at the same time (-1=unlimited). Besides the user *admin*, another user can have FTP access to the device. The unencrypted FTP access only allows access to the log data on the device (path: */ext/Log*). This permission can only be enabled at the time the user is created.

- A separate FTP user (e. g. *ftp*) allows a remote client to download the stored log data (manually or automatically), whithout having access to other services or data on the device.

For reconfiguring an already existing user, the editing dialogue can be opened by double clicking its entry or via the context menu item ***Edit***. This dialogue has the same structure as the dialogue for adding a user. For resetting the password of an existing user, the **Set Password** checkbox has to be set. If the **Set Password** checkbox is not set, the user password is not changed or reset during this configuration process. A user password cannot be read.

The configuration can be finished with the **Ok** button or cancelled with the **Cancel** button.

The permissions of a user are directly set in the user list. If a user has write permission to a tab, the user automatically gets the permission to see the tab (read access).

Using the button **Delete** or the context menu item with the same name, all selected users (with the exception of the *admin* user) can be deleted.

The **Save** button is used to save the user configuration.

4.11 Tab Log

The ***Log*** tab allows accessing log information and status outputs. That facilitates the analysis of the behaviour and troubleshooting.

- The extent of the log entries depends largely on the settings in the **Log mode** field in the ***General*** tab (see Section 4.3).
- For viewing the raw data logs of the meter interfaces, the **Raw data log** field in the ***Configuration*** tab must be activated (see Section 4.6).

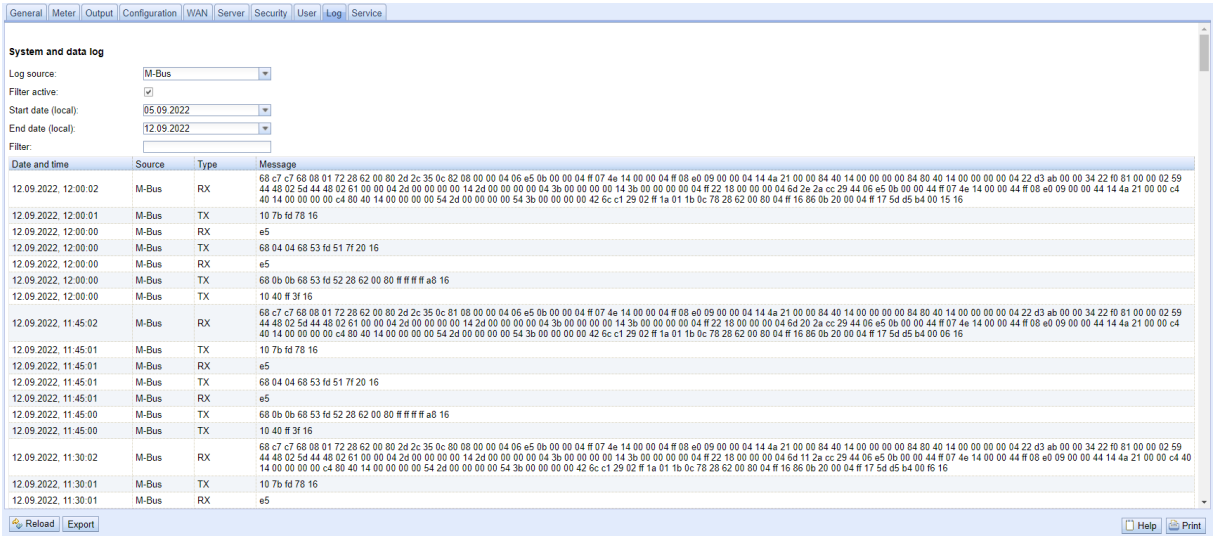


Figure 27: Tab Log

The following parameters are available:

Column name	Description
Log source	Selects the source of the log entries. ▪ *System log*: Show the log entries of the system (Linux) and the application ▪ *Application*: Show the log entries of the application ▪ *M-Bus*: Show the raw data of the M-Bus interface (if **Raw data log** is active in the ***Configuration*** tab) ▪ *wM-Bus*: Show the raw data of the wM-Bus interface (if **Raw data log** is active in the ***Configuration*** tab) ▪ *DLDE*: Show the raw data of the DLDE interface (if **Raw data log** is active in the ***Configuration*** tab) ▪ *Modbus Master RTU*: Show the raw data of the Modbus Master RTU interface (if **Raw data log** is active in the ***Configuration*** tab) ▪ *Modbus Slave RTU*: Show the raw data of the Modbus Slave RTU interface (if **Raw data log** is active in the ***Configuration*** tab)
Filter active	Enables filtering by time range and string expression
Start date (local)	Start date of the time range for the log entries
End date (local)	End date of the time range for the log entries
Filter	String expression used for filtering the log (search for keyword or regular expression in the **Message** column)

Table 16: Fields in the Log tab

The **Reload** button updates the log entries according to **Log source** and the filter settings (including the time range).

- ✔ Using the keyword *serial=* allows filtering for one meter's secondary ID in the raw data log, e. g. *serial=12345678*. Only telegrams from this meter are shown then.
- ✔ Depending of the extent of the log entries, it may take some time to generate the table.
- ✔ The filter settings are kept when changing between tabs. So, coming back to this tab, the old filter is still active. This will ease the troubleshooting but may cause increased load times for extensive logs.
- ℹ If no log entries are shown, please check the filter settings. If necessary, extend the specified time range, reset the filter or deactivate it.
- ℹ The number of log entries shown is limited to 500. Use the filter or the time range to reduce the entries.

The **Export** button generates a CSV file containing all log entries matching the filter and time range for downloading it. This download may take some time depending on the size of the log.

4.12 Tab Service

The ***Service*** tab allows maintenance and provides related information or functions:

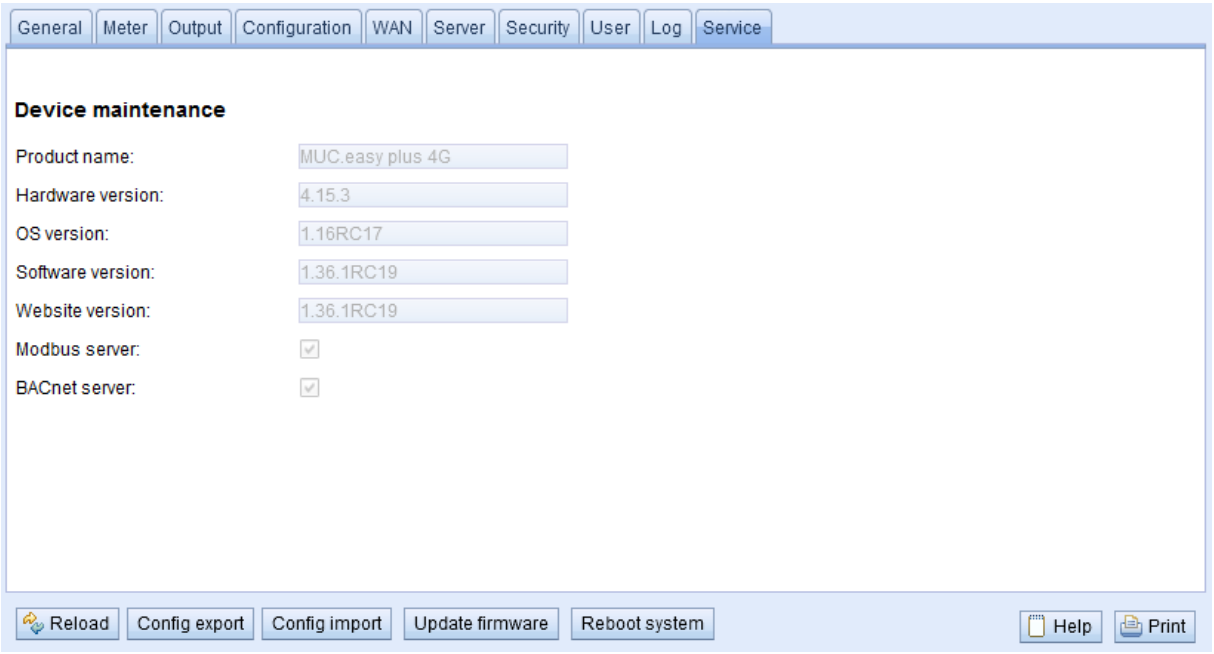


Figure 28: Tab Service

The following parameters are available:

Column name	Description
Product name	Product name
Hardware version	Version of the hardware
OS version	Version of the operating system
Software version	Version of the software
Website version	Version of the website
M-Bus load profile	If available and ticked: licence for load profile active
Modbus server	If available and ticked: licence for Modbus server active
BACnet server	If available and ticked: licence for BACnet server active
M-Bus slave	If available and ticked: licence for M-Bus slave active

Table 17: Fields in the Service tab

The values are updated using the **Reload** button.

The **Config export** and **Config import** buttons allow to download the configuration from the device or upload the configuration to the device.

When exporting the configuration, a selection dialogue permits choosing which data is downloaded from the device:

- Certificates
- Device configuration
- Network configuration
- Device name
- Meter configuration

✔ The network configuration and the device name are part of the device configuration. If the device configuration is to be transferred to another device, it is recommended not to export the network configuration and the device name. Usually these should not be transferred to other devices.

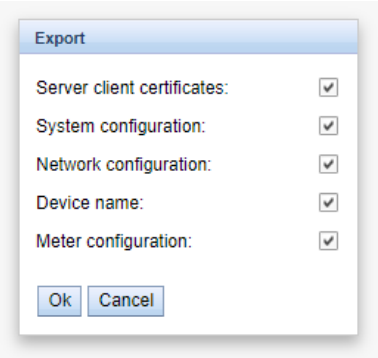


Figure 29: Options for exporting the configuration

The configuration is downloaded as a **. tar. gz* file. This compressed archive is an excerpt from the file system of the device. It can be stored as a backup or modified for uploading it later to the same or another device. It is useful for transferring a valid configuration to a replacement device or for commissioning many similar devices (see Section 3.7).

When importing the configuration, a file selection dialogue comes up for selecting the corresponding **. tar. gz* file.

Using the **Update firmware** button opens a file selection dialogue as well. An update file can be selected here. The solvimus GmbH provides updates as **. enc* files on a regular basis. These files can then be uploaded to the device. After successfully uploading them, the update process is started automatically and the device is then restarted. An alternative procedure for updating the firmware is described in Section 3.7.

The device can be restarted using the **Reboot system** button. All internal processes are shut down and re-initialized after the restart. Meter data pending to be sent via the WAN interface is transferred after a restart. Use this button if you intend to manually modify the configuration via FTP(S) or after a manual update.

4.13 Print page

The **Print** button (see Figure 16, bottom right) can be used for getting an entire overview of the configuration or for exporting it via the clipboard. The website generates an additional browser window containing all available configured parameters and meters according to the access rights. The print page is automatically closed after a user has logged out from the website (at the top right of the web-based front end, if not already closed).

- ✔ The meter list displayed is also suitable for inserting it into a spreadsheet.

MUC.easy^{plus}

Configuration

General configuration	
Device name:	MUC.easy plus 4G
Serial number:	6891d0803d4d
DHCP:	on
IP address:	192.168.3.21
Subnet mask:	255.255.255.0
Gateway IP address:	192.168.3.254
DNS IP address (primary):	192.168.1.161
DNS IP address (secondary):	192.168.1.162
VPN:	0
Free space log (kB):	2237116
Free space Flash (kB):	114670
System date (local):	Thu Nov 02 2023 10:50:00 GMT+0100 (Mitteleuropäische Normalzeit)
SNTP server:	pool.ntp.org
Log mode:	All
Configuration of meter interfaces	
Readout cycle mode:	Quarterly
Readout cycle:	900
Readout cycle date (local):	Wed Nov 01 2023 00:00:00 GMT+0100 (Mitteleuropäische Normalzeit)
Description mode:	Standard
Maximum device count:	500
Maximum value count:	0
Store meter values:	Automatic
Raw log active:	on
M-Bus mode:	Master
M-Bus addressing:	Secondary scan
Primary start address:	0
Primary final address:	250
Secondary address mask:	FFFFFFFF
M-Bus baud rate:	2400
M-Bus timeout (ms):	500
M-Bus idle timeout (ms):	100
M-Bus full timeout (ms):	10000
M-Bus request mode:	Standard
M-Bus reset mode:	Standard
M-Bus max. multipage:	3
M-Bus transparent port:	5000

Figure 30: Print page of the device (excerpt), here the example of a MUC.easy^{plus}

4.14 Troubleshooting the front end

Using a standard web browser for accessing the web server running on the device is an easy and intuitive way to manage the device. Nevertheless, impairments or unwanted behaviour may occur.

- ✔ One potential error source is the browser cache, especially if several devices are operated with the same IP address or after an update has been applied. To eliminate this error source, first terminate the web session by using the **Logout** button and then completely reload the website. Depending on the browser, this is initiated using a shortcut, e. g. ⟨**CTRL+F5**⟩ or ⟨**CTRL+R**⟩.

4.14.1 Website or front end cannot be accessed

The website cannot be loaded or the error message „webservice not available“ appears.

Inspect the IP settings of the device and of your computer. The IP addresses should be in the same subnet or a route must be set up. If possible, change the IP addresses accordingly. Please ask your administrator. Alternatively, you can also use DHCP to assign a valid IP address (see Tool Netdiscover in Chapter 3). Below there are two examples of a valid configuration:

- Device: 192.168.1.101 (default IP), subnet mask: 255.255.255.0 → PC: 192.168.1.xxx (xxx = 0-254, except 101 and other already used IP addresses), recommended for direct connection 1:1 device and PC
- PC: 192.168.178.21, subnet mask: 255.255.255.0 → device: 192.168.178.xxx (xxx = 0-254, except 1, 21, 254 and other already used IP addresses), typical for connection to a router in the home network

Please check whether the device is listed in the Netdiscover tool (see Chapter 3). Please check the connectivity in general via a ping test integrated in the Netdiscover tool.

Please check whether a firewall is blocking the data transmission or whether the routing is configured accordingly. Please ask your administrator.

In the case of an HTTPS connection, the browser may block the access under certain circumstances. Please confirm the provided certificate in the browser or „trust“ the website and its certificate if you are sure to access the correct device.

If errors could not be eliminated, please contact our customer support:
E-Mail: support@solvimus.de
Phone: +49 3677 7613065

4.14.2 Login to website is refused

Please check the user settings and permissions for the website as well as the user credentials.

There may be another user already logged in while the number of active sessions is limited. Then the login is denied. Please check the user credentials and the number of active sessions in the ***User*** tab.

If errors could not be eliminated, please contact our customer support:
E-Mail: support@solvimus.de
Phone: +49 3677 7613065

4.14.3 All input fields or buttons are greyed out

Buttons greyed out are indicating that write permission is not granted. Please note that only one logged in user gets write access.

Please check whether another session is already active. This can also occur if a browser window is just closed without logging out first. The session is then still active for a short time. Please log out again and wait about one minute. Please check the user's permissions and the number of active sessions in the ***User*** tab.

Please check whether the user has write permissions.

If errors could not be eliminated, please contact our customer support:
E-Mail: support@solvimus.de
Phone: +49 3677 7613065

4.14.4 Not all tabs are visible

Please check the user's read permissions. Only those tabs are available with granted read permission to the user. Please check the user's permissions in the ***User*** tab.

If errors could not be eliminated, please contact our customer support:
E-Mail: support@solvimus.de
Phone: +49 3677 7613065

4.14.5 Export of the meter readings of one/several meters is empty

Meter readings are only stored when a report is active in order to optimize the memory. Please check whether a report is active in the ***Server*** tab.

Please check the time range for the export. The chosen time of the report has to start before a valid readout. For example, for exporting the readout from 29/09/2020 01:15 pm, the time for export should be set to

29/09/2020 01:10 pm. The report then contains all readouts starting from 01:10 pm until the end of the **Report cycle** configured for instance 1 in the ***Server*** tab or 15 minutes.

If errors could not be eliminated, please contact our customer support:
E-Mail: support@solvimus.de
Phone: +49 3677 7613065

4.14.6 The Log is empty

Please check the filter settings. If no filter is active, entries should always be available for the **Log source** *System log*. If not, this indicates a misconfiguration on system level. This can be resolved by calling the command *solcmd config-partitions* in the SSH console (see Section 10.1.2).

Please check whether the raw data log for the interfaces is active (see ***Configuration*** tab). Only then the raw data for the **Log source**, e. g. *M-Bus*, will be generated.

If errors could not be eliminated, please contact our customer support:
E-Mail: support@solvimus.de
Phone: +49 3677 7613065

5 Reading meters via M-Bus

5.1 General information

A widely used interface for the automated meter reading is the wired M-Bus (Meter-Bus). This was originally specified in EN 1434-3. It was then moved to a separate standard EN 13757:

- EN 13757-2 Communication systems for meters - Part 2: Wired M-Bus communication
- EN 13757-3 Communication systems for meters - Part 3: Application protocols
- EN 13757-7 Communication systems for meters - Part 7: Transport and security services

Originally developed for heat meters, the M-Bus is now available for all types of consumption meters as well as sensors and actuators. Thus, it is very important for reading out consumption data.

Fundamental features and advantages of the M-Bus are:

- The M-Bus is a digital interface for the electronic meter reading.
- All consumption meters in a building/property can be operated and read via a single cable.
- All consumption meters are individually addressable.
- The readout is protected against transmission errors and is very robust.
- The data is machine-readable and therefore easy to process.
- The data is self-describing.
- High readout rates are possible.
- The M-Bus is manufacturer independent, there is a wide range of devices.

5.2 Signalling on the M-Bus

The M-Bus is a single master multiple slave bus. Therefore, a single bus master controls the bus and the data traffic on the bus. Several slaves, i.e. meters, can be connected to the bus.

A second physical master is not allowed on the M-Bus.

On a physical level, the M-Bus uses voltage and current modulation to transmit data. The master transmits telegrams by modulating the bus voltage, the slave transmits telegrams by modulating the current through the bus. This is shown schematically in the following figure (values of current and voltage may deviate):

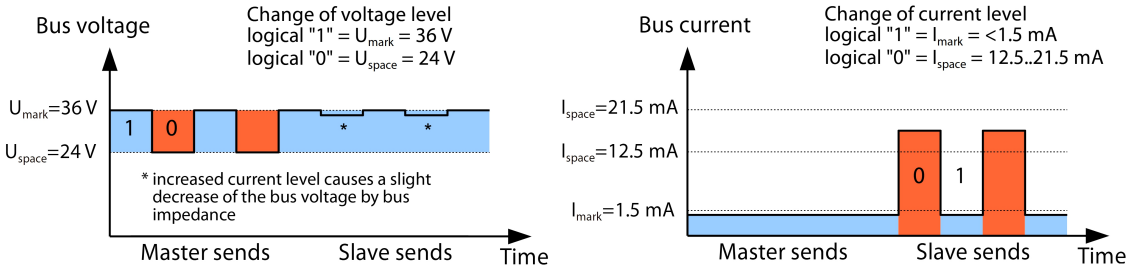


Figure 31: Signalling on the M-Bus

The M-Bus follows the principle of request-response, i. e. the master initiates the communication by a request/command which is then answered/confirmed by the slave. Spontaneous data transmission on the part of the slaves is not allowed.

Certain terms are used in the M-Bus standard. The basics of communication are taken from IEC 60870-5-101. Key terms are explained in the table below:

Term	Description
ACK	ACKnowledge, confirmation of a command, transmitted over the M-Bus as a single character telegram with content 0xE5.
Application reset	Reset of the application layer, command to reset the meter to the default state and to reset the meter for consecutive telegrams (multipaging).
Broadcast	Broadcast, command or request is sent to all slaves, special addresses 0xFE and 0xFF are used.
C field	Command field, code that indicates the direction in which a telegram is exchanged and the meaning of the telegram.
Checksum	Check number for checking transmission errors, the checksum the M-Bus uses, results from the addition of the transmitted data (without telegram header, up to checksum).
Single character	One of the three telegram formats the M-Bus uses with a length of exactly 1 byte, telegram header and end, consisting of checksum and 0x16, are not present, used on the M-Bus for ACK
FCB	Frame Count Bit, bit in the C field, which is alternately set to 1 or 0 in consecutive telegrams, consecutive telegrams can be retrieved when the bit changes in the request.
I_{mark}	Transmit current of the slave at logical 1, usually 1 UL.
I_{space}	Transmit current of the slave at logical 0, usually 12.5-21.5 mA.
Short frame	One of the three telegram formats the M-Bus uses with a length of exactly 5 bytes, is only sent from the master to the slave (e. g. commands and instructions), the telegram header is 0x10 and the telegram ends with checksum and 0x16.
Long frame	One of the three telegram formats the M-Bus uses with a variable length, the telegram header consists of 0x68 LL LL 0x68 (LL is the length of the telegram in each case), the telegram ends with checksum and 0x16.
Multipaging	M-Bus method of distributing large amounts of data into several logically consecutive telegrams, use of the FCB for sequence control.
Primary address	M-Bus Link layer Address, this is used to address the requests/commands, address range 0-250, special addresses 253 (0xFD), 254 (0xFE) and 255 (0xFF).
REQ_UD2	REQuest User Data type 2, request for consumption data, transmitted over the M-Bus by the master as a short frame telegram.
RSP_UD	ReSPond User Data, response of the meter to a request for data, transmitted over the M-Bus by the slave as a long frame telegram.
Secondary address	Worldwide unique identification number of the meter, consisting of manufacturer code, 8-digit serial number, medium ID and version number.
Slave select	Procedure for extending the address space to the secondary address of the meter, use of the SND_UD for selecting the meter via the application layer, then selected meter can be addressed via special address 0xFD.
Unit load	Defined idle current that a meter may draw from the M-Bus, according to the standard 1 UL=1.5 mA.
SND_NKE	Send Link Reset, initialization command to the slave (reset FCB bit and selection), transmitted by the master as a short frame telegram on the M-Bus.
SND_UD	SeND User data, sending data or commands to the meter, transmitted by the master as a long frame telegram on the M-Bus.
U_{mark}	Mark voltage, upper voltage of the M-Bus signals at the master, representation of the logical 1, idle state, usually 24-42 V.
U_{space}	Space voltage, lower voltage of the M-Bus signals at the master, representation of the logical 0, usually 12-30 V.
UL	Unit of unit load (see above)

Table 18: M-Bus specific terms

5.3 Configuration of the interface on the web-based front end

5.3.1 M-Bus mode

The parameter **M-Bus mode** in the ***Configuration*** tab activates the M-Bus interface and defines the fundamental functionality. The following modes are available:

- *Disabled*
- *Master*
- *Transparent/TCP*
- *Transparent/UDP*
- *Master & Transparent/TCP*

The *Transparent* modes allow the access to the physics of the M-Bus interface via a TCP or UDP port. The data stream is forwarded from the M-Bus interface to an IP interface (network (LAN) or cellular radio (WAN)).

The device then works in a way similar to an Ethernet-M-Bus converter or even to a cellular router with an M-Bus interface. The network port to be used is defined in the parameter **M-Bus transparent port**.

- The transparent mode allows direct communication with meters via the M-Bus interface. This requires appropriate M-Bus software on the control system (host system). The device provides the physical connection. This allows to transfer any kind of data with the meter and to use manufacturer specific protocols.

The mode *Master & Transparent/TCP* allows a combination of the transparent transmission and the Master capability of the device. In the absence of a client to a transparent TCP port, the M-Bus master uses the interface and reads out the meters according to the configuration in the mode *Master*. Once a client connects to the TCP port, it gets exclusive access to the interface as in the mode *Transparent/TCP*. A readout of meters or a scan of the M-Bus by the device is not possible as long as a client is connected. A readout fails if configured in this time. Once the client disrupts the connection, the interface is once again run by the M-Bus master, and meters are read out. An inactive connection to the transparent port is closed after 60 seconds in order to rule out a jamming of the M-Bus by open connections. In this mode, a client should assure that the connection is unblocked after usage. As an initiated readout of a meter is first completed upon connection by a client, a larger timeout is recommended for the first communication by the client when establishing the connection ($\geq$5 seconds).

5.3.2 Addressing, scanning and scan range

The M-Bus differentiates between primary addressing and secondary addressing. The M-Bus interface allows also mixed addressing. Meters can be searched first using primary addressing, and a subsequent scan can detect meters using secondary addressing.

The primary address is used for access control on link layer level. It is the basis of communication between master and slaves on the M-Bus and is used for communication in every telegram except the single character frame. The secondary address is an extension of the addressing and additionally controls the access on application layer level.

The valid address range for the primary addresses is 0-250, whereby the address *0* is a special case. According to the standard, only unconfigured meters (ex works) are allowed to have it. The address *253* is a special address used for the secondary addressing, the addresses *254* and *255* are used for the broadcast with and without response. The addresses *251* and *252* are reserved.

The secondary address consists of 4 parts. These are the *secondary ID* (an 8-digit decimal number), the *manufacturer ID* (value of 0-65535), the *medium ID* (value of 0-255), and the *version number* (value of 0-255). Thus, the address space includes theoretically $115.19*10^{15}$ unique values.

- The *manufacturer ID* can be converted to a manufacturer code maintained by the *DLMS User Association*. An overview can be found here: www.dlms.com/flag-id/flag-id-list

In case of primary addressing, this slave responds whose primary address matches the address in the request. This allows a simple and quick communication.

- If the primary address is not unique, primary addressing will cause collisions and communication may be disturbed. Several slaves are then responding at the same time.

Secondary addressing, on the other hand, uses a so-called selection (slave select) on the basis of the secondary address. This selection allows addressing of a meter with a matching secondary via the primary address *253*. The non-matching meters are deselected in the same step. Therefore, the process is more complex since a selection with confirmation is required additonally. Communication takes a longer time. However, the address space is much larger. Collisions do not occur, and more than 250 meters can be addressed on one bus system. In addition, commissioning is faster because not every meter has to be configured to a unique primary address.

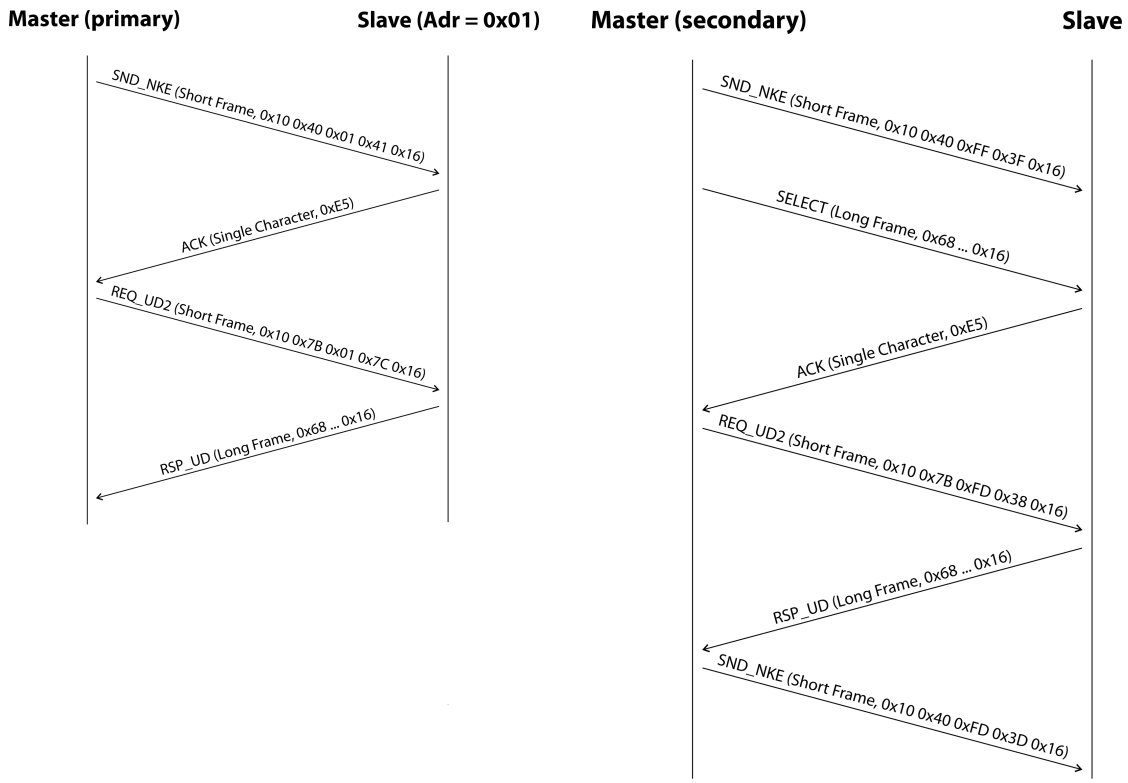


Figure 32: Example of primary and secondary addressing in comparison

Secondary addressing is supporting wildcards. For example, this allows using the 8-digit *secondary ID* for selection only. The other parts are masked with the placeholder 0xFF (255) or 0xFFFF (65535). Individual digits of the *secondary ID* can be masked with 0xF (16) as well.

- The M-Bus uses the BCD notation for the *secondary ID*. The 8-digit decimal number is encoded by an 8-digit hexadecimal number. Per each digit the characters A-F can be mapped to special features, but only the *F* is used as a placeholder at the respective digit.

The placeholders are the basis of the secondary scan. This divides the secondary address space piece by piece using the placeholders and checks whether there are meters in the respective part. If so, this part is further subdivided until there is at most only one meter per part or further subdivision is not possible. The common procedure here is to mask the *manufacturer ID*, *medium ID* and *version number* and to scan the 8-digit number range of the *secondary ID* only.

The range 00000000-99999999 is divided by sending the selection to *0FFFFFFF*, i. e. selecting all meters with a *0* at the first digit of the *secondary ID*. A request is then sent to the selected meters using the primary address *253*. If no response is received, no meter is in this range. So, the least significant, unmasked digit can then be incremented and the process continues with *1FFFFFFF*. If you get an undisturbed response, there is only one meter in this range. This meter is found here and could be registered. The process will then continue with the next step by incrementing the least significant, unmasked digit. If a disturbed response or collision is received, the process switches to the next, still masked digit and runs it from *0* to *9*. It is difficult to estimate what time a secondary scan will take in advance. There is a variability of the process depending on the meters and the distribution of the *secondary ID* in the address space.

Primary scan, in contrast, is very direct and determinate. Every primary address is requested and depending on a valid answer a meter is then registered or not. Thus, 250 requests are always necessary for a complete scan.

The parameter **Primary start address** and **Primary final address** in the ***Configuration*** tab limit the primary scan by specifying the start and end. The parameter **Secondary address mask** is used to mask the *secondary ID* for limiting the scan to a certain address range. For example, a mask *33FFFFFF* limits the scan to all meters having a *secondary ID* starting with *33*.

5.3.3 M-Bus baud rate

The parameter **M-Bus baud rate** in the ***Configuration*** tab is used to configure the bit presentation on the M-Bus interface. The baud rate essentially determines the speed of the data transmission.

- ✔ M-Bus usually uses 2400 bps. Other common baud rates are 300 bps and 9600 bps. Many meters detect the baud rate automatically.
- ✔ The other parameters for the bit presentation on the M-Bus interface are fixed to 8 data bits, even parity and 1 stop bit (8-E-1).

5.3.4 M-Bus timeouts

The M-Bus interface comes with three different timeouts: **M-Bus timeout**, **M-Bus idle timeout** as well as **M-Bus full timeout** (in transparent mode **M-Bus idle timeout** only). These can be parameterized in the ***Configuration*** tab.

The **M-Bus idle timeout** specifies how long the M-Bus interface must be „idle“, i. e. no data is sent/received, for detecting the end of a telegram (end of communication). It is mainly used for framing the packets of the M-Bus data stream, i. e. the assignment of incoming data to a logical unit (data packet).

The **M-Bus timeout** specifies how long the device is waiting for a response from the meter. If no data is received within this time after the request, the readout attempt is aborted.

The **M-Bus full timeout** specifies how long the device will accept incoming data. The reception is then aborted and the data is processed. This parameter also terminates reception if the **M-Bus idle timeout** is not reached because data is continuously received (without idle state, e. g. in case of faults).

5.3.5 M-Bus request mode

By default, the command REQ_UD2 is send from the master to the meter for reding it out. This is answered by the meter with the RSP_UD, which usually contains the meter data (consumption data).

In addition, the parameter **M-Bus request mode** in the ***Configuration*** tab can be used to explicitly define the requested data before the actual readout. Devices from solvimus GmbH can send a so-called global readout request to the meter before the actual request. A SND_UD is sent to the meter for this purpose. The user data then consists of only one or two characters. There are two implementations with the same functionality, depending on the manufacturer one or the other is supported:

- User data consisting of 2 Byte: DIF=0x7F, VIF=0x7E → **M-Bus request mode** *Extended 1*
- User data consisting of 1 Byte: DIF=0x7F → **M-Bus request mode** *Extended 2*

- ✔ This command is usually not necessary, because all meter values are transmitted by default using the normal request.
- ℹ Using this functionality may cause a change in the structure of the meter data.

5.3.6 M-Bus reset mode

The M-Bus there uses different variants and applications of a reset. A distinction is made between:

- Link layer reset → SND_NKE
- Application layer reset → Application reset using SND_UD

According to EN 13757, the link layer reset is only used for initializing the communication sequence on the link layer. Therefore, it resets the selection based on the secondary address, deselects the meter, and also resets the FCB mechanism (see Section 5.3.7).

The application layer reset, on the other hand, resets the application in the meter (or its communication application).

The parameter **M-Bus reset mode** in the ***Configuration*** tab can be used to select the variants and addressing of the resets. The resets are then sent at the beginning of a scan procedure and before each readout of a meter:

- *None*: Neither a link layer reset nor an application layer reset is sent.
- *Standard*: A link layer reset is sent to the broadcast address 0xFF and, in the case of primary addressing, also to the respective primary address.

- *Extended 1*: A link layer reset is explicitly sent to the selection address 0xFD before the link layer resets of the *Standard* mode.
- *Extended 2*: After the link layer reset to the selection address 0xFD, an application layer reset is sent to the broadcast address 0xFF. This is followed by the link layer resets of the *Standard* mode.

5.3.7 M-Bus multipaging

If the data of a meter do not fit into a single telegram (maximum 255 bytes user data), there is the possibility to split these data into several logically related, consecutive telegrams. The FCB mechanism according to IEC 60870-5-2 is used by the readout sequence. The solvimus GmbH calls this process „multipaging“.

In order to request possibly existing telegrams from the meter, the master has to toggle the FCB with each new request REQ_UD2. The meter then replies with the next telegram. If the master does not toggle the FCB, the meter will always respond with the same telegram again. The REQ_UD2 then alternately have a C field of 0x5B or 0x7B.

The parameter **M-Bus max. multipage** in the ***Configuration*** tab restricts the maximum number of consecutively requested telegrams. Especially in the case of meters having a lot of data (e. g. load profiles, due date records), the readout time can be shortened, and less relevant values are not read out at all.

- For most applications, it is sufficient to use the first telegram of the telegram sequence.
- The M-Bus does not provide a mandatory mechanism to directly access certain telegrams of the sequence. As a rule, the procedure always starts from the first telegram. At least all relevant telegrams have to be requested then.
- An „Application reset“ send to the meter reset the sequence to the first telegram.

5.4 Troubleshooting the M-Bus

5.4.1 Physical troubleshooting

In order to determine why meters on the M-Bus do not respond or are not found during the scan, it is recommended to check the M-Bus network physically. It is relatively easy to determine fundamental parameters, e. g. whether the M-Bus is at least correctly wired.

A standard multimeter is sufficient for simple measurements. The most important measurement is the voltage measurement between both M-Bus wires. The voltage measurement shows that:

- the M-Bus-Master correctly supplies the Bus: approx. 30-40 V are present
- the meter is correctly connected to the M-Bus: approx. 30-40 V are present
- the voltage drop is not too high: the voltage at the master is only slightly higher than at the meter
- the telegrams of the master are received at the meter: when master is sending, the value in the display of the multimeter „wobbles“

Another important measurement is the current measurement on the two M-Bus wires. The current measurement shows that:

- the load on the M-Bus is in a valid range: approx. (number of meters)*1.5 mA are flowing
- no external currents are present: current through both lines is identical
- the telegrams of the meter are received at the master: when meter is responding, the value in the display of the multimeter „wobbles“

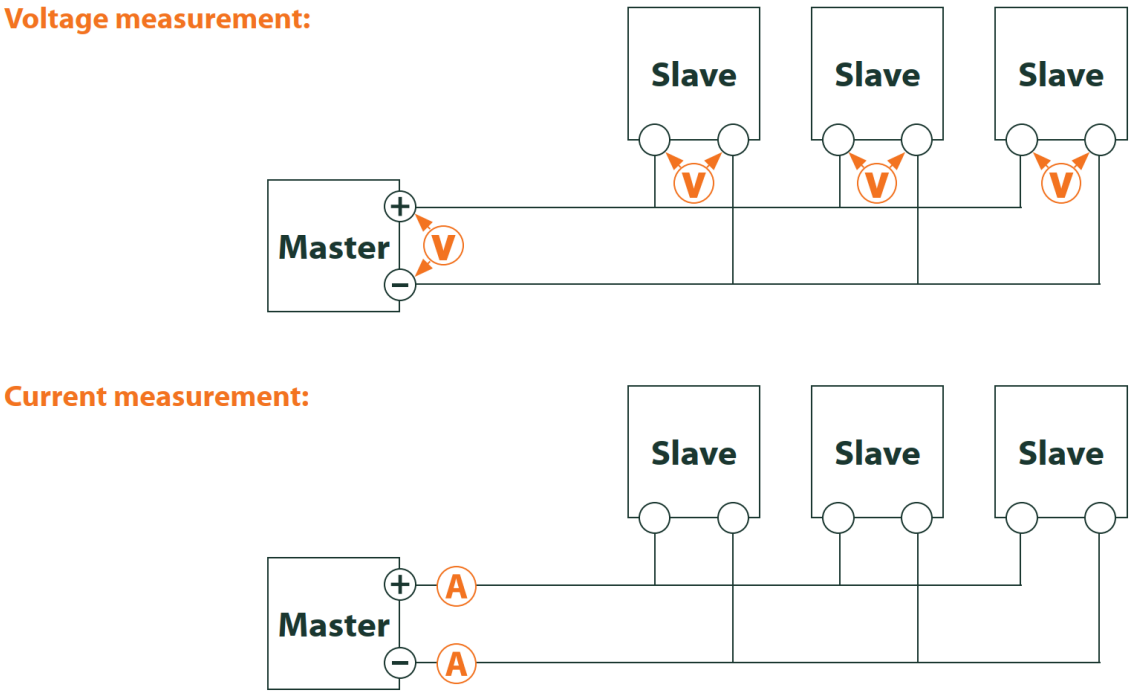


Figure 33: Troubleshooting the M-Bus by measurements with a multimeter

5.4.2 M-Bus meters are not found

Check the cables between the device and the meter, and replace faulty cables if necessary. While the device is switched on, please measure the M-Bus voltage (approx. 30-40 V) between the two M-Bus contacts at the device and also at the meter.

Ensure that the M-Bus interface is activated via the parameter **M-Bus mode** on the the web-based front end in the ***Configuration*** tab and that the scan mode configured therein (secondary or primary) is supported by the meter(s).

Please use an address mask or restrict the range for scanning the M-Bus step by step (e. g. **Primary start address**, **Secondary address mask**).

Additionally, the M-Bus requests can be adapted using the following parameters:

- **M-Bus request mode**
- **M-Bus reset mode**

Please scan again with different M-Bus baud rates (300, 2400 or 9600) or increase the timeouts.

Please remove other meters (if any) to eliminate a possible source of failure.

If another M-Bus meter (possibly of the same type) is available, you can perform another communication test with the other meter to localize the source of failure.

The number of attempts for an M-Bus request can also be increased. The extended configuration of the device in the file *app/chip.ini* (see Section 10.3) offers the parameter **MBUS_MAXRETRY**. This helps to find meters that do not answer every request. The default value here is 3. Please start the scan again.

If the same primary or secondary addresses are present more than once during the scan procedure, collisions can occur. Duplicated addresses are common when using primary addressing, especially in new installations. Therefore we are recommending secondary addressing. In this case collisions can occur as well, but are very unlikely. Due to the default value of the parameter **MBUS_SELECTMASK**=*14* (see Section 10.3), only the 8-digit serial number is searched for during the scan. It can be extened to the manufacturer, medium and version of the meter using other values for **MBUS_SELECTMASK**.

Please activate the raw data log by using **Raw data log** in the ***Configuration*** tab (see Section 4.6). The communication process can be analyzed very well using this raw data log.

If errors could not be eliminated, please contact our customer support:
Email: support@solvimus.de
Phone: +49 3677 7613065

5.4.3 M-Bus meters are found, but do not show any data

Some meters are sending incorrect secondary address or encryption information in the data telegram. As a result, they may not be addressable for readout or may be processed incorrectly.

The parameter **MBUS_SELECTMASK** (see Section 10.3) can be used for masking the invalid parts of the secondary address. The parameter **MBUS_DISABLEDECRYPTION**=*1* (see Section 10.3) can be used to disable the uncommon decryption of M-Bus telegrams if they pretend to be encrypted.

Please restart the scan or start a readout.

If errors could not be eliminated, please contact our customer support:
Email: support@solvimus.de
Phone: +49 3677 7613065

5.4.4 The scan takes a long time

The scanning for M-Bus meters can take a long time under certain circumstances. A duration of more than 1 hour is possible, especially when scanning for secondary addresses of meters with consecutive serial numbers.

Please use an address mask or restrict the range for scanning the M-Bus step by step (e. g. **Primary start address**, **Secondary address mask**).

Please decrease the value of the parameter **MBUS_MAXRETRY** in the file *app/chip.ini* (see Section 10.3) or decrease the timeouts.

Use a different scan mode in the ***Configuration*** tab (see Section 4.6). In particular, the reverse secondary scan *Secondary scan reverse* may help in this case. Please start the scan again.

In the event of interference on the M-Bus, long scan times may also occur. Interference may be processed as a received packet and thus a meter is assumed to be present in each single step.

If errors could not be eliminated, please contact our customer support:
Email: support@solvimus.de
Phone: +49 3677 7613065

5.4.5 Device restarts during scan

For safety reasons, the device uses an internal watchdog, which is intended to prevent the device from becoming unreachable. If the scan takes a very long time, this watchdog may cause the device to restart. If the scan takes a long time, it is recommended to increase the value of the parameter **WATCHDOG_SCAN** in the file *app/chip.ini* (see Section 10.3). Please start the scan again.

There may also be heavy collisions on the bus under certain circumstances, e. g. if all meters are responding at the same time. In exceptional cases, these heavy collisions and the associated large increase in current may cause the device to restart. Please use an address mask or restrict the range for scanning the M-Bus step by step (e. g. **Primary start address**, **Secondary address mask**). If necessary, split the M-bus into physical parts and scan the sections one after another.

If errors could not be eliminated, please contact our customer support:
Email: support@solvimus.de
Phone: +49 3677 7613065

6 Reading meters via Modbus RTU or Modbus TCP

6.1 General information

Originally, the Modbus protocol was developed by the company Modicon (today: Schneider Electric) for the data exchange with its controllers. Data were transferred as 16-bit registers (integer format) or as state information in the form of data bits. Over time, the protocol has undergone continual extensions.

Depending on the interface, the chief variants are:

- Modbus RTU: transfer of binary data via a serial interface
- Modbus ASCII: transfer of human readable data via a serial interface
- Modbus TCP: transfer of binary data via TCP packets in the network

Usually, and depending on the available interface, either Modbus RTU (serial interface, e.g. RS-485) or Modbus TCP (Ethernet interface) is employed. Modbus ASCII and the hybrid Modbus RTU over TCP are rarely met.

➔ A specification can be found at: http://www.modbus.org

The Modbus protocol is a single master protocol. This master controls the entire transfer and monitors potential timeouts (no response from the addressed device). The connected devices are allowed to send telegrams to the master exclusively upon request.

This holds for Modbus RTU via RS-485 and also for Modbus TCP via Ethernet.

A meter with Modbus interface requires a manual configuration. Whereas the Ethernet interface is permanently active in the devices of the solvimus GmbH, enabling an uninterrupted use of Modbus TCP, the serial interface for Modbus RTU needs to be activated its parameters set.

A description of the parameters is given in Section 7.2. The parameter **Serial mode** must be *Modbus RTU* in order to use the RS-485 interface for Modbus RTU.

6.2 Configuration of the meter in the web-based front end

This section describes how to configure meters with Modbus interface.

The configuration is identical for Modbus TCP and Modbus RTU. Only the addressing is different. For Modbus RTU, the serial interface (RS-485) needs to be activated.

A Modbus meter can be added in the tab ***Meter***, see Section 7.3.

The meter is created first via the button **Add** respectively the context menu. The interface **Interface** needs to be set to *Modbus* in the dialogue.

The field **Link** specifies how to address the meter. For Modbus RTU, the slave address of the meter needs to be inserted here.

- ✔ The valid address space is 1..247
- ✔ The address 0 is the broadcast address
- ✔ The addresses 248..255 are reserved

Modbus TCP exploits a vast address space. The IP-address and the TCP-port are added to it. The address scheme adheres to this convention: IP:port/slave address, e.g.: 192.168.1.124:502/1.

- ✔ The TCP-port for Modbus TCP is usually 502.

The field **Byte order** specifies the data representation in Modbus. Modbus uses the data representation *big endian* for bytes and words. Should the meter not respect the standard, other sequences can be set here with *little endian*, *big endian* and *big endian*.

The allocation of the meter data to the meter is assured by the parameters **Serial** and **Manufacturer** whose input is thereby absolutely necessary (see Figure 34 and Figure 35). Further parameters **Medium** or **User label** are optional and can be defined. For the field **Medium**, one can refer to Table 26. This ensures a uniform display across all meters. Use the **Ok** button to accept the entries and the meter is created in the meter list in the ***Meter*** tab.

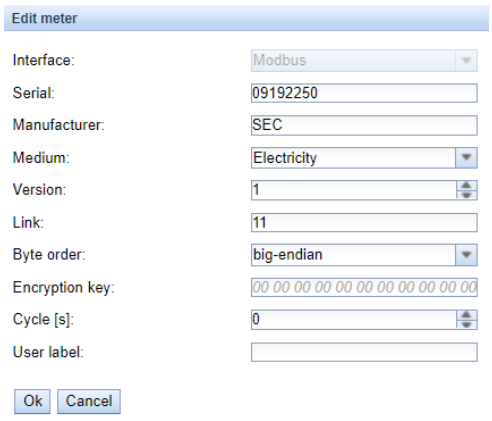


Figure 34: Creation of a Modbus RTU meter (example data)

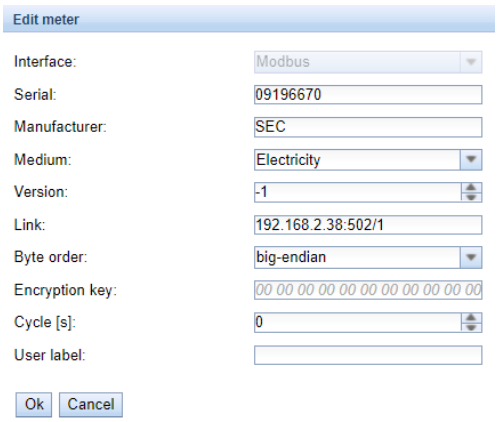


Figure 35: Creation of a Modbus TCP meter (example data)

Then, a value needs to be added to the newly created meter. This is achieved by right-clicking on the newly created Modbus meter and the command **Add value** in the context menu. A dialogue is opened for setting the parameters of the value.

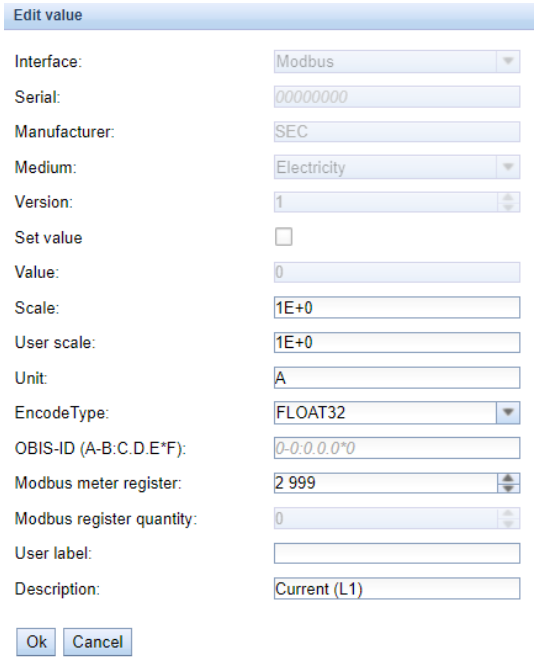


Figure 36: Creation of a value for a Modbus meter (example data)

The allocation of the meter values for Modbus is based on function codes and addresses. The devices of the solvimus GmbH support the function codes *0x03* (*Read Holding Register*) and *0x04* (*Read Input Register*) to capture meter data. The address space for Modbus is 0..65535.

The field **Modbus register address** is configured on that. The function code *0x03* (*Read Holding Register*) can be used if the desired address is fed in directly or the number *300000* is added to the desired address.

For using the function code *0x04* (*Read Input Register*), the number *400000* is added to the desired address. Adding an offset allows to easily switch between Input and Holding registers. A few examples for *Modbus register address*:

- 17 → Read Holding Register, start address 17
- 300017 → Read Holding Register, start address 17
- 400017 → Read Input Register, start address 17

✔ Depending on the manufacturer, the addressing in the data sheet can adhere to Modbus, counting from 0, or deviate from it, counting from 1. The latter has the consequence that the address has to be decremented by 1.

The parameter **Encode type** specifies the number of the registers to be read and their data format. This is a precondition for the correct interpretation of the data. Diverse formats are supported and have to be matched with the data sheet of the meter.

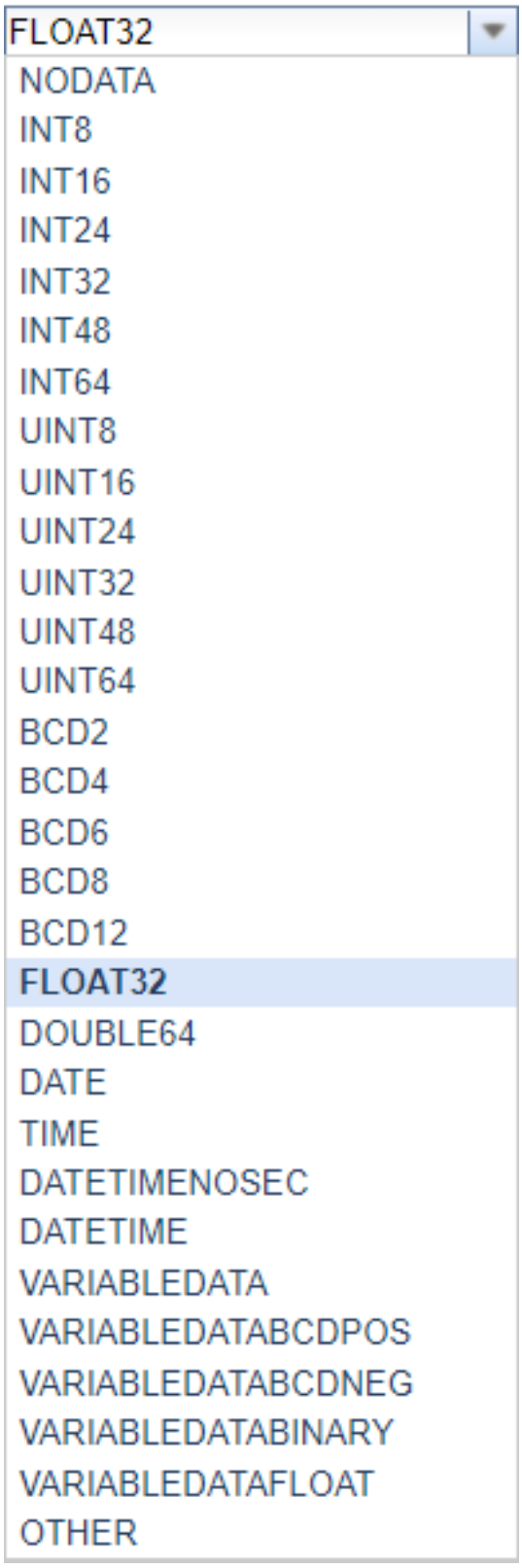


Figure 37: Available Encode Types for Modbus

The parameters **Unit** and **Scale** should also be set according to the meter.

- ✔ We are recommending using basic units such as *Wh* and the factor **Scale** of *1e+3* in contrast to the common unit *kWh* and factor *1e+0*, especially for electricity meters.

For the fields **Description** and **Unit**, the user can refer to Table 27 and Table 28. This ensures a uniform display across all meters.

The value thus configured will then be read out cyclically. For Modbus meters, several values are often transmitted in diverse registers, and therefore further values can be added to the meter.

6.3 Using Templates

As opposed to M-Bus meters, the automatic creation of meter data is not possible for Modbus. To nevertheless enable a swift integration, the devices of the solvimus GmbH provide the possibility to automatically allocate the configuration of a certain value to a newly created meter using templates. Manually adding a value is thus no longer needed.

6.4 Troubleshooting for the Modbus interface

If errors could not be eliminated, please contact our customer support:
E-Mail: support@solvimus.de
Phone: +49 3677 7613065

7 Reading meters via serial interface

7.1 General information

One way for reading meters is the serial communication. Physically, RS-485, RS-232, optical interface (D0) or current loop interface (C0) are typical options.

Some devices from solvimus GmbH are offering an RS-485 interface or an RS-232 interface. Coupling of other physics requires appropriate converters (e. g. optical head for RS-485).

In addition to the physics, the meter's protocol is relevant. Here you can find several variants as well:

- EN 62056-21, also IEC 61107 or IEC 1107 (*ASCII* protocol, called DLDE by us), part of DLMS
- „Real“ DLMS according to EN 62056 series
- SML
- Modbus RTU

The devices from solvimus GmbH support both, SML as well as EN 62056-21 (Mode A and Mode C). SML is only processed when pushed by the meter (unidirectional). EN 62056-21 allows both, the data push and the data pull (request) from the meter (data request).

Devices coming with a serial interface can also access data via Modbus RTU, alternatively to SML and EN 62056-21. Please have a look in Chapter 6 for this functionality. The following sections are mainly related to the general configuration and to SML or EN 62056-21.

7.2 Setup of the interface on the web-based front end

Setting up a meter with serial interface can only be done manually.

First, the serial interface must be activated and parameterized. This is done in the ***Configuration*** tab using the parameter set **Serial...** and **DLDE...** (see Section 4.6).

7.2.1 Serial mode

The parameter **Serial mode** activates the serial interface and defines the fundamental functionality:

- *Disabled*
- *DLDE*
- *Modbus RTU*
- *Transparent/TCP*
- *Transparent/UDP*

The *Transparent* modes allow the access to the physics of the serial interface via a TCP or UDP port. The data stream is forwarded from the serial interface to an IP interface (network (LAN) or cellular radio (WAN)). The device then works in a way similar to an Ethernet-Serial converter or even to a cellular router with a serial interface. The network port to be used is defined in the parameter **Serial transparent port**.

✔ The transparent mode allows reading meters via serial interface even if their protocol is not directly supported by the device. The protocol can then be processed in the control system (host system) while the device provides physical connectivity.

The modes *DLDE* and *Modbus RTU* activate the reading of meters by the device itself. This means that the protocol is handled directly in the device and the meter must be set up accordingly (see Section 7.3).

✔ Regardless of the mode, the parameters for baud rate, bit representation and timeouts must be set accordingly (see Section 7.2.2).

7.2.2 Serial baud rate, data bits, stop bits and parity

The parameters **Serial baud rate**, **Serial data bits**, **Serial stop bits** and **Serial parity** are used to configure the bit representation on the serial interface.

The baud rate essentially determines the speed of the data transmission. The other parameters describe the byte representation:

- The number of data bits is either 7 Bit or 8 Bit.
- The parity activates an additional bit to enable an error detection. While parity *None* (no parity, N) is not using this additional bit, the modes *Even* (even parity, E) or *Odd* (odd parity, O) add a corresponding bit which supplements the data bits in such a way as to obtain an even or odd number of ones (1) in the data stream. The modes *Mark* (M) and *Space* (S) complement either a 1 or a 0, but are practically not used.
- The number of stop bits is either 1 Bit or 2 Bit.

Usual settings are for example:

- 2400-8-E-1 (e. g. for M-Bus)
- 300-7-E-1 (e. g. for meters according to EN 62056-21)
- 9600-8-N-1 (e. g. for meters with SML-Push or according to DLMS)
- 19200-8-N-1 (e. g. for Modbus RTU)

7.2.3 DLDE mode

Three variants of the protocol according to EN 62056-21 are implemented. The parameter **DLDE mode** selects the one to be used.

The mode *Push* is used for meters that are sending their data cyclically, unsolicited. Supported data formats are EN 62056-21 and SML protocol.

Meters which need a data request according to EN 62056-21 can be read out either via the mode *Request* or the mode *Request (C-Mode)*. Using *Request* activates the Mode A described in the standard. When the meter is queried, it sends its meter values directly in the response. The Mode C described in the standard allows a baud rate change before the responding with meter data. For this purpose an additional telegram exchange is mandatory (baud rate negotiation). The exchange is supported in the mode *Request (C-Mode)*, but the already used baud rate is requested.

7.2.4 Serial timeouts

The serial interface uses three different timeouts, which are **Serial first timeout**, **Serial idle timeout** and **Serial full timeout** (in transparent mode only the **Serial idle timeout** is used, in mode Modbus RTU only the **Serial first timeout**).

The **Serial idle timeout** specifies what time the serial interface has to be „idle“, i. e. no data is sent/received, in order to detect the end of a telegram (end of communication). It is mainly used for packetising serial data stream, i. e. the assignment of incoming data to a logical unit (data packet). In *Push* mode this time is used to detect the start of the telegram. Therefore, the meter has to interrupt its transmission for at least this period of time.

The **Serial first timeout** specifies how long the device has to wait for incoming data from the meter. If no data is received within this period of time from the request, the readout attempt is aborted.

The **Serial full timeout** specifies the latest time at which reception is interrupted in order to process the received meter data. This parameter also terminates reception if the **Serial idle timeout** is not reached because data is continuously received (without idle time, e. g. in the event of failure).

7.3 Setup of a meter on the web-based front end

This section describes how to set up meters with DLDE interface (EN 62056-21) and is relevant only for a device MUC.easy^{plus}. For meters with Modbus RTU interface, this is explained in Section 6.2.

After activating and parameterizing the serial interface, the meter can be added in the ***Meter*** tab.

The meter is created using the **Add** button or the context menu. In the dialogue, the **Interface** has to be set to *DLDE*. Further data such as manufacturer code, serial number, **Medium** or **User label** are optional and can be assigned. The user may refer to Table 26 for the **Medium** field. This ensures a uniform display across all meters. Use the **Ok** button to accept the entries and the meter is created in the meter list in the ***Meter*** tab.

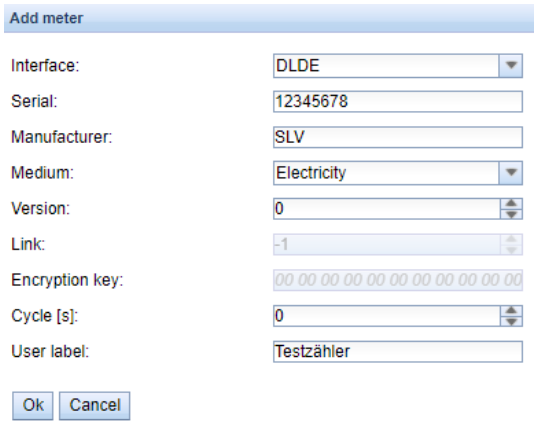


Figure 38: Creating a DLDE meter (sample data)

A meter value now has to be added to the newly created meter. This is done by right-clicking on the newly added DLDE meter and selecting the **Add value** command from the context menu. This command opens a dialogue for entering the parameters of the meter value.

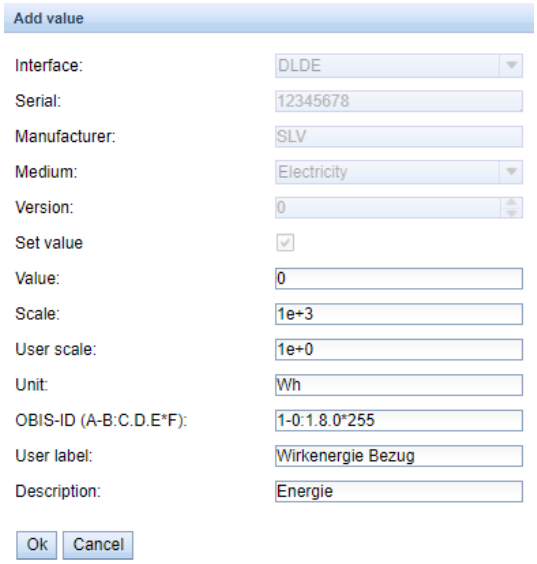


Figure 39: Creating the meter value of a DLDE meter (sample data)

The mapping of meter values in EN 62056-21 (DLDE) is based on *OBIS* codes. This code consists of six octets and is standardized worldwide for clearly describing the measured value. Therefore, it is mandatory to assign the correct code in the parameter **OBIS-ID (A-B:C.D.E*F)**. The parameters **Unit** and **Scale** should also be set according to the meter.

- ✔ We are recommending using basic units such as *Wh* and the factor **Scale** of *1e+3* in contrast to the common unit *kWh* and factor *1e+0*, especially for electricity meters.

The user can refer to Table 27 and Table 28 for filling in the fields **Description** and **Unit**. This ensures a uniform display across all meters.

The meter value set up in this way is now read out from the meter and recorded cyclically. DLDE meters are often transmitting multiple values for various OBIS codes, so additional meter values can be added to the meter. Here are a few examples of commonly used OBIS codes, especially for energy meters:

- 1-0:1.8.0*255 → Total active energy import
- 1-0:1.8.1*255 → Total active energy import (tariff 1)
- 1-0:1.8.2*255 → Total active energy import (tariff 2)
- 1-0:2.8.0*255 → Total active energy export
- 1-0:3.8.0*255 → Total apparent energy import
- 1-0:4.8.0*255 → Total apparent energy export
- 1-0:1.7.0*255 → Instantaneous active power import
- 1-0:31.7.0*255 → Instantaneous current phase 1
- 1-0:51.7.0*255 → Instantaneous current phase 2
- 1-0:71.7.0*255 → Instantaneous current phase 3
- 1-0:32.7.0*255 → Instantaneous voltage phase 1
- 1-0:52.7.0*255 → Instantaneous voltage phase 2
- 1-0:72.7.0*255 → Instantaneous voltage phase 3

7.4 Troubleshooting the serial interface

7.4.1 Meters are not read out

Please check whether the parameters of the serial interface are set correctly in the ***Configuration*** tab.

Please check whether the meter supports the protocol according to EN 62056-21 (**DLDE mode** *Request*) or transmits data cyclically according to EN 62056-21 or SML format (formatBefehlDLDE mode *Push*).

Please check the timeout parameters of the serial interface in the ***Configuration*** tab.

Please activate the raw data log using the parameter **Raw data log** in the ***Configuration*** tab. The communication process can be analyzed very well using this raw data log.

If errors could not be eliminated, please contact our customer support:
Email: support@solvimus.de
Phone: +49 3677 7613065

8 Reporting of metering data

8.1 General information

Regarding the transmission of metering data to third-party systems such as meter data management, energy management or monitoring systems, a fundamental distinction is made between actively sending data, the data push, and data is getting fetched, the data pull.

Using the client-server model, in the case of data push the device from solvimus GmbH is the client and the third-party system is the server. In the case of the data pull, the device from solvimus GmbH is the server and the third-party system is the client. The client always establishes the connection and controls the data transmission. The server answers the requests and executes the commands of the client.

This chapter describes the data push, which can be configured on the data concentrators of solvimus GmbH in the ***Server*** tab.

The data pull is described separately, e. g. in Section 8.10, Chapter 11, Chapter 12 or in Section 2.7.

8.2 Instances and database

Devices from solvimus GmbH with the „Multi Channel Reporting“ (MCR) function can handle 10 independent report instances. The parameters such as cycle time, data format, transport mechanism and others can be set for each of these reports in the ***Server*** tab (see Section 4.8).

The data sent in the reports is stored in a database on the devices from solvimus GmbH. The database is file-based and uses *SQLITE*. The report instances therefore are handling the same data.

- The database containing the meter data and the meta information is active if either at least one report instance is active or the configuration parameter MUC_FORCESTOREREADOUT in the configuration file *chip.ini* (see Table 24 and field name 'Store meter values' in Table 10) is set to the value 1. Otherwise (value 0, and no report defined), no meter values are stored in the database.
- Only activated values (column *Active* in the ***Meter*** tab) are written to the database. Other values are not available later.

8.3 General settings

Each instance has a parameter set. This can be configured on the web-based front end in the ***Server*** tab. Some parameters are always to be configured, others depend on the set mode.

The following parameters are available and have to be configured for each instance:

- **Report mode**: Sets the operating mode of the respective instance or deactivates it (see also Section 4.8).
- **Report format**: Sets the data format used for the transmission of the respective instance (see also Section 4.8).
- **Report cycle mode**: Format for specifying the report cycle of the respective instance (see also Section 4.8)
- **Report cycle**: Report cycle of the respective instance (see also Section 4.8)
- **Report cycle date (local)**: First report day of the respective instance in case of daily to yearly specification of the report cycle, depending on the interval format the entered month is used, the year is not relevant (see also Section 4.8)
- **Report cycle time (local)**: Report time of the respective instance for daily to annual specification of the report cycle (see also Section 4.8)

8.4 Defined data and file formats

The devices from solvimus GmbH are offering some defined data formats.

8.4.1 XML format

Several XML formats are available for reporting data. XML is a data stream using so-called tags or markups (entries/elements and attributes) for presenting hierarchically structured data. This data is usually in plain text and therefore readable by both humans and machines.

The XML format is specified as follows:

Entry	Attribute	Description
interface		Contains a complete packet with one or more muc entries.
	MESSAGE_TYPE	Specifies the type/version of the packet: e. g. 1
muc		Contains the data for one respective device with corresponding meter entries.
	MUC_ID	Hexadecimal notation of the serial number of the device (corresponds to the serial number/MAC address on the web-based front end in the ***General*** tab).
	VERSION	Protocol version
	TIMESTAMP	UNIX time (UTC) at time of report
meter		Contains the data for one respective meter with corresponding data entries.
	INTERFACE	Interface of the meter, as number (up to XML-8) or as text (XML-9) 1: S0 2: M-Bus 5: wM-Bus 6: DLDERS 10: System 11: Modbus
	METER_ID	Serial number of the meter
	USER	User-specific description of the meter (column **User label** in ***Meter*** tab)
	MAN	Manufacturer code of the meter
	VER	Version number of the meter
	MED	Medium of the meter, see second column in Table 26
	MED_ID	Medium ID of the meter, see first column in Table 26
data		Contains one or more meter values of the same type in the respective entry items. The values are specified via the attributes.
	OBIS_ID	OBIS code according to OBIS specification, configured via the web-based front end (column **OBIS-ID** in ***Meter*** tab). In version XML-8 the raw DIF/DIFE/VIF/VIFE fields coming from M-Bus/wM-Bus for that meter value are reported here.
	DESCRIPTION	See second column in Table 27
	MEDIUM	Medium of the meter, see second column in Table 26
	UNIT	See second column in Table 28, energy values in Wh are converted to kWh
	SCALE	Signed scaling factor (scientific notation): (scale of the meter) · (User Scale)
	DIF	DIF/DIFE fields from the M-Bus/wM-Bus raw data, in hexadecimal byte notation
	VIF	VIF/VIFE fields from the M-Bus/wM-Bus raw data, in hexadecimal byte notation
	USER	User-specific description of the meter value (column **User label** in ***Meter*** tab)
entry		Data entry consisting of a parameter timestamp (T) and a parameter value (VAL)
parameter		Contains a parameter item.
	NAME="T"	The associated parameter item represents the UNIX time (UTC) at the time of the measurement, if transmitted by the meter together with the measured value.
	NAME="T_MUC"	The associated parameter item represents the system time of the device at the time of data reception as UNIX time (UTC).
	NAME="VAL"	The associated parameter item represents the received value of the meter value specified in data.

Table 19: Format of XML data

The following table illustrates the different protocol versions:

Entry	Attribute	XML-3	XML-6	XML-7	XML-8	XML-9
interface		x	x	x	x	x
	MESSAGE_TYPE	x	x	x	x	x
muc		x	x	x	x	x
	MUC_ID	x	x	x	x	x
	VERSION	1F4	1F7	1F8	1F9	9
	TIMESTAMP	x	x	x	x	x
meter		x	x	x	x	x
	INTERFACE	Numerical	Numerical	Numerical	Numerical	Text
	METER_ID	x	x	x	x	x
	USER		x	x	x	x
	MAN			x	x	x
	VER			x	x	x

Continued on next page

Table 20 – *Continued from previous page*

Entry	Attribute	XML-3	XML-6	XML-7	XML-8	XML-9
	MED			×	×	×
	MED_ID					×
data		×	×	×	×	×
	OBIS_ID	×	×	×	Raw data	×
	DESCRIPTION	×	×	×	×	×
	MEDIUM	×	×	×	×	
	UNIT	×	×	×	×	×
	SCALE	×	×	×	×	×
	VIF					×
	DIF					×
	USER		×	×	×	×
entry		×	×	×	×	×
parameter		×	×	×	×	×
	NAME="T"	×	×	×	×	×
	NAME="T_MUC"	×	×	×	×	×
	NAME="VAL"	×	×	×	×	×

Table 20: Data in different XML versions

A sample XML packet in version XML-3 looks like this:

```
<?xml version="1.0" encoding="utf-8"?>
<interface MESSAGE_TYPE="1">
 <muc MUC_ID="13fd0" VERSION="1F4" TIMESTAMP="1252004322">
 <meter METER_ID="92752244" INTERFACE="5">
 <data DESCRIPTION="VOLUME" UNIT="m^3" SCALE="0.001" MEDIUM="WATER"
 OBIS_ID="8-0:1.0.0*255">
 <entry>
  <parameter NAME="T">1253000282</parameter>
  <parameter NAME="T_MUC">1253000282</parameter>
  <parameter NAME="VAL">2850427</parameter>
 </entry>
 <entry>
  <parameter NAME="T">1253000482</parameter>
  <parameter NAME="T_MUC">1253000482</parameter>
  <parameter NAME="VAL">2850428</parameter>
 </entry>
 </data>
 <data ...>
 ...
 </data>
 </meter>
 <meter ...>
 ...
 </meter>
 </muc>
</interface>
```

8.4.2 CSV format

Several CSV formats are available for transmission of raw frames. CSV is a table-like file format which uses a character, solvimus GmbH uses a semicolon „;“ (in **CSV-10** a comma), for separating numerical values and texts (columns) from each other. This allows easy processing or viewing e. g. in Excel.

The first line in the file (in all protocol versions except **CSV-0** and **CSV-1**) specifies the column heading. The following lines contain the data of the meters and its meter values at a particular readout time.

The CSV data has the following format:

Column heading	Description
Information related to meters	
Index	Indexes the different meters within a CSV file

Continued on next page

Table 21 – *Continued from previous page*

Column heading	Description
Timestamp	System time of the device at the time of data reception as UNIX time (UTC) or in readable format
DeviceId	ID of the meter, consisting of manufacturer code, serial number, version number and medium ID
Link	Primary address of the meter for M-Bus resp. reception quality (RSSI, in steps of -0.5 dBm) for wM-Bus
User	User-specific description of the meter (column **User label** in ***Meter*** tab)
METER_ADDRESS	ID of the meter, composed of manufacturer code, serial number, version number and media type
READING_DATE	Unix timestamp (UTC) or readable time of the device at the instant of the readout
RAW_TELEGRAM	Telegram
Information related to meter values	
IndexX	Indexes the different meter values of a meter
ValueX	Meter value
ScaleX	Signed scaling factor (scientific notation): (scale of the meter) · (User Scale)
UnitX	See second column in Table 28
DescriptionX	See second column in Table 27
UserX	User-specific description of the meter value (column **User label** in ***Meter*** tab)
TimestampX	The timestamp transmitted by the meter (UNIX time (UTC) or in readable format), or 0 if not available
ObisidX	OBIS-ID (column **OBIS-ID** in ***Meter*** tab)

Table 21: CSV format

The first columns of each line contain data of the meter, including the meter identification (address) and the time at which the data was read out. The other columns are added dynamically according to the configured meters and number of meter values, whereby the meter values are inserted by counting from 0 (e. g. Value0).

The following table illustrates the different protocol versions:

Column	CSV-0	CSV-1	CSV-3	CSV-4	CSV-5	CSV-6	CSV-9	CSV-10
Index						x	x	
Timestamp	Unix	Unix	Unix	Unix	Unix	Unix	Klartext	
DeviceId	x	x	x	x	x	x	x	
Link				x	x	x	x	
User					x	x	x	
METER_ADDRESS								x
READING_DATE								x
RAW_TELEGRAM								x
IndexX						x	x	
ValueX	x	x	x	x	x	x	x*	
ScaleX	x	x	x	x	x	x		
UnitX	x	x	x	x	x	x	x	
DescriptionX	x	x	x	x	x	x	x	
UserX			x	x	x	x	x	
TimestampX			Unix	Unix	Unix	Unix	Klartext	
ObisIdX		x	x	x	x	x	x	

* (meter value) · (scale of the meter) · (User Scale)

Table 22: Data in different CSV versions

A sample CSV file in version **CSV-3** is shown in the following figure:

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	Timestamp	DeviceId	Value0	Scale0	Unit0	Description0	User0	Timestamp0	ObisId0	Value1	Scale1	Unit1	Description1
2	1370135021	EMU-000238	987	1,00E+00	Wh	Energy		0					
3	1370135025	EMH-003898	18354	1,00E+00	h	On Time		1339357800		24214	1,00E+01	Wh	Energy
4	1370135028	ZRM-314040	90	1,00E-03	m^3	Volume	label5	1369836720		1943	1,00E-02	Grad C	Flow Tempe
5	1370135030	LUG-6666020	436	1,00E+03	Wh	Energy	label 1	1370141940	1-0:0.0.0*0	650	1,00E-03	m^3/h	Volume Flov
6	1370135031		245	1,00E-03	m^3			0	0-2:2.0.0*0				
7	1370200016	EMU-000238	987	1,00E+00	Wh	Energy		0					
8	1370200020	EMH-003898	18373	1,00E+00	h	On Time		1339422780		24228	1,00E+01	Wh	Energy
9	1370200022	ZRM-314040	90	1,00E-03	m^3	Volume	label5	1369901700		1945	1,00E-02	Grad C	Flow Tempe
10	1370200025	LUG-6666020	436	1,00E+03	Wh	Energy	label 1	1370206920	1-0:0.0.0*0	650	1,00E-03	m^3/h	Volume Flov
11	1370200026		245	1,00E-03	m^3			0	0-2:2.0.0*0				
12													
13													

Figure 40: Excerpt of a CSV file

The transmission of data in the **CSV-10** format requires setting in the configuration file *app/chip.ini* (see Section 10.3) that the frames of the meters are joined to the data by defining the configuration parameter

MUC_SHOWDATAFRAME=1. If the meters had been created before, the values of the frames must be activated subsequently. A sample data set in **CSV-10** format is given here (long lines are wrapped):

METER_ADDRESS, READING_DATE, RAW_TELEGRAM
15686402,13:45:56 23/07/2021,4544B4090264681509077A3D2000000C13420100000F1B2C16870111201623
07210E00000E00000E00000E00000E00000E00000E00000E00000E00000E00000E00000E000000
00000048,13:46:54 23/07/2021,1E44B05C48000000011B7AA20000002F2F0A66310202FD971D00002F2F2F2F

8.4.3 JSON format

One JSON format is available for the reports. JSON is a compact, serialized data stream for representing structured data. This data is usually readable by both humans and machines and separated by delimiters.

Object	Property	Data type	Description
muc		Object	Contains the data for one respective device with corresponding meter entries.
	MUC_ID	String	Hexadecimal notation of the serial number of the device (corresponds to the serial number/MAC address on the web-based front end in the ***General*** tab).
	VERSION	String	Protocol version
	TIMESTAMP	Integer	UNIX time (UTC) at time of report
	meter	Array	Array of meter objects
meter		Object	Contains the data for one respective meter with corresponding data entries.
	METER_ID	String	Serial number of the meter
	INTERFACE	String	Interface of the meter S0 MBus wMBus DLDERS System Modbus
	MAN	String	Manufacturer code of the meter
	VER	String	Version number of the meter
	MED	String	Medium of the meter, see second column in Table 26
	MED_ID	String	Medium ID of the meter, see first column in Table 26
	USER	String	User-specific description of the meter (column **User label** in ***Meter*** tab)
	data	Array	Array of data objects
data		Object	Contains the data for one respective meter value with the corresponding entries.
	DESCRIPTION	String	See second column in Table 27
	UNIT	String	See second column in Table 28, energy values in Wh are converted to kWh
	SCALE	String	Signed scaling factor (in decimal form): (scale of the meter) · (User Scale)
	OBIS_ID	String	OBIS code according to OBIS specification, configured via the web-based front end (column **OBIS-ID** in ***Meter*** tab).
	USER	String	User-specific description of the meter value (column **User label** in ***Meter*** tab)
	DIF	String	DIF/DIFE fields from the M-Bus/wM-Bus raw data, in hexadecimal byte notation
	VIF	String	VIF/VIFE fields from the M-Bus/wM-Bus raw data, in hexadecimal byte notation
	entry	Array	Array of entry objects
entry		Object	Data entry consisting of a parameter timestamp (T) and a parameter value (VAL)
	T_MUC	Integer	System time of the device at the time of data reception as UNIX time (UTC)
	T	Integer	UNIX time (UTC) at the time of the measurement, if transmitted by the meter together with the measured value
	VAL	String	Value of the meter value specified in data

Table 23: Format of the JSON data

A sample JSON packet looks like this (line feeds are inserted for better illustration):

```
{"muc":{ "MUC_ID":"6891d0800e62","VERSION":"1","TIMESTAMP":1601297784,"meter":[
{"METER_ID":"00000001","INTERFACE":"MBus","MAN":"SIE","VER":21,"MED":"Electricity",
"MED_ID":2,"USER":"metering1","data":[
{"DESCRIPTION":"Energy","UNIT":"kWh","SCALE":0.001,"OBIS_ID":"1-0:1.8.0*255",
"USER":"energy3","DIF":"04","VIF":"03","entry":[
```

```
{"T_MUC":1601297679,"VAL":"537980"},{"T_MUC":1601297761,"VAL":"537980"},
{"T_MUC":1601297765,"VAL":"537980"},{"T_MUC":1601297770,"VAL":"537980"}]}]},
{"METER_ID":"00094824","INTERFACE":"MBus","MAN":"BEC","VER":32,"MED":"Electricity",
"MED_ID":2,"data":[
{"DESCRIPTION":"Energy","UNIT":"kWh","SCALE":0.01,"DIF":"0E","VIF":"84 00","entry":[
{"T_MUC":1601297679,"VAL":"2887897"},{"T_MUC":1601297761,"VAL":"2887897"},
{"T_MUC":1601297765,"VAL":"2887897"},{"T_MUC":1601297770,"VAL":"2887897"}]},
{"DESCRIPTION":"Power","UNIT":"W","SCALE":0.01,"DIF":"04","VIF":"A9 00","entry":[
{"T_MUC":1601297679,"VAL":"382207"},{"T_MUC":1601297761,"VAL":"382207"},
{"T_MUC":1601297765,"VAL":"382207"},{"T_MUC":1601297770,"VAL":"382207"}]}]}]}}
```


8.4.4 User format

If the options above do not fit or are not sufficient, the report can be switched to be script-based by setting **Report format** *User* in the ***Server*** tab.

This provides an integrated XSLT parser to the user for generating specific data formats. An overview is given in Section 10.7 and particularly in Section 10.7.1.

- ✔ For each instance, an individual User format can be used. The file name is used for the mapping.

8.5 Reporting data via TCP

A common communication method for transferring data is using TCP packets and their data container. The data is thus sent as a data stream to the remote station, where it is gathered and processed.

Using TCP, the data is transmitted unencrypted. If encryption is necessary, the data should be sent via TLS (see Section 8.6).

Since the systems for the data processing are usually using databases or similar things, data formats which can be processed automatically, such as XML or JSON, are preferred here. But any data format can be transferred.

According to the destination the parameters **Report address**, **Report port** and **Report directory** have to be set. An empty path specified in **Report directory** generates a TCP data stream, a non-empty path generates an HTTP data stream (e. g. /", /upload").

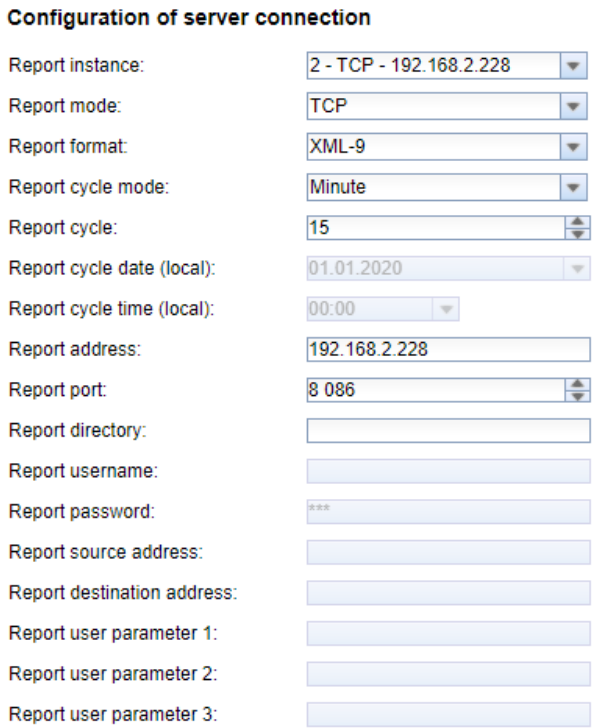


Figure 41: Sample configuration for reporting XML data via TCP every 15 minutes

8.6 Reporting data via TLS

As a rule, transmitting data via an unencrypted TCP connection (see Section 8.5) is not recommended for commercial or industrial applications. Encryption is common here.

Using TLS, the TCP data stream is asymmetrically encrypted. Each participant has both a private key known only to him and a public key known to everyone. Data that is exchanged gets encrypted with the public key of the other participant. The decryption is then performed using the secret private key on the recipient side.

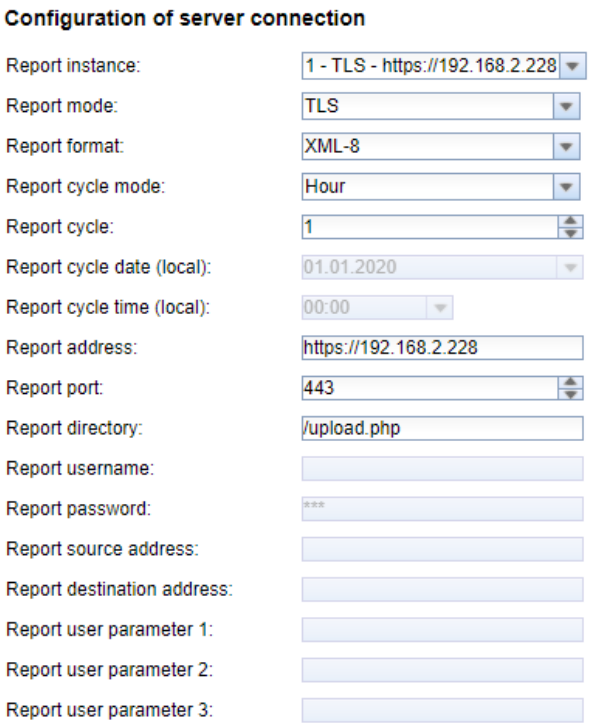


Figure 42: Sample configuration for reporting XML data via TLS every hour

TLS also offers mutual authenticity checks of client and server by means of signed certificates. This provides a very high level of security. A distinction is made between server-side authentication and client-side authentication, depending on which side is authenticating. The products from solvimus GmbH are supporting both variants, also in combination.

- ✔ The devices from solvimus GmbH are using certificates in the *PEM* format (RFC 7468).

In the case of server-side authentication, the device from solvimus GmbH checks if the server is trustworthy. This requires an installed certificate (public key) issued by the certification authority to be relied upon, and who has signed the certificate of the server.

- ✔ Unless otherwise specified and available, the devices from solvimus GmbH are using *app/cacert.pem* for checking the authenticity of the server (RFC 4945).

In the case of client-side authentication, the client has to authenticate itself. In the case of data concentrators and gateways this means the device itself. This requires an issued certificate and a secret private key.

- ✔ Unless otherwise specified and available, the devices from solvimus GmbH are using *app/clicert.pem* as the certificate of the device (RFC 5280).
- ✔ Unless otherwise specified and available, the devices from solvimus GmbH are using *app/clikey.pem* as the private key of the device (RFC 5958).

The certificates can be uploaded manually via SFTP (see also Section 3.5). However, it is also possible to import them via the ***Service*** tab (see Section 4.12). The files have to be archived into a **.tar.gz* file in this case.

- ➔ The free, open source software *7-Zip* can be used for creating a **.tar.gz* archive. As an example, the file *cacert.pem* can first be packed into a **.tar* ball without sub-directory and packed into a **.gz* archive afterwards.

For using individual certificates for each server instance or if the naming or path has to be different, the file app/chip.ini allows to enter other file names and paths manually (see also Section 10.3).

In section *[REPORT_x]* in the file *app/chip.ini* the following parameters are used for assigning *PEM* files to the respective report:

- CA_FILE: the public key of the certification authority matching the server certificate, e. g.: `CA_FILE=app/srv_instance1.pem`
- CERT_FILE: the certificate of the device for the respective report, e. g.: `CERT_FILE=app/dcu.pem`
- KEY_FILE: the private key matching the certificate of the device, e. g.: `KEY_FILE=app/key.pem`

8.7 Sending files via FTP

Another common communication method for transferring data is using the FTP protocol, especially if the transmission is file-based.

Using classical FTP, the data is transmitted unencrypted. In general, unencrypted FTP is not recommended for file transmission in commercial or industrial applications. An encryption is possible by using FTP via a TLS connection (FTPS) or SFTP.

The device supports the following protocols:

- ftp: Unencrypted FTP
- ftpes: Explicit FTPS, unencrypted setup of the connection and subsequent start of the encryption using STARTTLS
- ftps: Implicit FTPS, FTP protocol via a TLS-encrypted connection
- sftp: Transmission via SSH (see Section 8.7.1)

The desired protocol must precede the server address in the field **Report address**. In the absence of a protocol, ftpes is assumed.

Commonly, FTP servers permit using an encryption via Explicit FTPS on the same port as unencrypted FTP.

For all encrypted protocols, both the login and the file transmission occur via encrypted connections.

When using FTPS, the Root CA certificates authenicating the server must be stored on the device (see Section 8.6).

Since files are transferred, the CSV format is preferred here. It allows easy import into Excel or databases among other things. However, other data formats can also be transferred.

According to the destination the parameters **Report address**, **Report port**, **Report directory**, **Report username** and **Report password** have to be set.

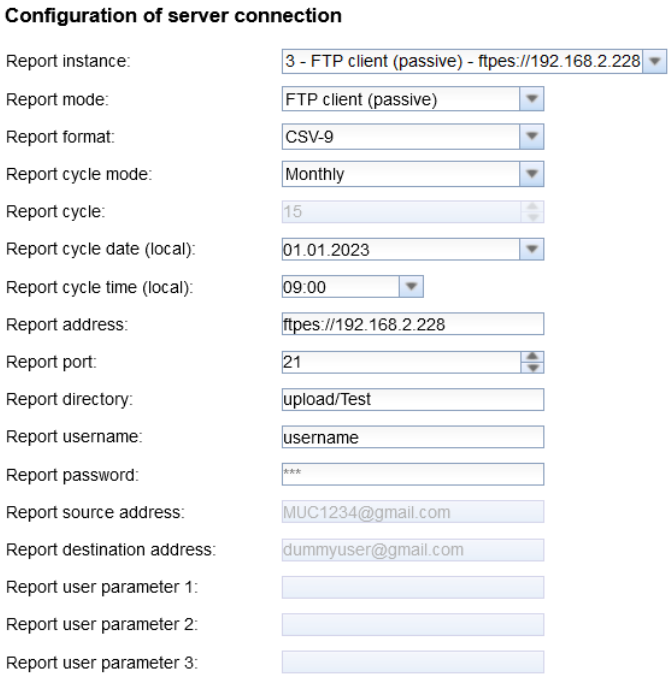


Figure 43: Sample configuration for reporting CSV data via FTP every month

The **Report mode** is either *FTP (active)* or *FTP (passive)*. Both differ in the process of determining the port to be used for the data connection. FTP uses one TCP port for the control connection, e. g. for transmitting control commands, and a second TCP port for the data connection. The client (the device) defines the data port in *active* mode, the server defines the data port in the *passive* mode. Therefore, *FTP (passive)* is usually used, because firewalls on the server side are often allowing only outgoing connections on an „arbitrary“ port.

✔ If no **Report port** is entered, the default port *21* is used.

8.7.1 Sending files via SFTP or FTPS

SFTP is an emulated FTP via SSH and permits an encrypted file transmission. Contrary to FTPS, SFTP has the advantage that SSH and therefore only one port is used (usually port *22*).

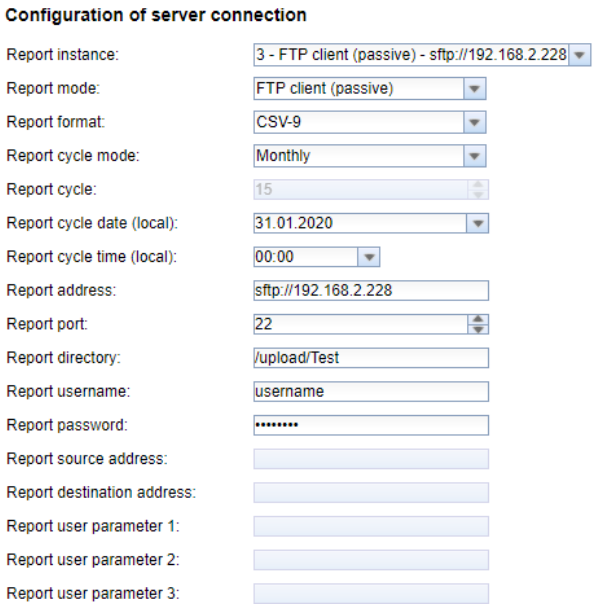


Figure 44: Sample configuration for reporting CSV data via SFTP every month

The respective certificates or finger prints have to be provided and configured. The background and the procedure for handling certificates are described in Section 8.6.

In contrast to the certificates, fingerprints for SSH are handled differently. SSH and thus SFTP use the asymmetric encryption and are secured by certificates. Both remote stations have both a private and a public key. A PKI (Public Key Infrastructure) is used to check the authenticity. This is usually associated with administrative work. Therefore, the authenticity can also be confirmed by the user.

For this purpose, a finger print is exchanged during the initial connection, which uniquely identifies the remote station. The finger print is the public key of the remote station. Now the user can manually check and trust this. If this remote station is a trusted host, its fingerprint must be entered into the file *app/ssh/known_hosts*. This is done by adding such a line to the file:

- 192.168.2.34 ecdsa-sha2-nistp256 AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAy[...]

Therefore, the corresponding fingerprint of the server must first be fetched in order to be entered into this file. There are two possibilities:

- The fingerprint is fetched directly from the server and manually entered into the file *app/ssh/known_hosts*.
- The server is accessed via SSH from the device and its fingerprint accepted by confirmation of the security prompt. Then the fingerprint is automatically written to the file *app/ssh/known_hosts*.

This can be done directly from the device via the SSH console:

```
> ssh admin@192.168.2.34 <ENTER>
The authenticity of host '192.168.2.34 (192.168.2.34)' can't be established. ECDSA key
fingerprint is SHA256:HtAa1pkvafJSmAiMJmi1ZvJi6spgf5i0yt/A2rJ/OnY. Are you sure you
want to continue connecting (yes/no/[fingerprint])?
yes <ENTER>
Warning: Permanently added '192.168.2.13' (ECDSA) to the list of known hosts.
```

This allows an encrypted cyclic upload of meter data via SFTP.

8.8 Sending emails via SMTP

Data can also be sent via email. SMTP is used for this purpose.

SMTP itself is not encrypted. The STARTTLS extension provides a secure connection based on TLS, but an unencrypted connection is established for compatibility reasons, and then encrypted prior to the login. Another alternative is smtps which immediately creates a TLS-encrypted connection.

The protocol in front of the server address in the field **Report address** determines the SMTP-variant to be implemented. The device supports the following protocols:

- smtp: Unencrypted SMTP
- smtps: SMTP via TLS-encrypted connection
- smtpes: SMTP with encryption with STARTTLS extension

In the absence of a protocol, smtpes will be used.

The parameters **Report address**, **Report port**, **Report username**, **Report password**, **Report source address** and **Report destination address** have to be set according to the email server and the email data.

- ✔ The following ports are commonly used: *25* for unencrypted SMTP, *587* for SMTP with STARTTLS and *465* for SMTPS.
- ✔ When using TLS (SMTP mit STARTTLS oder SMTPS), the respective certificates have to be provided. Please have a look at Section 8.6. Please contact our customer support.

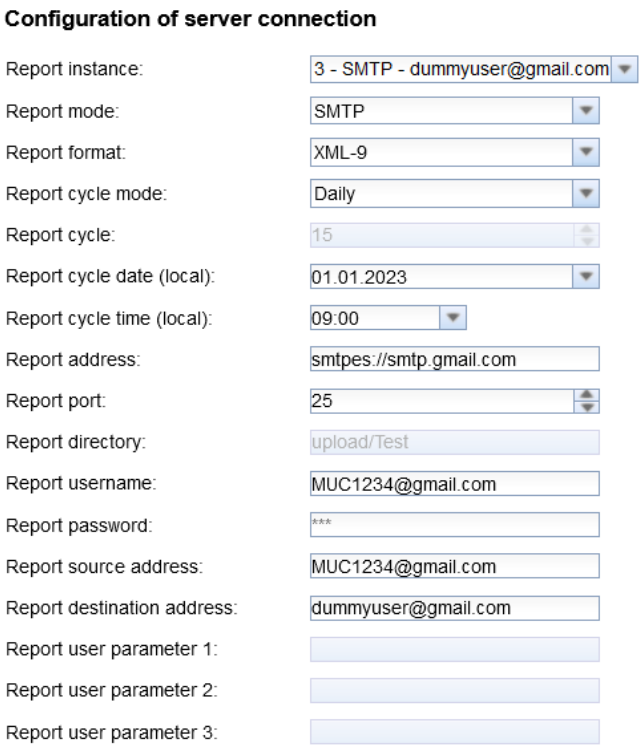


Figure 45: Sample configuration for reporting XML data via email every day

Depending on the requirement, it is possible to send this data in the text content of the email or as an attachment.

8.8.1 Emailing the report as content

Using *SMTP* as **Report mode**, devices from solvimus GmbH are sending the data in the content (text) of an email. Only the parameters in the ***Server*** tab need to be configured.

8.8.2 Emailing the report as attachment

Using *SMTP with Attachment* as **Report mode**, devices from solvimus GmbH are sending the data as attachment to the email, the content (text) of the email remains empty. Only the parameters in the ***Server*** tab need to be configured.

8.9 Reporting data via MQTT

MQTT is a very popular standard in cloud communication, especially for sending data to a cloud system. It is an open network protocol which can be used in M2M communication having potentially high delays and networks which are not continuously available. The TCP ports 1883 and 8883 are reserved for MQTT, the latter for encrypted communication using the TLS protocol.

MQTT differentiates between:

- Publisher: Device or service that sends the data, e. g. a sensor or a data concentrator.
- Subscriber: Device or service that processes the data, e. g. a visualization or a billing software.
- Broker: Central data hub for MQTT, it also manages the network and ensures robustness.

MQTT uses so-called topics to classify messages hierarchically. This can be compared to specifying a path on the file system. The publisher sends data of these topics to the Broker. This then distributes the data to the subscribers.

Certificates must be provided on the device for the encrypted connection via port 8883. Background information can be found in Section 8.6. Please ask your administrator in this case.

✔ Unencrypted MQTT requires the scheme mqtt:// at the beginning of the server address.

8.9.1 Example Azure cloud

For connecting to an Azure cloud, the parameters need to be set as follows:

- **Report address**: Internet address of the Azure cloud server
- **Report directory**: Device ID and topic for the Azure cloud
- **Report user name**: User name for the Azure cloud, usually consisting of internet address, device name and API version
- **Report password**: Password for the Azure cloud, usually a composition of access key, signature and expiration date

The following example should clarify the parameters:

- **Report address**: `SolvimusHub.azure-devices.net`
- **Report directory**: `devices/MUC063C/messages/events`
- **Report user name**: `SolvimusHub.azure-devices.net/MUC063C/?api-version=2018-06-30`
- **Report password**: `SharedAccessSignature sr=SolvimusHub.azure-devices.net%2fdevices%2f MUC063C&sig=rQXaVuN%2bjWqh0vVr9E6ybo7VbMBQ4QQNOidzMtoqI2g%3d&se=1639260907`

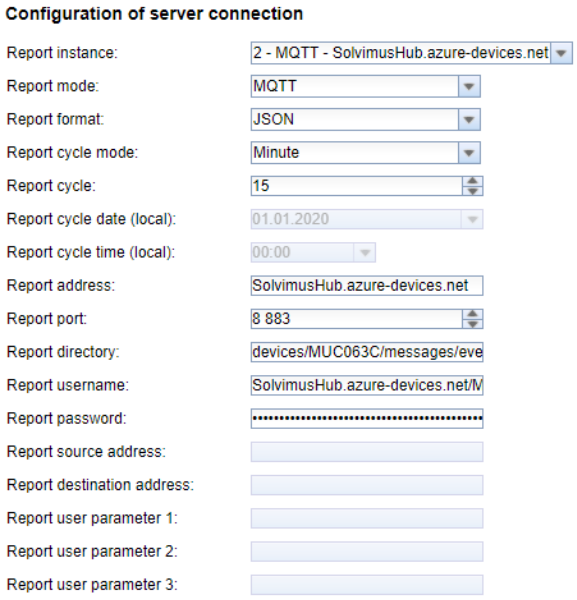


Figure 46: Sample configuration of a report to the Azure cloud

8.9.2 Example AWS cloud

For connecting to an AWS cloud, the parameters need to be set as follows:

- **Report address**: Internet address of the AWS cloud server
- **Report directory**: Device ID and topic for the AWS cloud
- **Report user name**: User name for the AWS cloud
- **Report password**: Password for the AWS cloud

The following example should clarify the parameters:

- **Report address**: `b-fbf31b71-1234-5678-a052-3b5a4fafabcd-1.mq.eu-central-1.amazonaws.com`
- **Report directory**: `demo201909/testing`
- **Report user name**: `demo201909`
- **Report password**: `YXcajMTbZ7WUBzrsst`

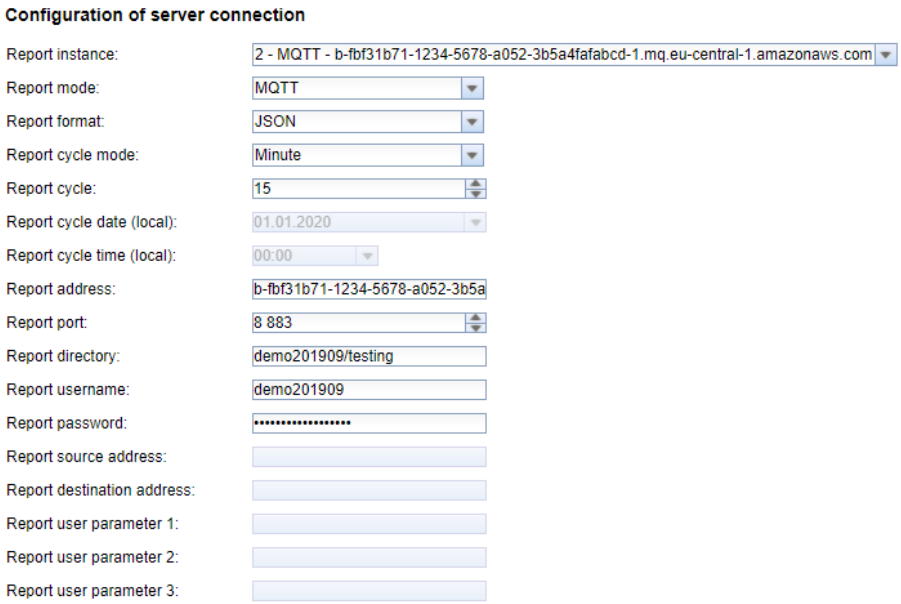


Figure 47: Sample configuration of a report to the AWS cloud

8.10 Local file storage

The metering data can also be stored directly on the device as a file. This allows downloading the data e. g. via FTP by a third party system. This is called a data pull.

Like for any other **Report mode**, the predefined formats and also the user-specific format can be chosen.

According to the parameters set for this report, the files are stored in the folder *ext/Log/YYYY/MM*. YYYY corresponds to the respective year and MM corresponds to the respective month of the report (according to the system time of the device).

The following settings, for example, will generate a CSV file containing all readings of the previous report period on the device every day at 01:00 local time:

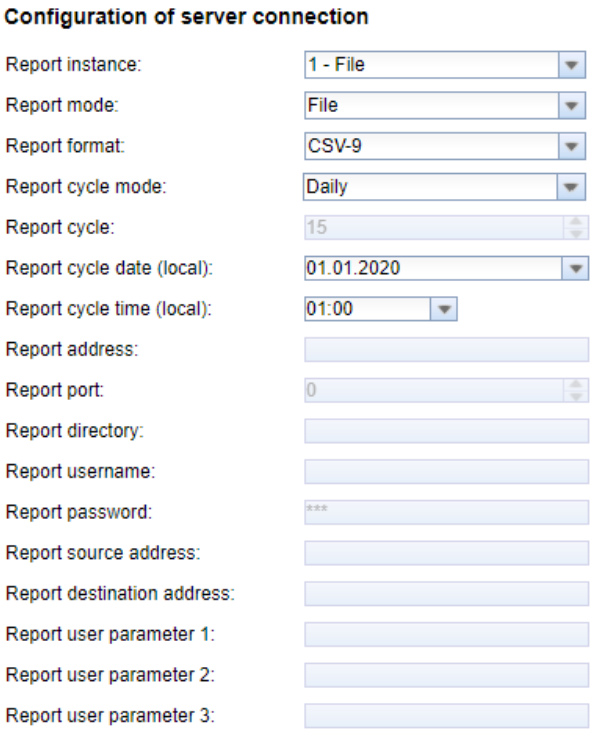


Figure 48: Sample configuration for storing file locally

8.11 Script-based report

If the options above do not fit or are not sufficient, the report can be switched to be script-based by setting **Report format** *User* in the ***Server*** tab.

This enables the user to use the powerful Linux tools installed on the device. Each instance is assigned its individual script. An overview is given in Section 10.7 and particularly in Section 10.7.2 showing an example.

Since the script-based report offers a lot of possibilities, additional parameters **Report user parameter 1**, **Report user parameter 2** and **Report user parameter 3** are available. Any kind of text can be entered here for each report instance. This information is then available during runtime of the script. The parameters of the report instance can be used in the script, but do not have to.

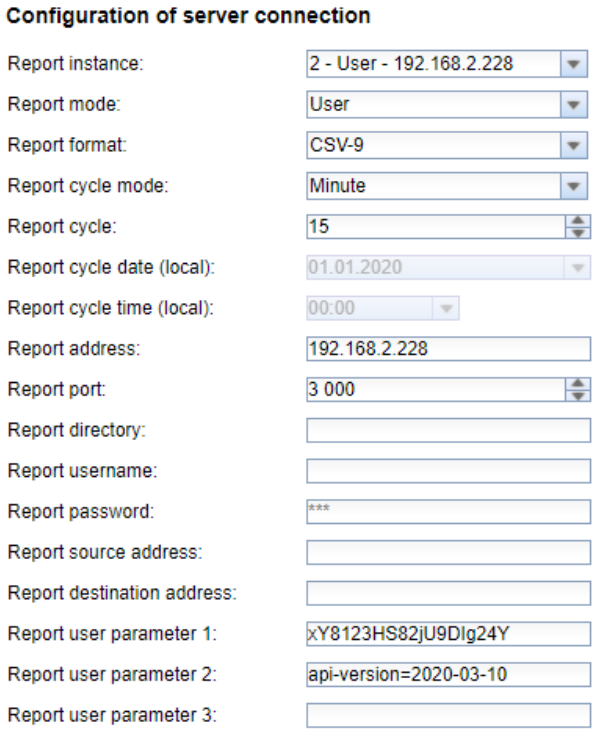


Figure 49: Sample configuration for reporting CSV data via a user script every 15 minutes

8.12 Troubleshooting the report

Troubleshooting the transfer of metering data is very complex. Typically, connectivity or authentication/encryption are the issues here. Indications of the reason or of the failure can be found in the ***Log*** tab.

Please check whether the remote station is available. For example the *ping* command from the SSH console of the device can be used for this purpose (see also Section 10.1.2). This will also check the name resolution (DNS). A host name should be converted to an IP address when pinging.

Please check whether a firewall blocks the data transmission or whether the routing is configured accordingly. Please ask your administrator in this case.

In the case of TLS encryption, please check whether all necessary certificates are available, especially the CA certificate for the remote station.

Please check the correct setting of **Report username** and **Report password** as well as **Report address**, **Report port** and **Report directory** of the respective instance.

If errors could not be eliminated, please contact our customer support:
Email: support@solvimus.de
Phone: +49 3677 7613065

8.13 Retry of a readout

The default behaviour in case of a failed report is as follows:

- If a report fails, e. g. because there is no connection to internet, it will be resent after 1/10 of **Report cycle time (local)** (see Table 12) or at least 10 minutes. This is reiterated till the report is sent successfully.
- For reports with a time interval according to **Report cycle mode** (see Table 12): The time interval of the report is not modified for the retry. If the connection is perturbed for a longer period, so that another report would have to be sent, it will be queued. It will be transmitted as soon as the original report could be sent. Thus, several reports can be sent consecutively.
- For reports according to *On Readout* for **Report cycle mode** (see Table 12): If several readouts pile up during the pertubation, the time period of the report will be extended. For repeated transmission attempts, the data of the new reports will be contained in the report.

The parameters *RETRY_INTERVAL*, *MIN_SEND_INTERVAL* and *MAX_BACKLOG* in the configuration file *chip.ini* (see Section 10.3) permit user-specific settings.

9 Provision of meter data via M-Bus

9.1 General information

The MBUS-GSLE requests the data of the configured M-Bus meters according to their readout intervals and provides the raw data for the transfer to the original master. Data of meters not configured in the MBUS-GSLE will not be transferred.

Direct access on the meters by the original M-Bus master does not occur. The readout interval and the readout mode of the M-Bus data (M-Bus-GetAll-Data) must be configured in the MBUS-GSLE so that these are compatible to the existing request of the original M-Bus master.

The M-Bus master can then request the data via REQ_UD2, whereby the MBUS-GSLE transfers the data.

A transfer of Multipage requests is not supported by the MBUS-GSLE. Consequently, the MBUS-GSLE replaces DIF:1F with DIF:0F.

The baud rate of the inner M-Bus network and of the outer M-Bus network can be set in the tab ***Configuration*** (see Section 4.6).

10 Advanced configuration options

10.1 Linux operating system

The devices from solvimus GmbH are based on the Linux operating system. This ensures that the devices continuously follow the state of the art and that errors in the software are quickly found and corrected due to a large community. It also ensures a certain basic functionality and security for the user.

The Linux operating system is built using the Yocto/openembedded build environment. All components are included according to the latest version and security patches. The Linux itself is unchanged except a few specific tools and customizations (e. g. solcmd). Corresponding Linux documentation can thus be used directly. For customer-specific projects, additional components provided on the Yocto/openembedded platform can be made available on the target system.

10.1.1 User roles and user rights

Linux supports and offers user roles. The operating system internally comes with the user *root* having full access to all operating system functions. In addition, further users with restricted access can be created. Their permissions can be set by groups and names, mostly file access permissions (read, write or execute).

In addition to the user *root*, the devices from solvimus GmbH are coming with the user *admin*. This user has read and write access to the partitions *app* and *ext* and can execute files there. For the operator, *admin* is the user who can completely configure the device.

- ✔ The user *web* is created as the default user for the web interface, but has no access rights to the file system.
- ✔ For reasons of downward compatibility, the user *ftp* is created as the default user for FTP access to the directory *ext/Log*.
- ℹ The user *root* has no external access to the device. This ensures security, privacy and safety for the operator. Only the user *admin* can grant access to the user *root*.
- ℹ The password of the user *root* is generated randomly and device-specific during production and stored access-protected in a database.

10.1.2 Command line

On the devices from solvimus GmbH, the Linux operating system offers a command line based on *BASH*. It allows the user and also other applications to execute commands via the command line.

The user can access the command line via an SSH console. The Netdiscover tool (see Chapter 3) opens an SSH console using a Putty client.

Standard commands

The Linux operating system and the command line *BASH* provide certain built-in standard commands. Examples are:

- *help*: Display list of all integrated commands
- *cd*: Navigation in the directory tree
- *ls*: List directory contents
- *cat*: View file contents
- *cp*: Copying files/directories
- *mv*: Move/rename files/directories
- *rm*: Delete files/directories
- *sync*: Write the data from the RAM buffer to the FLASH memory

- *chmod*: Adjust access rights
- *grep*: Search for text content
- *echo*: Output text
- *date*: Display system date and time
- *ps*: List all running processes
- *tail*: Display last lines of a file
- *netstat*: Query the status of the network interfaces
- *ping*: Network connectivity test
- *nslookup*: Display of the DNS configuration
- */sbin/ifconfig*: Overview of the network interfaces

Further commands are provided by programmes:

- *tcpdump*: Recording network traffic
- *openssl*: Using encryption, certificates and PKI
- *curl*: Retrieval and transmission of files via HTTP, FTP or SMTP/e-mail
- *socat*: Connecting two interfaces
- *vi*: Editing files
- *xsltproc*: Executing XSL transformations

solcmd command interpreter

Due to the system access rights for users, solvimus GmbH adds a command interpreter *solcmd*. It offers special application functions via the command line. The interpreter can be called with various parameters and thus provides access to the application and allows controlling it.

The following parameters are supported:

- *format-partition-app*: Format the configuration partition *app*
- *format-partition-ext*: Format the logging partition *ext*
- *config-partitions*: Reset the access rights to the partitions
- *config-users*: Activate changed user settings
- *config-hostname*: Activate changed device name
- *config-timezone*: Activate changed time zone settings
- *restart-eth0*: Restart the Ethernet interface
- *restart-wifi*: Restart the WIFI interface (only if WIFI is available)
- *filter-vlan*: VLAN filter for network interface (only if switch is integrated)
- *start-ppp0*: Establish the PPP dial-in connection (mobile network)
- *stop-ppp0*: Terminate the PPP dial-in connection (mobile network)
- *start-vpn*: Establish a VPN connection (OpenVPN)
- *stop-vpn*: Terminate a VPN connection (OpenVPN)
- *manual-vpn*: Establish a VPN connection (OpenVPN) in the foreground, e. g. for entering the password manually
- *restart-server*: Restart the server services
- *regenerate-server-keys*: Re-create the keys for secured server services
- *start-solapp*: Start the main application
- *stop-solapp*: Stop the main application
- *start-transparent-tty*: Activate transparent data forwarding of a serial interface to an Ethernet port
- *stop-transparent-tty*: Deactivate transparent data forwarding of a serial interface to an Ethernet port
- *start-virtual-tty*: Activate a virtual interface via an Ethernet port

- *stop-virtual-tty*: Deactivate a virtual interface via an Ethernet port
- *update-rtc*: Write the system time to the buffered real-time clock
- *factory-reset*: Reset the device to factory settings
- *update-system*: Perform a system update
- *reboot-system*: Restart the system
- *help*: Command overview with explanation and examples

10.2 Update

The firmware can be updated manually or conveniently via the web interface (see Section 4.12).

For a manual update, access via SSH is necessary. In preparation, the easiest way to provide the update file on the system is to upload it via SFTP. The tools are provided by the Netdiscover tool (see Chapter 3).

First, the appropriate and signed update file **.enc* has to be uploaded via SFTP into the directory *ext/Upd* (see Section 3.5). This is restricted to the user *admin*.

After uploading the file, the user has to log in as *admin* via SSH (see Section 3.6). In the command line (see Section 10.1.2), the command *solcmd update-system* has to be executed then. After completion, a reboot is necessary. This is triggered by the command *solcmd reboot-system*.

10.3 Configuration file chip.ini

The file *app/chip.ini* contains the general system parameters and is therefore the central configuration file. The parameters are grouped into different sections. If the parameters are not configured in *chip.ini*, the default values are used.

- The device needs to be rebooted after changing the file *chip.ini* manually for taking effect. Reboot can be triggered via the web-based front end using the button **Reboot system** in the ***Service*** tab or via the command line.
- Manual changes are permanently stored on the flash not instantly, but after a few minutes. As a result, changes are possibly lost if the device is rebooted by switching the power supply off and on.
- A range „0, 1“ without further explication means: 0 = inactive/no, and 1 = active/yes.
- The file *chip.ini* can be transferred to other devices via FTPS. Some settings like the network configuration (e. g. different IP address) needs to be taken into account.

Parameter	Description	Range	Standard
Group [IP]			
ADDRESS	IP address of the device	0.0.0.0-255.255.255.255	192.168.1.101 (explicit)
DHCP	Activation of the DHCP client	0, 1	0 (explicit)
DHCP_HOSTNAME	Host name to log on to the DHCP server	Text, max. 255 characters, %SERIAL%: MAC address of the device	Name of the device from group [DEVICE]
GATEWAY	IP address of the gateway	0.0.0.0-255.255.255.255	192.168.1.254 (explicit)
NETMASK	Subnet mask of the device	0.0.0.0-255.255.255.255	255.255.255.0 (explicit)
Group [DEVICE]			
NAME	Name of the device in the tool Netdiscover	Text, max. 50 characters	*Product name* (explicit)
TIMEZONE	Time zone of the device	Text, max. 255 characters	Universal, corresponds to GMT
Group [DNS]			
NAME_SERVER1	IP address of the primary DNS server, IP or host name	Text, max. 255 characters	Not set
NAME_SERVER2	IP address of the secondary DNS server, IP or host name	Text, max. 255 characters	Not set
Group [VPN]			
CONFIGFILE	Path to the OpenVPN configuration file	Text, max. 255 characters	vpn/config.ovpn
ENABLE	Activation of the OpenVPN client	0, 1	0

Continued on next page

Table 24 – *Continued from previous page*

Parameter	Description	Range	Standard
Group [WEB]			
CERT_COMMON_NAME	The fully qualified domain name	Text, max. 255 characters	Not set
CERT_COUNTRY	Country code	Text, max. 255 characters	Not set
CERT_LOCATION	Location/city	Text, max. 255 characters	Not set
CERT_ORGANISATION	Legal name of the organisation	Text, max. 255 characters	Not set
CERT_ORGANISATION_UNIT	Unit/department	Text, max. 255 characters	Not set
CERT_STATE	State or region	Text, max. 255 characters	Not set
HTTP_ENABLE	Activation of the HTTP server	0, 1	1
HTTPS_ENABLE	Activation of the HTTPS server	0, 1	1
HTTP_PORT	Network port of the HTTP server	0-65535	80
HTTPS_PORT	Network port of the HTTPS server	0-65535	443
Group [FTP]			
CERT_COMMON_NAME	The fully qualified domain name	Text, max. 255 characters	Not set
CERT_COUNTRY	Country code	Text, max. 255 characters	Not set
CERT_LOCATION	Location/city	Text, max. 255 characters	Not set
CERT_ORGANISATION	Legal name of the organisation	Text, max. 255 characters	Not set
CERT_ORGANISATION_UNIT	Unit/department	Text, max. 255 characters	Not set
CERT_STATE	State or region	Text, max. 255 characters	Not set
ENABLE	Activation of the FTP server	0, 1	1
Group [SSH]			
ENABLE	Activation of the SSH server	0, 1	1
Group [UDPCFG]			
ENABLE	Activation of the UDP-based search and configuration protocol	0, 1	1
IPCFG_PASSWORD	Password for the modification of the IP address via the UDP configuration protocol	Text, max. 255 characters	Not set
Group [SOLVIMUS]			
BACNET_BBMD	IP of the BACnet BBMD (BACnet Broadcast Management Device)	Text, max. 255 characters	Not set
BACNET_BROADCAST	BACnet Broadcast IP address (system configuration will be used if not set)	Text, max. 255 characters	Not set
BACNET_CONFIGURE NETWORK	Activation of a BACnet-specific network configuration (additional IP address)	0, 1	0
BACNET_DEVICEID	BACnet device ID	1-4294967295	1
BACNET_DEVICENAME	BACnet device name	Text, max. 255 characters	Not set
BACNET_ENABLE	Activation of the BACnet communication	0, 1	0
BACNET_IP	BACnet IP (system configuration will be used if not set)	Text, max. 255 characters	Not set
BACNET_LOCATION	BACnet location information	Text, max. 255 characters	metering
BACNET_NETMASK	BACnet Network mask (system configuration will be used if not set)	Text, max. 255 characters	Not set
BACNET_PORT	BACnet network port	0-65535	47808
DLDERS_ADDRESS DISABLE	DLDE request with meter serial number (=0) respectively wildcard request (=1). In the latter case only 1 meter is permitted.	0, 1	0
DLDERS_BAUDRATE	Baud rate for the serial DLDE communication	300, 600, 1200, 1800, 2400, 4800, 9600, 19200, 38400, 57600, 115200, 230400, 460800	9600

Continued on next page

Table 24 – *Continued from previous page*

Parameter	Description	Range	Standard
DLDERS_DATABITS	Data bits for the serial DLDE communication	7, 8	7
DLDERS_DEVPATH	Linux path for the serial DLDE communication	Text, max. 255 characters	Not set
DLDERS_ENABLE	Activation of the serial DLDE interface	0, 1	0
DLDERS_FIRSTTIMEOUT	Request mode: timeout for initial reception of data from meter. Push mode: time without registration of data (Wait idle, in ms)	0-65535	3000
DLDERS_FIXEDLAYOUT		0, 1	0
DLDERS_ FLOWCONTROL	Handshake for the serial DLDE communication: 0: none, 1: XON/XOFF when sending, 2: RTS/CTS, 8: XON/XOFF when receiving, 9: XON/XOFF when sending and receiving	0, 1, 2, 8, 9	0
DLDERS_FULLTIMEOUT	Maximum timeout for reading a meter (in ms)	0-65535	30000
DLDERS_IDLETIMEOUT	Idle time for detection of the end of communication (in ms)	0-65535	100
DLDERS_ LOADPROFILE_ MAXRDAYS		0-65535	366
DLDERS_ LOADPROFILE_ SKIPINVALIDENTRY		0, 1	0
DLDERS_MODE	Communication mode for the serial DLDE interface	REQUEST, REQUEST_ECHO, PUSH	REQUEST_ECHO
DLDERS_PARITY	DLDE parity: 0: none, 1: odd, 2: even, 3: mark, 4: space	0-4	2
DLDERS_RAWLOG ENABLE	Activation of the raw data logging to the directory *ext/*	0, 1	0
DLDERS_RS485ENABLE	Activation of the RS-485 interface for the DLDE communication	0, 1	1
DLDERS_SMLENABLE	Activation of processing SML protocol data	0, 1	0
DLDERS_STOPBITS	Stop bits for the serial DLDE interface	1, 2	1
DLDERS_TRANSPARENT	Activation of the transparent transmission of the serial DLDE interface to a network port: NONE: transmission deactivated, TCP: transmission of a TCP port, UDP: transmission to a UDP port	NONE, TCP, UDP	NONE
DLDERS_TRANSPARENT PORT	Network port for the transparent transmission via TCP or UDP	0-65535	0
FASTRESCAN_TIME	Cycle time for updating the temporary meter list of received wM-Bus meters (in s)	1-4294967295	60
I2C_DEBUGOUT	Activation of raw data output for the internal I2C communication in the system log	0, 1	0
MBMSTMETER_ BAUDRATE	Baud rate for the serial Modbus communication (Master RTU)	300, 600, 1200, 1800, 2400, 4800, 9600, 19200, 38400, 57600, 115200, 230400, 460800	19200
MBMSTMETER_ DATABITS	Data bits for the serial Modbus communication (Master RTU)	7, 8	8
MBMSTMETER_ MAXRETRY	Number of retries for a Modbus request to the meter (Master RTU)	0-255	3

Continued on next page

Table 24 – *Continued from previous page*

Parameter	Description	Range	Standard
MBMSTMETER_PARITY	Parity of the serial Modbus communication (Master RTU): 0: none, 1: odd, 2: even, 3: mark, 4: space	0-4	0
MBMSTMETER_ STOPBITS	Stop bits for the serial Modbus communication (Master RTU)	1, 2	1
MBMSTMETER_ SERIALENABLE	Activation of the serial Modbus (Master RTU)	0, 1	0
MBMSTMETER_ SILENTINTERVAL	Timeout between two bytes in a data packet / a response (Master RTU, in ms)	0-65535	20
MBMSTMETER_ TCPCONNECTTIMEOUT	Timeout for a connection to a Modbus TCP meter (in ms)	1-4294967295	5000
MBMSTMETER_ TIMEOUT	Timeout for the response of the meter (Master RTU, in ms)	0-65535	500
MBUS_ALLOWINSECURE	Deactivates the authentication check when decrypting	0, 1	0
MBUS_BAUDRATE	Baud rate for the M-Bus communication	300, 600, 1200, 1800, 2400, 4800, 9600, 19200, 38400, 57600, 115200, 230400, 460800	2400
MBUS_DATABITS	Data bits for the M-Bus communication	7, 8	8
MBUS_DEVPATH	Linux path for the M-Bus interface	Text, max. 255 characters	Not set
MBUS_DISABLE DECRYPTION	Deactivation of decrypting the M-Bus packets (status field)	0, 1	0
MBUS_ENABLE	Activation of the M-Bus interface	0, 1	1
MBUS_FIRST FCBBIT_NEG	Begins reading the M-Bus meters with a specific FCB-bit value: 0: first FCB-bit set, 1: first FCB-bit not set	0, 1	0
MBUS_FIXEDLAYOUT		0, 1	0
MBUS_FLOWCONTROL	Handshake for the M-Bus communication: 0: none, 1: XON/XOFF when sending, 2: RTS/CTS, 8: XON/XOFF when receiving, 9: XON/XOFF when sending and receiving	0, 1, 2, 8, 9	0
MBUS_FORCE	Compatibility mode for reading of faulty M-Bus meters, emulates correct ACK	0-2	0
MBUS_FREEZE STORAGENUM	Storage number for Freeze meter data	0-4294967295	0
MBUS_FULLTIMEOUT	Maximum timeout for reading a meter (in ms)	0-65535	10000
MBUS_IDLETIMEOUT	Idle time for detection of the end of communication (in ms)	0-65535	100
MBUS_IGNORECRCFIELD	Compatibility mode for reading faulty M-Bus meters, disregards the CRC field	0, 1	0
MBUS_IGNORELENGTH FIELD	Compatibility mode for reading faulty M-Bus meters, disregards the length field	0, 1	0
MBUS_LOADPROFILE MANUFACTURER	Manufacturer code for identification of load profile meters, according to M-Bus standard: „EMH“=(0xA8 0x15) → 0x15A8=5544	0-65535	5544
MBUS_LOADPROFILE MAXCOUNT	Number of load profile entries initially requested by the meter	1-65535	65535
MBUS_LOADPROFILE MODE	Activation of load profile readings for electricity meters via M-Bus	DISABLED, DIZH, DIZG	DISABLED
MBUS_MAXMULTIPAGE	Limits the number of Multipage requests	0-255	3
MBUS_MAXPRIMARY ADDRESS	Upper address for the M-Bus primary search	0-250	250

Continued on next page

Table 24 – *Continued from previous page*

Parameter	Description	Range	Standard
MBUS_MAXRETRY	Number of retries for an M-Bus or Multipage request	0-255	3
MBUS_MINPRIMARY ADDRESS	Lower address for the M-Bus primary search	0-250	0
MBUS_NOADDRESS VERIFY	Deactivation of the address verification in primary addressing	0, 1	0
MBUS_PARITY	Parity of the M-Bus communication: 0: none, 1: odd, 2: even, 3: mark, 4: space	0-4	2
MBUS_RAWLOGENABLE	Activation of the raw data logging to the directory *ext/*	0, 1	0
MBUS_REQUESTMODE	Request mode	ALL, EXT, ONLY, FREEZE	ONLY
MBUS_RESETMODE	Reset Modes: 0: NKE after Select, 1: NKE before Select 2: No NKE 3: NKE at 0xFD and NKE at 0xFF before the communication 4: NKE at 0xFD, application reset at 0xFF and NKE at 0xFF before the communication	0-4	0
MBUS_RS485ENABLE	Activation of the RS-485 interface for the M-Bus communication	0, 1	0
MBUS_SCANMODE	Search algorithm for the M-Bus	PRIMARYSCAN, SECONDARYSCAN, SECONDARYSCAN ALLOC, SECONDARYSCAN REVERSE, SECONDARYSCAN ALLOCREVERSE	SECONDARYSCAN
MBUS_SECMASK MANUFACTURER	Predefined manufacturer ID for the secondary search	Precisely 4 characters, each 0-9/A-F	0xFFFF
MBUS_SECMASK MEDIUM	Predefined medium ID for the secondary search	Precisely 2 characters, each 0-9/A-F	0xFF
MBUS_SECMASKSERIAL	Secondary search for the meter serial number	Precisely 8 characters, each 0-9/A-F	0xFFFFFFFF
MBUS_SECMASK VERSION	Predefined version number for the secondary search	Precisely 2 characters, each 0-9/A-F	0xFF
MBUS_SELECTMASK	Ignoring of selected ranges, for these placeholders are used (setting via bit mask): +1: serial number +2: manufacturer +4: version field +8: medium	0-15	14
MBUS_SMLENABLE	Activation of processing SML protocol data	0, 1	0
MBUS_SOCPAGESELECT ENABLE	Activates Pageing according to the specification of the company Socomec	0, 1	0
MBUS_SOC MANUFACTURER	Manufacturer code for identification of meters with Socomec pageing, according to M-Bus standard: „SOC“=(0xE3 0x4D) → 0x4DE3=19939	0-65535	19939
MBUS_SPXMETER CONVERT	Activation of manufacturer-specific decoding (manufacturer code SPX)	0, 1	0
MBUS_STOPBITS	Stop bits for the M-Bus communication	1, 2	1
MBUS_TIMEOUT	Timeout till first data are received from the meter (in ms)	0-65535	2000

Continued on next page

Table 24 – *Continued from previous page*

Parameter	Description	Range	Standard
MBUS_TRANSPARENT	Activation of the transparent transmission of the M-Bus interface to a network port or an M-Bus slave interface: NONE: transmission deactivated, MBUS: Master TCP: transmission to a TCP port, UDP: transmission to a UDP port, TCP_ONDEMAND: Master & Transparent/TCP	NONE, MASTER, TCP, UDP, TCP_ONDEMAND	NONE
MBUS_TRANSPARENT PORT	Network port for the transparent transmission via TCP or UDP	0-65535	0
MBUS_WAKEUPENABLE	Activation of the specific wakeup requests	0, 1	0
MBUSSLV_BAUDRATE	Baud rate for the M-Bus slave communication	300, 600, 1200, 1800, 2400, 4800, 9600, 19200, 38400, 57600, 115200, 230400, 460800	2400
MBUSSLV_DATABITS	Data bits for the M-Bus slave communication	7, 8	8
MBUSSLV_DEBUGOUT	Activation of the raw data output for the M-Bus slave communication in the system log	0, 1	0
MBUSSLV_DEVPATH	Linux path for the M-Bus slave interface	Text, max. 255 characters	Not set
MBUSSLV_ FLOWCONTROL	Handshake for the M-Bus slave communication: 0: none, 1: XON/XOFF when sending, 2: RTS/CTS, 8: XON/XOFF when receiving, 9: XON/XOFF when sending and receiving	0, 1, 2, 8, 9	0
MBUSSLV_ FULLTIMEOUT	Maximum timeout for the request of a meter (in ms)	0-65535	10000
MBUSSLV_ IDLETIMEOUT	Idle time for detection of the end of communication (in ms)	0-65535	100
MBUSSLV_PARITY	Parity for the M-Bus slave communication: 0: none, 1: odd, 2: even, 3: mark, 4: space	0-4	2
MBUSSLV_RS485ENABLE	Activation of the RS-485 interface for the M-Bus slave communication	0, 1	0
MBUSSLV_STOPBITS	Stop bits for the M-Bus slave communication	1, 2	1
MBUSSLVMETER_MODE	Activation of the M-Bus slave interface: DEFAULT: product-specific activated, NONE: deactivated, TCP: activation via TCP port, UDP: activation via UDP port, MBUS: activation via the M-Bus slave interface	DEFAULT, NONE, TCP, UDP, MBUS	DEFAULT
MBUSSLVMETER_PORT	Network port for access to the M-Bus slave interface via TCP or UDP	0-65535	5040
MBUSSLVMETER_ WMBUSALLOW ENCRYPTED	Activation of the transfer of encrypted wM-Bus meters via the M-Bus slave interface	0, 1	0
MBUSSLVMETER_ WMBUSALLOW EXTENDEDHEADER	Activation of the transfer of specific wM-Bus header data (e. g. AFL/ELL) via the M-Bus slave interface	0, 1	0
MBUSSLVMETER_ WMBUSALLOWOTHER	Activation of the transfer in spite of unknown wM-Bus header data via the M-Bus slave interface	0, 1	0

Continued on next page

Table 24 – *Continued from previous page*

Parameter	Description	Range	Standard
MBUSSLV2METER_ MODE	Activation of the second M-Bus slave interface: NONE: deactivated, TCP: activation via a TCP port, UDP: activation via a UDP port	NONE, TCP, UDP	NONE
MBUSSLV2METER_PORT	Network port for access to the second M-Bus slave interface via TCP or UDP	0-65535	5050
MBUSSLV2METER_ WMBUSALLOW ENCRYPTED	Activation of the transfer of encrypted wM-Bus meters via the second M-Bus slave interface	0, 1	0
MBUSSLV2METER_ WMBUSALLOW EXTENDEDHEADER	Activation of the transfer of specific wM-Bus header data (e. g. AFL/ELL) via the second M-Bus slave interface	0, 1	0
MBUSSLV2METER_ WMBUSALLOWOTHER	Activation of the transfer in spite of unknown wM-Bus header data via the second M-Bus slave interface	0, 1	0
METER_ADJUST TIMESTAMPS			0
METER_CYCLEMODE			SECOND
METER_CYCLE TIMESTAMP			Not set
METER_DELAY	Delay for reading of meter data according to the configured reading cycle (in s)	0-4294967295	0
METER_PRESENT VALUESONLY			0
METER_MAXALLVALUE COUNT	Limitation of total meter data (0: no limitation)	0-65535	0
METER_MAXDEVICE COUNT	Limitation of the number of meters (0: no limitation)	0-65535	500
METER_MAXVALUE COUNT	Limitation of meter values per meter (0: no limitation)	0-65535	25
METER_OBISGEN	Automatic generation of OBIS codes for meter values from DIF/VIF codes when creating M-Bus and wM-Bus meters 0: off, 1: on	0, 1	0
METER_RETRYDIVIDER	Reduces the quantity of values read and used for reporting. Only values every METER_ RETRYDIVIDER are retained for reporting. All read values are used for visualisation and for other interfaces (Modbus resp. BACnet).	0-65535	0
METER_STAT_CONFIG	Path to the meter configuration file	Text, max. 255 characters	app/device_handle.cfg
METER_TIME	Cycle time for reading meters (unit according to METER_ CYCLEMODE), caution: with small cycle times and a large quantity of meters, significant log files can be created	1-4294967295	900
METER_ VIFSTRINGMODE	Placement of the VIF string in the data flow: 0: VIF string after last VIFE, 1: VIF string immediately after VIF string identification	0, 1	1
METERSYSTEM_ ENABLE	Activation of the system meter function	0, 1	1
METERSYSTEM_SCRIPT TIMEOUT	Timeout after whose expiration the system meter scripts are aborted (in s)	0-65535	0
MODBUS_ADDRESS	Primary Modbus address resp. Unit identifier	0-255	0
MODBUS_APPLICATION	Application information within the device identification	Text, max. 255 characters	Modbus TCP Gateway

Continued on next page

Table 24 – *Continued from previous page*

Parameter	Description	Range	Standard
MODBUS_BAUDRATE	Baud rate for the serial Modbus communication (RTU)	300, 600, 1200, 1800, 2400, 4800, 9600, 19200, 38400, 57600, 115200, 230400, 460800	19200
MODBUS_CONNECTION TIMEOUT	Timeout of the Modbus TCP connection (in s)	0-65535	60
MODBUS_DATABITS	Data bits for the serial Modbus communication (RTU)	7, 8	8
MODBUS_DEBUGOUT	Activation of raw data output for the Modbus communication in the system log	0, 1	0
MODBUS_DEVPATH	Linux path for the serial Modbus interface	Text, max. 255 characters	Not set
MODBUS_DISCONNECT TIMEOUT	Timeout after whose expiration inactive Modbus TCP connections are aborted (in s)	0-1000	60
MODBUS_ENABLE	Activation of the Modbus slaves	0, 1	0
MODBUS_ FLOWCONTROL	Handshake for the serial Modbus communication: 0: none, 1: XON/XOFF when sending, 2: RTS/CTS, 8: XON/XOFF when receiving, 9: XON/XOFF when sending and receiving	0, 1, 2, 8, 9	0
MODBUS_IP			Not set
MODBUS_ MAXCONNECTIONS	Maximum number of parallel Modbus TCP connections	0-80	5
MODBUS_MODE		Serial, TCP, UDP	TCP
MODBUS_MODEL	Device information within the device identification	Text, max. 255 characters	Standard
MODBUS_NWPORT	Network port of the Modbus TCP slaves	0-65535	502
MODBUS_PARITY	Parity of the serial Modbus communication: 0: none, 1: odd, 2: even, 3: mark, 4: space	0-4	0
MODBUS_PRODUCT CODE	Device code for the Modbus function „Read Device Identification“	Text	A code defined by solvimus GmbH and dependent on the device is returned.
MODBUS_RS485ENABLE	Activation of the RS-485 interface for the serial Modbus communication (RTU)	0, 1	0
MODBUS_SPAN			1
MODBUS_STOPBITS	Stop bits for the serial Modbus communication (RTU)	1, 2	1
MODBUS_VENDOR	Manufacturer information within the device identification	Text, max. 255 characters	[Branding]
MODBUS_VENDORURL	Website information on manufacturer within the device identification	Text, max. 255 characters	[Branding]
MODBUS_VERSION	Version of the firmware indicated by Modbus within the device identification. If not set explicitly, it corresponds to the software version on the configuration page.	Text, max. 255 characters	-
MODBUS_WRITEACCESS			READONLY
MODBUSMETER_ PROTOCOLVERSION	Protocol version of the Modbus meter data: Bit 0: 2 registers per value (only floating point value), Bit 1: Multislave activated, Bit 2: Word-Swapping of 32-Bit floating point values, Bit 3: Dummy mode activated	0-16	0
MUC_CONFIG_VER	Version of the configuration, corresponding to the firmware version that it had saved. Set exclusively by the application.	0-65535	-

Continued on next page

Table 24 – *Continued from previous page*

Parameter	Description	Range	Standard
MUC_ FORCESTOREREADOUT	Database mode to „Store meter values“ (see Table 10) 0: automatic 1: on	0, 1	0
MUC_LOG	Sets the level of system output via system log	DEFAULT, NONE, ERRORONLY, ALL	DEFAULT
MUC_LOGCYCLE DIVIDER	This parameter enables that not all readouts are written to the database and transferred into the reports. For example, if this parameter equals 4 when fixing *Readout cycle* to 15 minutes, only every fourth value will be written to the database and the report lists only one value per hour. This allows smaller *Readout cycle*, e. g. for Modbus or BACnet resp. for display on the web page. A value of 0 deactivates this function.	0-65535	0
MUC_METER DESCRIPTION_ ENABLEFLAGS	Enable Flags for representing the description on the website: Bit 0: Description Bit 1: Storage number, tariff, value type Bit 2: DIF/VIF raw data Bit 3: All raw data of the data value entry	0 - 16	1
MUC_PASS_ENCMODE	Activation of the encryption of the passwords in the configuration files: 0: no encryption, 1: encryption without MAC, 2: encryption with MAC	0, 1, 2	0
MUC_REPORT FATALREBOOTTIMEOUT			0
MUC_REPORT SCRIPTABORTTIMEOUT			30
MUC_SCALEVALUES	Scaled values within the CSV and XML log data	0, 1	0
MUC_SETDEVICES	Activation of setting the meter values. The setting of meter values must be supported by the meters. INTERNAL: S0 and digital outputs of the system meter, INTERNALORDIGTALOUT: S0 and digital outputs, ALL: all meter values, NONE: no meter values	INTERNAL, INTERNALORDIGTAL-OUT, ALL, NONE	INTERNAL
MUC_SETDEVICETIME			0
MUC_SHOWDATAFRAME	Explicit listing of the raw data frame as meter value, for Multi-page meters one entry is added per frame	0, 1	0
MUC_SHOWMETER STATUSBYTE	Explicit listing of the status byte of the meter (M-Bus and wM-Bus) as meter value	0, 1	0
MUC_SHOWTIMESTAMP ENTRIES	Explicit representation of the timestamps of a meter	0, 1	0
MUC_SHOWVENDOR RAWDATA	Explicit listing of manufacturer-dependent data as meter value	0, 1	0
MUC_SHOWVENDOR RAWDATAWEB	Representation of binary data on the website (manufacturer-dependent resp. data container)	0, 1	0
MUC_SHOWWMBUS RSSIVALUE			0
MUC_TRIMVALUES			0
MUC_USE_FREEZE	Activation of the Freeze command for reading meters	0, 1	0

Continued on next page

Table 24 – *Continued from previous page*

Parameter	Description	Range	Standard
SHOW_KEYS	Show decrypted data on the website	0, 1	1
SNTP_ENABLE	Activation of the reference via SNTP server	0, 1	1
SNTP_REQTIMEOUT	Timeout for a SNTP request (in ms)	1-65535	15000
SNTP_RETRY	Number of retries for a SNTP request	0-255	2
SNTP_TIMEOUT	Timeout for a renewed SNTP time query (explicit, in s)	1-4294967295	86400
SNTPIP	Address of the time server (SNTP)	Text, max. 255 characters	pool.ntp.org
SNULL_ENABLE	Activation of the S0 interface	0, 1	0
SNULL_MODE	Counting mode for S0	RELATIVE, ABSOLUTE	RELATIVE
WAN_APN	Access point for WAN	Text, max. 255 characters	Not set
WAN_AUTH	Authentication procedure for accessing WAN	NONE, PAP, CHAP	CHAP
WAN_BAUDRATE	Baud rate for WAN communication	300, 600, 1200, 1800, 2400, 4800, 9600, 19200, 38400, 57600, 115200, 230400, 460800	115200
WAN_DATABITS	Data bits for the WAN communication	7, 8	8
WAN_DEBUGOUT	Activation of raw data output for the WAN communication in the system log 0, none: off (default), 1, basic: display of the AT communication and of the power cycles, 2, extended: as 1 and additional state requests at the modem like e. g. SIM card settings for preferred providers, 3, all: as 2 and additional Raw binary communication data and parsed replies	0, 1, 2, 3	0
WAN_DEVPATH	Linux path for the WAN interface	Text, max. 255 characters	Not set
WAN_ENABLE	Activation of the WAN communication (mobile radio)	0, 1	0
WAN_FLOWCONTROL	Handshake for the WAN communication: 0: none, 1: XON/XOFF when sending, 2: RTS/CTS, 8: XON/XOFF when receiving, 9: XON/XOFF when sending and receiving	0, 1, 2, 8, 9	0
WAN_FULLTIMEOUT			0
WAN_IDLETIMEOUT			0
WAN_MAXRETRY	Number of retries for establishing the WAN connection (0: no limit)	0-255	0
WAN_OLDBAUDRATE	Baud rate for the WAN communication, affects only older devices (0: inactive)	0, 300, 600, 1200, 1800, 2400, 4800, 9600, 19200, 38400, 57600, 115200, 230400, 460800	0
WAN_PARITY	Parity of the WAN communication: 0: none, 1: odd, 2: even, 3: mark, 4: space	0-4	0
WAN_PASSWORD	Password to access WAN	Text, max. 255 characters	Not set
WAN_PIN	PIN for the SIM card	Text, max. 255 characters	Not set
WAN_PROVIDER			Not set
WAN_PUK	PUK for the SIM card	Text, max. 255 characters	Not set

Continued on next page

Table 24 – *Continued from previous page*

Parameter	Description	Range	Standard
WAN_RECONNECT MONITOR	Mode for the monitoring of the radio connection and forced disconnection as well as renewal of the radio connection	OFF, WAN_ACTIVITY, REPORT_ANY, REPORT_ALL, REPORT_SPECIFIC, PING	OFF
WAN_RECONNECT PINGHOST	Host/IP-address which is monitored	String	-
WAN_RECONNECT PINGINTERVAL	Interval in which a ping is sent (in s)		1800
WAN_RECONNECT PINGTIMEOUT	Timeout for the reception of a response (in ms)		30000
WAN_RECONNECT REPORTINSTANCE	Number of the report selected for monitoring. Only active if WAN_RECONNECTMONITOR = REPORT_SPECIFIC	1 to number of supported reports (integer)	1
WAN_RECONNECT TIMEOUT	Interval which is monitored (in seconds). If no response on a ping is received within this limit, another attempt to establish the connection will be undertaken.	1800-4294967295	86400
WAN_RS485ENABLE	Activation of the RS-485 interface for WAN communication	0, 1	0
WAN_RSSITEST			0
WAN_STOPBITS	Stop bits for the WAN communication	1, 2	1
WAN_TECHNOLOGY	Selected radio technology. The preset mode DEFAULT is interpreted as the intended resp. reasonable value according to the modem type. If the selected mode is not supported by the modem (e. g. LTE on NB-IoT), an error is logged and the modem remains in the prior state.	DEFAULT, LTE, GSM, UMTS, NBIOT, CATM, LTE_GSM, LTE_UMTS, UMTS_GSM, LTE_UMTS_GSM	DEFAULT
WAN_USER	Username for accessing WAN	Text, max. 255 characters	Not set
WATCHDOG_IDLE	Watchdog timeout for the idle state (in s)	1-4294967295	120
WATCHDOG_PROCESS	Watchdog timeout in the busy state (in s)	1-4294967295	900
WATCHDOG_READOUT	Watchdog timeout during readout (in s)	1-4294967295	Quadruple of the readout cycle, at least: WATCHDOG_PROCESS
WATCHDOG_SCAN	Watchdog timeout during scanning (in s)	1-4294967295	43200000
WEBCOM_PASSWORD PATTERN			Not set
WEBCOM_ ADMINLOGIN_ SWITCHREQ		0, 1	1
WEBCOM_USESWITCH			Not set
WEBCOM_TIMEOUT	Timeout for a web session after automatic logout of a user (in ms)	1-4294967295	60000
WMBUS_ALLOW INSECURE			0
WMBUS_BAUDRATE	Baud rate for the wM-Bus communication	300, 600, 1200, 1800, 2400, 4800, 9600, 19200, 38400, 57600, 115200, 230400, 460800	19200
WMBUS_CACHESIZE	wM-Bus cache size for temporary storage of received meter data	1-500	500
WMBUS_CACHE TIMEOUT	Storage time of received wM-Bus packets in the cache list (in s, 0: no limit)	0-4294967295	0
WMBUS_DATABITS	Data bits for the wM-Bus communication	7, 8	8
WMBUS_DECRYPTUSE LINKLAYERID			0
WMBUS_DEVPATH	Linux path of the wM-Bus interface	Text, max. 255 characters	Not set
WMBUS_FIXEDLAYOUT		0, 1	0

Continued on next page

Table 24 – *Continued from previous page*

Parameter	Description	Range	Standard
WMBUS_FLOW CONTROL	Handshake for the wM-Bus communication: 0: none, 1: XON/XOFF when sending, 2: RTS/CTS, 8: XON/XOFF when receiving, 9: XON/XOFF when sending and receiving	0, 1, 2, 8, 9	0
WMBUS_FULLTIMEOUT	Maximum time (in ms) for a „packet“ in the transparent mode of the wM-Bus which will be transmitted via TCP/UDP in a consolidated form. The Idle Timeout defined by WMBUUS_ IDLETIMEOUT is respected.	0-65535	1000
WMBUS_IDLETIMEOUT	Idle time (in ms) after which the „packet“ in the transparent mode of the wM-Bus, which will be transmitted via TCP/UDP in a consolidated form, is regarded as completed.	0-65535	20
WMBUS_MODE	Mode of the wM-Bus module	S, T, C, C_T	C_T
WMBUS_NETWORK_ ROLE	Function of the wM-Bus interface	DISABLED, MASTER, SLAVE	MASTER
WMBUS_PARITY	Parity of the wM-Bus communication: 0: none, 1: odd, 2: even, 3: mark, 4: space	0-4	0
WMBUS_ RAWDATAINCLUDERSSI		0, 1	0
WMBUS_RAWLOG ENABLE	Activation of the raw data logging to the directory *ext/*	0, 1	0
WMBUS_RS485ENABLE	Activation of the RS-485 interface for the wM-Bus communication	0, 1	0
WMBUS_SMLENABLE	Activation of processing SML protocol data	0, 1	0
WMBUS_STOPBITS	Stop bits for the wM-Bus communication	1, 2	1
WMBUS_TRANSPARENT	Activation of the transparent transmission of the wM-Bus interface to a network port: NONE: transmission deactivated, TCP: transmission of a TCP port, UDP: transmission to a UDP port	NONE, TCP, UDP	NONE
WMBUS_TRANSPARENT PORT	Network port for the transparent transmission via TCP or UDP	0-65535	0
WMBUS_TRANSPARENT RSSI	Activation of the integration of the RSSI value in transparent mode	0, 1	0
WMBUS_TRANSPARENT STARTSTOP	Activation of the integration of a start byte and stop byte in transparent mode	0, 1	0
WMBUS_USE LINKLAYERID	Compatibility mode for reading of faulty wM-Bus meters, uses link layer address instead of extended link layer address	0, 1	0
WMBUS2_BAUDRATE	Baud rate for the wM-Bus communication (channel 2)	300, 600, 1200, 1800, 2400, 4800, 9600, 19200, 38400, 57600, 115200, 230400, 460800	19200
WMBUS2_DATABITS	Data bits for the wM-Bus communication (channel 2)	7, 8	8
WMBUS2_DEVPATH	Linux path of the wM-Bus interface (channel 2)	Text, max. 255 characters	Not set

Continued on next page

Table 24 – *Continued from previous page*

Parameter	Description	Range	Standard
WMBUS2_FLOW CONTROL	Handshake for the wM-Bus communication (channel 2): 0: none, 1: XON/XOFF when sending, 2: RTS/CTS, 8: XON/XOFF when receiving, 9: XON/XOFF when sending and receiving	0, 1, 2, 8, 9	0
WMBUS2_MODE	Mode of the wM-Bus module (channel 2)	S, T, C, C_T	C_T
WMBUS2_PARITY	Parity of the wM-Bus communication (channel 2): 0: none, 1: odd, 2: even, 3: mark, 4: space	0-4	0
WMBUS2_RS485ENABLE	Activation of the RS-485 interface for the wM-Bus communication (channel 2)	0, 1	0
WMBUS2_STOPBITS	Stop bits for the wM-Bus communication (channel 2)	1, 2	1
WMBUS2_ TRANSPARENT	Activation of the transparent transmission of the wM-Bus interface (channel 2) to a network port: NONE: transmission deactivated, TCP: transmission of a TCP port, UDP: transmission to a UDP port	NONE, TCP, UDP	NONE
WMBUS2_ TRANSPARENTPORT	Network port for the transparent transfer of the wM-Bus interface (channel 2) via TCP or UDP	0-65535	0
WMBUS2_ TRANSPARENTRSSI	Activation of the integration of the RSSI value in transparent mode of the wM-Bus interface (channel 2)	0, 1	0
WMBUS2_ TRANSPARENT STARTSTOP	Activation of the integration of a start byte and stop byte in transparent mode of the wM-Bus interface (channel 2)	0, 1	0
MODBUS_TLSENABLE			0
MODBUS_CA_FILE			0
MODBUS_CERT_FILE			0
MODBUS_KEY_FILE			0
MODBUS_INSECURE			0
MBUS_TRANSPARENT_ TLSENABLE			0
MBUS_TRANSPARENT_ CA_FILE			0
MBUS_TRANSPARENT_ CERT_FILE			0
MBUS_TRANSPARENT_ KEY_FILE			0
MBUS_TRANSPARENT_ INSECURE			0
WMBUS_ TRANSPARENT_ TLSENABLE			0
WMBUS_ TRANSPARENT_CA_FILE			0
WMBUS_ TRANSPARENT_CERT_ FILE			0
WMBUS_ TRANSPARENT_KEY_ FILE			0
WMBUS_ TRANSPARENT_ INSECURE			0
WMBUS2_ TRANSPARENT_ TLSENABLE			0

Continued on next page

Table 24 – *Continued from previous page*

Parameter	Description	Range	Standard
WMBUS2_ TRANSPARENT_CA_FILE			0
WMBUS2_ TRANSPARENT_CERT_ FILE			0
WMBUS2_ TRANSPARENT_KEY_ FILE			0
WMBUS2_ TRANSPARENT_ INSECURE			0
DLDERS_ TRANSPARENT_ TLSENABLE			0
DLDERS_ TRANSPARENT_CA_FILE			0
DLDERS_ TRANSPARENT_CERT_ FILE			0
DLDERS_ TRANSPARENT_KEY_ FILE			0
DLDERS_ TRANSPARENT_ INSECURE			0
MBUSSLVMETER_ TLSENABLE			0
MBUSSLVMETER_CA_ FILE			0
MBUSSLVMETER_CERT_ FILE			0
MBUSSLVMETER_KEY_ FILE			0
MBUSSLVMETER_ INSECURE			0
MBUSSLV2METER_ TLSENABLE			0
MBUSSLV2METER_CA_ FILE			0
MBUSSLV2METER_ CERT_FILE			0
MBUSSLV2METER_KEY_ FILE			0
MBUSSLV2METER_ INSECURE			0
Group [REPORT_x]*			
MODE	Mode of the report instance resp. deactivation		DISABLED
FORMAT	Format employed of the report instance		Not set
HOST	Remote station of the report instance		Not set
PORT	Network port of the remote station of the report instance		
PATH	Path for the remote station of the report instance		Not set
USER	Username for the remote station of the report instance		Not set
PASSWORD	Password for the remote station of the report instance		Not set
TOADDRESS	Receiver address of the report instance, particularly SMTP		Not set
FROMADDRESS	Sender address of the report instance, particularly SMTP		Not set
PARAM1	User-specific parameter (1) of the report instance, particularly user format or user mode		Not set
PARAM2	User-specific parameter (2) of the report instance, particularly user format or user mode		Not set

Continued on next page

Table 24 – *Continued from previous page*

Parameter	Description	Range	Standard
PARAM3	User-specific parameter (3) of the report instance, particularly user format or user mode		Not set
BASENAME	Basic file name for files to be transmitted (XML or CSV)		
CONTENTTYPE			
CONVERTARG			
EXTENSION			
INSECURE			0
CA_FILE	Path to the CA certificate for the report instance		
CERT_FILE	Path to the device certificate for the report instance		
KEY_FILE	Path to the device key for the report instance		
CYCLEMODE			MINUTE
CYCLE	Cycle time for meter reading (unit according to CYCLEMODE)		15
CYCLEDELAY			0
CYCLETIMESTAMP			Not set
RANDOMDELAY			
READOUT_FILTER	Selection if all values, or only the newest, or only the oldest value from a particular time span should to be transmitted in a cyclic report	ALL, NEWEST, OLDEST	ALL
RETRY_INTERVAL	Interval for the retry of failed reports: -1: no repetition, failed reports are not retransmitted, 0: automatic (for cyclic reports retry after 1/10 of the Report Cycle Time with minimum 10 minutes, for reports with „On Readout“ retry after 10 minutes), >0: time in seconds after which a failed report is retransmitted	-1, 0, arbitrary positive integer	0
MIN_SEND_INTERVAL	Minimum interval for sending the report. Assures that at least this delay (in seconds) is respected after the successful transmission of a report or the failure to send a report before transmitting the subsequent report. The parameter is not effective if reports are prompted by Readout or manually via the website.	0, arbitrary positive integer	0
MAX_BACKLOG	Maximum time into the past for which reports are sent (in seconds). See complement underneath this table.	arbitrary positive integer	0

*x denotes the report instance 1-10

Table 24: chip.ini parameters

- Complement to MAX_BACKLOG:
  - For cyclic reports, only reports are transmitted whose data range is not entirely older than this period. If the beginning of the data range is older and the end newer than this time for a report, then the report will be transmitted with its entire data range.
  - For a report triggered with „On Readout“, the beginning of the data range is limited to the Backlog time.
  - The analysis occurs upon system start, reconfiguration or the generation of a report by due date, retry after failure or readout. If reports fail continually, no retry of reports older than the indicated time will occur.

10.4 Configuration file Device_Handle.cfg

The file *app/Device_Handle.cfg* contains the meter configuration. If this file does not exist, it can be created via the web-based front end using the ***Meter*** tab. All wM-Bus meters collected during operation

are integrated permanently into the list after a scan process or by manually saving the configuration. Only those parameters need to be stored in that file which deviate from the defined default values (version entry excluded).

- ⚠ The file has to be saved as UTF8 encoded XML file.
- ⚠ To devices with older software without a database (prior to 1.34) applies the following: When the meter configuration is changed, all files in the directory *ext/Tmp* have to be deleted manually (if present). Data which has not been reported is discarded when the meter configuration is changed.
- ⚠ To devices with newer software with a database (from 1.34) applies the following: When the file *Device_Handle.cfg* is changed manually, the parameter <*layoutversion*> stated therein has to be incremented.
- ℹ The device needs to be rebooted after changing the file *Device_Handle.cfg* manually for the change to take effect. The reboot can be triggered via the web-based front end using the button **Reboot system** in the ***Service*** tab or via the command line.
- ℹ Manual changes are permanently stored on the flash not instantly, but after a few minutes. As a result, changes are possibly lost if the device is rebooted by switching the power supply off and on.
- ✔ The file *Device_Handle.cfg* can be transferred to other devices via FTPS. The attached meters need to be taken into account.

The file is an XML file and has the following structure:

Parent	Element	Description	Standard	Example
	root	Root element	-	-
root	version	Version number of the XML specification	Not set	0x06
root	layoutversion	Layout number of the database	Not set	0x06
root	meter	Parent element for each meter	-	-
meter	interface	Interface of the meter: M-Bus, wM-Bus, DLDERS, S0, Modbus	Not set	M-Bus
meter	serial	Meter number (serial number), BCD notation, leading „0x“	0xFFFFFFFF	0x30101198
meter	manufacturer	Manufacturer code of the meter (wildcard 0xFFFF)	0xFFFF	0x3B52 (NZR)
meter	version	Version number of the meter	0xFF	0x01
meter	medium	Medium of the meter, see second column in Table 26 (wildcard 0xFF, if not set)	Not set	Electricity
meter	primaryaddress	Primary address of the meter (M-Bus, S0 or Modbus)	0	0x03
meter	addressmode	Addressing mode 0: secondary, 1: primary	0	0
meter	readoutcycle	Specific readout cycle (in s)	0	900
meter	maxvaluecount	Limitation of the number of meter values	0	12
meter	encryptionkey	Key for encrypted communication, e.g.: AES for wM-Bus	Not set, 0	0x82 0xB0 0x55 0x11 0x91 0xF5 0x1D 0x66 0xEF 0xCD 0xAB 0x89 0x67 0x45 0x23 0x01
meter	active	Activates the meter for logging or for reporting.	1	1
meter	rssi	RSSI value of the last reception (wM-Bus)	0	123
meter	register	Register assignment (e. g. Modbus slave)	0	250
meter	user	User-specific text (see **User label** column in the ***Meter*** tab)	Not set	Floor-1-Right
meter	dbid	Unique database key of the meter, if the meter is activated for reporting	Not set	1
meter	value	Parent element for each meter value of the meter	-	-
value	description	Description of the meter value, see second column in Table 27	None	Energy
value	unit	Unit of the meter value, see second column in Table 28	None	Wh
value	encodetype	Coding of the meter value	NODATA	INT32

Continued on next page

Table 25 – *Continued from previous page*

Parent	Element	Description	Standard	Example
value	scale	Scaling factor of the meter value (scientific notation)	1e0	1e-3
value	userscale	User-specific scaling factor of the meter value (scientific notation)	1e0	1e-1
value	valuetype	Type of meter values: INSTANTANEOUS, MAXIMUM, MINIMUM, ERRORSTATE	instantaneous	instantaneous
value	storagenum	Storage number of the meter value	0	2
value	tariff	Tariff information of the meter value	0	3
value	confdata	Generic data, OBIS code of the meter value (X-X:X.X.X*X; X=0-255; see **OBIS-ID** column in the ***Meter*** tab)	Not set	0x01 0x00 0x01 0x08 0x00 0xFF
value	rawdata	Raw data of the meter value for M-Bus and wM-Bus	Not set	07 FB 0D 00 00 00 00 00 00 00
value	dif	Data information fields of the meter value for M-Bus and wM-Bus	Not set	07
value	vif	Value information fields of the meter value for M-Bus and wM-Bus	Not set	FB 0D
value	active	Activates the meter value for logging or for reporting.	1	1
value	register	Register assignment (e. g. Modbus slave)	0	250
value	user	User-specific text (see **User label** column in the ***Meter*** tab)	Not set	Room 2
value	bacnetreg	Object number for BACnet	Not set	8

Table 25: Structure of the Device_Handle.cfg

10.5 OpenVPN Client

An OpenVPN client is integrated on the devices from solvimus GmbH for enabling an encrypted remote access. This offers a comfortable way to configure and operate the devices remotely. The configuration of the devices themselves is very simple and intuitive.

The use of a VPN is restricted or even prohibited by law in some countries. Every user is obliged to inform himself about the laws applicable in his country.

10.5.1 Configuration of the device

Using the OpenVPN client is simple. Only the configuration file *config.ovpn* for the client has to be stored on the device in the directory *app/vpn*. This directory can be created when connecting via FTP. The configuration file can be obtained from the administrator of your VPN. The device needs to be restarted by pressing the button **Reboot system** in the ***Service*** tab or via the command line. The OpenVPN client is activated by using the checkbox **VPN** in the ***General*** tab (see Section 4.3).

Please be aware of the exact file name: *config.ovpn*.

When saving the configuration via the web-based front end, the OpenVPN client is started and the VPN connection is established.

OpenVPN usually uses the UDP port 1194. A firewall needs to allow this port.

Please ask your administrator for providing a client configuration file.

10.6 Preconfiguration of the meter list

Manual editing of a meter list for large installations with many meters is demanding and time-consuming.

This can be automated with two approaches.

10.6.1 File meter-conf-import.csv

The first approach uses the *app/meter-conf-import.csv*. It is used to add meta information such as the **Encryption key** or the **User label** when scanning/listing a meter.

- ✔ If the meter is already listed or configured in the ***Meter*** tab, the data from the file will not be transferred. The meter has to be removed from the list first.

The file can be manually uploaded to the device via FTPS (see also Section 3.5). However, it is also possible to import it via the ***Service*** tab (see Section 4.12). The file has to be provided as packed **.tar.gz* file.

- For creating a **.tar.gz* archive, the free, open source software *7zip* can be used. First, the file *meter-conf-import.csv* needs to be packed without subdirectory into a **.tar* ball and afterwards into a **.gz* archive.

The following columns can be used in the CSV file:

- Interface: the interface via which the meter is read out (M-Bus, wM-Bus).
- Serial: 8-digit meter serial number
- Encryption key: Encryption key of the meter in hexadecimal byte notation (optional)
- User label: User-specific label of the meter (optional)
- Cycle: Readout interval of the meter (in seconds, optional)
- Max readout values: Limit to the quantity of meter values if the meter provides additional meter values (optional). If not set, the parameter „Maximum value count“ from the tab ***Configuration*** is used.

Here is an example:

```
Interface; Serial; Encryptionkey; user label; cycle; Max readout values
WMBUS;12345670;00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F;Apartment 01;;
WMBUS;12345671;01 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F;Apartment 02;;
WMBUS;12345672;02 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F;Apartment 03;;
WMBUS;12345673;03 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F;Apartment 04;;
WMBUS;12345674;04 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F;Apartment 05;;
WMBUS;12345675;05 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F;Apartment 06;;
WMBUS;12345676;06 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F;Apartment 07;;
WMBUS;12345677;07 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F;Apartment 08;;
WMBUS;12345678;08 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F;Apartment 09;;
WMBUS;12345679;09 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F;Apartment 10;;
```

10.6.2 File Device_Config.cfg

The second approach uses the file *app/Device_Config.cfg*.

10.7 Scripting

Extending the functional scope of the standard device by customer-specific functionalities is the main purpose for scripting. Its basis are source codes which are executed or interpreted on the target system, i.e. the device.

Standard environments such as *XSLTPROC* or *BASH* are available as interpreters on the devices from solvimus GmbH, wherein the transformation of the meter data to the destination format is performed by the transformation language XSL. Scripts can run in these environments and enable various functions.

10.7.1 XSLT parser

XSLTPROC is an interpreter for applying XSLT stylesheets to XML documents.

- More information can be found at: http://xmlsoft.org/XSLT/xsltproc.html

Extensible Stylesheet Language Transformation (XSLT) is a description language for transforming an XML document into another document. This can be an XML document, a text document (e. g. CSV file or JSON file) or even a binary file.

Source and target files are considered as logical trees in XSLT. The transformation rule describes which nodes of the tree are processed and how the new content is derived from them. Conditional statements and loops can also be used.

The main intention for offering XSLT on the devices from solvimus GmbH is the generation of user-specific data formats. The device internally uses a proprietary XML format to provide the meter data. In order to generate the format that the operator uses or prefers, an XSLT conversion rule is used. In this way, the standard formats are generated (see Section 4.8) and additional user-specific formats can be provided.

- ✔ Only one single user-specific format is available for the standard operating modes (e. g. TCP or FTP) of the report instances. If several different user-specific formats are required, other instances must be to set to *User* mode.

Here are some possible applications:

- CSV file per meter
- JSON data stream for IoT communication
- Time displayed as readable ASCII string instead of UNIX timestamp
- Fixed point notation in CSV file
- Changed column arrangement in CSV file
- Combine several meter values of identical type in one line if read out at the same time

The transformation files can be used either within the scripts for the transformation of the format or via the configuration website in the ***Meter*** tab (button **Export**, format: USER) for an export. These can be stored in the following paths. The **.xsl* files are stored in *app/report*. The file name is specific to the instance and composed of *report_* and the number of the instance (n = 1-10). Thus, an individual user-specific format can be realized for each report instance: *report_1.xsl*, *report_2.xsl*, ... For a ***Report format*** *User* (see Section 8.4.4) selected via the front end, the respective file *app/report/report_n.xsl* will be used for each instance (n = 1-10). If the file specific to the instance is not available, the path instance *app/report/report.xsl* will be used which is also employed for the export of the meter value data. The path check occurs when initializing the application.

10.7.2 Report script

In addition to the operator, the application can also issue commands via the command line (see Section 10.1.2). This allows implementing user-specific processes on the devices from solvimus GmbH.

If the mode of a report instance is set to *User*, this function comes into play. Instead of the hard-coded processes like TCP or FTP, the provided *BASH* script is now called. The command sequence contained therein is processed and then the script is terminated. In this way, third-party tools available for Linux can also be used for transferring data or for implementing orthogonal functionality. Here are some possible applications:

- MQTT for IoT communication
- Connectivity to an InfluxDB
- Request to server before sending data (conditional data transfer)
- Reporting to different file servers, depending on the **User label** set
- Checking thresholds and alarming

The script files are stored as **.sh* in *app/report*. The file name is specific to the instance and composed of *report_* and the number of the instance (n = 1-10). Thus, an individual user-specific script can be realized for each report instance: *report_1.sh*, *report_2.sh*, ... For a ***Report mode*** *User* (see Section 8.4.4) selected via the front end, the respective file *app/report/report_n.sh* will be used for each instance (n = 1-10). If the file specific to the instance is not available, the path instance *app/report/report.sh* will be used. The path check occurs when initializing the application.

The following example sends user-specific data via MQTT. Therefore, *XSLTPROC* is called before the MQTT call is made via *mosquitto_pub* (long lines are wrapped):

```
#!/bin/bash
exec 1> >(logger -t report) 2>&1
set -e
set -o pipefail

shopt -s nullglob
rm -rf /tmp/reportfiles || true
mkdir /tmp/reportfiles
mcsvtoxml -m -c | xsltproc --stringparam serial "$SOLAPP_SERIAL"
  --stringparam timestamp "$(date +%s)" /mnt/app/report/report.xsl -

for file in /tmp/reportfiles/*/*; do
  subpath=$(echo ${file#/tmp/reportfiles/} | cut -d "." -f 1)
  mosquitto_pub -u "$SOLAPP_REPORT_USER" -P "$SOLAPP_REPORT_PASSWORD"
    -h "$SOLAPP_REPORT_HOST" -p "$SOLAPP_REPORT_PORT"
    --cafile "/var/conf/app/cacert.pem" --cert "/var/conf/app/clicert.pem"
    --key "/var/conf/app/clikey.pem" -t "$SOLAPP_REPORT_PATH/$subpath"
    -f "$file" --id "$HOSTNAME" --insecure
done
```

10.7.3 System meter script

Like the report using report scripts (see Section 10.7.2), the system meter (see Section 4.4.1) can also be extended user-specifically with system meter scripts.

Here, a *BASH* script is called at the readout time. It could return a meter value after completion. The return value needs to contain the following values in this order, separated by *newline* characters:

- Description of the meter value, *Description* column
- Unit of the meter value, *Unit* column
- Value of the meter value *Value* column

Here are some possible applications:

- Measure ping times for network quality monitoring
- Display outdoor temperature via Web API access

The script files are stored as **.sh* file in *app/metersystem*. The respective file name is composed of *value* and a consecutive number from *1* upwards. Thus, user-specific values can be realized: *value1.sh*, *value2.sh*, ...

The following example adds the ping time to example.com to the system meter:

```
#!/bin/bash
echo -ne "Ping\nms\n"
ping=$(ping -n -c 3 example.com 2> /dev/null)
if [ $? -eq 0 ]; then
  echo $ping | awk -F '/' 'END {print $4}'
else
  echo -1
fi
```

10.8 Media types, measurement types and units

In the EN 13757-3 standard, media types, measurement types (measurement value descriptions) and units and are predefined. The devices from solvimus GmbH are using it for allowing a uniform data display.

The following table contains the predefined values for the medium:

Index	Description
0	Other
1	Oil
2	Electricity
3	Gas
4	Heat (outlet)
5	Steam
6	Warm water
7	Water
8	Heat cost allocator
9	Compressed air
10	Cooling (outlet)
11	Cooling (inlet)
12	Heat (inlet)
13	Combined heat / cooling
14	Bus / System component
15	Unknown medium
16-19	Reserved
20	Calorific value
21	Hot water
22	Cold water
23	Dual register (hot/cold) water
24	Pressure
25	A/D Converter
26	Smoke detector
27	Room sensor
28	Gas detector
29-31	Reserved
32	Breaker (electricity)
33	Valve (gas or water)
34-36	Reserved
37	Customer unit
38-39	Reserved
40	Waste water
41	Waste
42	Carbon dioxide
43-48	Reserved
49	Communication controller
50	Unidirectional repeater
51	Bidirectional repeater
52-53	Reserved
54	Radio converter (system side)
55	Radio converter (meter side)
56-255	Reserved

Table 26: Media types

The following table contains the predefined measurement types (descriptions for the measured value). Depending on the meter's interface, user-specific text-based measurement types (indication by index 31) can also be configured.

Index	Description
0	None
1	Error flags (Device type specific)
2	Digital output
3	Special supplier information
4	Credit
5	Debit
6	Volts
7	Ampere
8	Reserved
9	Energy
10	Volume
11	Mass
12	Operating time
13	On time
14	Power
15	Volume flow
16	Volume flow ext
17	Mass flow
18	Return temperature
19	Flow temperature

Continued on next page

Table 27 – *Continued from previous page*

Index	Description
20	Temperature difference
21	External temperature
22	Pressure
23	Timestamp
24	Time
25	Units for H. C. A.
26	Averaging duration
27	Actuality duration
28	Identification
29	Fabrication
30	Address
31	Meter specific description (text based)
32	Digital input
33	Software version
34	Access number
35	Device type
36	Manufacturer
37	Parameter set identification
38	Model / Version
39	Hardware version
40	Metrology (firmware) version
41	Customer location
42	Customer
43	Access code user
44	Access code operator
45	Access code system operator
46	Access code developer
47	Password
48	Error mask
49	Baud rate
50	Response delay time
51	Retry
52	Remote control (device specific)
53	First storagenum. for cyclic storage
54	Last storagenum. for cyclic storage
55	Size of storage block
56	Storage interval
57	Vendor specific data
58	Time point
59	Duration since last readout
60	Start of tariff
61	Duration of tariff
62	Period of tariff
63	No VIF
64	wM-Bus data container
65	Data transmit interval
66	Reset counter
67	Cumulation counter
68	Control signal
69	Day of week
70	Week number
71	Time point of day change
72	State of parameter activation
73	Duration since last cumulation
74	Operating time battery
75	Battery change
76	RSSI
77	Day light saving
78	Listening window management
79	Remaining battery life time
80	Stop counter
81	Vendor specific data container
82	Reactive energy
83	Reactive power
84	Relative humidity
85	Phase voltage to voltage
86	Phase voltage to current
87	Frequency
88	Cold/Warm Temperature limit
89	Cumulative count max. power
90	Remaining readout requests

Continued on next page

Table 27 – *Continued from previous page*

Index	Description
91	Meter status byte
92	Apparent energy
93	Apparent power
94	Security key
95	Data frame
96-255	Reserved

Table 27: Measurement types

The following table contains the predefined units. Depending on the meter's interface, user-specific units can also be configured.

Index	Unit	Symbol	Description
0	None		None
1	Bin		Binary
2	Cur		Local currency units
3	V	V	Volt
4	A	A	Ampere
5	Wh	Wh	Watt hour
6	J	J	Joule
7	m^3	m^3	Cubic meter
8	kg	kg	Kilogram
9	s	s	Second
10	min	min	Minute
11	h	h	Hour
12	d	d	Day
13	W	W	Watt
14	J/h	J/h	Joule per Hour
15	m^3/h	m^3/h	Cubic meter per hour
16	m^3/min	m^3/min	Cubic meter per minute
17	m^3/s	m^3/s	Cubic meter per second
18	kg/h	kg/h	Kilogram per hour
19	Degree C	°C	Degree Celsius
20	K	K	Kelvin
21	Bar	Bar	Bar
22			Dimensionless
23-24			Reserved
25	UTC		UTC
26	bd	bd	Baud
27	bt	bt	Bit time
28	mon	mon	Month
29	y	y	Year
30			Day of week
31	dBm	dBm	Decibel (1 mW)
32	Bin		Bin
33	Bin		Bin
34	kVARh	kVARh	Kilo voltampere reactive hour
35	kVAR	kVAR	Kilo voltampere reactive
36	cal	cal	Calorie
37	%	%	Percent
38	ft^3	ft^3	Cubic feet
39	Degree	°	Degree
40	Hz	Hz	Hertz
41	kBTU	kBTU	Kilo british thermal unit
42	mBTU/s	mBTU/s	Milli british thermal unit per second
43	US gal	US gal	US gallon
44	US gal/s	US gal/s	US gallon per second
45	US gal/min	US gal/min	US gallon per minute
46	US gal/h	US gal/h	US gallon per hour
47	Degree F	°F	Degree Fahrenheit
48-255			Reserved

Table 28: Units

11 Transmission of read out meter data via Modbus TCP

11.1 General information

The Modbus protocol was originally developed by the company Modicon (now Schneider Electric) for easy data exchange with their controllers. Data were transmitted as 16-bit registers (integer format) or as state information in the form of data bits. Over the course of time, the protocol has been continually extended. Modbus TCP is one variant.

- Modbus TCP is part of the standard IEC 61158
- A specification can be found at: http://www.modbus.org

The Modbus protocol is a single master protocol. This master controls the entire transfer and monitors potential timeouts (no response from the addressed device). The connected devices may only send telegrams upon request by the master.

The devices from solvimus GmbH are, if option available, a Modbus TCP server and thus a Modbus TCP slave.

The Modbus communication requires an active TCP connection between a client (e. g.: PC or controller) and the server (this device). The TCP port configured in the ***Server*** tab is used for the Modbus communication. This is configured to 502 by default (see Section 4.8).

- ✔ If there is a firewall between server and client, ensure that the configured TCP port is enabled.

The devices from solvimus GmbH allow multiple simultaneous Modbus TCP connections in the standard configuration. This means, for example, that in addition to a classic PLC, you can also connect a BMS and a Modbus-capable display in parallel. The queries of these Modbus clients are not influencing each other. The configuration parameter *MODBUS_MAXCONNECTIONS* (*app/chip.ini*, see Section 10.3) determines the maximum number of simultaneous Modbus queries. If this limit is exceeded, the oldest existing Modbus TCP connection is disconnected by the device. So, the newly requested connection is now allowed.

- ✔ The device supports up to five simultaneous Modbus TCP connections in the standard configuration.
- ✔ The device supports Modbus TCP as well as the uncommon Modbus UDP. The mode is selected by **Modbus mode** in the ***Server*** tab. Besides connectivity aspects, the behaviour in both modes is almost the same.

11.2 Function codes and addressing

The following function codes are supported by the devices from solvimus GmbH:

Code	Name	Description
0x01	Read Coil	Currently without function
0x03	Read Holding Register	Request of meter data, register layout according to tables in Section 11.3
0x05	Write Single Coil	Currently without function
0x06	Write Single Register	Currently without function
0x10	Write Multiple Register	Currently without function
0x0F	Force Multiple Coil	Currently without function
0x2B	Read Device Identification	Request of device information using *MEI = 0x0E*

Table 29: Function codes for Modbus TCP or Modbus UDP

The function codes marked „without function“ are responded with *ILLEGAL DATA ADDRESS (0x02)*. All other not listed codes are responded with the error message *ILLEGAL FUNCTION (0x01)*.

If the function code *0x2B* with *MEI = 0x0E* is used, the device returns an identification telegram. The values *0x01* and *0x02* are supported as *Read Device ID code*. This allows requesting the basic data set *(basic device identification)* and the standard data set *(regular device identification)*. The following data can be requested via the device identification:

Object ID	Name	Data type	Example	Type
0x00	VendorName	String	[Branding]	Basic
0x01	ProductCode	String	1036	Basic
0x02	MajorMinorRevision	String	001	Basic
0x03	VendorUrl	String	[Branding]	Regular
0x04	ProductName	String	MBUS-GE80M*	Regular
0x05	ModelName	String	Standard	Regular
0x06	UserApplicationName	String	Modbus TCP Gateway	Regular

*Corresponds to the configured *Device name* in the ***General*** tab.

Table 30: Device identification

Modbus allows addressing of different stations on the bus via a slave address. Primarily, Modbus TCP uses directly the IP address of the device for addressing. Therefore, the slave address remains usually unused. It is recommended to use *0xFF (255)* for Modbus TCP.

- ✔ The devices from solvimus GmbH are not checking the slave address in the standard configuration, but are always responding if the IP address matches.
- ✔ The standard implementation of the Modbus server is not separating the connected meters and their data logically. The data can be requested across several meters with only one query.

11.3 Data representation

The solvimus GmbH uses the common data arrangement in the Modbus registers. Addressing starts with *0*, and the *big endian* layout is used. Therefore, in the 16-Bit registers the higher byte is sent first, the lower byte then afterwards (this is also called *most significant byte first* or *MSB*).

Example: value 0x1234 → transmitting: 0x12 first, 0x34 then

Numbers and data ranges that exceed 16 Bit are represented alike. Again, the most significant 16-Bit register is sent first, so it is at the lowest register address (also referred to as *most significant word first* or *MSW*).

Example: value 0x12345678 → transmitting: 0x12 first, 0x34, 0x56 and 0x78 then

The devices use 10 Modbus registers for each entry in the meter list. This includes meta information such as readout time, unit and readout status. This results in the following Modbus register specification with a fixed grid of 10 Modbus registers.

- ℹ The register addresses are counted starting from the value 0.
- ℹ Data types that span more than one register are encoded with the more significant word at the lower address.
- ℹ The Modbus registers are read out via the function code *0x03 (Read holding register)* (see Section 11.2).
- ✔ In the Modbus protocol, the data is transmitted as integers or floating values. Other data formats specified for the M-Bus (e. g.: BCD) are already converted internally into integer values before transmission.

The 10 Modbus registers starting at address 0 are status registers of the device itself and are defined according to the following table:

Address	Description	Data width	Comments
0-1	Serial number	32 Bit	The serial number is encoded in hexadecimal.
2	Protocol version	16 Bit	Protocol version of the Modbus data (value = 1)
3	Version	16 Bit	Software version of the device (integer value)
4-5	Timestamp	32 Bit	Current system time of the device as UNIX time (UTC). Therefore, the system time of the device has to be correctly set (manually or via SNTP).
6	Reserved		Reserved
7	Type field/Reserved	16 Bit	The type field (value = 1 for device entry) is transmitted in the most significant byte. The least significant byte is reserved.
8-9	Reserved		Reserved

Table 31: Modbus registers representing the data set of the device.

These first 10 Modbus registers are now followed by entries for meters and entries for meter values according to the hierarchy in the meter list. An entry for meters is followed by associated entries for meter values, before

a new entry for the next meter follows, and so on.

The 10 Modbus registers of a meter entry are defined according to the following table. The offset has to be added to the configured Modbus address (**Register**) in the ***Meter*** tab.

Offset	Description	Data width	Comments
0-1	Serial number	32 Bit	The serial number is encoded in hexadecimal. Unlike M-Bus or wM-Bus, this is an integer and not a BCD.
2	Manufacturer code	16 Bit	The three ASCII characters of the manufacturer code are encoded via individual bit areas: Bits 10-14: first character, Bits 5-9: second character and Bits 0-4: third character. The particular character results from the respective value (significant bit at the highest position) by counting up, starting with the letter „A“ at a value of 1.
3	Version/Medium	16 Bit	The version of the meter is encoded in the most significant byte and the medium ID in the least significant byte of the register. The medium is assigned using Table 26. The transferred value corresponds to the index.
4-5	Timestamp	32 Bit	System time of the device at the time of last readout as UNIX time (UTC). Therefore, the system time of the device has to be correctly set (manually or via SNTP).
6	Reserved		Reserved
7	Type field/Reserved	16 Bit	The type field (value = 2 for meter entry) is transmitted in the most significant byte. The least significant byte is reserved.
8	Flags	16 Bit	Bit 0: value 1: meter not read, value 0: meter correctly read Bit 1: value 1: not all meter values up to date, value 0: all meter values up to date Bit 2-15: Reserved
9	Reserved		Reserved

Table 32: Modbus registers representing the data set of a meter

The 10 Modbus registers of a meter value entry are defined according to the following table. The offset has to be added to the configured Modbus address (**Register**) in the ***Meter*** tab:

Offset	Description	Data width	Comments
0-3	Meter value	64 Bit	Signed integer meter value (unscaled). Only available if the meter value is not transmitted by the meter as Float32/Double64 floating point value. This is given by **Edit value**, **Encode type** (see Figure 36). To assure the transmission of unmodified meter values, a back-calculation to the integer (modified value and scaling factor) is not intended.
4-5	Meter value	32 Bit	Floating point meter value (scaled according to the unit in register 7), IEEE 754
6	Scaling factor	16 Bit	Signed scaling factor to base 10.
7	Type field/Unit	16 Bit	The type field (value = 0 for meter value entry) is transmitted in the most significant byte. The unit of the value is transmitted in the least significant byte. The unit is assigned using Table 28. The transferred value corresponds to the index.
8-9	Timestamp	32 Bit	Time which is provided for this meter value by the meter itself. It is transmitted as UNIX time (UTC). If the meter does not provide a time, this timestamp is 0.

Table 33: Modbus registers representing the data set of a meter value

- Under certain conditions the registers with offset 0-3 do not contain meter values, but 0. This is the case if the meter transmits data as FLOAT32 e.g. via M-Bus. Neither the „next“ integer nor a scaling is computed. This can be discerned by the presence or absence of a comma in the column *Value* of the respective meter value on the website. A comma indicates, as a rule, a FLOAT32 value and hence not an integer and merely the registers with offset 4 and 5 contain meter values.
- Floating point formats have a limited resolution. This may result in slight deviations between the represented value and the exact value.
  - Example: 0x449a522b = 1234.5677490234375 instead of 1234.5678
- For string values (e. g. customer name) via M-Bus, everything equals 0.
- The scaling factor contains only the exponent. For S0-meters with certain pulse ratios (mantissa not equal to 1), the complete conversion factor is thus not given.
  - Example: scaling 0.01 m^3/pulse → Scale = 1e-2 → Modbus register = -2 = 0xFFFE
  - Example: scaling 0,005 m^3/pulse → Scale = 5e-3 → Modbus register = -3 = 0xFFFD
- Herein, „Scale“ refers to the column of the same name on the website in the tab ***Meter*** (see Section 4.4) resp. to the entry of the same name in the dialogue **Add value** for the creation of a meter value (see the section depending on the interface).

The following figure shows an example configuration of the Modbus addresses on the web-based front end:

MBus	66600106	LUG	Heat (outlet)	2					10
—					4	1e+0	s	Actuality Duration	0
—					4	1e+0	s	Averaging Duration	0
—					267	1e+3	Wh	Energy	20
—					372876	1e-2	m^3	Volume	0
—					0	1e+2	W	Power	0

Figure 50: Configured Modbus registers on the web-based front end

The following data is transmitted to the Modbus master in this example:

Address	Value	Description	Decoded value
Device entry			
0 1	0x0002 0x993A	Serial number	0x0002993A
2	0x0001	Protocol version	1
3	0x006F	Version	Version = 0x006F = 111 → v1.11
4 5	0x519C 0xC16D	Timestamp	0x519CC16D = 1369227629: Wednesday, 22 May 2013, 15:00:29 GMT+2
6	0x0000	Reserved	
7	0x0100	Type field/Reserved	Type= 1 → device entry
8	0x0000	Reserved	
9	0x0000	Reserved	
Meter entry			
10 11	0x03F8 0x3CAA	Serial number	0x03F83CAA = 66600106
12	0x32A7	Manufacturer code	0x32A7 = '0011.0010.1010.0111' 1st character: '_011.00__._____._____' → 0x0C = 12 → L 2nd character: '_____.__10.101_._____' → 0x15 = 21 → U 3rd character: '_____._____.____0.0111' → 0x07 = 7 → G
13	0x0204	Version/Medium	Version = 2 Medium = 4 = Heat (outlet)
14 15	0x519C 0xC16D	Timestamp	0x519CC16D = 1369227629: Wednesday, 22 May 2013, 15:00:29 GMT+2
16	0x0000	Reserved	
17	0x0200	Type field/Reserved	Type = 2 → Meter entry
18	0x0000	Reserved	
19	0x0000	Reserved	
Meter value entry			
20 21 22 23	0x0000 0x0000 0x0000 0x010B	Meter value (integer)	0x000000000000010B = 267 Resulting meter value: 267 * 10^3 Wh
24 25	0x4882 0x5F00	Meter value (floating point)	0x48825F00 = 267000.000000 Wh
26	0x0003	Scaling factor	Factor = 10^3
27	0x0005	Type field/Unit	Type = 0 → meter value entry Unit = 5 → Wh
28 29	0x519C 0xBBB3	Timestamp	0x519CBBB3 = 1369226163: Wednesday, 22 May 2013, 14:36:03 GMT+2

Table 34: Example data for Modbus

11.4 Configuration via the web-based front end

The Modbus slave is activated and configured via the ***Server*** tab. The parameters are described in the Section 4.8. The settings are explained in detail below.

11.4.1 Modbus mode and Modbus port

The Modbus slave can be activated using the parameter *Modbus mode*. It can be set to *Modbus TCP* or *Modbus UDP*.

Modbus TCP is the most popular and common Modbus variant on IP networks. It uses TCP for communication. Using UDP for *Modbus UDP* is uncommon, but it is available as an option.

Both IP-based protocols are using the port specified in the parameter *Modbus port*. This is 502 by default.

If the parameter *Modbus port* is set to a value that is used by other services (e. g.: HTTP: Port 80), these services may block each other and access to the device is restricted.

11.4.2 Modbus test

Depending on the Modbus implementation, data representation and addressing may differ between the Modbus nodes. For checking the correct settings, the parameter *Modbus test* in the ***Server*** tab is enabling static test data in the Modbus slave (see Section 4.8). The following data is then provided via Modbus according to the register map in Section 11.3:

Address	Value	Description	Decoded value
0	0xD080	Serial number of the device, upper word	0xD0800DC1: last digits of the MAC address: 68:91:D0:80:0D:C1
1	0x0DC1	Serial number of the device, lower word	
2	0x0002	Version of the communication protocol of the device	2
3	0x0084	Software version of the device	0x84 = 132: Version 1.32
4	0x5CE5	System time of the device (timestamp), upper word	0x5CE55EAC = 1559054252: Wednesday, 22 May 2019, 16:37:32 GMT+2
5	0x5EAC	System time of the device (timestamp), lower word	
6	0x0000	Blank register	
7	0x0100	Type field of the data set in the upper byte	0x01: type is device entry
8	0x0000	Blank register	
9	0x0000	Blank register	
10	0x00BC	Serial number of the meter, upper word	0xBC614E = 12345678
11	0x614E	Serial number of the meter, lower word	
12	0x0443	Manufacturer code of the meter (see Section 11.3)	0x0443: ABC
13	0x0102	Version (upper byte) and medium (lower byte) of the meter	0x01: version = 1, 0x02: medium = 2 (electricity)
14	0x5CE5	Readout time of the meter (timestamp), upper word	0x5CE55EAC = 1559054252: Wednesday, 22 May 2019, 16:37:32 GMT+2
15	0x5EAC	Readout time of the meter (timestamp), lower word	
16	0x0000	Blank register	
17	0x0200	Type field of the data set in the upper byte	0x02: type is meter entry
18	0x0000	Flags in the lower byte	0x00: Meter correctly read and all values up to date
19	0x0000	Blank register	
20	0x0000	Meter value (integer), highest word	0xBC614E = 12345678: Resulting meter value: 12345678 * 10^{-4} = 1234.5678 Wh
21	0x0000	Meter value (integer)	
22	0x00BC	Meter value (integer)	
23	0x614E	Meter value (integer), lowest word	
24	0x449A	Meter value (floating point), upper word	0x449A522B = 1234.5677490234375 (rounding error using *FLOAT32*)
25	0x522B	Meter value (floating point), lower word	
26	0xFFFC	Scaling factor (power of 10)	0xFFFC = -4: factor = 10^{-4}
27	0x0005	Type field of the data set in the upper byte and unit in the lower byte (see Table 28)	0x00: type is meter value entry 0x05: unit = Wh
28	0x5CE5	Provided time of the meter value (timestamp), upper word	0x5CE55EAC = 1559054252: Wednesday, 22 May 2019, 16:37:32 GMT+2
29	0x5EAC	Provided time of the meter value (timestamp), lower word	

Table 35: Test data for Modbus TCP resp. Modbus UDP

The above values should be reproduced exactly(!) at the Modbus master. If not, the addressing and/or byte order probably do not match.

11.4.3 Modbus swap

Modbus uses the *big endian* data representation for bytes and words (individual registers). Addressing is starting at *0*. Depending on the manufacturer and implementation, the addressing and the data representation for data types larger than 16 Bit may differ between Modbus nodes.

There are two types of addressing, starting from *0* or from *1*. While this can be adjusted easily by an adding an offset, adjusting the word order is somewhat more complex.

Among others, the meter values are transmitted as floating point values (*FLOAT32*). The *FLOAT32* value is represented by 32 Bit and thus 4 Byte. These 4 Byte are stored in two Modbus registers. Each of the bytes

follows the *big endian* notation, but the byte order is not always consistent. Possible arrangements are shown as example.

For the example, the meter value out of the test data is used (12345678 * 10^{-4} = 1234.5678 Wh, see Table 35). This value is represented by the *FLOAT32* value 0x449A522B.

	Order of							
Mode	**Bits in byte**	**Bytes in word**	**Words**	**Byte 1**	**Byte 2**	**Byte 3**	**Byte 4**	**Short form**
Standard	big endian	big endian	MSW	0x44	0x9A	0x52	0x2B	ABCD
	big endian	little endian	MSW	0x9A	0x44	0x2B	0x52	BADC
Modbus swap	big endian	big endian	LSW	0x52	0x2B	0x44	0x9A	CDAB
	big endian	little endian	LSW	0x2B	0x52	0x9A	0x44	DCBA

Table 36: Modbus data alignment for the example data

According to the Modbus standard, the devices from solvimus GmbH are always representing the bits and bytes in the register in the *big endian* format. The registers themselves are represented either in the format of *most significant word first (MSW)* if *Modbus swap* is not active (default mode) or alternatively as *least significant word first (LSW)* if *Modbus swap* is active.

11.4.4 Modbus float only

In most applications, only the value itself is used for further processing. In this case, using the floating point representation of the meter values via Modbus is particularly suitable.

By omitting the meta information, the data representation via Modbus can be more compact for saving memory or communication traffic. Setting the parameter *Modbus float only* in the ***Server*** tab consolidates the Modbus address space. Only the serial number of the meter and the meter values themselves are then available. The serial number is represented as integer and the values as floating point numbers. This reduces the data grid to 2 Modbus registers. The device entry is then not available.

The meter entry contains the serial number of the meter only. It is formatted as follows:

Offset	Description	Data width	Comments
0-1	Serial number	32 Bit	The serial number is encoded in hexadecimal. Unlike M-Bus or wM-Bus, this is an integer and not a BCD.

Table 37: Modbus registers representing the reduced data set of a meter

The meter value entry only consists of the scaled floating point value, which is derived from the integer value of the meter, if the meter does not provide a floating point value. The meter value entry is formatted as follows:

Offset	Description	Data width	Comments
0-1	Meter value	32 Bit	Floating point meter value (scaled), IEEE 754

Table 38: Modbus registers representing the reduced data set of a meter value

11.4.5 Modbus multi slave

Depending on the usage and further processing of the data, it may be useful to logically separate meter data of different meters.

When setting the parameter *Modbus multi slave* in the ***Server*** tab, each of the meters gets its own Modbus address space. Each M-Bus slave in the meter list is thus managed as a separate virtual Modbus slave with its own Modbus address. The slave address of the respective meter is then displayed in the column *Register* in the ***Meter*** tab at the meter entry and can be adjusted there (see Section 4.4). The meter value entries show the corresponding Modbus register addresses within this virtual Modbus slave.

- If there are meters in the meter list, the addresses must be re-assigned after activating or deactivating the multi slave functionality.
- For selecting multiple entries in the meter list, the keys ⟨**SHIFT**⟩ or ⟨**CTRL**⟩ can be held down.
- The functions **Allocate** and **Deallocate** from the context menu can be used to reset or re-allocate the slave addresses and Modbus register addresses.

This allows the dedicated request of data of only one meter at a time. The addressing mechanism of the registers then restarts for each meter. This allows creating and using macros and other automation approaches when programming the Modbus client, if the same meter type is used several times.

- Since the slave address can only accept values 1-247, no more than 247 meters can be addressed logically.
- The slave address 0 is a broadcast address.
- The slave address 255 addresses the device itself.
- For each slave address, the register layout follows the conventions according to Section 11.3 or Section 11.4.4.

11.5 Application hints

11.5.1 How often is the data updated?

The meter data is read out independently of the Modbus requests. The meter data is updated on each automatic or manual reading of a meter and is then available via Modbus. You can set the required cycle time in the ***Configuration*** tab for all meters or an individual cycle time for particular meters in the ***Meter*** tab in the column *Cycle*.

11.5.2 How to detect if the meter is read or the value is up to date?

For monitoring applications for example in automation (e. g.: SCADA system, PLC), the quality of a value is very important. It is therefore recommended to check whether a meter could be read at all and whether the meter value is up to date.

The data set of the meter entry contains, among other things, the readout timestamp and a flag register that provides information about the readout status.

If the meter was read out completely during the last cycle, the flag register has the value *0*. Possible values are explained in Table 32. The readout timestamp can also be used for evaluating if meter data is up to date or since when no new data was received from the meter (in case of error).

11.5.3 Which data type has to be used?

The data set of the meter value entries contains two different data types. On the one hand there is the unscaled meter value as *INT64* value in combination with a scaling factor, and on the other hand there is the scaled value as *FLOAT32* value.

For exact billing applications, the *INT64* value is preferred, since this can be processed further without loss of accuracy. However, not all Modbus clients are capable of processing 64-Bit data. It should also be noted that the scaling factor has still to be multiplied. The *INT64* value can therefore be assumed to be a fixed point value.

- The scaling is defined and provided by the meter. Therefore the scaling might change at run time.

For monitoring applications for example in automation (e. g.: SCADA system, PLC), the *FLOAT32* value is more suitable. The subsequent scaling is hence not required and the accuracy of about 7 digits is sufficient in most cases.

11.5.4 What is the unit of value?

The data set of the meter value entries contains, among other things, the unit and the scaling of the value. An explanation can be found in Table 33.

11.5.5 How many Modbus masters can request data simultaneously?

In the standard configuration, the devices from solvimus GmbH allow up to 5 simultaneous Modbus TCP connections.

11.5.6 How can the data be mapped automatically?

Each data set, i. e. device entries, meter entries and meter value entries, contains a type field (see Table 31, Table 32 and Table 33). This field can be used to automatically identify the type of the entry.

If the register addresses in the ***Meter*** tab are assigned automatically (see Section 4.4), the values are arranged in the Modbus memory one after the other, in logical groups:

- Device entry
  - Meter entry 1
    - Meter value entry 1
    - Meter value entry 2
      ⋮
    - Meter value entry x
  - Meter entry 2
    - Meter value entry x+1
    - Meter value entry x+2
      ⋮
    - Meter value entry x+y

    ⋮
  - Meter entry n
    - Meter value entry x+y+..+1
    - Meter value entry x+y+..+2
      ⋮
    - Meter value entry x+y+..+z

This allows an iterative processing of the complete Modbus data. Using the grid of 10 registers, the hierarchy and the mapping determined automatically. The content of the respective entry can thus be used for reproducing the meter list in the ***Meter*** tab.

11.5.7 Writing meter value entries via Modbus

An access in write mode is possible via Modbus. The states of digital outputs, meter values or other parameters can be set. However, the implementation is highly specific and varies considerably. This option is deactivated by default.

Please contact our customer support for more information:
E-Mail: support@solvimus.de
Phone: +49 3677 7613065

11.6 Troubleshooting the Modbus slave

11.6.1 Why does the value via Modbus differ from the value on the web-based front end?

Deviations of a value can have various causes. A list is provided to explain the most common causes of error:

- If the web-based front end or the ***Meter*** tab is shown for some time, it is possibly not showing the current values. Please reload the ***Meter*** tab by using the **Reload** button.
- If you are comparing the web-based front end to the *FLOAT32* value, there may be small deviations from about the 7th digit. These are errors of accuracy coming from the data type.
- Please check if the correct data type is used. The meter values are available as *INT64* (plus scaling) and *FLOAT32*.
- Please check if the data alignment, especially the word order, is correctly set to *MSW* or *LSW* (see Section 11.4.3).

- Please check the register addresses. Inspect whether the counting starts from *0* or *1*. Please also take the offsets in the respective data set into account (e. g. for using the *FLOAT32* value).
- In case of using integer values, please check if the Modbus master can handle data types having 64 Bit.
- In case of using floating point values, please check if the Modbus master can handle *FLOAT32*. Fixed point data representation is not supported.
- Please use the test data to check various settings (see Section 11.4.2).

If errors could not be eliminated, please contact our customer support:
E-Mail: support@solvimus.de
Phone: +49 3677 7613065

11.6.2 Why is the device/the Modbus server not responding?

Connectivity issues on Modbus TCP or Modbus UDP can have various causes. A list is provided to explain the most common causes of error:

- Check the IP settings. Are the Modbus master and the Modbus client in the same IP address range and in the same subnet? If not, is the gateway or the route configured correctly? Pinging the slave from the master device can be used for debugging.
- Check if Modbus is activated in the ***Server*** tab of the device.
- Check if the port on the master and the slave are matching (usually 502). Please also check if another service on the device is blocking the port by mistake.
- Check if a firewall is blocking the communication.
- Check if the correct slave address is used on the Modbus.

If errors could not be eliminated, please contact our customer support:
E-Mail: support@solvimus.de
Phone: +49 3677 7613065

12 Transmission of read out meter data via BACnet IP

12.1 General information

BACnet (Building Automation and Control Networks) is a network protocol for building automation. It is standardised by ASHRAE, ANSI and as ISO 16484-5.

- This device is a BACnet server.

The devices from solvimus GmbH are, if option available, a BACnet IP Server. The BACnet communication requires the setup of a UDP connection between a client (e. g.: PC, controller or BMS) and the server (this device). The UDP port configured in the ***Server*** tab is used for the BACnet communication. This is configured to 47808 by default (see Section 4.8).

- If there is a firewall between server and client, it must be ensured that the configured UDP port is enabled.

12.1.1 Services implemented

The following BACnet services are supported by the device:

Service	implemented
BACnet Operator Workstation (B-OWS)	No
BACnet Advanced Operator Workstation (B-AWS)	No
BACnet Operator Display (B-OD)	No
BACnet Building Controller (B-BC)	No
BACnet Advanced Application Controller (B-AAC)	No
BACnet Application Specific Controller (B-ASC)	Yes
BACnet Smart Sensor (B-SS)	No
BACnet Smart Actuator (B-SA)	No

Table 39: Implemented BACnet services

12.1.2 Supported BACnet Interoperability Building Blocks (Annex K)

The following additional BACnet Interoperability Building Blocks are supported by the device:

Capability	supported
Able to send segmented messages (Window Size = 4)	Yes
Able to receive segmented messages	No

Table 40: Additional BACnet Interoperability Building Blocks

12.2 Configuration via the web-based front end

The BACnet function is activated and configured via the ***Server*** tab. The parameters are described in the Section 4.8. The settings are explained in detail below.

12.2.1 BACnet active

The BACnet IP function can be activated by the parameter *BACnet active*. *BACnet IP* is a widespread and common variant of BACnet on the base of IP and utilises UDP for communication.

12.2.2 BACnet config network, BACnet IP, BACnet netmask and BACnet broadcast

The device supports the activation of a second, virtual network interface for the BACnet service. The device can thus be integrated in two logical networks via a physical network connection. The function is activated by the parameter *BACnet config network*.

The second, virtual network interface is configured by the parameter *BACnet IP*, *BACnet netmask* and *BACnet broadcast*.

✔ The parameters *BACnet IP* and *BACnet netmask* are independent of the default settings in the ***General*** tab.

12.2.3 BACnet BBMD

When using BACnet IP, diverse messages to the Broadcast-MAC-address (FF:FF:FF:FF:FF:FF) are sent into the local network. All BACnet devices in the local network receive the message and respond accordingly. But routers transmitting in other subnets do not forward these messages. To remedy this problem, the BACnet Broadcast Management Device (BBMD) was introduced. The BBMD forwards IP broadcast messages, guided by a Broadcast Distribution Table (BDT), in other subnets. The IP address of BBMD in the network can be configured by the parameter *BACnet BBMD*. When configuring, the Bacnet BBMD remote unit must be detectable by network broadcast.

12.2.4 BACnet port

Both IP-based protocols use the port set by the parameter *BACnet port*. It is set to 47808 (0xBAC0) by default.

ℹ If the parameter *BACnet port* is set to a value used by other services (e. g.: HTTP: Port 80), these services can block each other and access on the device is inhibited.

12.2.5 BACnet device ID, BACnet device name and BACnet location

The parameters *BACnet device ID*, *BACnet device name* and *BACnet location* serve to identify the device in the BACnet network.

The default settings are as follows:

Identifier	Default value
BACnet device ID	1
BACnet device name	*Name of the device*
BACnet location	metering

Table 41: Default values for the identification parameters

12.3 Data representation

12.3.1 Meter values

All meter values are represented as "Analog Value" at the BACnet interface. The data are structured as follows, where a question mark is a placeholder for specific values:

```
Analog Value [1..n]
{
  object-identifier: (analog-value,1)
  object-name: "Name Meter 1"
  object-type: analog-value
  present-value: ?
  description: "Description Meter 1"
  status-flags: ?
  event-state: ?
  out-of-service: ?
  priority-array: {NULL, NULL, NULL, NULL, NULL, NULL, NULL,
    NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL}
  units: 95
  relinquish-default: ?
  cov-increment: 0.2
}
```

12.3.2 BACnet Device object

The Device object of the device is structured as follows, where a question mark is a placeholder for specific values:

```
object-identifier: (device,2)
{
  object-name: "ctrl_cb_buero1"
  object-type: device
  system-status: ?
  vendor-name: www.bektasic.de
  vendor-identifier: 725
  model-name: "www-ctrl"
  firmware-revision: "1.3.2"
  application-software-version: "14"
  location: "Buero CB"
  description: "www-controller for Automation"
  protocol-version: 1
  protocol-revision: 12
  protocol-services-supported:
  (
    +-- readProperty
    +-- readPropertyMultiple
    +-- deviceCommunicationControl
    +-- i-Have
    +-- i-Am
  }
  object-list:
  {
    (device,2),
    (analog-output,1),(analog-output,2),(analog-output,3),
    (analog-output,4),(analog-value,1),(analog-value,2),
    (analog-value,3),(analog-value,4),(analog-value,5),
    (analog-value,6),(analog-value,7),(analog-value,8),
    (analog-value,9),(analog-value,10),(analog-value,11),
    (analog-value,12),(analog-value,13)..(analog-value,n))
  }
  max-apdu-length-accepted: 1476
  segmentation-supported: 1 // only transmit
  max-segments-accepted: 4
  local-date: ?
  local-time: ?
  utc-offset: -60
  daylight-savings-status: ?
  apdu-segment-timeout: 3000
  apdu-timeout: 3000
  number-of-apdu-retries: 3
  device-address-binding: ?
  database-revision: 1
}
```

12.4 Troubleshooting

12.4.1 Why is the device/the BACnet server not responding?

Connectivity issues on BACnet IP can have various causes. A list is provided to explain the most common causes of error:

- Check the IP settings. Are the BACnet IP client and BACnet IP server in the same IP address range and in the same subnet? If not, is the gateway, the BBMD and the route configured correctly? Pinging the slave from the master device can be used for debugging.

- Check if BACnet IP is activated in the ***Server*** tab of the device.
- Check if the port on the master and the client are matching (usually 47808). Please also check if another service on the device is blocking the port by mistake.
- Check if a firewall is blocking the communication.

For further analyses it is helpful to record the network traffic. For that, tools like *Wireshark* can be used on a PC in a network or *tcpdump* in the command line of the device (see Section 10.1.2).

- The tool *Wireshark* can be found at: https://www.wireshark.org/
- An instruction for *tcpdump* can be found at: https://www.tcpdump.org/manpages/tcpdump.1.html

If errors could not be eliminated, please contact our customer support:
E-Mail: support@solvimus.de
Phone: +49 3677 7613065

13 Accessory

The solvimus GmbH recommends the external power supply PHOENIX CONTACT STEP-PS/1AC/24DC/1.75, article number of the solvimus GmbH: 103960.