User Guide
USER GUIDE: MC100

MC Technologies GmbH – July 2026

4G
LTE Cat 1

WLAN

GNU/Linux

RS-232
& RS-485

USB 2.0

Optional GNSS

MC Technologies GmbH
Kabelkamp 2
30179 Hannover
Germany

www.mc-technologies.com
info@mc-technologies.com
Tel: +49-511-676 999-0
Fax: +49-511-676 999-150

UG_MC100_V5.0_2026_05_07

1 Revision History

Version	Date	Author	Description
4.0	01.06.2025	hussein.soueidan@mc-technologies.com	Change document design and minimal changes in content
5.0	07.05.2026	sascha.dikhoff@mc-technologies.com hussein.soueidan@mc-technologies.com	• Added: • Safety instructions • Screenshots updated • default password • radio information • power diconnection
5.2	24.06.2026	Sascha.dikhoff@mc-technologies	Added chapter about cyber security

2 Table of Contents

3 Introduction

Thank you for choosing an MC Technologies product.

The MC100 is a family of Linux-based LTE gateways optimized for machine-to-machine (M2M) and industrial Internet of Things (IIoT) applications. A wide variety of models offers the perfect feature set for diverse industrial use cases in an economical way.

Options include:

• Different cellular modems optimized for specific applications (e.g. NB-IoT, 450 MHz)

• Radio interfaces: GNSS location services, WLAN, Bluetooth, wM-Bus and LoRaWAN

• Serial interfaces: RS-232, RS-485 (Modbus), CAN, 1-Wire, M-Bus

• SensT2 expansion: PT100/PT1000 RTD, 0–20 mA I/O, 10 V DAC, 1-Wire

• GPIO expansion

• SensorBox

Additional model-specific information may be found in model-dedicated appendices on our website:

https://mc-technologies.com/service-support/download/

3.1 Purpose of the Manual

This document provides comprehensive instructions for the installation, commissioning, operation, and maintenance of the IoT Gateway MC100. It aims to ensure that users can effectively utilize the device while adhering to all relevant safety and regulatory standards.

3.2 Target audience

This manual is intended for installers, system integrators, and operators who are responsible for installing, configuring, and operating the MC100 in industrial and commercial environments.

The reader is expected to have basic technical knowledge of electrical systems and network technologies. Installation and commissioning must be carried out by qualified personnel.

3.3 Intended Use

The MC100 is intended exclusively for professional and industrial use. The device is designed for integration into industrial equipment, control cabinets, and embedded systems, and is used for data communication, remote monitoring, and control applications in commercial and industrial environments. Installation, configuration, and maintenance must be carried out by qualified personnel with appropriate technical knowledge of electrical and network systems. The product is not designed or intended for direct consumer use. The device must be operated within the technical specifications and environmental conditions defined in this documentation.

4 Warranty provisions

Unauthorized use, non-observance of this documentation, the operation or maintenance by insufficiently qualified persons, and unauthorized modifications exclude the manufacturer's liability for resulting damages. Any modification to the device will void the manufacturer warranty. The provisions of our General Terms of Sale (AGB) apply. These can be found on our website:

https://mc-technologies.com/en/agb-aeb

4.1 Limitation of liability

The manufacturer and seller are not liable for damages caused by improper use, installation or maintenance. This includes consequential damage, personal injury, damage to property and damage caused by failure to observe the safety instructions. This exclusion of liability does not affect the statutory warranty claims. Liability is assumed for material and manufacturing defects within the statutory warranty period.

4.2 Approved accessories

This device must only be operated with suitable accessories approved for the appliance.

4.3 Inspection for damages

Check the device and all components for damages or anomalies before use. Do not use the appliance if it is damaged or shows signs of wear. Make sure that all cables are in a good condition. The appliance must not be used if cables or plugs are damaged.

4.4 Technical limits

The product is exclusively intended for use within the technical limitations and maximum ratings specified in this document. The following limitations must be observed in particular:

- The ambient temperature must not be exceeded or dropped below limits.
- The maximum air humidity must not be exceeded, and condensation must be avoided.
- The supply voltage must be within limits and maximum input ratings must not be exceeded.
- The maximum switching voltage and current must not be exceeded.

5 Safety instructions

These instructions enable the safe and efficient handling of the product. The instructions are an integral part of the product and must always be kept accessible to installation, maintenance, commissioning, and operating persons.

The safety and maintenance instructions must be strictly followed to ensure safe operation of the product. Only the consideration of all safety guidelines ensures protection of persons and the environment against hazards and the safe and trouble-free operation of the product.

General safety regulations and local guidelines for the area of application of the device as well as for the prevention of accidents along with procedures and operation instructions with safety-critical information must be followed strictly.

5.1 Obligations of the operator

The operator must follow regional regulations regarding the operation, functional testing, repair and maintenance of electronic devices at all times.

5.2 Qualification of installers

Installation and maintenance of the product may only be carried out by trained authorised installers which possess the necessary levels of qualification to ensure safe maintenance and operation. The qualified installer must have read and understood this documentation and follow its guidelines and instructions. This product may only be operated by or under the supervision of trained persons.

5.3 Guidelines for transport and storage

- Do not expose the product to moisture or other potentially harmful environmental conditions (radiation, gases, etc.) during transport or storage
- Protect the product from shocks during transport and storage (e.g. by using air-cushioned packaging)
- Before installing the product, check for damages caused by improper transport or storage. Damage in transit must be noted on the shipping documents. All claims for damages must be made immediately and before being handed to the carrier or company responsible for the storage or logistics respectively

5.4 Safety precautions

	**Electrostatic discharges, short circuits and voltage spikes increase the risk of fire, cause damage to the product and may cause bodily harm.** Observe the general precautions for handling electrostatically sensitive components. Turn off the power before performing any work on an electric device. Ensure a suitable surge protection is installed. Do not operate the device with visible or otherwise known damage.
	**Damage due to improper handling, repairs and modifications increase the risk of fire, cause damage to the product and may cause bodily harm.** It is not permitted to open the product for repair work or modifications beyond the removal and insertion of the provided plug-in cards. Ensure power accessory is well-suited for the purpose. Keep the device away from children and animals to prevent hazards like choking of parts and danger due to biting.
	**Dust, debris, moisture and liquids from the surrounding area could get inside the product and increase the risk of fire, cause damage to the product and may cause bodily harm.** The product must not be used in humid environments or in the immediate vicinity of water or other liquids. Install the product in a clean, dry place protected from splashing water, dust and debris that could enter the device.

	**Open flames, harsh chemicals and flammables including aerosols increase the risk of fire, cause damage to the product and may cause bodily harm.** The product must be kept away from direct sunlight, open flames, harsh chemicals, flammables, explosives and aerosols.
	**Extreme temperatures, insufficient heat dissipation or ventilation increase the risk of fire, cause damage to the product and may cause bodily harm.** Operate the device in a well-ventilated area away from direct sunlight. To ensure good heat dissipation, do not enclose or cover the device or its ventilation holes. The device must be operated well within the operating temperature limits.
	**Strong magnetic or electric fields, vibrations and shocks cause malfunctions and may damage the device.** Keep the device away from electronic appliances that generate strong magnetic or electric fields, such as a microwave oven, radar, electrical motor or generator. Ensure the device is fixed properly and avoid high accelerations.
	**A too small distance between antennas and persons might affect their health.** Be aware that wireless devices may affect the performance of e.g. hearing aids or pacemakers. The antennas must be placed at least 20 cm away from persons during operation. If applicable, respect the rules and regulations for device operations set forth in hospitals and health care facilities.

5.5 Electric safety requirements

Electrical installation, maintenance and commissioning of products operated with high or low voltage (>= 50 V AC or >=120 V DC respectively) may only be carried out by persons who, due to their specialist training, knowledge and experience including knowledge of the relevant standards and regulations, are authorized to carry out work on electrical systems and independently detect and avoid possible hazards e.g. as described in VDE 1000-10.

The electrical installation must be conducted in line with all applicable standards an(e.g. IEC, EN, VDE,…). The electrical installation must only be operated when defect-free and after testing using a recognized examination procedure like VDE 0100-600 / IEC 60364-6, with all necessary safety measures in place before operation.

Equipment must be inspected immediately after installation, expansion or any other change e.g. according to DIN EN 50699 (VDE 0702). The inspector must be qualified for inspection as e.g. described in TRBS 1203 to perform inspections and determine the inspection intervals through a risk assessment process. The resulting inspection report must be archived. Equipment passing the inspection should be marked by an inspection sticker summarizing the results and showing the next due date.

5.6 Electric safety precautions

- Improper installation can lead to electric shocks, short circuits or fires
- The cabling must be suitable e.g. according to DIN VDE 0100-520
- The supplying electric installation must be equipped with a residual-current device (RCD)

5.7 Cybersecurity

- General security concept
  This product incorporates security mechanisms designed to support safe operation in networked industrial environments and to protect against unauthorized access and misuse. The MC100 is intended for professional and industrial applications and must be integrated into a secure system environment. Secure operation depends on correct installation, configuration, and usage in accordance with this manual and applicable IT security guidelines.
- Access protection
  The device is delivered with default login credentials which must be changed immediately after commissioning. Strong passwords should be used. Access can be restricted via authentication, network configuration and firewall settings.
- Network security
  Only required services should be enabled. Secure protocols such as HTTPS, SSH and VPN should be used. Additional protection (e.g. network segmentation) is recommended for public networks.
- Software and updates
  Regular firmware updates must be applied. Only trusted firmware sources should be used and updates should be tested before deployment.
- Responsibility of the operator
  The operator is responsible for secure integration into the network, managing access rights and following IT security policies.
- Regulatory compliance
  This product complies with Directive 2014/53/EU including applicable cybersecurity requirements (Article 3.3, where relevant). Further details are provided in the EU Declaration of Conformity.
- Security incidents and reporting
  In case of suspected security incidents (e.g. unauthorized access, unexpected system behavior or detected vulnerabilities), appropriate measures must be taken immediately.

  The operator or system integrator is responsible for:
  - isolating affected systems if necessary
  - analyzing and mitigating the issue
  - restoring secure operation

  Where applicable, security incidents must be reported in accordance with internal

procedures and applicable regulations.

For support and vulnerability reporting, please contact:
support@mc-technologies.com

6 Product label

The label of the product is located on the side of the product. Besides information such as a E1 approval marking, serial number, MAC address, IMEI number and operating voltage, it may contain the following markings:

6.1 Special waste

This symbol indicates that the device must be disposed of separately from residual waste at suitable collection points. Please refer to the environmental protection section below and the disposal section at the end of this document.

6.2 CE marking

CE

- By affixing the CE marking, MC Technologies GmbH declares compliance with applicable EU legislation.
- Hereby, MC Technologies GmbH declares that the radio equipment type MC100 is in compliance with Directive 2014/53/EU (Radio Equipment Directive). The full text of the EU declaration of conformity is available at the following internet address: https://mc-technologies.com/doc-mc100/

7 Environmental protection

The product and the associated transport packaging are made largely from recyclable raw materials. It can be sent to MC Technologies GmbH for proper recycling. At the end of its useful life, the product must not be disposed as household waste.

The disposal of the product and its packaging must be carried out in accordance with all relevant environmental protection regulations. Recycle responsibly by separating the packaging materials like cardboard and paper from plastic and use the dedicated waste collection systems. Refer to the disposal section at the end of this document for instructions on how to return the product to MC Technologies for recycling.

8 Technical specifications

8.1 Physical characteristics and limitations

Physical characteristic / limitation	Value
Power supply	8 V ... 30 V DC (min. 14 W)*

Dimensions (W x H x D)	120 x 75 x 35 mm
Weight	~ 230 g (~ 4.23 oz)
Operating temperature	-20 °C to +70 °C (Sensorbox variant: -20°C to + 55°C)
Housing material	ABS

** See section below for details on power supply requirements.*

8.2 Radio Information

This device contains radio transmitters operating in the frequency bands listed below. The maximum transmitted radio-frequency power is specified for each technology.

The supported frequency bands and output power depend on the hardware variant used.

For safe operation, a minimum distance of 20 cm must be maintained between the device's antennas and any person during operation.

8.3 Mobile network features

The device integrates a cellular module depending on the product variant:

The default product variant is equipped with a Quectel EC21-E module and the variants marked with global with a Quectel EG25-G module.

Variants	Default module	Global module
Radio Module	Quectel EC21-E	Quectel EG25-G
Frequency bands LTE FDD	B1 (2100 MHz), B3 (1800 MHz), B5 (850 MHz), B7 (2600 MHz), B8 (900 MHz), B20 (800 MHz)	B1 (2100 MHz), B2 (1900 MHz), B3 (1800 MHz), B4 (1700/2100 MHz), B5 (850 MHz), B7 (2600 MHz), B8 (900 MHz), B12 (700 MHz), B13 (700 MHz), B18 (800 MHz), B19 (800 MHz), B20 (800 MHz), B25 (1900 MHz), B26 (850 MHz), B28 (700 MHz)
Frequency bands LTE TDD	-	B38 (2600 MHz), B39 (1900 MHz), B40 (2300 MHz), B41 (2500 MHz)
Frequency bands WCDMA	B1 (2100 MHz), B5 (850 MHz), B8 (900 MHz)	B1 (2100 MHz), B2 (1900 MHz), B4 (1700/2100 MHz), B5 (850 MHz), B6 (800 MHz), B8 (900 MHz), B19 (800 MHz)
Frequency bands (GSM):	B3 (1800 MHz), B8 (900 MHz)	B2 (1900 MHz), B3 (1800 MHz), B5 (850 MHz), B8 (900 MHz)
Maximum RF Output Power	LTE (FDD/TDD): up to 23 dBm UMTS (WCDMA): up to 24 dBm	LTE (FDD/TDD): up to 23 dBm UMTS (WCDMA): up to 24 dBm

	GSM: up to 33 dBm (850/900 MHz), up to 30 dBm (1800/1900 MHz) GSM (8-PSK): up to 27 dBm (850/900 MHz), up to 26 dBm (1800/1900 MHz)	GSM: up to 33 dBm (850/900 MHz), up to 30 dBm (1800/1900 MHz) GSM (8-PSK): up to 27 dBm (850/900 MHz), up to 26 dBm (1800/1900 MHz)

8.4 GNSS receiver

Depending on the product variant, the device may include a GNSS receiver integrated in the cellular module (e.g. Quectel EC21-E or Quectel EG25-G).

The GNSS functionality supports satellite navigation systems such as GPS and, depending on the module variant, additional systems including GLONASS, Galileo and BeiDou.

The GNSS receiver operates in the following frequency ranges:

1560 – 1610 MHz

1170 – 1210 MHz

The GNSS functionality is receive only (no radio transmission).

Note: GNSS performance depends on antenna placement, environmental conditions and satellite visibility. For optimal performance, a suitable external GNSS antenna with clear view of the sky is recommended.

8.5 WLAN features

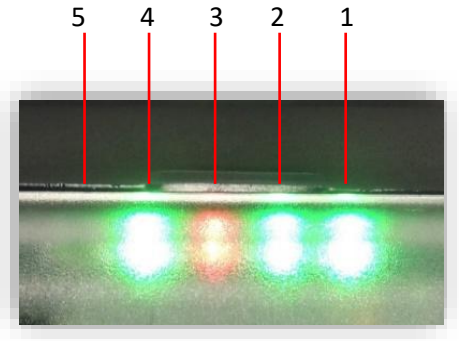
The device integrates a 2.4 GHz WLAN module based on IEEE 802.11 b/g/n (Wi-Fi 4), designed for reliable wireless communication in embedded and industrial IoT applications.

Variants	Variants with WLAN
Radio Module	Embedded module with IEEE 802.11 b/g/n in the 2.4 GHz frequency band (Wi-Fi 4)
Frequency bands 2.4 GHz	2400 – 2483.5 MHz
Maximum RF Output Power	up to 20 dBm (100 mW)

9 Ports, display and operating elements

The following figures show a maximum equipped version of the MC100. Depending on the variant, your MC100 may not have all connections, display or control elements.

9.1 LED indicators



LED	Color	Name	Description
LED 1	Green	Power	Power indicator
LED 2	Green	Info	Customizable
LED 3	Red	Warn	Warning state (under voltage situation, system upgrade in progress, custom warning)
LED 4	Green	Modem (NET LED)	Rapid blinking: Packet data transfer Long-time off: Searching for connection Long-time on: Connected
LED 5	Green	Status	Blinking: Booting Steady on: Ready

9.1.1 NET LED signal patterns

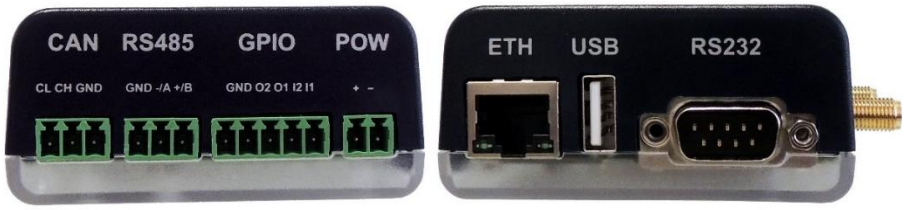
This LED indicates the status of the integrated GSM module.

Blink pattern	Network Status
Off	Modem is inactive
Short interval (200ms on, 1800ms off)	Network search
Long interval (1800 ms on, 200 ms off)	Idle state
Flickering (125 ms on, 125 ms off)	Data transfer
Always on	Call in progress

9.2 Buttons

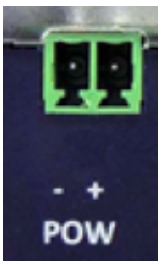
Name	Function
Reset	Used for resetting the gateway to factory defaults
User	Customizable

9.3 Base connectors



Name	Connector	Description
POW	Terminal socket	Power supply
GPIO	Terminal socket	Digital I/O
RS485	Terminal socket	RS-485 interface
CAN	Terminal socket	CAN 2.0B interface
ETH	RJ-45	Ethernet 10/100 Base-T
USB	USB type A socket	USB 2.0 host interface
RS-232	DE-9 male	RS-232 serial interface

9.3.1 Power (POW)



The gateway can be operated using LPS power supplies with a supply voltage of 8 - 30 V DC and a minimum rated output power of 14 W. The output power can be calculated as the product of voltage and current. For example, a 12 V / 1.2 A power supply is suitable, as its maximum power output is 14.4 W.

An LPS (Limited Power Source) is a power source that meets the requirements of IEC 62368-1, including limitations on voltage (max. 60 V DC) and power to ensure safe operation.

If an external power supply unit connected to the mains is used, the disconnection from the mains is achieved by disconnecting the power supply from the mains (e.g. by unplugging the mains plug).

The means of disconnection must remain easily accessible at all times.

Warning: Make sure the polarity is correct as it might otherwise destroy the device.

Port	Description
POW -	Power supply, negative potential
POW +	Power supply, positive potential

9.3.2 RS485

Warning: The base RS-485 interface is not galvanically isolated.

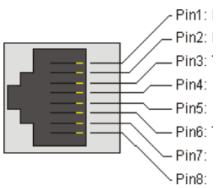
Port	Description
-/A	Inverted line
+/B	Non-inverted line
GND	Ground

9.3.3 CAN

Warning: The base CAN 2.0 B interface is not galvanically isolated.

Port	Description
CH	CAN High
CL	CAN Low
GND	Ground

9.3.4 Ethernet



Port	Signal	Description
1	TD+	Transmit plus
2	TD-	Transmit minus
3	RD+	Receive plus
4	CAPa	Internal 100 nF capacitance to GND
5	CAPb	Internal 100 nF capacitance to GND
6	RD-	Receive minus
7	Not connected	
8	Shield	

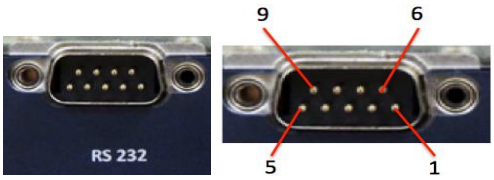
LED (Pos.)	Signal	Description
Left	SPEED	1 Blinks: LED Blinks once = 10Base link 2 Blinks: LED Blinks twice = 100Base link
Right	LINK	ON: Network link has been established Blinking: Network activity has been detected

9.3.5 USB

The USB 2.0 port is a standard type A socket.

9.3.6 RS-232

The RS-232 port is a DSUB-9 male socket (DCE).



Port	Signal	Description
2	TXD (OUT)	Transmit line (of the modem)
3	RXD (IN)	Receive line (of the modem)
5	GND	Ground
7	RTR / RTS (IN)	Ready To Receive / Request To Send (Terminal is ready to receive)
8	CTS (OUT)	Clear To Send (Modem is ready to receive)

Note: A hardware handshake using the RS-232 interface is not possible.

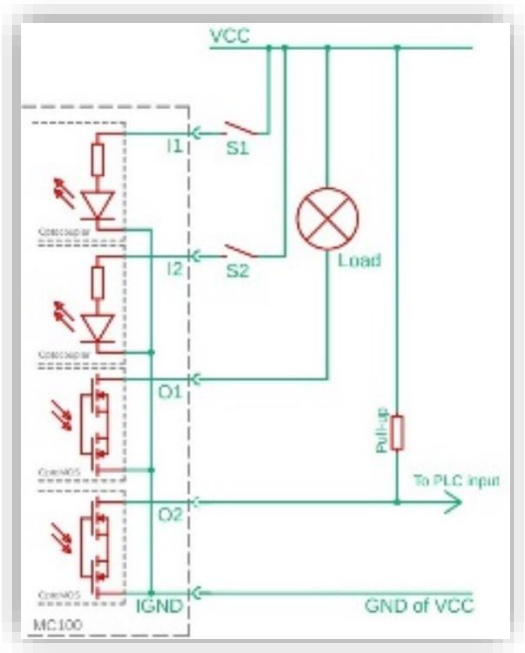
9.3.7 Digital IOs (GPIO)

The digital outputs allow to directly drive a load of up to 120 mA RMS/DC with a maximum voltage of 30 V DC (relay, LED etc.) or switch a signal (e.g. PLC). When active, it connects the output with the isolated ground reference (IGND) using inbuilt solid-state relays (OptoMOS). Otherwise, the connection is High-Z.

The digital inputs can detect a DC voltage signal > 6V (active-high, max. 30 V DC) in respect to the isolated ground reference (IGND) using optocouplers.

The circuit is exemplified in the figure below.

Port	Description
I1	Digital input 1, DC voltage 0 to 30V, switching threshold approx .6V DC
I2	Digital input 2, DC voltage 0 to 30V, switching threshold approx .6V DC
O1	Digital output 1, switching capacity max. 300 mA
O2	Digital output 1, switching capacity max. 300 mA
IGND	I/O ground, electrically isolated from standard GND of the device



9.3.8 Antenna connectors

All antenna connectors are SMA female.

Port	Description
LTE	Main LTE antenna SMA socket
DIV	Diversity LTE antenna SMA socket
GPS	GPS antenna SMA socket
WIFI	WLAN antenna SMA socket

9.3.9 SIM card slot

Port	Description
SIM	SIM card slot

9.4 MC100 GPIO connectors

Detailed pinouts can be found in the chapter *Wired interfaces.*

Port	Description
Isolated inputs	8 digital inputs
Isolated outputs	8 digital outputs (PhotoMOS solid state relais)
0 – 10 V	2 analog inputs 0 to 10 V
0 – 20 mA	2 analog inputs 0 to 20 mA
PWM	1 open-drain PWM (Pulse Width Modulation) output

9.5 MC100 SensT2 connectors

Detailed pinouts can be found in the chapter *Wired interfaces.*

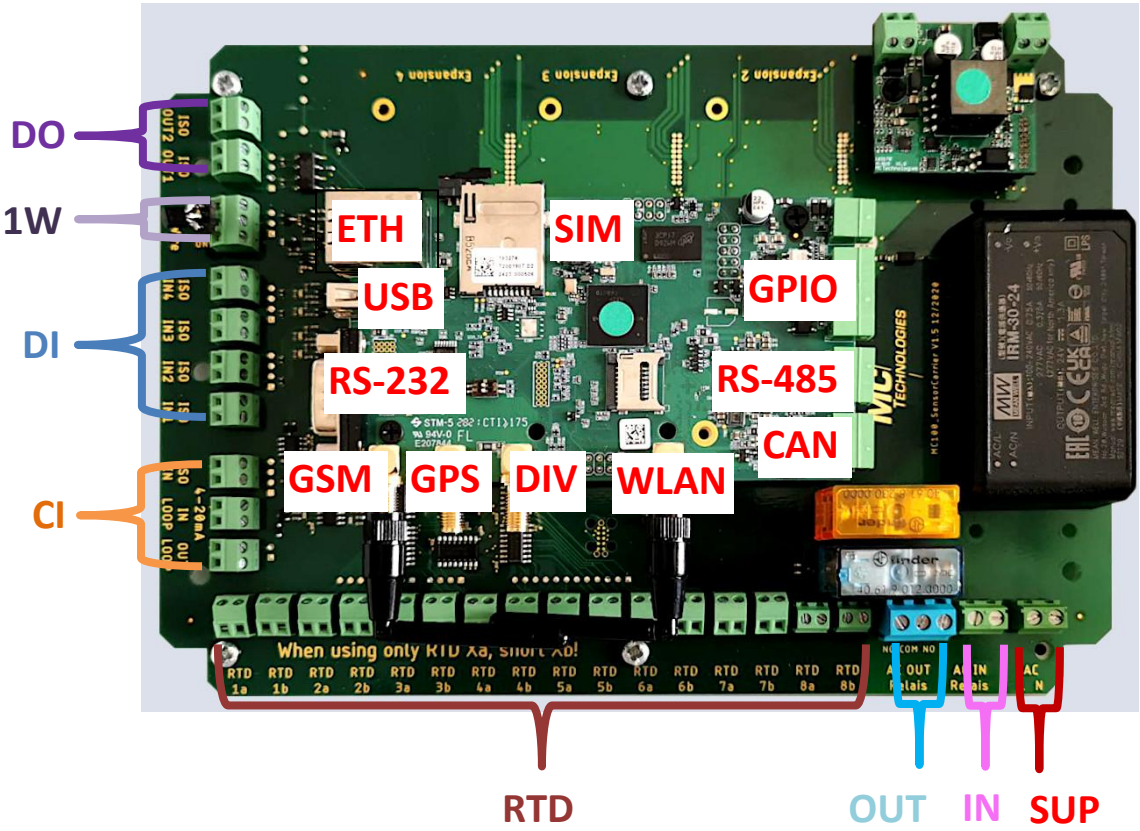
Port	Description
Isolated RS-485	1 x Serial RS-485 interface (galvanically isolated)
Isolated inputs	8 x Digital inputs
Isolated outputs	2 x Digital outputs (OptoMOS solid state relay)
PT100/PT1000	4 x PT100/PT1000 inputs
4 – 20 mA outputs	2 x Analog 4 – 20 mA outputs
0 – 10 V	2 x Analog inputs 0 – 10 V
4 – 20 mA LS	2 x Analog inputs 4 – 20 mA LS (Loop Supply)
1-Wire	1 x 1-Wire bus

9.6 MC100 SensorBox connectors

The SensorBox requires certified technicians for installation and maintenance. Please note the electrical safety requirements chapter in the beginning of this document and the special installation instructions in the chapter *Installation.*

Detailed pinouts can be found in the chapter *Wired interfaces.*

Warning: Do not connect the power of the base board and the AC power inlet simultaneously!



Port	Description
RS-485	Galvanically isolated RS-485 interface
DI	Galvanically isolated digital input (active-high, 5 … 30 V DC)
DO	Galvanically isolated digital output (uses PhotoMOS solid state relay)
1W	1-Wire bus
IN	230 V electromagnetic input relay
OUT	230 V electromagnetic output relay
SUP	85 … 264 V AC power inlet
CI	4-20 mA current input
M-BUS	M-Bus master interface
BPO	M-Bus power supply

10 Installation

10.1 Antenna

Mount the supplied antenna to the SMA connector on the MC100. Check whether the local network coverage of the mobile network provider is sufficient.

Warning: Persons must be at least 20 cm away from the antenna during device operation.

10.2 Inserting the SIM card

Switch off the power supply and remove all connection cables before inserting the Mini SIM card into the slot on the side. Mind the sliding lock mechanism.

10.3 Vehicle installation

The installation of the MC100, its periphery such as cables and antennas as well as the electrical connections in a motor vehicle must be carried out by a qualified specialist workshop. Only MC100 with an E1 certification are allowed to be installed or operated in vehicles (marked on their label).

10.4 SensorBox installation

As the SensorBox is a product for IP65 installations with relatively high voltages, special care must be taken. It requires certified technicians for installation and maintenance. Besides, the electrical safety requirements at the beginning of this document, the following must be carefully observed:

- The SensorBox must not be accessible to children
- Make sure the device is unplugged before opening
- There is a danger to life due to live connections
- Make sure all cables have adequate strain relief
- The device must be protected by an additional external all-pole circuit breaker installed in the building's electrical installation
- Make sure that the cable glands and seals in the cover fit correctly
- Make sure no splash water can reach the electronics
- The housing must be fixed to the wall using screws before the device is put into operation
- The SensorBox is wall-mounted using 4 screws on a suitable wall
- The holes are located at the corners of the housing and are suitable for screws up to 4 mm in diameter
- The mounting surface of the box is 300 x 230 mm
- The housing base of the SensorBox should rest evenly on the wall to avoid damage
- The distances between the holes can be found in the drawings in the *Dimensions* chapter
- A mounting height of 160 to 180 cm from the floor to the bottom edge of the housing is recommended
- To ensure protection against the ingress of water, the SensorBox should be mounted with the cable outlets facing downwards

11 Software

11.1 OpenWrt

The operating system of the gateway family is based on OpenWrt. OpenWrt is a vastly customisable and expandable GNU/Linux distribution created by network technology enthusiasts and professionals for like-minded people. Unleashing the full potential of OpenWrt requires the user to be willing to experiment and research as it is hardly possible to create a full-fledged topical guide.

Therefore, the configuration and application examples presented in this guide, can be considered no more than an introduction created by best effort with no liability for damages or any warranties for accurateness and topicality whatsoever.

11.2 Use of open-source software

This product includes open-source software, which was partially developed by third parties and distributed with a permissive license. The use of this software is royalty-free under the terms of the respective license. In case of a contradiction between our terms and the software license terms, the software license terms take priority as far as the software is affected.

The use of the open-source software is free of charge. We do not charge any usage fees or comparable fees for the open-source software contained in our products.

For retrieving a list of open-source software used in the product, please contact our support department (support@mc-technologies.com). Alternatively, a list of open-source software used can be found in the web interface in *System -> Software -> Installed*.

Customers may request the source code of software contained in our product, which license stipulates that the source code and/or modifications must be made available to the customer Examples of such licenses are the GNU General Public License (GPL), GNU Lesser General Public License (LGPL) and the Clarified Artistic License. We reserve the right to demand a compensation fee for the distribution costs of the source-code if applicable (e.g. postal fees and the cost of the medium).

11.3 Software development kit (SDK)

On request MC Technologies provides an SDK based on the OpenWrt SDK to customers for compiling and running custom software solutions on the device.

11.4 Liability for software

We however do not assume any warranty or liability for changes made to the software by the customer or third-parties or for the usage of the open-source software contained in our product in a way that does not comply with the intended use as described in the accompanying documentation or, if applicable, the contractually defined application purpose of the product.

This applies likewise to any use of the open-source software outside of our product.

12 Basic routines

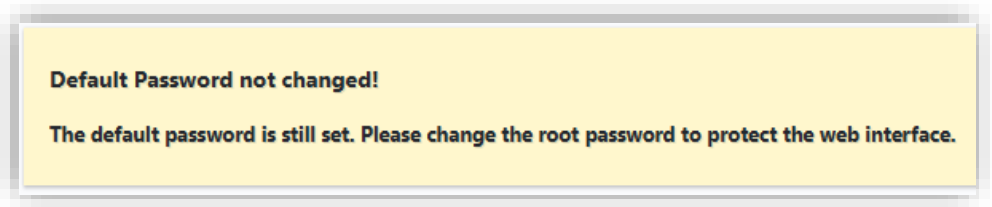
12.1 Accessing the web interface

The gateway can be configured using its integrated web interface. For accessing the web interface, connect your computer with one of the LAN interfaces of the gateway.

- If configured accordingly, the computer obtains an IP address automatically using DHCP
- On the computer, open up a web browser and navigate to *https://192.168.2.1*
- A login prompt shows up.
- The following default login credentials are required:
  - User: root
  - Password: Tech#5GR
- The default password must be changed immediately after the first login and before connecting the device to any network.
- For security reasons, the device must not be operated with default credentials.

12.2 Changing the password

After logging in for the first time, you must create a new password. Otherwise you will not be able to open any other pages.



12.3 Access via SSH

SSH can be used to access the Linux command line interface of the gateway. For convenient access, a dedicated terminal program like Putty is recommended.

Alternatively, Linux systems and recent Windows systems come with an SSH program built into their command shell. On Windows, the command shell can be opened up by pressing **Windows+r** and entering *cmd* in the Run prompt.
To establish an SSH connection execute:

ssh root@192.168.2.1

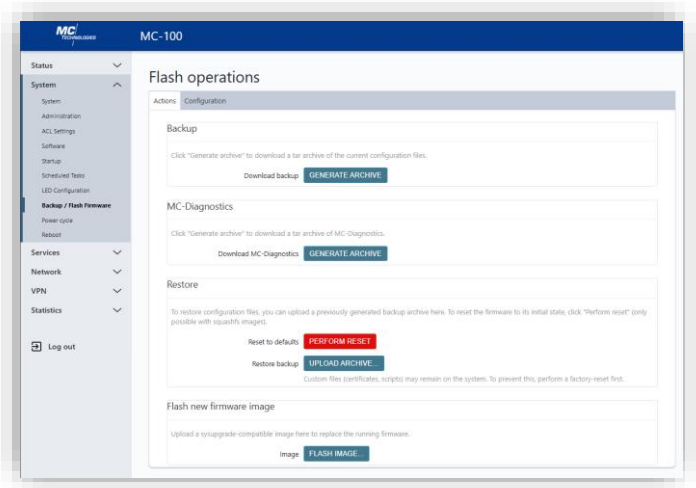
Where *root* is the username and *192.168.2.1* the IP address of the gateway. Provided a password was set, you will be prompted for the password, when the connection attempt succeeds.

12.4 Flash operations

In *System->Backup / Flash firmware* upgrades can be performed and configuration backups can be created and restored.

12.4.1 Configuration backup

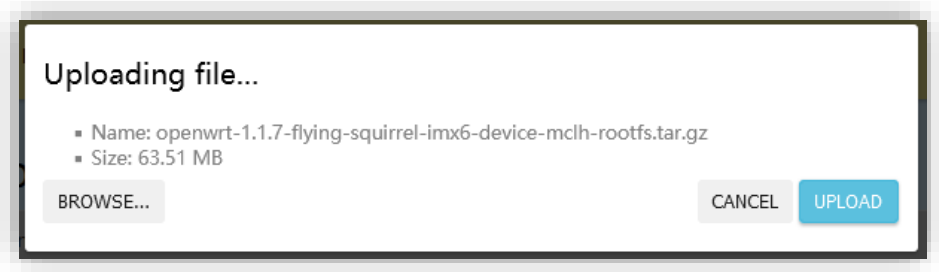
Backup archives containing the configuration files can be downloaded here. These archives can later be uploaded and restored or deployed to another device of the same kind with an equivalent firmware version. Backups must not be restored on different models or devices running different firmware versions, as it cannot be ensured, that the configuration format is identical. This could lead to untraceable errors, system instabilities and even security issues.



12.4.2 Firmware upgrade

The firmware upgrade functionality allows to upgrade the operating system base. This means not only software packages will be updated, but critical system components. Therefore, it is highly recommended, to check beforehand, if the upgrade process works in a simulated environment resembling the real-world installation and keeping the devices physically accessible for troubleshooting when performing the upgrade. This avoids outages and unnecessary service work. Keeping the devices up to date is essential for receiving the latest features, security, and stability fixes.

To upgrade the firmware, click *FLASH IMAGE...* Next click *BROWSE*, choose the firmware upgrade file, and then click *UPLOAD*. Uploading the firmware file may take a while. It is still safe to abort.

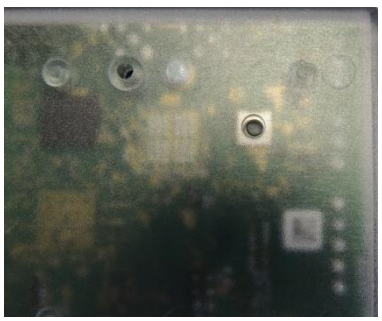


After the firmware has been uploaded, a new dialog will display the checksum of the file for confirmation. The upgrade process can be started by clicking the *Continue* button. Do not turn the gateway off while the upgrade is being performed. After a brief period, the device reboots and then executes additional migration routines, which can take up to 10 minutes. The upgrade is finished when the web interface is reachable again.

12.5 Factory reset

WARNING:
All files will be deleted, and user settings will be reset to factory defaults. Please ensure the device is physically accessible for troubleshooting.

- Locate the reset button on the back of the MC100
- Disconnect the power supply
- Use e.g. a paperclip to press and hold the reset button
- While keeping the button pressed, connect the power supply
- Release the button after the status LED rapidly blinked for about a second
- After about 10 seconds, the status LED will start blinking slowly
- The device is ready to be used when the LED is turned on steadily



13 Wired interfaces

This chapter aims to provide a basic understanding and offers recommendations for using the wired interfaces, especially the Linux-specific implementation. For model-specific descriptions of connectors (and pinouts of the base), have a look at chapter *Ports, display and operating elements.* Information on using the interfaces with Node-RED can be found in the corresponding *Node-RED* chapters.

Please note: Linux represents many interface devices by a “virtual” file structure. Interacting with these interfaces is as easy as reading or writing to any other files. In this chapter tables are given describing the file paths and the input and output format for interfaces to which this applies.

E.g. reading the value from digital input 1:

```
root@MC100:~# cat /sys/class/gpio/mc100:in1/value

0
```

The command outputs 0 which means the digital input is “low” in this case.

Likewise writing a 1 to the corresponding file of digital output 2, switches the output on.

```
root@MC100:~# echo 1 > /sys/class/gpio/mc100:out2/value
```

13.1 RS-232

While serial interfaces are also represented as files in Linux, they require special programs or libraries to be interacted with interactively and so specify setting such as baud rate (e.g. *picocom)*.

13.2 RS-485

The RS-485 interface */dev/ttymxc4* can be used like any other standard Linux serial device.

13.3 1-Wire

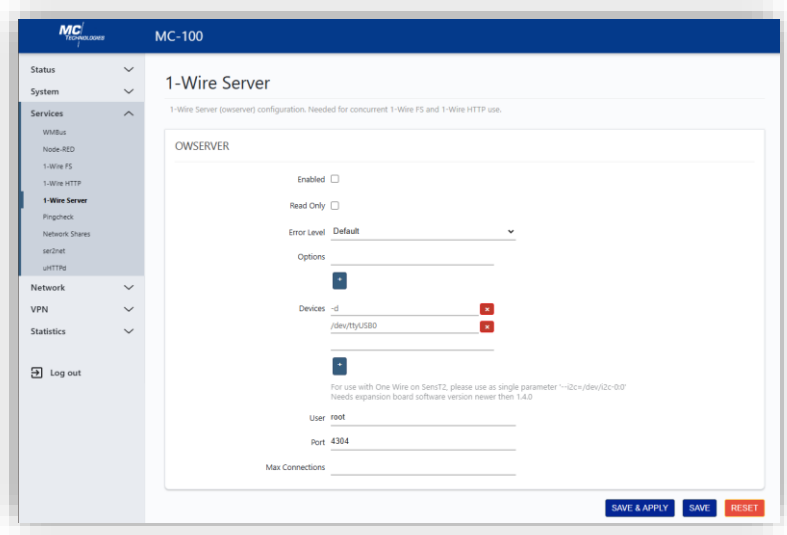
The 1-Wire driver is automatically loaded and started during the boot process.

For Linux users, the pre-installed OWFS 1-Wire file system is probably the most intuitive way to communicate with 1-Wire devices.

13.3.1 OWServer

The Owserver can be enabled using LuCI web Interface:

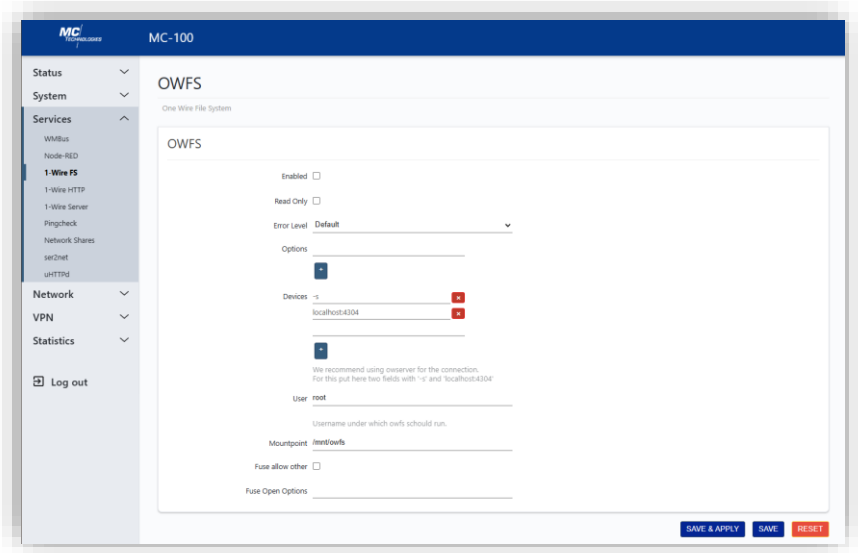
1. Navigate to *Services->1-Wire Server*
2. Check the *Enabled* checkbox
3. Set *Devices* to: -i2c=/dev/i2c-0:0
4. Change the *Port* as desired (default is 4304)
5. Click *SAVE&APPLY*



13.3.2 OWFS

OWFS can be Enabled using LuCI web Interface:

1. Navigate to Services->1-Wire FS
2. Check the Enabled checkbox
3. The mount point can be changed as desired (default is /mnt/owfs)
4. Click SAVE&APPLY



The 1-Wire file system abstraction can then be found in /mnt/owfs

A listing of the folder reveals the available devices as subfolders:

```
root@MC100:~#  ls /mnt/owfs/

10.0702A3030800/  alarm/           settings/        statistics/      system/

10.5B94A3030800/  bus.0/           simultaneous/    structure/       uncached/
```

The subfolders contain type-specific files to interact with the device, which is a temperature sensor in this example:

```
root@MC100:~# ls /mnt/owfs/10.0702A3030800/

address  crc8     family   latesttemp  power  r_id scratchpad   temphigh     type
alias    errata   id       locator     r_address    r_locator    temperature
templow
```

Reading the temperature is straightforward:

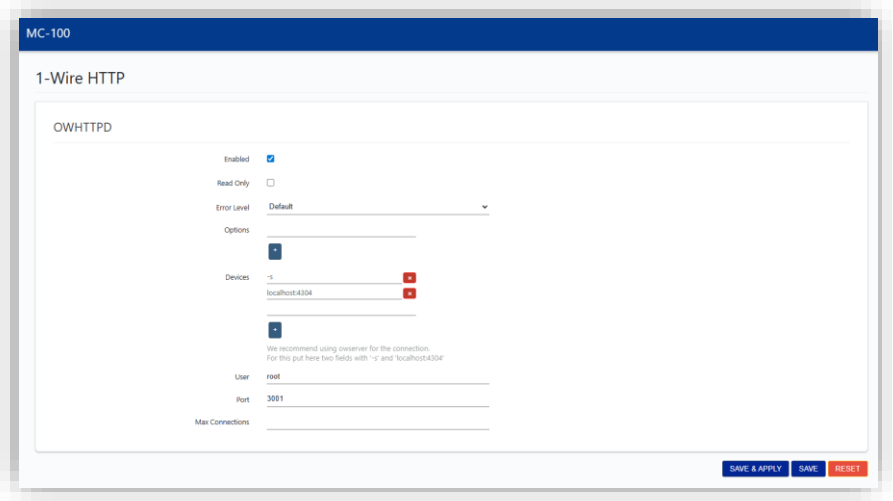
```
root@MC100:~# cat /mnt/owfs/10.0702A3030800/temperature

24.5625
```

13.3.3 Owhttpd

OWFS provides a small webserver that shows the Dallas/Maxim 1-Wire bus attached to a serial port. The main page shows the devices that have been found. It allows to view and change the properties of the devices.

The server can be enabled as follows:

1. Navigate to *Services->1-Wire HTTP*
2. Check the *Enabled* checkbox
3. The port can be changed as desired (default is 3001)
4. Click *SAVE&APPLY*



You can reach the owhttpd web interface using the same address as the standard web interface by specifying the port (e.g. 3001).

Example with default IP address: https://192.168.2.1:3001

13.4 M-Bus

The MC100 terminal/gateway can be equipped with an M-Bus module allowing to communicate with various kinds of sensors (heat cost allocators, electricity meters etc.).

The M-Bus interface */dev/ttymxc2* can be used like any other standard Linux serial device. The settings for M-Bus are 8 data bits, no parity and 1 stop bit. Most M-Bus devices communicate at a baud rate of 2400 baud.

Several programs based on libmbus for communication with M-Bus are pre-installed:

- mbus-serial-request-data-multipli-reply
- mbus-serial-request-data
- mbus-serial-scan
- mbus-serial-scan-secondary
- mbus-serial-set-address
- mbus-serial-select-secondary
- mbus-serial-switch-baudrate

Use the parameter -h to for a brief explanation of the programs and their parameters.

To list the connected devices, execute:

```
root@MC100:~# mbus-serial-scan -b 2400 /dev/ttymxc2
```

To request data from device 10, execute:

```
root@MC100:~# mbus-serial-request-data -b 2400 /dev/ttymxc2 10
```

13.5 CAN

The CAN interface Linux kernel stack is called SocketCAN. CAN devices are handled like network devices with special flags, constraints and usage requirements that must be observed.

A set of special programs called *can-utils* is pre-installed to allow convenient use of the CAN bus.

13.5.1 Activating the CAN interface

The CAN interface must be brought up before usage as any other network device. The following command illustrates this while setting the bitrate (speed of the CAN bus – 500 kbit/s in this example) and the error recovery time (time in ms after which a device restart is tried after a fault case):

```
root@MC100:~# ip link set ext-can1 up type can bitrate 500000 restart-ms 1000
```

To send a standard CAN frame with 0x101 as the ID and 0x41, 0x42, 0x43, 0x44 as a 4-byte payload, type:

```
cansend vcan0 101#41424344
```

To send an extended frame with 0xA1B2 as the ID, give the following command:

```
cansend vcan0 0000A1B2#3450
```

To capture all incoming can frames on vcan0 along with timestamps, type:

```
candump -t Absolute vcan0
```

Interface name can be any to capture all the interfaces, as follows:

```
candump -t A any
```

To store all captured frames in a log file, use the following command:

```
candump -l vcan0
```

13.6 Digital inputs

The MC100 base has two digital inputs. The inputs are active high.

Port	Parameter	Read	Write	Path
I1	1,0	x		/sys/class/gpio/mc100:in1/value
I2	1,0	x		/sys/class/gpio/mc100:in2/value

Example: Reading Input I1

Command: cat /sys/class/gpio/mc100:in1/value

Response: 1# or. 0#

13.7 Digital outputs

The MC100 base has two digital outputs. The outputs are solid state relays and switch against IGND.

Port	Parameter	Read	Write	Path
O1	1,0		x	echo 1 > /sys/class/gpio/mc100:out1/value
				echo 0 > /sys/class/gpio/mc100:out1/value
O2	1,0		x	echo 1 > /sys/class/gpio/mc100:out2/value
				echo 0 > /sys/class/gpio/mc100:out2/value

Example: Switch on O1

Command: echo 1 > /sys/mc100_gpios/OUT1

Response: #

13.8 MC100 GPIO wired interfaces

13.8.1 Digital inputs

Port	Description
DGND1	Digital Ground1, electrically isolated to all (D)GND
I1 bis I4	Digital inputs Input voltage: 0 to 30V Switching threshold: ~ 35.. 6V All input voltages with DGND1 as ground
VPU1	Not supported, please do not connect a signal
DGND2	Digital Ground 2, electrically isolated to all (D)GND

I1 bis I4	Digital inputs Input voltage: 0 to 30V Switching threshold: 36pprox.. 6V All input voltages with DGND2 as ground
VPU2	Not supported, please do not connect a signal

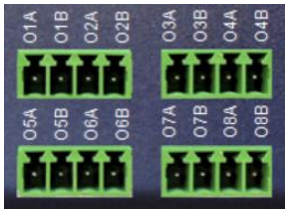
Port	Parameter	Read	Write	Path
I1	1,0	X		/sys/gpio_board/input/1
I2	1,0	X		/sys/gpio_board/input/2
I3	1,0	X		/sys/gpio_board/input/3
I4	1,0	X		/sys/gpio_board/input/4
I5	1,0	X		/sys/gpio_board/input/5
I6	1,0	X		/sys/gpio_board/input/6
I7	1,0	X		/sys/gpio_board/input/7
I8	1,0	X		/sys/gpio_board/input/8

Example: Read out Input I1

Command: cat /sys/gpio_board/input1/value

Response: e.g. 1# or 0#

13.8.2 Digital outputs



Like the basic digital outputs, the additional digital outputs of the MC100 GPIO are galvanically isolated solid-state relays. OxA is connected to one pin of the relay, OxB to the other pin. The maximum switching current is 300 mA (max. 30 V DC).

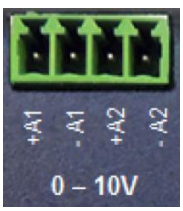
Port	Parameter	Read	Write	Path
O1A,O1B	1,0		X	/sys/gpio_board/output/1
O2A,O2B	1,0		X	/sys/gpio_board/output/2
O3A,O3B	1,0		X	/sys/gpio_board/output/3
O4A,O4B	1,0		X	/sys/gpio_board/output/4
O5A,O5B	1,0		X	/sys/gpio_board/output/5
O6A,O6B	1,0		X	/sys/gpio_board/output/6
O7A,O7B	1,0		X	/sys/gpio_board/output/7
O8A,O8B	1,0		X	/sys/gpio_board/output/8

Example: Switch on output O1A,O1B

Command: echo 1 > /sys/gpio_board/output/value

Response: #

13.8.3 Voltage inputs 0 – 10V



The input current at 10V is approx. 2 mA. The applied DC voltage must not exceed 10V.

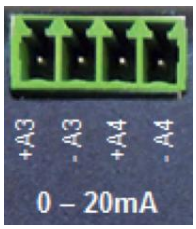
Port	Description
+A1	Positive connection Input 1
-A1	Negative connection Input 1
+A2	Positive connection Input 2
-A2	Negative connection Input 2

Port	Parameter	Read	Write	Path
+A1,-A1	Value	x		/sys/gpio_board/ voltage_in/1
+A2,-A2	Value	x		/sys/gpio_board/ voltage_in/2

Example: Read voltage at ADC input +A1,-A1
Command: cat /sys/gpio_board/ voltage_in1/value
Response: e.g. 6400

Note: Converting to volts
Formula: Voltage in Volt = value / 1000
Example: Value = 6400 corresponds to 6.4V

13.8.4 Current inputs 0 – 20 mA



A current of up to 20 mA from a constant current source can be fed into these inputs.

Port	Description
+A3	Positive connection Input 3
-A3	Negative connection Input 3
+A4	Negative connection Input 4
-A4	Negative connection Input 4

Port	Parameter	Read	Write	Path
+A3,-A3	Value	x		/sys/gpio_board/current_in/3
+A4,-A4	Value	x		/sys/gpio_board/current_in/4

Port	
Command	cat /sys/gpio_board/current_in1/value
Response	e.g. 2000

Note: Converting to mA

Formula: Current in mA = Value / 200
Example: Value = 2000 corresponds to 10 mA

13.8.5 PWM

The PWM (Pulse Width Modulation) output is open-drain. This means it connects to GND. For an active-high signal, a pull-up resistor is required.

Port	Description
GND	Ground
PW2	Open drain digital output.
PW1	Open drain digital output.
PU	Input for internal protection diode (freewheeling diode) for inductive loads. (positive supply voltage)

Port	Parameter	Read	Write	Path
PW2	Prescaler {0,1023}		x	/sys/gpio_board/pwm/prescalar
	Pulse {0,65535}			/sys/gpio_board/pwm/1
PW1	Prescaler {0,1023}		x	/sys/gpio_board/pwm/prescalar
	Pulse {0,65535}			/sys/gpio_board/pwm/2
	Period*{0,65535}		x	/sys/gpio_board/pwm/period

13.9 MC100 SensT2 wired interfaces

13.9.1 Digital inputs

Port	Description
DGND1	Digital Ground1, electrically isolated to all (D)GND
I1 bis I4	Digital inputs Input voltage: 0 to 30V Switching threshold: approx. 6V All input voltages with DGND1 as ground
VPU1	Not supported, please do not connect a signal

DGND2	Digital Ground 2, electrically isolated to all (D)GND
I1 bis I4	Digital inputs Input voltage: 0 to 30V Switching threshold: approx. 6V All input voltages with DGND2 as ground
VPU2	Not supported, please do not connect a signal

Port	Parameter	Read	Write	Path
I1	1,0	x		/sys/senst2_board/input/1
I2	1,0	x		/sys/senst2_board/input/2
I3	1,0	x		/sys/senst2_board/input/3
I4	1,0	x		/sys/senst2_board/input/4
I5	1,0	x		/sys/senst2_board/input/5
I6	1,0	x		/sys/senst2_board/input/6
I7	1,0	x		/sys/senst2_board/input/7
I8	1,0	x		/sys/senst2_board/input/8

Example: Read out input I1
Command: cat /sys/senst2_board/input1/value
Response: e.g. 1# or 0#

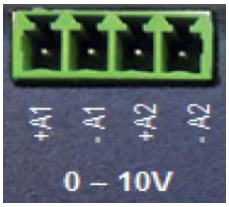
13.9.2 Digital outputs

Like the basic digital outputs, the additional digital outputs of the SensT2 are galvanically isolated solid-state relays. OxA is connected to one pin of the relay, OxB to the other pin. The maximum switching current is 300 mA (max. 30 V DC).

Port	Parameter	Read	Write	Path
O1A,O1B	1,0		X	/sys/senst2_board/output/1
O2A,O2B	1,0		X	/sys/senst2_board/output/2

Example: Switch on output O1A,O1B
Command: echo 1 > /sys/senst2_board/output/1
Response: #

13.9.3 Voltage inputs 0 – 10V



The applied DC voltage must not exceed 10V. The input draws a current of approximately 2 mA @ 10 V.

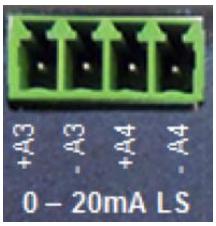
Port	Description
+A1	Positive connection input 1
-A1	Negative connection input 1
+A2	Positive connection input 2
-A2	Negative connection input 2

Port	Parameter	Read	Write	Path
+A1,-A1	Value	x		/sys/senst2_board/voltage_in/1
+A2,-A2	Value	x		/sys/senst2_board/voltage_in/2

Example: Read voltage at ADC input +A1,-A1
Command: cat /sys/senst2_board/voltage_in1/value
Response: e.g. 6400

Note: Conversion to volts
Formula: Voltage in Volt = value / 1000
Example: Value = 6400 corresponds to 6.4V

13.9.4 Current inputs 0 - 20 mA



The analog current inputs of the MC100 SensT2 have two operating modes:

1. Supply of a current - current from the sensor - sensor with own power supply
2. Low side shunt - with power supply for the sensor

Example: Supply of a current - current from the sensor

The connected sensor has its own power supply and a current output with a maximal current of 20 mA. The sensor is connected to -A3 or -A4 against ground.

Port	Description
-A3	Positive connection input 3, current output of the sensor
GND	Negative connection Input 3, sensor ground
-A4	Positive connection input 4, current output of the sensor
GND	Negative connection Input 4, sensor ground

Example: Low Side Shunt - power supply for the sensor

It is possible to simultaneously supply the sensor with power while measuring the current it draws over the same two wires. In this example the sensor is powered by the MC100 SensT2. A voltage is applied to the terminals +A3 or +A4. Connect the sensor to +A3 and A3 or +A4 and -A4.

Port	Description
+A3	Positive connection output 3, sensor supply, approx. 12V to 14V

-A3	Negative connection Input 3
+A4	Positive connection output 4, sensor supply, approx. 12V to 14V
-A4	Negative connection Input 4

Port	Parameter	Read	Write	Path
+A3,-A3	Value	x		/sys/senst2_board/current_in/3
+A4,-A4	Value	x		/sys/senst2_board/current_in/4

Example: Read current at ADC input +A3,-A3
Command: cat /sys/senst2_board/current_in1/value
Response: e.g. 1500

Note: Conversion to mA
Formula: Current in mA = Value * 20 / 3000
Example: Value = 1500 corresponds to 10mA

13.9.5 Current outputs 0 - 20 mA

Port	Parameter	Read	Write	Path
OI1	Value		x	/sys/senst2_board/current_out/1
OI2	Value		x	/sys/senst2_board/current_out/2

Note: Conversion from mA to value to be supplied
Formula: Value = Current in mA * 1000
Example: Current = 6mA Corresponds to Value = 6000

13.9.6 PT100 / PT1000 inputs

Port	Parameter	Read	Write	Path
RTD1	Value	x		/sys/senst2_board/rtd/1
RTD2	Value	x		/sys/senst2_board/rtd/2
RTD3	Value	x		/sys/senst2_board/rtd/3
RTD4	Value	x		/sys/senst2_board/rtd/4

Example: Read the value at RTD1
Command: cat /sys/senst2_board/rtd/1
Response: e.i. 100000

Conversion at port of a resistor in Ohm

Formula: Resistance in Ohm = Value / 100
Example: Value = 100000 corresponds 1000Ohm

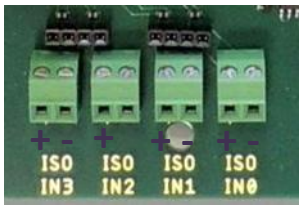
Conversion at port of a PT1000 temperature sensor to °Celsius

Formula: Temperature in Grad Celsius = (Value / 100 -1000) / 3,891
Example: Value = 112000 corresponds to +30,84°Celsius

Value = 100000 corresponds to 0°Celsius
Value = 90000 corresponds to -25,5°Celsius

13.10 MC100 SensorBox wired interfaces

13.10.1 Digital inputs



The galvanically isolated digital inputs can detect a DC voltage signal > 6V (active-high, max. 30 V DC) using optocouplers.

Port	Parameter	Read	Write	Path
IN0	1,0	x		/sys/sensor_carrier_board/inputs/isoInput0/value
IN1	1,0	x		/sys/sensor_carrier_board/inputs/isoInput1/value
IN2	1,0	x		/sys/sensor_carrier_board/inputs/isoInput2/value
IN3	1,0	x		/sys/sensor_carrier_board/inputs/isoInput3/value

Example: Read state of input IN0
Command: cat /sys/sensor_carrier_board/inputs/isoInput0/value
Response: e.g. 1# or 0#

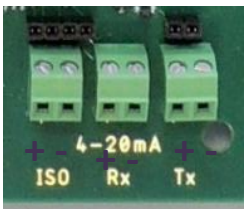
13.10.2 Digital outputs

Like the basic digital outputs, the additional digital outputs of the MC100 GPIO are galvanically isolated solid-state relays. The maximum switching current is 300 mA (max. 30 V DC).

Port	Parameter	Read	Write	Path
ISO OUT1	1,0		X	/sys/sensor_carrier_board/outputs/output1/value
ISO OUT2	1,0		X	/sys/sensor_carrier_board/outputs/output2/value

Example: Switch on output ISO OUT1
Command: echo 1 > /sys/sensor_carrier_board/outputs/output1/value
Response: #
Example: Value = 6400 corresponds to 6.4V

13.10.3 Current inputs 4 - 20 mA Rx / ISO



The analog current inputs of the MC100 SensorBox have two operating modes.

1. Supply of a current - current from the sensor - sensor with own power supply

2. Low side shunt - with power supply for the sensor

Example: Supply of a current - current from the sensor

The connected sensor has its own power supply and a current output with a current of 20 mA.

The sensor is connected to + and – of Rx.

Port	Description
+	Positive connection input, current output of the sensor
-	Negative connection Input, sensor ground

Example: Low Side Shunt - power supply for the sensor

It is possible both to supply the sensor with power and to measure the current it draws over the same two wires. The sensor will be powered from the MC100 SensorBox. A voltage is applied to the terminals + and – of 4-20 mA ISO for this purpose. Connect the sensor to + and – of 4-20 mA ISO.

Port	Parameter	Read	Write	Path
+,- Rx	Value	x		/sys/sensor_carrier_board/adcinputs/adcinput1/value
+,- ISO	Value	x		/sys/sensor_carrier_board/adcinputs/adcinput0/value

Example: Read current at ADC input +,- Rx
Command: cat /sys/sensor_carrier_board/adcinputs/adcinput1/value
Response: e.g. 1500

Note: Conversion to mA
Formula: Current in mA = Value * 20 / 3000
Example: Value = 1500 corresponds to 10mA

13.10.4 Current outputs 0 - 20 mA Tx

Port	Parameter	Read	Write	Path
+,- Tx	Value		x	/sys/sensor_carrier_board/adcoutputs/adcoutput/value

Example: Output of 6mA at OI1
Command: echo 6000 > /sys/sensor_carrier_board/adcoutputs/adcoutput/value
Response: none

13.10.5 RTD inputs

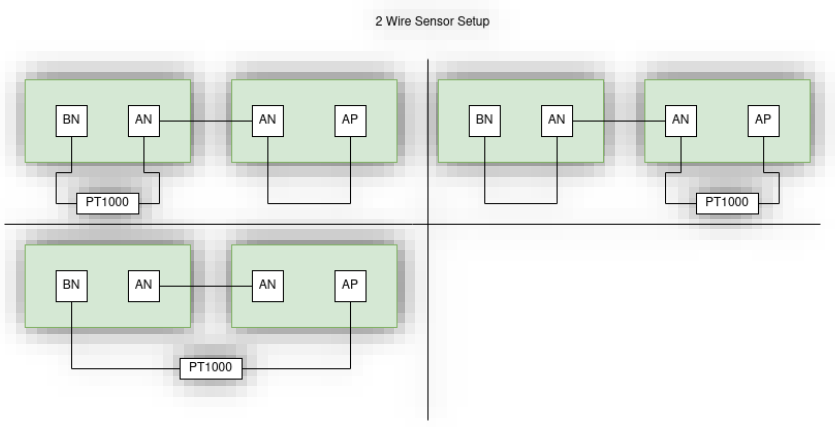
The RTD inputs can measure resistances for e.g. determining temperatures with PT100 or PT1000 sensors. There are different types of RTDs, including 2-wire, 3-wire, and 4-wire RTDs.

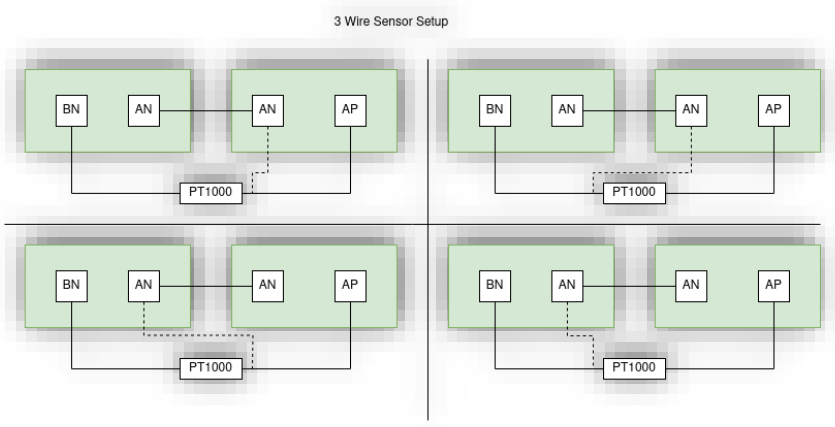
The selection of which type to use depends on the specific temperature range and precision requirements. With a 2-wire RTD, the resistance of the sensor and the lead wires sum up. Especially with a long cable and high temperature fluctuations the error may be critical. As a result, 3-wire RTDs have become the industry standard when greater accuracy is needed. Assuming all leads have the same length, material and diameter, a third lead allows to cancel out the wire resistance arithmetically.

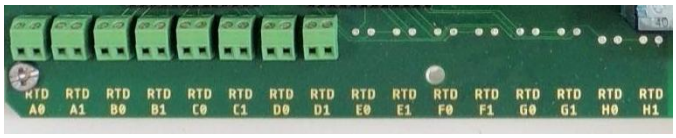
Please note: The SensorBox can be used for measuring 2- and 3-wire RTDs, but not 4-wire RTDs.

On the following page all possible combinations to connect 2-wire and 3-wire to the SensorBox are illustrated.

As can be seen from the illustration below, the RTD inputs only work in pairs (A0A1, B0B1, …). If e.g. A0 is to be used and A1 not, then A1 needs to be bridged using a short piece of wire.







Port	Parameter	Read	Write	Path
RTD A0	Value	x		/sys/sensor_carrier_board/rtd/rtd0/value
RTD A1	Value	x		/sys/sensor_carrier_board/rtd/rtd1/value
RTD B0	Value	x		/sys/sensor_carrier_board/rtd/rtd2/value
RTD B1	Value	x		/sys/sensor_carrier_board/rtd/rtd3/value
RTD C0	Value	x		/sys/sensor_carrier_board/rtd/rtd4/value
RTD C1	Value	x		/sys/sensor_carrier_board/rtd/rtd5/value
RTD D0	Value	x		/sys/sensor_carrier_board/rtd/rtd6/value
RTD D1	Value	x		/sys/sensor_carrier_board/rtd/rtd7/value
RTD E0	Value	x		/sys/sensor_carrier_board/rtd/rtd8/value
RTD E1	Value	x		/sys/sensor_carrier_board/rtd/rtd9/value
RTD F0	Value	x		/sys/sensor_carrier_board/rtd/rtd10/value
RTD F1	Value	x		/sys/sensor_carrier_board/rtd/rtd11/value
RTD G0	Value	x		/sys/sensor_carrier_board/rtd/rtd12/value
RTD G1	Value	x		/sys/sensor_carrier_board/rtd/rtd13/value
RTD H0	Value	x		/sys/sensor_carrier_board/rtd/rtd14/value
RTD H1	Value	x		/sys/sensor_carrier_board/rtd/rtd15/value

Example: Read the value at RTD A0
Command: cat /sys/sensor_carrier_board/rtd/rtd0/value

Conversion at port of a resistor in Ohm

Formula: Resistance in Ohm = Value / 100
Example: Value = 100000 corresponds 1000Ohm

Conversion at port of a PT1000 temperature sensor to °Celsius

Formula: Temperature in degrees Celsius = (Value / 100 -1000) / 3,891
Example: Value = 112000 corresponds to +30,84°Celsius
Value = 100000 corresponds to 0°Celsius
Value = 90000 corresponds to -25,5°Celsius

13.10.6 AC OUT relay

Port	Description
NC	Normally Close to COM
COM	Switch
NO	Normally Open to COM

Port	Parameter	Read	Write	Path
COM, NO	1,0		x	/sys/sensor_carrier_board/ac/acout/value

Example: Switch relay on
Command: echo 1 > /sys/sensor_carrier_board/ac/acout/value
Response: #

13.10.7 AC IN relay

This relay can be switched by connecting 230 V AC to the terminal blocks. The state of the relay can be detected like a digital input.

Port	Parameter	Read	Write	Path
AC IN	1,0		X	/sys/sensor_carrier_board/ac/acin/value

Example: Read out AC input status
Command: cat /sys/sensor_carrier_board/ac/acin/value
Response: 1,0

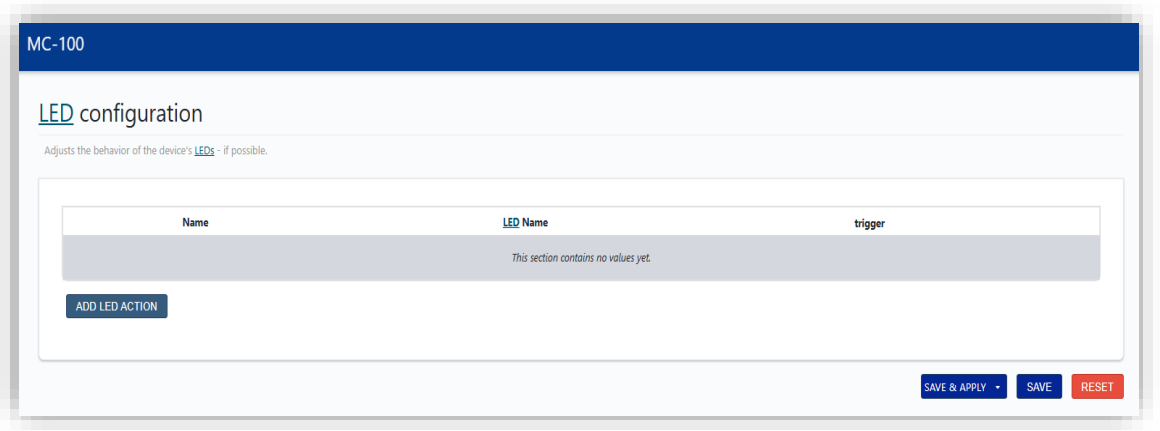
13.11 LEDs

MC100 have three LEDs which can be changed in their functionality: info, status and warn.

The signal triggers and blinking patterns of the LEDs can be customized in the web interface.

The settings can be found in *System->LED Configuration*.

Click *ADD LED ACTION* to create a new customized LED definition.



Options:

Name: Name of the LED configuration.

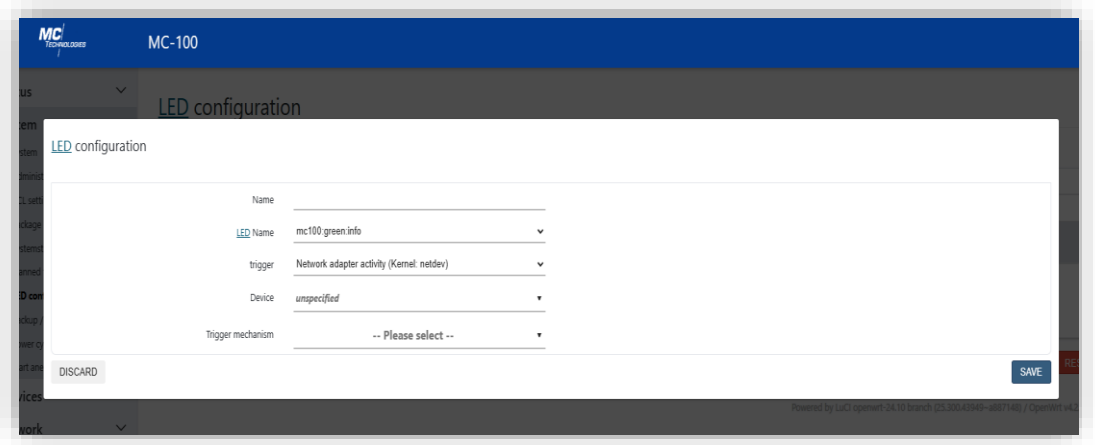
LED Name: Colour and name of the LED.

Default state of the LED: On/Off.

Trigger: One of the various triggers can be assigned to an LED to changes its states.

Possible values:

Trigger type	Description
heartbeat	Simulating actual heart beats
Always on	LED always stays on
Always off	LED always stays off
Custom flash interval	Blinking according to predefined timer pattern
netdev	Flashes according to link status and send/receive activity



The *Name* of the LED definition can be chosen arbitrarily but choosing the *LED Name* in the dropdown field is mandatory. The *Default state* defines whether the LED should be on or off initially before any trigger signal changes its state.
What signal source the LED blinking pattern is controlled by can be chosen using the *Trigger* dropdown field.

LED	LED Name
INFO	mclx:orange:info
STATUS	mclx:orange:status
WARN	mclx:red:warn
1	mclx-cb:green:led1
2	mclx-cb:red:led2

Click *SAVE*, then *SAVE & APPLY* for the LED configuration to come into effect.

14 Wireless interfaces

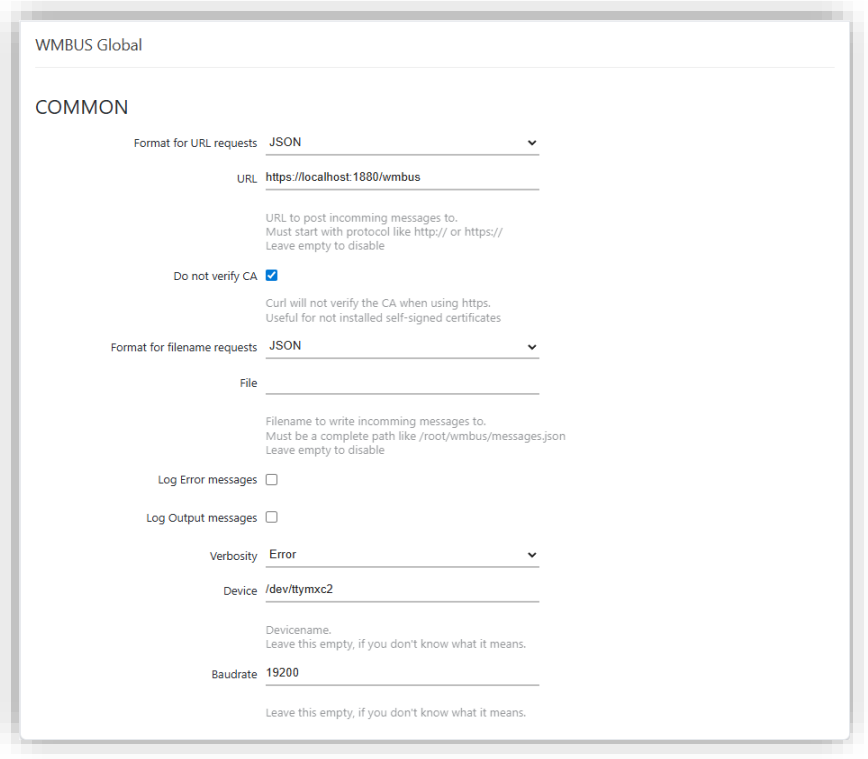
14.1 wM-Bus

The MC100 Terminal/Gateway can be supplied with an expansion board for Wireless M-Bus.

14.1.1 Enabling mcwmbus

Before using wM-Bus, it is required to enable the mcwmbus service. Find the corresponding entry in *System->Startup* and click the *DISABLED* button to enable automatic start up on boot if the service has not been enabled, yet. Then click the *START* button to start the service manually or reboot the device.

In the *Services* tab, click wM-Bus.



1. Click Enable
2. Choose the output format the POST request under *Format for URL requests*
3. Type desired URL. (If not needed leave it empty).
4. Choose the Output format to use to save output data as file under Format for file name requests

5. Enter the desired path and file name (optional)
6. Click *SAVE&APPLY*

14.1.2 Installing mcwmbus

To interact with the wM-Bus extension, the mcwmbus command line tool is needed.

14.1.3 Basic functionality

The tool supports the -h parameter to print information about its usage.

```
Usage: mcwmbus [options]

Copyright (C) 2020 MC-Technologies GmbH

Options:

-h        Show this help message and exit.

-V        Show version information and exit.

-v        Print verbose debug information

-d        DEVICE tty device (default: /dev/ttymxc2)

-b        BAUD Baud rate for communication (default: 19200)

-f        FILENAME Write data to file.

-u        URL Send Data via POST Request to URL

-c        Print Data on commandline

-F        FORMAT Data format for files (default: hex)
```

The output might look like this:

14.1.4 Output formats

The tool supports different output formats, which can be individually set for each output channel (URL, FILE, Command line)**.**

Hexadecimal:

HEX produces the message in hexadecimal. One message per line.

Example of hexadecimal output:

```
1644AF4C02000041011B7A980000000266E8000266E900
```

JSON

JSON interprets the message and gives as much information as possible. It also contains the raw message as a hexadecimal string.

Example of JSON output:


```
{
 "SlaveInformation": {
  "Id": 41000002,
  "Manufacturer": "SEO",
  "Version": 1,
  "ProductName": "Senseco Wireless M¬Bus 2 NTC Temperature
  Sensor",
  "Medium": "Ambient Sensor",
  "AccessNumber": 152,
  "Status": "00",
  "Signature": "0000"
 },
 "DataRecords": [
  {
   "id": 0,
   "Function": "Instantaneous value",
   "StorageNumber": 0,
   "VIF": 102,
   "VIFE": 0,
   "Unit": "External temperature (1e-1 deg C)",
   "Value": "232",
   "Timestamp": "2020-06-09T07:51:08Z"
  },
  {
  "id": 1,
   "Function": "Instantaneous value",
   "StorageNumber": 0,
   "VIF": 102,
   "VIFE": 0,
   "Unit": "External temperature (1e-1 deg C)",
   "Value": "233",
   "Timestamp": "2020-06-09T07:51:08Z"
   }
 ],
 "RawMessage": "1644AF4C02000041011B7A980000000266E8000266E900"
}
```

XML

XML output interprets the message and produces an XML output.
Example on xml output:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<MBusData>
  <SlaveInformation>
        <Id>41000002</Id>
        <Manufacturer>SEO</Manufacturer>
        <Version>1</Version>
        <ProductName>Sensoco Wireless M¬Bus 2 NTC Temperature
Sensor</ProductName>
        <Medium>Ambient Sensor</Medium>
        <AccessNumber>157</AccessNumber>
        <Status>00</Status>
        <Signature>0000</Signature>
  </SlaveInformation>

  <DataRecord id="0">
        <Function>Instantaneous value</Function>
        <StorageNumber>0</StorageNumber>
        <Unit>External temperature (1e-1 deg C)</Unit>
        <Value>232</Value>
        <Timestamp>2020-06-09T07:52:03Z</Timestamp>
  </DataRecord>

  <DataRecord id="1">
        <Function>Instantaneous value</Function>
        <StorageNumber>0</StorageNumber>
        <Unit>External temperature (1e-1 deg C)</Unit>
        <Value>233</Value>
        <Timestamp>2020-06-09T07:52:03Z</Timestamp>
  </DataRecord>
</MBusData>
```

json_hex

It is also possible to get the data in JSON format that is easier to read than pure HEX:


```
{"timestamp":"2020-06-09T09:53:47.295+0200","LEN": "22","C":
"44","MAN": "4caf","UID": "41000002","VER": "01","DEV": "1b","CI":
"7a","COUNTER": "a7","STATUS": "00","ENCRYPTION": "0000","DATA":
"0266e8000266e800"}
```


14.1.5 Posting to a REST API

It is possible to send the messages to a REST-API, by passing the command line „parameter –u“. For example, this can be used to send messages to the internal Node-RED server on the mc100:

```
root@   mcwmbus -u http://localhost:1880/wmbus
```

Or it can be used in combination with the integrated modem, to send messages to a server running in the cloud to have real time data available on your workstation.

14.1.6 Writing to the file system

When writing data to disk, it is possible to specify parts of the path dynamically by using magic strings:

```
%M - Manufacturer ID
%D - Device Type / Medium
%U - Ident Nr.
%V – Version
```

Example:

The following file-tree has been created by:

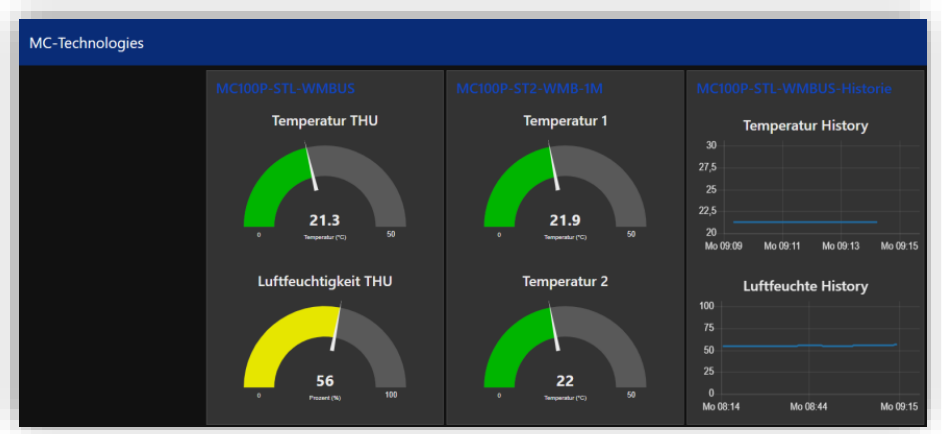
```
root@MC100:~# mcwmbus –f wmbus_messages/%M/%U.json
```

It allows tp preselect the manufacturer and device ID using the file path resulting in the following file system structure:

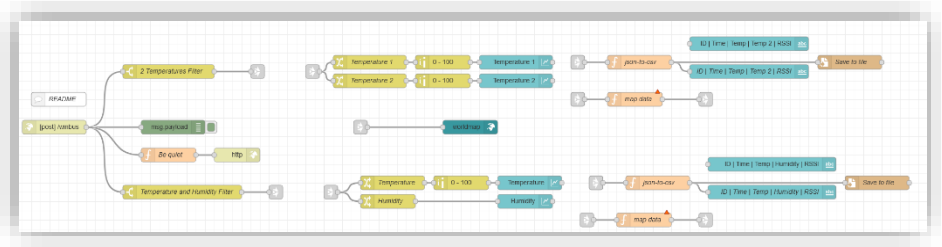
```
wmbus_messages
+-- 18c4
|   +-- 17949.json
+-- 25c5
|   +-- 33100007.json
+-- 4caf
|   +-- 30000007.json
|   +-- 41000002.json
```

14.1.7 Viewing live information in Node-RED

The example flow outlined in this chapter utilizes a REST-server and displays the information using the dashboard plugin. The result looks like this:



Node Red flow example:



Command:

```
root@MC100:~# mcwmbus -u http://localhost:1880/wmbus -c -C json
```

14.1.8 Aggregate data for 1 h, 6 h, 1 day and send via FTP/SCP

Collect messages for the whole day as interpreted JSON, but only send messages from manufacturer SEO to a server at the end of the day in a zip file.

```
while true;
do
mcwmbus -F json -f "wmbus_messages/%M.json" &
sleep 86400 # = 60*60*24 = 24 hours
killall mcwmbus
zip SEO.zip wmbus_messages/4caf.json
scp SEO.zip 192.168.1.1:/data/SEO-`date "+%Y-%m-%d"`.zip
rm SEO.zip
done
```

14.1.9 Troubleshooting

Please execute “mcwmbus –vV” and sent the output together with your error description and any error message to support@mc-technologies.net

It will be helpful if you run the command that produced the error with “-vvvvvv” to maximize the debug verbosity.

15 GNSS satellite navigation (GPS)

15.1 Enabling GNSS on startup

To activate the modem's GNSS and NMEA interface on startup, execute the command:

```
mcinfo -c "AT+QGPSCFG=\"autogps\",1"
```

To enable the widely used location daemon *GPSD* on startup, execute:

```
/etc/init.d/gpsd enable
```

Note: A reboot and eventually a power-cycle is required for the changes to come into effect.

16 Communication protocols

Modbus messaging protocol is used to establish client-server (master-slave) communication between devices. MC100 can either be used as master (server) or slave devices (client). As a master the MC100 can serve up to 247 slave devices. The MC100 was tested to query data using Modbus RTU trouble free in frequencies between 20 to 40 Hz.

16.1 Modbus master command line tool

16.1.1 Command line usage

- Read register 0 on slave 1:

```
root@MC100:~# mcmodbus -a 0
```

- Print debug information during execution:

```
root@MC100:~# mcmodbus -v -a 0
```

```
root@MC100:~# mcmodbus -vv -a 0
```

- Show help message:

```
root@MC100:~# mcmodbus –h
```

- Read register 0 on slave 17:

```
root@MC100:~# mcmodbus -s 17
```

- Set the output of slave 17 for the I/O Pins 4,5,6 to 1 0 1:

```
root@MC100:~# mcmodbus -o wb - a 4 -s 17 1 0 1
```

- Use a specific serial device with a baud rate of 115200:

```
root@MC100:~#mcmodbus -d /dev/ttyUSB10 -b 115200
```

- Set digital output at address 0x34 to ON:

```
root@MC100:~# mcmodbus -o wib -a 0x34 1

root@MC100:~# mcmodbus -o wib -a 064 1

root@MC100:~# mcmodbus -o wib -a 52 1
```

- Read analog input at address 0x20 and 0x21:

```
root@MC100:~# mcmodbus -o rir -a 0x20 -n 2
```

- Set register 8 to 0x4563:

```
root@MC100:~# mcmodbus -o wr -a 0x08 0x4563

root@MC100:~# mcmodbus -a 8 -o wr 17763
```

16.2 Modbus slave command line tool

```
Usage: mcmodbus-slave [OPTIONS]
Options:
  -h,--help              Print this help message and exit
  -c,--config-file TEXT  Json config for the address mappings. Default:
./mappings.json
  -d,--device-file TEXT  Serial device for modbus RTU
  -p,--port UINT         Port for modbus TCP.
  -b,--baud-rate UINT    Baud rate for the serial device. Default: 115200
  -v,--verbose           Activate debug input
```

16.2.1 Using Modbus RTU

Command: mcmodbus-slave -d <Serial-port-Device-file> -b <baud-rate> -c <JSON config>
Example (RS-485):

```
root@MC100:~# mcmodbus-slave -d /dev/ttymxc4 -b 115200
```

16.2.2 Using Modbus TCP

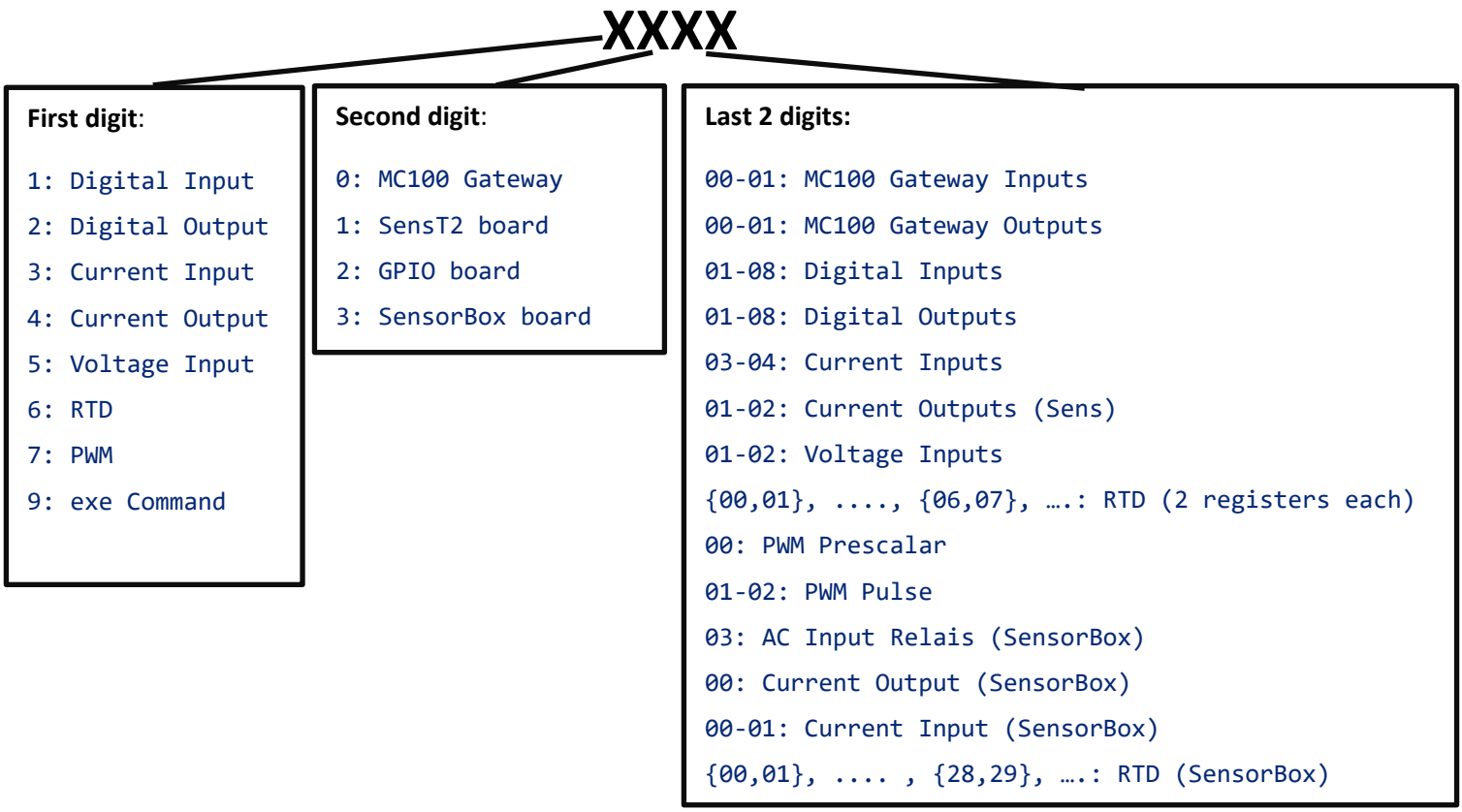
Command: mcmodbus-slave -p <port> -b <baud-rate> -c <Json Config>

For example:

```
root@MC100:~# mcmodbus-slave -p 502
```

16.2.3 MC100 default JSON mapping.

MC100 Modbus addressing map:



Example: The address of digital input 4 on MC100 GPIO is: 1204
The address of RTD 2 on MC100 SensT2 is: {6104,6105}

16.2.4 JSON configuration file

Example:

```
{
  "type": "file",
  "address": 4000,
  "num_addresses":  1,
  "register": true,
  "output": true,
  "filename": " /sys/sensor_carrier_board/adcoutputs/adcoutput/value",
  "factor": 1000,
  "isfloat": true,
  "MinValue": 4000,
  "MaxValue": 20000
},
```

Modbus data is most often read and written as "registers" which are 16-bit pieces of data. Consequently a 32-bit integer is usually represented using two, a 64 bit integer as four consecutive registers.

type	Type of mapping used. file: Contents of a file are mapped. exe: Execute a command
address	Modbus address used to be called by master device.
num_addresses	Number of registers used to map the file. [Default 1] 1 : 16-bit integer 2 : 32-bit integer
register	True: read / write a 16-bit integer False: read / write 1 or 0 [Default]
Output	True: write False: Read [Default]
filename	Path to the Input / Output file.
isfloat	True, if the input/output is float. [Default False]
factor	Factor used to transform float to integer. [Default 1]
MinValue	Minimum input value
MaxValue	Maximum input value
Command	Command to be executed. (Only when **Type** is „exe“)

17 Network interface configuration

A listing of the network interfaces can be found in *Network->Interfaces*.

Network interfaces in OpenWrt terminology are virtual in the sense that they represent a set of configuration options like the interfacing protocol, IP address, firewall zone etc. linked to a network device.

Network devices in OpenWrt terminology are what is traditionally understood as network interfaces in UNIX terms like hardware ethernet interfaces, virtual bridges etc.

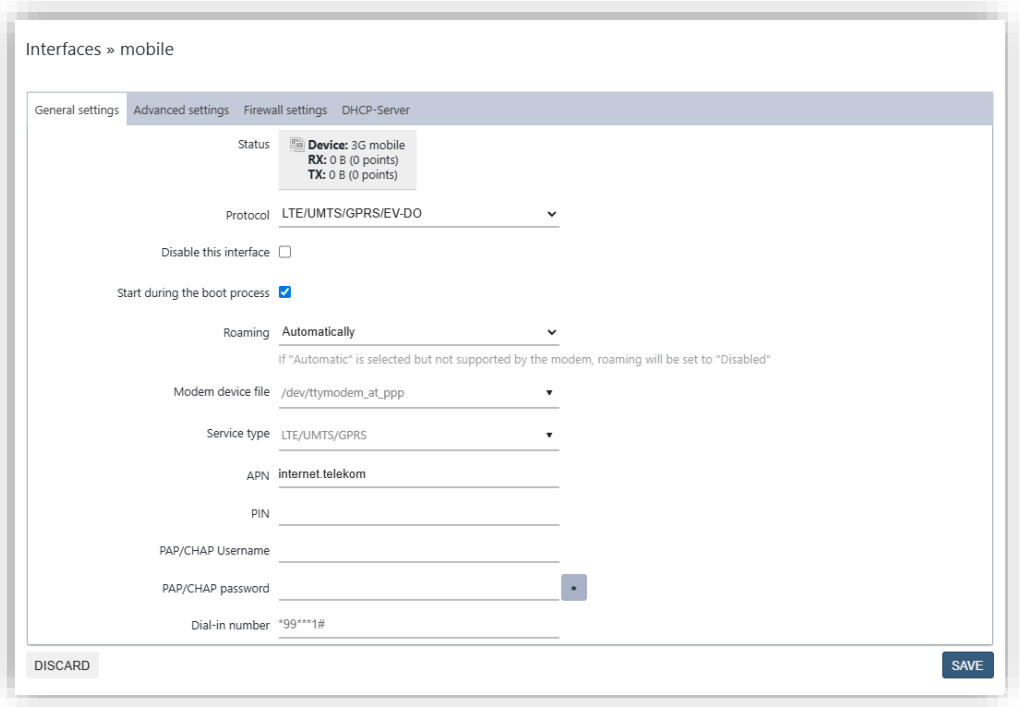
Example: In factory default state the *lan* interface is configured as an interface offering a DHCP server on the *br-lan* network device. *br-lan* is in fact a virtual bridge containing only the hardware interface **eth0** of the MC100.

17.1 Cellular connection setup

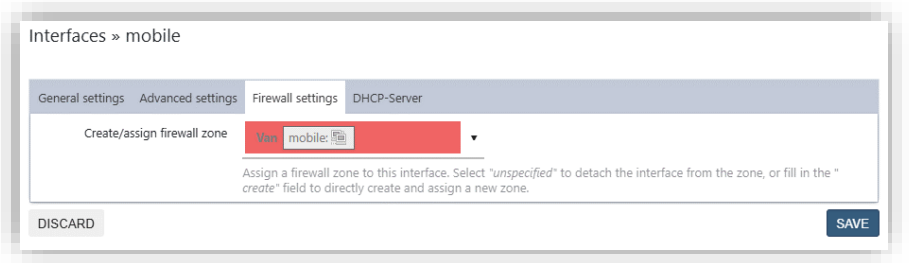
A mobile WAN interface configuration is present in factory default state, which should only require small provider-dependent adjustments to establish a mobile broadband connection.

Click *EDIT* next to the mobile interface.

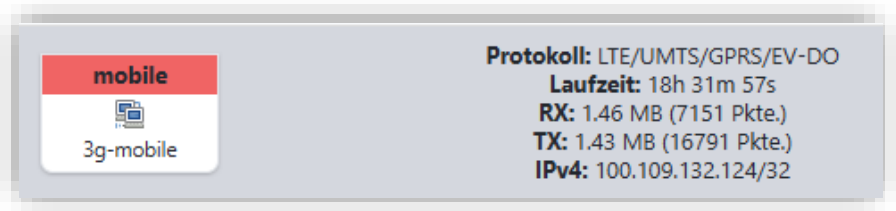
Access configuration details like Dial number, APN, username, and password need to be obtained from the mobile carrier. Many providers do not require authentication credentials in which case the username and password fields can be left empty. Enter the PIN number of your SIM card. Leave the field empty in case no PIN is set.



Switch to the *Firewall Settings* tab to ensure the mobile interface is added to the *wan* firewall zone.



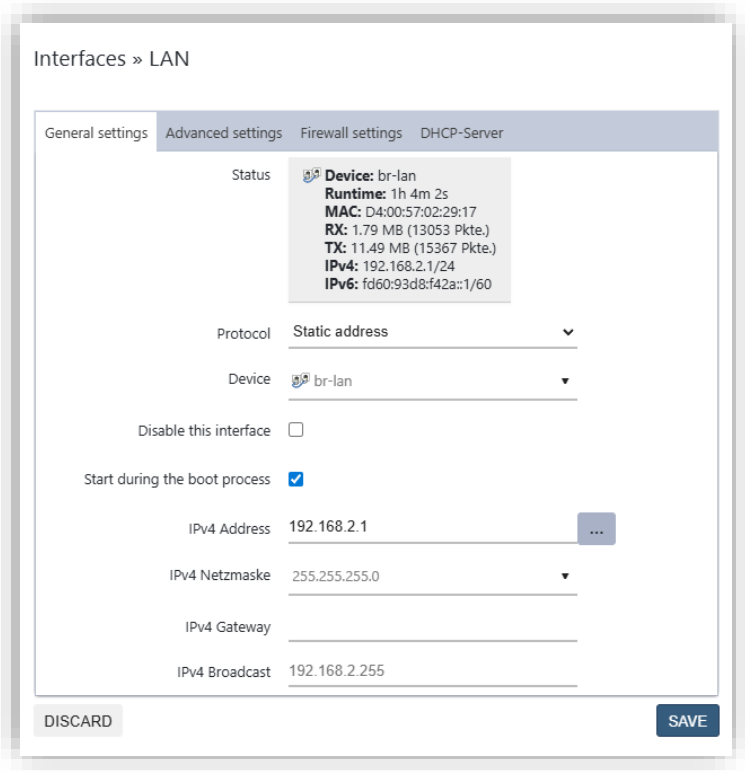
After clicking *SAVE & APPLY,* it is recommended to reboot or briefly disconnect the gateway to ensure the modem is properly reinitialized.



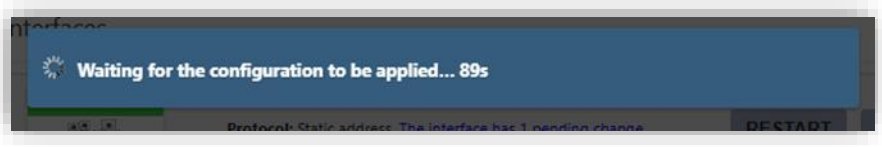
After the start-up, log-in to the web interface again. If the connection was established, the uptime, transmitted and received data statistics should be displayed in Network->Interfaces in the mobile interface entry's box.

17.2 Changing the LAN IP address

Navigate to *Network->Interfaces* and click *EDIT* next to the LAN interface. Change the IPv4 address, and netmask as required. In case a DHCP server is being provided by another device in the LAN network, the DHCP server must be turned off in the *DHCP Server* tab to avoid conflicts.
Click *SAVE*, but do not apply the changes, yet.



After applying the changes, a countdown will start. If this countdown elapses, before you were able to access the web interface using the new IP address, the gateway will revert the changes. This is a countermeasure against accidentally locking yourself out from the system. Once you are prepared for accessing the web interface using the new IP address, proceed by clicking *SAVE & APPLY*.



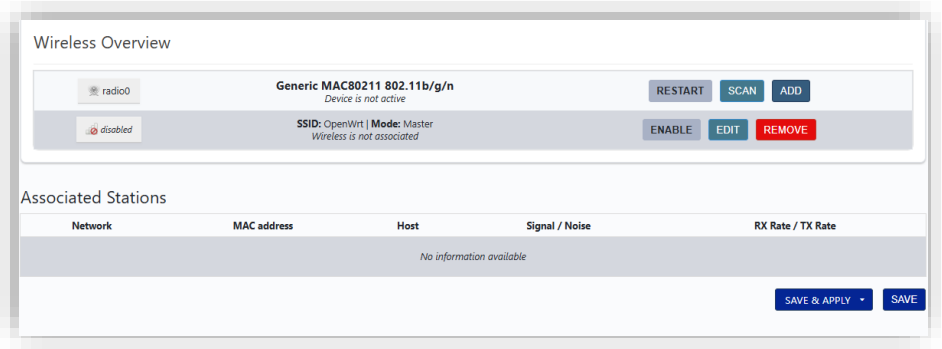
17.3 WLAN setup

WLAN is an optional feature integrated only in some models of the MC100 family. Please ensure the gateway in question has a corresponding SMA connector.

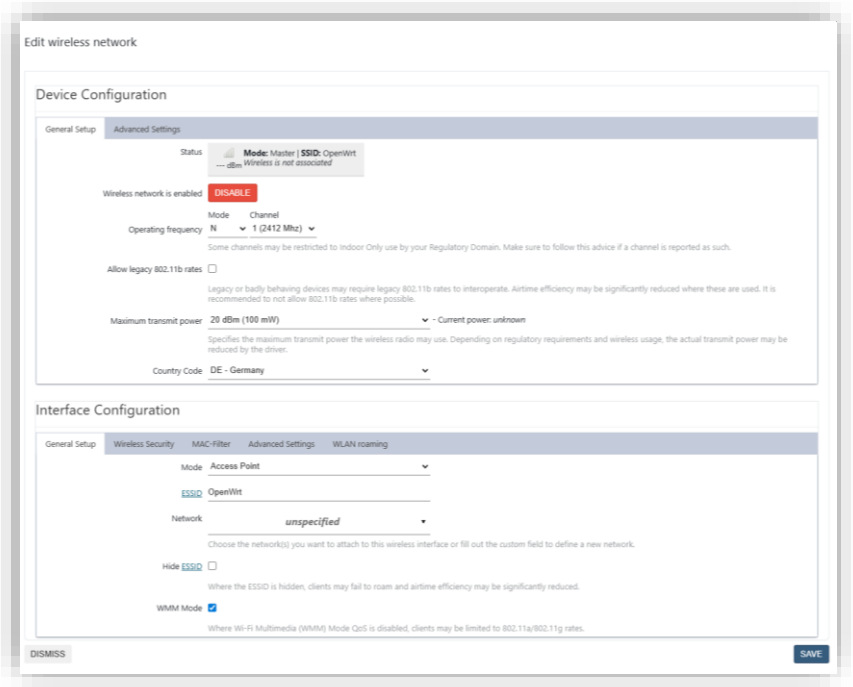
17.3.1 Access point mode (AP)

The gateway can be used as an access point for other devices to wirelessly connect to it.

Navigate to *Network->Wireless* and click *ADD* to add a virtual WLAN device to *radio0.*



Set *Network* to *LAN*, and set a name for the WLAN network in the *ESSID* field.
Click *Wireless Security* to set the type of encryption (WPA2 or better recommended).



Click *SAVE*, then *SAVE&APPLY*.
After a moment, you should be able to connect to the WLAN network and access the web interface.

17.3.2 Client mode (STA)

The gateway can be used as a WLAN client to connect to an access point. Navigate to *Network->Wireless* and click *SCAN* in the *Wireless Overview*. Choose the network to connect to and click *JOIN*.

Enter the WLAN password in the *WPA passphrase* field.
Click *SUBMIT*, then *SAVE&APPLY*.
The WLAN connection now serves as a WAN connection.

18 Firewall

18.1 Introduction

OpenWrt's firewall configuration is as powerful as it is complex. Understanding the concepts might be overwhelming for beginners, but rewards with an ability to maintain a more sophisticated network setup than other firewall solutions.

From an abstract view, the firewall is a set of mechanisms to regulate the exchange of network traffic between the host (the gateway), different networks like a DMZ (Demilitarized zone), proxy, VPN, WAN or LAN and their corresponding network interfaces. It does usually not influence routing decisions directly nor directly restricts any programs or services running on the gateway. More precisely, the firewall can be thought of as a mechanism filtering, marking and manipulating headers of packets travelling back and forth between applications or network interfaces and the operating system's routing mechanism. This journey of a packet continues until it has been rejected, silently dropped or delivered to its destination network interface or application socket.

This capability to inspect, filter and manipulate packets is steered by rule-based policies for handling the forwarding of packets between different network interfaces and the host. It allows to implement security measures for e.g. defining access policies to restrict access from one network to the other for specific IP address ranges and services (ports).

In combination with NAT (Network address translation) e.g. a limited IP address space can be shared and access to services running on devices in the LAN from the WAN network can be enabled by utilizing address mapping or port forwarding.

From an implementation perspective OpenWrt's firewall is basically a sophisticated stack of iptables rules that are being generated from an abstraction layer called fw3. Fw3 groups network interfaces into so-called *Zones.* This approach allows to keep an overview in complex setups, as the user does not need to create redundant iptables rules, which would otherwise be hardly manageable. The iptables rulesets generated by fw3 are intuitively understood. The interested user may want to inspect them by executing *iptables-save or fw3 show* in an SSH session.

Understanding the path, a data packet travels in an operating system's network stack, is very complicated. It is e.g. important, to keep in mind, how different kinds of data packets are handled - e.g. depending on whether they originate from the host itself, meaning they originate from an application or service running on the gateway like a wget command or a web server, or whether they originate from a network interface. Such packets do not necessarily follow the same rules a packet received on a network interface does. Only understanding the whole network stack, allows to be confident in a firewall. The comparison falls short but think of the firewall as a postal worker checking the addresses, the stamp, relabelling the address for rerouting and so forth and of the routing mechanism as a mail carrier who just delivers to the address on the label. You are the logistics engineer. To be up to the job, you need to know the whole process. It is unavoidable to read and understand the iptables documentation at some point:

https://www.frozentux.net/iptables-tutorial/iptables-tutorial.html

18.2 Overview

The firewall configuration has five subsections: General Settings, Port Forwards, Traffic Rules, NAT rules and Custom Rules.

In the *General Settings* or *Zone settings* tab, basic configuration options and default packet handling rules can be set for the so-called Zone concept, which is outlined in the subsequent chapters.

Port Forwards, as the name suggests, allows to set traditional port forwarding rules between different Zones.

Traffic rules allow to add exceptions to the Zone rules and fine-granular filtering capabilities for matching traffic.

With *NAT rules*, address rewriting and connection tracking mechanisms can be implemented.

18.3 General Settings (Zone Settings)

The Zone concept groups network interfaces into so-called Zones. Every zone has a basic set of rule chains (Input, Output and Forward) shared among its network interfaces. These rule chains represent the policy of handling packets for that zone. The policy is enforced upon all traffic originating from or destined to network interfaces in that zone. On top of that, the forwarding of traffic between zones can be policed and restricted on a zone-to-zone basis and fine-tuned using port forwards, traffic rules and NAT rules.

Explaining the policing of traffic flows between the host and the network interfaces of a zone deserves special emphasis as will be shown. *Host* denotes the gateway and the sockets of the applications and services running on the gateway themselves. Rules police the flow and especially the forwarding of traffic between interfaces of a zone among each other and, on a higher level, between zones and other zones.

The default policy is applied when no exceptive rule matches and can either be to accept (Accept), silently drop (Drop) or drop and reply to the packet with an ICMP unreachable message (Reject).

The input and output options set the default policies for traffic destined to or originating from the host (e.g. a webserver application).

The forward option describes the policy for handling traffic originating from the zone for which the routing algorithm has determined the destination to be another interface in that zone.

forwarded between different network interfaces within the zone. These must not be confused with the input and output chains of iptables.

18.3.1 Input rules

The Input rules handle the traffic originating from one of the zone's network interfaces destined to a socket of the host i.e. a webserver application. As an example, if the input policy for the LAN network is set to drop with no exceptions, devices in the LAN network are no longer able to reach the web interface of the gateway.

In this example the browser takes some time, then displays a timeout error. The reason is, that the packet gets dropped before it reaches the web server. If it had been set to reject, almost instantly an ICMP unreachable response would have been sent, which would have let the browser displayed a "website unreachable" error instead.

In case a packet is being accepted by the Input policy, a connection tracking entry will be created for this so-called "flow" allowing a bi-directional communication even if the Zone of the instantiator of the connection has a Drop or Reject output policy.

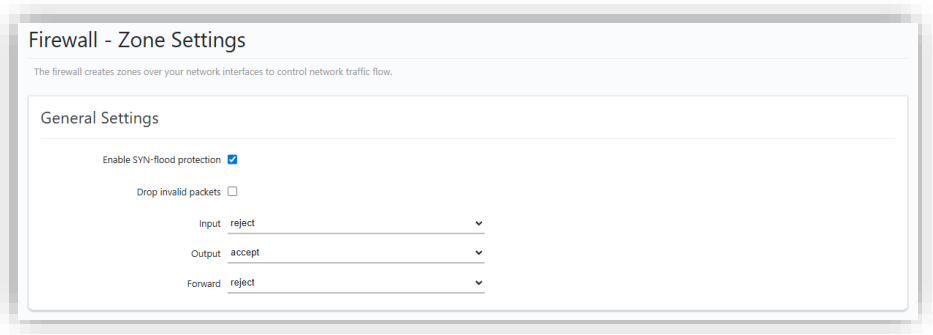
Accordingly, accepting the packet will not only pass the packet from the LAN interface to the web server, but allows the web server to reply regardless of the output policy of that zone.

18.3.2 Output rules

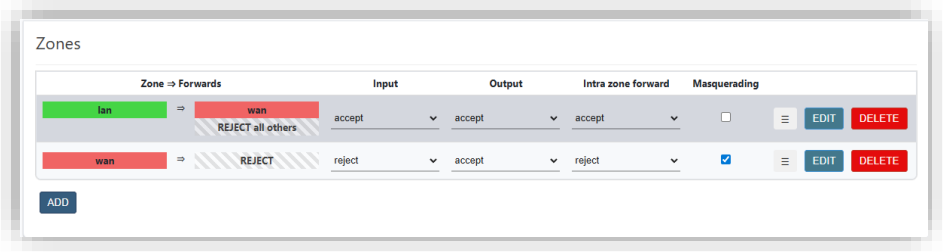
The output rules polices traffic that originates from the host that is destined to a network interface in the corresponding zone. As the instantiator of the connection is the host itself, this decision is done right before transmitting the packet on that network interface. The filter could still manipulate or even just drop the packet aborting the transmission procedure.

18.3.3 Forwarding rules

If set to *Accept*, traffic originating from a network interface in the zone is permitted to be forwarded to another network interface in that zone. This means it could be routed if a suitable route is present. Otherwise, it will be silently dropped (Drop) or dropped and replied to with an ICMP unreachable message (Reject).

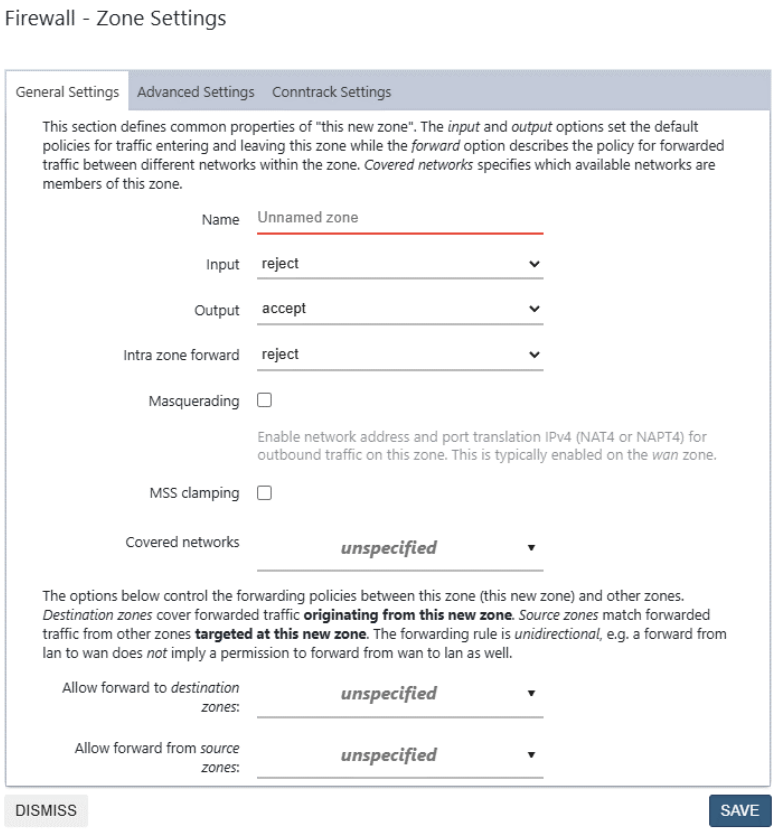


The zone default rules for Input, Output and Forward traffic are being overridden by a set of zone-specific rules defined below the basic configuration options.



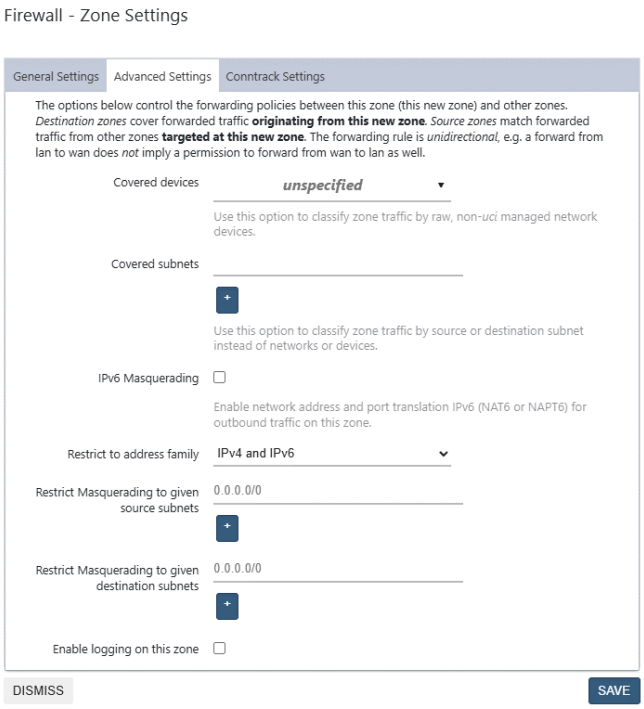
The forwarding rule is unidirectional, e.g. a forward from LAN to WAN does not imply a permission to forward from WAN to LAN as well.

18.3.4 General zone settings



MSS clamping: MSS clamping automatically fragments and defragments TCP packets forwarded between network interfaces with different MTU characteristics. Ordinarily PMTU discovery would lead the TCP connections to use a packet size fit for the lowest MTU along the path. This could limit the throughput if there is a big difference in MTU sizes. MSS clamping may improve this situation. It however requires a higher computational effort for fragmentation and defragmentation, which may lead to performing worse or causing latency issues. Sometimes it may be necessary to use MSS clamping if the MTU of a path is too small. E.g. IPv6 packets require a packet size of at least 1280 bytes and could otherwise not fit through an interface with a smaller MTU. Usually, it is safe to turn MSS clamping off.

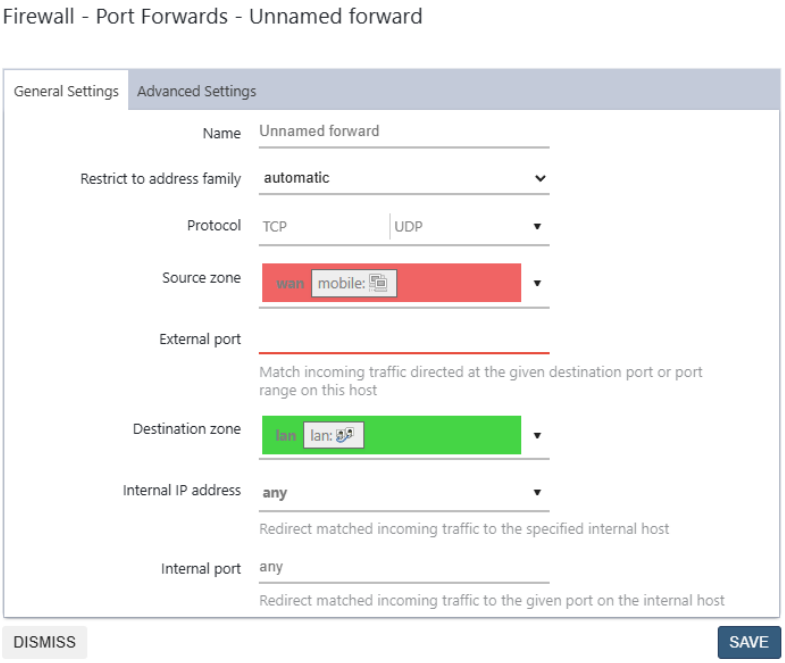
18.3.5 Advanced zone settings



- Restrict to address family defines to what IP families the zone belongs to (IPv4, IPv6 or both)
- Restrict masquerading to given source/destination subnets defines one or more subnets for which the masquerading option is applied to.
- Connection tracking and logging options enable additional information gathering on the zone.
- Controls of the forwarding policies between new/edited zone and other zones.
- Destination zones cover forwarded traffic originating from the new/edited zone.
- Source zones match forwarded traffic from other zones targeted at the new/edited zone.

18.4 Port forwards

The common use for port forwarding is to allow remote computers from the WAN network (Internet) to initiate a connection to a port of a service running on a computer within the private (firewalled) LAN. Simply and more generally expressed it allows to prick a hole for a specific port in the firewall and redirect its incoming packets to a pre-defined port-address combination in another network. Settings for the port forwarding of the device are defined as follows:



Name: The name of the port forwarding rule.

Protocol: Used protocol (Any/TCP/UDP/ICMP)

Source Zone: Informs which interface forward is matched to.

External port: Informs what port forward is matched to.

Destination Zone: Informs which interface is forwarded to.

Forward to: Informs where the port is forwarded to.

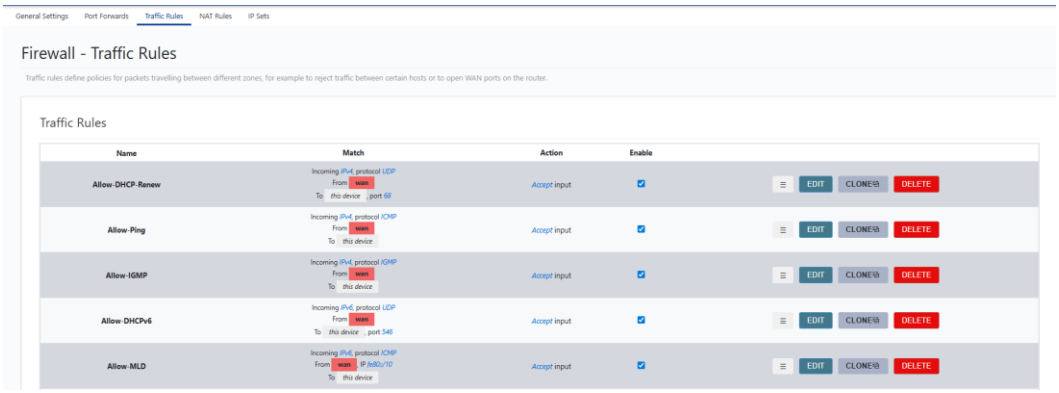
Internal IP address: Redirect matched incoming traffic to the specified internal host.

Internal port: Redirect matched incoming traffic to the given port on the internal host.

The user can add, edit, or delete port forwarding rules.

18.5 Traffic rules

Traffic rules define policies for packets traveling between different zones. The matching filter allows a fine granular definition for what kind of traffic the action is being performed.



Traffic can i.e. be matched based on:

- IP protocol
- Source and destination zones
- Source IP address and port
- Destination IP address and port
- Source MAC address
- Packet mark
- DSCP (QoS)

The name of the traffic rule is just for internal reference and can be arbitrarily chosen.

18.6 NAT rules

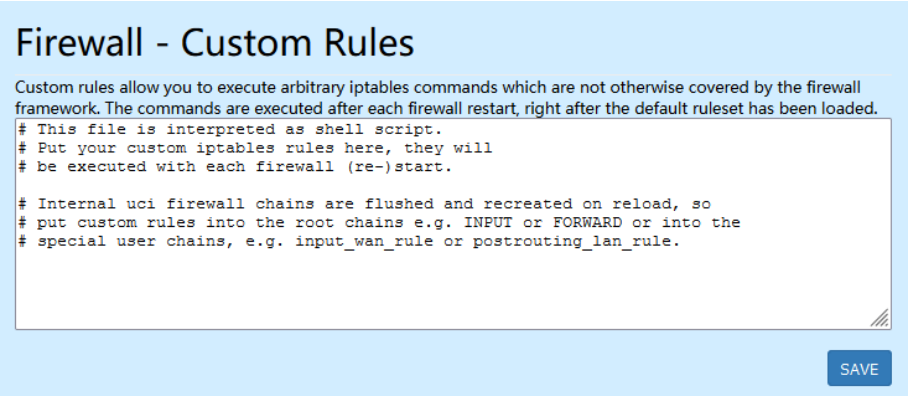
SNAT (Source NAT) allows to rewrite the source IP address of packets used for an outgoing traffic flow. In conjunction with connection tracking and a DNAT (Destination NAT) for the incoming replies of the traffic flow, this becomes the so-called Masquerading which is quite popular to share a limited set of WAN IP addresses between different devices in the LAN network.

The user can add, edit, or delete source NAT rules. For every rule these options can be defined:

- Name
- Protocol
- source and destination zones
- source
- destination
- SNAT IP addresses
- Ports
- extra arguments
- month
- weekdays
- start/stop dates and times, time in UTC.

18.7 Custom rules

Custom rules allow to execute arbitrary iptables commands which are not otherwise covered by the firewall framework. The commands are executed after each firewall restart, right after the default ruleset has been loaded.



19 VPN (Virtual Private Network)

A VPN is a concept for establishing cryptographically secured tunnels to otherwise isolated networks (e.g. an office network) via an unsecured medium (usually the internet).
By using cryptographic authentication, the integrity of the transmission can be checked while encryption hampers eavesdropping. The most widespread open-source software solutions for this task are IPSec, OpenVPN and WireGuard.

19.1 Protocol overview

As IPSec requires a deep cryptographic understanding to be configured securely, OpenVPN emerged as a viable alternative due to its simpler configuration, leading to widespread adoption.

From a performance perspective OpenVPN is inferior to IPSec and WireGuard as OpenVPN processes run in user land. At the time of writing network interfaces in user land require one syscall for each network packet to be processed. This causes a high number of context switches, thereby reducing cache hotness and overall system efficiency. Furthermore, OpenVPN does not support multi-threading. As a rule of thumb this performance disadvantage should be considered if more than 20 Mbps throughput are to be expected.

To address OpenVPN's performance limitations and further simplify and foolproof the cryptographic setup, WireGuard was introduced in 2020. However, WireGuard's drawback lies in its limited configuration parameters for encryption, authentication algorithms, and features like TAP/Layer 2 tunnels or certificate authentication. The intentional limitation of cryptographic configuration capabilities by WireGuard's author aims to prevent users from choosing insecure settings.

For operators of VPN server infrastructure WireGuard's lack of official support for backward compatibility between versions poses challenges. Deploying WireGuard on devices in the field, such as customer premise equipment (CPE), requires an alternative method for updating the devices independent from the VPN connection, as power failures could potentially lock the device out of the network if the server's version is updated in absence of the device. This, combined with the absence of certificate authentication capabilities, can make WireGuard deployment challenging in certain scenarios.

Together with the missing certificate authentication capabilities this sometimes makes it unattractive for professional CPE setups.

From a security point of view, given that a knowledgeable security expert constantly reviews the systems, it is wishful to be able to support a set of cryptographic algorithms on the server as they constantly evolve. Processing capabilities (CPU command set, hardware crypto accelerators) and security requirements might differ with devices and application scenarios. With IPSec's ability to negotiate the set of cryptographic algorithms it can serve individual demands to e.g. adjust the encryption strength and make efficient use of the hardware capabilities on a connection level. This way new cryptography can coexist with legacy configurations allowing smooth migrations.

To summarize: WireGuard is easy to use, OpenVPN is widely adopted and IPSec is complicated, but best fit for advanced setups.

19.1.1 Public key cryptography

Compared to shared secrets, public key or asymmetric cryptography offers the main advantage of being able to setup a secure communication channel via an insecure medium. Instead of having a shared secret, with public key cryptography both communication parties have their own secret key.

These so-called private keys come with a corresponding public key and allow their owner to sign data. The signatures can then be verified by everyone to belong to that specific public key.

The public key is not a secret. While it can be easily algorithmically derived from the private key, the other way round, deriving a private key from a public key, is considered hard for secure algorithms.

Besides being able to sign data and verify the resulting signatures, asymmetric encryption schemes allow encrypting data using a public key so that the corresponding private key is required for decryption. Simply put, it creates a ciphertext only the receiver can decrypt.

With this method, a secure communication channel is established then in which e.g. a shared secret can be negotiated. This way ciphers with more efficient symmetrical encryption schemes can be used.

19.1.2 Certificates

Using public key cryptography, it is possible to build a chain of trust. A commonly trusted third-party and its private/public key pair are chosen as the root of trust. This commonly trusted third party is usually referred

Certificates are CSRs (Certificate Signing Requests) that have been signed using public key cryptography by a CA (Certificate Authority).

CA (Certificate authority):

Certificate
Private Key
Public Key

19.1.3 Security concerns

There are basically 4 different files

To prevent attackers from compromising the root certificate’s corresponding private keys, it is good practice to generate and store the keys on a physically secured fully encrypted computer that will never have any network access or even wireless network peripherals. This computer must only be accessible to authorized persons for signing certificates. In high security environments this computer could be stored in a safe with a detached display and camera only. Special applications can be used to visually transfer the signing requests and signatures (QR codes) to prevent interface attacks (e.g. BadUSB attack).

1. The transmission of a certificate (*.crt) and its associated private key (*.key) over the same medium is strictly prohibited.

2. It is imperative that a private key (*.key) is kept in only one location at any given time.

3. Symmetric encryption of a private key (*.key) is essential, and EasyRSA requires a "passphrase" for this purpose. However, it is not practical to symmetrically encrypt a private key (*.key) for server use, as it would require manual entry of the "passphrase" every time the server is started. Therefore, other security measures must be in place to prevent physical access to the server.

4. To ensure maximum security, the certification authority signs certificate requests (*.csr) from clients and servers without knowledge of their private key (*.key). Refer to the README.quickstart.html file for more information.

5. EasyRSA utilizes key lengths of 2048 bits, but stronger key lengths can be configured separately if desired.

6. The creation of DH parameters with 4096 bits may take several days.

7. Server and client certificates generated with EasyRSA are valid for 2 years (825 days) and must be recreated or renewed thereafter. 1. The guidelines provided were formulated to the best of our understanding and belief, taking into account the latest technological advancements. Nevertheless, new security weaknesses may emerge on a daily basis, potentially rendering the procedures outlined in this manual unsafe. Consequently, utilizing this manual is done at your own peril and accountability. It is incumbent upon the user to evaluate the suggestions, principles, directives, and setups outlined herein. The decision of whether and how to adhere to these recommendations rests with you as a conscientious administrator.

19.2 OpenVPN

The OpenVPN protocol usually uses a client-server approach. The client acts as the initiator and establishes the connection to the server. It has not been standardised, but software client implementations exist for a lot of devices including smartphones.

OpenVPN offers two different types of tunnels. The usual one uses a TUN interface, which allows to exchange OSI layer 3 data packets (IP protocol) like a gateway, while the TAP interface type allows to exchange OSI layer 2 packets (ethernet) like a network switch connecting the two remote networks.

The initiation of an OpenVPN connection requires a matching client and server configuration defining the authentication, encryption, and address mapping settings.

19.2.1 Authentication

The three methods for authentication that OpenVPN supports are a username-password combination, a pre-shared key (PSK), public key authentication and authentication using certificates.

A username-password combination is the most intuitive approach but requires a strong password for adequate security. While this is not that much of a concern with the pre-shared key approach, one must consider that a compromised password or pre-shared key allows an attacker to imitate both the client and the server in a MITM attack scenario (Man-in-the-middle). This is especially problematic if multiple devices share the same credentials.

By using public keys or certificates for authentication however, the server and client can have their own set of private keys. They verify the authenticity of the initiation message using each other's public-key or confirm that the certificate has been signed by a trusted-third party (CA - certification authority). This way e.g. a compromised client's private key does not necessarily compromise the server. Certificates have the additional benefits that they can have an expiration date and in case e.g. a server's private key has been compromised, a new certificate can be issued for a new public-private key pair without having to inform the clients about the new server's public key.

19.2.1.1 Authentication using username and password

```
# Save username/password credentials

umask go=

cat << EOF > /etc/openvpn/client.auth

USERNAME

PASSWORD

EOF


# Configure VPN service

cat << EOF >> /etc/openvpn/client.conf

auth-user-pass client.auth

EOF

service openvpn restart
```

19.2.1.2 Creating a CA

A CA is a public-private key pair that is being used to sign certificates which usually contain public keys of entities to trust and definitions of the granted rights. In case of OpenWrt these are the server and client public keys.

19.2.2 Client configuration

In the following, an example of interpreting and configuring OpenWrt with an OpenVPN client configuration is given, like they are usually provided to their customers by commercial VPN providers. For setups that require a more sophisticated configuration, it is well-advised to read the official OpenVPN documentation.

The required certificates are usually also provided with the configuration details. OpenVPN can be configured either by using the web interface or by uploading the configuration files.

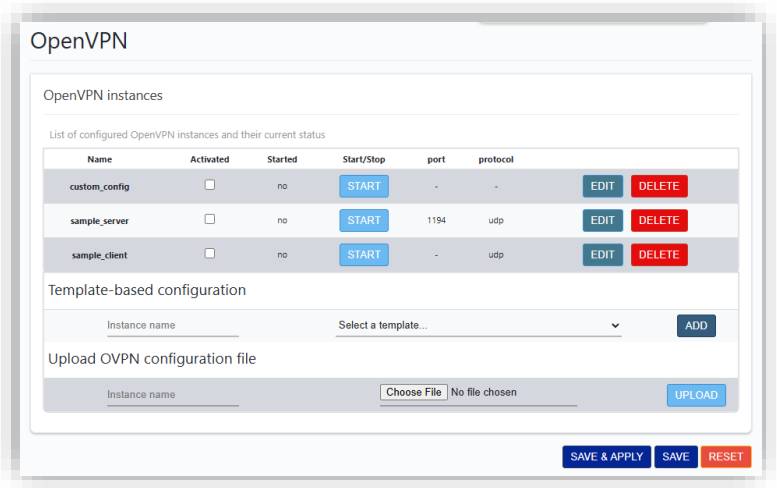
????OpenVPN will automatically attempt to load all *.conf files placed in the /etc/openvpn folder. OpenVPN example configurations are available online. These can usually be adapted with minor changes.

Before starting, create your own Certificate Authority (CA), certificates and keys for an OpenVPN server and clients.

You will need:

- Certificate authority (ca.crt)
- Server certificate (server.crt) and server key (server.key)
- Client certificate (client.crt) and client key (client.key)
- Client configuration

Navigate to *OpenVPN->VPN*.

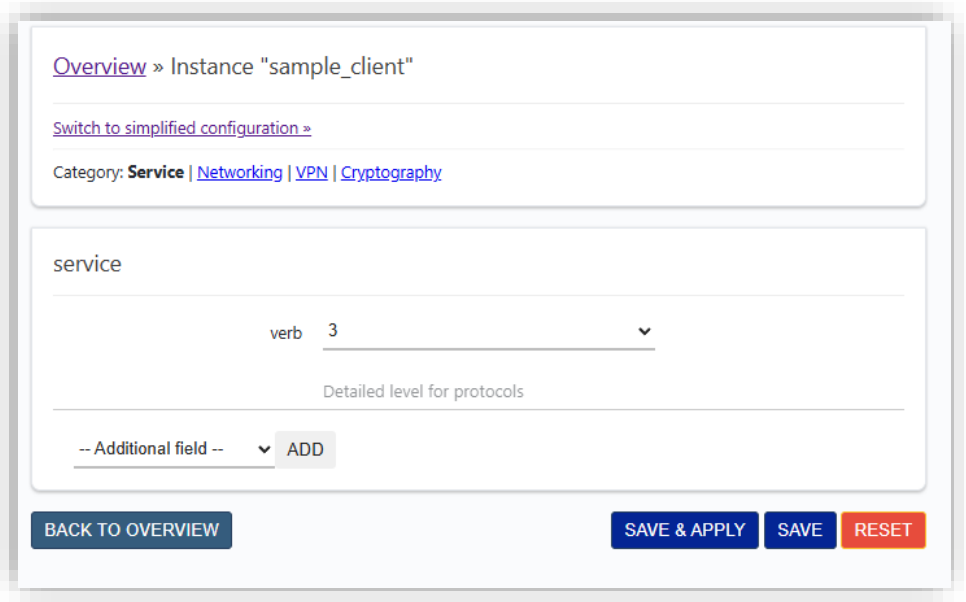


Click *EDIT* next to *sample_client*.

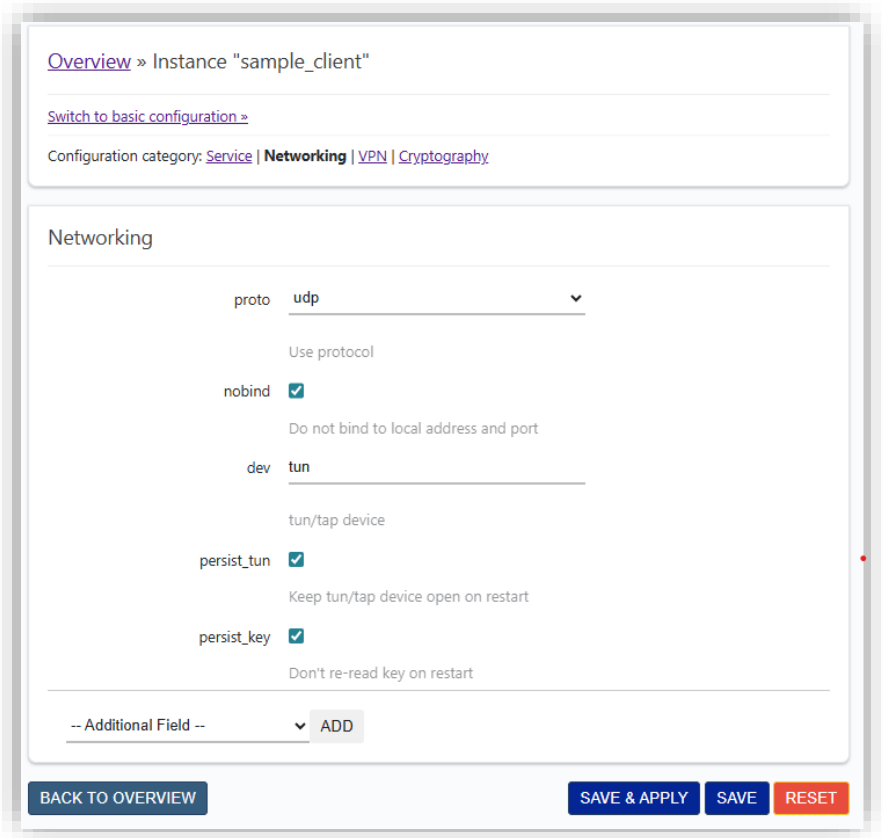
Click *Switch to advanced configuration*.

Example:

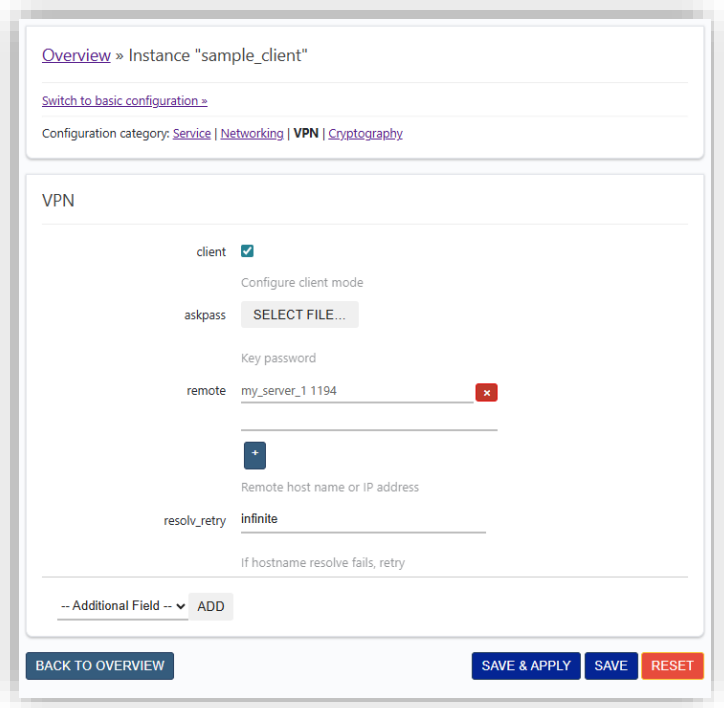
Under Service, change "verb" (verbosity) same needed.



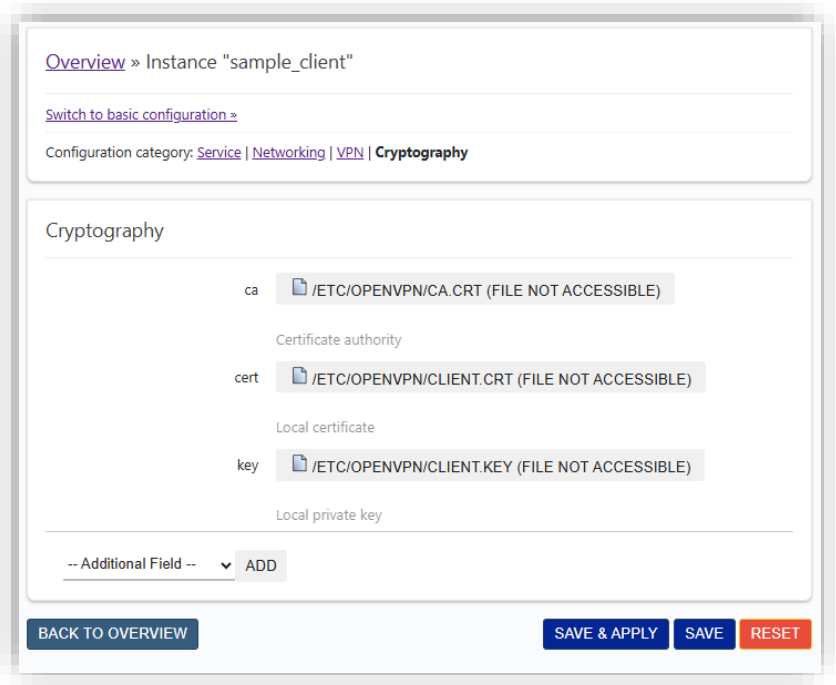
Under Networking, change every setting as same needed.



Apply the same settings for the VPN section.

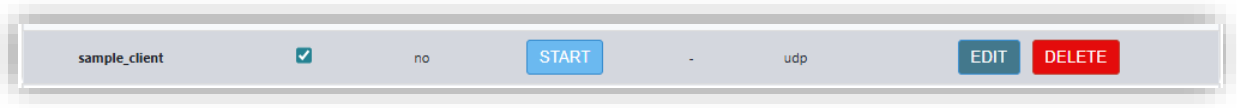


Under *Cryptography* change the cipher type to the one used for your certificates (You can add missing configuration using –Additional Field-- option).



Upload the files to the /etc/luci-uploads/ folder.

Click *SAVE & APPLY* and then *BACK TO OVERVIEW*.



Enable the configured instance, then click "SAVE AND APPLY", then "START".

20 Node-RED introduction

Node-RED is a visual programming tool that allows to easily interconnect and control web APIs, IoT devices, APIs and online services to be wired together. It is builds on Node.js, fully utilizing the advantages of its event-driven, non-blocking model.

It offers a browser-based editor that enables to easily wire flows by using the wide range of nodes in the palette.

You can reach the Node-RED web interface with the same IP address as the standard web interface and the specification of the port (e.g. 1880)

Example with default address: https://192.168.2.1:1880

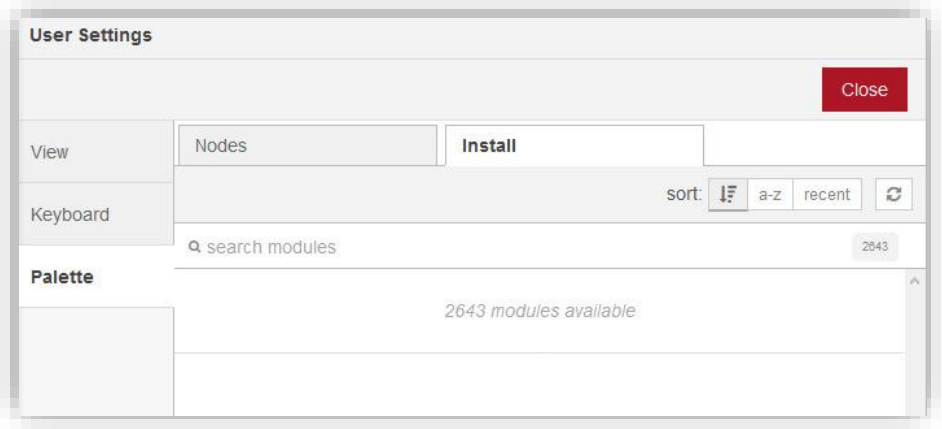
20.1 Enabling Node-RED

In System->Startup click the "Enable/Disable" button next to the “node-red” service.

20.2 Installation of modules

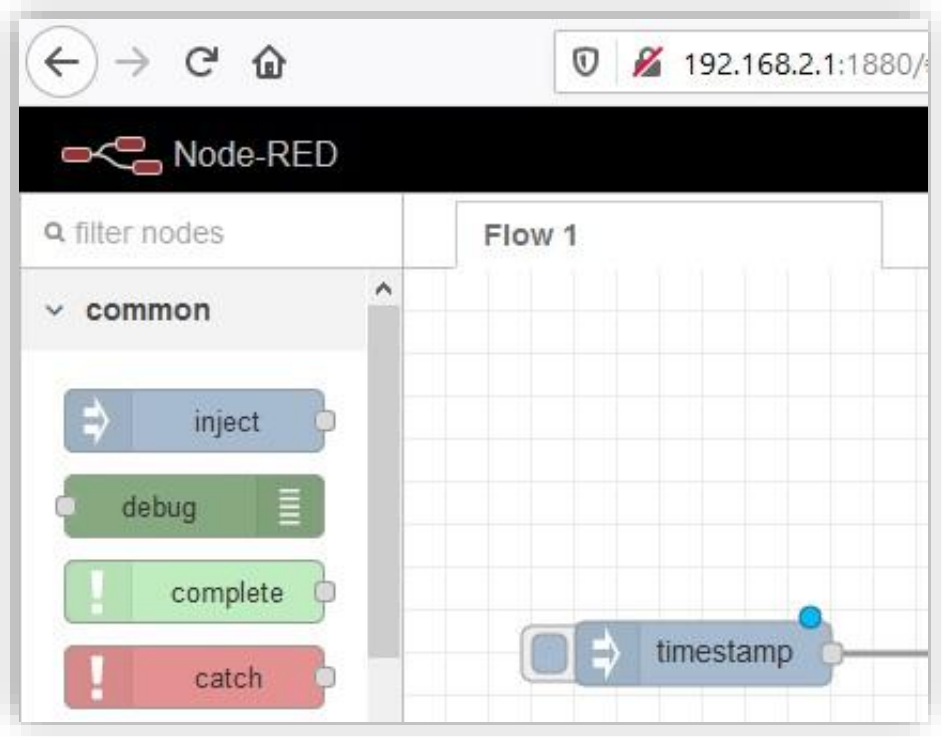
To install modules for Node-RED:

1. Navigate your browser to the Node-RED web interface
2. Open the burger menu 
3. Click *Manage Palette*.
4. Switch to the *Install* tab



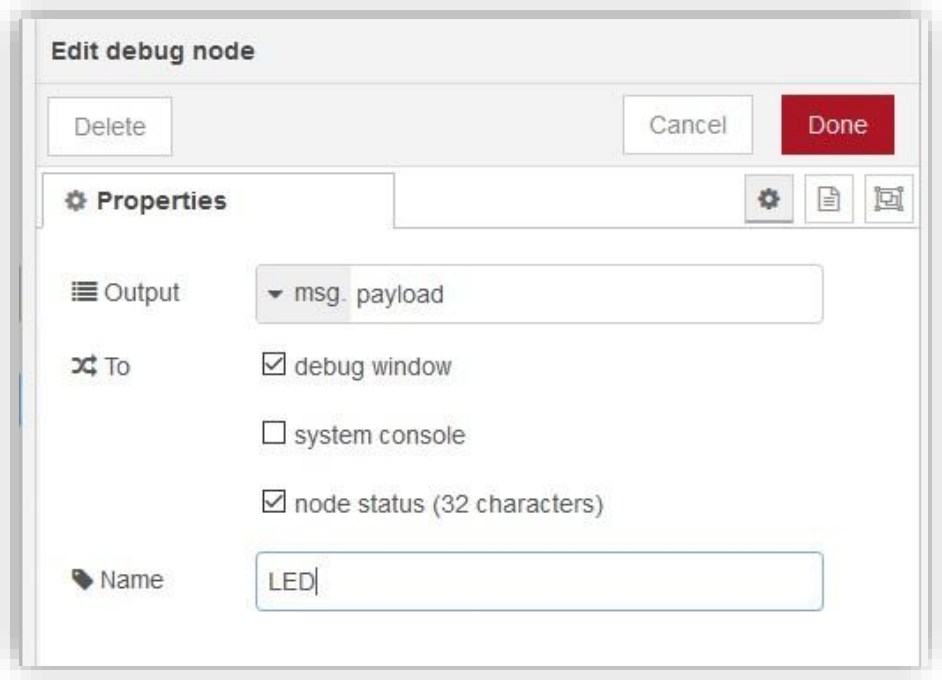
Enter the name of the module into the search input field and click *Search modules*.

20.3 Adding a node



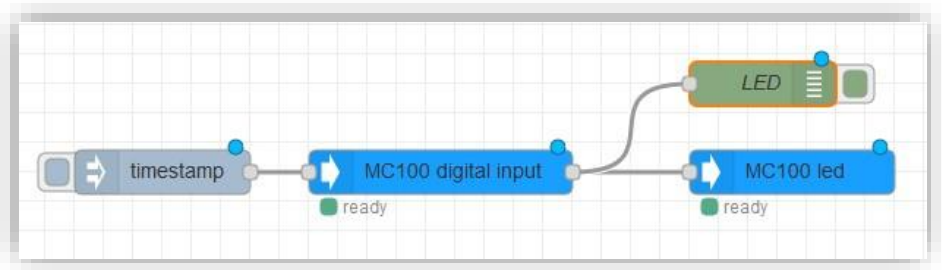
Select the node you want to add from the nodes list (i.e. Inject node, allow injecting messages into a flow). Drag it into the workspace.

20.4 Adding a debug node



The debug node displays the payload of the message or the entire message object. It can be renamed from its setting by double clicking on it.

20.5 Connecting the nodes



After adding all the desired nodes, wire them together by dragging a line from an output of one node to the input of another node thereby feeding the output data of one node to the input of the other.

20.6 Deploying

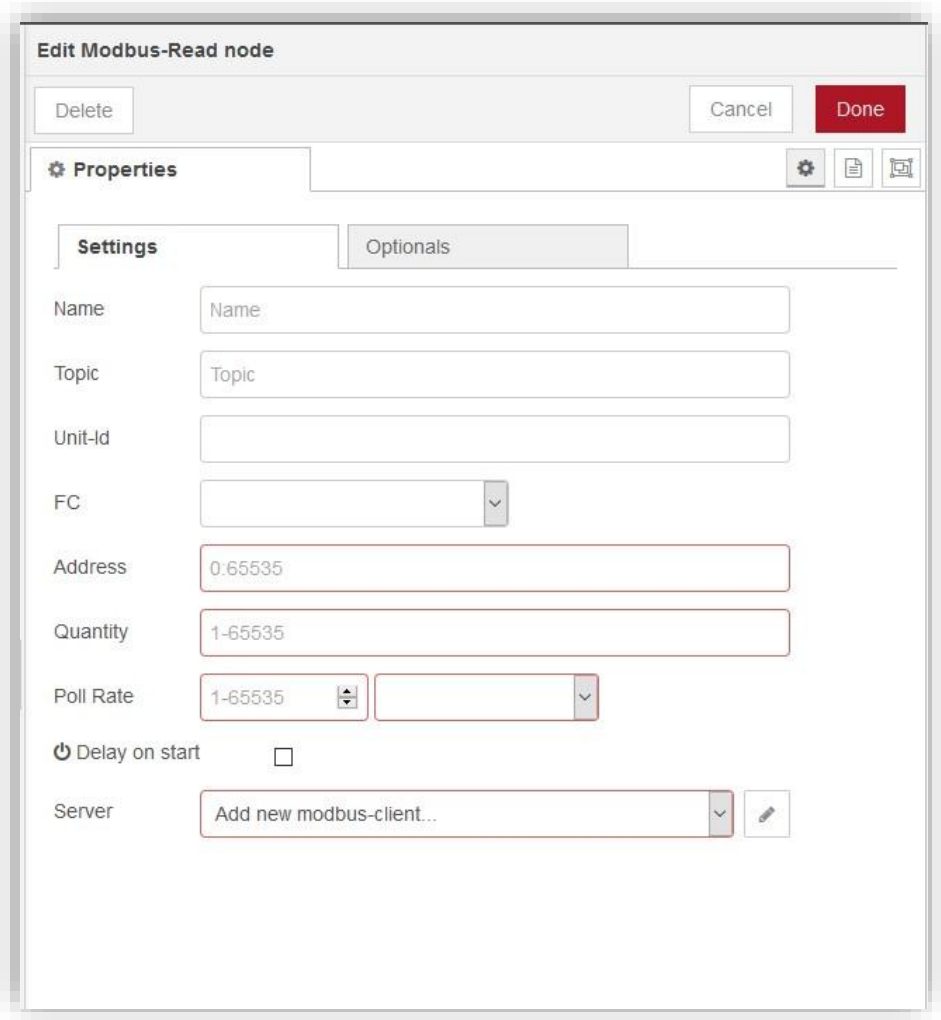
To make the changes come into effect and start the processing, click the *Deploy* button in the upper right corner.

20.7 Modbus with Node-RED

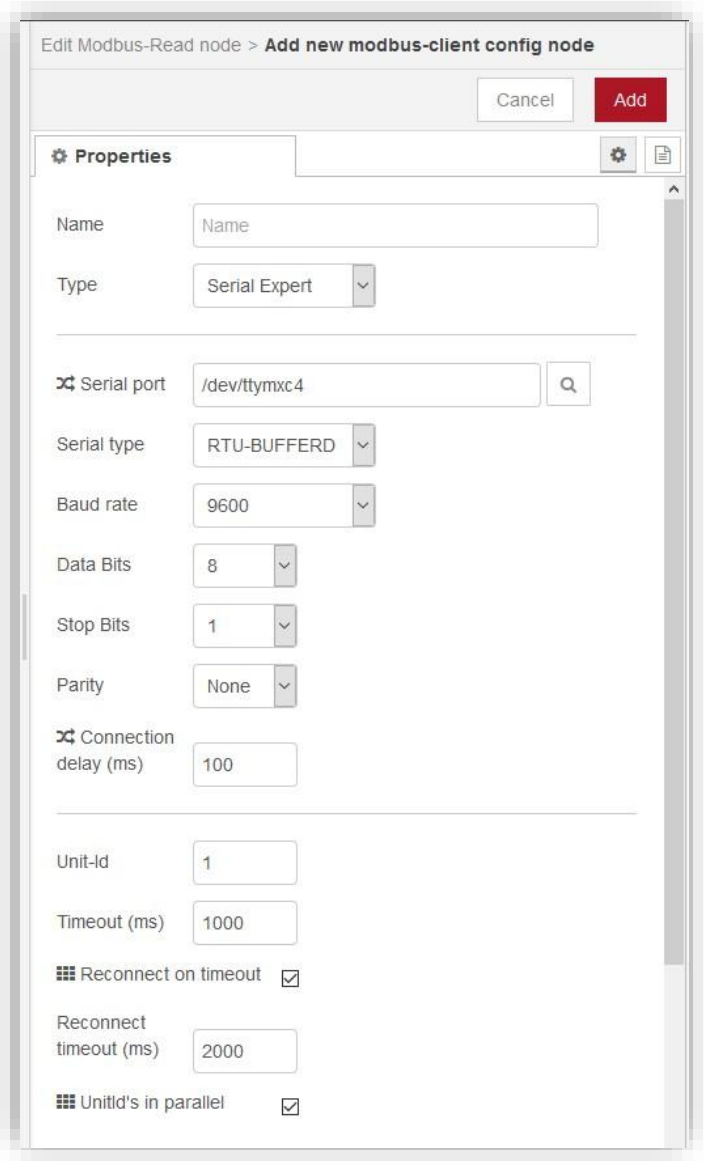
node-red-modbus is factory installed on MC100. Do not try to install the node-red-contrib-modbus package from within Node-RED.

20.7.1 Creating a first flow

1. Open Node-Red in your browser.
2. Add Debug node.
3. Add Modbus read node.
4. Wire them.
5. Double click the Modbus read node to configure it.
6. Change the settings depending on the device you want to read out.
7. Click the edit button near Server to configure the Modbus device.



8. Change Type to Serial Expert.



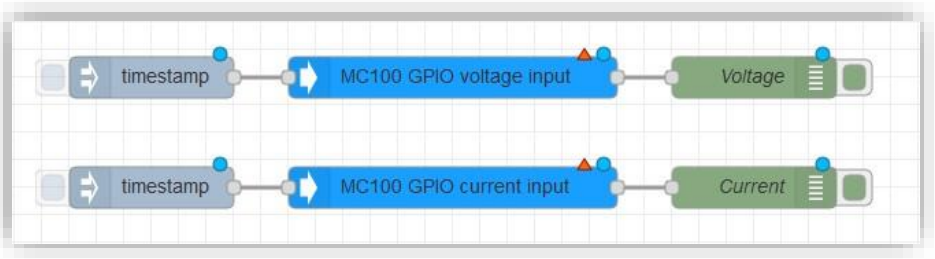
9. Change Serial port to /dev/ttymxc4
10. Make sure, that the Serial Type is RTU-BUFFERED and not RTU.
11. Deploy.

In the *Debug* tab the messages can be viewed.

20.8 MC100 GPIO

20.8.1 Analog inputs (current or voltage)

1. Add the inject node
2. Add voltage input/current input node
3. Add Debug node.
4. Wire and Deploy.

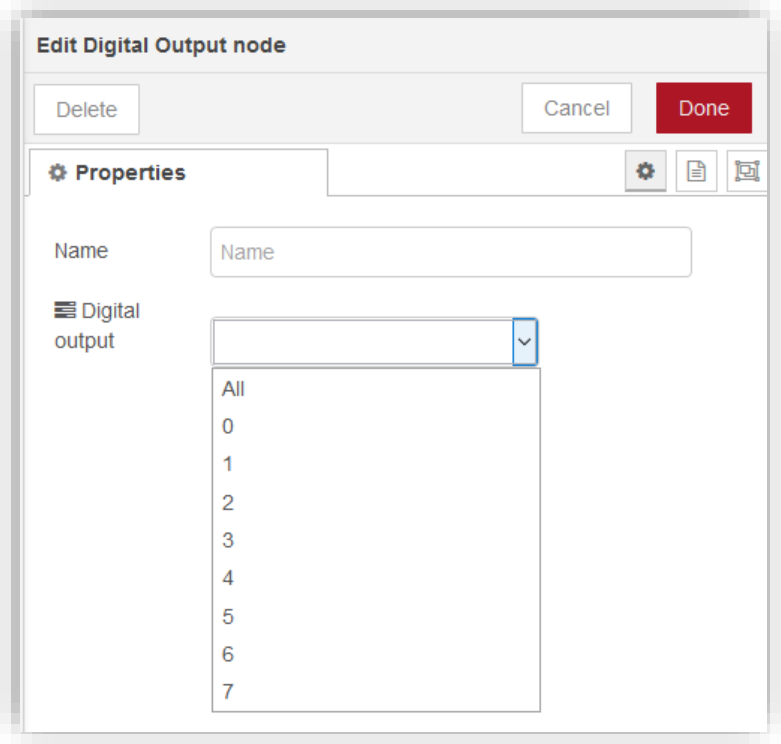


20.8.2 Digital inputs

1. Add the inject node.
2. Add the digital inputs node.
3. Double click the MC100 GPIO digital input node to open the settings menu and choose a Digital input.
4. Add Debug node (change name if desired).
5. Wire and Deploy.

20.8.3 Digital outputs

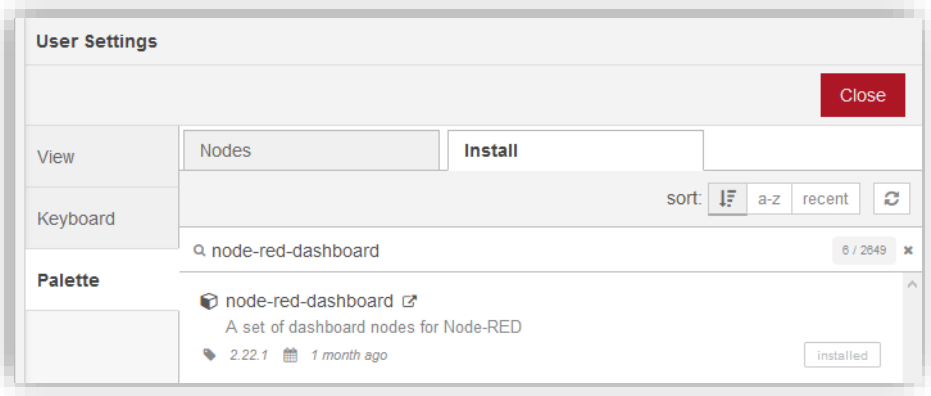
1. Add the Inject node.
2. Add the Digital output node.
3. Double click the MC100 GPIO digital output node to open the settings menu and choose one of the Digital inputs from the drop-down list.



4. Add Debug node (change name if desired).
5. Wire and Deploy.

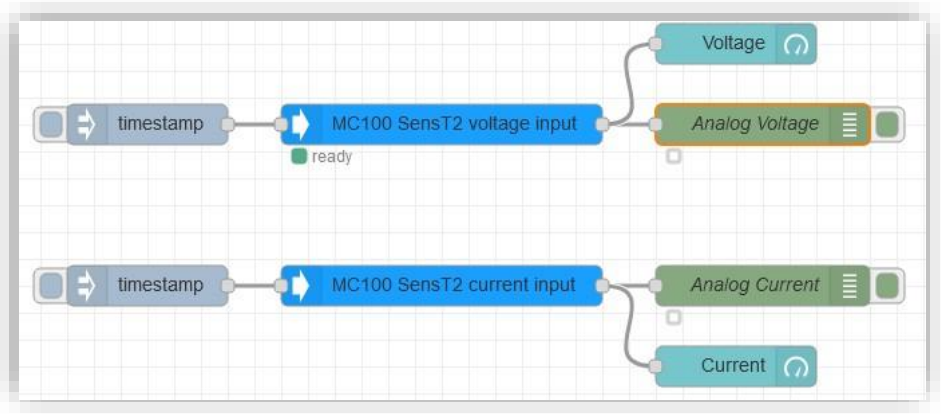
20.8.4 Dashboard

1. Use Menu - Manage palette to search for “node-red-dashboard” and click install.

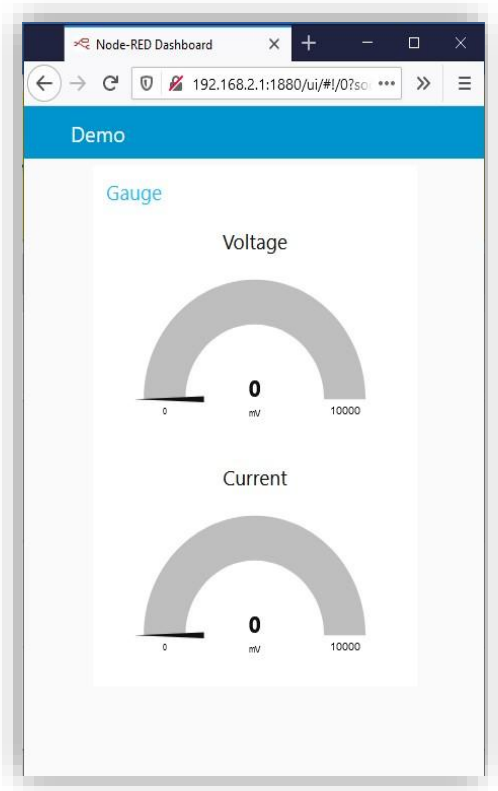


2. Restart Node-RED, dashboard tab should appear in the right-side panel.

3. From dashboard tab, add the desired nodes (e.g. gauge) and wire them.



4. Double click the nodes to change their properties as desired.
5. In a new tab open http://localhost:1880/ui (e.i. http://192.168.2.1:1880/ui)

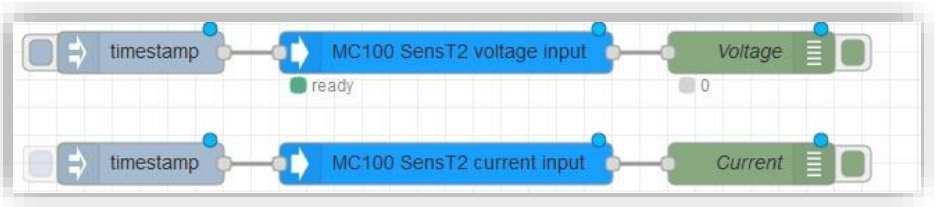


20.9 MC100 SensT2

20.9.1 Analog inputs

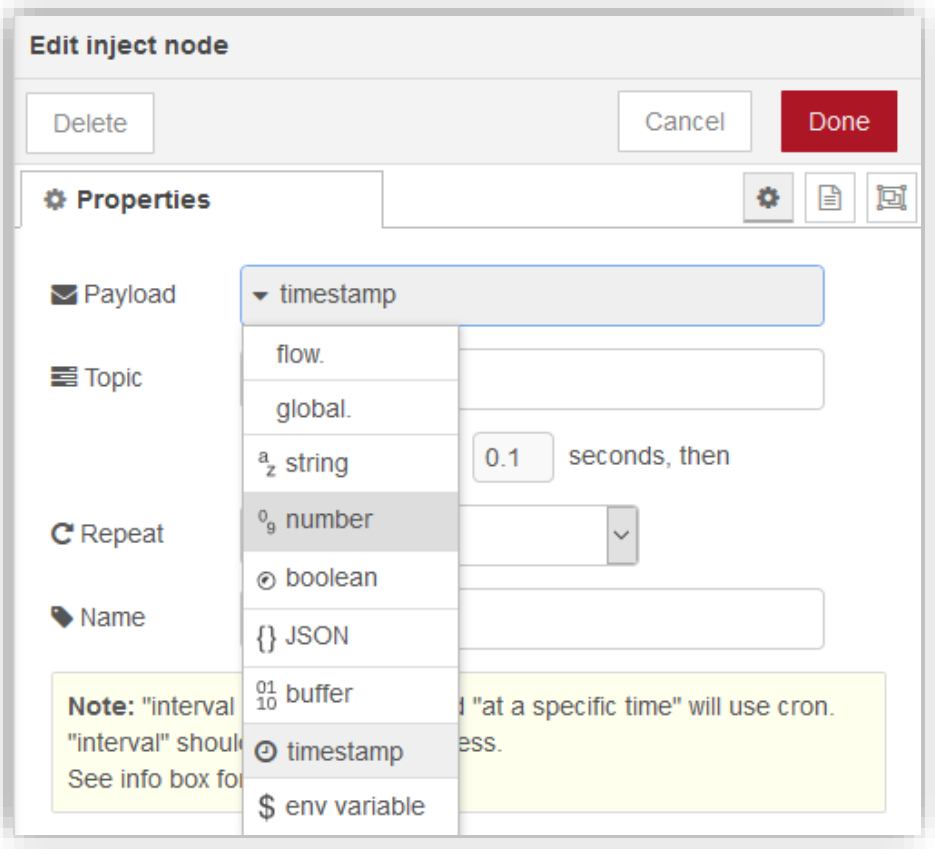
1. Add the inject node.
2. Add voltage input/current input node.

3. Add Debug node.
4. Wire (as shown in picture) and Deploy.



20.9.2 Write analog output

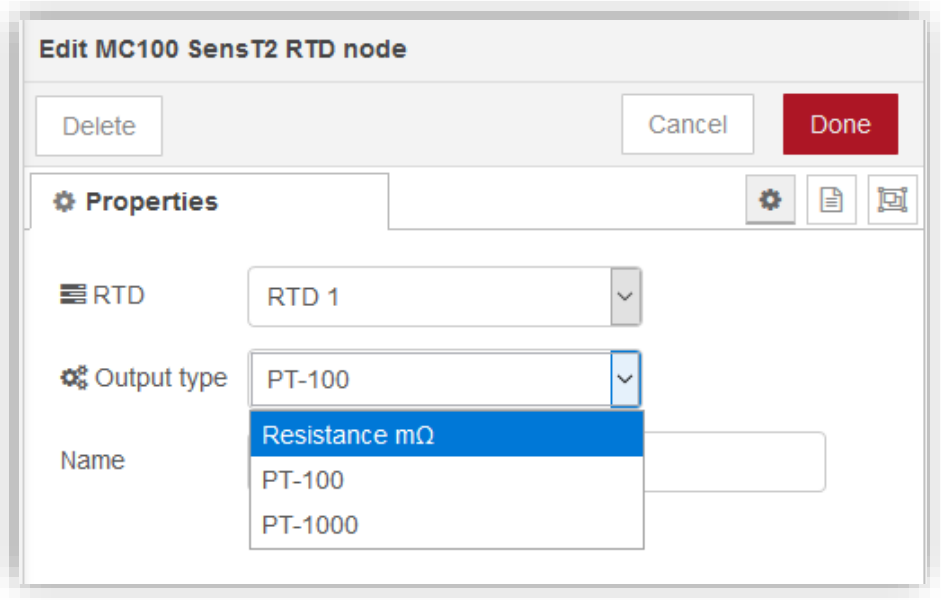
1. Add Inject node, double click it and change the payload to Numbers and enter the analog value (= Current in [4-20] mA).



2. Add MC100 SensT2 current output node.
3. Wire and Deploy.

20.9.3 PT100/1000

1. Add the Inject node
2. Add MC100 SensT2 RTD
3. Double click the MC100 SensT2 RTD node to open the settings menu and choose RTD and Output type.
4. Add Debug node (Change name if desired)



5. Wire and Deploy

20.10 SMS with Node-RED

To be able to receive and send SMS using Node-RED, we recommend using *smstools3* which comes pre-installed. It needs to be enabled in *System->Startup* in the device's main web interface. Additionally, the module *node-red-contrib-smstools* needs to be installed in the Node-RED web interface.

Example flow:

```
[
    {
        "id": "ba93e143ba53abb0",
        "type": "tab",
        "label": "Flow 1",
        "disabled": false,
        "info": "",
        "env": []
    },
    {
        "id": "7f6ca9491a224ed4",
        "type": "sms-in",
        "z": "ba93e143ba53abb0",
        "name": "",
        "x": 690,
        "y": 120,
        "wires": [
            [
                "5472e1d192fd03ae"
            ]
        ]
    },
    {
        "id": "809bf79731e1eb78",
        "type": "sms-out",
        "z": "ba93e143ba53abb0",
        "name": "",
        "topic": "+4915786336816",
        "x": 1040,
        "y": 280,
        "wires": []
    },
    {
        "id": "3a3810b48635eda8",
        "type": "inject",
        "z": "ba93e143ba53abb0",
        "name": "",
        "props": [
            {
                "p": "payload"
            },
            {
                "p": "topic",
                "vt": "str"
            }
        ],
        "repeat": "",
        "crontab": "",
        "once": false,
        "onceDelay": 0.1,
        "topic": "",
        "payload": "hallo",
        "payloadType": "str",
        "x": 690,
        "y": 280,
        "wires": [
            [
                "809bf79731e1eb78"
            ]
        ]
    },
    {
        "id": "5472e1d192fd03ae",
        "type": "debug",
        "z": "ba93e143ba53abb0",
        "name": "",
        "active": true,
        "tosidebar": true,
        "console": false,
        "tostatus": false,
        "complete": "true",
        "targetType": "full",
        "statusVal": "",
        "statusType": "auto",
        "x": 1030,
        "y": 120,
        "wires": []
    }
]
```


Note: A reboot may be necessary for the changes to come into effect.

21 Diagnostics and system monitoring

21.1 Connectivity check

A tool for performing ping and traceroute tests can be found in Network->Diagnostics. Click the corresponding buttons under "Diagnostics" to perform a test.

21.2 mcinfo

mcinfo is a command line utility installed on the gateway which can be used via SSH. It offers in-depth debugging information like the modem status and device information.

mcinfo info	Print general information about the modem.
mcinfo mobile	Print information about mobile communication status.

```
root@MCLH:~# mcinfo
Usage: mcinfo [options] [command[ command]]

Options:
  -h          Show this help message and exit.
  -v          Print verbose debug information to error
  -V          Show version information and exit.
  -d DEVICE   Set the tty device (default: /dev/ttyUSB3)
  -c COMMAND  Send COMMAND to modem.
  -t TIME     define the timeout in deciseconds
                  default: 1

Command:
info   Print general information about the modem.
mobile Print information about mobil communication status
gpio   Print information about external GPIO module pins
root@MCLH:~#
```

21.3 act8847 hardware watchdog

The MC100 comes with an inbuilt hardware watchdog integrated into its PMIC chip ACT8847. MC Technologies has developed a kernel module for this hardware watchdog complying to the Linux kernel API specification for hardware watchdog drivers.

21.3.1 Parameter overview

The module is named mc-act8847-wdt and can be loaded with the following parameters:

nowayout	Prohibit stopping the hardware watchdog once it has been started. (type=boolean, default=0)

heartbeat	The time interval in seconds in which the user space needs to ping the kernel module to prevent a power cycle (value=integer, default=60)
hardreboot	Do not stop the hardware watchdog before a reboot or shutdown. This allows to power cycle the device instead of an ordinary reboot. (type=boolean, default=0)
hardunload	Keep the hardware watchdog active even when kernel module is unloaded. This makes sure the device performs a power cycle 4 seconds after the module gets unloaded and can be used to forcefully power cycle the device by unloading the module. (type=boolean, default=0)

21.3.2 Unloading the module

In case the module is already loaded, the module can be unloaded with the command:

rmmod mc-act8847-wdt

21.3.3 Loading with parameters

Multiple parameters can be defined separated by spaces:

insmod /lib/modules/<kernelversion>/mc-act8847-wdt.ko <parameter>=<value>

21.3.4 Changing parameters at runtime

Some parameters are writeable at runtime using sysfs entries in the following directory:

/sys/module/mc_act8847_wtd/parameters/

21.3.5 Setting persistent options

For setting options to be applied at boot time one can create a config file in */etc/modules.d/* e.g. *mc-act8847-wdt.conf* with options defined in the following format:

option mc-act8847-wdt-heartbeat=30

option-mc-act8847-wdt-nowayout=1

21.3.6 Using the watchdog

By default, OpenWrt comes with a watchdog daemon integrated in procd which runs as the init process (with PID1). It takes control over the watchdog device and automatically starts the watchdog timer at boot time. It is possible to deactivate procd's watchdog daemon to e.g. let a critical

application do the pinging instead. In case the critical application crashes or fails to ping the watchdog, the watchdog triggers.

21.3.7 Reading the status of procd's watchdog daemon

The status can be read by running the command: *ubus call system watchdog*

Example output:

```
{
        "status": "running",
        "timeout": 30,
        "frequency": 5
}
```

21.3.8 Switching to manual control

Only one process can open the watchdog file at a time. To deactivate the procd watchdog daemon execute:

```
ubus call system watchdog '{"magicclose":true}'
```

```
ubus call system watchdog '{"stop":true}'
```

21.3.9 Example of pinging the watchdog

Once the watchdog file has been opened, the timer will start. A write to the watchdog file will reset the timeout counter and prevent the watchdog from triggering. The following loop will run until the user sends a SIGINT (Ctrl + c). The watchdog eventually timeouts and triggers.

```
while :; do echo 1 > /dev/watchdog; sleep 5; done
```

21.3.10 Example of stopping the watchdog

Unless the *nowayout* parameter is set to true, closing the watchdog file can stop the watchdog if the magic character "V" is written right before closing the file.

The following loop will run until the user sends a SIGINT (Ctrl + c) which will then terminate the watchdog by writing the special character "V".

```
trap 'echo -n V > /dev/watchdog; break' SIGINT && while :; do echo 1 >
/dev/watchdog && sleep 5; done
```

22 Configuration and application examples

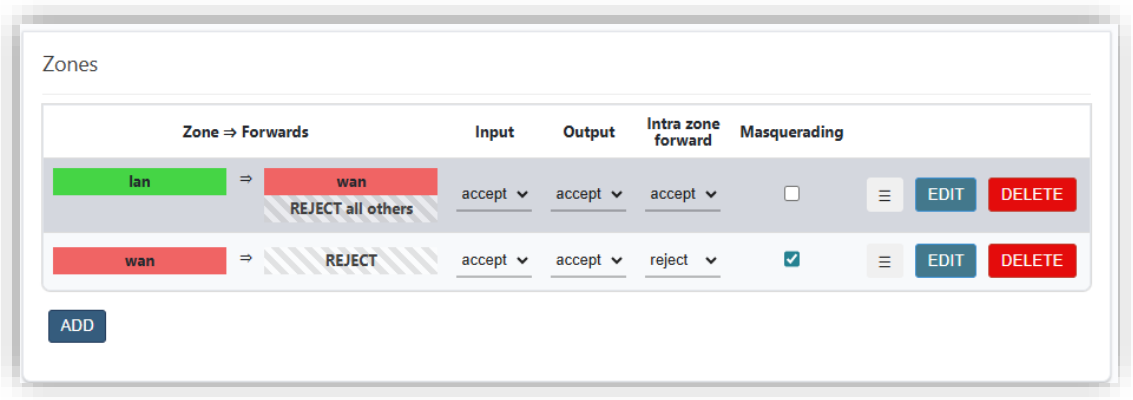
22.1 Reconfiguring an ethernet port as a WAN interface

If your gateway does not have a WAN interface configured already, it is necessary to reconfigure the existing LAN interface as a WAN interface. In case a WAN interface has been configured already, please ensure the settings match the suggestions outlined below.

22.1.1 Granting access to the web interface from the WAN network

Before removing the LAN interface, it is required to grant firewall access from the WAN network zone to the gateway. Otherwise, it will not be possible to access the gateway anymore and a factory reset might be the only resort to restore access. **Please be aware that this poses a security risk as the web interface and services will be exposed to all networks in that firewall zone including the mobile WAN network.** It is possible to e.g. create a separate zone for restricting access from the mobile WAN network, but that is out of scope for this example.

For granting access to the gateway from the WAN zone, select *accept* in the *Input* column of the *wan=>REJECT* forwarding rule and then click *SAVE & APPLY*.

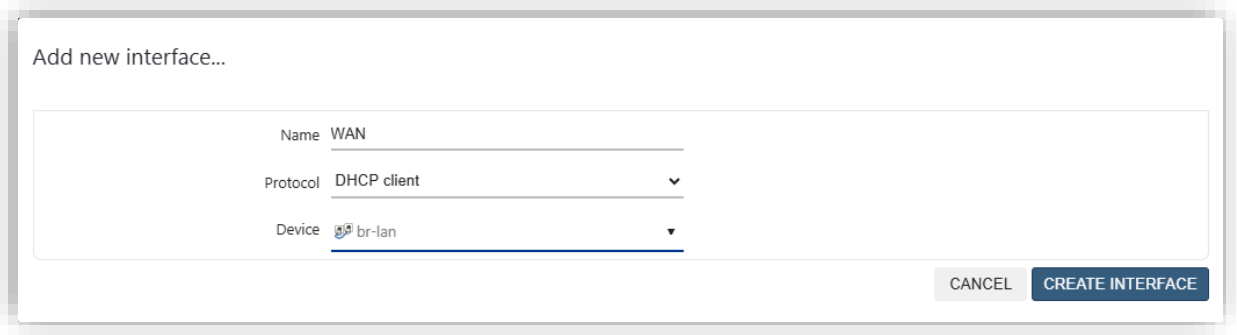


22.1.2 Removing the existing LAN interface

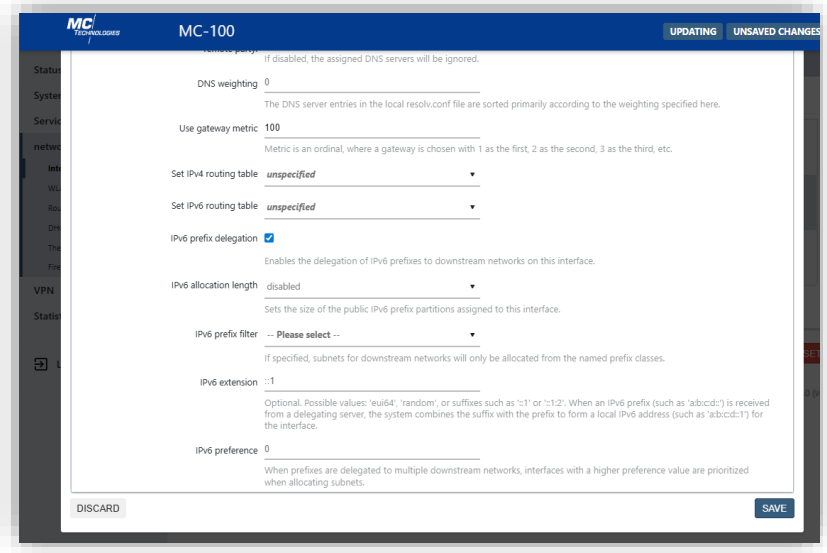
Navigate to *Network->Interfaces* and click the *Delete* button next to the LAN interface. **Do not apply the changes, yet.**

22.1.3 Creating a WAN interface

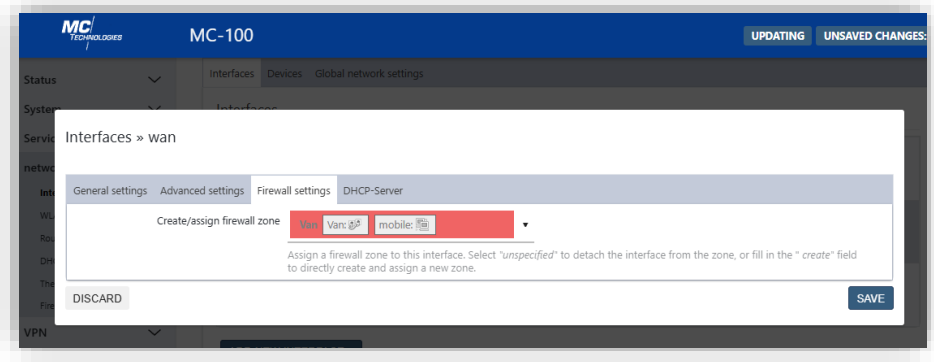
Click the *ADD NEW INTERFACE…* button. In the dialog, enter *wan* as the *Name*, select *DHCP client* as the *Protocol* and *br-lan* as the *Device*. Then click *CREATE INTERFACE*.



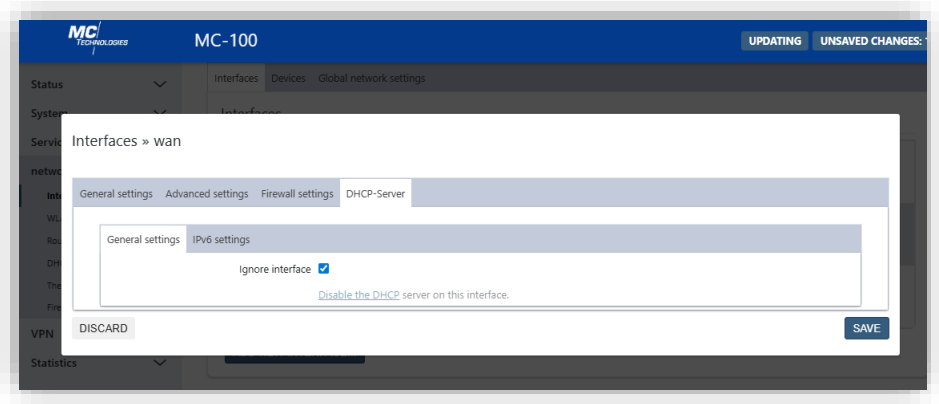
In the *Advanced Settings* tab, set *Use gateway metric* to 100.



Ensure that the interface is added to the firewall zone *wan* in the tab *Firewall Settings*.

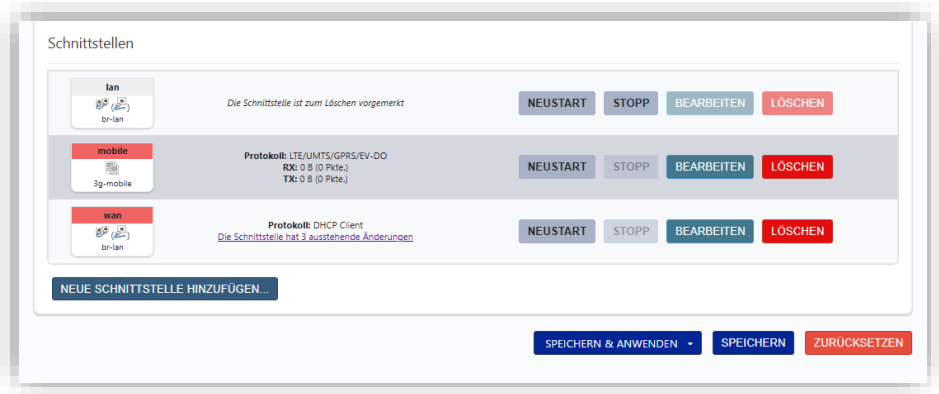


Also ensure that the *Ignore interface* checkbox is checked in the *DHCP Server* tab.



Click *SAVE*.

Before moving on to applying the changes, prepare for accessing the gateway from the WAN network in a timely fashion. After applying the changes, a countdown will start. If this countdown elapses before you were able to access the web interface from the WAN network, the gateway will revert the changes. This is a countermeasure against accidentally locking yourself out from the system. Once you are prepared for accessing the web interface from the WAN network, proceed by clicking *SAVE & APPLY*.



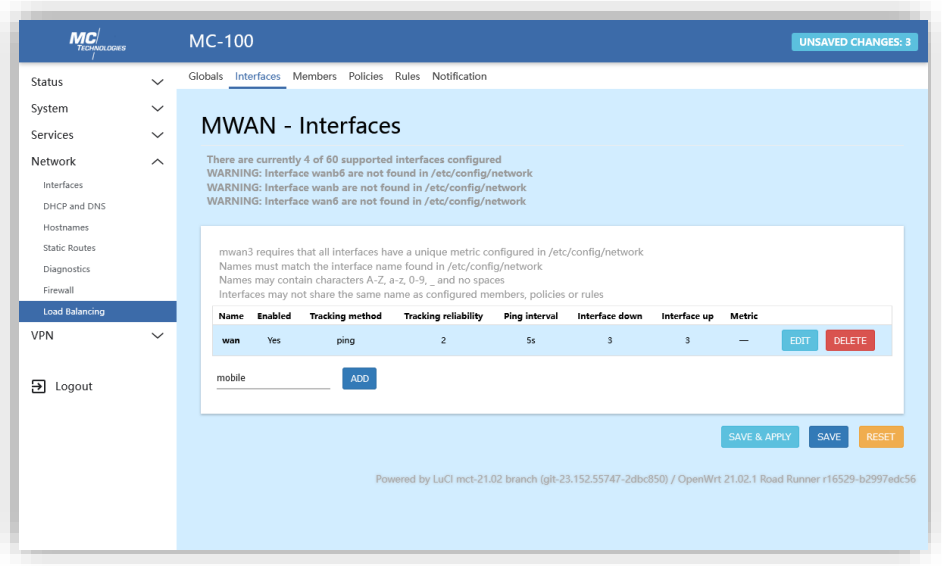
22.2 Connection fail-over and load balancing

It is highly recommended to set a metric value for all default routes of interfaces used with mwan3. This ensures that the routes with the lowest metric value will be preferred even in case of a failure.

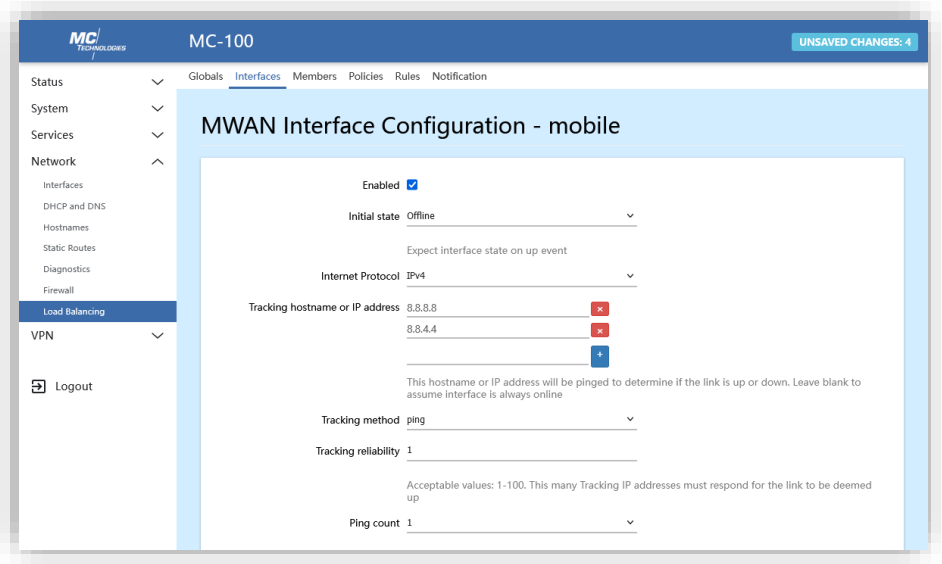
Click the *Edit* button next to the *mobile* interface. To configure the mobile interface as the preferred interface, set *Use gateway metric* to 50 in the *Advanced Settings* tab. Otherwise set it to 200.

22.2.1 Adding the interfaces for connectivity tracking

Go to *Network->Load Balancing->Interfaces* and delete all pre-configured interface definitions by clicking *DELETE* next to the entries. Then enter the name of the interface to add (wan or mobile in this example) in the lower-left text field and click the *ADD* button next to it.



You will be presented with many options for determining the connection status of the interface. The default is an ordinary ping command which will be executed according to the options specified herein. It is possible to define multiple *Tracking hostnames or IP addresses.* The *Tracking reliability* option defines how many of these addresses need to be pinged successfully for considering the connection working. Advanced settings allow connection quality checks by also evaluating the packet loss.



Google and OpenDNS servers (8.8.8.8, 8.8.4.4 and 208.67.222.222, 208.67.220.220 respectively) have been proven as reliable indicators for a working internet connectivity. Yet these services are not guaranteed to be available forever, nor that being able to ping them means that the internet connection to other servers is working properly. There are alternative test methods, which can be selected from the *Tracking method* dropdown field, each offering their own set of options. E.g. *httping* for checking the reachability of HTTP(S) servers.

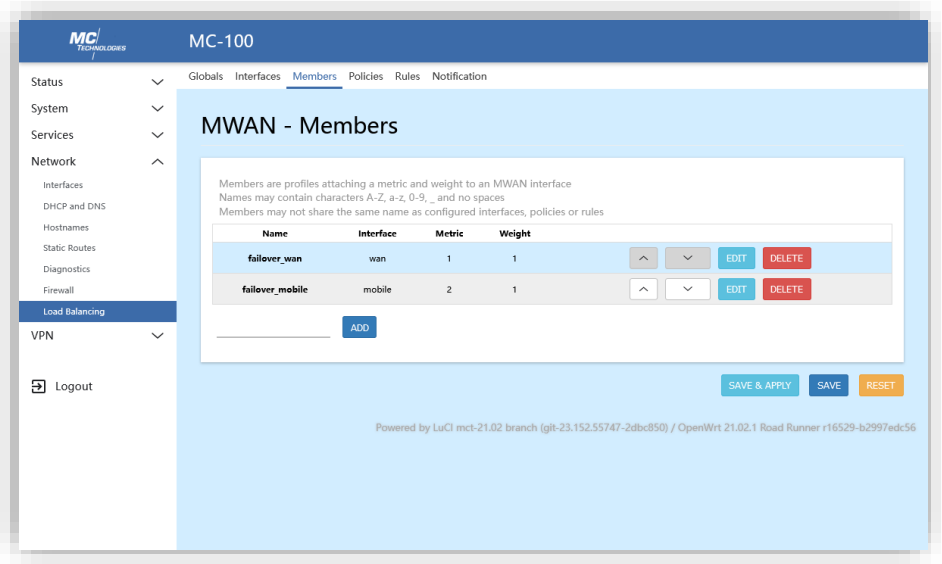
22.2.2 Grouping interfaces using Member profiles

Delete all pre-configured *Member* profiles by clicking the *DELETE* buttons next to the entries.

For grouping interfaces in (multiple) *Policies* later, they need to be added to *Member* profiles first. *Member* profiles basically have two options:

The *Metric* prioritizes the interfaces for a failover case (just as with a routing metric, a lower metric value means the interface is preferred compared to one with a higher value). Please do not confuse this with the default route metric configured earlier.

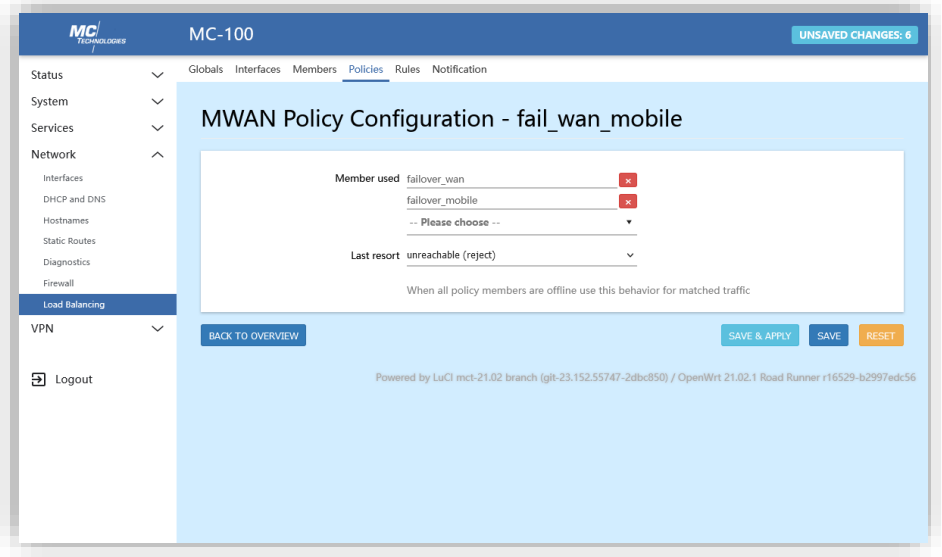
The *Weight* value is used for allowing load balancing of two or more interfaces sharing the same metric value. E.g. considering a policy consisting of two interfaces A and B, sharing the same metric value, with a weight value of 3 for interface A and 2 for interface B generally means that 60% (3 * 100% / (3 + 2)) of the newly instantiated connections will be using interface A.



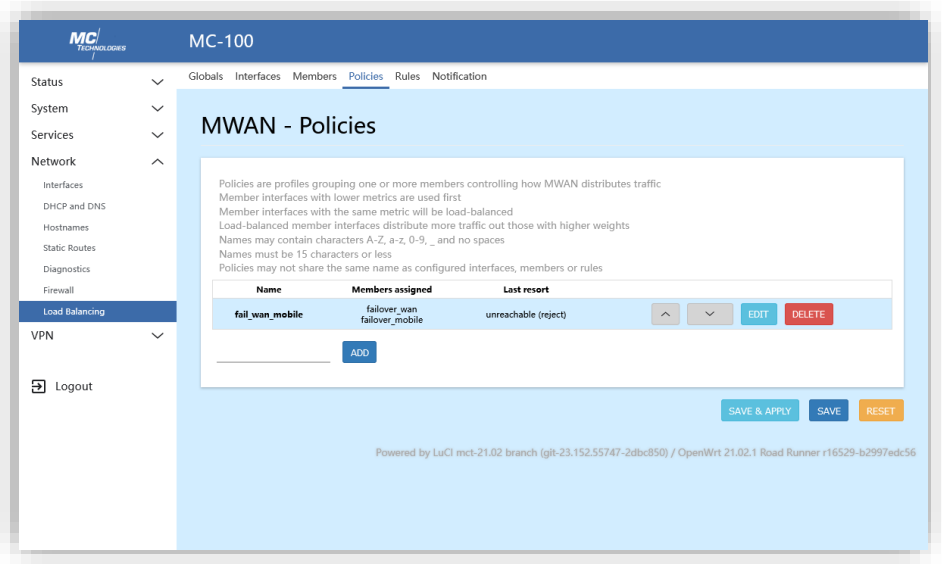
22.2.3 Policies

Delete all pre-configured *Policies* by clicking the *DELETE* buttons next to the entries.

Policies allow the grouping of *Members* for defining failover, load-balanced or even mixed *Rules.*



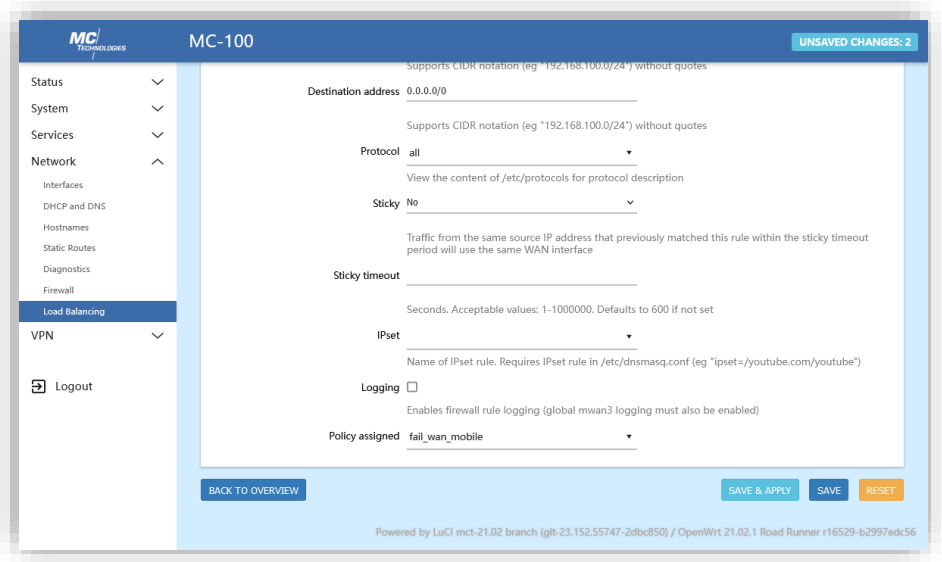
As stated previously, mwan3 checks the connectivity status of the *Members* in the first run ordered by their metric values. If multiple of the functional members with the lowest metric of that policy share the same metric value, only then the weight value is considered for load-balancing the connections.



22.2.4 Rules

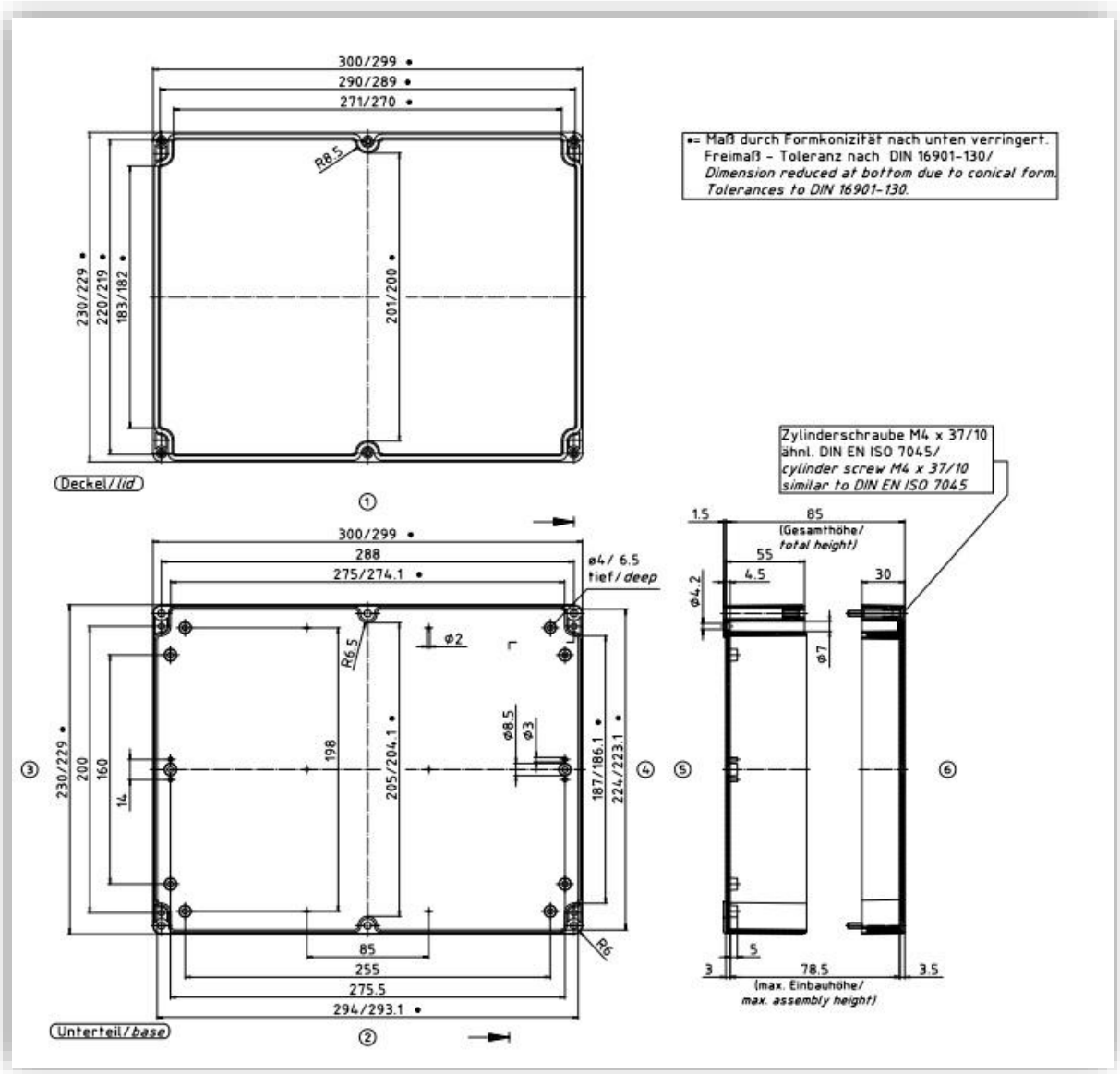
Finally, *Rules* need to be set up for defining the application cases of the policies. Delete all pre-configured *Rule* definitions by clicking the *DELETE* buttons next to the entries.

Rules define for what type of connections a policy shall be applied. Valid criteria are the IP protocols, the Source and Destination address, port range and advanced load balancing settings. It is even possible to define a custom *IPSet* for dynamic setups.



24 Dimensions

24.1 SensorBox



25 Product care and handling

25.1 Maintenance

The product is maintenance-free and requires no special regular maintenance. The device may however require regular inspection (see chapter *Electric safety requirements*).

25.2 Troubleshooting

If a fault occurs during operation of the product and you need assistance, please contact MC Technologies support. You can reach our support department by email or phone:

support@mc-technologies.com
+49-511-676 999-126

25.3 Repair

Only qualified personnel at MC Technologies GmbH are authorised to perform repairs.

Send defective products with a detailed error description to:

MC Technologies
-Repair-
Kabelkamp 2
30179 Hannover

Before shipping the device make sure to:

- call our support team and ask for an RMA number (Return to Manufacturer Authorisation)
- remove any personal belongings like inserted SIM cards
- back up any relevant data like configurations on the device

25.4 Disposal

In accordance with WEEE regulations, the return and recycling of old MC Technologies equipment for our customers is regulated as follows:

Please send your old devices carriage paid to the following address:

MC Technologies
-Disposal-
Kabelkamp 2
30179 Hannover