© W. Waag, Munich, 2020-2024

Operating manual
CAN-Gateway software

SW Version 30.020 / Android+Windows App 1.1.1

Update 1 (14.09.2024)

Table of contents

1 Important notes – please read carefully!

The German version of the operating instructions is legally binding.

1.1 Intended use of the software

This software is intended to read out and control devices from Hoval AG that are equipped with a TopTronic® E control (hereinafter referred to as TTE devices).

In this sense, reading out means that the parameters and information are read out from TTE devices via the device CAN bus and:

1. can be displayed to the user on the CAN-Gateway user interface,
2. Can be transmitted to other devices compatible with these communication protocols using the communication protocols available in the software (REST API, MQTT, KNX IP (routing), Modbus TCP),
3. Can be saved on a micro SD card.

Control in the above sense means that this software can change the parameters and information that can be changed in TTE devices via the CAN bus. This change can:

1. Can be controlled directly by the user via the user interface or Android app or Windows app

2. Controlled by a device connected to this software via one of the communication protocols listed above.

The CAN-Gateway software does not contain any logic that independently initiates changes to the parameters and information in TTE devices.

The software functionalities described above should enable the TTE devices to be integrated into a home automation system.

Which parameters and information can be read out and / or changed depends on the respective TTE device. The functionality of all parameters listed in the following file from Hoval AG is contractually guaranteed: http://www.hoval.com/misc/TTE/TTE-GW-Modbus-datapoints.xlsx. As a rule, it is also possible to read out all parameters that are displayed via a TTE operating module in the "Service" menu for all user levels, but this is not contractually guaranteed.

The software can be applied by running it on a compatible microcontroller. ESP32 dual-core microcontrollers from Espressif (https://www.espressif.com/) are compatible. In addition, a compatible wiring of the microcontroller must be available. Among other things, this wiring must enable the necessary power supply for the microcontroller as well as the connection of the physical interfaces such as CAN bus and Ethernet. To use the SD card functionality, an SD card slot must be connected to the microcontroller. The wiring must be based on one of the three reference examples:

a. CAN gateway hardware V5, as described in the operating instructions
b. CAN Gateway Development Board, see http://wled.shop/wp-content/uploads/2021/11/CAN_Gateway_Nutzungsinformationen.pdf
c. Olimex ESP32-EVB (-EA) development board Rev. I, see https://www.olimex.com/Products/IoT/ESP32/ESP32-EVB/open-source-hardware

According to these reference examples, the software is offered in three variants: a, b and c. When ordering, the customer must choose one of the three options.

For performance reasons, it is possible to use/activate only one communication protocol (MQTT, KNX IP, REST-API) or data storage on an SD card at the same time. Although parallel use is theoretically possible, it can lead to performance problems and is therefore not guaranteed.

1.2 Hazard warnings

WARNING: This software is based on the analysis of the communication between TTE devices and not on official information from the device manufacturer. It must be assumed that the analysis is incomplete, faulty and cannot be applicable to all devices and their versions, especially future versions. Therefore, the use of this software with safety-critical TTE devices must be supervised. As safety-critical those TTE devices are meant, in which errors in the control can lead to damage to people, the environment, other devices, buildings etc. Supervised means that this device and all TTE devices connected to it must be constantly monitored by a person who, based on their training or experience, is able to recognize errors and, if necessary, switch off all TTE devices within a few seconds. This particularly expressly, but not exclusively, applies to combustion heating appliances.

IMPORTANT NOTE: The Hoval AG itself sells gateways for the connection of its TTE devices to the home automation systems via KNX or ModBus RTU / TCP buses / protocols. The provider of this CAN-

Gateway software recommends productive usage of those gateways from Hoval AG with safety-critical TTE devices.

IMPORTANT NOTE: The usage of this software in connection with TTE devices may lead to the guarantee and warranty claims for these TTE devices being void. The user of this software is responsible for checking the existence of corresponding clauses in the documentation for his TTE device and the legal situation in his country.

WARNING: This software allows changing parameters of the TTE devices. This software does not check whether the change of the parameters is meaningful and harmless for the operation of the TTE device and other associated TTE devices and other systems as well as for users and operating personnel of these TTE devices and systems, also for the environment and the building where these devices and thus connected systems are installed. As a user of the software, you must be very careful with parameter changes and only allow and directly or indirectly control those parameterizations whose purpose you fully understand and whose harmlessness for the operation of the TTE device and other associated TTE devices and other systems as well as for users and operating personnel of these TTE devices and systems, also for the environment and the building where these devices and thus connected systems are installed.

WARNING: If you use the microcontroller on which this software is installed in your WiFi or LAN network, you must ensure that no unauthorized people gain access to this network, as everyone with access to your network can access the CAN-Gateway Software without restrictions.

WARNING: The microcontroller on which this software is installed must be physically inaccessible to unauthorized persons. Everyone who has physical access to the microcontroller can easily read out the information stored on it, including your settings and passwords.

IMPORTANT NOTE: Any use of the CAN-Gateway software other than described above is not intended and leads to an exclusion of warranty and liability.

IMPORTANT NOTE: Every change and modification of the software by the user leads to a warranty and liability exclusion of the software provider.

1.3 Liability of the software producer and the provider

The software producer and the software provider are liable for errors in the CAN-Gateway software in accordance with the statutory provisions of the German sales law (§§ 434 ff. BGB). In the event of slight negligence, the software producer and the software provider are only liable in the event of a breach of essential contractual obligations (cardinal obligations) as well as in the event of personal injury in accordance with the German Product Liability Act (“Produkthaftungsgesetz”). Incidentally, the pre-contractual, contractual and non-contractual liability of the software producer and the software provider is limited to intent and gross negligence, whereby the limitation of liability also in the case of the fault of a vicarious agent of software producer or provider applies.

The liability of the software producer and the software provider is completely excluded in the following cases:

1. An error and the resulting direct and / or indirect damages or injuries are caused by a defective execution of the software, e.g. due to defective hardware or due to external influences on the hardware (mechanical, electrical, electromagnetic or other type).
2. An error and the resulting direct and / or indirect damages or injuries are caused by incorrect usage of the software, including, for example, the initiation of inappropriate, incorrect or even dangerous parameter changes by the user. This includes direct control through the user interface, via Android or Windows app, or indirect control in cases where the user has programmed or set other devices in such a way that they cause the parameters to be changed using a communication protocol via CAN-Gateway software.
3. An error and the resulting direct and / or indirect damages or injuries are caused by other reasons for which the software producer and software provider are not responsible.

1.4 Binding of the software to a MAC address specified by the customer

This software, with the exception of the demo version, is linked to a MAC address of the WiFi module of the ESP32 microcontroller specified by the customer. This means that the software has its full range of functions only on the microcontroller with this MAC address.

The customer has the right to transfer the software to another microcontroller with a different MAC address (hereinafter referred to as transfer). To do this, he must contact the software provider by email, telephone or post and request the transfer. The MAC address of the old microcontroller and the MAC address of the new microcontroller (of their WiFi module) must be provided by the customer. The software provider will process the request within 10 working days. After commissioning the software on the new microcontroller, the customer must completely delete the software on the old microcontroller as well as any backup copies of the software that he may still have.

If a customer requests the transfer of the software often than once a year, he must provide a plausible explanation. If there is justified suspicion of misuse and infringement of the software provider's copyrights, the software provider can refuse to transfer the software.

© W. Waag, Munich, 2020-2024

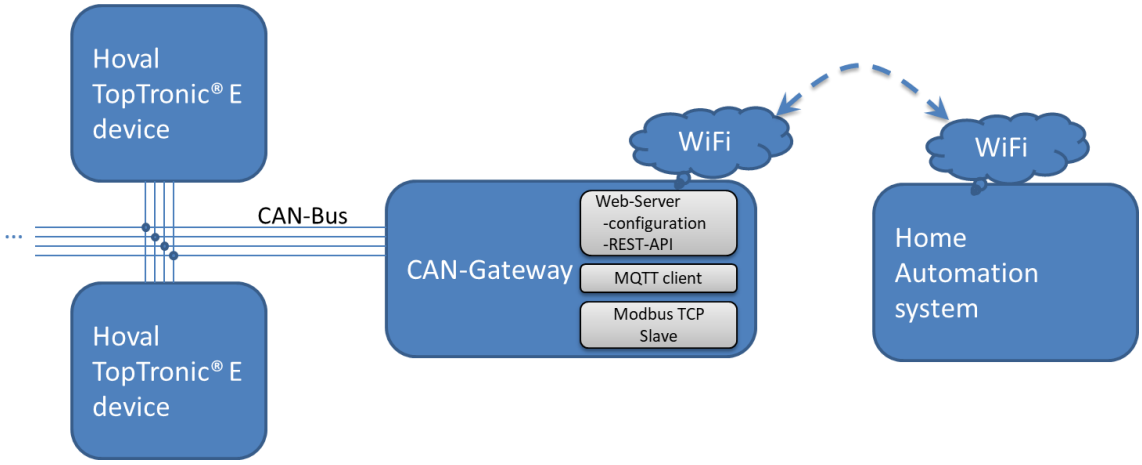
2 Introduction

This software make it possible to connect devices from series „TopTronic® E“ (TopTronic® E is the registered trademark of Hoval AG company) manufactured by **Hoval AG** to the home automation system. Such devices from series „TopTronic® E“ manufactured by **Hoval AG** are called Devices in the following.

These Devices have a CAN-Bus connection. It is used for intercommunication between devices, for example if many of them are installed together. CAN-Gateway can be connected to this CAN-Bus and can read the data that are sent over the bus. It also can generate data („write” the bus). This way it is possible to read information form the Devices and also control them for test reasons.

This software can be installed on an ESP32 DevKit board. The detailed specification of the required hardware and the whole assembly is given in the next section of this document. The whole assembly with installed software is called CAN-Gateway in the following.

ESP32 DevKit board has a WiFi interface. Using this interface CAN-Gateway can communicate to the home automation system (such as Home Assistance, OpenHab, ioBroker, Domoticz). As communication protocols MQTT, Modbus TCP or REST-API can be used. MQTT protocol is recommended and Modbus TCP protocol has some limitations. CAN-Gateway can be configured over the WiFi interface using an internet browser (Web interface).



Additionally the software has following functions:

- Log can data sent on the bus and show them in the Web interface
- Detect devices that communicate on the can bus and list them in the Web interface
- List all parameters communicated through the can bus

3 Required hardware and its assembly

There are various options:

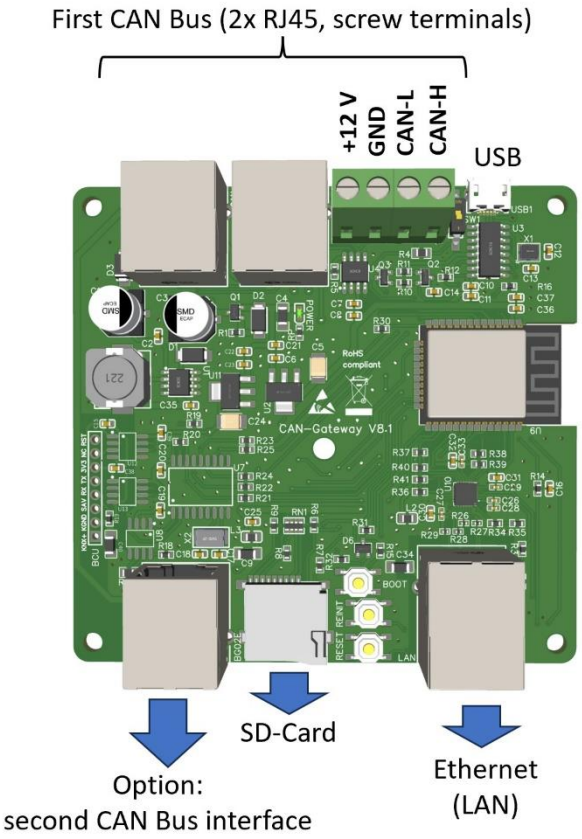
Option	Recom-mended?	Total hardware costs (including shipping to Germany, prices include shipping, as of February 15, 2022, prices may vary)	Advantages / Disadvantages
CAN-Gateway development board from https://wled.shop	Yes	approx. 52 €	Fully built and tested. Available throughout the EU.
CAN-Gateway Basis-Board (without uC) + Enclosure (https://shop.myhome-control.de) + ESP32 DevKit board (various offer on the market available)	Yes	approx. 44 €	Easy assembly (plugging in, screwing; no soldering etc.), interfaces are tested for functionality; BG02E interface is offered as an equipped option. Only available in Germany.
Olimex ESP32-EVB(-EA) development Board incl. enclosure	With restric-tions	approx. 50 €	No BG02E interface possible, CAN connection: only screw terminals. No power supply possible via Hoval CAN bus. Available as a version with an external antenna. Deliverable worldwide.
Do-It-Yourself made of prefabricated modules available on the market, reference design V5	No	approx. 15 € to 30 €	Soldering etc. necessary, deep understanding of electronics necessary, no Ethernet interface possible.

3.1 CAN-Gateway development board (recommended)

Please refer to http://wled.shop/wp-content/uploads/2021/11/CAN_Gateway_Nutzungsinformationen.pdf

This is the recommended option. The CAN-Gateway software can be installed on this board very easily with just a few clicks and without special knowledge. The board has Ethernet interface, "plug-and-play" CAN RJ45 interfaces, micro-SD slot, USB interface, enclosure, DIP switch for CAN termination. No disadvantages. **Attention**! The additional CAN interface on this board is not equipped as standard: a special BG02E interface is only possible with additional components soldered!

© W. Waag, Munich, 2020-2024



3.2 CAN-Gateway Basis-Board (without uC)

Please refer to https://shop.myhome-control.de/CAN-Gateway-Basis-Board-ohne-ESP32-uC/HW10001.2

This is another recommended option. In addition, an ESP32 DevKit Board (30-pin version, various suppliers) is required. The board has an Ethernet interface, "Plug-and-Play" CAN RJ45 interfaces, micro-SD slot, USB socket (only for power supply, no data, data interface via ESP32 DevKit Board!), housing, DIP switch for CAN termination. Additional equipment for the second CAN interface is available for a surcharge (only relevant for BG02E interface).

The SW variant is the same as for the CAN gateway development board from wled.shop.

3.3 Simple Do-It-Yourself hardware based on hardware modules available on the market

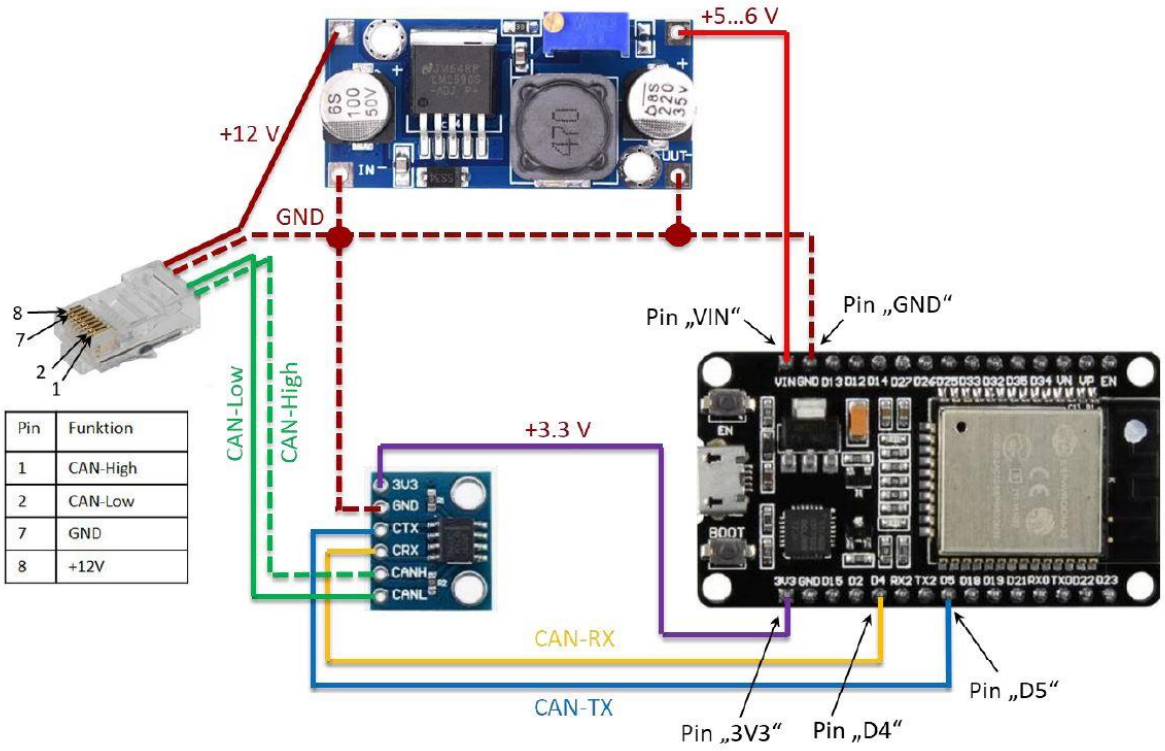
The following parts are required:

© W. Waag, Munich, 2020-2024

Part	Costs, approximately incl. shipping	Picture of the part (exemplarily)
ESP32 DevKit board with ESP-WROOM-32 module (not ESP32S !) and with 4MB Flash In the following 30-pin version is used. Other versions (for example 36-pin version) generally also can be sued, but have another pinning.	5-10 €	
SN65HVD230 CAN transceiver board (Attention: many China-parts are fake und does not work stable)	1,5-5 €	
LM2596 module for supply voltage (as 12 V to 5 V converter)	1,2-5 €	
Ethernet cable with RJ45 connector	2-5 €	

Attention! Ethernet cable is used, but it is for CAN bus and it has nothing common with a normal LAN communication. The CAN-Gateway should not be connected to a PC/laptop or router via Ethernet cable!

The assembly is as follows. Attention: LM2596 module must be adjusted first to 5V output voltage. This is done by the screw in the adjustable resistance (light blue on the picture below) and measuring the output voltage by a mustimeter. ESP32 module can work for a short time at 12 V, but gets hot over the time and can be damaged.



Pin	Funktion
1	CAN-High
2	CAN-Low
7	GND
8	+12V

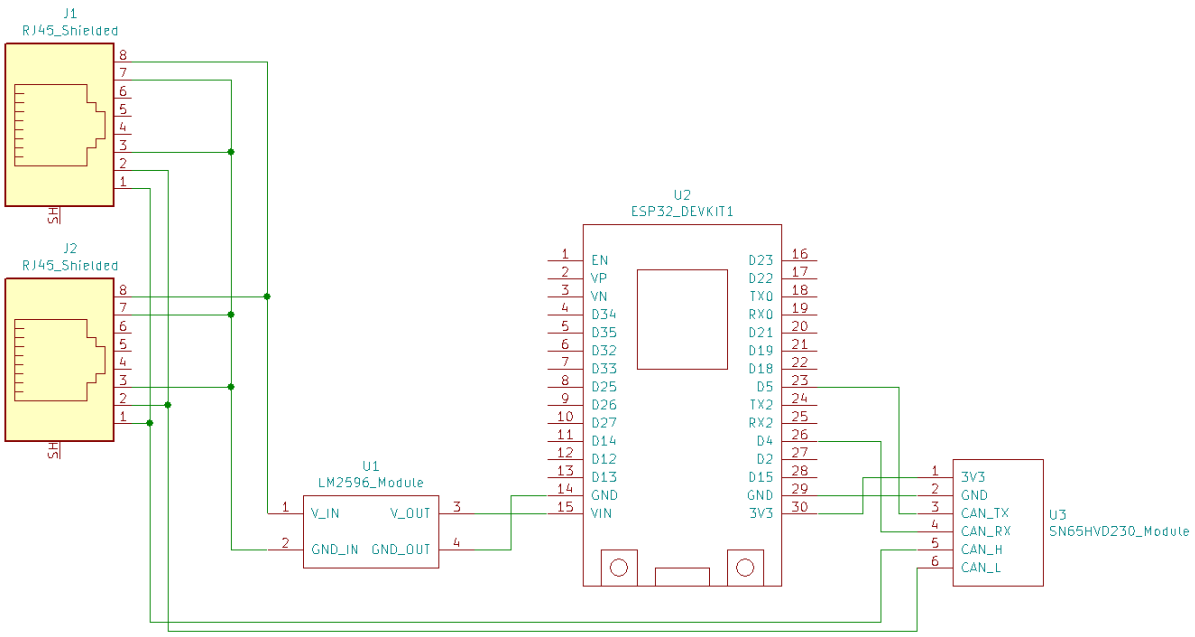
Important notes:

1) GND pin at RJ45: some devices use probably pin 3 instead of pin 7 or both. I recommend interconnect both pins (3 and 7) together to the ground.
2) CAN bus: CAN bus must be terminated on both ends. The devices from Hoval have termination resistances. They also have dip switches to switch the termination resistances on and off (for example, room control panel has them on the back side. Some devices have them close to the CAN connector, these dip-switches usually labelled as “R-CAN”). SN65VHD230 board has also termination resistance permanently installed. If you have only one Hoval device and can bus is not used, you can switch on the can-bus termination in your device and connect CAN-Gateway that is also terminated. If your installation already uses CAN bus (for example if you have ventilation device and room control unit), then your CAN bus is already fully terminated. You have to switch off one of the termination resistances using dip switches. If you are unclear regarding what you already have, you should use a multimeter to measure the resistance between CAN-high and CAN-low pins (devices must be switched off during this measurement!). The resistance must be approximately 60 Ohm in a full can bus system (including connected CAN-Gateway). If you measure resistance significantly lower than 60 Ohm, you CAN bus is “overterminated” and you have to switch off some termination resistances.). If you measure resistance significantly higher than 60 Ohm, you CAN bus is “underterminated” and you have to switch on some termination resistances. For more details regarding CAN bus termination please refer to the respective information you can find in the internet (for example on Wikipedia).

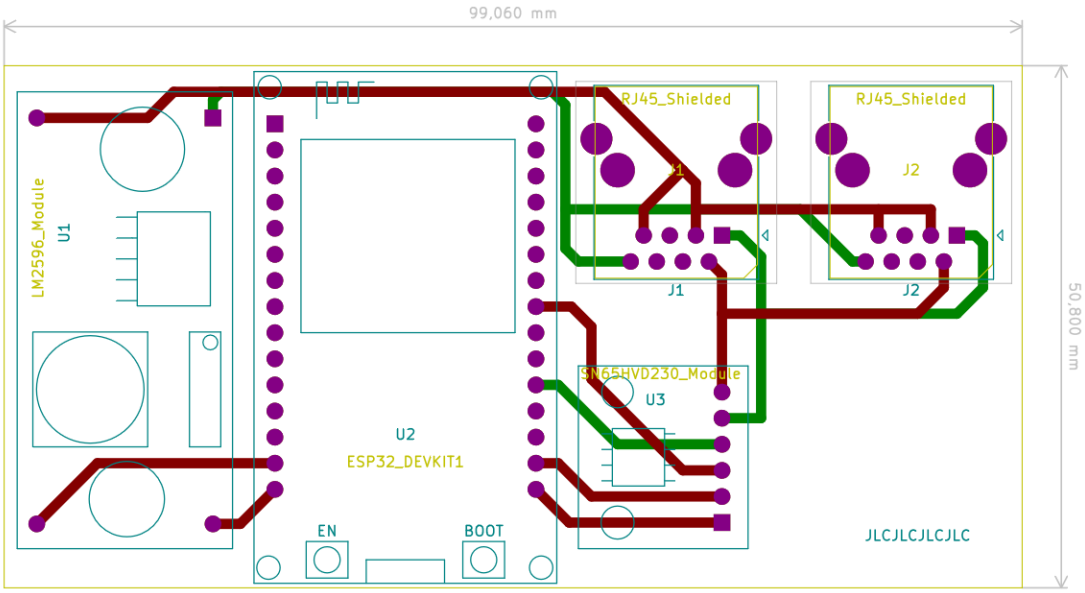
© W. Waag, Munich, 2020-2024

There is an example of CAN-Gateway board with two RJ45 connectors. Two connectors can be used to integrate CAN-Gateway “in the middle” of your existing CAN bus system.

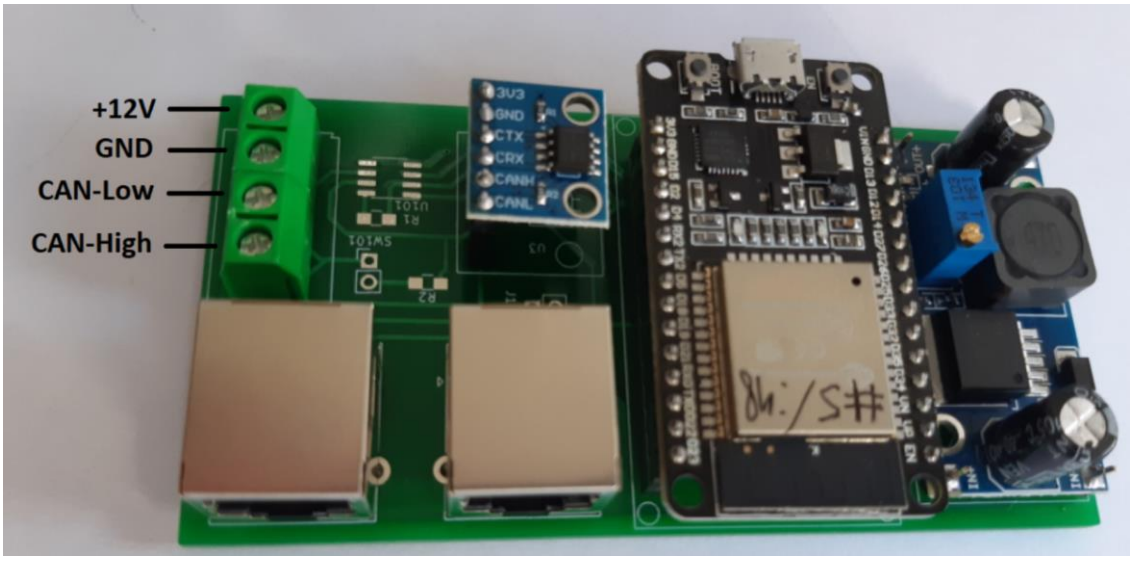
3) Power supply: you can supply the CAN gateway in two alternative ways:
    a. Via the 12 V wire of the Hoval CAN buses (recommended). All Hoval main control units provide 12 V. It is used, for example, to supply the room control units. The CAN gateway can also be supplied from this 12 V wire and does not consume more power than one room control unit. Advantage: no additional power supply required.
    b. Via the micro USB connection of the ESP32 board (optionally). If you already have many room control units that are supplied from a 12 V CAN bus line (please check Hoval documentation how many units can be supplied), you can supply the CAN gateway via the micro USB by a power supply (5 V / 0.5 A is sufficient, but please use high quality parts). In this case, the 12 V wire to Hoval devices must not be used, e.g. the corresponding wire in the RJ-45 cable must be disconnected. The GND wire must remain connected (in addition to CAN high and CAN low, of course).



© W. Waag, Munich, 2020-2024



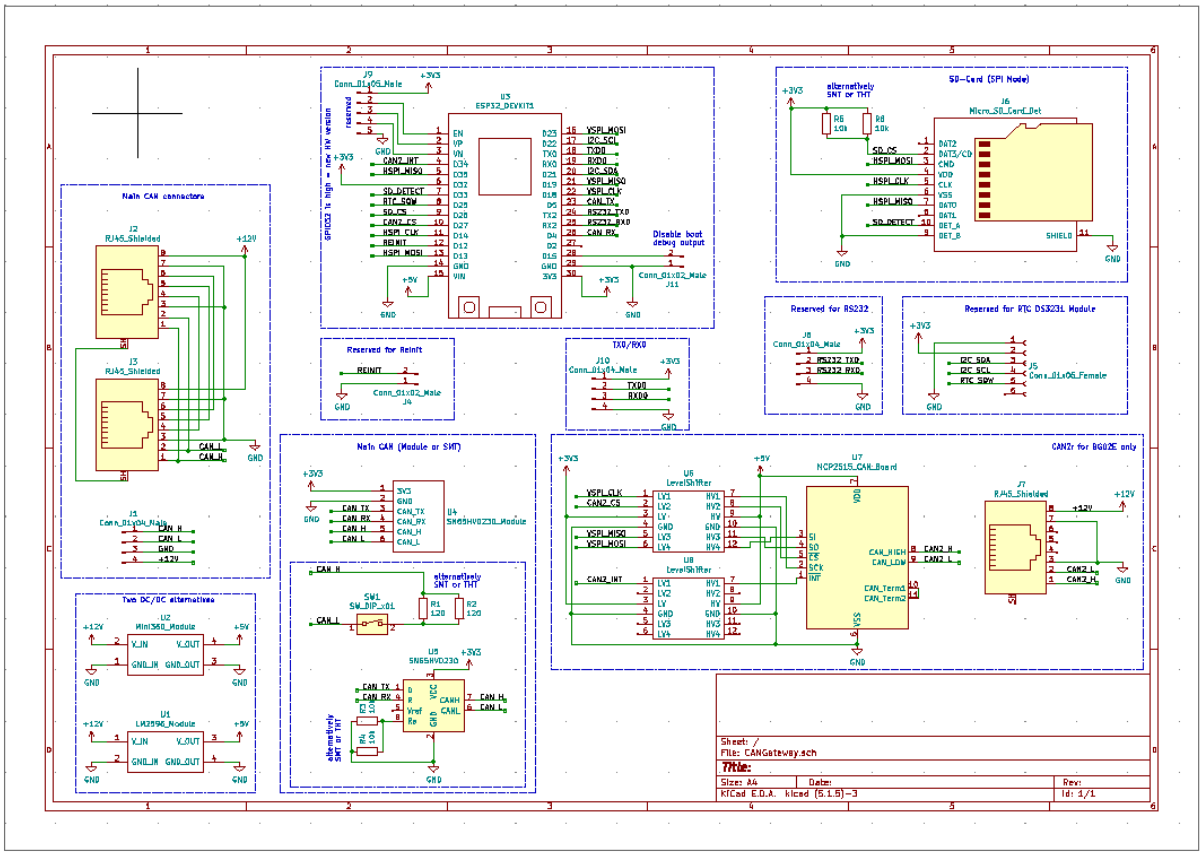
Example of a board with an additional screw connector:

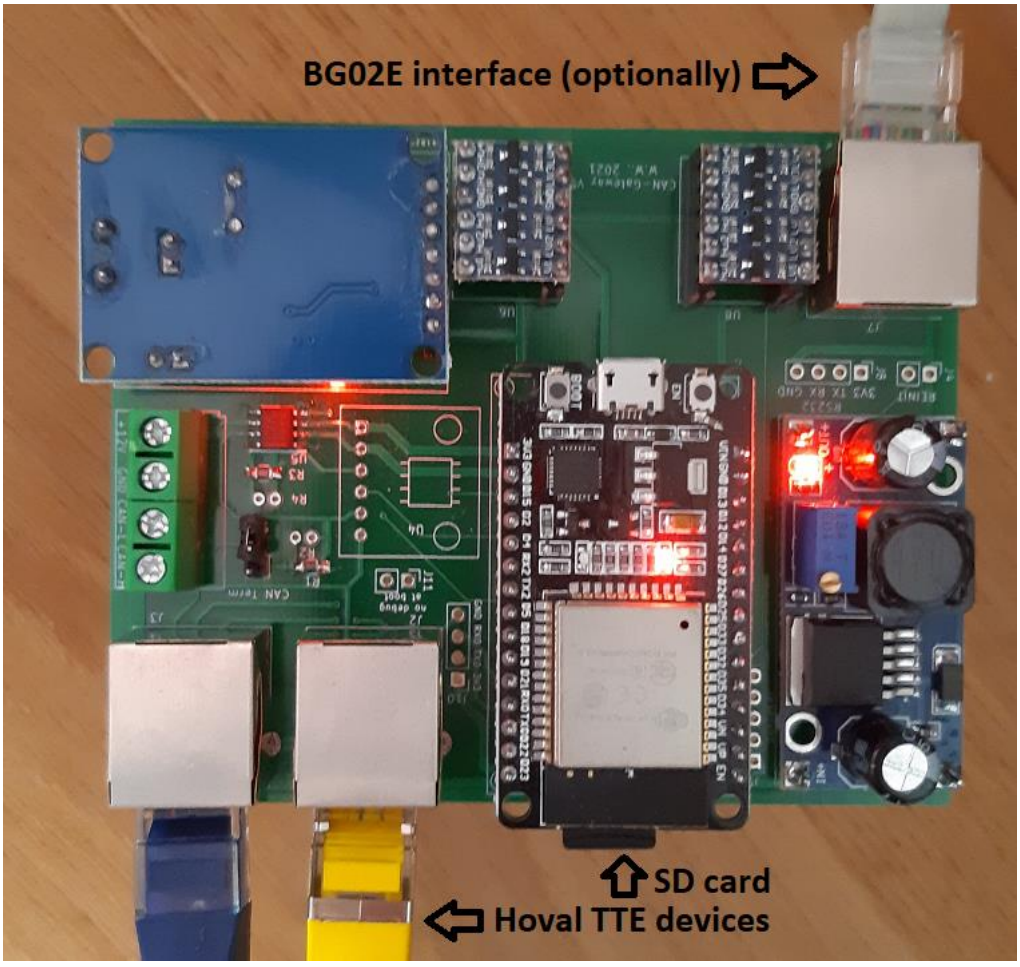


3.4 HW Version 5

HW version 5 has a SD card slot and optional BG02E Interface:

© W. Waag, Munich, 2020-2024





3.5 Use of the ESP32-EVB (-EA) development boards from Olimex as a CAN gateway

Company www.olimex.com offers a development board ESP-32-EVB (-EA), which can be used as CAN-Gateway hardware [**new since SW version 28.001**]:

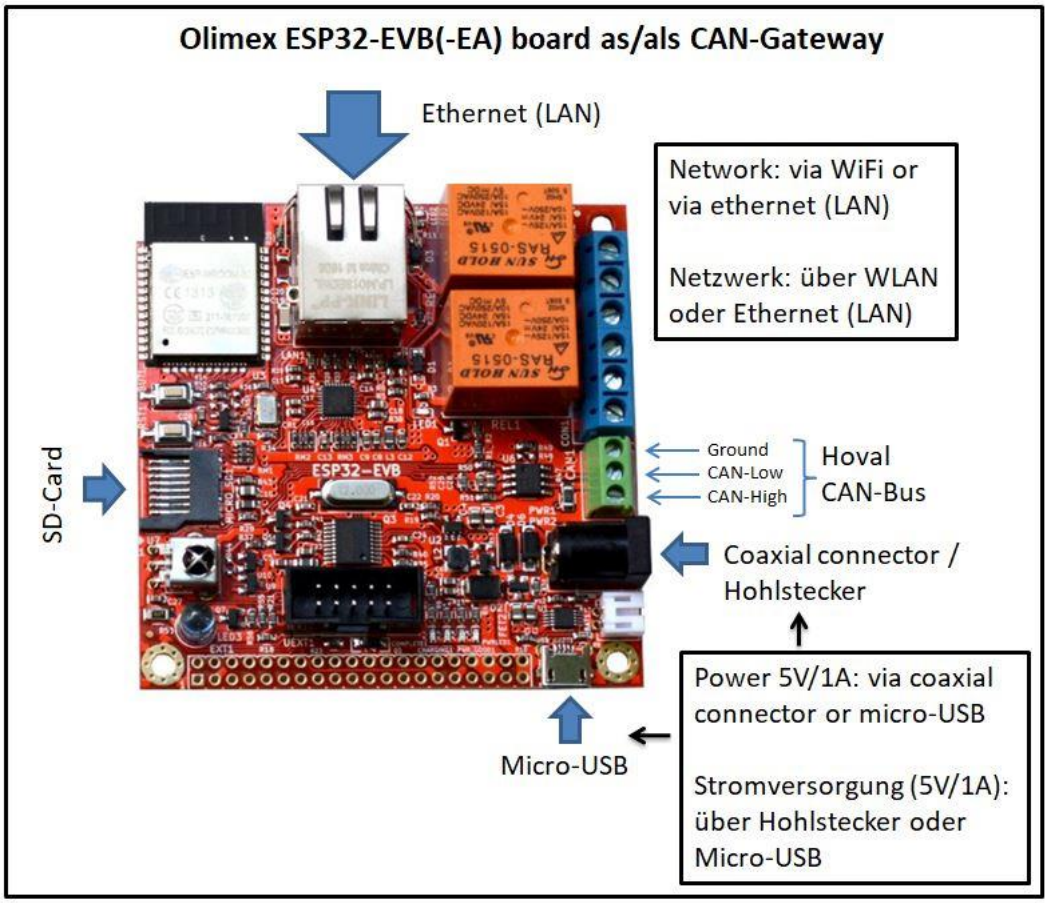
Advantages of the Olimex board compared to CAN-Gateway HW V5:

- Ethernet (LAN) as an alternative to WLAN (initial configuration always takes place via WLAN, but you can then switch to Ethernet / LAN)
- -EA version with external WLAN antenna
- Metallic housing is available from Olimex for the -EA version

Disadvantages of the ESP32-EVB (-EA) board compared to CAN-Gateway HW V5:

- No RJ45 plugs for Hoval CAN bus, only screw terminals
- SD card: no automatic mount/unmount when inserting / ejecting during operation
- External power supply required, no power supply via Hoval CAN bus
- No BG02E interface option

Interfaces:



4 How to flash the software

4.1 Automated (recommended)

The DEMO version of the software can be easily installed via the web-based installer: https://wladwnt.github.io The DEMO version can later be updated to the purchased full version via the WEB interface of the CAN-Gateway.

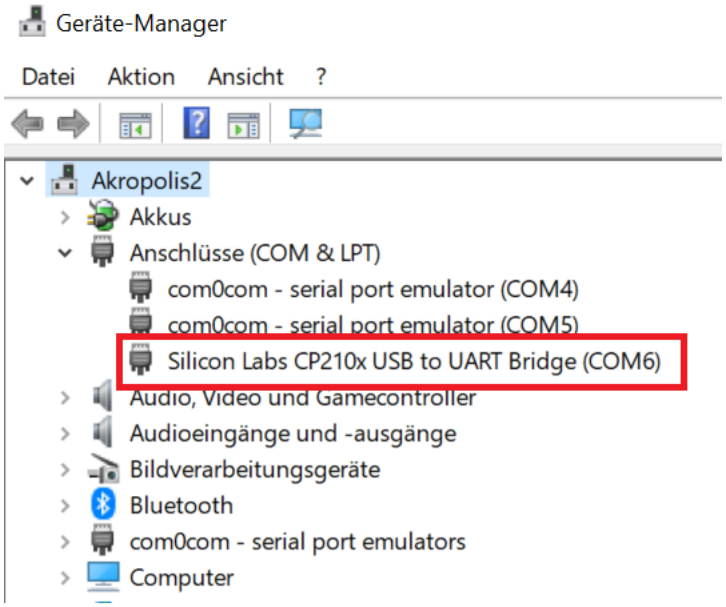
4.2 Automated (recommended), only for CAN-Gateway development board

CAN-Gateway development board from wled.shop is delivered with software that allows other software to be easily flashed on it via the web interface. This way the CAN-Gateway demo or full version can be easily installed with just a few clicks.

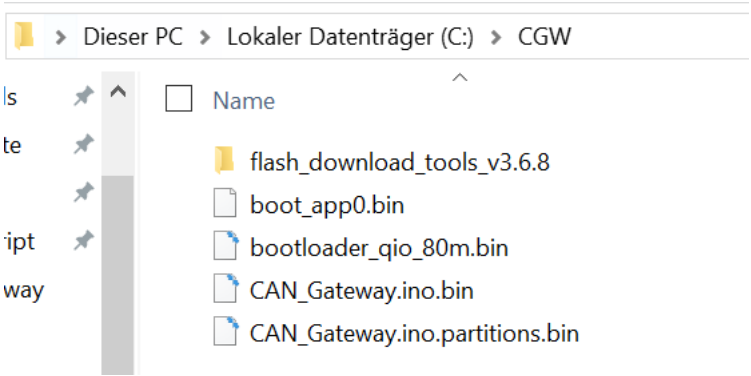
Manuell

The software is provided as a set of binaries und must be first installed on ESP32 based board. For that you have to connect the ESP32 board to the PC/notebook via USB cable and install the driver for the USB communication chip first. In total the following steps are required (assumed you use Windows 10):

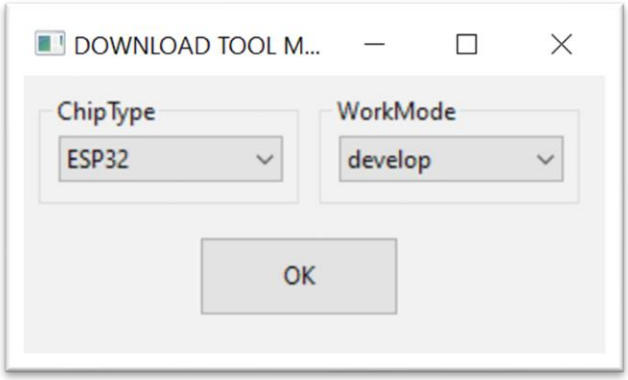
1) Find out the required USB driver. The major of ESP32 DevKit boards use CP210x USB communication chip. The driver for this chip can be found there: https://www.silabs.com/products/development-tools/software/usb-to-uart-bridge-vcp-drivers
   Olimex boards and some other use CH340 USB communication chip. The driver for this chip can be found there: http://www.wch-ic.com/downloads/CH341SER_EXE.html
2) Download the driver and install on your PC/notebook.
3) Restart your PC/laptop.
4) Check what COM-ports are already used in your PC/notebook. In Windows 10 do the following:
   a. Open the Device Manager (right click on Windows symbol on your desktop and then on Device Manager).
   b. In the Device Manager go to Ports (COM & LPT) and check the list.
5) Connect ESP32 DevKit board to the PC/notebook. A new virtual CAN-port will be set up and appear in the list. Remember its number.



6) Make a new folder on you C: drive. For example **C:\CGW**.
7) Download and unzip the ESP Download Tool in this folder. (https://www.espressif.com/en/support/download/other-tools, then under „Flash Download Tools“)
8) Unzip the CAN-Gateway software (files „CAN_Gateway.ino.bin” (or „CAN_Gateway_DEMO.ino.bin” for DEMO Version) in this folder too.
9) You will also need files (up to SW Version 29.101) „boot_app0.bin” (Source: https://github.com/espressif/arduino-esp32/blob/idf-release/v3.3/tools/partitions/boot_app0.bin), bootloader_qio_80m.bin (Source: https://github.com/espressif/arduino-esp32/tree/idf-release/v3.3/tools/sdk/bin/bootloader_qio_80m.bin) and CAN_Gateway.ino.partitions.bin (source: https://github.com/wladwnt/wladwnt.github.io/blob/main/cangw/CAN_Gateway.ino.partitions.bin). Download and save them in the folder above.
   Starting with SW Version 29.201 the respective files are: https://github.com/wladwnt/wladwnt.github.io/blob/main/cangw/boot_app0_v202.bin, https://github.com/wladwnt/wladwnt.github.io/blob/main/cangw/CAN_Gateway.ino.partitions_v202.bin and https://github.com/wladwnt/wladwnt.github.io/blob/main/cangw/CAN_Gateway.ino.bootloader_v202.bin.
10) Now you must have the folder like this:

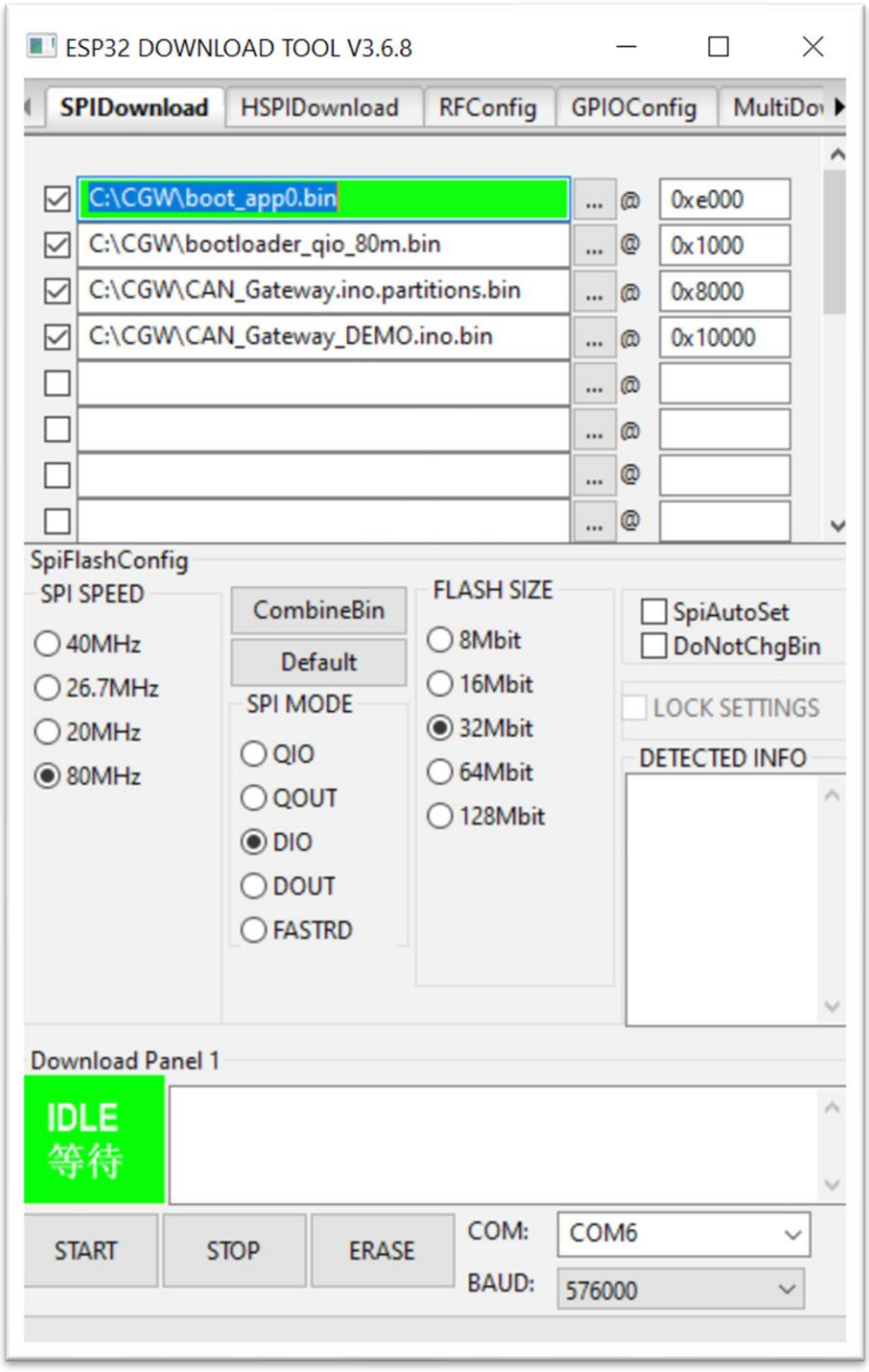


11) Execute ESP32 Download Tool (.exe from the folder flash_download_tool_vX.X.X, where X.X.X is the current version)
12) The following window will appear, please select for ChipType „ESP32” and forWorkMode “develop”.



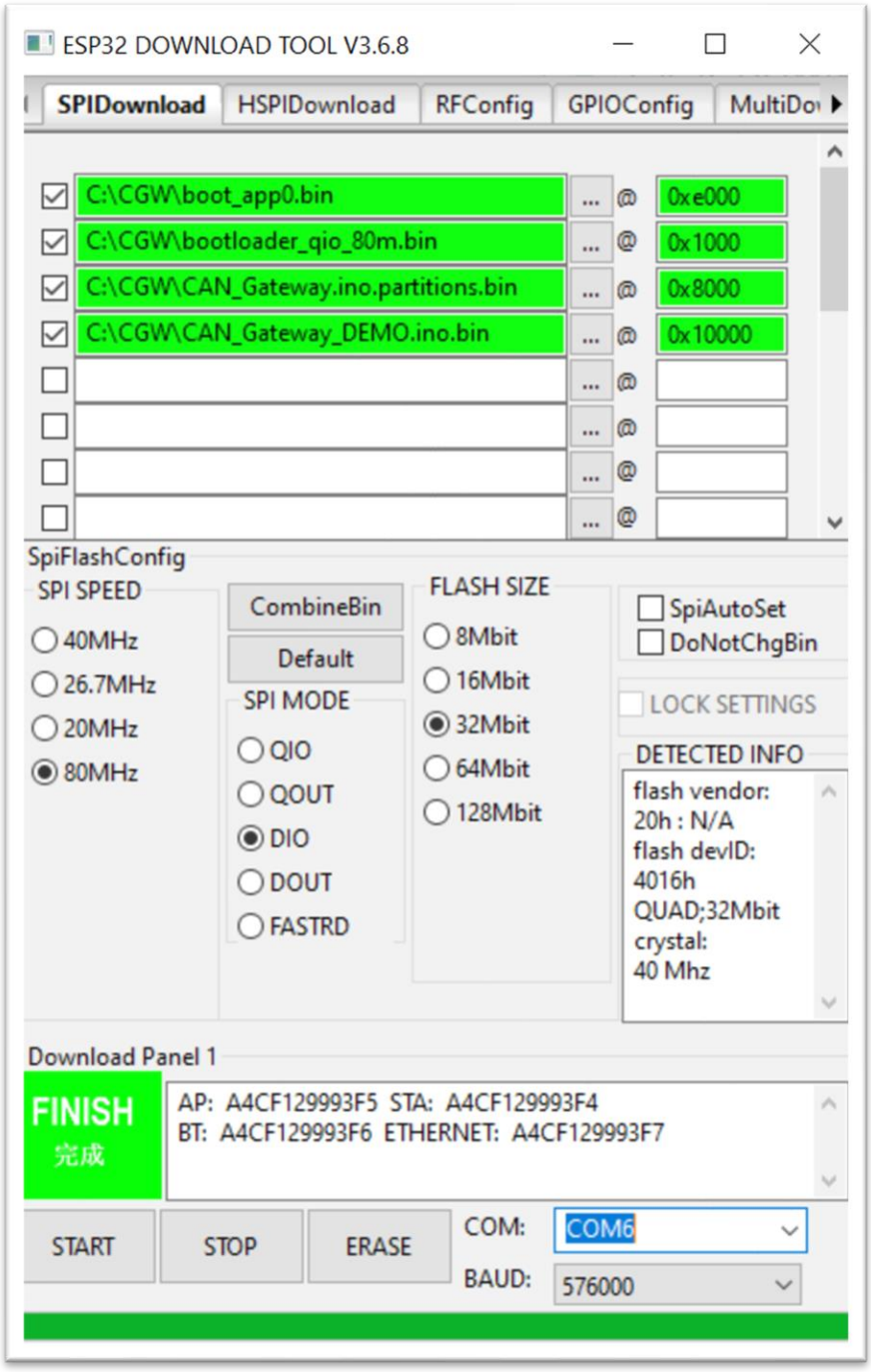
13) Please enter the data into the next dialog window as shown below. The only parameter you have to select is the COM-port number. If you get some issues during flashing process, you also can try to select lower baud rate. Please take care that all check boxes are selected

correctly. Then click on “Start”.



14) As soon as flashing ends, you will see green field „Finish“. Now you can exit flash tool.

Alternatively to the Flashtool other tools can be used (such as esptool.exe or esptool.py that are parts of the Arduino development environment with installed ESP32 Board extensions).



15) Restart ESP32 DevKit board either repower it via USB or using reset button on the board.

If you use esptool.exe for flashing, please use the following parameters and do not forget to use correct COM-Port number:

esptool.exe --chip esp32 --port COM6 --baud 256000 --before default_reset --after hard_reset write_flash -z --flash_mode dio --flash_freq 80m --flash_size detect 0xe000 C:\CGW/boot_app0.bin 0x1000 C:\CGW/bootloader_qio_80m.bin 0x10000 C:\CGW/CAN_Gateway.ino.bin 0x8000 C:\CGW/CAN_Gateway.ino.partitions.bin

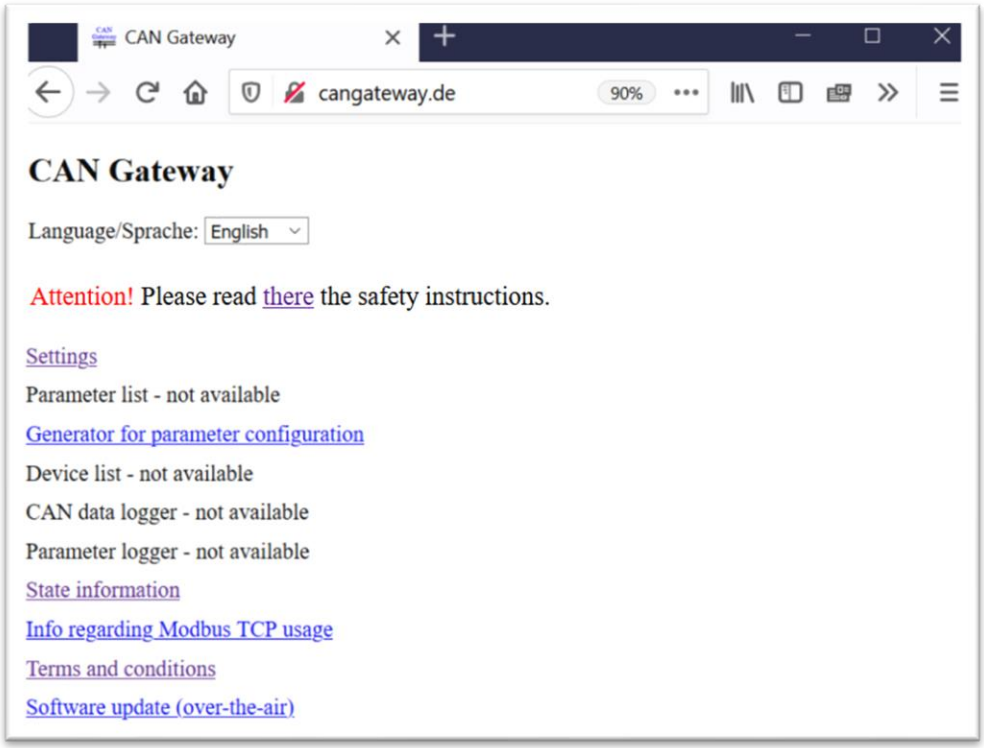
5 First initialisation

If you power on the CAN-Gateway in the initial state, meaning you first time installed the software, it generates a WiFi access point with the name “cangateway”. You have to connect PC/notebook/tablet to this access point to configure the CAN-Gateway. The password is 000999555. In details, the following steps are required (assuming you run Windows 10):

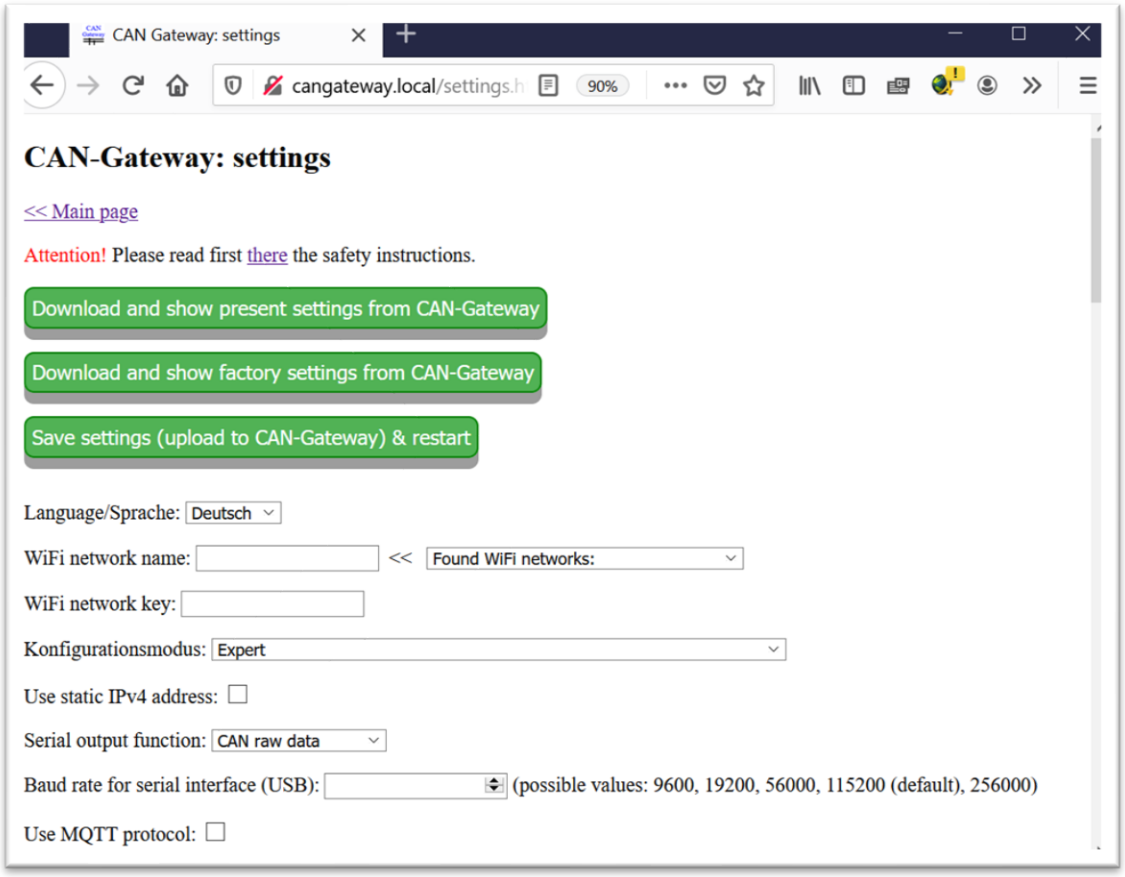
1) Be sure you know your present WiFi name and password. In next steps you will connect your PC to the cangateway access point. Later you will need to connect it again to you home WiFi network. For this you might need to enter you home WiFi password.
2) Click on the desktop on WiFi Symbol. In the widow opened please find network named „cangateway“. Click on it and then on “Connect”.

3) Then you need to enter the network key (000999555).
4) Wait until the connection is established. It might show that there is no internet connection; that is OK.
5) Now you are connected to the CAN-Gateway.
6) Open you WEB browser and go to the address http://cangateway.de. You will see the Web interface of the CAN-Gateway. This web page is not somewhere on an internet server but on your CAN-Gateway directly. At first you can choose between German and English interface language.

© W. Waag, Munich, 2020-2024



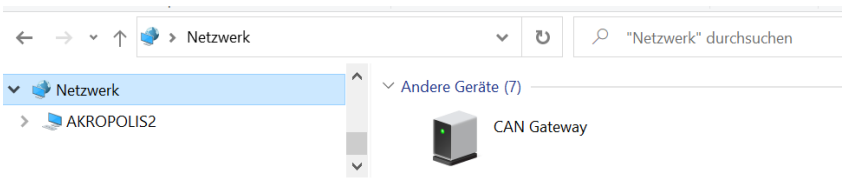
7) Click on „Settings" and read carefully Terms and Conditions. They are exactly the same as described in section 1.2 and 1.3 of this operating manual. If you agree please click on „**I agree to these terms and conditions.**":
8) On the next page click on „Download and show present CAN-Gateway settings". Firstly you can enter your WiFi network name and password. All other settings you can leave as it is. Then click on „Save settings (send them to CAN-Gateway) & restart". Now you can disconnect your PC/notebook/tablet from cangateway network and connect it back to your home WiFi access point. **Attention**: as soon as you change network name from "cangateway" to something different, the cangateway considers it as an existing WiFi network and does not initiates access point any more.



9) After restart the CAN-Gateway will try to connect to your home WiFi network. Your WiFi router must support 2,4 GHz frequency band, 802.11b/g/n standard, WEP/WPA-TKIP/WPA2-CCMP authentication as well as DHCP protocol (all that is typical for modern routers). Be sure that it is also configured to accept new WiFi clients (not to block unknown MAC addresses etc.). Now CAN-Gateway must be connected to your home WiFi network. I recommend restarting your PC/notebook now. It should allow you to reach CAN-Gateway interface under http://cangateway.local using your Web browser (works perfect for example with Firefox, but does not work with Chrome since it try to find this address in google, but this is a local address). CAN-Gateway uses mDNS protocol. Windows 10 identify new mDNS clients only if Bonjour service[1] is installed and only after restart. If you run iOS or Linux, they should support mDNS protocol out-of-the-box. If you run Android or other operating systems, the probably does not support mDNS protocol. If the address http://cangateway.local does not work, you have to find out IP address of the CAN-Gateway using your router. Please refer to your router documentation how to do that. If you have Fritzbox as a router, you can simple try "http://cangateway" (for CAN-Gateway software version 19.001 and newer) or "http://espressif" (for CAN-Gateway software below version 19.001).
[**new in SW 27.400**] CAN-Gateway supports SSDP protocol and is listed in Windows under "Network" section. With the right mouse click on the "CAN Gateway" you will get to "properties" and there you can read the IP address of the CAN Gateway.

[1] How to install bonjour service can be found at: https://softwarekeep.com/help-center/what-is-bonjour-service-on-windows-10. After installation complete, please restart your PC/notebook.

© W. Waag, Munich, 2020-2024

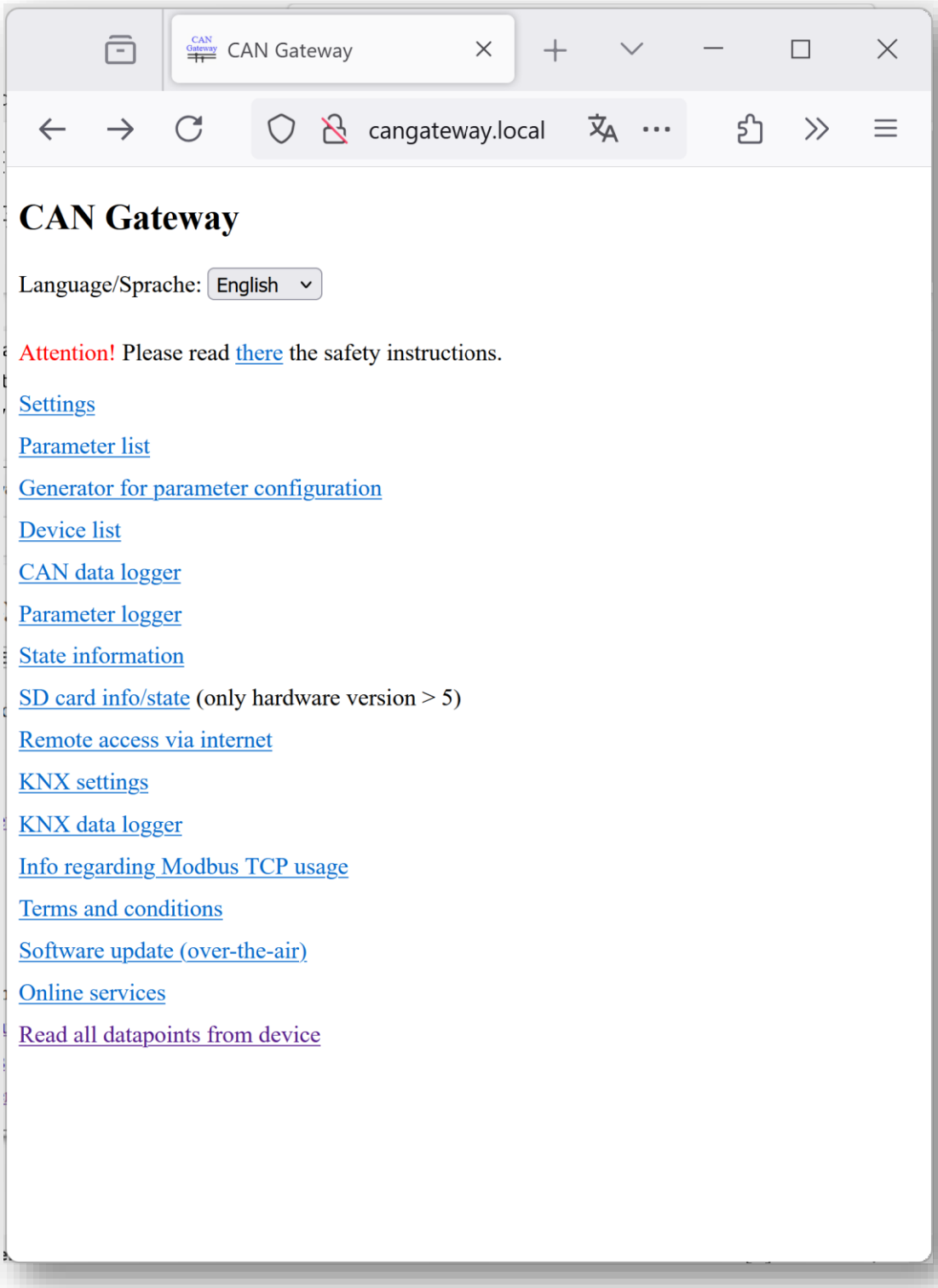


6 Web interface: Main page

CAN-Gateway has a Web interface. It makes possible to use various functions of CAN-Gateway: configure it, change settings, list parameters, list devices found, read raw can data etc. Web interface can be accessed via address http://cangateway.local. The prerequisite is that the operating system on your PC/notebook supports the mDNS protocol (please refer to the previous chapter). Web interface can also be accessed via IPv4 address.

On this page you can also select the language of the CAN-Gateway interface. This selection is only valid until next reboot, it is not stored permanently. To change the language preference permanently, you have to change the respective option in “Settings”.

© W. Waag, Munich, 2020-2024



7 General information about devices and their parameters

Parameters are data that can be read from devices or that can be changed in the devices by a write request to control the respective device. Examples: temperature measured by the device, device name, device mode, set value for the temperature controlled by the device. Different device types have different parameters. Each parameter, also called “datapoint”, has a datapoint ID (that is its unique number), belongs to a certain function that is described by a function number and the function belongs to a certain function group that is also described by a number. Since each device is identified by a device type and device address, each parameter in total can be uniquely described by five numbers: device type, device address, function group, function number and datapoint ID.

- **Device type:** The mapping of device types to the respective device type numbers can be found on the page with CAN-Gateway settings in web interface.
- **Device address:** The device address can be seen, for example, in the room control unit. It also can be changed using room control unit if you have access to the respective user level (regarding user level please refer to the next chapter) or also using this CAN-Gateway software. Usually, however, there is no need to change it. The only reason might be if you want to connect the device of the same type to the CAN bus. All devices of the same device type must have different addresses. The standard address is usually 8. For easily identification of all devices connected to the can bus, the CAN-Gateway analyses the CAN bus communication and list all devices found with their types and addresses. This list is accessible through the CAN-Gateway web interface.
- **Function group**: The function group can be seen in the room control unit, if you have it. Under “Service” all devices available in your installation are listed. For each device different function groups can be seen and in the brackets the respective number. Function group is also given in the Excel file with the list of available parameters provided by Hoval.
- **Function number**: The function number can be seen on the room control unit similar to the function group or also in the Excel file mentioned above.
- **Datapoint ID**: Datapoint IDs can be seen similar to functions numbers on the room control unit or also in the Excel file mentioned above. On the room control unit it is the number shown between the parameter name and parameter value. It is given in the form XX-XXX and displayed in grey. The hyphen must be just ignored, so it is just one number.

All parameters that can be read from devices or set in the devices are generally available on the CAN bus. On the can bus also parameters available, that does not shown on the room control unit at some User Levels.

CAN-Gateway has a parameter logger integrated. It shows in the Web interface all the parameters currently passed over the CAN bus. This function, however, is only meaningful if a room control unit or gateway is installed in the system because parameters are only passed over the CAN bus if some device request for them. Parameters configured in CAN-Gateway are periodically requested by CAN-Gateway and when the devices answer to the respective request, the parameter is also shown in the parameter logger window of the Web interface.

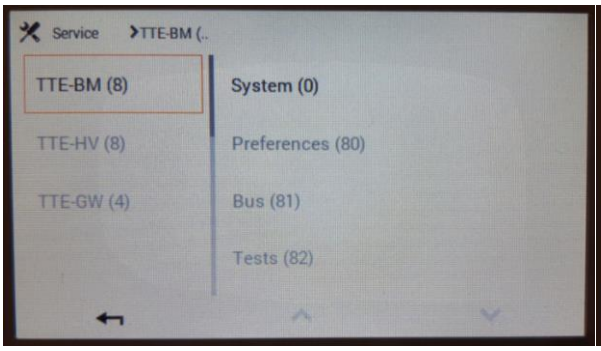
Basically it is possible to request over the CAN bus a list of parameters supported by a certain device. The room control unit, for example, uses this functionality. This is, however, is not supported in the present version of CAN-Gateway.

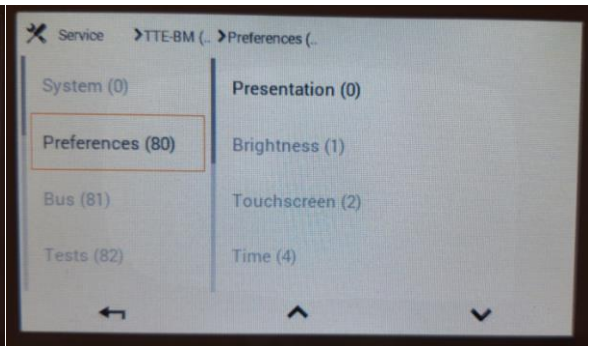
© W. Waag, Munich, 2020-2024

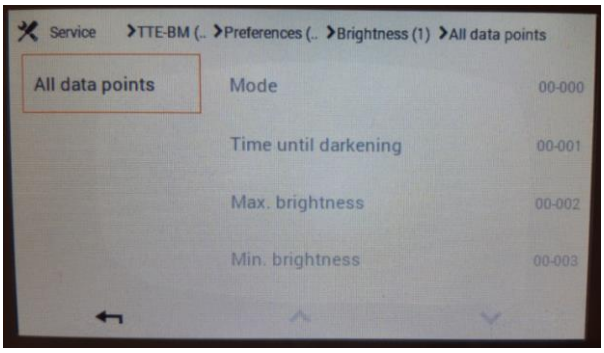
7.1 Example

In the following the parameter is considered that belong to the room control unit itself and defines the time delay until touch screen is darkened after is not touched for a while (“Time until darkening”).

Since the parameter belongs to the room control module (TTE-BM), te respective device type is 16. The device address in this example is 8. The parameter belongs to the function group “Preferences”. This function group has number 80. The function is „Brightness“ with the number=1. Datapoint ID is „00-001“, that means = 1. The present value is 60s.







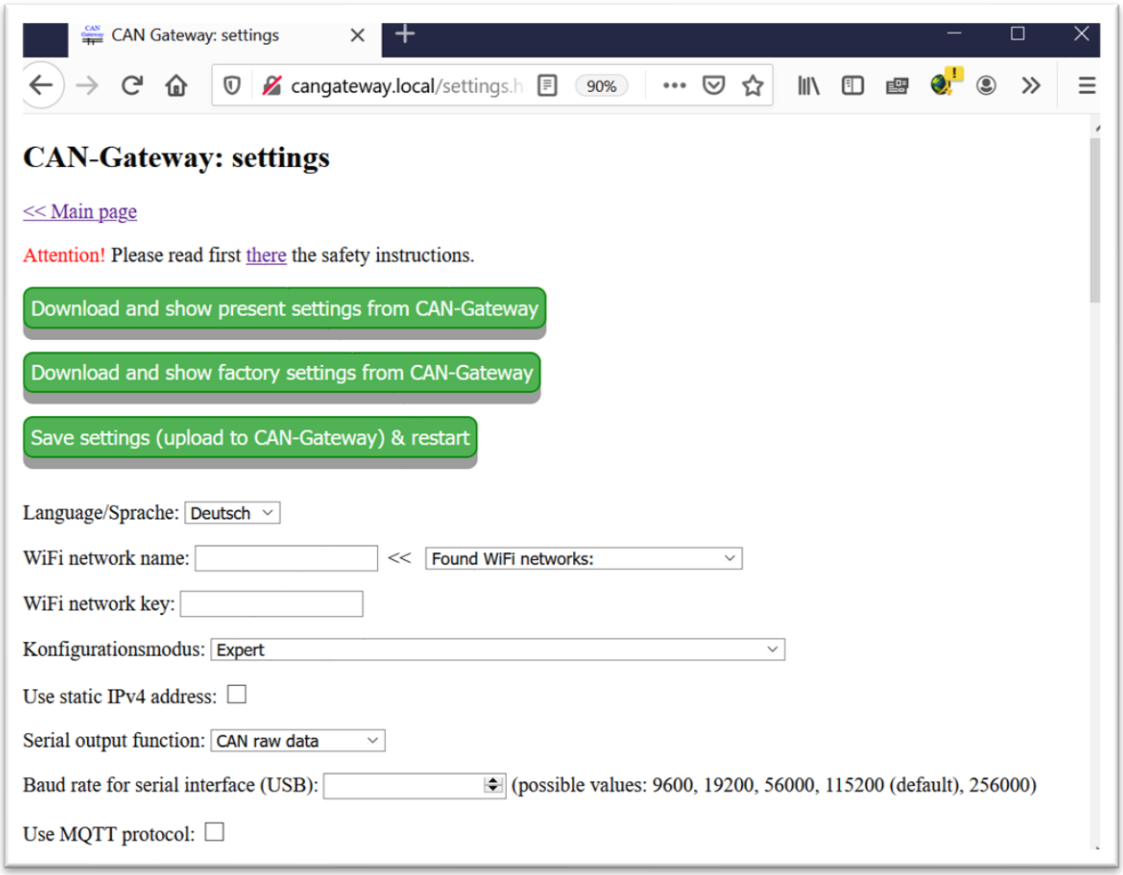
8 User Levels

User Level describes the access level: it defines what parameters the user can read and change using room control unit. In total there are 8 User Levels: 0 to 7. User Level = 0 (default) is the lowest one; User Level = 7 gives full access to all parameters available, also the write permissions for all parameters, even for those that should not be changed at all. Each user level (except the lowest one) is protected with a password. Important to understand, that User Levels are only related to the room control unit. On the CAN bus all parameters are available independent on User Level of the room control unit. That is why CAN-Gateway can read all parameters and even write all parameters (But please take care if you change values!). To mention, CAN-Gateway can work without room control unit. CAN-Gateway can also, for example, read out all User Level passwords from the room control unit.

9 Web interface: General settings

Web interface allows configuring the CAN-Gateway. The following settings are available:

- Web Interface language (German or English)
- [new in SW 28.001] Select WiFi or Ethernet (LAN, only for boards that support Ethernet: currently only Olimex ESP32-EVB(-EA) development boards)
- WiFi network name and network key (password)
- Option to use static IP address for WiFi network (default is the receiving the IP address from DHCP server of your router)
- Function of the serial interface (USB): there are 3 options to select:
    - **CAN raw data** – over the serial interface all can raw data are provided: CAN-ID, DLC, data bytes.
    - **Parameters** - over the serial interface all parameters detected on the CAN bus are provided.
    - **Debug information**: over the serial interface debug information and current status of the CAN-Gateway is provided. It can be helpful for trouble-shooting.
- Baud rate of the serial interface (USB)
- MQTT server IP, server port, user name, password. Must be according the settings in your MQTT broker.
- MQTT prefix. This is the common part for all MQTT topics. Please refer to section “Parameter transmission using MQTT Protocol”.
- [new in SW version 26.001] Support Home Assistant / MQTT Discovery for sensors.
- Identification of the CAN-Gateways on the CAN bus (device type and address).
- Option (on/off) that controls if the CAN-Gateway is shown in room control unit, in the list of devices available in the system and if CAN-Gateway can be controlled via the room control unit (please refer to section „Control CAN-Gateway using room control unit“).
- [new in SW version 22.001] Time server to get the current time information over NTP protocol.
- [new in SW version 22.001] UTC Offset for the local time in seconds.
- [new in SW version 23.001] Option how historical data are stored in RAM for each parameter. The following options are available:
    - Over last 6 hours: one value each 5 min.; over last 6 min.: one value each 10 sec.
    - Over last 12 hours: one value each 10 Min.; over last 6 min.: one value each 10 sec.
    - Over last 24 hours: one value each 20 Min.; over last 6 min.: one value each 10 sec.
    - Over last 24 hours: one value each 20 Min.; over last 36 minutes: one value each min.
    - Over last 3 days: one value each hour; over last 6 min.: one value each 10 sec.
    - Over last 3 days: one value each hour; over last 36 minutes: one value each min.
- [new in SW version 26.001, only with HW V5 and above] Storage of data on SD card.



Before changing the settings, you should download and show the present settings of the CAN-Gateway. Then you can change them and click on “Save settings” to store changes on CAN-Gateway. After that the CAN-Gateway performs soft reset automatically.

10 Web interface: Parameter settings

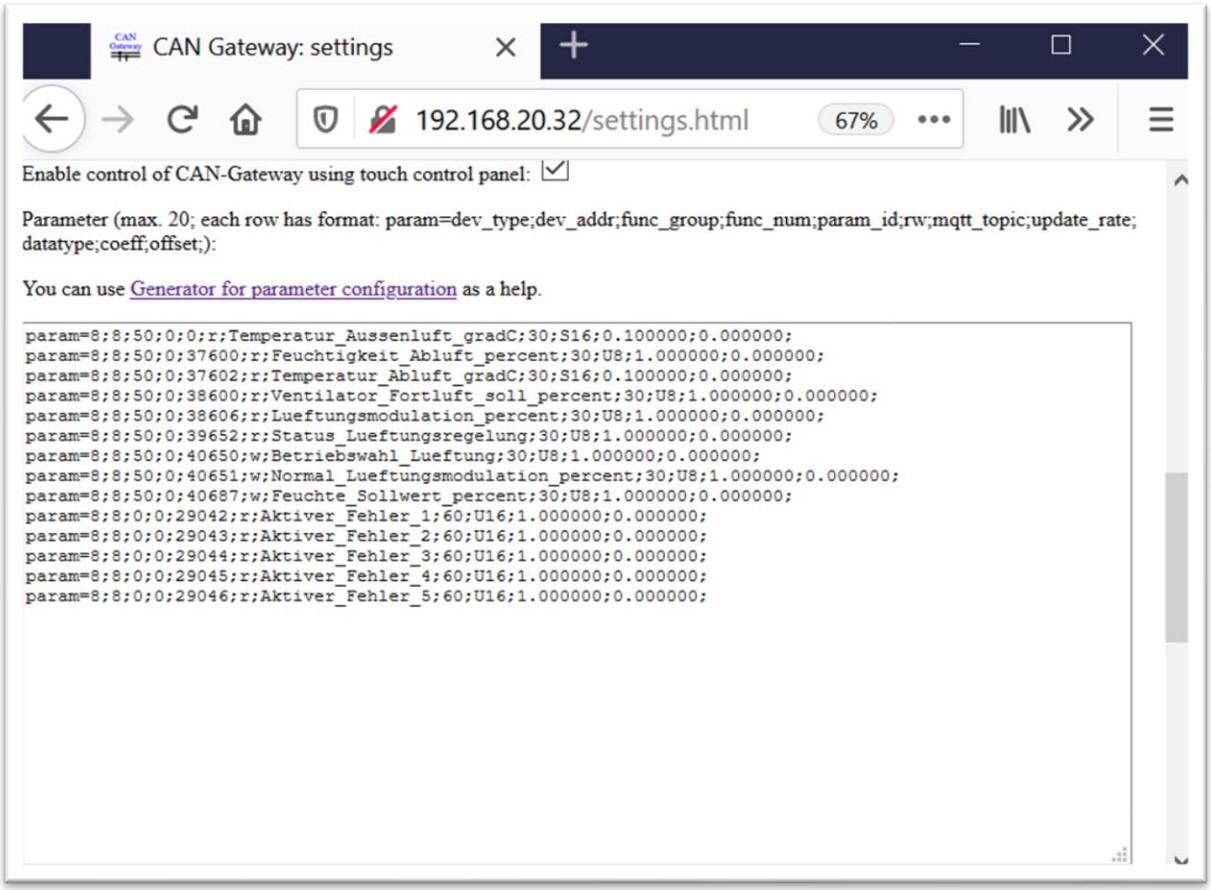
The CAN-Gateway must be configured to read parameters from devices and send them to the home automation system or to set the parameters as requested by the home automation system. At maximum 40 (20 before SW version 23.001) parameters can be configured. As a configuration of each parameter the following options must be set:

- Device type, device address, function group number, function number, datapoint ID. Please refer to the section “General information about devices and their parameters” for more information.
- Data type – please refer to subsection „Data types“.
- MQTT Topic – please refer to section „Parameter transmission using MQTT protocol“.
- Update rate. Update rate can have values 10, 30 or 60 and gives period in seconds that defines how often the CAN-Gateway requests the parameter value from the respective device. Usually the device will answer immediately. This value then read by the CAN-Gateway and transmitted to the MQTT broker (published) with a delay up to 100 ms. The value is also available over REST-API or over Modbus-TCP protocol. The value is published over MQTT-protocol independent if it is changed or not since last reading. To mention is that for configured parameters their values are always published over MQTT when the value is send over the CAN bus. It means, for example, that if other devices request the same parameters more often than defined by the update rate, the values will also more often

published. CAN-Gateway requests the parameter value only if since latest value transmission over the CAN bus more time past as set up by the update rate. The age of a value is given in the Web interface on the page “Parameter list”.

- Read/write attribute. Parameter can be defined as possible to be read (attribute=r) or read and written (attribute=w). If parameter is defined with the attribute=w, it can be requested to be changed over Web interface, MQTT protocol, over REST-API or over Modbus TCP. The request is then provided from the CAN-Gateway to the respective device. If the device accepts the change request, it changes the respective internal parameter and answer with the new value. This new value will be read by the CAN-Gateway and published over MQTT etc. Since the full chain from change request till confirmation answer takes a time, the home automation system must wait a little bit after generating change request before it reads the parameter again to judge if the request was accepted or not. The delay must be at least 0.5 s, 1 s is recommended. The home automation system can generate the change requests very often. To protect devices the CAN-Gateway feed forward these change requests no often than 1 time per 3 seconds. The home automation system must be preferably configured not to send the change requests too often. Otherwise it could theoretically happen that the microcontroller of the CAN-Gateway has too much to do and the whole system gets instable or even crashes.
- Coefficient/Offset. These values are always required, but practically not relevant for STR and RAW data types (might be any values, for example 0/0 in this case). Coefficient and Offset influences how the value, that is transmitted in binary from over the CAN bus, is interpreted and recalculated to the value used (represented via Web interface, published via MQTT protocol etc.). The only exception is Modbus-TCP protocol: in this case the values are transmitted always exactly as over the CAN bus. The required recalculation must be performed in Modbus master. For values represented via Web interface or published via MQTT, the values read from the CAN bus are multiplied with the Coefficient and then the Offset is added. The Coefficient/Offset can be integer or not, positive or negative. At change request the recalculation is performed backwards: for example from the value received by MQTT protocol the Offset is subtracted and then the result is divided by Coefficient. The final result is then requested over the CAN bus. These values required for Coefficient/Offset can be found comparing the raw values shown in CAN-Gateway with the values shown in the room control unit or determined from the parameter lists provided by Hoval for their KNX gateway. The Offset is almost always = 0. The Coefficient must be set depending on the Number of decimals given in the parameter lists in column labelled as “Decimal”. If the value in this column is zero, then Coefficient=1. If Decimal=1 , then Coefficient =0.1 (often for temperatures). If Decimal=2 , then Coefficient =0.01. If Decimal=3 , then Coefficient =0.001.

The parameters are configured using Web interface on the page “Settings”. For this in the field “Parameters” for each parameter a new line used that must start with “param=”. After “param=” the configuration data is written, each value separated from others by semicolon. In total up to 40 (20 before SW Version 23.001) parameters can be configured. Below the field “parameters” a short description of all configuration options is given. In the example below nine parameters are configured. The first seven are User Level passwords (to read them from room control unit).



10.1 Data types

Data type definition influences how the value, that is transmitted in binary from over the CAN bus, is interpreted and recalculated to the value used (represented via Web interface, published via MQTT protocol etc.). The only exception is Modbus-TCP protocol: in this case the values are transmitted always exactly as over the CAN bus. The required recalculation must be performed in Modbus master.

RAW: The value is given in a hexadecimal form without any change. For example, if the value is transmitted over the CAN bus as four following bytes: 0x31 0x32 0x61 0x5A, it will be represented as „3132615A“. The maximum number of bytes is 64.

STR: The value transmitted over the CAN bus will be interpreted as an ASCII coded character string. For example, if the value is transmitted over the CAN bus as four following bytes: 0x31 0x32 0x61 0x5A, it will be interpreted and shown as „12aZ“, because 0x31 code in ASCII represents „1“, 0x32=“2“, 0x61=“a“ and 0x5A=“Z“. The maximum number of bytes is 64.

U8 (unsigned 8-Bit): unsigned byte: value in the range 0 to 255.

S8 (signed 8-Bit): signed byte: value in the range -128 to 127.

U16 (unsigned 16-Bit): unsigned word (2 bytes), value in the range 0 to 65535.

S16 (signed 16-Bit): signed word (2 bytes), value in the range -32768 to 32767.

U32 (unsigned 32-Bit): unsigned double word (4 bytes), value in the range 0 to 4294967295.

S32 (signed 32-Bit): signed double word (4 bytes), value in the range -2147483648 to 2147483647.

The ranges above are without possible modifications based on Coefficient/Offset settings.

10.2 Hints regarding the lists

Some parameters are defined as so called lists. A list in this case is a map having numbers (0, 1, 2 ..) mapped to a certain meaning represented by a word or phrase. As an example the light mode of the room control unit can be considered. It belongs to the function group number 80, function number 1 and datapoint ID = 0. On the room control unit different modes can be selected, for example “time controlled”. Over the CAN bus, however, not the character string “time controlled” is transmitted, but the respective number, in this case =2.

10.3 Examples

10.3.1 Example number 1: Ventilation (HomeVent) – read and write humidity set value

The following example shows how to configure the parameter “humidity set value” of a ventilation device: device type = 8, device address = 8, function group = 50, function number = 0, datapoint ID=40687, read and write (w), MQTT topic = humidity_set_value, read each 10 s; data type: U8, coefficient =1, offset=0:

param=8;8;50;0;40687;w;humidity_set_value;10;U8;1.000000;0.000000;

10.3.2 Example number 2: Ventilation (HomeVent) –extract air temperature

The following example shows how to configure the parameter “extract air temp.” of a ventilation device: device type = 8, device address = 8, function group = 50, function number = 0, datapoint ID=37602, read only (r), MQTT topic = exctract_air_temp_, read each 10 s; data type: S16, coefficient =0.1, offset=0:

param=8;8;50;0;37602;r; exctract_air_temp_;10;S16;0.100000;0.000000;

10.3.3 Example number 3: Ventilation (HomeVent) –outside air temperature

The following example shows how to configure the parameter “outside air temp.” of a ventilation device: device type = 8, device address = 8, function group = 50, function number = 0, datapoint ID=0, read only (r), MQTT topic = outside_air_temp_, read each 10 s; data type: S16, coefficient =0.1, offset=0:

param=8;8;50;0;0;r; outside_air_temp_;10;S16;0.100000;0.000000;

10.3.4 Example number 4: Room control unit – read and write password for user level 7

The following example shows how to configure the parameter “password for user level 7” of a room control unit: device type = 16, device address = 8, function group = 89, function number = 1, datapoint ID=7, read and write (w), MQTT topic = password_level7, read each 60 s; data type: STR, coefficient =1 and offset=0, but could be any values because are not relevant for STR data type:

param=16;8;89;1;7;r;password_level7;60;STR;1;0;

10.3.5 Example number 5: Ventilation (HomeVent) – operating mode for ventilation

The following example shows how to configure the parameter “Op. choice ventilation**”** of a ventilation device: device type = 8, device address = 8, function group = 50, function number = 0, datapoint ID=40650, read and write (w), MQTT topic = op__choice_ventilation**,** read each 60 s; data type: U8, coefficient =1, offset=0:

param=8;8;50;0;40650;w; op__choice_ventilation;60;U8;1.000000;0.000000;

The operating mode is coded by a number in the range 0 to 5. According to the information in file (http://www.hoval.com/misc/TTE/TTE-GW-Modbus-datapoints.xlsx), the following modes are possible: Standby=0, Week1=1; Week 2=2; Constant operation=4; Eco mode=5.

10.3.6 Example number 6: Ventilation (HomeVent) – Date/Time

The following example shows how to configure Date and Time of a ventilation device:

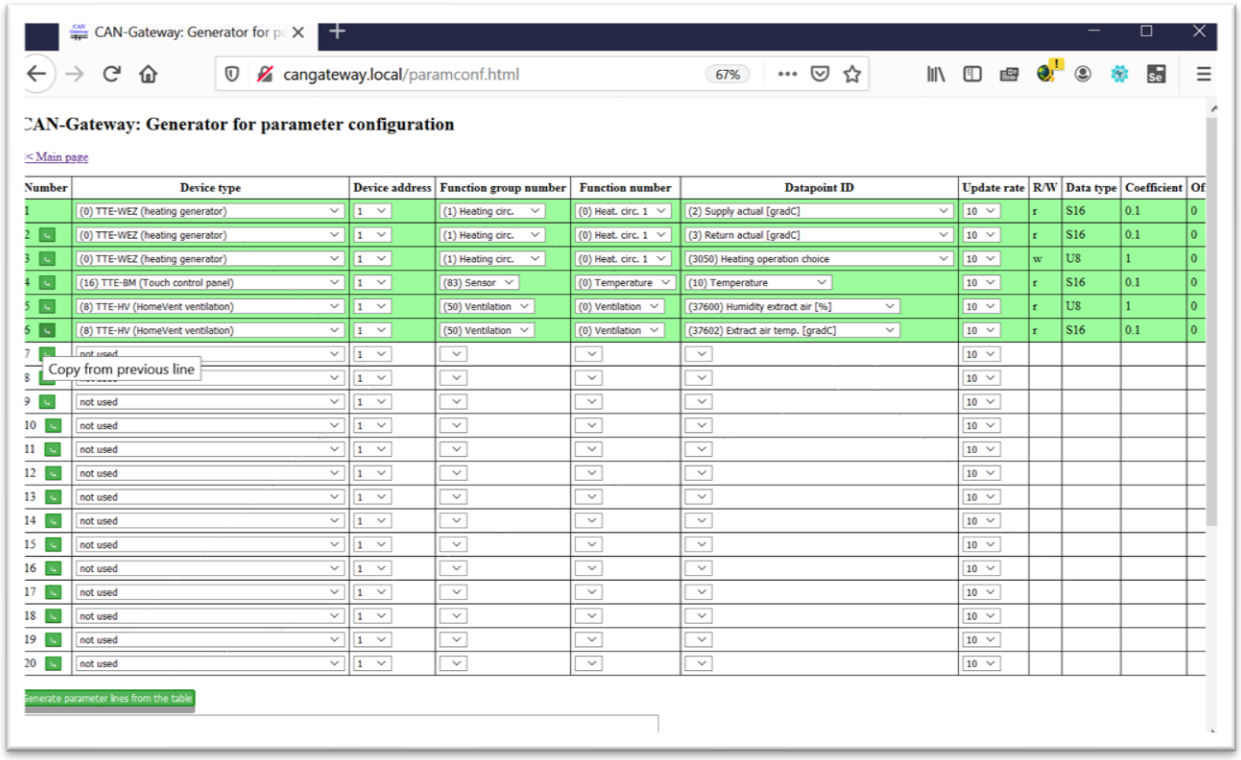
param=8;8;0;0;2070;w;Date;10;U16;1.000000;0.000000;
param=8;8;0;0;2072;w;Time;10;U16;1.000000;0.000000;

Thereby „Date“ is the number of minutes since beginning of the day; „Date“ is the number of days since 1/1/1900.

© W. Waag, Munich, 2020-2024

11 Web interface: Generator for parameter configuration

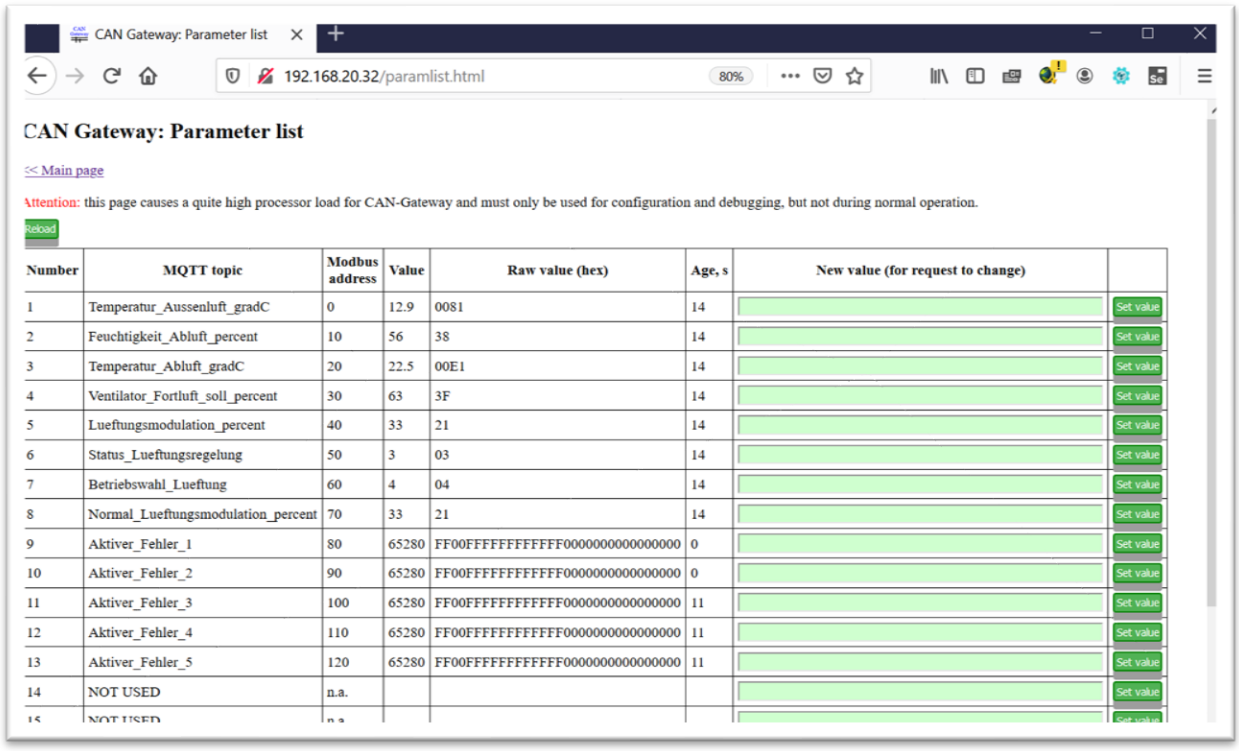
This generator generates configuration lines for the parameters. The user can choose in a table the parameters he needs and generate then the configuration lines. These lines can be then copied into clipboard and then can be paste into configuration input field on the settings page. This table contains lots of different parameters known for all device types.



© W. Waag, Munich, 2020-2024

12 Web interface: Parameter list

Using web interface it is possible to show the configured parameters with their present values as to generate change requests for parameters that can be changed. This page (and therefore the present values) is not updated periodically, but you can use the “Reload” button. To change the parameter value (or exactly to say, to generate the change request) the new value must be enter into the respective green field and then you have to click on the respective “Set value” button.



Number	MQTT topic	Modbus address	Value	Raw value (hex)	Age, s	New value (for request to change)	
1	Temperatur_Aussenluft_gradC	0	12.9	0081	14		Set value
2	Feuchtigkeit_Abluft_percent	10	56	38	14		Set value
3	Temperatur_Abluft_gradC	20	22.5	00E1	14		Set value
4	Ventilator_Fortluft_soll_percent	30	63	3F	14		Set value
5	Lueftungsmodulation_percent	40	33	21	14		Set value
6	Status_Lueftungsregelung	50	3	03	14		Set value
7	Betriebswahl_Lueftung	60	4	04	14		Set value
8	Normal_Lueftungsmodulation_percent	70	33	21	14		Set value
9	Aktiver_Fehler_1	80	65280	FF00FFFFFFFFFFFF0000000000000000	0		Set value
10	Aktiver_Fehler_2	90	65280	FF00FFFFFFFFFFFF0000000000000000	0		Set value
11	Aktiver_Fehler_3	100	65280	FF00FFFFFFFFFFFF0000000000000000	11		Set value
12	Aktiver_Fehler_4	110	65280	FF00FFFFFFFFFFFF0000000000000000	11		Set value
13	Aktiver_Fehler_5	120	65280	FF00FFFFFFFFFFFF0000000000000000	11		Set value
14	NOT USED	n.a.					Set value
15	NOT USED	n.a.					Set value

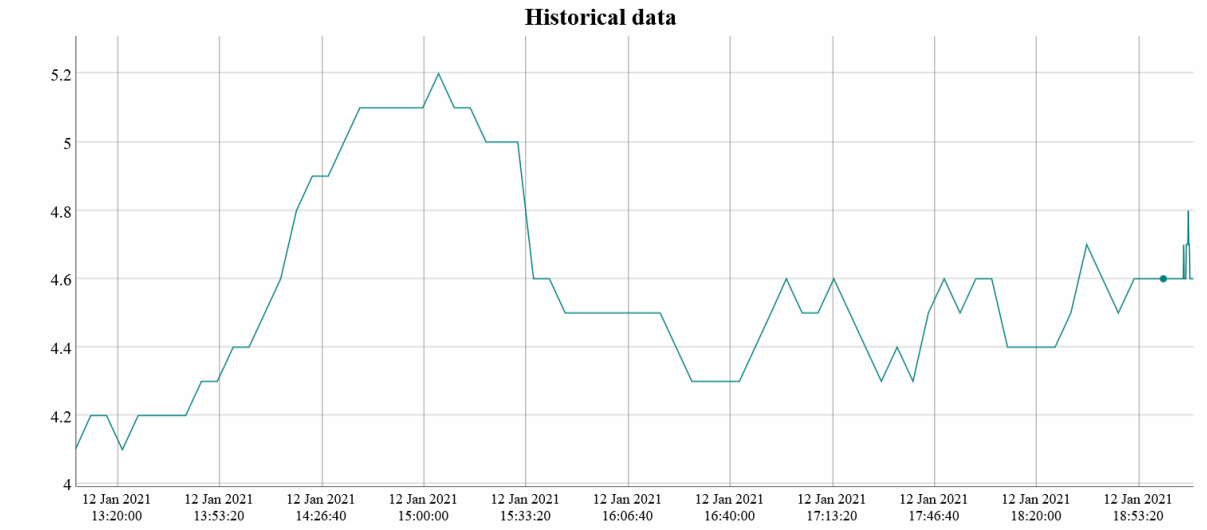
SW Version starting with 21.002 has also possibility to see historical data for parameters (up to latest 24 hours) as a table or a graph. For this it has two small green buttons called “H” and “P”:

Number	MQTT topic	Modbus address	Value	Raw value (hex)	Age, s	New value (for request to change)	
1 H P	Temperatur_Aussenluft_gradC	0	0.1	0001	30		Set value
2 H P	Feuchtigkeit_Abluft_percent	10	65	41	1		Set value

The historical data are not stored for parameters with data types STR and RAW.

© W. Waag, Munich, 2020-2024

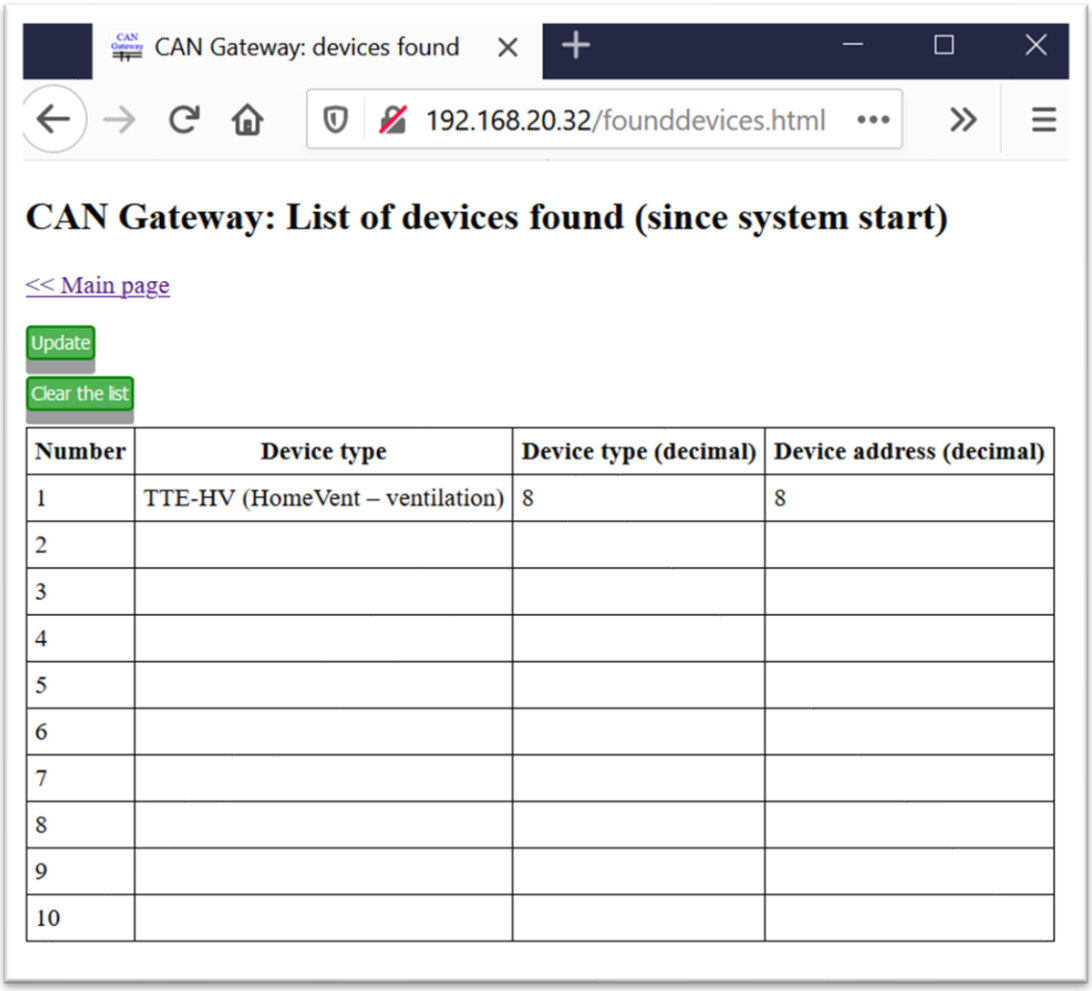
12 Jan 2021 19:01:31:
Temperatur_Aussenluft_gradC: 4.6



© W. Waag, Munich, 2020-2024

13 Web interface: Device list

When a device transmit something on the CAN bus, the device is detected by the CAN-Gateway and will be shown in this list.

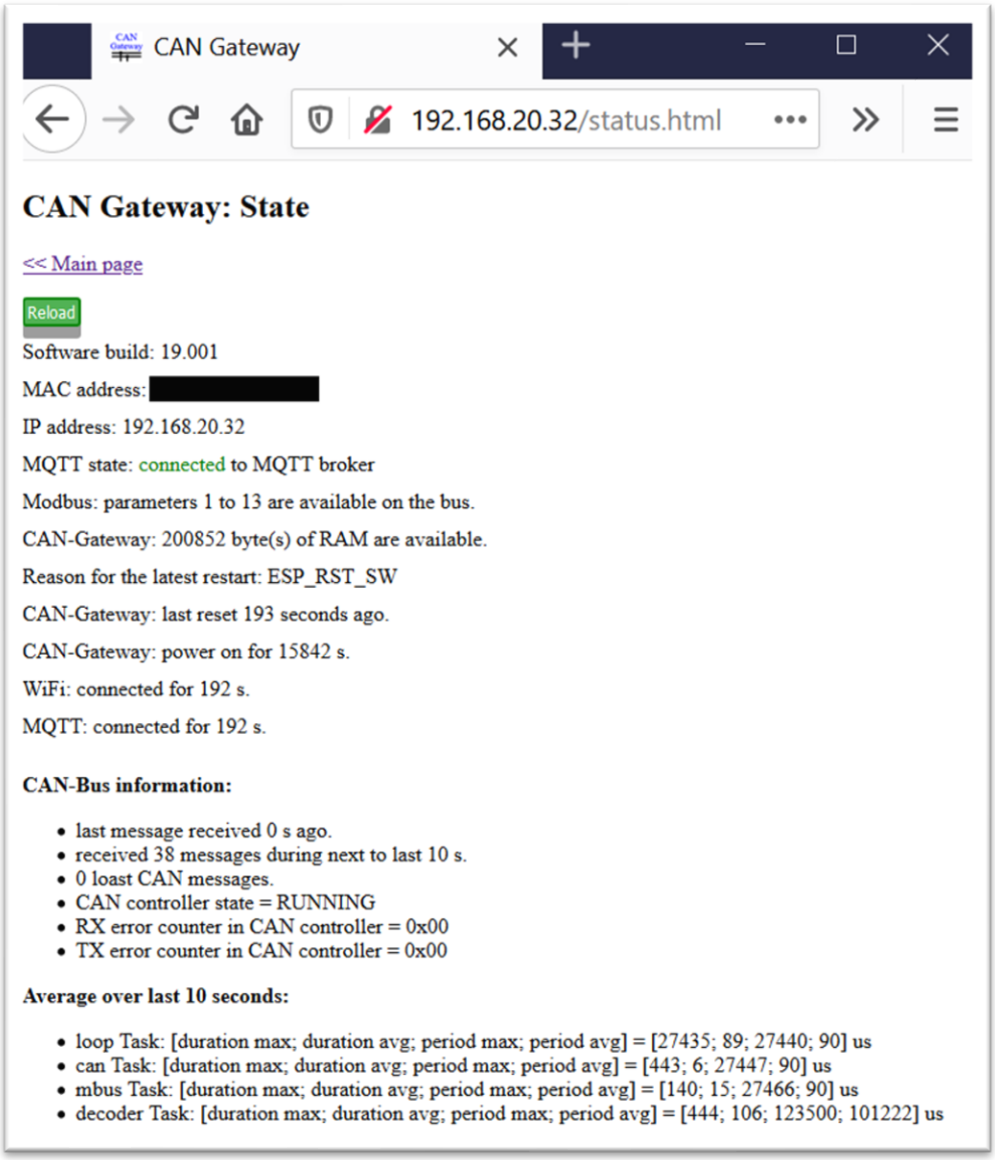


Number	Device type	Device type (decimal)	Device address (decimal)
1	TTE-HV (HomeVent – ventilation)	8	8
2			
3			
4			
5			
6			
7			
8			
9			
10			

© W. Waag, Munich, 2020-2024

14 Web interface: Status information

On this page the internal status information of the CAN-Gateway is shown. It could be helpful for troubleshooting.



© W. Waag, Munich, 2020-2024

15 WEB interface: SD card info / status

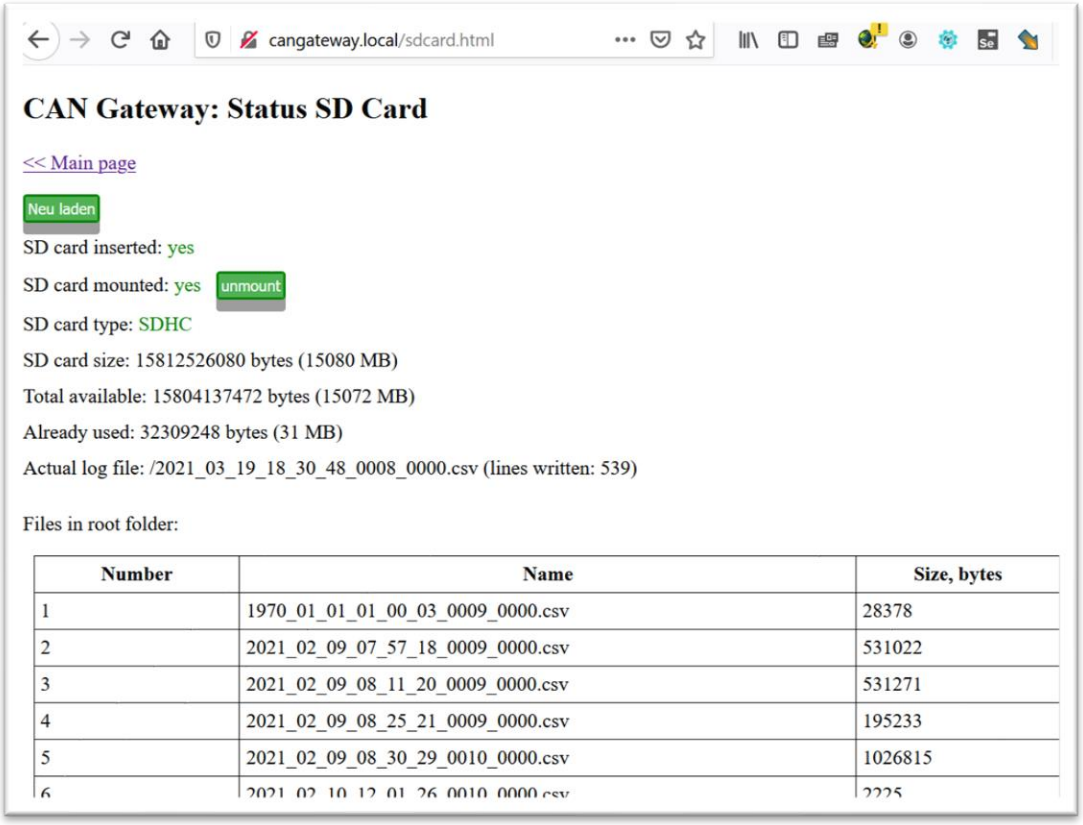
[new in SW 26.001, only in connection with HW V5 and higher or Olimex board]

The current status of the SD card is displayed. The SD card can basically be inserted and removed during operation. However, it is strongly recommended to "unmount" the SD card before unplugging. SD and SDHC micro SD cards are supported. The SD card must be formatted with FAT32 before usage.

The data, if activated, are saved in files. The file name is formed as follows:

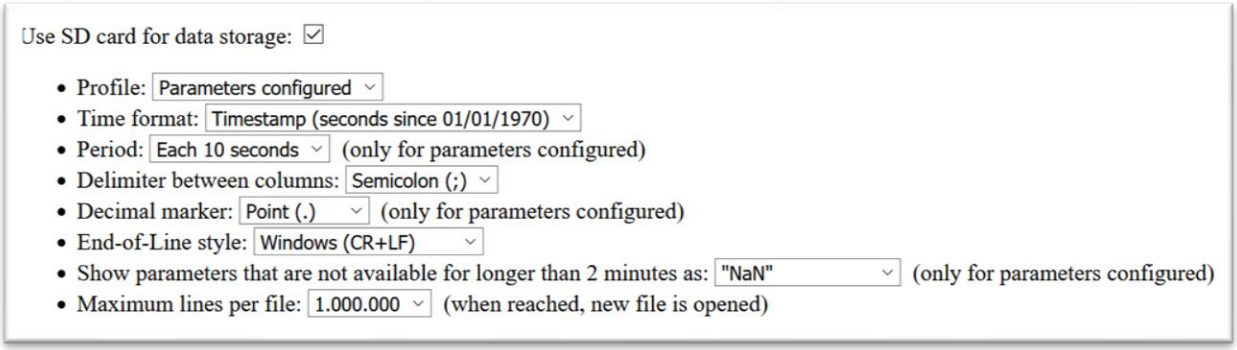
Year_month_day_hours_minutes_seconds_configurationnumber_0000.csv

Configuration number is simply a number that increases by 1 each time the configuration is changed.



Number	Name	Size, bytes
1	1970_01_01_01_00_03_0009_0000.csv	28378
2	2021_02_09_07_57_18_0009_0000.csv	531022
3	2021_02_09_08_11_20_0009_0000.csv	531271
4	2021_02_09_08_25_21_0009_0000.csv	195233
5	2021_02_09_08_30_29_0010_0000.csv	1026815
6	2021_02_10_12_01_26_0010_0000.csv	2225

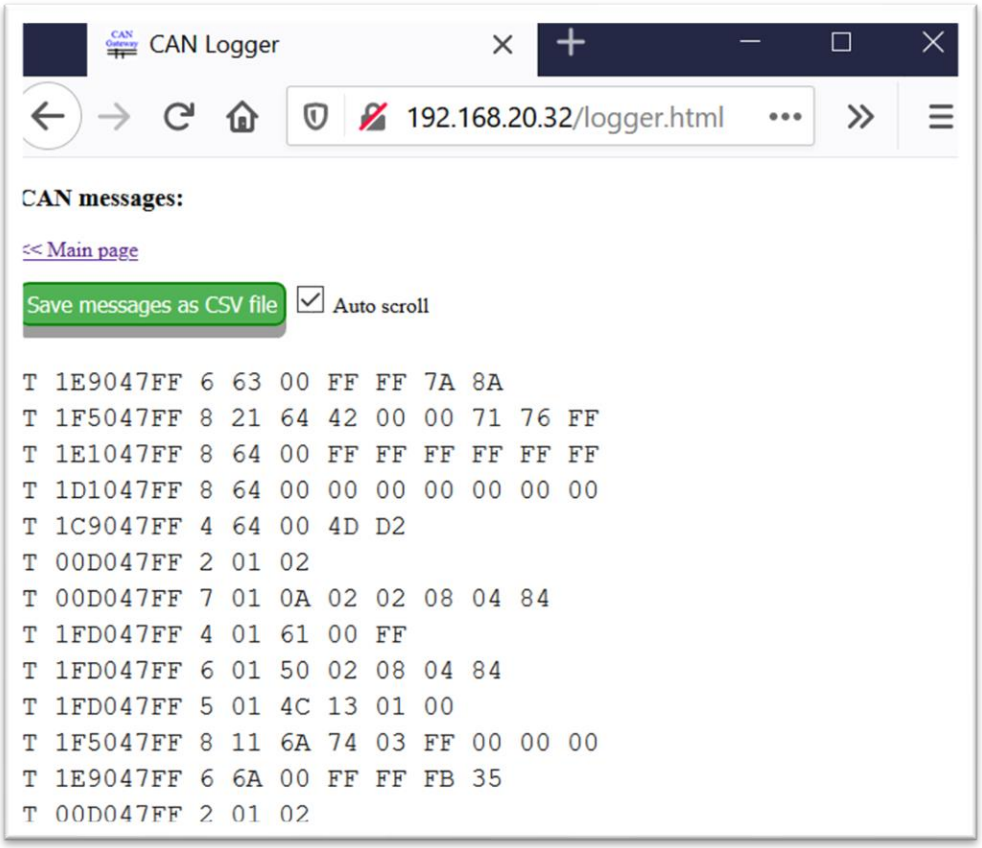
The format of the data can be configured in the settings:



© W. Waag, Munich, 2020-2024

16 Web interface: CAN data logger

All CAN messages received by the CAN-Gateway can be shown as a raw data. It could be helpful for troubleshooting.

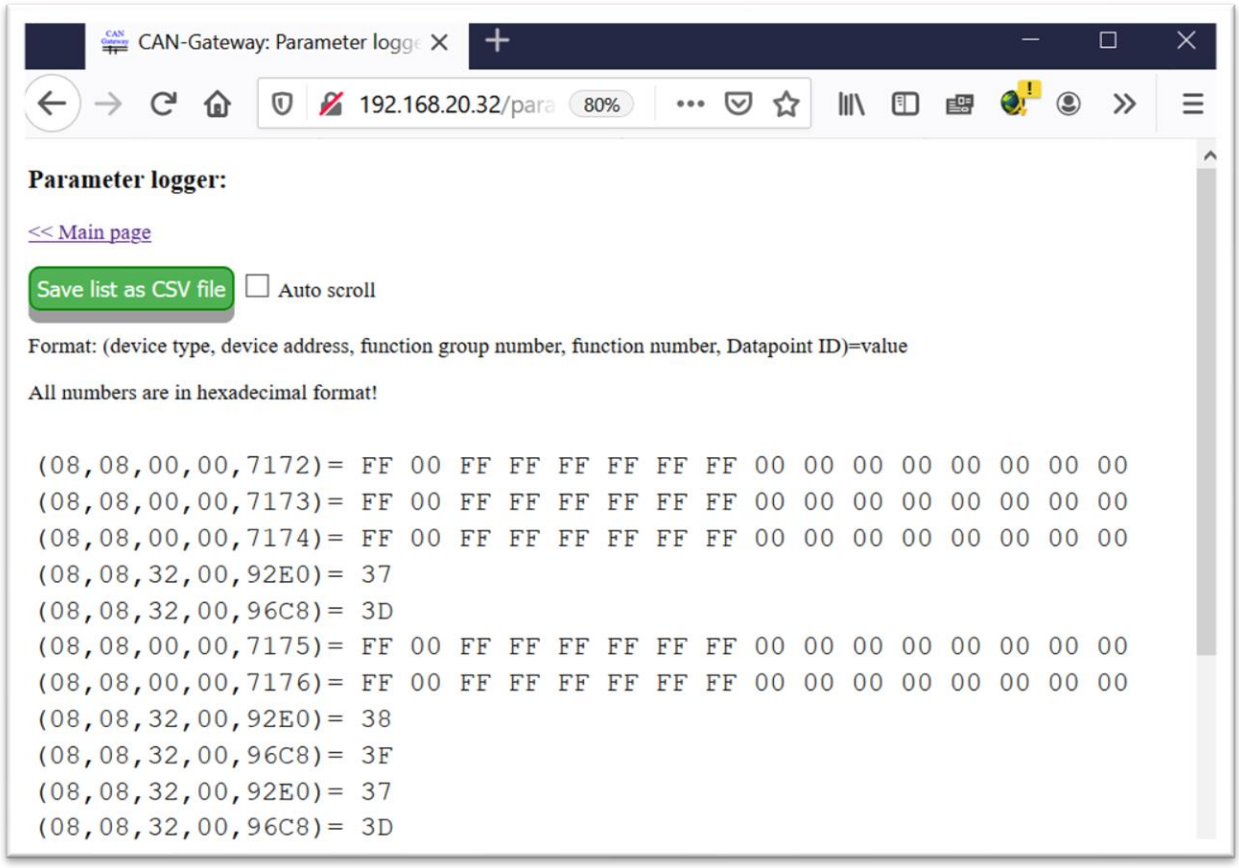


New messages received are loaded automatically each 5 seconds and shown at the end of the list. The messages can be exported to a CSV file. The first symbol “t” or “T” means a standard or extended CAN message frame. Then follow CAN-ID, DLC (data length) and 1 to 8 data bytes. Only messages containing can data are shown. RTR- and error-frames are ignored.

© W. Waag, Munich, 2020-2024

17 Web interface: Parameter logger

All parameters received by the CAN-Gateway are shown on this page. It could be helpful for troubleshooting.



The parameters received are loaded each 5 seconds and added at the end of the list. The parameters can be exported to a CSV file.

© W. Waag, Munich, 2020-2024

18 Web interface: KNX IP settings [new in SW version 29.001]

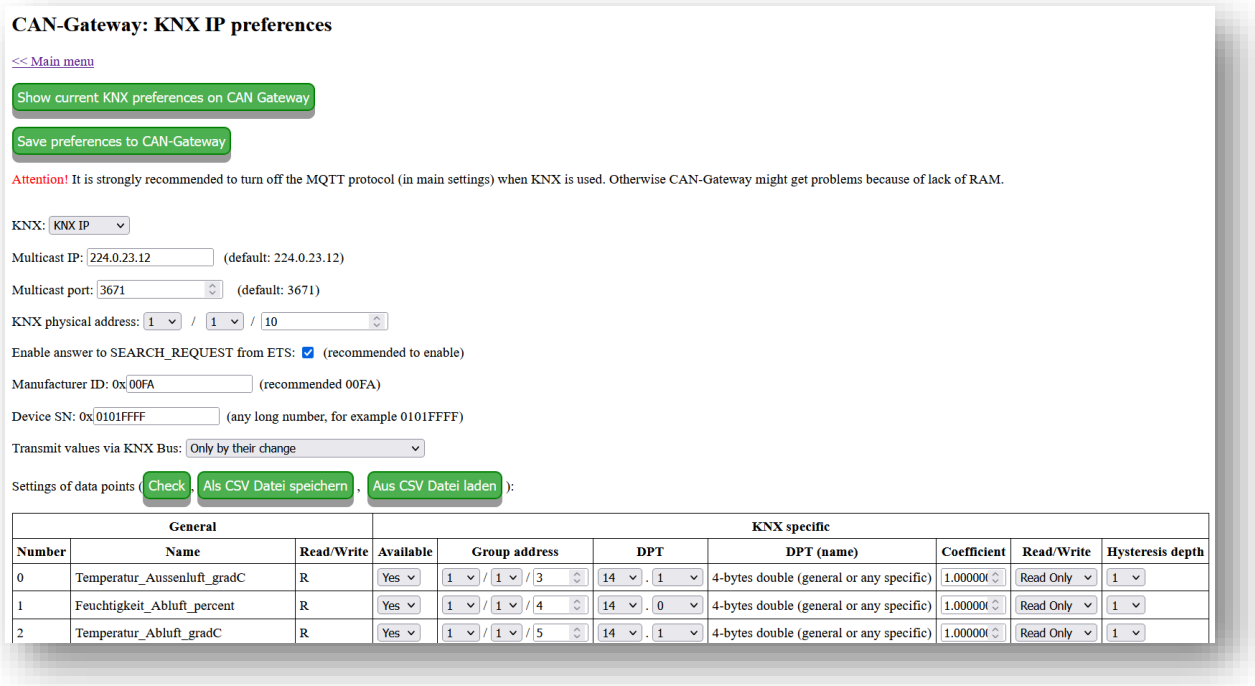
CAN-Gateway supports KNX IP multicasting [from SW version 29.001] (currently no tunnelling, no KNX IP Secure and not all datapoint types are supported).

It is strongly recommended to switch off the MQTT protocol (in main settings) if KNX IP is used. Otherwise problems (reboots, web UI is very slow etc.) can occur due to a lack of RAM on the CAN-Gateway.

You must have a KNX IP router (physical or knxd service on a Linux computer). The KNX IP parameters are set via the CAN-Gateway Web interface and not via the ETS software. The CAN-Gateway can be integrated as a dummy device in the ETS software.

CAN-Gateway has an option to answer to SERVICE_REQESTS from ETS5. Using this you can test CAN-Gateway KNX IP without having KNX IP Router/knxd.

If you are familiar with KNX, most of the settings are self-explanatory. Detailing is only necessary for the "Coefficient" and DPT (DataPoint Type) parameters:



Coefficient: should be "1.0" in most cases. The real values are multiplied by this coefficient before they are sent via KNX IP. Values other than "1.0" can be used in some special situations if required.

DPT: DPT (x, y) with the following main type "x" are supported:

- DPT(1,y) [1-Bit Boolean]
- DPT(5,y) [1-Byte unsigned]
- DPT(6,y) [1-Byte signed]
- DPT(7,y) [2-Byte unsigned]
- DPT(8,y) [2-Byte signed]
- DPT(9,y) [2-Byte double]

- DPT(12,y) [4-Byte long unsigned]
- DPT(13,y) [4-Byte long signed]
- DPT(14,y) [4-Byte double]

The user can generally assign any DPT to each parameter. It is recommended to use DPT (14, y) for physical values (such as temperature, humidity etc.). Subtype is ignored in the current implementation, e.g. there is no corresponding scaling of the value, which is sufficient in most cases.

[new in V 29.310] There is an option to select when the sensor values are sent by the CAN-Gateway via KNX:

- Always
- Only when changes
- Only when changes and every 5 minutes
- Only when changes and every 10 minutes
- Only when changes and every 30 minutes

Regardless of the option selected, the values are sent once when the CAN-Gateway is started/rebooted and when the CAN-Gateway is reconnected to the WLAN/LAN (e.g. after the LAN cable was temporarily unplugged, etc.).

For each parameter there is also the option of defining whether a hysteresis should be used for sending "on change". Hysteresis depth of "0" means that the value is resent with each change. Hysteresis depth of "1" means that the change by one step is ignored. E.g. if the temperature value of 20.5 °C was sent last, it will only be sent again at 20.7 °C if the temperature rises, but not at 20.6 °C.

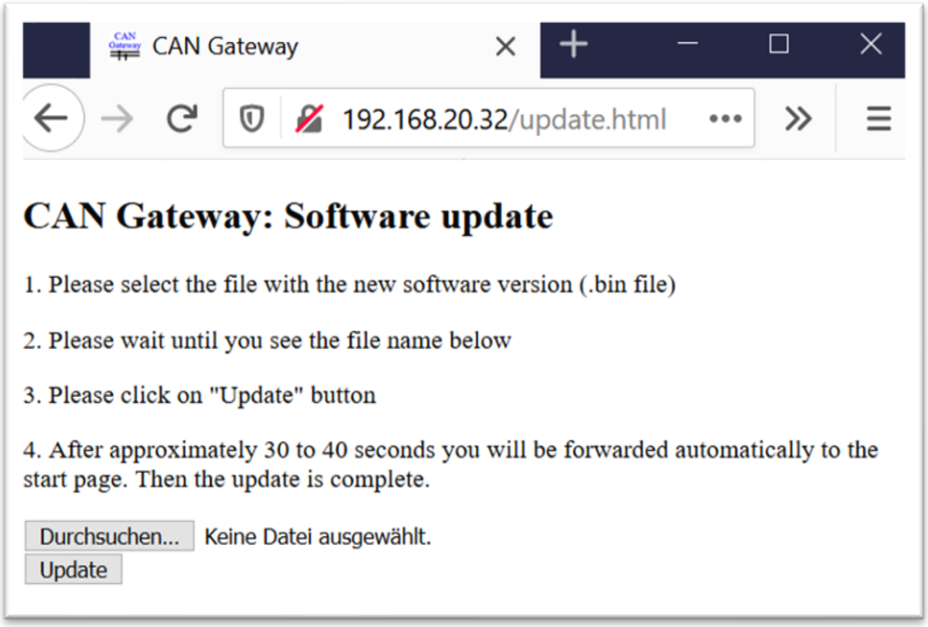
© W. Waag, Munich, 2020-2024

19 Web interface: Software update

CAN-Gateway software can be updated via Web interface.

DEMO version can be updated to a full version and vice versa.

Attention: Software break between version 29.101 and 29.200. Due to the changed basic software, version 29.101 or lower CANNOT be updated to software 29.200 or higher via the WEB interface. To update, please first update to the new DEMO version using the web-based installer (https://wladwnt.github.io/) and then the DEMO version can be updated to the full version via the WEB interface.



Update to the next major version (for example from SW 22.001 to 24.001) will reset all settings (including WiFi network name and password) to their initial values. In this case you have to configure the CAN-Gateway again. Update to the next minor version (for example from 22.001 to 22.003) will not reset settings.

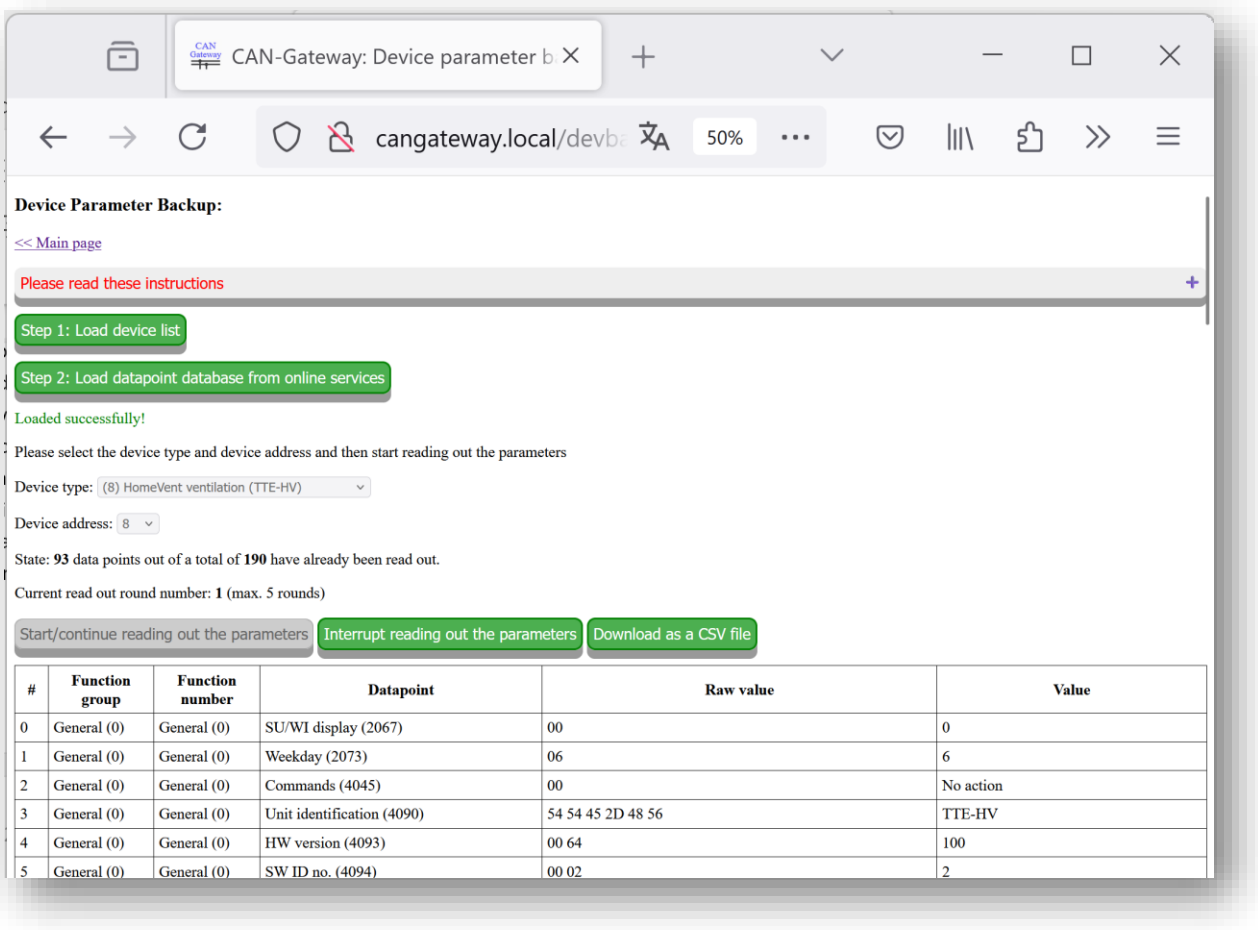
© W. Waag, Munich, 2020-2024

20 WEB interface: Read all datapoints from device

[new from SW 30.020] This function allows all data points of a device (a controller) to be read out. These can then be saved in a CSV file. In this way, a backup of the device parameters can be created. Or you can then compare the two files from two devices to find out what differences in configuration both devices have.

Reading out all data points can take quite a long time. With a TTE-WEZ controller, for example, typically around 1500 data points are read out.

Before starting, please read the instructions on the process very carefully. These can be seen by clicking on “Please read these instructions”.



#	Function group	Function number	Datapoint	Raw value	Value
0	General (0)	General (0)	SU/WI display (2067)	00	0
1	General (0)	General (0)	Weekday (2073)	06	6
2	General (0)	General (0)	Commands (4045)	00	No action
3	General (0)	General (0)	Unit identification (4090)	54 54 45 2D 48 56	TTE-HV
4	General (0)	General (0)	HW version (4093)	00 64	100
5	General (0)	General (0)	SW ID no. (4094)	00 02	2

21 Parameter transmission using MQTT protocol

In your home automation system you will need a MQTT broker running. MQTT broker is a server that handles MQTT data, for example collects data from sensors and provides them to other devices. CAN-Gateway then can be configured to have a steady connection to MQTT broker. CAN-Gateway support presently only unencrypted Modbus TCP communication. Anyone who has access to your WiFi Network, can access (sniff) parameter values over the MQTT protocol. Therefore, you should only use CAN-Gateway only in your private WiFi network that is sufficiently protected by password etc. For your safety and security you should disable in your internet router any port forwarding from

the internet to CAN-Gateway. Your MQTT broker must support authentication via username and password and unencrypted data transmission over TCP protocol.

In MQTT protocol each piece of data (in or case each parameter) is described by a unique name called “topic”. CAN-Gateway is configured to generate a topic for each configured parameter. The topic has a part common for all topics followed by “/” symbol and then followed by the part specific for each parameter. The common part (default: “cangateway”) can be chosen under general settings. The specific part (for example “temperature”) is defined in the configuration line for each parameter separately. The resulting topic for parameter (read) in this example “cangateway/temperature”. Change requests for parameters that have read/write attribute = w (write) must be sent to a separate topic. It has the same name as reading topic but followed by “/set”. In the example above it will be “cangateway/temperature/set”.

© W. Waag, Munich, 2020-2024

22 Read/Write parameter values over Modbus TCP

22.1 Notes regarding Modbus TCP implementation

CAN-Gateway supports presently only unencrypted Modbus TCP communication. Anyone who has access to your WiFi Network, can access parameter values over the Modbus TCP. Therefore you should only use CAN-Gateway only in your private WiFi network that is sufficiently protected by password etc. For your safety and security you should disable in your internet router any port forwarding from the internet to CAN-Gateway.

22.2 Modbus addresses

The parameter addresses in Modbus TCP are defined as follows. Generally each configured parameter has an address according to its order number:

Basis address = (order number-1)*10

In Modbus protocol an address is valid for one 16-bit value (also called one register). Therefore the address describe above is only valid for parameter values, that are fit into one register, meaning only for the following data types: U8, S8, U16 and S16.

Parameters with data types U32 or S32 will get two addresses, meaning will be mapped to two registers in terms of Modbus. The basis address above will address the two highest bytes (also will include the most significant bit), the next address (=basis address + 1) will address the two lowest bytes (also will include the last significant bit). For example, if a parameter number 5 has type U32, then it will get addresses 40 and 41. Parameters with data types RAW and STR will get 10 addresses, meaning they can be transmitted in 10 registers in terms of Modbus and their length is limited to 20 characters each. The rest characters (if any left) will be ignored.

Eventually not all parameters are available via Modbus TCP. CAN-Gateway can handle maximum 32 Modbus registers in total. Parameters with data types U8, S8, U16 and S16 need one register for reading and eventually one additional for writing. Parameters with data types U32 and S32 need two registers for reading and eventually two additional for writing. Parameters with data types RAW and STR need 10 registers each. It is recommended to configure the CAN-Gateway that way that parameters that are not needed over the Modbus are placed at the end of configuration list.

22.3 Register types

All parameters are defined as Modbus register type “Input Register” and can be read via Modbus TCP using Modbus function „Read Input Registers (0x04)“. Parameters, that can be changed/written (have attribute “w”), have additional register of type “Holding Register“ and can be written using Modbus functions „Write Single Register (0x06)“ und „Write Multiple Registers (0x10)“. If you try to read holding register with the Modbus function „Read Holding Register (0x03)“, you will not get the present parameter value, but the latest written value or zero. Better not to use this function at all.

Parameters of data type RAW and STR can generally only be read over the Modbus, but cannot be changed (independent from r/w attribute).

22.4 Transmitted values

All parameter values are transmitted over Modbus TCP as a raw data. No recalculation using coefficient and offset is performed. It must be performed, if necessary, on Modbus Master. You have to configure your Modbus Master accordingly.

Parameter with data type U8 and S8 are transmitted as 16-bit values, because Modbus registers are 16-bit.

Parameter with data type U32 and S32 are, as described above, transmitted using two registers. In your Modbus master you have to select the correct byte and register order.

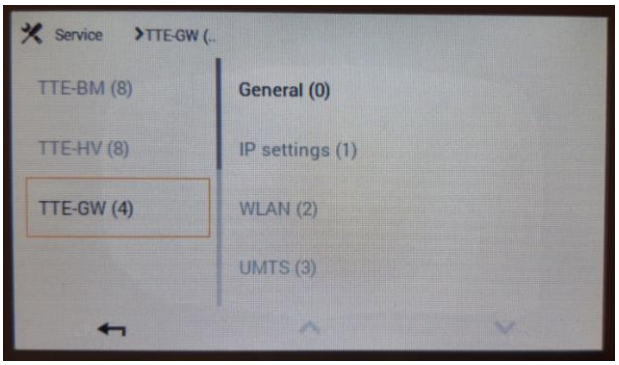
RAW and STR parameters are handled in the same way and are generally transmitted as raw data. It means, for example, that the character string „Enabled“ is transmitted as 16-bit HEX values 0x456E, 0x6162, 0x6C65, 0x6400 (ASCII coded, eventually extended by 00). The first two characters „En“ (0x456E) are transmitted in the first register with address calculated as shown above, the next two characters „ab“ (0x6162) in the next register (address+1) and so forth.

© W. Waag, Munich, 2020-2024

23 Control of the CAN-Gateway using room control unit

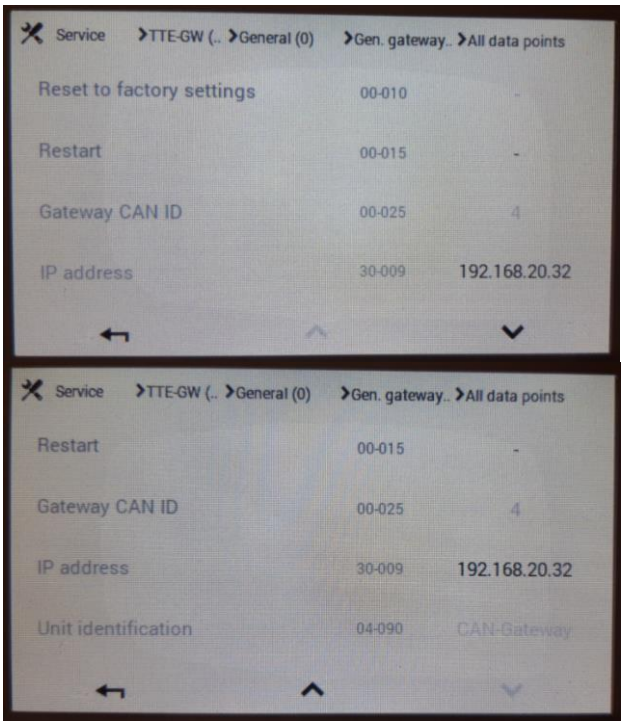
If the CAN-Gateway is connected to a room control unit (TTE-BM) over CAN bus, it can be controlled using this room control unit. The control is limited to some basic functions: CAN-Gateway can be restarted and the settings can be reset to factory values.

Additionally some information will be available such as CAN-Gateway WiFi status, network name and password, WiFi signal quality. The prerequisite is, however, that the option “Allow to control CAN-Gateway through room control unit” is activated in CAN-Gateway settings. In that case the CAN-Gateway will be as TTE-GW device in menu „Service“:



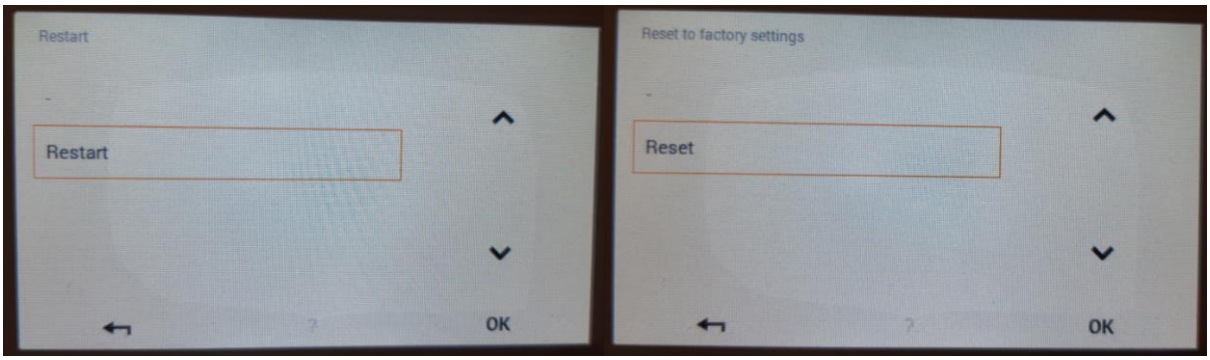
To access 5 options available: do click on **General(0) → General Gateway → Information (1)**, then scroll down to **All Datapoints**, then click on the list of data on the right-hand side:

- Reset to factory settings
- Restart
- Get Gateway CAN ID
- Get IP address
- Get unit identification (always as “CAN-Gateway”)



© W. Waag, Munich, 2020-2024

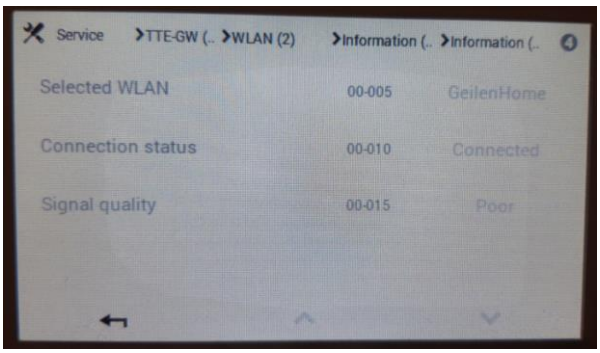
To restart or set the CAN-Gateway to factory settings, you must click to the respective line, symbol „-“, then click to “Restart” or “Reset” and then click on “OK“:



Reset option is only available, if you are in User Level 3 or higher.

To access the following values, please click on **WLAN (2) → Information (0) → Information (1)**:

- Selected WLAN
- Connection state
- Signal quality



© W. Waag, Munich, 2020-2024

24 REST-API

REST-API (application Interface) is a possibility to access the configured parameters (their values) over the http protocol. There also some other functions implemented as REST-API. Web interface of the CAN-Gateway itself uses this REST-API. In the following table the REST-API is described (not completely, its main functions).

URL	GET or POST method	Description
http://cangateway.local/getparam?num=XX	GET	The response contain the present value of the parameter number XX (XX=1 to 40)
http://cangateway.local/setparam?num=XX	POST	Change request for parameter number XX, the value must be transmitted in a POST body as text/plain.
http://cangateway.local/getparamraw?num=XX	GET	The response contain the present value of the parameter number XX (XX=1 to 40) as a raw data (as transmitted over CAN bus, without recalculations using coefficient/offset)
http://cangateway.local/getparamage?num=XX	GET	The response contain the age of the present value of the parameter number XX (XX=1 to 40)
http://cangateway.local/getmac	GET	The response contains the MAC address of CAN-Gateway
http://cangateway.local/getip	GET	The response contains the IP address of CAN-Gateway
http://cangateway.local/getmqttstate	GET	The response contains the state of MQTT connection to MQTT broker

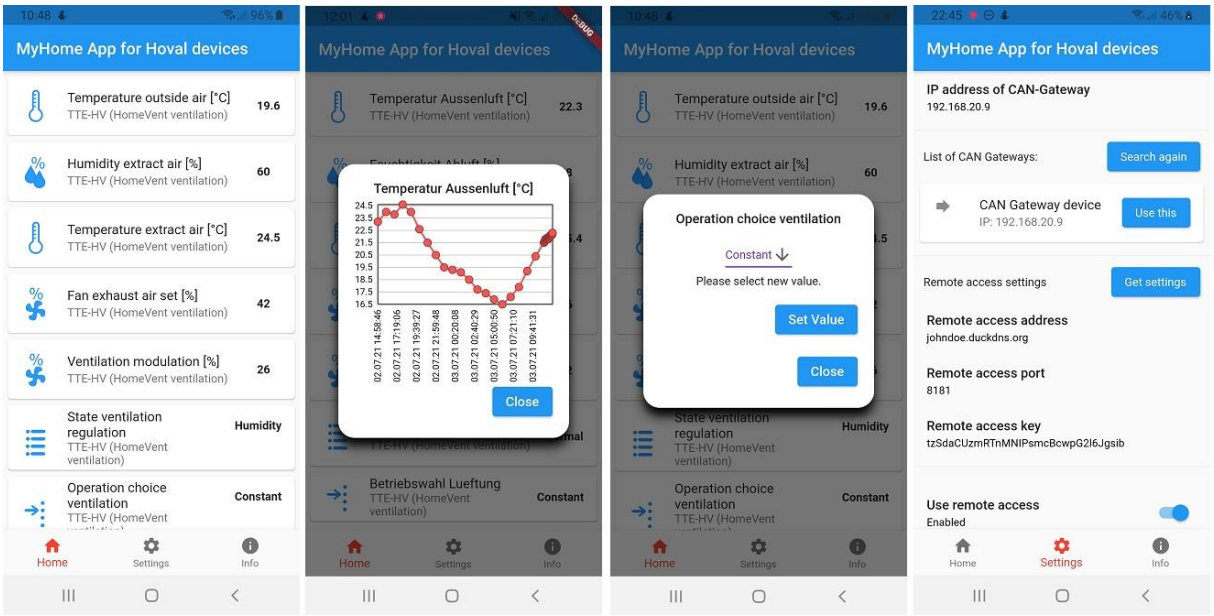
25 Special notes regarding changes of parameters

Change of parameter values (change requests) must be performed taking special attention. CAN-Gateway allows changing many parameters, even if their change is not meaningful at all. If CAN-Gateway is connected to your home automation system and is configured to be able to change some parameters (have write attribute), you must ensure that your home automation system only generates valid change requests with meaningful values.

26 Android App: MyHome App for Hoval devices

App for Android smartphones: can be used directly with CAN-Gateway (as long as the smartphone is in the same WiFi network as CAN-Gateway, starting from CAN-Gateway version 28.100 / App version 1.1 also via Internet). Available on Google Play:
https://play.google.com/store/apps/details?id=com.wladwnt.myhome_for_hoval_devices

CAN-Gateway with Software Version >=27.001 required.



27 Windows App: MyHome App for Hoval devices

Windows app is the same as Android app. It runs under Windows 64-bit. Unfortunately, 32-bit Windows is not supported. The Android / Windows app is programmed in Flutter. Windows support in Flutter is still in beta, so it could theoretically be that Windows App still has some bugs. However, so far I haven't noticed any problems.

28 Remote access to CAN-Gateway via Internet (from SW version ≥ 28.100)

From SW version 28.100 it is possible to access CAN-Gateway via Internet. The data are transmitted encrypted in both directions. The data transfer takes place via the HTTP protocol. Because of the encryption, access via the Internet with a normal WEB browser does not work; the Android/Windows app must be used. The access via web browser is still possible in the local WiFi / LAN network.

For access via the Internet, your in-house network currently must have a public IPv4 address (dynamic or static). There are now some internet providers who only offer a so-called Dual-Stack **Lite** by default (e.g. Vodafone / Cable Deutschland, Unitymedia). In this case your in-house network only gets public IPv6 address and no IPv4 address that would be only yours. Therefore it is not possible to easily establish a connection from the Android / Windows app to the CAN-Gateway. Experience has shown that you can ask for the network only activation of the dual stack (normal, not **Lite**) and you will most likely get it for free.

28.1 Establishing access

Secure access to the CAN-Gateway can be configured as follows:

1) To access the CAN-Gateway via Internet, the CAN-Gateway must have a valid web address. This can be set up using a so-called DDNS (dynamic DNS) service. There are various free DDNS providers (e.g. duckdns.org, dyndns.org, dynu.com etc.). You have to register with one of the providers (if you have not already done it for other purposes). You choose an available web address (such as johndoe.duckdns.org or johndoe1.dyndns.org). You also get a username or token and, usually, a password. You will need these three in the next but one step.

2) The CAN-Gateway must be accessible from the Internet via a TCP port (default: 8181, but another can be selected in the next step). Usually you are connected to the internet via a DSL / cable / WiFi router. By default, this prohibits external access from the Internet to other devices in the local WiFi / LAN network. You have to set up a port sharing for port number 8181 for the CAN-Gateway in your router. Regarding port sharing, please read the user manual for your router. In the case of a Fritzbox, for example, go to the menu "Internet → Permit access" and then under "Port sharing".

3) Select the item "Remote access via Internet" via the WEB interface. The following parameters must be set here:

Enable remote access: logically needs to be checked

Access address: the DDNS web address selected in the first step (complete, e.g. johndoe.duckdns.org)

Server port: 8181. The advanced users can choose a different number.

Key for encryption: generate a key by clicking on "Generate new key". This key will be automatically transferred to your CAN-Gateway Android/Windows app in the next step, so you don't necessarily have to remember it. Keep this key top secret! The key is there so that only you or your app can access the CAN-Gateway. All transmitted data are also encrypted with this key.

Control DDNS via CAN-Gateway: if you are not already using DDNS, e.g. through a router, this option must be checked.

IP address API provider: simply select one of the available options here. CAN-Gateway uses an IP API provider (no registration or similar is necessary for this) to determine its own externally assigned IP address. This is then communicated to the DDNS provider.

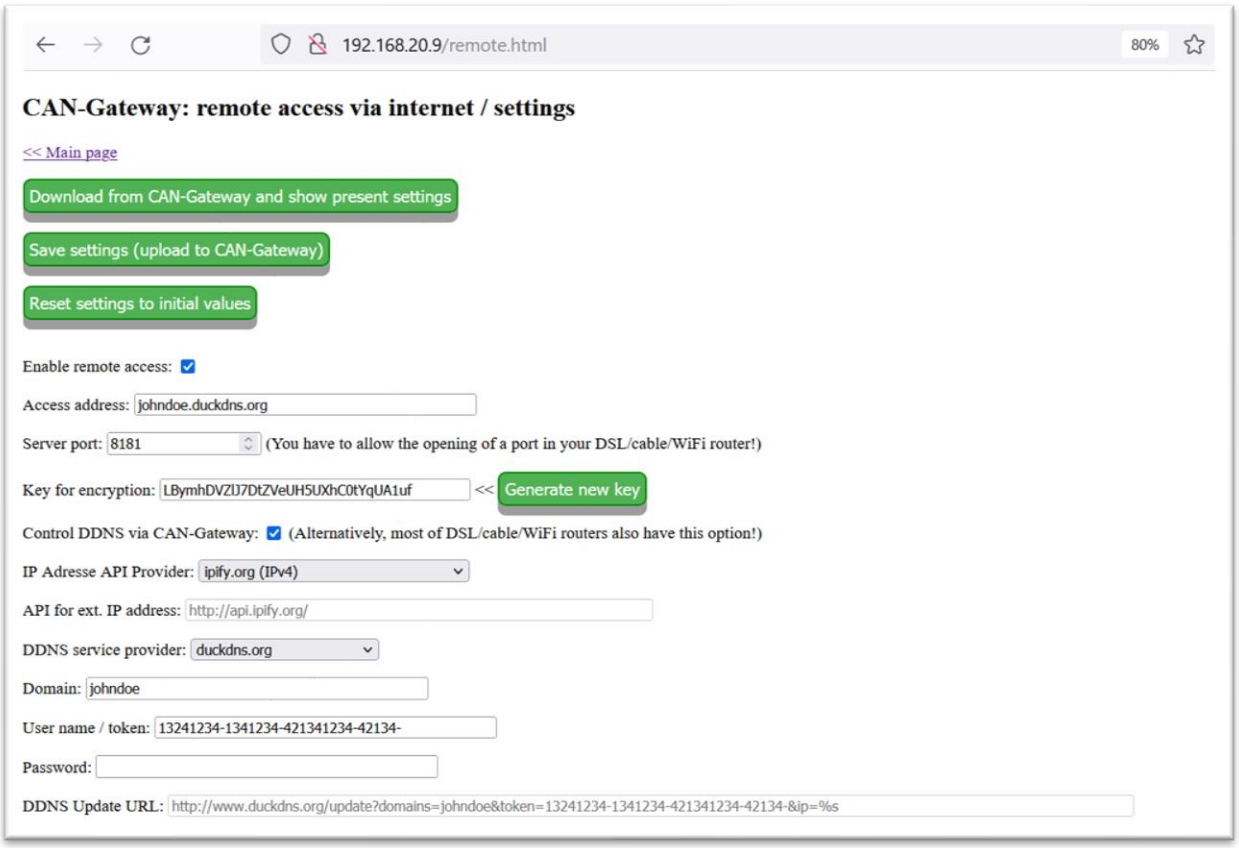
API for ext. IP address: is set automatically when the IP API provider is selected (unless you have selected "Set IP API URL manually", then you have the option of manually setting the IP API URL, which is intended for advanced users)

DDNS service provider: select a DDNS service that you registered with in step 1 and enter the domain, username / token and, if applicable, password. The domain is the address you selected in the first step (access address), but, in most cases, without the provider part. So e.g. "johndoe" without ".duckdns.org".

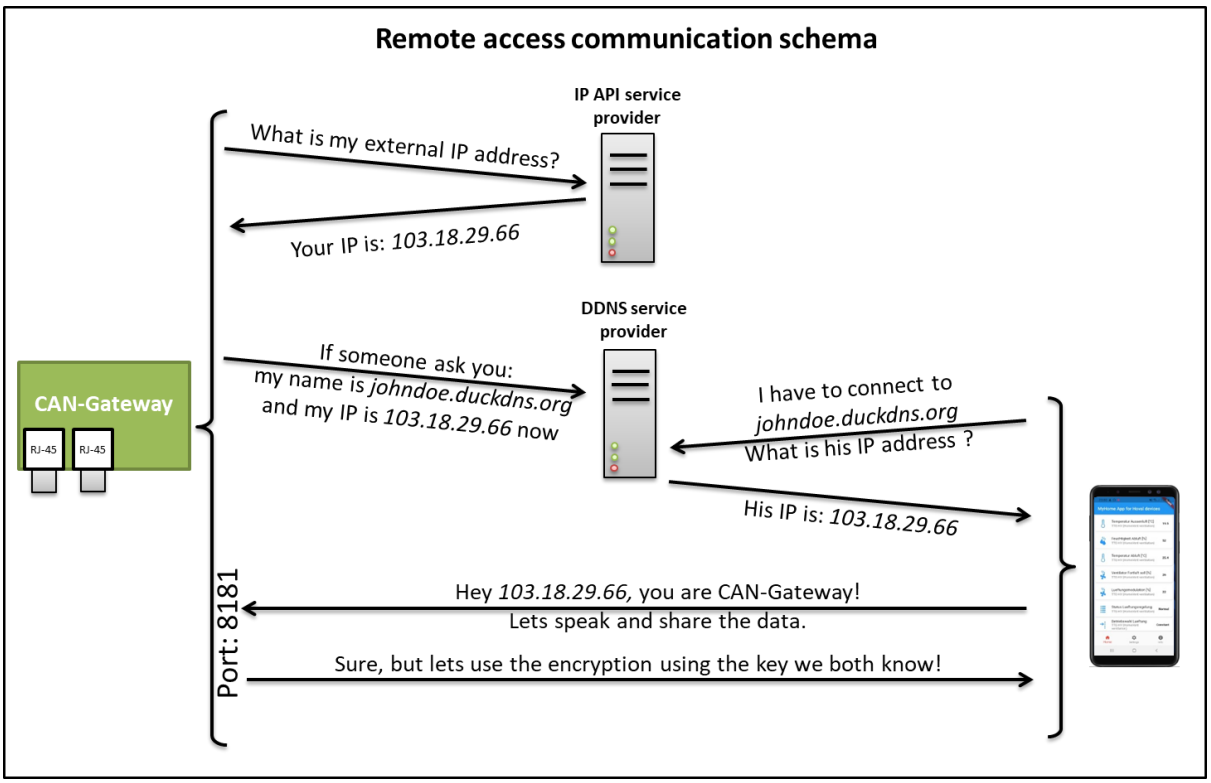
DDNS Update URL: is set automatically to match the DDNS settings (unless you have selected "Set update URL manually", which is intended for advanced users).

As soon as everything is set, you have to click on "Save settings (upload to CAN gateway)".

Note: CAN-Gateway only saves the resulting URLs internally, so after saving the settings or the next time you download the settings for remote access from the CAN-Gateway, the IP API and DDNS service selection is set to "manually". You don't have to change that again, the resulting URLs are already set correctly.



4) You can now open your CAN-Gateway Android/Windows app on your smartphone/PC/laptop/tablet. In order to use access via the Internet, the app must first be set up for access via WiFi / LAN. All you have to do is configure the internal IP address of the CAN gateway in "Settings" of the app. If the app is then running in the local WiFi / LAN network, you can click on "Get settings" in the "Remote access settings" area. The app then reads the required configurations from the CAN gateway (remote access address, port and key) via your local WiFi / LAN network. Now you can toggle the "Use remote access" switch below in Settings. You can then use your smartphone/PC/laptop/tablet and the CAN-Gateway app to control via the Internet and do not have to be connected to your local WiFi / LAN network.



28.2 Encryption details

AES-256 encryption in GCM mode is used for data transmission. This encryption is currently classified as very secure and automatically includes authentication. In addition, some other measures are implemented to avoid the attacks on the CAN-Gateway from Internet.

Please keep the key secret! If you suspect misuse, please generate a new key immediately!

29 Help!

29.1 Incorrect WiFi network name and/or network key are entered, I cannot reach Web interface. What to do?

If CAN-Gateway is configured with a wrong network name and/or network key (password), then the CAN-Gateway will not able to connect to your WiFi network and, as a consequence, you will not able to reach its Web interface. In this case you have 3 options:

1) The option to reset all settings of the CAN-Gateway to factory settings (including resetting it to access point mode with WiFi network name “cangateway” and WiFi network key “000999555”) using the room control unit (if you have one). This option is, however, is only available if your room control unit detects the CAN-Gateway. That requires the respective setting in the CAN-Gateway. Please refer to section “Control of the CAN-Gateway over the room control unit”.
2) **Only if software is used on a CAN gateway development board:** press and hold down the "REINIT" button for at least 6 seconds. CAN gateway resets its settings to the initial state and restarts as an access point with the name "cangateway".

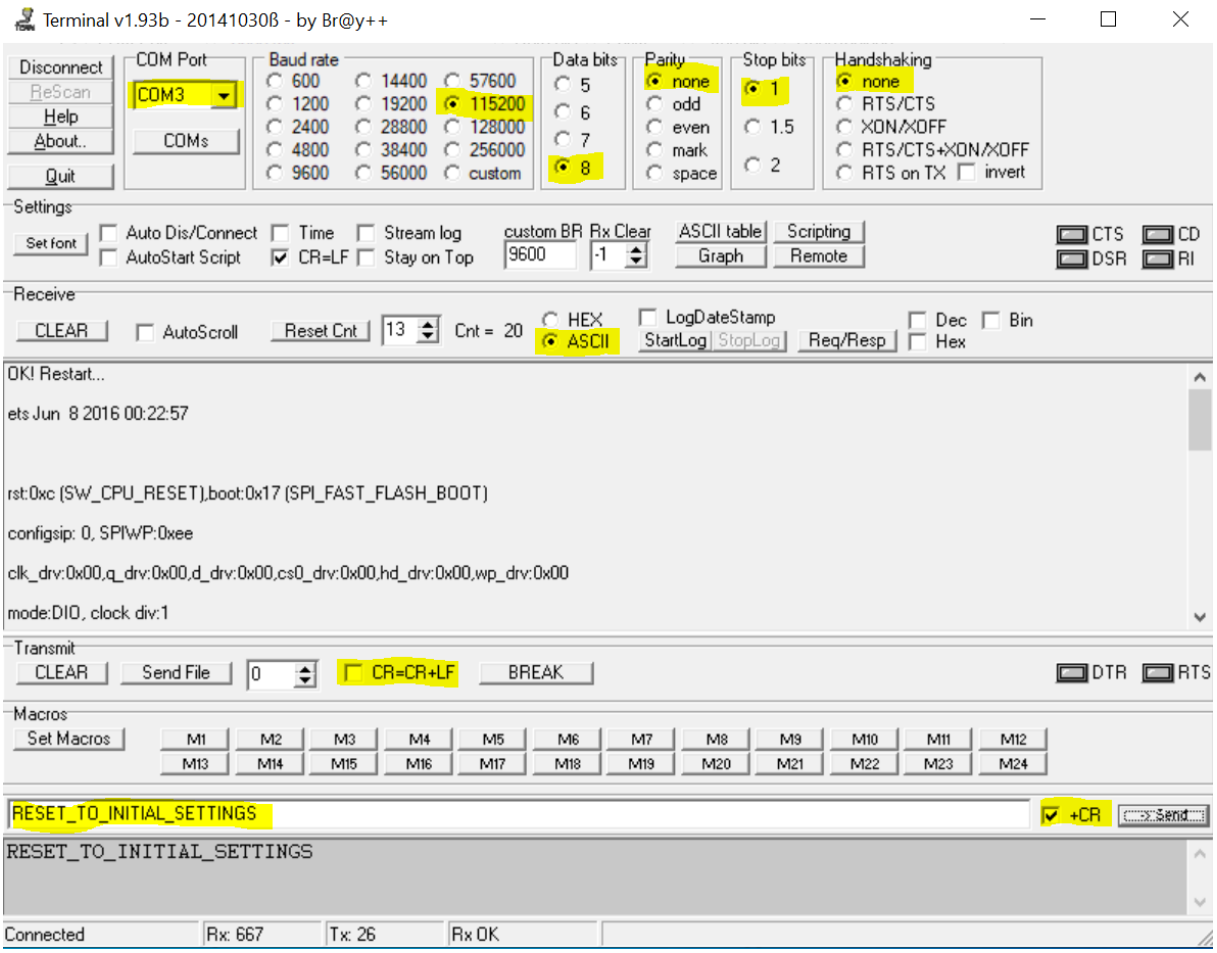
3) The option to reset either all settings or only WiFi network name and key to factory settings (cangateway, 000999555) using USB serial interface of the ESP32 board. For this please disconnect CAN-Gateway physically from your installation and connect it to the USB of your PC/notebook. Please refer to section “Software install” for details regarding USB drivers you will need. After that you have to use any terminal software that can connect to serial interfaces (COM-ports). Over the serial interface you have to send the character sequence „RESET_TO_INITIAL_SETTINGS” or „RESET_WLAN_SETTINGS” to the CAN-Gateway ESP32 board. Please take care that upper case letters must be used and at the end of character string you have to send a „Carriage Return” symbol (CR, ASCII Code = 13 (hexadecimal 0D)). CAN-Gateway will respond „OK! Restart...” and will restart with settings set back to factory values. It goes again to the access point mode (please refer to the section “First initialisation”). **Important**: If you connect your terminal software to the CAN-Gateway, you must select the correct baud rate. Initially it is 115200 baud, but you may have changed it. If you are not sure or the CAB-Gateway does not answer with “OK! Restart...” you might try different baud rates. After each try please repower CAN-Gateway, otherwise it might not recognize the reset command even if you choose the right baud rate.

[New in SW 26.001] If the CAN-Gateway with the currently set WiFi network name and network key has never been connected to WLAN, it only tries to connect to the network for approx. 45 seconds. If it doesn't work, it automatically switches to AP mode (name: cangateway, key: 000999555). So if you just entered the WiFi credentials incorrectly, you just have to wait about a minute until the CAN-Gateway goes into AP mode. However, if the CAN-Gateway had the correct WiFi credentials and you simply have another network, it will continue to try to connect to the network forever. In this case you have to use one of the two options described above.

In the following the usage of two different terminal programs is described:

29.1.1 Using Bray Terminal

1) Download, unzip and run the program : https://sites.google.com/site/terminalbpp/
2) Choose the correct COM port and baud rate. Other settings exactly as shown below.
3) Click on „Connect”.

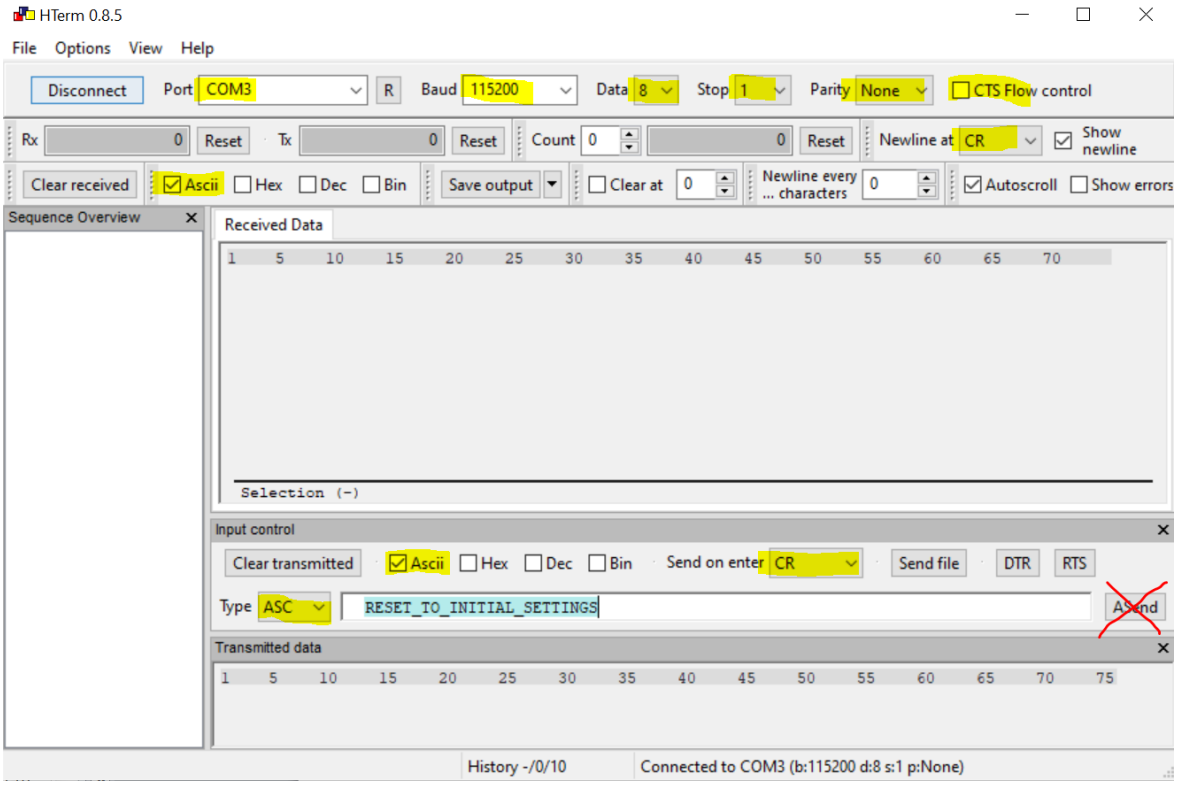


4) Enter „RESET_TO_INITIAL_SETTINGS” and then click on „->Send” button.
5) Cangateway answers „OK! Restart…” as shown in the picture above.
6) Click on „Disconnect” and exit the terminal program.

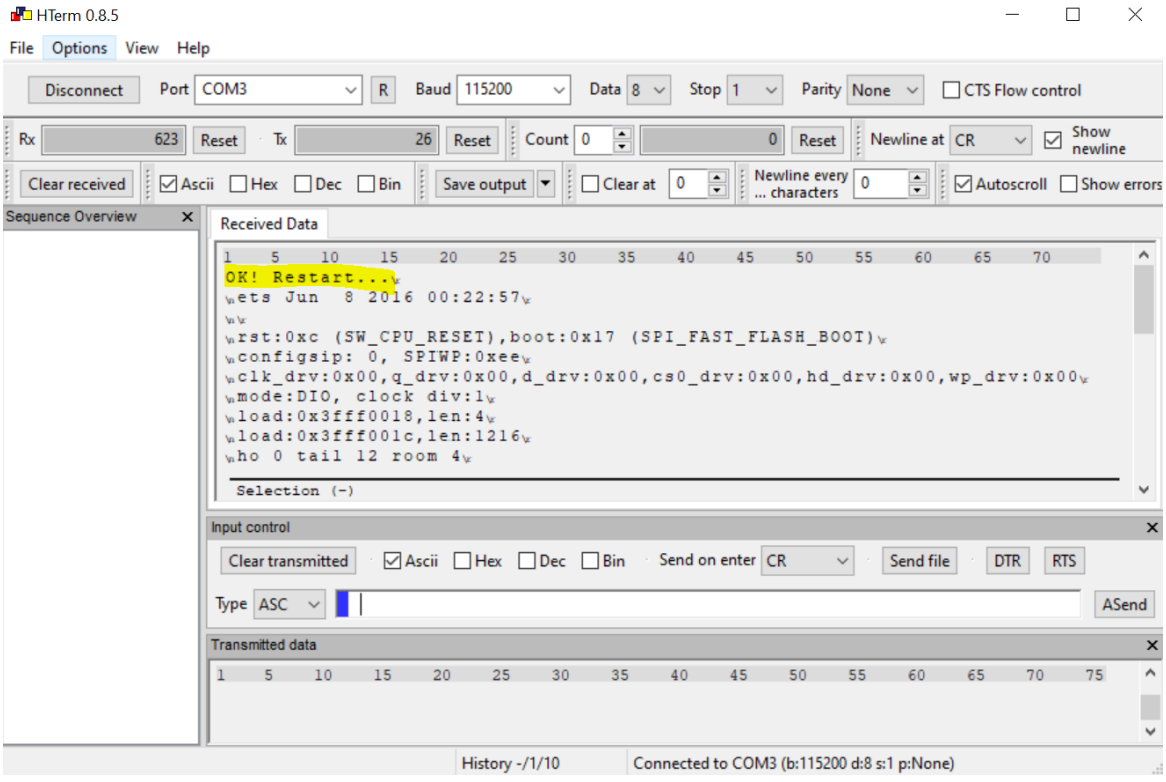
29.1.2 Mit HTerm

1) Download, unzip and run the program:
   http://www.der-hammer.info/pages/terminal.html
2) Choose the correct COM port and baud rate. Other settings exactly as shown below.
3) Click on „Connect”.

© W. Waag, Munich, 2020-2024



4) Enter „RESET_TO_INITIAL_SETTINGS” and then push the „Enter” key on your keyboard. DO NOT use „ASend” button.
5) Cangateway answers „OK! Restart…” as shown in the picture below. (eventually you need to scroll up to see it):



6) Click on „Disconnect” and exit the terminal program.

29.2 CAN-Gateway starts always with factory settings. Changes cannot be stored.

The most likely issue is that the hardware of your CAN-Gateway (flash) is corrupted. CAN-Gateway checks the correctness of the stored configuration performing CRC check. If this check fails, CAN-Gateway starts with factory settings. You have to replace the ESP32 board.

29.3 CAN-Gateway starts WiFi in access point mode, but I cannot connect my PC/notebook to the “cangateway” WiFi network

It might happens (especially with SW versions below 19.001) if you have already some settings changed and you device is physically connected to the CAN bus, but CAN-Gateway is set back to the access point mode by resetting the WiFi network name and password. You can try to disconnect CAN-Gateway physically from CAN bus and reboot (repower) it. The other option that always solves the issue is to reset all setting to the factory values using terminal program and using “RESET_TO_INITIAL_SETTINGS” as described above.

© W. Waag, Munich, 2020-2024

30 Notes regarding HomeVent comfort ventilation with BG02E control module

HomeVent devices from Hoval of the TTE type (type e.g. FR201, FR251, FR351, FRT351 etc.) are operated in a simple variant with a BG02E control unit (no touchscreen, two adjustment wheels for volume flow and humidity control). In this case you have to consider the following issue if you want to use a CAN gateway. BG02E control unit is quite "stupid", it simply sends the setpoint values for volume flow and humidity to the comfort ventilation system every second or so. If you connect a CAN gateway in this CAN network, you cannot control anything via the CAN gateway. More precisely, if the CAN gateway sends a setpoint for volume flow or humidity to comfort ventilation, it will be changed or "overwritten" in the next second by the BG02E control unit. It means practically, taht in this casek you can only use the CAN gateway to read out the parameters, but not for control. If you want to be able to control, you have to operate the comfort ventilation only with the CAN gateway and unplug the BG02E control unit. After all, you can also set and change the setpoints via the WEB interface from the CAN gateway (or using the MQTT protocol, etc.). If you also want to have a control unit from Hoval, you can only use the much more expensive Hoval control module with touchscreen. This unit namely (like the CAN gateway) sends the setpoints only once when they are changed, so it can work well in a connection with the CAN gateway.

Another issue regarding BG02E control unit: CAN-Gateway detects and show this unit on the “Web Interface: Device List” page as a TTE-WEZ device with Type=0 and Address = 0. It is just representation issue and does not have any influence on CAN-Gateway functionality.

[**New in SW version 27.100 and HW version 5**] CAN-Gateway hardware V5 has (OPTIONAL, if equipped) an additional CAN interface (implemented as an RJ45 socket), which is specially designed for connecting a BG02E operating module. This additional interface avoids the disadvantages mentioned above and the CAN-Gateway can be used together with a BG02E operating module without any drawbacks. It works in such a way that the CAN-Gateway only forwards the setpoints from the BG02E operating module (the volume flow and humidity setpoints as well as the party mode setting) to the ventilation unit if you have just turned the adjusting wheels or pressed the party button. The setpoints can also be changed via the CAN gateway too and are valid until they are "overwritten" by using BG02E module. The status LED on the BG02E shows - for at least 20 seconds after you have used the adjusting wheels or pressed the party button - the current status of the ventilation unit according to the original manufacturer's description. Otherwise it always lights up green when the BG02E is correctly connected to the CAN gateway.

31 Time setting in the CAN-Gateway [new in SW 22.001]

In order to get the correct time stamp, for example for logged data, the CAN gateway asks via the Internet the time server (see settings) for the current time. The query is carried out every hour. In between, the CAN gateway calculates the time itself.

If the time server cannot be reached at all, the CAN gateway uses a fictitious internal time that starts at every power-up with 01/01/1970, 00:00:00. This is the case, for example, if the time server is set incorrectly or if the CAN gateway is only operated in the local network without internet access.

If the CAN gateway has received the time from the time server at least once, it tries to calculate the time itself further. This also continues beyond a reset, i.e. as long as the power supply is available. However, since the internal clock is not particularly accurate, this time may deviate from the real time after many hours if the connection to the time server is no longer available. As soon as the connection is possible again, the CAN gateway will correct the internal time.

When displaying the historical (logged) data via WEB-Interface, the WEB-Interface tries to convert the times correctly to the current point in time when it determines that these are saved with a fictitious internal time (i.e. year 1970 etc.) an not with a real time.

32 Autorecovery

If the CAN-Gateway detects a serious problem, a reset is carried out automatically. This is a try to get rid of the problem. A serious problem can be, for example, that the WLAN connection becomes so marginal that it is constantly lost and reconnected. Or for example if a problems arise in communication with the MQTT server. Experience has shown that it can happen once every week. This is normal and does not limit functionality. If it occurs significantly more often, e.g. several times a day, you should check whether WiFi is good enough and, if necessary, check logging files of your MQTT server.

33 Integration into Home Assistant

33.1 Introduction

Home Assistant supports many different possibilities how external device might be integrated. CAN-Gateway has also different interfaces (MQTT, REST-API,..) that can be used.

In the following only one possibility is considered. Thereby it is assumed that:

1) Home Assistant is correctly installed and configured.
2) MQTT Broker is installed and integrated into Home Assistant for example as add-on: https://github.com/home-assistant/addons/blob/master/mosquitto/DOCS.md
3) CAN-Gateway is configured and connected to MQTT Broker (please refer to settings description in this operating manual).

© W. Waag, Munich, 2020-2024

33.2 Configuration

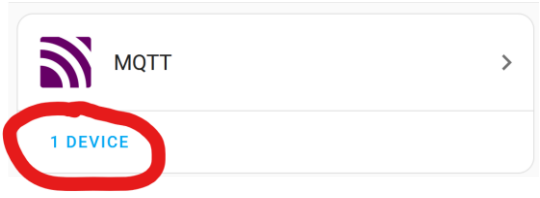
33.2.1 Automatic configuration using MQTT Discovery

CAN-Gateway supports automatic detection of all parameters in Home Assistant via MQTT. Up to CAN-Gateway version 29.326, all parameters are recognized as sensors. From version 29.327 a significantly improved MQTT Discovery support is implemented: Parameters are recognized either as sensors, as adjustable numerical values, as binary sensors, as switches or as selection lists and are automatically configured depending on how the parameter is configured in CAN-Gateway.

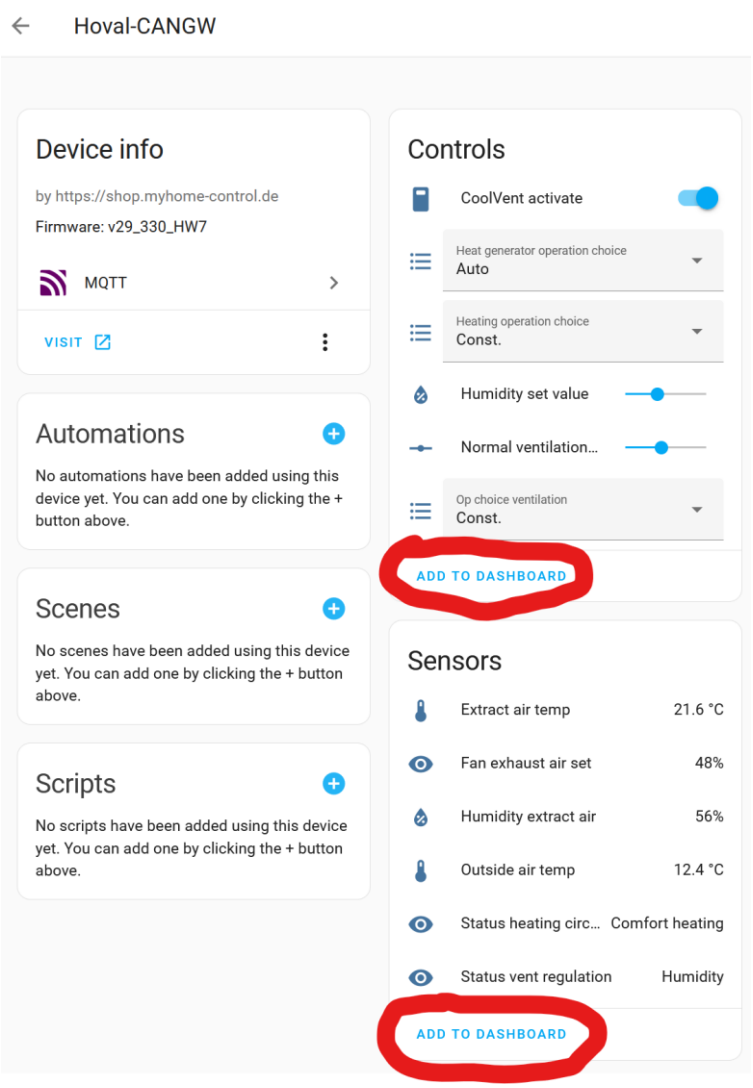
The assignment is as follows:

- Parameters that are set as a numeric type in CAN-Gateway (U8, S8, U16, S16, U32, S32) and are configured for reading and writing (attribute "w") are configured as adjustable numeric values. This applies, for example, to the adjustable target temperatures, target ventilation modulation or target humidity for the HomeVent, etc.
- Parameters are configured as binary sensors that are included in "Parameter configuration generator", can only have values "0" (off) or "1" (on) and in CAN gateway settings only for reading (attribute "r") are configured.
- Parameters are configured as binary sensors that are included in "Parameter configuration generator", can only have values "0" (off) or "1" (on) and in CAN -Gettings for reading and writing (attribute "w") are configured. An example is the activation of the CoolVent function of the HomeVent.
- Parameters are configured as selection lists, which are included in "Parameter configuration generator", can take values between 0 and 255, with each value being assigned a specific status and configured in CAN-Gateway settings for reading and writing (attribute "w"). Examples are the parameters "Operating mode ventilation" or "Operating mode heating".
- If none of the above apply, a parameter is configured as a sensor.

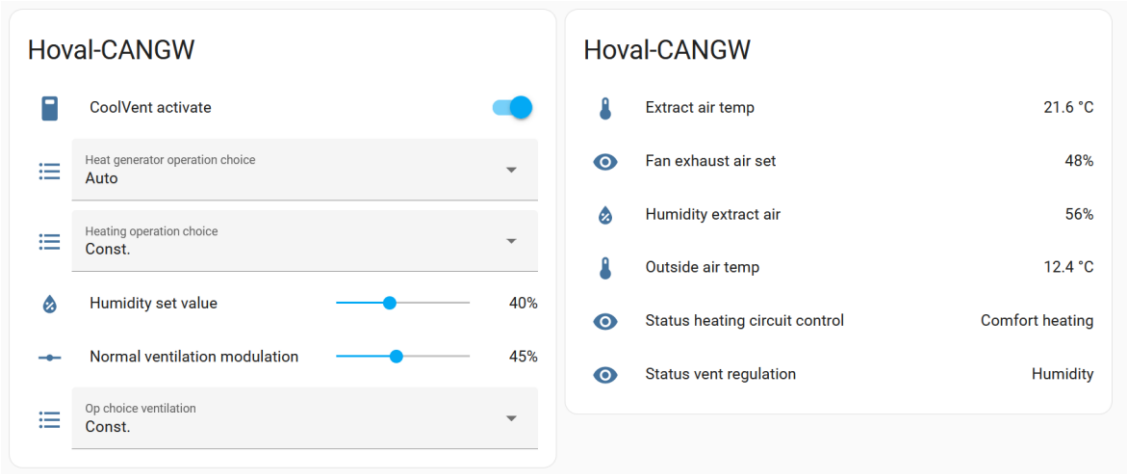
If the option "Support Home Assistant / MQTT Discovery" is activated in CAN-Gateway settings, the CAN-Gateway is connected to the MQTT Broker and the Home Assistant has set up the MQTT integration, then you can find the MQTT integration in Home Assistant under "Settings"-> "Devices & Services". It must show that at least one device and multiple entities are available. One of the devices is the CAN-Gateway (with name "Hoval-CANGW") and entities are parameters. Now you can add them to your dashboard.



© W. Waag, Munich, 2020-2024



After that you can see the sensors and control elements on your dashboard:



As usual in Home Assistant you can change now the names or icons of the elements.

The advantage of the automated configuration is that you do not have to set anything manually in Home Assistant. However, it only works correctly for the most commonly used parameters. Parameters that are not included in the configurator and that the user has set manually in CAN-Gateway may have to be configured in Home Assistant manually too, as described below.

33.2.2 Manual configuration

Please refer to https://www.home-assistant.io/integrations/mqtt/#manual-configured-mqtt-items. Advantage: you have full control. For example, the parameters can be displayed more beautifully or differently than MQTT Discovery allows. Home Assistant has extensive options for this. Settings must be made in the *configuration.yaml* file (please refer to https://www.home-assistant.io/getting-started/configuration/).

Example 1: (assumed you have configured one of the parameters in CAN-Gateway as:
param=8;8;50;0;39652;r;State_Ventilation_Control;10;U8;1.000000;0.000000;
):

```
mqtt:
    sensor:
      - name: "State ventilation"
        state_topic: "cangateway/State_Ventilation_Control"
        value_template: >-
              {% set mapper =  {
                  '0' : 'Off/Standby',
                  '1' : 'Normal',
                  '2' : 'VOC activated',
                  '3' : 'Humidity mode',
                  '4' : 'Freeze protection',
                  '5' : 'CoolVent',
                  '6' : 'Failure'} %}
              {{ mapper[value] if value in mapper else 'Unknown' }}
```

This example corresponds exactly to how MQTT Discovery would set this parameter. However, you can use the manual configuration to translate into another language, for example.

In the next example, a HomeVent is represented as a fan in Home Assistant. See the Home Assistant documentation for an explanation.

Example 2: assumed you have configured the following parameters in CAN-Gateway:

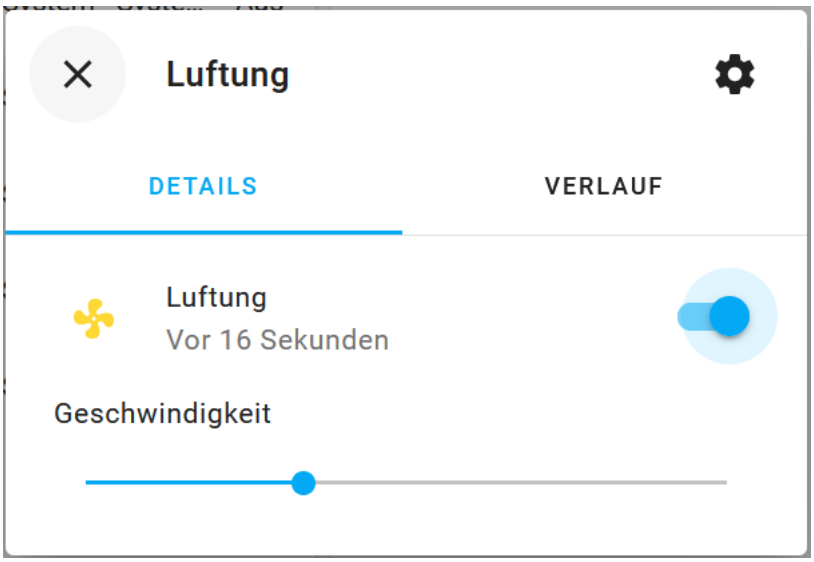
param=8;8;50;0;38606;r;Ventilation_modulation_percent;10;U8;1.000000;0.000000;
param=8;8;50;0;40650;w;Op_choice_ventilation;10;U8;1.000000;0.000000;
param=8;8;50;0;40651;w;Normal_ventilation_modulation_percent;10;U8;1.000000;0.000000;

Then the configuration in configuration.yaml can look like this:

```
mqtt:
    fan:
      - name: "Luftung"
        state_topic: "cangateway/ Op_choice_ventilation"
        command_topic: "cangateway/ Op_choice_ventilation/set"
        percentage_state_topic: "cangateway/ Ventilation_modulation_percent"
        percentage_command_topic: "cangateway/ Normal_ventilation_modulation_percent/set"
        payload_on: "4"
        payload_off: "0"
        speed_range_min: 1
        speed_range_max: 100
```

For the explanation of parameters please refer to https://www.home-assistant.io/integrations/fan.mqtt/

This allows you to switch the ventilation on and off and adjust the air volume (in new Home Assistant versions it looks slightly different):



The same control is also possible with automatic MQTT discovery, but is presented differently.

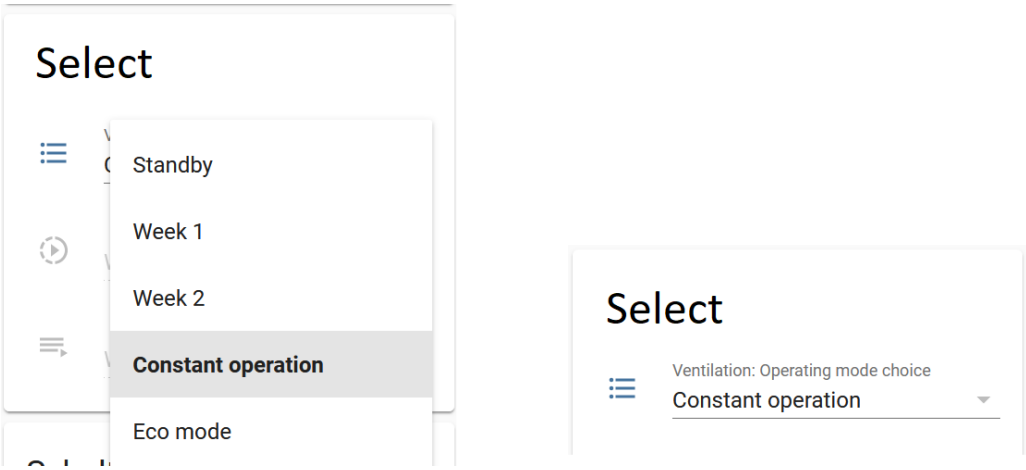
Example 3: Operation mode selection for ventilation. Assuming you have set up the following parameters in the CAN-Gateway:

param=8;8;50;0;40650;w;Op_choice_ventilation;10;U8;1.000000;0.000000;

According to the documentation (e.g. see http://www.hoval.com/misc/TTE/TTE-GW-Modbus-datapoints.xlsx), this parameter can have the following values: 0 (="Standby"), 1 (="Week 1 "), 2 (="Week 2"), 4 (="Constant mode"), 5 (="Eco mode"). You can implement it in Home Assistant, for example, as an MQTT “Select” element: insert the following in *configuration.yaml*:

```
mqtt:
    select:
      - command_topic: "cangateway/Op_choice_ventilation/set"
        command_template: >-
              {% set mapper =  {
                  'Standby' : '0',
                  'Week 1' : '1',
                  'Week 2' : '2',
                  'Constant operation' : '4',
                  'Eco mode' : '5'} %}
              {{ mapper[value] if value in mapper else 'Unknown' }}
        state_topic: "cangateway/Op_choice_ventilation"
        value_template: >-
              {% set mapper =  {
                  '0' : 'Standby',
                  '1' : 'Week 1',
                  '2' : 'Week 2',
                  '4' : 'Constant operation',
                  '5' : 'Eco mode'} %}
              {{ mapper[value] if value in mapper else 'Unknown' }}
        name: "Ventilation: Operating mode choice"
        options:
          - "Standby"
          - "Week 1"
          - "Week 2"
          - "Constant operation"
          - "Eco mode"
```

By using the templates, the values (0, 1, 2, 4, 5) that encode an operating mode are then displayed as understandable plain text and can be selected:



This example corresponds exactly to how MQTT Discovery would set this parameter. However, you can use the manual configuration to translate into another language, for example.

Example 4: Show current HomeVent operating mode. . Assuming you have set up the following parameters in the CAN-Gateway:

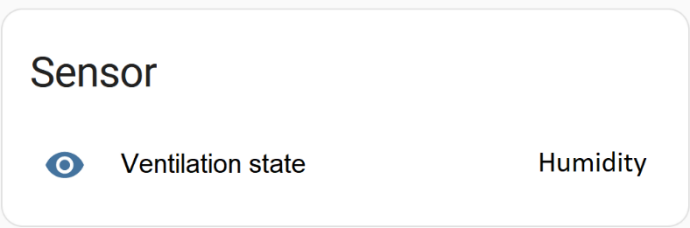
param=8;8;50;0;39652;r;Status_Ventilation;10;U8;1.000000;0.000000;

This parameter can have many numeric values. Each means certain operating state. According to the documentation (for example in http://www.hoval.com/misc/TTE/TTE-GW-Modbus-datapoints.xlsx or in CAN-Gateway WEB-Interface in section „Parameter List“ click to green „i“ next to this parameter) the meaning of values as follows: 0 = Standby, 1 = Normal ventilation, 2 = VOC, 3 = Humidity, 4 = Frost protection, 5 = CoolVent, 6 = Fault, 7 = Summer humidity.

This status can be implemented in Home Assistant, for example, as an MQTT "Sensor" element and configured so that the status is displayed in plain text instead of the numerical values: insert the following in *configuration.yaml*:

```
mqtt:
    sensor:
      - name: "Ventilation State"
        unique_id: "ID_custom003"
        state_topic: "cangateway/Status_Ventilation"
        value_template: >-
              {% set mapper =  {
                  '0' : 'Standby',
                  '1' : 'Normal ventilation',
                  '2' : 'VOC',
                  '3' : 'Humidity',
                  '4' : 'Frost protection',
                  '5' : 'CoolVent',
                  '6' : 'Fault',
                  '7' : 'Summer humidity'} %}
              {{ mapper[value] if value in mapper else 'Unbekannt' }}
```

The status is then displayed as follows:



This example corresponds exactly to how MQTT Discovery would set this parameter. However, you can use the manual configuration to translate into another language, for example.

34 Integration into OpenHab

OpenHab supports MQTT Discovery similar to Home Assistant. This makes it possible to set up the integration into OpenHab using the MQTT protocol with a few simple clicks. The necessary steps are summarized in the following instructions:

https://raw.githubusercontent.com/wladwnt/CAN-Gateway/master/OpenHab_configuration.pdf

Experienced users can also use other methods, such as integration via HTTP REST API.

35 Integration into Loxone

The integration into Loxone can be done in two alternative ways: as virtual HTTP inputs and outputs (rather recommended) or as Modbus TCP (rather not recommended). For both options, CAN-Gateway (since SW version 29.330) offers support in the form of a template generator. This generator generate XML templates for the parameters configured in the CAN-Gateway. These templates can then be imported into Loxone. This way you only have to manually edit a few optional settings (e.g. min/max values etc.) in Loxone.

35.1 Integration as virtual HTTP Inputs and Outputs

REST API is used on the CAN-Gateway side, which is described in Chapter 23 of this manual. Two REST API commands are used:

http://cangateway.local/json/getallvalues

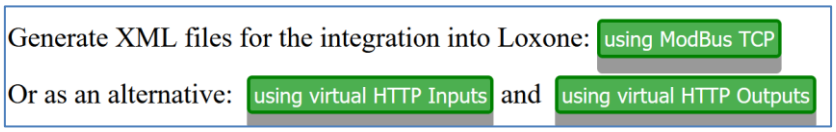
http://cangateway.local/setparam?num=XX

On the Loxone side, the virtual HTTP inputs and virtual HTTP outputs are used. The documentation for these possibilities can be found here:

- virtual HTTP Inputs: please refer to https://www.loxone.com/enen/kb/virtual-http-input/
- virtual HTTP Outputs: please refer to section „VIRTUAL OUTPUTS” (only outputs, inputs on this page is something different!) in https://www.loxone.com/enen/kb/virtual-inputs-outputs/

These can be configured manually or you can use an XML template for inputs and outputs that CAN-Gateway can generate. This is then imported into Loxone.

To generate a template, go to "Parameter list" in the CAN-Gateway web interface and then below the table to:



When you click on buttons "using virtual HTTP Inputs" and "using virtual HTTP Outputs" two files are generated: VI_can_gateway_template.xml and VO_can_gateway_template.xml. If no parameters are configured for writing, then the file for outputs (VO_can_gateway_template.xml) only contains the header and is otherwise empty.

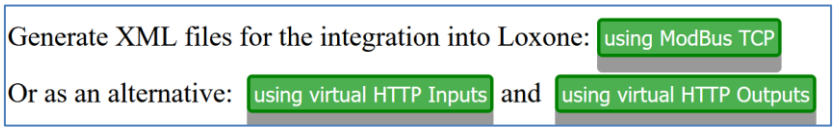
Warning: if you change the settings of the parameters in CAN-Gateway, these files will also change. You then have to regenerate them and import them into Loxone.

After these are imported into Loxone, some optional settings can/should be edited (e.g. min/max values, default values).

35.2 Integration as Modbus TCP device

This option is not recommended, since you have to set/change significantly more manually. But will be briefly described for completeness. The use of Modbus TCP on the CAN-Gateway is described in Chapter 21 of this manual. The use of the Modbus connections in Loxone can be found here: https://www.loxone.com/enen/kb/communication-with-modbus/.

To help with the setup, an XML template (MB_can_gateway_template.xml) can be generated in the CAN-Gateway Web Interface under "List of parameter values" and then under the table, which must then be imported into Loxone:



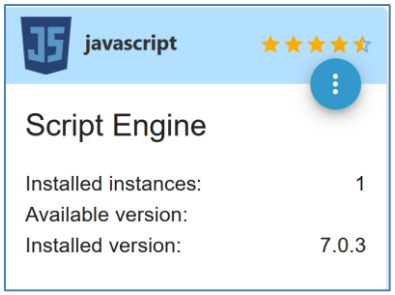
Warning: if you change the settings of the parameters in CAN-Gateway, this file will also change. You then have to regenerate it and import into Loxone again.

After this file has been imported into Loxone, some settings can/should be edited if necessary.

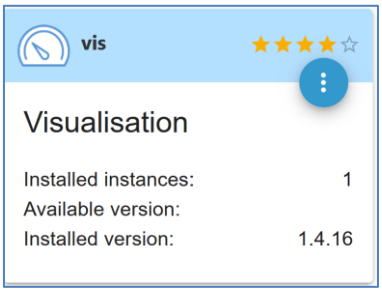
36 Integration into ioBroker

ioBroker offers various ways of integrating devices. The following describes one way of integrating CAN-Gateway into ioBroker using CAN-Gateway REST-API. Neither programming knowledge nor a deep understanding of ioBroker are necessary. The following steps have to be carried out:

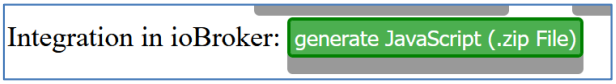
1) Install the JavaScript Adapter (Script Engine) in ioBroker under "Adapters" if it is not already installed:



2) Install the VIS Adapter (Visualisation) in ioBroker under "Adapters" if it is not already installed:

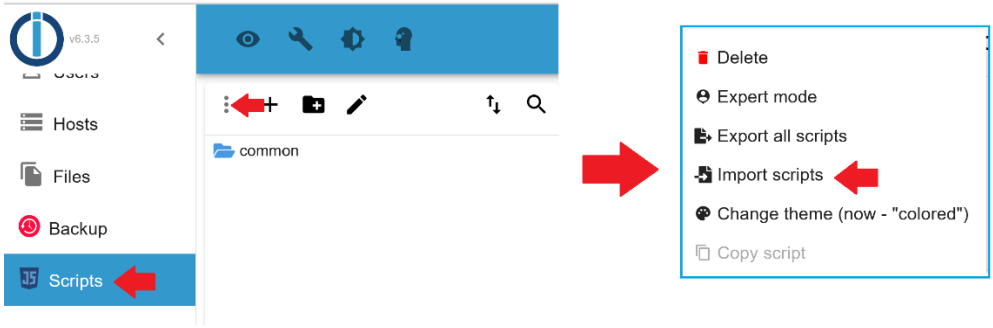


3) Create and download a .zip file in the CAN-Gateway WEB interface under "Parameter list" below the table with the button "generate JavaScript (.zip file)".

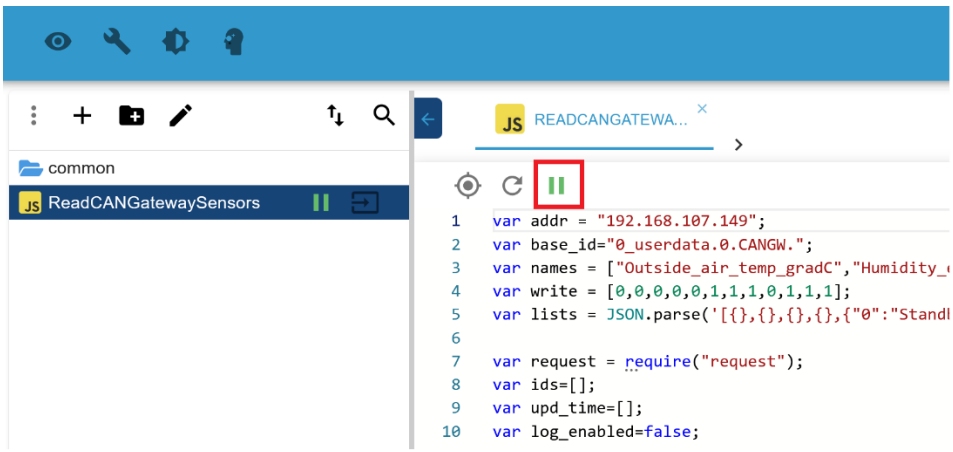


This file contains a JavaScript which can be imported into ioBroker in the next step. This script generates objects (states) in ioBroker for all parameters configured in CAN-Gateway. The script also ensures that these states are connected to the CAN gateway: it reads all parameter values from the CAN-Gateway every 10 seconds and sets the corresponding states in ioBroker. In addition, this script monitors parameters that can be written, whether the value is changed by ioBroker (e.g. by a widget or a Blockly program) and if there is a change, this change is sent to CAN-Gateway so that CAN-Gateway controls the Hoval device accordingly.

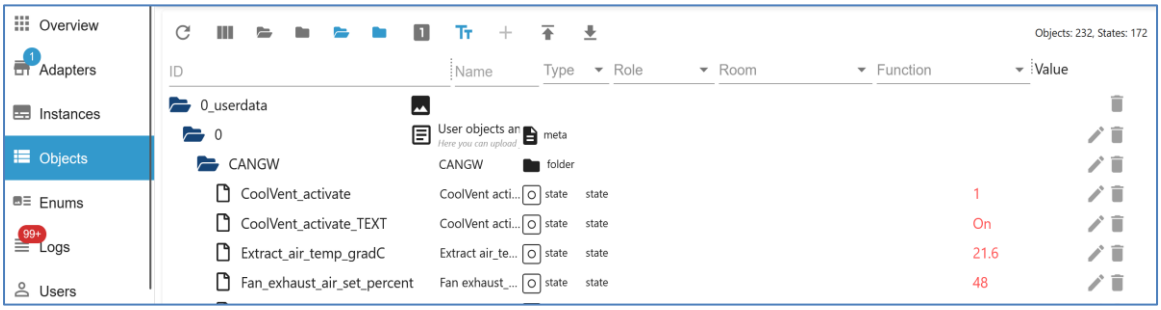
4) In ioBroker under "Scripts" via the menu (can be called up with three dots one below the other) click on "Import Scripts" and select the file generated by CAN-Gateway.



After confirming with "OK", the script is imported and executed. Now you have to see the generated objects under "Objects" in the area "0_userdata"->"0"->"CANGW".
Caution: each time settings are changed in CAN-Gateway, you must regenerate this script and import it again into ioBroker.
After the import, check if the script "ReadCAnGatewaySensors.js" is running: when double-clicking on the script name, you should see the content as well as the green pause sign at the top:

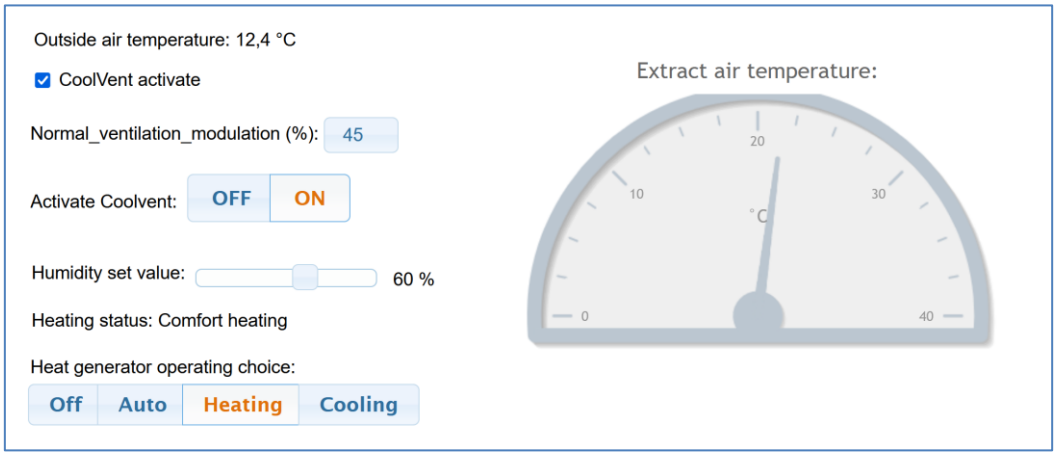


If everything is running, you will also see the corresponding parameter values in the "Value" column:



Note on parameters that represent a list: for parameters that represent a list (see Chapter 10.2 in this manual), two objects are created: one with the original ID/name and one with the addition "_TEXT" in the ID or " (textual)" by the name. This additional object contains the status in plain text and can be used more easily for a text display.

5) Visual elements can now be created in VIS and linked to the objects to show the parameter values or to change the writable parameters. Some simple examples are:



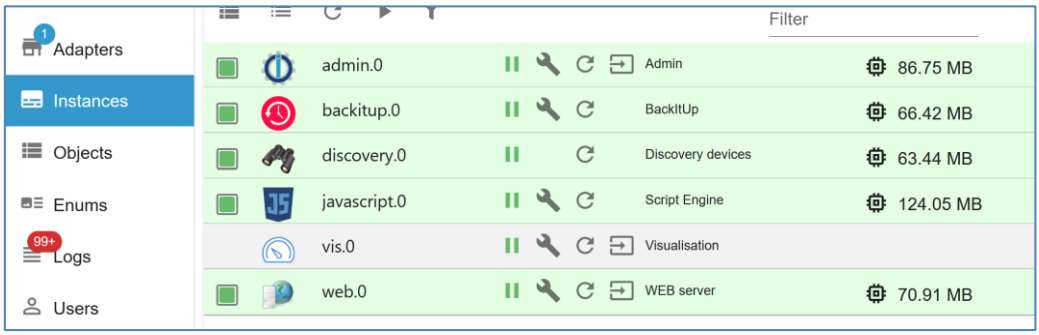
The steps described above are described in more detail with pictures in a separate file:

https://raw.githubusercontent.com/wladwnt/CAN-Gateway/master/ioBroker_configuration.pdf

36.1 Troubleshooting and tips for advanced ioBroker users

If something does not work as expected, you have the following options:

1) Check if the JavaScript Adapter is running: under "Instances" it must be green:



2) Check the first line in the script to see whether the IP address of the CAN gateway is correct. Change if necessary.
3) Check logs
4) In the script find line number 10 with "var log_enabled=false;" and write "true" instead of "false". Then look again in the logs: the script then writes some additional information in the logs that may help.

37 Limitations of the demo version

The demo version has following restrictions compared to the full version:

- Only two parameter can be configured instead of 40.
- The software shutdowns automatically after 60 minutes. After that the CAN-Gateway must be restarted manually (repowered).
- No user level passwords can be read out from the room control unit.
- No full access to online services (include an extended database with data points and error decoders).
- The “Read all data points of a device” function is not available (as it requires full access to online services).

Everything else works the same as in the full version. This gives you the opportunity to test everything before you buy a full version.

38 Over-The-Air update

The software of the CAN-Gateway includes Arduino OTA module, so it can be updated Over-The-Air using Arduino IDE. It means, among others, you will see in Arduino IDE cangateway as a board that can be updated. The hostname and password for OTA are both „cangateway“. It is, however, recommended to use update option in Web interface (please refer to section „Web interface: Software update“).

39 Licenses

This software is based, among others, on the following software components:

- Espressif IoT Development Framework (ESP-IDF, Copyright 2019 Espressif Systems (Shanghai) PTE LTD, Licensed under the Apache License, Version 2.0, Source: https://github.com/espressif/esp-idf)

© W. Waag, Munich, 2020-2024

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual,

worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

   (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

   (b) You must cause any modified files to carry prominent notices stating that You changed the files; and

   (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

   (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

   You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A

PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

- Arduino core for the ESP32 (For copyright information please refer to the source files. Licensed under GNU Lesser General Public License v2.1, Source: https://github.com/espressif/arduino-esp32)

```
### GNU LESSER GENERAL PUBLIC LICENSE

Version 2.1, February 1999

    Copyright (C) 1991, 1999 Free Software Foundation, Inc.
    51 Franklin Street, Fifth Floor, Boston, MA  02110-1301  USA

    Everyone is permitted to copy and distribute verbatim copies
    of this license document, but changing it is not allowed.

    [This is the first released version of the Lesser GPL.  It also counts
     as the successor of the GNU Library Public License, version 2, hence
     the version number 2.1.]

### Preamble

The licenses for most software are designed to take away your freedom
to share and change it. By contrast, the GNU General Public Licenses
are intended to guarantee your freedom to share and change free
software--to make sure the software is free for all its users.

This license, the Lesser General Public License, applies to some
specially designated software packages--typically libraries--of the
Free Software Foundation and other authors who decide to use it. You
can use it too, but we suggest you first think carefully about whether
this license or the ordinary General Public License is the better
strategy to use in any particular case, based on the explanations
below.

When we speak of free software, we are referring to freedom of use,
not price. Our General Public Licenses are designed to make sure that
you have the freedom to distribute copies of free software (and charge
for this service if you wish); that you receive source code or can get
it if you want it; that you can change the software and use pieces of
it in new free programs; and that you are informed that you can do
these things.

To protect your rights, we need to make restrictions that forbid
distributors to deny you these rights or to ask you to surrender these
rights. These restrictions translate to certain responsibilities for
you if you distribute copies of the library or if you modify it.
```

For example, if you distribute copies of the library, whether gratis or for a fee, you must give the recipients all the rights that we gave you. You must make sure that they, too, receive or can get the source code. If you link other code with the library, you must provide complete object files to the recipients, so that they can relink them with the library after making changes to the library and recompiling it. And you must show them these terms so they know their rights.

We protect your rights with a two-step method: (1) we copyright the library, and (2) we offer you this license, which gives you legal permission to copy, distribute and/or modify the library.

To protect each distributor, we want to make it very clear that there is no warranty for the free library. Also, if the library is modified by someone else and passed on, the recipients should know that what they have is not the original version, so that the original author's reputation will not be affected by problems that might be introduced by others.

Finally, software patents pose a constant threat to the existence of any free program. We wish to make sure that a company cannot effectively restrict the users of a free program by obtaining a restrictive license from a patent holder. Therefore, we insist that any patent license obtained for a version of the library must be consistent with the full freedom of use specified in this license.

Most GNU software, including some libraries, is covered by the ordinary GNU General Public License. This license, the GNU Lesser General Public License, applies to certain designated libraries, and is quite different from the ordinary General Public License. We use this license for certain libraries in order to permit linking those libraries into non-free programs.

When a program is linked with a library, whether statically or using a shared library, the combination of the two is legally speaking a combined work, a derivative of the original library. The ordinary General Public License therefore permits such linking only if the entire combination fits its criteria of freedom. The Lesser General Public License permits more lax criteria for linking other code with the library.

We call this license the "Lesser" General Public License because it does Less to protect the user's freedom than the ordinary General Public License. It also provides other free software developers Less of an advantage over competing non-free programs. These disadvantages are the reason we use the ordinary General Public License for many libraries. However, the Lesser license provides advantages in certain special circumstances.

For example, on rare occasions, there may be a special need to encourage the widest possible use of a certain library, so that it becomes a de-facto standard. To achieve this, non-free programs must be allowed to use the library. A more frequent case is that a free library does the same job as widely used non-free libraries. In this case, there is little to gain by limiting the free library to free software only, so we use the Lesser General Public License.

In other cases, permission to use a particular library in non-free programs enables a greater number of people to use a large body of free software. For example, permission to use the GNU C Library in non-free programs enables many more people to use the whole GNU operating system, as well as its variant, the GNU/Linux operating system.

Although the Lesser General Public License is Less protective of the users' freedom, it does ensure that the user of a program that is linked with the Library has the freedom and the wherewithal to run that program using a modified version of the Library.

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a "work based on the library" and a "work that uses the library". The former contains code derived from the library, whereas the latter must be combined with the library in order to run.

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any software library or other program which contains a notice placed by the copyright holder or

other authorized party saying it may be distributed under the terms of this Lesser General Public License (also called "this License"). Each licensee is addressed as "you".

A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.

The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is included without limitation in the term "modification".)

"Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.

1. You may copy and distribute verbatim copies of the Library's complete source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

- **a)** The modified work must itself be a software library.
- **b)** You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.
- **c)** You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.
- **d)** If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.

  (For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the application. Therefore, Subsection 2d requires that any application-supplied function or table used by this function must be optional: if the application does not supply it, the square root function must still compute square roots.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Library, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Library, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Library.

In addition, mere aggregation of another work not based on the Library with the Library (or with a work based on the Library) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may opt to apply the terms of the ordinary GNU General Public License instead of this License to a given copy of the Library. To do this, you must alter all the notices that refer to this License, so that they refer to the ordinary GNU General Public License, version 2, instead of to this License. (If a newer version than version 2 of the ordinary GNU General Public License has appeared, then you can specify that version instead if you wish.) Do not make any other change in these notices.

Once this change is made in a given copy, it is irreversible for that copy, so the ordinary GNU General Public License applies to all subsequent copies and derivative works made from that copy.

This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

4. You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange.

If distribution of object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place satisfies the requirement to distribute the source code, even though third parties are not compelled to copy the source along with the object code.

5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a "work that uses the Library". Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.

However, linking a "work that uses the Library" with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a "work that uses the library". The executable is therefore covered by this License. Section 6 states terms for distribution of such executables.

When a "work that uses the Library" uses material from a header file that is part of the Library, the object code for the work may be a derivative work of the Library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The threshold for this to be true is not precisely defined by law.

If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small inline functions (ten lines or less in length), then the use of the object file is unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the Library will still fall under Section 6.)

Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Section 6. Any executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.

6. As an exception to the Sections above, you may also combine or link a "work that uses the Library" with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit modification of the work for the customer's own use and reverse engineering for debugging such modifications.

You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by

this License. You must supply a copy of this License. If the work during execution displays copyright notices, you must include the copyright notice for the Library among them, as well as a reference directing the user to the copy of this License. Also, you must do one of these things:

- **a)** Accompany the work with the complete corresponding machine-readable source code for the Library including whatever changes were used in the work (which must be distributed under Sections 1 and 2 above); and, if the work is an executable linked with the Library, with the complete machine-readable "work that uses the Library", as object code and/or source code, so that the user can modify the Library and then relink to produce a modified executable containing the modified Library. (It is understood that the user who changes the contents of definitions files in the Library will not necessarily be able to recompile the application to use the modified definitions.)
- **b)** Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (1) uses at run time a copy of the library already present on the user's computer system, rather than copying library functions into the executable, and (2) will operate properly with a modified version of the library, if the user installs one, as long as the modified version is interface-compatible with the version that the work was made with.
- **c)** Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.
- **d)** If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.
- **e)** Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed for reproducing the executable from it. However, as a special exception, the materials to be distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you distribute.

7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined library, provided that the separate distribution of the work based on the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:

- **a)** Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the Sections above.
- **b)** Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

8. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, link with, or distribute the Library is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

9. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Library or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Library (or any work based on the Library), you indicate your acceptance of this License to do so, and

all its terms and conditions for copying, distributing or modifying the Library or works based on it.

10. Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a license from the original licensor to copy, distribute, link with or modify the Library subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties with this License.

11. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Library at all. For example, if a patent license would not permit royalty-free redistribution of the Library by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply, and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Library under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

13. The Free Software Foundation may publish revised and/or new versions of the Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Library does not specify a license version number, you may choose any version ever published by the Free Software Foundation.

14. If you wish to incorporate parts of the Library into other free programs whose distribution conditions are incompatible with these, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR

```
OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY
KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE
LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME
THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

**16.** IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN
WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY
AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU
FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR
CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE
LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING
RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A
FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF
SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH
DAMAGES.
```

- AsyncTCP Library (Copyright 2016 Hristo Gochkov, Licensed under GNU Lesser General Public License v3.0, Source: https://github.com/me-no-dev/AsyncTCP)

GNU LESSER GENERAL PUBLIC LICENSE
Version 3, 29 June 2007

Copyright (C) 2007 Free Software Foundation, Inc. <http://fsf.org/>
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

This version of the GNU Lesser General Public License incorporates the terms and conditions of version 3 of the GNU General Public License, supplemented by the additional permissions listed below.

0. Additional Definitions.

As used herein, "this License" refers to version 3 of the GNU Lesser General Public License, and the "GNU GPL" refers to version 3 of the GNU General Public License.

"The Library" refers to a covered work governed by this License, other than an Application or a Combined Work as defined below.

An "Application" is any work that makes use of an interface provided by the Library, but which is not otherwise based on the Library. Defining a subclass of a class defined by the Library is deemed a mode of using an interface provided by the Library.

A "Combined Work" is a work produced by combining or linking an Application with the Library. The particular version of the Library with which the Combined Work was made is also called the "Linked Version".

The "Minimal Corresponding Source" for a Combined Work means the Corresponding Source for the Combined Work, excluding any source code for portions of the Combined Work that, considered in isolation, are based on the Application, and not on the Linked Version.

The "Corresponding Application Code" for a Combined Work means the object code and/or source code for the Application, including any data and utility programs needed for reproducing the Combined Work from the Application, but excluding the System Libraries of the Combined Work.

1. Exception to Section 3 of the GNU GPL.

You may convey a covered work under sections 3 and 4 of this License without being bound by section 3 of the GNU GPL.

2. Conveying Modified Versions.

If you modify a copy of the Library, and, in your modifications, a facility refers to a function or data to be supplied by an Application that uses the facility (other than as an argument passed when the

facility is invoked), then you may convey a copy of the modified
version:

a) under this License, provided that you make a good faith effort to
ensure that, in the event an Application does not supply the
function or data, the facility still operates, and performs
whatever part of its purpose remains meaningful, or

b) under the GNU GPL, with none of the additional permissions of
this License applicable to that copy.

3. Object Code Incorporating Material from Library Header Files.

The object code form of an Application may incorporate material from
a header file that is part of the Library. You may convey such object
code under terms of your choice, provided that, if the incorporated
material is not limited to numerical parameters, data structure
layouts and accessors, or small macros, inline functions and templates
(ten or fewer lines in length), you do both of the following:

a) Give prominent notice with each copy of the object code that the
Library is used in it and that the Library and its use are
covered by this License.

b) Accompany the object code with a copy of the GNU GPL and this license
document.

4. Combined Works.

You may convey a Combined Work under terms of your choice that,
taken together, effectively do not restrict modification of the
portions of the Library contained in the Combined Work and reverse
engineering for debugging such modifications, if you also do each of
the following:

a) Give prominent notice with each copy of the Combined Work that
the Library is used in it and that the Library and its use are
covered by this License.

b) Accompany the Combined Work with a copy of the GNU GPL and this license
document.

c) For a Combined Work that displays copyright notices during
execution, include the copyright notice for the Library among
these notices, as well as a reference directing the user to the
copies of the GNU GPL and this license document.

d) Do one of the following:

0) Convey the Minimal Corresponding Source under the terms of this
License, and the Corresponding Application Code in a form
suitable for, and under terms that permit, the user to
recombine or relink the Application with a modified version of
the Linked Version to produce a modified Combined Work, in the
manner specified by section 6 of the GNU GPL for conveying
Corresponding Source.

1) Use a suitable shared library mechanism for linking with the
Library. A suitable mechanism is one that (a) uses at run time
a copy of the Library already present on the user's computer
system, and (b) will operate properly with a modified version
of the Library that is interface-compatible with the Linked
Version.

e) Provide Installation Information, but only if you would otherwise
be required to provide such information under section 6 of the
GNU GPL, and only to the extent that such information is
necessary to install and execute a modified version of the
Combined Work produced by recombining or relinking the
Application with a modified version of the Linked Version. (If
you use option 4d0, the Installation Information must accompany
the Minimal Corresponding Source and Corresponding Application
Code. If you use option 4d1, you must provide the Installation
Information in the manner specified by section 6 of the GNU GPL

for conveying Corresponding Source.)

5. Combined Libraries.

You may place library facilities that are a work based on the Library side by side in a single library together with other library facilities that are not Applications and are not covered by this License, and convey such a combined library under terms of your choice, if you do both of the following:

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities, conveyed under the terms of this License.

b) Give prominent notice with the combined library that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

6. Revised Versions of the GNU Lesser General Public License.

The Free Software Foundation may publish revised and/or new versions of the GNU Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library as you received it specifies that a certain numbered version of the GNU Lesser General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that published version or of any later version published by the Free Software Foundation. If the Library as you received it does not specify a version number of the GNU Lesser General Public License, you may choose any version of the GNU Lesser General Public License ever published by the Free Software Foundation.

If the Library as you received it specifies that a proxy can decide whether future versions of the GNU Lesser General Public License shall apply, that proxy's public statement of acceptance of any version is permanent authorization for you to choose that version for the Library.

- Asynchronous WebServer library for Espressif MCUs (Copyright 2016 Hristo Gochkov, Licensed under GNU Lesser General Public License v2.1, Source: https://github.com/me-no-dev/ESPAsyncWebServer)

```
Copyright (c) 2016 Hristo Gochkov. All rights reserved.
  This file is part of the esp8266 core for Arduino environment.

  This library is free software; you can redistribute it and/or
  modify it under the terms of the GNU Lesser General Public
  License as published by the Free Software Foundation; either
  version 2.1 of the License, or (at your option) any later version.

  This library is distributed in the hope that it will be useful,
  but WITHOUT ANY WARRANTY; without even the implied warranty of
  MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
  Lesser General Public License for more details.
```

- ModbusRTU and ModbusIP Master-Slave Library for ESP8266/ESP32 v3.0 (Copyright 2015, Andre Sarmento Barbosa, 2017 Alexander Emelianov (a.m.emelianov@gmail.com), Licensed under License https://github.com/emelianov/modbus-esp8266/blob/master/LICENSE.txt, Source: https://github.com/emelianov/modbus-esp8266)

```
Copyright (c) 2015, Andre Sarmento Barbosa
              2017, Alexander Emelianov (a.m.emelianov@gmail.com)
All rights reserved.


Redistribution and use in source and binary forms, with or without modification,
are permitted provided that the following conditions are met:
```

```
1. Redistributions of source code must retain the above copyright notice, this
list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this
list of conditions and the following disclaimer in the documentation and/or other
materials provided with the distribution.

3. Neither the name of the copyright holder nor the names of its contributors may
be used to endorse or promote products derived from this software without
specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND
ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED
WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.
IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT,
INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR
PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY,
WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE
POSSIBILITY OF SUCH DAMAGE.
```

- Dygraphs: JavaScript charting library, Copyright 2017 Dan Vanderkam (danvdk@gmail.com), Licensed under The MIT License (MIT), Source: https://github.com/danvk/dygraphs

```
Copyright (c) 2009 Dan Vanderkam

Permission is hereby granted, free of charge, to any person
obtaining a copy of this software and associated documentation
files (the "Software"), to deal in the Software without
restriction, including without limitation the rights to use,
copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the
Software is furnished to do so, subject to the following
conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES
OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT
HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY,
WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING
FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR
OTHER DEALINGS IN THE SOFTWARE.
```

- SSDP library: Copyright (c) 2015 Hristo Gochkov, Source und License: https://github.com/luc-github/ESP32SSDP

```
ESP32 Simple Service Discovery Copyright (c) 2015 Hristo Gochkov
Original (Arduino) version by Filippo Sallemi, July 23, 2014. Can be
found at: https://github.com/nomadnt/uSSDP

Permission is hereby granted, free of charge, to any person obtaining a
copy of this software and associated documentation files (the
"Software"), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be included
in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS
OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY
CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,
TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
```

- KNX IP library: Copyright (c) 2018 Nico Weichbrodt, Source und License: https://github.com/envy/esp-knx-ip

MIT License

Copyright (c) 2018 Nico Weichbrodt <nico@weichbrodt.me>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.