

Operating manual CAN-Gateway software

SW Version 19.001

Update 4 (22.10.2020)

Table of contents

Important notes – please read carefully!	2
Introduction	4
Required hardware and its assembly	5
How to flash the software	7
First initialisation	12
Web interface: Main page	16
General information about devices and their parameters	17
Example	18
User Levels	19
Web interface: General settings	20
Web interface: Parameter settings	21
Data types	23
Hints regarding the lists	24
Examples	24
Example number 1: Ventilation (HomeVent) – read and write humidity set value	24
Example number 2: Ventilation (HomeVent) – extract air temperature	24
Example number 3: Ventilation (HomeVent) – outside air temperature	24
Example number 4: Room control unit – read and write password for user level 7	24
Example number 5: Ventilation (HomeVent) – operating mode for ventilation	25
Web interface: Generator for parameter configuration	26
Web interface: Parameter list	27
Web interface: Device list	28
Web interface: Status information	29
Web interface: CAN data logger	30
Web interface: Parameter logger	31
Web interface: Software update	32
Parameter transmission using MQTT protocol	33

Read/Write parameter values over Modbus TCP	34
Notes regarding Modbus TCP implementation.....	34
Modbus addresses	34
Register types	34
Transmitted values	34
Control of the CAN-Gateway using room control unit.....	36
REST-API	38
Special notes regarding changes of parameters	39
Help!	39
Incorrect WiFi network name and/or network key are entered, I cannot reach Web interface. What to do?	39
CAN-Gateway starts always with factory settings. Changes cannot be stored.....	39
CAN-Gateway starts WiFi in access point mode, but I cannot connect my PC/notebook to the “cangateway” WiFi network.....	40
Limitations of the demo version	40
Over-The-Air update.....	40
Licenses	40

Important notes – please read carefully!

This software is a test and learning software. It is suitable only for testing and learning and is not intended for daily usage. Electronic units that use this software as well as other parts connected to it via busses must be used carefully and only under continuously supervision of a qualified person. This person must have a sufficient knowledge and experience with electronic components and be able to recognise failures and dangerous situations. It must also have possibility and be able to switch off all components within of some seconds to prevent hazardous situations.

This software and its documentation is provided „as is“ and without any warranty regarding its function and correctness. The author is not responsible for any direct or indirect damages that can be caused by this software, except situations that are regulated by German law (gross negligence, wilful deceit). The user of this software is fully responsible for its usage.

This software is based on an analysis of the communication between devices produced by Hoval Company (especially WLAN gateway, room control unit, ventilation system with software version 2.xx.xx) and not based on the official information from the manufacture. Therefore, it must be assumed that this software might not be applicable for all types and versions of devices and might be incomplete and/or faultily.

The usage of this software in connection with other devices might lead to a loss of warranty of those devices. The user is responsible to check for himself the respective rules by the device supplier.

The Hoval company itself sell a commercial gateway solutions for connection their devices to home automatization system via KNX or Modbus (RTU/TCP). The author of this software recommends those commercial gateways for normal daily usage.

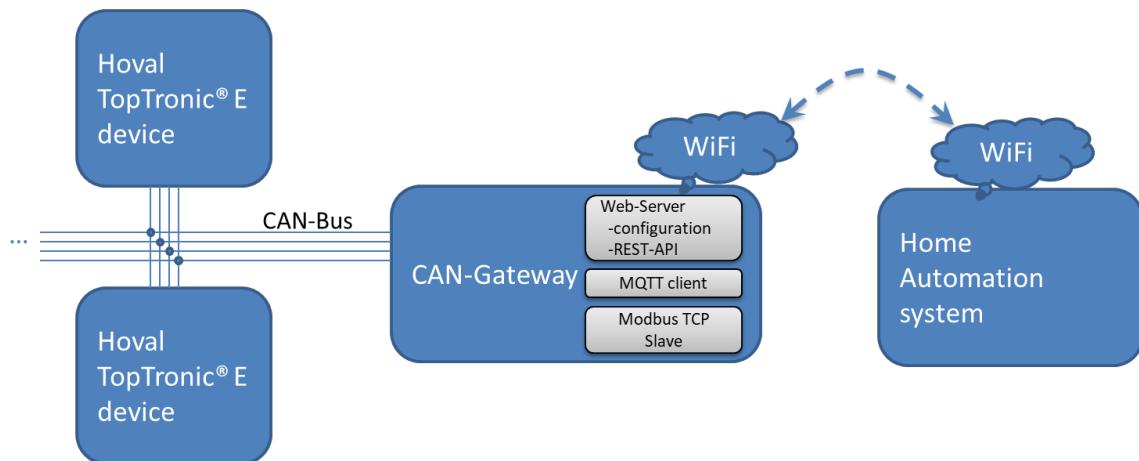
Introduction

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

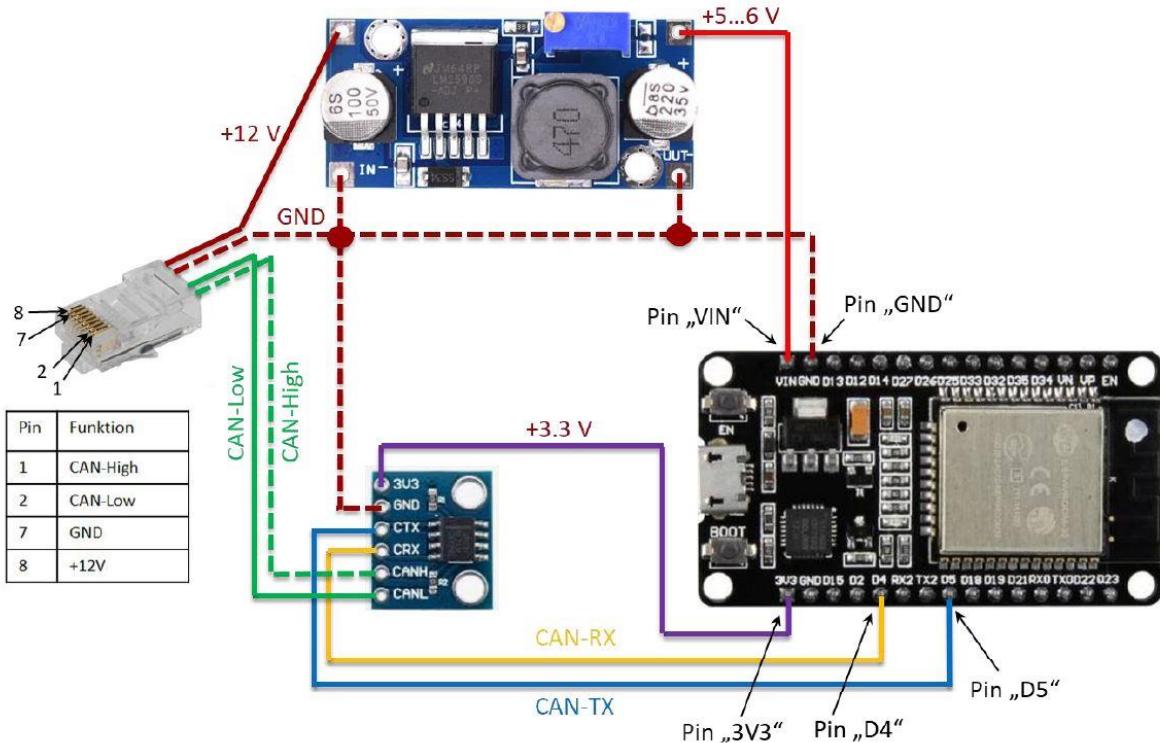
Required hardware and its assembly

The following parts are required:

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!

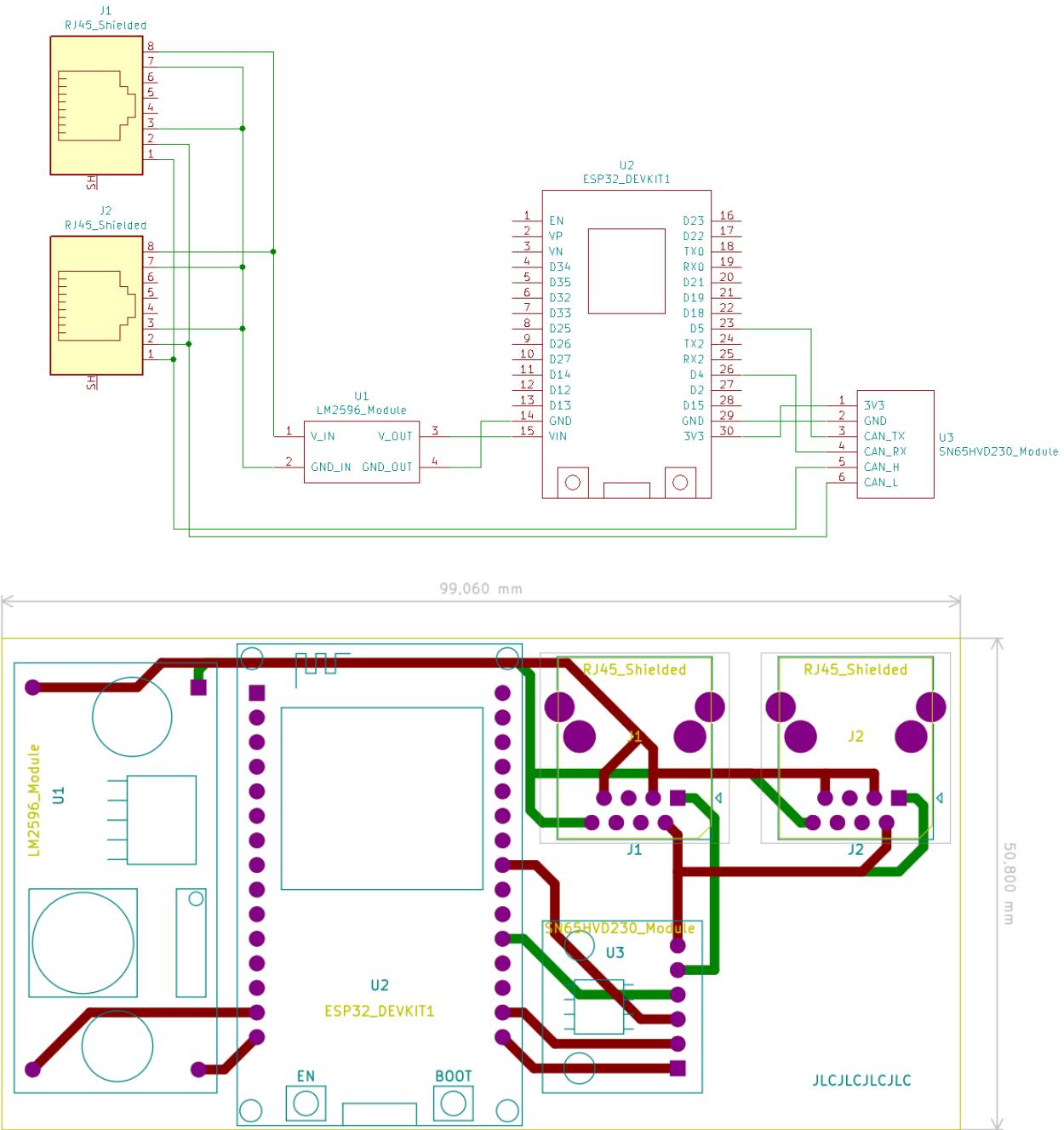
The assembly is as follows. **Attention: LM2596 module must be adjusted first to 5V output voltage.** This is done by the screw in the adjustable resistance (light blue on the picture below) and measuring the output voltage by a multimeter. ESP32 module can work for a short time at 12 V, but gets hot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



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, your CAN bus is "overterminated" and you have to switch off some termination resistances. If you measure resistance significantly higher than 60 Ohm, your 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).

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.



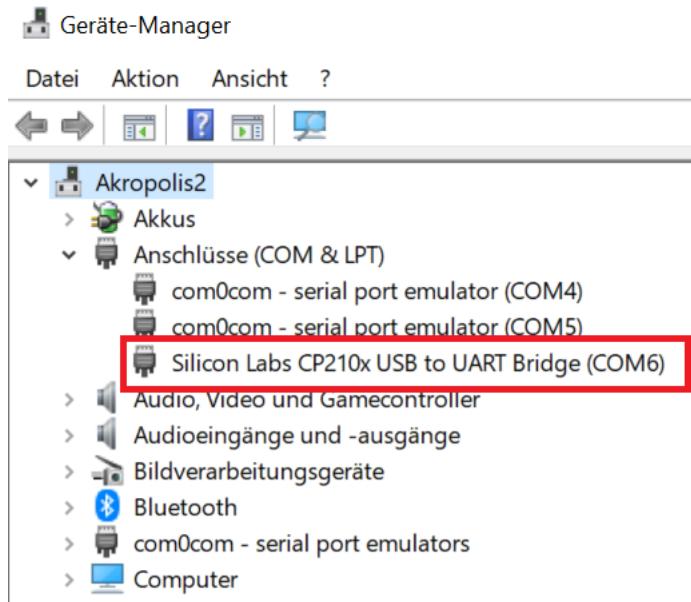
How to flash the software

The software is provided as a set of binaries and must be first installed on ESP32 DevKit 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):

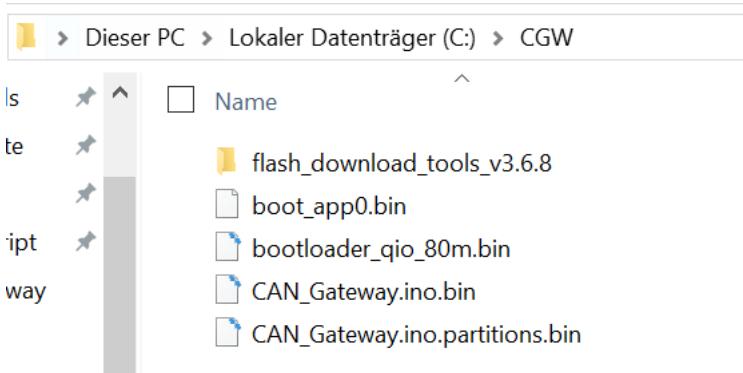
- 1) Find out the required USB driver. The major of ESP32 DevKit boards use CP210x USB communication chip. The driver for this chip can be found there:

<https://www.silabs.com/products/development-tools/software/usb-to-uart-bridge-vcp-drivers>

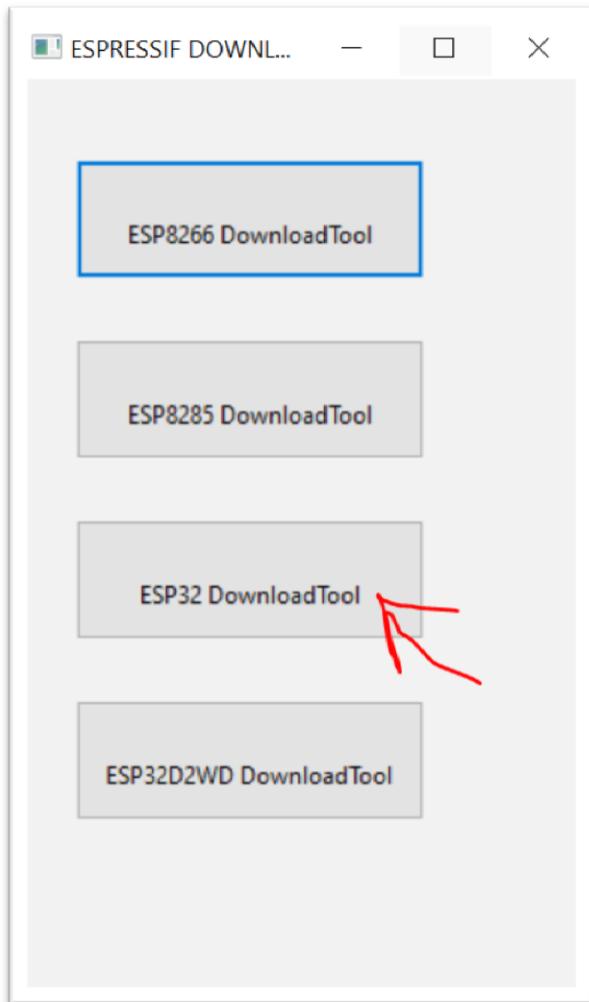
- 2) Download the driver and install on your PC/notebook.
- 3) Restart your PC/laptop.
- 4) Check what COM-ports are already used in your PC/notebook. In Windows 10 do the following:
 - a. Open the Device Manager (right click on Windows symbol on your desktop and then on Device Manager).
 - b. In the 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/products/hardware/esp32/resources>, then under „Tools“ „Flash Download Tools (ESP8266 & ESP32)“)
- 8) Unzip the CAN-Gateway software (files „CAN_Gateway.ino.bin“ (or „CAN_Gateway_DEMO.ino.bin“ for DEMO Version) and „CAN_Gateway.ino.partitions.bin“) in this folder too.
- 9) You will also need files „boot_app0.bin“ (Source: https://github.com/espressif/arduino-esp32/blob/master/tools/partitions/boot_app0.bin) and (bootloader_qio_80m.bin, Source: https://github.com/espressif/arduino-esp32/blob/master/sdk/bin/bootloader_qio_80m.bin). Download and save them in the folder above.
- 10) Now you must have the folder like this:

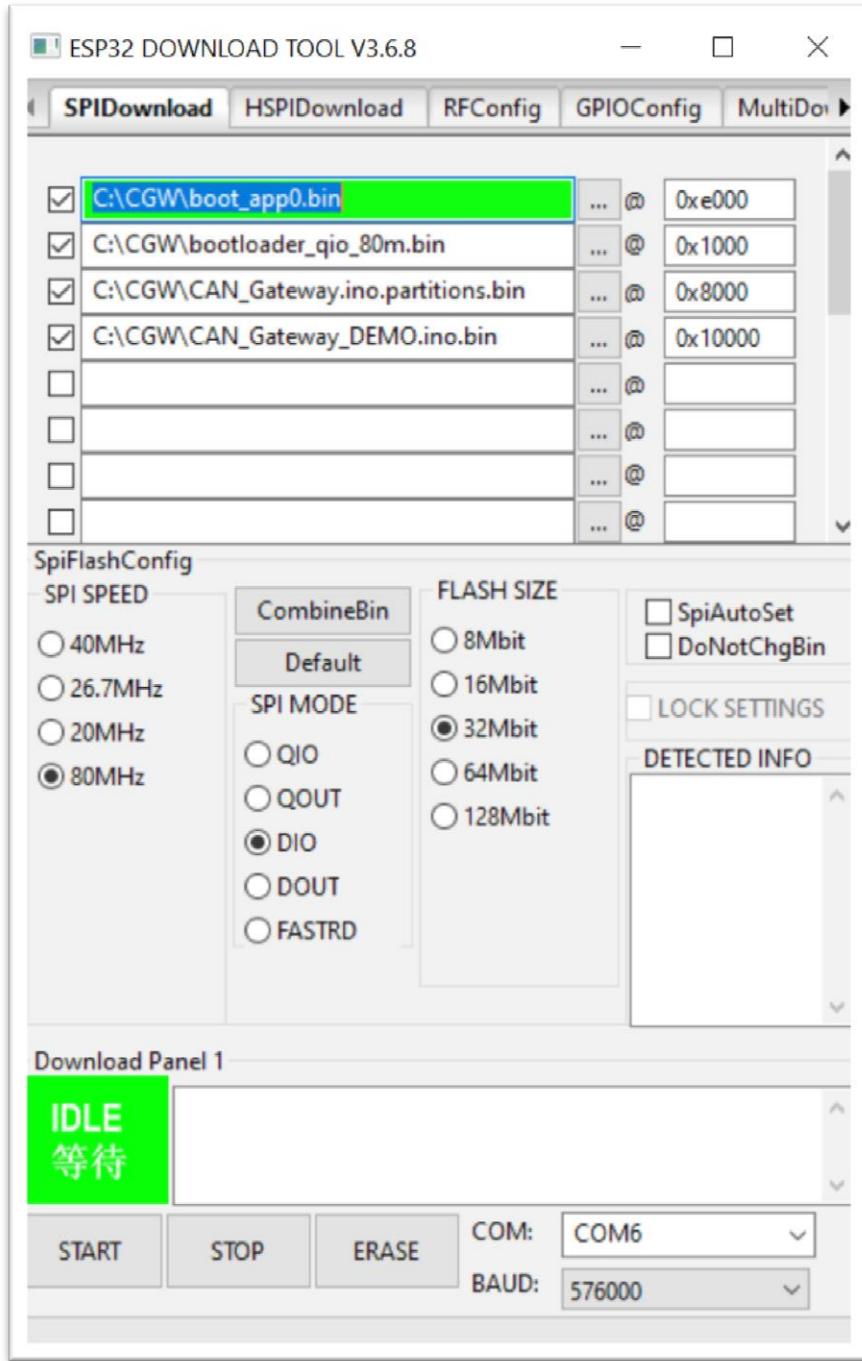


- 11) Execute ESP32 Download Tool (.exe from the folder flash_download_tool_vX.X.X, where X.X.X is the current version)
- 12) The following window will appear, please click on „ESP32 DownloadTool“.



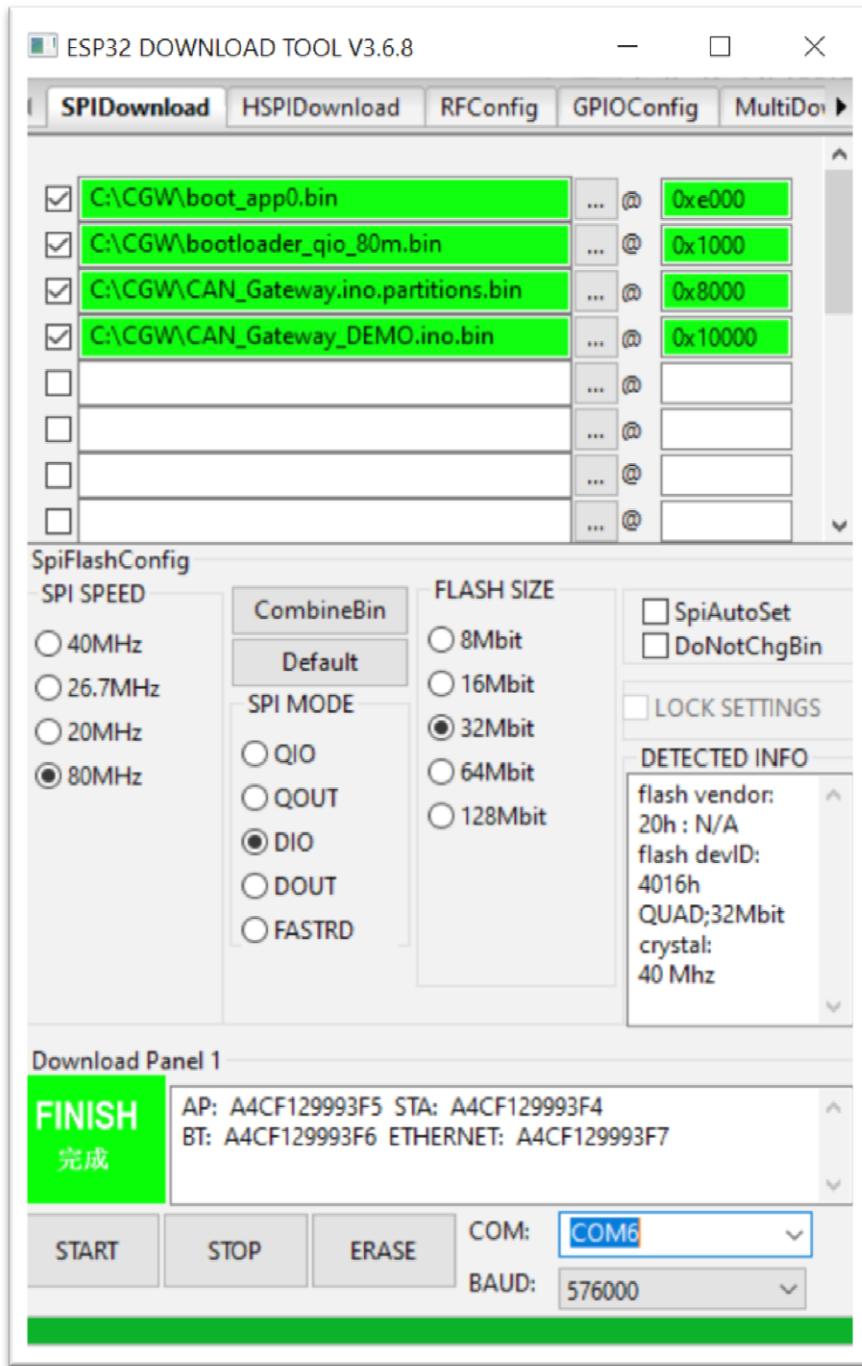
- 13) Please enter the data into the next dialog window as shown below. The only parameter you have to select is the COM-port number. If you get some issues during flashing process, you also can try to select lower baud rate. Please take care that all check boxes are selected

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
```

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 your home WiFi network. For this you might need to enter your home WiFi password.
- 2) Click on the desktop on WiFi Symbol. In the window 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 your 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.

The screenshot shows a web browser window titled "CAN Gateway". The address bar displays "cangateway.de". The page content includes a heading "CAN Gateway", a language selection dropdown set to "English", and a warning message: "Attention! This software is a test software. It is suitable only for testing and is not intended for daily usage. Electronic units that use this software as well as other parts connected to it via busses must be used carefully and only under continuously supervision of a qualified person. [More details here ...](#)". Below the warning, there is a bold note: "Presently the CAN-Gateway is configured as an access point. Therefore not all functions are available. Please configure the CAN-Gateway to connect it to your WiFi. After that you will get access to all functions." A list of links follows: "Settings", "Parameter list - not available", "Generator for parameter configuration", "Device list - not available", "CAN data logger - not available", "Parameter logger - not available", "State information", "Info regarding Modbus TCP usage", "Terms and conditions", and "Software update (over-the-air)".

- 7) Click on „Settings“ and read carefully Terms and Conditions. If you agree please click on „[I agree to these terms and conditions.](#)“:

The screenshot shows a web browser window titled "CAN Gateway" with the URL "cangateway.de/cond". The page content is titled "CAN Gateway: Terms and conditions". It includes a link "[<< Main page](#)". A warning message states: "Attention! This software is a test and learning software. It is suitable only for testing and learning and is not intended for daily usage. Electronic units that use this software as well as other parts connected to it via busses must be used carefully and only under continuously supervision of a qualified person. This person must have a sufficient knowledge and experience with electronic components and be able to recognise failures and dangerous situations. It must also have possibility and be able to switch off all components within of some seconds to prevent hazardous situations." A note follows: "This software and its documentation is provided „as is“ and without any warranty regarding its function and correctness. The author is not responsible for any direct or indirect damages that can be caused by this software, except situations that are regulated by German law (e.g. gross negligence, wilful deceit). The user of this software is fully responsible for its usage." Another note states: "This software is based on an analysis of the communication between devices produced by Hoval company (especially WLAN gateway, touch control module, ventilation system with software version 2.xx.xx) and not based on the official information from the manufacture. Therefore, it must be assumed that this software might not be applicable for all types and versions of devices and might be incomplete and/or faultily." A note about warranties: "The usage of this software in connection with other devices might lead to a loss of warranty of those devices. The user is responsible to check for himself the respective rules by the device supplier." A note about manual reading: "The user of this software shall read carefully and understand the respective manual." A final note at the bottom: "I accept these terms and conditions and want to proceed with configuration and settings." Below this, a note states: "This software is based. among others. on the following software components:"

- 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.

The screenshot shows a web browser window titled "CAN Gateway: settings". The address bar displays the URL "192.168.20.32/settings.html". The main content area is titled "CAN-Gateway: settings" and contains the following sections:

- Buttons:**
 - Download and show present settings from CAN-Gateway
 - Download and show initial settings from CAN-Gateway
 - Save settings (upload to CAN-Gateway) & restart
- Language/Sprache:** Deutsch
- WiFi network name:** [Input field] << Found WiFi networks: [Dropdown]
- WiFi network key:** [Input field]
- Serial output function:** CAN raw data
- Baud rate for serial interface (USB):** [Input field] (possible values: 9600, 19200, 56000, 115200 (default), 256000)
- MQTT server IP (v4):** [Input field]
- MQTT server port:** [Input field]
- MQTT user name:** [Input field]
- MQTT password:** [Input field]
- MQTT topic prefix:** [Input field]
- Identify CAN-Gateway as device type:** [Input field] (recommended: 18 (same as TTE-Gateway))
- Identify CAN-Gateway as device with address:** [Input field]
- Enable control of CAN-Gateway using room control unit:**

- 9) After restart the CAN-Gateway will try to connect to your home WiFi network. Your WiFi router must support WPA/WPA2 authentication as well as DHCP protocol. 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 tries to find this address in google, but this is a local address). CAN-Gateway uses mDNS protocol. The experience shows that Windows 10 identifies new mDNS clients only after restart. If you run Android or other operating systems, they probably do 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.

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”.

The screenshot shows the main page of the CAN Gateway web interface. At the top, there is a header bar with the title "CAN Gateway" and a navigation bar containing icons for back, forward, home, search, and other controls. Below the header, the page title "CAN Gateway" is displayed in large bold letters. Underneath the title, there is a dropdown menu labeled "Language/Sprache: English". A red warning message follows, stating: "Attention! This software is a test software. It is suitable only for testing and is not intended for daily usage. Electronic units that use this software as well as other parts connected to it via busses must be used carefully and only under continuously supervision of a qualified person. [More details here ...](#)". Below the warning, a list of links is provided for navigating through the software's features: "Settings", "Parameter list", "Generator for parameter configuration", "Device list", "CAN data logger", "Parameter logger", "State information", "Info regarding Modbus TCP usage", "Terms and conditions", and "Software update (over-the-air)".

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.

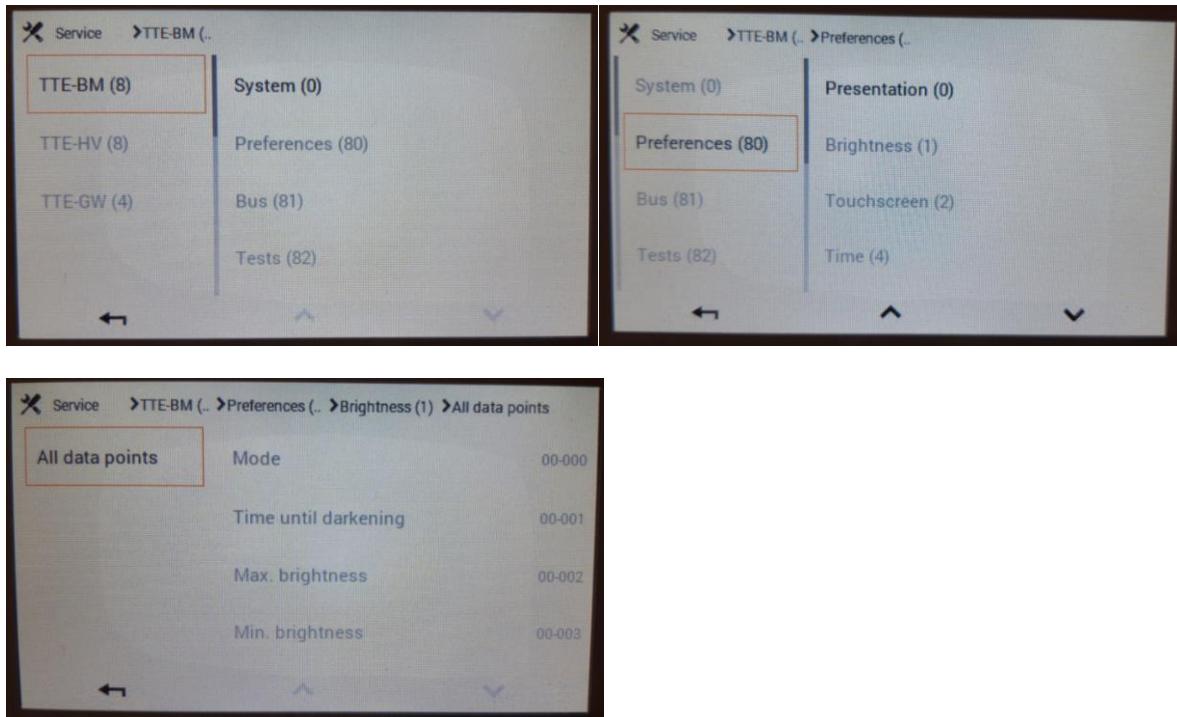
CAN-Gateway has a parameter logger integrated. It shows in the Web interface all the parameters currently passed over the CAN bus. This function, however, is only meaningful if a room control unit or gateway is installed in the system because parameters are only passed over the CAN bus if some device 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.

Example

In the following the parameter is considered that belongs 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), the 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.



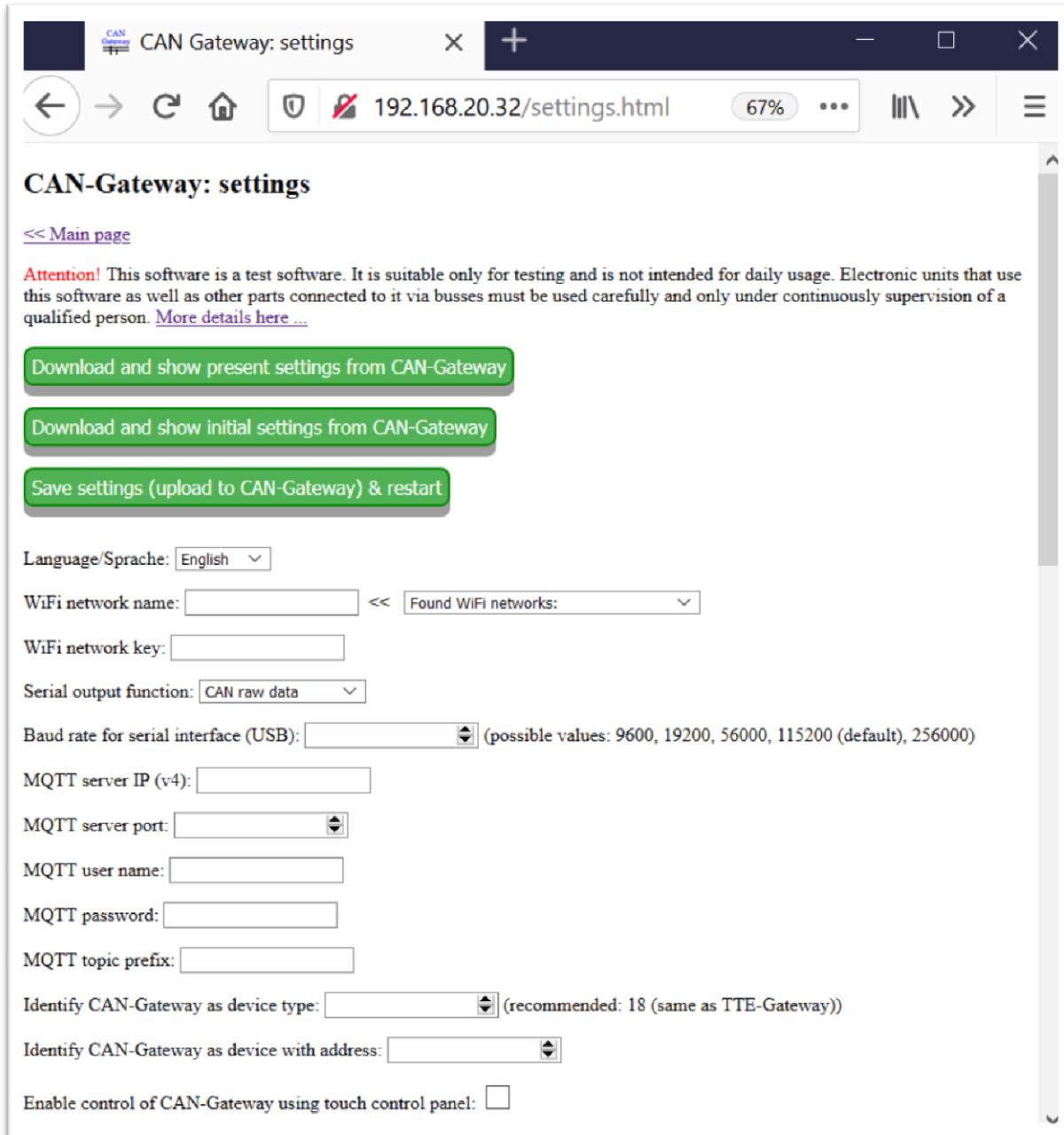
User Levels

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.

Web interface: General settings

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

- Web Interface language (German or English)
- WiFi network name and network key (password)
- Function of the serial interface (USB): there are 3 options to select:
 - o **CAN raw data** – over the serial interface all can raw data are provided: CAN-ID, DLC, data bytes.
 - o **Parameters** - over the serial interface all parameters detected on the CAN bus are provided.
 - o **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”.
- 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“).



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.

Web interface: Parameter settings

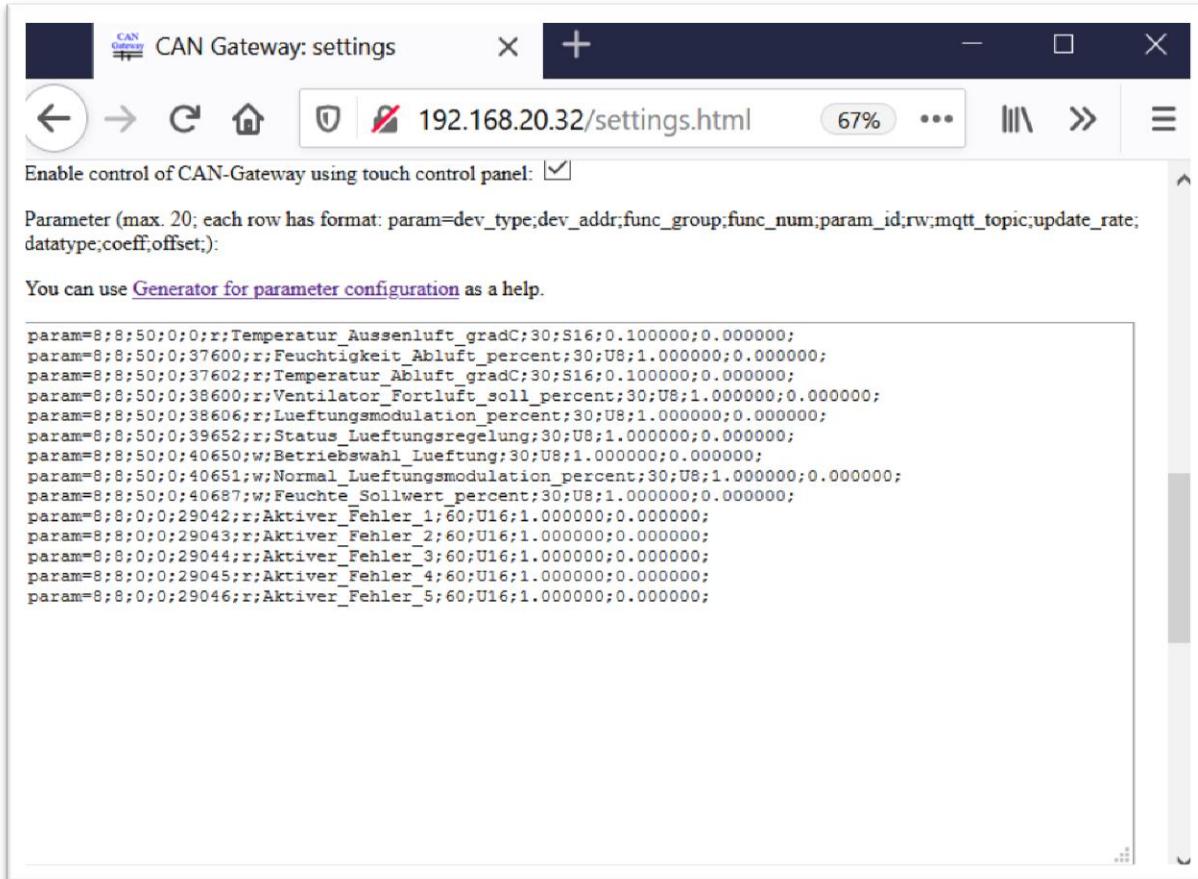
The CAN-Gateway must be configured to read parameters from devices and send them to the home automation system or to set the parameters as requested by the home automation system. At maximum 20 parameters can be configured. As a configuration of each parameter the following options must be set:

- 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 is 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 sent 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 be more often published. CAN-Gateway requests the parameter value only if since latest value transmission over the CAN bus more time has passed than 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 answers 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 feeds 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 unstable 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 is used that must start with “param=”. After “param=” the

configuration data is written, each value separated from others by semicolon. In total up to 20 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).



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.

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.

Examples

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;
```

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 = extract_air_temp_, read each 10 s; data type: S16, coefficient =0.1, offset=0:

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

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;
```

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;
```

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.

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.

Number	Device type	Device address	Function group number	Function number	Datapoint ID	Update rate	R/W	Data type	Coefficient	Offset
1	(8) TTE-HV (HomeVent ventilation)	8	(50) Ventilation	(0) Ventilation	(37600) Humidity extract air [%]	30	r	U8	1	0
2	(16) TTE-BM (Touch control panel)	8	(89) User	(1) Passwords	(4) Pin User Level 4	10	r	STR	1	0
3	not used									
4	not used									
5	not used									
6	not used									
7	not used									
8	not used									
9	not used									
10	not used									
11	not used									
12	not used									
13	not used									
14	not used									
15	not used									
16	not used									
17	not used									
18	not used									
19	not used									
20	not used									

[Generate parameter lines from the table](#)

```
param=8:8:50:0:37600:r:Humidity_extract_air_percent:30:U8:1:0:  
param=16:8:89:1:4:r:Pin User Level 4:10:STR:1:0:
```

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.

The screenshot shows a web browser window titled "CAN Gateway: Parameter list" with the URL "192.168.20.32/paramlist.html". The page contains a table with 15 rows of parameters. Each row includes fields for Number, MQTT topic, Modbus address, Value, Raw value (hex), Age, s, and New value (for request to change). A "Set value" button is located at the end of each row. An "Attention" note at the top states: "this page causes a quite high processor load for CAN-Gateway and must only be used for configuration and debugging, but not during normal operation." A "Reload" button is also visible.

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	<input type="text"/>	<button>Set value</button>
2	Feuchtigkeit_Abluft_percent	10	56	38	14	<input type="text"/>	<button>Set value</button>
3	Temperatur_Abluft_gradC	20	22.5	00E1	14	<input type="text"/>	<button>Set value</button>
4	Ventilator_Fortluft_soll_percent	30	63	3F	14	<input type="text"/>	<button>Set value</button>
5	Lueftungsmodulation_percent	40	33	21	14	<input type="text"/>	<button>Set value</button>
6	Status_Lueftungsregelung	50	3	03	14	<input type="text"/>	<button>Set value</button>
7	Betriebswahl_Lueftung	60	4	04	14	<input type="text"/>	<button>Set value</button>
8	Normal_Lueftungsmodulation_percent	70	33	21	14	<input type="text"/>	<button>Set value</button>
9	Aktiver_Fehler_1	80	65280	FF00FFFFFFFFFFFF0000000000000000	0	<input type="text"/>	<button>Set value</button>
10	Aktiver_Fehler_2	90	65280	FF00FFFFFFFFFFFF0000000000000000	0	<input type="text"/>	<button>Set value</button>
11	Aktiver_Fehler_3	100	65280	FF00FFFFFFFFFFFF0000000000000000	11	<input type="text"/>	<button>Set value</button>
12	Aktiver_Fehler_4	110	65280	FF00FFFFFFFFFFFF0000000000000000	11	<input type="text"/>	<button>Set value</button>
13	Aktiver_Fehler_5	120	65280	FF00FFFFFFFFFFFF0000000000000000	11	<input type="text"/>	<button>Set value</button>
14	NOT USED	n.a.				<input type="text"/>	<button>Set value</button>
15	NOT USED	n.a.				<input type="text"/>	<button>Set value</button>

Web interface: Device list

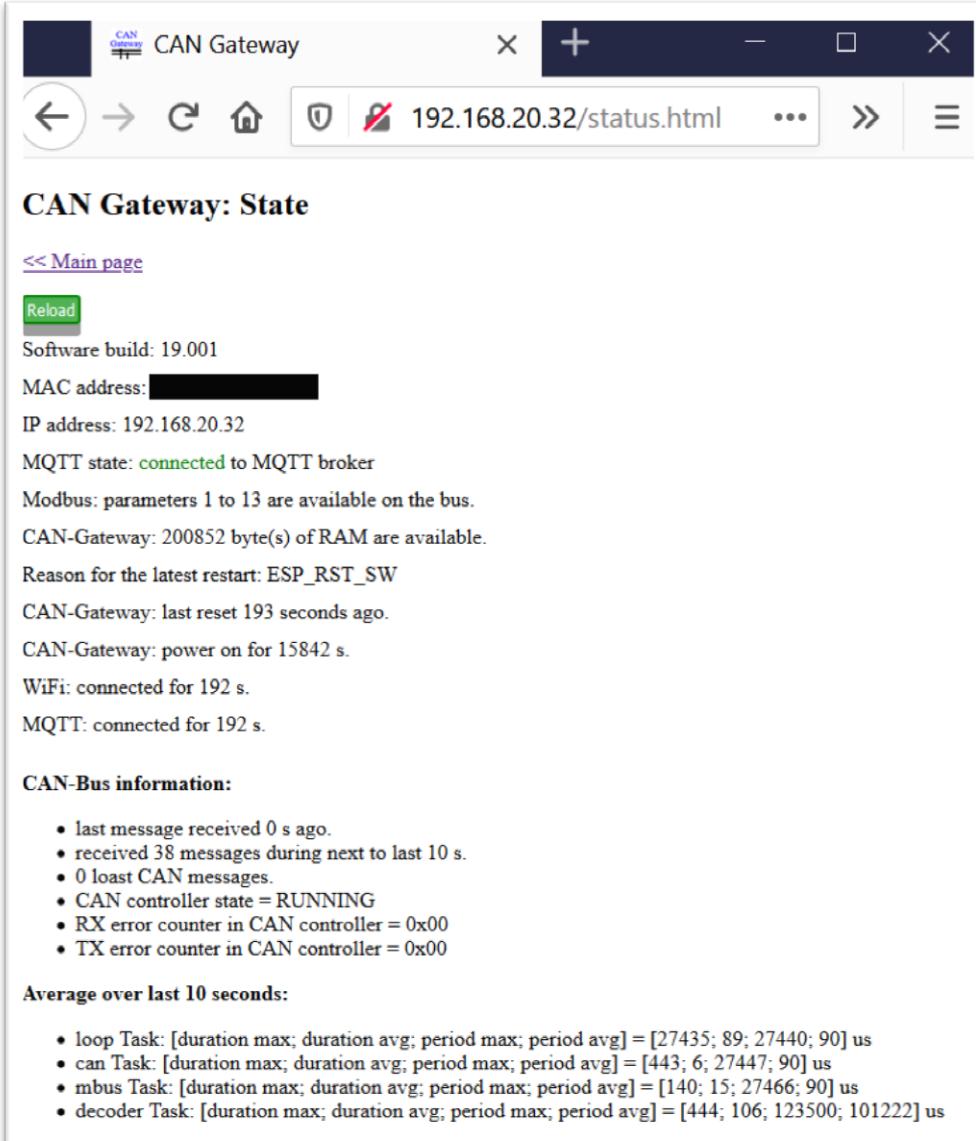
When a device transmit something on the CAN bus, the device is detected by the CAN-Gateway and will be shown in this list.

The screenshot shows a web browser window with the title "CAN Gateway: devices found". The address bar displays the URL "192.168.20.32/founddevices.html". The main content area is titled "CAN Gateway: List of devices found (since system start)". Below this, there are two buttons: "Update" and "Clear the list". A table is displayed with the following columns: Number, Device type, Device type (decimal), and Device address (decimal). The table has 10 rows, numbered 1 to 10. Row 1 contains the entry "1 TTE-HV (HomeVent – ventilation) 8 8". Rows 2 through 10 are empty.

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			

Web interface: Status information

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



The screenshot shows a web browser window titled "CAN Gateway". The address bar displays "192.168.20.32/status.html". The main content area is titled "CAN Gateway: State".

[<< Main page](#)

Reload

Software build: 19.001

MAC address: [REDACTED]

IP address: 192.168.20.32

MQTT state: connected to MQTT broker

Modbus: parameters 1 to 13 are available on the bus.

CAN-Gateway: 200852 byte(s) of RAM are available.

Reason for the latest restart: ESP_RST_SW

CAN-Gateway: last reset 193 seconds ago.

CAN-Gateway: power on for 15842 s.

WiFi: connected for 192 s.

MQTT: connected for 192 s.

CAN-Bus information:

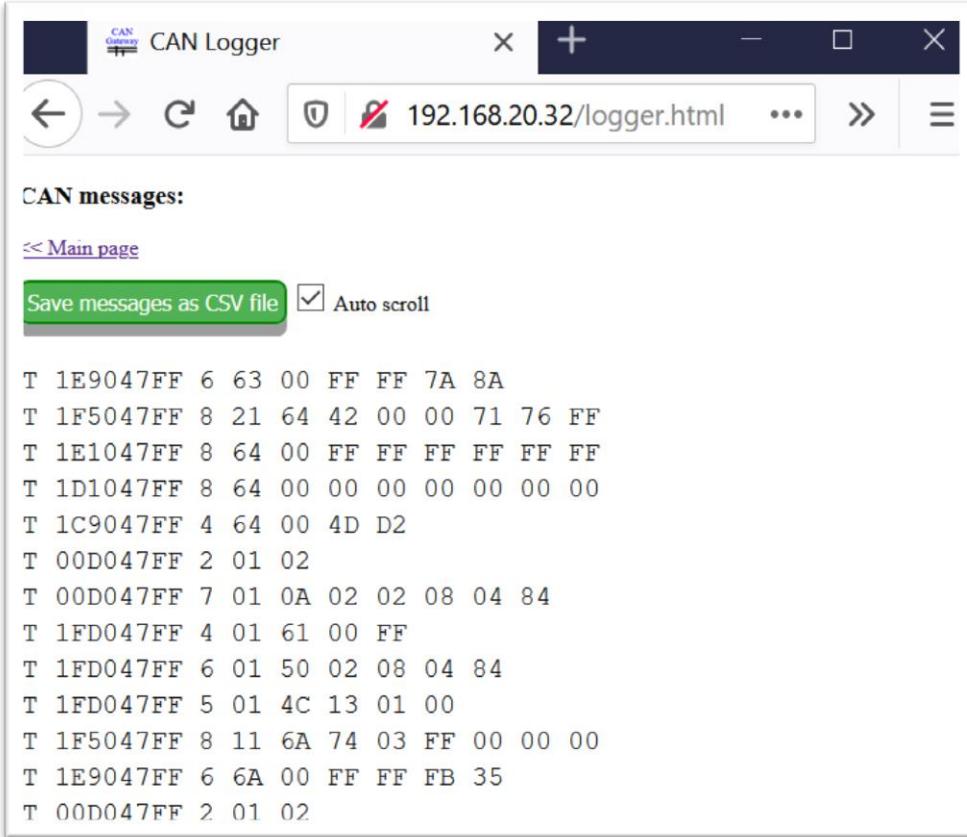
- last message received 0 s ago.
- received 38 messages during next to last 10 s.
- 0 lost CAN messages.
- CAN controller state = RUNNING
- RX error counter in CAN controller = 0x00
- TX error counter in CAN controller = 0x00

Average over last 10 seconds:

- loop Task: [duration max; duration avg; period max; period avg] = [27435; 89; 27440; 90] us
- can Task: [duration max; duration avg; period max; period avg] = [443; 6; 27447; 90] us
- mbus Task: [duration max; duration avg; period max; period avg] = [140; 15; 27466; 90] us
- decoder Task: [duration max; duration avg; period max; period avg] = [444; 106; 123500; 101222] us

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.



The screenshot shows a browser window titled "CAN Logger". The address bar displays "192.168.20.32/logger.html". The main content area is titled "CAN messages:" and contains a list of raw CAN messages. A green button at the top left says "Save messages as CSV file" with a checked checkbox next to it. The messages listed are:

```
T 1E9047FF 6 63 00 FF FF 7A 8A
T 1F5047FF 8 21 64 42 00 00 71 76 FF
T 1E1047FF 8 64 00 FF FF FF FF FF FF
T 1D1047FF 8 64 00 00 00 00 00 00 00
T 1C9047FF 4 64 00 4D D2
T 00D047FF 2 01 02
T 00D047FF 7 01 0A 02 02 08 04 84
T 1FD047FF 4 01 61 00 FF
T 1FD047FF 6 01 50 02 08 04 84
T 1FD047FF 5 01 4C 13 01 00
T 1F5047FF 8 11 6A 74 03 FF 00 00 00
T 1E9047FF 6 6A 00 FF FF FB 35
T 00D047FF 2 01 02
```

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.

Web interface: Parameter logger

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

The screenshot shows a web browser window titled "CAN-Gateway: Parameter logger". The address bar displays "192.168.20.32/para". The main content area is titled "Parameter logger:" and contains a list of parameter entries. A green button labeled "Save list as CSV file" is visible. The list of parameters is as follows:

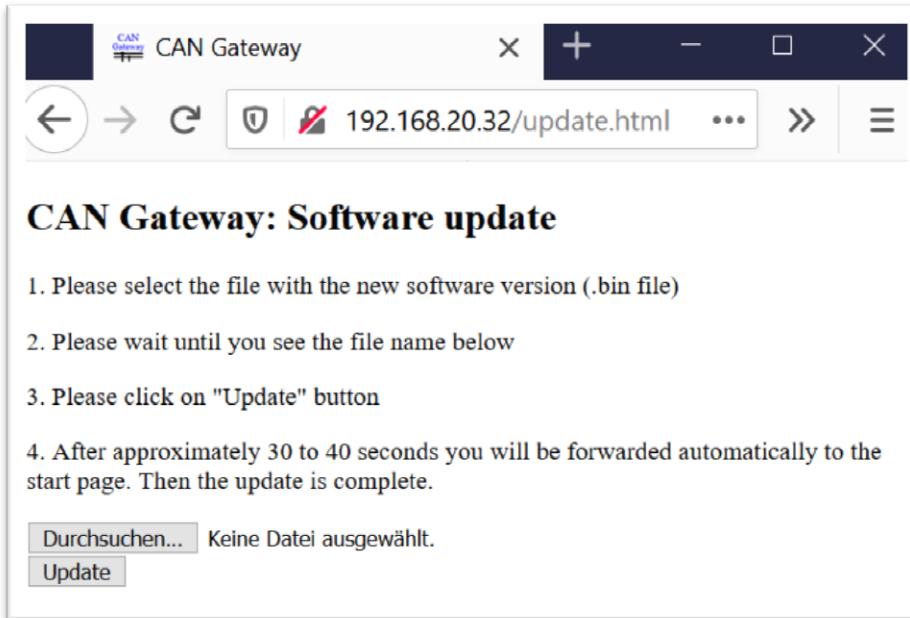
```
(08,08,00,00,7172)= FF 00 FF FF FF FF FF FF 00 00 00 00 00 00 00 00  
(08,08,00,00,7173)= FF 00 FF FF FF FF FF FF 00 00 00 00 00 00 00 00  
(08,08,00,00,7174)= FF 00 FF FF FF FF FF FF 00 00 00 00 00 00 00 00  
(08,08,32,00,92E0)= 37  
(08,08,32,00,96C8)= 3D  
(08,08,00,00,7175)= FF 00 FF FF FF FF FF FF 00 00 00 00 00 00 00 00  
(08,08,00,00,7176)= FF 00 FF FF FF FF FF FF 00 00 00 00 00 00 00 00  
(08,08,32,00,92E0)= 38  
(08,08,32,00,96C8)= 3F  
(08,08,32,00,92E0)= 37  
(08,08,32,00,96C8)= 3D
```

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

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.



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”.

Read/Write parameter values over Modbus TCP

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.

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 described 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 they 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.

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).

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.

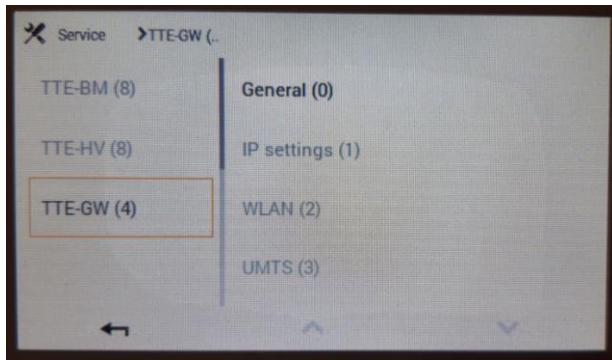
Parameter with data type U32 and S32 are, as described above, transmitted using two registers. In your Modbus master you have to select the correct byte and register order.

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.

Control of the CAN-Gateway using room control unit

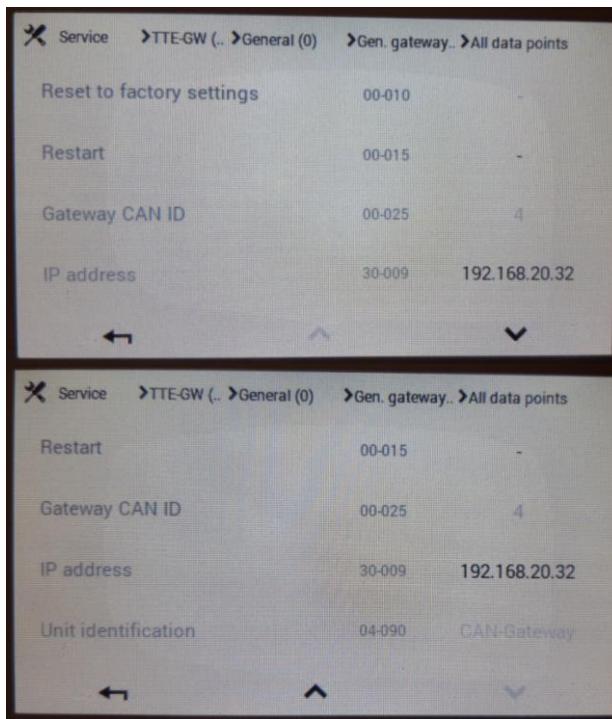
If the CAN-Gateway is connected to a room control unit (TTE-BM) over CAN bus, it can be controlled using this room control unit. The control is limited to some basic functions: CAN-Gateway can be restarted and the settings can be reset to factory values.

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“:

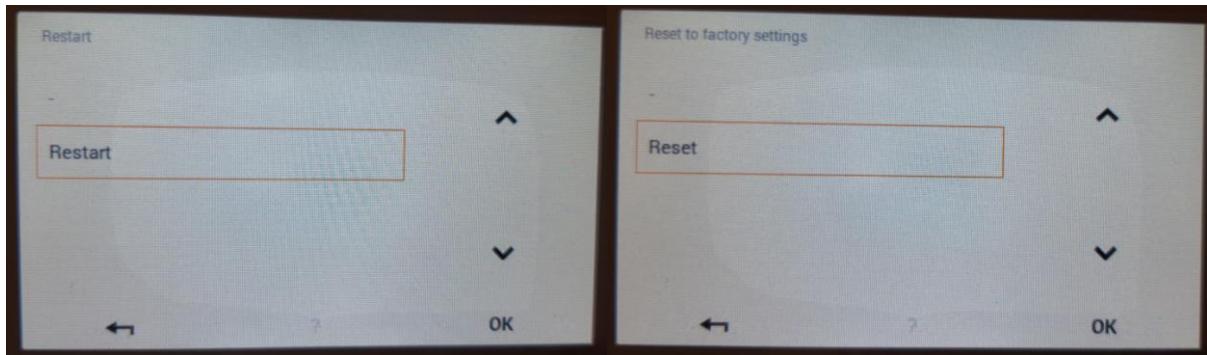


To access 5 options available: do click on **General(0)** → **General Gateway** → **Information (1)**, then scroll down to **All Datapoints**, then click on the list of data on the right-hand side:

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



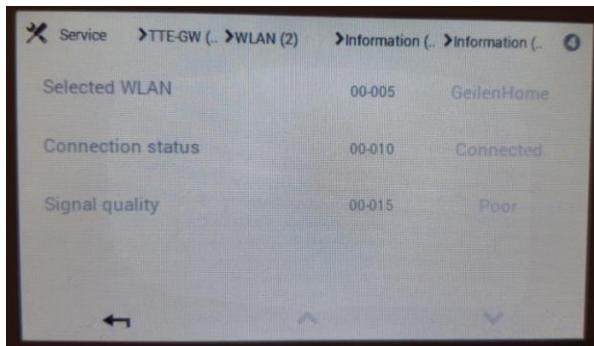
To restart or set the CAN-Gateway to factory settings, you must click to the respective line, symbol „-“, then click to “Restart” or “Reset” and then click on “OK”:



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



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 20)
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 20) 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 20)
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

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.

Help!

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 be able to connect to your WiFi network and, as a consequence, you will not be able to reach its Web interface. In this case you have two 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) 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 CAN-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.

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.

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

It might happen (especially with SW versions below 19.001) if you have already some settings changed and your 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.

Limitations of the demo version

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

- Only one parameter can be configured instead of 20.
- 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.

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“).

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>)
- 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>)
- Async MQTT client for ESP8266 and ESP32 (Copyright (c) 2015 Marvin Roger, Licensed under The MIT License (MIT), <https://github.com/marvinroger/async-mqtt-client>)
- AsyncTCP Library (Copyright 2016 Hristo Gochkov, Licensed under GNU Lesser General Public License v3.0, Source: <https://github.com/me-no-dev/AsyncTCP>)
- 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/ESPAAsyncWebServer>)
- 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>)