EtherCAN/2

Ethernet-CAN-Gateway

Hardware Manual

to Product C.2051.02

esd electronics gmbh
Vahrenwalder Str. 207 • 30165 Hannover • Germany
http://www.esd.eu
Phone: +49 (0) 511 3 72 98-0 • Fax: +49 (0) 511 3 72 98-68

N O T E

The information in this document has been carefully checked and is believed to be entirely reliable. **esd electronics** makes no warranty of any kind with regard to the material in this document, and assumes no responsibility for any errors that may appear in this document. In particular descriptions and technical data specified in this document may not be constituted to be guaranteed product features in any legal sense.

esd electronics reserves the right to make changes without notice to this, or any of its products, to improve reliability, performance or design.

esd electronics gmbh
Vahrenwalder Str. 207
30165 Hannover
Germany

Phone:	+49-511-372 98-0
Fax:	+49-511-372 98-68
E-Mail:	info@esd.eu
Internet:	www.esd.eu

This manual contains important information and instructions on safe and efficient handling of the EtherCAN/2. Carefully read this manual before commencing any work and follow the instructions.
The manual is a product component, please retain it for future use.

Document file:	I:\Texte\Doku\MANUALS\CAN\EtherCAN2\Englisch\EtherCAN-2_Hardware_en_20.odt
Date of print:	2021-10-27

Firmware Version:	from Rev. 4.2.0.16

Document History

The changes in the document listed below affect changes in the hardware as well as changes in the description of the facts, only.

Revision	Chapter	Changes versus previous version	Date
1.2	all	First version of English manual.	2009/11
1.3	2.2	LED description updated	2010/05
	4	Updated description for IP- and DHCP configuration	
	4.2	Updated images	
		Added description for UPnP	
	4.2.3	Added description for ELLSI clients	
1.4	3.2.2.8	Updated notes on reboot	2012/07
	3.2.2.5	Chapter “CAN Bridge” added	
	3.2.3.2	Chapter “Bridge” added	
	6.3	Chapter “Ethernet” added	
	-	Adapted to manual blueprint V1.3	
1.5	3.2.1	Image of website overview updated	2013/01
	3.2.2.5.1	Chapter “Short description/Quick start” added	
	3.2.2.6	Chapter “CAN TCP Client” added	
1.6	3.2.2.6	Chapter “CAN TCP Client” revised	2013/02
1.7	3.2.2.7	Chapter “WebSocket server“ added	2013/04
	3.2.3.4	Screenshot “Connected clients” updated	
1.8	3.2.2.7	Added note about ZIP format of WebSocket server upload	2013/11
1.9	-	Safety Instruction revised, note concerning data safety inserted	2015-12-09
	-	Classification of warning messages inserted	
	2.	Note to chapter “Correct Wiring Electrically Isolated CAN Networks” inserted	
	6.4	New chapter “Conductor Connection/Conductor Cross Sections”	
	10.	New chapter “References”	
	11.	EU Declaration of Conformity updated	
	12.	Order information revised	
2.0	7./8.	Chapter updated	2021-10-27
	11.	EU Declaration of Conformity updated	

Technical details are subject to change without further notice.

Classification of Warning Messages and Safety Instructions

This manual contains noticeable descriptions, warning messages and safety instructions, which you must follow to avoid personal injuries or death and property damage.

This is the safety alert symbol.
It is used to alert you to potential personal injury hazards. Obey all safety messages and instructions that follow this symbol to avoid possible injury or death.

DANGER, WARNING, CAUTION

Depending on the hazard level the signal words DANGER, WARNING or CAUTION are used to highlight safety instructions and warning messages. These messages may also include a warning relating to property damage.

DANGER
Danger statements indicate a hazardous situation which, if not avoided, will result in death or serious injury.

WARNING
Warning statements indicate a hazardous situation that, if not avoided, could result in death or serious injury.

CAUTION
Caution statements indicate a hazardous situation that, if not avoided, could result in minor or moderate injury.

NOTICE

Notice statements are used to notify people on hazards that could result in things other than personal injury, like property damage.

NOTICE
This NOTICE statement contains the general mandatory sign and gives information that must be heeded and complied with for a safe use.

INFORMATION

INFORMATION
Notes to point out something important or useful.

Safety Instructions

- When working with the EtherCAN/2 follow the instructions below and read the manual carefully to protect yourself from injury and the EtherCAN/2 from damage.
- Do not use damaged or defective cables to connect the EtherCAN/2 and follow the CAN wiring hints in chapter: "Correct Wiring of Electrically Isolated CAN Networks".
- In case of damages to the device, which might affect safety, appropriate and immediate measures must be taken, that exclude an endangerment of persons and domestic animals and property.
- Current circuits which are connected to the device have to be sufficiently protected against hazardous voltage (SELV according to EN 60950-1).
- The EtherCAN/2 may only be driven by power supply current circuits, that are contact protected. A power supply, that provides a safety extra-low voltage (SELV) according to EN 60950-1, complies with this conditions.

- Do not open the housing of the EtherCAN/2.
- The EtherCAN/2 has to be securely installed before commissioning.
- Never let liquids get inside the EtherCAN/2. Otherwise, electric shocks or short circuits may result.
- Protect the EtherCAN/2 from dust, moisture and steam.
- Protect the EtherCAN/2 from shocks and vibrations.
- The EtherCAN/2 may become warm during normal use. Always allow adequate ventilation around the EtherCAN/2 and use care when handling.
- Do not operate the EtherCAN/2 adjacent to heat sources and do not expose it to unnecessary thermal radiation. Ensure an ambient temperature as specified in the technical data.

Qualified Personnel
The installation and commissioning of the product may only be carried out by qualified personal, which is authorized to put devices, systems and electric circuits into operation according to the applicable national standards of safety engineering.

Conformity
The EtherCAN/2 is an industrial product and meets the demands of the EU regulations and EMC standards printed in the conformity declaration at the end of this manual.
Warning: In a residential, commercial or light industrial environment the EtherCAN/2 may cause radio interferences in which case the user may be required to take adequate measures.

Data Safety
This device is equipped with an Ethernet or other interface which is suitable to establish a connection to data networks. Depending on the software used on the device, these interfaces may allow attackers to compromise normal function, get illegal access or cause damage.
esd does not take responsibility for any damage caused by the device if operated at any networks. It is the responsibility of the device's user to take care that necessary safety precautions for the device's network interface are in place.

Intended Use
The guarantee given by esd does not cover damages which result from improper use, usage not in accordance with regulations or disregard of safety instructions and warnings.
- The operation of the EtherCAN/2 in hazardous areas, or areas exposed to potentially explosive materials is not permitted.
- The operation of the EtherCAN/2 for medical purposes is prohibited.

Service Note
The EtherCAN/2 does not contain any parts that require maintenance by the user. The EtherCAN/2 does not require any manual configuration of the hardware. Unauthorized intervention in the device voids warranty claims.

Disposal
Devices which have become defective in the long run have to be disposed in an appropriate way or have to be returned to the manufacturer for proper disposal. Please, make a contribution to environmental protection.

Table of Contents

Safety Instructions 5

1 Overview 8
1.1 Description of the EtherCAN/2 8

2 Starting Up 10

3 Software Configuration 11
3.1 Configuration of the IP Address 11
3.1.1 Configuration via DHCP 11
3.1.1.1 Using a Hostname Instead of the IP Address 12
3.1.2 Determining IP Address via UPnP 12
3.1.3 Configuration via esdcp 14
3.2 Web Based Configuration 16
3.2.1 Overview 16
3.2.2 Configuration 17
3.2.2.1 Security 17
3.2.2.2 Network Settings 18
3.2.2.3 Remote Logging 22
3.2.2.4 Firmware Update 23
3.2.2.5 CAN Bridge 25
3.2.2.5.1 Short description/Quick start 26
3.2.2.5.2 Important notes 28
3.2.2.5.3 Miscellaneous 29
3.2.2.5.4 Configuration file 29
3.2.2.5.5 Edit Configuration Online 30
3.2.2.5.6 Configuration file 33
3.2.2.6 CAN TCP Client 35
3.2.2.6.1 Protocol 37
3.2.2.6.2 Version information struct 39
3.2.2.6.3 CAN Frame struct 39
3.2.2.7 WebSocket server 40
3.2.2.8 Reboot 42
3.2.3 Status 43
3.2.3.1 CAN Statistics 43
3.2.3.2 Bridge 44
3.2.3.3 Ethernet 45
3.2.3.4 Connected clients 46
3.2.3.5 Alarms and Events 47

4 LEDs 48
4.1 Position of the LEDs 48
4.2 LED Indication 48

5 Technical Data 50
5.1 General Technical Data 50
5.2 Microprocessor and Memory 50
5.3 CAN Interface 51
5.4 Ethernet Interface 51
5.5 DIAG, Serial Interface via USB Interface 51
5.6 Software 52

6 Interfaces and Connector Assignments 53
6.1 CAN 53
6.2 24 V-Power Supply Voltage 54
6.3 Ethernet 100BASE-TX (IEEE 802.3) 55

6.4 Conductor Connection/Conductor Cross Sections....56

7 Correct Wiring Electrically Isolated CAN Networks....57
7.1 Standards concerning CAN Wiring....57
7.2 Light Industrial Environment (Single Twisted Pair Cable)....58
7.2.1 General Rules....58
7.2.2 Cabling....59
7.2.3 Branching....59
7.2.4 Termination....59
7.3 Heavy Industrial Environment (Double Twisted Pair Cable)....60
7.3.1 General Rules....60
7.3.2 Device Cabling....61
7.3.3 Branching....61
7.3.4 Termination....61
7.4 Electrical Grounding....62
7.5 Bus Length....62
7.6 Examples for CAN Cables....63
7.6.1 Cable for light industrial Environment Applications (Two-Wire)....63
7.6.2 Cable for heavy industrial Environment Applications (Four-Wire)....63

8 CAN Troubleshooting Guide....64
8.1 Termination....64
8.2 Electrical Grounding....65
8.3 Short Circuit in CAN Wiring....65
8.4 CAN_H/CAN_L-Voltage....65
8.5 CAN Transceiver Resistance Test....66
8.6 Support by esd....66

9 Appendix InRailBus (Option)....67
9.1 Order Information InRailBus Accessories....67
9.2 Connector Assignment 24V and CAN via InRailBus (Option)....68
9.3 Using InRailBus (Option)....69
9.3.1 Installation of the Module Using InRailBus Connector....69
9.3.2 Connecting Power Supply and CAN Signals to CBX-InRailBus....70
9.3.3 Connection of the Power Supply Voltage....71
9.3.4 Connection of CAN....71
9.4 Remove the CAN-CBX Module from InRailBus....72

10 References....73

11 Declaration of Conformity....74

12 Order Information....75

1 Overview

1.1 Description of the EtherCAN/2

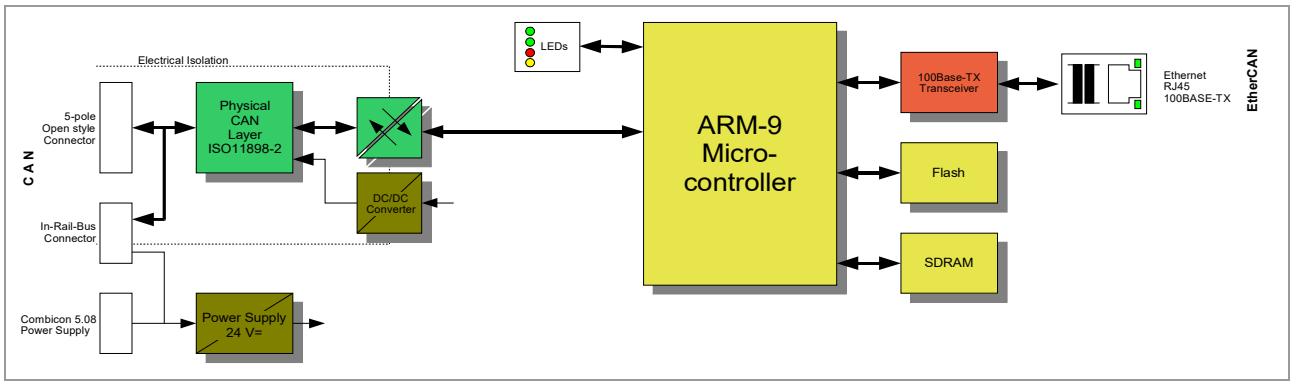


Figure 1: Block circuit diagram

The EtherCAN/2 is an Ethernet-CAN-Gateway equipped with an ARM9 processor, which controls the data transfer between CAN and Ethernet.
The Ethernet interface is suitable for 10 Mbit/s and 100 Mbit/s networks and can be connected via an RJ45-socket.
The CAN interface can be connected via a 5-pin Combicon connector.
The interface is in accordance with ISO 11898-2, is electrically isolated and can be used for transmission rates from 20 kbit/s up to 1 Mbit/s.
The connectors for the Ethernet-, CAN- and serial interface and the status LEDs are located in the front panel of the top hat rail module and are easily accessible.
The 4-pin Combicon connector for the power supply is located in the case top side, easily accessible.

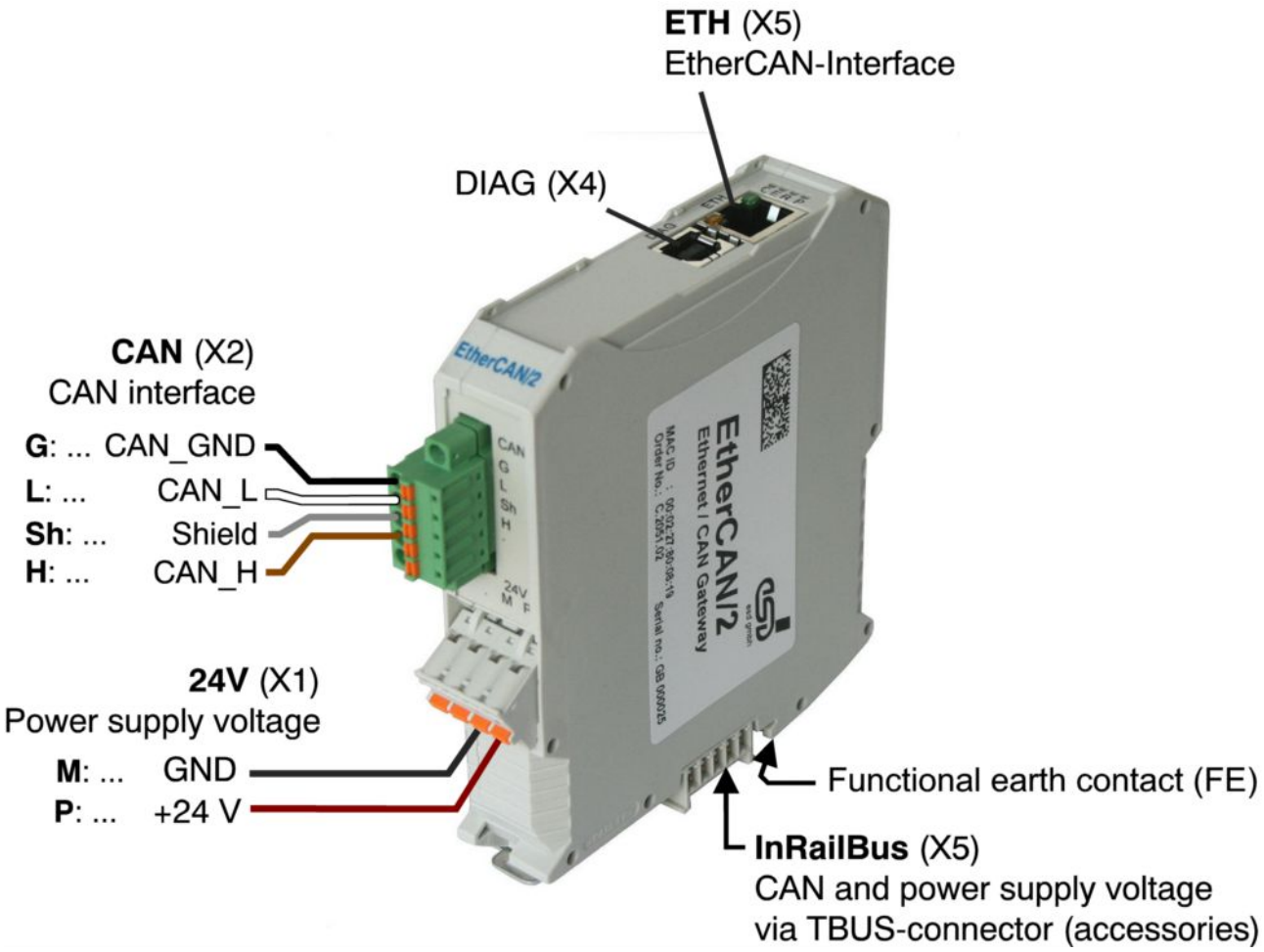


Figure 2: Connections in operable state

NOTICE
Read chapter “Starting Up” on page 10, before you start with the installation of the hardware!
Please refer to page 53 for signal assignment of the connectors.

2 Starting Up

To start up the EtherCAN/2 follow the instructions below.

Step	Action	see page
	**Read the safety instructions at the beginning of this document carefully before you start with the hardware installation!**	5
	**DANGER** Hazardous Voltage - Risk of electric shock. All current circuits which are connected to the device have to be sufficiently protected against hazardous voltage (SELV according to EN 60950-1) before you start with the installation.	
1.	Mount and wire the EtherCAN/2 module (power supply voltage, CAN, Ethernet).	9
2.	Please remember that the CAN bus has to be terminated at both ends. **esd** offers T-connectors and terminators. Additionally, the CAN-GND-signal has to be grounded at *exactly one* point in the CAN network. For details please read chapter “Correct Wiring Electrically Isolated CAN Networks” A CAN device with a CAN interface which is not electrically isolated corresponds to the grounding of the CAN-GND.	57
3.	Switch on the 24 V-power supply voltage of the EtherCAN/2.	-
4.	Assignment of a valid IP address for EtherCAN/2. 4a) If the current IP address is known, proceed to step 6. with the assignment of a fixed IP address and the further configuration. 4b) If the current IP address is not known, use esdcp to detect it and to assign a fixed IP address. For that purpose at first the host-driver software has to be installed as described under step 5.	11
5.	Insert the CAN-driver CD in the appropriate drive of your PC. Install the EtherCAN/2 host-driver software from the CAN-driver CD on your PC, as described in the manual 'NTCAN-API Part 2 Installation Guide' [1] for the EtherCAN/2 module. The installation will be guided by an installation program (*Install Shield* Wizard). The window of the installation program depends on the Windows operating system installed on your PC.	-
6.	Configure the EtherCAN/2 via web browser.	16
	Note: Please note that changes of the configuration generally will not be effective until a reboot.	42
7.	Install the host-driver software now as described under step 5., if not already done. Configure the host-driver on the Windows PC as described in the manual 'NTCAN-API Part 2 Installation Guide' for the EtherCAN/2 module.	-

3 Software Configuration

The following chapter describes the configuration of the EtherCAN/2 module in two steps:

1. Assignment of a valid IP address. (If not already known/reachable)
2. Configuration of the other parameters by means of a web browser. (see page 19)

The RJ45-socket has to be connected with the configuring host PC, like in later operation, via a Twisted-Pair cable with a switch or hub or via a Cross Twisted-Pair cable. The yellow (Link) LED has to turn on to indicate a correct connection.

3.1 Configuration of the IP Address

First a valid IP address has to be assigned to the EtherCAN/2. An IP address is an unique address for a device, that communicates in a TCP/IP network.
For the configuration it is therefore necessary to configure an unique IP address, which is not already assigned to another device in the network.

First configuration of the IP address:
In delivery status after switching-on the EtherCAN/2 attempts to get assigned an IP address by a DHCP-server (maximum approx. 2 min.).

In case that the DHCP-server does not assign an IP address to the EtherCAN/2, the EtherCAN/2 generates an Auto-IP address by itself in the range 169.254.X.X .
To change/determine the assigned IP address it is recommended to use the tool esdcp (see page 14).
To enable esdcp to find the device, the PC on which esdcp runs has to be at least in the same subnet.
If esdcp finds the device, IP address and netmask have to be set like for the later use.
Then (after a Reboot of the EtherCAN/2) the EtherCAN/2 can be configured by means of a web browser (see page 16 et seq.).

3.1.1 Configuration via DHCP

For a configuration via DHCP the DHCP-server has to be in the **same subnet** as the EtherCAN/2. Before usage the DHCP-server possibly has to be configured especially. Please contact your system administrator for this purpose.
The server assigns to the module a valid IP address, a net mask, a gateway address and the IP address of a name server. After a successful assignment the EtherCAN/2 operates with these values without reboot.

The IP address **<IP Address>** which is assigned to the device has to be detected by means of the logging mechanisms of the DHCP-server.
Further configuration of the network parameters can now be done by means of any web browser, that is in the same subnet, with the URL **http://<IP Address>**, as described in chapter: “3.2 Web based Configuration”.

NOTICE
Without further configuration a DHCP-server might possibly assign another IP address to a device at every reboot and this only for a specific period. For the driver software on the host-PC it is necessary, that this IP address is invariable and unchanged for the complete period of operation. If at every reboot the same IP address shall be assigned to the EtherCAN/2 module via DHCP, the system administrator has to ensure this.

3.1.1.1 Using a Hostname Instead of the IP Address

The DHCP Client of the EtherCAN/2 automatically transmits the hostname configured under *Network Settings* to the DHCP-Server.

This is subject to some restrictions:

- In addition to the DHCP-server a DNS-server has to be available and both servers must co-operate accordingly.
- To enable this is task of your system administrator, if necessary - further support by esd concerning this matter is not possible.
- The choice of the hostname might be restricted depending on the system and the DHCP-/DNS-server used: E.g. usually underscores are not allowed. In a test with a DNS-server a limitation of the length to 15 characters has been found. In another test the hostname has only been solved correctly, if it contained the domain-suffix. (Independent of the domain name configured!)
- Each time the EtherCAN/2 receives a new IP address from the DHCP-server, all network connections of the EtherCAN/2 are terminated.

Thus esd recommends to use a **fixed** IP address instead!

Standard Hostname

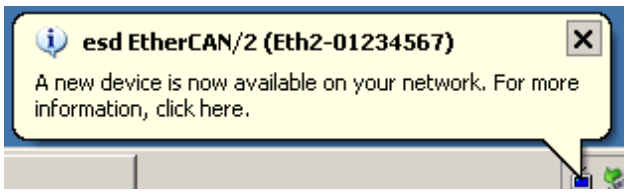
If the hostname is not configured in the web-based configuration, a standard name is generated containing the serial number.
For example for the serial number “AA001234” the hostname is: **Eth2-001234**.
(The same applies for entering “default” for the hostname in the esdcp software.)

3.1.2 Determining IP Address via UPnP

In the state of delivery the EtherCAN/2 furthermore operates as an UPnP-device. The IP address of the EtherCAN/2 can not be changed by this, but generally you can easily get to the web based configuration.
Precondition for this is that UPnP is supported by the operating system used. Furthermore the IP address used by the EtherCAN/2 must be accessible by this computer. (Firewall-settings concerning UPnP[1] might possibly have to be made.)

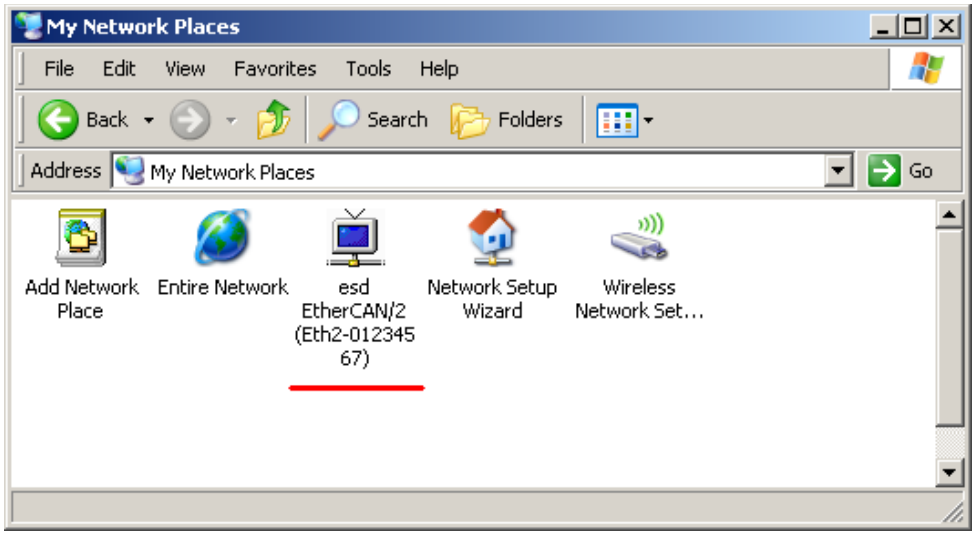
Example: Windows XP

Provided that UPnP is enabled, a note is shown when the EtherCAN/2 is identified:



1 Standards for UPnP are already defined for the Windows-Firewall (Proceeding after Table 1 automatically enables these). Otherwise: Incoming connections on UDP Port 1900 and TCP Port 2869 have to be permitted.

Furthermore the symbol of the EtherCAN/2 is shown in the window *My Network Places*:



A double click on the EtherCAN/2 icon automatically opens the configuration page in the standard web-browser.

Enable UPnP under Windows XP

1. Click **Start** and then click **Control Panel** 2. In Control Panel, double-click **Add or Remove Programs**. 3. Click **Add/Remove Windows Components**. 4. In the **Components** list, click **Networking Services**, and then click **Details**. 5. Select the **Universal Plug and Play** check box, and then click **OK**. 6. Click **Next**, and then click **Finish**. Furthermore the service “SSDP-Discovery Service” has to be enabled. (See **Control Panel** → **Administration Tools** → **Services**) *Reference: Microsoft (http://support.microsoft.com/kb/821371/)*

Table 1: Enable UPnP under Windows XP

Example: Windows Vista / Windows 7

The corresponding settings are generally enabled by default here, and the symbol of the EtherCAN/2 is shown in the window *My Network Places* (without a note as in Windows XP).
A double click to the icon of the device automatically opens the configuration page in the standard web-browser.

Additionally there is the context menu entry: “Open CAN Control Panel”, which starts the configuration software. (Also refer to the software manual: 'NTCAN-API Part 2, Installation Guide' [1])

3.1.3 Configuration via esdcp

esdcp is a tool to find and configure esd devices with Ethernet interface in a LAN. For this a stateless protocol – using UDP - is used.

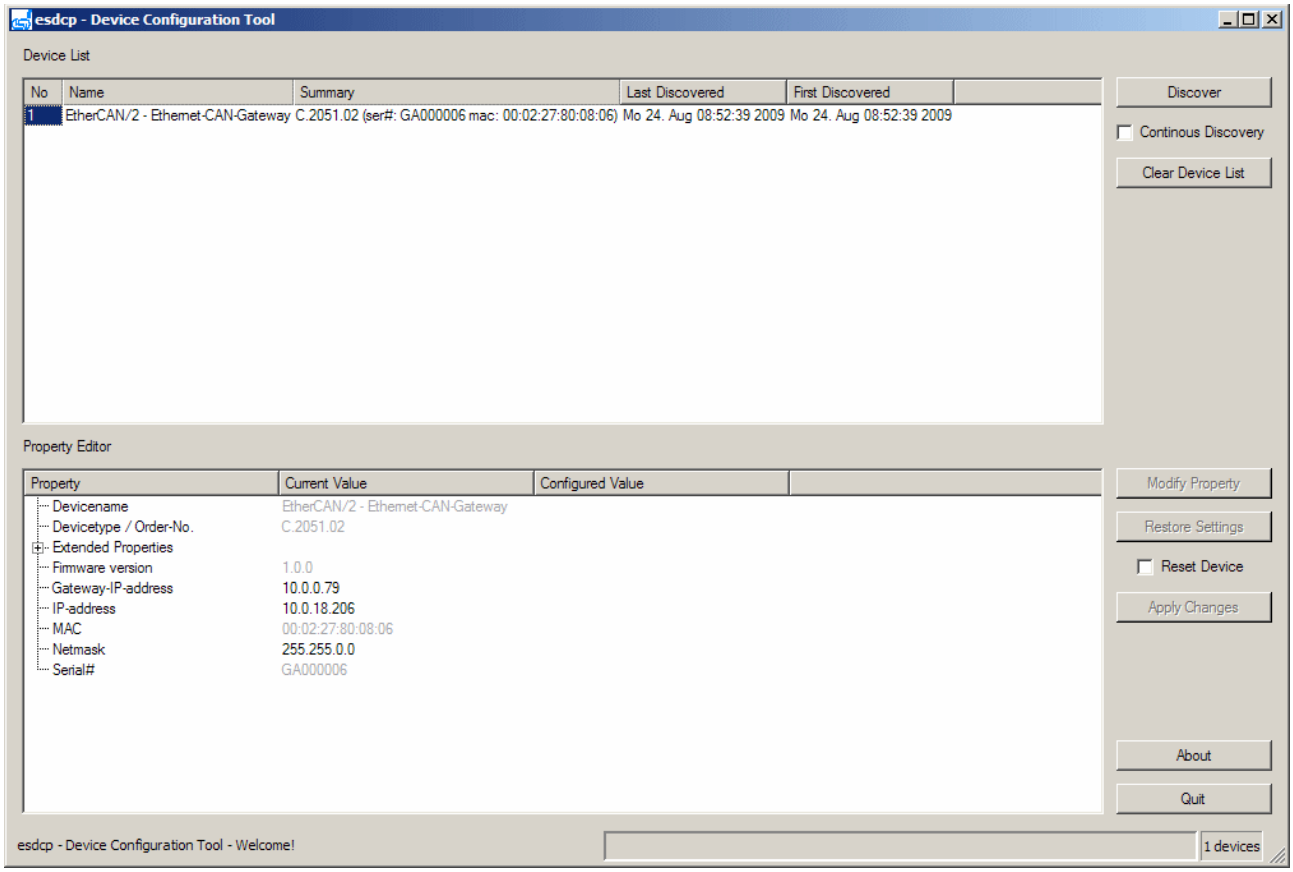


Figure 3: esdcp - Device Configuration Tool

To be able to detect devices with Ethernet interface the PC has to be in the same subnet.

Click the button Discover to search for esd devices. The detected devices will be shown in the *Device List*.
Enable the continuous search for esd devices by clicking the Check-Box ❑ *Continuous Discovery*.

Devices which are no longer available will not automatically be deleted from the list.
To delete the entries of the *Device List* click the button Clear Device List.

If no DHCP could have been been found and the EtherCAN/2 thus has got an IP address in the range of 169.254.x.x, the PC has to be configured accordingly (only once, for the configuration of the EtherCAN/2).

If the tool esdcp has detected the EtherCAN/2, at least IP address and subnet mask can/must be set there. For this double click in the window *Property Editor* the row *IP-address* to change the IP address or the row *Netmask* to change the netmask.
Now you can open the input field of the selected row also by clicking the button Modify Property.

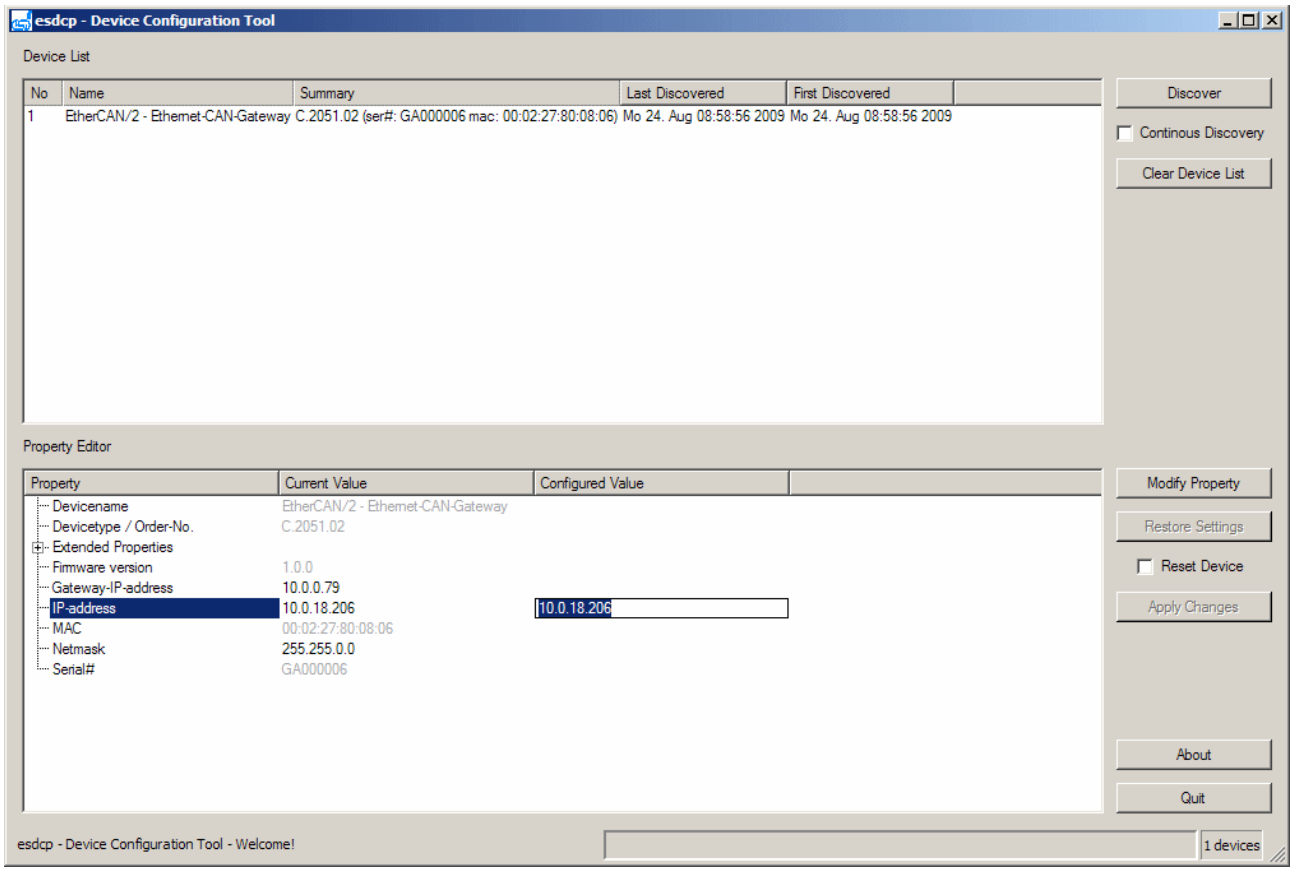


Figure 4: Modify the IP address via esdcp

Enter the IP address in the input field in the column *Configured Value*.
Enter the *Netmask* in the same manner.
Activate the check-box ❑ *Reset Device* for a reboot of the EtherCAN/2 after acknowledgement of the entry, because the changes are not transferred until the device is rebooted.
Confirm the entry of the IP address by clicking the button Apply Changes.

After a request of the password the value entered under *Configured Value* will then be accepted as current value and shown under *Current Value*.

The button Restore Settings deletes the entered values in the field *Configured Value*. The current values remain unchanged.

The esdcp default password is: esd

The password can be changed via page “Network settings” of the web-browser.

3.2 Web Based Configuration

The EtherCAN/2 offers an integrated HTTP-Server, which allows further configuration with a web-browser. It follows the description of the pages as they are display in the browser, see Figure 5 left.

3.2.1 Overview

A menu is shown on the left side of this program window.
In the menu item *Overview* the module specific details of the EtherCAN/2 are shown. Under *Gateway details* you find firmware revision, hardware revision, order number and serial number. Under *Gateway status* you find details about the CAN interface of the EtherCAN/2.

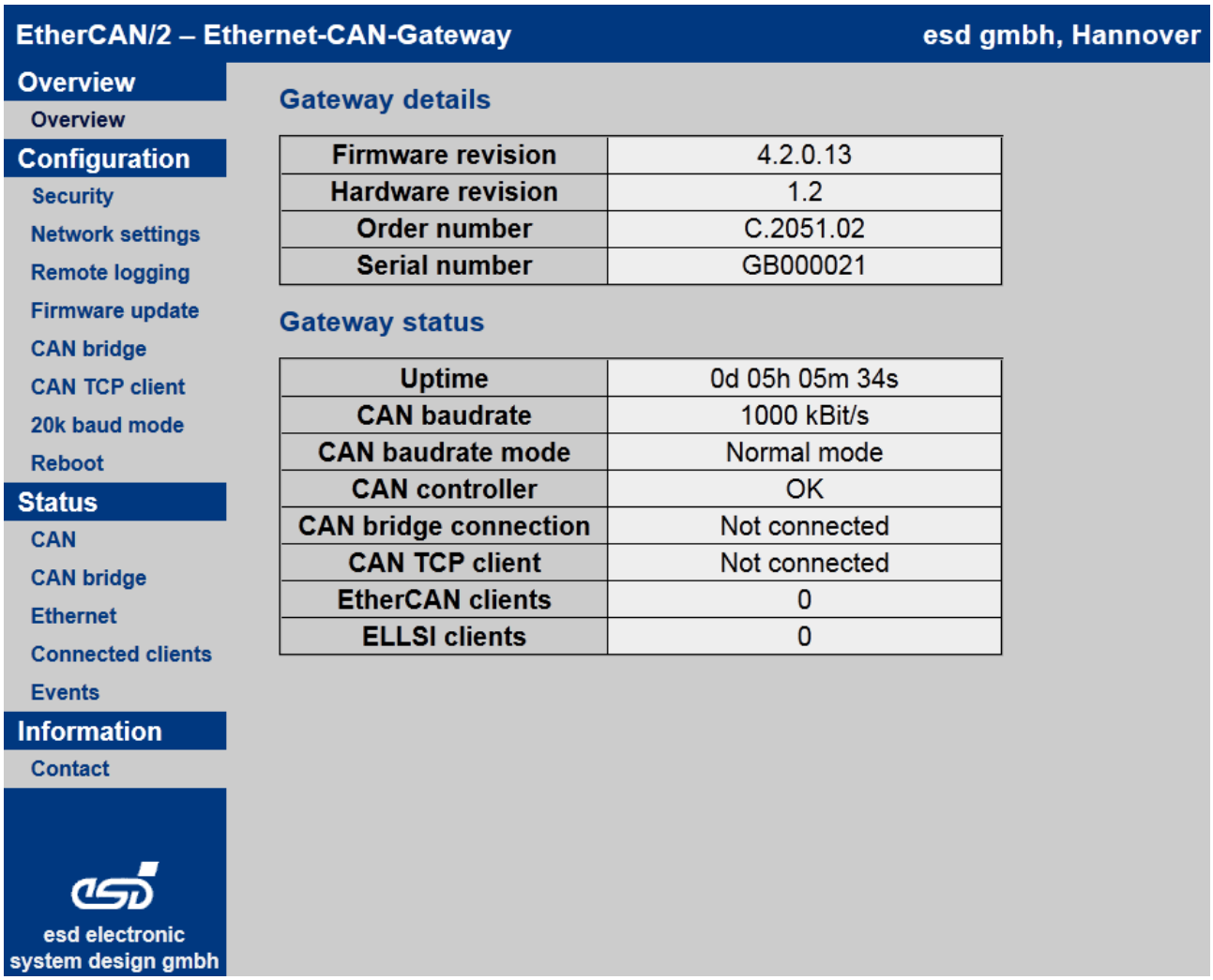


Figure 5: Overview

3.2.2 Configuration

All settings which can be made under menu item *Configuration*, are protected by a combination of user name and password. The default setting at delivery is:

User name:	esd
Password:	esd

3.2.2.1 Security

In the *Security Setup* page you can change the *Webserver Username* and the *Webserver Password* for access protection. These values will be asked for the authentication in the firmware update as described in the following chapter.

User name and password can be adapted. Usage of special characters and space characters is not allowed. Pay attention to case sensitivity.

Clicking the button Submit saves the changed data in a non-volatile memory of the EtherCAN/2 module. After a reboot the data will become active.

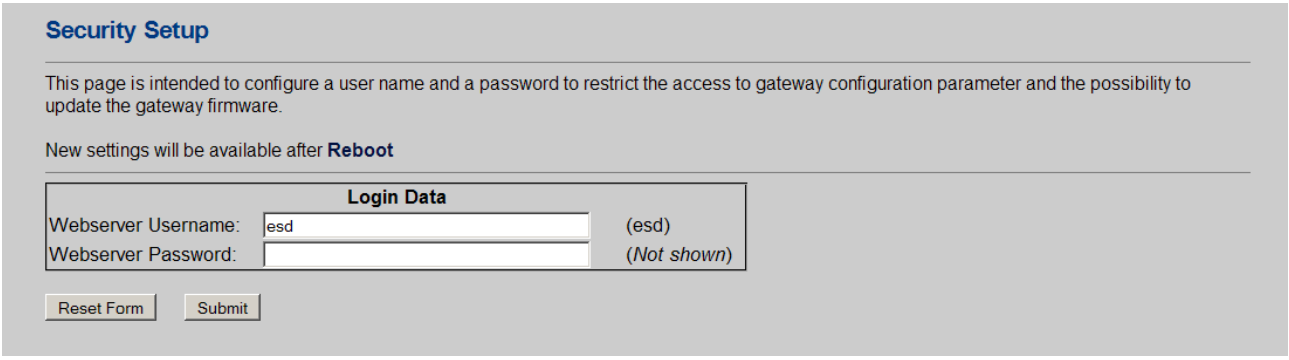


Figure 6: Enter user name and password

3.2.2.2 Network Settings

The menu item *Network Settings* contains an overview of the configured values for:

- **TCP/IP**
- **SMTP**
- **Time Server**
- **SNMP**
- **esdcp**

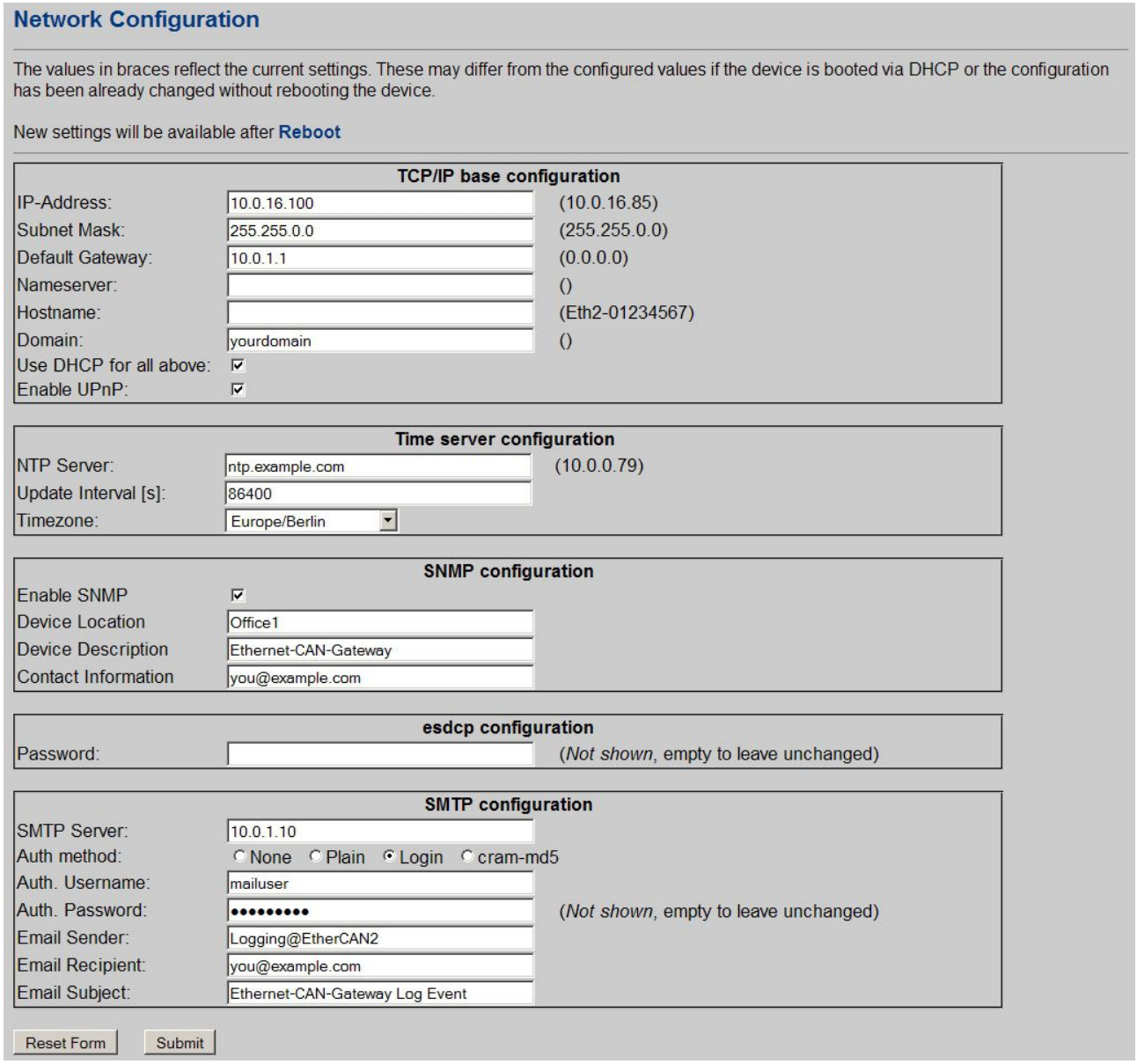


Figure 7: Network configuration

- **TCP/IP base configuration**

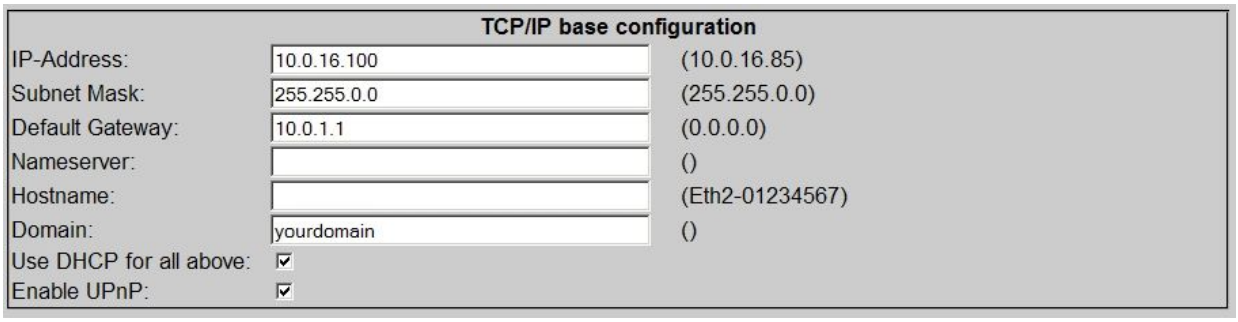


Figure 8: TCP/IP base configuration

Here the base TCP/IP network parameters can be configured. The current settings of the parameters are shown in brackets.

At the first time starting up at least the subnet mask and the IP address have to be adapted to the conditions of the network.

For the communication with other nets, e.g. to access Internet or time server, a *Default Gateway* has to be assigned.

The addresses of a nameserver and a NTP-time server can be optionally configured under *Time server configuration.* They will be evaluated by the *Remote Logging* support (see page 22).

Enable the check box ❑ *Use DHCP* to assign IP address, gateway and subnet mask via DHCP. This may cause problems as described in chapter 'Configuration via DHCP' (see page 11).
To use the parameters above (*IP address*, *Subnet Mask*, *Default Gateway* and *Nameserver*) the check box must be disabled.

When the checkbox ❑ *Enable UPnP* is enabled, the EtherCAN/2 also acts as an UPnP device. This is usually only used for an easier discovering when its IP address is unknown, see chapter: "3.1.2 Determining IP Address via UPnP".

A domain name can be entered in the input box *Domain*.

TCP/IP Default Parameters

At the first time starting up of the EtherCAN/2 the check box ❑ *Use DHCP* is enabled. Thus the IP address is per default assigned via the DHCP-Server.

- **SMTP configuration**

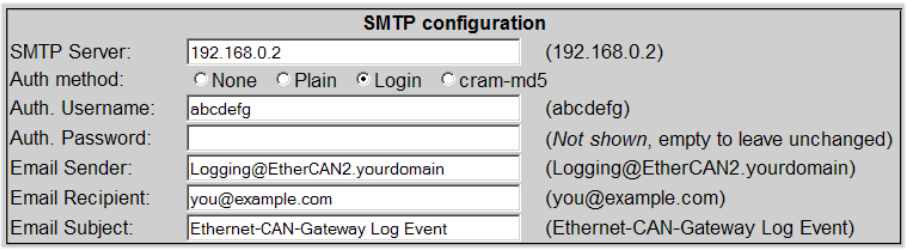


Figure 9: SMTP configuration

The data such as SMTP server IP, user name, password etc. can be configured here to forward alarms and events as email to a SMTP server.
Otherwise the configuration of these parameters is not necessary.

If the SMTP server requires a login, the method of the login can be selected under *Auth. method*.
Login is selected per default.

- **Time Server**

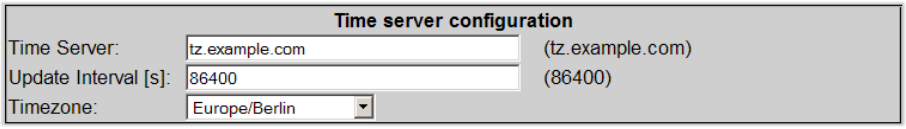


Figure 10: Time server configuration

The NTP-server can be configured in this field.
For correct date and time indication a time server must be configured.

To be able to use the Remote Logging-Support these parameters have to be configured.
Without an entry in this field, the calculation of times will restart from 1970-01-01 at 00:00 a.m with every reboot (power up) of the EtherCAN/2 module.

- **SNMP**

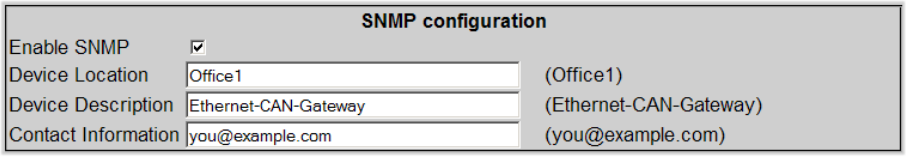


Figure 11: SNMP configuration

The SNMP configuration is disabled per default.
To use SNMP click the Check-Box ❑ *Enable SNMP*.

SNMP (Simple Network Management Protocol) is a protocol to control and monitor various network devices, e.g. router, printer etc.
The EtherCAN/2 gateway provides - besides various standard data as e.g. “Uptime” - the data of the CAN Statistics (see page 43).

These data can then be evaluated further by means of corresponding tools (SNMP manager). An appropriate ".mib" file for this is available in the installation directory of the host-driver.

The SNMP agent of the EtherCAN/2 gateway supports the versions 1 and 2c of the SNMP protocol. The community string is "public".

- **esdcp**

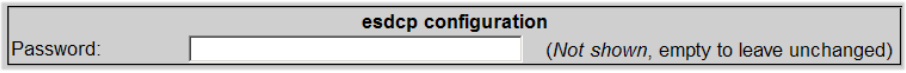


Figure 12: *esdcp configuration*

To change the esdcp password type in the new password in this input field.

Click the button Submit to save the changed data in a non-volatile memory of the EtherCAN/2 module. After a reboot the data is enabled.

3.2.2.3 Remote Logging

The *Remote Logging s*upport of the EtherCAN/2 module can be configured and enabled on this page. The module offers the feature to provide alarms and events not only on the local HTTP-server, but to forward them as e-mail to a SMTP-Server.
For the configuration of the SMTP parameters refer to *SMTP configuration* (see page 20).

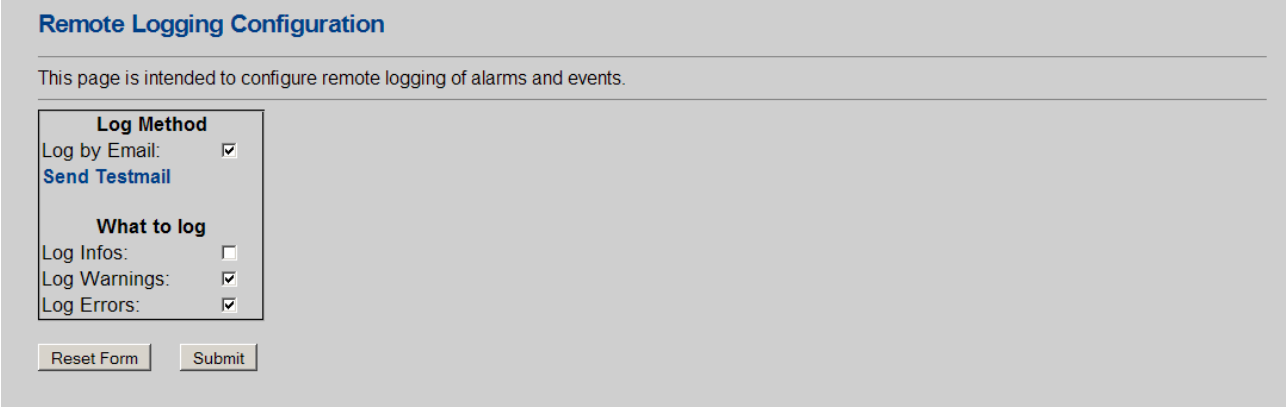


Figure 13: Configuration of the Remote Logging

The check box ❑ *Log by Email* enables/disables the Remote Logging Support.

Use the check boxes ❑ *Log Errors,* ❑ *Log Warnings* and ❑ *Log Infos* to configure which kind of event should be transmitted as e-mail.

Click the button Submit to save the changed data in a non-volatile memory of the EtherCAN/2 module. After a reboot the data is enabled.

3.2.2.4 Firmware Update

For an update of the firmware of the EtherCAN/2 click the menu item *Firmware Update* in the program window.

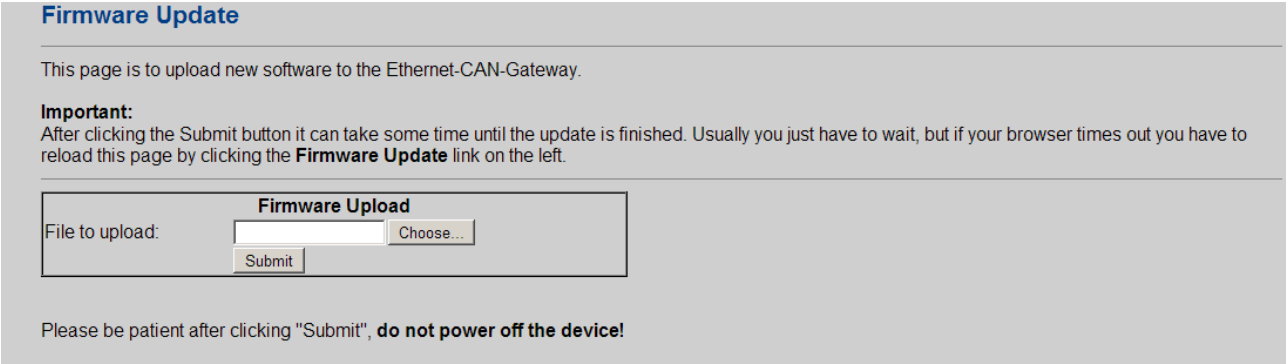


Figure 14: Firmware update

The upload of the file is done by means of the web browser. Enter the file name, or click the button Choose... to select a file.

Acknowledge the entry by clicking the button Submit . The firmware update is started now. This can take some time. The progress of the updates will be recorded.

NOTICE
During firmware update the power supply of the EtherCAN/2 must not be cut off, because the module could get into an inoperable state.

Example print of a firmware update protocol:

```
Starting firmware update... (Tue Aug 25 07:02:39 CEST 2009)

Serial No: GA000006
Hardware version: 1.1
Installed firmware version: 1.0.0
Update firmware version: 1.0.1


Updating kernel image...
Installed kernel image version: 1.0.0
Update kernel image version: 1.0.1

Erasing 128 Kibyte @ 0 --  0 % complete.
Erasing 128 Kibyte @ 20000 --  4 % complete.
Erasing 128 Kibyte @ 40000 --  8 % complete.
Erasing 128 Kibyte @ 60000 -- 12 % complete.
Erasing 128 Kibyte @ 80000 -- 16 % complete.
Erasing 128 Kibyte @ a0000 -- 20 % complete.
Erasing 128 Kibyte @ c0000 -- 25 % complete.
Erasing 128 Kibyte @ e0000 -- 29 % complete.
Erasing 128 Kibyte @ 100000 -- 33 % complete.
Erasing 128 Kibyte @ 120000 -- 37 % complete.
Erasing 128 Kibyte @ 140000 -- 41 % complete.
Erasing 128 Kibyte @ 160000 -- 45 % complete.
Erasing 128 Kibyte @ 180000 -- 50 % complete.
Erasing 128 Kibyte @ 1a0000 -- 54 % complete.
Erasing 128 Kibyte @ 1c0000 -- 58 % complete.
Erasing 128 Kibyte @ 1e0000 -- 62 % complete.
Erasing 128 Kibyte @ 200000 -- 66 % complete.
Erasing 128 Kibyte @ 220000 -- 70 % complete.
Erasing 128 Kibyte @ 240000 -- 75 % complete.
Erasing 128 Kibyte @ 260000 -- 79 % complete.
Erasing 128 Kibyte @ 280000 -- 83 % complete.
Erasing 128 Kibyte @ 2a0000 -- 87 % complete.
Erasing 128 Kibyte @ 2c0000 -- 91 % complete.
Erasing 128 Kibyte @ 2e0000 -- 95 % complete.
```

```
Writing data to block 0
Writing data to block 20000
Writing data to block 40000
Writing data to block 60000
Writing data to block 80000
Writing data to block a0000
Writing data to block c0000
Writing data to block e0000
Writing data to block 100000
Writing data to block 120000
Writing data to block 140000
Writing data to block 160000
Writing data to block 180000
Writing data to block 1a0000

Updating Ramdisk image...
Installed Ramdisk image version: 1.0.0
Update Ramdisk image version: 1.0.1

Erasing 128 Kibyte @ 0 --  0 % complete.
Erasing 128 Kibyte @ 20000 --  4 % complete.
Erasing 128 Kibyte @ 40000 --  8 % complete.
Erasing 128 Kibyte @ 60000 -- 12 % complete.
Erasing 128 Kibyte @ 80000 -- 16 % complete.
Erasing 128 Kibyte @ a0000 -- 20 % complete.
Erasing 128 Kibyte @ c0000 -- 25 % complete.
Erasing 128 Kibyte @ e0000 -- 29 % complete.
Erasing 128 Kibyte @ 100000 -- 33 % complete.
Erasing 128 Kibyte @ 120000 -- 37 % complete.
Erasing 128 Kibyte @ 140000 -- 41 % complete.
Erasing 128 Kibyte @ 160000 -- 45 % complete.
Erasing 128 Kibyte @ 180000 -- 50 % complete.
Erasing 128 Kibyte @ 1a0000 -- 54 % complete.
Erasing 128 Kibyte @ 1c0000 -- 58 % complete.
Erasing 128 Kibyte @ 1e0000 -- 62 % complete.
Erasing 128 Kibyte @ 200000 -- 66 % complete.
Erasing 128 Kibyte @ 220000 -- 70 % complete.
Erasing 128 Kibyte @ 240000 -- 75 % complete.
Erasing 128 Kibyte @ 260000 -- 79 % complete.
Erasing 128 Kibyte @ 280000 -- 83 % complete.
Erasing 128 Kibyte @ 2a0000 -- 87 % complete.
Erasing 128 Kibyte @ 2c0000 -- 91 % complete.
Erasing 128 Kibyte @ 2e0000 -- 95 % complete.
Writing data to block 0
Writing data to block 20000
Writing data to block 40000
Writing data to block 60000
Writing data to block 80000
Writing data to block a0000
Writing data to block c0000
Writing data to block e0000
Writing data to block 100000
Writing data to block 120000
Writing data to block 140000
Writing data to block 160000
Writing data to block 180000
Writing data to block 1a0000
Writing data to block 1c0000
Writing data to block 1e0000
Writing data to block 200000

Updating NV-RAM...
Installed NV-RAM image version: 1.0.0
Update NV-RAM image version: 1.0.1

Update successful. System reboots now... (Tue Aug 25 07:03:08 CEST 2009)
```

Figure 15: Firmware update protocol (example)

Wait until the protocol about the update is completely finished.
The system will reboot automatically.

3.2.2.5 CAN Bridge

On this page the “CAN Bridge” functionality of the EtherCAN/2 is configured. This one allows to connect two CAN networks via Ethernet.

To achieve this, this configuration page is used to set up two EtherCAN/2 modules so they are able to connect to each other to exchange the CAN messages.

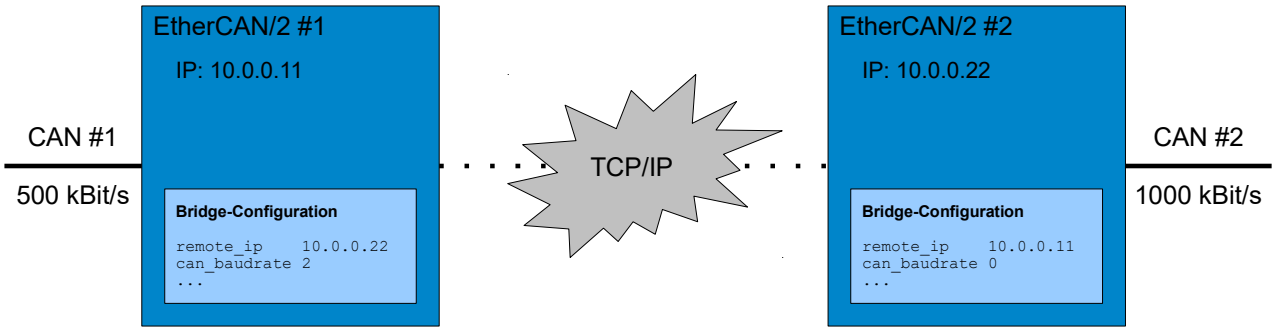


Figure 16: CAN Bridge sample usage

For that each EtherCAN/2 has to know the IP address[2] of the other side and its own CAN baud rate. Additionally it has to be configured which CAN-IDs are to be received from the local CAN network and transferred to the other side – to avoid unnecessary data transfer.

Besides that, a mapping from received CAN-IDs to send IDs may be set up to send the CAN frames from the other bridge side to the local CAN network with a different CAN-ID.

That configuration is made either directly within the web browser (Figure 17) or by a configuration file (see 3.2.2.5.6) that can be up- and download on that page. In Both cases a changed configuration becomes active only after the bridge-software was restarted, see “Restart CAN Bridge”.

It follows the description of that configuration page.

2 The connection is made by TCP port 41179, every side is TCP server as well as TCP client.

3.2.2.5.1 Short description/Quick start

On both sides of the bridge must be carried out at least the following steps on the configuration page:

1. Enter IP address of remote side
   - “Remote IP address” under “General”
2. Enter CAN baud rate
   - “Baud rate” under “CAN”
3. Select CAN-IDs
   - To transfer all IDs to the other side:
     Select “Clear and enable all IDs” under “CAN 29 bit filter” and
     “Clear and enable all IDs” under “CAN 11 bit filter”
4. Restart the device in bridge mode
   - Button “Restart CAN Bridge”

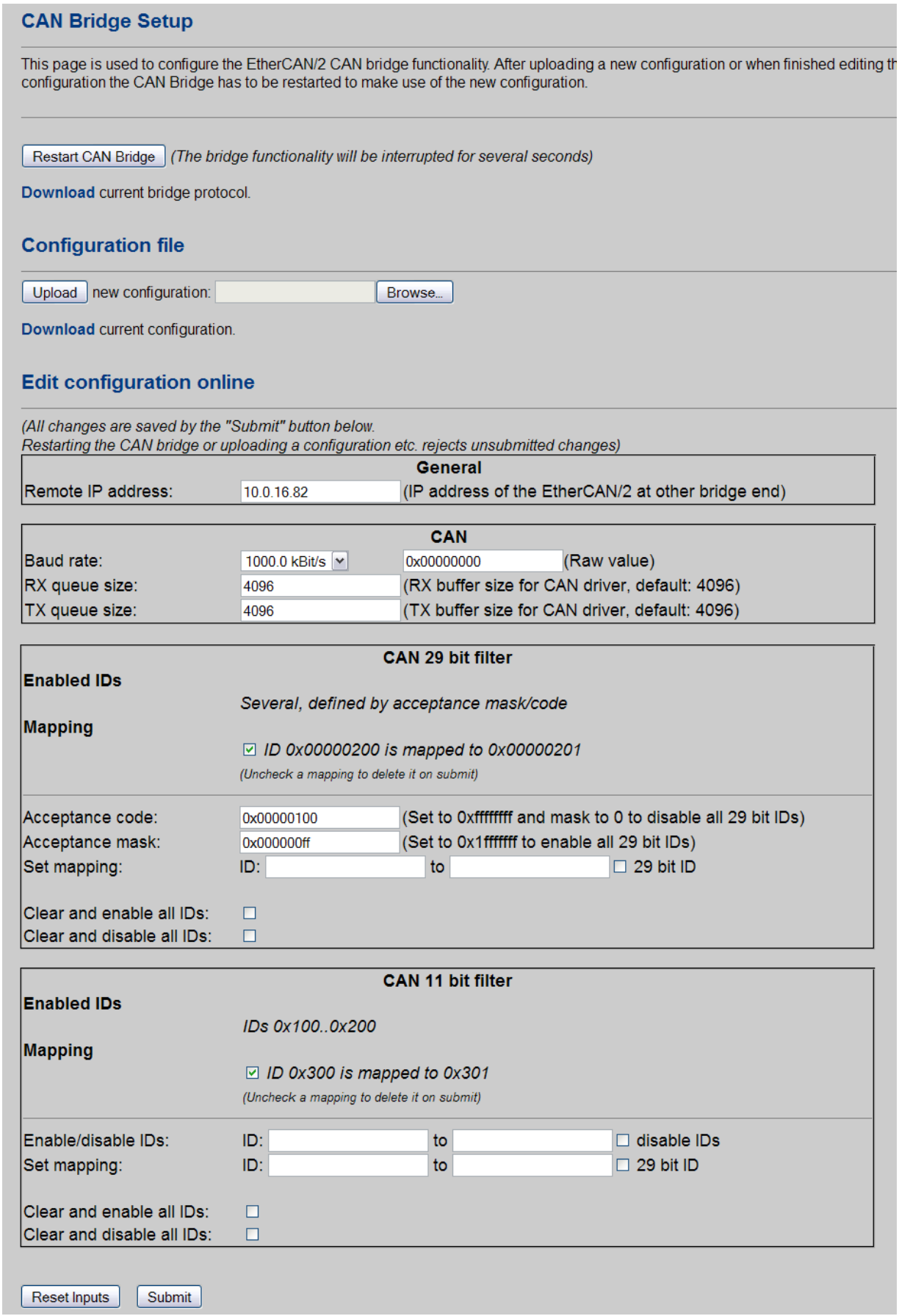


Figure 17: CAN Bridge configuration page

3.2.2.5.2 Important notes

- All numbers within the configuration page and -file are interpreted as decimal (base 10) by default. To use base 16 (“Hex”) a “0x” has to be prefixed. To use base 8 (“Octal”) a “0” has to be prefixed. Therefore an input of e.g. “10” equals an input of “0xa”/”0xA” and “012”
- CAN-IDs: These are displayed in hexadecimal notation, thereby 3 digits are used for 11 bit IDs and 8 digits for 29 bit IDs: “0x100” → 11 bit ID, “0x00000100” → 29 bit ID.
  When entering a CAN-ID this distinction is not possible and the input is interpreted as usual number (The configuration page uses a check box “29 Bit ID”, the configuration file uses bit 29 within the ID as marker – please follow the descriptions)
- A restart of a bridge side leads to a connection interrupt for the other side, too. This again triggers a restart of the bridge software there.
  Therefore a bridge-software restart on one side leads to an interrupted bridge functionality for about 45 seconds
- In case a bridge side can’t connect to the other it will retry to do so every 10 seconds approx. When no remote IP address is given the bridge-software is deactivated
- Loss of CAN frames: Despite the (usually) higher bandwidth of the TCP/IP connection between the bridge sides it can not be guaranteed there’s no lost CAN frame at all.
  On every side receive and send buffers for the CAN frames exist: They are read from the receive buffer of the network interface and transferred to the send buffer of the CAN interface; vice versa they are read from the receive buffer of the CAN interface and transferred to the network interface’s send buffer. During that it might happen that a buffer runs full – in that case CAN frames will be lost.
  This can be shown clearly with different CAN baud rates on each bridge side: While frames are sent from the “faster” side at its high rate the CAN send buffer on the other side will fill up because these frames can not be read from the network buffer as fast. When the buffer is full CAN frames must be dismissed.
  Beside different baud rates this can also happen together with a very high CAN bus load for other reasons, as there may be short delays at any time, e.g. by other network traffic or other actions the EtherCAN/2 performs simultaneously (Web server, SNMP server, etc.).
  Therefore the CAN buffer sizes may be adapted, see also “RX/TX queue size” in 3.2.2.5.5
- The deactivation of the CAN bridge functionality is done by leaving the “Remote IP address” empty

NOTICE
If the EtherCAN/2 is used e.g. as CAN interface for a PC or in bridge mode at the same time, it has to be ensured that no conflicts arise. (Especially different CAN baud rates must not be set at the same time)

3.2.2.5.3 Miscellaneous

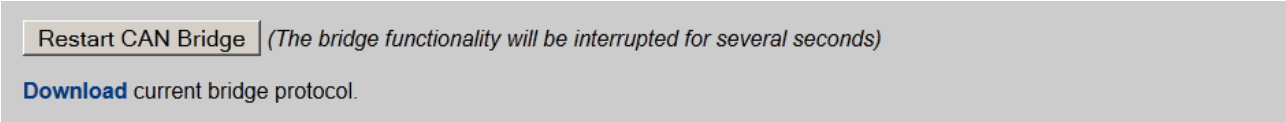


Figure 18: CAN Bridge – Miscellaneous settings

- Restart CAN Bridge
  - Ends the bridge-software and restarts it. With that the bridge configuration is also re read. In any case, i.e. even if the remote IP address is unchanged, the bridge connection is lost and no CAN frames are transferred during that time
    The bridge protocol is also restarted and its first entries are shown. To show that log at any time, see next item
- Download current bridge protocol
  - Downloads the protocol of the bridge-software. Used for trouble-shooting: it contains information about connection losses and CAN initialization errors etc.

3.2.2.5.4 Configuration file

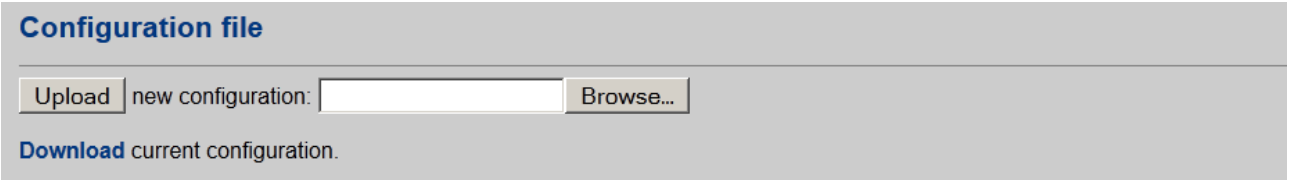


Figure 19: CAN Bridge – “Configuration file”

- Upload new configuration
  - Used to upload a bridge configuration file to the EtherCAN/2. By the “Browse” button (or “Choose” etc., depends on browser/language) the file is selected, the “Upload” button transfers it then. This will overwrite all current settings. For details about the configuration file format see 3.2.2.5.6
- Download current configuration
  - Used to download the bridge configuration. The file is always recreated, i.e. it will contain all changes made online, but any unused settings or comments from an eventually uploaded file will be ignored

3.2.2.5.5 Edit Configuration Online

Used to edit the configuration directly within the browser. If a configuration file was uploaded before its settings are already included here.
All changes will be transferred to the EtherCAN/2 only with usage of the “Submit” button, i.e. when leaving the page or triggering any other action, such as a bridge restart or configuration file upload, all inputs made here will be lost.

- **General**

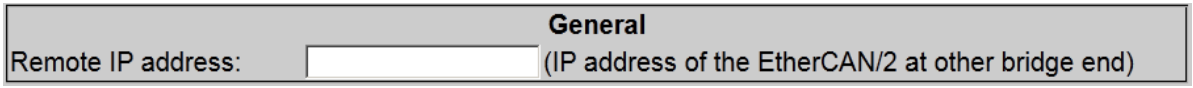


Figure 20: CAN Bridge - “General”

  - Remote IP address
    - The IP address of the other bridge side. On that side the “Remote IP address” must be the address of this side. (A hostname can’t be entered)
      When this is left empty the bridge functionality is disabled

- **CAN**

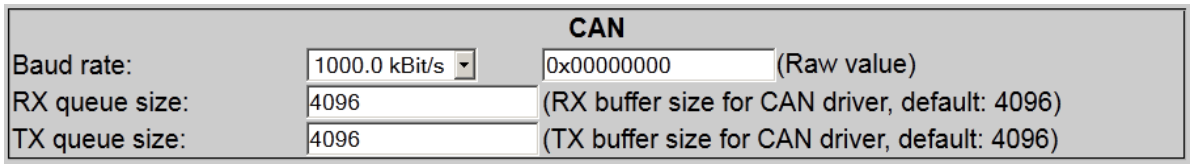


Figure 21: CAN Bridge - “CAN”

  - Baud rate
    - The CAN baud rate that is to be set locally. Does not necessarily have to match the other side’s CAN baud rate.
      Attention: With different baud rates it might happen easily that on the “slower” side not all CAN frames can be sent, see also “Loss of CAN frames” in 3.2.2.5.2.
      “Raw value”: (Only used when “By raw value” is selected in the drop down list) This is the value that is internally given to the canSetBaudrate() function, for details please refer to NTCAN API manual at “canSetBaudrate”.
      Usually it’s not necessary to use this value, selecting a baud rate by the drop down list is sufficient
  - RX queue size
    - Determines the size of the CAN receive buffer in the local CAN driver. High values, such as the default of 4096, nearly make sure no CAN frame is lost, but might also lead to a high delay for CAN frames passing the bridge – so when high latencies can’t be tolerated, but lost frames can be tolerated, this should be set to a very low value. See also “Loss of CAN frames” in 3.2.2.5.2
  - TX queue size
    - Determines the size of the CAN send buffer in the local CAN driver, analogue to “RX queue size”

- **CAN 29 bit filter**

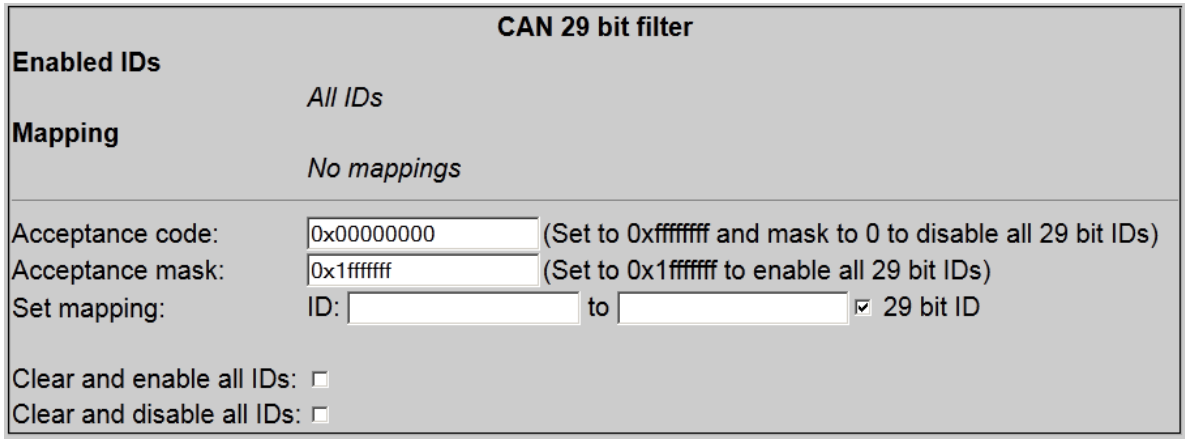


Figure 22: CAN Bridge - "CAN 29 bit filter"

- Enabled IDs
  - Lists which 29 bit CAN-IDs are received, i.e. transferred to the other bridge side
- Mapping
  - Lists the 29 bit CAN-ID-Mappings, i.e. on which CAN-ID a frame from the remote side is sent to the local CAN interface. Unchanged accepted CAN-IDs ("1:1 mapping") are not listed. When an item is unchecked it is deleted on "Submit"
- Acceptance code / Acceptance mask
  - Determines which 29 bit CAN-IDs are received (listed under "Enabled IDs"). For details see NTCAN API Manual "Transmitting and Receiving CAN 2.0B (29-Bit) Messages".
    Examples:
    - Code: 0x100, Mask: 0 → Only the 29 bit CAN-ID 0x00000100 is received
    - Code: 0x100, Mask: 0xff → The 29 bit CAN-IDs [0x00000100..0x000001ff] are received
- Set mapping
  - Adds an entry to "Mapping". The first input is the receive CAN-ID, the second the send ID. The first input is always interpreted as 29 bit CAN-ID, the second according to the checkbox "29 bit ID", i.e. when checked an input of e.g. "0x100" is interpreted as 29 bit ID, else as 11 bit ID
- Clear and enable all IDs
  - Deletes all mapping entries and resets "Acceptance code/mask" to receive/forward all 29 bit CAN frames
- Clear and disable all IDs
  - Deletes all mapping entries and resets "Acceptance code/mask" to receive/forward no 29 bit CAN frames

- **CAN 11 bit filter**

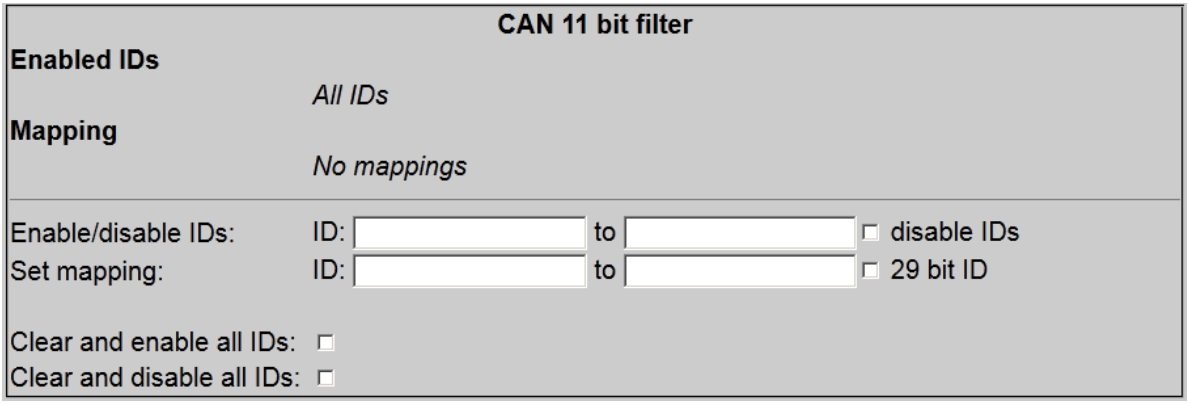


Figure 23: CAN Bridge - "CAN 11 bit filter"

  - Enabled IDs
    - Lists which 11 bit CAN-IDs are received, i.e. transferred to the other bridge side
  - Mapping
    - Lists the 11 bit CAN-ID-Mappings, i.e. on which CAN-ID a frame from the remote side is sent to the local CAN interface. Unchanged accepted CAN-IDs ("1:1 mapping") are not listed. When an item is unchecked it is deleted on "Submit"
  - Enable/disable IDs
    - Used to add and remove entries in "Enabled IDs". The first input is the 11 bit CAN-ID to be added/removed. When "disable IDs" is checked, it's removed, else it is added.

      The second input is optional and used to add/remove multiple items at a time: when a value is given here, it's interpreted as last value of a range to be added/removed – the first input determines the first value of that range then
  - Set mapping
    - Adds an entry to "Mapping". The first input is the receive CAN-ID, the second the send ID. The first input is always interpreted as 11 bit CAN-ID, the second according to the checkbox "29 bit ID", i.e. when checked an input of e.g. "0x100" is interpreted as 29 bit ID, else as 11 bit ID
  - Clear and enable all IDs

    Deletes all mapping entries and sets "Enable/disable IDs" so that all 11 Bit CAN frames are received/forwarded
  - Clear and disable all IDs

    Deletes all mapping entries and sets "Enable/disable IDs" so that no 11 Bit CAN frames are received/forwarded
- **Reset Inputs**
  - Rejects all changes made since last loading the page or submitting changes and restores the previous values
- **Submit**
  - Transmits all changes to the EtherCAN/2

3.2.2.5.6 Configuration file

The configuration file for the bridge-software is a simple text file that consists of key value pairs ordered in lines, see Figure 25.
The file name during upload is ignored, line breaks may be either “CR/LF” or “LF”, lines starting with “#” or “;” are comments and ignored, as well as empty lines. As delimiter between key and value spaces or tabs may be used. **Case sensitive**.

Key	Default Value	Comment
`version`	`0`	Currently unused (for future changes)
`remote_ip`		Equals “Remote IP address” in online configuration
`disable_nagles`	`1`	When “`0`” is specified the so called “Nagle algorithm” is not deactivated. When that is activated CAN frames might be forwarded to the other side with a little delay, to assemble multiple CAN frames in a single Ethernet frame – this saves bandwidth and might be useful with low bandwidth connections
`can_baudrate`	`0x7fffffff`	Equals “Raw value” of the baud rate setting in online configuration. “`0x7fffffff`“ means no CAN baud rate should be set. All other values according to baud rate indices in NTCAN API manual (see “canSetBaudrate”)
`can_rxqueue`	`4096`	Equals “RX queue size” in online configuration
`can_txqueue`	`4096`	Equals “TX queue size” in online configuration
`can_29bit_acceptmask`	`0`	Equals “Acceptance mask” in online configuration (“`0x1fffffff`”: all 29 bit frames are received)
`can_29bit_acceptcode`	`0xffffffff`	Equals “Acceptance code” in online configuration (“`0xffffffff`” and “`0`” for “`can_29bit_acceptmask`”: no 29 bit frames are received)
`can_29bit_map`		Adds a 29 bit mapping entry. To add e.g. ID 0x00000100 to be sent as ID 0x101 the line shall be: “`can_29bit_map 0x100 0x101`”. The receive ID is always interpreted as 29 bit ID, the send ID has to use bit 29 as marker. (So this sample line sends the 29 bit ID 0x100 as 11 bit ID 0x101) Furthermore the number of 29 bit mappings is limited to 16
`can_11bit_enable`		Adds entries to “Enabled IDs” of the 11 bit filter. As value the first and last ID of the range to add is used. For the single ID 0x100 the line is e.g. ”`can_11bit_enable 0x100 0x100`”. For all 11 bit IDs: “`can_11bit_enable 0 2047`”
`can_11bit_map`		Adds an 11 bit mapping entry. To add e.g. ID 0x100 to be sent as ID 0x00000101 the line shall be “`can_11bit_map 0x100 0x20000101`”. The receive ID is always interpreted as 11 bit ID, the send ID has to use bit 29 as marker. (So this sample line sends the 11 Bit ID 0x100 as 29 Bit ID 0x00000101)

Sample file

Line

```
1        #
2        # esd EtherCAN/2 CAN Bridge configuration file.
3        #
4
5
6        remote_ip                       10.0.1.2
7
8        can_baudrate                    0x00000000
9
10       can_29bit_acceptmask            0x000000ff
11       can_29bit_acceptcode            0x00000100
12       can_29bit_map                   0x00000120    0x00000100
13       can_29bit_map                   0x00000121    0x20000101
14
15       can_11bit_enable                0             2047
16       can_11bit_map                   0x00000100    0x20000100
17
```

Figure 24: CAN Bridge – Sample configuration file

- Line 6: (remote_ip)
  - The remote IP address is “10.0.1.2”
- Line 8: (can_baudrate)
  - The local CAN baud rate is 1000 kBit/s
- Line 10/11: (can_29bit_acceptmask/can_29bit_acceptcode)
  - The 29 bit CAN-IDs [0x00000100..0x000001ff] are received and sent to the remote side
- Line 12: (can_29bit_map)
  - When the 29 bit CAN-ID 0x00000120 is received from the remote side it is sent as 11 bit CAN-ID 0x100 to the local CAN interface
- Line 13: (can_29bit_map)
  - When the 29 bit CAN-ID 0x00000121 is received from the remote side it is sent as 29 bit CAN-ID 0x101 to the local CAN interface
- Line 15: (can_11bit_enable)
  - All 11 bit CAN-IDs [0x000..0x7ff] are received and sent to the remote side
- Line 16: (can_11bit_map)
  - When the 11 bit CAN-ID 0x100 is received from the remote side it is sent as 29 bit CAN-ID 0x100 to the local CAN interface

3.2.2.6 CAN TCP Client

On this page the “CAN TCP Client” functionality of the EtherCAN/2 is configured. This function allows it to forward CAN frames to a TCP server.

INFORMATION
This mode is a special application that is not needed for the usual use case of the EtherCAN/2, such as CAN interface for PCs or bridge mode.
(In this mode the EtherCAN/2 acts as client and autonomously tries to connect to a PC – usually the connection is initiated by the other side.)

To use this mode, at least the TCP server, its port and the CAN baud rate have to be entered, see Figure 25 – the protocol used for the communication is described in 3.2.2.6.1.
The configuration scope is limited: basically all CAN frames are transmitted (no ID filtering etc.) and the EtherCAN/2 constantly tries to keep the connection established (unlimited attempts to reconnect when connection was lost).

NOTICE
If the EtherCAN/2 is used e.g. as CAN interface for a PC or in bridge mode at the same time, it has to be ensured that no conflicts arise. (Especially different CAN baud rates must not be set at the same time)

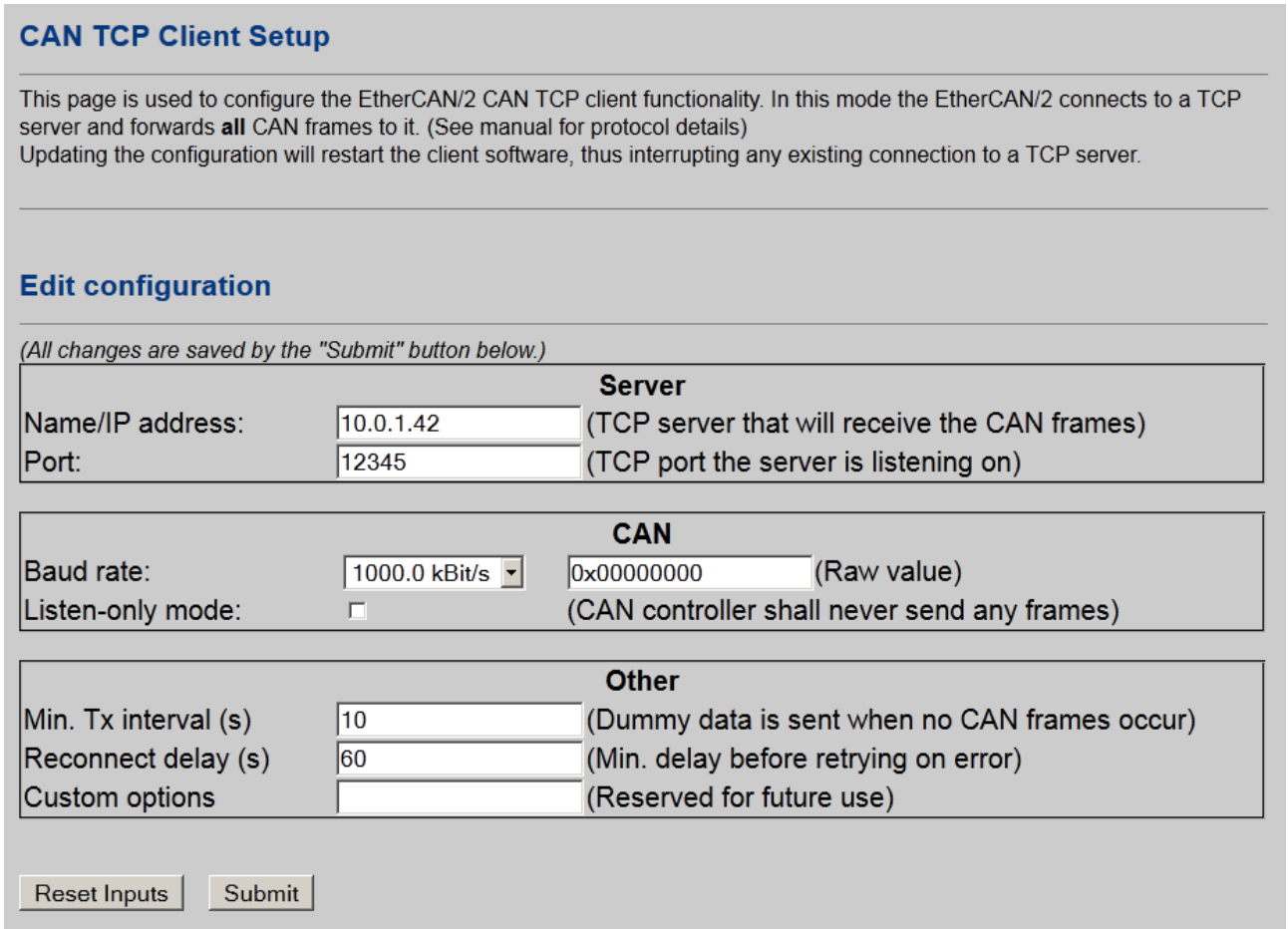


Figure 25: CAN TCP Client – configuration page

- **“Server”**
  - “Name/IP address”
    - IP address or name of the TCP server. A name can only be used if a name server is configured, see 3.2.2.2
  - “Port”
    - The TCP port number on which the server is waiting for the connection of the EtherCAN/2
- **“CAN”**
  - “Baud rate”
    - The CAN baud rate to be set locally
    - “Raw value”: (Used only if the selection list is on “By raw value:”)
      This is the value that is internally used for the canSetBaudrate() function, see NTCAN API manual for details, keyword “canSetBaudrate()”. (Usually it is not necessary to use this statement – setting the baud rate by the selection list is sufficient)
  - “Listen-only mode”
    - If this is enabled, then, while setting the baud rate, it's also chosen that the CAN controller shall work in “listen-only mode”, i.e. the EtherCAN/2 acts as a passive bus member and does not even send an acknowledge.
      (Keep in mind that when there's only one active bus member, it will retry sent frames for many times due to missing acknowledgment – even these frames will be received and sent to the TCP server by the EtherCAN/2 then)
- **“Other”**
  - “Min. Tx interval (s)”
    - The minimal interval data shall be sent to the TCP server, i.e. if no CAN frames are to be sent a “beacon” command is sent instead, (see 3.2.2.6.1) to signal that the TCP client is still active.
      If 0, this is disabled
  - “Reconnect delay (s)”
    - Waiting time for a failed connection before attempting to re-establish the connection. (At loss of connection the reconnection is retried immediately. A connection attempt is terminated after 10 seconds)
  - “Custom options”
    - Reserved for future use
- **“Reset Inputs”**
  - Rejects all changes made within “Edit configuration online” since last loading the page or submitting changes and restores the previous values
- **“Submit**”
  - Transmits all changes to the EtherCAN/2. The Client software on the EtherCAN/2 is restarted then, i.e. an existing connection to the TCP server is lost temporary (approx. 10 to 15 seconds)

3.2.2.6.1 Protocol

The data transmitted via TCP/IP is divided into “commands”. Each command begins with a byte that identifies the command. This is followed by the specific data, which, depending on the command, can be of different length.
All values within the data are, unless otherwise stated, transmitted in “network byte order”, i.e. “Big-endian”.
When the connection is established the EtherCAN/2 starts with sending its version information, unasked – the server has to do that, too, else the EtherCAN/2 won't accept commands from the server.

Command (ASCII)	Name	Data
66 ('B')	“Beacon”	None
86 ('V')	Version information	*31 Bytes*
118 ('v')	Version information request	None
99 ('c')	CAN frame array (little-endian)	*2 Bytes* (= *n* = No. of frames) *+ n * 16 Bytes*
67 ('C')	CAN frame array (big-endian)	*2 Bytes* (= *n* = No. of frames) *+ n * 16 Bytes*
116 ('t')	Reserved	*2 Bytes* (= *n*) *+ n * 24 Bytes*
84 ('T')	Reserved	*2 Bytes* (= *n*) *+ n * 24 Bytes*
112 ('p')	“Ping”	*8 Bytes*
80 ('P')	“Pong”	*16 Bytes*
100 ('d')	Change baud rate	*4 Bytes*
68 ('D')	Change baud rate reply	*8 Bytes*
103 ('g')	Read baud rate	None
71 ('G')	Read baud rate reply	*8 Bytes*
	Other/Future	*2 Bytes* (= *n* = Payload length) *+ n Bytes*

- “Beacon”
  - No data, just used to signal client is still active
- Version information
  - Must be sent as first command by every side, for contents see 3.2.2.6.2
- Version information request
  - Request for the other side to send its version information

- CAN frame array (little-endian)
  - The first two bytes specify the number of CAN frames that have to be read. These are followed by the CAN frames, which are transferred as follows: (see also 3.2.2.6.3)

Byte	Member name	Short description
0..3	id	CAN ID*
4	len	Length of CAN data*
5	msg_lost	Lost CAN frames
6..7	reserved	Reserved/unused
8..15	data	CAN data

  *Some bits with special meaning, please check CAN Frame struct (3.2.2.6.3)
- CAN frame array (big-endian)
  - As “CAN frame array (little-endian)”, but here the bytes of the CAN-ID are transferred in reverse order
- “Ping”
  - Request to the other side to send a “Pong”. The 8 data bytes are returned with that pong. These bytes usually are the system's time stamp (its frequency is sent with the version information – therefore this allows to approximate the time difference to the other side)
- “Pong”
  - Reply to the “Ping” command. The first 8 bytes are the bytes sent with the “Ping”, the other 8 bytes are the local time stamp when sending this
- Change baud rate
  - The 4 bytes refer to the “uint32_t” value that is used internally for the canSetBaudrate() function, see NTCAN API manual for details.
    (The “NTCAN_AUTOBAUD” feature is currently not supported by the EtherCAN/2)
- Change baud rate reply
  - The first 4 bytes are the same that were sent with “Change baud rate”, the other 4 bytes are the result. 0 indicates that the baud rate was changed successfully
- Read baud rate
  - Request a “Read baud rate reply” from the other side
- Read baud rate reply
  - The first 4 bytes contain the result of the attempt to read the baud rate. The other 4 bytes contain the baud rate value (according to the NTCAN API manual, see there for details). A result value of 0 indicates a successful operation, only then the baud rate value is valid
- Reserved/other/future commands
  - These are already defined to be compatible with future versions of the software, as every command has to be read from the TCP socket.
    For example, when the unknown command 'X' is received at first two bytes have to be read. These bytes specify the payload length, the length that has to read additionally. Thereafter follows the next command, and so on

3.2.2.6.2 Version information struct

Member name	Byte	
magic	0..2	Always 99, 50, 116 (ASCII: “c2t”). Used to ensure that the other side is actually an EtherCAN/2 CAN TCP Client/Server
reserved	3..6	Reserved for future use
version	7..10	Protocol version number
serial	11..14	Serial No. of the EtherCAN/2. The format is described in the NTCAN API manual
reserved	15..18	Reserved for future use
reserved	19..22	Reserved for future use
tsFrequency	23..30	Frequency of the timestamps used in “Ping” / “Pong” commands (in Hz)

Reserved values must be set to 0. Protocol version number currently is 0, too. (React on the client's version number only as/when described here)

3.2.2.6.3 CAN Frame struct

Taken from NTCAN API documentation of the CAN-SDK: http://esd.eu/en/products/can-sdk

Member name	Bit	
id	0..28	CAN ID (Bits 12..28 are zero in 11 Bit IDs)
	29	Flag 29 bit ID (0: 11 bit ID, 1: 29 bit ID)
	30	Flag “event” (1: frame is event instead of normal CAN frame)
	31	Reserved
len	0..3	Length of CAN data (member “data”). Values greater than 8 are possible, in that case only 8 data bytes exist anyway.
	4	Flag “RTR” (1: “Remote Transmission Request”, see CAN specification. Member “data” is not valid then)
	5	Flag “Mode” (1: Frame is “Interaction Frame”, i.e. it was sent from another software on the EtherCAN/2 itself)
	6..7	Reserved
msg_lost	0..7	“Rx Lost” counter (No. of lost CAN frames. Limited to 255, i.e. when more than 255 frames were lost this value is still set to 255) Lost frames usually occur when the EtherCAN/2 failed to send them to the TCP connection in time, i.e. when the connection is too slow or the server does not read them fast enough from its client socket.
reserved[0]..[1]	0..7	Reserved
data[0]..[7]	0..7	CAN data bytes

3.2.2.7 WebSocket server

On this page the EtherCAN/2 WebSocket server is configured. That server allows using the ELLSI protocol over Websockets, and thus “CAN in the browser”.
(ELLSI is an own, packet oriented protocol to transfer CAN- frames and commands – it’s documented separately, see “ELLSI Software Manual” which is delivered by the EtherCAN/2 host driver setup)
The HTML pages the WebSocket server delivers are completely user defined and uploaded to the EtherCAN/2 on this page. The delivery state includes comprehensive samples.

INFORMATION
Just activate „Server enabled“, click „Submit“ and study these samples first. With the default port 81 the URL for the browser is „**http://***<IP-Address>***:81**“.
(The sample files are found in the EtherCAN/2 host driver setup, too. Used as base for own applications should be its **esd-ellsi.js**)

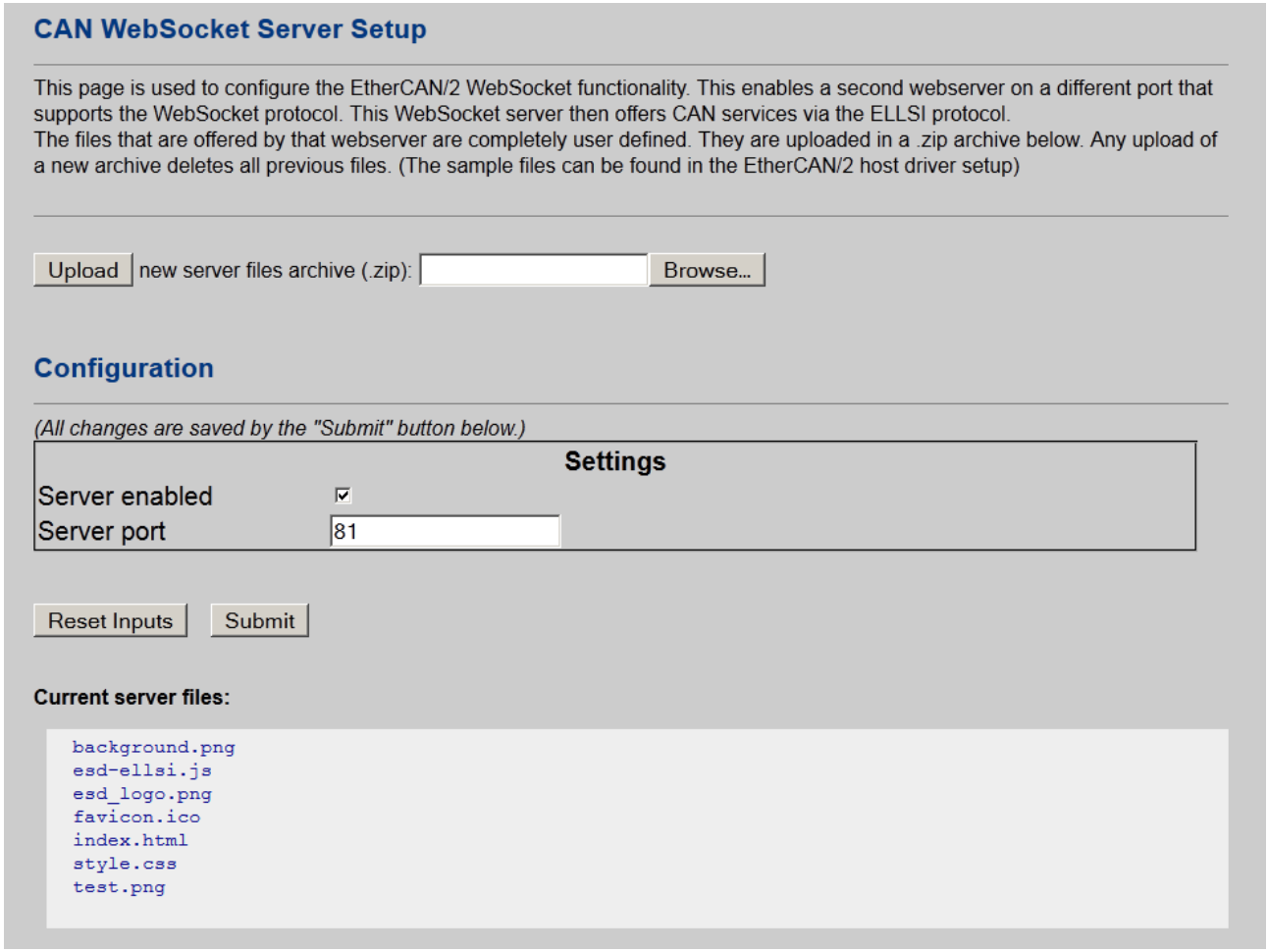


Figure 26: WebSocket server – configuration page

NOTICE
If the EtherCAN/2 is used e.g. as CAN interface for a PC or in bridge mode at the same time, it has to be ensured that no conflicts arise. (Especially different CAN baud rates must not be set at the same time)

- **„Upload new server files archive“**
  - With this the files the WebSocket server shall deliver are transferred to the EtherCAN/2. They must be within a “.zip” file.

    The upload of such a file deletes all existing files. A re-download of an archive from the EtherCAN/2 is not possible. The currently existing files are listed under “Current server files” below

    NOTICE
    Hints at archive:
    - It must contain an **index.html** file. (Must not be in a sub directory)
    - All files extracted must not exceed 20 MB total
    - As compression format/method “ZIP/Deflate” must be used, newer methods/extensions are not supported by the EtherCAN/2
- **„Settings“**
  - „Server enabled“
    - Determines whether the WebSocket server shall be active
  - „Server port“
    - Determines the TCP port of the server. The default is port 81. (The EtherCAN/2 denies the ports 80, 161, 162, 2209, 2210 and 41179)
- **“Reset Inputs”**
  - Rejects all changes made within “Configuration” since last loading the page or submitting changes and restores the previous values
- **“Submit**”
  - Transmits all changes to the EtherCAN/2. The WebSocket server on the EtherCAN/2 is restarted then, i.e. any existing connection to it is lost (restart takes approx. 10 to 15 seconds)

3.2.2.8 Reboot

Click the menu item *Reboot* in the program window to reboot the system .

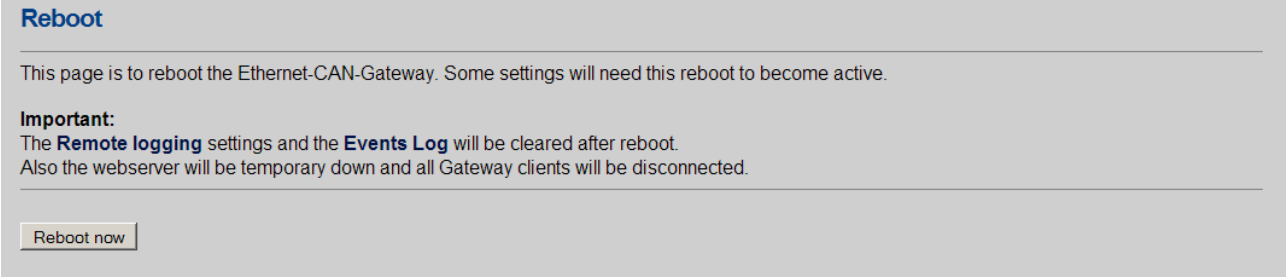


Figure 27: Reboot

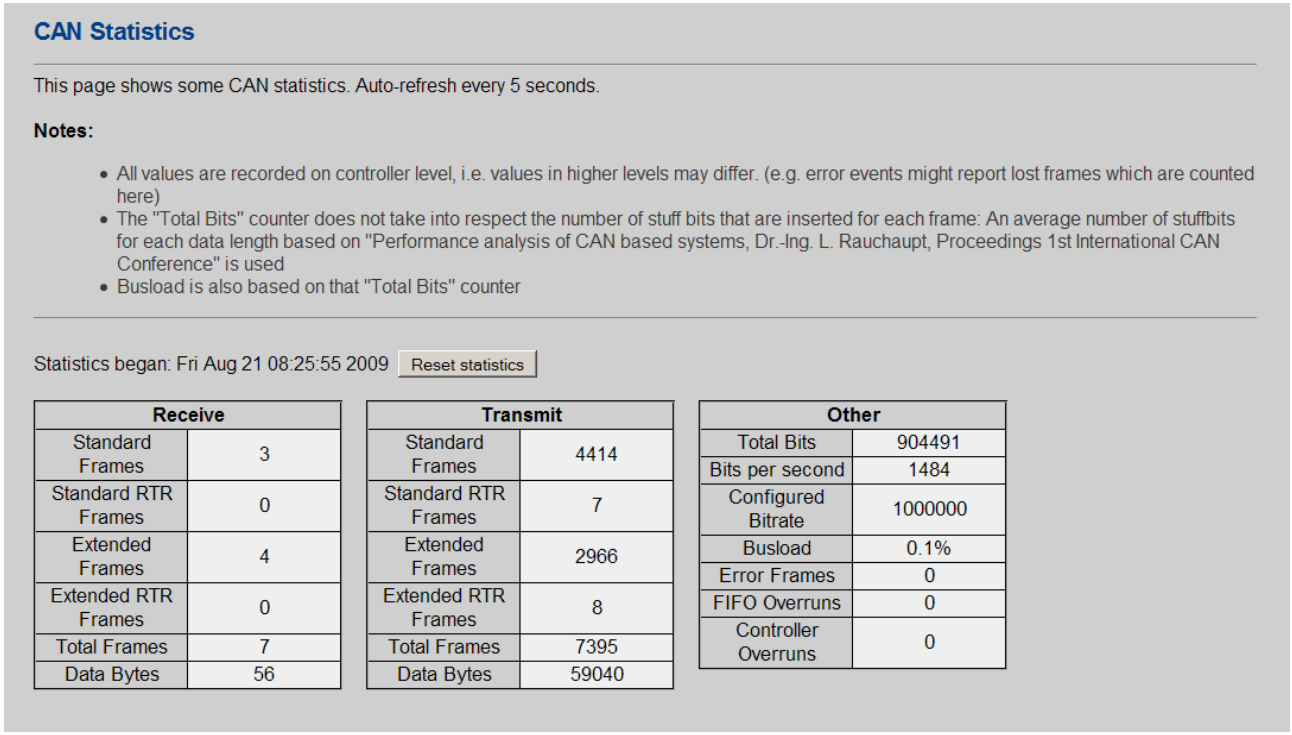
To reboot the system click the button Reboot now .

NOTICE
The **Events Log** will be cleared after reboot.
Furthermore the web server will be shut down and all links of the gateway clients/CAN Bridge will be disconnected.

3.2.3 Status

3.2.3.1 CAN Statistics

To view the status of the CAN bus click the menu item *CAN* in the program window.



CAN Statistics

This page shows some CAN statistics. Auto-refresh every 5 seconds.

Notes:

- All values are recorded on controller level, i.e. values in higher levels may differ. (e.g. error events might report lost frames which are counted here)
- The "Total Bits" counter does not take into respect the number of stuff bits that are inserted for each frame: An average number of stuffbits for each data length based on "Performance analysis of CAN based systems, Dr.-Ing. L. Rauchaupt, Proceedings 1st International CAN Conference" is used
- Busload is also based on that "Total Bits" counter

Statistics began: Fri Aug 21 08:25:55 2009 Reset statistics

Receive	
Standard Frames	3
Standard RTR Frames	0
Extended Frames	4
Extended RTR Frames	0
Total Frames	7
Data Bytes	56

Transmit	
Standard Frames	4414
Standard RTR Frames	7
Extended Frames	2966
Extended RTR Frames	8
Total Frames	7395
Data Bytes	59040

Other	
Total Bits	904491
Bits per second	1484
Configured Bitrate	1000000
Busload	0.1%
Error Frames	0
FIFO Overruns	0
Controller Overruns	0

Figure 28: Status CAN

The values shown are also provided via SNMP (see page 20) and can be evaluated by means of additional tools.

3.2.3.2 Bridge

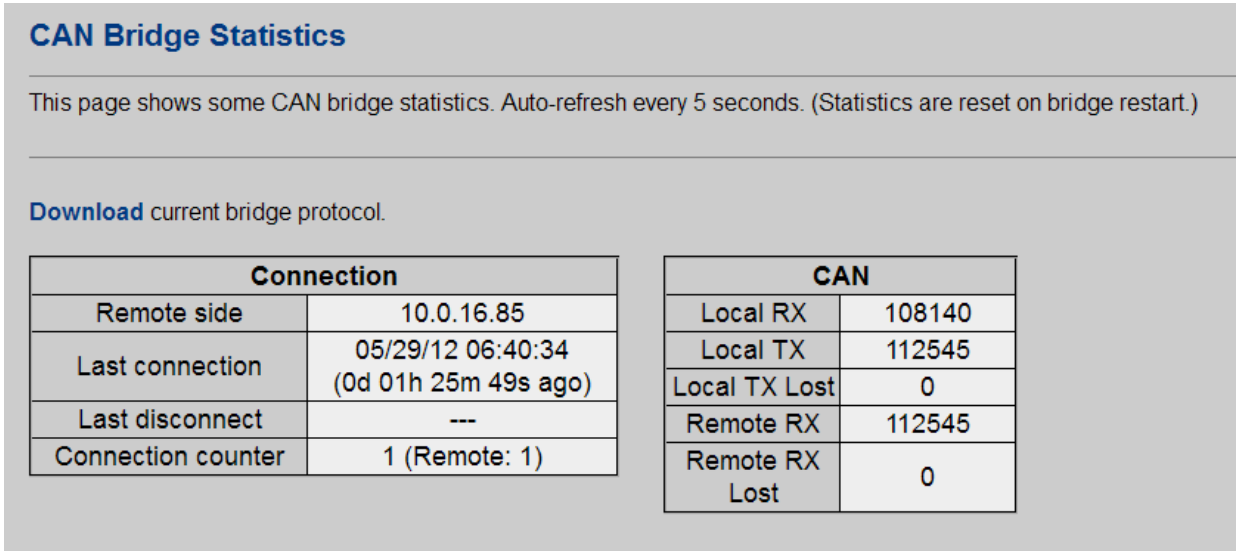


Figure 29: Status Bridge

“Connection counter”: Shows the number of successful TCP connections to the remote side. The value in brackets reflects the number of connections from the remote side.

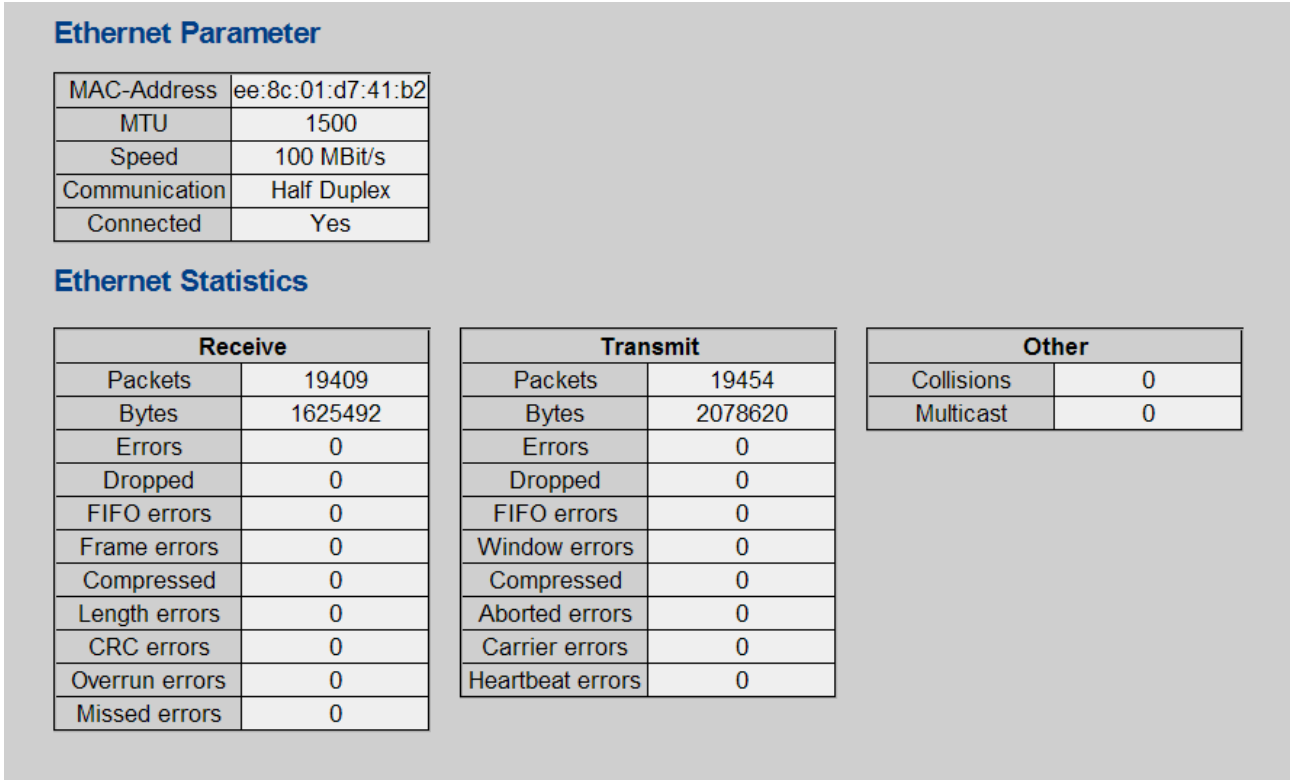
“Local TX Lost”: The number of CAN frames received from remote side that failed to be sent to local CAN interface. (Under some circumstances the actual value can be higher than displayed here)

“Remote RX Lost”: The number of CAN frames the remote side could not receive from its local CAN interface. (Under some circumstances the actual value can be higher than displayed here)

3.2.3.3 Ethernet

The Ethernet status can be displayed by choosing the menu item *Ethernet.*

On the Ethernet page the current transmission speed (10/100 Mbit/s), the communication mode (half/full duplex) and the MAC-ID of the EtherCAN/2 module are displayed among a number of statistical parameters of the Ethernet link.



Ethernet Parameter

MAC-Address	ee:8c:01:d7:41:b2
MTU	1500
Speed	100 MBit/s
Communication	Half Duplex
Connected	Yes

Ethernet Statistics

Receive	
Packets	19409
Bytes	1625492
Errors	0
Dropped	0
FIFO errors	0
Frame errors	0
Compressed	0
Length errors	0
CRC errors	0
Overrun errors	0
Missed errors	0

Transmit	
Packets	19454
Bytes	2078620
Errors	0
Dropped	0
FIFO errors	0
Window errors	0
Compressed	0
Aborted errors	0
Carrier errors	0
Heartbeat errors	0

Other	
Collisions	0
Multicast	0

Figure 30: Ethernet status

3.2.3.4 Connected clients

Click the menu item *Connected clients* for the status of the EtherCAN- and ELLSI clients.

Ethernet								CAN		
IP-Address	Port	Client-Port	Cmd-Frames	Rx-Bytes	Tx-Bytes	Keep-Alive-Frames	Connect-Time	Rx-Frames	Tx-Frames	Tx-Done-Frames
10.0.16.79	22080	53240	3944	1520644	1607084	0	04/16/13 13:13:48	19	39375	39090

ELLSI (UDP)

Idx	Reg.-count	Last Reg.	Last Rx	Last Tx	RX Frames	TX Frames	RX Out of Order	TX failed
0	1	28 sec	13 sec	7 sec	0	0	0	0
1	0	---	---	---	0	0	0	0
2	0	---	---	---	0	0	0	0
3	0	---	---	---	0	0	0	0
4	0	---	---	---	0	0	0	0
5	0	---	---	---	0	0	0	0
6	0	---	---	---	0	0	0	0
7	0	---	---	---	0	0	0	0

ELLSI (WebSocket)

Idx	Reg.-count	Last Reg.	Last Rx	Last Tx	RX Frames	TX Frames	RX Out of Order	TX failed
0	2	>4 min	>4 min	>4 min	0	0	0	0
1	1	>4 min	>4 min	>3 min	0	0	0	0
2	1	>4 min	>3 min	>2 min	1	1	0	0
3	1	>3 min	>2 min	104 sec	0	0	0	0
4	1	>2 min	102 sec	43 sec	0	0	0	0
5	1	100 sec	41 sec	3 sec	19	19	0	0
6	1	40 sec	2 sec	2 sec	0	0	0	0
7	0	---	---	---	0	0	0	0

Figure 31: Connected clients

ELLSI Clients

As the total number of connected clients is fixed, the number of table lines is fixed, too. Lines which contain clients that are not connected (or no longer connected) are displayed in a grey font.
For further information about ELLSI see ELLSI software-manual: *ELLSI_Software_Manual.pdf*.

3.2.3.5 Alarms and Events

For this window click the menu item *Events.*

This page shows the alarms and events from the start up of the EtherCAN/2. The events are classified depending on the severity into the categories *Error*, *Warning* and *Info*. The list will be deleted at every reboot. During the runtime of the module the occurring events can be transmitted per e-mail to another PC (see page 22).

For correct date and time indication a time server has to be configured (see page 20). Otherwise the calculation of times will restart from 1970-01-01 at 00:00 a.m. with every reboot (Power Up) of the EtherCAN/2 module.

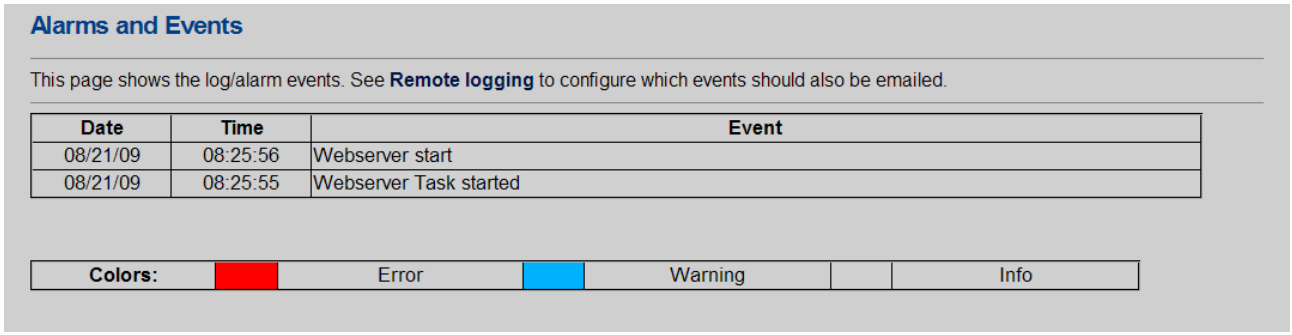


Figure 32: *Alarms and Events*

4 LEDs

4.1 Position of the LEDs

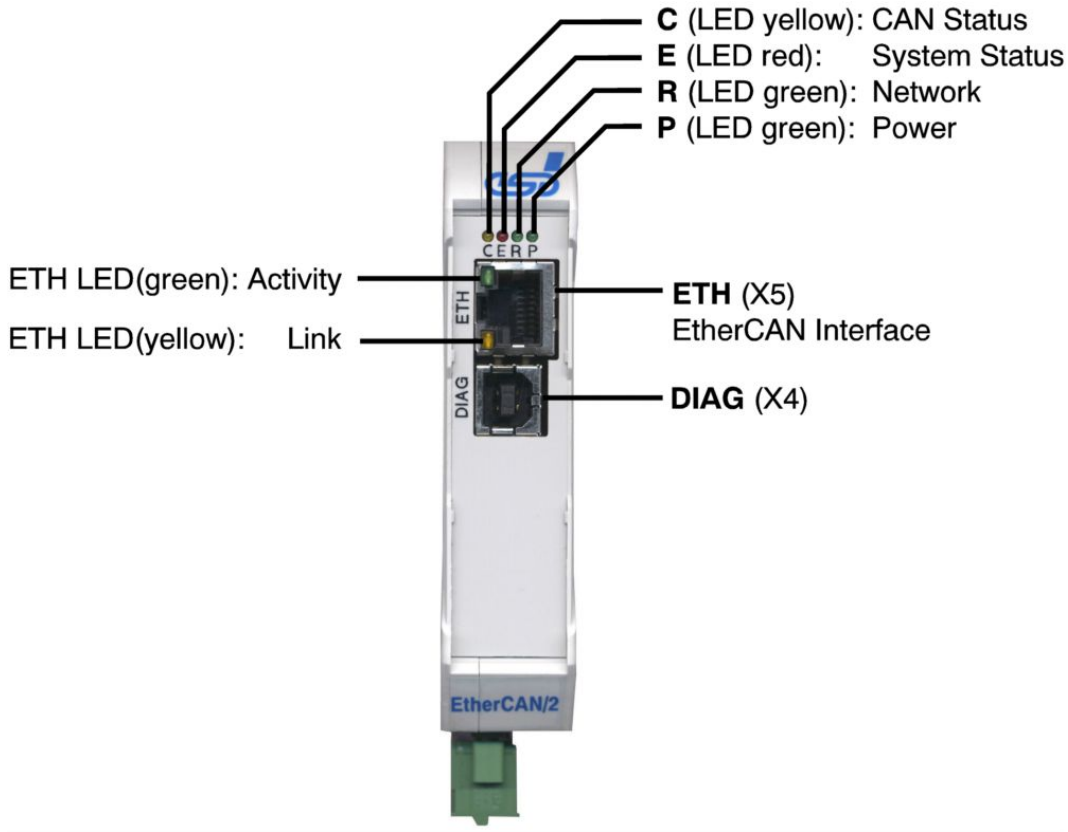


Figure 33: Position of the LEDs in the front panel

4.2 LED Indication

LEDs of the EtherCAN-RJ-45 socket

LED	Colour	Indication	Description
Activity	green	off	no Ethernet activity
		blinking	Ethernet activity (reception of Ethernet data)
Link	yellow	off	no Ethernet connection
		on	Link Status Ethernet (link to server or hub)

Table 2: Display function of ETH LEDs (RJ45-socket)

LEDs C, E, R, P

LED	Colour	Function	Indication	Description	LED-name in schematic diagram
C	yellow	CAN-Status	off	Bus-OK	LED1A
			on	Bus-OFF	
			blinking short on (long off)	Bus-Warn	
			blinking long on (short off)	Bus-Passive	
E	red	System-Status	off	no error detected	LED1B
			on	system start failed	
			blinking short on (long off)	web-server failure	
			blinking long on (short off)	net configuration failure (system has set IP address automatically)	
R	green	Network	off	no EtherCAN clients	LED1C
			blinking short on (long off)	net configuration active	
			blinking long on (short off)	EtherCAN clients connected	
P	green	Power	off	no power supply voltage or hardware error	LED1D
			on	power supply voltage supplied and hardware initialised correctly.	

Table 3: Display function of LEDs

During the boot up phase all LEDs (C, E, R, P) are additionally set to “on” for 0.5 seconds.

5 Technical Data

5.1 General Technical Data

Power	nominal voltage: typical: 24 V/DC, (min.: 18 V, max.: 32 V) current consumption: (24 V, 20 °C): typical: 100 mA
Connectors	24V 24 V-power supply voltage (X1, 4-pin Phoenix Contact connector with spring-cage connection) CAN CAN bus interface (X2, 5-pin Phoenix Contact MC 1,5/5-GF-3,81) ETH Ethernet interface (X5, 8-pin RJ45-socket) InRailBus CAN bus interface and power supply voltage via InRailBus (X6, 5-pin TBUS-connector, accessories) The following interface is for manufacturing purposes: DIAG DIAG interface (X4, USB-connector type B)
Temperature range	0 °C ... 70 °C ambient temperature
Humidity	max. 90 %, non-condensing
Dimensions	width: 22 mm, height: 112 mm, depth: 113 mm
Weight	130 g

Table 4: General technical data

5.2 Microprocessor and Memory

CPU	ARM9-Processor, 200 MHz, AT91SAM9263
Data Flash	1 MB
NAND Flash	256 MB
SDRAM	32 MB

Table 5: Microprocessor and Memory

5.3 CAN Interface

Number of CAN-Interfaces	1 x CAN
CAN controller	integrated in CPU
CAN protocol	according to ISO 11898-1
Physical Interface	High-Speed physical Layer according to ISO 11898-2, Transmission rate programmable up to 1 Mbit/s
Bus termination	Terminating resistor has to be set externally, if required.
Electrical isolation	500 V (effective) between CAN potential and module-system-potential with pollution degree 1, via optocoupler and DC/DC converter
Connector	CAN, 5-pin COMBICON (X2)

Table 6: Data of the CAN interface

5.4 Ethernet Interface

Number of the interfaces	1
Bit rate	100BASE-TX, 100 Mbit/s
Connection	Twisted Pair (compatible to IEEE 802.3), 100BASE-TX,
Electrical isolation	via transformer
Connector	RJ-45-socket with integrated LEDs in the front panel (X5)

Table 7: Data of the Ethernet interface

5.5 DIAG, Serial Interface via USB Interface

Type	USB, for manufacturing purposes only
USB specification	USB 2.0 Full Speed (12 Mbit/s)
Connector	DIAG (X4), USB-connector type-B

Table 8: Data of the DIAG interface

5.6 Software

The complete local firmware is stored in the internal flash and can be updated as required. The EtherCAN/2 module can be configured by means of a web-browser.

Bootloader	U-Boot
License information	This product uses the open source-bootloader “Das U-Boot”. The U-Boot-source code is released under the terms of the GNU Public License (GPL). The complete text of the license is contained in the esd-document “3rd Party Licensor Notice” as part of the product documentation. esd provides the complete bootloader-source code on request. esd strives to restore all changes on the bootloader into the official sources. The homepage of the U-Boot project is: http://www.denx.de/wiki/U-Boot.

Operating system	Linux, Kernel 2.6.x
Adaptation	Driver and API-functions are optimized for this system
License information	This product uses the operating system “Linux”. The Linux-source code is released under the terms of the GNU Public License (GPL). The complete text of the license is contained in the esd-document “3rd Party Licensor Notice” as part of the product documentation. esd provides the complete operating system source code on request.

Device drivers for Windows and Linux are available. Drivers for other operating systems, especially real-time operating systems, are available on request. (sales@esd.eu)

The Software-Installation and the Software-Drivers are described in the following manual:

„NTCAN-API Part 1: Application Developers Manual“[2] und
„NTCAN-API Part 2: Installation Guide“[1]
esd Order No.: C.2001.21

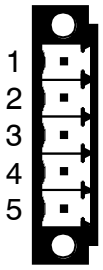
6 Interfaces and Connector Assignments

6.1 CAN

Device connector: Phoenix Contact MC 1,5/5-GF-3,81
Line connector: Phoenix Contact FK-MCP 1,5/5-STF-3,81, spring-cage connection
Phoenix Contact Order No.: 1851261 (included in delivery)
For conductor connection and conductor cross section
see page 56.

Pin Position: **Pin Assignment:**

(device connector view)

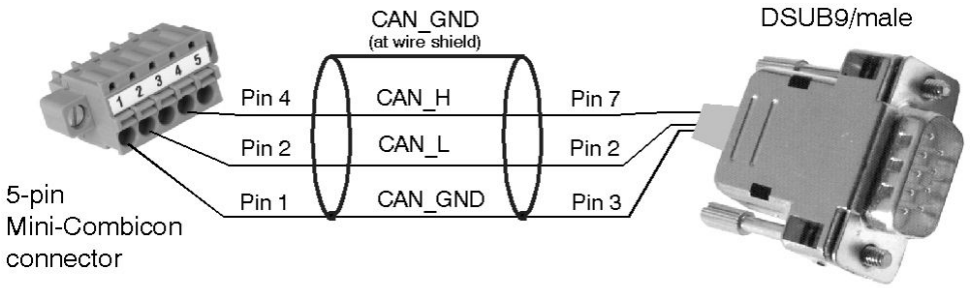


Pin	Signal
1	CAN_GND
2	CAN_L
3	Shield
4	CAN_H
5	-

Signal description:

CAN_L, CAN_H ... CAN signals
CAN_GND ... reference potential of the local CAN physical layer
Shield ... pin for line shield connection (using hat rail mounting direct contact to the mounting rail potential)
- ... not connected

Recommendation of an adapter cable from 5-pin COMBICON (here line connector FK-MCP1,5/5-STF_3,81 with spring-cage-connection) to 9-pin DSUB:



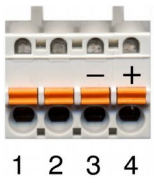
The assignment of the 9-pin DSUB-connector is designed according to CiA DS-102.

The assignment of the 5-pin Mini- COMBICON is designed according to CiA DR-303 Part 1

6.2 24 V-Power Supply Voltage

Device socket: Phoenix Contact MSTBO 2,5/4-G1L-KMGY
Line connector: Phoenix Contact FKCT 2,5/4-ST, 5.0 mm pitch,
spring cage connection,
Phoenix Contact order No.: 19 21 90 0 (included in the scope of delivery)
For conductor connection and conductor cross section see
page 56.

Pin Position:



Pin Assignment:

Pin	1	2	3	4
Labelling of the EtherCAN/2	•	•	M	P
Signal	Do not connect!	Do not connect!	M24 (GND)	P24 (+24 V)

Please refer to the connecting diagram page 9.

NOTICE
Feeding through the +24V power supply voltage can cause damage on the modules.
It is not permitted to feed through the power supply voltage through the connector X1 and to supply the power supply voltage to another CAN module station!

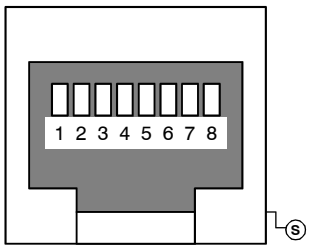
Signal description:

P24... power supply voltage +24 V ± 10 %
M24... reference potential

6.3 Ethernet 100BASE-TX (IEEE 802.3)

Device connector: RJ45 socket, 8-pin,
according to IEEE 802.3-2008, Table 25-3 'UTP MDI contact assignment'

Pin Position:



Pin Assignment:

Pin	Signal	Meaning
1	MDI0+ (TxD+)	Transmit Data +
2	MDI0- (TxD-)	Transmit Data -
3	MDI1+ (RxD+)	Receive Data +
4	-	-
5	-	-
6	MDI1- (RxD-)	Receive Data -
7	-	-
8	-	-

S	Shield	

Pin 1 to 8 are connected to a line termination.

Signal Description:

MDI0+/-, MDI0+/-,
MDI1+/-, MDI1+/- ... Ethernet data lines
- ... reserved for future applications, do not connect!
Shield... line shield connection (using hat rail mounting direct contact to the mounting rail potential)

NOTICE
Permissible cables: To ensure function in networks with up to 100 MBit/s cables of Cat. 5e or better have to be used. To ensure the EC Conformity cables with shielding SF/UTP or better have to be used.

6.4 Conductor Connection/Conductor Cross Sections

The following table contains an extract of the technical data of the cable plugs.

Characteristics	Connector Type[3]	
	Power Supply Voltage 24 V	CAN Connector
Connector type plug component (Range of articles)	FKCT 2,5/..-ST KMGY	FK-MCP 1,5/5-STF-3,81
Connection method	spring-cage connection	screw connection
Stripping length	10 mm	9 mm
Conductor cross section solid min. / max.	0.2 / 2.5 mm²	0.14 / 1.5 mm²
Conductor cross section stranded min. / max.	0.2 / 2.5 mm²	0.14 / 1.5 mm²
Conductor cross section stranded, with ferrule without plastic sleeve min. / max.	0.25 / 2.5 mm²	0.25 / 1.5 mm²
Conductor cross section stranded, with ferrule with plastic sleeve min. / max.	0.25 / 2.5 mm²	0.25 / 0.5 mm²
Conductor cross section AWG/kcmil min. / max.	24 / 12	26 / 16
2 conductors with same cross section, without TWIN ferrules with plastic sleeve	not allowed	not allowed
2 conductors with same cross section, stranded, TWIN ferrules with plastic sleeve, min.	0.5 / 1.0 mm²	not allowed
Minimum AWG according to UL/cUL	26	28
Maximum AWG according to UL/cUL	12	16

3 Technical Data from Phoenix Contact website, printed circuit board connector, plug component

7 Correct Wiring Electrically Isolated CAN Networks

NOTICE
This chapter applies to CAN networks with bit rates up to 1 Mbit/s.
If you work with higher bit rates, as for example used for CAN FD, the information given in this chapter must be examined for applicability in each individual case.
For further information refer to the CiA® CAN FD guidelines and recommendations (https://www.can-cia.org/).

For the CAN wiring all applicable rules and regulations (EU, DIN), e.g. regarding electromagnetic compatibility, security distances, cable cross-section or material, have to be observed.

7.1 Standards concerning CAN Wiring

The flexibility in CAN network design is one of the key strengths of the various extensions and additional standards like e.g. CANopen, ARINC825, DeviceNet and NMEA2000 that have been built on the original ISO 11898-2 CAN standard. In using this flexibility comes the responsibility of good network design and balancing these tradeoffs.

Many CAN organizations and standards have scaled the use of CAN for applications outside the original ISO 11898. They have made system level tradeoffs for data rate, cable length, and parasitic loading of the bus.

However for CAN network design margin must be given for signal loss across the complete system and cabling, parasitic loadings, network imbalances, ground offsets against earth potential and signal integrity. **Therefore the practical maximum number of nodes, bus length and stub length are typically much lower.**

esd has concentrated her recommendations concerning CAN wiring on the specifications of the ISO 11898-2. Thus this wiring hints forgoes to describe the special features of the derived standards CANopen, ARINC825, DeviceNet and NMEA2000.

The consistent compliance to ISO 11898-2 offers significant advantages:

- Durable operation due to well proven design specifications
- Minimizing potential failures due to sufficient margin to physical limits
- Trouble-free maintenance during future network modifications or during fault diagnostics due to lack of exceptions

Of course reliable networks can be designed according to the specifications of CANopen, ARINC825, DeviceNet and NMEA2000, **however it must be observed that it is strictly not recommended to mix the wiring guidelines of the various specifications!**

7.2 Light Industrial Environment (*Single* Twisted Pair Cable)

7.2.1 General Rules

NOTICE
esd grants the EU Conformity of the product, if the CAN wiring is carried out with at least single shielded **single** twisted pair cables that match the requirements of ISO 11898-2. Single shielded *double* twisted pair cable wiring as described in chapter 7.3. ensures the EU Conformity as well.

The following **general rules** for CAN wiring with single shielded *single* twisted pair cable should be followed:

1	A cable type with a wave impedance of about 120 Ω ±10% with an adequate conductor cross-section (≥ 0.22 mm²) has to be used. The voltage drop over the wire has to be considered.
2	For light industrial environment use at least a two-wire CAN cable. Connect • the two twisted wires to the data signals (CAN_H, CAN_L) and • the cable shield to the reference potential (CAN_GND).
3	The reference potential CAN_GND has to be connected to the functional earth (FE) at exactly **one** point.
4	A CAN net must not branch (exception: short cable stubs) and has to be terminated with the characteristic impedance of the line (generally 120 Ω ±10%) at both ends (between the signals CAN_L and CAN_H and **not** at CAN_GND).
5	Keep cable stubs as short as possible (l < 0.3 m).
6	Select a working combination of bit rate and cable length.
7	Keep away cables from disturbing sources. If this cannot be avoided, double shielded wires are recommended.

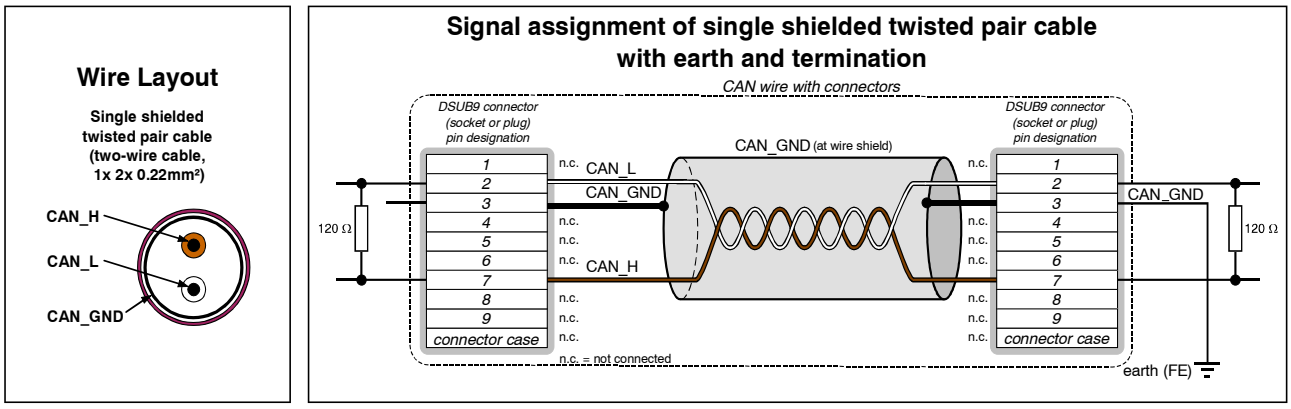


Figure 34: CAN wiring for light industrial environment

7.2.2 Cabling

- To connect CAN devices with just one CAN connector per net use a short stub (< 0.3 m) and a T-connector (available as accessory). If this devices are located at the end of the CAN network, the CAN terminator “CAN-Termination-DSUB9” can be used.

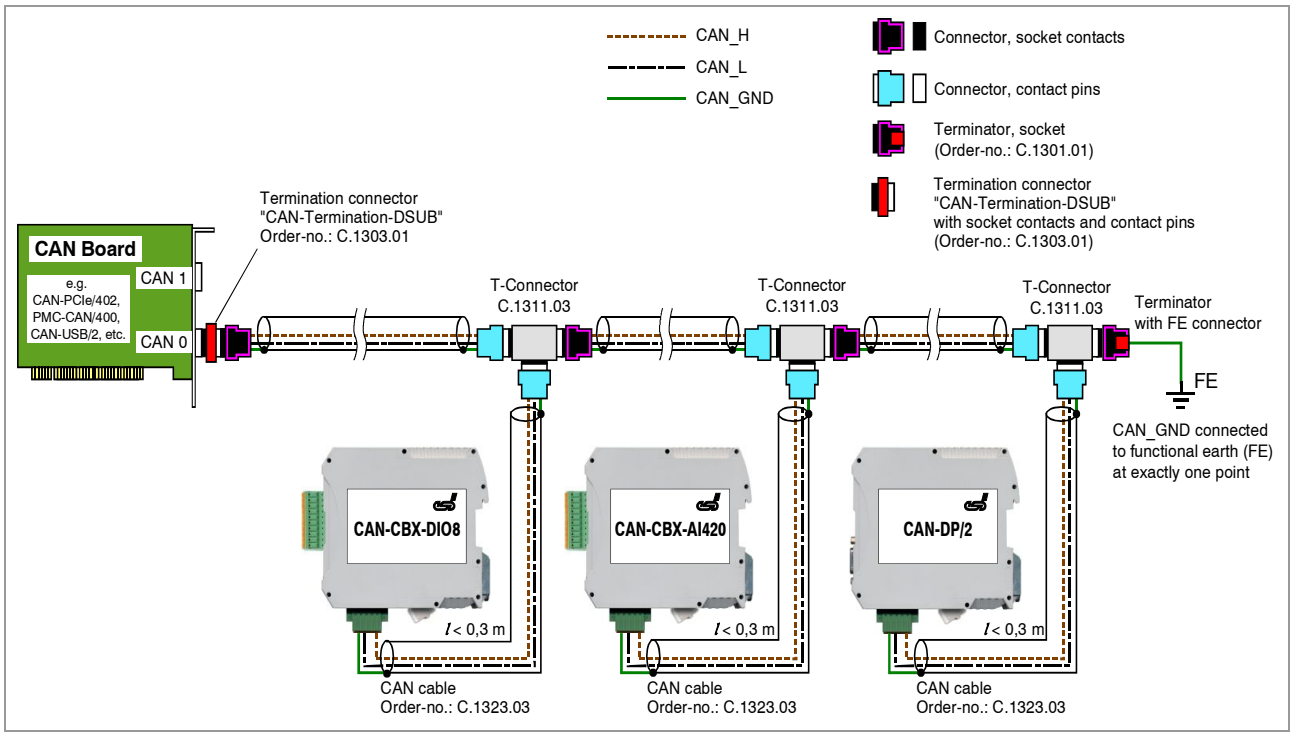


Figure 35: Example for proper wiring with single shielded single twisted pair wires

7.2.3 Branching

- In principle the CAN bus has to be realized in a line. The participants are connected to the main CAN bus line via short cable stubs. This is normally realised by so called T-connectors. esd offers the CAN-T-Connector (Order No.: C.1311.03)
- If a mixed application of single twisted and double twisted cables is unavoidable, take care that the CAN_GND line is not interrupted!
- Deviations from the bus structure can be realized by the usage of repeaters.

7.2.4 Termination

- A termination resistor has to be connected at both ends of the CAN bus.
  If an integrated CAN termination resistor which is equipped at the CAN interface at the end of the bus is connected, this one has to be used for termination instead of an external CAN termination plug.
- 9-pin DSUB-termination connectors with integrated termination resistor and pin contacts and socket contacts are available from esd (order no. C.1303.01).
- DSUB termination connectors with pin contacts (order no. C.1302.01) or socket contacts (order no. C.1301.01) and additional functional earth contact are available, if CAN termination and grounding of CAN_GND is required.

7.3 Heavy Industrial Environment (*Double* Twisted Pair Cable)

7.3.1 General Rules

The following **general rules** for the CAN wiring with single shielded *double* twisted pair cable should be followed:

1	A cable type with a wave impedance of about 120 Ω ±10% with an adequate conductor cross-section (≥ 0.22 mm²) has to be used. The voltage drop over the wire has to be considered.
2	For heavy industrial environment use a four-wire CAN cable. Connect • two twisted wires to the data signals (CAN_H, CAN_L) and • the other two twisted wires to the reference potential (CAN_GND) and • the cable shield to functional earth (FE) at least at one point.
3	The reference potential CAN_GND has to be connected to the functional earth (FE) at exactly **one** point.
4	A CAN bus line must not branch (exception: short cable stubs) and has to be terminated with the characteristic impedance of the line (generally 120 Ω ±10%) at both ends (between the signals CAN_L and CAN_H and **not** to CAN_GND).
5	Keep cable stubs as short as possible (l < 0.3 m).
6	Select a working combination of bit rate and cable length.
7	Keep away CAN cables from disturbing sources. If this can not be avoided, double shielded cables are recommended.

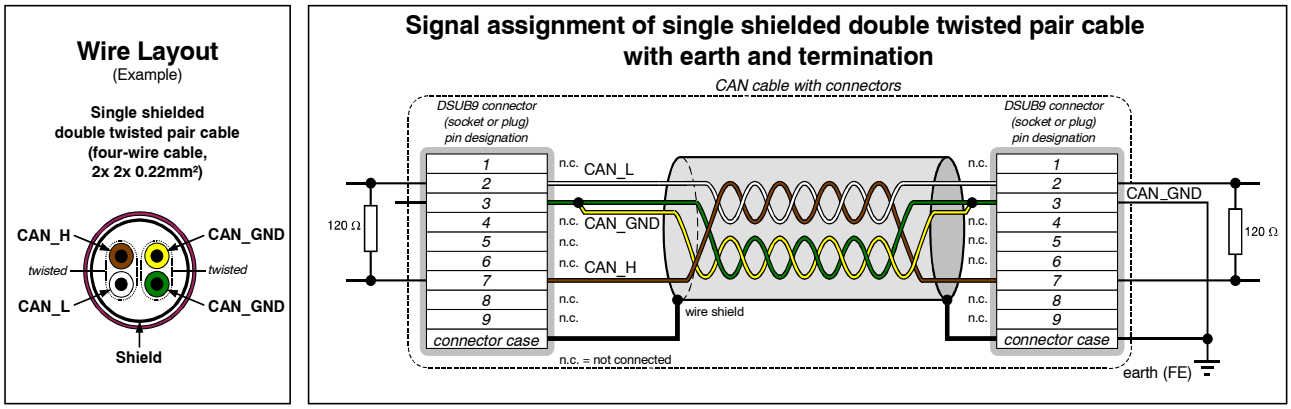


Figure 36: CAN wiring for heavy industrial environment

7.3.2 Device Cabling

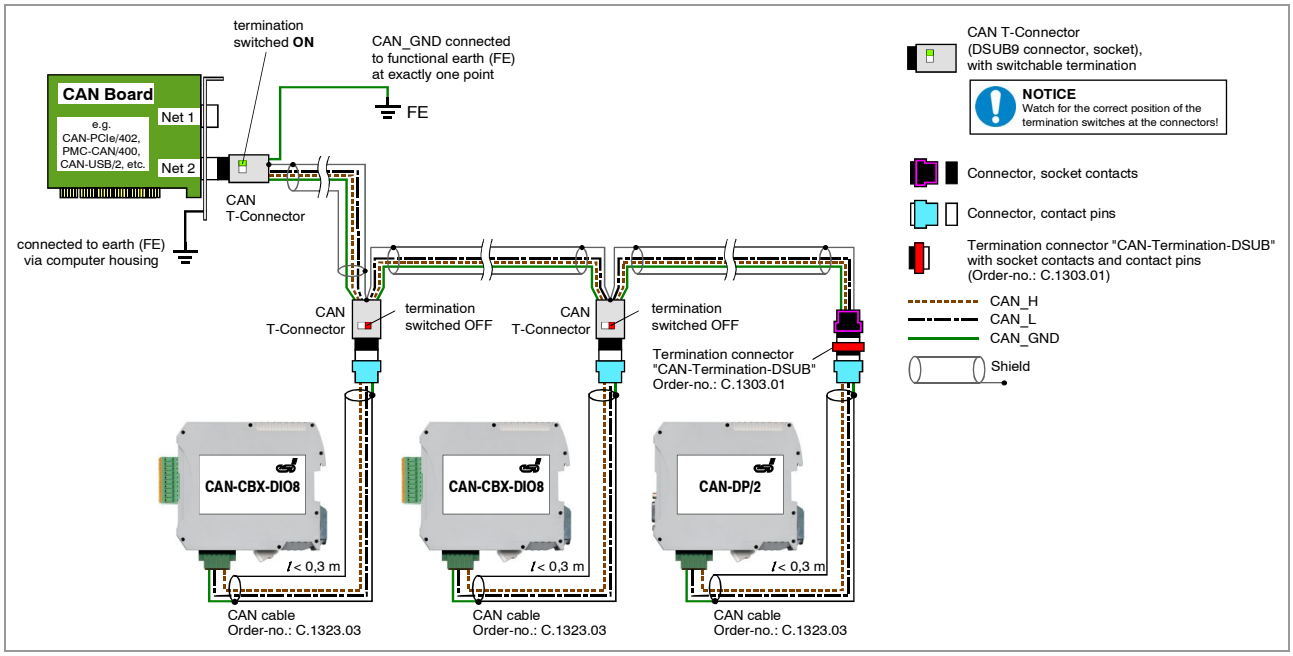


Figure 37: Example of proper wiring with single shielded double twisted pair cables

7.3.3 Branching

- In principle the CAN bus has to be realized in a line. The participants are connected to the main CAN bus line via short cable stubs. This is normally realised by so called T-connectors. When using esd's CAN-T-Connector (order no.: C.1311.03) it should be noted that the shield potential of the conductive DSUB housing is not looped through this T-Connector type. Thus the shielding is interrupted. Therefore you have to take adequate measures to connect the shield potentials, as described in the manual of the CAN-T-Connector. For further information on this read the CAN-T-Connector Manual (order no.: C.1311.21).
  Alternatively a T-connector can be used, in which the shield potential is looped through, e.g. the DSUB9 connector from ERNI (ERBIC CAN BUS MAX, order no.:154039).
- If a mixed application of single twisted and double twisted cables is unavoidable, take care that the CAN_GND line is not interrupted!
- Deviations from the bus structure can be realized by the usage of repeaters.

7.3.4 Termination

- A termination resistor has to be connected at both ends of the CAN bus.
  If an integrated CAN termination resistor which is equipped at the CAN interface at the end of the bus is connected, this one has to be used for termination instead of an external CAN termination plug.
- 9-pin DSUB-termination connectors with integrated termination resistor and pin contacts and socket contacts are available from esd (order no. C.1303.01).
- 9-pin DSUB-connectors with integrated switchable termination resistor can be ordered e.g. from ERNI (ERBIC CAN BUS MAX, socket contacts, order no.:154039).

7.4 Electrical Grounding

- For CAN devices with electrical isolation the CAN_GND must be connected between the CAN devices.
- CAN_GND should be connected to the earth potential (FE) at **exactly one** point of the network.
- Each *CAN interface with electrical connection to earth potential* acts as a grounding point. For this reason it is recommended not to connect more than one *CAN device with electrical connection to earth potential.*
- Grounding can be made e.g. at a termination connector (e.g. order no. C.1302.01 or C.1301.01).

7.5 Bus Length

NOTICE
Please note that the cables, connectors and termination resistors used in CANopen networks shall meet the requirements defined in ISO11898-2.
In addition, further recommendations of the CiA, like standard values of the cross section, depending on the cable length, are described in the CiA recommendation CiA 303-1 (see CiA 303 CANopen Recommendation - Part 1: „Cabling and connector pin assignment“, Version 1.8.0, Table 2).

Bit-Rate [kbit/s]	Theoretical values of reachable wire length **with esd interface** l_{max} [m]	**CiA recommendations** (07/95) for reachable wire lengths l_{min} [m]	Standard values of the cross-section according to CiA 303-1 [mm²]
1000	37	25	0,25 to 0,34
800 666,6 500 333,3 250	59 80 130 180 270	50 - 100 - 250	0,34 to 0,6
166 125	420 570	- 500	0,5 to 0,6
100 83,3 66,6 50	710 850 1000 1400	650 - - 1000	0,75 to 0,8
33,3 20 12,5 10	2000 3600 5400 7300	- 2500 - 5000	not defined in CiA 303-1

Table 9: Recommended cable lengths at typical bit rates (with esd-CAN interfaces)

- Optical couplers are delaying the CAN signals. esd modules typically reach a wire length of 37 m at 1 Mbit/s within a proper terminated CAN network without impedance disturbances like e.g. caused by cable stubs > 0.3 m.

7.6 Examples for CAN Cables

esd recommends the following two-wire and four-wire cable types for CAN network design. These cable types are used by esd for ready-made CAN cables, too.

7.6.1 Cable for light industrial Environment Applications (Two-Wire)

Manufacturer	Cable Type
U.I. LAPP GmbH Schulze-Delitzsch-Straße 25 70565 Stuttgart Germany www.lappkabel.com	e.g. UNITRONIC ®-BUS CAN UL/CSA (1x 2x 0.22) (UL/CSA approved) Part No.: 2170260 UNITRONIC ®-BUS-FD P CAN UL/CSA (1x 2x 0.25) (UL/CSA approved) Part No.: 2170272
ConCab GmbH Äußerer Eichwald 74535 Mainhardt Germany www.concab.de	e. g. BUS-PVC-C (1x 2x 0.22 mm²) Order No.: 93 022 016 (UL appr.) BUS-Schleppflex-PUR-C (1x 2x 0.25 mm²) Order No.: 94 025 016 (UL appr.)

7.6.2 Cable for heavy industrial Environment Applications (Four-Wire)

Manufacturer	Cable Type
U.I. LAPP GmbH Schulze-Delitzsch-Straße 25 70565 Stuttgart Germany www.lappkabel.com	e.g. UNITRONIC ®-BUS CAN UL/CSA (2x 2x 0.22) (UL/CSA approved) Part No.: 2170261 UNITRONIC ®-BUS-FD P CAN UL/CSA (2x 2x 0.25) (UL/CSA approved) Part No.: 2170273
ConCab GmbH Äußerer Eichwald 74535 Mainhardt Germany www.concab.de	e. g. BUS-PVC-C (2x 2x 0.22 mm²) Order No.: 93 022 026 (UL appr.) BUS-Schleppflex-PUR-C (2x 2x 0.25 mm²) Order No.: 94 025 026 (UL appr.)

INFORMATION
Ready-made CAN cables with standard or custom length can be ordered from **esd**.

8 CAN Troubleshooting Guide

The CAN Troubleshooting Guide is a guide to find and eliminate the most frequent hardware-error causes in the wiring of CAN networks.

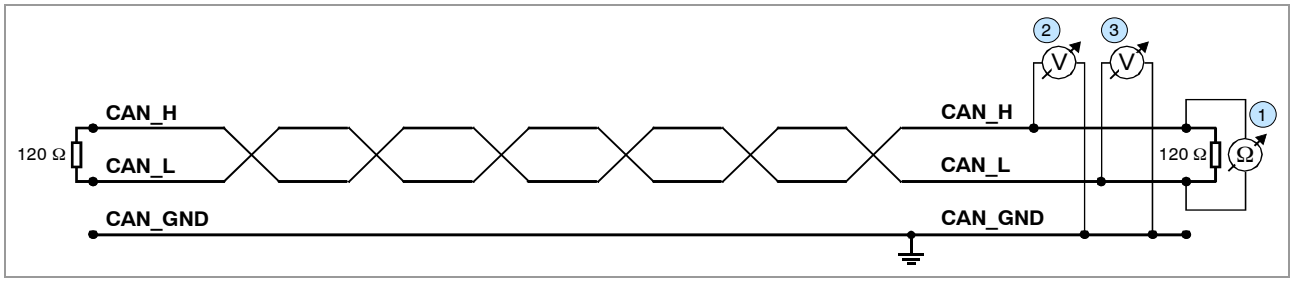


Figure 38: Simplified diagram of a CAN network

8.1 Termination

The termination is used to match impedance of a node to the impedance of the transmission line being used. When impedance is mismatched, the transmitted signal is not completely absorbed by the load and a portion is reflected back into the transmission line. If the source, transmission line and load impedance are equal these reflections are avoided. This test measures the series resistance of the CAN data pair conductors and the attached terminating resistors.

To test it ,please

1. Turn off all power supplies of the attached CAN nodes.
2. Measure the DC resistance between CAN_H and CAN_L at one end of the network (1) (see figure above).

The measured value should be between 50 Ω and 70 Ω.

If the value is below 50 Ω, please make sure that:
- there is no **short circuit** between CAN_H and CAN_L wiring
- there are **not more than two** terminating resistors connected
- the nodes do not have faulty transceivers.

If the value is higher than 70 Ω, please make sure that:
- there are no open circuits in CAN_H or CAN_L wiring
- your bus system has two terminating resistors (one at each end) and that they are 120 Ω each.

8.2 Electrical Grounding

The CAN_GND of the CAN network should be connected to the functional earth potential (FE) at only **one** point. This test will check if the CAN_GND is grounded in several places.
To test it, please

1. Disconnect the CAN_GND from the earth potential (FE).
2. Measure the DC resistance between CAN_GND and earth potential (see figure on the right).
3. Reconnect CAN_GND to earth potential.

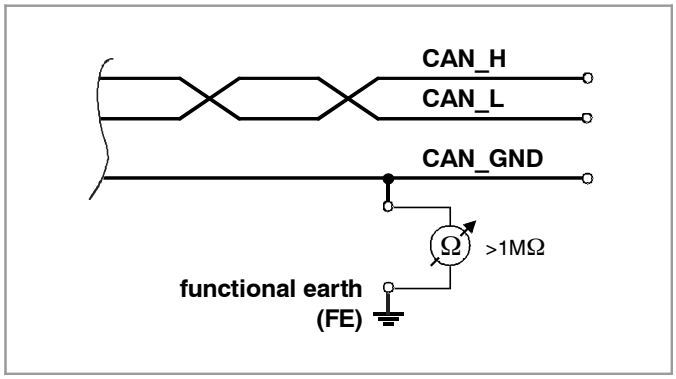


Figure 39: Simplified schematic diagram of ground test measurement

The measured resistance should be higher than 1 MΩ. If it is lower, please search for additional grounding of the CAN_GND wires.

8.3 Short Circuit in CAN Wiring

A CAN bus might possibly still be able to transmit data if there is a short circuit between CAN_GND and CAN_L, but generally the error rate will increase strongly. Make sure that there is no short circuit between CAN_GND and CAN_L!

8.4 CAN_H/CAN_L-Voltage

Each node contains a CAN transceiver that outputs differential signals. When the network communication is idle the CAN_H and CAN_L voltages are approximately 2.5 V measured to CAN_GND. Faulty transceivers can cause the idle voltages to vary and disrupt network communication.

To test for faulty transceivers, please

1. Turn on all supplies.
2. Stop all network communication.
3. Measure the DC voltage between CAN_H and CAN_GND (2) (see figure at previous page).
4. Measure the DC voltage between CAN_L and CAN_GND (3) (see figure at previous page).

Normally the voltage should be between 2.0 V and 3.0 V.

If it is lower than 2.0 V or higher than 3.0 V, it is possible that one or more nodes have faulty transceivers. For a voltage lower than 2.0 V please check CAN_H and CAN_L conductors for continuity.

To find the node with a faulty transceiver within a network please test the CAN transceiver resistance (see below) of the nodes.

8.5 CAN Transceiver Resistance Test

CAN transceivers have circuits that control CAN_H and CAN_L. Experience has shown that electrical damage of the circuits may increase the leakage current in these circuits.

To measure the current leakage through the CAN circuits, please use a resistance measuring device and:

1. Switch **off** the node and **disconnect** it from the network (4) (see figure below).

2. Measure the DC resistance between CAN_H and CAN_GND (5) (see figure below).

3. Measure the DC resistance between CAN_L and CAN_GND (6) (see figure below).

The measured resistance has to be about 500 kΩ for each signal. If it is much lower, the CAN transceiver it is probably faulty.
Another indication for a faulty transceiver is a very high deviation between the two measured input resistances (>> 200 %).

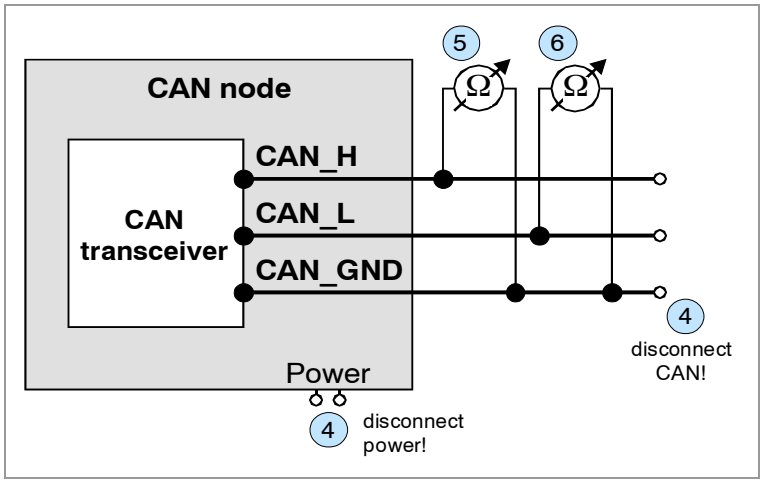


Figure 40: Measuring the internal resistance of CAN transceivers

8.6 Support by esd

If you have executed the fault diagnostic steps of this troubleshooting guide and you even can not find a solution for your problem our support department will be able to assist.
Please contact our support via email at **support@esd.eu** or by phone **+49-511-37298-130**.

9 Appendix InRailBus (Option)

9.1 Order Information InRailBus Accessories

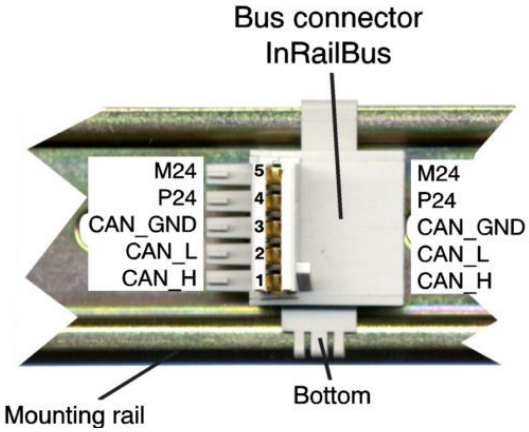
Type	Properties	Order No.
Accessories		
 **CAN-CBX-TBUS**	Mounting-rail bus connector of the CBX-InRailBus for CAN-CBX modules (order separately)	C.3000.01
 **CAN-CBX-TBUS-Connector**	Terminal plug of the CBX-InRailBus for the connection of the +24V power supply voltage and the CAN interface Female type	C.3000.02
 **CAN-CBX-TBUS-Connection adapter**	Terminal plug of the CBX-InRailBus for the connection of the +24V power supply voltage and the CAN-Interface Male type	C.3000.03

Table 10: Order Information

9.2 Connector Assignment 24V and CAN via InRailBus (Option)

Connector type: Mounting-rail bus connector of the CBX-InRailBus
Phoenix Contact ME 22,5 TBUS 1,5/5-ST-3,81 KMGY

Connector View:



Pin Assignment:

Pin	Signal
5	M24 (GND)
4	P24 (+24 V)
3	CAN_GND
2	CAN_L
1	CAN_H
S	FE (PE_GND)

Signal Description:

CAN_L,
CAN_H ... CAN signals
CAN_GND ... reference potential of the local CAN-Physical layers
P24... power supply voltage +24 V
M24... reference potential
FE... functional earth contact (EMC) (connected to mounting rail potential)

9.3 Using InRailBus (Option)

INFORMATION
This chapter describes the installation of the module using InRailBus for CAN-CBX-modules. For the EtherCAN/2 module the following points apply accordingly .

9.3.1 Installation of the Module Using InRailBus Connector

If the CAN bus signals and the power supply voltage shall be fed via the InRailBus, please proceed as follows:

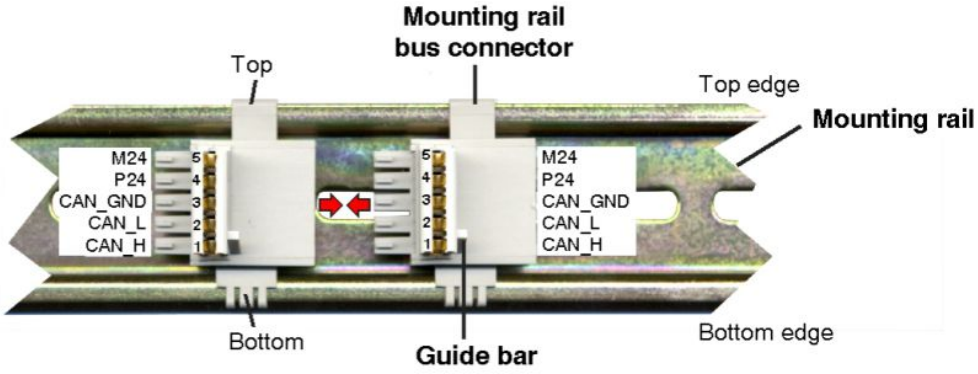


Figure 41: Mounting rail with bus connector

1. Position the InRailBus connector on the mounting rail and snap it onto the mounting rail using slight pressure. Plug the bus connectors together to contact the communication and power signals (in parallel with one). The bus connectors can be plugged together before or after mounting the CAN-CBX modules.

2. Place the CAN-CBX module with the DIN rail guideway on the top edge of the mounting rail.

Figure 42: Mounting CAN-CBX modules

3. Swivel the CAN-CBX module onto the mounting rail in pressing the module downwards according to the arrow as shown in figure 42. The housing is mechanically guided by the DIN rail bus connector.

4. When mounting the CAN-CBX module the metal foot catch snaps on the bottom edge of the mounting rail. Now the module is mounted on the mounting rail and connected to the InRailBus via the bus connector. Connect the bus connectors and the InRailBus, if not already done.

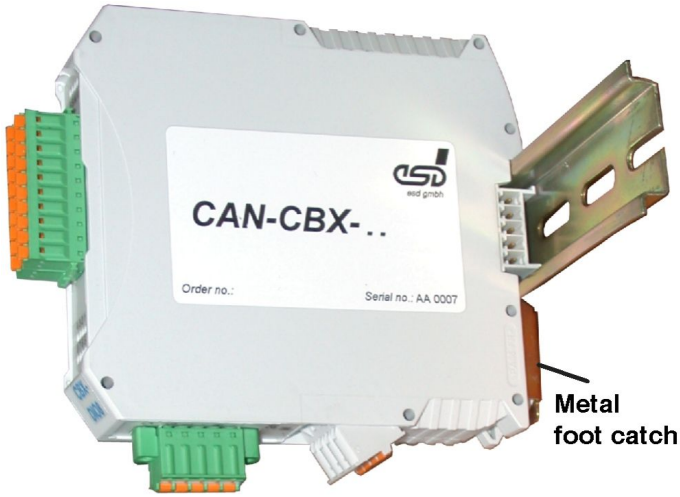


Figure 43: Mounted CAN-CBX module

9.3.2 Connecting Power Supply and CAN Signals to CBX-InRailBus

To connect the power supply and the CAN-signals via the InRailBus, a terminal plug is needed. The terminal plug is not included in delivery and must be ordered separately (order no.: C.3000.02, see order information for InRailBus Accessories, page 67).

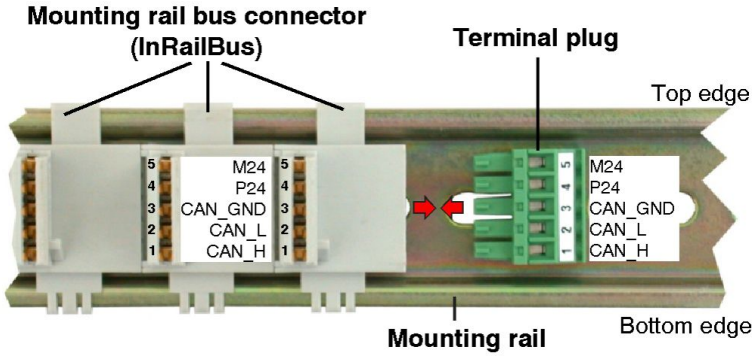


Figure 44: Mounting rail with InRailBus and terminal plug

Plug the terminal plug into the socket on the right of the mounting-rail bus connector of the InRailBus, as described in Figure 44. Then connect the CAN interface and the power supply voltage via the terminal plug.

9.3.3 Connection of the Power Supply Voltage

NOTICE
It is **not permissible** to feed through the power supply voltage through the CBX station and to supply it to another CBX station via 24V connector! A feed through of the +24 V power supply voltage can cause damage on the CBX modules.

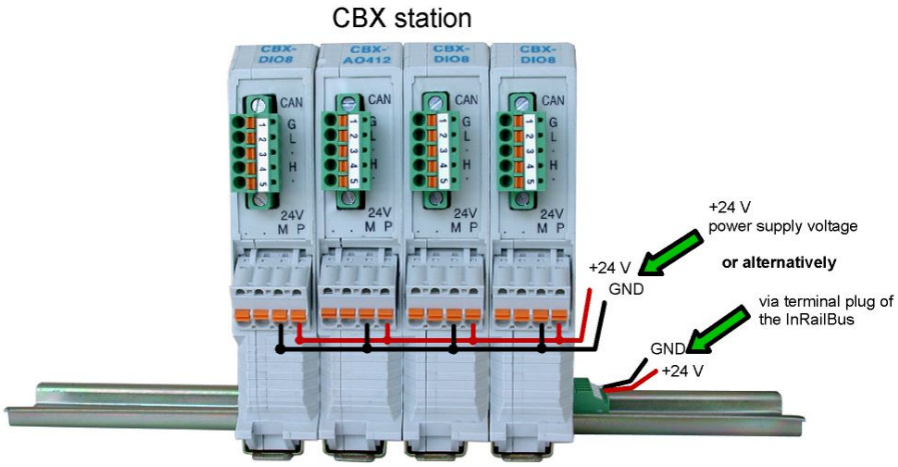


Figure 45: Connecting the power supply voltage to the CAN-CBX station

9.3.4 Connection of CAN

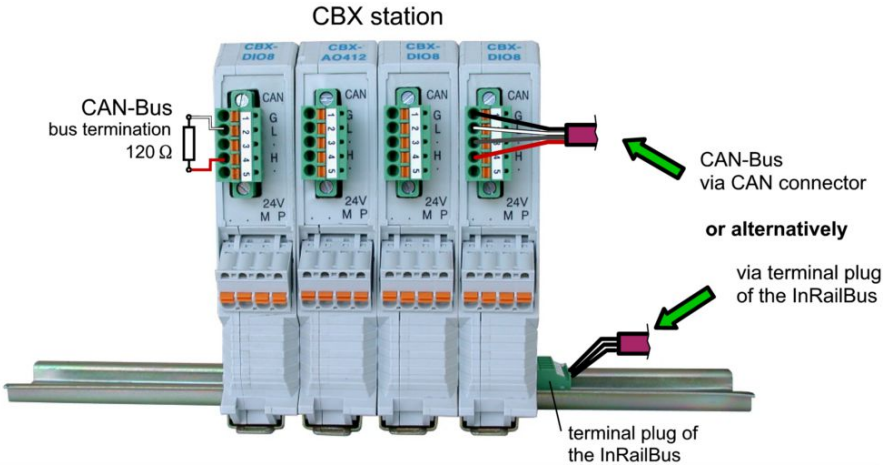


Figure 46: Connecting the CAN signals to the CAN-CBX station

Generally the CAN signals can be fed via the CAN connector of the first CAN-CBX module of the CBX station. The signals are then connected through the CAN-CBX station via the InRailBus. To lead through the CAN signals the CAN bus connector of the last CAN-CBX module of the CAN-CBX station has to be used. The CAN connectors of the CAN-CBX modules which are not at the ends of the CAN-CBX station must not be connected to the CAN bus, because this would cause incorrect branching.
A bus termination must be connected to the CAN connector of the CAN-CBX module at the end of the CBX-InRailBus (see Figure 46), if the CAN bus ends there.

9.4 Remove the CAN-CBX Module from InRailBus

If the CAN-CBX module is connected to the InRailBus please proceed as follows:

Release the module from the mounting rail in moving the foot catch (see Figure 43) downwards (e.g. with a screwdriver). Now the module is detached from the bottom edge of the mounting rail and can be removed.

INFORMATION
It is possible to remove individual devices from the whole without interrupting the InRailBus connection, because the contact chain will not be interrupted.

10 References

[1] esd electronic system design gmbh, Hannover;
NTCAN-API - Part 2: Installation Guide, Rev. 4.2, 2015-01-26
[2] esd electronic system design gmbh, Hannover;
NTCAN-API - Part 1: Application Developpers Manual, Rev. 4.7, 2015-10-12

11 Declaration of Conformity

EU-KONFORMITÄTSERKLÄRUNG
EU DECLARATION OF CONFORMITY

Adresse / *Address*
esd electronics gmbh
Vahrenwalder Str. 207
30165 Hannover
Germany

esd erklärt, dass das Produkt *esd declares, that the product*	Typ, Modell, Artikel-Nr. *Type, Model, Article No.*
EtherCAN/2 **EtherCAN/2-S7**	**C.2051.02** **C.2051.07**
die Anforderungen der Normen *fulfills the requirements of the standards*	**EN 61000-6-2:2005,** **EN 61000-6-4:2007+A1:2011**
gemäß folgendem Prüfbericht erfüllt. *according to test certificate.*	**H-K00-0336-09,** **H-Z01-0336-14**
Das Produkt entspricht damit der EU-Richtlinie „EMV" *Therefore the product conforms to the EU Directive 'EMC'*	**2014/30/EU**
Das Produkt entspricht den EU-Richtlinien „RoHS" *The product conforms to the EU Directives 'RoHS'*	**2011/65/EU, 2015/863/EU**

Diese Erklärung verliert ihre Gültigkeit, wenn das Produkt nicht den Herstellerunterlagen entsprechend eingesetzt und betrieben wird, oder das Produkt abweichend modifiziert wird.
This declaration loses its validity if the product is not used or run according to the manufacturer's documentation or if non-compliant modifications are made.

Name / *Name* T. Bielert
Funktion / *Title* QM-Beauftragter / *QM Representative*
Datum / *Date* Hannover, 2020-01-03

Rechtsgültige Unterschrift / *authorized signature*

I:\Texte\Doku\MANUALS\CAN\EtherCAN2\Konformität\EtherCAN2_EU-Konformitaetserklaerung_2020-01-03.odt

12 Order Information

Type	Properties	Order No.
EtherCAN/2	Ethernet-CAN-Gateway (incl. CAN-DRV-CD Windows/Linux)	C.2051.02
EtherCAN/2-S7	Ethernet-CAN-Gateway module inclusive S7-example project with function block to connect a S7-300/400 via Industrial Ethernet/UDP (incl. CAN-DRV-CD Windows/Linux)	C.2051.07
CAN-DRV-CD Windows/Linux	CAN-DRV-CD CD-ROM Windows & Linux (Incl. Hostdrivers for EtherCAN/2)	*
*Current drivers are available for download at www.esd.eu.		

Table 11: Order information

PDF Manuals
Manuals are available in English and usually in German as well. For availability of English manuals see table below.
Please download the manuals as PDF documents from our esd website www.esd.eu for free.

Manuals		Order No.
EtherCAN/2-MD	Hardware manual in German	C.2051.20
EtherCAN/2-ME	Hardware manual in English (this manual)	C.2051.21
CAN-API-ME	NTCAN API manual Part 1: Application Developers Manual NTCAN API manual Part 2: Installation Guide	C.2001.21

Table 12: Available manuals

Printed Manuals
If you need a printout of the manual additionally, please contact our sales team: sales@esd.eu for a quotation. Printed manuals may be ordered for a fee.