CANblue II

USER MANUAL

4.01.0126.20000 3.0 ENGLISH

HMS Connecting Devices™

Important User Information

Liability

Every care has been taken in the preparation of this document. Please inform HMS Industrial Networks of any inaccuracies or omissions. The data and illustrations found in this document are not binding. We, HMS Industrial Networks, reserve the right to modify our products in line with our policy of continuous product development. The information in this document is subject to change without notice and should not be considered as a commitment by HMS Industrial Networks. HMS Industrial Networks assumes no responsibility for any errors that may appear in this document.

There are many applications of this product. Those responsible for the use of this device must ensure that all the necessary steps have been taken to verify that the applications meet all performance and safety requirements including any applicable laws, regulations, codes, and standards.

HMS Industrial Networks will under no circumstances assume liability or responsibility for any problems that may arise as a result from the use of undocumented features, timing, or functional side effects found outside the documented scope of this product. The effects caused by any direct or indirect use of such aspects of the product are undefined, and may include e.g. compatibility issues and stability issues.

The examples and illustrations in this document are included solely for illustrative purposes. Because of the many variables and requirements associated with any particular implementation, HMS Industrial Networks cannot assume responsibility for actual use based on these examples and illustrations.

Intellectual Property Rights

HMS Industrial Networks has intellectual property rights relating to technology embodied in the product described in this document. These intellectual property rights may include patents and pending patent applications in the USA and other countries.

Table of Contents

Page

1 User Guide 5

1.1 Related Documents 5

1.2 Document History 5

1.3 Conventions 6

2 Safety and Dangers 7

2.1 Information on EMC 7

2.2 General Safety Notes 7

2.3 Bluetooth Connection 8

3 Scope of Delivery 8

4 Product Description and Features 9

4.1 Product Description 9

4.2 Operating Modes 9

4.3 Features 9

5 Installation 10

5.1 Installing the Software 10

5.1.1 Installing the Driver 10

5.1.2 Installing the CANblue II Software Package 10

5.2 Connectors 10

5.2.1 Power Connector 10

5.2.2 External Antenna 11

5.2.3 CAN Connector 11

5.3 Installing the Virtual COM Port 12

5.3.1 Windows XP 12

5.3.2 Windows 7, 8 and 10 16

6 Configuration 19

6.1 PC Interface with VCI driver for Windows Configuration 19

6.2 Generic Mode Configuration (PC Interface and Bridge) 19

6.2.1 Configuration Tool 19

6.2.2 Configuring an Interface 22

6.2.3 Configuring a Bridge 23

6.3 Settings in Generic Mode 24

6.3.1 Configuring the Filter 24

6.3.2 Autostart 25

6.3.3 Changing the Message Format 25

6.3.4 Setting the Transmitting Time 25

6.3.5 Reset to Factory Settings 25

6.3.6 Changing the Bluetooth Passkey 26
6.3.7 Visibility 26
6.3.8 Connection Security in Bridge Setup 26

7 Operation 27
7.1 Overview 27
7.2 Indicators 27
7.2.1 Mode LED 27
7.2.2 CAN LED 27
7.2.3 Bluetooth LED 27
7.3 Connection Behavior 27

8 Errors and Troubleshooting 28

9 PC Interface with VCI Driver Network and Device Communication 30

10 Generic Mode Network and Device Communication 30
10.1 ASCII Protocol 30
10.2 CAN Commands 31
10.2.1 Configuring the Communication Behavior 31
10.2.2 Initializing the CAN Controller 33
10.2.3 Configuring the Filter 34
10.2.4 Starting the CAN Controller 36
10.2.5 Stopping the CAN Controller 36
10.2.6 Reset the CAN Controller 36
10.3 Device Commands 37
10.3.1 Requesting Device Information 37
10.3.2 MAC Commands for Connecting Devices 38
10.3.3 MAC Commands Security 40
10.3.4 Configuring the Device 41
10.3.5 Reset the Device 43
10.4 CAN Messages in ASCII format 43
10.5 CAN Messages in Binary Format 44
10.6 Error Messages 45

11 Technical Data 46

12 Default Settings 46

13 Support/Return Hardware 47
13.1 Support 47
13.2 Return Hardware 47

14 Disposal 47

A Regulatory Compliance 49

A.1 EMC Compliance (CE) 49

A.2 FCC Compliance Statement 49

A.3 RoHs Directive 50

A.4 Japan Radio Equipment Compliance (TELEC) 50

B Disposal and recycling 50

C Measurements 51

D Configuration Examples 52

D.1 Example 1: Connecting a CAN Network With a Computer 52

D.2 Example 2: Configuring a CAN Bridge 53

D.3 Example 3: Configuring a Bridge Chain 55

This page intentionally left blank

1 User Guide

Please read the manual carefully. Make sure you fully understand the manual before using the product.

1.1 Related Documents

Document	Author
Installation Guide VCI Driver	HMS
User Manual of bus monitor in use	HMS
VCI Software Design Guides (.NET, C, C++)	HMS

1.2 Document History

Version	Date	Description
3.0	March 2017	Edited and revised in new design.

1.3 Conventions

Instructions and results are structured as follows:

► instruction 1

► instruction 2

➡ result 1

➡ result 2

Lists are structured as follows:

- item 1
- item 2

Bold typeface indicates interactive parts such as connectors and switches on the hardware, or menus and buttons in a graphical user interface.

```
This font is used to indicate program code and other
kinds of data input/output such as configuration scripts.
```

This is a cross-reference within this document: *Conventions, p. 6*

This is an external link (URL): www.hms-networks.com

Safety advice is structured as follows:

> ! Cause of the hazard!
>
> Consequences of not taking remediate action.
>
> How to avoid the hazard.

Safety signs and signalwords are used dependent on the level of the hazard.

ℹ *This is additional information which may facilitate installation and/or operation.*

> ! This instruction must be followed to avoid a risk of reduced functionality and/or damage to the equipment, or to avoid a network security risk.

> ⚠ **Caution**
> This instruction must be followed to avoid a risk of personal injury.

> ⚠ **WARNING**
> This instruction must be followed to avoid a risk of death or serious injury.

2 Safety and Dangers

> !
>
> Risk of disturbances and interferences if used with WLAN at the same time!
>
> Bluetooth and WLAN both work with the frequency of 2.4 GHz.

> ⚠
>
> **Caution**
> This equipment emits RF energy in the ISM (Industrial, Scientific, Medical) band. Make sure that all medical devices used in proximity to this device meet appropriate susceptibility specifications for this type of RF energy.

The CANblue II contains a small radio transmitter and receiver. During communication with other Bluetooth products the CANblue II receives and transmits electromagnetic fields (microwaves) in the frequency range 2.4 to 2.5 GHz. The output power of the radio transmitter is very low. The exposure to transmitted RF energy while using the device is well below the prescribed limits in all national and international RF safety standards and regulations.

2.1 Information on EMC

> !
>
> Risk of interference to radio and television if used in office or home environment!
>
> Use exclusively included accessories.
>
> Make sure shield of interface is connected with device plug and plug on other side.
>
> Use exclusively shielded cables.

2.2 General Safety Notes

- Protect product from moisture and humidity.
- Protect product from too high or too low temperature (see *Technical Data, p. 46*).
- Protect product from fire.
- Don’t throw, drop or try to bend the product.
- Don’t paint the product.
- Don’t modify or disassemble the product. Service must be carried out by HMS Industrial Networks.
- Don’t use modified products.
- Store products in dry and dust-free place.

2.3 Bluetooth Connection

Make sure the following conditions are met:

- preferably unobstructed line of sight between the antennas of the devices
- minimum distance of 50 cm between the devices (avoid interference)
- minimum distance of 10 m to WLAN recommended

Data transmission rate dependent on:

- distance between the communicating devices
- obstacles between the devices
- environment (texture of walls etc.)
- device configuration
- signal conditions

3 Scope of Delivery

Included in scope of delivery:

- CANblue II
- CAN bus monitor
- Installation Guide *VCI Driver*
- User Manual *CANblue II*
- CD with VCI driver and extended User Manual

4 Product Description and Features

4.1 Product Description

With CANblue II multiple CAN networks can be connected wirelessly using Bluetooth. Every CANblue II forwards the messages it receives from the CAN network to the Bluetooth connection. All messages received via a Bluetooth connection are transmitted to the CAN network and other existing Bluetooth connections.

The CANblue II provides an additional server. This connection can be used to configure the CANblue II. Different operation modes are supported.

4.2 Operating Modes

PC interface

- **VCI driver for Windows**
  - supported by VCI driver
  - operation with all IXXAT tools possible
  - operation with other VCI-based application programs and tools possible
- **Generic mode (ASCII/binary protocol)**
  - communication based on ASCII commands and optimized binary data transfer
  - usable in all systems, for example embedded computer systems
  - low latency

Bridge mode

- several CANblue II can be connected
- CANblue II can serve as Master and Slave
- transparent message exchange on layer 2
- can be used in DeviceNet, CANopen, J1939 and customer specific protocols
- use of CAN-ID filters possible

4.3 Features

- Bluetooth specification Bluetooth v4.0
- power supply 9 to 30 V DC
- ISO 11898-2 CAN bus coupling (9 pin Sub-D9)
- available with internal or external antenna
- different external antennas available
- CAN controller initialization with automatic baud-rate detection
- CAN message filtering

5 Installation

> **!** Connection problem after computer turned into sleep mode!
>
> Deactivate sleep mode of computer the CANblue II is connected to. In case of reconnecting problems see *Errors and Troubleshooting, p. 28*.

5.1 Installing the Software

5.1.1 Installing the Driver

For the operation of the CANblue II as VCI PC interface a driver is needed.

- ► Install VCI driver (see Installation Guide VCI Driver).

5.1.2 Installing the CANblue II Software Package

- ► Close all open applications.
- ► Make sure that all prior versions of CANblue II software package are uninstalled.
- ► Insert CD-ROM in CD drive.
- ► Run *CANblue_II_Generic_Setup.exe*.
- ► Follow instructions in installation program.

5.2 Connectors

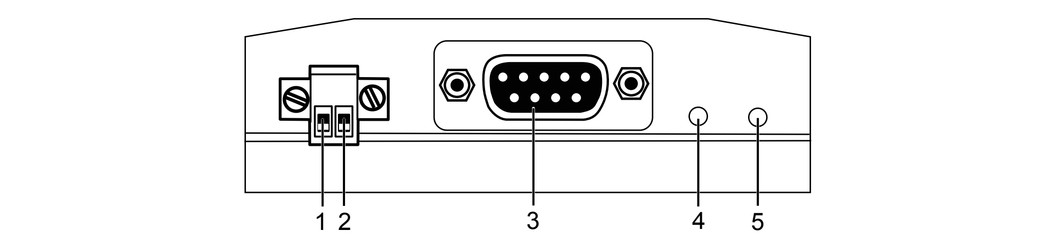


Fig. 1 Connectors

1	Power connector +
2	Power connector -
3	CAN connector
4	No function
5	Button **Reset to factory settings**

5.2.1 Power Connector

The device is protected against polarity reversal.

Pin allocation

Number	Pin designation	Signal
1	+	9 to 30 V DC
2	-	GND

5.2.2 External Antenna

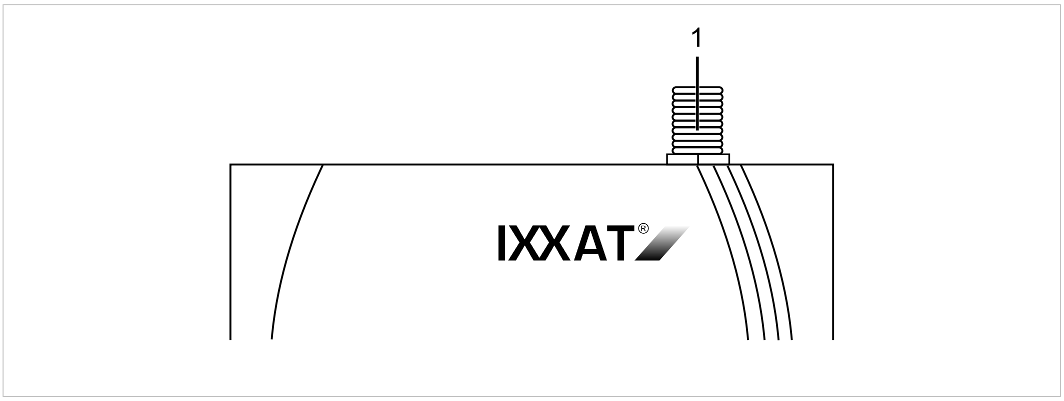


Fig. 2 Connector for external antenna

- ► Screw external antenna on connector (**1**).
- ► Use exclusively antennas approved by HMS Industrial Networks (by reason of radio certification).
- ► For further information about different antennas see www.ixxat.com.

5.2.3 CAN Connector

Pin allocation of Sub-D9 connector

Pin no.	Signal
1	-
2	CAN-Low
3	GND
4	-
5	-
6	-
7	CAN-High
8	-
9	-

5.3 Installing the Virtual COM Port

The CANblue II provides two virtual servers: Config and SPP. For the configuration of the CANblue II a Bluetooth-capable device that supports the serial port profile (SPP) must be connected to the Config server via a virtual COM port.

The COM port must have the following properties:

- baud-rate: 921600
- data bits: 8
- parity bit: none
- stop bits: 1
- flow control: hardware

5.3.1 Windows XP

► Open **Windows start menu** and select **Control Panel**.

► Open **Bluetooth devices**.

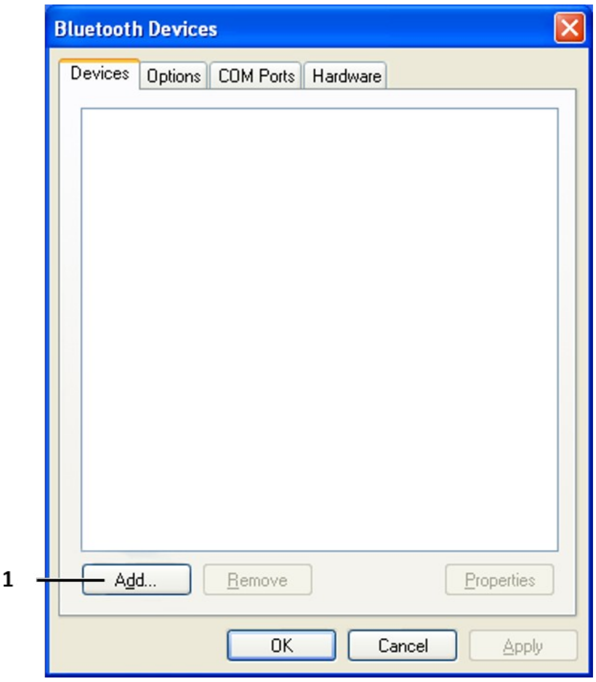
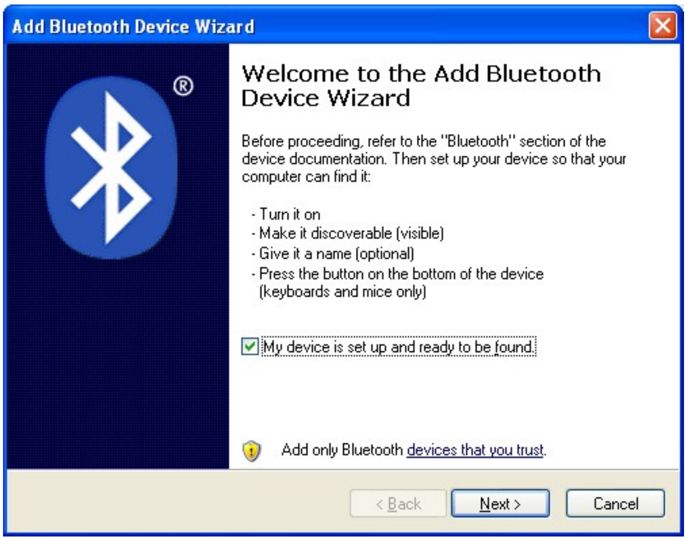


Fig. 3 Bluetooth devices

► Click button **Add** (**1**).

➡ Wizard to add Bluetooth devices is opened.



Welcome to the Add Bluetooth Device Wizard

Before proceeding, refer to the "Bluetooth" section of the device documentation. Then set up your device so that your computer can find it:

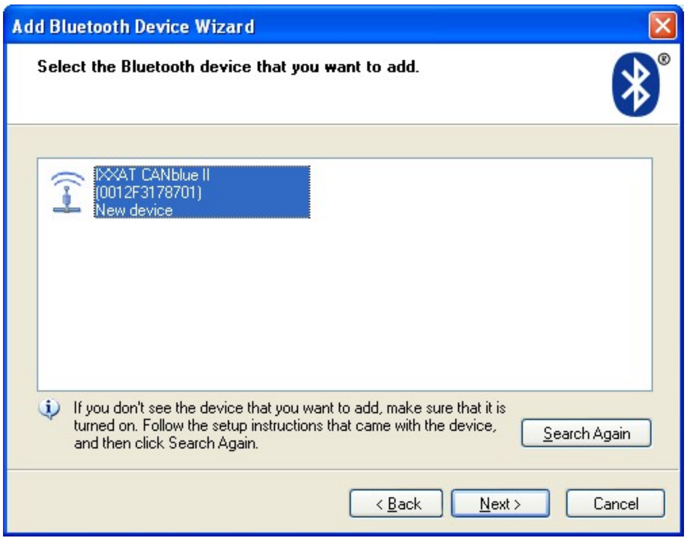
- Turn it on
- Make it discoverable (visible)
- Give it a name (optional)
- Press the button on the bottom of the device (keyboards and mice only)

☑ My device is set up and ready to be found.

Add only Bluetooth devices that you trust.

Fig. 4 Wizard to add Bluetooth device

- ► Activate checkbox **My device is set up and ready to be found.**
- ► To search for devices, click button **Next**.
  - ➡ All available devices are displayed.
  - ➡ CANblue II devices are named *IXXAT CANblue II ([MAC address])*.



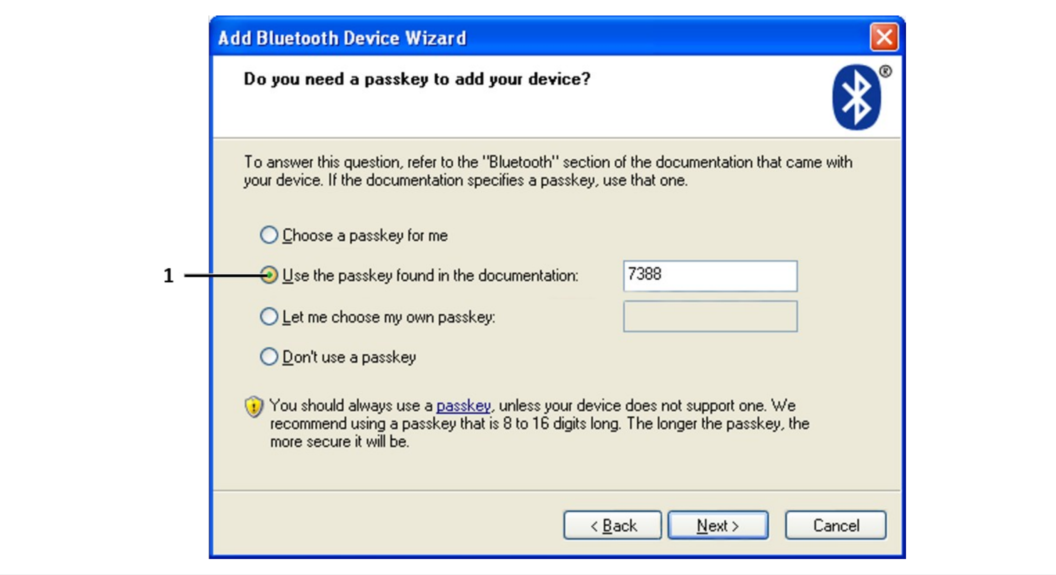
Select the Bluetooth device that you want to add.

IXXAT CANblue II
(0012F3178701)
New device

If you don't see the device that you want to add, make sure that it is turned on. Follow the setup instructions that came with the device, and then click Search Again.

Fig. 5 Found devices

- ► Check MAC address of the CANblue II printed on the back of the device.
- ► Select device to connect and click button **Next.**



Add Bluetooth Device Wizard

Do you need a passkey to add your device?

To answer this question, refer to the "Bluetooth" section of the documentation that came with your device. If the documentation specifies a passkey, use that one.

○ Choose a passkey for me

● Use the passkey found in the documentation: 7388

○ Let me choose my own passkey:

○ Don't use a passkey

You should always use a passkey, unless your device does not support one. We recommend using a passkey that is 8 to 16 digits long. The longer the passkey, the more secure it will be.

Fig. 6 Enter Passkey

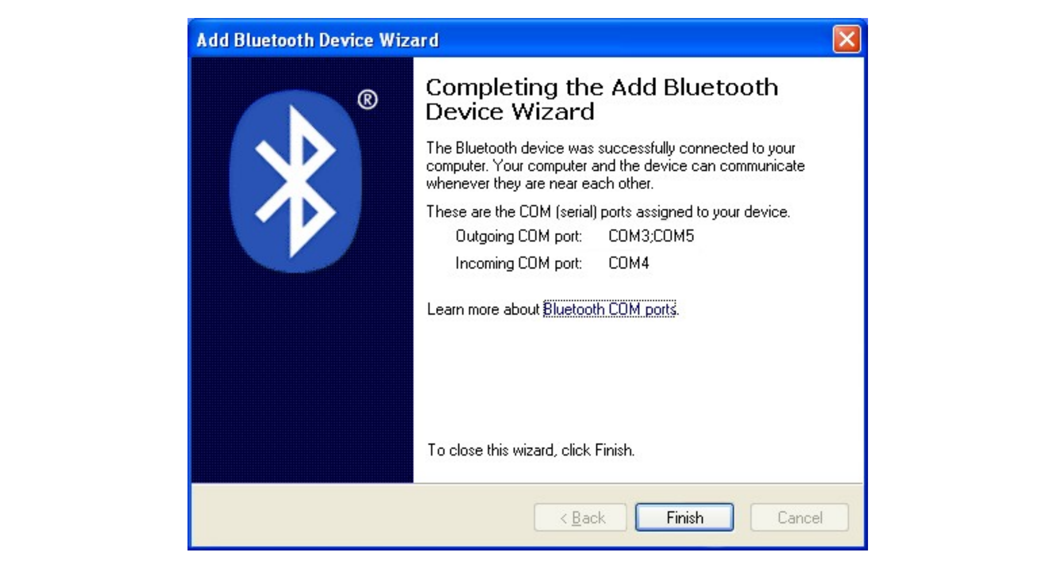
► Check **Use the passkey found in the documentation** (**1**) and enter default pairing code **7388**.

Some Bluetooth drivers do not ask for a pairing code. In this case pairing is possible without code.

► Click button **Next**.

➡ All drivers are installed.

➡ For the device created virtual COM ports are displayed.



Add Bluetooth Device Wizard

Completing the Add Bluetooth Device Wizard

The Bluetooth device was successfully connected to your computer. Your computer and the device can communicate whenever they are near each other.

These are the COM (serial) ports assigned to your device.

Outgoing COM port: COM3;COM5

Incoming COM port: COM4

Learn more about Bluetooth COM ports.

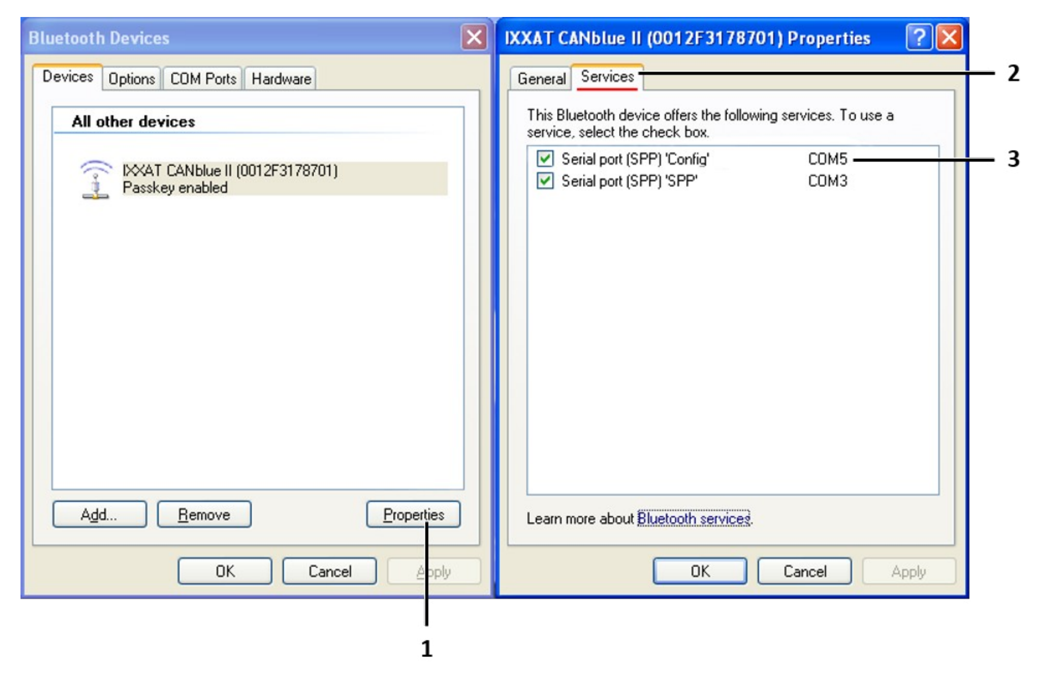
To close this wizard, click Finish.

Fig. 7 Virtual COM ports

► Click button **Finish.**

➡ Two outgoing COM ports are displayed. One COM port is provided for the Config connection of other devices than CANblue II.

Determine correct COM port:



This Bluetooth device offers the following services. To use a service, select the check box.

☑ Serial port (SPP) 'Config' COM5

☑ Serial port (SPP) 'SPP' COM3

Learn more about Bluetooth services.

Fig. 8 Properties of Bluetooth device

► In window **Bluetooth devices** select the newly added CANblue II and click button **Properties** (**1**).

➡ Window **CANblue II Properties** is opened.

► Select tab **Services** (**2**).

➡ Two SPP servers of the device are displayed.

➡ With the displayed COM port of **Serial port (SPP) 'Config'** (**3**) a connection to the CANblue II can be established.

➡ The COM port of **Serial port (SPP) 'SPP'** is reserved for a connection between two CANblue II devices.

► Make sure that checkboxes of **Serial port (SPP) 'Config'** and **Serial port (SPP) 'SPP'** are activated.

If checkboxes are not activated the driver may not be correctly installed. To download the driver, make sure an internet connection is established.

► Click button **Apply**.

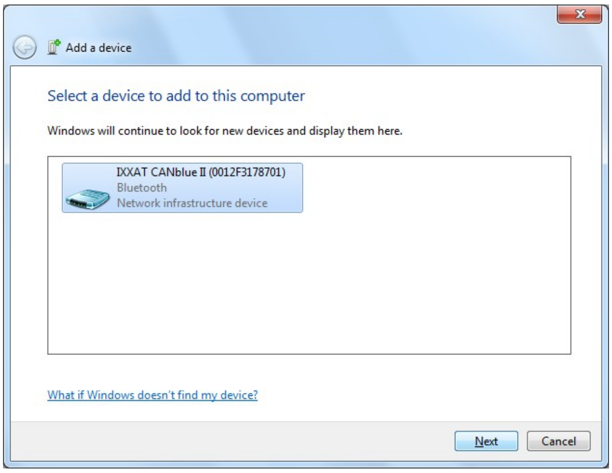
➡ The COM port of **Serial port (SPP) 'Config'** can be used to connect to the CANblue II.

5.3.2 Windows 7, 8 and 10

► In Windows task bar right-click on Bluetooth icon and select **Add a device**.

➡ All available devices are displayed.

➡ CANblue II devices are named *IXXAT CANblue II ([MAC address])*.



Add a device

Select a device to add to this computer

Windows will continue to look for new devices and display them here.

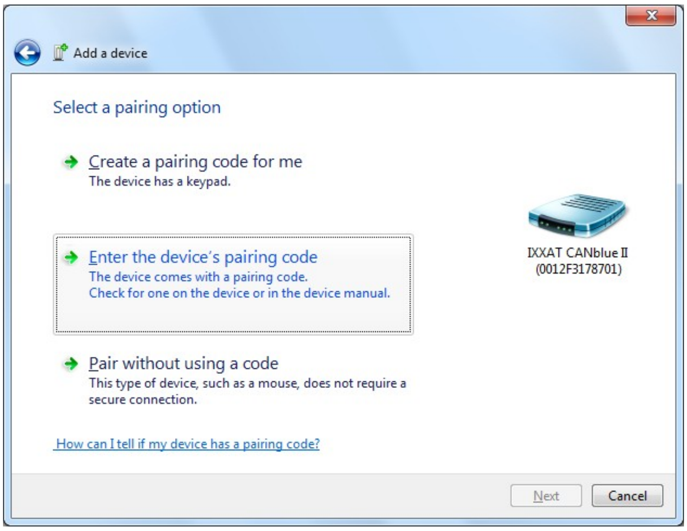
IXXAT CANblue II (0012F3178701)
Bluetooth
Network infrastructure device

What if Windows doesn't find my device?

Fig. 9 Add a device

► Check MAC address of the CANblue II printed on the back of the device.

► Select device to connect and click button **Next.**



Add a device

Select a pairing option

Create a pairing code for me
The device has a keypad.

Enter the device's pairing code
The device comes with a pairing code.
Check for one on the device or in the device manual.

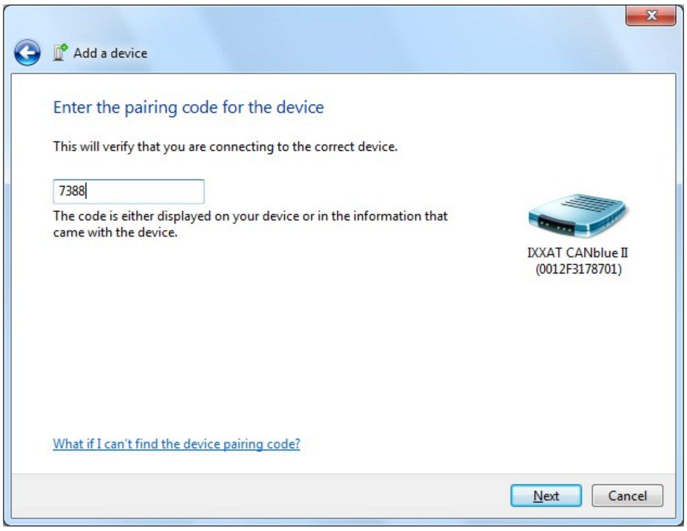
Pair without using a code
This type of device, such as a mouse, does not require a secure connection.

How can I tell if my device has a pairing code?

IXXAT CANblue II
(0012F3178701)

Fig. 10 Add a device

► Select **Enter the device's pairing code** and click button **Next.**



Add a device

Enter the pairing code for the device

This will verify that you are connecting to the correct device.

7388

The code is either displayed on your device or in the information that came with the device.

Fig. 11 Pairing code

► Enter default pairing code **7388** and click button **Next.**

➨ Added device is displayed in window **Devices and Printers**.

Some Bluetooth drivers do not ask for a pairing code. In this case pairing is possible without code.

Determine correct COM port:

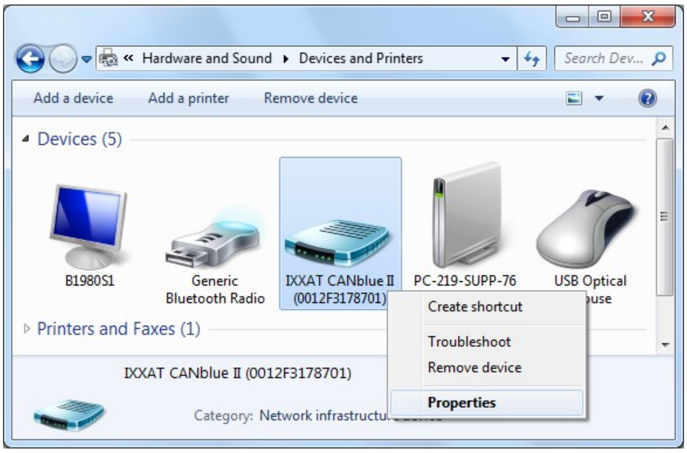
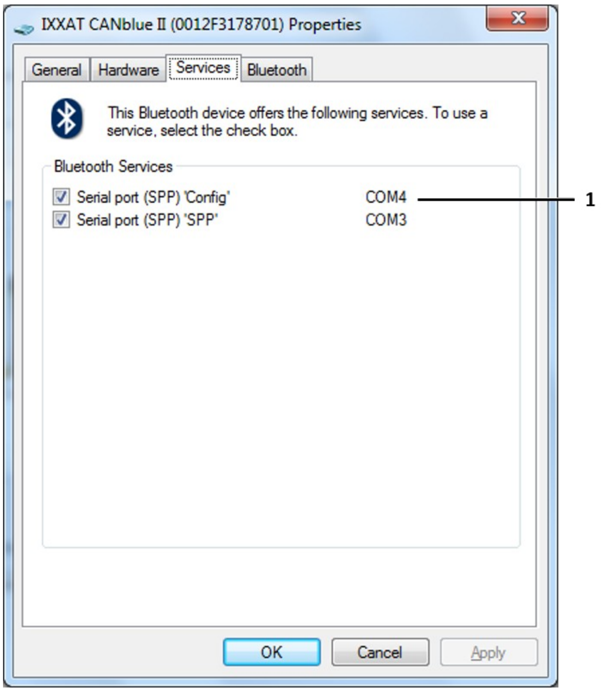


Fig. 12 Devices and printers

► In window **Devices and Printers** right-click on newly added CANblue II and in context menu select **Properties**.

➡ Window **CANblue II Properties** is opened.



IXXAT CANblue II (0012F3178701) Properties

This Bluetooth device offers the following services. To use a service, select the check box.

Bluetooth Services

☑ Serial port (SPP) 'Config' COM4

☑ Serial port (SPP) 'SPP' COM3

Fig. 13 CANblue II properties

➡ Two SPP server of the device are displayed.

➡ With the displayed COM port of **Serial port (SPP) 'Config'** (**1**) a connection to the CANblue II can be established.

➡ The COM port of **Serial port (SPP) 'SPP'** is reserved for a connection between two CANblue II devices.

► Make sure that checkbox of **Serial port (SPP) 'Config'** and **Serial port (SPP) 'SPP'** are activated.

ℹ *If checkboxes are not activated the driver may not be correctly installed. To download the driver, make sure an internet connection is established.*

► Click button **Apply**.

➡ The COM port of **Serial port (SPP) 'Config'** can be used to connect to the CANblue II

6 Configuration

6.1 PC Interface with VCI driver for Windows Configuration

ⓘ *HMS recommends to reset device to factory settings for optimal performance.*

ⓘ *Parallel usage with bridge mode is possible with reduced receive and transmit performance. Existing CAN filters are cleared in VCI mode and restored when VCI mode is closed.*

- ► Make sure that VCI driver is installed.
- ► Make sure that virtual Config COM port is installed (see *Installing the Virtual COM Port, p. 12*).
- ► Install hardware according to instructions in the *VCI Installation Guide*.
- ► With Windows 7 or higher make sure to access **Device Server Control** with administrator rights.
- ► Configure device with VCI based tool, e. g. with simple bus monitor canAnalyser included on delivery CD.
- ► To test if device is connected, check list of available devices in canAnalyser.

6.2 Generic Mode Configuration (PC Interface and Bridge)

ⓘ *Configuration examples for a generic interface and a bridge are included on the delivery CD in folder CANblueCon Examples.*

The examples can be loaded with CANblueCon Configuration Tool.

6.2.1 Configuration Tool

To configure the CANblue II the CANblueCon Configuration Tool or a terminal program can be used. With the CANblueCon Configuration tool it is possible to load configurations (txt- and bat-files).

Terminal Program

- ► Make sure that virtual Config COM port is installed.
- ► Select setting **serial** and correct COM port (see *Installing the Virtual COM Port, p. 12*).
- ► Activate local echo.
- ► Activate **transmitting of carriage return and linefeed with Enter key** at the end of an entered command.
- ► Enter virtual Config COM port.
  - ➡ Device is connected.
- ► Enter ASCII commands to configure the device (see *Generic Mode Network and Device Communication, p. 30*).
- ► Enter commands in capital letters.
- ► Execute commands with **Enter** key.

CANblueCon Configuration Tool

ⓘ *bat-files can be started directly from the file.*

Adjust COM port in bat-file with an editor and with bridge configurations adjust MAC address in txt-file.

To start bat-file in CANblueCon Configuration Tool, double-click on bat-file.

- ► Make sure, that virtual Config COM port is installed (see *Installing the Virtual COM Port, p. 12*).
- ► Start command line.
- ► Enter path to *CanBlueCon.exe*.

To load an existing configuration:

- ► With bridge configurations adjust MAC address in txt-file.
- ► Enter **CanBlueCon.exe <CONFIG_COM_PORT_NUMBER> <FILE_NAME>** in command line.
  - ➧ Batch mode is started.
  - ➧ Commands are read from the configuration file.

To define new configuration:

- ► Enter **CanBlueCon.exe <CONFIG_COM_PORT_NUMBER>**.
  - ➧ Interactive mode is started.
- ► Enter ASCII commands (see *Generic Mode Network and Device Communication, p. 30*) and additional local commands (see *Local Commands, p. 21*) to configure the device.
- ► Execute commands with **Enter** key.

ⓘ *The CANblueCon configuration tool supports a command history. Scrolling through former commands is possible with keys* ***Up*** *and* ***Down****.*

Local Commands

Additionally to the ASCII commands local commands are supported. The commands are interpreted locally and allow for example the implementation of cyclic transmission. Local commands are useful if a configuration is planned to use in Batch mode of CANblueCon, for example to implement loops or prints on screen.

Additional Commands with CANblueCon Configuration Tool

Command	Parameter	Description
#delay	<DELAY_TIME>	Delay in execution for specified time in sec.
#goto	<LABEL_NAME>	Continuing execution from string where label is defined.
#help	-	Shows a help screen.
#label	<LABEL_NAME>	Defines label.
#pause	-	Wait until any key is pressed.
#print	<TEXT>	Prints <TEXT> on the display.
#exit	-	Closes CANblueCon.

Examples

CANblue II command and CANblue II reply:

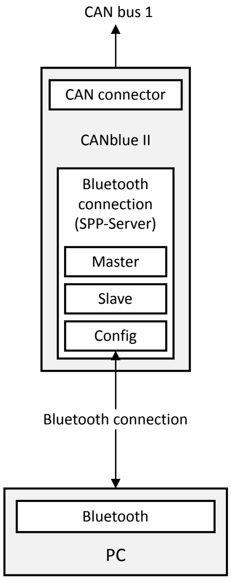
```
>c can_init 1000
I OK: CAN_INIT
```

Local command and local output:

```
>#print CANblue Generic
# CANblue Generic
```

6.2.2 Configuring an Interface

The installed virtual Config COM port is used to configure the CANblue II to exchange data with a CAN network connected to the CANblue II.



- ► Make sure virtual COM port is installed (see *Installing the Virtual COM Port, p. 12*).
- ► Reset device to factory settings with command `D SETTINGS_DEFAULT`.
  - ➡ Existing filters and settings are deleted.
  - ➡ Factory settings are set (information about settings see *Reset to Factory Settings, p. 25*).
- ► Initialize CAN controller to desired baud-rate with command `C CAN_INIT <baudrate>`.
- ► Set filter (see *Configuring the Filter, p. 24*).
- ► Specify further settings (see *Settings in Generic Mode, p. 24*).
- ► Check configuration with command `C CONFIG SHOW`.
- ► Save configuration with command `C CONFIG SAVE`.
- ► Start CAN controller with command `C CAN_START`.
  - ➡ If the CAN controller receives a message from the CAN network that matches one of the filters, the message is transmitted on the Bluetooth connection in ASCII format.
- ► To transmit CAN messages to the CANblue II or into the connected CAN network use ASCII or binary format (see *Generic Mode Network and Device Communication, p. 30*).
  - ➡ Transmission format of CAN messages is automatically matched to the received format.

Example message

- ► To transmit a CAN data frame with standard identifier 7FF and data bytes *1A 2B 3C 4D 5E 6F 70* to the CAN bus, use command `M SD7 7FF 1A 2B 3C 4D 5E 6F 70`.

6.2.3 Configuring a Bridge

Several Bluetooth devices can be connected as Master and Slave.

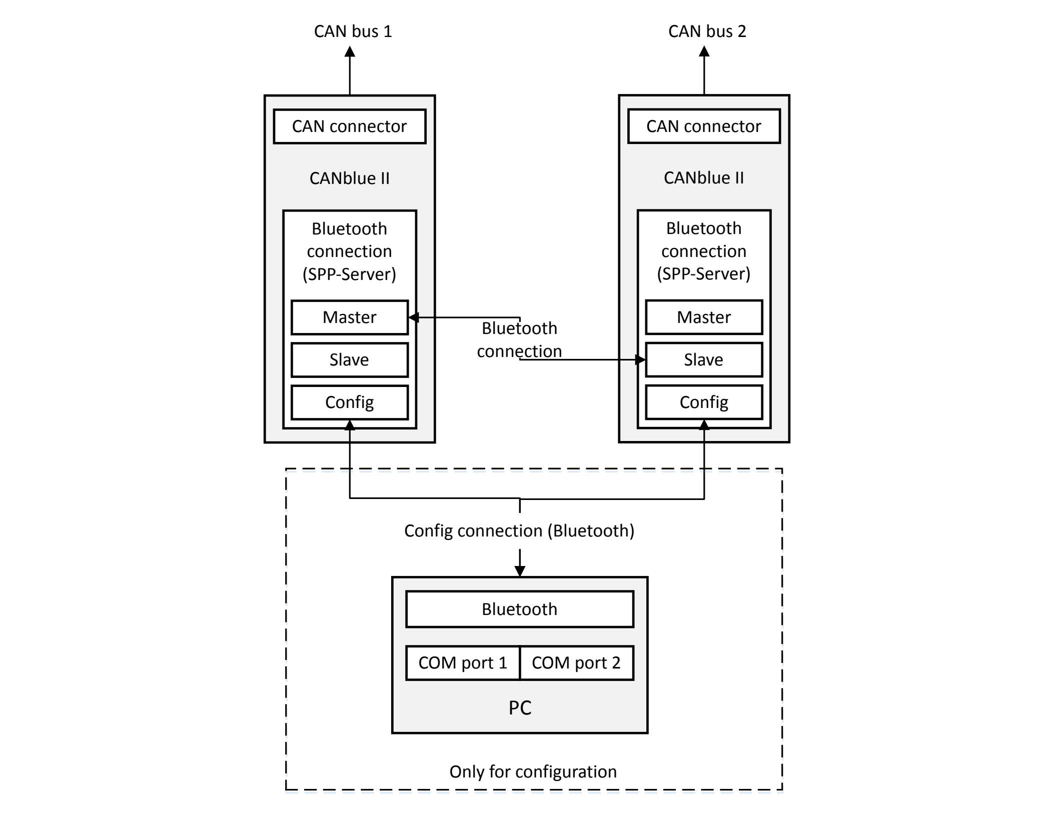


Fig. 14 Configuring a bridge

To simplify the configuration, turn off transmission of CAN messages by Master:

Stop CAN controller with command `C CAN_STOP`.

or

Disable transmission of CAN messages on the connection with command `C SEND_CAN_FRAMES OFF`.

- ► Make sure virtual COM ports for devices are installed and connection is established.
- ► Configure devices like an interface (see *Configuring an Interface, p. 22*).
- ► Enable autostart mode with command `C AUTOSTART ON` on both devices (further information see *Autostart, p. 25*).
- ► With desired Master device enter command `D MAC_ADD <address of slave>`.
  - ➨ Device acts as Master and connects to slave.
  - ➨ Devices start automatically.
  - ➨ Devices function as bridge between the two CAN networks.
- ► Save configurations with commands `C CONFIG SAVE` on both devices.
- ► To achieve highest possible data rate between the devices, disconnect Config connection from computer.

Since the connection is stored on both devices, devices reconnect automatically after turning off and on and resume forwarding of CAN messages.

Bridge Chain

Configuring a bridge chain is possible because every Slave can serve as Master for another Slave.

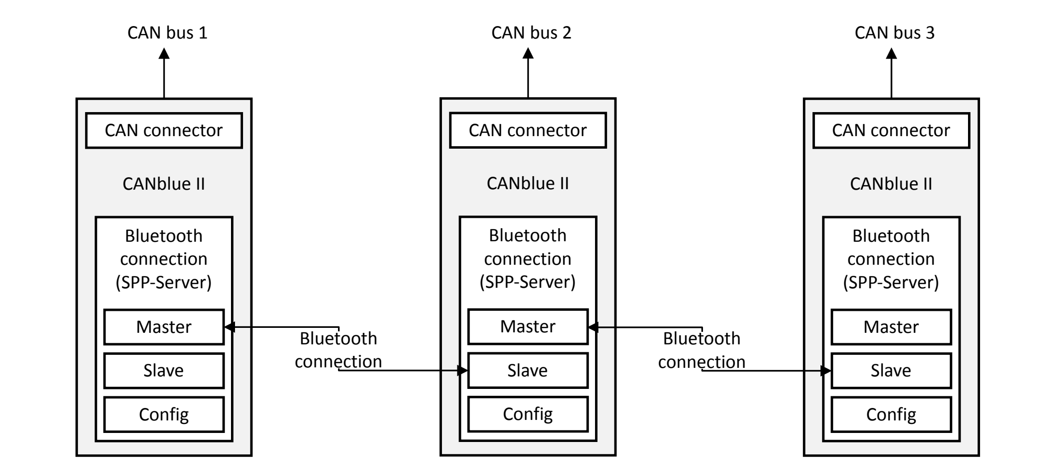


Fig. 15 Bridge configuration

Each additional CAN bus increases rate of CAN messages on the Bluetooth connections and reduces the maximum possible data rate of all connections.

6.3 Settings in Generic Mode

6.3.1 Configuring the Filter

Filtering of received messages is possible with the following criteria:

- identifier
- frame format (extended, standard)
- frame type (data, remote)

The filter works as a positive filter. CAN messages, with defined criteria in the filter list, received by the CAN controller are forwarded to the Bluetooth connection.

Up to 4096 standard filter entries (includes all possible identifiers of standard frame format) are supported.

For the extended filter 300 byte memory are provided. An extended filter entry occupies 8, 16, 24 or 32 bit, depending on the number of CAN ID digits. 75 to 300 extended messages can be filtered.

CAN ID range	Memory consumption in bytes
0–7F	1
80–7FFF	2
8000–7FFFFF	3
800000–7FFFFFFF	4

For information about the available commands to configure the filter see ASCII commands in *10.2 CAN Commands* in extended User Manual.

6.3.2 Autostart

If autostart mode of the device is enabled and Bluetooth connection is established, the device attempts to carry out a handshake to start the CAN controller.

► To enable autostart mode use command `C AUTOSTART ON`.

► To carry out a handshake between two devices, make sure that autostart mode is enabled with both devices.

If Config connection is established:

► Transmit response to handshake messages manually.

  ➨ Handshake is concluded.

  ➨ Devices exchange CAN messages in binary format.

6.3.3 Changing the Message Format

The format changes automatically in the following situation:

- With command `C CAN_START` the transmission format is switched to ASCII.
- When Config connection is used to transmit a CAN message to the device in ASCII or binary format the device switches to the same format.
- If the device is in autostart mode and a handshake is carried out on the Config connection, the device switches to the binary format.

► To switch from ASCII to binary format or to disable the receiving of CAN messages use command `C SEND_CAN_FRAMES` with Config connection.

6.3.4 Setting the Transmitting Time

With the standard configuration messages from the device are collected for up to 4 ms before transmission. The minimum time between the transmission of two consecutive transmission packets can be adjusted.

► Adjust time between two transmission packets with command `D BUFF_TIMEOUT` (see *10.3 Device Commands* in extended User Manual).

  – Transmitting is possible before a Bluetooth SPP packet is filled completely.

  – With timeout 0 data is transmitted immediately. Protocol overhead is increased.

The size of a packet depends on the other node in the connection. CANblue II devices use data packets of up to 669 bytes between themselves.

6.3.5 Reset to Factory Settings

Factory settings:

- controller stopped
- filters deleted
- configuration deleted
- master table deleted
- transmitting time set to 4 ms
- passkey set to 7388
- visibility timeout set to 0

With Bluetooth Connection (Config)

► To reset device to factory settings, use command `D SETTINGS_DEFAULT`.

Without Bluetooth Connection

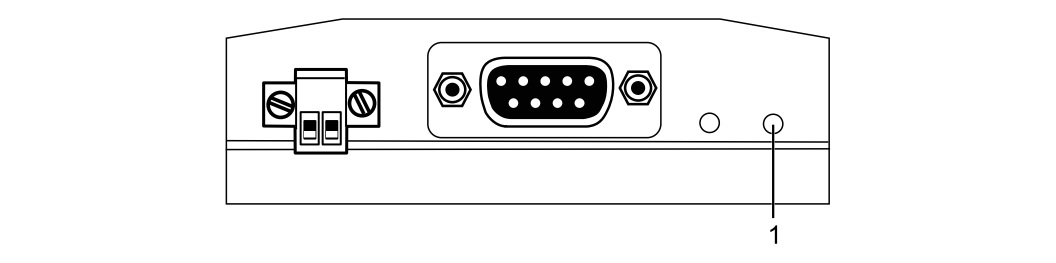


Fig. 16 Button Reset to factory settings

► Turn off the device.

► Press and hold button **Reset to factory Settings** (**1**).

► Turn on the device.

➡ CAN LED is flashing red and green.

► Release button **Reset to factory Settings** (**1**).

➡ If Mode LED flashes several times, the configuration is reset to factory settings.

6.3.6 Changing the Bluetooth Passkey

The default Bluetooth passkey is 7388.

► Change passkey with command `D PASSKEY_SET` (see *10.3 Device Commands* in extended User Manual).

– character strings with maximally 16 digits

6.3.7 Visibility

It is possible to adjust if the CANblue II is visible and how long it stays visible when connected to another device.

► To specify visibility, use command `D VISIBILITY TIMEOUT` (see *10.3 Device Commands* in extended User Manual).

– timeout: time device stays visible after connected to other device

– timeout 0: always visible

6.3.8 Connection Security in Bridge Setup

It is possible to configure the Slave devices to accept only Bluetooth connections from devices which MAC addresses are listed in the Master MAC address list.

► Add a MAC address to the Master MAC address list of the Slave device with command `D MAC_MASTER_ADD` (see *10.3 Device Commands* in extended User Manual).

– 10 entries are available

– 6 Byte hexadecimal

7 Operation

7.1 Overview

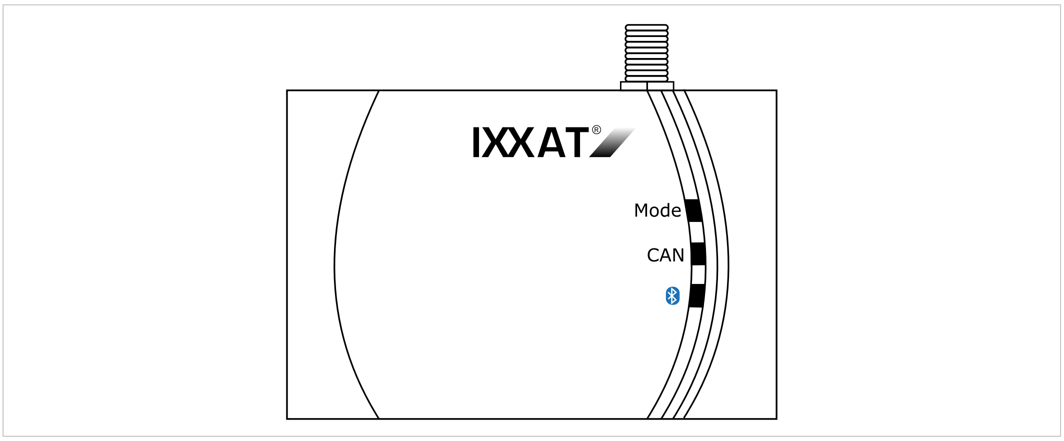


Fig. 17 LED array

7.2 Indicators

When the identity of device is requested, all LEDs are blinking.

7.2.1 Mode LED

LED	Description
Red	No Bluetooth MAC address stored in configuration, no connection to Slave on the unit

7.2.2 CAN LED

LED	Description
Green flashing	CAN message transmitted or received
Red flashing	CAN message transmitted or received, controller in *Warning* state
Red	CAN controller in *Bus-off* state

7.2.3 Bluetooth LED

LED	Description
Blue flashing (2 Hz)	Attempt to establish Bluetooth SPP connection with another device or established connection to the device
Blue flashing (10 Hz)	Bluetooth SPP used to transmit or receive data
Blue	Bluetooth SPP connection to another device is established.

7.3 Connection Behavior

If a Bluetooth MAC address is stored, the device attempts to establish a Bluetooth connection to that address for 5 seconds. If the connection attempt fails, every 2 seconds a new attempt is started.

The loss of a Bluetooth connection is detected after 3 seconds. The Master immediately attempts to establish a new connection.

If Bluetooth connection to CANblue II is lost, the CAN controller is automatically stopped.

8 Errors and Troubleshooting

Terminal program replies `E 99 Unknown Error` to correctly entered commands.	
Commands are not typed in capital letters.	► In terminal program enter commands in capital letters.
Loss of messages	
Messages are not forwarded • Former configuration stored on the device. Filters are not deleted with initialization.	► Before configuring the device, reset to factory settings or make sure all filters are deleted or disabled.
CAN transmission buffer overflow, oldest buffer entries are overwritten • CAN controller in *warning* or *bus off* state • Over 512 messages in buffer • Entries are overwritten to avoid blocking the receipt of data via the Bluetooth connection	► Reduce traffic.
Bluetooth transmission buffer overflow, incoming messages are discarded • Too many CAN messages	► Change transmission type. or ► Reduce traffic.
CAN receive buffer overflow, CAN messages are discarded, indicated by error message E 84 (in Config connection) • Config connection established during high traffic on the connected CAN network • Connection attempt of CANblue II	► Before establishing a Config connection reduce traffic or stop CAN controller.
Loss of device responses to commands transmitted on the Config connection	
High data traffic between Bluetooth and Config connection, if a command is transmitted via the Config connection, lines of the device response can be lost.	-
CAN controller in *warning* state	
Multiply incorrectly received or transmitted messages	Stopping and starting the CAN controller does not reset the *warning* state. ► Reset the device. or ► Make sure the device receives or transmits several valid CAN messages.

CAN controller in *bus off* state	
	Bus off recovery is started automatically: • 5 seconds after bus off CAN controller is stopped. • After 1 second CAN controller is started. • If CAN controller detects 128 valid messages on the bus (128 successive 11 bit sequences), all error flags are reset and CAN controller is in operating state. • Automatic bus off recovery is done until CAN controller is in *operating* state or stopped via Config connection. Manual bus off recovery: • Stop CAN controller via Config connection and restart.

Connection via previously used COM port is not possible after restart.	
If device is disconnected from power while still connected to Windows, COM port stays occupied.	► Make sure device is disconnected from Windows before disconnecting device from power.

CANblue II can not be accessed after computer was in sleep mode.	
If computer turns to sleep mode, the Bluetooth connection may not be closed completely. The connection remains and blocks all applications from connecting to the CANblue II. A restart of the CANblue II does not solve the issue.	If computer was in sleep mode, and connection is blocked: ► Reboot the computer. ► Stop CANblue II in **Device Server Control**. ► Stop CANblue II in canAnalyser by clicking the **Stop** button in the Control Panel. ► Stop all applications which access a CAN controller. ► In canAnalyser Control Panel remove the CANblue II via **Remove Device**. ► Reinstall the CANblue II (see Installation Guide *VCI Driver*).

9 PC Interface with VCI Driver Network and Device Communication

For information about network and device communication with VCI driver see Software Design Guides (.NET, C, C++) of VCI.

10 Generic Mode Network and Device Communication

To configure and transmit CAN messages via Bluetooth an ASCII protocol is defined. To permit a better data rate a binary format is also available for the transmission of Bluetooth CAN messages. To transmit messages between themselves CANblue II devices use the binary format.

10.1 ASCII Protocol

Structure of ASCII commands:

Message type	Command	Parameter 1	...	Parameter n	LF or CR-LF

Basic rules of ASCII protocol:

- Individual fields are separated by blanks.
- Multiple sequential blanks are considered to be a single blank.
- No distinction between capital and lower-case letters.
- Messages are terminated with ASCII linefeed control code (`LF` or `\n`) or with carriage return and linefeed (`CR LF` or `\r\n`).
- ASCII messages transmitted by the device are terminated with the same ASCII control codes as ASCII messages transmitted by the user. If no ASCII message is transmitted by the user, the device uses `CR-LF` as terminator.

The following message types (type defined by first byte) are supported:

- CAN commands (C)
- Device commands (D)
- CAN messages in ASCII format (M)
- CAN messages in binary format (X)
- Info messages (I)
- Error messages (E)

Examples

ASCII command	Response from device
C CAN_INIT 250	I OK: CAN_INIT
C CAN_START	I OK: CAN_START
C FILTER_ADD EXT 7FA1 RTR	I OK: FILTER_ADD
C SETTINGS_DEFAULT	I OK: SETTINGS_DEFAULT

10.2 CAN Commands

The commands are used to control the CAN controller on the device and to modify the filter settings.

Valid order of usage:

- ► Initialize CAN controller.
- ► Configure filter.
- ► Start CAN controller.
- ► Stop CAN controller.

10.2.1 Configuring the Communication Behavior

C CAN_INFO

Shows information about:

- current status of CAN controller
- software queues (Overrun flags are cleared after response is transmitted.)
- TX queue size
- transmitted CAN messages since last connection was established (TX counter is a WORD value and starts from zero when 65535+1, possible to implement TX handshake, if difference between local TX counter and CANblue II TX counter is calculated)

Syntax	`C CAN_INFO`
Parameter	-
Example	`C CAN_INFO`

Possible responses	`I CAN started` or `I CAN stopped I Tx queue size: 512`
	`I Tx counter: 0`
	`I CAN controller in WARNING LEVEL` or
	`I CAN controller in BUS OFF`
	`I Rx CAN controller OVERRUN`
	`I Rx SW queue OVERRUN`
	`I Tx SW queue OVERRUN`
	`I Tx pending`
	`I OK: CAN_INFO`

C CONFIG

It is possible to save, load and show the configuration.

Syntax	C CONFIG <operation>		
Parameter	operation	SAVE: Saves current configuration, can take several seconds. LOAD: Loads existing configuration. SHOW: Shows the configuration.	
Example	C CONFIG SHOW		
Possible responses	Save	I OK: CONFIG SAVE	
	Load	I OK: CONFIG LOAD	
	Show	I BT0=0, BT1=14 (1000 kBaud)	Values of bus timing registers. Name of configuration is given in brackets.
		I Bus coupling: HIGH	Bus coupling, exclusively HIGH supported
		I Autostart: ON	Autostart mode ON/OFF
		I STD filter list I CAN Id: 1 I CAN Id: 4, RTR bit set I STD filter enabled	Content of standard filter list
		I EXT filter list: I CAN Id: 4, RTR bit set I CAN Id: 7FFFF I EXT filter disabled	Content of extended filter list
		I MAC-Slave: 001122334455 Can-Bluet.-form.: binary, State: disconnected	Information about connection: MAC address, format of CAN messages (ASCII, BINARY, OFF), connection status (connected, disconnected)
		I MAC-Master: C44619F9813A Can-Bluet.-form.: off, State: connected	Information about Master
		I TX-Buff. timeout: 0	Timeout value of transmitting buffer
		I Passkey: 7388	Bluetooth passkey
		I Visibility: 0	Bluetooth visibility
		I MAC-Master List:	List with MAC IDs
		I OK: CONFIG SHOW	
Errors	E 63 Error while saving config	Error occurred during saving of configuration. Configuration is lost.	
	E 61 No valid config	No valid configuration	

C SEND_CAN_FRAMES

Enables or disables transmission of CAN messages in the directions the command comes from and sets the message format.

Syntax	C SEND_CAN_FRAMES<mode>	
Parameter	mode	Message format for transmitting via Bluetooth Value: ASCII/BINARY/OFF
Example	C SEND_CAN_FRAMES ASCII	
Possible responses	I OK: SEND_CAN_FRAMES	

10.2.2 Initializing the CAN Controller

C CAN_INIT

Initializes CAN controller with given baud-rate. Exclusively CiA standard baud-rates are supported (10, 20, 50, 100, 125, 250, 500, 800, 1000 kBaud).

Syntax	`C CAN_INIT <baud-rate><buscop>`	
Parameter	baud-rate	Baud-rate in kBaud. CAN controller is initialized with given baud-rate. Value: 10–1000 decimal (exclusively CiA standard)
	buscop	Mode of bus coupling, exclusively HIGH is supported. Value: HIGH
Example	`C CAN_INIT 500 HIGH`	
Possible responses	`I OK: CAN_INIT`	
Errors	`E 22 Baud-rate not supported`	Baud-rate is not supported. Use CiA supported baud-rate.
	`E 31 Error while initializing CAN`	Internal error while initializing CAN controller.
	`E 4 Unsupported parameter`	Bus coupling LOW is not supported.

C CAN_INIT_AUTO

Initializes the CAN controller with automatic baud-rate detection. CAN controller is set into TX passive mode and all CiA baud-rates are tested until a valid CAN message is received. CAN controller is initialized with detected baud-rate and response with the same baud-rate is transmitted.

Syntax	`C CAN_INIT_AUTO <timeout><buscop>`	
Parameter	timeout	Time in seconds to test for the CiA baud-rate. Value: 1–1000 decimal (optional, default: 1)
	buscop	Mode of bus coupling, exclusively HIGH is supported. Value: HIGH
Example	`C CAN_INIT_AUTO 10 HIGH`	
Possible responses	`I 100` (recognized baud-rate 100 kBaud) `I OK: CAN_INIT_AUTO`	
Errors	`E 23 Baud-rate not detected`	No baud-rate detected within the specified timeout. Maximum response time is 10 times of timeout value.
	`E 4 Unsupported parameter`	Bus coupling LOW is not supported.

C CAN_INIT_CUSTOM

Initializes the CAN controller with custom baud-rate. Parameters bt0 and bt1 correspond to bus timing register of Phillips SJA 1000 CAN controller with a clock frequency of 16 MHz.

Bit 7 of parameter bt1 is ignored, because the CANblue II CAN controller doesn’t support different sample rates.

Syntax	`C CAN_INIT_CUSTOM <bt0><bt1><buscop><name>`	
Parameter	bt0	SJA1000, bit timing register 0 Value: 0–FF hexadecimal
	bt1	SJA1000, bit timing register 1 Value: 0–FF hexadecimal
	buscop	Mode of bus coupling, exclusively HIGH is supported. Value: HIGH
	name	String enclosed in “”, max. 30 characters Name of bus timing configuration, name is used for command `C CONFIG SHOW`. If no name is given, baud-rate is used as name.
Example	`C CAN_INIT_CUSTOM 0 1C HIGH 1000KBAUD CUSTOM`	
Possible responses	`I OK: CAN_INIT_CUSTOM`	
Errors	`E 31 Error while initializing CAN`	Internal error while initializing CAN controller.
	`E 4 Unsupported parameter`	Bus coupling LOW is not supported.

10.2.3 Configuring the Filter

C FILTER_ADD

Adds a filter entry to the filter list. The filter works as a positive filter. Received messages which are in the list are forwarded. Messages received via Bluetooth are not filtered.

For information about space and used memory see *Configuring the Filter, p. 24*.

Syntax	`C FILTER_ADD <msg_typ><id><rtr>`	
Parameter	msg_typ	Message type of filter entry (standard or extended) Value: STD/EXT
	id	CAN ID of filter entry. Standard: 0-7FF, Extended: 0-1FFFFFFF
	rtr	Data or remote frame Value: DATA/RTR (optional, default: DATA)
Example	`C FILTER_ADD STD 3A RTR`	
Possible responses	`I OK: FILTER_ADD`	
Errors	`E 34 Error adding ID to filter`	Out of memory for extended filter elements.

C FILTER_REMOVE

Removes a filter from entry list.

Syntax	`C FILTER_REMOVE <msg_typ><id><rtr>`	
Parameter	msg_typ	Message type of filter entry (standard or extended) Value: STD/EXT
	id	CAN ID of filter entry. Standard: 0-7FF, Extended: 0-1FFFFFFF
	rtr	Data or remote frame Value: DATA/RTR (optional, default: DATA)
Example	`C FILTER_REMOVE STD 3A RTR`	
Possible responses	`I OK: FILTER_REMOVE`	

C FILTER_CLEAR

Erases standard or extended filter list.

Syntax	`C FILTER_CLEAR <id-typ>`	
Parameter	id-typ	Message type of filter entry (standard or extended) Value: STD/EXT
Example	`C FILTER_CLEAR EXT`	
Possible responses	`I OK: FILTER_CLEAR`	

C FILTER_ENABLE

Enables standard or extended filter list. Messages are forwarded if ID is found in the filter list. Filter list for standard and extended IDs must be enabled or disabled separately.

Syntax	`C FILTER_ENABLE <id-typ>`	
Parameter	id-typ	Message type of filter entry (standard or extended) Value: STD/EXT
Example	`C FILTER_ENABLE EXT`	
Possible responses	`I OK: FILTER_ENABLE`	

C FILTER_DISABLE

Disables standard or extended filter list. Filter list for standard and extended IDs must be disabled separately.

Syntax	`C FILTER_DISABLE <id-typ>`	
Parameter	id-typ	Message type of filter entry (standard or extended) Value: STD/EXT
Example	`C FILTER_DISABLE EXT`	
Possible responses	`I OK: FILTER_DISABLE`	

10.2.4 Starting the CAN Controller

C CAN_START

Starts the CAN controller. Message format for transmitting CAN messages over Bluetooth is set to ASCII mode.

Syntax	`C CAN_START`	
Parameter	-	
Example	`C CAN_START`	
Possible responses	`I OK: CAN_START`	
Errors	`E 32 Error starting CAN`	Internal error while initializing CAN controller.

C AUTOSTART

Enables or disables autostart mode (further information see *Autostart, p. 25*).

Syntax	`C AUTOSTART<mode>`	
Parameter	mode	Enable or disable autostart mode. Value: ON/OFF
Example	`C AUTOSTART ON`	

Possible responses	`I AUTOSTART ON` or `I AUTOSTART OFF`
	`I OK: AUTOSTART`	

10.2.5 Stopping the CAN Controller

C CAN_STOP

Stops the CAN controller.

Syntax	`C CAN_STOP`	
Parameter	-	
Example	`C CAN_STOP`	
Possible responses	`I OK: CAN_STOP`	
Errors	`E 33 Error stop CAN`	Internal error while initializing CAN controller.

10.2.6 Reset the CAN Controller

C CAN_RESET

Resets the CAN controller.

Syntax	`C CAN_RESET`
Parameter	-
Example	`C CAN_RESET`
Possible responses	`I OK: CAN_RESET`

10.3 Device Commands

10.3.1 Requesting Device Information

D VERSION

Gets firmware version of CANblue II.

Syntax	`D VERSION`
Parameter	-
Example	`D VERSION`

Possible responses	`I CANblue Generic — Bridge v2.00.03`
	`I OK: VERSION`

D PROTOCOL

Gets ASCII protocol version

Syntax	`D PROTOCOL`
Parameter	-
Example	`D PROTOCOL`

Possible responses	`I ASCII Extended Protocol v1.2`
	`I OK: PROTOCOL`

D IDENTIFY

Gets hardware version number and name of CANblue II. Device name contains Bluetooth MAC address. All LEDs of the CANblue II are flashing.

Syntax	`D IDENTIFY`
Parameter	-
Example	`D IDENTIFY`

Possible responses	`I Name: IXXAT CANblue II (1A2B3C4D5E6F)`
	`I HW-Number: HW 999999`
	`I OK: IDENTIFY`

D INFO

Shows information about the configured Bluetooth connection settings and the Bluetooth connections. Additional information like connection quality, receive signal strength or transmission power is shown for each connection.

Syntax	`D INFO`	
Parameter	-	
Example	`D INFO`	
Possible responses	`I Link-policy parameter:`	Bluetooth connection settings
	`I Settingname: DEFAULT`	Name of configured connection settings (see *D LINK_POLICY, p. 42*)
	`I Packettype: CC18`	Bluetooth packet types
	`I PagescanInterval: 800`	Page scan interval
	`I PagescanWindow: 12`	Page scan window
	`I PagescanType: 0`	Page scan type
	`I Latency (wished): 40`	Max. Bluetooth latency in Bluetooth time slots of 625 µs
	`I Tx-Power (max): 14 dBm`	Max. allowed Bluetooth transmission power
	`I MAC, Latency, Link quality, RSSI, Tx-Power, PacketType`	Table of current Bluetooth connections
	`I 123456789ABC, 40*625us, 100%, 15 dB, 1 dBm, CC18`	Table entry of one connection: MAC address, latency in µs, connection quality in %, receive signal strength indication in dB (-127 dB to + 128 dB), transmission power in dBm (-18 dBm to +14 dBm), Bluetooth packet types in use
	`I OK: INFO`	

10.3.2 MAC Commands for Connecting Devices

D MAC_ADD

Adds MAC address to the connection list of a CANblue II. The device tries to establish a connection to a Bluetooth device with the added MAC address.

Syntax	`D MAC_ADD<adr>`	
Parameter	adr	MAC address of second CANblue II (Slave) Value: 6 Byte hexadecimal
Example	`D MAC_ADD 001122334455`	
Possible responses	`I OK: MAC_ADD`	
Errors	`E 51 MAC list is full`	Only one MAC address is supported.
	`E 53 MAC address already exists`	MAC address is already used for a connection to a server.

D MAC_REMOVE

Removes a MAC address from the connection list of a CANblue II. An active connection or attempt to establish a connection is closed, when the command is called. This can cause a delayed response up to 5 seconds.

Syntax	`D MAC_REMOVE<adr>`	
Parameter	adr	MAC address of CANblue II (Slave) to be removed from list. Value: 6 Byte hexadecimal
Example	`D MAC_REMOVE 001122334455`	
Possible responses	`I OK: MAC_REMOVE`	
Errors	`E 52 Wrong MAC address`	MAC address is not valid or not in connection list.

D MAC_CLEAR

Removes all MAC addresses from the connection list of a CANblue II. An active connection or attempt to establish a connection is closed, when the command is called. This can cause a delayed response up to 5 seconds.

Syntax	`D MAC_CLEAR`
Parameter	-
Example	`D MAC_CLEAR`
Possible responses	`I OK: MAC_CLEAR`

D MAC_SCAN

Starts scan for other Bluetooth devices. After scan time is expired the response lists all active devices with name and Bluetooth MAC address. Due to a device name query the response can be delayed up to 5 seconds for every device. Maximally 10 devices can be listed.

Syntax	`D MAC_SCAN<time>`	
Parameter	time	Scan time in seconds Value: 1–255 decimal (optional, default: 10)
Example	`D MAC_SCAN 20`	

Possible responses	`I MAC-Address, Name`
	`I 001122334455 Device 1`
	`I 010203040506 IXXAT CANblue(010203040506)`
	`I 012345678901 Mobile Phone X`
	`I 010203040507 IXXAT CANblue II(010203040507)`
	`I OK: MAC_SCAN`	

Errors	`E 52 Wrong MAC address`	MAC address is not valid or not in connection list.

10.3.3 MAC Commands Security

D MAC_MASTER_ADD

Adds a MAC address to the Master MAC address list. Slave devices then only accept Bluetooth connections from devices which MAC addresses are listed in the Master MAC address list. In the Master MAC address list 10 entries are available.

Syntax	`D MAC_MASTER_ADD<adr1>`	
Parameter	adr1	MAC address Value: 6 Byte hexadecimal
Example	`D MAC_MASTER_ADD 001122334455`	
Possible responses	`I OK: MAC_MASTER_ADD`	
Errors	`E 51 MAC list is full`	Only 10 MAC address list entries are supported.
	`E 52 Wrong MAC address`	MAC address is invalid. Valid MAC address consists of 12 digits.
	`E 53 MAC address already exists`	MAC address is already used for a connection to a server.

D MAC_MASTER_REMOVE

Removes a MAC address from the Master MAC address list. Address is exclusively removed if MAC address match an entry in the Master MAC address list.

Syntax	`D MAC_MASTER_REMOVE<adr>`	
Parameter	adr	MAC address to be removed from list. Value: 6 Byte hexadecimal
Example	`D MAC_MASTER_REMOVE 001122334455`	
Possible responses	`I OK: MAC_MASTER_REMOVE`	
Errors	`E 52 Wrong MAC address`	MAC address is not valid or not in connection list. Valid MAC address consists of 12 digits.

D MAC_MASTER_CLEAR

Removes all MAC addresses from the Master MAC address list. After clearing the list, a slave device accepts a Bluetooth connection from all devices.

Syntax	`D MAC_MASTER_CLEAR`
Parameter	-
Example	`D MAC_MASTER_CLEAR`
Possible responses	`I OK: MAC_MASTER_CLEAR`

10.3.4 Configuring the Device

D CONFIG

It is possible to save, load and show the configuration.

Syntax	`D CONFIG <operation>`		
Parameter	Operation	`SAVE`: Saves current configuration, can take several seconds. `LOAD`: Loads existing configuration. `SHOW`: Shows the configuration.	
Example	`D CONFIG SHOW`		
Possible responses	Save	`I OK: CONFIG SAVE`	
	Load	`I OK: CONFIG LOAD`	
	Show	`I BT0=0, BT1=14 (1000 kBaud)`	Values of bus timing registers. Name of configuration is given in brackets.
		`I Bus coupling: HIGH`	Bus coupling, exclusively HIGH supported
		`I Autostart: ON`	Autostart mode ON/OFF

		`I STD filter list`
		`I CAN Id: 1`
		`I CAN Id: 4, RTR bit set`
		`I STD filter enabled`
			Content of standard ID filter list

		`I EXT filter list:`
		`I CAN Id: 4, RTR bit set`
		`I CAN Id: 7FFFF`
		`I EXT filter disabled`
			Content of extended ID filter list

		`I MAC-Slave: 001122334455 Can-Bluet.-form.: binary, State: disconnected`	Information about connection: MAC address, format of CAN messages (ASCII, BINARY, OFF), connection status (connected, disconnected)
		`I MAC-Master: C44619F9813A Can-Bluet.-form.: off, State: connected`	Information about Master
		`I TX-Buff. timeout: 0`	Timeout value of transmitting buffer
		`I Passkey: 7388`	Bluetooth passkey
		`I Visibility: 0`	Bluetooth visibility
		`I MAC-Master List:`	List with MAC IDs
		`I OK: CONFIG SHOW`	

D PASSKEY_SET

Changes the Bluetooth passkey.

Syntax	`D PASSKEY_SET <key>`	
Parameter	key	Bluetooth passkey, up to 16 digits Value: character string
Example	`D PASSKEY_SET 1234567890ABCD`	
Possible responses	`I OK: PASSKEY_SET`	
Errors	`E 13 Wrong data length`	Passkey is invalid. Valid passkey consists of maximally 16 digits.

D VISIBILITY TIMEOUT

Changes the Bluetooth visibility.

Syntax	`D VISIBILITY <timeout>`	
Parameter	timeout	Time in seconds after which device is invisible to other devices. If timeout is 0, device is always visible. If timeout is unequal 0, device is invisible after connection to another device is established or after timeout is exceeded. Value: 0–60000 decimal
Example	`D VISIBILITY 60`	
Possible responses	`I OK: VISIBILITY`	
Errors	`E 2 Wrong parameter`	Timeout value is out of range.

D BUFF_TIMEOUT

Sets timeout for transmitting buffer resp. time between two consecutive TX Bluetooth packets of the CANblue II. Timeout is applied to all Bluetooth connections of the device (further information see *Setting the Transmitting Time, p. 25*).

Syntax	`D BUFF_TIMEOUT <time>`	
Parameter	time	Collecting time of RX CAN message in milliseconds. Value: 0–1000 decimal
Example	`D BUFF_TIMEOUT 4`	
Possible responses	`I OK: BUFF_TIMEOUT`	

D LINK_POLICY

Sets properties of Bluetooth connection.

Syntax	`D LINK—POLICY <conf>`	
Parameter	conf	Predefined Bluetooth configurations, selection is applied on all Bluetooth connections to get best results. **DEFAULT**: Balanced configuration, suitable for more than one connection in parallel and for *none*-CANblue II devices **SHORTEST_LATENCY**: Reduced latency for Bluetooth messages. Settings also reduces data rate to approx. 2000 CAN Msg/s per direction. With this setting only one connection per device is possible. If a connection between the devices is established they cannot be found by a Bluetooth scan. **QUICKEST_CONNECTION**: Allows faster establishment of a Bluetooth bridge. Setting increases power consumption of the device and reduces data rate. **MOST_ROBUST_CONNECTION**: Allows bridging a long distance and Bluetooth connection is more insusceptible to disturbances. Setting reduces data rate to approx. 3000 CAN Msg/s per direction.
Example	`D LINK_POLICY SHORTEST_LATENCY`	
Possible responses	`I OK: LINK_POLICY`	

10.3.5 Reset the Device

D RESET

The device transmits the response and resets itself. Any established Bluetooth connections are lost.

Syntax	`D RESET`
Parameter	—
Example	`D RESET`
Possible responses	`I OK: RESET`

D SETTINGS_DEFAULT

The configuration is reset to the factor default settings. Stored configurations are deleted.

Syntax	`D SETTINGS_DEFAULT`
Parameter	—
Example	`D SETTINGS_DEFAULT`
Possible responses	`I OK: SETTINGS_DEFAULT`

10.4 CAN Messages in ASCII format

CAN Messages coded in ASCII format are called M-type messages.

M-type messages are used to transmit CAN messages over a Bluetooth connection to another device. The receiving device forwards the message to all established Bluetooth connections and if the local CAN controller is started the message is transmitted to the CAN network.

Remote messages are transmitted without any data bytes, but the value of the data length (DLC) can be a value between 0 and 8.

Syntax	`M FTD ID D0 D1 D2 D3 D4 D5 D6 D7`	
Parameter	FTD	Three characters defining the message format. 1. character: frame format (S — Standard, E — Extended) 2. character: frame type (D — Data, R — Remote) 3. character: DLC (“0–8” data length)
	ID	CAN message identifier Standard: 0–7FF hexadecimal Extended: 0–7FFFFFFF hexadecimal
	D0–D7	Data bytes of message, messages consist of up to 8 data bytes, every byte is separated by a blank. Value: 0–FF hexadecimal
Example	`M SD4 1A2 11 22 33 4`	
Possible responses	–	
Errors	`E 85 Tx SW queue OVERRUN`	Overrun of the transmitting queue, for example CAN controller is in *error warning* or *bus off* state or data could not be transmitted fast enough due to slow baud-rate.

10.5 CAN Messages in Binary Format

CAN Messages coded in binary format are called X-type messages.

Basic features of binary format:

- allows faster transmission of CAN messages
- data of the CAN message is transmitted uncoded in a binary value
- fields are not separated by blanks
- fields are without CR/LF characters

Used to transmit CAN messages over a Bluetooth connection to another device. The receiving device forwards the message to all established Bluetooth connections and if the local CAN controller is started the message is transmitted to the CAN network.

Syntax	Standard CAN message:
	`X FI ID_HB ID_LB D0 D1 D2 D3 D4 D5 D6 D7`
	Extended CAN message:
	`X FI ID_HW_HB ID_HW_LB ID_LW_HB ID_LW_LB D0 D1 D2 D3 D4 D5 D6 D7`	

Parameter	FI (bit field)	FF (bit 7): Frame format (0 — Standard, 1 — Extended) RTR (bit 6): Frame type (0 — Data, 1 — Remote) DLC (bit 0–3): Data length 0–8
	ID_HB	High byte of standard CAN ID (0–7F)
	ID_LB	Low byte of standard CAN ID (0–FF)
	ID_HW_HB	High word, high byte of extended CAN ID (0–1F)
	ID_HW_LB	High word, low byte of extended CAN ID (0–FF)
	ID_LW_HB	Low word, high byte of extended CAN ID (0–FF)
	ID_LW_LB	Low word, low byte of extended CAN ID (0–FF)
	D0–D7	Up to 8 data bytes (0–FF)
Example	`0x58, 0x85, 0x01, 0x02, 0x03, 0x04, 0x19, 0x2A, 0x3B, 0x4C, 0x5D` `0x58` X(binary message type) `0x85` [FF=1 (Ext); RTR=0 (Data); DLC = 5] `0x01020304` ID `0x19, 0x2A, 0x3B, 0x4C, 0x5D` 5 data bytes	
Possible responses	–	
Errors	`E 85 Tx SW queue OVERRUN`	Overrun of the transmitting queue, for example CAN controller is in *error warning* or *bus off* state or data could not be transmitted fast enough due to slow baud-rate.

10.6 Error Messages

Error message	Description
`E 1 Unknown command`	Invalid command or message type is received.
`E 2 Wrong parameter`	Parameter of a command is invalid.
`E 3 Unsupported command`	Received command is not supported.
`E 4 Unsupported parameter`	Parameter of a command is not supported.
`E 11 Wrong message type`	Invalid message type is received (valid: standard or extended).
`E 12 Wrong frame type`	Invalid frame type is received (valid: data or remote).
`E 13 Wrong data length`	Invalid data length is received (valid: 0–8).
`E 14 Wrong message ID`	Invalid ID is received (valid: 0–7FF or 0–1FFFFFFF).
`E 15 Wrong number of data bytes`	Number of data bytes does not match data length.
`E 21 Unknown Bus Coupling value`	Invalid bus coupling value (valid: high).
`E 22 Baudrate not supported`	Baud-rate is no CiA baud-rate.
`E 23 Baudrate not detected`	No valid baud-rate detected by automatic baud-rate detection within the specified timeout.
`E 31 Error while initializing CAN`	CAN controller not initialized. Try to initialize again.
`E 32 Error starting CAN`	CAN controller not started. Try to start again.
`E 33 Error stop CAN`	CAN controller not stopped. Try to stop again.
`E 41 Error adding ID to filer`	Out of memory for extended filter elements.
`E 51 MAC-list is full`	Not possible to add another MAC address.
`E 52 Wrong MAC Address`	MAC address is not valid (valid: 6 byte hexadecimal).
`E 53 MAC Address already exist`	MAC address is already used for a connection to a server.
`E 61 No valid config`	No valid configuration to load.
`E 63 Error while saving config`	Error during saving of the configuration. Configuration is lost.
`E 81 CAN controller in BUS OFF`	CAN controller is in bus off state.
`E 82 CAN controller in WARNING LEVEL`	CAN controller is in error warning state.
`E 84 Rx SE queue OVERRUN`	One or more consecutive CAN messages are lost because of a software overrun.
`E 85 Tx SW queue overrun`	One or more consecutive CAN messages are lost before transmitting by the CAN controller because CAN controller is in *bus off* or error *warning* state or because of slow baud-rate.
`E 91 Can’t show more`	Not all filter elements are shown. Depending on free space of transmitting buffer only a limited amount of filter element can be shown with the command `C CONFIG SHOW`.
`E 99 Unknown Error`	Internal error occurred. No specific error message specified.

11 Technical Data

Bluetooth qualification	v4.0 (Bluetooth classic)
Output power	11 dBm, internal antenna 13 dBm, external antenna
Bluetooth output frequency	2.402 to 2.480 GHz, ISM band
CAN transceiver	Texas Instruments SN65HVD251
Max. number of CAN bus nodes	120
Power supply	9 to 30 V DC
Power consumption	Typ. 50 MA at 12 V
Dimensions	81 x 66 x 26 mm
Weight	Approx. 83 g
Operating temperature	-40 °C to 85 °C
Relative humidity	10 to 95%, non-condensing
CAN interface isolation working voltage	130 V AC/DC (continuous) 1000 V DC (1 second)
External antenna	RP-SMA connector, max. antenna gain 3.4 dBi
Bridge set-up time	Typ. 3 to 4 seconds
Bluetooth transfer delay	Approx. 4 ms (average), CAN — Bluetooth, or Bluetooth — CAN
CAN transmission rate	100% bus load at 1 MBit
Maximal distance between two devices in bridge mode	200 m/650 ft

12 Default Settings

Pairing Code	7388
MAC address	Printed on the back of the device

13 Support/Return Hardware

Observe the following information in the support area on www.ixxat.com:

- information about products
- FAQ lists
- installation notes
- updated product versions
- updates

13.1 Support

- ► For problems or support with the product request support at www.ixxat.com/support.
- ► If required use support phone contacts on www.ixxat.com.

13.2 Return Hardware

- ► Fill in the form for warranty claims and repair on www.ixxat.com.
- ► Print out the Product Return Number (PRN resp. RMA).
- ► Pack product in a physically- and ESD-safe way, use original packaging if possible.
- ► Enclose PRN number.
- ► Observe further notes on www.ixxat.com.
- ► Return hardware.

14 Disposal

- ► Dispose of product according to national laws and regulations.
- ► Observe further notes about disposal of products on www.ixxat.com.

This page intentionally left blank

A Regulatory Compliance

> ! The IXXAT CANblue II with external antenna port is for OEM integrations only. The end-user product will be professionally installed in such a manner that only the authorized antennas are used. A list of authorized antennas is available from www.ixxat.com.

A.1 EMC Compliance (CE)

CE

The product is in compliance with the Electromagnetic Compatibility Directive. More information and the Declaration of Conformity is found at www.ixxat.com.

A.2 FCC Compliance Statement

This device complies with Part 15 of the FCC Rules. Operation is subject to the following two conditions:

- This device may not cause harmful interference.
- This device must accept any interference received, including interference that may cause undesired operation.

Product name	CANblue
Model	II
Responsible party	HMS Industrial Networks Inc
Address	35 E. Wacker Dr, Suite 1700 Chicago , IL 60601
Phone	+1 312 829 0601

> ! Any changes or modifications not expressly approved by HMS Industrial Networks could void the user's authority to operate the equipment.

> **!** This equipment has been tested and found to comply with the limits for a Class B digital device, pursuant to Part 15 of the FCC rules. These limits are designed to provide reasonable protection against harmful interference in a residential installation. This equipment generates, uses and can radiate radio frequency energy and, if not installed and used in accordance with the instructions, may cause harmful interference to radio communications. However, there is no guarantee that interference will not occur in a particular installation. If this equipment does cause harmful interference to radio or television reception, which can be determined by turning the equipment off and on, the user is encouraged to try to correct the interference by one or more of the following measures:
>
> Reorient or relocate the receiving antenna.
>
> Increase the separation between the equipment and the receiver.
>
> Connect the equipment into an outlet on a circuit different from that to which the receiver is connected.
>
> Consult the dealer or an experienced radio/TV technician for help.

A.3 RoHs Directive

The product is in compliance with the RoHs Directive 2002/95/EC (Restriction of the use of certain hazardous substances in electrical and electronic equipment).

A.4 Japan Radio Equipment Compliance (TELEC)

CANblue II uses the cB-0946 module which complies with the Japanese Technical Regulation Conformity Certification of Specified Radio Equipment (ordinance of MPT N°. 37, 1981), Article 2, Paragraph 1, Item 19, “2.4 GHz band wide band low power data communication system”. The cB-0946 MIC certification number is 204-210003.

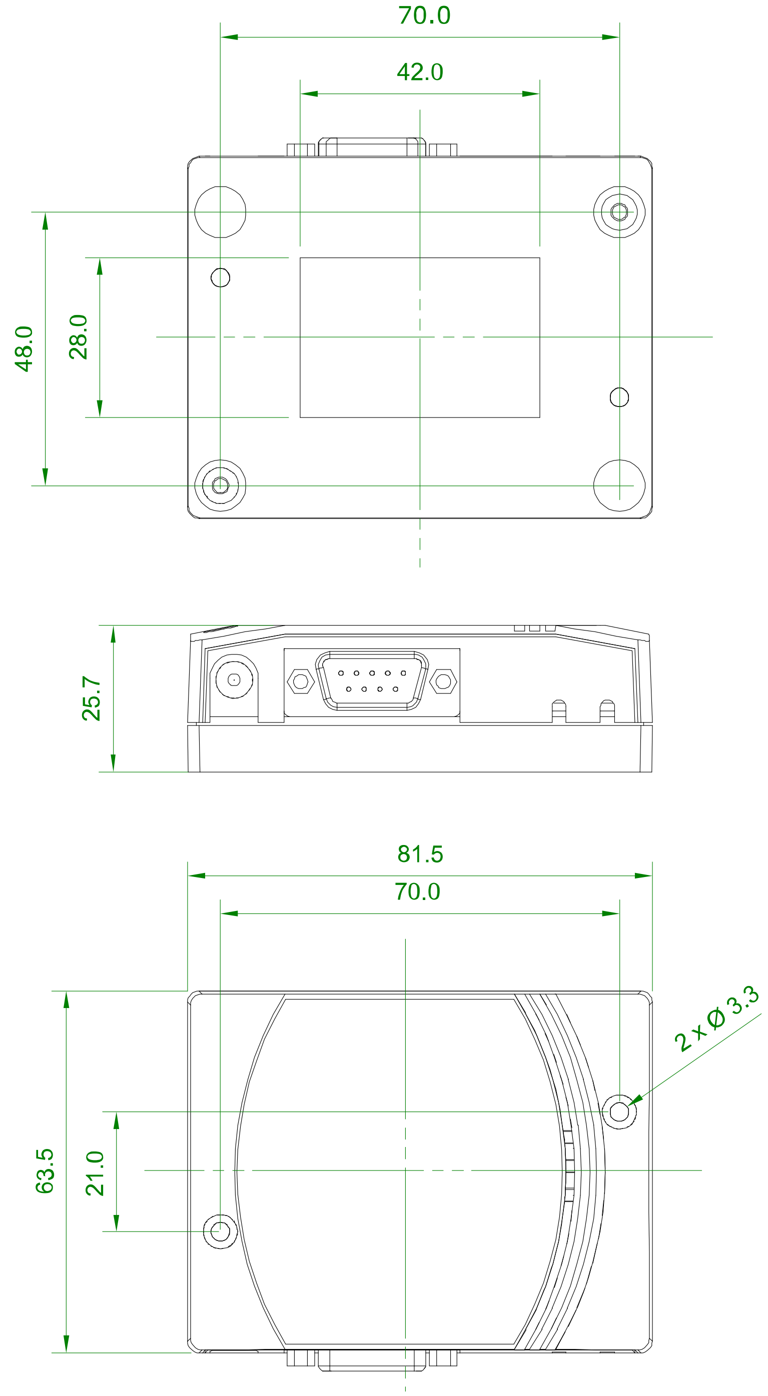
R **204-210003**

B Disposal and recycling

You must dispose of this product properly according to local laws and regulations. Because this product contains electronic components, it must be disposed of separately from household waste. When this product reaches its end of life, contact local authorities to learn about disposal and recycling options, or simply drop it off at your local HMS office or return it to HMS.

For more information, see www.hms-networks.com.

C Measurements



D Configuration Examples

D.1 Example 1: Connecting a CAN Network With a Computer

The example shows how an installed virtual COM port can be used to configure the CANblue II to exchange data with a CAN network connected to the CANblue II.

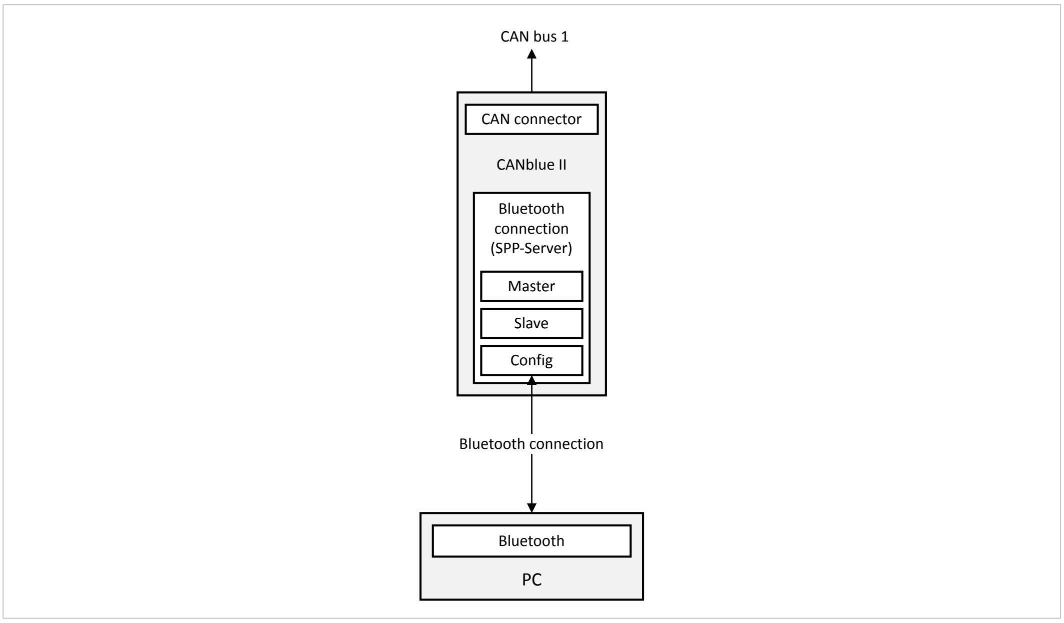


Fig. 18 PC Interface

The following specifications apply in the example:

- CAN network is operated at data rate of 500 kBaud.
- Exclusively the following messages are forwarded by the CANblue II:
  - data and remote frames with standard identifier 5
  - remote frames with standard identifier 1F
  - data frames with extended identifier 1A2B3C

► Make sure virtual COM port is installed.

► Reset device to factory settings with command `D SETTINGS_DEFAULT`.

► Initialize CAN controller to 500 kBaud with command `C CAN_INIT 500`.

► To set the filter, use the following commands:

- `C FILTER_ADD 5`
- `C FILTER_ADD STD 5 RTR`
- `C FILTER_ADD STD 1F`
- `C FILTER_ADD EXT 1A2B3C`

► Activate standard filter with command `C FILTER_ENABLE STD`.

► Activate extended filter with command `C FILTER_ENABLE EXT`.

► Check configuration with command `C CONFIG SHOW`.

► Save configuration with command `C CONFIG SAVE`.

- ► Start CAN controller with command `C CAN_START`.
  - ➡ If CAN controller receives a message from CAN network that matches one of the filters, the message is transmitted on the Bluetooth connection in ASCII format.
- ► To transmit CAN messages to the CANblue II or into the connected CAN network use ASCII or binary format (see *Generic Mode Network and Device Communication, p. 30*).
  - ➡ Transmission format of CAN messages are automatically matched to the received format.
- ► To transmit a CAN data frame with standard identifier 7FF and data bytes *1A 2B 3C 4D 5E 6F 70* to the CAN bus, use command `M SD7 7FF 1A 2B 3C 4D 5E 6F 70`.

D.2 Example 2: Configuring a CAN Bridge

The example shows how a CANblue II (configured as in example 1) is connected to a second CANblue II.

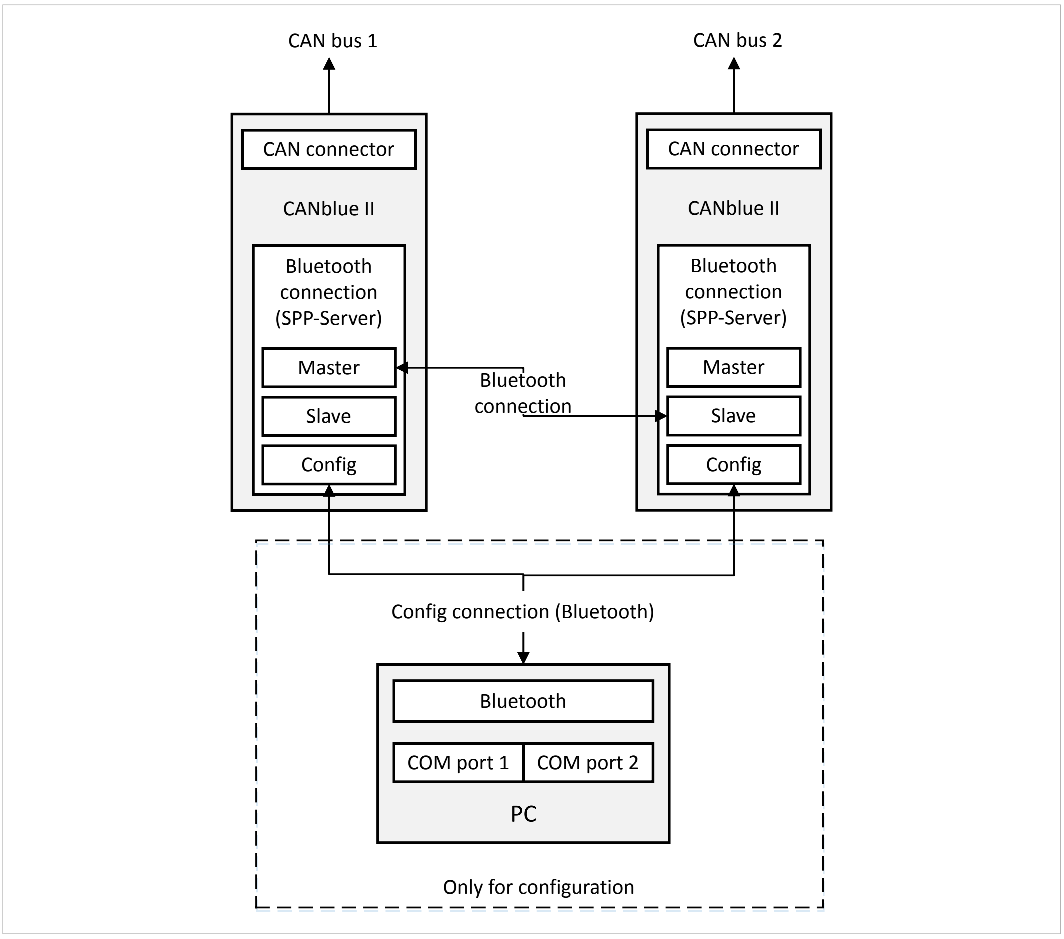


Fig. 19 Configuring a bridge

The following specifications apply in the example:

- Slave is connected to 1000 kBaud CAN network.
- Master (configured as in example 1) forwards all standard CAN messages and filters out all extended CAN messages.
- Slave forwards all CAN messages.

Slave Device

- Make sure virtual COM port is installed and connection is established.
- Reset device to factory settings with command `D SETTINGS_DEFAULT`.
  - ➡ CAN controller is automatically initialized to 1000 kBaud (preset in factory settings).
- Enable autostart mode with command `C AUTOSTART ON`.
- Save configuration with command `C CONFIG SAVE`.

Master Device

- Make sure Master (configured as in example 1) is connected to virtual COM port.

To simplify the configuration, turn off transmission of CAN messages by Master:

- Stop CAN controller with command `C CAN_STOP`.

  or

- Disable transmission of CAN messages on the specific connection with command `C SEND_CAN_FRAMES OFF`.
- Disable filter of Master with command `C FILTER_DISABLE STD`.
  - ➡ All standard CAN messages are forwarded by Master.
- Delete all extended filter entries with command `C FILTER_CLEAR EXT`.
  - ➡ Device filters out all extended CAN messages (set by `C FILTER_ENABLE EXT` in example 1).
- Enable autostart mode of Master with command `C AUTOSTART ON`.
- With Master use command `D MAC_ADD <address of Slave>`.
  - ➡ Devices connect as Master and Slave and start automatically.
  - ➡ Devices function as bridge between the two CAN networks.
- Save configuration with command `C CONFIG SAVE`.
- To achieve highest possible data rate between the devices, disconnect Config connection to the computer.

Since the connection is stored on both devices, devices reconnect automatically after turning off and on and resume forwarding of CAN messages.

D.3 Example 3: Configuring a Bridge Chain

Configuring a bridge chain is possible because every Slave can serve as Master for another Slave.

To connect a third CAN bus using an additional CANblue II with the CAN buses configured in example 1 and 2, two options are possible:

- connecting Slave to new device (Slave serves as Master for new device)
- connecting new device to Master (Master serves as Slave for new device)

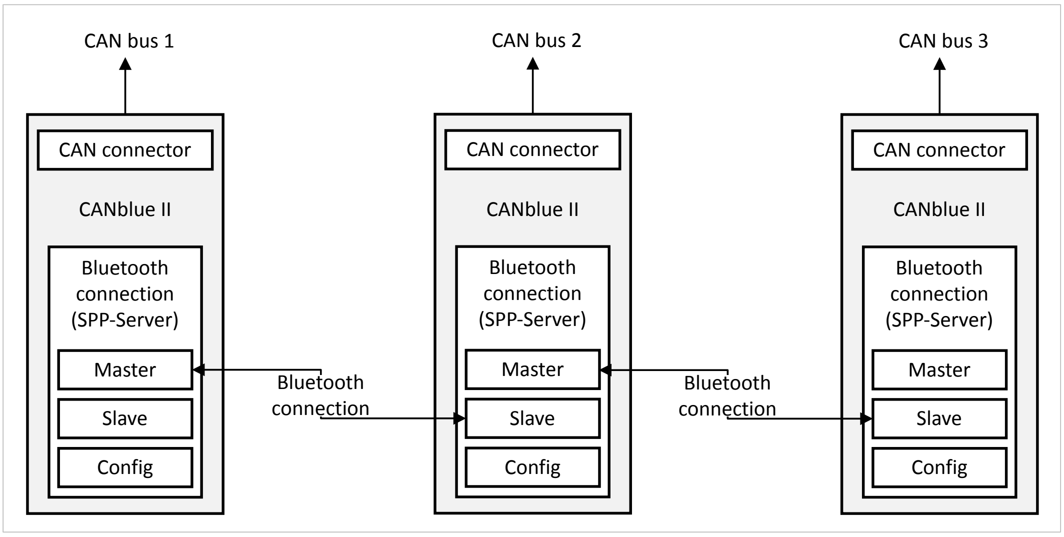


Fig. 20 Bridge configuration

Each additional CAN bus increases rate of CAN messages on the Bluetooth connections and reduces the maximum possible data rate of all connections.

Box 4126
300 04 Halmstad, Sweden

info@hms.se