# Operating Manual

# UCC 1, Universal CAN Controller

| | |
|---|---|
| Document version: | 1.0 |
| Created on: | 29.06.2014 |
| Last modification on: | 07.07.2014 |
| Valid for devices with part number: | 10000612 |
| Valid for devices from serial number: | 1 |

Table of Content

# Overview

The UUC 1 device supports serial communication with MICBAC compatible control units via CAN interface. Up to eight units may be controlled simultaneously.

# Interfaces

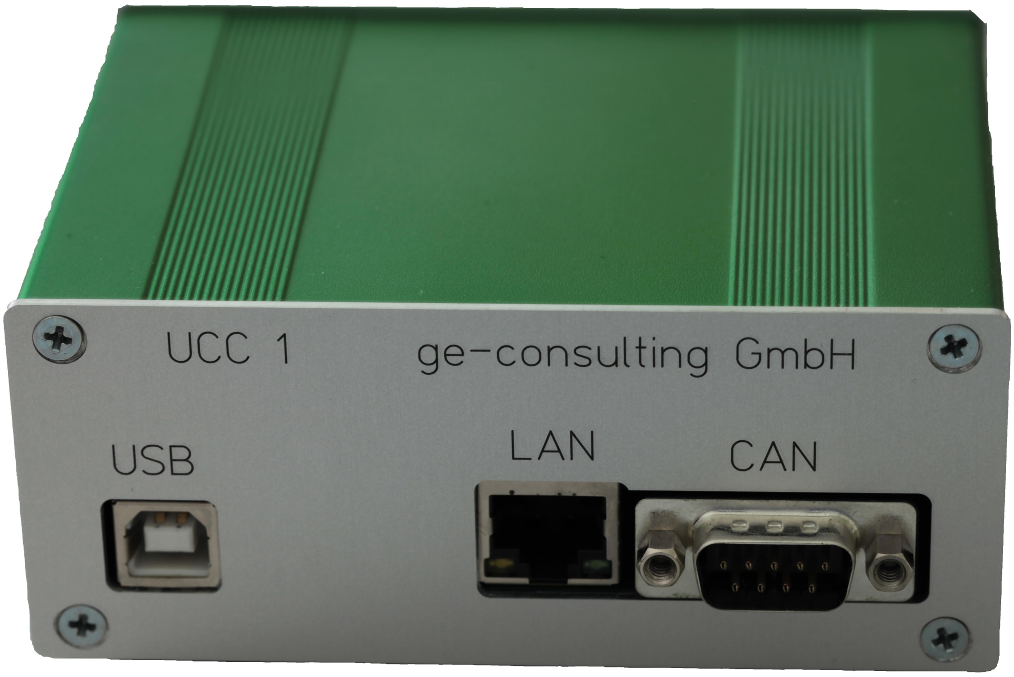
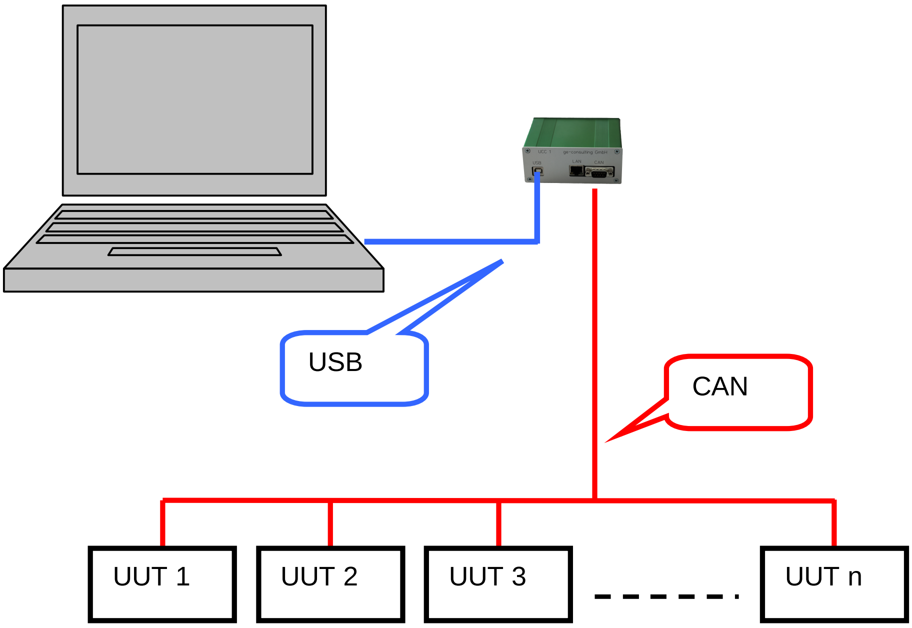


**Figure 1: Interfaces**

| Interface connector | Connect to | Remark |
|---|---|---|
| USB | Controlling device (e. g. Laptop or similar) | USB 2.0 high speed |
| CAN | UUT | Up to 8 UUTôs may be controlled simultaneously, but the number of UUTôs connected to the CAN bus is not limited. |
| Ethernet (optional) | Controlling device (e. g. Laptop or similar) via LAN | Available on request |

**Table 1: Interfaces**

## Connection Diagram



# Setup

## Software

### System Requirements

Microsoft Windows XP, Vista, Windows 7, Windows 8, UNIX, LINUX

Terminal software supporting serial communication via COM port like PuTTY, OC-Console or similar

USB 2.0 interface

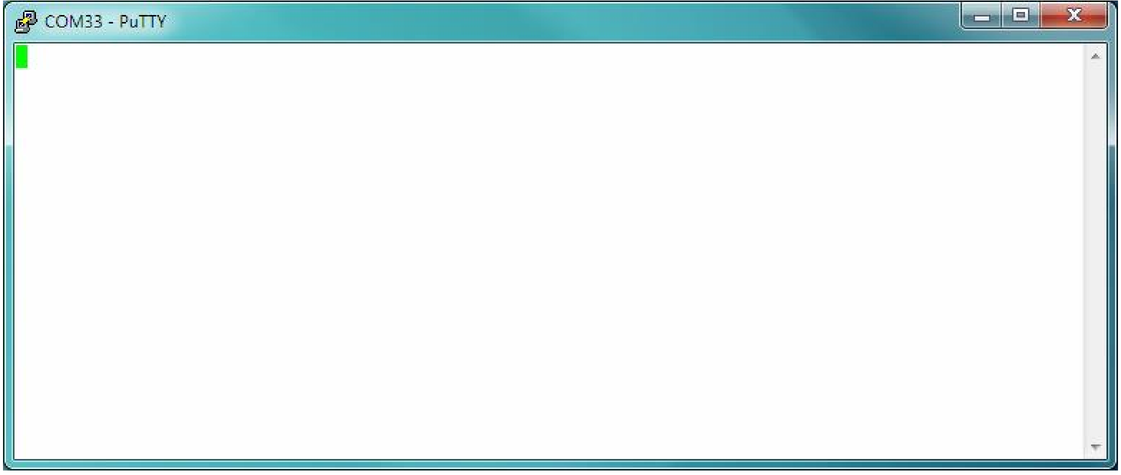
### USB Driver Setup[1]

1. Login to your PC as administrator
2. Put the CD into your disk drive
3. Connect your PC with UCC 1
4. Open the device manager
5. Navigate to: COM & LPT -> Universal CAN Controller and right click, properties
   - Chose the driver tab and click refresh driver
   - Search for driver software manually
   - Navigate to the root folder of the setup CD and then click o.k.

Now the UCC 1 should appear as Universal CAN Controller within the COM port list with a port number associated to it and is ready for use.

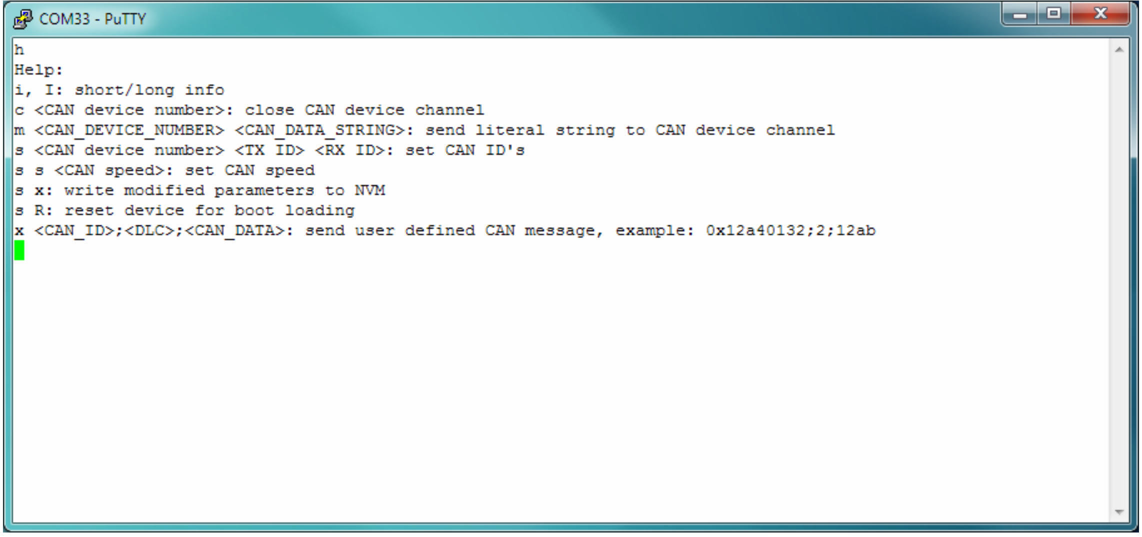
## Hardware

Start your preferred terminal software and connect to the COM port associated to the UCC 1. A blank terminal window should open.



[1] Only required for Microsoft Operating systems. For UNIX like operating systems consult your operating systems manual.

Enter ȟòto test the connection, the response should look like this:



```
h
Help:
i, I: short/long info
c <CAN device number>: close CAN device channel
m <CAN_DEVICE_NUMBER> <CAN_DATA_STRING>: send literal string to CAN device channel
s <CAN device number> <TX ID> <RX ID>: set CAN ID's
s s <CAN speed>: set CAN speed
s x: write modified parameters to NVM
s R: reset device for boot loading
x <CAN_ID>;<DLC>;<CAN_DATA>: send user defined CAN message, example: 0x12a40132;2;12ab
```

Now you may set up CAN speed and CAN identifiers as required

# Operation

The UCC 1 device distinguishes between its own, device internal command set and commands directed to the UUTös. All Command mnemonics may be entered in upper or lower case.

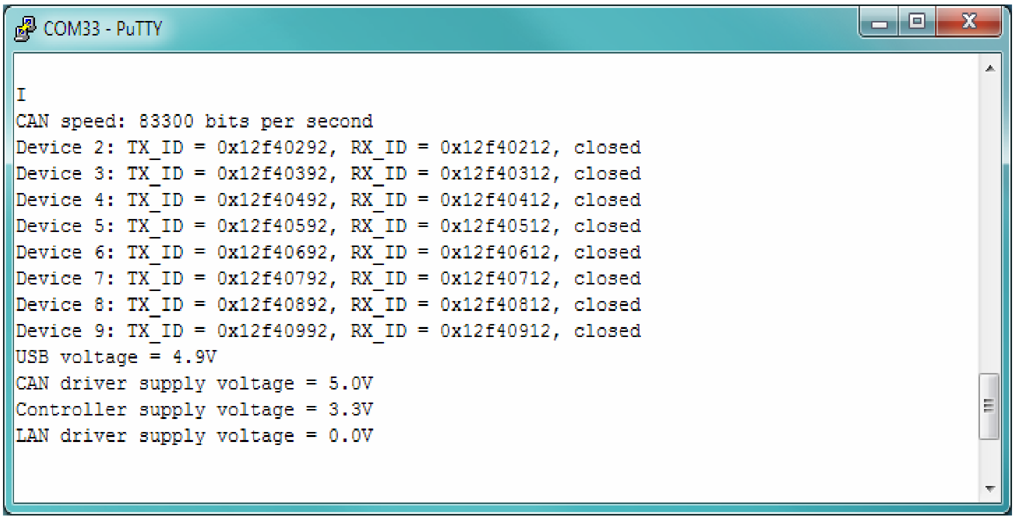
## UCC 1 Command Set

The command parts and parameters are separated by space characters.
Command syntax: command mnemonic <SPACE><optional parameters>

H, h: Help command. Displays a list of the available commands

i: Short device information

I: Detailed device information including CAN bus speed, CAN identifiers, device channel information and UCC 1 internal values



```
I
CAN speed: 83300 bits per second
Device 2: TX_ID = 0x12f40292, RX_ID = 0x12f40212, closed
Device 3: TX_ID = 0x12f40392, RX_ID = 0x12f40312, closed
Device 4: TX_ID = 0x12f40492, RX_ID = 0x12f40412, closed
Device 5: TX_ID = 0x12f40592, RX_ID = 0x12f40512, closed
Device 6: TX_ID = 0x12f40692, RX_ID = 0x12f40612, closed
Device 7: TX_ID = 0x12f40792, RX_ID = 0x12f40712, closed
Device 8: TX_ID = 0x12f40892, RX_ID = 0x12f40812, closed
Device 9: TX_ID = 0x12f40992, RX_ID = 0x12f40912, closed
USB voltage = 4.9V
CAN driver supply voltage = 5.0V
Controller supply voltage = 3.3V
LAN driver supply voltage = 0.0V
```

c <CAN_DEVICE_NUMBER> close open device channel and stop communication with the UUT connected to this channel.
Example: c 2, closes the communication channel to CAN device #2

m <CAN_DEVICE_NUMBER> <CAN_DATA_STRING>: send the content of CAN_DATA_STRING to the UUT connected to channel number CAN_DEVICE_NUMBER. This command is useful to communicate with UUTös which are not following the MICBAC command syntax completely. The string content is sent without modification except that multiple consecutive spaces are replaced by single spaces.
Example: m 3 H UUT_DEVICE_NAME PARAMETER_1

***Note: This command distinguishes between upper and lower case characters, therefore the UUT dependent command must be entered exactly as required by the UUT.***

s <CAN_DEVICE_NUMBER> <TX_ID> <RX_ID>: set the CAN identifiers for a dedicated CAN device number. The CAN identifiers are entered as HEX code and stored temporarily[2].
Example: s 7 0x13g50192 0x13g50112

s s <CAN_SPEED>: set the CAN speed.
Example: s s 83300 sets the CAN bit rate to 83.3kbit per second.

***Note: All parameter settings or modifications are temporary. To make them permanent use the following command***

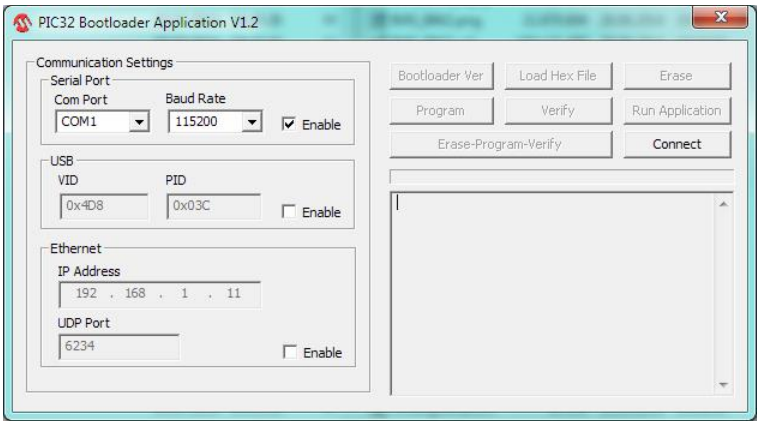
s x: write all modified parameters to permanent memory.

s R: reset device for boot loading. Note the required upper case ŕRò

[2] Temporarily means as long as power is supplied to the UCC 1.

# Firmware Update

1. Connect to the UCC 1 as described above.
2. In the terminal window enter: s R for device reset.
3. Close the terminal window.
4. Start the boot loader software. A window similar to the one below should open.



PIC32 Bootloader Application V1.2

Communication Settings

Serial Port: Com Port COM1, Baud Rate 115200, ☑ Enable

USB: VID 0x4D8, PID 0x03C, ☐ Enable

Ethernet: IP Address 192 . 168 . 1 . 11, UDP Port 6234, ☐ Enable

5. Check the enable check box within the USB settings.
6. Click the connect button.
7. Load the hex file you want to program to the UCC 1.
8. Click ñEraseòand then ñProgramò.
9. Click ñRun Applicationòand then ñDisconnectò, close the boot loader application.
10. Remove the USB cable and connect it again.

***Note: The UCC 1 device should be now accessible by a terminal program as described above.***

# Scope of Delivery

- UCC 1: 1 piece
- USB cable: 1 piece
- CD containing USB setup file (ucc.inf for Microsoft Windows operating systems), operating manual and boot loading software.

# Specification

| | |
|---|---|
| CAN bus rate: | 20kbit per second up to 1Mbit per second |
| CAN identifier: | extended |
| Up to eight MICBAC channels | |
| Ambient temperature, operation: | -40°C to +55°C |
| Ambient temperature, storage: | -55°C to +80°C |
| Power supply: | USB bus powered |
| Current consumption, USB unconfigured: | < 100mA |
| Current consumption, USB configured: | < 500mA |
| Galvanically isolated CAN interface: | Isolation voltage: 500V |
| Length: | 90mm |
| Width: | 110mm |
| Height: | 50mm |
| Weigth: | 215 g |

# Acronyms and Abbreviations

| | |
|---|---|
| MICBAC: | Micro System Bus Access Channel |
| CAN: | Controller Area Network |
| CD: | Compact Disc |
| PC: | Personal Computer |
| LAN: | Local Area Network |

# Warranty

The UCC 1 device is warranted under normal usage against defects in workmanship and materials to the original purchaser for two years from the date of purchase.
Transportation, handling and insurance charges are excluded.
To claim warranty please return the device carefully packed together with the bill of sale and a description of the defect.

***Note: The warranty is limited to the UCC 1 device and does not cover any damages on equipment connected to it.***
***This warranty is void if faults are caused by mishandling, shock, over voltage, high temperatures or modifications.***

Return the device to:

**ge-consulting GmbH**
**D-75305 Neuenbürg**
**Wiesenweg 25**