voice INTER connect

# Dokumentation vicCONTROL go

Revision: 18

Date:
2023-05-03

Ref. number: 1-0010-01-02-18

# Document history

| Revision | Changes | Date |
|---|---|---|
| 1 | Documentation of *vicCONTROL go* | 2019-09-03 |
| 2 | More distinctive documentation for digital I/Os | 2019-09-06 |
| 3 | Error correction and formatting | 2019-09-24 |
| 4 | Document dependency on “Visual C++ Redistributable for Visual Studio 2012” | 2019-10-02 |
| 5 | Update designer installation instructions | 2019-10-08 |
| 6 | Corrections ”Custom serial output” | 2020-03-04 |
| 7 | CE | 2020-03-11 |
| 8 | Particularise Power plug, update troubleshooting | 2020-05-11 |
| 9 | Error correction | 2020-05-27 |
| 10 | Dialog design and firmware update process | 2020-09-25 |
| 11 | vicCORE-3_2 audio connector description | 2021-01-20 |
| 12 | Hardware | 2021-02-02 |
| 13 | Error corrections | 2021-05-26 |
| 14 | Error corrections | 2021-08-16 |
| 15 | Clarifications, updated trouble shooting section and error corrections | 2021-09-17 |
| 16 | Update possible vendor for Java runtime | 2022-09-22 |
| 17 | Error corrections vicCORE-3 X105,X106,X126,X128 | 2023-03-14 |
| 18 | vicBASE-3_5 schematic, Designer setup simplified | 2023-05-03 |

# Contents

# 1 Overview

This document describes the speech recognition system *vicCONTROL go* of *voice INTER connect GmbH*.

## 1.1 Scope of delivery

***vicCONTROL go kit*** is delivered with the following components:

- evaluation board vicBASE-3_5 incl. vicCORE-3_2
- Micro-USB cable
- USB stick with software and documentation
- Quick Start Guide

***vicCONTROL go stamp*** is delivered with the following components:

- core module vicCORE-3_2

The USB stick contains:

- full documentation
- example dialogs and audio prompts
- software
  - CRCCalc (tool for generating checksums for serial communication)
  - FirmwareUpdate (firmware update and recovery tool)
  - USBDriver (USB driver for vicBASE-3_5)
  - vicCONTROL Designer (individual speech dialog generation for *vicCONTROL go*)
  - wavesurfer (audio editor)

## 1.2 Speech recognition and key features

- stand-alone speech recognition on the hardware platform
- configuration without programming skills using vicCONTROL Designer
- extensive choice for the platform language
- arbitrary vocabulary for command and control
- speaker-independent; no training necessary
- up to 10 programmable digital inputs and outputs
- analog audio input and output (available on the evaluation board: line-in/out, microphone, headphone and speaker connectors)

- serial communication via USB/UART
- audio output from prompt files

## 1.3 Safety Note

Speech recognition allows additional control options and works by employing statistical methods. To achieve this, it must adapt to constantly changing environmental conditions. It can not be guaranteed that a phrase is processed without delay or recognized correctly or that it is recognized at all. Similarly, sudden speech recognition errors may occur in which a spoken phrase is recognized, although it has not been uttered. *vicCONTROL go* must therefore not be used alone or without independent safety measures if the application is safety-critical.

## 1.4 Electro Magnetic Conformity

Declaration of Electro Magnetic Conformity of „vicCONTROL go“ product

**CE**

„vicCONTROL go“-voice control (henceforth products) are designed for installation in electrical appliances or as dedicated Evaluation Boards (i.e.: for use as a test and prototype platform for hardware/software development) in laboratory environments.

„vicCONTROL go“-products without protective housing are ESD-sensitive and must therefore be unpacked and handled or processed at ESD protected workplaces by trained personnel. Furthermore, ”vicCONTROL go” products should not be operated without a protective circuit if the connections to the contacts of the product are longer than 3 m.

„vicCONTROL go“-products fulfill the norms of the European Union’s Directive for Electro Magnetic Conformity in accordance with the descriptions and rules of usage indicated in this hardware manual (particularly in respect to the contacts, connectors and and interfaces to a host-PC or any host-processor).

Implementation of „vicCONTROL go“-products into target devices, as well as user modifications and extensions of „vicCONTROL go“-products, is subject to renewed establishment of conformity to and certification of Electro Magnetic Directives. Users should ensure conformity following any modifications to a product as well as implementation of a product into target systems.

## 1.5 Editions

***vicCONTROL go kit***

The standard edition consists of the vicBASE-3_5 baseboard (Figure 1), which serves as an evaluation platform for the *vicCONTROL go stamp*. Its comprehensive features allow an easy creation and testing of voice control solutions. It is suitable for evaluating your own speech recognition application as well as for the realization of prototypes and small series.

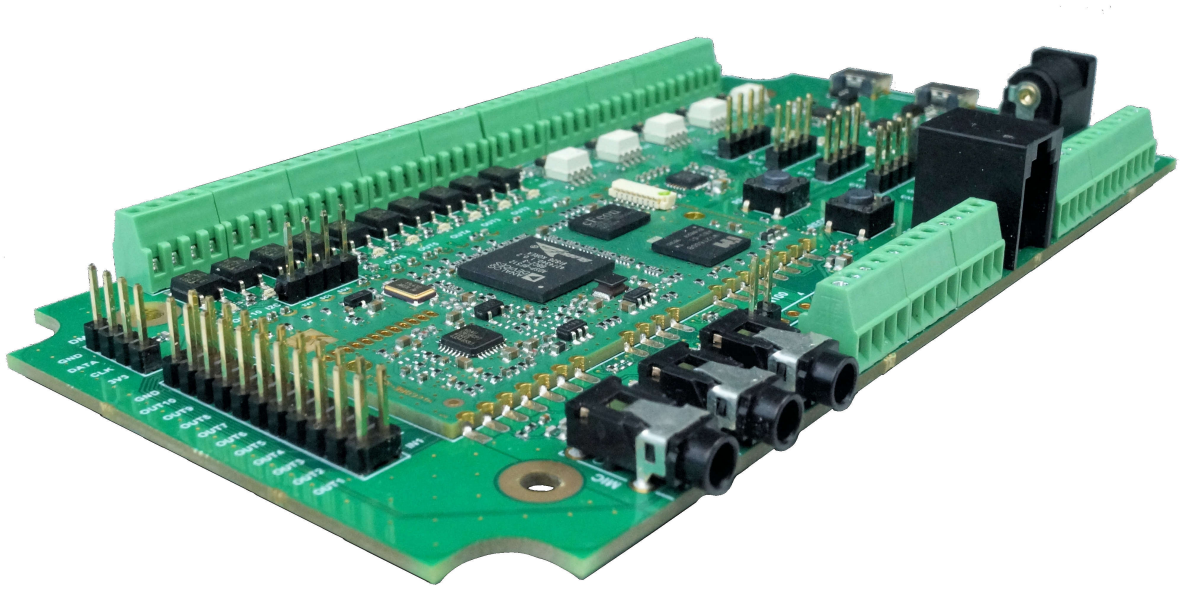


**Figure 1:** vicBASE-3_5

The corresponding block diagram is illustrated in Figure 2.

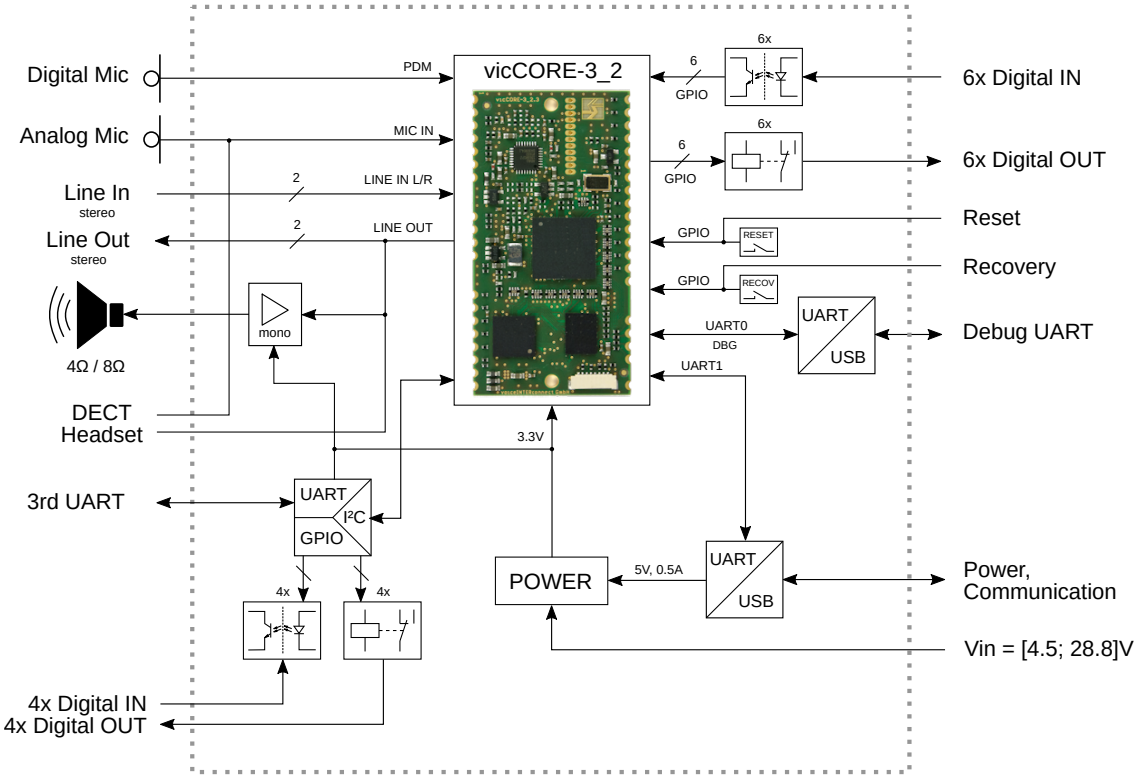



**Figure 2:** Block diagram of vicBASE-3_5

### *vicCONTROL go stamp*

The *vicCONTROL go stamp* edition uses the vicCORE-3_2 module. Thanks to its edge contacts, it can be integrated easily into own circuits.

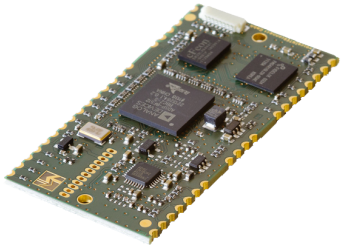


**Figure 3:** vicCORE-3_2

Please contact us for further product versions.

# 2 Hardware

## 2.1 vicCORE-3_2

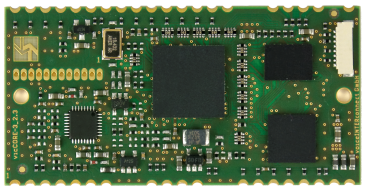


**Figure 4:** vicCORE-3_2

### 2.1.1 Version

- Designator: vicCORE-3_2
- Board version: 2.3

### 2.1.2 Operating Conditions

**Absolute Maximum Ratings**

| Symbol | Name | Wert | Unit |
|---|---|---|---|
| $T_{STORE}$ | Storage temperature range | −65 …120 | °C |
| $V_{CP}$ | Working temperature range | −40 …85 | °C |
| $FC$ | UL94 flammability class | V-0 | |
| $V_{DD}$ | Operating voltage | 2.7 …3.6 | V |

**Operating Conditions**

| Symbol | Name | Min | Nom | Max | Einheit |
|---|---|---|---|---|---|
| $V_{DD}$ | Operating voltage | 3.0 | 3.3 | 3.5 | V |
| $I_{DD}$ | Current consumption | 80 | | 110 | mA |
| $U_{MIC}$ | Voltage on microphone input MIC_IN_R (at V MIC = 20 dB) | 0 | 30 | 100 | $mV_{RMS}$ |
| $U_{IN}$ | Voltage on LINE_IN_R und LINE_IN_L | 0 | 1 | 1 | $V_{RMS}$ |
| $U_{OUT}$ | Voltage on LINE_OUT_R und LINE_OUT_L | 0 | 1 | 1 | $V_{RMS}$ |
| $U_{RESET}$ | Voltage on Reset input | 0 | 3.3 | $V_{DD}$ | V |

### 2.1.3 Connections

#### Audio inputs

The input impedance is defined by the inner circuit of the vicCORE-3_2 (see Figure 5). The amplification-dependent input resistors of the audio codec are a significant factor. The association between amplification and input resistors is reflected in the table below. The maximum DC offset is 16 V. At the moment the amplification is set rigidly to 35 dB; hence, the input impedance is 2 kiloohms.

For use with electret microphone the MIC_BIAS connector does provide phantom power. A 2.2 kiloohms series resistor is already equipped on the vicCORE-3_2 board for the MIC_BIAS connector, which means it can get connected directly to MIC_IN_R to be used with an electret microphone.

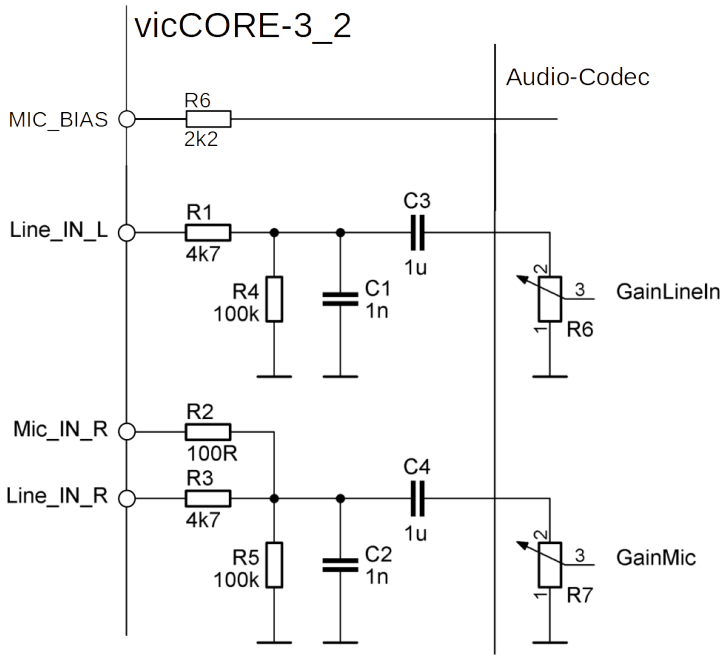



**Figure 5:** Audio inputs of vicCORE-3_2
The depicted components are already equipped at the vicCORE-3_2 board

Input R6/R7

| V in dB | $R_{in}$ k |
|---|---|
| −12 | 85 |
| 0 | 53 |
| 25 | 5.6 |
| 35 | 2.0 |

**Audio outputs vicCORE-3_2**

The output impedance is determined by the low-resistance output resistor of the audio codec, which is not specified in the data sheet. The maximum DC offset is 16 V.

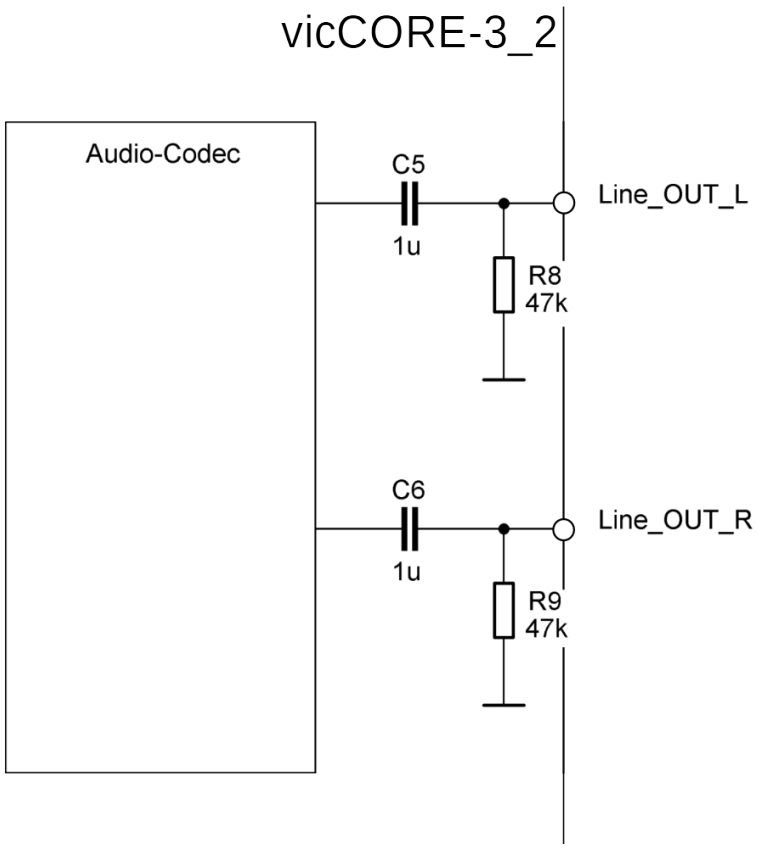



**Figure 6:** Audio outputs of vicCORE-3_2
The depicted components are already equipped at the vicCORE-3_2 board

**Edge contacts**

All connections of the vicCORE-3_2 are implemented as edge contacts for directly soldering the module onto a baseboard (X101–X146; see Figure 7). Alternatively, it is also possible to connect a ribbon connector or a header (see CON101 in Figure 8).

All Inputs (IN_1 to IN_6, RECOV, and RESET) are low active.

The pull-up/down termination is realized by 47-kiloohm resistors.

Firmware updates and the connection to vicCONTROL Designer are carried out via the command interface (RX1/TX1)

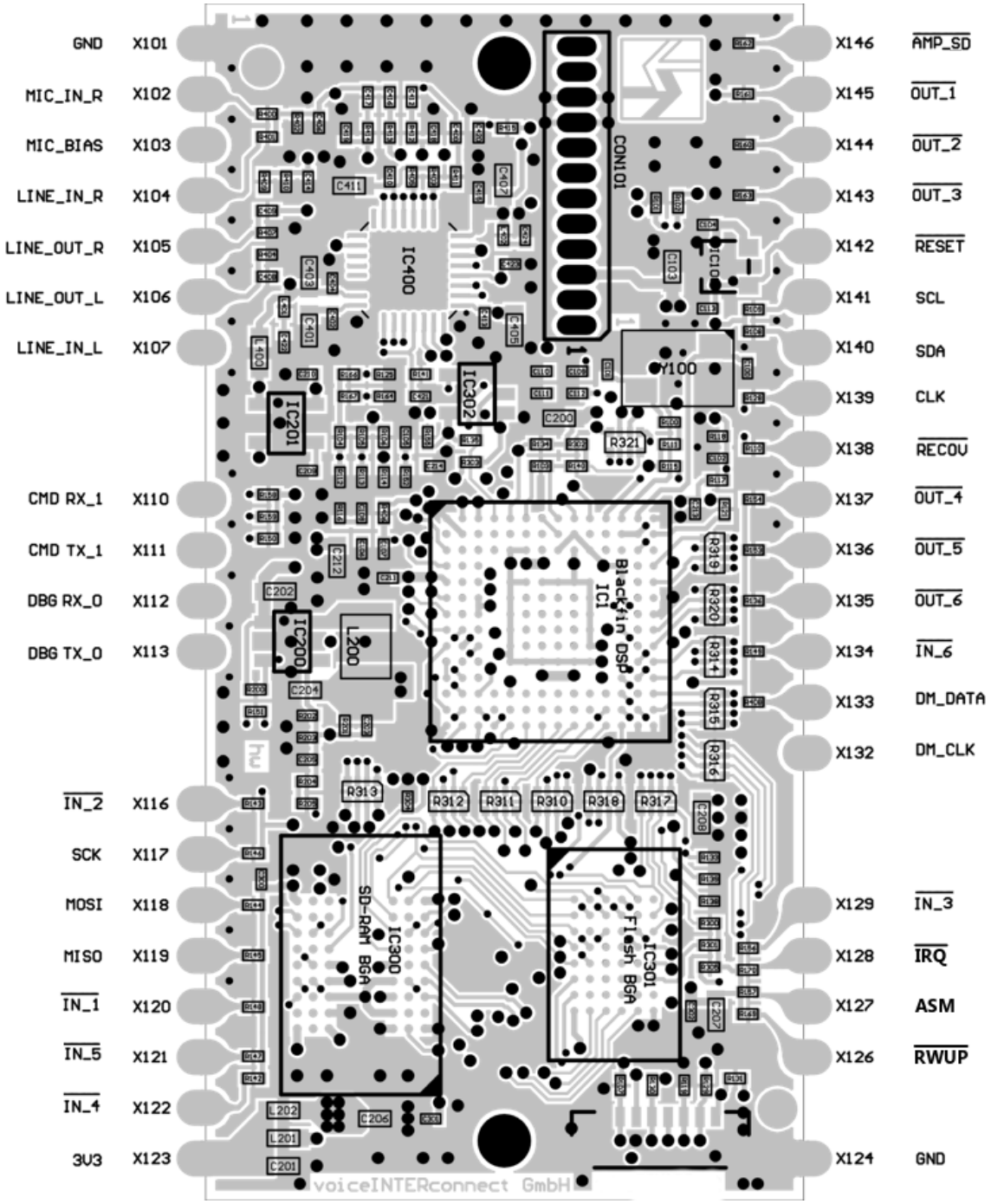



**Figure 7:** Edge connectors

| Connection | Type | Term. | Name | Description |
|---|---|---|---|---|
| X101 | Input | | GND | ground for audio connectors X102–X107 |
| X102 | Input | | MIC_IN_R | Microphone input (+) for condenser microphone |

| | | | | |
|---|---|---|---|---|
| X103 | Output | | MIC_BIAS | Bias voltage microphone (3 V) A 2.2 kiloohms series resistor is already equipped at the vicCORE-3_2 board. This pin can be connected directly to MIC_IN_R for electret microphone) |
| X104 | Input | | LINE_IN_R | Alternative line level input |
| X105 | Output | | LINE_OUT_R | Audio signal output of current input signal |
| X106 | Output | | LINE_OUT_L | Audio signal output of prompts |
| X107 | Input | | LINE_IN_L | Not used |
| X110 | Input | Pull-up | RX_1 | UART-1-RX-Signal command interface |
| X111 | Output | Pull-up | TX_1 | UART-1-TX-Signal command interface |
| X112 | Input | Pull-up | RX_0 | UART-0-RX-Signal debug interface |
| X113 | Output | Pull-up | TX_0 | UART-0-TX-Signal debug interface |
| X116 | Input | Pull-up | IN_2 | Input 2 |
| X117 | Output | Pull-up | SCK | Internal usage |
| X118 | Output | Pull-up | MOSI | Internal usage |
| X119 | Input | Pull-up | MISO | Internal usage |
| X120 | Input | Pull-up | IN_1 | Input 1 |
| X121 | Input | Pull-up | IN_5 | Input 5 |
| X122 | Input | Pull-up | IN_4 | Input 4 |
| X123 | Input | | 3V3 | Supply voltage (+) |
| X124 | Input | | GND | Supply voltage (ground) |
| X126 | In-/Output | Pull-up | RWUP | Internal usage |
| X127 | In-/Output | Pull-up | ASM | Internal usage |
| X128 | In-/Output | Pull-up | IRQ | Internal usage |
| X129 | Input | Pull-up | IN_3 | Input 3 |
| X132 | Output | | DM_CLK | Internal usage |
| X133 | Input | Pull-up | DM_DATA | Internal usage |
| X134 | Input | Pull-up | IN_6 | Input 6 |

| | | | | |
|---|---|---|---|---|
| X135 | Output | Pull-up | OUT_6 | Output 6 |
| X136 | Output | Pull-up | OUT_5 | Output 5 |
| X137 | Output | Pull-up | OUT_4 | Output 4 |
| X138 | Input | Pull-up | RECOV | Recovery configuration |
| X139 | Output | | CLK | Clock 18.432 MHz |
| X140 | In-/Output | Pull-up | SDA | I2C |
| X141 | In-/Output | Pull-up | SCL | I2C |
| X142 | Input | Pull-up | RESET | Hardware-Reset signal |
| X143 | Output | Pull-down | OUT_3 | Output 3 |
| X144 | Output | Pull-down | OUT_2 | Output 2 |
| X145 | Output | Pull-up | OUT_1 | Output 1 |
| X146 | Output | Pull-down | AMP_SD | Indicates audio output activity (amplifier shut down) |

**Ribbon connector**

The 12-pin ribbon connection is shown in Figure 8. The numeration of pins starts from the center of the board with "1". A 1.27 mm raster is used.

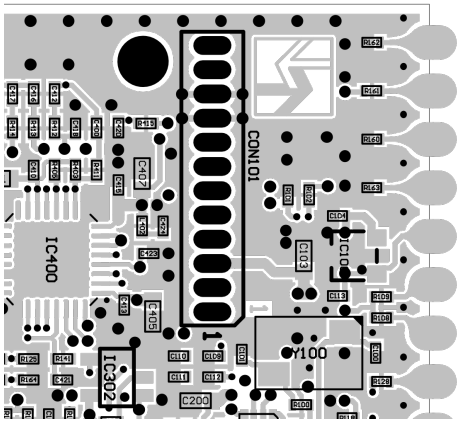



**Figure 8:** Ribbon connector

| Connection | Type | Name | Explanation |
|---|---|---|---|
| Pin 1 | In-/Output | SDA | I2C |
| Pin 2 | In-/Output | SCL | I2C |
| Pin 3 | Input | 3V3 | Supply voltage |
| Pin 4 | Input | LINE_IN_L | Optional microphone input (line level) |
| Pin 5 | Output | LINE_OUT_L | Audio signal output of prompts |
| Pin 6 | Output | LINE_OUT_R | Audio signal output of current input signal |
| Pin 7 | Output | RX_1 | UART-1-RX-Signal command interface |
| Pin 8 | Input | TX_1 | UART-1-TX-Signal command interface |
| Pin 9 | Input | GND | Ground for supply voltage |
| Pin 10 | Input | GND | Ground for audio connectors |
| Pin 11 | Output | MIC_BIAS | Bias voltage for condenser microphone (3.3 V; 2.2 kiloohm) |
| Pin 12 | Input | MIC_In_R | Microphone input (+) for condenser microphone |

### 2.1.4 Dimensions

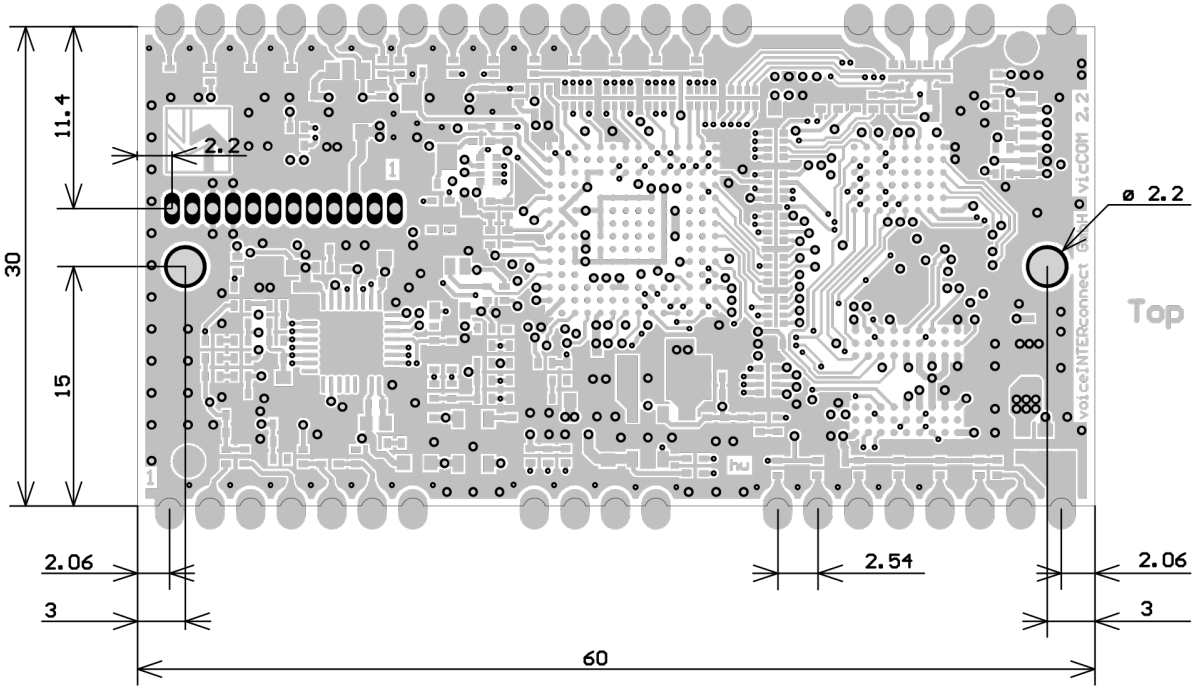



**Figure 9:** Dimensions (in mm)

### 2.1.5 Recommended footprint

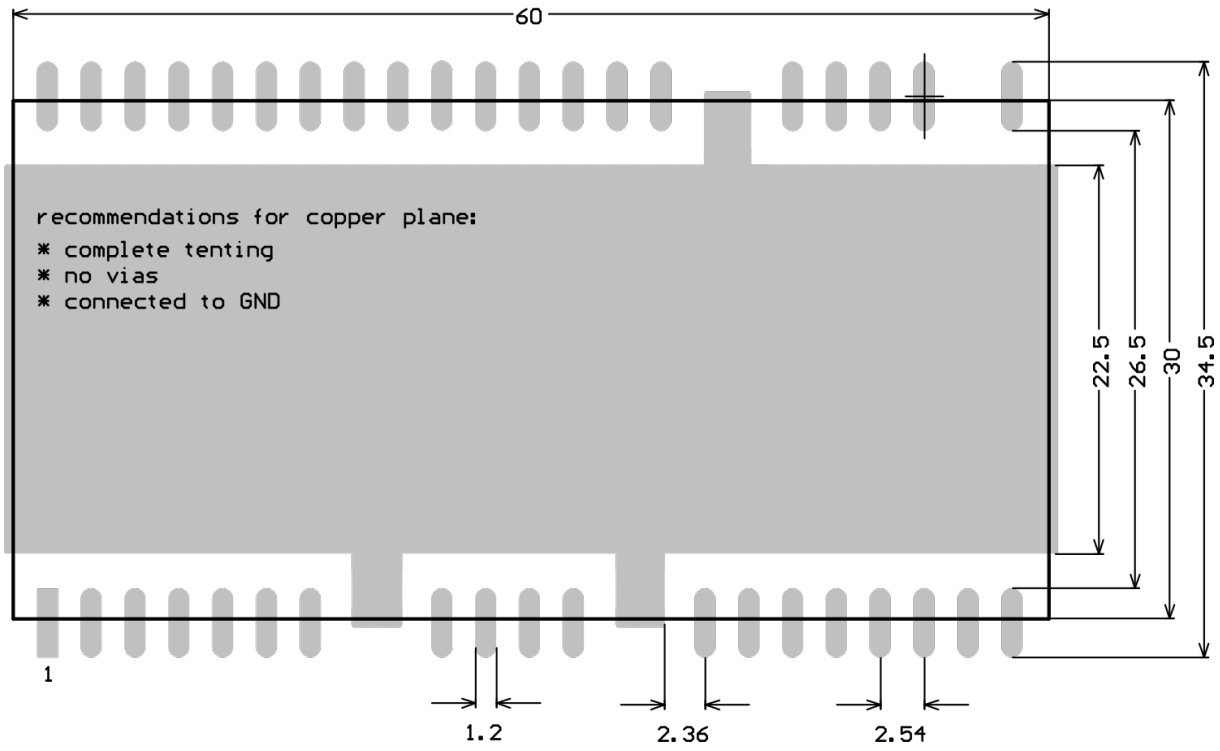



**Figure 10:** Recommended footprint (all dimensions in mm)

### 2.1.6 Reflow profile

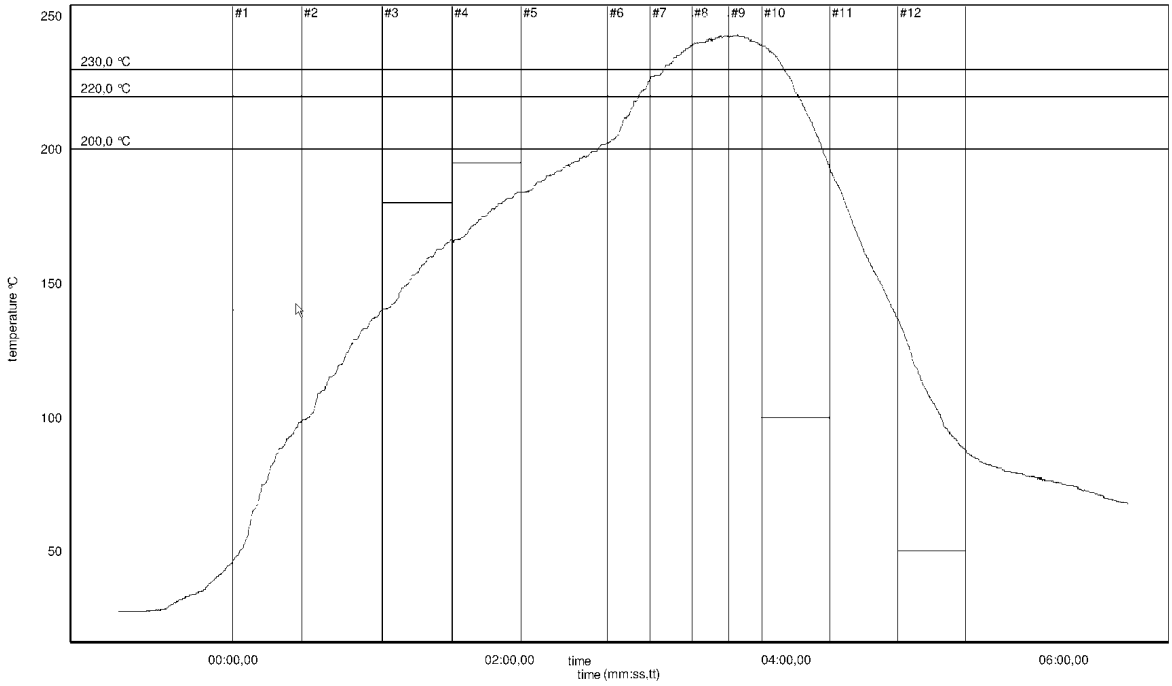



**Figure 11:** Reflow profile vicCORE-3_2

## 2.2 vicBASE-3_5

The baseboard for the vicCORE-3_2 offers power and communication over USB, an extensive audio interface, and connectors for 10 digital inputs and 10 digital outputs as well.

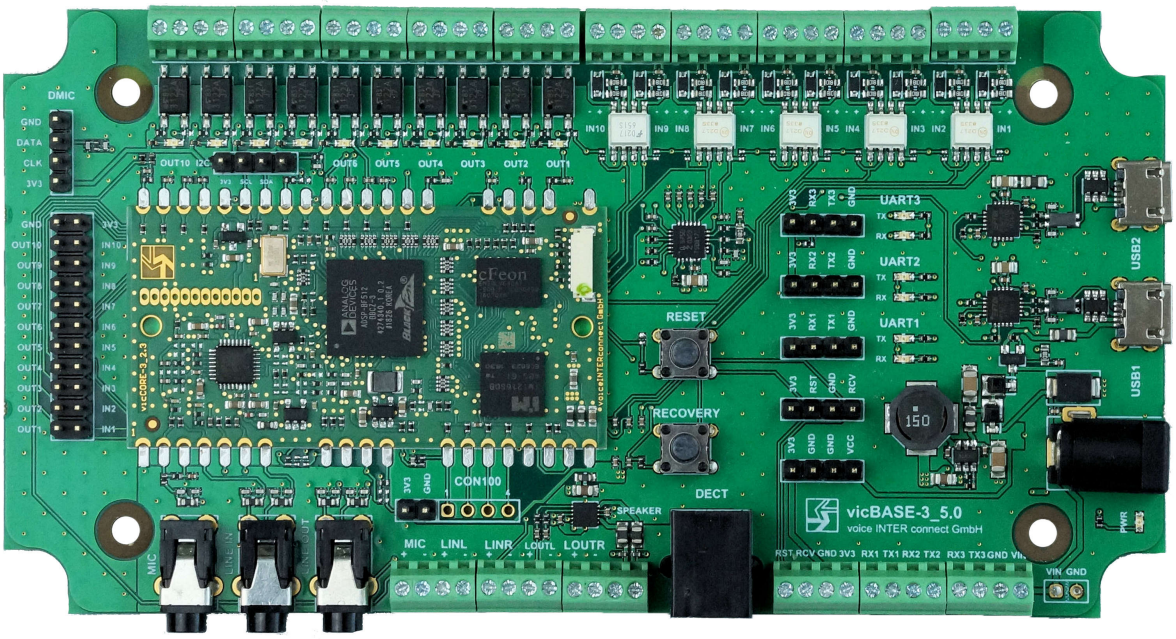



**Figure 12:** vicBASE-3_5

### 2.2.1 Version

- Designator: vicBASE-3_5
- Board version: 5.1

### 2.2.2 Operating Conditions

| Symbol | Name | Min | Nom | Max | Unit |
|---|---|---|---|---|---|
| $V_{CC}$ | input power supply | 4.5 | 24 | 28 | V |
| $P_{in}$ | power consumption, stand-by, outputs inactive | | 0.35 | | W |
| | power conumption, all outputs acitve | | 0.7 | | W |
| | power conumption, all outputs acitve, sinus signal output ($R_L$ = 8 ohms), volume 100 % | | 2.3 | | W |
| | power conumption, all outputs acitve, sinus signal output ($R_L$ = 4 ohms), volume 100 % | | 3.7 | | W |
| $R_L$ | loudspeaker impedance | 4 | 8 | | |
| $P_{SPK}$ | power loudspeaker output ($R_L$ = 8 ohms) | | | 0.6 | W |
| | power loudspeaker output ($R_L$ = 4 ohms) | | | 1.1 | W |
| $V_{LineIn}$ | voltage at Line_IN | | 1 | 1 | $V_{RMS}$ |
| $V_{LineOut}$ | voltage at Line_OUT | | 1 | 1 | $V_{RMS}$ |
| $V_{Mic}$ | voltage at microphone input | | 30 | 100 | $mV_{RMS}$ |
| $V_{In}$ | voltage for electrically isolated digital inputs (optocoupler) | 3.3 | | 12 | V |
| $V_{Rel}$ | rated voltage at digital outputs (off-state) | | | 60* | V |
| | recomended voltage at digital outputs (off-state) | | | 48 | V |
| $I_{Rel}$ | current through digital outputs (on-state) | | | 1.4 | A |
| $I_{3V3}$ | current availability of 3.3-V-Output | | | 500 | mA |

* Operating the digital outputs at maximum rated voltage continously may cause this product to decrease in its reliability.

### 2.2.3 Connections and controls

The position and name of the connections and buttons can be seen in Figure 13.

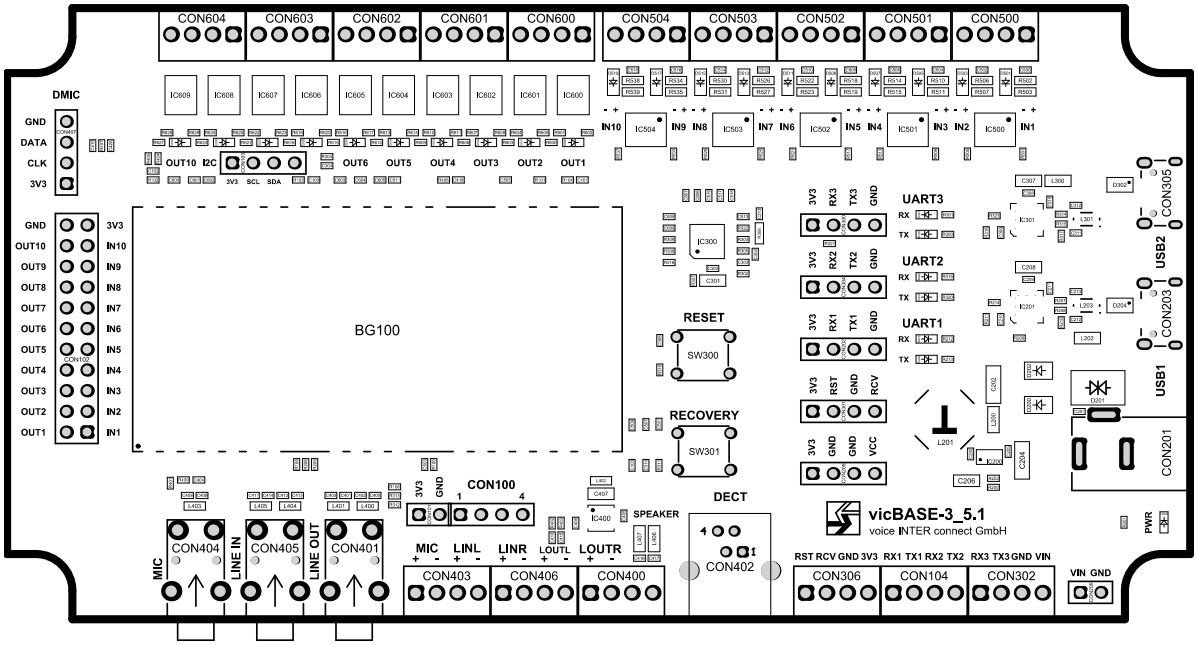



**Figure 13:** vicBASE-3_5 connections

| Connector | Type | Designator | vicCORE-3_2 connection | Description |
|---|---|---|---|---|
| CON100 | pin header, 4pos. | CON100_1 | IN_2 | Input 2 (configurable as SPISSEL, if R100 is removed) |
| | | CON100_2 | SCK | SPI Clock (not used) |
| | | CON100_3 | MOSI | SPI Master Out Slave In (not used) |
| | | CON100_4 | MISO | SPI Master In Slave Out (not used) |
| CON101 | pin header, 2pos. | 3V3 | 3V3 | 3.3 V supply voltage (output) |
| | | GND | GND | ground |
| CON102 | pin header, 22pos. | OUT1 | OUT_1 | Output 1 |
| | | OUT2 | OUT_2 | Output 2 |
| | | OUT3 | OUT_3 | Output 3 |
| | | OUT4 | OUT_4 | Output 4 |
| | | OUT5 | OUT_5 | Output 5 |
| | | OUT6 | OUT_6 | Output 6 |

| | | | | |
|---|---|---|---|---|
| | | OUT7 | I2C | Output 7 |
| | | OUT8 | I2C | Output 8 |
| | | OUT9 | I2C | Output 9 |
| | | OUT10 | I2C | Output 10 |
| | | IN1 | IN_1 | Input 1 |
| | | IN2 | IN_2 | Input 2 |
| | | IN3 | IN_3 | Input 3 |
| | | IN4 | IN_4 | Input 4 |
| | | IN5 | IN_5 | Input 5 |
| | | IN6 | IN_6 | Input 6 |
| | | IN7 | I2C | Input 7 |
| | | IN8 | I2C | Input 8 |
| | | IN9 | I2C | Input 9 |
| | | IN10 | I2C | Input 10 |
| | | GND | GND | ground |
| | | 3V3 | 3V3 | 3.3 V supply voltage (output) |
| CON103 | pin header, 4pos. | I2C_3V3 | 3V3 | 3.3 V supply voltage (output) |
| | | I2C_SCL | SCL | I2C Clock (not used) |
| | | I2C_SDA | SDA | I2C Data (not used) |
| | | I2C_GND | GND | ground |
| CON104 | screw terminal | RX1 | RX_1 | UART1 receive line for command interface |
| | | TX1 | TX_1 | UART1 transmit line for command interface |
| | | RX2 | RX_0 | UART2 receive line for debug interface |
| | | TX2 | TX_0 | UART2 transmit line for debug interface |
| CON200 | pin header, 2pos. | VIN | | input power supply |
| | | GND | GND | ground |
| CON201 | power jack (6.4/2.5 mm) | VIN | | input power supply for 5.5/2.5 mm plugs, ground at outer contact |
| CON202 | pin header, 4pos. | 3V3 | 3V3 | 3.3 V supply voltage (output) |

| | | | | |
|---|---|---|---|---|
| | | RX1 | RX_1 | UART1 receive line for command interface |
| | | TX1 | TX_1 | UART1 transmit line for command interface |
| | | GND | GND | ground |
| CON203 | Micro-USB port | USB1 | RX_1/TX_1 | power supply and communication with command interface via USB |
| CON205 | pin header, 4pos. | 3V3 | 3V3 | 3.3 V supply voltage (output) |
| | | GND | GND | ground |
| | | GND | GND | ground |
| | | VCC | | protected and filtered input power supply |
| CON300 | pin header, 4pos. | 3V3 | 3V3 | 3.3 V supply voltage (output) |
| | | RX3 | | UART3 receive line (not used) |
| | | TX3 | | UART3 transmit line (not used) |
| | | GND | GND | ground |
| CON301 | pin header, 4pos. | 3V3 | 3V3 | 3.3 V supply voltage (output) |
| | | RST | RESET | Hardware reset (low active) |
| | | GND | GND | ground |
| | | RCV | RECOV | Recovery configuration (low active) |
| CON302 | screw terminal | RX3 | | UART3 receive line (not used) |
| | | TX3 | | UART3 transmit line (not used) |
| | | GND | GND | ground |
| | | VIN | | input power supply |
| CON304 | pin header, 4pos. | 3V3 | | 3.3 V supply voltage (output) |
| | | RX2 | RX_0 | UART2 receive line for debug interface |
| | | TX2 | TX_0 | UART2 transmit line for debug interface |
| | | GND | GND | ground |
| CON305 | Micro-USB port | USB2 | RX_0/TX_0 | communication with debug interface via USB |
| CON306 | screw terminal | RST | RESET | Hardware reset (low active) |

| | | | | |
|---|---|---|---|---|
| | | RCV | RECOV | Recovery configuration (low active) |
| | | GND | GND | ground |
| | | 3V3 | 3V3 | 3.3 V supply voltage (output) |
| CON400 | screw terminal | LOUTR | LINE_OUT_R | audio output of current input signal |
| | | SPEAKER | | audio output with 12 dB gain of LINE_OUT_L signal |
| CON401 | 3.5 mm audio jack | LINEOUT | LINE_OUT_R/L | low impedance Line-Out for output of prompts (L) and for current input signal (R) |
| CON402 | 4P4C socket | DECT | MIC_IN_R, LINE_OUT_L | connector for DECT-Headset |
| CON403 | screw terminal | MIC | MIC_IN_R | microphone input for electret microphones |
| | | LINL | LINE_IN_L | high impedance Line-In for input of analoge audio signals |
| CON404 | 3.5 mm audio jack | MIC | MIC_IN_R | microphone input for electret microphones |
| CON405 | 3.5 mm audio jack | LINEIN | LINE_IN_R/L | high impedance Line-In for input of analoge audio signals (L) |
| CON406 | screw terminal | LINR | LINE_IN_R | not used |
| | | LOUTL | LINE_OUT_L | output of prompts |
| CON407 | pin header, 4pos. | DMIC_GND | GND | ground |
| | | DMIC_DATA | DM_DATA | digital microphone, data line (not used) |
| | | DMIC_CLK | DM_CLK | digital microphone, clock (not used) |
| | | DMIC_3V3 | 3V3 | 3.3 V supply voltage (output) |
| CON500 | screw terminal | IN1 | IN_1 | Input 1 (floating) |
| | | IN2 | IN_2 | Input 2 (floating) |
| CON501 | screw terminal | IN3 | IN_3 | Input 3 (floating) |
| | | IN4 | IN_4 | Input 4 (floating) |
| CON502 | screw terminal | IN5 | IN_5 | Input 5 (floating) |

| | | | | |
|---|---|---|---|---|
| | | IN6 | IN_6 | Input 6 (floating) |
| CON503 | screw terminal | IN7 | I2C | Input 7 (floating) |
| | | IN8 | I2C | Input 8 (floating) |
| CON504 | screw terminal | IN9 | I2C | Input 9 (floating) |
| | | IN10 | I2C | Input 10 (floating) |
| CON600 | screw terminal | OUT1 | OUT_1 | Output 1 (floating) |
| | | OUT2 | OUT_2 | Output 2 (floating) |
| CON601 | screw terminal | OUT3 | OUT_3 | Output 3 (floating) |
| | | OUT4 | OUT_4 | Output 4 (floating) |
| CON602 | screw terminal | OUT5 | OUT_5 | Output 5 (floating) |
| | | OUT6 | OUT_6 | Output 6 (floating) |
| CON603 | screw terminal | OUT7 | I2C | Output 7 (floating) |
| | | OUT8 | I2C | Output 8 (floating) |
| CON604 | screw terminal | OUT9 | I2C | Output 9 (floating) |
| | | OUT10 | I2C | Output 10 (floating) |
| SW300 | Button | RESET | RESET | Hardware reset (low active) |
| SW301 | Button | RECOVERY | RECOV | Recovery configuration (low active) |

There are 2 Micro-USB ports available. In addition to the communication with the vicBASE-3_5, USB1 can also be used as the sole energy source, if an 8-ohm loudspeaker is used. The specified current availability of USB-2.0 of 500 mA is not sufficient for use with a 4-ohm loudspeaker, when the output volume is at 100 % (see table operating conditions).
Alternatively, there is a power jack or a screw terminal with an input voltage range from 4.5 V to 24 V.
It is also possible to make use of an USB power supply wall adapter with at least 800 mA. Without using the audio amplifier, also weaker power supplies starting from 200 mA can be used.

There are 3.5 mm audio jacks for microphone (MIC), headphone (LINE OUT), and audio input (LINE IN) available. The signals are realized on screw terminals. An integrated 1-W audio amplifier allows direct connection of a 4- or 8-ohm loudspeaker. The gain is 12 dB. The loudspeaker can be connected at the screw terminal (SPEAKER).

Standard condenser electret microphone can be used as microphone input (e.g. Plantronics Audio 40, desktop microphones or standard analoge headsets).

A DECT-Headset can be connected to provide a wireless, free, distant voice recognition. For this purpose, the base station needs to be connected to the DECT socket of the vicBASE-3_5.

To reset or recover the board, it is mounted with 2 buttons. Both signals are low active and also realized as screw terminal and pin header.

There are 3 UARTs, which can be connected via the screw terminals and pin headers. UART1 and UART2 are also accessible as virtual COM Port via Micro-USB.

In total, there are up to 10 digital inputs and 10 digital outputs provided. The IOs are electrically isolated from the remaining board when they are tapped at the screw terminals. All IO signals are available at a pin header as well (without electrical isolation). The vicBASE-3_5 is also in a second assembly variant available with a reduced scope of 6 IOs and without a third UART.

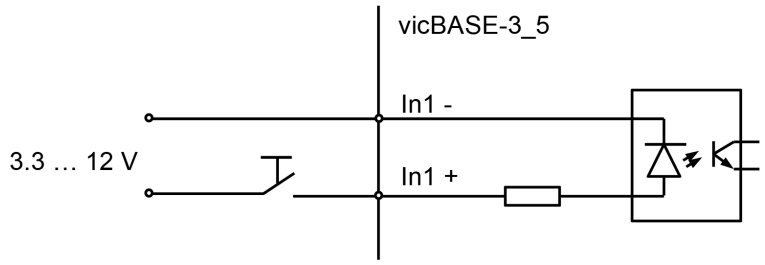



**Figure 14:** Example usage of digital inputs

### 2.2.4 Dimensions

The dimensions are 145 x 75 mm. The maximum of 18 mm height comes with the DECT socket.

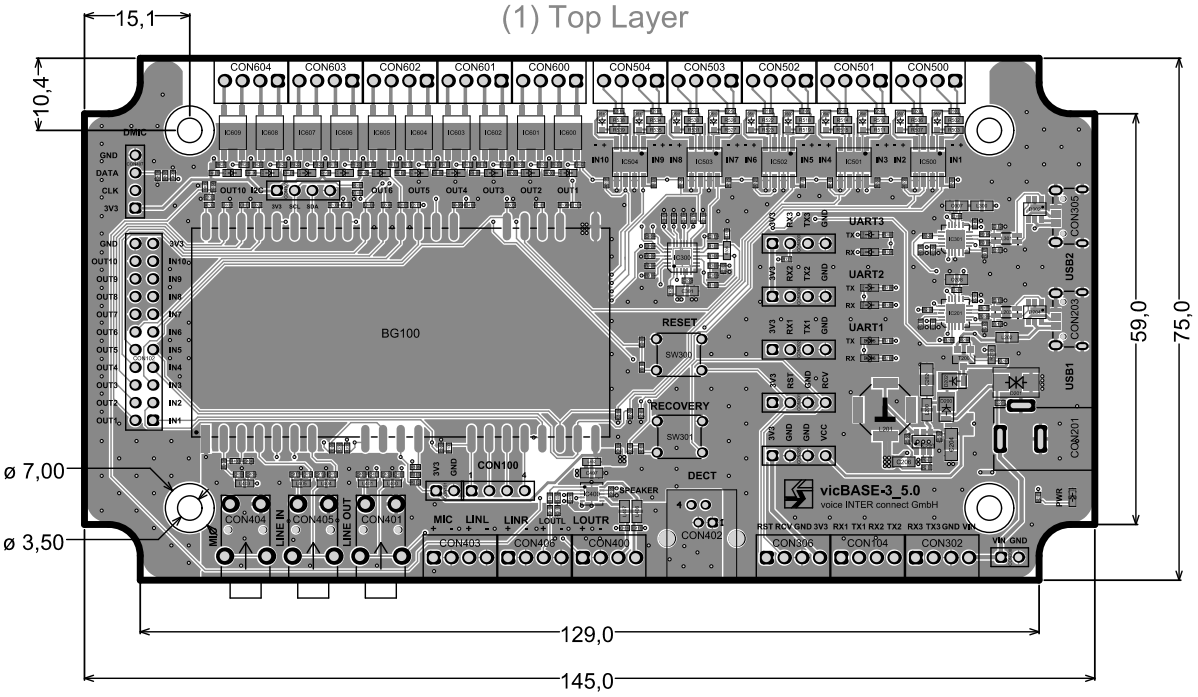



**Figure 15:** vicBASE-3_5 top view (dimensions in mm)

# 3 Firmware

To configure the speech recognition, a PC with Windows 7 or newer is necessary.

## 3.1 Delivery state

The example dialog that is shown below at this manual and also part of the delivery is preconfigured at the platform. The short tune “p1.wav” can be found at all audio prompt slots except at the slot “Prompt 2”. This slot contains “p2.wav”.

Currently, 30 languages are supported. One language can be installed at the platform at a time.

The installed language can be changed via the Update/Recovery tool (Section 3.3) by uploading one of the vicCONTROL_LanguagePack_... .vcupd files from the delivery.

## 3.2 Control via the serial interface

### 3.2.1 Settings of the serial connection

The vicCORE-3_2 provides two UARTs. When employing vicBASE-3_5, both UARTs can be used as virtual serial connections via USB ports. The first UART is used for communicating with the platform, wheras the second UART is used for debug messages.

The settings of the serial connection can be changed by the vicCONTROL Designer (see Chapter 5). The following values are the defaults:

- Baudrate: 115200
- Data bits: 8
- Parity: none
- Stop bits: 1
- Handshaking: none

### 3.2.2 Protocol

- Only point to point connections are available.
- The protocol is telegram-based.
- There are different kinds of telegrams.
- The interchange of several connected telegram is called process.
- If there is no active process, both partners can initiate a new one.
- Processes initiated by the VCMS platform are always unidirectional. That means there is no need for the respond to these telegrams.

- If there is a collision at the beginning of a process, the host needs to detect and handle it.
- The algorithm for checksum calculation is CRC-CCITT (CRC 16):
  - mode = forward
  - start = 0xFFFF
  - polynomial = 0x1021
- The characters are encoded in UTF-8.

### 3.2.3 Telegram structure

- Length between 2 and 268 Byte
- Always starts with “v1”
- The third byte specifies the type. There are three types:
  - Command “C”
  - Answer “A”
  - Message “M”
- The next 4 bytes are the name
- The next two bytes specify the length of the data payload as hex dump
- The payload follows
- After the payload 4 characters representing the two byte checksum (CRC16 on all previous bytes of the telegram) as hexdump
- A carriage return (CR) sign as terminator

| Number of bytes | 2 | 1 | 4 | 2 | 0 …255 | 2 | 1 |
|---|---|---|---|---|---|---|---|
| Meaning | Head | Type | Name | Parameter length | Parameter | Check sum | CR |
| Example (start speech recognition) | v1 | C | Recg | 00 | | C32C | “\r” |

### 3.2.4 List of commands, answers and messages

**Commands**

Commands are sent to the platform

| Name | Meaning | Parameter length | Parameter | Range |
|---|---|---|---|---|
| Recg | Recognition mode | 0 | | |
| Stop | No recognition mode | 0 | | |
| GtSt | Get current menu/state | 0 | | |
| SwSt | Switch to state (menu) | 1 …255 | Menu name (text) | UTF-8 |
| TrEv | Trigger event | 1 …10 | Command Class-ID (decimal) | 1 …500 |
| PlPr | Play prompt | 1 …10 | Index of the prompt | 1 …24 |
| GetV | Get firmware version | 0 | | |
| GLng | Get language | 0 | | |
| GtPN | Get platform | 0 | | |

Examples (ANSI presentation):
Start speech recognition
v1CRecg00C32C\r
Stop speech recognition
v1CStop00A146\r
Get current menu/state:
v1CGtSt000278\r
Switch to menu “Untermenü”
v1CSwSt0A**UntermenÃ¼**132D\r
Trigger Event “1”
v1CTrEv011**1**0BAE\r
Play prompt 1 1
v1CPlPr011**1**BDB3\r
Get firmware version
v1CGetV0022B6\r
Get language
v1CGLng0003E7\r
Get platform
v1CGtPN009BC0\r

**Answers**

The platform uses answers to react on commands

| Name | Meaning | Parameter length | Parameter | Range |
|---|---|---|---|---|
| Retn | Answer | 3 | Error code (decimal)<br>000 = no error | |
| Vers | firmware version | 0 …255 | Text | |
| Lang | language | 0 …255 | Text | |
| PfNm | Platform | 0 …255 | Text | |

Example:
Answer: “No Error” v1ARetn03**000**E9C4\r

**Messages**

Messages are generated asynchronously by the platform, for example if a phrase was recognized.

| Name | Meaning | Parameter length | Parameter | Range |
|---|---|---|---|---|
| Volm | Input volume | 4 | value as hexdump | 0x0000 …0x7FFF |
| RsLn | A result with a number of hypothesis was recognized | 2 | number of hypothesis (decimal) | 0 …20 |
| RsHy | Hypothesis | 9 …255 | confidence in "[]" followed by Triggerclass-ID(s), which are separated by "," and closed with ";" | |
| RsTx | Text that was recognized | 0 …255 | text | UTF-8 |
| StCh | State changed / answer on GtSt command | 0 …255 | name of new menu (State) | UTF-8 |
| Boot | Boot done | 0 | | |

Speech recognition results (RsLn, RsHy, RsTx) and state changes (StCh) are put out only if at vicCONTROL Designer no "Custom serial output" is configured for a transition (see Chapter 5).

For each speech recognition result RsLn is put out first. It tells the number of hypotheses of the result. Then for each hypothesis RsHy and RsTx are put out. RsHy indicates the confidence of the hypothesis and its Triggerclass-ID. RsTx tells the recognized text.

Example:
Boot done
v1MBoot00F88F\r
Current volume
v1MVolm04**0127**03C0\r
Result witch one hypothesis
v1MRsLn02**01**5630\r
hypothesis with confidence 04758 and Triggerclass-ID 2
v1MRsHy09[**04758**]**2**;1F04\r
Result text
v1MRsTx07**Heizung**5161\r
Change to now state named "Heizung"
v1MStCh07**Heizung**F45F\r

### Access via terminal emulator

To test the serial connection under Windows, you can use the software PuTTY, for example.

Configure it like shown below. Please replace "COM10" with the name that was assigned to the (virtual) COM Port to which the vicCONTROL platform is connected.

After a successful connection, type the command to get the version ("v1CGetV0022B6")
In return, you should receive the current firmware version.

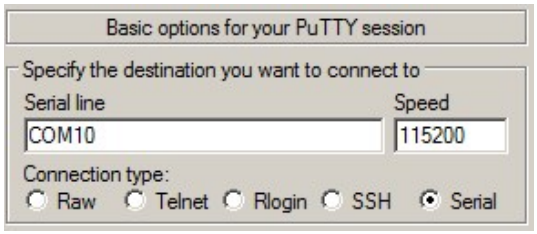



**Figure 16:** PuTTY configuration 1

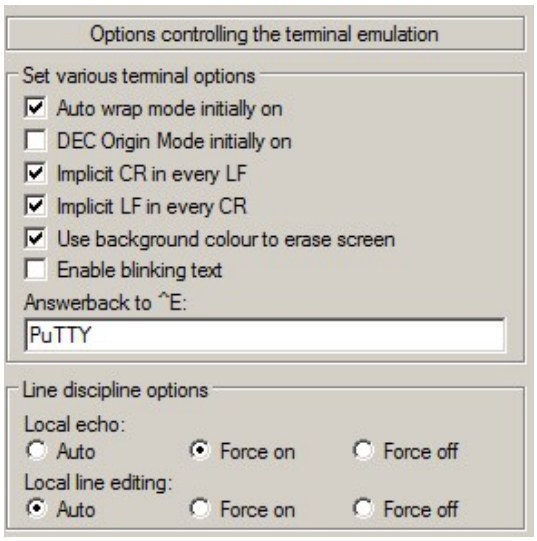



**Figure 17:** PuTTY configuration 2

## 3.3 Update and Recovery

For all firmware updates and as a recovery solution after unsuccessful configuration attempts by vicCONTROL Designer, the program vicCONTROL_update2.2.0.exe is provided. It can be found on the USB stick.

Depending on the used Windows version, administrator privileges are necessary. Figure 18 shows the user interface. An installation of the software is not necessary.

Firmware updates are provided as .vcupd files. They usually come in two flavors:

1. "Firmware" updates which are the full updates containing the latest software with new features and bug-fixes.
2. "LanguagePacks" are updates that run faster since less data has to be transmitted onto the board if only the language should be changed. Before using LanguagePacks it must be ensured, that the software running on the the board is from the same release version. This is achieved by installing a full "Firmware" update first before starting to use the LanguagePacks delivered with that same release version.

All updates no matter if "Firmware" or "LanguagePack" will reset the board to factory defaults.

**vicBASE-3_5**

- Connect the vicBASE-3_5 via USB1 to your PC.
- Start the update program.
- Choose the correct COM port at the top left (The list of available ports can be refreshed by pushing rescan)
- Use the "Open Update File" button to open an update file. The file extension of update files is "vcupd". Information and further instructions about the update will be presented.
- At the vicBASE-3_5: Push and hold the RECOVERY button. Push the RESET button while still holding the RECOVERY button. Keep holding the RECOVERY button for at least 3 seconds after the RESET button is released.
  The board is in update mode now.
- Push "Start" inside the update program to run the update process.

**vicCORE-3_2**

A functioning serial connection (COM) has to be set up between your PC and the vicCORE-3_2 (e.g. by MAX232).

- The Input RECOV (X138) has to be connected to ground (GND) before powering the board.
- Start the update program.
- Choose the correct COM port at the top left (The list of available ports can be refreshed by pushing rescan).
- Use the "Open Update File" button to open an update file. The file extension of update files is "vcupd". Information and further instructions about the update will be presented.
- Push "Start Update" to run the update process.

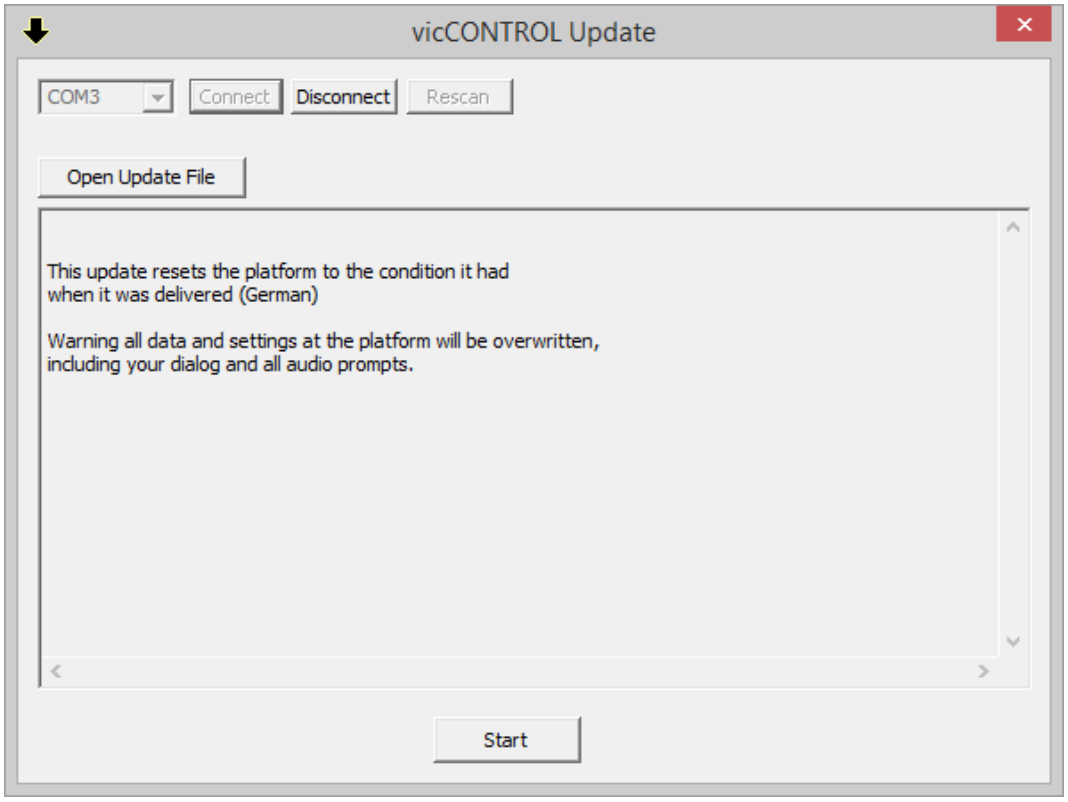



**Figure 18:** Screenshot of vicCONTROL_update2.2.0.exe

If the process is interrupted it can/must be repeated.

# 4 USB Driver

A driver, which offers a virtual COM port, is required for communicating with *vicCONTROL go* via the USB interface of the vicBASE-3_5. This is usually installed automatically the first time plugging in when using Windows OS. Aside from that, the driver is available in the delivery of *vicCONTROL go* and through the manufacturers website:
https://www.ftdichip.com/Drivers/VCP.htm

Depending on the windows version, in order to install the driver, administrator privileges are necessary. For users without administrator privileges, it is recommended to use always the same USB port.

# 5 vicCONTROL Designer

vicCONTROL Designer is the modeling tool that is used to configure *vicCONTROL go*. A configuration is called dialog. vicCONTROL Designer is tested to work at Windows 10 and 11 (x64).

The Designer does also need a **x86** "Visual C++ Redistributable for Visual Studio 2012"which is included at the delivery. It will be installed together with the designer.

## 5.1 Installation and Configuration

- Run the vicCONTROL Designer setup program.
- Start vicCONTROL Designer by clicking on the link at your desktop or from the start menu.
- Open the window menu and choose preferences. Here you can setup the target hardware and the language for which a dialog should be created.

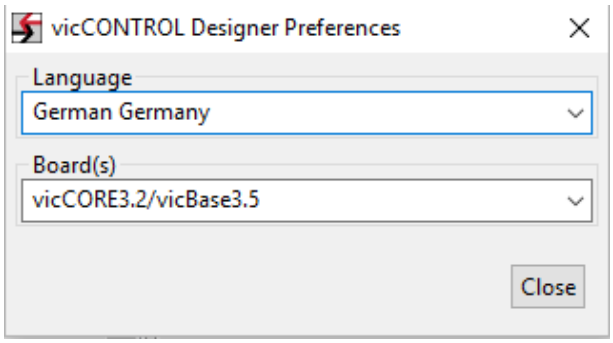



**Figure 19:** Preferences dialog

## 5.2 User interface

The user interface consists of a main window and several sub windows. The position and size of each sub window can be configured freely. Figure 20 shows a possible layout. At the example, the dialog file "Exam_German Germany.vdd2" is opened. On the upper right corner, the Properties window can be seen, which is used to edit the dialog element's properties. At the bottom right is a simplified graphical preview of the dialog. Behind that preview is the console window that shows status and error messages. At the bottom left position, the platform window is shown that is used for the interaction with the hardware platform. At top are the menu and command bar.

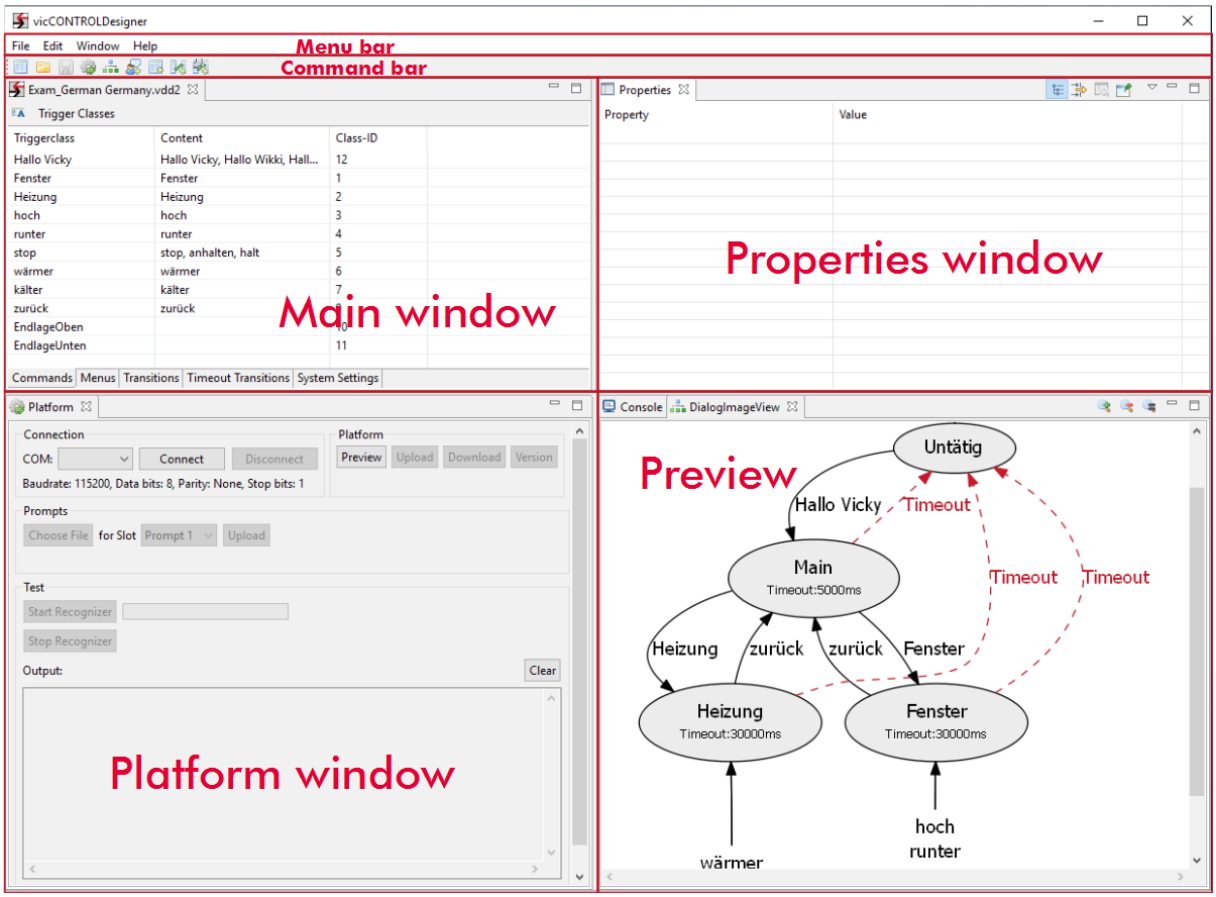



**Figure 20:** User interface

- The main window is divided into tabs (“Commands”, “Menus”, “Transitions”, “Timeout Transitions”, and “System Settings”).
- GUI elements can be repositioned within their table by dragging them with the left mouse key.
- To delete an element from one of the tables, right-click on it and select “Delete” at the context menu.
- The complete speech control application is called dialog.
- A dialog consists of the following top level elements:
  - “Trigger” classes (created and edited via the “Commands” tab)
  - “Menus” (created and edited via the “Menus” tab)
  - “Transitions” (created and edited via the “Transitions” tab)
  - “Timeout transitions” are special transitions that are created and edited via the “Timeout Transitions” tab.
  - At the tab “System Settings”, basic properties of the dialog are edited.
- GUI elements can be positioned by Drag and Drop.
- For each element selected at the main window, more properties are shown at the “Properties” sub window if available. That is where they get edited as well.

- If a sub window was closed, you can open it again via “Window” at the main menu.

### 5.2.1 Creating a dialog

The five tabs of the main window are described in the following:

**Commands**
At the command editor, new so called trigger classes are created or existing ones are edited. Trigger classes have the following properties that can be edited via the main window:

- Name
- Content (the speech commands that triggers the class)
- ID (automatically assigned)

**Menus**

A menu comprises a number of actions that are realized as transitions. Actions can be speech inputs, button pushes or value changes at digital inputs. The following properties can be edited per menu at the main window:

- Name
- Timeout:
  After a specified time period (in milliseconds), a time out will be triggered. At the "Timeout Transitions" tab, the behaviour can be defined, e.g. switching to another menu (see below for details).
- Score High:
  Score (confidence) set must be met to take a result hypothesis of the speech recognition as correct.
- Score Low:
  If this score is not met, the hypothesis is judged as wrong. It is neither reported nor anything else is done. If the score of a recognition hypothesis is between High and Low, it is judged as unsure. In the "System Settings" editor, it is specified what happens when the speech recognition output is unsure. If the values of "Score High" and "Score Low" are equal, there is no unsure region for this menu and the unsure action (like a prompt asking "Please repeat!") will be inhibited for this menu.
- Use wakeup word:
  This defines whether that menu should be guarded by the global wakeup word that is configured at the "System Settings" tab.

The following additional options are available at the Properties window for each menu:

- Garbage words:
  This field is used to input words that should definitely not be recognized at this menu.
- Is Special Menu:
  This option enables the user to edit additional properties of the speech recognizer. Menus with that option set get shown in yellow at the preview window.

- Wakeup timeout
  Time after which the user will have to repeat the wakeup phrase for this menu.

**Transitions**

At the “Transitions” tab, transitions between menus/states are configured. A transition includes all properties to define the behaviour when a transition is triggered. The following properties can be edited at the main window:

- Old Menu:
  Menu/state the transition starts at.
- New Menu:
  Menu/state the transition goes to.
- Triggerclass:
  This is the trigger; it is defined at the command editor.
- Comment:
  Comments are shown at the preview in order to distinguish transitions more easily.

Additional properties are shown at the Properties window. They depend on the board selected at the preferences dialog (see Chapter 2 and Section 5.1)

- Custom serial output:
  If a custom serial output is configured inside this field, it will be send via the serial command interface (connectors X110 X111 of the vicCORE-3 or USB1 at the kit). If the field is left empty, a standard result will be send (see Section 3.2.4). Common C-like escape sequences canbe used e.g. “\n” for new line. For non-printable and special bytes, octal sequences can be used, e.g.“\177” for a “DEL” (0x7F).
- Inx Guard:
  A guard is a condition that must be met in order to perform a transition (e.g. In1 Guard = low, which means that digital input 1 must be low).
- Prompt:
  Set the audio prompt to play.
- Outx Action:
  Defines the behaviour of the relays or digital outputs. Relays can be switched on or off or keep their state (DontCare).
- Outx TO:
  The time out for an output action in milliseconds. If a value is set to greater than 0, the relay will switch off after that time.

### Timeout Transitions

Here, if a timeout was set at the menu editor, the behaviour of the system is modelled if this timeout is triggered. Timeout transitions got the following properties:

- Old Menu:
  Menu/state the transition starts at.
- New Menu:
  Menu/state the transition goes to.
- Comment:
  Comments are shown at the preview in order to distinguish Timeout Transitions more easily.

Timeout Transitions have the same additional properties as “normal” Transitions.

### System Settings

The following system settings are available:

- Audio settings
  - Digital input amplification
    The input audio signal can be amplified digitally if necessary.
  - Enable Talk Through
    If this switch is set to true, the input audio signal, which is processed by the recognizer, is put out to the right audio output channel.
  - Out Level
    This is used to control the volume of audio output (e.g. if the .wav files for output prompts are normalized to full volume).
- Digital inputs to event mapping
  This allows to map events at the digital inputs to trigger classes. If, for example, a trigger class should be triggered when input 3 changes from high to low, then this class has to be chosen behind In 3 H $\rightarrow$ L Event.
- Serial output configuration
  The serial connection settings should only be changed **if really necessary**.
  The standard settings are:

  | | |
  |---|---|
  | Baudrate | 115200 |
  | Data Bits | Bits 8 |
  | Parity | None |
  | Stop | Bits 1 |

  After these settings have been adjusted, the dialog needs to be transferred to the platform. During that process, the connection will break at the moment the new settings get active. Therefore, it is necessary to go to the platform window and set the new values there **as well** (double click the text with the serial settings at the platform window) and **reconnect**. If the serial settings of a platform are unknown, the recovery tool (Section 3.3) can always be used to reset it.

  It is also possible to adjust which messages are sent by the platform:

  | | |
  |---|---|
  | Signal Menu Changes | Message on menu change |
  | Signal Trigger Ids | The ID of a Trigger class will be sent |
  | Signal Trigger Text | The recognized text will be sent |
  | Signal Volume | Message about current volume enable/disable |

- Wakeup
  Configures the speech recognition parameters of the globally available wake up phrases (including the actual words constituting them)

- Action on wakeup
  Defines globally what should happen if the wake up phrase guarding a menu was recognized (audio prompt, output action, custom message via the serial interface). This is configured like a transition at the Properties window.

- Action on sleep
  Defines globally what should happen if the wakeup timeout has passed (audio prompt, output action, custom message via the serial interface). This is configured like a transition at the Properties window.

- Action if speech recognition between high and low score
  Defines globally what should happen if the confidence score of a speech recognition result is between low and high score (audio prompt, output action, custom message via the serial interface). This is configured like a transition at the Properties window.

### 5.2.2 The Platform Window

The platform window is used to do all board specific tasks. After a connection is established, dialogs can be prepared and uploaded.

After the dialog is transferred, the system can be tested by enabling the recognizer:

- Start Recognizer: start speech recognition
- Stop Recognizer: stop speech recognition

In the bottom area of the window, it is shown what the platform returned via the serial interface. If custom serial returns are defined (Custom serial output), they are only shown correctly if they only use printable ASCII characters and end with a CR character ("\r").

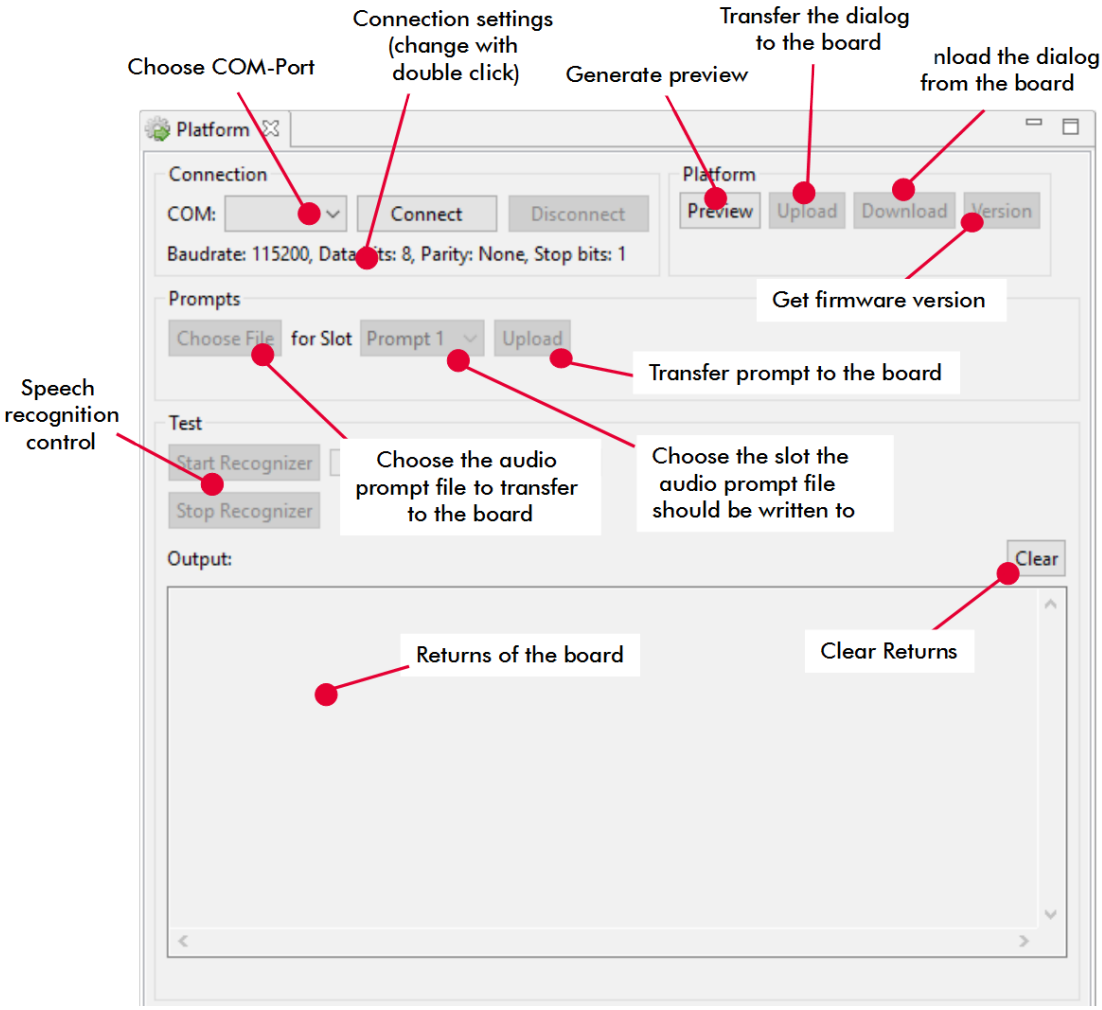



**Figure 21:** "Platform" window

If audio prompts should be used, they can be uploaded as well. Each of the up to 8 audio prompts can be 4 seconds long. It has to be RIFF-WAV formatted (16 bits, 16 kHz, Mono) with a 44 byte long standard header.

The audio prompt upload process:

It is advisable to stop the recognizer, push the RESET button on the board and wait for 3 seconds before uploading a prompt

1. Select “Choose file” and choose the audio file you want to load onto the board
2. Select the slot from the drop-down list right of the “Choose file” button
3. Push Upload right of the drop down list to start the transfer

If a prompt transfer fails with an error message at the very end, that does not mean that data was not transferred, but the board should be reset again before doing another operation (push RESET button on the board and wait for 3 seconds)

## 5.3 Creating an example dialog

**ATTENTION:**
The stop position switches in this example serve to illustrate the usage of the digital inputs. *vicCONTROL go* alone must never be used for security-related functions!

This example shows a simple home control application for shutter and heating. The project file can be found on the USB stick.

The shutter should be driven up or down. Shutter movement should be stoppable at every time. There are stop position switches for the shutter. They trigger as soon as the shutter is at its topmost or bottommost position. The example expects that an engine drives the shutter up as long as relay 1 is closed and the upper end position switch is not closed. As long as relay 2 is closed and the bottom end position switch is not closed, the shutter is driven down.

It is expected that the temperature of the heating is increased by a 500-ms impulse at relay 3 and decreased by a 500-ms impulse at relay 4.

Figure 22 shows a schematic overview.

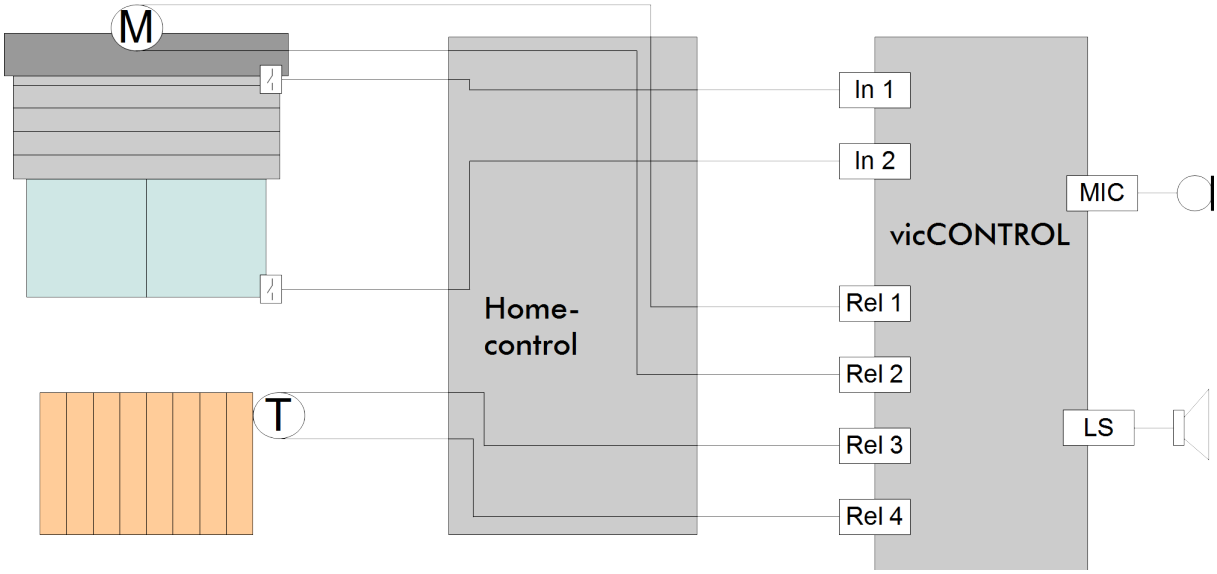
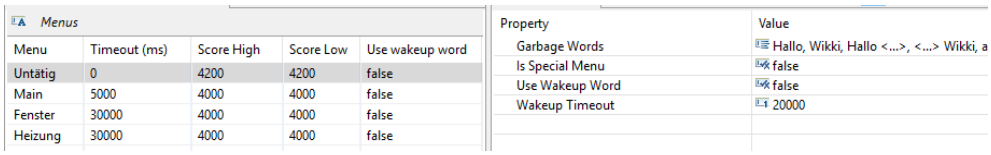



**Figure 22:** Example application

Procedure:

- At first create a new dialog by clicking: → New or
- Define a menu
  - Go to the Menus tab at the main window
  - Four menus are necessary. Push the button "Create Menu" four times!
  - Change the names to Idle, Main, Fenster and Heizung

- The Idle will contain the activation phrase “Hallo Vicky” (Hello Vicky). Since we do not want any feedback about unsure recognition at this menu, both scores are set to the same value. “Hallo” is added as Garbage Word in order to achieve that the recognizer reacts on “Hallo”.
- The Main menu gets a time out of 5 seconds.
- The menus Fenster and Heizung get timeouts of 30 seconds each.

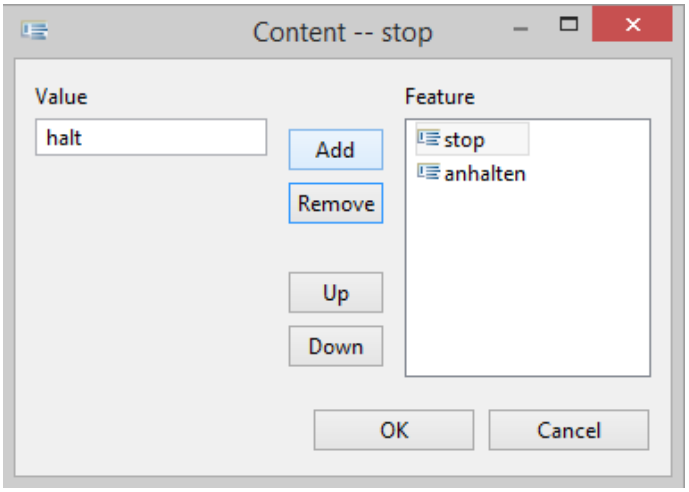


Menus

| Menu | Timeout (ms) | Score High | Score Low | Use wakeup word |
|---|---|---|---|---|
| Untätig | 0 | 4200 | 4200 | false |
| Main | 5000 | 4000 | 4000 | false |
| Fenster | 30000 | 4000 | 4000 | false |
| Heizung | 30000 | 4000 | 4000 | false |

| Property | Value |
|---|---|
| Garbage Words | Hallo, Wikki, Hallo <...>, <...> Wikki, a |
| Is Special Menu | false |
| Use Wakeup Word | false |
| Wakeup Timeout | 20000 |

- Create Triggerclasses for the transitions
  - Go to the Commands tab at the Main window
  - Push the button “Create Triggerclass”  . The Class-ID is set automatically.
  - The following classes are created for speech commands:
    - Hallo Vicky
    - Fenster
    - Heizung
    - hoch
    - runter
    - stop
    - wärmer
    - kälter
    - zurück
  - After clicking on the Content fields of a class, a  Button will be shown. It opens a dialog that is used to define the speech commands of a class.




- The field below value is used to input new speech commands. They can be added or removed to the class by clicking the corresponding button.
- After adding the speech command classes, two classes are added for the stop positions. Their (speech) content is left blank.
  * EndlageOben
  * EndlageUnten

Trigger Classes

| Triggerclass | Content | Class-ID |
|---|---|---|
| Hallo Vicky | Hallo Vicky, Hallo Wikki, Hallo Wiggi, Hallowikk... | 12 |
| Fenster | Fenster | 1 |
| Heizung | Heizung | 2 |
| hoch | hoch | 3 |
| runter | runter | 4 |
| stop | stop, anhalten | 5 |
| wärmer | wärmer | 6 |
| kälter | kälter | 7 |
| zurück | zurück | 9 |
| EndlageOben | | 10 |
| EndlageUnten | | 11 |

Commands | Menus | Transitions | Timeout Transitions | System Settings

- Define transitions (changes between States/Menus)
  - Switch to the Transitions tab.
  - Use the  Button to create a new transition.
  - Create 4 transitions for the changes between the menus first.

Transitions

| Old Menu | Triggerclass | New Menu | Comment |
|---|---|---|---|
| Idle | Hallo Vicky | Main | |
| Main | Fenster | Fenster | |
| Fenster | zurück | Main | |
| Main | Heizung | Heizung | |
| Heizung | zurück | Main | |

Commands | Menus | Transitions | Timeout Transitions | System Settings

– At the heating menu, a transition to increase the temperature is required. If this transition is triggered, the system will stay at the same menu. In the properties, relay 3 is set to active for 500 ms.

– A similar transition is added for “cooler”.

Transitions

| Old Menu | Triggerclass | New Menu | Comment |
|---|---|---|---|
| Idle | Hallo Vicky | Main | |
| Main | Fenster | Fenster | |
| Fenster | zurück | Main | |
| Main | Heizung | Heizung | |
| Heizung | zurück | Main | |
| Heizung | wärmer | Heizung | |
| Heizung | kälter | Heizung | |
| Fenster | hoch | Fenster | |
| Fenster | runter | Fenster | |
| Fenster | stop | Fenster | |
| Fenster | EndlageOben | Fenster | |
| Fenster | EndlageUnten | Fenster | |

| Property | Value |
|---|---|
| Custom serial output | |
| In1 Guard | none |
| In2 Guard | none |
| In3 Guard | none |
| In4 Guard | none |
| PB1 Guard | none |
| PB2 Guard | none |
| Prompt | Prompt01 |
| Rel1 Action | DontCare |
| Rel1 Timeout | 0 |
| Rel2 Action | DontCare |
| Rel2 Timeout | 0 |
| Rel3 Action | SwitchOn |
| Rel3 Timeout | 500 |
| Rel4 Action | DontCare |

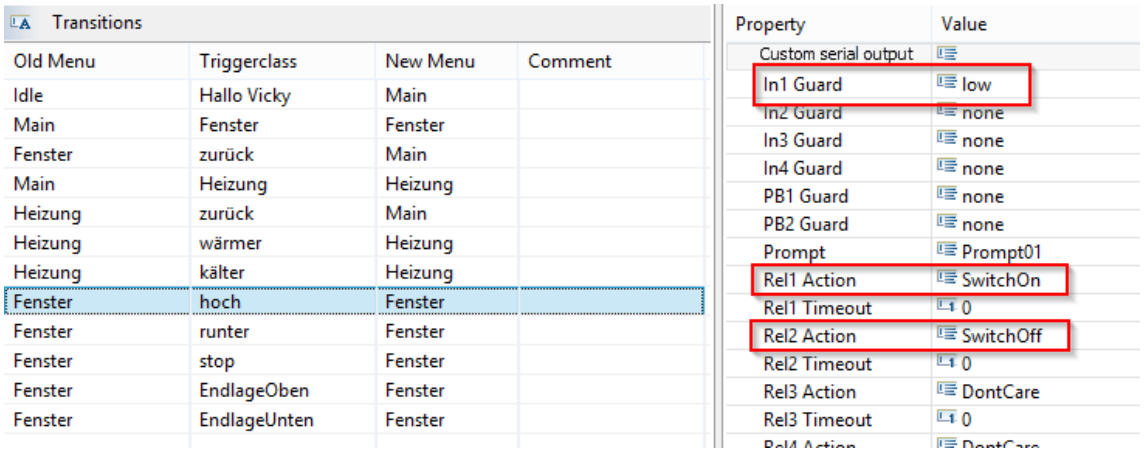
– At the Fenster menu, a transition is necessary to drive the shutter up:
  * Relay 1 is activated (Rel1Action = SwitchOn)
  * Relay 2 is deactivated (Rel2 Action = SwitchOff)
  * It is only done if the upper stop position is not enabled (In1 Guard = low)

– Another transition is necessary to drive the shutter down:
  * Rel1Action = SwitchOff
  * Rel2Action = SwitchOn
  * In2 Guard = low

– Another transition is necessary to stop the shutter:
  * Rel1Action = SwitchOff
  * Rel2Action = SwitchOff

– Another transition is necessary to stop the shutter when upper stop position is reached:
  * Rel1Action = SwitchOff

  - Another transition is necessary to stop the shutter when bottom stop position is reached:
    * Rel2Action = SwitchOff
  - Finally, the shutter should stop if the Fenster menu is exited. The properties of the existing transition from "Fenster" to "Main" have to be adjusted:
    * Rel1Action = SwitchOff
    * Rel2Action = SwitchOff
  - For all transitions, Prompt 1 is chosen as feedback sound.

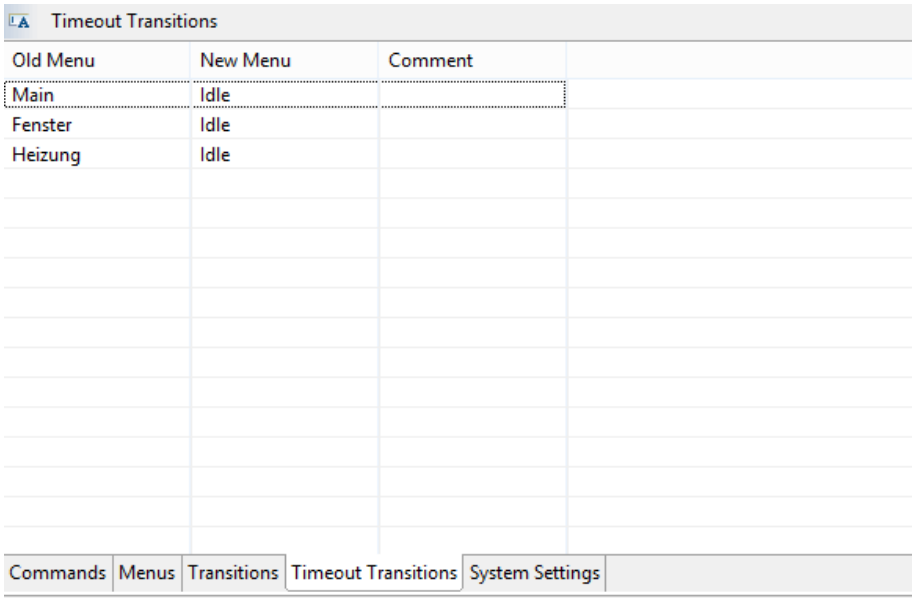


Transitions

| Old Menu | Triggerclass | New Menu | Comment |
|---|---|---|---|
| Idle | Hallo Vicky | Main | |
| Main | Fenster | Fenster | |
| Fenster | zurück | Main | |
| Main | Heizung | Heizung | |
| Heizung | zurück | Main | |
| Heizung | wärmer | Heizung | |
| Heizung | kälter | Heizung | |
| Fenster | hoch | Fenster | |
| Fenster | runter | Fenster | |
| Fenster | stop | Fenster | |
| Fenster | EndlageOben | Fenster | |
| Fenster | EndlageUnten | Fenster | |

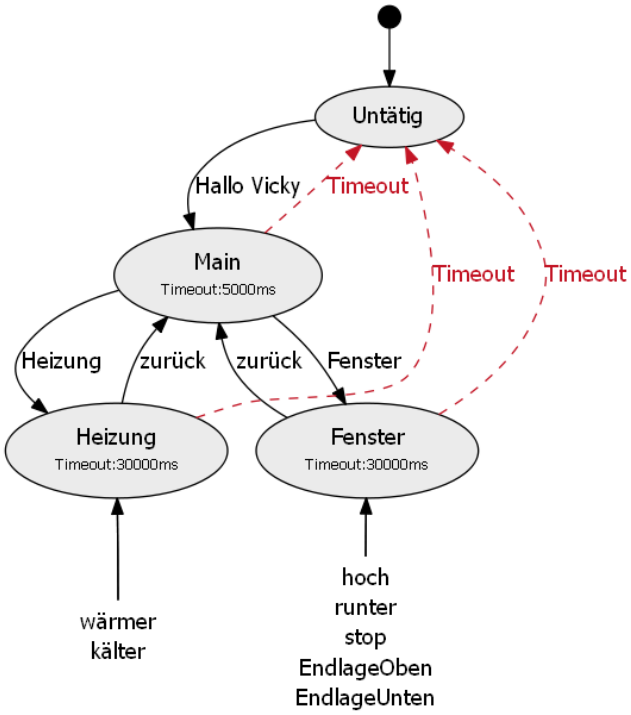
| Property | Value |
|---|---|
| Custom serial output | |
| In1 Guard | low |
| In2 Guard | none |
| In3 Guard | none |
| In4 Guard | none |
| PB1 Guard | none |
| PB2 Guard | none |
| Prompt | Prompt01 |
| Rel1 Action | SwitchOn |
| Rel1 Timeout | 0 |
| Rel2 Action | SwitchOff |
| Rel2 Timeout | 0 |
| Rel3 Action | DontCare |
| Rel3 Timeout | 0 |

- Timeout transitions: Since the menus Main, Fenster, and Heizung are defined in terms of timeout, it has to be defined what happens after this time has elapsed.
  - Go to the Timeout Transitions tab.
  - Create 3 Timeout Transitions by pushing  three times.
  - The new menu is always "Idle".
  - Prompt 2 should be played each time a timeout occurs.




- System Settings
  - Switch to the System Settings tab and choose the “Digital inputs to event mapping” entry.
  - The stop position switches are connected to the digital inputs 1 and 2. When the signal changes from low to high, a stop position transition should happen. At this point, the physical events are connected with a trigger class. (E1 L → H Event means input 1 Low to High)

- The dialog is fully modeled and can now be tested.
- Use the Preview or  buttons to preview the dialog.




  - Go to the platform window.
  - The board is expected to be ready for programming, that is:
    * power supply is on
    * USB or serial connection to the PC is established
  - Choose the correct COM port and push the connect button.
  - Push the Upload button to write the dialog to the platform.
  - After successful upload, speech recognition can be started by pushing Start Recognizer

The board is programmed and working.

# 6 Dialog design

## 6.1 General considerations

*vicCONTROL go* features a very flexible and powerful speech recognition engine that allows to create robust and rich speech control applications. However, depending on the task, some time for dialog optimization should be scheduled.

100 percent correct speech recognition cannot be guaranteed with this technology, but very low error rates can be achieved through appropriate dialog design and configuration.

The dialog size is currently restricted to at most 500 commands, 100 menus, and 500 transitions to limit memory consumption on the embedded platform.

## 6.2 Activation

Continuous speech recognition places special demands on the technology. This means that the speech recognizer "listens" all the time and is not activated by, for example, a button (push to talk) or another sensor (e.g. motion detector). In this case, it makes sense to use an activation phrase (or wakeup phrase/wakeup word), such as "Hallo Vicky" and recognize the actual commands in the second step. The following general rules apply to the activation phrase:

- **not too short**
- the less pronunciation variants of a word exist, the better
- the less sibilants ("sh", "th", ...), the better

The start or idle menu should also contain garbage words like the separate words "Hallo" and "Vicky".

If more than one phrase is active at the same time, they should be as different as possible.
"Ruf aufbauen" and "Ruf beenden" instead of
"Ruf aufbauen" and "Ruf abbauen"

## 6.3 Feedback

It is important to give feedback to the user. You can use audio prompts or optical signals. At multi-step dialogs, the user should be informed on any menu change. It might be useful to inform the user about every recognition by playing a short tune as in the example. An acoustic feedback like the demand "Please repeat!" on unsure recognition results is only useful in menus with "real" commands but not for activation phrases.

The recognizer not active during playback of sounds, so it does not "listen" in this case. That is even true if it is just playing back "silence" at the end of a sound file.

If speed is prioritized and if there is a clear system reaction on a command anyways, then we do recommend to abstain from playing an additional sound file to signal acknowledgment, but let the recognizer be ready for the next command faster instead.

## 6.4 One Shot Activation

Here one shot activation refers to the ability of the users to utter the activation phrase (or wakeup phrase/wakeup word) and the actual command at once without the need of a pause in between. This improves the user expierience and does speed up the usage of the system but there are also pitfalls to be aware of and this mode of operation might not be optimal for any application.

As an example a machine (German: Maschine) should be switched on or off (German: "einschalten" / "ausschalten"). That means the two commands should be combined with the activation phrase.

First the Commands get created at the "Commands" tab of vicCONTROL designer as usually.

The assignment of the activation phrase is done at the `System Settings` under `Wakeup`. There the Word "Maschine" gets defined. In general one shot activation phrases need some level of optimization. The system tries to estimate the quality of a given phrase (in our example, when trying to upload the project to the board, there will be a warning, that the word "Maschine" is not optimal since a slightly longer phrase would be more save). But, each activation phrase has to be evaluated and proofed for its applications environment.

Since some users always prefer to wait for an reaction on the wakeup phrase instead of speaking it together with the command without a pause, this possibility is always available.

At the `System Settings` under `Wakeup` and `Wakeup Speech Timeout` you can define how many seconds the system should wait for commands after the activation phrase was recognized (e.g. 10 seconds).

At the `System Settings` under `Action on wakeup` and `Action on sleep` you can configure the system's reaction on an activation phrase and the behavior when the active phase ends.

The reactions on normal commands are defined as usually with Menus and (Timeout) Transitions. The one shot activation can be activated per menu via the `Use Wakeup Word` at the properties of each of the menus.

On the USB-Stick or at the download area for customers you can find an example project file that showcases more sophisticated concepts around one shot activation.

At this file output 1 is used to indicate the current state of activation. There are also prompts configured to be played back on activation / deactivation.

The "Reset" menu at this example ensures that the activations phrase needs to be uttered again after each command. Otherwise it would be possible to utter additional commands until the full 10 seconds of timeout expire (see `System Settings -> Wakeup -> Wakeup Speech Timeout`).

One possible edge case (that might happen for long `Wakeup Speech Timeout`) is that users get distracted and forget that they already said the activation phrase. Than they repeat the phrase and expect the same reaction as on the first attempt and that the timeout gets reset to allow the full 10 seconds of commands again.

This "issue" is solved by the Menu "Main2" and the command "Maschine (spät)" (English: machine (late)) in the example file.

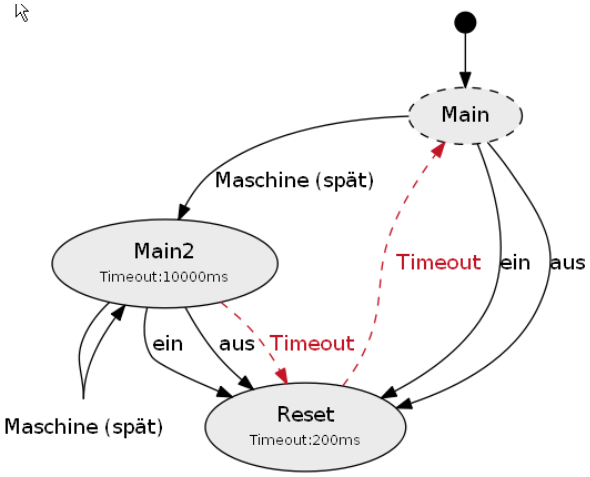



**Figure 23:** Oneshot Beispielprojekt

If you take a look at the synonyms of the commands in the example file, you will notice that there is always a variant with the activation phrase in front. This is implemented that way because experience tells that there are always users who prefer waiting for the reaction on the activation phrase, before going on with the actual command. When these users get distracted they might hesitate (e.g. 5 seconds) and after that in a relevant amount of cases they will say the activation phrase again but including the actual command this time.

The example deliberately showcases a pretty complex trade-off that tries to implement many possible usage scenarios at the same time. It is possible to implement more simple behaviors or even more edge cases. This is always up to the environment and your specific application.

## 6.5 Handling arbitrary unwanted speech (garbage), limited word spotting

This section describes advanced strategies and capabilities to optimize the recognition performance in special cases and environments.

*vicCONTROL go*'s general design aims to provide an easy to use way, to build fast and reliable speech control systems, for well defined sets of commands, which are spoken by cooperative users.

The tools described here can improve the robustness of your speech application and even allow to push usage beyond the original design goals of the system. But, there are no guaranties or automatic checks for wrong usage of these capabilities since it highly depends on the project, application and situation, whether the effects that can be achieved are desired or not.

### Explicit "Garbage Word" modelling

Sometimes similar sounding words are regularly spoken by the users of a speech interface, but only one of these words should be recognized as a command.

As example in German language there is the possible command consisting of two words meaning "Turn on the light!": "Licht einschalten!".

On the other hand there is a pair of words that does mean “Do not enable!” which sounds very similar: “Nicht einschalten!”

That means if there is a speech application which uses the command to turn on the light running in an environment where people often advise others to not turn on something it would be desirable that the speech recognizer knows the probable candidate of confusion and is able to handle it.

One way to model this is to simply add the unwanted phrase as a command and ignore it when it triggers. To ease the definition of such unwanted phrases, each menu defined in vicCONTROL Designer has the property “Garbage Words”. Each word or phrase that you put into that list will be ignored while the menu is active.

It is the speech application designers responsibility to make sure that there are no overlaps or conflicts with other command active in a menu and defined via the outgoing transitions.

### Ignoring “any speech” not defined otherwise

A special expression for a general garbage model is available which has the meaning: “Any other speech that should be ignored”.

In its context within a spoken phrase the general garbage model does include silence (no speech).

The general garbage model is defined with the character string “<...>”.

The general garbage model can be used when defining the “Contents” of commands or the “Garbage Words” of a menus.

Using the general garbage model “<...>” pushes to the limits of *vicCONTROL go*’s recognition engine and hardware resources. Therefore this expression’s use should be limited and accompanied by careful testing.

The freedom of using an expression with such a broad meaning includes the duty to take care that there are no overlaps or conflicts with other parts of your project.

To maintain a good performance it is advisable to restrict the use of the general garbage model to the beginning or end of a phrase.

- “<...> word”: higher probability to achieve goal
- “word <...>”: higher probability to achieve goal
- “word <...> wort”: higher probability of declining performance

The strength of this feature can be adjusted per menu via the advanced setting “Recg Garbage”.

## 6.6 Advanced speech recognition settings

*vicCONTROL go* is optimized to work in a wide range of possible applications and without the need to tune low level advanced parameters of the speech recognition.

While the freedom to tune the values of advanced parameters offers the ability to find solutions for special use cases, unsuitable values will harm the system's performance.

All advanced speech recognition settings can be set per menu. The full set is available if "Is Special Menu" is set to "true" and if "Recg Use Automatic" is set to "false".

- **Extra Event Enable** enables the detection of special acoustical events and should be kept enabled usually
- **Fx Sensitivity** and **Fx VAD Threshold** control the voice activity detection
- **Fx Speech Timeout** defines how long (in milliseconds), after the latest detected begin of speech, the recognition will be stopped if there is no end of speech detected. Since this value does directly affect the usage of hardware resources it can lead to performance problems if set to high.
- **Fx Trailing Silence** and **Recg Trailing Silence** both influence the time of silence (speech pause) in milliseconds that is necessary for the engine to recognize the end of a phrase. They are two parameters since this is evaluated at different stages of the recognition pipeline. As a rule of thumb the "Recg Trailing Silence" should be twice as high as the "Fx Trailing Silence"
- **Phrase Length Min** and **Phrase Length Max** define the minimal and maximal allowed lengths of a phrase in milliseconds. Since the maximum length directly affects the usage of hardware resources, it can lead to performance problems if set to high.
- **Accuracy** should be kept at the default maximum value usually
- **Garbage** controls the strength of the general garbage model
- **NBest Second** should be kept at the default value for a good trade-off between hardware resources and accuracy

# 7 Troubleshooting

**Fehlermeldung "vicCONTROL Designer does not start"**

If the error message does contain the text "(x86)", then please ensure first, that a **64 Bit** JRE (Java Runtime Environment) oder JDK (Java Development Kit) is installed and reachable via the System PATH.

An easy way to achieve this is to first un-install any existing JREs or JDKs and to install a single 64 Bit JRE or JDK afterwards.

**Fehlermeldung "Can not build for recognizer"**
Please make sure that the "Visual C++ Redistributable for Visual Studio 2012" for **x86** is installed on your machine.

**Transmission errors**
Upload of the dialog using vicCONTROL Designer failed and keeps failing when trying again. Please use the recovery program described at Section 3.3 to recover the board firmware.

**Unable to (re-)open a file**

- close the sub window of the file at the main window
- reopen it by choosing File → Open from the menu bar

**Issues connecting to the serial port**

Issues with the serial port up to program crashes of the designer can be caused by a to long Path environment variable on Windows 10. A too large list might be inherited from a Windows 7 that was upgraded to Windows 10 or by other software installations. Eventually it might lead to issues running the java environment.

Attention fixing this issue of the Windows setup, if it is really the case, needs some higher experience. It might be easier to just try with another (fresh) machine.

To check if you are facing that issue follow these steps:

1. Open Settings
2. Search for "path" and select "Edit the system environment variables" from the search results
3. Select "Environment Variables..."
4. Find the Path variable in the lower "System Variables" list. It is case insensitive and might be path or PATH in your case.

5. Try to add a “New” entry. If that fails with an error message, you need to shorten this list by removing entries that are not necessary any distribute the entries on more than many variable that can be nested into the final path variable.
   More reading: https://stackoverflow.com/questions/34491244

# 8 Migrating to vicCONTROL 4

With version 4.0, technical core components of the recognizer have been updated.

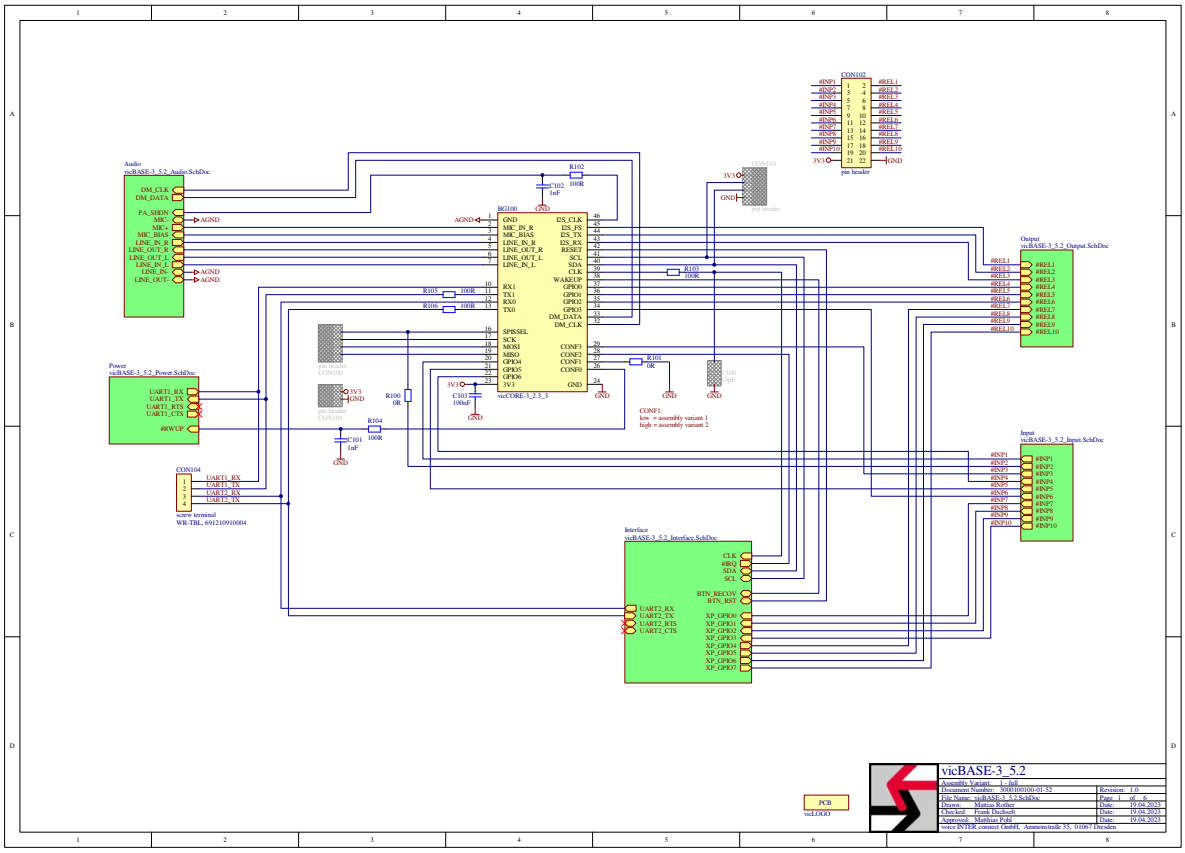
The general interface and behaviour were kept as stable as possible.

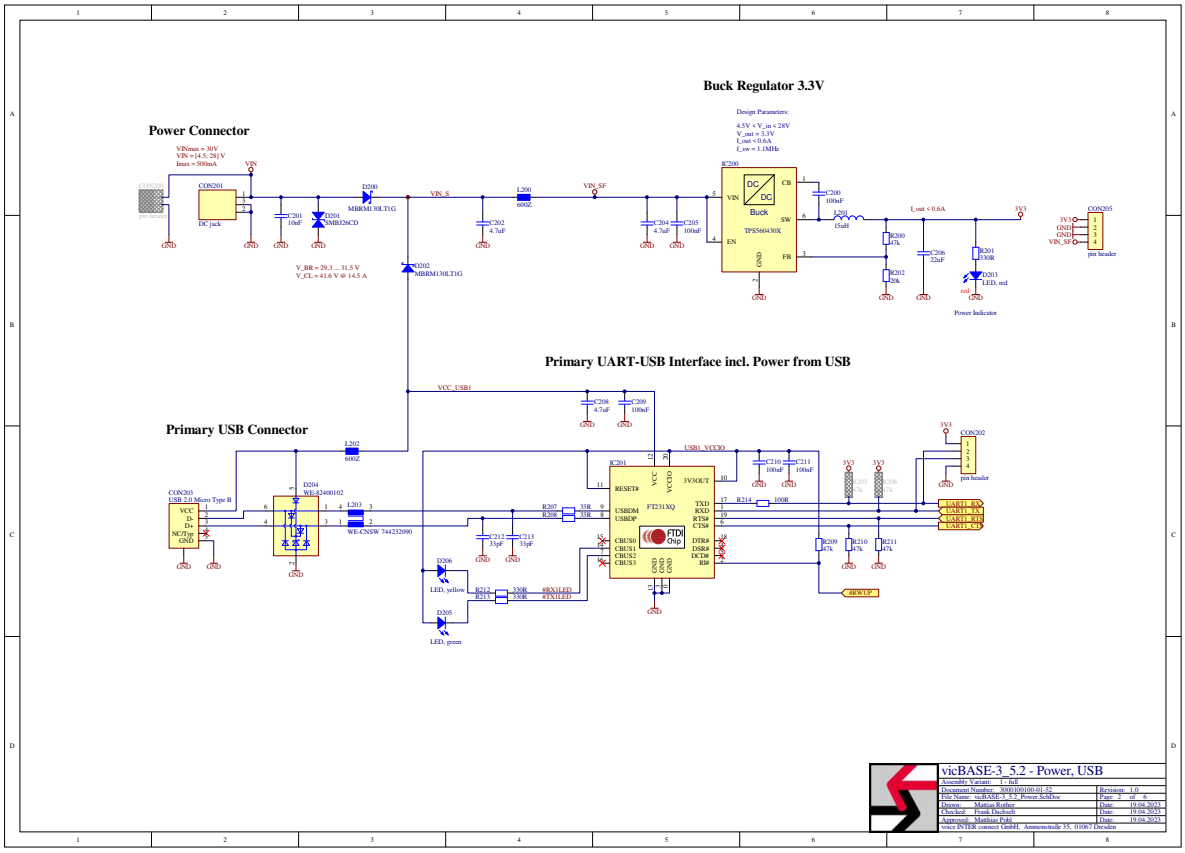
This chapter reveals the most important changes and explains how they can be dealt with in existing vicCONTROL projects.

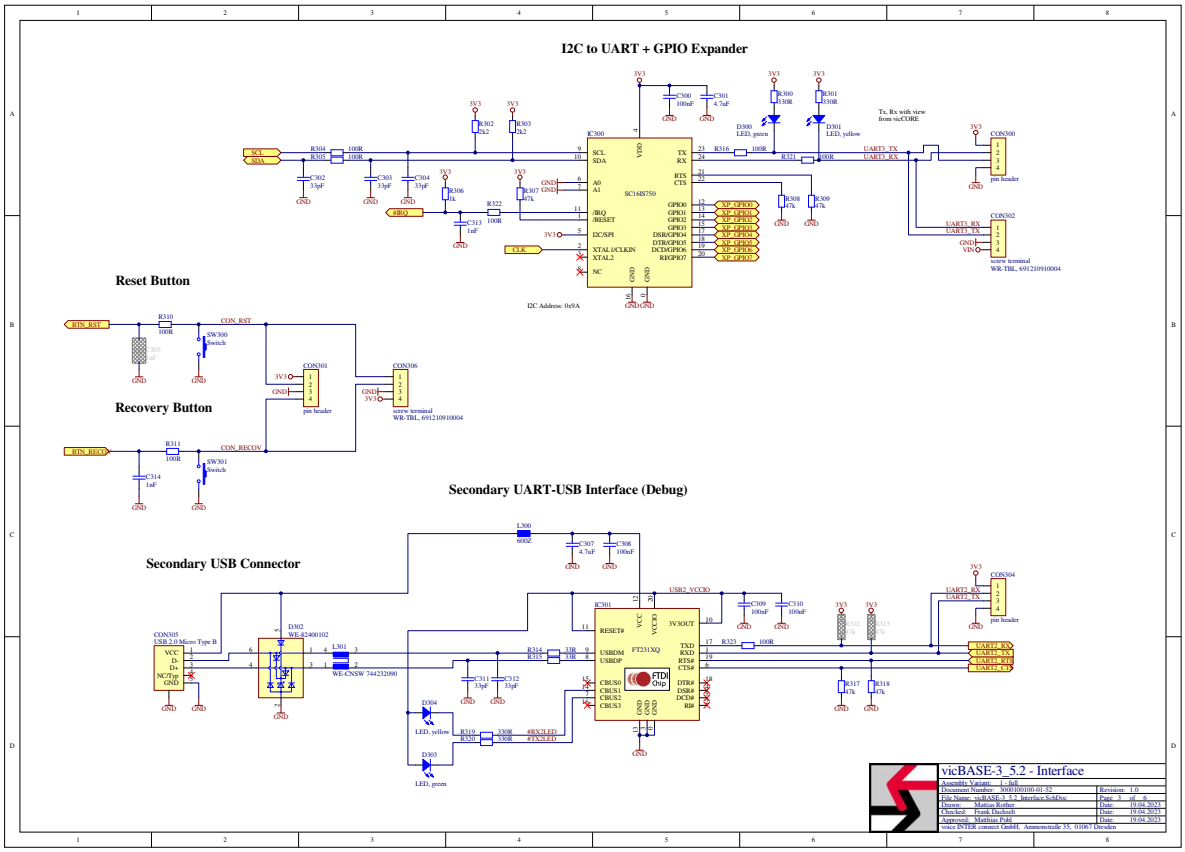
**Important changes**

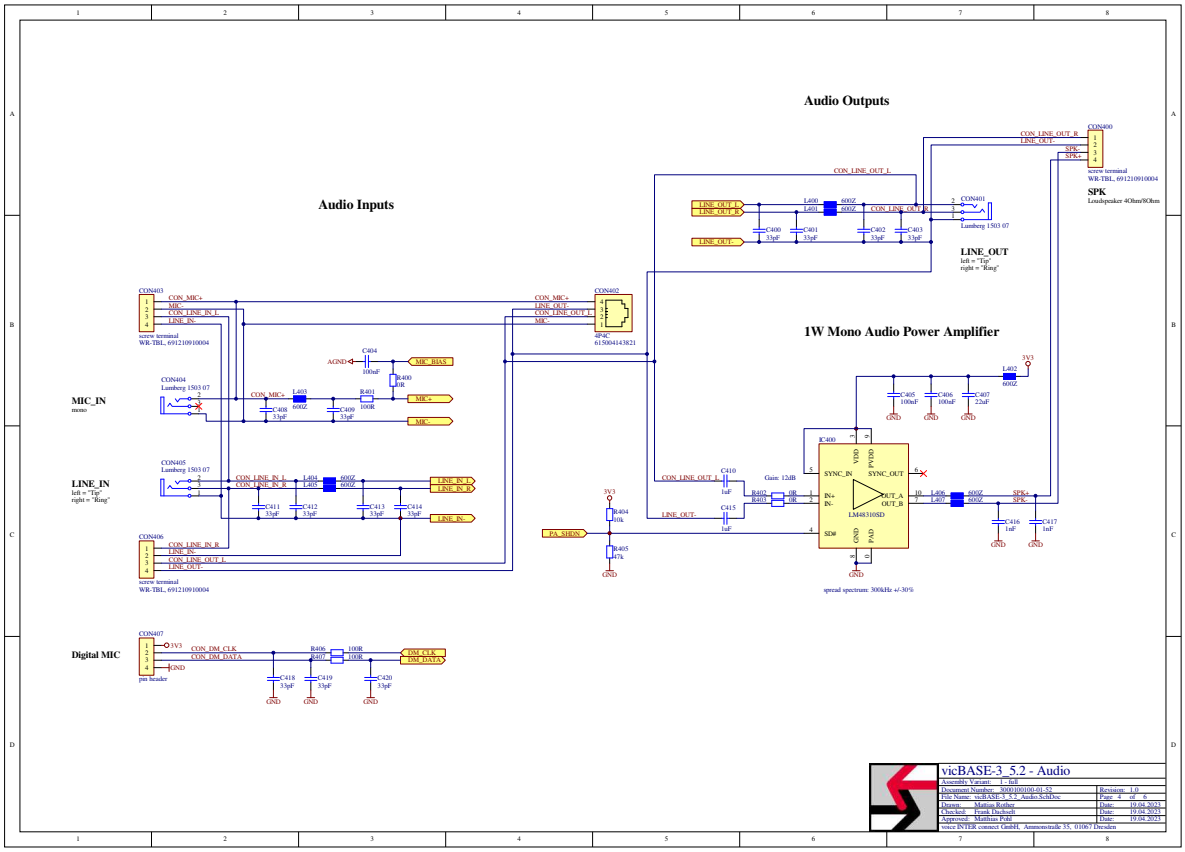
- since 3.9
  - new baseboard named vicBASE-3_5 is supported and always delivered with the coreboard vicCORE-3_2
  - I/O logic of the pins of the vicCORE-3_2 is reversed
  - discontinued support for the coreboard vicCORE-2 and the matching baseboard vicBASE-2 (EMVCR)
  - new languages supported (Arabian, Chinese, Thai)
  - number of available audio prompts for playback reduced to 8
  - slight performance differences are possible
- since 3.1
  - many new languages supported
  - new speech recognition components/engine
    - A slightly reduced speed is noticed on the same hardware platform.
    - Parameters/settings/optimizations might have to be re-tested and set.
  - Changes at the serial protocol
    - The serial protocol is mostly the same as in previous versions. Due to the new engine, timings and messages may have changed based on the particular use case.
    - Unnecessary empty messages and errors are removed.
    - UTF-8 is used instead of ISO8859-1 to support the increased amount of languages available.
    - CR replaces LF as the line ending character. This change provides better compatibility with UTF-8 capable terminal emulators.

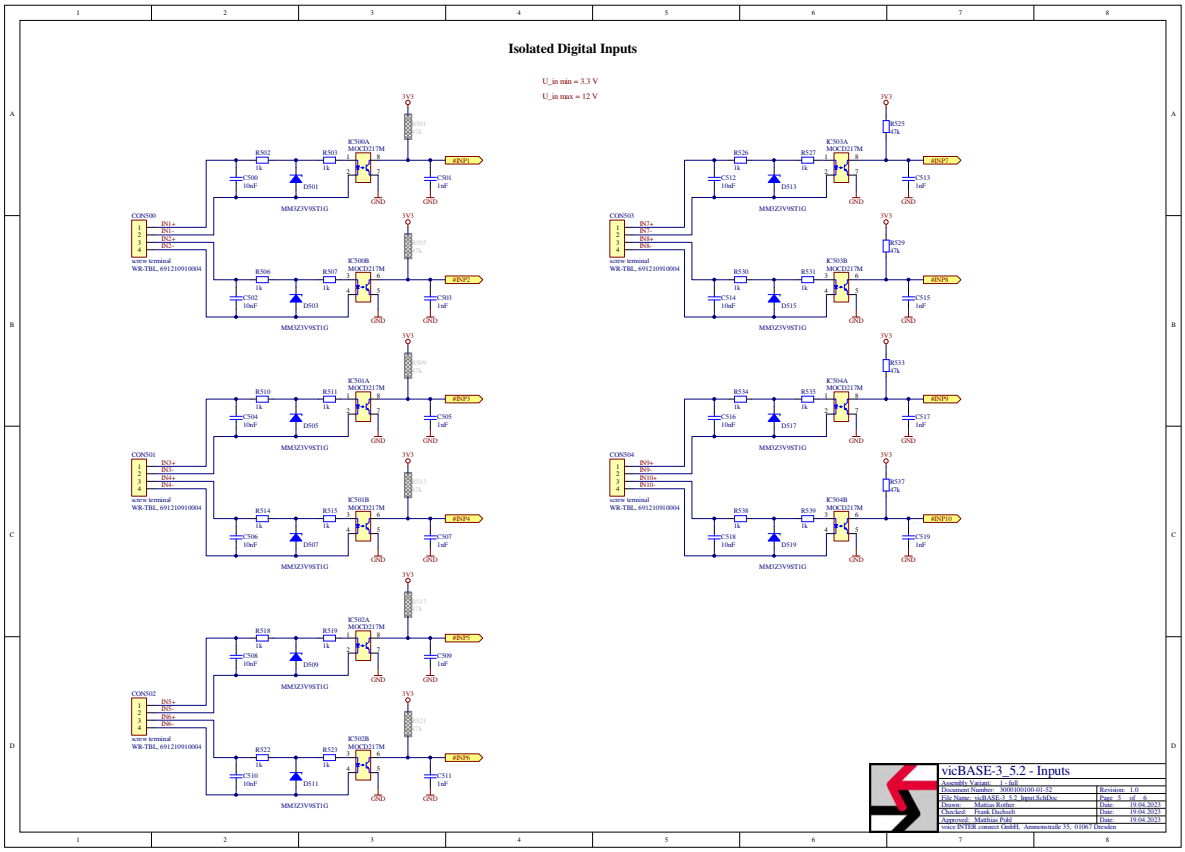
# A vicBASE-3_5.2 Schematic

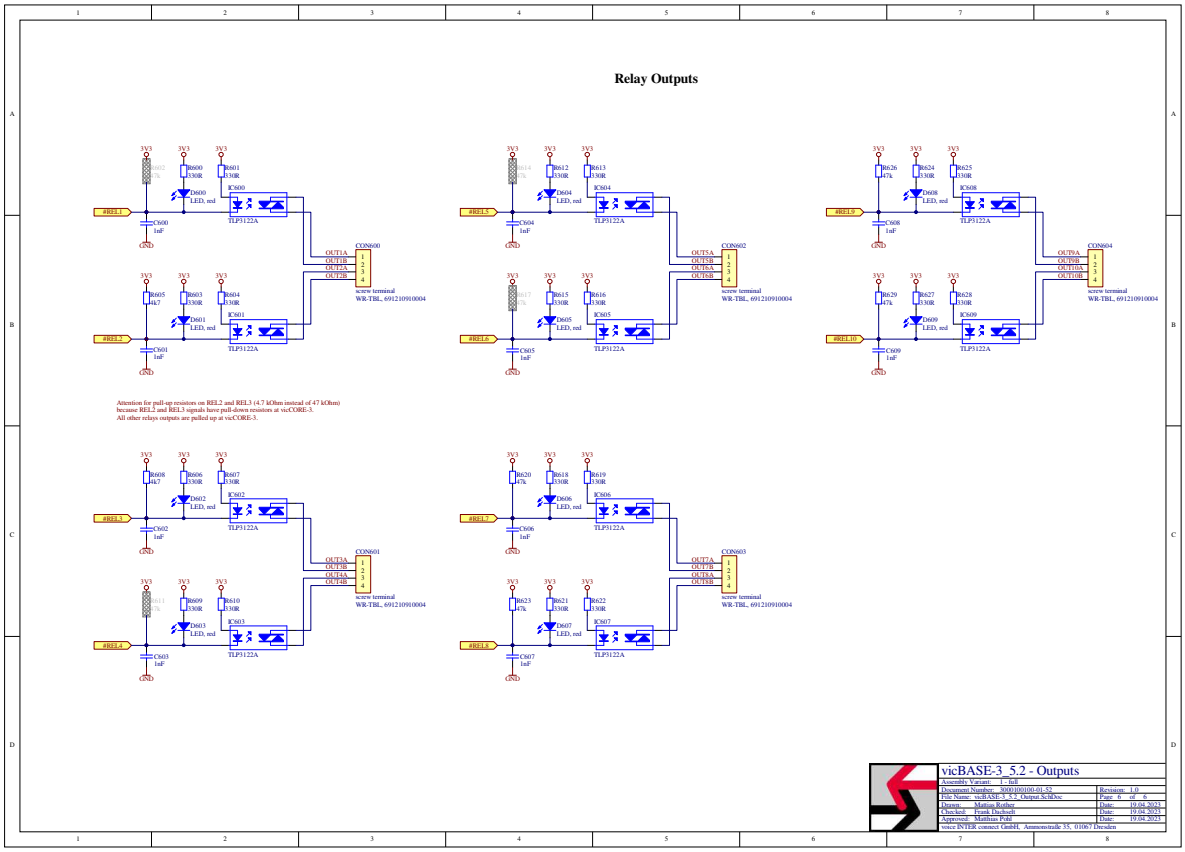



I2C to UART + GPIO Expander
Reset Button
Recovery Button
Secondary UART-USB Interface (Debug)
Secondary USB Connector
vicBASE-3_5.2 - Interface


Audio Outputs
Audio Inputs
1W Mono Audio Power Amplifier
Digital MIC
vicBASE-3_5.2 - Audio


Isolated Digital Inputs
U_in min = 3.3 V
U_in max = 12 V
vicBASE-3_5.2 - Inputs


Relay Outputs
vicBASE-3_5.2 - Outputs