



voice INTER connect

Dokumentation vicCONTROL go

Revision: 12

Datum:

2. Februar 2021

Sachnummer: 1-0010-01-01-12

©voice INTER connect GmbH

Alle Rechte vorbehalten.

Dokument-Historie

Revision	Änderungen	Datum
1	Dokumentation für vicCONTROL go	2019-09-03
2	Erkennbarere Dokumentation der Digital I/Os	2019-09-06
3	Fehlerkorrektur und Formatierung	2019-09-24
4	Dokumentation der Abhängigkeit von "Visual C++ Redistributable für Visual Studio 2012"	2019-10-02
5	Designer Installationsanweisungen aktualisiert	2019-10-08
6	Korrekturen Custom serial output"	2020-03-04
7	CE	2020-03-11
8	Detaillierung Spannungsstecker, Update Problembhebung	2020-05-11
9	Fehlerkorrektur	2020-05-27
10	Dialoggestaltung und Firmware Update-Prozess	2020-09-25
11	vicCORE-3_2 Beschreibung der Audioanschlüsse	2021-01-20
12	Hardware	2021-02-02

Inhaltsverzeichnis

1 Überblick	6
1.1 Lieferumfang	6
1.2 Spracherkennung und Key-Features	6
1.3 Sicherheitshinweis	7
1.4 Elektromagnetischen Konformität	7
1.5 Varianten	8
2 Hardware	10
2.1 vicCORE-3_2	10
2.1.1 Version	10
2.1.2 Betriebswerte	10
2.1.3 Anschlüsse	11
2.1.4 Abmessungen	17
2.1.5 Empfohlenes Footprint (in mm)	17
2.1.6 Reflow Profil	18
2.2 vicBASE-3_5	19
2.2.1 Version	19
2.2.2 Betriebswerte	20
2.2.3 Anschlüsse und Bedienelemente	21
2.2.4 Abmessungen	26
3 Firmware	28
3.1 Auslieferungszustand	28
3.2 Steuerung über das serielle Interface	28
3.2.1 Einstellungen der seriellen Schnittstelle	28
3.2.2 Protokollbeschreibung	28
3.2.3 Nachrichtenaufbau	29
3.2.4 Liste der Befehle, Antworten und Meldungen	30
3.3 Update und Recovery	34
4 USB-Treiber	36
5 vicCONTROL Designer	37
5.1 Installation und Einrichtung	37
5.2 Oberfläche	37
5.2.1 Dialogerstellung	40
5.2.2 Das Platform-Fenster	46
5.3 Aufbau eines Beispieldialogs	47
6 Dialoggestaltung	54
6.1 Allgemeines	54
6.2 Aktivierung	54
6.3 Feedback	54
6.4 One-Shot-Aktivierung	55
6.5 Umgang mit beliebiger ungewollter Sprache (Garbage), begrenzte Schlüsselwortsuche	56

6.6	Erweiterte Spracherkennungseinstellungen	58
7	Problembhebung	59
8	Migration auf vicCONTROL 4	60

1 Überblick

Dieses Dokument beschreibt das Spracherkennungssystem *vicCONTROL go* der Firma *voice INTER connect GmbH*.

1.1 Lieferumfang

vicCONTROL go kit wird mit folgenden Komponenten geliefert:

- Evaluierungsboard *vicBASE-3_5* inkl. *vicCORE-3_2*
- Micro-USB-Kabel
- USB-Stick mit Software und Dokumentation
- Kurzanleitung

vicCONTROL go stamp wird mit folgenden Komponenten geliefert:

- Kernmodul *vicCORE-3_2*

Der USB-Stick enthält:

- die vollständigen Dokumentationsunterlagen
- Beispieldialoge und Audio-Prompts
- Software
 - CRCCalc (Tool zur Erzeugung von Checksummen, wie sie das serielle Protokoll verwendet)
 - FirmwareUpdate (Firmware Update und Recovery Tool)
 - USBDriver (USB-Treiber für *vicBASE-3_5*)
 - *vicCONTROL Designer* (Erzeugen individueller Sprachdialoge für *vicCONTROL go*)
 - *wavesurfer* (Audio-Editor)

1.2 Spracherkennung und Key-Features

- komplett autonome Sprachsteuerung auf der Hardware-Plattform
- Konfiguration ohne Programmierkenntnisse mit Hilfe der mitgelieferten PC-Anwendung *vicCONTROL Designer*
- umfangreiche Auswahl an Plattformsprachen
- frei wählbarer Wortschatz für Command und Control
- sprecherunabhängig, kein Training notwendig
- bis zu je 10 frei programmierbare Digitalein- und -ausgänge

- analoge Audio Ein- und Ausgabe (für das Evaluierungssystem verfügbar: Line-In/Out-, Mikrofon-, Kopfhörer- und Lautsprecheranschlüsse)
- serielle Kommunikation über USB/UART
- Ausgabe von Audio-Prompts

1.3 Sicherheitshinweis

Die Spracherkennung ermöglicht zusätzliche Steuerungsmöglichkeiten und funktioniert mit Hilfe statistischer Methoden. Dafür muss sie sich an ständig ändernde Umgebungsbedingungen anpassen. Es kann nie völlig ausgeschlossen werden, dass eine Phrase einmal verzögert oder falsch oder gar nicht erkannt wird. Ebenso kann es zu spontanen Fehlerkennungen kommen, bei denen eine gesprochene Phrase erkannt wird, obwohl sie nicht gesagt wurde. *vicCONTROL go* darf daher nicht allein/ohne unabhängige Sicherheitsmaßnahmen eingesetzt werden, wenn die Anwendung sicherheitskritisch ist.

1.4 Elektromagnetischen Konformität

Erklärung der elektromagnetischen Konformität des „*vicCONTROL go*“-Produktes

CE

„*vicCONTROL go*“-Sprachsteuerungen (im Folgenden Produkte) sind für den Einbau in elektrische Geräte oder als dedizierte Evaluation Boards (d.h.: für den Einsatz als Test- und Prototyp-Plattform für die Hardware-/Softwareentwicklung) in Laborumgebungen konzipiert.

„*vicCONTROL go*“-Produkte ohne Schutzgehäuse sind ESD empfindlich und müssen daher an ESD geschützten Arbeitsplätzen von geschultem Fachpersonal ausgepackt und gehandhabt bzw. verarbeitet werden. Darüber hinaus sollten „*vicCONTROL go*“-Produkte nicht ohne Schutzschaltung betrieben werden, wenn die Verbindungen zu den Kontakten des Produkts länger als 3 m sind.

„*vicCONTROL go*“-Produkte erfüllen die Normen der Richtlinie der Europäischen Union für elektromagnetische Konformität gemäß den in diesem Hardware-Handbuch angegebenen Beschreibungen und Verwendungsregeln (insbesondere in Bezug auf die Kontakte, Steckverbinder, Anschlüsse und Schnittstellen zu einem Host-PC oder sonstigem Host-Prozessor).

Die Implementierung von „*vicCONTROL go*“-Produkten in die Zielgeräte sowie Anwendermodifikationen und -erweiterungen von „*vicCONTROL go*“-Produkten unterliegen der erneuten Feststellung der Konformität und Zertifizierung der elektro-magnetischen Richtlinien. Benutzer sollten die Konformität nach allen Änderungen an einem Produkt sowie die Implementierung eines Produkts in Zielsysteme sicherstellen.

1.5 Varianten

vicCONTROL go kit

Die Standardvariante besteht aus der Trägerplatine vicBASE-3_5 (Abbildung 1), welche als Evaluierungsplattform für das vicCONTROL go stamp dient. Die umfangreiche Ausstattung erlaubt es, direkt eigene Anwendungen mit einer Sprachsteuerung umzusetzen. Es eignet sich zur Evaluierung ihrer Spracherkennungsanwendung sowie zur Umsetzung von Prototypen und Kleinserien.

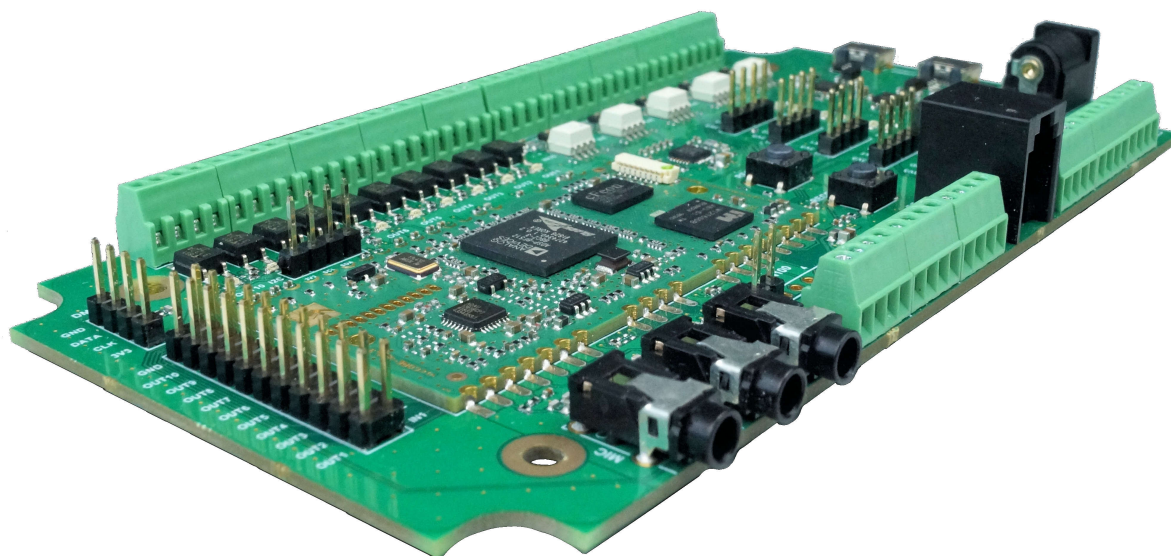


Abbildung 1: vicBASE-3_5

Das entsprechende Blockschaltbild ist in Abbildung 2 dargestellt.

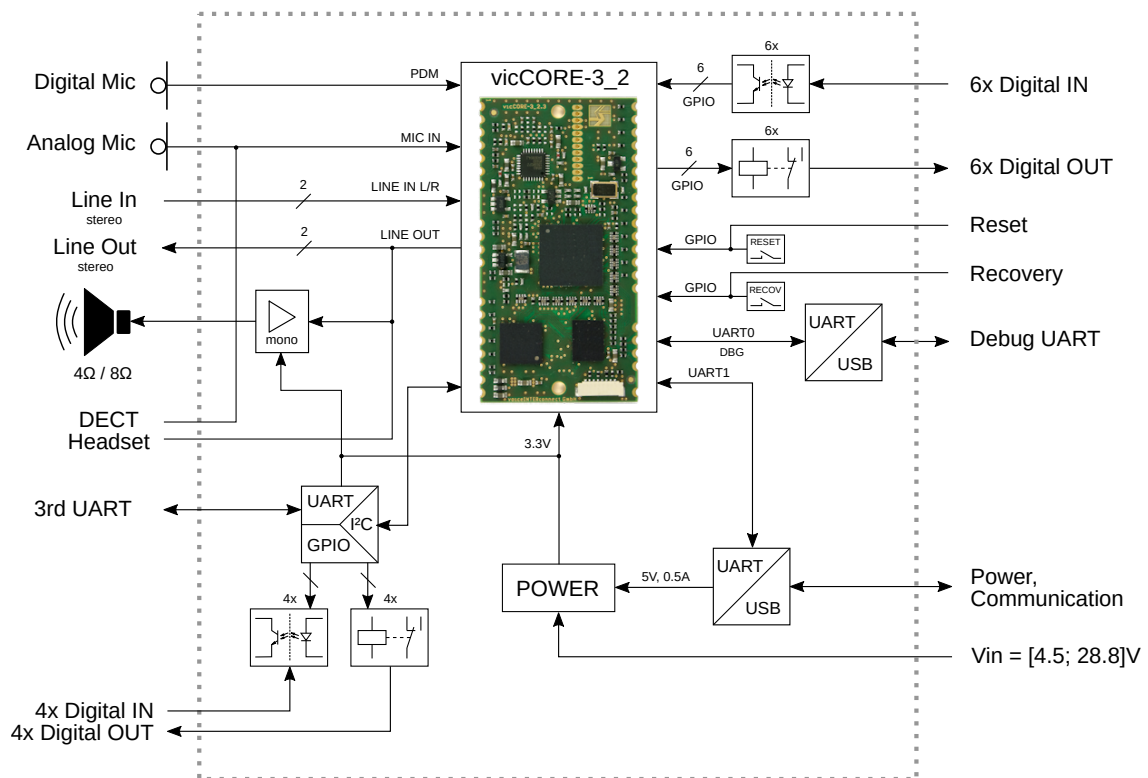


Abbildung 2: Blockschaltbild vicBASE-3_5

vicCONTROL go stamp

Die Variante *vicCONTROL go stamp* verwendet das *vicCORE-3_2*, welches sich dank seiner Randkontakte leicht in eigene Schaltungen integrieren lässt.

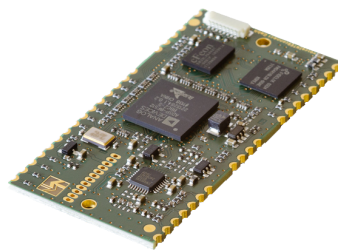


Abbildung 3: vicCORE-3_2

Bitte kontaktieren Sie uns für weitere Produktvarianten.

2 Hardware

2.1 vicCORE-3_2

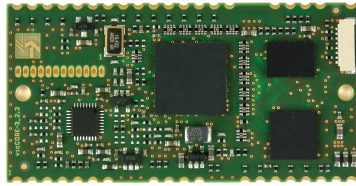


Abbildung 4: vicCORE-3_2

2.1.1 Version

- Bezeichnung: vicCORE-3_2
- Boardversion: 2.3

2.1.2 Betriebswerte

Grenzwerte

Symbol	Name	Wert	Einheit
T_{STORE}	Lagertemperaturbereich	-65 ... 120	°C
V_{CP}	Arbeitstemperaturbereich	-40 ... 85	°C
FC	Brennbarkeitsklasse nach UL94	V-0	
V_{DD}	Betriebsspannung	2,7 ... 3,6	V

Betriebswerte

Symbol	Name	Min	Nom	Max	Einheit
V_{DD}	Betriebsspannung	3,0	3,3	3,5	V
I_{DD}	Stromaufnahme	80		110	mA
U_{MIC}	Spannung am Mikrofoneingang MIC_IN_R (bei V MIC = 20 dB)	0	30	100	mV _{RMS}
U_{IN}	Spannung an LINE_IN_R und LINE_IN_L	0	1	1	V _{RMS}
U_{OUT}	Spannung an LINE_OUT_R und LINE_OUT_L	0	1	1	V _{RMS}
U_{RESET}	Spannung Reset-Eingang	0	3,3	V_{DD}	V

2.1.3 Anschlüsse

Audioeingänge

Die Eingangsimpedanz der Audioeingänge ergibt sich aus der Innenschaltung des vicCORE-3_2 (siehe Abbildung 5). Dabei haben die verstärkungsabhängigen Eingangswiderstände des Audio-Codex einen erheblichen Einfluss (Verstärkungen einstellbar mit GainMic und GainLineIn). Der Zusammenhang zwischen den Verstärkungsfaktoren und den Eingangswiderständen ist in der darauffolgenden Tabelle beschrieben. Der maximal erlaubte Gleichspannungsoffset beträgt 16 V. Da die Verstärkung derzeit auf 35 dB festgesetzt ist, beträgt die Eingangsimpedanz 2,0 Kiloohm.

Für die Nutzung von Elektretmikrofonen wird am MIC_BIAS Anschluss eine Vorspannung bereitgestellt. Ein 2,2 Kiloohm Vorwiderstand für den MIC_BIAS Anschluss ist bereits auf dem vicCORE-3_2 Board enthalten, was bedeutet das er für ein Elektretmikrofon direkt mit MIC_IN_R verbunden werden kann.

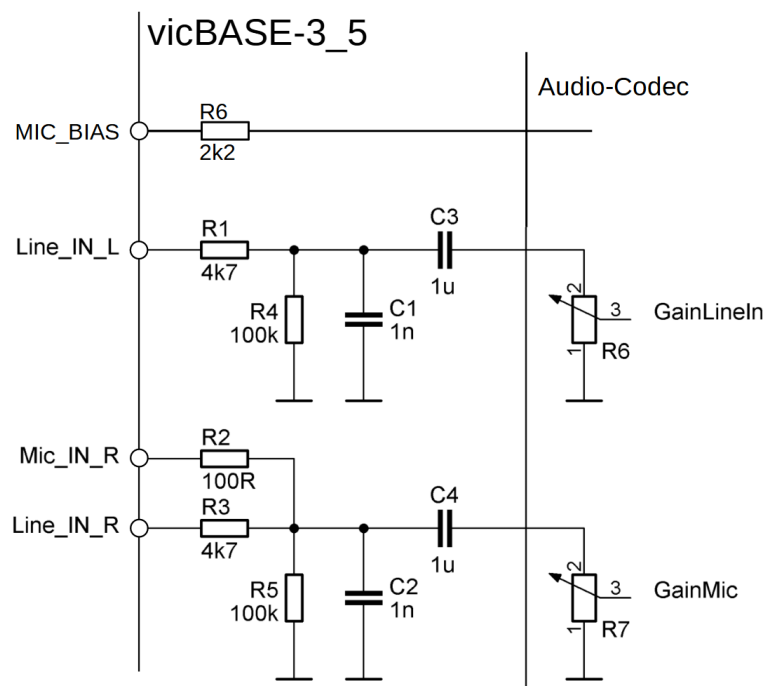


Abbildung 5: vicCORE-3_2 – Audioeingänge

Die dargestellten Bauelemente sind bereits auf dem vicCORE-3 Board bestückt

Eingangswiderstände R6, R7

V in dB	R _{in} k
-12	85
0	53
25	5,6
35	2,0

Audioausgänge

Die Ausgangsimpedanz wird vom niederohmigen Ausgangswiderstand des Audio-Codex bestimmt, der im Datenblatt nicht näher spezifiziert wird. Der maximal erlaubte Gleichspannungsoffset beträgt 16 V.

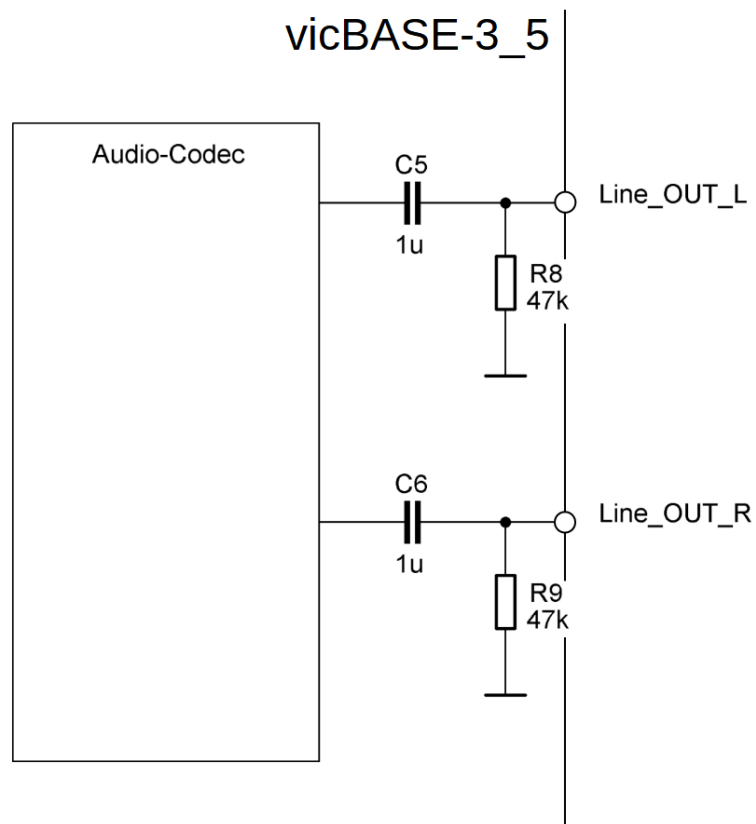


Abbildung 6: vicCORE-3_2 – Audioausgänge

Randkontakte

Alle Anschlüsse des vicCORE-3_2 sind als Randkontakte zum Bestücken auf eine Trägerplatine ausgeführt (siehe X101 – X146 in Abbildung 7). Alternativ kann ein Teil der Anschlüsse über den Stegleitungsanschluss kontaktiert werden (siehe CON101 in Abbildung 8).

Die Eingänge IN_1 bis IN_6 sowie RECOV und RESET verhalten sich Low-Aktiv.

Die Terminierung der Ein- und Ausgänge erfolgt mittels 47 Kiloohm Widerständen.

Firmware Updates und die Verbindung mit dem vicCONTROL Designer erfolgen über die Kommandoschnittstelle (RX1/Tx1).

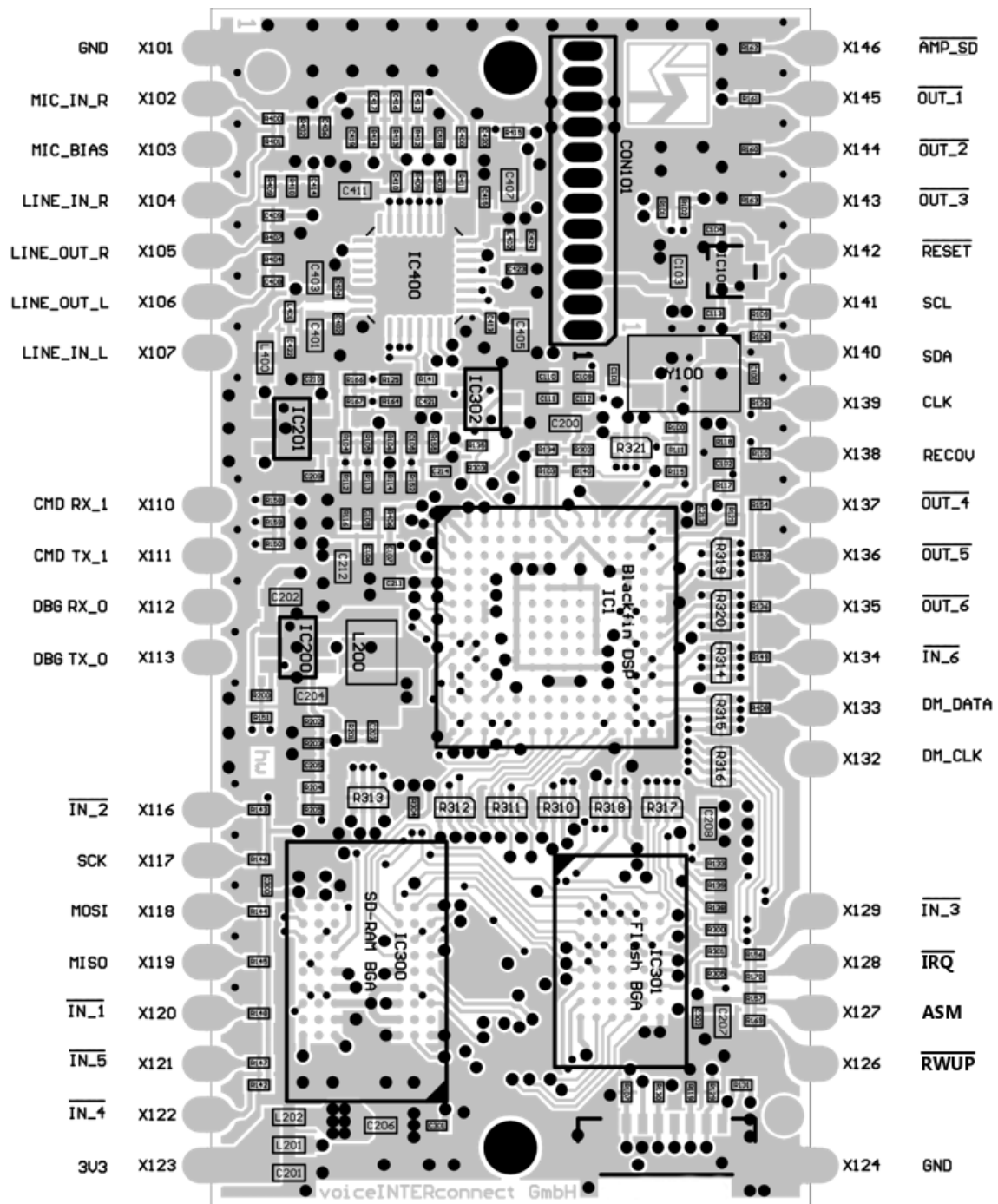


Abbildung 7: Anschlüsse Randkontakte

Anschluss	Art	Term.	Bezeichnung	Beschreibung
X101	Eingang		GND	Masse für Audioanschlüsse X102 – X107
X102	Eingang		MIC_IN_R	Mikrofoneingang für Elektretmikrofon

X103	Ausgang		MIC_BIAS	Bias-Spannung (3 V) Ein 2,2 Kiloohm Vorwiderstand ist bereits auf dem vicCORE-3_2 Board enthalten. Kann für Elektretmikrofon direkt mit MIC_IN_R verbunden werden
X104	Eingang		LINE_IN_R	zzt. nicht genutzt
X105	Ausgang		LINE_OUT_R	Audiosignal Ausgabe des aktuellen Eingangssignals
X106	Ausgang		LINE_OUT_L	Audiosignal Ausgabe der Prompts
X107	Eingang		LINE_IN_L	Optionaler Mikrofoneingang (Line-Pegel)
X110	Eingang	Pull-up	RX_1	UART-1-RX-Signal Kommandoschnittstelle
X111	Ausgang	Pull-up	TX_1	UART-1-TX-Signal Kommandoschnittstelle
X112	Eingang	Pull-up	RX_0	UART-0-RX-Signal Debugschnittstelle
X113	Ausgang	Pull-up	TX_0	UART-0-TX-Signal Debugschnittstelle
X116	Eingang	Pull-up	IN_2	Eingang 2
X117	Ausgang	Pull-up	SCK	zzt. nur für internen Gebrauch
X118	Ausgang	Pull-up	MOSI	zzt. nur für internen Gebrauch
X119	Eingang	Pull-up	MISO	zzt. nur für internen Gebrauch
X120	Eingang	Pull-up	IN_1	Eingang 1
X121	Eingang	Pull-up	IN_5	Eingang 5
X122	Eingang	Pull-up	IN_4	Eingang 4
X123	Eingang		3V3	Versorgungsspannung (+)
X124	Eingang		GND	Versorgungsspannung (Masse)
X126	Eingang	Pull-up	IRQ	zzt. nur für internen Gebrauch
X127	Eingang	Pull-up	ASM	zzt. nur für internen Gebrauch
X128	Ausgang	Pull-up	RWUP	zzt. nur für internen Gebrauch
X129	Eingang	Pull-up	IN_3	Eingang 3
X132	Ausgang		DM_CLK	zzt. nur für internen Gebrauch
X133	Eingang	Pull-up	DM_DATA	zzt. nur für internen Gebrauch

X134	Eingang	Pull-up	IN_6	Eingang 6
X135	Ausgang	Pull-up	OUT_6	Ausgang 6
X136	Ausgang	Pull-up	OUT_5	Ausgang 5
X137	Ausgang	Pull-up	OUT_4	Ausgang 4
X138	Eingang	Pull-up	RECOV	Recovery Konfiguration
X139	Ausgang		CLK	Clock 18,432 MHz
X140	Ein-/Ausgang	Pull-up	SDA	I2C Data
X141	Ein-/Ausgang	Pull-up	SCL	I2C Clock
X142	Eingang	Pull-up	RESET	Hardware-Reset
X143	Ausgang	Pull-down	OUT_3	Ausgang 3
X144	Ausgang	Pull-down	OUT_2	Ausgang 2
X145	Ausgang	Pull-up	OUT_1	Ausgang 1
X146	Ausgang	Pull-down	AMP_SD	Aktivität des Audioverstärkers (amplifier shut down)

Stegleitung

Der 12-polige Stegleitungsanschluss ist in Abbildung 8 nochmals dargestellt. Die Nummerierung beginnt von der Platinenmitte her mit Pin 1. Das Raster beträgt 1,27 mm.

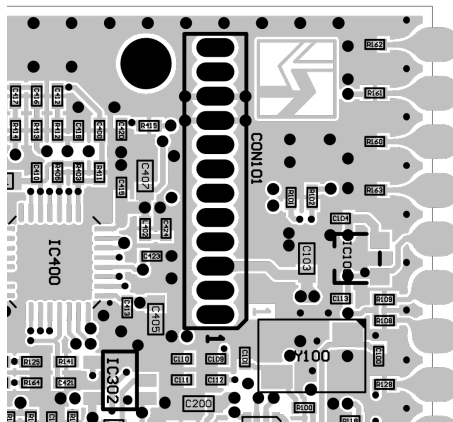


Abbildung 8: Stegleitung

Anschluss	Art	Bezeichnung	Beschreibung
Pin 1	Ein-/Ausgang	SDA	I2C
Pin 2	Ein-/Ausgang	SCL	I2C
Pin 3	Eingang	3V3	Versorgungsspannung
Pin 4	Eingang	LINE_IN_L	Optionaler Mikrofoneingang (Line-Pegel)
Pin 5	Ausgang	LINE_OUT_L	Audiosignal Ausgabe der Prompts
Pin 6	Ausgang	LINE_OUT_R	Audiosignal Ausgabe des aktuellen Eingangssignals
Pin 7	Ausgang	RX_1	UART-1-RX-Signal Kommandoschnittstelle
Pin 8	Eingang	TX_1	UART-1-TX-Signal Kommandoschnittstelle
Pin 9	Eingang	GND	Masse für Versorgungsspannung
Pin 10	Eingang	GND	Masse für Audioanschlüsse
Pin 11	Ausgang	MIC_BIAS	Bias-Spannung für Elektretmikrofon (3,3 V; 2,2 Kiloohm)
Pin 12	Eingang	MIC_IN_R	Mikrofoneingang für Elektretmikrofon

2.1.4 Abmessungen

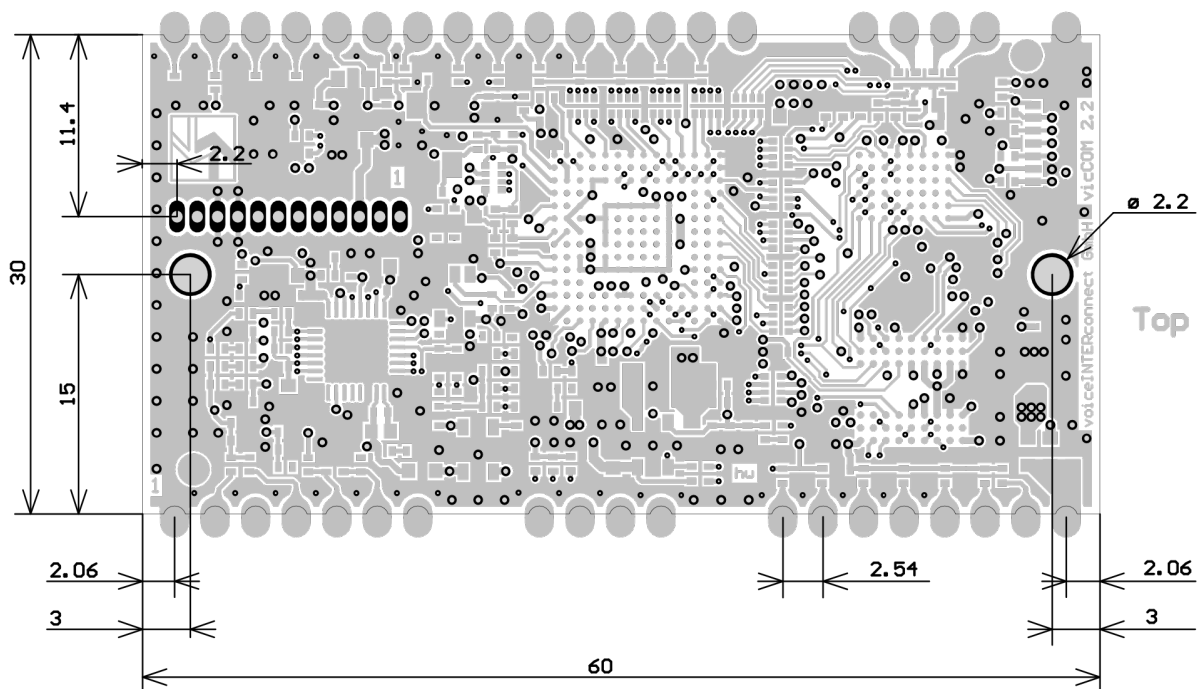


Abbildung 9: Abmessungen (in mm)

2.1.5 Empfohlenes Footprint (in mm)

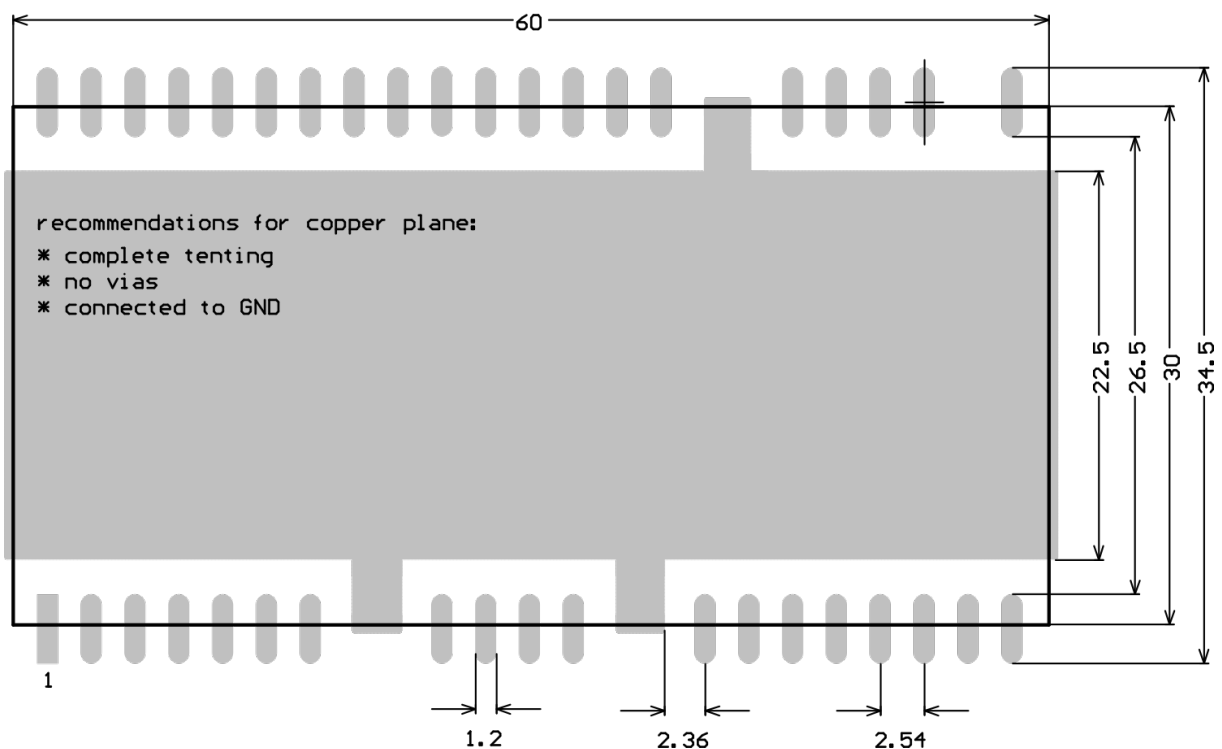


Abbildung 10: Empfohlenes Footprint (alle Angaben in mm)

2.1.6 Reflow Profil

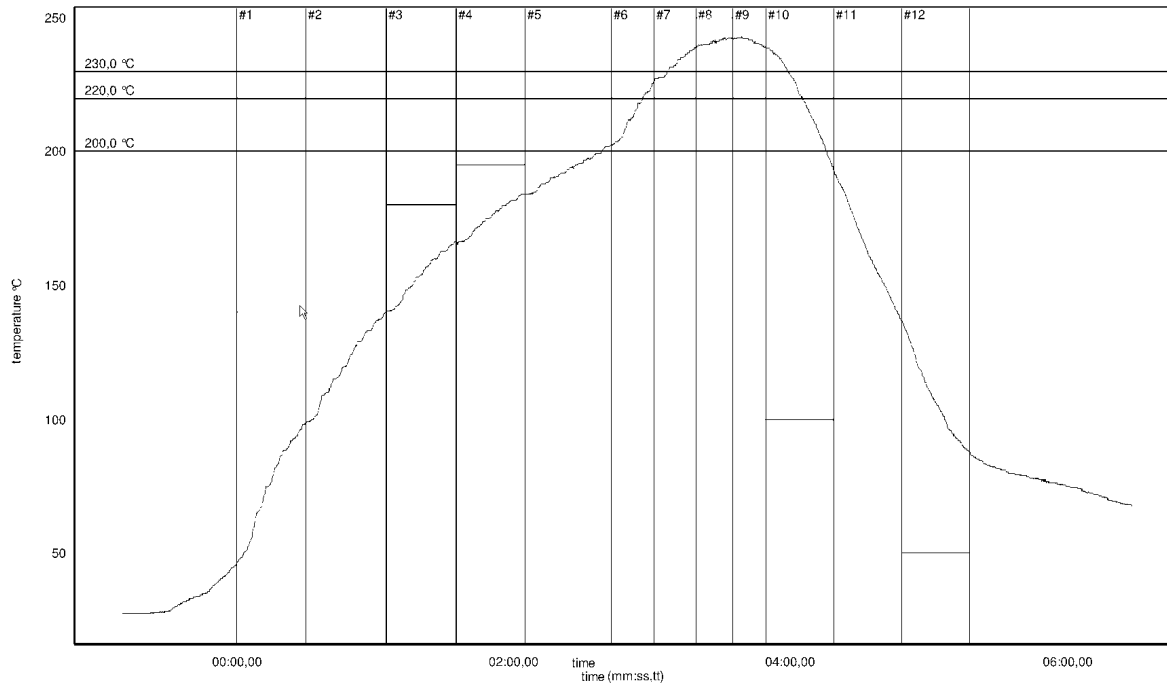


Abbildung 11: Reflow Profil für vicCORE-3_2

2.2 vicBASE-3_5

Die Trägerplatte für das vicCORE-3_2 bietet, neben der Energieversorgung und Kommunikation über USB, ein umfangreiches Audiointerface und verschiedene Anschlussmöglichkeiten für bis zu je 10 digitale Ein- und Ausgänge.

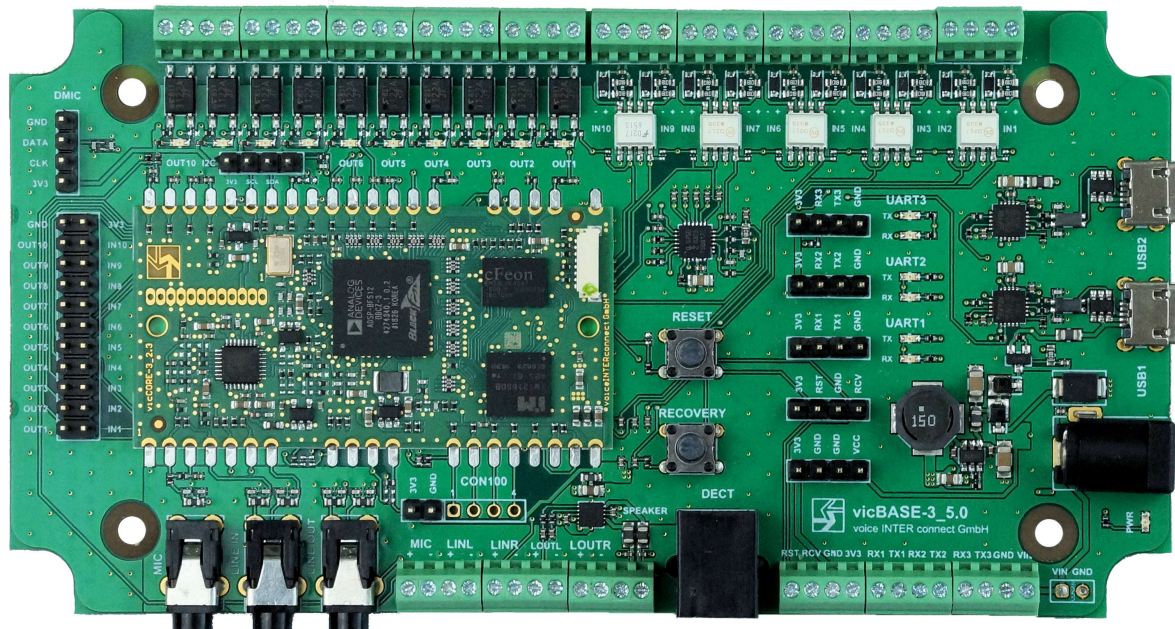


Abbildung 12: vicBASE-3_5

2.2.1 Version

- Bezeichnung: vicBASE-3_5
- Boardversion: 5.1

2.2.2 Betriebswerte

Symbol	Name	Min	Nom	Max	Einheit
V_{CC}	Betriebsspannung	4,5	24	28	V
P_{in}	Leistungsaufnahme, Stand-By, Ausgänge inaktiv		0,35		W
	Leistungsaufnahme, alle Ausgänge aktiv		0,7		W
	Leistungsaufnahme, alle Ausgänge aktiv, Sinus-Tonausgabe ($R_L = 8 \text{ Ohm}$), Lautstärke 100 %		2,3		W
	Leistungsaufnahme, alle Ausgänge aktiv, Sinus-Tonausgabe ($R_L = 4 \text{ Ohm}$), Lautstärke 100 %		3,7		W
R_L	Lautsprecherimpedanz	4	8		
P_{SPK}	Ausgangsleistung Lautsprecher Ausgang ($R_L = 8 \text{ Ohm}$)			0,6	W
	Ausgangsleistung Lautsprecher Ausgang ($R_L = 4 \text{ Ohm}$)			1,1	W
U_{LineIn}	Spannung an Line_IN		1	1	V_{RMS}
$U_{LineOut}$	Spannung an Line_OUT		1	1	V_{RMS}
U_{Mic}	Spannung am Mikrofoneingang		30	100	mV_{RMS}
U_{In}	Spannung für galvanisch getrennte Digitaleingänge (Optokoppler)	3,3		12	V
U_{Rel}	Zulässige Spannung an Digitalausgängen (off-state)			60*	V
	Empfohlene Spannung an Digitalausgängen (off-state)			48	V
I_{Rel}	Strom durch Digitalausgänge (on-state)			1,4	A
I_{3V3}	Stromtragfähigkeit 3,3 V-Ausgang			500	mA

* Die dauerhafte Verwendung der Digitalausgängen, mit der maximal zulässigen Spannung, kann die Zuverlässigkeit des Produktes verringern.

2.2.3 Anschlüsse und Bedienelemente

Die Lage und Bezeichnungen der Anschlüsse und Bedienelemente können Abbildung 13 entnommen werden.

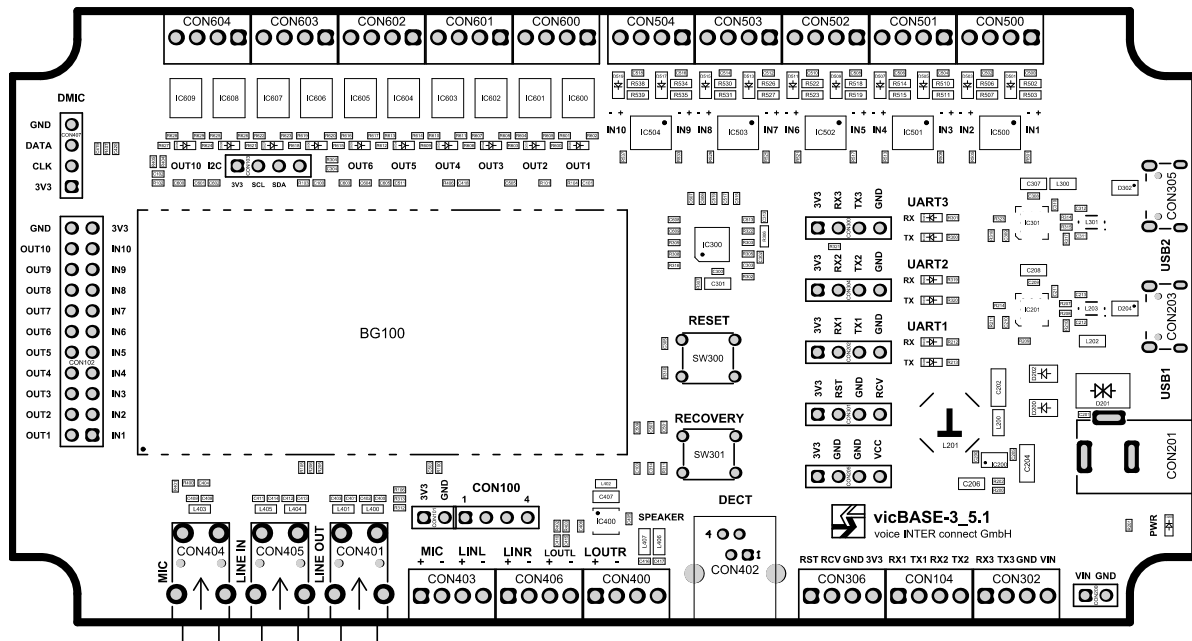


Abbildung 13: vicBASE-3_5 – Anschlüsse

Anschluss	Steckverbindung	Beschriftung	vicCORE-3_2 Anschluss	Beschreibung
CON100	Stiffleiste, 4pol.	CON100_1	IN_2	Eingang 2 (konfigurierbar als SPISEL, wenn R100 entfernt wird)
		CON100_2	SCK	SPI Clock (nicht verwendet)
		CON100_3	MOSI	SPI Master Out Slave In (nicht verwendet)
		CON100_4	MISO	SPI Master In Slave Out (nicht verwendet)
CON101	Stiffleiste, 2pol.	3V3	3V3	3,3 V Spannungsversorgung
		GND	GND	Bezugspotential
CON102	Stiffleiste, 22pol.	OUT1	OUT_1	Ausgang 1
		OUT2	OUT_2	Ausgang 2
		OUT3	OUT_3	Ausgang 3
		OUT4	OUT_4	Ausgang 4
		OUT5	OUT_5	Ausgang 5
		OUT6	OUT_6	Ausgang 6

		OUT7	I2C	Ausgang 7
		OUT8	I2C	Ausgang 8
		OUT9	I2C	Ausgang 9
		OUT10	I2C	Ausgang 10
		IN1	IN_1	Eingang 1
		IN2	IN_2	Eingang 2
		IN3	IN_3	Eingang 3
		IN4	IN_4	Eingang 4
		IN5	IN_5	Eingang 5
		IN6	IN_6	Eingang 6
		IN7	I2C	Eingang 7
		IN8	I2C	Eingang 8
		IN9	I2C	Eingang 9
		IN10	I2C	Eingang 10
		GND	GND	Bezugspotential
		3V3	3V3	3,3 V Spannungsversorgung
CON103	Stiflleiste, 4pol.	I2C_3V3	3V3	3,3 V Spannungsversorgung
		I2C_SCL	SCL	I2C Clock (nicht verwendet)
		I2C_SDA	SDA	I2C Data (nicht verwendet)
		I2C_GND	GND	Bezugspotential
CON104	Schraubklemme	RX1	RX_1	UART1 Empfangsleitung für Kommando-Schnittstelle
		TX1	TX_1	UART1 Sendeleitung für Kommando-Schnittstelle
		RX2	RX_0	UART2 Empfangsleitung für Debug-Schnittstelle
		TX2	TX_0	UART2 Sendeleitung für Debug-Schnittstelle
CON200	Stiflleiste, 2pol.	VIN		Eingangsspannungsversorgung
		GND	GND	Bezugspotential
CON201	Hohlbuchse (6,4/2,5 mm)	VIN		Eingangsspannungsversorgung für 5,5/2,5 mm Stecker, Bezugspotential (Masse) außen
CON202	Stiflleiste, 4pol.	3V3	3V3	3,3 V Spannungsversorgung

		RX1	RX_1	UART1 Empfangsleitung für Kommando-Schnittstelle
		TX1	TX_1	UART1 Sendeleitung für Kommando-Schnittstelle
		GND	GND	Bezugspotential
CON203	Micro-USB	USB1	RX_1/TX_1	Energieversorgung und Kommunikation mit der Kommando-Schnittstelle über USB
CON205	Stiffliste, 4pol.	3V3	3V3	3,3 V Spannungsversorgung
		GND	GND	Bezugspotential
		GND	GND	Bezugspotential
		VCC		Eingangsspannung nach Schutzschaltung und Filterung
CON300	Stiffliste, 4pol.	3V3	3V3	3,3 V Spannungsversorgung
		RX3		UART3 Empfangsleitung (zzt. nicht verwendet)
		TX3		UART3 Sendeleitung (zzt. nicht verwendet)
		GND	GND	Bezugspotential
CON301	Stiffliste, 4pol.	3V3	3V3	3,3 V Spannungsversorgung
		RST	RESET	Hardware Reset (low active)
		GND	GND	Bezugspotential
		RCV	RECOV	Recovery Konfiguration (low active)
CON302	Schraubklemme	RX3		UART3 Empfangsleitung (zzt. nicht verwendet)
		TX3		UART3 Sendeleitung (zzt. nicht verwendet)
		GND	GND	Bezugspotential
		VIN		Eingangsspannungsversorgung
CON304	Stiffliste, 4pol.	3V3		3,3 V Spannungsversorgung
		RX2	RX_0	UART2 Empfangsleitung für Debug-Schnittstelle
		TX2	TX_0	UART2 Sendeleitung für Debug-Schnittstelle
		GND	GND	Bezugspotential

CON305	Micro-USB	USB2	RX_0/TX_0	Kommunikation mit der Debug-Schnittstelle über USB
CON306	Schraubklemme	RST	RESET	Hardware Reset (low active)
		RCV	RECOV	Recovery Konfiguration (low active)
		GND	GND	Bezugspotential
		3V3	3V3	3,3 V Spannungsversorgung
CON400	Schraubklemme	LOUTR	LINE_OUT_R	Ausgabe des aktuellen Eingangssignals
		SPEAKER		Audioausgang mit 12 dB Verstärkung des LINE_OUT_L Signals
CON401	3,5 mm Klinkenbuchse	LINE OUT	LINE_OUT_R/L	niederohmiger Line-Out-Ausgang zur Ausgabe des Prompts (L) und des aktuellen Eingangssignals (R)
CON402	4P4C-Buchse	DECT	MIC_IN_R, LINE_OUT_L	Anschluss für DECT-Headset
CON403	Schraubklemme	MIC	MIC_IN_R	Mikrofoneingang für Elektretmikrofon
		LINL	LINE_IN_L	hochohmiger Line-In-Eingang zur Einspeisung analoger Audiosignale
CON404	3,5 mm Klinkenbuchse	MIC	MIC_IN_R	Mikrofoneingang für Elektretmikrofon
CON405	3,5 mm Klinkenbuchse	LINE IN	LINE_IN_R/L	hochohmiger Line-In-Eingang zur Einspeisung analoger Audiosignale (L)
CON406	Schraubklemme	LINR	LINE_IN_R	zzt. nicht genutzt
		LOUTL	LINE_OUT_L	Ausgabe des Prompts
CON407	Stiftleiste, 4pol.	DMIC_GND	GND	Bezugspotential
		DMIC_DATA	DM_DATA	Digitalmikrofon Datenleitung (nicht verwendet)
		DMIC_CLK	DM_CLK	Digitalmikrofon Clock (nicht verwendet)
		DMIC_3V3	3V3	3,3 V Spannungsversorgung
CON500	Schraubklemme	IN1	IN_1	Eingang 1 (potentialfrei)
		IN2	IN_2	Eingang 2 (potentialfrei)
CON501	Schraubklemme	IN3	IN_3	Eingang 3 (potentialfrei)

		IN4	IN_4	Eingang 4 (potentialfrei)
CON502	Schraubklemme	IN5	IN_5	Eingang 5 (potentialfrei)
		IN6	IN_6	Eingang 6 (potentialfrei)
CON503	Schraubklemme	IN7	I2C	Eingang 7 (potentialfrei)
		IN8	I2C	Eingang 8 (potentialfrei)
CON504	Schraubklemme	IN9	I2C	Eingang 9 (potentialfrei)
		IN10	I2C	Eingang 10 (potentialfrei)
CON600	Schraubklemme	OUT1	OUT_1	Ausgang 1 (potentialfrei)
		OUT2	OUT_2	Ausgang 2 (potentialfrei)
CON601	Schraubklemme	OUT3	OUT_3	Ausgang 3 (potentialfrei)
		OUT4	OUT_4	Ausgang 4 (potentialfrei)
CON602	Schraubklemme	OUT5	OUT_5	Ausgang 5 (potentialfrei)
		OUT6	OUT_6	Ausgang 6 (potentialfrei)
CON603	Schraubklemme	OUT7	I2C	Ausgang 7 (potentialfrei)
		OUT8	I2C	Ausgang 8 (potentialfrei)
CON604	Schraubklemme	OUT9	I2C	Ausgang 9 (potentialfrei)
		OUT10	I2C	Ausgang 10 (potentialfrei)
SW300	Button	RESET	RESET	Hardware Reset (low active)
SW301	Button	RECOVERY	RECOV	Recovery Konfiguration (low active)

Es sind 2 USB-Anschlüsse verfügbar, welche beide als Micro-USB-Buchse ausgeführt sind. Der Anschluss USB1 kann zusätzlich zur Kommunikation mit dem vicBASE-3_5 zur alleinigen Energieversorgung dienen, wenn ein 8 Ohm Lautsprecher als Ausgabe verwendet wird. Die spezifizierte Stromtragkraft von USB-2.0 mit 500 mA ist allerdings nicht mehr ausreichend für den Anschluss eines 4 Ohm Lautsprechers bei voller Ausgabelautstärke (siehe Tabelle Betriebswerte).

Alternativ steht eine Hohlbuchse oder eine Schraubklemme mit einem Eingangsspannungsbereich von 4,5 V bis 24 V für die Energieversorgung zur Verfügung.

Die Versorgung über ein USB-Steckernetzteil mit mind. 800 mA ist möglich. Ohne Nutzung des Audioverstärkers sind auch schwächere Netzteile ab 200 mA verwendbar.

Für Mikrofon (MIC), Kopfhörer (LINE OUT) und Audioeingang (LINE IN) stehen je 3,5 mm Klinkenbuchsen zur Verfügung. Die Signale sind auf Schraubklemmen ausgeführt. Ein integrierter 1 W Audioverstärker steht für den Anschluss eines Lautsprechers mit einer Impedanz von 4 oder 8 Ohm bereit. Die Verstärkung beträgt 12 dB. Der Lautsprecher wird an die Schraubklemme (SPEAKER) angeschlossen.

Für den Mikrofoneingang können handelsübliche Kondensator-Elektret-Mikrofone verwendet werden (z.B. Plantronics Audio 40, Desktopmikrofone oder Standardheadsets).

Zur Verwendung einer kabellosen, freien, entfernten Spracherkennung kann ein DECT-fähiges Headset verwendet werden. Hierzu wird die Basisstation mit der DECT-Buchse des vicBASE-3_5 verbunden.

Für den Reset oder Recovery Fall ist das Board mit 2 Buttons bestückt. Die Signale sind active-low und auch als Schraubklemme und Stiftleiste ausgeführt.

Es stehen 3 UARTs zur Verfügung, die an Schraubklemmen und an Stiftleisten ausgeführt sind. UART1 und UART2 sind ebenso als virtueller COM Port via Micro-USB ansteuerbar.

Insgesamt können mit dem vicBASE-3_5 bis zu 10 digitale Eingänge und 10 digitale Ausgänge bedient werden. Die IOs sind galvanisch von der restlichen Schaltung entkoppelt, wenn sie an den Schraubklemmen abgegriffen werden. Alle 20 IO-Signale sind ebenso auf einer Stiftleiste (ohne galvanische Trennung) ausgeführt. Neben der Bestückvariante 1 ist eine zweite Bestückvariante mit reduziertem Umfang auf je 6 Ein- und Ausgängen und ohne eine dritte UART verfügbar.

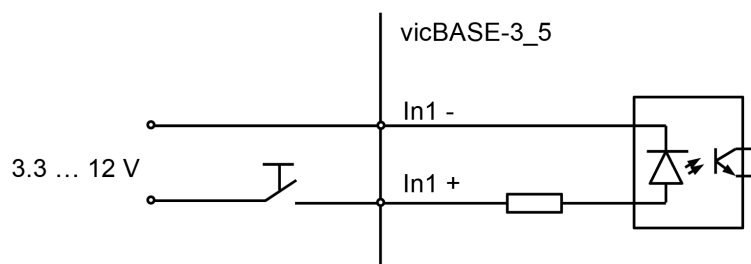


Abbildung 14: Beispiel Verwendung Digitaleingang

2.2.4 Abmessungen

Die Außenabmessungen betragen 145 x 75 mm. Die maximale Höhe befindet sich an der DECT-Buchse und beträgt 18 mm.

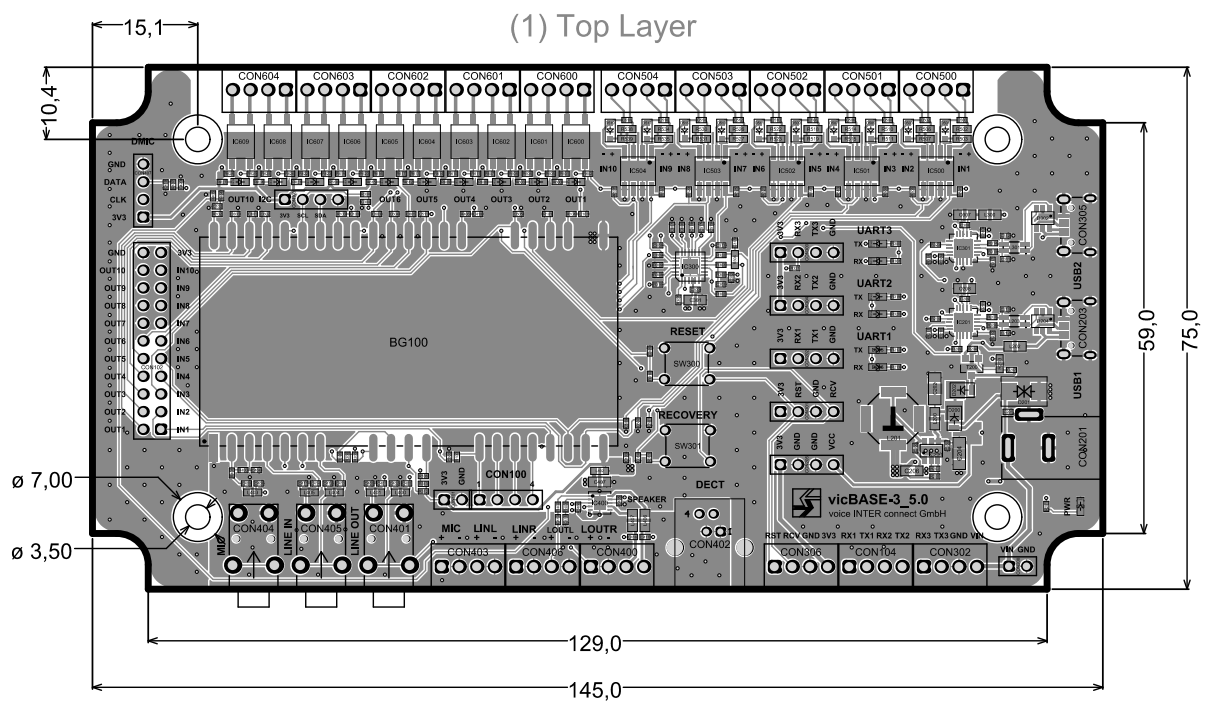


Abbildung 15: vicBASE-3_5 Draufsicht (Angaben in mm)

3 Firmware

Zur Konfiguration der Steuerung mittels Spracherkennung ist ein PC mit Windows 7 oder neuer notwendig.

3.1 Auslieferungszustand

Der in Abschnitt 5.3 vorgestellte Beispieldialog ist im Auslieferungszustand voreingestellt. In allen Audio-Prompt-Slots befindet sich der gleiche kurze Bestätigungston „p1.wav“ mit Ausnahme von Prompt 2, der einen abweichenden Bestätigungston „p2.wav“ enthält.

Aktuell werden 30 Sprachen unterstützt. Auf der Plattform kann jeweils eine Sprache gleichzeitig installiert sein.

Die installierte Sprache kann mit dem Update/Recovery Tool (siehe Abschnitt 3.3) geändert werden, indem die entsprechende vicCONTROL_LanguagePack_*.vcupd Sprachdatei aus der Lieferung hochgeladen wird.

3.2 Steuerung über das serielle Interface

3.2.1 Einstellungen der seriellen Schnittstelle

Das vicCORE-3_2 stellt zwei UARTs zur Verfügung. Auf dem vicBASE-3_5 sind beide Kommunikationsschnittstellen als virtuelle, serielle Verbindung per USB nutzbar. Die erste UART dient zur Kommunikation mit der Plattform, die zweite UART wird für Debug-Meldungen verwendet.

Die Einstellungen der seriellen Schnittstelle können mithilfe der Konfigurationssoftware vicCONTROL Designer (siehe Kapitel 5) geändert werden. Die folgenden Werte stellen den Standard dar.

- Datenrate: 115200 Baud
- Datenbits: 8 bit
- Parität: keine
- Stopbit: 1
- Handshaking: nein

3.2.2 Protokollbeschreibung

- Es können nur Punkt-zu-Punkt-Verbindungen hergestellt werden.
- Das Protokoll arbeitet mit Nachrichten.
- Es gibt unterschiedliche Arten von Nachrichten.
- Der Austausch mehrerer zusammengehörender Nachrichten wird als Vorgang bezeichnet.
- Wenn kein Vorgang aktiv ist, können beide Teilnehmer einen neuen Vorgang starten.
- Vorgänge die von der Plattform ausgehen, sind immer unidirektional. Das heißt, der Host muss niemals bestätigen oder antworten.

- Wenn es zu Beginn eines Vorgangs zu einer Kollision kommt, muss diese durch den Host erkannt und ggf. abgearbeitet werden.
- Als Algorithmus für die Checksummen kommt CRC-CCITT (CRC16) zum Einsatz
 - Modus = vorwärts
 - Startwert = 0xFFFF
 - Polynom = 0x1021
- Die Zeichen sind UTF-8 kodiert

3.2.3 Nachrichtenaufbau

Nachrichten sind charakterisiert durch folgende Merkmale:

- Länge zwischen 12 und 268 Byte
- Beginnen immer mit „v1“
- Das dritte Byte legt den Typ fest. Folgende Typen gibt es:
 - Befehl „C“
 - Antwort „A“
 - Meldung „M“
- Die folgenden 4 Zeichen bilden den Namen
- Es folgen 2 Zeichen, welche die Länge des Parameters als Hexdump angeben
- Der Datenbereich ist entsprechend lang.
- Nach dem Datenbereich folgen 2 Byte mit der Checksumme (CRC16 über alle vorherigen Zeichen, Ausgabe als Hexdump)
- Den Abschluss bildet ein Carriage return-Zeichen (CR)

Bytezahl	2	1	4	2	0 ...255	2	1
Bedeutung	Kopf	Typ	Name	Parameter- länge	Parameter	Check- summe	CR
Beispiel (Spracherkenner starten)	v1	C	Recg	00		C32C	„\r“

3.2.4 Liste der Befehle, Antworten und Meldungen

Befehle

Befehle können der Plattform erteilt werden.

Name	Bedeutung	Parameterlänge	Parameter	Bereich
Recg	Spracherkenner starten	0		
Stop	Spracherkenner anhalten	0		
GtSt	Aktuelles Menü abfragen	0		
SwSt	Zustand wechseln	1 ...255	Menüname (Text)	UTF-8
TrEv	Event Auslösen	1 ...10	Command Class-ID (dezimal)	1 ...500
PIPR	Prompt abspielen	1 ...10	Index des Prompts (dezimal)	1 ...24
GetV	Firmware Version abfragen	0		
GLng	Sprache abfragen	0		
GtPN	Plattform abfragen	0		

Beispiele (ANSI Darstellung):

Spracherkenner aktivieren

```
v1CRecg00C32C\r
```

Spracherkenner deaktivieren

```
v1CStop00A146\r
```

Aktuelles Menü abfragen

```
v1CGtSt000278\r
```

In das Menü „Untermenü“ wechseln

```
v1CSwSt0AUntermenü¼132D\r
```

Das Event 1 1 auslösen

```
v1CTrEv01110BAE\r
```

Prompt 1 1 abspielen

```
v1CPIPr0111BDB3\r
```

Version der Firmware abfragen

```
v1CGetV0022B6\r
```

Sprache abfragen

```
v1CGLng0003E7\r
```

Plattform abfragen

```
v1CGtPN009BC0\r
```

Antworten

Eine Antwort wird als Folge eines Befehls von der Plattform generiert.

Name	Bedeutung	Parameterlänge	Parameter	Bereich
Retn	Antwort	3	Fehlercode (dezimal) 000 = kein Fehler	
Vers	Version der Firmware	0 ...255	Text	
Lang	Sprache der Firmware	0 ...255	Text	
PfNm	Plattform	0 ...255	Text	

Beispiel:

Antwort: „Kein Fehler“ v1ARetn03**000**E9C4\r

Meldungen

Meldungen werden asynchron von der Plattform generiert, z.B. wenn eine Phrase erkannt wurde.

Name	Bedeutung	Parameterlänge	Parameter	Bereich
Volm	Eingangsspegel	4	Wert als Hexdump	0x0000 ...0x7FFF
RsLn	Ergebnis mit mehreren Hypothesen erkannt	2	Anzahl Hypothesen (dezimal)	0 ...20
RsHy	Hypothese	9 ...255	Confidence in „[]“ gefolgt von Triggerclass-ID(s) die durch „;“ getrennt und mit „;“ abgeschlossen werden	
RsTx	Text der erkannt wurde	0 ...255	Text	UTF-8
StCh	Zustandswechsel	0 ...255	Name des neuen Menüs (State)	UTF-8
Boot	Bootvorgang abgeschlossen	0		

Spracherkennungsergebnisse (RsLn, RsHy, RsTx) und Zustandsänderungen (StCh) werden nur ausgegeben, wenn im vicCONTROL Designer für einen Zustandsübergang (Transition) kein „Custom serial output“ konfiguriert wurde (siehe Kapitel 5).

Für jedes Spracherkennungsergebnis wird zuerst RsLn ausgegeben, das die Anzahl der Hypothesen enthält. Danach werden jeweils pro Hypothese RsHy und RsTx ausgegeben. RsHy enthält die Confidence, mit der das Ergebnis erkannt wurde, sowie die Triggerclass-ID(s). RsTx enthält den erkannten Text.

Beispiele:

Bootvorgang abgeschlossen

```
v1MBoot00F88F\r
```

Aktueller Eingangsspegel

```
v1MVolm04012703C0\r
```

Ergebnis mit einer Hypothesen

```
v1MRsLn02015630\r
```

Hypothese mit Konfidenz 04758 und der Triggerclass-ID 2

```
v1MRsHy09[04758]2;1F04\r
```

Ein Ergebnis Text

```
v1MRsTx07Heizung5161\r
```

Übergang in das Menü „Heizung“

```
v1MStCh07HeizungF45F\r
```


Zugriff mittels Terminal Emulator

Um die serielle Verbindung zu testen, können sie unter Windows zum Beispiel die Software PuTTY verwenden.

Konfigurieren Sie PuTTY wie unten abgebildet. Ersetzen Sie „COM10“ durch den Bezeichner, den die (virtuelle) serielle Schnittstelle trägt, an der die vicCONTROL Plattform angeschlossen ist.

Nach erfolgreicher Verbindung können sie zum Testen das Kommando für die Versionsabfrage eingeben („v1CGetV0022B6“) und dann die Eingabetaste drücken. Sie erhalten die aktuelle Version der Firmware als Rückgabe.

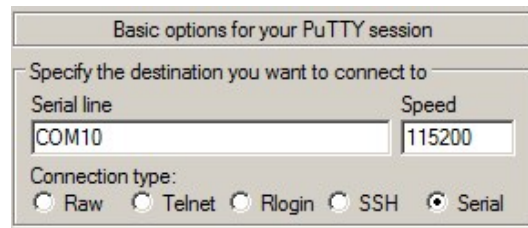


Abbildung 16: PuTTY Konfiguration 1

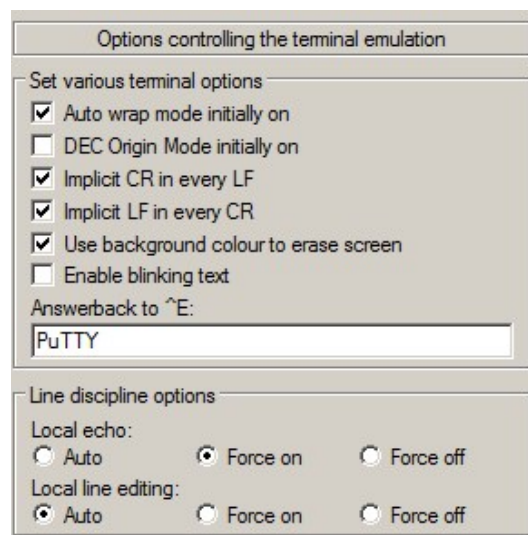


Abbildung 17: PuTTY Konfiguration 2

3.3 Update und Recovery

Für alle Firmware-Updates und als Wiederherstellungslösung nach missglückten Konfigurationsvorgängen mittels vicCONTROL Designer wird das Programm vicCONTROL_update2.2.0.exe zur Verfügung gestellt. Es ist auf dem USB-Stick zu finden.

Das Update-Programm benötigt abhängig von der Windows-Version ggf. Administratorrechte. Abbildung 18 zeigt die Programmoberfläche. Eine Installation der Update-Software ist nicht notwendig.

Firmware Updates werden normalerweise in zwei Varianten geliefert:

1. Firmware"Updates sind die vollen Updates inklusive der Software mit neuen Funktionen und Fehlerbehebungen.
2. LanguagePacks"sind Updates die schneller durchlaufen weil weniger Daten auf das Board übertragen werden müssen, um nur die Sprache zu ändern. Vor dem Einsatz der LanguagePacks muss sichergestellt sein, dass die Software auf dem Board aus der gleichen Version ist. Dies wird erreicht, indem zuerst ein volles Firmware"Update installiert wird bevor angefangen wird die LanguagePacks aus der gleichen Version zu verwenden.

Alle Updates egal ob Firmware"oder LanguagePack"setzen das Board auf die Werkseinstellungen zurück.

vicBASE-3_5

- Verbinden Sie das vicBASE-3_5 über USB1 mit Ihrem PC
- Starten Sie das Update-Programm
- Wählen Sie oben links den korrekten COM-Port (Mit „Rescan“ können Sie die Liste der verfügbaren COM-Ports aktualisieren)
- Mit Hilfe der Schaltfläche „Open Update File“ öffnen Sie eine .vcupd Update-Datei. Es erscheinen Informationen über das Firmware-Update und ggf. weitere Anweisungen.
- Auf dem vicBASE-3_5: Drücken Sie den RECOVERY Button und halten Sie ihn fest. Drücken Sie dann den RESET Button während Sie den RECOVERY Button immernoch festhalten. Halten Sie den RECOVERY Button für mindestens weitere 3 Sekunden Fest nachdem Sie den RESET Button losgelassen haben. Das board befindet sich jetzt im Update-Modus.
- Mit der Schaltfläche „Start“ im Update-Programm wird der Vorgang ausgelöst.

vicCORE-3_2

Sie benötigen eine funktionierende serielle Verbindung (COM) zwischen ihrem PC und dem vicCORE-3_2 (z.B. mittels MAX232)

- Der Eingang RECOV (X138) muss für das Update vor Inbetriebnahme des Boards auf Masse (GND) liegen.
- Starten Sie das Update-Programm
- Wählen Sie oben links den korrekten COM-Port (Mit „Rescan“ können Sie die Liste der verfügbaren COM-Ports aktualisieren)
- Mit Hilfe der Schaltfläche „Open Update File“ öffnen Sie eine .vcupd Update-Datei. Es erscheinen Informationen über das Firmware-Update und ggf. weitere Anweisungen.
- Mit der Schaltfläche „Start Update“ wird der Vorgang ausgelöst.

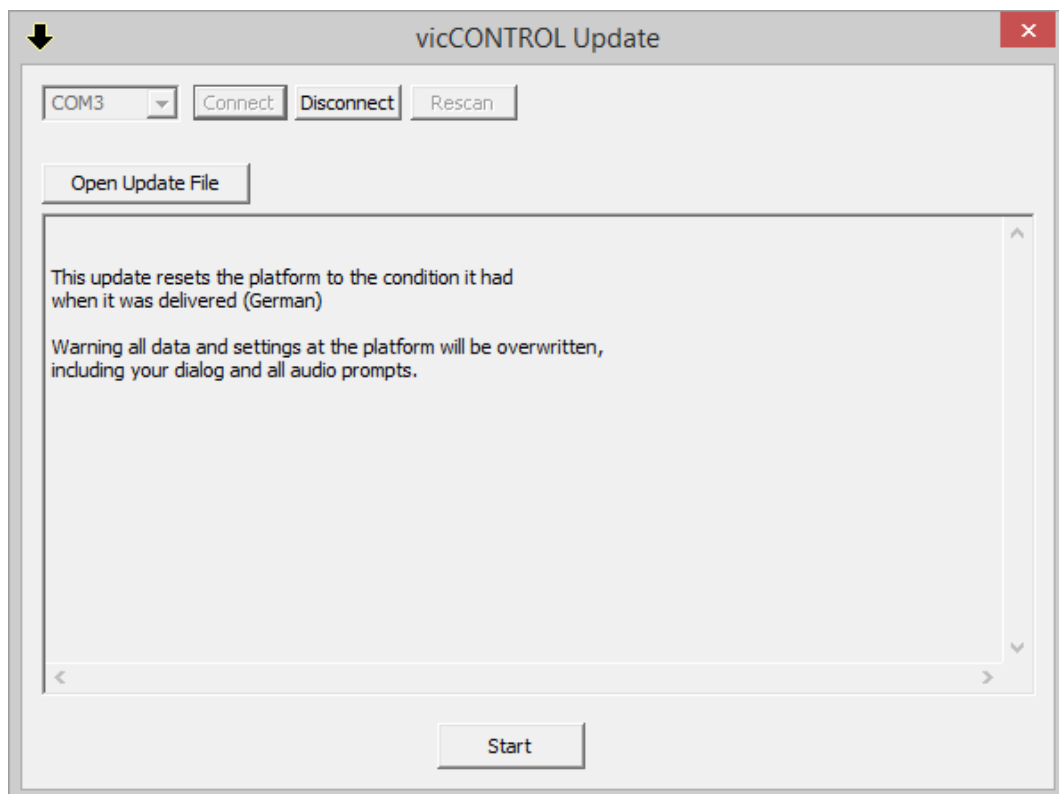


Abbildung 18: Screenshot – vicCONTROL_update2.2.0.exe

Ein unterbrochener Vorgang kann/muss erneut durchgeführt werden.

4 USB-Treiber

Für die Kommunikation mit *vicCONTROL go* über die USB-Schnittstelle des *vicBASE-3_5* ist ein Treiber erforderlich, der einen virtuellen COM-Port bereitstellt. Dieser sollte über ein Windows Update automatisch beim ersten Anstecken installiert werden. Den Treiber erhalten Sie mit der Lieferung von *vicCONTROL go* und über die Herstellerwebsite:

<https://www.ftdichip.com/Drivers/VCP.htm>

Zur Installation der Treiber sind Administratorrechte notwendig. Wenn Benutzer ohne Administratorrechte mit der USB-Schnittstelle arbeiten, wird empfohlen, das Board immer am gleichen Anschluss zu belassen.

5 vicCONTROL Designer

vicCONTROL Designer ist die Modellierungssoftware, die zur Einrichtung von vicCONTROL go genutzt wird. Die Konfiguration wird auch als Dialog bezeichnet. vicCONTROL Designer ist unter Windows 7 und 10 lauffähig und setzt eine Java Laufzeitumgebung voraus.

Eine mögliche Bezugsquelle für ein Java SDK ist das ojdkbuild Projekt (<https://github.com/ojdkbuild/ojdkbuild>).

Der Designer benötigt ebenfalls ein installiertes **x86** "Visual C++ Redistributable für Visual Studio 2012" Es wird zusammen mit dem Designer installiert.

5.1 Installation und Einrichtung

- Führen Sie das vicCONTROL Designer Setup-Programm aus
- Starten Sie vicCONTROL Designer mit Hilfe der Verknüpfung auf dem Desktop bzw. aus dem Startmenü.
- Öffnen Sie das Window-Menü und wählen Sie „Preferences“ aus. In diesem Dialog können Sie die Zielhardware und die Sprache für die der Dialog gebaut werden soll wählen.

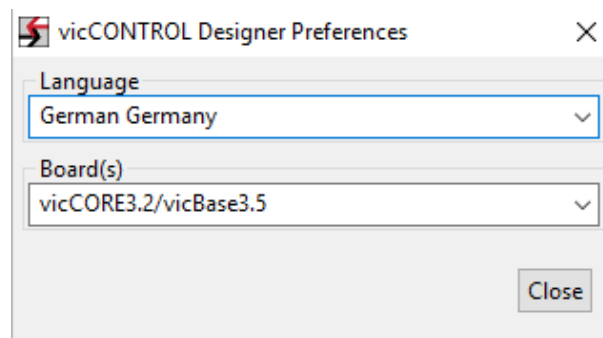


Abbildung 19: Preferences Dialog

5.2 Oberfläche

Die Oberfläche besteht aus einem Hauptfenster und mehreren Unterfenstern. Die Position und Größe der Unterfenster kann frei gewählt werden. Abbildung 20 zeigt eine Beispielkonfiguration – im Hauptfenster ist die Dialogdatei „Exam_German Germany“ geöffnet. Rechts oben befindet sich das Eigenschaftsfenster, mit dessen Hilfe man die Eigenschaften der Dialogelemente anpassen kann. Unten rechts wird eine vereinfachte grafische Vorschau auf den Dialog angezeigt. Dahinter verborgen befindet sich das Konsolenfenster, welches Status- und Fehlermeldungen anzeigt. Unten Links ist das „Platform-Fenster“ angeordnet, in dem die Interaktion mit der Hardwareplattform stattfindet. Über den Fenstern befinden sich die Menüleiste und die Befehlsleiste.

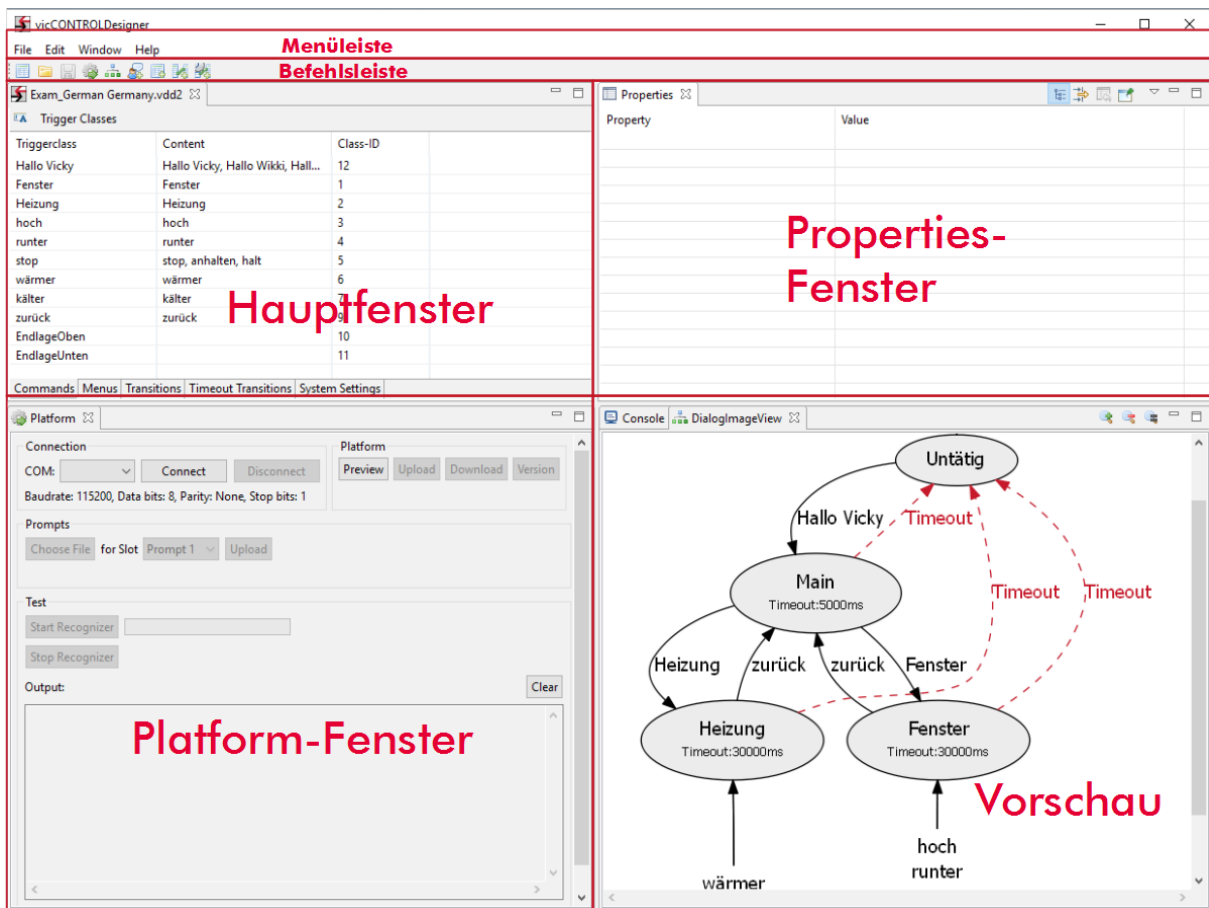


Abbildung 20: vicCONTROL Designer Oberfläche

- Das Hauptfenster ist unterteilt in die Tabs „Commands“, „Menus“, „Transitions“, „Timeout Transitions“ und „System Settings“.
- GUI-Elemente können durch halten der linken Maustaste innerhalb Ihrer Tabelle verschoben werden.
- Um ein Element aus einer der Tabellen zu löschen, klicken Sie mit der rechten Maustaste auf das entsprechende Element und wählen Sie im Kontextmenü „Delete“.
- Eine komplette Konfiguration der Sprachsteuerung wird als Dialog bezeichnet.
- Ein Dialog setzt sich aus folgenden Top-Level-Elementen, welche im Abschnitt 5.2.1 näher erläutert werden, zusammen:
 - „Triggerclasses“ (Erzeugung und Bearbeitung über den „Commands“ Tab)
 - „Menus“ (Erzeugung und Bearbeitung über den „Menus“ Tab)
 - „Transitions“ (Erzeugung und Bearbeitung über den „Transitions“ Tab)
 - „TimeOutTransitions“ sind spezielle Transitionen, die über den „Timeout Transitions“ Tab erzeugt und editiert werden können.
 - Im Tab „System Settings“ können grundlegende Eigenschaften des Dialogs verändert werden.
- GUI-Elemente können mittels Drag and Drop verschoben werden.

- Für jedes Element, das im Hauptfenster angeklickt wird, erscheinen die Eigenschaften im „Properties“ Fenster. Dort können sie auch verändert werden, wenn ihre Bearbeitung zulässig ist.
- Ein geschlossenes Unterfenster kann im Hauptmenü über „Window“ wieder geöffnet werden.

5.2.1 Dialogerstellung

Im Folgenden werden die fünf Tabs des Hauptfensters beschrieben:

Commands

Im Commands Tab ist es möglich, neue sog. Triggerklassen zu erzeugen oder bestehende Triggerklassen zu ändern. Triggerklassen haben die folgenden Eigenschaften, die alle im Hauptfenster erscheinen:

- Name
- Content/Inhalt (Die Sprachkommandos, welche die Klasse auslösen)
- ID (wird automatisch vergeben)

Menus

Ein Menü fasst eine Menge von Aktionen zusammen, die als Transitionen realisiert werden. So hat jedes Menü eine spezielle Anzahl von möglichen ausführbaren Aktionen, welche entweder durch Spracheingabe, das Betätigen eines Buttons oder Ereignisse an den Digitaleingängen ausgelöst werden. Im Menu Tab können neue Menüs angelegt oder bestehende verändert werden. Die folgenden Eigenschaften können für jedes Menü im Hauptfenster angepasst werden.

- Name
- Timeout:
Nach der festgelegten Zeitspanne in Millisekunden wird ein Timeout ausgelöst. Im Timeout Transition Tab (siehe unten) kann festgelegt werden, was dann passieren soll (z.B. Menüwechsel).
- Score High:
Dieser Wert gibt die Grenze an, ab der ein gesprochenes Kommando als sicher verstanden gilt und Aktionen ausgelöst werden.
- Score Low:
Wird dieser Wert unterschritten, dann gilt der Befehl als nicht erkannt. In dem Bereich zwischen Score High und Score Low gilt ein Befehl als unsicher. Im Tab „System Settings“ kann definiert werden, was bei unsicherer Spracherkennung passieren soll. Wenn „Score Low“ und „Score High“ den gleichen Wert haben, gibt es keine unsichere Region und die Unsicherheitsaktion (wie z.B. ein Prompt mit dem Inhalt „Wie Bitte?“) wird in diesem Menü unterdrückt.
- Use wakeup word:
Definiert ob dieses Menü durch das globale „wakeup word“ – das unter „System Settings“ konfiguriert wird – geschützt wird.

Im Properties-Fenster stehen folgende zusätzliche Einstellungen zur Verfügung:

- Garbage words:
Hier können Sprachkommandos eingetragen werden, die in diesem Menü auf keinen Fall erkannt werden sollen.

- **Is Special Menu:**
Diese Option schaltet mehr Parameter des Spracherkenners für die Veränderungen durch die Benutzer frei. In der Vorschau werden solche Menüs gelb angezeigt.
- **Wakeup timeout:**
Zeit nach der das „wakeup word“ für dieses Menü wiederholt werden muss.

Transitions

Im Transitions Tab können Übergänge (Transitionen) zwischen den einzelnen Menüs bearbeitet werden. Diese Transitionen enthalten alle nötigen Parameter, um die Aktionen zu beschreiben, welche durch den Übergang ausgelöst werden. Folgende Eigenschaften bearbeiten Sie im Hauptfenster:

- **Old Menu:**
Diese Eigenschaft bestimmt das Ursprungsmenü der Transition.
- **New Menu:**
Gibt das Menü an, in dem sich das Programm nach dem Übergang befinden wird.
- **Triggerclass:**
Die Triggerklasse ist der Auslöser für eine Transition. Sie wird im „Command Editor“ definiert.
- **Comment:**
Kommentare werden in der Vorschau angezeigt und vereinfachen die Unterscheidung von Transitionen.

Darüber hinaus stehen weitere Eigenschaften im Eigenschaftsfenster zur Verfügung. Welche Eigenschaften das genau sind, hängt davon ab, welche Hardware im Preferences-Dialog ausgewählt wurde (vgl. Kapitel 2 und 5.1)

- **Custom serial output:**
Ein hier eingetragener „Custom serial output“ wird bei Ausführung der Transition über die serielle Kommandoschnittstelle ausgegeben (d.h. Kontakte X110 und X111 des vicCORE-3 bzw. USB1 beim kit) Wenn kein Text angegeben wird, gibt das Board Standardmeldungen aus (siehe Abschnitt 3.2.4). Die üblichen aus C bekannten Escape-Sequenzen können verwendet werden. Zum Beispiel erzeugt „\n“ einen Zeilenumbruch.
Für nichtdruckbare Zeichen oder spezielle bytes kann eine Oktale Sequenz angegeben werden. Zum Beispiel „\177“ für „DEL“ (0x7F)
- **Inx Guard:**
Ein Guard ist eine Bedingung, die zusätzlich zur auslösenden Triggerclass erfüllt sein muss, damit eine Transition ausgeführt werden kann (z.B. In1 Guard = low bedeutet, dass der digitale Eingang In1 low sein muss).
- **Prompt:**
Gibt den Audio-Prompt an, der abgespielt wird, wenn die Transition ausgeführt wird.

- **Outx Action:**
Definiert das Verhalten der Relais bzw. Digitalausgänge während die Transition ausgeführt wird. Es gibt die Möglichkeiten „anschalten“ (SwitchOn), „ausschalten“ (SwitchOff) oder „im alten Zustand bleiben“ (DontCare).
- **Outx TO:**
Timeout für Ausgangsaktion in Millisekunden. Wenn ein Wert ungleich 0 eingetragen ist, schaltet das Relais/der Ausgang nach entsprechender Zeit automatisch ab.

Timeout Transitions

Wenn im „Menu Editor“ ein Timeout für ein Menü festgelegt wurde, kann hier das anschließende Verhalten definiert werden. Folgende Eigenschaften bearbeiten Sie im Hauptfenster:

- Old Menu:
Diese Eigenschaft bestimmt das Ursprungsmenü der Transition.
- New Menu:
Gibt das Menü an, in dem sich das Programm nach dem Übergang befinden wird.
- Comment:
Kommentare werden in der Vorschau angezeigt und vereinfachen die Unterscheidung von Transitionen.

Darüber hinaus stehen im Eigenschaftsfenster die gleichen Möglichkeiten wie bei „normalen“ Transitionen zur Verfügung.

System Settings

Die folgenden Systemeigenschaften können festgelegt werden:

- Audio settings
 - Digital input amplification:
Das Audioeingangssignal kann bei Bedarf digital verstärkt werden.
 - Enable Talk Through:
Wenn dieser Schalter auf „true“ gesetzt ist, wird das Audioeingangssignal, das der Erkenner bearbeitet, auf dem Audioausgangskanal wiedergegeben.
 - Out Level:
Hiermit kann die Audioausgabelautstärke angepasst werden. (z.B. wenn die .wav Dateien mit den Audioausgaben auf volle Lautstärke normalisiert sind)
- Digital inputs to event mapping
Dies ermöglicht die Zuordnung von Ereignissen an den Digitaleingängen zu Triggerklassen. Wenn zum Beispiel eine Triggerklasse festgelegt werden soll, die bei einer Flanke von High nach Low am Digitaleingang 3 ausgelöst wird, dann wird diese Klasse hinter In 3 H → L Event ausgewählt.
- Serial output configuration
Die Verbindungseinstellungen sollten **nur bei Bedarf** verändert werden.
Im Standardeinstellungen sind wie folgt:

Baudrate	115200
Data	Bits 8
Parity	None
Stop	Bits 1

Nachdem diese Einstellungen angepasst wurden, muss der Dialog an die Plattform übertragen werden. Dabei wird die Verbindung in dem Moment unterbrochen, in dem die neuen Werte aktiv werden.

Es ist daher notwendig, **auch im Platform-Fenster** die neuen Werte anzugeben (Doppelklick auf den Text mit den seriellen Einstellungen im Platform-Fenster) und vicCONTROL Designer **neu zu verbinden**. Falls die seriellen Einstellungen einer Plattform unbekannt sind, kann man sich mit dem Recovery-Programm behelfen (Kapitel 3.3).

Weiterhin ist es möglich, die vom Board gesendeten Meldungen anzupassen:

Signal Menu Changes	Meldung bei Menüwechsel
Signal Trigger Ids	Die ID eines Spracherkennungsergebnisses wird gemeldet
Signal Trigger Text	Der erkannte Text wird gemeldet
Signal Volume	Lautstärke melden oder/nicht

- Wakup
Konfiguriert Spracherkennungsparameter der global verfügbaren „wake up phrases“ (inklusive der Wörter die sie ausmachen)

- Action on wakeup
Legt global fest, was passieren soll, wenn die „wake up phrase“, die ein Menü schützt, erkannt wurde (Audio-Prompt, Ausgangs-/Relaisaktionen, Benutzerdefinierte Meldungen an der seriellen Schnittstelle). Sie wird wie eine Transition im Properties-Fenster konfiguriert.
- Action on sleep
Legt global fest, was passieren soll, wenn das „wakeup timeout“ abgelaufen ist (Audio-Prompt, Ausgangs-/Relaisaktionen, Benutzerdefinierte Meldungen an der seriellen Schnittstelle). Sie wird wie eine Transition im Properties-Fenster konfiguriert.
- Action if speech recognition between high and low score
Legt global fest, was passieren soll, wenn die Spracherkennung zu unsicher war (Audio-Prompt, Ausgangs-/Relaisaktionen, Benutzerdefinierte Meldungen an der seriellen Schnittstelle). Sie wird wie eine Transition im Properties-Fenster konfiguriert.

5.2.2 Das Platform-Fenster

Mit Hilfe des Platform-Fensters werden Board-spezifische Aufgaben durchgeführt. Nachdem eine Verbindung aufgebaut ist, können Dialoge vorbereitet und übertragen werden.

Wenn man Audio-Prompts verwenden möchte, können diese ebenfalls übertragen werden. Jeder der bis zu 8 Audio-Prompts darf maximal 4 Sekunden lang sein und muss im RIFF-WAV-Dateiformat (16 Bit, 16 kHz, Mono) mit 44 Byte Standard-Header vorliegen.

Wenn die Verbindung hergestellt ist und ein funktionsfähiger Dialog übertragen wurde, kann der Spracherkenner getestet werden.

- Start Recognizer: Spracherkenner starten
- Stop Recognizer: Spracherkenner deaktivieren

Im unteren Teil des Fensters werden die Rückgaben des Boards dargestellt. Wenn eigene Ausgaben (Custom serial output) über die serielle Schnittstelle definiert wurden, werden diese nur korrekt dargestellt, wenn sie lediglich druckbare ASCII Zeichen enthalten und mit einem CR-Zeichen („\r“) enden.

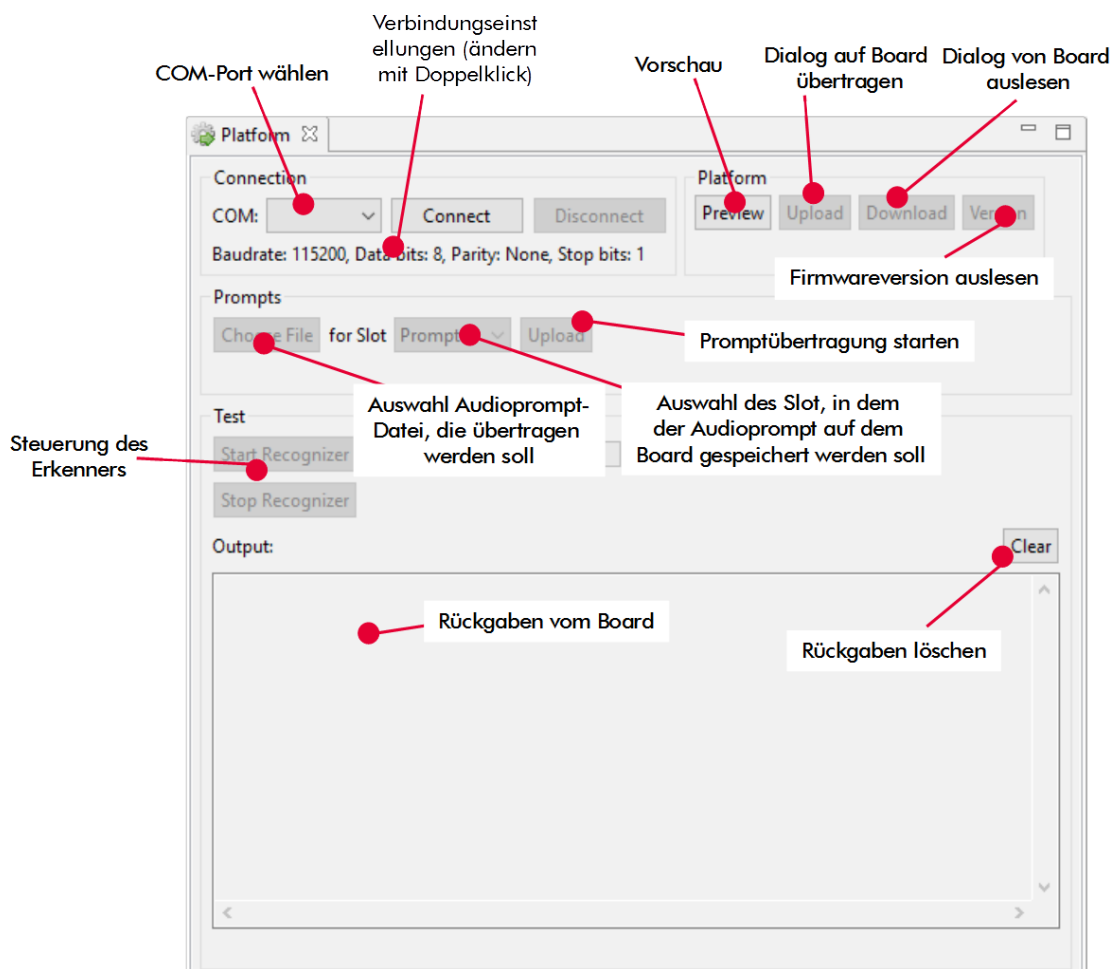


Abbildung 21: „Platform“-fenster

5.3 Aufbau eines Beispieldialogs

ACHTUNG:

Die Endlagenschalter in diesem Beispiel dienen zur Veranschaulichung der Benutzung von Digital-eingängen. vicCONTROL go sollte niemals allein für sicherheitsrelevante Funktionen verwendet werden!

Dieses Beispiel zeigt, wie ein Dialog für eine Haussteuerung (Rollläden und Heizung) mit Hilfe von vicCONTROL Designer realisiert werden kann. Die entsprechende Projektdatei befindet sich auf dem USB Stick.

Der Rollladen soll nach oben und unten gefahren werden können. Die Bewegung soll jederzeit unterbrechbar sein. Außerdem sind am Rollladen zwei Endlagenschalter angebracht, die auslösen, sobald er ganz oben bzw. ganz unten ist. Das Beispiel geht davon aus, dass der Motor des Rollladens so lange nach oben fährt, wie das Relais 1 am vicCONTROL eingeschaltet ist und nicht der obere Endlagenschalter geschlossen ist. Solange Relais 2 aktiv ist, und nicht der untere Endlagenschalter ausgelöst wurde, fährt der Rollladen nach unten.

Für die Heizung wird davon ausgegangen, dass diese durch einen 500 ms langen Impuls am Relais 3 eine Stufe wärmer bzw. durch einen 500 ms Impuls am Relais 4 eine Stufe kälter gestellt wird.

Abbildung 22 zeigt eine schematische Darstellung des Aufbaus.

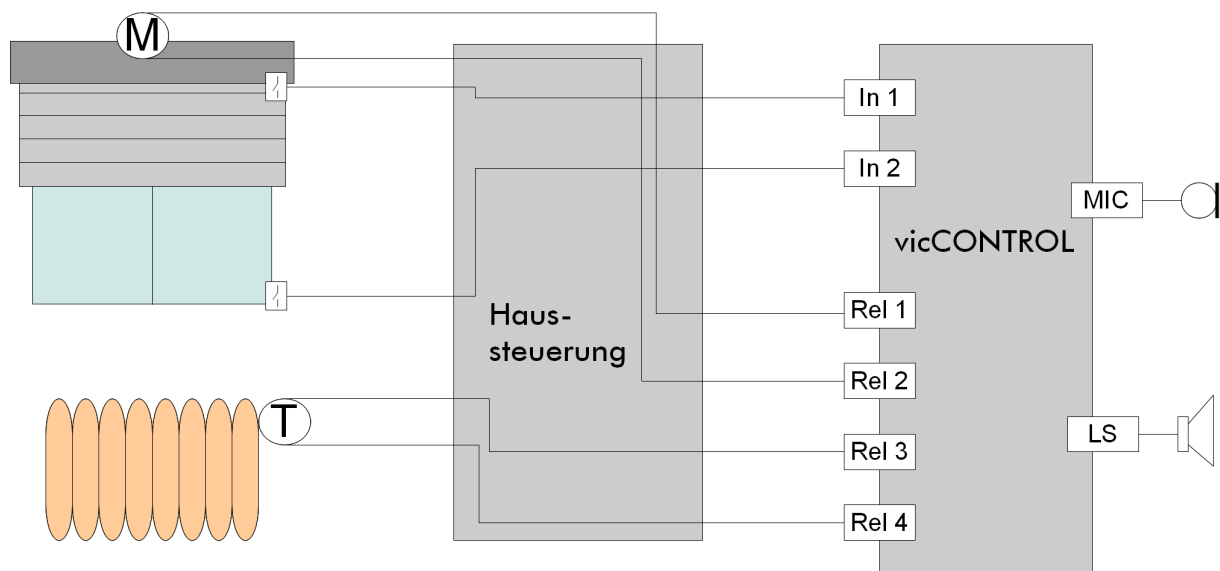





Abbildung 22: Beispielanwendung


Ablauf:

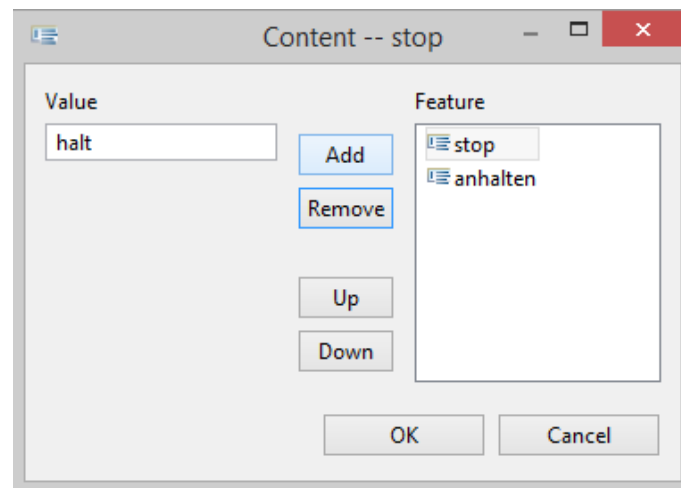
- Zuerst wird ein neuer Dialog erstellt: File → New oder 
- Menüs definieren
 - Wechsel zum Menu Tab im Hauptfenster
 - Es werden vier Menüs benötigt. Den Button „Create Menu“  vier mal betätigen!
 - Die Namen in Idle, Main, Fenster und Heizung ändern

- Das Idle Menü wird die Aktivierungsphrase „Hallo Vicky“ enthalten. Da in diesem Menü keine Rückmeldung bei unsicherer Erkennung gewünscht ist, erhalten beide Scores den gleichen Wert. „Hallo“ wird als Garbage Word hinzugefügt, das der Erkenner zwar auf „Hallo Vicky“, aber nicht auf ein einfaches (ähnlich klingendes) „Hallo“ reagieren soll.
- Das Main Menü erhält ein Timeout von 5 Sekunden.
- Die Menüs Fenster und Heizung erhalten ein Timeout von jeweils 30 Sekunden

Menus					Property	Value
Menu	Timeout (ms)	Score High	Score Low	Use wakeup word	Garbage Words	Hallo, Wikki, Hallo <...>, <...> Wikki, a
Untätig	0	4200	4200	false	Is Special Menu	false
Main	5000	4000	4000	false	Use Wakeup Word	false
Fenster	30000	4000	4000	false	Wakeup Timeout	20000
Heizung	30000	4000	4000	false		

- Triggerklassen (Auslöser-Klassen) für Transitionen erstellen
 - Zum Commands Tab im Hauptfenster wechseln
 - Den Button „Create Triggerclass“  betätigen, um eine neue Triggerklasse zu erstellen. Die Class-ID wird automatisch verteilt.
 - Folgende Klassen werden für Sprachbefehle erstellt:
 - * Hallo Vicky
 - * Fenster
 - * Heizung
 - * hoch
 - * runter
 - * stopp
 - * wärmer
 - * kälter
 - * zurück


Wenn auf das „Content“-Feld einer Klasse geklickt wird, erscheint ein  Button. Er führt zu einem Dialog, mit dessen Hilfe die Sprachbefehle definiert werden, die zu einer Klasse gehören.



- Im Feld unter „Value“ werden neue Sprachkommandos angelegt und dann mittels Add/Remove zur Klasse hinzugefügt/entfernt.
- Folgende Klassen werden für die Endlagenschalter erstellt. Sie erhalten keinen (Sprach)-Content.
 - * 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

- Transitionen (Zustandsübergänge definieren)
 - Zum „Transition Editor“ wechseln
 - Mit Hilfe des  Buttons wird eine neue Transition hinzugefügt.
 - Zuerst werden die Transitionen für die Menüübergänge erstellt

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


- Im Heizungsmenü wird eine Transition zum Wärmerstellen benötigt. Wenn also die Triggerklasse „wärmer“ (enthält Sprachkommando wärmer) ausgelöst wird, dann bleibt die Steuerung im gleichen Menü. In den Properties der Transition wird ausgewählt, dass Relais 3 für 500 ms eingeschaltet werden soll.
- Eine ähnliche Transition wird für „kälter“ benötigt. In ihren Properties wird jedoch eingestellt, dass Relais 4 für 500 ms eingeschaltet werden soll.

Transitions				Property	Value
Old Menu	Triggerclass	New Menu	Comment		
Idle	Hallo Vicky	Main		Custom serial output	none
Main	Fenster	Fenster		In1 Guard	none
Fenster	zurück	Main		In2 Guard	none
Main	Heizung	Heizung		In3 Guard	none
Heizung	zurück	Main		In4 Guard	none
Heizung	wärmer	Heizung		PB1 Guard	none
Heizung	kälter	Heizung		PB2 Guard	none
Fenster	hoch	Fenster		Prompt	Prompt01
Fenster	runter	Fenster		Rel1 Action	DontCare
Fenster	stop	Fenster		Rel1 Timeout	0
Fenster	EndlageOben	Fenster		Rel2 Action	DontCare
Fenster	EndlageUnten	Fenster		Rel2 Timeout	0
				Rel3 Action	SwitchOn
				Rel3 Timeout	500
				Rel4 Action	DontCare

- Im „Fenster“ Menü wird eine Transition benötigt, um den Rollladen nach oben fahren zu lassen.
 - * Wenn diese auftritt, wird Relais 1 eingeschaltet. (Rel1Action = SwitchOn)
 - * Relais 2 wird ausgeschaltet (Rel2 Action = SwitchOff)
 - * Außerdem wird sie nur dann ausgeführt, wenn der obere Endlagenschalter nicht aktiv ist. (In1 Guard = low)
- Es wird eine zusätzliche Transition für das Herunterfahren benötigt. Sie erhält die folgenden Eigenschaften:
 - * Rel1Action = SwitchOff
 - * Rel2Action = SwitchOn
 - * In2 Guard = low
- Eine weitere Transition erlaubt es, dem Benutzer den Rollladen anzuhalten.
 - * Rel1Action = SwitchOff

- * Rel2Action = SwitchOff
- Eine Transition wird benötigt, um den Rollladen anzuhalten, wenn er die obere Endlage erreicht.
 - * Rel1Action = SwitchOff
- Eine weitere Transition für die untere Endlage.
 - * Rel2Action = SwitchOff
- Zum Abschluss soll sichergestellt werden, dass der Rollladen anhält, wenn das Fenstermenü verlassen wird. Entsprechend wird die bereits bestehende Transition von „Fenster“ nach „Main“ um folgende Eigenschaften ergänzt:
 - * Rel1Action = SwitchOff
 - * Rel2Action = SwitchOff
- In allen Transitionen wird Prompt 1 als Bestätigungston gewählt.

Transitions				Property	Value
Old Menu	Triggerclass	New Menu	Comment	Custom serial output	
Idle	Hallo Vicky	Main		In1 Guard	low
Main	Fenster	Fenster		In2 Guard	none
Fenster	zurück	Main		In3 Guard	none
Main	Heizung	Heizung		In4 Guard	none
Heizung	zurück	Main		PB1 Guard	none
Heizung	wärmer	Heizung		PB2 Guard	none
Heizung	kälter	Heizung		Prompt	Prompt01
Fenster	hoch	Fenster		Rel1 Action	SwitchOn
Fenster	runter	Fenster		Rel1 Timeout	0
Fenster	stop	Fenster		Rel2 Action	SwitchOff
Fenster	EndlageOben	Fenster		Rel2 Timeout	0
Fenster	EndlageUnten	Fenster		Rel3 Action	DontCare
				Rel3 Timeout	0
				Rel4 Action	DontCare


- Timeout Transitionen – da die Zustände Main, Fenster, und Heizung ein Timeout besitzen, muss definiert werden, was nach Ablauf der Zeit passiert; dazu folgende Vorgehensweise:
 - In den Timeout Transitions Tab wechseln
 - Drei Timeout Transitionen anlegen, indem 3 mal  betätigt wird.
 - Als Ziel wird jeweils der Startzustand „Idle“ gewählt.
 - Prompt 2 wird bei Auftreten aller Timeout Transitionen abgespielt

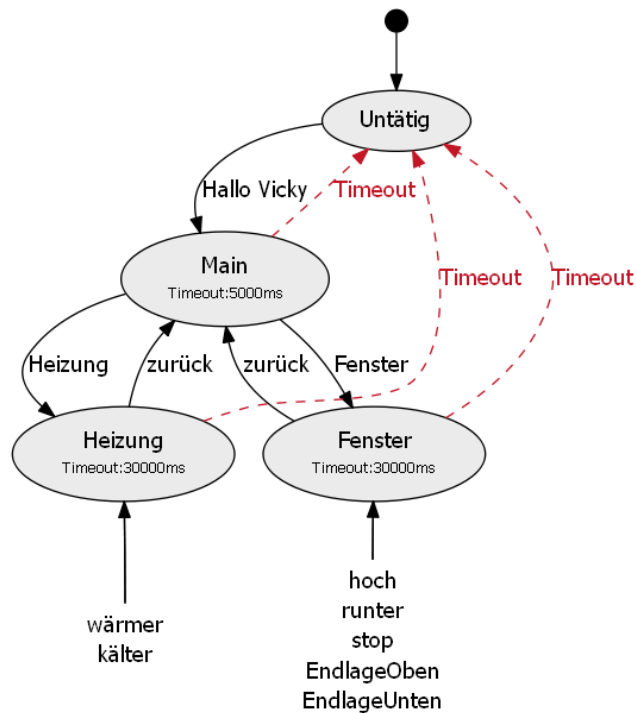
Timeout Transitions		
Old Menu	New Menu	Comment
Main	Idle	
Fenster	Idle	
Heizung	Idle	
Commands		Menus
Transitions		Timeout Transitions
System Settings		

- Systemeigenschaften anpassen

- In den „System Settings“ Tab wechseln und den Eintrag „Digital inputs to event mapping“ auswählen.
- Die Endlagenschalter sind mit den Eingängen 1 und 2 verknüpft. Wenn eine Signalflanke von Low nach High auftritt, soll eine Endlagen Transition ausgelöst werden. Hier werden die physikalischen Ereignisse am Eingang einer Triggerklasse zugeordnet. (E1 L→ H Event steht für Eingang 1 Low nach High)

System Settings	Property	Value
Audio settings	In1 Hto LEvent	┌
Digital inputs to event mapping	In1 Lto HEvent	EndlageOben
Serial output configuration	In2 Hto LEvent	┌
Speech recognizer settings	In2 Lto HEvent	EndlageUnten
Action on wakeup	In3 Hto LEvent	┌
Action on sleep	In3 Lto HEvent	┌
Action if speech recognition confidence between high and low score	In4 Hto LEvent	┌
	In4 Lto HEvent	┌

- Der Dialog ist vollständig modelliert und kann nun getestet werden.
- Zunächst mit Hilfe des Buttons **Preview** im „Platform“-Fenster oder  die Vorschau erstellen.



- Zum „Platform“-Fenster wechseln
- Es wird davon ausgegangen, dass das Board bereit für die Programmierung ist, d.h.:
 - * Spannungsversorgung ist eingeschaltet
 - * USB oder RS232 ist erfolgreich mit PC verbunden
- Den richtigen COM-Port auswählen und den Button „Connect“ betätigen.
- Mit Hilfe des **Upload** Button wird der Dialog auf die Plattform übertragen.
- Wenn das Hochladen erfolgreich abgeschlossen ist, kann der Spracherkennungsmittel **Start Recognizer** eingeschaltet werden.

Das Board ist programmiert und in Betrieb.

6 Dialoggestaltung

6.1 Allgemeines

vicCONTROL go bietet einen sehr flexibel konfigurierbaren und leistungsfähigen Spracherkennung, der es erlaubt, eine robuste Sprachsteuerung aufzubauen. Je nach Art des umzusetzenden Dialogs, sollte jedoch immer Zeit zur Optimierung dieses Dialogs angesetzt werden.

Eine 100 % korrekte Spracherkennung ist mit der vorliegenden Technologie nicht zu garantieren. Aber durch eine geeignete Dialoggestaltung und Konfiguration kann man sehr geringe Fehlerraten erreichen.

Die Dialoggröße ist derzeit auf 500 Kommandos, 200 Menüs und 500 Transitionen beschränkt, um den Speicherverbrauch auf der Embedded-Plattform zu begrenzen.

6.2 Aktivierung

Besondere Anforderungen an die Technologie stellt die kontinuierliche Spracherkennung. Das heißt, dass der Spracherkennung die ganze Zeit „zuhört“ und nicht durch z.B. einen Knopf (Push-to-Talk) oder einen anderen Sensor (z.B. Bewegungsmelder) aktiviert wird. In diesem Anwendungsfall ist es sinnvoll eine Aktivierungsphrase (bzw. Wakeup-Phrase oder Wakeup-Word) wie „Hallo Vicky“ zu verwenden und erst im zweiten Schritt die eigentlichen Kommandos zu erkennen. Für die Aktivierungsphrase gelten die folgenden allgemeinen Regeln:

- **Nicht zu kurz**
- Je weniger Aussprachevarianten existieren, desto besser.
- Möglichst wenige Zischlaute („ch“ „ts“, „sch“, „ft“, ...) verwenden

Im Startzustand sollten auch Garbage-Phrasen enthalten sein, also zum Beispiel die jeweils einzelnen Phrasen „Hallo“ und „Vicky“.

Wenn mehrere Sprachkommandos gleichzeitig aktiv sind, sollten sie sich möglichst stark unterscheiden, z.B.: „Ruf aufbauen“ und „Ruf beenden“ statt „Ruf aufbauen“ und „Ruf abbauen“

6.3 Feedback

Es ist wichtig dem Nutzer Feedback zu geben. Hierfür können z.B. Audio-Prompts oder optische Signale verwendet werden. Bei mehrstufigen Dialogen sollte der Benutzer auf jeden Fall über einen Menüwechsel informiert werden. Im Beispiel wird sogar bei jedem erkannten Kommando ein kurzer Signalton ausgegeben. Eine akustische Rückfrage „Wie Bitte?“ bei unsicherer Erkennung macht in Menüs mit „echten“ Kommandos Sinn, aber nicht im Startmenü.

Der Spracherkennung ist während Töne abgespielt werden nicht aktiv, „hört“ also nicht zu. Das gilt z.B. auch wenn am Ende einer Tondatei noch Stille vorhanden ist, die „abgespielt“ wird.

Wenn es auf Geschwindigkeit ankommt und auf ein Kommando ohnehin eine direkt wahrnehmbare Systemreaktion erfolgt, dann empfehlen wir auf zusätzliche Bestätigungstöne zu verzichten, damit der Erkenner schneller wieder für das nächste Kommando bereit ist.

6.4 One-Shot-Aktivierung

One-Shot-Aktivierung beschreibt hier die Möglichkeit die Aktivierungsphrase (bzw. Wakeup-Phrase oder Wakeup-Wort) und das eigentliche Kommando in einem Zug also ohne Pause zu sagen. Daraus ergibt sich ein Komfortgewinn und eine Steigerung der Nutzungsgeschwindigkeit. Es gilt jedoch auch Fallstricke zu vermeiden und die Bedienmethode ist nicht für alle Anwendungen optimal.

Als Beispiel wollen wir eine „Maschine“ „einschalten“ bzw. „ausschalten“. Das heißt, es soll die Aktivierungsphrase „Maschine“ mit den beiden möglichen Kommandos „ein(schalten)“ und „aus(schalten)“ kombiniert werden.

Die Kommandos werden wie üblich im vicCONTROL Designer unter „Commands“ angelegt. In diesem Beispiel sind es „einschalten“, „ausschalten“ und ihre typischen Synonyme.

Die Festlegung der Aktivierungsphrase findet in den System Settings unter Wakeup statt. Dort wird das Wort „Maschine“ definiert. One-Shot-Aktivierungsphrasen sind generell etwas optimierungsbedürftig. Das System versucht abzuschätzen, wie gut ein Wort funktionieren kann (in diesem Beispiel erhalten Sie beim Upload der Anwendung auf das Board einen Hinweis, dass die Phrase „nicht optimal“ ist. Zum Beispiel eine etwas längere Phrase wäre besser. Letztlich kann die Qualität aber nur an der konkreten Sprachanwendung und ihrer Umgebung getestet und überprüft werden.

Weil es teilweise bevorzugt wird, zuerst auf eine Reaktion nach der Aktivierungsphrase zu warten, obwohl eigentlich direkt weitergesprochen werden könnte, steht diese Möglichkeit grundsätzlich immer zur Verfügung.

In den System Settings unter Wakeup und Wakeup Speech Timeout wird definiert, dass das System z.B. bis zu 10 Sekunden auf ein Kommando wie („einschalten“ / „ausschalten“) wartet, wenn nur die Aktivierungsphrase gesagt wurde.

In den System Settings unter Action on wakeup bzw. Action on sleep wird konfiguriert, wie das System bei Aktivierung bzw. am Ende der aktiven Phase reagieren soll.

Die Reaktionen auf die normalen Kommandos werden wie immer über Menüs und (Timeout) Transitionen konfiguriert. Die One-Shot-Aktivierung kann dabei pro Menü über das Flag Use Wakeup Word in den Properties des jeweiligen Menüs aktiviert werden.

Auf dem USB-Stick bzw. im Download-Bereich für vicCONTROL go Kunden finden Sie eine Beispielprojektdatei, die auch komplexere Konzepte rund um Aktivierungsphrasen zeigt:

In ihr wird mittels Output 1 angezeigt, ob die Aktivierung aktiv ist und das System somit auf ein Kommando wartet, und es wird jeweils ein Audiodownload abgespielt bei Aktivierung / Deaktivierung.

Das „Reset“ Menü in der Beispieldatei stellt sicher, dass die Aktivierungsphrase nach jedem Kommando wieder neu gesagt werden muss. Sonst wäre es möglich bis zum Ablauf der vollen 10 Sekunden (vgl. System Settings -> Wakeup -> Wakeup Speech Timeout) immer weitere Kommandos auszulösen, ohne das Wakeup-Word nochmals neu zu nennen.

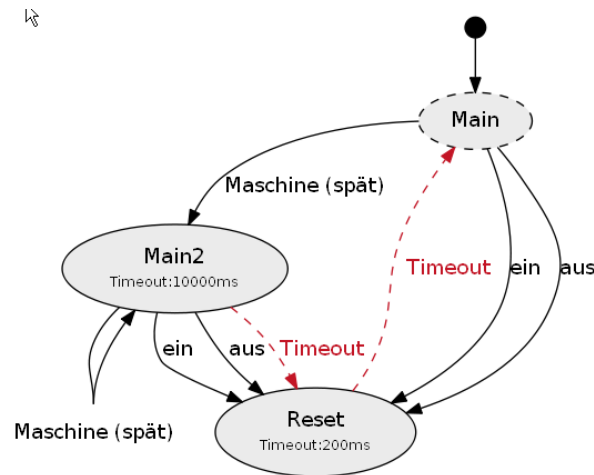


Abbildung 23: Oneshot Beispielprojekt

Ein möglicher Sonderfall (kann bei langem Wakeup Speech Timeout auftreten) ist, dass Nutzer, durch Ablenkungen vergessen können, dass sie die Aktivierungsphrase bereits gesagt haben. Dann wiederholen sie das Wakeup-Wort und erwarten, dass wieder die typische Reaktion des Systems sowie die vollen 10 Sekunden Timeout zur Verfügung stehen, obwohl das System ja eigentlich bereits einige Sekunden wach ist.

Dieser „Problemfall“ wird im Beispiel mit Hilfe des Menüs „Main2“ und des Kommandos „Maschine (spät)“ behandelt.

Bei Betrachtung der Synonyme in der Beispieldatei wird außerdem auffallen, dass alle Varianten jeweils auch mit der Aktivierungsphrase als Präfix hinterlegt sind. Das liegt daran, dass es erfahrungsgemäß Nutzer gibt, die lieber die Reaktion auf die Aktivierungsphrase abwarten, bevor sie das eigentliche Kommando sagen. Wenn diese Nutzer dann abgelenkt werden, dann können sie wie bereits beschrieben etwas länger zögern (zum Beispiel 5 Sekunden), und es kommt recht häufig dazu, dass nochmals die Aktivierungsphrase inklusive Kommando gesagt wird.

Die Beispieldatei zeigt bewusst einen recht komplexen Kompromiss, der versucht vielen Nutzungsszenarien gerecht zu werden. Grundsätzlich können einfachere Abläufe oder auch noch mehr Sonderfälle modelliert werden, je nachdem was Ihre Anwendung und Umgebung erfordert.

6.5 Umgang mit beliebiger ungewollter Sprache (Garbage), begrenzte Schlüsselwortsuche

Dieser Abschnitt beschreibt fortgeschrittene Strategien und Möglichkeiten, die Erkennungsleistung in besonderen Fällen und Umgebungen zu optimieren.

Das generelle Design von *vicCONTROL go* zielt darauf ab, schnelle und zuverlässigen Sprachsteuerungssysteme für wohldefinierte Befehlssätze, die von kooperativen Nutzern gesprochen werden, zu erstellen.

Die hier beschriebenen Werkzeuge können die Robustheit Ihrer Sprachanwendung erhöhen und sogar Nutzungen jenseits vom ursprünglichen Designziel des Systems erlauben, aber es gibt keine Garantien oder automatische Prüfungen gegen eine Fehlbenutzung dieser Möglichkeiten, denn es kommt jeweils sehr auf das Projekt, die Anwendung und die Situation an, ob die erreichbaren Effekte gewünscht oder unerwünscht sind.

Explizites “Müll“-Model

Manchmal kommt es vor, dass die Nutzer ähnlich klingende Wörter äußern, von denen aber nur eines als Kommando erfasst werden soll.

Ein Beispiel im Deutschen wäre: “Licht einschalten!” und “Nicht einschalten”.

Wenn im Umfeld einer Sprachanwendung, die das Kommando “Licht einschalten” enthält oft die Aufforderung “Nicht einschalten” geäußert wird, dann wäre es wünschenswert, dass der Spracherkenner die potentielle Verwechslung kennt und damit umgehen kann.

Eine einfache Methode dies zu modellieren, ist die ungewollte Phrase als Kommando zu hinterlegen, dass ignoriert wird, wenn es ausgelöst wird. Dies lässt sich bequem erreichen indem im vicCONTROL Designer in den Eigenschaften jedes Menüs, unter “Garbage Words”, die Wörter hinzugefügt werden, die in dem jeweiligen Menü ignoriert werden sollen.

Es liegt in der Verantwortung des Designers der Sprachanwendung dafür zu sorgen, dass es nicht zu Überlappungen oder Konflikten mit den anderen Kommandos des Menüs, die über die ausgehenden Transitionen definiert sind, kommt.

Ignorieren “beliebiger Sprache” die nicht anderweitig definiert ist

Es steht eine besondere Anweisung für die Definition eines allgemeinen Müllmodells zur Verfügung. Sie bedeutet: “Beliebige andere Sprache, die ignoriert werden soll”.

Im Kontext einer gesprochenen Phrase enthält dieses allgemeinen Müllmodell auch Stille (keine Sprache).

Das allgemeine Müllmodell wird mit der Zeichenkette “<...>” definiert.

Das allgemeine Müllmodell kann in der Definition des Inhalts von Kommandos oder den “Garbage Words” der Menüs benutzt werden.

Die Nutzung des allgemeinen Müllmodells reizt die Grenzen der Spracherkennung und Hardware von vicCONTROL go aus. Daher sollte es sparsam und nur in Begleitung ausführlicher Tests verwendet werden.

Die Freiheit eine Anweisung mit so allgemeiner Bedeutung zu verwenden, enthält die Pflicht ein besonderes Augenmerk auf potentielle Überschneidungen und Konflikte mit anderen Bereichen des Projektes zu legen.

Um eine gute Leistung zu erhalten, sollte die Nutzung auf den Anfang oder das Ende von Phrasen beschränkt werden.

- “<...> Wort”: Höhere Wahrscheinlichkeit gewünschtes Ziel zu erreichen
- “Wort <...>”: Höhere Wahrscheinlichkeit gewünschtes Ziel zu erreichen
- “Wort <...> Wort”: Höhere Wahrscheinlichkeit von Leistungsverlusten

Die Stärke dieser Möglichkeit kann pro Menü in den “Advanced Settings” mit dem Parameter “Recg Garbage” beeinflusst werden.

6.6 Erweiterte Spracherkennungseinstellungen

vicCONTROL go ist darauf optimiert für einen weiten Bereich möglicher Sprachanwendungen zu funktionieren, ohne dass es notwendig ist, erweiterte Parameter auf niedriger Systemebene anzupassen.

Die Freiheit erweiterte Parameter zu beeinflussen erlaubt es Lösungen für spezielle Szenarien zu finden. Unpassende Werte könne die Gesamtleistung des Systems verschlechtern.

Alle erweiterten Parameter stehen pro Menü zur Verfügung wenn "Is Special Menu" auf "true" gestellt ist und wenn "Recg Use Automatic" auf "false" gestellt ist.

- **Extra Event Enable** aktiviert die Erkennung besonderer akustischer Ereignisse und sollte üblicherweise aktiv sein
- **Fx Sensitivity** und **Fx VAD Threshold** kontrollieren die Erkennung von Sprachephrasen als solche
- **Fx Speech Timeout** bestimmen wieviel Zeit (in Millisekunden) vergehen darf, bevor die Spracherkennung abbricht, nachdem der Beginn von Sprache erkannt wurde. Da dieser Wert direkt den Hardware-Ressourcen-Verbrauch beeinflusst, kann er direkt zu Performance-Problemen führen, wenn er zu hoch ist.
- **Fx Trailing Silence** und **Recg Trailing Silence** beeinflussen beide die Stille (Sprachpause) in Millisekunden die für den Erkennen notwendig ist um das Ende einer Phrase zu erkennen. Es sind zwei Parameter, weil die Bewertung in unterschiedlichen Abschnitten der Verarbeitungskette stattfindet. Als Daumenregel sollte "Recg Trailing Silence" doppelt so lang wie "Fx Trailing Silence" sein.
- **Phrase Length Min** und **Phrase Length Max** bestimmen die maximale und minimale erlaubte Länge einer Phrase in Millisekunden. Da die maximale Länge den Hardware-Ressourcen-Verbrauch beeinflusst, kann sie direkt zu Performance-Problemen führen, wenn sie zu hoch ist.
- **Accuracy** sollte immer auf dem standardmäßigen Maximum gehalten werden.
- **Garbage** steuert die die Stärke des allgemeinen Müllmodells
- **NBest Second** sollte für einen guten Kompromiss aus Hardware-Nutzung und Genauigkeit auf dem Standardwert bleiben.

7 Problembhebung

vicCONTROL Designer startet nicht

Falls die angezeigte Fehlermeldung den Text "(x86)" enthält, dann stellen Sie bitte zuerst sicher, dass ein **64 Bit** JRE (Java Runtime Environment) oder JDK (Java Development Kit) installiert ist und das dieses über den Systempfad (PATH) verfügbar ist.

Eine einfache Möglichkeit diesen Zustand zu erreichen, besteht darin eventuell vorhandenen JREs oder JDKs zu deinstallieren und danach ausschließlich ein 64 Bit JRE oder JDK zu installieren.

Fehlermeldung "Can not build for recognizer"

Bitte stellen Sie sicher, dass das "Visual C++ Redistributable for Visual Studio 2012" für **x86** auf Ihrem Rechner installiert ist.

Übertragungsfehler

Übertragen des Dialogs mit Hilfe von vicCONTROL Designer schlug fehl und lässt sich auch nicht mehr durchführen. Bitte nutzen Sie das Recovery-Programm, das in Abschnitt 3.3 beschrieben ist, um die Firmware neu einzuspielen.

Fehler beim (Wieder-)Öffnen einer Datei

- Bitte schließen Sie das Unterfenster der Datei im Hauptfenster
- Öffnen Sie die Datei erneut, indem Sie im Menü File → Open wählen

8 Migration auf vicCONTROL 4

Mit Version 4.0 wurden technische Kernkomponenten der vicCONTROL Sprachsteuerung überarbeitet.

Das grundlegende Verhalten wurde weitestgehend beibehalten.

Dieses Dokument versucht, die wichtigsten Änderungen darzustellen und wie mit ihnen in bestehenden vicCONTROL Projekten umgegangen werden kann.

Wichtige Änderungen

- Seit 3.9
 - neues Baseboard vicBASE-3_5 das immer mit dem vicCORE-3_2 Coreboard geliefert wird
 - I/O-Logic der Pins des vicCORE-3_2 invertiert
 - Coreboard vicCORE-2 und das passende Baseboard vicBASE-2 (EMVCR) werden nicht mehr unterstützt
 - neue Sprachen (Arabisch, Chinesisch, Thai)
 - Anzahl der verfügbaren Audio-Prompts, die abgespielt werden können, sind auf 8 reduziert
 - leichte Unterschiede der Performance möglich
- Seit 3.1
 - viele neue Sprachen werden unterstützt
 - neue Spracherkennungskomponente (Engine)
 - * Bei gleicher Hardware kann es zu leichten Geschwindigkeitseinbußen kommen.
 - * Die Parametrierung und Optimierung ist ggf. neu durchzuführen.
 - Änderungen am seriellen Protokoll
 - * Das Protokoll ist weitgehend das gleiche Protokoll wie in den Vorgängerversionen. Durch die neue Engine kann sich aber das Timing und die Art der Ausgaben im konkreten Anwendungsfall verändert haben.
 - * Unnötige leere Meldungen und Fehlermeldungen wurden entfernt.
 - * Es wird UTF-8 statt ISO8859-1 verwendet, um der gestiegenen Zahl unterstützter Sprachen gerecht zu werden.
 - * Als Zeilenende-Zeichen wird jetzt CR statt LF verwendet. Dies gestattet eine bessere Kompatibilität mit UTF-8 fähigen Terminal Emulatoren.