

Hallo, Datenschutz!

Lokale Sprachsteuerung selbst gebaut

Virtuelle Assistenten wie Siri, Amazon Echo oder Google Home, die per Spracheingabe gesteuert werden, sind für den Anwender ein praktisches, zusätzliches User-Interface. Allerdings funktionieren diese Systeme nur durch Intelligenz in der Cloud, die die Spracheingabe analysiert und mittels programmierter Vorgaben entsprechende Aktionen auslöst. Wir schauen uns in diesem Beitrag zur Sprachsteuerung mit vicCONTROL go ein Offline-System an, das lokal Spracheingaben erkennt, auf sie reagiert und beispielsweise direkt Geräte ansteuern kann. Außerdem programmieren wir mithilfe von Node-RED eine Smart Home Ansteuerung mit Spracherkennung und Sprachausgabe.

The screenshot displays the vicCONTROLDesigner interface. The top part shows a 'Transitions' table with columns for 'Old Menu', 'Triggerclass', 'New Menu', and 'Comment'. Below this is a 'Connection' section with settings for COM3, 115200 baud rate, and a 'Test' section with 'Start Recognizer' and 'Stop Recognizer' buttons. The 'Output' section shows the current state as 'Hauptmenü' and the previous state as 'Warten auf Aktivierungswort'. On the right, a state machine diagram illustrates the logic: 'Warten auf Aktivierungswort' leads to 'Hauptmenü' on 'Hallo Vicky' (with a red 'Timeout' label). 'Hauptmenü' leads to 'Licht' on 'Licht' and 'Büro' on 'Büro', and back to 'Hauptmenü' on 'zurück' (with a red 'Timeout' label). 'Licht' leads to 'Licht an' on 'Licht an' and 'Licht aus' on 'Licht aus'. 'Büro' leads to 'Luft' on 'Luft' and back to 'Hauptmenü' on 'zurück' (with a red 'Timeout' label). All states have a 'Timeout:5000ms' label.

Old Menu	Triggerclass	New Menu	Comment
Warten auf Aktivierungswort	Hallo Vicky	Hauptmenü	
Hauptmenü	Licht	Licht	
Licht	Licht aus	Licht	
Licht	zurück	Hauptmenü	
Licht	licht an	licht	
Büro	Luft	Büro	
Hauptmenü	Büro	Büro	
Büro	zurück	Hauptmenü	



Hey, Siri!

Spracherkennung hat im Massenmarkt erst mit dem Smartphone so richtig Einzug gehalten. Zwar wird schon seit den 1960er-Jahren an der Spracherkennung geforscht und Ende der 1990er-Jahre erschienen erste PC-basierte Spracherkennungssysteme, die hauptsächlich als Diktiersoftware genutzt wurden. Doch die Grundlagen der Spracherkennung sind hoch komplex und erst durch den Einsatz neuer Technologien im Bereich Künstlicher Intelligenz und zunehmender Vernetzung von weltweiten Servern für Endanwender möglich. Sehr ausführlich und gut verständlich beschrieben sind die umfangreichen Grundlagen der Spracherkennung zum Beispiel in „The Voice in the Machine: Building Computers That Understand Speech“ [1] von Roberto Pieraccini.

Die breite Masse erreichte die Technologie erst 2011, als Apple mit Siri (Speech Interpretation and Recognition Interface) die Möglichkeit für den Nutzer einführte, gesprochene Befehle mit dem Smartphone aufzunehmen und auszuwerten. Hier, wie auch bei allen anderen virtuellen Assistenten, wird die Sprache an eine Software weitergeleitet, interpretiert und je nach vorgegebener Programmierung eine Information an den Anwender zurückzugeben oder eine Aktion ausgelöst. Spricht man also „Hey, Siri!“ in sein Smartphone, wird dieser Dienst zunächst aktiviert und erst dann erfolgt die eigentliche Anfrage des Nutzers und die anschließende Antwort des Sprachassistenten.

Ein erstes eigenständiges Gerät, das – abgesehen von einem Drehring und zwei Tasten – ausschließlich Spracherkennung als Benutzerschnittstelle bietet, führte das US-amerikanische Unternehmen Amazon Ende 2014 mit dem „Amazon Echo“ ein (Bild 1).

Neben der Möglichkeit zur Spracherkennung besaß das erste Gerät aus dieser Reihe einen integrierten Lautsprecher, Bluetooth-Funktionalität und war per WLAN an das heimische Netzwerk angebunden. Auch hier wird mit dem sogenannten Aktivierungswort (wake word) „Alexa“ zunächst die Spracherkennung aktiviert und die dann folgende Spracheingabe an die Server des Unternehmens weitergeleitet, um dort von dem Dienst Alexa Voice Services (AVS), der für die Spracherkennung zuständig ist, ausgewertet zu werden. Dabei „lauscht“ der intelligente Lautsprecher mithilfe seines Mikrofonarrays, das beim Amazon Echo der ersten Generation aus sechs im Kreis und einem zentral angeordneten MEMS-Mikrofon

Bild 1: Amazon Echo
Erste Generation



besteht (Bild 2, Bauteile MK1–MK7, rote Kreise), zwar die ganze Zeit. Aber erst nachdem das Aktivierungswort, das per Hardware ausgewertet wird, erkannt wurde, leitet dieser die Sprach-Audio-Daten an die Server des Unternehmens weiter.

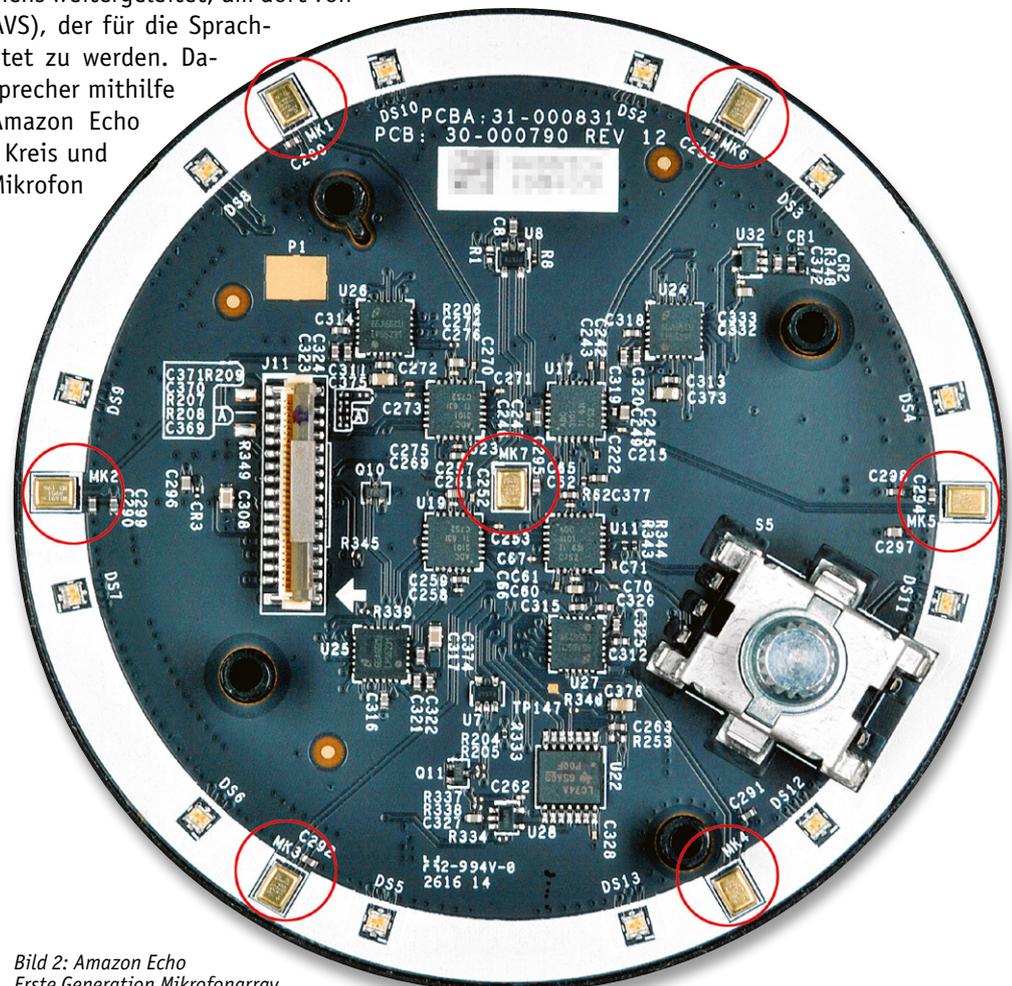


Bild 2: Amazon Echo
Erste Generation Mikrofonarray



Bild 3: Deaktivierte Spracherkennung

Die Mikrofone und damit die Spracherkennung lassen sich per Taster komplett abstellen (Bild 3). Durch die kreisförmige Anordnung des Mikrofonarrays kann übrigens nicht nur die Sprache an sich, sondern auch die Richtung, aus der diese Spracheingabe erfolgt, erkannt werden.

Mit bei Amazon sogenannten Skills lässt sich die Funktionalität des Sprachassistenten erweitern. So können neben dem Abspielen von Musik zusätzliche Dienste von Drittanbietern abgerufen und so beispielsweise Smart Home Geräte angesteuert werden. Die Ausgabe der Informationen in Form der Sprachausgabe erfolgt beim „Echo“ über den integrierten Lautsprecher. Über eine App auf dem Smartphone kann man zudem die Historie der Spracheingaben abrufen, Sprach-Audio-Daten löschen oder zusätzliche Informationen erhalten und Einstellungen vornehmen.

Mittlerweile gibt es zahlreiche weitere Geräte mit integrierter Spracherkennung von Amazon. Andere Anbieter wie Google haben mit „Google Home“ im Bereich der Spracherkennung nachgezogen.

Datenschutz und Datensicherheit

Datenschutz und Datensicherheit von Daten, die auf (fremden) Servern, heute umgangssprachlich als „Cloud“ bezeichnet, verarbeitet und gespeichert werden, sind eine wichtige Anforderung an diese Systeme. Wer will schon seine privaten Gespräche, die möglicherweise durch einen Sprachassistenten zufällig aufgezeichnet wurden, einer breiten Öffentlichkeit zugänglich machen oder Inhalte, die unter Umständen Geschäftsgeheimnisse verraten, der Konkurrenz mitteilen? Denn nicht immer wird das Aktivierungswort einwandfrei identifiziert und so können Aufzeichnungen auch ungewollt starten.

Die Problematik zum Datenschutz und zur Datensicherheit betrifft übrigens nicht nur Sprachassistenten. Sie ist heute allgegenwärtig durch die Nutzung von Messenger-Apps, Video-Konferenz-Tools und anderen cloudbasierten Diensten, die unter diesen Aspekten kritisch zu betrachten sind.

Am Beispiel von Amazon, das die Spracheingaben dauerhaft in der Amazon-Cloud speichert, wenn man sie nicht selbst in seinem Profil löscht, sieht man, wie leicht die zuvor sicher geglaubten Daten an Dritte gelangen können. So ereignete sich eine gravierende Datenpanne Ende 2018. Die deutsche Niederlassung von Amazon übermittelte einem Kunden auf dessen Datenauskunftsantrag rund 1700 Audiodateien und eine PDF-Datei mit Informationen zur Nutzung. Der Kunde hatte allerdings nie den Sprachassistenten verwandt, die Daten stammten von einem anderen Nutzer [2].

Auch wenn es sich dabei möglicherweise um einen Einzelfall gehandelt hat, wird klar, wie einfach hochsensible Daten in fremde Hände gelangen können. Es müssen also nicht unbedingt erst Server von Hackern kompromittiert werden, um an persönliche Daten Dritter zu gelangen.

Ganz wichtig: Cloud-Dienst ist nicht gleich Cloud-Dienst. Stehen die Server beispielsweise in Deutschland und die Daten werden geschützt und möglicherweise sogar anonymisiert übertragen, überwiegen die Vorteile möglicherweise sogar gegenüber lokalen Systemen.

Lokale Spracherkennung

Während Siri, Alexa, Google Home & Co. überall auf der Welt verteilte Server zur Sprachanalyse und Verarbeitung nutzen, hat die Firma „voice INTER connect“ aus Dresden mit vicCONTROL go eine Sprachsteuerung

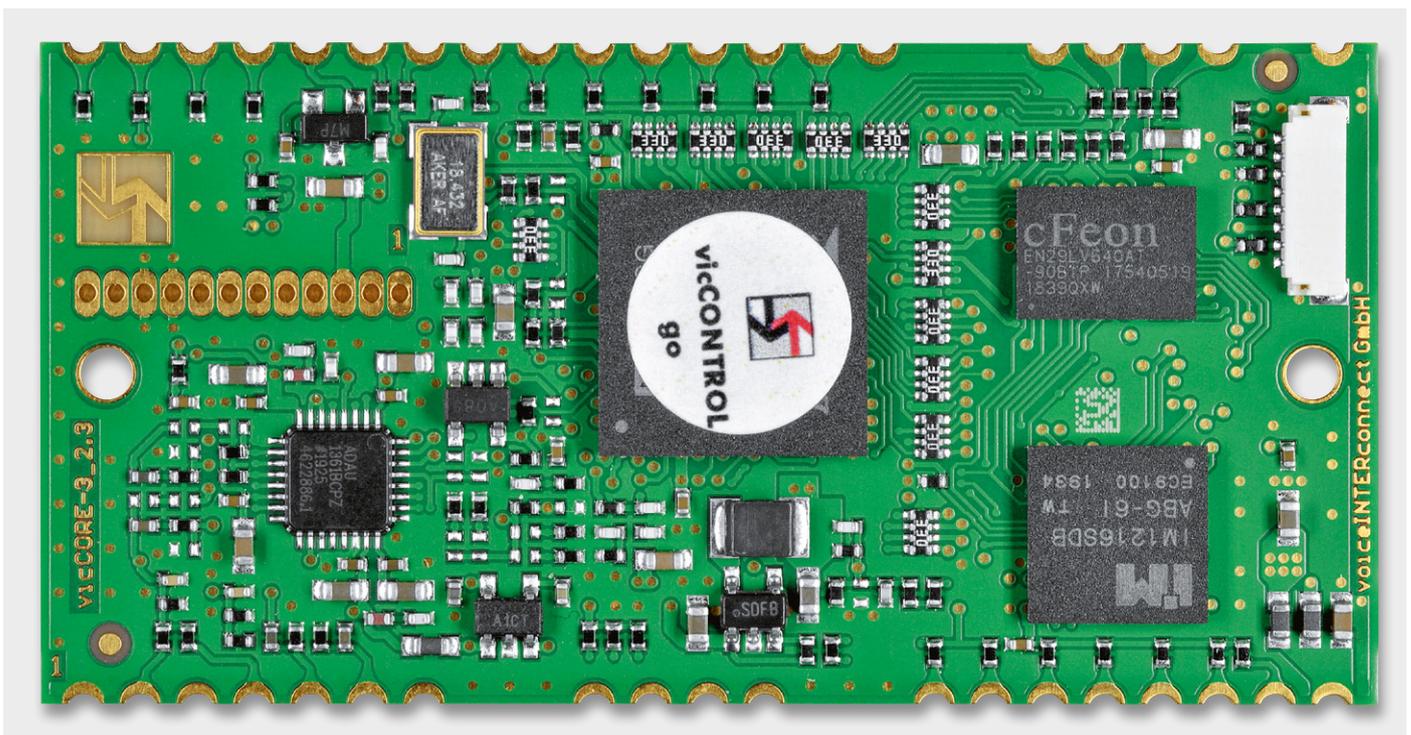


Bild 4: vicCONTROL go stamp



erung mit lokaler Signalverarbeitung entwickelt. Das Unternehmen beschäftigt sich seit über 20 Jahren mit Kommunikation und ist neben der Sprachsteuerung in Bereichen wie Intercom, Voice-over-IP (VoIP), Echo-kompensation und Beamforming tätig.

Dabei arbeitet vicCONTROL go ähnlich wie die großen Anbieter, nur dass weder eine Internetverbindung noch (fremde) Server zur Spracherkennung notwendig sind. Auch hier kann ein Aktivierungswort verwendet werden und anschließend wertet das System lokal auf einer Hardware mit digitalem Signalprozessor (DSP – Analog Devices Blackfin ADSP-BF512) die Spracheingabe aus. Zur Spracherkennung sind 30 länderspezifische Sprachmodelle verfügbar, allerdings kann jeweils nur eine Sprache gleichzeitig auf dem System verwendet werden. Die Sprachdaten bleiben damit geschützt, das System arbeitet auch bei fehlender Internetverbindung weiter und bietet somit eine hohe Zuverlässigkeit und Verfügbarkeit. Zudem ist eine sehr schnelle Verarbeitung aufgrund der nicht notwendigen Internetverbindung möglich und eine Industrietauglichkeit wird laut Unternehmen durch einen erhöhten Temperaturbereich und die Robustheit der Sprachsteuerung gewährleistet. Allerdings ist die sogenannte Dialoggröße – dazu kommen wir noch bei der Beschreibung des Systems – auf derzeit 500 Kommandos beschränkt. Das sollte für den Elektronik-Enthusiasten, Smart Home Fan, aber auch für professionelle, genau definierte Anwendungen völlig ausreichen.

vicCONTROL go

Für die lokale Sprachsteuerung bietet die Firma voice INTER connect zwei verschiedene Ausbaustufen von zugrundeliegender Elektronik an: den *vicCONTROL go stamp* (Bild 4, [3]) und das *vicCONTROL go kit* (Bild 5, [3]).

Bei den Bildern (4 und 5) der beiden Varianten erkennt man mit dem vicCORE-Modul die Kernkomponente der lokalen Spracherkennung. Das 60 x 30 mm große vicCONTROL go stamp entspricht diesem Modul und wird ohne weitere Peripherie oder Software [3] geliefert.

Das vicCONTROL go kit ist das Evaluierungsboard (146 x 75 mm) für die Sprachsteuerung und beinhaltet neben dem Kernmodul zahlreiche Peripherie wie Energieversorgung und Kommunikation über USB, ein umfangreiches Audiointerface und verschiedene Anschlussmöglichkeiten für bis zu je zehn digitale Ein- und Ausgänge. Zudem liegt hier die benötigte Software wie:

- vicCONTROL Designer (Erzeugen individueller Sprachdialoge für vicCONTROL go)
 - Firmware Update (Firmware Update und Recovery Tool)
 - Wavesurfer (Audio-Editor)
 - USB-Treiber
- bereits bei.

Mit beiden Boards kann man eine lokale Sprachsteuerung realisieren, bei dem vicCONTROL go stamp muss man allerdings die Software, die zum Erzeugen der Sprachdialoge notwendig ist, separat erwerben und die Peripherie zur Ansteuerung zum Teil selbst realisieren. Unabhängig davon benötigt man bei beiden außerdem ein Mikrofon – z. B. ein Elektret- (s. [Elektronikwissen, \[4\]](#)) oder ein MEMS-Mikrofon [5] zur Aufnahme der Sprache und (optional) einen Lautsprecher zur Sprachausgabe sowie beim vicCONTROL go stamp einen USB-Seriell-Wandler [6] und gegebenenfalls optional bei Verwendung eines Lautsprechers einen Audioverstärker.

Schnittstellen

Schauen wir uns die Möglichkeiten der elektronischen Ansteuerung an und werfen wir zunächst einen Blick auf das Blockschaltbild des Evaluierungsboards vicCONTROL go kit (Bild 6). Im Blockschaltbild sind alle vorhandenen Schnittstellen aufgeführt. Bei der Spracheingabe sind mehrere Möglichkeiten gegeben, Mikrofone anzuschließen. Neben dem direkten Verwenden u. a. von handelsüblichen Elektret-/MEMS-Mikrofonen (analog) ist zum Beispiel auch der Anschluss eines Headsets über die DECT-Schnittstelle möglich. Bei der Audioausgabe wird auf dem rechten Kanal das Audiosignal des aktuellen Eingangssignals – der eingegebenen Sprache – ausgegeben, auf dem linken Kanal können sogenannte „Prompts“ – vordefinierte Sprachausgaben oder Töne – erzeugt werden.

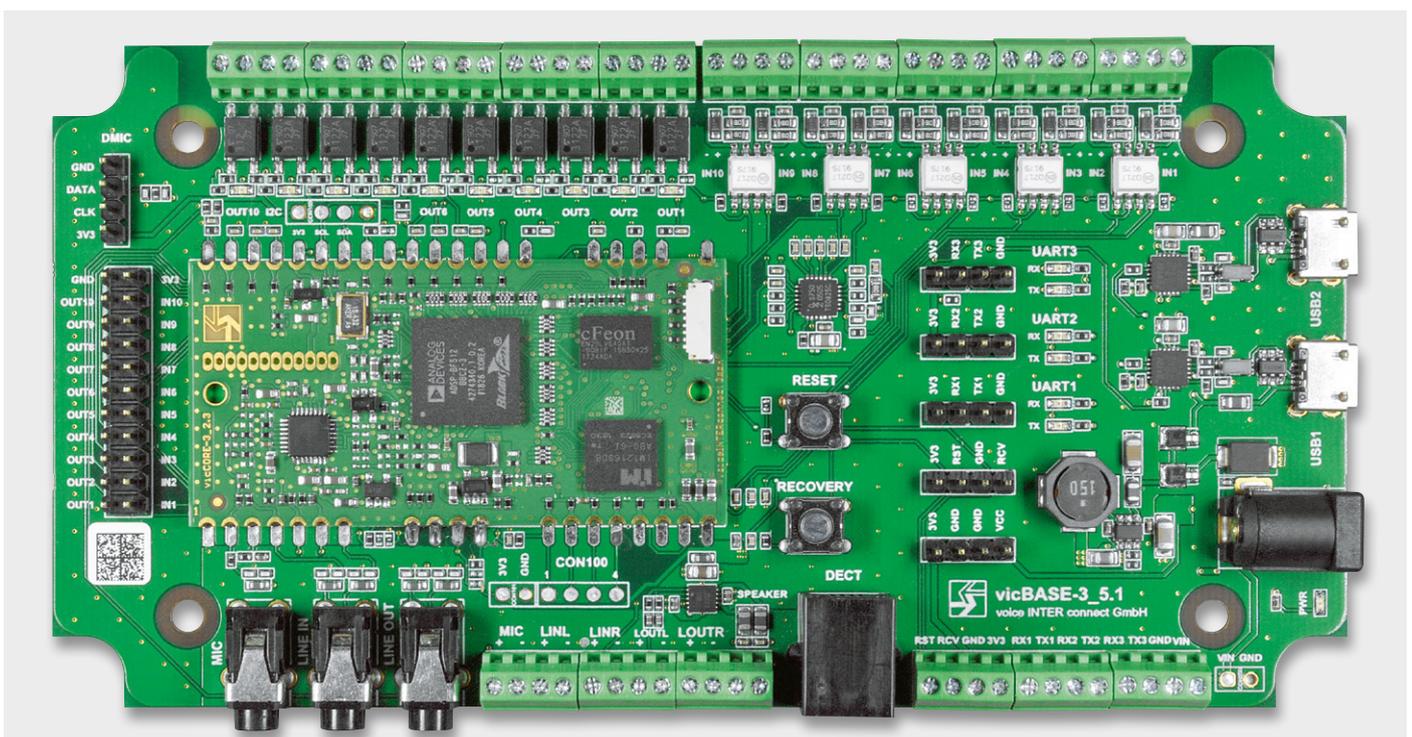


Bild 5: vicCONTROL go kit

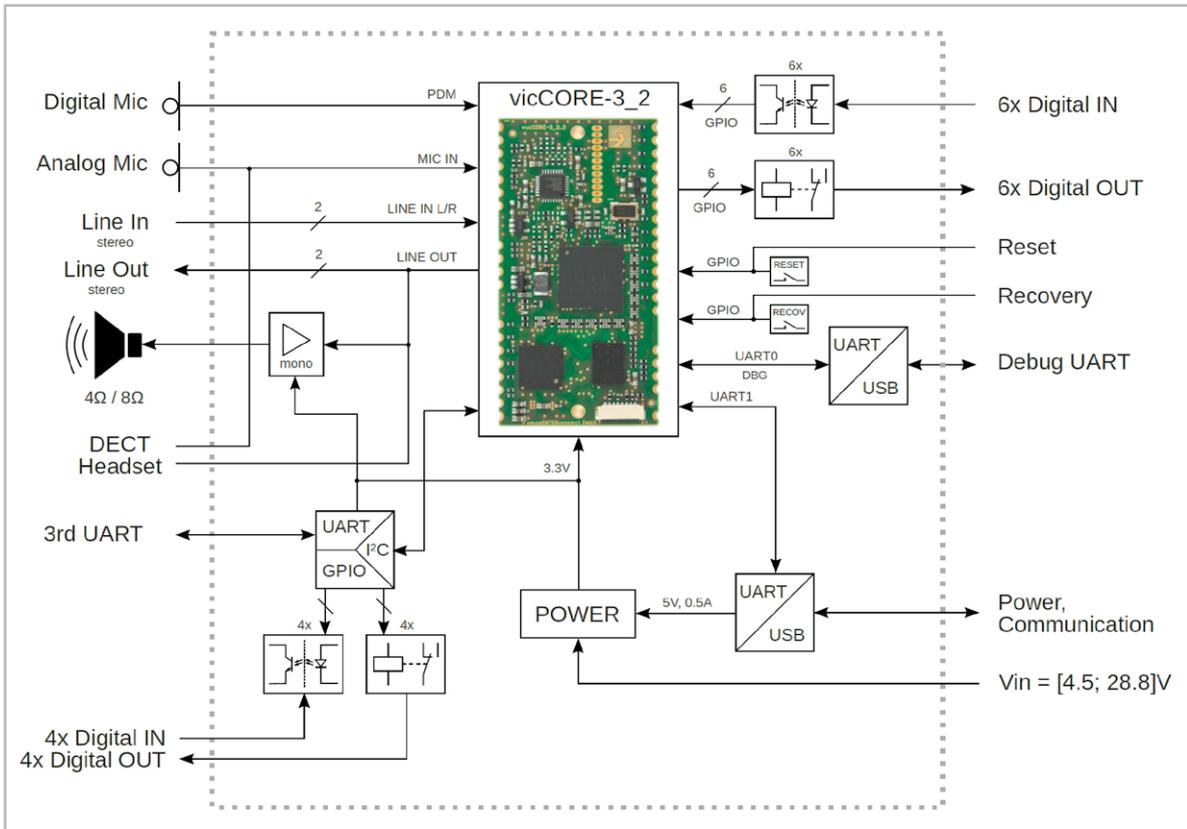


Bild 6: Blockschaltbild vicCONTROL go kit (Quelle: Dokumentation vicCONTROL go [7])

Beim Evaluierungsboard stehen zehn digitale Ein- und Ausgänge zur Verfügung. Die Spannung für die per Optokoppler galvanisch getrennten Digitaleingänge kann bei maximal 12 V (max. 60 mA) liegen. Die Ausgänge können maximal 60 V mit 1,4 A treiben. Auf dem kleinen vicCONTROL go stamp ist die Ansteu-

erung auf sechs digitale Ein- und Ausgänge beschränkt, die mit einer eigenen Beschaltung für die Ein- und Ausgänge versehen werden müssen.

Es gibt zwei aktive UART-Schnittstellen, die beim Evaluierungsboard go kit der Stromversorgung und Kommunikation (UART1) sowie dem Debugging (UART2) dienen. Beim vicCONTROL go stamp muss die UART-Schnittstelle zur Kommunikation mit einem USB-Seriell-Wandler [6] ver-

The screenshot shows the vicCONTROL Designer software interface. The top window displays a transition table with columns for Old Menu, Triggerclass, New Menu, and Comment. The 'Licht' row is highlighted, and a red '1' is next to the 'Licht' comment. The Properties window on the right shows a list of properties for 'Custom serial output' with values set to 'Licht1An'. A red '2' is next to the 'Licht1An' value. The Platform window at the bottom left shows connection settings for COM3, Baudrate: 115200, Data bits: 8, Parity: N. A red '3' is next to the 'Start Recognizer' button. The Console window at the bottom right shows a state machine diagram with states: 'Warten auf Aktivierungswort', 'Hauptmenü', 'Licht', and 'Büro'. Transitions are labeled with 'Hallo Vicky', 'Licht', 'Büro', 'zurück', and 'Timeout'. A red '4' is next to the 'Hauptmenü' state.

Bild 7: vicCONTROL Designer



bunden und die Stromversorgung separat realisiert werden. Bei dem Evaluierungsboard besteht neben der Stromversorgung per USB noch die Möglichkeit, per Hohlsteckerbuchse Spannung zuzuführen.

Let's go

Zunächst muss die Software „vicCONTROL Designer“ (Modellierungssoftware), die zum Erzeugen individueller Sprachdialoge für vicCONTROL go verwendet wird, installiert werden. Dazu wird ein 64-Bit-Java-Run-time-Environment benötigt, z. B. ojdkbuild, das mit dem Quellcode von OpenJDK aus dem CentOS-Projekt erstellt wurde [8]. Die Modellierungssoftware läuft nur unter Windows.

In Bild 7 sehen wir die einzelnen Bestandteile der Software mit dem Hauptfenster zur Konfiguration des Dialogs 1, Eigenschaften zu einzelnen Elementen 2, Verbindungs- und Debug-Informationen 3 und einer vereinfachten Visualisierung des Dialog-Ablaufs 4.

Dialog

Die Konfiguration aller Elemente für die Spracherkennung, die in einer Datei zusammengefasst und später auf das Board (go stamp/go kit) heraufgeladen wird, wird als Dialog bezeichnet. Die Dialoggröße ist derzeit auf 500 Kommandos, 200 Menüs und 500 „Transitions“ (s. u.) beschränkt, um den Speicherverbrauch auf der Embedded-Plattform zu begrenzen.

Hauptfenster 1

Im Hauptfenster wird der Ablauf des Dialogs konfiguriert. Hier wird das Aktivierungswort definiert und die einzelnen Kommandos (Registerkarte „Commands/Triggerclasses“) werden festgelegt. Die Kommandos können zu weiteren Untermenüs („Menüs“) bzw. zu einem Übergang („Transition“), der mit einer bestimmten Eigenschaft versehen wird, führen. Außerdem können in „System Settings“ entsprechende Einstellungen wie beispielsweise die Verstärkung des Mikrofoneingangs oder des seriellen Ports vorgenommen werden.

Eigenschaften (2 „Properties“)

Hier können die Eigenschaften und Werte zu den im Hauptfenster aufgeführten Elementen eingestellt werden, also beispielsweise das Ansteuern eines Ausgangs bei Erkennung eines bestimmten Sprachbefehls (Übergang/Transition).

Verbindungs- und Debug-Informationen (3 „Platform“)

In diesem Fenster findet die Konfiguration und Verbindung über die serielle Schnittstelle statt und der Prozess zur Spracherkennung kann gestartet und beendet werden. Außerdem gibt es in einer Konsole ein Debug-Fenster zur aktuellen Spracherkennung.

Visualisierung des Dialog-Ablaufs (4 „Console/DialogImageView“)

In diesem Fenster werden die im Hauptfenster konfigurierten Elemente in einer vereinfachten Vorschau dargestellt und der Build-/Upload-Vorgang in einem Log dargestellt.

Die Möglichkeiten der Einstellungen in der Modellierungssoftware vicCONTROL Designer sind sehr umfangreich. Deshalb werden wir an einem konkreten Beispiel sowohl die Konfiguration und den Prozess der Spracherkennung als auch die Ansteuerung einer Lampe in Verbindung mit einem Raspberry Pi und der Open-Source-Software Node-RED zeigen. Für unser Beispiel nutzen wir das Evaluierungsboard, beschreiben im Folgenden aber auch die Unterschiede zum kleineren vicCONTROL go stamp für dieses spezielle Projekt.

Hardware

Mikrofon

Als Aufnahmequelle nutzen wir ein einfaches Elektret-Mikrofon [4], das wir am Mikrofoneingang (Klinke/Schraubklemme) des Evaluierungsboards anschließen. Bei beiden Boards ist die notwendige Beschaltung für das Mikrofon bereits vorhanden (Bild 8).

Das Elektret-Mikrofon kann daher auch beim vicCONTROL go stamp direkt verbunden werden, je nach zwei- oder dreipoliger Ausführung (s. Elektronikwissen) muss es entsprechend angeschlossen werden. Der Ausgang MIC_BIAS (3,3 V) dient dabei der Vorspannung (Spannungsversorgung) des Elektret-Mikrofons, MIC_IN_R als Eingang für die Mikrofonsignale und die Mikrofonmasse wird mit GND für die Audioanschlüsse verbunden.

Lautsprecher

Um eine für die Testphase vorläufige Kontrolle über eine erfolgreiche Spracherkennung zu haben, schließen wir am Lautsprecherausgang des Evaluierungsboards einen 8-Ω-Lautsprecher (z. B. [9]) an. Dieser gibt über einen auf dem Board vorhandenen Verstärker für den linken Kanal von LINE_OUT die WAV-Datei (Prompt) aus, die in der Modellierungssoftware dafür ausgewählt wurde. Beim kleineren Board muss für diesen Zweck noch ein Verstärker an den jeweiligen Kanal angeschlossen werden, um den Lautsprecher zu treiben. Über LINE_OUT_R kann man die Ausgabe des aktuellen Eingangssignals abhören und so für Testzwecke die Sprachqualität akustisch beurteilen.

Kommunikation und Spannungsversorgung

Die Spannungsversorgung und Kommunikation erfolgt beim Evaluierungsboard per Micro-USB vom PC, beim vicCONTROL go stamp müssen für diese Zwecke eine externe 3,3-V-Spannungsversorgung (max. 110 mA werden benötigt) und ein USB-Seriell-Wandler [6] angeschlossen werden.

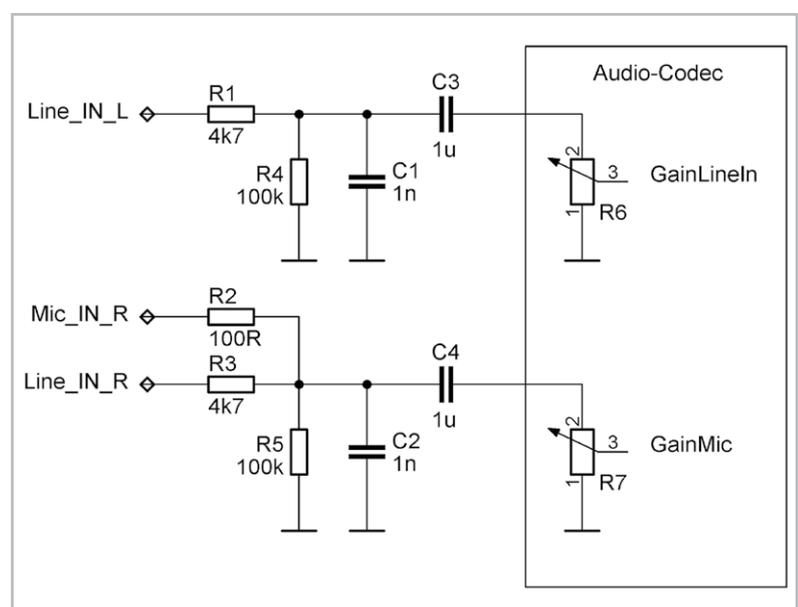
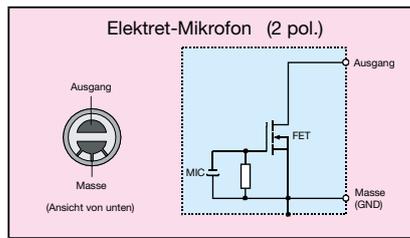
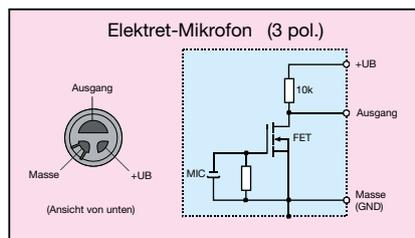
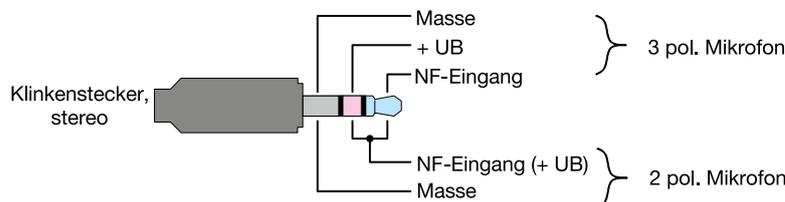


Bild 8: Beschaltung der Audioeingänge (Quelle: Dokumentation vicCONTROL go [7])



Die genaue Anschlussbelegung des Klinkensteckers und verschiedener Elektret-Mikrofone



Anschlussbelegung von Elektret-Mikrofonen

Elektret-Mikrofone bzw. Elektret-Mikrofonkapseln sind die häufigsten verwendeten Mikrofone weltweit. Elektret-Mikrofone haben eine kompakte Bauweise, einen geringen Preis und gute Signalqualität und werden in sehr großen Stückzahlen hergestellt. Sie finden Anwendung in verschiedensten Geräten wie z. B. Headsets, Kommunikationsgeräten, Handys usw. Diese Mikrofone benötigen eine Versorgungsspannung, da sich im Inneren ein

Verstärker (FET) befindet. Es gibt 2-polige und 3-polige Bauformen (siehe Bild oben). Bei der 3-poligen Variante ist der Drain-Widerstand im Mikrofongehäuse integriert, weshalb hier ein weiterer Anschluss benötigt wird. In jüngster Zeit haben sich die 2-poligen Mikrofone durchgesetzt. Im Bild ist zudem die Belegung von Klinkensteckern dargestellt, die in der Regel zur Kontaktierung von Elektret-Mikrofonen verwendet werden.

Detaillierte Informationen zur Funktionsweise von Elektret-Mikrofonen findet man unter:

<https://de.wikipedia.org/wiki/Elektret-Mikrofon>

Software

Mit den hardwareseitig vorbereiteten Boards kann nun eine Verbindung mit der Modellierungssoftware vicCONTROL Designer über den entsprechenden COM-Port hergestellt werden. Ab Werk ist die deutsche Sprache als Modell für die Spracherkennung eingestellt und ein Spracherkennungsbeispiel aufgespielt.

Per Klick auf „Start Recognizer“ startet man den Testmodus der Spracherkennung und kann mit „Hallo, Vicky“ das Aktivierungswort testen. Über den Lautsprecher sollte dabei ein Bestätigungston ausgegeben werden. Von hier aus kann man per Sprachbefehl „Heizung“ in ein Untermenü gelangen und eine fiktive Heizung mit dem Befehl „wärmer“ steuern.

Alle Sprachbefehle sollten mit einem Bestätigungston quittiert werden. Im Outputfenster im Bereich Platform werden dazu weitere Informationen ausgegeben.

Da nun der hard- und softwareseitige Aufbau steht, machen wir uns an unser eigenes Beispiel zur Steuerung einer Lampe, das im Download-Bereich dieses Beitrags [10] zur Verfügung steht. Hier findet man auch ein erweitertes Beispiel zum Auslesen eines CO₂-Sensors und die später für Node-RED benötigten Flows.

Dialog erstellen

Wir greifen einen Schritt vor und schauen uns in Bild 9 zunächst den bereits fertiggestellten Dialog in der Visualisierung an. Mit der grafischen Aufbereitung der Konfiguration kann der Ablauf gut erklärt werden und man erhält außerdem schnell einen Überblick über den gesamten Dialog.

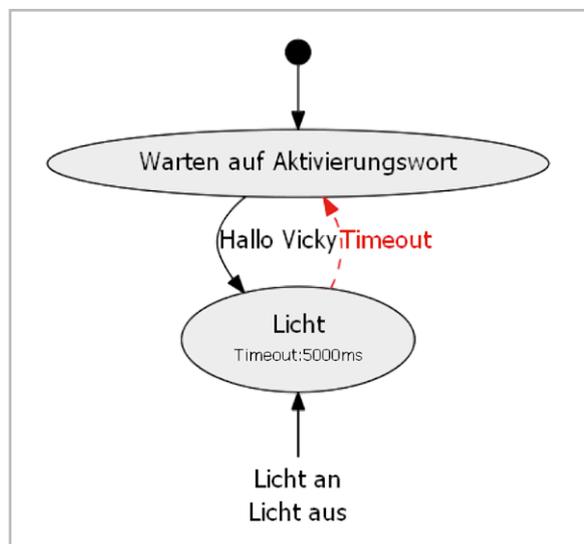


Bild 9: Visualisierung des Dialogs

Warten auf das Aktivierungswort

Die vicCONTROL go-Boards lauschen im Stand-alone-Modus nach Inbetriebnahme automatisch auf das Aktivierungswort – in unserem Fall „Hallo, Vicky“. Bei der Verbindung mit der Modellierungssoftware muss man mit „Start Recognizer“ diesen Prozess manuell starten.

In der Visualisierung wird nur der Befehl „Hallo Vicky“ aufgeführt, diese sogenannte Triggerclass kann aber verschiedene Inhalte (Content) haben. In dem Beispiel mit deutschen Dialogen sind dafür z. B. „Hallo Vicky“, „Hallo Wikki“, „Hallo Wiggi“, „Hallowikki“, „HalloWiggi“, „allo Wikki“, „allo Wiggi“, „allowikki“ und „allowiggi“ vorgegeben.

Licht

Wurde das Aktivierungswort erkannt, geht der Dialog in das von uns so benannte Menü „Licht“. Dieser Übergang (Transition) wird mit einem Bestätigungston (Prompt) quittiert, soweit dieser dafür aktiviert wurde. Der Dialog bleibt 5000 ms (5 sek) im Menü Licht und wartet dort auf eine Spracheingabe. Erfolgt das Timeout, springt der Dialog wieder zurück und wartet erneut auf das Aktivierungswort.



Triggerclass	Content	Class-ID
Hallo Vicky	Hallo Vicky, Hallo Wikki, Hallo Wiggi, Hallowikki, HalloWiggi, allo Wikki, allo Wiggi, allowikki, allowiggi	12
Licht	Licht, Lampe, Gehe zum Licht, Lichtsteuerung, Zur Lichtsteuerung bitte	1
Licht an	an, ein, anschalten, einschalten, Lampe an, Lampe einschalten, Licht einschalten, Licht an, Mach mal hell, hell, Lampe ein	2
Licht aus	aus, ausschalten, Lampe aus, Licht aus, Lampe ausschalten, Mach mal dunkel, dunkel	3

Bild 10: Triggerclasses und Content

Licht an/Licht aus

Wurde im Menü Licht ein Sprachbefehl aus der Triggerclass „Licht an“/„Licht aus“ erkannt, erfolgt wieder ein Übergang. Auch hier kann eine festgelegte Aktion für den Übergang von Licht zu „Licht an/Licht aus“ ausgeführt werden, beispielsweise das Aktivieren eines Ausgangs auf dem vicCONTROL go kit-Board und der nachfolgenden Ansteuerung eines Verbrauchers oder das Versenden einer Nachricht über UART bzw. eine Bestätigung per Prompt.

Beim Evaluierungsboard sind die zehn vorhandenen Ausgänge bereits beschaltet. Beim vicCONTROL go stamp muss die externe Ansteuerung selbst realisiert werden.

Mit Menüs oder „One-Shot“

Wir haben hier bewusst beispielhaft einen Ansatz mit Menüs gewählt. Menüs und entsprechende Befehle können die Spracherkennung robuster machen. Im Smart Home könnte man so zum Beispiel Geräte in ver-

schiedenen Räumen (Wohnzimmer, Büro ...) mit entsprechenden Menüs anwählen. Die Triggerclass „Licht an“ wäre dann nur einmalig für alle Räume anzulegen und würde je nach Raum (Menü) ausgeführt werden.

Bei vicCONTROL go gibt es außerdem die Möglichkeit, für eine sogenannte „One-Shot-Aktivierung“ [7], bei der Aktivierungswort und Befehl in einem Zug (ohne Pause) gesagt werden können.

Triggerclass, Transitions, Timeouts

Für unser Beispiel erzeugen wir im Hauptfenster folgende Triggerclasses mit entsprechenden Inhalten (Bild 10):

Triggerclass	Content
Licht	Licht, Lampe, Gehe zum Licht, Lichtsteuerung, Zur Lichtsteuerung bitte
Licht an	an, ein, anschalten, einschalten, Lampe an, Lampe einschalten, Licht einschalten, Licht an, Mach mal hell, hell, Lampe ein
Licht aus	aus, ausschalten, Lampe aus, Licht aus, Lampe ausschalten, Mach mal dunkel, dunkel

The screenshot shows the vicCONTROLDesigner interface. The 'Transitions' table is highlighted with a red border and contains the following data:

Old Menu	Triggerclass	New Menu	Com...
Warten auf Aktivierungswort	Hallo Vicky	Licht	
Licht	Licht aus	Licht	
Licht	Licht an	Licht	

The 'Properties' panel on the right is also highlighted with a red border and shows the following configuration:

Property	Value
Custom serial output	Licht1Aus
In10 Guard	none
In1 Guard	none
In2 Guard	none
In3 Guard	none
In4 Guard	none
In5 Guard	none
In6 Guard	none
In7 Guard	none
In8 Guard	none
In9 Guard	none
Out10 Action	DontCare
Out10 Timeout	0
Out1 Action	SwitchOn
Out1 Timeout	1000
Out2 Action	DontCare
Out2 Timeout	0
Out3 Action	DontCare
Out3 Timeout	0
Out4 Action	DontCare
Out4 Timeout	0
Out5 Action	DontCare
Out5 Timeout	0
Out6 Action	DontCare
Out6 Timeout	0
Out7 Action	DontCare
Out7 Timeout	0
Out8 Action	DontCare
Out8 Timeout	0
Out9 Action	DontCare
Out9 Timeout	0
Prompt	Prompt01

Bild 11: Konfiguration der Transitions



Die Worte im Content sind dabei mögliche Phrasen, die geäußert werden müssen, damit der Befehl erkannt wird. In unserem Dialog bräuchte Licht keinen Content – wir haben das aber beispielhaft mit aufgeführt.

Als Nächstes erzeugen wir im Reiter „Menus“ ein Menü „Licht“ mit einem Timeout von 5000 ms und den Werten 4000 für Score High und Score Low. Für „Use wakeup word“ setzen wir „false“. Die Transitions konfigurieren wir wie in [Bild 11](#).

Zusätzlich wird bei allen Einträgen im Properties-Fenster der Prompt01 aktiviert, bei Licht → Licht an bei Custom serial output „Licht1An“ und bei Licht → Licht aus bei Custom serial output „Licht1Aus“ eingegeben.

Will man beim Übergang von Licht zu „Licht an“/„Licht aus“ nur einen Ausgang schalten, könnte man hier einen der zehn (sechs) Ausgänge an- oder ausschalten und das Timeout für diesen Schaltvorgang festlegen.

Nun muss noch das Verhalten bei einem Timeout für das Menü Licht eingestellt werden. Das geschieht unter Timeout Transitions ([Bild 12](#)).

Old Menu	New Menu	Comment
Licht	Warten auf Aktivierungswort	

Bild 12: Konfiguration der Timeout Transitions

Damit ist die Konfiguration beendet und wir können uns im Platform-Fenster per Preview die Visualisierung des Dialogs ansehen, die dem [Bild 9](#) entsprechen sollte. Anschließend kann dieser Dialog auf das Board hochgeladen und wie bei dem obigen Spracherkennungsbeispiel getestet werden.

Im Downloadbereich des Beitrags [\[10\]](#) haben wir dieses und ein erweitertes Beispiel mit zwei Menüs hinterlegt.

Ansteuerung mit Node-RED

Mit der Software Node-RED haben wir uns im ELVjournal bereits einige Male beschäftigt und verweisen daher u. a. auf den dreiteiligen Einführungskurs [\[11, 12, 13\]](#) und gehen nur an den wichtigsten Stellen auf die Konfiguration ein. Die Beispiel-Flows zu diesem Beitrag finden Sie ebenfalls im Download-Bereich [\[10\]](#).

Schauen wir uns die Anwendung in Node-RED an ([Bild 13](#)). Per Serial-in-Node **1** empfangen wir die Mitteilungen der vicCONTROL go-Boards,

die per UART gesendet werden (Licht1Aus/Licht1An), und geben diese über einen Debug-Node **2** aus. Dabei sieht man im Debug-Fenster, dass auch automatisch Statusmeldungen gesendet werden **3**, also zum Beispiel die erkannte Phrase und der aktuelle Status. Die Mitteilungen werden in der Dokumentation der Boards ausführlich beschrieben [\[7\]](#). Uns interessieren aber nur die beiden UART-Nachrichten, die wir für „Licht an“ (Licht1An) und „Licht aus“ (Licht1Aus) bei den Transitions konfiguriert haben **4**.

Den Datenstrom der seriellen Schnittstelle prüfen wir mithilfe der Switch-Nodes **5** auf diese Inhalte. Bei Vorhandensein weisen wir ihnen zum einen Text für die Audioausgabe zu **6**, die wir zum Debugging nutzen, zum anderen einen Wert **7**, den wir per MQTT versenden können. So kann beispielsweise eine per MQTT angeschlossene Lampe angesteuert werden **8**. Zur Funktionsweise von MQTT im Zusammenhang mit Node-RED haben wir im ELVjournal unter [\[13\]](#) eine Einführung in das Thema gegeben.

Hallo, Vicky – Licht an!

Zum Testen schließen wir ein vicCONTROL go-Board mit dem vorher hochgeladenen Dialog an der USB-Schnittstelle des Raspberry Pi an. Dann kopieren wir den heruntergeladenen Flow [\[10\]](#) und müssen in dem Serial-in-Node unter Umständen noch die Schnittstelle (z. B. /dev/ttyUSB0) konfigurieren. Anschließend deployen wir den Flow und sprechen die Phrase „Hallo, Vicky“. Das Aktivierungswort sollte erkannt werden und anschließend kann mit „Licht an“ bzw. „Licht aus“ eine Lampe per MQTT geschaltet werden.

Fazit

Im Stand-alone-Modus stehen bei den vicCONTROL go-Boards zehn (sechs) digitale Ein- und Ausgänge bereit, um per Sprachbefehl und dem daraufhin ausgelösten (zeitgesteuerten) Schaltbefehl Geräte direkt zu steuern oder Eingaben zu empfangen. Weitere Software ist für diesen Fall also gar nicht notwendig. Mit unserem Node-RED-Anwendungsbeispiel kratzen wir zudem nur an der Oberfläche der Möglichkeiten sowohl bei der lokalen Spracherkennung von vicCONTROL go als auch der nachgeschalteten Auswahl von Applikationen, von denen Node-RED nur eine der Möglichkeiten darstellt.

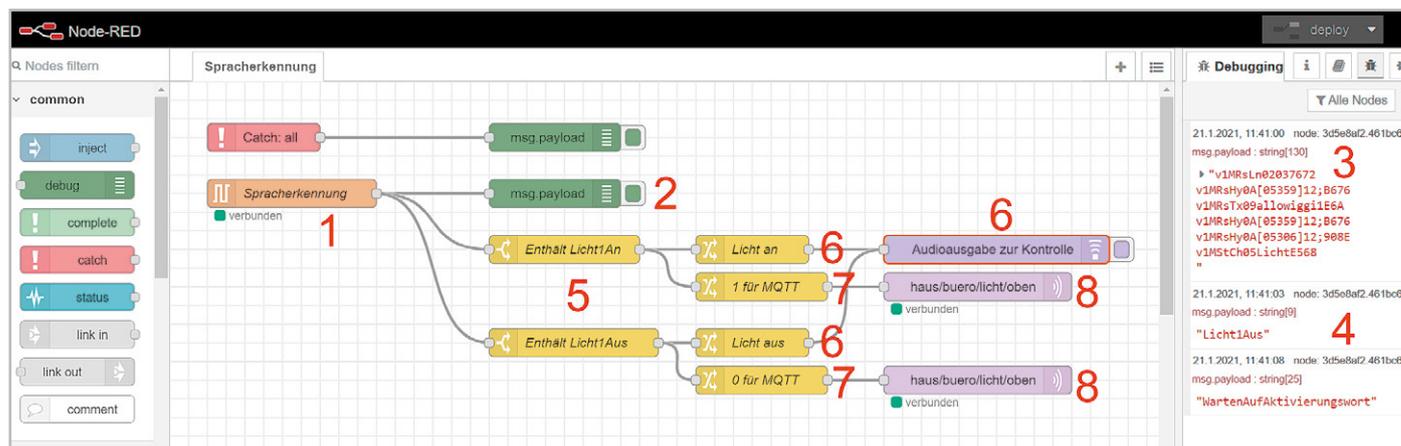


Bild 13: Node-RED-Flow zur Lichtsteuerung



Auch andere Softwarelösungen können serielle Nachrichten auswerten und entsprechende Aktionen auslösen. Prinzipiell kann ein vicCONTROL go-Board im Fall der Ausgabe von Mitteilungen über UART also an jede Anwendung angebunden werden, die serielle Daten verarbeiten kann.

Die Einarbeitungszeit ist durch die Modellierungssoftware, die keine Programmierkenntnisse voraussetzt, gering. Die Visualisierung hilft zudem (kompliziertere) Dialoge nachzuvollziehen. Trotz der Einfachheit

bietet das System einen frei wählbaren Wortschatz, um anschließend sprecherunabhängig gesteuert zu werden.

In unseren Tests ohne große weitere Anpassungen funktionierte das vicCONTROL go kit Evaluierungsboard mit einem einfachen Elektret-Mikrofon erstaunlich gut. Und das autonom, lokal und ohne Cloud.

ELV

Weitere Infos:

- [1] The Voice in the Machine: Building Computers That Understand Speech von Roberto Pieraccini, ISBN-13: 978-0262533294
- [2] c't, heise-Verlag, Amazon gibt intime Alexa-Sprachdateien preis:
<https://www.heise.de/newsticker/meldung/Amazon-gibt-intime-Sprachdateien-preis-4254716.html>
- [3] Evaluierungsboard: vicCONTROL go kit, Artikel-Nr. 251951; Basismodul: vicCONTROL go stamp, Artikel-Nr. 251952; Software: vicCONTROL go stick, Artikel-Nr. 251953
- [4] Elektret-Mikrofonkapsel, Artikel-Nr. 069345
- [5] ELV Bausatz Mikrofon MEMS1, Artikel-Nr. 151456
- [6] ELV Bausatz USB-Modul UM2102N, Artikel-Nr. 150952
- [7] Entwicklungskit für Sprachsteuerung vicCONTROL go kit:
https://www.voiceinterconnect.de/de/Entwicklungskit_Sprachsteuerung-vicCONTROL
- [8] 64-Bit-Java-Runtime_Environment: <https://github.com/ojdkbuild/ojdkbuild>
- [9] Kleinlautsprecher 57 mm, 8 Ω, 0,25~1 W: Artikel-Nr. 006235
- [10] Download Dialog „Steuerung einer Lampe“: Artikel-Nr. 251980
- [11] ELVjournal 4/2020: Programmieren (fast) ohne Code – Node-RED als universelles Prototyping-Tool
Teil 1: Artikel-Nr. 251410
- [12] ELVjournal 5/2020: Programmieren (fast) ohne Code – Node-RED als universelles Prototyping-Tool
Teil 2: Artikel-Nr. 251516
- [13] ELVjournal 6/2020: Programmieren (fast) ohne Code – Node-RED: Ein Blick unter die Haube und praktische Anwendungen,
Teil 3: Artikel-Nr. 251603

Mein ELVprojekt

Viele Ideen für Ihr Smart Home

Bei uns erwarten Sie viele spannende, ausführlich beschriebene Projekte für Einsteiger und Profis. Diese Projekte haben wir als Produktmanager und Techniker alle selbst erfolgreich umgesetzt.

Wir zeigen Ihnen z. B., wie Sie für mehr Komfort und Energieeinsparung Rollläden automatisieren, mit einer intelligenten Heizungssteuerung Energiekosten sparen oder Ihr Zuhause vor Einbrechern wirkungsvoll schützen können. Sie erhalten Informationen zum geschätzten Zeitaufwand und zum Schwierigkeitsgrad und alle verwendeten Produkte aus unserem Sortiment werden für Sie übersichtlich aufgeführt. Für viele Projekte gibt es außerdem hilfreiche Installationsvideos.

Setzen Sie nun Ihr Projekt mit ELV erfolgreich um!



Alle Projekte finden Sie online unter:

de.elv.com/elvprojekte
at.elv.com/elvprojekte
ch.elv.com/elvprojekte

