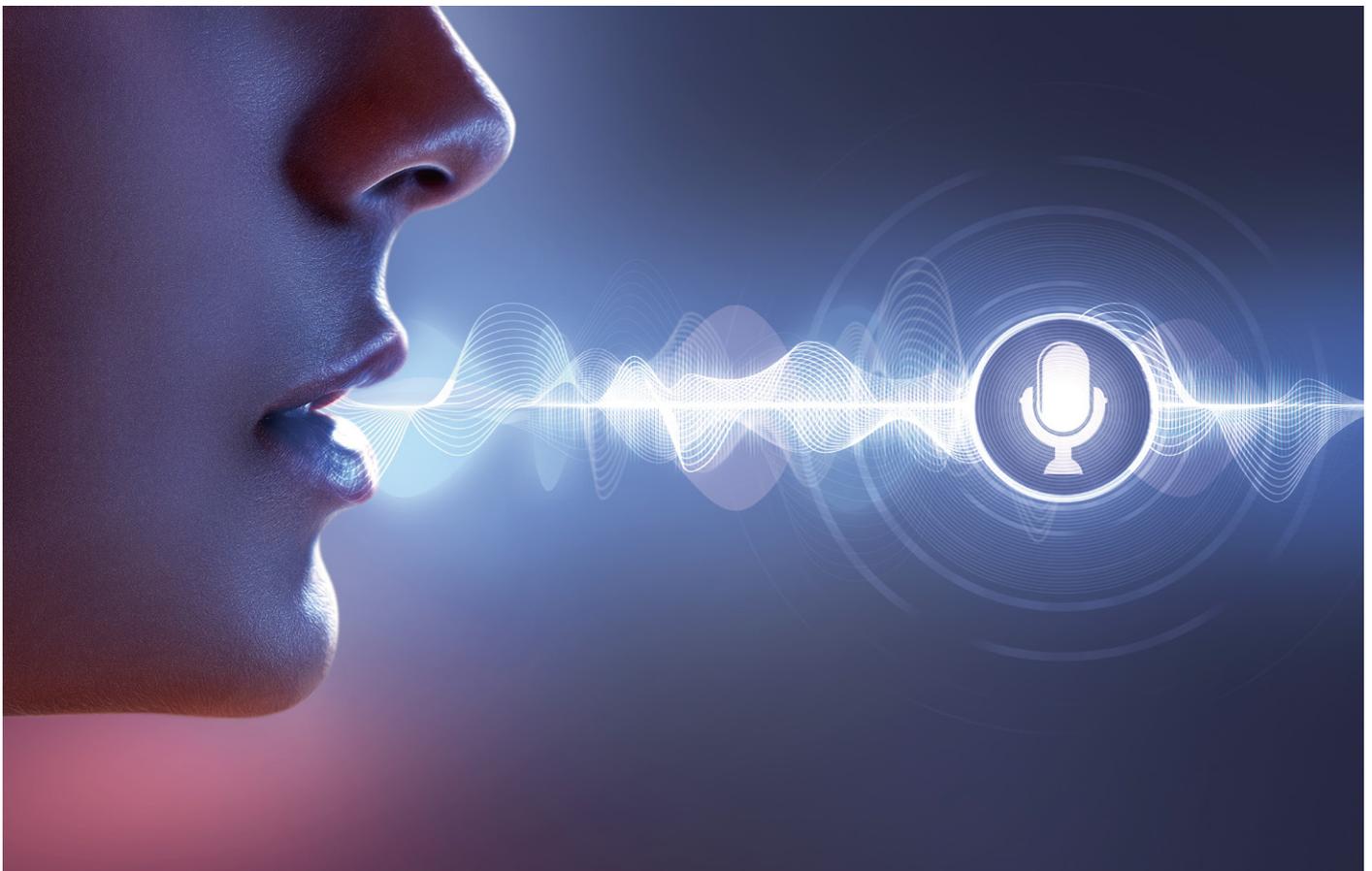


Hallo, Homematic!

Lokale Sprachsteuerung für Homematic Geräte

Inspiziert vom Beitrag „Hallo, Datenschutz!“ im ELVjournal 2/2021 verfolgt das Projekt unseres Lesers Dr. Peter Tschulik aus Wien das Ziel, auf Basis des Entwicklungsboards „vicCONTROL go stamp“ eine kompakte Steuerung aufzubauen, die bis zu 254 unterschiedliche Sprachbefehle versteht und ebenso viele Homematic Aktionen auslösen kann. Sie ist zudem so kompakt, dass man sie in einen Bluetooth-Lautsprecher einbauen kann. Die Bedienung über einige wenige Taster soll dabei alle Möglichkeiten des „vicCONTROL go stamp“ ausschöpfen.



*Siehe Seite 112

Peter Tschulik

hat für seinen Beitrag zum Leserwettbewerb einen Gutscheincode* über 200,- Euro erhalten.

Lokale Sprachsteuerung

Wie im Artikel „Hallo, Datenschutz“ im ELVjournal 2/2021 [1] beschrieben, bietet die Firma „voice INTER connect“ aus Dresden ein Modul zur Sprachsteuerung mit lokaler Signalverarbeitung an, die im Gegensatz zu Siri, Alexa, Home & Co. unabhängig von einer Internetverbindung für eine serverbasierte Sprachanalyse arbeitet. Wie in dem Beitrag ausgeführt, gibt es neben dem reinen Sprachmodul vicCONTROL go stamp (Bild 1), das ohne weitere Peripherie oder Software geliefert wird, auch das Evaluierungsboard vicCONTROL go kit (Bild 2), das neben der Software umfangreiche Peripherie wie Energieversorgung und Kommunikation über USB, ein umfangreiches Audiointerface und verschiedene Anschlussmöglichkeiten für Ein- und Ausgänge bietet.

Für weitere Details wird auf den kostenlos herunterladbaren Beitrag unter [1] verwiesen.

Für den Einsatz in einer umfangreichen Steuerung für Homematic hat das vicCONTROL go kit einige Nachteile:

- Das vicCONTROL go kit ist fast doppelt so teuer wie das vicCONTROL go stamp.
- Das vicCONTROL go kit ist sehr groß und passt deshalb kaum in einen Bluetooth-Lautsprecher.
- Über die Schaltausgänge können nur wenige Schaltbefehle ausgelöst werden. Verwendet man die serielle Schnittstelle, so sind viele Komponenten des Evaluierungsboards überflüssig und man benötigt weitere Hardware, um mit Homematic kommunizieren zu können.

Daraus entstand die folgende Anforderungsliste für das Projekt:

- Kompaktes Design, das in einen Bluetooth-Lautsprecher eingebaut werden kann. Von diesem sollen die Tasten und der Lautsprecher wiederverwendet werden.
- Eine USB-Schnittstelle, über die das Gerät versorgt wird und über die Dialoge eingespielt werden können und ein Upgrade des vicCONTROL go stamp erfolgt.
- Einfache Bedienung über möglichst wenige Tasten und LEDs
- „Over the air“-Upgradebarkeit des Steuerprozessors
- Ausführung von bis zu 254 Homematic Schaltbefehlen
- Einseitige Platinen, die leicht selbst herstellbar sind, und Einsatz von Modulen bzw. bedrahteten Bauteilen für einen einfachen Aufbau

Für den Einsatz in diesem Projekt müssen zunächst einmal das Modul vicCONTROL go stamp und der vicCONTROL go stick mit dem Softwarepaket beschafft werden [2].

Schaltung

Bild 3 zeigt das Blockschaltbild der Schaltung. Die Stromzufuhr und die Kommunikation mit der Außenwelt erfolgen über ein Micro-USB-Breakoutboard (s. Materialliste [3]). Im Block „Stromversorgung“ wird diese 5-Volt-Spannungsversorgung in 3,3 Volt umgewandelt, mit der alle Elemente der Schaltung mit Strom versorgt werden. Mit der Taste A kann die Stromversorgung eingeschaltet werden.

Zentrales Steuerelement ist ein ESP32 Pico Kit, das über die notwendige Anzahl an Ein- und Ausgängen verfügt, leicht erhältlich, gut dokumentiert und mittels Bootloader leicht „Over the air“ upgradebar ist (s. Materialliste [3]).

Über die zwei Steuerleitungen zur Stromversorgung kann sowohl die Taste A abgefragt als auch die Stromversorgung ein- bzw. abgeschaltet werden. Mittels acht Datenleitungen und einer Steuerleitung wird das hochinteressante ELV Modul HM-MOD-EM-8Bit [3] mit Daten versorgt, um einen 8-Bit-Wert über die Homematic Funkschnittstelle zu übertragen. Die Daten vom vicCONTROL go stamp müssen über einen USB-zu-TTL-Converter umgewandelt werden. Da das vicCONTROL go stamp leider nur über einen seriellen Port verfügt, über den einerseits die Kommunikation mit einem Prozessor, ander-



Bild 1: vicCONTROL go stamp

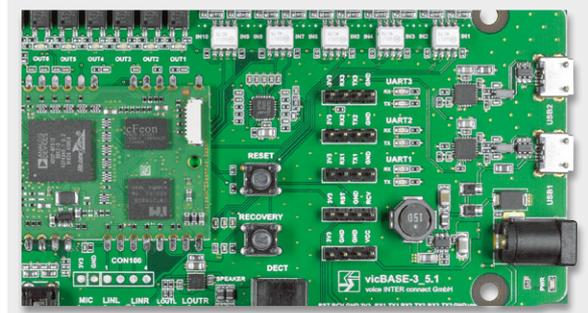


Bild 2: vicCONTROL go kit

erseits auch das Aufspielen der Dialoge und das SW-Update läuft, muss man sich etwas Besonderes einfallen lassen: Im Normalfall steuert der ESP32 einen Multiplexer so an, dass die serielle Schnittstelle mit dem ESP32 verbunden ist, damit dieser auf erkannte Sprachbefehle lauschen kann. Im Modus „Update“ und „Recover“, der später beschrieben wird, wird die serielle Schnittstelle über den USB-zu-TTL-Converter und das Micro-USB Breakoutboard mit einem PC verbunden. Der ESP32 steuert über die Reset-Leitung und die Recover-Leitung das vicCONTROL go stamp. Während das Mikrofon direkt am vicCONTROL go stamp angeschlossen wird, muss der Lautsprecher über einen Verstärker angesteuert werden, der vom ESP32 stummgeschaltet werden kann.

Hauptplatine

Daraus ergibt sich die folgende Detailschaltung, die auf zwei Platinen aufgeteilt ist. Den Schaltplan für die erste Platine zeigt Bild 4. Diese beinhaltet die Stromversorgung sowie das Homematic Interface bestehend aus dem ESP32 Pico Kit und dem ELV Modul HM-MOD-EM-8Bit. Für die Bauteilliste inklusive Bezugsquellen siehe [3].

Noch ein paar Worte zur Stromversorgung. Zum Einsatz kommt ein Step-down-Modul, dessen Ausgang über den Enable-Eingang abgeschaltet werden kann, indem dieser auf GND gelegt wird. Nach dem Anlegen der Versorgungsspannung über den Micro-USB-Anschluss wird der Enable-Eingang über den Transistor T1 mit der Beschaltung R1 und R2 auf GND gelegt. Wird die Taste A gedrückt, so liegt an R3 eine Spannung von 5 V an, die den Transistor T2 durchsteuert und T1 sperrt und damit die Ausgangsspannung von 3,3 V für die Dauer des Tastendrucks freigibt. Damit startet die Ausführung der Software im ESP32 und diese setzt den Ausgang POWER_OFF

als eine der ersten Aktionen auf High, wodurch der Transistor T4 über R6 durchsteuert, damit T1 weiterhin gesperrt bleibt (Selbsterhaltungsschaltung). Über R4 und T3 kann der ESP32 den Zustand der Taste A jederzeit abfragen, da diese sowohl für eine weitere Funktion als auch zum Abschalten der Stromversorgung dient. Soll das Gerät abgeschaltet werden, setzt der ESP32 den Ausgang POWER_OFF auf Low.

Zusatzplatine

Den Schaltplan für die zweite Platine zeigt das Bild 5. Diese beinhaltet den USB-Adapter inklusive Multiplexer und das vicCONTROL go stamp inklusive Verstärkermodul. Die beiden Platinen sind über J5 miteinander verbunden und als Sandwich ausgeführt. Beim USB-zu-TTL-Converter habe ich verschiedenste Module ausprobiert. Wie es scheint, ist das vicCONTROL go stamp besonders beim Update der Software sehr empfindlich und bei anderen Modulen als das in der Bezugsquelle [3] beschriebene gab es immer wieder Probleme und Abbrüche.

Als Multiplexer kommt der CMOS-Schaltkreis 4053 zum Einsatz, der auch mit 3,3V gut funktioniert. Mit J6 kann zwischen einem 3-poligen und einem 2-poligen Mikrofon gewählt werden (siehe [1]). J6 sollte bei dem in der Materialliste [3] verwendeten Mikrofon gesteckt sein. Mit J7 und J8 kann gewählt

werden, ob die WAV-Dateien (Prompts) oder das aktuelle Eingangssignal ausgegeben werden. Für den Normalbetrieb wird J7 gesteckt und J8 bleibt offen. Das Mikrofon wird über eine 3,5-mm-Buchse angeschlossen und der Lautsprecher über Schraubklemmen an J9.

Das Verstärkermodul ist ein kompaktes Class-D-Verstärkermodul, das vom ESP32 über den Eingang SD stummgeschaltet werden kann.

Nachbau und Inbetriebnahme

Für den einfachen Aufbau wurden mit dem Programm „Eagle“ zwei Printplatten erstellt, die unter [3] im Directory „Eagle Files\“ verfügbar sind und beliebig modifiziert werden können. Neben den Eagle-Dateien sind auch der Schaltplan, der Bestückungsplan sowie das Layout als JPEG-Bilder verfügbar.

Die eigens erstellten Symbole sind in der Eagle Library „Sprachsteuerung.lbr“ im Directory „Eagle Files\LIBS“ enthalten. Meine Platinen habe ich auf meiner Fräse hergestellt, als Alternative können diese auch in Eigenregie geätzt oder bei einem Printplattenhersteller in Auftrag gegeben werden. Bild 6 zeigt die beiden unbestückten Platinen.

Zuerst einmal müssen zwölf Drahtbrücken bestückt werden, auf jeder Platine je sechs wie in Bild 7 gezeigt.

Danach werden die Entstörkondensatoren bestückt, die sich unterhalb der Module befinden, das sind C6 auf der Platine mit dem ESP Pico-Kit und C7, C8 und C10 auf der Platine mit dem vicCONTROL go stamp Modul. Diese Kondensatoren werden um 90 Grad umgebogen, um die Höhe zu reduzieren, sodass sie liegend auf der Platine bestückt sind. Danach werden zuerst alle Widerstände, dann die Transistoren und schließlich alle Kondensatoren bestückt, wobei bei den Elektrolytkondensatoren auf die korrekte Polung zu achten ist.

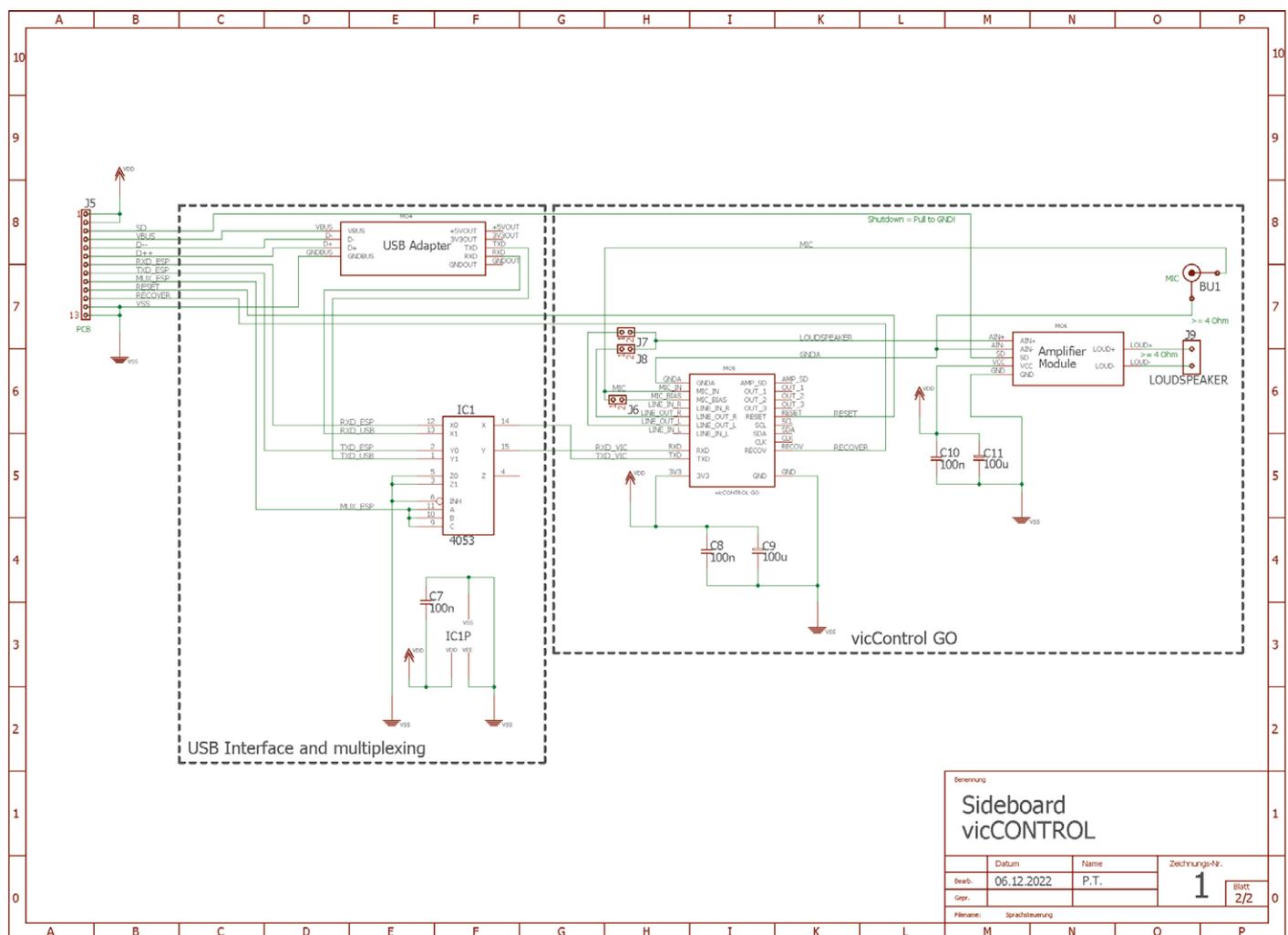


Bild 5: Schaltplan der zweiten Platine

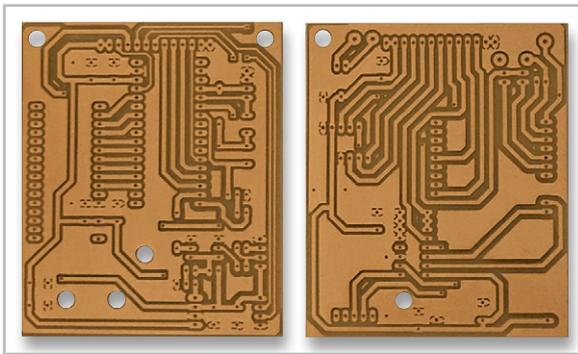


Bild 6: Unbestückte Platinen

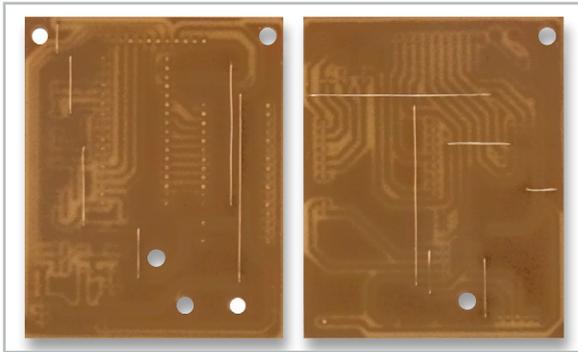


Bild 7: Bestückte Drahtbrücken

Beginnen wir mit der Fertigstellung der Platine mit dem ESP32-Pico-Kit-Modul: Zuerst wird das Power-Supply-Modul nach der beiliegenden Anleitung vorbereitet: Die Leiterbahn für die variable Spannungseinstellung wird durchtrennt und die Lötbrücke für 3,3-Volt-Fixspannung gesetzt. Danach wird das Modul über Stiftleisten auf die Platine gelötet.

Dann erfolgt die Bestückung des ESP32-Pico-Kit-Moduls. Dieses wird ebenfalls über Stiftleisten auf die Platine gelötet, sodass der USB-Anschluss zum Stecker J3 zeigt.

Schließlich wird das ELV Homematic Funk-Sendemodul HM-MOD-EM-8Bit über Stiftleisten angelötet, eine Verpolung ist wegen der Abstände bei den Versorgungspins nicht möglich.

Danach lötet man die Stiftleisten J1, J2 und J4 auf, an die später Kabel für die Taster und die LED angelötet werden. Nun fehlt noch die Bestückung der Stiftleiste J3, die von der Bauteilseite her erfolgt, um diese Platine später mit der Platine mit dem vicCONTROL-go-stamp-Modul aufstecken zu können. Das Bild 8 zeigt die fertig bestückte Platine.

Nun erfolgt die Fertigstellung der Platine mit dem vicCONTROL-go-stamp-Modul: Zuerst erfolgt die Bestückung des Multiplexer IC1. Danach werden in die Platine an den vorgesehenen Pins für das vicCONTROL-go-stamp-Modul Stiftleisten eingesetzt, aber noch nicht verlötet.

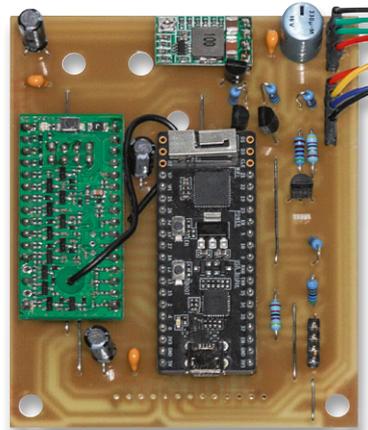


Bild 8: Bestückte Platine mit dem ESP3-Pico-Kit-Modul

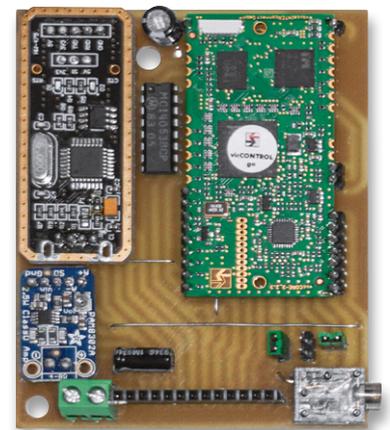


Bild 9: Bestückte Platine mit dem vicCONTROL-go-stamp-Modul

Danach wird das vicCONTROL-go-stamp-Modul vorsichtig aufgesetzt, sodass die Pinleisten in den halbrunden Pins zu liegen kommen. Dann werden die Stiftleisten zuerst mit dem vicCONTROL-go-stamp-Modul verlötet und danach auf der Unterseite der Platine.

Beim USB-zu-TTL-Converter-Modul wird vorsichtig der USB-Stecker entfernt bzw. ausgelötet, ebenso die Jumperleisten für die Spannungseinstellung und die Pinleiste auf der gegenüberliegenden Seite des USB-Steckers. Auf der Unterseite wird dann die Spannung mit einer Lötbrücke auf 3,3 V eingestellt und das Modul über Pinleisten aufgelötet.

Beim Verstärkermodul wird die Schraubklemme für den Lautsprecher ausgelötet und das Modul wieder über Stiftleisten auf die Platine gelötet. Schließlich werden die Jumper J6 bis J8 bestückt, danach die Schraubklemme J9 für den Lautsprecher sowie die 3,5-mm-Klinkenbuchse BU1 für das Mikrofon und die Buchsenleiste J5. Das Bild 9 zeigt die fertig bestückte Platine.

Nach dem Stecken der Jumper J7 und J6 werden die beiden Platinen aufeinandergesteckt und unter Verwendung von Schrauben, Abstandsrollchen und Muttern miteinander verbunden. Für die Inbetriebnahme erfolgt die Verkabelung wie in Bild 10 zu sehen.

Ein Hinweis zum Mikrofon: Als Mikrofon wurde das Olympus ME-52W Rauschunterdrückungs-Mikrofon eingesetzt, das von der Firma voice INTER connect empfohlen wird und relativ leicht erhältlich ist.

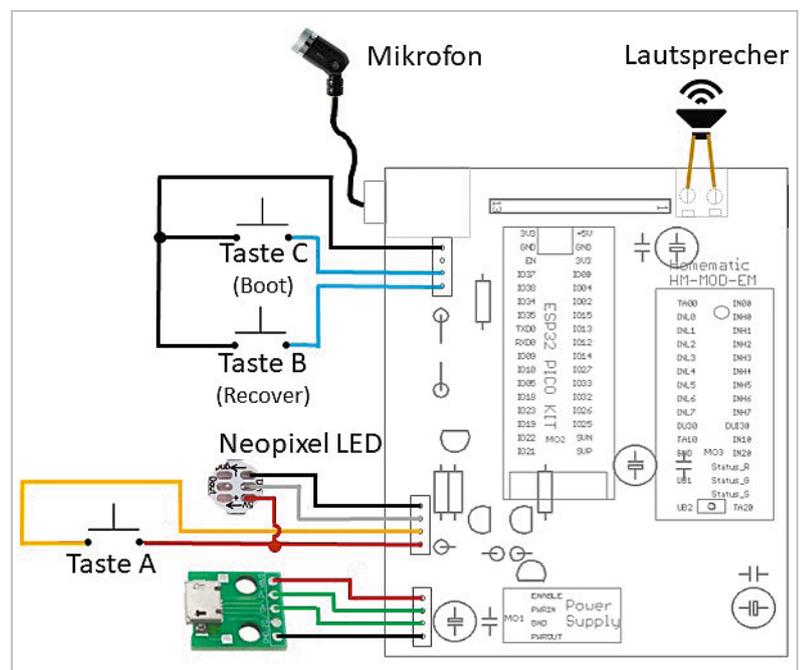


Bild 10: Verkabelung der fertigen Platinen

Aufspielen der Firmware

Nun folgt das Aufspielen der Firmware. Diese wurde in der Arduino IDE (Version 1.8.12) erstellt. Folgende Vorbereitungsarbeiten sind vor dem Übertragen der Software notwendig:

Die Libraries „Fast LED for ESP32“, „Button2“ und „CRC“ müssen installiert werden. Die Links zu den Libraries finden sich in Zeile 45–47 von „Viccontrol.ino“. Der Sourcecode ist unter [3] im Directory „Arduino Source\ Viccontrol“, die Libraries im Directory „Arduino Libraries“ zu finden. Im Programm müssen folgende Zeilen angepasst werden:

- Damit der Bootloader über das WLAN korrekt funktioniert, muss in Zeile 117 eine fixe IP-Adresse eingestellt werden. In Zeile 118 ist die IP-Adresse des Routers einzutragen und in Zeile 119 die Subnet-Maske. In Zeile 120 erfolgt die Angabe der WLAN SSID und in Zeile 121 das WLAN-Passwort.
- In Zeile 90 kann der Befehl „#define _DEBUG_“ durch Entfernung der beiden // am Beginn der Zeile aktiviert werden. Dann werden viele zusätzliche Informationen über den Serial Monitor der Arduino IDE mit 115.200 Baud ausgegeben. Das ist sehr hilfreich, wenn etwas nicht funktioniert oder der Code erweitert oder geändert werden soll.

Nun geht es darum, das Programm zu übertragen. Die Schaltung wird über das Micro-USB-Breakout-Board mit Strom versorgt (nicht über den Micro-USB-Anschluss des ESP32). Die Taste A wird dauerhaft überbrückt, da die Selbsthaltefunktion durch die fehlende Software noch nicht korrekt funktioniert. Danach sollte die rote Power-LED des ESP32 aufleuchten.

Dann wird das ESP32-Pico-Kit-Modul mit dem PC über ein weiteres passendes USB-Kabel verbunden. In der Arduino IDE muss die ESP32-Erweiterung (ich verwende aktuell ESP32 Arduino extension 1.0.4) installiert sein. Als Board wird das ESP-Pico-Kit ausgewählt, danach die korrekte Schnittstelle. Danach wird der Sketch mittels „Sketch ⇒ Upload“ kompiliert und übertragen. Hat dies funktioniert, so kann das USB-Kabel vom ESP32 abgezogen werden und die Brücke von Taster A entfernt werden.

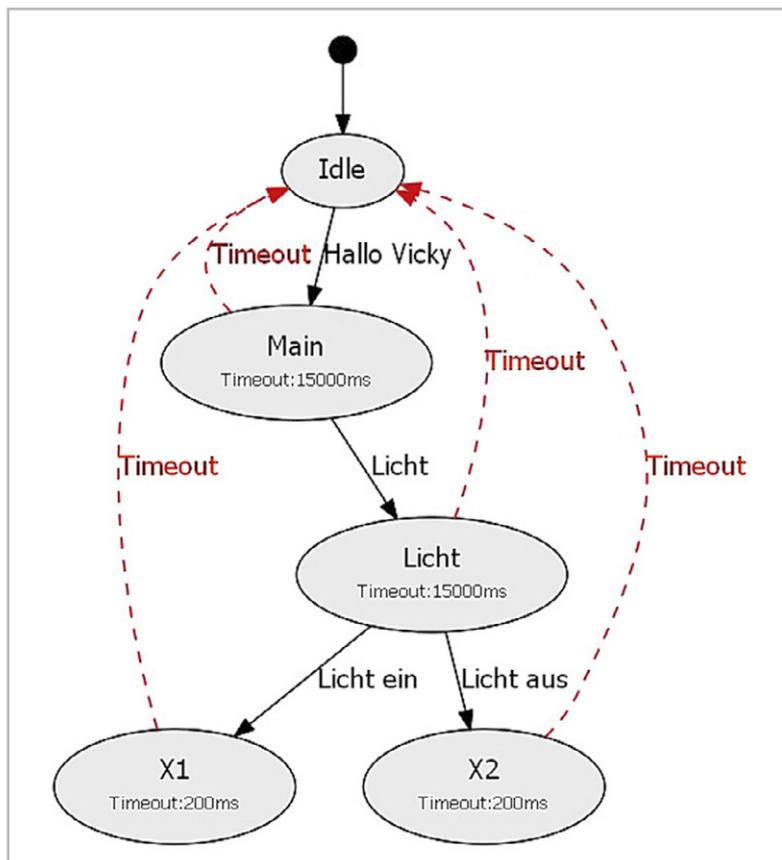


Bild 11: Ablaufdiagramm Beispiel Test1.vdd2

Drückt man Taste A dann für mindestens eine Sekunde, so schaltet sich das Gerät ein. Die LED beginnt mit der Frequenz von einer Sekunde ein paar Mal grün zu blinken. In dieser Zeit wird die serielle Schnittstelle initialisiert, das Homematic Interface und andere Ein- und Ausgänge und die Tasten konfiguriert, das vicCONTROL go stamp zurückgesetzt, auf den Abschluss des Reset-Vorgangs gewartet und die FW-Version, die Plattform und die Sprache des vicCONTROL go stamp abgefragt und über die serielle Schnittstelle ausgegeben.

Ist kein Fehler aufgetreten, so leuchtet die LED nach einigen Sekunden dauerhaft grün.

Vorbereitung des vicCONTROL go stamp

Nun muss das Beispiel „Test1.vdd2“ unter [3] im Directory „vic Control Beispiele“ auf das vicCONTROL go stamp übertragen werden. Dazu muss die Taste B für ca. eine Sekunde gedrückt werden. Die LED wechselt auf die blaue Farbe. In diesem Modus ist nun das vicCONTROL go stamp direkt über das USB-zu-TTL-Converter-Modul mit dem Micro-USB-Breakout-Board verbunden. Wird in der vicCONTROL-Designer-Software das File „Test1.vdd2“ geladen, so zeigt diese den Dialog wie in Bild 11 an.

Im Idle-State wartet das vicCONTROL go stamp auf das Triggerwort „Hallo, Vicky“ und wechselt nach der korrekten Erkennung in den State „Main“. Wird vor dem Timeout das Wort „Licht“ gesprochen, so wird in den State „Licht“ gewechselt. Wird vor einem neuerlichen Timeout das Wort „Ein“ oder „Aus“ erkannt, so wird in den State X1 bzw. X2 gewechselt und nach einer kurzen Pause zu „Idle“ gesprungen. Die Bilder 12 bis 15 zeigen die dazugehörigen anderen Einstellungen.

Für eine korrekte Funktion ist es wichtig, dass im Menu „System Settings ⇒ Serial output configuration“ bei den Properties unter Misc nur „Signal Menu Changes“ auf „true“ gesetzt ist, wie Bild 16 zeigt.

Für weitere Informationen zur Erstellung von Dialogen sei auf [1] und auf die Dokumentation verwiesen, die von der Webseite der Firma voice INTER heruntergeladen werden kann.

Wichtig ist, das folgende Funktionsprinzip der Software zur Auswertung der empfangenen Befehle zu verstehen. Die Software im ESP32 ignoriert alle Befehle, die nicht mit „X“ beginnen. Deshalb sollte auch kein anderes Menu mit X beginnen! Danach versucht der ESP32 die Ziffer nach dem X zu extrahieren, beispielsweise aus X1 die 1. Diese wird dann über das Homematic Interface übertragen. Dies stellt sicher, dass die Software im ESP32 nicht ständig angepasst werden muss, wenn neue oder geänderte Dialoge eingespielt werden.

Das Einspielen des Beispiels erfolgt im „Lademodus“ (Taste B ca. eine Sekunde gedrückt halten und die LED leuchtet blau) wie folgt: Unter „COM“ wählt man die serielle Schnittstelle aus und drückt dann „Connect“. War dies erfolgreich, drückt man auf „Upload“, um den Dialog zu übertragen. Um den vollen Umfang der Software ausnutzen zu können, werden die Prompts wie in der Tabelle 1 angegeben verwendet.

Triggerclass	Content	Class-ID
Hallo Vicky	Hallo Vicky, Hallo Wikki, Hallo Wiggi, Hallo...	12
Licht	Licht	2
Licht ein	Ein	6
Licht aus	Aus	7

Bild 12: Commands von Beispiel Test1.vdd2

Menu	Timeout (ms)	Score High	Score Low	Use wakeup word
Idle	0	4200	4200	false
Main	15000	4000	4000	false
Licht	15000	4000	4000	false
X1	200	4000	4000	false
X2	200	4000	4000	false

Bild 13: Menüs von Beispiel Test1.vdd2

Old Menu	Triggerclass	New Menu	Comment
Idle	Hallo Vicky	Main	
Main	Licht	Licht	
Licht	Licht ein	X1	
Licht	Licht aus	X2	

Bild 14: Transitions von Beispiel Test1.vdd2

Old Menu	New Menu	Comment
Main	Idle	
Licht	Idle	
X2	Idle	
X1	Idle	

Bild 15: Timeout-Transitions von Beispiel Test1.vdd2

Test1.vdd2

System Settings

- Audio settings
- Digital inputs to event mapping
- Serial output configuration
- Wakeup
- Action on wakeup
- Action on sleep
- Action if speech recognition confidence between high and low score

Properties

Property	Value
Connection	
Baudrate	115200
Data Bits	8
Parity	None
Stop Bits	1
Misc	
Signal Menu Changes	true
Signal Trigger IDs	false
Signal Trigger Text	false
Signal Volume	false

Bild 16: System Settings von Beispiel Test1.vdd2

Diese acht Beispiel-Prompts sind unter [3] im Directory „vicCONTROL Beispiele\Prompts“ verfügbar und können mit der von der Firma voice INTER connect bereitgestellten einfachen Aufnahme-Software Wavesurfer, die sich im Directory „Software“ befindet, selbst aufgenommen werden. Natürlich kann auch der Inhalt der Aufnahmen individuell festgelegt werden.

Die Prompts müssen individuell eingespielt werden. Da Prompt 1 und Prompt 2 bereits im vicCONTROL go stamp vorhanden sind, betrifft dies nur die Prompts 3–8.

Im vicControl-Designer wählt man unter Prompts zuerst die wav-Datei mit „Choose File“ aus, danach wählt man unter „for Slot“ den Prompt aus, den man aktualisieren will, und drückt dann auf „Upload“. Das wiederholt man für die Prompts 3–8.

Prompts und deren Verwendung in der ESP32-Software		
Prompt-Nr.	Filename	Verwendungszweck
1	p1.wav	Standard „Hoher Ton“ (bereits auf der vicCONTROL go stamp aufgespielt): Wird bei jedem Menüübergang außer bei der Erkennung des Triggerwortes abgespielt.
2	p2.wav	Standard „Tiefer Ton“ (bereits auf der vicCONTROL go stamp aufgespielt): Wird bei jedem Timeout bzw. Übergang in den „Idle“-State abgespielt.
3	p3.wav	„Ja, Meister“: Wird nach der Erkennung des Triggerwortes beim Übergang zum „Main“-State abgespielt.
4	p4.wav	„Bereit“: Wird abgespielt, wenn das Gerät bereit ist, Sprachbefehle entgegenzunehmen (LED leuchtet grün).
5	p5.wav	„Stand-by“: Wird abgespielt, wenn die Spracherkennung deaktiviert ist (LED leuchtet orange).
6	p6.wav	„Lernmodus“: Wird abgespielt, wenn der Lernmodus aktiviert ist (LED leuchtet blau).
7	p7.wav	„Recover-Modus“: Wird abgespielt, wenn der Recover-Modus aktiviert ist, z. B. zum Update einer neuen Software (LED leuchtet violett).
8	p8.wav	„Bootloader“: Wird abgespielt, wenn der ESP32-Bootloader-Modus aktiviert ist (LED blinkt langsam cyan).

Tabelle 1

Name	Kanal	Modul	Modul-ID	Modul-Name	Modul-Typ	Modul-Standard	Modul-Funktion	Modul-Status	Modul-Optionen	Modul-Buttons	
Sprachmodulsteuerung	HM-MOD-FM-8Bit		Funk-Sendemodul, 8-Bit	NFQ1547318	RidCos-RF	Standard	Steuerung	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Einstellen Löschen Direkte Programme
Sprachmodulsteuerung:1 Taster	HM-MOD-EM-8Bit		Funk-Sendemodul, 8-Bit	NEQ1547318:1	Sender	Standard	Steuerung	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Einstellen Direkte Programme
Sprachmodulsteuerung:2 Taster	HM-MOD-FM-8Bit		Funk-Sendemodul, 8-Bit	NFQ1547318:2	Sender	Standard	Steuerung	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Einstellen Direkte Programme
Sprachmodulsteuerung:3 Sender für 8-Bit Entscheidungswert	HM-MOD-EM-8Bit		Funk-Sendemodul, 8-Bit	NEQ1547318:3	Sender	Standard		<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Einstellen Direkte Programme

Bild 17: Vergabe des Namens für das Gerät HM-MOD-EM-8Bit in Homematic

Name	Kanal	Parameter
Sprachmodulsteuerung:3 Sender für 8-Bit Entscheidungswert	Ch.: 3	Max. Sendeveruche <input type="text" value="3"/> (1 - 10) Datenübertragungsbedingung <input type="text" value="Modus 1"/> Datenstabilitätsfilterzeit vor der Sendung <input type="text" value="0,0"/> s (0,0 - 111600,0) Dateneingang invertieren: Eingang 0 <input type="checkbox"/> Eingang 1 <input type="checkbox"/> Eingang 2 <input type="checkbox"/> Eingang 3 <input type="checkbox"/> Eingang 4 <input type="checkbox"/> Eingang 5 <input type="checkbox"/> Eingang 6 <input type="checkbox"/> Eingang 7 <input type="checkbox"/>

Bild 18: Konfiguration von Kanal 3

Funktionstest

Nun wird es spannend, da der erste Funktionstest erfolgt: Man trennt das Gerät von der Stromversorgung, wartet ein paar Sekunden und versorgt es danach über das Micro-USB-Breakout-Board mit Strom. Danach drückt man die Taste A länger als eine Sekunde und die grüne LED blinkt ein paar Mal.

Nun sollte die grüne LED permanent leuchten und aus dem Lautsprecher sollte die Nachricht „Bereit“ ertönen. Sagt man das Schlüsselwort „Hallo, Vicky“, so antwortet das Gerät mit „Ja, Meister“ (Prompt 3) und die LED blinkt kurz auf. Sagt man danach „Licht“, so ertönt ein kurzer höherer Ton (Prompt 1) und die LED blinkt wieder. Sagt man danach „Ein“, so ertönt nochmals ein kurzer höherer Ton (Prompt 1) und nach kurzer Zeit der tiefere Ton (Prompt 2) und die LED blinkt kurz.

Was jetzt noch fehlt, ist die Verknüpfung mit einem Homematic Aktuator, die wie folgt beschrieben wird.

Verbindung mit Homematic

Zuerst einmal wird das ELV HM-MOD-EM-8Bit Modul an die Smart Home Zentrale CCU3 – wie in der Bedienungsanleitung beschrieben – angelehrt und ein Name wie beispielsweise „Sprachmodulsteuerung“ vergeben (Bild 17). Kanal 3 muss wie in Bild 18 zu sehen konfiguriert sein.

Danach wird ein neues Programm unter „Programme und Verknüpfungen → Programme & Zentralenverknüpfung“ mit dem Namen „Sprachmodulsteuerung“ wie in Bild 19 zu sehen angelegt.

Zuerst wird der Kanal 3 des Geräts „Sprachmodulsteuerung“ ausgewählt, danach im Dropdown-Menü „Wert des Dateneingangs“ eingestellt, danach im Wertebereich „Größer oder gleich 0“ festgelegt und im letzten Dropdown-Menü „bei Aktualisierung auslösen“ ausgewählt. Unter Aktivität „Skript auswählen“ wird das folgende Skript angelegt (siehe auch [3]):

Name	Beschreibung	Bedi
Sprachmodulsteuerung	Auswertung der Nachrichten der Sprachmodulsteuerung	Kanalzustand: Sprachmodulsteuerung:3 bei Wert des Datene
Bedingung: Wenn... Geräteauswahl <input type="text" value="Sprachmodulsteuerung:3"/> bei Wert des Dateneingangs <input type="text" value="im Wertebereich größer oder gleich 0"/> bei Aktualisierung auslösen <input type="text" value=""/>		
<input type="checkbox"/> UND <input type="checkbox"/>		
<input type="checkbox"/> ODER <input type="checkbox"/>		
Aktivität: Dann... <input checked="" type="checkbox"/> Vor dem Ausführen alle laufenden Verzögerungen für diese Aktivitäten beenden (z.B. Retriggern). Skript <input type="text" value="! Skript to receive data from the speech module ! Protokoll ..."/> sofort <input type="text" value=""/>		
<input type="checkbox"/> Vor dem Ausführen alle laufenden Verzögerungen für diese Aktivitäten beenden (z.B. Retriggern).		

Bild 19: Anlegen eines Programms

```

! Skript to receive data from the speech module
! Protokoll D7 D6 D5 D4 D3 D2 D1 D0
!           0  0  0  0  0  0  0  1 --> Licht Arbeitszimmer ein
!
!           0  0  0  0  0  0  1  0 --> Licht Arbeitszimmer aus
!
!Name of Hm module
var devName = „Sprachmodulsteuerung:3“;
!Name of actuator
var devActuator= „BidCos-RF.JEQ0295095:1.STATE“;
! If exist Devicename then devExist contains the name of the device else it is NULL
var devExist=dom.GetObject(devName);
! If exist Actuatorname then devActuator contains the name of the actuator else it is NULL
var actExist=dom.GetObject(devActuator);
! Check if device and actuator exist?
if (devExist && actExist)
{
  ! Read state of channel 3
  var readValue=devExist.State();
  WriteLine(„Sprachmodulsteuerung Datenwert: „ # readValue );
  if (readValue == 1)
  {
    dom.GetObject(devActuator).State(true);
    WriteLine(„Licht eingeschaltet!“ )
  }
  if (readValue == 2)
  {
    dom.GetObject(devActuator).State(false);
    WriteLine(„Licht Arbeitszimmer ausgeschaltet!“ );
  }
}
}

```

Noch ein Hinweis zu der Zeile

var devActuator= „BidCos-RF.JEQ0295095:1.STATE“;
 JEQ0295095 ist die Seriennummer des Aktuators, und bei Geräten von Homematic IP muss „BidCos-RF“ durch „HmIP-RF“ ersetzt werden.

Nachdem die Inbetriebnahme erfolgreich durchgeführt ist, können das Gerät – wie beispielsweise in Bild 20 gezeigt – in einen ausgeschlachteten Bluetooth-Lautsprecher eingebaut werden.

Bedienung

Die Bedienung des Geräts erfolgt ausschließlich über die drei Bedientasten, wobei zwischen einem kurzen Tastendruck (Betätigungszeit liegt zwischen 100 ms und 2 s) und einem langen Tastendruck (Betätigungszeit länger als 2 s) unterschieden wird. Die verschiedenen Funktionen der drei Bedientasten sind in Tabelle 2 zu sehen.

Funktion der drei Bedientasten		
Taste	Betätigung	Aktion
Taste A	Kurzer Tastendruck	<u>Gerät ist eingeschaltet</u> : Wechselt zwischen dem Modus „Spracherkennung ein“ (LED leuchtet grün, Prompt p4.wav wird abgespielt) und dem Modus „Spracherkennung aus“ (LED leuchtet orange, Prompt p5.wav wird abgespielt).
Taste A	Langer Tastendruck	<u>Gerät ist ausgeschaltet</u> : Schaltet das Gerät ein <u>Gerät ist eingeschaltet</u> : Schaltet das Gerät aus
Taste B	Kurzer Tastendruck	<u>Gerät ist im Modus „Spracherkennung ein“ oder „Spracherkennung aus“</u> : Wechselt in den „Lernmodus“ (LED leuchtet blau, Prompt p6.wav wird abgespielt). <u>Gerät ist im „Lernmodus“</u> : Wechselt in den vorher eingestellten Modus „Spracherkennung ein“ (LED leuchtet grün, Prompt p4.wav wird abgespielt) oder den Modus „Spracherkennung aus“ (LED leuchtet orange, Prompt p5.wav wird abgespielt).
Taste B	Langer Tastendruck	<u>Gerät ist in Modus „Spracherkennung ein“ oder „Spracherkennung aus“</u> : Wechselt in den „Recover-Modus“ (LED blinkt zuerst violett und leuchtet dann permanent violett, Prompt p7.wav wird abgespielt). <u>Gerät ist im „Recover-Modus“</u> : Wechselt in den Modus „Spracherkennung ein“ (LED blinkt zuerst violett und leuchtet dann permanent grün, Prompt p4.wav wird abgespielt).
Taste C	Kurzer Tastendruck	Die vicCONTROL go stamp wird zurückgesetzt. Während des Resets blinkt die LED langsam rot. Nach dem Reset wechselt das Gerät in den Modus „Spracherkennung ein“ (LED leuchtet grün, Prompt p4.wav wird abgespielt).
Taste C	Langer Tastendruck	Der ESP32 wird in den Modus „Bootloader“ versetzt. (LED blinkt cyan, wenn eine Verbindung mit dem WLAN erfolgreich aufgebaut werden konnte, Prompt p8.wav wird abgespielt.)

Tabelle 2



Bild 20: Die Platinen der fertiggestellten Sprachsteuerung können beispielsweise in einem ausgeschlachteten Bluetooth-Lautsprecher eingebaut werden.

Im Lernmodus können Dialoge oder Prompts über die vicCONTROL-Designer-Software aufgespielt werden. Im Recover-Modus können Sprachpakete und die Firmware des vicCONTROL go stamp aktualisiert werden. Nachdem der Modus aktiviert ist (LED leuchtet violett), wird das Programm „vicCONTROL_update2.2.0.exe“ gestartet, das von der Firma voice INTER connect im Directory „Software“ auf dem USB-Stick bereitgestellt wird. Die korrekte Schnittstelle wird ausgewählt, bevor man sich mit „Connect“ verbindet. Danach muss noch das Updatepaket (Sprachpaket oder Sprachpaket inklusive FW-Update) ausgewählt werden und mit „Start“ ausgeführt werden.

Bitte davor unbedingt die Dokumentation der Firma voice INTER connect herunterladen und aufmerksam lesen!

Der Bootloader-Modus ist ein sehr praktisches Feature der ESP32-Familie, bei der ein Software-Update über ein WLAN durchgeführt werden kann, als ob das Gerät an einem USB-Port angeschlossen wäre. Das Gerät muss daher für ein Software-Update nicht immer aufgeschraubt werden. Wird der Bootloader-Modus gewählt, versucht der ESP32 sich mit dem WLAN zu verbinden.

Ist die Verbindung nicht erfolgreich, wird dies durch ein rasches rotes Blinken der LED signalisiert. Ist die Verbindung mit dem WLAN erfolgreich, blinkt die LED langsam in Cyan. In der Arduino IDE sollte unter „Port“ ein Anschluss mit der IP-Adresse des Geräts angezeigt werden. Wählt man diesen aus, so kann die Software über den Bootloader übertragen werden.

Hinweis: Nach einem erfolgreichen ESP32-SW-Update startet das Gerät neu, ein Unterbrechen des Modus „Bootloader“ kann nur erfolgen, indem die Spannung des Geräts unterbrochen wird!

In [Tabelle 3](#) sind die Bedeutung der Farben und Blinkmuster der LED aufgeführt.

Beschreibung der Software

Das umfangreiche Arduino-Programm wurde in der Arduino IDE 1.8.12 erstellt, ist ausführlich dokumentiert und wird hier nur kurz beschrieben: Es besteht aus dem Hauptprogramm „Viccontrol“ und den Reitern „Basis“, „LED“, „Stamp“ und „Wifi“. „Viccontrol“ besteht aus der bei Arduino üblichen Setup- und Loop-Routine.

In der Setup-Routine wird zuerst einmal die Selbsthaltefunktion aktiviert, die serielle Schnittstelle für das Debug-Interface und das vicCONTROL go stamp initialisiert. Danach werden die Ein- und Ausgänge für den Anschluss des ELV HM-MOD-EM-8Bit Moduls und anderer Steuersignale konfiguriert und das vicCONTROL go stamp zurückgesetzt, um danach auf das Ende des Bootvorgangs zu warten.

War der Bootvorgang erfolgreich, so werden die Firmware-Version, die Plattform und die Sprache des vicCONTROL go stamp abgefragt und über die Debugging-Schnittstelle ausgegeben. In der Haupt-routine wird der Empfang von Nachrichten des vicCONTROL go stamp geprüft und mögliche Tastenbetätigungen der drei Taster werden ausgewertet.

Unter „Basis“ sind Basisroutinen zusammengefasst:

- Die Subroutine „SUB_LONG_PRESS“ wertet kurze und lange Tastenbetätigungen der drei Taster aus.
- Die Subroutine „SUB_TOGGLE_MODE“ wechselt zwischen dem Modus „Spracherkennung ein“ und dem Modus „Spracherkennung aus“ hin und her.
- „SUB_TOGGLE_LOAD“ bzw. „SUB_TOGGLE_RECOVER“ aktiviert bzw. deaktiviert den „Lernmodus“ bzw. „Recover-Modus“.
- Die Subroutine „SUB_RESET_VIC“ setzt das vicCONTROL go stamp zurück.
- „SUB_ESP_BOOTLOADER“ startet den ESP32-Bootloader, nachdem eine Verbindung mit dem WLAN versucht wird.

Bedeutung der LED-Anzeige

Farbe	Art	Bedeutung
Grün	Langsames Blinken	Gerät startet nach dem Einschalten
Grün	Dauerhaft	Modus „Spracherkennung ein“
Grün	Kurzes einmaliges Blinken	Ein Sprachbefehl wurde erkannt
Orange	Dauerhaft	Modus „Spracherkennung aus“
Blau	Dauerhaft	Lernmodus
Violett	Langsames Blinken	„Recover-Modus“ wird entweder gestartet oder beendet
Violett	Dauerhaft	„Recover-Modus“
Cyan	Langsames Blinken	Modus „Bootloader“
Rot	Langsames Blinken	Das vicCONTROL go stamp wird gerade zurückgesetzt
Rot	Schnelles Blinken	Ein Fehler in der Kommunikation mit dem vicCONTROL go stamp wurde erkannt. Dies kann nur durch eine Spannungsunterbrechung behoben werden.

- Die Ansteuerung des HM-MOD-EM-8Bit erfolgt in der Subroutine „SUB_TRANS_HOMEMATIC“.
 - In der Subroutine „SUB_ERROR_ROUTINE“ werden Kommunikationsfehler mit dem vicCONTROL go stamp durch ein endloses rasches rotes Blinken der LED behandelt. Unter „LED“ sind Routinen zur Ansteuerung der Neopixel-LEDs zusammengefasst. Die ersten beiden Routinen sind die LED-Basisroutinen für die Ansteuerung der LED.
 - Über die Subroutine „SUB_NEO_SEND_COLOR“ wird die Farbe der LED eingestellt.
 - Unter „Stamp“ sind Routinen für die Kommunikation mit dem vicCONTROL go stamp zusammengefasst.
 - „SUB_RECEIVE_ANSWER_VIC“ wartet auf eine Antwort vom vicCONTROL go stamp, nachdem ein Befehl an dieses gesendet wurde.
 - „SUB_RECEIVE_MESSAGE_VIC“ empfängt eine Nachricht vom vicCONTROL go stamp, beispielsweise wenn ein Sprachbefehl erkannt wird. In der Subroutine „SUB_STRING_TO_CRC“ extrahiert CRC die Checksumme aus dem Empfangsstring.
 - Unter „Wifi“ befindet sich die Subroutine „SUB_WLAN_CONNECT“, die vor der Aktivierung des Bootloaders eine Verbindung mit dem WLAN versucht.
- Viel Spaß mit der Sprachsteuerung für Homematic, mit der Sie auf jeden Fall Bewunderung bei Ihren Verwandten und Bekannten ernten werden! **ELV**

i Weitere Infos

- [1] Hallo Datenschutz! Lokale Sprachsteuerung selbst gebaut, ELVjournal 2/2021: Artikel-Nr. 251980
- [2] Evaluierungsboard vicCONTROL go kit: Artikel-Nr. 251951
 Basismodul vicCONTROL go stamp: Artikel-Nr. 251952
 Software vicCONTROL go stick: Artikel-Nr. 251953
- [3] Downloads zum Beitrag: Artikel-Nr. 253021
- [4] Bausatz Homematic Funk-Sendemodul 8-Bit HM-MOD-EM-8Bit: Artikel-Nr. 150253

Alle Links finden Sie auch online unter: de.elv.com/elvjournal-links

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 Ihre 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
ch.elv.com/elvprojekte

