Een open-source spraakherkenningsserver...

...en de ESP BOX

Kristian Kielhofner (Verenigde Staten)

Velen van ons zijn dol op Alexa Echo en soortgelijke apparaten, maar er zijn altijd zorgen geweest over privacy en datagebruik. Het Willow Platform is een gratis open-source alternatief. Naast de Willow Inference-server is er een hoogwaardige spraakgestuurde gebruikersinterface nodig die de audio opneemt en oppoetst. De ESP BOX van Espressif heeft niet alleen microfoons en audio processing voor een aantrekkelijke prijs, maar ook een krachtig software-ecosysteem.

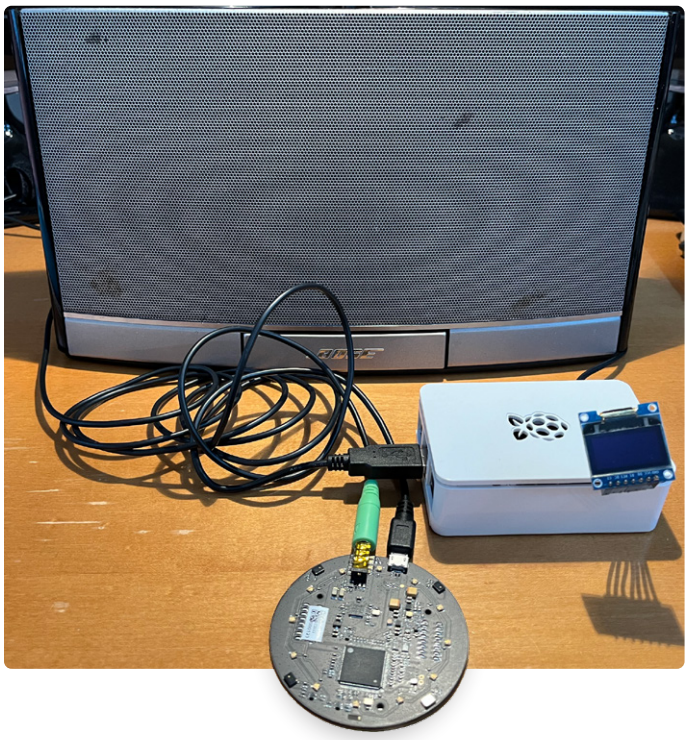
Sinds de introductie van het Alexa-platform acht jaar geleden heeft Amazon meer dan 500 miljoen Echo-apparaten verkocht. Er bleven echter altijd bezorgdheid en controverses met betrekking tot privacy, gegevensgebruik en de alsmaar toenemende pogingen van Amazon om nog meer geld te verdienen aan Echo-gebruikers. Willow [1] is een alternatief platform dat een gratis, open source en maker-vriendelijke Alexa-spraakinterface biedt zonder dat dit ten koste gaat van de kwaliteit of de portemonnee.

Hardware

Raspberry Pi

Een spraakinterface van hoge kwaliteit moet bestaan in en communiceren met de fysieke wereld. Jarenlang heeft het open-source ecosysteem spraakinterfaces geprobeerd op basis van de Raspberry Pi, verschillende microfoons enzovoort (**figuur 1**). Deze aanpak gaat gepaard met enkele grote uitdagingen:

- **Prijs.** Tegen de tijd dat je een Raspberry Pi, aanraak-LCD, hoogwaardig microfoonarray, luidspreker, geschikte behuizing enzovoort bij elkaar hebt, kijk je al tegen een prijs aan die minstens driemaal zo hoog is als die van een Echo-apparaat ($ 50 of minder). Bij deze aanpak moet je componenten



Figuur 1. Eerdere pogingen van de auteur met de Raspberry Pi.

zoeken, alles zorgvuldig opbouwen, een geschikte behuizing maken, de software ontwikkelen enzovoort. Doe dit voor meerdere apparaten in je omgeving en de benodigde tijd en kosten lopen aanzienlijk op.

- **Verkrijgbaarheid.** Het is de laatste tijd beter geworden, maar alleen al met de Raspberry Pi waren er grote problemen met de leverbaarheid van deze componenten. De afgelopen jaren varieerde de verkrijgbaarheid van onmogelijk tot extreem duur vanwege secundaire doorverkoop.
- **Beheer.** Oplossingen op basis van de Raspberry Pi vereisen vaak een volledige Linux-distributie inclusief ondersteuning voor hardware-componenten, alsmede een verbijsterende hoeveelheid software die nodig is voor een spraak-UI. Het beheren van een half dozijn Linux-computers kan voor veel gebruikers een uitdaging blijken.
- **Kwaliteit.** Amazon (en anderen) hebben veel geïnvesteerd in het ontwerp van een spraakinterface die goed functioneert in akoestisch uitdagende omgevingen met een grote verscheidenheid aan menselijke sprekers. Het is meer dan alleen maar een microfoon en luidspreker op een Pi aansluiten.
- **Ecosysteem.** Als je eenmaal een nauwkeurig transcript van een spraakcommando hebt, moet je er ook iets mee doen en de gebruiker feedback geven.
- **Prestaties.** Met de meest optimale spraakherkennings-toepassingen kan een Raspberry Pi 4 bij benadering real-time spraakherkenning uitvoeren met het laagste kwaliteitsmodel. De nauwkeurigheid laat veel te wensen over en 'real time' is gewoon niet snel genoeg – vooral als je bedenkt dat je bij een fout in de transcriptie het commando moet herhalen, waardoor het gemak van een spraakinterface wegvalt.

In **tabel 1** kun je zien dat een Raspberry Pi iets sneller is dan realtime spraakherkenning bij gebruik van het model met de geringste kwaliteit.

Interessant is dat spraakherkenningsmodellen betere realtime-factoren hebben met langere spraakfragmenten. Helaas zijn de gesproken commando's voor spraakassistenten erg kort, waardoor de prestaties veel slechter worden. Deze benchmark is eigenlijk in het voordeel van de real-time spraakherkenningsprestaties van de Raspberry Pi.

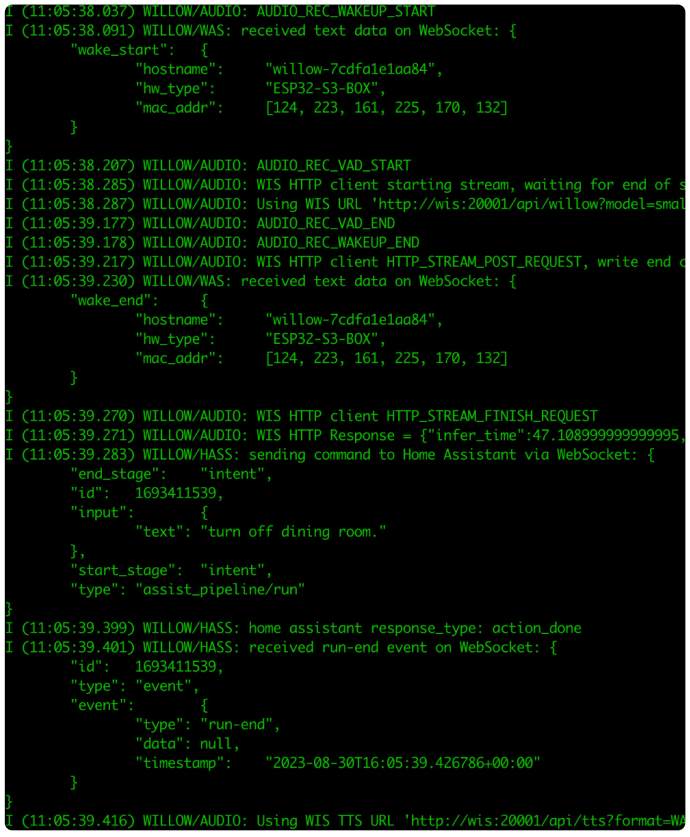
We kennen de Raspberry Pi allemaal en houden ervan – het is een geweldig platform voor een breed scala aan toepassingen en gebruiksscenario's. Helaas hoort een assistent met spraakherkenning en -synthese daar niet bij.

Zoals het screenshot van **figuur 2** laat zien, duurde de spraakherkenning van Willow met de Willow Inference Server van het einde van de spraak naar de bevestiging van de door het Home Assistant-domoticasysteem voltooide actie 222 milliseconden. Dit is ruwweg 25% van de tijd die de Raspberry Pi 4 nodig heeft om alleen de spraakherkenning uit te voeren – en dat bij gebruik van een spraakherkenningsmodel dat aanzienlijk nauwkeuriger is dan het eenvoudige model dat nauwelijks realtime is op de Raspberry Pi.

Net als vele anderen heb ik in de loop der jaren een paar pogingen gedaan om zelf een spraakinterface te bouwen, met verschillende benaderingen. Helaas was het resultaat altijd hetzelfde – veel werk en kosten die een interessante demo kunnen opleveren, maar die in de echte wereld volledig onpraktisch en onbruikbaar zijn vanwege de hierboven genoemde (en andere) problemen.

Table 1: Raspberry Pi 4 speech recognition performance.

Model	Beam Size	Spraakduur (ms)	Inferencetijd (ms)	Realtime-factor
tiny	1	3.840	3.333	1,15×
base	1	3.840	6.207	0,62×
medium	1	3.840	50.807	0,08×
large-v2	1	3.840	91.036	0,04×



Figuur 2. De prestaties van Willow-spraakherkenning.

ESP BOX

Toen ontdekte ik het ESP BOX-ontwikkelplatform van Espressif (**figuur 3**). Zoals zovelen gebruik ik al een jaar of tien Espressif-onderdelen voor verschillende toepassingen en ik weet hoe robuust, veelzijdig, kosteneffectief en feitelijk beschikbaar de hardware is.

Figuur 3. ESP32-S3-BOX-3 wordt geleverd met een 2,4-inch display, twee microfoons en een luidspreker.

Ik weet ook van eerdere projecten dat Espressif veel ondersteunende software en documentatie van hoge kwaliteit levert bij hun hard- en software.
De ESP BOX van Espressif is een ontwikkelplatform dat speciaal voor dit soort toepassingen is gemaakt. Voor ongeveer $ 50 bij je favoriete DHZ-elektronicawinkel of -distributeur krijg je het volgende:

- een ESP32-S3 met 16 MB flash en 16 MB high speed RAM (via SPI);
- een 2,4"-display met capacitief aanraakscherm;
- twee microfoons (erg belangrijk – hierover later meer);
- luidspreker voor audio-uitvoer;
- hardware mute-knop;
- veel uitbreidingsmogelijkheden met de nieuwe ESP32-S3-BOX-3 printen en componenten.

En dat alles is klaar voor gebruik in een esthetisch aantrekkelijke, akoestisch geoptimaliseerde behuizing. Theoretisch zou ik met de ESP BOX en $ 50 een apparaat kunnen hebben dat ik uit de doos kan halen, kan flashen en in mijn keuken, slaapkamer, kantoor of waar dan ook kan opstellen.

Software

Espressif levert niet alleen deze bijna perfecte hardware; het bedrijf is al lange tijd een voorstander van open source. Ik vond het geweldig om te zien dat ze de ESP BOX ondersteunen met een verscheidenheid aan vrije en open source-bibliotheken:

- ESP IDF als standaard SDK;
- ESP ADF voor audiotoepassingen;
- ESP SR voor spraakherkenning;
- ESP DSP voor sterk geoptimaliseerde signaalverwerkingsroutines (FFT, vectorberekeningen enzovoort);
- een LVGL-component om LC-displays aan te sturen (met aanraakfunctionaliteit).

Met de ESP BOX en deze bibliotheken heb ik binnen een week een zeer eenvoudig proefconcept ontwikkeld dat spraak kan vastleggen, naar mijn spraakherkenning-implementatie kan sturen en het transcript naar een platform kan sturen om actie te ondernemen.

Audio

Zoals ik had geleerd van mijn eerdere (mislukte) pogingen, begint een spraakinterface bij het wekwoord. Je moet een spraakinterface met een specifiek commando kunnen aanspreken, het zo doen ontwaken zodat het begint spraak vast te leggen. Het wekwoord is het equivalent van een aan/uit-knop – hij mag niet 'zomaar' inschakelen en net als bij een echte knop moet hij telkens betrouwbaar werken. Als je voor spraak-toepassingen jezelf meerdere keren moet herhalen om het apparaat te activeren, kom je al snel in een scenario terecht waarin het sneller, gemakkelijker en veel minder frustrerend is om gewoon je telefoon te pakken en te doen wat je wilde doen.
Gelukkig biedt het ESP spraakherkennings-framework (SR, Speech Recognition) niet alleen een engine voor wekwoorden, maar bevat het ook verschillende ingebouwde wekwoorden. Ik vond de ontwaak-implementatie extreem betrouwbaar – het ontwaakte consistent en minimaliseerde onterechte activering. Dankzij ESP SR had ik een betrouwbare gesproken "aan/uit-knop".
Dan de volgende uitdaging - het verkrijgen van zuivere audio. Opnieuw kwam ESP-SR te hulp. ESP-SR bevat de Espressif AFE (audio front-end). Over deze AFE zouden we een boek kunnen volschrijven, maar in het kort biedt het een audioverwerkingslaag tussen de microfooningang en de audio-opname met de volgende functies:

Acoustic Echo Cancellation (AEC). Een van de vele uitdagingen bij spraakherkenning op grotere afstand zijn de akoestische eigenschappen van de fysieke omgeving. Afstand, harde reflecterende oppervlakken, ingewikkelde geluidsgolf-trajecten (hoeken, in de weg staande objecten) enzovoort kunnen veel echosignalen veroorzaken. De AEC-implementatie in ESP SR elimineert een groot deel van deze echo's, en verwijdert ook echo in scenario's met bidirectionele audio (zoals een speakerphone-toepassing).

Blind Source Separation (BSS). In luidruchtige omgevingen is het belangrijk om andere geluiden dan spraak, die kunnen bijdragen aan een slechte kwaliteit van de spraakherkenning, te elimineren. De ESP-SR BSS-implementatie, die gebruik maakt van meerdere microfoons, kan in wezen de audio-opname 'richten' op het binnenkomende geluid om zo de hoeveelheid achtergrondgeluid aanzienlijk te verminderen.

Noise Suppression (NS). In gevallen waar er slechts één microfoon is (of slechts één microfoon actief is) kan ruisonderdrukking de opname van niet door mensen gemaakte geluiden aanzienlijk reduceren. Voor toepassingen waarbij op maat gesneden hardware

wordt gebruikt, kan het gebruik van een enkele microfoon de materiaalkosten en ontwerpcomplexiteit aanzienlijk verlagen.

Voice Activity Detection (VAD). Een betrouwbaar wekwoord is slechts één stukje van de puzzel. Bij het ontwaken starten we met het opnemen van audio, en we moeten stoppen zodra de persoon ophoudt te spreken. VAD kan het begin en einde van spraak detecteren, zodat de gebruiker de opname niet handmatig hoeft te beëindigen.

Willow Inference Server

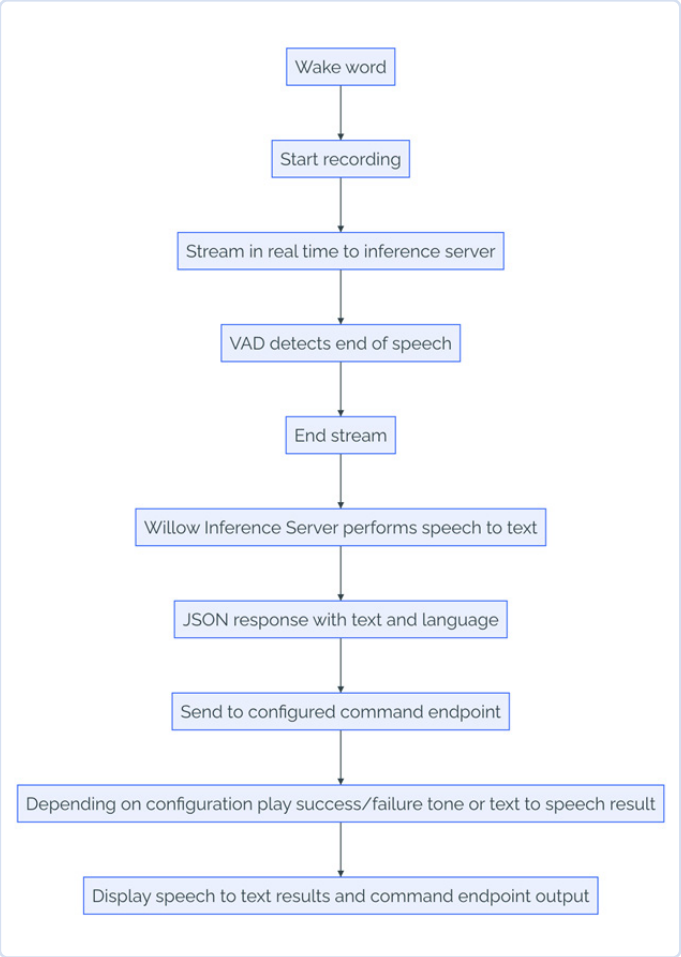
Nu we kunnen wekken, zuivere spraak kunnen opnemen en aan het einde van spraak kunnen stoppen, moeten we het ergens naartoe sturen. Gelukkig biedt Espressif het Audio Development Framework (ADF) voor deze taak aan. In mijn snelle en eenvoudige proefconcept gebruikte ik het ESP ADF *HTTP stream pipeline*-voorbeeld om AFE-bewerkte opgenomen audio in gedeelten via HTTP POST naar een HTTP-eindpunt te sturen. Ik kon mijn spraakherkenning inference server-implementatie snel aanpassen om binnenkomende audioframes van de ESP-BOX te bufferen, te wachten op een eindmarkering (dankzij VAD) en deze buffer onmiddellijk door te geven aan het onderliggende spraakherkenningsmodel. Wanneer de inference server het spraaktranscript retourneert, wordt dit als een HTTP JSON-antwoord aangeboden aan de ESP-BOX voor verdere uitvoering. Deze implementatie van een spraakherkennings-inference server werd bekend als de *Willow Inference Server* (*WIS*).

De ESP-BOX met Willow kan het spraaktranscript ontvangen en doorsturen naar een door de gebruiker geconfigureerde Home Assistant, OpenHAB of aangepast HTTP REST-eindpunt. Willow toont het transcript en het antwoord van het eindpunt op het display. Afhankelijk van de gebruikersconfiguratie zal het ook een geslaagd/mislukt-toon afspelen of tekst-naar-spraak van de Willow Inference Server gebruiken om de uitgevoerde tekst van het spraakcommando uit te spreken.

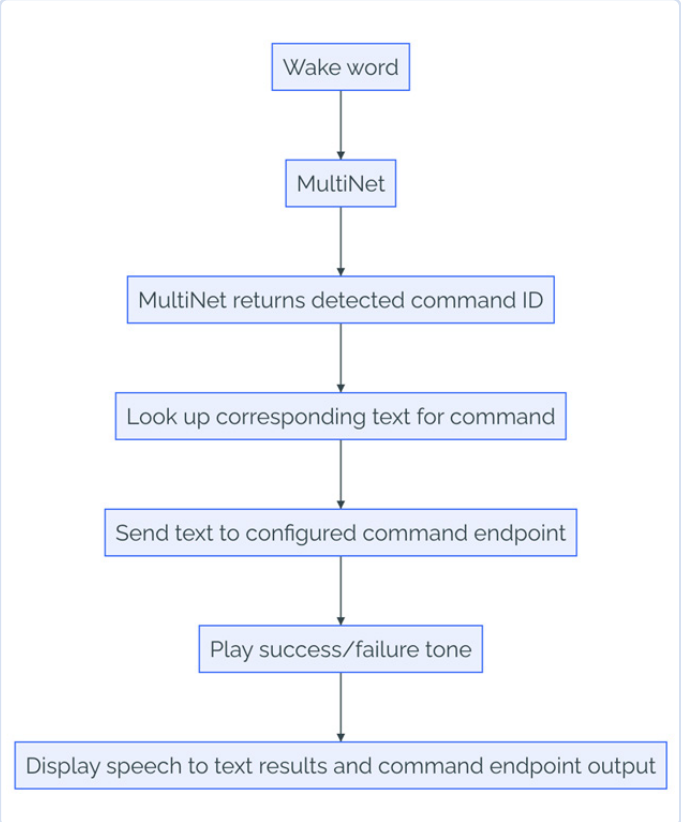
Nu kan ik met de ESP-BOX, Willow en de Willow Inference Server een wekwoord uitspreken, een commando opnemen en naar de Home Assistant sturen – met een latency van het einde van de spraak tot het voltooien van de actie van minder dan 500 ms (**figuur 4**), afhankelijk van de WIS-hardware en -configuratie.

Multinet: geen extra servers of hardware

De magie van ESP SR is echter nog niet voorbij. Naast de meegeleverde wekwoord-modellen bevat ESP SR een spraakherkenningsmodel genaamd MultiNet voor volledige on-device spraakherkenning. ESP SR biedt de mogelijkheid om tot 400 voorgedefinieerde spraakcommando's volledig op het apparaat te herkennen (zonder Willow Inference Server). Willow biedt ondersteuning voor MultiNet en wanneer het gebruikt wordt met Home Assistant kan het zelfs de namen van geconfigureerde entiteiten ophalen om automatisch deze grammatica te genereren (**figuur 5**) – zonder extra componenten en met prestaties en nauwkeurigheid die vergelijkbaar zijn met die van de Willow Inference Server voor deze voorgedefinieerde commando's.



Figuur 4. Willow Inferentie Server-flow.



Figuur 5. MultiNet modus-flow.

Nog steeds maker- en hacker-vriendelijk

Hoewel de ESP BOX met Willow Alexa-apparaten in enkele minuten kan vervangen, blijft hij trouw aan zijn Espressif-maker-afkomst. De ESP BOX ondersteunt een groot aantal componenten met diverse sensoren, GPIO, USB host-interface en meer via de uitbreidingsinterface [2]. Serieel en JTAG zijn natuurlijk beschikbaar via de USB C-poort op de hoofdunit.

Willow is geschreven in C met ESP IDF, maar de ESP BOX ondersteunt ook platforms als Arduino, PlatformIO en CircuitPython.

Met Willow en de ESP BOX hebben we nu wat lang de 'heilige graal' van open-source spraakinterfaces is geweest: een Alexa-achtige ervaring voor een vergelijkbare prijs maar met meer flexibiliteit, controle en volledige privacy die ook de open-source software- en hardware-interfaces biedt waar we allemaal van genieten!

230564-03 (vertaling: Hans Adams)

Vragen of opmerkingen?

Hebt u technische vragen of opmerkingen naar aanleiding van dit artikel? Stuur een e-mail naar de auteur via kris@tovera.com of naar de redactie van Elektor via redactie@elektor.com.

Over de auteur

Kristian Kielhofner is de oprichter van Willow – een open-source project om een lokale, zelf-gehoste spraakassistent te maken die een alternatief is voor Amazon Echo of Google Home. Sinds hij als kind met de Apple IIe speelde, is Kristian een technologie-enthousiast gebleven en heeft hij meerdere open-source projecten en startups in de spraak- en machine-learning-wereld opgericht. Kristian werkt nu aan Willow, andere open-source projecten en geeft advies aan beginnende technologiebedrijven.

WEBLINKS

[1] Willow-platform: https://heywillow.io

[2] ESP32-S3-BOX-3: https://www.espressif.com/en/news/ESP32-S3-BOX-3

Gerelateerde producten

- **ESP32-S3-BOX-3**
  www.elektor.nl/20627