un serveur de reconnaissance de la parole open-source...

...et l'ESP BOX

Kristian Kielhofner (États-Unis)

Beaucoup d'entre nous apprécient l'Alexa Echo et les appareils similaires, mais toujours avec des inquiétudes concernant la vie privée et l'utilisation des données. La plateforme Willow est une alternative gratuite et open-source. Outre le Willow Inference Server, il faut une interface utilisateur vocale de haute qualité pour capturer l'audio et la nettoyer. Avec un prix attractif, l'ESP BOX d'Espressif procure non seulement les microphones et la puissance de traitement audio, mais elle s'accompagne également d'un puissant écosystème logiciel.

Depuis le lancement de la plateforme Alexa il y a huit ans, Amazon a vendu plus de 500 millions d'appareils Echo. En revanche, il demeure des inquiétudes et des controverses concernant la vie privée, l'utilisation des données et les tentatives de plus en plus ennuyeuses d'Amazon pour faire débourser davantage aux utilisateurs d'Echo. Willow [1] est une plateforme qui procure une interface utilisateur vocale concurrente d'Alexa, conviviale pour les makers, gratuite et open source, sans sacrifier la qualité ni se ruiner.

Matériel

Raspberry Pi

Une interface utilisateur vocale de haute qualité doit s'intégrer dans le monde physique et interagir avec lui. L'écosystème open-source a tenté depuis longtemps de créer des interfaces utilisateur vocales en utilisant le Raspberry Pi, divers microphones, etc. (**figure 1**), mais cette approche présentait des inconvénients significatifs :

- Le prix. En achetant un Raspberry Pi, un écran LCD tactile, un réseau de microphones de haute qualité, un haut-parleur, un boîtier adapté, etc., vous atteignez un prix au moins trois fois supérieur à celui d'un appareil Echo (50 € ou moins). Cette approche nécessite l'approvisionnement des composants, un assemblage minutieux, la création d'un boîtier adapté, le développement de logiciels, etc. Répéter cette opération pour plusieurs appareils dans votre entourage augmente considérablement le temps et le coût nécessaires.
- La disponibilité. La situation s'est améliorée récemment mais, y compris pour le Raspberry Pi, la chaîne d'approvisionnement de ces composants a connu des problèmes.

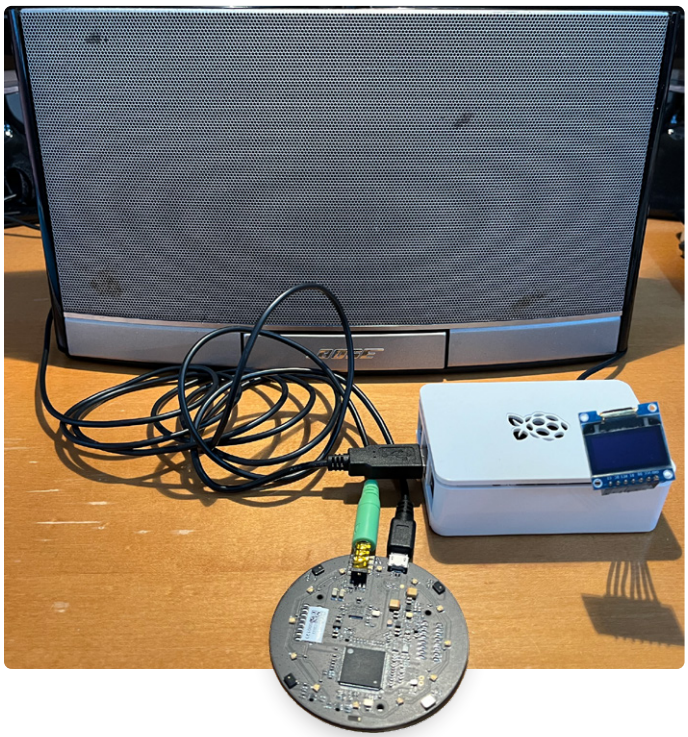

Figure 1. D'après les tentatives antérieures de l'auteur avec le Raspberry Pi.

Au cours des dernières années, obtenir ces composants tenait de l'impossible ou de l'inutilement coûteux à cause des coûts de distribution.

- La gestion. Les solutions à base de Raspberry Pi nécessitent souvent une distribution Linux complète qui prenne en charge les composants matériels, ainsi qu'un ensemble conséquent de logiciels nécessaires pour une interface utilisateur vocale. La gestion d'une demi-douzaine d'ordinateurs Linux peut s'avérer difficile pour de nombreux utilisateurs.
- La qualité. Amazon (et d'autres) ont investi des ressources considérables dans la conception d'une interface utilisateur vocale capable de fonctionner dans des environnements acoustiquement difficiles avec une grande variété de locuteurs humains. Il ne s'agit pas simplement d'installer un microphone et un haut-parleur sur un Raspberry Pi.
- L'écosystème. Une fois que vous avez obtenu une transcription précise d'une commande vocale, vous devez en faire quelque chose et fournir un retour d'information à l'utilisateur.
- La performance. Avec les implémentations de reconnaissance de la parole les plus optimisées, un Raspberry Pi 4 peut effectuer une reconnaissance presqu'en temps réel avec la qualité de modèle la plus basse. La précision laisse à désirer et le « temps réel » n'est tout simplement pas assez rapide - surtout si l'on considère qu'une erreur de transcription vous obligera à répéter la commande, ce qui élimine l'aspect pratique d'une interface utilisateur vocale.

Le **tableau 1** montre qu'un Raspberry Pi est légèrement plus rapide que la reconnaissance vocale en temps réel avec le plus bas niveau de qualité de modèle disponible.

Il est intéressant de noter que les modèles de reconnaissance de la parole ont une performance globale en temps réel plus élevée avec des segments de parole plus longs. Malheureusement, les commandes vocales des assistants vocaux sont très courtes, ce qui entraîne des pertes de performance significatives. Ce benchmark favorise en fait la performance de reconnaissance de la parole en temps réel du Raspberry Pi.

Nous connaissons et apprécions tous le Raspberry Pi – il est fantastique pour un large éventail d'applications et de cas d'usage. Malheureusement, un assistant vocal avec reconnaissance et synthèse de la parole n'en fait pas partie.

Comme le montre un exemple de capture d'écran (**figure 2**), la reconnaissance de la parole de Willow avec Willow Inference Server a pris 222 millisecondes de la fin du discours à la confirmation de l'action effectuée par le système domotique Home Assistant. Cela représente environ 25 % du temps nécessaire au Raspberry Pi 4 pour effectuer la reconnaissance de la parole seule, tout en utilisant un modèle de reconnaissance de la parole nettement plus précis que le « petit » modèle qui est à peine en temps réel sur le Raspberry Pi.

Comme beaucoup d'autres, j'ai tenté à plusieurs reprises avec différentes approches au fil des ans de créer une interface utilisateur vocale. Malheureusement, le résultat a toujours été le même : beaucoup de travail et de coûts pour une démo intéressante, mais totalement impraticable et inutilisable dans le monde réel à cause des problèmes mentionnés ci-dessus (plus d'autres).

Tableau 1. Performances de reconnaissance de la parole du Raspberry Pi 4.

Modèle	Taille du faisceau	Durée de la parole (ms)	Temps d'inférence (ms)	Rapport au temps réel
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×

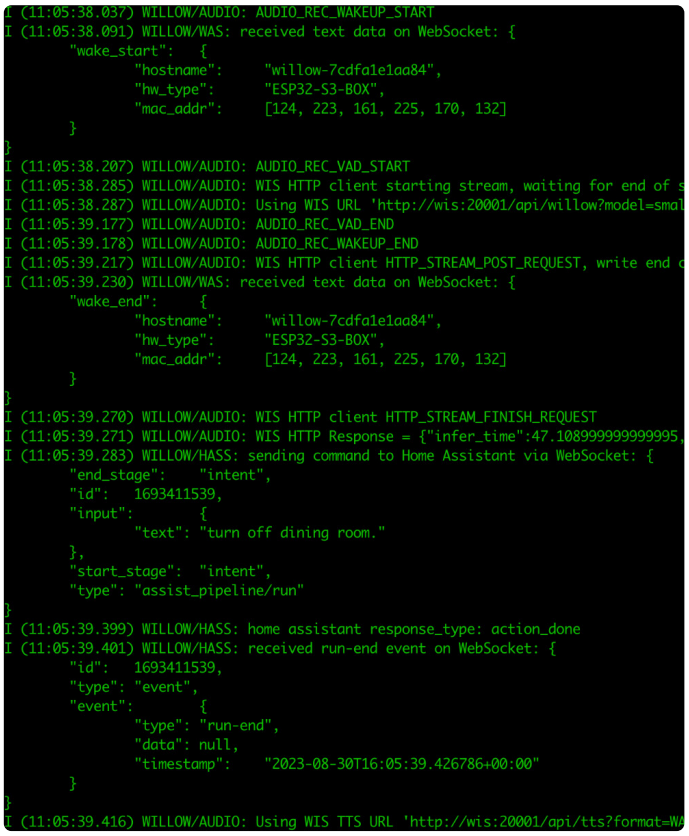

Figure 2. Performances de reconnaissance de la parole de Willow.

ESP BOX

J'ai alors découvert la plateforme de développement ESP BOX d'Espressif (**figure 3**). Comme beaucoup d'entre vous, j'utilise des appareils Espressif pour diverses applications depuis une dizaine d'années et je sais à quel point le matériel est robuste, riche en fonctionnalités, rentable et facilement disponible. Je sais également, grâce à des projets antérieurs, qu'Espressif fournit de nombreuses solutions logicielles et une documentation de grande qualité pour son matériel et ses logiciels.

L'ESP BOX d'Espressif est une plateforme de développement spécialement conçue pour ce cas d'usage. Pour environ 50€ chez votre détaillant ou distributeur d'électronique de bricolage préféré, vous obtenez :

Figure 3. L'ESP32-S3-BOX-3 est équipée d'un écran de 2,4 pouces, de deux microphones et d'un haut-parleur.

- Un ESP32-S3 avec 16 Mo de mémoire flash et 16 Mo de RAM connectée par SPI à haute vitesse
- Un écran tactile capacitif de 2,4 pouces
- Deux microphones (très important – voir plus loin)
- Un haut-parleur pour la sortie audio
- Un bouton de mise en sourdine du matériel
- De nombreuses possibilités d'extension avec les nouveaux ensembles et composants ESP32-S3-BOX-3

Le tout prêt à l'emploi dans un boîtier esthétique et acoustiquement optimisé. Théoriquement, avec l'ESP BOX et 50€, je pourrais avoir un appareil que je pourrais sortir de la boîte, flasher et placer dans ma cuisine, ma chambre, mon bureau, etc.

Logiciel

En plus de fabriquer ce matériel presque parfait, Espressif est depuis longtemps un champion de l'open-source. J'ai été ravi de voir que l'ESP BOX est prise en charge par une sélection de bibliothèques libres et gratuites :

- ESP IDF comme environnement de développement
- ESP ADF pour les applications audio
- ESP SR pour la reconnaissance de la parole
- ESP DSP pour des routines de traitement du signal hautement optimisées (FFT, calcul vectoriel, etc.)
- Un composant LVGL pour piloter des écrans LCD (tactiles compris)

Avec l'ESP BOX et ces bibliothèques, j'ai développé en l'espace d'une semaine une preuve de concept rudimentaire, capable de capturer la parole, de l'envoyer à mon implémentation de reconnaissance puis de fournir la transcription à une plateforme pour qu'elle réagisse.

Audio

Comme je l'avais appris lors de mes précédentes tentatives infructueuses, une interface utilisateur vocale commence par un mot de réveil. Vous devez pouvoir vous adresser à cette interface en prononçant un mot spécifique, pour qu'elle se réveille et commence à capturer la parole. Le mot de réveil est l'équivalent d'un bouton d'alimentation – il ne doit pas s'allumer au hasard, et lorsque vous appuyez sur le bouton, il doit fonctionner à chaque fois. Pour les applications d'interface vocale, si vous devez vous répéter plusieurs fois pour que l'appareil s'active, vous vous retrouvez rapidement dans un scénario où il est plus rapide, plus facile et beaucoup moins frustrant de sortir votre téléphone de votre poche et de faire ce que vous essayez d'y faire.
Heureusement, le canevas ESP Speech Recognition fournit non seulement un moteur de mots de réveil, mais aussi plusieurs mots de réveil. J'ai trouvé que l'implémentation du réveil était extrêmement fiable – réveillant systématiquement tout en minimisant les fausses activations. Grâce à ESP SR, je disposais d'un « bouton d'alimentation » vocal fiable.
Le défi suivant consiste à obtenir un son propre. Une fois de plus, l'ESP SR vient à la rescousse. L'ESP SR inclut l'AFE (audio front end) d'Espressif. On pourrait beaucoup dire sur l'AFE, mais en résumé, il fournit une couche de traitement audio performante qui réalise, entre l'entrée du microphone et la capture audio:

L'annulation de l'écho acoustique (AEC). Les propriétés acoustiques de l'environnement physique constituent l'un des nombreux défis de la reconnaissance de la parole en champ lointain. La distance, les surfaces réfléchissantes dures, les chemins audios alambiqués (coins, objets sur le parcours), etc. peuvent produire beaucoup d'écho. La mise en œuvre de l'AEC dans l'ESP SR élimine une grande partie de cet écho, ainsi que l'écho dans les scénarios où il y a de l'audio bidirectionnel (comme dans une application de téléphone à haut-parleur).

La séparation aveugle des sources (BSS). Dans les environnements bruyants, il est important d'éliminer les bruits non vocaux qui peuvent contribuer à une mauvaise qualité de la reconnaissance de la parole. L'implémentation ESP SR BSS, qui utilise plusieurs microphones, peut en particulier « focaliser » la capture audio sur la direction du son entrant afin de réduire de manière significative le bruit de fond capturé.

La suppression du bruit (NS). Dans les cas où il n'y a qu'un seul microphone (ou un seul actif), la suppression du bruit peut réduire fortement la capture de sons non humains. Pour les applications où du matériel personnalisé est utilisé, des microphones uniques peuvent réduire significativement les coûts de la nomenclature et la complexité de la conception.

La détection de l'activité vocale (VAD). Un mot de réveil fiable n'est qu'une pièce du puzzle. Lorsque nous nous réveillons et commençons à capturer de l'audio, nous devons arrêter la capture lorsque la personne a fini de parler. La VAD est capable de détecter le début et la fin de la parole, de sorte que l'utilisateur n'a pas besoin de mettre fin manuellement à l'enregistrement.

Willow Inference Server

Maintenant que nous pouvons réveiller, capturer de la parole propre et nous arrêter à la fin de celle-ci, nous devons l'envoyer quelque part. Heureusement, Espressif fournit son environnement de développement audio (ADF) pour cette tâche. Dans ma démonstration rapide de faisabilité, j'ai utilisé l'exemple de pipeline de flux HTTP de l'ESP ADF pour fournir des séquences audio capturées post-AFE via HTTP POST à un point de terminaison HTTP. J'ai pu rapidement adapter l'implémentation de mon serveur d'inférence de reconnaissance de la parole pour mettre en mémoire tampon les trames audios entrantes de l'ESP-BOX, attendre un marqueur de fin (grâce à VAD) et transmettre immédiatement cette mémoire tampon au modèle de reconnaissance de la parole sous-jacent. Lorsque le serveur d'inférence de reconnaissance revient avec la transcription de la parole, celle-ci est fournie en tant que réponse HTTP JSON à l'ESP-BOX pour être exécutée en suivant. Cette implémentation du serveur d'inférence de reconnaissance vocale est connue sous le nom de *Willow Inference Server (WIS)*.

L'ESP-BOX avec Willow est capable de prendre la transcription de la parole et de l'envoyer à Home Assistant ,configuré par l'utilisateur, à OpenHAB ou à un point de terminaison HTTP REST personnalisé. Willow affichera sur l'écran la transcription de la reconnaissance de la parole et la réponse du dispositif de commande. En fonction de la configuration de l'utilisateur, il émettra également une tonalité de réussite ou d'échec ou utilisera la synthèse vocale de Willow Inference Server pour énoncer le texte de sortie résultant de la commande vocale.

Aujourd'hui, avec l'ESP-BOX, Willow et Willow Inference Server, je suis capable de prononcer un mot de réveil, de saisir une commande et de l'envoyer à Home Assistant – avec une latence entre la fin de la parole et la réalisation de l'action bien inférieure à 500 ms (**figure 4)**, en fonction du matériel et de la configuration du WIS.

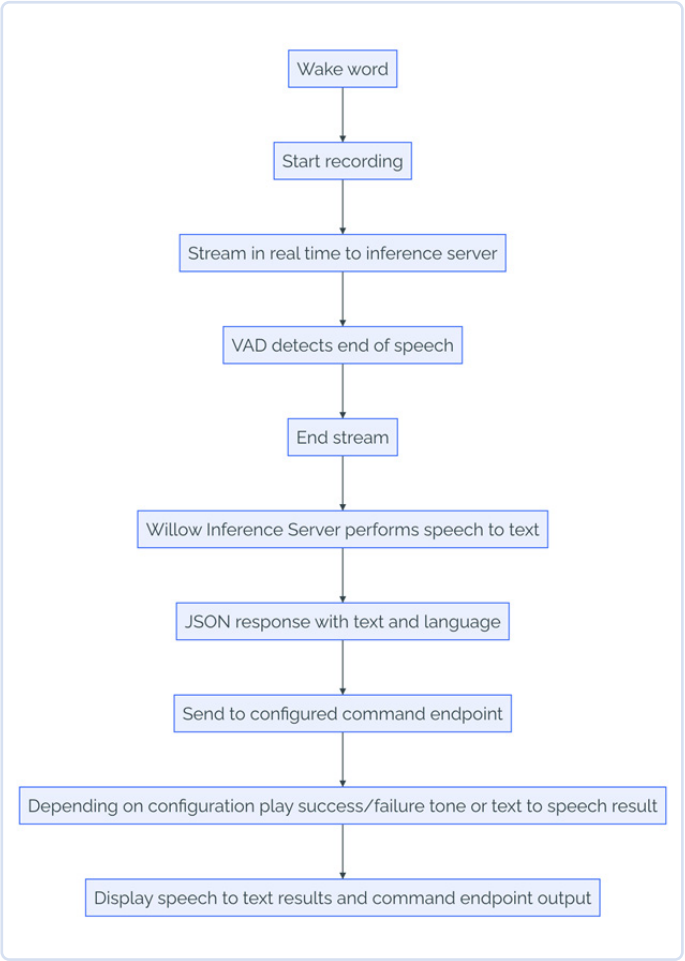

Figure 4. Flux du serveur d'inférence Willow..

Multinet : Pas de serveurs ou de matériel supplémentaires

Mais la magie de l'ESP SR ne s'arrête pas là. En plus des modèles de mots de réveil fournis, ESP SR inclut un modèle de reconnaissance de la parole sur l'appareil appelé *MultiNet* pour une reconnaissance des commandes vocales entièrement sur l'appareil. L'ESP SR permet de reconnaître jusqu'à 400 commandes vocales prédéfinies entièrement sur l'appareil (sans Willow Inference Server). Willow prend en charge MultiNet et, lorsqu'il est utilisé avec Home Assistant, il peut même extraire les noms des entités configurées pour générer automatiquement cette grammaire (**figure 5)** - sans aucun composant supplémentaire et avec des performances et une précision comparables à celles de Willow Inference Server pour ces commandes prédéfinies.

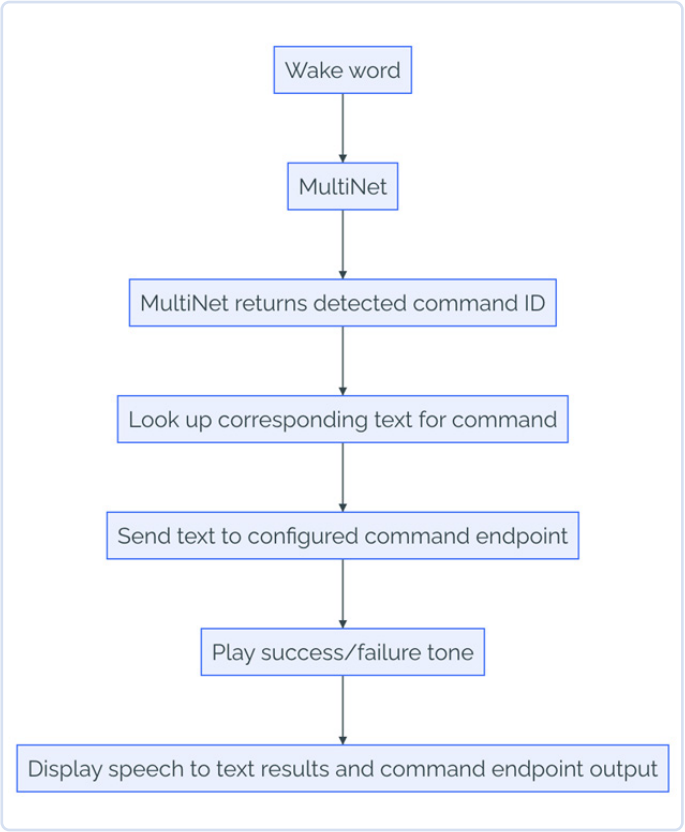

Figure 5. Flux du mode MultiNet.

Toujours à l'écoute des créateurs et des passionnés

L'ESP-BOX avec Willow peut remplacer les appareils Alexa en quelques minutes, tout en restant fidèle à ses racines pour maker d'Espressif.

L'ESP-BOX supporte une large gamme [2] de composants avec divers capteurs, GPIO, interface hôte USB et plus encore via l'interface d'extension. Les interfaces série et JTAG sont bien sûr disponibles via le port USB C de l'unité principale.

Willow est écrit en C avec ESP-IDF mais l'ESP-BOX prend en charge également des plateformes comme Arduino, PlatformIO, et CircuitPython.

Avec Willow et l'ESP-BOX, nous avons maintenant ce qui a longtemps été le « Sacré Graal » des interfaces vocales open-source : une expérience comparable à Alexa à un prix similaire avec plus de flexibilité, de contrôle et une confidentialité totale. Le tout avec des logiciels open-source et des interfaces matérielles de maker que nous apprécions tous !

VF : Denis Lafourcade — 230564-04

Questions ou commentaires ?

Envoyez un courriel à l'auteur (kris@tovera.com) ou contactez Elektor (redaction@elektor.fr).

À propos de l'auteur

Kristian Kielhofner est le fondateur de Willow, un projet open-source visant à créer un assistant vocal local et autonome apte à rivaliser avec Amazon Echo/Google Home. Depuis son enfance passée à jouer avec l'Apple IIe, Kristian a toujours été un passionné de technologie et a fondé de nombreux projets open-source et startups dans les domaines de la voix et de l'apprentissage automatique. Kristian consacre aujourd'hui son temps à Willow et à d'autres projets open-source, tout en conseillant des entreprises technologiques en phase de démarrage.

Produit

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

LIENS

[1] Plate-forme Willow : https://heywillow.io
[2] ESP32-S3-BOX-3 : https://www.espressif.com/en/news/ESP32-S3-BOX-3