

Spracherkennung ohne Cloud mit Snips.AI

Problem: Alexa, Siri, Cortana & Co übertragen Sprachdaten an unbekannte Server in der Cloud. Was mit den Daten nach der Auswertung passiert, ist unklar.

Lösung: Wir übertragen keine Daten in die Cloud, sondern verarbeiten per Mikrofon aufgenommenen Sprach-Audiodateien direkt auf dem Raspberry Pi (Umwandlung Speech-To-Text!), identifizieren Wörter mit Wahrscheinlichkeit und starten anhand der erkannten Befehle Skripte, die z.B. Relais schalten. Hierfür ist gar kein Netzwerk erforderlich.

Mittel: Snips.AI wird über eine Webplattform „vortrainiert“ und mit bestimmten, zu erkennenden Begriffen „vorgeladen“.

Nachteil von Snips.AI: Während der „Natural Language Support“ einer freien Lizenz unterliegt, ist die ASR („Audio Speech Recognition“)-Komponente ist **Nicht OpenSource** (nur „free for private use“), und müsste bei Serienproduktion und Verkauf von Produkten lizenziert werden (pro-Gerät-Lizenz). Alternativen:

Task 1: Installation

1. Betriebssystem (Raspbian) installieren.

2. Snips.ai Platform installieren (siehe <https://docs.snips.ai/getting-started/quick-start-raspberry-pi#step-3-install-the-snips-platform>)

Spracheingabe erfolgt über USB-Mikrofon, Lautsprecher (für akustisches Feedback Alexa-like) über 3,5mm Klinke. Hierzu ist ggf. Netzwerk erforderlich, wenn es nicht über die vorgeschlagene qemu-chroot-Umgebung direkt auf der SD-Karte passiert.

NPM (NodeJS Plattform Tool) installieren, Info unter:

<https://websiteforstudents.com/install-the-latest-node-js-and-nmp-packages-on-ubuntu-16-04-18-04-its/>

```
sudo apt install npm  
sudo apt install curl
```

Achtung: Bitte SICHERSTELLEN, dass auf dem Raspi die richtige Uhrzeit gesetzt ist, sonst ist das SSL-Zertifikat nicht gültig und der folgende „curl“-Befehl liefert eine leere Seite!!!

```
curl -sL https://deb.nodesource.com/setup_8.x | sudo bash -  
sudo apt install nodejs
```

Um zu sehen ob alles richtig installiert wurde folgendes eingeben:

```
node -v
```

```
npm -v
```

Wenn hier eine Versionsnummer erscheint, ist alles OK.

Jetzt kann man erst den Befehl aus der Anleitung ausführen:

```
sudo npm install -g snips-sam
```

Achtung: Für das nächste Kommando muss der SSH Server auf dem Pi laufen.
(`sudo /etc/init.d/ssh start`)

```
sudo ssh pi@raspberrypi  
(pi, pw: raspberry)
```

Achtung: Aktuell (28.9.2019) gibt's noch kein „buster“ Release von snips, daher muss für das neueste Raspbian eine ältere Installations-URL eingetragen und ein paar Dependencies händisch installiert werden. Nur bei Raspbian „buster“ erforderlich:

```
echo "deb https://raspbian.snips.ai/stretch stable main" | \
    sudo tee /etc/apt/sources.list.d/snips.list
sudo apt update
sudo apt install libgfortran3
sudo apt install libatlas3-base=3.10.3-8+rpil
sudo apt install libttspic0-data
sudo apt install snips-asr
wget http://ftp.fr.debian.org/debian/pool/non-free/s/svox/libttspic0-
utils_1.0+git20130326-9_armhf.deb
wget
http://ftp.fr.debian.org/debian/pool/non-free/s/svox/libttspic0_1.0+git201303
26-9_armhf.deb

sudo dpkg -i libttspic0-utils_1.0+git20130326-9_armhf.deb
sudo dpkg -i libttspic0_1.0+git20130326-9_armhf.deb

sudo apt install snips-tts
```

Ansonsten (Raspbian stretch):

```
sudo systemctl start snips-init
```

Achtung: Laut Snips-Anleitung sollte der Pi hier rebooten. Falls nicht, selbst neu starten, damit die snips-Dienste per systemd gestartet werden.

```
sudo systemctl status snips-init
```

Status muss "live" oder „idle“ sein

```
sudo systemctl start snips-audio
```

```
Is it a Snips Makers Mit? n
```

```
sudo systemctl start snips-test-microphone / speaker
```

Task 2: Erstellen und Deployen

(Um den gewünschten Assistant zu installieren, muss zuvor ein Account auf der Snips Web-Plattform eingerichtet und dort ein Assistant mit den gewünschten Keywords erstellt worden sein, der dann heruntergeladen und lokal installiert wird. Danach ist KEIN Netzwerk mehr erforderlich).

```
sam login
sam install assistant
```

(Kann sehr lange dauern)

Alternativ: Manuelle Installation (ohne sam):

<https://docs.snips.ai/articles/raspberrypi/manual-setup>

```
sam logout
sam watch
```

Nun kann getestet werden: über das *Wake Word* "Hey Snips" und dann einem in dem Assistant antrainiertem Wort z.B. an, aus, rechts, links, stufe (1-5)

"Hey Snips" kurz warten "aus"

Tests

Testergebnis

Erkennungsrate (m/w) **ohne Hintergrundgeräusche** bei **Nahaufnahme**:

Steckmikrofon: 100%

Raum Mikrofon: 100%

Erkennungsrate (m/w) **mit Hintergrundgeräusche** bei **Nahaufnahme**:

Steckmikrofon: 77,8%

Raum Mikrofon: 83,3%

Eigenes Wake Word "Alisa" erstellen

Tutorial: <https://docs.snips.ai/articles/platform/wakeword/personal#4-test-your-model-and-adjust-its-sensitivity>

Problem: Das Personal Wake Word wird i.d.R. nur über die Stimme, mit der aufgezeichnet wurde.

Event-Handler implementieren um Kaffeemaschine oder Ventilator zu steuern

1. Schritt: Event-Handler implementieren

Entweder alle benötigten Dateien aus Git kopieren (`index.js`, `node_modules`, `packages.json`) und im gewünschten Verzeichnis einfügen

oder:

Gehe in das Verzeichnis `cd /home/pi`

Erstelle einen neuen Projektordner mit `mkdir` und wechsle zum Projektordner.

Gebe den Befehl `npm init` an. Dieser erzeugt ein `package.json` welcher u.a die Projektbeschreibung beinhaltet.

Füge mit `npm install mqtt --save` das MQTT Paket hinzu

Installiere die Shell-Dependencies `npm install shelljs --save`

Erstelle eine neue Datei mit dem Namen `index.js`

Füge dieser Datei den Event-Handler hinzu (Script zur GPIO-Ansteuerung
Software/snips/Kaffeemaschine

Mit `node index.js` kann der Event Handler getestet werden.

2. Schritt: Shell-Skripte einfügen

Gehe in das Verzeichnis `/opt/kaffeemaschine` und füge die Shell-Skripte hinzu

für `an.sh` wird der GPIO27 benutzt

für `aus.sh` wird der GPIO22 benutzt

für `kaffee.sh` wird der GPIO4 benutzt

für `espresso.sh` wird der GPIO17 benutzt

Um die Shell-Skripte, die die GPIOs mit root-Rechten verändern, ohne Passworteingabe ausführen zu können, muss folgendes beachtet werden:

Der Benutzer, der das Skript aufruft, muss OHNE PASSWORT ein `sudo` nach `root` durchführen können (sonst kein Zugriff auf GPIOs).

Eintrag für `User_Snips`: `/etc/sudoers`

`snips ALL=NOPASSWD: ALL`

3. Schritt: Event-Handler automatisch während Bootvorgang starten

Bitte folgende Anleitung verwenden

<https://docs.snips.ai/getting-started/quick-start-raspberry-pi#step-5:-create-a-constants-file-and-set-the-realm-cloud-instance-url-4>

In der Service Datei die in der Anleitung erstellt wurde, muss noch der Pfad der zuvor erstellten `index.js` (Event-Handler) angepasst werden, dort werden dann die Shell-Skripte durch den `snips`-Daemon aufgerufen.

Links

<https://www.elektormagazine.de/news/review-snips-spracherkennung-auf-dem-raspberry-pi>