



Talking-Pi Personalisierungsanleitung



Index

- 1. Einführung
- 2. Einrichtung der Cloud Speech API
- 3. Benutzerdefinierte Wake-Words
- 4. Anlernen eigener Befehle
- 5. Support



1. Einführung

Sehr geehrter Kunde,

vielen Dank, dass sie sich für unser Talking-Pi Produkt entschieden haben.

Die eigene Programmierung und Modifizierung des Systems ist ein wesentlicher und wichtiger Teil dieses Projektes.

In den folgenden Schritten werden wir Ihnen Schritt für Schritt erklären und beibringen, wie Sie die Google Assistant API einrichten können, ihren Talking-Pi einsatzbereit machen und wie Sie Ihren Talking-Pi nach Ihren eigenen Wünschen programmieren und modifizieren können.

Sollten dennoch, während der Arbeit mit Ihrem Talking-Pi, Probleme auftreten, so können Sie uns selbstverständlich gerne kontaktieren.

Aktuelle Anleitungen, Projekte und ein Community-Forum ist außerdem hier zu finden:

Talking-Pi Website



2. Einrichtung der Cloud Speech API

Mit unserem vorinstallierten Talking-Pi Betriebssystem wartet der Talking-Pi geduldig darauf aktiviert zu werden.

Das Image können Sie direkt <u>hier</u> herunterladen.

Installieren Sie das Image auf einer micro-SD-Karte (eine Kapazität von mindestens 8GB wird empfohlen) und legen Sie diese anschließend in Ihren Raspberry Pi ein.

Mit einer geänderten Cloud Speech API stehen Ihnen 80 Sprachen zur Verfügung. Außerdem können Sie eigene Wake-Words und Befehle in das System einpflegen.



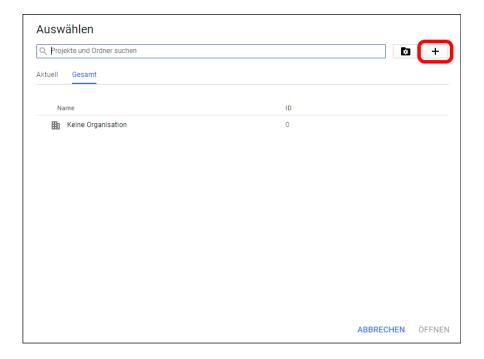
Achtung! Die Google Cloud Speech API ist kostenfrei, insofern Sie diese weniger als 60 Minuten pro Monat verwenden. Andernfalls wird Ihnen ein Betrag von 0,06\$ pro 15 Sekunden berechnet. Sie erhalten allerdings zuvor eine Benachrichtigung, falls Sie das Limit von 60 Minuten überschreiten sollten.

Öffnen Sie zunächst die Google Cloud Console.

Loggen Sie sich mit Ihrem Google-Konto ein oder erstellen Sie ein neues, falls Sie noch keines besitzen. Öffnen Sie hier die Projektübersicht und erstellen Sie ein neues Projekt.







Als nächstes wählen Sie aus dem linken Menü die Option "Abrechnung".

Fügen Sie hier eine neue Bezahlmöglichkeit hinzu und folgen Sie den weiteren Anweisungen.

Kehren Sie nach dem Hinzufügen einer neuen Abrechnungsoption zum Abrechnungs-Menü zurück und stellen Sie sicher, dass Ihr neu erstelltes Projekt mit dieser Zahlungsoption verknüpft ist.

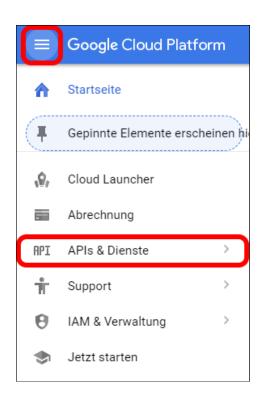
Um Ihr Projekt mit einer Zahlungsoption zur verknüpfen können Sie einfach die drei Punkte anwählen.

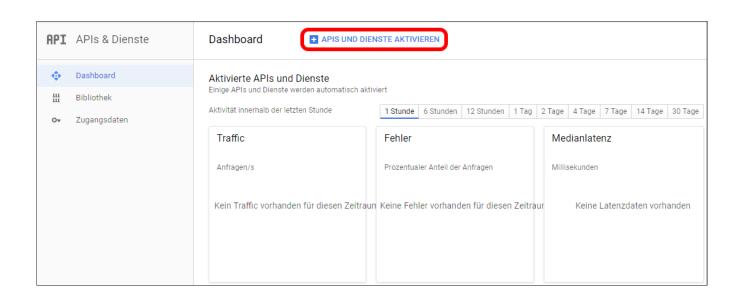
Als nächstes werden die benötigten Dienste aktiviert.

Verwenden Sie hier das im Bild markierte Symbol um ein neues Projekt zu Ihrer Liste hinzuzufügen.

Geben Sie dann einfach einen beliebigen Namen für Ihr Projekt ein und schließen Sie den Vorgang ab.

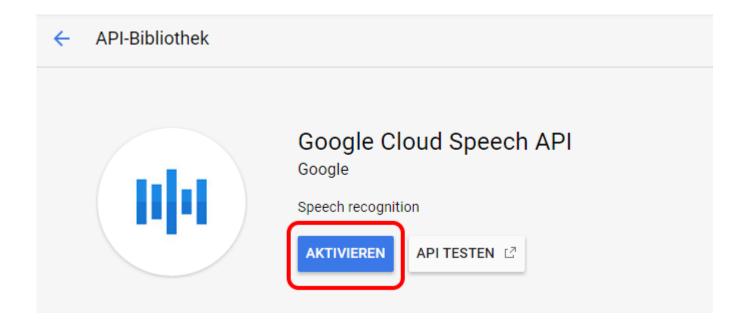
Öffnen Sie dann das Menü und wählen Sie hier die Option "APIs & Dienste" und aktivieren Sie das Menü.



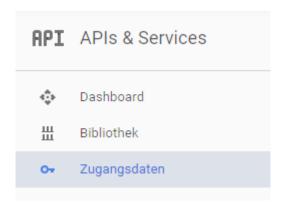




In der sich nun öffnenden API-Bibliothek suchen Sie nach dem Begriff "Google Cloud Speech API". Wählen Sie die gefundene API aus und aktivieren Sie diese.



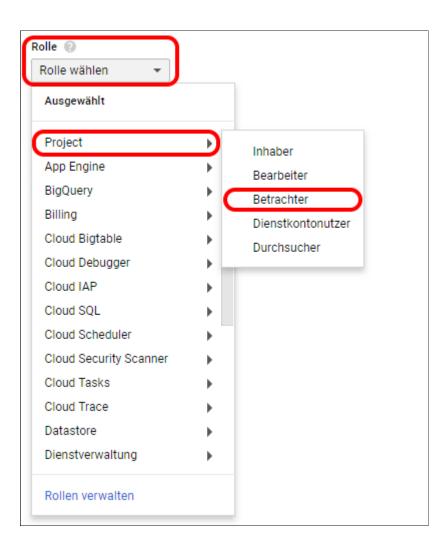
Wählen Sie nun in der "API & Services" Übersicht den Punkt "Zugangsdaten" aus.



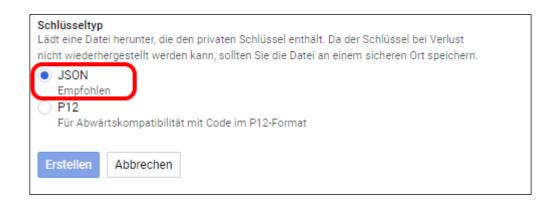


Wählen Sie hier "Anmeldedaten erstellen" und erstellen Sie hier einen "Dienstkontoschlüssel".

Als Rolle definieren Sie hier Project und Betrachter:



Als Schlüsseltyp wählen Sie "JSON" aus.

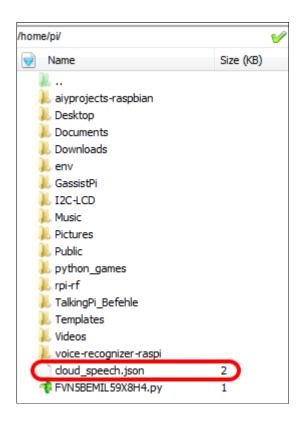




Erstellen Sie nun Ihre Zugangsdaten, indem Sie auf "Erstellen" klicken.

Der Download Ihrer Daten startet automatisch.

Nenne Sie die heruntergeladene Datei in **cloud_speech.json** um und kopieren Sie die Datei in das Benutzerverzeichnis Ihres TalkingPis.



Nach einem Neustart ist die Cloud Speech API auf Ihrem TalkingPi einsatzbereit.



3. Benutzerdefinierte Wake-Words

Die Aktivierung kann entweder durch das Drücken des Knopfes erfolgen oder durch ein so genanntes Wake-Word, welches vorher in das System integriert wurde.

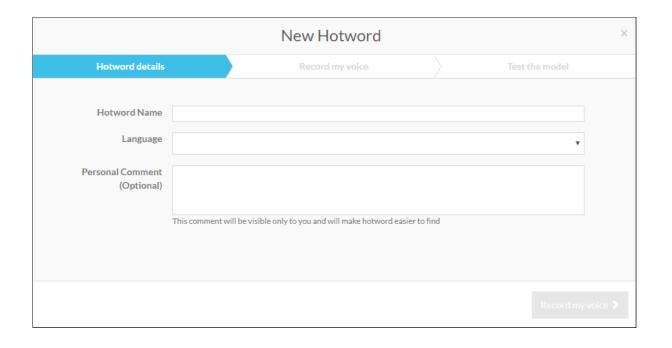
In dem von uns vorbereitetem System reagiert der Talking-Pi auf die beiden Wake-Words "Talking-Pi" und "Alexa".

Sollten Ihnen diese beiden Wake-Words nicht ausreichen, oder sollten Sie Ihren Talking-Pi etwas personalisieren möchten, so können Sie Ihrem Talking-Pi selbstverständlich auch eigene Wake-Words beibringen.

Für die benutzerdefinierte Wake-Word Erkennung verwenden wir die Gassist Pi Bibliothek in Verbindung mit der Snowboy Hotword Detection.

Diese beiden Bibliotheken sind auf unserem vorbereiteten Betriebssystem selbstverständlich bereits installiert.

Für die benutzerdefinierten Wake-Words können Sie im Snowboy Dashboard entweder ein eigenes Hotword aufnehmen, oder ein bereits vorhandenes Wake-Word verwenden.



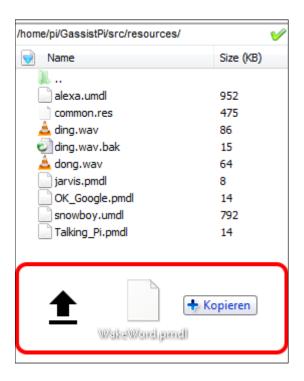


Hierbei nehmen Sie das Wake-Word drei mal auf, um es an Ihre Stimme zu adaptieren.

Im Anschluss können Sie die auf Ihr Wake-Word abgestimmte .pmdl-Datei herunterladen.

Kopieren Sie diese Datei anschließend in den folgenden Ordner Ihres Talking-Pi's:

/home/pi/GassistPi/src/resources/



Um nun das von Ihnen erstellte Wake-Word In Ihrem Talking-Pi zu aktivieren ist es nötig die Konfigurationsdatei der Gassist Pi Bibliothek zu modifizieren.

Geben Sie dafür den folgenden Befehl in ein Terminal auf Ihrem Talking-Pi:

sudo nano /home/pi/GassistPi/src/talkingPi_snowboy.py



Modifizieren Sie, wie im folgenden Beispiel zu sehen, das **models**-Array um eigene Wake-Word-Konfigurationsdateien hinzuzufügen oder zu entfernen.

```
models = ['/home/pi/GassistPi/src/resources/alexa.umdl', '/home/pi/GassistPi/
src/resources/Talking_Pi.pmdl', 'home/pi/GassistPi/src/resources/
[meinwakeword].pmdl']
```

Beachten Sie hierbei bitte, dass die Phrase "[meinwakeword]" ein Platzhalter ist, der durch den Dateinamen Ihres eigenen Wake-Words zu ersetzen ist.

Ihre Änderungen können Sie anschließend mit der Tastenkombination **STRG+O** speichern und den Editor mit der Kombination **STRG+X** verlassen.

Nach einem Neustart des Systems sollte Ihr persönliches Wake-Word im System integriert sein und der Talking-Pi auf sein neues Kommando reagieren.



4. Anlernen eigener Befehle

Um eigene Sprachkommandos hinzuzufügen, die der Talking-Pi als eigenen Befehl erkennt, ist es nötig den Befehl zuerst als erwartete Eingabe zu definieren.

Dies erhöht die Erkennungsrate.

Öffnen Sie dazu die Datei "googlecloudtalkingpi.py" mit Hilfe des folgenden Befehls:

```
sudo nano /home/pi/voice-recognizer-raspi/src/googlecloudtalkingpi.py
```

Fügen Sie hier den folgenden **recognizer.expect_phrase**-Befehl zu den bereits vorhandenen Phrasen hinzu:

```
recognizer.expect_phrase('[Ihr eigenes Sprachkommando]')
```

Tauschen Sie hierbei den Platzhalter durch Ihr eigenes Sprachkommando aus.

```
def main():
    # Definition aller zu erwartenden Befehle
    # Diese Erwartungen erhoehen die Erkennungsgenauigkeit des Talking-Pis
    # Aehnlich klingende Befehle, leichte Abweichungen oder auch Akzente in der Sprache
    # koennen durch die Definierung der Erwartungen besser erkannt werden
    recognizer = aiy.cloudspeech.get_recognizer()
    # Sie koennen hier weitere Phrasen hinzufuegen!
    recognizer.expect_phrase('mach das Licht an')
    recognizer.expect_phrase('mach das Licht aus')
    recognizer.expect_phrase('mach das Licht oben an')
    recognizer.expect_phrase('mach das Licht oben aus')
```

Da wir nun die zu erwartende Spracheingabe definiert haben, kann nun der Befehl aus eigentliche Erkennung definiert werden.

Fügen Sie dazu, im weiteren Verlauf der bereits geöffneten Datei, den folgenden Inhalt zu den bereits existierenden Definitionen hinzu.



Achten Sie bitte darauf auch hier den Platzhalter durch Ihren eigenen Befehl auszutauschen.

Auch die Einrückungen müssen Sie unter Umständen nach dem Kopieren korrigieren.

Orientieren Sie sich hierfür einfach an den bereits vorhandenen Befehls-Definitionen.

```
print('You said "', text, '"')
# Pruefe auf folgenden Befehl
if 'mach das Licht an' in text:
    # Wenn der Befehl passt, dann fuehre die folgenden Befehls-Dateien aus
    if Display:
        # Der folgende Befehl fuer das Display wird nur ausgefuert wenn der entsprechende Wert oben gesetzt wurde. Dieser ist standardmaessig aktiv.
        subprocess.call(["python","/home/pi/I2C-LCD/Befehl1.py"])
    subprocess.call(["bash", "/home/pi/TalkingPi_Befehle/Licht/LichtAn.sh"])
# Spiele ausserdem folgende Tondatei ab
    subprocess.Popen(["aplay", "/home/pi/GassistPi/src/resources/dong.wav"], stdin=subprocess.PIPE, stdout=subprocess.PIPE, stderr=subprocess.PIPE)
# Warte danach 2 Sekunden
    time.sleep(2)
elif 'mach das Licht aus' in text:
    if Display:
        subprocess.call(["python","/home/pi/I2C-LCD/Befehl2.py"])
    subprocess.call(["bash", "/home/pi/TalkingPi_Befehle/Licht/LichtAus.sh"])
subprocess.Popen(["aplay", "/home/pi/GassistPi/src/resources/dong.wav"], stdin=subprocess.PIPE, stdout=subprocess.PIPE, stderr=subprocess.PIPE)
    time.sleep(2)
```

Innerhalb der neu definierten Befehlsaufrufe, werden drei verschiedene Dateien aufgerufen.

Die ersten beiden Dateien ("/home/pi/I2C-LCD/EigenerBefehl.py" und "/home/pi/TalkingPi_Befehle/ Receiver/EigenerBefehl") definieren die Aktionen die beim Ausführen des entsprechenen Sprachbefehls aufgerufen werden.

Dabei stellt in diesem Beispiel die erste Datei die Ausgabe auf dem Display und die zweite Datei die eigentliche Befehlsfolge dar.

Die dritte Datei definierten den Ton der ausgegeben wird, sobald der Sprachbefehl erkannt wurde.

Sie können selbstverständlich beliebig viele aufzurufende Dateien hier eintragen.

Die von uns angelegte Struktur der drei Dateien (Display, Befehl, Ton) dient nur der Übersicht.



Bitte beachten Sie, dass die auszuführenden Dateien von Ihnen noch in den entsprechenden Ordnern angelegt werden müssen.

Sie können natürlich so viele Dateien definieren wie Sie benötigen.

In unserer ausgelieferten Standardkonfiguration haben wir allerdings jeweils die folgenden Dateien zur Ausführung definiert.

An diesen können Sie sich für ihre persönliche Konfiguration orientieren.

Display-Ansteuerung:

Dateiort: /home/pi/I2C-LCD/[IhreDatei].py

Beispiel:

```
import lcddriver
from time import *

lcd = lcddriver.lcd()
lcd.lcd_clear()

lcd.lcd_display_string("+++ Ausgabe +++", 1)
lcd.lcd_display_string("+Ich bin eine Ausgabe+", 2)
```

Ton-Ausgabe:

Dateiort: /home/pi/GassistPi/src/resources/[IhreTonDatei].wav

Weitere auszuführende Befehlsdateien:

Dateiort: /home/pi/TalkingPi_Befehle/

Dateiformat: .sh

Hinweis: Hier sind die einzelnen Befehlsdateien, um eine Übersicht zu ermöglichen, nochmals in die

Ordner Beamer, Licht, Computer und Receiver unterteilt.

Sie können selbstverständlich weitere Kategorien hinzufügen.

Bitte achten Sie jedoch darauf, diese dann bei der Dateireferenzierung richtig anzugeben.

Nach einem Neustart Ihres Talking-Pi's sollte Ihr neuer Sprachbefehl erfolgreich erkannt und verarbeitet werden können.



5. Support

Wir sind auch nach dem Kauf für Sie da. Sollten noch Fragen offen bleiben oder Probleme auftauchen stehen wir Ihnen auch per E-Mail, Telefon und Ticket-Supportsystem zur Seite.

E-Mail: service@joy-it.net

Ticket-System: http://support.joy-it.net

Telefon: +49 (0)2845 98469 – 66 (11- 18 Uhr)

Für weitere Informationen besuchen Sie unsere Website:

www.joy-it.net