🎤 Computer Voice Assistant - Wiki

Willkommen zum offiziellen Wiki des **Computer Voice Assistant** Projekts!

📋 Übersicht

Der Computer Voice Assistant ist ein **Star Trek-inspirierter** Voice Assistant mit Custom Wake-Word "Computer", entwickelt in Python.

Features:

- ✅ Custom Wake-Word "Computer" (Porcupine)
- ✅ Offline Speech-to-Text (Vosk)
- ✅ Natürliche Text-to-Speech (Edge TTS)
- ✅ Lokale Befehls-Ausführung
- ✅ LLM-Integration (ChatGPT, Perplexity)
- ✅ Home Assistant Integration
- ✅ Cross-Platform (Windows, Linux, Raspberry Pi)

🚀 Quick Start

1. Installation

```
git clone https://github.com/KoMMb0t/Computer-Voice-Assi.git
cd Computer-Voice-Assi
pip install -r requirements.txt
```

2. Konfiguration

```
# Erstelle .env Datei
echo "PICOVOICE_ACCESS_KEY=your_key_here" > .env

# Bearbeite config.ini
nano config.ini
```

3. Modelle herunterladen

```
# Vosk Modell (Deutsch)
cd models/
wget https://alphacephei.com/vosk/models/vosk-model-small-de-0.15.zip
unzip vosk-model-small-de-0.15.zip
```

4. Wake-Word trainieren

1. Gehe zu Picovoice Console
2. Erstelle neues Wake-Word: "computer"
3. Download `computer.ppn`
4. Speichere in `models/computer.ppn`

5. Starten

```
python 15_voice_assistant_configurable.py
```

📚 Dokumentation

Haupt-Dokumentation

- **Installation** - Detaillierte Installations-Anleitung

- **Befehle** - Liste aller verfügbaren Befehle
- **Konfiguration** - config.ini Referenz
- **Wake-Word Training** - Custom Wake-Word erstellen

Erweiterte Themen

- **LLM-Integration** - ChatGPT & Perplexity
- **Home Assistant** - Smart Home Steuerung
- **Cross-Platform** - Raspberry Pi, Jetson Nano
- **Audio-Processing** - Noise Reduction, VAD
- **Troubleshooting** - Häufige Probleme

🎯 Use-Cases

Basis-Befehle

```
"Computer, öffne YouTube"
"Computer, wie spät ist es?"
"Computer, öffne den Taschenrechner"
```

LLM-Fragen

```
"Computer, wie wird das Wetter morgen?"
"Computer, was ist 15 mal 23?"
"Computer, erkläre mir Quantenphysik"
```

Smart Home

```
"Computer, mach das Licht im Wohnzimmer an"
"Computer, stelle die Heizung auf 22 Grad"
"Computer, starte die Kaffeemaschine"
```

🛠️ Entwicklung

Projekt-Struktur

```
Computer-Voice-Assi/
├── 15_voice_assistant_configurable.py  # Haupt-Programm
├── 16_llm_integration_prototype.py     # LLM Manager
├── config.ini                          # Konfiguration
├── .env                                # API Keys
├── models/                             # Wake-Word & STT Modelle
│   ├── computer.ppn
│   └── vosk-model-small-de-0.15/
├── docs/                               # Dokumentation
└── tests/                              # Tests
```

Beitragen

Contributions sind willkommen! Siehe CONTRIBUTING.md

Schritte:

1. Fork das Repository
2. Erstelle Feature-Branch (`git checkout -b feature/AmazingFeature`)
3. Commit Änderungen (`git commit -m 'Add AmazingFeature'`)
4. Push zu Branch (`git push origin feature/AmazingFeature`)
5. Öffne Pull Request

📊 Roadmap

v4.0 - LLM Integration (Januar 2026)

- ✅ ChatGPT API
- ✅ Perplexity API
- ✅ Command vs. Question Classification
- ⏳ Konversations-History

v5.0 - Multi-Device (März 2026)

- ⏳ Raspberry Pi Support
- ⏳ Android App
- ⏳ Synchronisation

v6.0 - Smart Home (Mai 2026)

- ⏳ Home Assistant Integration
- ⏳ Licht-/Thermostat-Steuerung
- ⏳ Musik-Steuerung

🤝 Community

- **GitHub:** Computer-Voice-Assi
- **Issues:** Bug Reports & Feature Requests
- **Discussions:** Q&A & Ideas

📄 Lizenz

🙏 Credits

Entwickelt von: KoMMb0t
Inspiriert von: Star Trek Computer
Powered by:

- Picovoice Porcupine - Wake-Word Detection
- Vosk - Speech-to-Text
- Edge TTS - Text-to-Speech
- OpenAI - LLM Integration

Viel Spaß mit deinem Voice Assistant! 🎉