BSc Artificial Intelligence in Software Engineering
CDS205 Computer Science
Lecturer: Prof. Dr. Ana Petrus

# Talking Alarm Clock Using Raspberry Pi and State-of-the-Art AI

**Kai Zuberbühler**[1,*]

[1] *University of Applied Sciences of the Grisons*
[*] *kai.zuberbuehler@stud.fhgr.ch*

January 9, 2025

**Abstract**

In this project a simple voice assistant system using state-of-the-art artificial intelligence (AI) technologies is developed and deployed on a Raspberry Pi, exploring the possibility of easily integrating voice control systems into household devices. Utilizing a Raspberry Pi 4 Model B and a sound card HAT, the system incorporates Automatic Speech Recognition (ASR), Text-to-Speech (TTS), and natural language processing to create a voice assistant capable of real-time interaction for controlling basic functions of an alarm clock. A modular software architecture allows for the easy substitution of service providers, ensuring flexibility and extensibility. Key functionalities include setting and managing alarms, announcing the time, and engaging in conversational interactions with the user. The project demonstrates the feasibility of embedding state-of-the-art AI technologies into affordable hardware, offering a seamless and intuitive user experience. This work serves as a basic proof-of-concept for future developments in AI-enhanced household devices, highlighting the potential for broader applications and further research in this domain. More technical information and code is available at https://github.com/k-zubi/cds-205-raspberry-pi under the MIT License.

# 1 Introduction

In recent years, artificial intelligence (AI) has made significant progress in the areas of natural language processing, speech recognition, and speech synthesis, opening up new paths to human-machine interaction. Large language models (LLMs) like GPT-4 (OpenAI et al., 2024) exhibit human-like conversational abilities and can call tools and functions as part of software systems, allowing for novel ways of interaction with software through natural language. At the same time, speech recognition systems like Whisper (Radford et al., 2022) are approaching human-level accuracy and robustness and speech synthesis systems like the ones from ElevenLabs (ElevenLabs, 2022) allow for speech output with human-like voices.

Despite these advancements, neither most common home assistants nor household devices are integrating these technologies so far. In this work, I explore whether these technologies could be effectively integrated into affordable hardware and offer the functionality of a household device, specifically an alarm clock, and allow for accurate and practical interaction with the device using natural language.

# 2 Methodology

## 2.1 Hardware Configuration

For this project a Raspberry Pi 4 Model B provided by the university was used. Despite offering robust processing power, a Raspberry Pi does not offer the necessary computational power to effectively run a language model with sufficient speed. (Anthony et al., 2023; Pounder, 2023) Due to the Raspberry Pi lacking a sound card, a WM8960 Hi-Fi Sound Card HAT was installed in order to easily record and play audio. This sound card also features a small integrated microphone and a small button addressable via GPIO. For testing two small speakers and the microphone integrated on the sound card was primarily used.

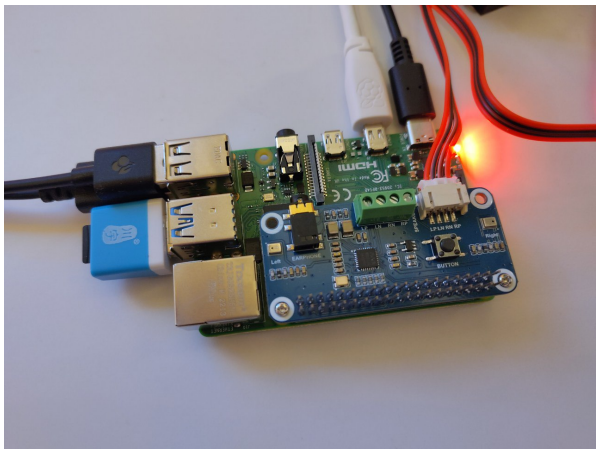

Figure 1: Photo of the Raspberry Pi with the sound card HAT installed

## 2.2 Software Architecture

The system was developed using Python and a modular design approach to enable easy replacement of all the key components, such as switching to another service provider.

The main script orchestrates the entire workflow by interfacing between the user's real-time input, the ASR for transcription, the chat module for processing, and the TTS module for delivering responses.

An audio recorder component is responsible for recording the audio for it to be later passed

to the speech recognition. The default implementation uses GPIO to react to the button on the sound card HAT being pressed to allow the user to start and stop speaking.

An Automatic Speech Recognition (ASR) component is responsible for converting spoken language into text. The "Wizper" model, an adaption of Whisper available on the API from Fal.Ai, was selected for its combination of high accuracy with speed. (Artificial Analysis, 2024b)

A language model component is responsible for generating the text. The API service from Cerebras, an AI chip company, was used due to its unparalleled speed. (Artificial Analysis, 2024a) To generate the text responses, Llama 3.3 70B, a language model developed by Meta (Meta, 2024), offering state-of-the-art performance for its size, was used.

A Text-to-Speech (TTS) component is responsible for converting the generated text response to playable speech audio. Eleven Labs TTS service was used, due to its ability to generate highly natural and customizable voices. The Turbo v2.5 model is used by default as it offers a high quality along with good speed and pricing (Artificial Analysis, 2024c).

A chat module manages the dialogue and processes user inputs, including to make calls to various functions as required. The model is instructed to use a structured output in markdown with sections for thinking, responding to the user and calling functions. A description of all available functions calls is provided to the model in the system message. Based on these, it calls the necessary functions along with necessary parameters using a specified YAML format.

A functions module encapsulates the logic for each specific function the assistant can call, such as setting an alarm. It is designed so that adding new functions to the catalogue is easy and straightforward.

# 3 Results

The system successfully integrated state-of-the-art cloud AI services in speech recognition, natural language processing, and speech synthesis into a compact, modular setup, allowing the user to control basic functions of an alarm clock using their voice. Implementing the architecture in a modular style worked as intended and simplified testing.

During tests, the ASR component showed high levels of accuracy in user-device interaction, depending however on the physical distance of the user to the microphone, the type of microphone used and background noise. But even the small microphone built on top of the sound card HAT was adequate when speaking within an arms length of it and no significant background noise. The latency averaged at a few seconds from finishing speaking until the audio from the TTS model starts playing, which is good for such types of interaction but not on the level of typical human conversation.

Apart from some minor challenges when transmitting an audio stream to the TTS API, which

could be resolved, the project could be implemented without problems.

# 4 Discussion

The system achieved functional interaction with good latency but remains significantly slower than typical human conversation. The emergence of "omni" models, such as GPT-4o (OpenAI, 2024), which integrate the three main components (ASR, LM and TTS) into one model suggests potential for reduced latency in the future. Testing revealed that environmental factors, such as noise, the distance of the speaker to the microphone and the type of microphone used, can impact speech recognition accuracy, highlighting the need for an adequate microphone or robust noise-cancelling in any voice-controlled device.

The current implementation's dependence on cloud services raises important considerations about internet dependency, service reliability and privacy that would need addressing in a commercial product. While local processing would resolve privacy issues, the computational requirements currently exceed the capabilities of affordable hardware in this price range. However, given rapid advances in AI model efficiency (Xiao et al., 2024) and specialized hardware this limitation may soon be overcome.

The project demonstrates that sophisticated AI interaction can be implemented using relatively affordable hardware, suggesting potential viability in some household devices in the future. Additionally, the ongoing costs of cloud API usage would need consideration in any commercial implementation.

Several promising avenues for future development emerge from this work, including the integration with broader smart home ecosystems, the implementation of continuous wake-word detection, the exploration of local processing options using specialized AI hardware or the integration of multi-modal interaction.

# 5 Conclusion

This project demonstrates the viable integration of advanced AI technologies for voice control into affordable hardware platforms. By successfully combining state-of-the-art ASR, TTS, and LLM capabilities provided by cloud APIs, intuitive user-machine interaction using natural language is achievable with affordable hardware configurations. Challenges remain in areas such as running the models locally for privacy protection, achieving human-level response latency and improved robustness to noise.

Overall, the system successfully establishes a basic proof of concept for the interaction with household devices through natural conversation that can be easily adjusted and expanded due to the modular software architecture. Several promising directions for future development emerge including, most notably, the integration of emerging "omni" models for reduced latency and the exploration of local processing options using specialized AI hardware.

# References


Anthony, Q., Biderman, S., & Schoelkopf, H. (2023, April 18). *Transformer math 101* [EleutherAI blog]. Retrieved December 26, 2024, from https://blog.eleuther.ai/transformer-math/

Artificial Analysis. (2024a). *Comparison of AI models across quality, performance, price | artificial analysis*. Retrieved December 26, 2024, from https://artificialanalysis.ai/models

Artificial Analysis. (2024b). *Speech to text (ASR) providers leaderboard & comparison | artificial analysis*. Retrieved December 26, 2024, from https://artificialanalysis.ai/speech-to-text

Artificial Analysis. (2024c). *Text to speech models and providers leaderboard | artificial analysis*. Retrieved December 26, 2024, from https://artificialanalysis.ai/text-to-speech

ElevenLabs. (2022, October 17). *First long-form speech synthesis platform for publishers and creators*. Retrieved December 26, 2024, from https://elevenlabs.io/blog/long-form-speech-synthesis-for-publishers-and-creators

Meta. (2024). *Llama 3.3 | model cards & prompt formats*. Retrieved December 26, 2024, from https://www.llama.com/docs/model-cards-and-prompt-formats/llama3_3/

OpenAI. (2024). Gpt-4o system card. https://cdn.openai.com/gpt-4o-system-card.pdf

OpenAI, Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F. L., Almeida, D., Altenschmidt, J., Altman, S., Anadkat, S., Avila, R., Babuschkin, I., Balaji, S., Balcom, V., Baltescu, P., Bao, H., Bavarian, M., Belgum, J., . . . Zoph, B. (2024). Gpt-4 technical report. https://arxiv.org/abs/2303.08774

Pounder, L. (2023, March 25). *How to create your own AI chatbot server with raspberry pi 4* [Tom's hardware]. Retrieved December 26, 2024, from https://www.tomshardware.com/how-to/create-ai-chatbot-server-on-raspberry-pi

Radford, A., Kim, J. W., Xu, T., Brockman, G., McLeavey, C., & Sutskever, I. (2022). Robust speech recognition via large-scale weak supervision. https://arxiv.org/abs/2212.04356

Xiao, C., Cai, J., Zhao, W., Zeng, G., Lin, B., Zhou, J., Zheng, Z., Han, X., Liu, Z., & Sun, M. (2024, December 6). Densing law of LLMs. https://doi.org/10.48550/arXiv.2412.04315