SCIENTIFIC Publishing Limited

Journal Name

http://ojs.ukscip.com/index.php/xxxx

Article Type (Article, Review, Communication or Editorial)

Interactive Conversational AI with IoT Devices for Enhanced Human-Robot Interaction

Received: Day Month Year; **Accepted**: Day Month Year; **Published**: Day Month Year

Abstract

Significance - The rapid advancements in conversational AI and IoT technologies have opened up new possibilities for human-machine interaction. Despite the progress, a gap exists in integrating these two fields to create more centralized, intuitive, and engaging user experiences. Current integrations typically consist of specialized hardware-software pairs that do not fully leverage the capabilities of advanced conversational models, thereby limiting their applicability. This research proposes a general solution to bridge the capabilities of various IoT devices with the oversight and control abilities of AI language models, enhancing the potential for more versatile and natural IoT-AI-human interactions.

Aim and Approach - This research presents the design and development of an IoT system operated by an AI language model and conversationally managed by humans to operate robots. Based on this setup, the initial goal is to create a framework for interactively controlling a robotic arm. The approach involves using a Raspberry Pi as a central control system and ChatGPT API to manage conversations and execute given commands.

Results - The developed IoT-AI system demonstrated efficient and reliable human-robot interaction where the user can entertain a conversational interaction with the robotic arm. It effectively captures user voice inputs, processes them through advanced AI models, and generates appropriate commands for the robotic arm, achieving an average voice-to-motion latency of 5.5 s. An example of commands are “engage arm”, “move right 20” (i.e. move the robotic arm to the right of 20 cm) combined with more conversational commands such as “can you hear me?”, “what’s your name?”. While some latency and voice recognition challenges exist, the overall performance confirms the viability of using conversational AI for natural and intuitive robotic control.

Conclusions - This research successfully integrates conversational AI with IoT devices, resulting in a more user-centric and efficient human-robot interaction. The system highlights the significant potential of precisely translating natural language commands into robotic actions, enhancing user experience and operational efficiency.

Keywords: Artificial Intelligence; Intelligent Communication; Conversational AI; IoT; Human Machine Inter4action; Human Robot Interaction keyword.

---

1. Introduction

The robotics industry is increasingly incorporating *Artificial Intelligence* (AI) and *Internet of Things* (IoT) technologies, leading to faster deployment, superior exteroceptive awareness and better *Human-Machine Interaction* (HMI). Conversational AI, particularly with the introduction of *Large Language Models* (LLMs) like OpenAI's GPT series, has demonstrated remarkable capabilities in understanding and generating human-like responses [1]. These models can engage in meaningful dialogues, answer questions, and even assist in various tasks, making them a powerful tool for enhancing HMI [2]. Meanwhile, IoT devices have become increasingly ubiquitous, enabling seamless integration, connectivity and data exchange between various devices and systems [3]. The IoT ecosystem has expanded to include a wide range of devices, from smart home appliances to industrial sensors and robots, creating a vast network of interconnected devices [4].

Despite these advancements, a gap exists in integrating these three fields to create more centralized, intuitive, and engaging user experiences. Current integrations are often specialized as software hardware package pairs and do not fully leverage the capabilities of advanced language models, limiting their broader applicability [5]. For example, many IoT devices rely on mobile apps or web interfaces for control and monitoring, which can be cumbersome and require users to learn specific commands or navigate complex menus [6]. Additionally, the lack of a unified solution or platform for managing multiple IoT devices can lead to a fragmented user experience and increased cognitive load [7].

In this context, conversational systems are usually combined with IoT systems and, moreover, with robotic devices, however the usual setting is more focused on home applications or on pure conversational mode where the device answer to the end-user command and execute action in a daily life environment. Not so much has been developed in terms of designing conversational system where the operator can interact with industrial devices and specify kinematics details in terms of position and orientation such as the operator effectively impart precise commands.

This research explores a general solution to bridge the capabilities of various IoT devices with the oversight and control of AI language models, enhancing the potential for more versatile and natural IoT-AI-based human-robot interaction and control. By leveraging the power of conversational AI, users can interact with IoT devices using natural language, making the experience more intuitive and accessible. Moreover, integrating multiple devices under a single AI-driven platform can facilitate the user experience and enable more complex and coordinated tasks [8].

This work aims:

- to present the design and development of an IoT system operated by an AI language model and controlled conversationally by humans.
- The approach involves using a Raspberry Pi as a central control unit and a ChatGPT model to manage conversations and execute given commands. The Raspberry Pi is a popular choice for IoT projects due to its accessible cost, flexibility, and extensive community support [9].
- Using the ChatGPT API, the system can leverage the state-of-the-art language understanding and generative capabilities of GPT-3.5, enabling more natural, safe and context-aware interactions [10].

Based on this configuration, the goal is

- to design a framework for interactively controlling a robotic arm. Robotic arms have been widely used in industrial settings for many tasks, such as assembly, packaging, and quality control. However, their adoption in smaller-scale applications has been limited due to the complexity of programming and control [11].
- By integrating a robotic arm with conversational AI, this study seeks to make robotic control more affordable and user-friendly, enabling novice users to perform complex tasks and providing greater freedom to define and automate robotic actions.

The integration of conversational AI with IoT devices has been explored in various contexts, such as smart homes [12], healthcare and industrial automation [13]. These studies have demonstrated the potential benefits of using natural language interfaces for controlling and interacting with IoT devices, improving usability and accessibility. However, most of these implementations rely on rule-based or limited-domain chatbots, lacking the flexibility and generalization capabilities of large language models like GPT-3.5. Rule-based chatbots are limited by their predefined set of rules and responses, making them less adaptable to new situations and user needs. On the other hand, large language models like GPT-3.5 can generate more diverse and contextually appropriate responses, enabling more natural and engaging conversations [14].

This research builds upon the existing literature by leveraging the state-of-the-art conversational AI capabilities of ChatGPT3.5-Turbo model and integrating it with a modular IoT architecture. The proposed system aims to provide a more natural and intuitive way of interacting with robotic devices, enabling users to control them through voice commands and engage in meaningful dialogues. The generalization capabilities of large language models are explored to handle a wide variety of user inputs and adapt to new situations, improving the overall user experience.

The structure of this research is as follows: The subsequent section, Methodology, will evaluate the architectural design of the IoT-AI voice interaction and control system. It will justify the integration of each hardware and software component while analyzing their capabilities and functionality. The Results section comprehensively evaluates the system's functionality, latencies, and user experience and discusses various issues, limitations, and potential solutions. The Conclusions chapter re-evaluates the accomplishments and their significance while setting goals for future improvements. The paper ends with Appendix 1, which presents the instructions to setup Raspberry Pi device to migrate booting from SD card to booting from SSD.

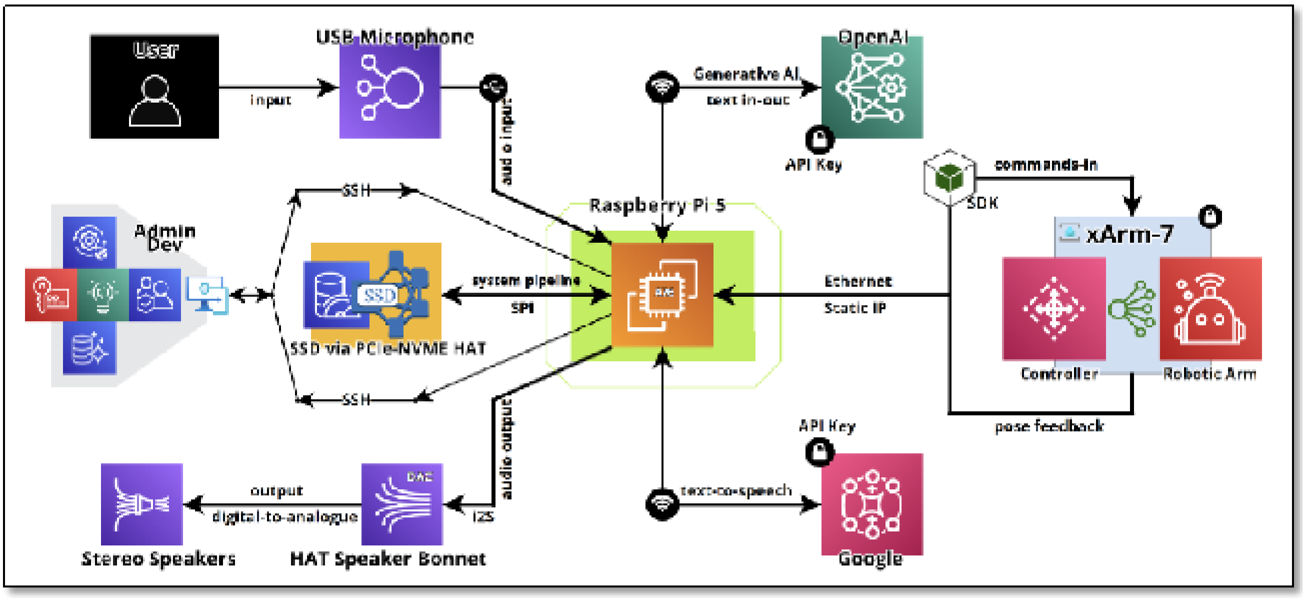


Figure 1. The IoT AI Voice Interaction and Control System Architecture

2. Materials and Methods

The following Section presents the main system architecture and components of the proposed system.

2.1. The Core Processing Unit - Raspberry Pi 5

Enhanced Processing Power

The Raspberry Pi 5 serves as the computational core of this IoT system (Fig. 1). It was chosen for its advanced capabilities, explicitly addressing the need for real-time processing and efficient data management and for its rapid deployment features. The device is powered by a Broadcom BCM2712, featuring a 64-bit quad-core ARM Cortex-A76 CPU running at 2.4GHz and offers a significant increase in CPU performance compared to its predecessors (Fig. 2). This substantial computing power is important for the project as it ensures real-time data processing from multiple peripheral devices and fast API-based data exchanges, enabling efficient handling of complex operations at low latencies. Its high clock speed and multiple cores are advanced architecture support that enhances parallel processing capabilities, allowing the system to efficiently manage simultaneous tasks and interactions across various devices and applications. These characteristics allow the integration of more complex functionalities and improved system adaptability, both crucial for maintaining high performance as project requirements expand and evolve.

Graphics and Display Capabilities

Although not essential for this work, Raspberry Pi 5 is equipped with an impressive 800MHz VideoCore VII GPU and support for dual 4Kp60 HDMI outputs, empowering the board to handle advanced graphical tasks and display outputs simultaneously. These GPU resources have the potential to be integrated into multi-threading processing configurations, thereby improving overall performance of the system. During the initial setup phase of the operating system, the video output option proved vital, simplifying the installation process by providing access to the GUI desktop control via the two micro-HDMI ports (2).

Memory Capacity

The version of Pi-5 used in this research is configured with 8GB of LPDDR4-3200 SDRAM. This synchronous dynamic random-access memory (SDRAM) is engineered to operate at a lower operating voltage and higher data rate, which reduces power consumption and increases the bandwidth between the memory and the processor to 3200 Mbps. Although 8GB of RAM is enough for this use case, the device capacity can be effectively augmented by swapping SSD space as virtual memory. In the current settings, the SDRAM direct access to data bits over the memory bus is superior for read-write operations and provides almost instantaneous executions. As a future improvement, the project will integrate SSD swap space into the functionality of the system as a fallback mechanism for cases when the SDRAM reaches full capacity. This integration can leverage the non-volatile storage advantage of SSDs, ensuring that data is retained even when the power is off. This configuration allows for efficient memory management and enhances the overall performance and reliability of the system.

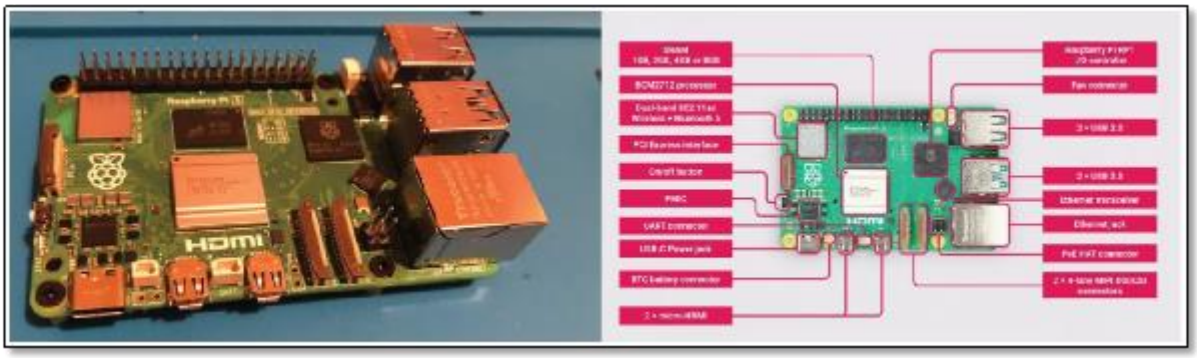


Figure 2. The Raspberry Pi 5 Pi Ports Layout with no components attached

Enhanced Storage Solutions

The Raspberry Pi 5 is configured by default to utilize the latest SDR104 micro-SD cards for storage and booting, featuring an SD card slot that supports the UHS-I (Ultra High Speed) standards. This slot is capable of theoretical peak speeds of up to 104 MB/s for read operations and up to 90 MB/s for write operations. However, consistently achieving these maximum speeds depends on the card quality and can often be challenging, rendering them suboptimal for the objectives of this project. To address these limitations, this research has leveraged the device's single-lane PCIe 2.0 interface to significantly enhance storage and performance capabilities. By employing a PCIe-based SSD HAT (Hardware Attached on Top) interface, the system has adopted a Kingston 256GB SSD NVMe M.2, optimized for PCIe Gen3, as its primary boot and storage device. This configuration leaps the read/write speeds to approximately 2100/1200 MB/s, substantially improving the original setup (Fig.3). The only difficulty with this upgrade is that it requires advanced knowledge of Linux system configuration and setups to transfer the OS boot from the SD card to the solid-state drive. The commands for such a transfer are documented in-depth in Appendix 1.

Connectivity Enhancements & Power Management

One of the main reasons for selecting the Raspberry Pi 5 for this study is its comprehensive networking capabilities, including both wired and wireless options (Fig. 4A). It incorporates a Gigabit Ethernet interface that supports Power over Ethernet+ (PoE+), compliant with the IEEE 802.3ab standard. This ethernet setup utilizes Cat5e or higher twisted pair cables and RJ45 connectors. This setup ensures a theoretical maximum data transmission rate of 1000 Mbps, making it ideal for data-intensive tasks where signal integrity is crucial, as these cables are inherently resistant to electromagnetic interference. Complementing its wired capabilities, the Pi 5 also offers dual-band wireless networking in accordance with the IEEE 802.11ac (Wi-Fi 5) standard. It operates on both 2.4 GHz and 5 GHz frequency bands, making it adaptable to various applications. The 5 GHz band provides increased data rates up to 1300 Mbps and reduced interference, suitable for high-bandwidth applications like streaming or intensive API data exchanges. Meanwhile, the 2.4 GHz band delivers a maximum data rate of 450 Mbps, offering extended range and compatibility with older devices. Additionally, the Pi 5 supports Bluetooth 5.0, which includes enhancements like faster speeds and greater ranges compared to previous versions, as well as Bluetooth Low Energy (BLE) for efficient communication with low-power devices.

Moreover, the Raspberry Pi 5 is supplied with a 5V/5A DC power input via USB-C with Power Delivery support, which enables the device to handle higher power loads necessary for running advanced power-hungry applications (Fig. 4B). The integration of a real-time clock with an external battery backup further enhances the system utility, especially in environments where time-synced data logging is crucial.

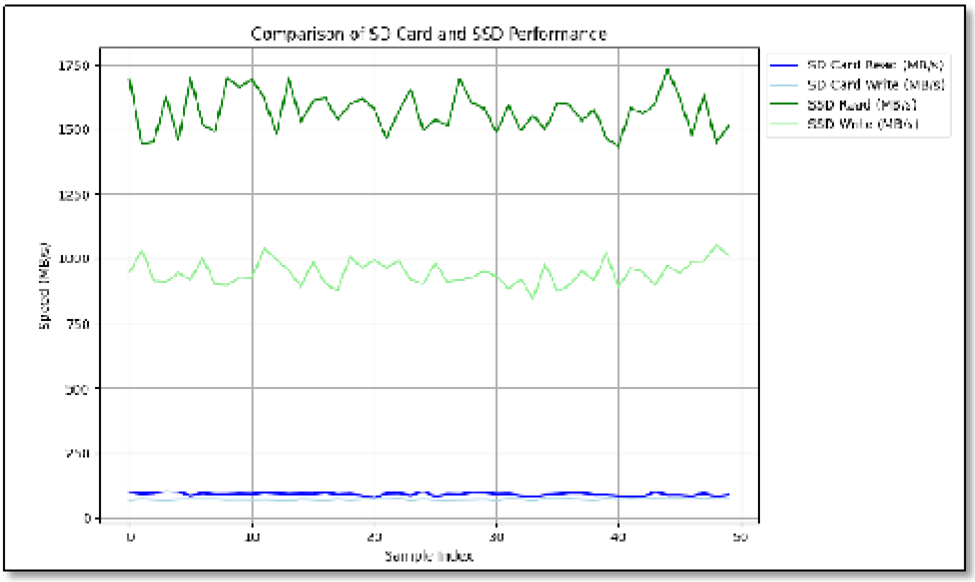


Figure 3. SD Card vs SSD, Read and Write Performance

This robust combination of wired, wireless, and Bluetooth connectivity options makes the Raspberry Pi 5 the most suitable IoT device to serve as the core processing unit for this project. The Ethernet connection is used to establish direct, high-speed, and low-latency control with the xArm-7 robotic arm. Wireless connectivity enables the Pi 5 system to maintain a permanent, stable internet connection, supporting data exchanges between three APIs: voice-to-text, OpenAI response generation, and text-to-voice. Additionally, the wireless connection facilitates access to the system via SSH for a developer-supervising computer unit. The Bluetooth connectivity is intended to be used to exhibit control via AI-interaction of a nearby drone. In addition, there is also the perspective for future expansion of the system include leveraging the three unused USB ports (two USB 2.0 and one USB 3.0) to connect and control three more IoT devices in a similar manner.

From a connectivity standpoint, as an IoT board, the Raspberry Pi 5 is demonstrating to be well-equipped to handle diverse networking scenarios, connecting seamlessly with other IoT or Edge devices, handling demanding data transfers or versatile wireless communications, including low-energy options.

Peripheral Compatibility and Integration

With extensive support for various peripherals, the Pi 5 can easily connect with a wide range of devices. The device includes two USB 2.0 ports suitable for general peripheral connections with a maximum theoretical throughput of 480 Mbps (Fig. 4A). Additionally, it includes two USB 3.0 ports, essential for high-speed data transfers, supporting speeds up to 5 Gbps. These high-speed interfaces are vital for this application that demands rapid data communication with external devices. The board also offers an extensive array of GPIO (General

Purpose Input/Output) pins, enabling direct interfaces with various sensors and actuators (Fig. 5A). This capability is fundamental for the customization of the system to easy integrate with specialized hardware components, supporting the scalability and functional expansion of this IoT project.

In the system architecture developed in this study, one of the USB-3 ports is used to connect the voice-in user input lane via a 360-degrees microphone, to capture omnidirectional voice commands.

Ecosystem, Community Support and Linux-based OS

In addition, Raspberry Pi 5 is integral to a dynamic ecosystem encompassing many HAT modules, bonnets and peripherals, enhancing its adaptability across various applications. These HATs provide diverse functionalities such as augmented I/O capabilities, sophisticated power management, sound-capturing devices, environmental sensing, and precise motor control, enabling customization for many specialized tasks. The ecosystem is further enriched by an array of compatible display and camera modules alongside extensive third-party hardware options. The engineering community surrounding the Raspberry is highly active, offering extensive documentation, tutorials, and a collaborative forum environment facilitated by the Raspberry Pi Foundation. This global network of developers, enthusiasts, and educators contributes a wealth of open-source resources, project guides, and software tools, making it an invaluable asset for both troubleshooting and innovation in hardware and software development.

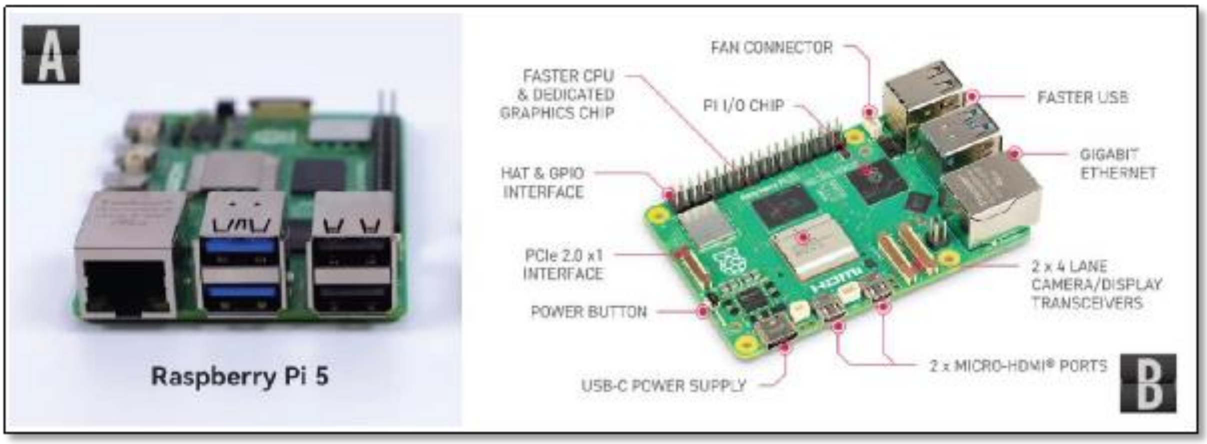


Figure 4. The Raspberry Pi 5 Connectivity (panels A and B)

Another key strength of Raspberry is its support for various operating systems, primarily Linux-based, which offer stability, security, and extensive customization options. Among these, Raspberry Pi OS x64 Desktop, based on Debian 12 "Bookworm," is used during this research (Fig. 5B). The advantages of using a Linux-based OS include robust security features, extensive community support, and a vast repository of software packages. Pi OS Desktop provides a user-friendly interface while leveraging the stability and security of Debian 12, making it ideal for both development and deployment in IoT applications.

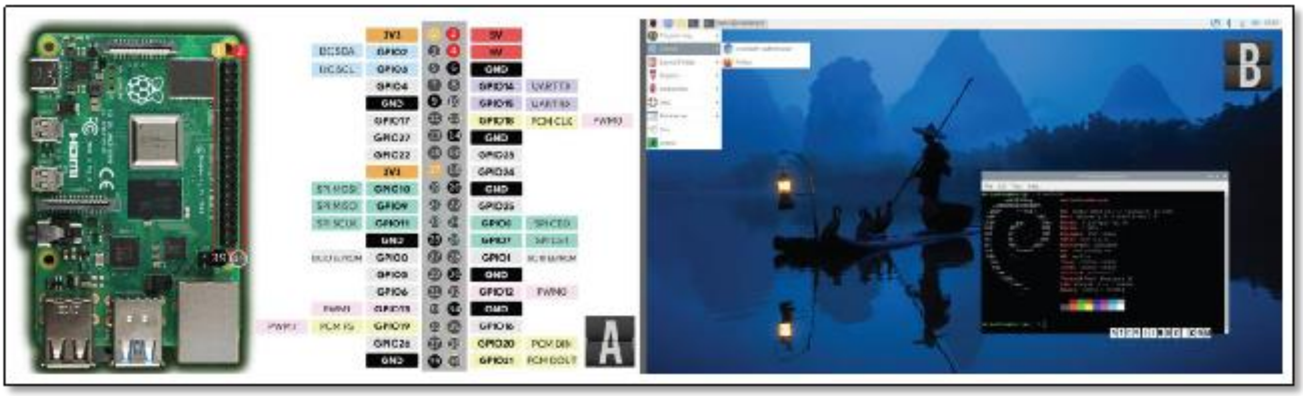


Figure 5. The Raspberry Pi 5 GPIOs (panel A) and the Raspberry Pi OS Bookwork (Debian 12) (panel B)

Furthermore, this project utilizes Python programming language, which is deeply integrated with the Linux-based Raspberry Pi OS, promoting efficient script execution and system management. Python's compatibility with the Debian system, combined with its extensive library ecosystem, supports the development of complex applications and simplifies the interaction with the hardware peripherals connected to the Pi device.

This robust combination of a versatile hardware platform, supportive community network, flexible operating system options, and integrated programming environments justifies that the Raspberry Pi 5 is a complete solution for this research and, overall, remains an important resource for educational and prototyping initiatives in technology and engineering fields.

2.2. The Adafruit Speaker Bonnet & 3W Stereo Speakers

The *Adafruit Speaker Bonnet* was selected as the primary audio output device for this development, specifically tasked with broadcasting the conversational AI's text-to-speech responses through its stereo speakers (Fig. 6, panels A & B). Its function is to enhances audio interactivity within IoT systems by delivering high-quality audio directly from the *Raspberry Pi*. This device is equipped with a built-in digital class-D amplifier, capable of driving up to 3 Watts per channel into 4-ohm stereo speakers. This setup ensures robust and clear audio output, which is paramount for delivering interactive commands and user feedback. Operating based on the *I2S* (Inter-IC Sound) digital sound interface; the bonnet guarantees high-fidelity audio by transmitting data via a digital communication link from the Raspberry Pi. This method provides cleaner and more reliable audio performance than traditional analogue outputs, which are often prone to noise and distortion. The bonnet design facilitates easy installation onto the Raspberry Pi via *GPIO* pins while providing access to all pins. This semi-HAT device optimizes the use of the I2S interface, reducing connectivity complexity and maximizing audio data throughput. Such integration ensures minimal latency, enabling instantaneous audio responses that are mandatory for interactive applications in IoT settings.

The project also incorporates stereo-enclosed speakers, chosen for their compatibility with the Adafruit Speaker Bonnet and suitability for the research interactive audio output requirements. These devices can deliver clear and high-fidelity sound, an essential feature for interactive user communication in various environments. The speakers are designed to handle 3 W of power with a 4 Ohm impedance. These characteristics ensure that the audio remains loud and clear in various settings, from noisy industrial locations to interactive public spaces, enhancing the system ability to provide uninterrupted audio feedback and instructions. Their connectivity with the Adafruit bonnet is a plug-and-play setup designed for reliable and quick installation.

The Adafruit Speaker Bonnet with the 3W stereo speakers are used in the project to make audible notifications and deliver voice feedback from AI-driven processes. The output setup provides efficient and compatible integration with the Raspberry Pi 5 hardware layout, making it fast to incorporate into software and rapid to deploy. This approach improves user engagement and system accessibility, making the IoT concept more intuitive and effective.

Figure 6. The Raspberry Pi 5 platform with the Adafruit Stereo Bonnet on top (panel A); the Adafruit Bonet with the 3W Speakers (panel B); all the Raspberry Pi 5 attachable components (panel C)

2.3. USB-3 Microphone 360

In this IoT project, a USB Conference Microphone is employed, featuring omnidirectional condenser technology to capture high-quality audio from all directions (Fig. 7B). This omnidirectional pickup pattern is invaluable in environments with multiple simultaneous interactions, ensuring comprehensive audio coverage. The microphone is equipped with sophisticated noise reduction algorithms that enhance audio clarity by filtering out background noise, a feature essential for robust voice recognition and processing in varied environmental conditions. Additionally, the microphone offers plug-and-play compatibility with the Raspberry Pi 5 OS, significantly boosting its adaptability and enabling effortless integration and deployment without the need for additional drivers. Furthermore, the microphone offers a plug-and-play functionality aligned with the Raspberry Pi 5 OS (Debian 12). These aspects enhance its adaptability, facilitating swift integration and deployment into the system without necessitating additional driver installations. This seamless integration was a key factor in choosing the device, which proves to be important in maintaining the system operational reliability and simplifying the software development around it.

2.4. Robotic Arm - UFactory xArm-7

In this research architecture, audio user commands are processed by generative AI and transmitted to the UFactory xArm-7 Robotic Arm (Fig. 7, panel A). Functioning primarily as an edge device, the robot processes data and executes commands locally, which is fundamental for low latency and real-time processing in automation tasks. Its control box, equipped with processors and controllers for real-time decision-making, exemplifies its edge capabilities.

The xArm-7 is a 7-*Degrees-of-Freedom* (DoF) robotic arm designed for versatility in industrial automation, research, and education. It features a modular design, high precision, and repeatability, making it ideal for pick-and-place, assembly, testing, and human-robot interaction. With seven rotational joints, it offers extensive motion range and flexibility, allowing complex task execution from various angles. The arm's 3.5 kg payload capacity and 700 mm reach enable it to manipulate a wide array of objects and tools. Additionally, the arm achieves repeatability accuracy of ±0.1 mm, ensuring consistent performance in repetitive tasks, while its maximum joint speed of 180°/s supports efficient and rapid movements. Its modular nature facilitates maintenance and customization, as users can interchange end effectors or integrate sensors and peripherals tailored to specific needs.

Figure 7. The xArm 7 UFactory Robotic Arm (panel A) and the 360° USB Microphone (panel B)

Power Management

A dedicated control box serves as the central hub for power management, communication, and control. It connects to the robot via a specialized communication cable and offers ports for external devices connectivity such as Ethernet, USB, and digital I/O. Safety features include an emergency stop button and a power switch. The control box manages the xArm-7's power supply, ensuring stable and reliable operation. It converts AC input (100-240V, 50/60Hz) to DC power, supplying 48V DC to servo motors and 12V DC to control electronics. It also incorporates protection mechanisms like over-voltage, over-current, and short-circuit protection alongside intelligent power management functions such as soft-start and soft-stop to prevent power surges and ensure smooth operation. The system also monitors the temperature of the power supply and motors, initiating safety shutdowns if temperatures exceed safe limits. Moreover, the xArm-7 is compatible with various control systems, including xArm Studio software, ROS, and programming languages like Python and C++ through SDK packages. In addition, the control box enhances communication between the arm and the control system, facilitating an easy integration into existing setups and custom application development. Safety features like collision detection, force limiting, and emergency stop functionality ensure safe operations in collaborative environments.

These aspects have facilitated the integration of the robot into the current study. The Raspberry Pi is connected to the control box via an Ethernet cable and sends movement commands to the robotic arm using the xArm-Python-SDK. These commands, as established earlier, are mapped to specific user voice inputs but are ultimately decided by the generative AI model. This model has the authority to filter out commands and can also choose not to execute them if the robot reaches its motion constraints.

2.5. OpenAI ChatGPT3.5-Turbo as Generative AI

The project's strategic choice of using OpenAI's ChatGPT is primarily driven by the need for sophisticated natural language understanding and generation capabilities within an interactive robotic control system. The built system showcases how ChatGPT is integrated to process user commands, enabling specific robotic actions and general dialogue interactions.

Advanced Language Processing & Contextual Understanding

ChatGPT excels at understanding and generating natural language, making it an ideal choice for interpreting complex user commands, programming logic, and queries. These features are crucial in applications where users interact with robots through speech, as they enable the robotic system to comprehend instructions that vary in linguistic structure, tonality, and complexity through the AI model.

The algorithm developed during this research configures ChatGPT to handle different contexts by setting up specific modes such as `XARM_MODE`, `DIALOGUE_MODE`, and `DRONE_MODE`. This kind of flexibility allows the AI to tailor its responses based on the operational context and recent discussions, enhancing the user experience and the system's effectiveness. For example, in `XARM_MODE`, ChatGPT focuses on generating clear and direct commands for robotic arm operations without accepting unfamiliar directives or other deviations, whereas in `DIALOGUE_MODE`, it shifts to engaging in general discussions, providing information as requested, based on the vast knowledge of the model used.

Real-time Interaction

Robotic systems, particularly those involved in assembly or assistance that need to perform operations around or in collaboration with humans, must operate in fast to be effective. Integrating AI models that can process natural language quickly allows these systems to understand and execute commands without significant delays. This immediacy ensures that the robots can respond to changes in their environment or to instructions quicker, which is vital when working alongside humans who operate at a natural, often unpredictable pace. This type of integration can help AI-assisted robotic systems understand contexts and instructions better than they currently do, in a more human-like manner. Such features can allow for more intuitive interactions, reducing the learning curve and improving the efficiency of joint operations.

This represents an evolutional use case of advanced human-robot bonding, offering the perspective of better robotic systems compliance, resulting in robots being more aligned with human needs and behaviors, thereby making their integration into Human-Centric environments smoother and more natural.

Development Integration, Scalability & Customization

OpenAI provides a well-documented API that facilitates easy integration of ChatGPT into existing systems. The built system demonstrates how the client model is defined, initialized and used to generate responses to user inputs. This ease of integration reduces development time and complexity, allowing developers to focus on other critical aspects of the system. Furthermore, the capabilities can be extended or customized through different settings and parameters provided by the OpenAI API. This allows the algorithm to be adapted for future enhancements or applications, ensuring the system remains versatile and scalable.

Overall, the choice to use OpenAI ChatGPT in this project is justified by its superior language processing abilities, extensive customization options, and ease of integration with the hardware and the other APIs, all essential for creating an interactive and responsive robotic control system that strives for intuitiveness and efficiency.

2.6. Google Text-to-Speech Engine

The choice of Google Text-to-Speech (TTS) Engine for this project is decided by its proven reliability and extensive language support, making it an ideal solution for interactive systems that require verbal output. Google's TTS technology is known for its high-quality voice synthesizer, closely mimicking human speech patterns, providing a natural and engaging user experience. This factor is essential for this type of application, where user interaction represents a central role, as it enhances the system's accessibility and usability across diverse user groups. Furthermore, Google TTS supports multiple languages and dialects, allowing for scalability in global applications. The ease of integration provided by the *gtts* Python library, which interfaces seamlessly with Google's TTS API, significantly reduces development complexity and accelerates deployment timelines.

In the system architecture, the TTS is used as a precursor of the output, processing the AI-generated text responses via an API. This results in a synthesized voice mp3 audio file that is subsequently played through the speakers using the mpg321 command-line audio player.

2.7. Google Voice Recognition Engine

Sufficient Implementing Google's Voice Recognition Engine through the *SpeechRecognition* library offers several critical benefits for this research. Google's engine is renowned for its accuracy and robust performance in diverse acoustic environments, which are a priority for reliable speech-to-text conversion in real-time applications. The engine's ability to recognize and process various languages and accents enhances its versatility, making it suitable for multi-lingual environments, which are future improvement objectives for this current project.

Additionally, Google's advanced AI and natural language processing algorithms effectively handle colloquial phrases and complex sentence structures, ensuring that user commands are interpreted correctly and efficiently. The integration with *Python's SpeechRecognition* library also provides a straightforward implementation path that complements the project's need for fast development cycles and robust performance.

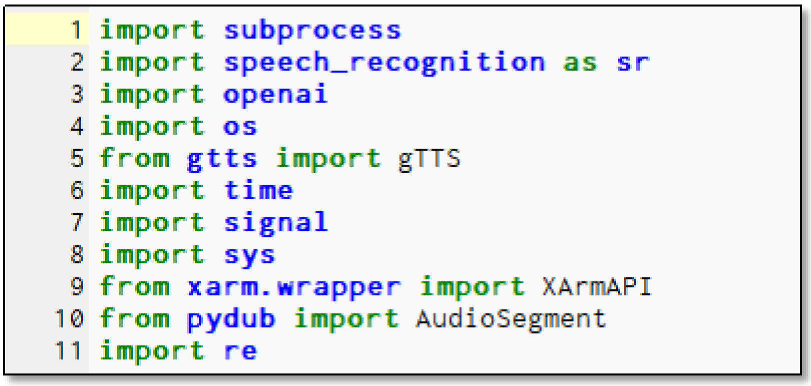
The voice recognition API engine facilitates data exchanges through the system input channel. The USB-connected omnidirectional microphone captures any audio in 3-second recording batches in .wav format. These batches are sent to the *Google SpeechRecognition API*, which returns text transcripts that are further processed and sent to the OpenAI API for AI responses. These technologies support the core functionality of the system and ensure that it remains adaptable and forward-compatible with any future upgrades, emerging needs and technologies.

3. Results & Discussion

The IoT-AI system developed in this research has been evaluated for its operational efficiency and reliability in facilitating human-robot interactions. Following the above methodology, an operator interact with a USB microphone providing audio input which are processed by the Raspberry Pi board. The board is connected to a stereo speaker for mutual conversation and to the Text-to-speech Google API. An ethernet cable provides connection to the robotic arm (Figure 1). Although this development is not designed to generate real-time robot control commands, the latency is sufficiently low to ensure reliable and safe human-robot interaction. The voice input segment of the pipeline, comprising the microphone and the voice-to-text API, effectively records user commands. However, it sometimes requires users to repeat commands due to hardware limitations, variations in voice tonality, accents, or other phonetic challenges. These challenges may also arise if users are uncertain when to start or stop speaking. To address this, integrating a visual signaling system, such as an on/off LED, could help synchronize user interactions with the algorithm more effectively.

Furthermore, the AI has demonstrated its capability to accurately classify responses and generate appropriate commands for the robotic arm. The xArm executes these commands sequentially, ensuring that each movement is carried out precisely and safely. This methodical execution confirmed the robot ability to interact closely with human operators, assisting without compromising safety. Such features make the robot particularly useful in environments where direct human collaboration is necessary, enhancing both the efficiency and safety of operations.

The source algorithm of the developed system can be found on GitHub. An overview of the main set-up is shown in Figure 8.



```
import subprocess
import speech_recognition as sr
import openai
import os
from gtts import gTTS
import time
import signal
import sys
from xarm.wrapper import XArmAPI
from pydub import AudioSegment
import re
```

Figure 8. Initialization of the system where speech-recognition and x-arm robot device are set to mutually interact – details of the overall code are reported at the following link on GitHub

This access allows further insight into the system capabilities and invites collaboration and feedback to refine the system further.

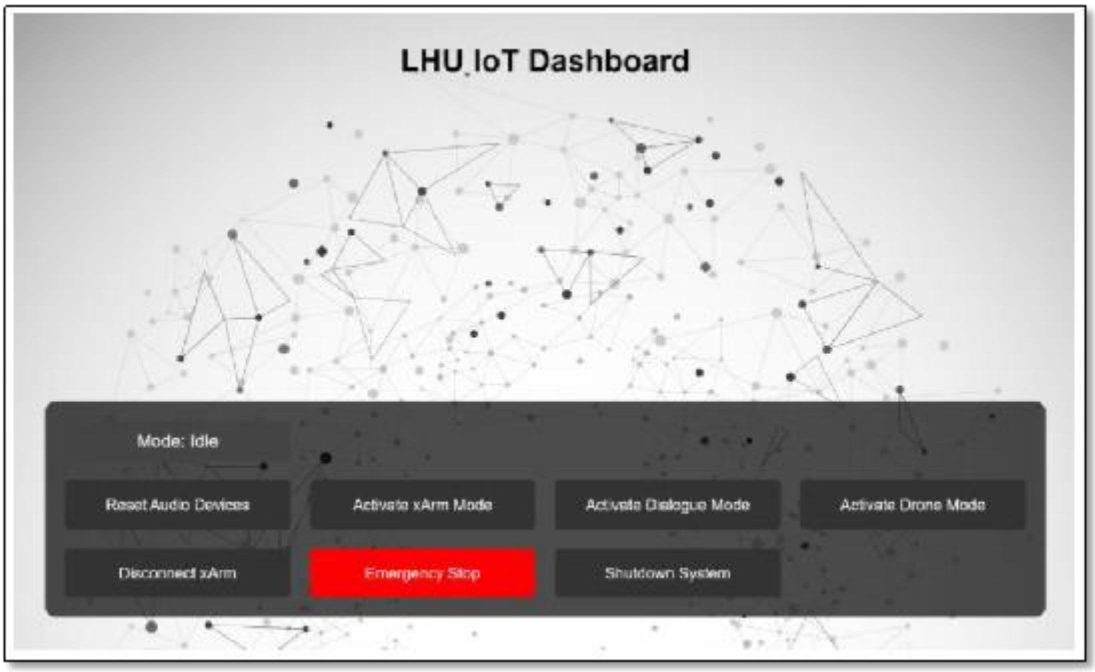


Figure 9. Control Dashboard GUI running on Pi via Flask Server

3.1. System Functionality Overview

This IoT-AI-based human-robot interaction and control system integrates three primary computational structures, streamlining complex tasks through a configuration of hardware, software and data flow interactions:

1. **Main Control Structure - Raspberry Pi 5 Core Unit**

The Raspberry Pi 5 acts as the central control hub. It leverages its hardware capabilities to interface with peripheral devices critical for input and output processes necessary for interaction. There is also a control dashboard available to user on the local network, which contains the essential commands of the system (*Fig. 9*). These peripherals and connection capabilities are:

- **Microphone**: Captures user voice inputs, converting spoken commands into digital data for processing by the Google Voice Recognition API.

- **Stereo Speakers**: Outputs AI-generated responses thought Google Text-to-Speech API, facilitating real-time communication between the user and the system.

- **Internet Connectivity**: A stable Wi-Fi connection is essential for accessing cloud-based API services which handle various stages of data processing and response generation (input speech-to-text, text-to-text AI-response, output text-to-voice).

- **Process Flow**:

  A. **Speech-to-Text**: Utilises the Google Voice-Recognition API through the SpeechRecognition library to transcribe spoken words captured by microphone into text.

B. **AI Response Generation**: The OpenAI API processes the transcribed text, utilising the ChatGPT-3.5-Turbo model. This model operates in three distinct modes:

1. **Xarm Mode**: Generates commands for controlling the robotic arm based on user input.

2. **Dialogue Mode**: Engages in general conversation, providing information based on user prompts input.

3. **Drone Mode**: Still under development – Generates commands for drone control, enhancing the system's applicability to various robotic platforms.

C. **Text-to-Speech**: The Google Text-to-Speech API converts the AI's textual responses back into audible speech, ensuring the communication loop is maintained and user is always aware of the robotic arm incoming motions.

D. **Control Dashboard**: Although still under development, there is a control dashboard accessible to users on the local network, incorporating the essential commands of the system. This dashboard utilizes the Flask library to facilitate straightforward management and operation directly from a web browser (*Fig. 8*).

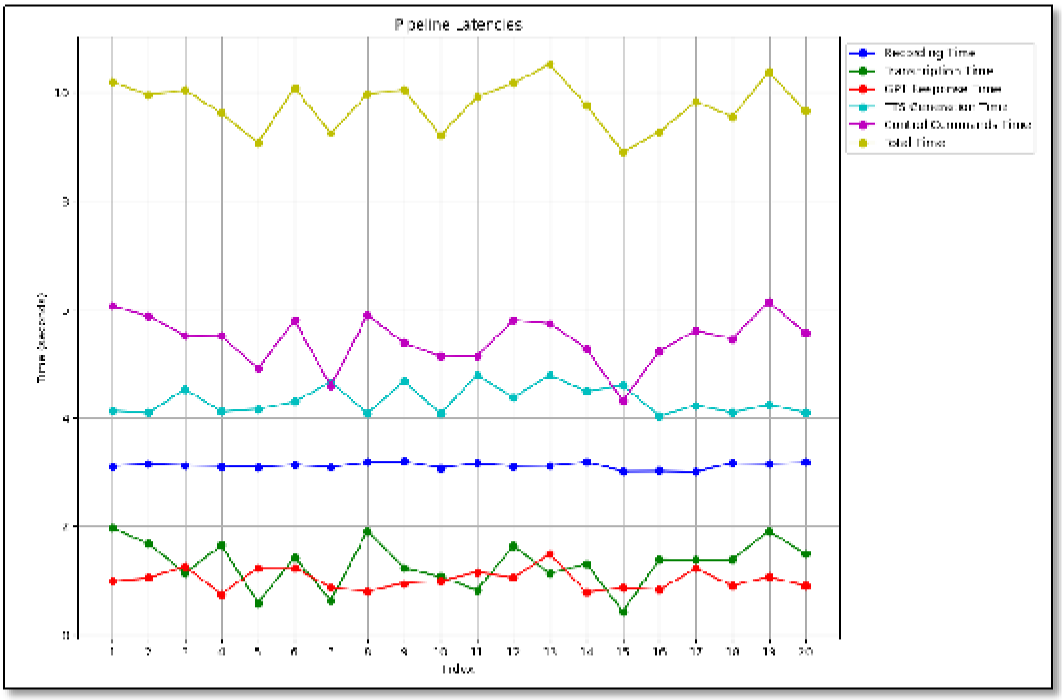


Figure 10. Latencies through the system pipeline

2. **Admin System**

- This overview system focuses on development, updates, and system monitoring, providing necessary oversight and administrative capabilities.

- **Network Security**: Utilizes SSH (Secure Shell) for secure communication between the admin computer and the Raspberry Pi 5, safeguarding system integrity through encrypted connections that require authentication.

3. **Edge Device - Peripheral Control Device: xArm-7 Robotic Arm**

- **Connection**: The xArm-7 is linked to the Raspberry Pi via Ethernet, allowing for robust and reliable command and control signals.

- **Functionality**: Serves as a direct-action endpoint within the system, executing physical tasks based on processed commands from the Raspberry Pi generated thought the interaction pipeline.

This architecture supports the primary goal of creating a responsive and interactive environment but also ensures that the system remains adaptable and scalable. By integrating advanced AI with practical robotic execution and thorough administrative oversight, the system architecture shows that is able to handle complex interactions and perform a wide range of tasks efficiently and securely.

An example of the resulting set of possible commands that the conversational interaction allows between the operator and the robotic arm are: “What’s your name?”, “Zora, do you hear me?” and so on; when interacting with the robot the user can set commands such as “Zora, engage arm” and then can move the arm with commands such as “move right 20” (i.e. move the robotic arm to the right of 20 cm), “move arm left 40”, “move down 10”; following the reception of the command Zora will verbally confirm execution and re-iterate the command reporting explicitly the unit of measurement of the displacement, such as “moving the x-arm right by 20 cm”.

3.2. Latencies Analysis

The system has been tested in laboratory condition and a set of preliminary trials have been performed where the end-user delivered a set of vocal commands to the robotic arm. In xArm mode, the average latency for a robot command to reach the robotic arm is approximately 5.5 seconds. This latency consists of a 3-second audio recording of the user input, followed by approximately 1.5 seconds for the speech-to-text API to return a transcription and about 1 second for generating a response via the AI-GPT model. Figure 10 shows the main pipelines latencies where the *Recording Time*, the *Transcription Time*, *GPT Response* and *TTS Generation Times* and well as the *Control Commands* and *Overall Time* are reported. ON average the system has to comply with 0-1 s of *Recording and Transition Times* followed by a 2-6 s of *Recording*, *TTs* and *Control Commands Times*. Additionally, during server peak hours, the Google APIs (speech-to-text and text-to-speech) may experience significant delays, potentially increasing response times by up to 5 seconds each TTS and STT. Possible solutions to protect the latencies of the system and even improve it could be:

Upgrading to Google paid services: Opting for the paid versions of Google's services could offer stable API responses, with even faster and more consistent processing times.

Exploring alternative technologies: testing other speech recognition and processing services could uncover more efficient options. Technologies such as IBM Watson, Microsoft Azure Speech, and Amazon Transcribe might offer enhanced features or superior performance. Additionally, offline models like eSpeak, Mycroft Mimic, or the Festival Speech Synthesis System could provide enhanced privacy, reduced latency, and improved performance.

Optimization of the Recording Process: The current recording process is not fully optimized and improving it could significantly reduce the average 3-second recording time. Additionally, executing the recording process in a separate thread, upgrading to superior hardware, or utilizing more advanced sound-capturing libraries could refine these latency results.

3.3. Future Objectives

Here are some additional future improvements for enhancing human-machine and human-robot interaction of the system:

- *Verbal cues for recording*: Integrate start and end recording verbal cues into dialogue mode to improve user interaction.

- *Advanced robotic control*: Expand the robotic control mode to record each motion, allowing for replication of sequences, including in reverse, and the ability to combine these sequences to create complex motion patterns. This feature can give users greater freedom to define and automate robot actions.

- *3D vision integration*: Test the benefits of integrating a 3D vision system with the conversational AI feature to enhance situational awareness and interaction capabilities.

- *Expansion to other devices*: Expand the IoT-AI control system to include other robotic devices, enlarging its scope and applications.

4. Conclusions

This research successfully demonstrates the feasibility and potential of integrating conversational AI, specifically OpenAI ChatGPT, with IoT devices to create a more intuitive and efficient human-robot interaction system which further highlight the importance of Human-Centric design nowadays [17-20]. The framework effectively translates user voice commands into robotic actions by incorporating the powerful Raspberry Pi 5 as the central control unit and leveraging Google speech recognition and text-to-speech engines. The integration of the xArm-7 robotic arm exemplifies the development capabilities, showcasing how complex robotic tasks can be executed through natural language commands. Given the stage of the prototype we have design and implemented a set of preliminary tests in laboratory condition: nevertheless, a proper validation of the system and further extension of these tests with a comparison vs other benchmarks would be needed. In this context it would also be beneficial to validate the robustness of the system under different conditions, namely in the daily (and industrial) life context where the voice commands may interact and mix with other disturbances and noises. There is also room for testing the efficiency of the system under different languages, tones and accents, since a robust response and clear understanding vs different users is going to be compulsory for the success of such an approach at industrial level and in whatever context the system could be used (see for example vocal commands when using a robot for surgical procedure).

The modular design architecture ensures flexibility and scalability, making it adaptable for control over a wide range of IoT or edge devices such as drones, smart home appliances, or various types of assembly and assistive robots. Such adaptability highlights the potential to enhance human-robot interaction across numerous domains, emphasizing safety, responsiveness and reliability in operations, which are critical when robots perform around or in collaboration with humans. Rapid natural language processing by AI models allows these robots to comprehend and act on instructions without delay, which is essential for maintaining pace with the unpredictable rhythm of human activities.

Despite its promising performance, the system does face challenges, such as latency during peak server times and occasional inaccuracies in voice recognition. Future research will address these issues by exploring alternative speech recognition technologies, enhancing the recording process, and integrating other feedback mechanisms to improve system robustness and reliability. In this context it is also of interest considering other technologies which could be integrated and combined with the conversational interactions, such as, for example, human movement detection and use of wearable sensors embedded on the end-user body which could further enhance real-time (and intuitive) interaction between the operator and the machine (i.e. the robot in the current scenario) [19-23]. In this context, one of the important characteristics of the proposed system is related to its latency, as we reported along the paper. However, there are other aspects which inherently involve such latency and related to safety, especially in applications where the robotic arm (or another device activated by voice) is interacting with the human body and the actions of the voice operator could affect and damage the interacting body.

In conclusion, this research intends to contribute to the field of human-robot interaction, providing a robust framework for integrating IoT devices and conversational AI with robotic systems. It promotes more natural and intuitive interactions, reducing the learning curve and improving the efficiency of collaborative operations. This development describes an evolutional use case of advanced human-robot bonding, offering the perspective of better robotic systems compliance, resulting in robots being more aligned with human needs and behaviors, thereby making their integration Human-Centric, smoother and more natural.

Supplementary Materials

The source algorithm of the developed system can be found at the following link on GitHub.

Author Contributions

Conceptualization, D.M.; methodology, D.M.; software, D.M.; validation, D.M.; writing—original draft preparation, D.M.; writing—review and editing, E.S.; supervision, H.A.. All authors have read and agreed to the published version of the manuscript.

Funding

This work received no external funding.

Institutional Review Board Statement

Not applicable.

Informed Consent Statement

Not applicable.

Data Availability Statement

No applicable.

Acknowledgments

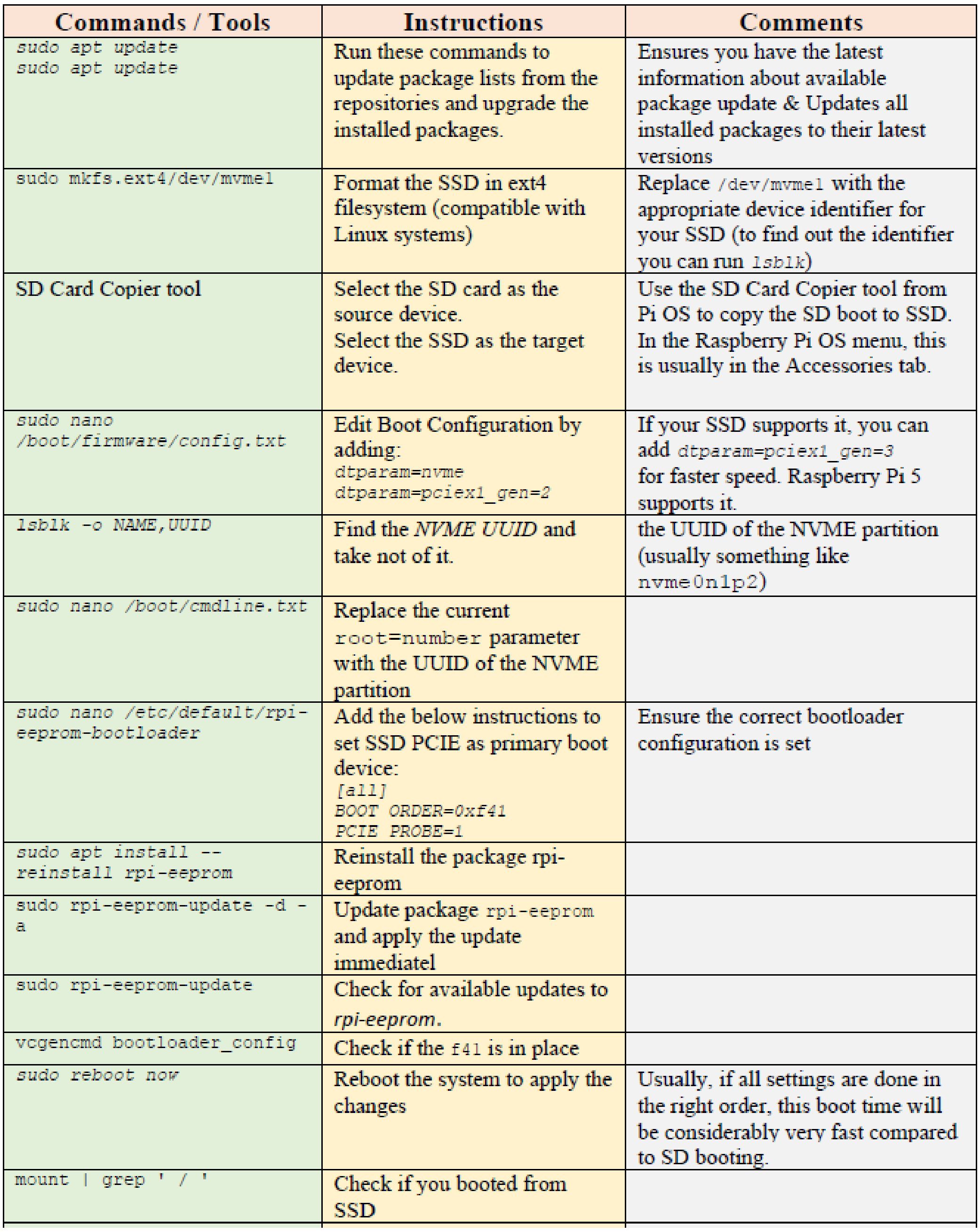
This work was presented in coursework form in fulfilment of the requirements for the MEng Robotics Engineering for the student D Manolescu under the supervision of Dr Hamzah AlZu'bi from the AI Lab, School of Computer Science and the Environment, Liverpool Hope University.

Conflicts of Interest

The authors declare no conflict of interest.

Appendix 1 – Instructions to Boot Pi from SSD

Below are the main steps and instructions in order to setup the Raspberry Pi device and migrate the booting of the board from the SD card into the local SSD.



Commands / Tools	Instructions	Comments
`sudo apt update` `sudo apt update`	Run these commands to update package lists from the repositories and upgrade the installed packages.	Ensures you have the latest information about available package update & Updates all installed packages to their latest versions
`sudo mkfs.ext4/dev/mvme1`	Format the SSD in ext4 filesystem (compatible with Linux systems)	Replace `/dev/mvme1` with the appropriate device identifier for your SSD (to find out the identifier you can run `lsblk`)
SD Card Copier tool	Select the SD card as the source device. Select the SSD as the target device.	Use the SD Card Copier tool from Pi OS to copy the SD boot to SSD. In the Raspberry Pi OS menu, this is usually in the Accessories tab.

`sudo nano /boot/firmware/config.txt`	Edit Boot Configuration by adding:
	`dtparam=nvme`
	`dtparam=pciex1_gen=2`
		If your SSD supports it, you can add `dtparam=pciex1_gen=3` for faster speed. Raspberry Pi 5 supports it.

`lsblk -o NAME,UUID`	Find the *NVME UUID* and take not of it.	the UUID of the NVME partition (usually something like `nvme0n1p2`)
`sudo nano /boot/cmdline.txt`	Replace the current `root=number` parameter with the UUID of the NVME partition	

`sudo nano /etc/default/rpi-eeprom-bootloader`	Add the below instructions to set SSD PCIE as primary boot device:
	`[all]`
	`BOOT_ORDER=0xf41`
	`PCIE_PROBE=1`
		Ensure the correct bootloader configuration is set

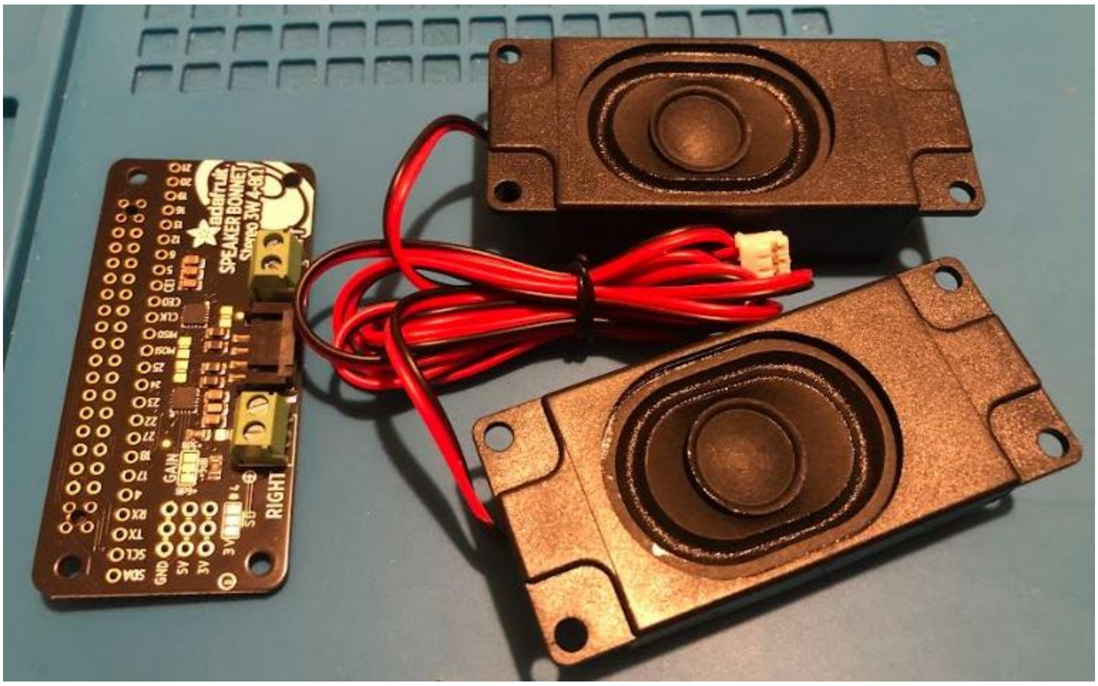
`sudo apt install --reinstall rpi-eeprom`	Reinstall the package rpi-eeprom	
`sudo rpi-eeprom-update -d -a`	Update package `rpi-eeprom` and apply the update immediatel	
`sudo rpi-eeprom-update`	Check for available updates to *rpi-eeprom*.	
`vcgencmd bootloader_config`	Check if the `f41` is in place	
`sudo reboot now`	Reboot the system to apply the changes	Usually, if all settings are done in the right order, this boot time will be considerably very fast compared to SD booting.
`mount | grep ' / '`	Check if you booted from SSD	

Table 1. Booting of the board from the SD card into the local SSD

Appendix 2 – Assembly and integration of the parts

Below are reported some details of the main parts of the system (see also Fig. 6)

Figure 11. The The main parts of the system, including the power supply, speakers and processing unit (namely the Raspberry PI)



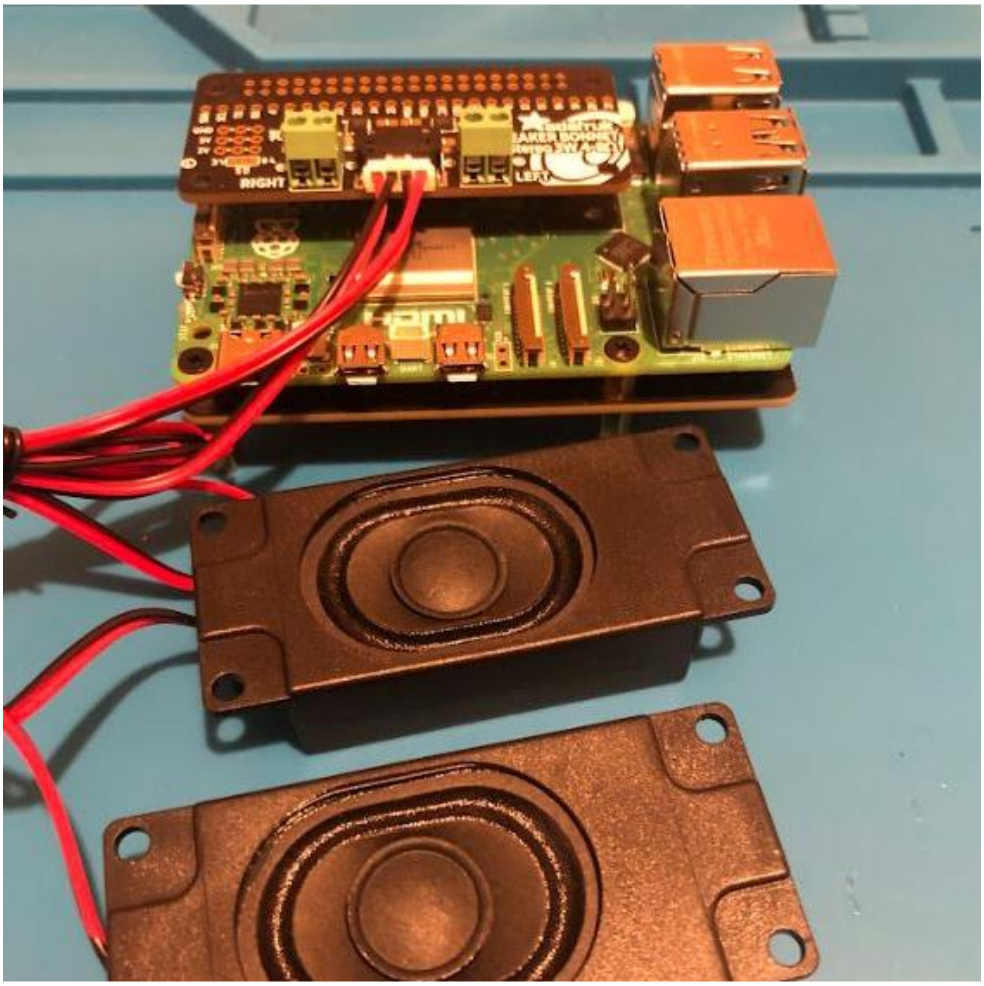


Figure 12. The Adafruit Speaker Bonnet & 3W Stereo Speakers (top panel) and their assembly (bottom panel)

References

1. Radford, A., Narasimhan, K., Salimans, T. & Sutskever, I., 2018. Improving Language Understanding by Generative Pre-Training. [Online] Available at: https://s3-us-west-2.amazonaws.com/openai-assets/research-covers/language-unsupervised/language_understanding_paper.pdf [Accessed 2024].
2. Brown, T. et al., 2020. Language Models are Few-Shot Learners. [Online] Available at: https://arxiv.org/pdf/2005.14165 [Accessed 2024].
3. Gubbi, J., Buyya, R., Marusic, S. & Palaniswami, M., 2013. Internet of Things (IoT): A vision, architectural elements, and future directions. [Online] Available at: https://doi.org/10.1016/j.future.2013.01.010 [Accessed 2024].
4. Al-Fuqaha, A., Guizani, M., Mohammadi, M. & al, e., 2015. Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications. [Online] Available at: https://doi.org/10.1109/COMST.2015.2444095 [Accessed 2024].
5. Piyare, R. & Tazil, M., 2011. Bluetooth Based Smart Automation System Using Android. [Online] Available at: http://dx.doi.org/10.1109/ISCE.2011.5973811 [Accessed 2024].
6. Becerik-Gerber, B. et al., 2022. The field of human building interaction for convergent research and innovation for intelligent built environments. [Online] Available at: https://www.nature.com/articles/s41598-022-25047-y [Accessed 2024].
7. Masreliez, L., 2021. THE FRAGMENTED SMART HOME: A comprehensive analysis of available interoperable solutions to connect wireless smart home communications. [Online] Available at: https://www.diva-portal.org/smash/get/diva2:1584387/FULLTEXT01.pdf [Accessed 2024].
8. Capgemini, R.-I., 2019. How organizations and consumers are embracing voice and chat assistants. [Online] Available at: https://www.capgemini.com/wp-content/uploads/2019/09/Report-%E2%80%93-Conversational-Interfaces_Web-Final.pdf [Accessed 2024].
9. RaspberryPi, F., 2024. 2024. [Online] Available at: https://forums.raspberrypi.com/viewforum.php?f=15
10. OpenAI, 2023. GPT-3.5 Turbo fine-tuning and API updates. [Online] Available at: https://openai.com/index/gpt-3-5-turbo-fine-tuning-and-api-updates/ [Accessed 2024].

11. Liu, D. & Cao, J., 2022. Determinants of Collaborative Robots Innovation Adoption in Small and Medium-Sized Enterprises: An Empirical Study in China. [Online] Available at: http://dx.doi.org/10.3390/app121910085 [Accessed 2024].
12. Guo, X., Shen, Z., Zhang, Y. & Wu, T., 2019. Review on the Application of Artificial Intelligence in Smart Homes. [Online] Available at: https://doi.org/10.3390/smartcities2030025 [Accessed 2024].
13. Thakare, V., Khire, G. & Kumbhar, M., 2022. Artificial Intelligence (AI) and Internet of Things (IoT) in Healthcare: Opportunities and Challenges. [Online] Available at: https://iopscience.iop.org/article/10.1149/10701.7941ecst/pdf [Accessed 2024].
14. Roller, S. et al., 2020. Recipes for building an open-domain chatbot. [Online] Available at: https://arxiv.org/abs/2004.13637v2 [Accessed 2024].
15. EL Secco, Y Noh, Human-like Robotic Hands for Biomedical Applications and Beyond, Frontiers in Robotics & AI, Biomedical Robotics
16. D Bell, EL Secco, Design of a 3D-Printed Accessible and Affordable Robotic Arm and a User-Friendly Graphical User Interface, Congress on Intelligent Systems, 195-205
17. M Innes, EL Secco, An Understanding of How Technology Can Assist in the Epidemic of Medicine Nonadherence with the Development of a Medicine Dispenser, European Journal of Applied Sciences 11 (3), 522-550
18. J Hutton, EL Secco, Development of an interface for real time control of a dexterous robotic hand, using MYO muscle sensor, Acta Scientific Computer Sciences, 2023
19. Bilawal Latif, N Buckley, EL Secco, Hand Gesture & Human-Drone Interaction, Intelligent Systems Conference (IntelliSys), 3, 299-308, 2022, DOI: 10.1007/978-3-031-16075-2
20. D Manolescu, B Mutinda, EL Secco, Human–robot interaction via wearable device - A wireless glove system for remote control of 7-DoF robotic arm, Academia Engineering, 2024, *in press*
21. M Mowlai, M Mahdavimanshadi, I Sayyadzadeh Adapting Transformer-Based Multi-Style Networks for Human Pose Prediction with a Custom Data Pipeline in Industrial Human-Robot Collaboration, 2024 Systems and Information Engineering Design Symposium (SIEDS), 274-279
22. EL Secco, DD McHugh, N Buckley, A CNN-based Computer Vision Interface for Prosthetics' application, EAI MobiHealth 2021 - 10th EAI International Conference on Wireless Mobile Communication and Healthcare, 41-59, 2022
23. TS Chu, AY Chua, EL Secco, A Study on Neuro Fuzzy Algorithm Implementation on BCI-UAV Control Systems, ASEAN Engineering Journal (AEJ), 12, 4, 75-81, 2022