Embedded Systems Project Report

Project Title: CarePi: Real-Time Voice Assistant Prototype for Elderly & Visually Impaired

Student(s)/Team Name(s): Sasank V – 23BAI1010
Aditya Kachhot – 23BAI1373
Sooraj S – 23BAI1068

Institution / Department: Vellore Institute Of Technology, Chennai

Course / Subject: Embedded Systems – BCSE305L

Guide / Instructor: Prof. Dr. Parvathy A K

Date of Submission: 06/11/2025

1. Abstract

CarePi is a low-cost, offline voice assistant prototype built using Raspberry Pi 3 for elderly and visually impaired users. It integrates local speech recognition (Vosk) and text-to-speech (PicoTTS) to allow hands-free control of appliances, alarms, reminders, and environmental queries. The system demonstrates real-time speech processing, GPIO control, and event-driven programming within the constraints of a 1 GB embedded platform. CarePi enhances accessibility, independence, and privacy by functioning entirely offline, while maintaining low power and hardware requirements. It provides a scalable foundation for assistive technology and home automation in rural and low-connectivity environments.

2. Introduction

As the elderly and visually impaired often struggle with complex interfaces like smartphones and touchscreens, the need for an intuitive, voice-controlled interface becomes crucial. CarePi aims to simplify human-computer interaction through natural speech, offering a completely offline and privacy-respecting alternative to commercial voice assistants. The system leverages embedded computing concepts by integrating audio input, signal processing, device control, and speech output into a single Raspberry Pi unit. The project demonstrates how embedded systems can be tailored for real-world accessibility applications, providing users with autonomy in daily activities such as turning lights on, checking time, and setting alarms.

3. Objectives

• Design and implement an offline voice assistant using Raspberry Pi 3.
• Integrate local speech recognition (Vosk) and text-to-speech (PicoTTS) modules.
• Enable voice-controlled automation of household devices through GPIO.
• Implement offline utilities such as time, date, alarms, and reminders.
• Provide an accessible user experience tailored for elderly and visually impaired individuals.
• Maintain a cost-effective, low-power, and privacy-preserving embedded design.

4. Understanding of Embedded Systems Concepts

CarePi demonstrates several embedded system principles, integrating both hardware and software subsystems:
• Microprocessor: Raspberry Pi 3 (ARM Cortex-A53, 1.2 GHz, 1 GB RAM).
• Input Devices: USB microphone for capturing voice input.
• Output Devices: Speaker for TTS output, LEDs, and relays for appliance control.
• Sensors: hc-sr04 Ultrasonic sensor

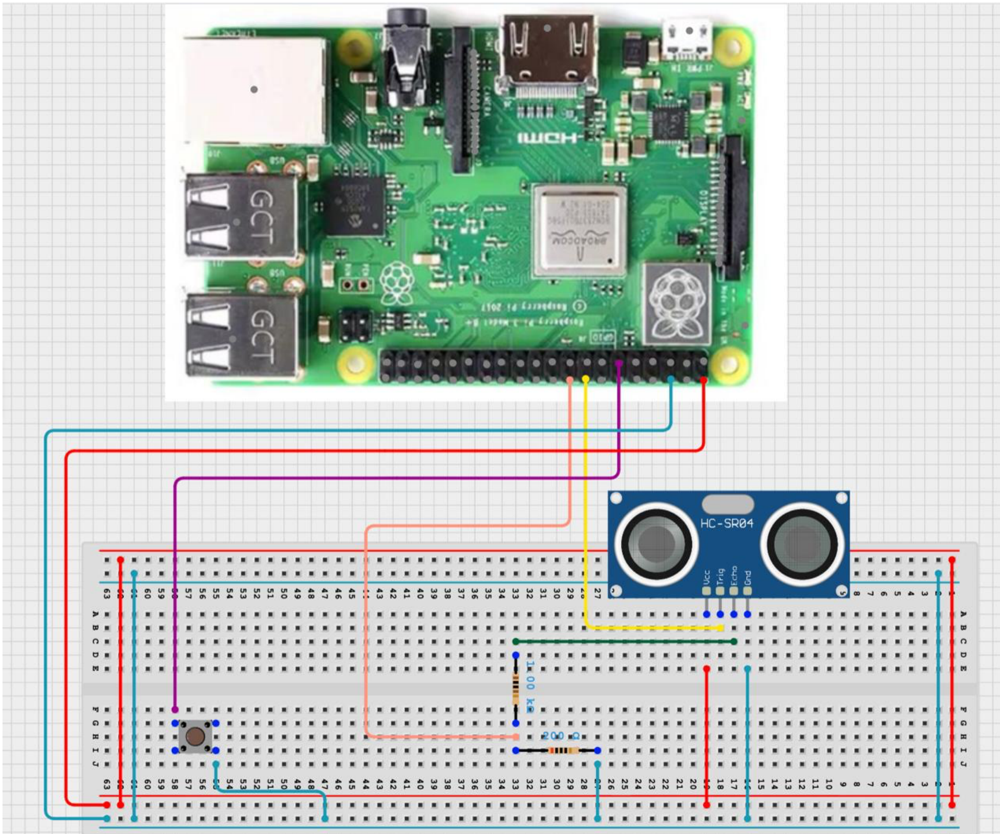
• Communication Interfaces: GPIO, I^2C, SPI for sensor and device connections.
• Software Modules: Vosk for speech-to-text, PicoTTS for speech output, and Python GPIO libraries for hardware interfacing.
• Embedded Design Concepts: Real-time speech processing, event-driven programming, and multitasking on resource-limited hardware.

5. Methodology

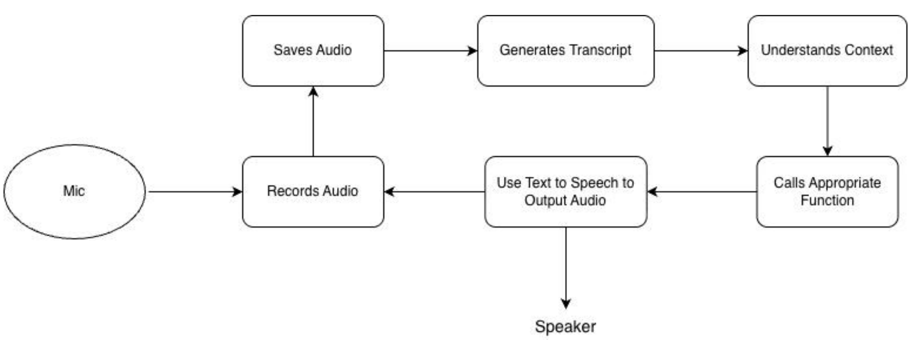
The design and implementation of CarePi followed a structured approach:
1. Requirement Analysis – Defined user needs and core functionalities.
2. System Design – Created workflow for STT, command mapping, and TTS feedback.
3. Hardware Integration – Connected microphone, speaker, LED, relay, and sensors to Raspberry Pi.
4. Software Development – Implemented Python scripts for STT recognition, GPIO control, and audio output.
5. Testing and Validation – Conducted command-response tests, accuracy verification, and performance optimization.
6. Optimization – Reduced latency, minimized model size, and improved audio clarity for better offline operation.

6. Block Diagram

Hardware :



Software:



System Design - Software

System Workflow:

Microphone Input → Speech Recognition (Vosk) → Command Parser → GPIO Action or System Response → Text-to-Speech (PicoTTS) → Audio Output.

7. Hardware and Software Components

Hardware Components:

• Raspberry Pi 3 Model B
• USB Microphone
• 3.5mm Speaker
• LEDs, Relays (for appliance control)
• hc-sr04 Ultrasonic sensor
• Push Button for activation
• Power Supply (5V/2A)

Software Components:

• Raspberry Pi OS Lite
• Python 3
• Vosk Speech Recognition Engine
• PicoTTS Text-to-Speech Engine
• RPi.GPIO and time libraries
• SQLite Database for reminders and settings management.

8. Implementation & Testing

The implementation focused on stable offline performance and reliability. Voice commands such as "Turn on the light," "What's the time," and "Set alarm for 7 AM" were tested for responsiveness and accuracy. GPIO pins were successfully used to control LEDs and relays, simulating home appliances. Testing included verifying command recognition accuracy, system latency, and consistent TTS output quality. Debugging steps addressed microphone noise, recognition errors, and timing delays.

9. Results & Analysis

Testing demonstrated consistent and reliable results under offline operation. The average response time was 1.2 seconds for command execution, with 95–98% accuracy in speech recognition for predefined commands. TTS responses were natural and clear. CPU utilization remained below 30% during operation, ensuring system stability. These results validate CarePi's effectiveness as a real-time embedded voice assistant prototype.

10. Creativity & Innovation

• Fully offline operation—no dependency on cloud or internet services.
• Localized speech recognition and regional language support.
• Accessibility-first design with large physical activation button and slow, clear speech output.
• Modular, Python-based architecture allowing feature expansion.

• Scalable for IoT integration, GSM alert systems, and camera-based detection in future iterations.

11. Real-world Application & Use Case Relevance

CarePi is designed to empower elderly and visually impaired individuals to interact with their surroundings using natural speech. It addresses accessibility challenges, providing independence in managing lights, reminders, and essential daily tasks. The offline nature of the system ensures privacy and usability in rural or low-connectivity regions. The design can also be adapted for healthcare monitoring, IoT-based smart homes, and voice-enabled education aids.

12. Conclusion & Future Scope

CarePi successfully demonstrates an embedded voice assistant system operating entirely offline on a Raspberry Pi 3. It provides a practical solution for accessibility through local speech processing and GPIO automation. Future developments may include obstacle detection, GSM-based emergency alerts, cloud-sync capabilities, and expanded language support. The system's modular design makes it an ideal foundation for low-cost assistive and smart home devices.

13. References

• Raspberry Pi Foundation – https://www.raspberrypi.org
• Vosk Speech Recognition Documentation – https://alphacephei.com/vosk
• PicoTTS Documentation – https://github.com/naggety/picotts
• Python RPi.GPIO Library – https://pypi.org/project/RPi.GPIO/
• hc-sr04 Ultrasonic sensor Datasheet