Robocar VIoT: A Voice and IoT Controlled Robotic Car

By

Naimur Rahman
Roll: 1907031

&

Md. Abu Saeed
Roll: 1907057

Supervisor:

Dr. Md. Sheikh Sadi

Professor

Dept. of Computer Science and Engineering

Khulna University of Engineering & Technology

____
Signature

Department of Computer Science and Engineering

Khulna University of Engineering & Technology

Khulna 9203, Bangladesh

November 2023

Acknowledgements

First of all, we express our gratitude to the Almighty Allah for endowing us with the skills and capabilities necessary for undertaking and completing this project. We are thankful for the blessings that have enabled us to contribute to the field of IoT and robotics.

We extend our deepest appreciation to Dr. Md. Sheikh Sadi, Professor of the Department of Computer Science and Engineering and the dedicated supervisor of this project. His profound expertise and steadfast guidance have been the bedrock of my journey in developing the project. His continuous support, invaluable insights, and encouraging feedback have played a pivotal role in advancing this project to its current state. His unwavering commitment to excellence, combined with his constructive criticism, has significantly shaped the trajectory of this project.

We are truly grateful for the enduring support and scholarly mentorship provided by Dr. Md. Sheikh Sadi sir. Without his enthusiastic motivation and steadfast encouragement, the successful completion of this project would not have been possible. His commitment to fostering a rich learning environment has been a source of inspiration throughout this project journey.

This project has been a collaborative effort, and we acknowledge the contributions of all those who have supported and inspired us throughout this endeavor.

Authors

Naimur Rahman

Md. Abu Saeed

Abstract

Robocar VIoT stands at the forefront of robotic innovation, presenting a fusion of Voice, IoT, and Bluetooth technologies in the creation of a Voice and IoT Controlled Robotic Car. This project redefines the landscape of remote-controlled robotics by integrating advanced features that serve to a diverse set of user preferences. Powered by Rock Pi 3a and an Android app, users can exercise dynamic control over the robotic car using Bengali voice commands transmitted through IoT protocols, ensuring seamless communication over extended distances and through Bluetooth for short range communications. Through IoT and Bluetooth connectivity, this project introduces a novel control paradigm by incorporating both Bengali voice commands and remote-controller style controls through an Android app. The dual-mode control mechanism of this project offers users unparalleled flexibility, allowing them to steer the robotic car seamlessly using the Android app. Users can issue Bengali commands effortlessly through voice interactions. The Bengali voice system uses Google Speech Recognition API to convert the Bengali voice input to a set of strings. Then the set of strings are decoded into a set of integer values and are passed into the WebSocket database from where the micro-controller can read the data and execute it accordingly. Similarly, the remote-controller inputs are decoded into integer values and read write operations are done between the database and the micro-controller. The comprehensive hardware setup includes Rock Pi 3a microcontroller, motor driver L293D, and some basic components, creating an interactive and adaptive robotic system. Robocar VIoT not only signifies a paradigm shift in human-robot interaction but also sets a new standard for user-friendly control interfaces. By embracing both Bengali voice and remote-controller style controls, this project addresses a spectrum of user preferences, making the robotic platform accessible and versatile. The integration of Bluetooth ensures efficient close-range control, expanding the utility of Robocar VIoT across various applications. This project marks a milestone in robotics, showcasing the potential of integrating diverse control mechanisms for enhanced user experience and application adaptability.

Contents

List of Tables

List of Figures

1 Introduction

The "Robocar VIoT: A Voice and IoT Controlled Robotic Car" project revolutionizes robotic control through a unified platform, seamlessly integrating Bengali voice commands and remote-controller style controls. With a focus on cultural sensitivity, the system employs IoT for long-range interactions and Bluetooth for short-range precision. Multimodal robotics, including advanced voice recognition, enhances user experience. The project not only sets new standards for human-robot interaction but also explores future studies, such as localization, autonomous navigation, and sustainability, promising a dynamic and inclusive technological landscape.

1.1 Background

In the realm of robotics and smart devices, achieving seamless and intuitive control mechanisms is essential. Traditional robotic control systems often face challenges in providing versatile and user-friendly interfaces for interaction. With the surge in Internet of Things (IoT) and voice recognition technologies, there exists an opportunity to revolutionize how users engage with robotic systems. However, current robotic control applications often lack integration, presenting obstacles such as limited control options, fragmented functionalities, and a gap in user experience. Also, there are very few robotic control systems that seamlessly integrate Bengali Language control facilities. Recognizing these challenges, our project aims to develop a cutting-edge solution, "Robocar VIoT," which stands at the intersection of Voice, IoT, and Bluetooth technologies. The goal is to create a unified platform that not only empowers users with Bengali voice command capabilities but also incorporates remote-controller style controls through an Android app for both long-range and close-range control and interaction with the robotic car. This initiative seeks to redefine the standards of human-robot interaction by providing a comprehensive and adaptable control interface, addressing the existing limitations in traditional robotic control systems and opening up new possibilities for enhanced user experience and application versatility.

1.2 Objectives

The primary goal of the Robocar VIoT project is to push the boundaries of traditional robotic control systems, offering users a revolutionary and adaptable platform that redefines human-robot interaction and sets new standards in the field of robotics. To fulfil this goal, this project sets a few objectives which are:

- ✓ to seamlessly integrate Bengali voice commands and remote-controller style controls, offering users flexible robotic car control,
- ✓ to prioritize user-friendliness, providing an adaptable interface for both long-range IoT and short-range Bluetooth interactions,
- ✓ to enable precise control via Bluetooth, ensuring direct and responsive user control through the Android app.

1.3 Scope

The scope of the "Robocar VIoT: A Voice, IoT, and Bluetooth Controlled Robotic Car" project is expansive, aiming to redefine the landscape of robotic control and human-robot interaction. This project seeks to develop a unified platform that seamlessly integrates Bengali voice commands and remote-controller style controls through an Android app. Users will have the flexibility to choose their preferred control method, allowing for versatile and intuitive interactions with the robotic car. For precise and nuanced control within a specific short-range distance, Bluetooth interactions are implemented through the Android app, ensuring responsive control within a range of approximately 20 meters depending on the Bluetooth version of the Android phone from where the controlling app will be used. The project further explores long-range IoT control capabilities, enabling users to command the robotic car over extended distances, acknowledging potential delays inherent in long-range communication. Cultural sensitivity is a key focus, with the implementation of Bengali voice commands to cater specifically to Bengali-speaking users. The addition of linguistic interactions, the exploration of innovative multimodal robotics, and the pursuit of application versatility round out the ambitious scope of the Robocar VIoT project. The main goal is to provide users with a revolutionary and adaptable platform that sets new standards for human-robot interaction, incorporating both technological innovation and cultural inclusivity.

1.4 Unfamiliarity of the problem

Within the realm of robotic control applications, diverse platforms cater to various control methods, each with distinct purposes. Navigating these platforms poses challenges, as they often lack a unified solution. The "Robocar VIoT: A Voice and IoT Controlled Robotic Car" project addresses this unfamiliarity with a multifaceted approach:

1.4.1 Holistic control integration: Unlike existing platforms, this project seamlessly incorporates Bengali voice commands and remote-controller style controls through an Android app, presenting users with a unified and intuitive control experience.

1.4.2 Adaptable control options: Acknowledging user preferences, the platform offers a choice between voice commands and remote-controller style controls, introducing adaptability and personalization in robotic control.

1.4.3 Boundary-pushing technologies: By exploring both long-range IoT control and short-range Bluetooth interactions, the project ventures into uncharted territories, enabling dynamic control over various distances.

1.4.4 Cultural sensitivity: Introducing Bengali voice commands ensures cultural inclusivity, addressing the unfamiliarity of control interfaces for Bengali-speaking users and fostering a more inclusive interaction.

The Robocar VIoT project endeavors to bridge the gap in unfamiliar robotic control systems, introducing a pioneering solution that not only tackles existing challenges but also sets new standards for adaptability and user-centric control interfaces.

1.5 Project planning

The Robocar VIoT project follows a systematic approach encompassing various stages to ensure a comprehensive development process. The project plan includes:

1.5.1 Research and requirements: In-depth research to understand the intricacies of robotic control and user preferences. Gathering requirements to inform the design and development phases.

1.5.2 Design: Crafting a user-friendly interface that seamlessly integrates Bengali voice commands and remote-controller style controls through the Android app.

1.5.3 Prototyping: Developing initial prototypes to visualize the user interaction and control features, allowing for early refinement.

1.5.4 **Development:** Implementing the core functionalities of the Robocar VIoT platform, including the integration of long-range IoT control, short-range Bluetooth interactions, and Bengali voice command capabilities.

1.5.5 **Testing:** Rigorous testing to ensure the robustness and functionality of the platform, addressing any issues and optimizing the performance.

1.5.6 **Quality assurance:** Conducting comprehensive quality assurance checks to guarantee a seamless and error-free user experience.

In Figure 1.1, a Gantt chart illustrates the project timeline, outlining key milestones from the research and design phases to prototyping, development, testing, and quality assurance. This schedule serves as a preliminary estimate, indicating the time and efforts allocated to each stage. The goal is to ensure a well-organized and efficient progression toward the deployment and completion of the Robocar VIoT project.

Figure-1.1: Gantt Chart for project planning & task distribution

2 Related Work

The exploration encompasses a diverse range of related works, showcasing the progressive developments that lay the foundation for the "Robocar VIoT: A Voice and IoT Controlled Robotic Car" project.

2.1 Existing solutions

There are some related research papers which work on IoT or voice control mechanisms implemented via different micro-controllers and systems. These systems have their own different approaches and building block for a detailed research and descriptive analysis of using these specific technologies. A basic comparison of these researches is noted in Table 2.1:

Table 2.1: Related work in IoT and voice controlled robotic vehicles

Serial No.	Title	Authors	Motivation	Features
1	Surveillance Robocar Using IoT and Blynk App [1]	N. Kumar, B. Dey, C. Chetri, A. Biswas	Real-time application of an unmanned vehicle for surveillance in harsh terrains for military purposes using IoT	Robocar with a pick-and-place robotic arm controlled by the Blynk app, capable of picking and placing objects, live monitoring, off-road capability
2	IoT Based Voice Controlled Autonomous Robotic Vehicle Through Google Assistant [3]	M. Gupta, R. Kumar, R. K. Chaudhary, J. Kumari	Voice-controlled automated vehicle utilizing Arduino IDE, Google Assistant, and cloud computing	Movement control (forward, backward, left, right, stop) through voice commands

3	Voice Controlled Robotic Car Using Mobile Application [4]	S. Chakraborty, N. De, D. Marak, M. Borah, S. Paul, V. Majhi	Human Robotic Interface (HRI) using voice commands through a mobile application	Overcoming problems of manual wheelchairs for the physically handicapped
4	Implementation of Human Voice Controlled Robotic Car [5]	R. L. Khan, D. Priyanshu, F. S. Alsulaiman	Recognizes human voice instructions to operate a voice-controlled car using an Android app and UART protocol	Basic actions like going ahead, turning left or right controlled by human voice

2.2 Solutions to mitigate the gap in existing works

There are some significant stages that set our project at a better level than the other related works. These are mostly clarified below:

2.2.1 **Integrated IoT and voice control:** Seamlessly combining IoT and voice control, this project presents a holistic approach to robotic system control, distinguishing itself in the field.

2.2.2 **Bengali Language integration:** Having the voice interaction be through Bengali language is a key benefit for our project as it opens up to more possibilities and appealing to more people, especially Bengali speaking people.

2.2.3 **Using more advanced microcontroller:** In this project a more recent and more highly powerful microcontroller is used as it leaves with more future feature integrations and room for more research implementations.

2.2.4 **Adding Bluetooth for shorter range communication:** In the shorter ranges, using Bluetooth gives it a more responsive and precise control over the robot compared to always trying to use IoT systems which can be a great optimization.

2.2.5 Transparent status and error passing to user: The user can easily see and find out if an error has occurred in the communication to the micro-controller of the hardware. Also, the user can view the status and communications from the micro-controller to the app and thus be sure that the provided commands have been executed on the hardware system properly.

Thus, it can be said that the Robocar VIoT has more functionalities and is more improved version from the previous integrations of any similar type of research or project.

3 System Design

The system is designed into separate parts that works together as a complete system environment. These parts or components each serve their own purpose to the system and eventually as a whole creates the desired system.

3.1 Analysis of the system

The system works in a comprehensive manner. The app, the communication protocol and the hardware elements all serve their own distinct purpose. The system workflow of the project is intricately designed to ensure a seamless and responsive user experience. The process begins with users initiating control through the Android app, choosing between Bengali voice commands and remote-controller inputs. For long-range interactions, the voice commands are transmitted via IoT protocols, enabling users to command the robotic car over extended distances. In contrast, short-range interactions are facilitated through Bluetooth communication within a specific range of approximately 20 meters, ensuring precise control. The system intelligently processes user inputs, translating them into actionable commands that guide the robotic car's movements. The multimodal nature of the system allows users to switch between control modes fluidly, adapting to different scenarios. Throughout the workflow, the project maintains a delicate balance between range, responsiveness, and adaptability, culminating in a dynamic and user-centric system that redefines the landscape of robotic control.

The system can be divided into 3 separate categories:

1. Software
2. Communication
3. Hardware

3.1.1 Software

The software contains the app side of the system. Here, data is handled into 2 separate command systems. They are:

A. Bengali voice control
B. Remote controller

A. Bengali voice control

The voice control model comes from an integrated voice recognition system from Google API. This API can recognize voice and decode into text on its own. Here a Bengali Language model is used to decode voice commands from Bengali to a string type datatype. This string type data is based on the following model:

1. Forward - সামনে যাও (Samne jao)
2. Backward - পিছনে যাও (Pichone jao)
3. Left - বামে যাও (Bame jao)
4. Right - ডানে যাও (Dane jao)
5. Stop - থামো (Thamo)

After recognition, the strings are first passed through the Levenshtein algorithm for checking the closest match and then later converted into integer values in order to pass through into the communication part. The integer values are as follows:

1. Forward - 3
2. Backward - 4
3. Left - 1
4. Right - 2
5. Stop – 5

Figure 3.1 shows the complete voice input system from the app to the robocar.

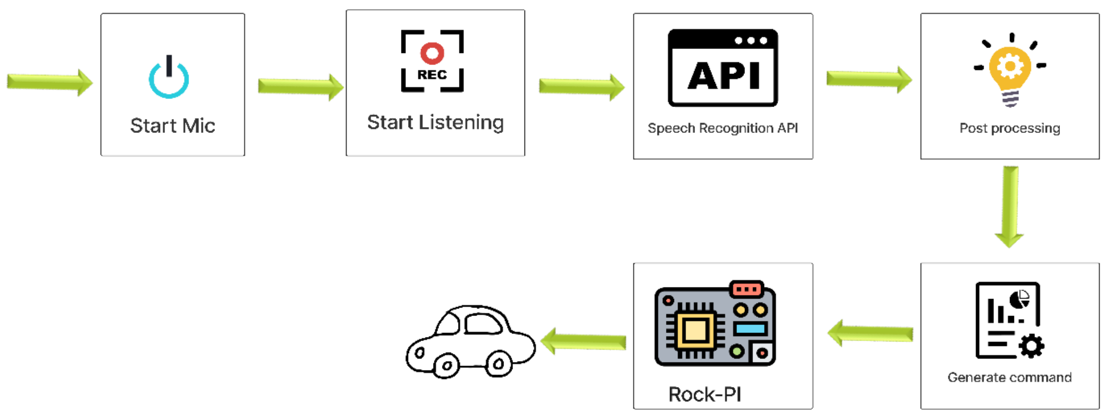

Figure-3.1: Voice control process

B. Remote controller

The remote-controller is used as a normal remote-control protocol with simple commands with forward, backward, left, right and stop button in the middle. The remote-controller input is also similarly converted into integer type data and passed into the communication part of the system. It uses exactly the same values as described in the Bengali voice control system integer conversions. That being:

1. Forward - 3
2. Backward - 4
3. Left - 1
4. Right - 2
5. Stop - 5

This is how the Remote controller part comes to an end.

Other Features in Software

- ✓ **User Authentication:** In this section the user needs to login with the robocar ID that the app will control and the password attached with it. This section is done so that the user has access to the unique vehicle that he wants to control and based the robocar ID the data will be passed to only that specific robocar in order to control that car. The password is used to give security to the cars from unwanted and malicious access from random users or hackers.
- ✓ **Connection establishment:** In order to establish connection to the robocar, a button is used. Upon pressing the button, the user will try to connect to the robocar and also establish connection to the database and command storage facilities. This is done to give flexibility to the user for when he wants to pass data and control the vehicle. Also, it saves data leakage or unwanted commands to pass into the database because of other apps in the mobile phone.
- ✓ **Log Cats:** This is the part to show the user that if an error has occurred while connecting to the robocar. It also receives the communication from the robocar as an acknowledgement that the robocar has received the data passed to it.

- ✓ **Permissions:** The app requires some permissions to pass data into the communication system. These include Bluetooth as it uses it as an option of communicating to the robocar.

3.1.2 Communication

The communication part can be divided into 2 parts. These are as follows:

A. Long distance communication using IoT
B. Short distance communication using Bluetooth

A. IoT (Internet of Things)

In the context of the "Robocar VIoT: A Voice and IoT Controlled Robotic Car" project, the Internet of Things (IoT) serves as a pivotal communication method. IoT enables the robotic car to connect to the internet, facilitating seamless, almost real-time communication between the car and the controlling device. Through IoT, users can command and control the robotic car remotely over extended distances. This connectivity enhances the responsiveness and dynamic capabilities of the system, allowing for a versatile and interactive user experience. IoT integration ensures that the robotic car can be accessed and controlled from virtually anywhere, transforming the project into a cutting-edge and globally accessible technology. Figure 3.2 shows the complete IoT system used in the project.

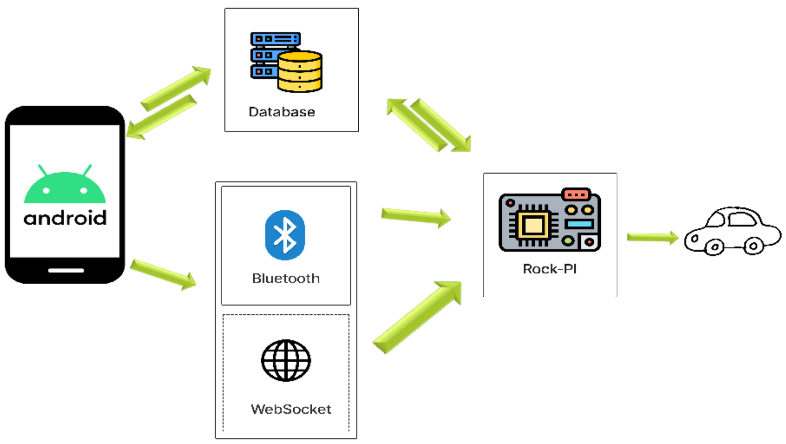

Figure-3.2: IoT communication system

B. Bluetooth

Bluetooth technology plays a crucial role in providing short-range communication capabilities within the "Robocar VIoT" project. It allows users to establish a direct and responsive connection with the robotic car through the Android app. This short-range communication is particularly useful for precise and nuanced control options, offering users the ability to manage the robotic car with enhanced accuracy. Bluetooth communication complements the long-range capabilities of IoT by providing users with a close-range, hands-on control experience. The combination of IoT for extended distances and Bluetooth for short-range interactions ensures a comprehensive and adaptable communication framework, catering to various user preferences and scenarios in the robotic control system. Figure 3.3 shows the Bluetooth communication system in the project.

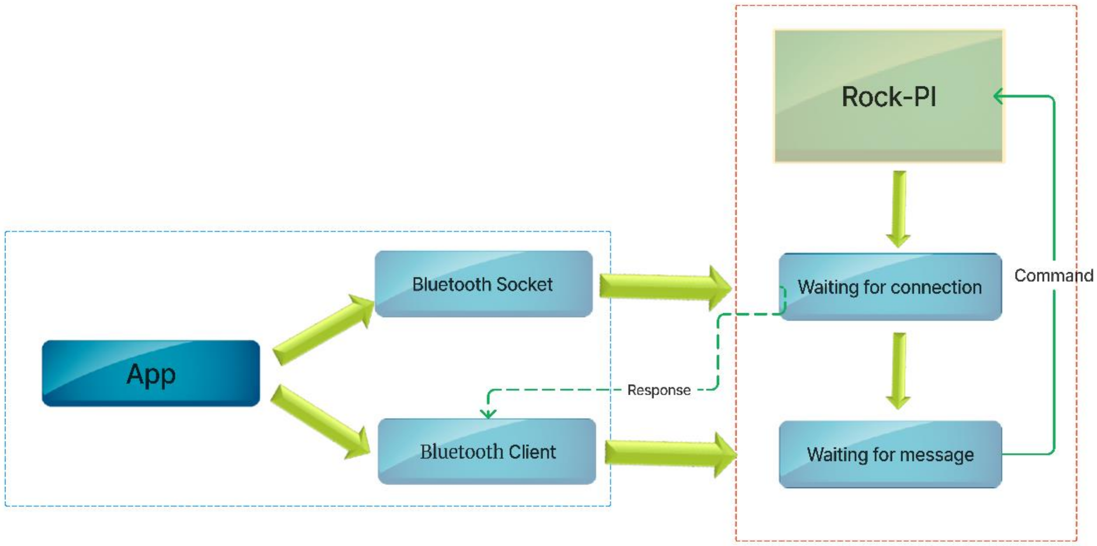

Figure-3.3: Bluetooth communication system

3.1.3 Hardware

The hardware part can be divided into 3 parts:

A. Rock Pi 3a Micro-controller
B. L293D Motor Driver
C. Python scripts through Debian OS from the micro-controller

A. Rock Pi 3a Micro-controller

The Rock Pi 3a Micro-controller serves as the brain of the "Robocar VIoT" project, providing the computational power and interface capabilities essential for controlling the robotic car. This micro-controller, based on the Rock Chip RK3328 processor, offers a combination of high performance and energy efficiency. It features multiple connectivity options, including Wi-Fi and Bluetooth, enabling seamless integration into the Internet of Things (IoT) framework. The Rock Pi 3a is equipped with GPIO (General Purpose Input/Output) pins, allowing it to interface with various hardware components such as sensors and the L293D Motor Driver. Its Debian OS support provides a stable and versatile platform for running Python scripts that control the robotic car's movements, ensuring a robust and adaptable hardware foundation for the project. Figure 3.4 shows a Rock Pi 3a micro-controller.

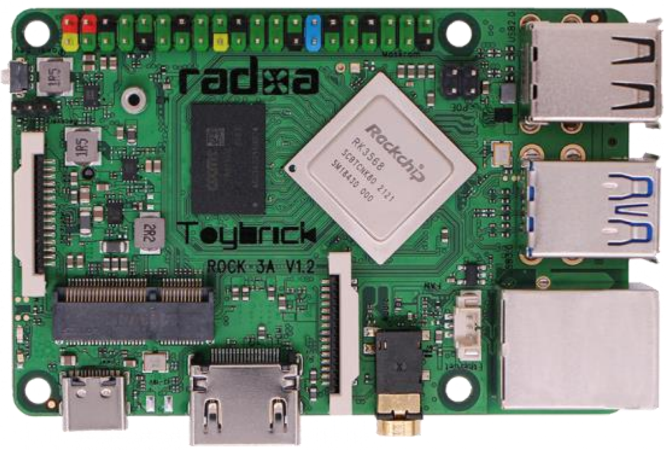

Figure-3.4: Rock Pi 3a Micro-controller

B. L293D Motor Driver

The L293D Motor Driver acts as the interface between the Rock Pi 3a Micro-controller and the motors of the robotic car. This motor driver is specifically designed for controlling the direction and speed of DC motors. With its dual H-bridge configuration, the L293D can drive two motors independently, making it well-suited for a two-wheeled robotic car. The motor driver receives signals from the Rock Pi 3a's GPIO pins, translating these signals into precise control of the motors. By enabling forward, backward, left, and right movements, the L293D Motor Driver ensures the accurate execution of commands issued by the micro-controller. Its compact design and efficient motor control capabilities make it an integral component in the hardware architecture of the project. Figure 3.5 shows an L293D motor driver.

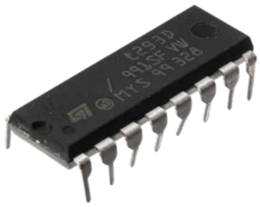

Figure-3.5: L293D Motor Driver

C. Python scripts through Debian OS from the micro-controller

Python scripts running on the Debian OS of the Rock Pi 3a Micro-controller form the software layer responsible for controlling the movements and behavior of the robotic car. These scripts utilize the GPIO pins to send signals to the L293D Motor Driver, dictating the direction and speed of the motors. The use of Python allows for a flexible and user-friendly programming environment, facilitating the implementation of various control algorithms and behaviors. Through the integration of Debian OS, the Rock Pi 3a provides a stable and well-supported platform for running these scripts, ensuring the efficient execution of commands and responsive control of the robotic car. The Python scripts enable the seamless integration of voice commands, joystick controls, and IoT functionalities, offering a comprehensive and adaptable control interface for users. Figure 3.6 shows the complete data passing system from Rock Pi to the Android App.

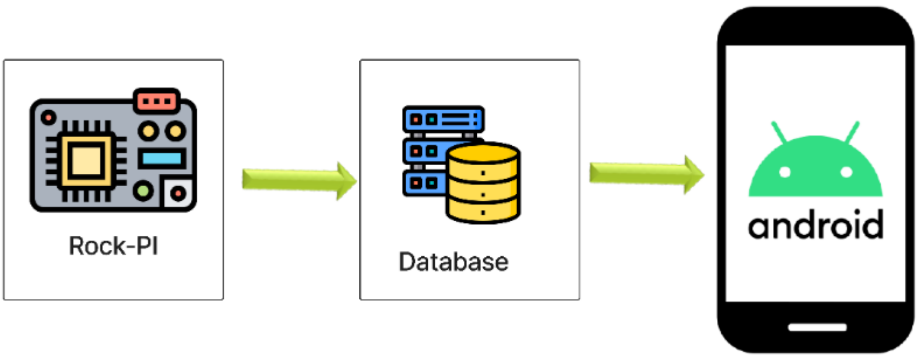

Figure-3.6: Data passing from micro-controller

It is divided into 4 separate threads:

1. Read
2. Update
3. Bluetooth
4. Socket

Each thread controls each component related operations.

Other components such as Battery as power, connecting wires and DC motors round off the hardware section of the project.

3.2 System architecture

The complete system architecture includes all the components working together as a whole to create the perfect environment as documented in Figure 3.7.

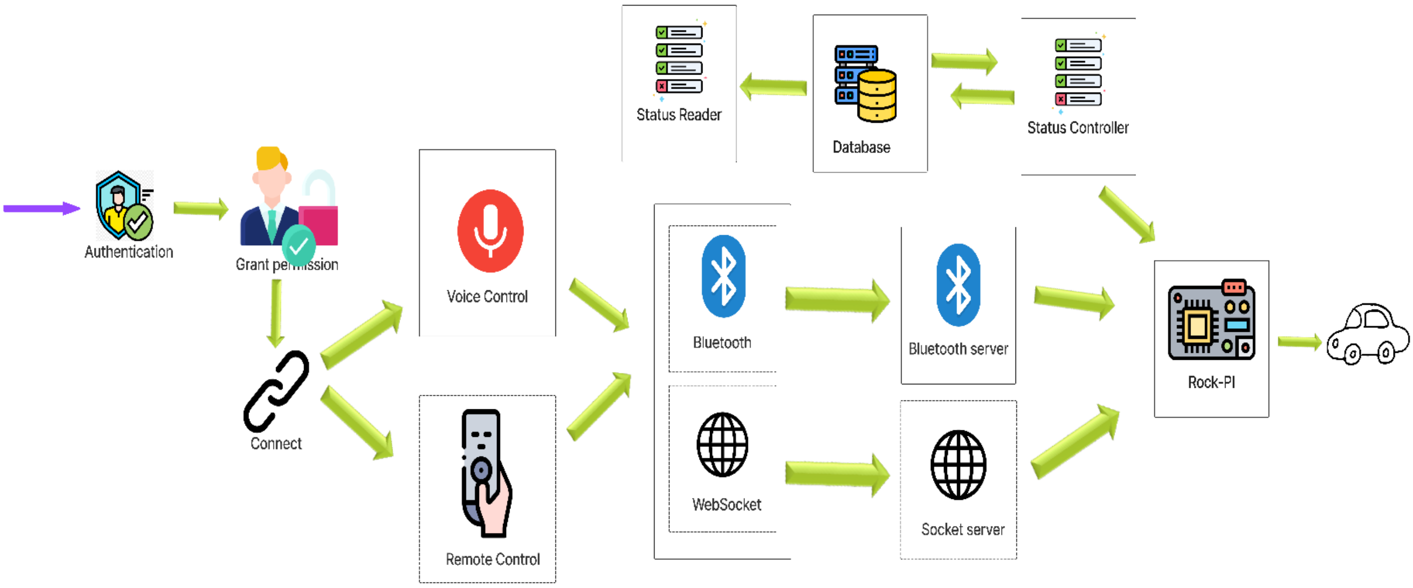

Figure-3.7: Complete system architecture

3.2.1 Software components

- ✓ **Android App**

  Function: Acts as the user interface for controlling the robotic car, offering both voice command and joystick-style control options.

  Features: Provides a seamless and adaptable control experience, ensuring a personalized interaction with the robotic car.

- ✓ **Voice recognition system**

  Function: Integrates voice command capabilities into the system, allowing users to control the robotic car through Bengali voice commands.

  Features: Utilizes voice processing algorithms for real-time recognition, enhancing the user experience

- ✓ **Remote-Controller interface**

  Function: Enables users to control the robotic car through an Android app using remote-controller style controls, providing a hands-on and intuitive control option.

  Features: Incorporates input from the Android app to generate corresponding signals for the L293D Motor Driver, enhancing the overall versatility of the control system.

3.2.2 Communication protocols

- ✓ **Internet of Things (IoT)**

  Function: Enables long-range control of the robotic car, allowing users to command and monitor the vehicle remotely over the internet.

  Features: Utilizes Wi-Fi connectivity on the Rock Pi 3a, ensuring seamless integration into the IoT framework.

- ✓ **Bluetooth**

  Function: Provides short-range communication for precise and nuanced control of the robotic car through the Android app.

  Features: Enables direct and responsive control, enhancing the hands-on experience for users.

3.2.3 Hardware components

- ✓ **Rock Pi 3a Micro-controller**

  Function: Serves as the central processing unit, running Debian OS and executing Python scripts for controlling the robotic car.

  Features: Rock Chip RK3328 processor, GPIO pins, Wi-Fi, and Bluetooth connectivity for IoT integration.

- ✓ **L293D Motor Driver**

  Function: Interfaces with the Rock Pi 3a, controlling the direction and speed of the motors for precise robotic car movements.

Features: Dual H-bridge configuration, allowing independent control of two motors, ensuring accurate execution of commands.

- ✓ **Python scripts**

  Function: Responsible for interpreting user commands, interfacing with the GPIO pins, and generating signals for the L293D Motor Driver.

  Features: Python offers a versatile and user-friendly programming environment, allowing for the implementation of various control algorithms and behaviors.

- ✓ **Debian OS**

  Function: Serves as the operating system for the Rock Pi 3a, providing a stable platform for running Python scripts and managing system resources.

  Features: Debian OS offers robust support, ensuring efficient execution of commands and responsive control of the robotic car.

3.3 Tools used

There are a few important tools that are used to complete the project. These tools serve an important part in both hardware and software parts of the project.

3.3.1 Android Studio

Purpose: Android Studio serves as the primary integrated development environment (IDE) for building the Android application.

Features: Offers a comprehensive set of tools for designing, coding, testing, and debugging Android apps.

3.3.2 Java

Purpose: Java is the primary programming language for developing the Android application.

Features: Provides platform independence, robustness, and ease of development for Android applications.

3.3.3 Firebase

Purpose: Firebase is utilized for real-time data synchronization and cloud-based storage.

Features: Offers a NoSQL database, authentication services, and cloud functions for seamless integration with the Android app.

3.3.4 WebSocket

Purpose: WebSocket is employed for establishing a two-way communication channel between the Android app and the robotic car.

Features: Enables real-time data exchange, facilitating responsive and dynamic control.

3.3.5 Google Voice Recognition API

Purpose: Integrates voice recognition capabilities into the system, allowing users to control the robotic car through voice commands.

Features: Utilizes advanced voice processing algorithms for accurate and real-time recognition.

3.3.6 Python

Purpose: Python is the primary scripting language used for controlling the Rock Pi 3a micro-controller.

Features: Known for its readability and versatility, Python enables the implementation of various control algorithms and behaviors.

3.3.7 Shell scripts

Purpose: Shell scripts are employed for system-level tasks and automation on the Debian OS running on the Rock Pi 3a.

Features: Facilitates efficient execution of commands and system management.

3.3.8 Periphery GPIO

Purpose: Periphery GPIO library is used for interacting with the GPIO pins on the Rock Pi 3a, facilitating hardware control.

Features: Provides a convenient interface for managing GPIO operations and interacting with connected hardware components.

3.3.9 Multiprocessing library (Python)

Purpose: The Multiprocessing library in Python is utilized for parallelizing and optimizing certain tasks, enhancing the efficiency of the control system.

Features: Enables the execution of multiple processes concurrently, improving overall performance.

3.3.10 Async IO (Python)

Purpose: Async IO is employed for handling asynchronous input/output operations in Python scripts.

Features: Enhances the responsiveness of the system by allowing non-blocking communication with different components.

3.3.11 Levenshtein Library

Purpose: Facilitates enhanced user experience and adaptability to diverse voices through the Levenshtein distance metric.

Features: Accurate string distance calculations for precise voice command recognition and error correction.

These tools collectively contribute to the development, communication, and control aspects of the Robocar VIoT project, ensuring a comprehensive and efficient system architecture.

3.4 Pseudocode

The basic pseudocode of the project can be adjusted into 3 sections. They are as follows:

1. Python script for Bluetooth

2. Python script for IoT
3. Levenshtein calculation for voice recognizer strings

3.4.1 Python script for Bluetooth

Input: No input parameter is needed

Output: No output is generated

```
function readFromBT():
    // Initialize Bluetooth server socket
    server_sock = createBluetoothSocket(RFCOMM)
    // Bind and listen for incoming connections
    port = 1
    server_sock.bind(("", port))
    server_sock.listen(1)
    // Accept incoming connection
    client_sock, address = server_sock.accept()
    try:
        // Receive and process data
        while True:
            data = client_sock.recv(1024)
            if not data:
                break
            command = int(data.decode())
            execute(command)
    except Exception as e:
        // Handle exceptions
        sleep(4)
    finally:
        // Close sockets
        client_sock.close()
        server_sock.close()
    // Recursive restart
    readFromBT()
```

3.4.2 Python script for IoT

Input: No input parameter is needed

Output: No output is generated

```
function readFromSocket():
    async function handler(websocket, path):
        // Asynchronously receive data from the WebSocket
        data = await websocket.recv()
        // Convert the received data to an integer and execute the command
        command = parseInt(data)
        execute(command)
    try:
        // Set up a WebSocket server
        start_server = websockets.serve(handler, "0.0.0.0", 8000)
        // Run the WebSocket server
        asyncio.get_event_loop().run_until_complete(start_server)
        asyncio.get_event_loop().run_forever()
    except Exception as e:
        // Handle exceptions by recursively restarting the function
        readFromSocket()
```

3.4.3 Levenshtein calculation for voice recognizer strings

Input: str1 and str2 to be compared, m and n indices of str1 and str2 correspondingly, till which comparing needs to be done.

Output: An integer representing the Levenshtein distance between str1 and str2.

//Finds the minimum number of single-character edits (insertions, deletions, or substitutions) required to transform one string into the other.

```
function levenshteinRecursive(str1, str2, m, n):
    // Base cases
    if m == 0:
        return n
    if n == 0:
        return m
```

```
// If the last characters are the same
if str1[m - 1] == str2[n - 1]:
    return levenshteinRecursive(str1, str2, m - 1, n - 1)
// If the last characters are different
return 1 + min(
    // Insert operation
    levenshteinRecursive(str1, str2, m, n - 1),
    // Remove operation
    min(
        levenshteinRecursive(str1, str2, m - 1, n),
        // Replace operation
        levenshteinRecursive(str1, str2, m - 1, n - 1)
    )
)
```

4 Project Implementation

The project can be implemented practically for various cases and these cases can create a very important impact in our daily lives.

4.1 System implementation

In real life, the project can serve a wide range of implementations. Such as:

4.1.1 Voice-controlled and IoT enabled wheelchair: Adapting the technology to create a wheelchair that responds to Bengali voice commands, providing individuals with mobility challenges greater independence and control.

4.1.2 Smart home assistance for people with disabilities: Enabling individuals with disabilities to control home devices and appliances through voice commands, enhancing accessibility within their living spaces.

4.1.3 Autonomous vehicles: Integrating similar control mechanisms into autonomous vehicles for both short-range and long-range interactions, allowing users to control and monitor their vehicles remotely.

4.1.4 Telepresence robot for remote communication: Developing a telepresence robot that allows users to virtually visit and interact with distant locations, facilitating remote communication for work or personal purposes.

4.1.5 Assistive robotics in healthcare: Implementing robotic systems to assist individuals with limited mobility in healthcare settings, such as fetching items, opening doors, or providing companionship.

4.1.6 Smart retail: Implementing robotic systems for inventory management and customer assistance in retail environments, providing a unique and interactive shopping experience.

4.1.7 Autonomous drones: Integrating similar control mechanisms into drones for various applications, such as aerial photography, surveillance, and search-and-rescue operations.

4.1.8 Research and exploration: Using robotic systems for research purposes, such as exploring hazardous environments, conducting experiments, or collecting data in remote locations.

4.1.9 Security systems: Designing surveillance robots equipped with cameras and sensors for security purposes, allowing users to patrol and monitor their properties remotely.

4.1.10 Smart agriculture: Utilizing robotic vehicles for precision agriculture, enabling farmers to remotely monitor and control equipment for tasks like planting, harvesting, and spraying.

4.2 Morality or ethical issues

Implementing a project like "Robocar VIoT: A Voice and IoT Controlled Robotic Car" raises several ethical considerations and moral implications. Here are some potential morality and ethical issues associated with the project:

4.2.1 Privacy concerns: The project involves the use of IoT and potentially collects data, raising concerns about user privacy. Ethical considerations must be given to how user data is collected, stored, and used.

4.2.2 Security risks: As the project relies on IoT for long-range control, ensuring the security of the communication channels is crucial. Unauthorized access or hacking could lead to ethical issues, especially if it compromises user safety or privacy.

4.2.3 Bias in voice recognition: If the voice recognition system exhibits bias, it may inadvertently discriminate against certain users or struggle to understand specific accents or dialects. Addressing and mitigating bias is an ethical imperative.

4.2.4 User safety: Ensuring the safety of users and those around them when controlling the robotic car is paramount. Any potential risks, malfunctions, or unintended consequences need to be carefully considered and mitigated.

4.2.5 Autonomy and accountability: The level of autonomy of the robotic car raises ethical questions about responsibility and accountability in case of accidents or unintended actions. Clear guidelines and regulations should be in place.

4.2.6 Equitable access: Ensuring that the technology is accessible to a broad range of users and not restricted to specific demographics is essential for ethical deployment.

4.2.7 Unintended consequences: The project may have unintended consequences or be used in ways that were not initially anticipated. Regular ethical assessments and impact analyses should be conducted to address any emerging issues.

Addressing these morality and ethical issues requires a proactive approach, involving ethical design, clear communication with users, adherence to privacy standards, and continuous monitoring of the system's impact on individuals and society.

4.3 Socio-economic impact and sustainability

The "Robocar VIoT: A Voice and IoT Controlled Robotic Car" project has the potential to generate significant socio-economic impact and contribute to sustainability in various ways:

4.3.1 Enhanced accessibility and inclusion: The project, particularly in the context of a voice-controlled wheelchair, can enhance accessibility for individuals with mobility challenges. This contributes to greater inclusion in various socio-economic activities, such as education, employment, and community engagement.

4.3.2 Increased independence for persons with disabilities: Providing a voice and IoT-controlled solution for mobility assistance empowers individuals with disabilities, allowing them to navigate their environments more independently. This increased independence has positive socio-economic implications, fostering self-sufficiency and reducing dependence on external support.

4.3.3 Job creation in the technology sector: The development and maintenance of the project, including software, hardware, and support services, can contribute to job creation in the technology sector. This includes roles in software development, system maintenance, customer support, and more.

4.3.4 Advancements in assistive technologies: The project contributes to advancements in assistive technologies, which can have a broader impact on the development of innovative solutions for individuals with diverse abilities. This, in turn, supports a growing sector focused on improving the quality of life for people with disabilities.

4.3.5 Efficient healthcare and rehabilitation: The project's applications in healthcare, such as robotic assistance in hospitals, can contribute to more efficient patient care and rehabilitation processes. This can potentially reduce healthcare costs and enhance the overall quality of healthcare services.

4.3.6 Reduced dependence on human assistance: By offering a voice and IoT-controlled solution for tasks that might require human assistance, the project contributes to reducing the burden on caregivers and support systems. This can lead to more efficient allocation of resources and time.

4.3.7 Long-term cost savings: If the project is adopted widely, particularly in the healthcare sector, there is potential for long-term cost savings. Efficient robotic assistance can contribute to streamlined processes, reducing the overall cost of care.

To maximize the socio-economic impact and sustainability of the project, it's essential to consider factors such as affordability, scalability, adaptability to different contexts, and ongoing community engagement throughout the development and implementation phases. Additionally, collaborating with relevant stakeholders, including disability advocacy groups, healthcare professionals, and technology experts, can enhance the project's effectiveness and sustainability.

4.4 Financial analyses and budget

Creating a financial analysis and budget for a project involves estimating the costs associated with development, testing, implementation, and ongoing maintenance. Below is a simplified template for financial analyses and a budget for the "Robocar VIoT: A Voice and IoT Controlled Robotic Car" project:

4.4.1 Financial analysis

A. Development costs

- ✓ Hardware components:

    - Rock Pi 3a Micro-controller
    - L293D Motor Driver
    - Python Scripts
  - ✓ Software development:
    - Android App (Development and Testing)
    - Voice Recognition System:
    - Integration with Google Voice Recognition API
  - ✓ Additional tools: Android Studio, Java, Firebase, WebSocket, Shell Scripts, Periphery GPIO library, Multiprocessing Library, Async IO etc.

B. Testing and quality assurance

- ✓ Comprehensive testing of hardware and software components
- ✓ Debugging and addressing any issues found during testing

C. Implementation and deployment

- ✓ Integration of all components
- ✓ Deployment of the system for practical use

D. Maintenance and support

- ✓ Ongoing support for bug fixes and updates
- ✓ Potential future enhancements and feature additions

4.4.2 Development costs

- ✓ Hardware components: BDT 12,000
- ✓ Software development: Free
- ✓ Voice recognition system: Free
- ✓ Additional tools: Free

This budget template provides a starting point, and actual costs may vary based on project complexity, scale, and specific requirements.

5 Conclusion

In conclusion, the "Robocar VIoT: A Voice, IoT, and Bluetooth Controlled Robotic Car" project represents a groundbreaking initiative at the intersection of robotics, voice recognition, and connectivity technologies. With a focus on enhancing mobility and control, especially in scenarios like a voice-controlled wheelchair, the project holds significant promise for individuals with mobility challenges. The seamless integration of Bengali voice commands, remote-controller controls, and IoT functionalities opens up new possibilities for human-robot interaction.

5.1 Conclusion and challenges faced

While developing the "Robocar VIoT: A Voice and IoT Controlled Robotic Car" project, several challenges may be encountered. Here are some potential challenges and considerations:

5.1.1 **Technical complexity:** Integrating voice recognition, IoT, and Bluetooth technologies into a unified system can be technically complex. Ensuring seamless communication and synchronization between these components may pose challenges.

5.1.2 **Voice recognition accuracy:** Achieving high accuracy in voice recognition, especially in a multilingual context like Bengali, may be challenging. Overcoming dialect variations and ensuring effective communication is crucial.

5.1.3 **Security concerns:** Ensuring the security of the IoT and Bluetooth communication channels is essential. Protecting against unauthorized access and potential cyber threats requires robust security measures.

5.1.4 **Hardware compatibility:** Ensuring compatibility and optimal performance across various hardware components, including the Rock Pi 3a micro-controller and L293D Motor Driver, can be a challenge. Addressing potential hardware limitations is crucial.

5.1.5 User interface design: Designing an intuitive and user-friendly interface for both voice commands and joystick controls requires careful consideration. Ensuring accessibility for users with diverse abilities adds an extra layer of complexity.

Despite these challenges, each presents an opportunity for innovation and improvement. Addressing these issues with a proactive and collaborative approach will contribute to the successful development and implementation of the Robocar VIoT project.

5.2 Future work

The "Robocar VIoT: A Voice and IoT Controlled Robotic Car" project opens avenues for several future studies and research directions. These are outlined shortly as follows:

- ✓ Implement advanced voice recognition models for enhanced accuracy and multilingual support.
- ✓ Investigate autonomous navigation capabilities, including obstacle detection and collision avoidance.
- ✓ Explore edge computing for real-time processing, reducing latency and enhancing responsiveness.
- ✓ Implement additional accessibility features, such as gesture-based controls, for a more inclusive design.

These future works aim to further enhance the capabilities, usability, and societal impact of the Robocar VIoT project, contributing to ongoing advancements in the fields of robotics, human-computer interaction, and assistive technology.

References

[1] N. Kumar, B. Dey, C. Chetri, and A. Biswas, "Surveillance Robocar Using IoT and Blynk App," *in Advances in Communication, Devices and Networking*, 2020, pp. 423–431.

[2] Trisha and S. D. Kumar, "Design and Development of IoT-based Robot," *2020 International Conference for Emerging Technology (INCET)*, Belgaum, India, 2020, pp. 1-4, doi: 10.1109/INCET49848.2020.9154175.T. Nakagaki, H. Yamada, and Á. Tóth, “Maze-solving by an amoeboid organism,” *Nature*, vol. 407, no. 6803, pp. 470–470, Sep. 2000, doi: 10.1038/35035159.

[3] M. Gupta, R. Kumar, R. K. Chaudhary and J. Kumari, "IoT Based Voice Controlled Autonomous Robotic Vehicle Through Google Assistant," *2021 3rd International Conference on Advances in Computing, Communication Control and Networking* (ICAC3N), Greater Noida, India, 2021, pp. 713-717, doi: 10.1109/ICAC3N53548.2021.9725526.

[4] S. Chakraborty, N. De, D. Marak, M. Borah, S. Paul and V. Majhi, "Voice Controlled Robotic Car Using Mobile Application," *2021 6th International Conference on Signal Processing, Computing and Control (ISPCC)*, Solan, India, 2021, pp. 1-5, doi: 10.1109/ISPCC53510.2021.9609396.

[5] R. L. Khan, D. Priyanshu and F. S. Alsulaiman, "Implementation of Human Voice Controlled Robotic Car," *2021 10th International Conference on System Modeling & Advancement in Research Trends (SMART)*, MORADABAD, India, 2021, pp. 640-646, doi: 10.1109/SMART52563.2021.9676319.

[6] M. R. Huda, M. L. Ali and M. S. Sadi, "Real-Time Hand-Gesture Recognition for the Control of Wheelchair," *2022 12th International Conference on Electrical and Computer Engineering (ICECE)*, Dhaka, Bangladesh, 2022, pp. 384-387, doi: 10.1109/ICECE57408.2022.10088702.

[7] S. Islam, M. Ali, and M. Sadi, ‘Development of an IoT-Enabled Digital Noticeboard’, 10 2022, pp. 479–492.

[8] M. Alotaibi, T. Alhmiedat, Z. Bassfar, M. Sadi, R. Islam, and M. Islam, ‘Finger-Gesture Controlled Wheelchair with Enabling IoT’, Sensors, vol. 22, 11 2022.