Software Design Document

IoT-Based Bluetooth Controlled Smart Car

Submitted by:

Muhammad Farhan (F22BINFT1E02106)

Muhammad Junaid (F22BINFT1E02132)

Submitted to:

Dr. Musarat Karim

Department of Information Technology

Faculty of Computing

The Islamia University of Bahawalpur

Summary

This Software Design Document (SDD) provides a clear and detailed design of the **IoT-Based Bluetooth Controlled Smart Car**. The main purpose of this document is to convert the functional and non-functional requirements mentioned in the Software Requirements Specification (SRS) into an understandable and implementable system design.

The proposed system combines both hardware and software components to allow wireless control of a smart robotic car using a mobile application. The system uses Arduino UNO as the main controller, HC-05 Bluetooth module for wireless communication, and L298N motor driver to control the motors. This document explains the system architecture, design modules, data model, and user interface in a simple manner. The design focuses on simplicity, reliability, ease of use, and future enhancement possibilities

Contents

1 Application Architecture

The IoT-Based Bluetooth Controlled Smart Car follows a **layered embedded system architecture**. In this architecture, the complete system is divided into separate layers, where each layer performs a specific task. This design approach makes the system easier to understand, develop, and maintain.
Layered architecture also helps in debugging and allows new features to be added in the future without disturbing the entire system.

1.1 Architectural Overview

The system architecture is divided into the following layers:

- **User Interface Layer** – This layer includes the mobile application used by the user to send commands.
- **Communication Layer** – This layer handles Bluetooth communication using the HC-05 module.
- **Control Layer** – This layer consists of the Arduino UNO microcontroller that processes commands.
- **Actuator Layer** – This layer includes the motor driver and DC motors that perform movement.

Each layer interacts only with the next layer, which improves system stability and modularity.

1.2 Hardware Architecture

The hardware architecture explains the physical components used in the system and how they are connected with each other. Arduino UNO acts as the main control unit and is connected to the Bluetooth module and motor driver.

Hardware Components

- Arduino UNO
- HC-05 Bluetooth Module
- L298N Motor Driver
- DC Motors
- Battery Pack
- LED Indicators

The motor driver controls the speed and direction of the motors based on the signals received from Arduino. The battery provides power to all hardware components.

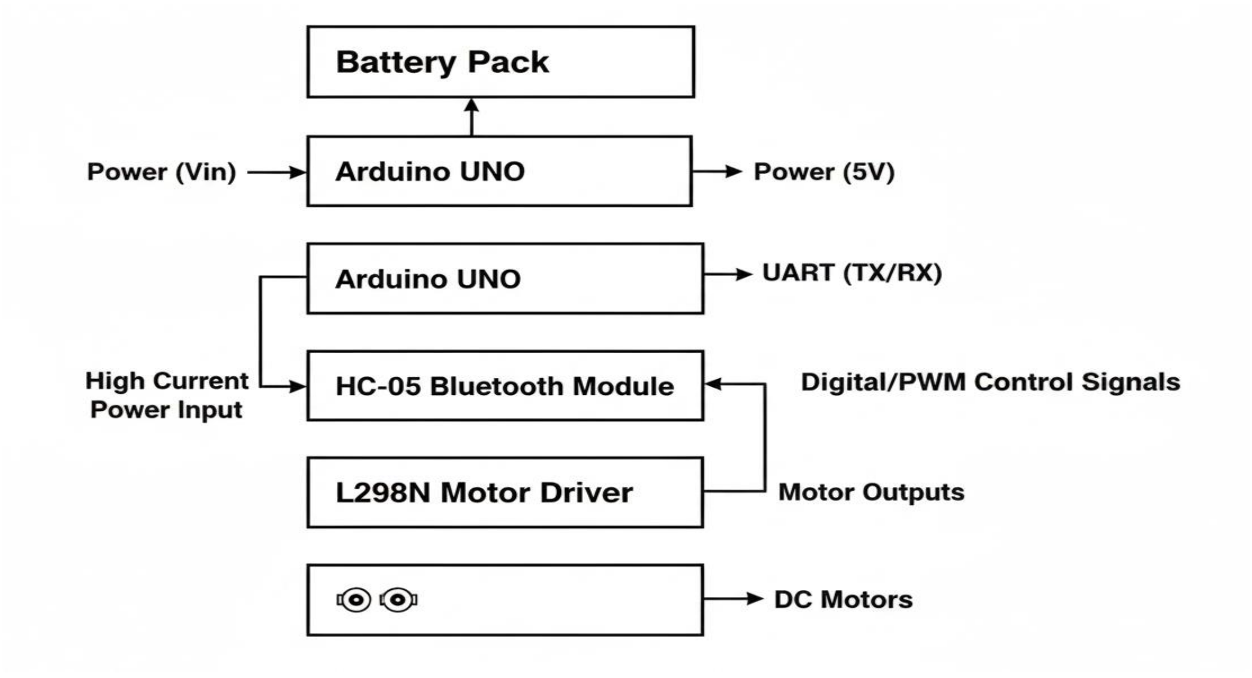

Figure 1.2: Hardware Block Diagram

1.3 Software Architecture

The software architecture describes how the system processes user commands internally. The mobile application sends simple character-based commands such as **F (Forward), B (Backward), L (Left), R (Right), and S (Stop)**.

The Bluetooth module receives these commands and forwards them to Arduino through serial communication. Arduino interprets the received command and sends appropriate signals to the motor driver to perform the required action.

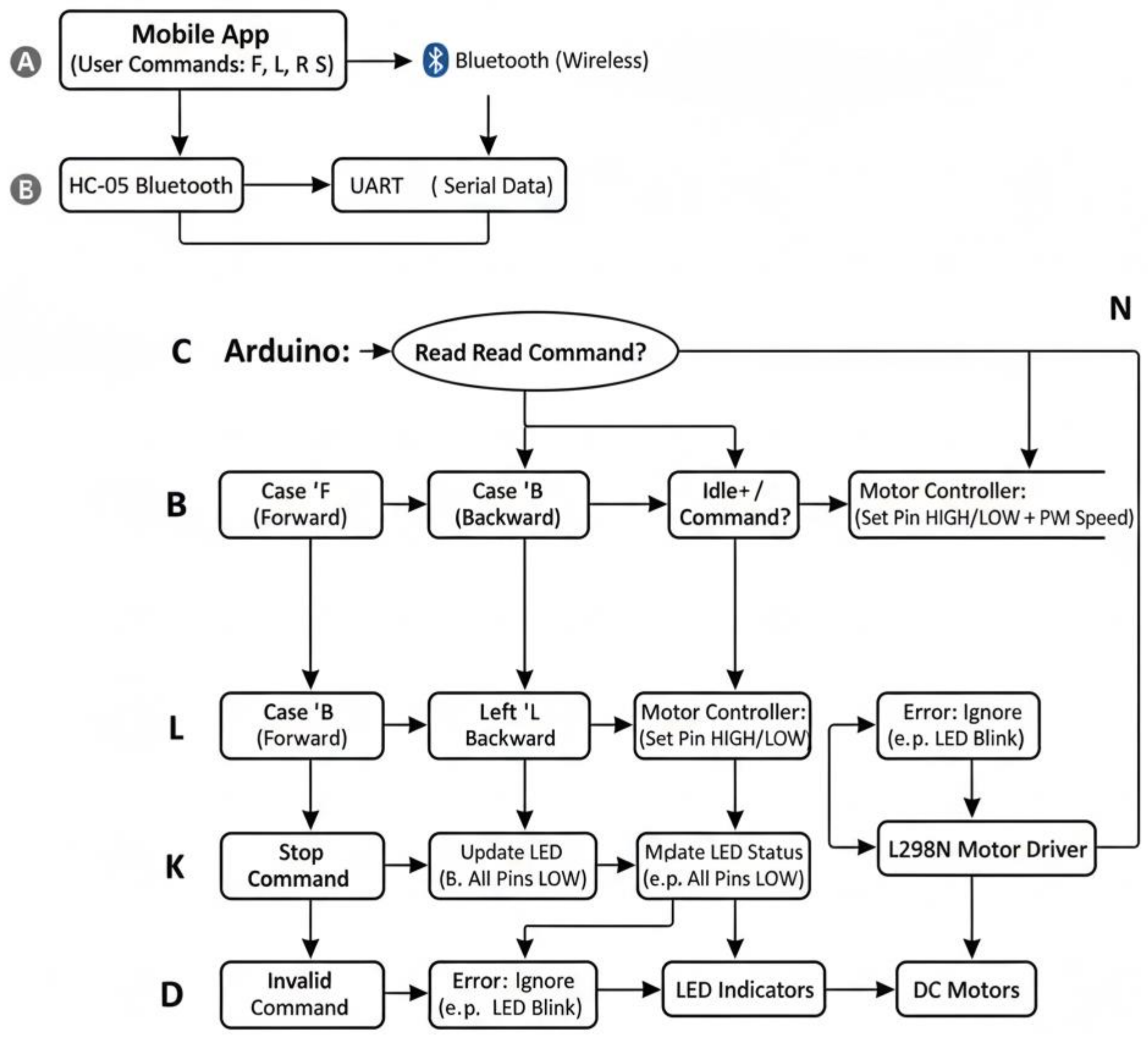

Figure 1.3: Software Flow Diagram

2 System Design Overview

The system uses a **modular controller-based design**, which is more suitable for embedded systems than complex architectures like MVC. Each module is responsible for a specific task, making the system efficient and easy to manage.

2.1 Main System Modules

- **Bluetooth Controller**
  Handles Bluetooth pairing
- **Command Processor**
  Interprets the received user commands and validates them before execution.
- **Motor Controller**
  Controls the direction and speed of the motors based on processed commands.
- **LED Controller**
  Displays system status and movement indication using LEDs and receives data from the mobile application.

3. UML Design Diagrams

UML diagrams are used to visually represent the working and structure of the system. These diagrams help in understanding system behavior and interactions.

3.1 Use Case Diagram

The Use Case Diagram shows how the user interacts with the smart car system. It highlights all possible actions that the user can perform through the mobile application.

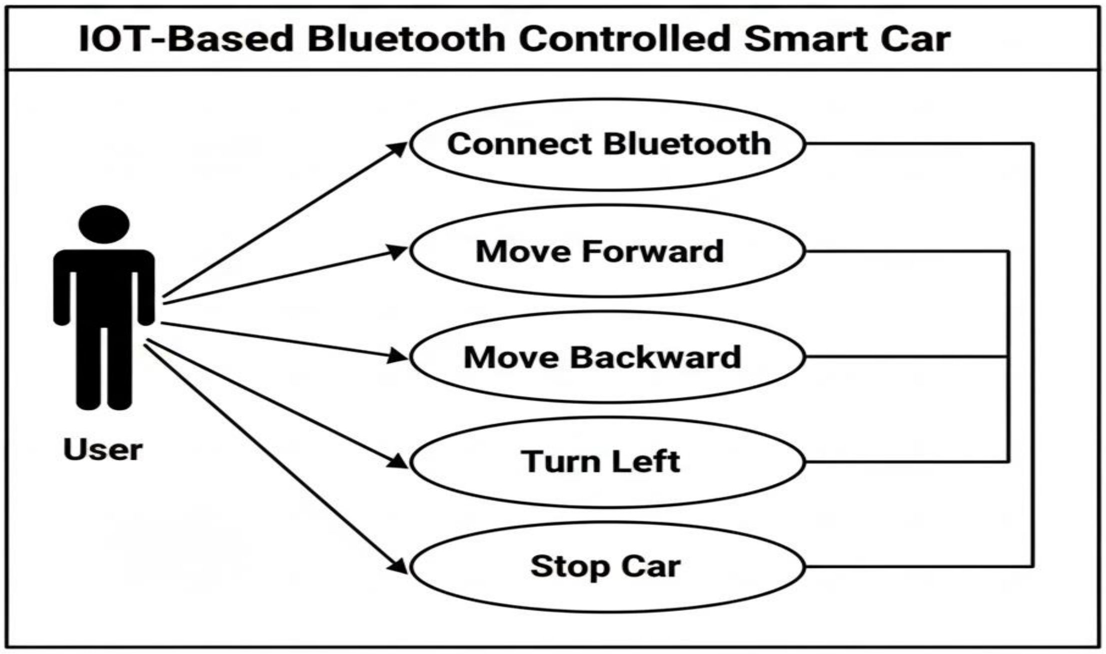

Figure 3.1: Use Case Diagram

3.2 Sequence Diagram

The Sequence Diagram explains the sequence of events when the user sends a command to the car. It shows how data flows from the mobile application to the motors step by step.

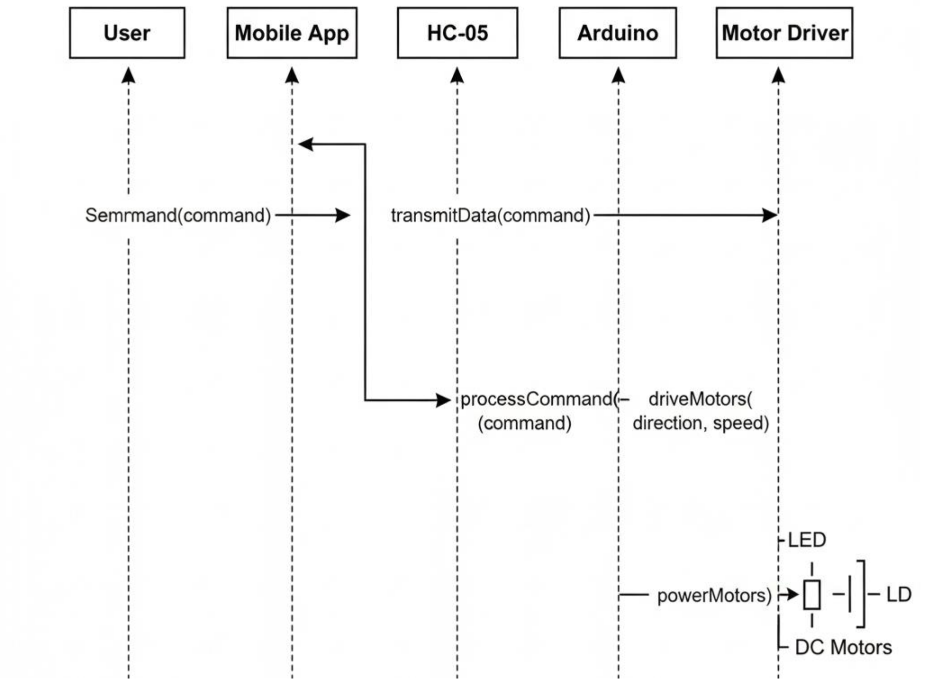

Figure 3.2: Sequence Diagram

3.3 Class Diagram

The Class Diagram represents the logical structure of the system. It shows different classes, their responsibilities, and basic functions.

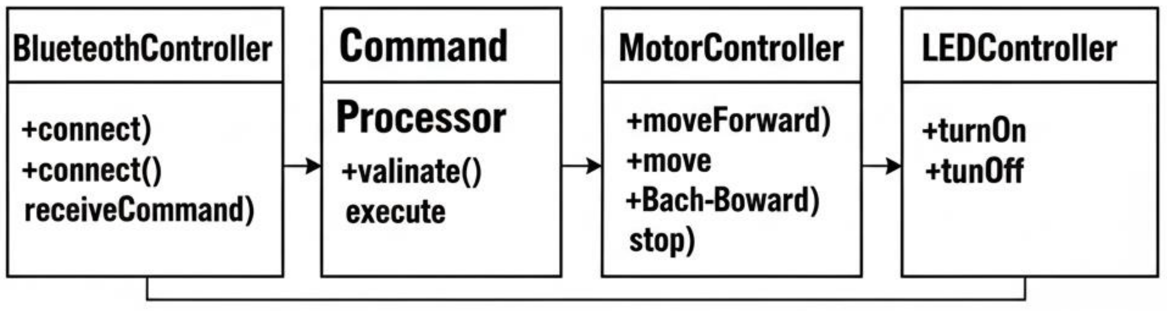

Figure 3.3: Class Diagram

4. Data Model Schema

Although the system does not use a database, logical data models are defined to explain how data is handled inside the system.

4.1 Data Entities

Command Entity

- Command ID
- Command Type
- Timestamp

MotorState Entity

- Motor ID
- Direction
- Speed

ConnectionStatus Entity

- Connection ID
- Connection State

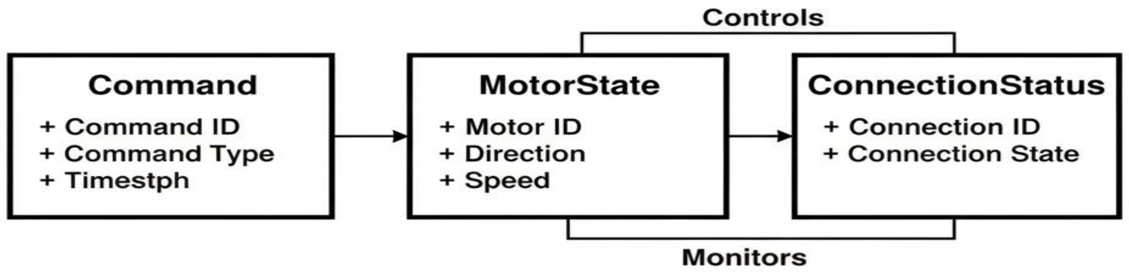

Figure 4.1: ER Diagram

5. User Interface Design

The user interface is designed to be simple and easy to use so that any user can control the smart car without technical knowledge.

5.1 Design Philosophy

The user interface is designed to be:

- Simple
- Responsive
- Clear buttons
- Easy Navigation

5.2 Screens Description

Bluetooth Connection Screen

- Device List
- Connect Button
- Status Indicator

Control Dashboard

- Forward Button
- Backward Button
- Left Button
- Right Button
- Stop Button

- LED Status Display

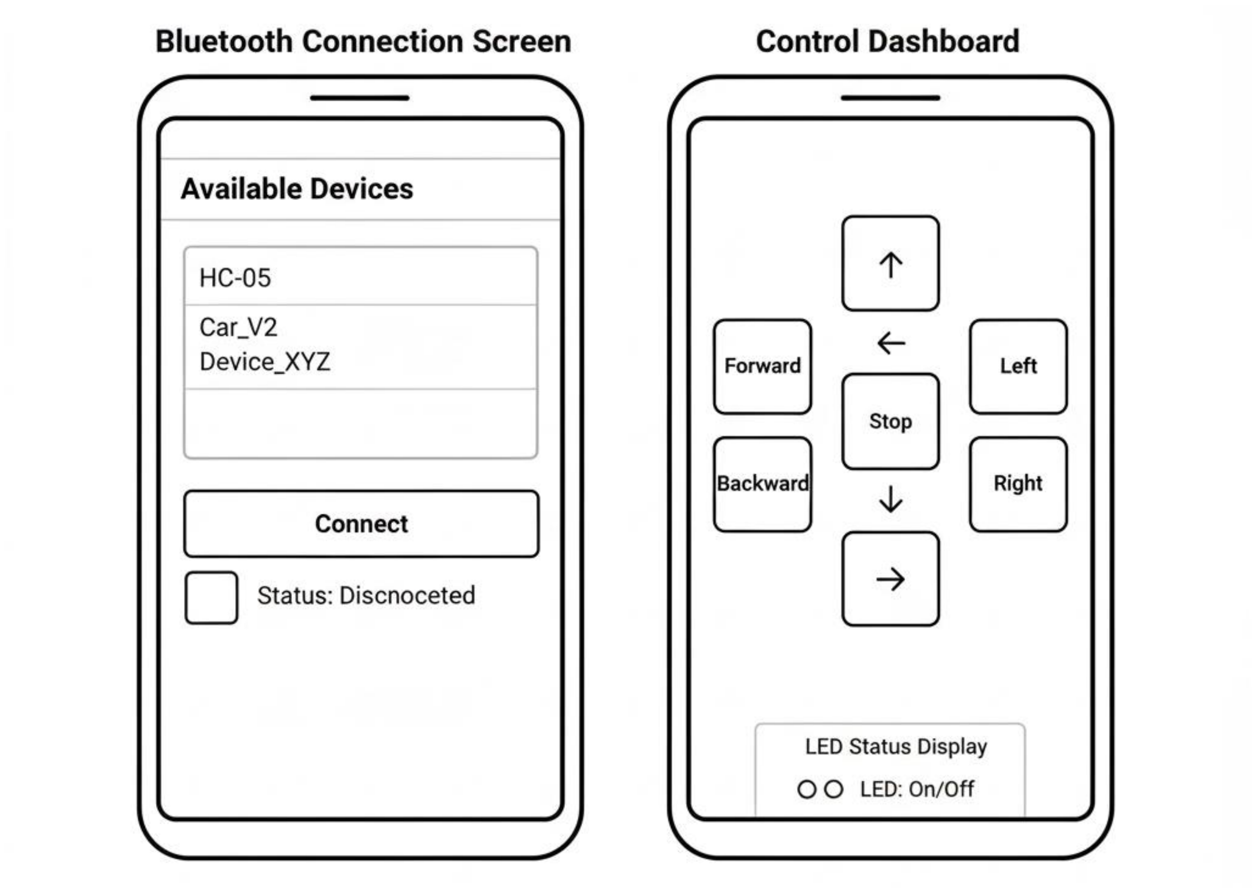

Figure 5.1: Mobile App UI Wireframe

6. Coding Conventions

The system code follows standard Arduino programming practices to ensure clarity and reliability:

- Meaningful variable and function names
- Modular and reusable functions
- Proper indentation
- Inline comments for understanding
- Basic error handling for invalid inputs