Project Proposal

Customizable Arduino Car Control App in Flutter

Objective

This project aims to develop a Flutter-based mobile application for controlling an Arduino-powered car via ESP32. The app will offer a customizable interface where users can add, remove, and adjust control elements like buttons or a joystick, each configured to send specific commands to the car. The app will connect to the ESP32 over WiFi, automatically detecting and displaying the IP and port of the connected device.

Introduction

Existing Arduino car apps often lack flexibility in user interface design. This project seeks to replicate and extend the Arduino Car App's functionality by providing a UI that can be personalized to the user's control preferences. Users can dynamically adjust control elements, making the app versatile and adaptable to different needs.

The app will:

- Connect to the ESP32 via WiFi, auto-detecting IP and port.

- Allow users to customize controls, defining shapes, sizes, and actions.

- Enable command-based communication, where each control sends a character understood by the ESP32 to perform actions on the car.

Flutter Libraries:

- Flutter socket library for TCP connection

- Flutter widgets for dynamic UI

Project Architecture

ESP32 as Client:

The ESP32 connects to the phone's hotspot network as a client and communicates with the app on the designated IP and port. It interprets command characters sent by the app to control car movements.

Flutter App as Controller:

The Flutter app connects to the ESP32 over WiFi. Users can send commands via customizable controls, and the app auto-detects the IP and port based on the phone's hotspot setup, enabling quick and seamless connection.

Development Phases

1. ESP32 Configuration

Set up the ESP32 to connect to the phone's hotspot, with IP and port for communication. The ESP32 will listen for incoming commands and execute corresponding car movements.

2. Flutter App Development

User Interface: Provide draggable, resizable control elements, including buttons and joystick options, which users can arrange and configure. Each element can be assigned a command character for the ESP32.

Connection Management: The app detects the phone's IP and port dynamically, displaying this information to the user.

3. Real-Time Command Handling

The app sends commands to the ESP32 based on user interactions, allowing real-time control of the car's movements.

Features and Functionalities

1. Customizable Controls

Users can add, remove, resize, and position control elements. Each button or joystick can be customized with a shape, size, label, and assigned command.

2. Automatic IP and Port Detection

The app auto-detects the IP address and port when connecting via WiFi, making setup easy for users.

3. Seamless Communication

Commands are sent to the ESP32 in real-time, enabling responsive control.

Testing and Validation

1. Connect the ESP32 to the phone's hotspot and verify IP detection.
2. Test each control to ensure it sends the correct command to the ESP32.
3. Confirm the ESP32 accurately interprets commands and controls the car as expected.
4. Validate UI customization, ensuring users can adjust and arrange controls easily.

Future Enhancements

1. Bluetooth Connectivity: Add support for Bluetooth as an alternative connection.
2. Real-Time Feedback: Display car status data, such as speed or battery level.
3. Saved Configurations: Allow users to save and load different control layouts.

Conclusion

This project will result in a highly customizable and user-friendly control app for Arduino-based cars, providing a personalized and engaging interface. It combines Flutter's UI flexibility with robust communication capabilities, delivering a versatile tool for remote car control.