# WebArmBot

# ESP8266-Powered WiFi Car with Interactive 3-Servo Arm

## Synopsis:

The ESP8266 Car with 3-Servo Arm project introduces an innovative robotic platform powered by the ESP8266 WiFi module. This project seamlessly integrates motor control for movement with a versatile three-servo arm, providing users with a unique web-based interface to interact with and control the car and its manipulative arm remotely. This synthesis of hardware and software engineering opens avenues for both educational exploration and practical applications.

## Key Components and Features:

**ESP8266 WiFi Module:**

The ESP8266 microcontroller acts as the central processing unit, offering WiFi connectivity for seamless communication with a web-based interface.

**Motor Control for Mobility:**

Incorporates motor drivers for precise control over the car's movement, allowing forward, backward, left, and right navigation.

**3-Servo Manipulator Arm:**

Features a flexible and articulate three-servo arm for performing precise and controlled tasks. Each servo can be individually controlled for a wide range of motion.

**Web-Based Control Interface:**

Develops a responsive web-based control panel accessible through a browser. This interface allows users to drive the car, control the manipulator arm, and receive live feedback from onboard sensors.

**Customizable Arm Movements:**

Empowers users to define and execute intricate movements by controlling each servo independently. This feature is especially useful for tasks requiring precision and flexibility.

**Educational Exploration:**

Designed with educational purposes in mind, the project serves as a practical learning tool for understanding robotics, IoT, and web-based control systems.

**Remote Task Execution:**

Offers the capability to execute tasks remotely, making it suitable for applications such as pick-and-place operations, surveillance, or even educational demonstrations.

## Circuit Diagram:

## circuit Diagram

## Libraries Used:

- **ESP8266WiFi:** Handles WiFi connectivity for the NodeMCU.

- **WiFiClient:** Enables communication with the WiFi network.

- **ESP8266WebServer:** Facilitates the creation of a web server to handle HTTP requests.

- **Servo:** Provides functionality for controlling servo motors.

## Pin Configurations:

- Defines the pin configurations for motor control (ENA, ENB, IN\_1, IN\_2, IN\_3, IN\_4) and servo connections (SERVO\_1\_PIN, SERVO\_2\_PIN, SERVO\_3\_PIN).

## Global Variables:

- **`speedCar`:** Initial motor speed for the car.

- **`speed\_Coeff`:** Speed coefficient for turning.

- `**servo1Position`, `servo2Position`, `servo3Position`:** Initial positions for the three servos.

## WiFi Configuration:

- Sets up the NodeMCU as an Access Point (AP) with a specified SSID (`"Prototype 1"`).

## Web Server Initialization:

- Creates an instance of the `ESP8266WebServer` class on port 80.

- Defines handlers for root ("/"), control ("/control"), and login ("/login") endpoints.

## Motor Control Functions:

- **stopMotors()**: Stops the car by setting motor control pins to LOW.

- **goAhead(), goBack(), goLeft(), goRight():** Control the car's movement in different directions by configuring motor control pins accordingly.

## Servo Control Function:

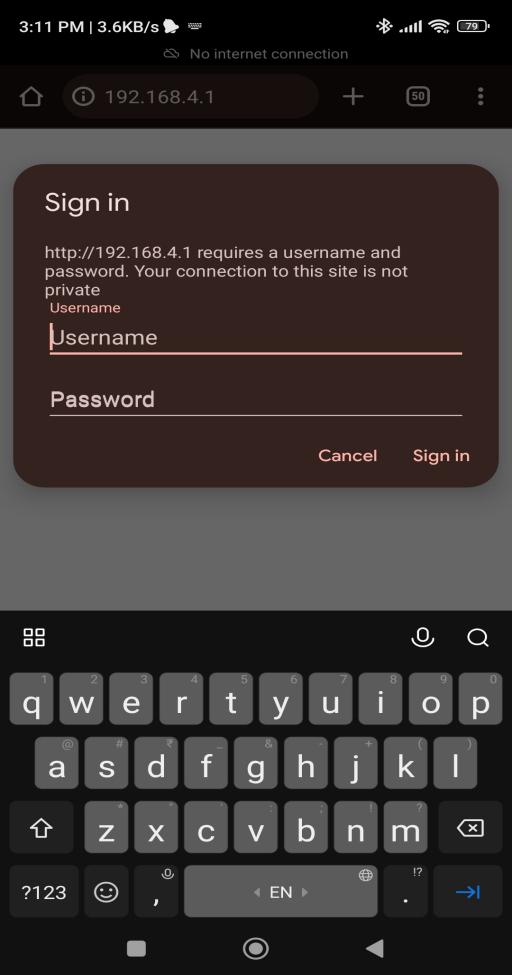
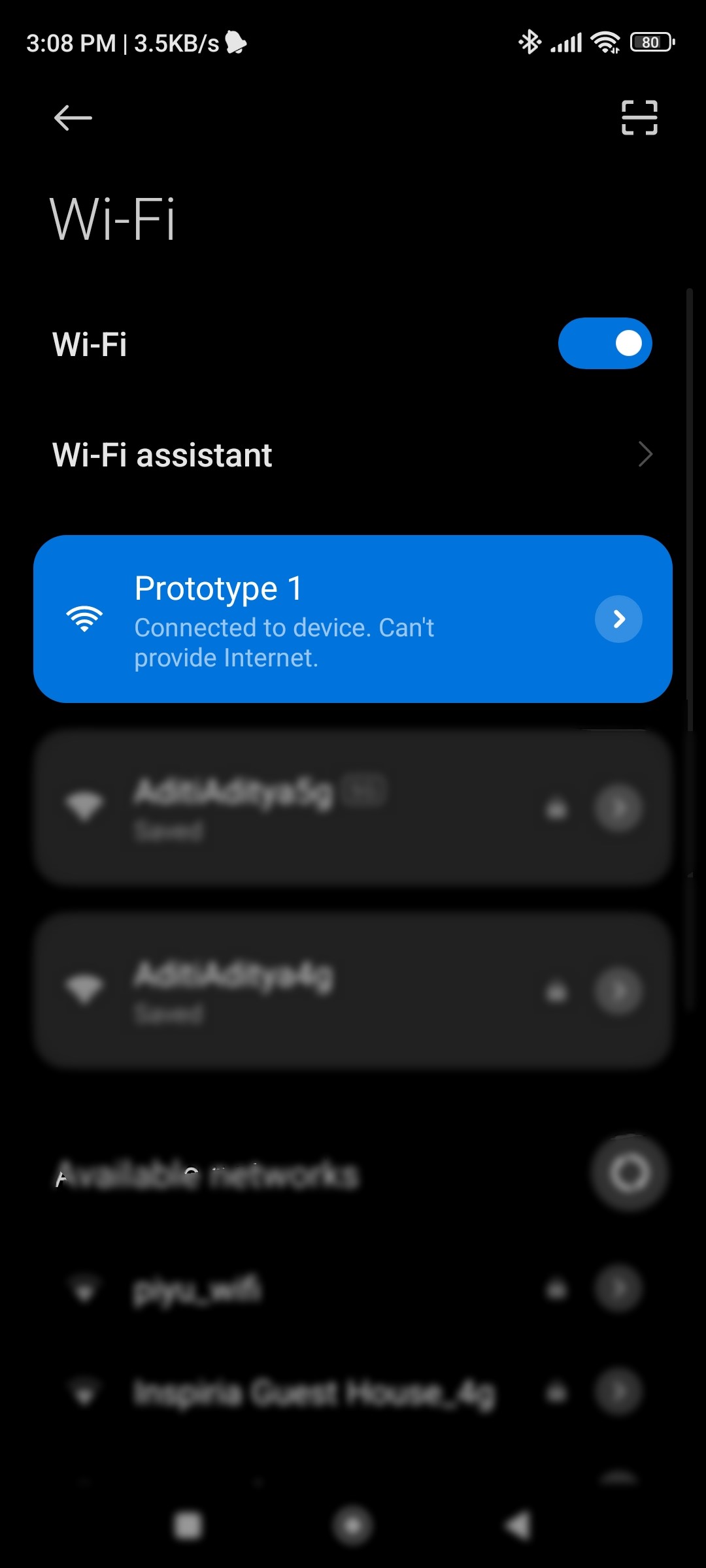
- **moveServo(int servoNumber, int position):** Moves the specified servo (1, 2, or 3) to the specified position.

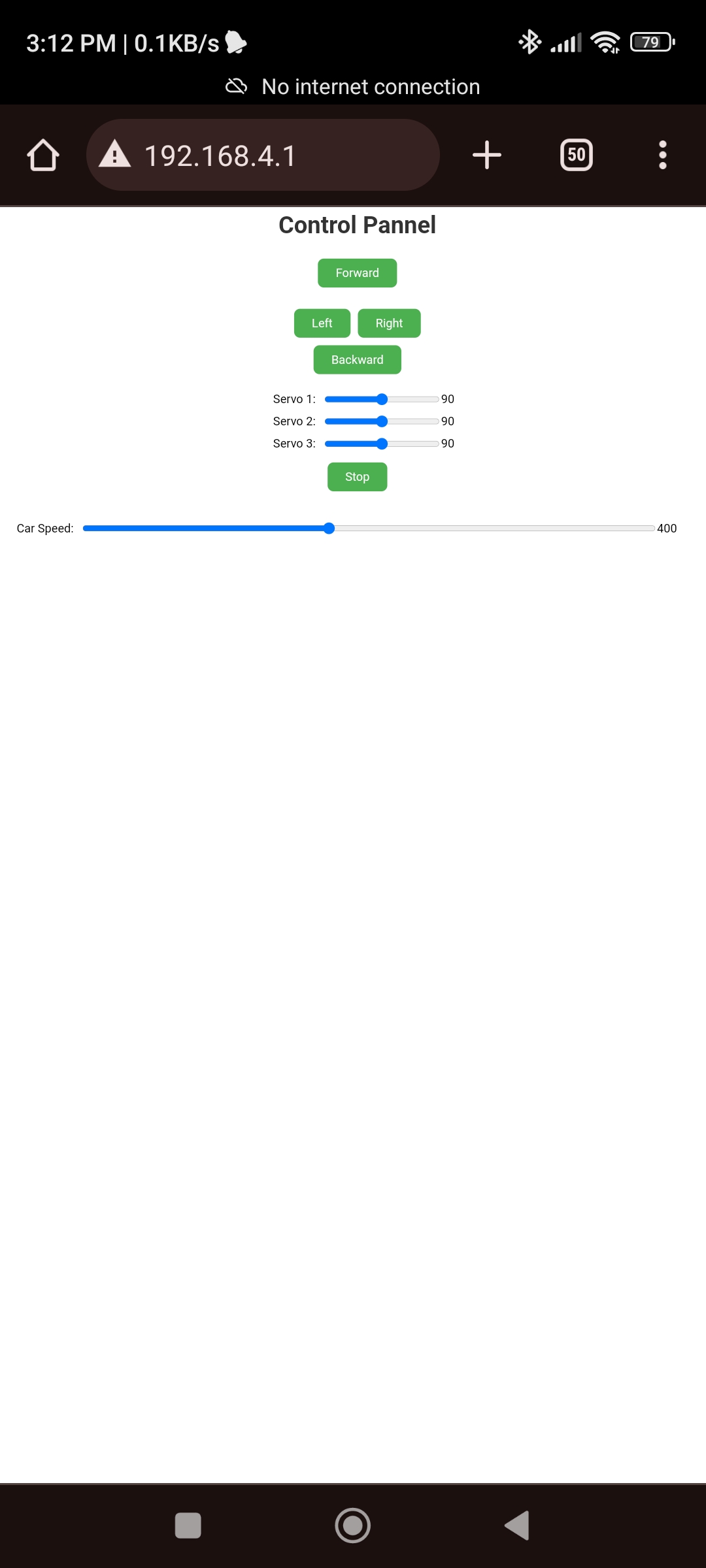
## Web Interface:

- Implements a basic login mechanism for the web interface.

- Generates an HTML-based control panel with buttons and sliders for forward, backward, left, right movements, and servo control.

- Utilizes JavaScript to send asynchronous XMLHttpRequests to the server when interacting with the controls.





## Handling HTTP Requests:

- **HTTP\_handleRoot():** Manages the root endpoint, serving the main control panel page.

- **HTTP\_handleLogin():** Handles login requests, authenticating users with a predefined username and password.

- **HTTP\_handleControl()**: Processes control commands received through the "/control" endpoint, adjusting car movement, speed, and servo positions accordingly.

## Loop Function:

- Calls **server.handleClient()** in the loop to continuously handle incoming HTTP requests.

This code essentially creates a web-based interface for users to control the NodeMCU car and its 3-servo arm. Users can adjust car movement, speed, and servo positions through a browser, providing a user-friendly means of interacting with the IoT device.