ObstaLUX: Wi-Fi Controlled Car with Dynamic Obstacle Avoidance and Auto Light Control

---

Introduction

ObstaLUX is a smart robotic vehicle built on the NodeMCU (ESP8266) platform, capable of being remotely controlled via Wi-Fi. Designed to be responsive and autonomous, it integrates obstacle detection using an infrared (IR) sensor and automatically controls headlights based on ambient lighting conditions detected by a light-dependent resistor (LDR). This fusion of manual and automatic control makes it ideal for prototyping smart vehicle concepts.

Problem Statement

Traditional robotic cars often lack dynamic responsiveness and environmental interaction. Manual control systems are constrained by limited range, and many lack safety or adaptability in changing conditions. There is a need for a low-cost, IoT-enabled robotic vehicle that:

- Offers seamless wireless control.
- Avoids obstacles autonomously to prevent collisions.
- Adapts lighting based on environmental brightness.

Objective

- To develop a Wi-Fi-based robotic vehicle using NodeMCU that can be controlled through a web interface.
- To implement real-time obstacle detection and automatic stopping.
- To control LED headlights automatically in response to ambient light intensity.

Components Used

Sensors

- **IR Sensor** – Detects nearby obstacles.
- **LDR (Light Dependent Resistor)** – Measures light intensity for headlight control.

Actuators

- **2 DC Motors** – Drive the vehicle in different directions.
- **LED** – Serves as the headlight that turns on in dark conditions.

Others

- **NodeMCU (ESP8266)** – Wi-Fi-enabled microcontroller, handles control logic and server functions.
- **L298N Motor Driver** – Interfaces DC motors with NodeMCU for direction and speed control.
- **Chassis and Wheels** – Mechanical base for movement.
- **Battery Pack** – Powers the motors and control unit.

Major Components and Their Functionalities

Component	Function
NodeMCU ESP8266	Central control unit and web server for remote command handling.
L298N Motor Driver	Motor driving circuit for directional control and speed modulation.
IR Sensor	Detects obstacles and triggers stop mechanism to avoid collision.
LDR Sensor	Senses light level to control the car's headlight.
LED Headlight	Turns ON/OFF automatically based on ambient light sensed by the LDR.

Pin Configuration Details

NodeMCU Pin	Label	Component	Function
D5 (GPIO14)	ENA	L298N	Controls speed of right motor
D6 (GPIO12)	ENB	L298N	Controls speed of left motor
D8 (GPIO15)	IN1	L298N	Direction control (right motor)
D7 (GPIO13)	IN2	L298N	Direction control (right motor)
D4 (GPIO2)	IN3	L298N	Direction control (left motor)
D3 (GPIO0)	IN4	L298N	Direction control (left motor)
A0	LDR	LDR Sensor	Analog light input
D2 (GPIO4)	LED	LED Headlight	Output for headlight control
D1 (GPIO5)	IR	IR Sensor	Obstacle detection input

Wiring Connection Overview

- **Motor Driver (L298N)**
  - IN1 → D8, IN2 → D7 (Right Motor)
  - IN3 → D4, IN4 → D3 (Left Motor)
  - ENA → D5, ENB → D6 (PWM control for motor speed)
- **IR Sensor**: OUT → D1 (GPIO5)
- **LDR Sensor**: VCC → 3.3V, GND → GND, OUT → A0 (Analog Input)
- **LED Headlight**: Anode → D2 through resistor, Cathode → GND
- **Power**: Motors powered via L298N using external battery; NodeMCU powered via USB or regulated source

Working Mechanism

A. Wi-Fi Controlled Car

- NodeMCU runs a local web server accessible via mobile/PC.
- Commands like “F”, “B”, “L”, “R” sent over HTTP.
- Based on commands, it triggers motor direction and PWM speed via the L298N module.

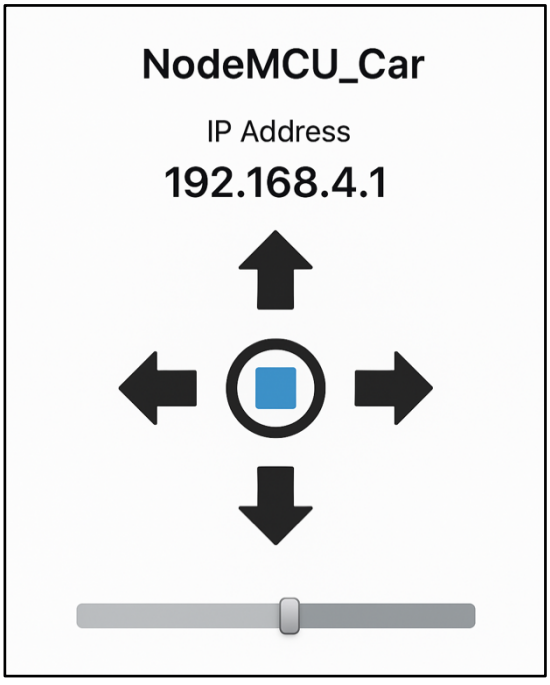
B. Dynamic Obstacle Avoidance

- The IR sensor outputs `LOW` when an object is detected.
- When `LOW` is read from the IR pin, motion is stopped by invoking `stopRobot()`.
- Ensures real-time collision avoidance.

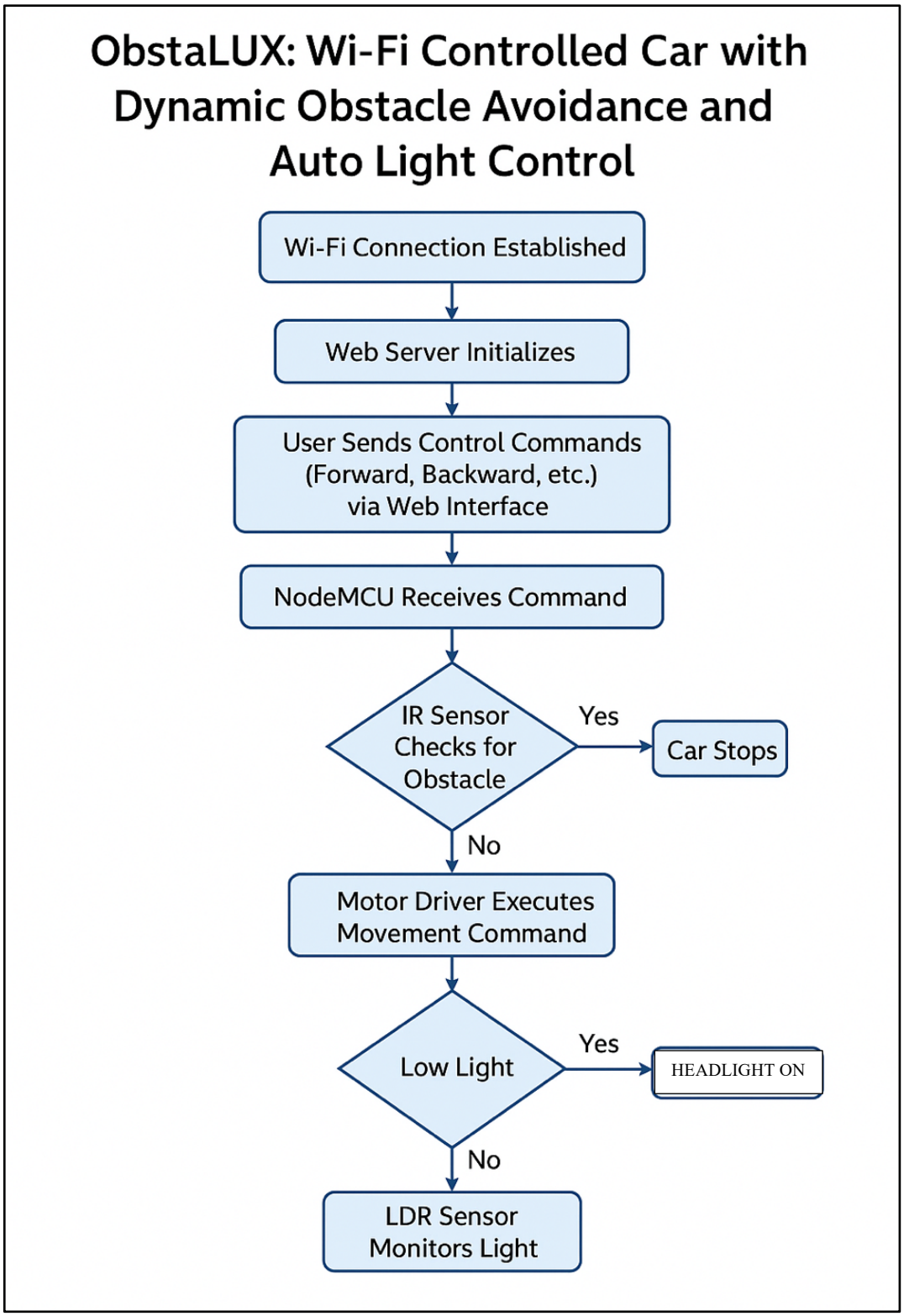
C. Auto Light Control

- LDR reads light intensity (0–1023 scale).
- If value < 200 (low light), LED headlight is turned ON.
- If value > 200 (bright light), LED turns OFF.
- Simulates intelligent lighting behaviour in dark areas.

Mobile Interface



Workflow Diagram



Source Code

```
#define ENA   14         // Enable/speed motors Right        GPIO14(D5)
#define ENB   12         // Enable/speed motors Left         GPIO12(D6)
#define IN_1  15         // L298N in1 motors Right           GPIO15(D8)
#define IN_2  13         // L298N in2 motors Right           GPIO13(D7)
#define IN_3  2          // L298N in3 motors Left            GPIO2(D4)
#define IN_4  0          // L298N in4 motors Left            GPIO0(D3)


#define LDR_PIN A0       // LDR connected to analog pin A0
#define LED_PIN 4        // LED connected to GPIO4 (D2)
#define IR_SENSOR_PIN 5   // IR sensor input - GPIO5 (D1)


#include <ESP8266WiFi.h>
#include <WiFiClient.h>
#include <ESP8266WebServer.h>


String command;
int speedCar = 800;       // Default speed
int speed_Coeff = 3;
bool irDetected = false;


const char* ssid = "NodeMCU Car";
ESP8266WebServer server(80);
```

```
void setup() {
  pinMode(ENA, OUTPUT);
  pinMode(ENB, OUTPUT);
  pinMode(IN_1, OUTPUT);
  pinMode(IN_2, OUTPUT);
  pinMode(IN_3, OUTPUT);
  pinMode(IN_4, OUTPUT);

  pinMode(LED_PIN, OUTPUT);
  pinMode(IR_SENSOR_PIN, INPUT);  // IR sensor

  Serial.begin(115200);

  WiFi.mode(WIFI_AP);
  WiFi.softAP(ssid);
  Serial.print("AP IP address: ");
  Serial.println(WiFi.softAPIP());

  server.on("/", HTTP_handleRoot);
  server.onNotFound(HTTP_handleRoot);
  server.begin();
}
```

```
// Movement Functions
void goAhead() {
  if (irDetected) {
    stopRobot();
    return;
  }
  digitalWrite(IN_1, LOW); digitalWrite(IN_2, HIGH); analogWrite(ENA, speedCar);
  digitalWrite(IN_3, LOW); digitalWrite(IN_4, HIGH); analogWrite(ENB, speedCar);
}

void goBack() {
  digitalWrite(IN_1, HIGH); digitalWrite(IN_2, LOW); analogWrite(ENA, speedCar);
  digitalWrite(IN_3, HIGH); digitalWrite(IN_4, LOW); analogWrite(ENB, speedCar);
}

void goRight() {
  digitalWrite(IN_1, HIGH); digitalWrite(IN_2, LOW); analogWrite(ENA, speedCar);
  digitalWrite(IN_3, LOW);  digitalWrite(IN_4, HIGH); analogWrite(ENB, speedCar);
}

void goLeft() {
  digitalWrite(IN_1, LOW);  digitalWrite(IN_2, HIGH); analogWrite(ENA, speedCar);
  digitalWrite(IN_3, HIGH); digitalWrite(IN_4, LOW); analogWrite(ENB, speedCar);
```

```
}

void goAheadRight() {
  if (irDetected) {
    stopRobot();
    return;
  }
  digitalWrite(IN_1, LOW); digitalWrite(IN_2, HIGH); analogWrite(ENA, speedCar / speed_Coeff);
  digitalWrite(IN_3, LOW); digitalWrite(IN_4, HIGH); analogWrite(ENB, speedCar);
}

void goAheadLeft() {
  if (irDetected) {
    stopRobot();
    return;
  }
  digitalWrite(IN_1, LOW); digitalWrite(IN_2, HIGH); analogWrite(ENA, speedCar);
  digitalWrite(IN_3, LOW); digitalWrite(IN_4, HIGH); analogWrite(ENB, speedCar / speed_Coeff);
}

void goBackRight() {
  digitalWrite(IN_1, HIGH); digitalWrite(IN_2, LOW); analogWrite(ENA, speedCar / speed_Coeff);
```

```
  digitalWrite(IN_3, HIGH); digitalWrite(IN_4, LOW); analogWrite(ENB, speedCar);
}

void goBackLeft() {
  digitalWrite(IN_1, HIGH); digitalWrite(IN_2, LOW); analogWrite(ENA, speedCar);
  digitalWrite(IN_3, HIGH); digitalWrite(IN_4, LOW); analogWrite(ENB, speedCar /
speed_Coeff);
}

void stopRobot() {
  digitalWrite(IN_1, LOW); digitalWrite(IN_2, LOW); analogWrite(ENA, 0);
  digitalWrite(IN_3, LOW); digitalWrite(IN_4, LOW); analogWrite(ENB, 0);
}

void HTTP_handleRoot(void) {
  if (server.hasArg("State")) {
    Serial.println(server.arg("State"));
  }
  server.send(200, "text/html", "");
  delay(1);
}

void loop() {
  server.handleClient();
```

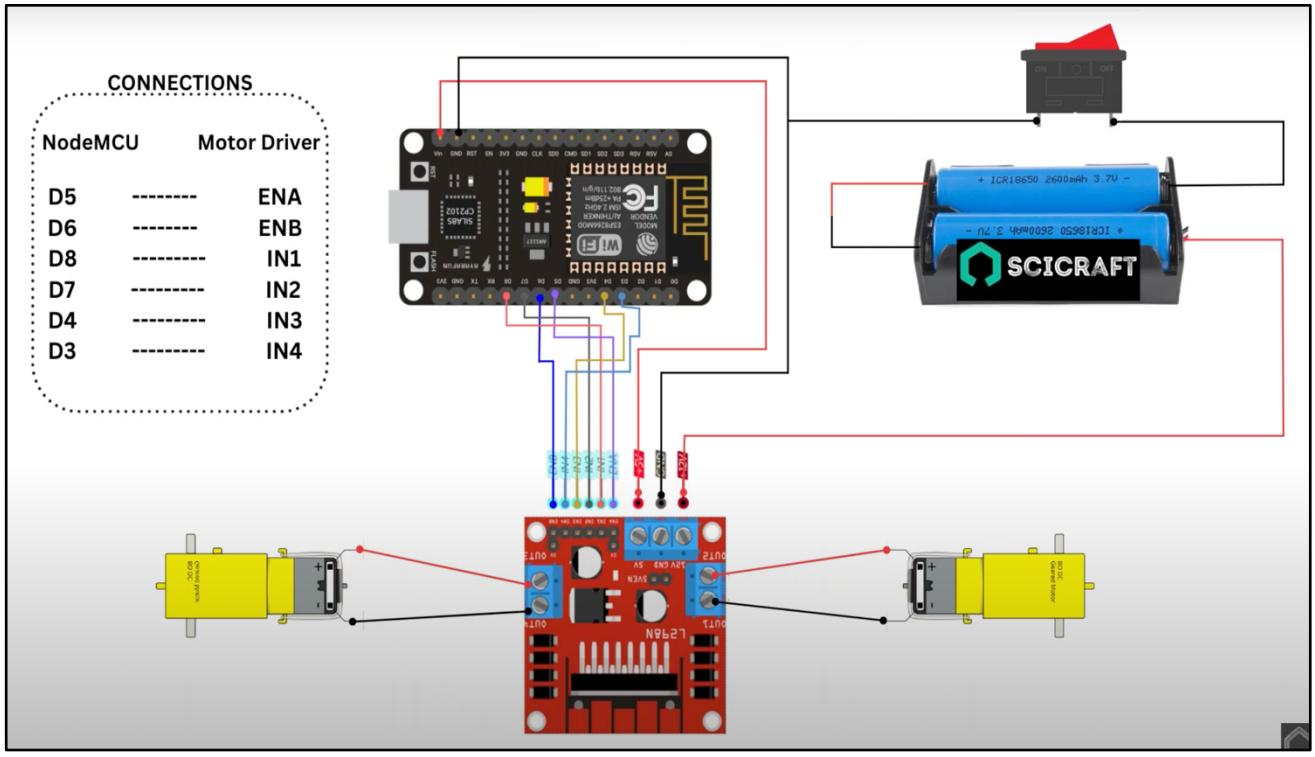
```
// IR Sensor
irDetected = (digitalRead(IR_SENSOR_PIN) == LOW);
Serial.print("IR Obstacle: ");
Serial.println(irDetected ? "Detected" : "Clear");

// LDR LED Control
int ldrValue = analogRead(LDR_PIN);
Serial.print("LDR: "); Serial.println(ldrValue);
if (ldrValue > 200) digitalWrite(LED_PIN, LOW);
else digitalWrite(LED_PIN, HIGH);

// Web Command Control
command = server.arg("State");
if (command == "F") goAhead();
else if (command == "B") goBack();
else if (command == "L") goLeft();
else if (command == "R") goRight();
else if (command == "I") goAheadRight();
else if (command == "G") goAheadLeft();
else if (command == "J") goBackRight();
else if (command == "H") goBackLeft();
else if (command == "0") speedCar = 400;
```

```
  else if (command == "1") speedCar = 470;
  else if (command == "2") speedCar = 540;
  else if (command == "3") speedCar = 610;
  else if (command == "4") speedCar = 680;
  else if (command == "5") speedCar = 750;
  else if (command == "6") speedCar = 820;
  else if (command == "7") speedCar = 890;
  else if (command == "8") speedCar = 960;
  else if (command == "9") speedCar = 1023;
  else if (command == "S") stopRobot();
  delay(100);
}
```

Circuit Diagram



Results and Observations

- The vehicle responds reliably to Wi-Fi commands with minimal delay.
- The IR-based obstacle detection halts the car effectively, avoiding collisions.
- The LDR sensor consistently toggles the LED in changing lighting conditions.
- Speed adjustment works across defined levels (0–9), making motion smooth and tuneable.

Conclusion

ObstaLUX demonstrates a functional and intelligent robotic car system capable of remote control, autonomous safety, and environmental adaptability. It is a low-cost and scalable prototype ideal for smart automation and IoT learning applications. The fusion of manual web-based control and automated sensory behaviour makes it a well-rounded, interactive engineering solution.