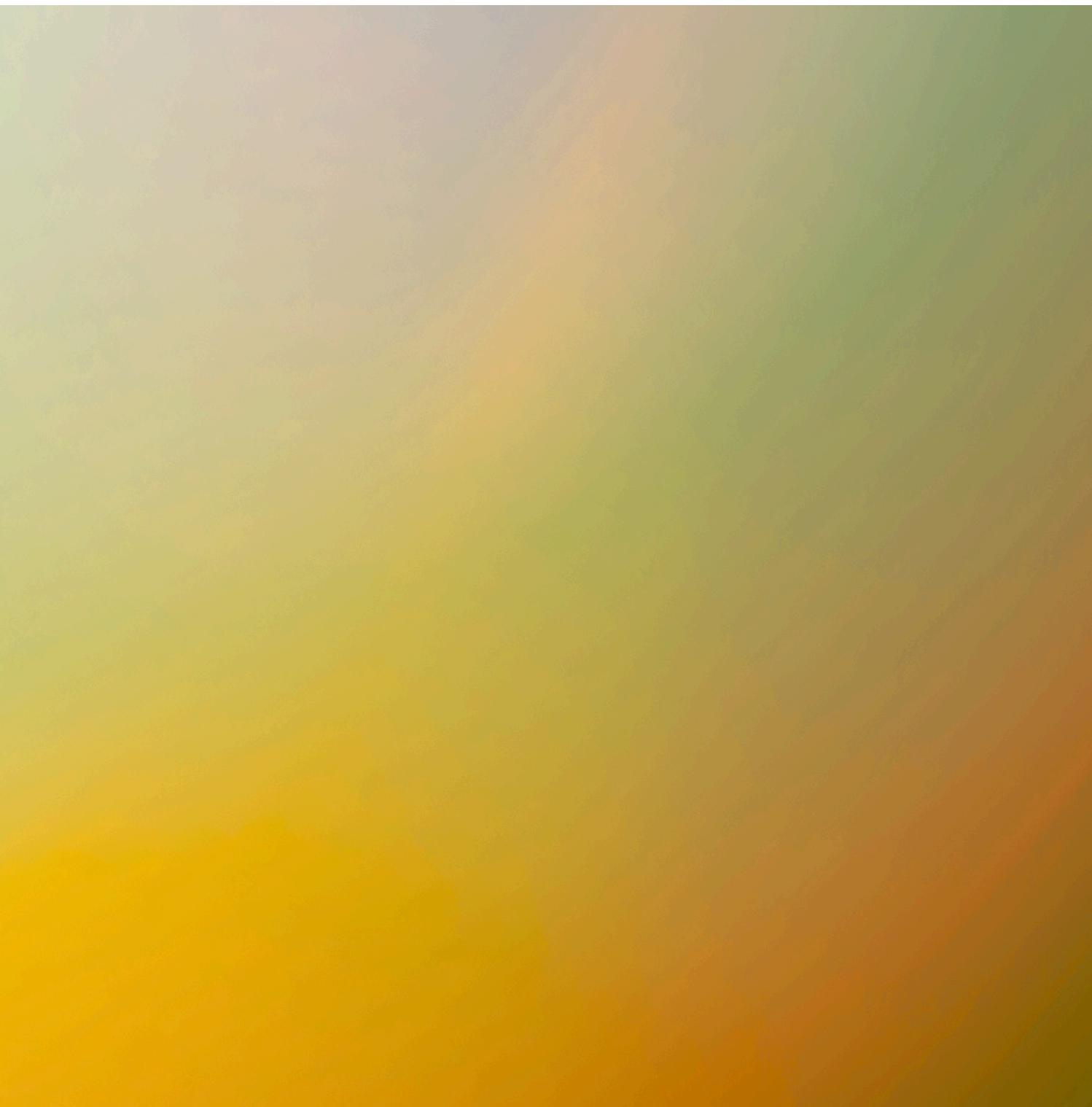


HSHL

# PROTOTYPING

ELE-4

Muhammad Ali  
Moeez Mufti



# INTRODUCTION

**Goal:** Develop an autonomous robot car that can follow a line, detect obstacles and identify their color.

**Key Features:**

- Accurate line tracking using Infrared (IR) sensors
- Real-time obstacle detection with an ultrasonic sensor
- Color detection of obstacles using a color sensor
- Smart obstacle avoidance with a servo-mounted scanning sensor

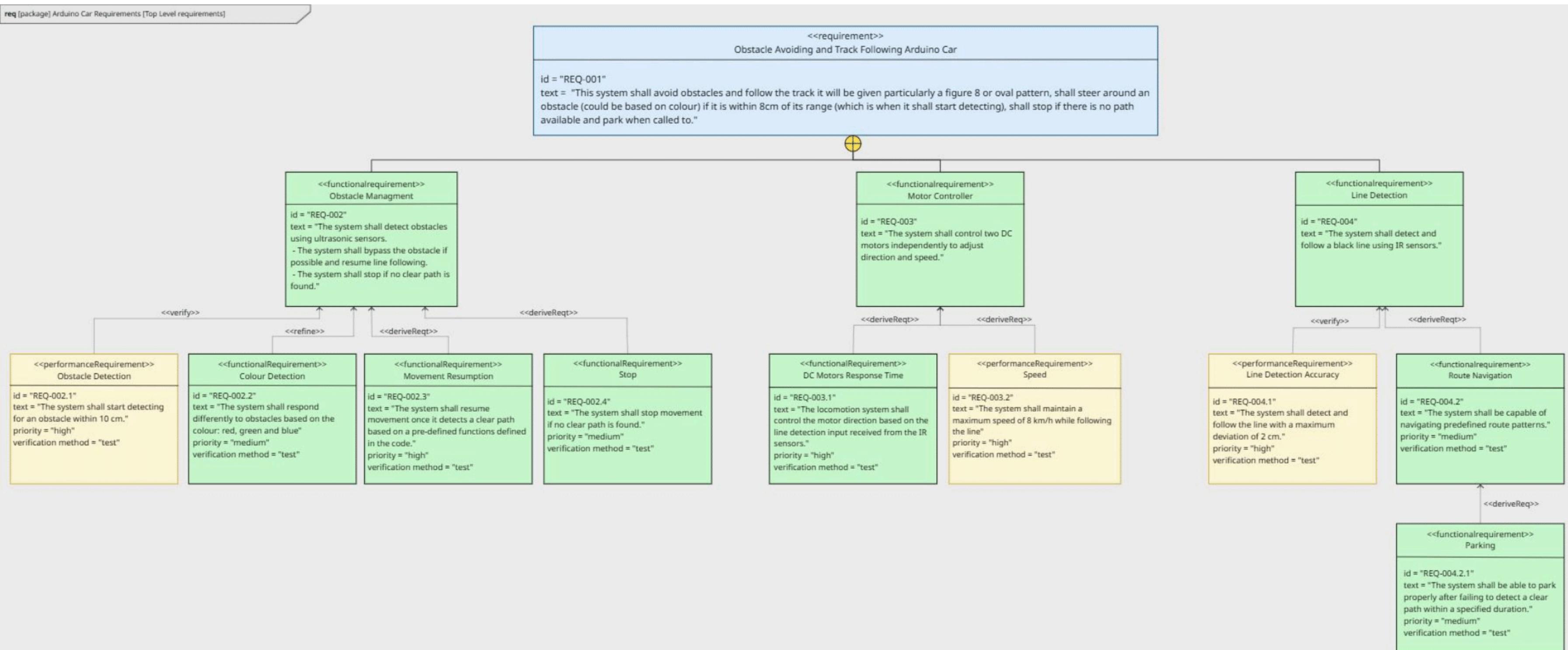
**Technology:**

- Controlled by an Arduino R4 Uno microcontroller
- Driven by DC motors & L298N motor driver module
- Integrated sensors: IR, Ultrasonic, Servo, and Color sensor

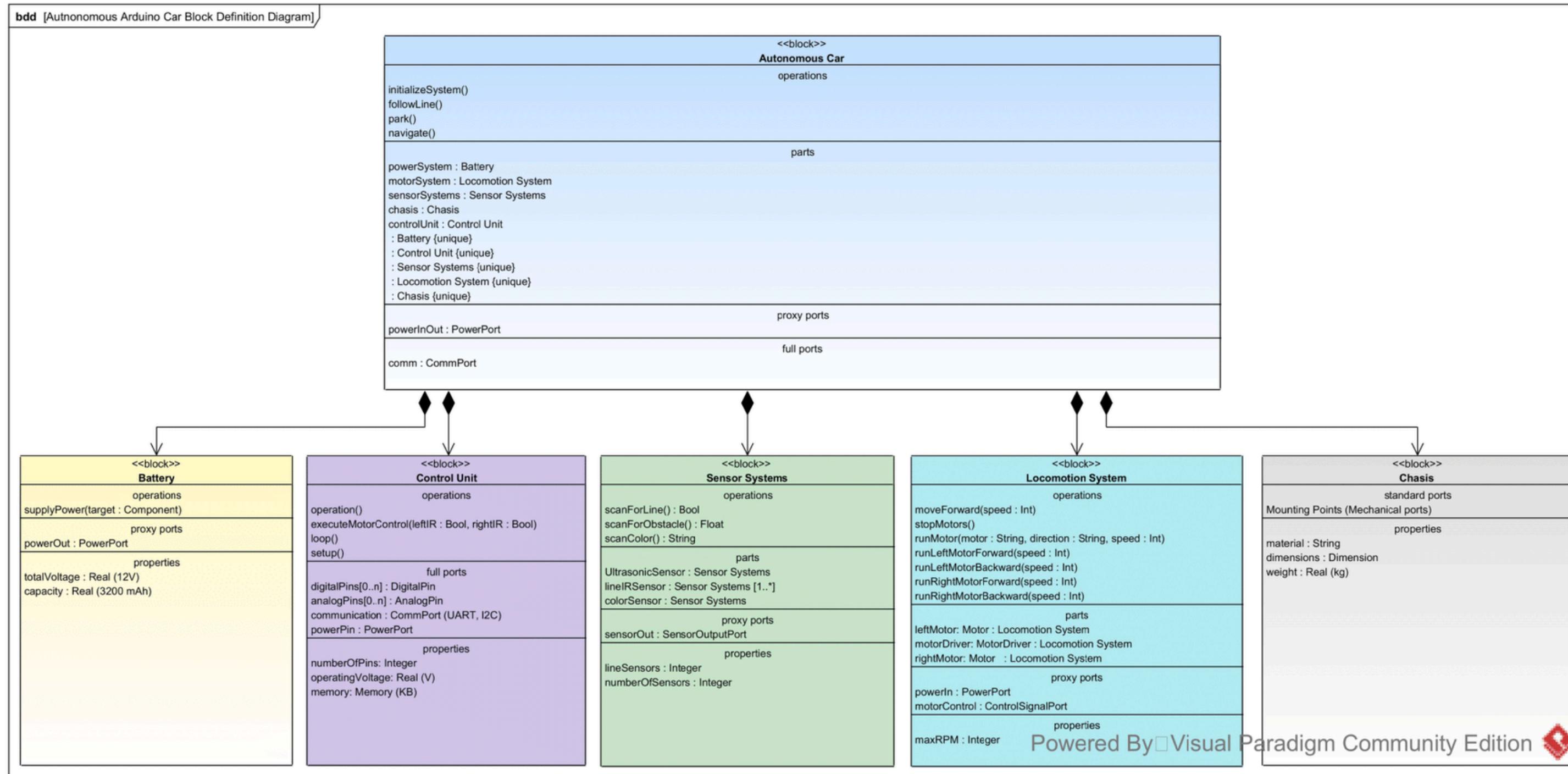
**Purpose:** Showcases principles of autonomous navigation, multi-sensor data fusion, and intelligent decision-making in embedded systems.

**Applications:** Robotics education, prototype for self-driving vehicle behavior, and groundwork for more complex autonomous robots.

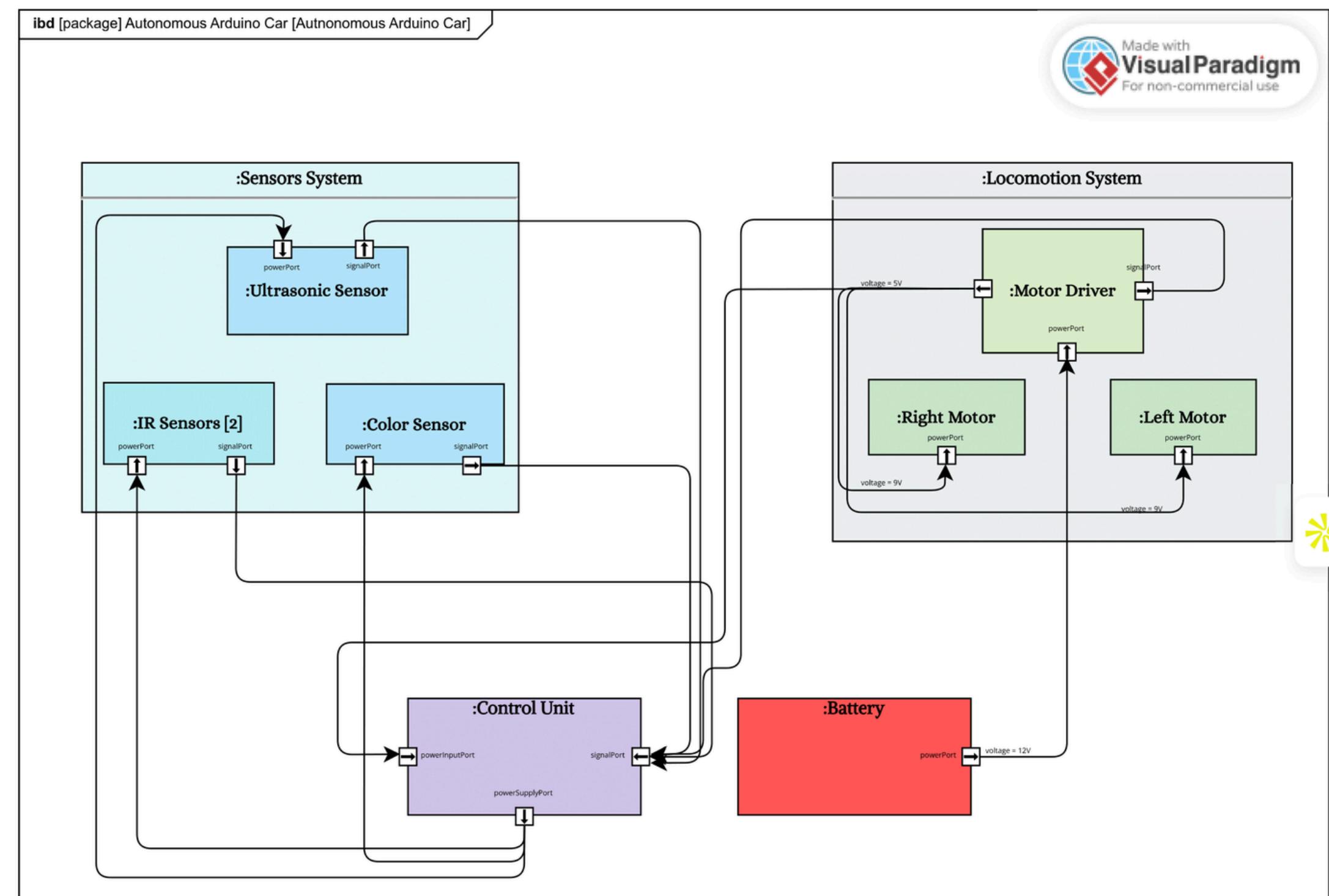
# REQUIREMENTS DIAGRAM



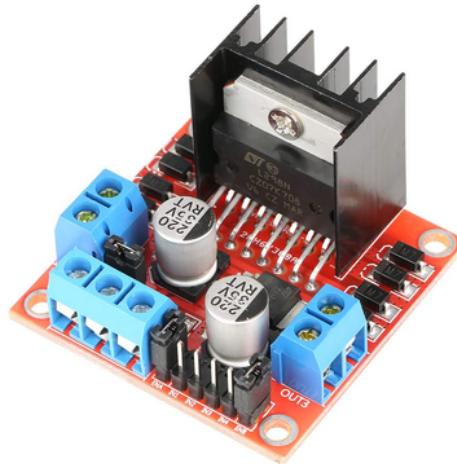
# BLOCK DEFINITION DIAGRAM



# INTERNAL BLOCK DIAGRAM



# COMPONENTS



**L298N**



**Ultrasonic  
Sensor**



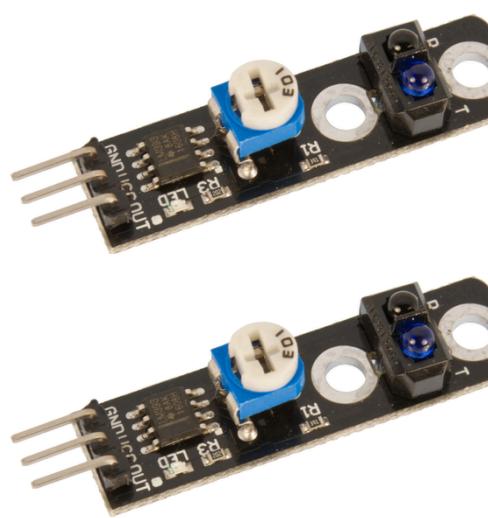
**Color  
Sensor**



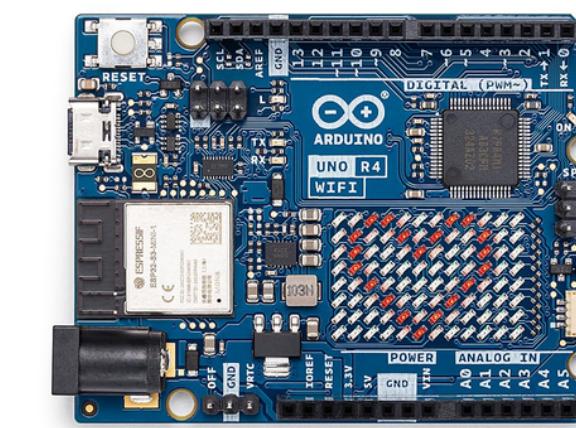
**Wheels &  
DC Motors**



**Servo Motor**



**IR Sensors**

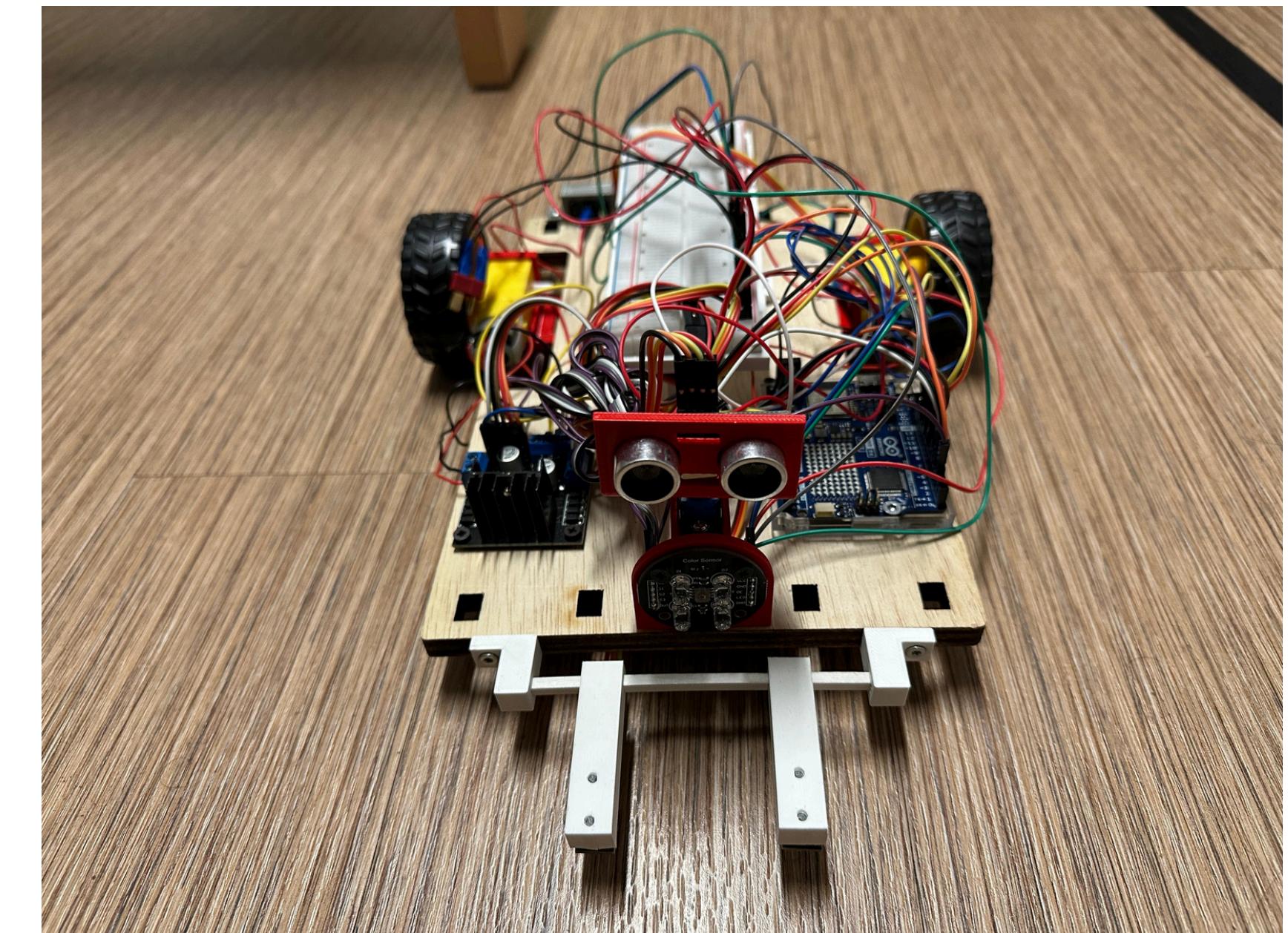
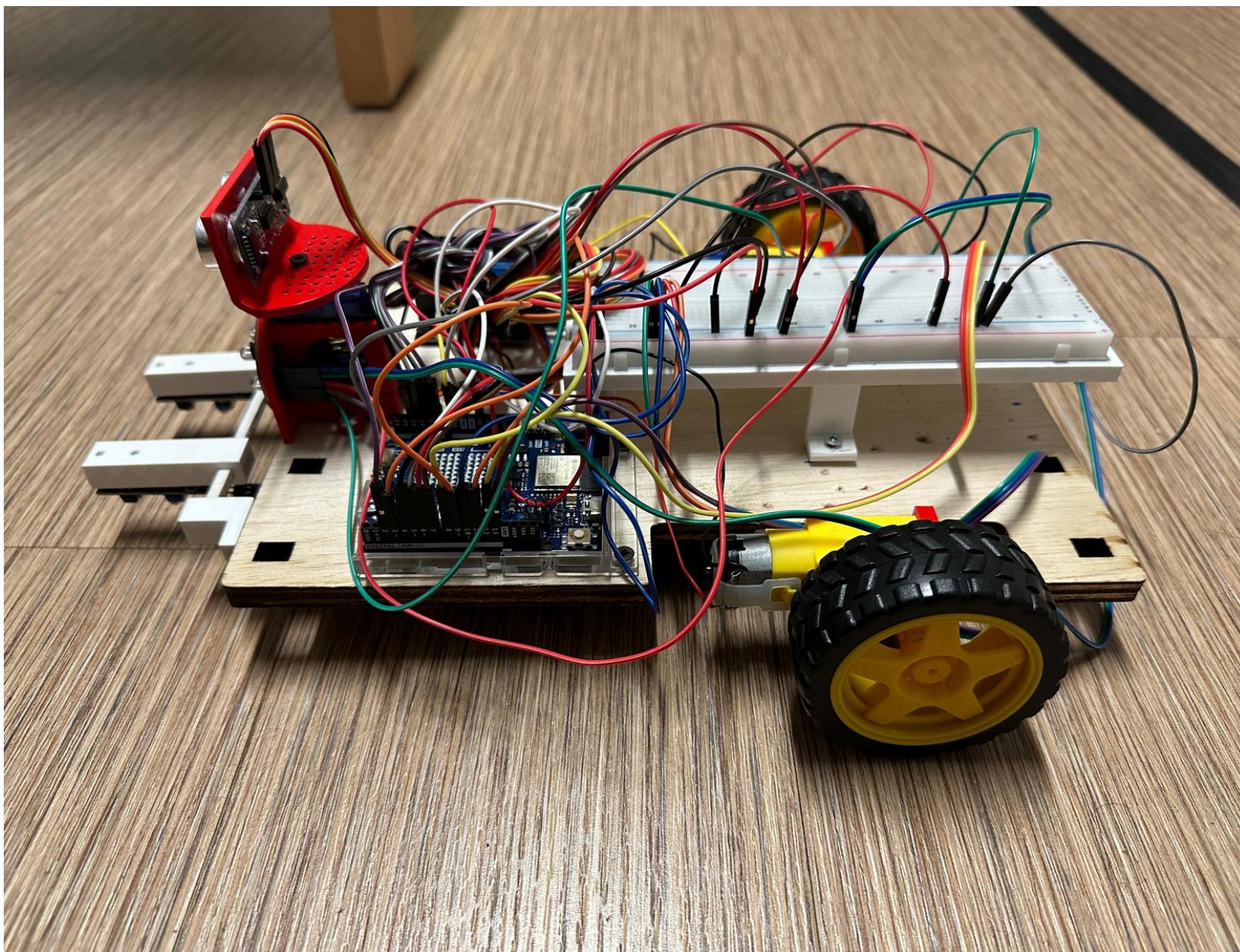


**Arduino Uno R4**

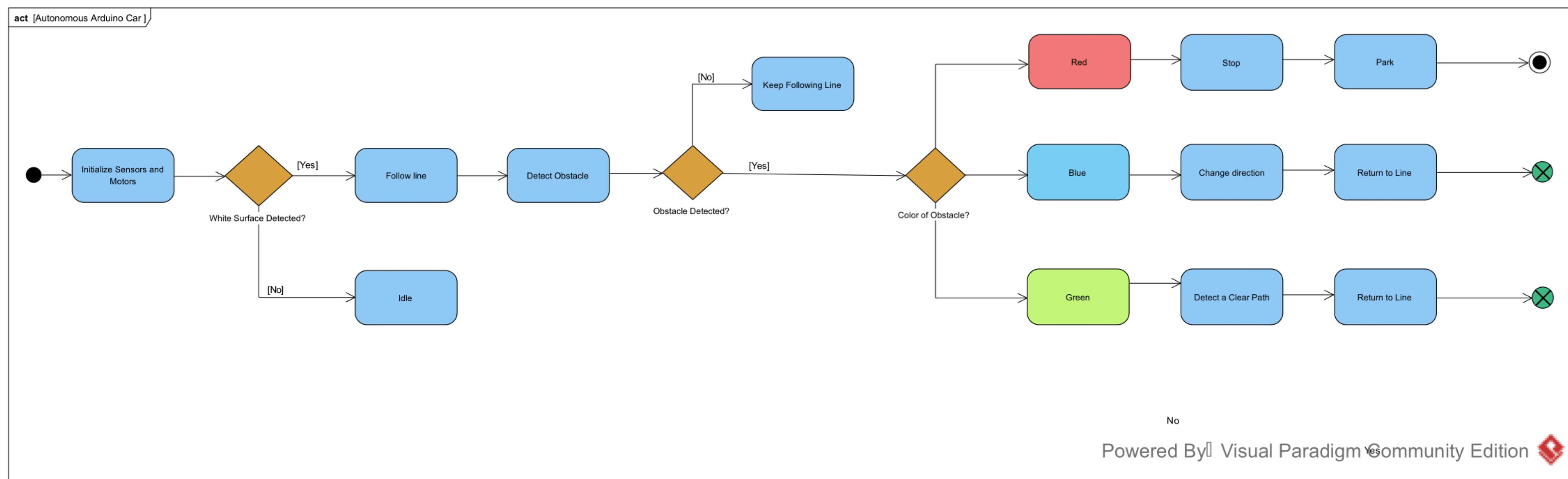


**LiPo Battery**

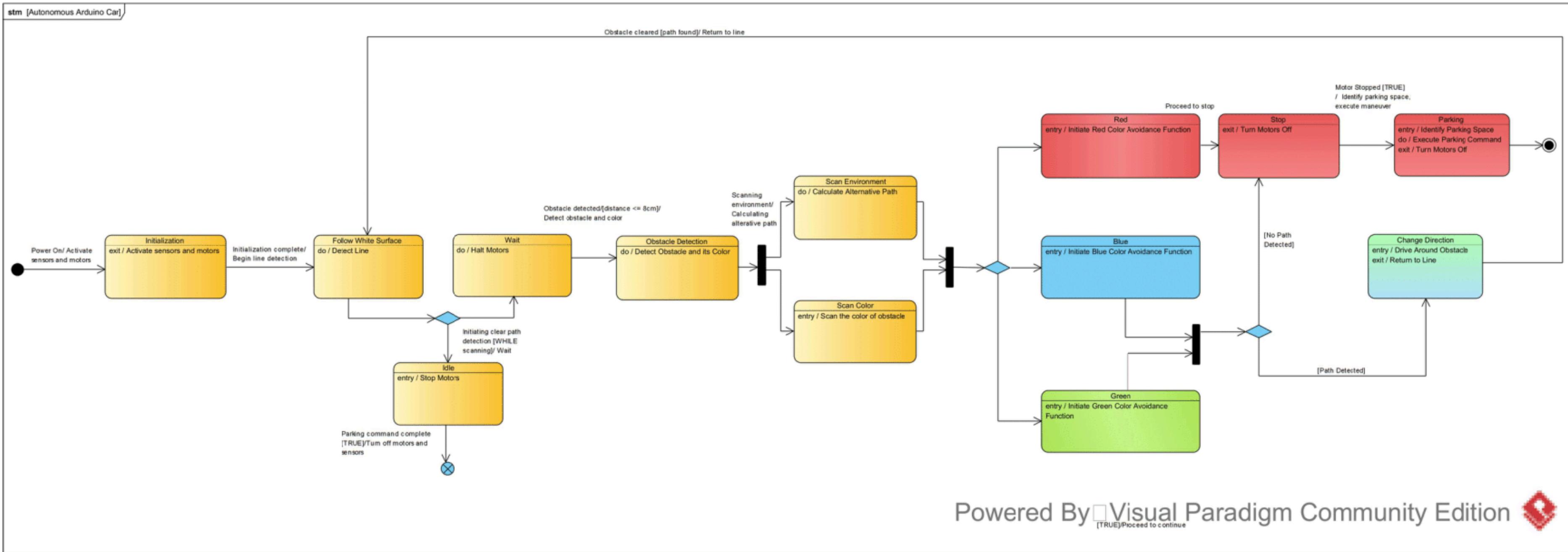
# CAR ASSEMBLY



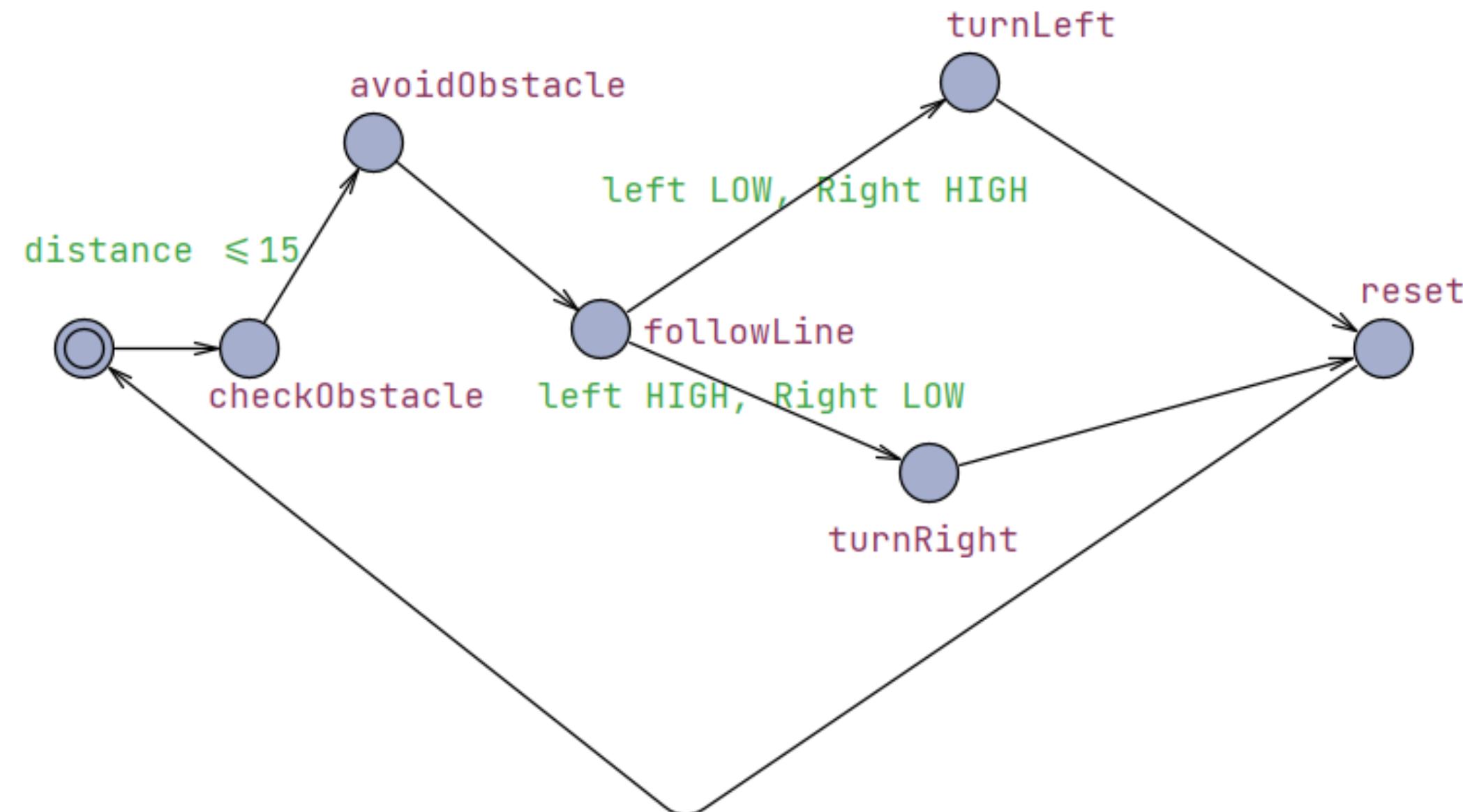
# ACTIVITY DIAGRAM



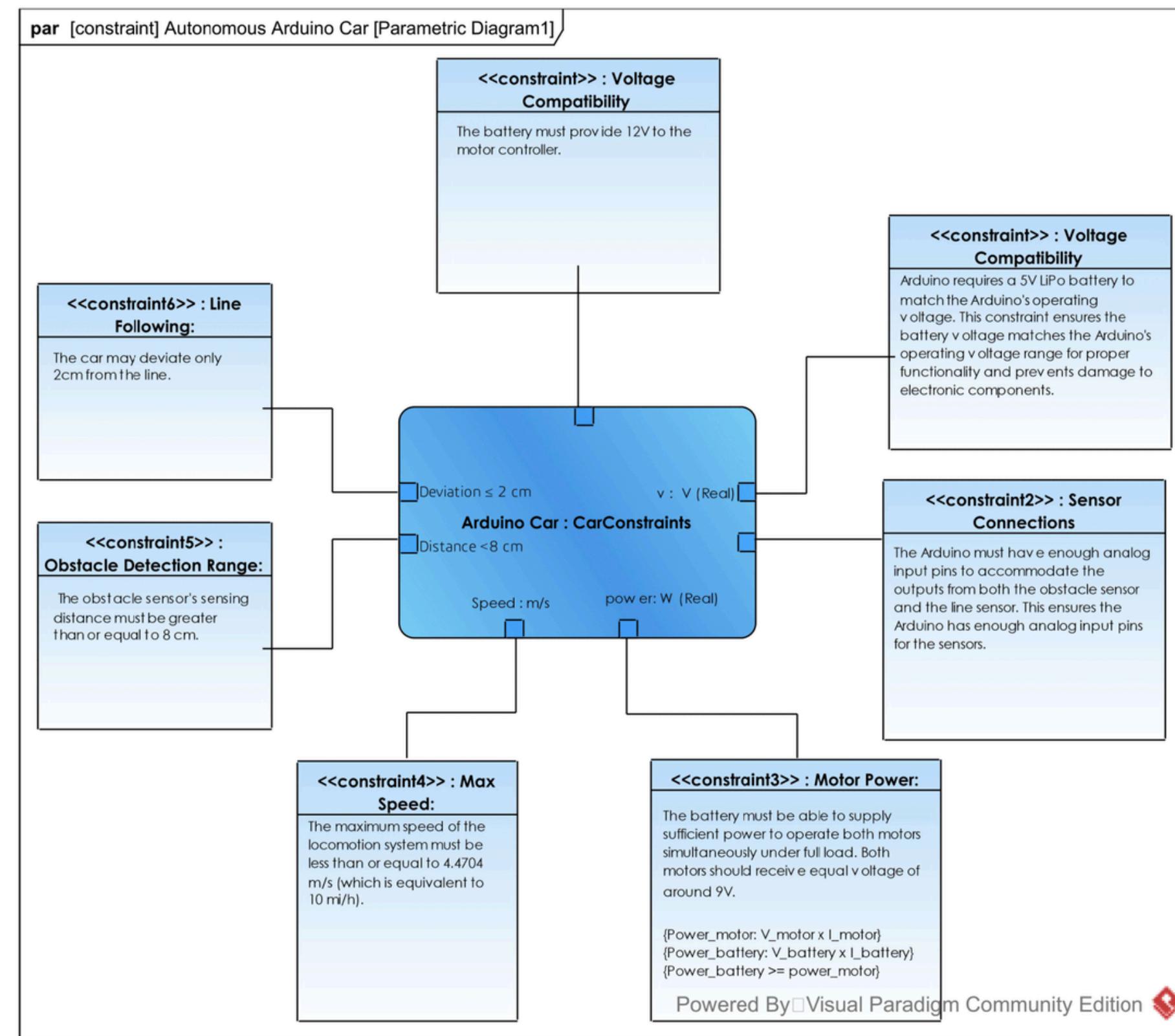
# STATE CHART



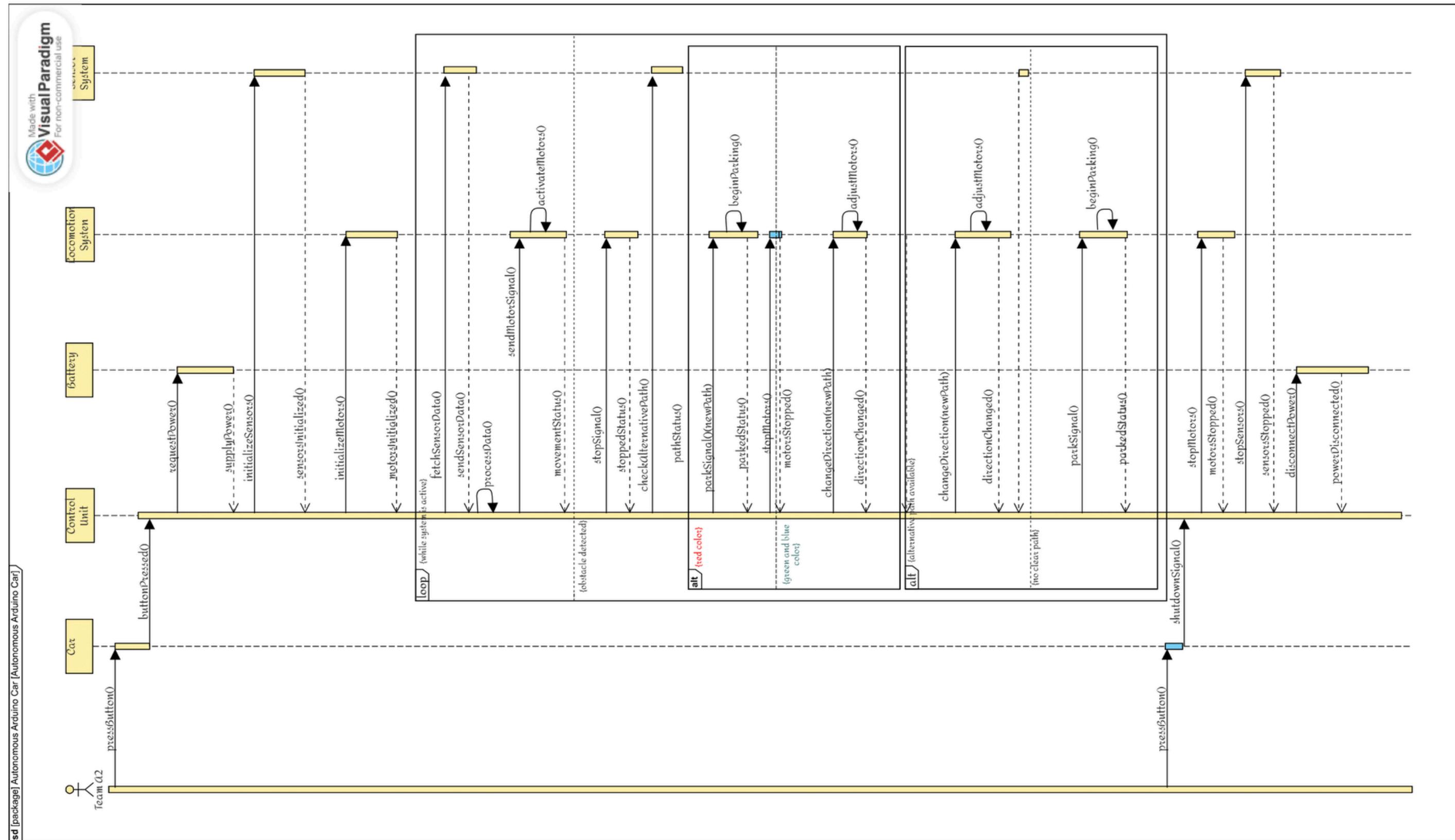
# STATE MACHINE: UPPAAL



# PARAMETRIC DIAGRAM



# SEQUENCE DIAGRAM



# CODE SOLUTION

```
1 //*****
2 * Line Following + Obstacle Avoidance +
3 * Color Detection
4 * Group A2 - Prototyping
5 * SS25 HSHL
6 *****/
7
8 // Pin Definitions
9
10 // Motor Definitions
11 #define ENA 10 // Right motor speed
12 #define IN1 4 // Right motor direction
13 #define IN2 5 // Right motor direction
14
15 #define ENB 11 // Left motor speed
16 #define IN3 6 // Left motor direction
17 #define IN4 7 // Left motor direction
18
19 // IR Sensors
20 #define IR_LEFT 12
21 #define IR_RIGHT 13
22
23 // Ultrasonic sensor
24 #define TRIG_PIN 9
25 #define ECHO_PIN 8
26
27 // TCS3200 Color Sensor pins
28 #define S0 2
29 #define S1 3
30 #define S2 A0
31 #define S3 A1
32 #define OUT A2
33 #define OE A3 // Output Enable (LOW = sensor ON)
34 #define LED A4 // LED control (HIGH = ON)
35
36 // Adjustable speed
37 int motorSpeed = 65; // 0-255
38
39 void setup() {
40   Serial.begin(9600);
41
42   // Motor pins
43   pinMode(IN1, OUTPUT);
44   pinMode(IN2, OUTPUT);
45   pinMode(ENA, OUTPUT);
46   pinMode(IN3, OUTPUT);
47   pinMode(IN4, OUTPUT);
48   pinMode(ENB, OUTPUT);
```

```
50   // IR Sensors
51   pinMode(IR_LEFT, INPUT);
52   pinMode(IR_RIGHT, INPUT);
53
54   // Ultrasonic
55   pinMode(TRIG_PIN, OUTPUT);
56   pinMode(ECHO_PIN, INPUT);
57
58   // TCS3200
59   pinMode(S0, OUTPUT);
60   pinMode(S1, OUTPUT);
61   pinMode(S2, OUTPUT);
62   pinMode(S3, OUTPUT);
63   pinMode(OE, OUTPUT);
64   pinMode(LED, OUTPUT);
65   pinMode(OUT, INPUT);
66
67   // Configure TCS3200 frequency scaling
68   digitalWrite(S0, HIGH);
69   digitalWrite(S1, HIGH);
70   digitalWrite(OE, LOW); // Enable output
71   digitalWrite(LED, HIGH); // Turn on LEDs
72
73   stopMotors();
74 }
75
76 void loop() {
77   long distance = getDistance();
78
79   if (distance < 15 && distance > 0) {
80     stopMotors();
81     delay(200);
82
83     detectColor();
84     avoidObstacle();
85   } else {
86     followLine();
87   }
88 }
```

```
90   // Modular
91
92   // Line Following
93
94   void followLine() {
95     int leftIR = digitalRead(IR_LEFT);
96     int rightIR = digitalRead(IR_RIGHT);
97
98     // Case 1: Both LOW -> Both motors forward
99
100    if (leftIR == LOW & rightIR == LOW) {
101      moveForward();
102    }
103
104    // Case 2: Both HIGH -> Both motors stopped
105    else if (leftIR == HIGH & rightIR == HIGH) {
106      stopMotors();
107    }
108
109    // Case 3: Mismatch - run motor on side with LOW sensor only
110    else {
111      if (leftIR == LOW & rightIR == HIGH) {
112        runRightMotor();
113        stopLeftMotor();
114      }
115
116      else if (leftIR == HIGH & rightIR == LOW) {
117        runLeftMotor();
118        stopRightMotor();
119      }
120
121    void moveForward() {
122      digitalWrite(IN1, HIGH);
123      digitalWrite(IN2, LOW);
124      analogWrite(ENA, motorSpeed);
125
126      digitalWrite(IN3, LOW);
127      digitalWrite(IN4, HIGH);
128      analogWrite(ENB, motorSpeed);
129    }
130
131    void stopMotors() {
132      stopLeftMotor();
133      stopRightMotor();
134    }
135
136    void runLeftMotor() {
137      digitalWrite(IN3, HIGH);
138      digitalWrite(IN4, LOW);
139      analogWrite(ENB, motorSpeed);
140  }
```

```
141
142 void stopLeftMotor() {
143   digitalWrite(IN3, LOW);
144   digitalWrite(IN4, LOW);
145   analogWrite(ENB, 0);
146 }
147
148 void runRightMotor() {
149   digitalWrite(IN1, HIGH);
150   digitalWrite(IN2, LOW);
151   analogWrite(ENA, motorSpeed);
152 }
153
154 void stopRightMotor() {
155   digitalWrite(IN1, LOW);
156   digitalWrite(IN2, LOW);
157   analogWrite(ENA, 0);
158 }
159
160 // Ultrasonic sensor
161
162 long getDistance() {
163   digitalWrite(TRIG_PIN, LOW);
164   delayMicroseconds(2);
165   digitalWrite(TRIG_PIN, HIGH);
166   delayMicroseconds(10);
167   digitalWrite(TRIG_PIN, LOW);
168
169   long duration = pulseIn(ECHO_PIN, HIGH, 20000);
170   long distance = duration * 0.034 / 2;
171   return distance;
172 }
173
174 // Color detection
175
176 void detectColor() {
177   Serial.println("Detecting color...");
178
179   int red = readColor(LOW, LOW);
180   int green = readColor(HIGH, HIGH);
181   int blue = readColor(LOW, HIGH);
182
183   Serial.print("R: ");
184   Serial.print(red);
185   Serial.print(" G: ");
186   Serial.print(green);
187   Serial.print(" B: ");
188   Serial.println(blue);
```

# CODE SOLUTION

```
190  if (red < green && red < blue) {  
191    Serial.println("Detected: RED");  
192  } else if (green < red && green < blue) {  
193    Serial.println("Detected: GREEN");  
194  } else if (blue < red && blue < green) {  
195    Serial.println("Detected: BLUE");  
196  } else {  
197    Serial.println("Unknown Color");  
198  }  
199 }  
200  
201 int readColor(bool s2, bool s3) {  
202  digitalWrite(S2, s2);  
203  digitalWrite(S3, s3);  
204  delay(50);  
205  return pulseIn(OUT, LOW);  
206 }  
207  
208 // Obstacle avoidance  
209  
210 void avoidObstacle() {  
211  Serial.println("Scanning for clear path...");  
212  
213  // Turn left to check  
214  runLeftMotor();  
215  runRightMotorReverse();  
216  delay(500);  
217  stopMotors();  
218  delay(200);  
219  
220  if (getDistance() > 15) {  
221    Serial.println("Path clear left");  
222    runLeftMotor();  
223    runRightMotorReverse();  
224    delay(500);  
225    return;  
226  }  
227  
228  // Turn right to check  
229  runLeftMotorReverse();  
230  runRightMotor();  
231  delay(1000);  
232  stopMotors();  
233  delay(200);  
234  
235  if (getDistance() > 15) {  
236    Serial.println("Path clear right");  
237    runLeftMotorReverse();  
238    runRightMotor();  
239    delay(500);  
240    return;  
241 }  
242 }  
243 // Stuck - stop  
244 stopMotors();  
245 Serial.println("No clear path found");  
246 }  
247  
248 // Helper functions for reverse motors (used in avoidance)  
249  
250 void runLeftMotorReverse() {  
251  digitalWrite(IN3, LOW);  
252  digitalWrite(IN4, HIGH);  
253  analogWrite(ENB, motorSpeed);  
254 }  
255  
256 void runRightMotorReverse() {  
257  digitalWrite(IN1, LOW);  
258  digitalWrite(IN2, HIGH);  
259  analogWrite(ENA, motorSpeed);  
260 }  
261
```

# TEAM OVERVIEW

Muhammad Ali: 3D Modeling in AutoCAD, Vehicle Assembly, Initial Component Testing, Coding

Moeez Mufti: GitHub Moderation, UML Diagrams, Version Control, Component Testing, Vehicle Assembly