Cognitive Robotics Lab

Lab 3 - Arduino Smart Car Kit Basic Movements

Srikar Ganesh 22BAI1080

04-08-2025

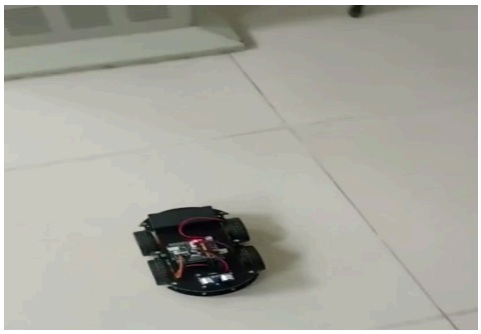
Aim:

To program and control an Arduino-based smart car kit to perform basic movements including forward motion, backward motion, left and right turns, speed acceleration, and deceleration, thereby understanding motor control, PWM (Pulse Width Modulation), and directional logic using Arduino.

Components:

1. **Arduino Uno (or compatible board)** – acts as the main microcontroller to control the smart car
2. **Smart Car Chassis Kit** – includes the frame, wheels, and motor mounts
3. **DC Motors (typically 2 or 4)** – to drive the wheels for movement
4. **Motor Driver Module (L298N or L293D)** – controls the direction and speed of the motors
5. **Battery Pack (e.g., 4x AA or Li-ion)** – provides power to the motors
6. **Jumper Wires** – for making electrical connections between components and the Arduino
7. **USB Cable (Type A to B)** – to upload code from PC to Arduino
8. **Wheels (2 or 4)** – attached to the motors to enable movement
9. **Caster Wheel** – provides balance and smooth turning for 3-wheel cars
10. **Breadboard (optional)** – for quick prototyping of connections
11. **Switch (optional)** – to turn the power supply on/off
12. **Resistors (optional)** – used if adding LEDs or other sensors
13. **LEDs (optional)** – for basic status indicators

Output:



Source Code:

```
const int ENA = 5;
const int IN1 = 8;
const int IN2 = 9;
const int ENB = 6;
const int IN3 = 10;
const int IN4 = 11;

int speedMotor = 150;

void setup() {
  pinMode(ENA, OUTPUT); pinMode(IN1, OUTPUT); pinMode(IN2, OUTPUT);
  pinMode(ENB, OUTPUT); pinMode(IN3, OUTPUT); pinMode(IN4, OUTPUT);
}

void forward(int speed) {
  digitalWrite(IN1, HIGH); digitalWrite(IN2, LOW);
  digitalWrite(IN3, HIGH); digitalWrite(IN4, LOW);
  analogWrite(ENA, speed); analogWrite(ENB, speed);
}

void turnLeft(int speed) {
  digitalWrite(IN1, LOW); digitalWrite(IN2, LOW);
  digitalWrite(IN3, HIGH); digitalWrite(IN4, LOW);
  analogWrite(ENA, 0); analogWrite(ENB, speed);
}

void turnRight(int speed) {
  digitalWrite(IN1, HIGH); digitalWrite(IN2, LOW);
  digitalWrite(IN3, LOW); digitalWrite(IN4, LOW);
  analogWrite(ENA, speed); analogWrite(ENB, 0);
}

void stopCar() {
  digitalWrite(IN1, LOW); digitalWrite(IN2, LOW);
  digitalWrite(IN3, LOW); digitalWrite(IN4, LOW);
  analogWrite(ENA, 0); analogWrite(ENB, 0);
}

void loop() {
  forward(speedMotor);
  delay(1000);

  stopCar(); delay(500);

  turnLeft(speedMotor);
```

```
  delay(700);

  stopCar(); delay(500);

  forward(speedMotor);
  delay(1000);

  stopCar(); delay(500);

  turnRight(speedMotor);
  delay(700);

  stopCar(); delay(500);

  forward(speedMotor);
  delay(1000);

  stopCar(); delay(500);
}
```

Explanation:

The code controls a two-motor Arduino-based smart car using a motor driver (like L298N). It defines motor control pins and sets up movement functions such as forward, turnLeft, turnRight, and stopCar. In the loop(), the car follows a specific movement pattern to trace the shape of the number "8". It moves forward, then makes a left turn, moves forward again, and then makes a right turn, repeating the process. After each segment of motion or turn, the car brakes (stops briefly), simulating safe, deliberate movement,mimicking how a real autonomous car might behave at intersections or curves.

Result:

When uploaded to the Arduino and connected correctly to the motor driver and motors, the smart car executes a smooth figure-8 movement on the ground. It moves forward, brakes, turns left, brakes again, and continues this pattern with a right turn following the second forward motion. The pauses introduced by the braking commands at each turn ensure that the car doesn’t skid or overshoot its path, making the movement controlled and clear. This experiment successfully demonstrates directional control, speed modulation, and motion planning using basic Arduino motor functions.