# Working: Autonomous Vacuum Cleaner Robot

**Project Overview**

This project involves a simple autonomous robot that uses DC motors for movement and an additional motor for a vacuum feature. The robot is designed to move forward continuously while simultaneously operating the vacuum.

**Key Components**

* **DC Motors**: Four motors control the locomotion of the robot, allowing it to move forward.
* **Vacuum Motor**: A dedicated motor powers the vacuum mechanism, enabling the robot to pick up debris while moving.
* **Motor Driver (Adafruit Motor Shield)**: Controls the speed and of the locomotion.
* **Arduino Board**: Acts as the main controller, executing the program logic to manage both motor functions.

**Conclusion**

This project involves a simple autonomous robot that moves forward continuously using four DC motors while simultaneously operating a vacuum motor for cleaning. The Arduino board controls both the locomotion and vacuum motors through a motor driver

## CODE

#include <AFMotor.h>

#include <Servo.h>

#define TRIG\_PIN A0

#define ECHO\_PIN A1

#define MAX\_SPEED 100 // Set speed to 100

#define MAX\_SPEED\_OFFSET 20

AF\_DCMotor motor1(1, MOTOR12\_1KHZ); AF\_DCMotor motor2(2, MOTOR12\_1KHZ);AF\_DCMotor motor3(3, MOTOR34\_1KHZ);

AF\_DCMotor motor4(4, MOTOR34\_1KHZ);Servo myservo;

boolean goesForward = false;int distance = 100;int speedSet = 0;

void setup() {

myservo.attach(10);

myservo.write(115);

delay(2000);

pinMode(TRIG\_PIN, OUTPUT);

pinMode(ECHO\_PIN, INPUT);

}

void loop() {

int distanceR = 0;

int distanceL = 0;

delay(40);

distance = readPing(); // Read the distance

if (distance <= 15) {

moveStop();

delay(100);

moveBackward();

delay(300);

moveStop();

delay(200);

distanceR = lookRight();

delay(200);

distanceL = lookLeft();

delay(200);

if (distanceR >= distanceL) {

turnRight();

moveStop();

} else {

turnLeft();

moveStop();

}

} else {

moveForward();

}

}

int lookRight() {

myservo.write(50);

delay(500);

int distance = readPing();

delay(100);

myservo.write(115);

return distance;

}

int lookLeft() {

myservo.write(170);

delay(500);

int distance = readPing();

delay(100);

myservo.write(115);

return distance;

}

int readPing() {

digitalWrite(TRIG\_PIN, LOW);

delayMicroseconds(4);

digitalWrite(TRIG\_PIN, HIGH);

delayMicroseconds(10);

digitalWrite(TRIG\_PIN, LOW);

long duration = pulseIn(ECHO\_PIN, HIGH); // Get the echo response time

int cm = duration / 29 / 2; // Convert time to distance in cm

if (cm == 0) {

cm = 250; // Set a large value if no object is detected

}

return cm;

}

void moveStop() {

motor1.run(RELEASE);

motor2.run(RELEASE);

motor3.run(RELEASE);

motor4.run(RELEASE);

}

void moveForward() {

if (!goesForward) {

goesForward = true;

motor1.run(FORWARD);

motor2.run(FORWARD);

motor3.run(FORWARD);

motor4.run(FORWARD);

for (speedSet = 0; speedSet < MAX\_SPEED; speedSet += 1) { // Gradual speed increase

motor1.setSpeed(speedSet);

motor2.setSpeed(speedSet);

motor3.setSpeed(speedSet);

motor4.setSpeed(speedSet);

delay(10); // Moderate delay for smoother acceleration

}

}

}

void moveBackward() {

goesForward = false;

motor1.run(BACKWARD);

motor2.run(BACKWARD);

motor3.run(BACKWARD);

motor4.run(BACKWARD);

for (speedSet = 0; speedSet < MAX\_SPEED; speedSet += 1) { // Gradual speed increase

motor1.setSpeed(speedSet);

motor2.setSpeed(speedSet);

motor3.setSpeed(speedSet);

motor4.setSpeed(speedSet);

delay(10); // Moderate delay for smoother acceleration

}

}

void turnRight() {

motor1.run(FORWARD);

motor2.run(FORWARD);

motor3.run(BACKWARD);

motor4.run(BACKWARD);

delay(500);

motor1.run(FORWARD);

motor2.run(FORWARD);

motor3.run(FORWARD);

motor4.run(FORWARD);

}

void turnLeft() {

motor1.run(BACKWARD);

motor2.run(BACKWARD);

motor3.run(FORWARD);

motor4.run(FORWARD);

delay(500);

motor1.run(FORWARD);

motor2.run(FORWARD);

motor3.run(FORWARD);

motor4.run(FORWARD);

}