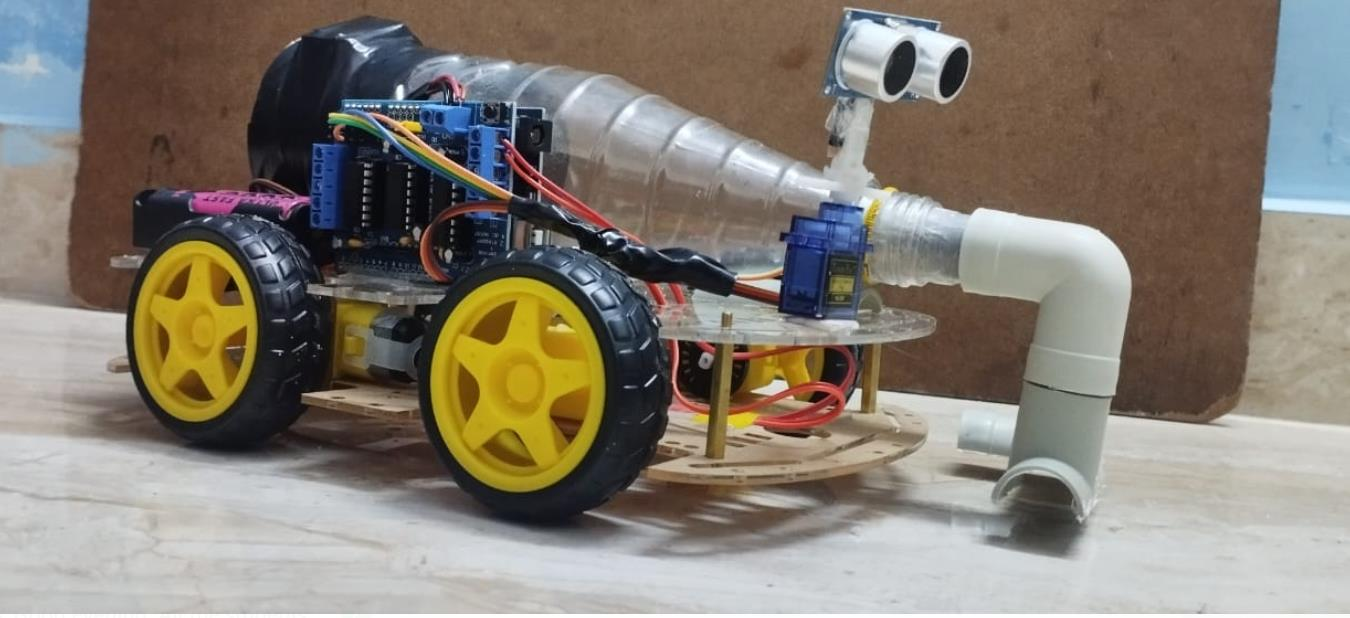
SMART VACCUM CLEANER(VC Robot)

In today’s world IoT(Internet Of Things) is a fast growing and interesting field. IoT has sufficient intelligence to cover the maximum area of provided space. Introduces the design and implementation of an autonomous Smart Vacuum Cleaner using ultrasonic wave sensor in this thesis. In the current hectic schedule, cleaning houses and surrounding environment is more arduous. At present, there are vacuum cleaners which require humans to handle it. Thus, there is a dire need to implement vacuum cleaner which works without human intervention. An efficient method to clean the desired area has been implemented through this project. This system has an Ultrasonic

sensor attached to it, that helps in avoiding large obstacles such as tables, chairs, walls

etc. The vacuum cleaner is designed with a CPU fan and a pipe is attached to the mouth of the bottle. The entire system is run by batteries.



#include <AFMotor.h>

#include <NewPing.h>

#include <Servo.h>

#define TRIG\_PIN A1

#define ECHO\_PIN A0

#define MAX\_DISTANCE 200

#define MAX\_SPEED 150 // sets speed of DC motors

#define MAX\_SPEED\_OFFSET 20

NewPing sonar(TRIG\_PIN, ECHO\_PIN, MAX\_DISTANCE);

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() {

Serial.begin(9600);

myservo.attach(10);

myservo.write(115);

delay(2000);

distance = readPing();

delay(100);

distance = readPing();

delay(100);

distance = readPing();

delay(100);

distance = readPing();

delay(100);

}

void loop() {

int distanceR = 0;

int distanceL = 0;

delay(40);

if(distance<=35)

{Page | 27

View publication stats

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();

}

distance = readPing();

}

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;

delay(100);Page | 28

View publication stats

}

int readPing() {

delay(70);

int cm = sonar.ping\_cm();

if(cm==0)

{

cm = 250;

}

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 +=2) // slowly bring the speed up to avoid

loading down the batteries too quickly

{

motor1.setSpeed(speedSet);

motor2.setSpeed(speedSet);

motor3.setSpeed(speedSet);

motor4.setSpeed(speedSet);

delay(5);

}

}

}

void moveBackward() {

goesForward=false;

motor1.run(BACKWARD);

motor2.run(BACKWARD);Page | 29

View publication stats

motor3.run(BACKWARD);

motor4.run(BACKWARD);

for (speedSet = 0; speedSet < MAX\_SPEED; speedSet +=2) // slowly bring the speed up to avoid

loading down the batteries too quickly

{

motor1.setSpeed(speedSet);

motor2.setSpeed(speedSet);

motor3.setSpeed(speedSet);

motor4.setSpeed(speedSet);

delay(5);

}

}

void turnRight() {

motor1.run(FORWARD);

motor2.run(FORWARD);

//moveForward()

motor3.run(BACKWARD);

motor4.run(BACKWARD);

//moveBackward()

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);

}