Surface Disinfecting Bot Using UV Lights

Problem statement

We will be building an **ultraviolet sanitization robot** which will be able to kill pathogens in its environment and for that, we are going to be using an Arduino, some UV LEDs, and ultrasonic sensors.

Team Sanitizers

Divya Milind Jadkar 190260017
Akash Verma 190260005
Abhishek Kumar Soni 190260002

Components Required

1. Arduino Uno (1)
2. HC-SR04 Ultrasonic module (1)
3. Motor driver (1) - link
4. Motors (2) - link
6. Motor wheels (2) - link
7. UV LEDs (5) - link
8. Voltage Regulator (IC 7805) (1)
9. 9V DC Battery
10. Resistors (10)
11. Breadboard (1)
13. Castor wheel (1) - link

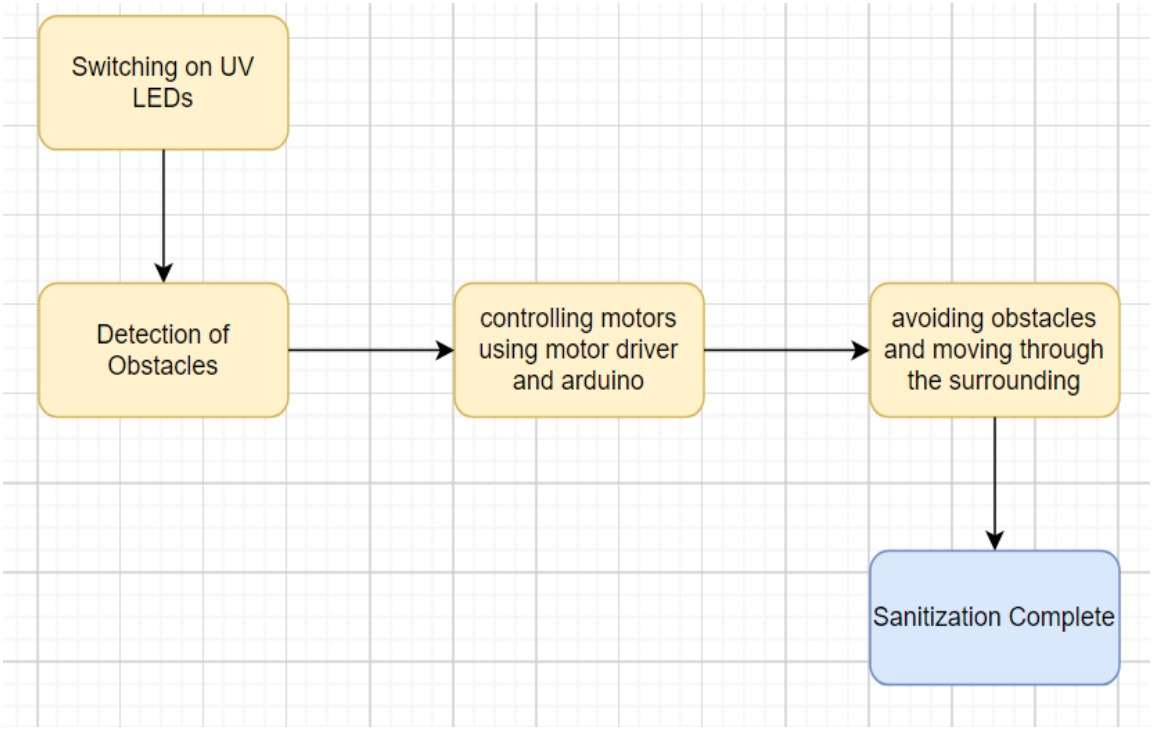
Description

We aim to build a UV sanitization robot which with the help of its onboard Ultrasonic distance sensors will be able to move through the environment avoiding the obstacles in its way and simultaneously cleansing its surroundings. We will be using 3 **HC-SR04 ultrasonic distance sensors** (One on each side except rear). The bot will be able to map its surroundings for moving through the area. If the bot detects any obstacle it just changes its path to avoid collision. This bot will have **UV LEDs** onboard**.** Bio-organisms such as bacteria and viruses are known to be ineffective when exposed to UV light. Ultraviolet light destroys the genetic material in pathogens—DNA in bacteria and RNA in viruses. While the bot is powered in, the LEDs will stay on and the sterilization process will continue. It has a total of 5 UV LEDs which gives it a 360°+ downside sterilization.

Data from the ultrasonic sensors is fed to map the environment so that the bot knows where the obstacles are. This map helps to provide the bot with a local goal. Arduino is used to calculate the

distance from obstacles; and also used to control the motors through motor drivers.

Block diagram

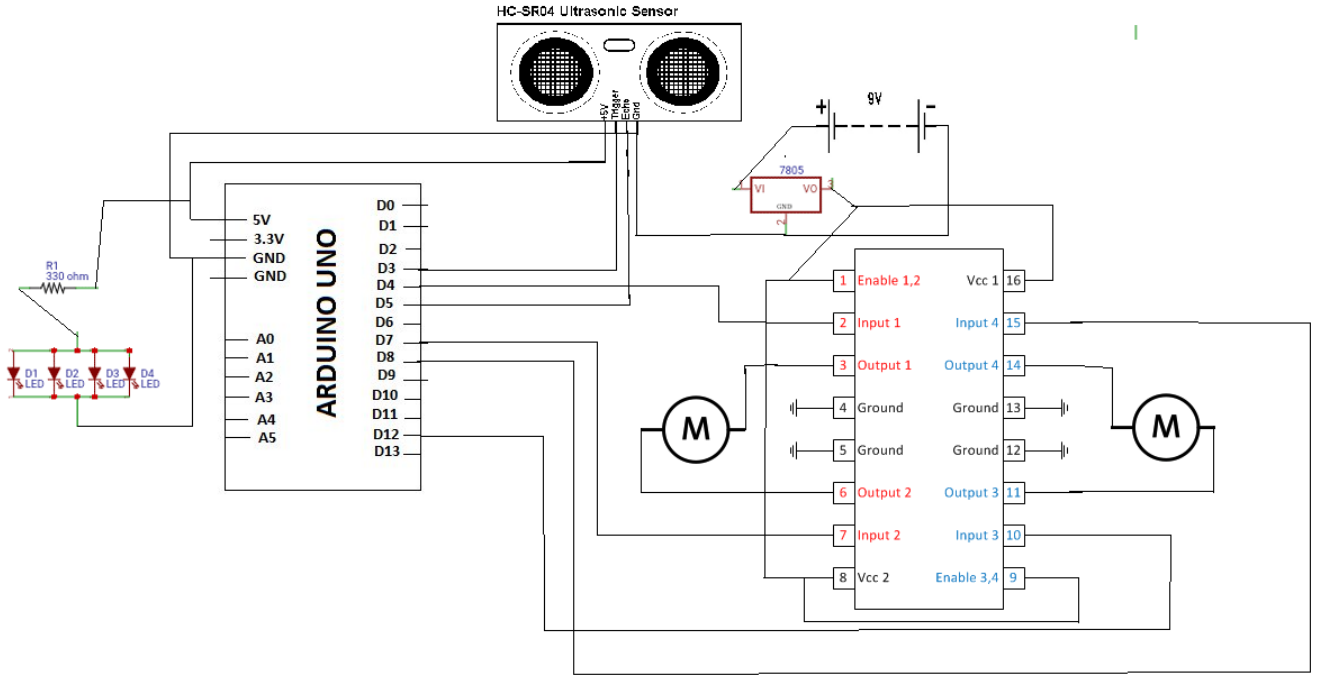


Design details

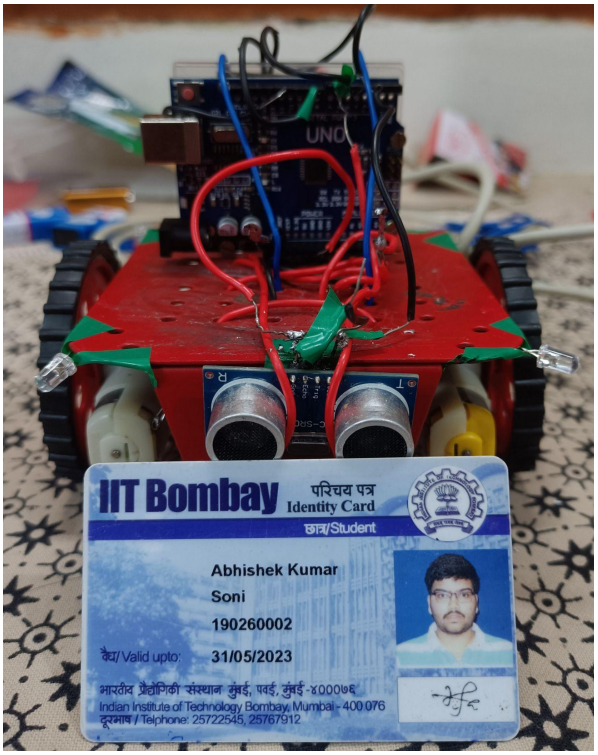
Upon powering the bot using the usb cable, UV LEDs are switched on which starts the sanitising action of the robot.
We are using a single HCSR04 Ultrasonic module. This module sends an ultrasonic pulse which travels from the bot to the nearest obstacle in its path. This module gives us the time taken by the pulse to travel from itself to the obstacle and back. This duration is fed to the microcontroller which in turn is calculating the distance between the obstacle and the bot. If this distance is greater than a threshold value(which in our case is 20cm) the arduino is instructing the motor-driver to move the wheels straight. If this distance was less than 20cm, the microcontroller tells the motor driver to turn the wheels towards the right. So if the distance obtained from the ultrasonic sensor is less than 20cm our robot turns right and hence it avoids the obstacle in its way.

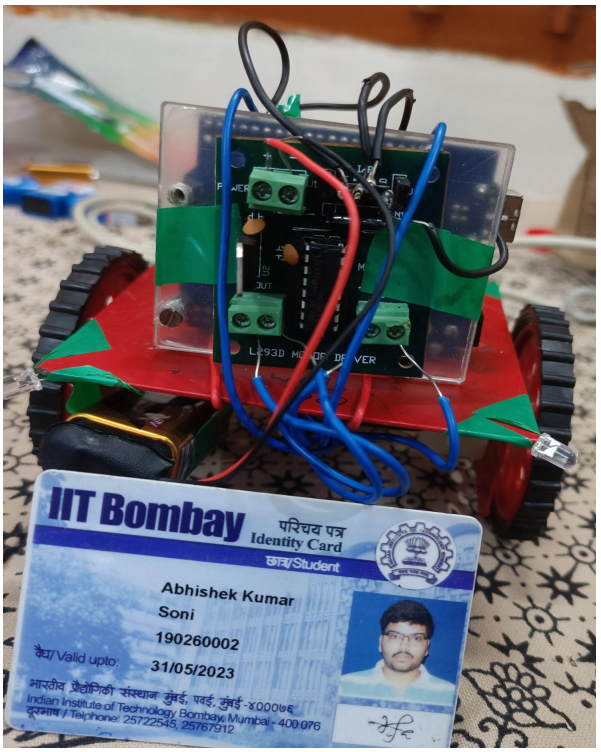
Circuit diagram and image

Circuit Diagram



Images





Status and conclusion

We aimed to build a robot which can sanitize a given surrounding and at the same time can maneuver through the environment without colliding with obstacles. We were able to complete the task using a single ultrasonic sensor. Our bot is successfully able to sanitize its environment and avoid obstacle collision.

Source codes

```
// defining the pins
const int trigPin1 = 3;
const int echoPin1 = 6;

long duration1;
int distancefront;

void setup() {
  pinMode(trigPin1, OUTPUT); // for ultrasonic sensor
  pinMode(echoPin1, INPUT);

  Serial.begin(9600);

  pinMode(4, OUTPUT); // for motor driver
  pinMode(7, OUTPUT);
  pinMode(8, OUTPUT);
  pinMode(12, OUTPUT);
}
void loop() {
  digitalWrite(trigPin1, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin1, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin1, LOW);
  duration1 = pulseIn(echoPin1, HIGH);

  distancefront = duration1 * 0.034 / 2; // 0.034 is speed of sound
  Serial.print("Distance1: ");
  Serial.println(distancefront);

  if (distancefront > 20)   // go straight
  {
    digitalWrite(4, HIGH);
    digitalWrite(7, LOW);
    digitalWrite(8, HIGH);
    digitalWrite(12, LOW);
  }
  else{                          // right turn using differential mechanism
    digitalWrite(4, HIGH);
    digitalWrite(7, LOW);
    digitalWrite(8, LOW);
    digitalWrite(12, HIGH);
  } }
```