Obstacle Avoidance Robot

IOT, ROBOTICS, EMBEDDED SYSTEMS

Internship Project-2

BACHELOR OF TECHNOLOGY

(ELECTRONICS AND COMMUNICATION ENGINEERING)

Submitted by

Mr CHIGICHERLA VINAY REDDY

SREE VIDYANIKETHAN
Engineering College (Autonomous)

TABLE OF CONTENTS

TITLE	PAGE NO.
ABSTRACT	1
INDRODUCTION	1
METHADOLOGY	2-15
CONCLUSION	16

ABSTRACT:

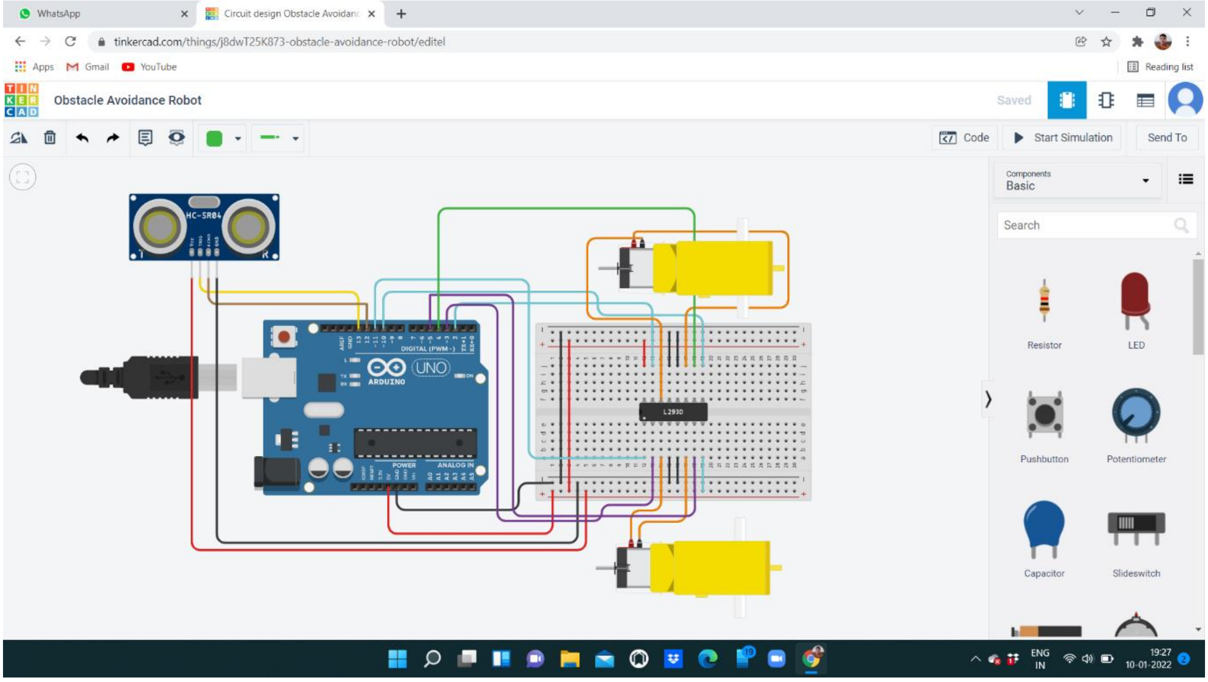
In this project we have seen how to build an obstacle avoidance robot. Obstacle detection and avoidance can be considered as the major issue in designing mobile robots. This project is useful to the robots which can sense the obstacle and can move in a particular direction without damaging itself. In this project an Obstacle Avoiding Robot is designed which can detect obstacles in its path and may avoid the obstacle from making any collision. It is a robot vehicle that works on Arduino Microcontroller and employs three ultrasonic distance sensors to detect obstacles. The Arduino board was selected as the microcontroller platform and its software counterpart, Arduino Software, was used to carry out the programming. Being a fully autonomous robot, it can be successfully used in unknown environments without any collision. The hardware used in this project is widely available and inexpensive which makes the robot easily replicable.

INDRODUCTION:

In this project we constructed a robot that avoids the obstacle which comes in its path this robot is introduced because in many of the industries we have seen that many heavy components which they have to move for one place to another place which is not possible without the help of machines. With this we got idea and we introduce the robot named as Obstacle avoidance robot using Arduino. Obstacle avoidance robot is design to allow robot to navigate in unknown environment by avoiding collisions. Obstacle avoiding robot senses obstacles in the path, avoids it and resumes its running. We have make use of sensors to achieve this objective. We have used one D.C.MOTOR i.e battery operated motor. The reason behind using BO motors is it consumes less power supply and can work properly on nine volt battery. The construction of the robot circuit is easy and small. The idea proposed in this program is by using machine vision to guide the robot. The field of machine vision has growing at a fast pace. The best part of our project is that if any obstacle is encountered by the robot the robot automatically stops or changes the direction.

METHADOLOGY:

This is the circuit diagram of the Obstacle Avoidance Robot.



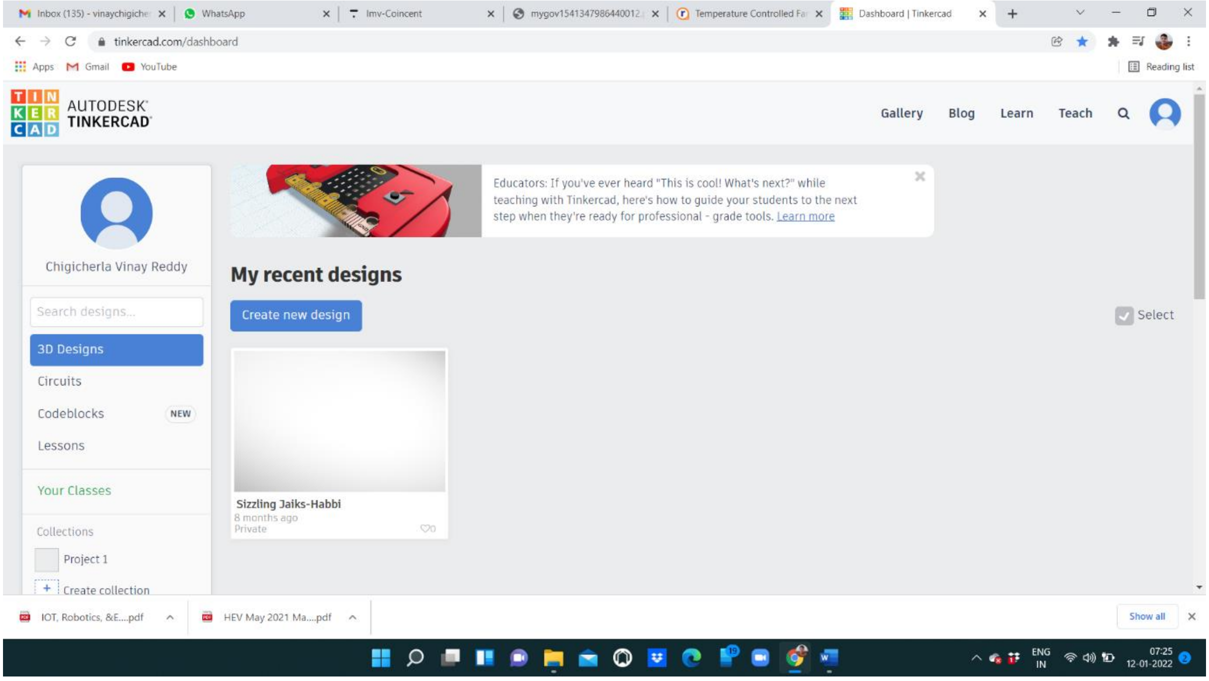
CODE:

```
const int trigPin= 13;

const int echoPin= 12;
const int in1= 3;
const int in2= 5;
const int in3= 4;
const int in4= 2;
const int en12= 11;
const int en34= 10;


void setup(){
pinMode(trigPin, OUTPUT);
pinMode(echoPin, INPUT);
pinMode(in1, OUTPUT);
pinMode(in2, OUTPUT);
pinMode(in3, OUTPUT);
pinMode(in4, OUTPUT);
  Serial.begin(9600);
}
long duration;
int distance;

void loop(){
 analogWrite(10, 100);
 analogWrite(11, 100);
```

```
  digitalWrite(trigPin, LOW);
  delay(2);
  digitalWrite(trigPin, HIGH);
  delay(10);
  digitalWrite(trigPin, LOW);
  duration= pulseIn(echoPin, HIGH);
  distance= duration*0.034/2;
  Serial.println(distance);
  if (distance<50)
  {
              digitalWrite(in1, LOW);
    digitalWrite(in2, LOW);
    digitalWrite(in3, LOW);
    digitalWrite(in4, LOW);
    delay(500);
    digitalWrite(in1, LOW);
    digitalWrite(in2, HIGH);
    digitalWrite(in3, LOW);
    digitalWrite(in4, HIGH);
    delay(500);
        digitalWrite(in1, LOW);
    digitalWrite(in2, LOW);
    digitalWrite(in3, LOW);
    digitalWrite(in4, LOW);
    delay(500);
    digitalWrite(in1, HIGH);
    digitalWrite(in2, LOW);
    digitalWrite(in3, LOW);
    digitalWrite(in4, LOW);
    delay(500);
  }
  else {
              digitalWrite(in1, HIGH);
    digitalWrite(in2, LOW);
    digitalWrite(in3, HIGH);
    digitalWrite(in4, LOW);
  }
  delay(500);
}
```

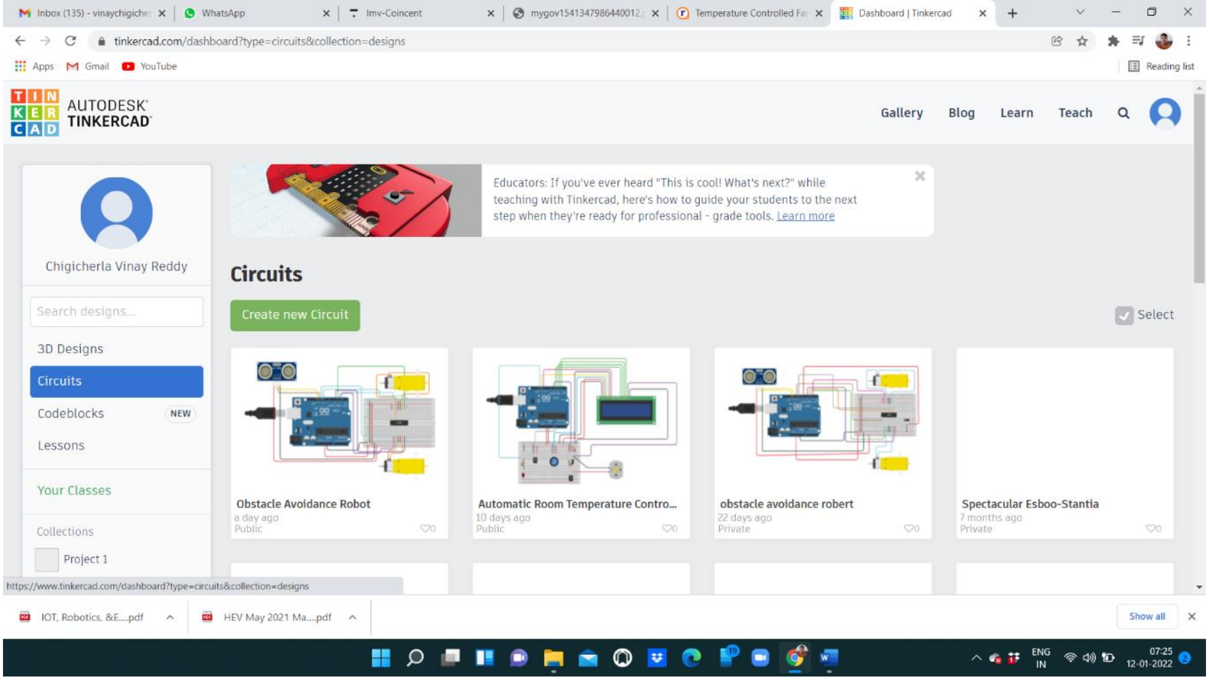
Now we discuss How we constructed the circuit and how we stimulated the circuit in a sequence.

First we have to create the account in the tinkercad using our email and we have to login into the account.



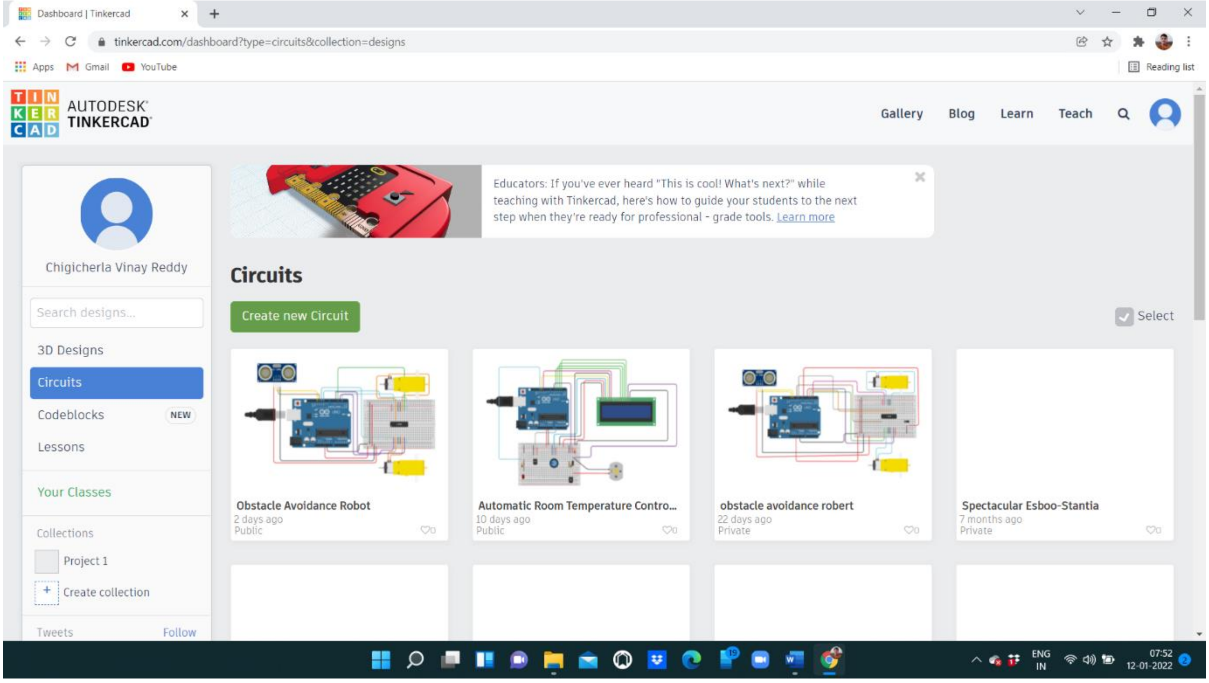
Educators: If you've ever heard "This is cool! What's next?" while teaching with Tinkercad, here's how to guide your students to the next step when they're ready for professional - grade tools. Learn more

Then we have to open the circuits in the tinkercad platform

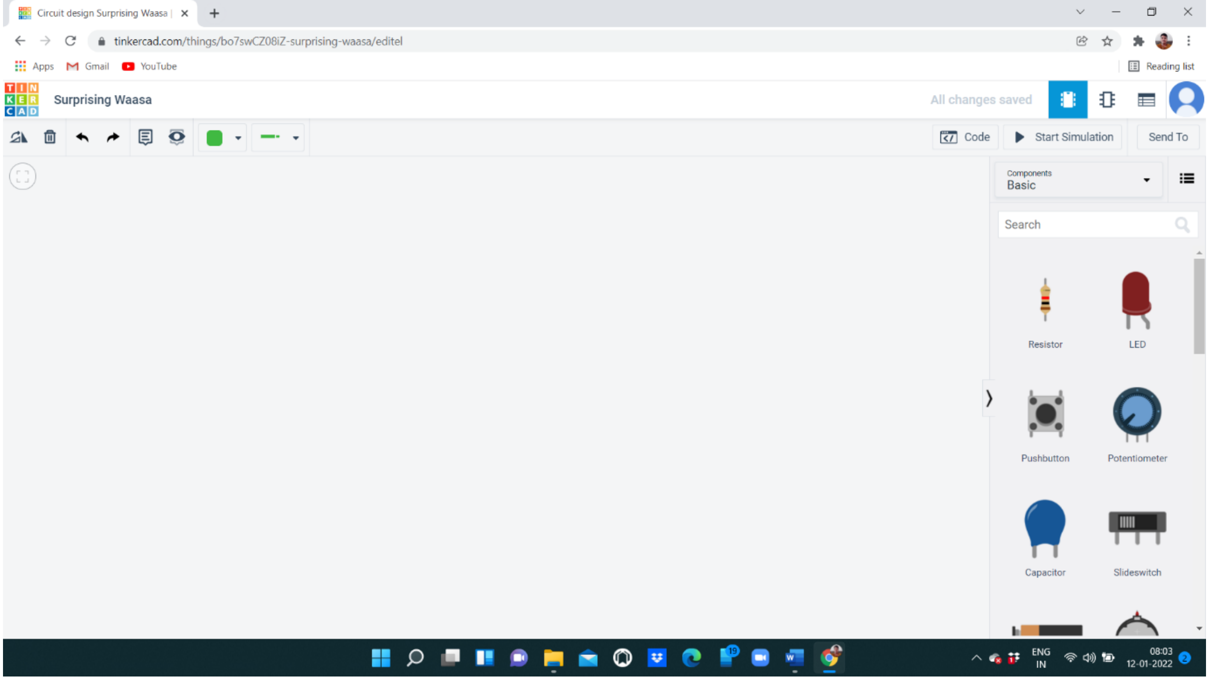


Educators: If you've ever heard "This is cool! What's next?" while teaching with Tinkercad, here's how to guide your students to the next step when they're ready for professional - grade tools. Learn more

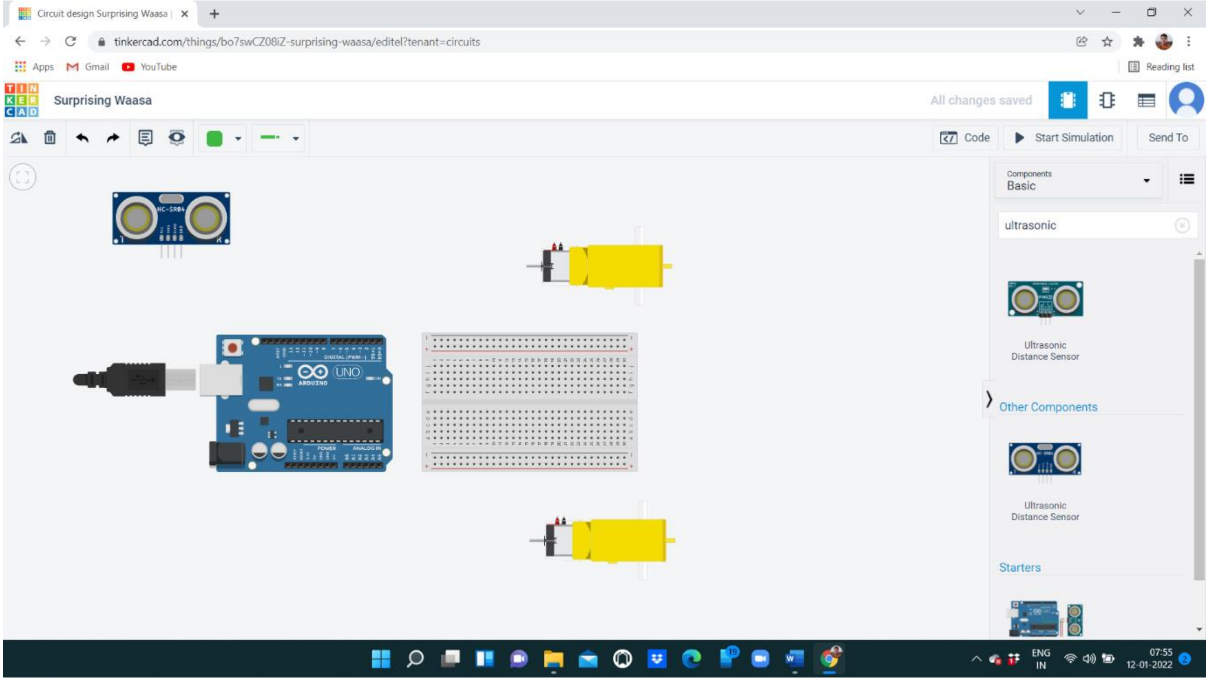
Next we have to click on create new circuit



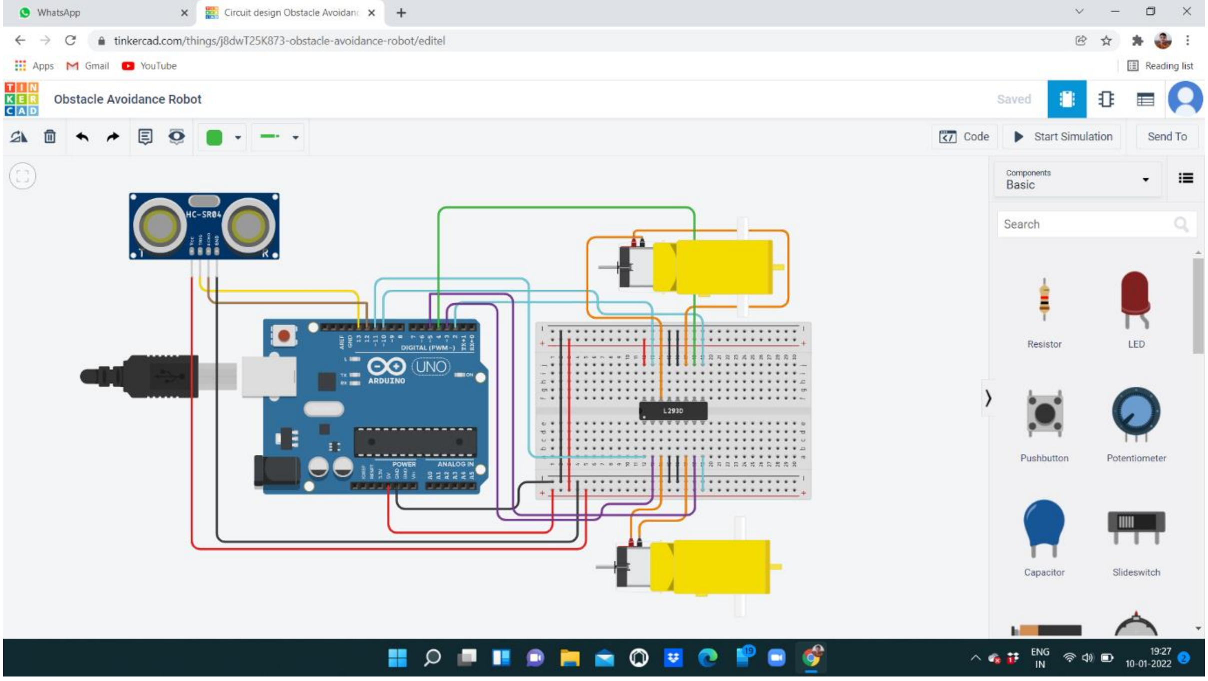
Then we have to bring all the components on to the new platform



The components are Arduino board, bread board, ultrasonic sensor, motors

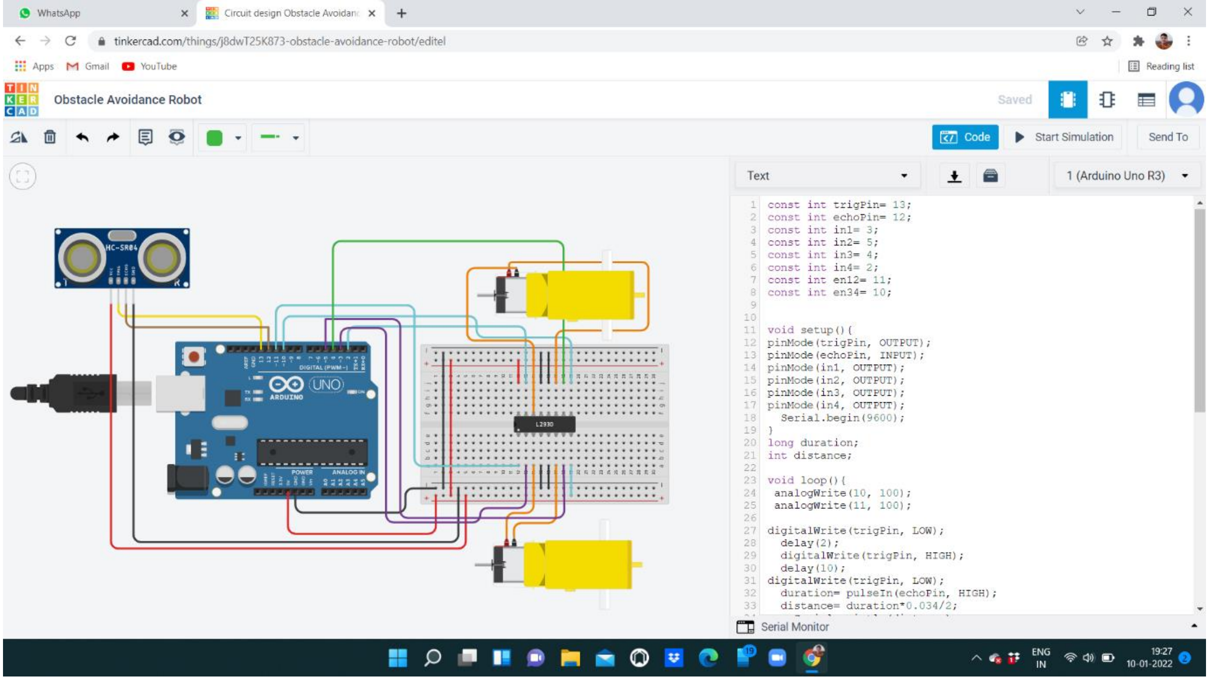


Next we connect the connections as required



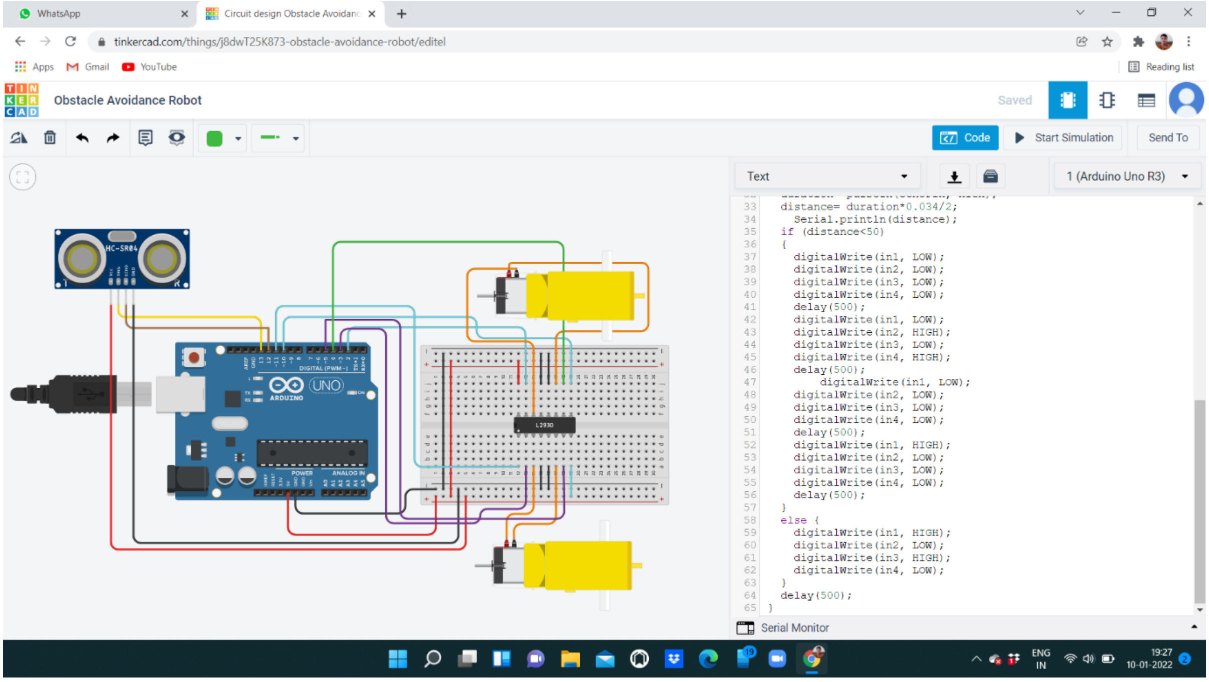
Next we write the code with required logic:

First we declare all the input and output pins in Arduino which we used.



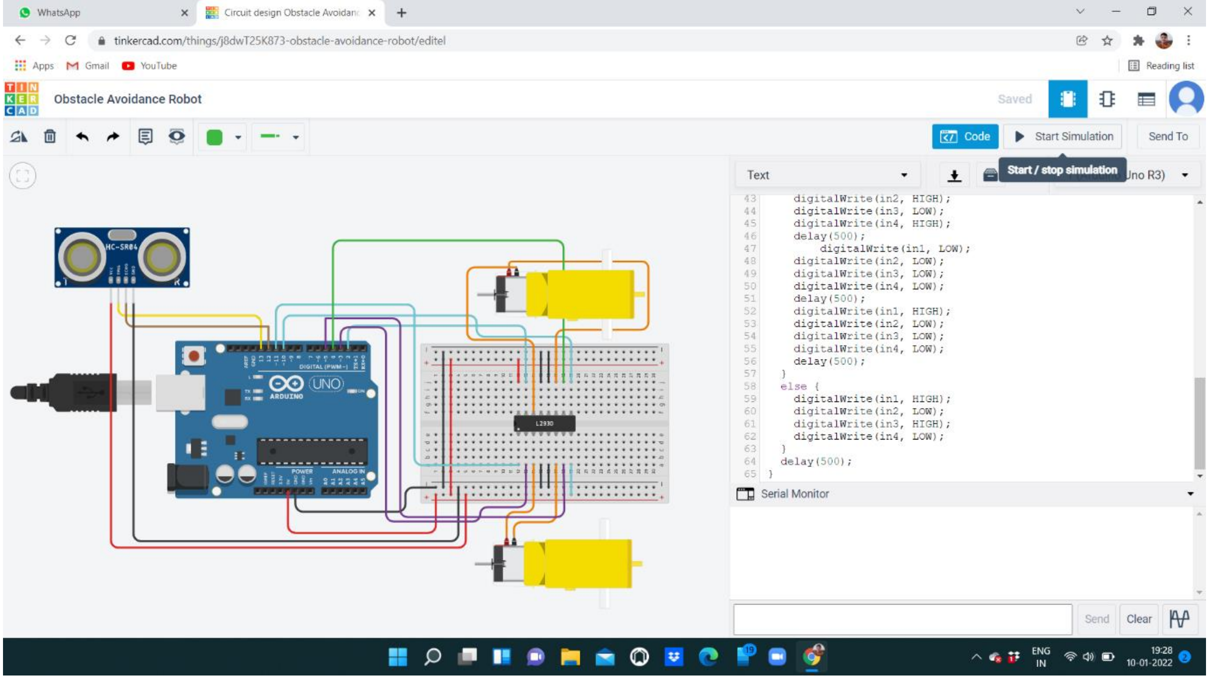
```
const int trigPin= 13;
const int echoPin= 12;
const int in1= 3;
const int in2= 5;
const int in3= 4;
const int in4= 2;
const int en12= 11;
const int en34= 10;


void setup(){
pinMode(trigPin, OUTPUT);
pinMode(echoPin, INPUT);
pinMode(in1, OUTPUT);
pinMode(in2, OUTPUT);
pinMode(in3, OUTPUT);
pinMode(in4, OUTPUT);
  Serial.begin(9600);
}
long duration;
int distance;

void loop(){
 analogWrite(10, 100);
 analogWrite(11, 100);

digitalWrite(trigPin, LOW);
  delay(2);
  digitalWrite(trigPin, HIGH);
  delay(10);
digitalWrite(trigPin, LOW);
  duration= pulseIn(echoPin, HIGH);
  distance= duration*0.034/2;
```

Then we right the code to detect the obstacle and we can find the distance of obstacle.



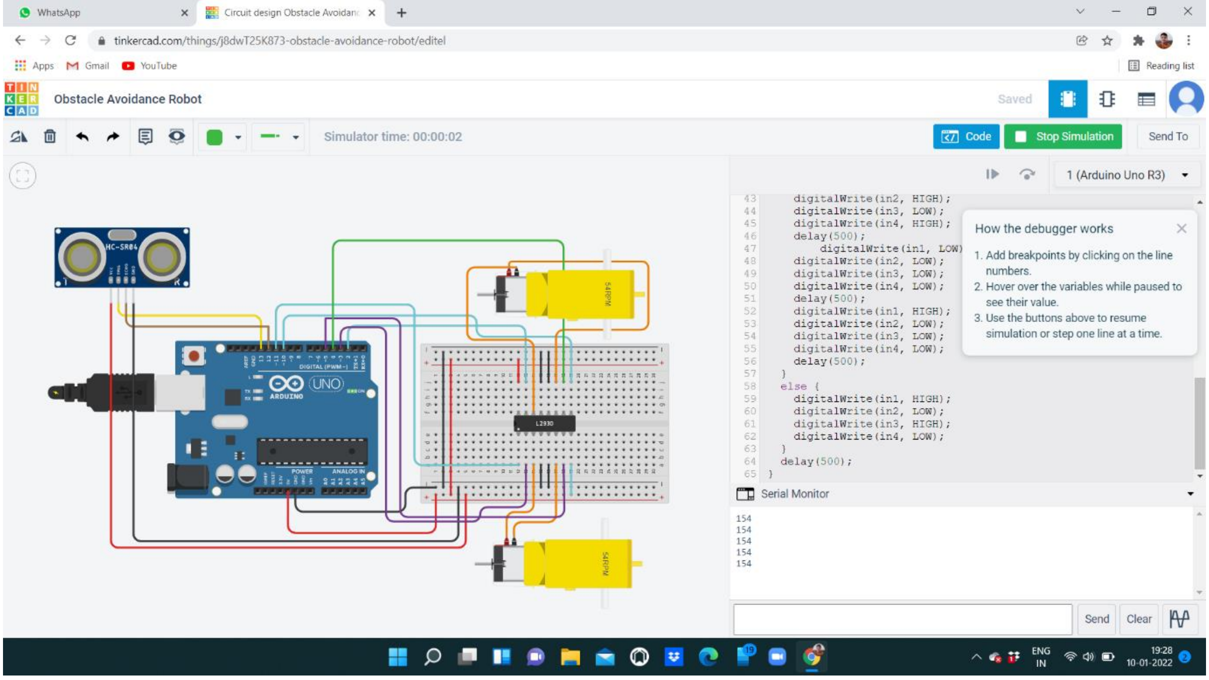
```
  distance= duration*0.034/2;
    Serial.println(distance);
  if (distance<50)
  {
    digitalWrite(in1, LOW);
    digitalWrite(in2, LOW);
    digitalWrite(in3, LOW);
    digitalWrite(in4, LOW);
    delay(500);
    digitalWrite(in1, LOW);
    digitalWrite(in2, HIGH);
    digitalWrite(in3, LOW);
    digitalWrite(in4, HIGH);
    delay(500);
        digitalWrite(in1, LOW);
    digitalWrite(in2, LOW);
    digitalWrite(in3, LOW);
    digitalWrite(in4, LOW);
    delay(500);
    digitalWrite(in1, HIGH);
    digitalWrite(in2, LOW);
    digitalWrite(in3, LOW);
    digitalWrite(in4, LOW);
    delay(500);
  }
  else {
    digitalWrite(in1, HIGH);
    digitalWrite(in2, LOW);
    digitalWrite(in3, HIGH);
    digitalWrite(in4, LOW);
  }
  delay(500);
}
```

We can see the distance of obstacle in the serial monitor



```
    digitalWrite(in2, HIGH);
    digitalWrite(in3, LOW);
    digitalWrite(in4, HIGH);
    delay(500);
        digitalWrite(in1, LOW);
    digitalWrite(in2, LOW);
    digitalWrite(in3, LOW);
    digitalWrite(in4, LOW);
    delay(500);
    digitalWrite(in1, HIGH);
    digitalWrite(in2, LOW);
    digitalWrite(in3, LOW);
    digitalWrite(in4, LOW);
    delay(500);
  }
  else {
    digitalWrite(in1, HIGH);
    digitalWrite(in2, LOW);
    digitalWrite(in3, HIGH);
    digitalWrite(in4, LOW);
  }
  delay(500);
}
```

Now we start the stimulation and can see the distance of the obstacle in the Serial monitor



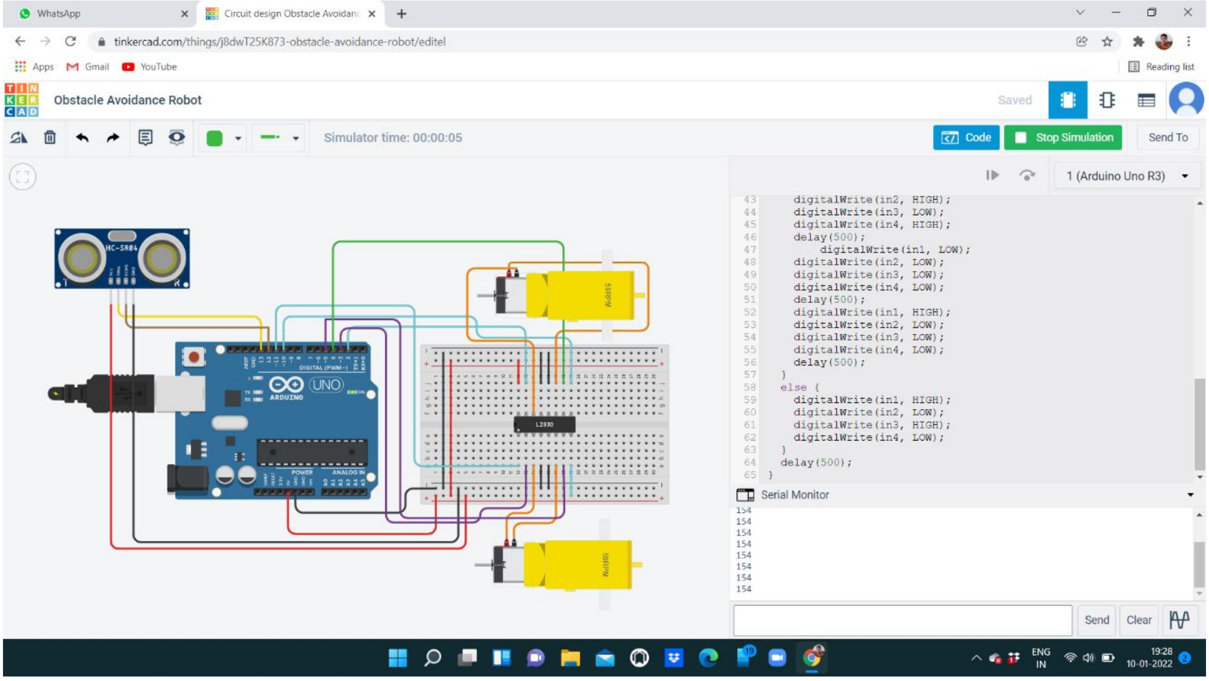
```
    digitalWrite(in2, HIGH);
    digitalWrite(in3, LOW);
    digitalWrite(in4, HIGH);
    delay(500);
        digitalWrite(in1, LOW)
    digitalWrite(in2, LOW);
    digitalWrite(in3, LOW);
    digitalWrite(in4, LOW);
    delay(500);
    digitalWrite(in1, HIGH);
    digitalWrite(in2, LOW);
    digitalWrite(in3, LOW);
    digitalWrite(in4, LOW);
    delay(500);
  }
  else {
    digitalWrite(in1, HIGH);
    digitalWrite(in2, LOW);
    digitalWrite(in3, HIGH);
    digitalWrite(in4, LOW);
  }
  delay(500);
}
```

How the debugger works

1. Add breakpoints by clicking on the line numbers.
2. Hover over the variables while paused to see their value.
3. Use the buttons above to resume simulation or step one line at a time.

```
154
154
154
154
154
```

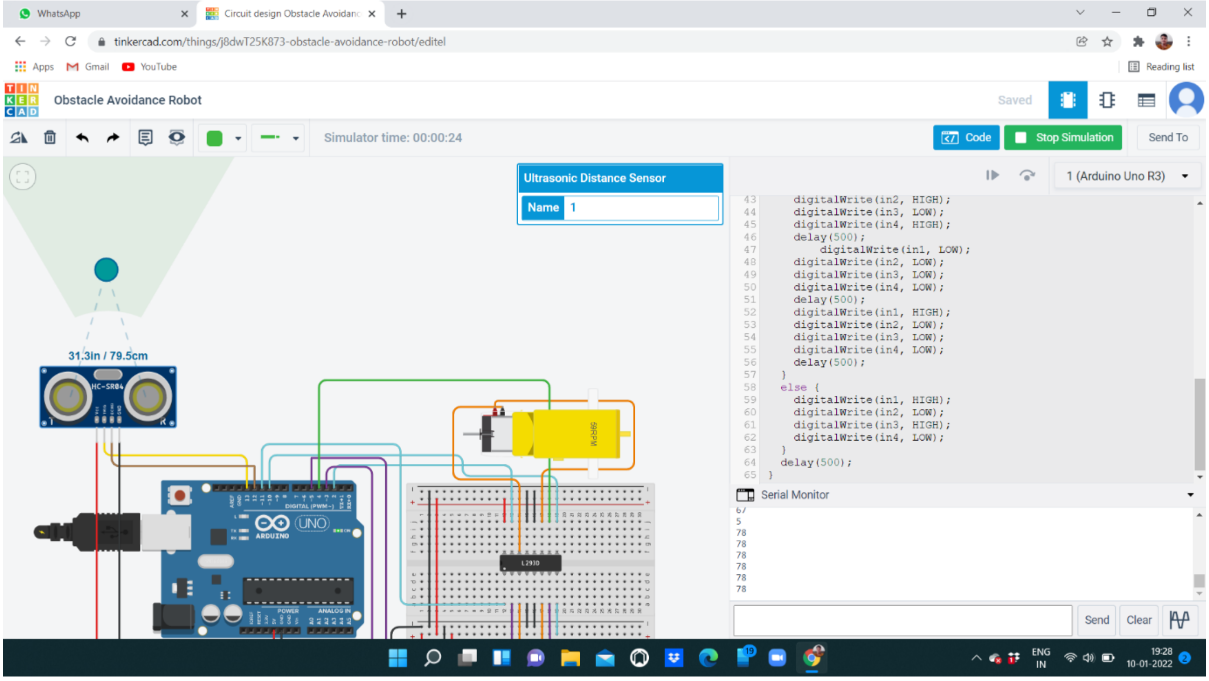
In this screenshot we can see the distance of obstacle in the serial monitor



```
      digitalWrite(in2, HIGH);
      digitalWrite(in3, LOW);
      digitalWrite(in4, HIGH);
      delay(500);
          digitalWrite(in1, LOW);
      digitalWrite(in2, LOW);
      digitalWrite(in3, LOW);
      digitalWrite(in4, LOW);
      delay(500);
      digitalWrite(in1, HIGH);
      digitalWrite(in2, LOW);
      digitalWrite(in3, LOW);
      digitalWrite(in4, LOW);
      delay(500);
    }
    else {
      digitalWrite(in1, HIGH);
      digitalWrite(in2, LOW);
      digitalWrite(in3, HIGH);
      digitalWrite(in4, LOW);
    }
    delay(500);
  }
```

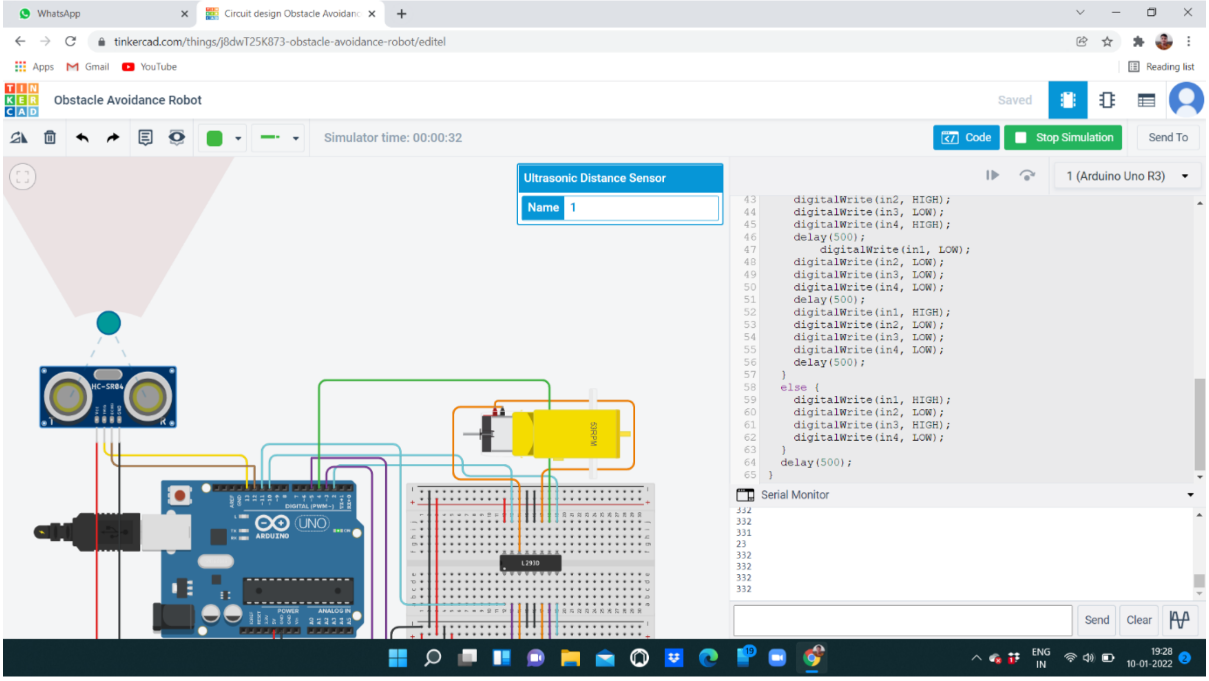
```
154
154
154
154
154
154
154
154
```

We can see the different distances in the serial monitor in below screenshots



```
      digitalWrite(in2, HIGH);
      digitalWrite(in3, LOW);
      digitalWrite(in4, HIGH);
      delay(500);
          digitalWrite(in1, LOW);
      digitalWrite(in2, LOW);
      digitalWrite(in3, LOW);
      digitalWrite(in4, LOW);
      delay(500);
      digitalWrite(in1, HIGH);
      digitalWrite(in2, LOW);
      digitalWrite(in3, LOW);
      digitalWrite(in4, LOW);
      delay(500);
    }
    else {
      digitalWrite(in1, HIGH);
      digitalWrite(in2, LOW);
      digitalWrite(in3, HIGH);
      digitalWrite(in4, LOW);
    }
    delay(500);
  }
```

```
67
5
78
78
78
78
78
78
```



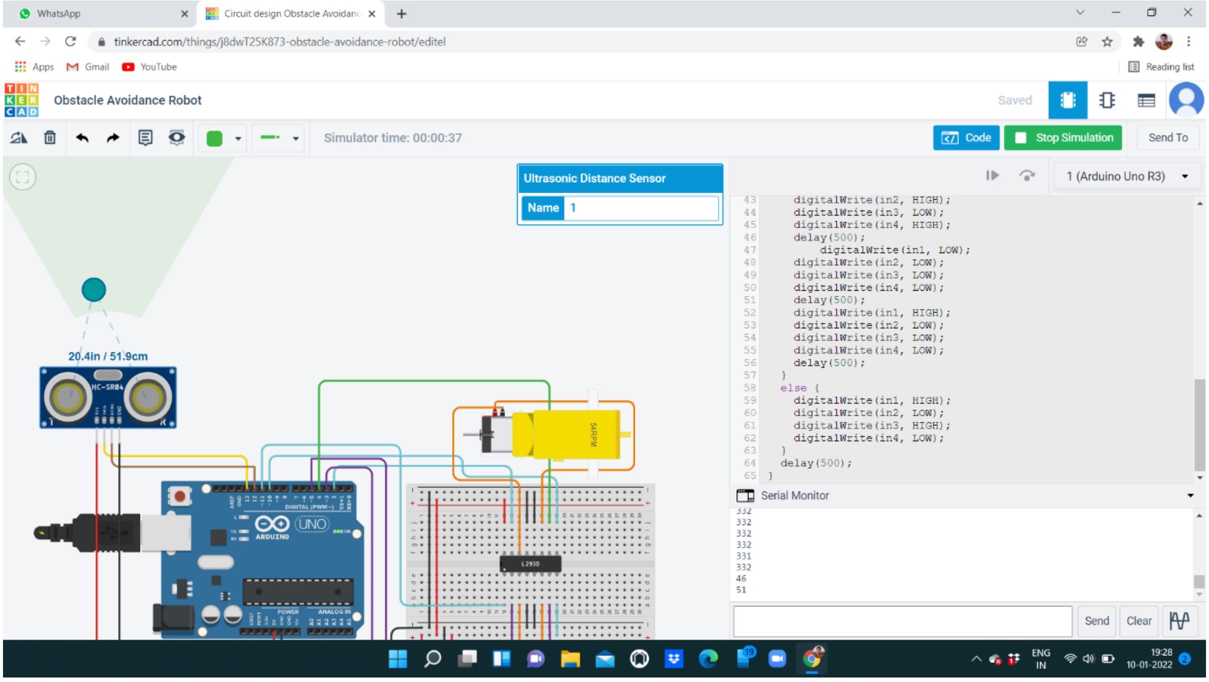
Circuit design Obstacle Avoidanc
tinkercad.com/things/j8dwT25K873-obstacle-avoidance-robot/editel
Obstacle Avoidance Robot
Saved
Code
Stop Simulation
Send To
Ultrasonic Distance Sensor
HC-SR04
ARDUINO
UNO
L293D
1 (Arduino Uno R3)
Serial Monitor
Send
Clear

Simulator time: 00:00:32

Name 1

```
    digitalWrite(in2, HIGH);
    digitalWrite(in3, LOW);
    digitalWrite(in4, HIGH);
    delay(500);
        digitalWrite(in1, LOW);
    digitalWrite(in2, LOW);
    digitalWrite(in3, LOW);
    digitalWrite(in4, LOW);
    delay(500);
    digitalWrite(in1, HIGH);
    digitalWrite(in2, LOW);
    digitalWrite(in3, LOW);
    digitalWrite(in4, LOW);
    delay(500);
  }
  else {
    digitalWrite(in1, HIGH);
    digitalWrite(in2, LOW);
    digitalWrite(in3, HIGH);
    digitalWrite(in4, LOW);
  }
  delay(500);
}
```

```
332
332
331
23
332
332
332
332
```

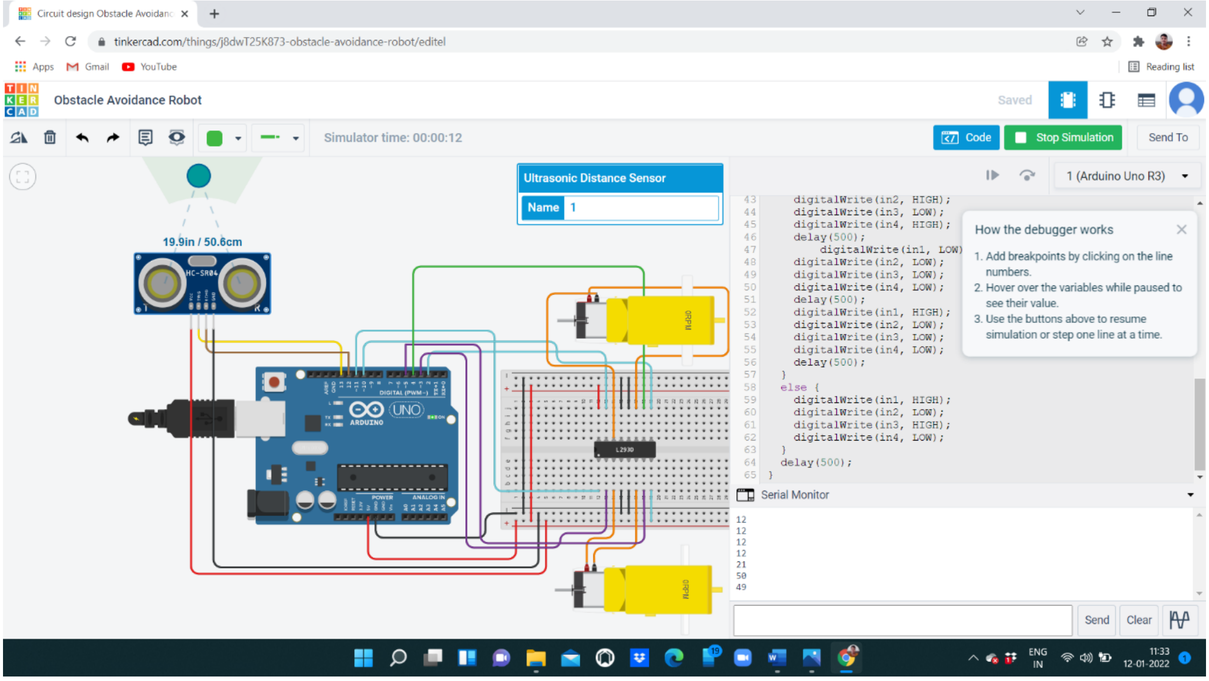


Simulator time: 00:00:37

Name 1

```
    digitalWrite(in2, HIGH);
    digitalWrite(in3, LOW);
    digitalWrite(in4, HIGH);
    delay(500);
        digitalWrite(in1, LOW);
    digitalWrite(in2, LOW);
    digitalWrite(in3, LOW);
    digitalWrite(in4, LOW);
    delay(500);
    digitalWrite(in1, HIGH);
    digitalWrite(in2, LOW);
    digitalWrite(in3, LOW);
    digitalWrite(in4, LOW);
    delay(500);
  }
  else {
    digitalWrite(in1, HIGH);
    digitalWrite(in2, LOW);
    digitalWrite(in3, HIGH);
    digitalWrite(in4, LOW);
  }
  delay(500);
}
```

```
332
332
332
332
331
332
46
51
```

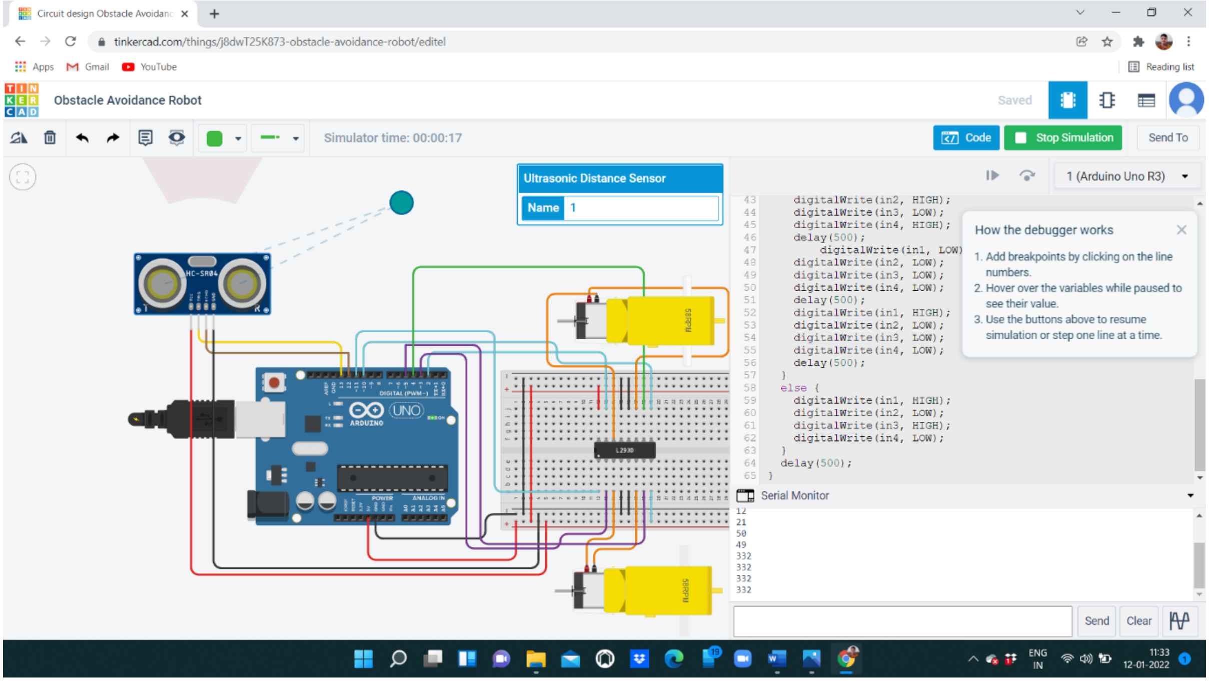

Circuit design Obstacle Avoidanc
tinkercad.com/things/j8dwT25K873-obstacle-avoidance-robot/editel
Apps
Gmail
YouTube
Reading list
Obstacle Avoidance Robot
Saved
Code
Stop Simulation
Send To
Ultrasonic Distance Sensor
Name
19.9in / 50.6cm
HC-SR04
UNO
ARDUINO
L293D
1 (Arduino Uno R3)
How the debugger works
1. Add breakpoints by clicking on the line numbers.
2. Hover over the variables while paused to see their value.
3. Use the buttons above to resume simulation or step one line at a time.
Serial Monitor
Send
Clear
ENG
IN
11:33
12-01-2022

Simulator time: 00:00:12

Name: 1

```
    digitalWrite(in2, HIGH);
    digitalWrite(in3, LOW);
    digitalWrite(in4, HIGH);
    delay(500);
        digitalWrite(in1, LOW)
    digitalWrite(in2, LOW);
    digitalWrite(in3, LOW);
    digitalWrite(in4, LOW);
    delay(500);
    digitalWrite(in1, HIGH);
    digitalWrite(in2, LOW);
    digitalWrite(in3, LOW);
    digitalWrite(in4, LOW);
    delay(500);
  }
  else {
    digitalWrite(in1, HIGH);
    digitalWrite(in2, LOW);
    digitalWrite(in3, HIGH);
    digitalWrite(in4, LOW);
  }
  delay(500);
}
```

```
12
12
12
12
21
50
49
```

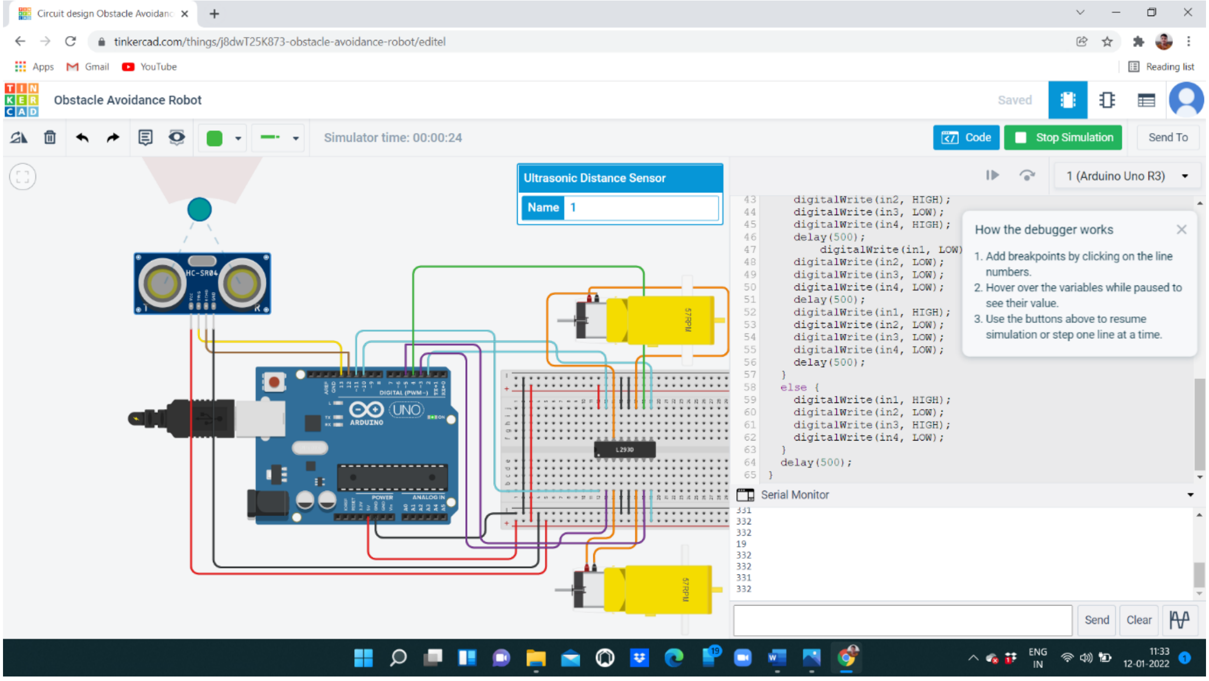

Circuit design Obstacle Avoidanc
tinkercad.com/things/j8dwT25K873-obstacle-avoidance-robot/editel
Apps
Gmail
YouTube
Reading list
Obstacle Avoidance Robot
Saved
Code
Stop Simulation
Send To
Ultrasonic Distance Sensor
Name
HC-SR04
UNO
ARDUINO
L293D
1 (Arduino Uno R3)
How the debugger works
1. Add breakpoints by clicking on the line numbers.
2. Hover over the variables while paused to see their value.
3. Use the buttons above to resume simulation or step one line at a time.
Serial Monitor
Send
Clear
ENG
IN
11:33
12-01-2022

Simulator time: 00:00:17

Name: 1

```
    digitalWrite(in2, HIGH);
    digitalWrite(in3, LOW);
    digitalWrite(in4, HIGH);
    delay(500);
        digitalWrite(in1, LOW)
    digitalWrite(in2, LOW);
    digitalWrite(in3, LOW);
    digitalWrite(in4, LOW);
    delay(500);
    digitalWrite(in1, HIGH);
    digitalWrite(in2, LOW);
    digitalWrite(in3, LOW);
    digitalWrite(in4, LOW);
    delay(500);
  }
  else {
    digitalWrite(in1, HIGH);
    digitalWrite(in2, LOW);
    digitalWrite(in3, HIGH);
    digitalWrite(in4, LOW);
  }
  delay(500);
}
```

```
12
21
50
49
332
332
332
332
```

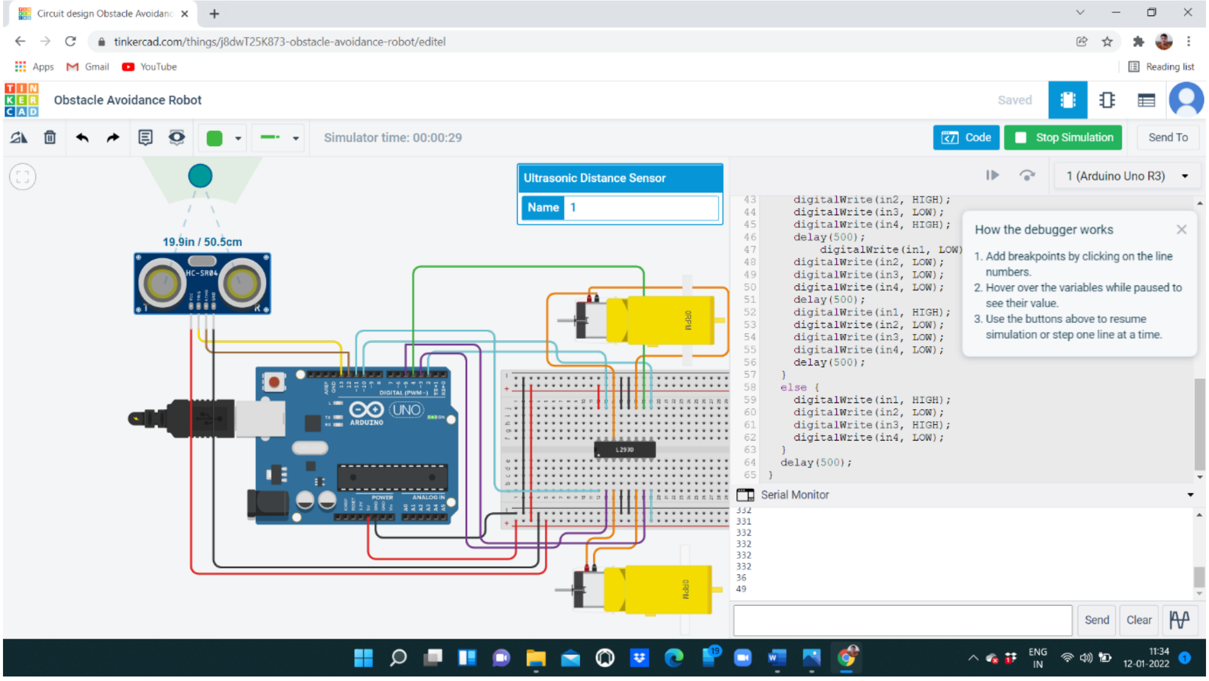


```
    digitalWrite(in2, HIGH);
    digitalWrite(in3, LOW);
    digitalWrite(in4, HIGH);
    delay(500);
        digitalWrite(in1, LOW)
    digitalWrite(in2, LOW);
    digitalWrite(in3, LOW);
    digitalWrite(in4, LOW);
    delay(500);
    digitalWrite(in1, HIGH);
    digitalWrite(in2, LOW);
    digitalWrite(in3, LOW);
    digitalWrite(in4, LOW);
    delay(500);
  }
  else {
    digitalWrite(in1, HIGH);
    digitalWrite(in2, LOW);
    digitalWrite(in3, HIGH);
    digitalWrite(in4, LOW);
  }
  delay(500);
}
```

How the debugger works

1. Add breakpoints by clicking on the line numbers.
2. Hover over the variables while paused to see their value.
3. Use the buttons above to resume simulation or step one line at a time.

```
331
332
332
19
332
332
331
332
```

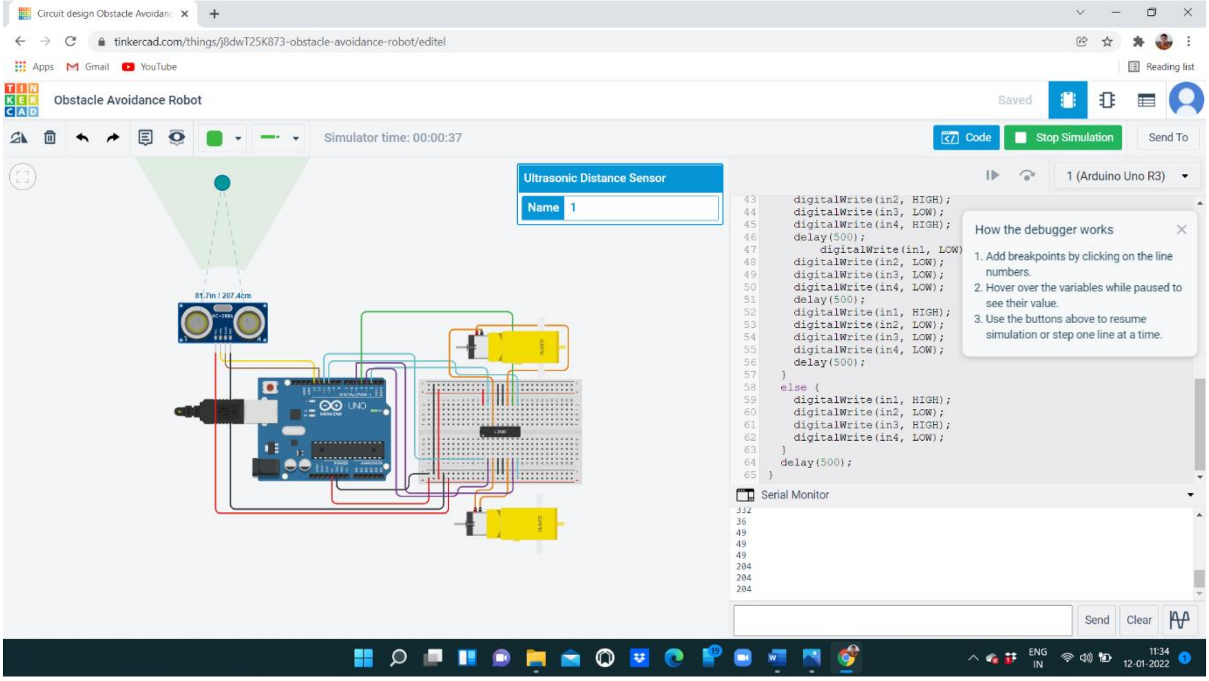


```
    digitalWrite(in2, HIGH);
    digitalWrite(in3, LOW);
    digitalWrite(in4, HIGH);
    delay(500);
        digitalWrite(in1, LOW)
    digitalWrite(in2, LOW);
    digitalWrite(in3, LOW);
    digitalWrite(in4, LOW);
    delay(500);
    digitalWrite(in1, HIGH);
    digitalWrite(in2, LOW);
    digitalWrite(in3, LOW);
    digitalWrite(in4, LOW);
    delay(500);
  }
  else {
    digitalWrite(in1, HIGH);
    digitalWrite(in2, LOW);
    digitalWrite(in3, HIGH);
    digitalWrite(in4, LOW);
  }
  delay(500);
}
```

How the debugger works

1. Add breakpoints by clicking on the line numbers.
2. Hover over the variables while paused to see their value.
3. Use the buttons above to resume simulation or step one line at a time.

```
332
331
332
332
332
332
36
49
```



Simulator time: 00:00:37

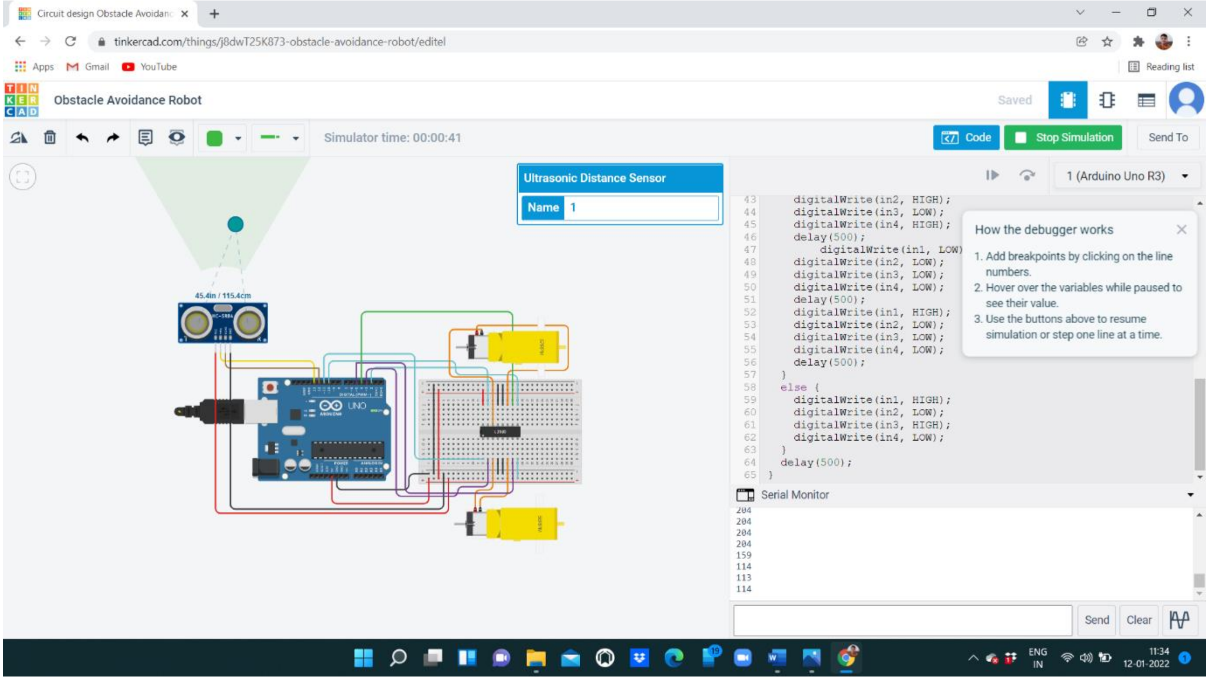
Name 1

```
    digitalWrite(in2, HIGH);
    digitalWrite(in3, LOW);
    digitalWrite(in4, HIGH);
    delay(500);
        digitalWrite(in1, LOW)
    digitalWrite(in2, LOW);
    digitalWrite(in3, LOW);
    digitalWrite(in4, LOW);
    delay(500);
    digitalWrite(in1, HIGH);
    digitalWrite(in2, LOW);
    digitalWrite(in3, LOW);
    digitalWrite(in4, LOW);
    delay(500);
  }
  else {
    digitalWrite(in1, HIGH);
    digitalWrite(in2, LOW);
    digitalWrite(in3, HIGH);
    digitalWrite(in4, LOW);
  }
  delay(500);
}
```

How the debugger works

1. Add breakpoints by clicking on the line numbers.
2. Hover over the variables while paused to see their value.
3. Use the buttons above to resume simulation or step one line at a time.

```
332
36
49
49
49
204
204
204
```



Simulator time: 00:00:41

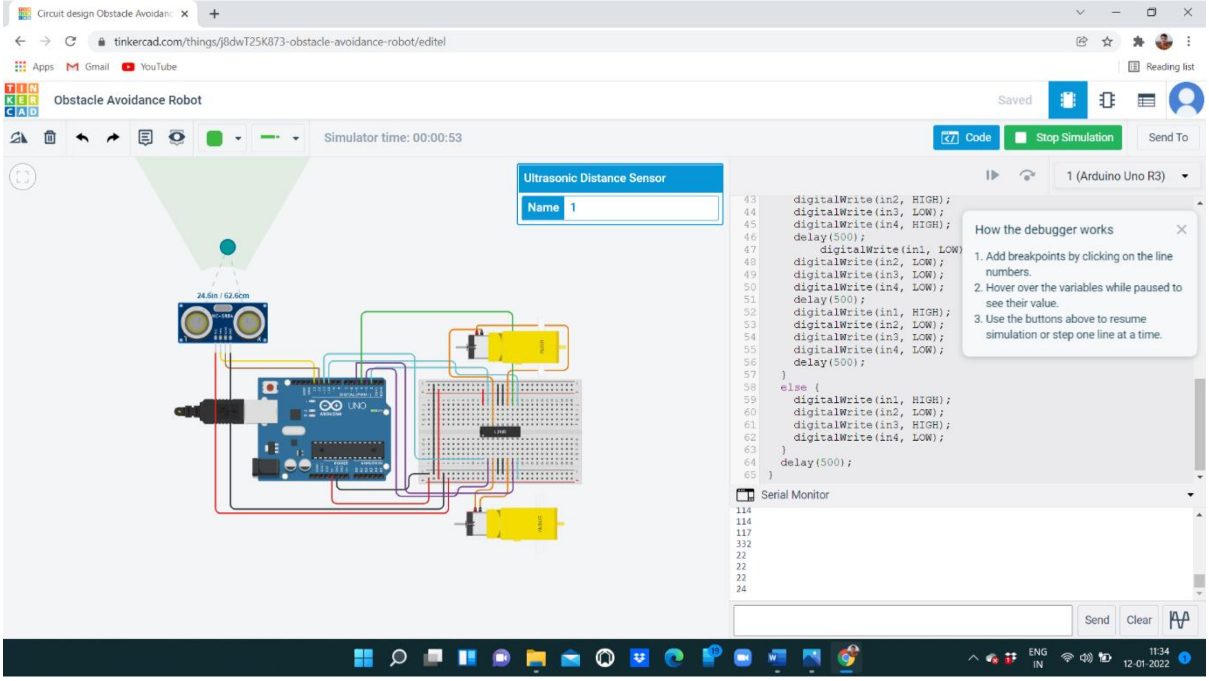
Name 1

```
    digitalWrite(in2, HIGH);
    digitalWrite(in3, LOW);
    digitalWrite(in4, HIGH);
    delay(500);
        digitalWrite(in1, LOW)
    digitalWrite(in2, LOW);
    digitalWrite(in3, LOW);
    digitalWrite(in4, LOW);
    delay(500);
    digitalWrite(in1, HIGH);
    digitalWrite(in2, LOW);
    digitalWrite(in3, LOW);
    digitalWrite(in4, LOW);
    delay(500);
  }
  else {
    digitalWrite(in1, HIGH);
    digitalWrite(in2, LOW);
    digitalWrite(in3, HIGH);
    digitalWrite(in4, LOW);
  }
  delay(500);
}
```

How the debugger works

1. Add breakpoints by clicking on the line numbers.
2. Hover over the variables while paused to see their value.
3. Use the buttons above to resume simulation or step one line at a time.

```
204
204
204
204
159
114
113
114
```



Obstacle Avoidance Robot

Simulator time: 00:00:53

Ultrasonic Distance Sensor

Name: 1

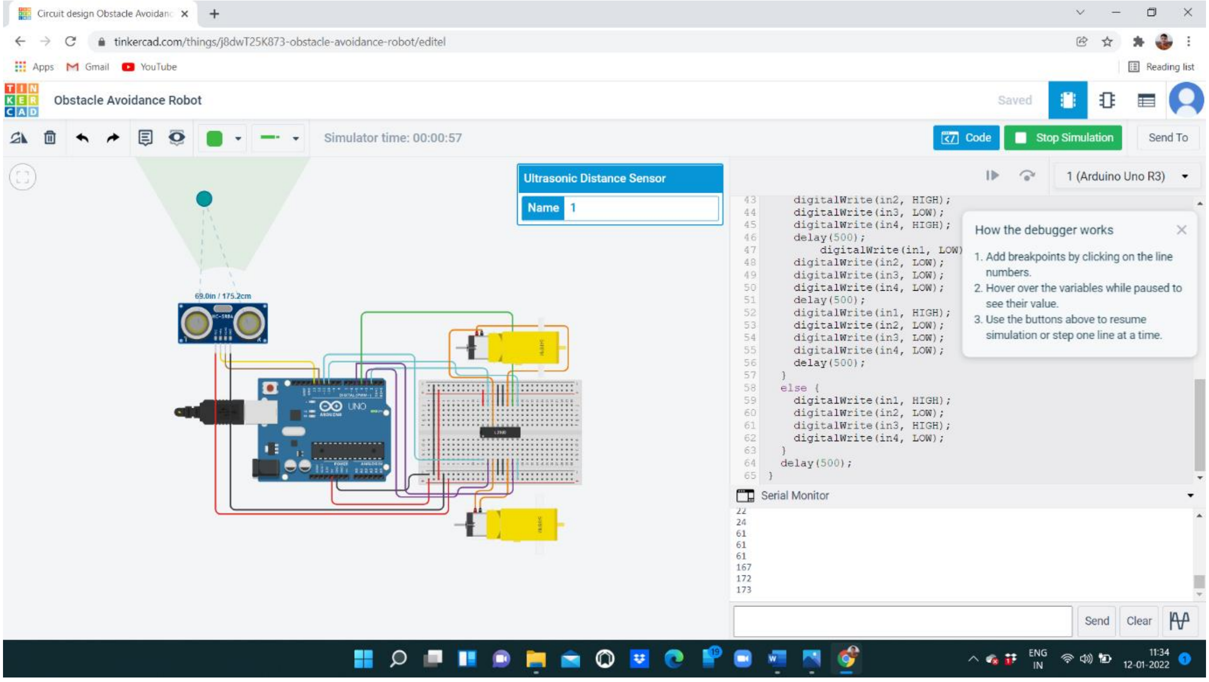
```
    digitalWrite(in2, HIGH);
    digitalWrite(in3, LOW);
    digitalWrite(in4, HIGH);
    delay(500);
        digitalWrite(in1, LOW)
    digitalWrite(in2, LOW);
    digitalWrite(in3, LOW);
    digitalWrite(in4, LOW);
    delay(500);
    digitalWrite(in1, HIGH);
    digitalWrite(in2, LOW);
    digitalWrite(in3, LOW);
    digitalWrite(in4, LOW);
    delay(500);
  }
  else {
    digitalWrite(in1, HIGH);
    digitalWrite(in2, LOW);
    digitalWrite(in3, HIGH);
    digitalWrite(in4, LOW);
  }
  delay(500);
}
```

How the debugger works

1. Add breakpoints by clicking on the line numbers.
2. Hover over the variables while paused to see their value.
3. Use the buttons above to resume simulation or step one line at a time.

Serial Monitor

```
114
114
117
332
22
22
22
24
```



Obstacle Avoidance Robot

Simulator time: 00:00:57

Ultrasonic Distance Sensor

Name: 1

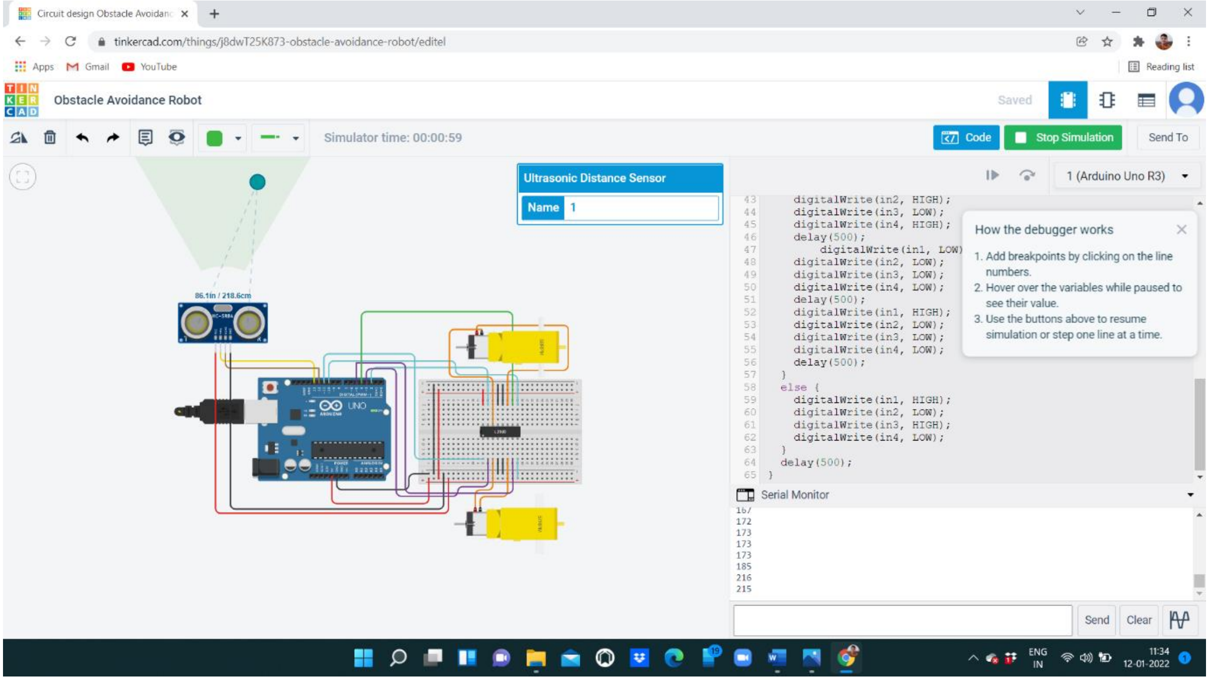
```
    digitalWrite(in2, HIGH);
    digitalWrite(in3, LOW);
    digitalWrite(in4, HIGH);
    delay(500);
        digitalWrite(in1, LOW)
    digitalWrite(in2, LOW);
    digitalWrite(in3, LOW);
    digitalWrite(in4, LOW);
    delay(500);
    digitalWrite(in1, HIGH);
    digitalWrite(in2, LOW);
    digitalWrite(in3, LOW);
    digitalWrite(in4, LOW);
    delay(500);
  }
  else {
    digitalWrite(in1, HIGH);
    digitalWrite(in2, LOW);
    digitalWrite(in3, HIGH);
    digitalWrite(in4, LOW);
  }
  delay(500);
}
```

How the debugger works

1. Add breakpoints by clicking on the line numbers.
2. Hover over the variables while paused to see their value.
3. Use the buttons above to resume simulation or step one line at a time.

Serial Monitor

```
22
24
61
61
61
167
172
173
```


Circuit design Obstacle Avoidanc
tinkercad.com/things/j8dwT25K873-obstacle-avoidance-robot/editel
Apps
Gmail
YouTube
Reading list
Saved
Code
Stop Simulation
Send To
Send
Clear
ENG
IN
11:34
12-01-2022

Obstacle Avoidance Robot

Simulator time: 00:00:59

Ultrasonic Distance Sensor

Name: 1

86.1in / 218.6cm

1 (Arduino Uno R3)

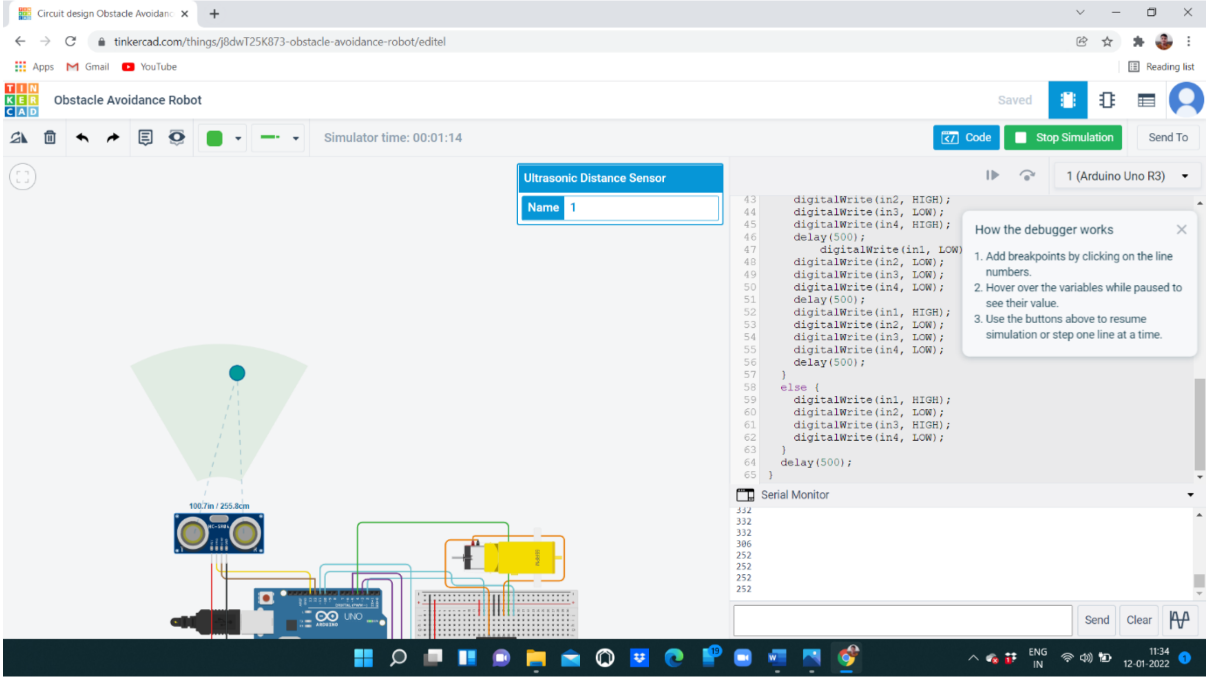
```
    digitalWrite(in2, HIGH);
    digitalWrite(in3, LOW);
    digitalWrite(in4, HIGH);
    delay(500);
        digitalWrite(in1, LOW)
    digitalWrite(in2, LOW);
    digitalWrite(in3, LOW);
    digitalWrite(in4, LOW);
    delay(500);
    digitalWrite(in1, HIGH);
    digitalWrite(in2, LOW);
    digitalWrite(in3, LOW);
    digitalWrite(in4, LOW);
    delay(500);
  }
  else {
    digitalWrite(in1, HIGH);
    digitalWrite(in2, LOW);
    digitalWrite(in3, HIGH);
    digitalWrite(in4, LOW);
  }
  delay(500);
}
```

How the debugger works

1. Add breakpoints by clicking on the line numbers.
2. Hover over the variables while paused to see their value.
3. Use the buttons above to resume simulation or step one line at a time.

Serial Monitor

```
167
172
173
173
173
185
216
215
```


Circuit design Obstacle Avoidanc
tinkercad.com/things/j8dwT25K873-obstacle-avoidance-robot/editel
Apps
Gmail
YouTube
Reading list
Saved
Code
Stop Simulation
Send To
Send
Clear
ENG
IN
11:34
12-01-2022

Obstacle Avoidance Robot

Simulator time: 00:01:14

Ultrasonic Distance Sensor

Name: 1

100.7in / 255.8cm

1 (Arduino Uno R3)

```
    digitalWrite(in2, HIGH);
    digitalWrite(in3, LOW);
    digitalWrite(in4, HIGH);
    delay(500);
        digitalWrite(in1, LOW)
    digitalWrite(in2, LOW);
    digitalWrite(in3, LOW);
    digitalWrite(in4, LOW);
    delay(500);
    digitalWrite(in1, HIGH);
    digitalWrite(in2, LOW);
    digitalWrite(in3, LOW);
    digitalWrite(in4, LOW);
    delay(500);
  }
  else {
    digitalWrite(in1, HIGH);
    digitalWrite(in2, LOW);
    digitalWrite(in3, HIGH);
    digitalWrite(in4, LOW);
  }
  delay(500);
}
```

How the debugger works

1. Add breakpoints by clicking on the line numbers.
2. Hover over the variables while paused to see their value.
3. Use the buttons above to resume simulation or step one line at a time.

Serial Monitor

```
332
332
332
306
252
252
252
252
```

CONCLUSION:

This is very useful to all. This project is very useful to all the people because now a days robot has become a part of every bodies life. By this project we can prevent the Roberts from collision with obstacle.

The link of the project is given below:

https://www.tinkercad.com/things/j8dwT25K873-obstacle-avoidance-robot/editel?sharecode=L0OVEiIHJfdqvJGNHKz7VaXtNlM4u18PMG53gRS1YYY