VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

ECE1005-SENSORS AND INSTRUMENTATION

J-COMPONENT PROJECT FINAL REPORT

Prof. Arunkumar Chandrasekhar

OBSTACLE AVOIDING ROBOTIC CAR

GROUP MEMBERS:

GOWRI NAMBOODIRI -20BEC0779

VINYAS A SHETTY-20BEC0780

AJINKYA BAGMAR -20BEC0781

RASHTRA YADAV -20BEC0787

TABLE OF CONTENTS:

INTRODUCTION:

In today's world robotics is an interesting, fast-growing and advanced field. It is the simplest way for the latest technology modification. Nowadays automation is an integral part of the most recent advanced technologies, so we decided to work in the field of robotics, and design something which will make human life simpler on a day to day basis. An obstacle avoiding robot is an intelligent device, which can automatically sense and overcome obstacles on its path. Obstacle Avoidance is a robotic discipline to move vehicles based on sensorial information. The use of these methods in front of classic methods (path planning) is a natural alternative when the scenario is dynamic with unpredictable behaviour. In these cases, the surroundings do not remain invariable, and thus the sensory information is used to detect the changes consequently adapting to moving. It will automatically scan the surrounding for further paths. This project is the basic stage of any automatic robot. This robot has sufficient intelligence to cover the maximum area of provided space. It has an ultrasonic sensor that is used to sense the obstacles coming in between the path of the robot. It will move in a particular direction and avoid the obstacle which is coming in its path. We have used two D.C motors to give motion to the robot. The construction of the robot circuit is easy and small. The electronics parts used in the robot circuits are easily available and cheap too. So, in this project, we have simulated an obstacle avoiding robotic vehicle which can automatically detect an obstacle, avoid it and move in a different path.

Working of an Obstacle avoiding robotic car:

The obstacle avoiding robotic vehicle uses ultrasonic sensors for its movements. A microcontroller is used to achieve the desired operation. The motors are connected through the motor driver IC to the microcontroller. The ultrasonic sensor is attached in front of the robot. Whenever the robot is going on the desired path the ultrasonic sensor transmits the ultrasonic waves continuously from its sensor head. Whenever an obstacle comes ahead of it the ultrasonic waves are reflected from an object and that information is passed to the microcontroller. The microcontroller controls the motors to move the car left, right, back or front based on the ultrasonic signals.

Components used in our project:

1. Arduino Uno R3
2. H-Bridge Motor Driver(L293D)
3. DC Motor x 2
4. Ultrasonic Distance Sensor x 3-HCSR04
5. Micro servo
6. Battery
7. Bredboard

Ultrasonic Sensor:

The ultrasonic sensor is an electronic component used for obstacle detection. The ultrasonic sensor transmits the ultrasonic waves from its sensor head and again receives the ultrasonic waves reflected from an object.

When an electrical pulse of high voltage is applied to the ultrasonic transducer it vibrates across a specific spectrum of frequencies and generates a burst of sound waves. These propagate in the air at the velocity of sound. If they hit any object, they reflect an echo signal to the sensors. The detector detects these signals and sends an electrical signal back to the microcontroller which calculates the time taken between sending sound waves and receiving the echo. The echo patterns will be compared with the patterns of sound waves to determine the detected signal's condition.

To measure the distance the sound has travelled we use the formula;

$$\text{Distance} = (\text{Time} \times 0.034) / 2$$

Where ‘0.034’ is the speed of sound in centimetres per microsecond and ‘2’ is in the formula because the sound has to travel back and forth.

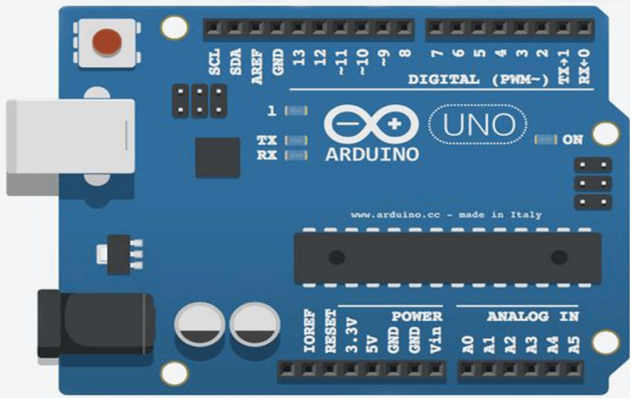
The ultrasonic sensor enables the robot to virtually see and recognize an object, avoid obstacles, measures distance. The operating range of our ultrasonic sensor is from 3 cm to 335 cm.

The ultrasonic sensor is very compact and has a very high performance. There are many applications for ultrasonic sensors like instruction alarm systems, automatic door openers, etc.

Arduino Uno:

Arduino is an open-source platform used for building electronics projects. Arduino consists of both a physical programmable circuit board (often referred to as a microcontroller) and a piece of software, or IDE (Integrated Development Environment) that runs on a computer, used to write and upload computer code to the physical board. Arduino Uno is a microcontroller board based on the ATmega328P. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analogue inputs, a 16 MHz quartz crystal, a USB connection, a power jack, an ICSP header and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with

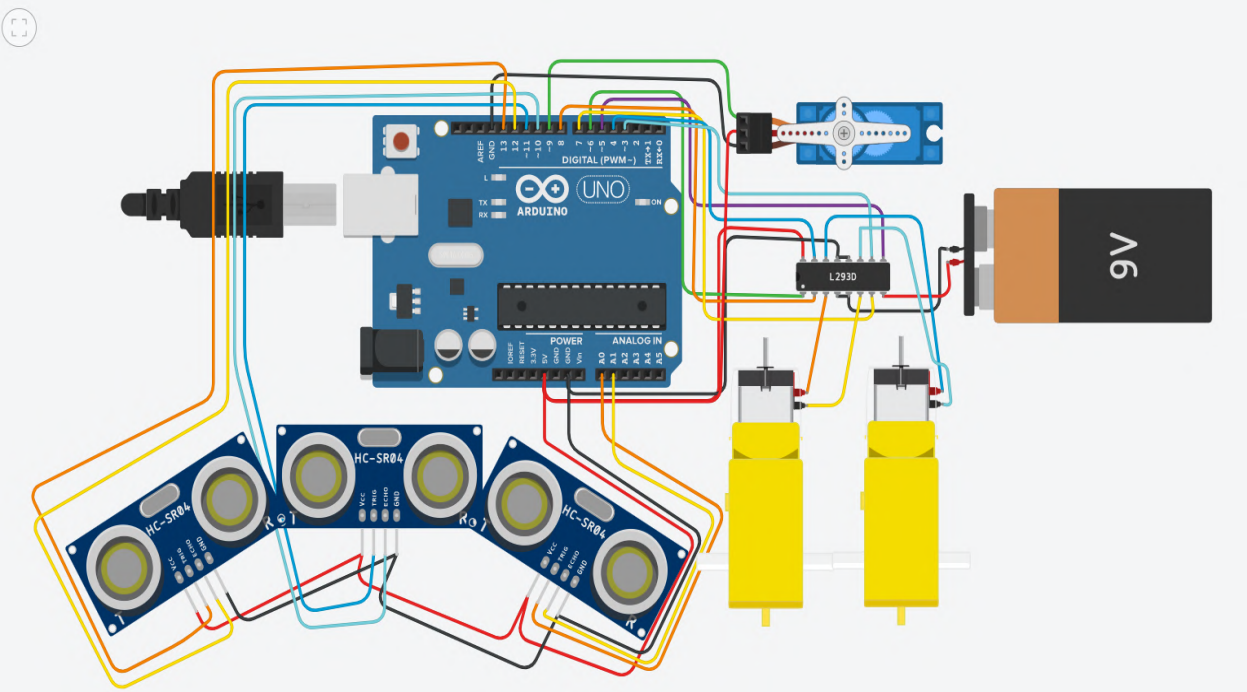
an AC-to-DC adapter or battery to get started. "Uno" means one in Italian and was chosen to mark the release of Arduino Software (IDE) 1.0. The Uno board and version 1.0 of Arduino Software (IDE) were the reference versions of Arduino, now evolved to newer releases. The Uno board is the first in a series of USB Arduino boards and the reference model for the Arduino platform.



Motor Drivers (L293D IC):

Motor drivers take a low current control signal but provide a higher current signal, thus acting as a current amplifier. The higher current signal drives the motors. L293D is a motor driver that allows a direct current (DC) motor to drive in either direction. It contains two inbuilt H-bridge driver circuits. To rotate the motor in the clockwise or anticlockwise direction, the voltage needs to change its direction. H-bridge circuit allows voltage to be flown in either direction. Hence H-bridge IC is ideal for driving a DC motor. There are 4 input pins for L293d, pin 2,7 on the left and pin 15,10 on the right. Left input pins will regulate the rotation of the motor connected across the left side and right input for the motor on the right-hand side. The motors are rotated based on the inputs provided across the input pins as LOGIC 0 or LOGIC 1. For rotating the motor in the clockwise direction, the input pins have to be provided with Logic 1 and Logic 0. Enable pins 1 and 9 (corresponding to the two motors) must be high for motors to start operating. When an enable input is high, the associated driver gets enabled. As a result, the outputs become active and work in phase with their inputs. Similarly, when the enable input is low, that driver is disabled, and their outputs are off and in the high-impedance state

Circuit Diagram:



CODE:

```
// C++ code
#include <Servo.h>
int frontdist = 0;
int leftdist = 0;
int rightdist = 0;
Servo servo_9;
long readUltrasonicDistance(int triggerPin, int echoPin)
{
  pinMode(triggerPin, OUTPUT);  // Clear the trigger
  digitalWrite(triggerPin, LOW);
  delayMicroseconds(2);
  // Sets the trigger pin to HIGH state for 10 microseconds
```

```
  digitalWrite(triggerPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(triggerPin, LOW);
  pinMode(echoPin, INPUT);
  // Reads the echo pin, and returns the sound wave travel time in microseconds
  return pulseIn(echoPin, HIGH);
}
void setup()
{
  servo_9.attach(9, 500, 2500);
  Serial.begin(9600);
  pinMode(6, OUTPUT);
  pinMode(5, OUTPUT);
  pinMode(8, OUTPUT);
  pinMode(4, OUTPUT);
  pinMode(7, OUTPUT);
  pinMode(3, OUTPUT);
}
void loop()
{
  servo_9.write(90);
  delay(1000); // Wait for 1000 millisecond(s)
  frontdist = 0.01723 * readUltrasonicDistance(11, 10);
  if (frontdist > 20) {
    // forward
    Serial.println("forward");
    servo_9.write(90);
    delay(1000); // Wait for 1000 millisecond(s)
```

```
  analogWrite(6, 255);
  analogWrite(5, 255);
  digitalWrite(8, HIGH);
  digitalWrite(4, HIGH);
  digitalWrite(7, LOW);
  digitalWrite(3, LOW);
 } else {
  // 1stop
  Serial.println("1stop");
  analogWrite(6, 0);
  analogWrite(5, 0);
  digitalWrite(8, LOW);
  digitalWrite(4, LOW);
  digitalWrite(7, LOW);
  digitalWrite(3, LOW);
  delay(2000); // Wait for 2000 millisecond(s)
  // 1reverse
  Serial.println("1reverse");
  analogWrite(6, 127);
  analogWrite(5, 127);
  digitalWrite(8, LOW);
  digitalWrite(4, LOW);
  digitalWrite(7, HIGH);
  digitalWrite(3, HIGH);
  delay(2000); // Wait for 2000 millisecond(s)
  leftdist = 0.01723 * readUltrasonicDistance(13, 12);
  rightdist = 0.01723 * readUltrasonicDistance(A0, A1);
  if (leftdist < 20 && rightdist > 20) {
```

```
  // right
  Serial.println("right");
  servo_9.write(0);
  delay(1000); // Wait for 1000 millisecond(s)
  analogWrite(6, 127);
  analogWrite(5, 127);
  digitalWrite(8, HIGH);
  digitalWrite(4, HIGH);
  digitalWrite(7, LOW);
  digitalWrite(3, LOW);
  delay(2000); // Wait for 2000 millisecond(s)
 } else {
  if (leftdist > 20 && rightdist < 20) {
   // left
   Serial.println("left");
   servo_9.write(180);
   delay(1000); // Wait for 1000 millisecond(s)
   analogWrite(6, 127);
   analogWrite(5, 127);
   digitalWrite(8, HIGH);
   digitalWrite(4, HIGH);
   digitalWrite(7, LOW);
   digitalWrite(3, LOW);
   delay(2000); // Wait for 2000 millisecond(s)
  } else {
   // 2stop
   Serial.println("2stop");
   analogWrite(6, 0);
```

```
      analogWrite(5, 0);
      digitalWrite(8, LOW);
      digitalWrite(4, LOW);
      digitalWrite(7, LOW);
      digitalWrite(3, LOW);
      delay(2000); // Wait for 2000 millisecond(s)
      // 2reverse
      Serial.println("2reverse");
      analogWrite(6, 127);
      analogWrite(5, 127);
      digitalWrite(8, LOW);
      digitalWrite(4, LOW);
      digitalWrite(7, HIGH);
      digitalWrite(3, HIGH);
      delay(2000); // Wait for 2000 millisecond(s)
      // uturn
      Serial.println("uturn");
      analogWrite(6, 200);
      analogWrite(5, 150);
      digitalWrite(8, HIGH);
      digitalWrite(4, LOW);
      digitalWrite(7, LOW);
      digitalWrite(3, HIGH);
      delay(3000); // Wait for 3000 millisecond(s)
    }
   }
  }
}
```

Our Robot Car's Working:

Our obstacle avoidance robot uses 4 ultrasonic sensors to measure the distance between itself and the obstacle, based on the input provided by these sensors, the microcontroller(Arduino) is used to achieve the desired operation. The motors are connected to the motor drivers which are connected to the Arduino. In our simulation, the object must be at least 50cm away from the sensors to not consider it as an obstacle.

The Arduino prioritizes the front, right, left and back respectively. This means that if there is no obstacle in the front, it continues moving forward. If it detects any obstacle in the front, it checks for obstacles in the right and moves towards the right by switching off the right wheel. When the sensors detect the front and right being blocked, it checks for obstacles in the left and moves to the left by switching off the left wheel. If there are obstacles detected on the front, right and left, it checks for obstacles in the back and reverses itself. If there are obstacles on all sides the car stops moving.

Experimental Results:

1. When there was no obstacle in the front side – the car moved forward
2. When there was an obstacle in front and left side – the car turned right
3. When there was an obstacle in the front and right side – the car turned left
4. When there was an obstacle in front, left and right side – the car moved backwards and took an U-TURN

Applications:

1. Smart vacuum cleaner
2. Automatic lawn mower
3. Automatic Parking assistant
4. Industries
5. Unmanned military operations
6. Surveying different landscapes and mapping them.

Conclusion:

We have just simulated how a simple obstacle avoiding robotic vehicle(car) works for a few common real-life cases set by us, but a fully-fledged automatic vehicle uses various types of sensors and is highly advanced with many machine learning capabilities that consider lots of factors to change its path when an obstacle is detected.

References:

https://www.instructables.com/obstacle-avoiding-robot-with-servo-motor-arduino/

https://create.arduino.cc/projecthub/sora-jy/obstacles-avoiding-robot-with-servo-motor-719f6b

The website used for simulation:

- https://www.tinkercad.com

Simulation Link

https://www.tinkercad.com/things/8NcIebuij9J-ece1005-obstacle-avoiding-robot/editel

PPT→

OBSTACLE AVOIDING ROBOTIC CAR

ECE 1005 TD1 PROJECT FINAL REVIEW

MEMBERS:

- AJINKYA BAGMAR -20BEC0781
- VINYAS A SHETTY-20BEC0780
- GOWRI NAMBOODIRI -20BEC0779
- RASHTRA YADAV -20BEC0787

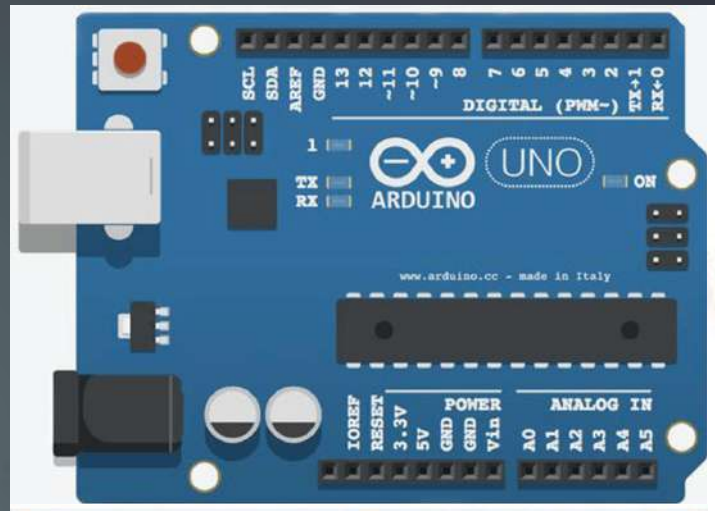
ABSTRACT

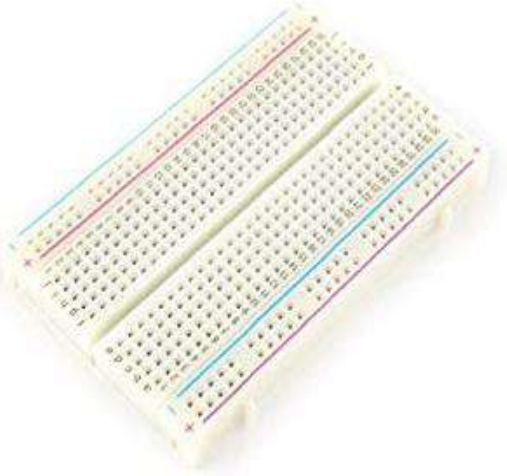
Obstacle avoidance is one of the most important aspects of mobile robotics. this project proposes a robotic vehicle that can move away when an obstacle is detected in its path.an ultrasonic sensor is used to detect any obstacle ahead of it and depending upon the input signal the micro-controller redirects the robot to move in an alternate direction by modulating the motor.

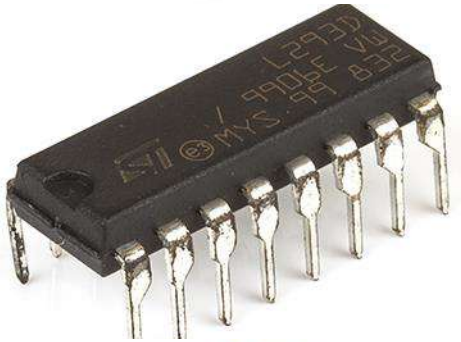
COMPONENTS USED IN OUR PROJECT

- ARDUINO UNO R3
- H-BRIDGE MOTOR DRIVER(L293D)
- DC MOTOR X 2
- ULTRASONIC DISTANCE SENSOR X 3
- MICRO SERVO
- BATTERY

Quantity	Component
1	Arduino Uno R3
1	H-bridge Motor Driver
2	Hobby Gearmotor
1	Positional Micro Servo
1	9V Battery
3	Ultrasonic Distance Sensor

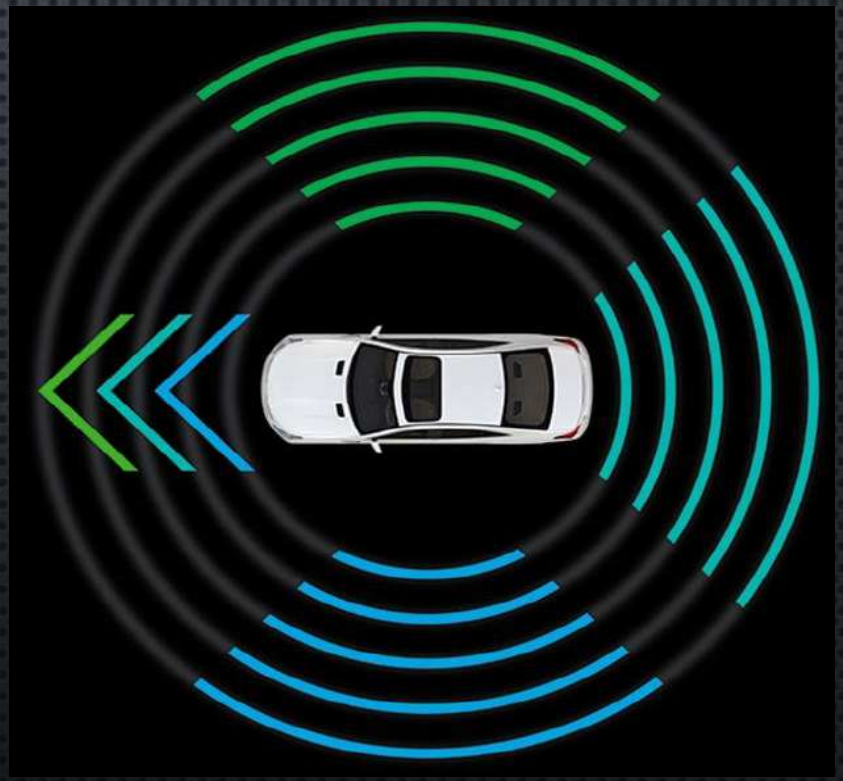






WORKING PRINCIPLE

- THE OBSTACLE AVOIDANCE ROBOTIC VEHICLE USES ULTRASONIC SENSORS FOR ITS MOVEMENTS. A MICROCONTROLLER IS USED TO ACHIEVE THE DESIRED OPERATION. THE MOTORS ARE CONNECTED THROUGH THE MOTOR DRIVER IC TO THE MICROCONTROLLER. THE ULTRASONIC SENSOR IS ATTACHED IN FRONT OF THE ROBOT.
- WHENEVER THE ROBOT IS GOING ON THE DESIRED PATH THE ULTRASONIC SENSOR TRANSMITS THE ULTRASONIC WAVES CONTINUOUSLY FROM ITS SENSOR HEAD. WHENEVER AN OBSTACLE COMES AHEAD OF IT THE ULTRASONIC WAVES ARE REFLECTED FROM AN OBJECT AND THAT INFORMATION IS PASSED TO THE MICROCONTROLLER. THE MICROCONTROLLER CONTROLS THE MOTORS LEFT, RIGHT, BACK, FRONT, BASED ON ULTRASONIC SIGNALS.



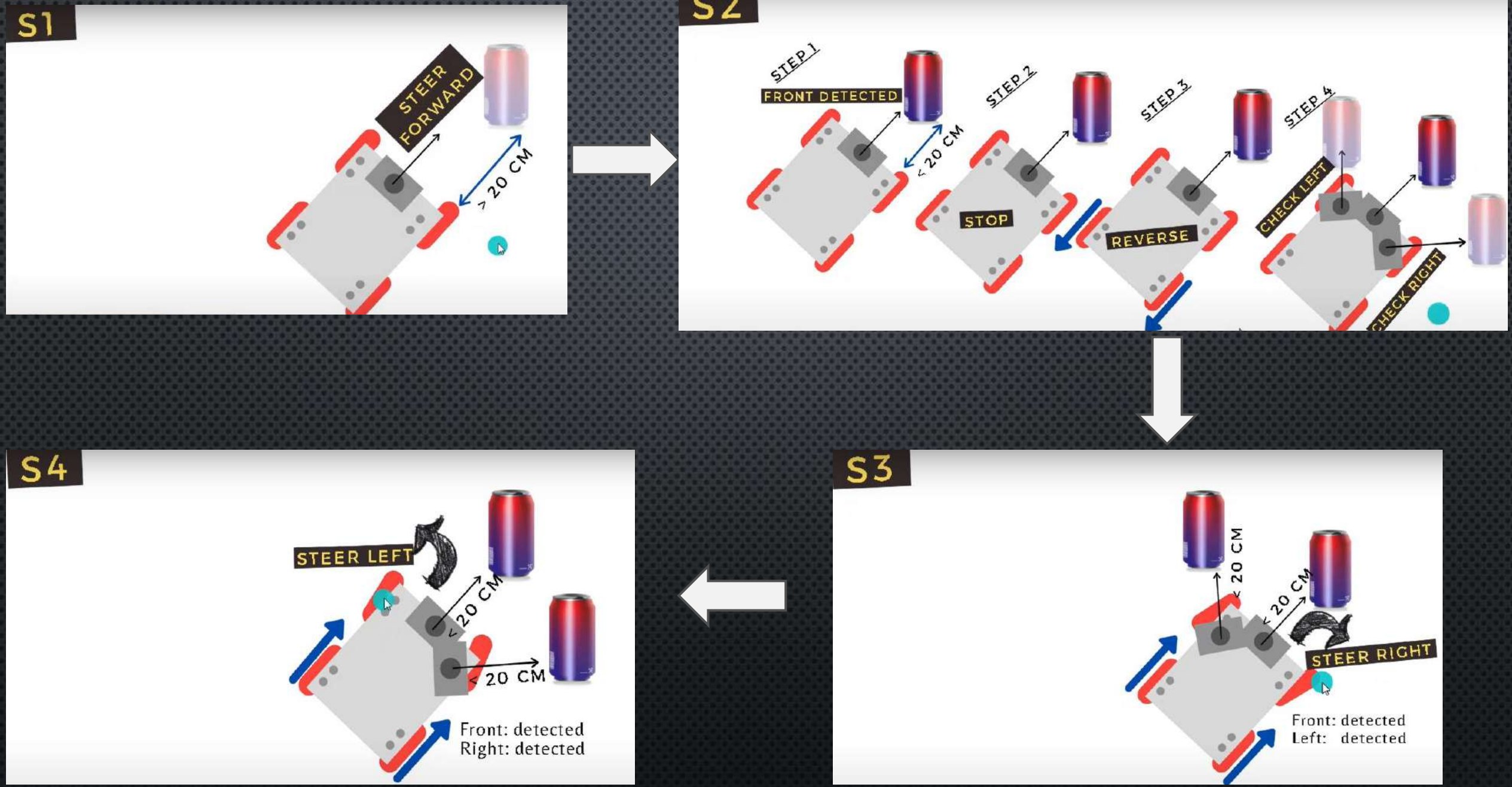
Our robot car's working

Our obstacle avoidance robot uses 3 ultrasonic sensors to measure the distance between itself and the obstacle, based on the input provided by these sensors, the micro controller(Arduino) is used to achieve the desired operation. The motors are connected to the motor drivers which are connected to the Arduino. In our simulation the object must be at least 20cm away from the sensors to not consider it as an obstacle.

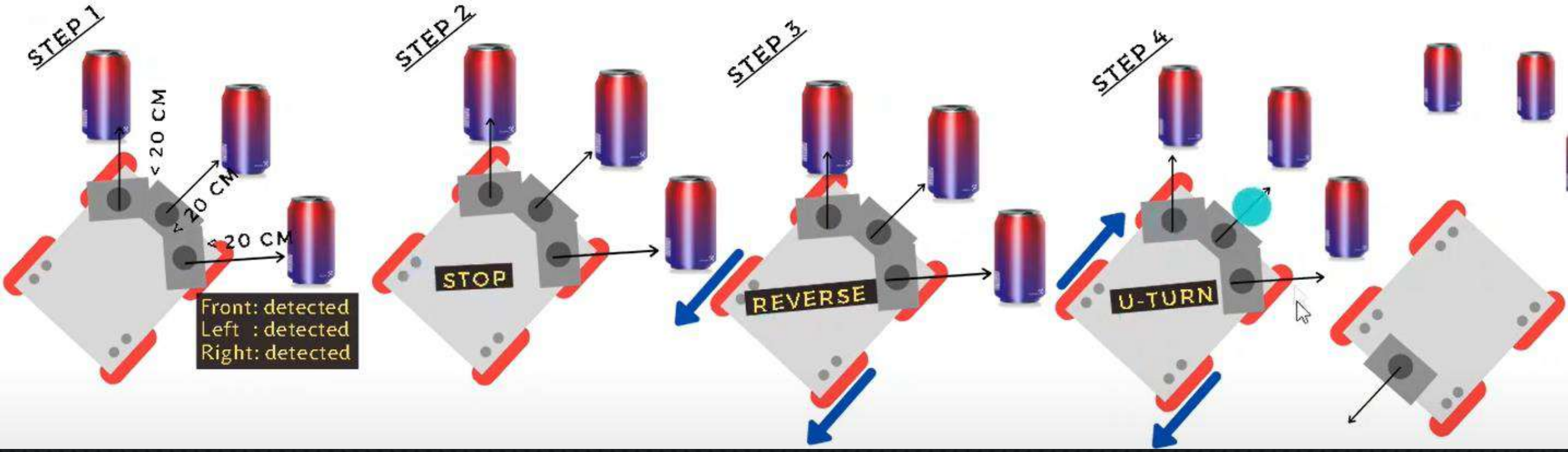
The Arduino prioritizes the front, right, left and back respectively. This means that if there is no obstacle in the front, it continues moving forward. If it detects any obstacle in the front, it checks for obstacles in the right and moves towards the right by switching off the right wheel. When the sensors detects the front and right being blocked, it checks for obstacles in the left and moves to the left by switching off the left wheel. If there are obstacles detected on the front, right and left, it checks for obstacles in the back and reverses itself and takes a U turn and the process is continued

OUR GOAL IS TO SIMULATE THE FOLLOWING CASES

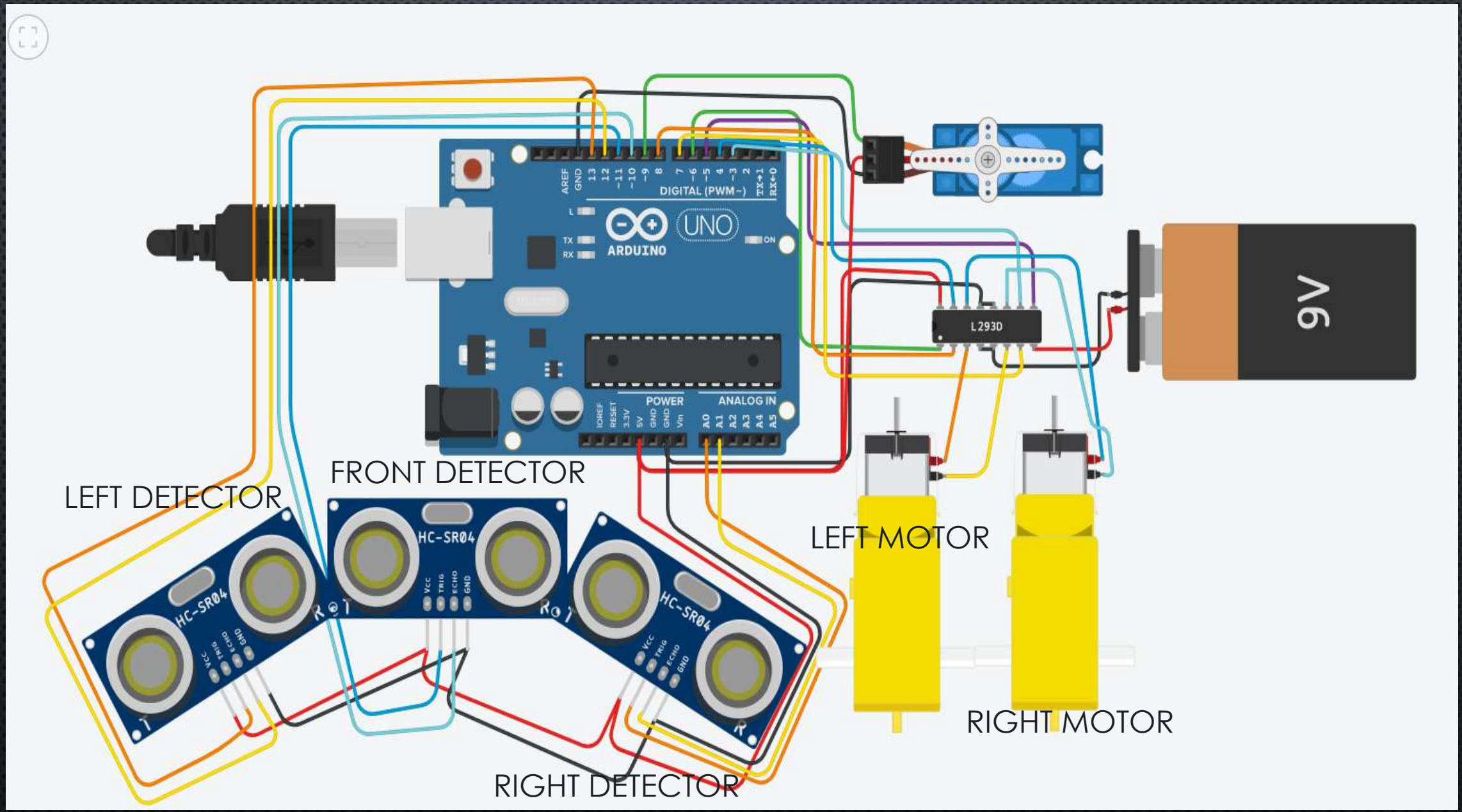
1. NO OBSTACLE IN THE FRONT
2. OBSTACLE IN FRONT AND LEFT SIDE
3. OBSTACLE IN FRONT AND RIGHT SIDE
4. OBSTACLE IN FRONT, LEFT AND RIGHT SIDE


S1
STEER FORWARD
> 20 CM
S2
STEP 1
FRONT DETECTED
< 20 CM
STEP 2
STOP
STEP 3
REVERSE
STEP 4
CHECK LEFT
CHECK RIGHT
S3
< 20 CM
< 20 CM
STEER RIGHT
Front: detected
Left: detected
S4
STEER LEFT
< 20 CM
< 20 CM
Front: detected
Right: detected

S5


STEP 1
< 20 CM
< 20 CM
< 20 CM
Front: detected
Left : detected
Right: detected
STEP 2
STOP
STEP 3
REVERSE
STEP 4
U-TURN

OUR DESIGN


LEFT DETECTOR
FRONT DETECTOR
RIGHT DETECTOR
LEFT MOTOR
RIGHT MOTOR
HC-SR04
L293D
9V
ARDUINO
UNO

OUR ARDUINO CODE :

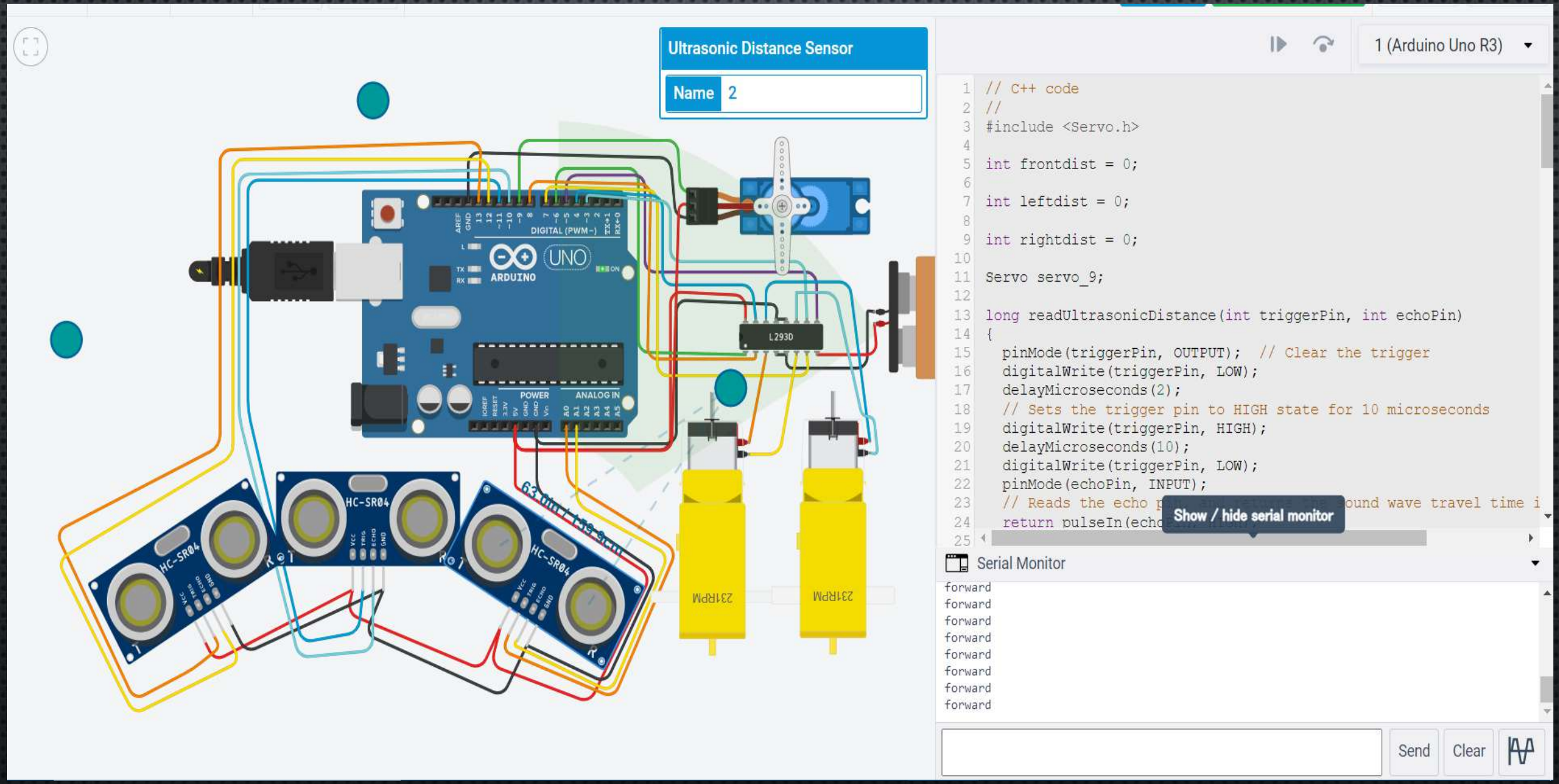
```
// C++ code
//
#include <Servo.h>

int frontdist = 0;

int leftdist = 0;

int rightdist = 0;

Servo servo_9;

long readUltrasonicDistance(int triggerPin, int echoPin)
{
  pinMode(triggerPin, OUTPUT);  // Clear the trigger
  digitalWrite(triggerPin, LOW);
  delayMicroseconds(2);
  // Sets the trigger pin to HIGH state for 10 microseconds
  digitalWrite(triggerPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(triggerPin, LOW);
  pinMode(echoPin, INPUT);
  // Reads the echo pin, and returns the sound wave travel time in microseconds
  return pulseIn(echoPin, HIGH);
}

void setup()
{
  servo_9.attach(9, 500, 2500);

  Serial.begin(9600);

  pinMode(6, OUTPUT);
  pinMode(5, OUTPUT);
  pinMode(8, OUTPUT);
  pinMode(4, OUTPUT);
  pinMode(7, OUTPUT);
  pinMode(3, OUTPUT);
}
```

```
void loop()
{
  servo_9.write(90);
  delay(1000); // Wait for 1000 millisecond(s)
  frontdist = 0.01723 * readUltrasonicDistance(11, 10);
  if (frontdist > 20) {
    // forward
    Serial.println("forward");
    servo_9.write(90);
    delay(1000); // Wait for 1000 millisecond(s)
    analogWrite(6, 255);
    analogWrite(5, 255);
    digitalWrite(8, HIGH);
    digitalWrite(4, HIGH);
    digitalWrite(7, LOW);
    digitalWrite(3, LOW);
  } else {
    // lstop
    Serial.println("lstop");
    analogWrite(6, 0);
    analogWrite(5, 0);
    digitalWrite(8, LOW);
    digitalWrite(4, LOW);
    digitalWrite(7, LOW);
    digitalWrite(3, LOW);
    delay(2000); // Wait for 2000 millisecond(s)
    // lreverse
    Serial.println("lreverse");
    analogWrite(6, 127);
    analogWrite(5, 127);
    digitalWrite(8, LOW);
    digitalWrite(4, LOW);
    digitalWrite(7, HIGH);
    digitalWrite(3, HIGH);
    delay(2000); // Wait for 2000 millisecond(s)
    leftdist = 0.01723 * readUltrasonicDistance(13, 12);
    rightdist = 0.01723 * readUltrasonicDistance(A0, A1);
```

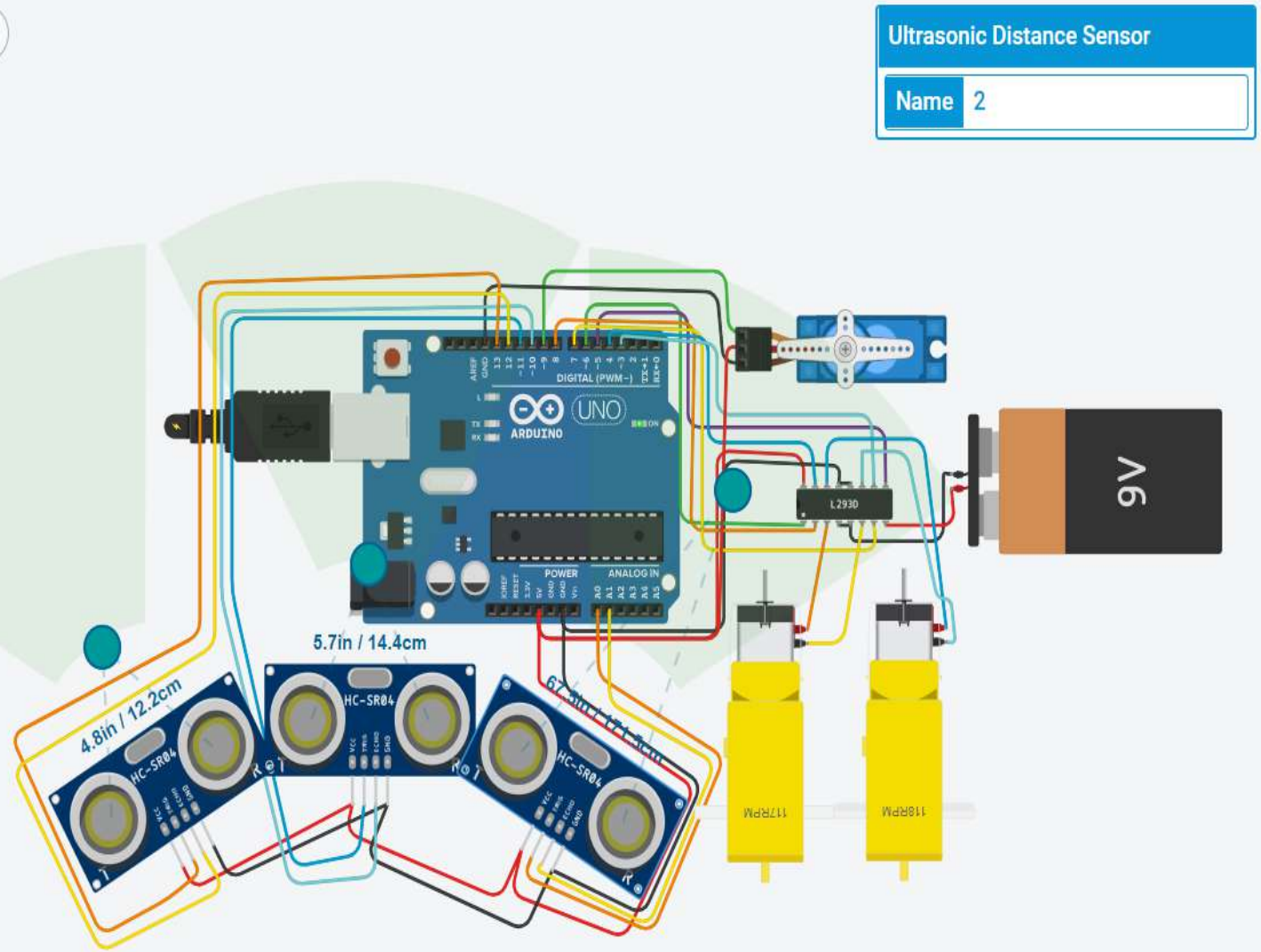
```
if (leftdist < 20 && rightdist > 20) {
  // right
  Serial.println("right");
  servo_9.write(0);
  delay(1000); // Wait for 1000 millisecond(s)
  analogWrite(6, 127);
  analogWrite(5, 127);
  digitalWrite(8, HIGH);
  digitalWrite(4, HIGH);
  digitalWrite(7, LOW);
  digitalWrite(3, LOW);
  delay(2000); // Wait for 2000 millisecond(s)
} else {
  if (leftdist > 20 && rightdist < 20) {
    // left
    Serial.println("left");
    servo_9.write(180);
    delay(1000); // Wait for 1000 millisecond(s)
    analogWrite(6, 127);
    analogWrite(5, 127);
    digitalWrite(8, HIGH);
    digitalWrite(4, HIGH);
    digitalWrite(7, LOW);
    digitalWrite(3, LOW);
    delay(2000); // Wait for 2000 millisecond(s)
  } else {
    // 2stop
    Serial.println("2stop");
    analogWrite(6, 0);
    analogWrite(5, 0);
    digitalWrite(8, LOW);
    digitalWrite(4, LOW);
    digitalWrite(7, LOW);
    digitalWrite(3, LOW);
    delay(2000); // Wait for 2000 millisecond(s)
```

```
        // 2reverse
        Serial.println("2reverse");
        analogWrite(6, 127);
        analogWrite(5, 127);
        digitalWrite(8, LOW);
        digitalWrite(4, LOW);
        digitalWrite(7, HIGH);
        digitalWrite(3, HIGH);
        delay(2000); // Wait for 2000 millisecond(s)
        // uturn
        Serial.println("uturn");
        analogWrite(6, 200);
        analogWrite(5, 150);
        digitalWrite(8, HIGH);
        digitalWrite(4, LOW);
        digitalWrite(7, LOW);
        digitalWrite(3, HIGH);
        delay(3000); // Wait for 3000 millisecond(s)
      }
    }
  }
}
```

NO OBSTACLES ON FRONT ,LEFT AND RIGHT



NO OBSTACLE ON RIGHT




```
// C++ code
//
#include <Servo.h>

int frontdist = 0;

int leftdist = 0;

int rightdist = 0;

Servo servo_9;

long readUltrasonicDistance(int triggerPin, int echoPin)
{
  pinMode(triggerPin, OUTPUT);  // Clear the trigger
  digitalWrite(triggerPin, LOW);
  delayMicroseconds(2);
  // Sets the trigger pin to HIGH state for 10 microseconds
  digitalWrite(triggerPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(triggerPin, LOW);
  pinMode(echoPin, INPUT);
  // Reads the echo pin, and returns the sound wave travel time i
  return pulseIn(echoPin, HIGH);
```

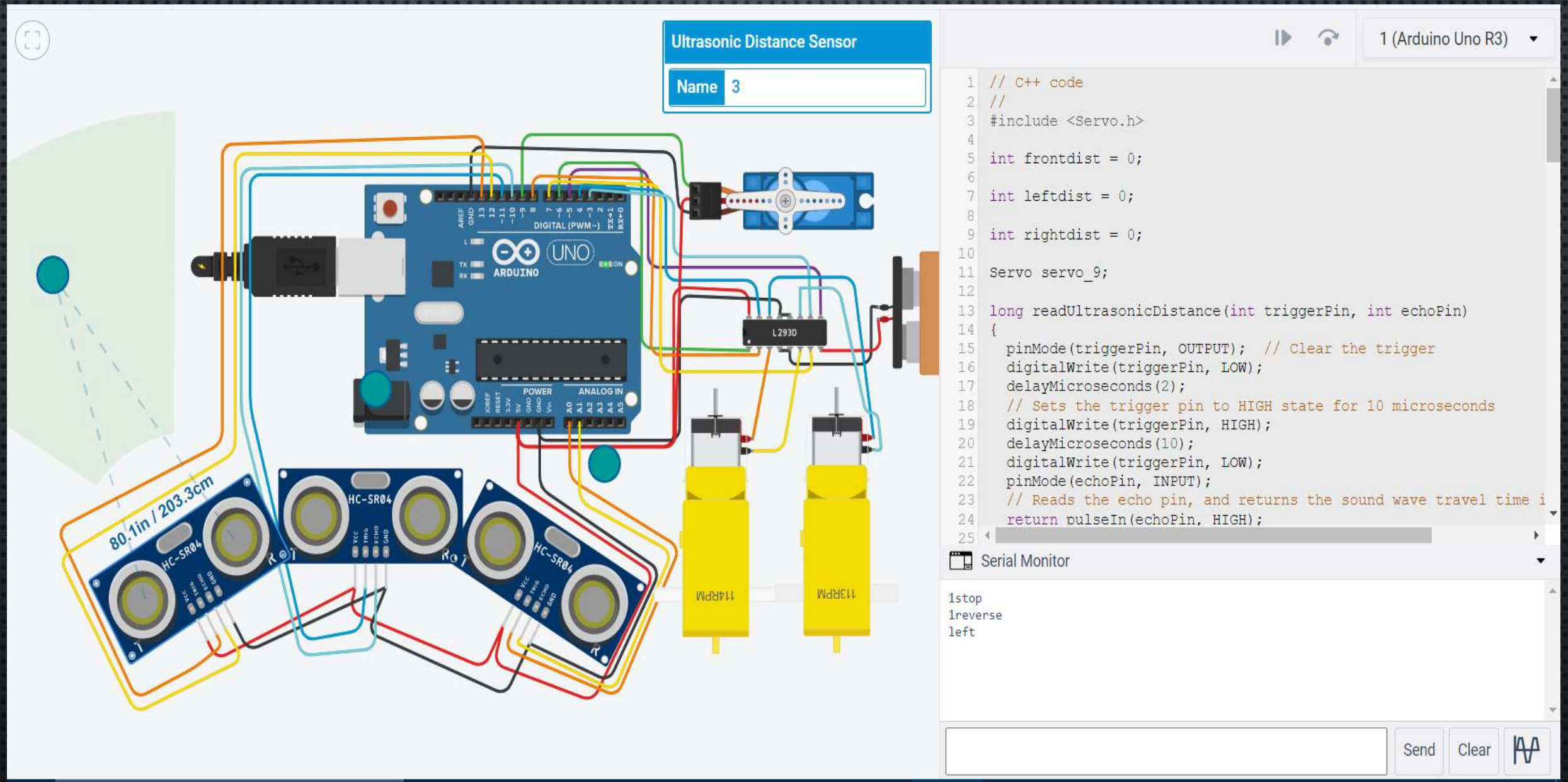
1 (Arduino Uno R3)

Serial Monitor

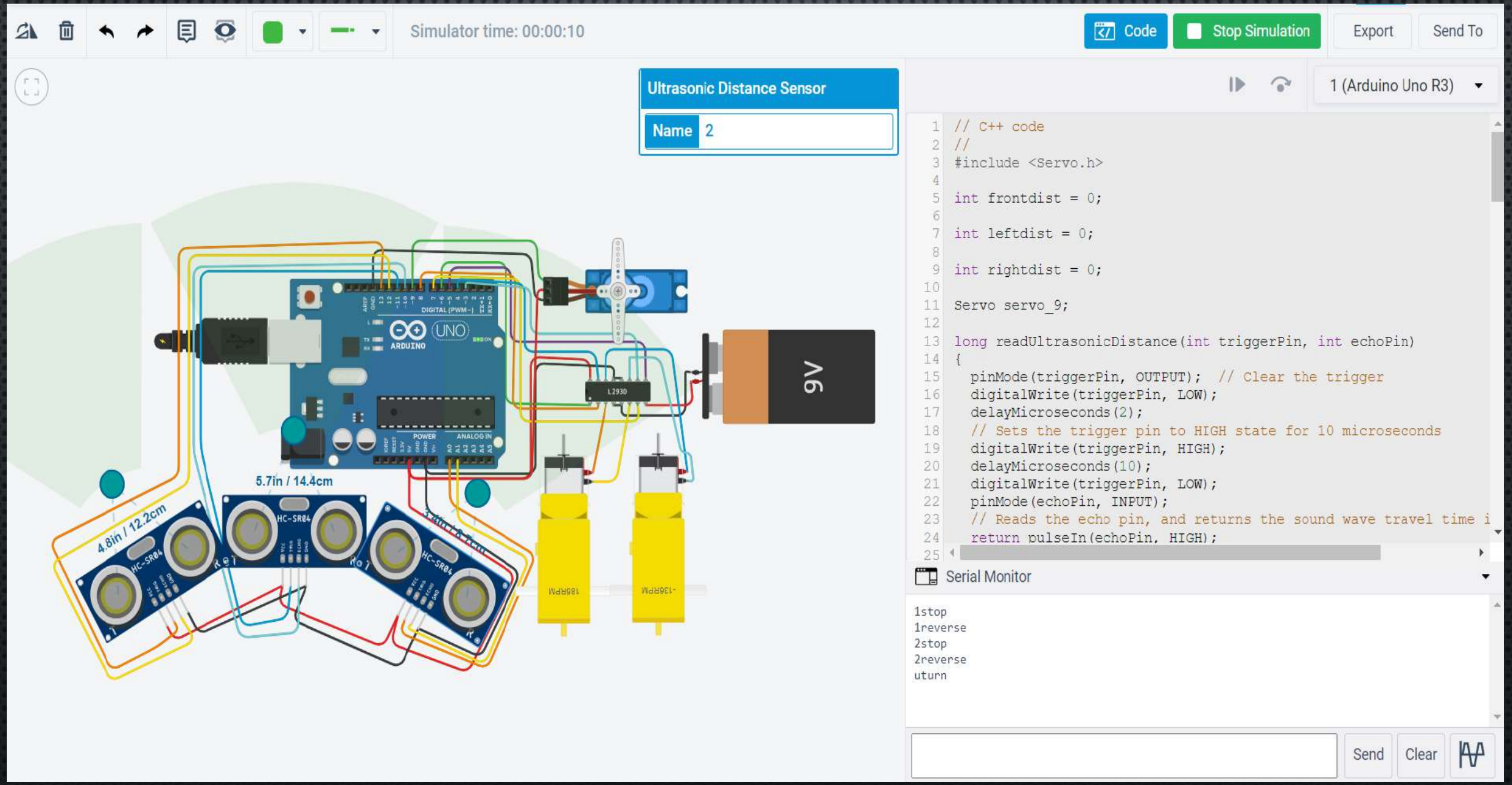
```
1stop
1reverse
right
```

Send Clear

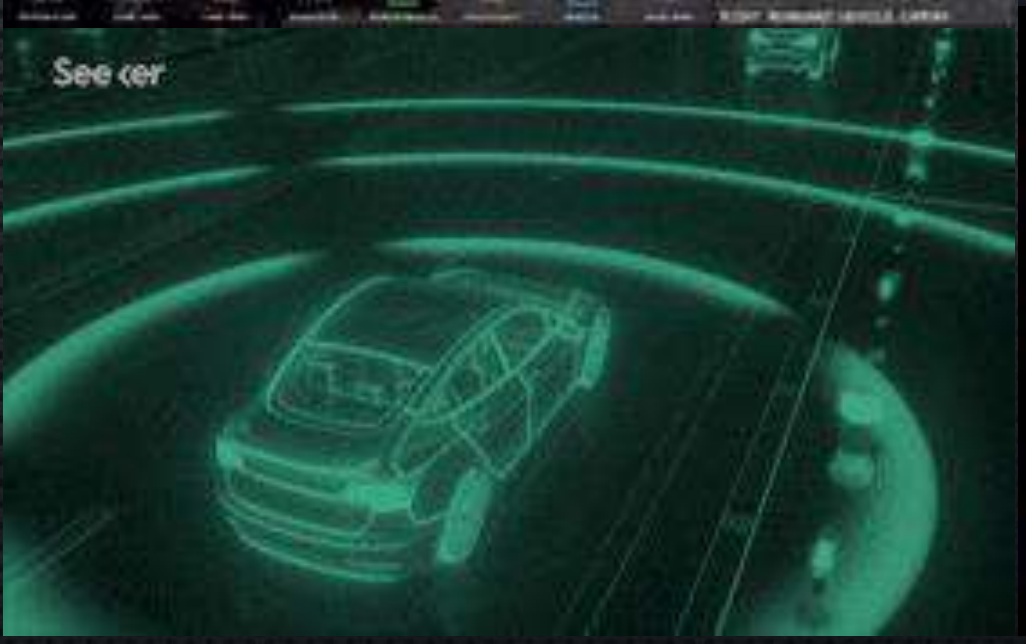
NO OBSTACLE ON LEFT



OBSTACLES ON FRONT ,LEFT AND RIGHT



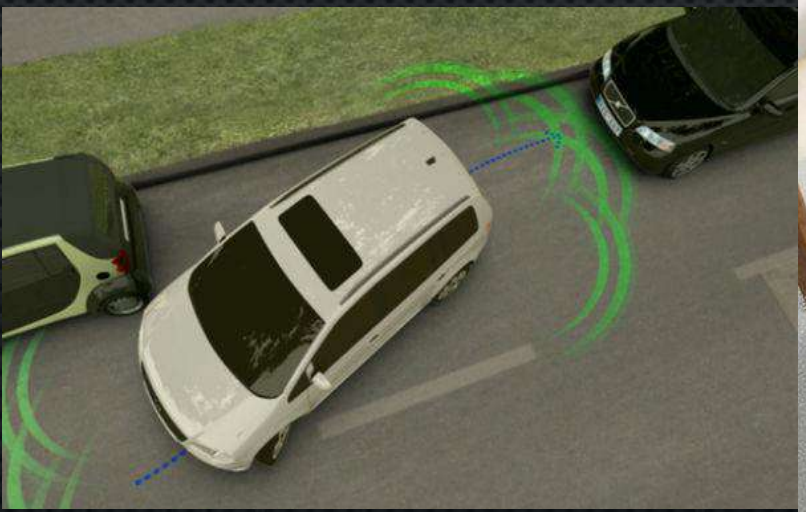
WE HAVE JUST SIMULATED HOW A SIMPLE OBSTACLE AVOIDANCE ROBOTIC VEHICLE WORKS. BUT THIS IS HOW A FULLY FLEDGED OBSTACLE AVOIDANCE VEHICLE LOOKS LIKE...



THANK YOU

WHAT ARE THE APPLICATIONS?

- FLOOR VACUUM CLEANER ROBOT
- AUTOMATIC PARKING ASSISTANT
- INDUSTRIES
- MILITARY OPERATIONS



REFERENCES:

https://www.instructables.com/obstacle-avoiding-robot-with-servo-motor-arduino/

https://create.arduino.cc/projecthub/sora-jy/obstacles-avoiding-robot-with-servo-motor-719f6b

WEBSITE USED FOR SIMULATION:

- https://www.tinkercad.com