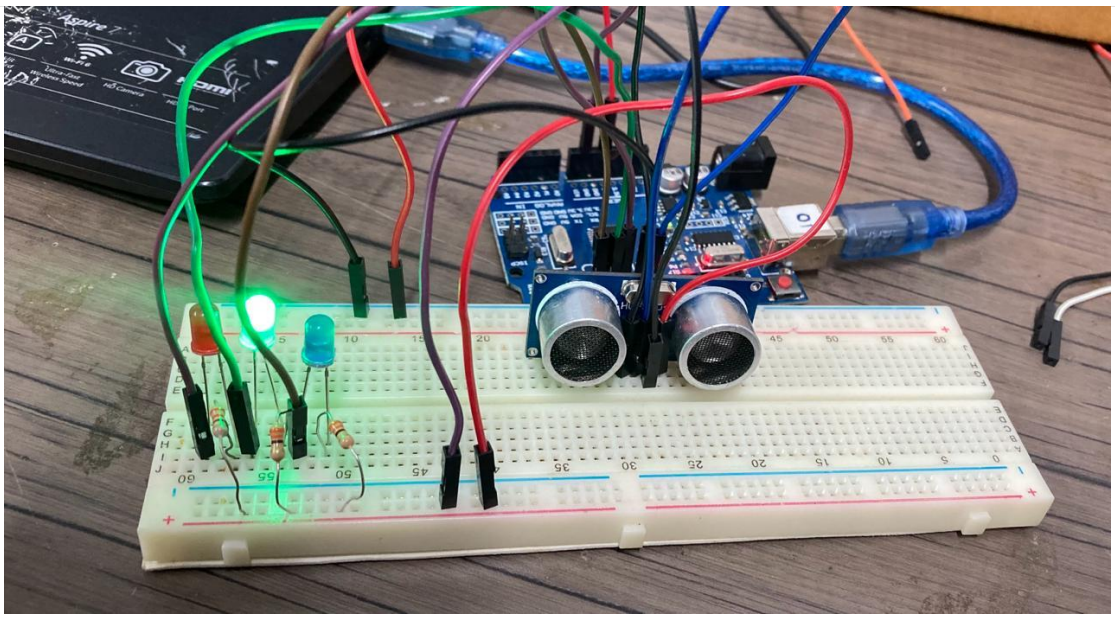
HC-SR04 SENSOR

INTRODUCTION:

Ultrasonic sensors are widely used in various applications, from distance measurement to object detection. Among them, the HC-SR04 ultrasonic sensor is one of the most commonly used modules for distance measurement with Arduino. In this project, if the distance is less than 10cm red LED will be high. If the distance is greater than 10cm and less than 100cm green led will become high and if distance is greater than 100cm blue led will become high.

.



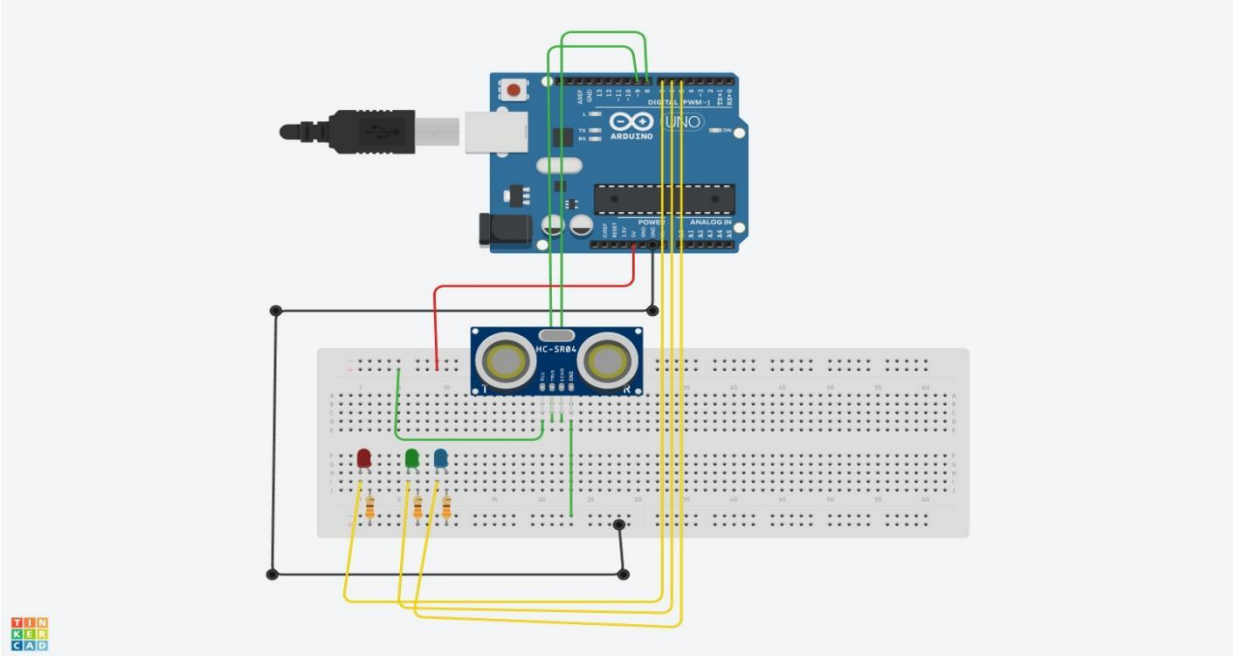
WORKING PRINCIPLE:

The HC-SR04 sensor works by emitting ultrasonic waves and then calculating the time it takes for the waves to bounce back after hitting an object. This time measurement can be used to calculate the distance to the object in front of the sensor**.**

COMPONENTS REQUIRED:

1. Arduino Board (e.g., Arduino Uno)
2. HC-SR04 Ultrasonic Sensor
   - Operating Voltage: DC 5V
   - Operating Current: <15mA
   - Operating Frequency: 40 kHz
   - Maximum Range: 4 meters (13 feet)
   - Minimum Range: 2 centimeters
   - Measuring Angle: 15 degrees
3. LED
4. Resistor (220Ω)
5. Breadboard
6. Jumper Wires

CIRCUIT DIAGRAM:



CIRCUIT CONNECTION:

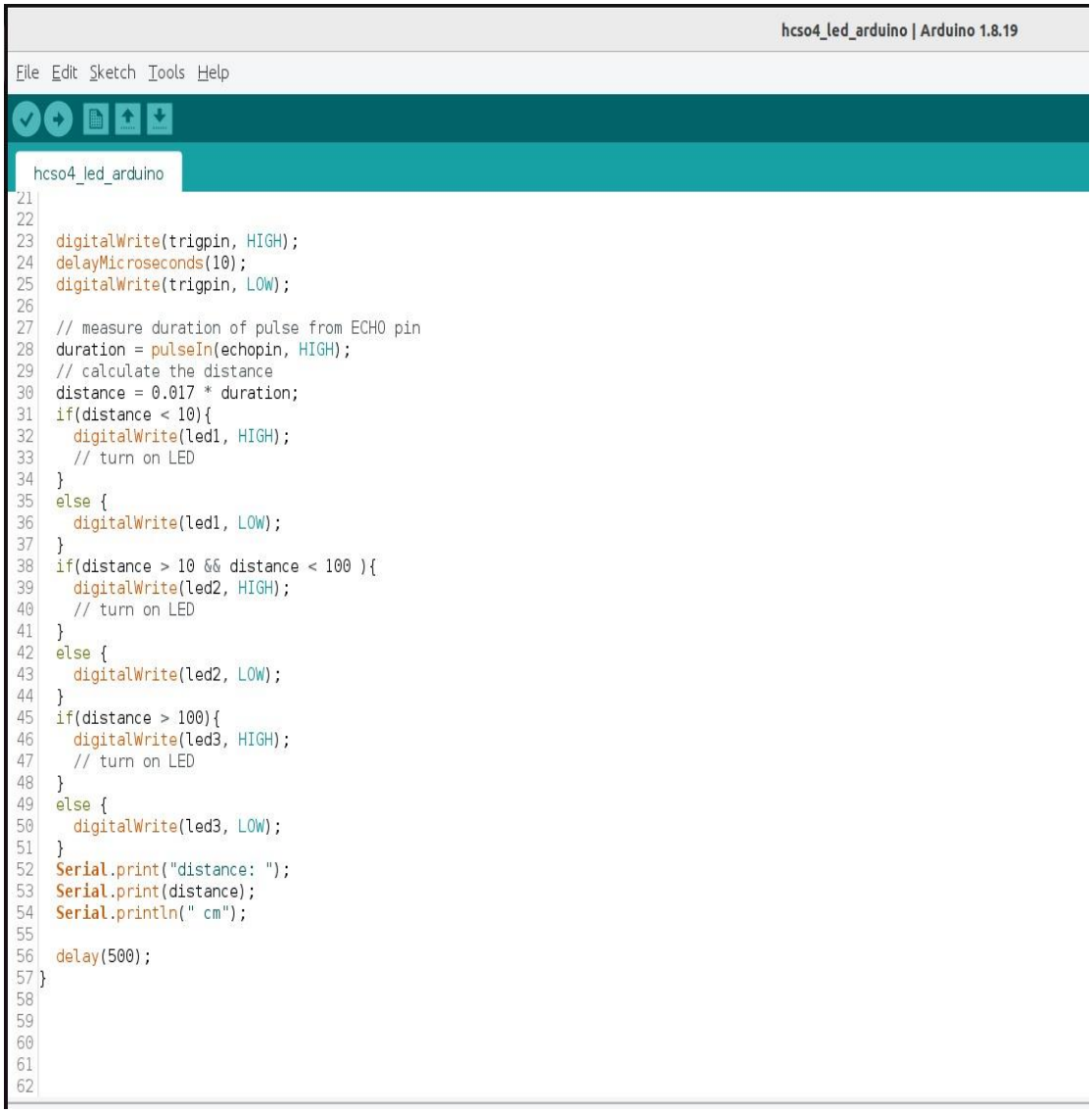
VCC pin of the HC-SR04 sensor to the 5V pin on the Arduino for power supply. Connect the GND pin of the sensor to any GND pin on the Arduino for the ground connection.Connect the Trig (trigger) pin of the sensor to a digital pin (e.g., pin 2) on the Arduino board. This pin is used

to trigger the sensor to send out an ultrasonic pulse. Connect the Echo pin of the sensor to another digital pin (e.g., pin 3) on the Arduino board. This pin receives the echo signal from the sensor, which is used to calculate the distance to the object.

ARDUINO CODE:

hcso4_led_arduino

```
const int trigpin = 9;
const int echopin = 8;
const int led1 = 7;
const int led2 = 6;
const int led3 = 5;
float duration, distance;

void setup() {
  pinMode(trigpin, OUTPUT); // set arduino pin to output mode
  pinMode(echopin, INPUT);  // set arduino pin to input mode
  pinMode(led1, OUTPUT);
  pinMode(led2,OUTPUT);
  pinMode(led3,OUTPUT);
  Serial.begin(9600);


}

void loop() {
  // put your main code here, to run repeatedly:


  digitalWrite(trigpin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigpin, LOW);

  // measure duration of pulse from ECHO pin
  duration = pulseIn(echopin, HIGH);
  // calculate the distance
  distance = 0.017 * duration;
  if(distance < 10){
    digitalWrite(led1, HIGH);
    // turn on LED
  }
  else {
    digitalWrite(led1, LOW);
  }
  if(distance > 10 && distance < 100 ){
    digitalWrite(led2, HIGH);
    // turn on LED
  }
  else {
```




```
21
22
23  digitalWrite(trigpin, HIGH);
24  delayMicroseconds(10);
25  digitalWrite(trigpin, LOW);
26
27  // measure duration of pulse from ECHO pin
28  duration = pulseIn(echopin, HIGH);
29  // calculate the distance
30  distance = 0.017 * duration;
31  if(distance < 10){
32    digitalWrite(led1, HIGH);
33    // turn on LED
34  }
35  else {
36    digitalWrite(led1, LOW);
37  }
38  if(distance > 10 && distance < 100 ){
39    digitalWrite(led2, HIGH);
40    // turn on LED
41  }
42  else {
43    digitalWrite(led2, LOW);
44  }
45  if(distance > 100){
46    digitalWrite(led3, HIGH);
47    // turn on LED
48  }
49  else {
50    digitalWrite(led3, LOW);
51  }
52  Serial.print("distance: ");
53  Serial.print(distance);
54  Serial.println(" cm");
55
56  delay(500);
57 }
58
59
60
61
62
```

PROCEDURE:

- Setup Hardware:

Gather all necessary components.
Connect them according to the circuit diagram.

- Install Arduino IDE:

Download and install Arduino IDE from the official website.

- Open Arduino IDE:

Launch the Arduino IDE software.

- Write Code:

Compose your program using Arduino programming language (based on C/C++).
Write setup and loop functions.

- Verify Code:

Click on the Verify button (checkmark icon) to check for any errors in the code.

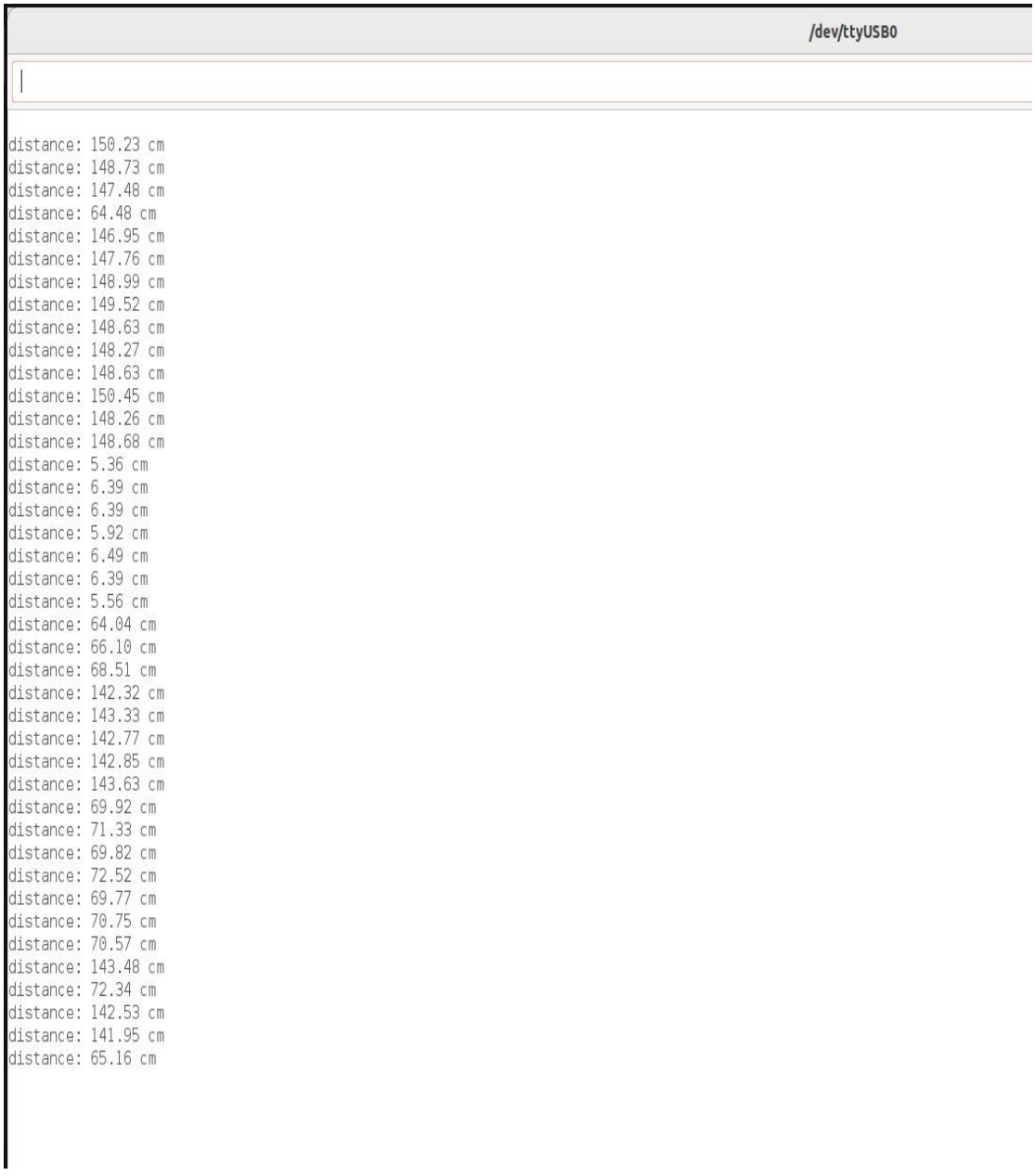
- Upload Code:

Connect Arduino board to the computer via USB.
Select the correct board and port from Tools menu.
Click on the Upload button (right arrow icon) to upload the code to the Arduino board.

- Test:

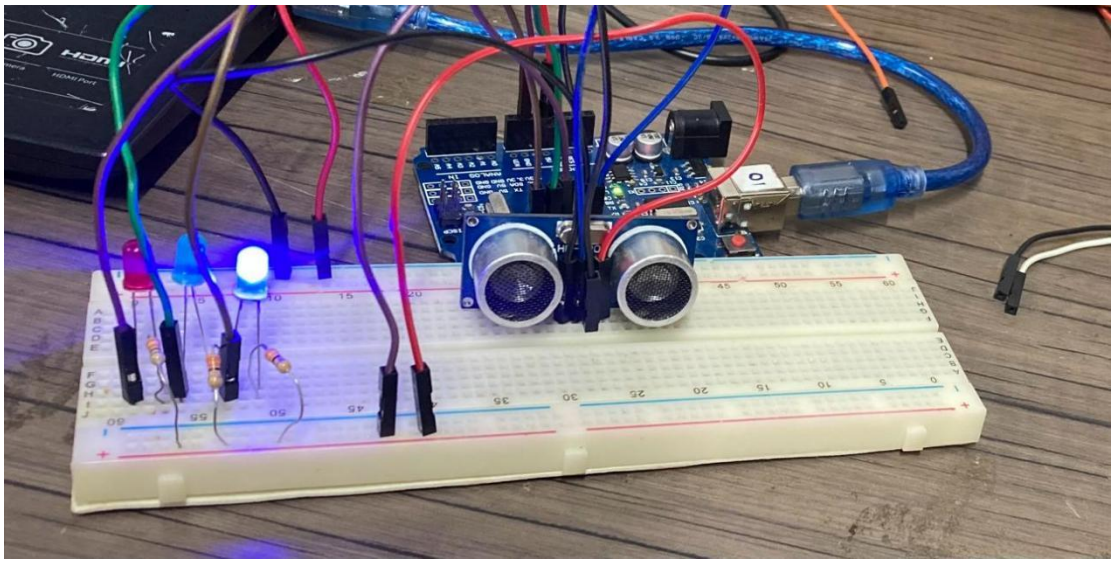
Make sure the hardware is powered on.

SERIAL MONITOR OUTPUT:



RESULT:

The output is obtained and the HC-SR04 sensor was integrated with arduino. When the object distance is less than 10cm red led became high and when it is greater than 10cm but less than 100cm the green was high and when the distance was greater than 100cm blue led was high.



Example: Object distance greater than 100cm

Done by:
AZWA HARSHAD
Intern at ICFOSS