SENSOR DISTANCE

REPORT

Presented To
DR AZIZA

Presented By
DURID ALMASRI - HASSAN ALSAGGAF - MOHAMMED ATTAS

TABLE OF CONTENT

Problem Statment:

The project aims to develop a reliable, accurate distance measurement system for robotics and automation, utilizing a distance sensor and an efficient circuit for real-time readings.

Objective:

The primary objective of the Distance Sensor project is to create a robust system that can measure distances accurately using ultrasonic technology. Specific goals include:

- **Implementing a circuit design that integrates an ultrasonic sensor with an Arduino microcontroller.**
- **Ensuring the system is capable of real-time distance monitoring.**
- **Testing the circuit under various conditions to validate its accuracy and reliability.**

Desing circuit:

The circuit design involves connecting an ultrasonic sensor to an Arduino microcontroller. The ultrasonic sensor emits sound waves, and the Arduino processes the received signals to calculate the distance between the sensor and an object. The following components are crucial to the circuit:

- **BreadBoard**
- **Jumper Wires**
- **3*(1K) Resistors**
- **Blue Led (5mm , 3.4V , 30mA)**
- **Yellow Led (5mm , 2V , 20mA)**
- **Red Led (5mm , 2V , 20mA)**
- **Ultrasonic Sensor (HC-SR04)**
- **Active Buzzer**
- **Arduino Microcontroller**

Components Information:

- **Ultrasonic Sensor:** Converts electrical energy into ultrasonic waves and measures the time taken for the waves to return after hitting an object.

- **Arduino Microcontroller:** Processes the incoming signals from the sensor, calculates distance, and displays the results

.

- **Display Interface (optional):** LED display to showcase real-time distance measurements.

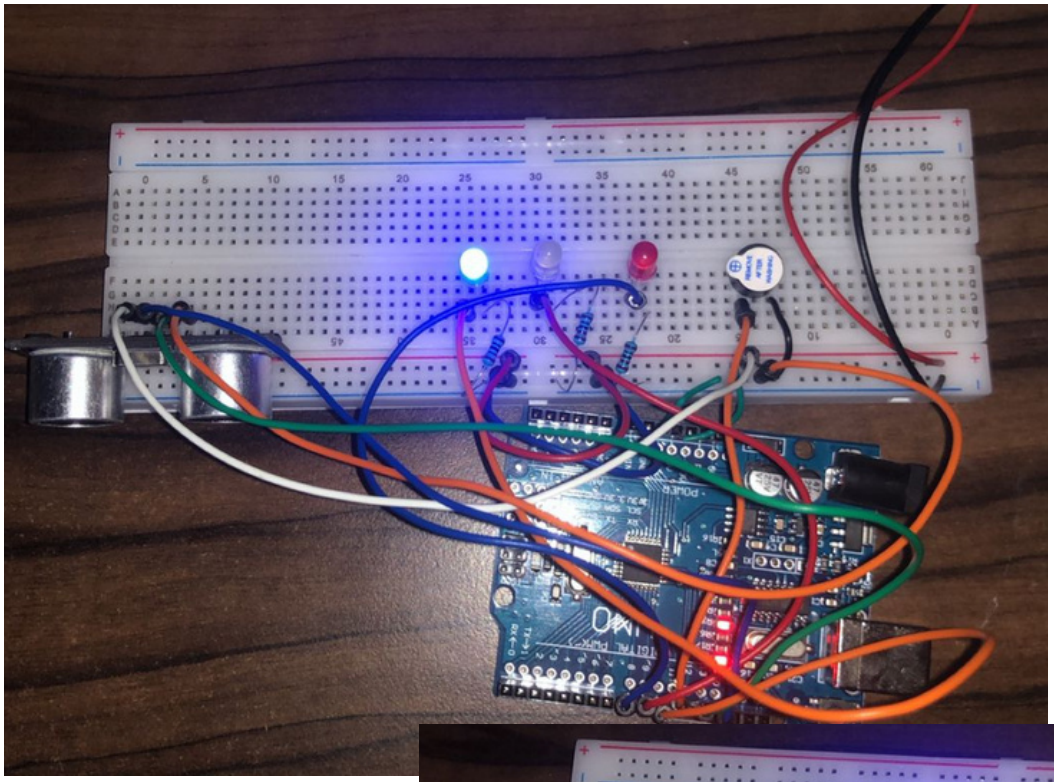
- **Power Supply:** Provides the necessary power to the entire circuit.

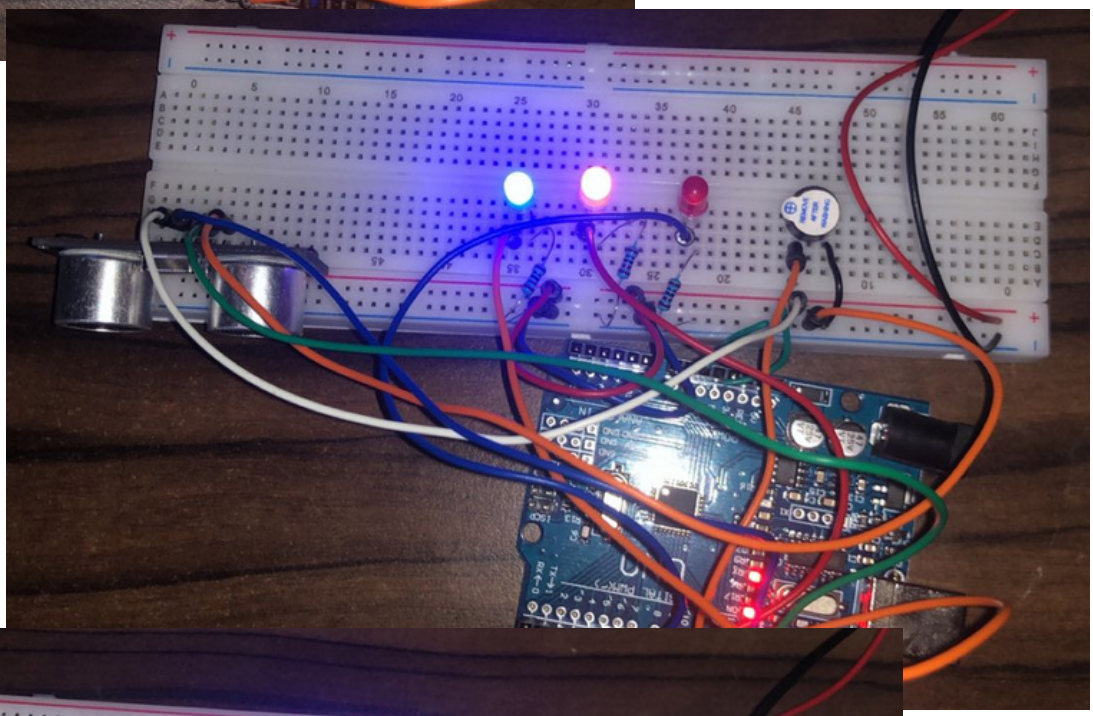
Test of Circuit:

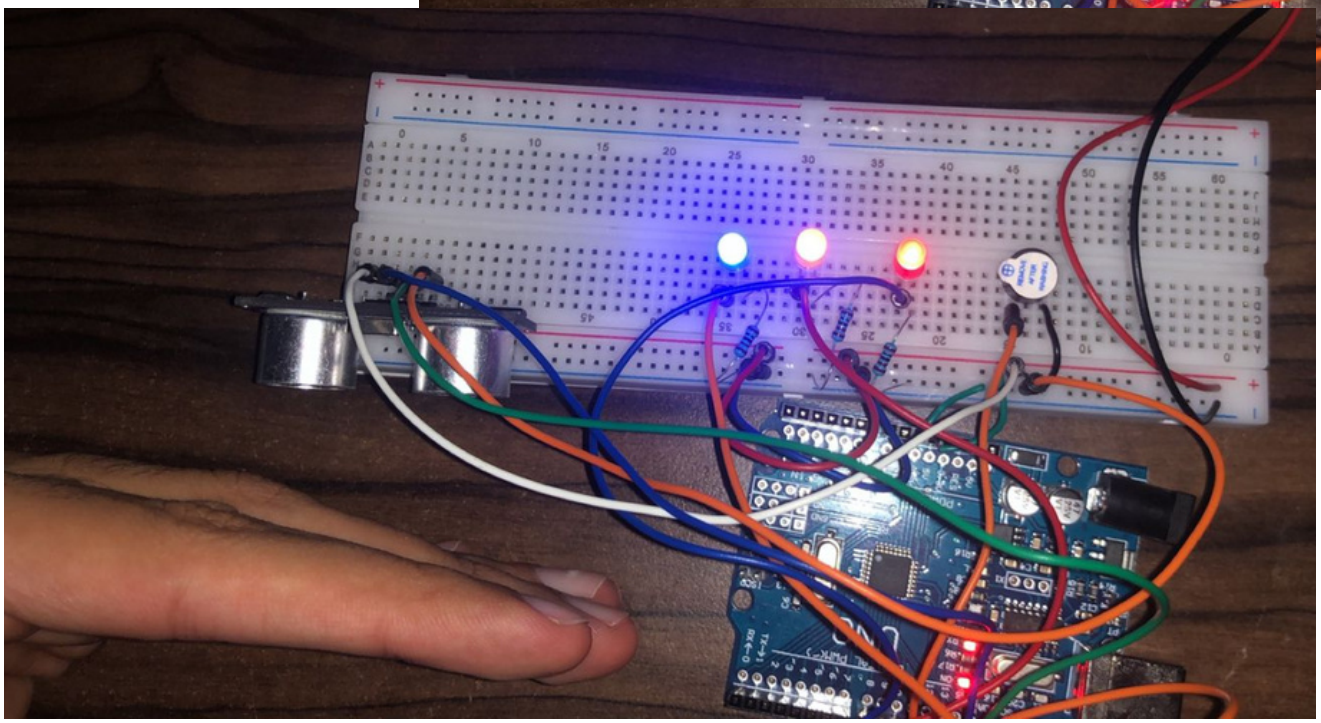
The circuit undergoes rigorous testing to ensure functionality and accuracy. The tests involve:

- Verifying proper connections and power supply.
- Conducting initial distance tests to identify any inconsistencies.
- Calibrating the system for optimal performance in different environments.

Results with Screenshots:







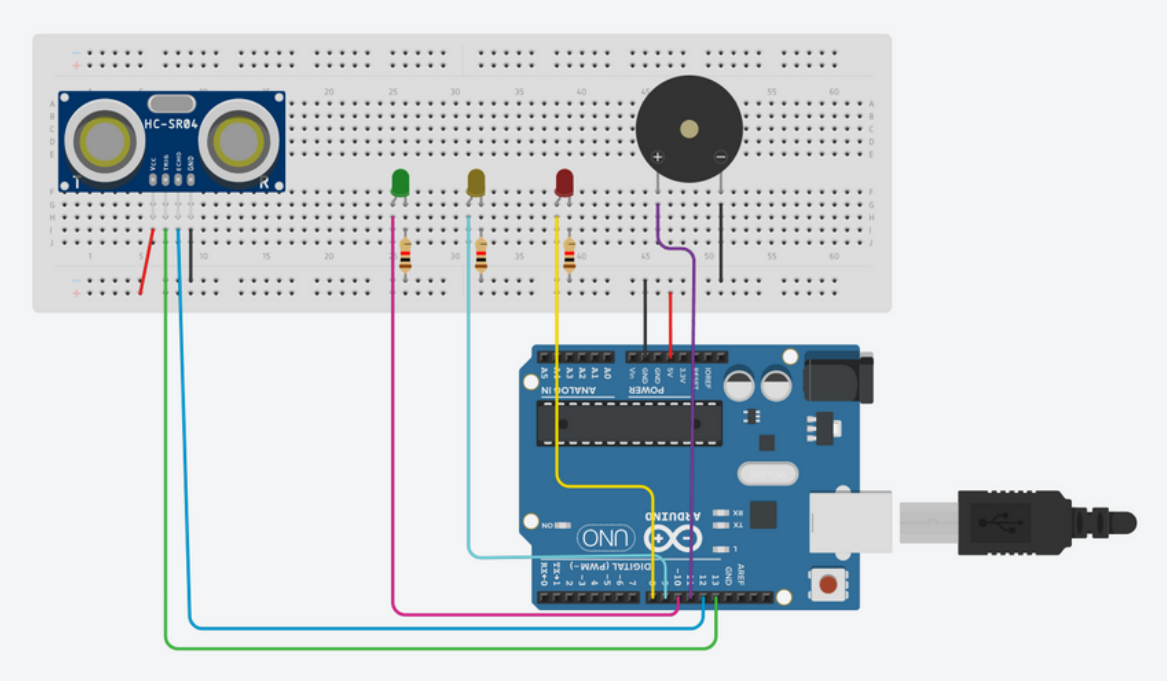
Code:

```
const int trigPin = 13;
const int echoPin = 12;

const int buzzer = 11;// use a Active buzzer for this project. Active buzzers will make noise with DC current

const int greenLED = 10;
const int yellowLED  = 9;
const int redLED = 8;

long duration;
double distance;

double maxDistanceInches = 12; // this is how far we want our sensor to read, ignore everything farther than 12 inches.

void setup() {
  pinMode(trigPin, OUTPUT); // Sets the trigPin as an Output
  pinMode(echoPin, INPUT); // Sets the echoPin as an Input
  pinMode(buzzer, OUTPUT);
  pinMode(greenLED, OUTPUT);
  pinMode(yellowLED, OUTPUT);
  pinMode(redLED, OUTPUT);
  digitalWrite(buzzer, LOW); // set all outputs low
  digitalWrite(greenLED, LOW);
  digitalWrite(yellowLED, LOW);
  digitalWrite(redLED, LOW);
  Serial.begin(9600); // Starts the serial communication
}
```

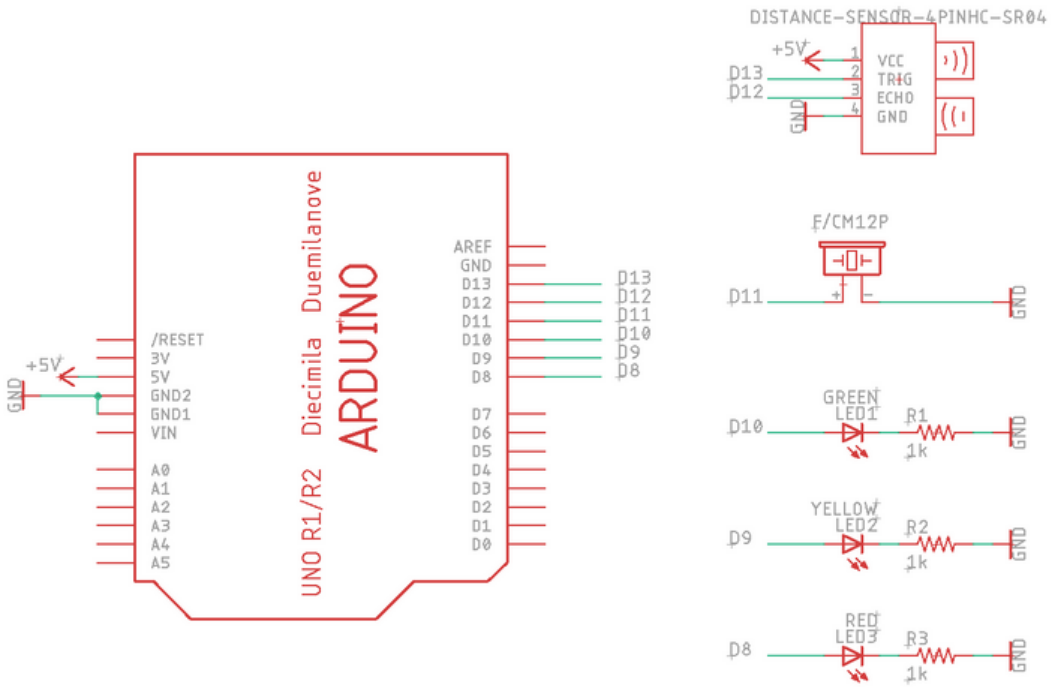
```
}
void loop() {
  // Clears the trigPin
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);

  // Sets the trigPin on HIGH state for 10 micro seconds
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);

  // Reads the echoPin, returns the sound wave travel time in microsecond
  duration = pulseIn(echoPin, HIGH);

  // Calculating the distance in cm
  distance = duration * 0.034 / 2;

  //to convert cm to inches
  distance = distance / 2.54;

  //Light up light depending on distance!
  if(distance < maxDistanceInches)
     digitalWrite(greenLED, HIGH);
  else
     digitalWrite(greenLED, LOW);

  if(distance < (maxDistanceInches* 0.66))
    digitalWrite(yellowLED, HIGH);
```

```
if(distance < (maxDistanceInches* 0.66))
  digitalWrite(yellowLED, HIGH);
else
  digitalWrite(yellowLED, LOW);

 if(distance < (maxDistanceInches * 0.33)){
  digitalWrite(redLED, HIGH);
  digitalWrite(buzzer, HIGH); // if too close to sensor sound the alarm!!
 }
 else{
 digitalWrite(redLED, LOW);
 digitalWrite(buzzer,LOW); // if not close turn off alaram
 }

// Prints the distance on the Serial Monitor
Serial.print("Distance: ");
Serial.println(distance);
```

Hookup Diagram:



Schematic Diagram:



Conclusions:

The Distance Sensor project is an accurate, adaptable distance measurement system that integrates an ultrasonic sensor with an Arduino microcontroller, providing real-time results with minimal deviation, making it a valuable solution for precise distance measurements in various environments.

References:

ChatGPT : https://openai.com/chatgpt

Tiktok : https://www.tiktok.com/@tsj_electronics/video/7278787152974990634?_r=1&_t=8hjP2ZWpINc

GitHub - TSJ-Electronics/Distance-Sensor-Project: Simple arduino distance sensor

https://www.electroschematics.com/wp-content/uploads/2013/07/HCSR04-datasheet-version-1.pdf

https://descargas.cetronic.es/WW05A3SBQ4-N.pdf

https://us.rs-online.com/m/d/6355b8aba0b01578df0bb7b871ceefd7.pdf

https://www.farnell.com/datasheets/1660999.pdf

http://www.electronicoscaldas.com/datasheet/LTE12-Series.pdf

1K Resistor Datasheet.pdf. (n.d.). https://docs.google.com/viewer?a=v&pid=sites&srcid=b25pZC5vcmVnb25zdGF0ZS5lZHV8ZWNlNDR4MjAxMzY4fGd4OjMwNGFjN2MwY2Q2NjA2ZjU

https://docs.arduino.cc/resources/datasheets/A000066-datasheet.pdf