PROJECT REPORT

ON

"Ultrasonic Sensor Distance Measure"

THIS PROJECT & RESEARCH REPORT HAS BEEN INDEPENDENTLY DEVELOPED BY ME AND IS NOT SUBMITTED TO ANY COLLEGE OR UNIVERSITY FOR ACADEMIC EVALUATION

Developed by

Name:-Suzen Kumar Mohanty (**Reg.No**.2421331063)



SKMP PROJECT MANAGEMENT

DEPARTMENT OF IOT

POLISHLINE.BALASORE, ODISHA, INDIA

MAY-2026

“Dedicated to my beloved parents ,and teachers who have worked hard throughout my education”

SKMP PROJECT MANAGEMENT

CERTIFICATE

This is to certify that the project entitled"

Ultrasonic Sensor Distance Measure”

Has been independently developed and completed by Suzen Kumar Mohanty for the purpose of enhancing technical knowledge, skills, and practical understanding.

This project is a self-initiated work carried out with person al interestand research. It has not been submitted to any college, university, or institution for the award of any degree or diploma.

The work presented in this project is genuine and has been completed by the developer himself.



Suzen Kumar Mohanty

(Project Developer)

Date: 20/05/2026

Place:-Balasore,Odisha,India

SELF ATTESTATION

I here by declare that I have personally worked on the project entitled "**Ultrasonic Sensor Distance Measure** ".

All the data, analysis, and implementation presented in this project are the result of my own efforts and genuine work .Any external sources of information, if used, have been properly acknowledged.

This project is created solely for learning , skill development ,and professional improvement, and has not been submitted to any institution for academic purposes.



SKMP PROJECT MANAGEMENT

Suzen Kumar Mohanty

(Project Developer)

Date: 20/05/2026

Place:Balasore,Odisha,India

DECLARATION

I hereby declare that the project titled "**Ultrasonic Sensor Distance Measure"i**s an original work carried out by me based on my own idea, research, and implementation.

This project that has not been submitted previously ,in part or in full, to any college, university, or institution for the award of any degree, diploma, or certification.

I confirm that all the information presented in this project is true to the best of my knowledge. Any references or sources used have been properly acknowledged.

I take full responsibility for the authenticity and originality of this work.

SKILL IP PROJECT MANAGEMENT

Name: Mr. Suzen Kumar Mohanty

Branch: Computer Science And Engineering

Institution:[Srinix Collage Of Engineering ,Independent]

Date:[20/05/2026]

My PortPolio Link:-https://suzen-kumar-mohanty-8fdv.vercel.app/

ACKNOWLEDGMENTS

I would like to express my sincere gratitude to everyone who contributed, directly or indirectly, to the development of this project, "**Ultrasonic Sensor Distance Measure**."

This project is the result of my personal interest, dedication, continuous learning, and exploration of electronics, automation, sensor technology, and control systems.

I am thankful to the developers of Arduino, the creators of open-source hardware and software resources, and the technical communities whose documentation, tutorials, and knowledge-sharing platforms helped me understand and implement the concepts used in this project.

I would also like to acknowledge the importance of self-learning, practical experimentation, and problem-solving throughout the development process. These experiences helped me strengthen my technical knowledge and improve my understanding of real-world automation systems.

Finally, I am grateful to everyone who supports innovation, DIY engineering, and independent technical projects. This work reflects my enthusiasm for learning, creating, and exploring new possibilities in technology.

Developed By:

Suzen Kumar Mohanty

ABOUT MYSELF



I am Mr. Suzen Kumar Mohanty , a passionate and dedicated Web Developer and SOC Analyst (Level 1) from the field of Computer Science and Engineering, with strong expertise in HTML, CSS, JavaScript, and I specialize in building scalable, responsive, and high-performance web applications that deliver seamless user experiences across all devices and screen sizes.

With a keen eye for detail, I focus on writing clean, maintainable, and well-structured code. My development approach is centered around Creating intuitive user interfaces, ensuring both functionality and aesthetic quality. I follow modern front-end development practices such as component-based architecture, responsive design principles, and performance optimization techniques to build efficient and user-friendly applications.

Beyond web development, I actively work on IoT-based solutions, where I integrate hardware and software to create smart, real-world systems. I have hands-on experience with Arduino Uno and various sensors, enabling me to develop projects such as environmental monitoring systems, smart automation solutions, and real-time data tracking applications.

One of my key strengths lies in connecting IoT devices with web technologies. I build interactive dashboards that allow real-time data visualization, giving users the ability to monitor, analyze, and control devices remotely. My work focuses on bridging the gap between the physical and digital world by transforming raw sensor data into meaningful insights.

I am deeply interested in developing innovative, data-driven solutions that combine web technologies with embedded systems. My goal is to create impactful applications that solve real-world problems while improving efficiency, automation, and user experience.

With a strong passion for continuous learning and innovation, I consistently explore new tools, frameworks, and technologies to enhance my skills and deliver high-quality, future-ready solutions.

Name: Mr. Suzen Kumar Mohanty

Branch: Computer Science And Engineering

MyPortPolioLink:-https://suzen-kumar-mohanty-8fdv.vercel.app/

ABSTRACT

Distance measurement is essential in robotics, automation, vehicle parking assistance, level measurement, obstacle avoidance, and industrial safety systems. This project uses the HC-SR04 ultrasonic sensor with an Arduino Uno to calculate distance based on the time-of-flight principle. The Arduino sends a short trigger pulse to the sensor; the sensor emits an ultrasonic burst, receives its reflection from an object, and returns an Echo pulse whose duration is proportional to the distance. The Arduino calculates the distance using the speed of sound and displays the result in centimetres and metres.atl.aim.gov+2

The proposed system is inexpensive, easy to assemble, and suitable for educational and prototype applications. The HC-SR04 can measure distances from approximately 2 cm to 400 cm, with a stated accuracy of up to 3 mm under favourable conditions

LIST OF FIGURES

Figure No.	Figure Title	Page No.
01	UltraSonic Sensor Distance Measure	12
02	Arduino Uno Board	18
03	Ultrasonic Sensor	19
04	Resistor 220 ohm	19
05	Bread board	20
06	Servo Motor(Sg90)	21
07	Jumper Wire	22
08	Led	23
09	Usb Cable	24
10	Power Supply 9v Battery with Connector	25
11	Circuit Diagram	27

SKMP PROJECT MANAGEMENT
SKMP PROJECT MANAGEMENT

INDEX

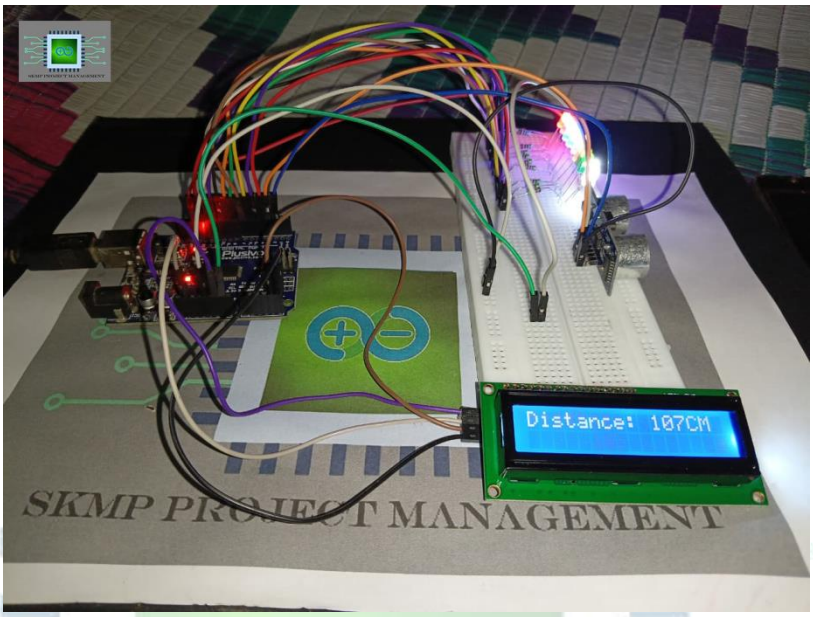
Sl. No.	Contents / Topics	Page No.
1	Introduction	2
1.1	Background	2
1.2	Problem Statement	3
1.3	Objectives	3
2	Literature Review	4
2.1	Ultrasonic Sensing	4
2.2	HC-SR04 Sensor	4
2.3	Arduino in Measurement Systems	4
3	Working Principle	5
3.1	Operation Sequence	5
4	System Architecture	6
4.1	Block Diagram	6
5	Hardware Requirements	7
5.1	Arduino UNO Board	7
5.2	HC-SR04 Ultrasonic Sensor	8
5.3	Resistor 220 Ohm	9
5.4	Breadboard	10
5.5	Servo Motor (SG90)	10
5.6	Jumper Wires	11
5.7	LED	12
5.8	USB Cable	13
5.9	Power Supply (9V Battery with Connector)	14
6	Software Requirements	15
7	Circuit Diagram	16
7.1	Connection Notes	17
8	Arduino Program Code	18
8.1	LCD I2C Address Finder Code	19

SKMP PROJECT MANAGEMENT
SKMP PROJECT MANAGEMENT

8.2	Distance Measuring Code	21
8.3	Effect 2 Code	25
9	Working of the Project	28
10	Testing Procedure	28
10.1	Test Setup	28
10.2	Sample Observation Table	29
11	Calibration	29
12	Advantages	30
13	Limitations	30
14	Applications	31
15	Conclusion & Future Scope	32
15.1	Future Scope	33
16	References	34–35
17	Project Link	36

SKMP PROJECT MANAGEMENT
SKMP PROJECT MANAGEMENT

1. Introduction



Fig(01)

1.1 Background

Ultrasonic sensors are widely used for non-contact distance measurement. They operate using sound waves above the range of human hearing, typically around 40 kHz in the HC-SR04 module. When an ultrasonic wave strikes an object, part of it is reflected back to the sensor. By measuring the time between transmission and reception, the distance to the object can be calculated.handsontec+1

SKMP PROJECT MANAGEMENT

The Arduino platform is ideal for this project because it provides simple digital input/output control, accurate timing functions, serial communication, and extensive community support. The project demonstrates the integration of sensing, embedded programming, signal timing, and real-time measurement.

1.2 Problem Statement

Manual distance measurement is inconvenient in many automation and safety applications. Optical sensors may be affected by colour, transparency, lighting conditions, or surface reflectivity. Therefore, an ultrasonic-based system is useful because it can measure distance without physical contact and works independently of visible light.handsontec+1

1.3 Objectives

- To design and build an Arduino-based ultrasonic distance-measurement system.
- To measure distance without physical contact using an HC-SR04 sensor.
- To calculate distance from ultrasonic echo time.
- To display distance readings through the Serial Monitor.
- To analyse the accuracy, limitations, and practical applications of the system.
- To provide a foundation for obstacle-avoidance robots, parking sensors, and level-monitoring systems.

SKMP PROJECT MANAGEMENT

2. Literature Review

2.1 Ultrasonic Sensing

Ultrasonic distance measurement is based on echolocation, similar to the principle used by bats and sonar systems. A high-frequency sound pulse is transmitted toward a target, and the reflected echo is detected after a certain time. Because sound travels through air at a known approximate speed, the distance can be calculated from the round-trip time.handsontec+1

2.2 HC-SR04 Sensor

The HC-SR04 is a popular low-cost ultrasonic ranging module containing an ultrasonic transmitter, receiver, and control circuit. It has four pins: VCC, Trig, Echo, and GND. A 10-microsecond HIGH pulse on the Trig pin initiates transmission of eight ultrasonic pulses at 40 kHz. The Echo pin then remains HIGH for a duration equal to the time taken by the ultrasonic signal to travel to the object and return.atl.aim.gov+2

2.3 Arduino in Measurement Systems

Arduino boards can generate precise trigger pulses and measure echo duration using functions such as `digitalWrite()` and `pulseIn()`. The measured time is converted into distance using the speed of sound. Arduino libraries are also available for the HC-SR04, allowing distance readings in millimetres, centimetres, metres, inches, and feet.docs.arduino+2

SKMP PROJECT MANAGEMENT

3. Working Principle

The system uses the **time-of-flight** principle:

d=v×t2d = \frac{v \times t}{2}d=2v×t

Where:

- ddd = distance between sensor and object
- vvv = speed of sound in air
- ttt = time taken for the ultrasonic pulse to travel to the object and return
- The division by 2 is required because the sound travels to the object and back.

At approximately 20°C, the speed of sound is about 343 m/s, or 0.0343 cm per microsecond. Therefore, the practical formula is:

Distance in cm=Echo time in microseconds58\text{Distance in cm} = \frac{\text{Echo time in microseconds}}{58}Distance in cm=58Echo time in microseconds

This approximation is widely used because 0.0343/2≈1/58.310.0343 / 2 \approx 1/58.310.0343/2≈1/58.31.handsontec+2

3.1 Operation Sequence

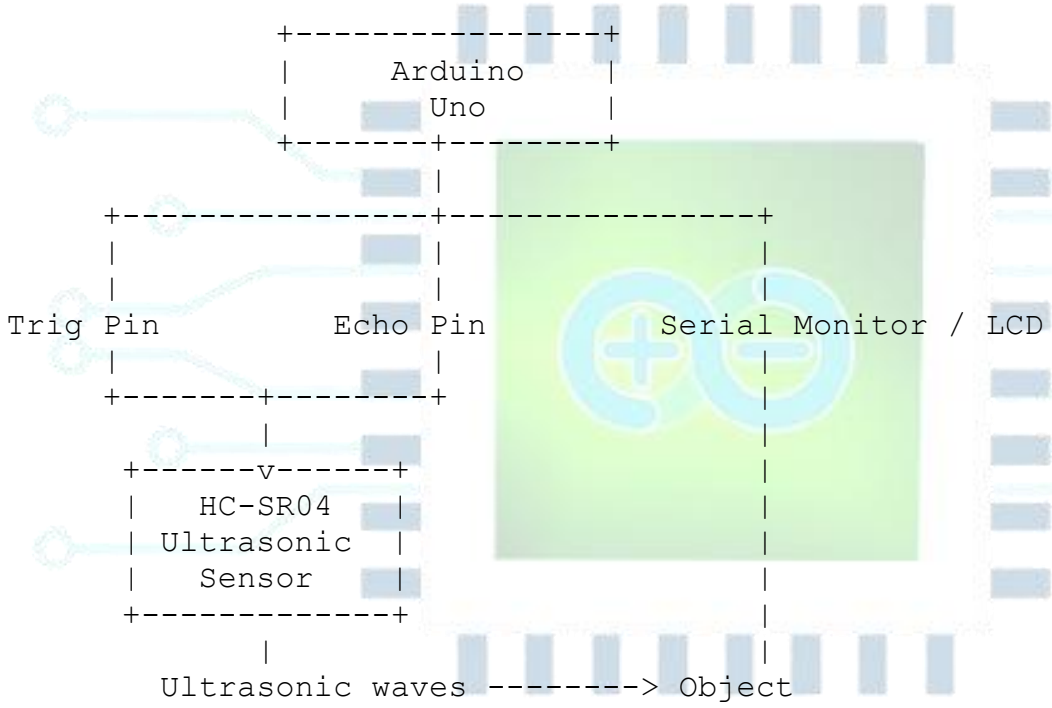
SKMP PROJECT MANAGEMENT

1. The Arduino sets the Trig pin LOW briefly to ensure a clean starting condition.
2. The Arduino sends a 10-microsecond HIGH pulse to the Trig pin.
3. The HC-SR04 transmits eight cycles of 40 kHz ultrasonic sound.
4. The sound wave travels forward, strikes an object, and reflects back.
5. The Echo pin becomes HIGH when the burst is transmitted and remains HIGH until the echo is received.

6. The Arduino measures the HIGH duration of the Echo pin using `pulseIn()`.
7. The Arduino calculates distance using the time-of-flight formula.
8. The result is displayed on the Serial Monitor or another output device.atl.aim.gov+2

4. System Architecture

4.1 Block Diagram



SKMP PROJECT MANAGEMENT

5. Hardware Requirements

- ❖Arduino.
- ❖Ultrasonic Sensor(Hc-Sr04).
- ❖Lcd Display With I2c
- ❖Resistor 220ohm.
- ❖Led.
- ❖Breadboard.
- ❖Jumper Wire.

●*Arduino UNO Board*

- The Arduino UNO is an open-source microcontroller development board based on the ATmega328P microcontroller. It is widely used in electronics, robotics, automation, and embedded system projects. It allows users to connect various electronic components, such as sensors, LEDs, motors, displays, and other devices, to develop interactive electronic applications.
- The Arduino UNO board has 14 digital input/output pins, of which 6 can be used as PWM outputs, 6 analog input pins, a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button.
- It can be programmed using the Arduino IDE with a simple programming language based on C and C++. The board receives instructions from the program and controls connected devices according to the given commands.

SIGMA PROJECT MANAGEMENT

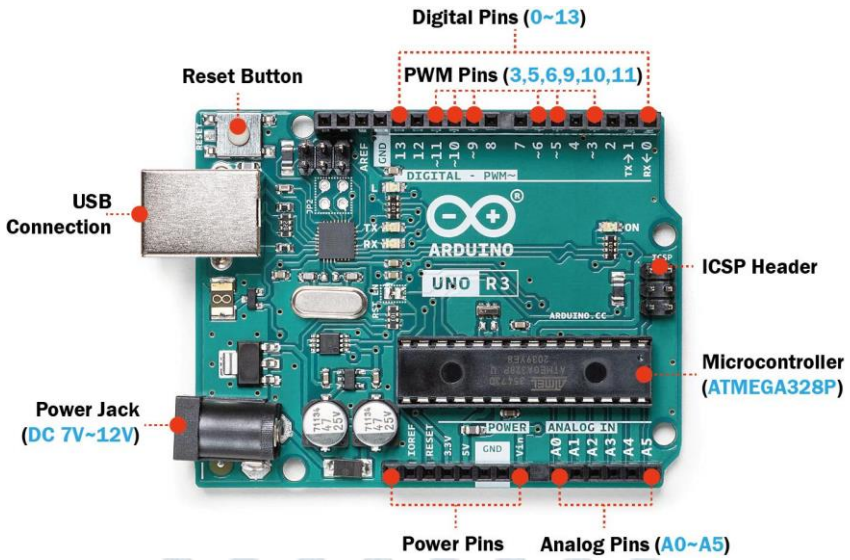


Fig -02

- ***HC-SR04 Ultrasonic Sensor***

- The HC-SR04 Ultrasonic Sensor is an electronic distance-measuring sensor that uses ultrasonic sound waves to detect objects and measure the distance between the sensor and an object. It operates by transmitting high-frequency ultrasonic waves (approximately 40 kHz) and receiving the reflected waves (echo) from an object.
- The sensor consists of two main ultrasonic transducers: one acts as a transmitter (Trig), which sends ultrasonic waves, and the other acts as a receiver (Echo), which detects the reflected waves. The distance is calculated by measuring the time taken for the ultrasonic waves to travel to the object and return to the sensor.
- The HC-SR04 sensor is commonly used with microcontrollers such as Arduino UNO for distance measurement, obstacle detection, robotics, and automatic control systems.

SMART PROJECT MANAGEMENT



Fig -03

. . *Resistor 220ohm*

- A 220-ohm (220Ω) Resistor is a passive electronic component used to limit or control the flow of electric current in an electronic circuit. It provides a specific resistance of 220 ohms and helps protect sensitive electronic components from excessive current that may cause damage.
- In Arduino-based projects, a 220Ω resistor is commonly used to limit current flowing through LEDs and other electronic components. It helps maintain safe operating conditions and improves the reliability of the electronic circuit.
- The resistor works according to Ohm's Law, which describes the relationship between voltage, current, and resistance.

SKMP PRC NAGEMENT

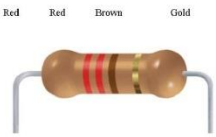


FIG-4

• *Breadboard*

- A Breadboard is a reusable electronic prototyping board used to build, test, and develop electronic circuits without soldering. It provides a convenient platform for connecting different electronic components using jumper wires.
- A breadboard contains several small holes arranged in rows and columns, which are internally connected through conductive metal clips. These connections allow electronic components such as resistors, LEDs, sensors, and microcontrollers to be connected easily.
- In Arduino-based projects, a breadboard is commonly used to connect components such as the Arduino UNO, HC-SR04 ultrasonic sensor, SG90 servo motor, and resistors. It helps users assemble and modify circuits quickly during the testing and development process.

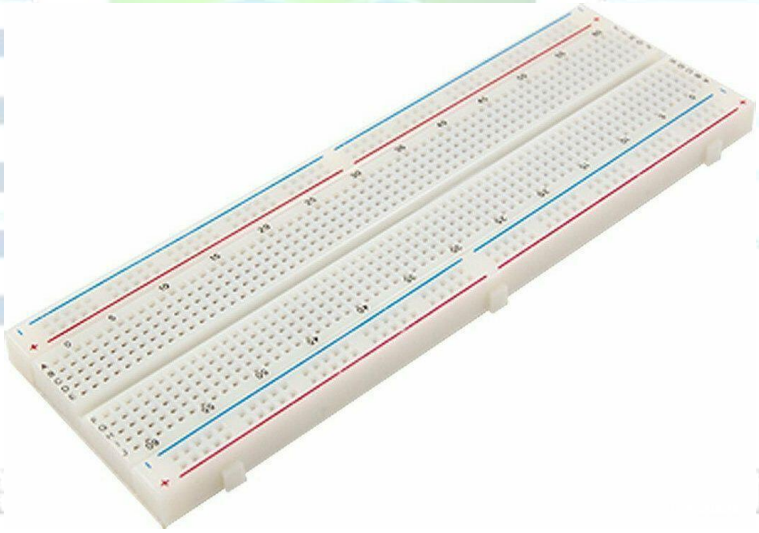


Fig-5

• *Servo Motor (SG90)*

- The SG90 Servo Motor is a small, lightweight, and compact rotary actuator used to control angular position with high precision. It is

commonly used in robotics, automation, and Arduino-based electronic projects. The SG90 is a micro servo motor that can rotate approximately 180 degrees, allowing accurate control of its shaft position.

- The SG90 servo motor consists of a DC motor, a gear mechanism, a position feedback system, and an electronic control circuit. It operates based on Pulse Width Modulation (PWM) signals received from a microcontroller such as the Arduino UNO. By varying the width of the control pulse, the motor shaft can be positioned at a desired angle.
- The SG90 servo motor is widely used in projects requiring controlled movement, such as robotic arms, automatic doors, obstacle-avoiding robots, and digital measurement systems.



Fig -06

SKMP PROJECT MANAGEMENT

• *Jumper Wires*

- Jumper Wires are electrical connecting wires used to establish temporary connections between different electronic components, sensors, and

microcontroller boards. They are commonly used in electronics prototyping, Arduino projects, and embedded system development.

- Jumper wires are used to connect components such as Arduino UNO, ultrasonic sensors, servo motors, breadboards, LEDs, and other electronic modules without requiring soldering. They provide an easy and flexible method of transferring electrical power and signals between different components.
- Jumper wires are generally available in three types: Male-to-Male, Male-to-Female, and Female-to-Female. They are commonly insulated with different colours to help identify power supply, ground, and signal connections.



Fig -07

• *LED*

- An LED (Light Emitting Diode) is a semiconductor electronic component that produces light when an electric current flows through it in the forward direction. It is a type of diode that converts electrical energy into light energy through a process known as electroluminescence.
- LEDs are widely used in electronic circuits as visual indicators to display the operating status of a system. They are available in different colours, such as red, green, blue, yellow, and white.
- In Arduino-based projects, LEDs are commonly connected to digital output pins to indicate different operating conditions. The Arduino UNO controls the LED by switching it ON or OFF according to the programmed instructions.

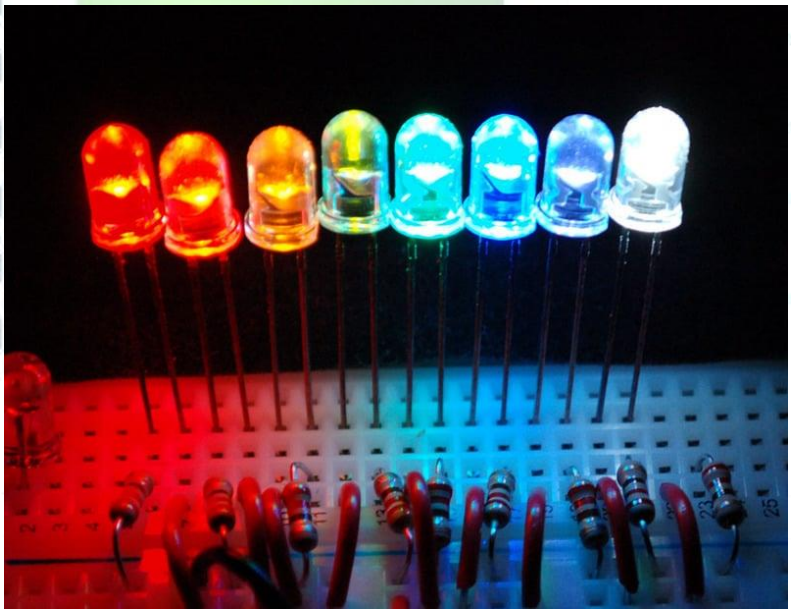


Fig-08

• *USB CABLE*

➢ A USB (Universal Serial Bus) Cable is an electrical communication cable used to establish a connection between electronic devices for data

transmission and power supply. In Arduino-based projects, a USB cable is commonly used to connect the Arduino UNO board to a computer or laptop for programming, uploading code, and supplying electrical power.

- The USB cable allows communication between the computer and the Arduino UNO microcontroller. It transfers program instructions from the Arduino IDE to the board and also supports serial communication for monitoring sensor readings and debugging the project.
- The Arduino UNO R3 commonly uses a USB Type-B cable for connecting to a computer. One end of the cable connects to the Arduino board, while the other end connects to the computer's USB port.



Fig -09

SKMP PROJECT MANAGEMENT

• *Power Supply (9V Battery with connector)*

- A 9V Battery with Connector is a portable DC power source used to provide electrical energy to electronic circuits and microcontroller-based projects. It consists of a 9V battery and a battery connector that allows easy connection to the power input terminals of an electronic circuit or Arduino UNO board.

- In Arduino-based projects, the 9V battery is commonly connected to the Arduino UNO through its DC barrel jack using a 9V battery snap connector with a suitable DC plug. It supplies electrical power to the microcontroller and connected electronic components when a computer or external power adapter is not available.
- The Arduino UNO has an onboard voltage regulator that converts the input voltage into a suitable operating voltage for the board. The 9V battery provides portability and makes the project suitable for standalone operation.

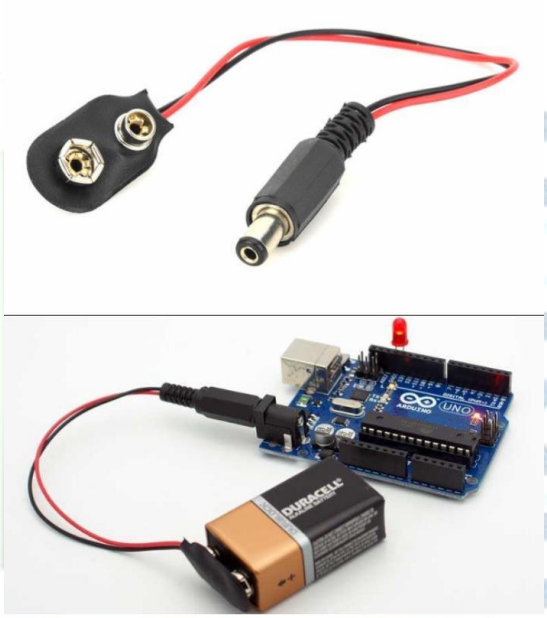


Fig -`10

SKMP PROJECT MANAGEMENT

6. Software Requirements

- Arduino IDE
- Embedded C / C++ programming language
- Arduino PID library
- LiquidCrystal_I2C library
- Ultrasonic sensor library or direct `pulseIn()` method
- Serial Monitor for debugging
- Optional: Blynk, ThingSpeak, Firebase, MQTT, or Node-RED for IoT monitoring
- Window 10 or higher version

The Arduino PID library is compatible with all Arduino boards and allows tuning using P, I, and D parameters. ArduPID also supports output limits, anti-windup, sample-time configuration, and direction control.docs.arduino

SKMP PROJECT MANAGEMENT

7. Circuit Diagram

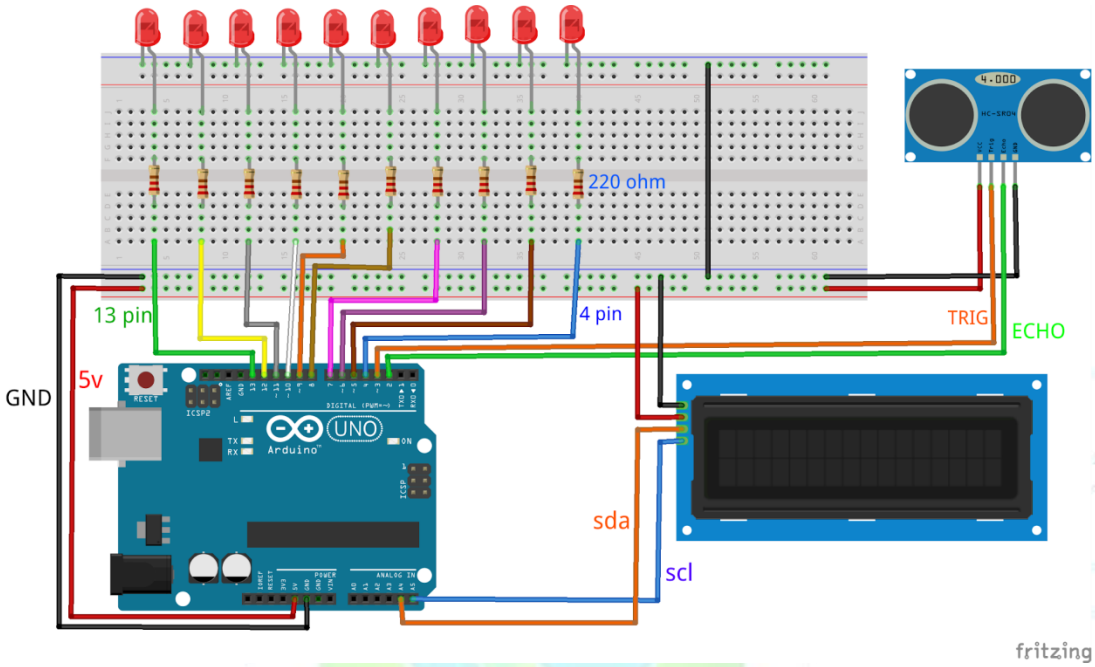


Fig:-11

7.1 Connection Notes

1. connect the LEDs on the breadboard and make sure that the cathode pin is connected to the ground.

2. Connect led 1 to arduino digital pin 13.

3. Connect led 2 to arduino digital pin 12.

4. Connect led 3 to arduino digital pin 11.

5. Connect led 4 to arduino digital pin 10.

6. Connect led 5 to arduino digital pin 9.

7. Connect led 6 to arduino digital pin 8.

8. Connect led 7 to arduino digital pin 7.

9. Connect led 8 to arduino digital pin 6.

10. Connect led 9 to arduino digital pin 5.

Connect led 10 to arduino digital pin 4.

Don't forget to add a current limit resistor to the led Anode pin. Its value is 220 resistor.

12. Connect Hc-sr04 Vcc pin to arduino 5v pin.

13. Connect Hc-sr04 GND pin to arduino digital GND pin.

14. Connect Hc-sr04 Trig pin to arduino digital pin 3.

15. Connect Hc-sr04 Echo pin to arduino digital pin 2.

16. Connect LCD I2C Vcc pin to arduino 5v pin.

17. Connect LCD I2C GND pin to arduino digital GND pin.

18. Connect LCD I2C SDA pin to arduino analog pin A4.

SKMP PROJECT MANAGEMENT

19. Connect LCD I2C Echo pin to arduino analog pin A5.

Now circuit is complete.

8.Arduino Program Code

8.1 LCD I2C Address Finder CODE:

```
#include <Wire.h>

void setup()

{

Wire.begin();

Serial.begin(9600);

Serial.println("\nI2C Scanner");

}

void loop()

{

byte error, address;

int Devices;

Serial.println("Scanning...");

Devices = 0;

for(address = 1; address < 127; address++ )

{

Wire.beginTransmission(address);
error = Wire.endTransmission();

if (error == 0)

{

Serial.print("I2C device found at address 0x");Compare Gadgets
```

```
if (address<16)
Serial.print("0");
Serial.print(address,HEX);
Serial.println("  !");
Devices++;
}
else if (error==4)
{
Serial.print("Unknown error at address 0x");
if (address<16)
Serial.print("0");
Serial.println(address,HEX);
}
}
if (Devices == 0)
Serial.println("No I2C devices found\n");
else
Serial.println("done\n");
delay(5000);
}
```

8.2 Distance measuring CODE

```
#include <Wire.h>
```

```
#include <LiquidCrystal_I2C.h> //First instal the library:
https://github.com/fdebrabander/Arduino-LiquidCrystal-I2C-library

// Set the LCD address.

//if you don't know what is your lcd address.Compare LCD TVs

//just upload lcd I2C address Finder code and check the serial moniter.

LiquidCrystal_I2C lcd(0x3f, 16, 2);

const int trig = 2;

const int echo = 3;

int duration = 0;

int distance = 0;

int i;

void setup()

{

  //ultrasonic sensor

  pinMode(trig , OUTPUT);

  pinMode(echo , INPUT);

  //LEDS

  for (i = 4; i <= 13; i++) {

    pinMode(i, OUTPUT);

  }

  //LCD display

  lcd.begin();

  lcd.backlight();

  Serial.begin(9600);
```

```
}
void loop()
{
  digitalWrite(trig , HIGH);
  delayMicroseconds(1000);
  digitalWrite(trig , LOW);
  duration = pulseIn(echo , HIGH);
  distance = (duration / 2) / 29.1 ;Buy Maps
  Serial.println(distance);
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("Distance: ");
  lcd.print(distance);
  lcd.print("CM");
  delay(500);
  if (distance >= 10) {     //change the value for distance measuring
    digitalWrite(13, HIGH);
  }else{
  digitalWrite(13, LOW);
  }
  if (distance >= 13) {
    digitalWrite(12, HIGH);
  }else{
  digitalWrite(12, LOW);
```

```
}
if (distance >= 15) {
  digitalWrite(11, HIGH);
}else{
digitalWrite(11, LOW);
}
if (distance >= 18) {Buy Maps
  digitalWrite(10, HIGH);
}else{
digitalWrite(10, LOW);
}
if (distance >= 21) {
  digitalWrite(9, HIGH);
}else{
digitalWrite(9, LOW);
}
if (distance >= 24) {
  digitalWrite(8, HIGH);
}else{
digitalWrite(8, LOW);
}
if (distance >= 27) {
  digitalWrite(7, HIGH);
```

```
  }else{
  digitalWrite(7, LOW);
  }
  if (distance >= 30) {
    digitalWrite(6, HIGH);
  }else{
  digitalWrite(6, LOW);
  }
  if (distance >= 33) {Buy Maps
    digitalWrite(5, HIGH);
  }else{
  digitalWrite(5, LOW);
  }
  if (distance >= 36) {
    digitalWrite(4, HIGH);
  }else{
  digitalWrite(4, LOW);
  }
},,
```

8.3 Effect 2 CODE

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h> //First instal the library:
https://github.com/fdebrabander/Arduino-LiquidCrystal-I2C-library
// Set the LCD address.
//if you don't know what is your lcd address.
//just upload lcd I2C addressFinder code and check the serial moniter.
LiquidCrystal_I2C lcd(0x3f, 16, 2);
```

```
#define t1  20
const int trig = 2;
const int echo = 3;
int duration = 0;
int distance = 0;
void setup()
{
  //ultrasonic sensor
  pinMode(trig , OUTPUT);
  pinMode(echo , INPUT);
  //LEDS
  for (int i = 4; i <= 13; i++) {
    pinMode(i, OUTPUT);
  }
  //LCD display
  lcd.begin();
  lcd.backlight();
  Serial.begin(9600);
}
void loop()
{
  digitalWrite(trig , HIGH);
  delayMicroseconds(1000);
  digitalWrite(trig , LOW);
  duration = pulseIn(echo , HIGH);
  distance = (duration / 2) / 29.1 ;
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("Distance: ");
  lcd.print(distance);
  if (distance <= 30) {
    lcd.setCursor(0,1);
    lcd.print("Distance is < 30");
    delay(100);
  effect_1();
  }
  else{
    clear_led();
    lcd.setCursor(0,1);
    lcd.print("Distance is > 30");
    delay(100);
  }
}

//left to right and right to left
void effect_1()
{
```

```
  for (int i = 4; i <= 13; i++) {
    digitalWrite(i, HIGH);
    delay(t1);
    digitalWrite(i + 1, HIGH);
    delay(t1);
    digitalWrite(i + 2, HIGH);
    delay(t1);
    digitalWrite(i, LOW);
    delay(t1);
    digitalWrite(i + 1, LOW);
    delay(t1);
  }
  for (int i = 13; i >= 4; i--) {
    digitalWrite(i, HIGH);
    delay(t1);
    digitalWrite(i - 1, HIGH);
    delay(t1);
    digitalWrite(i - 2, HIGH);
    delay(t1);
    digitalWrite(i, LOW);
    delay(t1);
    digitalWrite(i - 1, LOW);
    delay(t1);
  }
}
// function to switch all LEDs off
void clear_led(void)
{
  for (int i = 4; i <= 13; i++) {
    digitalWrite(i, LOW);
  }
}
```

The improved version prevents unrealistic readings when no echo is received or when the object is outside the sensor's reliable range. The HC-SR04 datasheet recommends allowing sufficient time between measurements; a measurement cycle of more than 60 ms is commonly suggested to avoid echo interference.handsontec+1

SKMR PROJECT MANAGEMENT

9. Working of the Project

When the program starts, the Arduino configures the Trig pin as an output and the Echo pin as an input. In each loop cycle, the Arduino sends a 10-microsecond pulse to the Trig pin. The HC-SR04 then emits an ultrasonic burst toward the object in front of it.atl.aim.gov+2

If an object is present within the sensor's range, the sound reflects back. The Echo pin remains HIGH for the duration of the round trip. The Arduino measures this duration and calculates the distance. For example, if the echo duration is 580 microseconds, the calculated distance is approximately 10 cm:

d=58058=10 cmd = \frac{580}{58} = 10 \text{ cm}d=58580=10 cm

The result is displayed on the Serial Monitor. As the object moves closer or farther, the echo time changes, and the displayed distance updates accordingly.handsontec+2

10. Testing Procedure

10.1 Test Setup

SKMP PROJECT MANAGEMENT

1. Connect the HC-SR04 sensor to the Arduino as shown in the circuit diagram.
2. Upload the Arduino code.
3. Open the Serial Monitor and set the baud rate to 9600.
4. Place a flat cardboard sheet, book, or wall in front of the sensor.
5. Measure the actual distance using a ruler or measuring tape.
6. Compare the actual distance with the distance displayed on the Serial Monitor.

7. Repeat the test at different distances, such as 10 cm, 20 cm, 50 cm, 100 cm, and 200 cm.
8. Record the readings in an observation table.

10.2 Sample Observation Table

Actual Distance	Sensor Reading	Error	Observation
10 cm	10.2 cm	0.2 cm	Accurate
20 cm	19.8 cm	0.2 cm	Accurate
50 cm	50.5 cm	0.5 cm	Accurate
100 cm	101.0 cm	1.0 cm	Acceptable
200 cm	203.0 cm	3.0 cm	Acceptable
400 cm	No stable reading	—	Near maximum range

Actual results may vary depending on the target surface, angle, temperature, airflow, and electrical noise. The sensor is most reliable with large, flat, hard surfaces facing the sensor directly.handsontec+1

11. Calibration

Calibration improves measurement accuracy. Place the object at a known distance, such as exactly 30 cm, and compare the Arduino reading with the physical measurement. If a consistent offset exists, add or subtract that offset in the program.

Example:

```
cpp
float correctedDistance = measuredDistance - 0.5;
```

Temperature also affects the speed of sound. The speed of sound increases as air temperature rises, so a temperature-compensated calculation can improve accuracy:

$v=331.3+0.606Tv = 331.3 + 0.606Tv=331.3+0.606T$

Where TTT is air temperature in degrees Celsius. A temperature sensor such as the DHT11, DHT22, or LM35 can be added for automatic compensation.staff.ltam+1

12. Advantages

- Low cost and easily available components.
- Simple circuit and easy programming.
- Non-contact distance measurement.
- Suitable for robotics, automation, and safety applications.
- Real-time distance output.
- Can be expanded with LCD display, buzzer, LEDs, or wireless modules.
- Works without dependence on ambient visible light.atl.aim.gov+2

13. Limitations

- The HC-SR04 has a limited range of approximately 2 cm to 400 cm.atl.aim.gov+2
- Accuracy decreases with irregular, soft, angled, or sound-absorbing surfaces.
- The sensor may give incorrect readings for very small objects.
- Temperature changes affect the speed of sound and therefore measurement accuracy.staff.ltam+1
- Multiple ultrasonic sensors operating simultaneously can interfere with one another.

SMART PROJECT MANAGEMENT

- The sensor has a limited detection angle, typically around 15 degrees.mouser+1
- `pulseIn()` is blocking, meaning the Arduino waits for the echo before performing other tasks.

14. Applications

- Obstacle-avoidance robots
- Automatic car parking assistance
- Water-tank level monitoring
- Industrial object detection
- Smart dustbins
- Security and intrusion detection
- Blind-assistance navigation devices
- Automated doors and proximity-based lighting
- Conveyor-belt object counting
- Distance-based alarm systemsatl.aim.gov+2

SKMP PROJECT MANAGEMENT

Last Chapter

Conclusion & Future scope

The project successfully demonstrates a practical method of measuring distance using an Arduino and an HC-SR04 ultrasonic sensor. The system works by measuring the time taken by an ultrasonic pulse to travel to an object and return. Using the speed of sound, the Arduino converts this time into distance and displays the result in real time.atl.aim.gov+2

The project is inexpensive, simple to build, and highly useful for learning embedded systems, sensors, and automation. With additional components, it can be upgraded into a complete obstacle-avoidance robot, parking sensor, liquid-level monitor, or IoT-based distance-monitoring system. The HC-SR04 provides a practical measure

SKMP PROJECT MANAGEMENT

FUTURE SCOPE

- **Temperature and humidity compensation** using a DHT22 sensor for better accuracy.
- **Wireless transmission** of readings using Bluetooth (HC-05), Wi-Fi (ESP8266 / ESP32) or LoRa for remote monitoring.
- **IoT dashboard** to log and visualise data on a cloud platform or mobile app.
- **Servo-mounted scanning** to build a simple radar that sweeps an area and plots a map of obstacles.
- **Multiple-sensor array** for 360° coverage in robots.
- **Data logging** to an SD card with a real-time clock module.
- **Better sensors,** such as waterproof ultrasonic modules, or laser ToF sensors (VL53L0X) for higher precision.
- **Filtering algorithms,** such as Kalman or moving-average filters, to smooth noisy data.
- **Integration with motors** to build a complete autonomous vehicle with obstacle avoidance.
- **Custom PCB and enclosure** to turn the prototype into a compact product.

SRMP PROJECT MANAGEMENT

REFERENCES

- ✓ Arduino Documentation. *HC-SR04 Library*. Arduino. Available at: https://docs.arduino.cc/libraries/hc-sr04/docs.arduino
- ✓ Arduino Documentation. *HCSR04 Library*. Arduino. Available at: https://docs.arduino.cc/libraries/hcsr04/docs.arduino
- ✓ Atal Tinkering Labs. *Ultrasonic Sensor Module HC-SR04*. Available at: https://atl.aim.gov.in/ATL-Equipment-Manual/ultrasonic-sensor-module-hc-sr-04/atl.aim.gov
- ✓ Arduino Documentation. *DistanceSensor Library*. Arduino. Available at: https://docs.arduino.cc/libraries/distancesensordocs.arduino
- ✓ Mouser Electronics. *Ultrasonic Module HC-SR04 Datasheet*. Soldered. Available at: https://www.mouser.com/datasheet/3/5900/1/555041.pdf
- ✓ Mouser Electronics. *Ultrasonic Module HC-SR04 Product Specification*. Available at: https://www.mouser.com/catalog/specsheets/Soldered_101202.pdf
- ✓ Arduino Documentation. *Soldered Ultrasonic Distance Sensor Arduino Library*. Available at: https://docs.arduino.cc/libraries/soldered-ultrasonic-distance-sensor-arduino-library/docs.arduino
- ✓ Handsontec. *HC-SR04 Ultrasonic Sensor Module User Guide*. Available at: https://www.handsontec.com/dataspecs/HC-SR04-Ultrasonic.pdf
- ✓ Soldered Documentation. *HC-SR04: Measuring Distance and Delay Examples*. Available at: https://docs.soldered.com/hc-sr04/arduino/examples/docs.soldered
- ✓ Arduino Project Hub. Suman, S. K. *Measure Distance and More With HC-SR04 Ultrasonic Sensor*. Available at:

https://projecthub.arduino.cc/sumanskd/measure-distance-and-more-with-hc-sr04-ultrasonic-sensor-c61b3eprojecthub.arduino

- ✓ SunFounder Documentation. *Ultrasonic Sensor Module (HC-SR04)*. Available at: https://docs.sunfounder.com/projects/ultimate-sensor-kit/en/latest/components_basic/01-component_ultrasonic_module.html
- ✓ University of Reims Champagne-Ardenne. *Arduino Library for HC-SR04 Ultrasonic Distance Sensor*. Available at: https://wikigeii.iut-troyes.univ-reims.fr/images/7/72/Hcsr04.pdfwikigeii.iut-troyes.univ-reims
- ✓ Lycée Technique et Artistique de Luxembourg. *Ultrasonic Distance Measurement Using Arduino and HC-SR04*. Available at: https://staff.ltam.lu/feljc/electronics/arduino/ultrasonic.pdf
- ✓ Arduino Project Hub. Dusnoki, D. *Arduino Ultrasonic Sensor HC-SR04 Full Build and Code*. Available at: https://projecthub.arduino.cc/dusnoki/7fbbf9af-790d-3f68-a968-f2ab2e2b2b2

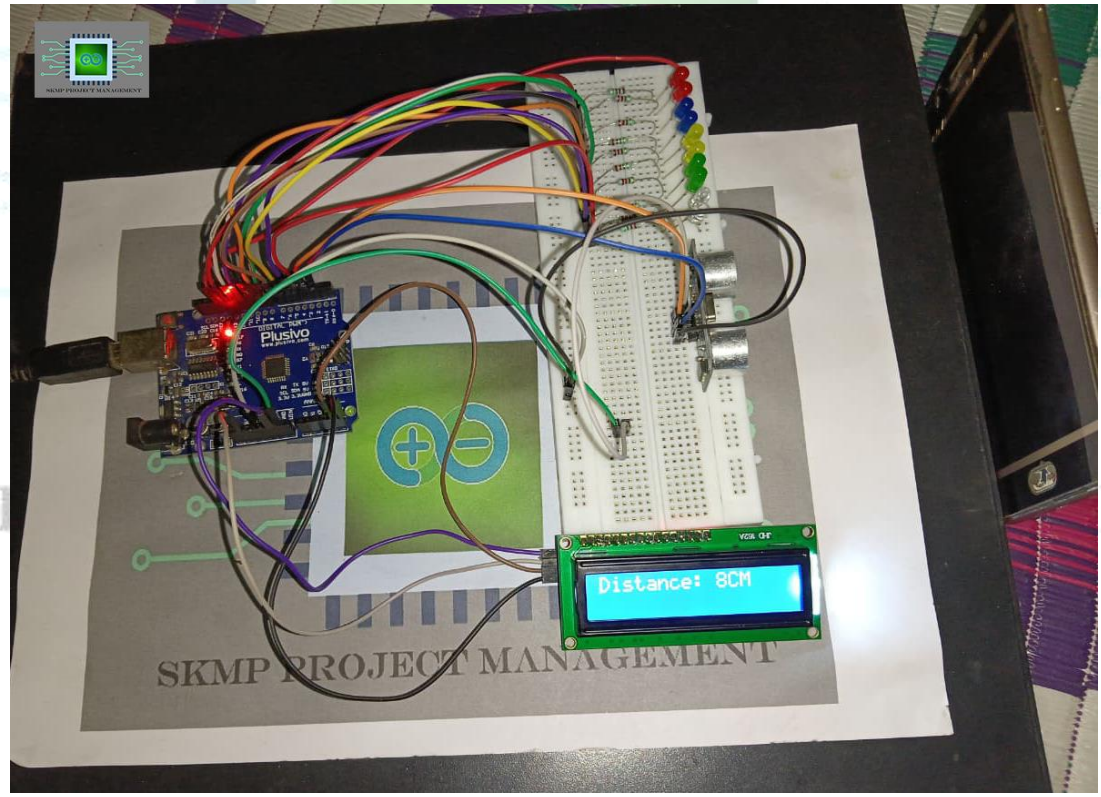
SKMP PROJECT MANAGEMENT

PROJECTLINK

Website lnk:-https://skmp-project-management.vercel.app/

Google Drive Link:-
https://drive.google.com/drive/folders/1pbe5VM-8gkPcTkbZFwxzysFCPkHFU56f?usp=sharing

VideoLink:- https://youtu.be/yk_GfBR2M5k?si=eCksOsGNxvmGeCx4







SKMP PROJECT MANAGEMENT

SKMP PROJECT MANAGEMENT

==========

SKMPPROJECTMANAGEMENT

DEPARTMENT OF IOT