

PROGRAM 1

Board & Environment Setup (ESP8266 - NodeMCU)

IDE

- Arduino IDE

Board Package Configuration (One-Time Setup)

1. Open Arduino IDE
2. Go to **File → Preferences**
3. In **Additional Board Manager URLs**, add:

```
http://arduino.esp8266.com/stable/package_esp8266com_index.json
```

4. Click **OK**
5. Go to **Tools → Board → Boards Manager**
6. Search for **ESP8266**
7. Install **esp8266 by ESP8266 Community**

Board & Port Selection

- **Tools → Board → NodeMCU 1.0 (ESP-12E Module)**
- **Tools → Port → COMx** (select the detected port)

Required Libraries

Install using **Sketch → Include Library → Manage Libraries**: - Adafruit GFX Library - Adafruit SSD1306 Library

Program Title

Distance Measurement using Ultrasonic Sensor with OLED Display and Buzzer using ESP8266

Components Used

- NodeMCU ESP8266
- HC-SR04 Ultrasonic Sensor
- OLED Display (128×64, SSD1306, I2C)
- Buzzer
- Jumper Wires

Pin Configuration

Ultrasonic Sensor (HC-SR04)

Sensor Pin	NodeMCU Pin	GPIO	Description
VCC	Vin	—	Power Supply
GND	GND	—	Ground
TRIG	D5	GPIO14	Trigger pulse from ESP8266
ECHO	D6	GPIO12	Echo signal to ESP8266

OLED Display (I2C)

OLED Pin	NodeMCU Pin	GPIO
SDA	D2	GPIO4
SCL	D1	GPIO5
VCC	3.3V / Vin	—
GND	GND	—

Buzzer

Buzzer Pin	NodeMCU Pin	GPIO
+	D4	GPIO2
-	GND	—

Working Explanation

1. ESP8266 sends a short trigger pulse to the ultrasonic sensor
2. Ultrasonic sensor emits sound waves
3. Waves reflect back from the obstacle
4. Echo pin stays HIGH for the travel duration
5. ESP8266 measures this duration
6. Distance is calculated using speed of sound
7. If distance is below a threshold, buzzer is activated
8. Distance and alert status are displayed on OLED

Code (With Self-Explanatory Comments)

```
#include <Arduino.h>          // Core Arduino functions
#include <Wire.h>              // I2C communication
#include <Adafruit_GFX.h>        // Graphics library
#include <Adafruit_SSD1306.h>    // OLED driver

#define SCREEN_WIDTH 128
#define SCREEN_HEIGHT 64

// Create OLED display object
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, -1);

// Pin definitions
const int trigPin = D5;      // GPIO14
const int echoPin = D6;       // GPIO12
const int buzzer = D4;        // GPIO2

// Constants
const float SPEED_OF_SOUND = 0.0343; // cm per microsecond
const int THRESHOLD = 20;           // Alert distance in cm

void setup() {
    pinMode(trigPin, OUTPUT); // Trigger pin as output
    pinMode(echoPin, INPUT);  // Echo pin as input
    pinMode(buzzer, OUTPUT); // Buzzer pin as output

    Serial.begin(115200);     // Start serial communication

    // Initialize OLED display
    if (!display.begin(SSD1306_SWITCHCAPVCC, 0x3C)) {
        while (true);         // Stop execution if OLED fails
    }

    display.clearDisplay();   // Clear display buffer
}

void loop() {
    // Send trigger pulse
    digitalWrite(trigPin, LOW);
    delayMicroseconds(2);
    digitalWrite(trigPin, HIGH);
    delayMicroseconds(10);
    digitalWrite(trigPin, LOW);

    // Measure echo duration
```

```

long duration = pulseIn(echoPin, HIGH);

// Calculate distance
float distance = (duration * SPEED_OF_SOUND) / 2;

// Display distance
display.clearDisplay();
display.setTextSize(2);
display.setTextColor(SSD1306_WHITE);
display.setCursor(0, 0);
display.print("Dist:");

display.setCursor(0, 30);
display.print(distance);
display.print(" cm");

// Alert condition
if (distance > 0 && distance < THRESHOLD) {
    digitalWrite(buzzer, HIGH);
    display.setCursor(0, 52);
    display.print("ALERT!");
} else {
    digitalWrite(buzzer, LOW);
}

display.display(); // Update OLED
delay(500);        // Stability delay
}

```

Possible Viva / Exam Questions

1. Why is distance divided by 2 in the formula?
2. What is the role of TRIG and ECHO pins?
3. Why is `delayMicroseconds()` used?
4. What happens if no echo is received?
5. Why does OLED require `display.display()`?
6. Difference between D pins and GPIO pins in ESP8266?