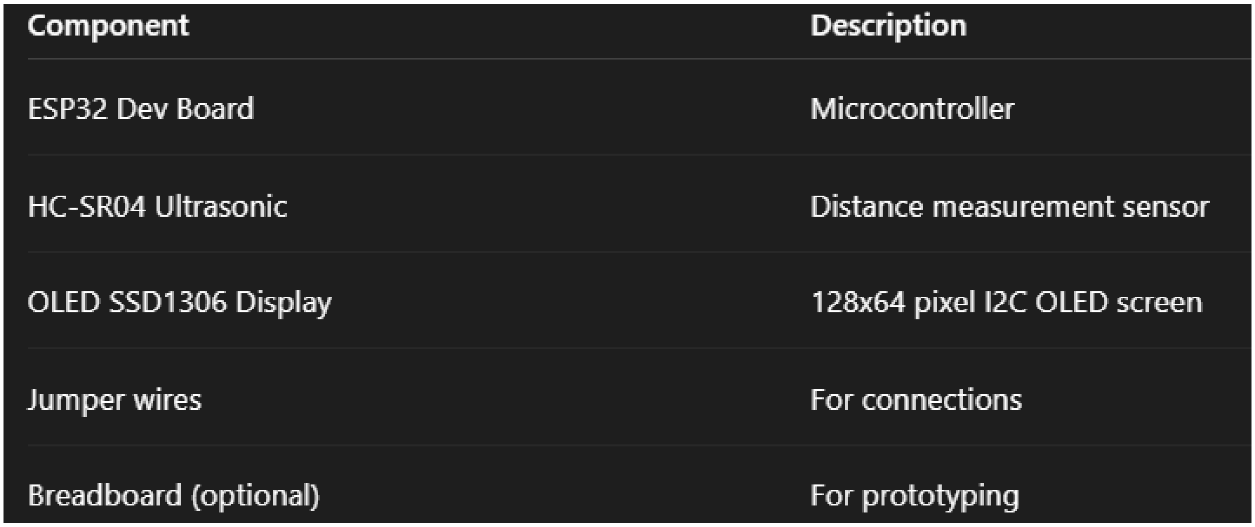
Ultrasonic Distance Sensor with OLED Display using ESP32

📌 1. Objective

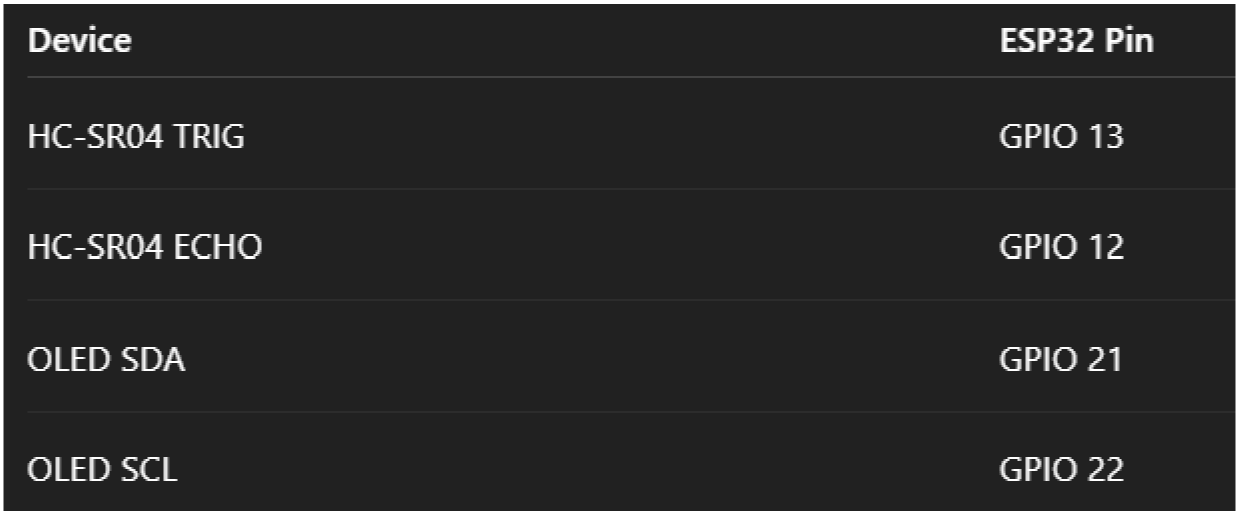
This project uses the HC-SR04 Ultrasonic Sensor to measure distance and displays the result in centimeters and inches on a 0.96" I2C OLED screen.

2. Hardware Components



Component	Description
ESP32 Dev Board	Microcontroller
HC-SR04 Ultrasonic	Distance measurement sensor
OLED SSD1306 Display	128x64 pixel I2C OLED screen
Jumper wires	For connections
Breadboard (optional)	For prototyping

3. Pin Connections



Device	ESP32 Pin
HC-SR04 TRIG	GPIO 13
HC-SR04 ECHO	GPIO 12
OLED SDA	GPIO 21
OLED SCL	GPIO 22

📝 Note: OLED uses I2C, so SDA/SCL pins are fixed on ESP32 unless redefined in code.

4. Line-by-Line Code Explanation

```
#include <Arduino.h>
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>

// HC-SR04 Pins
#define TRIG_PIN 13
#define ECHO_PIN 12

// OLED setup
#define SCREEN_WIDTH 128
#define SCREEN_HEIGHT 64
#define OLED_RESET -1
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, OLED_RESET);

void setup() {
  Serial.begin(115200);

  pinMode(TRIG_PIN, OUTPUT);
  pinMode(ECHO_PIN, INPUT);

  if (!display.begin(SSD1306_SWITCHCAPVCC, 0x3C)) {
    Serial.println("OLED init failed");
    while (1);
  }

  display.clearDisplay();
  display.setTextSize(1);
  display.setTextColor(SSD1306_WHITE);
  display.setCursor(0, 0);
  display.println("Ultrasonic Distance");
  display.display();
  delay(2000);
}

float getDistanceCM() {
  digitalWrite(TRIG_PIN, LOW);
  delayMicroseconds(2);

  digitalWrite(TRIG_PIN, HIGH);
  delayMicroseconds(10);
  digitalWrite(TRIG_PIN, LOW);
  long duration = pulseIn(ECHO_PIN, HIGH);
  float distance = duration * 0.034 / 2; // speed of sound
  return distance;
}

void loop() {
  float distance_cm = getDistanceCM();
  float distance_inch = distance_cm / 2.54;

  Serial.print("Distance: ");
  Serial.print(distance_cm);
  Serial.print(" cm | ");
  Serial.print(distance_inch);
  Serial.println(" inch");

  display.clearDisplay();
  display.setCursor(5, 0);
  display.setTextSize(1);
  display.println("Distance");

  display.setCursor(0, 20);
  display.setTextSize(2);
  display.print("CM:");
  display.print(distance_cm, 2);  // 2 decimal places

  display.setCursor(0, 35);
  display.print("IN:");
  display.print(distance_inch, 2);  // 2 decimal places

  display.display();

  delay(1000);
}
```

Explanation

```
#include <Arduino.h>
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>
```

<Arduino.h>: Core Arduino functions.
<Wire.h>: Handles I2C communication.
<Adafruit_GFX.h>: Graphics library for OLED.
<Adafruit_SSD1306.h>: Controls the SSD1306 OLED screen.

📌 HC-SR04 Pins

```
// HC-SR04 Pins
#define TRIG_PIN 13
#define ECHO_PIN 12
```

TRIG_PIN (GPIO 13): Sends a 10µs pulse to initiate measurement.
ECHO_PIN (GPIO 12): Receives the pulse reflection to calculate distance.

OLED Display Setup

```
// OLED setup
#define SCREEN_WIDTH 128
#define SCREEN_HEIGHT 64
#define OLED_RESET -1
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, OLED_RESET);
```

OLED resolution: 128x64 pixels.
OLED_RESET: Set to -1 (not used).
display: Instantiates the OLED screen.

```
void setup()
```

– Initialization

```
  Serial.begin(115200);
```

Starts the serial monitor at 115200 baud for debugging.

```
  pinMode(TRIG_PIN, OUTPUT);
  pinMode(ECHO_PIN, INPUT);
```

Sets the ultrasonic sensor TRIG as output and ECHO as input.

```
  if (!display.begin(SSD1306_SWITCHCAPVCC, 0x3C)) {
    Serial.println("OLED init failed");
    while (1);
  }
```

Initializes the OLED display at I2C address 0x3C.
If it fails, it shows error and halts.

```
  display.clearDisplay();
  display.setTextSize(1);
  display.setTextColor(SSD1306_WHITE);
  display.setCursor(0, 0);
  display.println("Ultrasonic Distance");
  display.display();
  delay(2000);
```

Shows a startup message on the OLED for 2 seconds.

```
float getDistanceCM()
```

– Distance Calculation

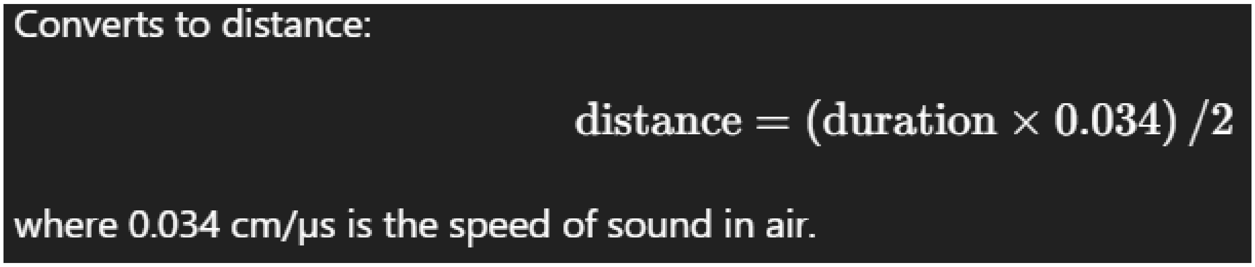
```
float getDistanceCM() {
  digitalWrite(TRIG_PIN, LOW);
  delayMicroseconds(2);

  digitalWrite(TRIG_PIN, HIGH);
  delayMicroseconds(10);
  digitalWrite(TRIG_PIN, LOW);
```

Send a 10µs HIGH pulse on TRIG to start measurement.

```
  long duration = pulseIn(ECHO_PIN, HIGH);
  float distance = duration * 0.034 / 2; // speed of sound
  return distance;
}
```

Measures time the echo pin stays HIGH (in microseconds).



Converts to distance:

$$\text{distance} = (\text{duration} \times 0.034)\,/2$$

where 0.034 cm/µs is the speed of sound in air.

```
void loop()
```

– Continuous Operation

```
  float distance_cm = getDistanceCM();
  float distance_inch = distance_cm / 2.54;
```

Converts distance from cm to inches.

```
  Serial.print("Distance: ");
  Serial.print(distance_cm);
  Serial.print(" cm | ");
  Serial.print(distance_inch);
  Serial.println(" inch");
```

Prints measured values to the serial monitor.

OLED Display Output

```
display.clearDisplay();
display.setCursor(5, 0);
display.setTextSize(1);
display.println("Distance");
```

Clears previous content and writes "Distance" heading.

```
display.setCursor(0, 20);
display.setTextSize(2);
display.print("CM:");
display.print(distance_cm, 2);  // 2 decimal places
```

Prints distance in centimeters with 2 decimal places.

```
display.setCursor(0, 35);
display.print("IN:");
display.print(distance_inch, 2);  // 2 decimal places
```

Prints distance in inches.

```
display.display();
```

Updates the screen with new content.

```
delay(1000);
```

Waits for 1 second before the next reading.

5. Summary of Working

- Ultrasonic pulse is sent via TRIG.
- ECHO receives the reflected signal.
- Time is measured and converted to distance.
- Distance values in cm and inch are:
- Printed on Serial Monitor
- Displayed on OLED screen
- Repeats every 1 second.

6. Testing Tips

- Ensure HC-SR04 has 5V power and GND connected.
- Use a level shifter or voltage divider if your ESP32 cannot tolerate 5V on input pins.
- Confirm OLED's I2C address is 0x3C using an I2C scanner sketch if nothing shows up.