

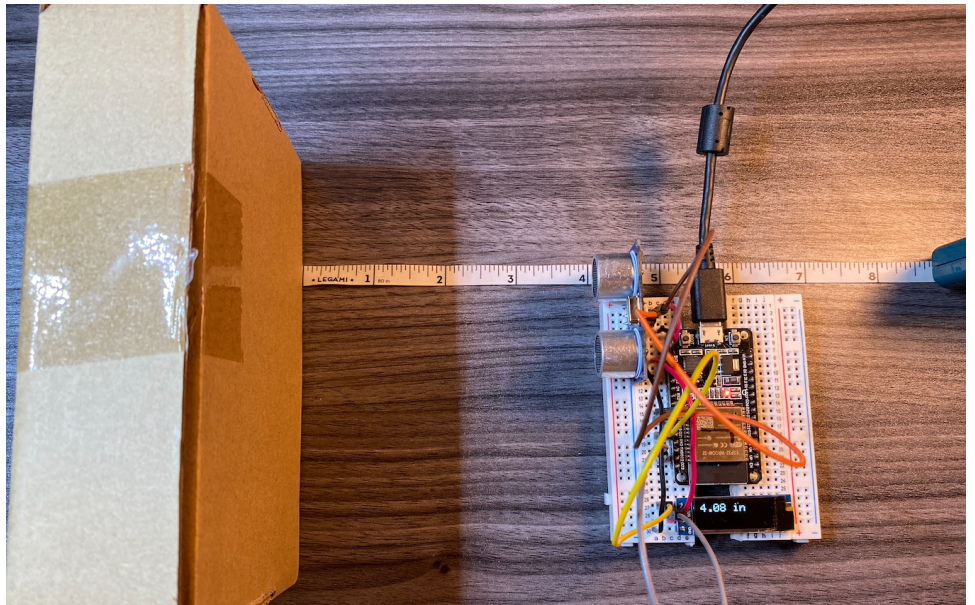
DIY ESP32 Oled Display and Distance Sensor Tutorial

Created by DerTech

In this two step tutorial you'll learn how to wire and program an oled Display and ultrasonic distance sensor to display distance data, its like a electronic measuring tape.

Parts Needed:

- ESP32
- Wires
- SSD1306 Oled Display
- Bread board (I cut mine in half to fit the esp32 better)
- Ultrasonic Distance Sensor



Link for components: [Kit Link](#)

If your new to electronics or microcontrollers heres a tutorial for getting started with the Arduino Ide and Esp32 so you can make projects like this: [Arduino and Esp Tutorial](#)

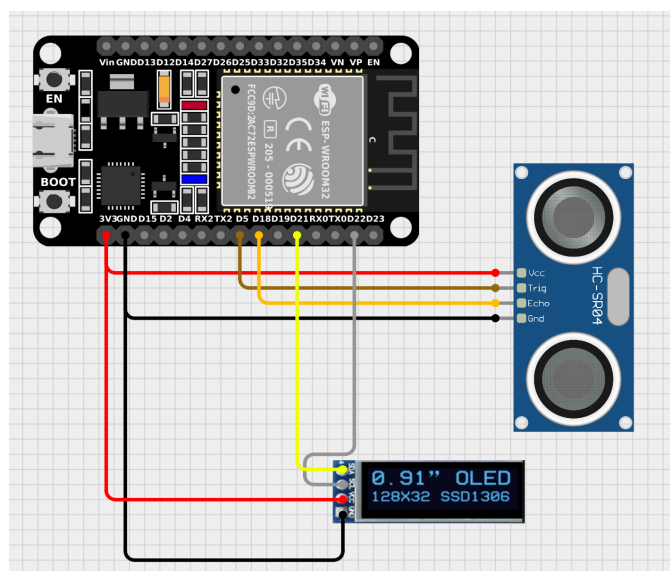
Step 1: Wiring Distance Sensor and Oled Display to the esp32

Distance sensor:

- VCC to 3v3
- Trig to Pin 5
- Echo to Pin18
- Gnd to Gnd

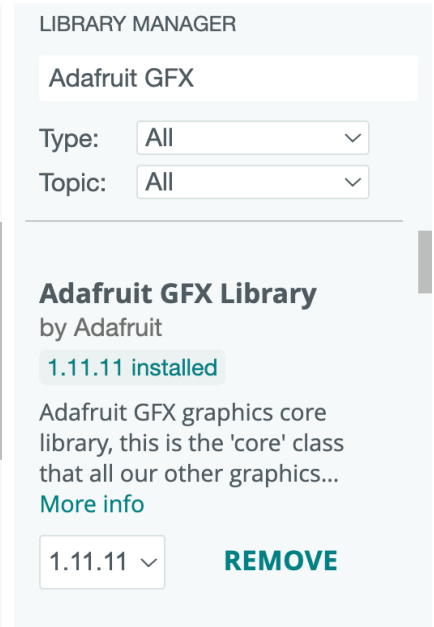
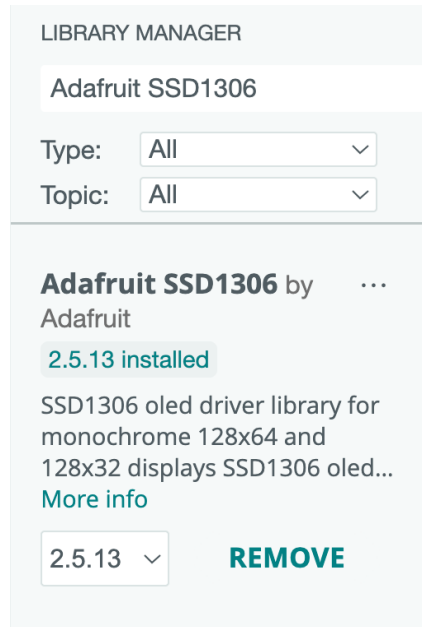
Oled Display:

- SDA to 21
- SCL to 22
- VCC to 3v3
- Gnd to Gnd



Step 2: Programming

Download the following libraries on the arduino IDE library manager to use the SSD1306 Oled Display.



Copy and Paste the following code and upload it to your board.

```
C/C++
#include <Wire.h> // I2C library for communication with the OLED
display
#include <Adafruit_GFX.h> // Graphics library for drawing on the OLED
display
#include <Adafruit_SSD1306.h> // OLED display library

// Create an SSD1306 display object (128x32 resolution)
Adafruit_SSD1306 display(128, 32, &Wire, -1);
const int trigPin = 5;
const int echoPin = 18;
#define SOUND_SPEED 0.034 // Speed of sound in cm per microsecond
#define CM_TO_INCH 0.393701 // Conversion factor from cm to inches
long duration; // Variable to store pulse duration
float distanceCm;
float distanceInch;

void setup() {
  Serial.begin(115200); // Start serial communication for debugging
  pinMode(trigPin, OUTPUT); // Set the trigPin as an OUTPUT
  pinMode(echoPin, INPUT); // Set the echoPin as an INPUT
```

```
display.begin(SSD1306_SWITCHCAPVCC, 0x3C); // Initialize the OLED display
delay(250); // Wait for display initialization

display.clearDisplay(); // Clear the display buffer
display.setTextColor(WHITE); // Set text color to white
display.setTextSize(2); // Set text size to 2
}

void loop() {
    display.clearDisplay(); // Clear the display before updating
    display.setCursor(0, 0); // Set the cursor to the top-left corner

    // Send a pulse to trigger the ultrasonic sensor
    digitalWrite(trigPin, LOW); // Ensure trigPin is LOW to start
    delayMicroseconds(2);
    digitalWrite(trigPin, HIGH); // Generate a 10 microsecond HIGH pulse
    delayMicroseconds(10); // Pulse duration
    digitalWrite(trigPin, LOW); // End pulse

    duration = pulseIn(echoPin, HIGH); // Measure the pulse duration on echoPin
    distanceCm = duration * SOUND_SPEED / 2; // Calculate the distance in cm
    distanceInch = distanceCm * CM_TO_INCH; // Convert to inches

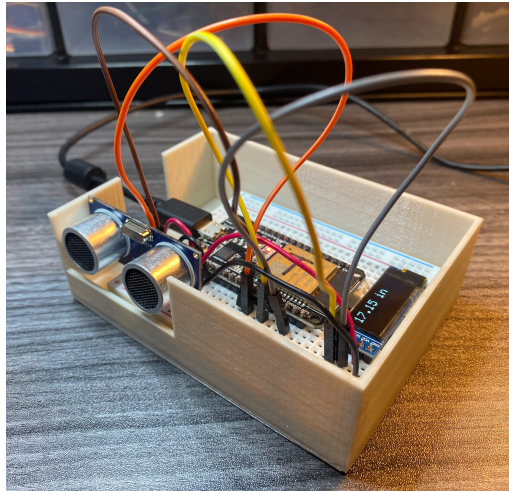
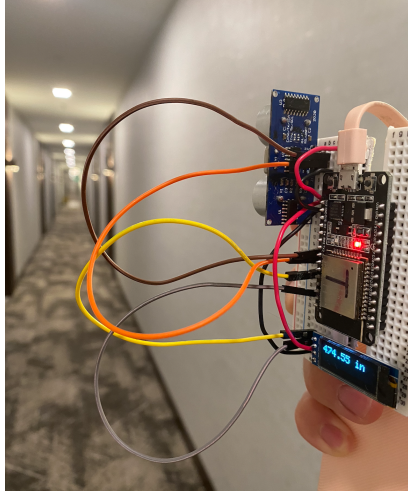
    // Print distance to Serial Monitor for debugging
    Serial.print("Distance (cm): ");
    Serial.println(distanceCm);
    Serial.print("Distance (inch): ");
    Serial.println(distanceInch);

    // Display the distance on the OLED
    // you can change the displayed distance to cm by replacing distanceInch with
    distanceCm and replace the string with "cm"

    display.print(distanceInch); // Display the distance in inches
    display.print(" in");
    display.display(); // Update the OLED display
    delay(250); // Wait before next reading
}
```

Final Notes:

The ESP32 should now be reading the distance data and printing it onto the serial monitor and Oled display. The max distance is about 474 inches as shown on the image. Enjoy your project!



More Tech Tutorials!: [Tutorials Link](#)

Support me on social media!: [Instagram Link](#)

