



## Bibliothèques à installer (dans Arduino IDE)

Avant de commencer, installe ces bibliothèques via le **Gestionnaire de bibliothèques** :

1. [Adafruit GFX Library](#)
2. [Adafruit SSD1306](#)

Explication pas à pas du code :

### 1. Les bibliothèques

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

Ces lignes importent les bibliothèques nécessaires pour gérer l'écran OLED.

Wire.h gère la communication I2C.

Adafruit\_GFX.h gère les graphismes (textes, formes, etc.).

Adafruit\_SSD1306.h gère le contrôleur de l'écran OLED.

### 2. Initialisation de l'écran OLED

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

On crée un objet `display` qui représente l'écran OLED de 128×64 pixels.

Le `-1` signifie que la broche de reset n'est pas utilisée.

### 3. Déclaration des broches pour le capteur

```
const int trigPin = 5;
const int echoPin = 18;
```

Le capteur à ultrasons HC-SR04 a deux broches principales :

trig : déclenche l'émission d'une onde.      echo : reçoit l'onde réfléchie.

Ici, on les relie aux broches 5 et 18 de l'ESP32.

#### 4. La fonction `setup()`

```
Serial.begin(115200);
pinMode(trigPin, OUTPUT);
pinMode(echoPin, INPUT);
```

On configure la communication série et les broches du capteur :

trigPin en sortie (output),  
echoPin en entrée (input).

#### 5. Initialisation de l'écran OLED

```
if (!display.begin(SSD1306_SWITCHCAPVCC, 0x3C)) {
    Serial.println(F("Écran OLED non détecté"));
    while (true);
}
```

On démarre l'écran OLED. L'adresse I2C 0x3C est celle utilisée par la plupart des écrans SSD1306.

#### 6. Mesure de la distance

```
digitalWrite(trigPin, LOW);
delayMicroseconds(2);
digitalWrite(trigPin, HIGH);
delayMicroseconds(10);
digitalWrite(trigPin, LOW);
```

Ceci envoie une impulsion de **10 µs** sur `trig`, pour que le capteur émette une onde ultrasonore.

```
duration = pulseIn(echoPin, HIGH);
distance = duration * 0.034 / 2;
```

pulseIn() mesure le temps en microsecondes pendant lequel `echoPin` reste HIGH (c'est le temps que l'onde met à revenir).

On multiplie par 0.034 cm/µs (vitesse du son)

Puis on divise par 2 (aller-retour)

## 7. Affichage sur l'écran OLED

```
display.clearDisplay();
display.setTextSize(2);
display.setTextColor(SSD1306_WHITE);
display.setCursor(10, 0);
display.print("Distance: ");
display.setCursor(10, 20);
display.print(distance, 1);
display.println(" cm");
display.display();
```

Cette séquence : Efface l'écran,

Affiche le texte "Distance: X.X cm" avec une taille lisible.

## 8. Pause entre deux mesures

```
delay(1000);
```

On attend **1 seconde** entre chaque mesure pour éviter que l'écran ne clignote trop vite.