

Дальномір з OLED

Відстань до предмета показуємо числом і смугою на екрані.

- Arduino Uno, HC-SR04, макетна плата, USB-кабель і дроти.
- OLED 128×64 з контролером SSD1306 та інтерфейсом I²C (чотири контакти).
- Для фізичного OLED із логікою лише 3,3 В — двонаправний узгоджувач рівнів I²C 5↔3,3 В.
- Лінійка та плоский картон.

Перед складанням перевір маркування екрана: SSD1306, 128×64, I²C. Для іншої моделі потрібно змінити бібліотеку або налаштування.

Перед зміною схеми від'єднай живлення. У Wokwi зупини симуляцію. Справжню схему перевір разом з учителем перед підключенням USB.

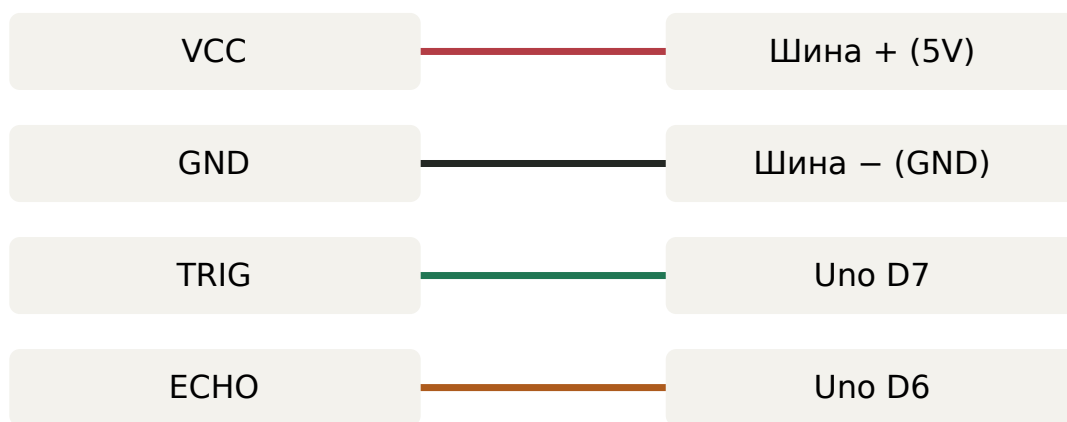
Повторюємо з'єднання HC-SR04

1. Від'єднай живлення. Прочитай написи біля чотирьох контактів HC-SR04.
2. З'єднай **5V** Uno з верхньою шиною **+**, а **GND** Uno — з верхньою шиною **–** макетної плати.
3. Проведи від **VCC** датчика червоний дріт до шини **+**, а від **GND** — чорний до шини **–**.
4. З'єднай **TRIG** з **D7**. Цим виходом Arduino запускає вимірювання.
5. З'єднай **ECHO** з **D6**. На цьому вході Arduino вимірює тривалість відповіді.

Шини **+** і **–** мають з'єднані між собою отвори вздовж плати. Через них підводимо живлення до датчика. На фізичній платі перевір, чи немає розриву посередині шини; за потреби з'єднай її половини перемичкою.

Звіряй саме написи: напрямок роз'єму може відрізнитися від малюнка. Для модуля з ніжками й гнізд Uno зручні дроти «мама-тато».

HC-SR04 → Arduino Uno



Електричні з'єднання · не масштаб плати

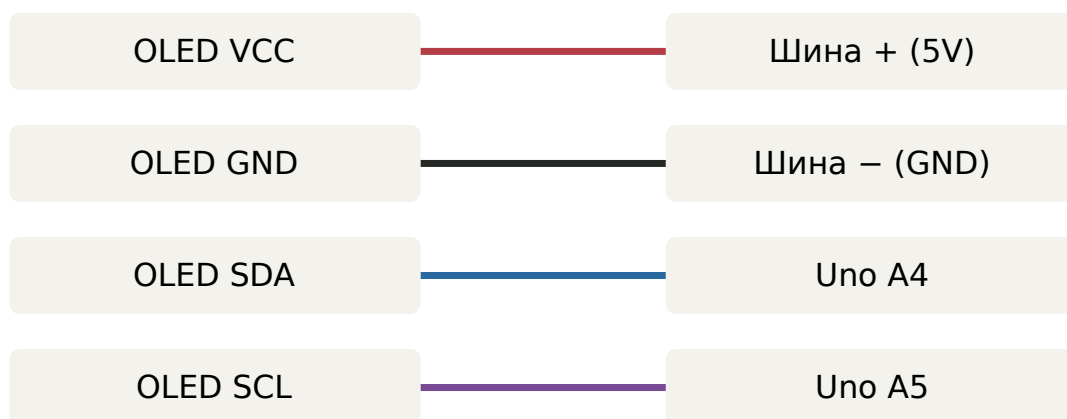
Дві лінії I²C

1. У готовій симуляції OLED **VCC** → верхня шина **+** (**5V**), **GND** → верхня шина **–**. Датчик та екран отримують живлення паралельно від тих самих шин.
2. **SDA** → **A4** Uno: передає дані.
3. **SCL** → **A5** Uno: задає тактові імпульси.

I²C дозволяє передавати команди й малюнок двома сигнальними дротами. Кожен пристрій має адресу; тут це **0x3C**.

Це з'єднання моделі Wokwi. Перед підключенням фізичного OLED виконай наступний крок: не кожен модуль витримує 5 В.

SSD1306 · Wokwi



Електричні з'єднання · не масштаб плати

Живлення та рівні сигналів

Разом з учителем перевір паспорт саме вашого модуля: допустиму напругу живлення і напругу на SDA/SCL. Напис «живлення 3,3–5 В» не завжди означає, що сигнальні входи витримують 5 В.

Для OLED, що працює від 3,3 В і потребує логіки 3,3 В, використовуємо двонапрямний I²C-узгоджувач із підтягувальними резисторами:

1. OLED **VCC** та **LV** узгоджувача → 3.3V Uno; **HV** → **5V** Uno.
2. Усі **GND** → **GND** Uno.
3. **A4** → **HV1**; **LV1** → OLED SDA.
4. **A5** → **HV2**; **LV2** → OLED SCL.

HV1 і LV1 — дві сторони одного каналу, HV2 і LV2 — другого. Не з'єднуй HV з LV напругу. HC-SR04 і далі живиться від **5V**.

Перевір, що споживання екрана з іншими навантаженнями 3.3V не перевищує можливості Uno (50 мА). Якщо модуль підтверджено сумісний із 5 В і за живленням, і за I²C, узгоджувач не потрібен.

Додаємо керування екраном

Бібліотека — готовий набір функцій. Тут `Adafruit SSD1306` керує дисплеєм, а `Adafruit GFX Library` малює текст і фігури.

1. У Wokwi відкрий Library Manager (менеджер бібліотек).
2. Додай `Adafruit SSD1306` та `Adafruit GFX Library`. Залежність `Adafruit BusIO` теж потрібна.
3. Якщо перенесиш ZIP, скопіюй також файл `libraries.txt`.

В Arduino IDE встанови ті самі бібліотеки через Library Manager та погодься на встановлення залежностей.

```
#include <Arduino.h> // Основні команди Arduino.
#include <Wire.h> // Обмін даними I2C.
#include <Adafruit_GFX.h> // Текст і графічні фігури.
#include <Adafruit_SSD1306.h> // Керування дисплеєм SSD1306.

const byte TRIG_PIN = 7; // Команда датчику почати вимірювання.
const byte ECHO_PIN = 6; // Вхід тривалості відлуння.
const byte OLED_ADDRESS = 0x3C; // Адреса дисплея на шині I2C.
Adafruit_SSD1306 display(128, 64, &Wire, -1); // Екран 128×64 без окремого
RESET.
```

Перевіряємо адресу екрана

`Wire.begin()` запускає I²C. Перша перевірка надсилає запит на адресу `0x3C`. Ненульовий результат означає, що пристрій не відповів.

`display.begin()` готує екран і пам'ять для зображення. Якщо адреса недоступна або підготовка не вдалася, програма друкує повідомлення в монітор порту та зупиняє подальшу роботу.

```
Wire.begin(); // На Uno: SDA=A4, SCL=A5.
Wire.beginTransmission(OLED_ADDRESS); // Перевіряємо відповідь за адресою.
byte error = Wire.endTransmission(); // Нуль означає, що пристрій відповів.

if (error != 0 || !display.begin(SSD1306_SWITCHCAPVCC, OLED_ADDRESS)) { //
Немає екрана або пам'яті.
    Serial.println(F("OLED error: check power, A4/A5, address")); // Підказка
в моніторі.

    while (true) { // Зупиняємо приклад до виправлення й перезапуску.
        delay(1000); // Не продовжуємо малювати на недоступному екрані.
    }
}
```

Адреса `0x3D` зустрічається на інших модулях. Змінюй `OLED_ADDRESS` лише після перевірки документації або адреси I²C-сканером.

Очищаємо, малюємо, показуємо

1. `readDistanceCm()` вимірює відстань так само, як у парктроніку.
2. `clearDisplay()` очищає попередній кадр у пам'яті.
3. `setCursor` задає місце тексту, `setTextSize` — розмір.
4. `print` пише число; `drawRect` малює рамку, `fillRect` — заповнення.
5. `display()` передає готовий кадр на екран.

Початок координат — у верхньому лівому куті. X зростає праворуч, Y — униз. Екран має 128 стовпчиків і 64 рядки пікселів.

Смуга показує діапазон 0–100 см: понад 100 вона повна, але число продовжує показувати відстань. Якщо відлуння немає, бачимо `No echo`. Написи латиницею, бо стандартний шрифт бібліотеки не містить українських літер.

Число та смуга

Запуск у Wokwi

Відкрий новий проєкт Arduino Uno. Скопіюй повний код із блока «Код і файли» у `sketch.ino`, а вміст завантаженої схеми — у `diagram.json`. Натисни кнопку запуску.

Адреса нового проєкту: <https://wokwi.com/projects/new/arduino-uno>. Схему можна зібрати вручну за кроками уроку. Повний код — на останній сторінці.

У Wokwi натисни на HC-SR04 та змінюй відстань повзунком. На справжній платі тримай перед датчиком плаский картон на відомій відстані, перевіряючи її лінійкою.

Не притискай предмет до датчика: ближче 2 см показ ненадійний. М'яка тканина, нахилена поверхня або порожній простір можуть не дати відлуння.

1. Перевір, що бібліотеки встановлені.
2. Установи 20 см: число близько 20, смуга приблизно на п'яту частину.
3. Установи 50 см: смуга наполовину.
4. Установи 120 см: число 120, смуга повна.

На справжній схемі забори картон і спрямуй датчик так, щоб не було відлуння: очікуй No echo. Порожній простір інколи все одно дає відбиття від стіни.

Чорний екран чи неправильне число?

Екран чорний

Відкрий монітор порту на 115200. Перевір **VCC**, **GND**, SDA/**A4**, SCL/**A5**, напруги й адресу. Повідомлення про OLED означає невдалий запуск; після виправлення перезапусти плату.

Число правильне, а смуга вже повна

Це нормально понад 100 см: constrain обмежує тільки довжину смуги. Сам показ сантиметрів не обрізається.

Но echo замість відстані

Екран працює. Перевір HC-SR04, **D7** / **D6** і пласку мішень на 20–50 см.

Помилка компіляції про Adafruit

Перевір наявність SSD1306, GFX та BusIO у менеджері бібліотек; назва заголовка має збігатися з кодом.

Смуга до двох метрів

Зміни шкалу смуги з 0–100 на 0–200 см, не змінюючи числовий показ.

1. Знайди рядок, який обчислює `width`.
2. Зміни верхню межу і в `constrain`, і у вхідному діапазоні `map`.
3. Перевір 100 см і 200 см.

Відповідь після спроби

`map(constrain((int)cm, 0, 200), 0, 200, 0, 124)`. На 100 см смуга наполовину, на 200 см повна. 124 — ширина всередині рамки, її не змінюємо.

Дальномір з OLED

Arduino Uno показує відстань на OLED SSD1306 128×64.

```
#include <Arduino.h> // Основні команди Arduino.
#include <Wire.h> // Обмін даними I²C.
#include <Adafruit_GFX.h> // Текст і графічні фігури.
#include <Adafruit_SSD1306.h> // Керування дисплеєм SSD1306.

const byte TRIG_PIN = 7; // Команда датчику почати вимірювання.
const byte ECHO_PIN = 6; // Вхід тривалості відлуння.
const byte OLED_ADDRESS = 0x3C; // Адреса дисплея на шині I²C.
Adafruit_SSD1306 display(128, 64, &Wire, -1); // Екран 128×64 без окремого RESET.

float readDistanceCm() { // Повертаємо сантиметри або -1 при помилці.
    digitalWrite(TRIG_PIN, LOW); // Початковий низький рівень.
    delayMicroseconds(2); // Даємо сигналу стабілізуватися.
    digitalWrite(TRIG_PIN, HIGH); // Починаємо запуск вимірювання.
    delayMicroseconds(10); // Тримаємо імпульс 10 мікросекунд.
    digitalWrite(TRIG_PIN, LOW); // Завершуємо імпульс.
    unsigned long duration = pulseIn(ECHO_PIN, HIGH, 25000UL); // Чекаємо не довше 25 мс.
    float cm = duration / 58.0; // Час шляху звуку туди й назад переводимо в см.

    if (duration == 0 || cm < 2 || cm > 400) { // Немає надійного вимірювання.
        return -1; // Окреме значення помилки, а не нуль сантиметрів.
    }

    return cm; // Віддаємо виміряну відстань.
}

void setup() { // Підготовка датчика та екрана.
    pinMode(TRIG_PIN, OUTPUT); // TRIG надсилає імпульс.
    pinMode(ECHO_PIN, INPUT); // ECHO приймає імпульс.
    digitalWrite(TRIG_PIN, LOW); // До першого вимірювання тримаємо LOW.
    Serial.begin(115200); // Монітор порту для перевірки.
    Wire.begin(); // На Uno: SDA=A4, SCL=A5.
    Wire.beginTransmission(OLED_ADDRESS); // Перевіряємо відповідь за адресою.
    byte error = Wire.endTransmission(); // Нуль означає, що пристрій відповів.

    if (error != 0 || !display.begin(SSD1306_SWITCHCAPVCC, OLED_ADDRESS)) { // Немає екрана або пам'яті.
        Serial.println(F("OLED error: check power, A4/A5, address")); // Підказка в моніторі.

        while (true) { // Зупиняємо приклад до виправлення й перезавпуску.
            delay(1000); // Не продовжуємо малювати на недоступному екрані.
        }
    }
}
```

```

display.clearDisplay(); // Очищаємо графічний буфер.
display.setTextColor(SSD1306_WHITE); // Світлі пікселі на темному тлі.
display.display(); // Передаємо очищений буфер на екран.
}

void loop() { // Новий кадр із новим вимірюванням.
    float cm = readDistanceCm(); // Вимірюємо відстань.
    display.clearDisplay(); // Прибираємо попередні цифри та шкалу.
    display.setTextSize(1); // Малий шрифт заголовка.
    display.setCursor(0, 0); // Початок верхнього рядка.
    display.print(F("Distance")); // Латиниця підтримується стандартним шрифтом.
    display.setCursor(0, 20); // Місце для результату.

    if (cm < 0) { // Відлуння не отримано або значення поза межами.
        display.setTextSize(1); // Помилка має вміститися в ширину екрана.
        display.print(F("No echo")); // Не показуємо помилковий нуль.
    } else { // Є вимірювання.
        display.setTextSize(2); // Великі цифри.
        display.print(cm, 0); // Виводимо округлену відстань.
        display.print(F(" cm")); // Одиниця вимірювання.
        int width = map(constrain((int)cm, 0, 100), 0, 100, 0, 124); // Шкала 0–100 см.
        display.drawRect(0, 49, 128, 14, SSD1306_WHITE); // Контур шкали.
        display.fillRect(2, 51, width, 10, SSD1306_WHITE); // Заповнюємо частину шкали.
    }

    display.display(); // Лише тепер показуємо готовий кадр.
    Serial.println(cm); // Дублюємо вимірювання для діагностики.
    delay(100); // Пауза між вимірюваннями більша за 60 мс.
}

```