УРОК 08 · КРОК 1 / 6

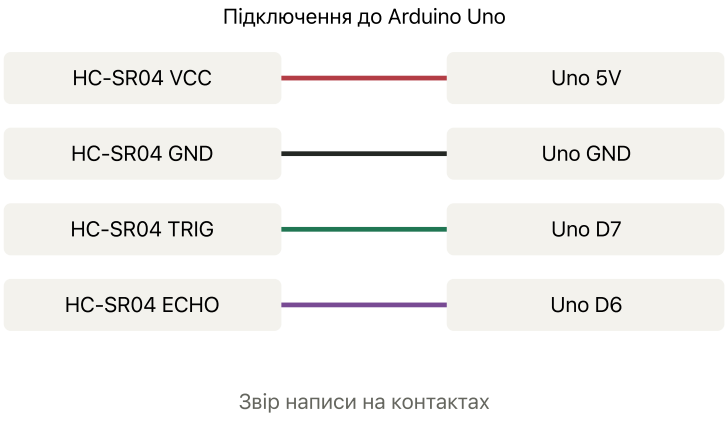
Відстань уже є в моніторі

1. Візьми справну схему з уроку 07.
2. Додай готовий LCD I²C, макетну плату й дроти.
3. Переконайся, що монітор показує відстань до картону.

Датчик перевірений окремо. Спочатку працює код без LCD з цього кроку.

Що можна змінити

READ_MS = 500. LCD_ADDRESS з’явиться в наступній програмі.



Звір написи на контактах

Крок 1: код · 1/1

Це одна програма. Частини коду читаємо по черзі.

```
#include <Arduino.h>
const byte TRIG_PIN = 7;
const byte ECHO_PIN = 6;
const unsigned long READ_MS = 500; // Пауза між показами

float readDistanceCm() {
  digitalWrite(TRIG_PIN, LOW); // Готуємо короткий імпульс
  delayMicroseconds(2);
  digitalWrite(TRIG_PIN, HIGH); // Запускаємо вимірювання
  delayMicroseconds(10);
  digitalWrite(TRIG_PIN, LOW);
  unsigned long echoUs = pulseIn(ECHO_PIN, HIGH, 25000UL);
  float cm = echoUs / 58.0; // Шлях туди й назад уже врахований

  if (echoUs == 0 || cm < 2 || cm > 400) {
    return -1; // Немає надійного вимірювання
  }

  return cm;
}

void setup() {
  pinMode(TRIG_PIN, OUTPUT); // TRIG — команда датчику
  pinMode(ECHO_PIN, INPUT); // ECHO — відповідь датчика
  digitalWrite(TRIG_PIN, LOW);
  Serial.begin(115200); // Така сама швидкість у моніторі
}

void loop() {
  float cm = readDistanceCm();

  if (cm < 0) {
    Serial.println("No echo");
  } else {
    Serial.print("Distance: ");
    Serial.print(cm, 1);
    Serial.println(" cm");
  }

  delay(READ_MS);
}
```

УРОК 08 · КРОК 2 / 6

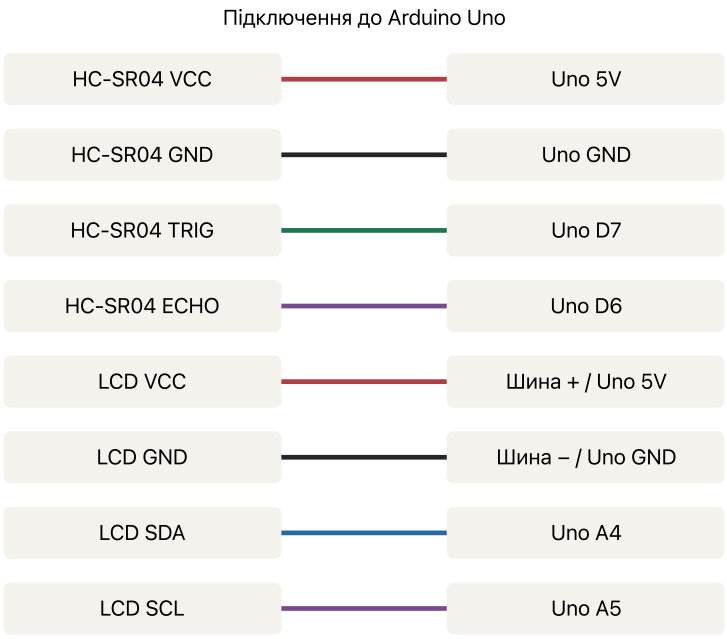
Спільне живлення двох модулів

1. Від’єднай USB.
2. Uno `5V` → шина +, `GND` → шина –.
3. `VCC` датчика й LCD → шина +; обидва `GND` → шина –.
4. LCD SDA → `A4`, SCL → `A5`.
5. TRIG лиш на `D7`, ECHO на `D6`.
6. Звір підключення зі схемою, під’єднай USB і завантаж код.

Датчик і екран під’єднані паралельно. Якщо шина розірвана посередині, користуйся однією половиною. Встанови LiquidCrystal I2C, як у 06.

Що можна змінити

LCD_ADDRESS — адреса, перевірена вчителем (Wokwi: 0x27). TITLE — заголовок до 16 символів.



Звір написи на контактах

Крок 2: код · 1/2

Це одна програма. Частини коду читаємо по черзі.

```
#include <Arduino.h>
const byte TRIG_PIN = 7;
const byte ECHO_PIN = 6;
const unsigned long READ_MS = 500; // Пауза між показами
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
const byte LCD_ADDRESS = 0x27; // Звірити адресу фізичного LCD
const char TITLE[] = "Distance:"; // До 16 символів
LiquidCrystal_I2C lcd(LCD_ADDRESS, 16, 2);

float readDistanceCm() {
  digitalWrite(TRIG_PIN, LOW); // Готуємо короткий імпульс
  delayMicroseconds(2);
  digitalWrite(TRIG_PIN, HIGH); // Запускаємо вимірювання
  delayMicroseconds(10);
  digitalWrite(TRIG_PIN, LOW);
  unsigned long echoUs = pulseIn(ECHO_PIN, HIGH, 25000UL);
  float cm = echoUs / 58.0; // Шлях туди й назад уже врахований

  if (echoUs == 0 || cm < 2 || cm > 400) {
    return -1; // Немає надійного вимірювання
  }

  return cm;
}

void setup() {
  pinMode(TRIG_PIN, OUTPUT); // TRIG — команда датчику
  pinMode(ECHO_PIN, INPUT); // ECHO — відповідь датчика
  digitalWrite(TRIG_PIN, LOW);
  Serial.begin(115200); // Така сама швидкість у моніторі
```

Крок 2: код · 2/2

Це одна програма. Частини коду читаємо по черзі.

```
  lcd.init();
  lcd.backlight();
  lcd.setCursor(0, 0);
  lcd.print(TITLE);
}

void loop() {
  float cm = readDistanceCm();

  if (cm < 0) {
    Serial.println("No echo");
  } else {
    Serial.print("Distance: ");
    Serial.print(cm, 1);
    Serial.println(" cm");
  }

  lcd.setCursor(0, 1);
  lcd.print("                "); // 16 пробілів стирають старий показ
  lcd.setCursor(0, 1);

  if (cm < 0) {
    lcd.print("No echo");
  } else {
    lcd.print(cm, 1); // Один знак після крапки
    lcd.print(" cm");
  }

  delay(READ_MS);
}
```

УРОК 08 · КРОК 3 / 6

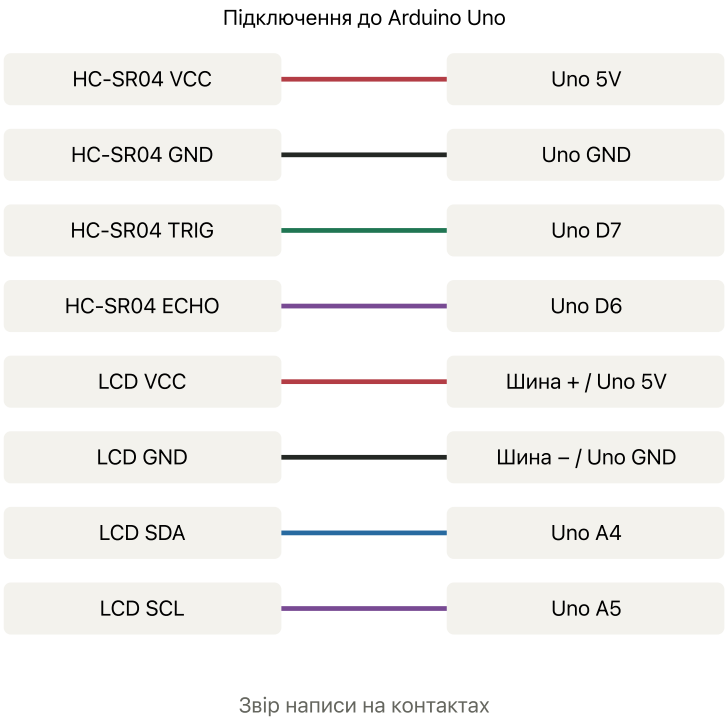
Одна відстань у двох місцях

1. В Arduino IDE вибери плату Arduino Uno та її порт.
2. Завантаж код.
3. Відкрий «Інструменти → Монітор порту» (Tools → Serial Monitor).
4. Унизу вибери 115200 baud — як у Serial.begin(115200).
5. Поле надсилання залиш порожнім.
6. У Wokwi запусти симуляцію: внизу з'явиться Serial Monitor.
7. Подивись одночасно на LCD і монітор.
8. Перемісти картон з 20 на 40 см.

У LCD другий рядок показує число й cm. У моніторі — те саме вимірювання; невелика затримка між оновленнями нормальна.

Що можна змінити

READ_MS: 200–1000 мс. TITLE: наприклад "Vidstan:". Контраст налаштуй регулятором.



Звір написи на контактах

Крок 3: код · 1/2

Це одна програма. Частини коду читаємо по черзі.

```
#include <Arduino.h>
const byte TRIG_PIN = 7;
const byte ECHO_PIN = 6;
const unsigned long READ_MS = 500; // Пауза між показами
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
const byte LCD_ADDRESS = 0x27; // Звірити адресу фізичного LCD
const char TITLE[] = "Distance:"; // До 16 символів
LiquidCrystal_I2C lcd(LCD_ADDRESS, 16, 2);

float readDistanceCm() {
  digitalWrite(TRIG_PIN, LOW); // Готуємо короткий імпульс
  delayMicroseconds(2);
  digitalWrite(TRIG_PIN, HIGH); // Запускаємо вимірювання
  delayMicroseconds(10);
  digitalWrite(TRIG_PIN, LOW);
  unsigned long echoUs = pulseIn(ECHO_PIN, HIGH, 25000UL);
  float cm = echoUs / 58.0; // Шлях туди й назад уже врахований

  if (echoUs == 0 || cm < 2 || cm > 400) {
    return -1; // Немає надійного вимірювання
  }

  return cm;
}

void setup() {
  pinMode(TRIG_PIN, OUTPUT); // TRIG — команда датчику
  pinMode(ECHO_PIN, INPUT); // ECHO — відповідь датчика
  digitalWrite(TRIG_PIN, LOW);
  Serial.begin(115200); // Така сама швидкість у моніторі
```

Крок 3: код · 2/2

Це одна програма. Частини коду читаємо по черзі.

```
  lcd.init();
  lcd.backlight();
  lcd.setCursor(0, 0);
  lcd.print(TITLE);
}

void loop() {
  float cm = readDistanceCm();

  if (cm < 0) {
    Serial.println("No echo");
  } else {
    Serial.print("Distance: ");
    Serial.print(cm, 1);
    Serial.println(" cm");
  }

  lcd.setCursor(0, 1);
  lcd.print("                "); // 16 пробілів стирають старий показ
  lcd.setCursor(0, 1);

  if (cm < 0) {
    lcd.print("No echo");
  } else {
    lcd.print(cm, 1); // Один знак після крапки
    lcd.print(" cm");
  }

  delay(READ_MS);
}
```

УРОК 08 · КРОК 4 / 6

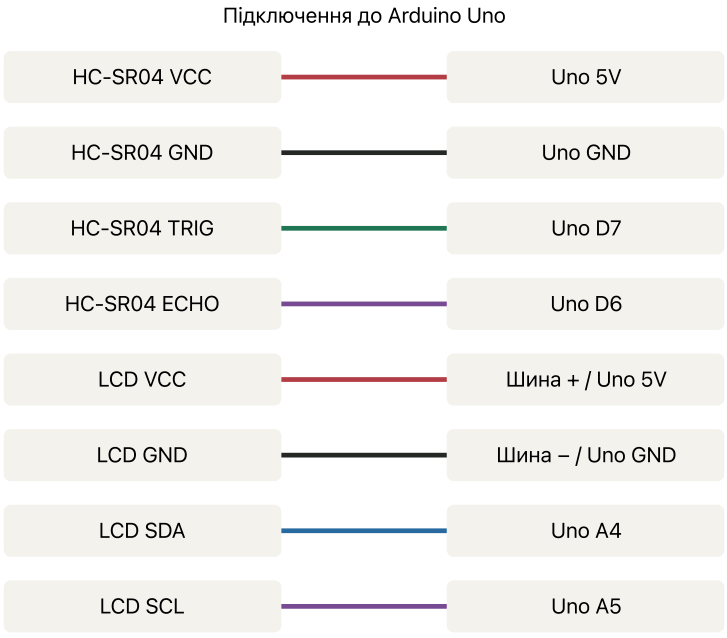
Зі 120 см до 8 см

1. У Wokwi задай 120 см, потім 8 см.
2. На фізичному датчику використай доступні в класі велику й малу відстані.

Після коротшого числа не лишаються цифри попереднього. 16 пробілів стирають лише другий рядок; курсор повертається на його початок.

Що можна змінити

Зміни TITLE на власний заголовок. Залиш 16 пробілів та обидва setCursor(0, 1).



Звір написи на контактах

Крок 4: код · 1/2

Це одна програма. Частини коду читаємо по черзі.

```
#include <Arduino.h>
const byte TRIG_PIN = 7;
const byte ECHO_PIN = 6;
const unsigned long READ_MS = 500; // Пауза між показами
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
const byte LCD_ADDRESS = 0x27; // Звірити адресу фізичного LCD
const char TITLE[] = "Distance:"; // До 16 символів
LiquidCrystal_I2C lcd(LCD_ADDRESS, 16, 2);

float readDistanceCm() {
  digitalWrite(TRIG_PIN, LOW); // Готуємо короткий імпульс
  delayMicroseconds(2);
  digitalWrite(TRIG_PIN, HIGH); // Запускаємо вимірювання
  delayMicroseconds(10);
  digitalWrite(TRIG_PIN, LOW);
  unsigned long echoUs = pulseIn(ECHO_PIN, HIGH, 25000UL);
  float cm = echoUs / 58.0; // Шлях туди й назад уже врахований

  if (echoUs == 0 || cm < 2 || cm > 400) {
    return -1; // Немає надійного вимірювання
  }

  return cm;
}

void setup() {
  pinMode(TRIG_PIN, OUTPUT); // TRIG — команда датчику
  pinMode(ECHO_PIN, INPUT); // ECHO — відповідь датчика
  digitalWrite(TRIG_PIN, LOW);
  Serial.begin(115200); // Така сама швидкість у моніторі
```

Крок 4: код · 2/2

Це одна програма. Частини коду читаємо по черзі.

```
  lcd.init();
  lcd.backlight();
  lcd.setCursor(0, 0);
  lcd.print(TITLE);
}

void loop() {
  float cm = readDistanceCm();

  if (cm < 0) {
    Serial.println("No echo");
  } else {
    Serial.print("Distance: ");
    Serial.print(cm, 1);
    Serial.println(" cm");
  }

  lcd.setCursor(0, 1);
  lcd.print("                "); // 16 пробілів стирають старий показ
  lcd.setCursor(0, 1);

  if (cm < 0) {
    lcd.print("No echo");
  } else {
    lcd.print(cm, 1); // Один знак після крапки
    lcd.print(" cm");
  }

  delay(READ_MS);
}
```

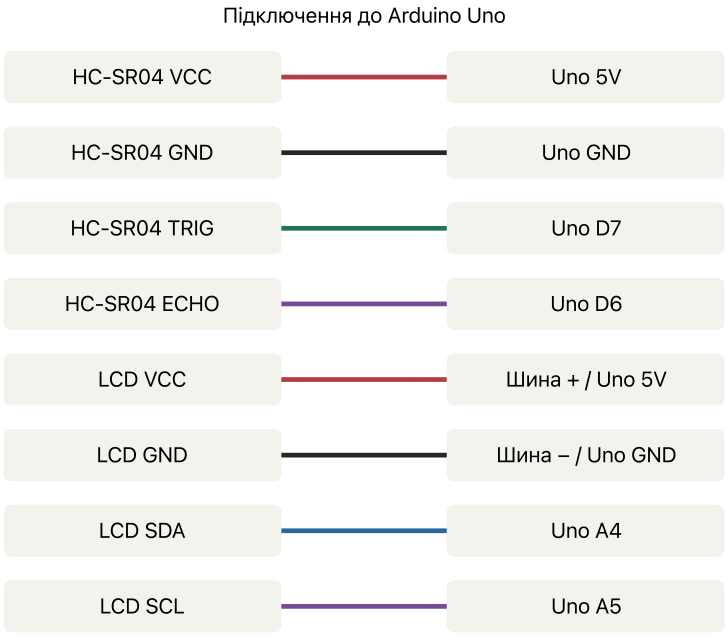
Замість вигаданого числа

1. Поверни картон убік або прибери його.
2. Поспостерігай, коли замість відстані з’являється No echo.

No echo з’являється замість числа. Коли відлуння повертається, знову бачиш сантиметри без залишків старого напису.

Що можна змінити

Не міняй умову cm < 0: вона відокремлює помилку від відстані.



Звір написи на контактах

Крок 5: код · 1/2

Це одна програма. Частини коду читаємо по черзі.

```
#include <Arduino.h>
const byte TRIG_PIN = 7;
const byte ECHO_PIN = 6;
const unsigned long READ_MS = 500; // Пауза між показами
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
const byte LCD_ADDRESS = 0x27; // Звірити адресу фізичного LCD
const char TITLE[] = "Distance:"; // До 16 символів
LiquidCrystal_I2C lcd(LCD_ADDRESS, 16, 2);

float readDistanceCm() {
  digitalWrite(TRIG_PIN, LOW); // Готуємо короткий імпульс
  delayMicroseconds(2);
  digitalWrite(TRIG_PIN, HIGH); // Запускаємо вимірювання
  delayMicroseconds(10);
  digitalWrite(TRIG_PIN, LOW);
  unsigned long echoUs = pulseIn(ECHO_PIN, HIGH, 25000UL);
  float cm = echoUs / 58.0; // Шлях туди й назад уже врахований

  if (echoUs == 0 || cm < 2 || cm > 400) {
    return -1; // Немає надійного вимірювання
  }

  return cm;
}

void setup() {
  pinMode(TRIG_PIN, OUTPUT); // TRIG — команда датчику
  pinMode(ECHO_PIN, INPUT); // ECHO — відповідь датчика
  digitalWrite(TRIG_PIN, LOW);
  Serial.begin(115200); // Така сама швидкість у моніторі
```

Крок 5: код · 2/2

Це одна програма. Частини коду читаємо по черзі.

```
  lcd.init();
  lcd.backlight();
  lcd.setCursor(0, 0);
  lcd.print(TITLE);
}

void loop() {
  float cm = readDistanceCm();

  if (cm < 0) {
    Serial.println("No echo");
  } else {
    Serial.print("Distance: ");
    Serial.print(cm, 1);
    Serial.println(" cm");
  }

  lcd.setCursor(0, 1);
  lcd.print("                "); // 16 пробілів стирають старий показ
  lcd.setCursor(0, 1);

  if (cm < 0) {
    lcd.print("No echo");
  } else {
    lcd.print(cm, 1); // Один знак після крапки
    lcd.print(" cm");
  }

  delay(READ_MS);
}
```

УРОК 08 · КРОК 6 / 6

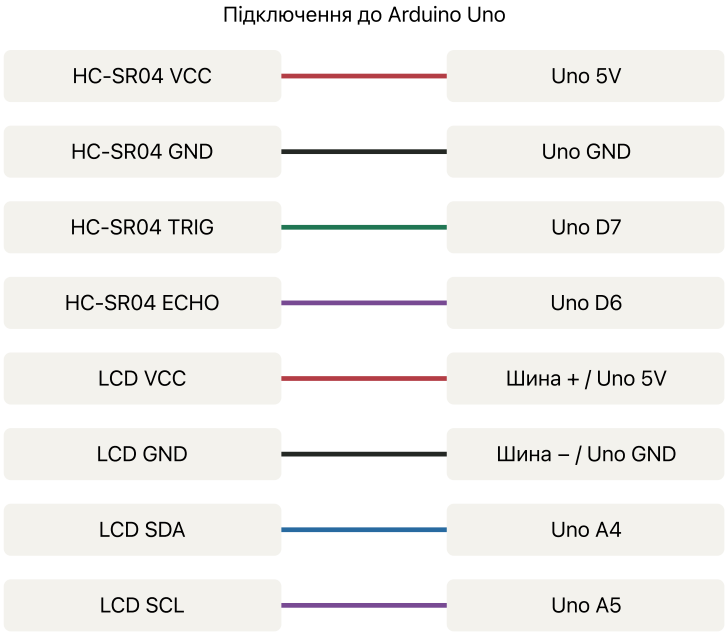
Власний заголовок і темп

1. Задай TITLE = "Vidstan:" та READ_MS = 300.
2. Перевір 10, 20 і 40 см, збережи скетч Distance_LCD.

Заголовок залишається на місці, а відстань оновлюється приблизно тричі на секунду.

Що можна змінити

TITLE — текст; READ_MS — пауза. Датчик, формула та піни залишаються тими самими.



Звір написи на контактах

Крок 6: код · 1/2

Це одна програма. Частини коду читаємо по черзі.

```
#include <Arduino.h>
const byte TRIG_PIN = 7;
const byte ECHO_PIN = 6;
const unsigned long READ_MS = 500; // Пауза між показами
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
const byte LCD_ADDRESS = 0x27; // Звірити адресу фізичного LCD
const char TITLE[] = "Distance:"; // До 16 символів
LiquidCrystal_I2C lcd(LCD_ADDRESS, 16, 2);

float readDistanceCm() {
  digitalWrite(TRIG_PIN, LOW); // Готуємо короткий імпульс
  delayMicroseconds(2);
  digitalWrite(TRIG_PIN, HIGH); // Запускаємо вимірювання
  delayMicroseconds(10);
  digitalWrite(TRIG_PIN, LOW);
  unsigned long echoUs = pulseIn(ECHO_PIN, HIGH, 25000UL);
  float cm = echoUs / 58.0; // Шлях туди й назад уже врахований

  if (echoUs == 0 || cm < 2 || cm > 400) {
    return -1; // Немає надійного вимірювання
  }

  return cm;
}

void setup() {
  pinMode(TRIG_PIN, OUTPUT); // TRIG — команда датчику
  pinMode(ECHO_PIN, INPUT); // ECHO — відповідь датчика
  digitalWrite(TRIG_PIN, LOW);
  Serial.begin(115200); // Така сама швидкість у моніторі
```

Крок 6: код · 2/2

Це одна програма. Частини коду читаємо по черзі.

```
  lcd.init();
  lcd.backlight();
  lcd.setCursor(0, 0);
  lcd.print(TITLE);
}

void loop() {
  float cm = readDistanceCm();

  if (cm < 0) {
    Serial.println("No echo");
  } else {
    Serial.print("Distance: ");
    Serial.print(cm, 1);
    Serial.println(" cm");
  }

  lcd.setCursor(0, 1);
  lcd.print("                "); // 16 пробілів стирають старий показ
  lcd.setCursor(0, 1);

  if (cm < 0) {
    lcd.print("No echo");
  } else {
    lcd.print(cm, 1); // Один знак після крапки
    lcd.print(" cm");
  }

  delay(READ_MS);
}
```

Наступний урок: Серво: перші положення →