LCD: два рядки по 16 символів

1. Візьми Uno, USB-кабель, LCD 1602 із припаяним I²C-перехідником і 4 дроти «мама–тато».
2. Знайди на перехіднику контакти `GND`, `VCC`, SDA та SCL.
3. Якщо перехідник окремо — попроси підготовлений екран.

Є готовий модуль із чотирма контактами. Макетна плата й датчики поки не потрібні.

Іди по черзі. На кроках із кодом подано повну програму: замінюй нею весь скетч. Не додавай новий код під старим. Змінюй один параметр, запускай і порівнюй результат.



Звір написи на контактах

З'єднай LCD з Uno

1. Від'єднай USB.
2. Підключи `GND` → `GND`, `VCC` → `5V`, SDA → `A4`, SCL → `A5` за схемою нижче.
3. Звір підключення зі схемою й під'єднай USB.

Усі чотири дроти на своїх місцях. SDA передає дані, SCL задає такт. Uno отримує живлення через USB, LCD — від `5V` Uno; `GND` спільний.

Який дріт за що відповідає

- `GND` **— ground, спільний нуль.** З'єднуємо з `GND` Uno: це спільна точка відліку напруги й зворотний шлях струму.
- `VCC` **— живлення модуля.** З'єднуємо з `5V` Uno. Через цей контакт екран отримує енергію.
- **SDA — Serial Data, лінія даних.** З'єднуємо з `A4` Uno. Нею передаються команди й символи для екрана.
- **SCL — Serial Clock, тактова лінія.** З'єднуємо з `A5` Uno. Arduino подає імпульси, за якими модуль визначає, коли читати черговий біт даних.

SDA й SCL разом утворюють інтерфейс I²C. На Uno ці самі лінії також виведені на контакти з підписами SDA та SCL. Тут використовуємо `A4` та `A5`, як на схемі.



Звір написи на контактах

Перший напис Hello!

1. В Arduino IDE встанови через менеджер бібліотек LiquidCrystal I2C від Frank de Brabander.
2. Відкрий новий скетч, заміни весь код кодом цього кроку.
3. Вибери Arduino Uno та порт і натисни «Завантажити».

У першому рядку з'являється Hello! Якщо видно лише світле тло або квадрати — переходь до налаштування контрасту.

Що можна змінити

TEXT — напис у лапках. На цьому кроці залиш "Hello!", щоб перевірити запуск.

Від команди до напису

`lcd.init()` готує екран до роботи. `lcd.setCursor(0, 0)` обирає початок першого рядка. `lcd.print(TEXT)` передає символи змінної TEXT на екран.

Бібліотека перетворює ці команди на обмін через SDA й SCL. I²C-перехідник приймає їх і керує LCD. `LCD_ADDRESS = 0x27` — адреса модуля на шині, а не номер піна; номер `A4` у `lcd.print()` писати не потрібно.

У Wokwi відкрий новий Arduino Uno. Перенеси diagram.json з архіву кроку до вкладки схеми, sketch/sketch.ino — до вкладки коду, libraries.txt — до бібліотек. Натисни запуск.



Звір написи на контактах

Крок 3: код · 1/1

Це одна програма. Частини коду читаємо по черзі.

```
#include <Arduino.h>
#include <Wire.h>
#include <LiquidCrystal_I2C.h>

const byte LCD_ADDRESS = 0x27; // Адреса, яку перевірив учитель
LiquidCrystal_I2C lcd(LCD_ADDRESS, 16, 2);
const char TEXT[] = "Hello!"; // Текст латиницею
const byte COLUMN = 0; // Позиція: 0–15
const byte ROW = 0; // Рядок: 0 або 1
const bool LIGHT_ON = true; // Підсвічування: true / false

void setup() {
  lcd.init(); // Готуємо екран

  if (LIGHT_ON) {
    lcd.backlight();
  } else {
    lcd.noBacklight();
  }

  lcd.setCursor(COLUMN, ROW); // Обираємо місце
  lcd.print(TEXT); // Друкуємо текст
}

void loop() {
}
```

Знайди найчіткіші символи

1. Залиш програму Hello! запущеною.
2. Маленькою викруткою повільно обертай синій регулятор на звороті LCD.
3. Порівняй слабкі символи, чіткий текст і надто темні клітинки.
4. Залиш найчитабельніше положення.

Літери чіткі, порожні клітинки не зливаються в темні прямокутники. Це контраст символів, а не яскравість лампи. Не тисни на упор регулятора. Цей дослід виконується на справжньому модулі; Wokwi контраст не моделює.



Звір написи на контактах

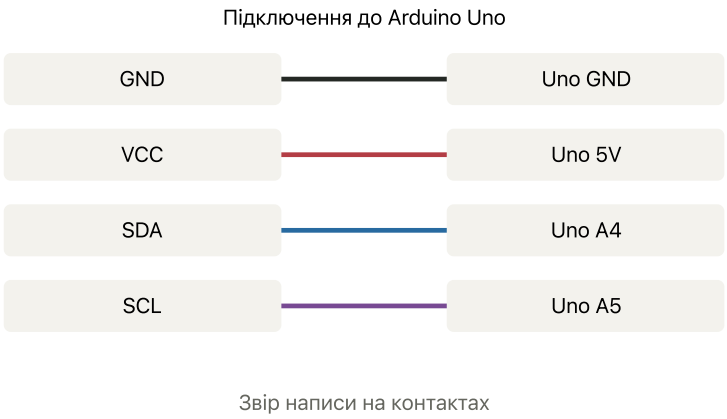
Власне привітання

1. Заміни Olya у TEXT на своє ім’я латиницею.
2. Залиш лапки.
3. Завантаж змінений код.

На екрані твоє привітання. Разом із пробілами й розділовими знаками — не більше 16 символів. Український текст у UTF-8 цей приклад напряму не підтримує.

Що можна змінити

TEXT: наприклад "Ya Maksym" або "Pryvit, ya Eva!". Порахуй довжину перед запуском.



Звір написи на контактах

Крок 5: код · 1/1

Це одна програма. Частини коду читаємо по черзі.

```
#include <Arduino.h>
#include <Wire.h>
#include <LiquidCrystal_I2C.h>

const byte LCD_ADDRESS = 0x27; // Адреса, яку перевірив учитель
LiquidCrystal_I2C lcd(LCD_ADDRESS, 16, 2);
const char TEXT[] = "Pryvit, ya Olya!"; // Текст латиницею
const byte COLUMN = 0; // Позиція: 0–15
const byte ROW = 0; // Рядок: 0 або 1
const bool LIGHT_ON = true; // Підсвічування: true / false

void setup() {
  lcd.init(); // Готуємо екран

  if (LIGHT_ON) {
    lcd.backlight();
  } else {
    lcd.noBacklight();
  }

  lcd.setCursor(COLUMN, ROW); // Обираємо місце
  lcd.print(TEXT); // Друкуємо текст
}

void loop() {
}
```

Позиція починається з нуля

1. Завантаж цей короткий напис.
2. Зміни COLUMN з 0 на 3 та завантаж ще раз.

Olya починається з четвертої клітинки. COLUMN = 0 означає першу. Після зміни коду його треба завантажити заново.

Що можна змінити

COLUMN: 0–12 для тексту "Olya". Загальне правило: COLUMN + довжина TEXT ≤ 16.



Звір написи на контактах

Крок 6: код · 1/1

Це одна програма. Частини коду читаємо по черзі.

```
#include <Arduino.h>
#include <Wire.h>
#include <LiquidCrystal_I2C.h>

const byte LCD_ADDRESS = 0x27; // Адреса, яку перевірив учитель
LiquidCrystal_I2C lcd(LCD_ADDRESS, 16, 2);
const char TEXT[] = "Olya"; // Текст латиницею
const byte COLUMN = 0; // Позиція: 0–15
const byte ROW = 0; // Рядок: 0 або 1
const bool LIGHT_ON = true; // Підсвічування: true / false

void setup() {
  lcd.init(); // Готуємо екран

  if (LIGHT_ON) {
    lcd.backlight();
  } else {
    lcd.noBacklight();
  }

  lcd.setCursor(COLUMN, ROW); // Обираємо місце
  lcd.print(TEXT); // Друкуємо текст
}

void loop() {
}
```

Перенеси ім’я вниз

1. Зміни ROW з 0 на 1.
2. Завантаж код і порівняй положення напису.

Ім’я у другому рядку. ROW = 0 — верхній, ROW = 1 — нижній. Число 2 тут не використовуємо.

Що можна змінити

ROW: 0 або 1. COLUMN і TEXT залиш без змін для чистого порівняння.



Звір написи на контактах

Крок 7: код · 1/1

Це одна програма. Частини коду читаємо по черзі.

```
#include <Arduino.h>
#include <Wire.h>
#include <LiquidCrystal_I2C.h>

const byte LCD_ADDRESS = 0x27; // Адреса, яку перевірив учитель
LiquidCrystal_I2C lcd(LCD_ADDRESS, 16, 2);
const char TEXT[] = "Olya"; // Текст латиницею
const byte COLUMN = 3; // Позиція: 0–15
const byte ROW = 0; // Рядок: 0 або 1
const bool LIGHT_ON = true; // Підсвічування: true / false

void setup() {
  lcd.init(); // Готуємо екран

  if (LIGHT_ON) {
    lcd.backlight();
  } else {
    lcd.noBacklight();
  }

  lcd.setCursor(COLUMN, ROW); // Обираємо місце
  lcd.print(TEXT); // Друкуємо текст
}

void loop() {
}
```

Ім’я та коротке повідомлення

1. Завантаж код цього кроку.
2. Впиши власне ім’я в TEXT, а коротке повідомлення — у SECOND_LINE.

Перший і другий рядки мають різні написи. lcd.setCursor(0, 1) переходить на початок другого рядка; print друкує від курсора.

Що можна змінити

TEXT і SECOND_LINE: кожний напис до 16 символів. У цьому кроці COLUMN = 0 і ROW = 0.



Звір написи на контактах

Крок 8: код · 1/1

Це одна програма. Частини коду читаємо по черзі.

```
#include <Arduino.h>
#include <Wire.h>
#include <LiquidCrystal_I2C.h>

const byte LCD_ADDRESS = 0x27; // Адреса, яку перевірив учитель
LiquidCrystal_I2C lcd(LCD_ADDRESS, 16, 2);
const char TEXT[] = "Pryvit, ya Olya!"; // Текст латиницею
const byte COLUMN = 0; // Позиція: 0–15
const byte ROW = 0; // Рядок: 0 або 1
const bool LIGHT_ON = true; // Підсвічування: true / false
const char SECOND_LINE[] = "Mii pershyi LCD"; // Другий рядок

void setup() {
  lcd.init(); // Готуємо екран

  if (LIGHT_ON) {
    lcd.backlight();
  } else {
    lcd.noBacklight();
  }

  lcd.setCursor(COLUMN, ROW); // Обираємо місце
  lcd.print(TEXT); // Друкуємо текст
  lcd.setCursor(0, 1); // Початок другого рядка
  lcd.print(SECOND_LINE);
}

void loop() {
}
```

Світло і символи — різні речі

1. Зміни LIGHT_ON з true на false.
2. Завантаж код.
3. Потім поверни true й завантаж знову.

Підсвічування вимикається та вмикається; без нього текст може бути ледь видимим при зовнішньому освітленні. Регулятор контрасту залиш на знайденому положенні.

Що можна змінити

LIGHT_ON: тільки true або false. Стандартний I²C-перехідник у цьому уроці не має плавного регулювання яскравості 0–255.



Звір написи на контактах

Крок 9: код · 1/1

Це одна програма. Частини коду читаємо по черзі.

```
#include <Arduino.h>
#include <Wire.h>
#include <LiquidCrystal_I2C.h>

const byte LCD_ADDRESS = 0x27; // Адреса, яку перевірив учитель
LiquidCrystal_I2C lcd(LCD_ADDRESS, 16, 2);
const char TEXT[] = "Pryvit, ya Olya!"; // Текст латиницею
const byte COLUMN = 0; // Позиція: 0–15
const byte ROW = 0; // Рядок: 0 або 1
const bool LIGHT_ON = true; // Підсвічування: true / false
const char SECOND_LINE[] = "Mii pershyi LCD"; // Другий рядок

void setup() {
  lcd.init(); // Готуємо екран

  if (LIGHT_ON) {
    lcd.backlight();
  } else {
    lcd.noBacklight();
  }

  lcd.setCursor(COLUMN, ROW); // Обираємо місце
  lcd.print(TEXT); // Друкуємо текст
  lcd.setCursor(0, 1); // Початок другого рядка
  lcd.print(SECOND_LINE);
}

void loop() {
}
```

Два власні написи

1. Напиши своє ім’я в першому рядку й улюблене заняття в другому.
2. Увімкни підсвічування.
3. Збережи скетч як My_LCD.

Обидва написи мають вміщуватися у свої рядки. За потреби скороти текст або підлаштуй контраст.

Що можна змінити

Змінюй по одному параметру: TEXT, SECOND_LINE, LIGHT_ON. Для цієї візитівки залиш COLUMN = 0 і ROW = 0.



Звір написи на контактах

Крок 10: код · 1/1

Це одна програма. Частини коду читаємо по черзі.

```
#include <Arduino.h>
#include <Wire.h>
#include <LiquidCrystal_I2C.h>

const byte LCD_ADDRESS = 0x27; // Адреса, яку перевірив учитель
LiquidCrystal_I2C lcd(LCD_ADDRESS, 16, 2);
const char TEXT[] = "Pryvit, ya Olya!"; // Текст латиницею
const byte COLUMN = 0; // Позиція: 0–15
const byte ROW = 0; // Рядок: 0 або 1
const bool LIGHT_ON = true; // Підсвічування: true / false
const char SECOND_LINE[] = "Mii pershyi LCD"; // Другий рядок

void setup() {
  lcd.init(); // Готуємо екран

  if (LIGHT_ON) {
    lcd.backlight();
  } else {
    lcd.noBacklight();
  }

  lcd.setCursor(COLUMN, ROW); // Обираємо місце
  lcd.print(TEXT); // Друкуємо текст
  lcd.setCursor(0, 1); // Початок другого рядка
  lcd.print(SECOND_LINE);
}

void loop() {
}
```

Наступний урок: Відстань у моніторі порту →