Практичне заняття №6.

Робота з LCD – дисплеєм по протоколу I2C

Мета: написати програму для виводу даних і керування LCD – дисплеем через інтерфейс I2C.

Завдання: зібрати схему з використанням перехідника I2C і написати програму виведення на дисплей символів.

Обладнання: плата Arduino; дисплей LCD 1602 (або 2004) з підключеним перехідником I2C; проводи; макетна плата; кабель USB.

Загальні відомості

LCD – дисплей

LCD екран на базі 1602 (або 2004) з керуванням по шині I2C (TWI, IIC) дуже зручний при нестачі вільних виводів на Arduino. Досить підключити всього два провідники (само-собою не рахуючи живлення) і можна повністю управляти дисплеєм. Працювати з подібними дисплеями можна через бібліотеку LiquidCrystal_I2C.

Підключення стандартне, як і для всієї лінійки подібних дисплеїв на цьому контролері (рис. 1). Контраст екрану налаштовується змінним резистором зі зворотного боку плати. Живлення - 5В.

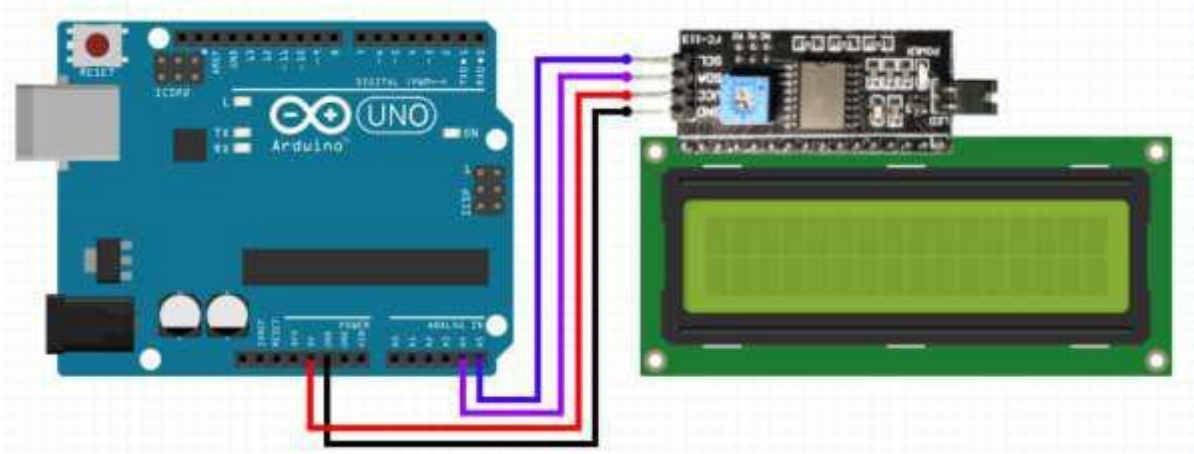

Рис. 1 – Схема підключення LCD – дисплею

I2C – адреса може бути: ***0x27 0x20 или на 0x3F***.

Табл. 1 – Співвідношення виводів, що відповідають за інтерфейс I2C на платах Arduino на базі різних контролерів

LCD I2C модуль	**Arduino Nano**	**Leonardo**	**MEGA, ADK, DUE**
SCL	A5	D3	D21
SDA	A4	D2	D20
VCC	+5V	+5V	+5V
GND	GND	GND	GND

Бібліотеки для роботи з LCD – дисплеєм

Для взаємодії Arduino з LCD 1602 (2004) по шині I2C необхідно як мінімум дві бібліотеки:

- Бібліотека Wire.h для роботи з I2C вже є в стандартній програмі
- Бібліотека LiquidCrystal_I2C.h, яка включає в себе велику різноманітність команд для управління монітором по шині I2C і дозволяє зробити скетч простішим і коротшим.

Після підключення до скетчу всіх необхідних бібліотек треба створити програмний об'єкт LCD-дісплей та використовувати всі його функції.

Опис функцій і методів бібліотеки LiquidCrystal_I2C:

- ***init ()*** – ініціалізація дисплею.
- ***backlight ()*** – підключення підсвічування.
- ***setCursor ()*** – установка курсора.
- ***print ()*** – набір тексту.
- ***home ()*** і ***clear ()*** – перша функція дозволяє повернути курсор в початок екрану, друга теж, але при цьому видаляє все, що було на моніторі до цього.
- ***write (ch)*** – дозволяє вивести одиночний символ *ch* на екран.
- ***cursor ()*** і ***noCursor ()*** – показує / приховує курсор на екрані.
- ***blink ()*** і ***noBlink ()*** – курсор блимає / не блимає (якщо до цього було включено його відображення).

- ***display ()*** і ***noDisplay ()*** – дозволяє підключити / відключити дисплей.
- ***scrollDisplayLeft ()*** і ***scrollDisplayRight ()*** – прокручує екран на один знак вліво / вправо.
- ***autoscroll ()*** і ***noAutoscroll ()*** – дозволяє включити / вимкнути режим автопрокручування. У цьому режимі кожен новий символ записується у одному і тому ж місці, витісняючи раніше написане на екрані.
- ***leftToRight ()*** і ***rightToLeft ()*** – установка напрямку виведеного тексту - зліва направо або справа наліво.
- ***createChar (ch, bitmap)*** – створює символ з кодом *ch* (0 - 7), використовуючи масив бітових масок *bitmap* для створення чорних і білих точок.

Приклад програми:

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h> // Підключення бібліотеки

LiquidCrystal_I2C lcd (0x27,16,2); // Вказуємо I2C адресу
(найбільш поширене значення), а також параметри екрану (в цьому
випадку LCD 1602 – 2 рядки по 16 символів в кожному)

void setup ()
{
  lcd.init (); // Ініціалізація дисплею lcd.backlight (); //
  Підключення підсвічування lcd.setCursor (0,0); //
  Установка курсора в початок першого
рядка
  lcd.print ( "Hello"); // Набір тексту на першому рядку
  lcd.setCursor (0,1); // Установка курсора в початок другого
рядка
  lcd.print ( "ArduinoMaster"); // Набір тексту на другому рядку
}
void loop ()
{
}
```

Якщо після завантаження скетчу не з'явилося жодної написи на дисплеї, треба виконати наступні дії:

- збільшити або зменшити контрастність монітора,
- перевірити чи правильно підключено контакти, чи підключено живлення підсвічування,
- може бути вказано неправильна адреса. Треба змінити в скетчі адресу пристрою з 0x27 0x20 або на 0x3F. У різних виробників можуть бути зашиті різні адреси за замовчуванням.

Створення власних символів

Варто зазначити, що літери англійського алфавіту зашиті в пам'ять контролера всередині дисплея і з ними проблем немає. Проте літери кирилиці модуль екрана не підтримує, тому необхідно їх створити самому.

Дисплей підтримує до 8 користувацьких символів розміром 5x8 пікселів. Зовнішній вигляд кожного користувацького символу задається масивом з восьми байт, кожен з яких характеризує відповідний рядок.

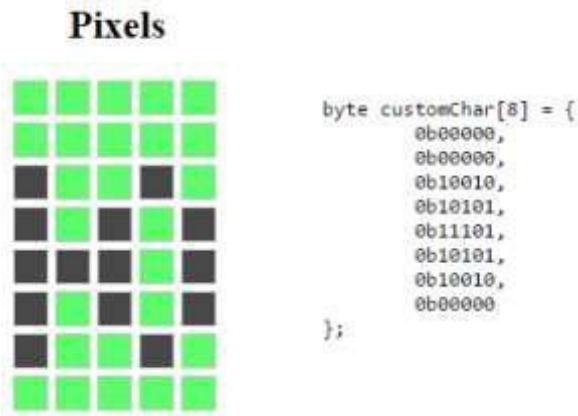

Рис. 2 – Приклад створення масиву станів пікселів

П'ять молодших біт кожного байта визначають стан пікселів у відповідному рядку. На рис. 4.17 приведений приклад створення масиву. На рис. добре видно, що дивлячись на масив, можна легко зрозуміти, який символ буде створено.

Після запису станів пікселів до масиву, за допомогою цього ж масиву створюється відповідний символ.

Хід виконання роботи

1. Зібрати макет відповідно завдання.
2. Підключити схему до живлення(5В).
3. Завантажити програму в мікроконтролер Arduino.
4. Перевірити правильність роботи програми.

Завдання

Зібрати схему зображену на рис. 1.

Створити програму, котра виводить на дисплей номер варіанта та прізвище українською мовою. Невистачаючі символи кирилиці створити власноруч.

Підготувати звіт згідно ДСТУ 3008-95 (лістинг програми, висновки, перелік посилань).

Контрольні питання

1. Опишіть особливості протоколу I2C / IIC (Inter-Integrated Circuit) та зазначте основні його переваги та недоліки.
2. Що таке LCD – дисплей та як з ним працювати?
3. Які існують бібліотеки для роботи з LCD – дисплеєм?
4. Які особливості функцій *scrollDisplayLeft ()* і *scrollDisplayRight (), autoscroll ()* і *noAutoscroll ()*?
5. Опишіть принцип створення власних символів для виводу їх на LCD – дисплей.