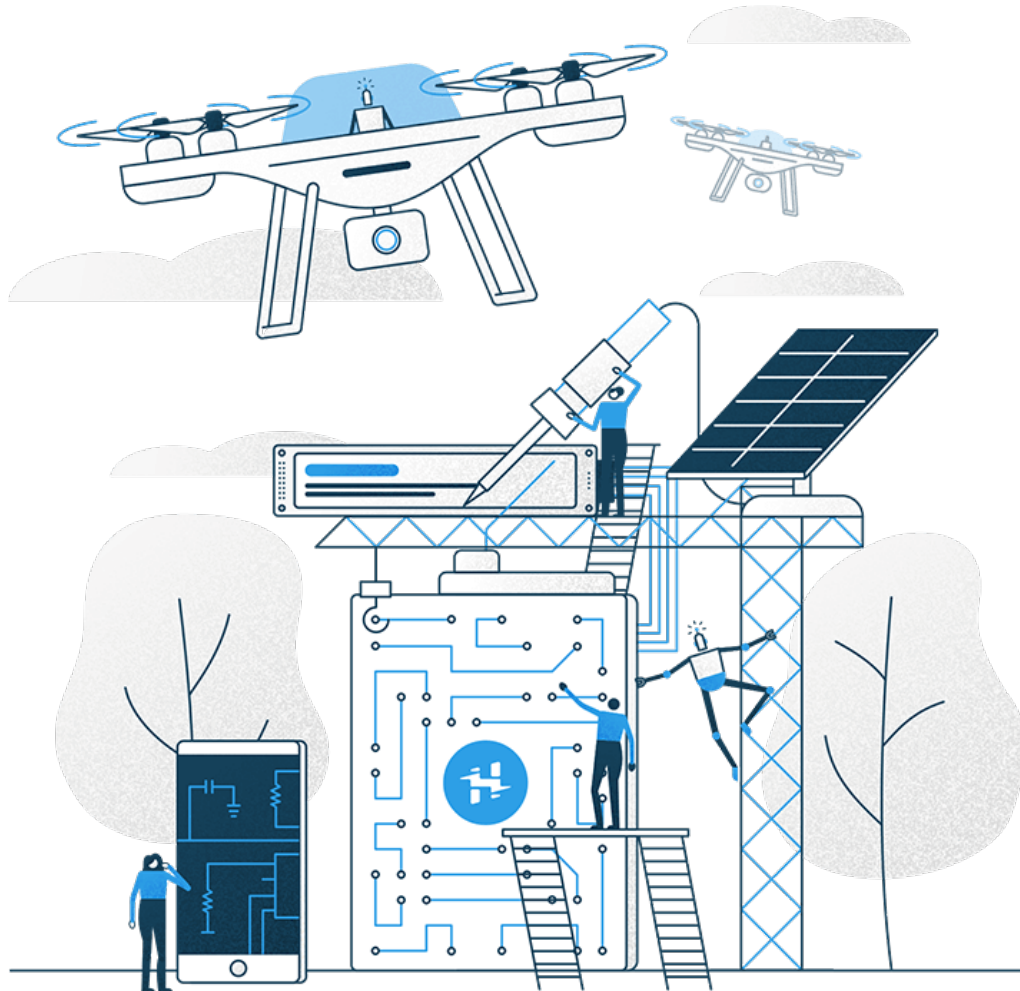


# МІКРОПРОЦЕСОРНІ СИСТЕМИ УПРАВЛІННЯ

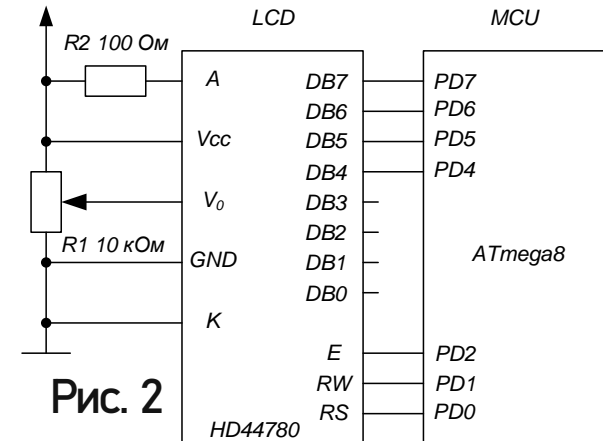
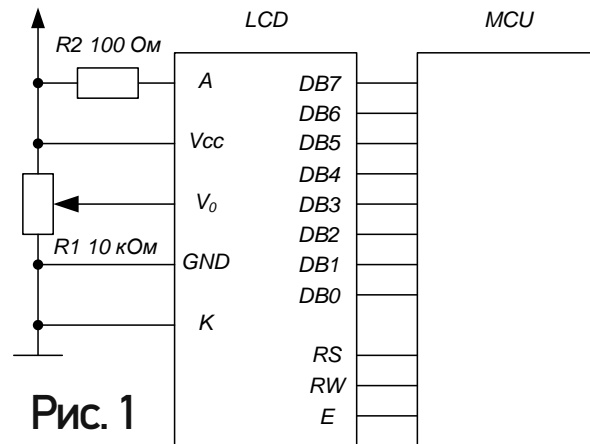


Lesson 6

# Елементи індикації

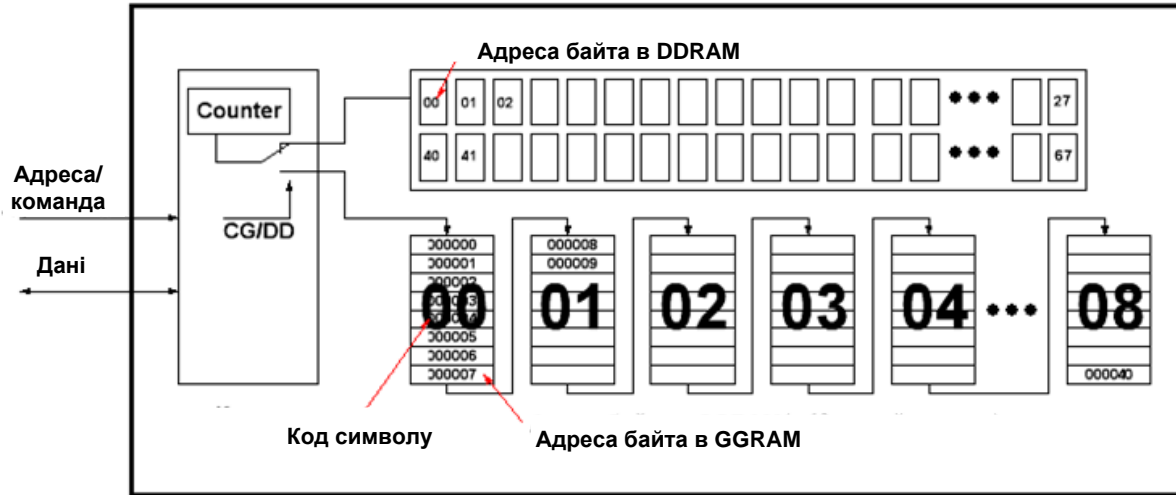
Для відображення поточних даних та інформації досить часто застосовують LCD індикатор з контролером типу HD44780 фірми Hitachi, який є промисловим стандартом де-факто на ринку цифро-буквених індикаторів.

LCD на базі HD44780 підключається до AVR мікроконтролера безпосередньо до портів (рис. 1). Є два способи підключення – на 8 біт і на 4 біти. У восьми бітному режимі простіше передавати байти – не потрібно зсувати байт, але в чотирьох бітному режимі використовується на чотири ніжки контролера менше. Для підключення індикатора до мікроконтролера буде задіяний порт D мікроконтролера ATmega8 у чотирьох бітному режимі (рис. 2).



Виводи DB7...DB0 це шина даних/адреси. E – вхід стробування. Імпульс напруги на цій лінії повідомляє дисплею, що потрібно забирати/віддавати дані з/на шину даних. RW – визначає в якому напрямку рухаються дані. Якщо 1, то читання з дисплея, якщо 0 то запис у дисплей. RS – визначає, що передається команда (RS=0) або дані (RS=1). Дані будуть записані на по поточній адресі, а команда виконана контролером. V0 – вхід контрастності. Vcc – плюс живлення, 5В. GND – мінус, він же загальний.

## Елементи індикації



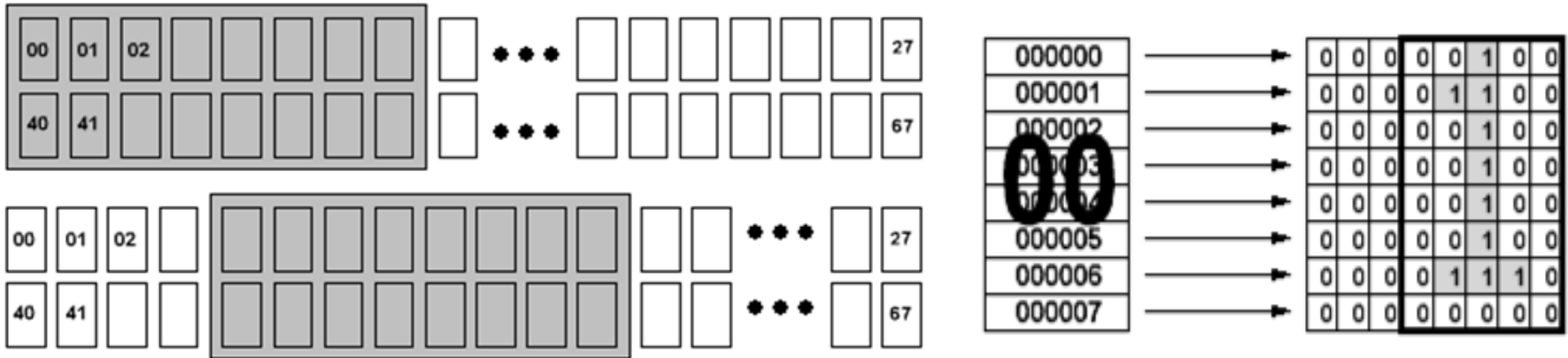
## Структура адресації контролера HD44780

Контролер HD44780 має свій блок керування, що обробляє команди й пам'ять. Пам'ять ділиться на три види: DDRAM, CGROM, CGRAM.

DDRAM – пам'ять дисплея. Все що запишеться в DDRAM буде виведено на екран. Тобто, наприклад, записали код 0x31 – на екрані відображається символ «1», так як 0x31 це ASCII код цифри 1. Але є одна особливість – DDRAM пам'ять набагато більша ніж видима область екрана. Як правило, DDRAM містить 80 осередків – 40 у першому рядку й 40 у другому; на дисплеї може рухатися по лінійці осередків. Наприклад, можна записати в DDRAM відразу п'ять пунктів меню, а потім просто показувати по одному пункті. Для переміщення дисплея є спец команда. Також є поняття курсору – це місце в яке буде записаний наступний символ, тобто поточне значення лічильника адреси. Курсор не обов'язково може бути на екрані, він може розташовуватися й за екраном або бути відключений зовсім.

# Елементи індикації

Адреса байта в DDRAM



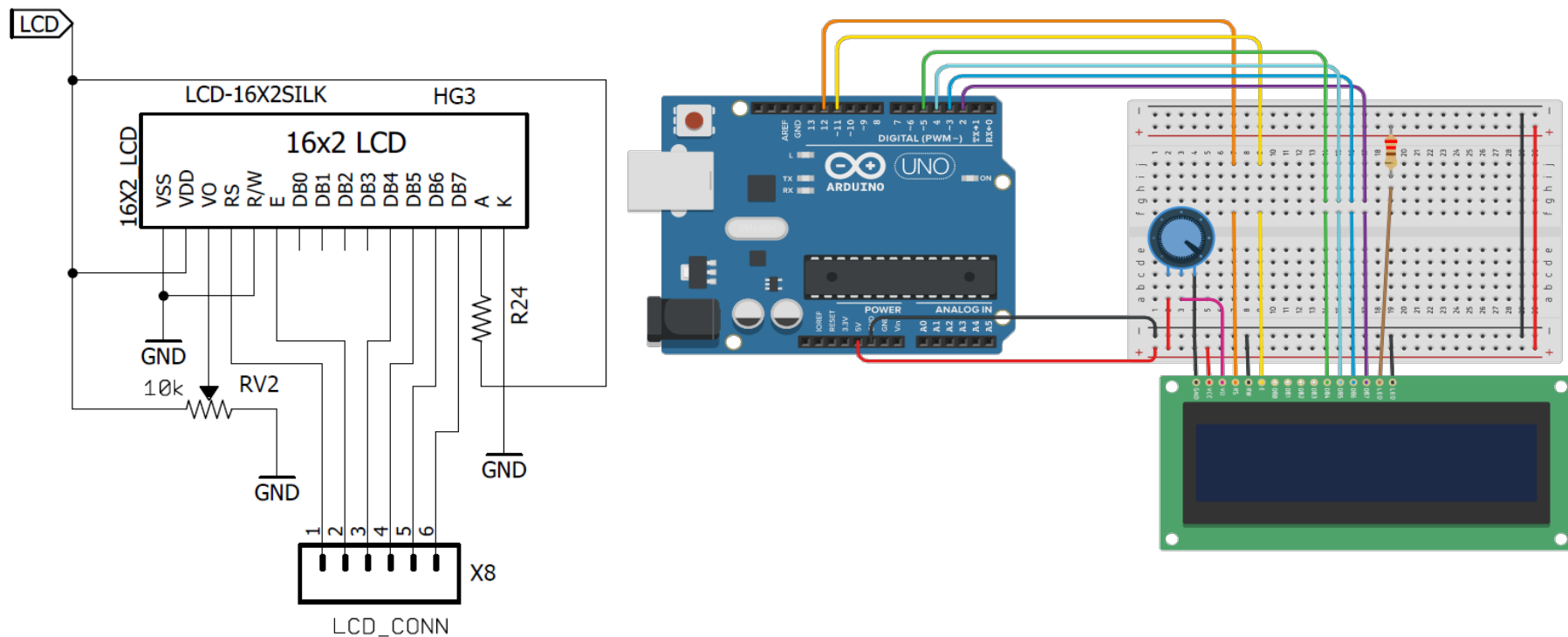
Видима й прихована область екранної пам'яті

CGROM – таблиця символів. Коли записуємо в осередок DDRAM байт, то з таблиці береться символ і рисується на екрані. CGROM не можна змінити, тому важливо, щоб вона мала англійські або російські букви.

CGRAM – теж таблиця символів, але її можна міняти, створюючи свої символи. Адресується вона лінійно, тобто спочатку йде 8 байт одного символу, як наведено на рис.

Усього в CGRAM може бути 8 символів, відповідно CGRAM має 64 байт пам'яті. Ці символи мають коди від 0x00 до 0x07. Так що, записавши, наприклад, у перші 8 байт CGRAM (перший символ з кодом 00) будь який символ, і записавши в DDRAM нуль (код першого символу в CGRAM) побачимо на екрані цей символ. Командою вибирається в яку саме пам'ять і починаючи з якої адреси будуть записуватись дані.

# LCD індикатор



Для відображення даних на індикаторі необхідно підключити виводи RS, E, DB4-DB7 (з'єднувач X8) до Arduino та подати живлення на індикатор.

# LCD індикатор

Бібліотека `LiquidCrystal` дозволяє Arduino управляти різними LCD дисплеями, побудованими на базі поширеного чіпсета Hitachi HD44780 (або сумісного). У бібліотеці реалізований як 4-х, так і 8-бітний режим роботи. Бібліотека має такі функції: `LiquidCrystal()`, `begin()`, `clear()`, `home()`, `setCursor()`, `write()`, `print()`, `cursor()`, `noCursor()`, `blink()`, `noBlink()`, `display()`, `noDisplay()`, `scrollDisplayLeft()`, `scrollDisplayRight()`, `autoscroll()`, `noAutoscroll()`, `leftToRight()`, `rightToLeft()`, `createChar()`.

`LiquidCrystal (rs, rw, enable, d0, d1, d2, d3, d4, d5, d6, d7)` задає конфігурацію підключення та режим роботи, де `rs`, `rw`, `enable`: номер виводу Arduino, з'єднаного з виводом RS, RW, E LCD-індикатора; `d0`, `d1`, `d2`, `d3`, `d4`, `d5`, `d6`, `d7`: номери виводів Arduino, які підключені до відповідних цифрових виводів LCD-індикатора. Параметри `rw`, `d0`, `d1`, `d2` і `d3` є не обов'язковими. Якщо `d0`, `d1`, `d2` і `d3` не вказані, то LCD буде працювати у 4-х бітному режимі (`d4`, `d5`, `d6`, `d7`).

```
LiquidCrystal(rs, enable, d4, d5, d6, d7)
LiquidCrystal(rs, rw, enable, d4, d5, d6, d7)
LiquidCrystal(rs, enable, d0, d1, d2, d3, d4, d5, d6, d7)
LiquidCrystal(rs, rw, enable, d0, d1, d2, d3, d4, d5, d6, d7)
```

```
#include <LiquidCrystal.h>
LiquidCrystal lcd(12, 11, 10, 5, 4, 3, 2);
void setup(){
    lcd.begin(16,1);
    lcd.print("hello, world!");
}
void loop() {}
```

# LCD індикатор

**lcd.begin (cols, rows)** ініціалізує інтерфейс для взаємодії з LCD-індикатором та задає розміри (ширину і висоту) області виведення екрану, де lcd: змінна типу LiquidCrystal, cols: кількість стовпців екрану, rows: кількість рядків екрану. При роботі з LCD-дисплеєм, функція begin() повинна викликатися першою і передувати іншим командам з бібліотеки LiquidCrystal.

**lcd.clear ()** очищає LCD-екран і переміщує курсор в лівий верхній кут.

**lcd.home ()** переміщує курсор в лівий верхній кут екрану (наступний текст буде виводиться з початку екрану).

**lcd.setCursor (col, row)** встановлює позицію, в якій буде виводитися наступний текст, де col: координата X позиції курсора (0 означає перший стовпець); row: координата Y позиції курсора (0 означає перший рядок).

**lcd.write (data)** виводить символ на LCD-індикатор, де data: символ, який необхідно вивести на екран.

**lcd.print(data) / lcd.print(data, BASE)** виводить текст на LCD-індикатор, де data: дані, які необхідно вивести (тип char, byte, int, long або string); BASE (не обов'язковий параметр): основа системи числення, в якій необхідно виводити числа: BIN – двійкова, DEC – десяткова, OCT – вісімкова, HEX – шістнадцятькова.

**lcd.cursor() / lcd.noCursor ()** показує/ не показує на LCD-екрані курсор: символ підкреслення в тій позиції, куди буде виведений наступний символ.

**lcd.blink() / lcd.noBlink ()** включає / відключає на LCD-індикаторі курсор, що мигає.

**lcd.noDisplay() / lcd.display()** вимикає / вмикає LCD-екран. Текст, якщо не відображається на екрані, зберігається в пам'яті.

# LCD індикатор

**lcd.scrollDisplayLeft()** / **lcd.scrollDisplayRight()** здійснює прокручування вмісту дисплея (весь текст і курсор) на один символ ліворуч / праворуч.

**lcd.autoscroll()** включає автоматичну прокрутку тексту на LCD. Це означає, що при виведенні кожного нового символу, всі попередні символи будуть зсуватись на одну позицію. Якщо встановлений режим перегляду тексту зліва-направо (за замовчуванням), то прокрутка буде здійснюватися ліворуч; якщо встановлений режим зправа-наліво, то прокрутка буде здійснюватися праворуч. Таким чином, кожен новий символ буде виводиться в одній і тій же позиції LCD.

**lcd.noAutoscroll()** функція відключає автоматичну прокрутку тексту в LCD.

**lcd.leftToRight()** / **lcd.rightToLeft()** встановлює режим перегляду тексту на LCD зліва-направо (режим за замовчуванням) / зправа-наліво.

**lcd.createChar (num, data)** створює символ користувача для LCD-індикатора. Дисплей підтримує до 8 символів користувача (пронумерованих від 0 до 7) розміром 5x8 пікселів. Зовнішній вигляд кожного символу користувача задається масивом з восьми байт, кожен з яких характеризує відповідний рядок. П'ять молодших біт кожного байту визначають стан пікселів у відповідному рядку. Для того, щоб вивести певний символ користувача, використовується функцію **write()** з його номером, де **num**: номер призначеного символу користувача, який необхідно створити (від 0 до 7); **data**: дані у пікселях символу користувача

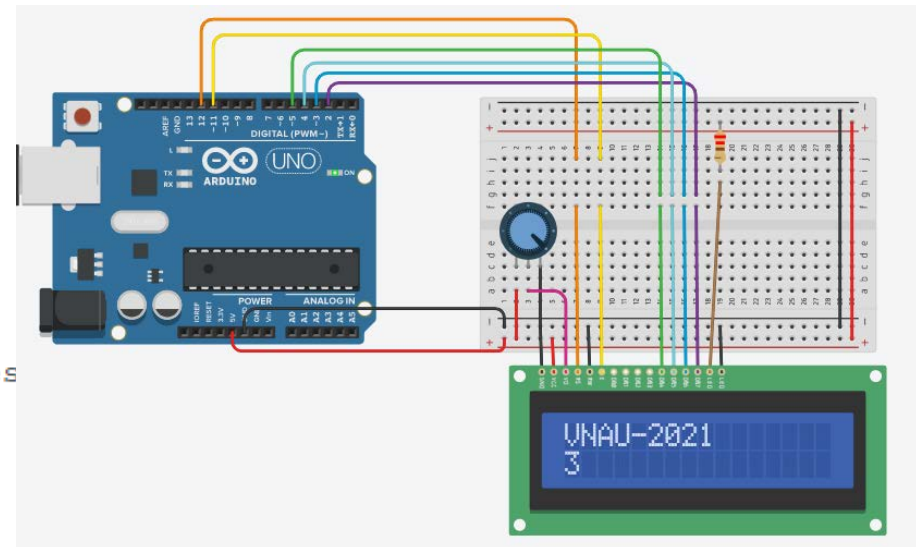
```
#include <LiquidCrystal.h>
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
byte smiley[8] = {
    B00000,
    B10001,
    B00000,
    B00000,
    B10001,
    B01110,
    B00000,
};
void setup() {
    lcd.createChar(0, smiley);
    lcd.begin(16, 2);
    lcd.write(byte(0));
}
void loop() {}
```



# LCD індикатор

Приклад застосування команд `LiquidCrystal(rs, enable, d4, d5, d6, d7)`, `lcd.begin (cols, rows)`, `lcd.print(data)`, `lcd.setCursor (col, row)`

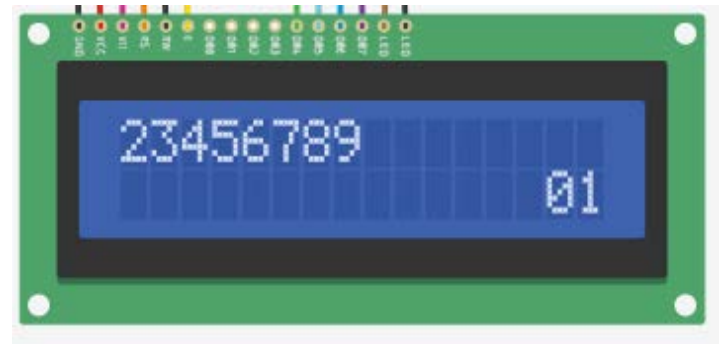
```
1 //Arduino pins RS, E, D4, D5, D6, D7
2 #include <LiquidCrystal.h>
3
4 // initialize the library with the numbers of the interface pins
5 LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
6
7 void setup() {
8   // set up the LCD's number of columns and rows:
9   lcd.begin(16, 2);
10  // Print a message to the LCD.
11  lcd.print("VNAU-2021");
12 }
13
14 void loop() {
15   // set the cursor to column 0, line 1
16   lcd.setCursor(0, 1);
17   // print the number of seconds since reset
18   lcd.print(millis() / 1000);
19
20 }
```



# LCD індикатор

```
1 #include <LiquidCrystal.h>
2
3 LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
4
5 void setup() {
6   lcd.begin(16, 2);
7 }
8
9 void loop() {
10  lcd.setCursor(0, 0);
11  // виводимо цифри від 0 до 9:
12  for (int thisChar = 0; thisChar < 10; thisChar++) {
13    lcd.print(thisChar);
14    delay(500);
15  }
16
17  lcd.setCursor(16, 1);
18  // включається автоматична прокрутка
19  lcd.autoscroll();
20  for (int thisChar = 0; thisChar < 10; thisChar++) {
21    lcd.print(thisChar);
22    delay(500);
23  }
24  // виключається автоматична прокрутка
25  lcd.noAutoscroll();
26
27  // очистка екрану
28  lcd.clear();
29 }
```

Приклад застосування команд  
LiquidCrystal(rs, enable, d4, d5, d6, d7),  
lcd.begin (cols, rows), lcd.print(data),  
lcd.setCursor (col, row), **lcd.autoscroll()**,  
**lcd.noAutoscroll()**, **lcd.clear()**



# LCD індикатор

```
1 #include <LiquidCrystal.h>
2 LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
3
4 void setup() {
5   lcd.begin(16, 2);
6   lcd.print("hello, world!");
7 }
8
9 void loop() {
10  lcd.noBlink();
11  delay(3000);
12  lcd.blink();
13  delay(3000);
14 }
```

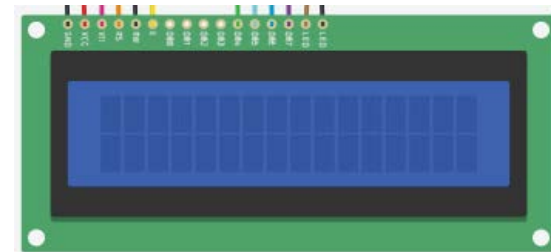
Приклад застосування команд  
LiquidCrystal(rs, enable, d4, d5, d6, d7),  
lcd.begin (cols, rows), lcd.print(data),  
lcd.blink(), lcd.noBlink()



# LCD індикатор

```
1 #include <LiquidCrystal.h>
2 LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
3
4 void setup() {
5   lcd.begin(16, 2);
6   lcd.print("hello, world!");
7 }
8
9 void loop() {
10  lcd.noBlink();
11  delay(3000);
12  lcd.blink();
13  delay(3000);
14 }
```

Приклад застосування команд  
LiquidCrystal(rs, enable, d4, d5, d6, d7),  
lcd.begin (cols, rows), lcd.print(data),  
lcd.display(), lcd.noDisplay



# LCD індикатор

Приклад застосування команд  
LiquidCrystal(rs, enable, d4, d5, d6, d7),  
lcd.begin (cols, rows), lcd.print(data),  
lcd.createChar (num, data),  
lcd.write(byte(0))

```
1 #include <LiquidCrystal.h>
2 LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
3 byte heart[8] = {
4   0b000000,
5   0b010101,
6   0b111111,
7   0b111111,
8   0b111111,
9   0b011110,
10  0b001000,
11  0b000000
12 };
13 byte smiley[8] = {
14   0b000000,
15   0b000000,
16   0b010101,
17   0b000000,
18   0b000000,
19   0b100011,
20   0b011110,
21   0b000000
22 };
23 byte frownie[8] = {
24   0b000000,
25   0b000000,
26   0b010101,
27   0b000000,
28   0b000000,
29   0b000000,
30   0b011110,
31   0b100011
32 };
33 byte armsDown[8] = {
34   0b001001,
35   0b010101,
36   0b001001,
37   0b001001,
38   0b011110,
39   0b101011,
40   0b001001,
41   0b010101
42 };
43 byte armsUp[8] = {
44   0b001001,
45   0b010101,
46   0b001001,
47   0b101011,
48   0b011110,
49   0b001001,
50   0b001001,
51   0b010101
52 };
53 void setup() {
54   lcd.begin(16, 2);
55   lcd.createChar(0, heart);
56   lcd.createChar(1, smiley);
57   lcd.createChar(2, frownie);
58   lcd.createChar(3, armsDown);
59   lcd.createChar(4, armsUp);
60   lcd.setCursor(0, 0);
61   lcd.print("I ");
62   lcd.write(byte(0));
63   lcd.print(" Arduino! ");
64   lcd.write(byte(1));
65 }
66 void loop() {
67   // read the potentiometer on A0:
68   int sensorReading = analogRead(A0);
69   // map the result to 200 - 1000:
70   int delayTime = map(sensorReading, 0, 1023, 200, 1000);
71   // set the cursor to the bottom row, 5th position:
72   lcd.setCursor(4, 1);
73   lcd.write(3); // draw the little man, arms down
74   delay(delayTime);
75   lcd.setCursor(4, 1);
76   lcd.write(4); // draw him arms up:
77   delay(delayTime);
78 }
79
80
```



# LCD індикатор

```
1 #include <LiquidCrystal.h>
2 LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
3 void setup() {
4   lcd.begin(16, 2);
5   lcd.print("hello, world!");
6   delay(1000);
7 }
8
9 void loop() {
10  // зсув на 13 позицій (довжина рядка) ліворуч
11  for (int position = 0; position < 13; position++) {
12    // зсув на одну позицію
13    lcd.scrollDisplayLeft();
14    delay(150);
15  }
16
17  // зсув на 29 позицій праворуч (довж. рядка + довж. індик.)
18  for (int position = 0; position < 29; position++) {
19    // зсув на одну позицію праворуч
20    lcd.scrollDisplayRight();
21    delay(150);
22  }
23
24  // зсув на 16 позицій (довж. рядка + довж. індик) ліворуч
25  // щоб повернути текст до центру
26  for (int position = 0; position < 16; position++) {
27    lcd.scrollDisplayLeft();
28    delay(150);
29  }
30  delay(1000);
31 }
```

Приклад застосування команд LiquidCrystal (rs, enable, d4, d5, d6, d7), lcd.begin (cols, rows), lcd.print(data), lcd.scrollDisplayRight(), lcd.scrollDisplayLeft()



# Індивідуальне завдання

1. Реалізувати програму, яка виводить прізвище, ім'я та по батькові у вигляді рядка на LCD-індикаторі, що «біжить».
2. Ввести зміни до попередньої програми, щоб виводилась ще поточна дата.
3. Реалізувати програму виведення випадкового числа на LCD-індикаторі за натисненням кнопки.
4. Реалізувати програму, яка відображає на LCD-індикаторі напис «LOAD». За натисненням кнопки S1 відображається напис «PLAY», а S2 – «STOP».

