

**МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ  
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ “ЛЬВІВСЬКА ПОЛІТЕХНІКА”**

**Кафедра САП**



**ЗВІТ**

до виконання лабораторної роботи №10

На тему: " Робота з матричним світлодіодним індикатором"  
з курсу "Мікропроцесорні системи "

Варіант - 4

**Виконав:**

Студент гр.ПП-31

Гаврилюк Н. О.

**Прийняв:**

Доцент кафедри САП

Головатий А.І.

**ЛЬВІВ - 2025**

**Мета:** ознайомитись з принципом роботи матричного світлодіодного індикатора 8×8 точок, драйвера керування роботою світлодіодною матрицею MAX7219; навчитись програмувати виведення інформації через інтерфейс SPI на матричний світлодіодний індикатор з використанням драйвера керування MAX7219.

## Завдання

1. Реалізувати програму, яка передбачає використання світлодіодної матриці.

Засвічуються стовпці матриці справа-наліво, потім рядки згори- донизу, створюючи ефект роботи сканера.

Код програми Arduino IDE:

```
#include "LedControl.h"
```

```
// DIN, CLK, CS пін підключення
```

```
LedControl lc = LedControl(12, 11, 10, 1); // 1 матриця
```

```
void setup() {
```

```
    lc.shutdown(0, false); // увімкнути матрицю
```

```
    lc.setIntensity(0, 8); // яскравість
```

```
    lc.clearDisplay(0); // очистити матрицю
```

```
}
```

```
void loop() {
```

```
    // Сканування стовпців справа-наліво
```

```
    for (int col = 7; col >= 0; col--) {
```

```
        for (int row = 0; row < 8; row++) {
```

```
            lc.setLed(0, row, col, true);
```

```
        }
```

```
        delay(200);
```

```
        for (int row = 0; row < 8; row++) {
```

```
            lc.setLed(0, row, col, false);
```

```
        }
```

}

```
// Сканування рядків згори-донизу
```

```
for (int row = 0; row < 8; row++) {
```

```
    for (int col = 0; col < 8; col++) {
```

```
        lc.setLed(0, row, col, true);
```

```
}
```

```
delay(200);
```

```
    for (int col = 0; col < 8; col++) {
```

```
        lc.setLed(0, row, col, false);
```

```
}
```

```
}
```

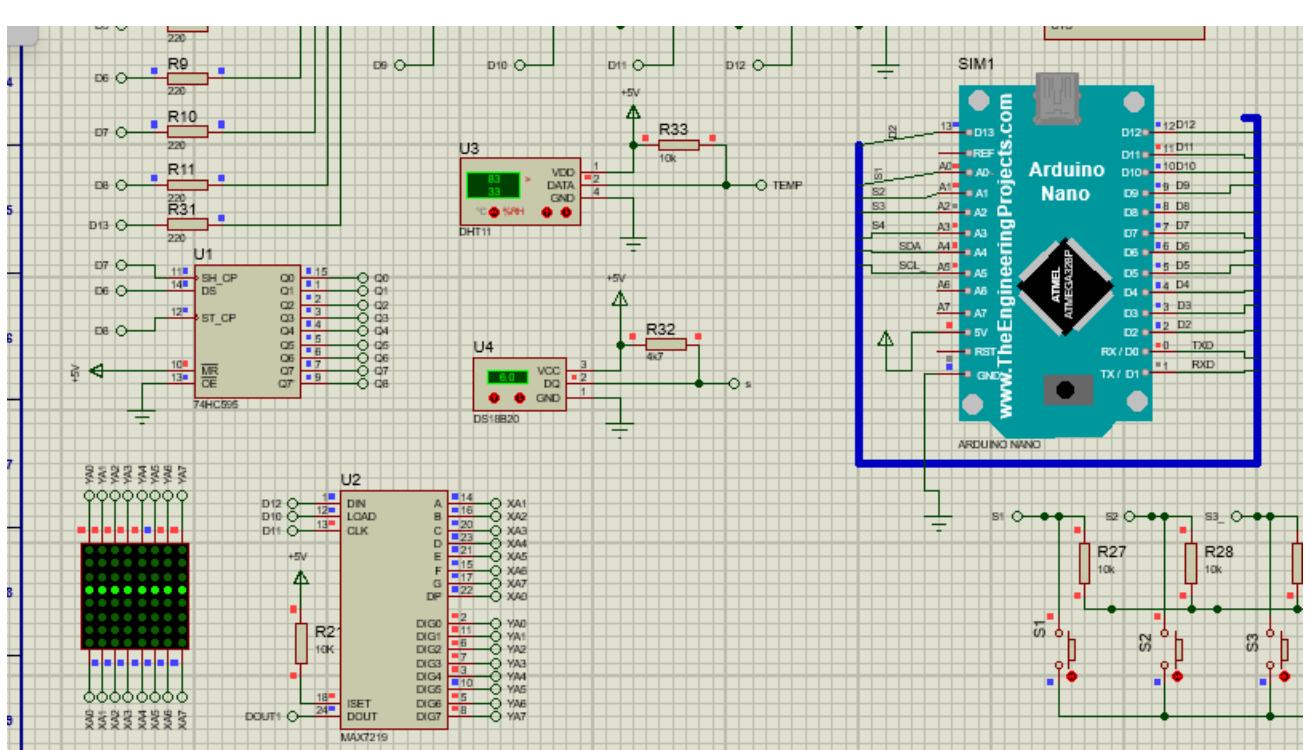


Рис. 1. Результат програми

Код програми MK AVR:

```
#define F_CPU 16000000UL
```

```
#include <avr/io.h>
```

```

#include <util/delay.h>
#include <stdint.h>

// Піни для MAX7219
#define DIN PB0
#define CLK PB1
#define CS PB2

uint8_t row_data[8]; // стан рядків

// Надсилання одного байта на MAX7219
void max7219_send(uint8_t data) {
    for(int8_t i=7; i>=0; i--) {
        if(data & (1<<i)) PORTB |= (1<<DIN);
        else PORTB &= ~(1<<DIN);

        PORTB |= (1<<CLK);
        PORTB &= ~(1<<CLK);
    }
}

// Надсилання команди MAX7219: реєстр + дані
void max7219_write(uint8_t reg, uint8_t data) {
    PORTB &= ~(1<<CS);      // CS low
    max7219_send(reg);
    max7219_send(data);
    PORTB |= (1<<CS);      // CS high
}

```

```

// Ініціалізація MAX7219

void max7219_init() {
    DDRB |= (1<<DIN)|(1<<CLK)|(1<<CS);

    max7219_write(0x09, 0x00); // decode mode: none
    max7219_write(0x0A, 0x08); // intensity: середня
    max7219_write(0x0B, 0x07); // scan limit: всі рядки
    max7219_write(0x0C, 0x01); // нормальній режим
    max7219_write(0x0F, 0x00); // display test off

    // Очистити дисплей
    for(uint8_t i=0;i<8;i++){
        row_data[i]=0;
        max7219_write(i+1,0);
    }
}

// Засвічує один стовпець
void set_column(uint8_t col) {
    for(uint8_t i=0;i<8;i++){
        row_data[i] = (1 << (7-col)); // дзеркально
        max7219_write(i+1, row_data[i]);
    }
}

// Засвічує один рядок
void set_row(uint8_t row) {

```

```
for(uint8_t i=0;i<8;i++){  
    row_data[i]=0;  
}  
  
row_data[row]=0xFF; // весь рядок  
  
for(uint8_t i=0;i<8;i++){  
    max7219_write(i+1, row_data[i]);  
}  
}
```

```
int main(void) {  
    max7219_init();  
    _delay_ms(500);  
  
    while(1) {  
        // Сканування стовпців справа-наліво  
        for(uint8_t col=0; col<8; col++){  
            set_column(col);  
            _delay_ms(150);  
        }  
  
        // Сканування рядків згори-донизу  
        for(uint8_t row=0; row<8; row++){  
            set_row(row);  
            _delay_ms(150);  
        }  
    }  
}
```

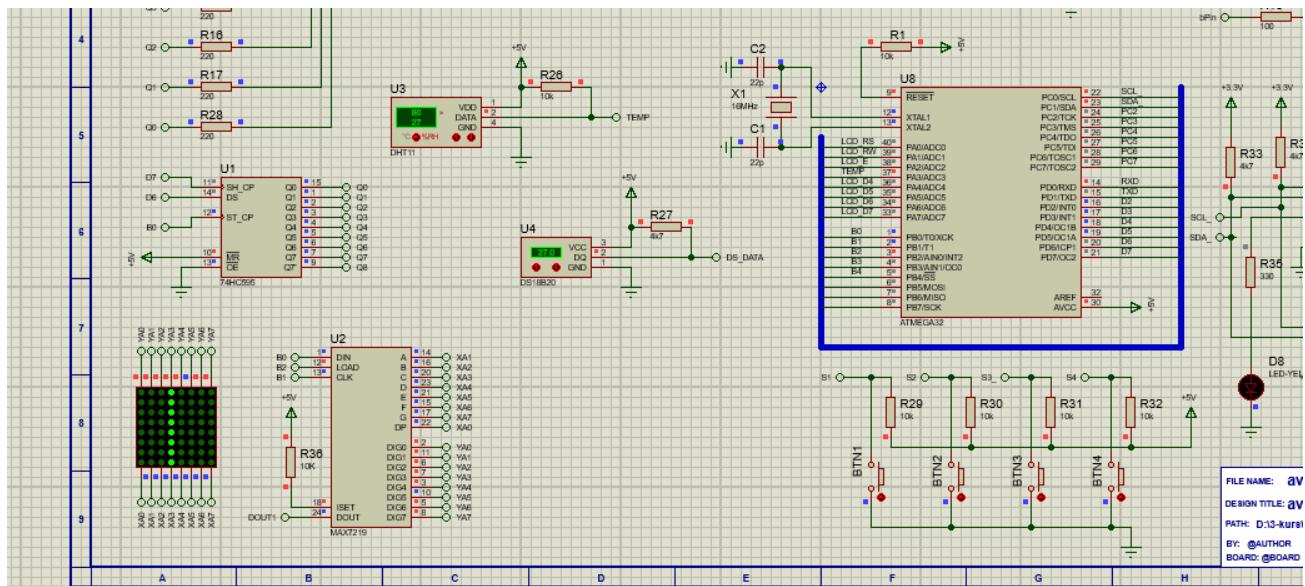


Рис. 2. Результат программи

2. Реалізувати програму, яка передбачає використання світлодіодної матриці. Засвічується крапка, яка «бігає» по контуру матриці.

Код програми Arduino IDE:

```
#include "LedControl.h"
```

```
// DIN, CLK, CS
LedControl lc = LedControl(12, 11, 10, 1);
```

```
void setup() {
    lc.shutdown(0, false); // увімкнути матрицю
    lc.setIntensity(0, 8); // яскравість
    lc.clearDisplay(0); // очистити матрицю
}
```

```
void loop() {
    // верхній рядок
    for (int col = 0; col < 8; col++) {
        lc.clearDisplay(0);
        lc.setLed(0, 0, col, true);
```

```
delay(200);

}

// правий стовпець

for (int row = 1; row < 8; row++) {

lc.clearDisplay(0);

lc.setLed(0, row, 7, true);

delay(200);

}

// нижній рядок

for (int col = 6; col >= 0; col--) {

lc.clearDisplay(0);

lc.setLed(0, 7, col, true);

delay(200);

}

// лівий стовпець

for (int row = 6; row > 0; row--) {

lc.clearDisplay(0);

lc.setLed(0, row, 0, true);

delay(200);

}

}
```

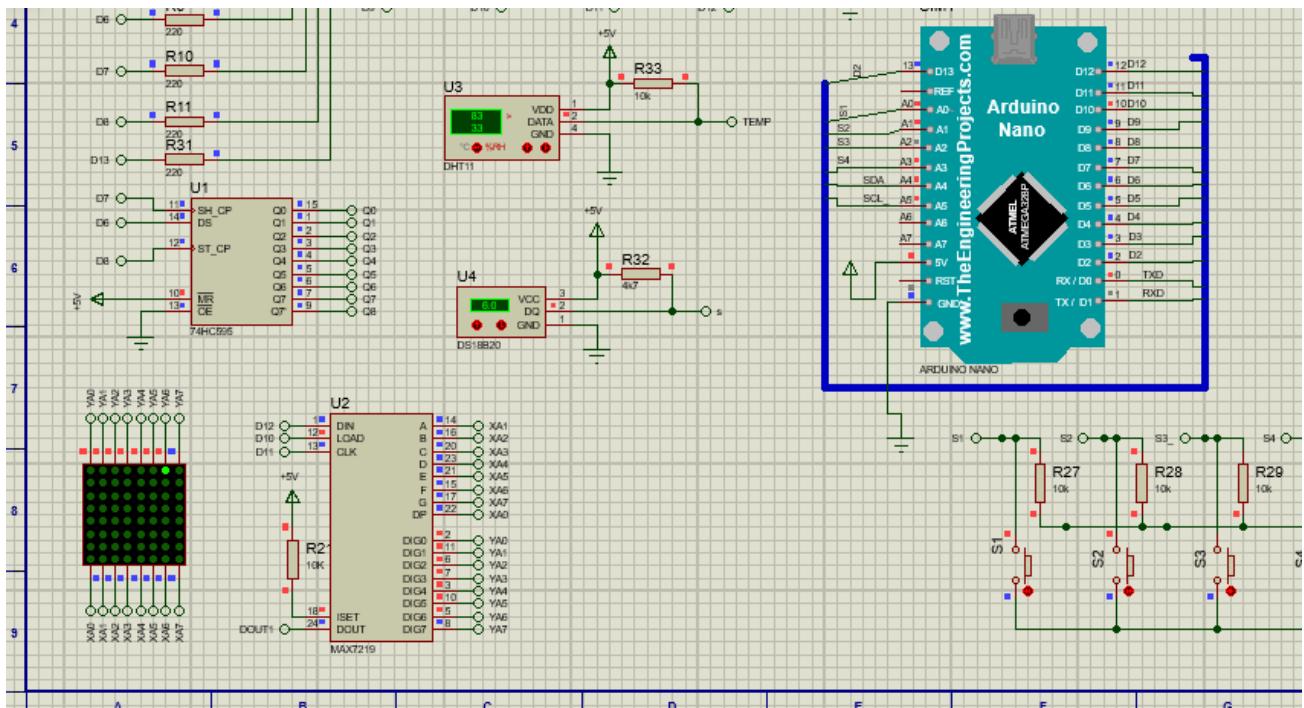


Рис. 3. Результат програми

Код програми MK AVR:

```
#define F_CPU 16000000UL
```

```
#include <avr/io.h>
```

```
#include <util/delay.h>
```

```
#include <stdint.h>
```

```
// Піни для MAX7219
```

```
#define DIN PB0
```

```
#define CLK PB1
```

```
#define CS PB2
```

```
uint8_t row_data[8]; // стан рядків
```

```
// Надсилання одного байта на MAX7219
```

```
void max7219_send(uint8_t data) {
```

```
for(int8_t i=7; i>=0; i--) {
```

```
if(data & (1<<i)) PORTB |= (1<<DIN);
```

```
    else PORTB &= ~(1<<DIN);

    PORTB |= (1<<CLK);
    PORTB &= ~(1<<CLK);

}

}
```

// Надсилання команди MAX7219: реєстр + дані

```
void max7219_write(uint8_t reg, uint8_t data) {
    PORTB &= ~(1<<CS);      // CS low
    max7219_send(reg);
    max7219_send(data);
    PORTB |= (1<<CS);      // CS high
}
```

// Ініціалізація MAX7219

```
void max7219_init() {
    DDRB |= (1<<DIN)|(1<<CLK)|(1<<CS);

    max7219_write(0x09, 0x00); // decode mode: none
    max7219_write(0x0A, 0x0F); // intensity: max
    max7219_write(0x0B, 0x07); // scan limit: all rows
    max7219_write(0x0C, 0x01); // shutdown register: normal operation
    max7219_write(0x0F, 0x00); // display test: off
```

// Очистити всі рядки

```
for(uint8_t i=0;i<8;i++){
    row_data[i]=0;
```

```

        max7219_write(i+1,0);

    }

}

// Встановлює один піксель у рядку, очищаючи старий

void set_pixel(uint8_t row, uint8_t col) {
    for(uint8_t i=0;i<8;i++) row_data[i]=0; // очищаємо всі рядки
    row_data[row] = 1 << (7-col);      // біт для колонки (дзеркально)
    for(uint8_t i=0;i<8;i++)
        max7219_write(i+1, row_data[i]);
}

int main(void) {
    max7219_init();
    _delay_ms(1000);

    while(1) {
        // Верхній рядок (ліворуч -> праворуч)
        for(uint8_t col=0; col<8; col++){
            set_pixel(0,col);
            _delay_ms(200);
        }
        // Правий стовпець (зверху -> вниз)
        for(uint8_t row=1; row<8; row++){
            set_pixel(row,7);
            _delay_ms(200);
        }
        // Нижній рядок (праворуч -> ліворуч)
    }
}

```

```

for(int8_t col=6; col>=0; col--){

    set_pixel(7,col);

    _delay_ms(200);

}

// Лівий стовпець (знизу -> вверх)

for(int8_t row=6; row>0; row--){

    set_pixel(row,0);

    _delay_ms(200);

}

}

```

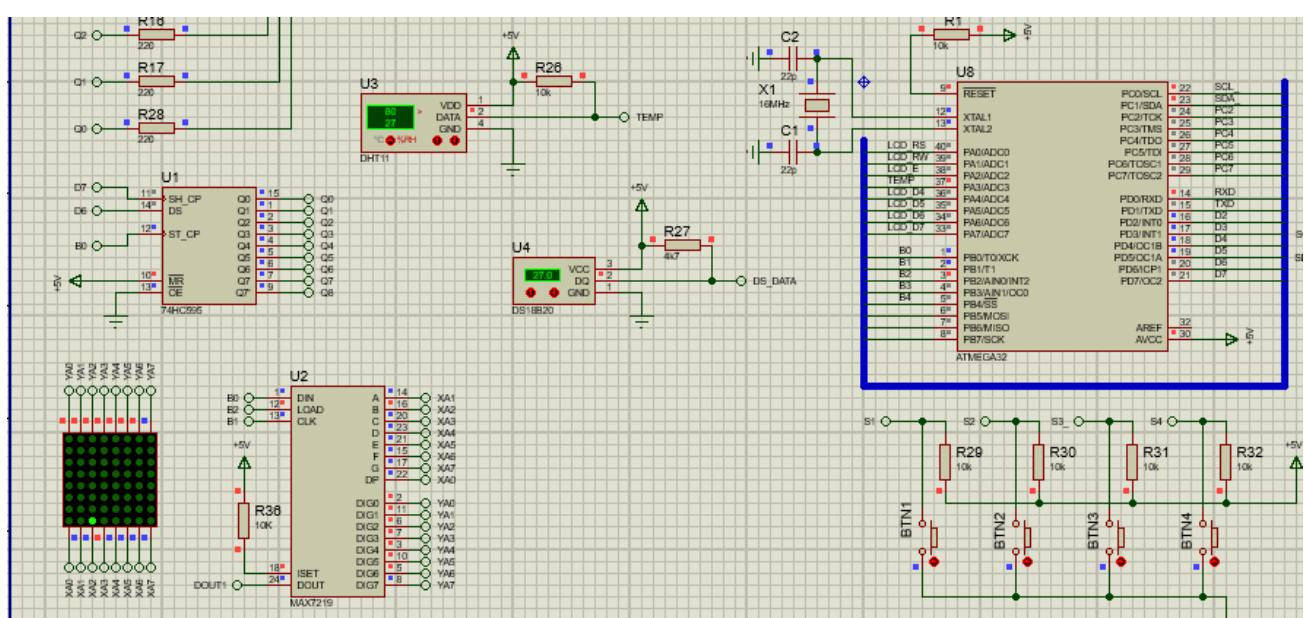


Рис. 4. Результат програми

**Висновок:** У ході виконання лабораторної роботи я ознайомився з принципом роботи матричного світлодіодного індикатора 8×8 точок, драйвером керування роботою світлодіодною матрицею MAX7219; навчився програмувати виведення інформації через інтерфейс SPI на матричний світлодіодний індикатор з використанням драйвера керування MAX7219.