

Додаток Д

Програма SEG1

Програма демонструє роботу з 4 позиційним семисегментним індикатором з загальним катодом. Алгоритм програми простий – на індикатор виводиться число 0123 в режимі динамічної індикації з частотою 100Гц (25 Гц на кожному позицію)

Лістинг програми SEG1

```
// Pin 2-8 is connected to the 7 segments of the display.
int pinA = 2;
int pinB = 3;
int pinC = 4;
int pinD = 5;
int pinE = 6;
int pinF = 7;
int pinG = 8;
int D1 = 9;
int D2 = 10;
int D3 = 11;
int D4 = 12;

void setup() {
// initialize the digital pins as outputs.
  pinMode(pinA, OUTPUT);
  pinMode(pinB, OUTPUT);
  pinMode(pinC, OUTPUT);
  pinMode(pinD, OUTPUT);
  pinMode(pinE, OUTPUT);
  pinMode(pinF, OUTPUT);
  pinMode(pinG, OUTPUT);
  pinMode(D1, OUTPUT);
  pinMode(D2, OUTPUT);
  pinMode(D3, OUTPUT);
  pinMode(D4, OUTPUT);
}

void loop() {
  digitalWrite(D1, HIGH);
  digitalWrite(D2, LOW);
  digitalWrite(D3, LOW);
  digitalWrite(D4, LOW);
  //0
  digitalWrite(pinA, HIGH);
  digitalWrite(pinB, HIGH);
  digitalWrite(pinC, HIGH);
  digitalWrite(pinD, HIGH);
  digitalWrite(pinE, HIGH);
  digitalWrite(pinF, HIGH);
  digitalWrite(pinG, LOW);
  delay(10);
```

```

digitalWrite(D1, LOW);
digitalWrite(D2, HIGH);
digitalWrite(D3, LOW);
digitalWrite(D4, LOW);
//1
digitalWrite(pinA, LOW);
digitalWrite(pinB, HIGH);
digitalWrite(pinC, HIGH);
digitalWrite(pinD, LOW);
digitalWrite(pinE, LOW);
digitalWrite(pinF, LOW);
digitalWrite(pinG, LOW);
delay(10);

digitalWrite(D1, LOW);
digitalWrite(D2, LOW);
digitalWrite(D3, HIGH);
digitalWrite(D4, LOW);
//2
digitalWrite(pinA, HIGH);
digitalWrite(pinB, HIGH);
digitalWrite(pinC, LOW);
digitalWrite(pinD, HIGH);
digitalWrite(pinE, HIGH);
digitalWrite(pinF, LOW);
digitalWrite(pinG, HIGH);
delay(10);

digitalWrite(D1, LOW);
digitalWrite(D2, LOW);
digitalWrite(D3, LOW);
digitalWrite(D4, HIGH);
//3
digitalWrite(pinA, HIGH);
digitalWrite(pinB, HIGH);
digitalWrite(pinC, HIGH);
digitalWrite(pinD, HIGH);
digitalWrite(pinE, LOW);
digitalWrite(pinF, LOW);
digitalWrite(pinG, HIGH);
delay(10);
/*
//4
digitalWrite(pinA, HIGH);
digitalWrite(pinB, LOW);
digitalWrite(pinC, LOW);
digitalWrite(pinD, HIGH);
digitalWrite(pinE, HIGH);
digitalWrite(pinF, LOW);
digitalWrite(pinG, LOW);
delay(10);

//5
digitalWrite(pinA, LOW);

```

```

digitalWrite(pinB, HIGH);
digitalWrite(pinC, LOW);
digitalWrite(pinD, LOW);
digitalWrite(pinE, HIGH);
digitalWrite(pinF, LOW);
digitalWrite(pinG, LOW);
delay(10);

//6
digitalWrite(pinA, LOW);
digitalWrite(pinB, HIGH);
digitalWrite(pinC, LOW);
digitalWrite(pinD, LOW);
digitalWrite(pinE, LOW);
digitalWrite(pinF, LOW);
digitalWrite(pinG, LOW);
delay(10);

//7
digitalWrite(pinA, LOW);
digitalWrite(pinB, LOW);
digitalWrite(pinC, LOW);
digitalWrite(pinD, HIGH);
digitalWrite(pinE, HIGH);
digitalWrite(pinF, HIGH);
digitalWrite(pinG, HIGH);
delay(10);

//8
digitalWrite(pinA, LOW);
digitalWrite(pinB, LOW);
digitalWrite(pinC, LOW);
digitalWrite(pinD, LOW);
digitalWrite(pinE, LOW);
digitalWrite(pinF, LOW);
digitalWrite(pinG, LOW);
delay(10);

//9
digitalWrite(pinA, LOW);
digitalWrite(pinB, LOW);
digitalWrite(pinC, LOW);
digitalWrite(pinD, HIGH);
digitalWrite(pinE, HIGH);
digitalWrite(pinF, LOW);
digitalWrite(pinG, LOW);
delay(10);
*/
}

```

Програма SEG2

Програма демонструє роботу з 4 позиційним семисегментним індикатором з загальним катодом та регістрами портів Arduino. Алгоритм програми простий

– на індикатор виводиться число 0123 в режимі динамічної індикації з частотою 200Гц (50 Гц на кожну позицію)

Лістинг програми SEG2

```
/*
    A
    ---
F |   | B
  | G |
  ---
E |   | C
  |   |
  ---
    D
A-PD2, B-PD3, C-PD4, D- PD5, E-PD6, F-PD7
G-PB0, D1-PB1, D2-PB2, D3-PB3, D4-PB4
*/
#define P0 B00000010;
#define P1 B00000100;
#define P2 B00001000;
#define P3 B00010000;

void setup() {
    DDRD=0xFF;
    DDRB=0xFF;
    PORTD =0x00;
    PORTB = 0<<0;
}
void loop() {
    PORTD=B11111100; //0
    PORTB=P0;
    delay (5);

    PORTD=B00011000;//1
    PORTB=P1;
    delay (5);

    PORTD=B01101100;//2
    PORTB=P2;
    PORTB|=1; //PB0=1
    delay (5);

    PORTD=B00111100;//3
    PORTB=P3;
    PORTB|=1; //PB0=1
    delay (5);
}
```

Програма SEG3

Програма демонструє роботу з 4 позиційним семисегментним індикатором з загальним катодом та регістрами портів Arduino. У програмі використовується оператор SWITCH та цикл FOR для вибору позиції та цифри та бітові операції

bitSet, bitClear для установки / скидання сегменту G індикатора. Алгоритм програми простий – у кожену позицію індикатора по черзі виводиться цифра від 0 до 9 з затримкою 0,5 секунди.

Лістинг програми SEG3

```
//A-PD2, B-PD3, C-PD4, D-PD5, E-PD6, F-PD7
//G-PB0, D1-PB1, D2-PB2, D3-PB3, D4-PB4

# define P0 B00000010;
# define P1 B00000100;
# define P2 B00001000;
# define P3 B00010000;
int tmp=0;

void setup() {
    DDRD=0xFF;
    DDRB=0xFF;
    PORTD =0x00;
    PORTB = 0x00;
}

void loop() {
    for (int j=0; j<=3; j++){
        for (int i=0; i<10; i++){
            Cifra(i);
            delay(500);
        }
        switch (j){
            case 0:
                PORTB=P0;
                break;
            case 1:
                PORTB=P1;
                break;
            case 2:
                PORTB=P2;
                break;
            case 3:
                PORTB=P3;
                break;
            default:
                break;
        }
    }
}

void Cifra(int x){
    switch (x){
        case 0:
            PORTD=0xFC;
            break;
        case 1:
            PORTD=0x18;
            break;
    }
}
```

```

    case 2:
        PORTD=0x6C;
        bitSet(PORTB,0);
        break;
    case 3:
        PORTD=0x3C;
        bitSet(PORTB,0);
        break;
    case 4:
        PORTD=0x98;
        bitSet(PORTB,0);
        PORTB|=1;
        break;
    case 5:
        PORTD=0xB4;
        bitSet(PORTB,0);
        break;
    case 6:
        PORTD=0xF4;
        bitSet(PORTB,0);
        break;
    case 7:
        PORTD=0x1C;
        bitClear(PORTB,0);
        break;
    case 8:
        PORTD=0xFC;
        bitSet(PORTB,0);
        break;
    case 9:
        PORTD=0xBC;
        bitSet(PORTB,0);
        break;
    default:
        break;
}
}

```

Програма SEG4

Програма демонструє роботу з 4 позиційним семисегментним індикатором з загальним катодом та регістрами портів Arduino. У програмі використовується функція `Cifra()` та бітові операції `bitSet`, `bitClear` для формування цифри, яка буде виводитись на індикатор, у семисегментному коді. Алгоритм програми простий – на індикатор виводиться число 5678 в режимі динамічної індикації з частотою 200Гц (50 Гц на кожен позицію).

Лістинг програми SEG4

```

//A-PD2, B-PD3, C-PD4, D- PD5, E-PD6, F-PD7
//G-PB0, D1-PB1, D2-PB2, D3-PB3, D4-PB4
# define P0 B00000010;
# define P1 B00000100;

```

```

#define P2 B00001000;
#define P3 B00010000;
int tmp=5;
int tmp1=6;
int tmp2=7;
int tmp3=8;

void setup() {
    DDRD=0xFF;
    DDRB=0xFF;
    PORTD =0x00;
    bitClear(PORTB,0);
}

void loop() {
    PORTB=P0;
    Cifra(tmp);
    delay (5);
    PORTB=P1;
    Cifra(tmp1);
    delay (5);
    PORTB=P2;
    Cifra(tmp2);
    delay (5);
    PORTB=P3;
    Cifra(tmp3);
    delay (5);
}

void Cifra(int x){
    switch (x)
    {
        case 0:
            PORTD=0xFC;
            break;
        case 1:
            PORTD=0x18;
            break;
        case 2:
            PORTD=0x6C;
            bitSet(PORTB,0);
            break;
        case 3:
            PORTD=0x3C;
            bitSet(PORTB,0);
            break;
        case 4:
            PORTD=0x98;
            bitSet(PORTB,0);
            //PORTB|=1;
            break;
        case 5:
            PORTD=0xB4;
            bitSet(PORTB,0);
            break;
    }
}

```

```

    case 6:
        PORTD=0xF4;
        bitSet(PORTB,0);
        break;
    case 7:
        PORTD=0x1C;
        bitClear(PORTB,0);
        break;
    case 8:
        PORTD=0xFC;
        bitSet(PORTB,0);
        break;
    case 9:
        PORTD=0xBC;
        bitSet(PORTB,0);
        break;
    default:
        break;
}
}

```

Програма SEG5

Програма демонструє роботу з 4 позиційним семисегментним індикатором з загальним катодом. У програмі використовується масиви *numeral[]*, *segmentPins[]*, *digitPins[]*, щоб задати конфігурацію підключення індикатора та представлення цифри, яка буде виводитись на індикатор, у семисегментному коді. У програмі використовується функції *showNumber()* та *showDigit()*, які організовують динамічну індикацію, переводять значення змінної *value* у BCD (двійково-десятковий) код, який відображається на індикаторі. Алгоритм програми простий – на індикатор виводиться число зі змінної *value* 5678 в режимі динамічної індикації з частотою 200Гц. Програму можна використовувати для відображення інформації з АЦП або аналогового датчика. Для цього потрібно їх значення зчитати командою *analogRead()* до змінної *value*.

Лістинг програми SEG5

```

const int numeral[10] = {
    //ABCDEFGH
    B11111100, // 0
    B01100000, // 1
    B11011010, // 2
    B11110010, // 3
    B01100110, // 4
    B10110110, // 5
    B00111110, // 6
    B11100000, // 7
    B11111110, // 8
    B11100110, // 9
}

```



```

};

//                                H,G,F,E,D,C,B,A
const int segmentPins[] = {13,8,7,6,5,4,3,2};
const int nbrDigits= 4;

//digital                                0  1  2  3
const int digitPins[nbrDigits] = {9,10,11,12};

void setup()
{
    for(int i=0; i < 8; i++) {
        pinMode(segmentPins[i], OUTPUT);
    }
    for(int i=0; i < nbrDigits; i++) {
        pinMode(digitPins[i], OUTPUT);
    }
}

void loop()
{
    //int value = analogRead(0);
    int value = 1025;
    showNumber(value);
}

void showNumber( int number)
{
    if(number == 0) {
        showDigit( 0, nbrDigits-1) ;
    }
    else {
        for( int digit = nbrDigits-1; digit >= 0; digit--){
            if(number > 0){
                showDigit( number % 10, digit) ;
                number = number / 10;
            }
        }
    }
}

void showDigit( int number, int digit)
{
    digitalWrite( digitPins[digit], HIGH );
    for(int segment = 1; segment < 8; segment++) {
        boolean isBitSet = bitRead(numeral[number], segment);
        digitalWrite( segmentPins[segment], isBitSet);
    }
    delay(5);
    digitalWrite( digitPins[digit], LOW );
}

```

Програма SEG6

Програма демонструє роботу з 4 позиційним семисегментним індикатором з загальним катодом. У програмі використовується переривання від Timer1 для формування динамічної індикації та переведення значення змінної *count* у BCD (двійково-десятковий) код. Функції *Disp()* виконує перекодування цифри у BCD коді в код семисегментного індикатора. Алгоритм програми простий – на індикатор виводиться число зі змінної *count* в режимі динамічної індикації. Значення змінної *count* збільшується з кожним натисненням тактової кнопки від 0 до 9999. Програму можна використовувати для більш складних проектів, наприклад, таймер, годинник, терморегулятор.

Лістинг програми SEG6

```
#define button A0

// segment pin definitions
#define SegA 2
#define SegB 3
#define SegC 4
#define SegD 5
#define SegE 6
#define SegF 7
#define SegG 8

// common pins of the four digits definitions
#define Dig1 9
#define Dig2 10
#define Dig3 11
#define Dig4 12

// variable declarations
byte current_digit;
int count = 0;

void setup()
{
    pinMode(button, INPUT_PULLUP);
    pinMode(SegA, OUTPUT);
    pinMode(SegB, OUTPUT);
    pinMode(SegC, OUTPUT);
    pinMode(SegD, OUTPUT);
    pinMode(SegE, OUTPUT);
    pinMode(SegF, OUTPUT);
    pinMode(SegG, OUTPUT);
    pinMode(Dig1, OUTPUT);
    pinMode(Dig2, OUTPUT);
    pinMode(Dig3, OUTPUT);
    pinMode(Dig4, OUTPUT);

    disp_off(); // turn off the display

    // Timer1 module overflow interrupt configuration
```

```

    TCCR1A = 0;
    TCCR1B = 1; // enable Timer1 with prescaler = 1
    TCNT1 = 0; // set Timer1 preload value to 0 (reset)
    TIMSK1 = 1; // enable Timer1 overflow interrupt
}

ISR(TIMER1_OVF_vect) // Timer1 interrupt service routine (ISR)
{
    disp_off(); // turn off the display

    switch (current_digit)
    {
        case 1:
            disp(count / 1000);
            digitalWrite(Dig1, HIGH); // turn on digit 1
            break;

        case 2:
            disp( (count / 100) % 10);
            digitalWrite(Dig2, HIGH); // turn on digit 2
            break;

        case 3:
            disp( (count / 10) % 10);
            digitalWrite(Dig3, HIGH); // turn on digit 3
            break;

        case 4:
            disp(count % 10);
            digitalWrite(Dig4, HIGH); // turn on digit 4
    }
    current_digit = (current_digit % 4) + 1;
}

// main loop
void loop()
{
    if(digitalRead(button) == 0)
    {
        count++; // increment 'count' by 1
        if(count == 9999)
            count = 0;
        delay(200); // wait 200 milliseconds
    }
}

void disp(byte number)
{
    switch (number)
    {
        case 0: // print 0
            digitalWrite(SegA, HIGH);
            digitalWrite(SegB, HIGH);
            digitalWrite(SegC, HIGH);
            digitalWrite(SegD, HIGH);
            digitalWrite(SegE, HIGH);
    }
}

```

```
        digitalWrite(SegF, HIGH);
        digitalWrite(SegG, LOW);
        break;

    case 1: // print 1
        digitalWrite(SegA, LOW);
        digitalWrite(SegB, HIGH);
        digitalWrite(SegC, HIGH);
        digitalWrite(SegD, LOW);
        digitalWrite(SegE, LOW);
        digitalWrite(SegF, LOW);
        digitalWrite(SegG, LOW);
        break;

    case 2: // print 2
        digitalWrite(SegA, HIGH);
        digitalWrite(SegB, HIGH);
        digitalWrite(SegC, LOW);
        digitalWrite(SegD, HIGH);
        digitalWrite(SegE, HIGH);
        digitalWrite(SegF, LOW);
        digitalWrite(SegG, HIGH);
        break;

    case 3: // print 3
        digitalWrite(SegA, HIGH);
        digitalWrite(SegB, HIGH);
        digitalWrite(SegC, HIGH);
        digitalWrite(SegD, HIGH);
        digitalWrite(SegE, LOW);
        digitalWrite(SegF, LOW);
        digitalWrite(SegG, HIGH);
        break;

    case 4: // print 4
        digitalWrite(SegA, LOW);
        digitalWrite(SegB, HIGH);
        digitalWrite(SegC, HIGH);
        digitalWrite(SegD, LOW);
        digitalWrite(SegE, LOW);
        digitalWrite(SegF, HIGH);
        digitalWrite(SegG, HIGH);
        break;

    case 5: // print 5
        digitalWrite(SegA, HIGH);
        digitalWrite(SegB, LOW);
        digitalWrite(SegC, HIGH);
        digitalWrite(SegD, HIGH);
        digitalWrite(SegE, LOW);
        digitalWrite(SegF, HIGH);
        digitalWrite(SegG, HIGH);
        break;

    case 6: // print 6
```

```

        digitalWrite(SegA, HIGH);
        digitalWrite(SegB, LOW);
        digitalWrite(SegC, HIGH);
        digitalWrite(SegD, HIGH);
        digitalWrite(SegE, HIGH);
        digitalWrite(SegF, HIGH);
        digitalWrite(SegG, HIGH);
        break;

    case 7: // print 7
        digitalWrite(SegA, HIGH);
        digitalWrite(SegB, HIGH);
        digitalWrite(SegC, HIGH);
        digitalWrite(SegD, LOW);
        digitalWrite(SegE, LOW);
        digitalWrite(SegF, LOW);
        digitalWrite(SegG, LOW);
        break;

    case 8: // print 8
        digitalWrite(SegA, HIGH);
        digitalWrite(SegB, HIGH);
        digitalWrite(SegC, HIGH);
        digitalWrite(SegD, HIGH);
        digitalWrite(SegE, HIGH);
        digitalWrite(SegF, HIGH);
        digitalWrite(SegG, HIGH);
        break;

    case 9: // print 9
        digitalWrite(SegA, HIGH);
        digitalWrite(SegB, HIGH);
        digitalWrite(SegC, HIGH);
        digitalWrite(SegD, HIGH);
        digitalWrite(SegE, LOW);
        digitalWrite(SegF, HIGH);
        digitalWrite(SegG, HIGH);
    }
}

void disp_off()
{
    digitalWrite(Dig1, LOW);
    digitalWrite(Dig2, LOW);
    digitalWrite(Dig3, LOW);
    digitalWrite(Dig4, LOW);
}

// end of code.

```

