

Додаток Е

Програма SHIFT

Програма демонструє роботу з 4 позиційним семисегментним індикатором з загальним катодом та регістром зсуву 74НС595. У програмі використовується переривання від Timer1 для формування динамічної індикації та переведення значення змінної *count* у BCD (двійково-десятковий) код. Функції *Disp()* виконує вивід цифри через регістр зсуву 74НС595 з використанням вбудованої функції *shiftOut()* у коді семисегментного індикатора. Алгоритм програми простий – на індикатор виводиться число зі змінної *count=9987* в режимі динамічної індикації. Значення змінної *count* збільшується з кожним натисненням тактової кнопки до 9999 із скиданням в 0. Програму можна використовувати для більш складних проектів, наприклад, таймер, годинник, терморегулятор для зменшення кількості (ущільнення) портів Arduino, що використовуються.

Лістинг програми SHIFT

```
//Q7-A, Q6-B, ...Q0-H
//SC and RC - D7, SER-D6, SHIFT-VCC, OE-GND, Button-A0

#define button    A0

// shift register pin definitions
#define clockPin  7    // clock pin
#define dataPin   6    // data pin

// common pins of the four digits definitions
#define Dig1      5
#define Dig2      4
#define Dig3      3
#define Dig4      2

byte current_digit;
int  count = 9987;

void disp(byte number);

void setup()
{
    pinMode(button, INPUT_PULLUP);
    pinMode(Dig1, OUTPUT);
    pinMode(Dig2, OUTPUT);
    pinMode(Dig3, OUTPUT);
    pinMode(Dig4, OUTPUT);
    pinMode(clockPin, OUTPUT);
    pinMode(dataPin, 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); // prepare to display digit 1
            digitalWrite(Dig1, HIGH); // turn on digit 1
            break;

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

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

        case 4:
            disp(count % 10); // prepare to display digit 4
            digitalWrite(Dig4, HIGH); // turn on digit 4
    }

    current_digit = (current_digit % 4) + 1;
}

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
            shiftOut(dataPin, clockPin, MSBFIRST, 0xFC);
            digitalWrite(clockPin, HIGH);
            digitalWrite(clockPin, LOW);
            break;
        case 1: // print 1
            shiftOut(dataPin, clockPin, MSBFIRST, 0x60);
            digitalWrite(clockPin, HIGH);
            digitalWrite(clockPin, LOW);
    }
}

```

```

        break;
    case 2: // print 2
        shiftOut(dataPin, clockPin, MSBFIRST, 0xDA);
        digitalWrite(clockPin, HIGH);
        digitalWrite(clockPin, LOW);
        break;
    case 3: // print 3
        shiftOut(dataPin, clockPin, MSBFIRST, 0xF2);
        digitalWrite(clockPin, HIGH);
        digitalWrite(clockPin, LOW);
        break;
    case 4: // print 4
        shiftOut(dataPin, clockPin, MSBFIRST, 0x66);
        digitalWrite(clockPin, HIGH);
        digitalWrite(clockPin, LOW);
        break;
    case 5: // print 5
        shiftOut(dataPin, clockPin, MSBFIRST, 0xB6);
        digitalWrite(clockPin, HIGH);
        digitalWrite(clockPin, LOW);
        break;
    case 6: // print 6
        shiftOut(dataPin, clockPin, MSBFIRST, 0xBE);
        digitalWrite(clockPin, HIGH);
        digitalWrite(clockPin, LOW);
        break;
    case 7: // print 7
        shiftOut(dataPin, clockPin, MSBFIRST, 0xE0);
        digitalWrite(clockPin, HIGH);
        digitalWrite(clockPin, LOW);
        break;
    case 8: // print 8
        shiftOut(dataPin, clockPin, MSBFIRST, 0xFE);
        digitalWrite(clockPin, HIGH);
        digitalWrite(clockPin, LOW);
        break;
    case 9: // print 9
        shiftOut(dataPin, clockPin, MSBFIRST, 0xF6);
        digitalWrite(clockPin, HIGH);
        digitalWrite(clockPin, LOW);
    }
}

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

```