

## Додаток И

### Програма LM35\_SEG

Програма демонструє роботу з датчиком температури LM35 та 4 позиційним семисегментним індикатором з загальним катодом. У програмі використовується переривання від Timer1 для формування динамічної індикації. Функції Disp() виконує перекодування цифри в код семисегментного індикатора. Алгоритм програми – на індикатор виводиться значення температури в діапазоні від 0 до 99,9°C, яке нормовано до значення опорної напруги 1,1В, у режимі динамічної індикації. Значення температури відображається з одним знаком після коми.

#### *Лістинг програми LM35\_SEG*

```
#define LM35_pin 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
#define SegDP 13

// 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 temp;

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

// set positive reference voltage to 1.1V
analogReference(INTERNAL);
}

ISR(TIMER1_OVF_vect)
{
    disp_off(); // turn off the display

    switch (current_digit)
    {
        case 1:
            disp((temp / 100) % 10);
            digitalWrite(Dig1, HIGH);
            digitalWrite(SegDP, LOW); // turn on digit 1
            break;

        case 2:
            disp((temp / 10) % 10); // prepare to display digit 2
            digitalWrite(SegDP, HIGH); // print decimal point ( . )
            digitalWrite(Dig2, HIGH); // turn on digit 2
            break;

        case 3:
            disp(temp % 10); // prepare to display digit 3
            digitalWrite(Dig3, HIGH); // turn on digit 3
            digitalWrite(SegDP, LOW);
    }

    current_digit = (current_digit % 3) + 1;
}

void loop(){
    temp = 10 * analogRead(LM35_pin) / 9.355;
    // read analog voltage and convert it to B °C
    // (9.3 = 1023 / (1.1 * 100))
    delay(1000);
}

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);
}

```

### Програма LM35\_Shift

Програма демонструє роботу з датчиком температури LM35, 4 позиційним семисегментним індикатором з загальним катодом та регістром зсуву 74НС595. У програмі використовується переривання від Timer1 для формування динамічної індикації. Функції Disp() виконує вивід цифри через регістр зсуву

74НС595 з використанням вбудованої функції shiftOut() у коді семисегментного індикатора. Алгоритм програми – на індикатор виводиться значення температури в діапазоні від 0 до 99,9°C, яке нормовано до значення опорної напруги 1,1В, у режимі динамічної індикації. Значення температури відображається з одним знаком після коми та символом «С».

### *Лістинг програми LM35\_Shift*

```
//7-segment display with 74HC595 shift register
//4-Digit counter example
//Common catode 7-segment display is used
//Q7-A, Q6-B,...Q0-H, SC and RC - D7, SER-D6, SHIFT-VCC, OE-GND

// counter button definition
#define button    A1
#define LM35_pin 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

// variable declarations
byte current_digit;
int  temp;
void disp(int number, bool dec_point =0 );

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

    //set positive reference voltage to 1.1V
    analogReference(INTERNAL);}

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

    switch (current_digit)
    {
        case 1:
            disp( (temp / 100) % 10 );
            digitalWrite(Dig1, HIGH); // turn on digit 1
            break;

        case 2:
            disp( (temp / 10) % 10, 1 );
            digitalWrite(Dig2, HIGH); // turn on digit 2
            break;

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

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

    current_digit = (current_digit % 4) + 1;
}

void loop(){
    temp = 10*analogRead(LM35_pin)/9.3;
    // read analog voltage and convert it to B°C
    // (9.3 = 1023/(1.1*100))
    delay(1000); // wait 1 second
}

void disp(int number, bool dec_point){
    switch (number)
    {
        case 0: // print 0

```

```
    shiftOut(dataPin, clockPin, MSBFIRST, 0xFC | dec_point);
    digitalWrite(clockPin, HIGH);
    digitalWrite(clockPin, LOW);
    break;

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

case 2: // print 2
    shiftOut(dataPin, clockPin, MSBFIRST, 0xDA | dec_point);
    digitalWrite(clockPin, HIGH);
    digitalWrite(clockPin, LOW);
    break;

case 3: // print 3
    shiftOut(dataPin, clockPin, MSBFIRST, 0xF2 | dec_point);
    digitalWrite(clockPin, LOW);
    break;

case 4: // print 4
    shiftOut(dataPin, clockPin, MSBFIRST, 0x66 | dec_point);
    digitalWrite(clockPin, HIGH);
    digitalWrite(clockPin, LOW);
    break;

case 5: // print 5
    shiftOut(dataPin, clockPin, MSBFIRST, 0xB6 | dec_point);
    digitalWrite(clockPin, HIGH);
    digitalWrite(clockPin, LOW);
    break;

case 6: // print 6
    shiftOut(dataPin, clockPin, MSBFIRST, 0xBE | dec_point);
    digitalWrite(clockPin, HIGH);
    digitalWrite(clockPin, LOW);
    break;

case 7: // print 7
    shiftOut(dataPin, clockPin, MSBFIRST, 0xE0 | dec_point);
    digitalWrite(clockPin, HIGH);
    digitalWrite(clockPin, LOW);
    break;

case 8: // print 8
    shiftOut(dataPin, clockPin, MSBFIRST, 0xFE | dec_point);
```

```

        digitalWrite(clockPin, HIGH);
        digitalWrite(clockPin, LOW);
        break;

    case 9: // print 9
        shiftOut(dataPin, clockPin, MSBFIRST, 0xF6 | dec_point);
        digitalWrite(clockPin, HIGH);
        digitalWrite(clockPin, LOW);
        break;

    case 10: // print C
        shiftOut(dataPin, clockPin, MSBFIRST, 0x9C | dec_point);
        digitalWrite(clockPin, HIGH);
        digitalWrite(clockPin, LOW);
        break;
}
}

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

```

### Програма LM35\_LCD

Програма демонструє роботу з датчиком температури LM35 та LCD-індикатором. Алгоритм програми – на індикатор при старті виводиться напис «Digital Thermometer»; виконується вимірювання даних з LM35 та їх нормування до значення опорної напруги 1,1В. Значення температури відображається так: у верхньому рядку виводиться по центру напис «Temperature»; у нижньому рядку значення температури з одним знаком після коми та символом «°C».

#### *Лістинг програми LM35\_LCD*

```

// A0 - LM35
//Arduino pins 12, 11, 5, 4, 3, 2
//Arduino pins RS, E, D4, D5, D6, D7

#include <LiquidCrystal.h>
LiquidCrystal lcd(12,11,5,4,3,2);

#define sensor A0
byte degree[8] =
{
    0b00011,
    0b00011,
    0b00000,

```



```

0b000000,
0b000000,
0b000000,
0b000000,
0b000000
};
void setup(){
  lcd.begin(16,2);
  lcd.createChar(1, degree);
  lcd.setCursor(0,0);
  lcd.print(" Digital ");
  lcd.setCursor(0,1);
  lcd.print(" Thermometer ");
  delay(2000);
  lcd.clear();
  // set positive reference voltage to 1.1V
  analogReference(INTERNAL);
}

void loop(){
  /*-----Temperature-----*/
  float reading=analogRead(sensor);
  float temperature= reading /9.385;
  //read analog voltage and convert it to °C
  delay(10);

  /*-----Display Result-----*/
  lcd.clear();
  lcd.setCursor(2,0);
  lcd.print("Temperature");
  lcd.setCursor(4,1);
  lcd.print(temperature, 1);
  // 1 задає кількість знаків після коми
  lcd.write(1);
  lcd.print("C");

  delay(1000);
}

```

