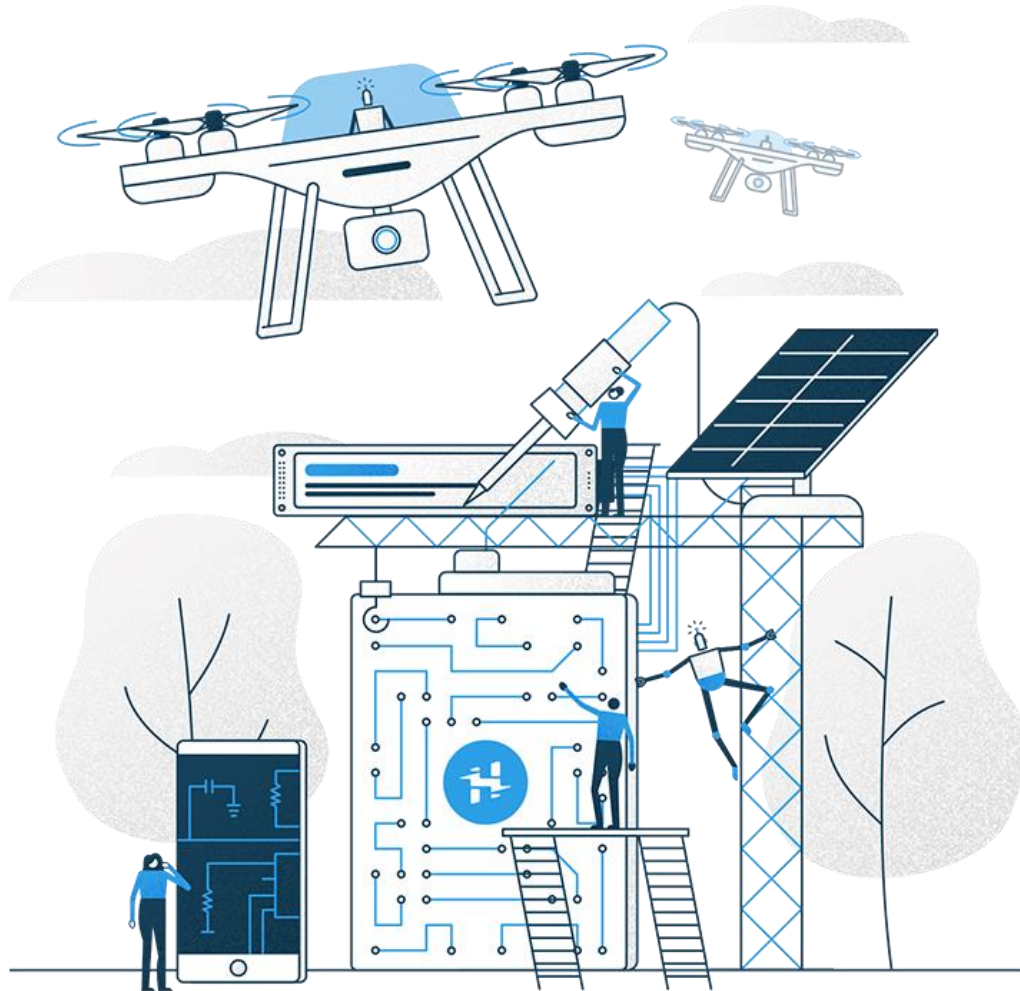


МІКРОПРОЦЕСОРНІ СИСТЕМИ УПРАВЛІННЯ

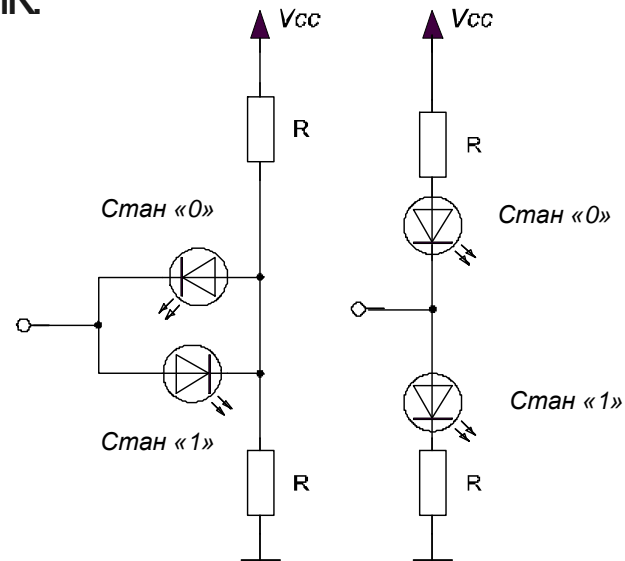
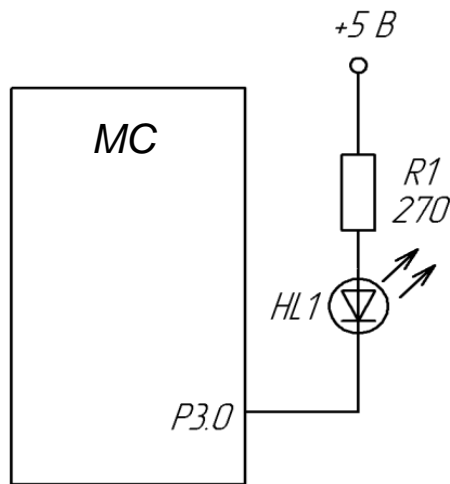


Lesson 11

Елементи індикації

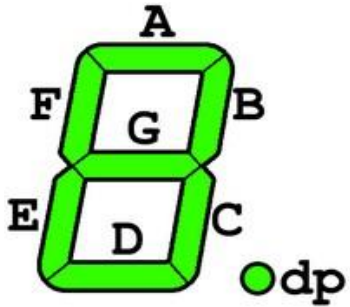
Практично кожна мікропроцесорна система управління містить елементи індикації. Як індикатори в даний час найчастіше застосовуються світлодіоди. На ринку є величезний вибір світлодіодів найрізноманітніших видів і розмірів.

У мікропроцесорній системі управління LED індикатори можуть служити для відображення різних режимів роботи: попередження про критичні ситуації, відображення ходу прийому сигналів керування тощо. Підключити одиночний світлодіодний індикатор до МК дуже просто. На рис. наведена схема підключення світлодіода безпосередньо до виводу порту МК.



При роботі порту в режимі виходу, залежно від стану "0" або "1" горить відповідно верхній або нижній світлодіод. При перемиканні в Z-стан, і при відповідному виборі резисторів струм через світлодіоди дуже малий і їх світіння майже непомітно.

Семисегментний індикатор



On	88	Off	8FF
Close	EE05E	Play	PEAY
Pause	PAUSE	Stop	SE8E
List	EESE	Shuffle	SHUFFLE
Error	EE88E	Wait	VADE

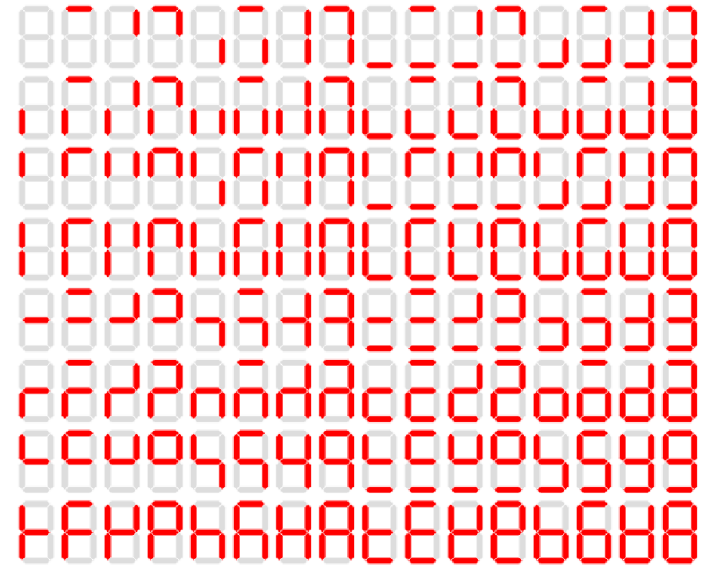


Схема з загальним анодом

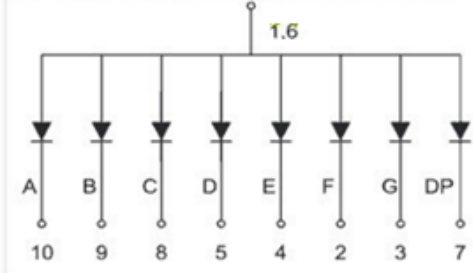
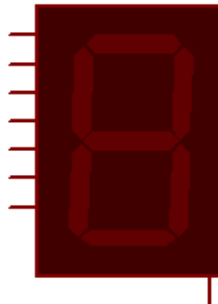
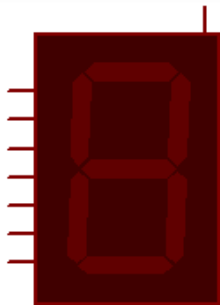
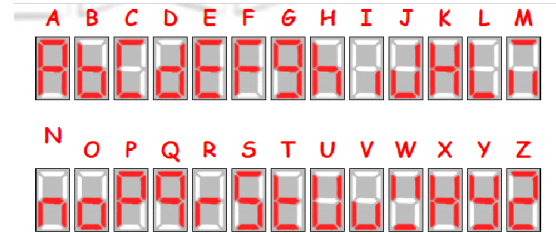
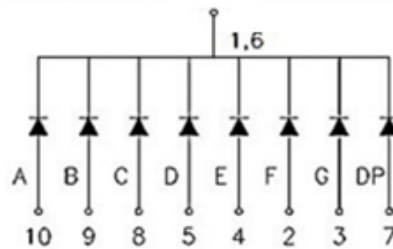


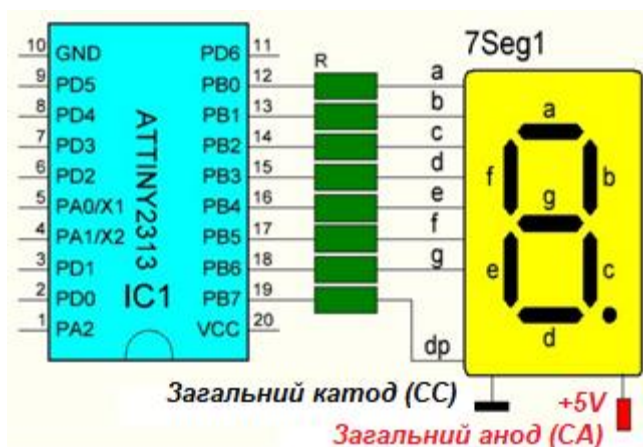
Схема з загальним катодом



Семисегментний індикатор

Якщо індикатор із загальним катодом, то його загальний вивід підключається до «землі», а керування сегментами відбувається подачею логічної одиниці на вивід порту.

Якщо індикатор з загальним анодом, то на його загальний провід подають «плюс» напруги, а керування сегментами відбувається подачею логічного нуля на вивід порту.



Характеристики індикатора

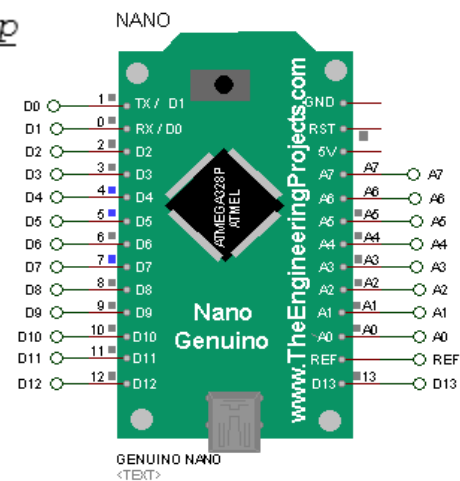
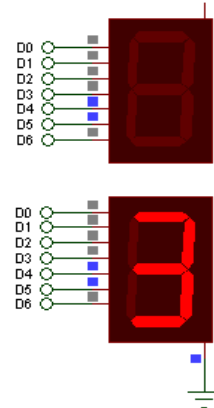
- робоча напруга – 2В
- робочий струм ток – 10 мА
- напруга живлення 5В

Формула розрахунку:

$$R = U/I = (5-2)/0,01 = 300 \text{ Ом}$$

Лістинг програми 1. Вивід інформації на 7 сегментний індикатор

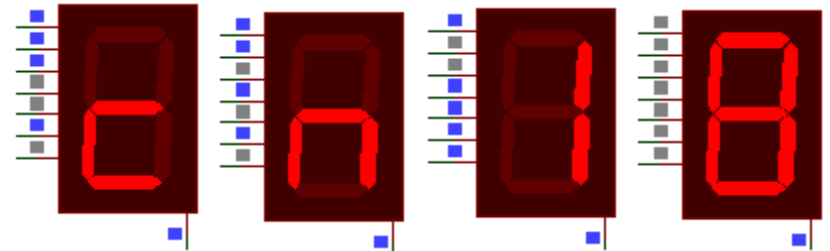
```
void setup() {  
  //виставляємо всі біти порти D як вихід  
  DDRD = 0xFF;  
}  
void loop() {  
  PORTD = 0x4F; //виводимо 3 CC  
  //PORTD = 0xB0; //виводимо 3 CA  
  delay (1000);  
}
```

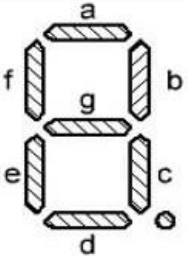


Семисегментний індикатор

Лістинг програми 2. Вивід інформації на 7 сегментний індикатор

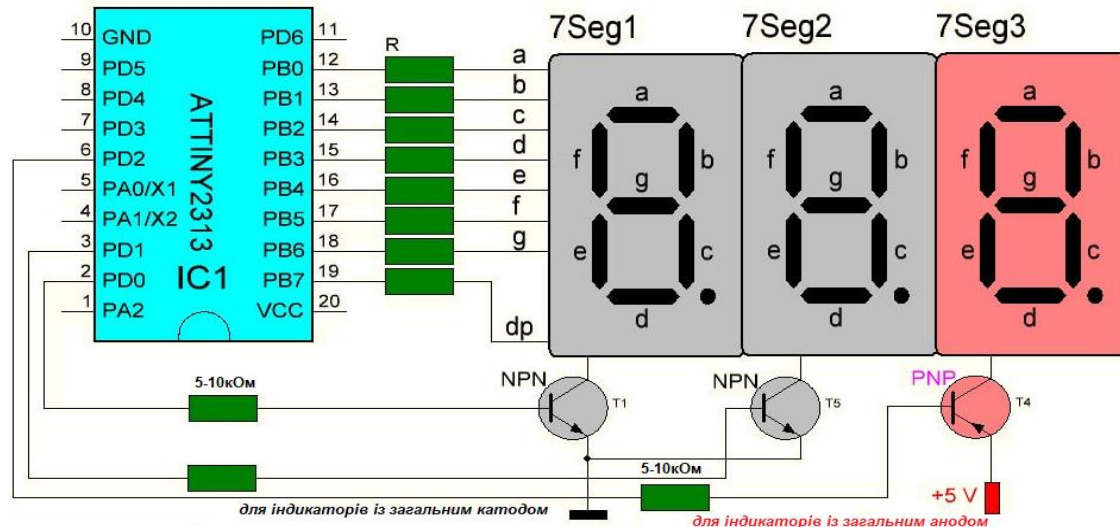
```
byte text[]={0x58, 0x54, 0x06, 0x7F};  
//From CA{0xA7,0xAB,0xF9, 0x80}  
void setup() {  
  //виставляємо всі біти порти D як вихід  
  DDRD = 0xFF;  
}  
void loop() {  
  for (int i=0; i<4; i++){  
    PORTD = text[i];  
    delay (500);  
    PORTD = 0x00;  
    delay (500);  
  }  
}
```



	dp	G	F	E	D	C	B	A	h
	7	6	5	4	3	2	1	0	
0	0	0	0	0	0	0	0	0	
1		0	1	1	1	1	1	1	3F
2		0	0	0	0	1	1	0	6
3		1	0	1	1	0	1	1	5B
4		1	0	0	1	1	1	1	4F
5		1	1	0	0	1	1	0	66
6		1	1	1	1	1	0	1	6D
7		1	1	1	1	1	0	1	7D
8		0	0	0	0	1	1	1	7
9		1	1	1	1	1	1	1	7F
		1	1	0	1	1	1	1	6F

Багаторозрядний семисегментний індикатор

Багаторозрядні світлодіодні індикатори часто працюють за динамічним принципом: виводи однойменних сегментів всіх розрядів з'єднані разом. Щоб виводити інформацію на такий індикатор, мікроконтролер повинний циклічно подавати сигнал на загальні виводи всіх розрядів, в той час як на виводи сегментів сигнал подається в залежності від того, який сегмент в даному розряді має світитися.



Здiйснення iндикацiї розрядами динамiчним способом:

- виставляється двiйковий код вiдповiдної цифри на виходах порту РВ для 1 розряду, потiм подається логiчний рiвень на керуючий транзистор першого розряду;
- виставляється двiйковий код вiдповiдної цифри на виходах порту РВ для 2 розряду, потiм подається логiчний рiвень на керуючий транзистор другого розряду;
- виставляється двiйковий код вiдповiдної цифри на виходах порту РВ для 3 розряду, потiм подається логiчний рiвень на керуючий транзистор третього розряду;
- усе повторяемо.

Багаторозрядний семисегментний індикатор

```
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() {
  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);
```

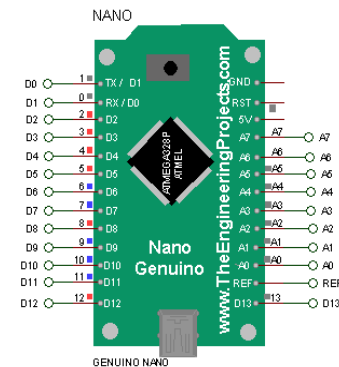
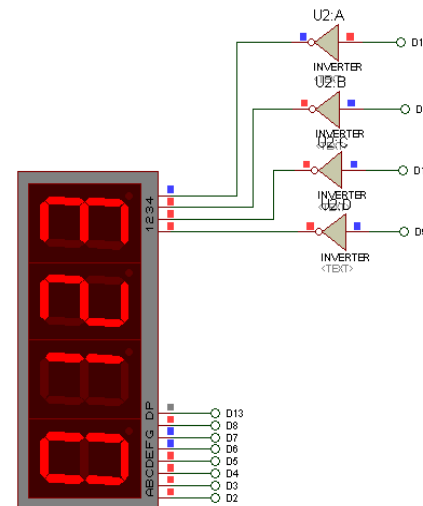
```
digitalWrite(pinA, HIGH);
digitalWrite(pinB, HIGH);
digitalWrite(pinC, HIGH);
digitalWrite(pinD, HIGH);
digitalWrite(pinE, HIGH);
digitalWrite(pinF, HIGH);
digitalWrite(pinG, LOW);
delay(10);

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(45);
```



Багаторозрядний семисегментний індикатор

```
//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() {
```

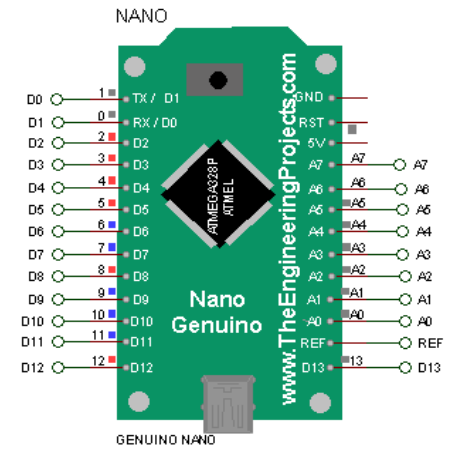
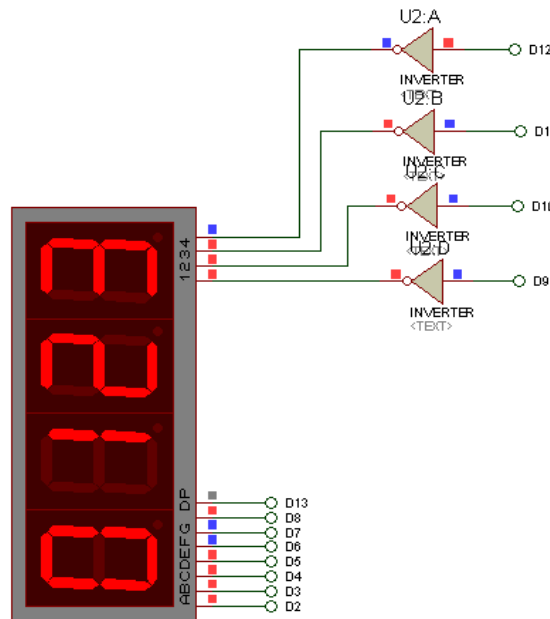
```
  DDRD=0xFF;
  DDRB=0xFF;
  PORTD =0x00;
  PORTB = 0<<0;
}
```

```
void loop() {
  PORTD=B11111100; //0
  PORTB=P0;
  delay (40);
```

```
  PORTD=B00011000; //1
  PORTB=P1;
  delay (40);
```

```
  PORTD=B01101100; //2
  PORTB=P2;
  PORTB|=1;
  delay (40);
```

```
  PORTD=B00111100; //3
  PORTB=P3;
  PORTB|=1;
  delay (40);
}
```



Багаторозрядний семисегментний індикатор

```
// counter button definition
#define button A0

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

#define Dig1 9
#define Dig2 10
#define Dig3 11
#define Dig4 12

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 ( 16 ticks each 1 μs)
  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 (most left)
      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 (most right)
      digitalWrite(Dig4, HIGH); // turn on digit 4
  }
}
```

Багаторозрядний семисегментний індикатор

```
current_digit = (current_digit % 4) + 1; digitalWrite(SegD, LOW); digitalWrite(SegG, HIGH);
} digitalWrite(SegE, LOW); break;
digitalWrite(SegF, LOW);
digitalWrite(SegG, LOW);
break;

// 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);

    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);

    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;

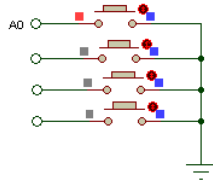
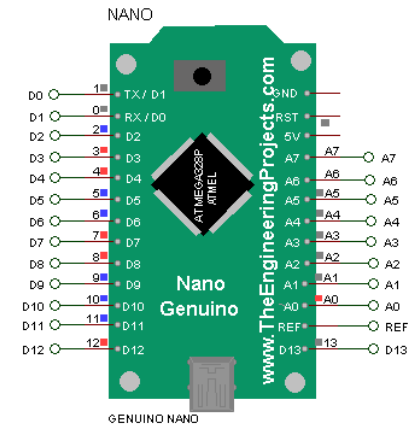
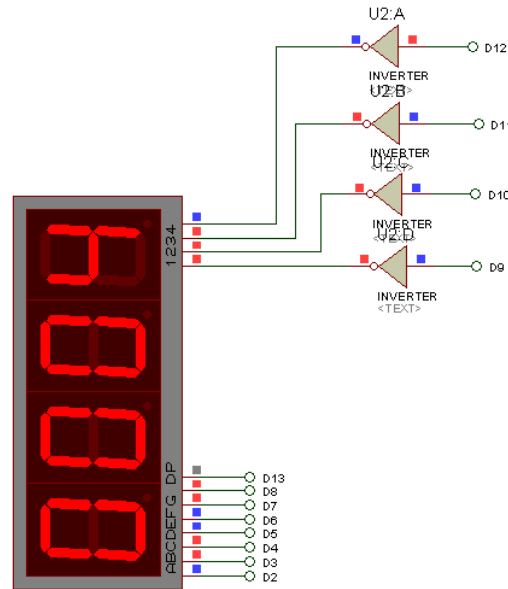
    case 7: // print 7
      digitalWrite(SegA, HIGH);
      digitalWrite(SegB, HIGH);
      digitalWrite(SegC, HIGH);
      digitalWrite(SegD, LOW);
      digitalWrite(SegE, LOW);
      digitalWrite(SegF, LOW);
      digitalWrite(SegG, LOW);
      break;
  }
}
```

Багаторозрядний семисегментний індикатор

```
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);
```

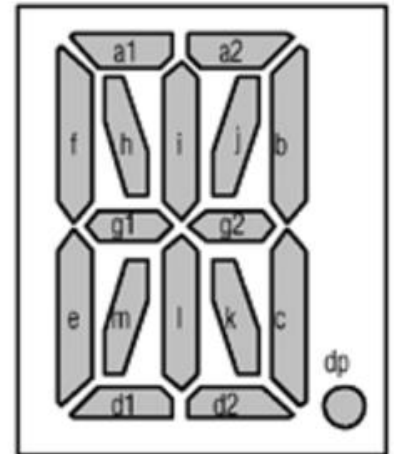
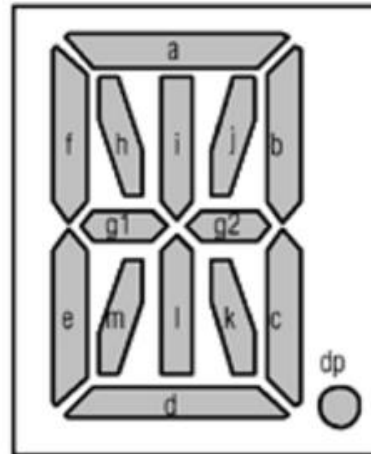
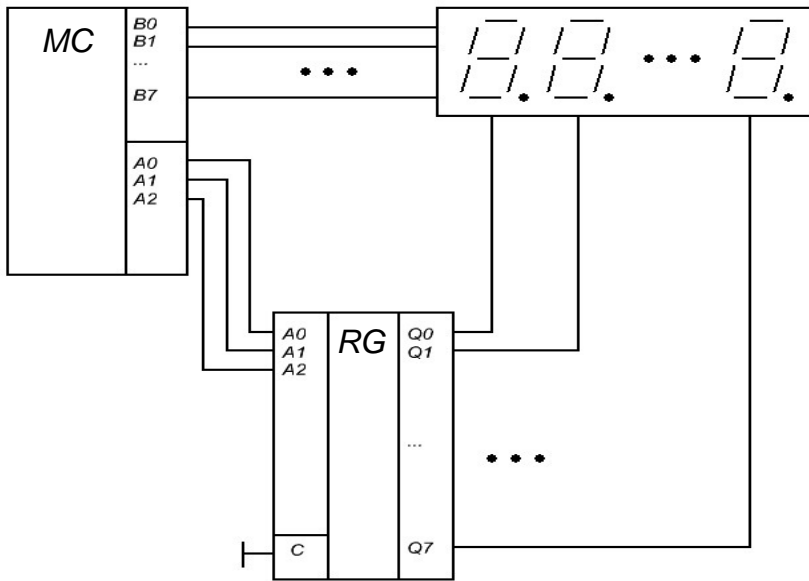
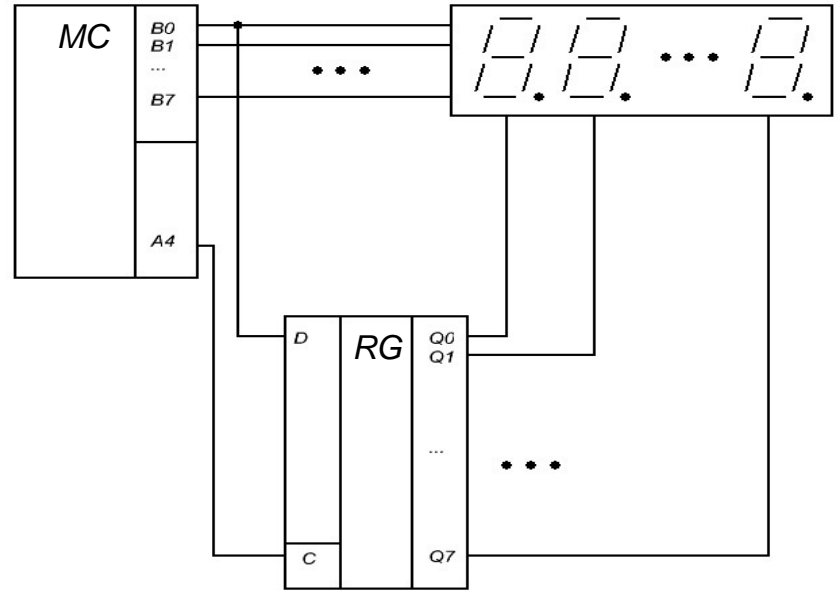
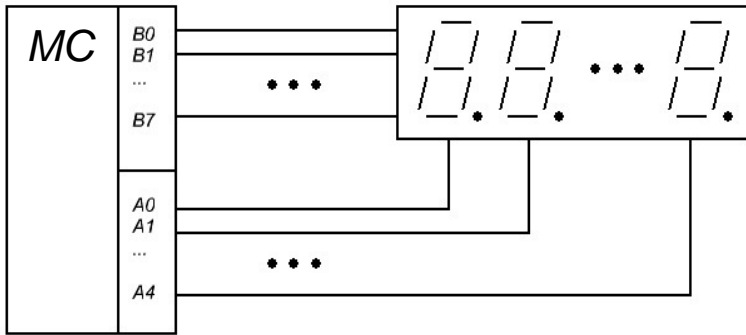


```
}  
}
```

```
void disp_off()
```

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

Багаторозрядний семисегментний індикатор



Індивідуальне завдання

1. Реалізувати секундомер на базі модуля Arduino та 4 розрядного семисегментного індикатора у середовищі Proteus.
2. Реалізувати генератор псевдовипадкових чисел від 0 до 9999. При натисненні кнопки індикатор гасне. При відпусканні кнопки на індикаторі відображається генероване число

