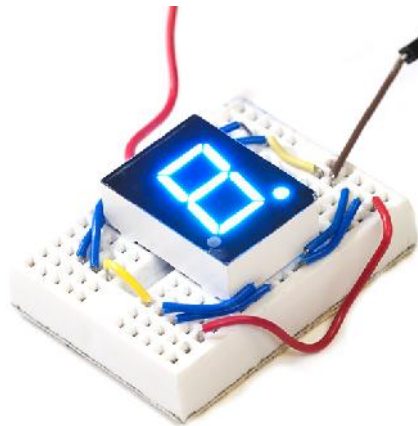


## Тема 3. Індикація

### Урок 3.1. Сегментний світлодіодний індикатор

#### 3.1.1. Пристрій сегментного індикатора



Контури цифр.

Ми вже познайомилися з світлодіодом одним з найбільш часто використовуваних індикаторів. Звичайним світлодіодом легко інформувати користувача про включення або виключенні будь-якого пристрою, про перевищення порогових значень і т.п. Але що, якщо потрібно інформувати користувача кількісно? Іншими словами, що якщо прилад повинен повідомляти вимірювані значення явно, у вигляді чисел?

Для цих цілей, в електроніці часто використовується

сегментний світлодіодний індикатор. Цей прилад являє собою набір звичайних світлодіодів, розташованих таким чином, що всі вони разом створюють конструктивно, світлодіодні індикатори поділяються на прилади з загальним катодом, і з загальним анодом. Загальний катод, наприклад, означає що всі світлодіоди всередині індикатора з'єднані разом катодами, а їх аноди розведені по окремим контактам. Саме такий тип індикатора використовується в наших завданнях.

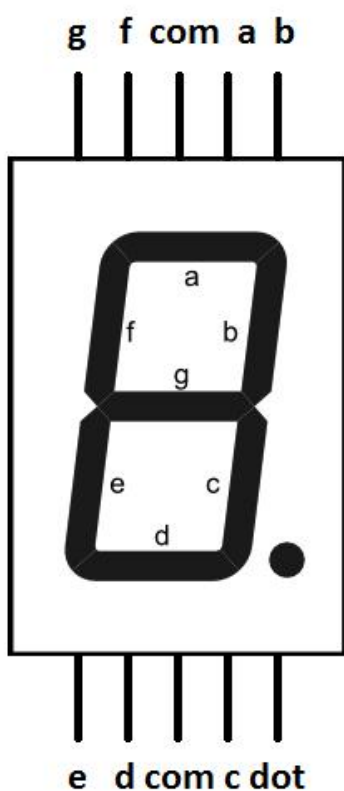


Схема підключення сегментного індикатора представлена на малюнку зліва.

Всі сегменти промарковані літерами латинського алфавіту, починаючи з а і закінчуючи g. Як правило, рахунок ведуть з самого верхнього сегмента за годинниковою стрілкою. Точка маркується окремо, словом dot (точка).

Крім контактів, з'єднаних з сегментами, на схемі видно два контакту com і один dot. Як вже було написано раніше, слово dot використовується для позначення точки, а com той самий загальний катод (Або анод).

## Завдання

### Завдання 1. Висновок цифри

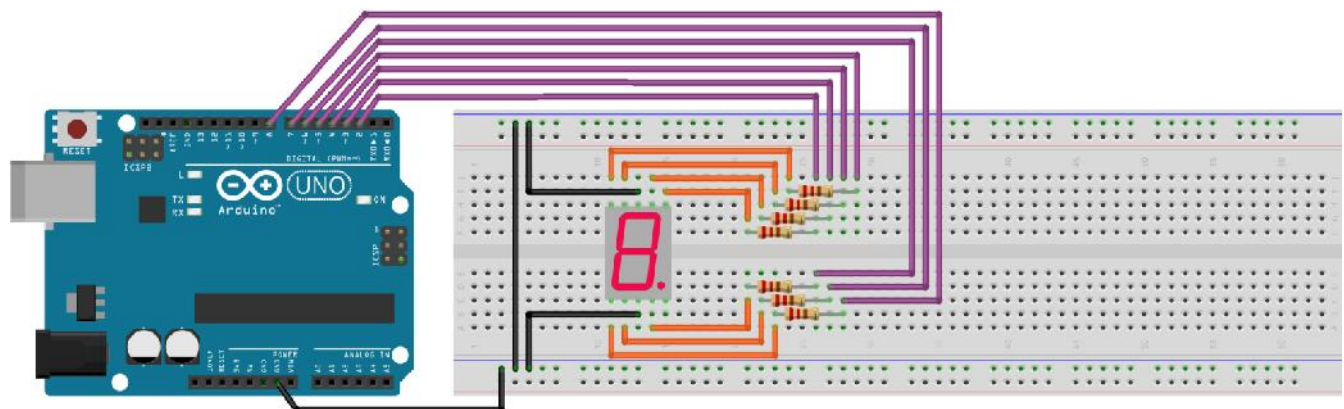
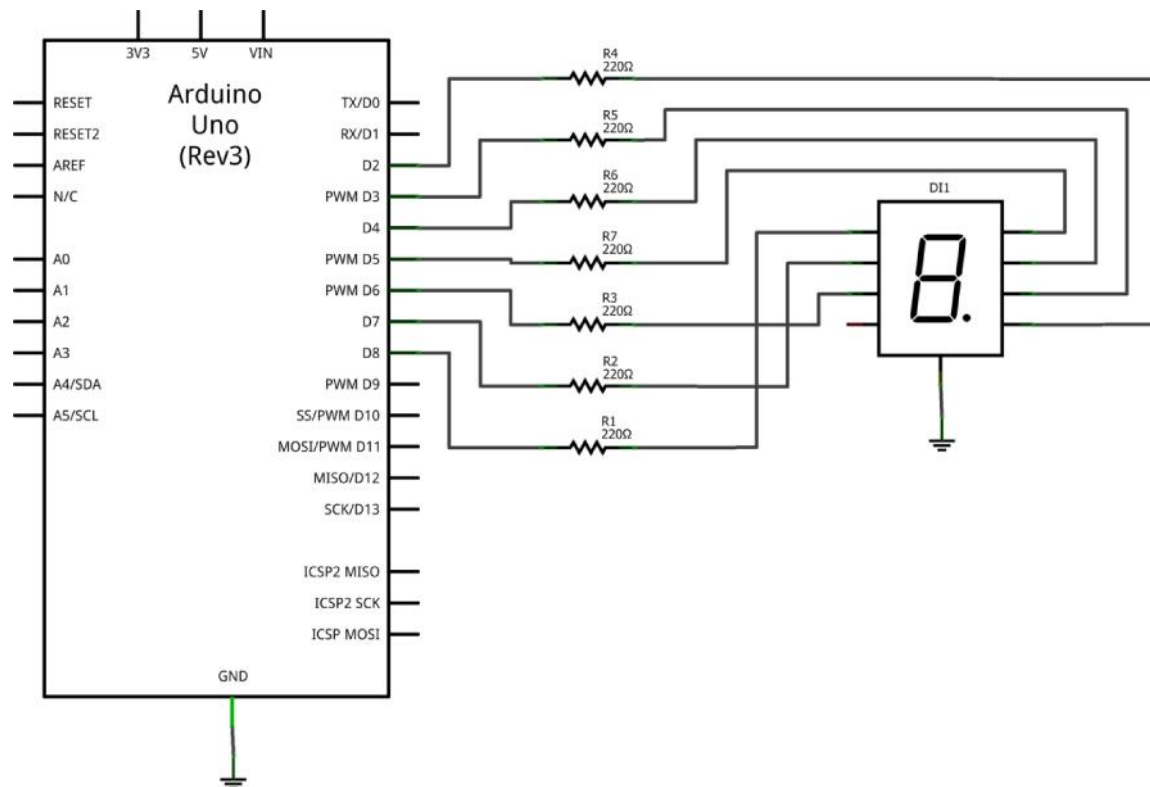
#### Результат праці

Після запуску програми, на індикаторі повинна з'явитися будь-яка цифра.

#### Використовувані компоненти



#### Схема установки



## Програма

```
int seg_a = 3;
int seg_b = 2;
int seg_c = 6;
int seg_d = 7;
int seg_e = 8;
int seg_f = 4;
int seg_g = 5;
void setup () {
pinMode (seg_a, OUTPUT);
pinMode (seg_b, OUTPUT);
pinMode (seg_c, OUTPUT);
pinMode (seg_d, OUTPUT);
pinMode (seg_e, OUTPUT);
pinMode (seg_f, OUTPUT);
pinMode (seg_g, OUTPUT);
digitalWrite (seg_a, HIGH);
digitalWrite (seg_d, HIGH);
digitalWrite (seg_e, HIGH);
digitalWrite (seg_f, HIGH);
digitalWrite (seg_g, HIGH);
}
void loop () {
}
```

### Завдання 2. Зворотний відлік

#### Результат праці

Після запуску програми, на індикаторі висвічується цифра 9. З кожною секундою, цифра зменшується на одиницю.

### Завдання 3. Лічильник з кнопкою (самостійно)

#### Результат праці

Схема містить кнопку і сегментний індикатор. Після запуску, програма переходить в режим очікування. Після кожного натискання кнопки, цифра на індикаторі збільшується на одиниць, і скидається, при досягненні цифри 9.

### Завдання 4. Анімація сегментів (самостійно)

#### Результат праці

Схема містить тільки сегментний індикатор. Після запуску програми, починають по черзі запалюватися і гаснуть всі крайні сегменти. Послідовність: нижній лівий, верхній лівий, верхній, верхній правий, нижній правий, нижній.

## Урок 4.1. рідкокристалічний дисплей

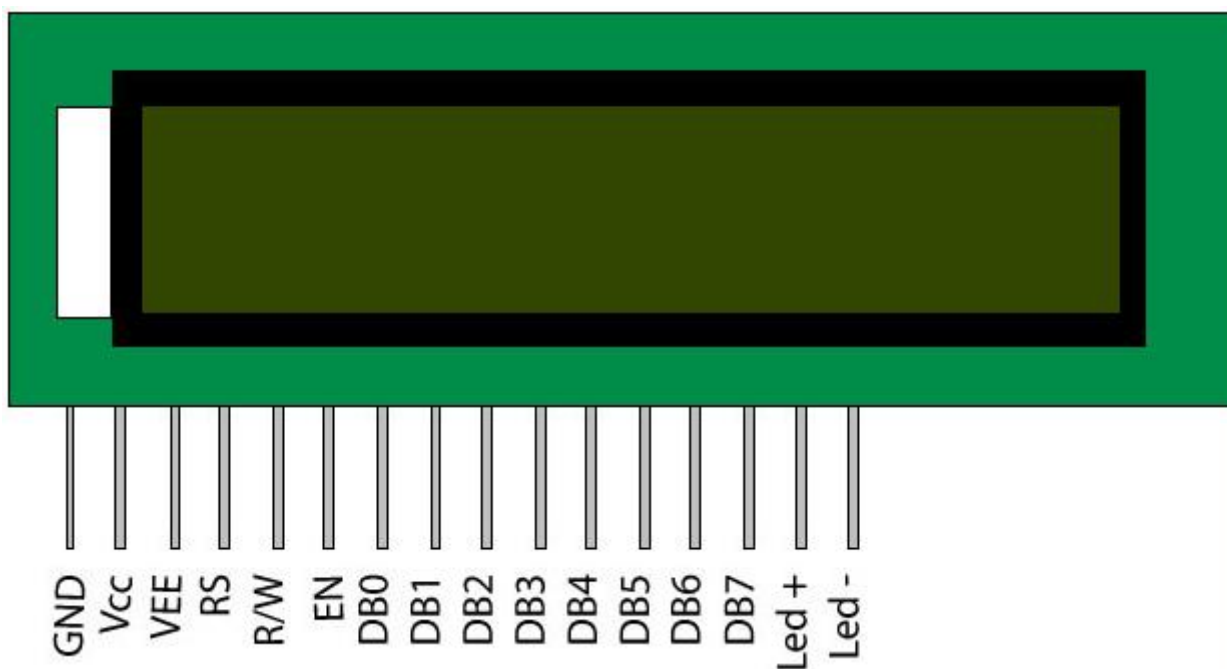
### 4.1.1. Пристрій ЖК дисплея



Самий інформативний, і разом з тим найскладніший вид індикаторів це дисплеї. У нашому курсі ми будемо працювати з рідкокристалічним індикатором, представників якого можна зустріти в багатьох електронних пристроях.

Електронний годинник, калькулятори, екрани старих стільникових телефонів, всі вони використовують аналогічні ЖК індикатори.

#### Виводи ЖК дисплею 1602



**GND** - земля (мінус живлення);

**Vcc** - живлення +5 В;

**VEE** - регулятор контрасту дисплея;

**RS** - вибір регістру;

**R / W** - напрямок передачі даних (запис / читання);

**EN** - синхронізація;

**DB0DB7** - шина даних;

**Led, Led +** - катод і анод підсвічування (+5 В);

### 4.1.2. Бібліотека для роботи з РК дисплеєм

Для роботи з РК дисплеями різних розмірів і типів, в редакторі Arduino IDE є спеціальна бібліотека LiquidCrystal. Для того, щоб скористатися цією бібліотекою в програмі, необхідно на початку коду вказати наступну директиву:

```
#include <LiquidCrystal.h>
```

Для ініціалізації дисплея, необхідно вказати які саме контакти, використовуються для підключення до Arduino:

```
LiquidCrystal lcd (12, 11, 5, 4, 3, 2);
```

де перший аргумент контакт RS

другий EN

а з третього по шостий D4..D7

Виведення тексту на дисплей здійснюється за допомогою функції print, що має формат:

```
lcd.print (текст_ілі_переменная);
```

в якості аргументу функції допускається вказувати текст, відокремлений подвійними лапками, або змінні будь-яких типів.

Оскільки дисплей являє собою набір символів, згрупованих у два рядки по 16 знакомісць в кожному, в бібліотеці передбачена функція для явної вказівки позиції виведення:

```
lcd.setCursor (номер_колонки, номер_рядка);
```

Так, для виведення символу у другому рядку потрібно викликати **setCursor** з наступними аргументами:

```
lcd.setCursor (0, 1); // колонка з індексом 0, рядок з індексом 1
```

Приклад використання бібліотеки.

```
#include <LiquidCrystal.h>
LiquidCrystal lcd (12, 11, 5, 4, 3, 2); // ініціалізація дисплея
void setup () {
  lcd.begin (16, 2); // параметри дисплея 16символів, 2строки
  lcd.print ( "hello, world!"); // вивід на дисплей тексту
}
void loop () {
}
```

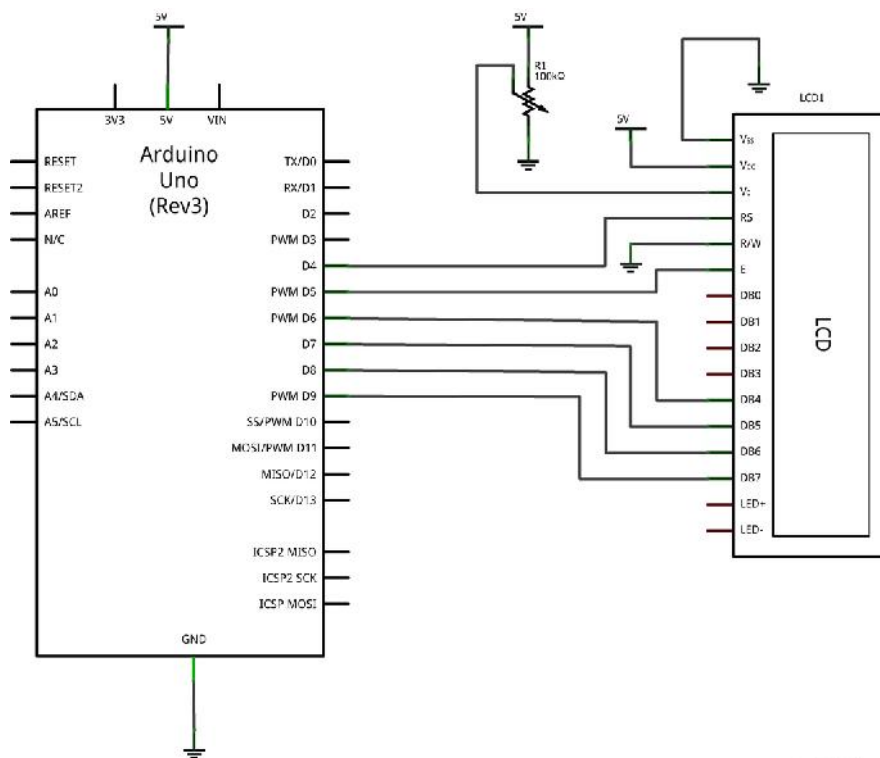
## Завдання

### Завдання 1. Висновок довільного тексту і лічильника

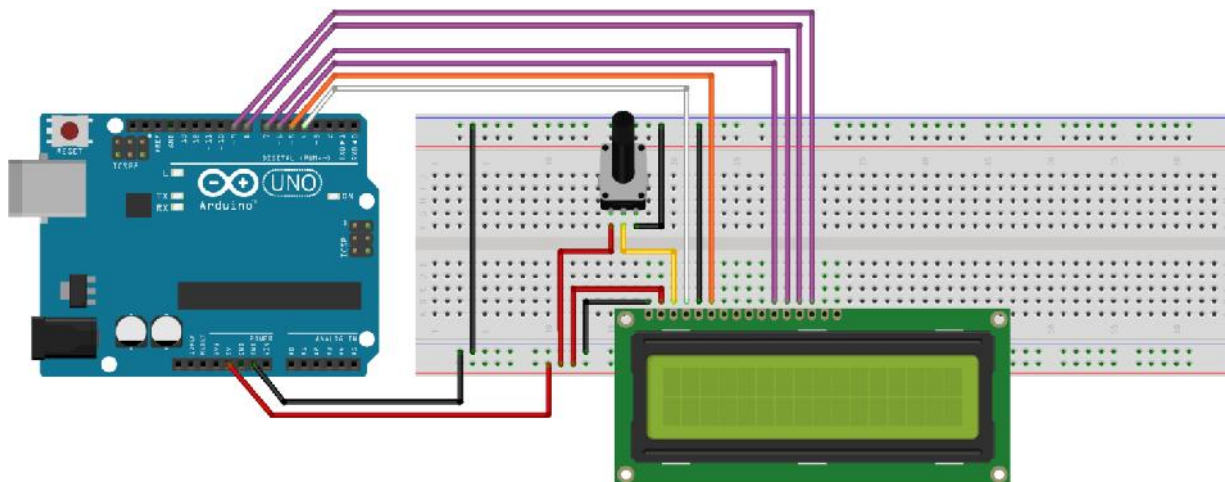
#### Результат праці

Після запуску програми, в першому рядку РК дисплея з'являється напис, а у другому рядку починається рахунок.

Використовувані компоненти:



Made with  fritzing.org



Made with  Fritzing.org

## **Завдання 2. Лічильник з кнопкою (самостійно)**

### **Результат праці**

Після запуску програми, в першому рядку РК дисплея з'являється напис. При натисканні кнопки, лічильник у другому рядку програми додається на одиницю.

## **Завдання 3. Та, що біжить рядок (самостійно)**

### **Результат праці**

Після запуску програми, по першому рядку РК дисплея починає переміщатися текст.

## **Урок 3.3. Передача повідомлень на персональний комп'ютер**

### **3.3.1. Функція запису в послідовний порт ПК**

Arduino може надсилати текстові повідомлення на персональний комп'ютер. Для відображення цих повідомлень, в редакторі Arduino IDE передбачено спеціально вікно "Монітор порту".

Для того щоб скористатися функцією передачі, необхідно в розділі setup форматувати послідовний порт пристрою:

```
Serial.begin (9600);
```

Безпосередньо для передачі короткого повідомлення використовується функція print:

```
Serial.print ( "text");
```

### **Приклад програми**

```
void setup () {  
  Serial.begin (9600);  
  Serial.println ( "Hello world!");  
}  
void loop () {  
}
```

## **Завдання**

### **Завдання 1. Передача довільного тексту в послідовний порт ПК (самостійно)**

#### **Результат праці**

Після запуску програми, в моніторі порту з'являється довільна напис англійською мовою.

## Завдання 2. Поточний час (самостійно)

### Результат праці

Після запуску програми, в моніторі порту кожную секунду з'являється поточні час і дата.

## Завдання 3. Відображення спрацьовувань кнопки (брязкіт контактів) (самостійно)

### Результат праці

Схема містить одну кнопку. Після запуску, програма переходить в режим очікування. Під час натискання кнопки, в моніторі порту відображається слово "push".

## Урок 3.4. індикація часу

### 3.4.1. Бібліотека для роботи з часом і датою

На відміну від звичайного персонального комп'ютера, в Arduino немає вбудованих годин реального часу. Внаслідок цього, Arduino не може постійно зберігати в собі поточний час, а тільки здатний відраховувати його після запуску з деякою позиції.

Для роботи з часом передбачена бібліотека Time. Для підключення бібліотеки, на початку коду програми необхідно додати інструкцію:

```
#include <Time.h>
```

Потім в розділі setup слід встановити поточну дату і час:

```
setTime (години, хвилини, секунди, місяць, число, рік);
```

Наприклад, так:

```
setTime (12, 0, 0, 1, 1, 13); // 01.01.2013 12:00:00
```

Для отримання компонентів дати і часу використовуються наступні функції:

**year ()** рік

**month ()** місяць

**day ()** день

**hour ()** годину

**minute ()** хвилину

**second ()** секунда



## Приклад програми

```
#include <Time.h>
void setup () {
  Serial.begin (9600);
  setTime (12,0,0,1,1,13);
}
void loop () {
  Serial.print (hour ());
  Serial.print ( ":");
  Serial.print (minute ());
  Serial.print ( ":");
  Serial.print (second ());
  Serial.print ( "");
  Serial.print (day ());
  Serial.print ( "");
  Serial.print (month ());
  Serial.print ( "");
  Serial.print (year ());
  Serial.println ();
  delay (1000);
}
```

### Завдання

#### Завдання 1. Електронний годинник

##### Результат праці

Після запуску програми, на ЖК дисплеї відображається поточний час, яке оновлюється кожну секунду.

#### Завдання 2. Електронний годинник з будильником (самостійно)

##### Результат праці

Після запуску програми, на ЖК дисплеї відображається поточний час. За допомогою однієї кнопки користувач може вказати хвилину (у другому рядку дисплея), після досягнення якої повинен включитися зумер.