

Лабораторна робота №1

Виведення статичної інформації на індикатори

Тема: виведення статичної інформації на 7ми- та 16ти-сегментні індикатори за допомогою восьмибітного контролера Atmega128.

Мета: дослідити основні методи виведення статичної інформації на індикатори.

Теоретичні відомості

Сегментний індикатор — індикатор, елементи відображення якого є сегментами, згрупованими в одне або кілька знакомиць.

Сегментом називається елемент відображення інформації знаковинтезувального індикатора, контур якого являє собою прямі та (або) криві лінії. На відміну від матричного індикатора, в якому всі елементи зображення однакові за формою, в сегментному індикаторі кожен сегмент унікальний. Форма і положення сегментів на індикаторі розробляється спеціально для передачі певного набору символів або знаків. Символи на таких індикаторах формуються сукупністю кількох сегментів.

Основна відмінність сегментного індикатора від матричного — це порівняно невелика кількість елементів індикації і відповідно спрощена схема управління. Додатковий 8-й світлодіод іноді використовується в тому самому корпусі, що дозволяє вказувати десяткову крапку (DP), коли два або більше 7-сегментних дисплеїв з'єднані разом для відображення чисел більше десяти.

Випромінювання фотонів від 7-сегментного дисплея відбувається, коли діодний перехід кожного сегмента зміщується зовнішньою напругою, що дозволяє струму протікати через його перехід, і в електроніці ми називаємо цей процес електролюмінесценцією. Фактичний колір видимого світла, випромінюваного світлодіодом, у діапазоні від синього до червоного та помаранчевого, визначається спектральною довжиною хвилі випромінюваного світла, яка сама по собі залежить від суміші різних домішок, доданих до напівпровідникових матеріалів, що використовуються для його виробництва. Світлодіоди мають багато переваг перед традиційними лампочками та лампами, головними з яких є їхній малий розмір, тривалий термін служби, різноманітні кольори, дешевизна та доступність, а також легкість сполучення з різноманітними іншими електронними компонентами та цифровими схемами. Кожен із семи світлодіодів на дисплеї має позиційний сегмент, один з його з'єднувальних контактів витягується прямо з прямокутної пластикової упаковки. Ці індивідуальні світлодіодні контакти позначені від а до g, що означає кожен окремий світлодіод. Інші світлодіодні контакти з'єднані між собою та утворюють загальний контакт.

Таким чином, зміщуючи відповідні контакти світлодіодних сегментів у певному порядку, деякі сегменти будуть світлими, а інші темними, дозволяючи генерувати на дисплеї бажаний шаблон символів числа. Це дозволяє нам відображати кожен з десяти десяткових цифр від 0 до 9 на тому самому 7-сегментному дисплеї. Різниця між двома дисплеями, як впливає з їх назви, полягає в тому, що загальний катод має всі катода 7-сегментів, з'єднані безпосередньо разом, а загальний анод має всі аноди 7-сегментів, з'єднані разом.

1. Загальний катод (CC) – на дисплеї із загальним катодом усі катодні з'єднання сегментів світлодіодів з'єднані разом із логічним «0» або землею. Окремі сегменти підсвічуються

шляхом застосування сигналу «HIGH» або логічної «1» через резистор обмеження струму для прямого зміщення окремих клем анода (ag).

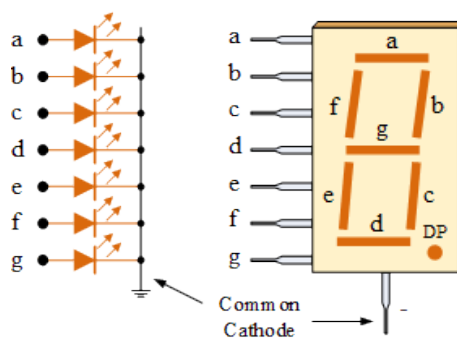


Рис1.1 – Спільний катод.

2. Загальний анод (СА) – на дисплеї із загальним анодом усі анодні з’єднання сегментів світлодіодів об’єднані разом за логікою «1». Окремі сегменти підсвічуються шляхом подачі заземлення, логічного сигналу «0» або «LOW» через відповідний резистор обмеження струму на катод певного сегмента (ag).

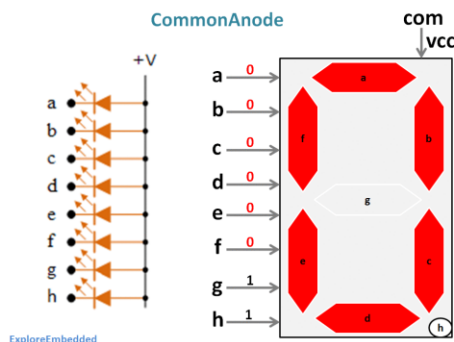


Рис.1.2 – спільний анод.

Загалом дисплеї зі звичайним анодом є більш популярними, оскільки багато логічних схем можуть споживати більше струму, ніж вони можуть отримати. Також зауважте, що дисплей із загальним катодом не є прямою заміною в схемі дисплея із загальним анодом і навпаки, оскільки це те саме, що підключення світлодіодів у зворотному порядку, і, отже, випромінювання світла не відбуватиметься.

Залежно від десяткової цифри, яка буде відображатися, конкретний набір світлодіодів має пряме зміщення. Наприклад, щоб відобразити числову цифру 0, нам потрібно буде засвітити шість світлодіодних сегментів, що відповідають a, b, c, d, e і f. Таким чином, різні цифри від 0 до 9 можна відобразити за допомогою 7-сегментного дисплея, як показано.

Керування 7-сегментним дисплеєм

Хоча 7-сегментний дисплей можна розглядати як один дисплей, це все одно сім окремих світлодіодів в одному корпусі, і тому ці світлодіоди потребують захисту від перевантаження по струму. Світлодіоди випромінюють світло лише тоді, коли воно має пряме зміщення, причому кількість випромінюваного світла пропорційна прямому струму.

Це означає, що інтенсивність світла світлодіодів зростає приблизно лінійно зі збільшенням струму. Таким чином, цей прямий струм повинен контролюватися та

обмежуватися до безпечного значення за допомогою зовнішнього резистора, щоб запобігти пошкодженню сегментів світлодіодів.

Пряме падіння напруги на сегменті червоного світлодіода є дуже низьким, приблизно від 2 до 2,2 вольт (сині та білі світлодіоди можуть досягати 3,6 вольт), тому для правильного освітлення сегменти світлодіода повинні бути підключені до джерела напруги в перевищення цього значення прямої напруги з послідовним опором, який використовується для обмеження прямого струму до бажаного значення.

Як правило, для стандартного 7-сегментного дисплея червоного кольору кожен світлодіодний сегмент може споживати близько 15 мА для правильного освітлення, тому на 5-вольтовій цифровій логічній схемі значення струмообмежувального резистора буде приблизно 200 Ом $(5 \text{ В} - 2 \text{ В})/15 \text{ мА}$, або 220 Ом до найближчого вищого бажаного значення.

Шістнадцятисегментний дисплей (SISD) — це тип дисплея, заснований на шістнадцяти сегментах, які можна вмикати або вимикати для створення графічного візерунка. Це розширення більш поширеного семисегментного дисплея, додавання чотирьох діагональних і двох вертикальних сегментів і розділення трьох горизонтальних сегментів навпіл. Інші варіанти включають чотирнадцятисегментний дисплей, який не розділяє верхній і нижній горизонтальні сегменти, і двадцятидвохсегментний дисплей [1], який дозволяє символи нижнього регістру з спадними знаками.

Часто генератор символів використовується для перекладу 7-бітових кодів символів ASCII у 16 бітів, які вказують, який із 16 сегментів увімкнути або вимкнути.

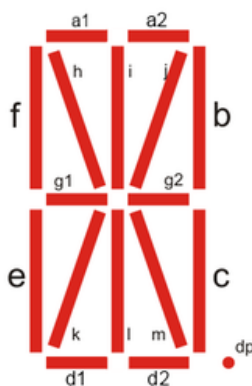


Рис.1.3 – Загальний вигляд шістнадцятисегментного дисплею.

Шістнадцятисегментні дисплеї спочатку були розроблені для відображення буквено-цифрових символів (латинських літер і арабських цифр). Пізніше їх використовували для відображення тайських цифр і перських символів. Неелектронні дисплеї, що використовують цей шаблон, існували ще в 1902 році.

До появи недорогих матричних дисплеїв шістнадцяти та чотирнадцятисегментні дисплеї використовувалися для створення буквено-цифрових символів на калькуляторах та інших вбудованих системах. Пізніше вони використовувалися на відеомагнітофонах (VCR), DVD-програвачах, мікрохвильових печах, автомобільних стереосистемах, телефонних дисплеях Caller ID та ігрових автоматах.

Шістнадцятисегментні дисплеї можуть базуватися на одній із кількох технологій, три найпоширеніші типи оптоелектроніки : LED , LCD і VFD . Світлодіодний варіант зазвичай виготовляється в пакетах з одним або двома символами, які за потреби об'єднуються в текстові рядкові дисплеї відповідної довжини для відповідного застосування; їх також можна скласти для створення багаторядкових дисплеїв.

Як і в семи- та чотирнадцятисегментних дисплеях, десяткова точка та/або кома можуть бути присутніми як додатковий сегмент або пара сегментів; кома (використовується для групування трьох цифр або як десятковий роздільник у багатьох регіонах) зазвичай утворюється шляхом поєднання десяткової крапки з тісно «прикріпленим» сегментом у формі дуги, що спадає зліва. Таким чином, крапка або кома можуть відображатися між позиціями символів замість того, щоб займати цілу позицію окремо, що було б у випадку використання нижнього середнього вертикального сегмента як крапки та нижнього лівого діагонального сегмента як коми.



Рис.1.4 - Арабські цифри , літери базового латинського алфавіту ISO та знаки пунктуації на типовому 16-сегментному дисплеї

Найчастіше використовуються два типи сегментних індикаторів:

- Цифровий семисегментний індикатор, що має вісім елементів — сім сегментів для індикації цифри і один — для крапки.
- Цифро-літерний індикатор, що має дев'ять, чотирнадцять або шістнадцять сегментів. Такі індикатори мають можливість показати більшість символів латинського алфавіту та кирилиці, не рахуючи цифр і спеціальних знаків.

Приклад №1

Принципова електрична схема, де 7-сегментний індикатор підключений до мікроконтролера ATmega128. Виходи з індикатора підключені до виводів PB0–PB7 мікроконтролера (порт В). Схема демонструє з'єднання між пінів мікроконтролера та 7-сегментного дисплея, що дозволяє керувати сегментами індикатора для відображення цифр. Код наведено в додатку 1.

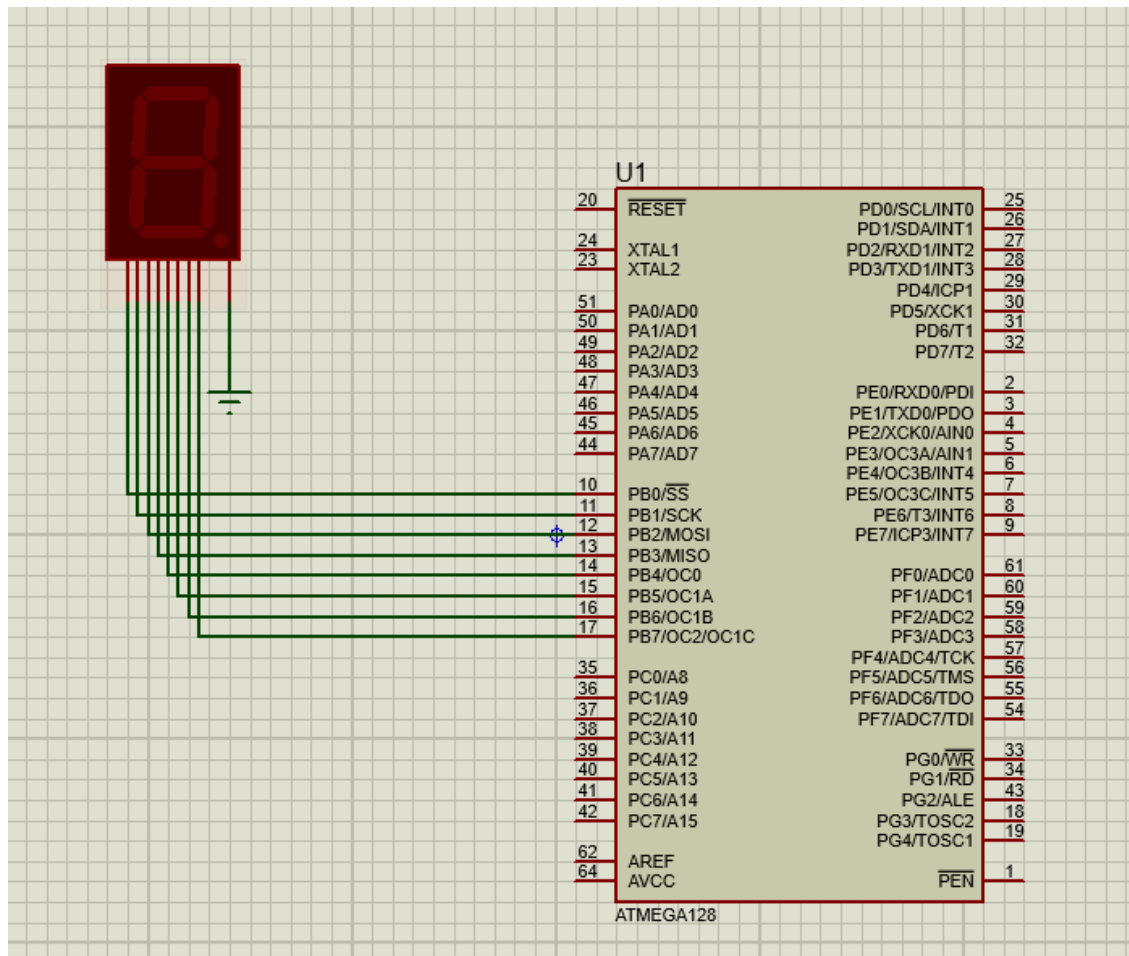


Рис.1.5 – Принципово-електрична схема для 7-місегментного індикатора.

Приклад №2

Принципова електрична схема, де до кожного сегмента підключено окремий дріт, і всі вони підключені до виводів мікроконтролера ATmega128. 16-сегментний індикатор, як правило, має 16 входів, які відповідають за керування кожним із сегментів. Ці сегменти поділяються на дві групи: Сегменти А-Н — керуються за допомогою одного порту. Сегменти І-Р — керуються через інший порт.

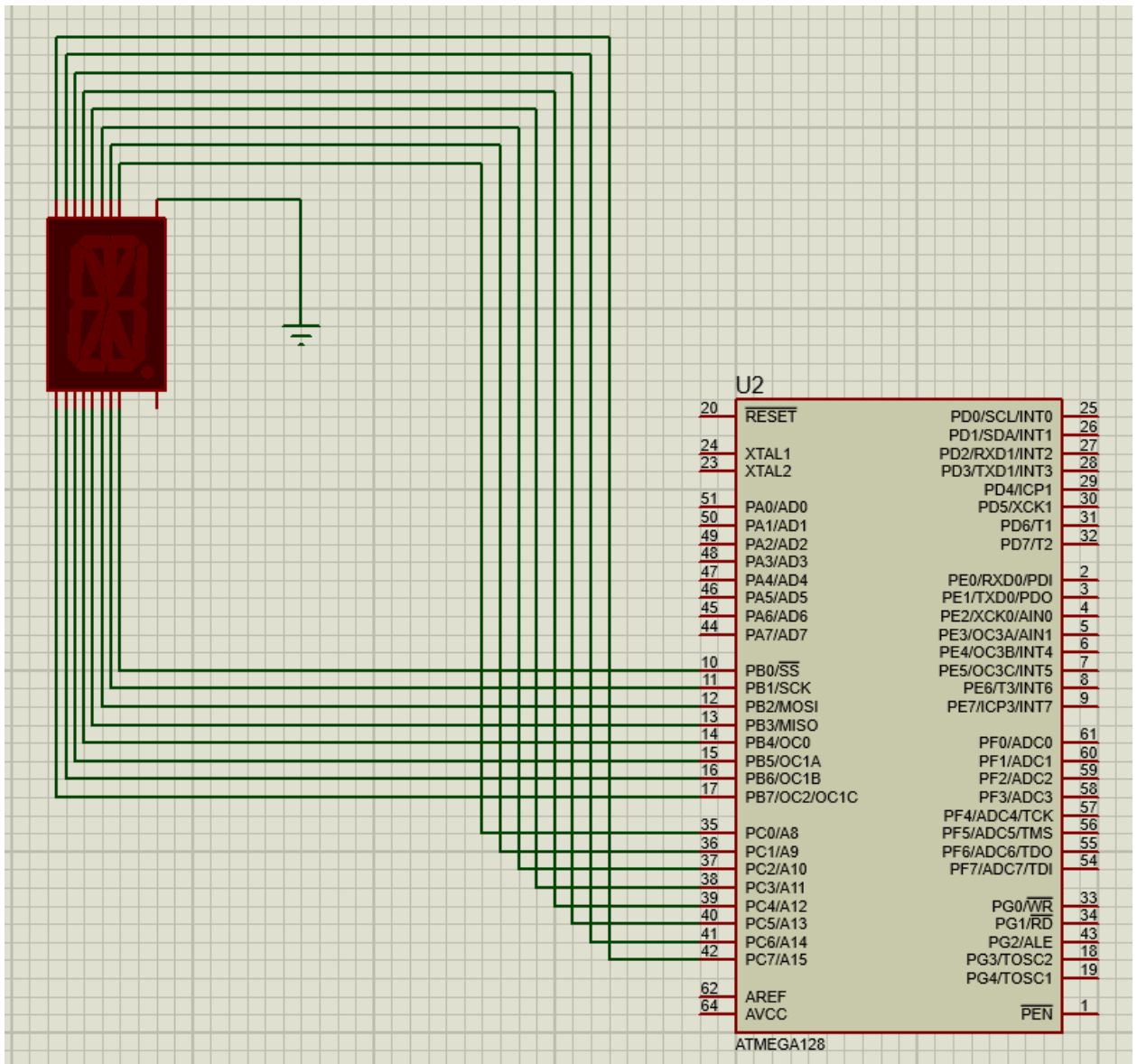


Рис.1.6 – Принципово-електричка схема для 1битсегментного індикатора.

1.2. Порядок виконання лабораторної роботи

1. Ознайомитись з теоретичними відомостями.
2. Зібрати схему підключення 4х семисегментних індикаторів використовуючи вільні порти мікроконтролера та вивести на них дату свого народження у фортматі DD.MM
3. Зібрати схему підключення 3х шістнадцятисегментних індикаторів використовуючи вільні порти мікроконтролера та вивести на них свої ініціали.
4. Зробити висновки

```
// Виведення цифр від 0-9 послідовно

#include <avr/io.h>
#include <util/delay.h>

// Таблиця значень для 7-сегментного індикатора (спільний катод)
unsigned char digits[] = {
    0b00111111, // 0
    0b00000110, // 1
    0b01011011, // 2
    0b01001111, // 3
    0b01100110, // 4
    0b01101101, // 5
    0b01111101, // 6
    0b00000111, // 7
    0b01111111, // 8
    0b01101111  // 9
};

void display_digit(uint8_t digit) {
    // Виводимо значення на PORTB
    PORTB = digits[digit];
}

int main(void) {
    // Налаштовуємо PORTB як вихід
    DDRB = 0xFF; // Всі біти PORTB на вихід (сегменти індикатора)

    while (1) {
        for (uint8_t i = 0; i < 10; i++) {
            display_digit(i); // Відображаємо цифру від 0 до 9
            _delay_ms(1000);  // Затримка 1 секунда між цифрами
        }
    }
}
```

```
#include <avr/io.h>
#include <util/delay.h>

// Значення для кожної букви на 16-сегментному індикаторі (катодний
тип)
// Перші 8 сегментів (a-h) керуються через PORTC
unsigned char letters_PORTC[] = {
    0b00110000, // 'A' - Сегменти b, c, увімкнені
    0b00000000 // 'Y' - Сегменти не вмикаються
    // Інші символи або цифр
};

// Сегменти i-p керуються через PORTB
unsigned char letters_PORTB[] = {
    0b00110010, // 'A' - Сегменти j, k, g2 увімкнені
    0b10100100 // 'Y' - Сегменти h, j, i увімкнені
    // Інші символи або цифр
};

// Функція для виведення символу на 16-сегментний індикатор
void display_letter(uint8_t letter) {
    // Виводимо перші 8 сегментів (a-h) на PORTB
    PORTB = letters_PORTB[letter];

    // Виводимо інші 8 сегментів (i-p) на PORTC
    PORTC = letters_PORTC[letter];
}

int main(void) {
    // Налаштовуємо PORTB і PORTC як виходи
    DDRB = 0xFF; // Всі біти PORTB на вихід
    DDRC = 0xFF; // Всі біти PORTC на вихід

    while (1) {
        // Виводимо букву 'A'
        display_letter(0); // A в масиві знаходиться на індексі 0
        _delay_ms(1000); // Затримка 1 секунда

        // Виводимо букву 'Y'
        display_letter(1); // B в масиві знаходиться на індексі 1
        _delay_ms(1000); // Затримка 1 секунда
    }
}
```