

Лабораторна робота № 2

ОРГАНІЗАЦІЯ ДИНАМІЧНОЇ ІНДИКАЦІЇ

Мета роботи: Вивчення особливостей роботи динамічних цифрових індикаторів. Розробка програм для мікроконтролера AT90S2313 відображення інформації на індикаторах динамічного типу.

Теоретичні відомості

1 Організація динамічної індикації

Дуже часто МК використовується не тільки для керування роботою конструкції, але й для того, щоб повідомити що-небудь користувачеві і/або отримати від нього які-небудь вказівки про подальшу роботу. Наприклад, електронний годинник, крім власне відліків часу, повинен його ще відображати, а також дозволяти змінювати покази (встановлювати точний час). Якщо вся "інформація" зводиться до мигання парю світлодіодів, яких-небудь спеціальних зусиль з відображення інформації з боку розробника конструкції не вимагається, але якщо таких світлодіодів виявляється два-три десятки, тут вже потрібне застосування додаткових засобів (як апаратних, так і програмних). Як правило, в цьому випадку відображення інформації виконують у режимі динамічної індикації – це найбільш економний за кількістю використовуваних ліній спосіб.

Для організації динамічної індикації застосовується матриця, що складається з ліній рядків і ліній стовпців (рис. 1). На перетині стовпця і рядка матриці розташований індикаторний елемент – світлодіод.

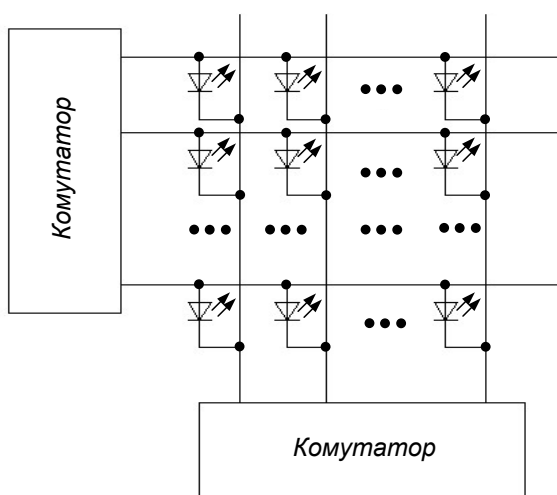


Рисунок 1 – Організація динамічної індикації

Для того, щоб запалити той або інший елемент, необхідно подати на матрицю не один, як у звичайних індикаторах, а два сигнали: логічна 1 на

відповідному рядку і логічний 0 на відповідному стовпці матриці. Через односторонню провідність світлодіода кожна комбінація сигналів на входах рядків і стовпців однозначно включає рівно один індикаторний елемент.

Головна перевага динамічної індикації – невелике число ліній, що керують: для матриці світлодіодів розміром $N \times N$ елементів потрібно всього $2N$ сигналів, що керують. За таку економію, втім, доводиться платити – справа в тому, що при почерговому виведенні інформації на кожен світлодіод матриці його яскравість світіння буде в N^2 разів нижча, ніж при безпосередньому виведенні інформації на один світлодіод, що "окремо стоїть". Тому в пристроях, що використовують динамічну індикацію, виведення інформації здійснюється не на кожен світлодіод окремо, а на один рядок або на один стовпець повністю – у цьому випадку яскравість світіння світлодіодів падає тільки в N разів.

Розглянемо декілька варіантів реалізації динамічної індикації із застосуванням МК. Як індикаторний елемент передбачається застосування семисегментних індикаторів, але кожен такий індикатор з легкістю можна замінити і групою світлодіодів.

Схема реалізації динамічної індикація без додаткових елементів наведена на рисунку 2. До порту B МК підключені катооди всіх світлодіодів матриці, а до порту A – аноди кожного з індикаторів, що створюють матрицю. На лініях порту A організовується одиниця, що "біжить". На лінії порту B при кожному положенні одиниці, що біжить, виводиться семисегментний код того символу, який повинен горіти в даному знакомісці. Для індикаторів із загальним катодом замість одиниці, що біжить, використовується нуль, що біжить. Перевага такого способу індикації – у відсутності яких-небудь додаткових компонентів (окрім самих світлодіодних індикаторів), головний недолік – значна перевитрата ліній портів. Таке рішення для МК може забезпечити роботу не більше 5 семисегментних індикаторів одночасно, і при цьому МК стає "глухим і сліпим", оскільки жодної вільної лінії у нього не залишається. При використанні інших МК з великою кількістю ніжок вказана проблема знімається.

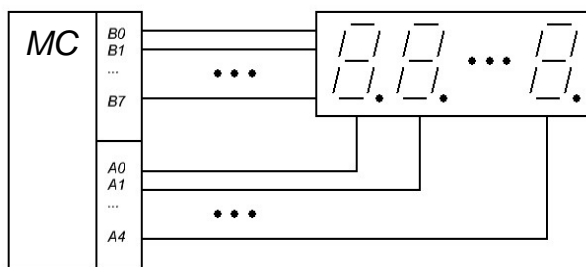


Рисунок 2 – Схема реалізації динамічної індикація без додаткових елементів

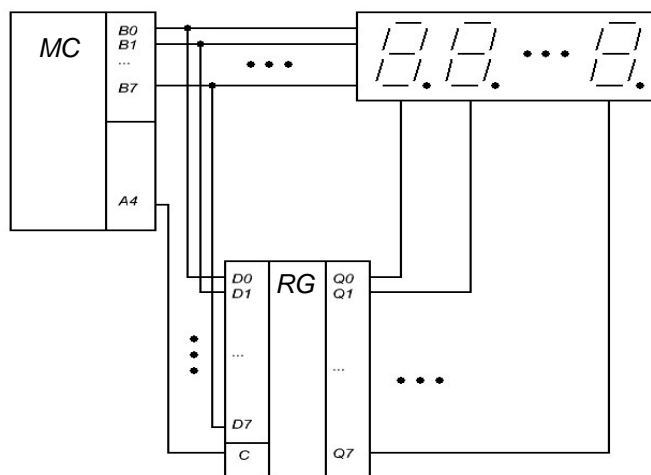


Рисунок 3 – Схема реалізації динамічної індикації з одним додатковим елементом

Схема реалізації динамічної індикації з одним додатковим елементом наведена на рисунку 3. До порту *B* МК підключені катоди всіх світлодіодів матриці, а також – входи буферного регістра. До виходів же буферного регістра підключені аноди кожного з семисегментних індикаторів. На порт *B* спочатку виводиться одиниця, що біжить, яка записується в буферному регістрі сигналом *C*, що утворюється однією з ліній порту *A*. Потім на лінії порту *B* видається семисегментний код символу, який повинен горіти в знакомісті, визначуваному сигналом на виході буферного регістра. В даному випадку лінії порту *B* використовуються в режимі часового мультиплексування, тобто, ними по черзі передається і код символу, і номер знакомістя. Перевага такого способу індикації – менша витрата ліній портів МК (окрім порту *B* – всього одна додаткова лінія) і можливість роботи до 8 індикаторів одночасно. Ще один варіант індикації з одним додатковим елементом наведений на рисунку 4.

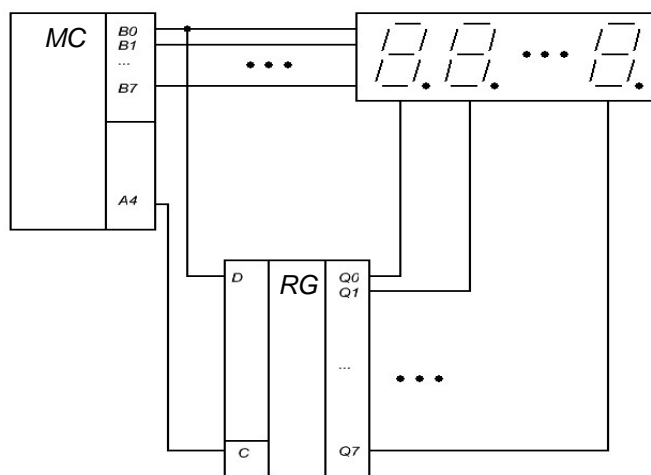


Рисунок 4 – Схема реалізації динамічної індикації з регістром зсуву

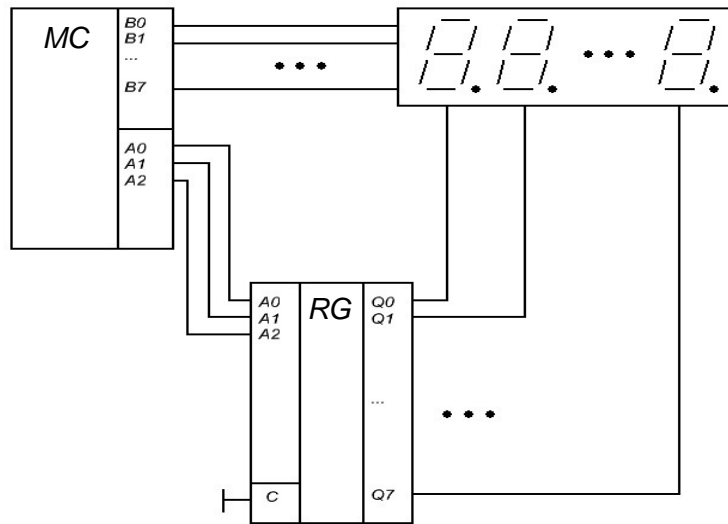


Рисунок 5 – Схема реалізації динамічної індикації з дешифратором

У цьому варіанті одиниця, що біжить, реалізується за допомогою регістра зсуву. Порт *B* у цій схемі також використовується в режимі часового мультиплексування – як для видачі символу, так і для занесення чергового біта до регістра зсуву. Схема також вимагає всього одну додаткову лінію (крім порту *B*), але за габаритами виходить трохи менше попереднього варіанта. Ще один варіант реалізації схеми з одним додатковим елементом наведений на рисунку 5. Така схема придатна тільки для індикаторів із загальним катодом, оскільки для організації нуля, що біжить, в ній використовується дешифратор. Схема вимагає три додаткові лінії (крім порту *B*), проте у багатьох випадках вона може виявитися цілком корисною.

Схема реалізації динамічної індикації з двома додатковими елементами наведена на рисунку 6.

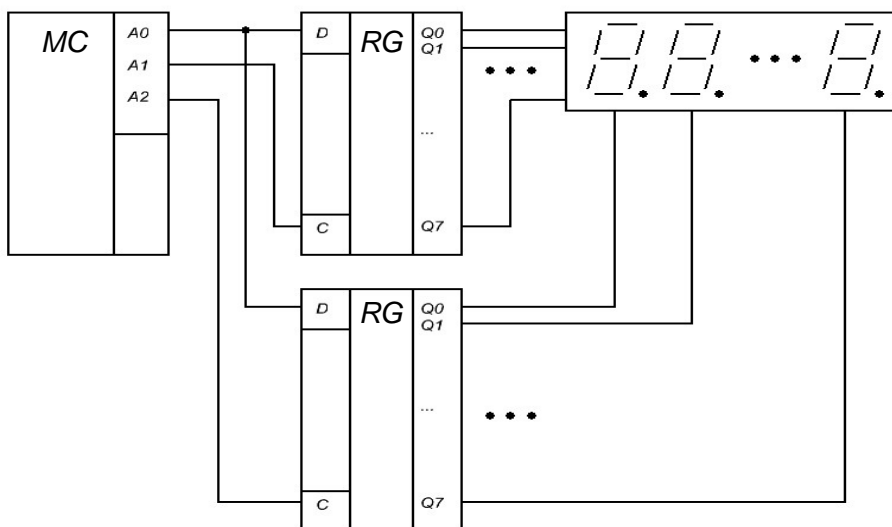


Рисунок 6 – Схема реалізації динамічної індикації з двома додатковими елементами

У схемі використовуються два послідовні регістри зсуву: один – для розгортки зображення по стовпцях (замінює порт *A*), інший – для розгортки зображення по рядках (замінює порт *B*). Перевага такого рішення – всього три лінії порту МК. Крім того, при такому варіанті реалізації блок динамічної індикації легко оформити і у вигляді окремої плати. Недолік такого рішення – два додаткових компоненти.

Ще одна схема реалізації наведена на рисунку 7. Перевага такої схеми – всього дві лінії порту. Недолік – складніша програма керування і велика тривалість формування вихідних сигналів, що викликає деяке паразитне підсвічування індикаторів (помітно лише в темряві).

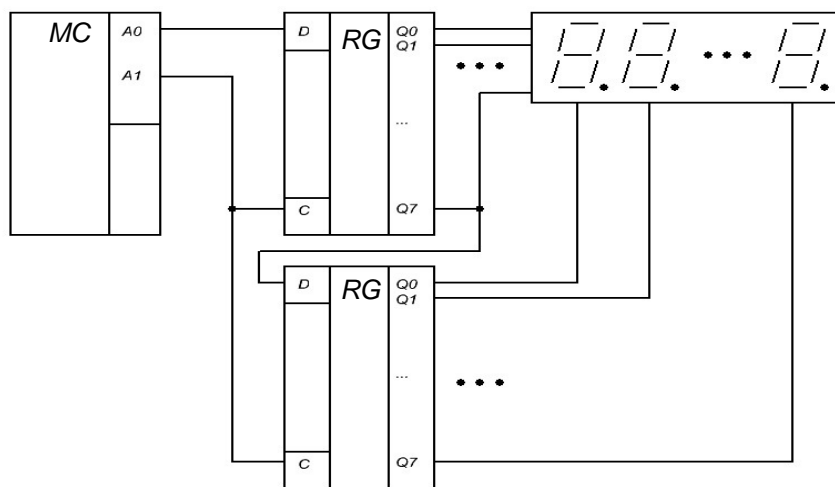


Рисунок 7 – Схема динамічної індикації з двома лініями керування

Практична схема підключення динамічного індикатора на МК AT90S2313 за схемою, що наведена на рисунку 5, зображена на рисунку 11. Символ, що буде світитись на індикаторі визначається сигналом логічної одиниці на вивід COM1, 2, 3, 4 індикатора. На шині даних (*A*, *B*, *C* ...) активним рівнем є рівень логічного 0.

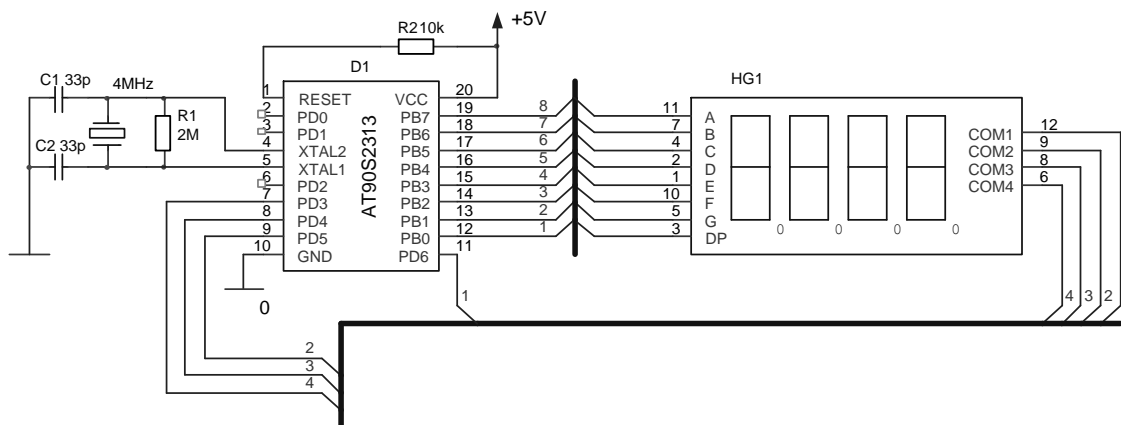


Рисунок 8 – Підключення динамічного індикатора до МК AT90S2313

2 Масиви чисел та їх застосування в мікроконтролерах AVR

У будь-якому місці програми можемо написати:

```
MyArray:  
.db 1,15,4,9,12,145,67,90
```

Це є масив "MyArray", що складається з 8 елементів. Масив розміщується у пам'яті програм починаючи з адреси, на якій стоїть мітка (MyArray). Тобто, адресі мітки відповідає 0-й елемент масиву.

Щоб отримати доступ до 1-го, 2-го, 3-го і т.д. елементу, потрібно прочитати осередок FLASH пам'яті програм за адресою, яка більше адреси мітки відповідно на 1, 2, 3. Розглянемо програму, яка читає послідовно елементи масиву MyArray і виводить їх в порт D.

```
    ldi Temp,0           ;ініціалізація регістра  
                        ;внутрішньої адресації масиву  
ReadArray:  
    ldi ZH,High(MyArray*2) ;завантаження адреси 0-го  
    ldi ZL,Low(MyArray*2)  ;елементу у рег. пару Z  
    ldi Temp1,0  
    add ZL,Temp            ;додавання  
    adc ZH,Temp1          ;внутр. адреси  
    lpm                   ;завантаження з пам'яті програм  
    mov Temp1,R0          ;копіювання  
    out PortD,Temp1      ;вивід у порт  
    inc Temp              ;збільш. внутр. адреси  
    rjmp ReadArray       ;на початок циклу  
MyArray:  
.db 12,16,3,4,10,17,255,37,158,14,13,98  
.db 14,85,30,9,145,52,64,49,119,72,209,46
```

Розглянемо структуру:

```
    ldi ZH,High(MyArray*2)
```

Командою завантажуюємо старшу частину адреси в старшу частину пари Z (ZH) по мітці MyArray. Що значить "*2"? Кожна команда містить два байти інформації і займає, таким чином, два осередки ПЗП. Тому, лічильник команд рахує 2 адреси як одну. Мітка містить саме дані для лічильника команд. Щоб одержати реальну адресу ПЗП, необхідно збільшити адресу мітки в 2 рази. Що і робимо. По-друге: у масиві два рядки. Це не означає, що масив - двомірний. Просто, розбиваємо один довгий рядок на декілька коротших - для зручності. Рядків може бути скільки завгодно. *При розбитті, не допускається непарної кількості елементів в рядку, інакше після останнього елементу поточного рядка перед першим елементом наступної, читається один "неіснуючий" елемент.* Це також пов'язано з

особливостями адресації програмної пам'яті. Непарним може бути тільки останній рядок. По-третє, дані у масиві можуть бути представлені у вигляді десяткових, шістнадцятиричних, двійкових чисел, а також, еквівалентних значень символів ASCII.

.db 3,4,5,75,32,12 ; десятикові числа
.db 0x2A,0x34,0x17,0xDF ; шістнадцяткові числа
.db 0b01101001,0b11011100 ; двійкові числа
.db "Тут був Вася" ; еквіваленти ASCII

Завдання до лабораторної роботи

1. Запустити в Proteus файл Din_IND. Підключити файл Din_IND.HEX. Перевірити як працює схема. Розібрати особливості програми Din_IND.ASM. Скласти алгоритм роботи програми.
2. Запустити в Proteus файл Din_IND_KLAV. Підключити файл INDICA~1.HEX. Перевірити як працює схема. Розібрати особливості програми INDICATE+KEYBOARD.ASM. Скласти алгоритм роботи програми.
3. Запустити в Proteus файл Din_IND. Підключити файл DIN_OGON.HEX. Перевірити як працює схема. Розібрати особливості програми DIN_OGON.ASM. Підключити файл DIN_TEN.HEX. Перевірити як працює схема. Розібрати особливості програми DIN_TEN.ASM. Підключити файл. DIN_TO.HEX. Перевірити як працює схема. Розібрати особливості програми DIN_TO.ASM.
4. Використовуючи програми DIN_OGON.ASM, DIN_TEN.ASM, DIN_TO.ASM написати програму, яка створює ефект «вогник, що біжить» по зовнішньому колу (для непарних варіантів) та ефект «тінь, що біжить» по зовнішньому колу (для парних варіантів) – оцінка задовільно. Виконати п.4 для 6 динамічних індикаторів – оцінка добре. Виконати п.4 для 6 динамічних індикаторів з використанням масивів – оцінка відмінно.

Зміст звіту

1. Алгоритм роботи програми та схема до п.1
2. Алгоритм роботи програми та схема до п.2
3. Схема, алгоритм, програма та результати виконання п.4 або п.5 у середовищі Proteus

Приклад виконання завдання

Програма, що створює ефект «вогник, що біжить» по зовнішньому колу для 4 розрядного динамічного індикатора (для схеми Din_IND)

```
.include "2313def.inc"
.def Temp=R18
.def Temp1=R19
ldi R20,0b11111111      ;настройка портів
out ddrb,R20
ldi R20,0b00111111
out ddrd,R20
ldi R16,RamEnd          ;ініціалізація стека
out SPL,R16
ldi Temp,0              ;ініціалізація регістра
                          ;внутрішньої адресації масиву

cli
CLR R27
LDI R26,$60

ReadArray:
ldi ZH,High(MyArray*2) ;завантаження адреси 0-го
ldi ZL,Low(MyArray*2)  ;елементу в рег. пару Z
ldi Temp1,0
add ZL,Temp             ;додавання
adc ZH,Temp1            ;внутр. адреса

lpm                     ;завантаження з пам'яті програм
ST X, R0                ;копіювання
INC R26
inc Temp                 ;збільшення внутр. адреси
CPI TEMP,45
BRNE ReadArray          ;на початок циклу
RJMP IndicCycle

MyArray:
.db $0E,$09,$09,$39
.db $0D,$09,$09,$39
.db $0B,$09,$09,$39
.db $0F,$01,$09,$39
.db $0F,$09,$01,$39
.db $0F,$09,$09,$31
.db $0F,$09,$09,$29
.db $0F,$09,$09,$19
.db $0F,$09,$09,$38
.db $0F,$09,$08,$39
.db $0F,$08,$09,$39

;*****
;MAIN
;*****
IndicCycle:  LDI R26,$60
              rcall Display      ;цикл індикації
              rjmp IndicCycle
;*****
;послідовний вивід на динамічний індикатор
Display:
```


...
RET

Контрольні запитання

1. Розкажіть про принцип динамічної індикації та наведіть його схему.
2. Наведіть схемну реалізацію динамічної індикації без додаткових елементів.
3. Наведіть схемну реалізацію динамічної індикації з одним додатковим елементом.
4. Наведіть схемну реалізацію динамічної індикації з регістром зсуву.
5. Наведіть схемну реалізацію динамічної індикації з дешифратором.
6. Наведіть схемну реалізацію динамічної індикації з двома лініями керування.
7. Для яких задач доцільно використовувати масиви чисел у програмі?
8. Наведіть масив чисел для перекодування двійкового числа від 0 до 15 в код семисегментного індикатора з загальним катодом.