

# Основи мікропроцесорної техніки

Цикл 3 Мікроконтролери AVR

**Методичні вказівки  
до виконання лабораторних робіт**

для студентів напрямів підготовки  
6.050803 "Акустотехніка" та 6.050903 "Телекомунікації"  
усіх форм навчання

*Затверджено методичною радою НТУУ „КПІ”  
22 вересня 2011*

КИЇВ, НТУУ „КПІ”

2011

УДК 621.3.011 (075.8)  
ББК 31.211я73 Г94

Основи мікропроцесорної техніки [Текст]: Метод.вказівки до викон. лаборатор. Робіт для студентів напрямів підготов. 6.50803 „Акустотехніка” та 6.50903 „Телекомунікації” усіх форм навчання/ Укладачі: В. Б. Швайченко, Д.В. Тітков. – К.: НТУУ „КПІ”, 2011. -82 с.

*Гриф надано Методичною радою НТУУ „КПІ”  
(протокол №\_\_\_ від \_\_\_. \_\_\_.2011)*

Навчальне видання

## **Основи мікропроцесорної техніки**

**Цикл 3 Мікроконтролери AVR**

### **Методичні вказівки**

**до виконання лабораторних робіт**

**для студентів напрямів підготовки**

**6.050803 “Акустотехніка” та 6.050903 “Телекомунікації”  
усіх форм навчання**

Укладачі: *Швайченко Володимир Борисович, канд.техн.наук, доц.*

*Тітков Дмитро Валерійович*

Відповідальний

редактор *В.В. Пілінський, канд.техн.наук, проф.*

Рецензент *Т.О.Терещенко, д-р. техн.наук, проф.*

*За редакцією укладачів*

*Надруковано з оригінал-макета замовника*

## ЗМІСТ

|                                                                                                             |    |
|-------------------------------------------------------------------------------------------------------------|----|
| Вступ .....                                                                                                 | 4  |
| Загальні вимоги, щодо оформлення лабораторних робіт<br>з дисципліни “Основи мікропроцесорної техніки” ..... | 5  |
| Витяг з РСО щодо оцінювання лабораторних робіт.....                                                         | 5  |
| Лабораторна робота №1 .....                                                                                 | 6  |
| Лабораторна робота №2 .....                                                                                 | 10 |
| Лабораторна робота №3 .....                                                                                 | 13 |
| Лабораторна робота №4 .....                                                                                 | 16 |
| Лабораторна робота №5 .....                                                                                 | 20 |
| Лабораторна робота №6 .....                                                                                 | 23 |
| Лабораторна робота №7 .....                                                                                 | 26 |
| Лабораторна робота №8 .....                                                                                 | 28 |
| Перелік посилань .....                                                                                      | 32 |

## **Вступ**

Даний курс лабораторних робіт призначений для отримання початкових практичних навичок роботи з мікроконтролерами серії AVR фірми Atmel.

В якості лабораторного стенду використовується налагоджувальний модуль, який працює на базі мікроконтролера ATmega16. Цей налагоджувальний модуль дозволяє проводити налагодження програмного й апаратного забезпечення мікроконтролерів. На лабораторному стенді також можна вивчати структуру системи команд, схемотехнічне та програмне забезпечення, порядок функціонування й методи програмування МК ВІС серії AVR.

## **Загальні вимоги щодо оформлення протоколів лабораторних робіт**

### **Протокол повинний містити такі структурні елементи:**

- 1) титульний аркуш із зазначенням вузу, кафедри, прізвища, імені і по батькові студента, що виконав лабораторну роботу, його групу, а також прізвище керівника лабораторних занять;
- 2) мету роботи;
- 3) відповіді на контрольні питання;
- 4) схеми, таблиці, текст програм, рисунки у відповідності з вимогами завдань;
- 5) лістинги програм
- 6) висновки за результатами виконання кожного завдання.

### **Витяг з РСО щодо оцінювання лабораторних робіт**

#### **Лабораторні роботи**

Ваговий бал – 6. Максимальна кількість балів за всі лабораторні роботи дорівнює  $6 \text{ балів} \times 16 = 96 \text{ балів}$ .

Критерії оцінювання з визначенням двох-трьох рівнів за кожний етап:

##### **підготовка до роботи (2 бали):**

перший рівень – знає відповіді на більшість контрольних питань з теорії та підготовлено проект протоколу (2 бали),

другий рівень – знає відповіді на деякі контрольні питання з теорії та підготовлено проект протоколу (1 бали),

третій рівень - не знає відповіді на контрольні питання з теорії або не підготовлено проект протоколу (0 балів);

##### **виконання лабораторної роботи (2 бали):**

перший рівень – приймає особисту участь у вимірюванні, обробці та оформленні результатів (2 бали),

другий рівень– приймає особисту участь у вимірюванні або обробці та оформленні результатів (1 бал),

третій рівень – не приймає особисту участь у вимірюванні, обробці та оформленні результатів (0 балів);

##### **якість захисту роботи (2 бали):**

перший рівень – у висновках зроблено порівняння отриманих практичних результатів з положенням теорії, якість оформлення протоколу відповідає вимогам ДСТУ, надано відповідь на додаткове питання за темою (2 бали)

другий рівень – у висновках зроблено порівняння отриманих практичних результатів з положенням теорії та якість оформлення протоколу частково відповідає вимогам ДСТУ, надано відповідь на додаткове питання за темою (1бал)

третій рівень – у висновках не зроблено порівняння отриманих практичних результатів з положенням теорії, або якість оформлення протоколу частково відповідає вимогам ДСТУ, або не надано відповідь на додаткове питання за темою (0 балів).

# Лабораторна робота №1

## Вивчення архітектури мікроконтролерів фірми ATMEL

**Мета роботи** - Ознайомитися з принципами побудови та роботи мікроконтролерів (МК) типу AVR фірми Atmel.

### Домашнє завдання

Ознайомитися з особливостями мікроконтролерів фірми Atmel. Повторити особливості RISC та CISC архітектури, призначення регістрів, тактові частоти, АЛП, ОЗП, ПЗП та інших структурних вузлів. Згадати особливості інтерфейсів, які використовують у мікропроцесорній техніці.

### Теоретичні відомості

Схему типового представника мікро контролерів фірми Atmel представлено на рис.1.1

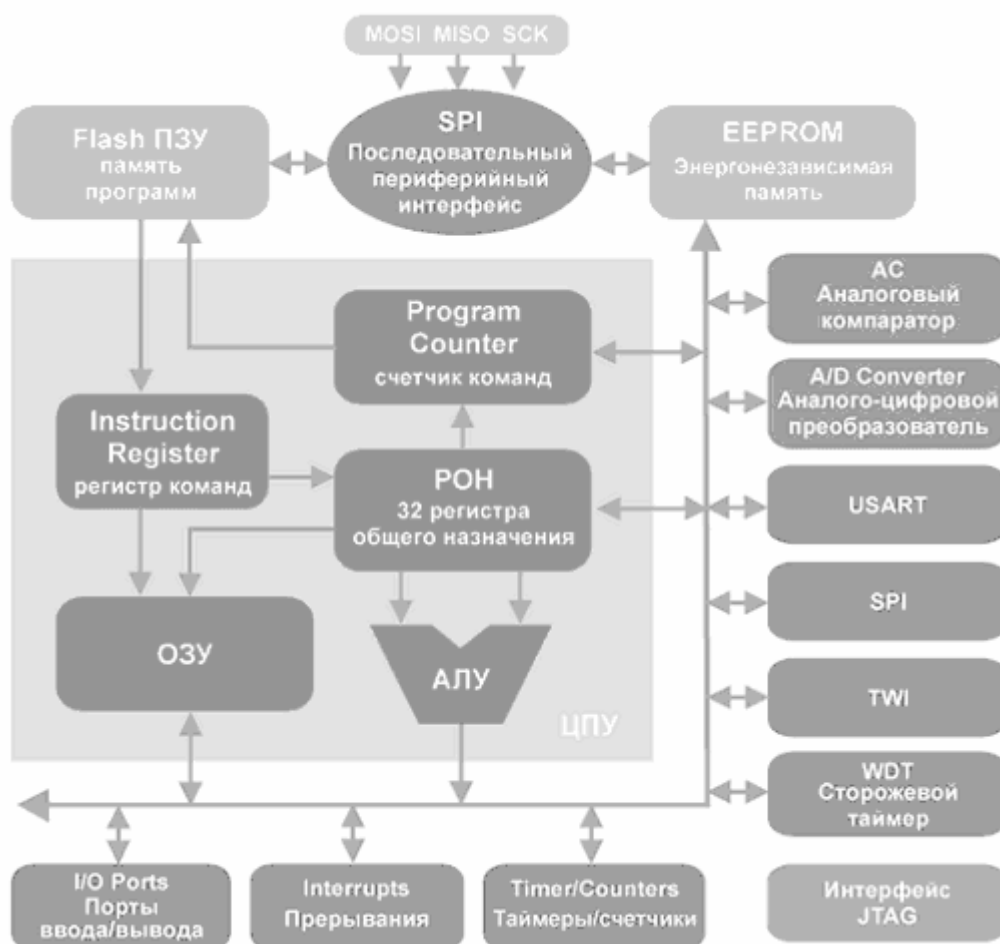


Рисунок 1.1 – Структурна схема мікроконтролерів AVR

**Мікроконтрoлер** (англ. *microcontroller*), або однокристална мікроЕОМ — виконана у вигляді мікросхеми спеціалізована мікропроцесорна система, що містить власне мікропроцесор, блоки пам'яті для збереження коду програм і даних, порти введення-виведення і блоки зі спеціальними функціями (лічильники, компаратори, АЦП тощо).

Застосовують для керування електронними пристроями. Загалом, це — однокристальний комп'ютер, здатний виконувати прості завдання. Використання однієї мікросхеми значно знижує розміри, енергоспоживання і вартість пристроїв, побудованих на базі мікроконтролерів.

**Компанія ATMEL Corp** . - один зі світових лідерів у виробництві широкого спектру мікросхем енергонезалежній пам'яті, FLASH-мікроконтролерів і мікросхем програмованої логіки, взяла старт з розробки RISC-мікроконтролерів в середині 90-х років, використовуючи всі свої технічні рішення, накопичені до цього часу.

Концепція нових швидкісних мікроконтролерів була створена групою розробників дослідного центру ATMEL у Норвегії, ініціали яких потім сформували марку AVR. Перші мікроконтролери AVR AT90S1200 з'явилися в середині 1997 р. і швидко здобули визнання споживачів. Ці 8-розрядні RISC-мікроконтролери для вбудованих додатків є найбільш затребуваними та прогресивними у галузіх, що стрімко розвиваються .

Термін **RISC** (*Reduced Instruction Set Computer* - обчислювач з скороченим набором команд) означає, що процесорні ядро оперує з мінімізованим набором машинних команд, і, отже, кількість різних машинних циклів невелика. Це дозволяє в значній мірі скоротити час виконання машинного циклу, і команди відповідно. Таким чином, відношення тривалості машинного циклу до тривалості такту зменшується - від 12 у класичних контролерів сімейства MCS-51 до 1-4 у контролерів сімейства AVR. Таким чином, при однаковому значенні тактової частоти продуктивність зростає в кілька разів.

AVR-архітектура, на основі якої побудовані мікроконтролери сімейства AVR, поєднує потужний RISC-процесор за Гарвардською архітектурою з роздільним доступом до пам'яті програм і даних, 32 регістри загального призначення, кожен з яких може працювати як регістр-акумулятор, і розвинену систему команд фіксованої 16-бітової довжини. Більшість команд виконують за один машинний такт з одночасним виконанням поточної і вибіркою наступної команди, що забезпечує продуктивність до 1 MIPS на кожний мегагерц тактової частоти.

Регістри загального призначення (РЗП) утворюють регістровий набір швидкого доступу, де кожен з 32 регістрів безпосередньо пов'язаний з арифметично-логічним пристроєм (АЛП). За один такт з регістрового набору вибирають два операндів, виконують операцію, і результат повертають в регістровий набір. АЛП підтримує арифметичні та логічні операції з регістрами, між регістром і константою або безпосередньо з регістром. Регістровий файл також доступний як частина пам'яті даних. Шість з 32 регістрів можна використовувати як три 16-розрядних регістри-показники для непрямої адресації. Старші мікроконтролери сімейства AVR мають у складі АЛП апаратний помножувач.

Базовий набір команд AVR містить 120 інструкцій. Інструкції бітових операцій включають інструкції установки, очищення та тестування бітів. Однак у порівнянні з контролерами сімейства MCS-51 мнемоніка команд залежить від типу адресації (навіть команди пересилання), багато команд переходу мають нестандартний синтаксис (без позначок, з пропуском фіксованого числа команд).

Всі мікроконтролери AVR мають вбудовану Flash-ROM з можливістю внутрішнього програмування через послідовний 4-провідний інтерфейс типу SPI.

Периферія МК AVR містить таймери-лічильники, широтно-імпульсні модулятори, підтримку зовнішніх переривань, аналогові компаратори, 10-розрядний багатоканальний АЦП, паралельні порти (від 3 до 53 ліній введення та виведення), інтерфейси UART (USART), SPI, TWI, вбудовані генератори з внутрішнім RC-ланцюжком, сторожовий таймер і пристрій скидання за ввімкнення живлення. Усі ці якості перетворюють AVR-мікроконтролери в потужний інструмент для побудови сучасних, високопродуктивних і економічних контролерів різного призначення.

В рамках єдиної базової архітектури AVR-мікроконтролери підрозділяють на три підродини: **Tiny AVR**, **Classic AVR**, **Mega AVR**.

Мікроконтролери сімейства **Tiny** мають невеликі обсяги пам'яті (1 .. 2 Кбайт) і досить обмежену периферію. Практично всіх їх випускають в 8-вивідних корпусах і вони призначені для рішень, які приймаються в умовах жорстких фінансових обмежень. Область застосування цих мікроконтролерів - інтелектуальні датчики різного призначення (контрольні, пожежні, охоронні), іграшки, зарядні пристрої, різна побутова техніка.

Мікроконтролери сімейства **Mega** мають найбільш розвинену периферію, найбільші серед всіх контролерів AVR обсяги пам'яті програм і даних. Вони призначені для використання в телекомунікаційній апаратурі, контролерах різного периферійного обладнання (принтери, сканери, сучасні дискові накопичувачі, приводи CD-ROM/DVD-ROM і т.п.), складної офісної техніки та стійках управління технологічним обладнанням (станки з ЧПУ).

Всі AVR-мікроконтролери сумісні по кодах команд, підтримують кілька режимів зниженого енергоспоживання, мають блок переривань, сторожовий таймер і допускають програмування безпосередньо в готовому пристрої (внутрішньосистемне програмування - ISP).

Сімейство забезпечено комплектом програм і системами налагодження, що охоплюють макроасемблер, відладчик/симулятори програм, внутрішньосхемні емулятори. Популярним програмним продуктом для розробки є система AVR Studio, що вільно розповсюджується корпорацією Atmel.



**Основні електричні характеристики новітніх мікроконтролерів AVR фірми Atmel** повністю статична схемотехніка - МК працюють за тактової частоти до 20 МГц; діапазон напруг живлення від 2,7 В до 6,0 В; різні режими енергозбереження: пасивний (idle) і стоповий (power down).

### **Робоче завдання**

Вивчити характеристики та можливості мікроконтролерів фірми Atmel. Провести порівняльний аналіз підродин МК

### **Контрольні питання**

1. Які особливості мікроконтролерів ви знаєте?
2. Які основні особливості мікроконтролерів фірми Atmel?
3. У чому відмінність підродини Tiny від Mega?
4. Яку модель розподілу пам'яті реалізовано у мікроконтролерах фірми Atmel?
5. Які інші архітектури мікроконтролерів Ви знаєте?
6. Що таке тактова частота, як вона визначає швидкодію МК?
7. Що таке РЗП та яке його призначення?
8. Чим відрізняються регістри загально призначення від спеціальних регістрів?

## Лабораторна робота №2

# Ознайомлення з особливостями мікроконтролера ATmega16. Процедури опитування стану вхідного та керування вихідним портами

**Мета роботи** - ознайомитися з мікроконтролерами AVR на прикладі мікроконтролера ATmega16, навчитися виводити інформацію через світлодіодні індикатори та зчитувати стан кнопок через вхідні та вихідні порти

### Домашнє завдання

Вивчити характеристики та можливості мікроконтролера ATmega16. З'ясувати особливості портів введення-виведення. Проаналізувати електричні характеристики та схеми підключення світлодіодів та кнопок до МК.

### Теоретичні відомості

Визначимо особливості роботи з портами мікроконтролера на прикладі роботи з світлодіодними індикаторами та кнопками.

Для цього необхідно налаштувати макет так, як наведено на рис. 2.1.

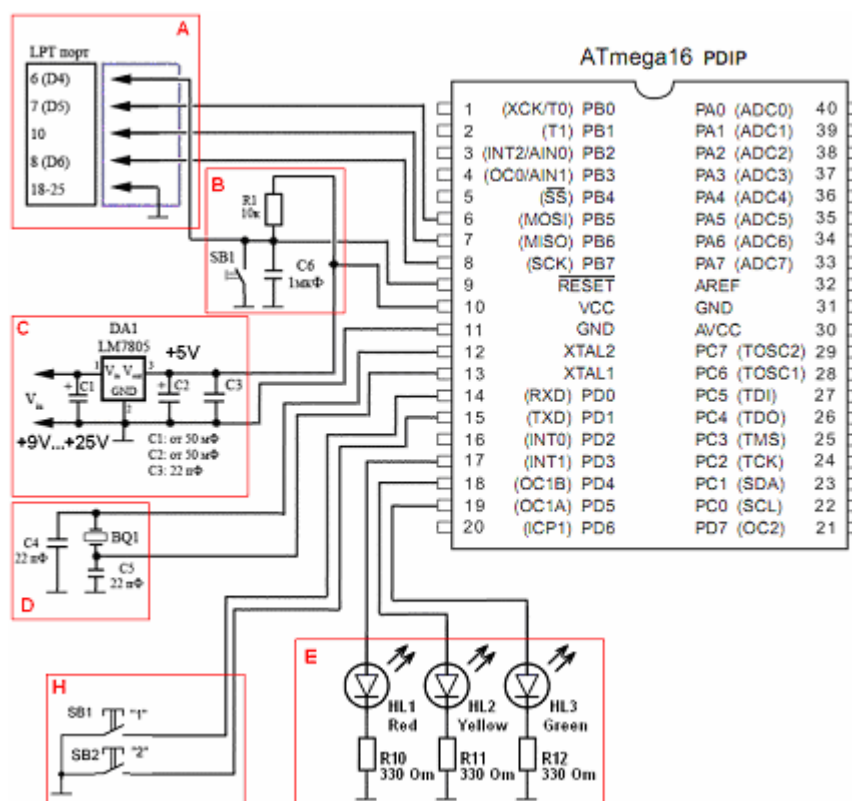


Рисунок 2.1 – Схема макета

Блок А – рознім для підключення програматора.

Блок В – типова схема підключення кнопки скидання „reset”.

Блок С – типова схема вхідного стабілізатору напруги.

Блок D –схема підключення зовнішнього тактового генератору.

Блок E –блок з трьох світлодіодів.

Блок H –блок кнопок.

Для того, щоб подати сигнал на світлодіод, або прочитати стан кнопки, їх треба під'єднати до одного з портів МК. Для того, щоб засвітився світлодіод HL1, треба подати високу напругу (рівень логічної „1”) на вивід номер 17 (це порт D). Перед тим, як працювати з портом, треба перевести його в один з станів, для роботи з світл діодами треба перевести порт у стан виводу інформації, а для зчитування стану кнопок треба перевести порт у стан введення інформації. Мікроконтролери AVR дозволяють налаштовувати режими для кожного виводу окремо. Для вказування режимів роботи виводів використовують спеціальні регістри.

Визначимо особливості налаштування порту A:

Регістр **DDRA** задає режим роботи, одиниці вказують на те, що відповідні виводи порту працюють у режимі виводу, а нулі у режимі введення інформації.

Наприклад DDRA=0x0F означає що молодші чотири біти порту A налаштовано у режим виводу інформації, а старші чотири у режим введення інформації. Крім того, можна задати стан сигналів на виводах порту, це робиться за допомогою налаштування регістрі **PORTA**. Переведемо у стан логічної одиниці біти 0, 1, 4, 5 а інші у стан логічного нуля: PORTA=0x33. На рис. 2.2 показано, як задати такий режим роботи порту A у програмі CodeVision.

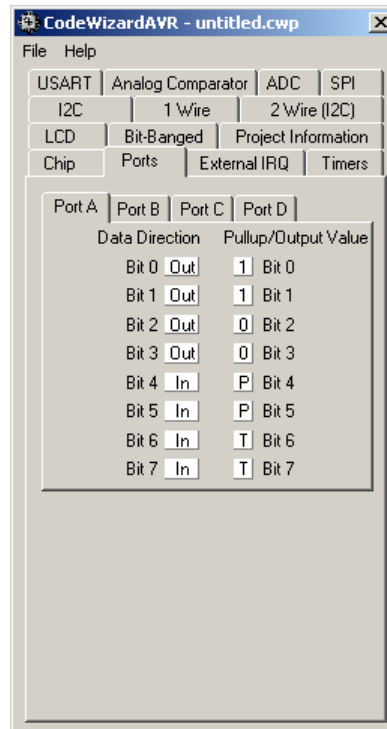


Рисунок 2.2 – Налаштування виводів порту A

Зчитати стан виводів порту можна через регістр **PINA**.

Для інших портів слід замінити останню літеру у назвах регістрів.

Вважаємо, що до виводу PA2 підключено кнопку, а до виводу PA4 - світлодіод за стандартними схемами, аналогічно наведеним на рис. 2.1.

Програма, представлена нижче, аналізує стан кнопки, і якщо кнопка натиснута, то вмикає (засвічує) світлодіод.

```
#include <mega16.h>

void main(void)
{
    PORTA=0x04; //включити підтягуючий резистор для виводу PA2
    DDRA=0x10;  //налаштування стану виводів порту, PA4 на виведення, усі інші на введення

    while (1)
    {
        if( PINA.2 == 0 ) PORTA.4 = 1;
    };
}
```

### **Робоче завдання**

1. Ознайомитися з характеристиками та можливостями мікроконтролера ATMega16.
2. Навчитися налаштовувати стан виводів портів.
3. Навчитися аналізувати стан кнопок підключених до мікроконтролера.
4. Засвітити всі світлодіоди.
5. Погасити світлодіод HL1.
6. Засвітити комбінацію світлодіодів, зазначену викладачем.
7. Розробити програму, яка аналізує стан кнопок і реалізує такий алгоритм: світлодіод HL3 горить завжди, якщо натиснута кнопка SB1, то горить світлодіод HL1, якщо натиснута кнопка SB2, то горить світлодіод HL2.
8. Розробити програму, яка аналізує стан кнопок і реалізує такий алгоритм: світлодіоди відображають стан трибітного лічильника у двійковій формі (світлодіод HL1 відображає молодший розряд тощо.), натиснення кнопки SB1 збільшує вміст лічильника на одиницю, а відповідно кнопки SB2 зменшує.

### **Контрольні питання**

1. Що таке порт?
2. Які регістри використовуються для роботи з портами у МК AVR?
3. Для чого послідовно з світлодіодом ставлять резистор?
4. Як розрахувати параметри резистора, який ставлять послідовно з світлодіодом?
5. Наведіть типові схеми підключення кнопок до МК.
6. Поясніть особливості підключення кнопок за рис. 2.1
7. Чи можна підключити світлодіод не до загального провідника, а до живлення?

Рисунок 3.1 – Схема підключення семисегментного індикатора до МК

Блок А – рознім для підключення програматора.

Блок В – типова схема підключення кнопки скидання „reset”.

Блок С – типова схема вхідного стабілізатору напруги.

Блок D – схема підключення зовнішнього тактового генератору.

Блок G – блок підключення семисегментного індикатора.

Семисегментні індикатори являють собою збірки світлодіодів. Існують два варіанти збірок. Схема с *загальним анодом*, та схема с *загальним катодом*. На рис. 3.1 наведено, що у макеті використано схему с загальним катодом. Це означає, що світлодіоди сегментів цифр з’єднано катодами. Для того, щоб отримати зображення на такому індикаторі, то треба на загальний катод цифри подати рівень напруги логічного нуля, а на ті сегменти, які беруть участь у формуванні цифри подати високий рівень напруги (логічної „1”). Наприклад, для формування цифри “2” на першому сегменті індикатора треба подати логічний нуль на вивід індикатора 12 (у МК це вивід 37 – PA3), а на виводи A, B, G, E, D (це виводи з номерами 11, 7, 5, 1, 2 а на МК відповідно 28 – PC6, 22 – PC0, 23 – PC1, 27 – PC5, 26 – PC4) подати логічну одиницю. Схему одного сегмента представлено на рис. 3.2.

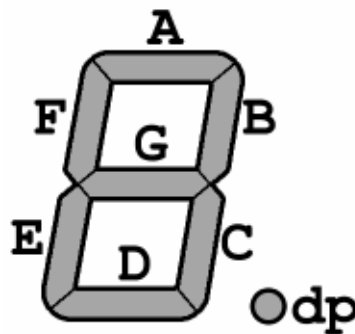


Рисунок 3.2 – Схема позначення сегментів індикатора

**УВАГА!** Схема підключення індикатора розрахована на роботу у режимі динамічної індикації, у режимі статичної індикації струм на катодах може перевищувати граничне допустиме значення для мікроконтролерів AVR, що призведе до виходу за ладу МК. Перед подачею живлення на макет слід отримати дозвіл викладача.

В попередній роботі було розглянуто алгоритм засвічування світлодіода, використовуючи набуті навички виведемо цифру „1” на перший індикатор. Для цього треба подати „0” на вивід PA3 а на вивід PC0 та PC2 подати „1”, цей процес ілюструє рис 3.3.

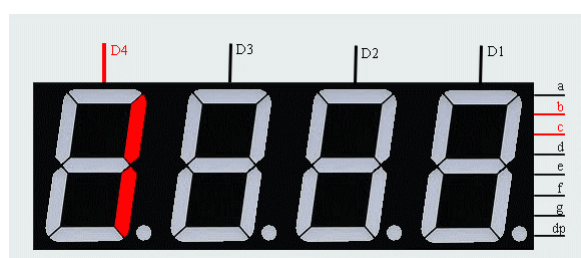


Рисунок 3.2 – Процес активації першої цифри індикатора

Програма виглядає таким чином.

```
#include <mega16.h>

void main(void)
{
    PORTA=0xF7;
    DDRA=0xFF;

    PORTC=0x05;
    DDRC=0xFF;

    while (1)
    {
        };
    }
}
```

### **Робоче завдання**

1. Повідомити викладачу величину граничного струму, який може проходити через виводи МК.
2. Навчитися розраховувати струми, які проходять через вивід МК для ситуацій з одним світлодіодом та з семисегментним індикатором.
3. Отримати у викладача комбінацію, яку треба засвітити на семисегментному індикаторі.
4. Розрахувати струмі, які будуть проходити через виводи МК, та переконатися, що вони менші, ніж гранично допустимі.
5. **УВАГА!** Отримати у викладача дозвіл на роботу з семисегментним індикатором у статичному режимі.
6. Засвітити комбінацію на семисегментному індикаторі.
7. Запропонувати схему подолання обмеження на кількість підключених одночасно сегментів.

### **Контрольні питання**

1. Який гранично допустимий струм для виводів МК AVR?
2. Як розрахувати величину струму, який протікає для певної комбінації сегментів?
3. Навіщо послідовно з світлодіодом ставлять резистор?
4. Як розрахувати параметри резистора, який ставлять послідовно з світлодіодом?

# Лабораторна робота №4

## Динамічна індикація

**Мета роботи** - Навчитися виводити інформацію на семисегментний світлодіодний індикатор у режимі динамічної індикації. Опанувати динамічну індикацію

### Домашнє завдання

Вивчити особливості та принцип динамічної індикації, повторити систему команд щодо керування портами введення-виведення

### Теоретичні відомості

Визначимо особливості роботи мікроконтролера на прикладі роботи з семисегментними індикаторами у режимі динамічної індикації. Для цього зберемо макет, як наведено на рис. 4.1. (подібний рисунку рис 3.1 але з додаванням кнопок)

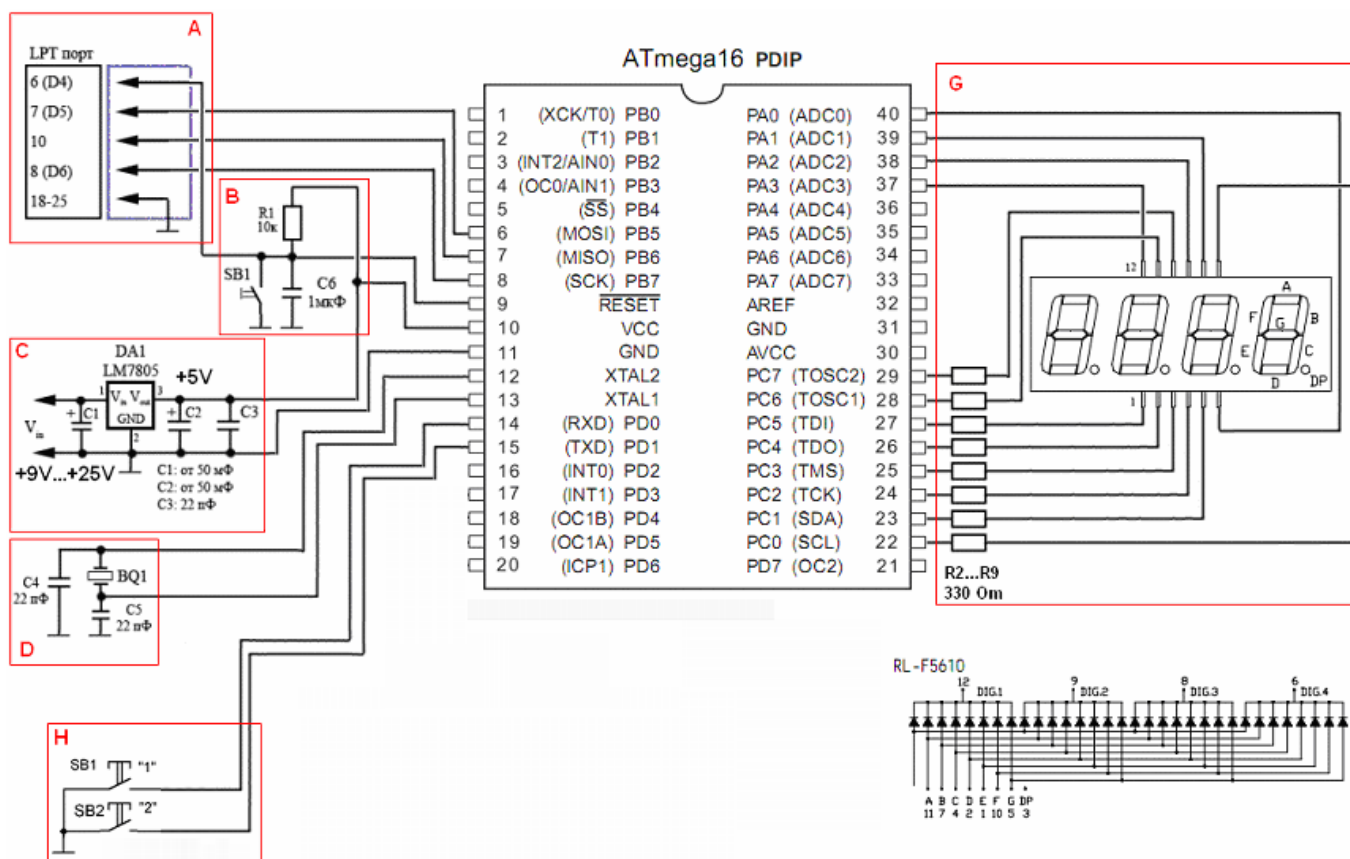


Рисунок 4.1 – Схема підключення семисегментного індикатора до МК

Блок А – рознім для підключення програматора.

Блок В – типова схема підключення кнопки скидання „reset”.

Блок С – типова схема вхідного стабілізатору напруги.

Блок D – схема підключення зовнішнього тактового генератора.

Блок Н – блок кнопок.

Блок G – блок підключення семисегментного індикатора.



Динамічна індикація полягає у тому, що цифри на індикаторі формують по черзі, спочатку формують першу цифру на анодах і подають низький рівень на перший загальний катод, потім формують другу цифру на анодах і подають низький рівень на другий загальний катод тощо (див. рис. 4.2)

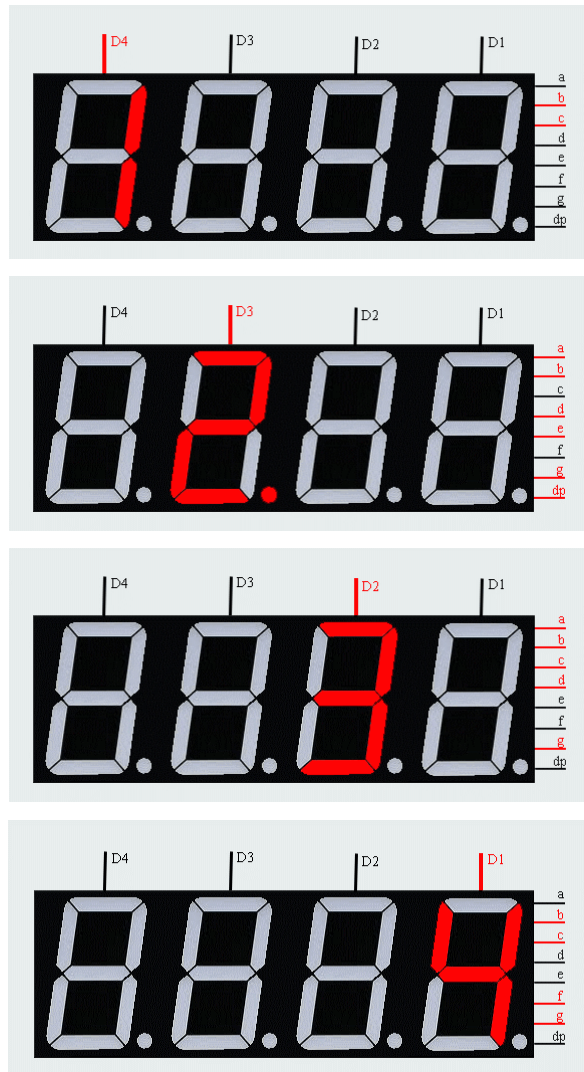


Рисунок 4.2 – Послідовна активація цифр

Програма для засвічування на індикаторі фіксованої комбінації виглядає таким чином.

```
#include <mega16.h>
#include <delay.h>

void main(void)
{
    while (1)
    {
        PORTA=0xF7; // перший катод
        DDRA=0xFF;

        PORTC=0x05; // цифра 1
        DDRC=0xFF;

        delay_ms( 10 );

        PORTA=0xFB; // другий катод
        DDRA=0xFF;

        PORTC=0x73; // цифра 2
```

```

DDRC=0xFF;

delay_ms( 10 );

PORTA=0xFD; // третий катод
DDRA=0xFF;

PORTC=0x57; // цифра 3
DDRC=0xFF;

delay_ms( 10 );

PORTA=0xFE; // четвертый катод
DDRA=0xFF;

PORTC=0x87; // цифра 4
DDRC=0xFF;

delay_ms( 10 );

};
}

```

Перед роботою з МК можна побудувати аналогічну схему (див. рис. 4.3) і перевірити її працездатність у пакеті програм Proteus.

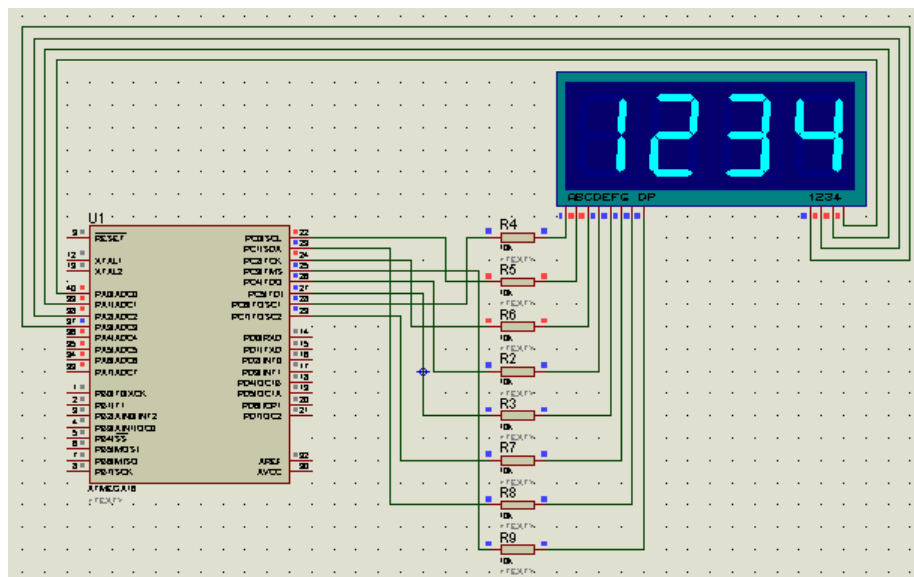


Рисунок 4.3 – Схема підключення семи сегментного індикатора побудована у програмі Proteus

### Робоче завдання

1. Отримати у викладача комбінацію, яку треба вивігити на семисегментному індикаторі.
2. Розрахувати мінімальну частоту регенерації для повністю засвіченого індикатору.
3. Узгодити значення струмів через виводи МК.
4. **УВАГА!** Отримати у викладача дозвіл на роботу з семисегментним індикатором у динамічному режимі.
5. Розробити програму для висвітлення комбінації на семисегментному індикаторі.
6. Модифікувати програму таким чином, щоб на індикатор можна було виводити будь яку комбінацію знаків.
7. Модифікувати програму таким чином, щоб при натисканні на кнопку SB1 число на індикаторі збільшувалося, а при натисканні на кнопку SB2 - зменшувалося.

### **Контрольні питання**

1. У чому полягає принцип динамічної індикації?
2. Чому катоди підключають через транзистор?
3. Чому можна підключати катоди напряму до виводів МК?
4. Як вибрати частоту, з якою треба перемикати розряди індикатора?

# Лабораторна робота №5

## Обмін даними з комп'ютером через послідовний інтерфейс RS232

**Мета роботи** - Організувати взаємодію мікроконтролера з ЕОМ за допомогою стандартного інтерфейсу RS232. Опанувати виведення інформації через RS232. Здобути навички програмування МК для виведення на індикатор, та, одночасно, через RS232 значення лічильника, який працює у нескінченному циклі

### Домашнє завдання

Опанувати роботу з послідовним портом, розібратися з принципами організації обміну даними з ЕОМ через послідовний інтерфейс RS 232.

### Теоретичні відомості

Зберемо макет як показано на Рис. 5.1.

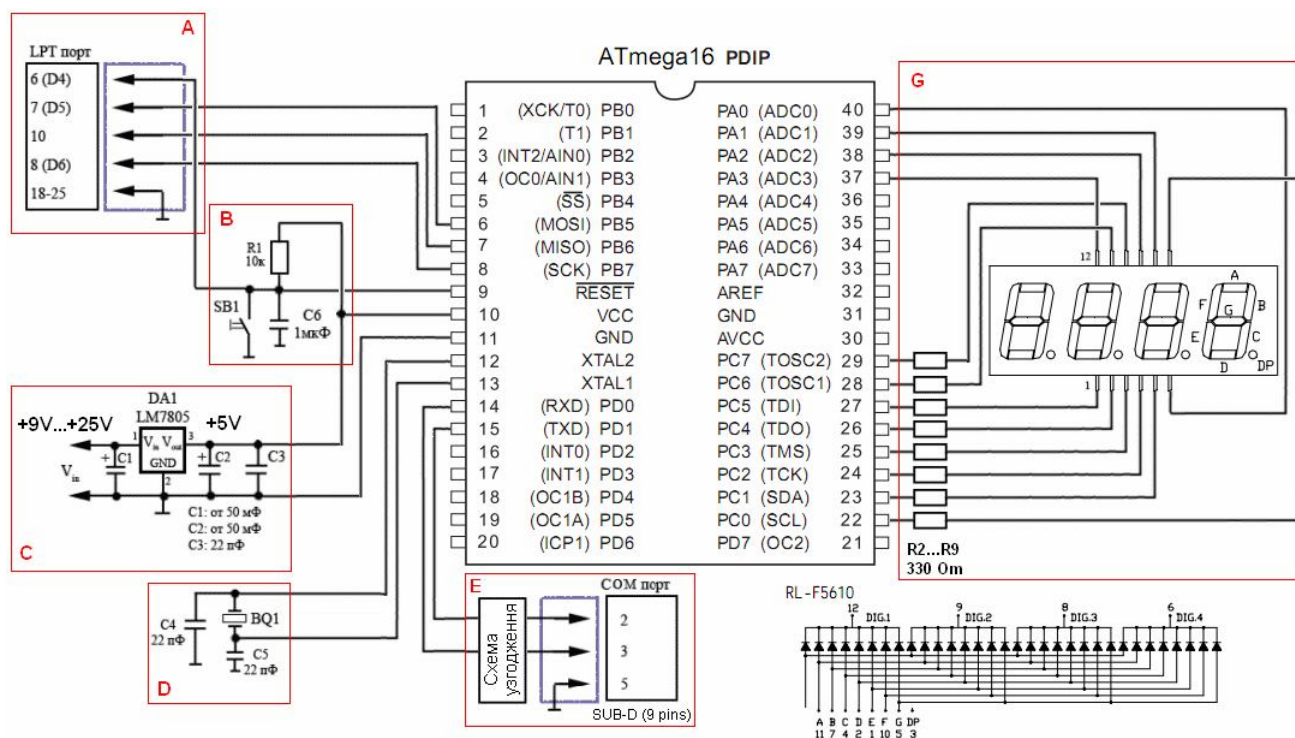


Рисунок 5.1 – Схема підключення семисегментного індикатора та підключення до порту RS232 ЕОМ.

Блок А – рознім для підключення програматора.

Блок В – типова схема підключення кнопки скидання „reset”.

Блок С – типова схема вхідного стабілізатору напруги.

Блок D – схема підключення зовнішнього тактового генератору.

Блок E – блок підключення до послідовного порту.

Блок G – блок підключення семисегментного індикатора.

При роботі з послідовним портом з використанням бібліотечних функцій, які надає компілятор CodeVision спочатку треба налаштувати порт МК на потрібну швидкість передачі, та вибрати режим у якому буде відбуватися передача даних. Частіше всього використовують швидкість передачі 9600, та асинхронний режим передачі з такими параметрами: 8 інформаційних біт, 1 стоп біт, без контролю парності. ілюструє рис. 5.2.

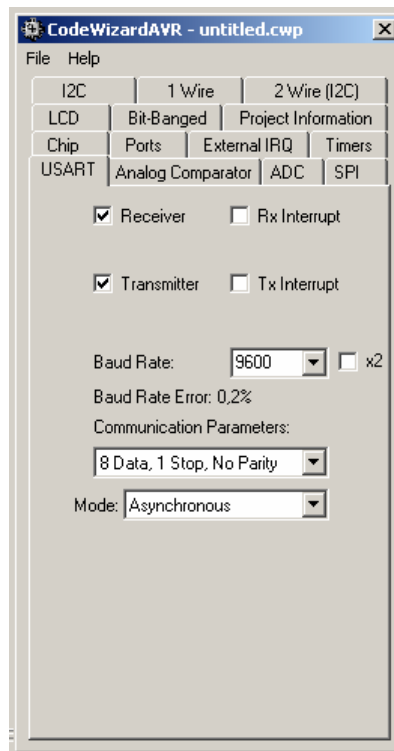


Рисунок 5.2 – налаштування режиму USART.

За налаштування відповідають регістри UCSRA, UCSRB, UCSRC, UBRRH, UBRRL.

Приклад ініціалізації регістрів наведено нижче.

```
// USART initialization
// Communication Parameters: 8 Data, 1 Stop, No Parity
// USART Receiver: On
// USART Transmitter: On
// USART Mode: Asynchronous
// USART Baud Rate: 9600
UCSRA=0x00;
UCSRB=0x18;
UCSRC=0x86;
UBRRH=0x00;
UBRRL=0x33;
```

Для виводу даних у порт використовують стандартну функцію мови C, printf.

Нижче представлено лістинг програми, яка виводить у порт значення 0 і 1.

```
#include <mega16.h>

// Standard Input/Output functions
#include <stdio.h>

// Declare your global variables here

int A = 0;

void main(void)
{
```

```

// USART initialization
// Communication Parameters: 8 Data, 1 Stop, No Parity
// USART Receiver: On
// USART Transmitter: On
// USART Mode: Asynchronous
// USART Baud Rate: 9600
UCSRA=0x00;
UCSRB=0x18;
UCSRC=0x86;
UBRRH=0x00;
UBRRL=0x33;

printf( "A = %d\r", A );

A++;

printf( "A = %d\r", A );

while (1)
{
    // Place your code here
};
}

```

### **Робоче завдання**

1. Отримати у викладача комбінацію яку треба засвітити на семисегментному індикаторі.
2. Засвітити комбінацію на семисегментному індикаторі.
3. Вивести вказану комбінацію у порт та продемонструвати викладачу роботу програми на екрані комп'ютера.

### **Контрольні питання**

1. Що таке інтерфейс RS232?
2. Як організувати взаємодію мікроконтролера з іншим пристроєм?
3. Як розрахувати і заповнити регістри конфігурації для організації обміну через інтерфейс RS232?
4. Як перевірити функціонування інтерфейсу RS232?
5. Для чого використовують мікросхему MAX232 або її аналоги?

# Лабораторна робота №6

## Тема: „Система переривання”

**Мета роботи** - Опанувати роботу з перериваннями, навчитися використовувати таймери для вимірювання часових інтервалів. Скласти програму, яка вимірює час між моментами натискання на кнопку

### Домашнє завдання

Вивчити особливості апаратних та програмних переривань. Вміти розробляти фрагменти програми обробки переривань.

### Теоретичні відомості

Поєднання налаштування програми залежить від схемної організації АПК. Тому для цього треба налаштувати макет, схему якого наведено на рис. 6.1.

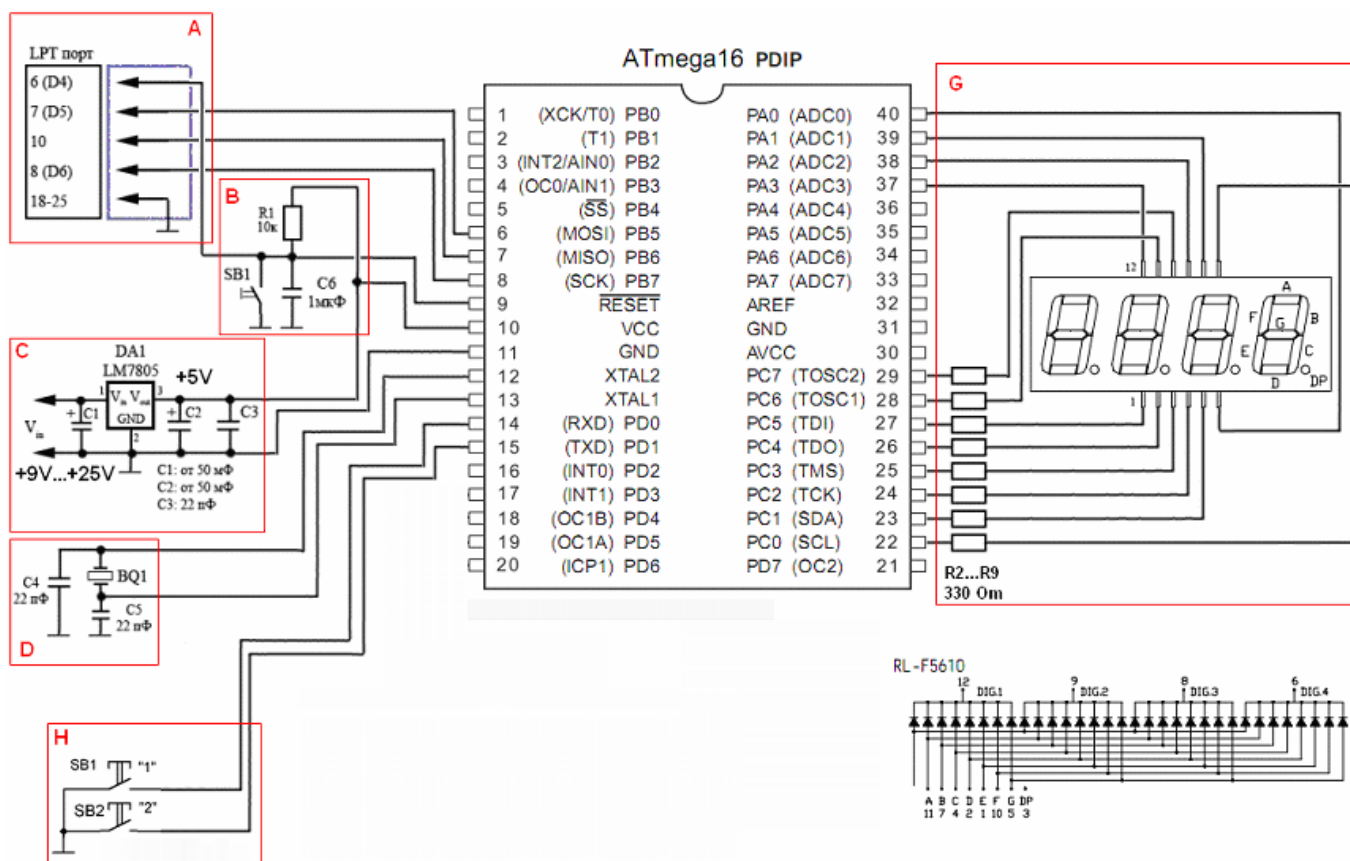


Рисунок 6.1 – Схема макета для вивчення системи переривань

Блок А – рознім для підключення програматора.

Блок В – типова схема підключення кнопки скидання „reset”.

Блок С – типова схема вхідного стабілізатору напруги.

Блок D – схема підключення зовнішнього тактового генератора.

Блок Н – блок кнопок.

Блок G – блок підключення семисегментного індикатора.

Для того, щоб виміряти часовий інтервал за допомогою використання системи переривань, треба зафіксувати момент першого натискання кнопки і потім рахувати кожне спрацювання переривання до моменту другого натискання кнопки. Помноживши кількість переривань на час між перериваннями, можна визначити тривалість часового інтервалу.

Для того, щоб організувати інтервал між перериваннями з частотою приблизно 1 кГц, треба провести такі налаштування - у вікні CodeWizardAVR відкрити закладку Timers і встановити відповідне значення параметра Clock Value (див. рис.6.2).

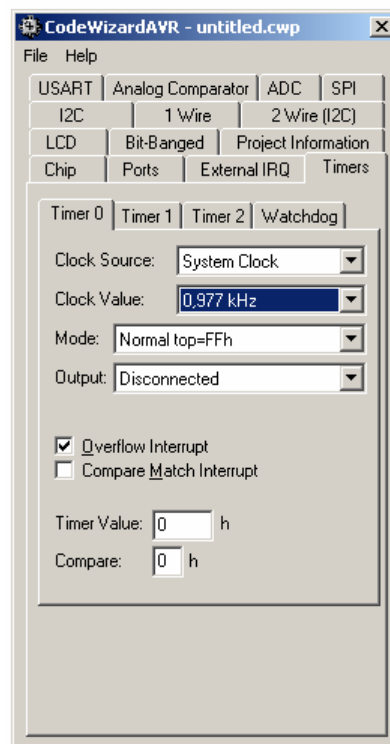


Рисунок 6.2 - Налаштування режиму переривань

Нижче представлено програму, яка забезпечує з частотою 1 Hz миготіння молодшого розряду індикатора. Необхідно звернути увагу на стан регістрів, які відповідають за встановлення режиму роботи таймерів та встановлення коефіцієнту поділу тактового генератора.

```
#include <mega16.h>

int Cnt = 1000;

// Timer 0 overflow interrupt service routine
interrupt [TIM0_OVF] void timer0_ovf_isr(void)
{
    if( Cnt > 0 ) { Cnt--; return; }

    Cnt = 1000;
    PORTC = ~PORTC;
}

// Declare your global variables here

void main(void)
{
    PORTA=0xfe;
    DDRA=0xff;

    PORTC=0x00;
    DDRC=0xff;
}
```



```

// Timer/Counter 0 initialization
// Clock source: System Clock
// Clock value: 0,977 kHz
// Mode: Normal top=FFh
// OC0 output: Disconnected
TCCR0=0x05;
TCNT0=0x00;
OCR0=0x00;

// Timer/Counter 1 initialization
// Clock source: System Clock
// Clock value: Timer 1 Stopped
// Mode: Normal top=FFFFh
// OC1A output: Discon.
// OC1B output: Discon.
// Noise Canceler: Off
// Input Capture on Falling Edge
// Timer 1 Overflow Interrupt: Off
// Input Capture Interrupt: Off
// Compare A Match Interrupt: Off
// Compare B Match Interrupt: Off
TCCR1A=0x00;
TCCR1B=0x00;
TCNT1H=0x00;
TCNT1L=0x00;
ICR1H=0x00;
ICR1L=0x00;
OCR1AH=0x00;
OCR1AL=0x00;
OCR1BH=0x00;
OCR1BL=0x00;

// Timer/Counter 2 initialization
// Clock source: System Clock
// Clock value: Timer 2 Stopped
// Mode: Normal top=FFh
// OC2 output: Disconnected
ASSR=0x00;
TCCR2=0x00;
TCNT2=0x00;
OCR2=0x00;

// External Interrupt(s) initialization
// INT0: Off
// INT1: Off
// INT2: Off
MCUCR=0x00;
MCUCSR=0x00;

// Timer(s)/Counter(s) Interrupt(s) initialization
TIMSK=0x01;

// Global enable interrupts
#asm("sei")

while ( 1 == 1 )
{
    // Place your code here
}
}

```

### **Робоче завдання**

1. Отримати у викладача значення частоти, з якою треба забезпечити миготіння, та засвічення розряду індикатора, розробити програму.
2. Провести тестування та налагодження програми, довести її працеспроможність у разі завантаження у макетний зразок

### **Контрольні питання**

1. Які особливості системи переривання?
2. Які особливості таймерів та регістрів керування ними?
3. Як організувати спрацювання таймеру з частотою, відмінною від стандартної?

# Лабораторна робота №7

## Тема: „Виведення інформації на LCD дисплей”

**Мета роботи** - Організувати виведення інформації з МК на LCD дисплей. Розробити програму, яка реалізує лічильник з можливістю його нарощування та зменшення за допомогою кнопок, вивести значення лічильника у десятковій та шістнадцятковій системах на LCD дисплей

### Домашнє завдання

Опанувати роботу з LCD дисплеєм, для організації роботи кнопок збільшення та зменшення лічильника розібратися з принципами програмного усунення ефекту брязкіта контактів.

### Теоретичні відомості

LCD-дисплей потребує налаштування відповідного вихідного порту на роботу з ним. Тому попередньо необхідно підключити його до МК як макет, схему якого наведено на рис. 7.1.

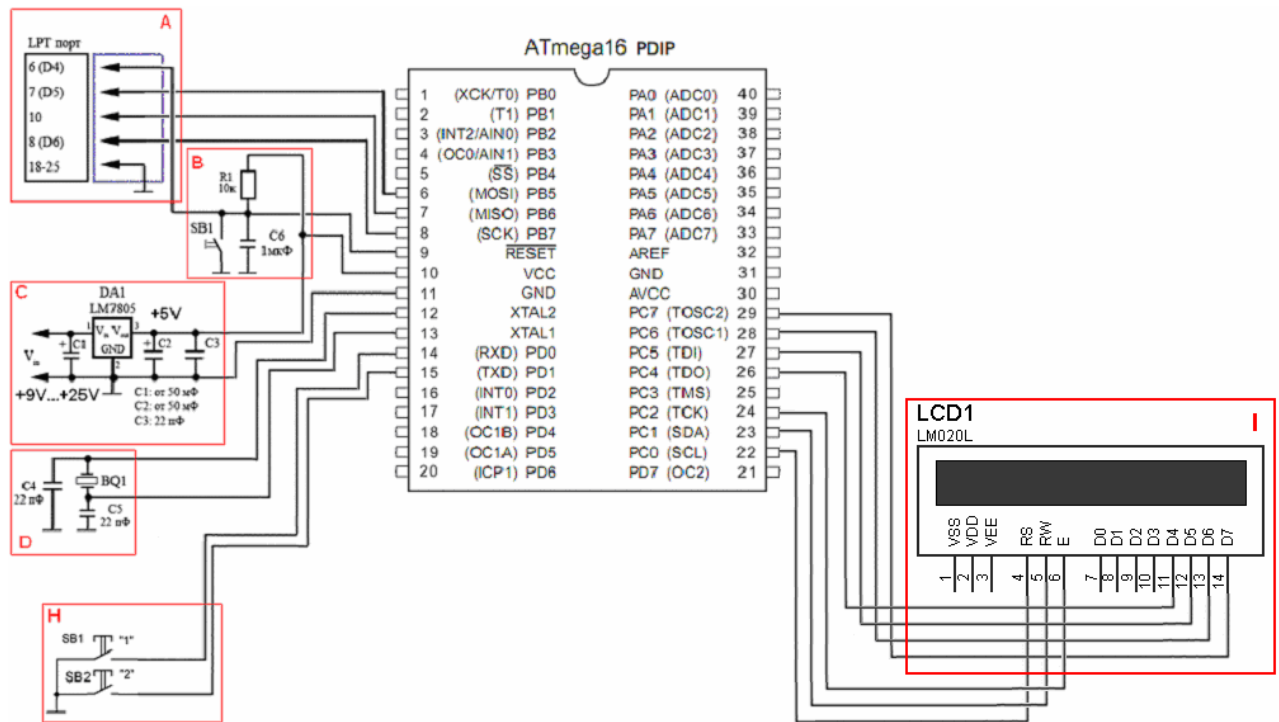


Рисунок 7.1 – Схема підключення LCD дисплея.

Блок А – рознім для підключення програматора.

Блок В – типова схема підключення кнопки скидання „reset”.

Блок С – типова схема вхідного стабілізатору напруги.

Блок D – схема підключення зовнішнього тактового генератору.

Блок Е – блок підключення до послідовного порту.

Блок І – блок підключення LCD дисплея.

При роботі з послідовним портом з використанням бібліотечних функцій, які надає компілятор CodeVision, спочатку треба налаштувати порт МК для роботи з дисплеєм, тому підключимо його до порту C та будемо забезпечувати роботу у режимі пересилання половини байта. Налаштування дуже прості, їх представлено на рис. 7.2.

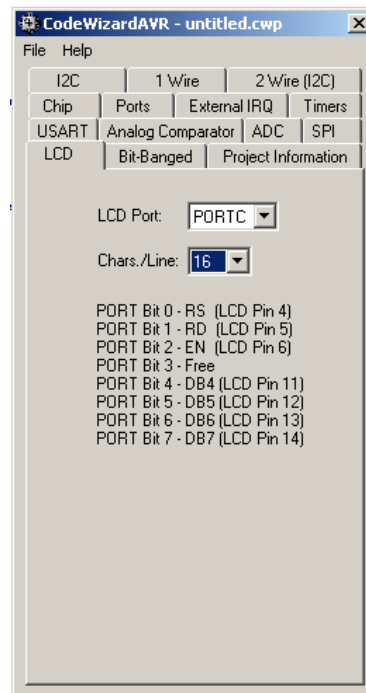


Рисунок 7.2 – Налаштування порту для роботи з LCD дисплеєм.

Для виводу даних на дисплей використовують функції з бібліотеки компілятору CodeVision. Нижче представлено лістинг програми, яка виводить на екран символ \$, а потім слово test.

```
#include <mega16.h>
// Alphanumeric LCD Module functions
#asm
.equ __lcd_port=0x15 ;PORTC
#endasm
#include <lcd.h>

void main(void)
{
// LCD module initialization
lcd_init(16); //ініціалізація дисплея

while (1)
{
    lcd_gotoxy(8,0); // поставимо курсор приблизно по середині екрана
    lcd_putchar('$'); // виведемо один символ
    lcd_puts("test"); // виведемо текст
};
}
```

### **Робоче завдання**

1. Отримати у викладача текст який треба висвітлити на дисплеї, та позицію у якій це треба зробити.
2. Вивести на дисплей текст.
3. Організувати обробку кнопок та реалізувати лічильник, вміст якого кнопкою SB1 можна збільшувати, а кнопкою SB2 зменшувати. Вивести на дисплей покази лічильника у

десятковому та шістнадцятьковому вигляді, продемонструвати викладачу роботу програми.

### **Контрольні питання**

1. Як влаштований і для чого використовують LCD дисплей?
2. Як організувати взаємодію мікроконтролера з LCD дисплеєм?
3. Які існують режими роботи LCD дисплея, чим вони відрізняються, які кращі?
4. Що таке брязкіт контактів, як його усунути програмно?

# Лабораторна робота №8

## Тема: „Робота АЦП”

**Мета роботи** - Опанувати роботу з АЦП, вивести значення напруги на індикатор. Навчитися вимірювати напругу за допомогою АЦП. Скласти програму, яка забезпечує вимірювання напруги на резистивного дільнику

### Домашнє завдання

Теоретично підготуватися до роботи з АЦП, вивчити команди введення аналогових даних через порт, налаштований в режим роботи АЦП, повторити теорію електричних кіл, та основи цифрової схемотехніки.

### Теоретичні відомості

АЦП потребує налаштування відповідного вхідного порту на роботу в цьому режимі. Тому попередньо необхідно підключити його до МК як макет, схему якого наведено на рис. 8.1.

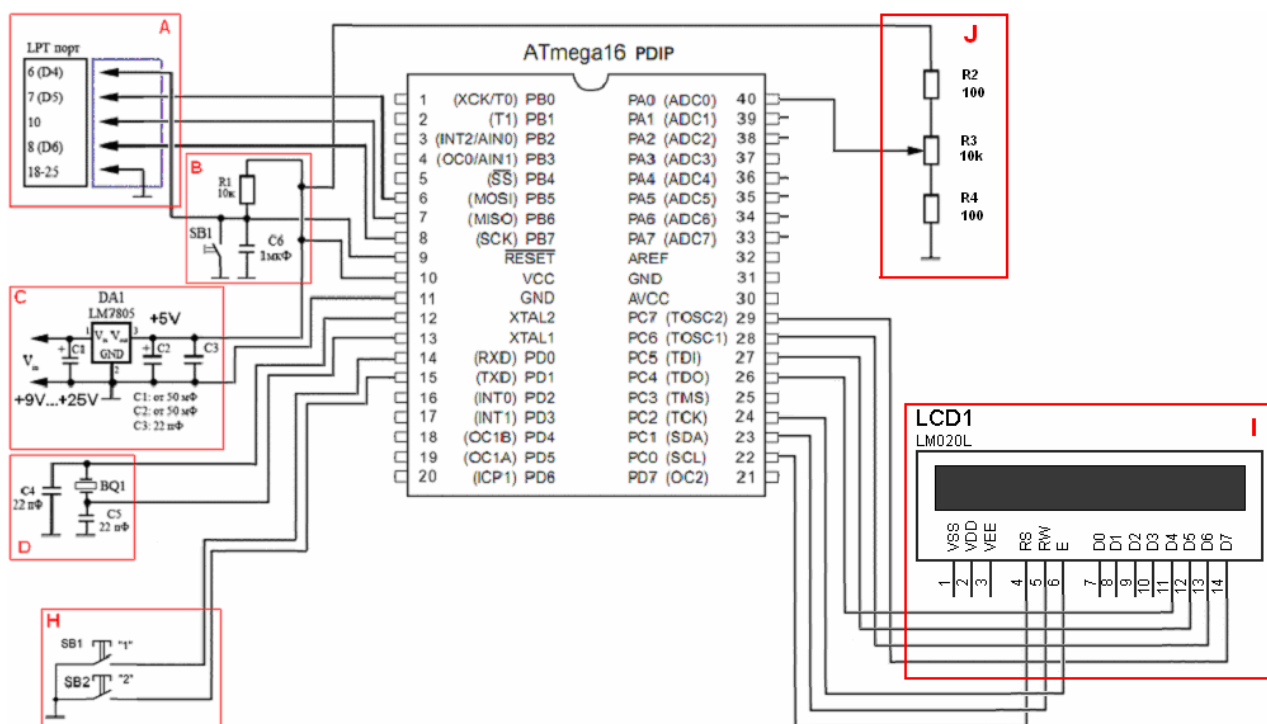


Рисунок 8.1 – Схема для роботи з АЦП.

Блок А – рознім для підключення програматора.

Блок В – типова схема підключення кнопки скидання „reset”.

Блок С – типова схема вхідного стабілізатору напруги.

Блок D – схема підключення зовнішнього тактового генератора.

Блок H – блок кнопок.

Блок I – блок підключення LCD дисплея.

Блок J – блок АЦП.

Для того щоб організувати роботу з АЦП, треба спочатку включити АЦП, рис. 8.2 ілюструє цю процедуру, тобто як це зробити . Показано тільки один за багатьох режимів роботи АЦП.

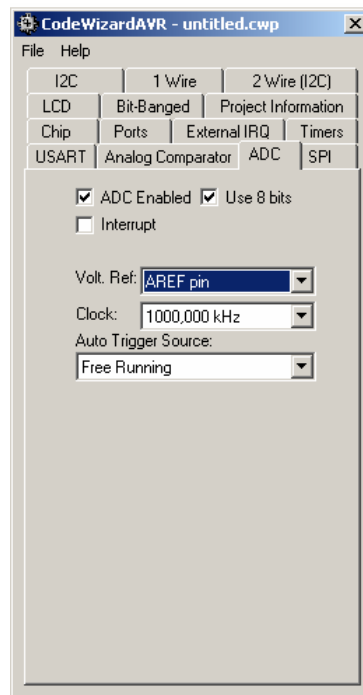


Рисунок 8.2 – Налаштування АЦП.

Нижче представлено програму, яка читає значення АЦП (канал 0).

```
#include <mega16.h>

#include <delay.h>

#define ADC_VREF_TYPE 0x60

// Read the 8 most significant bits
// of the AD conversion result
unsigned char read_adc(unsigned char adc_input)
{
    ADMUX=adc_input | (ADC_VREF_TYPE & 0xff);
    // Delay needed for the stabilization of the ADC input voltage
    delay_us(10);
    // Start the AD conversion
    ADCSRA|=0x40;
    // Wait for the AD conversion to complete
    while ((ADCSRA & 0x10)==0);
    ADCSRA|=0x10;
    return ADCH;
}

void main(void)
{
    unsigned char ADC;

    // ADC initialization
    // ADC Clock frequency: 1000,000 kHz
    // ADC Voltage Reference: AVCC pin
    // ADC Auto Trigger Source: Free Running
    // Only the 8 most significant bits of
    // the AD conversion result are used
    ADMUX=ADC_VREF_TYPE & 0xff;
    ADCSRA=0xA3;
    SFIOR&=0x1F;

    while (1)
    {
        // Place your code here

        ADC = read_adc( 0 ); //читаємо АЦП

        delay_ms( 500 );
    };
}
```

### **Робоче завдання**

1. Написати програму, яка забезпечує зчитування значень АЦП та виведення значення на екран LCD.
2. Відкалібрувати поправочні коефіцієнти для роботи програми за допомогою мультиметра.
3. Виміряти значення напруги, яке попередньо встановив викладач потенціометром R3.
4. Порівняти отримане значення та встановлене викладачем, шляхом калібрування та корекції поправочних коефіцієнтів добитися узгодження показів мультиметра та АПК за програмою.

### **Контрольні питання**

1. Що таке АЦП, який принцип роботи АЦП?
2. Які особливості системи команд налаштування АЦП?
3. Для чого використовують АЦП?
4. Як організувати роботу АЦП у мікро контролерах AVR?
5. Як відкалібрувати АЦП для отримання коректних даних?

## Перелік посилань

1. Высокопроизводительные 8-разрядные RISC микроконтроллеры семейства AVR  
<http://www.gaw.ru/html.cgi/txt/ic/Atmel/micros/avr/start.htm>
2. Высокопроизводительные RISC микроконтроллеры семейства AVR  
<http://www.gaw.ru/html.cgi/txt/doc/micros/avr/about.htm>
3. Примеры применения семейства AVR  
<http://www.gaw.ru/html.cgi/txt/app/micros/avr/index.htm>
4. Программное обеспечение и инструментальные средства для 8-разрядных AVR-микроконтроллеров  
<http://www.gaw.ru/html.cgi/txt/app/Atmel/micros/avr/start.htm>
5. 8-разрядный AVR-микроконтроллер с внутрисистемно программируемой флэш-памятью емкостью 128 кбайт  
<http://www.gaw.ru/html.cgi/txt/doc/micros/avr/arh128/index.htm>
6. AVR. Учебный курс. Архитектура.   
<http://easyelectronics.ru/avr-uchebnyj-kurs-arxitektura.html>
7. Что такое контроллер, и с чем его едят?  
[http://radiokot.ru/start/mcu\\_fpga/avr/01/](http://radiokot.ru/start/mcu_fpga/avr/01/)
8. Управление портами микроконтроллера  
[http://www.myrobot.ru/stepbystep/pr\\_mcports.php](http://www.myrobot.ru/stepbystep/pr_mcports.php)
9. Настройка портов ввода-вывода  
[http://radiokot.ru/start/mcu\\_fpga/avr/06/](http://radiokot.ru/start/mcu_fpga/avr/06/)
10. Делаем устройство "Бегущий огонек" на микроконтроллере  
[http://radiokot.ru/start/mcu\\_fpga/avr/05/](http://radiokot.ru/start/mcu_fpga/avr/05/)
11. Подключение семисегментных индикаторов.
12. <http://www.pcports.ru/articles/avr4.php>
13. Учебный курс. Семисегментный индикатор. Динамическая индикация  
<http://www.chipenable.ru/index.php/programming-c/58-semisegmentnyj-indikator-dinamicheskaja-indikacija.html>
14. Семисегментный индикатор и динамическая индикация на AVR микроконтроллере ATmega8  
<http://www.avrlab.com/node/130>
15. Динамическая индикация  
[http://www.radiokot.ru/start/mcu\\_fpga/avr/15/](http://www.radiokot.ru/start/mcu_fpga/avr/15/)



МІНІСТЕРСТВО ОСВІТИ І НАУКИ, МОЛОДІ ТА СПОРТУ УКРАЇНИ  
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
"Київський політехнічний інститут"



# СВІДОЦТВО ПРО НАДАННЯ ГРИФА

ЕЛЕКТРОННОМУ НАВЧАЛЬНОМУ ВИДАННЮ

НМУ № Е 11/12-011

*Основи мікропроцесорної техніки: Цикл 3: Мікроконтролери AVR;  
Методичні вказівки до виконання лабораторних робіт для студентів  
напряму підготовки 6.050803 "Акустотехніка", 6.050903  
"Телекомунікації" усіх форм навчання*

автор (-и) (укладач (-и)) Тітков Дмитро Валерійович,  
Швайченко Володимир Борисович

обсяг 82 стор.

надано гриф «Рекомендовано Методичною радою НТУУ «КПІ»

від « 22 » вересня 20 11 р., протокол № 1

Голова Методичної ради НТУУ «КПІ»  
перший проректор



Ю.І. Якименко