Лабораторна робота №5 Моделювання цифрового вольтметра

Тема: Моделювання цифрового вольтметра

Мета: Користуючись пакетом PROTEUS дослідити моделювання цифрового вольтметра

1 Порядок виконання роботи

- створити модель пристрою в пакеті Proteus 8.6
- розробити схему алгоритму роботи моделі та робочу програму
- створити hex-файл та підключити його до мікроконтролера
- запустити модель та виконати її дослідження згідно методичних вказівок
- зробити відповідні висновки.

2 Стислі теоретичні відомості

1 Особливості архітектури модуля АЦП у складі мікроконтролера ATmega32

Основним елементом цифрового вольтметра є аналого-цифровий перетворювач, в якості якого використовується відповідний модуль мікроконтролера ATmega32. Архітектуру цього модуля цього модуля розглянуто у підрозділах 4.3.2.1...4.3.2.9.

2 Опис моделі

Робочу модель цифрового вольтметра показано на рисунку 1.

З лівого боку моделі зображено 7-сегментний чотирьохпозиційний цифровий дисплей, який з’єднано з портами C та D мікроконтролера ATMega32 відповідними лініями зв’язку. У верхній частині рисунку знаходиться батарея BAT1 та резистор RV1, напругу з виходу якого можна змінювати. Ця вхідна аналогова напруга подається на лінію PA0 (ADC0) порту A мікроконтролера та перетворюється модулем АЦП у цифровий еквівалент. Особливості керування цифровим дисплеєм описано у попередньому підрозділі. Різниця між виведенням результату

перетворення АЦП, який описано вище, полягає в тому, що раніше на дисплей виводилася цифрова інформація у чотирьохрозрядному десятковому коді, а в моделі цифрового вольтметра відображається абсолютне значення вхідної напруги у вольтах з точністю до сотих долей вольта.

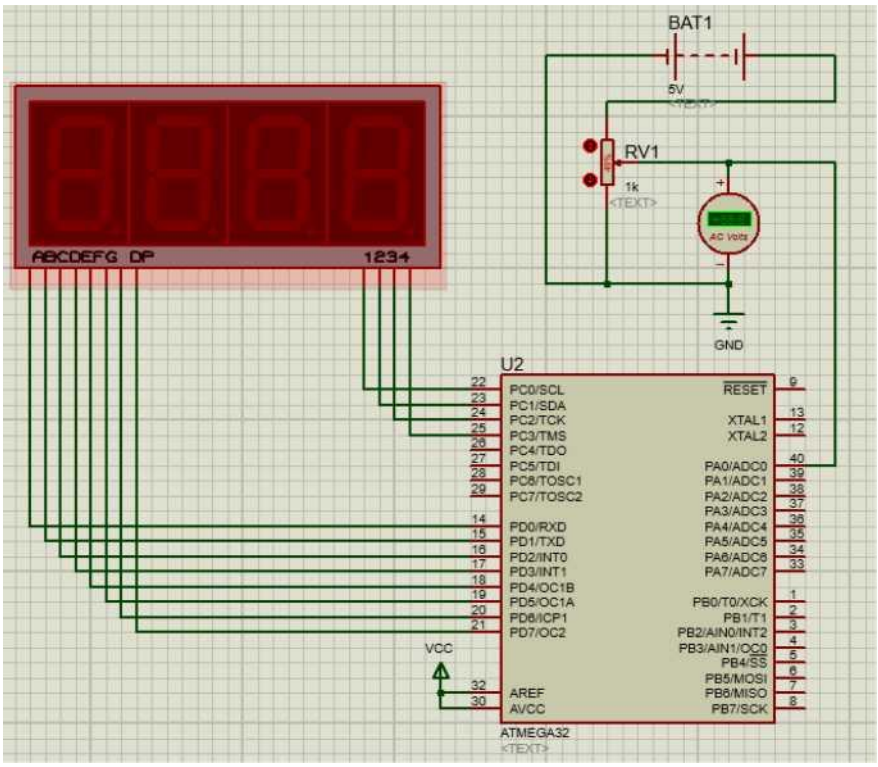

Рисунок 1 - Схема моделі цифрового вольтметра

На рисунку 2 наведено зовнішній вигляд 7-сегментного індикатора, який входить до складу чотирьохпозиційного цифрового дисплею. Для отримання на індикаторі чисел потрібно керувати сегментами індикатора A, B, C, D, E, F, G та точкою Dp згідно до таблиць 1, 2.

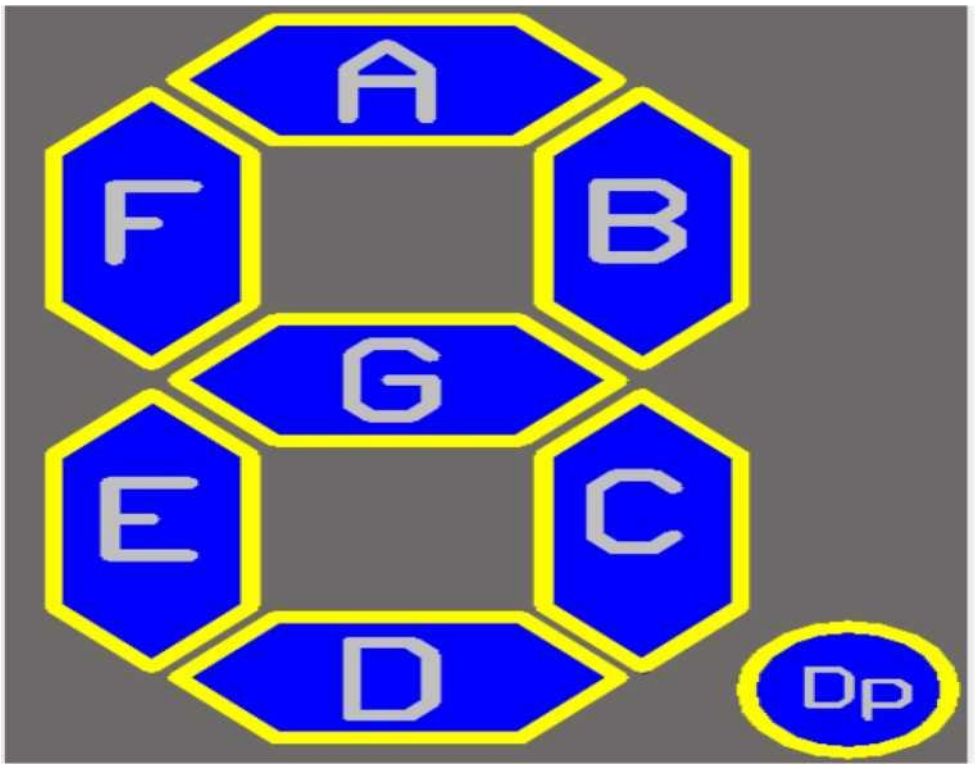

Рисунок 2 - 7-сегментний дисплей

Таблиця 1 - Зв’язок між цифрами на дисплеї та значенням керуючих сигналів без використання точки

Цифра на дисплеї	Dp	G	F	E	D	C	B	A	Значення керуючих сигналів
0	0	0	1	1	1	1	1	1	0x3F
1	0	0	0	0	0	1	1	0	0x06
2	0	1	0	1	1	0	1	1	0x5B
3	0	1	0	0	1	1	1	1	0x4F
4	0	1	1	0	0	1	1	0	0x66
5	0	1	1	0	1	1	0	1	0x6D
6	0	1	1	1	1	1	0	1	0x7D
7	0	0	0	0	0	1	1	1	0x07
8	0	1	1	1	1	1	1	1	0x7F
9	0	1	1	0	0	1	1	1	0x6F

Таблиця 2 - Зв’язок між цифрами з точкою на дисплеї та значенням керуючих сигналів з використанням точки

Цифра на дисплеї	Dp	G	F	E	D	C	B	A	Значення керуючих сигналів
0	1	0	1	1	1	1	1	1	0xBF
1	1	0	0	0	0	1	1	0	0x86
2	1	1	0	1	1	0	1	1	0xDB
3	1	1	0	0	1	1	1	1	0xCF
4	1	1	1	0	0	1	1	0	0xE6
5	1	1	1	0	1	1	0	1	0xED
6	1	1	1	1	1	1	0	1	0xFD
7	1	0	0	0	0	1	1	1	0x87
8	1	1	1	1	1	1	1	1	0xFF
9	1	1	1	0	0	1	1	1	0xEF

Нижче на рисунках 3, 4 наведено декілька прикладів роботи моделі.

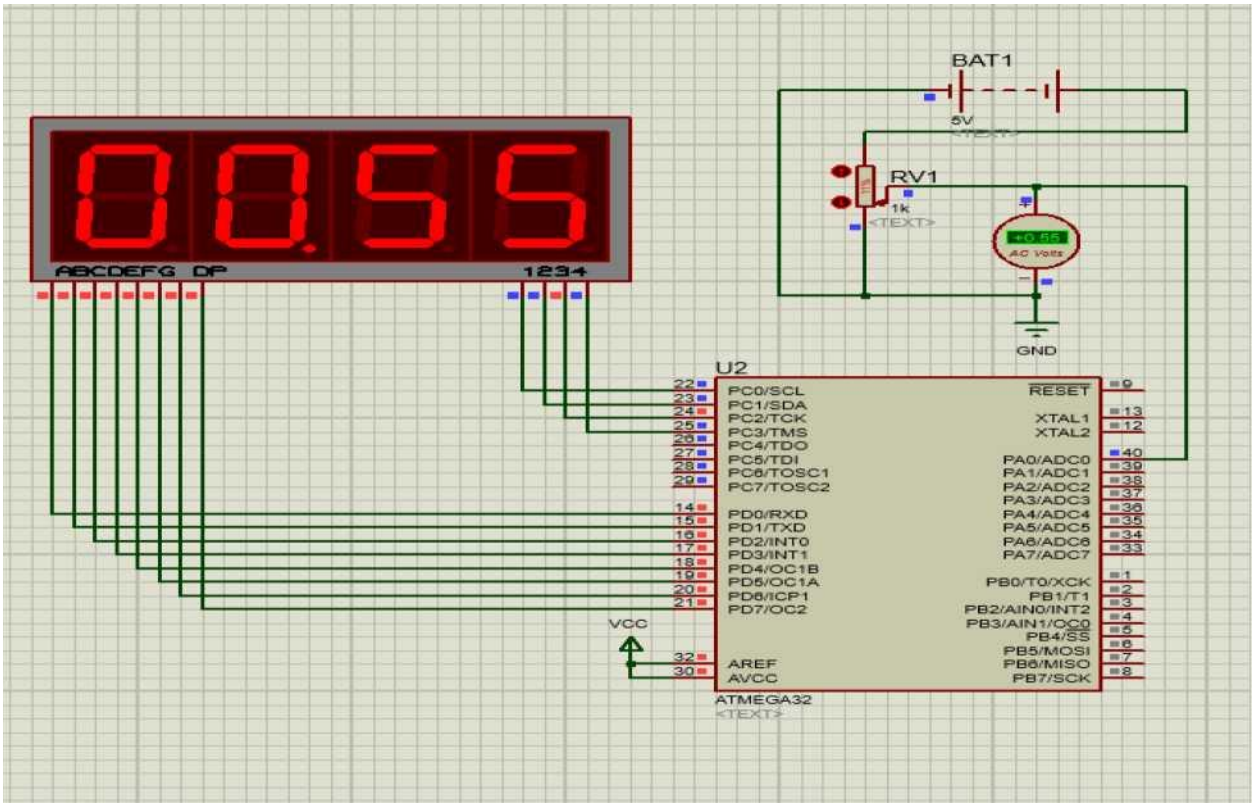

Рисунок 3 - Робота цифрового вольтметра при $U_{ВХ}$ = 0,55В

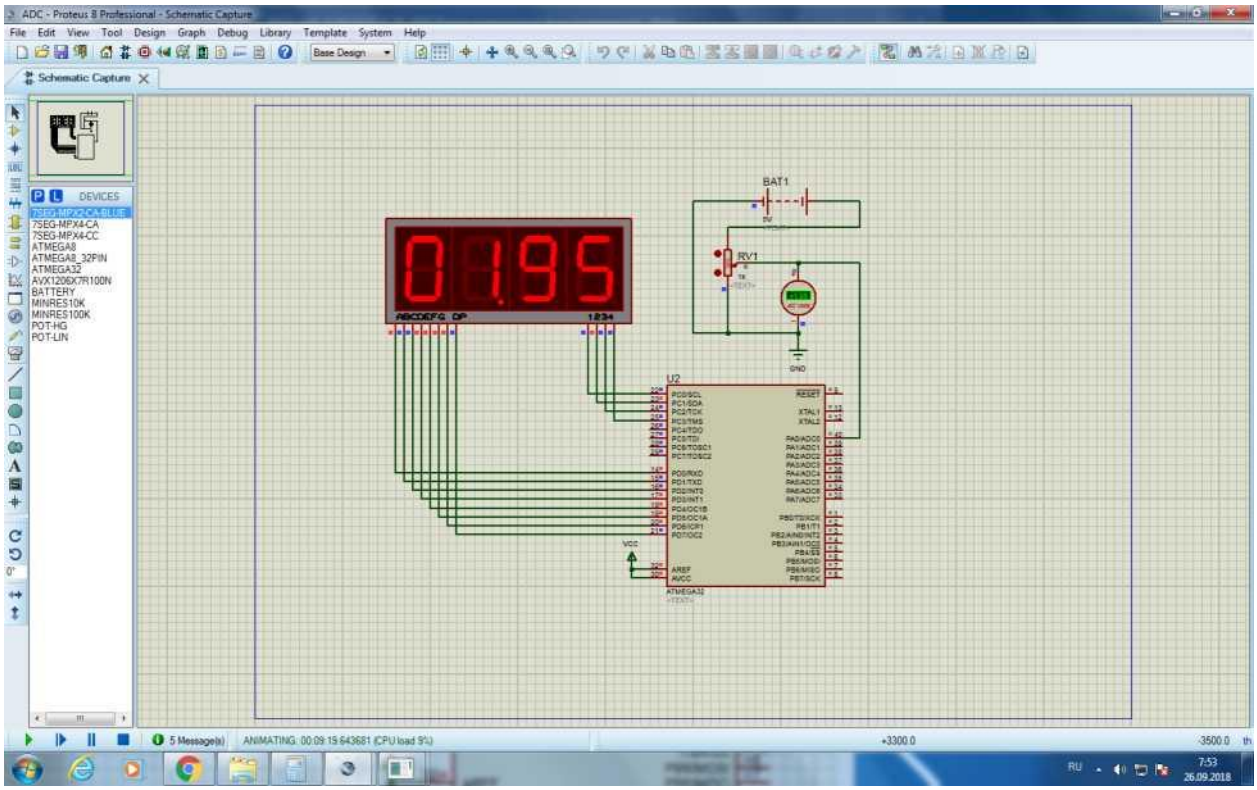

Рисунок 4 - Робота цифрового вольтметра при $U_{ВХ}$ = 1,95В

Схема алгоритму роботи модуля

Схема алгоритму роботи цифрового вольтметра подібна схемі алгоритму роботи модуля АЦП, яку наведено у лабораторній роботі 4 на рисунках 14…17.

Ініціалізація модуля АЦП, таймера Т2 мікроконтролера та робота моделі у більшості виконується аналогічно описанному в лабораторній роботі 4у підрозділі 2.10. Різниця полягає у виведенні результату моделювання на дисплей.

Нижче наведено пояснення перетворення значення ADC, яке отримується після завершення роботи модуля АЦП у десятковому коді, у значення вхідної напруги у вольтах для його виведення на дисплей.

Як відмічено в лабораторній роботі 4 у підрозділі 2.9 для каналів з одно полярним (несиметричним) входом результат перетворення АЦП визначається виразом:

$$ADC = 1023 \cdot U_{IN}/U_{REF},$$

де UIN - значення вхідної напруги, UREF - величина опорної напруги, ADC - десятковий еквівалент двійкового коду на виході АЦП.

Коефіцієнт передачі АЦП

$$К_{ПЕР} = 1023/U_{REF}\ [МЗР/мВ].$$

При U_{REF} = 5В, $К_{ПЕР}$ = 1023/5000 = 0,2046 [МЗР/мВ].

Значення вхідної напруги

$$U_{IN} = \frac{ADC}{K_{пер}} = \frac{ADC \cdot 5}{1023}.$$

Наприклад, якщо на вхід АЦП було подано напругу 0,55 В, тоді згідно з результатом моделюванням АЦП (див. підрозділ 5.4.10) ADC = 113.

Перетворення значення ADC_value = ADC в змінну «display», виконується згідно з виразом:

$$display = (ADC\_value) \cdot (5/1023) \cdot 100.$$

Тоді при ADC = 113

$$display = 0113 \cdot (5/1023) \cdot 100 = 55.$$

Тобто, при U_{IN} = 0,55В значення «display» згідно з робочою програмою дорівнює 55. Для виведення цього значення використовується математична операція «остача», яка в мові програмування «С» обчислюється оператором «%».

Це значення програма обробляє наступним чином:

1) 55%10000/1000 = 55/1000 = 0,055. Цифра 0 виводиться на перший індикатор.
2) Далі відбуваються наступні дії:
   на другий індикатор завжди виводиться число з крапкою
   55%1000/100 = 55/100 = 0,55. Цифра 0 з десятковою крапкою виводиться на другий індикатор;
3) 55%100/10 = 55/10 = 5,5. Цифра 5 виводиться на третій індикатор;
4) 55%10/1 = 5/1 = 5. Цифра 5 виводиться на четвертий індикатор.

Наприклад, якщо на вхід АЦП було подано напругу 1,95В, тоді згідно з моделюванням АЦП (див. підрозділ 5.4.10) ADC = 399.

Значення «display = 0399 · (5/1023) · 100 = 195.

Це значення програма обробляє наступним чином:

195%10000/1000 = 195/1000 = 0,195. Цифра 0 виводиться на перший індикатор.

195%1000/100 = 195/100 = 1,95. Цифра 1 з десятковою крапкою виводиться на другий індикатор;

195%100/10 = 95/10 = 9,5. Цифра 9 виводиться на третій індикатор;

195%10/1 = 5/1 = 5. Цифра 5 виводиться на четвертий індикатор.

Робоча програма мовою С

```
// Підключення файлів бібліотек
#include <avr/io.h>
#include <avr/interrupt.h>
#include <util/delay.h>

// Масив значень для відображення цифр на семисегментних індикаторах char SEG[] = {0x3F, 0x06, 0x5B, 0x4F, 0x66, 0x6D, 0x7D, 0x07, 0x7F, 0x6F};
char SEG_with_dot[] = {0xBF, 0x86, 0xDB, 0xCF, 0xE6, 0xED, 0xFD, 0x87, 0xFF, 0xEF};

// Ініціалізації глобальних змінних
volatile unsigned char current_indicator = 0;
volatile unsigned int display = 0;
volatile unsigned int ADC_value;

// Блок 1 - Ініціалізація програми
int main (void)
{
   // Блок 1.1 - Обнулення портів семисегментних індикаторів
   DDRC = 0xFF;
   PORTC = 0x00;
   DDRD = 0xFF;
   PORTD = 0x00;

   // Блок 1.2 - Налаштування таймера T2
   TIMSK |= (1 << TOIE2); // Дозвіл переривання від таймера T2
   TCCR2 |= (1 << CS21); // Переддільник на 8
   // Блок 1.3 - Налаштування АЦП
   ADCSRA |= (1 << ADEN) // Дозвіл АЦП
   |(1 << ADSC) // Запуск перетворення
```

```
    |(1 << ADATE) // Безперервний режим роботи АЦП
    |(1 << ADPS2)|(1 << ADPS1) // Переддільник на 64
    |(1 << ADIE); // Дозвіл переривань від АЦП

    ADMUX &= (~(1 << REFS1))&(~(1 << REFS0)); // Зовнішнє ДОН

    // Блок 1.4 - Глобальний дозвіл переривань sei();

    // Блок 1.5 - Основний цикл
    while(1)
    {
        _delay_ms(50); // Затримка 50 мс
    }
}
// Блок 2 - Умова переривання від АЦП
ISR (ADC_vect)
{// Блок 3 - Обробник переривань від АЦП
    ADC_value = ADC; // Блок 3.1 - Присвоювання глобальній змінній поточне значення
АЦП
}

// Блок 4 - Умова переривання від таймера T2
ISR (TIMER2_OVF_vect)
{  // Блок 5 - Обробник переривань від таймера T2
    PORTD = 0xFF; // Блок 5.1 - Вимикання всіх сегментів
    PORTC = (1 << current_indicator); // Блок 5.2 - Обирання поточного індикатора
                                          //починаючи зліва направо)
    display = (ADC_value)*(5/1023)*100; // Обрахування значення напруги, яка
                                              //виводиться на дисплей у вольтах
    //Блок 5.3
    switch (current_indicator)
    {
        case 0:
        PORTD = ~(SEG[display % 10000 / 1000]); // Вмикання цифри десятків break;
        case 1:
        PORTD = ~(SEG_with_dot [display % 1000 / 100]); // Вмикання цифри //одиниць
        break;
        case 2:
        PORTD = ~(SEG[display % 100 / 10]); // Вмикання цифри десятих
```

```
    break;
    case 3:
    PORTD = ~(SEG[display % 10 / 1]); // Вмикання цифри сотих
    break;
  }
  if ((current_indicator++) > 2) // Блоки 5.4; 5.5 - Перехід на наступний //індикатор,
                                 якщо current_indicator не більше двох
  current_indicator = 0; // Блок 5.6 - Обнулення current_indicator, якщо він //більше двох
}
```

3 Зміст звіту

Звіт по роботі повинен містити:

- схему моделі;
- схему алгоритму роботи моделі;
- робочу програму;
- формули за потребою.

Контрольні запитання

1. Опишіть відмінності в програмах при моделюванні модуля АЦП та цифрового вольтметра.
2. Опишіть особливості програмування мовою С виведення на дисплей результату роботи модуля АЦП та цифрового вольтметра.
3. Як розраховується відносна похибка АЦП від квантування за рівнем?
4. Опишіть схему моделювання модуля АЦП в пакеті PROTEUS 8.6.
5. В який режим програмується модуль АЦП при його моделюванні?
6. Чому дорівнює коефіцієнт передачі модуля АЦП при його моделюванні?
7. В якому вигляді виводиться на індикацію результат моделювання модуля?
8. За допомогою якої математичної операції визначається відповідна цифра при її виведенні на індикатор? Відповідь пояснити.
9. Який таймер/лічильник мікроконтролера використовується при виведенні на семисегментні індикатори?
10. Який сигнал використовується в якості тактового для таймера 2? Відповідь пояснити.
11. Чим відрізняється виведення результату моделювання цифрового вольтметра від моделювання модуля АЦП?