

Лабораторна робота № 2 ПРОГРАМА УПРАВЛІННЯ СВІТЛОДІОДАМИ

Мета роботи: ознайомитись з принципом управління дискретними пристроями, що приєднані безпосередньо до портів мікроконтролера, а також програмним забезпеченням WinAVR, Codevision AVR, Flowcode for AVR. Набути навиків створення та компіляції власної програми для мікроконтролера.

Обладнання: навчально-відлагоджувальна плата AVR-Easy; мікроконтролери ATtiny2313, ATmega8, ATmega16, Atmega8515; середовища програмування WinAVR, Codevision AVR, Flowcode for AVR; внутрішньосхемний програматор; програма для прошивки мікроконтролерів AVR8 Bum-O-Mat.

Теоретичний матеріал

Порти введення / виведення МК AVR мають число незалежних ліній "Вхід / Вихід" від 3 до 53. Кожен розряд порту може бути запрограмований на введення або на виведення інформації. Вихідні драйвери забезпечують струмову навантажувальну здатність 20 мА на лінію порту при максимальному значенні 40 мА, що дозволяє, наприклад, безпосередньо підключати до мікроконтролера світлодіоди і біполярні транзистори. Загальна струмова навантаження на всі лінії одного порту не повинна перевищувати 80 мА (всі значення наведено для напруги живлення 5 В).

Для роботи з будь-яким портом «x» існують три регістри управління: DDRx, PORTx, PINx.

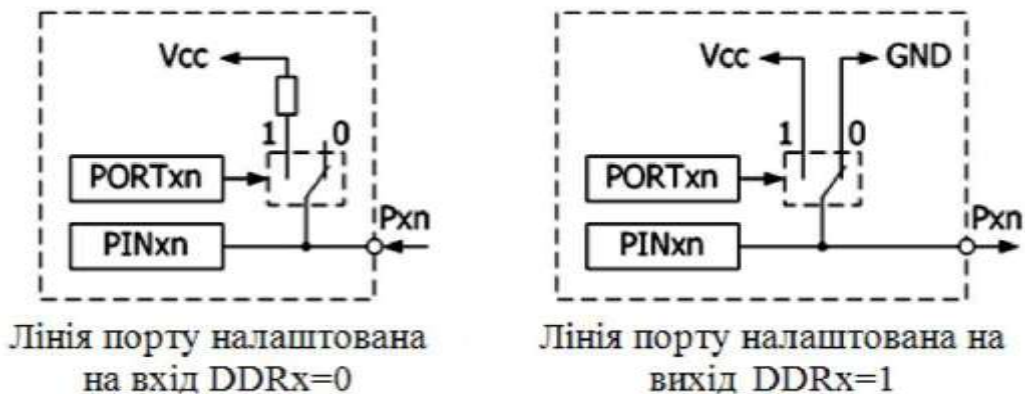


Рис. 2.1. Схеми налаштування портів на режими введення / виведення

DDR_x (The Port X Data Direction Register) - визначає напрям передачі даних кожної лінії порту X: 1 - вихід, 0 - вхід. Регістр DDR_x містить 8 біт DD_x0 .. DD_x7, кожен з яких відповідає за свою лінію порту, яка збігається з номером P_x0 .. P_x7.

PORT_x (The Port X Data Register) - регістр даних порту «x». Принцип роботи з цим регістром залежить від того, в якому режимі, входу або виходу працює лінія порту. Якщо лінія порту працює на вхід, то біти даного регістра відповідають за підключення до лінії внутрішнього опору, який підтягує напругу лінії до напруги живлення. Якщо лінія порту працює на вихід, то біти даного регістра керують

станом вихідної лінії.

PINx (The Port X Input Pins Address) - діапазон адрес, читання по яким надає доступ до інформації з буферних регістрів на вході порту «x». Даний регістр призначений тільки для зчитування станів ліній порту.

Плата містить 32 світлодіода, підключених до портів А, В, С, D мікроконтролера. Для того щоб засвітити світлодіод необхідно ввімкнути потрібний порт перемикачем SW6 і записати '1' в потрібний розряд. Тобто, низький високий рівень на портах мікроконтролера вмикає світлодіод, низький вимикає.

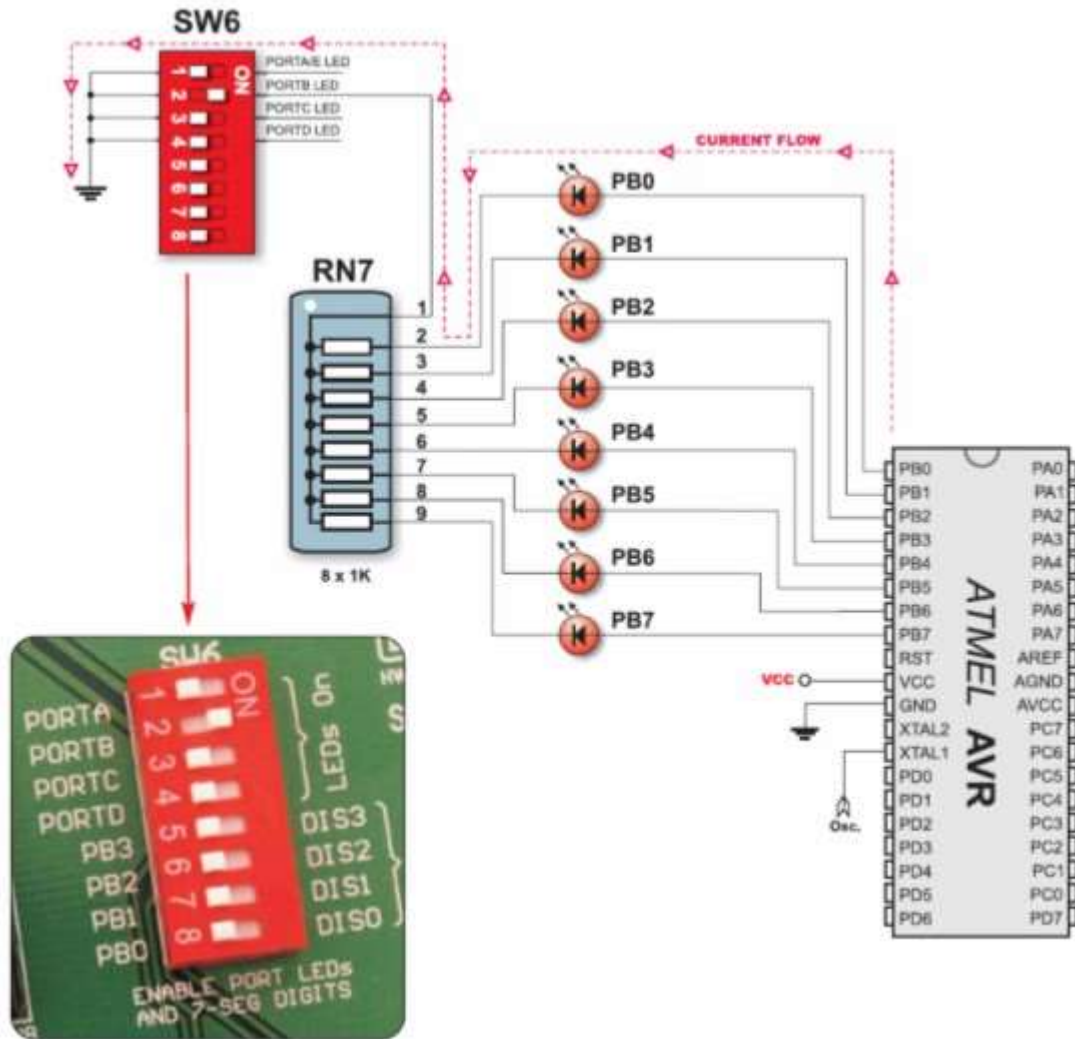


Рис. 1. Схема ввімкнення світло діодів

При виконанні лабораторних робіт для розробки програм рекомендовано використання AVR Studio.

```
Приклад програми миготіння двома світлодіодами порту В
# define F_CPU 8000000UL           // Вибираємо частоту МК
#include <avr/io.h>                 // Для роботи с портами
#include <util/delay.h>             // для затримки
```

```

int main(void)
{
    // 2 и 3 виводи порту В як вихід
    DDRB = 0b00001100;           // двійковий код
    DDRB = 0x0C;                 // 16 - вий код
    DDRB = 12;                   // 10 - вий код
    DDRB |= ( 1 << 2 ) | ( 1 << 3 ); // розряди 2 та 3 порта В на вихід

    // на 2 и 3 виводи подаємо логічну 1 ( + 5В )
    PORTB |= ( 1 << 2 ) | ( 1 << 3 );

    while(1)
    {
        // мигаємо 2 и 3 виводами
        PORTB ^= 0b00001100;
        _delay_ms(200);
    }
}

```

Для затримки використовуємо функцію `_delay_ms ()`. Функція `_delay_ms ()` формує затримку в залежності від переданого їй аргументу, вираженого в мілісекундах (в одній секунді 1000 мілісекунд). Максимальна затримка може досягати 262.14 мілісекунд. Якщо користувач передасть функції значення більш 262.14, то відбудеться автоматичне зменшення дозволу до 1/10 мілісекунди, що забезпечує затримки до 6.5535 секунд. Функція `_delay_ms ()` міститься в файлі `delay.h`, тому нам буде необхідно підключити цей файл до програми. Крім того, для нормальної роботи цієї функції необхідно вказати частоту, на якій працює мікроконтролер, в герцах.

В наступній програмі організований нескінченний цикл за допомогою оператора безумовного переходу "goto".

```

#define F_CPU 8000000UL // вказуємо частоту в герцах

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

int main ( void ) { // початок основної програми

    DDRD = 0xff; // все висновки порту D конфігурувати як виходи

start: // мітка для команди goto start

    PORTD |= _BV (PD1); // встановити "1" (високий рівень) на виведення PD1,
    // Запалити світлодіод

```

```

_delay_ms (250);      // чекаємо 0.25 сек.

PORTD & = ~ _BV (PD1); // встановити "0" (низький рівень) на виведення
PD1,
                        // Погасити світлодіод

_delay_ms (250);      // чекаємо 0.25 сек.

goto start;           // перейти до мітки start

}                     // Закриває дужка основної програми

```

У бібліотеці avr/io.h знаходяться деякі функції вводу-виводу та описи регістрів МК. Без цієї бібліотеки не обходиться жодна програма для AVR. У бібліотеці util/delay.h розміщені функції затримки (паузи).

Головна функція - main. Мікроконтролер виконує її після скидання або ввімкнення живлення. Вона закінчується нескінченним циклом та поверненням нуля.

Перш за все відбувається налаштування портів. Команда DDR* вказує напрям роботи. При передачі команді «1» - відповідний пін порту налаштовується на «вихід». При передачі «0» - відповідний пін порту налаштовується на «вхід». Команда PORTx налаштовує стан порту на початку програми.

Далі йде нескінченний цикл for(;;). На відміну від комп'ютерної програми, програма мікроконтролера працює у нескінченному циклі до вимкнення живлення або скидання МК.

Функція _delay_ms(10) з бібліотеки util/delay.h організує затримку 10 мілісекунд. Написати _delay_ms(500) не можна, бо максимальне значення аргументу 262.14 мс, поділене на тактову частоту в МГц. Безпечне значення для усіх мікроконтролерів AVR та тактових частот є 10 мс. Щоб зробити затримку 0,5 с (500 мс), потрібно повторити 50 разів затримку 10 мс.

Для того, щоб не всі світлодіоди блимали, то необхідно вмикати лише потрібні світло діоди, наприклад, другий, третій та сьомий світлодіоди. Для цього треба виставляти в "1" лише біти 1, 2 та 6 (згадаємо, що перший світлодіод під'єднаний до біту 0), інші біти - в "0". У програму потрібно вписати деяке число у шістнадцятковій системі числення (HEX). Приставка "0x" означає, що число записане у шістнадцятковому вигляді. Щоб його отримати, запишемо значення всіх восьми бітів, починаючи з найстаршого.

Таблиця 2.1

Приклади кодування станів поту в шістнадцятко вій системі

Біти								Число у HEX- форматі
7	6	5	4	3	2	1	0	
0	1	0	0	0	1	1	0	0x46
1	1	0	1	1	0	0	1	0xD9
0	1	0	1	1	1	1	1	0x5F

Розділимо біти на групи по 4 і замінимо кожену групу однією цифрою згідно таблиці. Отримаємо потрібне число.

Таблиця 2.2

Таблиця відповідності двійкових значень шістнадцятковим

Двійкове число				Шістнадцяткова цифра	Двійкове число				Шістнадцяткова цифра
0	0	0	0	0	1	0	0	0	8
0	0	0	1	1	1	0	0	1	9
0	0	1	0	2	1	0	1	0	A
0	0	1	1	3	1	0	1	1	B
0	1	0	0	4	1	1	0	0	C
0	1	0	1	5	1	1	0	1	D
0	1	1	0	6	1	1	1	0	E
0	1	1	1	7	1	1	1	1	F

Лінійка світлодіодів знаходиться у двох станах по черзі. У першому стані усі світлодіоди горять (0xFF), у другому усі погашені (0x00). Загалом, якщо в обох станах значення біту рівне “1”, відповідний світлодіод завжди горить, якщо “0”, то не горить, а якщо значення біту змінюється, то світлодіод блимає.

Хід роботи

1. Розібратися з принципом роботи тестової програми та призначенням кожного оператора у програмі.
2. Створити свою програму згідно з індивідуальним завданням та скомпілювати її.
3. Під'єднати внутрішньосхемний програматор до плати та комп'ютера. Увімкнути плату.
4. Запрограмувати МК.
5. Перевірити правильність виконання програми.

Індивідуальні завдання

Створити програму для свого типу мікроконтролера, яка виконує наступні функції: світлодіоди з вказаними номерами увімкнені, вимкнені або блимають згідно з таблицею.

Таблиця 2.3

Варіанти індивідуальних завдань

Варіант	Увімкнені	Вимкнені	Блимають
1	1,2	3,4	5, 6, 7, 8
2	3,6	2,7	1, 4, 5, 8
3	3, 6, 8	2,5	1, 4, 7
4	2,5	1, 7, 8	3, 4, 6
5	2,3	4,5	1, 6, 7, 8
6	2, 3, 4	1,5	6, 7, 8
7	1, 5, 8	3,6	2, 4, 7
8	2,4	6,8	1, 3, 5,7
9	1,7	3,5	2, 4, 6, 8
10	4,5	3,6	1, 2, 7, 8
11	1	2, 4, 6	3, 5, 7, 8
12	1,8	2,7	3, 4, 5,6
13	3, 4, 8	6,7	1, 2, 5
14	6, 7, 8	3	1, 2, 4, 5
15	1, 2, 8	3, 6, 7	4, 5, 6

Контрольні запитання

1. Кодування чисел в двійкову та шістнадцяткову системи числення;
2. Час виконання команд (поняття такту, машинного циклу);
3. Призначення портів введення/виведення AVR-мікроконтролерів;
4. Структура порту введення/виведення AVR-мікроконтролерів;
5. Регістри керування портами введення/виведення AVR-мікроконтролерів та їх призначення.

Зміст звіту

1. Тема та мета роботи.
2. Перелік використаного обладнання.
3. Стислий зміст теоретичних відомостей.
4. Лістинг власної програми з детальним поясненням кожного рядка.
5. Відповіді на контрольні запитання.
6. Висновки.

