

Лабораторна робота № 4 ВИКОРИСТАННЯ КНОПОК ДЛЯ КЕРУВАННЯ РОБОТОЮ МІКРОКОНТРОЛЕРА

Мета роботи: ознайомитись з портами вводу-виводу мікроконтролера, принципом обробки сигналів дискретних датчиків. Набути навичок відображення інформації за допомогою світлодіодних індикаторів.

Обладнання: навчально-відлагоджувальна плата AVR-Easy; мікроконтролери ATtiny2313, ATmega8, ATmega16, Atmega8515; середовища програмування WinAVR, Codevision AVR, Flowcode for AVR; внутрішньосхемний програматор.

Теоретичний матеріал

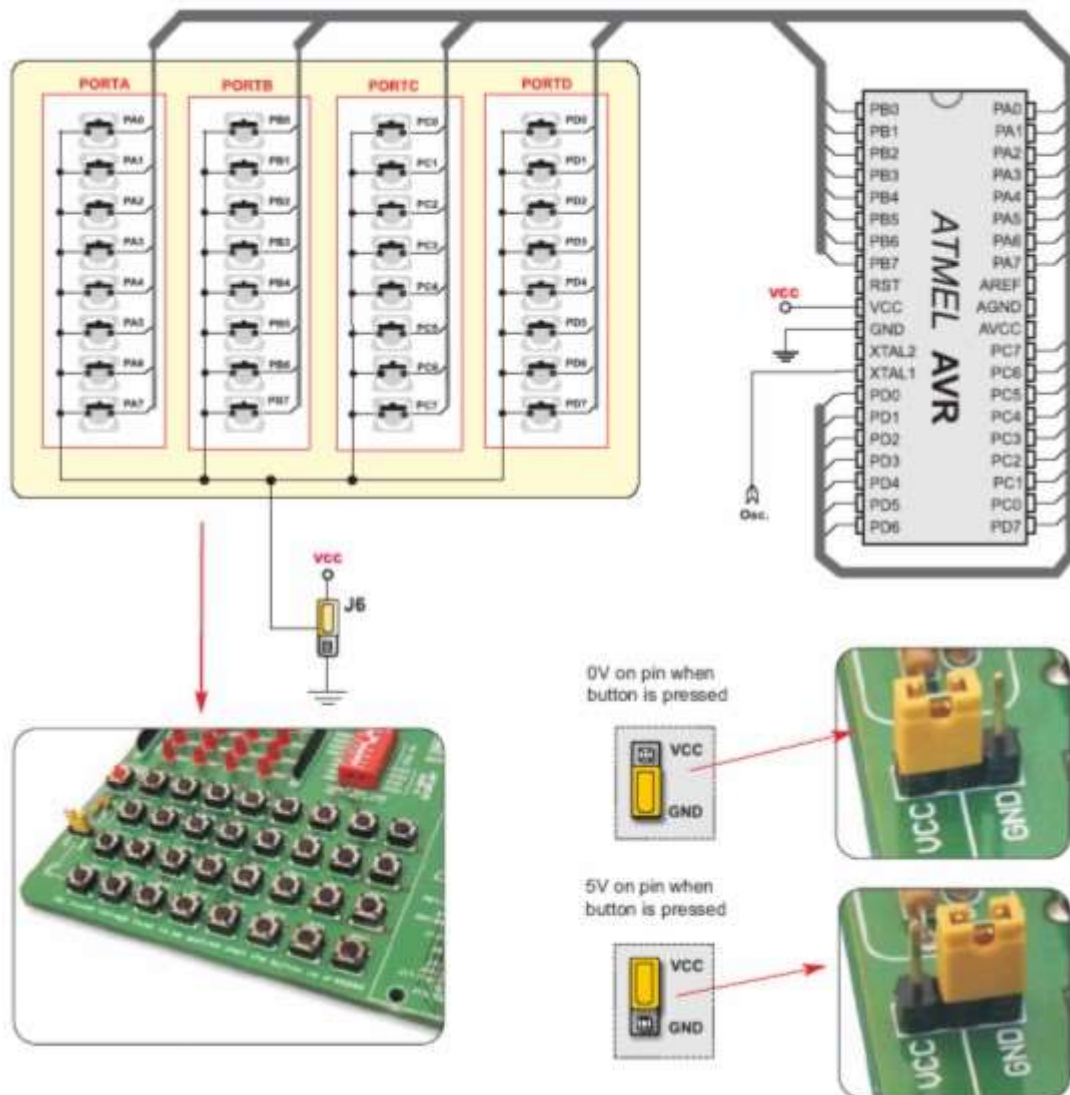


Рис. 2.8. Схема підключення кнопок до мікроконтролера на навчально - відлагоджувальній платі AVR-Easy-Kit.

Навчальна відлагоджувальна плата містить кнопку початкової установки контролера RESET і 32 кнопки, що дозволяють симулювати вхідні дії на порти А, В, С, D. Загальний вивід кнопок може бути підключений до ланцюга живлення або "землі" за допомогою J6. Це дає змогу посилати в порти мікроконтролера сигнали високого або низького рівня.

Управління світлодіодами за допомогою кнопок

Якщо натиснута кнопка D0, горить два з лінійки світлодіодів порту В, а якщо ні то всі світлодіоди. J6 підключено до 5 В.

```
#define F_CPU 8000000UL // частота в герцах

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

int main(void) {

    DDRB = 0xff;          // всі виводи порту В сконфігурувати як виходи
    DDRD = 0x00;         // всі виводи порту D сконфігурувати як входи
    PORTB = 0xff;        // подати всі виводи порту В логічні одиниці
    PORTD = 0x00; // виводи порту D в високоімпедансний стан
start:                  //

    if ((PIND&0x01)==1) // перевірка чи натиснута клавіша D0
    { // зчитуємо стан порту D

        _delay_ms(50); // затримка для усунення брязкоту контактів
        if ((PIND&0x01)==1) // перевірка чи ще натиснута клавіша D0
            PORTB = 0x88; // запалення двох світлодіодів
        }
        else
            PORTB = 0xff;

    goto start;

}

}
```

На початку програми усі лінії порту В встановлюються як виходи з високим рівнем. Усі лінії порту D встановлюються як входи з високоімпедансним станом.

Управління індикаторами за допомогою кнопок

Якщо натиснута кнопка D0, на лівій частині індикатора горить число 99, інакше - 11. Якщо натиснута кнопка D1- то на правій частині горить 87, інакше - 23. Кожна кнопка управляє своєю половиною індикатора незалежно одна від одної. Завжди горять перша і четверта десяткові крапки.

```
// A B C D E F G H(dp)
// A0 A1 A2 A3 A4 A5 A6 A7

// Digit 1      Digit2      Digit3      Digit4
// B0          B1          B2          B3

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

```

#include <util/delay.h>

int main(void)
{
DDRA = DDRB = 0xFF; // всі розряди портів D і A на вихід
DDRD = 0b00000000; // всі розряди порта D на вхід
PORTB = 0xFF;
PORTD = 0xFF; // включення на D внутрішніх підтягуючих резисторів до 1

unsigned char i=20; //затримка мікросекунд

for(;;)
{
PORTB |= _BV(0);
PORTB &= ~_BV(1);
PORTB &= ~_BV(2);
PORTB &= ~_BV(3); //вмикаємо розряд "1", всі інші – вимикаємо
// визначення чи нульовий біт D дорівнює нулю
if(bit_is_clear(PIND, PD0)) PORTA = 0b00000110;
// комбінація вмикає цифру "1"
else PORTA = 0b01101111; // комбінація вмикає цифру "9"
_delay_us(i);
//часова затримка в мікросекундах на свічення даного розряду
PORTD &= ~_BV(0);
PORTD |= _BV(1);
PORTD &= ~_BV(2);
PORTD &= ~_BV(3); //вмикаємо розряд "2", всі інші – вимикаємо
// визначення чи нульовий біт D дорівнює нулю
if(bit_is_clear(PIND, PD0)) PORTA = 0b00000110;
// комбінація вмикає цифру "1"
else PORTA = 0b01101111; // комбінація вмикає цифру "9"
_delay_us(i);

PORTD &= ~_BV(0);
PORTD &= ~_BV(1);
PORTD |= _BV(2);
PORTD &= ~_BV(3);
//вмикаємо розряд "3", всі інші - вимикаємо
if(bit_is_clear(PIND, PD1)) PORTA = 0b01011011;
// комбінація вмикає цифру "2"
else PORTA = 0b01111111; // комбінація вмикає цифру "8"
_delay_us(i);

PORTD &= ~_BV(0);

```

```

PORTD &= ~_BV(1);
PORTD &= ~_BV(2);
PORTD |= _BV(3); //вмикаємо розряд "4", всі інші - вимикаємо
if(bit_is_clear(PIND, PD1)) {PORTA = 0b01001111;}
// комбінація вмикає цифру "3"
else PORTA = 0b00000111; // комбінація вмикає цифру "7"
_delay_us(i);
}
return 0;
}

```

Хід роботи

1. Знайти на навчальній платі потрібні кнопки та світлодіодні індикатори.
2. Розібратися з принципом роботи тестових програм та призначенням кожного оператора у програмі.
3. Створити свою власну програму, яка забезпечує функції, описані в індивідуальному завданні, та скомпілювати її.
4. Запрограмувати МК, перевірити правильність виконання.

Індивідуальні завдання

1. На індикаторі світиться число 0. Коли натискають будь-яку кнопку, число збільшується на 1111 (вийдуть числа 1111, 2222 тощо) з інтервалом 300 мс. Після досягнення числа 9999 збільшення припиняється і програма повертається у початковий стан.
2. На індикаторі світиться число 1234. Коли натискають першу кнопку, число збільшується на 100, коли другу — на 1.
3. На індикаторі світиться число 9999. При натисненні першої кнопки остання цифра зменшується на 1 (виходить 9998), при наступному натисненні — третя цифра зменшується на 1 (виходить 9988), аналогічно для другої і першої цифри. Потім зменшується знову остання цифра.
4. Коли не натиснута жодна кнопка, світиться число 1111, коли натиснута перша — 8811, друга — 1188, обидві — 8888.
5. На індикаторі світиться число 4444. Коли натискають першу кнопку, число зменшується вдвічі, коли другу — збільшується на 123.
6. На індикаторі світиться число 9999. Коли натискають будь-яку кнопку, число зменшується на 1111 (вийдуть числа 8888, 7777 тощо) з інтервалом 300 мс. Після досягнення числа 0000 зменшення припиняється і програма повертається у початковий стан.
7. Коли не натиснута жодна кнопка, світиться число 9999, коли натиснута перша — 4949, друга — 9494, обидві — 4444.
8. На індикаторі світиться число 4320. Коли натискають будь-яку кнопку, число збільшується на 40 (вийдуть числа 4360, 4400 тощо) з інтервалом

300 мс. Після досягнення числа, більшого за 5000, збільшення припиняється і програма повертається у початковий стан.

9. На індикаторі світиться число 8675. Коли натискають будь-яку кнопку, число зменшується на 85 (вийдуть числа 8590, 8505 тощо) з інтервалом 300 мс. Після досягнення числа, меншого за 7500, зменшення припиняється і програма повертається у початковий стан.

10. На індикаторі світиться число 1. Коли натискають будь-яку кнопку, число збільшується вдвічі (вийдуть числа 2, 4, 8 тощо) з інтервалом 300 мс. Після досягнення числа, більшого за 9999, збільшення припиняється і програма повертається у початковий стан.

11. На індикаторі світиться число 0. При першому натисненні першої кнопки перша цифра збільшується на 1 (виходить 1000), при другому натисненні — друга цифра збільшується на 1 (виходить 1100), аналогічно для третьої і четвертої цифри. Потім збільшується знову перша цифра.

12. На індикаторі світиться число 1. Коли натискають будь-яку кнопку, число збільшується втричі (вийдуть числа 3, 9, 27 тощо) з інтервалом 300 мс. Після досягнення числа, більшого за 9999, збільшення припиняється і програма повертається у початковий стан.

13. На індикаторі число збільшується на 1 з інтервалом 1 мс. При натисненні першої кнопки збільшення припиняється та число тримається на індикаторі 5 секунд. Потім програма повертається у початковий стан. Рахунок продовжується з нуля.

14. На індикаторі світиться число 5678. Коли натискають першу кнопку, число зменшується на 200, коли другу — на 2.

15. Коли не натиснута жодна кнопка, світиться число 6543, коли натиснута перша — 1234, друга — 2198, обидві — 3333.

Контрольні запитання

1. Привести основні можливі схемні рішення для приєднання кнопок до мікроконтролерів.

2. Для кожного з приведених схемних рішень включення кнопок привести приклади попередньої ініціалізації порту та подальшого програмного методу опрацювання події натиснення на кнопку.

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.

Висновки.

Додаток

Робота з портами у мові C

Старе позначення	Можлива заміна	Дія
<code>outp(0xff,DDRB);</code>	<code>DDRB = 0xff;</code>	всі виводи порту В конфігурувати як виходи
<code>outp(0xff,PORTB);</code>	<code>PORTB = 0xff;</code>	у всіх битах порту В встановити "1"
<code>sbi(DDRB, DDB2);</code>	<code>DDRB = 1<<2;</code>	конфігурувати лінію 2 порту В як вихід
<code>cbi(DDRB, DDB2);</code>	<code>DDRB &= ~(1<<2);</code>	конфігурувати лінію 2 порту В як вхід
<code>sbi(PORTB,PB2);</code>	<code>PORTB = _BV(PB2);</code> або <code>PORTB = 1<<2;</code> або <code>PORTB = 1<<PINB2;</code>	встановити "1" на лінії 2 порту В
<code>cbi(PORTB,PB2);</code>	<code>PORTB &= ~_BV(PB2);</code> або <code>PORTB &= ~(1<<2);</code> або <code>PORTB &= ~1<<PINB2;</code>	встановити "0" на лінії 2 порту В
<code>if (bit_is_set(PIND,3))</code> { ... }	<code>if (PIND & (1<<PIND3))</code> { ... }	перевірити "1" на лінії 3 порту D
<code>if (bit_is_clear(PIND,3))</code> { ... }	<code>if (!(PIND & (1<<PIND3)))</code> { ... }	перевірити "0" на лінії 3 порту D

Зверніть увагу: використання `_BV ()` більш доцільно, так як в цьому випадку компілятор сам виконує порозрядному зрушення і вставляє результат в компільований код. Це забезпечує відсутність витрат часу під час безпосереднього виконання коду в мікроконтролері.