**Лабораторна робота № 6**

**Тема**: Використання паралельного інтерфейсу для виводу даних.

**Мета**:

1. Розглянути роботу з портами введення/виведення;

2. Побудувати програмні затримки часу без використання переривань;

3. Засвоїти принцип статичної індикації.

**Завдання:**

1. Ініціалізувати порт *B* для виведення даних;

2. Вивести на порт *B* початкову послідовність бітів (табл. 1.2);

3. Циклічно повторювати послідовність дій (табл. 1.1, табл. 1.2).

*Таблиця 1.1 Дії над поточним вмістом порту B*

|  |  |
| --- | --- |
| **Режим** | **Дія над поточним вмістом порту *B*** |
| 1 | Циклічний зсув праворуч через час *T*1 (протягом часу *T*2) |
| 2 | Циклічний зсув ліворуч через час *T*1 (протягом часу *T*2) |
| 3 | Інверсія через час *T*3 (протягом часу *T*4) |
| 4 | Інверсія тільки парних бітів через час *T*3 (протягом часу *T*4) |
| 5 | Інверсія тільки непарних бітів через час *T*3 (протягом часу *T*4) |
| 6 | Збереження парних бітів, зміна їх на 0, витримка часу *T*5,  відновлення парних бітів, витримка часу *T*5 (протягом часу *T*6) |
| 7 | Збереження парних бітів, зміна їх на 1, витримка часу *T*5,  відновлення парних бітів, витримка часу *T*5 (протягом часу *T*6) |
| 8 | Збереження непарних бітів, зміна їх на 0, витримка часу *T*5,  відновлення непарних бітів, витримка часу *T*5 (протягом часу *T*6) |
| 9 | Збереження непарних бітів, зміна їх на 1, витримка часу *T*5,  відновлення непарних бітів, витримка часу *T*5 (протягом часу *T*6) |

Для оцінки результатів виконання програми слід використовувати засоби емуляції, які надаються середовищем розробки *MPLAB*, та схему, яку наведено на рис. 1.1. Індикація змісту порту *B* забезпечується світлодіодами *VD*1-*VD*8, струм яких обмежується резисторами *R*1-*R*8.



Рис. 1.1 Схема підключення світлодіодів до порту B

**Теоретичні відомості**

Мікроконтролери *PIC*16*F*627/*PIC*16*F*628 мають 16 ліній введення/виведення, які відносяться до портів *A* (*RA*0-*RA*7) та *B* (*RB*0-*RB*7).

Кожна лінія порту може бути запрограмована на введення або на виведення інформації. Для управління лініями кожен порт має по два регістри: *TRISX* та *PORTX* (де *X* – ім’я порту). Регістр *TRISX* відповідає за напрям передачі інформації (запис 0 визначає режим виведення, а запис 1 – режим введення). Читання регістру *PORTX* повертає стан на виводах порту *X*, а запис призводить до передачі даних на виводи порту *X* у режимі виведення. Порт *B* додатково має можливість підключення внутрішніх підтягувальних резисторів. Це відбувається при скиданні біту *RBPU*# в регістрі *OPTION\_REG*.

Для реалізації затримок часу без використання апаратних модулів мікро-контролеру необхідно сформувати процедури, які виконують певну кількість команд. Виконання кожної команди відбувається за декілька машинних циклів (див. Додаток Б). Машинний цикл мікроконтролерів сімейства *PIC*16 відповідає 4 періодам тактового генератора. Найбільш компактні процедури для реалізації затримок часу складаються з декількох вкладених циклів з лічильником. Наведено приклад циклу з лічильником (*cnt\_cycle*) із зазначенням затримок часу (у машинних циклах):

*movlw N* ; *T*1=1

*movwf cnt\_cycle* ; *T*2=1

*Start\_cycle*: ; Мітка початку циклу

; \*\*\*\*\*\*\*\*\*\*\*\* ; Тіло циклу (*T*т залежить від команд у циклі)

*decfsz cnt\_cycle,f* ; *T*31=1 (зациклення) або *T*32=2 (вихід з циклу)

*goto Start\_cycle* ; *T*41=2 (зациклення) або *T*42=0 (вихід з циклу)

Ініціалізація лічильника циклу виконується одноразово, а тіло циклу

та команди зациклення виконуються кожен раз при проході циклу. Час виконання циклу: *T*=*T*1+*T*2+(*T*т+*T*31+*T*41)∙(*N*-1)+*T*т+*T*32+*T*42=1+(*T*т+3)∙*N*. Статична індикація є найбільш простим засобом передачі даних до зовнішнього спостерігача. При статичній індикації кожен елемент зображення має індивідуальну лінію управління. Тому зміна сигналів управління відбувається тільки при зміні зображення. Такий підхід не потребує складних алгоритмічних дій та звільняє мікроконтролер для виконання інших завдань. Але при статичної індикації икористовується значна кількість виводів мікроконтролеру, що у багатьох ипадках може бути неприпустимо.

**Код програми:**

processor 16F628

#include P16f628.inc

radix dec

N1 equ 0

N2 equ 0

CNT1 equ 0x20

CNT2 equ 0x21

MODE equ 0x22

TIMER equ 0x23

org 0x00

goto main

org 0x04

goto services

main

bsf STATUS,RP0

bcf STATUS,RP1

clrf TRISB

bcf STATUS,RP0

movlw b'10011101'

movwf PORTB

movlw 1

movwf MODE

movlw 20

movwf TIMER

GLOBAL\_LOOP

clrwdt

call DELAY

movlw 1

subwf MODE,w

btfsc STATUS,Z

call SHIFT\_LEFT

movlw 2

subwf MODE,w

btfsc STATUS,Z

comf PORTB,f

movlw 3

subwf MODE,w

btfsc STATUS,Z

call SHIFT\_RIGHT

movlw 4

subwf MODE,w

btfsc STATUS,Z

comf PORTB,f

decfsz TIMER,f

goto GLOBAL\_LOOP

incf MODE,f

movlw 5

subwf MODE,w

btfsc STATUS,Z

call SET\_MODE1

movlw 1

subwf MODE,w

btfsc STATUS,Z

call TIME1

movlw 2

subwf MODE,w

btfsc STATUS,Z

call TIME2

movlw 3

subwf MODE,w

btfsc STATUS,Z

call TIME3

movlw 4

subwf MODE,w

btfsc STATUS,Z

call TIME1

goto GLOBAL\_LOOP

SET\_MODE1

movlw 1

movwf MODE

return

TIME1

movlw 20

movwf TIMER

return

TIME2

movlw 10

movwf TIMER

return

TIME3

movlw 30

movwf TIMER

return

SHIFT\_LEFT

rlf PORTB,w

rlf PORTB,f

return

SHIFT\_RIGHT

rrf PORTB,w

rrf PORTB,f

return

DELAY

movlw N1

movwf CNT1

START\_1

movlw N2

movwf CNT2

START\_2

decfsz CNT2,f

goto START\_2

decfsz CNT1,f

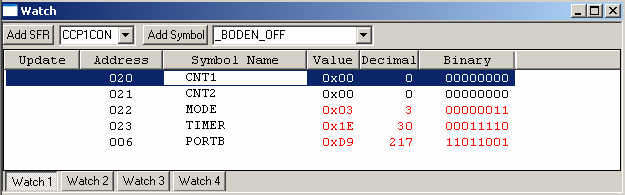
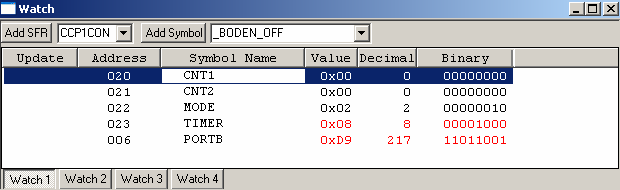
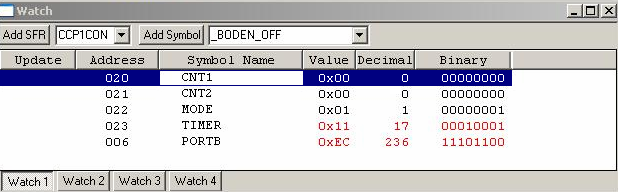
goto START\_1

return

services

retfie

end



**Висновок:** в даній лабораторній роботі ми розглянули роботу з портами введення/виведення; Побудували програмні затримки часу без використання переривань; засвоїли принцип статичної індикації.