**T62 Tutorial 8**

PORTS B and D are connected to the 4 x 7-segment LED.

Write a program to perform the following tasks:

1. Display “0”, “1”, “2”, “3” on the first 7-segment LED
2. Repeat some times
3. Display “4”, “5”, “6”, “7” on the second 7-segment LED
4. Repeat some times
5. Display “8”, “9”, “A”, “b” on the third 7-segment LED
6. Repeat some times
7. Display “C”, “d”, “E”, “F” on the fourth 7-segment LED
8. Repeat some times
9. Goto 1

Store the 7-segment LED decoder table in a look-up table. Your program must use table processing. Copy the program from the editor window.

(12 marks)

LIST P=18F4520 ;directive to define processor

#include <P18F4520.INC> ;CPU specific variable definitions

CONFIG OSC = XT

CONFIG WDT = OFF

CONFIG LVP = OFF

ORG 0x0000

movlw 0x0f

movwf ADCON1

clrf TRISB ;Set Port B direction output

clrf TRISD ;Set Port D direction output

clrf PORTB

clrf PORTD

CBLOCK 0x20

DELAY\_REG\_inner

DELAY\_REG\_mid

DELAY\_REG\_outer

input\_times ;repeat times for each set of number

output\_4times ;4 output display -> loop 4 times for each LED

output\_placement\_ini ;inital place for the LED

input\_reg ;input reg for the table

ENDC

movlw 0x00

movwf output\_placement\_ini

goto Start

ORG 0x00F0

Start: movlw 0x04

movwf output\_4times

call Big\_loop

goto Start

Big\_loop: call repeat\_times

inner\_loop:

movff output\_placement\_ini, PORTB

movf output\_placement\_ini, w

mullw 0x04

movff PRODL, input\_reg

call table\_1

movwf PORTD

call Delay

incf input\_reg, f

call table\_1

movwf PORTD

call Delay

incf input\_reg, f

call table\_1

movwf PORTD

call Delay

incf input\_reg, f

call table\_1

movwf PORTD

call Delay

decf input\_times, f

bnz inner\_loop

incf output\_placement\_ini, f

decf output\_4times, f

bnz Big\_loop

return

;repeat times for the display: set to 2

repeat\_times: movlw 0x03

movwf input\_times

return

;delay function with 200ms

Delay: movlw d'2'

movwf DELAY\_REG\_outer

Delay\_2: movlw d'250'

movwf DELAY\_REG\_mid

Delay\_3: movlw d'50'

movwf DELAY\_REG\_inner

Delay\_4: nop

nop

decf DELAY\_REG\_inner, f

bnz Delay\_4

decf DELAY\_REG\_mid, f

bnz Delay\_3

decf DELAY\_REG\_outer, f

bnz Delay\_2

return

;look-up table function

table\_1: movlw low table\_1\_data

movwf TBLPTRL

movlw high table\_1\_data

movwf TBLPTRH

movlw upper table\_1\_data

movwf TBLPTRU

movf input\_reg, w

mullw 0x02

movf PRODL, w

addwf TBLPTRL

movlw 0x00

addwfc TBLPTRH

addwfc TBLPTRU

TBLRD\*

movf TABLAT, w

return

ORG 0x0030

table\_1\_data

db 0x3f ;0

db 0x06

db 0x5b

db 0x4f

db 0x66 ;4

db 0x6d

db 0x7d

db 0x07

db 0x7f ;8

db 0x6f

db 0x77

db 0x7c

db 0x39 ;12

db 0x5e

db 0x79

db 0x71

END

**Submission**

Enter your name, student ID number, and the answers in the MS Word document file. Re-name the file with your student ID number, e.g. 12345678.docx. **Deduct 4 marks for wrong file name.**

Submit the file by e-mail ([itklchan@cityu.edu.hk](mailto:itklchan@cityu.edu.hk)) before 3:00 pm. **Late submission will not be accepted.**

In case word does not show what I typed:

LIST P=18F4520 ;directive to define processor

#include <P18F4520.INC> ;CPU specific variable definitions

CONFIG OSC = XT

CONFIG WDT = OFF

CONFIG LVP = OFF

ORG 0x0000

movlw 0x0f

movwf ADCON1

clrf TRISB ;Set Port B direction output

clrf TRISD ;Set Port D direction output

clrf PORTB

clrf PORTD

CBLOCK 0x20

    DELAY\_REG\_inner

    DELAY\_REG\_mid

    DELAY\_REG\_outer

    input\_times             ;repeat times for each set of number

    output\_4times           ;4 output display -> loop 4 times for each LED

    output\_placement\_ini    ;inital place for the LED

    input\_reg               ;input reg for the table

ENDC

movlw 0x00

movwf output\_placement\_ini

goto Start

ORG 0x00F0

Start:  movlw 0x04

        movwf output\_4times

        call Big\_loop

        goto Start

Big\_loop:   call repeat\_times

            inner\_loop:

                    movff output\_placement\_ini, PORTB

                    movf output\_placement\_ini, w

                    mullw 0x04

                    movff PRODL, input\_reg

                    call table\_1

                    movwf PORTD

                    call Delay

                    incf input\_reg, f

                    call table\_1

                    movwf PORTD

                    call Delay

                    incf input\_reg, f

                    call table\_1

                    movwf PORTD

                    call Delay

                    incf input\_reg, f

                    call table\_1

                    movwf PORTD

                    call Delay

                    decf input\_times, f

                    bnz inner\_loop

            incf output\_placement\_ini, f

            decf output\_4times, f

            bnz Big\_loop

return

;repeat times for the display: set to 2

repeat\_times:   movlw 0x03

                movwf input\_times

return

;delay function with 200ms

Delay:      movlw d'2'

            movwf DELAY\_REG\_outer

Delay\_2:    movlw d'250'

            movwf DELAY\_REG\_mid

Delay\_3:    movlw d'50'

            movwf DELAY\_REG\_inner

Delay\_4:    nop

            nop

            decf DELAY\_REG\_inner, f

            bnz Delay\_4

            decf DELAY\_REG\_mid, f

            bnz Delay\_3

            decf DELAY\_REG\_outer, f

            bnz Delay\_2

return

;look-up table function

table\_1:    movlw low table\_1\_data

            movwf TBLPTRL

            movlw high table\_1\_data

            movwf TBLPTRH

            movlw upper table\_1\_data

            movwf TBLPTRU

            movf input\_reg, w

            mullw 0x02

            movf PRODL, w

            addwf TBLPTRL

            movlw 0x00

            addwfc TBLPTRH

            addwfc TBLPTRU

            TBLRD\*

            movf TABLAT, w

return

ORG 0x0030

table\_1\_data

    db 0x3f ;0

    db 0x06

    db 0x5b

    db 0x4f

    db 0x66 ;4

    db 0x6d

    db 0x7d

    db 0x07

    db 0x7f ;8

    db 0x6f

    db 0x77

    db 0x7c

    db 0x39 ;12

    db 0x5e

    db 0x79

    db 0x71

END