

## T62 Tutorial 5

1. Change the highlighted instruction “**movlw 0x0F**” to “movlw 0xFF”. Build the machine code and then upload the program to the development kit again. What is your observation? Discuss the relationship between the modified instruction and your observation.

(4 marks)

The count up frequency or blinking frequency decreased.

The larger the value in the movlw instruction, the lower the LEDs blinking frequency.

2. Click the “Step Over” button a number of times. What are your observations on LEDs and the Watch window?

(4 marks)

- i. Every time the “Step Over” button is clicked, the PICKIT3 instructs the PIC18 to run one instruction (PCL is updated) and then halt. If the instruction is function call, the function will be executed and the result returned without debugging each line.
- ii. Every time when the incf location0 is executed, the value of location0 is incremented by 1.
- iii. The value written to PORTD is displayed on LEDs and LATD.
- iv. The values of LATD and PORTD are difference.

3. Discover the functions/effect of “Animate” and “Halt” buttons. Notice the change of green arrow location in source file window and the 8 LEDs blinking status on development kit. What is the function of the “Animate” button? What is the function of the “Halt” button?

(4 marks)

The “Animate” button causes the debugger to automatically execute the instruction one-by one.

The “Halt” button stops the execution of program code.

4. Write a program to display the last four digits of your student ID number in binary representation on the four rightmost LEDs continuously. Copy the program from the editor window.

(4 marks)

```
LIST    P=18F4520        ; directive to define processor
        #include <P18F4520.INC> ; CPU specific variable definitions
;-----

;Variable definitions
;
        CBLOCK    0x000
        location0
        DELAY_H
        DELAY_L
        ENDC
;-----
;Reset vector
;Student ID number is 12345678
        ORG 0x0000    ; code origin, program starts from here
        goto    Main
; Start of main program
Main:    clrf    TRISD    ; set Port D direction "output"
        clrf    PORTD
        clrf    location0
MainLoop:
        movlw    0x05
        movwf    PORTD
        call    Delay
        movlw    0x06
        movwf    PORTD
        call    Delay
        movlw    0x07
        movwf    PORTD
        call    Delay
        movlw    0x08
        movwf    PORTD
        call    Delay
        bra     MainLoop
;-----
Delay:   movlw    0x02
        movwf    DELAY_H
LOP_1:   movlw    0x02
        movwf    DELAY_L
LOP_2:   decf    DELAY_L, F
        bnz     LOP_2
        decf    DELAY_H, F
        bnz     LOP_1
        return
;
        END        ; End of program
```

5. Complete the 7-segment LED decoder table.

(2 marks)

Digit/Letter	dp	g	f	e	d	c	b	a	Hex
	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
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	1	1	1	1	0x6F
A	0	1	1	1	0	1	1	1	0x77
b	0	1	1	1	1	1	0	0	0x7C
C	0	0	1	1	1	0	0	1	0x39
d	0	1	0	1	1	1	1	0	0x5E
E	0	1	1	1	1	0	0	1	0x79
F	0	1	1	1	0	0	0	1	0x71

6. Write a program to display all digits of your student ID number (from the first to the last digit) on the 7-segment LED continuously. Each digit should hold for a certain delay time. Copy the program from the editor window.

(6 marks)

```

LIST      P=18F4520          ; directive to define processor
          #include <P18F4520.INC> ; CPU specific variable definitions
;-----

          CBLOCK 0x000
          DELAY_U
          DELAY_H
          DELAY_L
          ENDC

;Reset vector
;Student ID number is 12345678
          ORG 0x0000 ; code origin, program starts from here
          goto Main

          ORG 0x0100
Main:      clrfsd    TRISC    ; set Port C direction "output"
          clrfsd    PORTC

Loop:
          movlw     0x06
          movwf     PORTC
          call      Delay
          movlw     0x5B
          movwf     PORTC
          call      Delay
          movlw     0x4F
          movwf     PORTC
          call      Delay
          movlw     0x66
          movwf     PORTC
          call      Delay
          movlw     0x6D
          movwf     PORTC
          call      Delay
          movlw     0x7D
          movwf     PORTC
          call      Delay
          movlw     0x07
          movwf     PORTC
          call      Delay
          movlw     0x7F
          movwf     PORTC
          call      Delay
          goto      Loop

Delay:     movlw     d'10'
          movwf     DELAY_U
LOP_0:     movlw     0x80
          movwf     DELAY_H
LOP_1:     movlw     0xFF
          movwf     DELAY_L
LOP_2:     decfsd    DELAY_L, F
          bnz       LOP_2
          decfsd    DELAY_H, F
          bnz       LOP_1
          decfsd    DELAY_U, F
          bnz       LOP_0
          return

          END          ; End of program

```