EE 215 Microprocessors LAB #4

Student name：Li Xianzhe
Student ID number：2022214880

Explanation of Approach

QUESTIONS:

1. What microcontroller GPIO pins are LED1 and LED2 connected to?
2. Explain what port control registers commands are needed for the LEDs to blink.
3. What is a watchdog timer? Why is it turned off for this application?

ANSWERS:

1.LED1 is connected to P8.1 and LED2 is connected to P3.7
2.
(1)PxDIR|=BITx; The purpose of this command is to put Px.x into input mode
(2)PxOUT&=~BITx; The purpose of this command is to make the Px.x output level low (to initialize the LEDs connected to the Px.x to turn off)
(3)PxOUT|=BITx; The purpose of this command is to make the Px.x output high (light up the LED connected to the Px.x)
(4)PxOUT^=BITx; What this command does is make the Px.x output a level opposite to the previous state (off if on, or vice versa)
3.
The main purpose of a watchdog timer is to protect the system from software failures, such as unexpected infinite loops.
By default, the watchdog for the MSP430 is turned on. If the program is not closed at the beginning, it will automatically reset when the program is executed for a certain period of time, so the program cannot be executed normally. So it needs to be turned off.

Code

```
#include <msp430.h>
#define CPU_F ((double)1000000)
#define delay_ms(x) __delay_cycles((long)(CPU_F*(double)x/1000.0))

int main(void)
{
    WDTCTL = WDTPW | WDTHOLD;

    P8DIR|=BIT1;
    P3DIR|=BIT7;
    P7DIR|=BIT4;
    P6DIR|=BIT3;
    P6DIR|=BIT4;
    P3DIR|=BIT5; //These lines of code are used to set the IO port to
output mode


    P8OUT&=~BIT1;
    P3OUT&=~BIT7;
    P7OUT&=~BIT4;
    P6OUT&=~BIT3;
    P6OUT&=~BIT4;
    P3OUT&=~BIT5; //Initialize the six lights of the MSP430 to turn it off


    while(1)//Cycle to turn on/off 6 leds
    {

           P8OUT^=BIT1;//turn on led
           delay_ms(800);//used to delay
           P8OUT^=BIT1;//turn off led

           P3OUT^=BIT7;
           delay_ms(800);
           P3OUT^=BIT7;

           P7OUT^=BIT4;
           delay_ms(800);
           P7OUT^=BIT4;

           P6OUT^=BIT3;
           delay_ms(800);
           P6OUT^=BIT3;

           P6OUT^=BIT4;
           delay_ms(800);
           P6OUT^=BIT4;

           P3OUT^=BIT5;
           delay_ms(800);
           P3OUT^=BIT5;

    }

    return 0;
}
```