EEE204 - Introduction to Embedded Systems

Experiment 6

Objectives:

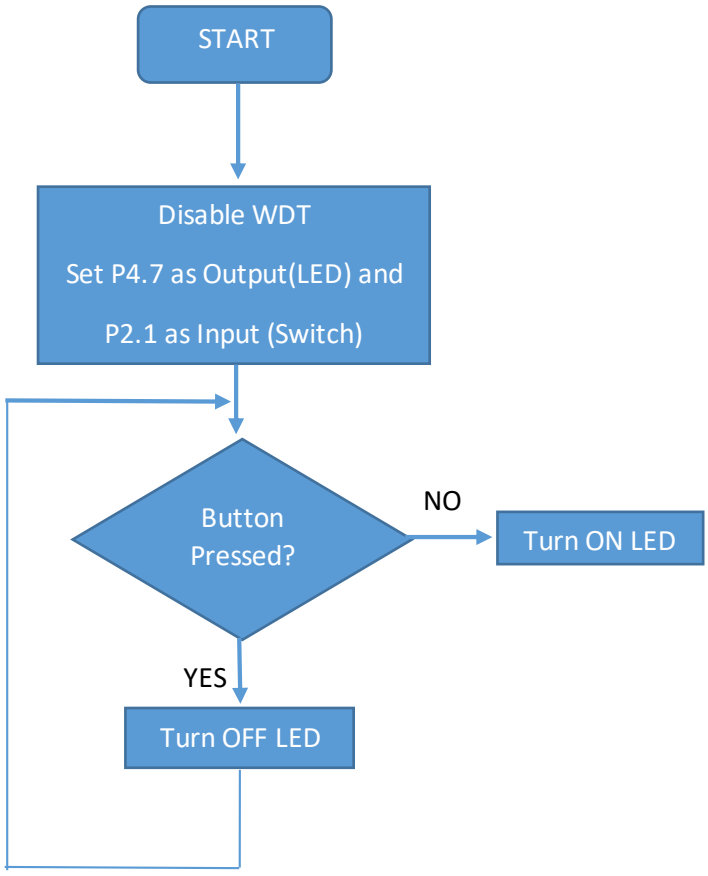
The MSP-EXP430F5529LP LaunchPad has two general-purpose digital I/O pins connected to red (P1.0) and green (P4.7) LEDs for visual feedback. It has also two push buttons connected to pins P1.1 and P2.1 on the MSP-EXP430F5529LP LaunchPad for user feedback. The main objective of this experiment is to control the on-board LEDs of the MSP-EXP430F5529LP LaunchPad by an input from the on-board switch. This experiment will help you to learn and understand the procedure for programming the MSP430F5529 GPIO pins as input and output.

Materials

- Code Composer Studio IDE
- MSP-EXP430F5529LP USB LaunchPad development kit

Experimental Work

E1 In this experiment, the input from the left push button switch (S1) connected to Port P2.1 is read by the processor for LED control. If the switch is pressed the green LED is turned OFF by output at port P4.7 reset to '0'. Else, the output at port P4.7 is set HIGH and the green LED is turned ON. The LED is therefore controlled by the switch S1. **Write an Assembly language program that performs all the given operations above.**



Flowchart for Controlling LED with a Switch

```
init:
                bis.b       #BIT7, &P4DIR
                bic.b       #BIT7, &P4OUT

                bic.b       #BIT1, &P2DIR
                bis.b       #BIT1, &P2REN
                bis.b       #BIT1, &P2OUT

main:
                bit.b       #BIT1, &P2IN
                jnz         ButtonUp
ButtonDown:
                bic.b       #BIT7, P4OUT
                jmp         main
ButtonUp:
                bis.b       #BIT7, &P4OUT
                jmp         main
```

E2 Write an Assembbly language program that realizes all given below.

a) Alter the code in E1 to turn the LED ON when the button is pressed and OFF when it is released.

```
init:
                bis.b       #BIT7, &P4DIR
                bic.b       #BIT7, &P4OUT

                bic.b       #BIT1, &P2DIR
                bis.b       #BIT1, &P2REN
                bis.b       #BIT1, &P2OUT

main:
                bit.b       #BIT1, &P2IN
                jnz         ButtonUp
ButtonDown:
                bis.b       #BIT7, P4OUT
                jmp         main
ButtonUp:
                bic.b       #BIT7, &P4OUT
                jmp         main
```

b) Alter the code to make the green LED stay ON for a while every time the button is pressed.

```
init:
                bis.b       #BIT7, &P4DIR
                bic.b       #BIT7, &P4OUT


                bic.b       #BIT1, &P2DIR
                bis.b       #BIT1, &P2REN
```

```
                bis.b      #BIT1, &P2OUT

main:
                bit.b      #BIT1, &P2IN
                jnz        ButtonUp
ButtonDown:
                bis.b      #BIT7, P4OUT
                mov.w      #50000, R4
delay1:
                mov.w      #5,R5
delay2:
                dec.w      R5
                jnz        delay2
                dec.w      R4
                jnz        delay1
                jmp        main
ButtonUp:
                bic.b #BIT7, &P4OUT
                jmp   main
```

c) Alter the code to turn the red LED ON when the button is pressed and the green LED ON when the button is released

```
;-------------------------------------------------------------------------------
; Main loop here
;-------------------------------------------------------------------------------
init:
                bis.b #BIT7, &P4DIR
                bic.b #BIT7, &P4OUT
                bis.b #BIT0, &P1DIR
                bic.b #BIT0, &P1OUT

                bic.b #BIT1, &P2DIR
                bis.b #BIT1, &P2REN
                bis.b #BIT1, &P2OUT

main:
                bit.b #BIT1, &P2IN
                jnz        ButtonUp
ButtonDown:
                bis.b #BIT0, P1OUT
                bic.b #BIT7, P4OUT
                jmp        main
ButtonUp:
                bic.b #BIT0, &P1OUT
                bis.b #BIT7, &P4OUT
                jmp   main
```