Using the attached reference specifically looking at program 4.2 and all of chapter 4 on assembly code for help create a solution to the following problem.

This code currently has the two blinking LEDs working now all I need is for the LEDs to only blink while a button is being pressed. If the button is not actively being pressed, then the LEDs should be off. The button is connected to P6.0 on the board.

Using a MSP430FR5994 Microcontroller

This code currently works to blink LED A which is connected to P3.0 but I want it to also blink LED B which is connected to P3.1. Please make the changes to do this.

LED A is connected to P3.0 on the board

LED B is connected to P3.1 on the board

The button is connected to P6.0 on the board

For System B, students must write an assembly code for the following functions.

Blink the LED\_A and LED\_B on the BH board simultaneously while the button is pressed. If the button is released, blinking should stop.

The blink function should keep repeating indefinitely.

The blinking duration must be between 0.5 seconds to 1 second.

;-------------------------------------------------------------------------------

; MSP430 Assembler Code Template for use with TI Code Composer Studio

;

;

;-------------------------------------------------------------------------------

.cdecls C,LIST,"msp430.h" ; Include device header file

;-------------------------------------------------------------------------------

.def RESET ; Export program entry-point to

; make it known to linker.

;-------------------------------------------------------------------------------

**.text** ; Assemble into program memory.

.retain ; Override ELF conditional linking

; and retain current section.

.retainrefs ; And retain any sections that have

; references to current section.

;-------------------------------------------------------------------------------

RESET **mov.w** #\_\_STACK\_END,SP ; Initialize stackpointer

StopWDT **mov.w** #WDTPW|WDTHOLD,&WDTCTL ; Stop watchdog timer

;-------------------------------------------------------------------------------

; Main loop here

;-------------------------------------------------------------------------------

**bic.b** #LOCKLPM5, &PM5CTL0 ; clear LOCKLPM5 bit

**bis.b** #0x01, &P3DIR; Set P3.0 as output

**bis.b** #0x01, &P3OUT; Set P3.0 high (LED ON)

**\_loop:**

**xor.b** #0x01, &P3OUT; Toggle P3.0

**mov.w** #0xF400, R10; R10<-0xF400

**\_lp1:**

**dec.w** R10 ; Decrement R10

**cmp.w** #0x00, R10 ; Compare with 0

**jne** \_lp1 ; Conditional branch

**jmp** \_loop; Branch back to loop

**nop**

;-------------------------------------------------------------------------------

; Stack Pointer definition

;-------------------------------------------------------------------------------

**.global** \_\_STACK\_END

**.sect** .stack

;-------------------------------------------------------------------------------

; Interrupt Vectors

;-------------------------------------------------------------------------------

**.sect** ".reset" ; MSP430 RESET Vector

**.short** RESET

;-------------------------------------------------------------------------------

; MSP430 Assembler Code Template for use with TI Code Composer Studio

;

;

;-------------------------------------------------------------------------------

.cdecls C,LIST,"msp430.h" ; Include device header file

;-------------------------------------------------------------------------------

.def RESET ; Export program entry-point to

; make it known to linker.

;-------------------------------------------------------------------------------

**.text** ; Assemble into program memory.

.retain ; Override ELF conditional linking

; and retain current section.

.retainrefs ; And retain any sections that have

; references to current section.

;-------------------------------------------------------------------------------

RESET **mov.w** #\_\_STACK\_END,SP ; Initialize stackpointer

StopWDT **mov.w** #WDTPW|WDTHOLD,&WDTCTL ; Stop watchdog timer

;-------------------------------------------------------------------------------

; Main loop here

;-------------------------------------------------------------------------------

**bic.b** #LOCKLPM5, &PM5CTL0 ; Clear LOCKLPM5 bit

**bis.b** #0x03, &P3DIR ; Set P3.0 and P3.1 as output

**bis.b** #0x03, &P3OUT ; Set P3.0 and P3.1 high (LEDs ON)

**\_loop:**

**xor.b** #0x03, &P3OUT ; Toggle P3.0 and P3.1

**mov.w** #0xF400, R10 ; R10 <- 0xF400

**\_lp1:**

**dec.w** R10 ; Decrement R10

**cmp.w** #0x00, R10 ; Compare with 0

**jne** \_lp1 ; Conditional branch

**jmp** \_loop ; Branch back to loop

**nop**

;-------------------------------------------------------------------------------

; Stack Pointer definition

;-------------------------------------------------------------------------------

**.global** \_\_STACK\_END

**.sect** .stack

;-------------------------------------------------------------------------------

; Interrupt Vectors

;-------------------------------------------------------------------------------

**.sect** ".reset" ; MSP430 RESET Vector

**.short** RESET