Delay Subroutine using Timer

Lab Goal

Understand the effect and use of subroutines with parameters.
Observe the use of a timer peripheral and an Interrupt Service Routine (ISR)

Instructions/Questions:

1) Open Workspace and Project Lab Timer A. Connect the FET kit to the computer. Edit the asm.s43 file for this project.

2) The code is already set up for the following:

- Peripheral Timer0_A3 will increment the value in TA0R at a rate of 500kHz
  - Use &TA0R to read and write to this register
- P1OUT is setup to allow P1.0 to be used to drive the LED
- The main loop is coded to do the main activity, toggle the LED, call a delay function and repeat forever.
- The subroutine "delay_usx2" label and return instruction have been coded

3) You must finish the "delay_usx2" subroutine so that it does not return to the main program until the delay has "expired". It must do the following:

1. Accept a "parameter" via R12 that tells the subroutine how long to delay
2. Stay in the subroutine for the appropriate amount of time with a range of 0 to 100ms

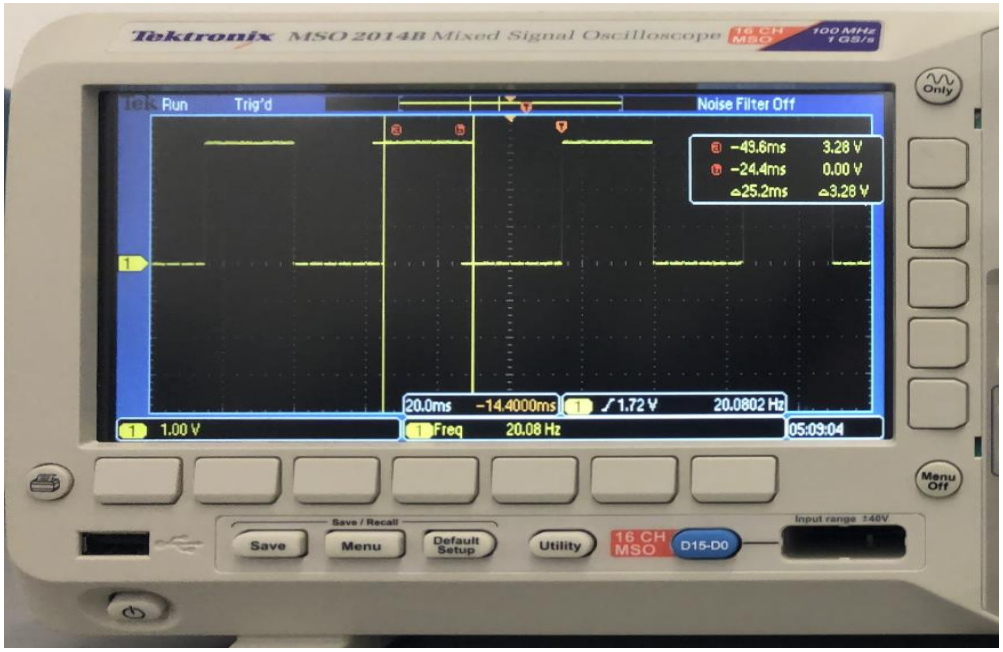
4) Show your results and code

Test your code using the development board and measuring the LEDs blinking rate using the Oscilloscope/Logic Analyzer.

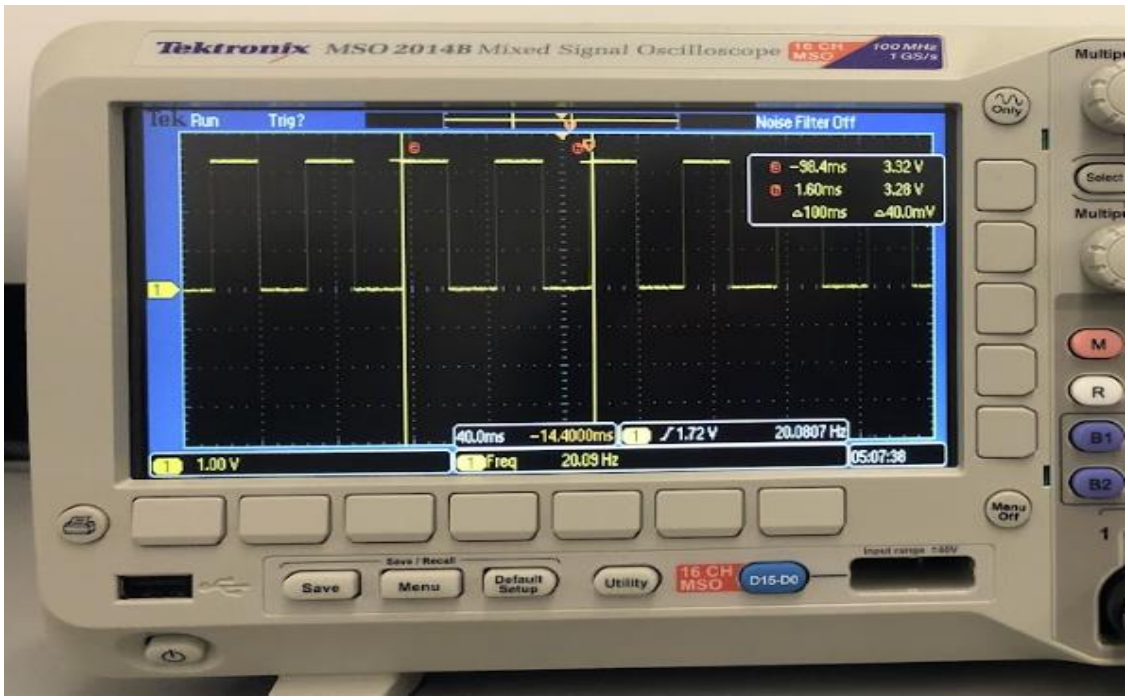
Show a screen shot of the waveform for a 25ms delay, this should look like a 50ms period.

Show a screen shot of the waveform for a 100ms delay, this should look like a 200ms period.

25ms Delay waveform screen shot



100ms Delay waveform screen shot



Paste the code of your subroutine below.

CODE :

```
delay_usx2
        MOV.W #0000h, TA0R
delay_25ms
        CMP #12500, TA0R
        JLO delay_25ms
        ret

delay_usx2
        MOV.W #0000h, TA0R
delay_100ms
        CMP #50000, TA0R
        JLO delay_100ms
        ret
```

asm.s43

```
3             ORG     0F800h                  ; Program Reset
4   ;-------------------------------------------------------------------------
5   RESET       mov.w   #0280h,SP               ; Initialize stackpointer
6   StopWDT     mov.w   #WDTPW+WDTHOLD,&WDTCTL  ; Stop WDT
7   Load        mov.b   &CALBC1_1MHZ,&BCSCTL1   ; Set DCO to 1MHz
8               mov.b   &CALDCO_1MHZ,&DCOCTL
9   SetupP1     bis.b   #001h,&P1DIR            ; P1.0 output
10  SetupTA     mov.w   #TASSEL_2+MC_2+ID_1,&TACTL   ; SMCLK, contmode
11
12  main
13
14              xor.b   #1,&P1OUT            ; Toggle P1.0
15              ;  pass a paramater to delay_us subroutine
16              call    #delay_usx2
17              jmp     main
18
19
20  delay_usx2
21
22              ret
23
24
25  ;-------------------------------------------------------------------------
26  ;           Interrupt Vectors
27  ;-------------------------------------------------------------------------
28              ORG     0FFFEh                  ; MSP430 RESET Vector
29              DW      RESET                   ;
30              END
31
```