Austin Petersen, Ayden Dauenhauer

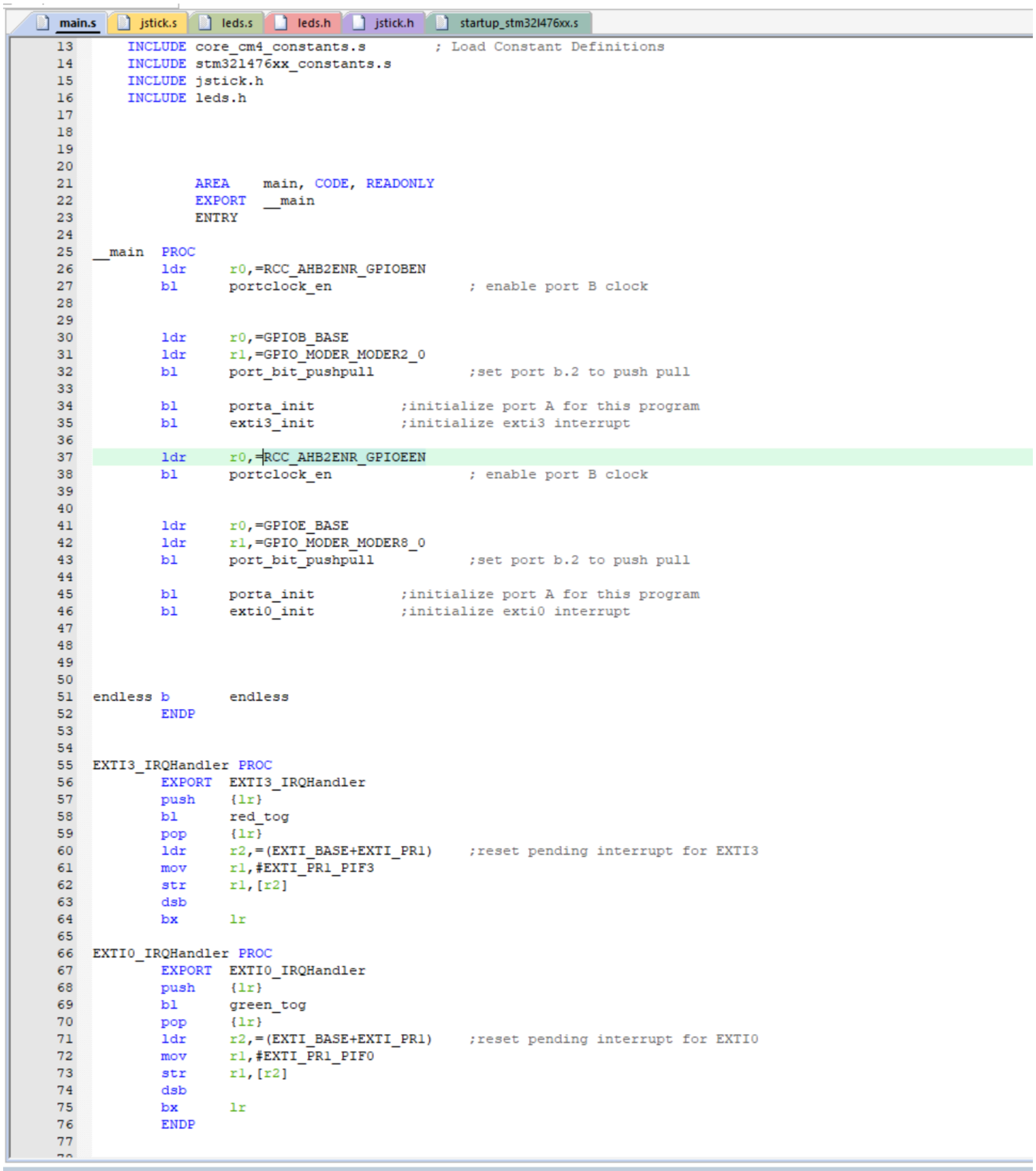
Prof. Wolfe

ELEN 120L Tuesday 2:15-5:00

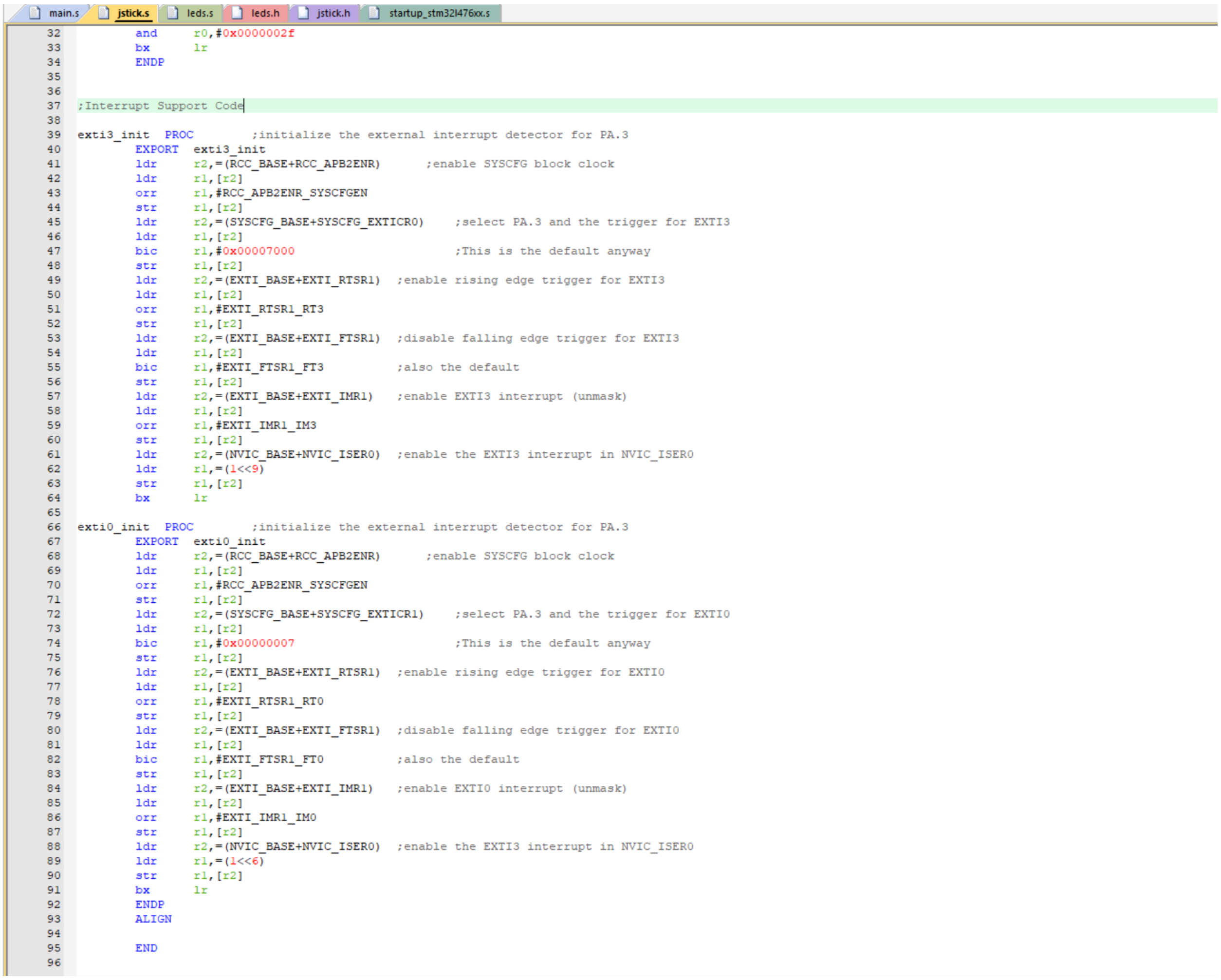
October 31, 2023

Report 5

Problem 1:

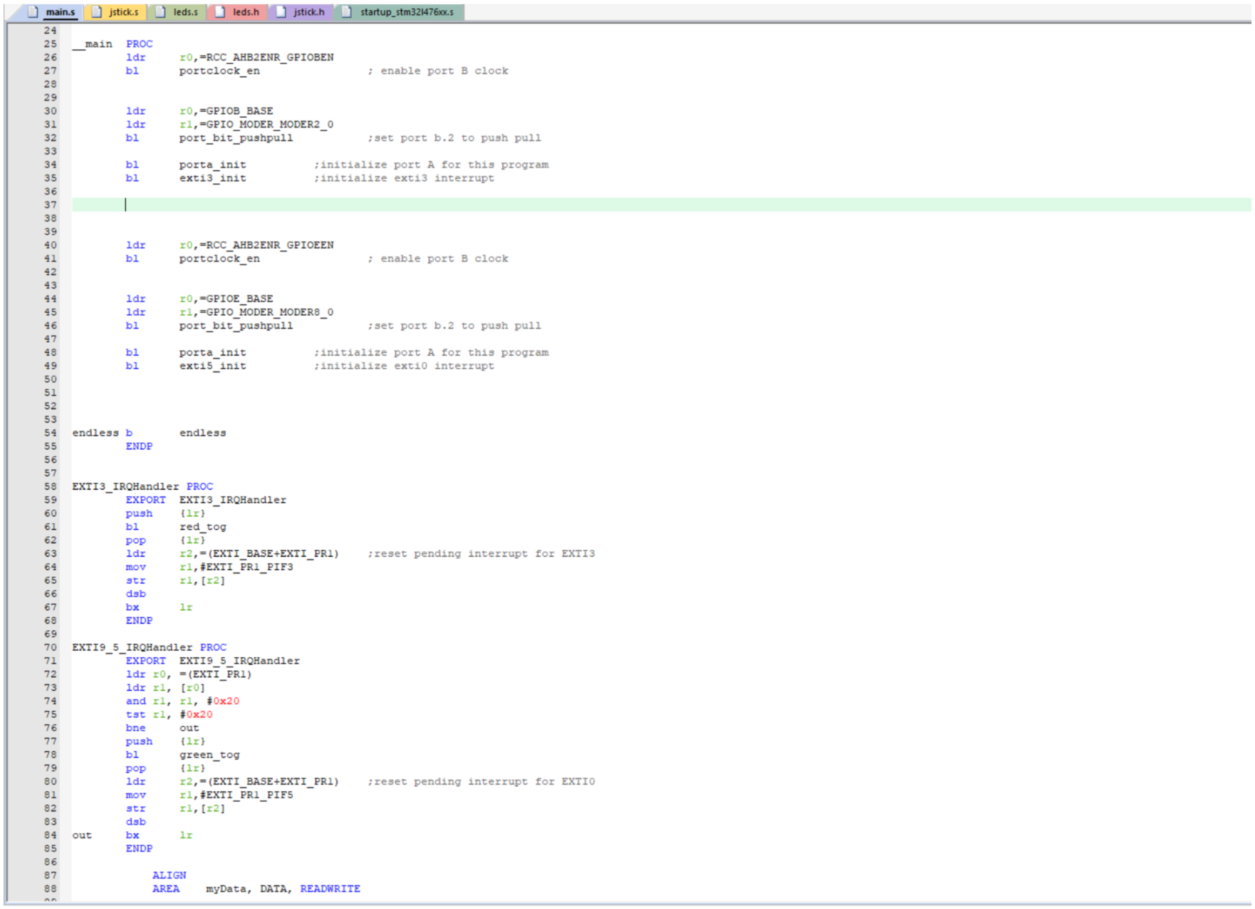


```
    INCLUDE core_cm4_constants.s         ; Load Constant Definitions
    INCLUDE stm321476xx_constants.s
    INCLUDE jstick.h
    INCLUDE leds.h




            AREA    main, CODE, READONLY
            EXPORT  __main
            ENTRY

__main  PROC
        ldr     r0,=RCC_AHB2ENR_GPIOBEN
        bl      portclock_en                ; enable port B clock


        ldr     r0,=GPIOB_BASE
        ldr     r1,=GPIO_MODER_MODER2_0
        bl      port_bit_pushpull           ;set port b.2 to push pull

        bl      porta_init          ;initialize port A for this program
        bl      exti3_init          ;initialize exti3 interrupt

        ldr     r0,=RCC_AHB2ENR_GPIOEEN
        bl      portclock_en                ; enable port B clock


        ldr     r0,=GPIOE_BASE
        ldr     r1,=GPIO_MODER_MODER8_0
        bl      port_bit_pushpull           ;set port b.2 to push pull

        bl      porta_init          ;initialize port A for this program
        bl      exti0_init          ;initialize exti0 interrupt




endless b       endless
        ENDP


EXTI3_IRQHandler PROC
        EXPORT  EXTI3_IRQHandler
        push    {lr}
        bl      red_tog
        pop     {lr}
        ldr     r2,=(EXTI_BASE+EXTI_PR1)    ;reset pending interrupt for EXTI3
        mov     r1,#EXTI_PR1_PIF3
        str     r1,[r2]
        dsb
        bx      lr

EXTI0_IRQHandler PROC
        EXPORT  EXTI0_IRQHandler
        push    {lr}
        bl      green_tog
        pop     {lr}
        ldr     r2,=(EXTI_BASE+EXTI_PR1)    ;reset pending interrupt for EXTI0
        mov     r1,#EXTI_PR1_PIF0
        str     r1,[r2]
        dsb
        bx      lr
        ENDP

```

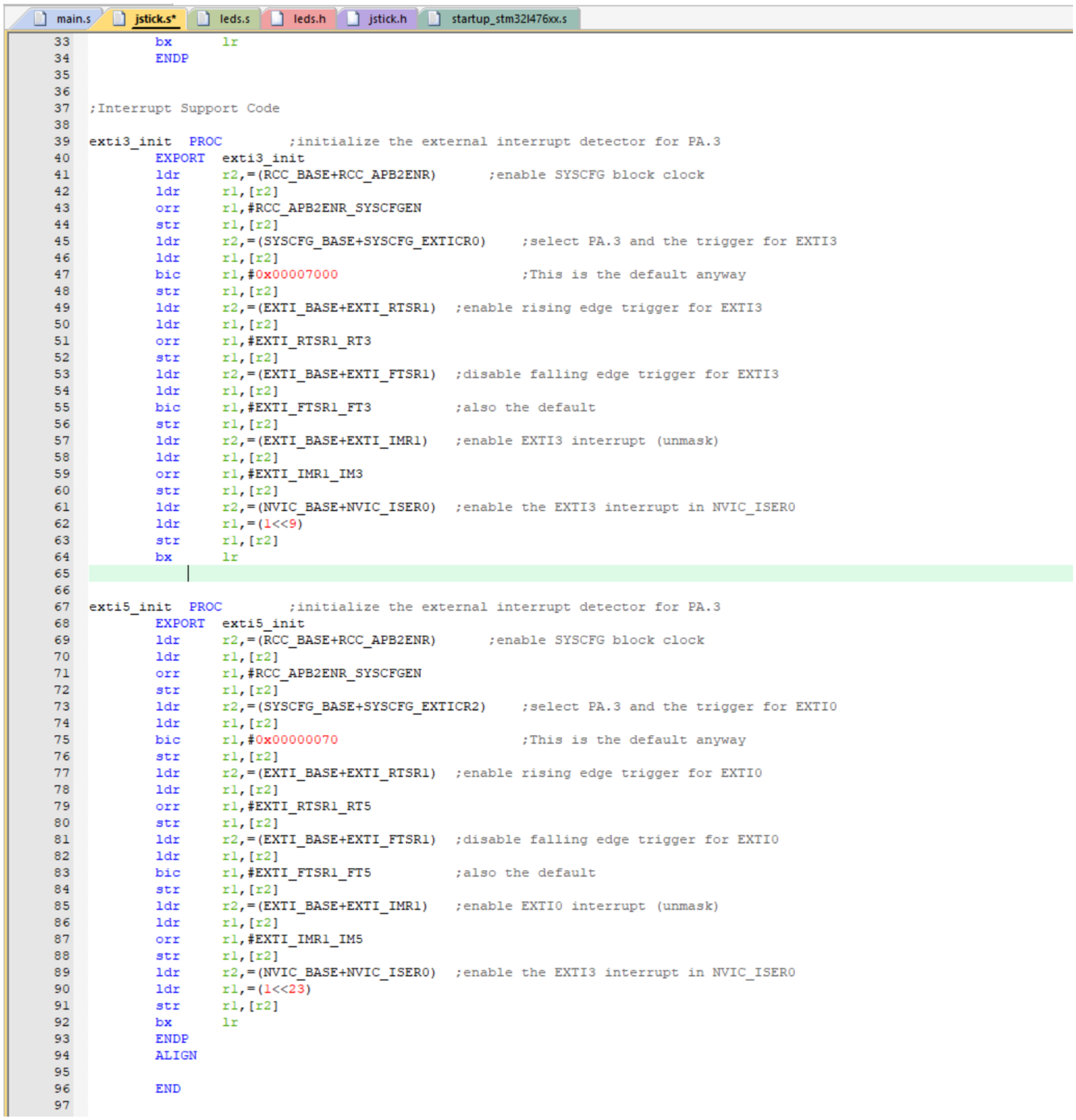


```
        and     r0,#0x0000002f
        bx      lr
        ENDP


;Interrupt Support Code

exti3_init  PROC        ;initialize the external interrupt detector for PA.3
        EXPORT  exti3_init
        ldr     r2,=(RCC_BASE+RCC_APB2ENR)      ;enable SYSCFG block clock
        ldr     r1,[r2]
        orr     r1,#RCC_APB2ENR_SYSCFGEN
        str     r1,[r2]
        ldr     r2,=(SYSCFG_BASE+SYSCFG_EXTICR0)    ;select PA.3 and the trigger for EXTI3
        ldr     r1,[r2]
        bic     r1,#0x00007000                      ;This is the default anyway
        str     r1,[r2]
        ldr     r2,=(EXTI_BASE+EXTI_RTSR1)  ;enable rising edge trigger for EXTI3
        ldr     r1,[r2]
        orr     r1,#EXTI_RTSR1_RT3
        str     r1,[r2]
        ldr     r2,=(EXTI_BASE+EXTI_FTSR1)  ;disable falling edge trigger for EXTI3
        ldr     r1,[r2]
        bic     r1,#EXTI_FTSR1_FT3          ;also the default
        str     r1,[r2]
        ldr     r2,=(EXTI_BASE+EXTI_IMR1)   ;enable EXTI3 interrupt (unmask)
        ldr     r1,[r2]
        orr     r1,#EXTI_IMR1_IM3
        str     r1,[r2]
        ldr     r2,=(NVIC_BASE+NVIC_ISER0)  ;enable the EXTI3 interrupt in NVIC_ISER0
        ldr     r1,=(1<<9)
        str     r1,[r2]
        bx      lr

exti0_init  PROC        ;initialize the external interrupt detector for PA.3
        EXPORT  exti0_init
        ldr     r2,=(RCC_BASE+RCC_APB2ENR)      ;enable SYSCFG block clock
        ldr     r1,[r2]
        orr     r1,#RCC_APB2ENR_SYSCFGEN
        str     r1,[r2]
        ldr     r2,=(SYSCFG_BASE+SYSCFG_EXTICR1)    ;select PA.3 and the trigger for EXTI0
        ldr     r1,[r2]
        bic     r1,#0x00000007                      ;This is the default anyway
        str     r1,[r2]
        ldr     r2,=(EXTI_BASE+EXTI_RTSR1)  ;enable rising edge trigger for EXTI0
        ldr     r1,[r2]
        orr     r1,#EXTI_RTSR1_RT0
        str     r1,[r2]
        ldr     r2,=(EXTI_BASE+EXTI_FTSR1)  ;disable falling edge trigger for EXTI0
        ldr     r1,[r2]
        bic     r1,#EXTI_FTSR1_FT0          ;also the default
        str     r1,[r2]
        ldr     r2,=(EXTI_BASE+EXTI_IMR1)   ;enable EXTI0 interrupt (unmask)
        ldr     r1,[r2]
        orr     r1,#EXTI_IMR1_IM0
        str     r1,[r2]
        ldr     r2,=(NVIC_BASE+NVIC_ISER0)  ;enable the EXTI3 interrupt in NVIC_ISER0
        ldr     r1,=(1<<6)
        str     r1,[r2]
        bx      lr
        ENDP
        ALIGN

        END

```

Problem 2:



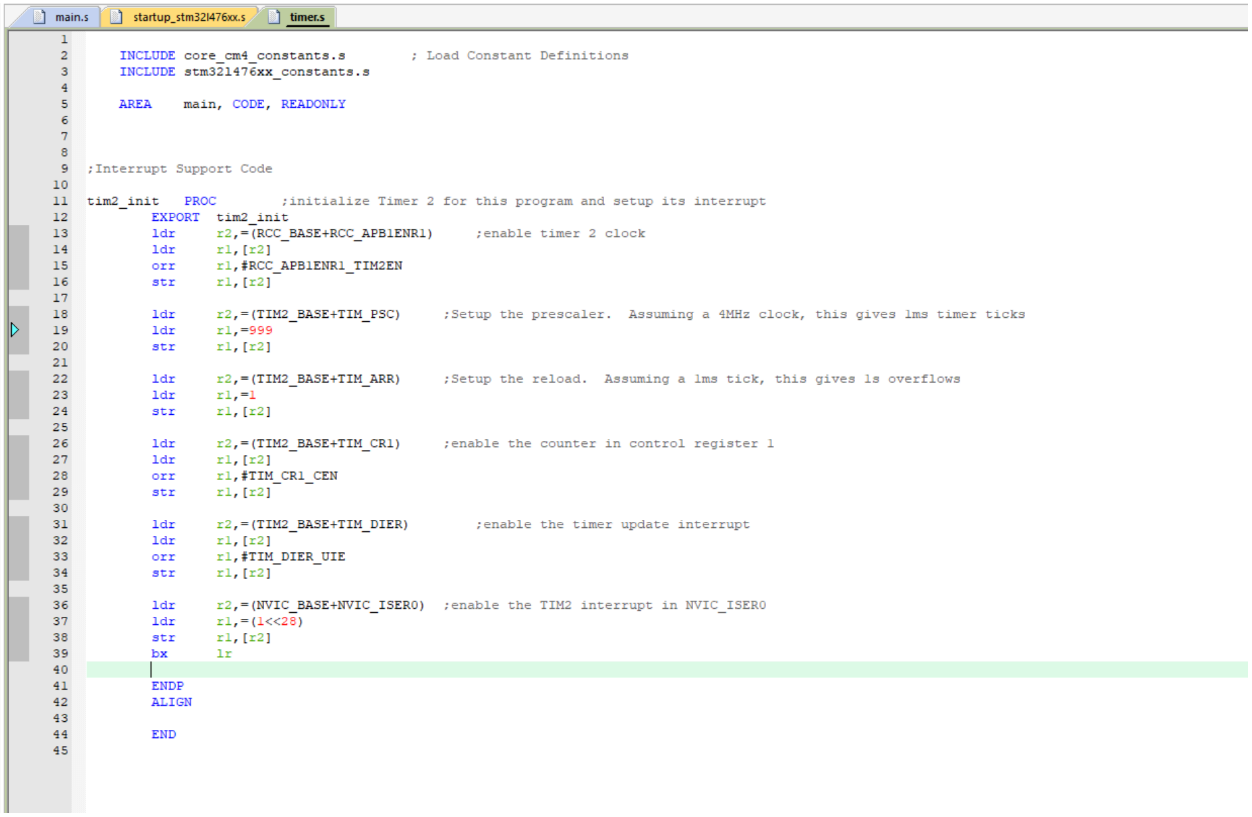
```

__main  PROC
        ldr     r0,=RCC_AHB2ENR_GPIOBEN
        bl      portclock_en                ; enable port B clock


        ldr     r0,=GPIOB_BASE
        ldr     r1,=GPIO_MODER_MODER2_0
        bl      port_bit_pushpull           ;set port b.2 to push pull

        bl      porta_init          ;initialize port A for this program
        bl      exti3_init          ;initialize exti3 interrupt




        ldr     r0,=RCC_AHB2ENR_GPIOEEN
        bl      portclock_en                ; enable port B clock


        ldr     r0,=GPIOE_BASE
        ldr     r1,=GPIO_MODER_MODER8_0
        bl      port_bit_pushpull           ;set port b.2 to push pull

        bl      porta_init          ;initialize port A for this program
        bl      exti5_init          ;initialize exti0 interrupt




endless b       endless
        ENDP


EXTI3_IRQHandler PROC
        EXPORT  EXTI3_IRQHandler
        push    {lr}
        bl      red_tog
        pop     {lr}
        ldr     r2,=(EXTI_BASE+EXTI_PR1)    ;reset pending interrupt for EXTI3
        mov     r1,#EXTI_PR1_PIF3
        str     r1,[r2]
        dsb
        bx      lr
        ENDP

EXTI9_5_IRQHandler PROC
        EXPORT  EXTI9_5_IRQHandler
        ldr r0, =(EXTI_PR1)
        ldr r1, [r0]
        and r1, r1, #0x20
        tst r1, #0x20
        bne     out
        push    {lr}
        bl      green_tog
        pop     {lr}
        ldr     r2,=(EXTI_BASE+EXTI_PR1)    ;reset pending interrupt for EXTI0
        mov     r1,#EXTI_PR1_PIF5
        str     r1,[r2]
        dsb
out     bx      lr
        ENDP

            ALIGN
            AREA    myData, DATA, READWRITE
```



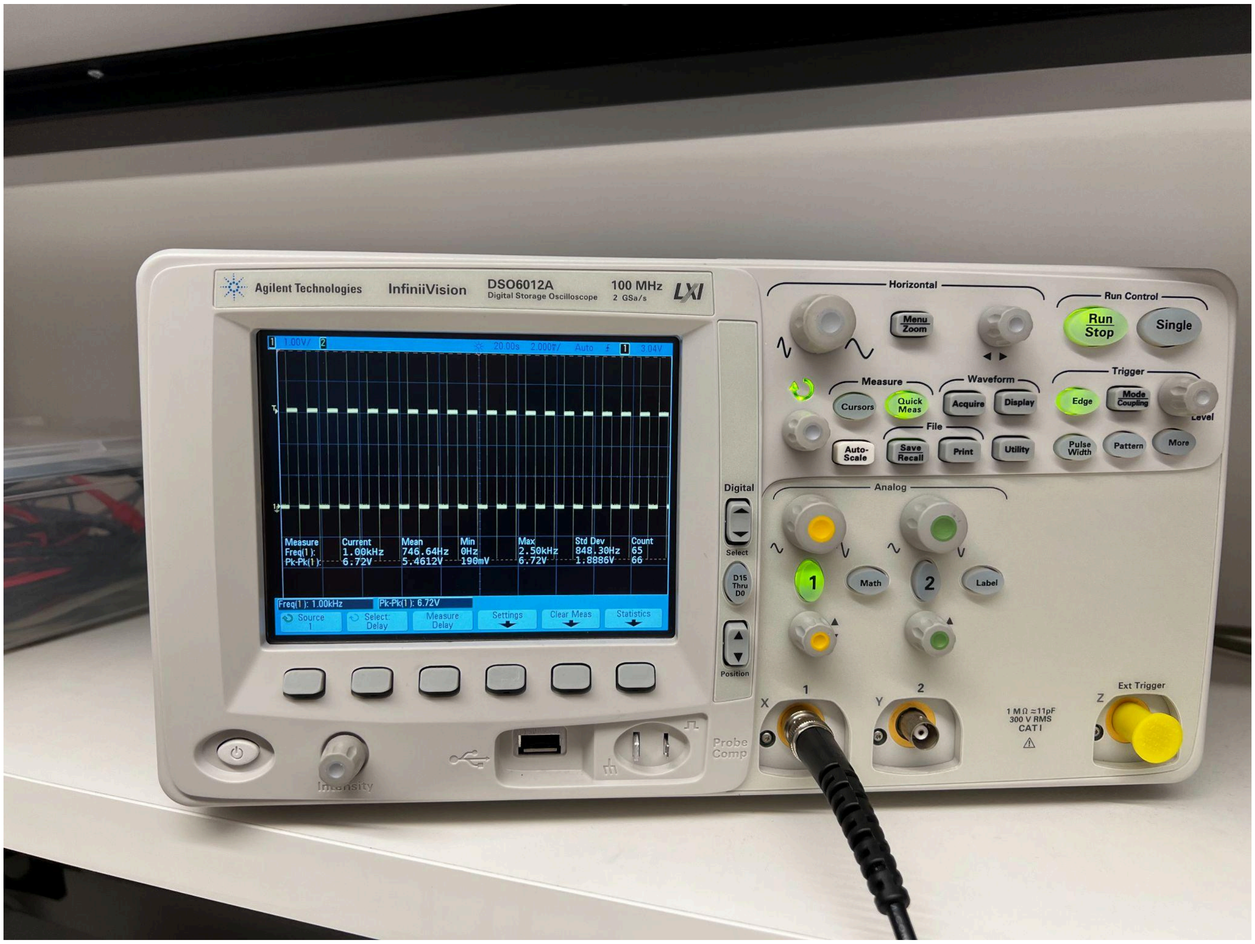
```
        bx      lr
        ENDP


;Interrupt Support Code

exti3_init  PROC        ;initialize the external interrupt detector for PA.3
        EXPORT  exti3_init
        ldr     r2,=(RCC_BASE+RCC_APB2ENR)      ;enable SYSCFG block clock
        ldr     r1,[r2]
        orr     r1,#RCC_APB2ENR_SYSCFGEN
        str     r1,[r2]
        ldr     r2,=(SYSCFG_BASE+SYSCFG_EXTICR0)    ;select PA.3 and the trigger for EXTI3
        ldr     r1,[r2]
        bic     r1,#0x00007000                      ;This is the default anyway
        str     r1,[r2]
        ldr     r2,=(EXTI_BASE+EXTI_RTSR1)  ;enable rising edge trigger for EXTI3
        ldr     r1,[r2]
        orr     r1,#EXTI_RTSR1_RT3
        str     r1,[r2]
        ldr     r2,=(EXTI_BASE+EXTI_FTSR1)  ;disable falling edge trigger for EXTI3
        ldr     r1,[r2]
        bic     r1,#EXTI_FTSR1_FT3          ;also the default
        str     r1,[r2]
        ldr     r2,=(EXTI_BASE+EXTI_IMR1)   ;enable EXTI3 interrupt (unmask)
        ldr     r1,[r2]
        orr     r1,#EXTI_IMR1_IM3
        str     r1,[r2]
        ldr     r2,=(NVIC_BASE+NVIC_ISER0)  ;enable the EXTI3 interrupt in NVIC_ISER0
        ldr     r1,=(1<<9)
        str     r1,[r2]
        bx      lr


exti5_init  PROC        ;initialize the external interrupt detector for PA.3
        EXPORT  exti5_init
        ldr     r2,=(RCC_BASE+RCC_APB2ENR)      ;enable SYSCFG block clock
        ldr     r1,[r2]
        orr     r1,#RCC_APB2ENR_SYSCFGEN
        str     r1,[r2]
        ldr     r2,=(SYSCFG_BASE+SYSCFG_EXTICR2)    ;select PA.3 and the trigger for EXTI0
        ldr     r1,[r2]
        bic     r1,#0x00000070                      ;This is the default anyway
        str     r1,[r2]
        ldr     r2,=(EXTI_BASE+EXTI_RTSR1)  ;enable rising edge trigger for EXTI0
        ldr     r1,[r2]
        orr     r1,#EXTI_RTSR1_RT5
        str     r1,[r2]
        ldr     r2,=(EXTI_BASE+EXTI_FTSR1)  ;disable falling edge trigger for EXTI0
        ldr     r1,[r2]
        bic     r1,#EXTI_FTSR1_FT5          ;also the default
        str     r1,[r2]
        ldr     r2,=(EXTI_BASE+EXTI_IMR1)   ;enable EXTI0 interrupt (unmask)
        ldr     r1,[r2]
        orr     r1,#EXTI_IMR1_IM5
        str     r1,[r2]
        ldr     r2,=(NVIC_BASE+NVIC_ISER0)  ;enable the EXTI3 interrupt in NVIC_ISER0
        ldr     r1,=(1<<23)
        str     r1,[r2]
        bx      lr
        ENDP
        ALIGN

        END

```

To make sure its pin 5 that is sending the signal we and the exit interrupt register with the bit we are looking for and then test the bit we are looking for with the result of the and. If it is 0 then it will continue the code and if the test fails it will break the exit interrupt condition.

Problem 3:



```

    INCLUDE core_cm4_constants.s        ; Load Constant Definitions
    INCLUDE stm32l476xx_constants.s

    AREA    main, CODE, READONLY



;Interrupt Support Code

tim2_init   PROC        ;initialize Timer 2 for this program and setup its interrupt
        EXPORT  tim2_init
        ldr     r2,=(RCC_BASE+RCC_APB1ENR1)     ;enable timer 2 clock
        ldr     r1,[r2]
        orr     r1,#RCC_APB1ENR1_TIM2EN
        str     r1,[r2]

        ldr     r2,=(TIM2_BASE+TIM_PSC)     ;Setup the prescaler.  Assuming a 4MHz clock, this gives 1ms timer ticks
        ldr     r1,=999
        str     r1,[r2]

        ldr     r2,=(TIM2_BASE+TIM_ARR)     ;Setup the reload.  Assuming a 1ms tick, this gives 1s overflows
        ldr     r1,=1
        str     r1,[r2]

        ldr     r2,=(TIM2_BASE+TIM_CR1)     ;enable the counter in control register 1
        ldr     r1,[r2]
        orr     r1,#TIM_CR1_CEN
        str     r1,[r2]

        ldr     r2,=(TIM2_BASE+TIM_DIER)        ;enable the timer update interrupt
        ldr     r1,[r2]
        orr     r1,#TIM_DIER_UIE
        str     r1,[r2]

        ldr     r2,=(NVIC_BASE+NVIC_ISER0)  ;enable the TIM2 interrupt in NVIC_ISER0
        ldr     r1,=(1<<28)
        str     r1,[r2]
        bx      lr

        ENDP
        ALIGN

        END

```



```
Measure     Current    Mean       Min      Max       Std Dev    Count
Freq(1):    1.00kHz    746.64Hz   0Hz      2.50kHz   848.30Hz   65
Pk-Pk(1):   6.72V      5.4612V    190mV    6.72V     1.8886V    66
```

```
Freq(1): 1.00kHz    Pk-Pk(1): 6.72V
```