Austin Petersen, Ayden Dauenhauer

Prof. Wolfe

ELEN 120L Tuesday 2:15-5:00

October 31, 2023

Report 5

Problem 1:

main.s | jstick.s | leds.s | leds.h | jstick.h | startup_stm32l476xx.s

```
13      INCLUDE core_cm4_constants.s        ; Load Constant Definitions
14      INCLUDE stm32l476xx_constants.s
15      INCLUDE jstick.h
16      INCLUDE leds.h
17
18
19
20
21              AREA    main, CODE, READONLY
22              EXPORT  __main
23              ENTRY
24
25  __main  PROC
26          ldr     r0,=RCC_AHB2ENR_GPIOBEN
27          bl      portclock_en                ; enable port B clock
28
29
30          ldr     r0,=GPIOB_BASE
31          ldr     r1,=GPIO_MODER_MODER2_0
32          bl      port_bit_pushpull           ;set port b.2 to push pull
33
34          bl      porta_init          ;initialize port A for this program
35          bl      exti3_init          ;initialize exti3 interrupt
36
37          ldr     r0,=RCC_AHB2ENR_GPIOEEN
38          bl      portclock_en                ; enable port B clock
39
40
41          ldr     r0,=GPIOE_BASE
42          ldr     r1,=GPIO_MODER_MODER8_0
43          bl      port_bit_pushpull           ;set port b.2 to push pull
44
45          bl      porta_init          ;initialize port A for this program
46          bl      exti0_init          ;initialize exti0 interrupt
47
48
49
50
51  endless b       endless
52          ENDP
53
54
55  EXTI3_IRQHandler PROC
56          EXPORT  EXTI3_IRQHandler
57          push    {lr}
58          bl      red_tog
59          pop     {lr}
60          ldr     r2,=(EXTI_BASE+EXTI_PR1)    ;reset pending interrupt for EXTI3
61          mov     r1,#EXTI_PR1_PIF3
62          str     r1,[r2]
63          dsb
64          bx      lr
65
66  EXTI0_IRQHandler PROC
67          EXPORT  EXTI0_IRQHandler
68          push    {lr}
69          bl      green_tog
70          pop     {lr}
71          ldr     r2,=(EXTI_BASE+EXTI_PR1)    ;reset pending interrupt for EXTI0
72          mov     r1,#EXTI_PR1_PIF0
73          str     r1,[r2]
74          dsb
75          bx      lr
76          ENDP
77
```

main.s | jstick.s | leds.s | leds.h | jstick.h | startup_stm32l476xx.s

```
32          and     r0,#0x0000002f
33          bx      lr
34          ENDP
35
36
37  ;Interrupt Support Code
38
39  exti3_init  PROC        ;initialize the external interrupt detector for PA.3
40          EXPORT  exti3_init
41          ldr     r2,=(RCC_BASE+RCC_APB2ENR)      ;enable SYSCFG block clock
42          ldr     r1,[r2]
43          orr     r1,#RCC_APB2ENR_SYSCFGEN
44          str     r1,[r2]
45          ldr     r2,=(SYSCFG_BASE+SYSCFG_EXTICR0)    ;select PA.3 and the trigger for EXTI3
46          ldr     r1,[r2]
47          bic     r1,#0x00007000                      ;This is the default anyway
48          str     r1,[r2]
49          ldr     r2,=(EXTI_BASE+EXTI_RTSR1)  ;enable rising edge trigger for EXTI3
50          ldr     r1,[r2]
51          orr     r1,#EXTI_RTSR1_RT3
52          str     r1,[r2]
53          ldr     r2,=(EXTI_BASE+EXTI_FTSR1)  ;disable falling edge trigger for EXTI3
54          ldr     r1,[r2]
55          bic     r1,#EXTI_FTSR1_FT3          ;also the default
56          str     r1,[r2]
57          ldr     r2,=(EXTI_BASE+EXTI_IMR1)   ;enable EXTI3 interrupt (unmask)
58          ldr     r1,[r2]
59          orr     r1,#EXTI_IMR1_IM3
60          str     r1,[r2]
61          ldr     r2,=(NVIC_BASE+NVIC_ISER0)  ;enable the EXTI3 interrupt in NVIC_ISER0
62          ldr     r1,=(1<<9)
63          str     r1,[r2]
64          bx      lr
65
66  exti0_init  PROC        ;initialize the external interrupt detector for PA.3
67          EXPORT  exti0_init
68          ldr     r2,=(RCC_BASE+RCC_APB2ENR)      ;enable SYSCFG block clock
69          ldr     r1,[r2]
70          orr     r1,#RCC_APB2ENR_SYSCFGEN
71          str     r1,[r2]
72          ldr     r2,=(SYSCFG_BASE+SYSCFG_EXTICR1)    ;select PA.3 and the trigger for EXTI0
73          ldr     r1,[r2]
74          bic     r1,#0x00000007                      ;This is the default anyway
75          str     r1,[r2]
76          ldr     r2,=(EXTI_BASE+EXTI_RTSR1)  ;enable rising edge trigger for EXTI0
77          ldr     r1,[r2]
78          orr     r1,#EXTI_RTSR1_RT0
79          str     r1,[r2]
80          ldr     r2,=(EXTI_BASE+EXTI_FTSR1)  ;disable falling edge trigger for EXTI0
81          ldr     r1,[r2]
82          bic     r1,#EXTI_FTSR1_FT0          ;also the default
83          str     r1,[r2]
84          ldr     r2,=(EXTI_BASE+EXTI_IMR1)   ;enable EXTI0 interrupt (unmask)
85          ldr     r1,[r2]
86          orr     r1,#EXTI_IMR1_IM0
87          str     r1,[r2]
88          ldr     r2,=(NVIC_BASE+NVIC_ISER0)  ;enable the EXTI3 interrupt in NVIC_ISER0
89          ldr     r1,=(1<<6)
90          str     r1,[r2]
91          bx      lr
92          ENDP
93          ALIGN
94
95          END
96
```

Problem 2:

main.s | jstick.s | leds.s | leds.h | jstick.h | startup_stm32l476xx.s

```
24
25  __main  PROC
26          ldr     r0,=RCC_AHB2ENR_GPIOBEN
27          bl      portclock_en                ; enable port B clock
28
29
30          ldr     r0,=GPIOB_BASE
31          ldr     r1,=GPIO_MODER_MODER2_0
32          bl      port_bit_pushpull           ;set port b.2 to push pull
33
34          bl      porta_init          ;initialize port A for this program
35          bl      exti3_init          ;initialize exti3 interrupt
36
37
38
39
40          ldr     r0,=RCC_AHB2ENR_GPIOEEN
41          bl      portclock_en                ; enable port B clock
42
43
44          ldr     r0,=GPIOE_BASE
45          ldr     r1,=GPIO_MODER_MODER8_0
46          bl      port_bit_pushpull           ;set port b.2 to push pull
47
48          bl      porta_init          ;initialize port A for this program
49          bl      exti5_init          ;initialize exti0 interrupt
50
51
52
53
54  endless b       endless
55          ENDP
56
57
58  EXTI3_IRQHandler PROC
59          EXPORT  EXTI3_IRQHandler
60          push    {lr}
61          bl      red_tog
62          pop     {lr}
63          ldr     r2,=(EXTI_BASE+EXTI_PR1)    ;reset pending interrupt for EXTI3
64          mov     r1,#EXTI_PR1_PIF3
65          str     r1,[r2]
66          dsb
67          bx      lr
68          ENDP
69
70  EXTI9_5_IRQHandler PROC
71          EXPORT  EXTI9_5_IRQHandler
72          ldr r0, =(EXTI_PR1)
73          ldr r1, [r0]
74          and r1, r1, #0x20
75          tst r1, #0x20
76          bne     out
77          push    {lr}
78          bl      green_tog
79          pop     {lr}
80          ldr     r2,=(EXTI_BASE+EXTI_PR1)    ;reset pending interrupt for EXTI0
81          mov     r1,#EXTI_PR1_PIF5
82          str     r1,[r2]
83          dsb
84  out     bx      lr
85          ENDP
86
87              ALIGN
88              AREA    myData, DATA, READWRITE
```

main.s | jstick.s* | leds.s | leds.h | jstick.h | startup_stm32l476xx.s

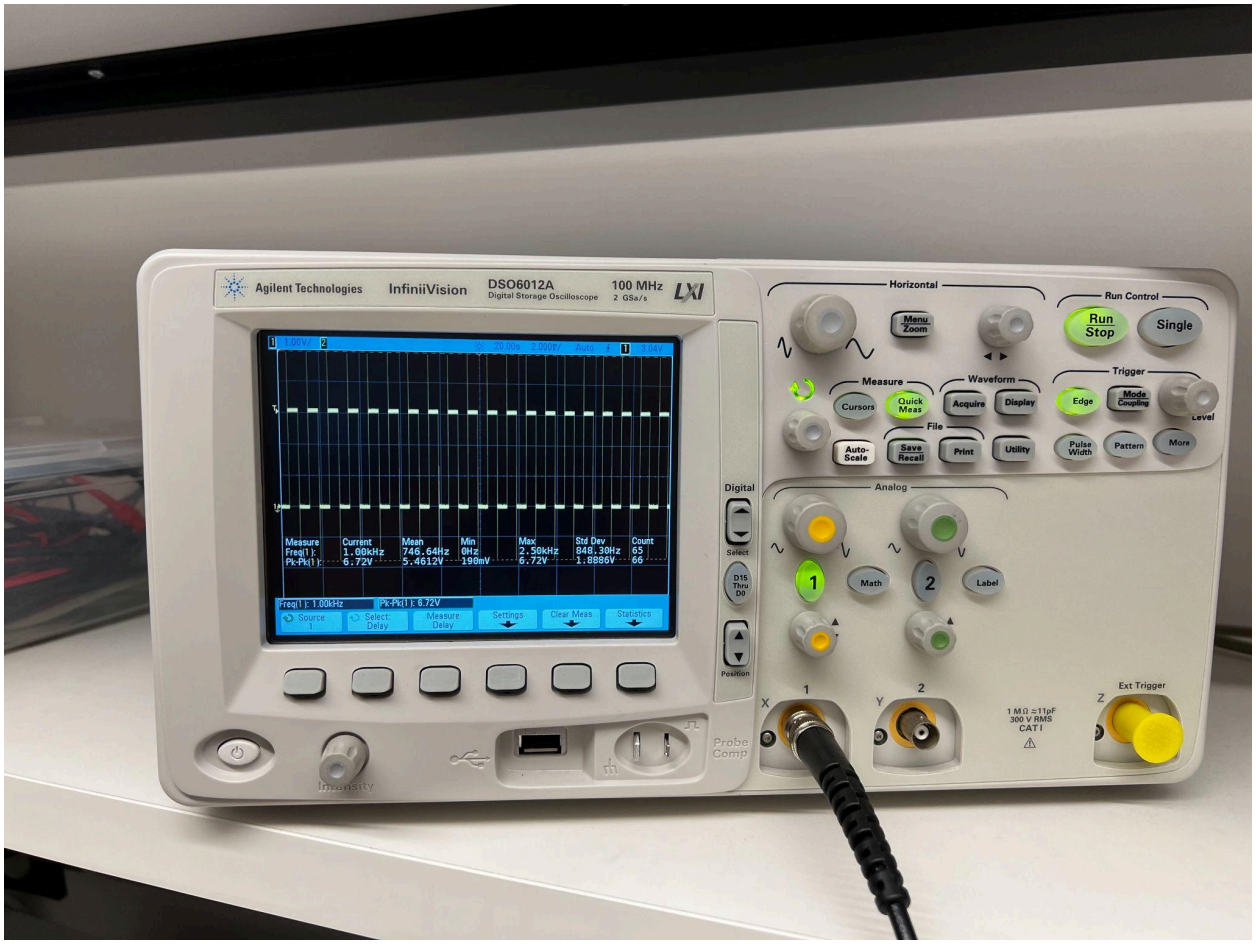
```
33          bx      lr
34          ENDP
35
36
37  ;Interrupt Support Code
38
39  exti3_init  PROC        ;initialize the external interrupt detector for PA.3
40          EXPORT  exti3_init
41          ldr     r2,=(RCC_BASE+RCC_APB2ENR)      ;enable SYSCFG block clock
42          ldr     r1,[r2]
43          orr     r1,#RCC_APB2ENR_SYSCFGEN
44          str     r1,[r2]
45          ldr     r2,=(SYSCFG_BASE+SYSCFG_EXTICR0)    ;select PA.3 and the trigger for EXTI3
46          ldr     r1,[r2]
47          bic     r1,#0x00007000                      ;This is the default anyway
48          str     r1,[r2]
49          ldr     r2,=(EXTI_BASE+EXTI_RTSR1)  ;enable rising edge trigger for EXTI3
50          ldr     r1,[r2]
51          orr     r1,#EXTI_RTSR1_RT3
52          str     r1,[r2]
53          ldr     r2,=(EXTI_BASE+EXTI_FTSR1)  ;disable falling edge trigger for EXTI3
54          ldr     r1,[r2]
55          bic     r1,#EXTI_FTSR1_FT3          ;also the default
56          str     r1,[r2]
57          ldr     r2,=(EXTI_BASE+EXTI_IMR1)   ;enable EXTI3 interrupt (unmask)
58          ldr     r1,[r2]
59          orr     r1,#EXTI_IMR1_IM3
60          str     r1,[r2]
61          ldr     r2,=(NVIC_BASE+NVIC_ISER0)  ;enable the EXTI3 interrupt in NVIC_ISER0
62          ldr     r1,=(1<<9)
63          str     r1,[r2]
64          bx      lr
65
66
67  exti5_init  PROC        ;initialize the external interrupt detector for PA.3
68          EXPORT  exti5_init
69          ldr     r2,=(RCC_BASE+RCC_APB2ENR)      ;enable SYSCFG block clock
70          ldr     r1,[r2]
71          orr     r1,#RCC_APB2ENR_SYSCFGEN
72          str     r1,[r2]
73          ldr     r2,=(SYSCFG_BASE+SYSCFG_EXTICR2)    ;select PA.3 and the trigger for EXTI0
74          ldr     r1,[r2]
75          bic     r1,#0x00000070                      ;This is the default anyway
76          str     r1,[r2]
77          ldr     r2,=(EXTI_BASE+EXTI_RTSR1)  ;enable rising edge trigger for EXTI0
78          ldr     r1,[r2]
79          orr     r1,#EXTI_RTSR1_RT5
80          str     r1,[r2]
81          ldr     r2,=(EXTI_BASE+EXTI_FTSR1)  ;disable falling edge trigger for EXTI0
82          ldr     r1,[r2]
83          bic     r1,#EXTI_FTSR1_FT5          ;also the default
84          str     r1,[r2]
85          ldr     r2,=(EXTI_BASE+EXTI_IMR1)   ;enable EXTI0 interrupt (unmask)
86          ldr     r1,[r2]
87          orr     r1,#EXTI_IMR1_IM5
88          str     r1,[r2]
89          ldr     r2,=(NVIC_BASE+NVIC_ISER0)  ;enable the EXTI3 interrupt in NVIC_ISER0
90          ldr     r1,=(1<<23)
91          str     r1,[r2]
92          bx      lr
93          ENDP
94          ALIGN
95
96          END
97
```

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:

main.s | startup_stm32l476xx.s | timer.s

```
1
2       INCLUDE core_cm4_constants.s        ; Load Constant Definitions
3       INCLUDE stm32l476xx_constants.s
4
5       AREA    main, CODE, READONLY
6
7
8
9   ;Interrupt Support Code
10
11  tim2_init   PROC        ;initialize Timer 2 for this program and setup its interrupt
12          EXPORT  tim2_init
13          ldr     r2,=(RCC_BASE+RCC_APB1ENR1)     ;enable timer 2 clock
14          ldr     r1,[r2]
15          orr     r1,#RCC_APB1ENR1_TIM2EN
16          str     r1,[r2]
17
18          ldr     r2,=(TIM2_BASE+TIM_PSC)     ;Setup the prescaler.  Assuming a 4MHz clock, this gives 1ms timer ticks
19          ldr     r1,=999
20          str     r1,[r2]
21
22          ldr     r2,=(TIM2_BASE+TIM_ARR)     ;Setup the reload.  Assuming a 1ms tick, this gives 1s overflows
23          ldr     r1,=1
24          str     r1,[r2]
25
26          ldr     r2,=(TIM2_BASE+TIM_CR1)     ;enable the counter in control register 1
27          ldr     r1,[r2]
28          orr     r1,#TIM_CR1_CEN
29          str     r1,[r2]
30
31          ldr     r2,=(TIM2_BASE+TIM_DIER)        ;enable the timer update interrupt
32          ldr     r1,[r2]
33          orr     r1,#TIM_DIER_UIE
34          str     r1,[r2]
35
36          ldr     r2,=(NVIC_BASE+NVIC_ISER0)  ;enable the TIM2 interrupt in NVIC_ISER0
37          ldr     r1,=(1<<28)
38          str     r1,[r2]
39          bx      lr
40          
41          ENDP
42          ALIGN
43
44          END
45
```


Agilent Technologies
InfiniiVision
DSO6012A
Digital Storage Oscilloscope
100 MHz
2 GSa/s
LXI
Horizontal
Menu Zoom
Run Control
Run Stop
Single
Measure
Cursors
Quick Meas
Waveform
Acquire
Display
Trigger
Edge
Mode Coupling
Level
File
Auto-Scale
Save Recall
Print
Utility
Pulse Width
Pattern
More
Digital
Select
D15 Thru D0
Position
Analog
1
Math
2
Label
Measure
Current
Mean
Min
Max
Std Dev
Count
Freq(1)
1.00kHz
746.64Hz
0Hz
2.50kHz
948.30Hz
65
Pk-Pk(1)
6.72V
5.4012V
190mV
6.72V
1.8986V
66
Source 1
Select Delay
Measure Delay
Settings
Clear Meas
Statistics
Intensity
Probe Comp
X
1
Y
2
Z
Ext Trigger
1 MΩ ≃11pF
300 V RMS
CAT I