

Austin Petersen and Ayden Dauenhauer

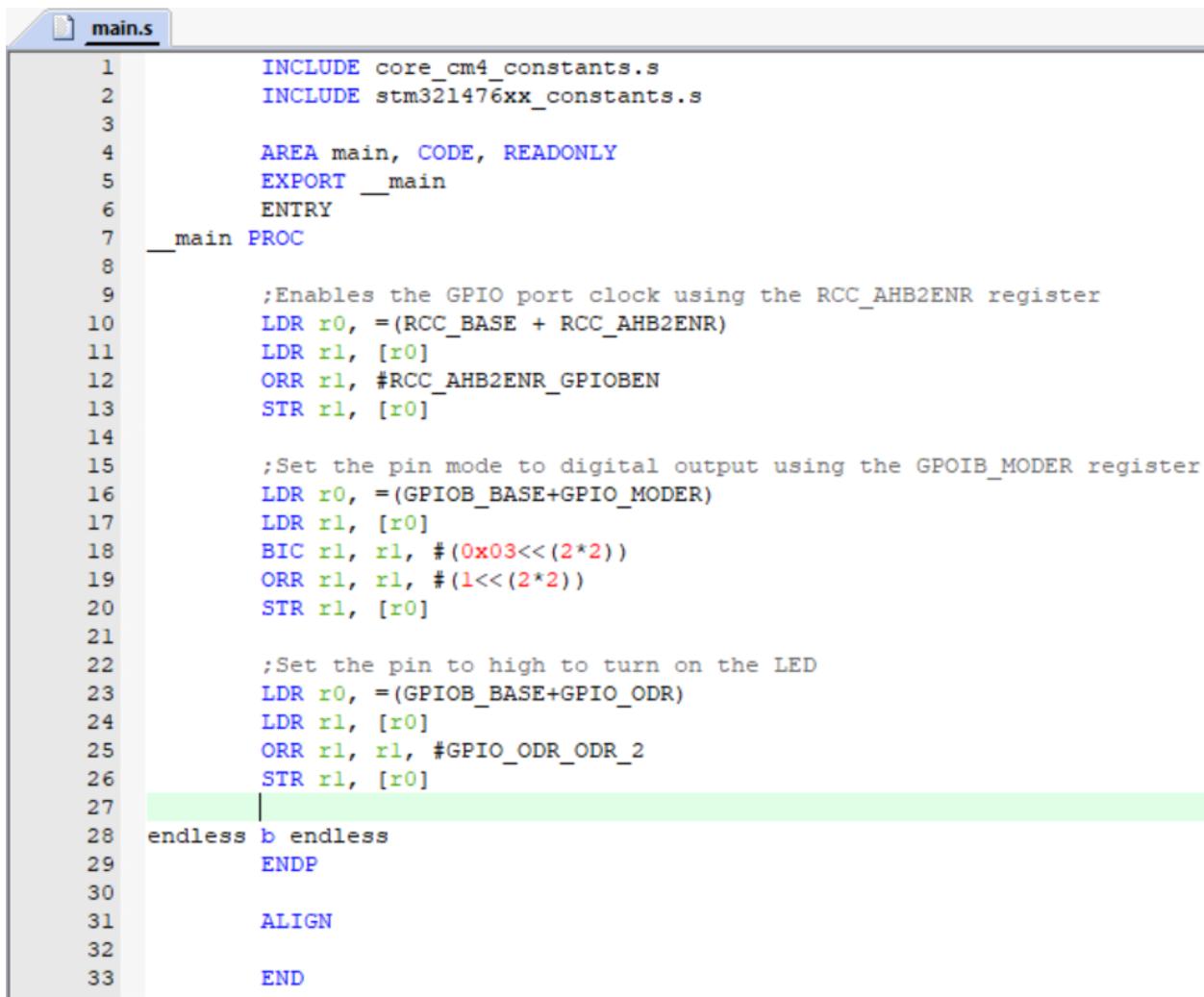
Prof. Wolfe

ELEN 120L Tuesday 2:15 p.m.

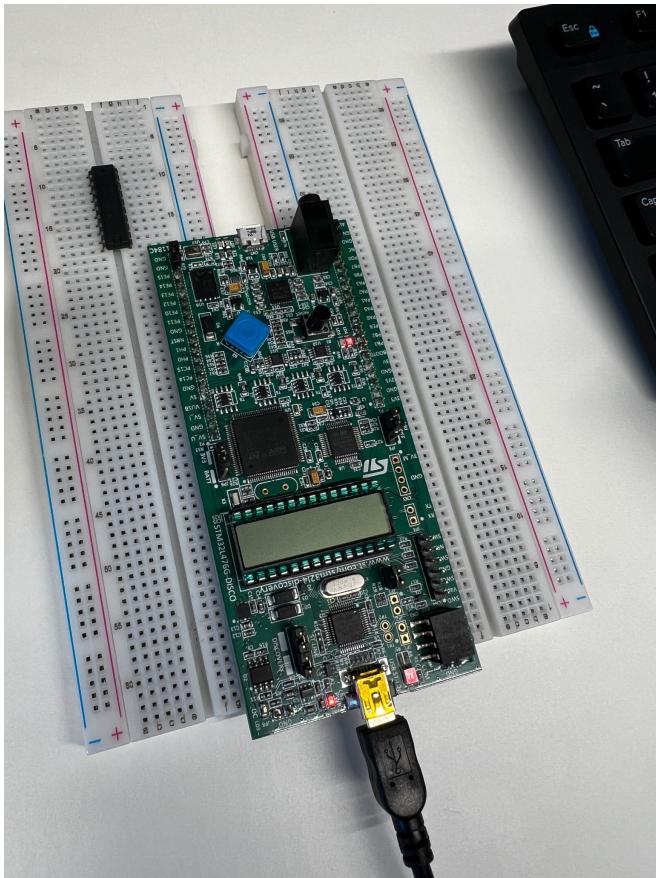
23 October 2023

Lab 3 Report

Problem 1:

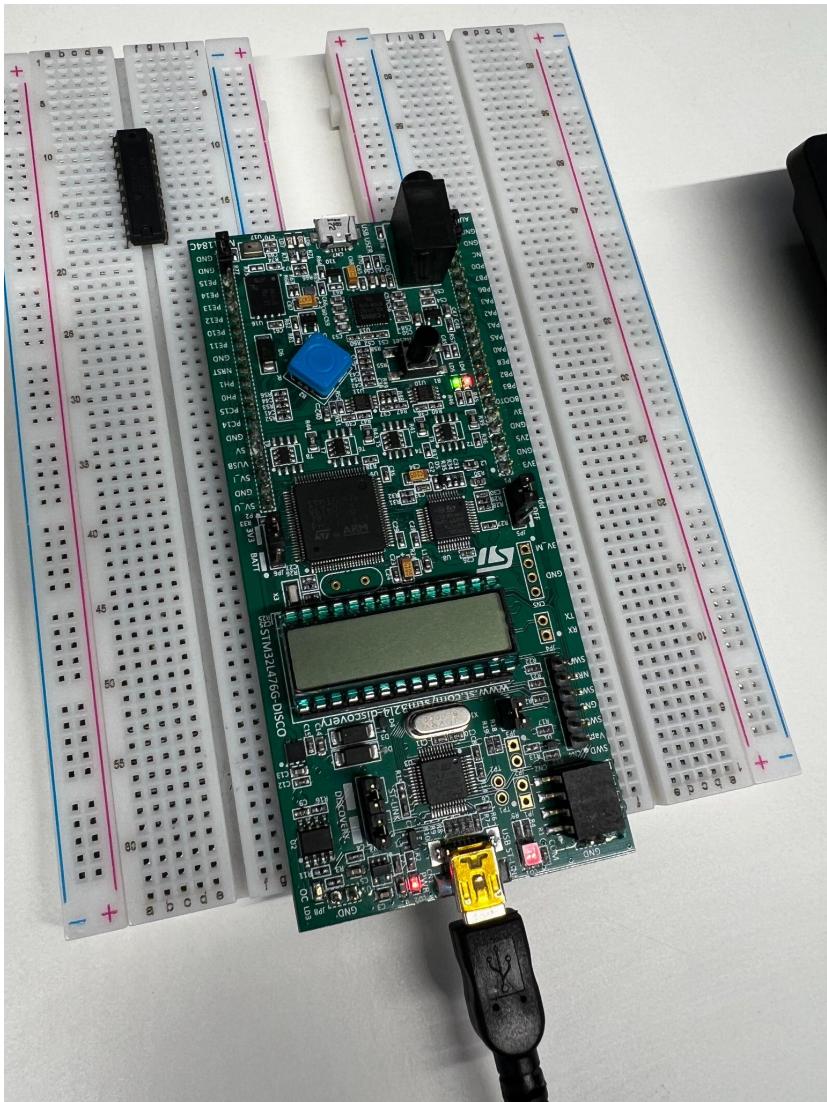


```
main.s
1      INCLUDE core_cm4_constants.s
2      INCLUDE stm321476xx_constants.s
3
4      AREA main, CODE, READONLY
5      EXPORT __main
6      ENTRY
7      __main PROC
8
9          ;Enables the GPIO port clock using the RCC_AHB2ENR register
10         LDR r0, =(RCC_BASE + RCC_AHB2ENR)
11         LDR r1, [r0]
12         ORR r1, #RCC_AHB2ENR_GPIOBEN
13         STR r1, [r0]
14
15         ;Set the pin mode to digital output using the GPOIB_MODER register
16         LDR r0, =(GPIOB_BASE+GPIO_MODER)
17         LDR r1, [r0]
18         BIC r1, r1, #(0x03<<(2*2))
19         ORR r1, r1, #(1<<(2*2))
20         STR r1, [r0]
21
22         ;Set the pin to high to turn on the LED
23         LDR r0, =(GPIOB_BASE+GPIO_ODR)
24         LDR r1, [r0]
25         ORR r1, r1, #GPIO_ODR_ODR_2
26         STR r1, [r0]
27
28     endless b endless
29     ENDP
30
31     ALIGN
32
33     END
```



Problem 2:

```
main.s
1      INCLUDE core_cm4_constants.s
2      INCLUDE stm321476xx_constants.s
3
4          AREA main, CODE, READONLY
5          EXPORT __main
6          ENTRY
7          __main PROC
8
9              ;Enables the GPIO port clock using the RCC_AHB2ENR register
10             LDR r0, =(RCC_BASE + RCC_AHB2ENR)
11             LDR r1, [r0]
12             ORR r1, #RCC_AHB2ENR_GPIOBEN
13             STR r1, [r0]
14
15             ;Set the pin mode to digital output using the GPOIB_MODER register
16             LDR r0, =(GPIOB_BASE+GPIO_MODER)
17             LDR r1, [r0]
18             BIC r1, r1, #(0x03<<(2*2))
19             ORR r1, r1, #(1<<(2*2))
20             STR r1, [r0]
21
22             ;Set the pin to high to turn on the red LED
23             LDR r0, =(GPIOB_BASE+GPIO_ODR)
24             LDR r1, [r0]
25             ORR r1, r1, #GPIO_ODR_ODR_2
26             STR r1, [r0]
27
28             ;Enables the GPIO port clock using the RCC_AHB2ENR register
29             LDR r0, =(RCC_BASE + RCC_AHB2ENR)
30             LDR r1, [r0]
31             ORR r1, #RCC_AHB2ENR_GPIOEEN
32             STR r1, [r0]
33
34             ;Set the pin mode to digital output using the GPOIE_MODER register
35             LDR r0, =(GPIOE_BASE+GPIO_MODER)
36             LDR r1, [r0]
37             BIC r1, r1, #(0x03<<(2*8))
38             ORR r1, r1, #(1<<(2*8))
39             STR r1, [r0]
40
41             ;Set the pin to high to turn on the green LED
42             LDR r0, =(GPIOE_BASE+GPIO_ODR)
43             LDR r1, [r0]
44             ORR r1, r1, #GPIO_ODR_ODR_8
45             STR r1, [r0]
46
47         | endless b endless
48             ENDP
49
50             ALIGN
51
52             END
```



Problem 3:

```
main.s startup_stm32l476xx.s
1     INCLUDE core_cm4_constants.s
2     INCLUDE stm32l476xx_constants.s
3
4     AREA main, CODE, READONLY
5     EXPORT __main
6     ENTRY
7     __main PROC
8
9         ;Enables the GPIO port clock using the RCC_AHB2ENR register
10        LDR r0, =(RCC_BASE + RCC_AHB2ENR)
11        LDR r1, [r0]
12        ORR r1, #RCC_AHB2ENR_GPIOBEN
13        STR r1, [r0]
14
15        ;Set the pin mode to digital output using the GPOIB_MODER register
16        LDR r0, =(GPIOB_BASE+GPIO_MODER)
17        LDR r1, [r0]
18        BIC r1, r1, #(0x03<<(2*2))
19        ORR r1, r1, #(1<<(2*2))
20        STR r1, [r0]
21
22        ;Enables the GPIO port clock using the RCC_AHB2ENR register
23        LDR r0, =(RCC_BASE + RCC_AHB2ENR)
24        LDR r1, [r0]
25        ORR r1, #RCC_AHB2ENR_GPIOEEN
26        STR r1, [r0]
27
28        ;Set the pin mode to digital output using the GPOIE_MODER register
29        LDR r0, =(GPIOE_BASE+GPIO_MODER)
30        LDR r1, [r0]
31        BIC r1, r1, #(0x03<<(2*8))
32        ORR r1, r1, #(1<<(2*8))
33        STR r1, [r0]
34
35        ;Set the pin to high to turn on the green LED
36        LDR r0, =(GPIOE_BASE+GPIO_ODR)
37        LDR r1, [r0]
38        ORR r1, r1, #GPIO_ODR_ODR_8
39        STR r1, [r0]
40
41        ;Set the pin to high to turn on the red LED
42        LDR r0, =(GPIOB_BASE+GPIO_ODR)
43        LDR r1, [r0]
44        ORR r1, r1, #GPIO_ODR_ODR_2
45        STR r1, [r0]
46
47        MOV r0, #0x00100010
48        MOV r1, #0
49    loop
50        ADD r1, #1
51        CMP r0, r1
52        BNE loop
53
54        ;Set the pin to high to turn on the green LED
55        LDR r0, =(GPIOE_BASE+GPIO_ODR)
56        LDR r1, [r0]
57        BIC r1, r1, #GPIO_ODR_ODR_8
58        STR r1, [r0]
59
60        ;Set the pin to high to turn on the red LED
61        LDR r0, =(GPIOB_BASE+GPIO_ODR)
62        LDR r1, [r0]
63        BIC r1, r1, #GPIO_ODR_ODR_2
64        STR r1, [r0]
65
66    endless b endless
67        ENDP
68
69        ALIGN
70
71        END
```

Runtime per 10,000 loops:

Before loop runtime: .00067970

After loop runtime: 0.00467970

Loop runtime per 10,000 loops: 0.004 seconds

Number of loops to reach 1 second: 2,500,000

This is 1 second in the compiler, but is not actually 1 second on the board.

It is about 2.5 seconds on the board, so dividing 2.5 million by 2.5 is 1 million. Changing the loop to 0x00100010 iterations gets us to approximately 1 second within 4.63%.

Problem 4:

```
main.s
startup_stm32l476xx.s

1      INCLUDE core_cm4_constants.s
2      INCLUDE stm32l476xx_constants.s
3
4      AREA main, CODE, READONLY
5      EXPORT __main
6      ENTRY
7      __main PROC
8
9          ;Enables the GPIO port clock using the RCC_AHB2ENR register
10         LDR r0, =(RCC_BASE + RCC_AHB2ENR)
11         LDR r1, [r0]
12         ORR r1, #RCC_AHB2ENR_GPIOBEN
13         STR r1, [r0]
14
15         ;Set the pin mode to digital output using the GPOIB_MODER register
16         LDR r0, =(GPIOB_BASE+GPIO_MODER)
17         LDR r1, [r0]
18         BIC r1, r1, #(0x03<<(2*2))
19         ORR r1, r1, #(1<<(2*2))
20         STR r1, [r0]
21
22         ;Enables the GPIO port clock using the RCC_AHB2ENR register
23         LDR r0, =(RCC_BASE + RCC_AHB2ENR)
24         LDR r1, [r0]
25         ORR r1, #RCC_AHB2ENR_GPIOEEN
26         STR r1, [r0]
27
28         ;Set the pin mode to digital output using the GPOIE_MODER register
29         LDR r0, =(GPIOE_BASE+GPIO_MODER)
30         LDR r1, [r0]
31         BIC r1, r1, #(0x03<<(2*8))
32         ORR r1, r1, #(1<<(2*8))
33         STR r1, [r0]
34
35         ;Set the pin to high to turn on the green LED
36     endless| LDR r0, =(GPIOE_BASE+GPIO_ODR)
37         LDR r1, [r0]
38         ORR r1, r1, #GPIO_ODR_ODR_8
39         STR r1, [r0]
40
41         MOV r0, #0x000100010
42         MOV r1, #0
43     loop1| ADD r1, #1
44         CMP r0, r1
45         BNE loop1
46
47         ;Set the pin to high to turn off the green LED
48         LDR r0, =(GPIOE_BASE+GPIO_ODR)
49         LDR r1, [r0]
50         BIC r1, r1, #GPIO_ODR_ODR_8
51         STR r1, [r0]
52
53         ;Set the pin to high to turn on the red LED
54         LDR r0, =(GPIOB_BASE+GPIO_ODR)
55         LDR r1, [r0]
```

```
56          ORR r1, r1, #GPIO_ODR_ODR_2
57          STR r1, [r0]
58
59          MOV r0, #0x00100010
60          MOV r1, #0
61      loop2  ADD r1, #1
62          CMP r0, r1
63          BNE loop2
64
65          ;Set the pin to high to turn off the red LED
66          LDR r0, =(GPIOB_BASE+GPIO_ODR)
67          LDR r1, [r0]
68          BIC r1, r1, #GPIO_ODR_ODR_2
69          STR r1, [r0]
70          b endless
71
72          ENDP
73
74          ALIGN
75
76          END
```