

Christian Garcia

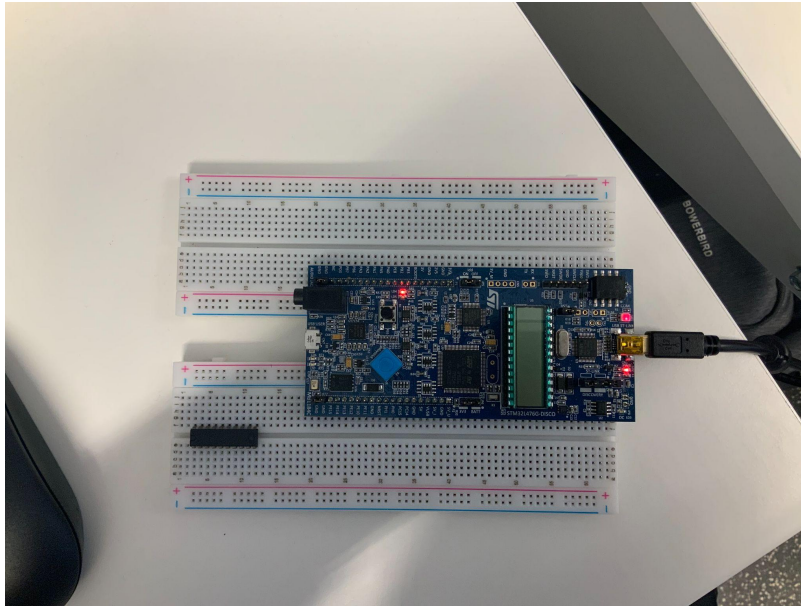
ELEN 120 Lab

19 October 2021

Lab 3: GPIO

Problem 1:

```
5 ; @note
6 ; This code is for the book "Embedded Systems with ARM Cortex-M
7 ; Microcontrollers in Assembly Language and C, Yifeng Zhu,
8 ; ISBN-13: 978-0982692639, ISBN-10: 0982692633 as used at Santa Clara Uni
9 ; *****
10
11
12
13 INCLUDE core_cm4_constants.s ; Load Constant Definitions
14 INCLUDE stm32l476xx_constants.s
15
16 AREA main, CODE, READONLY
17 EXPORT __main
18 ENTRY
19
20 __main PROC
21
22     ldr    r0, =(RCC_BASE+RCC_AHB2ENR)
23     ldr    r1, [r0]
24     orr    r1, #RCC_AHB2ENR_GPIOBEN ;Set bit 1 of AHB2ENR to 1
25     str    r1, [r0]
26
27     ldr    r0, =(GPIOB_BASE+GPIO_MODER)
28     ldr    r1, [r0]
29     bic    r1, r1, #(0x03<<(2*2)) ;Clear bit 5 of MODER
30     orr    r1, r1, #(1<<(2*2)) ;Set bit 4 of MODER to 1
31     str    r1, [r0]
32
33     ldr    r0, =(GPIOB_BASE+GPIO_ODR)
34     ldr    r1, [r0]
35     orr    r1, r1, #GPIO_ODR_ODR_2 ;Set bit 2 of ODR to 1
36     str    r1, [r0]
37
38 endless    b    endless
39
40     ENDP
41     ALIGN
42     AREA myData, DATA, READWRITE
43     ALIGN
44
45 counter    DCD    10
46
47     END
48
```

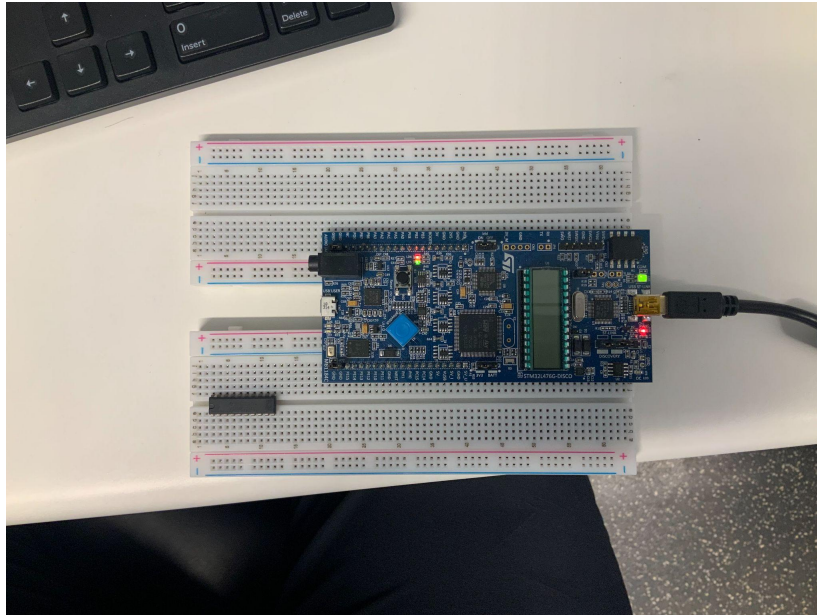


Problem 2:

```

19
20 __main PROC
21
22     ldr    r0, =(RCC_BASE+RCC_AHB2ENR)
23     ldr    r1, [r0]
24     orr    r1, #RCC_AHB2ENR_GPIOEN      ;Set bit 1 of AHB2ENR to 1
25     str    r1, [r0]
26
27     ldr    r0, =(GPIOB_BASE+GPIO_MODER)
28     ldr    r1, [r0]
29     bic    r1, r1, #(0x03<<(2*2))      ;Clear bit 5 of MODER
30     orr    r1, r1, #(1<<(2*2))          ;Set bit 4 of MODER to 1
31     str    r1, [r0]
32
33     ldr    r0, =(GPIOB_BASE+GPIO_ODR)
34     ldr    r1, [r0]
35     orr    r1, r1, #GPIO_ODR_ODR_2      ;Set bit 2 of ODR to 1
36     str    r1, [r0]
37
38     ldr    r0, =(RCC_BASE+RCC_AHB2ENR)
39     ldr    r3, [r0]
40     orr    r3, #RCC_AHB2ENR_GPIOEEN      ;Set bit 1 of AHB2ENR to 1
41     str    r3, [r0]
42
43     ldr    r0, =(GPIOE_BASE+GPIO_MODER)
44     ldr    r3, [r0]
45     bic    r3, #0x00020000              ;Clear bit 17 of MODER
46     orr    r3, #0x00010000              ;Set bit 16 of MODER to 1
47     str    r3, [r0]
48
49     ldr    r0, =(GPIOE_BASE+GPIO_ODR)
50     ldr    r3, [r0]
51     orr    r3, r1, #GPIO_ODR_ODR_8      ;Set bit 8 of ODR to 1
52     str    r3, [r0]
53
54 endless    b    endless
55
56 ENDP
57 ALIGN
58 AREA      myData, DATA, READWRITE
59 ALIGN
60

```

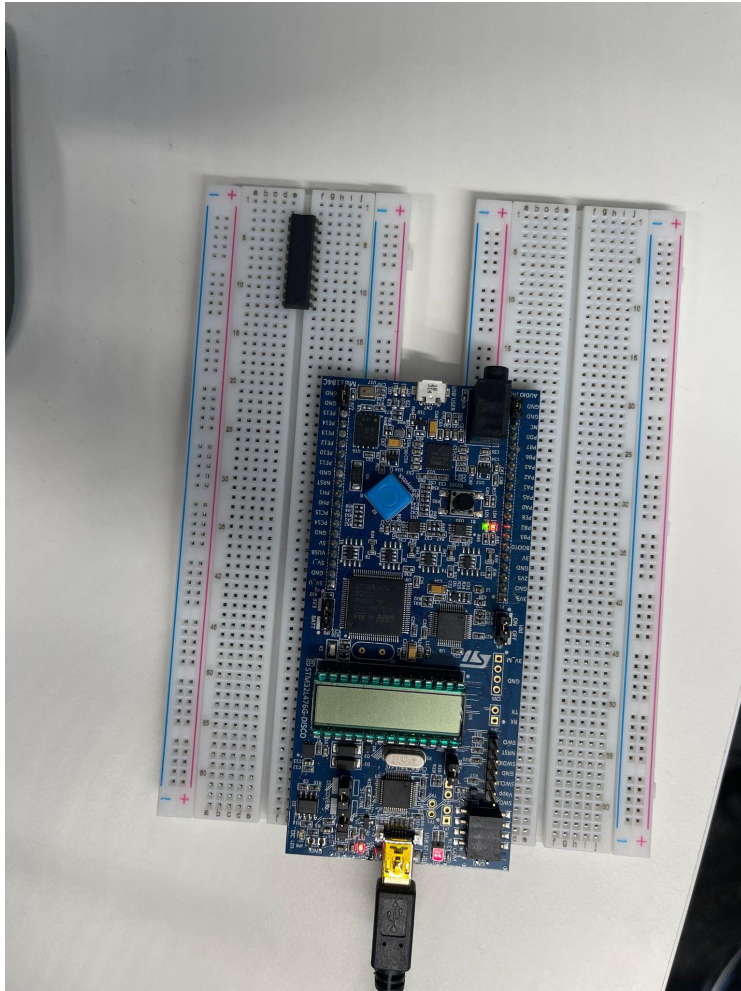


Problem 3:

```

20  _main PROC
21  |
22      ldr    r6, =counter
23      mov    r7, #1
24
25  loop    ldr    r8, [r6]
26
27      ldr    r0, =(RCC_BASE+RCC_AHB2ENR)
28      ldr    r1, [r0]
29      orr    r1, #RCC_AHB2ENR_GPIOEN        ;Set bit 1 of AHB2ENR to 1
30      str    r1, [r0]
31
32      ldr    r0, =(GPIOB_BASE+GPIO_MODER)
33      ldr    r1, [r0]
34      bic    r1, r1, #(0x03<<(2*2))        ;Clear bit 5 of MODER
35      orr    r1, r1, # (1<<(2*2))          ;Set bit 4 of MODER to 1
36      str    r1, [r0]
37
38      ldr    r0, =(GPIOB_BASE+GPIO_ODR)
39      ldr    r1, [r0]
40      orr    r1, r1, #GPIO_ODR_ODR_2        ;Set bit 2 of ODR to 1
41      str    r1, [r0]
42
43      ldr    r0, =(RCC_BASE+RCC_AHB2ENR)
44      ldr    r3, [r0]
45      orr    r3, #RCC_AHB2ENR_GPIOEEN        ;Set bit 1 of AHB2ENR to 1
46      str    r3, [r0]
47
48      ldr    r0, =(GPIOE_BASE+GPIO_MODER)
49      ldr    r3, [r0]
50      bic    r3, #0x00020000                ;Clear bit 17 of MODER
51      orr    r3, #0x00010000                ;Set bit 16 of MODER to 1
52      str    r3, [r0]
53
54      ldr    r0, =(GPIOE_BASE+GPIO_ODR)
55      ldr    r3, [r0]
56      orr    r3, r1, #GPIO_ODR_ODR_8        ;Set bit 8 of ODR to 1
57      str    r3, [r0]
58
59      add    r7, r7, #1
60      cmp    r7, r8
61
62      bne    loop
63
64      ldr    r0, =(GPIOB_BASE+GPIO_ODR)

```



For the 1 second delay I chose to create a loop that runs 50000 times. I got the number 50000 from the time it took to run one loop. On the left side of the screen it said that it took 0.000023 seconds to run 1 loop then if I divide 1 sec by 0.000023 sec then I get the value around 50000.

Problem 4:

```

13      INCLUDE core_cm4_constants.s      ; Load Constant Definitions
14      INCLUDE stm32l476xx_constants.s
15
16      AREA main, CODE, READONLY
17      EXPORT __main
18      ENTRY
19
20      __main PROC
21
22      endless
23          ldr    r0, =(RCC_BASE+RCC_AHB2ENR)
24          ldr    r1, [r0]
25          orr    r1, #RCC_AHB2ENR_GPIOBEN      ;Set bit 1 of AHB2ENR to 1
26          str    r1, [r0]
27
28          ldr    r0, =(GPIOB_BASE+GPIO_MODER)
29          ldr    r1, [r0]
30          bic    r1, r1, #(0x03<<(2*2))      ;Clear bit 5 of MODER
31          orr    r1, r1, #(1<<(2*2))          ;Set bit 4 of MODER to 1
32          str    r1, [r0]
33
34          ldr    r0, =(GPIOB_BASE+GPIO_ODR)
35          ldr    r1, [r0]
36          orr    r1, r1, #GPIO_ODR_ODR_2      ;Set bit 2 of ODR to 1
37          str    r1, [r0]
38
39          ldr    r7, =750000
40          mov    r6, #0x0
41      loop    add    r6, r6, #1
42             cmp    r7, r6
43             bne    loop
44
45             ldr    r0, =(GPIOB_BASE+GPIO_ODR)
46             ldr    r1, [r0]
47             bic    r1, r1, #GPIO_ODR_ODR_2      ;Clear bit 2 of ODR
48             str    r1, [r0]
49
50             ldr    r0, =(RCC_BASE+RCC_AHB2ENR)
51             ldr    r3, [r0]
52             orr    r3, #RCC_AHB2ENR_GPIOEEN      ;Set bit 1 of AHB2ENR to 1
53             str    r3, [r0]
54
55             ldr    r0, =(GPIOE_BASE+GPIO_MODER)
56             ldr    r3, [r0]
57             bic    r3, #0x00020000      ;Clear bit 17 of MODER

```

```

53             str    r3, [r0]
54
55             ldr    r0, =(GPIOE_BASE+GPIO_ODR)
56             ldr    r3, [r0]
57             bic    r3, #0x00020000      ;Clear bit 17 of MODER
58             orr    r3, #0x00010000      ;Set bit 16 of MODER to 1
59             str    r3, [r0]
60
61             ldr    r0, =(GPIOE_BASE+GPIO_ODR)
62             ldr    r3, [r0]
63             orr    r3, r1, #GPIO_ODR_ODR_8      ;Set bit 8 of ODR to 1
64             str    r3, [r0]
65
66             mov    r6, #0x0
67      aloop    add    r6, r6, #1
68             cmp    r7, r6
69             bne    aloop
70
71             ldr    r0, =(GPIOE_BASE+GPIO_ODR)
72             ldr    r3, [r0]
73             bic    r3, r1, #GPIO_ODR_ODR_8      ;Clear bit 8 of ODR
74             str    r3, [r0]
75
76
77      b        endless
78
79      ENDP
80      ALIGN
81      AREA myData, DATA, READWRITE
82      ALIGN
83
84      counter DCD    50000
85
86      END
87

```