## COMPENG 4DS4 Lab 0 Report

Aaron Pinto pintoa9 Jingming Liu liuj171 Raeed Hassan hassam41 Jeffrey Guo guoj69

January 26, 2023

## Listing 1: Problem 1

```
/* Definitions */
#define MEM_LOC_CHAR(x)
                                  *((char*)x)
  #define MEM_LOC_SHORT(x)
                                  *((short*)x)
#define MEM_LOC_INT(x)
                                  *((int*)x)
5 #define LOC1
                                  MEM_LOC_CHAR(0x20001000)
6 #define LOC2
                                  MEM_LOC_INT(0x20001001)
  #define LOC3
                                  MEM_LOC_SHORT (0x20001005)
 #define LOC4
                                  MEM_LOC_INT(0x20001007)
  #define ARBITRARY_LOCATION
                                        *((int*) 0x20001004)
  /* Problem 1 Function */
11
  void problem1() {
12
    LOC1 = OxAC;
13
    LOC2 = OxAABBCCDD;
14
    LOC3 = OxABCD;
    LOC4 = OxAABBCCDD;
16
  }
17
```

## Listing 2: Problem 3

```
/* Definitions */
#define BOARD_LED_GPIO_BLUE
                                      GPIOC
  #define BOARD_LED_GPIO_PIN_BLUE
4 #define BOARD_LED_GPIO_GREEN
                                      GPIOC
#define BOARD_LED_GPIO_PIN_GREEN
  #define BOARD_LED_GPIO_RED
                                      GPIOD
  #define BOARD_LED_GPIO_PIN_RED
   /* Problem 3 main Function */
   int main(void) {
       /* Define the init structure for the output LED pin*/
        gpio_pin_config_t led_config = {kGPIO_DigitalOutput, 0};
12
13
       /* Board pin, clock, debug console init */
14
       BOARD_InitBootPins();
       BOARD_InitBootClocks();
       BOARD_InitDebugConsole();
17
18
       /* Print a note to terminal. */
19
       PRINTF("\r\n GPIO Driver example\r\n");
20
       PRINTF("\r\n The LED is blinking.\r\n");
       /* Init output LED GPIO. */
23
       GPIO_PinInit(BOARD_LED_GPIO_BLUE, BOARD_LED_GPIO_PIN_BLUE, &led_config
24
          );
       GPIO_PinInit(BOARD_LED_GPIO_GREEN, BOARD_LED_GPIO_PIN_GREEN, &
25
          led_config);
       GPIO_PinInit(BOARD_LED_GPIO_RED, BOARD_LED_GPIO_PIN_RED, &led_config);
26
       while (1) {
28
           delay();
29
           GPIO_PortToggle(BOARD_LED_GPIO_BLUE, 1u << BOARD_LED_GPIO_PIN_BLUE
30
              );
```

```
delay();
31
            GPIO_PortToggle(BOARD_LED_GPIO_BLUE, 1u << BOARD_LED_GPIO_PIN_BLUE
32
               );
           delay();
33
            GPIO_PortToggle(BOARD_LED_GPIO_GREEN, 1u <<
34
               BOARD_LED_GPIO_PIN_GREEN);
            delay();
            GPIO_PortToggle(BOARD_LED_GPIO_GREEN, 1u <<</pre>
36
               BOARD_LED_GPIO_PIN_GREEN);
            delay();
           GPIO_PortToggle(BOARD_LED_GPIO_RED, 1u << BOARD_LED_GPIO_PIN_RED);</pre>
38
           delay();
39
            GPIO_PortToggle(BOARD_LED_GPIO_RED, 1u << BOARD_LED_GPIO_PIN_RED);
40
       }
41
   }
42
```

## Listing 3: Problem 4

```
/* Definitions */
   #define GPIOA ((GPIO_Struct*) 0x400FF000)
   #define GPIOB ((GPIO_Struct*) 0x400FF040)
   #define GPIOC ((GPIO_Struct*) 0x400FF080)
   #define GPIOD ((GPIO_Struct*) 0x400FF0C0)
   #define GPIOE ((GPIO_Struct*) 0x400FF100)
   /* Problem 4 Helper Functions */
   void initPin(GPIO_Struct *port, uint32_t pin) {
     writePin(port, pin, 1);
     port -> GPIO_PDDR |= (1UL << pin);</pre>
11
   }
12
13
   void togglePin(GPIO_Struct *port, uint32_t mask) {
14
     port -> GPIO_PDOR ^= (mask);
   }
16
   void writePin(GPIO_Struct *port, uint32_t pin, uint8_t output) {
18
     if (output == 0U) {
19
       port->GPIO_PDOR |= (1UL << pin);</pre>
20
     } else {
21
       port -> GPIO_PDOR &= ~(1UL << pin);</pre>
23
   }
24
25
   /* Problem 4 main Function */
   int main(void) {
2.7
       /* Board pin, clock, debug console init */
28
       BOARD_InitBootPins();
29
       BOARD_InitBootClocks();
30
       BOARD_InitDebugConsole();
31
       /* Print a note to terminal. */
33
       PRINTF("\r\nGPIO Driver example\r\n");
       PRINTF("\r\nThe LED is blinking.\r\n");
35
36
```

```
/* Init output LED GPIO. */
37
       initPin(BOARD_LED_GPIO_BLUE, BOARD_LED_GPIO_PIN_BLUE);
38
       initPin(BOARD_LED_GPIO_GREEN, BOARD_LED_GPIO_PIN_GREEN);
39
       initPin(BOARD_LED_GPIO_RED, BOARD_LED_GPIO_PIN_RED);
40
41
       while (1) {
42
           delay();
43
           togglePin(BOARD_LED_GPIO_BLUE, 1u << BOARD_LED_GPIO_PIN_BLUE);</pre>
44
45
           togglePin(BOARD_LED_GPIO_BLUE, 1u << BOARD_LED_GPIO_PIN_BLUE);
46
           delay();
47
           togglePin(BOARD_LED_GPIO_GREEN, 1u << BOARD_LED_GPIO_PIN_GREEN);
48
           delay();
49
           togglePin(BOARD_LED_GPIO_GREEN, 1u << BOARD_LED_GPIO_PIN_GREEN);
           delay();
           togglePin(BOARD_LED_GPIO_RED, 1u << BOARD_LED_GPIO_PIN_RED);
           delay();
53
           togglePin(BOARD_LED_GPIO_RED, 1u << BOARD_LED_GPIO_PIN_RED);</pre>
       }
  }
56
```

Listing 4: Problem 5 FTM/PWM Setup

```
void pwm_setup() {
2
     ftm_config_t ftmInfo;
     ftm_chnl_pwm_signal_param_t ftmParam;
3
     ftmParam.chnlNumber = kFTM_Chnl_1;
5
     ftmParam.level = kFTM_HighTrue;
6
     ftmParam.dutyCyclePercent = 0;
     ftmParam.firstEdgeDelayPercent = OU;
8
     ftmParam.enableComplementary = false;
9
     ftmParam.enableDeadtime = false;
11
     FTM_GetDefaultConfig(&ftmInfo);
12
13
     FTM_Init(FTM3, &ftmInfo);
14
     FTM_SetupPwm(FTM3, &ftmParam, 1U, kFTM_EdgeAlignedPwm, 5000U,
        CLOCK_GetFreq(kCLOCK_BusClk));
16
     FTM_StartTimer(FTM3, kFTM_SystemClock);
17
     ftmParam.chnlNumber = kFTM_Chnl_4;
     ftmParam.level = kFTM_HighTrue;
19
     ftmParam.dutyCyclePercent = 0;
20
     ftmParam.firstEdgeDelayPercent = OU;
21
22
     ftmParam.enableComplementary = false;
     ftmParam.enableDeadtime = false;
23
24
     FTM_GetDefaultConfig(&ftmInfo);
26
     FTM_Init(FTM3, &ftmInfo);
27
     FTM_SetupPwm(FTM3, &ftmParam, 1U, kFTM_EdgeAlignedPwm, 5000U,
28
        CLOCK_GetFreq(kCLOCK_BusClk));
     FTM_StartTimer(FTM3, kFTM_SystemClock);
29
```

```
ftmParam.chnlNumber = kFTM_Chnl_5;
31
     ftmParam.level = kFTM_HighTrue;
32
     ftmParam.dutyCyclePercent = 0;
33
     ftmParam.firstEdgeDelayPercent = OU;
34
     ftmParam.enableComplementary = false;
35
36
     ftmParam.enableDeadtime = false;
37
     FTM_GetDefaultConfig(&ftmInfo);
38
39
     FTM_Init(FTM3, &ftmInfo);
40
     FTM_SetupPwm(FTM3, &ftmParam, 1U, kFTM_EdgeAlignedPwm, 5000U,
41
        CLOCK_GetFreq(kCLOCK_BusClk));
     FTM_StartTimer(FTM3, kFTM_SystemClock);
42
  }
43
```

Listing 5: Problem 5 main

```
int main(void) {
2
     unsigned int duty_cycle_red = 0;
     unsigned int duty_cycle_green = 0;
3
     unsigned int duty_cycle_blue = 0;
4
     /* Init board hardware. */
6
     BOARD_InitBootPins();
     BOARD_InitBootClocks();
     BOARD_InitDebugConsole();
     pwm_setup();
11
12
13
     scanf("%2x%2x%2x", &duty_cycle_red, &duty_cycle_green, &duty_cycle_blue)
14
     printf("red = %x\n", duty_cycle_red);
     printf("green = %x\n", duty_cycle_green);
16
     printf("blue = %x\n", duty_cycle_blue);
17
18
     float red = (duty_cycle_red / 255.0) * 100;
19
     float green = (duty_cycle_green / 255.0) * 100;
20
     float blue = (duty_cycle_blue / 255.0) * 100;
21
22
     FTM_UpdatePwmDutycycle(FTM3, kFTM_Chnl_1, kFTM_EdgeAlignedPwm, (uint8_t)
23
        red):
     FTM_SetSoftwareTrigger(FTM3, true);
25
     FTM_UpdatePwmDutycycle(FTM3, kFTM_Chnl_5, kFTM_EdgeAlignedPwm, (uint8_t)
26
        green);
     FTM_SetSoftwareTrigger(FTM3, true);
27
28
     FTM_UpdatePwmDutycycle(FTM3, kFTM_Chnl_4, kFTM_EdgeAlignedPwm, (uint8_t)
29
        blue);
     FTM_SetSoftwareTrigger(FTM3, true);
30
31
     while (1) {
32
```

33 } 34 }