

# COMPENG 4DS4 Lab 0 Report

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

January 26, 2023

Listing 1: Problem 1

```

1  /* Definitions */
2  #define MEM_LOC_CHAR(x)      *((char*)x)
3  #define MEM_LOC_SHORT(x)    *((short*)x)
4  #define MEM_LOC_INT(x)      *((int*)x)
5  #define LOC1                 MEM_LOC_CHAR(0x20001000)
6  #define LOC2                 MEM_LOC_INT(0x20001001)
7  #define LOC3                 MEM_LOC_SHORT(0x20001005)
8  #define LOC4                 MEM_LOC_INT(0x20001007)
9  #define ARBITRARY_LOCATION   *((int*) 0x20001004)
10
11 /* Problem 1 Function */
12 void problem1() {
13     LOC1 = 0xAC;
14     LOC2 = 0xAABBCCDD;
15     LOC3 = 0xABCD;
16     LOC4 = 0xAABBCCDD;
17 }

```

Listing 2: Problem 3

```

1  /* Definitions */
2  #define BOARD_LED_GPIO_BLUE    GPIOC
3  #define BOARD_LED_GPIO_PIN_BLUE 8
4  #define BOARD_LED_GPIO_GREEN  GPIOC
5  #define BOARD_LED_GPIO_PIN_GREEN 9
6  #define BOARD_LED_GPIO_RED     GPIOD
7  #define BOARD_LED_GPIO_PIN_RED  1
8
9  /* Problem 3 main Function */
10 int main(void) {
11     /* Define the init structure for the output LED pin*/
12     gpio_pin_config_t led_config = {kGPIO_DigitalOutput, 0};
13
14     /* Board pin, clock, debug console init */
15     BOARD_InitBootPins();
16     BOARD_InitBootClocks();
17     BOARD_InitDebugConsole();
18
19     /* Print a note to terminal. */
20     PRINTF("\r\n GPIO Driver example\r\n");
21     PRINTF("\r\n The LED is blinking.\r\n");
22
23     /* Init output LED GPIO. */
24     GPIO_PinInit(BOARD_LED_GPIO_BLUE, BOARD_LED_GPIO_PIN_BLUE, &led_config);
25     GPIO_PinInit(BOARD_LED_GPIO_GREEN, BOARD_LED_GPIO_PIN_GREEN, &led_config);
26     GPIO_PinInit(BOARD_LED_GPIO_RED, BOARD_LED_GPIO_PIN_RED, &led_config);
27
28     while (1) {
29         delay();
30         GPIO_PortToggle(BOARD_LED_GPIO_BLUE, 1u << BOARD_LED_GPIO_PIN_BLUE);

```

```

31     delay();
32     GPIO_PortToggle(BOARD_LED_GPIO_BLUE, 1u << BOARD_LED_GPIO_PIN_BLUE
33     );
34     delay();
35     GPIO_PortToggle(BOARD_LED_GPIO_GREEN, 1u <<
36     BOARD_LED_GPIO_PIN_GREEN);
37     delay();
38     GPIO_PortToggle(BOARD_LED_GPIO_GREEN, 1u <<
39     BOARD_LED_GPIO_PIN_GREEN);
40     delay();
41     GPIO_PortToggle(BOARD_LED_GPIO_RED, 1u << BOARD_LED_GPIO_PIN_RED);
42     delay();
43     GPIO_PortToggle(BOARD_LED_GPIO_RED, 1u << BOARD_LED_GPIO_PIN_RED);
44 }
45 }

```

Listing 3: Problem 4

```

1  /* Definitions */
2  #define GPIOA ((GPIO_Struct*) 0x400FF000)
3  #define GPIOB ((GPIO_Struct*) 0x400FF040)
4  #define GPIOC ((GPIO_Struct*) 0x400FF080)
5  #define GPIOD ((GPIO_Struct*) 0x400FF0C0)
6  #define GPIOE ((GPIO_Struct*) 0x400FF100)
7
8  /* Problem 4 Helper Functions */
9  void initPin(GPIO_Struct *port, uint32_t pin) {
10     writePin(port, pin, 1);
11     port->GPIO_PDDR |= (1UL << pin);
12 }
13
14 void togglePin(GPIO_Struct *port, uint32_t mask) {
15     port->GPIO_PDOR ^= (mask);
16 }
17
18 void writePin(GPIO_Struct *port, uint32_t pin, uint8_t output) {
19     if (output == 0U) {
20         port->GPIO_PDOR |= (1UL << pin);
21     } else {
22         port->GPIO_PDOR &= ~(1UL << pin);
23     }
24 }
25
26 /* Problem 4 main Function */
27 int main(void) {
28     /* Board pin, clock, debug console init */
29     BOARD_InitBootPins();
30     BOARD_InitBootClocks();
31     BOARD_InitDebugConsole();
32
33     /* Print a note to terminal. */
34     PRINTF("\r\nGPIO Driver example\r\n");
35     PRINTF("\r\nThe LED is blinking.\r\n");
36 }

```

```

37  /* Init output LED GPIO. */
38  initPin(BOARD_LED_GPIO_BLUE, BOARD_LED_GPIO_PIN_BLUE);
39  initPin(BOARD_LED_GPIO_GREEN, BOARD_LED_GPIO_PIN_GREEN);
40  initPin(BOARD_LED_GPIO_RED, BOARD_LED_GPIO_PIN_RED);
41
42  while (1) {
43      delay();
44      togglePin(BOARD_LED_GPIO_BLUE, 1u << BOARD_LED_GPIO_PIN_BLUE);
45      delay();
46      togglePin(BOARD_LED_GPIO_BLUE, 1u << BOARD_LED_GPIO_PIN_BLUE);
47      delay();
48      togglePin(BOARD_LED_GPIO_GREEN, 1u << BOARD_LED_GPIO_PIN_GREEN);
49      delay();
50      togglePin(BOARD_LED_GPIO_GREEN, 1u << BOARD_LED_GPIO_PIN_GREEN);
51      delay();
52      togglePin(BOARD_LED_GPIO_RED, 1u << BOARD_LED_GPIO_PIN_RED);
53      delay();
54      togglePin(BOARD_LED_GPIO_RED, 1u << BOARD_LED_GPIO_PIN_RED);
55  }
56 }

```

Listing 4: Problem 5 FTM/PWM Setup

```

1  void pwm_setup() {
2      ftm_config_t ftmInfo;
3      ftm_chnl_pwm_signal_param_t ftmParam;
4
5      ftmParam.chnlNumber = kFTM_Chnl_1;
6      ftmParam.level = kFTM_HighTrue;
7      ftmParam.dutyCyclePercent = 0;
8      ftmParam.firstEdgeDelayPercent = 0U;
9      ftmParam.enableComplementary = false;
10     ftmParam.enableDeadtime = false;
11
12     FTM_GetDefaultConfig(&ftmInfo);
13
14     FTM_Init(FTM3, &ftmInfo);
15     FTM_SetupPwm(FTM3, &ftmParam, 1U, kFTM_EdgeAlignedPwm, 5000U,
16         CLOCK_GetFreq(kCLOCK_BusClk));
17     FTM_StartTimer(FTM3, kFTM_SystemClock);
18
19     ftmParam.chnlNumber = kFTM_Chnl_4;
20     ftmParam.level = kFTM_HighTrue;
21     ftmParam.dutyCyclePercent = 0;
22     ftmParam.firstEdgeDelayPercent = 0U;
23     ftmParam.enableComplementary = false;
24     ftmParam.enableDeadtime = false;
25
26     FTM_GetDefaultConfig(&ftmInfo);
27
28     FTM_Init(FTM3, &ftmInfo);
29     FTM_SetupPwm(FTM3, &ftmParam, 1U, kFTM_EdgeAlignedPwm, 5000U,
30         CLOCK_GetFreq(kCLOCK_BusClk));
31     FTM_StartTimer(FTM3, kFTM_SystemClock);

```

```

30
31     ftmParam.chnlNumber = kFTM_Chnl_5;
32     ftmParam.level = kFTM_HighTrue;
33     ftmParam.dutyCyclePercent = 0;
34     ftmParam.firstEdgeDelayPercent = 0U;
35     ftmParam.enableComplementary = false;
36     ftmParam.enableDeadtime = false;
37
38     FTM_GetDefaultConfig(&ftmInfo);
39
40     FTM_Init(FTM3, &ftmInfo);
41     FTM_SetupPwm(FTM3, &ftmParam, 1U, kFTM_EdgeAlignedPwm, 5000U,
42         CLOCK_GetFreq(kCLOCK_BusClk));
43     FTM_StartTimer(FTM3, kFTM_SystemClock);
44 }

```

Listing 5: Problem 5 main

```

1  int main(void) {
2      unsigned int duty_cycle_red = 0;
3      unsigned int duty_cycle_green = 0;
4      unsigned int duty_cycle_blue = 0;
5
6      /* Init board hardware. */
7      BOARD_InitBootPins();
8      BOARD_InitBootClocks();
9      BOARD_InitDebugConsole();
10
11     pwm_setup();
12
13     scanf("%2x%2x%2x", &duty_cycle_red, &duty_cycle_green, &duty_cycle_blue)
14         ;
15
16     printf("red = %x\n", duty_cycle_red);
17     printf("green = %x\n", duty_cycle_green);
18     printf("blue = %x\n", duty_cycle_blue);
19
20     float red = (duty_cycle_red / 255.0) * 100;
21     float green = (duty_cycle_green / 255.0) * 100;
22     float blue = (duty_cycle_blue / 255.0) * 100;
23
24     FTM_UpdatePwmDutycycle(FTM3, kFTM_Chnl_1, kFTM_EdgeAlignedPwm, (uint8_t)
25         red);
26     FTM_SetSoftwareTrigger(FTM3, true);
27
28     FTM_UpdatePwmDutycycle(FTM3, kFTM_Chnl_5, kFTM_EdgeAlignedPwm, (uint8_t)
29         green);
30     FTM_SetSoftwareTrigger(FTM3, true);
31
32     FTM_UpdatePwmDutycycle(FTM3, kFTM_Chnl_4, kFTM_EdgeAlignedPwm, (uint8_t)
33         blue);
34     FTM_SetSoftwareTrigger(FTM3, true);
35
36     while (1) {

```

```
33 }  
34 }
```