ESE 3025 Real-Time OS
Test #2
Lambton College in Toronto
instructor: Takis Zourntos

Name: Ronakkmar Bhailalbhai Sharma

Student ID: C0747019

Section: ____________________

please have your student ID cards displayed

For this test, you will compose a FreeRTOS application for the LPC1769 MCUXpresso eval board. Try to complete the test using only your memory rst. As this test is open-book, you can check your answers with actual code, but at least try to get things right on your own a few times. This will prepare you for any future job interviews in the embedded eld (and in the software area as well).

The FreeRTOS application is described as follows. We want to blink the Red, Green, and Blue LEDs in sequence, with no overlapping colours. In other words, your code should produce the following output: Red LED On, pause, Red LED O , pause, Green LED On, pause, Green LED O , pause, Blue LED On, pause, Blue LED O , then repeat.

Your application should use only one task function, passing in any information you need via pvParameters.

1. (10 points) Provide the code for your task function.

```
/* LED toggle thread */
static void vLED_ToggleTask(void *pvParameters) {

uint8_t LED_Number= *(uint8_t*)pvParameters ;


//If LED is Red(LED_Number 0) ,then Offset delay of 1 Second

if(LED_Number==0) {
vTaskDelay(configTICK_RATE_HZ);
}


//If LED is Green(LED_Number 1) ,then Offset delay of 2 Second

else if(LED_Number==1) {
vTaskDelay(2*configTICK_RATE_HZ);
}

//If LED is Blue(LED_Number 2) ,then Offset delay of 3 Second

else if(LED_Number==2) {
```

```c
vTaskDelay(3*configTICK_RATE_HZ);
}

        while (1) {

Board_LED_Set(LED_Number,LED_ON);       // LED on for 1 Second
vTaskDelay(configTICK_RATE_HZ);

Board_LED_Set(LED_Number,LED_OFF);      // LED off for 3.5 Second
vTaskDelay(3*configTICK_RATE_HZ+ configTICK_RATE_HZ/2);

}
}
```

2. (10 points) Provide the code for your main() function.

```c
int main(void)
{
prvSetupHardware();



/* RED toggle thread */
    xTaskCreate(vLED_ToggleTask, (signed char *) "vTaskLed1",
        configMINIMAL_STACK_SIZE, &RED_LED, (tskIDLE_PRIORITY + 3UL),
(xTaskHandle *) NULL);

/* GREEN toggle thread */
xTaskCreate(vLED_ToggleTask, (signed char *) "vTaskLed2",
configMINIMAL_STACK_SIZE, &GREEN_LED, (tskIDLE_PRIORITY + 2UL),
(xTaskHandle *) NULL);

/* BLUE toggle thread */
xTaskCreate(vLED_ToggleTask, (signed char *) "vTaskLed3",
configMINIMAL_STACK_SIZE, &BLUE_LED, (tskIDLE_PRIORITY + 1UL),
(xTaskHandle *) NULL);

/* Start the scheduler */
vTaskStartScheduler();

/* Should never arrive here */
return 1;
}
```

3. (10 points) In five sentences or fewer, describe the operation of your code in terms of the scheduler. What sort of policy is used by the scheduler?

- Based on RM policy I gave priority to Red, Green, Blue respectively
- As per assigning priority , vTaskStartscheduler() will start executing Red ,Green and Blue LED task
- By using inbuilt parameter configTICK-RATE-HZ I gave particular time of delay using vTask-Delay() API