Group 4, Team 1: Aw Rui Huan, Ong Yu Xuan, Hafidz

CG2271 lab 8 Report

Q1. Explain the THREE parameters that are passed when we call osSemaphoreNew().
The three parameters of osSemaphoreNew(max_count, intial_count, attr). max_count is the number maximum tokens, the initial_count is the initial number of tokens and attr is semaphore attribute.

Q2. Describe your observation. Explain why it is as such.
The LED does not light up. As the parameters passed to osSemaphoreNew() is 1,0,NULL, it means that it is starting with no tokens for the semaphore to acquire. Hence the threads cannot acquire the token to run. The switch does not affect the LEDs as well as it does not provide the token for the semaphores to acquire as well.

Q3. Describe your observation. Explain why it is as such.
Initially, the red led is in the off state as we initialized the osSemaphoreNew(1,0,NULL), initial count is 0. When the switch is pressed, the led_red_thread acquires the semaphore hence red led turns on and turns off. If the switch is pressed again, the green led turns on and off. If the switch is pressed quickly, sometimes yellow appears. When the switch is pressed, the semaphore is released, allowing for another thread to acquire the semaphore.

Q4. Provide the code for the UART ISR that handle the received data.

```
void UART2_IRQHandler(void) {
	NVIC_ClearPendingIRQ(UART2_IRQn);

	//if receive any data
	if (UART2->S1 & UART_S1_RDRF_MASK) {
		rx_data = UART2->D;
		osSemaphoreRelease(mySem);
	}
```

Q5. Explain the observation on the RGB LED every time you press the RED_LED_ON button.

If the RED_LED_ON button is pressed once, the red led will blink on and off. By pressing the RED_LED _ON button, it releases a semaphore. This semaphore will then be acquired by the red_led_thread, and it will proceed to blink by turning the red LED on and off, before looping and waiting for the next acquisition of the semaphore.

Q6. Demonstrate this functionality to your Lab TA. You must be able to send different commands from the App and decode the data in the UART IRQ handler. Subsequently, the appropriate Semaphore must be released to control the specific LED only. In this demo, you don't need to implement the LED_OFF buttons. Whenever the LED_ON button command is received, the appropriate LED must blink once. Provide the code for your UART IRQ handler.

```
void UART2_IRQHandler(void) {
	NVIC_ClearPendingIRQ(UART2_IRQn);

	//if receive any data
	if (UART2->S1 & UART_S1_RDRF_MASK) {
		rx_data = UART2->D;
		if(rx_data == LED_RED){
			if(BIT0_MASK(rx_data)){
				osSemaphoreRelease(myREDSem);
			}
		}
	}
		if(rx_data == LED_GREEN){
			if(BIT0_MASK(rx_data)){
				osSemaphoreRelease(myGREENSem);
			}
		}
	}
```