Kaustubh S. Jawalekar SJSU Id :- 012410944

Date:- 02/08/2018 **Assignment – 1:-Free RTOS Tasks** CMPE-244

Part 1:-

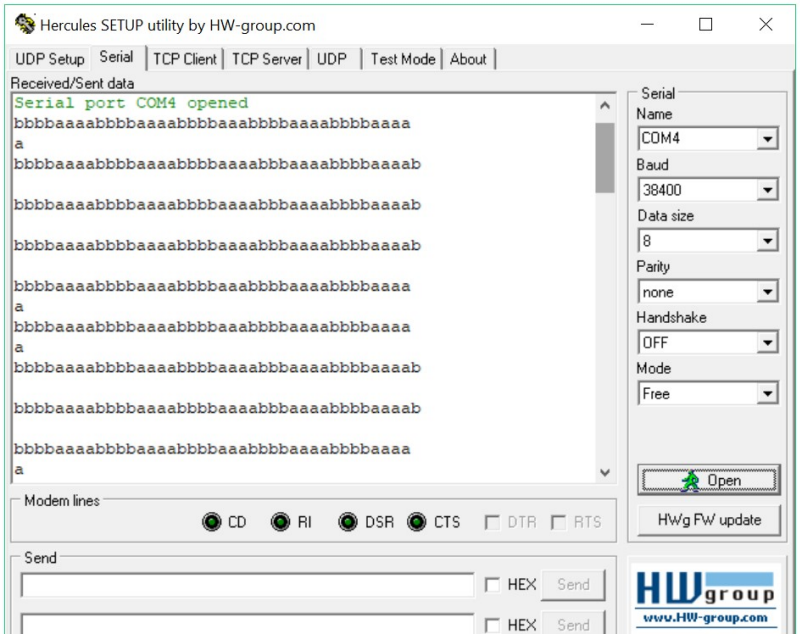
Task Create code:-

```
#include "FreeRTOS.h"
#include "task.h"
#include "uart0_min.h"

void vTaskOneCode(void *p)
{
    while(1)
    {
        uart0_puts("aaaaaaaaaaaaaaaaaaa");
        vTaskDelay(100);
    }
}

void vTaskTwoCode(void *p)
{
    while(1)
    {
        uart0_puts("bbbbbbbbbbbbbbbbbbb");
        vTaskDelay(100);
    }
}

int main(int argc, char const *argv[])
{
    const uint32_t STACK_SIZE = 1024;
    xTaskCreate(vTaskOneCode,(const char*)"task1",STACK_SIZE,0,1,NULL);
    xTaskCreate(vTaskTwoCode,(const char*)"task2",STACK_SIZE,0,1,NULL);
    vTaskStartScheduler();
    return 0;
}
```



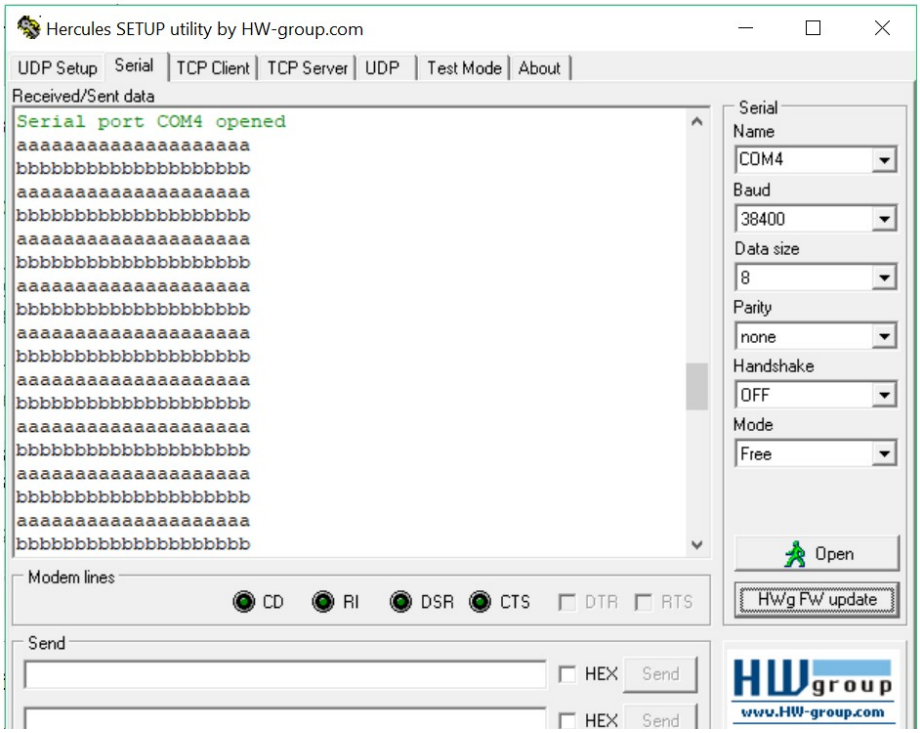
Part 2:-

1. When the 2 tasks in RTOS have same priorities, task2 enters in running state after 1 ms (1 clock tick) than task1. In 1ms, task1 can send and print 38.4 bits. Hence, sometimes 3 or sometimes 4 characters are sent and printed in 1 ms or 1 clock tick.
   Again, after 2 ms, task1 re-enters in running state and task2 exits after printing 3 or 4 characters.

   This occurs due to concurrent running of 2 tasks and sharing same processor core. Both tasks are rapidly entering and exiting the Running state. Both tasks are running at the same priority, and so share time on the same processor core.
2. When priorities are altered, the task with higher priority gets fully executed and then other task gets fully executed.
   Task 1 prints all 20 a's if task1 priority is higher and then task2 prints all 20 b's.
   Task 2 prints all 20 b's if task2 priority is higher and then task1 prints all 20 a's

Part 3:-

1. When task 1 priority is 2 and task 2 priority is 1,
   Task 1 is at high priority and executes first. It runs till the task 1 ends and prints whole data i.e. 20 a's.
   Then task 2 gets executed and prints 20 b's.



When task 2 priority is 1 and task 1 priority is 2,
Task 2 is at high priority and executes first. It runs till the task 2 ends and prints the whole data i.e. 20 b's.
Then task 1 gets executed and prints 20 a's.