**CG2271 Lab 6 Report**

**Q1. What is the purpose of the for-loop in the app\_main()?**

Most tasks are written in endless loops. The tasks state determines when they get to run.

**Q2. app\_main() is not explicitly called by main(). So when does app\_main() get executed?**

app\_main() gets executed the moment osThreadNew(app\_main, NULL, NULL) is called.

**Q3. At the point of time when osKernelStart() is called, what is the thread/task state of app\_main()?**

The thread/task state of the app\_main() is in the running state.

**Q4. When app\_main() calls osDelay(), what state does app\_main() transition into?**

The state of app\_main() goes to the blocked state

**Q5. Instead of the osDelay(), you decide to use the normal loop delay function that you had used in your earlier labs. Is there a difference?**

Yes, since currently there is only 1 thread. However, the use of osDelay() is to free up the CPU to do other important tasks. Normal delay is still making use of the CPU to execute code in order to generate the required delay.

**Q6. What are the changes you must make to rename a thread?**

The changes that need to be made are

1. osThreadNew(app\_main, NULL, NULL) -> osThreadNew(led\_red\_thread, NULL, NULL);
2. void app\_main (void \*argument) -> void led\_red\_thread (void \*argument)

**Q7. With both the threads being created in the main(), what is your observation? Explain why you make such an observation.**

We observed that either the red led or green led will light up only due to deadlock. Deadlock occurs in this scenario as there is no pre-emption since both the threads are of same priority. Hence, they are not able to be de-allocated or forcibly removed.

**Q8. Is your observation and your expectation the same?**

Yes.

**Q9. Explain your observation now. What was the significance of the change to the OS\_ROBIN\_ENABLE.**

Only 1 colour of LED either red or green is on. This is due to OS\_ROBIN\_ENABLE being set to 0 which means that RTX is unable to switch to tasks that are in READY state