Course: Real-Time Systems

Assignment 2

1) Create the two tasks in FreeRTOS.
   The first task should occupy the processor for a long time – you can create a task with a very long delay in the task code.
   Generally, this can be considered as a computation task (such as a large matrix multiplication task) on an embedded system, which occupies the processor for most of the time.
   There is another task which is a communication task, which needs to send data every time a certain number of computations/time duration elapses (say 100 computations) Please use the following code for communication task.

```
void task_comm()
{
        while (1) {
                printf("Sending data...\n");
                fflush(stdout);
                //Delay for 200 msec using for/while loop
                printf("Data sent!\n");
                fflush(stdout);
                //Delay for 200 msec using for/while loop
        }
}
```

   Please write the computation task and modify the communication task task_comm() to allow triggering of task_comm() based on a notification from the computation task. (Hint: Use Notifications).

2) There are 3 tasks as shown in Fig.8-7(b) of the textbook with priorities and execution times as shown. Please implement the priority inheritance protocol. (Hint: Use Notifications and Priority Changes)

Results Required:

1) Code for both the problems
2) Snapshot of the execution of tasks