

Resource Management through LED Display



November 24, 2018

sUBMITTED BY: Riya Sharan (2017H1120237P)

SONALI SHARMA(2017H1120239P)

DEEPTI KONDURI(2017H1120240P)

MONISHA NAIR(2017H1120241P)

# Title: **Resource Management through LED Display**

# Description of the application:

In this application we have focused on four main features for resource management between multiple threads. These four features have been described below:

1. **Serial Execution : Dependent threads**

In this application, serial execution of threads is shown in which each thread’s task is shown via blinking of LED’s in serial fashion. Here, the dependency between threads is taken into account in which the Red LED is dependent on Green LED so, in output Green LED will blink first and then Red LED.

1. **Parallel Execution : Independent threads**

In this application, parallel execution of threads is shown via blinking of LED’s. Here, independency between threads is taken into account in which both the LED’s are independent of each other and will blink parallel.

1. **Priority Based Scheduling of Threads : Execute threads in a predefined order.**

In this application, priority based scheduling of threads is shown in which resources are managed between tasks based on priority. The task with higher priority will get resources earlier and will execute first than task with lower priority.

1. **PriorityInversion : Executing lower priority tasks in the presence of higher priority tasks with help of binary semaphores.**

In this application, concept of priority inversion is introduced. Here, if semaphore is acquired by low priority task earlier than higher priority task then the low priority task will get the resources and will be executed. Later on after the semaphore will be released by low priority task the chance of higher priority task for execution will be considered.

# Assumptions and the flowchart:

With FreeRTOS

Without FreeRTOS

Serial Execution

With FreeRTOS and without using semaphore

With FreeRTOS and using binary semaphore

Priority Inversion

Priority Based Scheduling of Threads

Parallel Execution