Task set

I designed my system using 3 tasks. In this application the deaddline is assumed to be the same as periodicity.

Task	Execution time	Periodicity	Deadline
Task1	50ms	300ms	300ms
Task2	150ms	500ms	500ms
Task3	100ms	750ms	750ms

Hyperperiod

The hyperperiod is the least common multiple of the above tasks which is 1.5s

Simso

Here the chosen scheduler is EDF and with the task parameters as specified above

id	Name	Task type	Abort on miss	Act. Date (ms)	Period (ms)	List of Act. dates (ms)	Deadline (ms)	WCET (ms)	Followed by
1	TASK T1	Periodic	☑ Yes	0	300	-	300	50	
2	TASK T2	Periodic	☑ Yes	0	500	-	500	150	
3	TASK T3	Periodic	☑ Yes	0	750	-	750	100	

We get a cpu load of about 60% with each task occupying the %s shown

	Total load	Payload	System load
CPU 1	0.6000	0.6000	0.0000
Average	0.6000	0.6000	0.0000

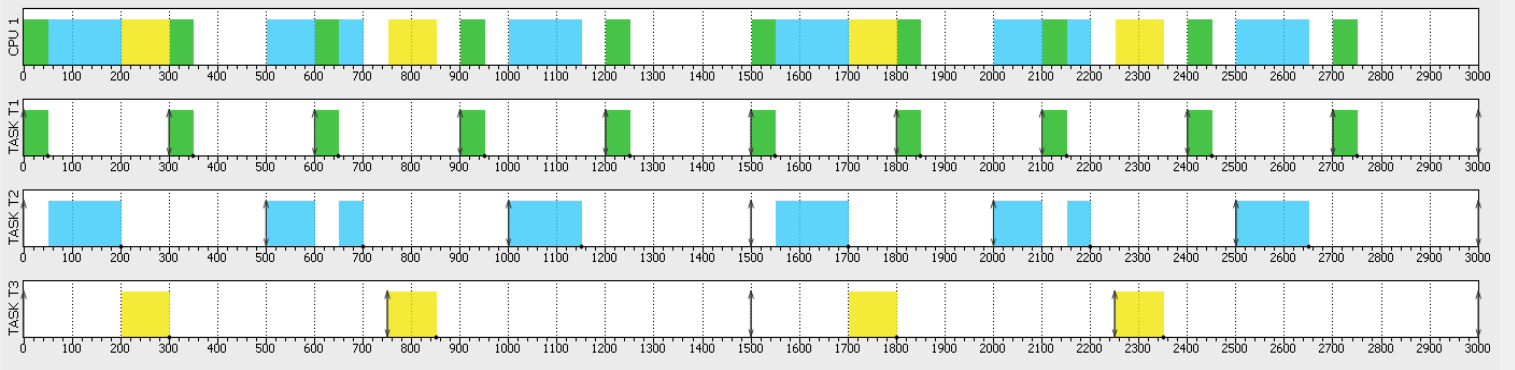
Task	min	avg	max	std dev	occupancy
TASK T1	50.000	50.000	50.000	0.000	0.167
TASK T2	150.000	150.000	150.000	0.000	0.300
TASK T3	100.000	100.000	100.000	0.000	0.133

If we calculate the cpu load manually,

For Task1 we have: (5 times per hyper * 0.05s execution time) = 0.25
For Task2 we have: (3 times per hyper * 0.15s execution time) = 0.45
For Task2 we have: (2 times per hyper * 0.1s execution time) = 0.2

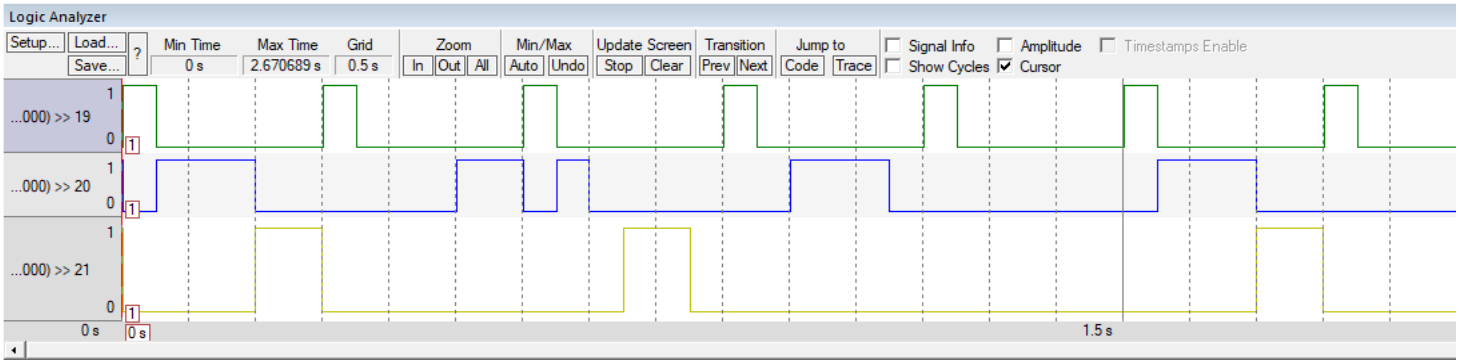
So total CPU load = Total load / hyperperiod = (0.25+0.45+0.2) / 1.5 = 0.9/1.5 = 0.6% CPU load
The same as calculated in the Simso

If we plot the Gantt chart, we can see that the system can be scheduled in the designed way with the edf scheduler as the cpu load is <1 and no task will miss its deadline as shown



Run time analysis

By using RTOS trace switch in and switch out features, I assigned a GPIO set an reset to each macro respectively and using the logic analyzer, we can see that in runtime the tasks are scheduled as they are designed in the offline simulator



And by using RTOS runtime anaylsis and printing it on uart, we get the following CPU load consumption of each task as shown which also matches the results we calculated from the offline simulator and manually