Project 2

Implementing EDF Scheduler

1-Calculate the system hyperperiod

Hyper Period = 100[ms].

2-Calculate the CPU load

- Task 1 -- > Button1
  Period : 50ms , deadline :50ms , execution Time : 1.9us
- Task 2 -- > Button2
  Period : 50ms , deadline :50ms , execution Time : 1.9us
- Task 3 -- > Periodic Transmitter
  Period : 100ms , deadline :100ms , execution Time : 6.67us
- Task 4 -- > Uart Receiver
  Period : 20ms , deadline :20ms , execution Time : 3.3167us
- Task 5 -- > Load 1
  Period : 10ms , deadline :10ms , execution Time : 5ms
- Task 6 -- > Load 2
  Period : 100ms , deadline :100ms , execution Time : 12ms

Utilization = Total Execution Time / Hyper Period .

$$\textbf{Utilization} = \frac{([2*1.9us] + [2*1.9us] + [1*6.67us] + [1*3.3167us] + [10*5000us] + [1*12000us])}{100000us} = 62\ \%$$

3-1- Check system schedulability Using URM technique

$$U=\left[\frac{1.6us}{50ms}\right]+\left[\frac{1.6us}{50ms}\right]+\left[\frac{6.67us}{100ms}\right]+\left[\frac{3.3167us}{20ms}\right]+\left[\frac{5\,ms}{10ms}\right]+\left[\frac{12ms}{100ms}\right]=.62$$

$$URM=\mathrm{n}(2^{\frac{1}{n}}-1)=0.756$$

$$U < URM$$

system is schedulable

3-2- Check system schedulability Using time demand analysis techniques

–Task 5: Load 1

$W_1(10)=5ms+0.$

$$\therefore w(10)<D=5<10$$

–Task 4:Uart_Receiver

$W2(20)=0.003167ms+[\frac{20}{10}*5ms].$

$$\therefore w(20)<D=10.003<20$$

- Task 1:Button_1

$$W_3(50)=1.9us+\left[\frac{50}{10}*5ms+\frac{50}{20}*3.3us\right]$$

$$\therefore w(50)<D=25.010<50$$

- Task 2:Button_2

$$W_4(50)=1.9us+\left[\frac{50}{10}*5ms+\frac{50}{20}*3.3us+\frac{50}{50}*1.9us\right]$$

$$\therefore w(50)<D=25.012<50$$

- Task 3:Periodic Transmitter

$$W5(100)=6.67us+[\frac{100}{10}*5ms+\frac{100}{20}*3.3us+\frac{100}{50}*1.9us+\frac{100}{50}*1.9us].$$

$$\therefore w(100) < D = 50.03 < 100$$

- Task 6 : Load 2

W6(100)=12ms +[$\frac{100}{10}$∗5ms+$\frac{100}{20}$∗3.3us + $\frac{100}{50}$∗∗1.9us + $\frac{100}{50}$∗∗1.9us +$\frac{100}{100}$∗∗6.67us].

$$\therefore w(100) < D = 62.03 < 100$$

system is schedulable

4. Using Simso offline simulator

The simulation is set to 1ms cycle duration and using Rate Monotonic

Fixed Priority. The Configuring the tasks to run:

After running the simulation, the result is as follow:

Model data

General | Scheduler | Processors | Tasks

id	Name	Task type	Abort on miss	Act. Date (ms)	Period (ms)	List of Act. dates (ms)	Deadline (ms)	WCET (ms)	Followed by
1	Load 1	Periodic	No	0	10.0	-	10.0	5	
2	Receiver	Periodic	No	0	50.0	-	50.0	0.02	
3	Button 1 Monitor	Periodic	No	0	50.0	-	50.0	0.02	
4	Button 2 Monitor	Periodic	No	0	50.0	-	50.0	0.02	
5	Periodic	Periodic	No	0	100	-	100	0.02	
6	Load 2	Periodic	No	0	100	-	100	12	

CPU Load

Results

General | Logs | Tasks | Scheduler

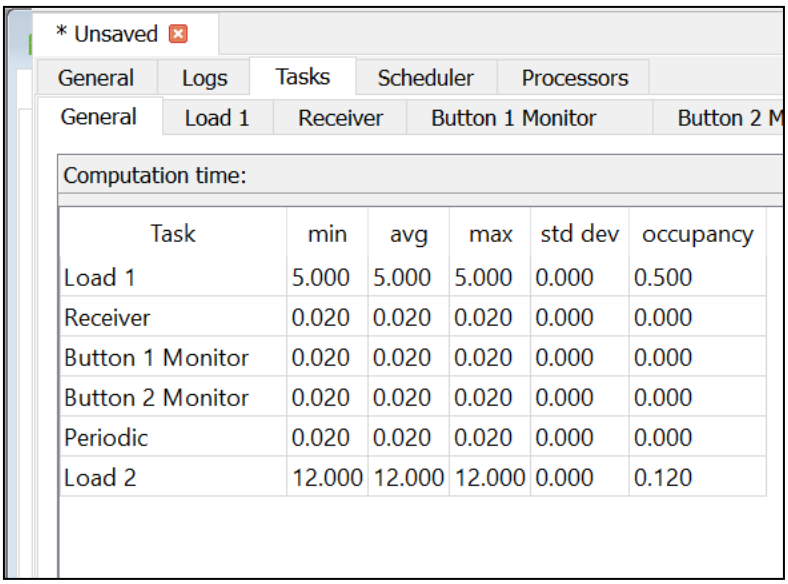
Observation Window:

from 0.00 to 100.00 ms | Configure...

	Total load	Payload	System loa
CPU 1	0.6214	0.6214	0.0000
Average	0.6214	0.6214	0.0000

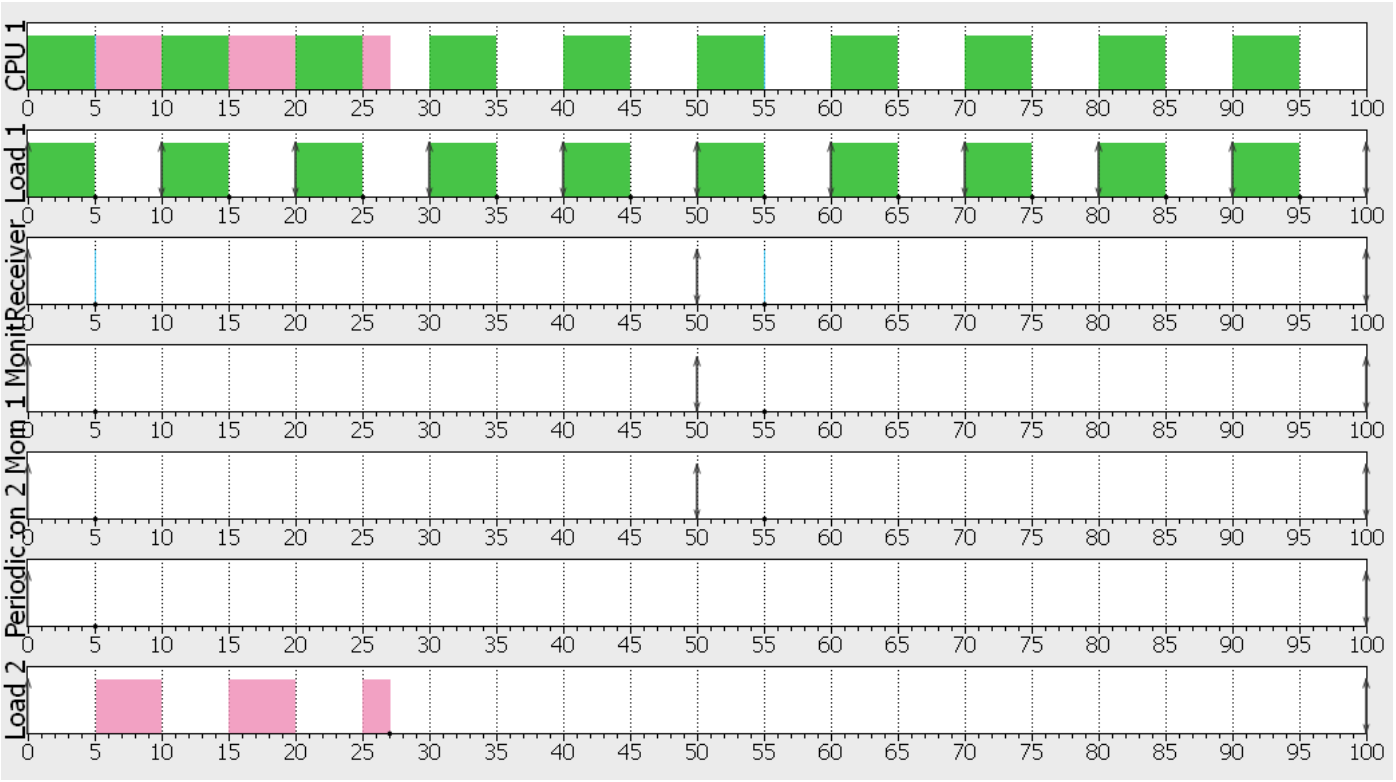
Which shows that CPU load is equal to 62.14% which is very close to mathematic calculation.

Tasks:



Task	min	avg	max	std dev	occupancy
Load 1	5.000	5.000	5.000	0.000	0.500
Receiver	0.020	0.020	0.020	0.000	0.000
Button 1 Monitor	0.020	0.020	0.020	0.000	0.000
Button 2 Monitor	0.020	0.020	0.020	0.000	0.000
Periodic	0.020	0.020	0.020	0.000	0.000
Load 2	12.000	12.000	12.000	0.000	0.120

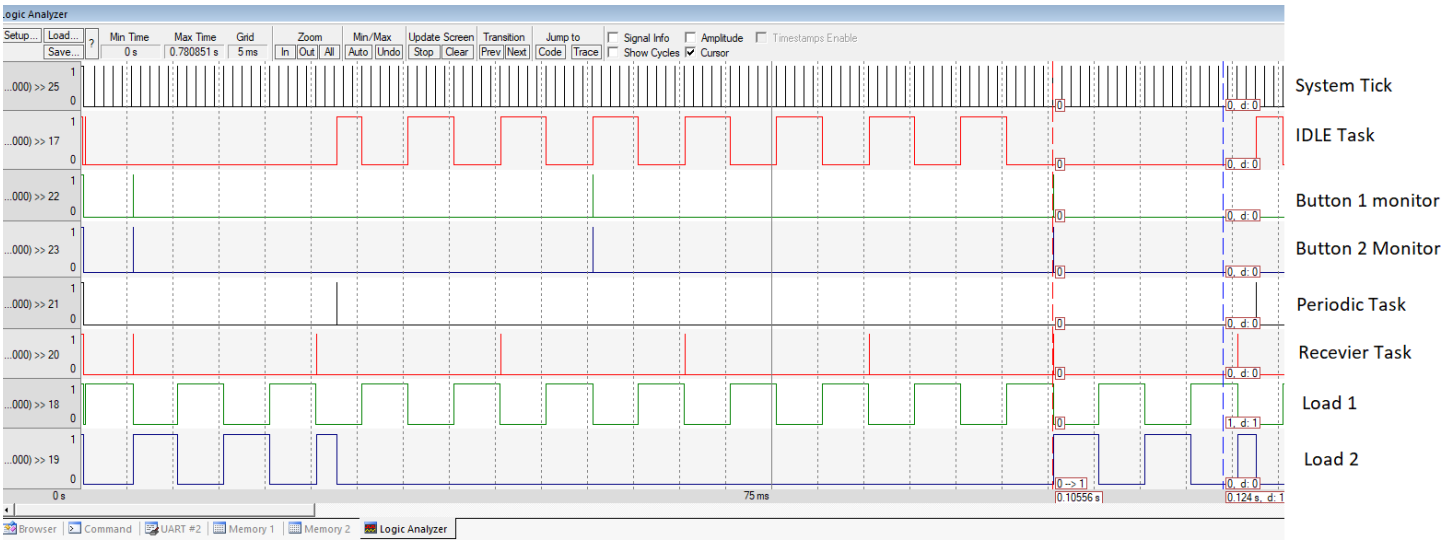
SimSo Gantt chart:



It shows that all tasks are executing correctly, and no task is missed.

Using Keil µVision Simulation

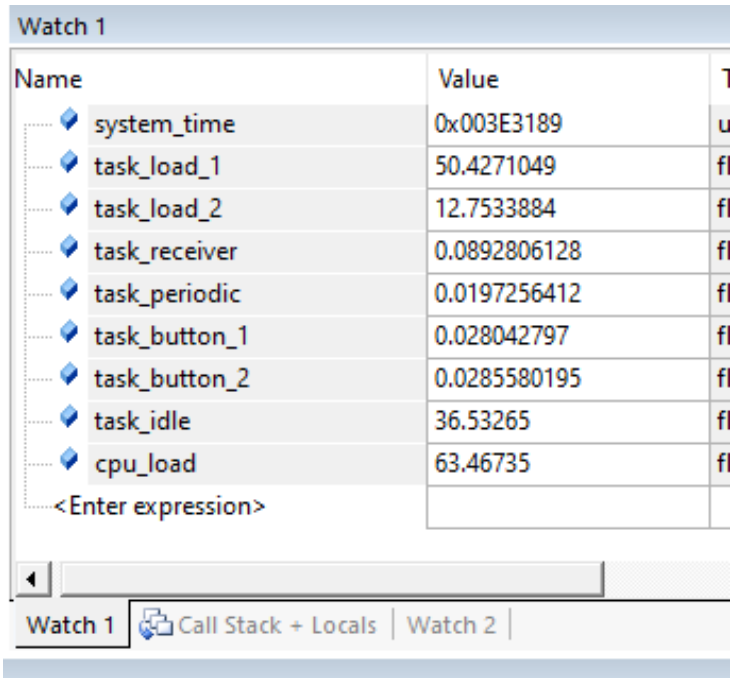
After Implementing EDF and then running the simulation and using logic analyzer:



It shows that all tasks are executed

same as Simso. And moving to Run-

time statistics shows:



That CPU Load is slightly higher than SimSo, and calculation which is logical due to neglecting context switching and FreeRTOS overhead mechanisms.