Embedded Systems Advanced Nano Degree
Real-Time Operating System Project
"Implementing EDF Scheduler Report"

Motaz Adel Mohamed Soliman

Motaz8413@gmail.com

Contents

1. Introduction

System implementation is tested using analytical techniques, an offline Simso simulator, and a run-time Keil simulator with the EDF (Earliest Deadline First) scheduler.

2. Analytical Methods

2.1. System Hyperperiod

Task	Periodicity
Button 1 Monitor	50
Button 2 Monitor	50
Periodic Tx	100
Uart Rx	20
Load 1 Simulation	10
Load 2 Simulation	100

Hyperperiod = Least Common Multiplier of tasks periodicities

Hyperperiod LCM(50,50,100,20,10,100) = 100

2.2. CPU Load

Task	Execution Time	Occurrence rate during hyperperiod
Button 1 Monitor	29 usec	2
Button 2 Monitor	29 usec	2
Periodic Tx	93 usec	1
Uart Rx	30 usec	5
Load 1 Simulation	5 msec	10
Load 2 Simulation	12 msec	1

Utilization = Total Execution Time During Hyperperiod / Hyperperiod

U =((29µ*2) + (29µ * 2) + (93µ) + (30µ + 5) + (5m + 10) + (12m)/100m) * 100% = 62.36%

2.3. System Schedulability

By using URM and time-demand analysis techs. "Assume the given tasks are scheduled using fixed priority rate-monotonic scheduler"

2.3.1. Rate Monotonic Utilization Bound

A system is schedulable if: U≤n ((2^(1/n))-1)

In our case:

U = 0.6236, Urm=6((2^(1/6))-1)= 0.7348

So U<Urm

The system is schedulable.

2.3.2. Time-Demand Analysis

Measure time required versus time provided:

$$w_i(t) = e_i + \sum_{k=1}^{i-1} \left\lceil \frac{t}{P_k} \right\rceil e_k$$

Where, w: worst response time
e: execution time
t: time instance
P: periodicity
i: task number

*Critical instant = 100 ms

Task	Periodicity	Execution Time
Load 1 Simulation	10	5 msec
Uart Rx	20	30 usec
Button 1 Monitor	50	29 usec
Button 2 Monitor	50	29 usec
Periodic Tx	100	93 usec
Load 2 Simulation	100	12 msec

1-For Task 1: Load 1 Simulation (E: 5 msec, P: 10 msec, Provided Time: 10 msec)

$w1(10) = 5m+0=5$, $w(10) = 5 < 10$

Task1 is schedulable.

2-For Task2: Uart Rx (E: 30 usec, P: 20 msec, Provided Time: 20 msec)

$w2\ (20) = 30\mu +(20/10)*5m=10.03$ msec, $w(20) = 10.03 <20$

Task2 is schedulable.

3-For Task3: Button 1 Monitor (E: 29 usec, P: 50 msec, Provided Time: 50 msec)

$w3\ (50) = 29\mu +(50/10) * 5m +(50/20)*30u= 25.059$ msec, $w(50)\ 25.059 < 50$

Task3 is schedulable.

4-For Task4: Button 2 Monitor (E: 29 usec, P: 50 msec, Provided Time: 50 msec)

$w1(50) = 29\mu +(50/10) * 5m +(50/20) * 30\mu +(50/50)*29\mu= 25.087$ msec

$w(50)\ 25.087 < 50$

Task4 is schedulable.

5-For Task5: Periodic Tx (E: 93 usec, P: 100 msec, Provided Time: 100 msec)

w5 (100)= 93µ+(100/10) *5m +(100/20)*30µ+(100/50)*29u+(100/50)*29u=50.359

w(10)= 50.359 <100

Task5 is schedulable

6-For Task6: Load 2 Simulation (E:12 msec, P: 100 msec, Provided Time: 100 msec)

w6(100)=12m+(100/10)*5m+(100/20)*30u+(100/50)*29u+(100/100)*93u=62.45

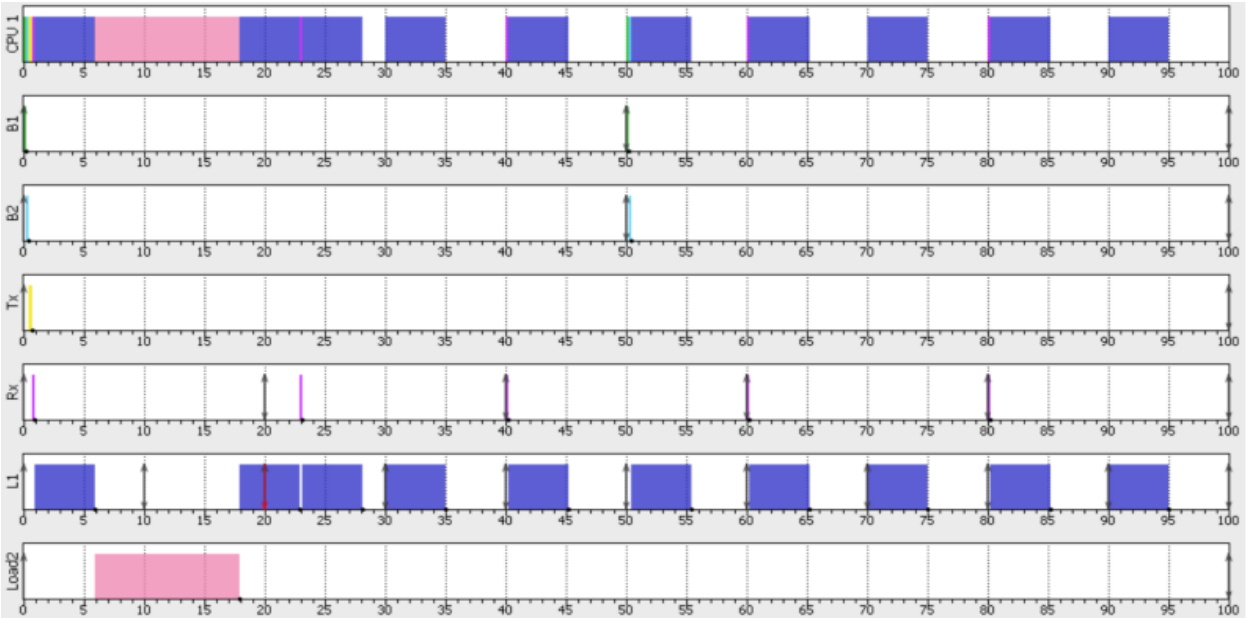
w(100)=62.452 < 100

Task6 is schedulable.

Therefore, The System Is Schedulable.

3. Simso Offline Simulator

Simulate the given tasks assuming fixed priority rate monotonic scheduler



4. Keil Simulator

Calculating the CPU timer usage using timer 1 and macros trace.

Where,

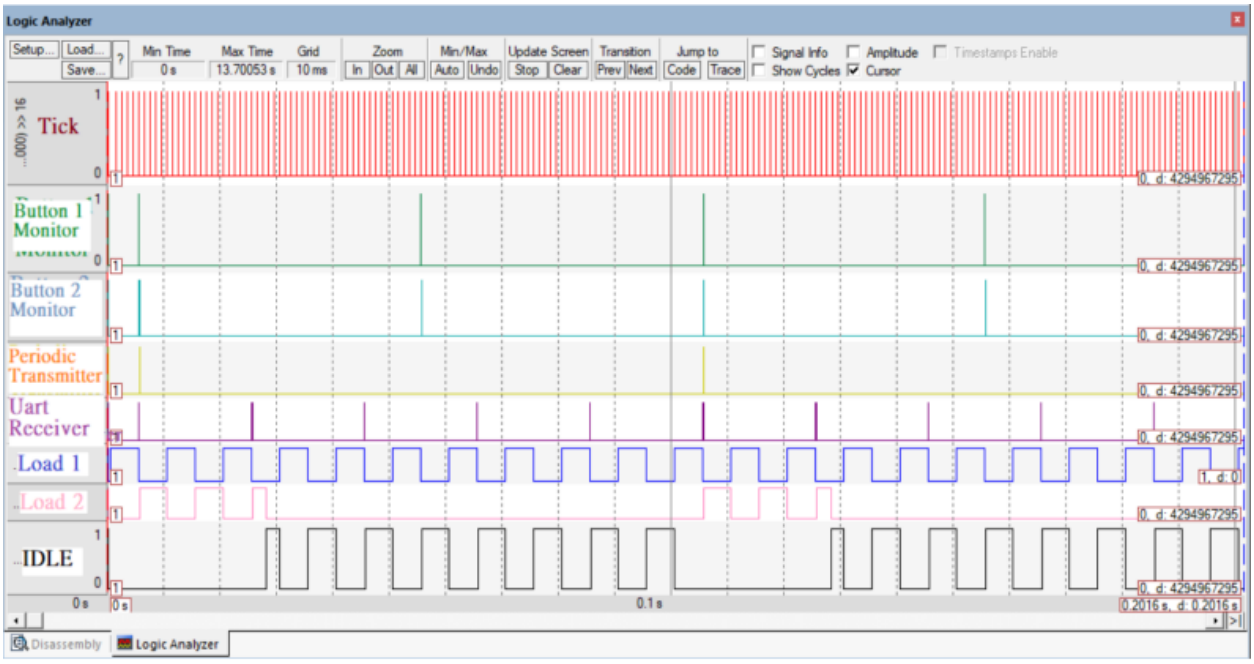
cpu_load: CPU load percentage
total_exe: total execution time for all tasks w.r.t. timer 1 ticks

T1TC: timer 1 ticks

Watch 1

Name	Value	Type
RTstats	0x00000000	uchar[300]
cpu_load	63	uint
total_exe	756275	uint
T1TC	1198526	ulong

Using macros trace and GPIO to plot the execution of all tasks, ticks, and idle task on logic analyzer.



5. Conclusion

1- Although using different verification strategies, final results tend to be the same.

2- EDF scheduler is a great scheduling policy for such tasks. As it keeps the system feasible.

3- Fixed priority rate monotonic scheduling policies do not keep the system feasible as tasks keep missing deadlines and there is not preemptive scheduling policy.