Implementing EDF Scheduler

Salma Zakaria Mohamed Ibrahim

21th Sep 2022

—

Embedded System Advanced Track

—

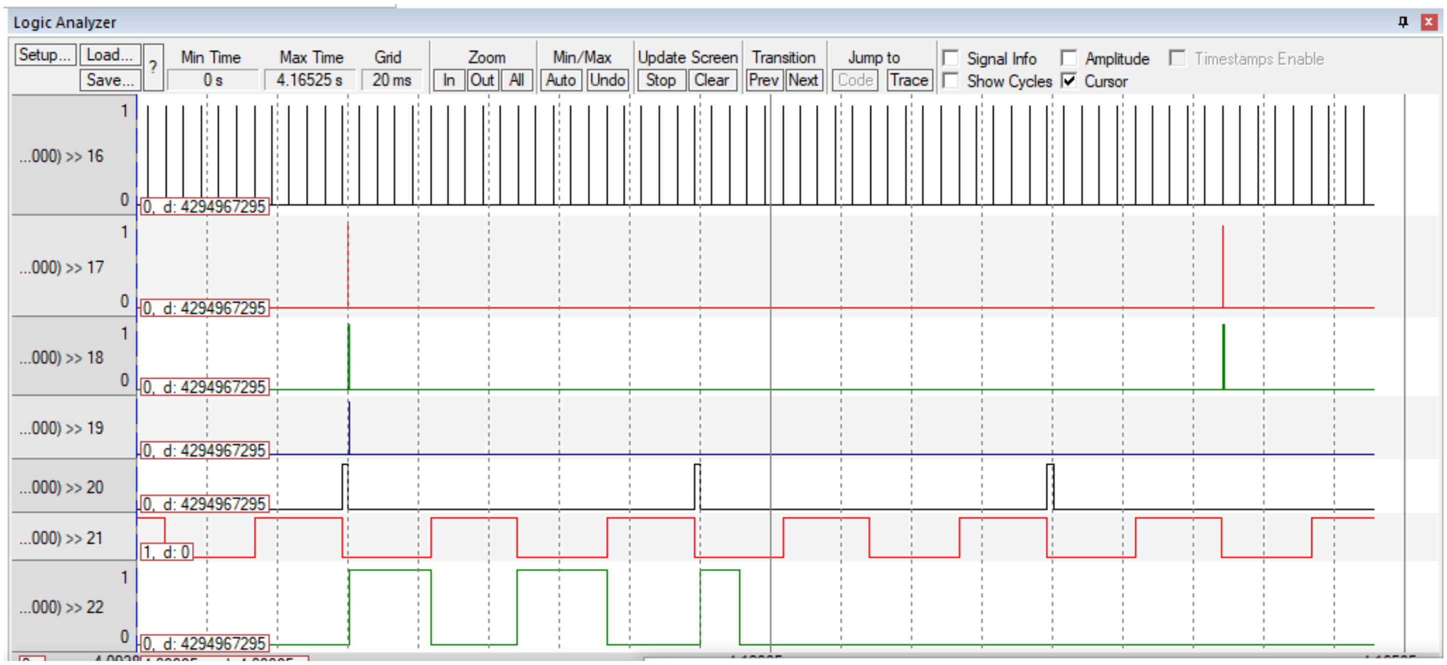
EGYPTfwd

System Validation

1. HyperPeriod
2. Offline Simulator Simso
3. CPU Load/Utilization
4. system schedulability using URM technique.
5. system schedulability Time Demand analysis technique.

1. HyperPeriod

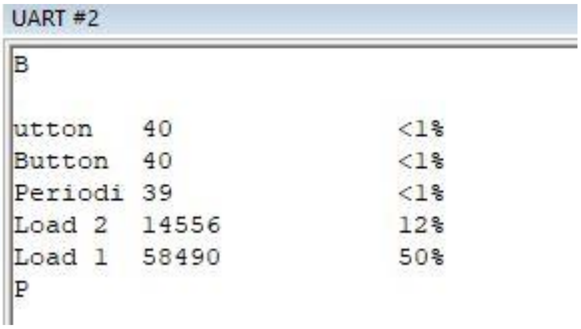
Assuming both period and deadline are the same.
Task Lists:



Task	*Period*	*Deadline*	*Execution(ms)*
B1	50	50	0.014767
B2	50	50	0.015033
Tx	100	100	0.029333
Rx	20	20	0.3528
L1	10	10	5
L2	100	100	12

Using Run-Time Stats API, we get the absolute time of each task through UART#2

Hyperperiod is the least common multiplier = 100ms

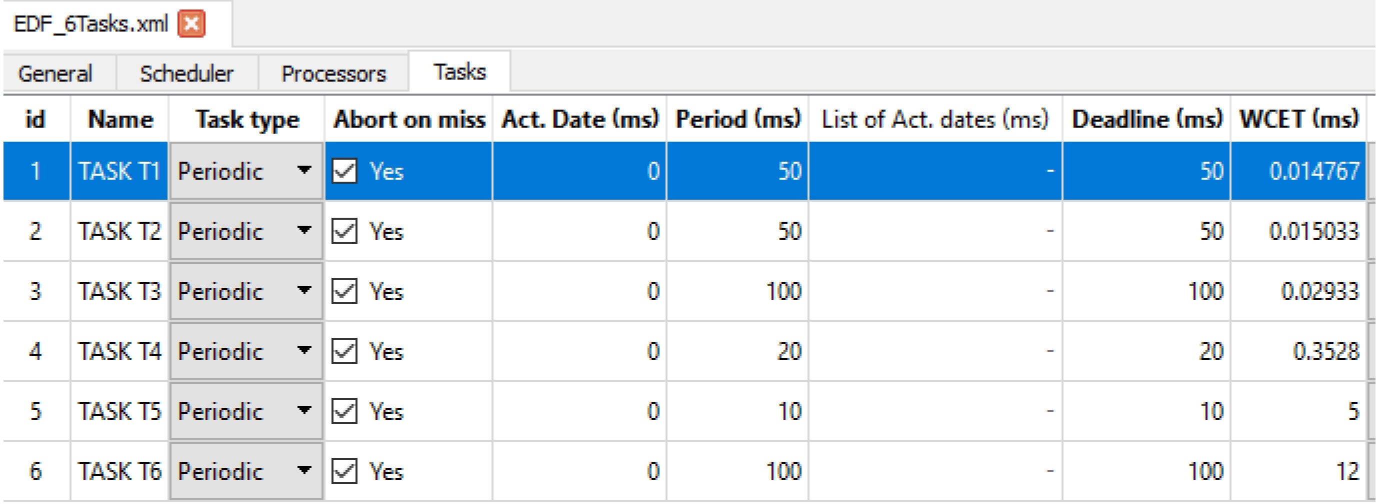


```
B

utton    40              <1%
Button   40              <1%
Periodi  39              <1%
Load 2   14556           12%
Load 1   58490           50%
P
```

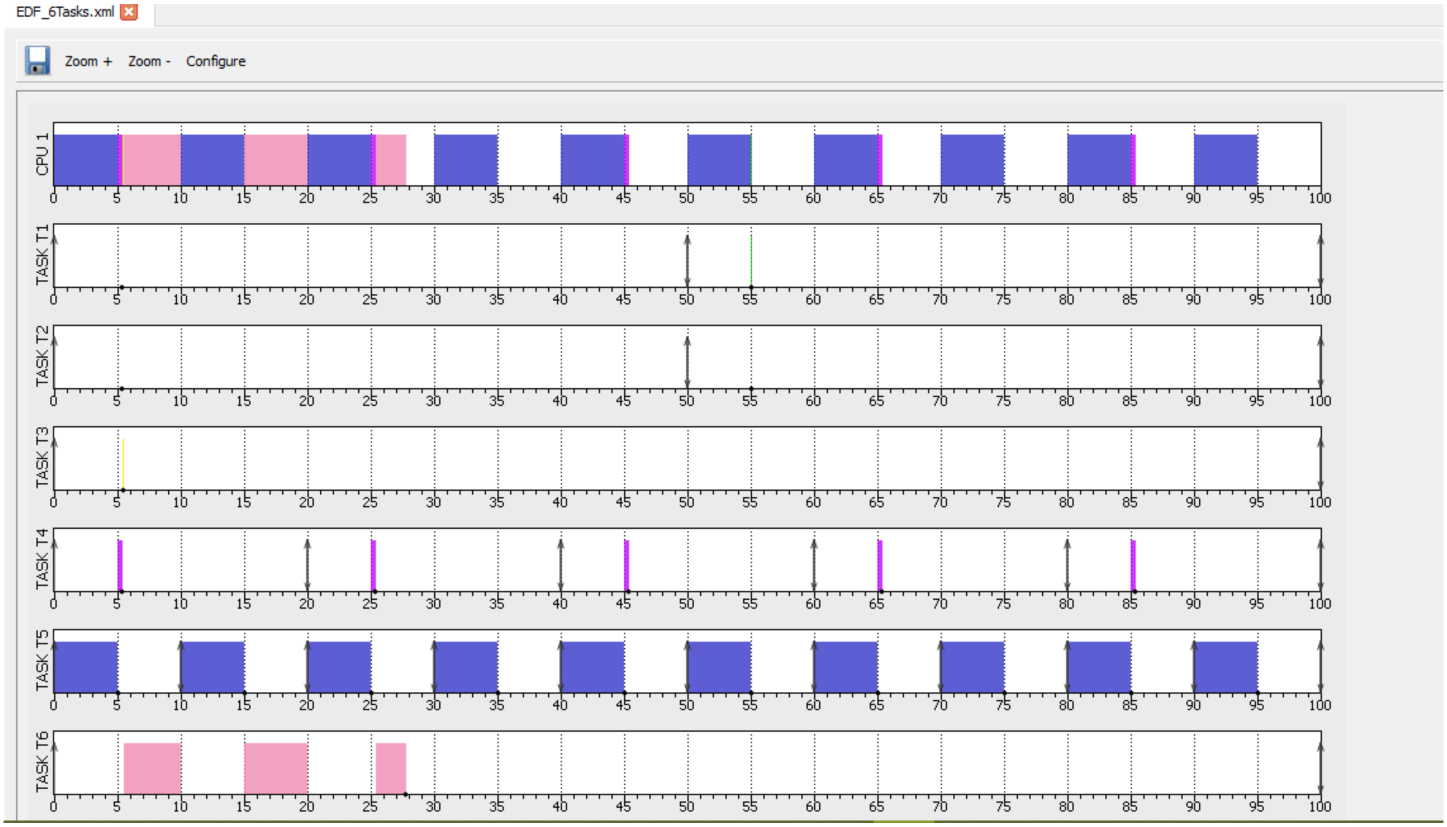
2. Offline Simulator Simso

Tasks:

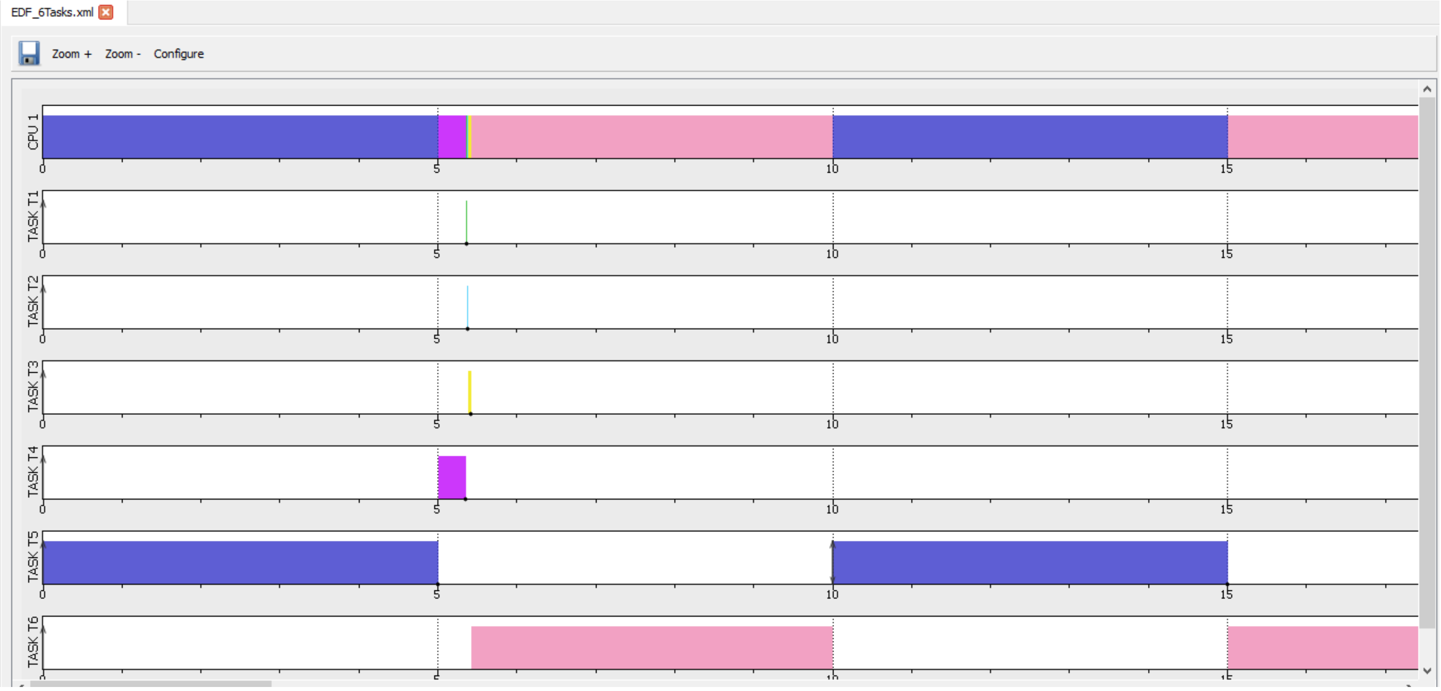


id	Name	Task type	Abort on miss	Act. Date (ms)	Period (ms)	List of Act. dates (ms)	Deadline (ms)	WCET (ms)
1	TASK T1	Periodic	☑ Yes	0	50	-	50	0.014767
2	TASK T2	Periodic	☑ Yes	0	50	-	50	0.015033
3	TASK T3	Periodic	☑ Yes	0	100	-	100	0.02933
4	TASK T4	Periodic	☑ Yes	0	20	-	20	0.3528
5	TASK T5	Periodic	☑ Yes	0	10	-	10	5
6	TASK T6	Periodic	☑ Yes	0	100	-	100	12

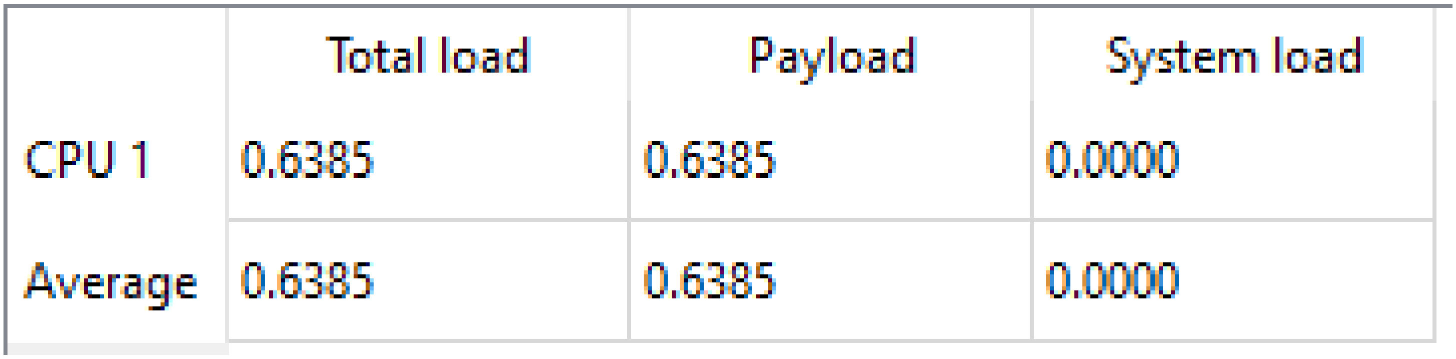
Results:



Zoomed In:



CPU Load: 63%



	Total load	Payload	System load
CPU 1	0.6385	0.6385	0.0000
Average	0.6385	0.6385	0.0000

3. CPU Load/Utilization

Using Trace hooks and timer:

U = Total Execution time / Hyperperiod
U = 63.85/100 = 0.6385 = 63.85 %

4. system schedulability using URM technique.

(Assuming the given set of tasks are scheduled using a fixed priority rate -monotonic scheduler)

$$\sum_{k=1}^{n} \frac{Ci}{Ti} \leq U = n\left(2^{1/n} - 1\right)$$

Figure 1 Rate monotonic Scheduling

U = 63.85 %
N “Number of tasks” = 6
63.85 < 73.47

Then the system is schedulable.

5. system schedulability Time Demand analysis technique.

$$w_i(t) = C_i + \sum_{k=1}^{i-1} \left\lceil \frac{t}{p_k} \right\rceil C_k$$

Ci : Execution time of ith task
- P : Periodicity
- Ui : Utilization of ith task
- wi : worst response time
- t : Current time point

The system is scheduable if the time demand for each task is less than the deadline of said task.

According to their priority,
Load 1 Task: T5(10,5,10)
W1(10) = 5 <10
Then its Schedulable

Rx Task: T4(20, 0.3528,20)
W2(20) = 0.3528 + 5 = 5.3528 < 20
Then its Schedulable

Button 2 Task : T2(50,0.015033,50)
W3(50) = 5 + 5.3528 + 0.015033 = 10.367833 < 50

Button 1 Task: T1(50, 0.014767,50)
W4(5) = 10.367833 + 0.014767 +5+5.3528= 20.735 < 50

Tx Task : T3(100, 0.029333,100)
W5(100) = 20.735 + 10.368 + 5.353+5 = 41.456 < 50

Load 2 Task: T6(100,12,100)
W6(100) = 41.456 + 20.735+10.368+5.352+5 = 82.911 < 100

Thus, the system is schedulable.