EDF Scheduler Project

Now we should verify the system implementation with the EDF scheduler using the following methods:

Task 1: ""Button_1_Monitor"", {Periodicity: 50, Deadline: 50},}, Execution time: 2ms

This task will monitor rising and falling edge on button 1 and send this event to the consumer task. (Note: The rising and failling edges are treated as separate events, hence they have separate strings)

Task 2: ""Button_2_Monitor"", {Periodicity: 50, Deadline: 50},}, Execution time: 2ms

This task will monitor rising and falling edge on button 2 and send this event to the consumer task. (Note: The rising and failling edges are treated as separate events, hence they have separate strings)

Task 3: ""Periodic_Transmitter"", {Periodicity: 100, Deadline: 100},}, Execution time: 4ms

This task will send preiodic string every 100ms to the consumer task

Task 4: ""Uart_Receiver"", {Periodicity: 20, Deadline: 20} , }, Execution time: 4ms

This is the consumer task which will write on UART any received string from other tasks

Task 5: ""Load_1_Simulation"", {Periodicity: 10, Deadline: 10}, Execution time: 5ms

Task 6: ""Load_2_Simulation"", {Periodicity: 100, Deadline: 100}, Execution time: 12ms

"1. Using analytical methods:

1- Calculate the system hyperperiod :

Hyperperiod (H) = LCM(Pi),
Where (Pi) is all task periodicities

H = 100 ms

2- Calculate the CPU Load :

- U = 100% - (Percentage of time that is spent in idle task)
- U = R/C

- U = Utilization
- R = Requirements which in simple terms is the BUSY TIME
- C = Capacity which is simple terms is BUSY TIME + IDLE TIME

E1 = 0.0016 * 2 = 0.0032
E2 = 0.0016 * 2 = 0.0032
E3 = 0.0012 * 1 = 0.0012
E4 = 0.0039 * 5 = 0.0195
E5 = 5 * 10 = 50
E6 = 12 * 1 = 12

Cpu load (U) = (0.0032 + 0.0032 + 0.0012 + 0.0195 + 50 + 12) / 100 = 0.62

U = 0.62

3- **Check system schedulability using URM :**

$$U = \sum_{i=1}^{n} \frac{C_i}{P_i} \leq n(2^{\frac{1}{n}} - 1)$$

U = Total Utilization
C = Execution time
P = Periodicity
N = Number of tasks

Case 1 : overloaded tasks is not included " 4 tasks "

U = (0.0016 /50) + (0.0016 /50) + (0.0012 /100) + (0.0039 /20) =
U = 0. 000271

URM = (6) * (2^(1/6) - 1) = 0.736

U < URM ==========➔ **The System is schedulable.**

Case 2 : overloaded tasks included " 6 tasks "

U = (0.0016 /50) + (0.0016 /50) + (0.0012 /100) + (0.0039 /20) + (5/10) + (12/100) = 0.620271

URM = (6) * (2^(1/6) - 1) = 0.736

U < URM ==========➔ **The System is schedulable .**

4- **Check system schedulability using time demand analysis:**

$$w_i(t) = e_i + \sum_{k=1}^{i-1} \left\lceil \frac{t}{p_k} \right\rceil e_k \quad for\ 0 < t \leq p_i$$

W = Worst response time
E = Execution time
P = Periodicity
T = Time instance

T1{P:50, E: 0.0016, D: 50} , T2{P: 50, E: 0.0016, D: 50} , T3{P:100 ,E: 0.0012 ,D:100} ,

T4{P: 20, E: 0.0039, D: 20} , T5{P: 10 ,E:5 ,D: 10} , T6{P: 100 ,E:12 ,D: 100}

- Calculate time demand for T5:

- W1 = 5 + 0 = 5
- W2 = 5 + 0 = 5
- .....
- W10 = 5 + 0 = 5

W10 < D

T5 is Schedulable

- Calculate time demand for T4:
- W1 = 0. 0039 + 5 = 5. 0039
- W2 = 0. 0039+ 5 = 5. 0039
- .....
- W10 = 0. 0039 + 5= 5. 0039
- .....
- W20 = 0. 0039 + (5*2) = 10. 0039

W20 < D

T4 is Schedulable

- Calculate time demand for T1:
- W1 = 0. 0016 + 5 + 0. 0039 = 5. 0055
- W2 = = 0. 0016 + 5 + 0. 0039 = 5. 0055
- .....
- W50 = 0. 0016 + 25 + 0. 0117 = 25.0135

W50 < D

T1 is Schedulable

- Calculate time demand for T2:
- W1 = 0. 0016 + 5 + 0. 0039 = 5. 0055
- W2 = = 0. 0016 + 5 + 0. 0039 = 5. 0055
- .....
- W50 = 0. 0016 + 25 + 0. 0117 = 25.0135

W50 < D

T2 is Schedulable

- Calculate time demand for T3:
- W1 = = 0. 0012 + 5 + 0.0039 +0. 0016 + 0. 0016 = 5 .0083
- W2 = = 0. 0012 + 5 + 0.0039 +0. 0016 + 0. 0016 = 5 .0083
- .....
- W100 =0.0039+ 50 + 0.006 +0.0032 + 0.0032= 50.0163

W100 < D

T3 is Schedulable

- Calculate time demand for T6:
- W1 = 12 +0. 0012 + 5 + 0.0039 +0. 0016 + 0. 0016 = 17 .0083
- W2 = 12 + 0. 0012 + 5 + 0.0039 +0. 0016 + 0. 0016 = 17 .0083
- .....
- W100 = 12+0.0039+ 50 + 0.006 +0.0032 + 0.0032= 62.0163

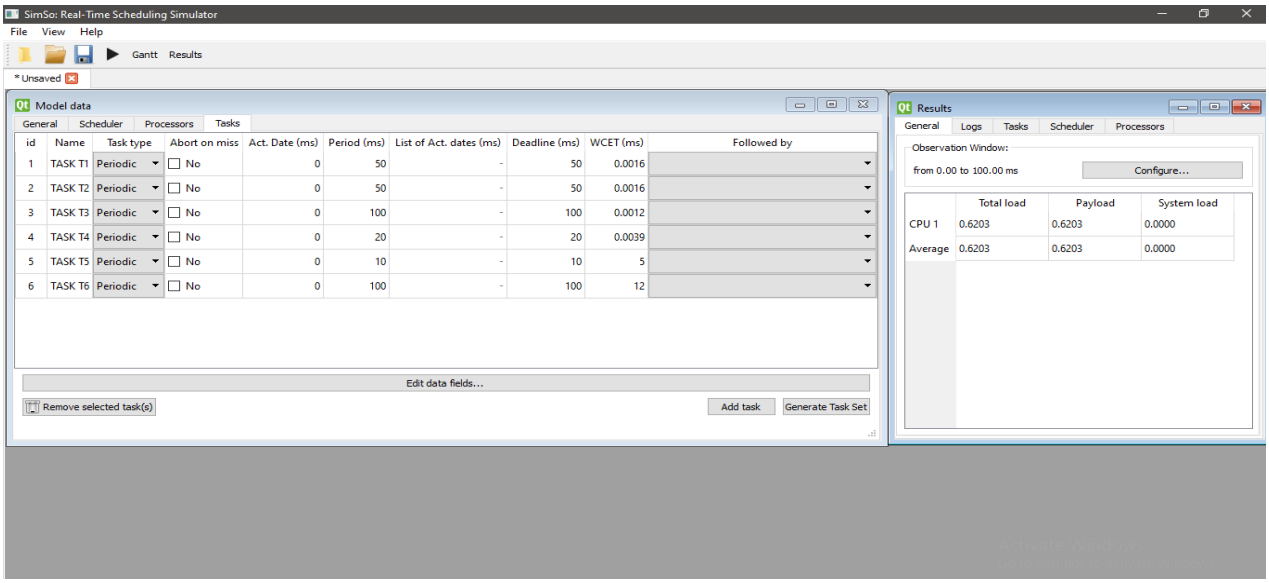
W100 < D

T6 is Schedulable

The system is Schedulable

2. Using Simso offline simulator:

1- Create tasks in simso and configure their parameters :

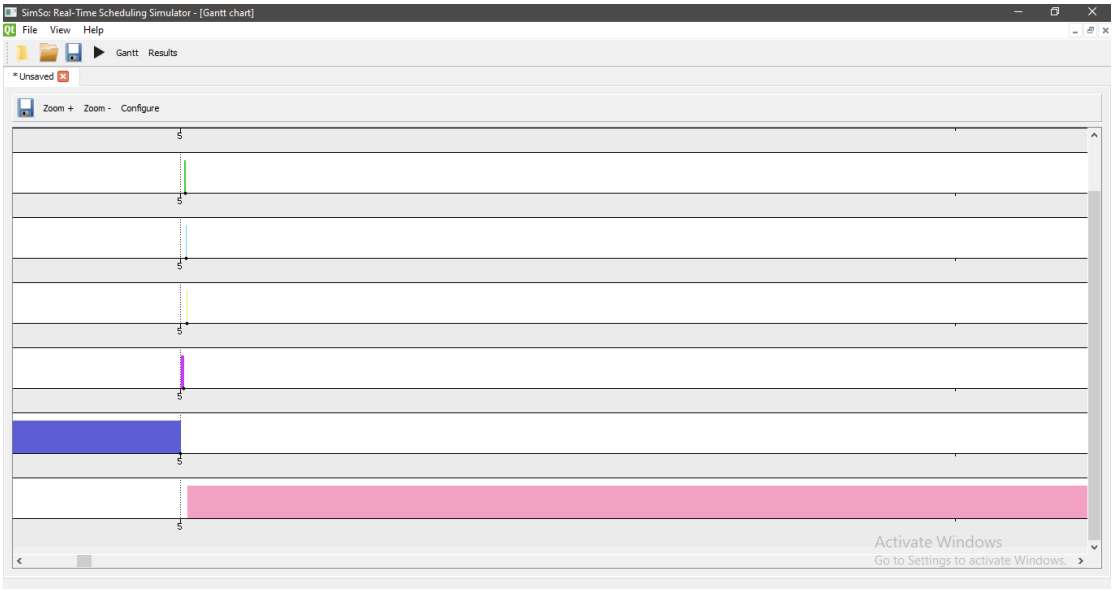


As expected from the analytical method and shown in the image above :

- **Hyper_Period H = 100**
- **CPU_Load U = 0.6203**

2- Run the simulator and show the time line of the Tasks :

To see the tasks 1 and 2 , we make zoom in as shown in the next image .



• The system is schedulable as all tasks in the image above doesn't miss the deadline .

"3. Using Keil simulator in run-time and the given set of tasks:

3- Calculate the CPU usage time using timer 1 and trace macros

U = 0.62

4- Using trace macros and GPIOs, plot the execution of all tasks, tick, and the idle task on the logic analyzer"