

# Project (2) Implementing EDF Scheduling

## 1 VERIFYING THE IMPLEMENTATION USING ANALYTICAL METHODS

### Calculating the System Hyperperiod

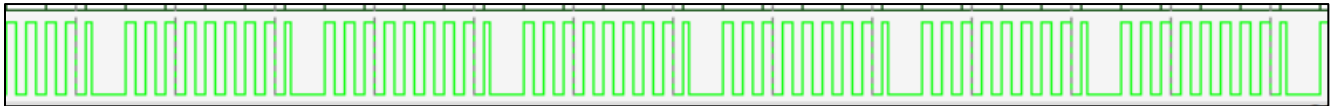
System hyperperiod = 100 msec

### Calculating the CPU load

#### Method (1):

The CPU load can be calculated from the idle task as follows:

- The time the idle task was called during the hyperperiod =  $1 * 3 + 8 * 5 = 43$  msec
- Idle task utilization =  $43/100 = 0.43$
- CPU utilization =  $1 - \text{Idle task utilization} = 57\%$



#### Method (2)

The CPU load is also calculated using the trace hook as shown below. It is found that it is as the one that is calculated analytically (in the previous section)

Name	Value	Type
task_1_totalTime	0x0000063D	int
task_2_totalTime	0x00000652	int
task_3_totalTime	0x00000592	int
task_4_totalTime	0x00000C8D	int
task_5_totalTime	0x00100CED	int
task_6_totalTime	0x00023F0B	int
cpuLoad	57.3954506	float
<Enter expression>		

#### Method (3)

Using the run-time analysis (shown below), it is found that the CPU load is approximately equals those calculated in the previous 2 methods

ART_Tx	4507	<1%
Load_2	44151	6%
IDLE	300417	42%
UART_Rx	1064	<1%
Button_1	299	<1%
Button_2	302	<1%
Load_1	350887	50%

## 2 CHECKING SYSTEM SCHEDULABILITY USING URM

The following table represents the execution time for each task calculated using the

Task Name	Execution time (msec)	Period (msec)	Task Utilization
T1 (button_1)	0.018	50	0.00036
T2 (button_2)	0.018	50	0.00036
T3 (UART_Tx)	0.015	100	0.00015
T4 (UART_Rx)	0.04	20	0.002
T5 (load_1)	5	10	0.5
T6 (Load_2)	12	100	0.12
Total Tasks utilization			0.623

Table 1: Tasks properties

- Utilization bound =  $n(2^{1/n} - 1) = 0.735$
- $0.623 < 0.735 \rightarrow$  The system is schedulable

## 3 TIME DEMAND ANALYSIS

Referring to table 1, the time demand is calculated for each task as follows

- Task (5) (Load-1) (Highest Priority Task)

Deadline	10
The time provided (msec)	5
<b>(5 &lt; 10) → Task (5) is schedulable</b>	

03365

- Task (4) UART Receiving Task

Deadline	20
The time provided (msec)	$0.04 + (20/10) * 5 = 10.04$
<b>10.04 &lt; 20 → Task (4) is schedulable</b>	

- Task (1): button (1) task

The time required (msec)	50
The time provided (msec)	$0.018 + (50/10) * 5 + (50/20) * 0.04 = 25.118$
<b>25.118 &lt; 50 → Task (1) is schedulable</b>	

- Task (1): button (1) task

The time required (msec)	50
The time provided (msec)	$0.018 + (50/10) * 5 + (50/20) * 0.04 + 0.018 = 25.136$
<b>25.136 &lt; 50 → Task (2) is schedulable</b>	

#### 5. Task (3): UART Transmitter

The time required (msec)	100
The time provided (msec)	$0.015 + (100/10) * 5 + (100/20) * 0.04 + (100/50) * 0.018 + (100/50) * 0.018 = 50.287$
<b>50.287 &lt; 100 → Task (3) is schedulable</b>	

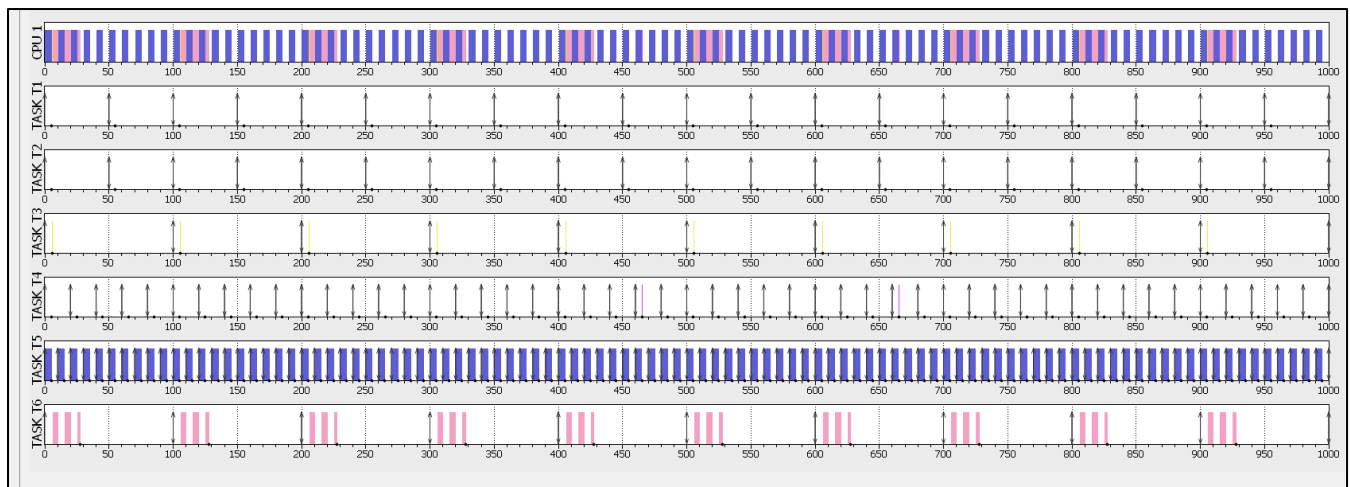
#### 6. Task (6): Load - 2

The time required (msec)	100
The time provided (msec)	$12 + (100/10) * 5 + (100/20) * 0.04 + (100/50) * 0.018 + (100/50) * 0.018 = 62.272$
<b>62.272 &lt; 100 → Task (6) is schedulable</b>	

It is concluded that the system is schedulable.

## 4 SIMSO

It is confirmed from the Simso that none of the tasks misses its deadline.



Name	Value	Type
task_1_totalTime	0x0000063D	int
task_2_totalTime	0x00000652	int
task_3_totalTime	0x00000592	int
task_4_totalTime	0x00000C8D	int
task_5_totalTime	0x00100CED	int
task_6_totalTime	0x00023F0B	int
cpuLoad	57.3954506	float
<Enter expression>		