

Real-Time Operating System Project

EDF Implementation Report

Name: Andrew Emeel Kamel

Email : androdod240@gmail.com

Verifying the System Implementation: Method 1: Analytically

using analytical methods:

1- System Hyper-period :

Task	Periodicity
Button 1 Monitor	50
Button 2 Monitor	50
Periodic Transmitter	100
UART Transmitter	20
Load 1 Simulation	10
Load 2 Simulation	100

Hyperperiod = Least Common Multiplier of all tasks
periodicities $\text{Hyperperiod} = LCM(50, 50, 100, 20, 10, 100)$
 $\text{Hyperperiod} = 100$

2- CPU Load:

Task	Execution Time	Occurrence During Hyper-period
Button 1 Monitor	18.5 μ s	2
Button 2 Monitor	19 μ s	2
Periodic Transmitter	19.2 μ s	1
UART Transmitter	21.6 μ s	5
Load 1 Simulation	5 ms	10
Load 2 Simulation	12 ms	1

$$\begin{aligned} \text{Utilization} &= \text{Total Execution Time During Hyper-period} / \\ &\text{Hyperperiod} U = ((T1*2)+(T2*2)+T3*1)+(T4*5)+T5*10)+(T6*1) \\ &/ 100\text{ms}) * 100\% \quad U \\ &= ((18,5\mu\text{s}*2)+(19\mu\text{s}*2)+(19.2\mu\text{s}*1)+(21,6\mu\text{s}*5)+(5\text{ms}*10)+(12 \\ &\text{ms}*1) / 100\text{ms}) * 100\% = 62.22\% \end{aligned}$$

3- System Schedulability: Schedulability Analysis (using URM and time demand analysis techniques: (Assuming the given set of tasks are scheduled using a fixed priority rate monotonic scheduler)

Using Rate Monotonic Utilization Bound :

$$U \leq n[2^{1/n} - 1] \quad U_{rm} = n(2^{1/n} - 1) = 6(2^{1/6} - 1) = 0.73477$$

$$U = \sum C_i/P_i = 18.5 \mu s / 50 ms + 19 \mu s / 50 ms + 19.2 \mu s / 100 ms + 21.6 \mu s / 20 ms + 5 ms / 10 ms + 12 ms / 100 ms = 0.62202$$

Since $U < U_{rm}$, Therefore System guaranteed Schedulable

Using Time Demand Analysis:

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

W = Worst response

E = Execution time

P = Periodicity

T = Time instance

Arrange Tasks according to Priority will be: [Task 5, Task 4, Task 1, Task 2, Task 3, Task 6]

Task	Task ID	Periodicity	Execution Time
Button 1 Monitor	Task 1	50	18.5 μ s
Button 2 Monitor	Task 2	50	19 μ s
Periodic Transmitter	Task 3	100	19.2 μ s
UART Transmitter	Task 4	20	21.6 μ s
Load 1 Simulation	Task 5	10	5 ms
Load 2 Simulation	Task 6	100	12 ms

For Task 5

(Load 1 Simulation)

$W(1) \dots\dots\dots W(10)$

$$W(1) = 5 + 0 = 5 \text{ ms} \diamond W(10) = 5 + 0 = 5 \text{ ms}$$

$W(10) < D = 5\text{ms} < 10 \text{ ms}$, Task 5 is Schedulable

For Task 4 (UART Transmitter)

$W(1) \dots\dots\dots W(20)$

$$W(1) = 21.6 \mu\text{s} + (1/10) * 5 \text{ ms} = 0.522 \text{ ms}$$

$$W(5) = 21.6 \mu\text{s} + (5/10) * 5 \text{ ms} = 2.5216 \text{ ms}$$

$$W(10) = 21.6 \mu s + (10/10) * 5 ms = 5.023 ms$$

$$W(20) = 21.6 \mu s + (20/10) * 5 ms = 10.021 ms$$

$W(20) < D = 10.021 ms < 20 ms$, Task 4 is Schedulable

For Task 1 (Button 1 Monitor)

$$W(1).....W(50)$$

$$W(1) = 18.5 \mu s + (1/10) * 5 ms + (1/20) * 21.6 \mu s ms = 0.519 ms$$

$$W(50) = 18.5 \mu s + (50/10) * 5 ms + (50/20) * 21.6 \mu s ms = 25.0725 ms$$

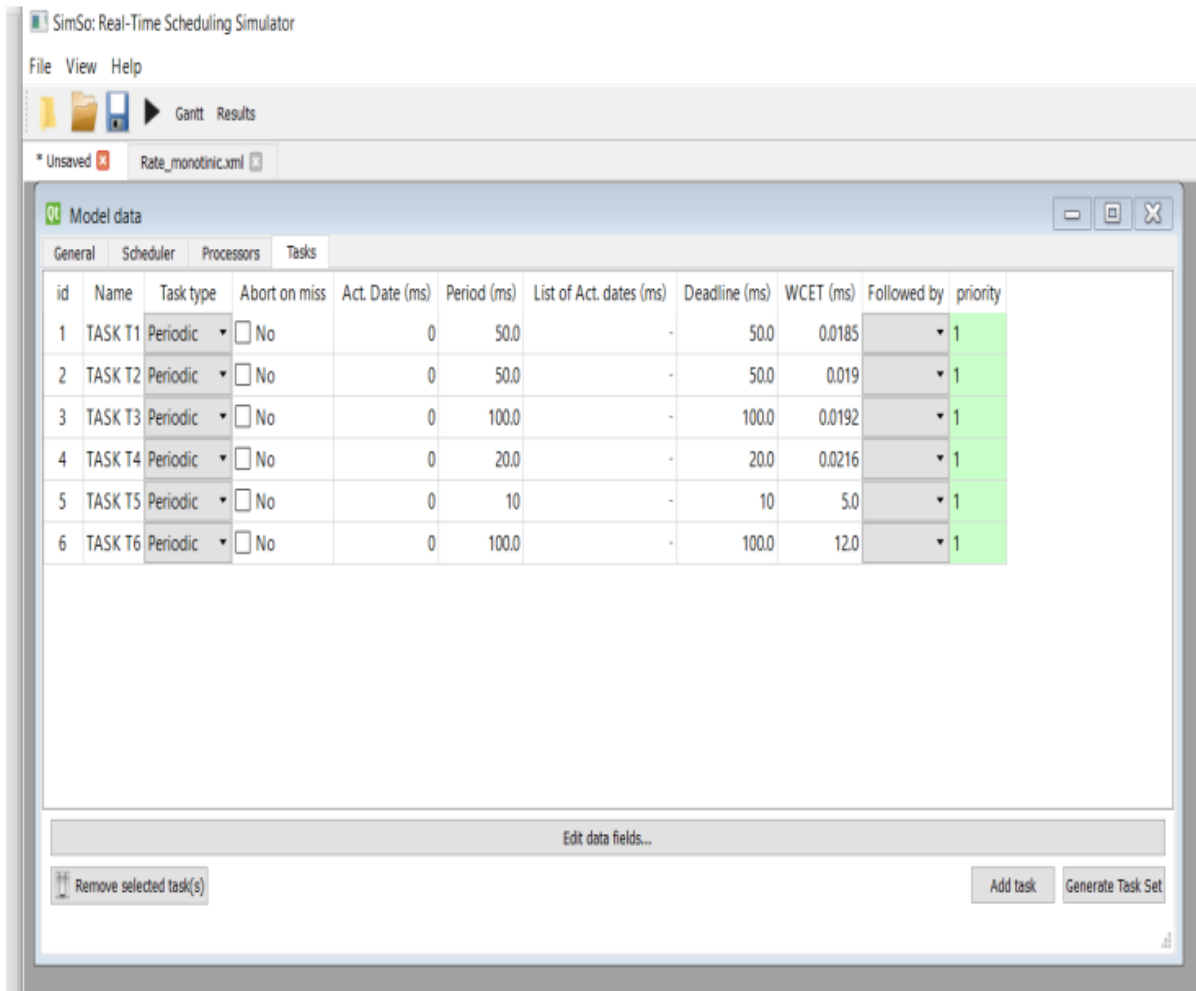
$$W(50) < D = 25.0725 ms < D = 25.0915 ms < D = 50.2022 ms < D = 62.2022 ms$$

Method 2: SIMSO

Using Simso offline simulator, simulate the given set of tasks assuming:

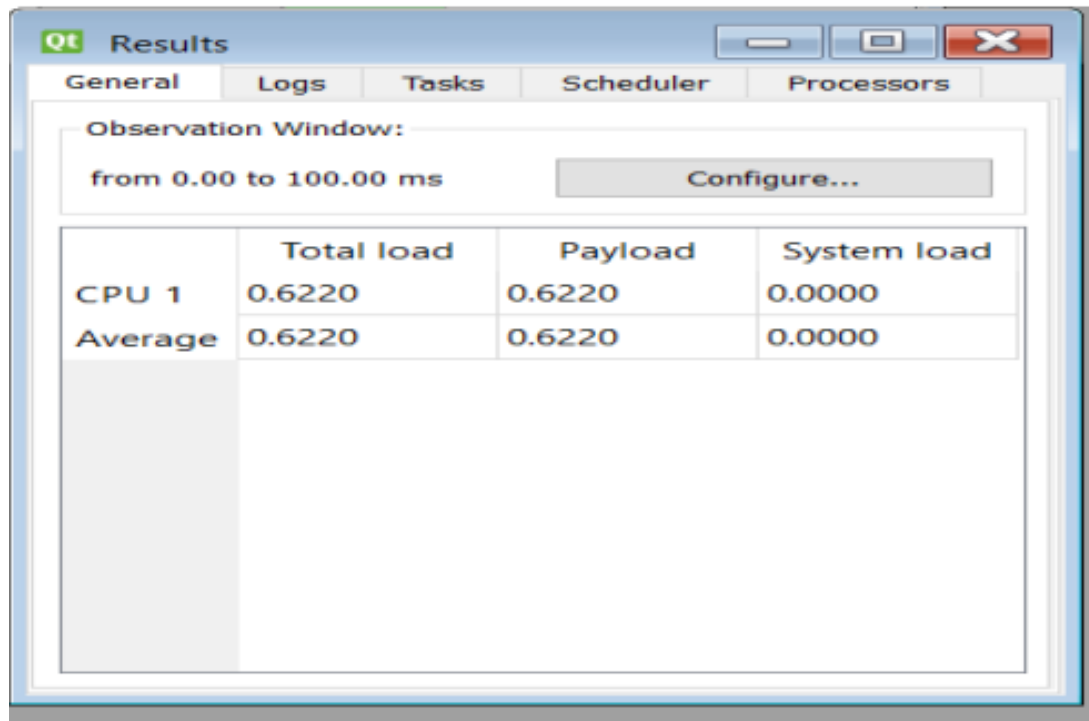
1. Calculate Hyper-period
2. CPU Load
3. Schedulability Analysis

*Tasks Creation



1. Calculate Hyper-period=100ms

2. CPU Load (Simso)



The image shows a Qt Results window with a blue title bar and standard window controls. It has five tabs: General, Logs, Tasks, Scheduler, and Processors. The General tab is active, displaying an 'Observation Window' from 0.00 to 100.00 ms with a 'Configure...' button. Below this is a table with four columns: CPU, Total load, Payload, and System load. The table contains two rows of data for CPU 1, both showing identical values: Total load 0.6220, Payload 0.6220, and System load 0.0000. The first row is labeled 'CPU 1' and the second row is labeled 'Average'.

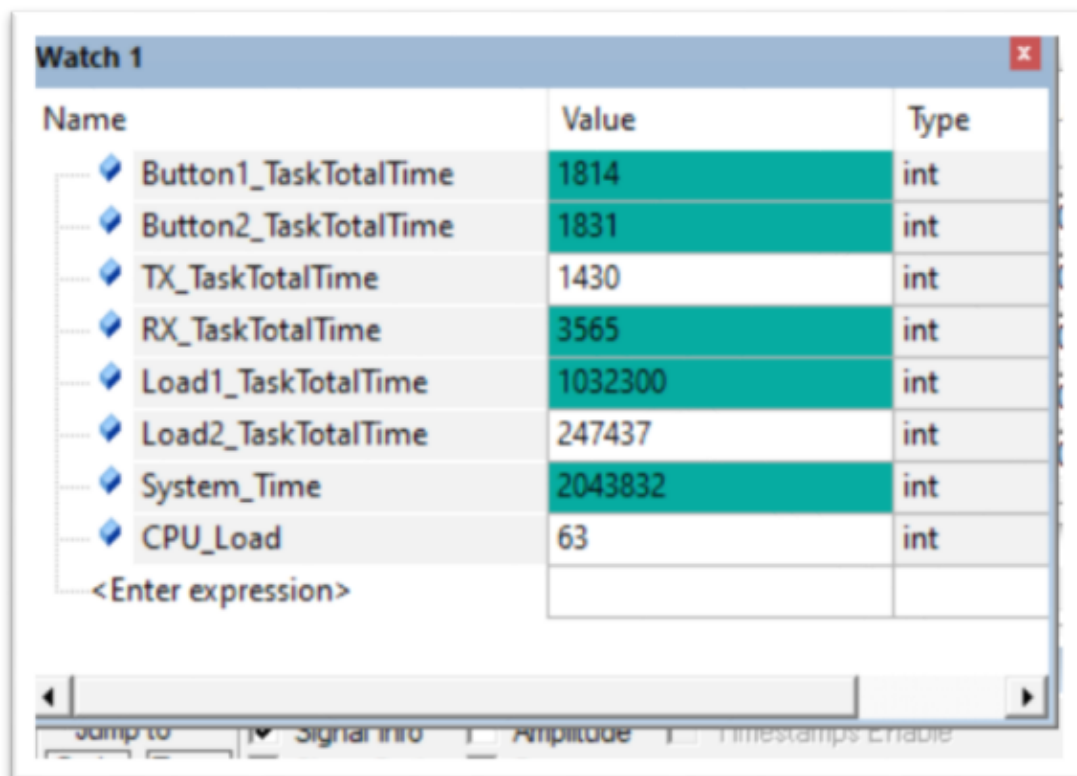
	Total load	Payload	System load
CPU 1	0.6220	0.6220	0.0000
Average	0.6220	0.6220	0.0000

3.Schedulability Analysis: system is Schedulability from Gantt Chart



Method 3:using Keil

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

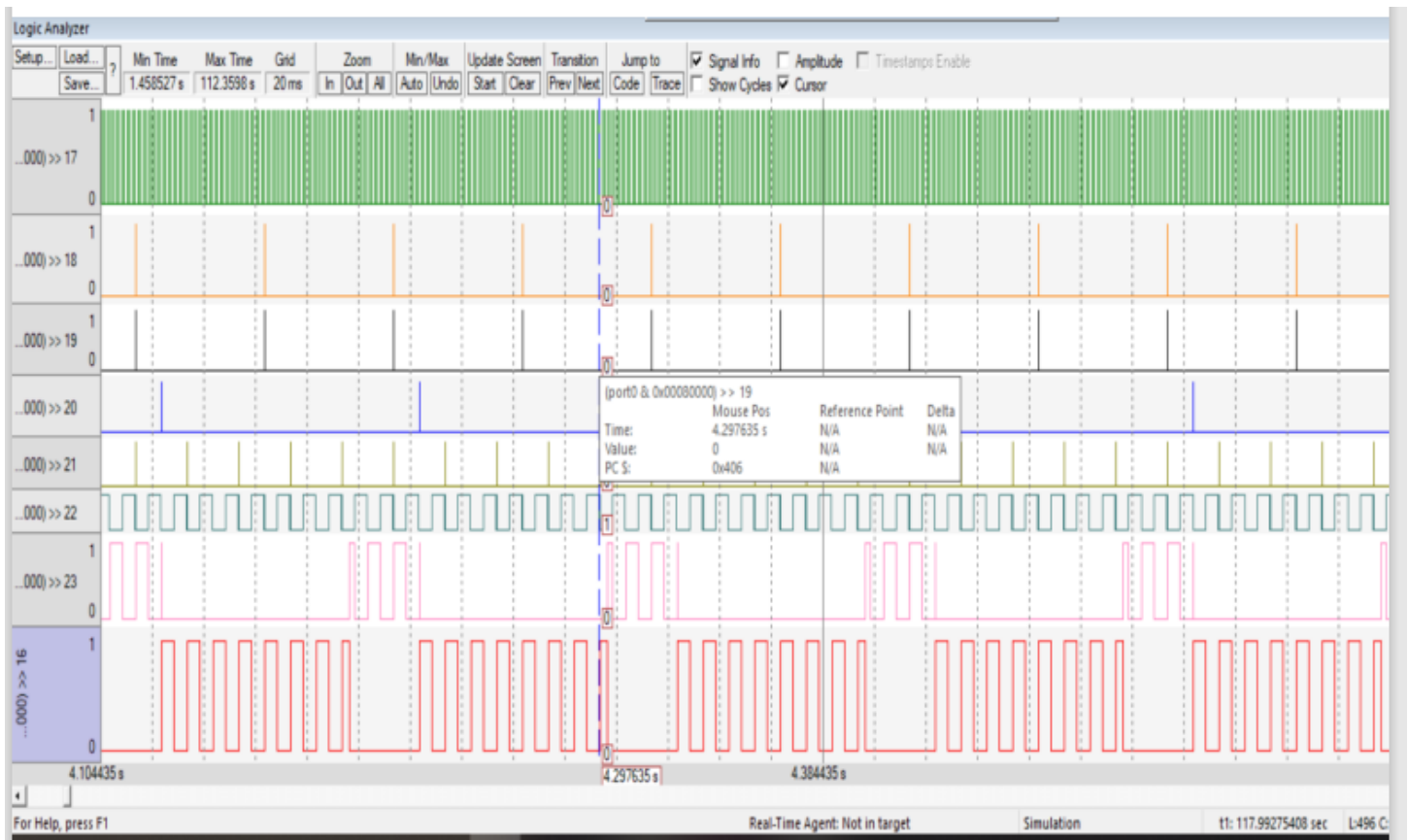


Watch 1

Name	Value	Type
Button1_TaskTotalTime	1814	int
Button2_TaskTotalTime	1831	int
TX_TaskTotalTime	1430	int
RX_TaskTotalTime	3565	int
Load1_TaskTotalTime	1032300	int
Load2_TaskTotalTime	247437	int
System_Time	2043832	int
CPU_Load	63	int
<Enter expression>		

Jump to: ☐ Signal info ☐ Amplitude ☐ Timestamps Enable

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



As We see the results of the three methods give the same CPU load = 62.7%, which means a successful implementation

