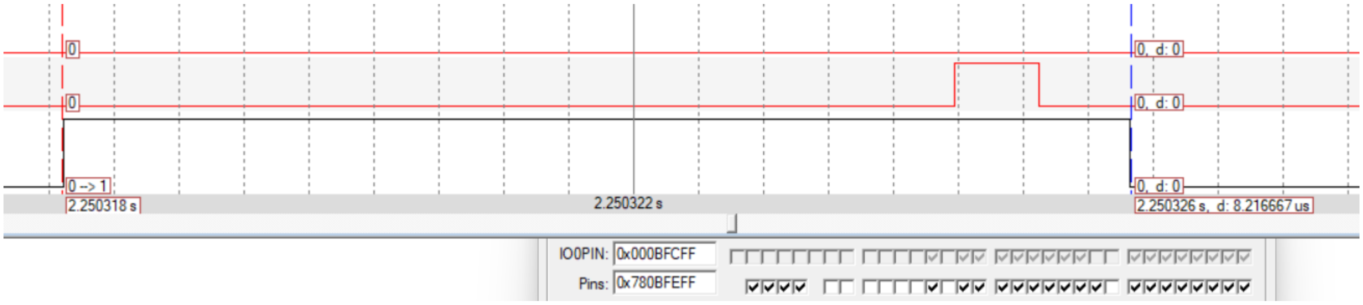
Implementing EDF Scheduler

(Verifying the system implementation)

1- Using Analytical Method

Execution Time for each task was calculated using GPIOs and logic analyzer:

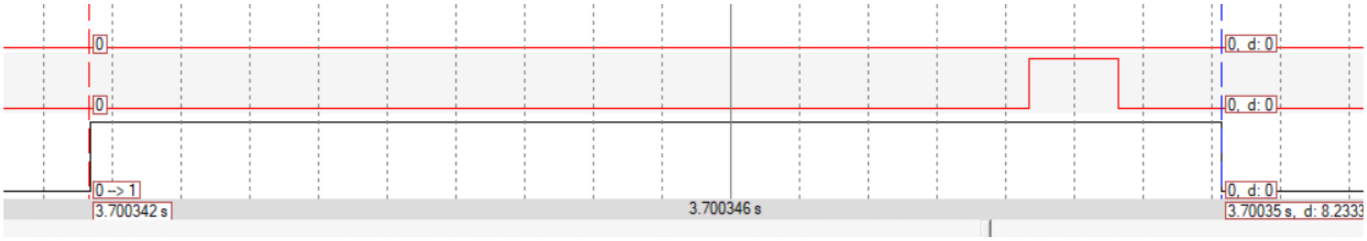
Task 1: ""Button_1_Monitor"":



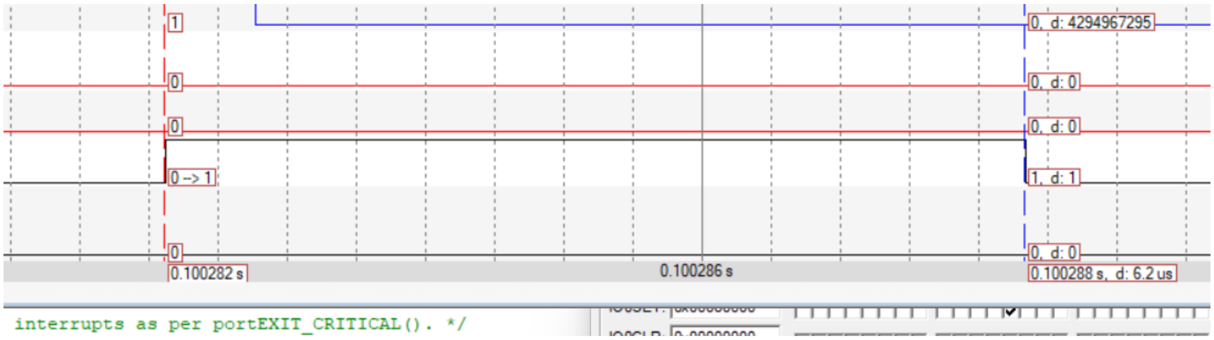
IO0PIN: 0x000BFCFF

Pins: 0x780BFEFF

Task 2: ""Button_2_Monitor"":

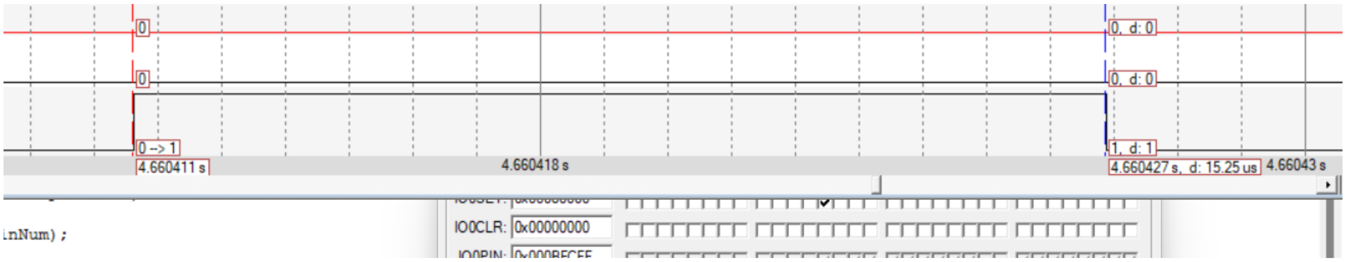


Task 3: ""Periodic_Transmitter"":



```
interrupts as per portEXIT_CRITICAL(). */
```

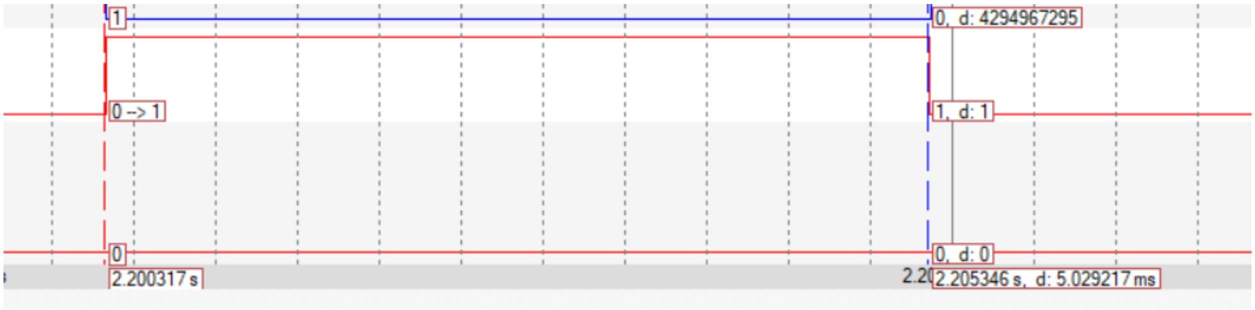
Task 4: ""Uart_Receiver"":



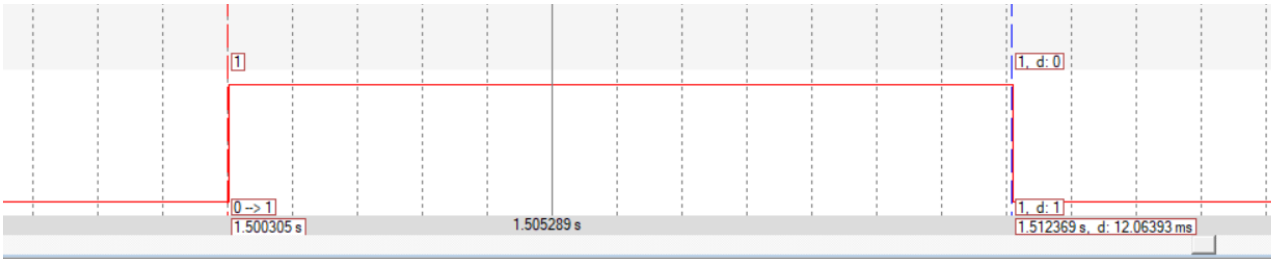
Note: The execution time is calculated for the worst scenario, for example in task 1: the longest execution will be when there's an edge detected (detection condition is true) and the string is sent to queue

Here are also screenshots for using GPIOs and logic analyzer to determine the X loop number for the two load simulation tasks :

Task 5: ""Load_1_Simulation"":



Task 6: ""Load_2_Simulation"":



Now, we have:

Task 1: ""Button_1_Monitor"", {Periodicity: 50, Deadline: 50, Execution Time: 8.3us} priority = 3

Task 2: ""Button_2_Monitor"", {Periodicity: 50, Deadline: 50, Execution Time: 8.3us} priority = 4

Task 3: ""Periodic_Transmitter"", {Periodicity: 100, D: 100, Execution Time: 6.25us} priority = 2

Task 4: ""Uart_Receiver"", {Periodicity: 20, Deadline: 20, Execution Time: 15.3us} priority = 5

Task 5: ""Load_1_Simulation"", {Periodicity: 10, Deadline: 10, Execution time: 5ms} priority = 6

Task 6: ""Load_2_Simulation"", {Periodicity: 100, D: 100, Execution time: 12ms} priority = 1

And we can calculate:

a) System Hyper Period (H) = 100

b) CPU load = $\left((2 \times 0.0083) + (2 \times 0.0083) + 0.00625 + (5 \times 0.0153) + (10 \times 5) + 12\ \right) \div 100\ =\ 0.62$

c) U = $\frac{0.0083}{50} + \frac{0.0083}{50} + \frac{0.00625}{100} + \frac{0.0153}{20} + \frac{5}{10} + \frac{12}{100} = \ \ 0.62$

URM = $6 \times \left(2^{\frac{1}{6}} - 1\right) =\ 0.73$
U < URM
System guaranteed schedulable

d) Critical Instant = Hyper Period = 100

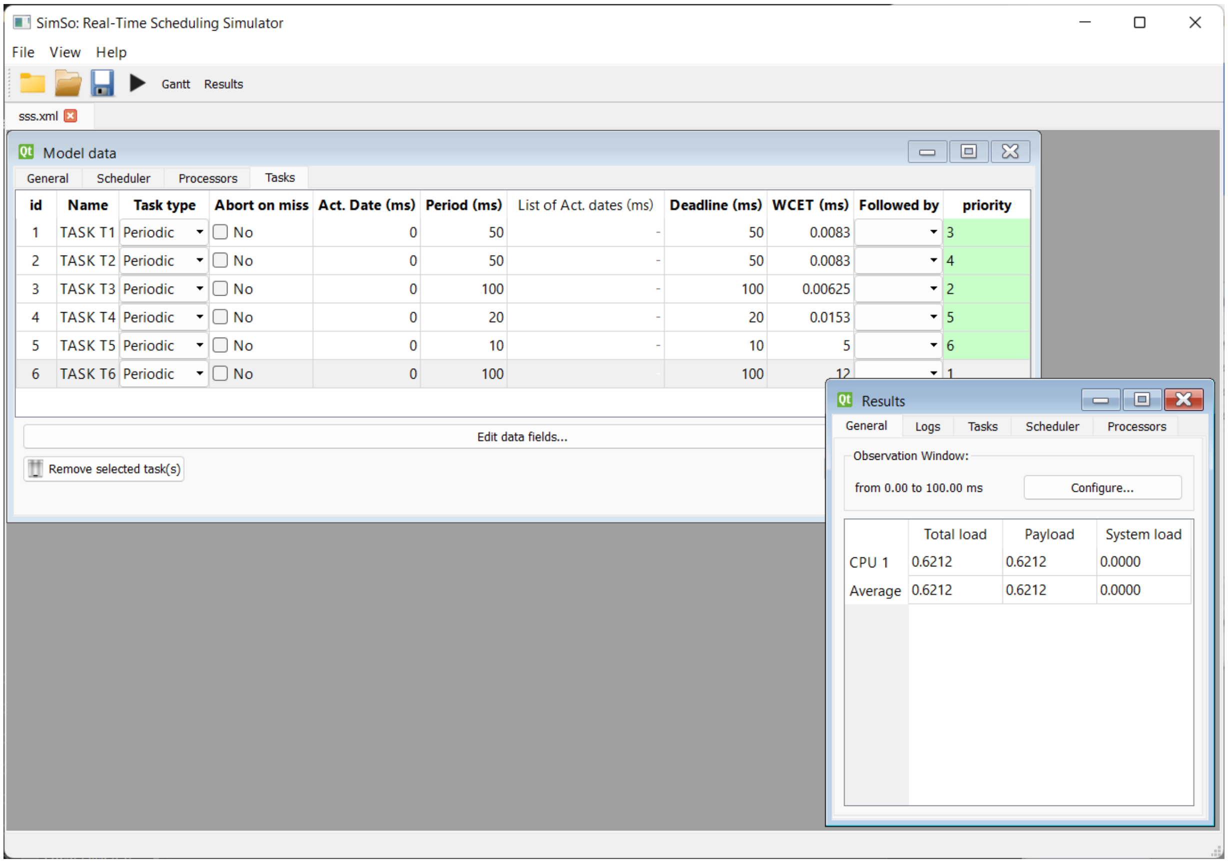
Assuming the given set of tasks are scheduled using a fixed priority rate -monotonic scheduler

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

For the highest priority task --> ""Load_1_Simulation"", {P: 10, D: 10, C: 5ms}
Required time is always 5ms and provided is 10ms, so this task is schedulable.

And so on knowing the required and provided time for each task we can know the schedulability, also by using equation

2- Using Simso offline simulator:



SimSo: Real-Time Scheduling Simulator
File View Help
Gantt Results
sss.xml
Model data
General Scheduler Processors Tasks

id	Name	Task type	Abort on miss	Act. Date (ms)	Period (ms)	List of Act. dates (ms)	Deadline (ms)	WCET (ms)	Followed by	priority
1	TASK T1	Periodic	□ No	0	50	-	50	0.0083		3
2	TASK T2	Periodic	□ No	0	50	-	50	0.0083		4
3	TASK T3	Periodic	□ No	0	100	-	100	0.00625		2
4	TASK T4	Periodic	□ No	0	20	-	20	0.0153		5
5	TASK T5	Periodic	□ No	0	10	-	10	5		6
6	TASK T6	Periodic	□ No	0	100	-	100	12		1

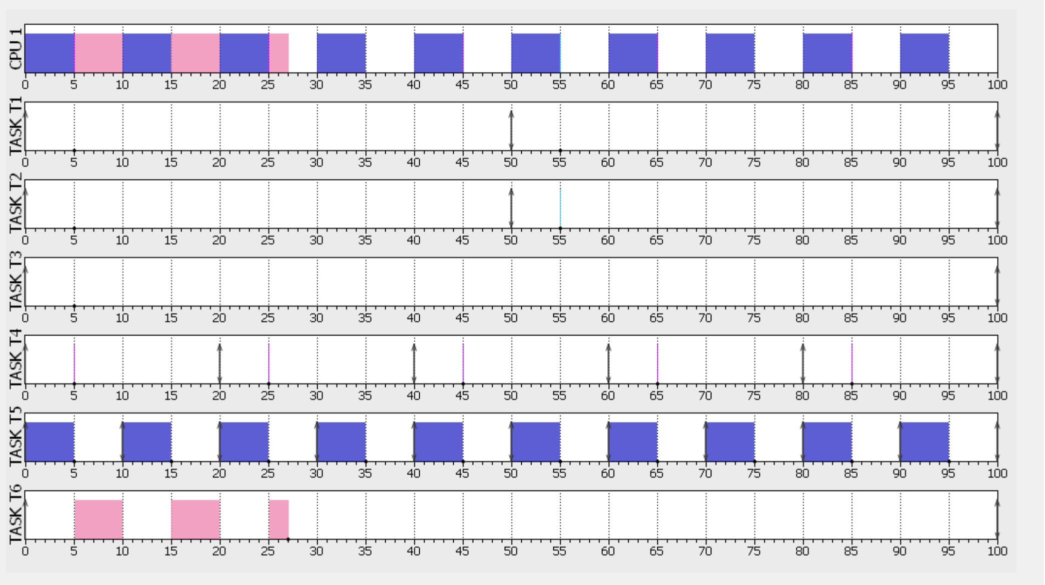
Edit data fields...
Remove selected task(s)
Results
General Logs Tasks Scheduler Processors
Configure...

Observation Window:

from 0.00 to 100.00 ms

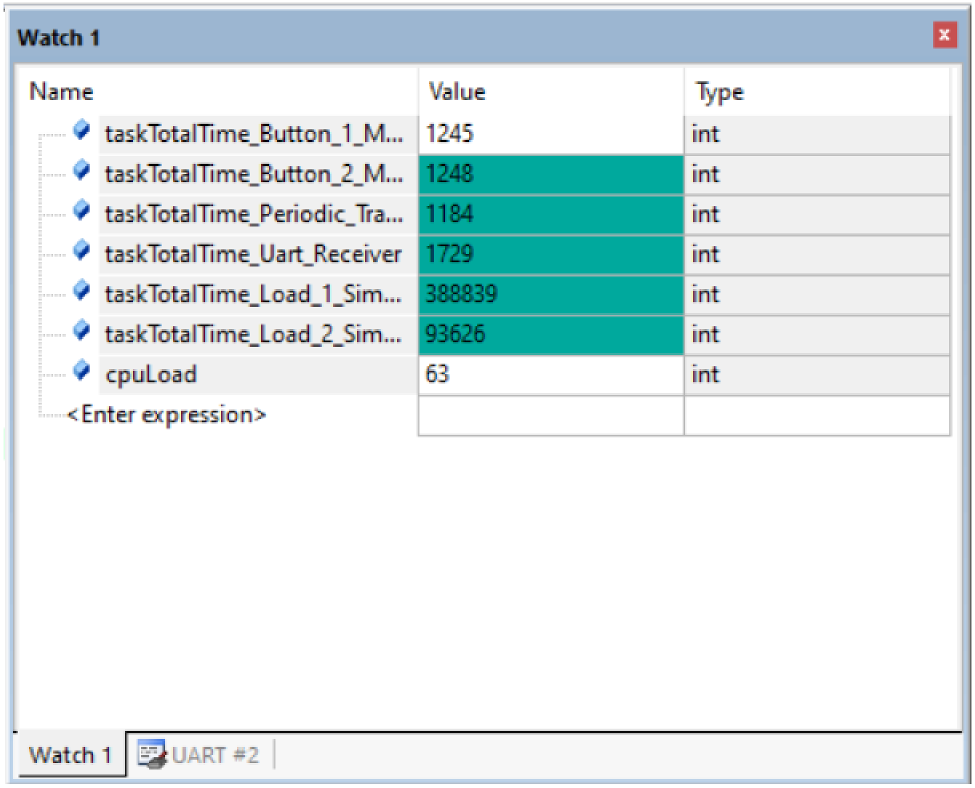
	Total load	Payload	System load
CPU 1	0.6212	0.6212	0.0000
Average	0.6212	0.6212	0.0000

we can see that CPU Load is 0.62 as calculated analytically and no missing occurs



3- Using Keil simulator in run-time:

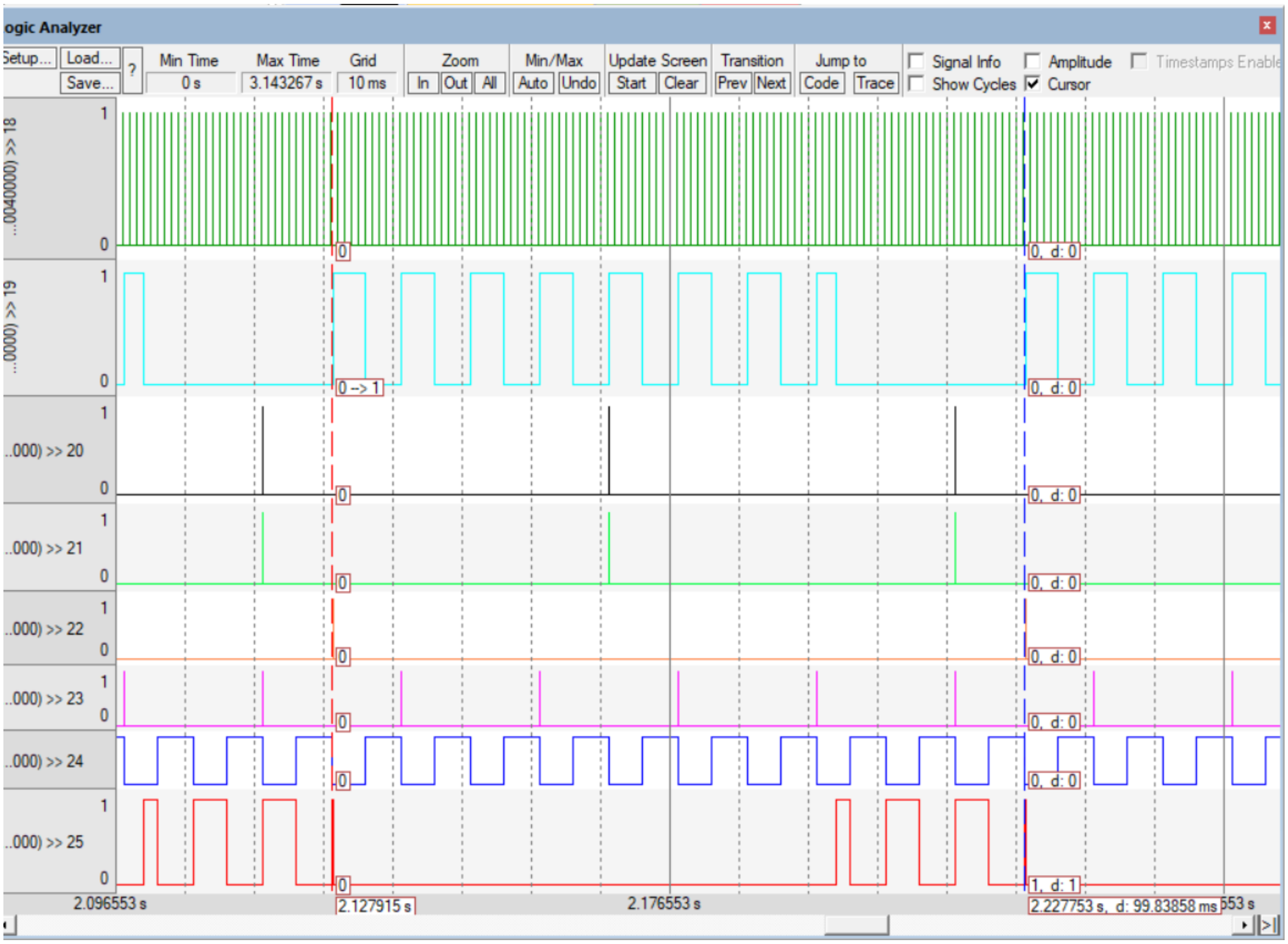
a) CPU usage time using timer 1 and trace macros:



Name	Value	Type
taskTotalTime_Button_1_M...	1245	int
taskTotalTime_Button_2_M...	1248	int
taskTotalTime_Periodic_Tra...	1184	int
taskTotalTime_Uart_Receiver	1729	int
taskTotalTime_Load_1_Sim...	388839	int
taskTotalTime_Load_2_Sim...	93626	int
cpuLoad	63	int
<Enter expression>		

The CPU load saturates at 63% which is also approximately equals to the 0.62 that we got from calculating analytically

b) Using trace macros and GPIOs, plot the execution of all tasks, tick, and the idle task on the logic analyzer:



ogic Analyzer

Min Time: 0 s
Max Time: 3.143267 s
Grid: 10 ms

☐ Signal Info
☐ Show Cycles
☐ Amplitude
☑ Cursor
☐ Timestamps Enable

Tick (green)

Idle Task (Indigo)

Task 1: ""Button_1_Monitor"" (Black)

Task 2: ""Button_2_Monitor"" (Light Green)

Task 3: ""Periodic_Transmitter"" (Orange)

Task 4: ""Uart_Receiver"" (Violet)

Task 5: ""Load_1_Simulation"" (Blue)

Task 6: ""Load_2_Simulation"   (Red)

By looking at the Idle task we can see that it's off for nearly 63% of the hyper period and this is what we are expecting.