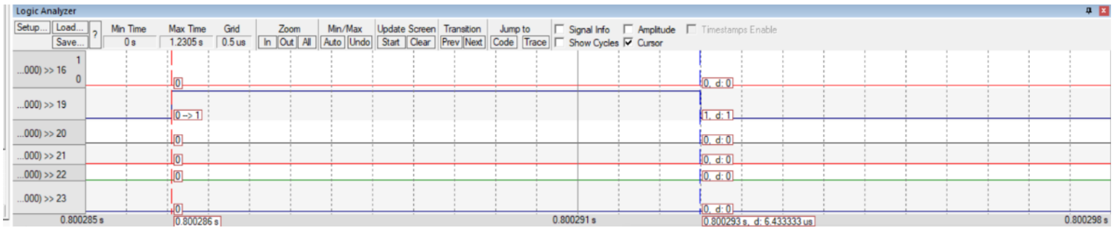
FreeRTOS EDF scheduler verification

- Tasks execution time calculation:

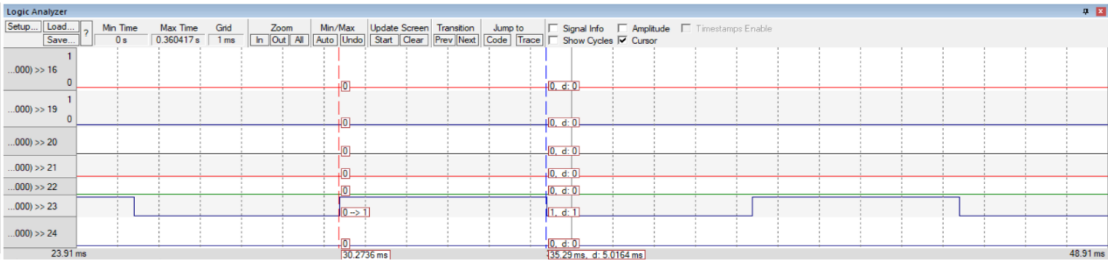
In order to start the verification methods we need first to calculate the execution time of each task using gpios and logic analyzer.

-"button1_monitor" and "button2_monito r" tasks execution time:



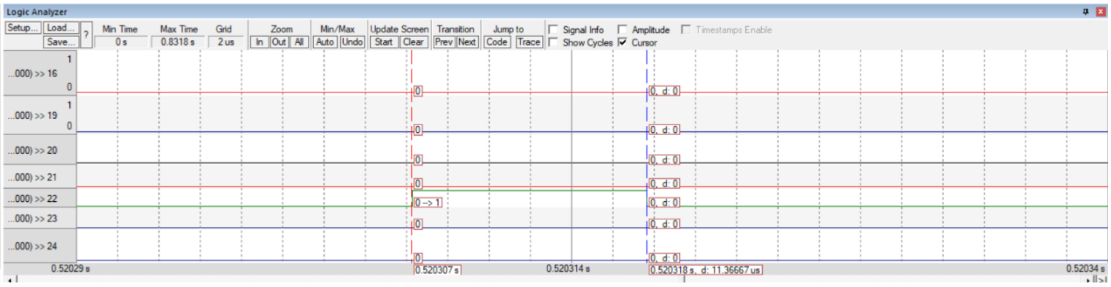
About 8 us.

-"periodic_transmitter" task execution time:



About 6 us.

-"uart_receiver" task execution time:



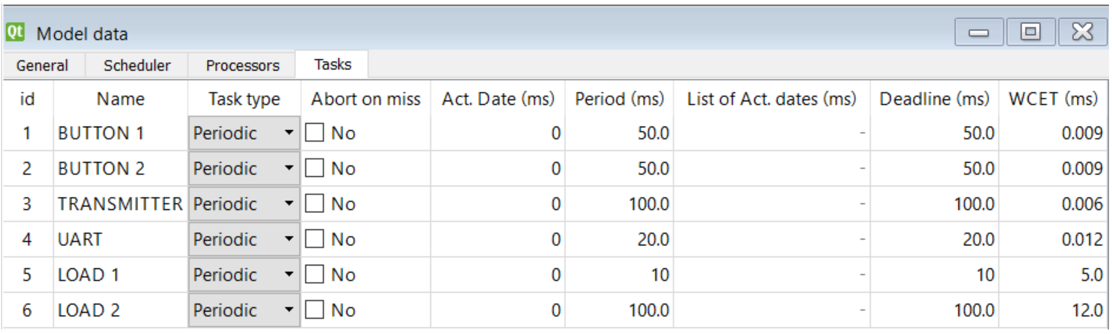
About 12 us.

-"load1_simulator" and "load2_simulator" tasks execution time is given as 5ms and 12ms.

- Verifying using analytical methods:

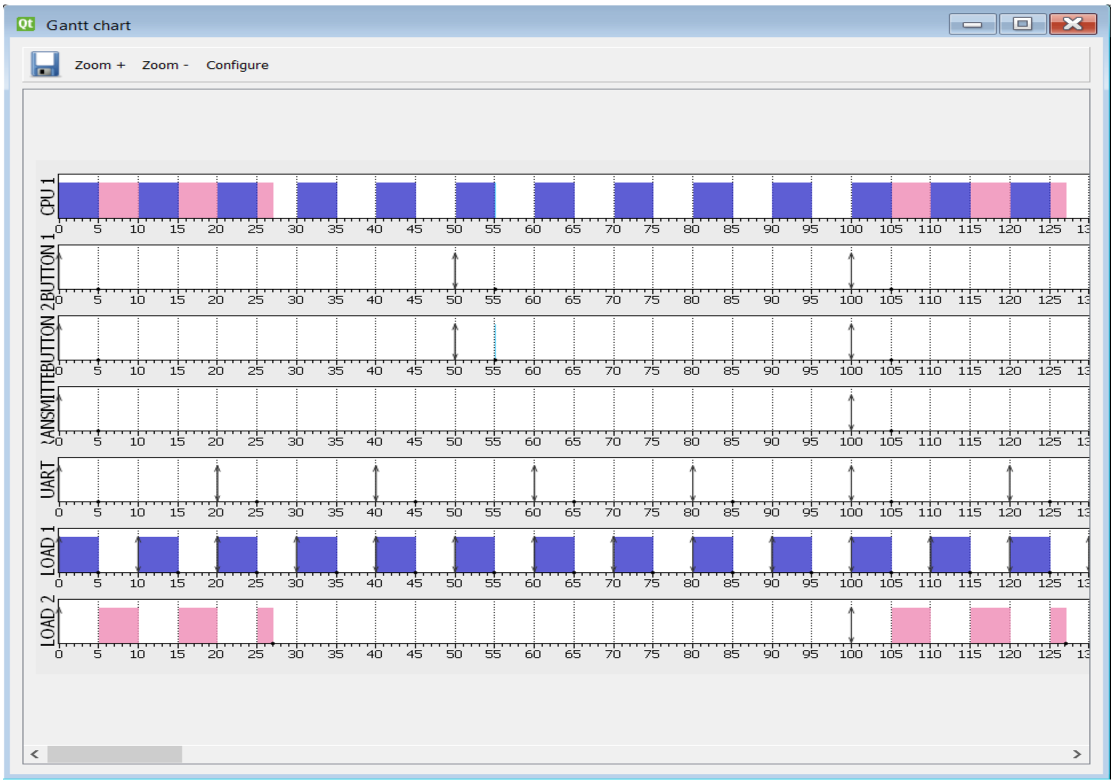
- Verifying using Simso offline simulator:

First, we choose the scheduler as rate-monotonic scheduler and set each task with the required period and execution time.



id	Name	Task type	Abort on miss	Act. Date (ms)	Period (ms)	List of Act. dates (ms)	Deadline (ms)	WCET (ms)
1	BUTTON 1	Periodic	□ No	0	50.0	-	50.0	0.009
2	BUTTON 2	Periodic	□ No	0	50.0	-	50.0	0.009
3	TRANSMITTER	Periodic	□ No	0	100.0	-	100.0	0.006
4	UART	Periodic	□ No	0	20.0	-	20.0	0.012
5	LOAD 1	Periodic	□ No	0	10	-	10	5.0
6	LOAD 2	Periodic	□ No	0	100.0	-	100.0	12.0

Second, we start the simulator to see each task behavior.

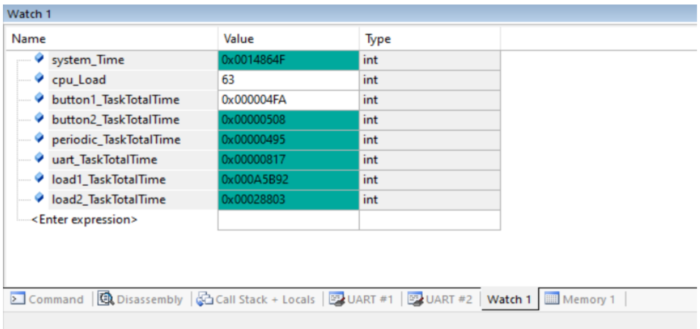


As we can see the execution time of the first 4 tasks is so small in compare with load1 and load2 tasks.

Load2 task is being pre-empted by load1 tasks as it have a lower periodicity as expected in rate-monotonic mode.

- Verifying using Keil simulator:

First cpu load calculation using trace macros and timer1.

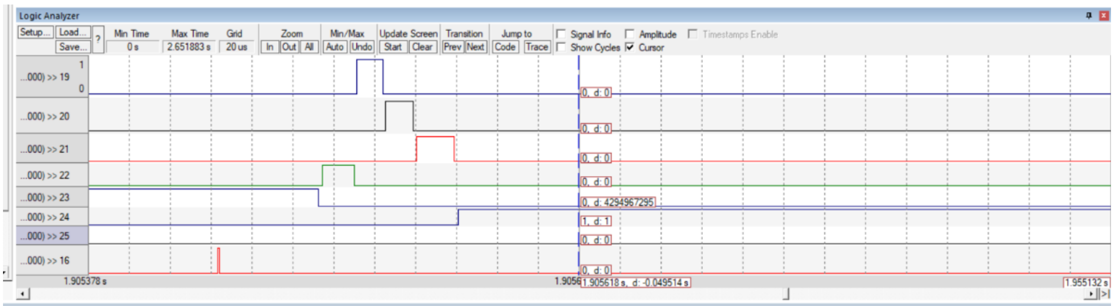


Watch 1

Name	Value	Type
system_Time	0x0014864F	int
cpu_Load	63	int
button1_TaskTotalTime	0x000004FA	int
button2_TaskTotalTime	0x00000508	int
periodic_TaskTotalTime	0x00000495	int
uart_TaskTotalTime	0x00000817	int
load1_TaskTotalTime	0x000A5B92	int
load2_TaskTotalTime	0x00028803	int
<Enter expression>		

The cpu load is 63% as showed, so we know that the system is not too much loaded and the implementation is successful.

Second, the execution of all tasks, tick and idle task using trace macros and gpios.

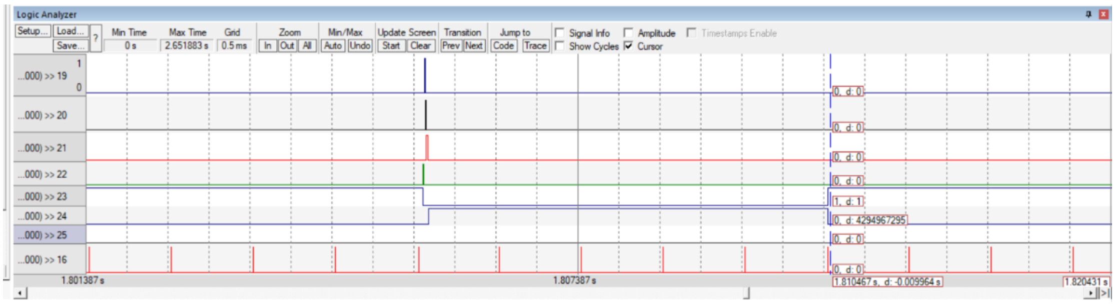


Logic Analyzer

Min Time 0 s | Max Time 2.651883 s | Grid 20 us

we expect the load1_simulator task to be the first to be executed as it has the earliest deadline and this happens as its represented by pin23.

Then, uart_receiver task take place as It have the second early deadline, and button1_monitor and button2_monitor came after each other as they have the same deadline and lastly periodic_transmitter task.



Logic Analyzer

Min Time 0 s | Max Time 2.651883 s | Grid 0.5 ms

We couldn't see the pre-emption in the last figure and the execution time of the tasks is very small, but in load1_simulator and load2_simulator we can see the load1_simulator always come first and pre-empt the second one each period using pin 24 and 25.