System Verification

Tasks:

Note: all tasks execution time is calculated from the actual implemented tasks using GPIOs and the logic analyzer.

Task Name	*Periodicity / Deadline (MS)*	*Execution Time (MS)*
Button_1_Monitor	50	0.008
Button_2_Monitor	50	0.008
Periodic Transmitter	100	0.0096
UART Receiver	20	0.017
Load 1 Simulation	10	5
Load 2 Simulation	100	12

Methods of Verification:

1- Using Analytical Method:

1.1- System Hyper period:

- It's the Least Common Multiple of all task's periods
- H = LCM (50, 50, 100, 20, 10, 100) = 100

1.2- CPU Load

- U = (E1 + E2 + E3 + E4 + E5 + E6) / H
- where E is the Execution time and H is the Hyper period.
- U = (0.008*2 + 0.008*2 + 0.0096 + 0.017*5 + 5*10 + 12) / 100 = 0.621 (62.1%)

1.3- System stimulability check using URM and Time Demand Analysis Techniques:

- $\sum_{i=1}^{n} \frac{c_i}{p_i} \leq n(2^{\frac{1}{n}} - 1)$
- $L.H.S = \sum_{i=1}^{n} \frac{c_i}{p_i} = \frac{0.008}{50} + \frac{0.008}{50} + \frac{0.0096}{100} + \frac{0.017}{20} + \frac{5}{10} + \frac{12}{100} = 0.621$
- $R.H.S = n\left(2^{\frac{1}{n}} - 1\right) = 0.753$
- $L.H.S \leq R.H.S$, so, the system is schedulable.

2- Time Demand Analysis:

a- Sort the tasks making the highest priority at the first:

Task Name	*Periodicity / Deadline (MS)*	*Execution Time (MS)*
1- Load 1 Simulation	10	5
2- UART Receiver	20	0.017
3- Button_1_Monitor	50	0.008
4- Button_2_Monitor	50	0.008
5- Periodic Transmitter	100	0.0096
6- Load 2 Simulation	100	12

b- Choose the critical instant 0 then:

$$w_1(10) = 5 + 0 = 5 < deadline \qquad w_2(20) = 0.017 + 5 * \frac{20}{10} = 5.017 < deadline$$

$$w_3(50) = 0.008 + 0.017 * \frac{50}{20} + 5 * \frac{50}{10} = 25.059 < deadline$$

$$w_4(50) = 0.008 + 0.008 * \frac{50}{50} + 0.017 * \frac{50}{20} + 5 * \frac{50}{10} = 25.067 < deadline$$

$$w_5(100) = 0.0096 + 0.008 * \frac{100}{50} + 0.008 * \frac{100}{50} + 0.017 * \frac{100}{20} + 5 * \frac{100}{10} = 50.1266 < deadline$$

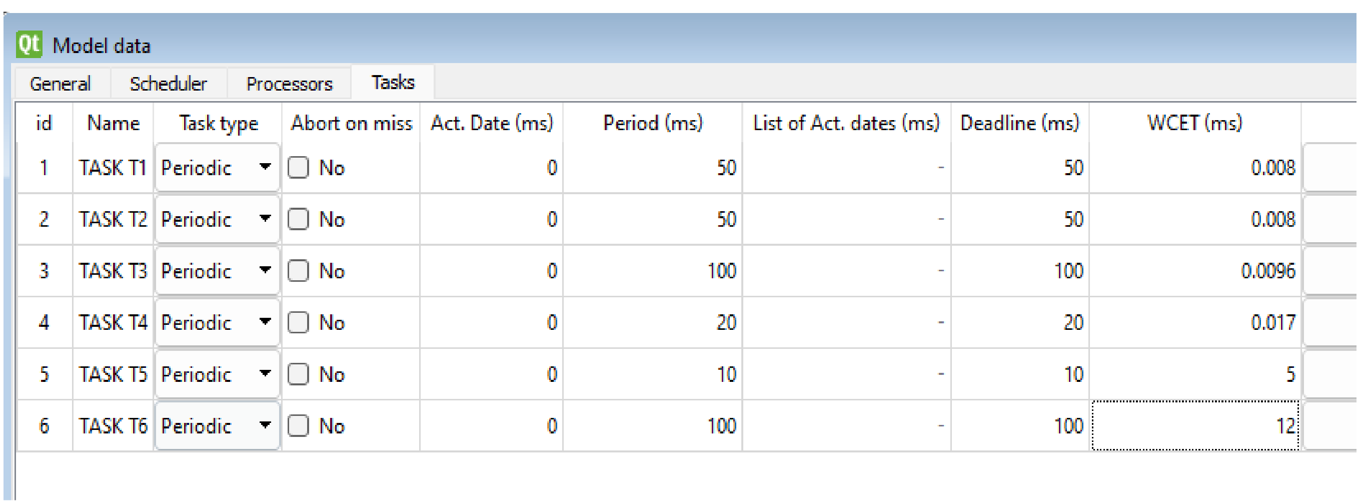
$$w_6(100) = 12 + 0.0096 * \frac{100}{100} + 0.008 * \frac{100}{50} + 0.008 * \frac{100}{50} + 0.017 * \frac{100}{20} + 5 * \frac{100}{10} = 62.1266 < deadline$$

As all Tasks are less Than the deadline. So, the system is schedulable.

2- Using SIMSO offline simulator:

Used Scheduler: Fixed priority rate monotonic.

Tasks Simulated:



Model data

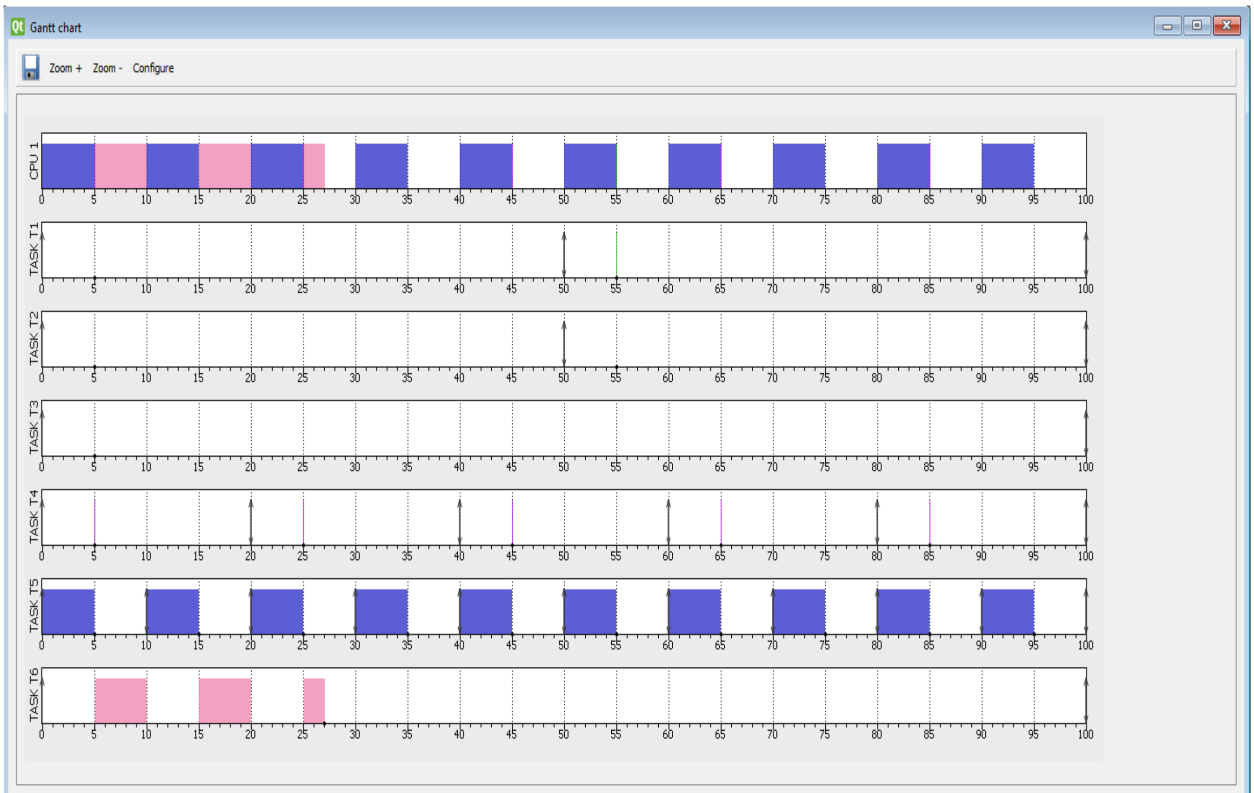
General | Scheduler | Processors | Tasks

id	Name	Task type	Abort on miss	Act. Date (ms)	Period (ms)	List of Act. dates (ms)	Deadline (ms)	WCET (ms)
1	TASK T1	Periodic	No	0	50	-	50	0.008
2	TASK T2	Periodic	No	0	50	-	50	0.008
3	TASK T3	Periodic	No	0	100	-	100	0.0096
4	TASK T4	Periodic	No	0	20	-	20	0.017
5	TASK T5	Periodic	No	0	10	-	10	5
6	TASK T6	Periodic	No	0	100	-	100	12

For The CPU load the is the same as the analytical mode

	Total load	Payload	System load
CPU 1	0.6213	0.6213	0.0000
Average	0.6213	0.6213	0.0000

Gantt chart over the Hyper period:



3- Using Keil Simulator at Runtime:

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

Watch 1

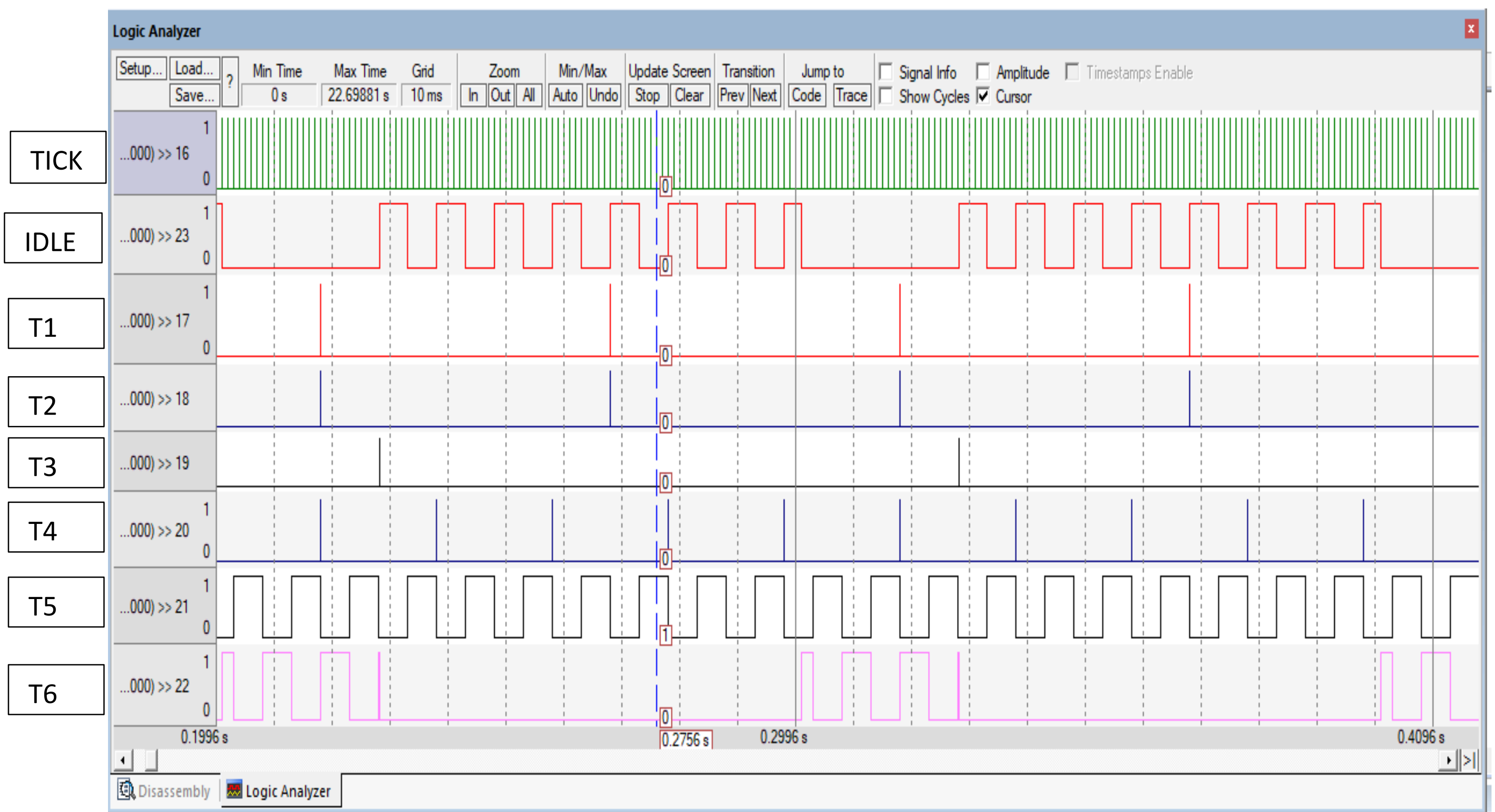
Name	Value	Type
miss	<cannot evaluate>	uchar
deadline_misses[0]	0x00000000	int
deadline_misses[1]	0x00000000	int
deadline_misses[2]	0x00000000	int
deadline_misses[3]	0x00000000	int
deadline_misses[4]	0x00000000	int
deadline_misses[5]	0x00000000	int
total_sys_time	0x0007F0E4	int
cpu_load	63	int
task_1_total	0x000004C2	int
task_2_total	0x000004C9	int
task_3_total	0x00000482	int
task_4_total	0x00000628	int
task_5_total	0x00040013	int
task_6_total	0x0000F59F	int
<Enter expression>		

Note:

1- The CPU load is the same as the calculated analytically and the obtained using SIMSO offline simulator.

2- None of The Tasks Miss the Deadline.

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



Note:

- The above chart was using the implemented EDF scheduler. As you can see tasks with closer deadline preempts other tasks.
- The IDLE task never interfered with my other main tasks so my modification in the IDLE task function to make sure it never preempts my main tasks is successful. As I made sure It always offsets the maximum main tasks deadline by a user defined amount So, it never executes unless there are no more tasks in the ready queue.