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. System Hyper period:

* It's the Least Common Multiple of all task’s periods
* H = LCM (50, 50, 100, 20, 10, 100) = 100
  1. 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. System stimulability check using URM and Time Demand Analysis Techniques:
* , so, the system is schedulable.

1. Time Demand Analysis:
2. Sort the tasks making the highest priority at the first:

|  |  |  |
| --- | --- | --- |
| Task Name | Periodicity / Deadline (MS) | Execution Time (MS) |
| 1. Load 1 Simulation | 10 | 5 |
| 1. UART Receiver | 20 | 0.017 |
| 1. Button\_1\_Monitor | 50 | 0.008 |
| 1. Button\_2\_Monitor | 50 | 0.008 |
| 1. Periodic Transmitter | 100 | 0.0096 |
| 1. Load 2 Simulation | 100 | 12 |

1. Choose the critical instant 0 then:

*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:



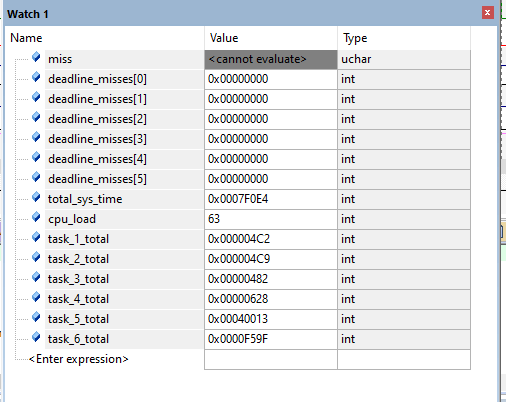
For The CPU load the is the same as the analytical mode



Gantt chart over the Hyper period:

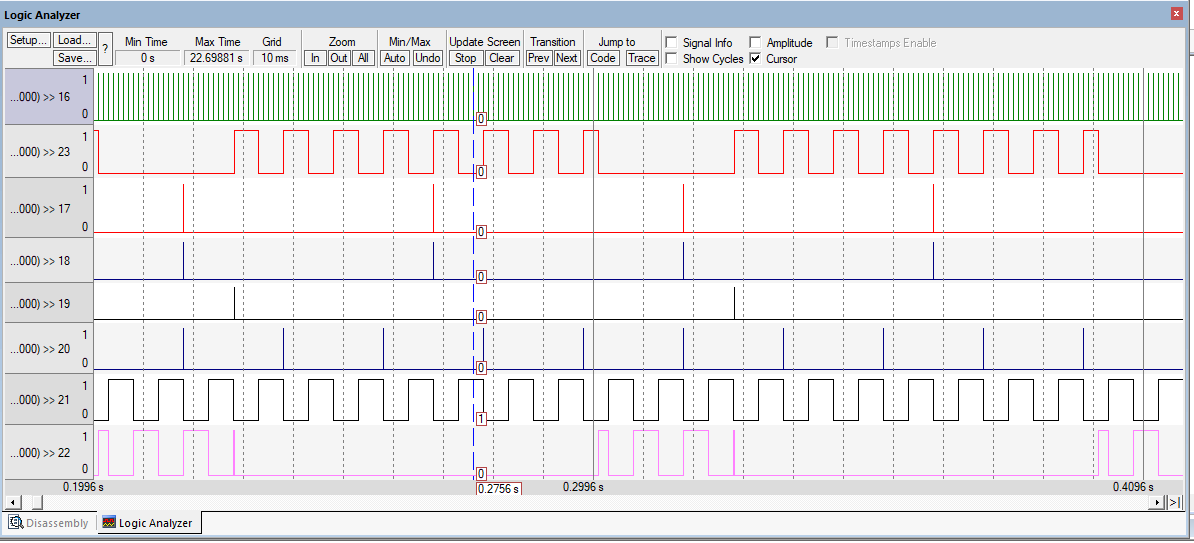


1. Using Keil Simulator at Runtime:
2. Calculate the CPU usage time using timer 1 and trace macros



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



TICK

IDLE

T1

T2

T3

T4

T5

T6

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.