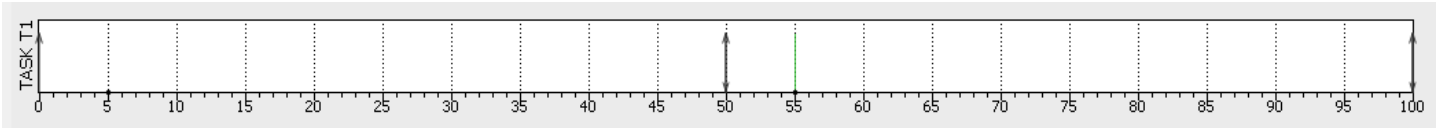
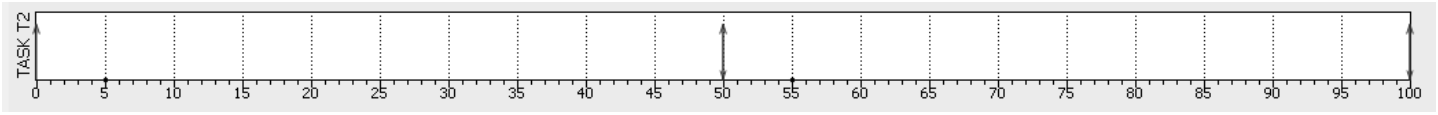
Verifying the system implementation

Tasks :

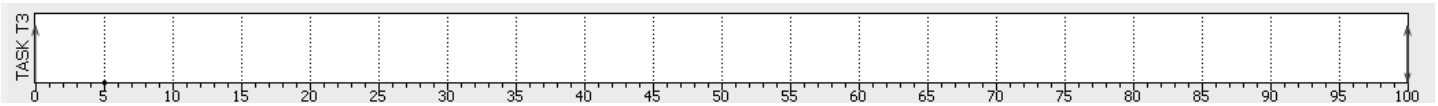
- Task 1: ""Button_1_Monitor"",
  $\{ Periodicity: 50, Deadline: 50, execution\ time: 2\mu s \}$



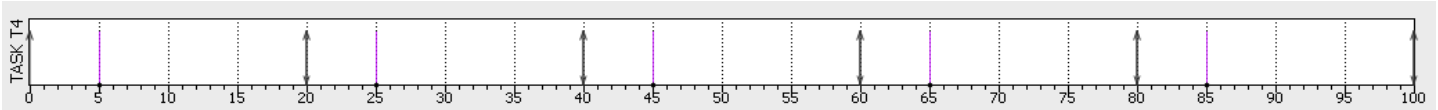
- Task 2: ""Button_2_Monitor"",
  $\{ Periodicity: 50, Deadline: 50, execution\ time: 2\mu s \}$



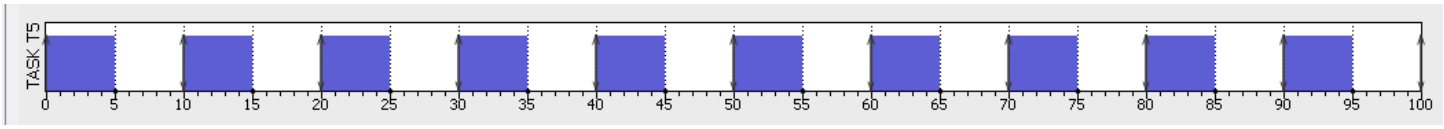
- Task 3: ""Periodic_Transmitter"",
  $\{ Periodicity: 100, Deadline: 100, execution\ time: 10\mu s \}$



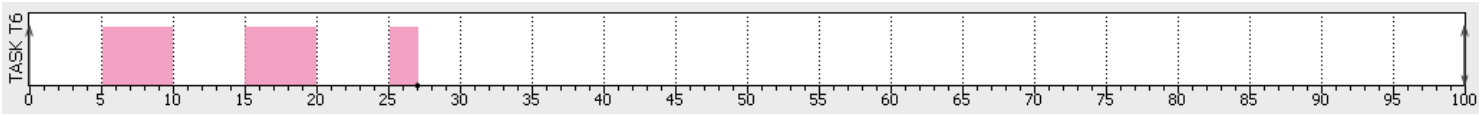
- Task 4: ""Uart_Receiver"",
  $\{ Periodicity: 20, Deadline: 20, execution\ time: 4\mu s \}$



- Task 5: ""Load_1_Simulation"",
  $\{ Periodicity: 10, Deadline: 10, execution\ time: 5ms \}$



- Task 6: ""Load_2_Simulation"",
  $\{ Periodicity: 100, Deadline: 100, execution\ time: 12ms \}$



1. **Hyper period :**

$$H = LCM(P_i) = LCM(50,100,20,10) = 100$$

That:

- H is the hyper period.
- P_i is all tasks periodicities.

2. **CPU load :**

$$U = {\sum_{i=1}^{n}(E_i)}/{H}$$

That:

- U is CPU utilization
- H is the hyper period.
- E_i is the ith task total execution time.
- n is the number of tasks.

So,

$$\text{U}= \frac{\left((2*0.002)+(2*0.002)+(1*0.01)+(5*0.004)+(10*5)+(1*12)\right)}{100} = 0.622$$

$$\therefore U\% = 62.2\ \%$$

3. **Schedulability :**

A. Using Rate-Monotonic utilization bound method (URM) :

$$U = \sum_{i=1}^{n} \frac{E_i}{P_i} \leq n * (2^{\frac{1}{n}} - 1)$$

That:

- U is total utilization.
- H is the hyper period.
- E_i is the ith task execution time.
- P_i is the ith tasks periodicity.
- n is the number of tasks.

$$U = \frac{0.002}{50} + \frac{0.002}{50} + \frac{0.01}{100} + \frac{0.004}{20} + \frac{5}{10} + \frac{12}{100} = 0.62$$

$$URM = 6 * \left(\left(2^{\frac{1}{6}}\right) - 1\right) = 0.73$$

$$U < URM$$

✓ System is guaranteed schedulable.

B. Using Time-Demand analysis :

For task i :

$$w_i(t) = e_i + \sum_{k=1}^{i-1} \left\lceil \frac{t}{P_k} \right\rceil e_k \qquad for\ 0 < t \leq P_i$$

That:

- w is the worst response time.
- t is the time instance.
- E_i is the ith task execution time.
- P_i is the ith tasks periodicity.

1. For task 1:

$w_1(10) = 0.002 + 0 = 0.002$

$w_1(20) = 0.002 + 0 = 0.002$

$w_1(30) = 0.002 + 0 = 0.002$

$w_1(40) = 0.002 + 0 = 0.002$

$w_1(50) = 0.002 + 0 = 0.002$

✓ $w_1(50) < D_1$ → task 1 is schedulable.

2. For task 2:

$$w_2(10) = 0.002 + \frac{10}{50} * 0.002 = 2.4 * 10^{-3}$$

$$w_2(20) = 0.002 + \frac{20}{50} * 0.002 = 2.8 * 10^{-3}$$

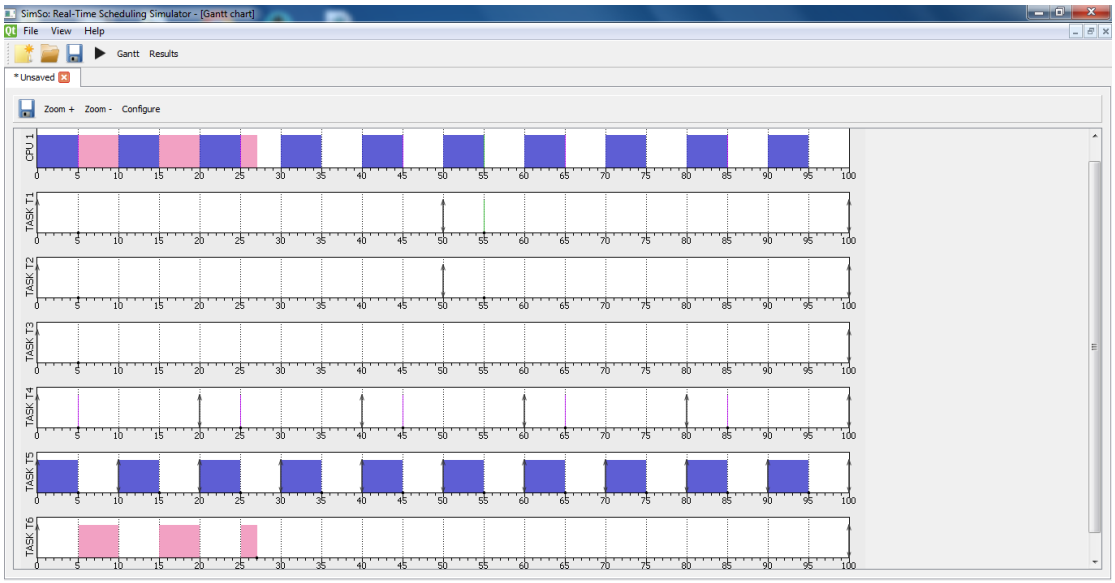
$$w_2(30) = 0.002 + \frac{30}{50} * 0.002 = 3.2 * 10^{-3}$$

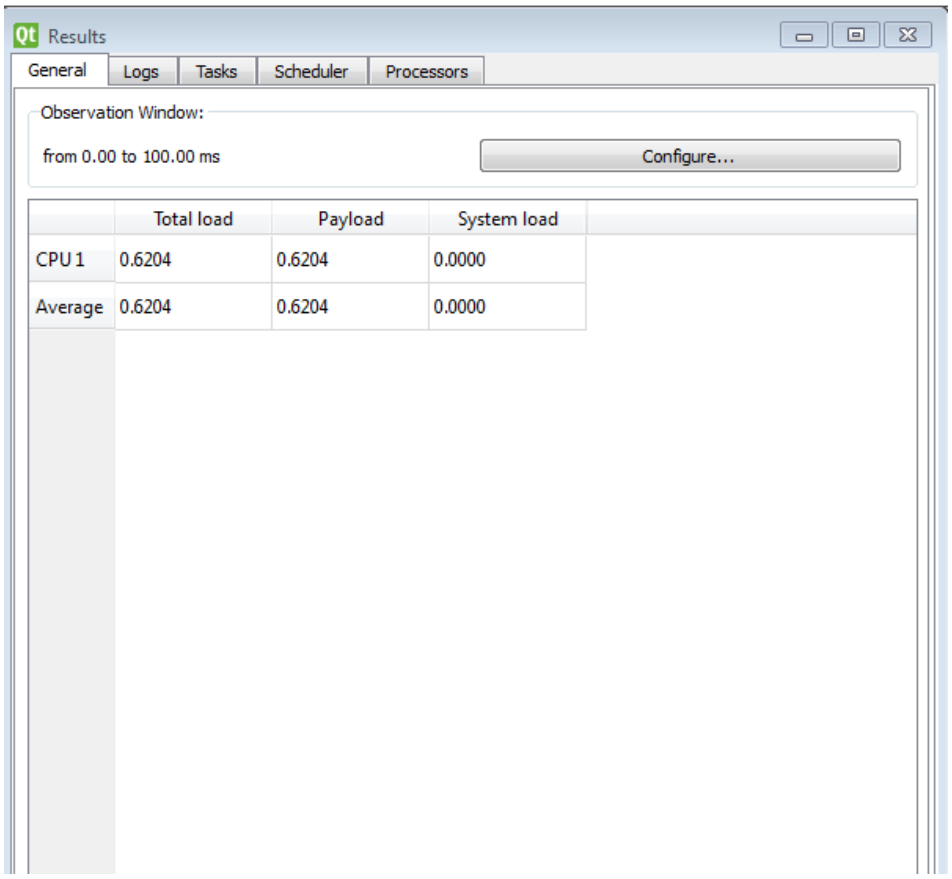
$$w_2(40) = 0.002 + \frac{40}{50} * 0.002 = 3.6 * 10^{-3}$$

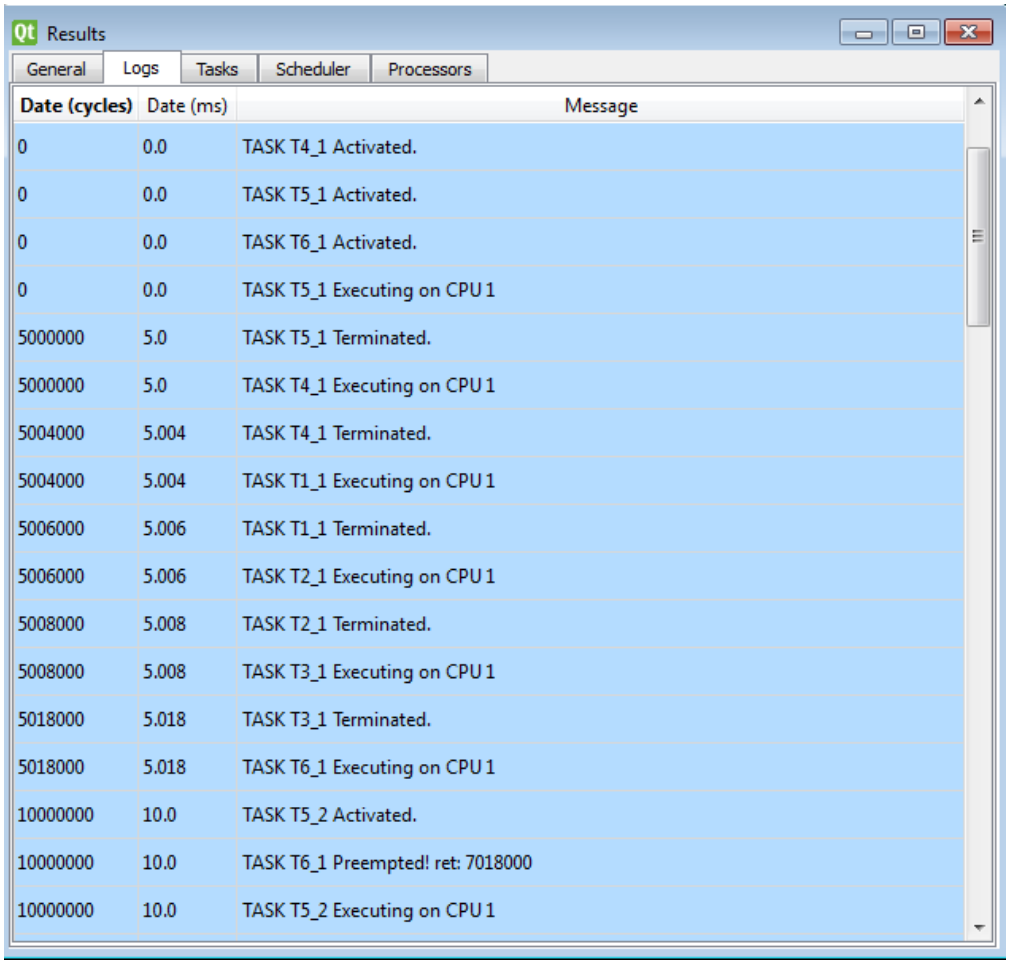
$$w_2(50) = 0.002 + \frac{50}{50} * 0.002 = 4 * 10^{-3}$$

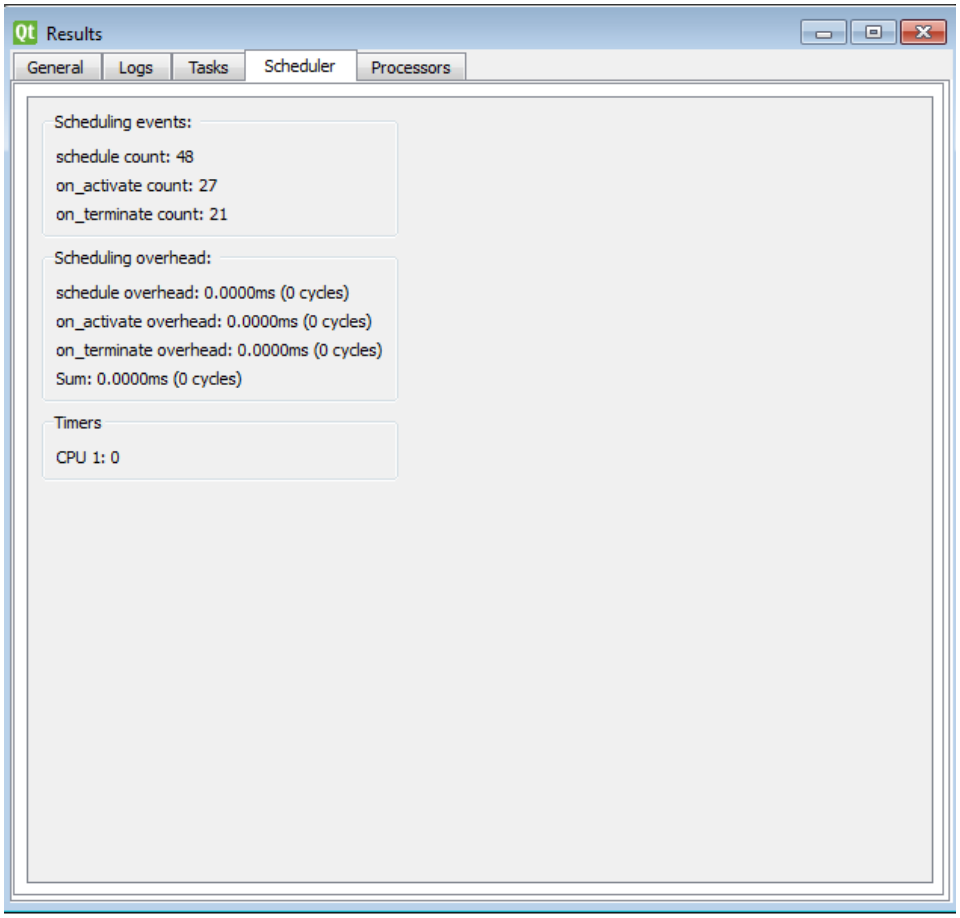
- ✓ $w_2(50) < D_2$ → task 2 is schedulable.
- ✓ And so on for rest of tasks…

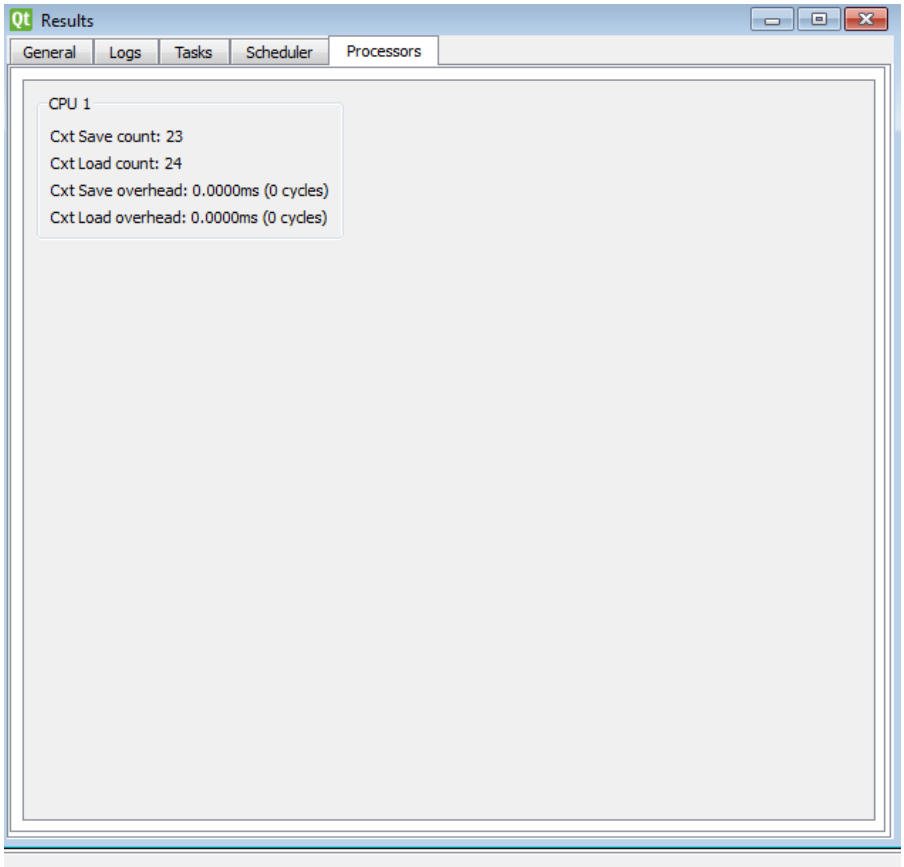
4. Simso simulation :




Results
General
Logs
Tasks
Scheduler
Processors
Observation Window:
from 0.00 to 100.00 ms
Configure...
Total load
Payload
System load
CPU 1
0.6204
0.6204
0.0000
Average
0.6204
0.6204
0.0000


Results
General
Logs
Tasks
Scheduler
Processors
Date (cycles)
Date (ms)
Message
0 0.0 TASK T4_1 Activated.
0 0.0 TASK T5_1 Activated.
0 0.0 TASK T6_1 Activated.
0 0.0 TASK T5_1 Executing on CPU 1
5000000 5.0 TASK T5_1 Terminated.
5000000 5.0 TASK T4_1 Executing on CPU 1
5004000 5.004 TASK T4_1 Terminated.
5004000 5.004 TASK T1_1 Executing on CPU 1
5006000 5.006 TASK T1_1 Terminated.
5006000 5.006 TASK T2_1 Executing on CPU 1
5008000 5.008 TASK T2_1 Terminated.
5008000 5.008 TASK T3_1 Executing on CPU 1
5018000 5.018 TASK T3_1 Terminated.
5018000 5.018 TASK T6_1 Executing on CPU 1
10000000 10.0 TASK T5_2 Activated.
10000000 10.0 TASK T6_1 Preempted! ret: 7018000
10000000 10.0 TASK T5_2 Executing on CPU 1


Results
General
Logs
Tasks
Scheduler
Processors
Scheduling events:
schedule count: 48
on_activate count: 27
on_terminate count: 21
Scheduling overhead:
schedule overhead: 0.0000ms (0 cycles)
on_activate overhead: 0.0000ms (0 cycles)
on_terminate overhead: 0.0000ms (0 cycles)
Sum: 0.0000ms (0 cycles)
Timers
CPU 1: 0


Results
General
Logs
Tasks
Scheduler
Processors
CPU 1
Cxt Save count: 23
Cxt Load count: 24
Cxt Save overhead: 0.0000ms (0 cycles)
Cxt Load overhead: 0.0000ms (0 cycles)

5. logic analyzer :