Analytical Methods

This document includes analytical methods of verifying EDF Scheduler based on FreeRTOS.

1. System Hyperperiod

- This project contains 6 tasks as follows :

Task	Period	Execution Time
Button_1_Monitor	50ms	13us
Button_2_Monitor	50ms	13.2us
Periodic_Transmitter	100ms	17.5us
Uart_Receiver	20ms	27.35us
Load_1_Simulation	10ms	5ms
Load_2_Simulation	100ms	12ms

- Using the above table, we can easily calculate the Hyperperiod for these tasks (the period after which all tasks repeat execution again)

Hyperperiod = 100ms

2. CPU Load

- To calculate the CPU load, we need to calculate the execution time for each task multiplied by number of times these tasks came through one hyperperiod, then by summing these times and divide by the Hyperperiod we get the CPU load.
- Using the same above table, we get :

CPU load =

$$\frac{[(13 * 2) + (13.2 * 2) + 17 \cdot 5 + (27.35 * 5)] * 10 - 3 + (5 * 10) + 12}{100} \times 100\%$$

CPU Load = 62.2%

3. System Schedulability

- This property can be determined using two methods:
  1. Rate-Monotonic utilization bound (only for RM Schedulers)
  2. Time demand analysis
- Let's start with Rate-Monotonic method

Rate Monotonic Utilization Bound

$$U = \sum_{i=1}^{n} U_i = \sum_{i=1}^{n} \frac{C_i}{T_i} \leq n\left(2^{1/n} - 1\right)$$

- There are two side of the above equation
  1. The right-hand side calculates the summation of ratio between the execution time of a task and the periodicity of that task
  2. The left-hand side is called URM which is considered the metric of the system schedulability
- By comparing these two sides of equation :
  1. if the right-hand side is less than or equal to the URM term, system is schedulable
  2. Otherwise, system is not schedulable
- By Applying this equation on our system :

- The right-hand side :

$$\frac{13*10^{-3}}{50} + \frac{13.2*10^{-3}}{50} + \frac{17.5*10^{-3}}{100} + \frac{27.35*10^{-3}}{20} + \frac{5}{10} + \frac{12}{100} = 0.622$$

- The left-hand side (URM) :

$$6\left(2^{\frac{1}{6}} - 1\right) = 0.73477$$

- So, $0.622 \leq 0.73477$ $\rightarrow$ System is Schedulable

Time Demand Analysis

• This method measures the time required against the time provided for each task

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

• Let's reorder our tasks based on priorities to match Rate-Monotonic scheduler rules

- tasks **w**ith higher periodicity **(**comes faster**)** take higher priorities

Priority	Task	Period	Execution Time
0	Load_1_Simulation	10ms	5ms
1	Uart_Receiver	20ms	27.35us
2	Button_1_Monitor	50ms	13us
2	Button_2_Monitor	50ms	13.2us
3	Periodic_Transmitter	100ms	17.5us
3	Load_2_Simulation	100ms	12ms

• Using the above equation, let's calculate response time for each task taking into consideration the effect of other tasks if they have higher priority

- Load_1_Simulation :

$W(10) = 5 + 0 = 5ms \le 5ms \rightarrow$ Schedulable

- Uart_Receiver :

$W(20) = 27.35 * 10^{-3} + 5 * \left(\frac{20}{10} \right) = 10.027ms \le 20ms \rightarrow$ Schedulable

- Button_1_Monitor :

$$W(50) = 13 * 10^{-3} + 27.35 * 10^{-3} * \frac{50}{20} + 5 * \left(\frac{50}{20} \right) =$$

$$= 25.095ms \leq 50ms \rightarrow$$ Schedulable

- Button_2_Monitor :

$$W(50) = 2 * 13 * 10^{-3} + 27.35 * 10^{-3} * \frac{50}{20} + 5 * \left(\frac{50}{20} \right) =$$

$$= 25.108ms \leq 50ms \rightarrow$$ Schedulable

- Periodic_Transmitter :

$$W(100) = (17.5 * 10^{-3}) + \left(2 * 13 * 10^{-3} * \frac{100}{50} \right) +$$

$$\left(27.35 * 10^{-3} * \frac{100}{20} \right) + \left(5 * \frac{100}{50} \right) = 25.108ms \leq 50ms \rightarrow$$ Schedulable

- Load_2_Simulation

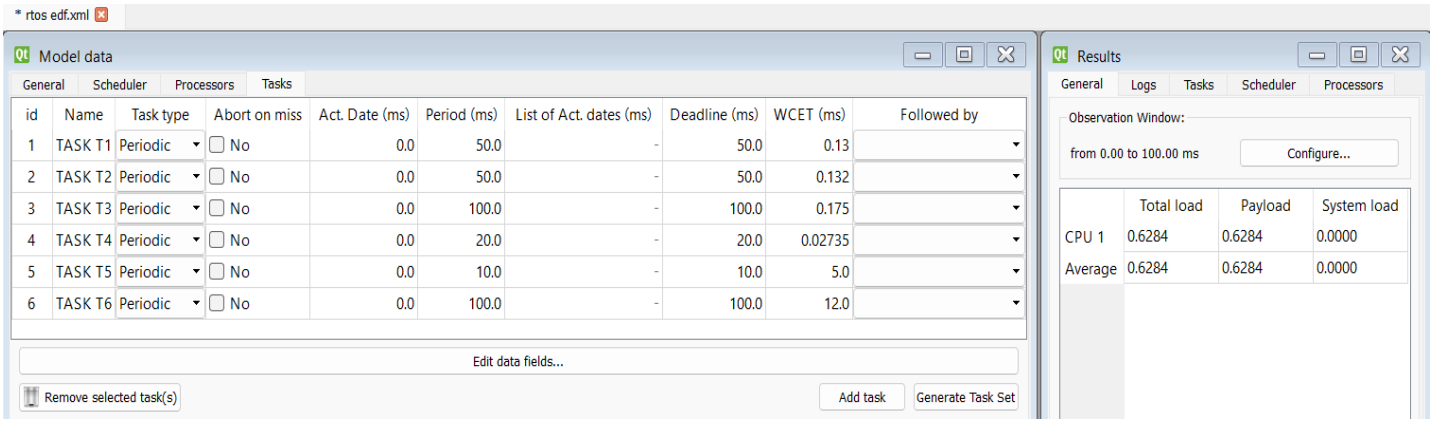
$$W(100) = 12 + (17.5 * 10^{-3}) + \left(2 * 13 * 10^{-3} * \frac{100}{50} \right) +$$

$$\left(27.35 * 10^{-3} * \frac{100}{20} \right) + \left(5 * \frac{100}{50} \right) = 25.108ms \leq 50ms \rightarrow$$ Schedulable

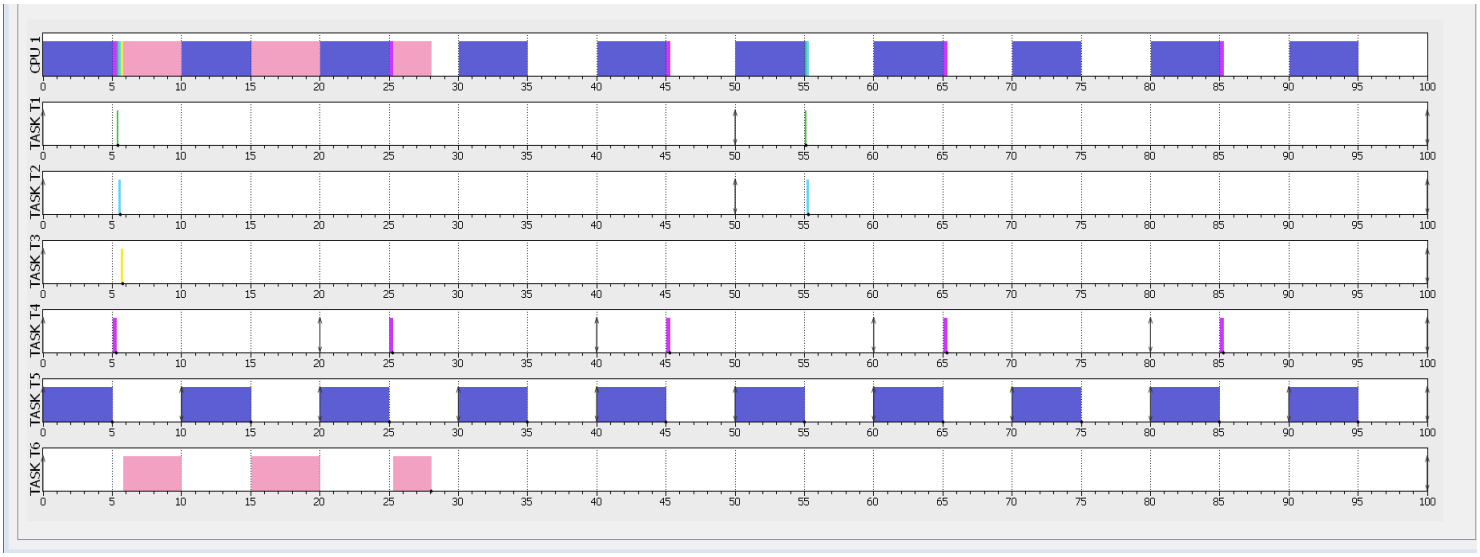
- Using the above results, all tasks are schedulable → System is Schedulable

SIMSO Offline Results

In this part we will show simso offline simulator results for our project including same tasks discussed above in the previous part. This step mainly purpose is to verify the above results of analytical approach and see if they match or not.



As we see that CPU load = 62% which similar to analytical approach



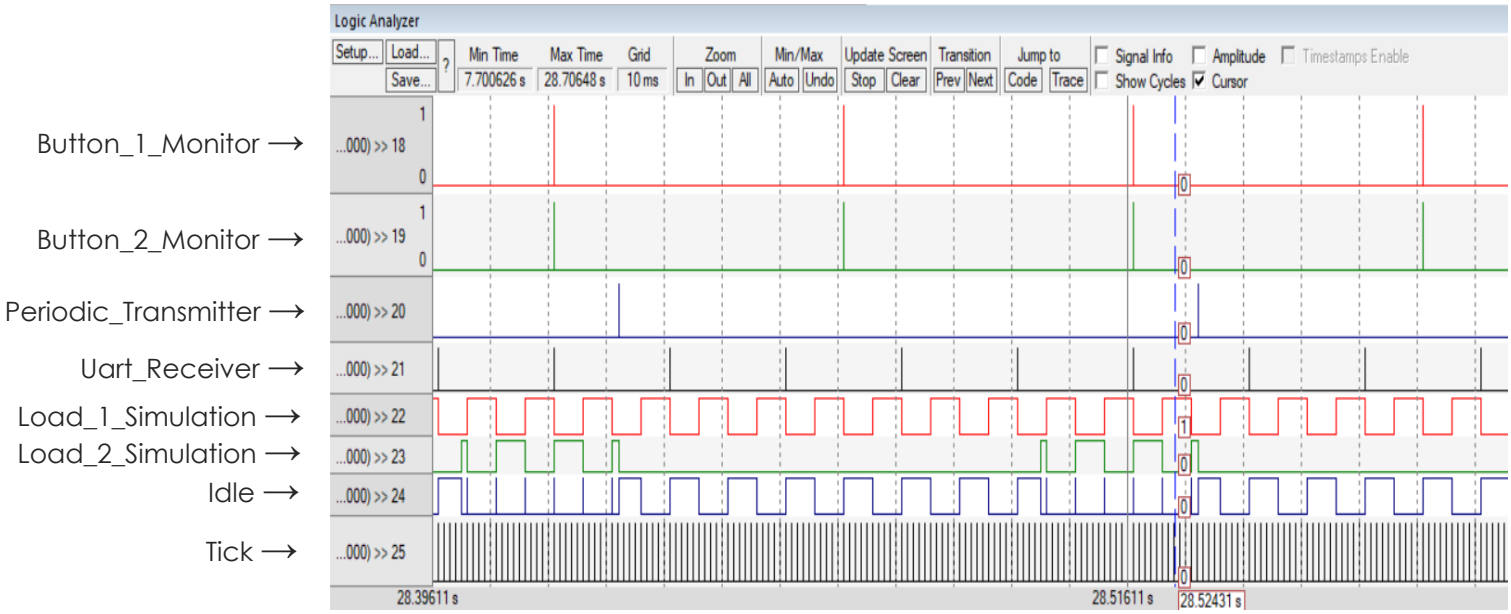
- Here is the whole situation including all tasks

- We can see the case of executing task6 while task5 came (higher priority and earlier deadline) , so task5 preempted task6 and this happens three times every execution of task6

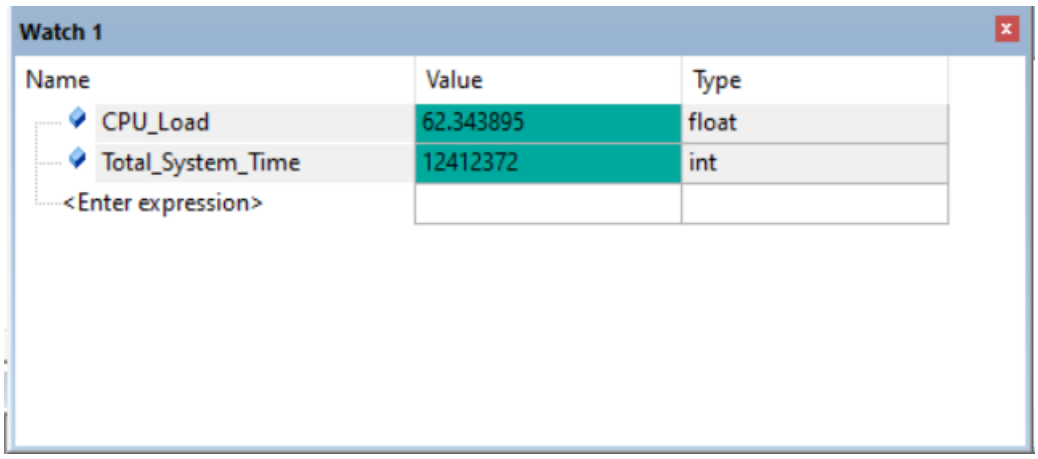
- Similarly with rest tasks we can see each task and when to execute

Keil Simulator Results

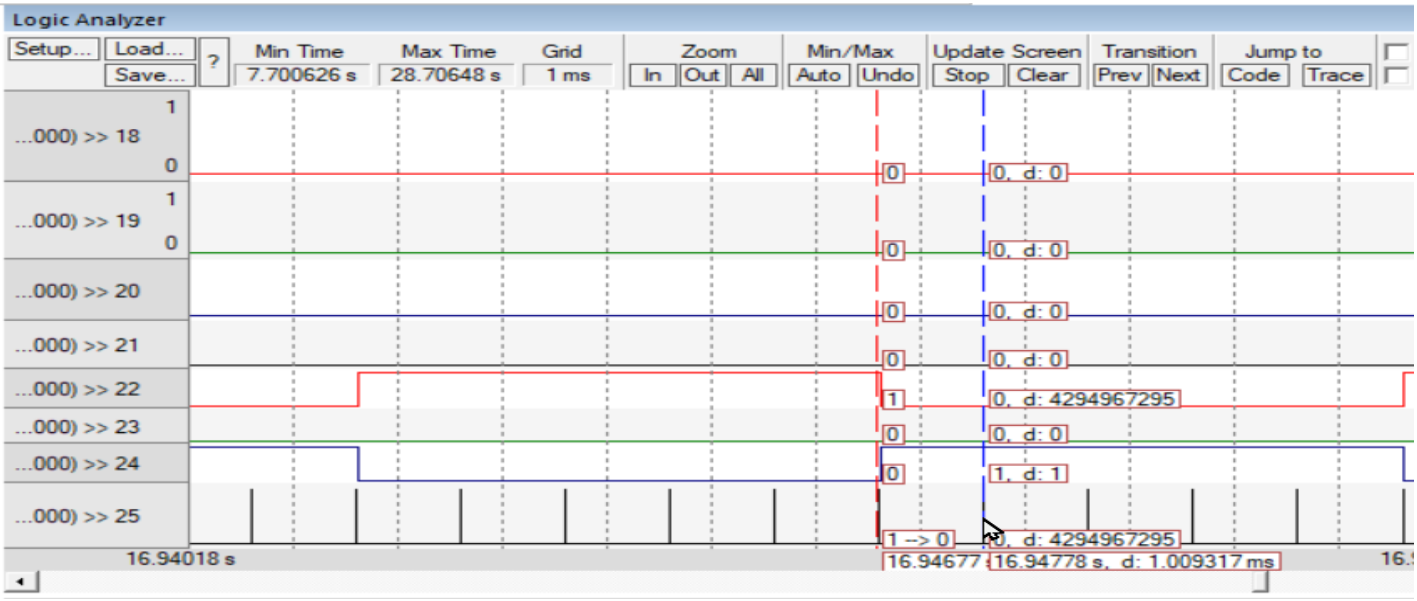
In this part we will show Keil simulator results for our project including same all tasks discussed above in the two previous parts. Using timer1 and macro tracing we will calculate the CPU usage (load) and verify that all tasks run well with correct time constraints.



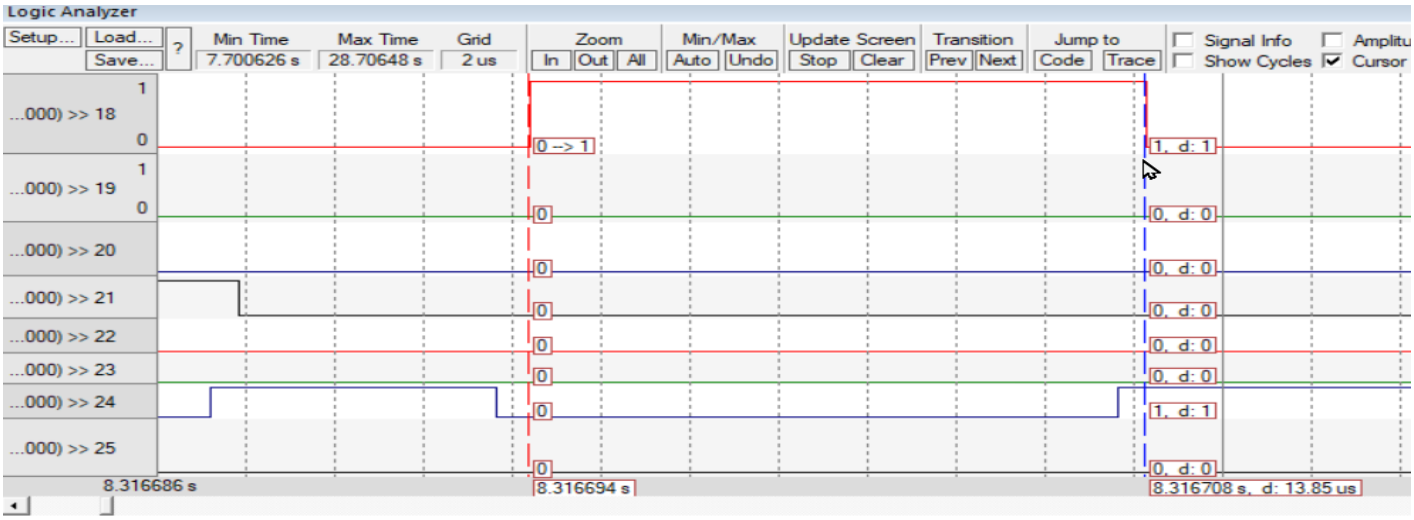
Here we can see that CPU_Load is about 62.5% (saturate between 62% and 63%).
This result matches our previous two approaches (analytical and simso simulator)



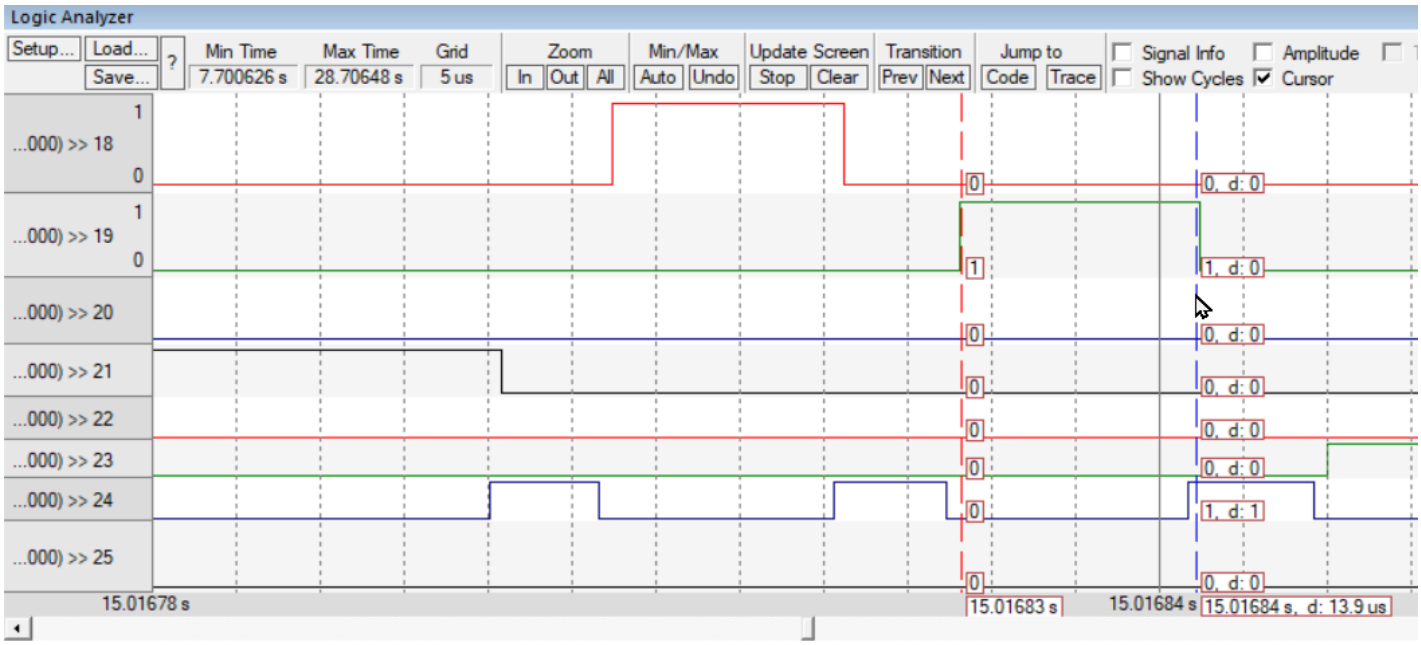
This shows the execution time of Tick in run time. Which is around 1ms.



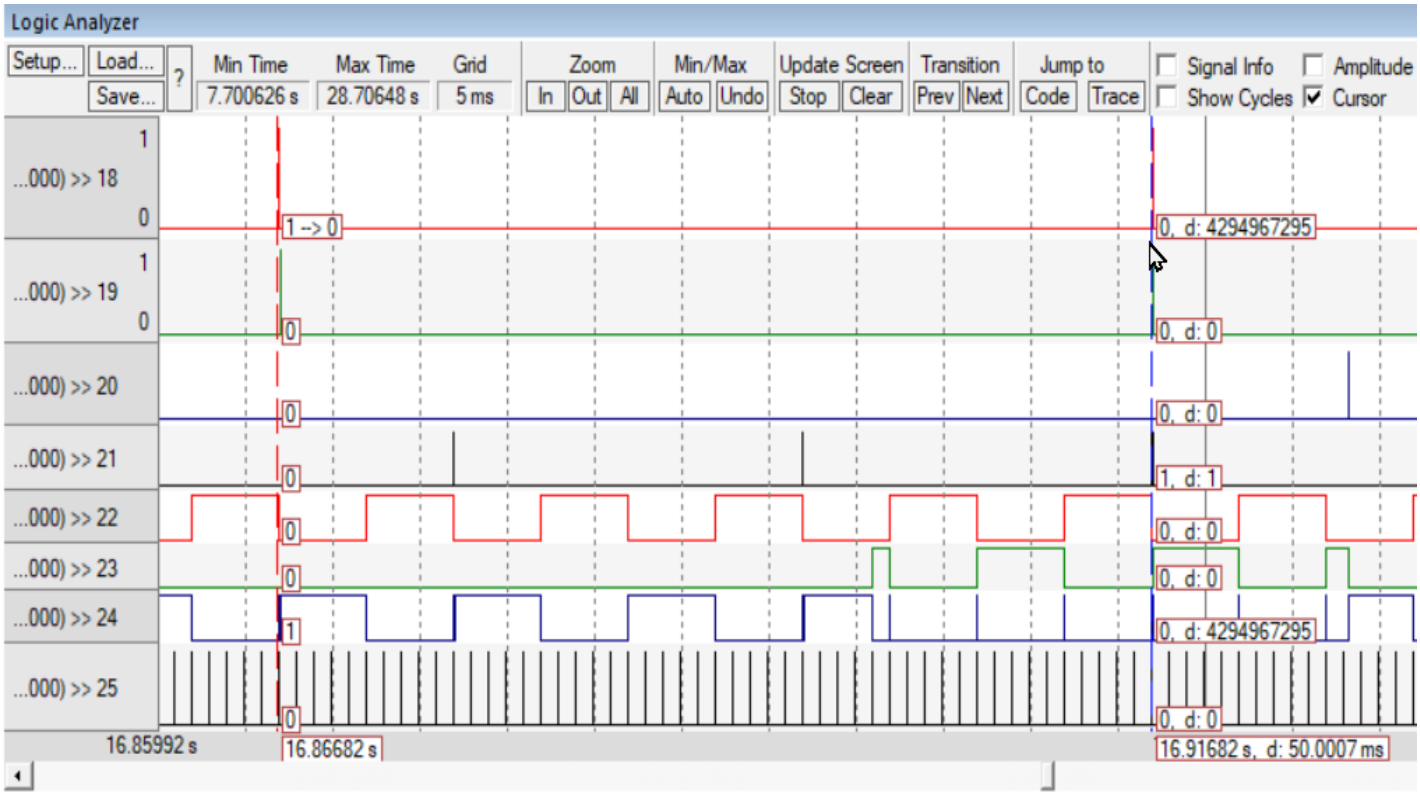
This shows the execution time of Button_1_Monitor task in run time. Which is around 13us.



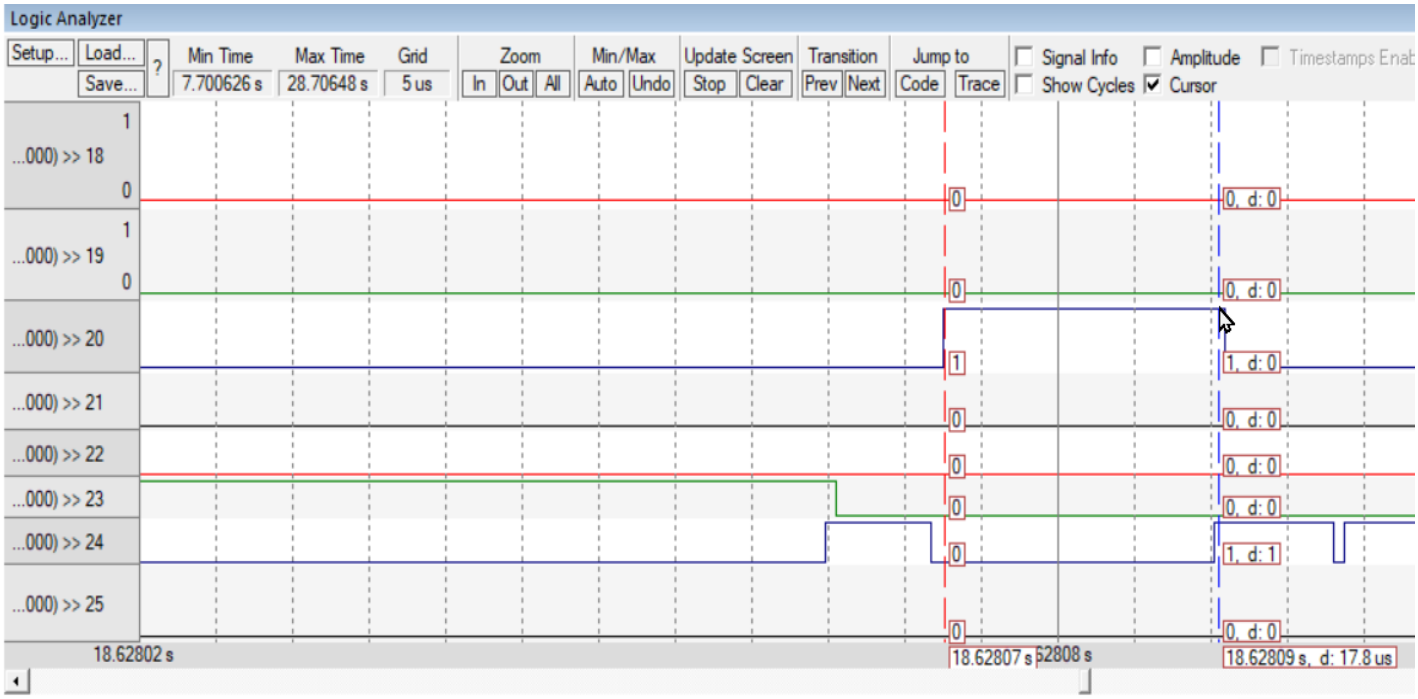
This shows the execution time of Button_2_Monitor task in run time. Which is around 13us.



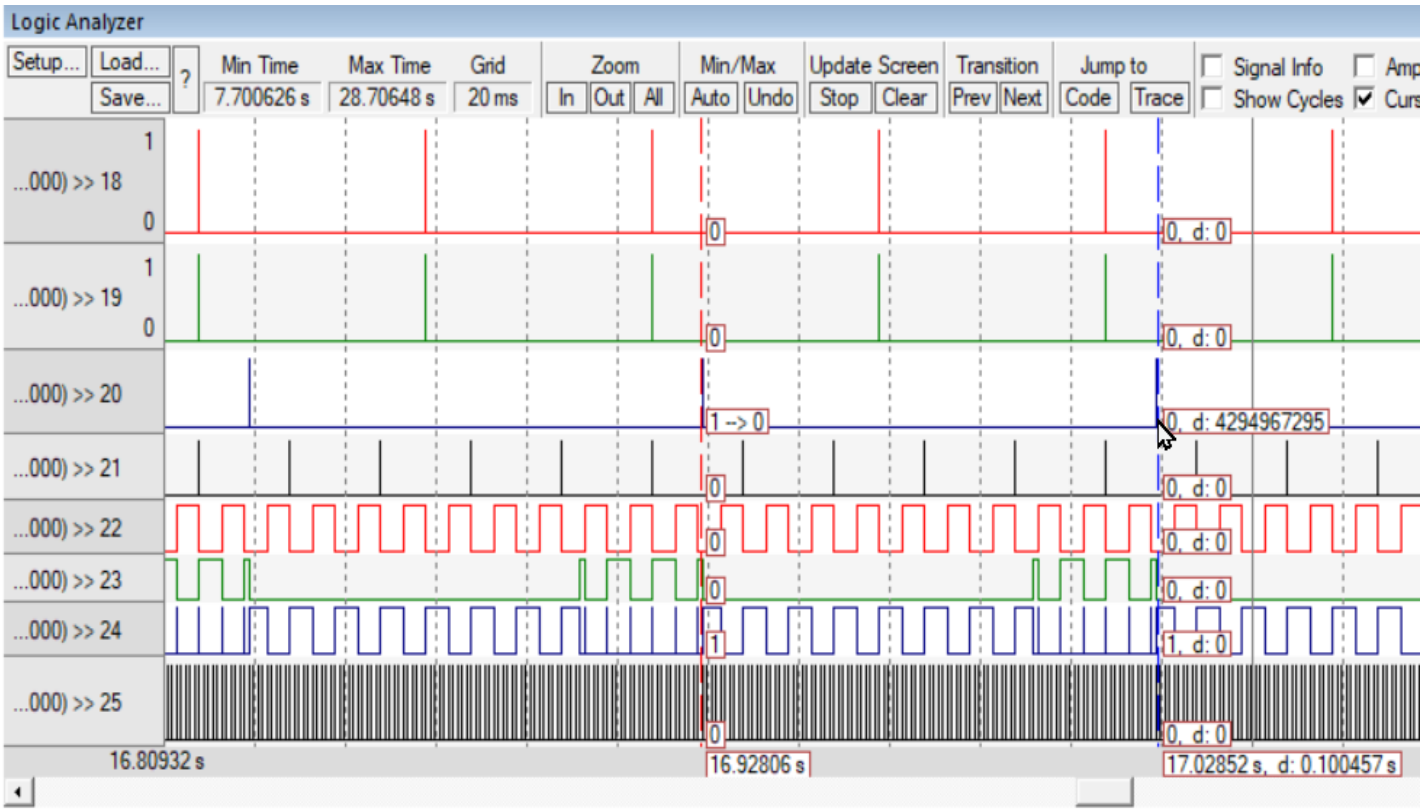
This shows That both Button_1_Monitor and Button_2_Monitor tasks periodicity is about 50ms



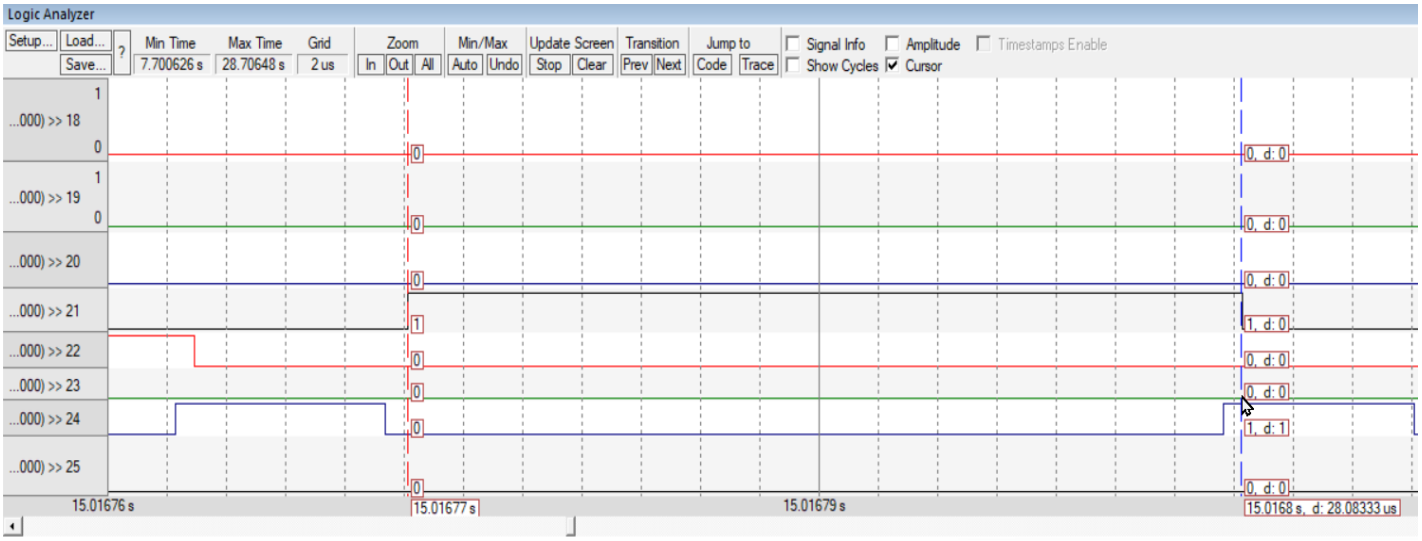
This shows the execution time of Periodic_Transmitter task in run time. Which is around 17.5us.



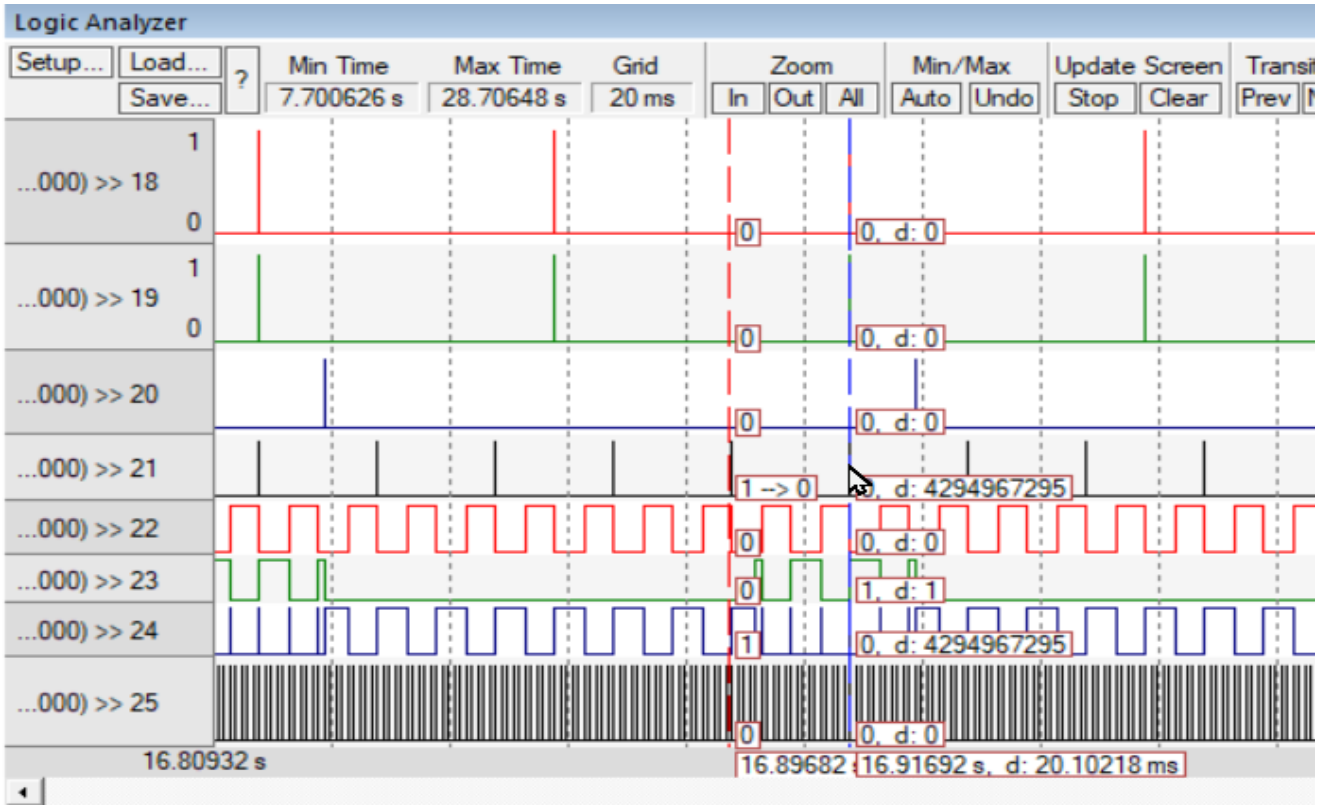
This shows the periodicity of Periodic_Transmitter task Which is around 100ms



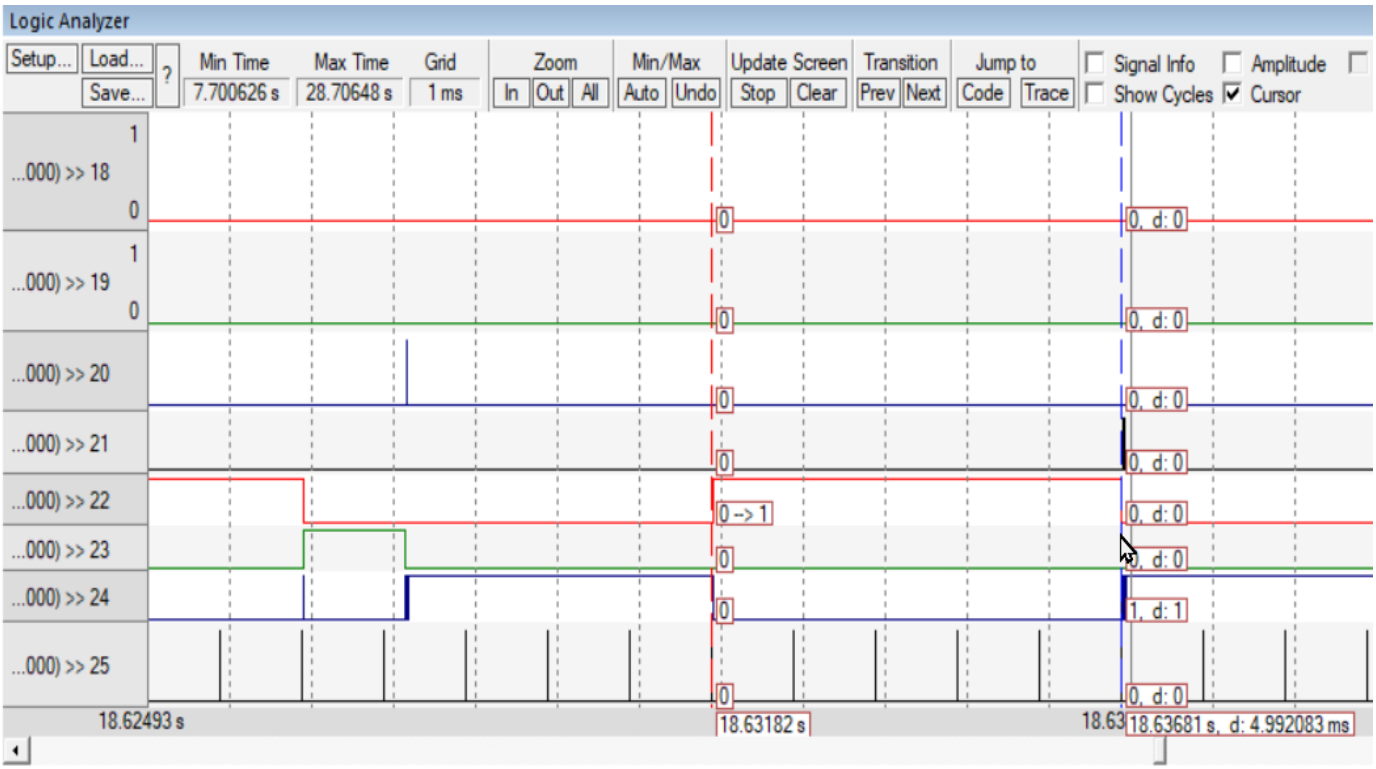
This shows the execution time of Uart_Receiver task in run time. Which is around 27.5us.



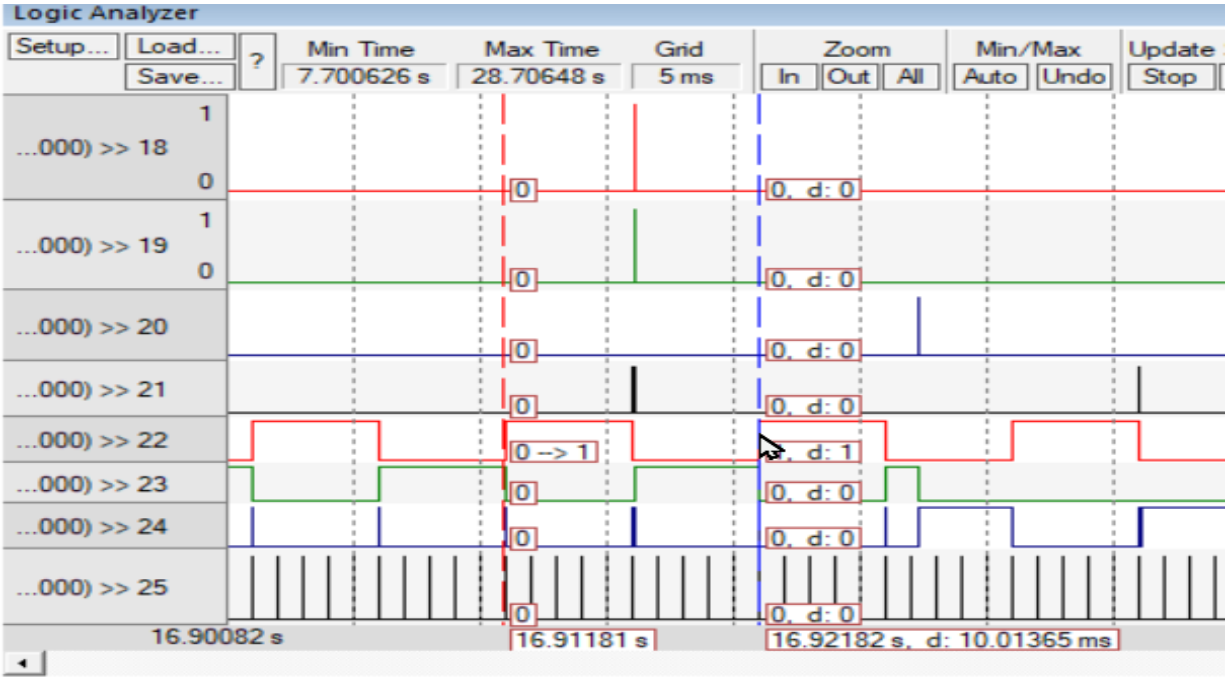
This shows the periodicity of Uart_Receiver task Which is around 20ms



This shows the execution time of Load_1_Simulation task in run time. Which is around 5ms as required.



This shows the periodicity of Load_1_Simulation task Which is around 10ms



This case shows that Load_2_Simulation task (the blue one) is preempted by the above task (Load_2_Simulation) three times, so total execution time for Load_2_Simulation task = 27 – (3*5) = 12ms which equals the time required in specifications.

This one shows the periodicity of Load_2_Simulation task (the blue one) which is 100ms.