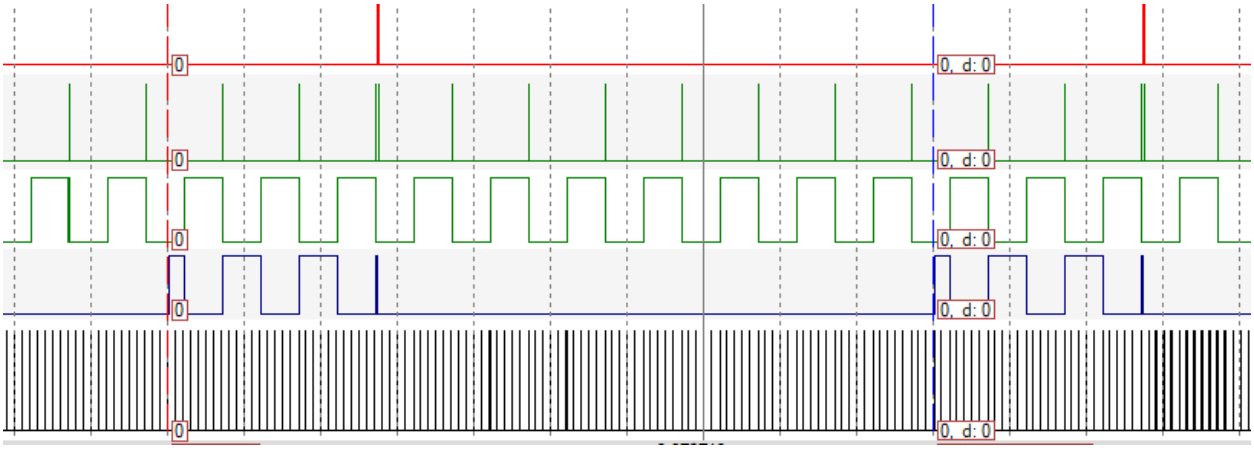
1. Analytical methods

A. The system hyper period = 100 ms calculated from analyzer and using basic logic as the largest period is 100 ms and all other periods are factors of 100 (10,50) then the cycle will repeat itself in 100s



B. Calculate the CPU load (first fully analytical, beneath it using freertos api)

From Value below total Task execution time per hyperloop=

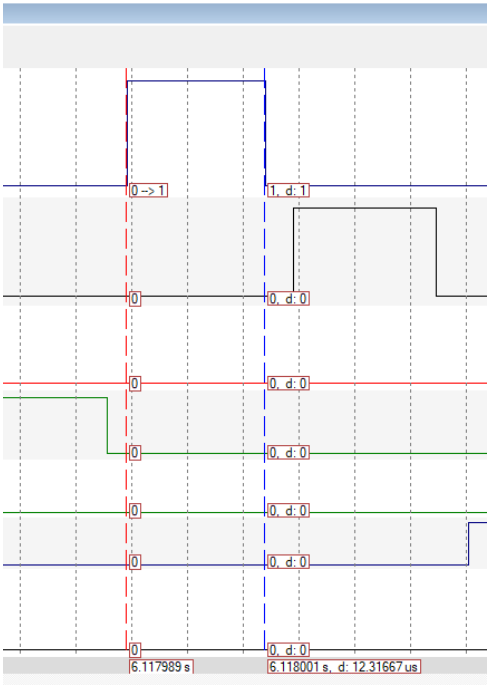
$$\frac{TASK1 \times hyper_{per}}{TASK1_{per}} + \frac{TASK2 \times hyper_{per}}{TASK2_{per}} + \frac{TASK3 \times hyper_{per}}{TASK3_{per}} + \cdots =$$

$$\frac{0.0123 \times 100}{50} + \frac{0.0128 \times 100}{50} + \frac{0.0207 \times 100}{100} + \frac{0.055 \times 100}{20} + \frac{5 \times 100}{10} + \frac{12 \times 100}{100}$$

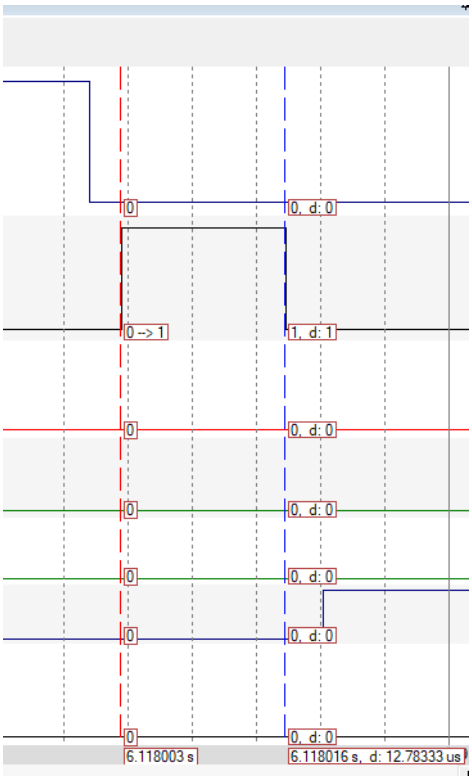
$$= 62.3459\ ms$$

$$CPU\ load = \frac{Total\ excecution\ time}{hyper\ period} = 62.3459\%$$

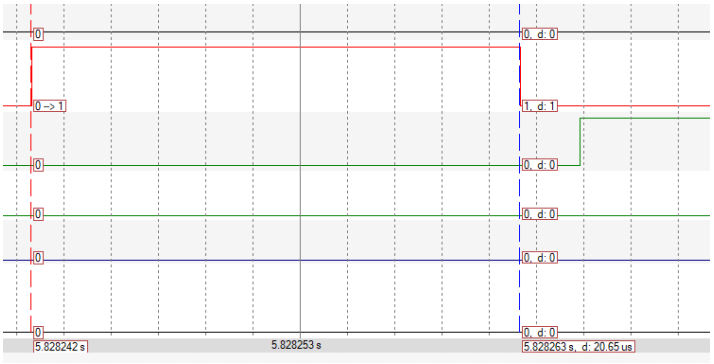
TASK1 execution time=12.3 μs

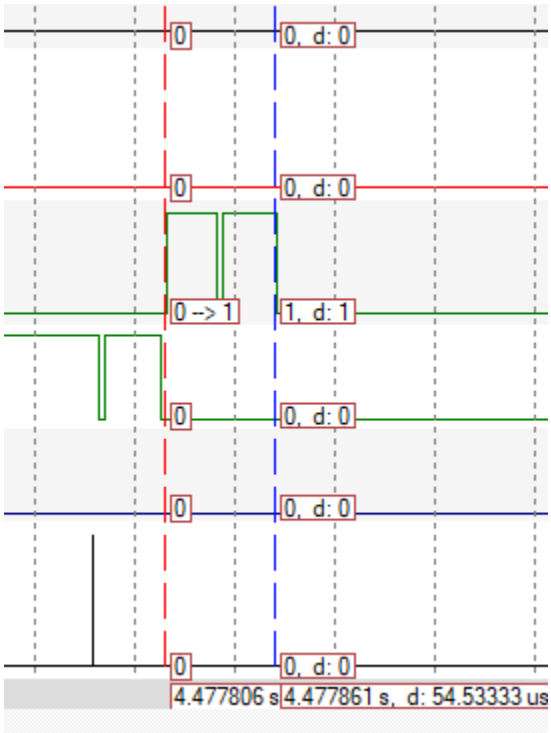


TASK2 Execution time=12.788μs



TASK 3 Execution Time=20.7μs





TASK 4 execution time=55 μs
TAASK 5 execution time= 5 ms
TASK 6 execution time= 12 ms

Using vTaskGetRunTimeStats(state_Buffer);
We find

Button_ 1 9	<1%	task 1
Button_ 2 1	<1%	task 2
Periodi 26	<1%	task 3
Uart_Re 580	<1%	task 4
Load_1_ 52300	51%	task 5
Load_2_ 12732	12%	task 6
Total	around 63 %	

C. Schedulablity assuming rate monotonic

Method 1:

$$\Sigma \frac{C_i}{T_i} \leq n(2^{\frac{1}{n}} - 1)$$

$$\frac{0.0123}{50} + \frac{0.0128}{50} + \frac{0.0207}{100} + \frac{0.055}{20} + \frac{5}{10} + \frac{12}{100} \leq 6 \times \left(2^{\frac{1}{6}} - 1\right)$$

$$0.623 \leq 0.73$$

The condition proves to be true the system is schedulable using rate monotonic scheduling

Method 2: Time Demand Analysis

$$w_i(t) = e_i + \Sigma \left\lceil \frac{t}{p_k} \right\rceil e_k$$

As The least period is the highest priority then Task 5 with 10 ms period will start first

$$w_1(1) = 5, w_1(2) = 5, w_1(3) = 5, w_1(4) = 5, w_1(5) = 5 \ldots\ldots w_1(10) = 5,$$

The second least is TASK4 with 20ms period

$$w_2(1) = 5 + 0.055 = 5.055 \ldots\ldots\ldots\ldots\ldots\ldots\ldots w_2(10) = 5.055$$

$$w_2(11) = 10 + 0.055 = 10.055, \ldots\ldots\ldots\ldots\ldots w_2(20) = 10.055$$

The third least task is either TASK1 or 2 at 50 ms period

Taking Task1 as the highest priority

$$w_3(1) = 5.055 + 0.0123 \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots w_3(10) = 5.0673$$

$$w_3(11) = 10.0673 \ldots\ldots\ldots\ldots\ldots\ldots w_3(20) = 10.0673$$

$$w_3(21) = 15.1223 \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots w_3(30) = 15.1223$$

$$w_3(31) = 20.1223 \ldots\ldots\ldots\ldots\ldots\ldots w(40) = 20.1223$$

$$w(41) = 25.1773 \ldots\ldots\ldots\ldots\ldots\ldots\ldots w(50) = 25.1773$$

Taking Task2 as the next highest priority we find

$$w_4(1 \rightarrow 10) = 5.08, \quad w_4(11 \rightarrow 20) = 10.08, w_4(21 \rightarrow 30) = 15.1351$$

$$, w_4(31 - 40) = 20.1351, w_4(41 \rightarrow 50) = 25.19$$

The next periodicity at 100 ms has two tasks Task 6 and Task 3

Taking Task 3 as the highest priority

$$w_5(1 \rightarrow 10) = 5.1, \quad w_5(11 \rightarrow 20) = 10.1, w_5(21 \rightarrow 30) = 15.1558$$

$$, w_5(31 - 40) = 20.1558, w_5(41 \rightarrow 50) = 25.21, w_5(51 \rightarrow 60) = 30.243,$$

$$w_5(61 \rightarrow 70) = 35.298, w_5(71 \rightarrow 80) = 40.298, w_5(81 \rightarrow 90) = 45.353$$

$$w_5 \rightarrow (91 - 100) = 50.353$$

Lowest priority Task 6

$$w_6(1 \rightarrow 10) = 17.1, \quad w_6(11 \rightarrow 20) = 22.1, w_6(21 \rightarrow 30) = 27.1558$$

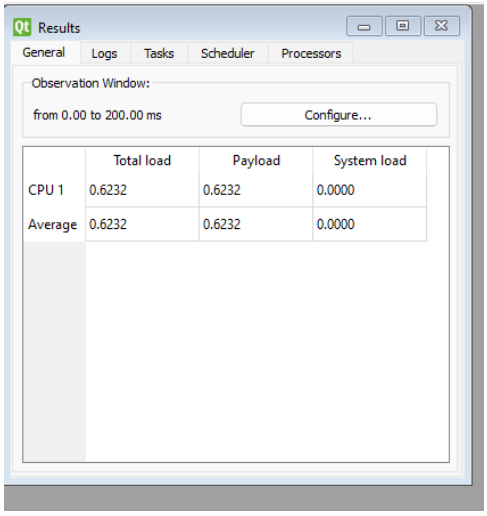
$$, w_6(31 - 40) = 32.1558, w_6(41 \rightarrow 50) = 37.21, w_6(51 \rightarrow 60) = 42.243,$$

$$w_6(61 \rightarrow 70) = 47.298, w_6(71 \rightarrow 80) = 52.298, w_6(81 \rightarrow 90) = 57.353$$

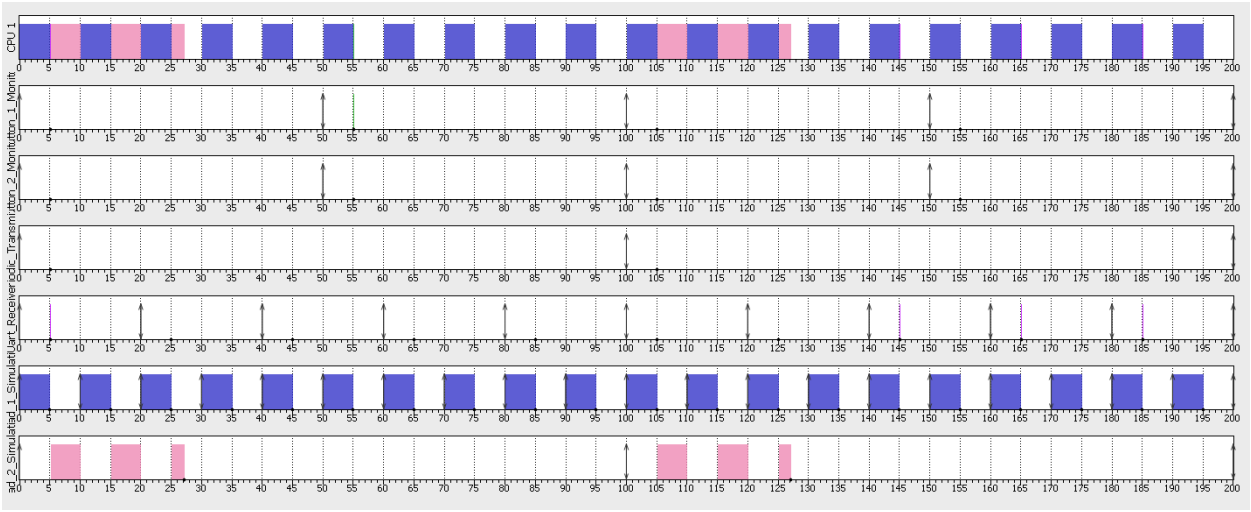
$$w_6 \rightarrow (91 - 100) = 62.353$$

As no task missis its deadline then all tasks are schedulable

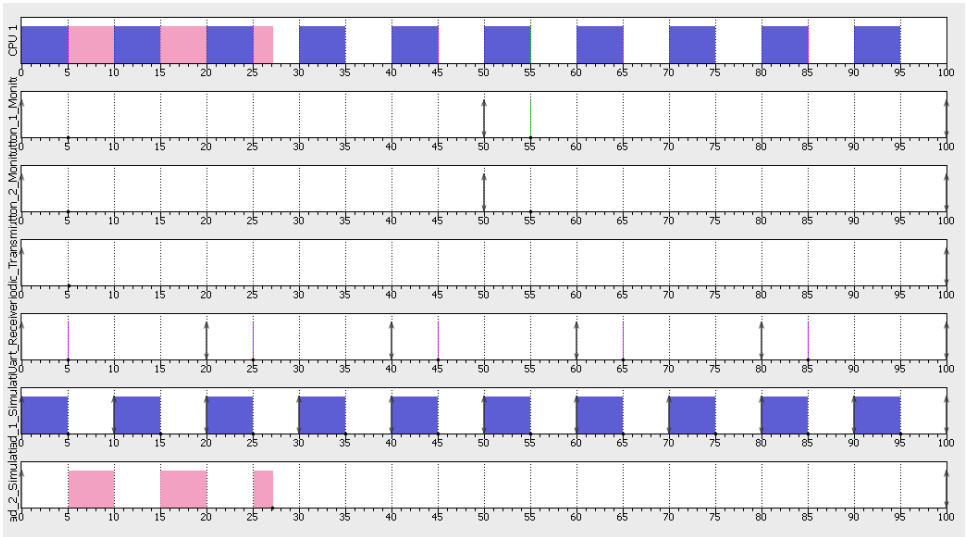
2. Using Simso offline simulator, simulate the given set of tasks assuming:



200 ms simulation(2 hyper periods):



100 ms simulation (1 hyper period)



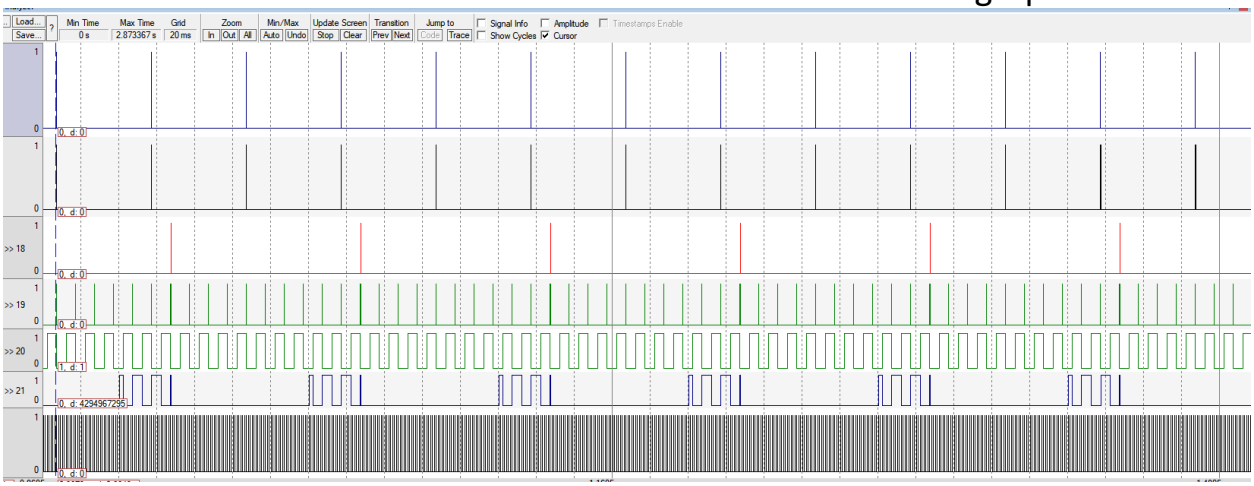
3. Using Keil simulator in run-time and the given set of tasks:

A. Calculate the CPU usage time using timer 1 and trace macros:
Using vTaskGetRunTimeStats(state_Buffer);
We find

Button_ 1 9	<1%	task 1
Button_ 2 1	<1%	task 2
Periodi 26	<1%	task 3
Uart_Re 580	<1%	task 4
Load_1_ 52300	51%	task 5
Load_2_ 12732	12%	task 6
Total	around	63 %

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

This is TASK1 to TASK 6 in order then the tick counter in the last graph



4. Comment on the results:

After testing and debugging It is now working as expected.
So I would consider this a successful implementation of the EDF schedular
I would like to note that the implementation here is made with assumption that the tasks are schedulable and if it is not one or more tasks will not be activated through the implementation while the remaining tasks will run normally without them.