

**Hesham Hesham**

**EDF Scheduler application report**

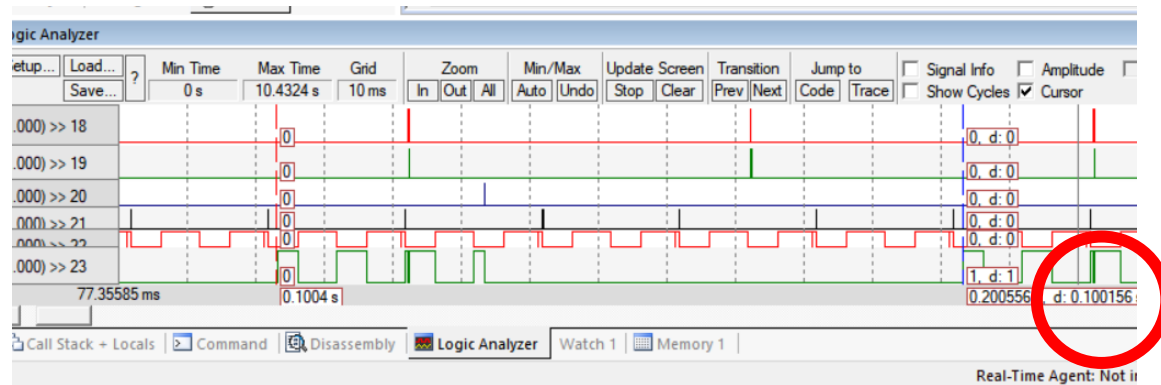
## Table of Contents

System Parameters Using Logic Analyzer .....	3
HyperPeriod .....	3
Task 1 .....	3
Task 2 .....	4
Task 3 .....	4
Task 4 .....	4
Task 5 and Task 6 .....	5
Task 5 .....	5
Task 6 .....	7
Calculations and implementation .....	9
CPU Load .....	9
CPU Load Calculated be simso application .....	9
CPU Load By Calculations.....	12
CPU Load By FreeRTOS APIs.....	12
System Schedulability .....	13
URM .....	13
Time Demand Analysis.....	13
Overview of the system running.....	15
Conclusions .....	15

## System Parameters Using Logic Analyzer

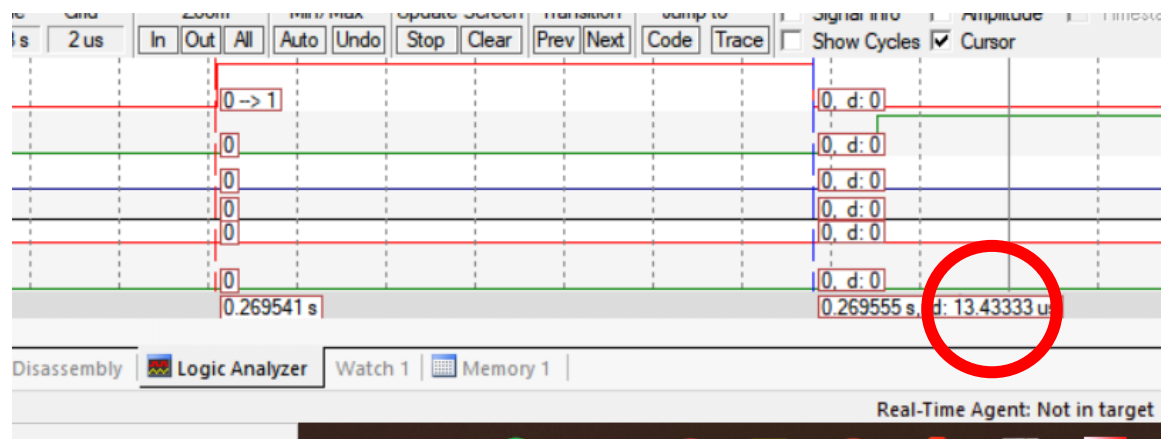
HyperPeriod

**Total hyper period is 100 ms**



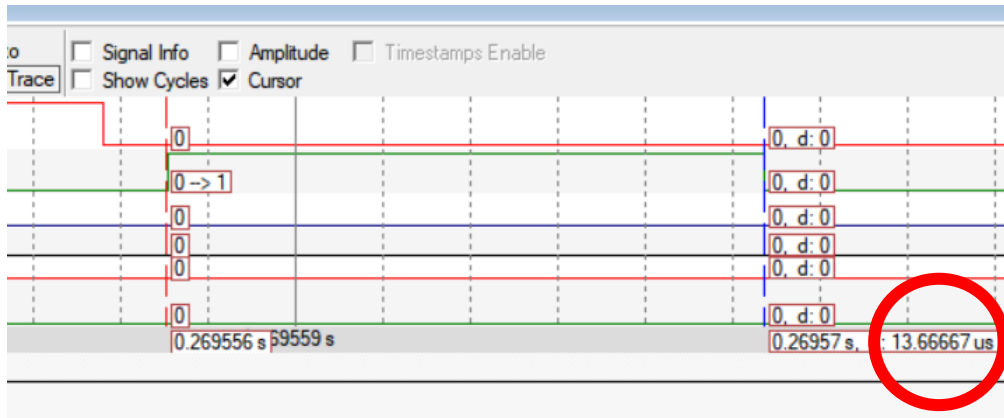
Task 1

**Execution Time = 0.0134 ms**



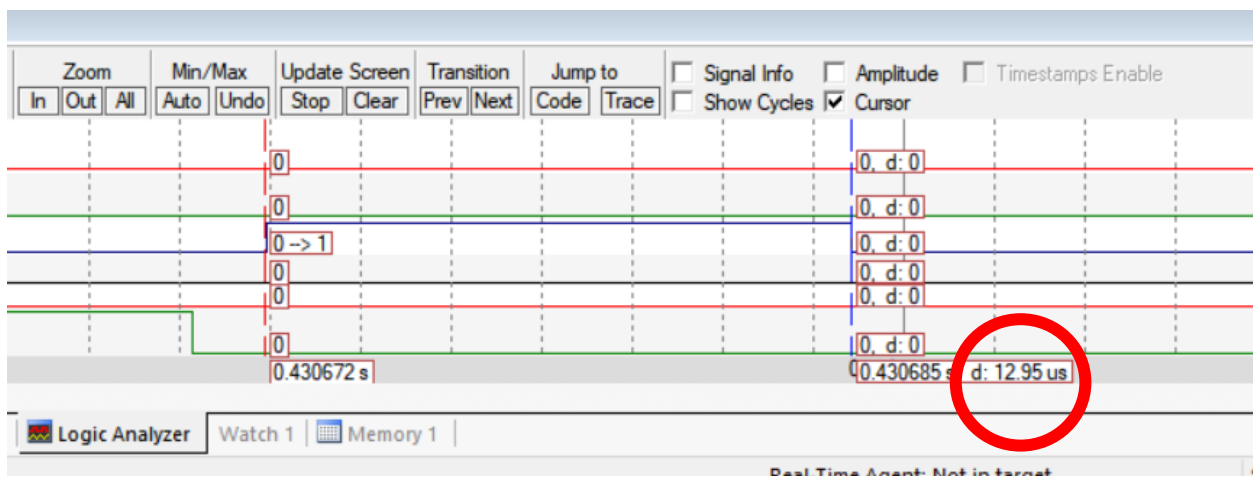
## Task 2

**Execution Time = 0.01367 ms**



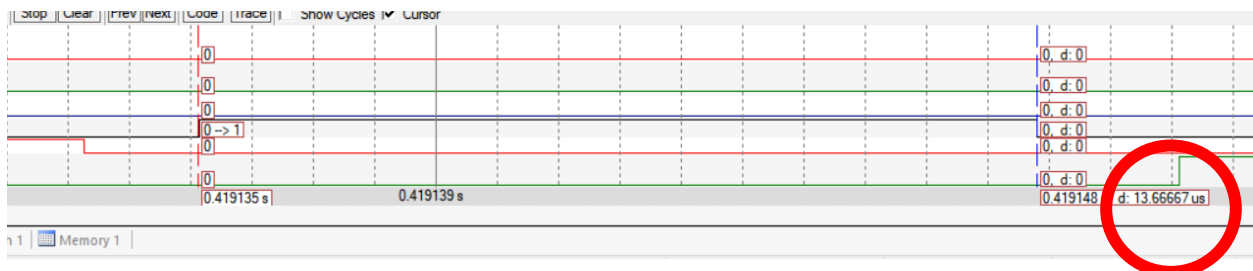
## Task 3

**Execution Time = 0.01295 ms**



## Task 4

**Execution Time = 0.01367 ms**

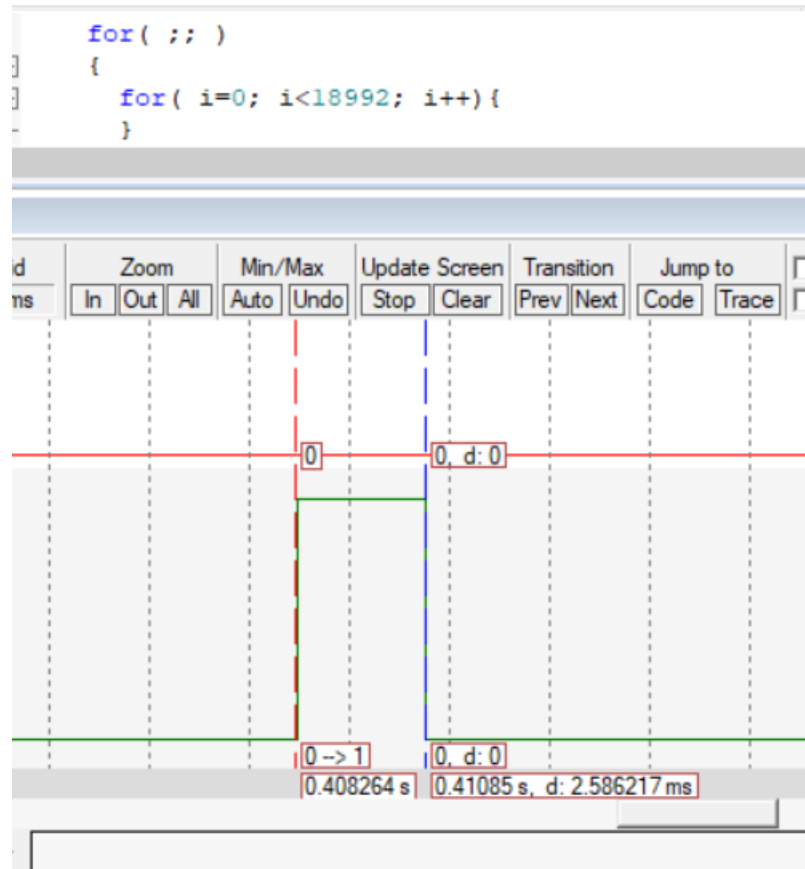
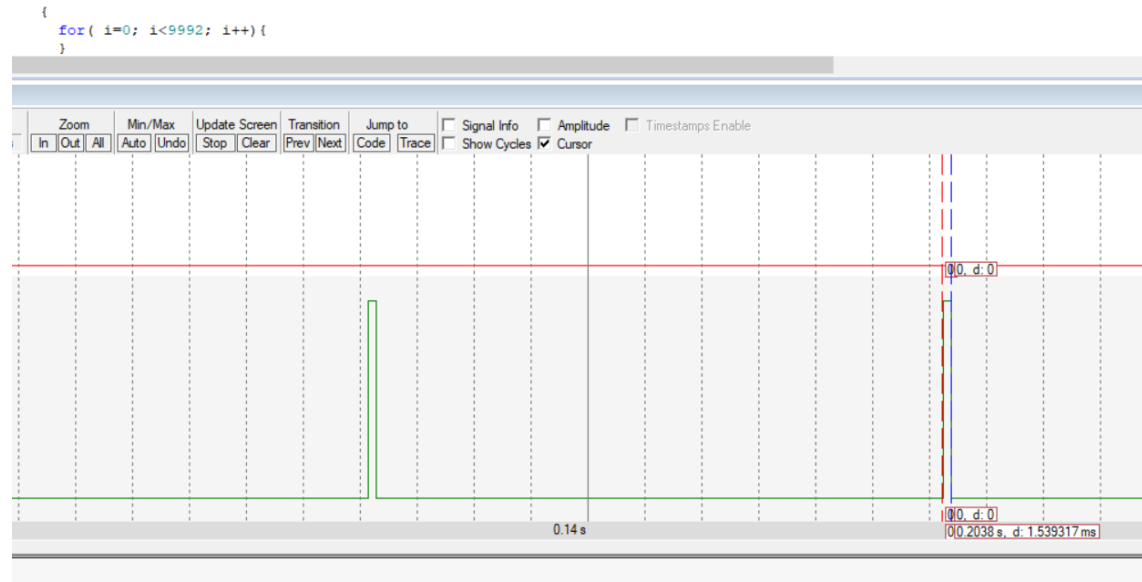


## Task 5 and Task 6

### Task 5

Task 5 is required to be 5 ms in execution time

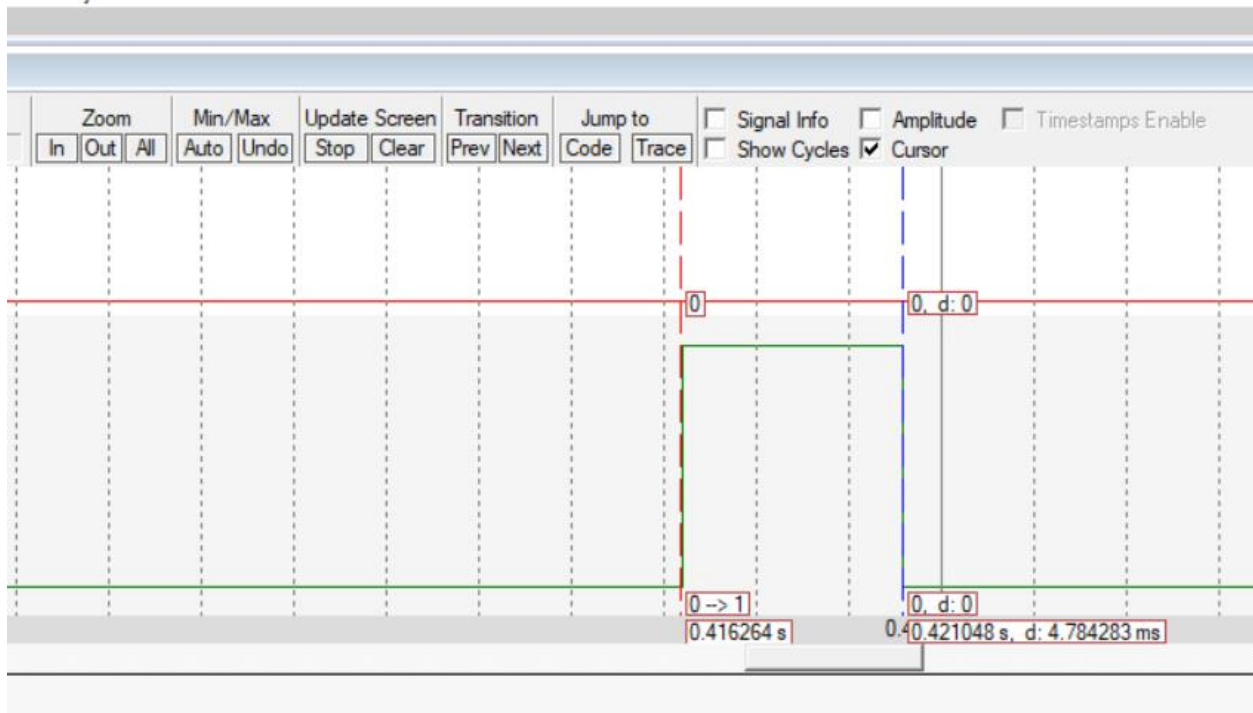
Trials to get exactly 5ms by changing the number of loops in the (for loop) and observing the logic analyser :



```

202 \ \ \ /
{
  for( i=0; i<35500; i++){
}

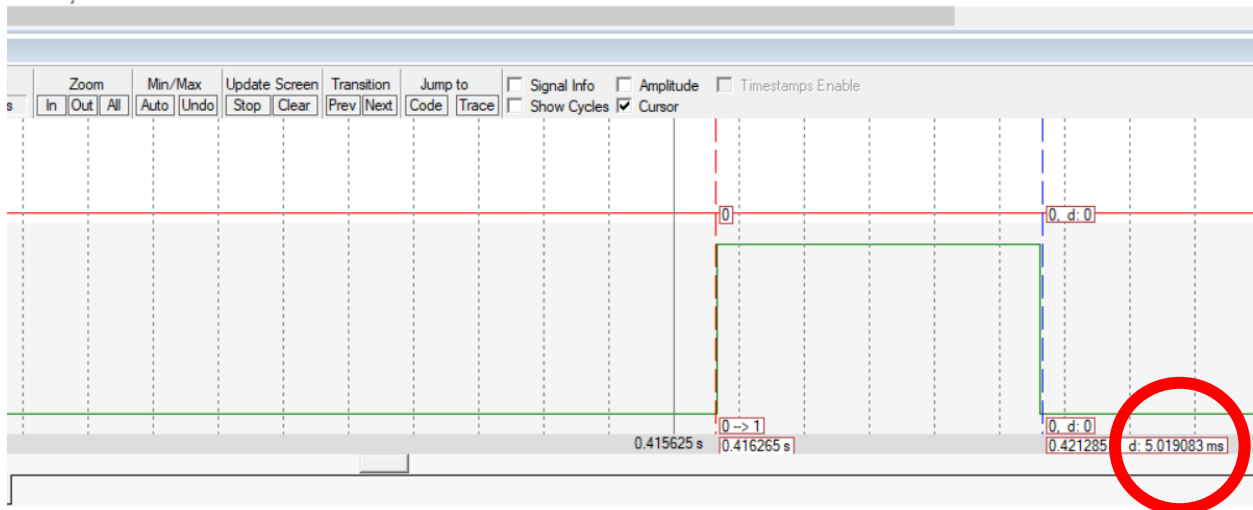
```



```

{
  for( i=0; i<37000; i++){
}

```

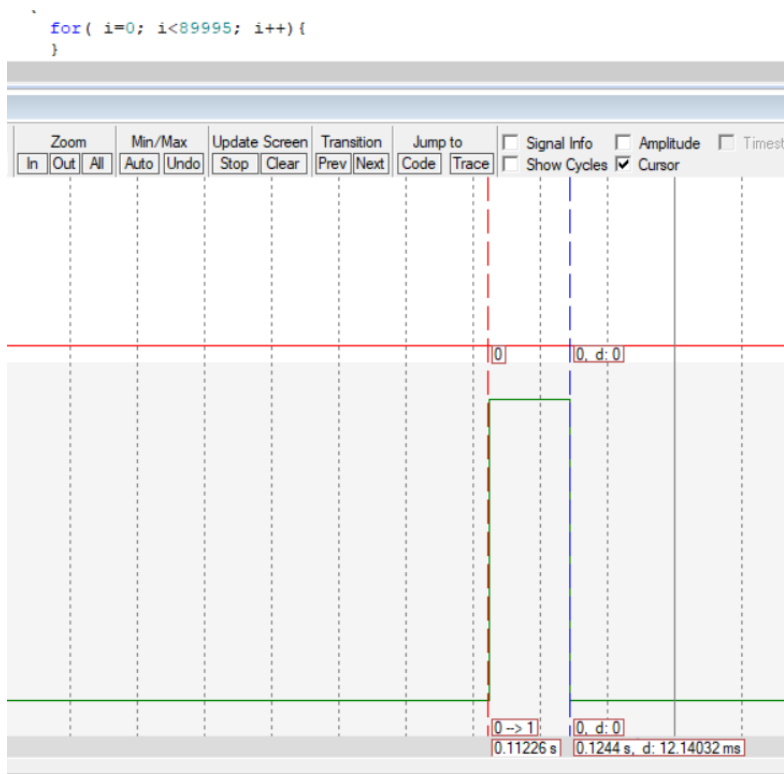
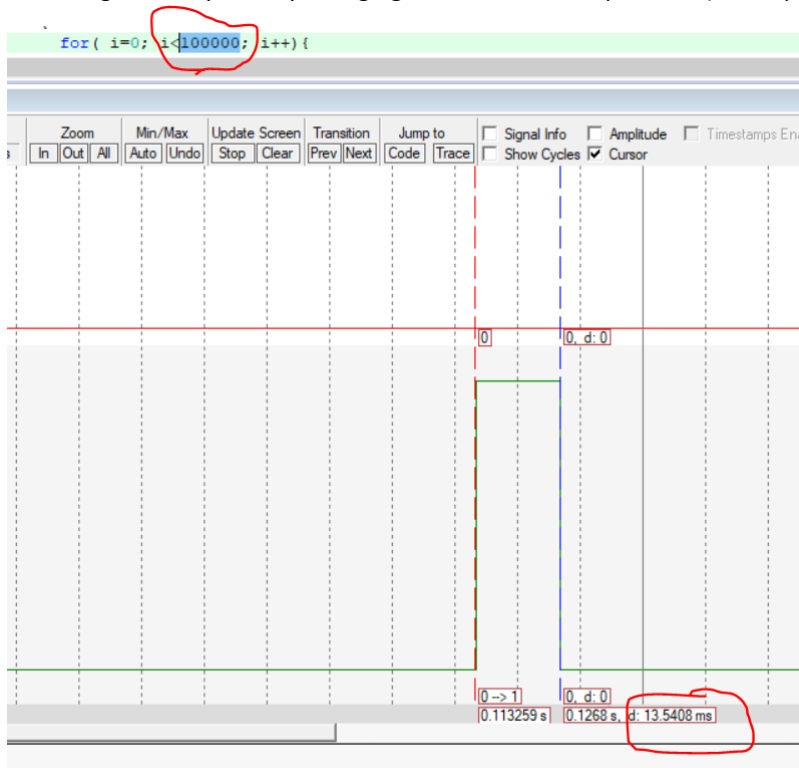


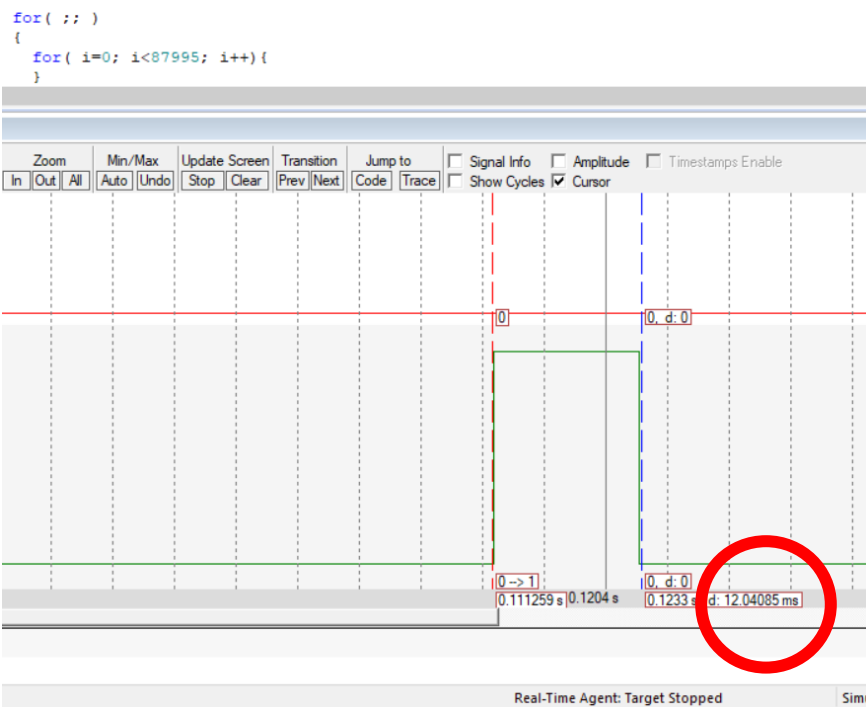
5ms occurs at a for loop that loops about 37000 times

## Task 6

Task 6 is required to be 12 ms

Trials to get exactly 5ms by changing the number of loops in the (for loop) and observing the logic analyzer





**5ms occurs at a for loop that loops about 87995 times**



# Calculations and implementation

## CPU Load

CPU Load Calculated by simso application

File View Help

\* Unsaved

Qt 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)	Followed by
1	TASK T1	Periodic	<input type="checkbox"/> No	0	50	-	50	0.0134	1
2	TASK T2	Periodic	<input type="checkbox"/> No	0	50	-	50	0.0136	1
3	TASK T3	Periodic	<input type="checkbox"/> No	0	100	-	100	0.01295	1
4	TASK T4	Periodic	<input type="checkbox"/> No	0	20	-	20	0.01367	1
5	TASK T5	Periodic	<input type="checkbox"/> No	0	10	-	10	5.019	1
6	TASK T6	Periodic	<input type="checkbox"/> No	0	100	-	100	12.04	1

Remove selected task(s)

Qt Results

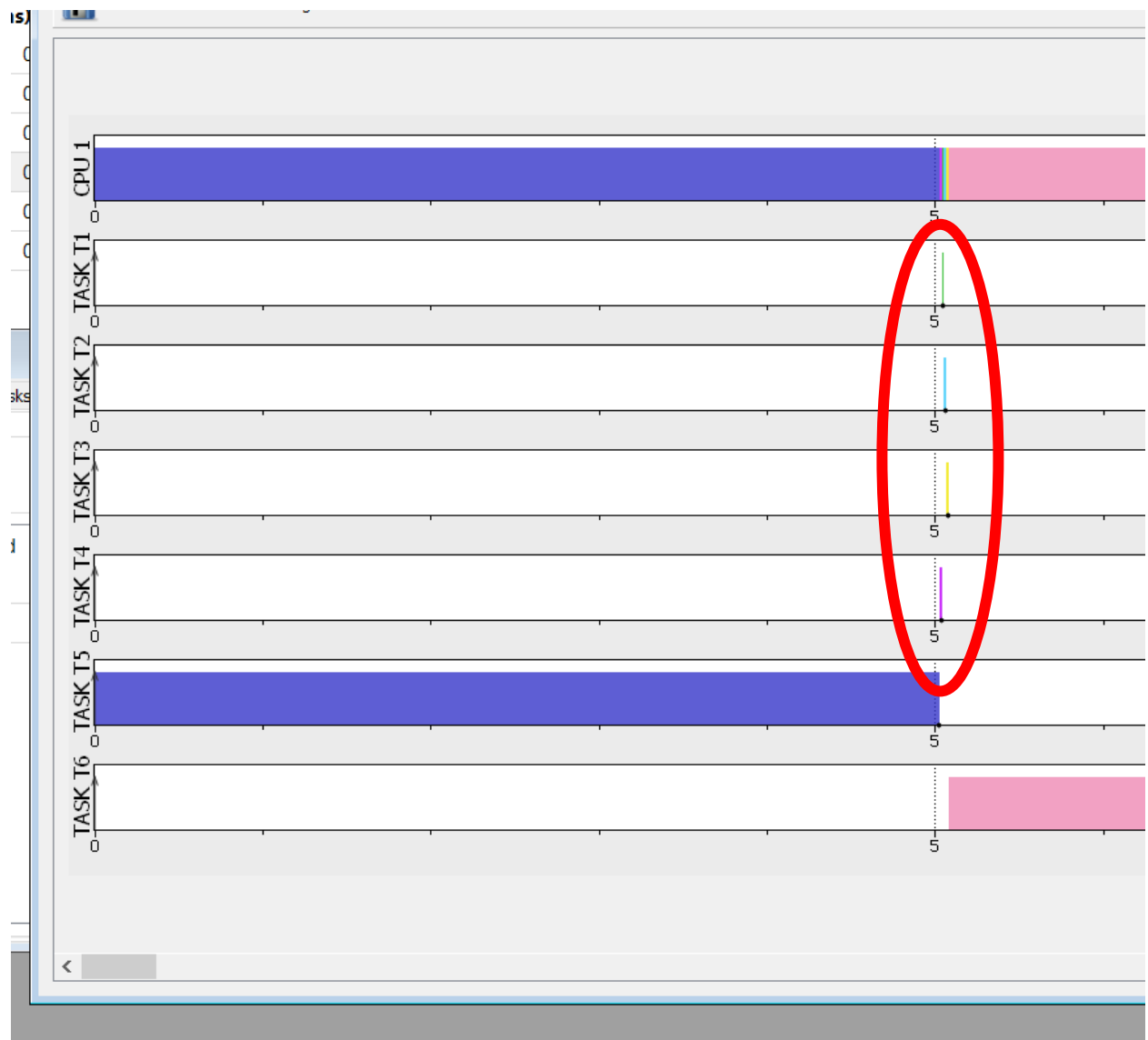
General Logs Tasks Scheduler Processors

Observation Window:  
from 0.00 to 100.00 ms [Configure...](#)

	Total load	Payload	System load
CPU 1	0.6237	0.6237	0.0000
Average	0.6237	0.6237	0.0000

Add task





We can see them after extremely zooming in

## CPU Load By Calculations

$U = (0.0134/50) + 0.0136/50 + (0.01295/100) + (0.01367/20) + (5.019/10) + (12.04/100) = \underline{\underline{0.6237}} = \underline{\underline{62.37\%}}$

$0.6237 < 1$

## CPU Load By FreeRTOS APIs

```
329 void task7( void * pvParameters ){    //CPU Load
330
331
332     int i=0;
333     TickType_t xLastWakeTime;
334     const TickType_t xFrequency = 20;
335     xLastWakeTime = xTaskGetTickCount();
336
337     vTaskSetApplicationTaskTag(NULL, (void *) 7 );
338
339     for( ;; )
340     {
341         vTaskGetRunTimeStats(temp_buffer);
342         vSerialPutString(NULL,temp_buffer, 190);
343         vSerialPutString(NULL,NEW_LINE, 4);
344
345         vTaskDelayUntil( &xLastWakeTime, xFrequency );
346     }
347 }
348
```

Task	Priority	CPU Load
task1	4	<1%
task2	2	<1%
task3	2	<1%
task6	1365	18%
task5	3231	43%
task4	5	<1%

```
365
366
367
368
369 xTaskCreatePeriodic(
370     task2,    /
371     "task2",
372     200,    /* S
```

Logic Analyzer

Signal	Min Time	Max Time	Grid	Zoom	Min/Max	Update Screen	Transition	Jump to
...000) >> 18	0 s	7.106267 s	0.5 s	In Out All	Auto Undo	Stop Clear	Prev Next	Code Trace

0 s 0 s 0.728 s, d: 0.728 s

## System Schedulability

### URM

Number of tasks = 6

$U = 0.6237$

$URM = 3 * (2^{1/3} - 1) = \mathbf{0.7798}$

**$U < URM$**

Then the system is schedulable

### Time Demand Analysis

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

W = Worst response time  
E = Execution time  
P = Periodicity  
T = Time instance

Task 5 has the highest priority as it has the lowest periodicity (10ms)

$W1(1) = 5$        $W1(2) = 5$       all the way to  $W1(10) = 5$

Task 4 has the second highest priority (20ms)

$W2(1) = 5 + 0.01367 = 5.01367$       all the way to  $W2(10) = 5.01367$

$W2(11) = 10 + 0.01367 = 10.01367$       to  $W2(20) = 10.01367$

Task 1

$W3(1) = 5.01367 + 0.0134 = 5.02707$       to  $W3(10) = 5.02707$

$W3(11) = 10.02707$       to  $W3(20) = 10.02707$

$W3(21) = 10.02707 + 5.01367 = 15.04074$       to  $W3(30) = 15.04074$

$W3(31) = 20.04074$       to  $W3(40) = 20.04074$

$W3(41) = 20.04074 + 5.01367 = 25.05441$       to  $W3(50) = 25.05441$

## Task 2

$W4(1) = 5.01367 + 0.0136 = 5.02727$	to $W4(10) = 5.02727$
$W4(11) = 10.02727$	to $W4(20) = 10.02727$
$W4(21) = 10.02727 + 5.01367 = 15.04094$	to $W4(30) = 15.04094$
$W4(31) = 20.04094$	to $W4(40) = 20.04094$
$W4(41) = 20.04094 + 5.01367 = 25.05461$	to $W4(50) = 25.05461$

## Task 3

$W5(1) = 5.02707$	to $W5(10) = 5.02707$
$W5(11) = 10.02707$	to $W5(20) = 10.02707$
$W5(21) = 10.02707 + 5.01367 = 15.04074$	to $W5(30) = 15.04074$
$W5(31) = 20.04074$	to $W5(40) = 20.04074$
$W5(41) = 20.04074 + 5.01367 = 25.05441$	to $W5(50) = 25.05441$
$W5(51) = 30.06808$	to $W5(60) = 30.06808$
$W5(61) = 35.08175$	to $W5(70) = 35.08175$
$W5(71) = 40.09542$	to $W5(80) = 40.09542$
$W5(81) = 45.10909$	to $W5(90) = 45.10909$
$W5(91) = 50.12276$	to $W5(100) = 50.12276$

## Task 6

$W6(1) = 17.1$	to $W6(10) = 17.1$
$W6(11) = 22.1$	to $W6(20) = 22.1$
$W6(21) = 27.1557$	to $W6(30) = 27.1557$
$W6(31) = 32.1558$	to $W6(40) = 32.1558$
$W6(41) = 42.243$	to $W6(50) = 42.243$
$W6(51) = 47.298$	to $W6(60) = 47.298$

W6(61) = 52.298

to W6(70) = 52.298

W6(71) = 52.298

to W6(80) = 52.298

W6(81) = 57.353

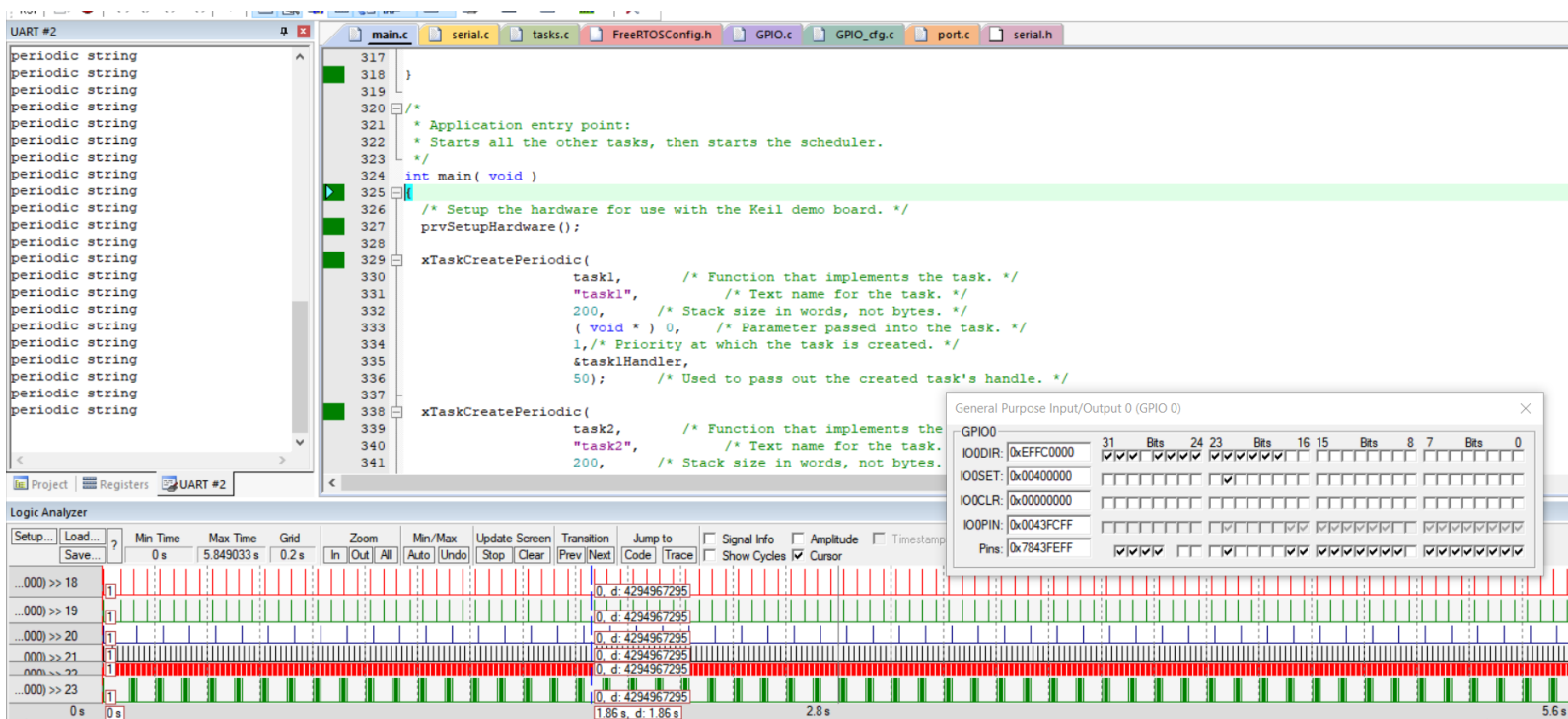
to W6(90) = 57.353

W6(91) = 62.353

to W6(100) = 62.353

They are all schedulable as there are no tasks missing their deadlines

## Overview of the system running



## Conclusions

- System is verified to be schedulable with the mentioned 6 tasks using:
  - CPU Load analysis
  - Time demand analysis
  - URM
- It could be loaded with even more tasks up to a certain point