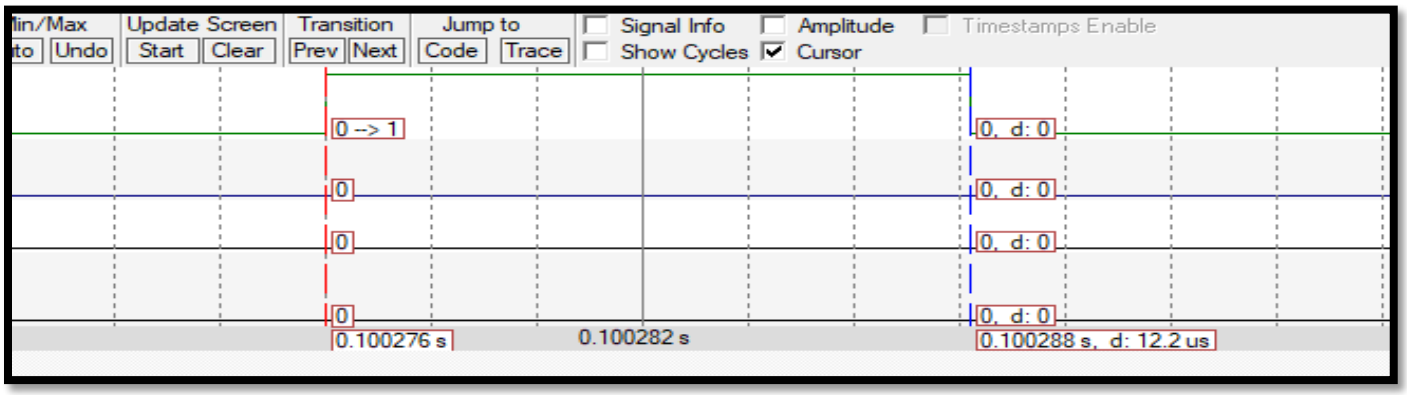
NAME: Ahmed Elaraby

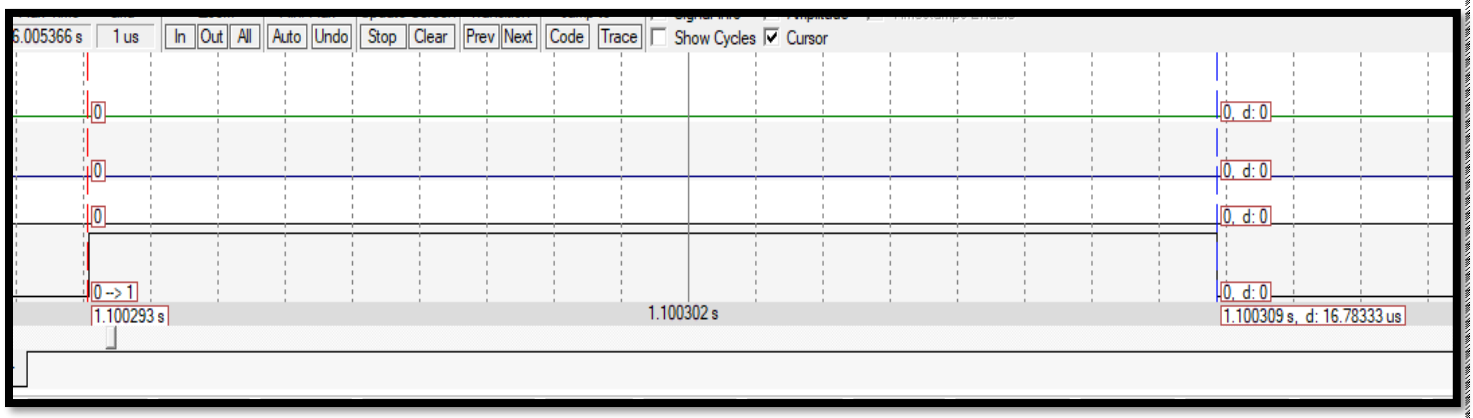
Email: es-ahmed.elaraby2023@alexu.edu.eg

Task 1&2



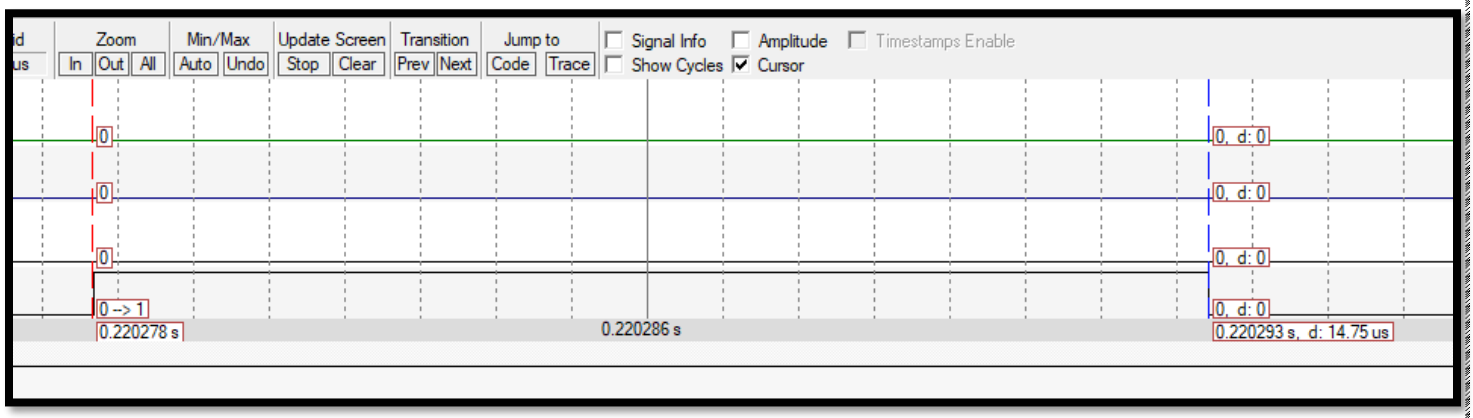
Execution Time = 12.2 us = 0.0122 ms

Task 3



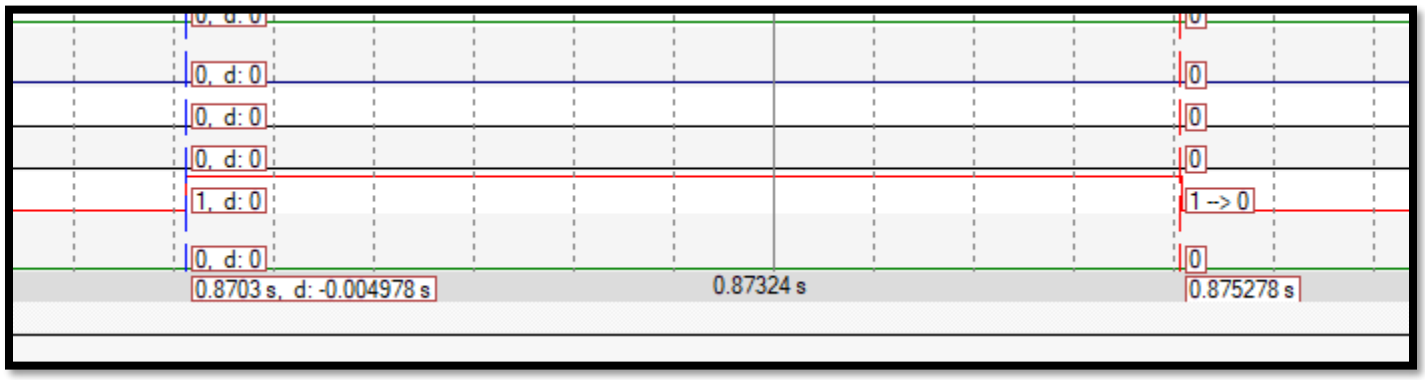
Execution Time = 16.78 us = 0.01678 ms

Task 4



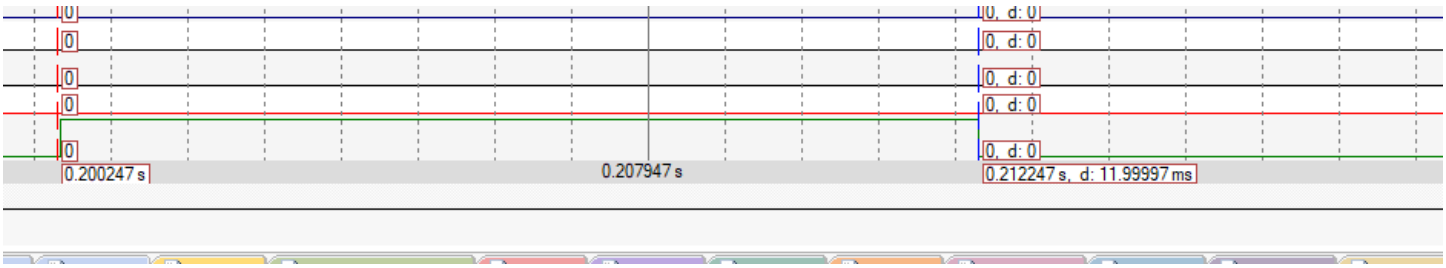
Execution Time = 14.75 us = 0.01475 ms

Task 5



Execution Time = 5ms

Task 6



Execution Time = 12.2 us = 12 ms

1-Hyperperiod

Hyperperiod (H) = LCM(Pi),
Where (Pi) is all task periodicities

***Hyperperiod** = 100 ms (after 100 ms system will repeat this cycle again)*

2-CPU load

- U = R/C

U** = (2*0.0122*2 + 0.01678 + 0.01475*5 + 5*10 + 12)/100 = **62.14%

*2*0.0122*2 (Execution time for **task 1 & task 2** at Hyperperiod)(every task executes **2** times)*

*0.01678 (Execution time for **task 3** at Hyperperiod) (this task executes only **1** time at Hyperperiod)*

*0.01475*5 (Execution time for **task 4** at Hyperperiod) (this task executes **5** times at Hyperperiod)*

*5*10 (Execution time for **task 5** at Hyperperiod) (this task executes only **5** time at Hyperperiod)*

*12 (Execution time for **task 6** at Hyperperiod) (this task executes only **1** time at Hyperperiod)*

3-URM

$$n(2^{\frac{1}{n}} - 1)$$

URM** = 6 (2^(1/6) -1) = **0.73478

URM > CPU Load ➔ Guaranteed Schedulable

4-time demand analysis

Task 1 (tasks 4 & 5 have higher priority) *W(0) = 0.0178*(1/20) + (1/10)*5 +0.0122 = 5.03* *W(1) = 0.0178*(2/20) + (2/10)*5 +0.0122 = 5.03* *W(2) = 0.0178*(3/20) + (3/10)*5 +0.0122 = 5.03* *W(3) = 0.0178*(4/20) + (4/10)*5 +0.0122 = 5.03* *W(4) = 0.0178*(5/20) + (5/10)*5 +0.0122 = 5.03* *W(5) = 0.0178*(5/20) + (5/10)*5 +0.0122 = 5.03* *W(50) = 0.0178*(50/20) + (50/10)*5 +0.0122 = 25.0656< 50 (DONE)*	*Task 2 (tasks 4 & 5 & 1 have higher priority)* *W(0) = 0.0178*(1/20) + (1/10)*5 +0.0122*2 = 5.042* *W(1) = 0.0178*(2/20) + (2/10)*5 +0.0122*2 = 5.042* *W(2) = 0.0178*(3/20) + (3/10)*5 +0.0122*2 = 5.042* *W(3) = 0.0178*(4/20) + (4/10)*5 +0.0122*2 = 5.042* *W(4) = 0.0178*(5/20) + (5/10)*5 +0.0122*2 = 5.042* *W(5) = 0.0178*(5/20) + (5/10)*5 +0.0122*2 = 5.042* *W(50) = 0.0178*(50/20) + (50/10)*5 +0.0122*2 = 25.0778 < 50 (DONE)*
Task 3 (tasks 4 & 5 & 1 & 2 have higher priority) *W(0) = 0.0178*(1/20) + (1/10)*5 +0.0122*2*(1/50)+0.01678 = 5.059* *W(1) = 0.0178*(2/20) + (2/10)*5 +0.0122*2*(2/50)+0.01678 = 5.059* *W(2) = 0.0178*(3/20) + (3/10)*5 +0.0122*2*(3/50)+0.01678 = 5.059* *W(3) = 0.0178*(4/20) + (4/10)*5 +0.0122*2*(4/50)+0.01678 = 5.059* *W(4) = 0.0178*(5/20) + (5/10)*5 +0.0122*2*(5/50)+0.01678 = 5.059* *W(5) = 0.0178*(6/20) + (6/10)*5 +0.0122*2*(6/50)+0.01678 = 5.059* *W(100) = 0.0178*(100/20) + (100/10)*5 +0.0122*2*(100/50)+0.01678 = 50.1545<100(DONE)*	*Task 4 (only task 5 has higher priority)* *W(0) = 0.0178 + (1/10)*5 = 5.0178* *W(1) = 0.0178 + (2/10)*5 = 5.0178* *W(2) = 0.0178 + (3/10)*5 = 5.0178* *W(3) = 0.0178 + (4/10)*5 = 5.0178* *W(4) = 0.0178 + (5/10)*5 = 5.0178* *W(5) = 0.0178 + (6/10)*5 = 5.0178* *W(20) = 0.0178 + (20/10)*5 = 10.0178<20 (DONE)*
Task 5 *W(0) = 5 + 0 = 5* *W(1) = 5 +0 = 5* *W(2) = 5 + 0 = 5* *W(3) = 5 +0 = 5* *W(4) = 5 + 0 = 5(DONE)*	*Task 3 (tasks 4 & 5 & 1 & 2 & 3 have higher priority)* *W(0) = 0.0178*(1/20) + (1/10)*5 +0.0122*2*(1/50)+0.01678 +12= 17.059* *W(1) = 0.0178*(2/20) + (2/10)*5 +0.0122*2*(2/50)+0.01678 +12= 17.059* *W(2) = 0.0178*(3/20) + (3/10)*5 +0.0122*2*(3/50)+0.01678 +12= 17.059* *W(3) = 0.0178*(4/20) + (4/10)*5 +0.0122*2*(4/50)+0.01678 +12= 17.059* *W(4) = 0.0178*(5/20) + (5/10)*5 +0.0122*2*(5/50)+0.01678 +12= 17.059* *W(5) = 0.0178*(6/20) + (6/10)*5 +0.0122*2*(6/50)+0.01678 +12 = 17.059* *W(100) = 0.0178*(100/20) + (100/10)*5 +0.0122*2*(100/50)+0.01678 +12 = 62.3<100(DONE)*

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

Fixed priority rate monotonic scheduler

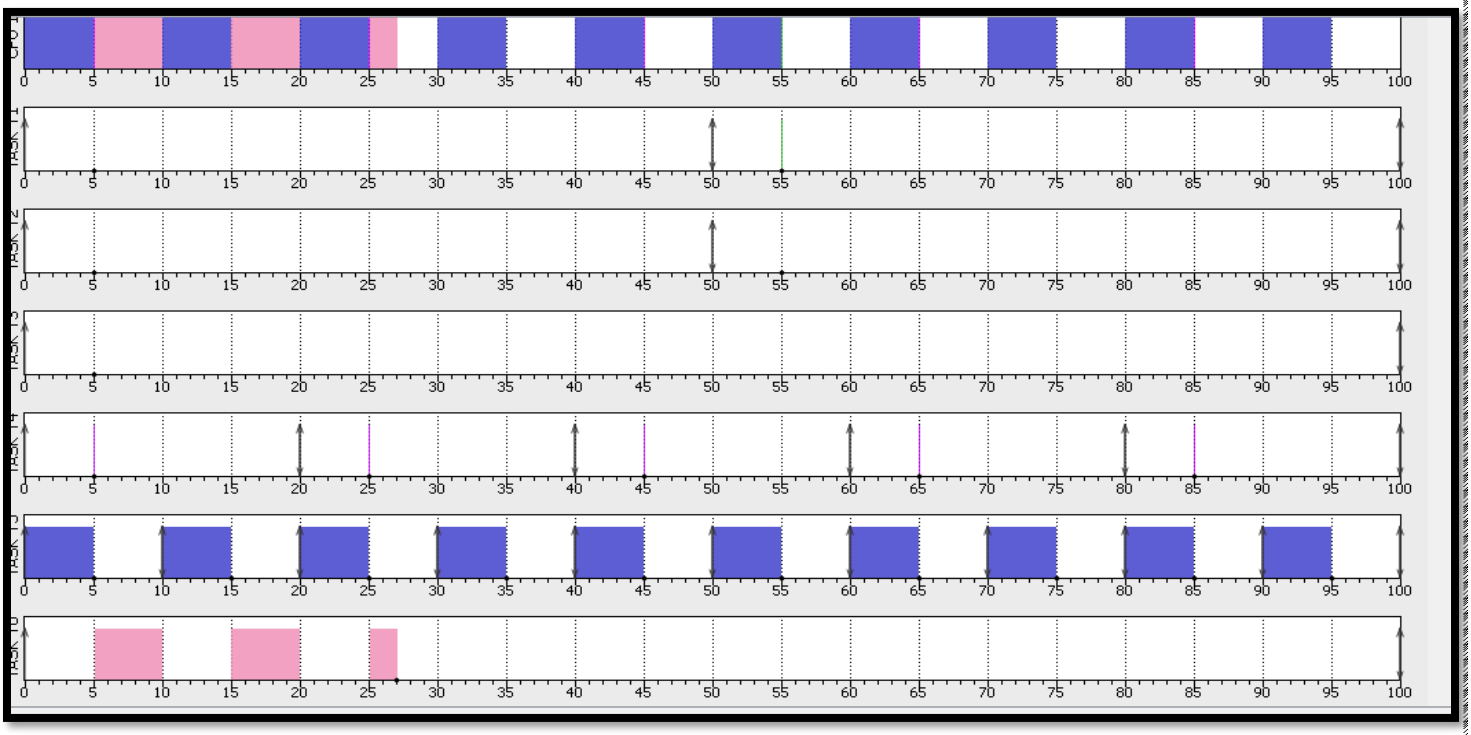
General | Scheduler | Processors | Tasks

id	Name	Task type	Abort on miss	Act. Date (ms)	Period (ms)	List of Act. dates (ms)	Deadline (ms)	WCET (ms)
1	TASK T1	Periodic	Yes	0	50	-	50	0.0122
2	TASK T2	Periodic	Yes	0	50	-	50	0.0122
3	TASK T3	Periodic	Yes	0	100	-	100	0.01678
4	TASK T4	Periodic	Yes	0	20	-	20	0.01475
5	TASK T5	Periodic	Yes	0	10	-	10	5
6	TASK T6	Periodic	Yes	0	100	-	100	12

Observation Window:

from 0.00 to 100.00 ms | Configure...

	Total load	Payload	System load
CPU 1	0.6214	0.6214	0.0000
Average	0.6214	0.6214	0.0000



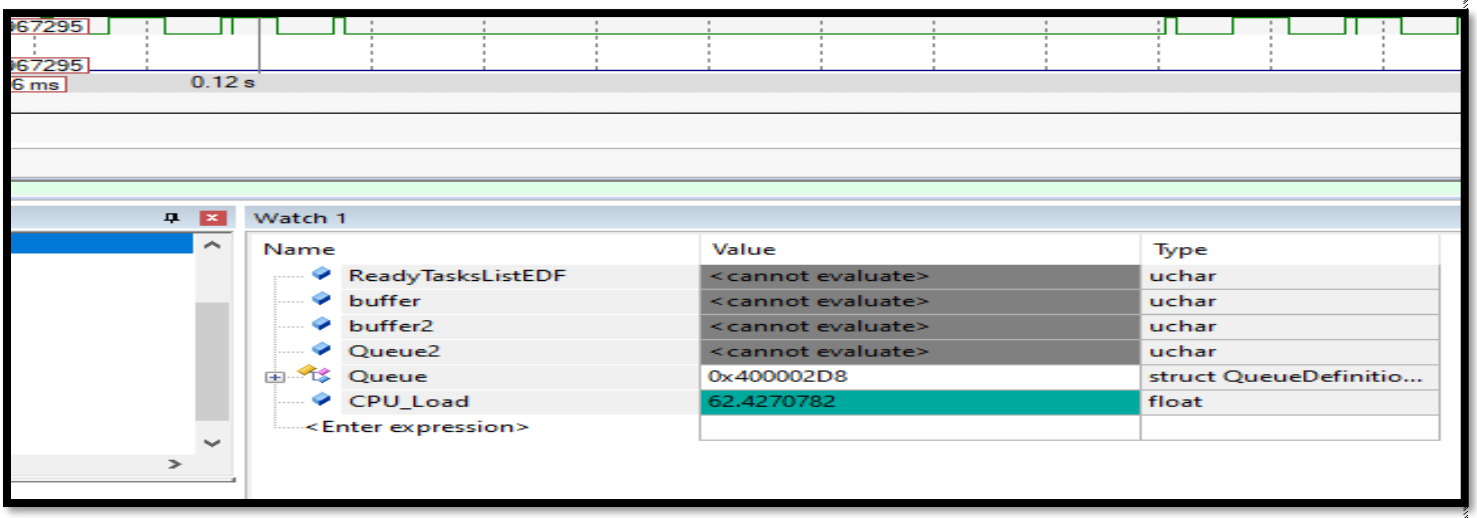
"3. Using Keil simulator in run-time and the given set of tasks:"

- Calculate the CPU usage time using timer 1 and trace macros

CODE

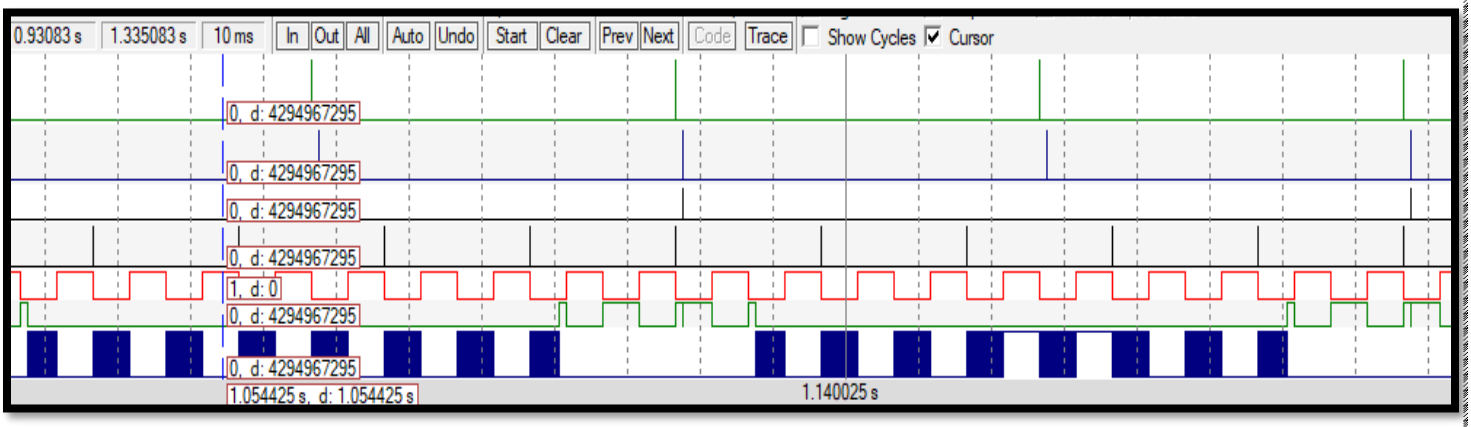
```
#define traceTASK_SWITCHED_OUT() do\
                                  { if((int)pxCurrentTCB->pxTaskTag == 1)\
                                  {GPIO_write(PORT_0,PIN1,PIN_IS_LOW); \
                                task_1_out = T1TC ; task_1_total +=(task_1_out - task_1_in);   }\
                                 else if((int)pxCurrentTCB->pxTaskTag == 2)\
                                  {GPIO_write(PORT_0,PIN2,PIN_IS_LOW);\
                                  task_2_out = T1TC ; task_2_total +=(task_2_out - task_2_in); }\
                                 else if((int)pxCurrentTCB->pxTaskTag == 3)\
                                  {GPIO_write(PORT_0,PIN3,PIN_IS_LOW);\
                                 task_3_out = T1TC ; task_3_total +=(task_3_out - task_3_in);   }\
                                  else if((int)pxCurrentTCB->pxTaskTag == 4)\
                                  {GPIO_write(PORT_0,PIN4,PIN_IS_LOW);\
                                  task_4_out = T1TC ; task_4_total +=(task_4_out - task_4_in); }\
                                 else if((int)pxCurrentTCB->pxTaskTag == 5)\
                                  {GPIO_write(PORT_0,PIN5,PIN_IS_LOW);\
                                task_5_out = T1TC ; task_5_total +=(task_5_out - task_5_in);     }\
                                  else if((int)pxCurrentTCB->pxTaskTag == 6)\
                                  {GPIO_write(PORT_0,PIN6,PIN_IS_LOW);\
                                  task_6_out = T1TC ; task_6_total +=(task_6_out - task_6_in); }\
                                 system_time = T1TC;\
      CPU_Load = (((task_1_total)+(task_2_total)+(task_3_total)+(task_4_total)+(task_5_total)+(task_6_total))/(float)system_time )*100;\
                                }while(0)
#define traceTASK_SWITCHED_IN() do\
                                  { if((int)pxCurrentTCB->pxTaskTag == 1)\
                                  { task_1_in = T1TC;                    \
                                    GPIO_write(PORT_0,PIN1,PIN_IS_HIGH);}\
```

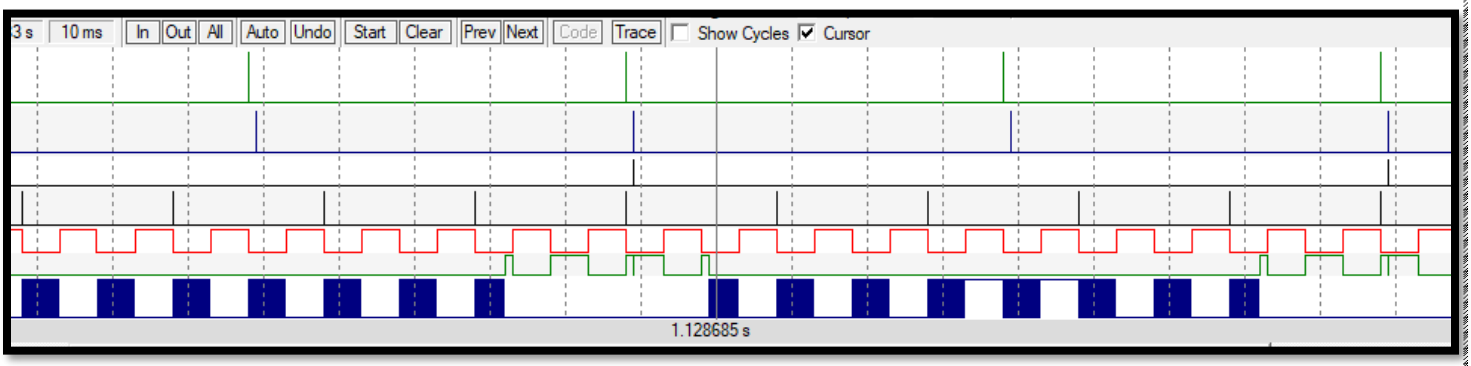
Calculation



Like simso result

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





As expected, when task 6 available (largest execution time) the idle task didn't work until task 6 end .