Egypt FWD 2022
Advanced Embedded
Track
RTOS Project

EDF SCHDUELER IN RTOS

2022

[You can add an abstract or other key statement here. An abstract is typically a short summary of the document content.]

TABLE OF CONTENTS

Contents

SYSTEM VALIDATION

Tasks Exe. Time

USING TRACE MACROS AND DIGITAL ANALYZER

Load1 Sim. Task exe = 5ms.

Load2 Sim. Task exe = 12 ms.

Button1 Monitor Task = 16 us

Button2 Monitor Task = 16 us

Transmitter Task = 23 us.

UART Task = 18 us.

SYSTEM HYPERPERIOD

LCM[All Tasks Periods]=LCM[10,100,50,50,100,20]=100 ms.

CPU LOAD

CPU Load= $\frac{\sum \text{Task Exe. Time}}{\text{Hyper Period}} = \frac{5*10+12*1+0.016*2+0.016*2+0.023+0.018*5}{100} = 62.18\%$

CHECK SYSTEM SCHEDULABILITY USING URM

$U = \sum T_i / P_i = 5/10+12/100+0.016/50+0.016/50+0.023/100+0.018/20 = 0.6218 < 1.$

$U_{RM} = 6*(2^{1/6}-1)=1.559.$

$U < U_{RM}$ ➔ System is schedulable using RM.

CHECK SYSTEM SCHEDULABILITY USING TIME-DEMAND ANALYSIS

Check System Shedulability using Time Analysis

Implementation :I have created Function to calculate Time analysis for tasks and send it to uart .

```
void CalcTimeDemand(void)
{
  int i,j,k,l;
  float TaskPeriodList[6]={10,20,50,50,100,100};
  float TaskExeTimeList[6]={5,0.018,0.017,0.017,12,0.023};
  char  TaskName[6][10]={"LD1","UART","B1","B2","LD2","TRANS."};
  float w;
  char RString[40];

  for(j=0;j<6;j++)//calculate Time Analysis for each Task
  {
      sprintf(RString,"Time Analysis %s Task \n",TaskName[j]); //Print Task Name
      vSerialPutString((const signed char*)RString,strlen((const char*)RString));
      for(l=0;l<40000;l++) //Delay to prevent serial buffer overflow
      {
    w=0;//init worst case exe to 0
    for(i=0;i<(TaskPeriodList[j]);i++) //calculate W for every Time instance for Task j from 1->Task Period
    {
      w=TaskExeTimeList[j];
      for(k=0;k<=(j-1);k++)
      {
        w+=(ceil((i+1)/TaskPeriodList[k])*TaskExeTimeList[k]);
      }
      sprintf(RString,"w[%d] = %5.5f \n",i,w);//Print Wi
      vSerialPutString((const signed char*)RString,strlen((const char*)RString));
      for(l=0;l<40000;l++)//Delay to prevent serial buffer overflow
      {
    }
    if(w<TaskPeriodList[j]) //check Task Shedulablity if Wi < Pi then task schedulable
    {
      sprintf(RString,"w[%d] < Task Period, Task is Schedulable\n",i);
      vSerialPutString((const signed char*)RString,strlen((const char*)RString));
    }
    else  //Wi>Pi Task is not schedulable
    {
      sprintf(RString,"w[%d] > Task Period, Task is not Schedulable\n",i);
      vSerialPutString((const signed char*)RString,strlen((const char*)RString));
```

Analysis Result

Time Analysis LD1 Task

w[0] = 5.00000

w[1] = 5.00000

w[2] = 5.00000

w[3] = 5.00000

w[4] = 5.00000

w[5] = 5.00000

w[6] = 5.00000

w[7] = 5.00000

w[8] = 5.00000

w[9] = 5.00000

w[10] < Task Period, Task is Schedulable

Time Analysis UART Task

w[0] = 5.01800

w[1] = 5.01800

w[2] = 5.01800

w[3] = 5.01800

w[4] = 5.01800

w[5] = 5.01800

w[6] = 5.01800

w[7] = 5.01800

w[8] = 5.01800

w[9] = 5.01800

w[10] = 10.01800

w[11] = 10.01800

w[12] = 10.01800

w[13] = 10.01800

w[14] = 10.01800

w[15] = 10.01800

w[16] = 10.01800

w[17] = 10.01800

w[18] = 10.01800

w[19] = 10.01800

w[20] < Task Period, Task is Schedulable

Time Analysis B1 Task

w[0] = 5.03500

w[1] = 5.03500

w[2] = 5.03500

w[3] = 5.03500

w[4] = 5.03500

w[5] = 5.03500

w[6] = 5.03500

w[7] = 5.03500

w[8] = 5.03500

w[9] = 5.03500

w[10] = 10.03500

w[11] = 10.03500

w[12] = 10.03500

w[13] = 10.03500

w[14] = 10.03500

w[15] = 10.03500

w[16] = 10.03500

w[17] = 10.03500

w[18] = 10.03500

w[19] = 10.03500

w[20] = 15.05300

w[21] = 15.05300

w[22] = 15.05300

w[23] = 15.05300

w[24] = 15.05300

w[25] = 15.05300

w[26] = 15.05300

;2w[27] = 15.05300

w[28] = 15.05300

w[29] = 15.05300

w[30] = 20.05300

w[31] = 20.05300

w[32] = 20.05300

w[33] = 20.05300

w[34] = 20.05300

w[35] = 20.05300

w[36] = 20.05300

w[37] = 20.05300

w[38] = 20.05300

w[39] = 20.05300

w[40] = 25.07100

w[41] = 25.07100

w[42] = 25.07100

w[43] = 25.07100

w[44] = 25.07100

w[45] = 25.07100

w[46] = 25.07100

w[47] = 25.07100

w[48] = 25.07100

w[49] = 25.07100

w[50] < Task Period, Task is Schedulable

Time Analysis B2 Task

w[0] = 5.05200

w[1] = 5.05200

w[2] = 5.05200

w[3] = 5.05200

w[4] = 5.05200

w[5] = 5.05200

w[6] = 5.05200

w[7] = 5.05200

w[8] = 5.05200

w[9] = 5.05200

w[10] = 10.05200

w[11] = 10.05200

w[12] = 10.05200

w[13] = 10.05200

w[14] = 10.05200

w[15] = 10.05200

w[16] = 10.05200

w[17] = 10.05200

w[18] = 10.05200

w[19] = 10.05200

w[20] = 15.07000

w[21] = 15.07000

w[22] = 15.07000

w[23] = 15.07000

w[24] = 15.07000

w[25] = 15.07000

w[26] = 15.07000

w[27] = 15.07000

w[28] = 15.07000

w[29] = 15.07000

w[30] = 20.07000

w[31] = 20.07000

w[32] = 20.07000

w[33] = 20.07000

w[34] = 20.07000

w[35] = 20.07000

w[36] = 20.07000

w[37] = 20.07000

w[38] = 20.07000

w[39] = 20.07000

w[40] = 25.08800

w[41] = 25.08800

w[42] = 25.08800

w[43] = 25.08800

w[44] = 25.08800

w[45] = 25.08800

w[46] = 25.08800

w[47] = 25.08800

w[48] = 25.08800

w[49] = 25.08800

w[50] < Task Period, Task is Schedulable

Time Analysis LD2 Task

w[0] = 17.05200

w[1] = 17.05200

w[2] = 17.05200

w[3] = 17.05200

w[4] = 17.05200

w[5] = 17.05200

w[6] = 17.05200

w[7] = 17.05200

w[8] = 17.05200

w[9] = 17.05200

w[10] = 22.05200

w[11] = 22.05200

w[12] = 22.05200

w[13] = 22.05200

w[14] = 22.05200

w[15] = 22.05200

w[16] = 22.05200

w[17] = 22.05200

w[18] = 22.05200

w[19] = 22.05200

w[20] = 27.07000

w[21] = 27.07000

w[22] = 27.07000

w[23] = 27.07000

w[24] = 27.07000

w[25] = 27.07000

w[26] = 27.07000

w[27] = 27.07000

w[28] = 27.07000

w[29] = 27.07000

w[30] = 32.07000

w[31] = 32.07000

w[32] = 32.07000

w[33] = 32.07000

w[34] = 32.07000

w[35] = 32.07000

w[36] = 32.07000

w[37] = 32.07000

w[38] = 32.07000

w[39] = 32.07000

w[40] = 37.08800

w[41] = 37.08800

w[42] = 37.08800

w[43] = 37.08800

w[44] = 37.08800

w[45] = 37.08800

w[46] = 37.08800

w[47] = 37.08800

w[48] = 37.08800

w[49] = 37.08800

w[50] = 42.12200

w[51] = 42.12200

w[52] = 42.12200

w[53] = 42.12200

w[54] = 42.12200

w[55] = 42.12200

w[56] = 42.12200

w[57] = 42.12200

w[58] = 42.12200

w[59] = 42.12200

w[60] = 47.14000

w[61] = 47.14000

w[62] = 47.14000

w[63] = 47.14000

w[64] = 47.14000

w[65] = 47.14000

w[66] = 47.14000

w[67] = 47.14000

w[68] = 47.14000

w[69] = 47.14000

w[70] = 52.14000

w[71] = 52.14000

w[72] = 52.14000

w[73] = 52.14000

w[74] = 52.14000

w[75] = 52.14000

w[76] = 52.14000

w[77] = 52.14000

w[78] = 52.14000

w[79] = 52.14000

w[80] = 57.15800

w[81] = 57.15800

w[82] = 57.15800

w[83] = 57.15800

w[84] = 57.15800

w[85] = 57.15800

w[86] = 57.15800

w[87] = 57.15800

w[88] = 57.15800

w[89] = 57.15800

w[90] = 62.15800

w[91] = 62.15800

w[92] = 62.15800

w[93] = 62.15800

w[94] = 62.15800

w[95] = 62.15800

w[96] = 62.15800

w[97] = 62.15800

w[98] = 62.15800

w[99] = 62.15800

w[100] < Task Period, Task is Schedulable

Time Analysis TRANS. Task

w[0] = 17.07500

w[1] = 17.07500

w[2] = 17.07500

w[3] = 17.07500

w[4] = 17.07500

w[5] = 17.07500

w[6] = 17.07500

w[7] = 17.07500

w[8] = 17.07500

w[9] = 17.07500

w[10] = 22.07500

w[11] = 22.07500

w[12] = 22.07500

w[13] = 22.07500

w[14] = 22.07500

w[15] = 22.07500

w[16] = 22.07500

w[17] = 22.07500

w[18] = 22.07500

w[19] = 22.07500

w[20] = 27.09300

w[21] = 27.09300

w[22] = 27.09300

w[23] = 27.09300

w[24] = 27.09300

w[25] = 27.09300

w[26] = 27.09300

w[27] = 27.09300

w[28] = 27.09300

w[29] = 27.09300

w[30] = 32.09300

w[31] = 32.09300

w[32] = 32.09300

w[33] = 32.09300

w[34] = 32.09300

w[35] = 32.09300

w[36] = 32.09300

w[37] = 32.09300

w[38] = 32.09300

w[39] = 32.09300

w[40] = 37.11100

w[41] = 37.11100

w[42] = 37.11100

w[43] = 37.11100

w[44] = 37.11100

w[45] = 37.11100

w[46] = 37.11100

w[47] = 37.11100

w[48] = 37.11100

w[49] = 37.11100

w[50] = 42.14500

w[51] = 42.14500

w[52] = 42.14500

w[53] = 42.14500

w[54] = 42.14500

w[55] = 42.14500

w[56] = 42.14500

w[57] = 42.14500

w[58] = 42.14500

w[59] = 42.14500

w[60] = 47.16300

w[61] = 47.16300

w[62] = 47.16300

w[63] = 47.16300

w[64] = 47.16300

w[65] = 47.16300

w[66] = 47.16300

w[67] = 47.16300

w[68] = 47.16300

w[69] = 47.16300

w[70] = 52.16300

w[71] = 52.16300

w[72] = 52.16300

w[73] = 52.16300

w[74] = 52.16300

w[75] = 52.16300

w[76] = 52.16300

w[77] = 52.16300

w[78] = 52.16300

w[79] = 52.16300

w[80] = 57.18100

w[81] = 57.18100

w[82] = 57.18100

w[83] = 57.18100

w[84] = 57.18100

w[85] = 57.18100

w[86] = 57.18100

w[87] = 57.18100

w[88] = 57.18100

w[89] = 57.18100

w[90] = 62.18100

w[91] = 62.18100

w[92] = 62.18100

w[93] = 62.18100

w[94] = 62.18100

w[95] = 62.18100

w[96] = 62.18100

w[97] = 62.18100

w[98] = 62.18100

w[99] = 62.18100

w[100] < Task Period, Task is Schedulable

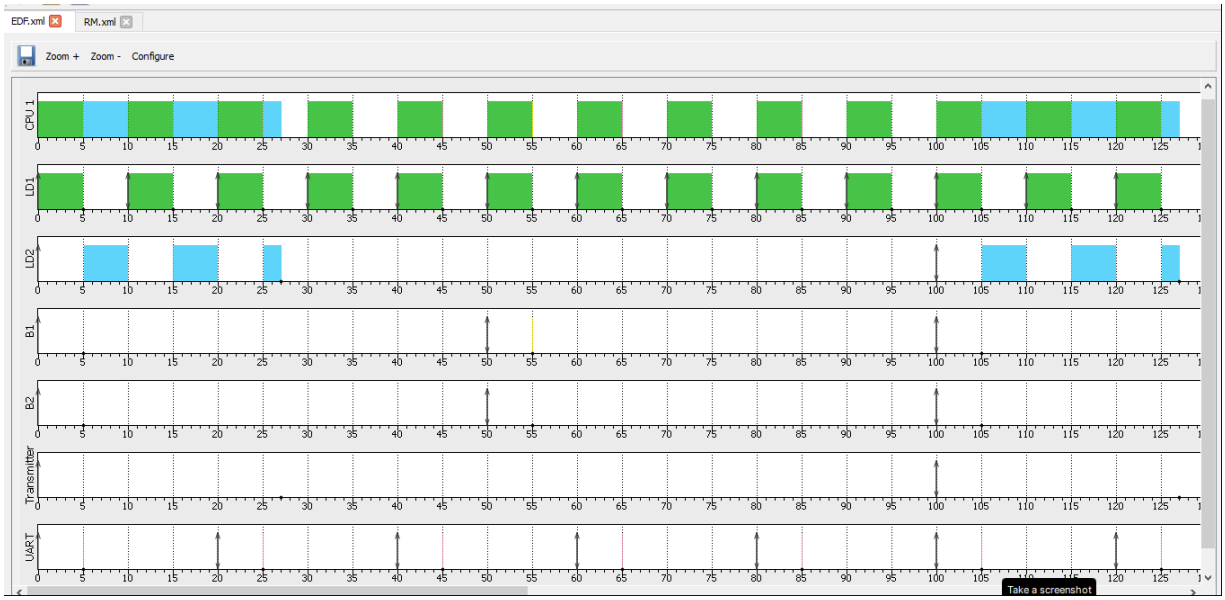
Summary:

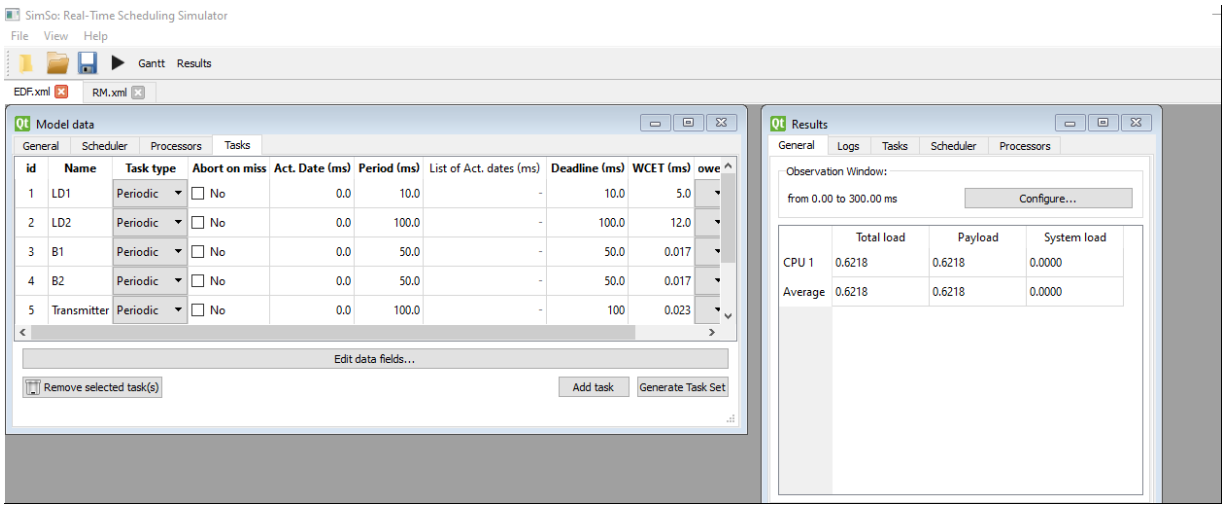
Time Analysis result for Given Task set , System is scheduable.

System validation using Simso offline simulator

USING EDF SCHEDULER

- Results

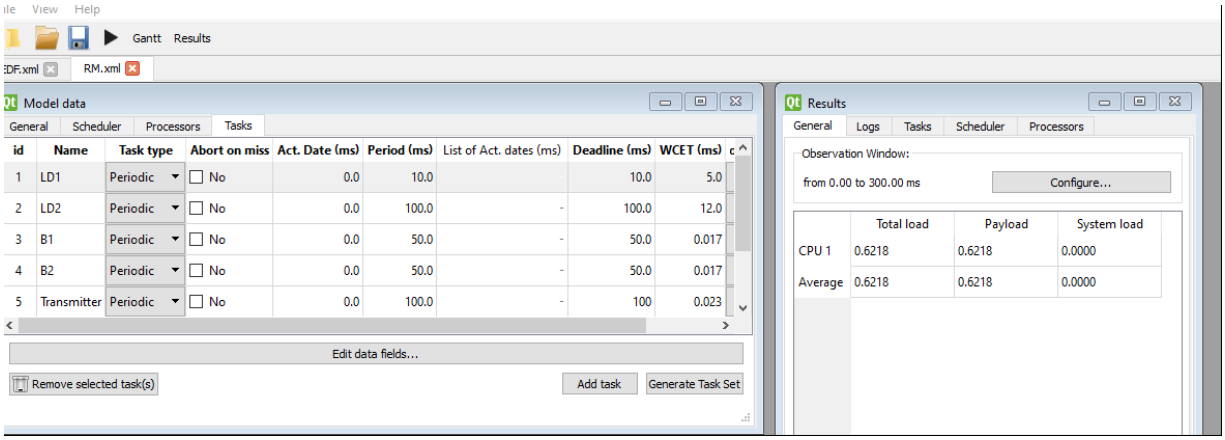




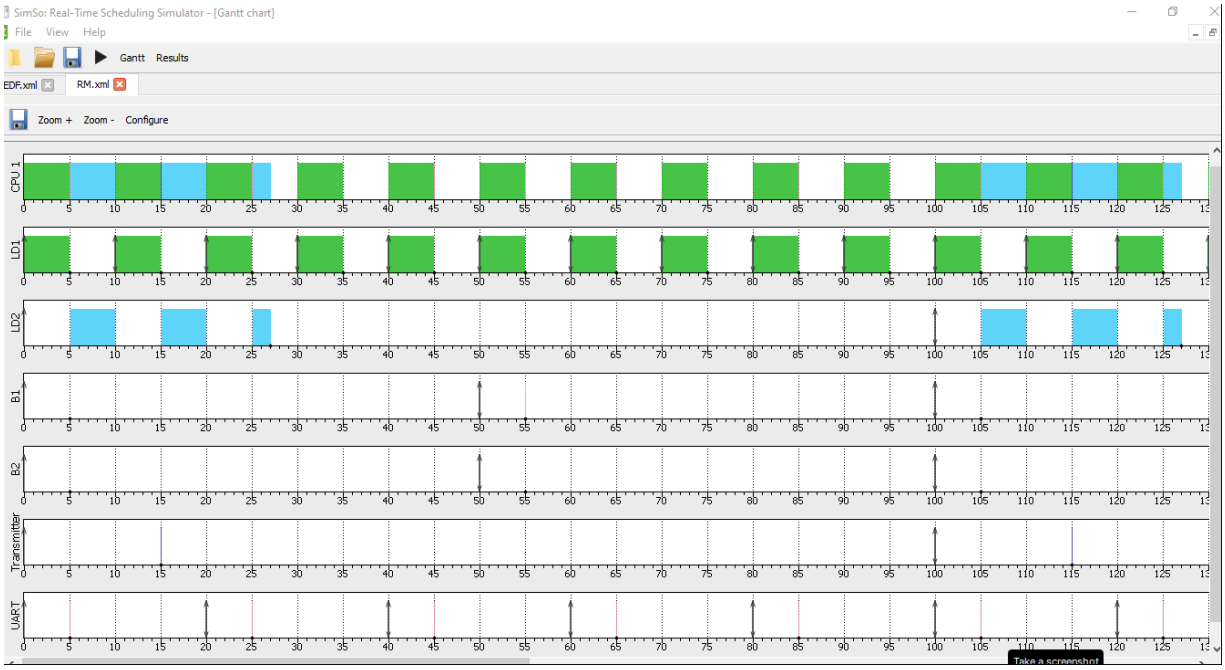
- Gantt chart

USING RM SCHEDULER

- Results

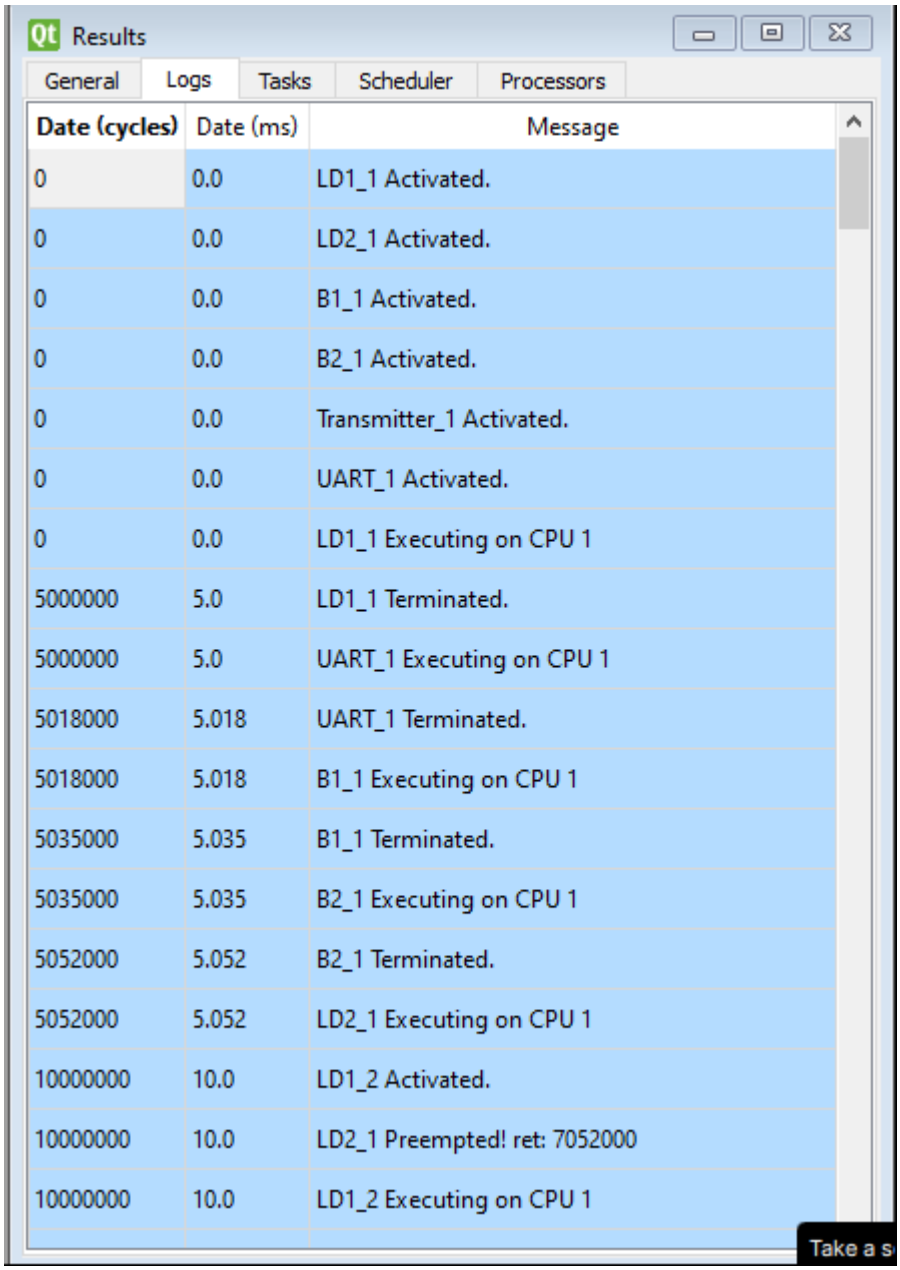


- Gantt chart



SYSTEM VALIDATION

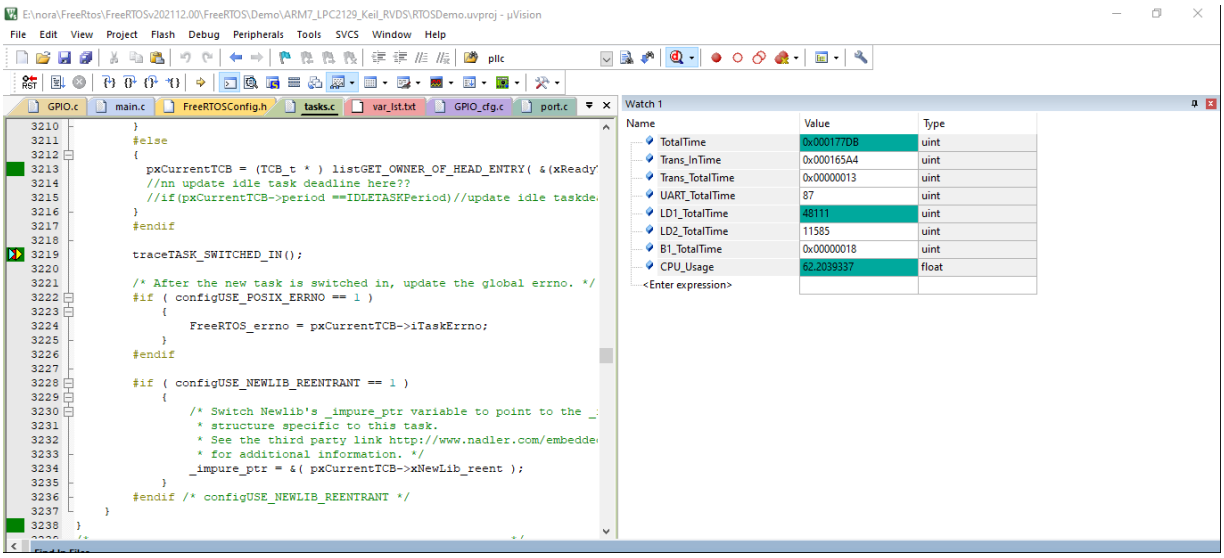
- Log



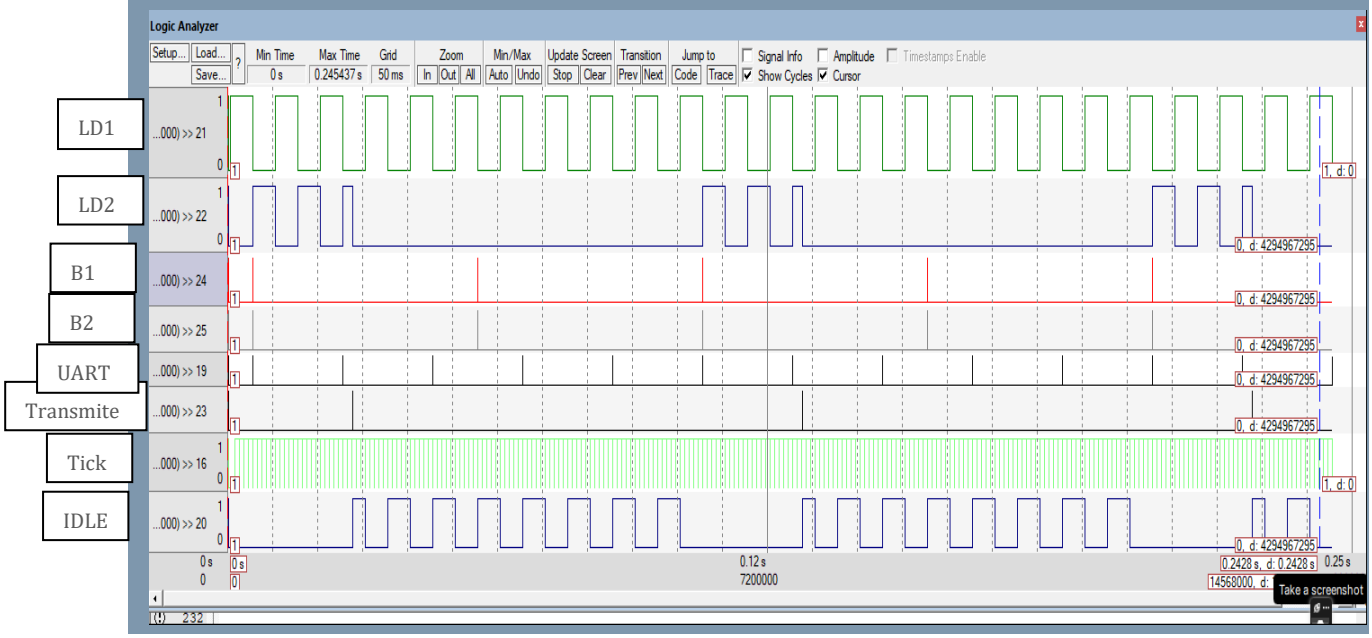
CALCULATION OF CPU LOAD USING KEIL SIM. (TIMER1,TRACE MACROS)

CPU_Usage=((LD1_TotalTime+LD2_TotalTime+B1_TotalTime+B2_TotalTime+Trans_TotalTime+UART_TotalTime)/(float)TotalTime)*100 =62.44%

SYSTEM VALIDATION



TOTAL SYSTEM TRACE USING KEIL SIM(TIMER1,GPIO,TRACE MACROS)



SYSTEM VALIDATION

Notes

IMMPLEMENTATION

Comparing Task Trace to Simso output for EDF I found results have identical task activation sequence .

Only when two tasks with same deadline become active together result may differ since any task can be chosen Randomly(as stated in EDF assumptions)

In the given Task set case: LD2 Task , Transmitter Task since they both same Deadline.

RESULTS

System with given Task set was proved to be schedulable by both scheduling techniques: RM, EDF.

EXTRA SYSTEM MODIFCATIONS NOTE

Extra changes were needed to complete EDF implementation:

- Problem: (In “prvAddNewTaskToReadyList ()”)

  After Task creation a check for created task priority against current task priority if created task priority is equal or higher than current task make created task current task.

  Since all tasks were created with same priority, the above condition resulted on last created task to run first though it is not the one having least deadline.

  Solution:

  Check if using EDF(i.e. configUSE_EDF_SCHEDULER==1) then get head of EDFReadyList instead of checking priorty.

- In “xTaskIncrementTick” function , gaureded with EDF macro whenever a task gets unlbocked Request CoTextSwitch.
- Made IdleTask Deadline with Macro define ” IDLE_TASK_DL” and chose value 150 (to be larger than farst task deadline in system).
- Updated Idle Task deadline every tick ,in “ xTaskIncrementTick()”
  Another better place is at “vTaskSwitchContext()” if switch is to Idle task .

- In validation stage call for set task tag should be done directly after task creation not at task start To ensure intial task startup is plotted.
- GPIO PINs 10→16 are used for uart1 so we can not use them in Task Trace with GPIO.
- Task execution time using Timer1and Trace Function:

SYSTEM VALIDATION

Time is measured in terms of Timer1 Tick ,where Tick =1/$Timer_{1\ freq}$ = 16.67usec ,
$Timer_{1\ freq}$ = System Freq/1000 =60MHZ/1000=60KHZ

LD1 Task EXE Time= 299 tick =4.983ms
LD2 Task EXE Time=725 tick =12.083ms
Transmitter Task EXE Time =1 tick.=16.67usec
UART EXE Task =1tick =16.67 usec
B1 Task EXE Time =1 tick.=16.67usec
B2 Task EXE Time =1 tick.=16.67usec

NOTE :BECAUSE EXECUATION TIMES FOR B1,B2,UART,TRANSMITTER ARE SO SMALL(IN TERMS ON MICRO SECONDS)AND TIMER1 RESOULTION IS NOT ENOUGH TO GIVE PRECISE TIMMINGS FOR THESE TASKS.

SYSTEM VALIDATION

Egypt FWD 2022

Advanced Embedded
Track
RTOS Project