

---

# EDF SCHEDULER IMPLEMENTATION USING FREERTOS V12

---

Udacity Embedded Systems advanced nanodegree



ABDELRAHMAN HISHAM LABEED

FWD/UDACITY  
RTOS/Project\_2

## Table of Contents

<b><i>System verification .....</i></b>	<b><i>2</i></b>
<b><i>1- Using analytical methods .....</i></b>	<b><i>2</i></b>
<b><i>1- System hyperperiod.....</i></b>	<b><i>2</i></b>

# System verification

## 1- Using analytical methods

Task	Period (ms)	Execution time (ms)	Times/100ms
Button 1	50	0.014	2
Button 2	50	0.014	2
Periodic Transmitter	100	0.020	1
UART	20	0.03	5
Load Simulation 1	10	5	10
Load Simulation 2	100	12	1

1 System hyper-period = LCM(50,100,20,10) = 100ms

## 2- CPU Load

$$\text{CPU Load} = 0.014 * 4 + 0.02 + 0.03 * 5 + 10 * 5 + 12 \approx 10 * 5 + 12 = 62.2\%$$

Comment: expected results because load 1 and load 2 tasks got much higher execution time than the other Tasks

## 3- System stimulability Using URM (for rate monatomic scheduler)

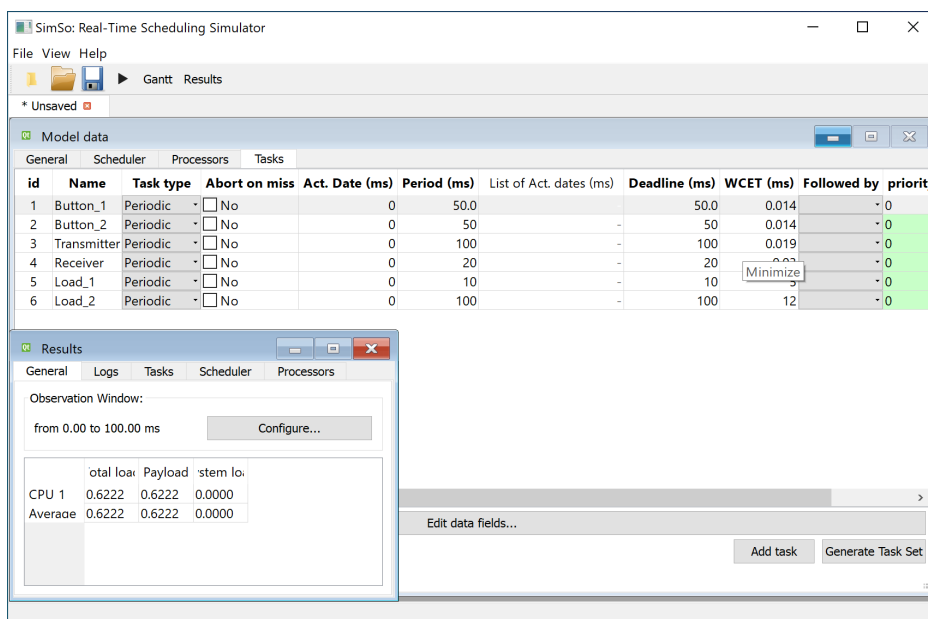
$$U = \sum \frac{C_i}{P_i} = \frac{0.014}{50} * 2 + \frac{0.02}{100} + \frac{0.03}{20} + \frac{5}{10} + \frac{12}{100} = 0.62$$

$$URM = 6 * (2^{\frac{1}{6}} - 1) = 0.73477228985$$

$URM > U \rightarrow$  system is schedulable

Comment: the system is schedulable using rate monatomic scheduler

## 2- Using offline simulator simso



### 3- Using Keil

```

main.c | FreeRTOSConfig.h | task.c | port.c | portASM.s
91  //Trace Hooks/
92  #define traceTASK_SWITCHED_IN() do\
93  {\
94  {\
95  if((int)pxCurrentTCB->pxTaskTag == 1)\
96  {\
97  GPIO_write(PORT_0, PIN3, PIN_IS_HIGH);\
98  GPIO_write(PORT_0, PIN2, PIN_IS_LOW);\
99  task_1_in_time = TITC;\
100  }\
101  else if((int)pxCurrentTCB->pxTaskTag == 2)\
102  {\
103  GPIO_write(PORT_0, PIN4, PIN_IS_HIGH);\
104  GPIO_write(PORT_0, PIN2, PIN_IS_LOW);\
105  task_2_in_time = TITC;\
106  }\
107  else if((int)pxCurrentTCB->pxTaskTag == 3)\
108  {\
109  GPIO_write(PORT_0, PIN5, PIN_IS_HIGH);\
110  GPIO_write(PORT_0, PIN2, PIN_IS_LOW);\
111  task_3_in_time = TITC;\
112  }\
113  else if((int)pxCurrentTCB->pxTaskTag == 4)\
114  {\
115  GPIO_write(PORT_0, PIN6, PIN_IS_HIGH);\
116  GPIO_write(PORT_0, PIN2, PIN_IS_LOW);\
117  task_4_in_time = TITC;\
118  }\
119  else if((int)pxCurrentTCB->pxTaskTag == 5)\
120  {\
121  GPIO_write(PORT_0, PIN7, PIN_IS_HIGH);\
122  GPIO_write(PORT_0, PIN2, PIN_IS_LOW);\
123  task_5_in_time = TITC;\
124  }\
125  else if((int)pxCurrentTCB->pxTaskTag == 6)\

```

```

main.c | FreeRTOSConfig.h | task.c | port.c | portASM.s
130  }while(0)
131  }
132  #define traceTASK_SWITCHED_OUT() do\
133  {\
134  {\
135  if((int)pxCurrentTCB->pxTaskTag == 1)\
136  {\
137  GPIO_write(PORT_0, PIN3, PIN_IS_LOW);\
138  task_1_out_time = TITC;\
139  task_1_total_time += (task_1_out_time-task_1_in_time);\
140  }\
141  else if((int)pxCurrentTCB->pxTaskTag == 2)\
142  {\
143  GPIO_write(PORT_0, PIN4, PIN_IS_LOW);\
144  task_2_out_time = TITC;\
145  task_2_total_time += (task_2_out_time-task_2_in_time);\
146  }\
147  else if((int)pxCurrentTCB->pxTaskTag == 3)\
148  {\
149  GPIO_write(PORT_0, PIN5, PIN_IS_LOW);\
150  task_3_out_time = TITC;\
151  task_3_total_time += (task_3_out_time-task_3_in_time);\
152  }\
153  else if((int)pxCurrentTCB->pxTaskTag == 4)\
154  {\
155  GPIO_write(PORT_0, PIN6, PIN_IS_LOW);\
156  task_4_out_time = TITC;\
157  task_4_total_time += (task_4_out_time-task_4_in_time);\
158  }\
159  else if((int)pxCurrentTCB->pxTaskTag == 5)\
160  {\
161  GPIO_write(PORT_0, PIN7, PIN_IS_LOW);\
162  task_5_out_time = TITC;\
163  task_5_total_time += (task_5_out_time-task_5_in_time);\
164  }\

```

