

Implementation of an EDF Scheduler based on freeRTOS

egFWD Embedded Systems
Advanced Track

By:

Mohamed Elsayed

Verifying System implementation

Analytical Method

ID	TASK	Periodicity	Execution Time	Calls in Hyper period
1	Button_1_Monitor	50	0.022	2
2	Button_2_Monitor	50	0.022	2
3	Periodic_Transmitter	100	0.028	1
4	Uart_Receiver	20	0.027	5
5	Load_1_Simulation	10	5	10
6	Load_2_Simulation	100	12	1

Calculation of HyperPeriod

HyperPeriod = $LCM(Periodicities)$ = $LCM(50,50,100,20,10,100)$

HyperPeriod = 100

CPU Load Calculations

CPU LOAD = Total Time * 100 / HyperPeriod

$$\text{CPU LOAD} = \frac{\text{Total Time}}{\text{HyperPeriod}} * 100$$

$$\text{Total Time} = \sum_{i=1}^6 \text{ExecutionTime}_i * \text{Num of Calls In HyperPeriod}_i$$

$$\text{Total Time} = 0.022 * 2 + 0.022 * 2 + 0.028 * 1 + 0.027 * 5 + 5 * 10 + 12$$

$$U = \text{CPU LOAD} = 62.138132 * 100 / 100 = 62.138132\%$$

Stimulability Analysis using Rate Monotonic Utilization Bound

(Assuming the given set of tasks are scheduled using a fixed priority rate -monotonic scheduler)

if $U \leq U_{rm}$ then the system is schedulable

$$U_{rm} = n[2^{1/n} - 1]$$

$$U_{rm} = 6*(2^{1/6}-1) = 0.7347723$$

The system is guaranteed schedulable since $U < U_{rm}$.

Schedulability Analysis using 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

Tasks are organized by lowest periodicity which is highest priority.

Task 5 -> Task 4 -> Task 1 -> Task 2 -> Task 3 -> Task 6

○ Task 5

$$W(4) = 5 + 0 = 5ms$$

Since $D = 10$ ms and $W(10) = 5ms$

$$W(10) < D$$

Task 5 is schedulable.

○ Task 4

$$W(20) = 0.027 + (20/10)*5 = 10.027ms$$

Since $D = 20$ ms and $W(20) = 10.027ms$

$$W(20) < D$$

Task 4 is schedulable.

○ Task 1

$$W(50) = 0.022 + (50/10)*5 + (50/20)*0.027 = 25.2875ms$$

Since $D = 50$ ms and $W(50) = 25.2875ms$

$$W(50) < D$$

Task 1 is schedulable.

Task 2

$$W(50) = 0.022 + (50/10)*5 + (50/20)*0.027 + (50/50)*0.022 = 25.1115ms$$

Since $D = 50$ ms and $W(50) = 25.1115ms$

$$W(50) < D$$

Task 2 is schedulable.

○ Task 3

$$W(100) = 0.028 + (100/50)*0.022 + (100/50)*0.022 + (100/10)*5 + (100/20)*0.027 = 50.251ms$$

Since $D = 100$ ms and $W(100) = 50.251ms$

$$W(100) < D$$

Task 3 is schedulable.

○ Task 6

$$W(100) = 12 + (100/100)*0.028 + (100/50)*0.022 + (100/50)*0.022 + (100/10)*5 + (100/20)*0.027 = 62.251ms$$

Since $D = 100$ ms and $W(100) = 62.251ms$

$$W(100) < D$$

Task 6 is schedulable.

Therefore, the system is schedulable.

Simso Simulation Method

SimSo: Real-Time Scheduling Simulator

File View Help

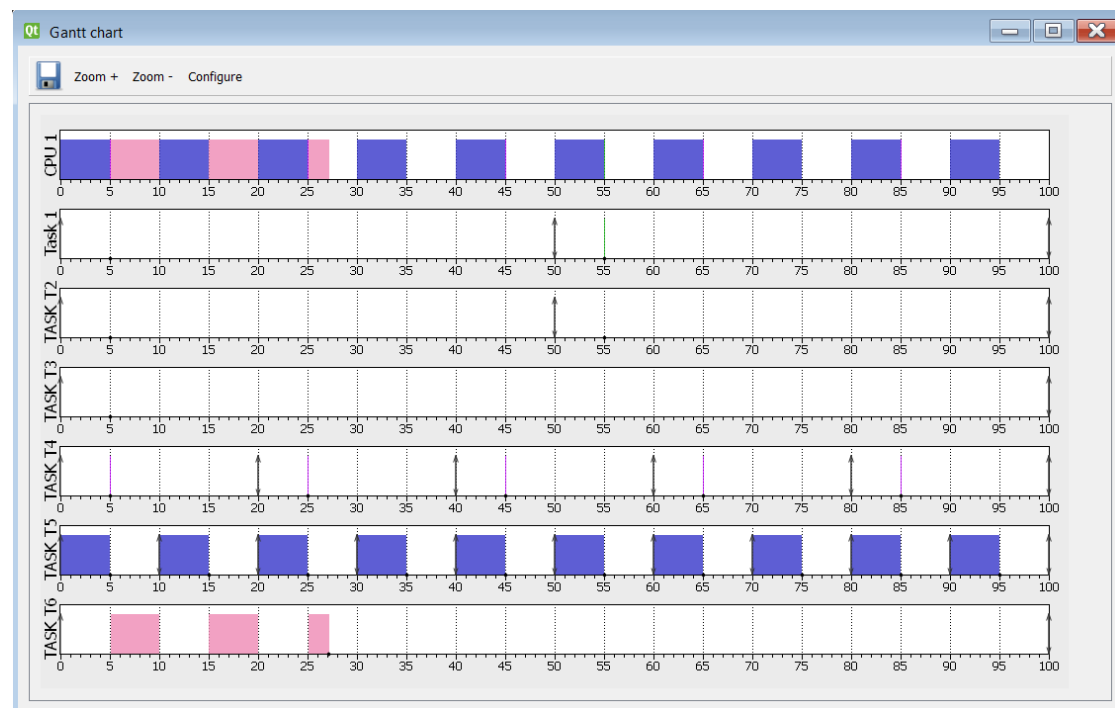
RTOS_Proj.xml

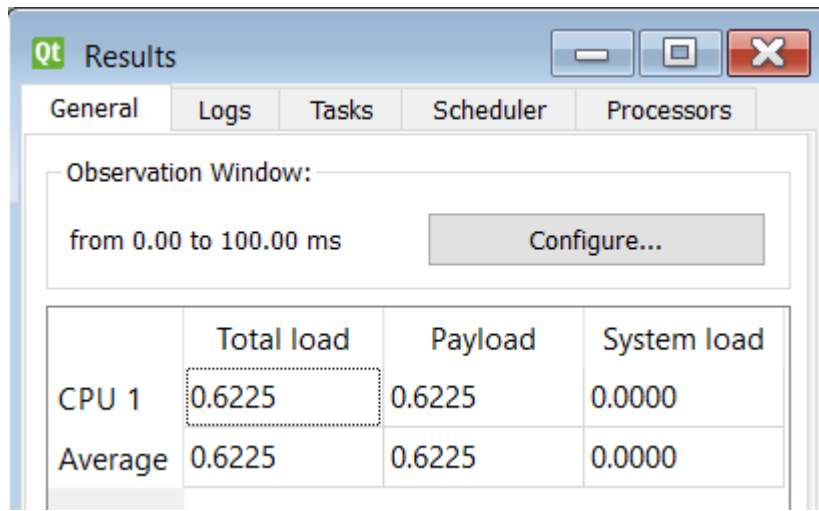
Model data

id	Name	Task type	Abort on miss	Act. Date (ms)	Period (ms)	List of Act. dates (ms)	Deadline (ms)	WCET (ms)	Followed by	priority
1	Task 1	Periodic	<input type="checkbox"/> No	0.0	50.0	-	50.0	0.022	2	2
2	TASK T2	Periodic	<input type="checkbox"/> No	0.0	50.0	-	50.0	0.022	2	2
3	TASK T3	Periodic	<input type="checkbox"/> No	0.0	100.0	-	100.0	0.028	1	1
4	TASK T4	Periodic	<input type="checkbox"/> No	0.0	20.0	-	20.0	0.027	3	3
5	TASK T5	Periodic	<input type="checkbox"/> No	0.0	10.0	-	10.0	5.0	4	4
6	TASK T6	Periodic	<input type="checkbox"/> No	0.0	100.0	-	100.0	12.0	1	1

Edit data fields...

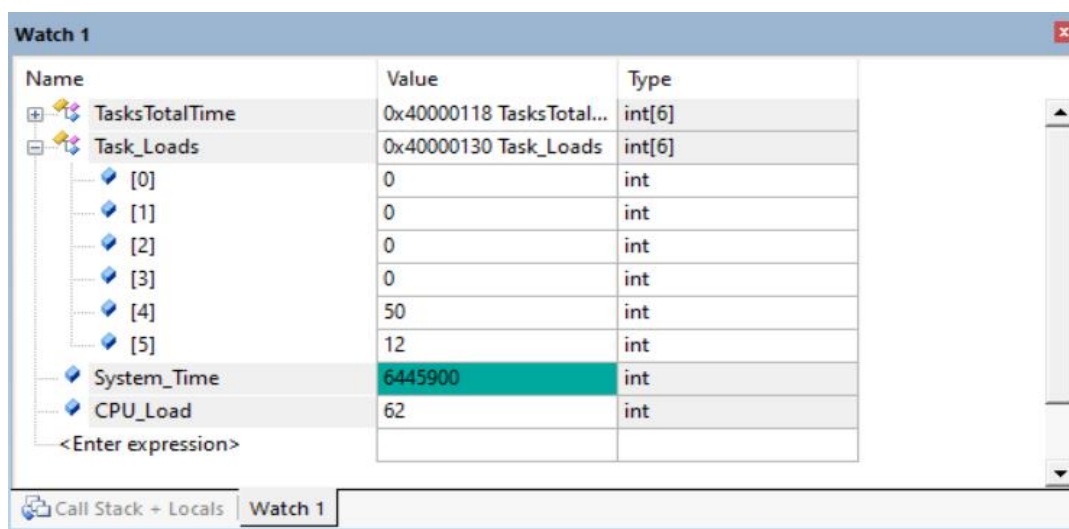
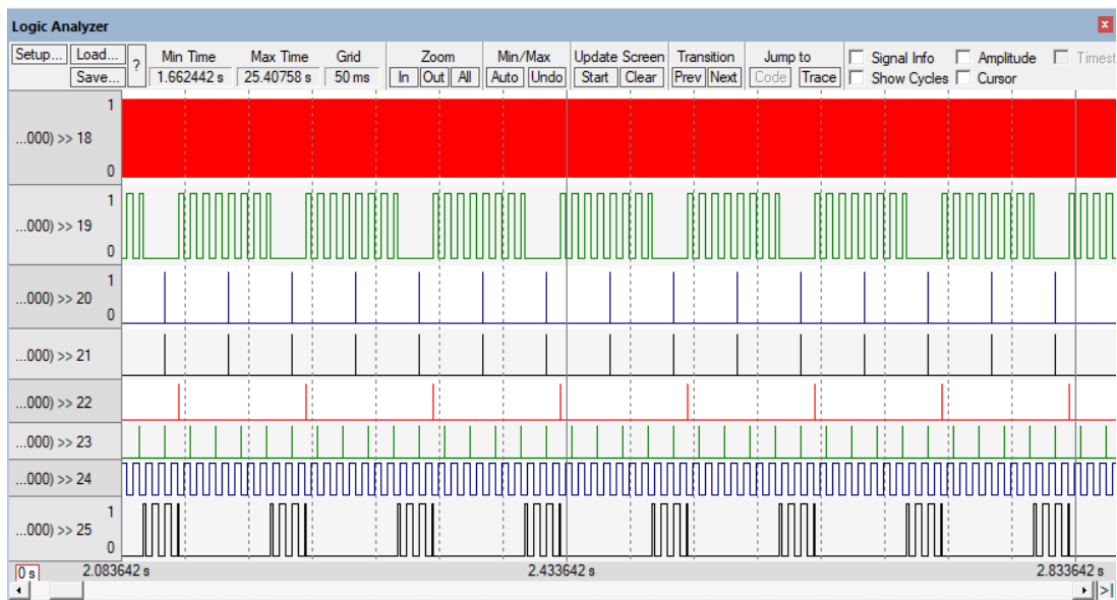
Remove selected task(s) Add task Generate Task Set





CPU Load almost the same as analytical method.

Keil Simulation Method



All CPU load calculations are almost the same which mean EDF scheduler implementation on freeRTOS kernel works as expected.