EDF Implementation Report

Verifying the System Implementation:

Method 1: Analytically

1. Calculate Hyper-period
2. CPU Load
3. Schedulability Analysis (using: A: Urm, B: Time Demand Analysis)

1. Calculate Hyper-period

Hyper-period (H)= LCM(Pi) = LCM (10 , 20 , 50 , 100) = 100

Therefore, Hyper-period = 100 ms

2. Calculate CPU Load

CPU Load (U):

U = R/C = Busy Time / (Busy Time + IDLE Time)

Note: Execution times of tasks calculated from the logic analyzer in Keil

U = ∑Ei/H = (17.7 µs*2 +18 µs *2+ 17.6 µs +49 µs *5+ 5 ms * 10 + 12 ms)/100 ms = 62.334%

3. Schedulability Analysis (using: A: Urm, B: Time Demand Analysis)

(A) Using Urm:

Urm =n $(2^{1/n} - 1) = 6(2^{1/6} - 1) = 73.477\%$

U = ∑ Ci/Pi = 17.7 µs / 50 ms + 18 µs / 50 ms + 17.6 µs / 100 ms + 49 µs / 20 ms + 5 ms /10 ms + 12 ms / 100 ms = 62.334%

Since U < Urm, Therefore System guaranteed Schedulable

(B) Using Time Demand Analysis:

Wi = ei + ∑ [t/Pk]*ek

Arrange Tasks according to Priority will be: [Task 5, Task 4, Task 1, Task 2, Task 3, Task 6]

For Task 5 :

W(1)…W(10)

W(1) = 5 + 0 = 5 ms , W(2) = 5 + 0 = 5 ms , W(3) = 5 + 0 = 5 ms , W(4) = 5 + 0 = 5 ms ,

W(5) = 5 + 0 = 5 ms, W(6) = 5 + 0 = 5 ms , W(7) = 5 + 0 = 5 ms , W(8) = 5 + 0 = 5 ms,

W(9) = 5 + 0 = 5 ms, W(10) = 5 + 0 = 5 ms , W(10) < D = 5ms < 10 ms , Task 5 is Schedulable

For Task 4 :

W(1)…W(20)

W(1) = 49 µs + (1/10)*5 = 0.55 ms , W(2) = 49 µs + (2/10)*5 = 1.05 ms, W(3) = 49 µs + (3/10)*5 = 1.55 ms, W(4) = 49 µs + (4/10)*5 = 2.05 ms, W(5) = 49 µs + (5/10)*5 = 2.55 ms, W(6) = 49 µs + (6/10)*5 = 3.05 ms, , W(7) = 49 µs + (7/10)*5 = 3.55 ms, W(8) = 49 µs + (8/10)*5 = 4.05 ms, W(9) = 49 µs + (9/10)*5 = 4.55 ms, , W(10) = 49 µs + (10/10)*5 = 5.05 ms, W(11) = 49 µs + (11/10)*5 = 5.55 ms , W(12) = 49 µs + (12/10)*5 = 6.05 ms, W(13) = 49 µs + (13/10)*5 = 6.55 ms, W(14) = 49 µs + (14/10)*5 = 7.05 ms, W(15) = 49 µs + (15/10)*5 = 7.55 ms, W(16) = 49 µs + (16/10)*5 = 8.05 ms, , W(17) = 49 µs + (17/10)*5 = 8.55 ms, W(18) = 49 µs + (18/10)*5 = 9.05 ms, W(19) = 49 µs + (19/10)*5 = 9.55 ms, , W(20) = 49 µs + (20/10)*5 = 10.05 ms, W(20) < D = 10.05 ms < 20 ms , Task 4 is Schedulable

And so on for the rest of the tasks, all tasks are guaranteed Schedulable

For Task 1:

Calculating W(1) … W(50)

W(50) = 17.7 µs + (50/10)*5 ms + (50/20)* 49 µs = 25.1402 ms , W(50) < D = 25.1402 ms < 50 ms , Task 1 is Schedulable

For Task 2:

Calculating W(1) … W(50)

W(50) = 18 µs + (50/50)*17.7 µs + (50/10)*5 ms + (50/20)* 49 µs = 25.1582 ms , W(50) < D = 25.1582 ms < 50 ms , Task 2 is Schedulable

For Task 3:

Calculating W(1) … W(100)

W(100) = 17.6 µs + (100/50)*18 µs + (100/50)*17.7 µs + (100/10)*5 ms + (100/20)* 49 µs = 25.1582 ms , W(100) < D = 50.334 ms < 100 ms , Task 3 is Schedulable

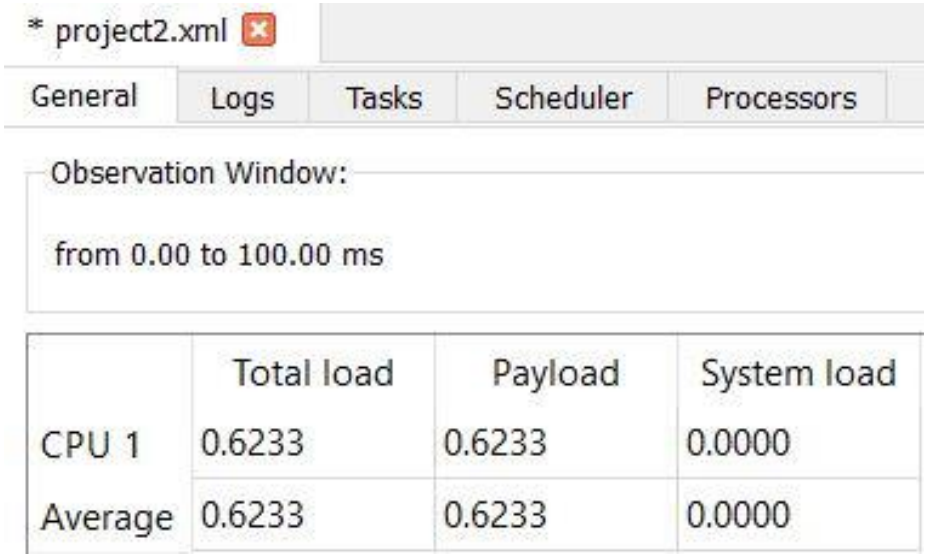
For Task 6:

Calculating W(1) … W(100)

W(100) = 12 ms + (100/100)17.6 µs + (100/50)*18 µs + (100/50)*17.7 µs + (100/10)*5 ms + (100/20)* 49 µs = 62.334 ms , W(100) < D = 62.334 ms < 100 ms , Task 6 is Schedulable

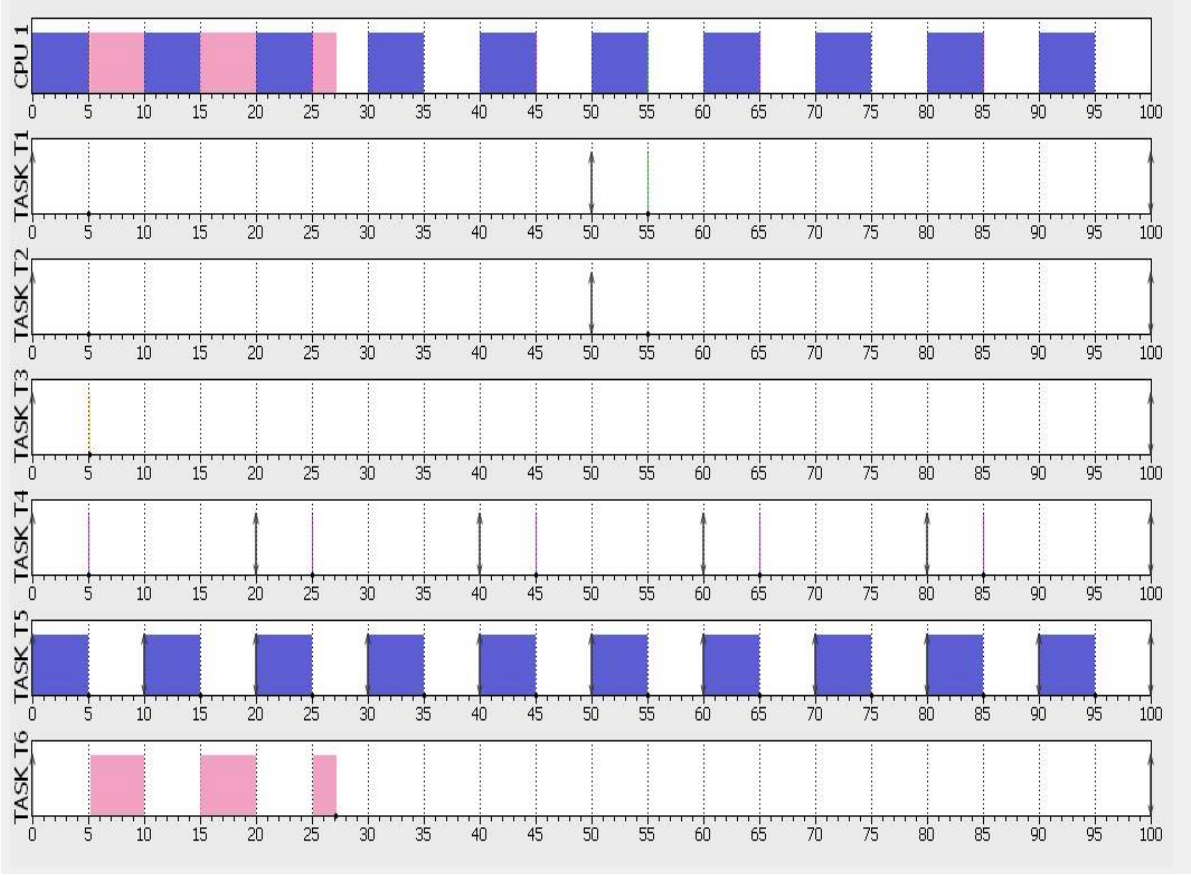
Method 2: Using SimSo Real-Time Scheduling Simulator

CPU Load



	Total load	Payload	System load
CPU 1	0.6233	0.6233	0.0000
Average	0.6233	0.6233	0.0000

Gantt Chart

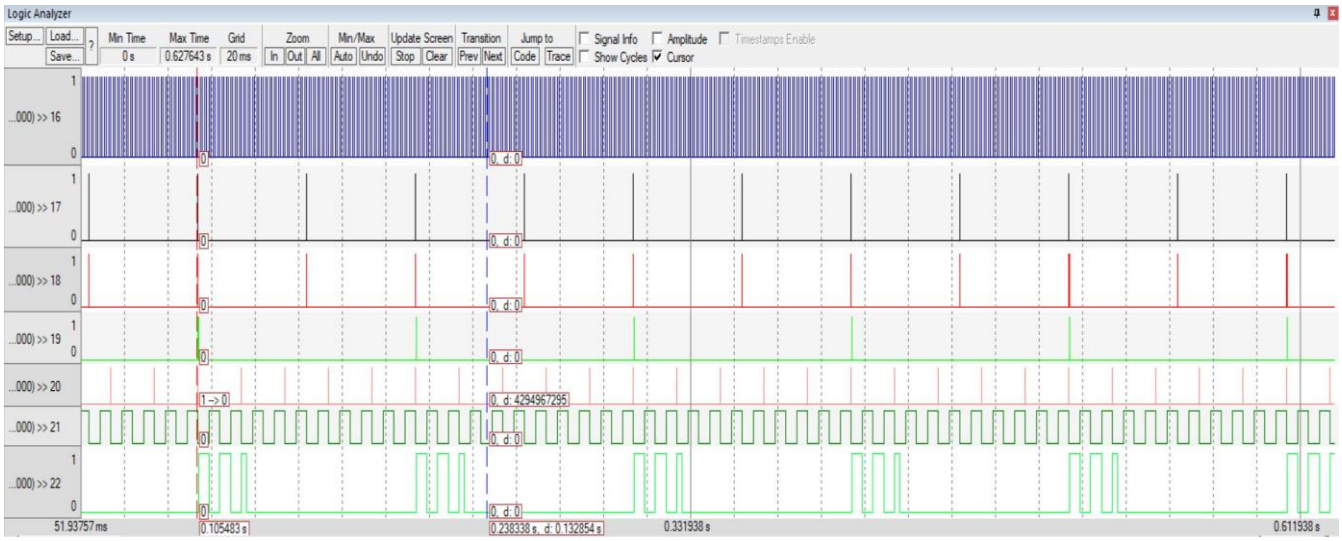


Tasks Creation

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	priority
1	TASK T1	Periodic	No	0	50	-	50	0.0177		0
2	TASK T2	Periodic	No	0	50	-	50	0.018		0
3	TASK T3	Periodic	No	0	100	-	100	0.0176		0
4	TASK T4	Periodic	No	0	20	-	20	0.049		0
5	TASK T5	Periodic	No	0	10	-	10	5		0
6	TASK T6	Periodic	No	0	100	-	100	12		0

Method 3: Using Keil



CPU Load(U) = 3E =62%

Watch 1

Name	Value	Type
system_time	0x00075F43	int
cpu_load	0x0000003E	int
(PORT0 & 0x00040000) >> 18	0x00000000	ulong
xReadyTasksListEDF	0x400001FC &xReady...	struct xLIST
cnt	<cannot evaluate>	uchar
xPortStartScheduler()	<cannot evaluate>	uchar
delayIDLE	0x00002043	int
delayA	<cannot evaluate>	uchar
Comm_Queue_Handle	0x400002D0	struct QueueDefinitio...
cpu_load	0x0000003E	int
<Enter expression>		

Comment on the Results:

As We see the results of the three methods give the same CPU load = 62%, which means a successful implementation.

Table of Tasks

Task	Task Information
TICK_HOOK	○ Logic Analyzer pin: Pin 0, Port 0
Task 1 Button_1_Monitor (Pin 0, Port 1)	○ Periodicity: 50 ms ○ Deadline: 50 ms ○ Execution Time: 17.7 µs ○ Task Tag: 1 ○ Logic Analyzer pin: Pin 1, Port 0 ○ Priority: 1 ○ Button1 _ID (ON): 1 ○ Button1_ID (OFF): 2
Task 2 Button_2_Monitor (Pin 1, Port 1)	○ Periodicity: 50 ms ○ Deadline: 50 ms ○ Execution Time: 18 µs ○ Task Tag: 2 ○ Logic Analyzer pin: Pin 2, Port 0 ○ Priority: 1 ○ Button2_ID (ON): 3 ○ Button2_ID (OFF): 4
Task 3 Periodic_Transmitter	○ Periodicity: 100 ms ○ Deadline: 100 ms ○ Execution Time: 17.6 µs ○ Task Tag: 3 ○ Logic Analyzer pin: Pin 3, Port 0 ○ Priority: 1 ○ Periodic_String_Available_ID: 5
Task 4 UART_Receiver	○ Periodicity: 20 ms ○ Deadline: 20 ms ○ Execution Time: 49 µs ○ Task Tag: 4 ○ Logic Analyzer pin: Pin 4, Port 0 ○ Priority: 1
Task 5 Load_1_Simulation	○ Periodicity: 10 ms ○ Deadline: 10 ms ○ Execution Time: 5 ms ○ Task Tag: 5 ○ Logic Analyzer pin: Pin 5, Port 0 ○ Priority: 1
Task 6 Load_2_Simulation	○ Periodicity: 100 ms ○ Deadline: 100 ms ○ Execution Time: 12 ○ Task Tag: 6

	○ Logic Analyzer pin: Pin 6, Port 0 ○ Priority: 1