Name: Yasser Waleed
Date: 04/10/2022

PROJECT 2
Implementing EDF Scheduler

Criteria 4 : Verifying the System Implementation

Note: The execution time for all tasks is calculated using GPIOs and the logic analyzer, check the last page.

1. Using analytical methods calculate the following for the given set of tasks:

<u>- Calculate the system hyperperiod:</u>

Hyperperiod is the lowest common multiple of the periodicity of all tasks. Since the periodicity of all tasks is given, we can find hyperperiod as follows:

Hyperperiod = LCM (50, 50, 100, 20, 10, 100) = 100

<u>- Calculate the CPU load:</u>

The CPU is the busy time over the total time. Let's consider the CPU for one hyperperiod, so total time = 100. It was found that tasks 1 to 4 share a similar execution time of around 14µs. Meanwhile, task 5 has 5ms execution time whereas task 6 has 12ms execution time. When it comes to the frequency of occurrence of each task within a single hyperperiod, we find that tasks 3 and 6 occur only once, tasks 1 and 2 occur twice, task 4 occurs 5 times, and task 5 occurs 10 times. In total that gives:

CPU_load = (0.014*(2 +2 + 1 + 5) + 5*10 + 12*1) / 100 = 0.6214 = 62.14%

<u>- Check system schedulability using URM and time demand analysis techniques</u>
(Assuming the given set of tasks are scheduled using a fixed priority rate -monotonic scheduler)

The schedulability test used for a rate monotonic scheduler is as shown below, where n is the number of tasks.

$$U \leq U_{RM} = n(2^{\frac{1}{n}} - 1)$$

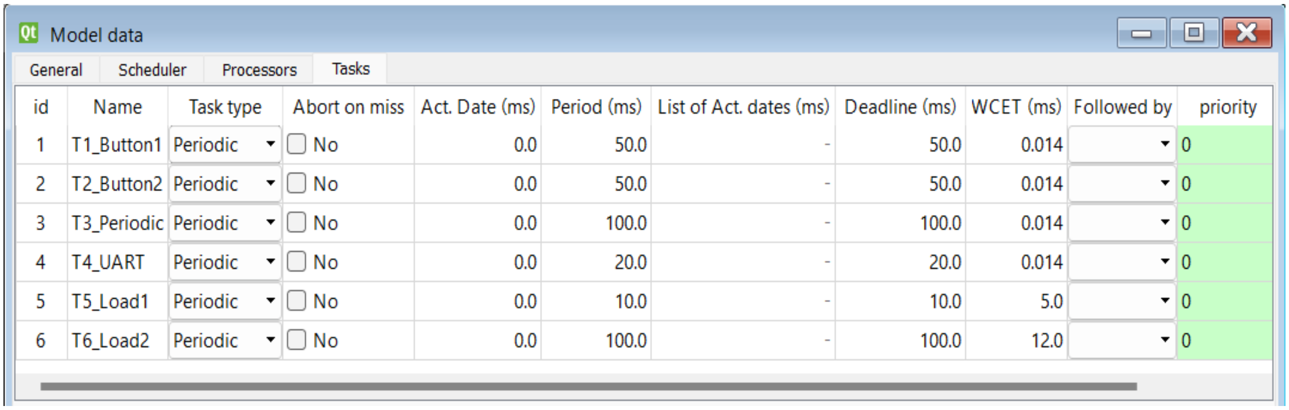
$$U_{RM} = 6(2^{\frac{1}{6}} - 1) = 0.7348$$

Since 0.6214 ≤ 0.7348, the system is schedulable.

Name: Yasser Waleed
Date: 04/10/2022

2. Using Simso (Offline Simulator)

Shown below are the tasks with their respective period, execution time, and deadline.

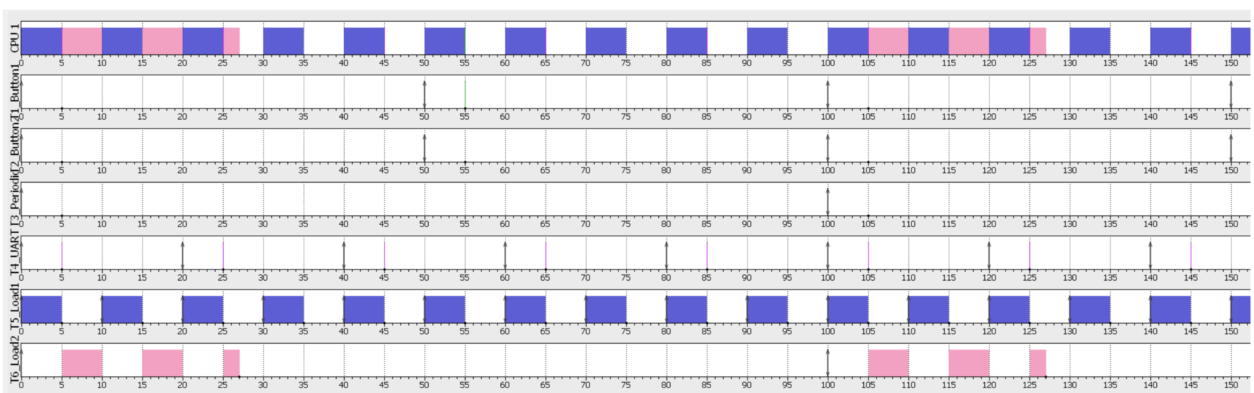


Model data

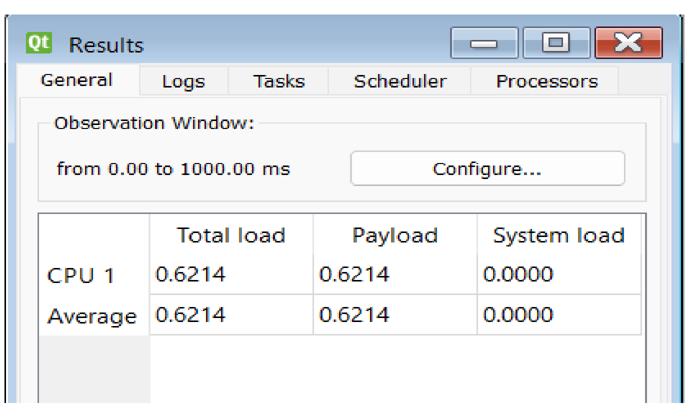
General | Scheduler | Processors | Tasks

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	T1_Button1	Periodic	No	0.0	50.0	-	50.0	0.014		0
2	T2_Button2	Periodic	No	0.0	50.0	-	50.0	0.014		0
3	T3_Periodic	Periodic	No	0.0	100.0	-	100.0	0.014		0
4	T4_UART	Periodic	No	0.0	20.0	-	20.0	0.014		0
5	T5_Load1	Periodic	No	0.0	10.0	-	10.0	5.0		0
6	T6_Load2	Periodic	No	0.0	100.0	-	100.0	12.0		0

With a fixed priority rate monotonic scheduler, the gantt chart is as shown below. The system repeats itself every 100ms, which is the hyperperiod as calculated above.



The resulting CPU load is shown below, also 62.14%.



Results

General | Logs | Tasks | Scheduler | Processors

Observation Window:

from 0.00 to 1000.00 ms

Configure...

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

Name: Yasser Waleed
Date: 04/10/2022

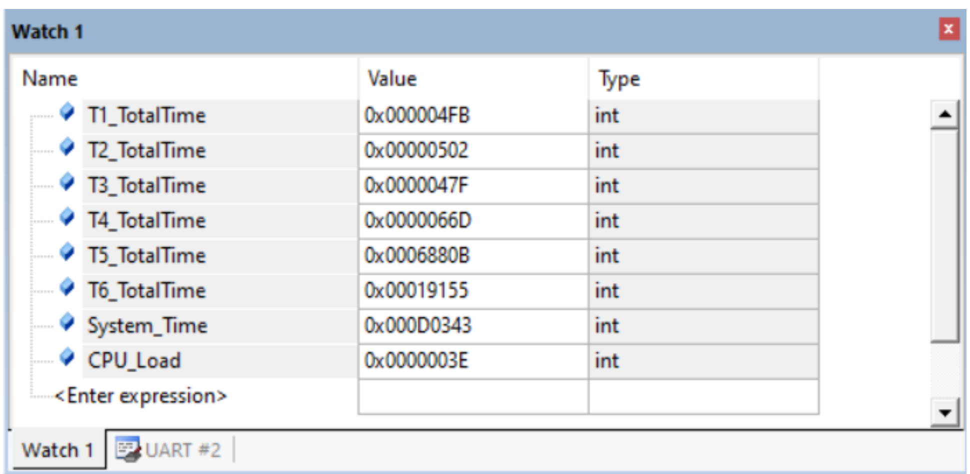
3. Using Keil simulator in run-time

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

Using the trace macros, traceTASK_SWITCHED_OUT and traceTASK_SWITCHED_IN, we can record the times of entry and the times of exit for each task, and add their difference to the total on duration of the task in the variable T#_TotalTime. As for the system time, we can directly record the latest timer value. The CPU_load updating code in the trace hook is as follows:

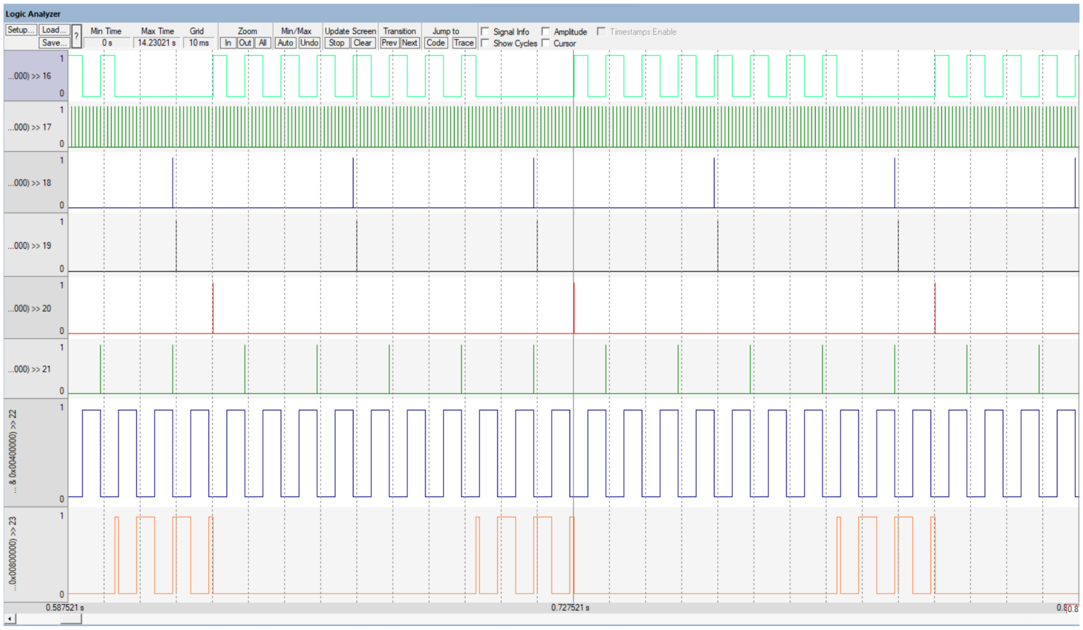
CPU_Load = ((T1_TotalTime + T2_TotalTime + T3_TotalTime + T4_TotalTime + T5_TotalTime + T6_TotalTime) / (float) System_Time) * 100;

With that, when running the simulation, we can view the value of the CPU_Load on the Watch window. Shown below is the CPU_Load percentage settling at 0x3E, that is 62%.



Name: Yasser Waleed
Date: 04/10/2022

Shown below is the logic analyzer with 8 graphs:



The 1st graph (pin16) is the idle task which sets the pin to 1, so whenever pin16 is high, the idle task is running, and whenever it is low, the other tasks are running (pin16 is set to low by all 6 tasks).

The 2nd graph (pin17) is the tick hook which is highly frequent, thus showing a denser graph.

The 3rd and 4th graph (pin18 and pin19) are button1 and button2 monitoring tasks, respectively. They both show up every 50ms.

The 5th graph (pin20) is the periodic transmitter showing up every 100ms.

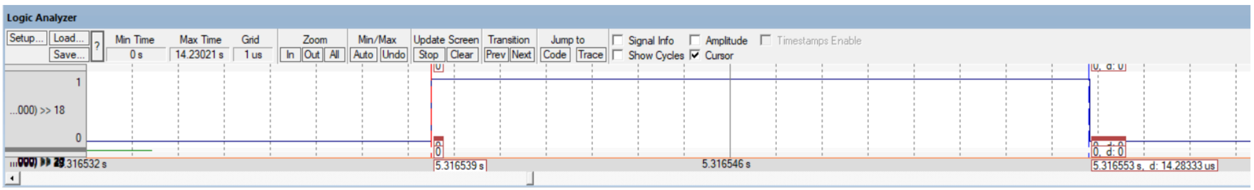
The 6th graph (pin21) is the UART receiver showing up every 20ms

The 7th and 8th graph (pin22 and pin23) are load1 and load2 simulating tasks, respectively. Load1 shows up every 10ms, and load2 shows up every 12ms.

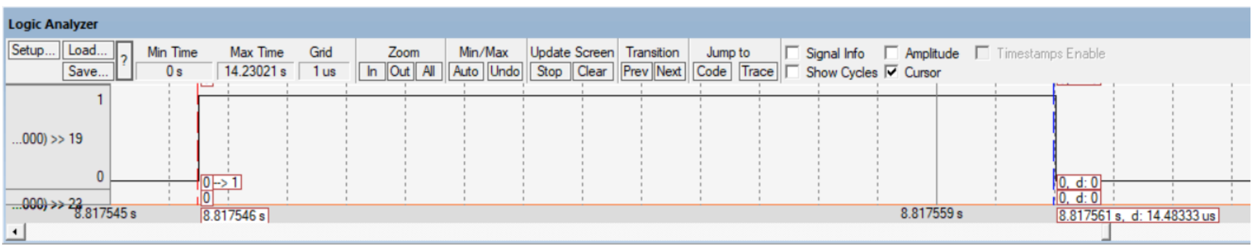
Name: Yasser Waleed
Date: 04/10/2022

For execution times of tasks 1 to 4, a closer look at each task's graph is needed:

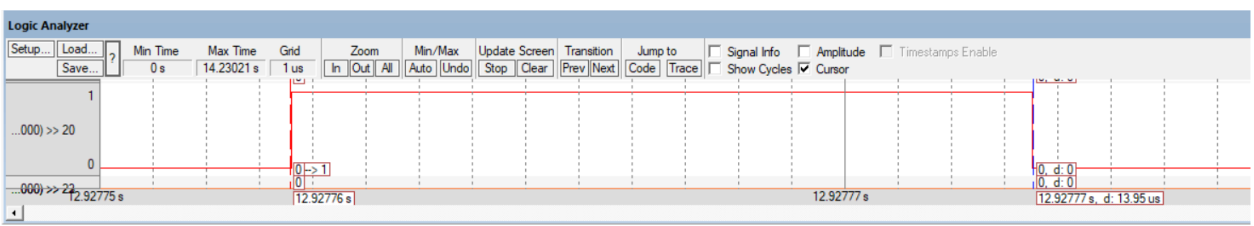
Task 1: 14.28µs



Task 2: 14.48µs



Task 3: 13.95µs



Task 4: 14µs