

University of
Waterloo



ECE455 Lab 4

EDF Scheduler on freeRTOS

Mahmoud Salem

m4salem@uwaterloo.ca

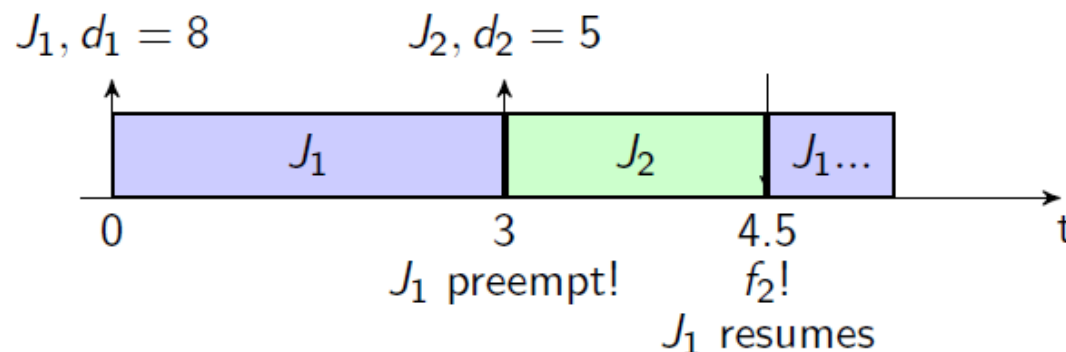
Please email through Learn

Earliest Deadline First (EDF)

Basics

- Priority-driven approach with dynamic priority assignment
Recall that priority-driven implies no idle time whenever possible!
- Ready queue is sorted: $J_i >_p J_k$, iff $d_i < d_k$.
- EDF is **optimal** with preemptive simple tasks on a uniprocessor.
- Optimal means EDF produces a feasible schedule, if one exists.

Example:



freeRTOS

- Documentation on website
<http://www.freertos.org/a00106.html>
- Demo project for KEIL MDK Simulator
[Download Here](#)
- freeRTOS scheduler uses
“Round Robin with pre-emption for tasks with equal priority”.

Start with the Sample Project

- Read the sample demo project [documentation](#).
 - “Queues” are not needed for the lab !
- Get it running as first step to understand how tasks are created and run periodically.
- Understand the logic analyzer output.

Implementing EDF on freeRTOS

- No Hardware needed !
- Do not modify the freeRTOS source code.
 - Implement the scheduler using the provided APIs.
- Understand APIs documentation on website, otherwise you will find unexpected scenarios.
- Design your **EDF scheduler** using these APIs.
- Implement and give good time for debugging.
 - Use breakpoints for this lab.

Lab 4 Problem

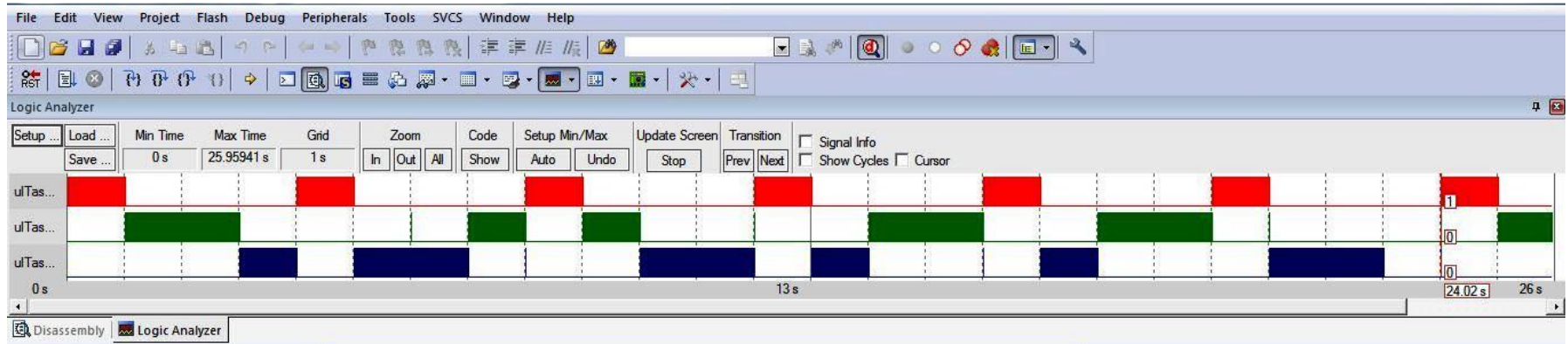
Consider 3 periodic tasks/jobs with execution times as follows:

T(exec time, period)

T1(1,4) T2(2,6) T3(3,8)

- **Show EDF scheduling diagram for the hyper-period**
 - **Assumption:**
 - On ties, Priorities $T1 > T2 > T3$.
 - T1,T2,T3 arrives at $t = 0$

Scheduling Diagram



- Not necessarily the exact output as this !
- Include it in your report and discuss it.
- [Optional] You can try it with different problem:
 - Ex: T1(1,4) T2(2,6) T3(5,12)

Important Tips

- Solve the problem by hand first !
- Change #defines to enable APIs and config RTOS.
- Tasks are implemented as endless loops
- Some APIs might cause immediate context switch.
- Make sure pre-emption won't lead to unexpected scenarios.
 - Hint: freeRTOS provides APIs for this.

Lab 4 Deliverables

- A zip file that have KEIL uVision project folder
 - Project folder should be compiling
 - Any resubmissions will not be accepted after deadline
- The binary file you used to test your code. (.axf)
 - Simply don't clean your project before submission.
- Short pdf report of maximum 1 page that briefly discuss your implementation and design.
 - ✓ **Snapshot of the scheduling diagram.**
 - ✓ Please write your name and ID in the report.

Lab Deadlines

- Posted to LEARN:
 - ~~Lab 1 is due on May 20th~~
 - ~~Lab 2 is due on June 10th~~
 - ~~Lab 3 is due on July 4th~~
 - **Lab 4 is due on July 20th (11.59pm)**

Demo

- Please choose time slot in this **online sheet**.
 - 10 minutes maximum per student
 - Time slot not for debugging !
- If you are **not** available from 9am to 12pm on Thursday 21 July and Friday 22 July, **Fill this online form to demo on Monday 25 July**.
- Demo using code submitted on LEARN
 - Show the scheduling diagram

TA Contact

- Use LEARN to send emails:
m4saalem@uwaterloo.ca
- Office E5-4119
- Make use of lab session on 14th of July.