

Assignments week 3 – Cooperative Scheduler

We will expand on the simple scheduler by implementing a list to which functions can be added to run at a specific period measured in ticks. We will thus need a **struct** in C to combine a function pointer with a period, a counter and a possible initial delay. This **struct** will describe a “task”. The goal is to be able to run a task at a specified period of e.g. 500 ticks.

3.1 Create a copy of the previous project and rename it to Assignment3.1.

- Using the description of this assignment, define a **struct** for a “task” and create a global array of 8 tasks.
- Create a function to initialize and add tasks to this task list (the array).
- In the SysTick ISR, walk through the task list and decrement each of the task counters.
- Think of, and expand on, the task struct to notify per task whether it is in a WAITING or READY state. Set the state in the ISR depending on the task counter.
- Create a function runReadyTasks() that will walk through the task list and execute any task in the READY state. Replace your **switch-case** rotation with a call to this function.
- Create 3 functions to toggle each led separately and add this to the task list with periods of: 200 ticks, 500 ticks, 750 ticks for red, yellow, and green respectively. Make use of a logic analyzer to verify the timing.

3.2 Now add initial delays to your tasks. Use an initial delay of 100, 200, 300 for red, yellow, and green respectively. Make use of a logic analyzer to verify the timing.

Enjoy the show of your ‘advanced’ cooperative (static) scheduler.