

UNC Charlotte, Department of Electrical and Computer Engineering

ECGR 3/5/6090, Fall 2003, Homework #11, Due: 12/1/03, at the beginning of class (20 points)

1. We have the following 3 tasks that we want to schedule with the Round Robin scheduler provided in `skp26.c`.

Tasks			
Task	Frequency	Priority	Duration
ToggleLed	Every 2 timer ticks	0	3ms
KickTheDog	Every 5 timer ticks	1	5ms
ReadADC	Every 3 timer ticks	2	12ms

- a. Assuming a timer tick of 10 ms, draw a timeline showing processor activity (the 3 tasks or scheduler) following the first 10 timer ticks. Note that this is a non-preemptive scheduler. Assume the scheduler takes a negligible amount of time to run, and no task is running when the first timer tick occurs. (5 points)
 - b. At the end of the tenth timer tick, show the contents of each task entry (`struct task_t`) for these 3 tasks. (3 points)
2. In `skp26.h` there is a macro called `mSECONDS(time)`. This macro is particularly helpful when you don't know the duration of the timer tick and want to schedule jobs using the Round Robin scheduler. Assuming that you don't know the length of the timer tick and the tasks in Q.1 executed every 20ms, 50ms and 30ms respectively; list the correct C code that will use the `addTask` function to add each of these tasks to the task list. (3 points)
3. For an 8 bit ADC with a 3V V_{ref} :
- a. What is the quantization error? (1 point)
 - b. What is the step size? (1 point)
 - c. What is the digital representation of 2.7V? (1 point)
 - d. What is the value of the analog signal whose representation is 10100100? (1 point)
4. Assuming that Thread B was running and the dispatcher decided to context switch to Thread A and the contents of the TCB for Thread A are as follows, show the contents of Thread A's stack before the `REIT` instruction in the dispatcher (shown on slide 11 of lecture notes 15) is executed. (5 points)

Thread A	Value
R0	0x13
R1	0x44
R2	0xA4
R3	0xC1
A0	0x0A56
A1	0x0670
Flag	0x35
ISP	0x0AFF
USP	0x0A00
FB	0x0A60
SB	0x04C0
PC	0xFA046

Address	Value
0x9FA	
0x9FB	
0x9FC	
0x9FD	
0x9FE	
0x9FF	
0xA00	
...	
0xAF9	
0xAFA	
0xAFB	
0xAFC	
0xAFD	
0xAFE	
0xAFF	