THE UNIVERSITY OF THE WEST INDIES
ST. AUGUSTINE

EXAMINATIONS OF December 2018

Code and Name of Course: **ECNG3006 Microprocessor Systems** Paper: **Final**

Date and Time: Duration: **Three (3) hours**

INSTRUCTIONS TO CANDIDATES: This paper has 5 pages and 3 questions. Max. Marks: **100**

ID# _____

Attempt ALL questions.
Questions 1 and 2 are each worth 25 marks.
Question 3 is worth 50 marks.

Questions Q3.a and Q3.b should be answered on page 5
and the script returned with your exam booklet.

The following reference information is provided:

- "Description of a Building Entry System" on page 4

. . . continued

Q1. FreeRTOS is an example of a commercial real-time operating system (RTOS) kernel which supports both mutexes and semaphores, as well as both co-operative, and pre-emptive, priority based multi-tasking systems.

(a) Explain, in your own words, why an RTOS is not necessarily required to implement a real-time embedded system. Your answer should identify at least two key characteristics of a real-time embedded system. *5 marks*

(b) Differentiate between the task scheduling, and context switching functions within the FreeRTOS kernel. *5 marks*

(c) If a task is taking too long to execute, what steps can be taken using FreeRTOS to eliminate or reduce any undesirable side-effects? *5 marks*

(d) Describe how priority inheritance is implemented within the FreeRTOS kernel. *5 marks*

(e) Identify and describe two mechanisms for implementing an aperiodic task using FreeRTOS such that the interference with periodic tasks is **bounded**. *5 marks*

[Q1 Total 25 marks]

Q2. A high-reliability temperature monitor for a walk-in refrigerator, consists of a PIC18 micro-controller connected to a 4x16 Liquid Crystal Display (LCD), as well as one or more analogue temperature sensor(s). Temperature sensors are read via the internal 10-bit successive approximation analog to digital converter (ADC) using an interrupt driven foreground task, and the LCD display is updated by the main background task.

The system is required to update the display and store the temperature readings each minute over a 24 hour period. Minimum, maximum and most recent readings are displayed on lines 1-3 of the LCD display respectively. To prevent inadvertent use of spoiled food, the system must display an alert on LCD line 4 if temperature rises above 4°C for more than 15 minutes at any time during the 24 hours. Each temperature reading is between 0°C and 19.9°C with accuracy of ±1°C and resolution of 0.1°C.

(a) Identify an appropriate representation scheme to store temperature readings. Use the readings 3.9°C and 18.5 °C, to illustrate your representation scheme, and justify your choice in terms of the system requirements. *5 marks*

(b) You may read a single sensor at 10-bit resolution, or read four sensors at 8-bit accuracy - averaging to get a 10-bit result. Which is better? Explain your answer. *5 marks*

(c) MISRA-C Rule 19.15 states: *"Precautions shall be taken in order to prevent the contents of a header file being included twice."* Use fragment(s) of C-code to show how compliance with this rule can be implemented. Predict how violation of this rule could potentially impact the background LCD display task. *5 marks*

(d) What effect will the internal analogue multiplexor have on the effective number of bits (ENOB) of the ADC temperature readings? What additional system information is needed to quantify the effect? *5 marks*

. . . continued

(e) *"Testing interrupt-rich code has always been recognized as difficult. This is due in large part to the un-predictability, and un-reproducibility, of real-world events".* Explain the rationale for this statement, and state whether you agree or disagree. Use the high-reliability temperature monitor to illustrate your point(s). *5 marks*

[Q2 Total 25 marks]

Q3. All questions in Q3 are based on Figure Q3 "Description of a Building Entry System". Jobs and tasks for such a system are described on page 4.

(a) Use the graph paper on page 5 to construct appropriate task timelines for the execution of these jobs in a foreground-background system where Tasks A,Q_i, are foreground tasks triggered as needed by the periodic kernel-service interrupt and Task B runs in the background. You should presume that card data is available on startup as described on page 4. *5 marks*

(b) Use the graph paper on page 5 to construct appropriate task timelines for the execution of these jobs assuming that they are scheduled using fixed priorities in a pre-emptive priority-based scheduling system with a 100ms kernel tick, where Task B has the highest priority, Task A has the lowest priority, where a blocked task will trigger a context switch to the next ready task, and each tick the next ready task in the queue is allowed to run. *10 marks*

(c) Compare the task and system real-time performance achieved using the foreground-background, and the pre-emptive priority-based scheduler. Your answer should clearly identify at least three criteria you could use for comparison, and explain which system you would prefer to implement on the basis of those criteria. *10 marks*

(d) You are told, that when executed, Job **JAj** has a relative deadline of 100 ms after the release of the associated instance of Task A. Use a precedence graph to determine the effective task deadlines of jobs in the **first** instance of Task A. Indicate the order in which Task A jobs would run if the Shortest-Job-First dynamic-priority-assignment algorithm was used. *10 marks*

(e) Identify three potential hazards/risks that the system may present, under any of the three specified scheduling schemes. Suggest mitigating strategies for each hazard/risk you identify. *15 marks*

[Q3 Total 50 marks]

END OF QUESTIONS

. . . continued

Figure Q3: Description of a Building Entry System

A building entry system is designed around a PIC18 micro-controller, powered at 5Vdc, with a 4MHz system clock. The system supports four entry/exit doors. There is a magnetic card reader and an electrically controlled lock at each door. A maximum of 1000 users may be issued with uniquely coded magnetic cards. The checksums for these cards are stored in an unsorted list in memory. The embedded kernel:

- allows the following tasks:
  - Task A responsible for serially polling each of the magnetic card readers, communicates with a different reader every 0.5 seconds (s).
  - Task B is responsible for verifying whether a specific magnetic card is valid, and is allowed entry.
  - Tasks Q_i where i= 0...3 each run once every 2s with a phase shift of $1.2 + 0.2 \times i$ seconds.
- supports inter-task messaging and/or synchronization
  - Tasks Q_i are synchronized with Task B; doors open only AFTER the associated instance of Task B has completed.
  - Task A sends 13 byte messages to Task B: (1 byte: door id, 1 byte: reader id, 10 bytes: card id, 1 byte: card id checksum)
- requires a kernel time-service interrupt every 0.1 seconds, with negligible overhead.

For analysis purposes you should consider operation of the system from t=0 to t=2s where:

- Card-data is available at all doors at t=0,
- Card read at Door 0 is last in Task B search list, at Door 1 is first in the Task B search list, at Door 2 is in the middle of the Task B search list, and at Door 3 is not in the Task B search list,
- Execution times for jobs JAb, JAf, and JBd are dependent upon serial response, or linear search time.

Task Name	Operation, Jobs, and Estimated Times
A	job(s) released periodically; i is a static variable request state of door i and clear buffer (JAa, e=1ms) wait for response from door i (JAb, e=1-5 ms) if door i has responded with the reader id (JAc, e=1 ms) add door number i, and reader id, to buffer (JAd, e=2 ms) while door i has more chars (JAe, max 11 iterations) request char from door i (JAf, e=1ms) wait for response from door i (JAg, e=1-5 ms) add char to buffer (JAh, e=1ms) if buffer is not empty (JAi, e=1ms) send 13 byte message to Task B (JAj, e=1ms) i = i+1 (JAk, e=1ms) if i is greater than 3 (JAl, e=1ms) i = 0 (JAm, e=1ms)
B	job(s) released on receipt of a message if door id, and reader identifier are both valid (JBa, e=4ms) calculate checksum (JBb, e=10ms) if checksum matches (JBc, e=1ms) top-down search list for checksum (JBd, e=1-1000ms) if checksum found in list (JBe, e=1ms) send semaphore to Task Q_i based on door id (JBf,e=1ms)
Q_i	job(s) released periodically if semaphore received from Task B (JDa, e=1ms) open door lock (JDb, e=1ms) else release door lock (JDc, e=1ms)

. . . continued

ID# _____

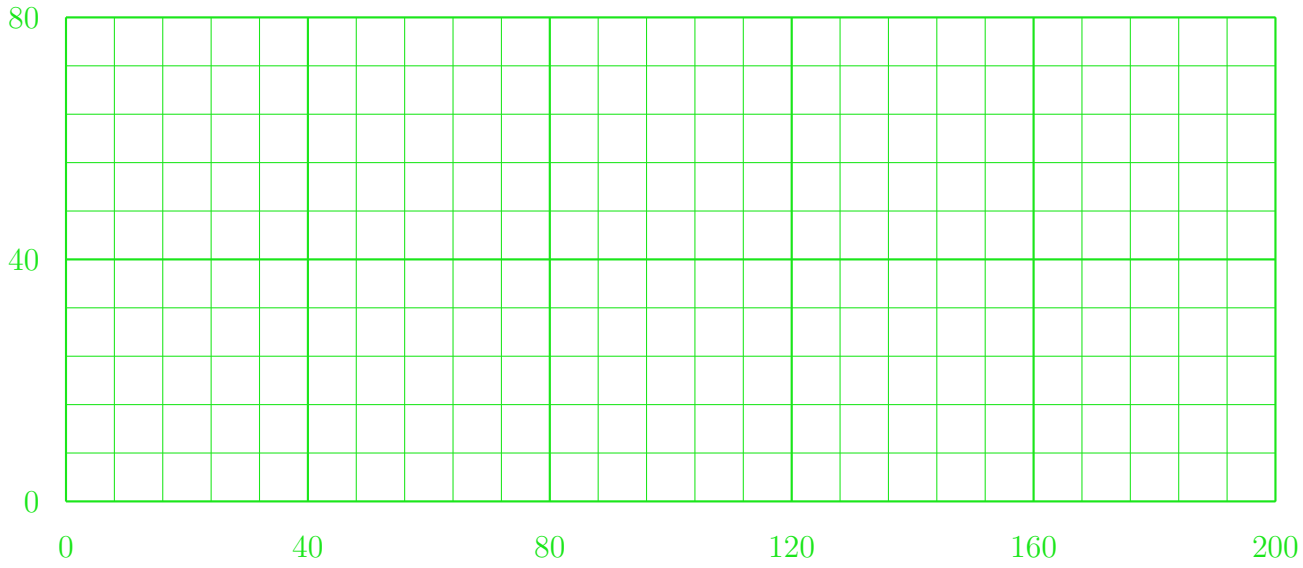

Figure Q3.a Grid for Foreground-Background Timelines

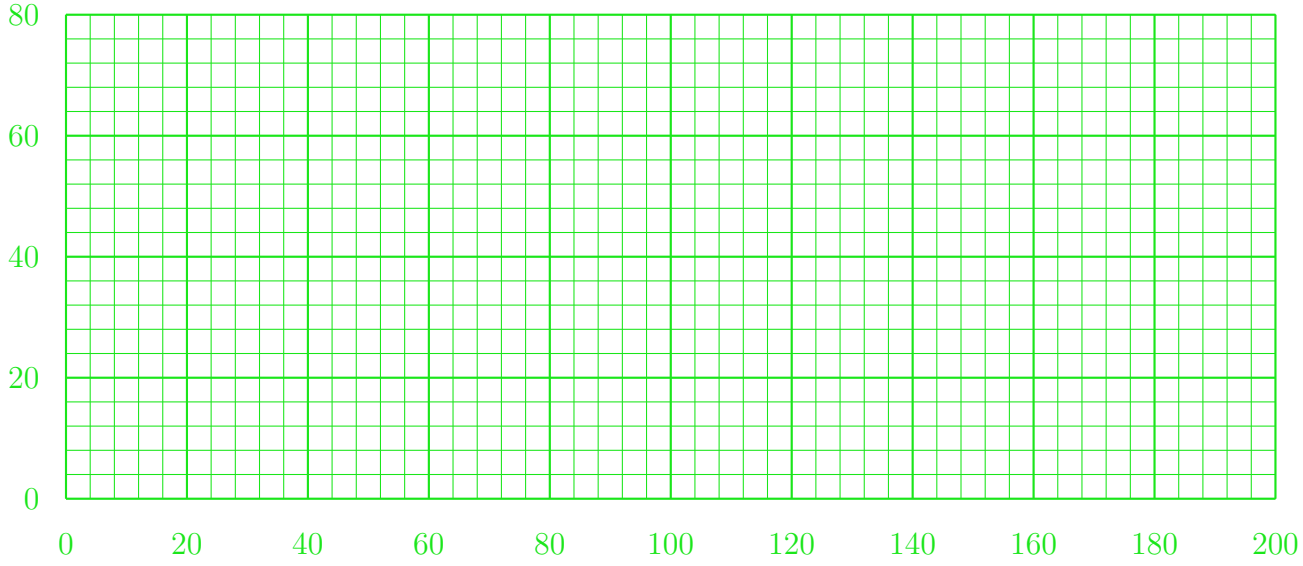

Figure Q3.b Grid for Pre-emptive Scheduler Timeline(s)

END OF EXAM PAPER