THE UNIVERSITY OF THE WEST INDIES
ST. AUGUSTINE

EXAMINATIONS OF December 2017

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 1-2 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 an Bluetooth-Enabled BLDC Inverter Refrigerator Controller on page 4

...continued

Q1. μCOS-II 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 μCOS-II kernel. *5 marks*

(c) Under what circumstances, will the use of mutexes cause deadlock in co-operative priority-based systems? Identify and explain features of μCOS-II which can be used to prevent deadlock. *5 marks*

(d) Identify elements of the Event Control Block(ECB) structure of the μCOS-II kernel. Outline how the identified elements facilitate message-related functions. *5 marks*

(e) Describe the concept of the idle task, and explain it's role in allowing the μCOS-II kernel to track actual system utilization. *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 4.1°C and 19.7 °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 15.2 states: "an unconditional break statement shall terminate every non-empty switch clause". Use a fragment of C-code to show a violation of this rule. Predict how violation of this rule could impact the background LCD display task. *5 marks*

. . . continued

(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*

(e) *"In high-reliability systems, where failures may endanger life, the use of interrupts is viewed with some suspicion."* Explain the rationale for this statement, and state whether you agree or disagree. Use the walk-in refrigerator to illustrate your point(s)? *5 marks*

[Q2 Total 25 marks]

Q3. Read Figure Q3.e "Description of an Bluetooth-Enabled BLDC Inverter Refrigerator Controller". These questions are based on this embedded system scenario with real time requirements. Jobs and tasks for such a system are described on page 4. The block diagram for the system is shown on page 4.

(a) Use the graph paper on page 5 to construct a cyclic executive schedule for the execution of Tasks 1 to 3 where the minor cycle is 50ms. You should presume that interrupts are never disabled. *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 a time-based task manager, with a 100ms tick, where blocked tasks will trigger a context switch to the next ready task, and each tick the next ready task in the queue is allowed to run. You should presume that interrupts are never disabled. *10 marks*

(c) Compare the task and system performance achieved using the cyclic executive, and the time-based task manager. 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 Job **J15** has a relative deadline of 50 ms after the release of Task 1. Use a precedence graph to determine the effective task deadlines of the jobs in Task 1, and indicate the order in which Task1 jobs would run if the Deadline Monotonic priority-assignment algorithm was used on the effective deadlines. *10 marks*

(e) Identify three potential hazards/risks that the system may present, if implemented using pre-emptive priority-based scheduling, where Task 1 has the highest priority, Task 3 has the lowest priority, and interrupts are never disabled. Suggest mitigating strategies for each hazard/risk. *15 marks*

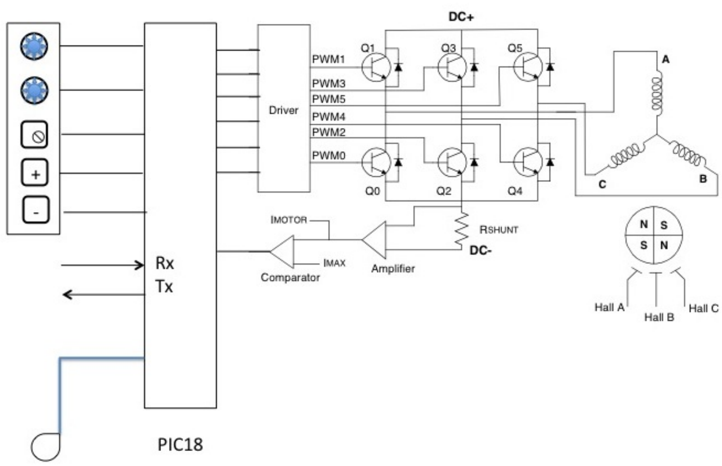
[Q3 Total 50 marks]

END OF QUESTIONS

. . . continued

Figure Q3 – Description of an Bluetooth-Enabled BLDC Inverter Refrigerator Controller

Based on: Microchip AN889 Brushless DC Motor Control Using PIC18FXX31 MCUs
`http://ww1.microchip.com/downloads/en/AppNotes/00899a.pdf`



The refrigeration controller consists of a PIC18 micro-controller, a control panel with 3 buttons and 2 LED's, an analogue temperature sensor, a variable-speed compressor (driven by a BLDC Inverter), and a BlueTooth serial replacement unit operating at 300 baud with 8 data bits, no parity and 1 stop bit.

System Input:	• Compressor Fault Signal (Fault/No Fault) • BLDC Hall Effect sensors (3-bit pattern - A, B, C) • Control Panel Buttons - momentary (3-bit pattern - Start/Stop, SP+, SP-) • Temperature Sensor (continuous value from 0 - 9 degrees Celsius) • Serial Port Rx (receives ASCII characters at 300baud, 8N1)
System Output:	• Control Panel Operating Light (On/Off) • Control Panel Alarm Light (On/Off) • BLDC Motor Switching Pattern (6-bit pattern); motor response time 10 ms • Serial Port Tx (transmits ASCII characters at 300 baud, 8N1)
System Resources:	**R1** Mutex associated with global variables `Running`, `Temperature Error` and `SetPoint` **S1** Semaphore associated with Compressor Faults, maximum value 1 **TxQ** Message queue containing characters for transmission via BlueTooth **RxQ** Message queue containing characters received via BlueTooth
Task 1: BLDC Control	Controls PWM signals for compressor; Runs at 100 ms intervals iff global boolean variable `Running` is TRUE. **J11, e=10ms** if `Running` == TRUE then release J12 and J13 **J12, e=5ms** from J11, set Operating Light ON; **J13, e=10ms** from J11, if Compressor Fault Signal == TRUE then generate S1 else release J14 and J15 **J14, e=5ms** from J13, lookup BLDC Motor Switching Pattern based on BLDC Hall Effect Sensor Pattern **J15, e=15ms** from J13, acquire R1; set PWM Duty Cycle based on `Temperature Error`; release R1
Task 2: Temperature Monitor	Reads Temperature Sensor; Runs at 0.5s intervals; local static variable `Temperature`. **J21, e=10ms** Update `Temperature` reading **J22, e=10ms** after J21, if `Temperature` exceeds 4°C then put '*' on TxQ **J23, e=10ms** after J21, acquire R1; calculate `Temperature Error` = `Temperature` - `SetPoint`; release R1
Task 3: Interface Control	Serial Communication + User Interface; Runs at 1s intervals. Phase Delay 0.25s; local variable `p`. **J31, e=20ms** If Start/Stop Button was pressed THEN Acquire R1, Running = ! Running, Release R1 **J32, e=20ms** if SP+ button was pressed THEN Acquire R1, Setpoint++, Release R1 **J33, e=20ms** if SP- button was pressed THEN Acquire R1, Setpoint–, Release R1 **J34, e=15ms** Consume S1 (5ms timeout) - If consumed Alarm Light = ON **J35, e=30ms** if TxQ is not empty, and Tx is IDLE, get character from TxQ and place in TxChar **J36, e=30ms** If RxQ is not empty, retrieve `p` from RxQ, if `p` is between '0' and '9' THEN Acquire R1, Setpoint = `p`-'0', Release R1
ISRHandler: Serial Communication	Manages TxQ and RxQ; Runs on demand - baud rate 300, 8 data bits, no parity bit, 1 stop bit. **Jisr1, e=2ms** if Rx Interrupt, Add RxChar to RxQ **Jisr2, e=2ms** if Tx Interrupt, get character from TxQ and place in TxChar

. . . continued

ID# ______________________

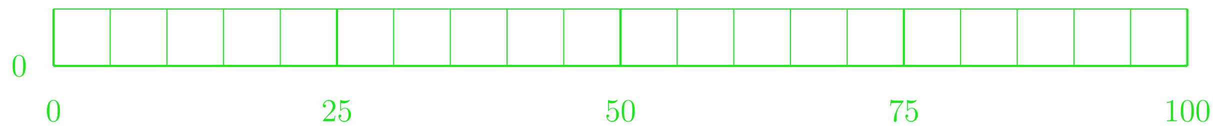


Figure Q3.a Grid for Cyclic Schedule for Job set $J11 \ldots J36$

Figure Q3.b Grid for Time-based Task Manager Timeline(s) for Job set $J11 \ldots J36$

END OF EXAM PAPER