Universiti Tun Hussein Onn Malaysia

UNIVERSITI TUN HUSSEIN ONN MALAYSIA

FINAL EXAMINATION
SEMESTER I
SESSION 2017/2018

COURSE NAME	:	REAL-TIME EMBEDDED SYSTEM
COURSE CODE	:	BEH 30802
PROGRAMME CODE	:	BEJ
EXAMINATION DATE	:	DECEMBER 2017 / JANUARY 2018
DURATION	:	2 HOURS 30 MINUTES
INSTRUCTION	:	ANSWER ALL QUESTIONS

THIS QUESTION PAPER CONSISTS OF **FIVE (5)** PAGES

Q1 "Almost every model car that rolls off the production line these days makes use of embedded technology in one form or the other; most of the embedded systems in automobiles are rugged in nature, as most of these systems are made up of a single chip. No driver clashes or 'systems busy' conditions happen in these systems. Their compact profiles enable them to fit easily under the cramped hood of a car. These systems can be used to implement features ranging from the auto suspension adjustment systems to antilock braking systems (ABS)."

(a) Define real-time system.

(2 marks)

(b) Name **TWO (2)** real time embedded systems in a model car.

(1 marks)

(c) Classify the identified real time embedded systems (from Q1 (b)) into either hard or soft real time systems with a concise justification each.

(4 marks)

(d) One of the main characteristic of a real time system is to achieve predictable behaviour. Relate this characteristic to the identified real time embedded systems (from Q1 (b)).

(3 marks)

Q2 (a) An LED is directly connected to the digital pin 6 of an Arduino Uno microcontroller that based on sinking mode. While a push button switch is connected to the digital pin 7 of the microcontroller that based on pull-down resistor concept so that the logic signal is low when the push button swich is released.

(i) Sketch a schematic for the microcontroller that connected to the LED and the push button.

(6 marks)

(ii) Write a complete C-statement code in ***void setup ()*** subroutine for relevent configurations.

(2 marks)

(iii) Write a complete C-statement code in ***void loop ()*** subroutine for controlling the LED based on the push button switch state, i.e. when the push button switch is released, the LED will be turned off, and vice versa.

(6 marks)

(b) A Bluetooth module is connected to the digital pin 0 and digital pin 1 of an Arduino (UNO) microcontroller as a transceiver; while an accelerometer sensor is connected to the analog pin A4 and A5 of the microcontroller.

(i) Distinguish synchronous and asynchronous serial communication methods.

(2 marks)

(ii) Classify the communication methods (i.e. synchronous or asynchronous) and protocol for the Bluetooth module and the accelerometer sensor (from Q2 (b)) with a concise explanation.

(4 marks)

Q3 (a) Give four good features of real time operating system (RTOS) over general purpose operating system (GPOS).

(2 marks)

(b) The FreeRTOS is a real-time operating system that can implements "fixed priority preemptive" scheduling algorithm. Draw and illustrate how the threads can change their state in FreeRTOS scheduler

(4 marks)

(c) The hardware system same as **Q2(a)** is used and its software is modified to handle three threads by using ChibiOS/RT. The code of the operation is given as follows:

```
#include < Arduino_FreeRTOS.h >
unsigned char BTN;

//Task 1
// to be completed as in Q3(c)(i) instruction.

//Task 2
void Task2(void *pvParameters __attribute__((unused))) {
  while (1){
    digitalWrite(4,BTN);
    vTaskDelay(50/portTICK_PERIOD_MS);
  }
}

//Task 3
void Task3(void *pvParameters __attribute__((unused))) {
  while (1){
    Serial.print(BTN);
    vTaskDelay(100/portTICK_PERIOD_MS);
  }
}

void setup()
{
  Serial.begin(9600);
  pinMode(3,INPUT);
  pinMode(4,OUTPUT);
  xTaskCreate(Task1,"T1",64,NULL,2,NULL);
  xTaskCreate(Task2,"T2",64,NULL,3,NULL);
  xTaskCreate(Task3,"T3",64,NULL,1,NULL);
}

void loop() { }
```

(i) Complete a task function for Task1 operation. This task will be executed every 25 ms (period) for reading a status of button switch and save in 'BTN' variable.

(4 marks)

(ii) Give detail explanation on the output prediction of the system operation.

(8 marks)

(iii) Give your comments on how to assign correct priority for each tasks.

(3 marks)

Q4 (a) Explain the following terminology in resource sharing application

(i) Counting Semaphore

(2 marks)

(ii) Mutex

(2 marks)

(iii) Deadlock

(2 marks)

(b) Analyse the operation of code in **Q3(c)** in term of deadlock, starvation and priority inversion events.

(6 marks)

(c) The coding in **Q3(c)** is going to add a binary semaphore for synchronising 'BTN' variable from Task1 to Task2 and Task3.

(i) Modify the existing code for declaring the semaphore token, wait and signal operation.

(6 marks)

(ii) Propose **TWO (2)** solutions to avoid deadlock condition when using semaphore.

(4 marks)

Q5 (a) Define the deadline and the maximum elapsed time (Max_E) of a task in temporal scope.

(2 marks)

(b) Assume a system has three independent tasks A, B, and C as given in **Table Q5**.

Table Q5: Task Specification

Task	Period (ms)	CPU resources (ms)
A	40	5
B	20	5
C	30	10

(i) If the priority level of Task A > Task B > Task C, draw a task activition diagram for the first 100ms of system operation.

(6 marks)

(ii) Construct a table to state the start delay, elapse time, and completion time for each task.

(6 marks)

(c) Another way to analyze the schedulability of the tasks in the **Table Q5** is by using full test of rate monotonic schedulability (RMS).

(i) Re-arrange the new priority level for each task that based on RMS concept.

(2 marks)

(ii) Calculate the worst-case completion time for each task.

(8 marks)

(iii) Comment on the ability of each task to meet its deadline and schedulability.

(3 marks)

-END OF QUESTIONS –