***King Fahd University of Petroleum and Minerals***

***College of Computer Science and Engineering***

***Computer Engineering Department***

**COE 306: INTRODUCTION TO EMBEDDED SYSTEMS**

**Term 161 (Fall 2016-2017)**

**Final Exam**

**Saturday Jan. 14, 2016**

**Time: 120 minutes, Total Pages: 10**

**Name:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ ID:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Section: \_\_\_\_\_\_\_**

**Notes:**

* Do not open the exam book until instructed
* Answer all questions
* All steps must be shown
* Any assumptions made must be clearly stated

|  |  |  |
| --- | --- | --- |
| **Question** | **Max Points** | **Score** |
| **Q1** | **27** |  |
| **Q2** | **14** |  |
| **Q3** | **8** |  |
| **Q4** | **11** |  |
| **Q5** | **15** |  |
| **Total** | **75** |  |

Dr. Aiman El-Maleh

# **[27 Points]**

# **(Q1)** Fill in the blank in each of the following questions:

## In a real time operating system, failure to meet hard deadlines causes \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ while failure to meet soft deadlines causes \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.

## Real time operating systems are based on the concepts of preemption and context switching. Preemption is \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.

## A process can be in one of three states: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.

## The context of a process is \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.

## Techniques for handling race conditions in shared memory that could lead to data inconsistency include \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.

## Disadvantages of using shared memory inter-process communication include\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

## \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.

## In message passing using pipes, one process writes data into \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ while another process \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.

## Using \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ socket, packets are independent with not necessarily in-order delivery and no guarantee of delivery.

## The advantage of serial transmission in comparison to parallel transmission is that it is \_\_\_\_\_\_\_\_\_\_\_ while the advantage of parallel transmission is that it is \_\_\_\_\_\_\_.

## The advantage of using differential signals in serial transmission is that it \_\_\_\_\_\_\_\_\_\_\_\_\_ the signal to noise ratio.

## 

## \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (synchronous/asynchronous) transmission is used for high-speed communication between computers.

## In \_\_\_\_\_\_\_\_\_\_\_ transmission, data flow is only in one direction while in \_\_\_\_\_\_\_\_\_\_\_\_\_\_ transmission, data flow is in both directions simultaneously.

## Given 9600 baud rate and 8 voltage levels used for transmitting each symbol, the bit rate is \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.

## Given a protocol with 3 bits of protocol (start, stop and parity), 7 bits of data, 9600 baud rate, and 1 bit per symbol (binary), the information rate is \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.

## SPI has \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (higher/lower) throughput than I2C.

## In SPI, transmission involves two \_\_\_\_\_\_\_\_\_ registers one in master and one in slave connected in \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ topology.

## Using UART, transmitting character 'M'=0x4D with even parity, 7 data bits and one stop bit, the following data waveform is generated:

## In UART, a framing error occurs when \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.

## In UART, an overrun error occurs when \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.

## In UART, a break condition is detected when \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.

**[14 Points]**

# **(Q2)** Assume that the following periodic processes are ready at time 0. We would like to schedule these processes using RMS and EDF scheduling techniques. In each case, compute the schedule for an interval equal to the least-common multiple of the periods of the processes. If processes have the same priority, schedule processes to minimize preemption and process waiting time. If processes have the same priority under all these criteria then give priority in the following order P1, P2, P3.

|  |  |  |
| --- | --- | --- |
| **Process** | **Execution Time** | **Period (Deadline)** |
| P1 | 1 | 3 |
| P2 | 1 | 4 |
| P3 | 2.5 | 6 |

1. Compute CPU utilization.
2. Schedule the processes using an RMS policy. If a schedule does not meet the required deadlines for all processes, complete the schedule and indicate this in your solution.
3. Schedule the processes using an EDF policy. If a schedule does not meet the required deadlines for all processes, complete the schedule and indicate this in your solution.
4. Is it possible to find a feasible schedule where all deadlines for all processes are met if the execution time of P2 is increase to 1.5? Justify your answer.

**[8 Points]**

# **(Q3)** Three periodic tasks T1, T2 and T3 are to be executed on a system with a single timer. The timer is configured to expire every 500μs. The task periods are 0.25s for T1, 0.5s for T2 and 1s for T3.

# Write a timer interrupt handler (ISR) in C that runs the tasks according to their periods using a single counter. Assume that each task is run by calling a function of its name, e.g. T1().

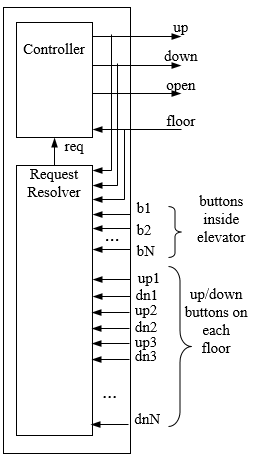
**[11 Points]**

**(Q4)**

1. It is required to interface a microcontroller as a master to three peripheral devices as slaves. **(6 points)**
2. Show the block diagram interconnection of the master and slaves using SPI interface.
3. Show the block diagram interconnection of the master and slaves using I2C interface.
4. Suppose that a master is interfaced to a slave using I2C interface. Assume that the slave address is 1100000. Show the timing diagram for the data and the clock line for reading a single byte from the slave. **(5 points)**

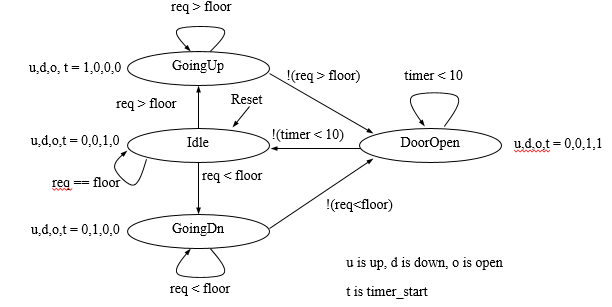
**[15 Points]**

**(Q5)** An elevator system of an N-Story building is composed of two units, a controller unit and a request resolver unit as shown below. The request resolver unit receives N buttons inside the elevator to indicate the desired floor. It also receives two buttons from each floor one for going up and another for going down requests. In addition, it receives a floor input indicating the current floor level and the two signals up and down indicating the direction of movement of the elevator. The controller unit receives the requested floor from the request resolver unit and a floor input indicating the current floor level. It generates three control signals as outputs: up, down and open. The signals up and down control the direction of movement of the elevator while the open signal control the elevator door.

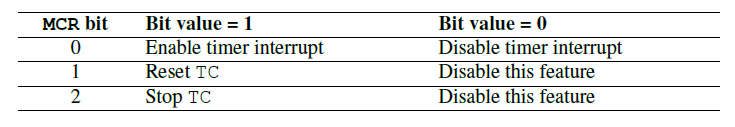


The system will move the elevator either up or down to reach the requested floor. Once at the requested floor, the elevator door is opened for at least 10 seconds, and is kept open until the requested floor changes. It must be ensured that the elevator door is never open while the elevator is moving.

The state machine diagram for the elevator controller unit is given below. In the FSM, the symbols U, D, O, T represent elevator going up, elevator going down, door is open, and start timer, respectively.



Write a C program that implements this state machine of the elevator controller. Assume that any change in the inputs **req** and **floor** will cause interrupts to update their values. You do not need to take care of that. Use Timer0 to implement the required timing requirement. Show the code for TIMER0\_IRQHandler(). To enable the use of Timer0 interrupt use NVIC\_EnableIRQ(1). Note that for TC registers to start counting, the least significant bit of the Timer Control Register (LPC\_TIM0->TCR) must be set. To clear the MR0 interrupt flag, set the least significant bit in the Interrupt Register (LPC\_TIM0->IR). Information on how timer actions using the MCR register are given below:



Timer0 match and MCR registers are LPC\_TIM0->MR0 and LPC\_TIM0->MCR. Assume that the clock frequency is 25 MHZ.