AIN SHAMS UNIVERSITY
FACULTY OF ENGINEERING
ICHEP; Mechatronics Engineering Program

May 8th, 2021 **Course Code: CSE 347** ***Time: 1 Hour***

Mid Term; Embedded System Design

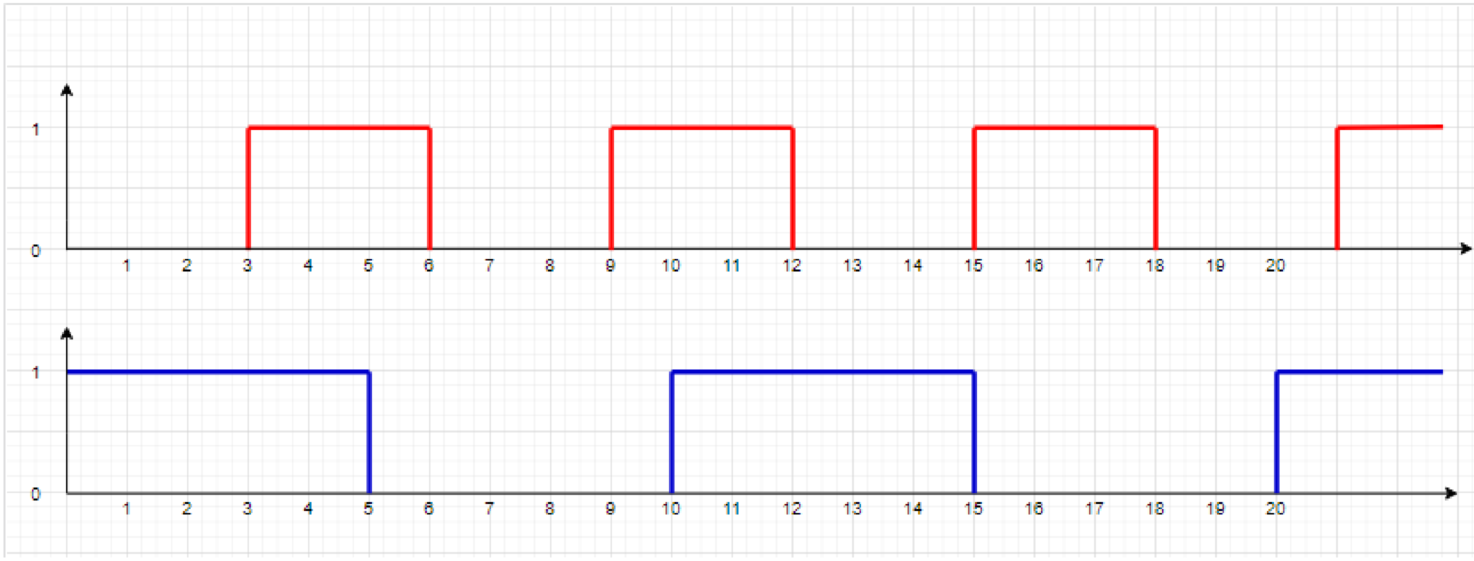
The Exam Consists of 4 Questions in 4 Pages ***Total Marks: 25 Marks***

Question 1: Complete the following (1 Mark Each)

1. In a FreeRTOS program, the vTaskDelay() API when called by a task changes the task status from …Running….. to …Suspended/Blocked…..
2. In a FreeRTOS program the taskYIELD() API when called by a task changes the task status from …….. to …….. **ALL ANSWERS ARE CORRECT – Question CANCELLED**
3. A……TCB….is created and stored per task to preserve its context switching data.
4. When the task is created it set in the……Ready…………..state.
5. When vTaskSuspend() is called for a task, the task is moved to…………Suspended/Blocked…………state.
6. When a context switch occurs from func1 to func2, the information indicating the instruction from which func1 should start executing again once we return to it is stored in…LR….register.
7. The freeRtos is a fixed priority preemptive kernel operating system, preemptive means that…Higher priority tasks are guaranteed to always execute first. OR tasks can preempt each other based on priority..etc.
8. The freeRtos is a fixed priority preemptive kernel operating system, fixed-priority means that Means that the OS does not have the ability to change priorities set by user
9. Two of the events/conditions which cause freeRtos to switch context are………………vTaskDelay/DelayUntil…and……OS Tick
10. Context switching is a set of instructions executed by the CPU and is instructed to it by the……compiler……in case of software function calls and by the……kernel……….in case of concurrent tasks execution.

Question 3: (9 Marks)

The graph below shows the result of drawing the output of PF1 (RED LED) and PF2 (BLUE LED) versus time. The rising edge indicates the led is switched ON, the falling edge indicates that the LED is switched OFF. FreeRTOS is used to create a code that gives the output as shown in the graph.



```
void main(void)
{

    xTaskCreate(.......InitTask........,"firstfunc",1024,NULL,7,&xHandle1 );

    xTaskCreate(..............BlueTask......,"firstfunc",1024,NULL,5,&xHandle2 );

    xTaskCreate(...........RedTask.........,"firstfunc",1024,NULL,2,&xHandle3 );

    vTaskStartScheduler();
}
```

```
void InitTask(void)
{
      // Function that initialize portF so that:
      // PB1 is connected to RED LED
      // PB2 is connected to BLUE LED
      PortF_init();

      // Suspend ourselves.
   vTaskSuspend( NULL );
}


void BlueTask(void)
{
  TickType_t xLastWakeTime;

   // Initialise the xLastWakeTime variable with the current time.
   xLastWakeTime = xTaskGetTickCount();

   //Inside the while loop blink the LED
    // and use the vTaskDelayUntil function
    while(1)
    {

            GPIO_PORTB_DATA_R ^=  0x04
            vTaskDelayUntil(  {any value that could mean 5s or 5 ms} )

    }
}
void RedTask(void)
{
      //Inside the while loop blink the LED
      // and use the vTaskDelay function
      while(1)
      {


              vTaskDelay(  {any value that could mean 3s or 3 ms} )
              GPIO_PORTB_DATA_R ^=  0x02
      }
}
```

Question 4: (6 Marks)

1. Having two tasks, each that executes its functionality within a while(1), what is/are the downgrade(s) of using the Program Counter (PC) to switch the execution of the tasks?

   Hacking the PC leads to having the tasks executing from the beginning of their code and until their time slice ends only, hence the first part only will keep repeating while context is not recalled.

2. State at least two of the CPU scheduling criteria.

   - CPU Utilization
   - Throughput
   - Turnaround time
   - Waiting time
   - Response