



May 8th, 2021

Course Code: CSE 345

Time: 1 Hour

Mid Term; Embedded System Design

The Exam Consists of **4** Questions in **4** Pages

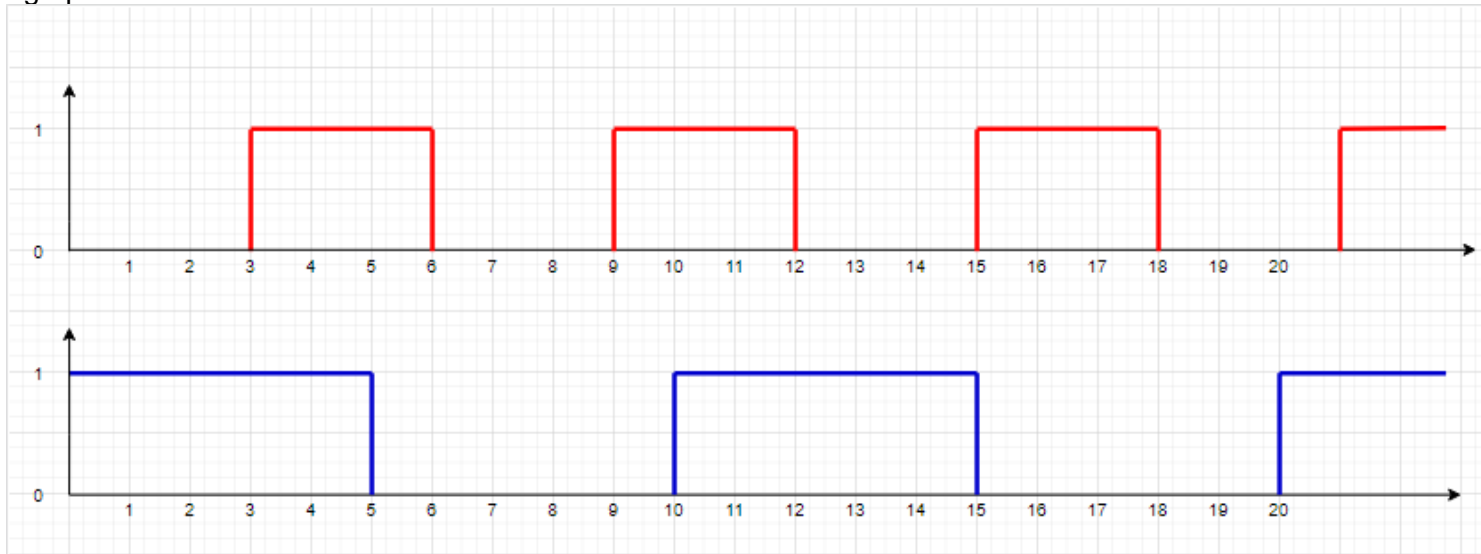
Total Marks: 25 Marks

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 to
2. In a FreeRTOS program the taskYIELD() API when called by a task changes the task status from to
3. A.....is created and stored per task to preserve its context switching data.
4. When the task is created it set in the.....state.
5. When vTaskSuspend() is called for a task, the task is moved to.....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.....register.
7. The freeRtos is a fixed priority preemptive kernel operating system, preemptive means that.....
8. The freeRtos is a fixed priority preemptive kernel operating system, fixed-priority means that.....
9. Two of the events/conditions which cause freeRtos to switch context are.....and.....
10. Context switching is a set of instructions executed by the CPU and is instructed to it by the.....in case of software function calls and by the.....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(....., "firstfunc", 1024, NULL, 7, &xHandle1 );
```

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

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

```
    vTaskStartScheduler();
```

```
}
```

Embedded System DesignThe Exam Consists of **4** Questions in **4** Pages**Total Marks: 25 Marks****3/4**

```
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)
    {

    }
}

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

    }
}
```

