

รหัสสนศ \_\_\_\_\_ ชื่อ-สกุล \_\_\_\_\_

# มหาวิทยาลัยสงขลานครินทร์

## คณะวิศวกรรมศาสตร์

สอบปลายภาค ประจำภาคการศึกษาที่ 1

วันที่ 11 ธันวาคม 2557

วิชา 242-440 Real-Time Operating Systems

ปีการศึกษา 2557

เวลา 1330-1630

ห้อง หัวหูน

- 
- ให้เขียนชื่อสกุล, รหัสสนศ. ในกระดาษคำตอบทุกหน้า (รวมปก)
  - ข้อสอบมีทั้งหมด 6 หน้า รวมปก
  - อนุญาตให้นำกระดาษ A4 เข้าได้ 1 แผ่น
  - ไม่อนุญาตให้นำเครื่องคำนวณและเอกสารอื่นๆเข้าห้องสอบ

ทุจริตในการสอบ โทษขั้นต่ำคือปรับตกในรายวิชาที่ทุจริต และพักการเรียน **2** ภาคการศึกษา

รหัสนศ \_\_\_\_\_ ชื่อ-สกุล \_\_\_\_\_

1. จากบทความบนเว็บไซต์ pandodaily (pando.com) เรื่อง “GM's hit and run: How a lawyer, mechanic, and engineer blew open the worst auto scandal in history.” จงตอบคำถามต่อไปนี้

1.1 สรุปเนื้อหาจากบทความที่ท่านได้อ่านมา (ถ้าเนื้อหาไม่พอให้เขียนต่อด้านหลัง) (10 คะแนน)

รหัสนศ \_\_\_\_\_ ชื่อ-สกุล \_\_\_\_\_

1.2 ปัญหาที่เกิดขึ้นเป็นปัญหาทางด้านซอฟต์แวร์หรือฮาร์ดแวร์ (อธิบาย)

(5 คะแนน)

1.3 จากมุมมองของวิศวกรทางด้านระบบฝังตัว (Embedded Systems) ถ้านำเอาระบบฝังตัวเข้าไปใช้เพื่อป้องกันปัญหาในลักษณะนี้ ในความเห็นของท่านสามารถทำได้อย่างไรบ้าง?

(5 คะแนน)

รหัสนศ \_\_\_\_\_ ชื่อ-สกุล \_\_\_\_\_

2. การพัฒนาซอฟต์แวร์เพื่อใช้ในงานระบบเวลาจริงโดยใช้ FreeRTOS จงเติมคำในช่องว่างให้ถูกต้อง (30 คะแนน)

2.1 ซอฟต์แวร์ FreeRTOS เป็นซอฟต์แวร์ ซึ่งใช้ซอฟต์แวร์ไลเซนส์ (software license) แบบ .....

2.2 ซอฟต์แวร์ FreeRTOS รุ่นล่าสุดคือเวอร์ชัน ..... ซึ่งรองรับ CPU หลายแพลตฟอร์ม แต่ในการทดลองใช้บอร์ดทดลองของบริษัท ETT ซึ่งใช้ CPU LPC2148 ซึ่งผลิตโดยบริษัท ..... ซึ่ง CPU LPC2148 ไม่ได้รองรับโดย FreeRTOS โดยตรง แต่มีรุ่นของ CPU ซึ่งรองรับ ที่ใกล้เคียงที่สุดคือ .....

2.3 CPU รุ่นที่รองรับดังกล่าว มีหน่วยความจำแบบ Flash ขนาด 128KB และมี RAM ขนาด 64KB ในขณะที่ ตัว LPC2148 มี Flash ขนาด 512KB และ RAM ขนาด 32KB + 8KB

ในการปรับแก้ เพื่อให้ FreeRTOS สามารถใช้งานได้ ในส่วนต้นของไฟล์ lpc2148-rom.ld จะต้องแก้เป็น

#### MEMORY

```
{  
    flash    : ORIGIN = 0, LENGTH = .....  
    ram      : ORIGIN = 0x40000000, LENGTH = .....  
}
```

```
__stack_end__ = 0x40000000 + ..... - 4;
```

2.4 สำหรับ Development tools ของ GNU ซึ่งใช้งานบน Linux จะต้องแก้ไข Makefile เพื่อกำหนดตัว คอมไพล์เลอร์ และโปรแกรมที่เกี่ยวข้องดังนี้

CC=.....

OBJCOPY=.....

ARCH=.....

```
USE_THUMB_MODE=YES  
RUN_MODE=RUN_FROM_ROM  
LDSCRIPT=lpc2148-rom.ld
```

2.5 จากไฟล์ FreeRTOSConfig.h ซึ่งกำหนดค่าของ ค่าความถี่สัญญาณ Clock ของ CPU ดังนี้

```
#define configCPU_CLOCK_HZ ( ( unsigned long ) 58982400 )
```

เพื่อให้ใช้กับบอร์ด ETT LPC2148 ซึ่งใช้ ค่าความถี่ของ Crystal เป็น 12MHz และกำหนดให้ใช้ค่าตัวคูณของ PLL เป็น 4 จะต้องกำหนดค่า configCPU\_CLOCK\_HZ เป็น

```
#define configCPU_CLOCK_HZ .....
```

2.6 เมื่อแก้ไขค่า config ทุกอย่างแล้ว เพื่อจะสร้าง binary ไฟล์ซึ่งใช้ในการเขียนลงบน flash ของบอร์ด จะต้องเรียกใช้คำสั่ง ..... และจะได้ output ซึ่งใช้เขียนลงบน flash คือ .....

2.7 ถ้าในโปรแกรมต้องการที่จะใช้พอร์ท UART สำหรับสำหรับการสื่อสารแบบ serial communication จะต้องกำหนดค่า mainCOM\_TEST\_BAUD\_RATE ซึ่งจะอยู่ในไฟล์ ..... โดยที่ถ้าต้องการใช้ความเร็วของการรับ/ส่งข้อมูลเป็น 115200 bps ก็จะต้องกำหนดค่าเป็น

```
#define mainCOM_TEST_BAUD_RATE .....
```

2.8 จากส่วนของโปรแกรมซึ่งใช้ในการควบคุมการกะพริบของ LED ซึ่งต้องการให้ LED ดวงที่ 0 กระพริบด้วยความถี่เป็น 2 เท่าของ LED ดวงที่ 1 และดวงที่ 1 เป็น 2 เท่าของดวงที่ 2 จะต้องกำหนดค่า mainLEDX\_FLASH\_PERIOD เป็นเท่าใด?

```
#define mainLED0_FLASH_PERIOD (( TickType_t ) ..... / portTICK_PERIOD_MS )
```

```
#define mainLED1_FLASH_PERIOD (( TickType_t ) 200 / portTICK_PERIOD_MS )
```

```
#define mainLED2_FLASH_PERIOD (( TickType_t ) ..... / portTICK_PERIOD_MS )
```

2.9 Task ฟังก์ชันต่อไปนี้ใช้สำหรับการตรวจสอบค่า push button switch จงเติมส่วนของ code ที่ขาดหายไป กำหนดค่า parameter อื่นๆ ที่ไม่ได้ระบุไว้ตามความเหมาะสม (2 คะแนน/คำตอบ)

```
static void prvTaskReadButton1( void *pvParameters )
{
    ( void ) ..... ;

    for(.....)
    {
        unsigned long ulState;
        ulState = GPIO0_IOPIN;
        if( ulState & mainBUTTON_BIT1 )
        {
            .....;
        }
        else
        {
            ulButtonStatus = 0;
        }
    }
    ..... ( NULL );
}
```

รหัสนศ \_\_\_\_\_ ชื่อ-สกุล \_\_\_\_\_

2.10 ในส่วนของฟังก์ชัน main() ด้านล่างมีการสร้าง Task โดยใช้ฟังก์ชัน xTaskCreate() สำหรับควบคุมการกะพริบ LED เอาไว้แล้ว ถ้ามีการกำหนด Task สำหรับการตรวจสอบ push button switch และ Task สำหรับการใช้งาน UART ซึ่งมีกำหนดฟังก์ชันโปรโตไทป์ไว้ดังนี้

```
static void prvTaskBlinkOnBoardLED( void *pvParameters );  
static void prvTaskReadButton1( void *pvParameters );  
static void prvTaskUART( void *pvParameters );
```

จงเพิ่มส่วนของโปรแกรมเพื่อเพิ่ม Task ทั้งสองเข้าไปในการทำงานด้วย (5 คะแนน)

```
int main( void )  
{  
    prvSetupHardware();  
    xTaskCreate( prvTaskBlinkOnBoardLED, "LEDP1.24",  
                configMINIMAL_STACK_SIZE, NULL, mainCHECK_TASK_PRIORITY, NULL );  
    .....  
    .....  
    .....  
    .....  
    .....  
    .....  
    vTaskStartScheduler();  
    return 0;  
}
```