

รหัสนศ _____

ชื่อ-สกุล _____

มหาวิทยาลัยสงขลานครินทร์

คณะวิศวกรรมศาสตร์

สอบปลายภาค ประจำภาคการศึกษาที่ 1

ปีการศึกษา 2558

วันที่ 9 ธันวาคม 2558

เวลา 13:30-16:30

วิชา 242-440 Real-Time Operating Systems

ห้อง A400

-
- ให้เขียนชื่อสกุล, รหัสนศ. ในกระดาษคำตอบทุกหน้า (รวมปก)
กระดาษคำตอบหน้าใด ไม่มีชื่อสกุล/รหัสนศ. จะไม่ตรวจให้คะแนนในหน้านั้น
ถ้าไม่เขียนชื่อสกุล/รหัสนศ. ในหน้าปก จะไม่ตรวจให้คะแนนทั้งหมด
 - ข้อสอบมีทั้งหมด 7 หน้า รวมปก มีคะแนนรวมทั้งหมด 53 คะแนน
 - อนุญาตให้นำกระดาษ A4 เข้าได้ 1 แผ่น
 - ไม่อนุญาตให้นำเครื่องคำนวณและเอกสารอื่นๆเข้าห้องสอบ

ทุจริตในการสอบ โทษขั้นต่ำคือปรับตกในรายวิชาที่ทุจริต และพักการเรียน 1 ภาคการศึกษา

รหัสสนศ _____ ชื่อ-สกุล _____

1. ในโครงการ Mars Path Finder ซึ่งส่งยานไปสำรวจดาวอังคารของ NASA เมื่อปี ค.ศ. 1997 หลังจากลงจอดและส่งยานสำรวจ Sojourner ออกไปสำรวจพื้นผิวแล้ว มีปัญหาในการส่งข้อมูลกลับมายังโลก (อนุญาตให้ใช้น้ำกระดาษด้านหลังสำหรับตอบคำถาม ถ้ากระดาษไม่พอ) (23 คะแนน)
- 1.1 จงอธิบายว่า ปัญหาที่เกิดขึ้นมีชื่อเรียกโดยทั่วไปว่าอย่างไร มีลักษณะอย่างไร (เขียน diagram ประกอบด้วย) (10 คะแนน)

รหัสสนศ _____ ชื่อ-สกุล _____

1.2 วิธีการแก้ปัญหาของ NASA ทำอย่างไร (10 คะแนน)

1.3 ระบบปฏิบัติการซึ่งใช้งานบน Mars Path Finder คือ _____ (3 คะแนน)

2. การพัฒนาซอฟต์แวร์เพื่อใช้ในงานระบบเวลาจริงโดยใช้ FreeRTOS จงเติมคำในช่องว่างให้ถูกต้อง

[System and Configuration]

(21 คะแนน – 1 คะแนนต่อคำตอบ)

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 เพื่อกำหนด

ตัวคอมไพเลอร์ 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)

3. การพัฒนาซอฟต์แวร์เพื่อใช้ในงานระบบเวลาจริงโดยใช้ FreeRTOS (9 คะแนน)

3.1 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 );
}
```

3.2 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;  
}
```