

Khoa Điện-Điện Tử	ĐỀ THI HỌC KỲ 2 – 2017/2018
Bộ Môn Điện Tử	MÔN: LẬP TRÌNH NHÚNG EE3031 Thời gian: 90 phút SV được sử dụng tài liệu, không sử dụng máy vi tính (Điểm số tổng cộng tối đa là 11đ)

Câu 1: (3đ)

- a) (1đ) Giải thích sự khác nhau giữa 3 khai báo:
- ```
int (* funcPtr) (int);
(int *) funcPtr (int);
(int *) (* funcPtr) (int);
```
- b) (0.5đ) Giải thích sự khác nhau giữa kiểu signed char và unsigned char.
- c) (0.5đ) Biến count được khai báo như sau: **unsigned short count = 65535;**  
Sau lệnh **count+=16;** giá trị count là bao nhiêu?
- d) (1đ) Viết hàm **int countSpace(char \*str)** đếm số khoảng trắng trong một chuỗi. (Mã ASCII của khoảng trắng là 32).

**Câu 2: (3đ)**

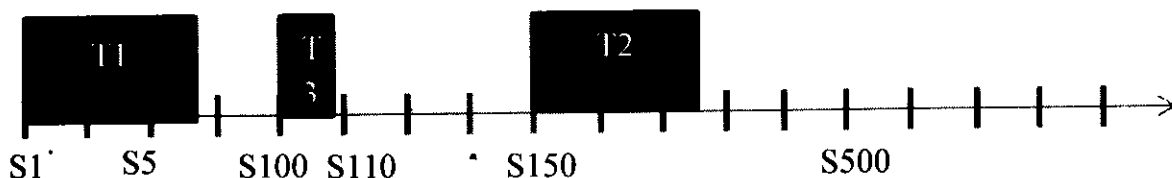
Ba tác vụ có tính chất như sau:

void T1(void): chạy sau mỗi 1000 ms, thời gian chạy trong khoảng từ 5ms đến 100ms.

void T2(void): chạy sau mỗi 400ms, thời gian chạy trong khoảng từ 10 đến 100ms.

void T3(void): chạy sau mỗi 200ms, thời gian chạy trong khoảng từ 1 đến 10ms.

- a) (1đ) Hãy vẽ sơ đồ sắp lịch cho 3 tác vụ (schedule) để đảm bảo cả ba chạy đúng timing. Mỗi SysTick cách nhau 1 ms. **Giải thích.** (Hình sau là ví dụ minh họa của 1 sơ đồ sắp lịch chưa phù hợp, S100 nghĩa là system tick thứ 100).



- b) (0.5đ) Viết hàm void initSysTick(void) khởi động SysTick Timer với chu kỳ 1 ms, cho phép ngắt.
- c) (1.5đ) Viết chương trình thực hiện 3 tác vụ trên với sơ đồ sắp lịch và SysTickTimer của câu a và b. (Chỉ viết chương trình phục vụ ngắt SysTick và vòng lặp chính, đồng thời khai báo các biến toàn cục cần thiết)

*//Khai báo các biến cần thiết ở đây*

```
void SysTickISRHandler()
```

```
{
```

*//write the code here*

```
}

int main(void)
{

 while(1)
 {
 //write the code here }}
}
```

**Câu 4: (2đ)**

- (0.5đ) Giải thích sự khác nhau cơ bản giữa hệ điều hành co-operative (non-preemptive) và pre-emptive.
- (0.5đ) Viết lệnh khởi tạo 1 queue tên là myQueue, gồm 100 phần tử kiểu float.
- (0.5đ) Sau khi thực hiện lệnh **xQueueSendFromISR( xRxQueue, &cIn, &xHigherPriorityTaskWoken );** biến **xHigherPriorityTaskWoken** sẽ có giá trị như thế nào? Giải thích các trường hợp có thể xảy ra. (0.25đ)
- (0.5đ) Để sử dụng được lệnh **xQueueSendFromISR**, mức ưu tiên của ISR mà lệnh này được gọi phải như thế nào?

**Câu 5: (3đ)**

Hãy viết chương trình dùng FreeRTOS thực hiện công việc sau:

Ban đầu tắt tất cả các LED, sau đó làm song song hai tác vụ sau:

- \*) Chớp Led đỏ với thời gian sáng/tắt là 1s
- \*) Nếu SW1 nhấn/nhả 3 lần liên tục, đảo Led xanh dương.

Cho trước các hàm:

void InitHardware(); hàm này khởi động các input/output cần thiết.

void setLed(int Led); hàm này bật Led, tham số Led=0 là Led đỏ, Led=1 là Led xanh dương

void clearLed(int Led); hàm này tắt Led, tham số Led=0 là Led đỏ, Led=1 là Led xanh dương

int readSW1(void); hàm này đọc SW1, bằng 0 nghĩa là SW nhấn, 1 nghĩa là SW nhả.

SW1



BM Điện tử

TS. Trương Quang Vinh

GV ra đề

Trần Hoàng Linh