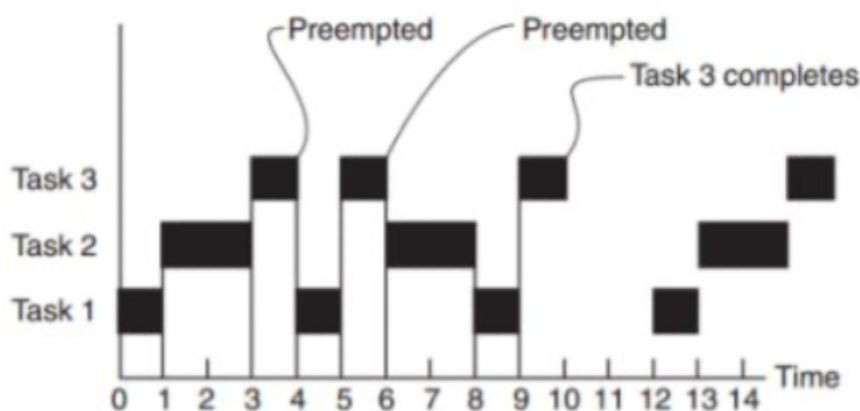




سوالات

- سوال اول) به مثال زیر توجه کنید. مثال زیر نشان دهنده چگونگی انجام سه task مختلف در سیستم عامل RTOS می باشد که توسط الگوریتم Monotonic Rate زمان بندی شده است.

Task	Execution Time	Period	Priority
T1	1	4	High
T2	2	6	Medium
T3	3	12	Low



حال به سوالات زیر پاسخ دهید:

- توضیح دهید که چرا در زمان ۳، تسک شماره ۳ در حال اجرا است. همچنین توضیح دهید که چرا در زمان ۴ اجرای این تسک متوقف می شود.
- بیشترین زمان اجرای T2 چقدر می تواند باشد به طوری که این سه تسک قابل برنامه ریزی باشند؟

- سوال دوم) در ابتدا کتابخانه FreeRTOS Arduino را بر روی IDE Arduino خود نصب کنید. حال از FreeRTOS -> Example -> File مثال BlinkAnalog-Read را انتخاب کنید. حال تابع TaskBlink مثال را به صورت زیر تغییر بدهید.

```
void TaskBlink(void *pvParameters) {

    (void) pvParameters;

    pinMode(LED_BUILTIN, OUTPUT);

    volatile int i = 0;
    for (;;) // A Task shall never return or exit.
    {
        digitalWrite(LED_BUILTIN, HIGH);
        for(i=0;i<30000;i++);
        digitalWrite(LED_BUILTIN, LOW);
        for(i=0;i<30000;i++);
    }
}
```

برنامه را اجرا کرده و به سوالات زیر پاسخ دهید.

- تابع TaskBlink جدید با تابع اولیه چه تفاوتی دارد؟
- متغیر i به چه دلیل volatile تعریف شده است؟
- کد اسمبلی تولید شده برای دو تابع (تابع جدید و قدیم TaskBlink) به چه صورت است؟
- در صورتی که متغیر i را به صورت volatile تعریف نکنیم و برنامه را اجرا کنیم، چه تفاوتی با حالت volatile دارد؟

- **سوال سوم)** با استفاده از قابلیت‌های ساخت Task به کمک کتابخانه Free RTOS در Arduino IDE، وظایف زیر را با استفاده از الگوریتم RM زمان‌بندی کنید. (برای یک HyperPeriod) فرض کنید در ابتدا همه تسک‌ها همزمان با هم می‌رسند. کد شما باید بروز هر گونه نقض موعد را در انتهای اجرا گزارش کند. برای این تمرین می‌توانید از Virtual Terminal در پروتئوس یا ابزارهای دیگر برای نمایش نحوه عملکرد کد استفاده کنید.

Task number	C_i	P_i
t_1	1	4
t_2	2	6
t_3	3	8

- **سوال چهارم)** با استفاده از قابلیت‌های ساخت task به کمک کتابخانه Free RTOS در Arduino IDE، برنامه‌ای بنویسید که دو task با مشخصات زیر را اجرا کند:
task شماره یک، در هر ۱۰۰ میلی ثانیه، ۱ This is Task را چاپ کند. task شماره ۲ هم در هر ۵۰۰ میلی ثانیه، ۲ This is Task را چاپ کند.
برای ایجاد این دو task یک بار اولویت آن‌ها را ۳ و ۱ قرار دهید و بار دیگر اولویت آن‌ها را ۱ و ۳ قرار داده، خروجی‌های بدست آمده را گزارش کرده و توضیح دهید.

- **سوال پنجم)** در سیستم عامل Free RTOS برای ایجاد ارتباط بین دو تسک می‌توان از سمافور (Semaphore) ها و یا صف (queue) استفاده کرد در مورد این دو روش تحقیق کرده و شیوه عملکرد هر کدام را توضیح دهید.
برای هر کدام از سناریوهای زیر کد تسک ۱ و تسک ۲ را نوشته و توضیح دهید که برای هر کدام چرا از صف و یا سمافور استفاده کرده‌اید:

سناریو اول: تسک اول از یک تا صد می‌شمارد و وقتی به ۵۰ رسید، به تسک دوم اطلاع می‌دهد تا شروع به چاپ اعداد از ۲۰۰ تا ۲۵۰ کند و در این حین، اجرای تسک اول متوقف نمی‌شود.

سناریو دوم: متغیر x بین دو تسک مشترک است، تسک اول آن را افزایش می‌دهد تا به ۱۰ برسد و تسک دوم آن را کاهش می‌دهد تا به ۰ برسد. مقدار اولیه متغیر ۵ است. می‌خواهیم که این متغیر بین ۰ و ۱۰ به طور یکنواخت نوسان کند. به بیان دیگر، عملیات افزایش تا ۱۰ به طور پیوسته توسط تسک اول و سپس کاهش به صفر توسط تسک ۲ باید انجام شود.

کد مربوط به این سوال را به گونه‌ای بنویسید که کار انجام شده توسط هر یک از تسک‌ها قابل تفکیک از یکدیگر باشد.

تعدادی از منابع کمکی برای حل این تمرین در ۱ و ۲ و نیز ۳ قابل دسترسی است.

موفق باشید