



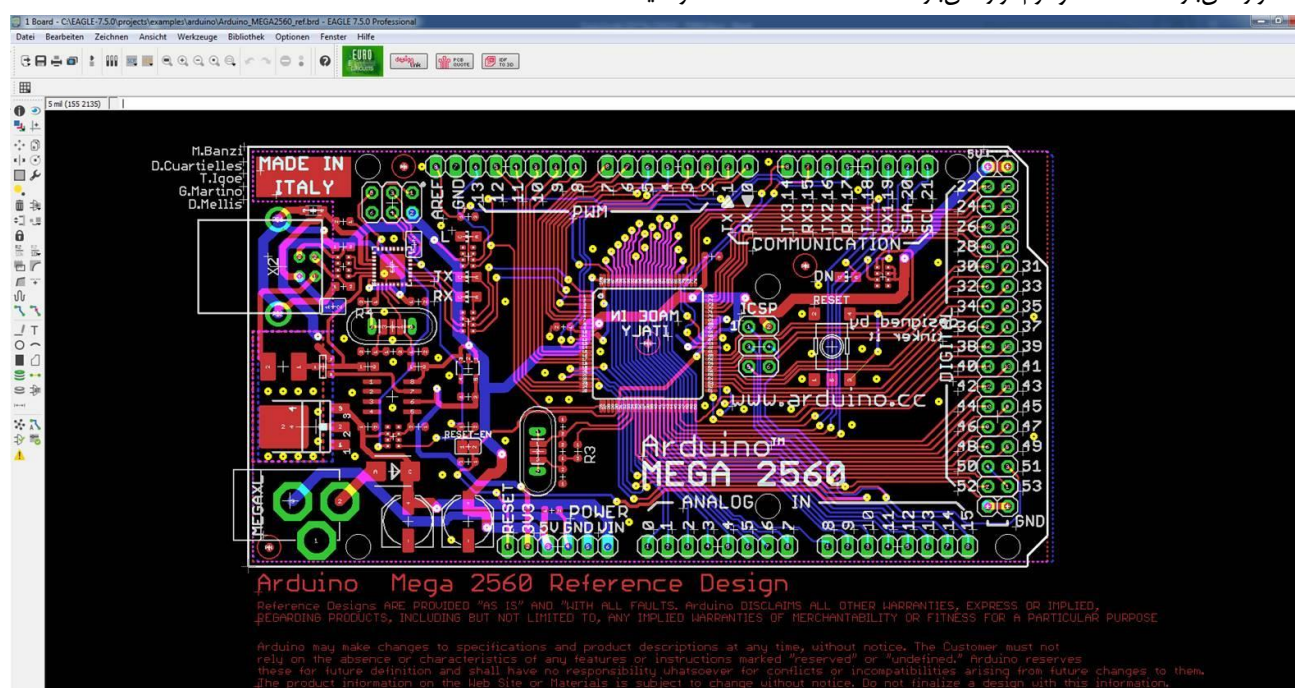
دانشگاه صنعتی امیرکبیر
(پلی تکنیک تهران)

به نام خدا

سیستم‌های نهفته و بی‌درنگ

تمرین چهارم

بسترهای متن‌باز کمک بسیاری به سرعت توسعه نرم‌افزار و سخت‌افزار کرده است. برای مثال در تصویر زیر تصویر مسیرگزینی و جایگذاری شده‌ی بورد الکترونیکی Arduino Mega2560 را می‌بینید. توسعه‌دهندگان می‌توانند با توجه به نیازهای خود در این طراحی تغییر ایجاد کنند. برای استفاده‌های عام‌منظوره، توسعه و طراحی از پایه‌ی قسمت‌هایی که نسخه‌ی متن‌باز دارند و با ایجاد تغییرات جزئی در آن‌ها مناسب استفاده‌ی پروژه می‌شوند مانند اختراع دوباره‌ی چرخ است. علاوه بر این، اشکالات بسترهای متن‌باز اکثراً به دلیل استفاده و توسعه توسط اشخاص و شرکت‌های مختلف به سرعت کشف و رفع می‌شود؛ درحالی‌که ممکن است رفع اشکال یک بستر که به صورت شخصی توسعه یافته است ماه‌ها و حتی سال‌ها زمان ببرد و هزینه‌های بسیاری را به شخص یا شرکت وارد کند. البته برای استفاده‌های خاص مانند پروژه‌های نظامی و صنایع حساس توسعه‌ی از ابتدا بسیار مناسب‌تر و ایمن‌تر است. تاکنون شما از سخت‌افزار متن‌باز Arduino و نرم‌افزار متن‌باز FreeRTOS استفاده کرده‌اید.



در این تمرین شما با ساختار فایل‌های موجود در کتابخانه‌ی FreeRTOS بیشتر آشنا می‌شوید و در ادامه با ایجاد تغییراتی در آن‌ها، زمان‌بند موجود را تغییر می‌دهید. ابتدا به صورت مختصر با فایل‌های موجود در این کتابخانه آشنا می‌شویم. برخی از فایل‌های این کتابخانه به شرح زیر است:

- **FreeRTOSConfig.h**: در این فایل کلیه تنظیمات FreeRTOS موجود است و بر اساس سخت‌افزار و کاربرد آن‌ها را تعیین می‌کنید.
- **task.h** و **tasks.c**: در این دو فایل توابع میانای برنامه‌ی کاربردی ساخت و کنترل کارها، توابع ابزاری، توابع داخلی زمان‌بند و ساختار بلوک کنترلی کارها تعریف شده‌اند. برخی از این توابع که قبلاً هم از آن‌ها استفاده کرده‌اید عبارت‌اند از:

- `xTaskCreate`
- `vTaskDelay`
- `vTaskSuspend`
- `vTaskResume`
- `xTaskGetTickCount`

- `list.h` و `list.c`: توابع مورد نیاز برای اضافه و حذف کردن کارها در ساختمان داده‌ی آرایه‌ی مورد استفاده برای لیست کارها در حالات مختلف (که در تمرین قبلی آشنا شدید) استفاده می‌شود.

دیگر فایل‌های موجود برای سمافورها، صف‌ها، تایمرها، `co-routine` ها و دیگر امکانات بسیار `FreeRTOS` است که از حوصله‌ی این بخش از درس خارج است و نیاز به آزمایشگاه دارد.

نکته‌ی جالب و قابل توجه در برنامه‌نویسی برای سامانه‌های نهفته در بهیمنی کد و کنترل قسمت‌های قابل کامپایل و حتی متغیرهای موجود است. برای مثال به کد زیر که در ساختار `TaskControlBlock_t` موجود است توجه کنید:

```
#if ( portCRITICAL_NESTING_IN_TCB == 1 )
    UBaseType_t uxCriticalNesting; /*< Holds the critical section nesting depth for ports that
do not maintain their own count in the port layer. */
#endif
```

همان‌طور که می‌بینید در صورت فعال بودن قابلیت ناحیه‌های بحرانی تودرتو این متغیر تعریف می‌شود. دلیل این کار مهم بودن کوچک‌ترین فضای حافظه‌ی مورد استفاده در سامانه‌های نهفته است. از طرفی دیگر ایجاد چندین نسخه برای کاربردهای مختلف کاری منطقی نیست. به همین جهت با استفاده از رهنمودهای کامپایلر، ما این مشکلات را برطرف می‌کنیم.

در این تمرین شما باید:

- ۴ کار تعریف کنید که نمود فیزیکی انجام آن‌ها، روشن شدن دیودهای نورانی با رنگ‌های مختلف باشد
- در هر لحظه فقط یک دیود و به مدت ۳ ثانیه روشن خواهد بود
- با توجه به توضیحات بالا و لینک‌های کمکی موجود، در ساختار کتابخانه‌ی `FreeRTOS` با اعمال تغییرات مناسب زمان‌بندهای زیر را پیاده‌سازی کنید:

- `FCFS`: در زمان‌بند کارها با توجه به ترتیب تعریف خود سرویس‌دهی می‌شوند
- `EDF`: در این زمان‌بند با تعریف ضرب‌الاجل در هنگام تعریف کار، کاری که ضرب‌الاجل آن نزدیک‌تر است اولویت بالاتری دارد. پس از انجام کار، ضرب‌الاجل به‌روزرسانی می‌شود.

- اضافه کردن تنظیمات لازم به فایل `FreeRTOSConfig` برای انتخاب بین زمان‌بند پیش‌فرض و زمان‌بندهای پیاده‌سازی شده در این تمرین

لینک‌های زیر می‌توانند کمک بسیاری به انجام این تمرین کنند.

- https://github.com/sachin-ik/EDF_FreeRTOS
- این لینک پیاده‌سازی زمان‌بند جدید برای `FreeRTOS` نسخه‌ی `ARM cortex M3` انجام داده است
- https://freertos.org/Documentation/161204_Mastering_the_FreeRTOS_Real_Time_Kernel-A_Hands-On_Tutorial_Guide.pdf
- این لینک توضیحات کامل برای ساختار `FreeRTOS` را تأمین کرده است و حتی در فصل ۳ با تغییرات در تنظیمات فایل `FreeRTOSConfig` زمان‌بندهای دیگری به وجود آورده است. توجه داشته باشید در نسخه‌ی `Arduino` برخی از این امکانات هنوز وارد نشده است

پ.ن: لطفاً فایل پروژه `Proteus`، پروژه `Arduino`، کتابخانه‌ی تغییر یافته و فایل `hex` به‌دست‌آمده را در قالب یک فایل `zip` با ساختار نام‌گذاری `StudentID.zip` در سامانه بارگذاری کنید.

با آرزوی موفقیت