

بسمه تعالی

پروژه نهایی درس

سیستم‌های طراحی سیستم‌های ریزپردازنده‌ای



کرنومتر دیجیتال با استفاده از ATmega328 ، تایمر و وقفه‌ها

هدف: طراحی و پیاده‌سازی یک کرنومتر دیجیتال با استفاده از میکروکنترلر ATmega328، بهره‌گیری از تایمرها و وقفه‌ها. پروژه شامل شروع، توقف و ریست کرنومتر با استفاده از کلیدهای فشاری و نمایش زمان روی نمایشگر سون سگمنت است. این پیاده‌سازی باید در نرم‌افزار Proteus شبیه‌سازی شده و کد آن در CodeVision AVR نوشته شود.

مواد مورد نیاز:

۱. نرم‌افزار Proteus (برای شبیه‌سازی)
۲. CodeVision AVR (برای برنامه‌نویسی)

وظایف دانشجویان:

۱. درک مسئله:
 - نحوه عملکرد یک کرنومتر را تحلیل کنید.
 - نقش تایمرها و وقفه‌ها در اندازه‌گیری زمان را شناسایی کرده و شرح دهید.

۲. طراحی مدار:

- یک شماتیک در Proteus با استفاده از ATmega328، نمایشگرهای سون سگمنت و کلیدها ایجاد کنید.
- از نمایشگر ۴ رقمی برای دقیقه و ثانیه فرمت (MM:SS) استفاده کنید.
- کلیدهای فشاری را به پین‌های مشخصی برای شروع، توقف و ریست متصل کنید.

۳. وظایف برنامه‌نویسی:

- تایمر ۱ را در حالت CTC برای تولید وقفه‌های ۱ ثانیه‌ای پیکربندی کنید.
- یک ISR (روال سرویس وقفه) برای افزایش زمان بنویسید.
- توابعی برای:
 - شروع و توقف تایمر.
 - ریست زمان به صفر.
 - نمایش زمان روی نمایشگر سون سگمنت.

۴. شبیه‌سازی:

- مدار را در Proteus شبیه‌سازی کنید.
- عملکرد کلیدهای شروع، توقف و ریست را بررسی کنید.

سوالات:

۱. توضیح دهید که تایمر میکروکنترلر ATmega328 چگونه در حالت CTC کار می‌کند.
۲. چرا استفاده از وقفه‌ها برای این کاربرد ضروری است؟
۳. چگونه می‌توان این پروژه را برای نمایش میلی‌ثانیه تغییر داد؟
۴. چه چالش‌هایی در واسط سون سگمنت داشتید و چگونه آن‌ها را حل کردید؟

۱. نقشه مدار در Proteus :

- شماتیک مدار را مطابق توضیحات داده شده در نرم افزار پروتئوس ترسیم نمایید.

۲. ساختار کد:

- تایمر ۱ را برای وقفه ۱ ثانیه‌ای تنظیم کنید.
- توابعی برای به‌روزرسانی نمایشگر و مدیریت فشار کلیدها بنویسید.
- و ...

نمونه کد:

```
#include <mega328.h>
#include <delay.h>

unsigned char seconds = 0, minutes = 0;
bit running = 0;

interrupt [TIM1_COMPA] void timer1_compa_isr() {
    if (running) {
        seconds++;
        if (seconds == 60) {
            seconds = 0;
            minutes++;
        }
    }
    // نمایش زمان را در اینجا به‌روزرسانی کنید
}

void main() {
    // تایمر 1 را مقداردهی اولیه کنید
    // ورودی/خروجی‌ها را برای کلیدها و سون سگمنت تنظیم کنید
    while (1) {
        // وضعیت کلیدها را بررسی کرده و حالت را به‌روزرسانی کنید
    }
}
```

مستندات:

- توضیحات مربوط به هر تابع و ISR در کد را به صورت کامل درج کنید.

معیارهای ارزیابی:

۱. عملکرد صحیح کرنومتر.
۲. استفاده مناسب از تایمرها و وقفه‌ها.
۳. دقت نمایشگر سون سگمنت.
۴. کد تمیز و مستندسازی شده.
۵. شماتیک Proteus واضح و سازمان‌یافته.
۶. ارائه گزارش تمیز و مرتب شامل پاسخ به سوالات بخش‌های فوق، شرح کد نوشته شده و الگوریتم انجام پروژه.

موارد تحویلی:

- فایل ارسالی باید شامل:
 ۱. فایل شبیه‌سازی (Proteus (.pdsprj)
 ۲. فایل کد منبع (c).
 ۳. گزارش کوتاهی که طراحی، کد و چالش‌های پیش‌رو را توضیح دهد.باشد.
 - تمامی موارد فوق را در یک پوشه به نام شماره دانشجویی قرار داده و فایل مورد نظر را فشرده نمایید. فایل فشرده با فرمت zip یا rar تولید شود.
 - برای ارسال فایل فقط از طریق فرم ایجاد شده برای آپلود فایل پروژه در آدرس <https://razeghizade.pudica.ir/microControllerSystemsDesign-1-140304.php> اقدام نمایید.
 - فایل‌های پروژه را نهایتاً تا ساعت ۲۳ روز ۵ شنبه ۱۴۰۳/۱۱/۰۴ ارسال نمایید.
- توجه: به هرگونه ارسال بعد از این تاریخ نمره‌ای تعلق نخواهد گرفت.

نمره تشویقی:

- افزودن ویژگی‌های اضافی، مانند نمایش میلی‌ثانیه یا استفاده از LCD به جای سون سگمنت یا شمارش معکوس کرنومتر علاوه بر پروژه اصلی، نمره تشویقی به همراه خواهد داشت. (ارزش پروژه ۳ نمره بوده و در صورت افزودن بخش‌های تشویقی، از ۳ نمره به ۵ نمره قابل افزایش خواهد بود)