

به نام خدا

پروژه نهایی درس معماری کامپیوتر

زمان تحویل پروژه بعدا اعلام خواهد شد

در محیط نرم افزار Keil uVision کدی به زبان اسمبلی برای میکرو کنترلر AT91SAM7X256 کمپانی Atmel بنویسید که پس از ریست شدن، میکروکنترلر الگوریتم کنترل چراغ راهنمایی را با قابلیت‌های کنترل اتوماتیک با زمان بندی پیش فرض و کنترل دستی به شرح زیر انجام دهد:

پایه‌های PA0, PA1 و PA2 میکروکنترلر به ترتیب به عنوان پایه خروجی برای روشن یا خاموش کردن چراغ‌های سبز، زرد و قرمز به کار می‌روند. با یک بودن هر یک از پایه‌ها چراغ نظیر روشن و با صفر بودن آن، چراغ خاموش می‌باشد. زمانبندی اتوماتیک و پیش فرضی چراغ‌ها ۱۰ ثانیه، ۳ ثانیه و ۳۰ ثانیه به ترتیب برای چراغ‌های سبز، زرد و قرمز می‌باشد.

یک کلید کنترل کننده به پایه PA3 میکروکنترلر متصل است. افسر راهنمایی با یک نگهداشتن این کلید، کنترلر چراغ را از حالت اتوماتیک به حالت کنترل دستی تغییر می‌دهد. در حالت کنترل دستی، چراغ با یک کلید متصل به پایه PA4 میکروکنترلر کنترل می‌شود. اگر این سیگنال صفر باشد چراغ سبز و اگر این سیگنال یک باشد، چراغ راهنمایی می‌بایست قرمز باشد. توجه کنید در گذار از سبز به قرمز، چراغ زرد می‌بایست به مدت ۳ ثانیه روشن شود. با صفر شدن کلید متصل به پایه PA3 کنترلر چراغ به حالت اتوماتیک تغییر می‌یابد. اگر هنگام سبز بودن چراغ کلید متصل به PA3 صفر شود، چراغ می‌بایست بلافاصله قرمز شده (پس از ۳ ثانیه زرد شدن) و با زمان بندی اتوماتیک شروع به کار نماید. اگر هنگام قرمز بودن چراغ، سیگنال اعمالی به PA3 صفر شود، چراغ می‌بایست با زمان بندی اتوماتیک از ابتدا زمان قرمز بودن را طی کند. اگر در مدت زمان زرد بودن چراغ، این سیگنال صفر شود، پس از طی شدن مدت زمان ۳ ثانیه چراغ زرد، چراغ قرمز با زمان بندی اتوماتیک شروع به کار می‌کند.

هنگامی که چراغ راهنمایی سبز می‌باشد، افسر راهنمایی این امکان را دارد که مقادیر زمان بندی اتوماتیک چراغ را تغییر دهد. بدین صورت که اگر هنگام سبز بودن چراغ کلید متصل به پایه PA5 یک باشد، دو مقدار ۶ بیتی با ضریب ۲ ثانیه می‌بایست از پایه‌های PA10...PA15 (PA15 بیت پر ارزش می‌باشد) برای مدت زمان چراغ سبز و PA20...PA25 (PA25 بیت پر ارزش می‌باشد) برای مدت زمان چراغ قرمز بارگذاری شود و با شروع از چراغ قرمز بعدی، از مقادیر جدید برای زمان بندی اتوماتیک استفاده گردد. اگر مقدار زمان‌های وارد شده کوچکتر از ۶ ثانیه باشند (پس از ضرب در ۲)، مقدار ۶ ثانیه را جایگزین آن‌ها نمایند.

برای ایجاد زمان بندی از تایمر Real-Time Timer (RTT) میکروکنترلر AT91SAM7X256 با $RTPRES = 0X8000$ برای ایجاد یک تایمر ۳۲ بیتی با کلاک ۱ هرتز استفاده نمایید. با این کلاک سرریز شدن تایمر حدود ۱۳۶ سال طول می‌کشد.

فایل Datasheet میکروکنترلر AT91SAM7X256 و فایل راهنمای پروژه در آدرس زیر قابل دسترسی می‌باشد:

..\\Sepehr\\faculty\\Dr.Raie

برای پیکربندی پایه‌های پورت A میکروکنترلر به صورت ورودی و خروجی و خواندن و نوشتن در آن‌ها از رجیسترهای زیر می‌بایست استفاده نمایید:

1. رجیستر ۳۲ بیتی PIO_PER (در آدرس 0XFFFFFF400): یک کردن بیت متناظر با هر پایه = تعریف پایه به عنوان پایه ورودی/خروجی (صفحه ۲۴۰ کاتالوگ)
2. رجیستر ۳۲ بیتی PIO_OER (در آدرس 0XFFFFFF410): یک کردن بیت متناظر با هر پایه = تعریف پایه به عنوان پایه خروجی (صفحه ۲۴۱ کاتالوگ). پس از ریست تمامی پایه‌ها در حالت ورودی قرار دارند.

3. برای خواندن پورت A میکروکنترلر می‌بایست رجیستر ۳۲ بیتی PIO_PDSR (در آدرس 0xFFFFF43C) را بخوانید. صفر/یک بودن هر بیت = صفر/یک بودن پایه متناظر (صفحه ۲۴۶ کاتالوگ). توجه کنید که فارغ از ورودی یا خروجی بودن پایه، همواره می‌توانید مقدار لحظه‌ای آن را با استفاده از رجیستر فوق بخوانید.

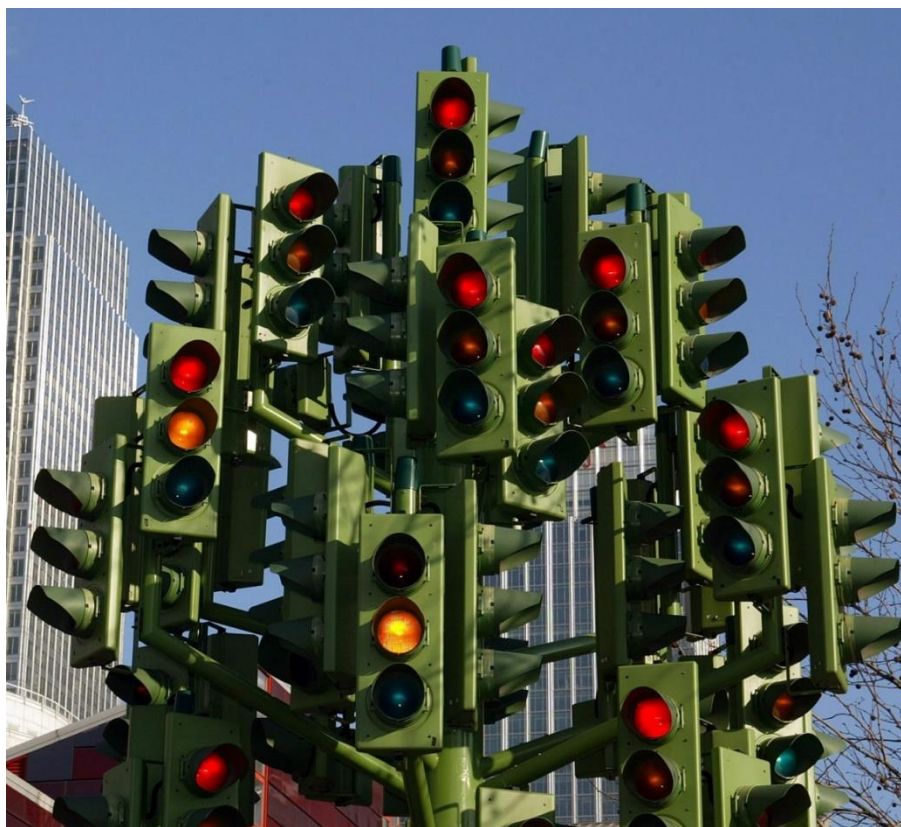
4. برای یک کردن پایه‌های خروجی می‌بایست بیت متناظر با پایه را در رجیستر PIO_SODR (در آدرس 0xFFFFF430) یک نمایید (صفحه ۲۴۴ کاتالوگ). برای صفر کردن پایه‌های خروجی می‌بایست بیت متناظر با پایه را در رجیستر PIO_CODR (با آدرس 0xFFFFF434) یک نمایید (صفحه ۲۴۵ کاتالوگ).

برای استفاده از Real-Time Timer:

1. در رجیستر RTT_MR (در آدرس 0xFFFFFD20) می‌بایست مقدار 0X00048000 را بنویسید.

توجه کنید که به محض نوشته شدن این رجیستر، تایمر از مقدار اولیه صفر شروع به کار می‌کند (صفحه ۷۶ کاتالوگ).

2. مقدار ۳۲ بیتی تایمر در هر لحظه در رجیستر RTT_VR (در آدرس 0xFFFFFD28) قابل دسترس می‌باشد (صفحه ۷۷ کاتالوگ).



😊 موفق باشید