دستور کار آزمایشگاه ریز پردازنده مبتنی بر شبیه سازی

نیم سال اول ۱۳۹۹

محمود نادران طحان، استادیار گروه مهندسی کامپیوتر

تیرداد صدری نژاد، کارشناسی ارشد گروه مهندسی کامپیوتر

دانشگاه شهید چمران اهواز

مقدمه

با توجه به برگزاری کلاس‌های درس به صورت مجازی به خاطر شیوع بیماری کرونا، به کمک نرم افزارهای شبیه سازی میکروکنترلر، تعدادی آزمایش در قالب شبیه سازی تعریف شده است. در هر شبیه سازی مطالب کمکی جهت اطلاع بیشتر ارائه شده است. بنابراین لازم است ضمن مشاهده ویدیو یا نوشتار مربوطه، تمرین مورد نظر را انجام دهید. در این درس از دو نرم افزار Proteus و Keil استفاده می‌شود که لینک دانلود آن‌ها را در جدول زیر مشاهده می‌کنید.

نرم افزار	لینک دانلود
Proteus	http://mnscu.parsaspace.com/sw/proteus
Keil	http://mnscu.parsaspace.com/sw/keil

همچنین فایل‌های زیر جهت آشنایی و یادآوری با میکروکنترلرهای LPC را مطالعه کنید.

- فایل پاورپوینت آشنایی با میکروکنترلر
- فایل یادآوری برنامه نویسی LPC

در این درس علاوه بر تمرین‌هایی که باید انجام شوند، یک پروژه نیز تعریف شده است که طبیعتاً انجام آن مستلزم وقت صرف بیشتری است. انجام تمرین‌ها می‌تواند به صورت تک نفره یا گروه‌های دو نفره باشد اما پروژه باید به صورت انفرادی انجام شود. در خصوص تمرین‌هایی که به صورت گروهی انجام می‌شوند لازم است به این نکته توجه شود که تمرین‌ها باید به صورت گردشی توسط اعضای گروه آپلود شوند. این که همیشه یک نفر از گروه تمرین‌ها را انجام دهد به معنی آن خواهد بود که عضو دیگر گروه فعالیت مستمر ندارد. جدول زیر بارم نمرات این درس را نشان می‌دهد.

فعالیت	نمره
شبیه سازی ۱	۱
شبیه سازی ۲	۱
شبیه سازی ۳	۲
شبیه سازی ۴	۲
شبیه سازی ۵	۳
شبیه سازی ۶	۳
شبیه سازی ۷	۳
پروژه	۲ تا ۳ نمره بسته به کیفیت کار
فعالیت مستمر (شرکت در کلاس‌های آنلاین، ارتباط با دستیار آزمایشگاه و تحویل به موقع تمرین‌ها)	۳

شبیه سازی ۱: چشمک زن ساده

در این تمرین، هدف آن است که چند LED را که به میکرو متصل هستند، به یک ترتیب خاص روشن خاموش کنید. کارهای زیر را انجام دهید:

۱) سه پایه به عنوان خروجی تعریف کنید و در شبیه سازی، به آنها LED با رعایت مثبت و منفی وصل کنید.

۲) این سه پایه را به ترتیب روشن کنید. در شبیه سازی به این شکل عمل کنید که نخست همه خاموش، بعد LED اول روشن، بعد LED دوم (اولی همچنان روشن است) و ...

۳) بین گام‌های روشن شدن به ترتیب، کمی وقفه بگذارید تا قابل دیدن در شبیه سازی باشد.

به نکات زیر توجه کنید:

- LEDهای استفاده شده باید جزء کتابخانه ACTIVE باشند.
- با کلیک راست کردن روی LEDها وارد بخش properties یا تنظیمات آنها شوید و در بخش model type، به جای مقدار پیش فرض analog، مقدار digital را انتخاب کنید. اینکار تاثیر بسیار زیادی روی سرعت شبیه سازی خواهد داشت.

شبیه سازی ۲: راه اندازی UART

ابتدا فایل «یادآوری ۲» و «واحد ارتباط سریال UART» را مطالعه کنید. سپس کارهای زیر را انجام دهید:

۱) پس از آشنایی با پروتکل UART با کمک راهنماهای داده شده، کتابخانه‌ی از پیش تهیه شده‌ی UART را دانلود کنید و در پروژه وارد کنید.

۲) فایل هدر (.h) را باز کنید و توضیحات آن را بخوانید تا ببینید چطور باید از توابع آن استفاده کنید.

۳) در شبیه ساز، قطعه‌ی terminal را وارد کنید و به میکروکنترلر وصل کنید. توضیحات بیشتر را در ویدئو ببینید.

۴) یک برنامه ساده بنویسید که یک متن دلخواه را روی ترمینال نشان دهد تا درست بودن کارهای انجام داده شده بررسی شود.

شبیه سازی ۳: GPIO و UART

در این تمرین، مباحثی که تا کنون فرا گرفتید به صورت ترکیبی استفاده می‌شوند. دو تمرین از تمرین های زیر را به دلخواه انتخاب کنید و انجام دهید. انجام تمرین اضافه، نمره اضافه خواهد داشت.

۱) با کمک دو کلید، برنامه‌ای بنویسید که سرعت چشمک زدن یک LED را تغییر دهد. این کار با تغییر مقدار delay با کمک کلیدها امکان پذیر است. با استفاده از UART روی ترمینال نمایش دهید که مقدار تاخیر بین دو چشمک چه مقداری است.

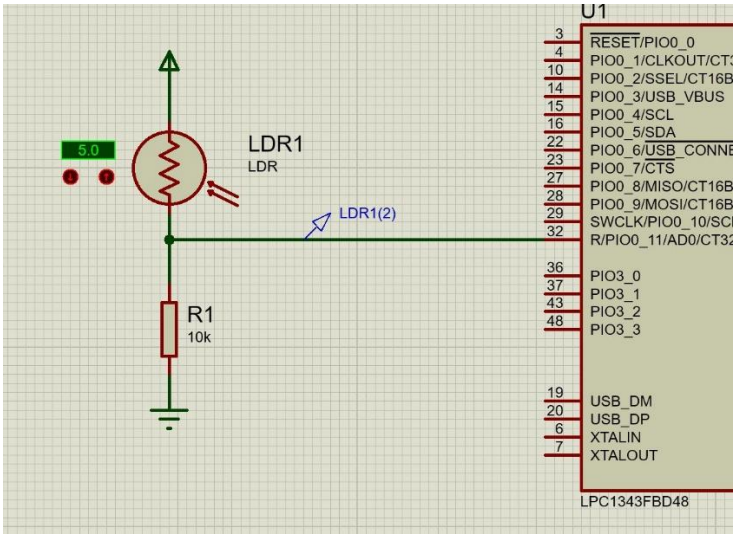
۲) دو دکمه و سه LED به همراه UART استفاده کنید. با زدن یکی از دکمه‌ها، LEDها به ترتیب روشن شوند (هر لحظه یکی) و با زدن دکمه‌ی دوم، این کار متوقف شود. وضعیت با زدن هر دکمه روی ترمینال نوشته شود.

۳) با کمک چند کلید و UART یک قفل ساده درست کنید که فقط با زدن ترکیب خاصی از کلیدها پیام موفقیت آمیز بودن نشان دهد. معمولاً روند کار به این شکل است که یک عدد به عنوان هدف انتخاب می‌شود. مثلاً عدد ۱۲و دو دکمه مقدار یک متغییر را کم و زیاد می‌کند. هرگاه مقدار ۱۲شد، آنگاه قفل باز می‌شود. یکی از دکمه‌ها ۵+ و دکمه‌ی دوم ۳- مقدار متغییر را تغییر

می‌دهند. در نتیجه برای به دست آوردن عدد ۱۲، باید دکمه‌ی اول را ۳ بار فشار داد (۱۵) و سپس یک بار دکمه‌ی دوم (۱۲=۳-۱۵) را فشار داد.

شبیه سازی ۴: کار با مبدل آنالوگ به دیجیتال (ADC)

ابتدا «راهنمای ویدیویی ADC» را ببینید و دو قطعه به شکل زیر قرار دهید: یکی از قطعات LDR است و دیگری یک مقاومت ساده. LDR یک مقاومت حساس به نور است که مقاومت آن با نور تغییر می‌کند. برای سنجش، نیاز به یک مقاومت دیگر به شکل زیر هم هست. هر دو قطعه با اسم LDR و RES قابل اضافه کردن به پروژه هستند.



با کلیک کردن فلش‌های بالا و پایین در کنار LDR می‌توان شدت نور ورودی را هم پیش از شروع شبیه سازی و هم در هنگام شبیه سازی به صورت زنده کنترل کرد. عدد نشان داده کنار آن، شدت نور مجازی تابیده شده به آن است (بدیهی است در یک شبیه سازی رایانه‌ای نور واقعی نداریم). واحد شدت نور، lux است و عدد ۵ نشان داده شده، بر این واحد است.

در ادامه کارهای زیر را انجام دهید:

۱) کدی بنویسید که رنج LUX (نور ورودی) را بخواند و با سه عبارت light، 2tone و dark روی ترمینال نشان دهد.

- اگر رنج نور ورودی بین ۰ تا ۱۰ بود کلمه dark را بنویسد.
- اگر بین ۱۰ تا ۳۰ بود کلمه 2tone را بنویسد.
- و اگر بیش از ۳۰ بود light را روی ترمینال نمایش دهد.

۲) با کمک خطایابی خطی، کدی بنویسید که با خطای ناچیز، مقدار lux ورودی را بنویسد. برای اینکار نیاز به کالیبره کردن به صورت خطی دارید. روش کار بدین صورت است که شما عدد خوانده در ADC را باید به صورت X در یک معادل خط ساده Y=aX+b در نظر بگیرید که باید Y درست (یعنی مقدار LUX داده شده توسط فلش های کنار LDR) را دهد. بنابراین شما نیاز دارید که a و b را به دست آورید. برای رسم یک خط، کافیست دو نقطه از آن را داشته باشیم و بعد از آن شیب و عرض از مبدا، یعنی a و b به دست می‌آیند. پس کافیست با دو بار امتحان شبیه سازی با دو ورودی مختلف، نقطه (X1,Y1) و (X2,Y2) را به دست آورید و سپس روی کاغذ a,b را به دست آورید. مثلاً یک بار ورودی را ۱۰ بدهید و ADC را بخوانید و بار دیگر ورودی را ۲۰ بدهید و ADC را بخوانید. ورودی‌ها Y هستند و مقدار ADCهمان X. در انتها رابطه مثلا به شکل Y=3X+5 خواهد شد که می‌توانید آن را به طور کلی را برای همه مقادیر ADC به کار ببرید. در

نتیجه، با تغییر LUX در ورودی، روی ترمینال عددی نشان داده می‌شود که تا حدود خوبی شبیه همان عدد ورودی است. ترجیحاً اسکرین شات از محیط شبیه سازی که هم خروجی ترمینال و هم عدد داده شده مشخص باشد در فایل زیپ برای تحویل تمرین بفرستید.

شبیه سازی ۵: مالتی پلکس کردن کلیدها

ابتدا راهنمای ویدیویی «مالتی پلکس کردن کلیدها» را ببینید و ابزارهای لازم را به پروژه پروتئوس اضافه کنید. برای اضافه کردن کیپد، واژه‌ی KEYPAD را در لیست قطعات جستجو کنید. اسم قطعه KEYPAD-PHONE است.

این تمرین شامل نوشتن یک تابع و استفاده از آن در حلقه while است که با کمک آن بتوانیم کلید فشار داده شده روی صفحه کلید را بخوانیم و مقدار آن را روی سریال نمایش دهیم. اینکه از کدام پایه‌های میکروکنترلر برای چه کاری استفاده کنید، به دلخواه شماست. اگر می‌خواهید درباره مالتی پلکس بیشتر بدانید، باید در اینترنت جستجو کنید. این روش بسیار معروف است و منابع زیادی در مورد آن خواهید یافت. نوشتن شکل تابع، مقادیر بازگشتی آن و استفاده از آن، به دلخواه شماست.

تابعی بنویسید که در هر بار فراخوانی، دکمه‌ی فشار داده شده در لحظه‌ی فراخوانی را برگرداند. از آنجا که تابع، فقط در لحظه‌ی فراخوانی از فشردن یک کلید مطلع می‌شود، بدیهی است که این تابع باید در حلقه‌ی اصلی برنامه قرار بگیرد تا هر موقع کاربر دکمه‌ای فشار داد، متوجه بشود.

شبیه سازی ۶: راه اندازی Stepper motor

بعد از مشاهده ویدیوی «موتورهای استپر»، با استفاده از نرم افزار شبیه سازی، استپر موتور را در حالت‌های زیر راه اندازی کنید. برای اضافه کردن قطعه مورد نظر در شبیه ساز، از منوی اضافه کردن قطعات stepper را جستجو کنید و از مدلی که در کتابخانه active دارد استفاده کنید. توجه داشته باشید که ترتیب وصل کردن پایه های استپر مهم است.

۱) کارهای زیر را انجام دهید:

- استپر موتور را در حالت عادی چرخش راه اندازی کنید. اینکار با فعال کردن کویل‌ها (در ویدئو توضیح داده شده است) به ترتیب مشخص انجام می‌شود.
- با تغییر فاصله‌ی زمانی بین فعال سازی هر کویل، سرعت چرخش آن را کنترل کنید.
- دو دکمه برای افزایش یا کاهش سرعت آن در نظر بگیرید.
- با استفاده از UART یا نمایشگر، مقدار تاخیر را نمایش دهید.

۲) کارهای زیر را انجام دهید:

- استپر موتور را در حالت درجه راه اندازی کنید. اینکار بسیار شبیه ساعت عقربه‌ای معمولی است.
- با کمک تاخیرهای زمانی، حرکت یک ساعت عقربه‌ای را با آن شبیه سازی کنید. نیازی نیست زاویه ها مانند یک ساعت دقیق باشند.

۳) در ادامه می‌خواهیم موتور DC معمولی با دو روش یک جهته و H-Bridge را راه اندازی کنیم. منظور از درایو یا راه اندازی، عمل تقویت جریان یا ولتاژ کم یک سیگنال (مثلا خروجی میکروکنترلر) است که آن را مناسب راه اندازی بارهای بزرگ مانند موتورها می‌کند. با استفاده از شبیه سازی، قطعه L298 را اضافه کنید و طبق دستورالعمل راه اندازی و کارهای زیر را انجام دهید:

- سه دکمه برای کنترل چپ در نظر بگیرید. یکی چرخش ساعتگرد، یکی چرخش پادساعتگرد و دیگری روشن و خاموش

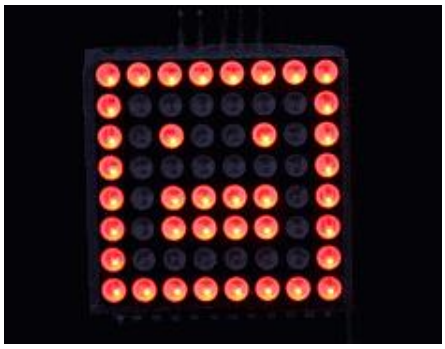
- هر سه عمل کلیدها را انجام دهید. برای تغییر جهت چرخش، یک وقفه‌ی کوتاه در نظر بگیرید. همانطور که هنگامی که با خودرو به جلو می‌روید و می‌خواهید دنده عقب بگیرید، ابتدا ترمز می‌کنید و می‌ایستید، بعد دنده عقب می‌یرید، برای موتور هم همین کار را انجام دهید.
- دکمه‌ی روشن و خاموش، صرف نظر از هر جهت، موتور را تغییر وضعیت می‌دهد. با یک بار فشردن آن، موتور خاموش و بار دیگر روشن شود.

شبیه سازی ۷: راه اندازی Seven Segment و Dot Matrix با میکروکنترلر

ابتدا با ساختار هر دو قطعه آشنا شوید. توجه داشته باشید که این دو بسیار شبیه هم هستند و فقط در نوع چینش LEDها متفاوت هستند. همچنین لازم است با ساختار مالتی پلکس که برای صفحه کلیدها هم استفاده می‌شود آشنا باشید.

۱) در قسمت اول هدف نمایش یک تصویر ۸*۸ پیکسل (به اندازه LEDهای دات ماتریس) روی دات ماتریس است. کارهای زیر را نجام دهید:

- یک شکل دلخواه، در قالب ۸*۸ طراحی کنید.



- با کمک روش توضیح داده شده، به هر سطر یک بایت اختصاص دهید (هر بایت ۸ بیت است و هر بیت متناظر یک چراغ در هر سطر خواهد بود)
- با یک حلقه، سطرها را به ترتیب روی نمایشگر نشان دهید تا شکل دلخواه روی نمایشگر به وجود آید.

۲) در قسمت دوم از سون سگمنت استفاده خواهید کرد. عملیاتی مشابه با دات ماتریس انجام می‌شود با این تفاوت که دیگر شکل دلخواه نیست. بلکه آرایه‌ای خاص تهیه می‌شود و در اختیار شما قرار داده می‌شود که با اعمال هر یک از آن‌ها روی سون سگمنت، عدد مورد نظر نقش می‌بندد. مثلاً با اعمال عنصر ۰ آرایه، عدد صفر، عنصر ۱، عدد ۱ و ...

- با کمک آن، یک شمارشگر ساده بسازید به طوری که در هر واحد زمانی، یک واحد به نمایشگر اضافه کند.
- یک کتابخانه برای سون سگمنت بنویسید که شامل حداقل دو تابع باشد. تابع نخست، همان تابع اصلی به روزرسانی مالتی پلکس است که ویژگی مشترک کیپد، دات ماتریس و سون سگمنت است و در حلقه‌ی اصلی برنامه نوشته می‌شود و تابع دیگر، صرفاً تابع مقدار دهی است. یعنی با یک تابع ساده، مقدار روی نمایشگر تغییر کند. این دو تابع به شکل زیر معرفی می‌شوند:

```
//loop function
Void SegmentLooper();
//set 7Segment number
Void SegmentNumber(uint16_t number);
```

پروژه

در این قسمت، سه تمرین به عنوان پروژه تعریف شده است که یکی از آن‌ها را به دلخواه انجام دهید. توجه کنید که لزوماً همه مطالب خواسته شده، در درس‌ها گفته نشده است و ممکن است لازم باشد برای انجام آن، در اینترنت جستجو کنید یا با یک واسط خاص آشنا شوید. بدیهی است که تمرین‌های سخت تر، نمره اضافه بیشتری دارند. کد تمیز و نوشتن کد در سورس‌های مختلف و استاندارد، نمره اضافه دارد.

۱) در این تمرین با کمک سنسور دمای آنالوگ LM35 (با همین نام در شبیه ساز موجود است) و یک موتور، یک سیستم تهویه ساده راه اندازی کنید. به این ترتیب که:

- یک نمایشگر (ترمینال سریال) در نظر بگیرید که یک دمای آستانه را نمایش می‌دهد.
- دو دکمه برای افزایش یا کاهش دمای دلخواه آستانه قرار دهید.
- دستگاه وقتی دمای سنسور، از دمای آستانه بیشتر شد، موتور را روشن کند (به معنی روشن شدن فن) و اگر کمتر شد، خاموش کند.

اهمیت و کاربرد این تمرین: بسیاری از سیستم‌های تهویه، از فن رایانه گرفته تا کولر، یخچال، دستگاه‌های جوجه کشی و غیره دقیقاً به همین روش کار می‌کنند و این تمرین، کاربردی واقعی دارد.

۲) در این تمرین هدف راه اندازی یک قفل پیشرفته با استفاده از سون سگمنت و کیپد است. قفل باید دارای امکاناتی مانند تغییر رمز پیش‌فرض باشد. در این تمرین فرضیات به عهده شماست و البته باید توجه داشته باشید که فرضیات باید واقعی باشند.

۳) در این تمرین، هدف آشنایی با واحد سریال SPI است. این واسط، یکی از سریع‌ترین پروتکل‌های ارتباط سریال را دارد. در رایانه‌ها و تلفن‌های همراه، بسیاری از قطعات مانند SD card، نمایشگرهای کوچک، سنسورهای تشخیص لمس و غیره از طریق این ارتباط به میکروکنترلر وصل می‌شوند.

- مقداری در مورد ارتباط SPI توضیح دهید. این واسط عمومی است و مختص مدل خاصی از میکروکنترلر نیست.
- بخش SPI در میکروکنترلر LPC1343 را راه اندازی کنید و یک ارتباط ساده تشکل دهید. می‌توانید به جای طرف دوم ارتباط که یک طرف میکروکنترلر است، از SPI ترمینال استفاده کنید که دقیقاً معادل ترمینال UART است که برای تمرین‌های قبل به کار می‌بردیم.