**آزمایشگاه ریز پردازنده آزمایش شماره ی 1**

**نام و نام** **خانوادگی**: هلیا سادات هاشمی پور **شماره** **دانشجویی**:9831106

**نام** **استاد**:استاد علیزاده **تاریخ**:14 مهر 1400

**هدف آزمایش:**

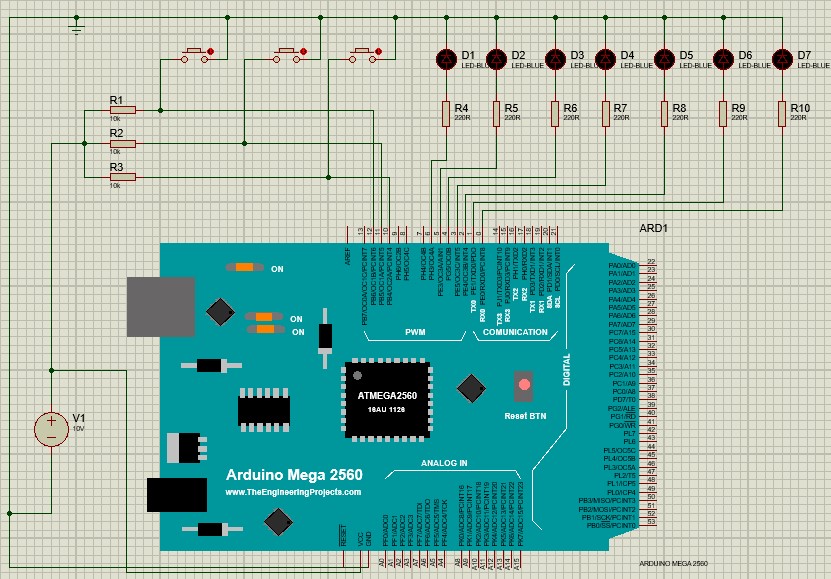
PIO آشنایی واحد ●

برای مدیریت واحد های جانبی (Interrupt-Driven) و وقفه محور (Polling) آشنایی با روش های سرکشی ●

مقایسه دو روش سرکشی و وقفه محور

1. -**برنامه آزمایش را به روش سرکشی بنویسید و پس از آن که از درستی کارکرد مدار و برنامه خود مطمئن شدید گام دوم رو انجام دهید.**

ابتدا یک پروژه ی جدید در محیط proteus می سازیم. با توجه به دستور کار یک برد آردینو  ATmega 2560 ، چند مقاومت به اندازه های ۲۲۰ اهم و ۱۰ کیلو اهم، یکVSOURCE  ۱۰ ولتی، سه کلید ، یک GROUND و ۷ لامپ LED-BLUE مطابق با شکل2.1 دستورکار در پروتئوس پیاده سازی می کنیم. سر مثبت منبع ولتاژ  را به پایه ی VCC برد و سرمنفی آن را به پایه ی GND وصل می کنیم.مدار پیاده سازی شده به صورت زیر است.



مقاومت های R1و R2و R3را به صورت Pull-Up بسته ایم. یک پروژه ی جدید در آردوینو می سازیم  و در بخش Tools برد

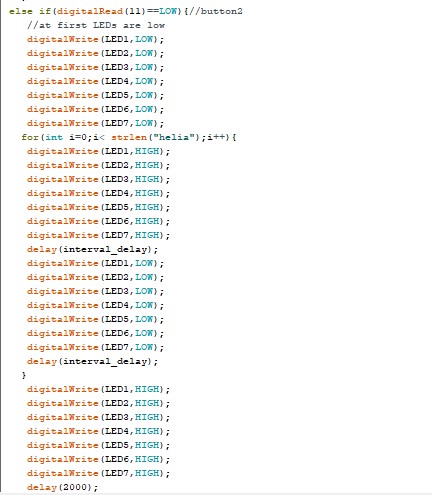
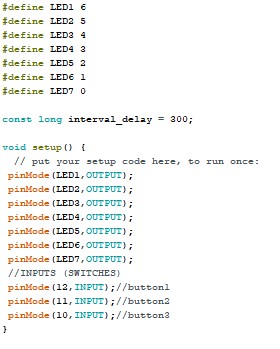
ATmega2560را انتخاب می کنیم. سپس باتوجه به خواسته ی سوال،کد مورد نظر را پیاده سازی می کنیم.پس از اجرا کردن مسیر فایلی  با پسوند .hex  را کپی می کنیم سپس در نرم افزار پروتئوس بر روی برد ATmega 2560 کلیک کرده و در بخش PROGRAM FILE ، paste میکنیم.سپس برنامه خود را ران می کنیم.

**توضیح درمورد کد آردوینو:**

کد نوشته شده در فضای آردوینو به این شکل است که ابتدا LEDهای مورد نظر را براساس شماره LEDهای موجود در برنامه ی پروتئوس Define می کنیم. با استفاده pinMode() وضعیت هر یک از پین ها(ورودی/خروجی)را مشخص میکنیم.با توجه به اینکه مدارمان به صورت Pull-Up می باشد؛ اگر سوییچ بسته باشدLOW مي باشد و اگر سوييچ باز باشد HIGH است.

مطابق دستور كار، ابتدا همه ى LEDها خاموش مى باشند. (با استفاده از digitalWrite() وضعیت خاموش بودن یا روشن بودن LEDها را با توجه به خواسته دستورکار مشخص می کنیم.)

* براى سوييچ نخست ، LED ها از سمت چپ به راست با تاخير ٣٠٠ ميلي ثانيه روشن مى شوند.
* اگر سوييچ دوم بسته باشد. با استفاده از تابع strlen() به اندازه ى اسم خودم(helia) مشخص مى كنيم كه LEDها چند بار با هم  روشن و خاموش شوند و بعد از اتمام آن روشن مى مانند.
* با بسته شدن سوييچ سوم  همه ى LEDها خاموش مى شوند.

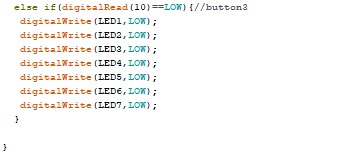


Define LEDS based on their pins number

**شرط دوم**

Set that which pins are input/output

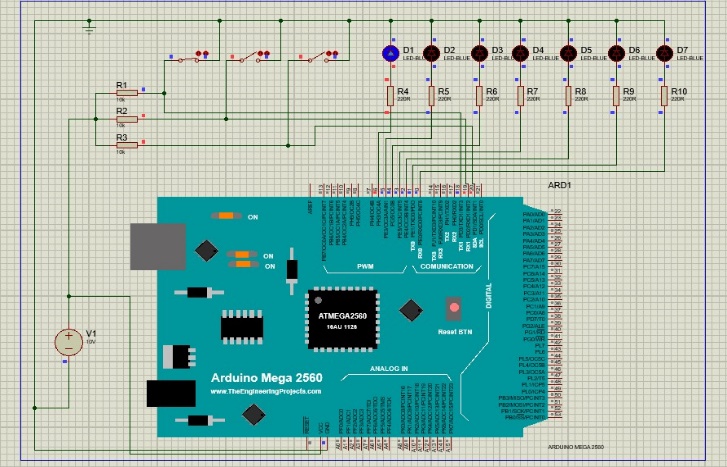


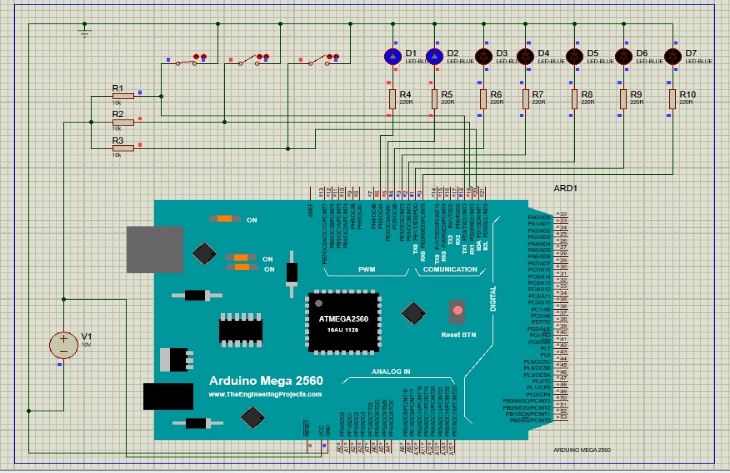


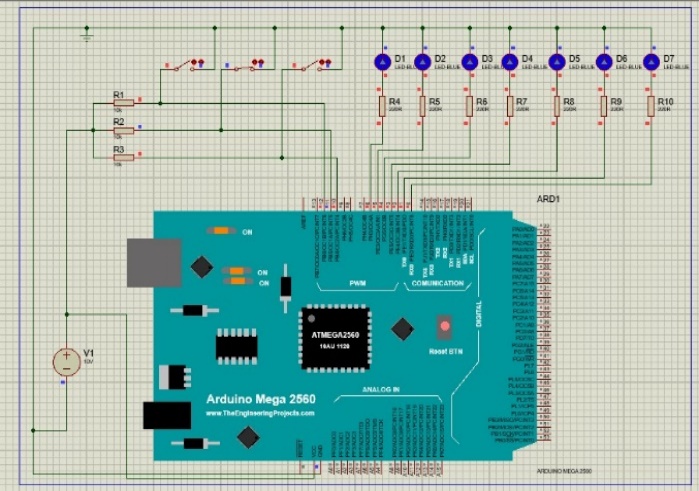
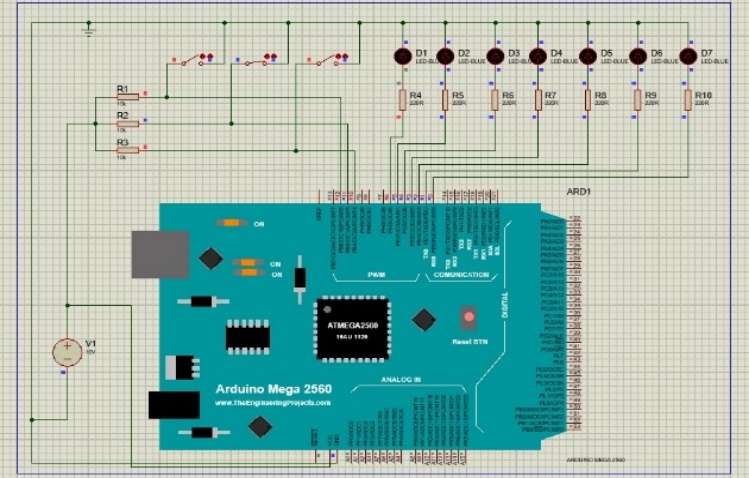
**شرط سوم**

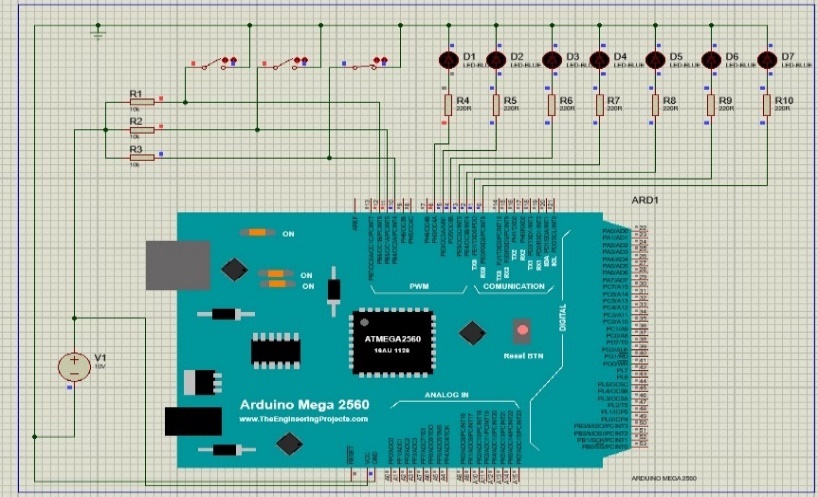
**شرط اول**

در مدارات زیر به جای button از switch استفاده شده است.

* با بسته شدن سوییچ اول LEDها به ترتیب از چپ به راست روشن می شوند.



* با بسته شدن سوییچ دوم LED ها ،5 بار چشمک زذه و در انتها روشن می مانند.
* با بسته شدن سوییچ سوم همه ی LEDها خاموش می شوند.



2. .به پرسشهای زیر پاسخ دهید:

* **اگر دکمه را در حالت فشرده برای زمان طولانی نگه داریم چه اتفاقی خواهد افتاد؟ آیا با منطق کارکرد خواسته شده سازگار است؟ چه راه حلی برای این مشکل (در صورت وجود) می توان پیشنهاد کرد؟**

اگر  سوییچ را به مدت طولانى نگه داريم چون درloop() به طور مداوم بسته بودن آن چك مى شود پس از آنكه يك بار كارش را انجام داد، به صورت مداوم آن عملیات تکرار می شود؛ که چنین چیزی نباید اتفاق بیفد و انجام آن به صورت یک بار مد نظر ما می باشد. راه حل پيشنهادى براى حل اين مشكل اين است كه مى توان يك شمارنده  برای هرسوییچ نظر گرفت(counter1,counter2,counteter3) و در هر كدام از شرط ها (if)  علاوه بر اينكه بسته بودن سوييچ مورد نظر چك مى شد، تعداد دفعات انجام عمليات نيز چك شود یعنی

شمارنده های هر سوییچ با تمام شدن یک دور از loop مورد نظر به اندازه ی یک واحد زیاد میشوند(counter++) اگر مقدار شمارنده ی مورد نظر از یک بیشتر بود و اگر كليد باز و بسته نشده بود حق اجراى دوباره را ندارد. توجه كنيم كه شمارنده ى مورد نظر هر سوييچ با هر بار باز شدن آن مقدارش صفر می شود ( به اصطلاح ريست مى شود=counter=0)

* **فرض کنید می خواهیم برد مورد نظر علاوه بر فراهم کردن کارکرد خواسته شده در بالا، عمل دیگری را نیز به را نیز هر LED صورت زمان دار انجام دهد. برای نمونه در کنار کارکرد بالا، وضعیت روشن یا خاموش بودن یک5 ثانیه یک بار تغییر دهد. روشی برای افزودن این کارکرد تازه به برنامه پیشنهاد دهید.**

ابتدا براى اينكه همزمان با كاركرد اول،كاركرد دومى را كه مى خواهيم انجام شود را داشته باشیم؛ از كتابخانه ىScheduler.h را به ابتداى

برنامه اضافه مى كنيم ( با اين كار مى توان چندين كار به صورت همزمان را انجام داد)

 يك متغير از  جنس بولین تعریف می کنیم(براىوضعيت روشن يا خاموش بودن LED)  سپس در loop اصلی با استفاده از دستور

digitalWrite() و يك شرط براى اينكه بدانيم وضعيت فعلى LED به چه صورت است تا با توجه به آن وضعیت LED را تغییر دهیم. (اگر HIGH بود به LOW تغييرش دهيم) حال براى اينكه ٥ ثانيه يك بار تغيير كند با استفاده ازدستور delay()مى توان پنج ثانيه تاخير را ايجاد كرد.

در روش بعدی می توان یک counter تعریف کرد که هر ثانیه یک بار تغییر کند و وقتی به 5 رسید حالت LED را تغییر دهد.

* **فرض کنید میخواهیم کارکرد دیگری را به دستگاه اضافه کنیم به این صورت که در صورت یک شدن یک پایه عملیات مشخصی را به عنوان پاسخ انجام دهد.(محدودیت زمانی برای پاسخ دادن وجود دارد) هیچ یک از اتفاق هاییک شدن پایه نباید از دست برود (بی پاسخ بماند). و یک شدن پایه نیز در هر زمانی ممکن است رخ دهد.آیا برنامهشما -که به روش سرکشی واحد های جانبی را بررسی میکند- می تواند در هر شرایطی (مثلا هنگام فشرده شدن کلید)این کارکرد را فراهم کند؟**

خير،هنگامى كه يك سوييچ بسته است (تا زمانى كه باز نشود) عمليات مورد نظر را انجام مى دهد. حال در اين بين اگر يك شدن يك پايه

 ديگر رخ دهد چون برنامه درحال اجراى عمليات سوييچ مورد نظر است؛ برنامه نمى تواند آن را بررسى بكند.

* **فرض کنید به دلیل محدودیت در توان مصرفی می خواهیم پردازنده در هنگام بیکاری به خواب برود. در زمان خواب پردازنده هیچ دستوری را اجرا نمی کند. روش سرکشی چه قدر با این نیازمندی سازگاری دارد؟ آیا می توان با این روش هم به خواب رفت و هم کارکرد درست آزمایش را فراهم کرد؟**

با روش سركشى نمى توان هم به خواب رفت و هم كاركرد درستى را فراهم كرد؛ زيرا در اين روش تمامى مراحل بصورت مداوم در حال اجرا

هستند و در صورتى كه به خواب برود چون وقفه اى وجود ندارد كه سيستم را بيدار كند ،كاركرد درستى ارائه نمى شود.

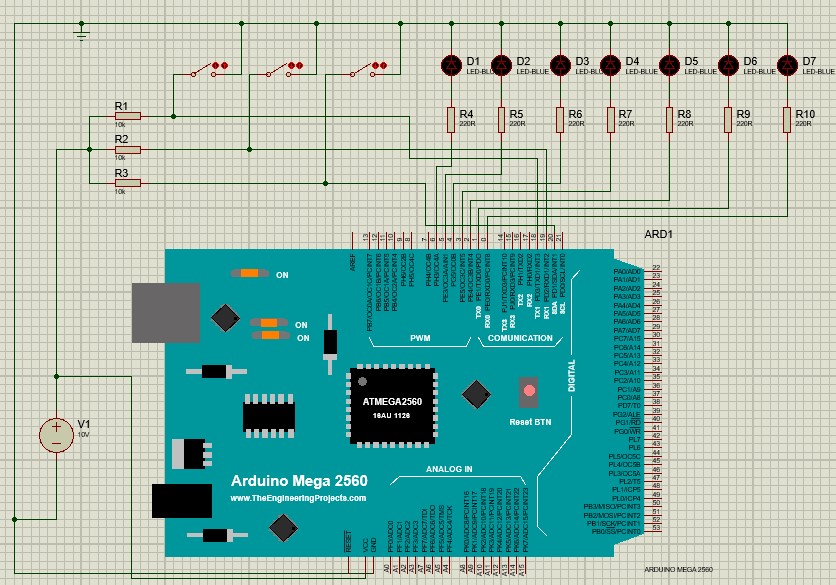
**3-با پاسخ به پرسش های بالا می توان دریافت که روش سرکشی برای کنترل واحد های جانبی با اینکه در برنامه های کوچک و به نسبت ساده قابل پیاده سازی است، همواره روش خوبی نیست و گاهی نمی تواند نیازمندی های ما را فراهم کند. اکنون آزمایش را به روش وقفه محور انجام دهید. پیاده سازی نیازمندی های خواسته شده در گام دوم را به روش سرکشی و وقفه محور مقایسه کنید.**

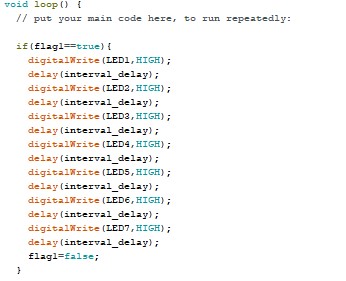
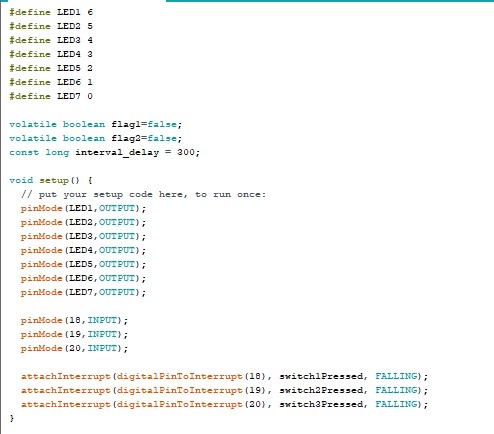
مطابق با مدارى كه براى قسمت سركشى بستيم در اين قسمت هم همان مدار را مى بنديم فقط براى سوييچ ها پين هاى مربوطه را (در  پيش گزارش پين هاى وقفه رامشخص كرديم) به پين هاى ١٨،١٩،٢٠ وصل مى كنيم.طبق دستور كار از دستور attachInterurpt()كه داراى سه  آرگومان هست استفاده مى كنيم.

* آرگومان نخست آن پین وقفه مورد نظر می باشد.
* آرگومان دوم ISRمى باشد.
* سومين آرگومان mode هست كه نشان مى دهد چه زمانى وقفه داريم. با توجه به اينكه مدار ما به صورت Pull-Up -هست آرگومان سوم FALLING میشود .

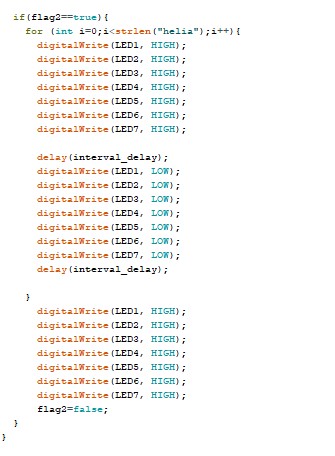
چون ISR تا حد امكان بايد كوتاه و سريع باشد و نمى توان بيشتر از چند ميكرو ثانيه تاخيرداشت بنابراين براى اطمينان از اينكه مقادير بين ISR برنامه به درستى به روزرسانى شده است ،دو متغير به صورت global به نام هاى flag1 و flag2 را  volatileتعريف مى كنيم و هر جا نياز به تاخير بود مقدار آن را  true میکنیم. برخلاف روش سركشى اگر سوييچ ها به مدت طولاني بسته باشند فقط يك بار عملیات مربوطه اجرا

می شود و از این روش می توان برای کاهش توان مصرفی ‍‍‍پردازنده بدون آنکه مشکلی پیش آید به خواب برود. مدار آن به صورت زیر است.

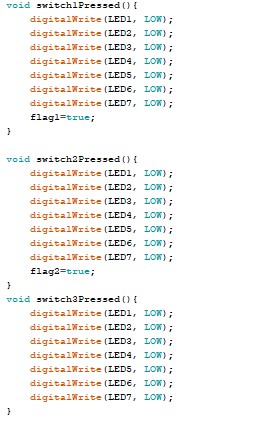
****

****

**شرط اول**

****

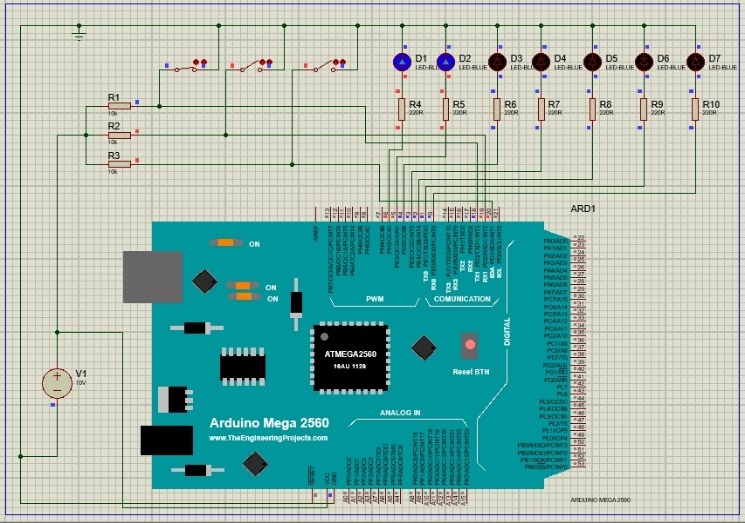
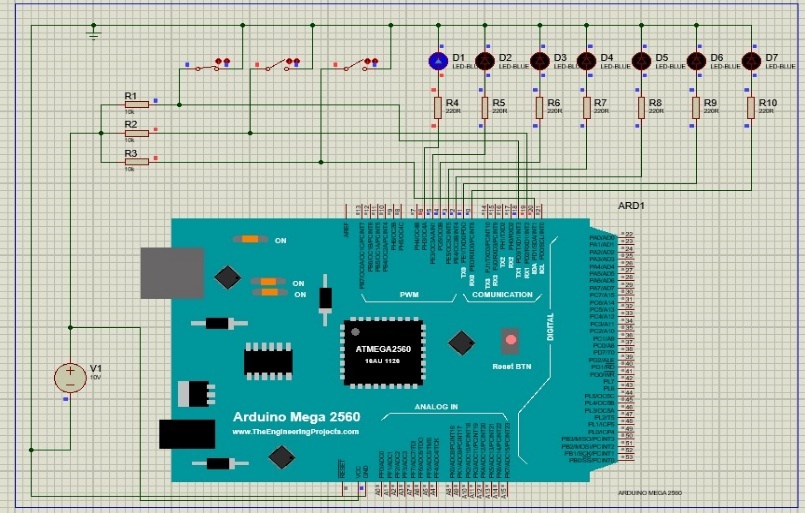
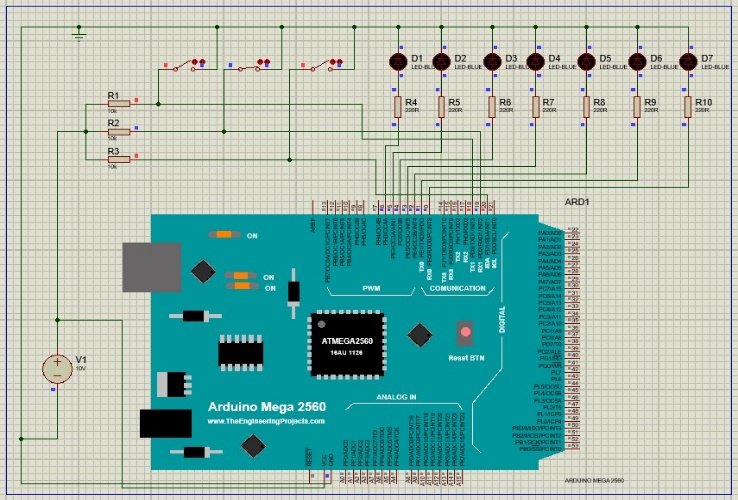
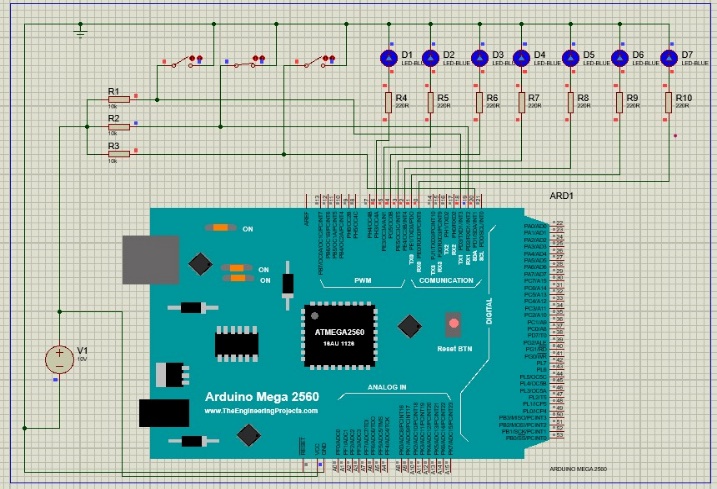
**پین های وقفه**

****

**شرط دوم**

**شرط سوم**

در مدارات زیر به جای button از switch استفاده شده است.

* ****با بسته شدن سوییچ اول LEDها به ترتیب از چپ به راست روشن می شوند.
* با بسته شدن سوییچ دوم LED ها ،5 بار چشمک زذه و در انتها روشن می مانند.
* با بسته شدن سوییچ سوم همه ی LEDها خاموش می شوند.

