

حسن کاظمی طهرانی

9629041

آزمایش شماره 4

آزمایشگاه ریزپردازنده و زبان اسمبلی

پیش گزارش:

کد ها در فایل ضمیمه موجود می باشد.

مفهوم PWM و کاربرد آن:

از آنجایی که در آردوینو در پین های دیجیتال می توان فقط 0 یا 1 را به عنوان خروجی داد نمی توان مقادیری مانند 0.75 یا 0.34 یا ... رو با یک پین دیجیتال که فقط مقادیر 0 یا 1 را میگیرد مشخص کرد . راهکار آن استفاده PWM و (تحلیل با پالس های مربعی) موج های مربعی است که تحلیل سیگنال به صورت افقی صورت می گیرد و با مدت زمانی که این سیگنال در یک دوره 1 بوده یا 0 بوده (Duty Cycle) مقادیر را استخراج می کنند . برای استفاده از PWM و موج های مربعی که به طور مثال برای کنترل سروو مورد استفاده قرار میگیرد باید از پین های PWM برد استفاده کرد یا به طور مثال خروجی دادن 2.5 ولت با خروجی منطقی (1و0) برای نیمه روشن کردن ال ای دی.

کاربرد های سروو موتور:

سروو موتور (: Servo Motor یک عملگر دورانی (خطی) است که امکان کنترل دقیق موقعیت زاویه ای یا خطی، سرعت و شتاب را فراهم می سازد. سروو موتور ها در نوع جریان مستقیم و جریان متناوب موجود هستند.

از کاربرد های آن می توان در باز و بسته کردن در مانند در گاوصندوق، جابه جا کردن اجسام سنگین ، راه رفتن ربات ها، تنظیم رادر و باله های هواپیما ها و به طور کل کنترل دقیق زاویه روتور در یک زاویه خاص و نگه داشتن آن.

توضیح در مورد ورودی آنالوگ و تحلیل آن و تابع analogRead() :

برای اینکه بتوان ورودی های آنالوگ را در آردوینو دریافت کرد نیاز است که به دیجیتال تبدیل شود که برد آردوینو با مداری که درونش وجود دارد از سیگنال آنالوگ نمونه برداری میکند در یک رنج خاصی مانند 0-1023 و هر ولتاژ را در زمان نمونه برداری به یک عدد نسبت میدهد و اینگونه ولتاژ

آنالوگ با نمونه برداری و کوانتیزه شدن به سیگنال دیجیتال تبدیل می شود که در برنامه نویسی برد با تابع `analogread()` میتوان از پین هایی که این قابلیت را دارند ولتاژ آنالوگ ورودی را دریافت کرد که خروجی این تابع یک عدد بین 0-1023 می باشد.

تعریف مختصر توابعه `servo.h` :

Attach() : این تابع پین کنترلی موتور مارا به عنوان ورودی می گیرد و به برد و برنامه معرفی میکند و بعد از ساخت شی موتور با این تابع می توان پین کنترلی آن را مشخص کرد.

Write() : عددی بین 0-360 در ورودی می گیرد و اگر سروو ما عادی بود زاویه آن را با توجه به ورودی تغییر می دهد و اگر 360 بود سرعت آن را تنظیم می کند.

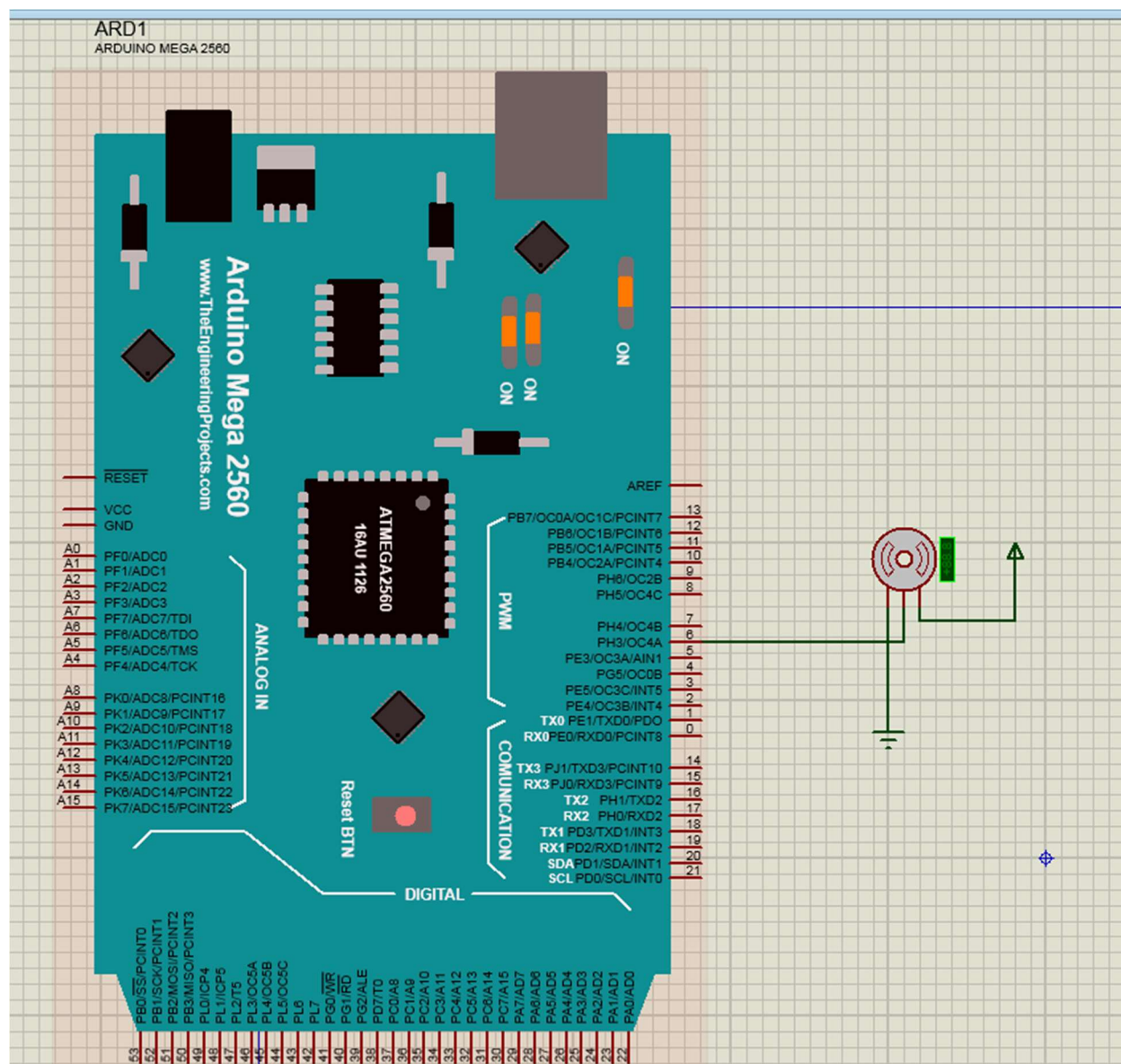
Read() : زاویه و یا سرعت سروو مارا میخواند و به ما برمیگرداند.

Writemicroseconds() : همان تابع `write` می باشد فقط دقت آن در سطح 1000 تا 2000 است .

Readmicroseconds() همان تابع `read` می باشد فقط دقت آن در سطح 1000 تا 2000 است .

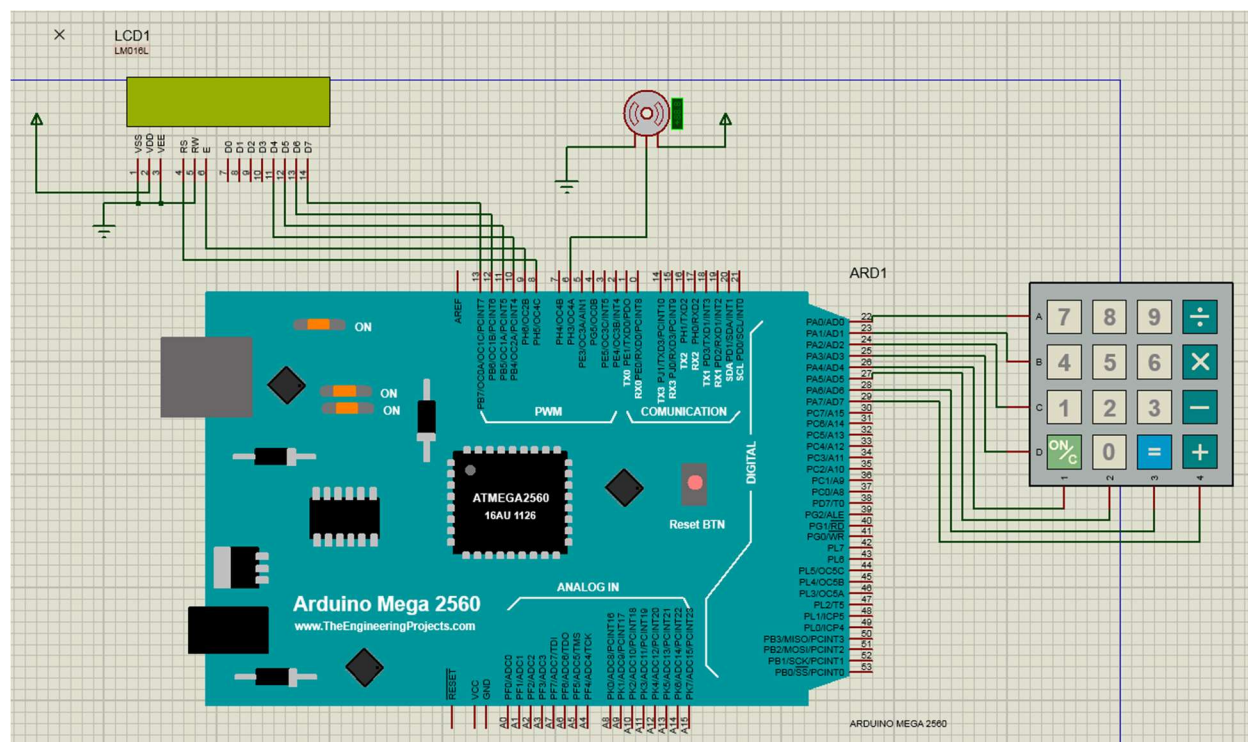
گزارش :

بخش 1



در این مدار میخواهیم به طور متناوب سروو موتور بین 90- و 90+ دوران کند. برای آن در کد از تابع write استفاده شده است که یک مقدار بین 0 تا 180 را اختیار میکند. 0 باعث چرخش 90- درجه و 180 باعث چرخش 90+ می شود.

بخش 2



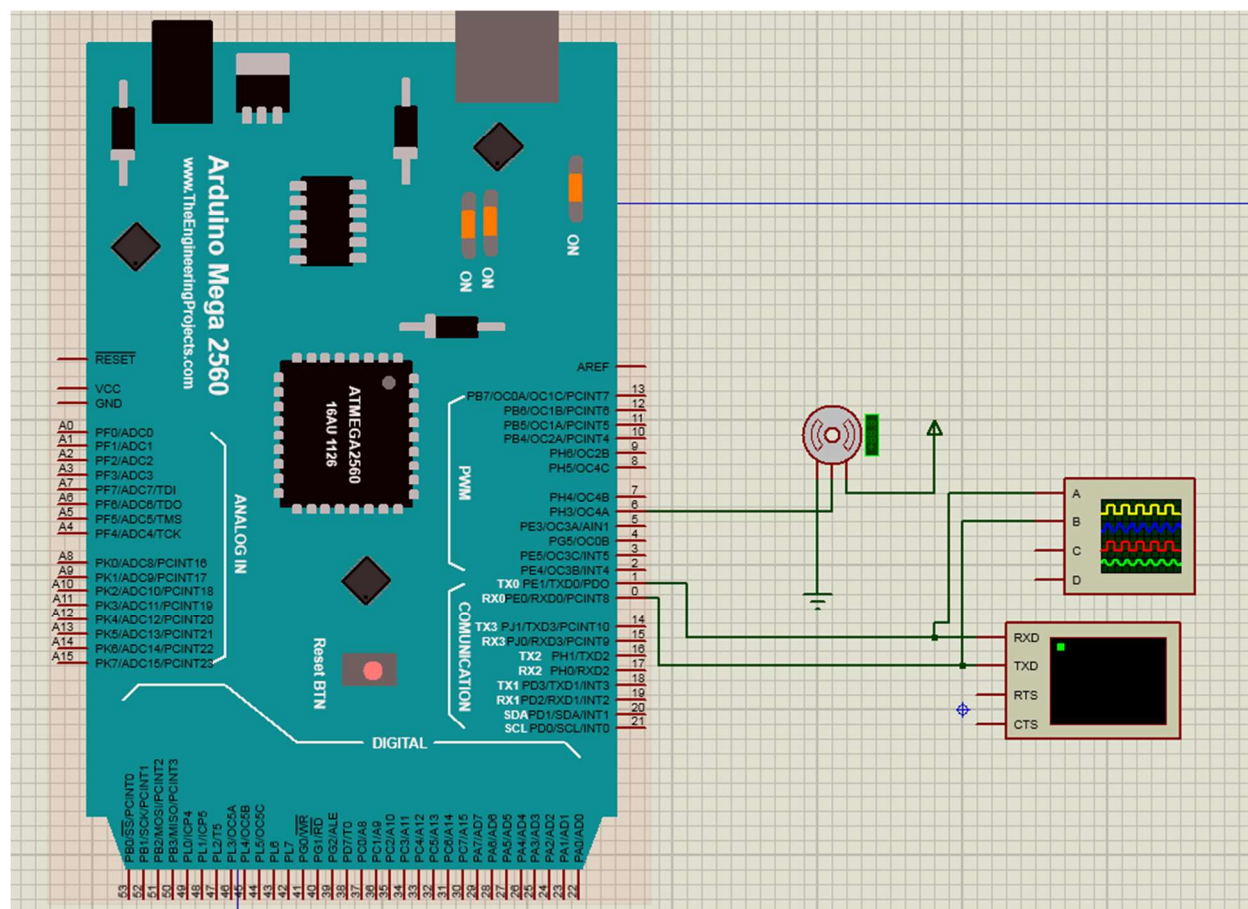
در این مدار از مدار ماشین حساب برای به کار انداختن سروو موتور استفاده شده است. به صورتی که با فشردن هر کلید شماره در صورت کوچکتر بودن از 180 به متغیر Number عدد اضافه میشود و در صورتی که کاربر کلید X را بفشارد آنگاه متغیر result برابر با true میشود و مقدار وارد شده به سروو موتور اعمال می شود و در غیر این صورت با فشردن هر کلید false می شود و تغییری مشاهده نمیشود.

برای تغییر مقدار سروو موتور از دستور زیر استفاده شده است:

```
myservo.writeMicroseconds(map(Number, 0, 180, 1000, 2000));
```

دستور map یک number را که بین 0 تا 180 می باشد و از جانب کاربر وارد شده را دریافت میکند و آن را بین 1000 تا 2000 مپ میکند. این اعداد بین 1000 تا 2000 برابر با طول پالس بر حسب میکرو ثانیه برای چرخاندن سروو موتور می باشد. عدد 1000 برای چرخیدن به صورت -90 درجه و 2000 برای چرخیدن به صورت +90 می باشد.

بخش 3

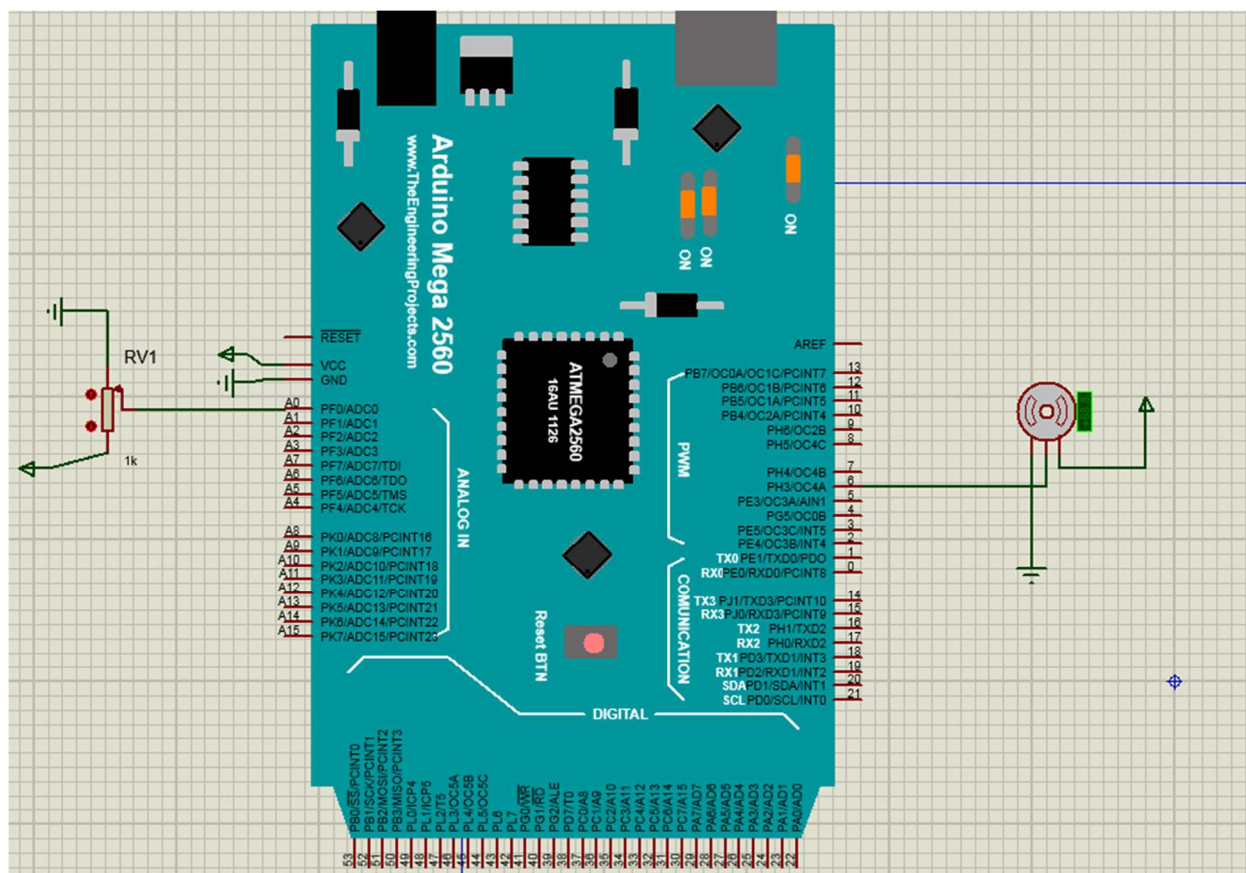


در این بخش می‌خواهیم به کمک Virtual Terminal مقداری که کاربر وارد میکند .

در بخش کد به کمک تابع `parseInt()` در یک `timeout` که به صورت پیش فرض 1000 میلی ثانیه می باشد ورودی را می خواند و در صورت وجود عدد آن را به `int` تبدیل میکند و خروجی میدهد و در غیر این صورت 0 بر میگردداند. لذا ورودی کاربر بین -90 تا +90 گرفته می شود. سپس به کمک `map` بین 1000 تا 2000 مپ می شود و بر سروو موتور اعمال می شود.

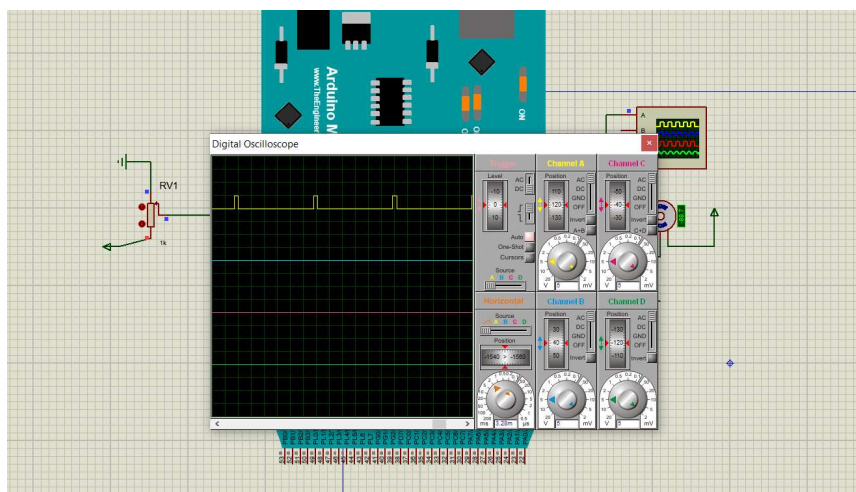
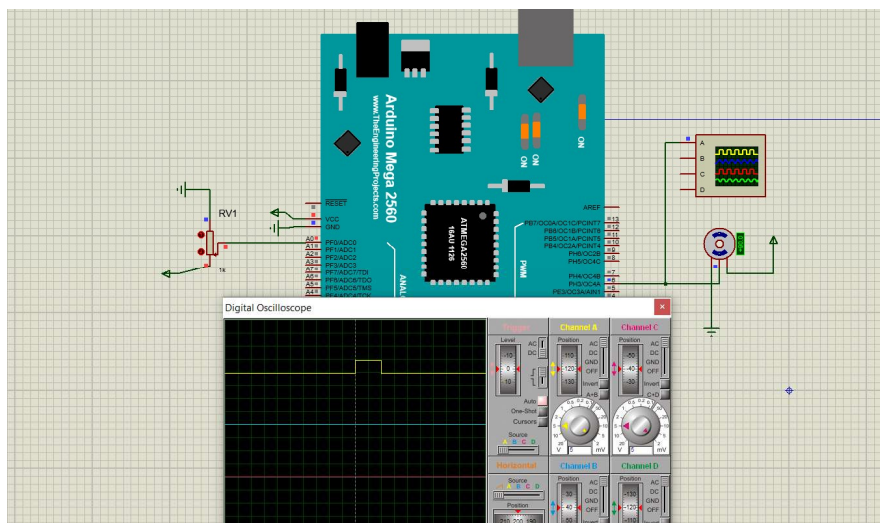
برای چاپ برر روی ترمینال نیز به کمک `print` و `println` می توان این کار را انجام داد.

بخش 4



در این بخش مقدار مقاومت پتانسیومتر به صورت آنالوگ به کمک دستور `analogRead()` خوانده میشود. مبدل داخل آردینو پس 100 میکرو ثانیه به یک عدد 10 بیتی بین 0 تا 1023 تبدیل میکند. این مقدار خوانده شده به کمک `map` بین 1000 تا 2000 مپ می شود و به کمک `writeMicroseconds()` نیز در سرو موتور اعمال میگردد.

بخش 5



با توجه به شکل های بالا طول هر مربع duty cycle و مدت زمان بین هر دو مربع دوره تناوب پایه می باشد.