



## برنامه نویسی میکروکنترلر جهت راه اندازی و کنترل انواع موتور الکتریکی

هدف از انجام این تمرین آشنایی با نحوه برنامه نویسی میکروکنترلر AVR و برد Arduino، شناخت چگونگی عملکرد انواع موتور الکتریکی (DC، Servo و Stepper) و راه اندازی و کنترل آن‌ها با استفاده از میکروکنترلر و آشنایی با مفاهیم PWM<sup>۱</sup> و ADC<sup>۲</sup> است. برای انجام این تمرین، از برد Arduino و شبیه ساز Proteus استفاده می‌شود.

### ➤ مفاهیم مقدماتی

- آشنایی با انواع موتورهای الکتریکی - موتور الکتریکی نوعی ماشین است که توانایی تبدیل انرژی الکتریکی به انرژی مکانیکی را دارد. موتور الکتریکی دارای سیم پیچ و آهنربا است و در اثر عبور جریان الکتریسیته از داخل سیم پیچ، میدان مغناطیسی تولید شده که باعث دوران شفت یا محور متحرک موتور می‌شود. در این تمرین با سه نوع مختلف از موتورهای الکتریکی متداول آشنا خواهیم شد:
- **موتور DC:** ساده ترین نوع موتورهای الکتریکی که با جریان مستقیم (DC) کار می‌کنند. حرکت چرخشی آنها توسط چند سیم‌پیچ تولید می‌شود. با توجه به شدت و جهت ولتاژ ورودی به این موتورها، می‌توان سرعت و جهت حرکت موتور را تعیین نمود.
- **موتور Stepper:** این نوع از موتور دور کامل را به تعدادی استپ یا پله مساوی تقسیم می‌کند و حرکت به جهت های چپ و راست تنها با زوایای معینی ممکن است. این موتورها در کاربرد هایی از جمله کنترل هارد دیسک، کنترل سنسور اسکنر، چاپگرهای ماتریس نقطه ای و روباتیک به منظور کنترل موقعیت استفاده می‌شوند. این موتورها در سرعت های بالا نیز قابل استفاده‌اند، به عنوان مثال برخی از انواع این موتور که در دستگاه‌های CD خوان استفاده شده‌اند تا ۴۰۰۰ دور در دقیقه سرعت چرخش دارند.
- **موتور Servo:** این نوع از موتور با کمک سنسورهای تعبیه شده، توانایی چرخش به اندازه هر زاویه‌ی دقیق دلخواه (بین ۰ تا ۳۶۰ درجه) که در ورودی به آن داده شود را دارد. این موتورها کاربرد گسترده‌ای در صنعت از جمله وسایل نقلیه رباتیک، حرکت نوار نقاله ها در خط تولید کارخانه ها، کنترل درب های اتوماتیک، فوکوس خودکار دوربین و کنترل زاویه در سیستم های ردیابی خورشیدی دارند. موتورهاى سروو بسیار سریع تر از موتورهاى استپر هستند و به طور کلی در سیستم‌هایی که نیاز به چرخش دقیق دارند، کاربرد دارند.

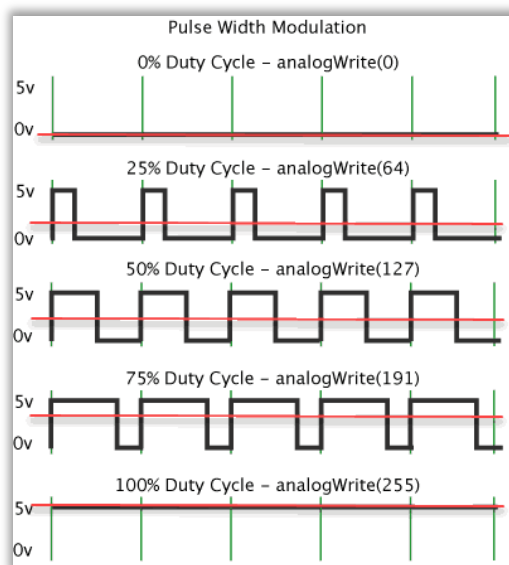
<sup>۱</sup> Pulse Width Modulation

<sup>۲</sup> Analog to Digital Converter

▪ کنترل موتورهای الکتریکی و آشنایی با مفهوم مدولاسیون عرض پالس (PWM) - کنترل سرعت و جهت حرکت بیشتر موتورهای مورد استفاده در سیستم های نهفته با تغییر ولتاژ ورودی آن ها انجام می شود. مسئله ای که در تعیین شدت ولتاژ ورودی موتورها مطرح می شود، این است که خروجی پورت های ورودی/خروجی میکروکنترلرها به صورت دیجیتال است و ولتاژ ثابتی برای نمایش ۰ و ۱ منطقی دارد. به این ترتیب به نظر می رسد نمی توان با استفاده از پورت های متداول، یک ولتاژ آنالوگ دلخواه برای موتور تولید کرد. برای رفع این مشکل از مفهومی به نام [PWM](#) برای ارسال سیگنال به موتور استفاده می کنیم.

مدولاسیون عرض پالس نوعی سیگنال به صورت قطاری از پالس های مربعی است که در هر زمان معین، موج در وضعیت بالا (High) یا پایین (Low) خواهد بود. چرخه کاری<sup>۲</sup>، مقدار زمان بالا بودن سیگنال نسبت به مدت یک دوره تناوب است. به این ترتیب که اگر سیگنال همواره ON باشد، چرخه کاری آن ۱۰۰ درصد و هنگامی که کاملاً Off باشد، مقدار چرخه کاری صفر است.

\* در روش PWM برای کنترل ولتاژ، از همان سطوح ولتاژ صفر و یک منطقی استفاده می شود، اما کنترل سرعت موتور توسط درصد زمانی که مقدار خروجی در یک بازه زمانی در سطح یک منطقی قرار دارد انجام می گیرد.



شکل ۱. مدولاسیون عرض پالس (PWM)

به عنوان مثال در قسمت دوم شکل بالا، مقدار سیگنال در ۲۵٪ از بازه زمانی یک است و موتور کنترل شده توسط این سیگنال با ۲۵٪ بیشینه سرعت خود، کار خواهد کرد. معمولاً برای نمایش بهتر، عدد اعشاری به دست آمده را در یک عدد مبدا (مثال ۲۵۶) ضرب می کنند تا خروجی یک عدد صحیح باشد. به طور مثال در قسمت دوم از شکل بالا که در ۲۵ درصد از زمان هر دوره تناوب، خروجی برابر یک است خواهیم داشت، ( $۲۵۶ \times ۰.۲۵ = ۶۴$ ) و بنابراین عددی که توسط PWM نشان داده میشود برابر ۶۴ است.

به عبارتی PWM را می‌توان برعکس مبدل آنالوگ به دیجیتال (ADC) فرض کرد که برای تولید سیگنال‌های آنالوگ از یک قطعه دیجیتال مانند میکروکنترلر به کار می‌رود.

#### ▪ ابزارهای شبیه سازی:

- **پلتفرم آردوینو (Arduino):** آردوینو یک پلتفرم سخت افزاری-نرم افزاری متن باز است که به منظور تولید سریع و ساده پروژه‌های سخت‌افزاری تعاملی و ساخت وسایلی که با محیط تعامل داشته باشند طراحی شده است. پلتفرم آردوینو از یک برد سخت افزاری قابل برنامه ریزی (شامل میکروکنترلر) و محیط نرم افزاری گرافیکی (IDE) به منظور برنامه نویسی و بارگذاری کدها بر روی میکروکنترلر تشکیل شده است. اغلب بوردهای آردوینو که تمام آن‌ها سخت‌افزار متن‌باز هستند بر پایه میکروکنترلرهای AVR و تعداد دیگری از بوردهای آردوینو بر پایه میکروکنترلرهای ARM طراحی شده‌اند و توسط درگاه‌های مختلف از جمله تعدادی پورت GPIO و ADC ارتباط میکروکنترلر با دنیای بیرون را برقرار می‌سازند. همچنین نرم‌افزار متن باز آردوینو که جهت نوشتن، ویرایش و عیب یابی کد مورد استفاده قرار می‌گیرد، دارای کتابخانه‌های بسیار کامل و رابط کاربری ساده‌ای است و با استفاده از آن می‌توان بوردهای آردوینو را بدون نیاز به پروگرامر از طریق کابل USB برنامه‌ریزی نمود. به عنوان مثالی از کاربردهای تعاملی، برد آردوینو می‌تواند مقادیر ورودی را از تعداد زیادی سنسور، کلید و... بخواند و بر اساس برنامه ای که درون آن بارگذاری شده است، می‌تواند کنترل تعدادی لامپ، موتور و... را کنترل کند. این بوردها در انواع، اندازه ها و کاربردهای مختلف با نام های Mini, Nano, Mega, Uno و... طراحی و تولید شده اند. برای آشنایی بیشتر با قواعد برنامه‌نویسی آردوینو می‌توانید به [این لینک](#) مراجعه کنید.

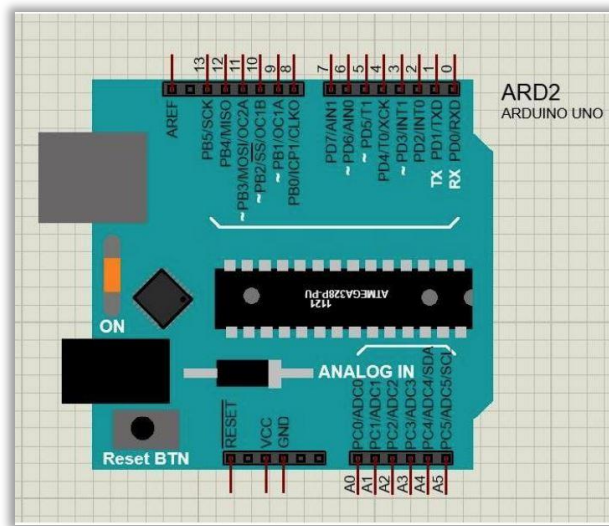
- **شبیه ساز پروتئوس (Proteus):** پروتئوس یکی از نرم افزارهای کاربردی جهت طراحی خودکار، مدلسازی، شبیه سازی، کد نویسی، طراحی شماتیک و مسیریابی در مدارات الکترونیکی و مدارهای قابل برنامه ریزی از جمله میکروکنترلرها و ریز پردازنده ها است.

\* شبیه سازی مداراتی که با استفاده از بوردهای آردوینو (مبتنی بر میکروکنترلرها) ساخته می‌شوند نیز یکی از توانایی های نرم افزار پروتئوس است. تمامی بوردهای آردوینو در نرم افزار پروتئوس قابلیت شبیه سازی دارند اما برای این منظور لازم است کتابخانه های این بوردها به نرم افزار اضافه شوند. در این تمرین، این روش شبیه سازی مورد استفاده قرار می‌گیرد.

## ➤ شرح تمرین

- شبیه سازی **Arduino** در **Proteus**: برای انجام این تمرین برد **Arduino UNO** که شامل میکروکنترلر **ATmega328P AVR** است، مورد استفاده قرار می گیرد. برای برنامه نویسی آردوینو نیاز به نصب برنامه ی **IDE** **Arduino** دارید که از [این لینک](#) قابل دریافت است.

برای استفاده از شبیه ساز **Arduino** در **Proteus** نیاز به نصب کتابخانه ی مربوط به آن را دارید. مراحل دانلود این کتابخانه و مراحل نصب آن را می توانید در [این لینک](#) مشاهده کنید. بعد از نصب این کتابخانه، می توانید همانند عکس زیر، بلوک مربوط به آردوینو را به پروتئوس اضافه کنید.



شکل ۲. بلوک **Arduino UNO** در نرم افزار **Proteus**

- این تمرین دو مرحله دارد که شامل کار با موتورهای مختلف خواهد بود. شما باید کد هر مرحله را برای **Arduino** تولید کرده و با کمک ابزار **Proteus** آن را شبیه سازی کنید.

۱. در بخش اول با موتور **DC** کار خواهیم کرد. نکته ی مهمی که باید برای کار با این موتور در نظر گرفت، نحوه ی اتصال آن به برد آردوینو است. آردوینو برای برقراری ارتباط با ابزارهای جانبی از درگاه های **GPIO** استفاده می کند. اما به دلیل پایین بودن جریان این درگاه ها امکان اتصال مستقیم آنها به موتور وجود ندارد. برای حل این مشکل از قطعه ای به نام **driver** استفاده می شود که هم به منبع تغذیه وصل می شود (برای دریافت جریان کافی) و هم به میکروکنترلر (برای دریافت سیگنال کنترلی). خروجی آن نیز به موتور مورد نظر وصل خواهد شد. در این تمرین، لازم است که موتور را با کمک درایور **L293** به برد متصل کنید.

در ابتدا موتور در جهت ساعتگرد و با حداکثر سرعت می چرخد. سپس باید با کمک چهار کلید که به پورتهای برد آردوینو متصل شده‌اند، دستوراتی را به برد ارسال کنید که در نتیجه‌ی آن، تغییرات زیر در رفتار موتور دیده شود:

- قطع یا ادامه‌ی حرکت موتور
- افزایش سرعت حرکت موتور
- کاهش سرعت حرکت موتور
- تغییر جهت حرکت موتور از ساعتگرد به پادساعتگرد و برعکس

برای تغییر سرعت حرکت موتور لازم است از PWM استفاده کنید. اگر حداکثر مقدار PWM برابر با ۲۵۵ و حداقل آن صفر در نظر گرفته شود، برای افزایش یا کاهش سرعت باید مقدار PWM را ۱۰ واحد تغییر دهید. همچنین توجه کنید که مقدار PWM کمتر از صفر یا بیشتر از ۲۵۵ نشود.

۲. در دومین مرحله این تمرین، با موتور Stepper کار می کنیم. در این تمرین از حالت Bipolar این موتور استفاده می شود. در این قسمت موتور در ابتدا در حال حرکت در جهت عقربه‌های ساعت در فاصله‌های ۱۰ درجه‌ای است. کاربر باید بتواند با کمک سه دکمه‌ی جانبی، فعالیت‌های زیر را انجام دهد:

- جهت چرخش‌های موتور هم‌جهت با عقربه‌های ساعت شود.
- جهت چرخش‌های موتور مخالف با عقربه‌های ساعت شود.
- حرکت موتور متوقف شود تا زمانی که کاربر یکی از دکمه‌های قبلی را فشار دهد.

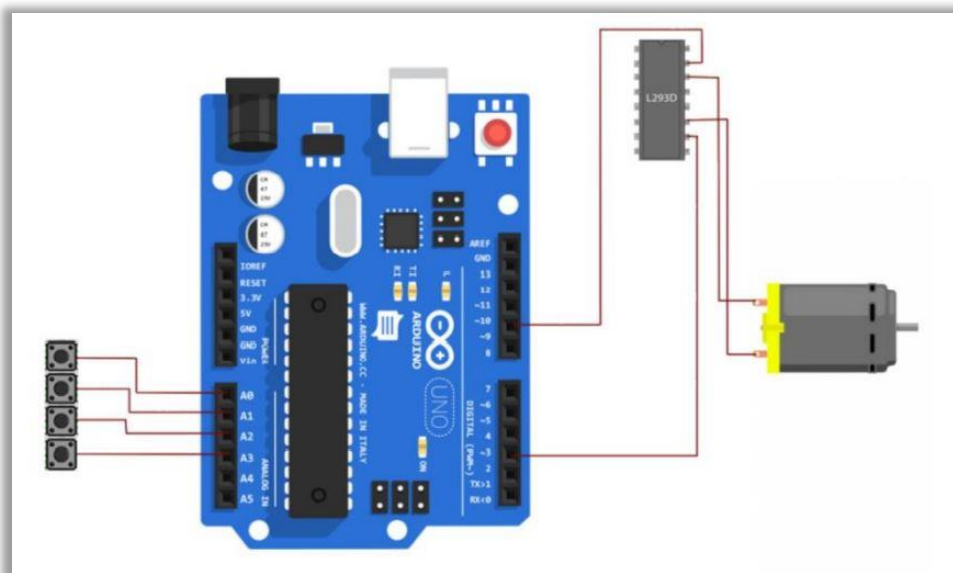
\* در گزارش کار لازم است علاوه بر توصیف نکات هر بخش به سوالات زیر پاسخ دهید:

۱. نحوه‌ی ساخت PWM را شرح دهید.
۲. استفاده از PWM در موتورهای DC و Servo چه تفاوتی دارد؟
۳. تفاوت موتورهای Servo و Stepper را شرح دهید و یک مورد کاربرد برای هر کدام بیان کنید.
۴. در بسیاری از تلفن‌های همراه، برای ساخت vibrator از ساختاری به نام ERM استفاده می شود. این ساختار از چه نوع موتوری استفاده می‌کند؟ ساختار کلی آن را شرح دهید.

➤ **طراحی مفهومی**<sup>۵</sup> کمک می کند که بدون در نظر گرفتن جزئیات، نحوه‌ی اتصالات و مشکلات پیش روی طراحی، توصیفی کلی از عملکرد سیستم و همچنین ارتباطات سطح بالای اجزا با هم ارائه شود.

طراحی مفهومی جزئی مهم در ارائه‌ی پروژه است. برای آشنایی بیشتر شما با کلیت این مفهوم، در این پروژه نمونه‌ای از آن برای اتصال موتور DC به آردوینو آورده شده است. بدیهی است که این مثال بسیار ساده بوده و طراحی مفهومی پیچیده‌ای نخواهد داشت و صرفاً کاربرد آشنایی شما با این مفهوم را دارد. ارتباطات اجزا در

این پروژه در تصویر زیر به نمایش درآمده است. دقت کنید که جزئیات معماری سیستم و جزئیات اتصالات مانند نحوه ی سیگنالینگ در طراحی مفهومی موضوع بحث نیست.



شکل ۳. طراحی مفهومی اتصال موتور DC به برد Arduino

این تصویر به صورت کلی به شما نشان میدهد که از ۴ کلید برای کنترل وضعیت موتور استفاده می شود. و برای کنترل سرعت و جهت موتور، از یک درایور L293 که به پورت های PWM برد متصل است استفاده شده است. نکاتی مثل power و جزئیات اتصالات مانند مدارمقاومتی Pull Up که برای کلیدها ضروری است و یا پروتکل ارتباطی میان قطعات در این طراحی نمایش داده نشده اند. همچنین، چون پیاده سازی این دست سیستم ها ترکیبی از سخت افزار و نرم افزار است، نوشتن یک سودوکد به درک بهتر عملکرد سیستم کمک خواهد کرد.

#### **Pseudocode:**

```
main function definition {  
  Initialization for motor state and pins;  
  while(true) {  
    if (pause_resumption_button)  
      stop or begin the rotation of motor;  
    if (increase_speed_button)  
      increase speed of motor;  
    if (decrease_speed_button)  
      decrease speed of motor;  
    if (change_direction_button)  
      change the direction of rotation;  
    update state;  
  }  
}
```

## ➤ نکات تحویل تمرین

- این تمرین تحویل حضوری دارد و باید در قالب گروه‌های چهار نفره انجام شود. موعد تحویل تمرین در صفحه درس اعلام شده است.
- گزارش کار کاملی از مراحل انجام تمرین ، نتایج شبیه سازی، تصاویر و نکات هر بخش و پاسخ به سوالات تهیه شود. برای قسمت دوم تمرین (موتور stepper) ارایه یک مدل مفهومی (شکل+شبه کد) لازم است.
- کدهای تمرین باید همراه باید با نامگذاری مناسب همراه با گزارش کار تحویل شوند.

موفق باشید