

## فهرست

| صفحه | عنوان                                                |
|------|------------------------------------------------------|
| ۳    | مقدمه                                                |
| ۴    | چرا روبات تعقیب خط؟                                  |
| ۴    | روبات تعقیب خط چه باید بکند؟                         |
| ۴    | بلوک دیاگرام یک روبات تعقیب خط                       |
| ۵    | عملکرد روبات تعقیب خط                                |
| ۷    | (۱) سنسورهای روبات تعقیب خط                          |
| ۷    | الف) دیود مادون قرمز                                 |
| ۸    | ب) مقاومت متغیر با نور (فوتو-رزلیستور یا فوتوسل)     |
| ۸    | ج) فوتو-ترانزیستور                                   |
| ۹    | تعداد سنسورها                                        |
| ۹    | (۲) تبدیل و انتقال خروجی سنسور به مدار پردازشگر      |
| ۱۰   | مدار سویچینگ                                         |
| ۱۱   | مدار ترانزیستوری                                     |
| ۱۱   | مدار سویچ با آپامپ                                   |
| ۱۲   | مدار اشمیت تریگر                                     |
| ۱۳   | تبدیل خروجی آنالوگ سنسور به ولتاژ دیجیتال به کمک A/D |
| ۱۴   | (۳) بخش پردازشگر                                     |
| ۱۵   | مدار منطقی                                           |
| ۱۵   | میکروکنترلر                                          |
| ۱۷   | (۴) موتور و مدارهای کنترل آن                         |
| ۱۷   | آرمیچر یا موتور DC                                   |
| ۱۷   | مدار کنترلی آرمیچر                                   |
| ۱۸   | معکوس کردن جهت چرخش موتور                            |
| ۱۸   | موتور پله‌ای                                         |
| ۱۹   | مدار کنترل موتور پله‌ای                              |
| ۱۹   | تولید رشته‌های «صفر» و «یک»                          |
| ۱۹   | میکروکنترلر                                          |
| ۲۰   | شیفت رجیستر                                          |
| ۲۰   | مدار تقویت جریان                                     |
| ۲۱   | بافر                                                 |
| ۲۱   | (۵) مکانیک روبات                                     |
| ۲۲   | از کجا شروع کنیم؟                                    |
| ۲۳   | مراجع                                                |
| ۲۴   | کلام آخر                                             |

## مقدمه

ساخت روبات از موضوعات بسیار جالب و آموزنده است که مجموعه اطلاعاتی از علوم مکانیک، الکترونیک، کامپیوتر (سخت‌افزار و نرم‌افزار) و هوش مصنوعی را در قالب یک وسیله هوشمند نمود عملی می‌بخشد. شاید ساخت تک‌تک اجزایی که در یک روبات به کار می‌رود (مانند مدار کنترلی، برنامه نرم‌افزاری، ارتباط با کامپیوتر، سنسورها، عملگرها، موتورها و بدنه) کار چندان مشکلی نباشد، اما در کنار هم قرار دادن این اجزاء، تأمین ارتباط مناسب و بهینه، جبران‌سازی ضعفهای یک بلوک به کمک بلوکهای دیگر و در نهایت به وجود آوردن روباتی که تا چند روز پیش چیزی جز چند قطعه و بدنه و مدار نبود و حال به صورت موجودی هوشمند به حرکت درآمده و آنچه شما دستور می‌دهید را انجام می‌دهد، یک «مهندسی کامل» است؛ آنچه انتظار می‌رود یک دانشجوی مهندسی در هر رشته‌ای (مکانیک، برق، کامپیوتر، صنایع و ...) قادر به انجام آن باشد.

این نوشتار یک کتاب درسی نیست؛ بنابراین توضیحات تخصصی راجع به سنسورها و عملگرها و مدار و مکانیک در آن کمتر وارد شده است. نگارنده این سطور نیز تحصیل کرده سخت‌افزار کامپیوتر است و آنچه از الکترونیک و مکانیک می‌داند را از دیگران که بیشتر می‌دانند آموخته است. هدف این نوشته آشنا کردن شما با دانش روباتیک و به صورت ویژه بیان اصول ساختن یک روبات بسیار ساده یعنی روبات تعقیب خط می‌باشد تا در حد توان در رفع نقیصه روباتیک کشورمان که همانا فقدان منابع گویا و ساده است کاری انجام داده باشیم.

مطالب این نوشته با فرض اینکه خواننده از اصول ابتدایی مدارات منطقی، میکروکنترلرها و الکترونیک آگاه است به رشته تحریر درآمده است. اما اگر هر کدام از این مطالب را نمی‌دانید، به جای اینکه تا وقتی که واحد درسی آن را بگیرید، به خود استراحت بدهید! تلاش کنید و با جستجو در منابع و پرسش از آنان که می‌دانند راه خود را بکشایید.

این مطالب حاصل دانسته‌ها و تجربیات نگارنده در گروه روباتیک دانشگاه اصفهان است. وظیفه خود می‌دانم از دکتر کمال جمشیدی سرپرست گروه روباتیک دانشگاه اصفهان که به من آموخت هر کس وظیفه‌ای در قبال آنچه هست دارد که با انجام آن وظیفه باید دین خود را نه به زبردستان که به زبردستان ادا کند و «حق‌الزحمه» آن را نه از مسئولان که از «او» که می‌بیند و می‌داند بگیرد، صمیمانه تشکر و قدردانی کنم. سپاس بی‌کران خود را نیز نثار دوستان گرانقدرم مهندس احسان خویش اردستانی و مهندس محمد حکمت‌نژاد می‌نمایم که یاران عزیز و یکدل من در گروه روباتیک دانشگاه اصفهان بودند و در نوشتن این مطالب از دانسته‌ها و نکاتی که یادآوری کرده‌اند بسیار استفاده برده‌ام.

امیدوارم این نوشته بتواند راهگشای شما برای آنچه که می‌توانید باشید و آنچه باید باشید، گردد. نظرات و پیشنهادهای خود را به آدرس [rasti@eng.ui.ac.ir](mailto:rasti@eng.ui.ac.ir) ارسال فرمایید تا نسخه‌های بعدی این نوشتار جامع‌تر و کاملتر ارائه گردد.

جواد راستی

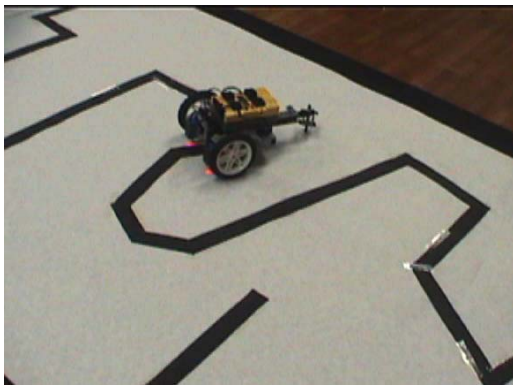
اسفند ۱۳۸۶

## چرا روبات تعقیب خط؟

در میان روباتهای مختلفی که در حیطة مسابقات (مانند روبات مین‌یاب، لایبرنت، موش پنیریاب، جنگجو، فوتبالیست و ...) یا در حیطة صنعتی (روباتهای نقاش، جوشکار، مونتاژ قطعات، تیرانداز و ...) وجود دارد، می‌توان روبات تعقیب خط را ساده‌ترین عضو خانواده روباتها انگاشت. عضوی که به رغم بی‌پیرایه بودن و سادگی، دارای تمام اجزایی است که در یک روبات باید وجود داشته باشد. به همین لحاظ، شاید بهترین و روشن‌ترین دروازه ورود به دانش روباتیک، روبات تعقیب خط باشد. به همین لحاظ در این نوشته به معرفی کلی اجزای این روبات می‌پردازیم و آنچه باید برای ساخت آن انجام شود را مرور می‌کنیم.

از همین ابتدا فراموش نکنید که این مطلب روزنامه نیست! به جای خواندن آن در حین تماشای تلویزیون یا داخل اتوبوس، دست به کار شوید و وسایل لازم برای ساخت این روبات که خوشبختانه ارزان و ساده است، را فراهم آورید و همگام با خواندن مطالب، اجزاء و قطعات را بسازید تا بتوانید به زودی توڈ روبات خود را جشن بگیرید!

## روبات تعقیب خط چه باید بکند؟



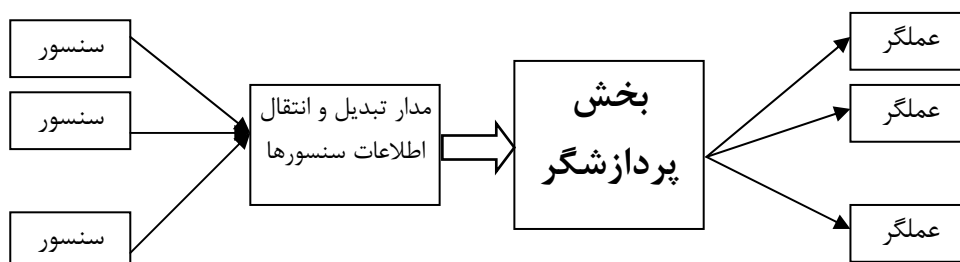
صورت مسأله ساخت روبات تعقیب خط در مسابقات مختلف دارای جزئیات زیاد و بعضاً متفاوتی است؛ اما اصل موضوع در همه آنها این است که روبات تعقیب خط باید بتواند یک خط مشکی‌رنگ به عرض  $1/8$  سانتی‌متر (معمولاً چسب برق) را در یک صفحه سفید دنبال کند. یعنی وقتی در ابتدای خط مشکی تنظیم و سپس روشن شود، بدون نیاز به دخالت کاربر به دنبال خط مشکی حرکت کند. یک مسیر مسابقات روبات تعقیب خط را در شکل روبرو می‌بینید.

اخیراً مقررات جدیدی از جمله امکان وجود بریدگی در خط، وجود تونل، پیچهای ۹۰ درجه، حلقه، تقاطع و ... به قوانین مسابقات اضافه شده است. ترسی به خود راه ندهید! اگر روبات خود را در ساده‌ترین حالت ممکن بسازید، اضافه کردن این امکانات به آن کار مشکلی نیست. دست‌کم بهتر از نشستن و تماشا کردن است!

## بلوک دیاگرام یک روبات تعقیب خط

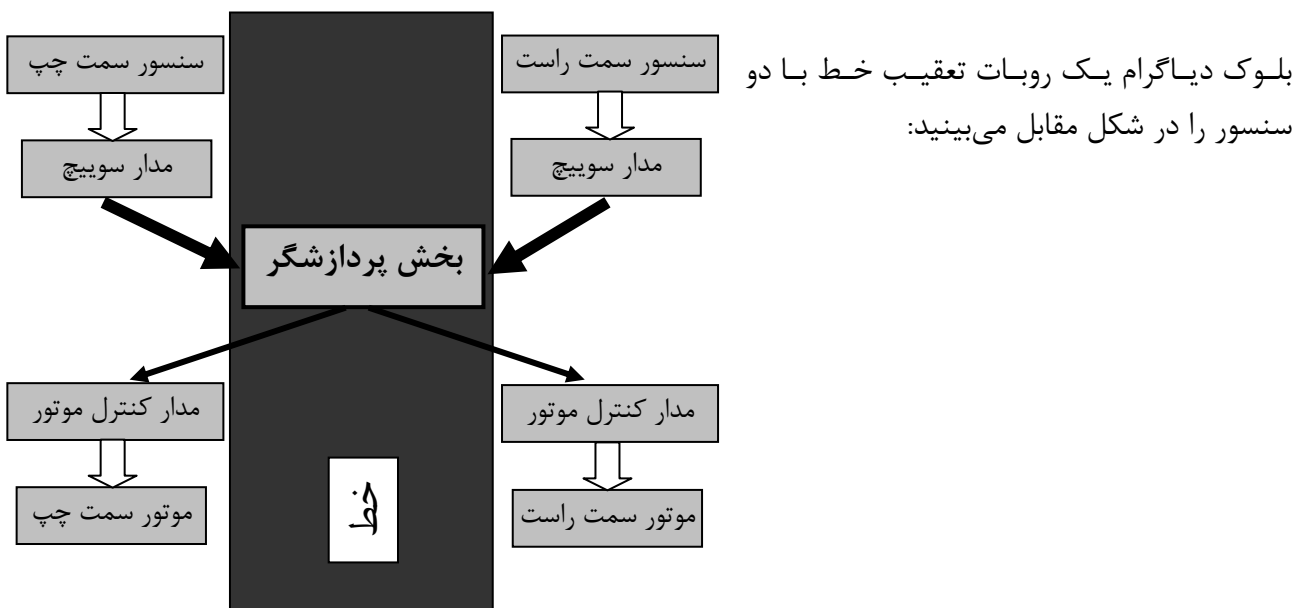
از نظر کلی هر روبات باید

- از طریق سنسورهایش (حسگرها) اطلاعاتی را از دنیای خارج جمع‌آوری کند،
- این اطلاعات را به اطلاعات قابل فهم مغز پردازشگر خود تبدیل و به آن منتقل کند،
- بر طبق برنامه‌ریزی قبلی، اطلاعات را پردازش و تصمیم مناسب را اتخاذ کند،
- تصمیمات نهایی را به کمک عملگرهایش اجرا کند.



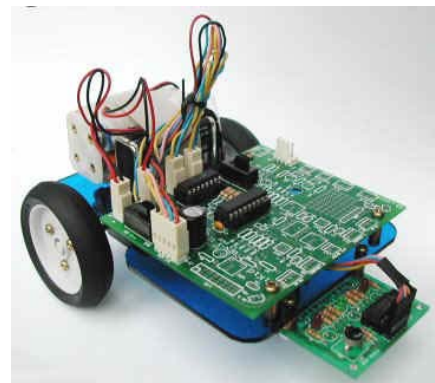
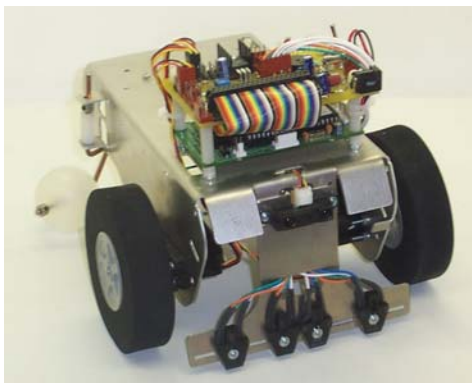
تفاوت روباتهای مختلف به ماهیت این مراحل برمی گردد. روبات تعقیب خط هم همین اجزاء را درون خودش دارد:

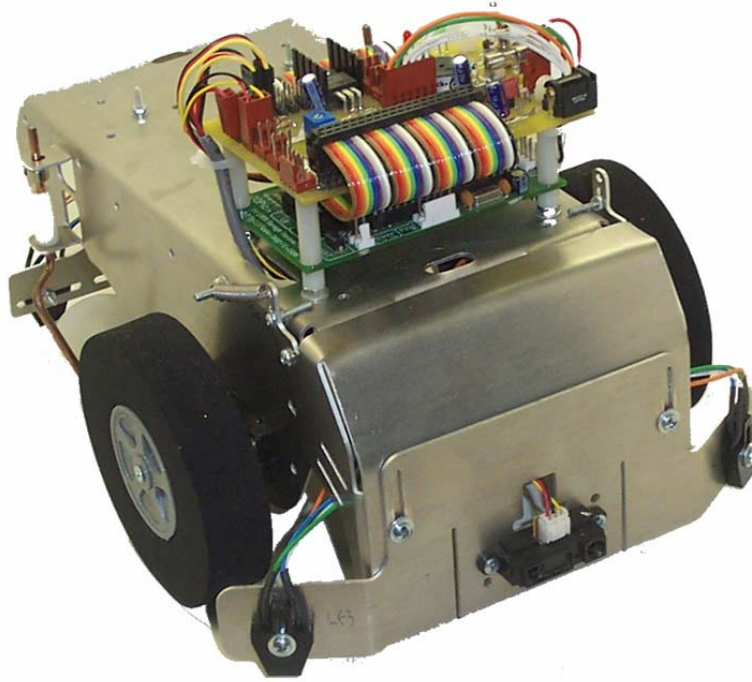
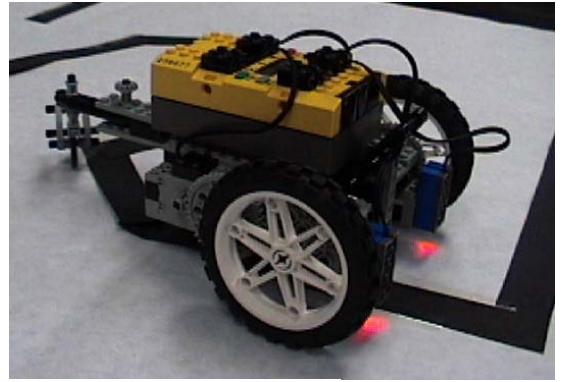
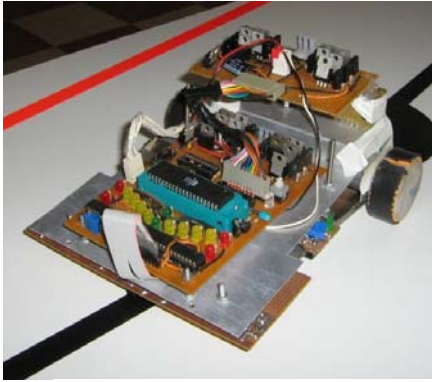
- برای حس کردن خط به دست کم دو سنسور نوری که تفاوت رنگ سفید و مشکی را درک کنند نیاز دارد. این سنسورها انواع مختلفی مانند دیود مادون قرمز، مقاومت متغیر با نور (فوتوسل)، فوتو-ترانزیستور و ... دارند.
- اطلاعات سنسور از طریق یک مدار سویچ یا مبدل آنالوگ به دیجیتال به واحد پردازشگر منتقل می شود.
- مغز پردازشگر این روبات می تواند یک مدار منطقی یا یک میکروکنترلر از هر نوعی باشد. روبات باید در قبال منحرف شدن خط، واکنش مناسب نشان داده و به قولی «هرجا خط پیچید او هم پیچد».
- تصمیم واحد پردازشگر مبنی بر ادامه مسیر یا پیچیدن به چپ و راست از طریق فرمانهایی که به موتورهای روبات که ممکن است از نوع DC (آرمیچر) یا پله ای (Stepper) باشند داده می شود، انجام می پذیرد.



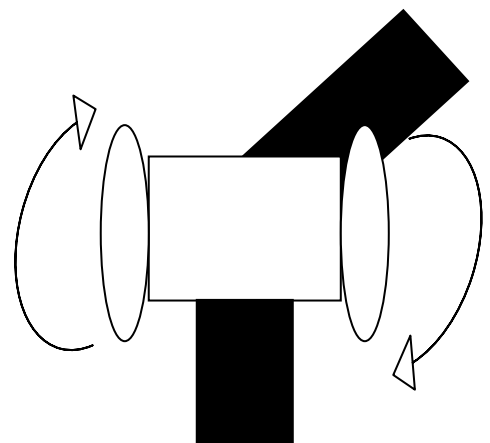
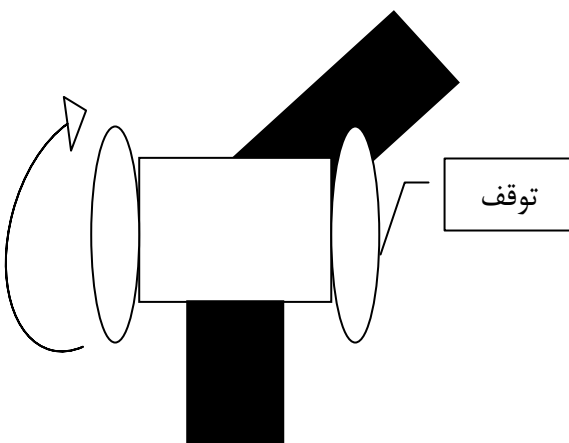
## عملکرد روبات تعقیب خط

روبات تعقیب خط بسیار ساده عمل می کند. این روبات دست کم دارای دو سنسور نوری است که دو طرف خط مشکی در زمینه سفید قرار می گیرند (تعداد سنسورها به طریقی که خواهیم دید می توان بیش از دو عدد باشد). دو موتور نیز وظیفه به حرکت درآوردن دو چرخ روبات را برعهده دارند. شکل چند روبات تعقیب خط را ببینید:





هرگاه روبات در حین حرکت به انحرافی در خط مشکلی برخورد کند، یکی از سنسورهایش وارد خط می‌شود؛ مثلاً اگر خط مشکلی به راست منحرف شود، سنسور سمت راست روبات وارد خط مشکلی می‌شود. روبات باید با توجه به سنسوری که وارد خط شده جهت انحراف خط را متوجه شود و به همان جهت بپیچد. ساده‌ترین راه برای پیچیدن به یک سمت، خاموش کردن موتور همان سمت و ادامه کار موتور سمت مخالف است.



برای پیچیدن به شیوه حرفه‌ای‌تر می‌توانید به جای خاموش کردن موتور سمت موافق، آن را در جهت معکوس بچرخانید. با این کار

روبات در حین پیچیدن روی زمین کشیده نمی‌شود و می‌تواند درجا بچرخد. چرخیدن روبات باید تا جایی انجام شود که سنسوری که وارد خط شده بود، از خط خارج شود؛ یعنی مجدداً تمام سنسورها خارج خط قرار گیرند تا روبات بتواند به حرکت عادی خود ادامه دهد.

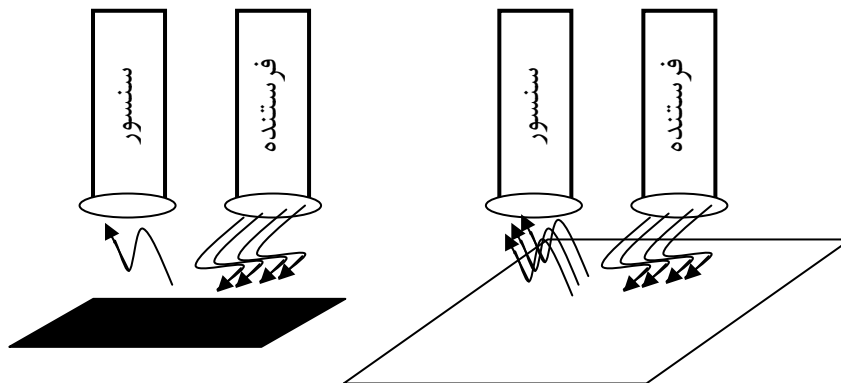
در ذیل اجزای یک روبات تعقیب خط را معرفی می‌کنیم. فراموش نکنید مطالب روی کاغذ هیچگاه به یک روبات تبدیل نمی‌شوند! هر بخش را که آموختید، در آزمایشگاه خودتان آن را بسازید.

## (۱) سنسورهای روبات تعقیب خط

همان‌گونه که گفته شد، روبات تعقیب خط باید از طریق سنسورهایش خط را حس کند و انحراف آن را تشخیص دهد. برای این کار به یک سنسور نوری نیاز داریم.

بلوک دیاگرام سنسور نوری به شکل روبرو است:

یک فرستنده نوری و یک گیرنده نوری را زیر روبات کنار هم قرار می‌دهیم؛ به صورتی که گیرنده مستقیماً در معرض نور ارسالی از فرستنده نباشد و تنها انعکاس آن از سطح مقابل را دریافت کند. از آنجایی که مقدار نور بازگشتی از سطح مشکی کمتر از مقدار نور بازگشتی از سطح سفید است، به کمک تحلیل مقدار نور دریافتی از سطح، می‌توان مشکی و سفید بودن سطح را تشخیص داد. شکل زیر را ببینید:

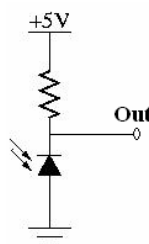


در ذیل سه نوع سنسور نوری مختلف را بررسی می‌کنیم

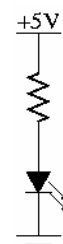
### الف) دیود مادون قرمز

فرستنده و گیرنده مادون قرمز دو دیود کاملاً مشابه هستند. یکی از آنها امواج مادون قرمز را از خود ساطع و دیگری دریافت می‌کند.

مدار گیرنده مادون قرمز را ببینید:



مدار فرستنده مادون قرمز به شکل زیر است:



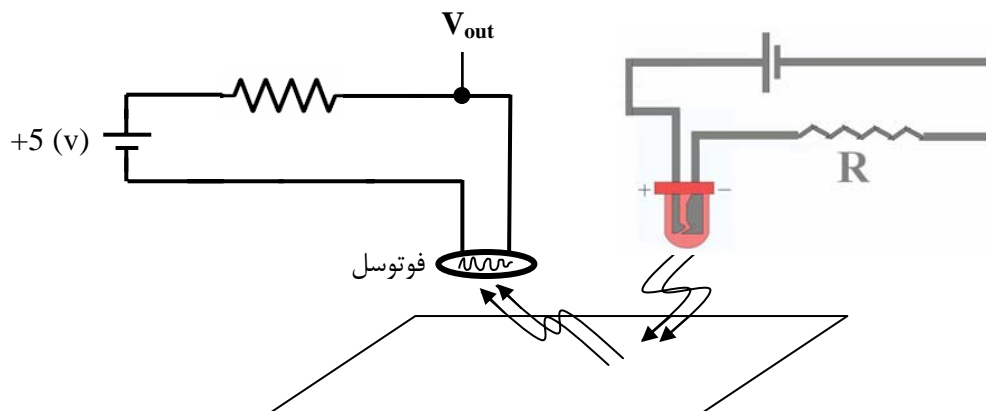


توجه کنید که دیود گیرنده در گرایش معکوس بسته شده است؛ بنابراین اگر میزان نور مادون قرمز که به آن می‌تابد کم باشد (مثلاً هنگام بازتابش از سطح مشکی)، دیود هدایت نمی‌کند و ولتاژ Out بالا خواهد بود. اگر نور مادون قرمز به میزان زیاد به دیود گیرنده بتابد، هدایت می‌کند و ولتاژ Out پایین می‌رود. بنابراین اگر برای تشخیص خط مشکی در زمینه سفید در روبات تعقیب خط از دیودهای مادون قرمز استفاده کنید، در حالتی که فرستنده/گیرنده مقابل سطح سفید قرار دارد، ولتاژ Out پایین‌تر از وقتی است که فرستنده/گیرنده وارد خط شده است.

### ب) مقاومت متغیر با نور (فوتو-رزیستور یا فوتوسل)

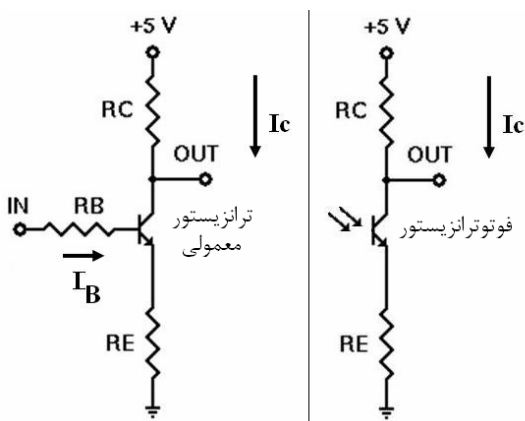
مقاومت‌های معمولی تحت شرایط مختلف (جریان، توان، دما، فشار، رطوبت، نور، ...) مقاومت تقریباً یکسانی دارند (البته وقتی این شرایط در بازه‌های مجاز مربوط به آن مقاومت تغییر کنند). اما مقاومت‌هایی نیز وجود دارند که با تغییر شرایط، مقاومت متغیری از خود بروز می‌دهند؛ مثلاً ترمیستور یک مقاومت متغیر با دما است و با تغییر دمای محیط مقدار مقاومت آن تغییر می‌کند. از ترمیستور به عنوان سنسور دما می‌توان استفاده کرد.

فوتوسل (PhotoCell) مقاومت متغیر با نور معمولی است. مقدار این مقاومت بسته به مقدار نوری که به آن می‌تابد تغییر می‌کند. برای استفاده از این سنسور در روبات تعقیب خط کافی است آن را همراه با یک فرستنده نوری معمولی مثلاً یک LED به کار برید. بهتر است رنگ LED را طوری انتخاب کنید که بیشترین درخشش را داشته باشد (مثلاً نارنجی تند). مدار زیر را ببینید:



بسته به مقدار نور انعکاسی از سطح (که به سفید یا مشکی بودن آن بستگی دارد)، ولتاژ  $V_{out}$  تغییر می‌کند. به بیان دیگر، ولتاژ  $V_{out}$  با توجه به مقدار مقاومت فوتوسل و مقاومت سری شده با آن، در دو سطح مشکی و سفید دو مقدار متفاوت دارد. از روی مقدار این ولتاژ، روبات می‌تواند خط را تشخیص دهد.

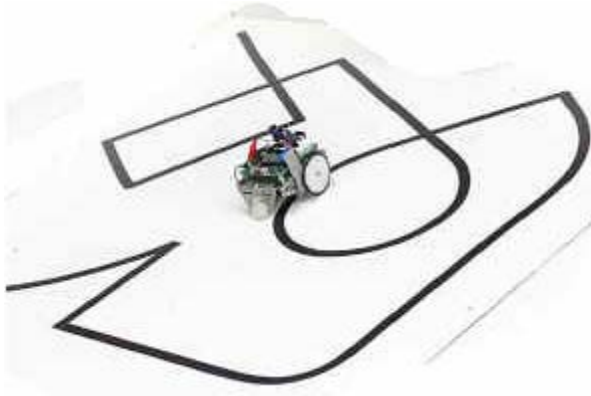
### ج) فوتو-ترانزیستور



در ترانزیستورهای معمولی دوپیوندی (BJT)، مقدار جریان  $I_c$  که از کلکتور به امیتر جریان دارد، بسته به جریان  $I_b$  که به بیس آنها وارد می‌شود تغییر می‌کند. بنابراین با یک پیکربندی مناسب می‌توان با تغییر ولتاژ (جریان) بیس، ولتاژ کلکتور را تغییر داد. در فوتو-ترانزیستور، آنچه باعث تغییر جریان کلکتور به امیتر می‌شود، نه جریان بیس، بلکه نوری است که به بیس می‌تابد. بنابراین بسته به نوری که به بیس فوتو-ترانزیستور می‌تابد (که با

انعکاس از دو سطح سفید و مشکی متفاوت است)، ولتاژ کلکتور تغییر می‌کند. این نور بسته به نوع فوتو-ترانزیستور، می‌تواند نور معمولی یا مادون قرمز باشد. مدار فرستنده-گیرنده فوتو-ترانزیستور به مدار فوتوسل شباهت زیادی دارد.

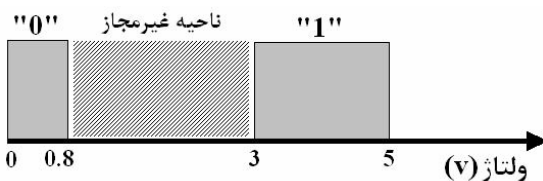
## تعداد سنسورها



اگر خط تنها پیچ و انحراف داشته باشد، دو سنسور که دقیقاً دو سمت خط قرار گرفته باشد، کافی است. اما معمولاً در مسابقات رسمی، خط دارای بریدگیها و تقاطع‌هایی با زوایا و اندازه‌های مختلف است. اگر تنها از دو سنسور استفاده کنید، ممکن است خط را گم کنید. بنابراین اگر چند سنسور در فواصل و زوایای مختلف تعبیه کنید، مفید خواهد بود. برای تجربه اول بهتر است از دو سنسور استفاده کنید و بعدها با اضافه کردن سنسورهای بیشتر، عملکرد روبات را ارتقا ببخشید.

## ۲) تبدیل و انتقال خروجی سنسور به مدار پردازشگر

خروجی هر کدام از مدارات سنسوری بخش قبل، یک ولتاژ آنالوگ (پیوسته) بین صفر تا ۵ ولت است که بسته به اینکه سنسور در محیط سفید یا مشکی باشد، مقادیر مختلفی خواهد داشت. مثلاً فرض کنید خروجی یک سنسور نوری در محیط کاملاً سفید حدود  $1/2$  ولت است و با ورود کامل آن سنسور به خط مشکی خروجی سنسور  $3/8$  ولت می‌شود (تلاش نکنید سنسور خود را طوری بسازید که ولتاژ خروجی آن دقیقاً منطبق بر این مقادیر باشد! این تنها مثالی برای درک شما از موضوع است). مدار پردازشگر باید بر اساس این دو مقدار واکنش مناسب را نشان دهد و روبات را هدایت کند. اما مدارهای پردازشگر (چه مدار منطقی و چه میکروکنترلر) تنها مقادیر دیجیتال را می‌شناسند. یعنی باید با آنها به زبان «صفر» و «یک» صحبت کرد. در منطق ولتاژهای دیجیتال، از ولتاژ بین صفر تا حدود  $0/8$  ولت به «صفر منطقی» و از ولتاژ بین ۳ تا ۵ ولت به «یک منطقی» تعبیر می‌شود.



بنابراین اگر مثلاً خروجی یک سنسور نوری در محیط کاملاً سفید  $1/2$  ولت و در محیط کاملاً مشکی  $3/8$  ولت باشد، یک مدار منطقی یا میکروکنترلر هیچ درکی از آن نخواهد داشت. بلکه این ولتاژها باید به مقادیر دیجیتال تبدیل شوند تا قابل درک توسط مدار پردازشگر باشند.

برای تبدیل خروجی آنالوگ سنسور به مقدار دیجیتال دو راه وجود دارد:

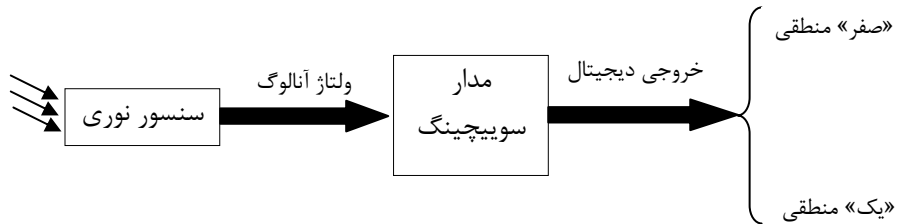
به کمک یک مدار سویچینگ شما می‌توانید از روی خروجی آنالوگ سنسور، «بودن» یا «نبودن» سنسور در خط را مشخص کنید.

اگر از مدار مبدل آنالوگ به دیجیتال (A/D) استفاده کنید، می‌توانید درصد وارد شدن سنسور به خط را نیز تشخیص دهید.

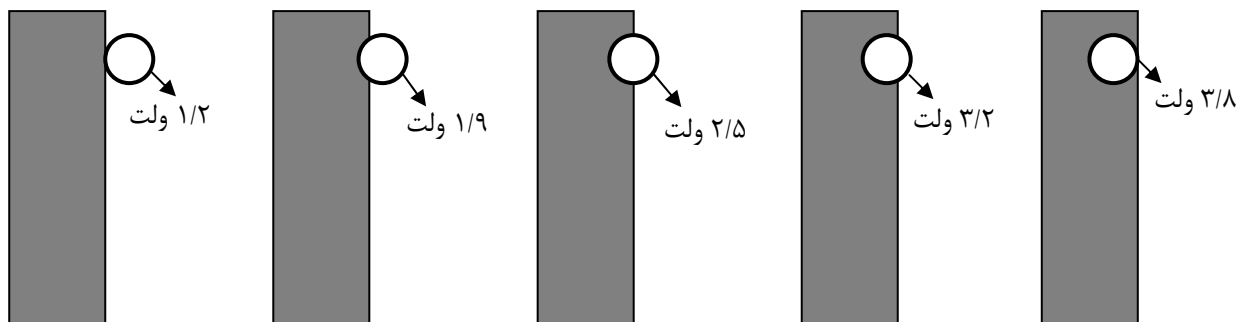
### مدار سویچینگ

به کمک یک مدار الکترونیکی سویچینگ می‌توان ولتاژ آنالوگ خروجی سنسور را به یک ولتاژ دیجیتال «صفر» یا «یک» قابل درک توسط مدار منطقی یا میکروکنترلر تبدیل کرد. به زبان ساده مداری می‌خواهیم که مثلاً وقتی سنسور در منطقه سفید است «صفر منطقی» و وقتی وارد منطقه مشکی می‌شود، «یک منطقی» را به عنوان خروجی تولید کند. با این کار، مدار پردازشگر از روی این ولتاژ متوجه «ورود» یا «عدم ورود» روبات به خط مشکی می‌شود.

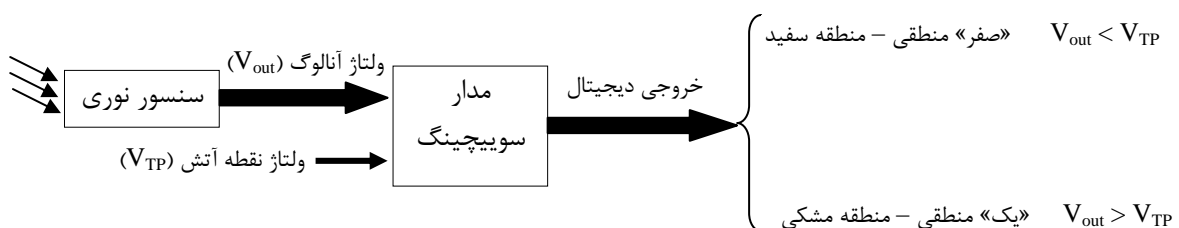
بلوک دیاگرام مدار سویچینگ به شکل زیر است:



به یک نکته مهم توجه کنید. فرض کنید هنگامی که سنسور کاملاً از خط مشکی خارج است (در محیط کاملاً سفید است) دارای خروجی  $1/2$  ولت می‌باشد و با ورود کامل به خط مشکی دارای ولتاژ  $3/8$  ولت می‌شود. می‌خواهیم این ولتاژهای آنالوگ را به کمک یک مدار سویچینگ به گونه‌ای به یک ولتاژ دیجیتال تبدیل کنیم که وقتی سنسور در منطقه سفید است «صفر منطقی» و وقتی وارد منطقه مشکی می‌شود، «یک منطقی» را به عنوان خروجی داشته باشیم. هنگامی که سنسور از زمینه سفید وارد خط مشکی می‌شود، خروجی سنسور به مرور از  $1/2$  ولت تا  $3/8$  ولت تغییر می‌کند.



به نظر منطقی می‌آید روبات باید طوری طراحی شود که با رسیدن ولتاژ خروجی سنسور به میانه این بازه (مثلاً  $2/5$  ولت) با توجه به اینکه سنسور کم و بیش وارد خط شده، خروجی مدار سویچینگ از «صفر» به «یک» تغییر کرده و جهت حرکت روبات تغییر کند. به این ولتاژ که با گذر از آن عمل سویچینگ اتفاق می‌افتد، اصطلاحاً نقطه آتش (Trigger Point) گفته می‌شود.



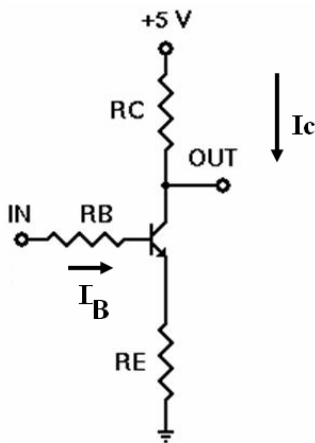
مقدار ولتاژ نقطه آتش چقدر باید باشد؟ به بیان دیگر ولتاژ خروجی سنسور باید به چه حدی برسد تا نتیجه بگیریم سنسور وارد خط شده و جهت حرکت روبات را اصلاح کنیم؟  $1/5$  ولت کم نیست؟  $2/5$  ولت بهتر نیست؟ اصلاً بهتر نیست صبر کنیم سنسور کاملاً وارد خط شود و ولتاژ خروجی آن به  $3/8$  ولت برسد و بعد جهت حرکت روبات را تغییر دهیم؟

واقعیت این است که پاسخ به این سؤال به مکانیک و نیز مدار کنترلی روبات شما بستگی دارد. در واقع باید ببینید دقت چرخش روبات شما در چه حد است؟ اگر می‌تواند بسیار سریع و دقیق بچرخد، نیازی نیست خیلی زودتر از ورود کامل سنسور به خط به روبات فرمان چرخش داده شود. اما اگر چرخش روبات کند و همراه با خطاست، شاید بهتر باشد زمانی که نیمی از سنسور وارد خط شده فرمان چرخش بدهیم تا روبات در زمان مناسب بتواند بچرخد.

اکنون می‌خواهیم یک مدار سویچینگ بسازیم که عبور ولتاژ خروجی سنسور از ولتاژ نقطه آتش را تشخیص دهد؛ به بیان دیگر باید مداری طراحی کنیم که مثلاً در ولتاژ ورودی کمتر از ولتاژ نقطه آتش، خروجی «صفر» (یعنی ولتاژی بین صفر تا  $0/8$  ولت) و در ولتاژ ورودی بیشتر از ولتاژ نقطه آتش، خروجی «یک» (ولتاژی بین  $3$  تا  $5$  ولت) بدهد. یک مدار منطقی یا میکروکنترلر می‌تواند از روی این خروجی «صفر» یا «یک»، متوجه «ورود» یا «عدم ورود» روبات به خط مشکی بشود.

ساخت مدار سویچینگ راههای مختلفی دارد.

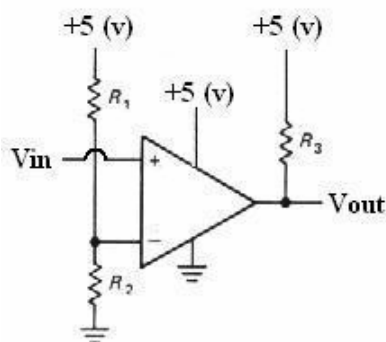
### مدار ترانزیستوری



مدار روبرو را ببینید. مقاومتها باید به نحوی انتخاب شوند که ولتاژ خروجی ترانزیستور (ولتاژ کلکتور) در حالت‌های قطع و اشباع برابر ولتاژهای دیجیتال (حدود صفر و حدود  $5$  ولت) باشد. اگر ولتاژ ورودی  $I_N$  کم باشد، ترانزیستور در حالت قطع بوده و ولتاژ خروجی آن (ولتاژ کلکتور OUT) «یک» خواهد بود. اگر ولتاژ ورودی ترانزیستور از حدی بالاتر برود، ترانزیستور سویچ کرده و وصل می‌شود و خروجی آن (ولتاژ کلکتور) از «یک» به «صفر» تغییر می‌کند. مدار منطقی یا میکروکنترلر از روی ولتاژ کلکتور متوجه ورود یا عدم ورود روبات به خط مشکی شده و تصمیم‌گیری لازم را انجام می‌دهد. برای مدارهای سویچینگ استفاده از ترانزیستورهای 2N2222 یا 2N3904 یا ترانزیستورهای مشابه مناسب است. این مدار بیشتر در

مواقعی استفاده می‌شود که ورودی به سرعت تغییر کند؛ وگرنه ممکن است خروجی OUT در منطقه غیرمجاز (نه «صفر» و نه «یک») قرار گیرد.

### مدار سویچ با آپ‌امپ



به کمک تراشه‌های مقایسه‌کننده که عموماً از نوع آپ‌امپ هستند و بدون فیدبک استفاده می‌شوند، نیز می‌توان عمل سویچینگ را انجام داد. مدار روبرو را ببینید.

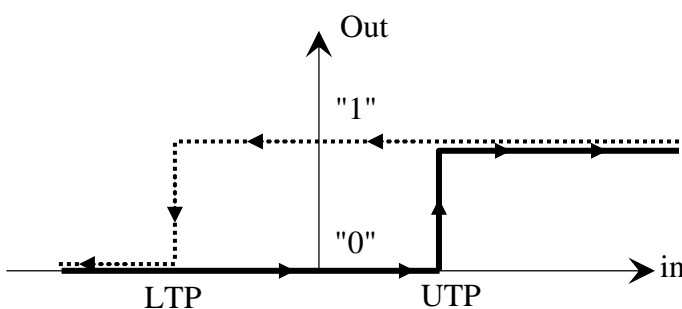
تراشه LM311 یکی از اعضای خانواده آپ‌امپ‌های مقایسه‌کننده است که به صورت بدون فیدبک برای انجام مقایسه به کار می‌رود. عملکرد آپ‌امپ بدون فیدبک به صورت ساده به این صورت است که اگر ولتاژ پایه ورودی

مثبت از پایه ورودی منفی بیشتر شود، خروجی آپ-امپ به اشباع مثبت رفته و «یک» می‌شود. اگر ولتاژ پایه ورودی منفی از پایه ورودی مثبت بیشتر شود، خروجی آپ-امپ به اشباع منفی رفته و «صفر» می‌شود. در مدار بالا، اگر خروجی سنسور ( $V_{in}$ ) که به ورودی مثبت مقایسه‌کننده متصل است، از ولتاژ مرجع (ولتاژ نقطه آتش) که به پایه منفی مقایسه‌کننده متصل است بیشتر شود، مقایسه‌کننده به اشباع مثبت می‌رود و خروجی آن ( $V_{out}$ ) «یک» می‌شود. در غیر این صورت در اشباع منفی قرار داشته و خروجی آن «صفر» است. مثلاً در مثال قبلی که خروجی سنسور نوری در محیط کاملاً سفید حدود  $1/2$  ولت و در محیط کاملاً مشکی حدود  $3/8$  ولت است، فرض کنید ولتاژ نقطه آتش را  $2/5$  ولت در نظر گرفته و به کمک تقسیم ولتاژ روی مقاومتهای  $R_1$  و  $R_2$  این ولتاژ را ایجاد و به پایه منفی آپ-امپ متصل می‌کنیم. اگر ولتاژ خروجی سنسور کمتر از  $2/5$  ولت باشد (زمینه سفید)، ولتاژ ورودی مثبت مقایسه‌کننده کمتر از ولتاژ ورودی منفی آن است و خروجی مقایسه‌کننده «صفر» می‌شود. با ورود سنسور به خط مشکی، ولتاژ ورودی مثبت مقایسه‌کننده بیشتر از ولتاژ ورودی منفی آن شده و خروجی مقایسه‌کننده «یک» می‌شود. مدار منطقی یا میکروکنترلر می‌تواند بر اساس خروجی مقایسه‌کننده عمل سویچ را انجام دهد.

### مدار اشمیت‌تریگر

مدارهای سویچینگ بالا با عبور از یک ولتاژ نقطه آتش سویچ می‌کنند. این مسأله ممکن است باعث دردسر شود. مثال قبلی را در نظر بگیرید (خروجی سنسور نوری در محیط کاملاً سفید حدود  $1/2$  ولت و در محیط کاملاً مشکی حدود  $3/8$  ولت است و ولتاژ نقطه آتش را  $2/5$  ولت در نظر گرفته‌ایم). فرض کنید سنسور در حال ورود به خط مشکی است؛ بنابراین ولتاژ خروجی آن بالا رفته و سرانجام در حالی که حدود نیمی از سنسور وارد خط شده است،  $2/5$  ولت می‌گذرد. خروجی مدار سویچینگ تغییر کرده و به قسمت کنترلی اعلام می‌کند سنسور وارد خط شده است. روبات تغییر جهت داده و می‌چرخد تا سنسور را از خط خارج کند. همین که بیش از نیمی از سنسور از خط خارج می‌شود، ولتاژ خروجی آن از  $2/5$  ولت کمتر شده و خروجی مدار سویچینگ به حالت قبل باز می‌گردد و به قسمت کنترلی اعلام می‌کند سنسور از خط خارج شده است. به محض اینکه روبات کمی از جای خود حرکت کند، مجدداً خروجی سنسور از  $2/5$  ولت بیشتر شده و مدار سویچ می‌کند، باز سنسور از خط خارج می‌شود و مجدداً مدار سویچ می‌کند و ... مشاهده می‌کنید که با حرکت سنسور در لبه خط که ولتاژ خروجی آن در حدود ولتاژ  $2/5$  ولت نوسان می‌کند، مرتباً مدار سویچ از «یک» به «صفر» و برعکس تغییر می‌کند و باعث می‌شود هیچ‌گاه روبات به درستی تنظیم نشود.

برای حل این مشکل می‌توان به جای یک نقطه آتش، دو نقطه آتش بالا (UTP: Upper Trigger Point) و نقطه آتش پایین (LTP) در نظر گرفت. در مثال قبلی، به جای یک ولتاژ آتش  $2/5$  ولت، یک ولتاژ آتش پایین به میزان  $2$  ولت و یک ولتاژ آتش بالا به میزان  $3$  ولت در نظر می‌گیریم. مداری طراحی می‌کنیم که وقتی ولتاژ ورودی آن از

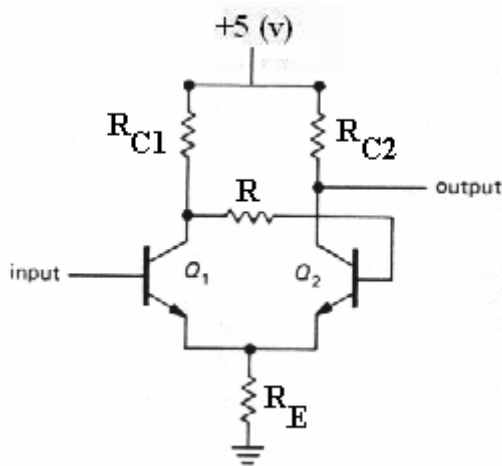


ولتاژ آتش بالا بیشتر شد، خروجی آن «یک» و وقتی ولتاژ ورودی آن از ولتاژ آتش پایین کمتر شد، خروجی آن «صفر» شود. به این ترتیب مشکل قبلی حل می‌شود و دیگر مدار سویچ پیرامون یک ولتاژ آتش مرتباً سویچ نمی‌کند. به تکنیک فوق، ایجاد هیستریزس

(Hysteresis) یا حافظه گفته می‌شود.

با استفاده از مدار دارای هیستریزیس با  $UTP = 3 (V)$  و  $LTP = 2 (V)$ ، وقتی سنسور وارد خط می‌شود، مدار سویچ بلافاصله واکنش نشان نمی‌دهد، بلکه صبر می‌کند تا سنسور قدری بیشتر وارد خط شود و ولتاژ خروجی آن از ۳ ولت بیشتر شود. در این حالت به روبات فرمان چرخش داده می‌شود. هنگامی که کمی از سنسور از خط خارج شد، باز مدار سویچ بلافاصله دستور قطع چرخش نمی‌دهد؛ بلکه صبر می‌کند تا حد زیادی از خط خارج شود و ولتاژ خروجی آن از ۲ ولت کمتر شود و بعد فرمان قطع چرخش می‌دهد.

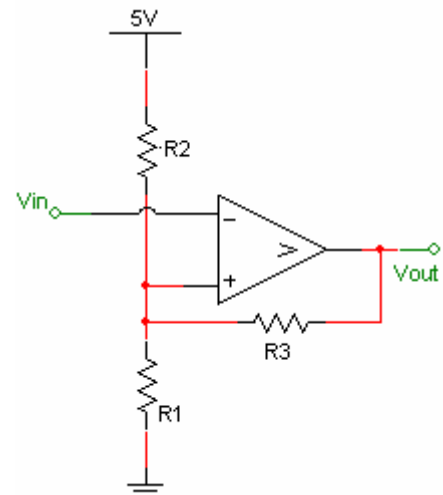
از معروفترین مدارات ایجاد هیستریزیس، مدار اشمیت تریگر (Schmitt Trigger) است که شکل ترانزیستوری آن را در شکل روبرو می‌بینید.



با تنظیم مقادیر مقاومتها می‌توانید ولتاژ نقطه آتش بالا و نقطه آتش پایین را تنظیم کنید.

مدار اشمیت تریگر را با آپ-امپ با فیدبک مثبت نیز می‌توان ساخت. توجه کنید عملکرد این مدار عکس مدار اشمیت تریگر ترانزیستوری است؛ یعنی با بیشتر شدن ولتاژ ورودی از  $UTP$  خروجی «صفر» و با کمتر شدن ولتاژ ورودی از  $LTP$  خروجی «یک» می‌شود.

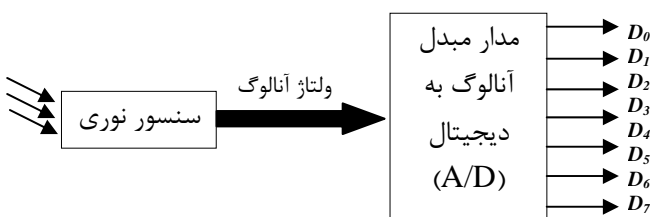
تراشه 74LS14 مدار اشمیت تریگر آماده را (البته با تنظیمات مفصل) ارائه می‌کند.



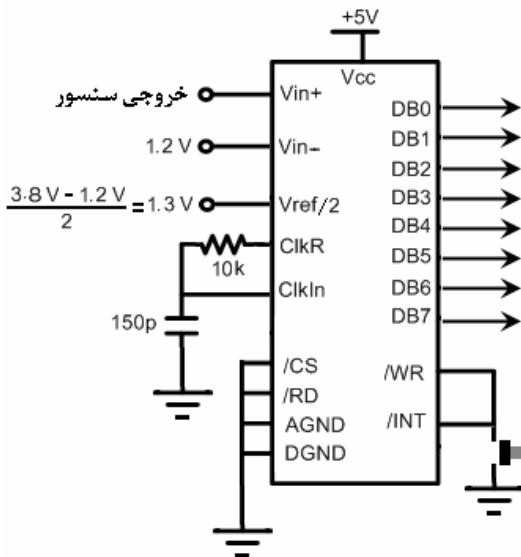
### تبدیل خروجی آنالوگ سنسور به ولتاژ دیجیتال به کمک A/D

مدارات سویچینگ، ولتاژ خروجی سنسور را به یک ولتاژ  $V_{out}$  (با دو حالت «صفر» یا «یک») تبدیل می‌کنند؛ به بیان دیگر، به کمک خروجی دوحالتی یک مدار سویچینگ می‌توان تنها دانست آیا روبات وارد خط شده است یا خیر؟ این در حالی است که ولتاژ خروجی سنسور از وقتی در زمینه سفید است تا زمانی که کاملاً وارد خط مشکی شود، به مرور تغییر می‌کند. بنابراین اگر بتوانیم ولتاژ خروجی سنسور را نه به صورت یک رقم منطقی «صفر» یا

«یک»، که مثلاً به کمک یک عدد ۸ بیتی بیان کنیم، آنگاه می‌توان به کمک ۲۵۶ عدد مختلفی که می‌توان با ۸ بیت ساخت، مقدار ورود سنسور به خط را نیز با دقت زیاد با عددی بین صفر تا ۲۵۵ مشخص کرد. مدار مبدل آنالوگ به دیجیتال این کار را انجام می‌دهد.



مثال قبلی را (خروجی سنسور نوری در محیط کاملاً سفید حدود ۱/۲ ولت و در محیط کاملاً مشکی حدود ۳/۸ ولت است) مجدداً در نظر بگیرید؛ به کمک یک A/D مداری طراحی می‌کنیم که اگر خروجی سنسور ۱/۲ ولت باشد خروجی A/D برابر صفر و اگر خروجی سنسور ۳/۸ ولت باشد خروجی A/D برابر ۲۵۵ شود؛ در این حالت می‌توان بر اساس عدد خروجی ۸ بیتی A/D، مقدار ورود سنسور به خط را دانست و براساس آن تصمیم لازم را اتخاذ کرد. مدار تراشه ADC804 را در شکل بعد می‌بینید.

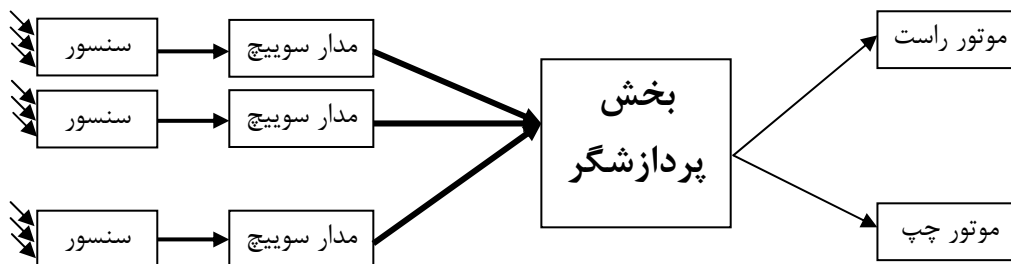


این مدار مرتباً ولتاژ  $V_{in(+)} - V_{in(-)}$  را به عدد ۸ بیتی تبدیل کرده و روی پایه‌های DB0 تا DB7 نمایش می‌دهد. برای مثال بالا، خروجی سنسور را به ورودی  $V_{in(+)}$  متصل کرده و به کمک یک پتانسیومتر، ولتاژ ۱/۲ ولت را ساخته و به  $V_{in(-)}$  متصل می‌کنیم. در این حالت اگر سنسور در محیط کاملاً سفید باشد (خروجی سنسور ۱/۲ ولت باشد)، خروجی A/D صفر ( $DB7-DB0 = 00000000$ ) و اگر سنسور در محیط کاملاً مشکی باشد (خروجی سنسور ۳/۸ ولت باشد) خروجی A/D برابر ۲۵۵ ( $DB7-DB0 = 11111111$ ) می‌شود. با اتصال  $DB7-DB0$  به مدار کنترلی (مثلاً میکروکنترلر)، می‌توان براساس عدد ۸ بیتی خروجی A/D تصمیم‌گیری لازم برای

حرکت روبات را انجام داد. مثلاً می‌توان خروجی ۱۲۷ ( $DB7-DB0 = 01111111 = 255/2$ ) را به عنوان نقطه آتش در نظر گرفت؛ یعنی هرگاه میکروکنترلر عدد ۱۲۷ را از A/D دریافت کند، می‌تواند نتیجه بگیرد که نیمی از سنسور وارد خط شده و عمل تغییر جهت روبات را انجام دهد. اگر بخواهید هیستریزیس را در سویچ رعایت کنید، می‌توانید دو نقطه آتش (مثلاً UTP برابر ۲۰۰ و LTP برابر ۱۰۰) در نظر بگیرید و برنامه میکروکنترلر را طوری بنویسید که این نقاط آتش رعایت شوند.

### ۳) بخش پردازشگر

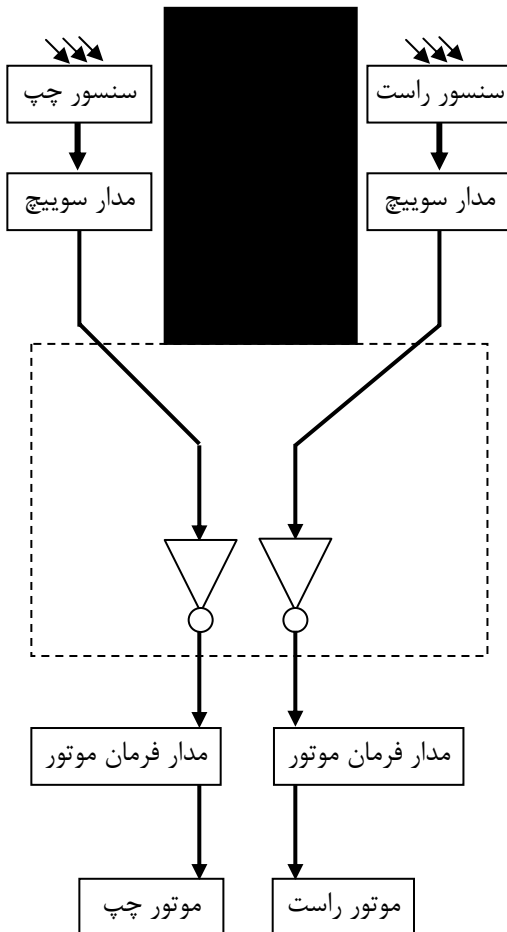
بلوک دیاگرام بخش پردازشگر به شکل زیر است:



همانطور که مشاهده می‌کنید، بخش پردازشگر روبات که یک مدار منطقی یا میکروکنترلر است، بر حسب سیگنالهای منطقی «صفر» و «یک» که از سنسورها و مدارهای سویچینگ دریافت می‌کند، باید تصمیمی مناسب برای حرکت مستقیم یا پیچیدن روبات به جهت مناسب را اتخاذ نماید و به کمک فرمانهایی که به موتورهای چپ و راست می‌دهد، تصمیم خود را عملی کند.

### مدار منطقی

بخش پردازشگر روبات تعقیب خط می‌تواند به سادگی یک مدار منطقی با گیت‌های NOT باشد. مثلاً فرض کنید تنها از دو سنسور نوری در سمت چپ و راست خط استفاده کرده‌ایم. همچنین فرض کنید مدارات سویچینگ به صورتی طراحی شده‌اند که اگر سنسور در زمینه سفید باشد خروجی مدار سویچینگ «صفر» و با ورود سنسور به خط مشکی خروجی مدار سویچینگ «یک» می‌شود. فرض دیگر



این است که برای روشن کردن یک موتور باید به مدار کنترلی آن موتور «یک» و برای خاموش کردن آن باید به مدار کنترلی آن «صفر» ارسال شود. مدار کنترلی روبات می‌تواند به شکل روبرو باشد.

این مدار بر اساس این واقعیت طراحی شده که اگر سنسور سمت راست وارد خط شود، به معنای انحراف خط به سمت راست است؛ بنابراین روبات باید به سمت راست بپیچد. برای این کار باید موتور سمت راست خاموش شده و موتور سمت چپ به کار خود ادامه دهد. یعنی اگر خروجی سنسور راست «یک» شود، باید به مدار کنترل موتور آن «صفر» ارسال شود. با خارج شدن سنسور سمت راست از خط (خروجی سنسور «صفر» شود)، موتور سمت راست باید مجدداً روشن شود (به مدار کنترلی آن «یک» ارسال شود). روند پیچیدن روبات به سمت چپ نیز به همین شکل خواهد بود.

به نظر بیش از حد ساده می‌آید؟! این مدار کنترلی عملاً پاسخ می‌دهد؛ البته مشروط به اینکه مکانیک و سرعت پیچیدن روبات شما در حد مطلوب باشد. توجه کنید اگر مدار کنترل موتور طوری

طراحی شده باشد که لازم باشد برای روشن کردن موتور «صفر» و برای خاموش کردن آن «یک» به آن ارسال کنیم، حتی به آن گیت‌های NOT نیز نیازی نیست! در مورد نحوه طراحی مدار کنترل موتور بعداً صحبت خواهیم کرد.

### میکروکنترلر

اگر ضعفی در بخش‌های دیگر روبات وجود داشته باشد، شاید بهتر باشد برای جبران آن ضعفها و برنامه‌ریزی دقیقتر بخش پردازشگر از یک میکروکنترلر استفاده کنیم. به علاوه گاهی برای افزایش دقت عملکرد روبات از بیش از دو سنسور نوری استفاده می‌شود (مثلاً دو سنسور در عقب و دو سنسور در جلوی روبات یا دو سنسور در سمت راست و دو سنسور در سمت چپ خط). در این حالت اگر بخواهید برای بخش پردازشگر از مدار منطقی استفاده کنید، باید جدول درستی مدار فوق را رسم کرده و ساده کنید و سپس براساس آن مدار را طراحی کنید. شاید استفاده از میکروکنترلر در این مواقع کار ساده‌تری باشد.

مانند قبل فرض کنید تنها از دو سنسور نوری در سمت چپ و راست خط استفاده کرده‌ایم. همچنین فرض کنید مدارات سویچینگ به صورتی طراحی شده‌اند که اگر سنسور در زمینه سفید باشد خروجی مدار سویچینگ



«صفر» و با ورود سنسور به خط مشکی خروجی مدار سویچینگ «یک» می‌شود. خروجی مدارهای سویچینگ سنسورهای سمت چپ و سمت راست روبات را به دو پایه میکروکنترلر متصل می‌کنیم. شبه‌کد برنامه میکروکنترلر به صورت زیر است:

Back:

```
// If both sensors are in white background, the robot should go straight.
```

```
If (RightSensor = 0 AND LeftSensor = 0) then
```

```
    Turn on Right Motor;
```

```
    Turn on Left Motor;
```

```
// If the right sensor enters black line, the robot should turn right.
```

```
If (RightSensor = 1 AND LeftSensor = 0) then
```

```
    Turn off Right Motor;
```

```
    Turn on Left Motor;
```

```
// If the left sensor enters black line, the robot should turn left.
```

```
If (RightSensor = 0 AND LeftSensor = 1) then
```

```
    Turn on Right Motor;
```

```
    Turn off Left Motor;
```

Loop Back

همانطور که می‌بینید، این شبه‌کد به قدری ساده است که شاید پیاده‌سازی آن با یک مدار منطقی معقول‌تر باشد. اما برای انجام پردازش‌های پیچیده‌تر استفاده از میکروکنترلر مناسب‌تر است. اگر برای تبدیل ولتاژ آنالوگ خروجی سنسور به ولتاژ دیجیتال قابل فهم بخش پردازشگر از A/D استفاده کنید، به دلیل اینکه خروجی A/D هشت بیت است (عددی بین صفر تا ۲۵۵ که نمادی از ولتاژ خروجی آنالوگ سنسور است) و بخش پردازشگر باید بر اساس این هشت بیت تصمیم بگیرد، استفاده از مدار منطقی در بخش پردازشگر معقول نیست و باید از میکروکنترلر استفاده شود. مثلاً فرض کنید A/D مربوط به سنسور سمت راست را به پورت P1 میکروکنترلر و A/D مربوط به سنسور سمت چپ را به پورت P2 میکروکنترلر متصل نموده‌ایم. ورود نیمی از سنسور به خط باعث می‌شود خروجی A/D بیش از ۱۲۷ ( $DB7 \sim DB0 = 01111111 = 255/2$ ) شود. شبه‌کد زیر را ببینید:

Back:

```
// If both sensors are in white background, the robot should go straight.
```

```
If (P1 < 127 AND P2 < 127) then
```

```
    Turn on Right Motor;
```

```
    Turn on Left Motor;
```

```
// If the right sensor enters black line, the robot should turn right.
```

```
If (P1 > 127 AND P2 < 127) then
```

```
    Turn off Right Motor;
```

```
    Turn on Left Motor;
```

```
// If the left sensor enters black line, the robot should turn left.
```

```
If (P1 < 127 AND P2 > 127) then
```

```
    Turn on Right Motor;
```

```
    Turn off Left Motor;
```

Loop Back

## ۴) موتور و مدارهای کنترل آن

مدار پردازشگر روبات (چه مدار منطقی و چه میکروکنترلر) باید به موتورها فرمان دهد. این فرمان بر حسب نوع موتور متفاوت خواهد بود که در ذیل به شرح انواع آن خواهیم پرداخت.

قبل از آن به نکته‌ای توجه کنید. هنگامی که به‌عنوان مثال روبات می‌خواهد به سمت چپ بپیچد، باید موتور سمت چپ را خاموش و موتور سمت راست را روشن کنیم. در این حالت روبات در حین پیچیدن کمی به جلو نیز حرکت کرده و روی زمین کشیده می‌شود. برای اینکه روبات درجا به سمت چپ بپیچد باید موتور سمت چپ رو به عقب و موتور سمت راست رو به جلو بچرخد. به همین دلیل ضمن بررسی مدارهای کنترل موتور، نحوه معکوس کردن جهت چرخش موتور را نیز بررسی می‌کنیم.

### آرمیچر یا موتور DC

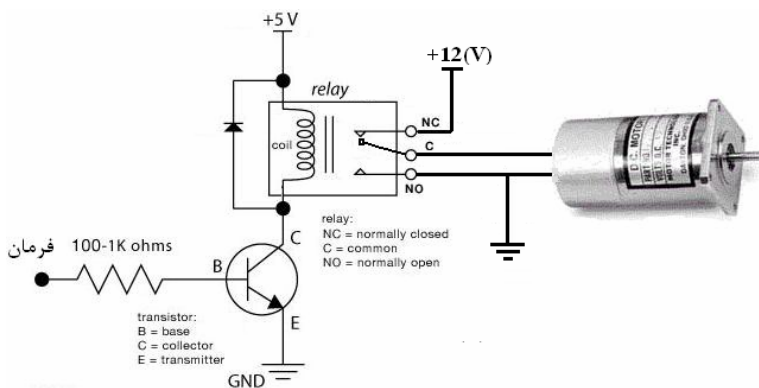
آرمیچر ساده‌ترین نوع موتور است که آن را در ضبط صوتهای کوچک یا اسباب‌بازی‌های برقی دیده‌اید. شکل روبرو یک آرمیچر را نشان می‌دهد. هنگامی که  $V_1 > V_2$  موتور به سمت راست و اگر  $V_1 < V_2$  موتور به سمت چپ می‌چرخد.



### مدار کنترلی آرمیچر

ساده‌ترین راه برای روشن و خاموش کردن آرمیچر، استفاده از یک رله است.

رله یک قطعه الکتریکی است که دارای پنج پایه می‌باشد. در حالت عادی که ولتاژی به پایه‌های بوبین (فرمان-coil) متصل نیست، پایه مشترک (Common) به پایه «معمولاً بسته» (Normally Close) متصل

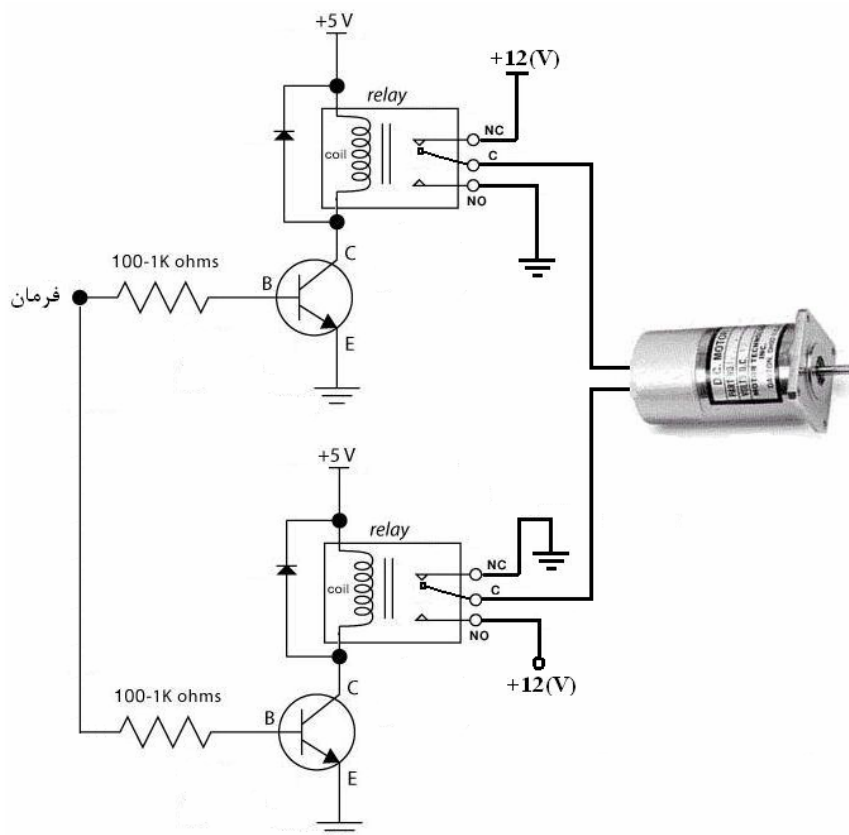


است. اگر فرمان به پایه‌های بوبین (فرمان) داده شود (مثلاً ولتاژ ۵ ولت به بیس ترانزیستور راه‌انداز رله متصل شود)، پایه مشترک به پایه «معمولاً باز» (Normally Open) متصل می‌شود. با قطع شدن فرمان، مجدداً پایه مشترک به پایه «معمولاً بسته» متصل می‌شود. برای تشخیص این پنج پایه می‌توانید از اهم‌تر استفاده کنید. معمولاً پشت رله نیز با عبارات N.O و N.C و C و رسم تصویر یک سلف در کنار پایه‌های بوبین، پایه‌ها از یکدیگر متمایز می‌شوند. هنگامی که پایه‌های فرمان را به ولتاژ متصل می‌کنید، صدای «چسبیدن» رله از داخل آن به گوش می‌رسد. چون فرمان دادن به موتور جریان زیادی می‌خواهد و ممکن است به مدار منطقی یا پایه میکروکنترلر آسیب برساند، از یک ترانزیستور برای فرمان استفاده می‌کنیم.

در مدار بالا هرگاه فرمانی که از بخش پردازشگر می‌آید «صفر» باشد، ترانزیستور و فرمان رله قطع است؛ بنابراین پایه مشترک به پایه «معمولاً بسته» وصل بوده و ولتاژ موتور متصل و موتور روشن خواهد شد. اگر فرمان خروجی بخش پردازشگر «یک» باشد، ترانزیستور و فرمان رله وصل می‌شود؛ بنابراین پایه مشترک به پایه «معمولاً باز» وصل بوده و ولتاژ موتور قطع و موتور خاموش می‌شود. بنابراین بخش پردازشگر باید به نحوی طراحی شود که در حالت عادی برای کار کردن موتور به مدار کنترلی آن «صفر» و برای خاموش کردن موتور به مدار کنترلی آن «یک» بدهد. ولتاژ آرمیچر ۱۲ یا ۲۴ ولت است.

## معکوس کردن جهت چرخش موتور

اگر بخواهیم هنگام چرخیدن روبات به یک سمت، موتور آن سمت خاموش نشود، بلکه در جهت عکس بچرخد تا چرخش درجا و دقیق و سریعی داشته باشیم، می توان از مدار زیر استفاده کرد:

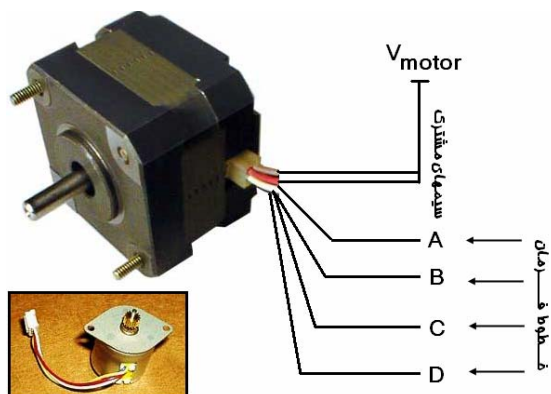


## موتور پله‌ای (Stepper Motor)

موتور DC کوچک و سرعت کارش بالاست و برای استفاده در روبات گزینه مناسبی است. اما تنظیم سرعت و دور موتور و عملکرد آن چندان ساده نیست. مواقعی که نیاز به کنترل دقیق سرعت و مقدار و جهت چرخش موتور داریم موتور پله‌ای گزینه بهتری است. ضعف موتور پله‌ای نسبت به آرمیچر، سرعت کند و حجم و وزن بالاتر آن

است. موتور پله‌ای همانطور که از اسمش برمی آید پله به پله حرکت می کند و برای چرخش هر پله باید فرمان جدیدی به آن داد؛ بنابراین سرعت و مقدار چرخش موتور را می توان به راحتی با تنظیم سرعت و تعداد ارسال فرمانها به موتور کنترل کرد. حالت هر فرمان نسبت به فرمان قبلی، جهت چرخش موتور را نیز تنظیم می کند. صدایی که از داخل فلاپی کامپیوتر می شنوید، صدای چرخش پله به پله موتور پله‌ای داخل آن است.

ساختار یک موتور پله‌ای ۶ سیمه به شکل روبرو است:



دو سیم مشترک (که معمولاً هم رنگ هستند)، به ولتاژ موتور (۵ یا ۱۲ یا ۲۴ ولت) متصل می شوند. به چهار سیم دیگر (A و B و C و D) رشته‌های «صفر» و «یک» داده می شود که با تحت تأثیر قرار دادن سیم پیچ‌های داخلی

باعث چرخش شافت موتور می‌شوند. این رشته‌های به ترتیب به صورت زیر باید به چهار سیم فرمان موتور داده شوند:

1100 , 0110 , 0011 , 1001 , 1100 , 0110 , ...

همانطور که می‌بینید چهار رشته صفر و یک اصلی وجود دارد که هر کدام با یک شیفت چرخشی به راست از رشته قبلی به دست می‌آیند. هر رشته جدیدی که به سیمهای فرمان موتور اعمال شود، باعث می‌شود شافت موتور به اندازه یک پله بچرخد. میزان چرخش یک پله روی موتور پله‌ای نوشته می‌شود (مثلاً ۱/۸ درجه). می‌بینید که کنترل این موتور بسیار ساده است:

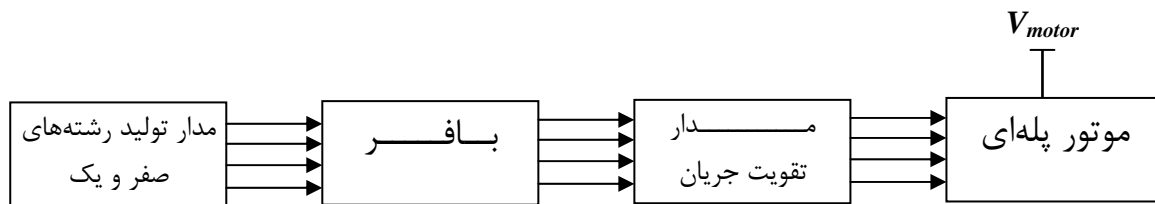
- با کنترل تعداد رشته‌های صفر و یک که به موتور پله‌ای داده می‌شود، می‌توان مقدار چرخش آن را به دقت تنظیم کرد.
- با تنظیم زمان بین ارسال رشته‌های صفر و یک، می‌توان سرعت موتور را کنترل کرد.
- اگر رشته‌های صفر و یک را با شیفت چرخشی به چپ رشته قبلی به دست آوریم، جهت چرخش موتور عکس می‌شود. رشته‌های ارسالی در این حالت عبارتند از:

1100 , 1001 , 0011 , 0110 , 1100 , 1001 , ...

توجه کنید در پاراگراف بالا منظور از «صفر» اتصال به زمین و منظور از «یک» اتصال به ولتاژ موتور است.

## مدار کنترل موتور پله‌ای

بلوک دیاگرام مدار کنترل موتور پله‌ای به صورت زیر است:



هر کدام از بلوکها را در زیر بررسی می‌کنیم

### تولید رشته‌های «صفر» و «یک»

#### میکروکنترلر

رشته‌های صفر و یک گفته شده را می‌توان به کمک میکروکنترلر به سادگی تولید کرد. کد زیر، رشته‌های صفر و یک را ایجاد کرده و به پورت P1 میکروکنترلر ۸۰۵۱ ارسال می‌کند:

```

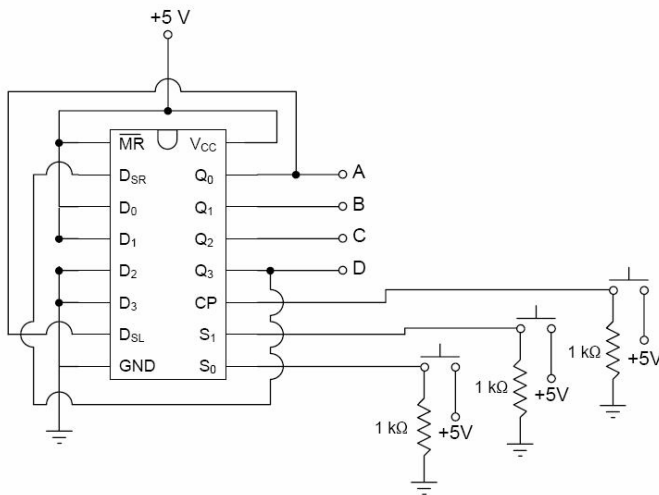
MOV  A, 11001100B
AGAIN:
    MOV  P1, A
    CALL DELAY
    RR   A
JMP  AGAIN
  
```

با دنبال کردن گام به گام این برنامه به سادگی متوجه می‌شوید که روی پینهای P1.3~P1.0 رشته‌های صفر و یک با چرخش به راست تولید می‌شوند (این بیتها روی پینهای P1.7~P1.4 نیز تولید می‌شوند). با تنظیم تأخیری که توسط زیربرنامه DELAY تولید می‌شود، می‌توانید سرعت چرخش موتور را تنظیم کنید. اگر به جای دستور RR (Rotate Right) از دستور RL (Rotate Left) استفاده کنید، رشته‌های صفر و یک به هدف چرخاندن موتور به سمت چپ روی پورت P1 تولید خواهند شد.

### شیفت رجیستر

رشته‌های صفر و یک گفته شده را می‌توان به کمک یک تراشه شیفت رجیستر 74LS194 نیز تولید کرد. مدار روبرو را ببینید.

در ابتدای کار ورودیهای S0 و S1 تراشه را «یک» می‌کنیم تا رشته 0011 روی پایه‌های خروجی تراشه شیفت رجیستر ظاهر شود. با قرار دادن ورودیهای انتخاب S1-S0 در حالت 01، با اعمال هر پالس ساعت (که در این مدار به کمک یک کلید

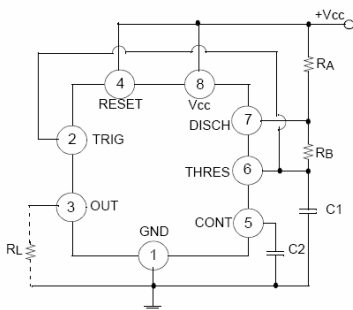


فشاری ایجاد می‌شود) به پین CP، خروجی شیفت رجیستر یک واحد به راست شیفت پیدا می‌کند که به دلیل اتصال ورودی سریال چپ به بیت کم‌ارزش خروجی، این شیفت از نوع چرخشی خواهد بود. اگر می‌خواهید شیفت چرخشی به چپ داشته باشید کافی است ورودیهای انتخاب S1-S0 را در حالت 10 قرار دهید. با تنظیم سرعت اعمال پالس ساعت به پین CP تراشه، می‌توان سرعت ایجاد رشته‌های صفر و یک (و در نتیجه سرعت چرخش موتور) را تنظیم کرد. مدار کنترلی روبات باید ورودیهای انتخاب S0 و S1 را در مواقع لازم مقدار دهی کند.

لازم به ذکر است اگر بخواهیم در هنگام چرخیدن روبات به یک سمت، موتور آن سمت در جهت عکس بچرخد، باید با عکس کردن حالت ورودیهای انتخاب S0 و S1 جهت چرخش بیت‌های خروجی را تغییر دهیم؛ اما اگر بخواهیم در هنگام چرخیدن روبات به یک سمت، موتور آن سمت تنها خاموش شود، کافی است ورودیهای S1-S0 را در حالت 00 قرار دهیم. تا چرخش بیت‌های خروجی متوقف شود.

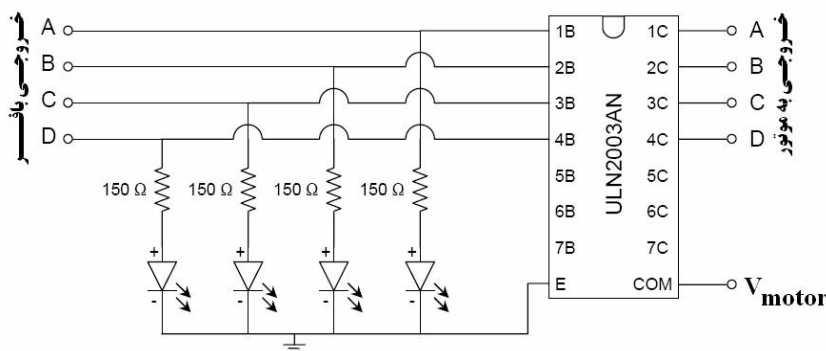
برای تولید متوالی پالس ساعت تراشه شیفت رجیستر، از تراشه 555 در حالت «ناپایا» (Astable) استفاده می‌کنیم.

با تنظیم مقاومتها و خازنهایی که به مدار پایه این تراشه متصل می‌شوند، می‌توان سرعت تولید پالس ساعت (و در نتیجه سرعت تولید رشته‌های صفر و یک و چرخش موتور) را تنظیم کرد. با اتصال پایه ۳ تراشه 555 به پایه CP شیفت رجیستر، پالس ساعت مرتباً به شیفت رجیستر اعمال می‌شود.



### مدار تقویت جریان

چون جریان خروجی از پایه‌های میکروکنترلر محدود است، برای راه‌اندازی موتور باید این جریان را تقویت کنیم. تراشه ULN2003 از پرکاربردترین تراشه‌ها برای این هدف است. شکل روبرو نحوه استفاده از این تراشه برای راه‌اندازی یک موتور پله‌ای



۱۲ ولت را نشان می‌دهد. برای تقویت بهتر جریان می‌توان خروجیهای ULN2003 را با مقاومت‌های مناسب به ولتاژ موتور Pull-up کرد.

وجود LEDهایی که در خروجی بافر کار گذاشته شده الزامی نیست؛ اما بسیار به کار می‌آید. مدار روبات از چند مدار ساده‌تر تشکیل می‌شود که مجموع آنها در کنار هم، مدار پیچیده‌ای است که تحلیل و به ویژه یافتن محل بروز اشکال در مواقعی که روبات به درستی کار نمی‌کند، کار ساده‌ای نیست. برای اشکالزدایی سریعتر، روی برد کنترلی روبات خود در هر نقطه مهم (خروجی مدارهای سویچینگ سنسورها، خروجی مدار فرمان رله، ورودی CP شیفت رجیستر، خروجی مدار تولید «صفر» و «یک»، خروجی بافر برای راه‌اندازی موتور پله‌ای و ...) از LED استفاده کنید تا با توجه به خاموش و روشن بودن این LEDها بتوانید مشکل روبات را به سادگی بفهمید.

### بافر

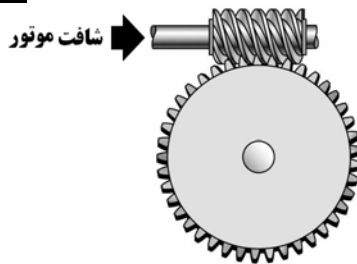
میکروکنترلر یا شیفت رجیستر وظیفه تولید رشته‌های صفر و یک برای هدایت موتور را بر عهده دارند. اما اگر خروجی میکروکنترلر را مستقیماً به مدار تقویت جریان و موتور متصل کنیم، ممکن است موتور جریان زیادی از مدار کنترل موتور بکشد و آن را بسوزاند. برای جدا کردن مدار راه‌اندازی موتور از مدار کنترل موتور، از یک بافر (مثلاً 74LS274 یا 74LS275) استفاده می‌کنیم. این تراشه ۸ بیت ورودی دارد که آنها را عیناً در خروجی تولید می‌کند. اگر از خروجی بافر جریان زیادی کشیده شود تنها بافر را می‌سوزاند و مدار کنترل موتور سالم می‌ماند. کاتالوگ تراشه‌های بافر در ضمیمه آمده است.

## ۵) مکانیک روبات

اگر آنچه را تاکنون گفته شده با موفقیت ساخته‌اید، باید به شما تبریک گفت. البته نه بابت ساخت روبات! بلکه بابت ساخت یک سیستم کنترلی که اطلاعاتی را از دنیای خارج جمع‌آوری می‌کند، بر اساس آن تصمیم‌گیری انجام می‌دهد و بر اساس تصمیم خود فرمانهایی صادر می‌کند.

اما اگر می‌خواهید روبات شما متولد شود، یک مرحله اساسی باقی مانده که اگر اهمیت آن از آنچه تا کنون انجام داده‌اید بیشتر نباشد، بدون شک کمتر هم نیست. اگر فرض کنید تاکنون عملاً یک میلیون روبات در دنیا ساخته شده باشند، به جرأت می‌توان گفت ده‌میلیون روبات به دلیل ساختار مکانیکی ضعیف که نتوانسته فرمانهای کنترلی را به خوبی اجرا کند، از کار بازمانده‌اند. این نکته را فراموش نکنید که هرچقدر سنسورها و مدار کنترلی روبات شما خوب کار کنند، هرچقدر از الگوریتمهای پیشرفته و پردازنده‌های سریع استفاده کنید و هرچقدر موتورهای روبات شما گرانقیمت و کارا باشند، اگر بدنه روبات طوری باشد که فرمانهای نهایی بخش پردازشگر را به خوبی اجرا نکند یا موتورها را به خوبی هدایت نکند، روبات شما تنها باعث تأسف شما خواهد شد!

مکانیک روبات بحث پیشرفته‌ای از رشته مکانیک است که در آن مباحث زیاد و متنوعی مطرح می‌شود. برای یک روبات ساده تعقیب خط، خود شما می‌توانید طرحهای بسیار جالب و متنوعی تدارک ببینید که به شرط صرف زمان مناسب و دقت در طراحی، همگی می‌توانند به خوبی کار کنند. به همین لحاظ در اینجا تنها به چند پیشنهاد اکتفا می‌کنیم. فراموش نکنید اگر تجربه مکانیکی ندارید، حتماً از کسی که علاقه یا تجربه‌اش در این زمینه بیش از شماست کمک بگیرید. وقتی می‌توان چرخها را به کمک دستگاه تراش با دقت زیاد ساخت، استفاده از او و سمباده و ... تنها زمان شما را تلف می‌کند و باعث می‌شود در نهایت به نتیجه مطلوب خود نرسید.



- ساده‌ترین طرح برای ساخت بدنه روبات تعقیب خط، استفاده از دو موتور است. این موتورها را می‌توانید مستقیماً به چرخهای روبات متصل کنید یا برای تقویت بیشتر نیروی موتورها، به کمک حلزونی و چرخ دنده این اتصال را برقرار سازید. شکل روبرو را ببینید.
- از هر موتوری نمی‌توان برای راه‌اندازی روبات استفاده کرد. توان موتور خود را متناسب با وزن بدنه‌ای که می‌خواهید بسازید انتخاب کنید.

روبات و چرخهای آن باید از ماده‌ای سبک با استحکام بالا (مثلاً تفلون یا پلاستیک) ساخته شود. برای اصطکاک بهتر چرخها با سطحی که باید روی آن حرکت کند، در رویه خروجی چرخها از پوششهای لاستیکی آجدار استفاده کنید.



- چون روبات دو چرخ دارد هنگام حرکت، بدنه به بالا و پایین ضربه و به اصطلاح «گه می‌زند». برای حل این مشکل در جلو یا عقب روبات در قسمت میانی، از یک چرخ هرزگرد (یا کف‌گرد) استفاده کنید تا ضمن حفظ تعادل روبات، به حرکت روان‌تر آن کمک کند. این چرخ هرزگرد می‌تواند یک ساچمه بزرگ یا بلبرینگ باشد.

- محل سنسورها بستگی به مدار کنترلی و مکانیک روبات و سرعت آن دارد. نصب سنسورها در قسمت زیرین جلوی روبات می‌تواند انتخاب خوبی باشد.
- یک روبات تعقیب خط باید در کمترین زمان به انتهای خط برسد. هیچ‌گاه سرعت را فدای دقت و دقت را فدای سرعت نکنید. روباتی که بسیار سریع حرکت کند، اما بعد از چند پیچ و انحراف خط را گم کند و روباتی که با حرکت لاک‌پشتی تمام خط را طی کند، هیچ‌کدام مطلوب نیست!
- هنگام طراحی بدنه روبات به مسأله نحوه تغذیه مدارهای روبات دقت کنید. اگر از باتری استفاده می‌کنید، وزن آن را نیز به وزن بدنه و مدارات بیفزایید. اگر از منبع تغذیه خارجی استفاده می‌کنید، باید سیم تغذیه را که به روبات متصل می‌کنید در حین حرکت آن به دنبالش حمل کنید. بنابراین محل مناسبی برای اتصال سیم تغذیه به روبات تعبیه کنید.

## از کجا شروع کنیم؟

مراحل ساخت یک روبات تعقیب خط (و شاید هر روبات دیگر) به صورت کلی و بدون ورود به جزئیات شرح داده شد. از این بعد همه چیز به همت و اراده شما بستگی دارد!

اجزاء یک روبات تعقیب خط را به صورت خلاصه مرور می‌کنیم:

- **سنسور:** برای ساخت این روبات به دو یا چند سنسور نوری نیاز دارید. این سنسور که انواع مختلف دارد (دیود مادون قرمز، فوتوسل، فوتو-ترانزیستور، ...)، مقدار نوری که از محیط دریافت می‌کند (که نشانگر بازتابش نور از سطح سفید یا مشکی است) را به صورت یک ولتاژ آنالوگ (پیوسته) گزارش می‌کند. وقتی می‌خواهید قبل از اتصال سنسورها به روبات آنها را امتحان کنید، حتماً این کار را در محیطی تاریک که مانند زیر روبات باشد، انجام دهید.
- **انتقال خروجی سنسور به بخش پردازشگر:** خروجی سنسور یک ولتاژ آنالوگ است که برای پردازش توسط بخش پردازشگر باید به ولتاژ دیجیتال («صفر» و «یک») تبدیل شود. مدار ترانزیستوری، مدار آپ-

امپی و مدار اشمیت تریگر ولتاژ آنالوگ را به صورت یک بیت منطقی «صفر» یا «یک» که نشان‌دهنده ورود یا عدم ورود روبات به خط مشکی است، تبدیل می‌کنند. تراشه A/D ولتاژ آنالوگ را به صورت یک عدد ۸ بیتی گزارش می‌کند.

- **بخش پردازشگر:** بخش کنترل‌کننده روبات می‌تواند در ساده‌ترین حالت یک مدار منطقی و در حالت‌های پیشرفته‌تر یک میکروکنترلر باشد.
- **موتورها:** آرمیچر یا موتور پله‌ای دو گزینه برای روبات تعقیب خط هستند. آرمیچرها موتورهایی کوچک و سبک و سریع هستند، اما کنترل آنها مشکل‌تر از موتورهای پله‌ای است. برای تنظیم دقیق مقدار و سرعت و جهت چرخش می‌توانید از موتورهای پله‌ای استفاده کنید که البته کندتر و بزرگتر و سنگین‌تر از آرمیچر هستند. متناسب با نوع موتور، باید از مدارات کنترلی و تقویت جریان مناسب استفاده کنید.
- **مکانیک:** بخش مکانیک روبات دارای اهمیت زیادی است و اگر به خوبی طراحی نشود، می‌تواند آنچه رشته‌اید را پنبه کند! در کنار طراحی قسمت‌های دیگر، وقت و نیروی کافی صرف ساخت بخش مکانیک کنید تا زحمات شما به هدر نرود.
- **منبع تغذیه روبات:** بسته به نوع موتورها و مدارات کنترلی ممکن است برای تغذیه روبات خود به ولتاژهای ۵ یا ۱۲ یا ۲۴ ولت نیاز داشته باشید. در حالت معمول، ولتاژ ورودی روبات ۱۲ یا ۲۴ ولت است که باید به کمک رگولاتور (مثلاً 7805)، ولتاژ ۵ ولت برای تغذیه مدارهای کنترلی را از آن استخراج کنید. نکته مهم در تغذیه روبات این است که موتور ممکن است در هنگام کار روی ولتاژ راه‌اندازش نویز ایجاد کند. اگر ولتاژ کاری موتور و ولتاژ تغذیه مدارات روبات از یک منبع یکسان باشند، ممکن است این نویز مشکل‌ساز شود و در کار مدارهای دیگر اختلال ایجاد کند. بنابراین باید تمهیدی برای رفع آن بیاندیشید. اگر از باتری استفاده می‌کنید مواظب تضعیف جریان باتری در اثر کارکرد مداوم موتورها باشید. بهترین گزینه برای تغذیه یک روبات، استفاده از منبع تغذیه کامپیوتر است که ولتاژهای ۵ و ۵- و ۱۲ و ۱۲- ولت را با جریان مناسب و به صورت ایزوله شده در اختیار شما قرار می‌دهد.

برای ساخت روبات تعقیب خط نمی‌توان لیستی از قطعات لازم را ارائه کرد. همانطور که دیدید، برای ساخت هر بخش، طرح‌های متنوعی وجود دارد که هرکدام قطعات خاص خود را می‌طلبند. به کمک اینترنت می‌توانید برای هر طرح نقشه‌ها و قطعات لازم را به دست آورید.

## مراجع

برای ساخت روبات مراجع کلاسیک زیادی وجود ندارد. کتاب «اصول و راهنمای روباتیک» با ترجمه محمد مشاقی طبری که توسط انتشارات کانون نشر علوم منتشر شده یکی از منابع آموزشی روباتیک است که مطالب مفیدی می‌توانید در آن بیابید.

در کتاب‌های الکترونیک عملی می‌توانید مطالبی راجع به مدارهای سویچینگ و کنترل موتور بیابید. کتاب «تکنیک پالس» دیوید بل که توسط محمود دیانی به فارسی ترجمه شده کتاب بسیار مفیدی برای ساخت مدارهای سویچینگ است.

در کتاب‌های زیر می‌توانید مطالب مفیدی در مورد نحوه راه‌اندازی موتورها، A/D و میکروکنترلرها بیابید:

- *"PIC Microcontroller and Embedded Systems", Muhammed Ali Mazidi and et al, Prentice Hall, 2007.*



- *"The 8051 Microcontroller and Embedded Systems", Muhammed Ali Mazidi and et al, Prentice Hall, 2006.*

این کتاب توسط دکتر قددت سپیدنام به فارسی ترجمه شده است.

- «مقدمه‌ای بر میکروکنترلر ۸۹۵۱» نوشته پیام سنایی.

## کلام آخر

روبات تعقیب خط یک بهانه است. بهانه‌ای برای ایجاد انگیزه و گشودن راهی برای آنچه می‌توانید انجام دهید. اگر موفق به ساخت این روبات نشدید، هرگز امید و انگیزه خود را از دست ندهید. دوباره تلاش کنید که هدف باارزش نه «پیروزی» که «تلاش برای پیروزی» است.

اگر این روبات را ساختید، این کامیابی بر شما مبارک! اما اگر انگیزه خود را از دست دهید و به همین اکتفاء کنید، این موفقیت نه یک پیروزی که شکستی بزرگ برایتان خواهد بود. به تلاش خود و لحظاتی که صرف ساخت تک‌تک اجزاء این روبات کرده‌اید بیاندیشید. اگر این لحظات را تنها برای ساخت همین روبات صرف کرده‌اید، بیگمان بعدها با افسوس این لحظات را به یاد خواهید آورد. پس عزم خود را جزم کنید و سدها را بشکنید و آنچه را در عمل آموخته‌اید به دیگران نیز بیاموزید و خود نیز با تمام قدرت و توان پیش بروید و راه روباتیک را در کشورمان بگشایید. جای روباتهای صنعتی، آتش‌نشان، مین‌یاب، جوشکار، نقاش و ... در کشور ما خالی است. پس دست به کار شوید!

**چشم امید ایران به دست توانای شماست...**

