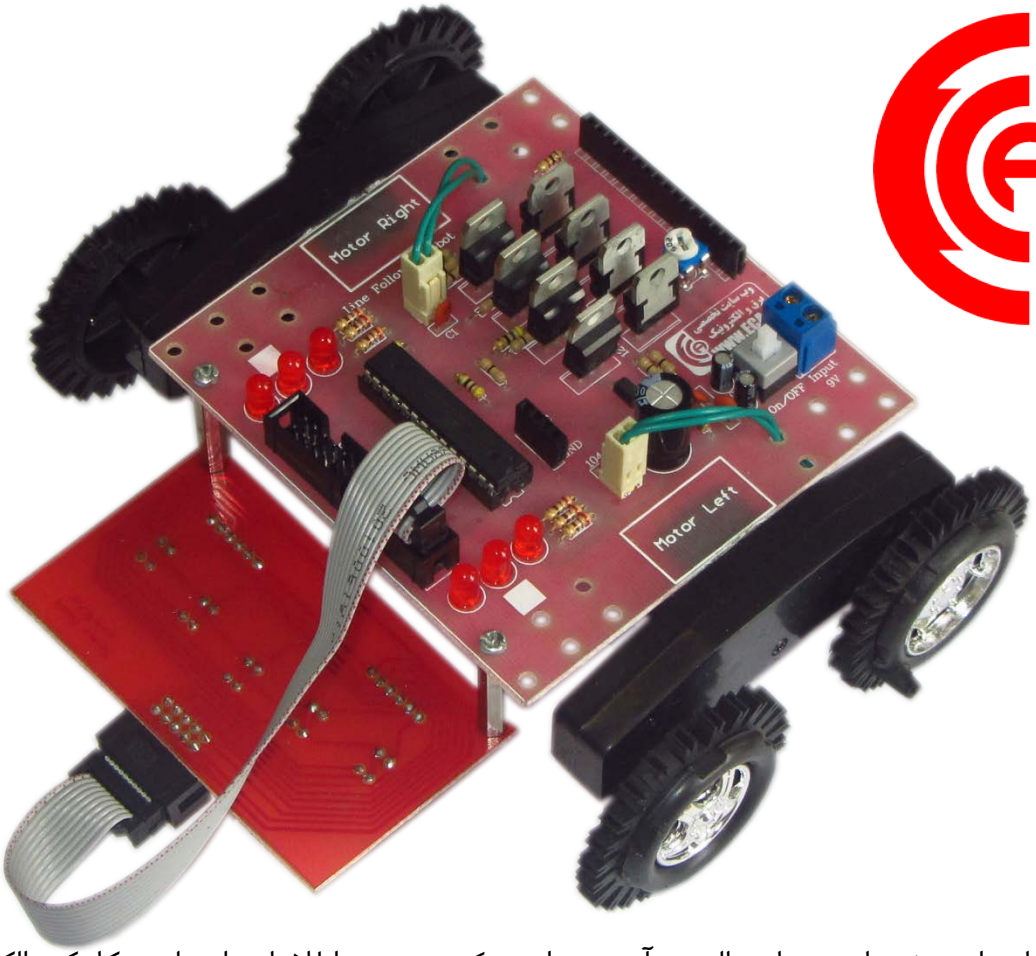
برد آموزشی ربات تعقیب خط



مقدمه

ساخت ربات از موضوعات بسیار جالب و آموزنده است که مجموعه اطلاعاتی از علوم مکانیک، الکترونیک، کامپیوتر (سخت افزار و نرم افزار) و هوش مصنوعی را در قالب یک وسیله هوشمند نمود عملی می بخشد. شاید ساخت تک تک اجزایی که در یک ربات به کار می رود (مانند مدار کنترلی، برنامه نرم افزاری، 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، جبران سازی ضعف های یک بلوک به کمک بلوک های دیگر و در نهایت به وجود آوردن رباتی 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 می دهید را انجام می دهد.

هدف این نوشته آشنا کردن شما با دانش رباتیک و به صورت ویژه بیان اصول ساختن یک ربات بسیار ساده یعنی ربات تعقیب خط می باشد تا در حد توان در رفع نقیصه رباتیک کشورمان که همانا فقدان منابع گویا و ساده است کاری انجام داده باشیم.

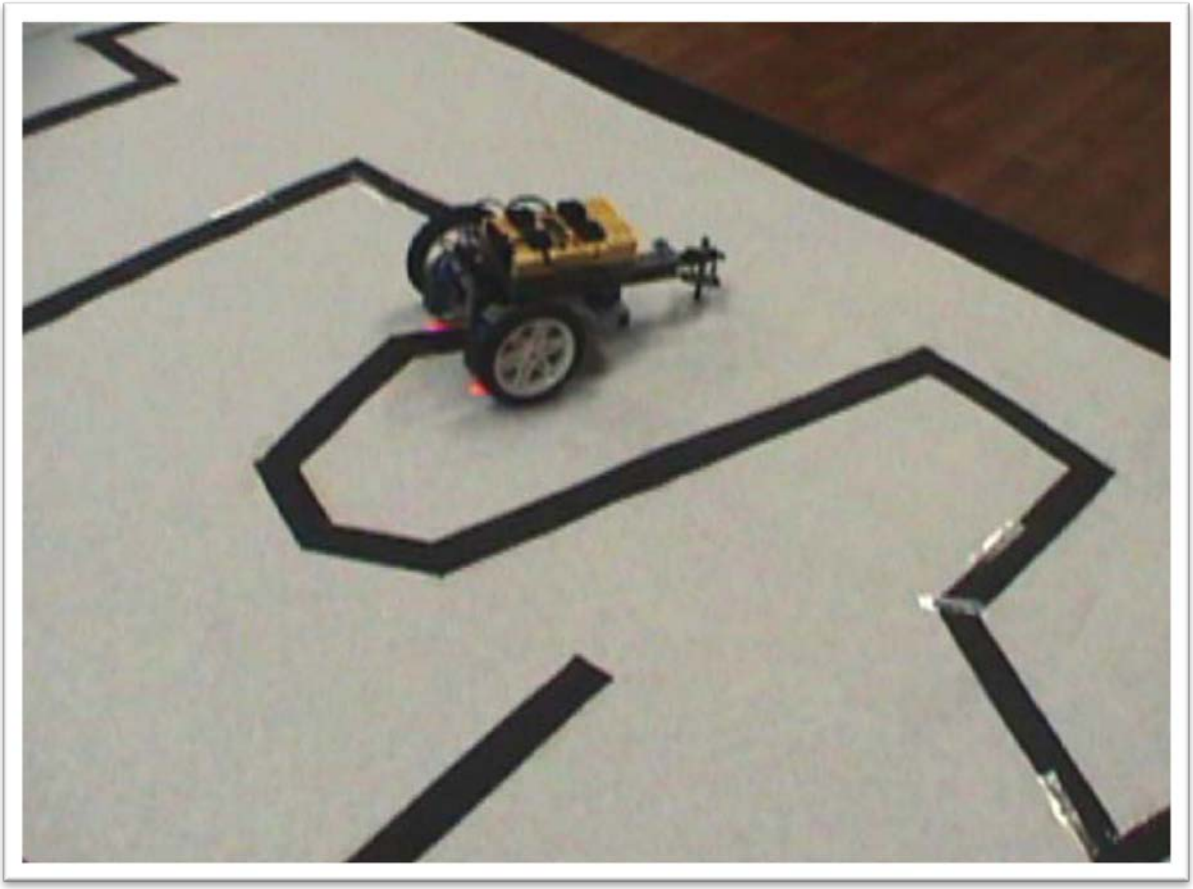
مطالب این نوشته با فرض اینکه خواننده از اصول ابتدایی مدارات منطقی، میکروکنترلرها و الکترونیک آگاه است به رشته تحریر درآمده است.

چرا ربات تعقیب خط؟

در میان ربات های مختلفی که در حیطه مسابقات (مانند ربات مین یاب، لابیرنت، موش پنیریاب، جنگجو، فوتبالیست و...) یا در حیطه صنعتی (ربات های نقاش، جوشکار، مونتاژ قطعات، تیرانداز و...) وجود دارد، می توان ربات تعقیب خط را ساده ترین عضو خانواده ربات ها انگاشت.عضوی که به ر غم بی پیرایه بودن و سادگی، دارای تمام اجزایی است که در یک ربات باید وجود داشته باشد.به همین لحاظ، شاید بهترین و روشن ترین دروازه ورود به دانش رباتیک، ربات تعقیب خط باشد. به همین لحاظ در این نوشته به معرفی کلی اجزای این ربات می پردازیم و آنچه باید برای ساخت آن انجام شود را مرور می کنیم.

ربات تعقیب خط چه باید بکند؟

صورت مسأله ساخت ربات تعقیب خط در مسابقات مختلف دارای جزییات زیاد و بعضاً متفاوتی است؛ اما اصل موضوع در همه آنها این است که ربات تعقیب خط باید بتواند یک خط مشکی رنگ به عرض ۱.۸ سانتی متر (معمولاً چسب لنت 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 ربات تعقیب خط را در شکل روبرو میبینید.

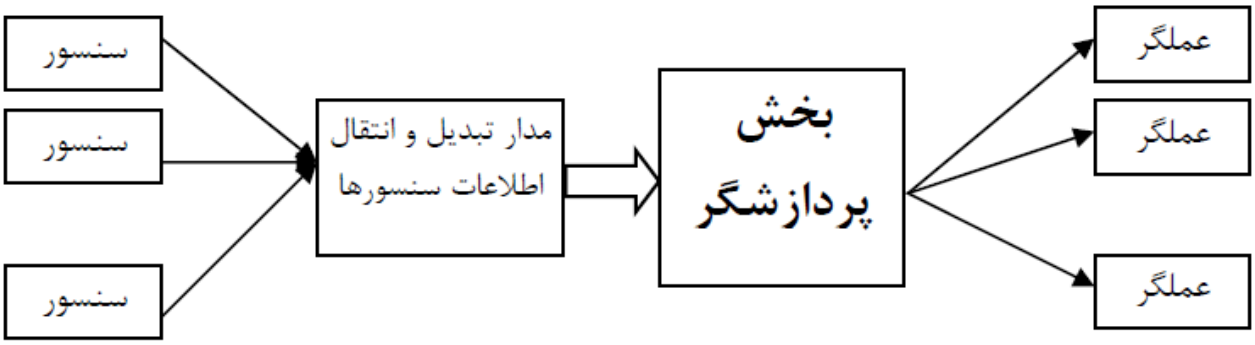


اخیرا مقررات جدیدی از جمله امکان وجود بریدگی در خط، وجود تونل، پیچهای 90 درجه، حلقه، تقاطع و...به قوانین مسابقات اضافه شده است.ترسی به خود راه ندهید ! اگر ربات خود را در ساده ترین حالت ممکن بسازید، اضافه کردن این امکانات به آن کار مشکلی نیست.

بلوک دیاگرام یک ربات تعقیب خط

از نظر کلی هر ربات باید:

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



بخش پردازشگر

تفاوت ربات های مختلف به ماهیت این مراحل برمیگردد.

ربات تعقیب خط هم همین اجزاء را درون خودش دارد:

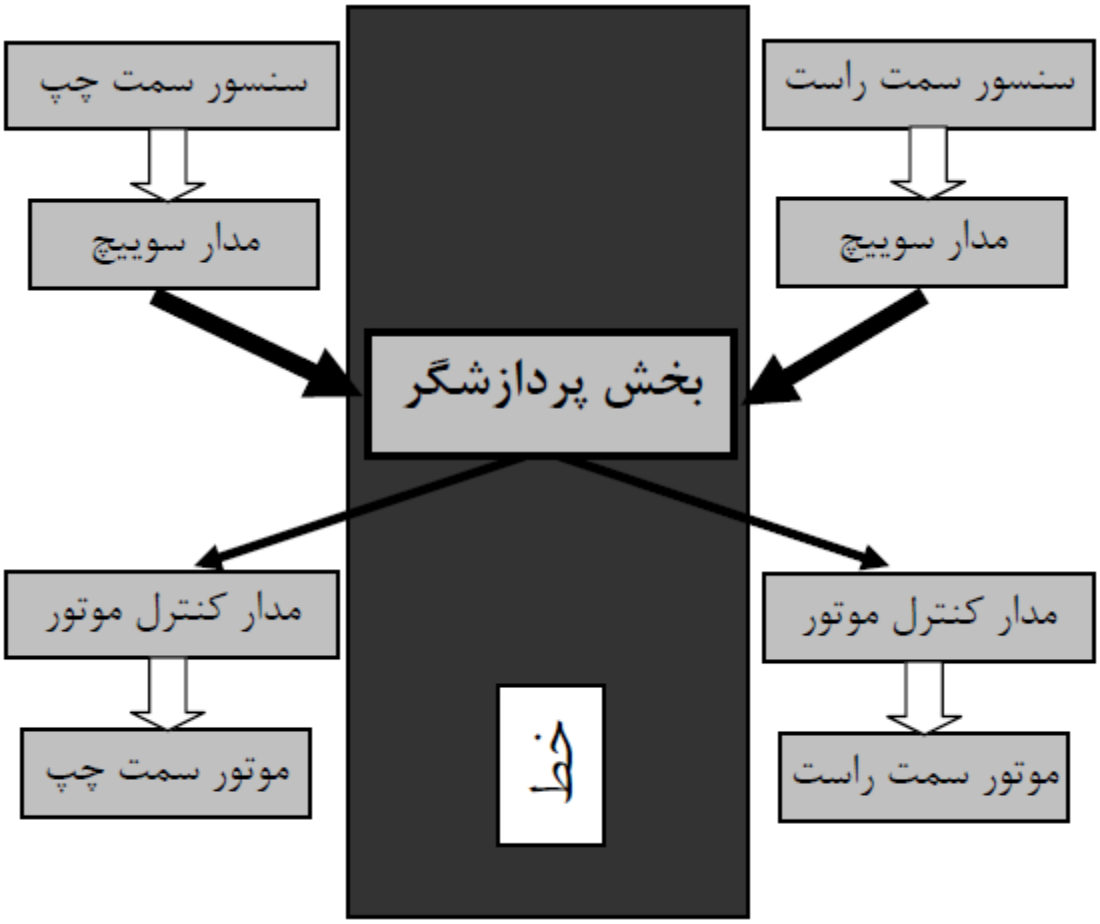
برای حس کردن خط به دست کم دو سنسور نوری که تفاوت رنگ سفید و مشکی را درک کنند نیاز دارد.این سنسورها انواع مختلفی مانند دیود مادون قرمز، مقاومت متغیر با نور (فوتوسل)، فوتو-ترانزیستور و...دارند.

اطلاعات سنسور از طریق یک مدار سوییچ یا مبدل آنالوگ به دیجیتال به واحد پردازشگر منتقل می شود.

مغز پردازشگر این ربات می تواند یک مدار منطقی یا یک میکروکنترلر از هر نوعی باشد.ربات باید در قبال منحرف شدن خط، واکنش مناسب نشان داده و به قولی هرجا خط پیچید او هم بپیچد.

تصمیم واحد پردازشگر مبنی بر ادامه مسیر یا پیچیدن به چپ و راست از طریق فرمان هایی که به موتورهای ربات که ممکن است از نوع DC یا پله ای (Stepper) باشند داده می شود، انجام میپذیرد.

بلوک دیاگرام یک ربات تعقیب خط با دو سنسور را در شکل مقابل میبینید:

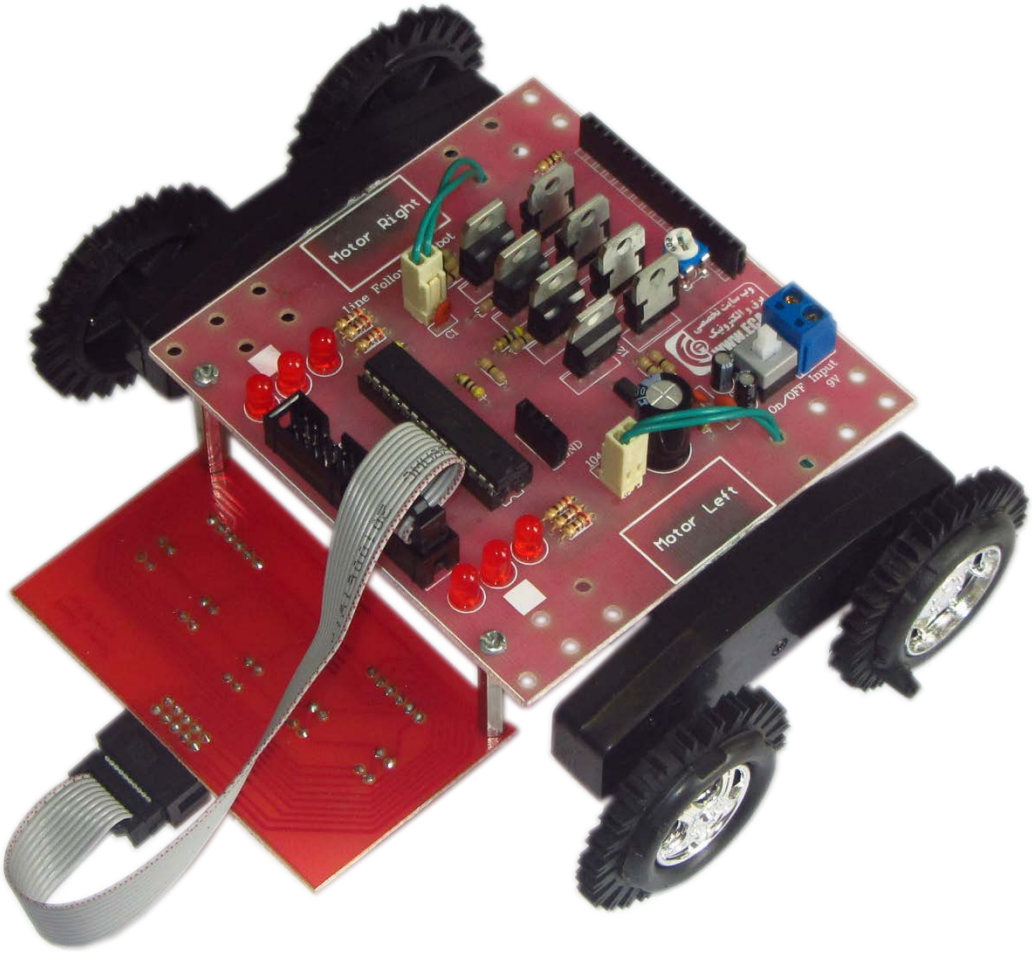


عملکرد ربات تعقیب خط

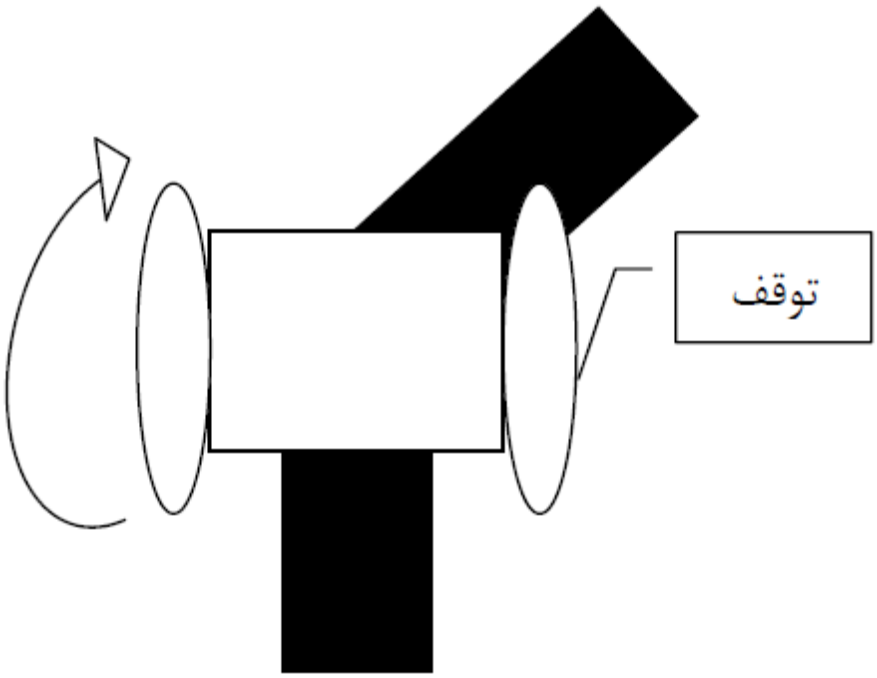
ربات تعقیب خط بسیار ساده عمل می کند.این ربات دست کم دارای دو سنسور نوری است که دو طرف خط مشکی در زمینه سفید قرار می گیرند (تعداد سنسورها به طریقی که خواهیم دید جهت کنترل دقیقتر باید بیش از دو عدد باشد). دو موتور نیز وظیفه به حرکت درآوردن دو چرخ ربات را برعهده دارند.

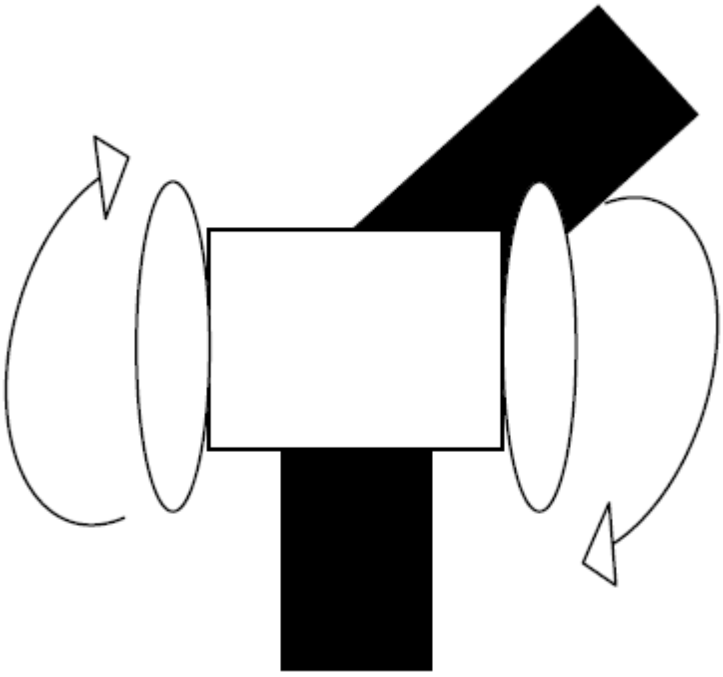
شکل چند ربات تعقیب خط را ببینید:

هرگاه ربات در حین حرکت به انحرافی در خط مشکی برخورد کند، یکی از سنسورهایش وارد خط می شود؛ مثلا اگر خط مشکی به راست منحرف شود، سنسور سمت راست ربات وارد خط مشکی می شود.



ربات باید با توجه به سنسوری که وارد خط شده جهت انحراف خط را متوجه شود و به همان جهت بپیچد. ساده ترین راه برای پیچیدن به یک سمت، خاموش کردن موتور همان سمت و ادامه کار موتور سمت مخالف است.





برای پیچیدن به شیوه حرفه ای تر می توانید به جای خاموش کردن موتور سمت موافق، آن را در جهت معکوس بچرخانید.با این کار ربات در حین پیچیدن روی زمین کشیده نمی شود و می تواند درجا بچرخد.

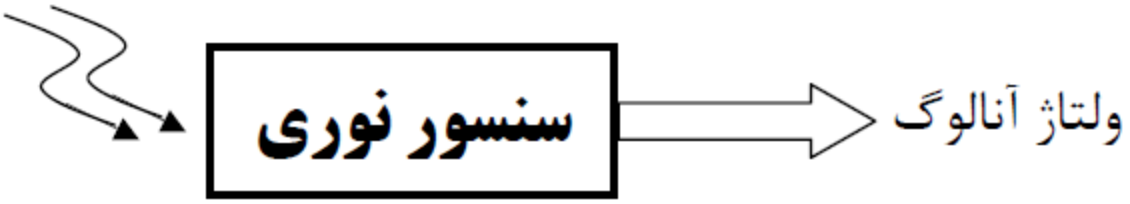
چرخیدن ربات باید تا جایی انجام شود که سنسوری که وارد خط شده بود، از خط خارج شود؛ یعنی مجددا تمام سنسورها خارج خط قرار گیرند تا ربات بتواند به حرکت عادی خود ادامه دهد.

در ذیل اجزای یک ربات تعقیب خط را معرفی می کنیم:

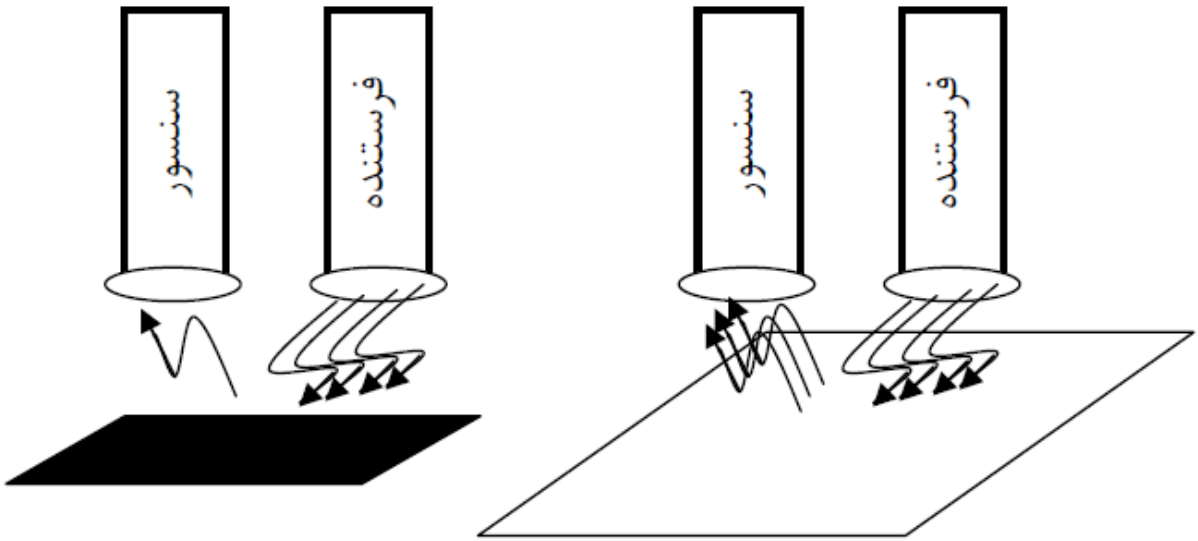
۱) سنسورهای ربات تعقیب خط

همان گونه که گفته شد، ربات 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 ربات کنار هم قرار می دهیم؛ به صورتی که گیرنده مستقیما 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:

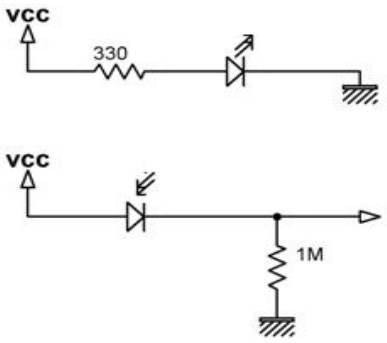


در ذیل سه نوع سنسور نوری مختلف را بررسی می کنیم:

الف) دیود مادون قرمز

فرستنده و گیرنده مادون قرمز دو دیود کاملامشابه هستند. یکی از آنها امواج مادون قرمز را از خود ساطع و دیگری دریافت می کند.

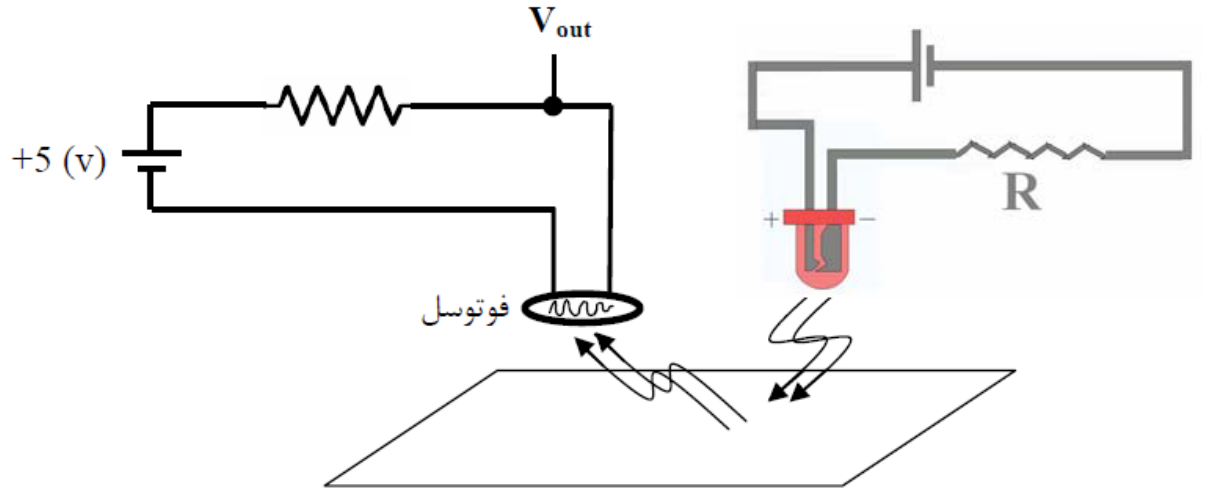
توجه کنید که دیود گیرنده در گرایش معکوس بسته شده است؛ بنابراین اگر میزان نور مادون قرمزی که به آن می تابد کم باشد (مثلاً هنگام بازتابش از سطح مشکی)، دیود هدایت نمی کند و ولتاژ out بالا خواهد بود. اگر نور مادون قرمز به میزان زیاد به دیود گیرنده بتابد، هدایت می کند و ولتاژ out پایین می رود. بنابراین اگر برای تشخیص خط مشکی در زمینه سفید در ربات 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)

مقاومت های معمولی تحت شرایط مختلف (جریان، توان، دما، فشار، رطوبت، نور،...) مقاومت تقریبا یکسانی دارند (البته وقتی این شرایط در بازه های مجاز مربوط به آن مقاومت تغییر کنند). اما مقاوت هایی نیز وجود دارند که با تغییر شرایط، مقاومت متغیری از خود بروز می دهند؛ مثلاً ترمیستور یک مقاومت متغیر با دما است و با تغییر دمای محیط مقدار مقاومت آن تغییر می کند. از ترمیستور به عنوان سنسور دما می توان استفاده کرد.

فوتوسل (Photocell) مقاومت متغیر با نور معمولی است. مقدار این مقاومت بسته به مقدار نوری که به آن می تابد تغییر می کند. برای استفاده از این سنسور در ربات تعقیب خط کافی است آن را همراه با یک فرستنده نوری معمولی مثلا یک LED به کار برید. بهتر است رنگ LED را طوری انتخاب کنید که بیشترین درخشش را داشته باشد (مثلا نارنجی تند).



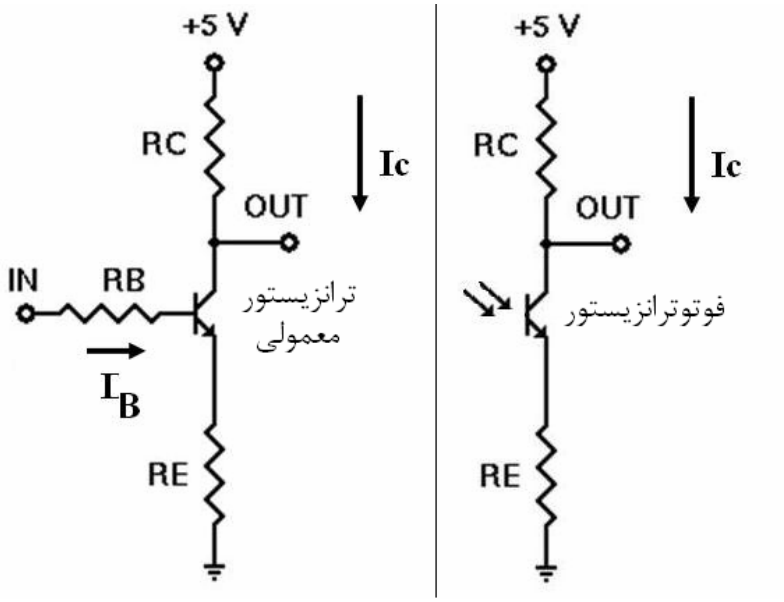
بسته به مقدار نور انعکاسی از سطح (که به سفید یا مشکی بودن آن بستگی دارد) ولتاژ Vout تغییر می کند. به بیان دیگر، ولتاژ Vout با توجه به مقدار مقاومت فوتوسل و مقاومت سری شده با آن، در دو سطح مشکی و سفید دو مقدار متفاوت دارد. از روی مقدار این ولتاژ، ربات می تواند خط را تشخیص دهد.

ج) فوتو-ترانزیستور

در ترانزیستورهای معمولی دوپیوندی (BJT) مقدار جریان Ic که از کلکتور به امیتر جریان دارد، بسته به جریانIB که به بیس آنها وارد می شود تغییر می کند. بنابراین با یک پیکربندی مناسب می توان با تغییر ولتاژ (جریان) بیس، ولتاژ کلکتور را تغییر داد.

درفوتو-ترانزیستور، آنچه باعث تغییر جریان کلکتور به امیتر می شود، نه جریان بیس، بلکه نوری است که به بیس می تابد.

بنابراین بسته به نوری که به بیس فوتو-ترانزیستور می تابد (که با انعکاس از دو سطح سفید و مشکی متفاوت است) ولتاژ کلکتور تغییر می کند. این نور بسته به نوع فوتو-ترانزیستور، می تواند نور معمولی یا مادون قرمز باشد. مدار فرستنده- گیرنده فوتو-ترانزیستور به مدار فوتوسل شباهت زیادی دارد.



معرفی سنسورهای پرکاربرد در ربات های تعقیب خط

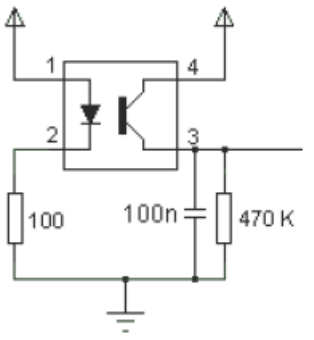
۱. **CNY70**

این سنسور یک سنسور انعکاسی مادون قرمز میباشد که از یک فرستنده مادون قرمز و یک گیرنده فوتو ترانزیستور تشکیل شده است.

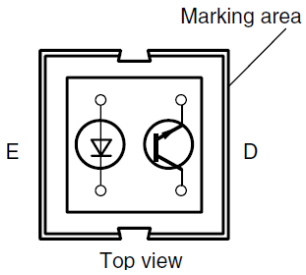
جدول مشخصات :

ABSOLUTE MAXIMUM RATINGS (1)				
PARAMETER	**TEST CONDITION**	**SYMBOL**	**VALUE**	**UNIT**
COUPLER				
Total power dissipation	$T_{amb} \leq 25$ °C	P_{tot}	200	mW
Ambient temperature range		T_{amb}	- 40 to + 85	°C
Storage temperature range		T_{stg}	- 40 to + 100	°C
Soldering temperature	Distance to case 2 mm, $t \leq 5$ s	T_{sd}	260	°C
INPUT (EMITTER)				
Reverse voltage		V_R	5	V
Forward current		I_F	50	mA
Forward surge current	$t_p \leq 10$ µs	I_{FSM}	3	A
Power dissipation	$T_{amb} \leq 25$ °C	P_V	100	mW
Junction temperature		T_j	100	°C
OUTPUT (DETECTOR)				
Collector emitter voltage		V_{CEO}	32	V
Emitter collector voltage		V_{ECO}	7	V
Collector current		I_C	50	mA
Power dissipation	$T_{amb} \leq 25$ °C	P_V	100	mW
Junction temperature		T_j	100	°C

BASIC CHARACTERISTICS (1)						
PARAMETER	**TEST CONDITION**	**SYMBOL**	**MIN.**	**TYP.**	**MAX.**	**UNIT**
COUPLER						
Collector current	$V_{CE} = 5$ V, $I_F = 20$ mA, d = 0.3 mm (figure 1)	I_C (2)	0.3	1.0		mA
Cross talk current	$V_{CE} = 5$ V, $I_F = 20$ mA, (figure 2)	I_{CX} (3)			600	nA
Collector emitter saturation voltage	$I_F = 20$ mA, $I_C = 0.1$ mA, d = 0.3 mm (figure 1)	V_{CEsat} (2)			0.3	V
INPUT (EMITTER)						
Forward voltage	$I_F = 50$ mA	V_F		1.25	1.6	V
Radiant intensity	$I_F = 50$ mA, $t_p = 20$ ms	I_e			7.5	mW/sr
Peak wavelength	$I_F = 100$ mA	λ_P	940			nm
Virtual source diameter	Method: 63 % encircled energy	d		1.2		mm
OUTPUT (DETECTOR)						
Collector emitter voltage	$I_C = 1$ mA	V_{CEO}	32			V
Emitter collector voltage	$I_E = 100$ µA	V_{ECO}	5			V
Collector dark current	$V_{CE} = 20$ V, $I_F = 0$ A, E = 0 lx	I_{CEO}			200	nA



21835

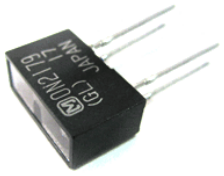


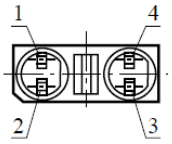
فروشگاه تخصصی برق و الکترونیک www.eShop.ECA.ir ایمیل: eShop.ECA@Gmail.com
شماره تماس: ۰۴۱۱۵۵۳۳۷۳۲ فکس: ۰۴۱۱۵۵۳۹۷۶۹

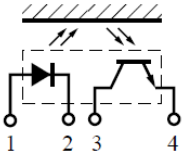
۲. ON2179

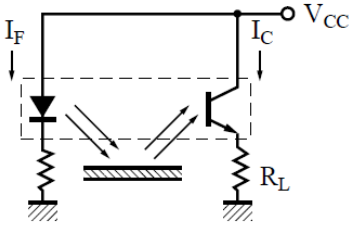
شاید بتوان از این سنسور به عنوان با کیفیت ترین سنسور موجود در بازار نام برد.

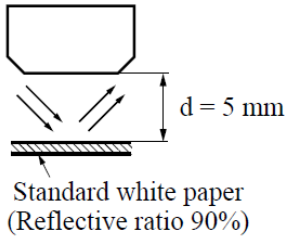
این سنسور دارای عملکرد مناسبی بوده و رنج ولتاژ خروجی مطلوبی دارد.











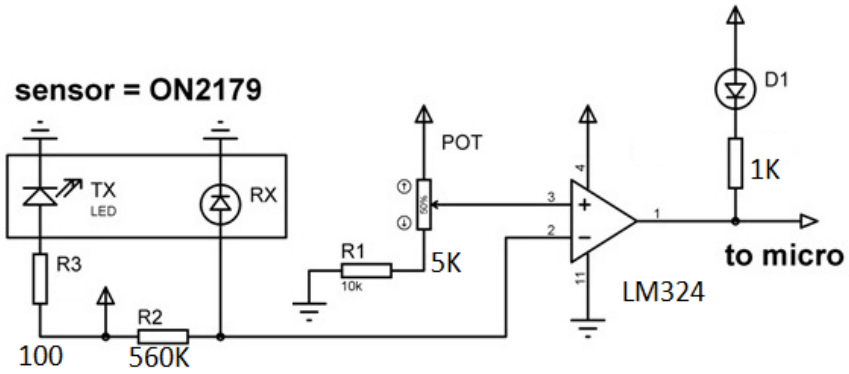
■ Absolute Maximum Ratings (Ta = 25°C)

Parameter		Symbol	Ratings	Unit
Input (Light emitting diode)	Reverse voltage (DC)	V_R	3	V
	Forward current (DC)	I_F	50	mA
	Power dissipation	P_D *1	75	mW
Output (Photo transistor)	Collector current	I_C	20	mA
	Collector to emitter voltage	V_{CEO}	20	V
	Emitter to collector voltage	V_{ECO}	5	V
	Collector power dissipation	P_C *2	100	mW
Temperature	Operating ambient temperature	T_{opr}	−25 to +80	°C
	Storage temperature	T_{stg}	−30 to +85	°C

■ Electrical Characteristics (Ta = 25°C)

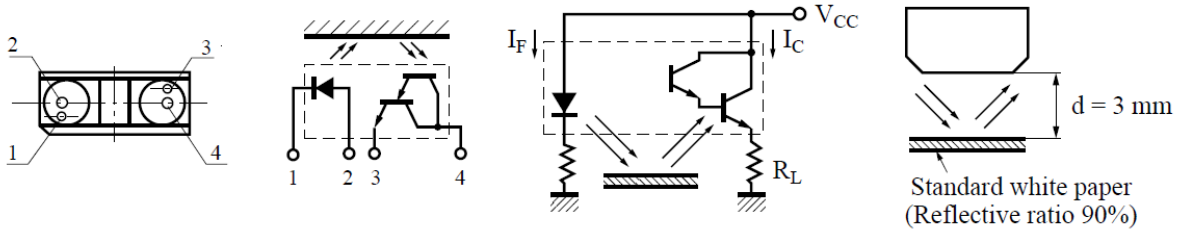
Parameter		Symbol	Conditions	min	typ	max	Unit
Input characteristics	Forward voltage (DC)	V_F	I_F = 50mA		1.3	1.5	V
	Reverse current (DC)	I_R	V_R = 3V			10	μA
Output characteristics	Collector cutoff current	I_{CEO}	V_{CE} = 10V			0.2	μA
Transfer characteristics	Collector current	I_C *1	V_{CC} = 5V, I_F = 20mA, R_L = 100Ω	180		1500	μA
	Response time	t_r *2, t_f *3	V_{CC} = 10V, I_C = 0.1mA, R_L = 100Ω		20		μs
	Collector to emitter saturation voltage	$V_{CE(sat)}$	I_F = 50mA, I_C = 0.1mA			0.5	V

فروشگاه تخصصی برق و الکترونیک www.eShop.ECA.ir ایمیل: eShop.ECA@Gmail.com
شماره تماس: ۰۴۱۱۵۵۳۳۷۳۲ فکس: ۰۴۱۱۵۵۳۹۷۶۹



۳. ON2253

سنسور ON2253 معادل مناسبی برای سنسور ON2179 بوده و از لحاظ کیفیت و عملکرد نیز در سطح مطلوبی قرار دارد. مدار راه انداز مشابهی داشته و تنها مقدار مقاومت های بایاس آن متفاوت است.



■ Absolute Maximum Ratings (Ta = 25˚C)

Parameter		Symbol	Ratings	Unit
Input (Light emitting diode)	Reverse voltage (DC)	V_R	3	V
	Forward current (DC)	I_F	50	mA
	Power dissipation	P_D*1	75	mW
Output (Photo transistor)	Collector to emitter voltage	V_{CEO}	20	V
	Emitter to collector voltage	V_{ECO}	5	V
	Collector current	I_C	30	mA
	Collector power dissipation	P_C*2	100	mW
Temperature	Operating ambient temperature	T_{opr}	−25 to +85	˚C
	Storage temperature	T_{stg}	−30 to +100	˚C

■ Electrical Characteristics (Ta = 25˚C)

Parameter		Symbol	Conditions	min	typ	max	Unit
Input characteristics	Forward voltage (DC)	V_F	I_F = 50mA		1.2	1.5	V
	Reverse current (DC)	I_R	V_R = 3V			10	μA
	Capacitance between terminals	C_t	V_R = 0V, f = 1MHz		50		pF
Output characteristics	Collector cutoff current	I_{CEO}	V_{CE} = 10V			0.5	μA
Transfer characteristics	Collector current	I_C*1*2	V_{CC} = 5V, I_F = 10mA, R_L = 100Ω	3		30	mA
	Response time	t_r*3 , t_f*4	V_{CC} = 10V, I_C = 1mA, R_L = 100Ω		150		μs
	Collector to emitter saturation voltage	$V_{CE(sat)}$	I_F = 50mA, I_C = 1mA			1.5	V

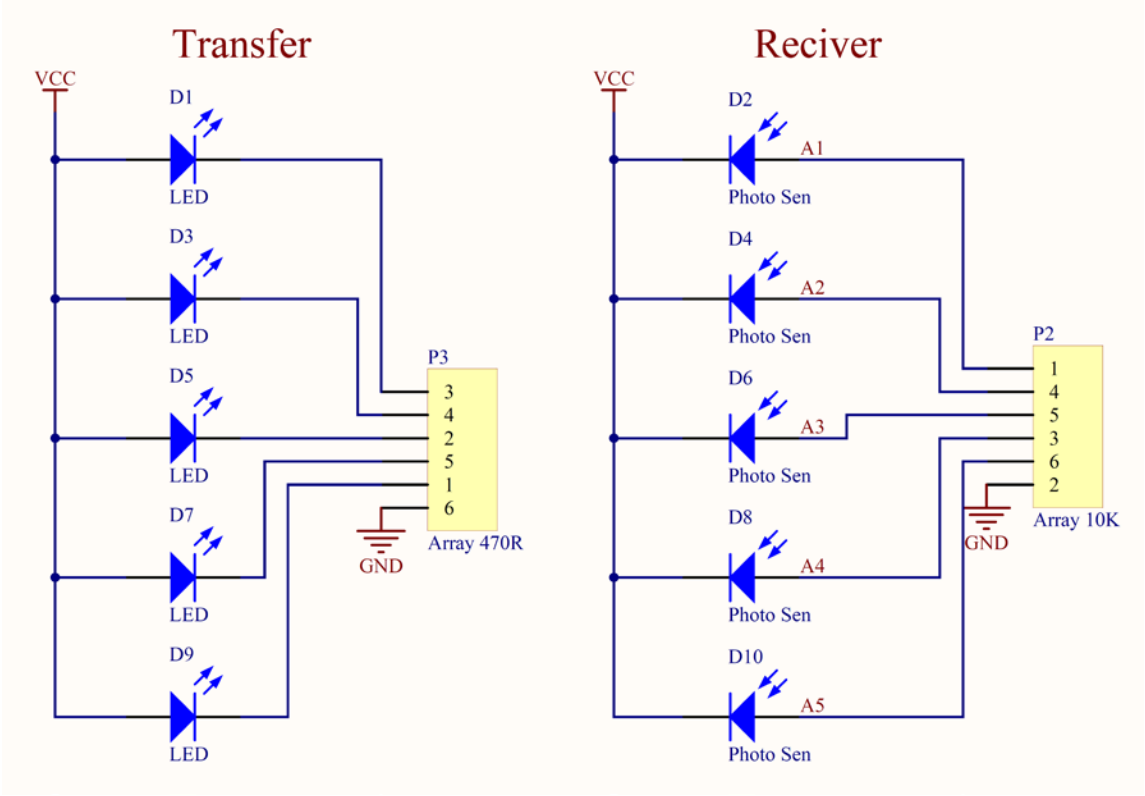
۴. سنسورهای فرستنده و گیرنده مادون قرمز مجزا

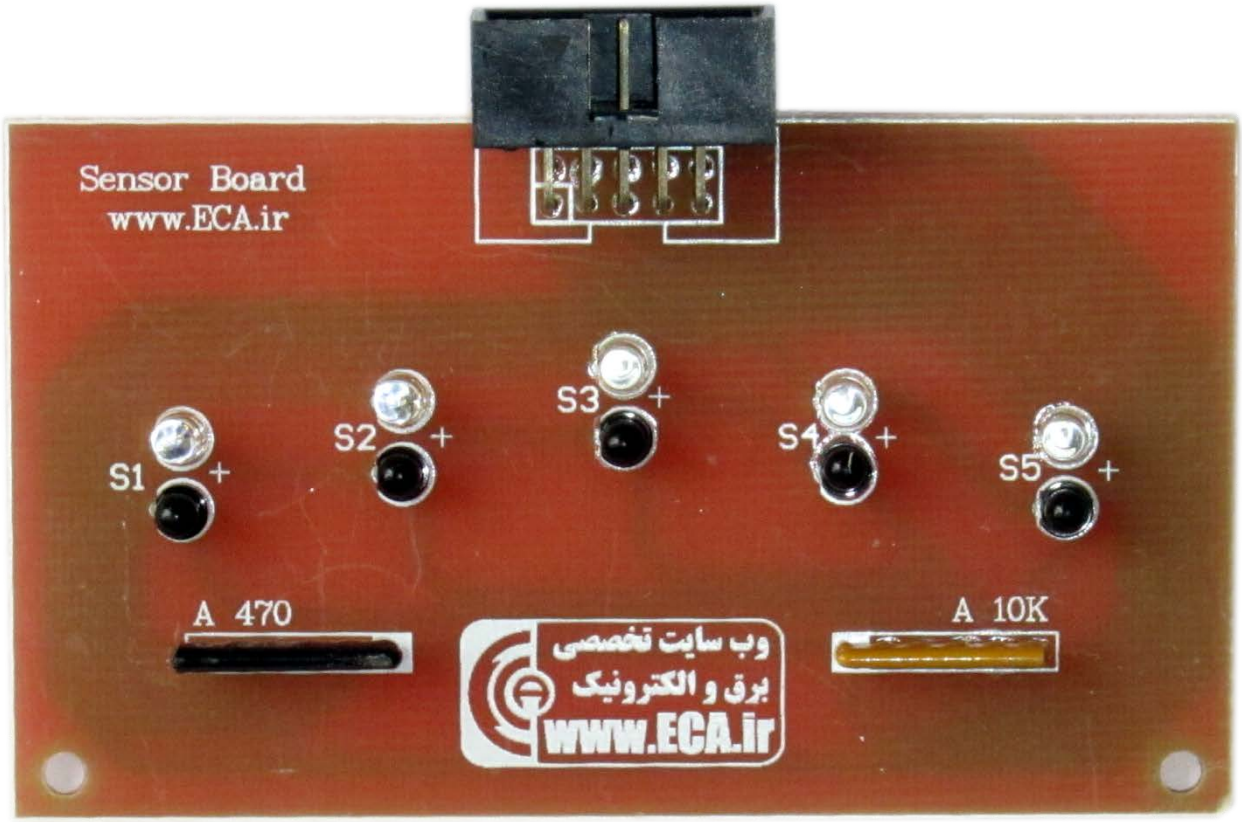
علاوه بر وجود سنسورهای فرستنده و گیرنده به صورت pack، سنسورهای فرستنده و گیرنده به صورت مجزا در بازار موجود می باشد که می توان از آنها در مدارات ربات خود استفاده کنید.

در بالا چگونگی عملکرد این سنسورها توضیح داده شد. در ربات آموزشی هم که در اختیار دارید از سنسورهای فرستنده و گیرنده مجزا استفاده شده است.

مزیت استفاده از این سنسورها به نسبت سنسورهای pack در این است که کاربر می تواند فاصله سنسور را بر طبق نیاز و شرایط خود انتخاب کند.

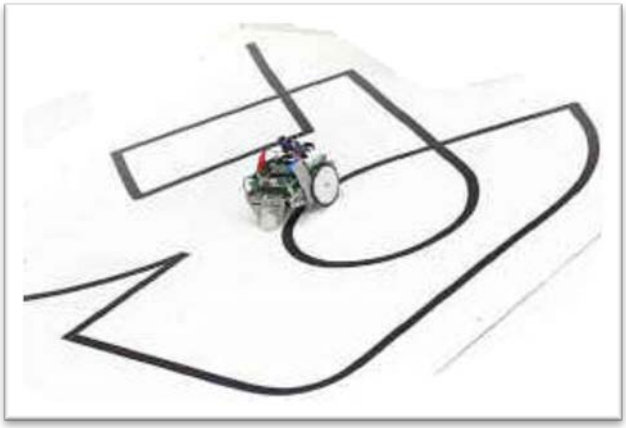
نحوه بایاس سنسورها در برد ربات آموزشی به صورت زیر می باشد.


Transfer
VCC
D1
LED
D3
LED
D5
LED
D7
LED
D9
LED
P3
3
4
2
5
1
6
GND
Array 470R
Reciver
VCC
D2
A1
Photo Sen
D4
A2
Photo Sen
D6
A3
Photo Sen
D8
A4
Photo Sen
D10
A5
Photo Sen
P2
1
4
5
3
6
2
GND
Array 10K


Sensor Board
www.ECA.ir
S1
S2
S3
S4
S5
A 470
A 10K
وب سایت تخصصی
برق و الکترونیک
www.ECA.ir

تعداد و آرایش های سنسورها

اگر خط تنها پیچ و انحراف داشته باشد، دو سنسور که دقیقاً دو سمت خط قرار گرفته باشد، کافی است. اما معمولا در مسابقات رسمی، خط دارای بریدگی ها و تقاطع هایی با زوایا و اندازه های 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 سنسور های بیشتر، عملکرد ربات را ارتقا ببخشید.



جهت کارایی بهتر ربات آرایش های مختلفی از سنسورها به کار می رود که متناسب با سطح مسابقات و البته توان برنامه نویسی کاربر قابل تغییر است. حتما دقت داشته باشید، هماهنقدر که تعداد سنسورهای کم باعث تشخیص کم ربات می شود، تعداد سنسورهای بیش از اندازه نیز کارایی چندانی نخواهد داشت. لذا به صورت معمول بین ۵ تا ۱۲ سنسور میتواند تعداد مناسبی باشد.

نحوه قرارگیری سنسورها

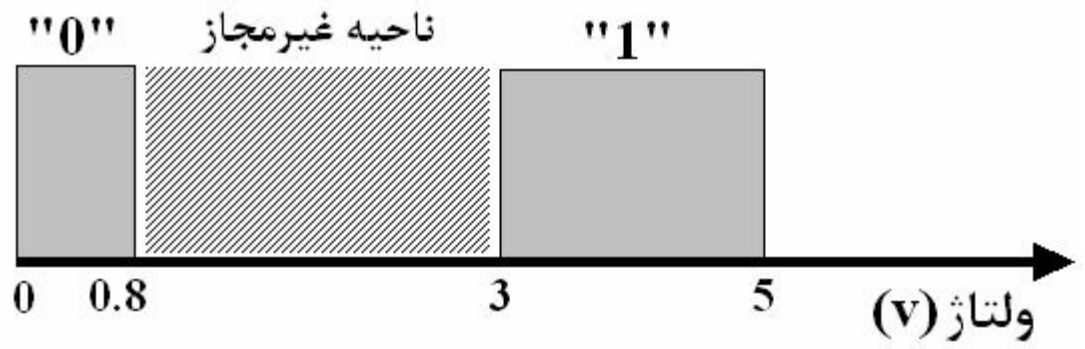
دقت کنید که سنسورها و نحوه قرارگیری آن ها باید دارای یک اصول خاص باشد. طی این اصل فاصله سنسورهای مجاور از هم باید حداقل به اندازه سطح مسیر مشکی باشد (لنت برق دارای ضخامت ۱۸ میلی متر است). و همچنین باید زاویه ها و پیچ ها را هم در نظر گرفت که به صورت کلی حداقل فاصله را ۳ میلی متر کمتر از مسیر مشکی در نظر می گیریم و در نهایت سنسورها را با فاصله ۱۵ میلی متر از هم قرار می دهیم.

از لحاظ ارتفاع هم دقت کنید چیدمان به صورتی باشد که سنسورهای سطر زیرین در وسط سنسورهای ردیف بالاتر قرار گیرند.

۲)تبدیل و انتقال خروجی سنسور به مدار پردازشگر

خروجی هرکدام 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 سنسور های مورد استفاده در مدار در محیط کاملاً سفید حدود ۳.۵ ولت است و با ورود کامل آن سنسور به خط مشکی خروجی سنسور ۱.۵ولت می شود (تلاش نکنید سنسور خود را طوری بایاس کنید که ولتاژ خروجی آن دقیقا 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 ربات را هدایت کند. اما مدارهای پردازشگر (چه مدار منطقی و چه میکروکنترلر) تنها مقادیر دیجیتال را می شناسند. یعنی باید با آنها به زبان یک و صفر صحبت کرد.



بنابراین اگر مثلاً خروجی سنسور فوق در محیط کاملاً سفید ۳.۸ ولت و در محیط کاملاً مشکی ۱.۵ 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.

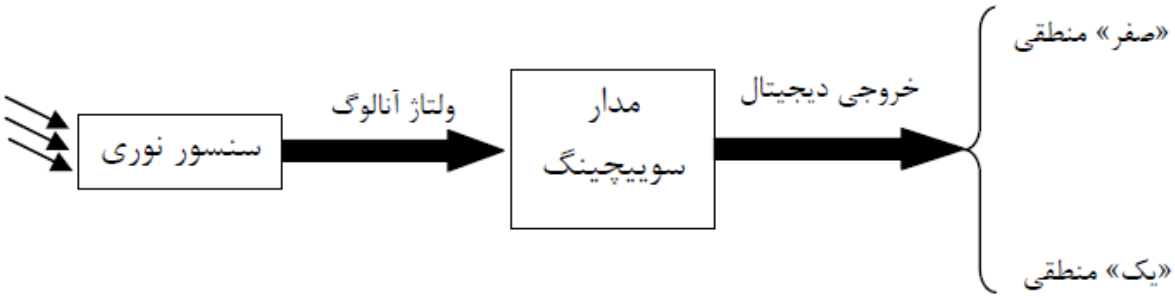
برای تبدیل خروجی آنالوگ سنسور به مقدار دیجیتال دو راه وجود دارد:

- به کمک یک **مدار سوییچینگ** شما می توانید 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.

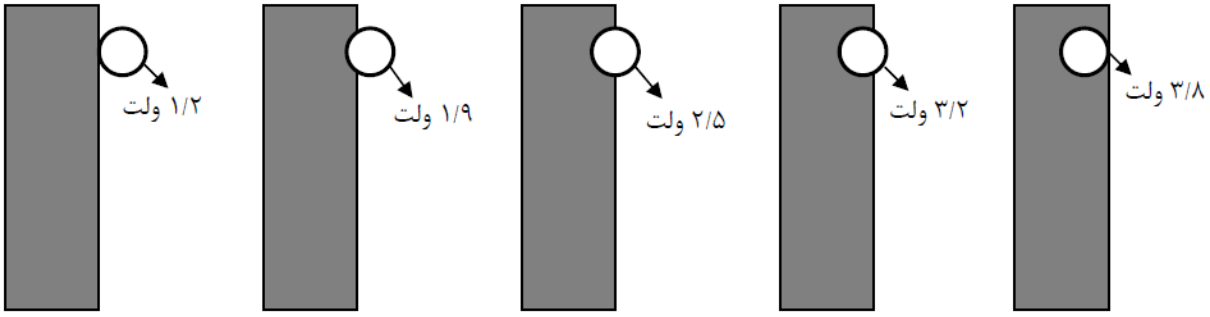
مدار سوییچینگ

به کمک یک مدار الکترونیکی سوییچینگ می توان 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 می خواهیم که مثلاً وقتی سنسور در منطقه سفید است صفر منطقی و و وقتی وارد منطقه مشکی می شود یک منطقی را به عنوان خروجی تولید کند. با این کار، مدار پردازشگر از روی این ولتاژ متوجه ورود یا عدم ورود ربات به خط مشکی می شود.

بلوک دیاگرام مدار سوییچینگ به شکل زیر است:

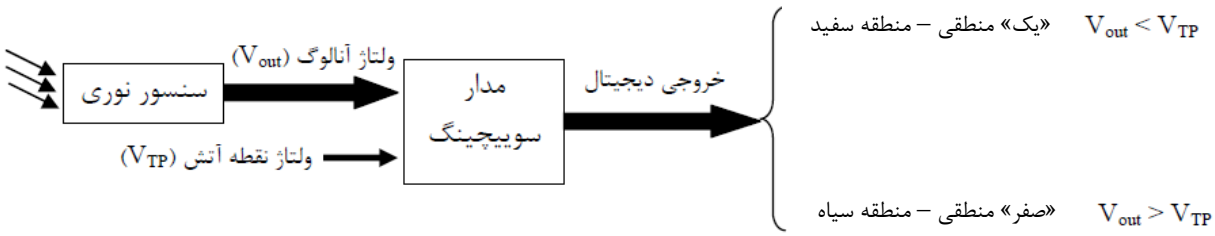


به یک نکته مهم توجه کنید. فرض کنید هنگامی که سنسور کاملا از خط مشکی خارج است (در محیط کاملا سفید است) دارای خروجی ۳.۸ ولت می باشد و با ورود کامل به خط مشکی دارای ولتاژ ۱.۵ ولت می شود. می خواهیم این ولتاژهای آنالوگ را به کمک یک مدار سوییچینگ به گونه ای به یک ولتاژ دیجیتال تبدیل کنیم که وقتی سنسور در منطقه سفید است یک منطقی و و وقتی وارد منطقه مشکی می شود صفر منطقی را به عنوان خروجی داشته باشیم. هنگامی که سنسور از زمینه سفید وارد خط مشکی میشود، خروجی سنسور به مرور از ۳.۸ ولت تا ۱.۵ ولت تغییر میکند.



به نظر منطقی می آید ربات باید طوری طراحی شود که با رسیدن ولتاژ خروجی سنسور به میانه این بازه (مثلا ۲.۵ ولت) با توجه به اینکه سنسور کم و بیش وارد خط شده، خروجی مدار سوییچینگ از یک به صفر تغییر کرده و جهت حرکت ربات تغییر کند. به این ولتاژ که با گذر از آن عمل سوییچینگ اتفاق می افتد، اصطلاح نًاقطه آتش (Trigger Point) گفته می شود.

فروشگاه تخصصی برق و الکترونیک www.eShop.ECA.ir
شماره تماس: ۰۴۱۱۵۵۳۳۷۳۲
ایمیل: eShop.ECA@Gmail.com
فکس: ۰۴۱۱۵۵۳۹۷۶۹



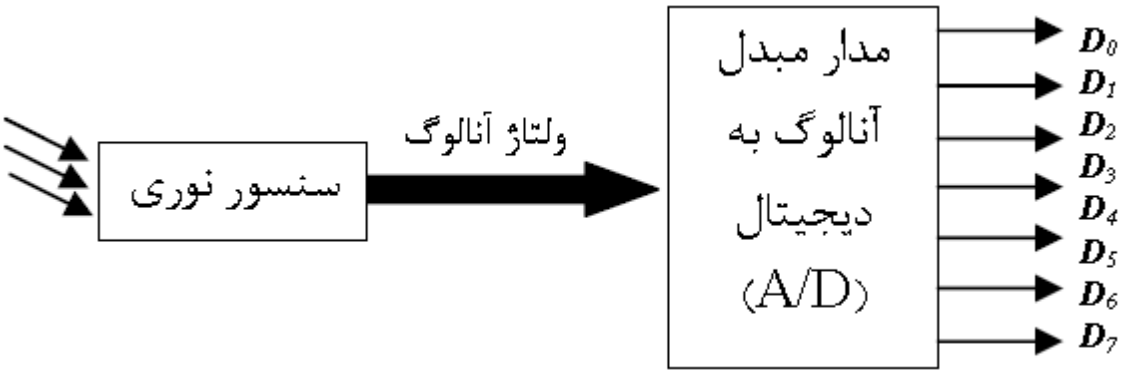
اینکه ولتاژ خروجی سنسور به چه حدی برسد که متوجه شویم سنسور وارد خط شده است یا نه به نوع مکانیک ربات، نور محیط، سنسور انتخابی و باقی موارد محیطی برمیگردد. لذا جهت طراحی این مدار باید ابتدا ولتاژ های خروجی مدار دقیقا بررسی شده متناسب با آن طراحی های لازم صورت گیرد.

از آنجایی که این روش نیاز به تنظیمات سخت افزاری زیادی در محیط مسابقه و یا حتی محیط آزمایشی دارد مورد بحث ما نیست و ما تنها به اشاره روش های معمول طراحی این مدار سوئیچ اشاره می کنیم :

- مدارهای ترانزیستوری
- مدار سوئیچ با آپ امپ
- مدارات اشمیت تریگر و یا IC های اشمیت تریگر مانند ۷۴۱۴

تبدیل خروجی آنالوگ سنسور به ولتاژ دیجیتال به کمکA/D

مدارات سوییچینگ، ولتاژ خروجی سنسور را به یک ولتاژVout تبدیل می کنند؛ به بیان دیگر، به کمک خروجی دوحالته یک مدار سوییچینگ می توان تنها دانست آیا ربات 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 می کند .بنابراین اگر بتوانیم ولتاژ خروجی سنسور را نه به صورت یک رقم منطقی صفر یا یک که مثلاً به کمک یک عدد 8 بیتی بیان کنیم، آنگاه می توان به کمک 256 عدد مختلفی که می توان با 8 بیت ساخت، مقدار ورود سنسور به خط را نیز با دقت زیاد با عددی بین صفر تا 255 مشخص کرد. مدار مبدل آنالوگ به دیجیتال این کار را انجام می دهد.



در میکروکنترلر ATMEGA48 که در برد آموزشی مورد استفاده قرار گرفته است یک پورت به عنوان مبدل آنالوگ به دیجیتال قرار داده شده تا بتوان با اعمال ولتاژ آنالوگ به هر یک از پین های این پورت، معادل دیجیتال آن را محاسبه نمود.

از مهمترین مزایای استفاده از این روش می توان به موارد زیر اشاره نمود :

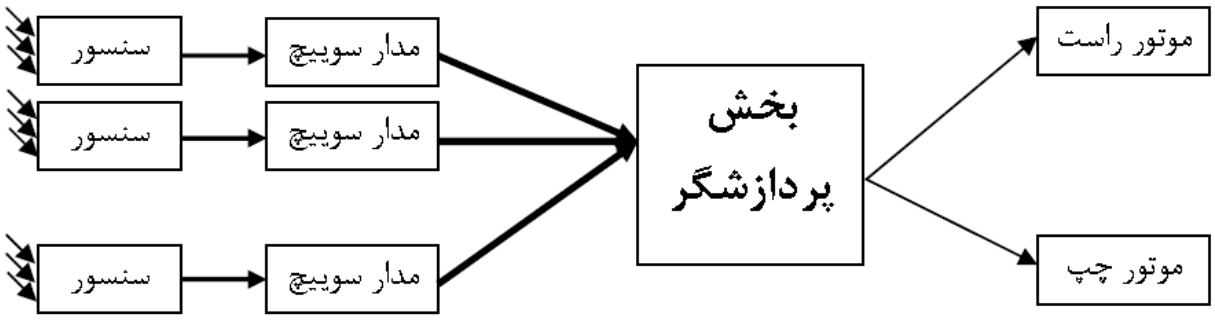
- ساده شدن مدار و کم کردن قطعات مورد استفاده در ربات
- برنامه نویسی حرفه ای تر و امکان تنظیم بسیار دقیقتر سنسورها
- قابلیت تنظیم نرم افزاری ربات بدون نیاز به دستکاری های سخت افزاری قبل و بعد از هر مسابقه و یا آزمایش

با استفاده از مبدل های AD ضمن شناسایی ورود و یا خروج سنسور از خط، میتوان اطلاعات ارسالی را برای موارد دیگری از جمله نمایش بر روی LCD و غیره استفاده نمود.

** آموزش کامل استفاده از مبدل آنالوگ به دیجیتال میکرو در داخل فایل های آموزش برنامه نویسی میکروکنترلرهای AVR در داخل CD محصول موجود هستند که می بایستی توسط کاربر مطالعه شوند.

۳) بخش پردازشگر

بلوک دیاگرام بخش پردازشگر به شکل زیر است:



برد آموزشی ربات های تعقیب خط، بر مبنای پردازش از طریق میکروکنترلر AVR طراحی گردیده است. دقت داشته باشید که در مدار طراحی شده، بخش مدار سوئیچ با استفاده از امکانات داخلی میکروکنترلرهای AVR حذف شده و به صورت برنامه نویسی انجام خواهد شد.

به دلیل گستردگی مبحث میکروکنترلرها، ما آموزش های مجزایی را در داخل CD قرار داده ایم تا کاربر با توجه به وقت و نیز خود اقدام به یادگیری این میکروکنترلر نماید لذا در اینجا از توضیح بیشتر در این مورد پرهیز می کنیم.

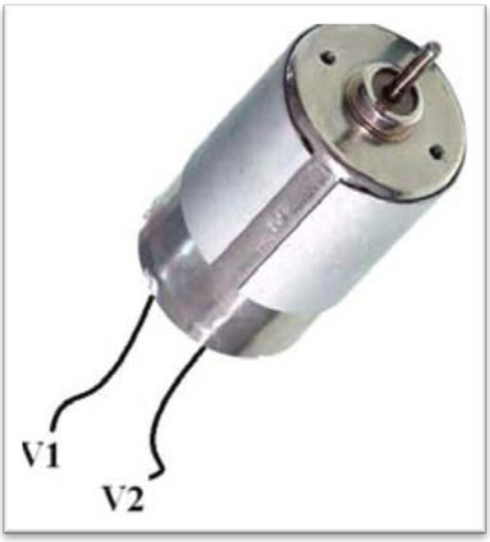
۴)موتور و مدارهای کنترل آن

مدار پردازشگر ربات (چه مدار منطقی و چه میکروکنترلر) باید به موتورها فرمان دهد. این فرمان بر حسب نوع موتور متفاوت خواهد بود که در ذیل انواع آن نام برده خواهند شد.

قبل از آن به نکته ای توجه کنید، هنگامی که به عنوان مثال ربات میخواهد به سمت چپ بپیچد، باید موتور سمت چپ را خاموش و موتور سمت راست را روشن کنیم. در این حالت ربات در حین پیچیدن کمی به جلو نیز حرکت کرده و روی زمین کشیده می شود. برای اینکه ربات 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 می کنیم.

موتور DC

موتور DC ساده ترین نوع موتور است که آن را در ضبط صوتهای کوچک اسباب بازیهای برقی دیده اید. شکل روبرو یک آرمیچر را نشان میدهد .هنگامی که V1>V2 موتور به سمت راست و اگر V2>V1 موتور به سمت چپ می چرخد.



موتور DC گیربوکس دار

یکی از انواع موتورهای الکتریکی، موتور گیربکس دار می باشد. همان گونه که می دانید موتورهای الکتریکی معمولاً سرعت چرخش بالا و قدرت پایینی دارند.

این خاصیت موتورهای الکتریکی می تواند در خیلی از موارد نیازهای مصرف کننده را رفع نماید ولی در برخی از کاربردها، ممکن است قدرت بالاتر و سرعت چرخش پایین تری مورد نیاز باشد. در این مواقع نمی توان از موتورهای الکتریکی عادی استفاده نمود و باید قدرت و سرعت چرخش آنها را به نحوی تنظیم نمود. بهترین راه برای افزایش قدرت و کاهش سرعت چرخش موتورهای الکتریکی، استفاد از گیربکس یا جعبه دنده می باشد .

گیربکس قادر است قدرت چرخش را افزایش و سرعت آن را کاهش دهد. هر چه نرخ کـاهش سـرعت در گیربکس بیشتر شود، قدرت آن بالاتر خواهد رفت.

نرخ کاهش سرعت و افزایش قدرت گیربکس را می توان با تغییر سایز چرخ دنده هـای گیـربکس تنظـیم نمود .

امروزه موتورهای گیربکس دار به صورت آماده و در مدل های مختلف در بازار موجود مـی باشـند. ممکـن است این مدل ها از لحاظ ظاهری برابر باشند، ولی گیربکس آن ها و سرعت چرخش آن ها با هـم متفـاوت خواهد بود. سرعت چرخش و قدرت یک موتور گیربکس دار معمولاً بر روی آن حک می شود .

موتورهای گیربکس داری که در برد آموزشی استفاده شده اند به صورت زیر می باشد.

دلیل استفاده از این موتورها، تغذیه مورد نیاز پایین و سرعت مناسب برای ربات های آموزشی می باشد. به صورتی که می توان عملکرد ربات و نحوه تعقیب خط را به وضوح دید و برنامه را طبق حرکت مشاهده شده به صورت حرفه ای تغییر داد.

موتور پله ای(*Stepper Motor*)

موتور پله ای یک موتور الکتریکی هست که حرکت آن کاملا دقیق و از پیش تعریف شده می باشد که ورودی الکتریکی دیجیتال را به یک حرکت مکانیکی تبدیل می کند و با ارسال بیتهای ۰,۱ به سیم پیچهای آن می توان موتور را حرکت داد. در واقع یک موتور پله ای ترکیبی از یک موتور الکتریکی DC و یک سلونویید است. موتورهای پله ای ساده توسط بخشی از یک سیستم دنده ای در حالت های موقعیتی معین قرار می گیرند، اما موتورهای پله ای نسبتا کنترل شده، می توانند بسیار آرام بچرخند. موتورهای پله ای کنترل شده با کامپیوتر یکی از فرمهای سیستم های تنظیم موقعیت هستند، بویژه وقتی که بخشی از یک سیستم دیجیتال دارای کنترل فرمان بار باشند. این موتورها به صورت درجه ای دوران می کنند و با درجه های مختلف در بازار موجود هستند.

در واقع واژه پله به معنی چرخش به اندازه درجه تعریف شده موتور است. مثلا استپ موتوری با درجه ۸.۱ باید ۲۰۰ پله انجام بدهد تا ۳۶۰ درجه یا یک دور کامل بچرخد (۲۰۰X۱.۸ = ۳۶۰) و یک استپ با درجه ۱۵ فقط باید ۲۴ پله برای یک دور کامل انجام بدهد (۲۴X ۱۵ =۳۶۰) . این ویژگی فوائد بسیار زیادی دارد از جمله امکان کنترل سرعت.

درایور کردن موتورها

موتورها پرمصرف ترین جزء ربات ها هستند و بسته به وزن ربات(بار وارد شده بر موتور) می توانند جریانی بین ۲۰۰ میلی آمپر تا چندین آمپر را مصرف کنند.

حال با توجه به گفته های بالا چطور می توان با استفاده از میکروکنترلری که ماکزیمم جریان خروجی هر پین ۴۰ میلی آمپر است یک موتور با جریان مورد نیاز ۲۰۰ میلی آمپر یا بیشتر را درایو کرد؟

برای این کار لازم است از یک قطعه واسط بین پردازنده و موتور استفاده کرد تا فرامین پردازنده را تقویت کرده و ولتاژ و جریان موردنیاز موتورها را تامین کند. این وظیفه در ربات ها بر عهده مدارات درایور(راه انداز) می باشد.

با توجه به تنوع فوق العاده موتورها، درایورها نیز تنوع زیادی دارند. برای انتخاب یک درایور باید ویژگی هایی را مدنظر داشت که به آن ها اشاره خواهیم کرد:

- درایور مورد استفاده توانایی تامین جریان و ولتاژ کافی برای موتور را داشته باشد.
- رایور قابلیت اجرای فرامین مورد نیاز جهت کنترل موتور را دارا باشد.
- درایور در بازار موجود بوده و قیمت آن مناسب باشد.

حال با توجه به مباحث گفته شده شروع به معرفی چند نمونه درایور و نحوه استفاده از آن ها خواهیم کرد.

به طور کلی۳ نوع فرمان را می توان به یک موتور DC اعمال کرد:

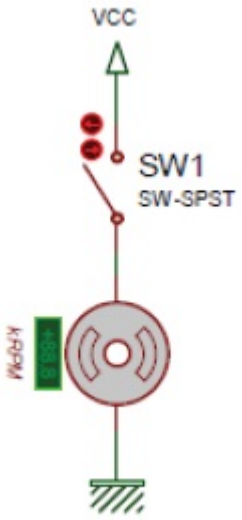
- جلو (Forward)
- عقب (Backward)
- توقف (Stop)

درایورهای موتورهای DC طبق فرامینی که می توانند به موتور بدهند به دو دسته کلی تقسیم می شوند:

۱- درایورهای یک طرفه

منظور از درایورهای یک طرفه موتورهای DC مداراتی هستند که قادرند موتور را تنها در یک جهت به حرکت درآورند و قادر به معکوس کردن جهت چرخش موتور نیستند. به عبارتی تنها می توانند فرمان توقف و عقب یا جلو را به موتور بدهند.

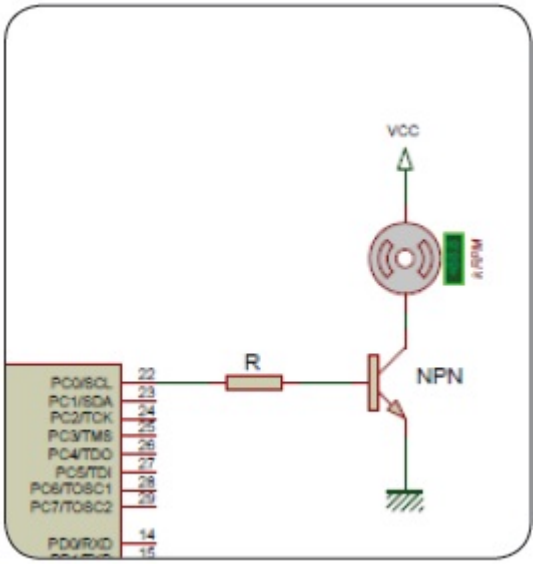
در اینگونه درایورها معمولا یک سر موتور به طور ثابت به یکی از خطوط تغذیه متصل می شود و کلیدی بین پایه ی دیگر موتور و خط دیگر تغذیه قرار می گیرد. در این مدار با بسته شدن کلید، موتور به حرکت درمی آید و با باز شدن کلید موتور متوقف خواهد شد.



با توجه به کاربرد موتور مورد استفاده، ما مجبور به استفاده از کلیدهای الکترونیکی هستیم تا توانایی کنترل موتور را با پالس های الکتریکی داشته باشیم.

a- درایور یک طرفه با ترانزیستور

ترانزیستور می تواند مانند یک کلید در مدار باعث قطع و وصل جریان عبوری شود. در شکل زیر در حالتی که خروجی میکروکنترلر یک شود ترانزیستور وصل شده و موتور به حرکت درمی آید و با صفر شدن پین موتور متوقف خواهد شد.

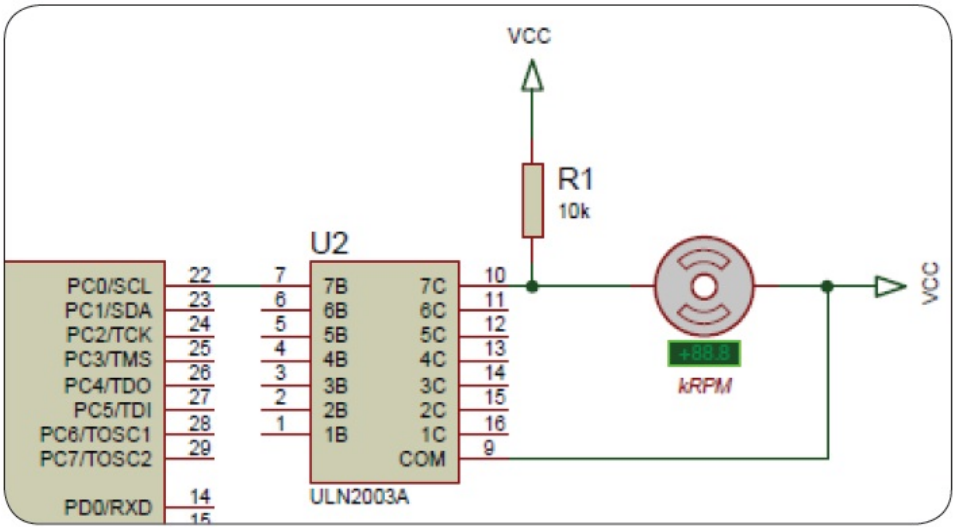


نکته قابل توجه این است که ماکزیمم جریانی که ترانزیستور در این مدار می تواند عبور دهد برابر است با حاصلضرب جریان بیس در ضریب تقویت ترانزیستور. بنابراین هرچه مقدار R بزرگتر باشد جریان بیس کمتر شده و به مراتب جریان کلکتور هم کاهش می یابد.

همچنین شما باید با توجه به موتور مورد استفاده و جریان راه اندازی مورد نیاز آن، ترانزیستور خود را انتخاب نمایید.

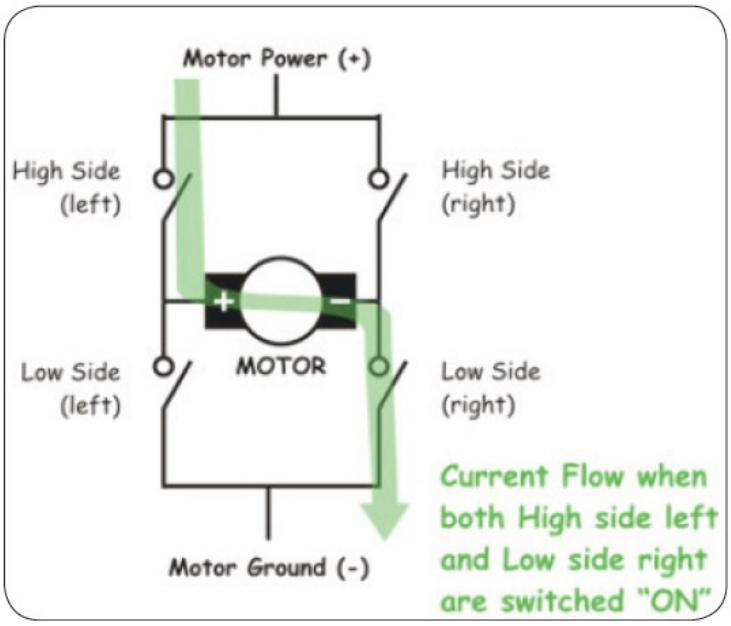
b- درایور یک طرفه با بافر

روش دیگر کنترل یک طرفه موتور DC استفاده از بافرها می باشد. بافرها گیت های منطقی می باشند که سطح منطق ورودی و خروجیشان یکسان می باشد و تنها وظیفه آن ها تقویت جریان است. از معروف ترین آی سی های بافر می توان به ULN2003 وULN2803 اشاره کرد. این تراشه ها توانایی تولید جریان ۵۰۰ میلی آمپر را در خروجی خود دارند و در صورت نیاز به جریان بیشتر می توان خروجی ها را با هم موازی کرد.



۲- درایورهای دو طرفه

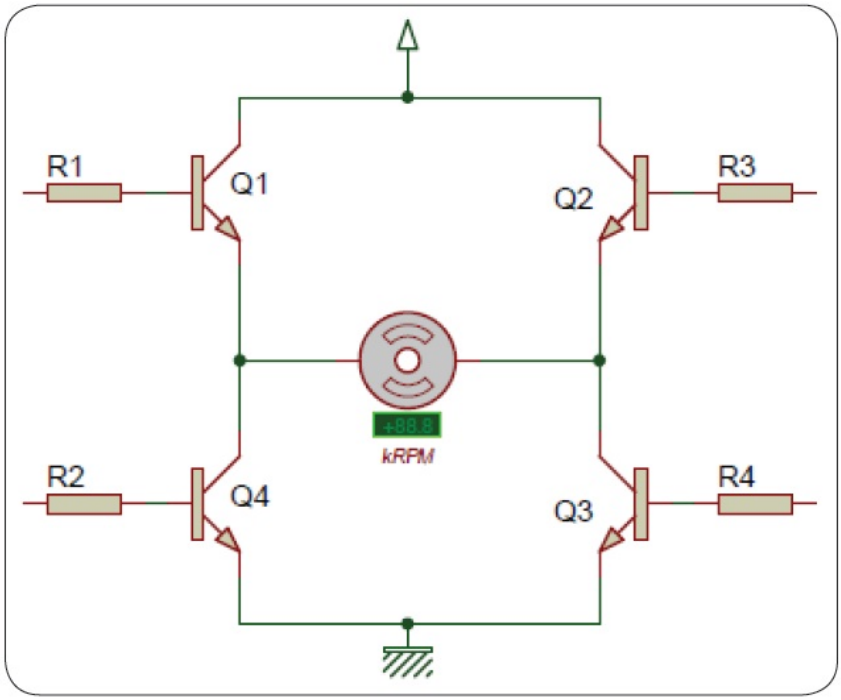
برای اینکه بتوان جهت گردش یک موتور DC را تغییر داد باید از طریق درایور بتوان خطوط تغذیه مثبت و منفی را روی پایه های موتور سویچ کرد. ساده ترین درایور برای این کار استفاده از ۴ کلید به صورت شکل زیر است.



اما همانطور که گفته شدما نیازمند یک سیستم الکترونیکی هستیم، پس به جای کلیدهای فوق از ترانزیستور استفاده می کنیم.

a- درایور پل H با استفاده از ترانزیستور

در مدار شکل زیر از چهار ترانزیستور جهت تغییر گردش موتور استفاده شده است. به این مدارات که شکل ظاهری آن ها شبیه به حرف H می باشد، اصطلاحا پل H (H Bridge) گفته می شود.



نحوه کارکرد مدار به این صورت است که اگر ترانزیستورهای Q1 و Q3 با هم تحریک شوند موتور در یک جهت شروع به حرکت خواهد کرد و اگر ترانزیستورهای Q2 و Q4 با هم روشن شوند موتور در جهت معکوس خواهد چرخید.

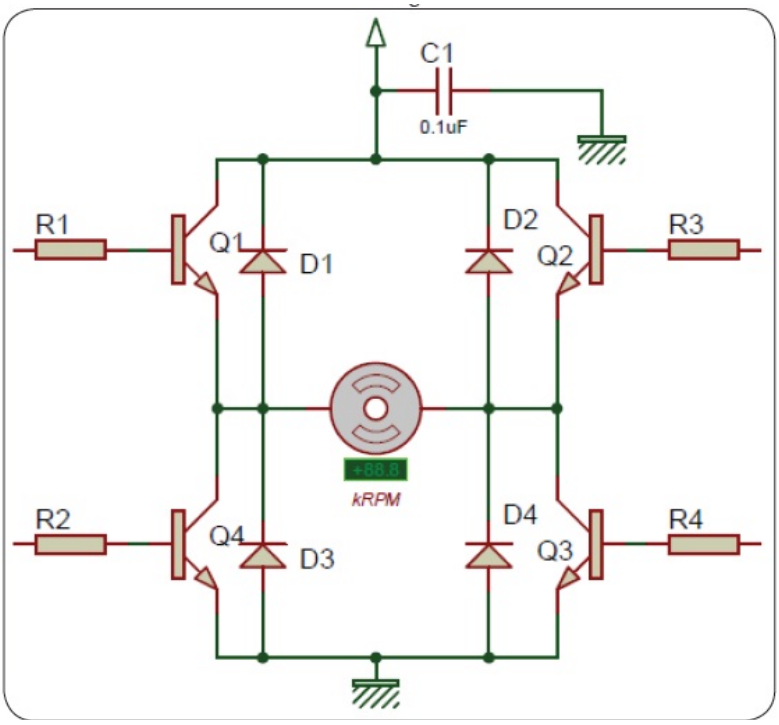
این مدار دارای ایراداتی است که باعث می شود کاربرد آن در مدارات به حداقل برسد. حال در زیر به رفع ایرادات موجود خواهیم پرداخت.

اثر نیروی محرکه القایی بازگشتی

موتورهای DC القاگرهای قدرتمندی هستند و خاصیت القایی آنها مثایل زیادی را برای مدارات رباتیک به وجود می آورد. القاگرها در مقابل تغییرات جریان از خود مقاومت نشان می دهند و هنگامی که یک القاگر به منبع تغذیه متصل باشد و ولتاژ دو سر آن ناگهانی قطع شود، القاگر برای ثابت نگه داشتن جریان عبوری تلاش می کند و این تلاش باعث ایجاد یک اختلاف پتانسیل خیلی بزرگتر از ولتاژ اولیه در دو سر آن خواهد شد. جهت این ولتاژ تولید شده عکس جهت ولتاژ دو سر القاگر است و به نیرو محرکه القایی بازگشتی یا

CEMF معروف می باشد. این ولتاژ بزرگ می تواند ترانزیستورهای راه انداز را بسوزاند. متداولترین روش برای حذف این ولتاژ استفاده از دیودهایی به صورت موازی با موتور است که اصطلاحا به این دیودهای بازگشتی دیودهای هرزگرد نیز گفته می شود.

دیودهای بازگشتی با اینکه مشکل سوختن درایورها را از بین می برند اما خود باعث به وجود آمدن مسئله ای دیگر می شوند. چراکه موازی شدن مسیر جریان بازگشتی با مسیر اصلی تغذیه باعث ایجاد یک ولتاژ گذرا در مدارات می شود که می تواند منشا تولید نویز باشد. برای حل این مشکل با مسیر تغذیه اصلی از خازن های بای پس استفاده می کنیم. خازن بای پس یک خازن عادی است که برای صاف کردن ریپل ها و نوسانات موقتی که بر روی تغذیه اصلی می افتد استفاده می شود. مقدار این خازن می تواند بین 0.1uf تا 0.22uf باشد. در شکل زیر مدار کامل شده پل H را می بینید.



نکته: دیودهای مورد استفاده بهتر است از نوع سرعت بالا(شاتکی) باشند.

نکته: در مدار شکل بالا باید دقت شود که هیچگاه دو ترانزیستوری که در یک ستون قرار دارند باهم روشن نشوند چون دراین صورت باعث اتصال کوتاه شدن منبع و سوختن ترانزیستورها می شود. برای جلوگیری از این عمل چند راهکار وجود دارد که مختصرا به توضیح هرکدام می پردازیم.

۱) دقت در برنامه نویسی

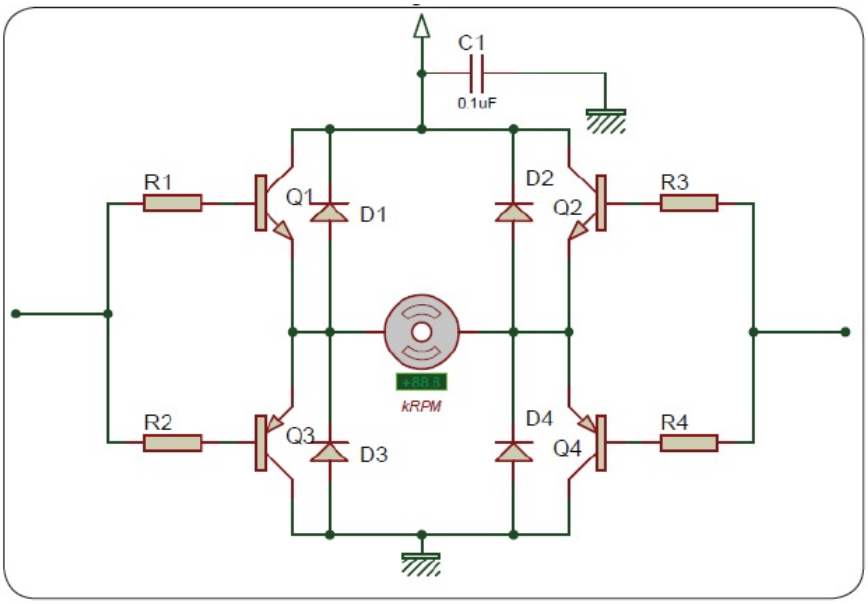
موقع برنامه نویسی پردازنده دقت شود که ترانزیستورهای موجود در یک ستون باهم روشن نشوند. این روش عملا قابل اطمینان نیست.

۲) استفاده از گیت NOT

برای جلوگیری از اتصال کوتاه در ستون های پل می توان یک گیت را بین ورودی های ترانزیستورهای هر کدام از ستون ها قرار داد.

۳) استفاده از ترانزیستورهای معکوس

بهترین راهکار استفاده از تزانزیستورهای معکوس می باشد بدین ترتیب که در ردیف بالای پل از ترانزیستورهای NPN و در ردیف پایین از ترازیستورهای PNP استفاده کرد.با این کار اگر به ستون ها ولتاژ مشترک داده شود همواره فقط یکی از ترانزیستورها روشن خواهد شد و عملا از اتصال کوتاه شدن منبع جلوگیری می شود.



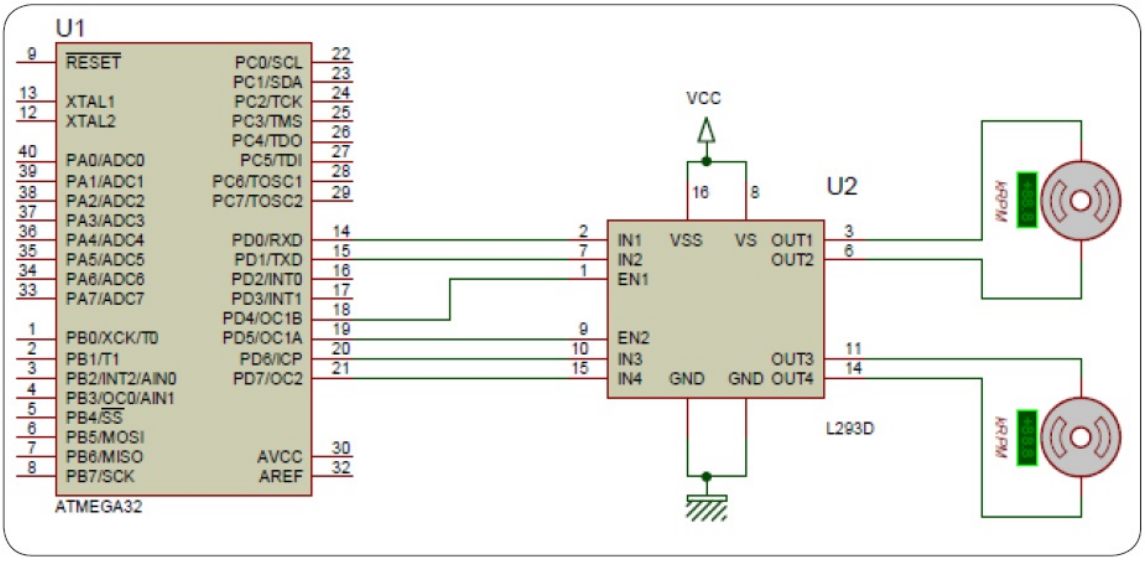
b. آی سی های پل H

به منظور استفاده راحت تر از پل های H این مدارات را به صورت مجتمع تحت عنوان آی سی های درایور طراحی کرده اند. آی سی های L293 و L298 از معروف ترین و متداول ترین آی سی های درایور می باشند.

الف) آی سی L293

این تراشه شامل دو پل H داخلی می باشد و توانایی راه اندازی دو موتور مجزا را دارا می باشد. ماکزیمم جریان عبوری از آن در حالت لحظه ای ۱.۲ آمپر و در حالت دائم ۶۰۰ میلی آمپر است.

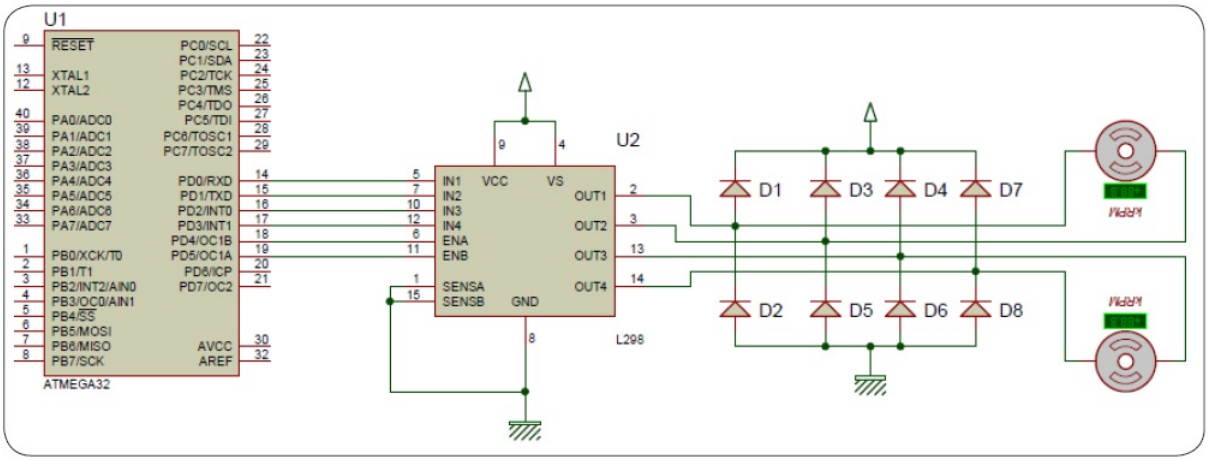
از مزیت های این تراشه می توان به وجود دیودهای هرزگرد داخلی آن اشاره کرد. همچنین این تراشه دارای پایه ای به اسم Enable جهت کنترل دور موتور و اعمال پالس PWM می باشد.



نکته: در صورتی که ولتاژ موتور مورد استفاده بیشتر از VCC باشد مو توانید ولتاژ مورد نیاز را به پایه VS اعمال کنید. توجه کنید که ماکزیمم ولتاژ اعمالی باد کمتر از ۳۶ ولت باشد.

ب) آی سی L298

این تراشه دقیقا شبیه به L293 عمل می کند با این تفاوت که جریان عبوری لحظه ای برابر ۴ آمپر و جریان پیوسته تا ۲ آمپر را تامین خواهد کرد. این تراشه فاقد دیودهای هرزگرد داخلی بوده و باید آن ها را از خارج به خروجی های تراشه متصل کرد تا ازآسیب دیدن آن جلوگیری شود.



نکته: در صورتی که ولتاژ موتور مورد استفاده بیشتر از VCC باشد مو توانید ولتاژ مورد نیاز را به پایه VS اعمال کنید. توجه کنید که ماکزیمم ولتاژ اعمالی باد کمتر از ۴۶ ولت باشد.

نکته: پایه های SensA و SensB برای کنترل جریان اعمالی به موتور می باشد.

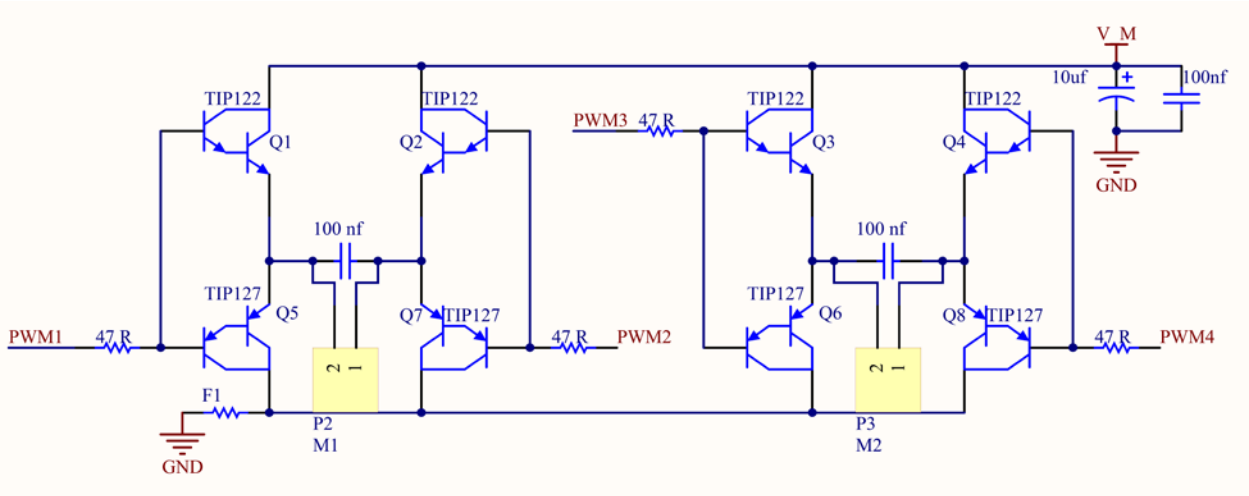
مقایسه ای بین پل H ساخته شده با ترانزیستور و آی سی های پل H

در مواقعی که جریان اعمالی به موتور کم باشد(در حد ۴ آمپر) آی سی های پل بهترین گزینه می باشند. اما در بسیاری از موارد جریان مورد نیاز بیشتر از توان تراشه های پل می باشد که در این مواقع می توان از ترانزیستورهای توان بالا استفاده کرد. البته گفتنی است که حتی در مواردی که جریان کم هم مورد نیاز باشد می توان از تراتزیستورها استفاده کرد و حتی این کار باعث کاهش هزینه ها تا حد زیادی خواهد شد. استفاده از ترانزیستورها معایبی هم دارد که بارزترین آن سرعت سویچ کم در حالت استفاده از موج PWM جهت کنترل دور موتور می باشد. و در موقع استفاده از ترانزیستورهای قدرت فرکانس کم (مانند BD139) باید فرکانس موج PWM را کمتر از ۱۰ کیلوهرتز در نظر گرفت.

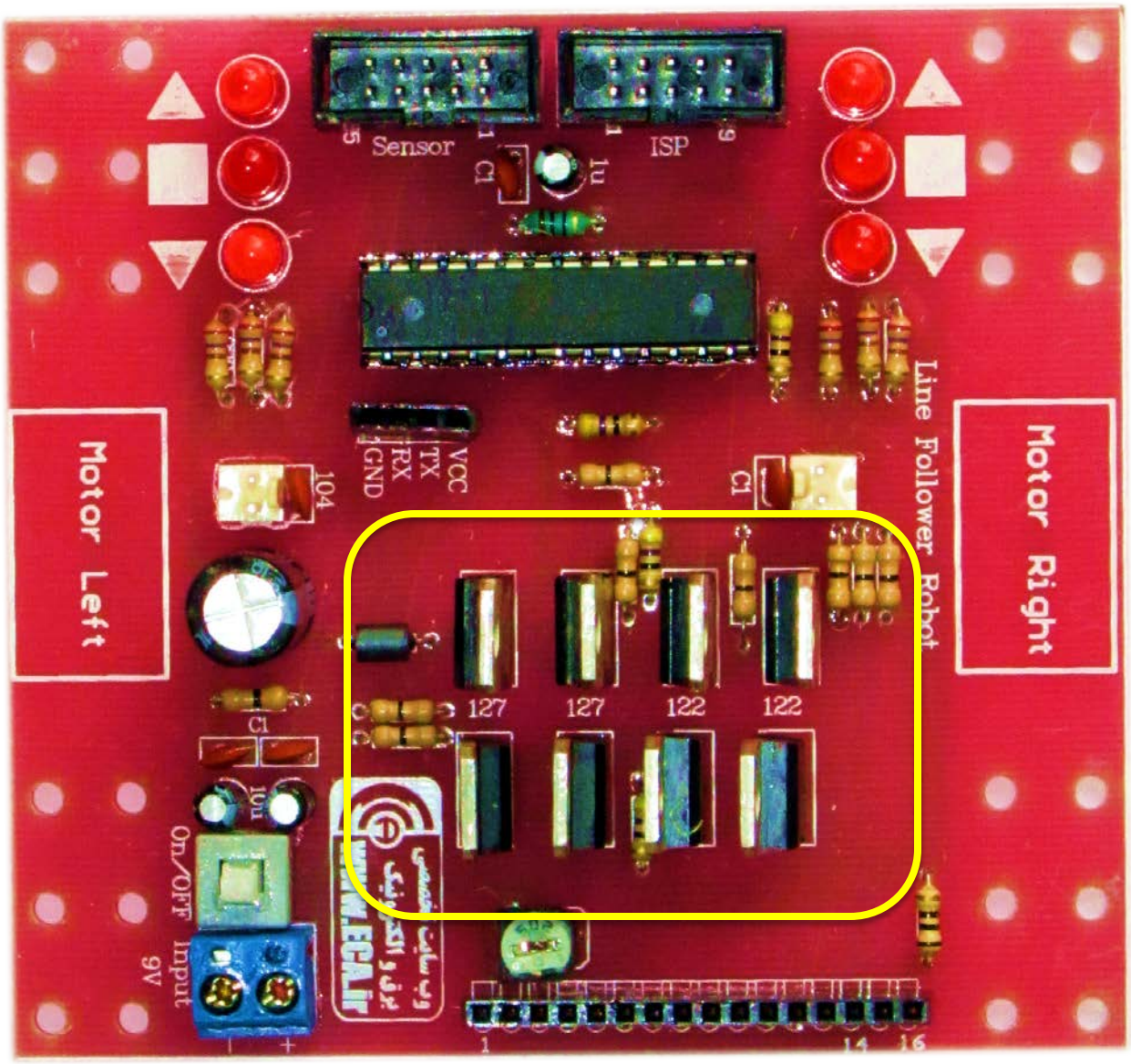
درایور استفاده شده در برد آموزشی

در برد آموزشی ربات برای تامین جریان موتورها و راه اندازی آن ها از درایور پلH با استفاده از ترانزیستورهای Tip127 و Tip122 که زوج دارلینگتون و مکمل هم می باشند استفاده شده است. دلیل استفاده از این درایور تامین جریان های بالا می باشد تا در صورت تمایل کاربر و استفاده از موتورهای قوی تر مشکلی در تامین جریان نداشته باشد.

شماتیک مدار راه انداز در زیر آورده شده است:



همانطور که مشاهده می کنید در این مدار از دیودهای هرزگرد خارجی استفاده نشده و این دیودها به صورت داخلی در ترانزیستورها قرار دارند.



۵)مکانیک ربات

اگر می خواهید ربات شما متولد شود، یک مرحله اساسی باقی مانده که اگر اهمیت آن از آنچه تا کنون انجام داده اید بیشتر نباشد، بدون شک کمتر هم نیست. اگر فرض کنید تاکنون عملایک میلیون ربات در دنیا ساخته شده باشند، به جرأت می توان گفت ده میلیون ربات به دلیل ساختار مکانی کی ضعیف که نتوانسته فرمان های کنترلی را به خوبی اجرا کند، از کار بازمانده اند. این نکته را فراموش نکنید که هرچقدر سنسورها و مدار کنترلی ربات شما خوب کار کنند، هرچقدر از الگوریتم های پیشرفته و پردازنده های سریع استفاده کنید و هرچقدر موتورهای ربات شما گرانقیمت و کارا باشند، اگر بدنه ربات طوری باشد

که فرمانهای نهایی بخش پردازشگر را به خوبی اجرا نکند یا موتورها را به خوبی هدایت نکند، ربات شما تنها باعث تأسف شما خواهد شد!

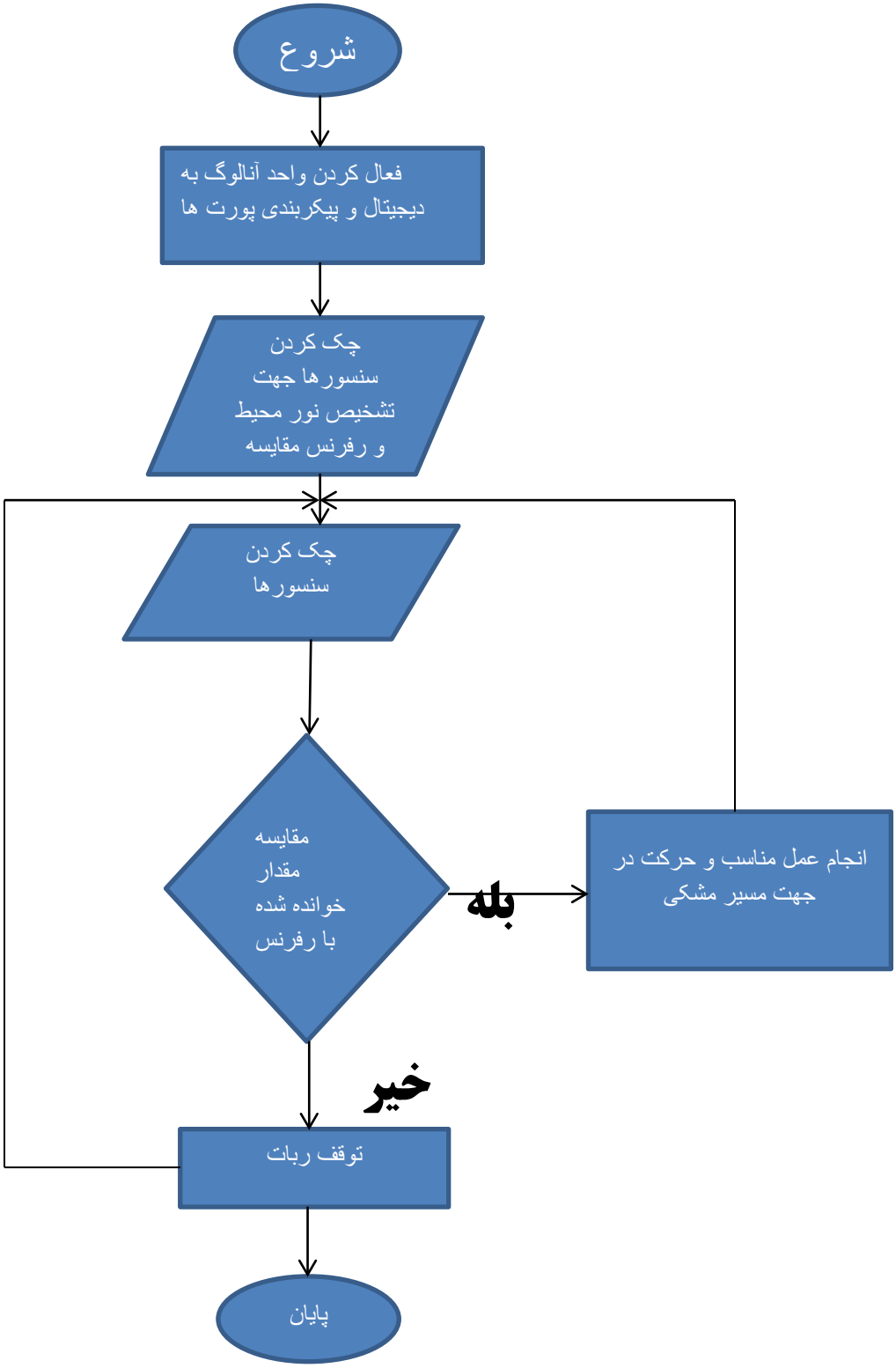
مکانیک ربات بحث پیشرفته ای از رشته مکانیک است که در آن مباحث زیاد و متنوعی مطرح می شود. برای یک ربات ساده تعقیب خط، خود شما می توانید طرحهای بسیار جالب و متنوعی تدارک ببینید که به شرط صرف زمان مناسب و دقت در طراحی، همگی می توانند به خوبی کار کنند. به همین لحاظ در اینجا تنها به چند پیشنهاد اکتفا می کنیم.

- ساده ترین طرح بر ای ساخت بدنه ربات تعقیب خط، استفاده از دو موتور است. این موتورها را می توانید مستقیما به چرخهای روبات متصل کنید.
- از هر موتوری نمی توان برای راه اندازی ربات استفاده کرد. توان موتور خود را متناسب با وزن بدنه ای که می خواهید بسازید انتخاب کنید.
- ربات و چرخهای آن باید از ماده ای سبک با استحکام بالا (مثلاً آلمینیوم یا پلاستیک) ساخته شود. برای اصطکاک بهتر چرخ ها با سطحی که باید روی آن حرکت کند، در رویه خروجی چرخ ها از پوشش های لاستیکی آجدار استفاده کنید.
- محل سنسورها بستگی به مدار کنترلی و مکانیک ربات و سرعت آن دارد. نصب سنسورها در قسمت زیرین جلوی ربات میتواند انتخاب خوبی باشد.
- یک ربات تعقیب خط باید در کمترین زمان به انتهای خط برسد. هیچ گاه سرعت را فدای دقت و دقت را فدای سرعت نکنید. رباتی که بسیار سریع حرکت کند، اما بعد از چند پیچ و انحراف خط را گم کند و رباتی که با حرکت لاکپشتی تمام خط را طی کند، هیچکدام مطلوب نیست!
- هنگام طراحی بدنه ربات به مسأله نحوه تغذیه مدارهای ربات دقت کنید. اگر از باطری استفاده می کنید، وزن آن را نیز به وزن بدنه و مدارات بیفزایید. اگر از منبع تغذیه خارجی استفاده می کنید، باید سیم تغذیه را که به ربات متصل می کنید در حین حرکت آن به دنبالش حمل کنید. بنابراین محل مناسبی برای اتصال سیم تغذیه به ربات تعبیه کنید.

ما در برد آموزشی از یک مکانیک ساده و کاربردی استفاده کرده ایم به گونه ای که این مکانیک نه تنها در امر آموزش می تواند بسیار مفید واقع شود بلکه با تغییرات جزئی می تواند در مسابقات رباتیک نیز مورد استفاده قرار بگیرد.

حال می خواهیم برنامه ای برای ربات بنویسیم که از طریق آن بتواند مسیر مشکی را از سفید تشخیص دهد. ابتدا فلوچارتی از عملکرد برنامه را می کشیم و سپس برنامه را به دو زبان سی و بیسیک نوشته و توضیح می دهیم.

فلوچارت برنامه



```
#include <mega48a.h>
#include <delay.h>
#include <alcd.h>
#include <stdlib.h>
#define xtal 8000000
//**************************************
#define adc 5
//**************************************
#define FIRST_ADC_INPUT 1
#define LAST_ADC_INPUT 5
unsigned int adc_data[LAST_ADC_INPUT-FIRST_ADC_INPUT+1];
#define ADC_VREF_TYPE 0x40
interrupt [ADC_INT] void adc_isr(void)
{
static unsigned char input_index=0;
adc_data[input_index]=ADCW;
if (++input_index > (LAST_ADC_INPUT-FIRST_ADC_INPUT))
input_index=0;
ADMUX=(FIRST_ADC_INPUT | (ADC_VREF_TYPE & 0xff))+input_index;
delay_us(10);
ADCSRA|=0x40;
}
//**************************************
void forward(void){
PORTB=0x12;
PORTD=0xA0;
}
void right(void){
PORTB=0x06;
PORTD=0xc0;
}
void left(void){
PORTB=0x10;
PORTD=0x38;
}
void stop(void){
PORTB=0x08;
PORTD=0x04;
}
//**************************************
unsigned int ref[5];
unsigned int i;
//**************************************
```

```
void main(void)
{
DDRB=0xff;
PORTB=0x00;
DDRD=0xff;
PORTD=0x00;
#pragma optsize-
CLKPR=0x80;
CLKPR=0x00;
#ifdef _OPTIMIZE_SIZE_
#pragma optsize+
#endif
DIDR0=0x00;
ADMUX=FIRST_ADC_INPUT | (ADC_VREF_TYPE & 0xff);
ADCSRA=0xED;
ADCSRB&=0xF8;
lcd_init(16);
lcd_gotoxy(3,0);
lcd_putsf("www.ECA.ir");
lcd_gotoxy(1,1);
lcd_putsf("Robot Training");
delay_ms(500);
#asm("sei")
for(i=0;i<=4;i++)
{
ref[i]=adc_data[i]-adc;
delay_ms(100);
}
//**************************************
while (1)
{
if(adc_data[0]<=ref[0])
{
left();
while(adc_data[0]<=ref[0]);
}
if(adc_data[3]<=ref[3])
{
forward();
while(adc_data[3]<=ref[3]);
}
if(adc_data[2]<=ref[2])
{
right();
while(adc_data[2]<=ref[2]);
```

```
}
if(adc_data[4]<=ref[4])
{
left();
while(adc_data[4]<=ref[4]);
}
if(adc_data[1]<=ref[1])
{
right();
while(adc_data[1]<=ref[1]);
}
if((adc_data[3]<=ref[3]) && (adc_data[2]<=ref[2]) && (adc_data[4]<=ref[4]))
{
forward();
while((adc_data[3]<=ref[3]) && (adc_data[2]<=ref[2]) && (adc_data[4]<=ref[4]));
}
if((adc_data[3]<=ref[3]) && (adc_data[2]<=ref[2]))
{
right();
while((adc_data[3]<=ref[3]) && (adc_data[2]<=ref[2]));
}
if((adc_data[3]<=ref[3]) && (adc_data[4]<=ref[4]))
{
left();
while((adc_data[3]<=ref[3]) && (adc_data[4]<=ref[4]));
}
if((adc_data[0]<=ref[0]) && (adc_data[4]<=ref[4]))
{
left();
while((adc_data[0]<=ref[0]) && (adc_data[4]<=ref[4]));
}
if((adc_data[1]<=ref[1]) && (adc_data[2]<=ref[2]))
{
right();
while((adc_data[1]<=ref[1]) && (adc_data[2]<=ref[2]));
}
if((adc_data[3]<=ref[3]) && (adc_data[2]<=ref[2]) && (adc_data[4]<=ref[4]) &&
(adc_data[1]<=ref[1]) && (adc_data[0]<=ref[0]))
{
stop();
while((adc_data[3]<=ref[3]) && (adc_data[2]<=ref[2]) && (adc_data[4]<=ref[4]) &&
(adc_data[1]<=ref[1]) && (adc_data[0]<=ref[0]));
}
else stop();
}
```

```
}
```

توضیحات مربوط به برنامه

```
#include <mega48a.h>
#include <delay.h>
#include <alcd.h>
#include <stdlib.h>
#define xtal 8000000
```

در این ۵ خط ابتدا میکروکنترلر مورد استفاده به کامپایلر معرفی شده و سپس فایل هدرهای مورد نیاز از قبیل هدر LCD، تاخیر و کتابخانه استاندارد تبدیلات به برنامه استفاده شده و در پایان فرکانس کاری میکروکنترلر برای کامپایلر مشخص شده است.

نکته: برای استفاده از اسیلاتور داخلی ۸ مگاهرتز باید از طریق فیوزبیت ها عمل کنید و خط بالا صرفا جهت عملکرد صحیح توابع تاخیر می باشد.

```
#define adc 5
```

در این خط ماکرویی تعریف شده تا توسط آن و تفریق با مقدار ADC اولیه که بعدا بیشتر توضیح داده خواهد شد دقت عملکرد ربات را تعیین کرد.

```
#define FIRST_ADC_INPUT 1
#define LAST_ADC_INPUT 5
unsigned int adc_data[LAST_ADC_INPUT-FIRST_ADC_INPUT+1];
#define ADC_VREF_TYPE 0x40
interrupt [ADC_INT] void adc_isr(void)
{
static unsigned char input_index=0;
adc_data[input_index]=ADCW;
if (++input_index > (LAST_ADC_INPUT-FIRST_ADC_INPUT))
  input_index=0;
ADMUX=(FIRST_ADC_INPUT | (ADC_VREF_TYPE & 0xff))+input_index;
delay_us(10);
ADCSRA|=0x40;
}
```

با استفاده از خطوط بالا به طور اتوماتیک کانال های ADC میکروکنترلر چک می شود و در آرایه ای به اسم adc_data ذخیره می شوند.

```
void forward(void){
```


فروشگاه تخصصی برق و الکترونیک www.eShop.ECA.ir ایمیل: eShop.ECA@Gmail.com
شماره تماس: ۰۴۱۱۵۵۳۳۷۳۲ فکس: ۰۴۱۱۵۵۳۹۷۶۹

```
PORTB=0x12;
PORTD=0xA0;
}
void right(void){
PORTB=0x06;
PORTD=0xc0;
}
void left(void){
PORTB=0x10;
PORTD=0x38;
}
void stop(void){
PORTB=0x08;
PORTD=0x04;
}
```

چهار تابع جهت حرکت سه گانه ربات و نیز توقف آن تعریف شده اند و در برنامه و طی شرط های مختلفی که قرار داده می شود با فراخوانی این توابع ربات در جهت مناسب حرکت خواهد کرد.

```
unsigned int ref[5];
unsigned int i;
```

دو متغییر سراسری که اولی به صورت آرایه و دومی متغیری از جنس int می باشد.

متغیر اول جهت اسکن اولیه محیط و تطبیق ربات با محیط می باشد و متغیر دوم در یک حلقه for کاربرد دارد و مقدار اولیه و اندیس حلقه را تشکیل می دهد.

```
DDRB=0xff;
PORTB=0x00;
DDRD=0xff;
PORTD=0x00;
#pragma optsize-
CLKPR=0x80;
CLKPR=0x00;
#ifdef _OPTIMIZE_SIZE_
#pragma optsize+
#endif
DIDR0=0x00;
ADMUX=FIRST_ADC_INPUT | (ADC_VREF_TYPE & 0xff);
ADCSRA=0xED;
ADCSRB&=0xF8;
```

در این خطوط که در تابع main اصلی می باشند پورت ها و ADC میکروکنترلر پیکربندی شده اند.

```
#asm("sei")
```

```
for(i=0;i<=4;i++)
{
   ref[i]=adc_data[i]-adc;
   delay_ms(100);
}
```

در این قسمت ابتدا وقفه سراسری جهت عملکرد ADC فعال شده و سپس در یک حلقه for که در بالا و قسمت تعریف متغیر بحث شد محیط برای بار اول چک شده و رفرنس هایی جهت مقایسه با مقدار خوانده شده ADC در حلقه بینهایت را تشکیل می دهند.

```
while (1)
{
if(adc_data[0]<=ref[0])
{
left();
while(adc_data[0]<=ref[0]);
}
if(adc_data[3]<=ref[3])
{
forward();
while(adc_data[3]<=ref[3]);
}
if(adc_data[2]<=ref[2])
{
right();
while(adc_data[2]<=ref[2]);
}
if(adc_data[4]<=ref[4])
{
left();
while(adc_data[4]<=ref[4]);
}
if(adc_data[1]<=ref[1])
{
right();
while(adc_data[1]<=ref[1]);
}
if((adc_data[3]<=ref[3]) && (adc_data[2]<=ref[2]) && (adc_data[4]<=ref[4]))
{
forward();
while((adc_data[3]<=ref[3]) && (adc_data[2]<=ref[2]) && (adc_data[4]<=ref[4]));
}
if((adc_data[3]<=ref[3]) && (adc_data[2]<=ref[2]))
{
right();
while((adc_data[3]<=ref[3]) && (adc_data[2]<=ref[2]));
}
if((adc_data[3]<=ref[3]) && (adc_data[4]<=ref[4]))
```

```
{
left();
while((adc_data[3]<=ref[3]) && (adc_data[4]<=ref[4]));
}
if((adc_data[0]<=ref[0]) && (adc_data[4]<=ref[4]))
{
left();
while((adc_data[0]<=ref[0]) && (adc_data[4]<=ref[4]));
}
if((adc_data[1]<=ref[1]) && (adc_data[2]<=ref[2]))
{
right();
while((adc_data[1]<=ref[1]) && (adc_data[2]<=ref[2]));
}
if((adc_data[3]<=ref[3]) && (adc_data[2]<=ref[2]) && (adc_data[4]<=ref[4]) && (adc_data[1]<=ref[1])
&& (adc_data[0]<=ref[0]))
{
stop();
while((adc_data[3]<=ref[3]) && (adc_data[2]<=ref[2]) && (adc_data[4]<=ref[4]) &&
(adc_data[1]<=ref[1]) && (adc_data[0]<=ref[0]));
}
else stop();
}
```

در یک حلقه بینهایت و با استفاده از چندین شرط if سنسورها مدام چک شده و با مقدار رفرنس که از حلقه for به دست آمد مقایسه می شوند و عملکرد مناسب با توجه به گفته های بالا در مورد بازگشت نور از اجسام سفید و مشکی توسط میکروکنترلر اتخاذ خواهد شد و ربات در جهت مناسب به حرکت درخواهد آمد.

فروشگاه تخصصی برق و الکترونیک www.eShop.ECA.ir ایمیل: eShop.ECA@Gmail.com
شماره تماس: ۰۴۱۱۵۵۳۳۷۳۲ فکس: ۰۴۱۱۵۵۳۹۷۶۹

```
$regfile = "m48def.dat"
$crystal = 8000000
'-----------------------------
Config Portb.2 = Output
Portb.2 = 0
Config Lcd = 16 * 2
Config Lcdpin = Pin , Rs = Portb.0 , E = Portb.3 , Db4 = Portb.4 , Db5 = Portb.5 , Db6 =
Portb.6 , Db7 = Portb.7
Initlcd
Cursor Off
'-----------------------------
Config Portb = Output
Config Portd = Output
'-----------------------------
Config Adc = Single , Prescaler = Auto , Reference = Avcc
Enable Adc
Start Adc
'-----------------------------
Dim Ref(5) As Word
Dim I As Byte
Dim Adc_data(5) As Word
Const Ad = 5
'-----------------------------
Cls
Locate 1 , 4 : Lcd "www.ECA.ir"
Locate 2 , 2 : Lcd "Robot Training"
Wait 1
'-----------------------------
Ref(1) = Getadc(1) - Ad
Ref(2) = Getadc(2) - Ad
Ref(3) = Getadc(3) - Ad
Ref(4) = Getadc(4) - Ad
Ref(5) = Getadc(5) - ad
'-----------------------------
Do
Adc_data(1) = Getadc(1)
Adc_data(2) = Getadc(2)
Adc_data(3) = Getadc(3)
```

```
Adc_data(4) = Getadc(4)
Adc_data(5) = Getadc(5)
If Adc_data(3) < Ref(3) Then
Portb = &H12
Portd = &HA0
End If
If Adc_data(3) < Ref(3) And Adc_data(2) < Ref(2) And Adc_data(4) < Ref(4) Then
Portb = &H12
Portd = &HA0
End If
If Adc_data(1) < Ref(1) And Adc_data(2) < Ref(2) And Adc_data(3) < Ref(3) And
Adc_data(4) < Ref(4) And Adc_data(5) < Ref(5) Then
Portb = &H08
Portd = &H04
End If
If Adc_data(1) < Ref(1) Then
Portb = &H06
Portd = &HC0
End If
If Adc_data(2) < Ref(2) Then
Portb = &H06
Portd = &HC0
End If
If Adc_data(1) < Ref(1) And Adc_data(2) < Ref(2) Then
Portb = &H06
Portd = &HC0
End If
If Adc_data(3) < Ref(3) And Adc_data(2) < Ref(2) Then
Portb = &H06
Portd = &HC0
End If
If Adc_data(5) < Ref(5) Then
Portb = &H10
Portd = &H38
End If
If Adc_data(4) < Ref(4) Then
Portb = &H10
Portd = &H38
End If
If Adc_data(5) < Ref(5) And Adc_data(4) < Ref(4) Then
Portb = &H10
Portd = &H38
```

```
End If
If Adc_data(3) < Ref(3) And Adc_data(4) < Ref(4) Then
Portb = &H10
Portd = &H38
End If
If Adc_data(1) > Ref(1) And Adc_data(2) > Ref(2) And Adc_data(3) > Ref(3) And
Adc_data(4) > Ref(4) And Adc_data(5) > Ref(5) Then
Portb = &H08
Portd = &H04
End If
Loop
End  'end program
```

توضیحات مربوط به برنامه

```
$regfile = "m48def.dat"
$crystal = 8000000
'-----------------------------
Config Portb.2 = Output
Portb.2 = 0
Config Lcd = 16 * 2
Config Lcdpin = Pin , Rs = Portb.0 , E = Portb.3 , Db4 = Portb.4 , Db5 = Portb.5 , Db6 = Portb.6
, Db7 = Portb.7
Initlcd
Cursor Off
'-----------------------------
Config Portb = Output
Config Portd = Output
'-----------------------------
Config Adc = Single , Prescaler = Auto , Reference = Avcc
Enable Adc
Start Adc
```

در خطوط بالا ابتدا تراشه مورد استفاده به برنامه معرفی شده و فرکانس کاری آن را در خط بعد برای کامپایلر مشخص کرده ایم. در خطوط بعد پین B.2 که متصل به پین WR در نمایشگر کاراکتری می باشد خروجی شده تا ب صفر کردن آن بتوان اطلاعات را بر روی نمایشگر نشان داد. البته قبل از نمایش اطلاعات باید با استفاده از دستور initlcd یکبار نمایشگر را مقدار دهی اولیه کرد.

به همین ترتیب پورت های مورد نیاز خروجی شده و مبدل آنالوگ به دیجیتال میکروکنترلر نیز در حالت Single-ended با ولتاژ رفرنس گرفته شده از پایه AVCC پیکربندی شده است.

```
Dim Ref(5) As Word
Dim Adc_data(5) As Word
```

```
Const Ad = 5
```

آرایه ref همانند برنامه نوشته شده به زبان سی برای گرفتن اعداد مرجع مبدل آنالوگ به دیجیتال می باشد. و آرایه adc_data نیز برای خواندن اعداد تبدیل شده کانال های مبدل.

ثابتی که با عنوان ad تعریف شده در اینجا نیز همانند قبل برای تنظیم دقت ربات و تشخیص مسیر مورد استفاده قرار می گیرد.

با استفاده از آرایه ref و ثابت ad با تغییر نور محیط نیازی به تغییر هیچ چیزی در مدار نخواهد بود و ربات به صورت اتوماتیک نور محیط را چک کرده و به نسبت نور محیط تلاش در تشخیص قسمت مشکی از سفید را خواهد کرد.

```
Ref(1) = Getadc(1) - Ad
Ref(2) = Getadc(2) - Ad
Ref(3) = Getadc(3) - Ad
Ref(4) = Getadc(4) - Ad
Ref(5) = Getadc(5) – Ad
```

جهت بدست آوردن اعداد مرجعی که توضیح دادیم قبل از رفتن به حلقه بینهایت برنامه نور محیط با استفاده از خطوط بالا چک شده و در متغیر آرایه ای ref مربوطه قرار می گیرند.

```
Do
Adc_data(1) = Getadc(1)
Adc_data(2) = Getadc(2)
Adc_data(3) = Getadc(3)
Adc_data(4) = Getadc(4)
Adc_data(5) = Getadc(5)
If Adc_data(3) < Ref(3) Then
Portb = &H12
Portd = &HA0
End If
If Adc_data(3) < Ref(3) And Adc_data(2) < Ref(2) And Adc_data(4) < Ref(4)
Then
Portb = &H12
Portd = &HA0
End If
If Adc_data(1) < Ref(1) And Adc_data(2) < Ref(2) And Adc_data(3) < Ref(3) And
Adc_data(4) < Ref(4) And Adc_data(5) < Ref(5) Then
Portb = &H08
Portd = &H04
End If
```

```
If Adc_data(1) < Ref(1) Then
Portb = &H06
Portd = &HC0
End If
If Adc_data(2) < Ref(2) Then
Portb = &H06
Portd = &HC0
End If
If Adc_data(1) < Ref(1) And Adc_data(2) < Ref(2) Then
Portb = &H06
Portd = &HC0
End If
If Adc_data(3) < Ref(3) And Adc_data(2) < Ref(2) Then
Portb = &H06
Portd = &HC0
End If
If Adc_data(5) < Ref(5) Then
Portb = &H10
Portd = &H38
End If
If Adc_data(4) < Ref(4) Then
Portb = &H10
Portd = &H38
End If
If Adc_data(5) < Ref(5) And Adc_data(4) < Ref(4) Then
Portb = &H10
Portd = &H38
End If
If Adc_data(3) < Ref(3) And Adc_data(4) < Ref(4) Then
Portb = &H10
Portd = &H38
End If
If Adc_data(1) > Ref(1) And Adc_data(2) > Ref(2) And Adc_data(3) > Ref(3) And
Adc_data(4) > Ref(4) And Adc_data(5) > Ref(5) Then
Portb = &H08
Portd = &H04
End If
Loop
End
```

در حلقه بینهایت برنامه هربار کانال های ADC چک شده و طی شرط های مختلفی که برای ربات در نظر گرفته شده، ربات می تواند عملکرد مناسب را بر روی مسیر داشته باشد.

معرفی بخش های برد آموزشی ربات تعقیب خط

۱- کانکتور VDC برای تامین تغذیه میکروکنترلر و سایر بخش های مدار به جز موتورها می باشد و باید به ولتاژ بیشتر از ۸ ولت وصل شود.

۲- کانکتور VM برای تامین جریان موتورها بوده و باید به ولتاژ مناسب با نوع موتورها وصل شود. بدین ترتیب امکان اتصال موتورهایی با ولتاژ کاری ۵ تا ۴۶ ولت مهیا می گردد.

۳- سوئیچ ON/OFF برای روشن، خاموش کردن تغذیه برد می باشد.

۴- جامپر J1 برای مواقعی می باشد که تنها یک منبع تغذیه در دسترس باشد و با وصل کردن آن جریان موتورها از کانکتور VDC تامین می شود.

۵- این کانکتور برای اتصال برد به پورت COM کامپیوتر و سایر ماژول های با رابط UART می باشد. بخش RS232 به مبدل سطح ولتاژ MAX232 متصل بوده و برای اتصال به پورت COM می باشد و بخش TTL مستقیما به پورت سریال میکرو متصل می باشد.

۶- دیپ سوئیچ SW2 برای فعال کردن مبدل سطح ولتاژ MAX232

۷- کانکتور برای اتصال به موتورهای M1 و M2

۸- کانکتور متصل به پورت B و LED های روی برد

۹- دیپ سوئیچ تکی برای فعال و غیر فعال کردن تمامی LED های روی برد

۱۰- کانکتور ISP برای پروگرام نمودن میکرو با رابط ISP

۱۱- کانکتور برای اتصال برد کنترلی به برد سنسورها

۱۲- پتانسیومتر جهت تنظیم شدت نور LCD

۱۳- کانکتور جهت اتصال LCD