



گرد آورنده: رامین عالمیان راد
کیوان رئیسی
مریم بوشهری
نسرتن اردستانیان

باسمه تعالی

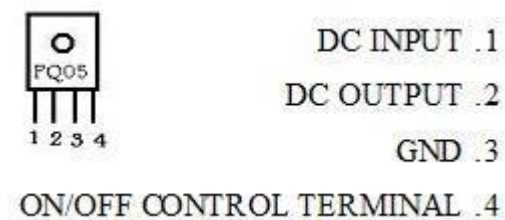
جلسه دوم آموزش الکترونیک

نحوه راه اندازی قطعات و طراحی آنها

در جلسه قبل با تعدادی از قطعات مختلف برای راه اندازی یک ربات آشنا شدیم . این جلسه قصد داریم نحوه راه اندازی یا درایو کردن این قطعات را بررسی کنیم .

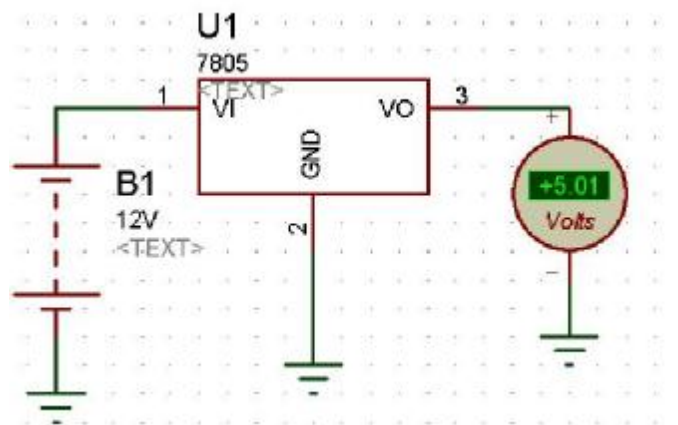
رگولاتور ولتاژ:

تفاوت رگولاتور PQ05 با رگولاتور یا 7805 در این است :



شکل ۱

همان طور که می دانیم برای راه اندازی ۷۸۰۵ ولتاژ ورودی نیاز به $8V$ حداقل ولتاژ دارد اما در PQ۰۵ ولتاژ ورودی می تواند $5.7V$ باشد .



شکل ۲

این ولتاژ کمتر سبب می شود که بتوانیم جریان بیشتری از رگولاتور دریافت کنیم. حال که مشخص شد چرا از این رگولاتور استفاده کردیم در بعضی از قسمت های مدار برای نویزگیری از یک خازن استفاده شده که نویز محیط را کامل بگیرد و نویزهای احتمالی در این قسمت از بین برود .

فرض کنید در شکل زیر جریان خروجی ۲ آمپر می باشد. میدانیم حداقل ولتاژ ورودی برای این رگولاتور ۸ ولت است. پس داریم:

$$P_i = (V_i)(I_i) = 8V \times 2A = 16W$$

$$P_o = (V_o)(I_o) = 5V \times 2A = 10W$$

$$P_d = P_i - P_o = 16W - 10W = 6W$$

مشاهده می کنید که ۶ وات توان در همان لحظه اول به گرما تبدیل می شود و تلفات زیادی داریم اما در آی سی PQ۰۵ حداقل ولتاژ ۵.۷ ولت می باشد پس داریم:

$$P_i = (V_i)(I_i) = 5.7V \times 2A = 11.4W$$

$$P_o = (V_o)(I_o) = 5V \times 2A = 10W$$

$$P_d = P_i - P_o = 11.4W - 10W = 1.4W$$

همان طور که مشاهده می کنید توان تلف شده کمتر است.

به این نکته توجه کنید چون در رگولاتور ها توان ثابت است هر چقدر ولتاژ کمتری به ورودی رگولاتور اعمال کنید جریان بیشتری را دریافت خواهید کرد.

برای مثال:

طی تجربیات عملی که بدست آوردیم توان رگولاتور ۱۴W-۱۳.۵ می باشد.

یعنی اگر ولتاژ ورودی رگولاتور را ۱۲V در نظر بگیریم:

$$P_i = V_i \cdot I_i \quad 14W = 12V \cdot I_i \quad I_i = 14W / 12V = 1.16A$$

حداکثر جریانی که می تواند رگولاتور تامین کند $1.16A$ می باشد .

حالا در نظر بگیرید که ولتاژ ورودی را به $9V$ کاهش دهیم خواهیم داشت:

$$P_i = V_i \cdot I_i \quad 14W = 9V \cdot I_i \quad I_i = 14W / 9V = 1.5A$$

مشاهده کردید که در حال حاضر بیشترین جریانی که می توان از رگولاتور گرفت به $1.5A$ افزایش یافت .

به این نکته توجه کنید که نمی توان به راحتی ولتاژ ورودی یک رگولاتور را کم کرد و باید به برگه های اطلاعاتی آن مراجعه کرد یعنی حداقل ولتاژ ورودی برای $7805IC$ $8V$ و برای $PQ05$ $5.7V$ می باشد پس دقت کنید.

در ادامه مطلب خواهیم گفت که چگونه رگولاتور خود را انتخاب کنیم که بتواند جریان مورد نظر را تامین کند.

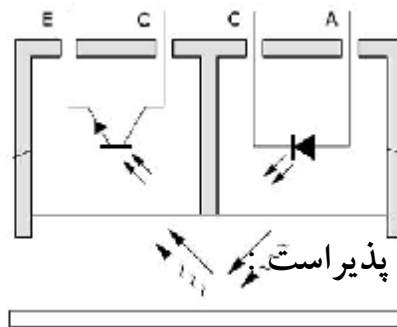
سنسور:

سنسور: IR

همان طور که می دانیم سنسورهای فاصله سنج یا IR مادون قرمز براساس طیف نوری (مادون قرمز) کار می کنند یعنی از دو قسمت گیرنده و فرستنده تشکیل شده اند.

قسمت فرستنده با فرستادن اشعه مادون قرمز (۳m و ۵m) و برخورد با جسم بازتاب می کند و به گیرنده می رسد که می تواند فاصله را تشخیص دهد در حالت عادی بدون درایو تا ۵cm قابل سنس است اما ۵cm نیاز ما را برطرف نمی کند .

در شکل زیر شمای فنی و حقیقی سنسور مادون قرمز را مشاهده می کنید :



برای راه اندازی آن ۳ طریق امکان پذیر است:

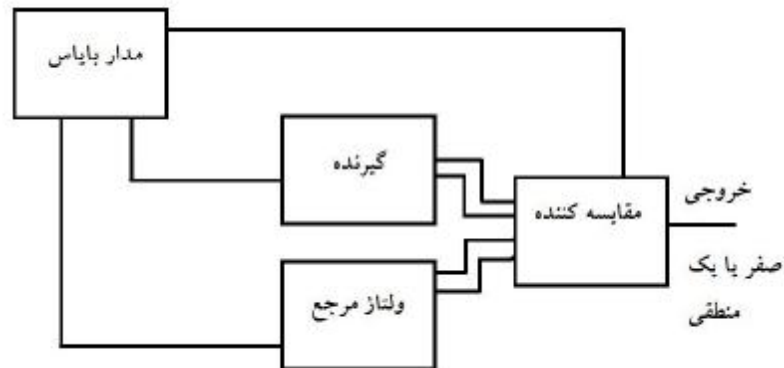
۱. تبدیل آنالوگ به دیجیتال

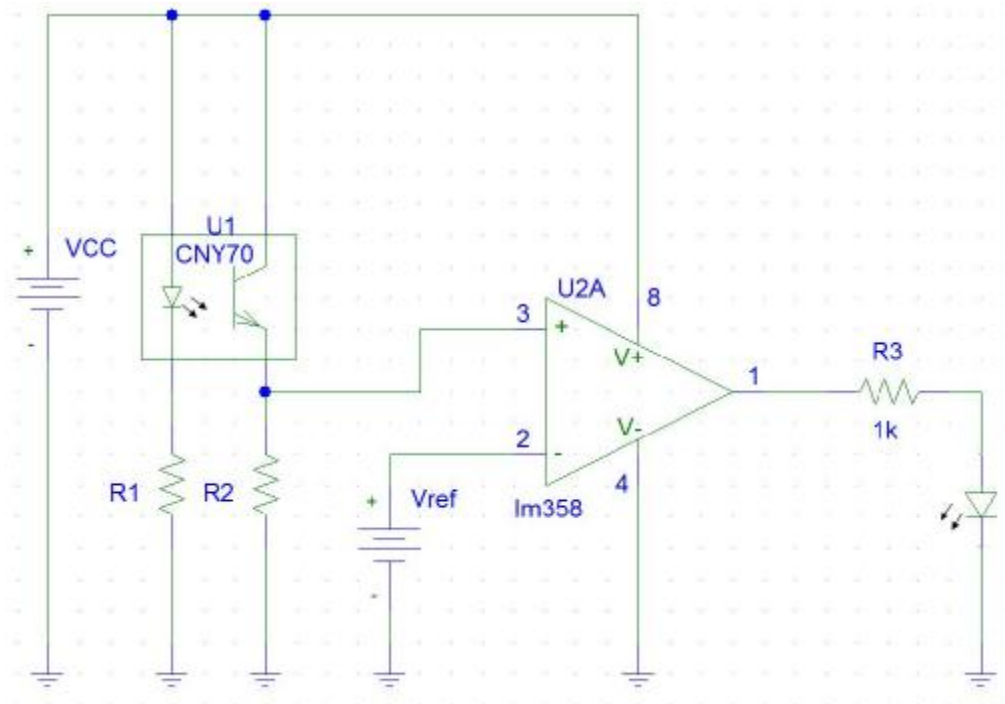
۲. اتصال به A/D میکرو

۳. اتصال به بافر

ساختار داخلی یک سنسور مادون قرمز یا فرستنده IR مانند LED در ورودی و خروجی آن یک فوتوترانزیستور است که از نوع PNP و NPN می باشد .

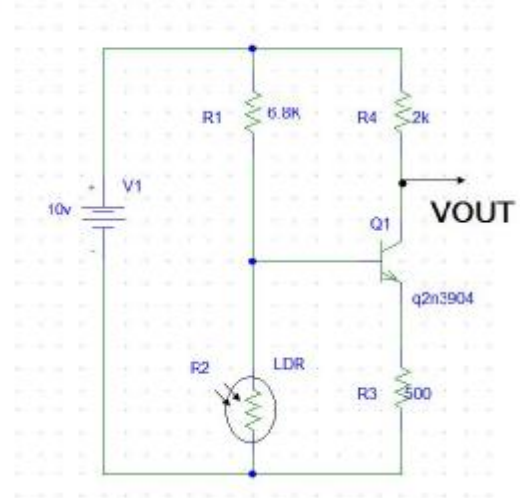
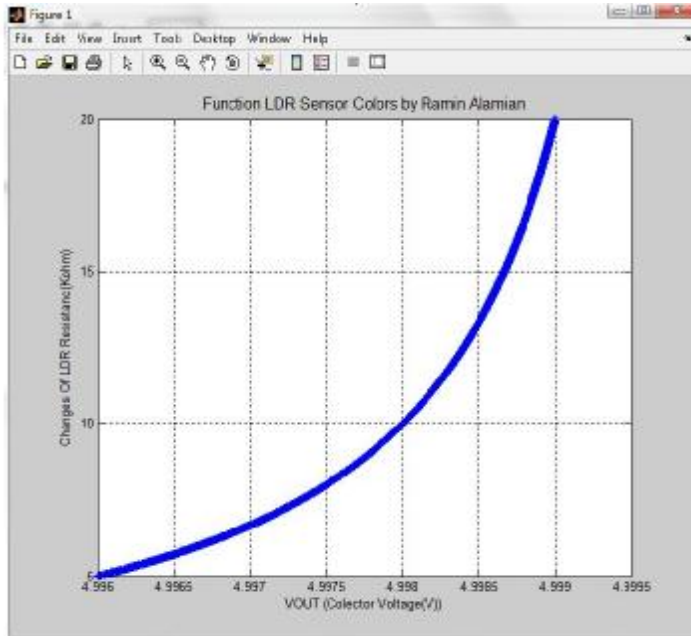
گیرنده و فرستنده نسبت به هم کاملاً عایق نوری هستند حال این فوتوترانزیستور را بایاس می کنیم تا بتوانیم از تغییرات جریان IC برای تبدیل نور به ولتاژ استفاده کنیم .





مقاومت R_1 را می توان بین انتخاب کرد $820\Omega - 80\Omega$ هر چقدر که مقاومت کمتر باشد به دلیل اینکه جریان عبوری از فرستنده بیشتر است فرستنده قوی تر عمل می کند و سطح ولتاژ در خروجی بالاتر می آید.

برای انتخاب مقاومت R_2 توسط نرم افزار matlab چند نمودار بدست آوردیم که بهترین مقاومت در این شرایط را برای آن انتخاب کنیم. با توجه به جریان عبوری از کلکتور ترانزیستور مقاومت $10k$ این جریان کم را به ولتاژ بزرگ تبدیل کرده و می توان فاصله را از 5 cm افزایش هم داد.



مقایسه کننده :

باید این المان، یک مقایسه کننده با سرعت بالا باشد تا ربات از خط منحرف نشود برای این منظور چند OP-Amp را در نظر گرفتیم:

۱- Lm ۳۳۱

۲- Lm ۳۵۸

۳- Lm ۳۳۹ و

سرعت چرخش زاویه در $Lm\ 358$ حدود $0.5\ us/V$ می باشد یعنی زمان برای به 5 ولت رسیدن $2.5\ us$ است اما در $Lm\ 339$ سرعت چرخش زاویه ای $13\ us/V$ است که زمان به 5 ولت رسیدن حدود $65\ us$ می باشد که $Lm\ 358$ بسیار سریعتر از نوع دوم خودش است.

$Lm\ 358$ یک Double op-Amp می باشد که دو کانال دارد $Lm\ 339$ یک Four op-Amp است که اگر مدار با آن طراحی شود برد کوچکتری به دست می آید.

ابتدا بهتر است راجع به نحوه تشخیص فاصله سنسور $Cny70$ بحث کنیم در ساختار کلی سنسور IR که از گیرنده و فرستنده تشکیل شده فرستنده با طول موج $950\ nm$ سیگنال IR را ارسال کرده و در خروجی فتوترانزیستور این سیگنال را دریافت می کند. رابطه فاصله با مقدار جریان عبوری از کلکتور فتوترانزیستور در شکل زیر مشخص می باشد.

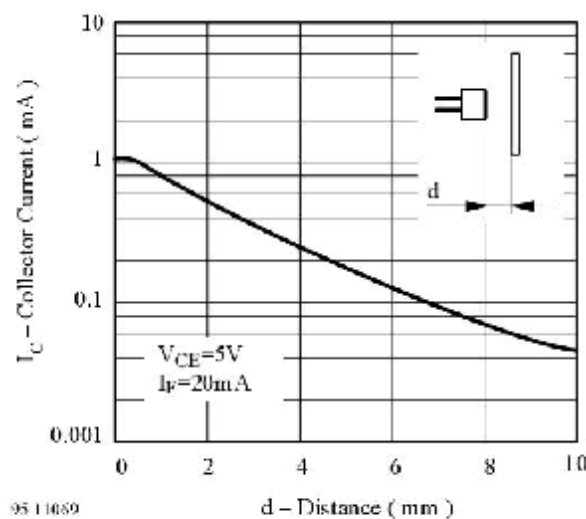


Figure 9. Collector Current vs. Distance

همانطور که از روی نمودار پیداست برای فاصله ۱۰ میلی متر سنسور $Cny70$ حدود ۰.۵ mA یعنی حدود $40\mu A$ در کلکتور خود جریان دارد به عبارتی برای بیشتر کردن این ولتاژ نیاز به مقاومت بزرگتری می باشد .

لذا همان طور که از نمودار matlab رنج مقاومت ها را مشاهده کردید مقاومت های با رنج کیلو اهم در این مدار بهتر عمل می کنند .

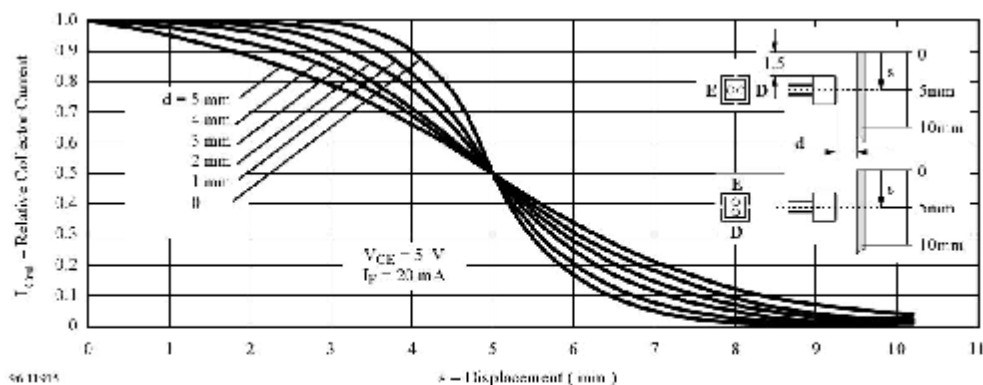
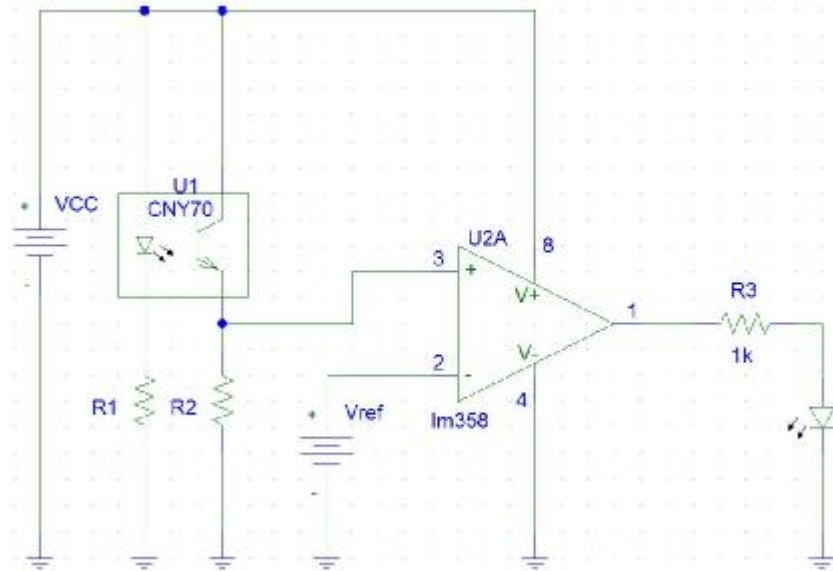


Figure 11. Relative Collector Current vs. Displacement

۱. اتصال سنسور به میکرو از طریق op- Amp A/D



در این مدار ابتدا فرستنده با یک جریان ثابت حدود 16mA اشعه IR را ارسال کرده ، طی فاصله ثابت یا متغیر این اشعه برگشتی به بیس (لنز) فتوترانزیستور گیرنده می رسد و در آمپتر یا کلکتور طی طراحی مدار ولتاژ مورد نظر را برای مدار تولید می کند .

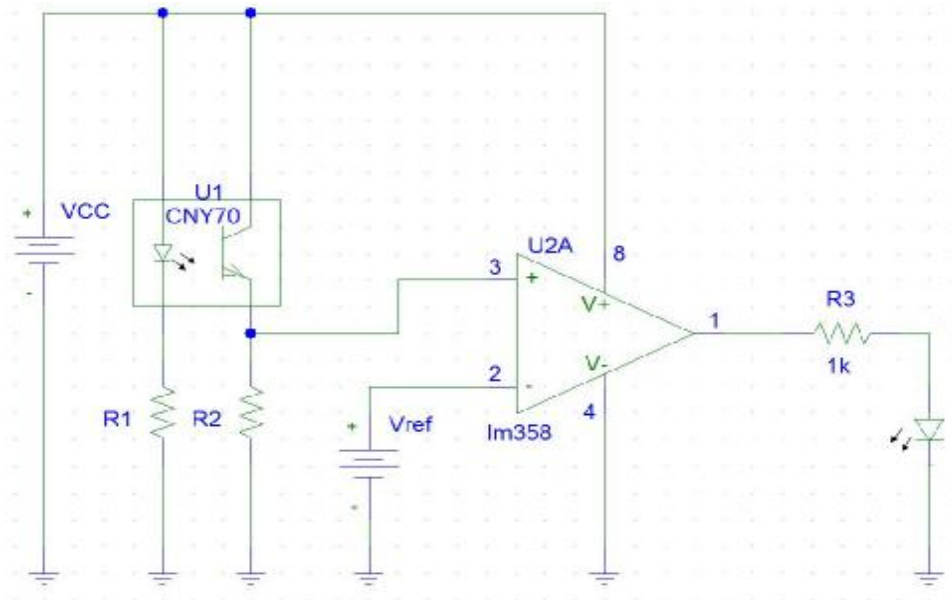
این ولتاژ می تواند با یک ولتاژ مرجع مقایسه شود و خروجی ۱ و ۰ منطقی دهد یعنی با حضور جسم "۱" و در صورت عدم وجود جسم در جلوی سنسور "۰" منطقی بوجود می آید .

۲. اتصال به ADC میکرو :

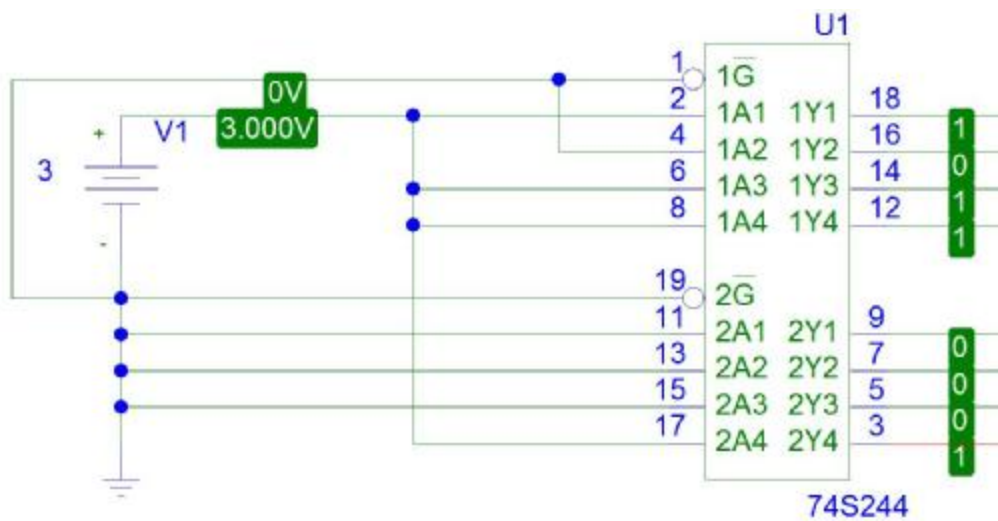
در بخش میکروکنترلر به توضیح کامل تمامی امکانات آن خواهیم پرداخت.

مدار انتخاب شده راه اندازی سنسور توسط تبدیل سیگنال آنالوگ به دیجیتال توسط مدارات

جانبی.



با این روش سطح TTL را به CMOS تبدیل می کنیم



شکل ۶

همانطور که در قسمت بالا ذکر شد سنسور IR اربات روی خط سفید دارای ولتاژ ۷V را می بینید و روی خط مشکی '۱' ولت در خروجی خود دارد. پس می توان با این مدار به سادگی صفر و یک منطقی را برای میکرو آماده کرد.

سنسور رنگ :

ابتدا برای نحوه تشخیص رنگ باید رنگ سیاه از سفید را کاملاً تفکیک کنیم که در قسمت های قبل کاملاً توضیح دادیم.

برای تشخیص رنگ حالت های مختلفی قابل طراحی است:

۱. استفاده از مقاومت نوری LDR با نور رنگی ثابت دو بیتی
۲. استفاده از مقاومت نوری LDR با نور رنگی جاروب شده
۳. استفاده از سنسور نوری TSL۲۳۰ با نور سفید ثابت و خروجی فرکانس
۴. استفاده از سنسور رنگ TCS ۲۳۰ با نور سفید ثابت و خروجی فرکانس
۵. استفاده از "سنسور سیاه و سفید" Cny۷۰ با ۳ بیت

در ادامه به شرح کامل هر ۵ روش می پردازیم.

روش اول :

۱. استفاده از مقاومت نوری LDR با نور ثابت قرمز و سبز "دو بیتی"

LDR یک مقاومت است که با تغییر نور به آن مقاومت آن تغییر خواهد کرد .



این مقاومت در تاریکی مطلق حدود ۹.۸ کیلو اهم مقاومت دارد و در نور عادی محیط حدود

۱.۵ کیلو اهم . این مقاومت رابطه کاملاً خطی با نور ندارد به همین دلیل ما با نورهای

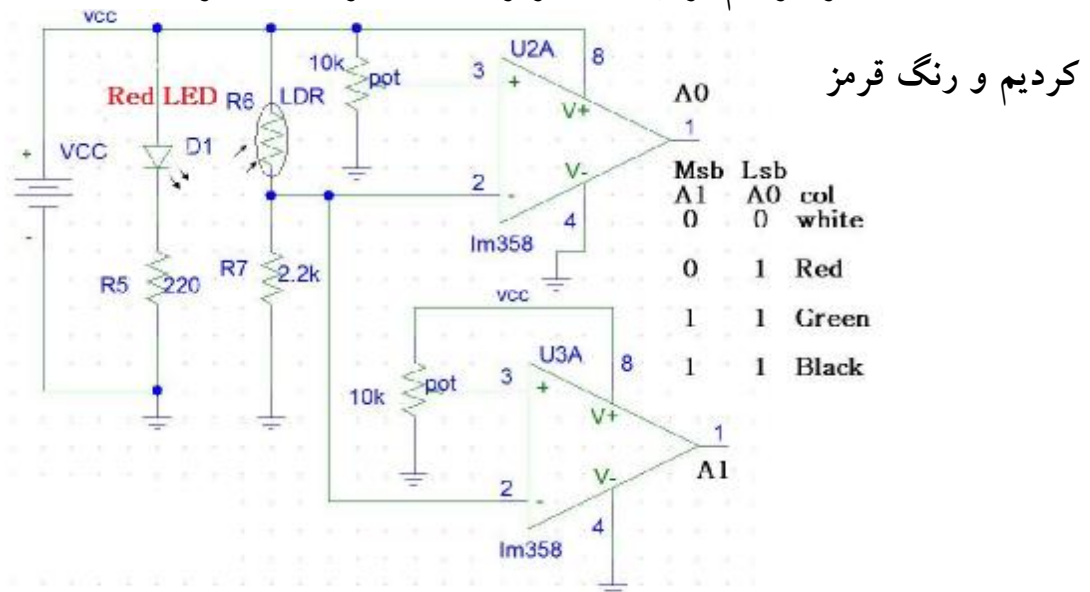
مختلف آن را آزمایش کردیم .

نور	مقاومت ($k\Omega$)
سفید	1.1Ω
سبز	$3.2k\Omega$
قرمز	$5.4k\Omega$

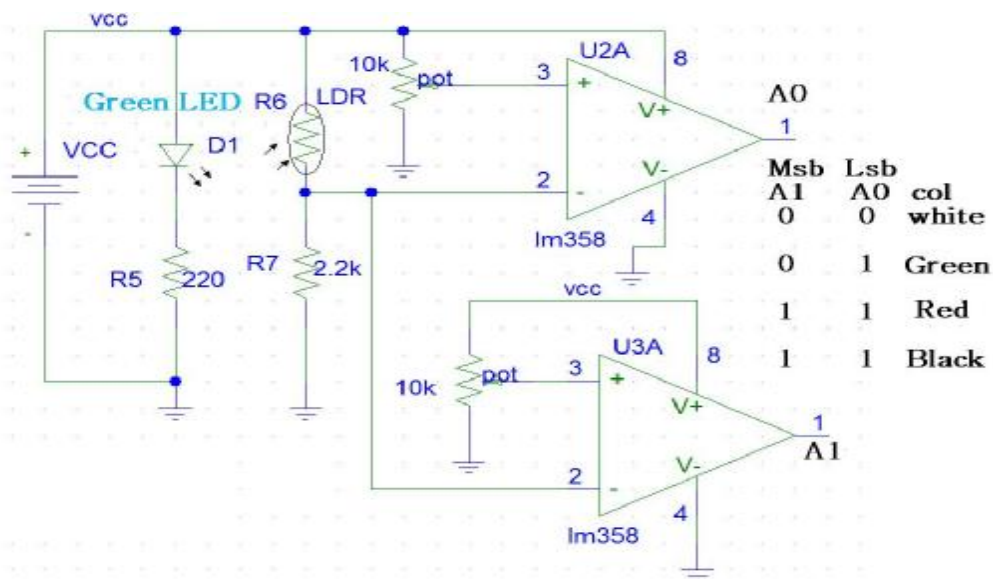
طبق جدول بالا به سادگی می توان نور را از هم تفکیک کرد .

مدار زیر می تواند دو رنگ سبز و قرمز را برای ما کاملاً تفکیک کند .

ما دو LDR در نظر گرفتیم برای رنگ قرمز و رنگ سبز. رنگ سبز را با LED سبز راه اندازی



با LED قرمز و بعد توسط op-Amp آنها را به سیگنال دیجیتال تبدیل کردیم.



پایه شماره ۳: OE : فعال ساز آی سی می باشد که باید حتماً با صفر منطقی فعال شود

پایه شماره ۶: OU: خروجی آی سی است که یک فرکانس می باشد .

پایه شماره ۵: VDD: ولتاژ ۵V به این پایه متصل می شود .

پایه های شماره ۲ و ۱ : S_0, S_1 : برای تنظیم حساسیت به نور

Sensitivity Frequency	S0	S1
Power Down	0	0
1X	0	1
10X	1	0
100X	1	1

پایه های شماره ۷ و ۸: S_0, S_3 :

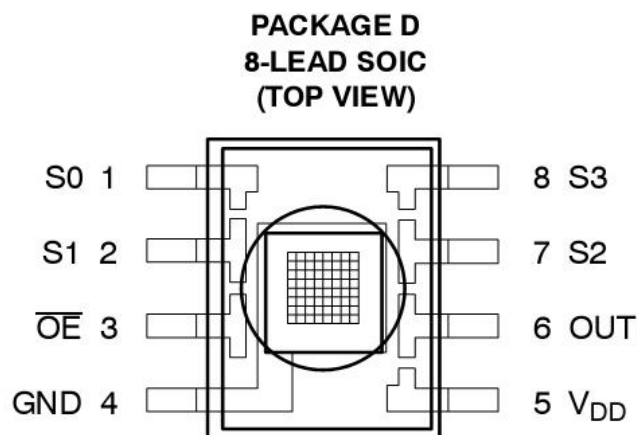
توسط این دو پایه می توان آی سی را به نور ۱ و ۱۰ و ۱۰۰ برابر حساس تر کرد .

I ^r out	S0	S1
1X	0	0
2X	0	1
10X	1	0
100X	1	1

توسط این پایه ها می توان فرکانس خروجی را ۱ و ۲ و ۱۰ و ۱۰۰ برابر کرد.

شرح کار مدار به این صورت است که یک LED در کنار این آی سی قرار می دهیم بر اثر برخورد به رنگ های قرمز و سبز فرکانس در حدود ۱۰۰k-۱k در خروجی ظاهر می شود که می توان این فرکانس را به تایمر میکرو متصل کرده و از آن استفاده نمود تایمر می تواند فرکانس های موجود را در یک بازه زمانی مثلاً ۱۰ ms بخواند و میزان فرکانس را تشخیص دهد .

۴- استفاده از سنسور TCS۲۳۰ خروجی فرکانس



پایه های شماره ۴ , ۵ : تغذیه به ترتیب زمین و VCC سنسور می باشند.

پایه شماره ۳: فعال ساز سنسور است که با منطق کار می کند.

پایه شماره ۶: خروجی از نوع فرکانس است.

پایه های شماره ۲ و ۱: S_0, S_1 : با این پایه ها می توان مقیاس فرکانس خروجی را از ۲٪ تا ۱۰۰٪ تنظیم کرد.

پایه های شماره ۸ و ۷: S_2, S_3 : با این پایه ها می توان رنگ مورد نظر را انتخاب کرد سبز آبی و قرمز

F out	S0	S1
Power Down	0	0
2%	0	1
10%	1	0
100%	1	1

Select Colors	S0	S1
Red	0	0
Blue	0	1
No filter	1	0
Green	1	1

نحوه راه اندازی این سنسور نیز مانند TCS۲۳۰ می باشد با این تفاوت که پایه های V_{CC} و GND را با یه توسط میکرو حتماً جابجاء کرد برای مثال اگر پایه های V_{CC} را عدد "۰۱" قرار داد و در خروجی فرکانس مشاهده کرد یعنی آی سی رنگ آبی را تشخیص داده است در این آی سی فرکانس خروجی ۵۰ KHZ می باشد .

سنسور رنگ TSL ۲۳۰

سنسور TSL ۲۳۰ یک سنسور تشخیص نور است، به این معنی که مقدار نور را با این سنسور می توان اندازه گیری نمود. از آنجایی که هر طیف نوری، یک دامنه و فرکانس خاص دارد، این سنسور نور نسبت به رنگ های مختلف، خروجی های متفاوتی دارد.

طرز کار این سنسور هنگامی که طیف نور به آن برخورد می کند، به این صورت است:

این سنسور دارای یک فتودیود و یک مدار داخلی است، که این مدار داخلی با یک سری از پایه ها قابلیت کنترل و برنامه ریزی دارد که نسبت به ورودی آنالوگ فتودیود، خروجی به صورت دیجیتال خارج می شود.

این سنسور دارای ۸ پایه است. یکی از پایه ها ولتاژ ۵V + است که ولتاژی بیش از این مقدار موجب آسیب رسیدن به سنسور می شود. یک پایه GND یا همان پایه منفی (مشخص کننده زمین) است.

یکی دیگر از پایه ها ، خروجی فرکانس است. فرکانس خروجی بر حسب مقدار نوری که به فتودیود برخورد می کند، دامنه خروجی فتودیود و پایه هایی که برای برنامه ریزی می باشند، تغییر می کند. یک پایه هم پایه ی Enable (پایه ی فعال ساز) است.

چهار پایه باقی مانده برای برنامه ریزی مدار داخلی این سنسور است که فرکانس خروجی را به 1X, 2X, 10X, 100X تقسیم می کند.

نحوه ی راه اندازی سنسور رنگ TSL ۲۳۰

پس از مشخص شدن نوع فرکانس خروجی و Low و high کردن پایه های مربوطه، فرکانس خروجی داریم.

به منظور مشخص نمودن نوع رنگ ، ابتدا باید سنسور را نزدیک یک طیف رنگ مربوطه قرار دهیم و مقدار خروجی را معین کنیم. مشخص نمودن ای فرکانس به چند پارامتر بستگی دارد:

(۱) طیف رنگ مربوطه

(۲) شدت طیف رنگ (شدت نور یا دامنه گسیل)

(۳) فاصله تا طیف رنگ

پس از مشخص شدن نوع فرکانس ، برای تعیین صحت طیف رنگ، دوباره مراحل مربوط را با پارامتر هایی که ذکر شد ، کنترل می کنیم.

برای نصب این سنسور روی روبات، سنسور باید به طور کامل کالیبره شده باشد. از این طریق می توانیم کلیه طیف های رنگی که فرکانس آنها از قبل معین شده اند را تشخیص دهیم.

نحوه ی ارتباط سنسور با میکروکنترلر

برای ارتباط این سنسور با میکروکنترلر از **Timer counter** استفاده می کنیم. به این صورت که **Timer counter** را در مد **counter** با فرکانس **1 MHz** انتخاب کرده و پایه **counter** را در حالت لبه ی بالا رونده قرار می دهیم.

لازم به ذکر است که فرکانس کاری سنسور را باید بین $20X$ تا $100X$ قرار دهیم. البته به تعداد خطوط نوشته شده در برنامه و سرعت فرکانس برنامه هم بستگی دارد.

فرکانس سنسور را در **Timer counter** با یک بازه زمانی **10ms** شمارش می کنیم.

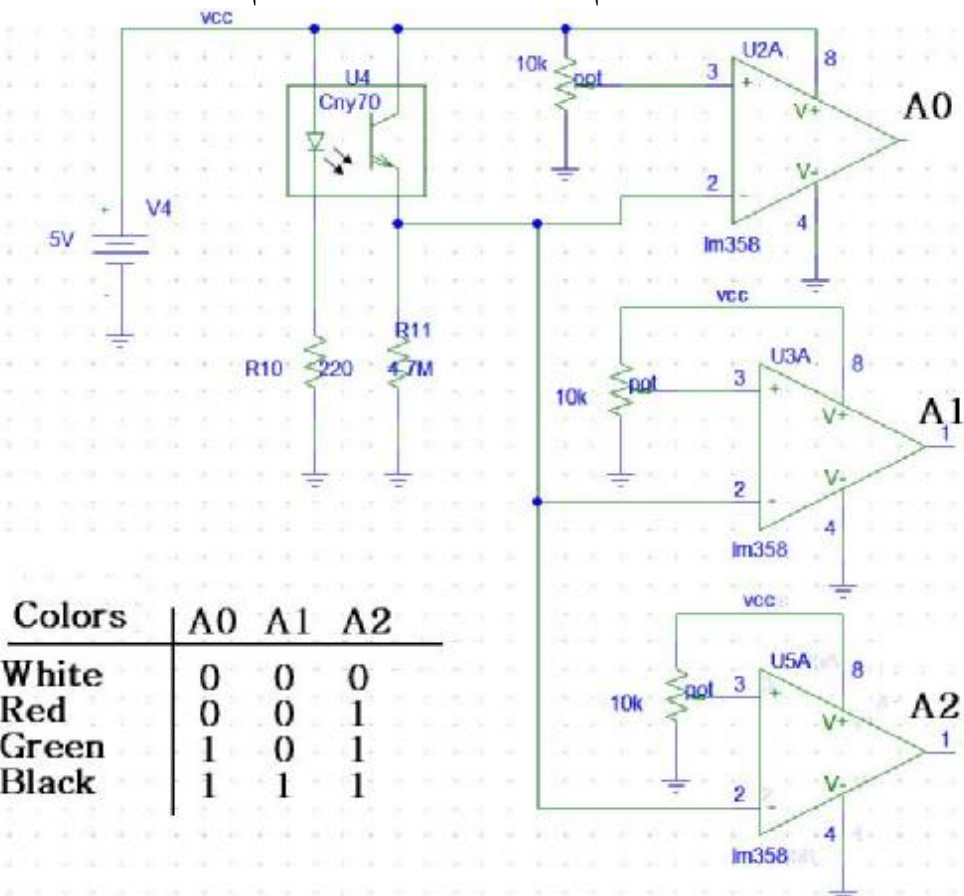
به عنوان مثال درطیف رنگ قرمز، با تعیین نوع فرکانس سنسور که روی $20X$ است و فرکانس **Timer counter** که روی **1MHz** است و فرکانس کل میکرو که روی **8MHz** است، عدد **186** را در یافت می کنیم که تعیین کننده ی طیف نور قرمز است.

حال برای اولین بار عدد به دست آمده را در یک متغیر جای می دهیم، که به منزله ی کالیبره شدن سنسور در طیف رنگی مورد نظر می باشد. در مراحل بعدی هر بار سنسور روی طیف رنگی قرارگرفت و عدد **186** را نشان داد، مشخص کننده ی طیف رنگ قرمز می باشد.

متأسفانه به دلیل اینکه روبات ها دارای تحرک می باشند ، در بعضی مواقع این عدد به طور ناگهانی ظاهر می شود که به عنوان نویز شناخته می شود. برای بر طرف کردن این اشکال، شما می توانید در بازه ی زمانی 30ms سه بار این عدد را کنترل کنید که اگر این عدد هر سه بار با مقدار عددی کالیبره شده توسط میکروکنترلر یکسان بود ، مطمئنا سنسور روبروی طیف نوری کالیبره شده قرار گرفته است.

۴. استفاده از سنسور Cny70 به عنوان سنسور رنگ

از یک سنسور Cny70 به عنوان سنسور رنگ استفاده کردیم که هر سنسور را به ۳ بیت ارتقا دادیم با این کار با یک سنسور می توانیم ۴ رنگ را تشخیص دهیم .



همانطور که از مدار پیداست این سنسور قابلیت تشخیص در ۴ رنگ را دارد البته این سیستم به حرکت نويز پذیر است و با فاکتور سرعت تناقص دارد .
میکروکنترولر:

اساسی ترین قسمت یک ربات قسمت CPU یا میکرو کنترولر سیستم می باشد. انتخاب کردن یک میکرو برای یک ربات بسیار مهم است در اکثر ربات هایی که در ایران ساخته می شود از میکروکنترولر سری Atmega استفاده می شود.

امکانات میکروکنترلرهای AVR:

بعضی از این امکانات در قسمت های قبل ذکر شدند که نیازی به توضیح ندارند. مانند RTC.

(۱) تایمر-کانتر: تایمر یعنی زمان سنج و کانتر یعنی شمارنده.

موارد استفاده از تایمرهای میکروکنترلر: ایجاد زمان های بسیار دقیق مثلاً می توانیم یک ثانیه را به طور بسیار دقیق ایجاد کنیم.

موارد استفاده از کانتر: کانتر در میکروکنترلر می تواند پالسهایی که به یک پایه ی مشخص میکرو کنترولر می رسد را شمارش کند و می تواند در کاربرد های مختلفی مانند ایجاد فرکانس متر خط تولید یک کارخانه و... استفاده گردد. مثلاً در ATmega۱۶ اگر به پین (T۱) (PB.۱) پالس اعمال کنیم در صورت برنامه ریزی کردن میکروکنترلر می توانیم این پالس ها را شمارش کنیم. و وقتی مثلاً ۱۰۰ پالس به میکرو اعمال شد، خروجی فعال شود.

۲) وقفه ی خارجی: در اکثر میکروکنترلرهای AVR سه وقفه ی خارجی وجود دارد. پایه های (Int₀، PD.۲ (Int₁، PD.۳ (Int₂ و PB.۲) در ATmega۱۶ نشان دهنده ی وقفه ی خارجی هستند. کلمه ی Int مخفف Interrupt به معنای وقفه می باشد. وقفه های خارجی می توانند نسبت به لبه ی بالا رونده (Rising) و یا لبه ی پایین رونده (Falling) حساس باشند. در لبه ی بالا رونده وقتی ولتاژ اعمال شده به پایه ی IntX از سطح پایین (صفر ولت) به سطح بالا (پنج ولت) تغییر می کند، وقفه ی خارجی رخ می دهد و وقتی این وقفه رخ داد، میکروکنترلر دستورات ما را اجرا می کند (این دستورات در زیر برنامه ی وقفه قرار می گیرند) در لبه ی پایین رونده وقتی ولتاژ از سطح بالا به سطح پایین تغییر کند، وقفه رخ می دهد.

۳) خود برنامه ریزی Boot Loader: می توانیم به گونه ای میکروکنترلر را برنامه ریزی کنیم که اگر فلان اتفاق رخ داد، کل حافظه یا بخشی از حافظه ی Flash میکرو پاک شود و برنامه ی جدیدی جایگزین گردد. مثلاً برای به روز رسانی برنامه ی میکروکنترلر می توانیم میکرو را از مدار جدا کنیم و آن را به کامپیوتر متصل کنیم که در بعضی از مواقع این کار غیر عملی می باشد (مثلاً در خط تولید یک کارخانه) بنابراین استفاده از خود برنامه ریزی بهترین گزینه به حساب می آید.

۴) حافظه ی EEPROM: در میکروکنترلر گاهی مقدار یک متغیر حیاتی می باشد، به این معنی که اگر تغذیه ی (برق) میکرو قطع شود، مقدار این متغیر که در درون حافظه ی RAM قرار دارد، پاک می شود. مثلاً در یک کارخانه تعداد کالاهای تولید شده شمارش می شود. ولی ناگهان برق قطع می شود و هنگام وصل برق شمارش از نو آغاز می شود برای جلوگیری از این حالت مقدار متغیر را در درون حافظه ی EEPROM قرار می دهیم تا در صورت قطع برق اطلاعات ما از بین نرود. یا در یک قفل رمزی الکترونیکی لازم است تا رمز قفل پاک نشود که باید این رمز را در

حافظه ی EEPROM قرار دهیم و هنگام نیاز آن را از حافظه ی EEPROM بخوانیم. دوام اطلاعات در حافظه ی EEPROM بیش از ۴۰ سال می باشد.

۵) مدهای مصرف کم توان: در مدهای مصرف کم توان CPU میکرو دستوری را اجرا نمی کند و به همین دلیل مصرف توان میکرو تا حد زیادی پایین می آید و فقط وقفه ها می توانند میکرو را از این حالت بیدار کنند. (یعنی CPU فعال شود)

بیشترین مدهای کم توان شش حالت می باشند.

IDLE: در این حالت کلاک پالس (Clock Pulse CPU) متوقف می شود و میکرو به SPI، USART، Analog Comparator، ADC، Timer/Counters، Watchdog، I²C و وقفه ها اجازه می دهد که کار کنند و اگر این وقفه ها فعال شده باشند می توانند میکرو را از این حالت بیدار کنند.

ADC NOISE REDUCTION: در این حالت کلاک CPU متوقف می شود ولی اگر ADC، I²C، Watchdog، Timer/Counter و وقفه ها فعال باشند، میکرو را از این حالت خارج می کنند. در این حالت اصولاً کلاک حافظه ی Flash، I/O و CPU متوقف می شوند ولی کلاک های دیگر کار می کنند. این مد در هنگام کار کردن مبدل آنالوگ به دیجیتال می باشد که نویز احتمالی را کم می کند.

Power Down: در این مد اسیلاتور خارجی متوقف می شود ولی وقفه های خارجی، Watchdog و I²C اگر فعال باشند، به کارشان ادامه می دهند. مدت زمان خارج شدن میکرو از این حالت توسط فیوز بیت های SUT و CKSEL قابل تنظیم می باشد.

Power Save: این مد با **Power Down** یکسان است با این تفاوت که تایمر ۲ می تواند میکرو را از این حالت بیدار کند. در این مد مصرف جریان به حدود $1\mu A$ می رسد. از این مد وقتی استفاده می کنیم که تایمر ۲ میکروکنترلر را در حالت آسنکرون پیکر بندی کرده باشیم.

وقتی تایمر ۲ در حالت آسنکرون باشد، یک کریستال بین پایه های $TOSC_1$ (PC.۶) و $TOSC_2$ (PC.۷) قرار می گیرد که مقدار آن $32768 Hz$ می باشد. از این حالت برای تولید یک ثانیه استفاده می شود.

Standby: وقتی می توانیم از این مد استفاده کنیم که کریستال خارجی میکرو فعال باشد. مد **Standby** با **Power Down** یکسان است با این تفاوت که در این حالت کریستال خارجی قطع نمی شود و میکرو پس از ۶ کلاک از این حالت خارج می شود.

Extended-Standby: وقتی می توانیم از این مد استفاده کنیم که کریستال خارجی میکرو فعال باشد. این مد با مد **Power Save** یکسان است با این تفاوت که در این حالت کریستال خارجی قطع نمی شود و میکرو پس از ۶ کلاک از این حالت خارج می شود.

SPI (۶): در این حالت میکرو کنترلر از طریق ۴ سیم می تواند با یک یا چند میکرو کنترلر دیگر در ارتباط باشد. مثلاً در میکرو کنترلر **ATmega16** چهار پایه ی $PB.7$ (SCK) برای دریافت کلاک، $PB.6$ (MISO) برای ورودی میکروی **Master** و خروجی میکروی **Slave**، $PB.5$ (MOSI) برای خروجی میکروی **Master** و ورودی میکروی **Slave** و $PB.4$ (SS) برای انتخاب میکروی **Master** مورد استفاده قرار می گیرد.

در میان اکثر میکروها برای پروگرم کردن باید از پایه های استفاده کرد:

Miso(۱)

Mosi(۲)

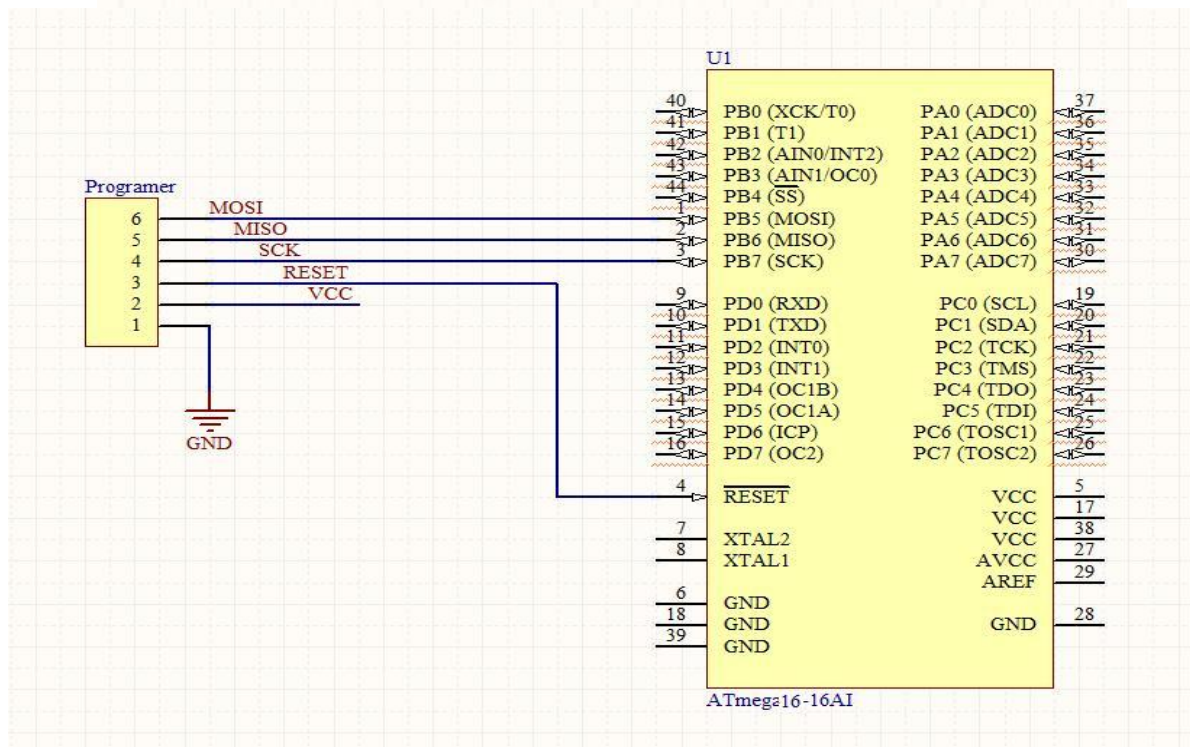
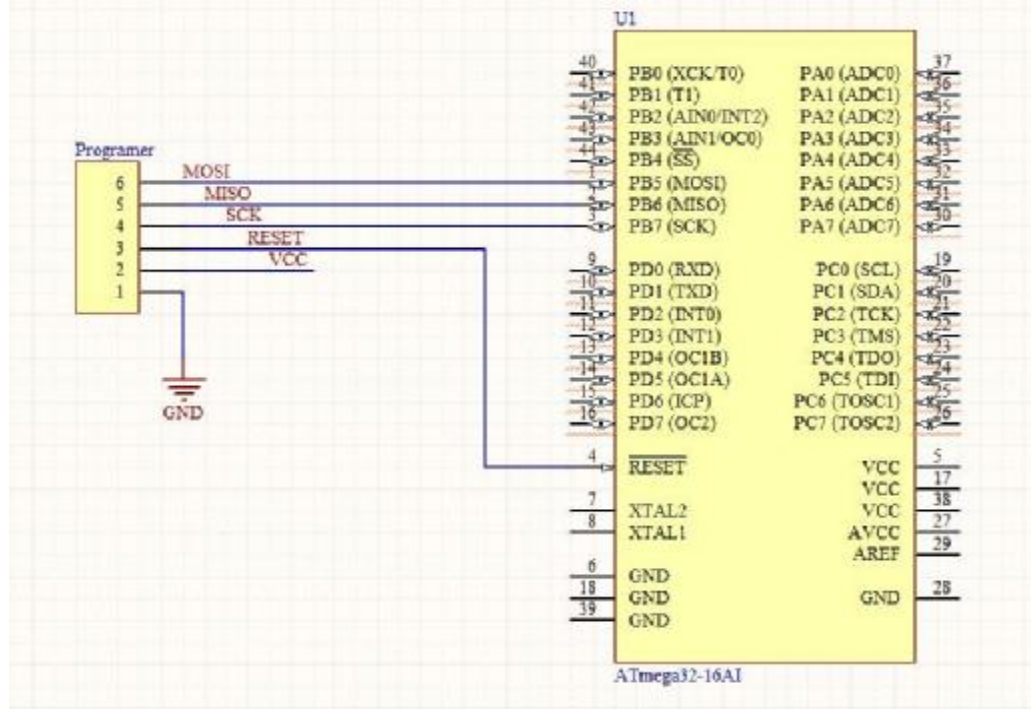
Sck(۳)

Reset(۴)

VCC (۴)

Gnd(۵)

این روش پروگرام کردن برای ۸-۱۶ - Atmega۳۲ بکار می رود.



برای پراگرم کردن **Atmega128,Atmega64** باید از پایه های

زیر استفاده کرد.

PDI

,DDO

,SCK

,RESET

,VVC,

.GND

برای پراگرم کردن این میکرو می توان با همان پراگرم **STK200/3001** هم پراگرم کرد, تنها کافی است که باید ها را به صورت زیر به پراگرم متصل کنید.

PDO=MISO

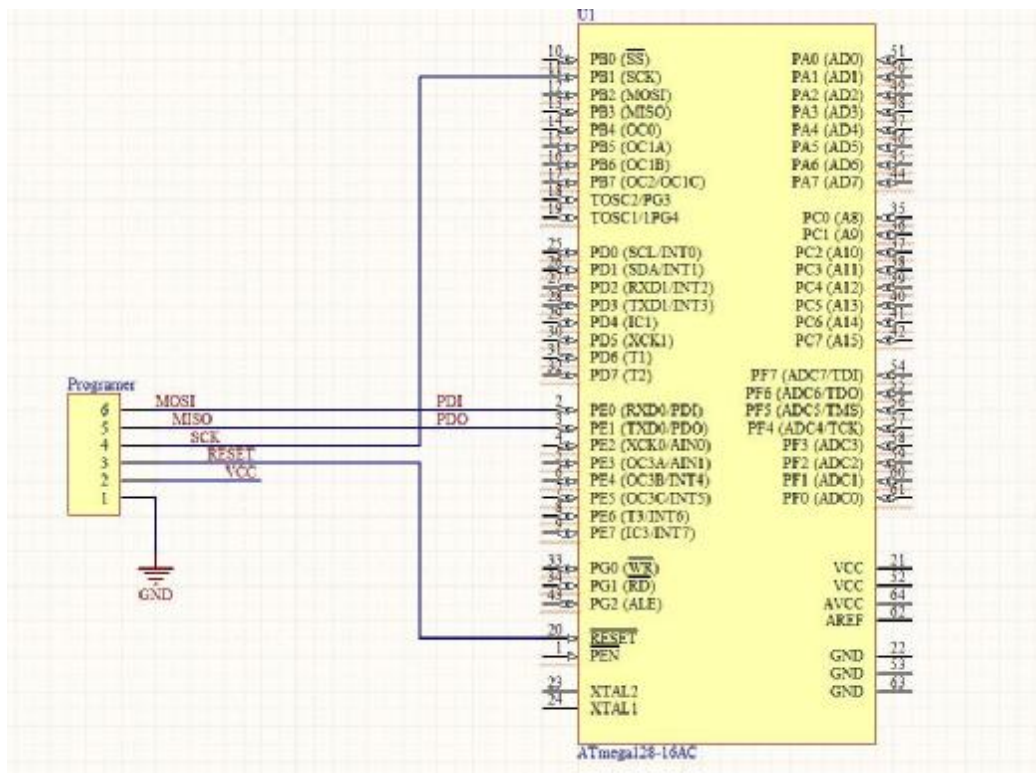
PDI=MOSI

SCK=SCK

RESET=RESET

Gnd=Gnd

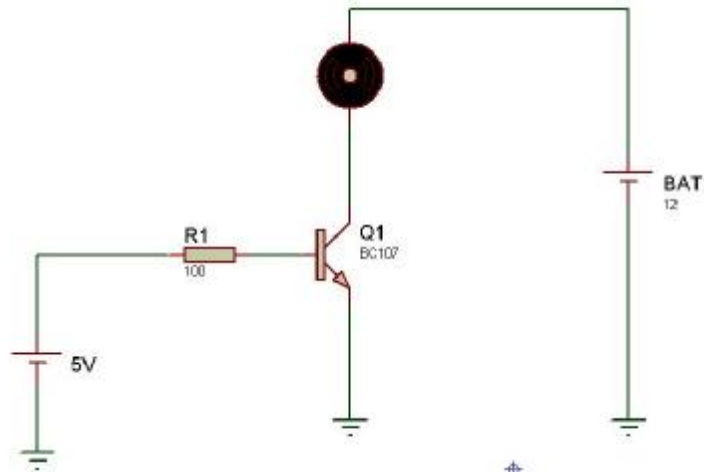
VCC=VCC



درایور موتور:

برای راه اندازی موتور همانطور که قبلا گفته بودیم باید از درایور موتور استفاده کرد. درایور موتور یک **TC** یا چند ترانزیستور یا حتی رله می تواند باشد.

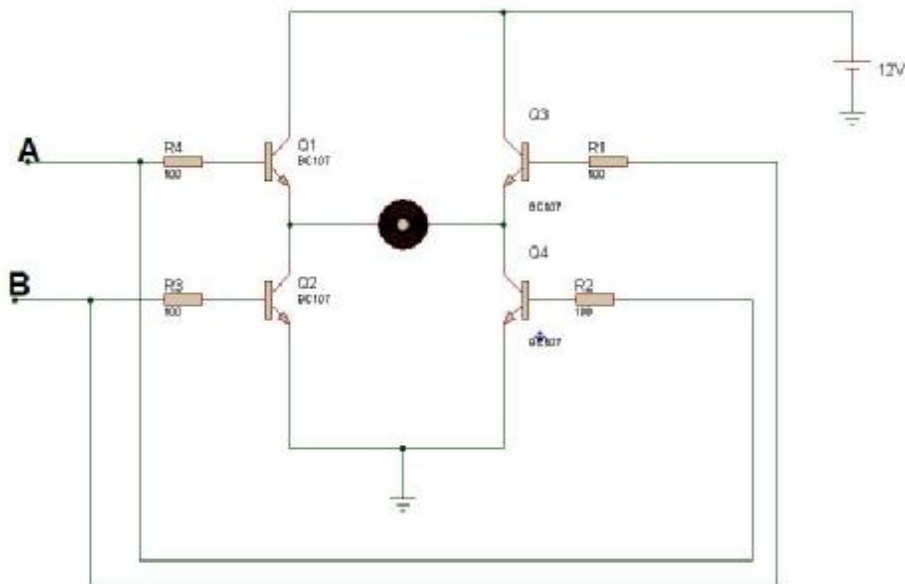
برای راه اندازی یک موتور می توان از مدار زیر استفاده کرد. باید با به قطع و اشباع بردن ترانزیستور این عمل را انجام داد.



$$I_B = (V_{CC} - V_{BE}) / 100 \Omega = 43 \text{ mA}$$

با این جریان ترانزیستور به اشباع می رود و موتور روشن می شود.

شکل زیر مدار یک موتور چپ گرد و راست گرد را نشان می دهد.



چند نمونه آی سی هم در بازار وجود دارد که در جلسه قبل نام برده شد. در این جلسه راه اندازی و طرز استفاده از دو IC دیگر را بیان می کنیم.

L۲۹۸:

راه انداز موتور:

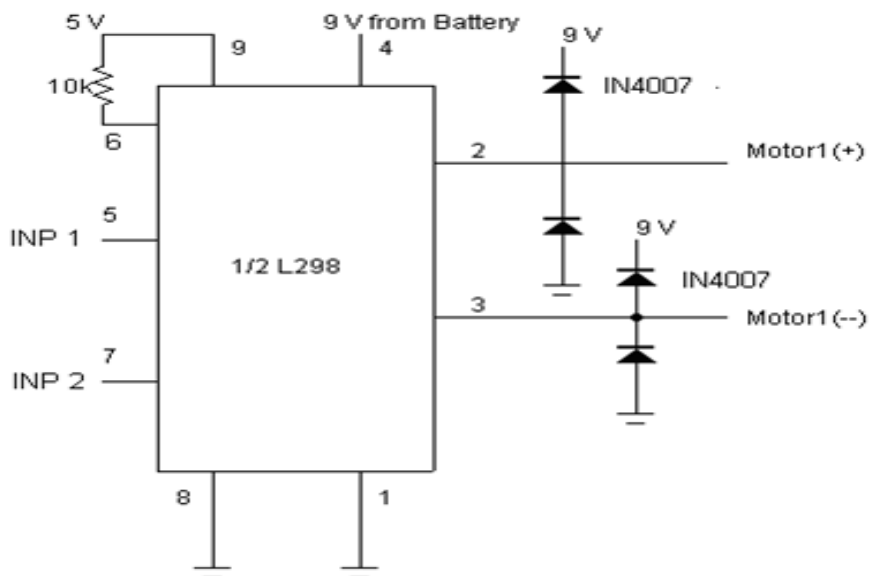
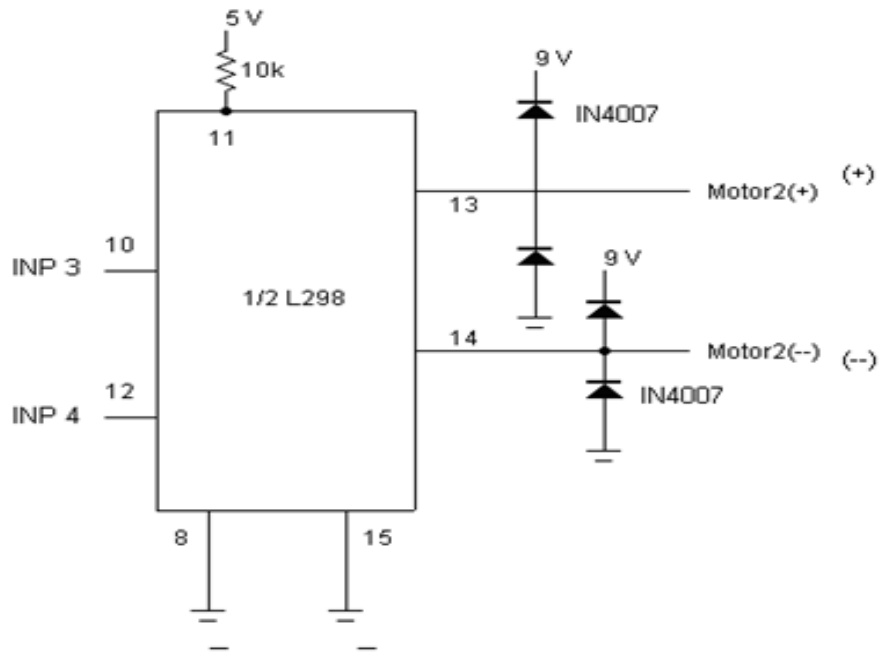
L۲۹۸ Motor Driver Circuitry



L۲۹۸

درایور L۲۹۸ یکی از قطعات مناسب جهت راه اندازی موتور است که با توجه به جریان دهی مناسب (تا یک آمپر در هر کانال) می تواند نیاز بسیاری از پروژه ها را مرتفع سازد. این قطعه با مدار ارائه شده می تواند دو موتور را به صورت مجزا راه اندازی کرده و جهت گردش آنها را کنترل نماید. که این کنترل توسط اعمال ولتاژ به چهار ورودی منطقی این قطعه صورت می گردد. (برای هر موتور دو ورودی) که می توان خروجی میکروکنترلر یا مدارات حسگر را به صورت مستقیم به این چهار ورودی متصل نمود و به راحتی موتور را کنترل کرد. در صورتی که از این قطعه برای راه اندازی موتورهای روبات خود بهره می گیرید دقت کنید که حتماً بر روی

آن حرارت گیر مناسب وصل نمایید.

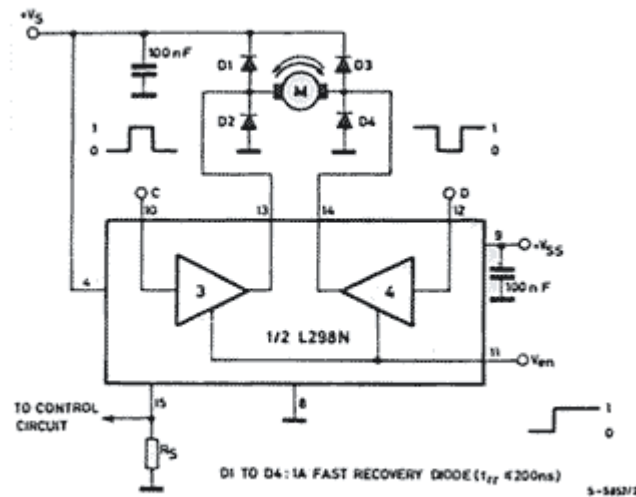


این مدار تنها یکی از راه های اتصال درایور L298 به موتور را نشان می دهد. در این مدار پایه های حسگر جریان (current sensing pins) به زمین متصل شده اند که با روشهایی می توان توسط این پایه ها جریان مصرفی موتور را کنترل نمود همچنین کنترل سرعت را می توان به روش مدولاسیون پهنای باند (Pulse Width Modulation) PWM و با اعمال فرکانس به پایه های ۶ و ۱۱ انجام داد که با اعمال +۵ ولت موتور روشن و با اعمال ۰ موتور خاموش می گردد. در این مدار پایه های مذکور به +۵ ولت متصل شده اند و موتور با حداکثر سرعت گردش خواهد نمود.

Inputs		Function
V _{en} = H	C = H ; D = L	Forward
	C = L ; D = H	Reverse
	C = D	Fast Motor Stop
V _{en} = L	C = X ; D = X	Free Running Motor Stop

L = Low H = High X = Don't care

جدول عکس العمل موتور نسبت به ورودی



تشریح پایه های درایور موتور L298 Pin Description

Pin ۱. CURRENT SENSING

از این پایه جهت کنترل جریان موتور **A** استفاده می گردد. همچنین می توان این پایه را به صورت مستقیم به خط منفی مدار **GND** اتصال داد که در این صورت کنترلی بر روی جریان وجود ندارد.

Pin ۲. OUTPUT ۱

این پایه به یکی از ترمینال های موتور **A** متصل می گردد. همچنین دیودها نیز جهت حفاظت به همین پایه متصل می شوند. (به نقشه مدار توجه کنید)

۲. OUTPUT ۳. Pin

این پین به ترمینال دیگر موتور **A** متصل شده و دیودها نیز مانند نقشه به آن متصل می گردند.

۴. SUPPLY VOLTAGE Pin (VS)

به پایه باید ولتاژ مورد نظر خود جهت اعمال به موتورها را متصل نمایید. این ولتاژ با توجه به موتورهای مورد استفاده شما حداکثر تا ۴۶ ولت می تواند افزایش یابد. برای ساخت رباتهای کوچک به طور معمول بین ۶ تا ۱۲ ولت نیاز است.

۱. INPUT ۵. Pin to driveMotor A TTL Compatible Inputs

این پایه باید به صفر یا پنج ولت متصل گردد که همراه با پین ۷ می توانند جهت گردش موتور را مشخص نمایند.

۱. ENABLE A TTL Compatible Enable Input forMotor A Pin ۶.

این پایه جهت روشن و خاموش کردن موتور **A** و در بیشتر مواقع جهت اعمال فرکانس **PWM** به موتور استفاده می گردد. پنج ولت موتور را روشن و صفر موتور را خاموش می کند.

۲. INPUT ۷. Pin to driveMotor A TTL Compatible Inputs

این پایه باید به صفر یا پنج ولت متصل گردد که همراه با پین ۵ می تواند جهت گردش موتور را مشخص نمایند.

Pin ۸. GND

اتصال به خط منفی مدار **GND**

(VSS)Pin ۹. LOGIC SUPPLY VOLTAGE

اتصال به ۵ تا ۷ ولت

Pin ۱۰. INPUT ۳ TTL Compatible Inputs ۱ to drive Motor B

این پایه باید به صفر یا پنج ولت متصل گردد که همراه با پین ۱۲ می تواند جهت گردش موتور **B** را مشخص نمایند.

Pin ۱۱. ENABLE B TTL Compatible Enable Input for Motor B

این پایه جهت روشن و خاموش کردن موتور **B** و در بیشتر مواقع جهت اعمال فرکانس **PWM** به موتور استفاده می گردد. پنج ولت موتور را روشن و صفر موتور را خاموش می کند.

Pin ۱۲. INPUT ۴ TTL Compatible Inputs ۲ to drive Motor B

این پایه باید به صفر یا پنج ولت متصل گردد که همراه با پین ۱۰ می تواند جهت گردش موتور B را مشخص نمایند.

۳. OUTPUT ۱۳. Pin

این پایه به یکی از ترمینال های موتور B متصل می گردد. همچنین دیودها نیز جهت حفاظت به همین پایه متصل می شوند.

۴. OUTPUT ۱۴. Pin

این ترمینال دیگر موتور B متصل می گردد. همچنین دیودها نیز جهت حفاظت به همین پایه متصل می شوند.

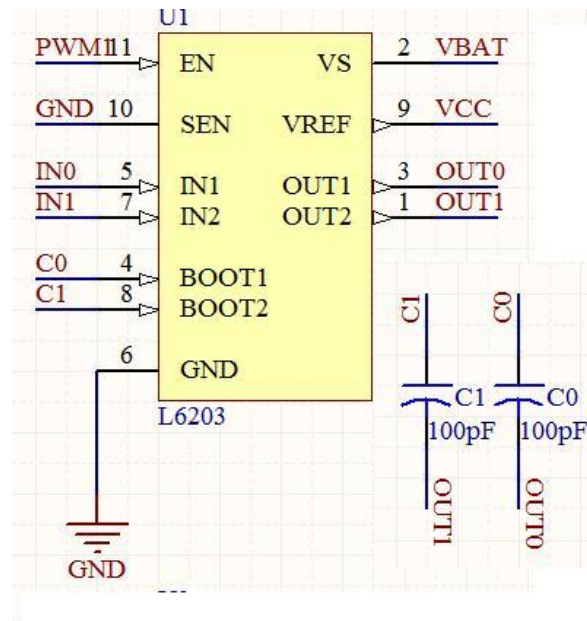
B CURRENT SENSING ۱۵. Pin

از این پایه جهت کنترل جریان موتور B استفاده می گردد. همچنین می توان این پایه را به صورت مستقیم به خط منفی مدار GND اتصال داد که در این صورت کنترلی بر روی جریان وجود ندارد.

۶۲۰۳.L:

این IC قابلیت کنترل جریان بالاتری را دارد اما هر IC توانایی حرکت یک موتور را به صورت

چیگرد و راستگرد دارد.



توضیح اینکه به پایه های ۱،۲ BOOT باید دو خازن متصل شود که این خازنهای ۱۰۰PF هستند.