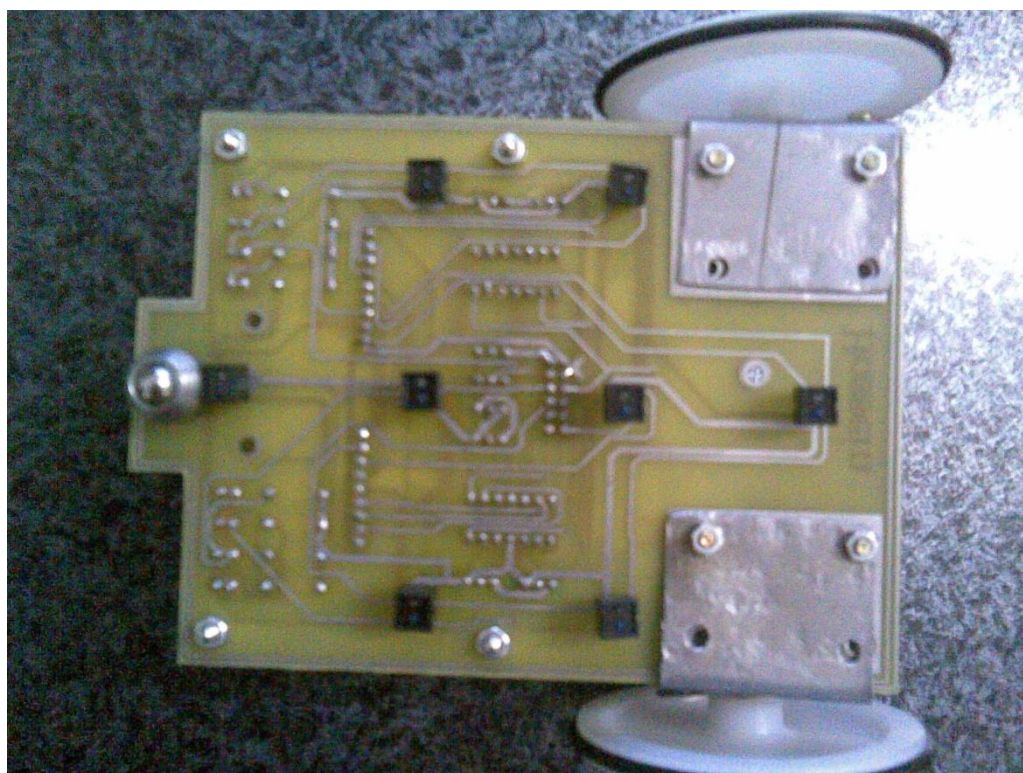
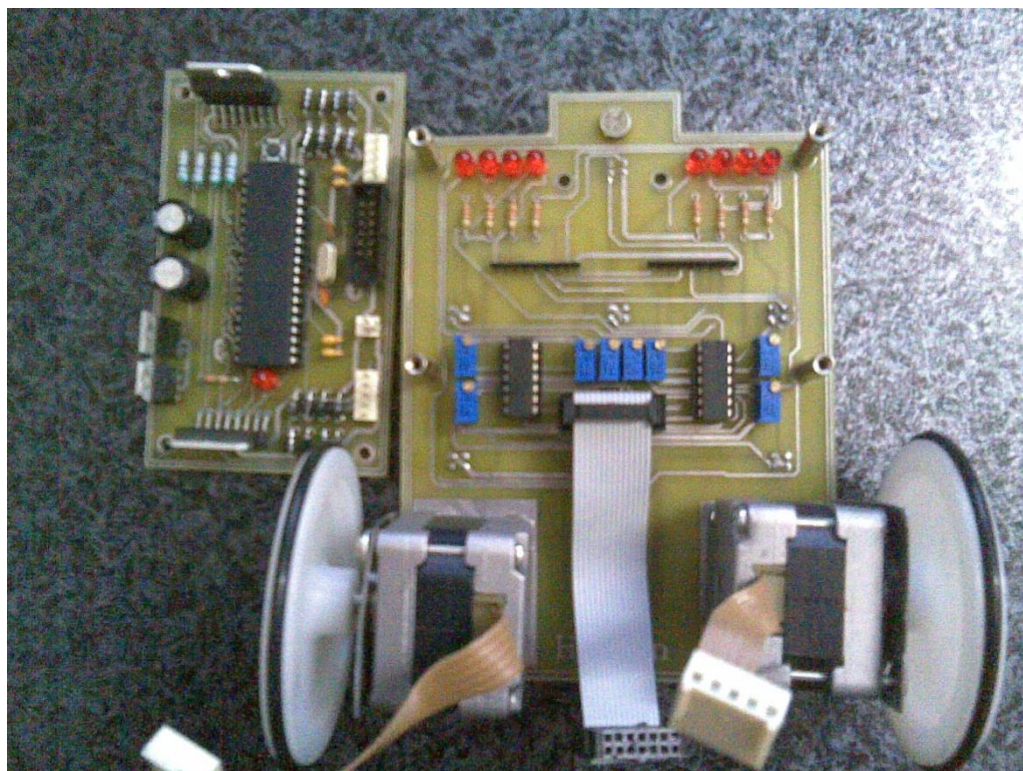


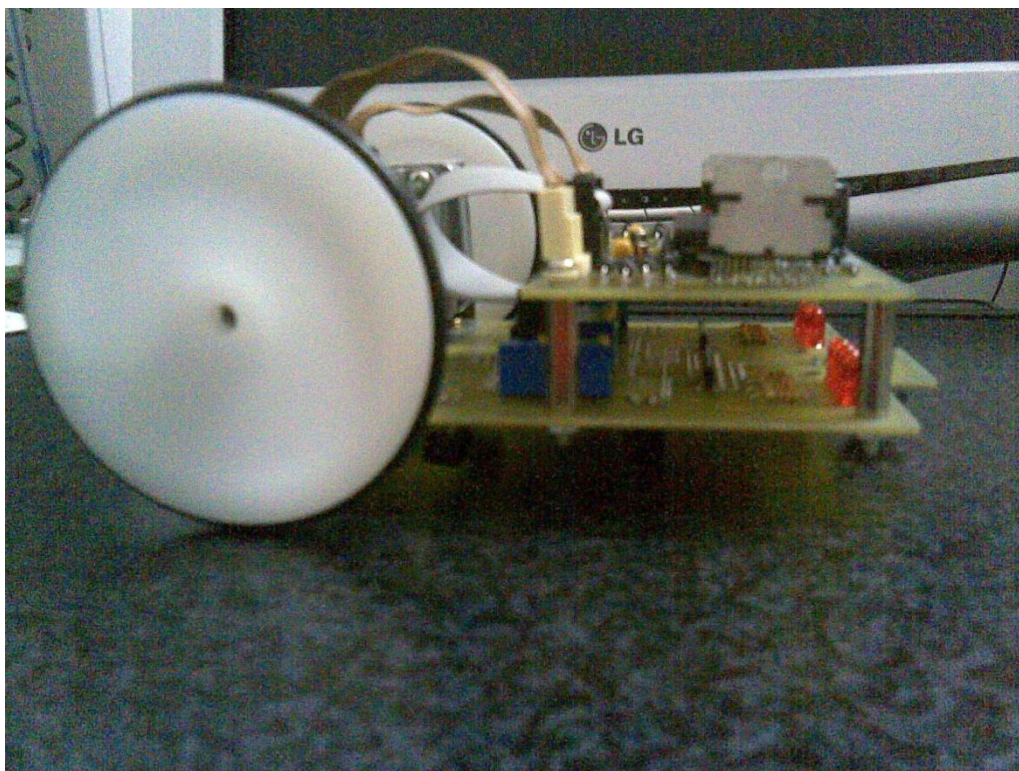
بنام خدا

آموزش ساخت

پروژه روبات مسیریاب







فهرست

پیشگفتار

7.....	آشنایی با رباتیک
7.....	اجزای یک ربات
7.....	تاریخچه رباتیک
10.....	انواع رباتها
13.....	مزایای رباتها
13.....	معایب رباتها
13.....	تعریف ربات مسیریاب
15.....	بلوک دیاگرام یک ربات مسیریاب

16.....	سخت افزار
16.....	سنسور
16.....	سنسورهای انفعالی (passive) و فعال (active)
16.....	درجه بندی و اندازه داری (calibration)
17.....	سنسور در روبات تعقیب خط
17.....	سنسورهای نوری
17.....	فتوسل (مقاومت نوری)
21.....	فتو ترانزیستورها
22.....	سنسور مادون قرمز
23.....	فرستنده گیرنده مادون قرمز (IR)
24.....	سنسور GP2S09
25.....	سنسور CNY70
26.....	پایه های CNY70
28.....	مدار درابو سنسورها و تشریح عملکرد آن
32.....	آی سی 741
33.....	آی سی LM324
34.....	نحوه چیدمان سنسورها
35.....	بررسی انواع موتورهای الکتریکی بکار رفته در روباتهای تعقیب خط
35.....	نحوه حرکت موتورهای الکتریکی
37.....	ویژگی های موتور DC
37.....	انتخاب موتور
37.....	اصول کار موتورهای DC
38.....	سرعت موتور
38.....	گشتاور موتور
39.....	جریان توقف
40.....	توان و بازده
41.....	اندازه گیری مقاومت داخلی موتور
42.....	کنترل جهت موتور
45.....	کنترل با رله
45.....	کنترل با ترانزیستور
47.....	مدارهای H-bridge حالت جامد
49.....	موتور پله ای چیست؟
49.....	ساختار موتور پله ای
49.....	نحوه پیدا کردن سیم های موتور پله ای
49.....	مشخصه های اساسی موتور پله ای

51.....	اصول کار موتور پله ای
51.....	موتور پله کامل و نیم پله
51.....	راه اندازی موتور پله ای
52.....	مزایای موتور پله ای
52.....	معایب موتور پله ای
53.....	نحوه کنترل
53.....	نحوه کنترل 1بیتی
54.....	نحوه کنترل 2بیتی
54.....	نحوه کنترل نیم پله
54.....	ویژگی های کنترل نیم پله یا half step
55.....	Micro Step
55.....	ملاحظات طراحی design consideration
55.....	اندوکتانس Inductance
55.....	ارتباطات سری و موازی
56.....	ولتاژ درایور
56.....	سختی موتور
56.....	گرمای موتور
57.....	سیستم درایو
57.....	بررسی آی سی های درایور موتور
59.....	درایور موتور SN754410
60.....	ULN2003
63.....	درایور موتور با L298
65.....	تشریح پایه های درایور موتور L298
69.....	درایور مورد استفاده
70.....	میکروکنترلر مورد استفاده
71.....	تغذیه مورد استفاده
72.....	بدنه روبات
73.....	برنامه روبات
88.....	نقشه مدارات
91.....	عکس هایی از روبات

پیشگفتار :

در این مقاله به بررسی و نحوه عملکرد یک روبات مسیریاب که به عنوان پروژه پایان تحصیلات دوره کاردانی مطرح شده است، می پردازیم .
در اینجا لازم است از جناب آقای مهندس دمیا " استاد راهنمای محترم پروژه " و جناب آقای مهندس انصاری " مدیر محترم پروژه " و سایر افرادی که در ارائه این پروژه ما را راهنمایی و یاری نموده اند تشکر نماییم .

حسین صادقی
تقی اسدی

آشنایی با روباتیک



روبات یک ماشین الکترومکانیکی هوشمند است که حداقل یکی از رفتارهای انسان را تقلید نماید ، با خصوصیات مقابل : می توان آن را مکرراً برنامه ریزی کرد. چند کاره است. کارآمد و مناسب برای محیط است و ...

اجزای یک روبات:

1. وسایل مکانیکی و الکتریکی:
شاسی، موتورها، منبع تغذیه، ...
2. حسگرها (برای شناسایی محیط):
دوربین ها، سنسورهای sonar، سنسورهای ultrasound، ...
3. عملکردها (برای انجام اعمال لازم)
بازوی روبات، چرخها، پاها، ...
4. قسمت تصمیم گیری (برنامه ای برای تعیین اعمال لازم):
حرکت در یک جهت خاص، دوری از موانع، برداشتن اجسام، ...
5. قسمت کنترل (برای راه اندازی و بررسی حرکات روبات):
نیروها و گشتاورهای موتورها برای سرعت مورد نظر، جهت مورد نظر، کنترل مسیر، ...

تاریخچه روباتیک:

- 270 ق م : زمانی که یونانیان به ساخت مجسمه های متحرك میپرداختند.
- حدود سال 1250 م: بیشاپ آلبرتوس ماگنوس (Albertus Magnus Bishop) ضیافتی ترتیب داد که در آن، میزبانان آهني از مهمانان پذیرایی می کردند. با دیدن این روبات، سنت توماس آکویناس (Thomas Aquinas) برآشفته شد، میزبان آهني را تکه تکه کرد و بیشاپ را ساحر و جادوگر خواند.

- سال 1640 م: دکارت ماشین خودکاري به صورت يك خانم ساخت و آن را Ma fille Francine " مي ناميد. اين ماشين كه دکارت را در يك سفر دريايي همراهي مي كرد، توسط كاپيتان كشتي به آب پرتاب شد چرا كه وي تصور مي كرد اين موجود ساخته شيطان است.

- سال 1738 م: ژاك دواكانسن (Jacques de Vaucanson) يك اردك مكانيكي ساخت كه از بيش از 4000 قطعه تشكيل شده بود. اين اردك مي توانست از خود صدا توليد كند، شنا كند، آب بنوشد، دانه بخورد و آن را هضم و سپس دفع كند. امروزه در مورد محل نگهداري اين اردك اطلاعي در دست نيست.

- سال 1805 م: عروسكي توسط ميلاردت (Maillardet) ساخته شد كه مي توانست به زبان انگليسي و فرانسوي بنويسد و مناظري را نقاشي كند.

- سال 1923 م: كارل چاپك (Karel Capek) براي اولين بار از كلمه روبات (robot) در نمايشنامه خود به عنوان آدم مصنوعي استفاده كرد. كلمه روبات از كلمه چك robota گرفته شده است كه به معني برده و كارگر مزدور است. موضوع نمايشنامه چاپك، كنترل انسانها توسط روباتها بود، ولي او هرگونه امكان جايزگزيني انسان با روبات و يا اينكه روباتها از احساس برخوردار شوند، عاشق شوند، يا تنفر پيدا كنند را رد مي كرد.

- سال 1940 م: شركت وستينگهاوس (Westinghouse Co.) سگي به نام اسپاركو (Sparko) ساخت كه هم از قطعات مكانيكي و هم الكتريكي در ساخت آن استفاده شده بود. اين اولين باري بود كه از قطعات الكتريكي نيز همراه با قطعات مكانيكي استفاده مي شد.

- سال 1942 م: كلمه روباتيك (robotics) اولين بار توسط ايزاك آسيموف در يك داستان کوتاه ارائه شد. ايزاك آسيموف (1920-1992) نويسنده كتابهاي توصيفي درباره علوم و داستانهاي علمي تخيلي است.

- دهه 1950 م: تكنولوجي كامپيوتر پيشرفت كرد و صنعت كنترل متحول شد. سؤلاتي مطرح شدند. مثلاً: آيا كامپيوتر يك روبات غير متحرك است؟

- سال 1954 م: عصر روبات ها با ارائه اولين روبات آدم نما توسط جرج دوول (George Devol) شروع شد.

امروزه، 90% روباتها، روباتهاي صنعتي هستند، يعني روباتهايي كه در كارخانه ها، آزمايشگاهها، انبارها، نيروگاهها، بيمارستانها، و بخشهاي مشابه به كارگرفته مي شوند. در سالهاي قبل، اكثر روباتهاي صنعتي در كارخانه هاي خودروسازي به كارگرفته مي شدند، ولي امروزه تنها حدود نيمي از روباتهاي موجود در دنيا در كارخانه هاي خودروسازي به كار گرفته مي شوند. مصارف روباتها در همه ابعاد زندگي انسان به سرعت در حال گسترش است تا كارهاي سخت و خطرناك را به جاي انسان انجام دهند. براي مثال امروزه براي بررسي وضعيت داخلي رآكتورها از روبات استفاده مي شود تا تشعشعات راديواكتيو به انسانها صدمه نزنند.

- سال 1956 م: پس از توسعه فعاليتهاي تكنولوجي يك كه بعد از جنگ جهاني دوم، يك ملاقات تاريخي بين جورج سي. دوول (George C. Devol) مخترع و كارآفرين صاحب نام، و ژوزف اف. انگلبرگر (Joseph F. Engelberger) كه يك مهندس با سابقه بود، صورت گرفت. در اين ملاقات آنها به بحث در مورد داستان آسيموف پرداختند. ايشان سپس به موفقيتهاي اساسي در توليد روباتها دست يافتند و با تأسيس شرکتهای تجاري، به توليد روبات مشغول شدند. انگلبرگر شرکت Unimate برگرفته از Automation Universal را براي توليد روبات پايه گذاري كرد. نخستين روباتهاي اين شرکت در كارخانه جنرال موتورز (General Motors) براي انجام كارهاي دشوار در خودروسازي به كار گرفته شد. انگلبرگر را "پدر روباتيك" ناميده اند.

- دهه 1960 م: روباتهاي صنعتي زيادي ساخته شدند. انجمن صنايع روباتيك اين تعريف را براي روبات صنعتي ارائه كرد:

"روبات صنعتي يک وسیله چند کاره و با قابليت برنامه ريزي چند باره است كه براي جابجايي قطعات، مواد، ابزارها يا وسايل خاص بوسيله حرکات برنامه ريزي شده، براي انجام كارهاي متنوع استفاده مي شود."

- سال 1962 م: شرکت خودروسازي جنرال موتورز نخستين روبات Unimate را در خط مونتاژ خود به كار گرفت.

- سال 1967 م: رالف موزر (Ralph Moser) از شرکت جنرال الكتريك (General Electric) نخستين روبات چهارپا را اختراع كرد.

- سال 1983 م: شرکت Odetics يك روبات شش پا ارائه کرد که مي توانست از موانع عبور کند و بارهاي سنگيني را نيز با خود حمل کند.

- سال 1985 م: نخستين روباتي که به تنهائي توانايي راه رفتن داشت در دانشگاه ايالتي اهايو (Ohio State Uneversity) ساخته شد.

- سال 1996 م: شرکت ژاپني هندا (Honda) نخستين روبات انسان نما را ارائه کرد که با دو دست و دو پا طوري طراحي شده بود که مي توانست راه برود، از پله بالا برود، روي صندلي بنشيند و بلند شود و بارهايي به وزن 5 کيلوگرم را حمل کند

روباتها روز به روز هوشمندتر مي شوند تا هرچه بيشتر در کارهاي سخت و پر خطر به ياري انسانها بيابند.

قانون روباتيك مطرح شده توسط آسيموف:

- 1- روبات ها نبايد هيچگاه به انسانها صدمه بزنند.
- 2- روباتها بايد دستورات انسانها را بدون سرپيچي از قانون اول اجرا کنند.
- 3- روباتها بايد بدون نقض قانون اول و دوم از خود محافظت کنند.

انواع ربات ها :

روباتهاي امروزي که شامل قطعات الکترونيکي و مکانیکي هستند در ابتدا به صورت بازوهاي مکانیکي براي جابجايي قطعات و يا کارهاي ساده و تکراري که موجب خستگي و عدم تمرکز کارگر و افت بازده ميشد بوجود آمدند. اينگونه رباتها جابجاگر (manipulator) نام دارند. جابجاگرها معمولا در نقطه ثابت و در فضاي کاملا کنترل شده در کارخانه نصب ميشوند و به غير از وظيفه اي که به خاطر آن طراحي شده اند قادر به انجام کار ديگري نيستند. اين وظيفه ميتواند در حد بسته بندي توليدات، کنترل کيفيت و جدا کردن توليدات بي کيفيت، و يا کارهاي پيچيده تري همچون جوشکاري و رنگزني با دقت بالا باشد.



نوع دیگر رباتها که امروزه مورد توجه بیشتری است رباتهای متحرک هستند که مانند رباتهای جابجا کننده در محیط ثابت و شرایط کنترل شده کار نمیکنند. بلکه همانند موجودات زنده در دنیای واقعی و با شرایط واقعی زندگی میکنند و سیر اتفاقاتی که ربات باید با آنها روبرو شود از قبل مشخص نیست. در این نوع ربات هاست که تکنیک های هوش مصنوعی میبایست در کنترلر ربات (مغز ربات) به کار گرفته شود.



رباتهای متحرک به دسته های زیر تقسیم بندی میشوند:

1- رباتهای چرخ دار

با انواع چرخ عادی



و یا شنی تانک



و با پیکربندی های مختلف یک، دو یا چند قسمتی



2-رباتهاي پادار مثل سگ اسباب بازي AIBO ساخت سوني که در شکل بالا نشان داده شد یا ربات ASIMO ساخت شرکت هوندا



3-رباتهاي پرنده



4-رباتهاي چند گانه(هايبريد) که ترکيبي از رباتهاي بالا یا ترکيب با جابجاگرها هستند



9 ...

مزایای روباتها:

- 1- روباتيك و اتوماسيون در بسياري از موارد مي توانند ايمني، ميزان توليد، بهره و كيفيت محصولات را افزايش دهند.
 - 2- روباتها مي توانند در موقعيت هاي خطرناك كار كنند و با اين كار جان هزاران انسان را نجات دهند.
 - 3- روباتها به راحتی محيط اطراف خود توجه ندارند و نيازهاي انساني براي آنها مفهومي ندارد. روباتها هيچگاه خسته نمي شوند.
 - 4- دقت روباتها خيلي بيشتر از انسانها است آنها در حد ميلي يا حتي ميكرو اينچ دقت دارند.
 - 5- روباتها مي توانند در يك لحظه چند كار را با هم انجام دهند ولي انسانها در يك لحظه تنها يك كار انجام مي دهند.
- معايير روباتها:

- 1- روباتها در موقعيتهاي اضطراري توانايي پاسخگويي مناسب ندارند كه اين مطلب مي تواند بسيار خطرناك باشد.
- 2- روباتها هزينه بر هستند.
- 3- قابليت هاي محدود دارند يعني فقط كاري كه براي آن ساخته شده اند را انجام مي دهند.

تعريف ربات مسيرياب

منظور از روبات مسير ياب دستگاهی است که به منظور طی کردن مسیر مشخص شده پردازد و از موانع و اشکالات احتمالی ایجاد شده در مسیر دچار سرگردانی نشده و مسیر خود را به درستی طی نماید تا به انتهای مسیر بدون هیچ گونه خطای برسد.

که برای تشخیص خط زمین ساخته شده برای سنسورهای مادون قرمز از دو رنگ

مختلف استفاده می شود که این دو رنگ سفید و سیاه می باشد که میتوان این زمین

را به دو شکل طراحی کرد.

1. خط مورد نظر برای طی کردن مسیر (سیاه) و زمینه (سفید)

2. خط مورد نظر برای طی کردن مسیر (سفید) و زمینه (سیاه)

3. استفاده از این روبات در صنعت

1. از این روبات (از نوع صنعتی) می توان در مراکز صنعتی و خدماتی به

عنوان انقال دهنده و حمل کننده ی کالا استفاده کرد یا استفاده پیشرفته تر از این

روبات می توان به اتبوسهای بدون راننده اشاره کرد که هم کنون در ژاپن در

حال استفاده می باشد که این روباتهای پیشرفته در یک مسیر که از پیش تعیین

شده است قرار می گیرند ودر این مسیر شروع به حرکت می کنند.

2. همچنین به عنوان نمونه کارخانه ایی را فرض کنید که دارای سیستم انبار

مکانیزه است و تحویل گرفتن و تحویل دادن کالا در این مجموعه به طور خودکار

انجام می شود که توسط این روبات ها این کالا ها از قسمتی به قسمت دیگر انتقال

داده می شودواین کار کاملاً اتوماتیک انجام می شود.

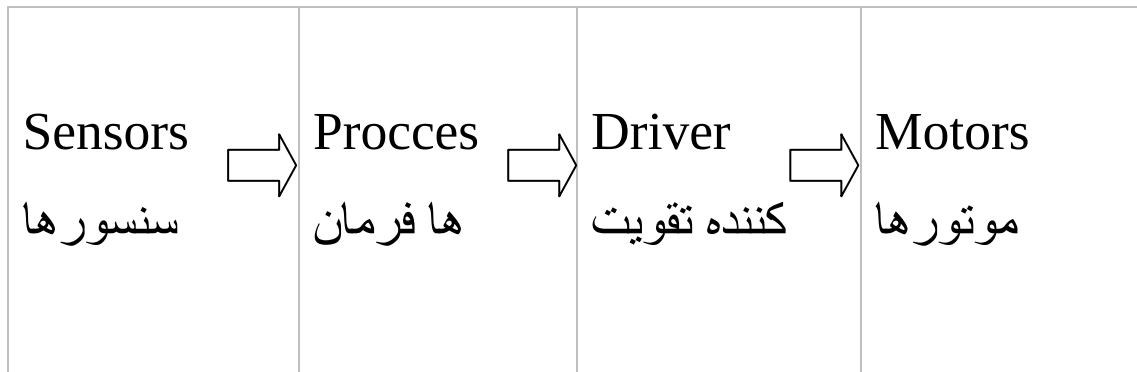
که می توان با مشخص کردن مسیرهای مختلفی برای قسمتهای مختلف از این

سیستم برای ارسال کالا ها به قسمتهای مختلف استفاده کرد.

که می توان مثالهای زیادی از این سیستم آورد در این قسمت می خواهیم به

معرفی اجمالی این سیستم پردازیم .

بلوک دیاگرام یک روبات مسیر یاب



همان طور که از شکل مشخص می باشد ابتدا سنسورها بعد از این که بایاس شدند زمین را حس کرده و بعد با توجه به سیاه و سفید بودن زمین 0 و 1 هایی به بلوکفرمان می فرستند و مدار فرمان با توجه به این کدها تصمیم گرفته و فرمان راصادر می کند بعد سیگنال فرمان به وسیله ی تقویت کننده تقویت شده و وارد موتورها می شود و روبات را به حرکت می اندازد .

این یک دید کلی از کارکرد روبات مسیریاب به صورت بلوکی بود تا اشنای کلی نسبت به این روبات پیدا کرده باشیم.

به طور کلی روبات از دو قسمت اساسی تشکیل شده است.

1. سخت افزار

2. نرم افزار

که به بررسی هر یک می پردازیم.

سخت افزار

برای ساخت یک روبات احتیاج به عناصر یا قطعاتی است تا ارتباط بین روبات و محیط خارج برقرار شود تا این اطلاعاتی که از محیط دریافت میشود روبات نسبت به ان واکنش نشان دهد.

که این کار توسط سخت افزار ها انجام می شود .

سخت افزارهای استفاده شده دراین ربات مسیر یاب عبارت است از:

سنسور

سنسور وسیله ای است که یک متغیر فیزیکی(مانند نور،حرارت،فشار،نیرو و...)را به یک سیگنال الکتریکی تبدیل می نماید که با توجه به نیاز میتواند منتقل شده و یا مورد پردازش قرار گیرد.

سنسورهای انفعالی (passive) و فعال (active):

سنسورهای انفعالی، سنسورهایی هستند که در فرایند اندازه گیری، انرژی اضافه نمی کنند، بلکه ممکن است در این فرایند انرژی را نیز هز بین ببرند.

سنسورهای فعال ، به محیط اندازه گیری به عنوان بخشی از فرایند اندازه گیری ، انرژی اضافه می کنند مانند سیستم sonar که امواج صوتی را می فرستند.

درجه بندی و اندازه داری (calibration):

رابطه بین متغیر فیزیکی ورودی و متغیر سیگنال خروجی برای یک سنسور مشخص را کالیبراسیون سنسور گویند که این کار معمولاً با مشاهده ی پاسخ یک ورودی شناخته شده انجام می پذیرد.

سنسورها در ربات تعقیب خط:

در ربات تعقیب خط تنها متغیر فیزیکی قابل اندازه گیری نور است . سنسورهای نوری پس از انجام فرآیند اندازه گیری نور محیط (زمینه سفید یا خط سفید) نتیجه را به صورت یک سیگنال الکتریکی (معمولاً یک سطح 5 ولت) ارائه می کنند که آماده ی پردازش در پردازنده ربات است.

سنسور های نوری:

سنسورهای نوری برای شناسایی رنگ پیست و خط بکار می روند و کلیه رباتهای تعقیب خط به آن احتیاج دارند. سنسورها این عمل را با تاباندن نور به سطح پیست و اندازه گیری میزان بازتاب نور انجام می دهند.

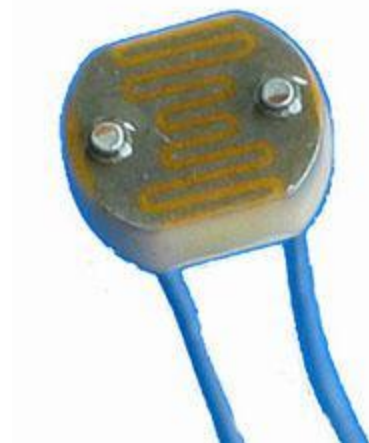
رنگهای مختلف نور را با شدت های مختلفی بازتاب می کنند ، زیرا رنگهای مختلف طیف های مختلفی از نور را جذب می کنند. رنگ سفید بیشتر نور را بازتاب می کند و رنگ سیاه بیشتر نور تابیده شده را جذب می کند.

منبع نور معمولاً یک LED (Light Emitting Diode) است و در کنار آن یک سنسور برای اندازه گیری شدت نور بازتابی نصب می شود.

رایج ترین سنسور هایی که در این زمینه بکار می روند عبارتند از : فتوسل ، فتودیود و فتوترانزیستور . در اینجا نگاهی به طراحی های با فتوسل (مقاومت نوری) و فتوترانزیستور خواهیم داشت ، سپس سنسور های مادون قرمز را بررسی خواهیم کرد .

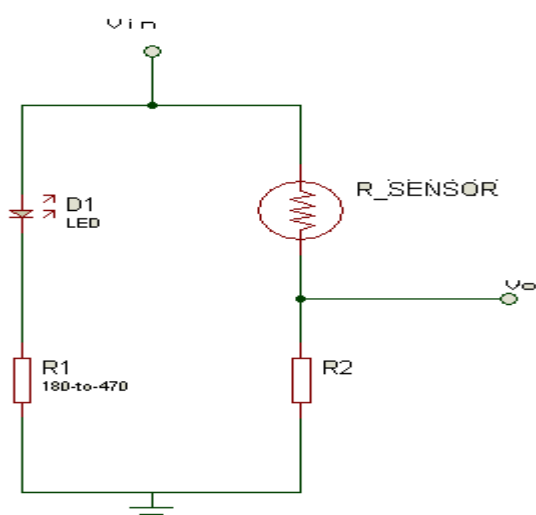
فتوسل (مقاومت نوری):

مقاومت نوری یک مقاومت متغیر است و مقدار مقاومت آن با توجه به شدت نور تابیده شده به آن تغییر می کند . این سنسورها معمولاً سولفید کادمیم (Cds) یا فتوسل نامیده می شوند.



مقاومت داخلی فتوسل عکس شدت نور تابیده شده به آن است . وقتی این سلولها در تاریکی کامل قرار می گیرند مقاوت داخلی آنها بسیار بالاست و در حدود 100 کیلو اهم تا 10 مگا اهم با توجه به نوع سلول است .

شکل زیر چگونگی استفاده از فتوسل را جهت اندازه گیری شدت نور بازتابی و با استفاده از یک مقسم ولتاژ نشان می دهد. مقاومت 470 اهمی برای محدود کردن مقدار جریان عبوری از LED بکار رفته است. استفاده از مقاومت های کوچکتر شدت روشنایی LED را افزایش می دهد و حساسیت سنسور را افزایش می دهد .



در فرمول زیر رابطه ی بین ولتاژ خروجی (V_{out}) با ولتاژ ورودی (V_{in}) و مقاومت نوری (R_{sensor}) و مقاومت دیگر (R_1) بیان شده است . برای استفاده از این سنسورها باید از میکرو کنترلرهایی که مبدل آنالوگ به دیجیتال (A/D) دارند و یا از یک مبدل آنالوگ به دیجیتال (A/D) خارجی مانند ADC081 استفاده کرد.

$$V_{out} = (V_{in} * R_1) / (R_1 + R_{sensor})$$

مقدار مقاومت R_1 اختیاری است . اما اگر مقدار آن کوچک باشد ولتاژ خروجی به تغییرات نور محیط حساس نخواهد بود و اگر مقدار آن بزرگ باشد به آزمایش های بیشتری برای تنظیم ولتاژ خروجی نیاز خواهد بود . پیشنهاد می شود بجای R_1 از یک پتانسیومتر که قادر به اندازه گیری مقادیر 50 تا 100 کیلو اهم باشد استفاده کنید . پس از آن ولتاژ خروجی را ضمن عبور از روی خط سیاه و زمینه سفید اندازه گیری کنید . با تغییر مقدار R_1 بهترین مقدار ممکن برای ولتاژ خروجی سنسور پیدا می شود .

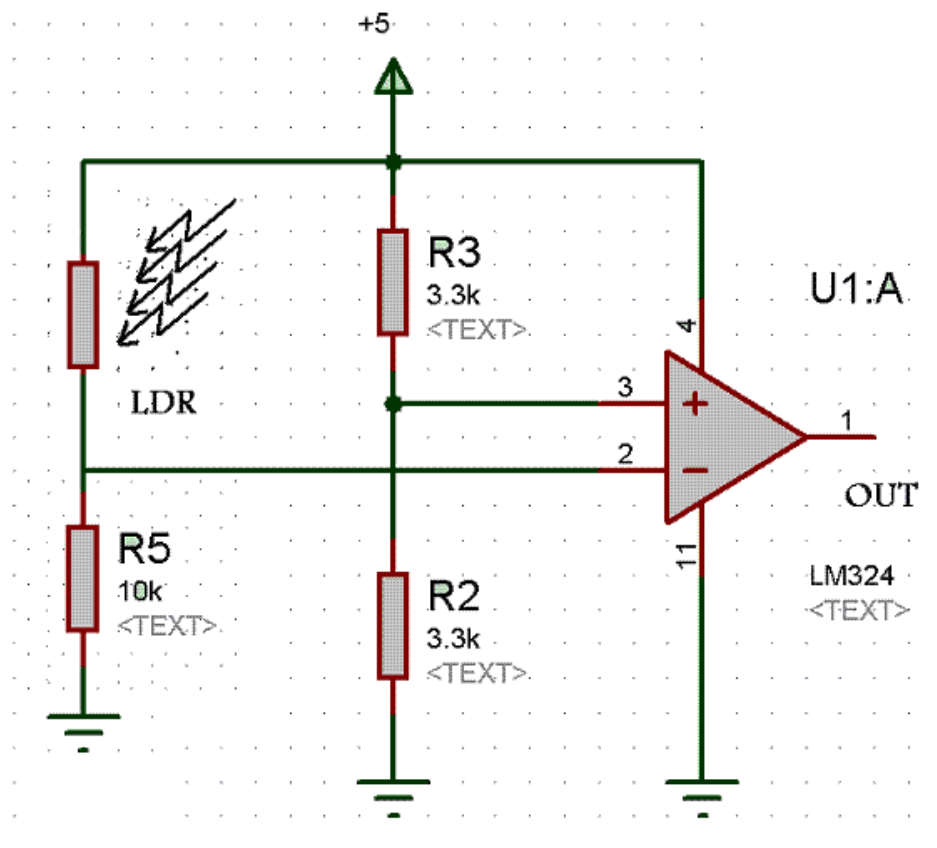
در مرحله بعد مقدار مقاومت پتانسیومتر را اندازه گیری کرده و آن را با یک مقاومت ثابت جایگزین می کنیم.

مجدداً سنسور را روی خط مشکی قرار داده در این حالت ولتاژ خروجی باید کمتر از یک ولت باشد . پس از آن سنسور را روی زمینه ی سفید قرار داده در این حالت ولتاژ خروجی باید بیش از 3 ولت باشد .

باید از LED های نور مرئی برای این نوع سنسورها استفاده کرد. زیرا فتوسل ها نسبت به نور مادون قرمز حساس نیستند.

همواره باید سنسورها را نسبت به نور محیط عایق بندی کرد تا نور محیط موجب تداخل در عملکرد سنسورها نگردد . وجود نور بیش از حد محیط باعث عدم تشخیص رنگ مشکی توسط سنسورها می شود.

مدار زیر را می‌توانید برای قسمت سنسور ربات استفاده شود . که از اپ امپ استفاده شده آی سی 741 یک اپ امپ می‌باشد میتوان از آی سی LM 324 نیز استفاده شود که شامل 4 اپ امپ می باشد .



ایراد اصلی فتوسل ها سرعت کم آنها در تشخیص رنگها در مقایسه با فتو ترانزیستورها است . برای رباتهای کندتر فتوسل ها بسیار کارآمد هستند ولی برای رباتهای سریع تر از فتو ترانزیستورها استفاده می شود .

همانطور که گفته شد از مزایای استفاده از LDR در ربات مسیر یاب فاصله مناسب آن از سطح زمین و سادگی آن میباشد . و از معایب آن : چون این سنسور حساس به نور معمولی میباشد احتمال تداخل نور محیط و اشتباه کردن ربات زیاد میباشد و اطراف این سنسور باید طوری عایق بندی شود که نور محیط به آن نتابد .

فتو ترانزیستورها :

به جای استفاده از فتوسل ها می توان از یک فتو ترانزیستور در یک مدار مشابه استفاده کرد و رنگ سیاه و زمینه ی سفید را بخوبی شناسایی نمود. در فتو ترانزیستورها مقدار جریانی که از ترانزیستور جاری می شود مقدار نور دریافتی را مشخص می کند.

مقاومت داخلی فتو ترانزیستور با افزایش شدت نور کمتر و کمتر می شود . در فتو ترانزیستور مقدار جریان تا اشباع شدن ترانزیستور افزایش می یابد در این حالت ترانزیستور مانند یک کلید بسته عمل می کند و نمی تواند جریان بیشتری را متناسب با افزایش شدت نور هدایت کند. وقتی ترانزیستور روی خط سیاه پیست قرار دارد نباید جریانی را عبور دهد . در این حالت مقاومت 10 کیلو اهمی به عنوان مقاومت پول آپ بکار رفته است . بنابراین ولتاژ خروجی در حدود 5 ولت است . وقتی سنسور بر روی سطح سفید پیست قرار دارد ، فتو ترانزیستور باید اشباع کامل شده باشد . بنابراین ولتاژ خروجی صفر خواهد شد . البته این نتایج در شرایط ایده آل بدست می آید و در شرایط واقعی باید سنسور ها را تنظیم کرد .

پتانسیومتر 1 کیلو اهمی برای تنظیم شدت نور LED بکار می رود . با تغییر شدت نور LED ، ولتاژ خروجی (هم روی خط مشکی و هم روی سطح سفید) تغییر خواهد کرد .

می توان از یک مبدل آنالوگ به دیجیتال برای محاسبه ی نقطه ی حساس ولتاژ استفاده کرد .

همچنین می توان خروجی ولتاژ را حداقل 3 ولت بر روی خط مشکی و صفر ولت روی زمینه سفید تنظیم کرد و آن را مستقیماً به میکرو کنترلر متصل نمود . بیشتر میکرو کنترلرها ولتاژ بالای 1.5 را به عنوان "یک منطقی" و ولتاژ زیر 1.5 را به عنوان "صفر منطقی" تفسیر می کنند و بنابراین اگر خروجی سنسور حداقل 3 ولت روی خط مشکی و زیر 1.5 ولت روی زمینه ی سفید باشد . میکروکنترلر میتواند از این تغییر مقیاس برای تشخیص عبور از روی خط مشکی یا زمینه سفید استفاده کند .

فتو ترانزیستورها بسیار سریعتر از فتوسل ها عمل می کنند(فتو ترانزیستورها خط را در چند میکرو ثانیه و فتوسل ها در چند میلی ثانیه تشخیص می دهند)

علاوه بر این ، نتیجه هر دو روش با توجه به فاصله بین سنسورها و سطح پیست تغییر می کند . این عامل با توجه به حساسیت مورد نیاز می تواند سودمند یا مضر باشد . برای یک مدار حساس یا با حساسیت در مقادیر خاص ، سنسور باید به پیست نزدیک باشد . افزایش فاصله باعث کاهش حساسیت سنسور خواهد شد .

در حالت کلی بهترین فاصله تنظیم سنسور 1.5 تا 3 میلی متر از سطح پیست است که به نوع سنسور نیز وابسته است .

سنسور مادون قرمز :

بر خلاف فتوسل ها ، فتو ترانزیستورها را هم می توان با نور مرئی و هم مادون قرمز بکار برد . سنسورهای مادون قرمز در مقایسه با سنسورهای نور مرئی حساسیت کمتری به نور محیط دارند.

از آنجایی که پرتوهای مادون قرمز دیده نمی شوند ، خاموش یا روشن بودن مدار به صورت مستقیم قابل تشخیص نیست . برای دیدن امواج مادون قرمز به انواعی از آشکارسازها نیاز دارید . برخی از انواع دوربین های دیجیتال نسبت به نور مادون قرمز حساسند . اگر LED مادون قرمز را با دوربین مشاهده کرد و LED از خود نور آبی ساطع کند ، LED روشن است و نور مادون قرمز منتشر می کند . بیاد داشته باشید که LED های مادون قرمز فقط با فتو ترانزیستورهای مادون قرمز بکار می روند و LED های نور مرئی فقط با فتو ترانزیستورهای نور مرئی استفاده می شوند . جابجا کردن دو LED کل نتایج را بی اعتبار می کند . وقتی LED و سنسور را به صورت آماده خریداری می کنیم باید دقت کرد تا طول امواج هر دو یکسان باشد . بیشتر LED های مادون قرمز موجود در بازار طول موج های 880 تا 940 نانو متر دارند .

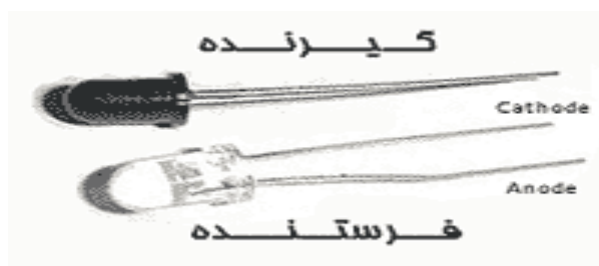
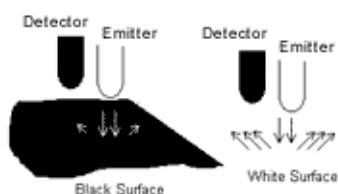
در برخی موارد LED و سنسور با طول موج های متفاوت بکار می روند . اما بیشترین حساسیت وقتی بدست می آید که LED و سنسور طول موج یکسان داشته باشند . بجای استفاده از LED های مادون قرمز و فتو ترانزیستور می توان از انواع بسته بندی شده آنها (LED و سنسور در یک بسته) استفاده کرد .

فرستنده گیرنده مادون قرمز (IR)

در این مدار به جای اینکه از سنسور هایی که در یک پک هستند استفاده کنیم از دو سنسور IR معمولی استفاده میکنیم .
سنسور های مورد نیاز ما سنسور های فرستنده و گیرنده معمولی IR - سه میلیمتری میباشند که هر کدام دو پایه دارند .
نوع سنسور : IR یا همان سنسور های کنترل و تلویزیون فرستنده : سنسور بی رنگ
گیرنده : سنسور تیره رنگ
فاصله بین سنسور فرستنده و گیرنده 2 تا 3 میلیمتر

نحوه بستن مدار

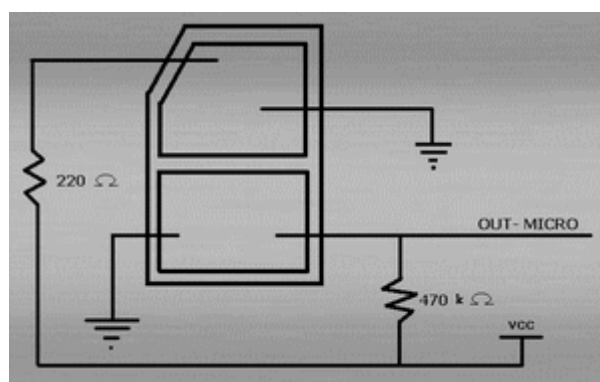
برای بستن مدار فرستنده باید پایه بلند(آند) را به یک مقاومت 330 و به VCC یا همون برق استانداردمون و پایه کوتاه آن(کاتد) رو به GND وصل کنیم
برای بستن مدار گیرنده باید پایه بلند رو به GND و پایه کوتاه اون رو به یک مقاومت 2 کیلو اهمی و به VCC وصل کرد
و از وسط پایه کوتاه و مقاومت گیرنده ولتاژ مورد نظر برای میکرو کنترلر گرفته میشود



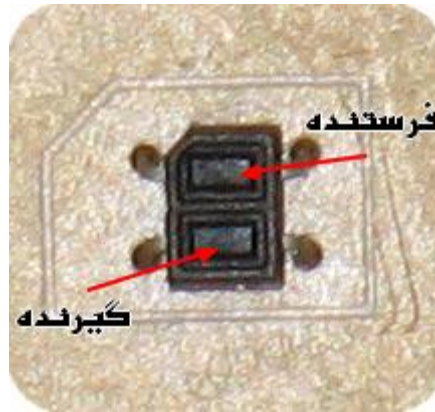
اما همین سنسورها در پک های بصورت آماده وجود دارند که کار کردن با آنها بسیار آسان میباشد . در زیر چند نمونه از آنها و نحوه استفاده از آنها را آورده ایم .

سنسور GP2S09

این سنسور یک پک آماده (فرستنده و گیرنده) می باشد سایز این سنسور بسیار کوچک میباشد و مطابق شکل زیر دارای 4 پایه میباشد.



این سنسور ساده قیمت مناسبی دارد و از معایب آن فاصله بسیار کم آن از سطح زمین میباشد (تقریبا چسبیده به زمین) برای افزایش فاصله میتوانیم مقاومت روی فرستنده (220 اهم) را کمتر بگیریم (مثلا اهم 150) تا ولتاژ سر فرستنده بیشتر شود که البته با انجام این کار دو مشکل دیگر پیش میآید یکی آنکه احتمال سوختن سنسور زیاد میشود دوم آنکه با کم کردن مقاومت مصرف آن زیاد میشود و تغذیه آن خود یک موزل میشود .



سنسور های دیگری از این نوع وجود دارند که تقریبا مشابه می باشند با اندک تفاوتی در چینش پایه ها مانند :

GP2S06 - GP2S07 - GP2S08 - GP2S09 - GP2S10

CNY70

این سنسور شامل یک دیود فرستنده مادون قرمز یا IR و یک ترانزیستورنوری (phototransistor) گیرنده می باشد که با طول موج 950 نانومتر کار میکند . این ترانزیستور مانند ترانزیستور معمولی میباشد که به جای تحریک بیس از طریق جریان بیس آن با نورتحریک میشود .

این سنسور از بهترین سنسور های موجود در بازار میباشد که مزایای همه سنسور های بالا را دارا می باشد و عیب آن قیمت گران آن می باشد . در زیر شکل و مدار آن را میبینید.

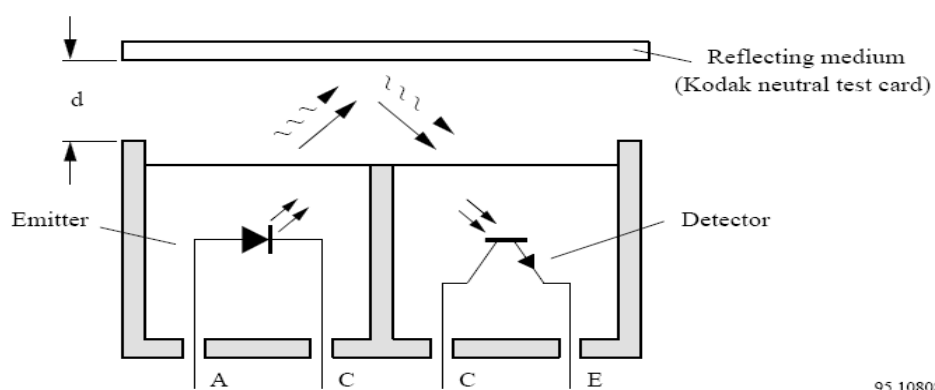
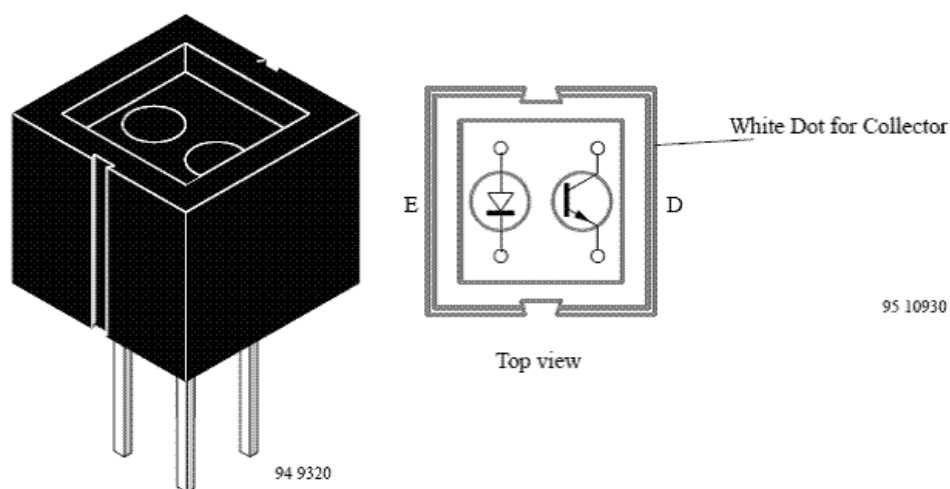
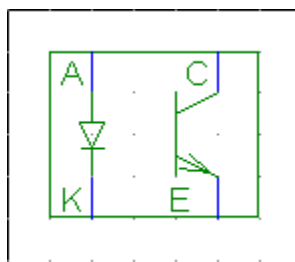


Figure 1. Test circuit

با بررسی سنسورهای موجود در بازار و بررسی دیتاشیت سنسورهای موجود و کارایی آنها در مدارات و پاسخ آنها ما نوع سنسور خود را از نوع CNY70 که به صورت یک پک فرستنده و گیرنده می باشد ، انتخاب کردیم .

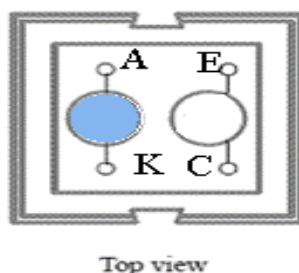
پایه های CNY70

برای تشخیص پایه های این سنسور می توان از دیتا شیت این سنسور استفاده کرد . با توجه به شکل شماتیک آن که در زیر آمده است دارای 4 پایه است .



هنگامی که قطعه را روبروی خودمان بگیریم دارای دو چشم دایره ای می باشد که یکی از آنها آبی رنگ است و دیگری بدون رنگ است . در حالتی که قطعه را روبروی خود گرفته ایم و چشم آبی در سمت چپ قرار دارد (شکل زیر) چشم آبی رنگ دیود فرستنده و چشم بی رنگ فتوترانزیستور است در همین حالت پایه ی بالایی چشم آبی آند دیود فرستنده و پایه ی پایین چشم آبی کاتد دیود فرستنده است . در همین حالت پایه ی بالای

چشم بدون رنگ امیتر فتوترانزیستور و پایه ی پایین آن کلکتور فتوترانزیستور است .

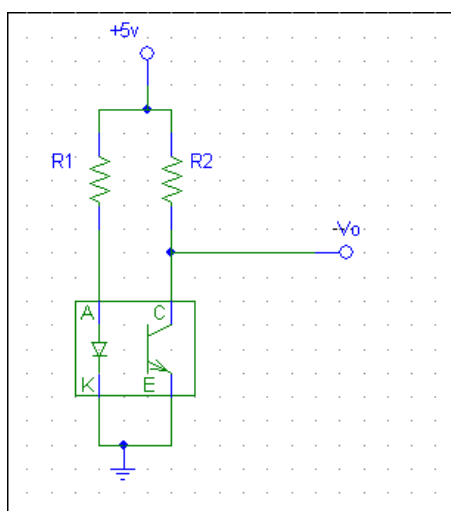


Top view

توجه شود با وجود آنکه در شکل شماتیک امیتر در پایین قرار دارد اما قطعه پایه امیتر در بالا قرار گرفته است .

مدار دایور سنسورها و تشریح عملکرد آن

چون نحوه ی عملکرد سنسورهای مادون قرمزی که در یک پک قرار دارند همانند عملکرد فتوترانزیستورها می باشد و تنها تفاوت آنها با فتوترانزیستورها در این است که فرستنده و گیرنده در یک بسته قرار دارند لذا مدار دایور سنسورها نیز همانند مدار دایور سنسورها می باشد. در شکل زیر مدار دایور سنسورها آمده است .



در این مدار R1 برای محدود کردن جریان دیود فرستنده بکار رفته و مقدار آن در حدود 200-470 اهم است . فتوترانزیستوری هم که در مدار قرار دارد بیس آن با نور تحریک شده و هنگامی که نور به حد کافی به آن برسد هدایت می کند . مقاومت R2 مقاومت کلکتور فتوترانزیستور است که برای بایاس آن بکار رفته است .

قبل از بررسی عملکرد مدار باید ذکر کنیم براساس قوانین انتشار نور به سطوح رنگی داریم :

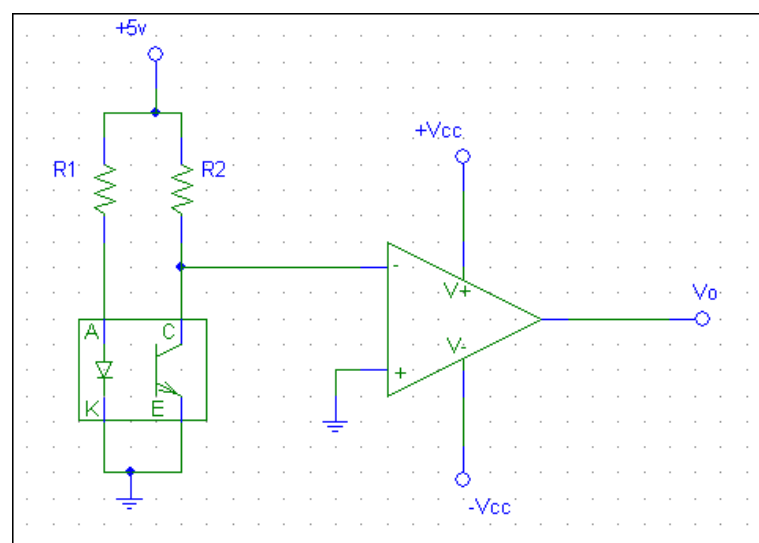
هنگامی که نوری به یک سطح رنگی برخورد می کند مقداری از آن نور جذب رنگ شده و مقداری از آن برگشت داده می شود . مقدار نور جذب شده و مقدار نور بازتاب به نوع رنگ بستگی دارد بدین ترتیب که در این انتشار رنگ سیاه بیشترین درصد نور را جذب کرده و کمترین بازتاب را دارد و رنگ سفید عکس نور سیاه کمترین درصد نور را جذب کرده و تقریباً تمام آن را بازتاب می کند . حال با توجه به این قانون به تشریح مدار می پردازیم :

هنگامی که سنسور روی نوار سیاه رنگ قرار می گیرد نور فرستاده شده از دیود فرستنده تا حدود زیادی جذب رنگ شده و مقدار کمی از آن بازتاب پیدا کرده و به بیس فتوترانزیستور می رسد که این مقدار نور برای هدایت فتوترانزیستور کافی نبوده و فتوترانزیستور را در حالت قطع نگه می دارد . در این حالت خروجی مدار (کلکتور فتوترانزیستور) برابر V_{CC} شده که سطح یک منطقی است .

عکس این عمل برای سنسور هنگامی که روی سطح سفید قرار دارد اتفاق می افتد یعنی وقتی سنسور روی سطح سفید قرار می گیرد نور فرستاده شده توسط دیود فرستنده توسط سطح سفید بازتاب می یابد و به بیس فتوترانزیستور رسیده و آنرا روشن می کند و مقدار V_0 متناسب با اینکه فتوترانزیستور به ناحیه ی قطع یا اشباع برود تا حد زیادی پایین می آید.

چون در مدار بالا خروجی بین دو سطح منطقی صفر و یک نوسان نکرده و چون میکروکنترلرها با دو سطح صفر و یک منطقی کار می کنند لذا باید خروجی مدار را توسط مداری مداری به صفر و یک منطقی تبدیل کرد . برای این کار می توان از op-amp استفاده کرد چرا که هنگامی که از

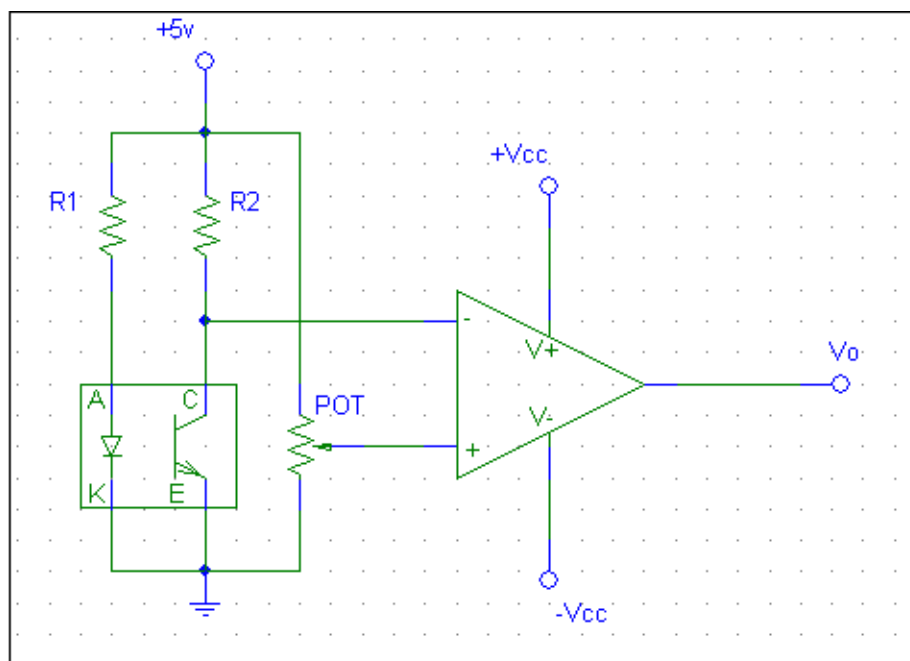
op-Amp در حلقه باز (open loop) استفاده کنیم خروجی op-Amp به ازای کوچکترین تغییرات ورودی آن به اشباع مثبت و منفی می رود و می توان از این ولتاژ برای سطح صفر و یک منطقی استفاده کرد پس مدار را به صورت زیر اصلاح می کنیم .



در این مدار هنگامی که سنسور روی خط سیاه باشد فتوترانزیستور خاموش می باشد و لذا پایه ی منفی op-Amp برابر V_{CC} شده و چون پایه ی مثبت op-Amp سطح صفر است و $V_i \rightarrow V_{i+}$ شده لذا op-Amp به سطح اشباع منفی می رود .

هنگامی که سنسور روی خط سفید قرار دارد ترانزیستور هدایت کرده و V_o پایین می آید اما هیچگاه به صفر ولت دقیق نمی رسد . حداقل مقدار V_o هنگامی است که فتوترانزیستور اشباع شود که در این حالت ولتاژ کلکتور امیتر آن حدود 0.2 ولت می شود (VCEsat) که در این حالت نیز با وجود آنکه فتوترانزیستور هدایت کرده اما باز هم op-Amp به اشباع منفی می رود چرا که ولتاژ پایه ی منفی op-Amp که در حدود 0.2 ولت است از ولتاژ پایه ی مثبت op-Amp که صفر ولت است بزرگتر است و این امر op-Amp را به اشباع منفی می رود .

در مدار بالا با وجود استفاده از op-Amp و تغییر رنگ زمینه ی سنسور اما خروجی op-Amp در اشباع منفی باقی می ماند و تغییر حالت نمی داد . پس باید مدار را طوری اصلاح کرد که با تغییر رنگ زمینه ی سنسور خروجی آن متناسب با رنگ تغییر حالت دهد . پس مدار را به صورت زیر اصلاح کرد .

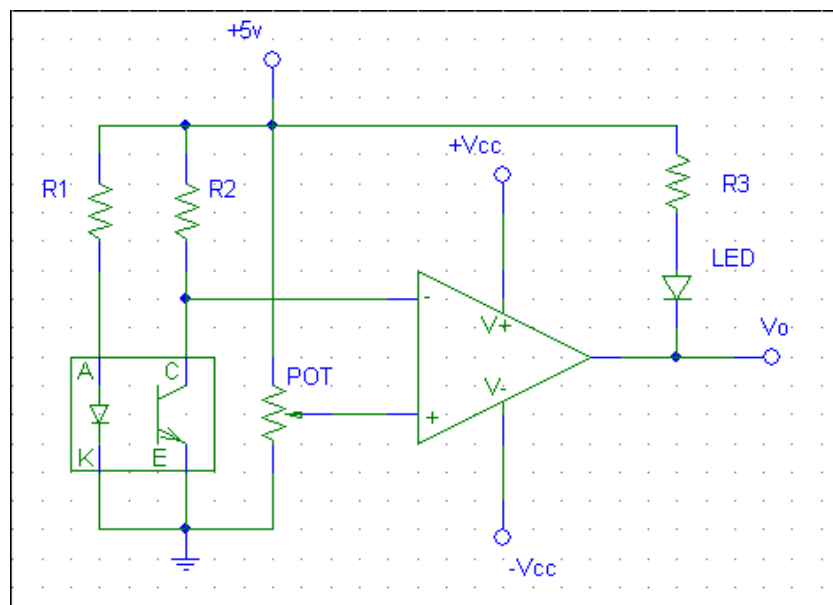


نحوه ی کارکرد این مدار به صورت مدارات قبلی می باشد با این تفاوت وقتی که سنسور روی خط سیاه قرار دارد ولتاژ کلکتور فتوترانزیستور برابر V_{CC} می شود و چون این ولتاژ از ولتاژ سر وسط پتانسیومتر (قسمتی از V_{CC}) بیشتر است باعث

می شود op-Amp به اشباع منفی برود ($V_i > V_{i+}$) و هنگامی که سنسور روی سطح سفید قرار می گیرد ولتاژ کلکتور فوتوترانزیستور پایین می آید چرا که نور برگشتی فوتوترانزیستور را روشن کرده است در این حالت چون ولتاژ پایه ی مثبت از ولتاژ پایه ی منفی بیشتر می شود op-Amp به اشباع مثبت می رود. از توضیحات بالا می توان نتیجه گرفت :

پتانسیومتر را باید در حدی تنظیم کرد که وقتی فوتوترانزیستور روشن است ولتاژ پایه ی مثبت op-Amp از ولتاژ پایه ی منفی بیشتر شود تا op-Amp را به اشباع مثبت بربرد (ولتاژ پایه ی مثبت $<$ ولتاژ کلکتور فوتوترانزیستور در حالتی که روی سطح سفید قرار دارد)

با قرار دادن یک LED در خروجی مدار با یک مقاومت که جریان آنرا محدود می کند مدار درایور سنسورها را تکمیل می کنیم . در این حالت می توان با تغییر دادن پتانسیومتر زمانی که LED روشن و خاموش میشود بهترین حالت را برای آن تنظیم کرد بدین ترتیب که مرز بین روشن و خاموش شدن LED ایده آل ترین حالت تنظیم پتانسیومتر است .



در مدار بالا

$R1=300$

$R2=10K$

$POT=20K$ or $10K$

Sensor=CNY70

Op-amp=LM324

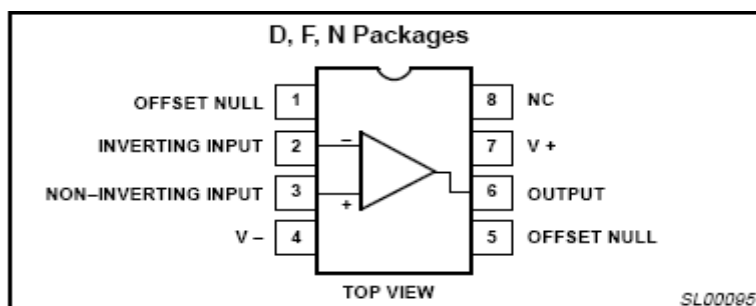
$R3=300$

$VCC=5V$

Op-amp مورد استفاده در مدار از نوع LM324 می باشد که می توان از 741 نیز استفاده کرد .

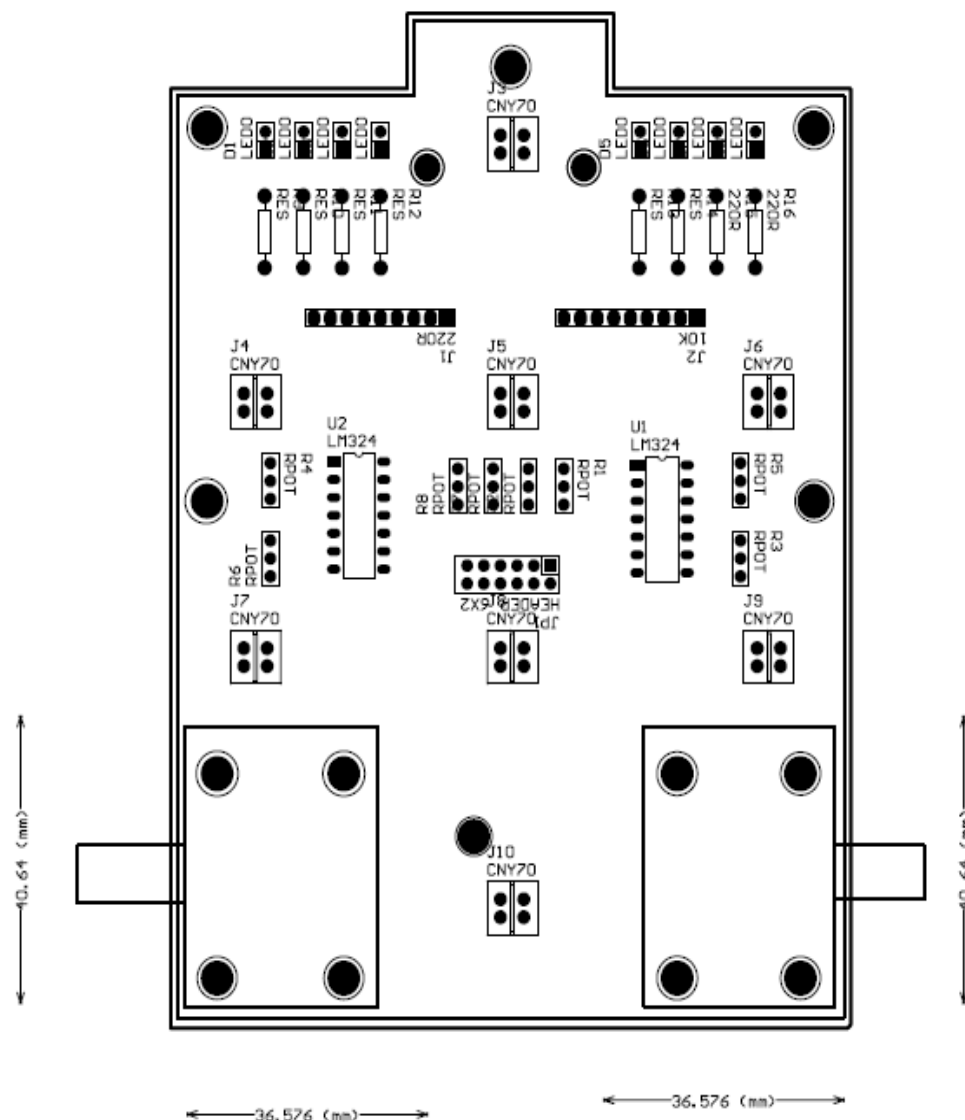
آی سی 741:

این آی سی شامل یک Op-amp می باشد مشخصات پایه های آن را در زیر می بینید :



نحوه چیدمان سنسورها :

تعداد سنسورهای مورد استفاده برای این روبات 8 عدد می باشد که با بررسی مسیر مسابقه سنسورها را مطابق شکل زیر چیدیم .



بجای استفاده از مقاومت های رنگی (شاخه آند و کاتد) برای 8 سنسور از مقاومت اهره ای با همان مقدار استفاده شده است .

برای هر سنسور از یک پتانسیومتر جداگانه استفاده شده که برای افزایش دقت از نوع مولتی ترن انتخاب شده اند .

درایور مورد استفاده :

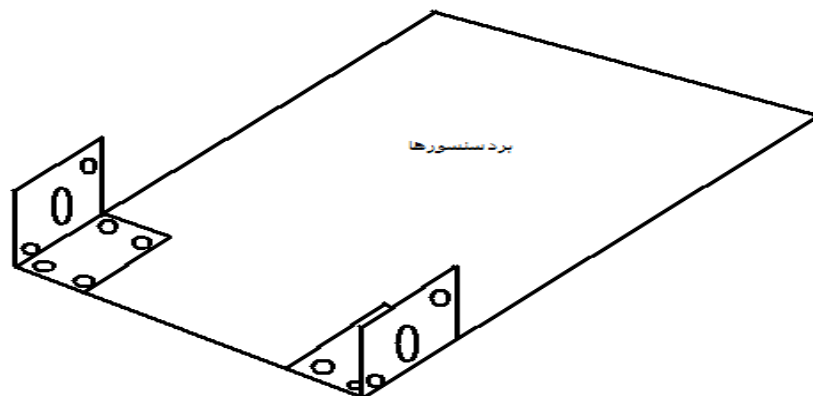
با بررسی آی سی های درایور موتور و جریان دهی آنها و نیز با توجه به موتورهای مورد استفاده درایور خود را از نوع L298 انتخاب کردیم .

برای جواب گرفتن از مدار درایور با توجه به مدار مورد استفاده شده در بالا برای این آی سی و نوشتن یک برنامه ساده که موتور را بچرخاند ، مدار درایور خود را تست کردیم .

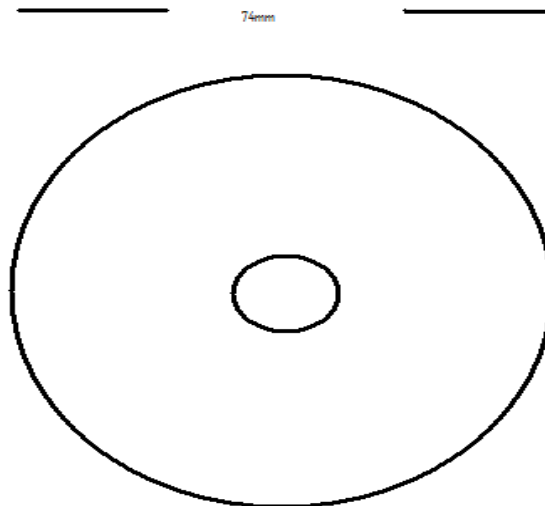
```
$regfile="m32def.dat
$crystal=8000000
Config PORTC=output
Dim a as byte ,b as byte , e as byte
Do
For a = 1 to 50
E = 128
For b = 1 to 40
Rotate e , left
Portc = e
Waitms 20
Next b
Next a
Loop
End
```

بدنه روبات :

به علت سادگی بدنه روبات مسیریاب و اینکه باید موتورها را طوری نصب کرد که بتواند روبات را راه ببرد همانگونه که از شکل بالا پیداست موتورها بر روی مدار درایور سنسورها و در عقب نصب شده است . از یک چرخ هرزگرد نیز در قسمت جلوی مدار درایور سنسورها استفاده شده است . از دو قطعه آلومینیوم برای سوار کردن موتورها بر روی برد سنسورها استفاده شده است . توجه شود که همانگونه از شکل زیر پیداست سوراخ های موتور به صورت کشیده می باشد که می توان فاصله سنسورها را با زمین در اندازه دلخواه تنظیم کرد .



از دو چرخ به قطر 74 میلی متر نیز برای چرخ های روبات استفاده شده است .



برنامه روبات :

/******

This program was produced by the

CodeWizardAVR V1.24.1c Standard

Automatic Program Generator

© Copyright 1998-2004 Pavel Haiduc, HP InfoTech s.r.l.

<http://www.hpinfotech.ro>

e-mail:office@hpinfotech.ro

Project :robot

Version :

Date : 2/1/2009

Author : 1180-0672-52BE-CA92

Company :

Comments:

Chip type : ATmega32

Program type : Application

Clock frequency : 8.000000 MHz

Memory model : Small

External SRAM size : 0

Data Stack size : 512

*****/

#include <mega32.h>

```

#include <delay.h>
// Declare your global variables here
char x,e=0,o=0,h=0,i=0,n=0,end=0;
void right(char a);
void leoright();
void rtoleft();
void left(char a);
void right_1(char a);
void left_1(char a);
void rightt();
void leftt();
void drive();
//void ch_360(char a);
//void ch_45r(char a);
//void ch_45l(char a);
void eslah();
void negative();
void left_n(char a);
void right_n(char a);
void eslah_n();
void ri_to_right();
void main(void)
{
// Declare your local variables here
// Input/Output Ports initialization
// Port A initialization
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T
PORTA=0x00;
DDRA=0x00;
// Port B initialization
// Func7=Out Func6=Out Func5=Out Func4=Out Func3=Out Func2=Out Func1=Out Func0=Out
// State7=0 State6=0 State5=0 State4=0 State3=0 State2=0 State1=0 State0=0
PORTB=0x00;
DDRB=0xFF;
// Port C initialization

```

```

// Func7=Out Func6=Out Func5=Out Func4=Out Func3=Out Func2=Out Func1=Out Func0=Out
// State7=0 State6=0 State5=0 State4=0 State3=0 State2=0 State1=0 State0=0
PORTC=0x00;
DDRC=0xFF;
// Port D initialization
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T
PORTD=0x00;
DDRD=0x00;
// Timer/Counter 0 initialization
// Clock source: System Clock
// Clock value: Timer 0 Stopped
// Mode: Normal top=FFh
// OC0 output: Disconnected
TCCR0=0x00;
TCNT0=0x00;
OCR0=0x00;
// Timer/Counter 1 initialization
// Clock source: System Clock
// Clock value: Timer 1 Stopped
// Mode: Normal top=FFFFh
// OC1A output: Discon.
// OC1B output: Discon.
// Noise Canceler: Off
// Input Capture on Falling Edge
TCCR1A=0x00;
TCCR1B=0x00;
TCNT1H=0x00;
TCNT1L=0x00;
ICR1H=0x00;
ICR1L=0x00;
OCR1AH=0x00;
OCR1AL=0x00;
OCR1BH=0x00;
OCR1BL=0x00;
// Timer/Counter 2 initialization

```

```

// Clock source: System Clock
// Clock value: Timer 2 Stopped
// Mode: Normal top=FFh
// OC2 output: Disconnected
ASSR=0x00;
TCCR2=0x00;
TCNT2=0x00;
OCR2=0x00;
// External Interrupt(s) initialization
// INT0: Off
// INT1: Off
// INT2: Off
MCUCR=0x00;
MCUCSR=0x00;
// Timer(s)/Counter(s) Interrupt(s) initialization
TIMSK=0x00;
// Analog Comparator initialization
// Analog Comparator: Off
// Analog Comparator Input Capture by Timer/Counter 1: Off
// Analog Comparator Output: Off
ACSR=0x80;
SFIOR=0x00;
while (1)
{
if(e!=4){
while(e<4){
if(PIND==0x3C)
drive();
if(PIND==0x3D)
drive();
if(PIND==0x3F)
drive();
if(PIND==0x2F){
left(28);
eslah();
}
}
}
}

```

```

if(PIND==0x3F)
drive();
if(PIND==0x1F){
right(28);
eslah();
}
if(PIND==0x3F)
drive();
if(e==4)
goto dd;
}
}
dd:
if(o==0){
do{
if(PIND==0x3C)
drive();
if(PIND==0x3D)
drive();
if((PIND==0x3E)|(PIND==0x7E)|(PIND==0x3F)|(PIND==0x7C)|(PIND==0x7D)|(PIND==0xBC)|(PIN
D==0xFC
)|(PIND==0xBE)|(PIND==0xFE)|(PIND==0xFD)|(PIND==0xEF)|(PIND==0xDF)|(PIND==0xBD)|(PIND
==0x7F
)|(PIND==0xBF))
drive();
if(PIND==0x7F){
drive(); drive();
if(PIND==0x7F)
left_1(4);
}
if(PIND==0x3D)
drive();
if(PIND==0x2D)
goto aa;
}while(1);
}

```

```

aa:
o=1;
do{
if(PIND==0x3F)
eslah();
if(PIND==0x2D)
left_1(45);
if(PIND==0x1D)
right_1(45);
if((PIND==0x3D)|(PIND==0x3C))
drive();
if((PIND==0x0C)|(PIND==0x00)|(PIND==0x30)|(PIND==0x10)|(PIND==0x20)|(PIND==0x1C)|(PIND
==0x2C
)|(PIND==0x38)|(PIND==0x34)|(PIND==0x08)|(PIND==0x04))
drive();
if(PIND==0x3F)
eslah();
if(PIND==0x1C){
drive();
if(PIND==0x1C)
goto d;
}
}while(1);
d:
if((PIND==0x3F)|(h>2))
eslah();
if((PIND==0x1C)|(PIND==0x35)){
drive();
h++;
}
ri_to_right();
};
}
/*****ri_to_right*****/
void ri_to_right(){
end++;

```

```

do{
x=(PIND)&(0b11000010);
if(x==2)
rightt();
if(x==0x42)
rightt();
if(x==0)
drive();
if(PIND==0x3C){
drive(); drive();drive();drive();drive();
if(n>=10)
rtoleft();
}
}while(1);
}

/*****right125*****/
void right(char a){
e++;
drive();
drive();
for(x=0;x<a;x++){
PORTB=0b00001000;
delay_ms(50);
PORTB=0b00000001;
delay_ms(50);
PORTB=0b00000010;
delay_ms(50);
PORTB=0b00000100;
delay_ms(50);
}
}

/*****LEFT125*****/
void left(char a){
e++;
drive();
if(PIND==0x3F)

```

```

goto hossein;
drive();
for(x=0;x<a;x++)
{
PORTC=0b00000001;
delay_ms(50);
PORTC=0b00001000;
delay_ms(50);
PORTC=0b00000100;
delay_ms(50);
PORTC=0b00000010;
delay_ms(50);
}
hossein:
}
/*****DRIVE*****/
void drive(){
PORTB=0b00001000;
PORTC=0b00000001;
delay_ms(50);
PORTB=0b00000001;
PORTC=0b00001000;
delay_ms(50);
PORTB=0b00000010;
PORTC=0b00000100;
delay_ms(50);
PORTB=0b00000100;
PORTC=0b00000010;
delay_ms(50);
}
/*****RIGHT_1*****/
void right_1(char a){
if(PIND==0x3D)
goto hossein1;
drive();
if(PIND==0x3D)

```

```

goto hossein1;
for(x=0;x<a;x++){
PORTB=0b00001000;
delay_ms(50);
PORTB=0b00000001;
delay_ms(50);
PORTB=0b00000010;
delay_ms(50);
PORTB=0b00000100;
delay_ms(50);
}
hossein1:
}
/*****LEFT_1*****/
void left_1(char a)
{
if(PIND==0x3D)
goto hossein2;
drive();
if(PIND==0x3D)
goto hossein2;
for(x=0;x<a;x++){
{
PORTC=0b00000001;
delay_ms(50);
PORTC=0b00001000;
delay_ms(50);
PORTC=0b00000100;
delay_ms(50);
PORTC=0b00000010;
delay_ms(50);
}
}
hossein2:
}
/*****eslah*****/
void eslah(){

```

```

do{
PORTC=0b00000001;
delay_ms(50);
PORTC=0b00001000;
delay_ms(50);
PORTC=0b00000100;
delay_ms(50);
PORTC=0b00000010;
delay_ms(50);
if(PIND==0x3C)
goto hossein3;
if(PIND==0x7F)
{
do{
PORTB=0b00001000;
delay_ms(50);
PORTB=0b00000001;
delay_ms(50);
PORTB=0b00000010;
delay_ms(50);
PORTB=0b00000100;
delay_ms(50);
if(PIND==0x3C)
goto hossein3;
}while(1);
}
}while(1);
hossein3:
}
/*****rtolft*****/
void rtolft(){
do{
x=(PIND)&(0b11000010);
if(x==2)
leftt();
if(x==0x42)

```

```

leftt();
if(x==0)
drive();
if(PIND==0x0D) /*END*/
PORTB=PORTC=0;
if(PIND==0x3C){
drive();drive();drive();drive();drive();drive();drive();
if(end==2)
rtolleft();
if(i>=15)
leoright();
}}while(1);
}
/*****leoright*****/
void leoright()
{
n=0;
do{
x=(PIND)&(0b11000010);
if(x==2)
rightt();
if(x==0x42)
rightt();
if(x==0){
drive();
}
if(PIND==0x3C){
drive();drive();drive();drive();drive();drive();
if(n>=10)
negative();
}}while(1);
}
/*****negative*****/
*****/
*****/
void negative(){

```

```

do{
if(PIND==0x3C)
drive();
if(PIND==0x3D)
drive();
if((PIND==0x0F)|(PIND==0x35)|(PIND==0x39)|(PIND==0x31)|(PIND==0x33)|(PIND==0x13)|(PIND
==0x23
)|(PIND==0x03)|(PIND==0x43))
drive();
if(PIND==0x3F)
right(1);
if(PIND==0xC3)
do{
if(PIND==~0x3C)
drive();
if(PIND==~0x3D)
drive();
if(PIND==~0x2D)
left_n(45);
if(PIND==~0x1D)
right_n(45);
if(PIND==~0x3F)
eslah_n();
if((PIND==~0x0F)|(PIND==~0x35)|(PIND==~0x39)|(PIND==~0x31)|(PIND==~0x33)|(PIND==~0x13
)|(PIND
==~0x23)|(PIND==~0x03)|(PIND==~0x43))
drive();
if(PIND==0x3C)
goto gg;
}while(1);
}while(1);
gg:
do{
n=0;
ri_to_right();
}while(1);

```

```

}

/*****rightt*****/

void rightt(){
x=(PIND)&(0b11000010);
PORTB=0b00001000;
delay_ms(50);
PORTB=0b00000001;
delay_ms(50);
PORTB=0b00000010;
delay_ms(50);
PORTB=0b00000100;
delay_ms(50);
x=(PIND)&(0b11000010);
//if(x==0)
n++;
//drive();
}

/*****LEFTt*****/

void leftt(){
x=(PIND)&(0b11000010);
PORTC=0b00000001;
delay_ms(50);
PORTC=0b00001000;
delay_ms(50);
PORTC=0b00000100;
delay_ms(50);
PORTC=0b00000010;
delay_ms(50);
PORTC=0b00000010;
delay_ms(50);
x=(PIND)&(0b11000010);
i++;
}

/*****RIGHT_n*****/

void right_n(char a){
if(PIND==~0x3D)
goto hossein1;
drive();

```

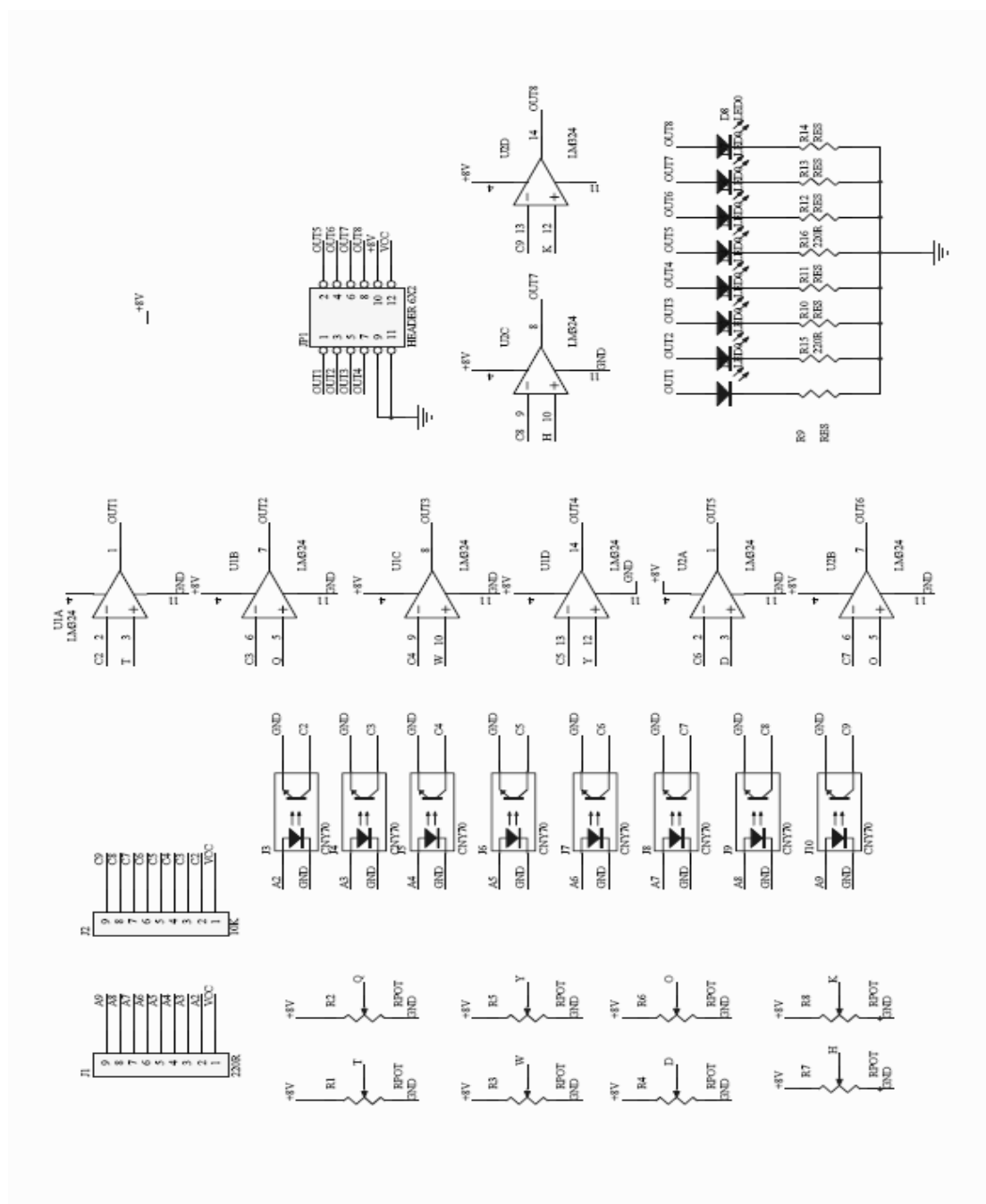
```

if(PIND==~0x3D)
goto hossein1;
for(x=0;x<a;x++){
PORTB=0b00001000;
delay_ms(50);
PORTB=0b00000001;
delay_ms(50);
PORTB=0b00000010;
delay_ms(50);
PORTB=0b00000100;
delay_ms(50);
}
hossein1:
}
/*****LEFT_n*****/
void left_n(char a)
{
if(PIND==~0x3D)
goto hossein2;
drive();
if(PIND==~0x3D)
goto hossein2;
for(x=0;x<a;x++)
{
PORTC=0b00000001;
delay_ms(50);
PORTC=0b00001000;
delay_ms(50);
PORTC=0b00000100;
delay_ms(50);
PORTC=0b00000010;
delay_ms(50);
}
hossein2:
}
/*****eslah_n*****/

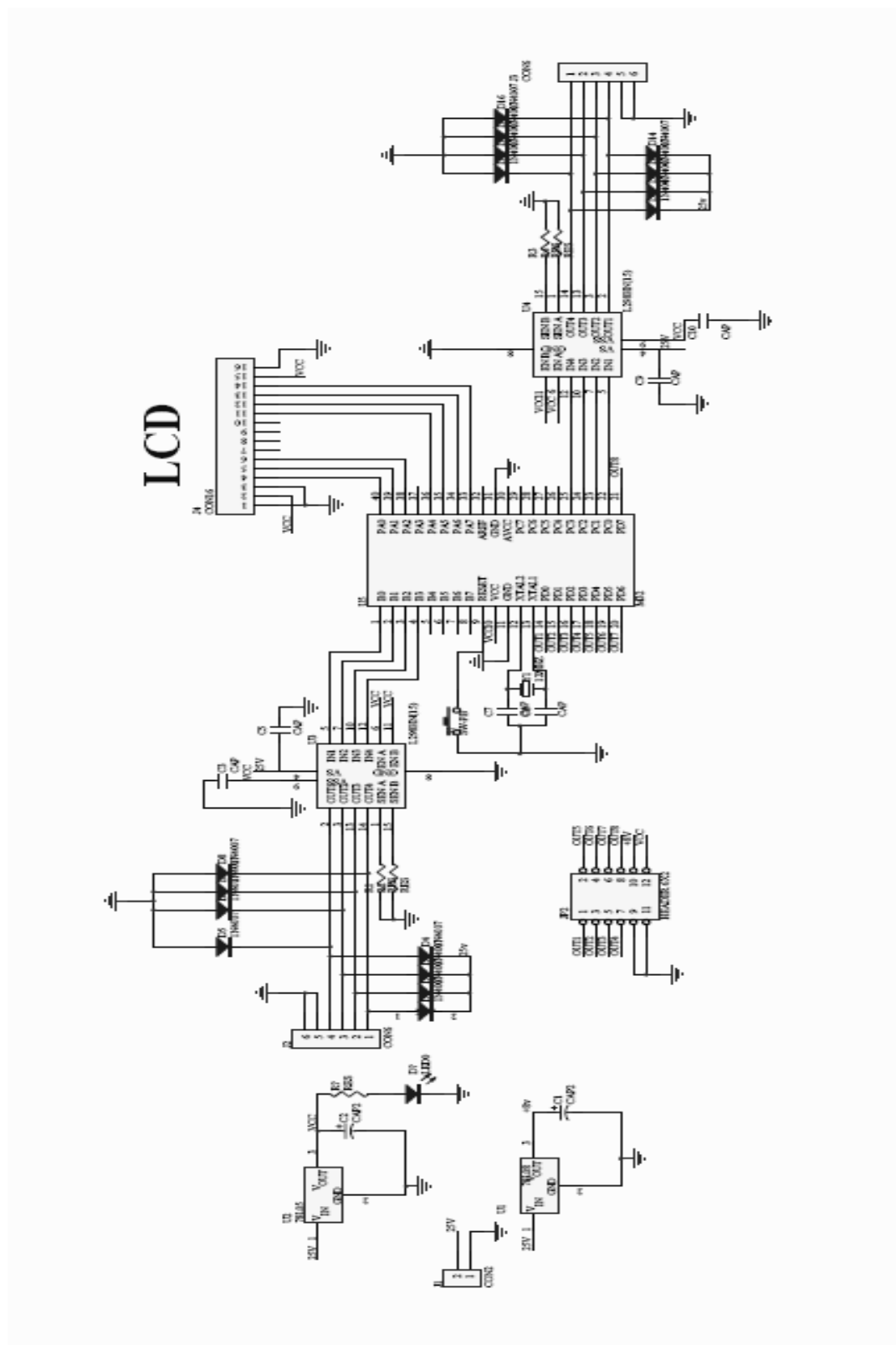
```

```
void eslah_n(){
do{
PORTC=0b00000001;
delay_ms(50);
PORTC=0b00001000;
delay_ms(50);
PORTC=0b00000100;
delay_ms(50);
PORTC=0b00000010;
delay_ms(50);
if(PIND==~0x3C)
goto hossein3;
if(PIND==~0x7F)
{
do{
PORTB=0b00001000;
delay_ms(50);
PORTB=0b00000001;
delay_ms(50);
PORTB=0b00000010;
delay_ms(50);
PORTB=0b00000100;
delay_ms(50);
if(PIND==~0x3C)
goto hossein3;
}while(1);
}
}while(1);
hossein3:
}
```

نقشه مدار درایور سنسورها :



برد میکرو و درایور :



شكل مدارات :

