

صلى الله عليه وسلم

پیشگفتار

روبات یک سیستم الکترومکانیکی است که حداقل یکی از رفتارهای انسان را تقلید نماید. براین اساس روباتهای مختلفی طراحی و ساخته شده است مانند روباتهای فوتبالیست، جنگجو، مین یاب، امدادگر، انسان نما، صخره نورد و... . موضوع هدایت روباتها از طریق بینایی در سالهای اخیر در مراکز دانشگاهی مورد تحقیق و مطالعه و اجرا قرار گرفته است ولی هنوز به طور وسیعی به بخش صنایع به خصوص در کشور ما انتقال پیدا نکرده است. پیشرفت در زمینه قدرت محاسباتی کامپیوتر و کاهش هزینه های قطعات در پنج سال اخیر به روباتهای هدایت شونده از طریق بینایی اجازه ظهور جدی را داده است. درکل دو روش کلی برای نگاشت از تصاویر ورودی به حرکت روبات وجود دارد. یکی سینماتیک محاسباتی و دیگری visual servoing می باشد. سینماتیک محاسباتی به هر الگوریتمی که انتقال بین صفحه تصویر و دنیای حقیقی را انجام دهد اطلاق می شود که شامل کالیبراسیون و روشهای مبتنی بر شبکه های عصبی می باشد. در سینماتیک محاسباتی موقعیت هدف (target) در مختصات مرجع (مختصات دنیای واقعی) محاسبه می شود و سپس یک بردار حرکت برای رسیدن روبات به آن موقعیت تولید می شود. در visual servoing خطای بین وضعیت تصویر گرفته شده و آنچه که مورد دلخواه است توسط یک قانون کنترلی به سمت صفر سوق داده می شود. روش visual servoing به دو حالت تقسیم می شود. حالت اول feature-based و حالت دوم position-based می باشد.

در برنامه شبیه سازی شده برای کنترل روبات توسط بینائی از روش visual servoing feature based استفاده شده است.

در این پروژه روباتی طراحی شده است که با استفاده از دوربین بی سیمی که در بالای آن قرار گرفته است تصاویر محیط اطراف را جمع آوری کرده و با استفاده از نرم افزار matlab 7.6 تصاویر گرفته شده را پردازش میکند و در ادامه سیگنال کنترلی را برای هدایت روبات در جهت مربوطه برای دنبال کردن شیء مورد نظر به صورت بی سیم به کنترل کننده موتورها ارسال میکند و در نتیجه روبات به سمت شیء مورد نظر به حرکت درمی آید.

محمد علی لهراسبی lohrasbi.ali@gmail.com

محسن انصاری منزه mehi_63@hotmail.com

فهرست مطالب

۱ مقدمه

فصل اول

۴ 1-1- آشنایی با پردازش تصاویر

۴ بینایی رایانه ای

۵ سامانه های بینایی رایانه ای

۶ استخراج ویژگی

۷ 2-1- پردازش تصویر

۷ پردازش سطح پایین

۷ پردازش سطح میانی

۷ پردازش سطح بالا

۸ 3-1- تصویر دیجیتالی چیست ؟

۹ 4-1- انواع تصویر

۹	تصاویر اندیس گذاری شده (indexed)
۱۰	تصاویر با شدت رنگ (intensity)
۱۱	تصاویر RGB

فصل دوم

۱۵	1-2- سخت افزار پروژه
۱۶	2-2- دوربین (wireless)
۱۷	3-2- گیرنده فرستنده (RF)
۱۷	4-2- درایور موتورهای DC
۱۷	5-2- میکرو کنترلر AVR
۲۰	6-2- بدنه اصلی روبات
۲۲	7-2- دیاگرام بلوکی کل سیستم
۲۲	8-2- خلاصه عملکرد سخت افزار
۲۳	9-2- نکات فنی قابل توجه

فصل سوم

- ۲۵ ----- شرح کلی عملکرد روبات
- ۲۶ ----- 1-3- خلاصه عملکرد نرم افزاری روبات
- ۲۷ ----- 2-3- مشخصات رایانه مورد نیاز برای انجام عمل پردازش
- ۲۸ ----- 3-3- نرم افزار رایانه
- ۲۹ ----- 4-3- محیط برنامه نویسی
- ۳۰ ----- 5-3- الگوریتم و برنامه اصلی
- ۳۱ ----- تعیین هدف و عملکرد
- ۳۲ ----- تهیه الگوریتم برای هدف مورد نظر
- ۳۴ ----- تهیه نرم افزار از روی فلوچارت
- ۳۵ ----- 6-3- شرح تفصیلی برنامه نوشته شده در محیط matlab

فصل چهارم

- 1-4- توضیحات تکمیلی
- ۴۱ ----- سیستم بینایی ماشین

۴۳	----- سخت افزار دوربین
۴۵	----- 2-4- طریقه دستیابی به تصویر و تنظیمات آن در نرم افزار matlab
۴۵	----- نصب سخت افزار مورد نیاز
۴۶	----- شناسایی کارت کیچر و بدست آوردن مشخصات آن در matlab
۵۰	----- ساختن یک video input object
۵۲	----- نمایش preview
۵۳	----- 3-4- تهیه فریم در یک ماتریس
۵۳	----- Getsnapshot
۵۳	----- Getdata
۵۴	----- 4-4- ارسال و دریافت داده
۵۵	----- 5-4- مدولاسیون FSK
۵۶	----- بلوک دیاگرام فرستنده و گیرنده
۵۸	----- 6-4- ارتباط سریال
۵۸	----- ارتباط سریال USART در AVR

۵۹ ----- 7-4- اتصال AVR به RS232

۵۹ ----- 8-4- ارتباط با پورت سریال در محیط MATLAB

فصل پنجم

۶۲ ----- 1-5- نتیجه گیری کلی و کارهای آینده

۶۳ ----- 2-5- نقاط قوت پروژه

۶۴ ----- 3-5- نقاط ضعف پروژه

۶۵ ----- 4-5- بهینه سازی و ارتقاء روبات

5-5- ضمائم و پیوستها

۶۷ ----- شماتیک مدار طراحی شده برای روبات

۶۸ ----- PCB مدار طراحی شده

۶۹ ----- نمائی از روبات

۷۰ ----- تصویر چند نمونه از روباتهای هوشمند

۷۲ ----- برنامه نوشته شده برای انجام عمل پردازش تحت محیط matlab

۷۵ ----- برنامه جهت پروگرام کردن میکرو کنترلر AVR

۸۲ ----- دیتا شیت قطعات مورد استفاده

۱۱۲ ----- منابع

مقدمه:

از حدود دو قرن پیش که انقلاب صنعتی انگلستان به وقوع پیوست تا به امروز صنعت گامهایی بزرگ برداشته است که همه در جهت راحتی انسان بوده ولی این پیشرفتهای با مشکلاتی نیز همراه بوده اند.

محصولات ساخت دست بشر دارای یک کیفیت استاندارد و برابر و شکل کاملاً همگون نبود و این امر موجب به وجود آمدن محصولات متفاوت از نظر کیفیت میشد که چندان مطلوب نبود. از طرف دیگر به علت محدودیت توانائی و قدرت انسان ، وضعیت تولید محصول از نظر کمیت نیز جلب نظر نمیکرد و این امر باعث میشد که قیمت تمام شده برای محصول بالا باشد و لذا توانائی خرید این محصولات برای قشر خاصی بود از طرف دیگر وجود خطاهای بسیار از جمله خطاهای انسانی باعث پایین آمدن بازده محصول میشد. همه این عوامل و عوامل دیگر نیاز به دستگاه و ماشینی را مینمود که بتوان آنرا جایگزین انسان کرد و یا حداقل به عنوان کمک در کنار دست انسانها به کار پردازد و این موضوع زمره های اولیه ساخت روباتها بود. ورود رشته های مرتبط با ساخت روبات و ایجاد ارتباط بین آنها باعث شد که این رشته و تکنولوژی وارد دانشگاهها، مراکز تحقیقاتی و پژوهشی شود و در چند سال اخیر هم به این امر کمک بیشتری شده است.

در این مراکز و دانشگاهها تحقیقات فراوانی انجام میشود و روباتهایی ظریف ، زیبا و در عین حال با قابلیت بالا ساخته شد و این امر تا آنجا پیشرفت که باعث به وجود آمدن مسابقات جهانی روباتها شد. روباتها به چند دسته طبقه بندی میشوند یک دسته روباتهایی هستند که صرفاً دارای قسمت مکانیکی بوده و از انجام عملیات پردازش و کنترل عاجز هستند. دسته

دیگری از روباتها هستند که از پیچیدگی و تکنیک بالاتری برخوردار هستند. اینها علاوه بر قسمتهای مکانیکی دارای واحد پردازش و کنترل نیز میباشند و به همین دلیل از تواناییهای بالاتری نسبت به دسته اول برخوردارند در این پایان نامه به یکی از انواع روباتهای دسته دوم پرداخته ایم. در این سیستم از پردازش تصویر به عنوان ابزار هوشمند استفاده شده است. پردازش تصویر دیجیتال دانش جدیدی است که سابقه آن به پس از اختراع رایانه های دیجیتال باز میگردد. با این حال این علم نوپا در چند دهه اخیر از هر دو جنبه نظری و عملی پیشرفتهای چشمگیری داشته است. سرعت این پیشرفت به اندازه ای بوده است که هم اکنون و پس از این مدت نسبتاً کوتاه ، به راحتی میتوان رد پای پردازش تصویر دیجیتال را در بسیاری از علوم و صنایع مشاهده نمود. بعضی از این کاربردها آنچنان به پردازش تصویر وابسته هستند که بدون آن اساساً قابل استفاده نمیباشند. یک نمونه از چنین کاربردهایی تشخیص تومور با استفاده از تصاویر CT-SCAN (تومور نگاری رایانه ای) است.

وسعت کارهای پردازش تصویر در طی سالیان اخیر ، ضرورت فراگیری جامع چنین مباحثی را برای دانشجویان رشته های برق و کامپیوتر به طور جدی مطرح نموده است. با توجه به نیاز پروژه های پردازش تصویر به سخت افزاری جامع که امکان تحرک دوربین را فراهم آورد سیستم حاضر با داشتن سخت افزاری مناسب نمونه ای نسبتاً کامل از کلیات کارهایی که در این زمینه انجام میشود میباشد.

این پیشرفتهای توجه انسان آگاه و صاحب خرد را به این موضوع معطوف میکند که خالق قادر و خلاق انسان را با این ظرافت آفریده است که قادر به اختیارگرفتن چنین تکنولوژی میباشد چنین خالق به شهادت هر عقل سلیم شایسته ستایش ، مباحثات و شکرگذاریست.

فصل اول

آشنایی با پردازش تصاویر

پردازش تصاویر امروزه بشتر به موضوع پردازش تصویر دیجیتال گفته میشود که شاخه ای از دانش رایانه است که با پردازش سیگنال دیجیتال که نماینده تصاویر برداشته شده از دوربین دیجیتال یا پویش شده توسط پویشگر هستند سرو کار دارد پردازش تصاویر دارای دو شاخه عمده بهبود تصاویر و بینایی ماشین بهبود تصاویر در بر گیرنده روشهایی چون استفاده از فیلتر محو کننده و افزایش تضاد براد بهتر کردن کیفیت دیداری تصاویر و اطمینان از نمایش درست آنها در محیط مقصد (مانند چاپگر یا نمایشگر رایانه) است. در حالیکه بینایی ماشین به روشهای می پردازد که به کمک آنها میتوان معنی و محتوای تصاویر را درک کرد تا از آنها در کارهایی چون رباتیک و محور تصاویر استفاده شود.

بینایی رایانه ای

بینایی رایانه ای (computer vision) یکی از شاخه های مدرن و پر تنوع هوش مصنوعی است که با ترکیب روشهای مربوط به پردازش تصاویر (image parocessing) و ابزارهای تعلم ماشینی، رایانه ها را به بیتایی اشیا ، مناظر ، و درک هوشمند خصوصیات گوناگون آنها توانا می گرداند.

وظایف اصلی در بینایی رایانه ای

تشخیص شی : تشخیص حضور و یا حالت شی در یک تصویر بعنوان مثال :

- جستجو برای تصاویر دیجیتال براساس محتوایشان
- شناسایی صورت انسان و موقعیت آنها در عکس ها

• تعیین حالت سه بعدی انسانها و اندامهایشان

پیگیری : پیگیری اشیای شناخته شده در میان تعدادی تصویر پشت سرهم. به عنوان مثال:

پیگیری یک شخص هنگامی که در یک مرکز خرید می رود.

تفسیر منظره : ساختن یک مدل از یک تصویر / تصویر متحرک. به عنوان مثال: ساختن یک

مدل از ناحیه پیرامونی به کمک تصاویری که از دوربین نصب شده بر روی یک روبات گرفته میشوند.

خود مکان یابی: مشخص کردن مکان و حرکت خود دزربین به عنوان عضو بیانی رایانه. به

عنوان مثال: مسیریابی یک روبات درون یک موزه.

سامانه های بینایی رایانه ای

یک سامانه نوعی بینایی رایانه ای را می توان به زیر سامانه های زیر تقسیم کرد:

تصویر برداری

تصویر با دنباله تصاویر با یک سامانه تصویربرداری (دوربین ، رادار ، لیدار ، سامانه توموگرافی)

برداشته می شود. معمولا سامانه تصویربرداری باید پیش از استفاده تنظیم شود.

پیش پرداز

در گام پیش پرداز، تصویر در معرض اعمال "سطح پایین" قرار می گیرد. هدف این گام کاهش نوفه (کاهش نویز-جدا کردن سیگنال از نویز) و کم کردن مقدار کلی داده هاست. این کار نوعاً با به کار گیری روش های گوناگون پردازش تصویر (دیجیتال) انجام می شود. مانند:

زیر نمونه گیری تصویر، اعمال فیلترهای دیجیتال، پیچش ها، همبستگی ها با فیلتر های خطی لغزش نابسته، عملکرد سوئل، محاسبه گرادیان X و Y (و احتمال گردیان زمانی)، تقطیع تصویر، آستانه گیری پیکسلی، انجام یک ویژه تبدیل بر تصویر، تبدیل فوریه، انجام تخمین حرکت برای ناحیه های محلی تصویر که به نام تخمین شارش نوری هم شناخته می شود، تخمین ناهمسانی در تصاویر بر جسته بینی و تحلیل چند دقتی.

استخراج ویژگی

هدف از استخراج ویژگی کاهش دادن بیشتر داده ها به مجموعه ای از ویژگی هاست، که باید به اغتشاشاتی چون شرایط نورپردازی، موقعیت دوربین.

- انجام آشکار سازی لبه
- استخراج ویژگی های گوشه ای
- استخراج تصاویر چرخش از نقشه های ژرفا
- بدست آوردن خطوط تراز و احتمالاً گذر از صفر های خمش

پردازش تصویر

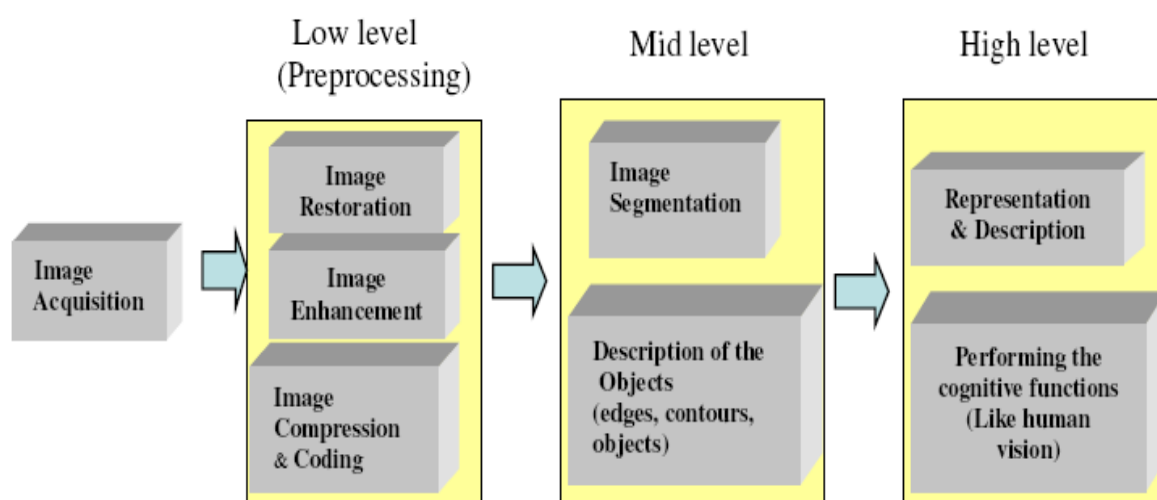
مرز مشخص بین پردازش تصویر از یک طرف و بینایی ماشین (computer vision) از طرف دیگر نمی‌توان مشخص کرد با این حال میتوان سه نوع پردازش سطح پایین (low level) سطح میانی (mid level) و سطح بالا (high level) تشخیص داد.

❖ پردازش سطح پایین شامل پردازشهای ابتدایی مانند پیش پردازش‌ها برای حذف نویز و بهبود کنترلست (contrast) و فیلتر کردن تصویر است مشخصه این نوع پردازش این است که ورودی و خروجی آن تصویر هستند.

❖ پردازش سطح میانی شامل بخش بندی تصویر (image segmentation) به منظور تقسیم آن به نواحی و اشیاء مختلف ، توصیف اشیاء به فریمی که برای پردازش کامپیوتر مناسب باشند و طبقه بندی یا تشخیص اشیاء مختلف است. ویژگی این پردازش این است که ورودی آن معمولاً تصویر و خروجی آن صفاتی از اشیاء تصویر مانند لبه‌ها، کانتورها و تشخیص اشیاء است.

❖ پردازش سطح بالا شامل فهمیدن روابط بین اشیاء تشخیص داده شده (making sense) ، استنباط و تفسیر صحنه و انجام تفسیر و تشخیص‌هایی است که سیستم بینایی انسان انجام میدهد بسیاری از پردازشهای سطح بالا در حیطه بینایی ماشین قرار میگیرد.

به طور مثال یک سیستم تشخیص اتوماتیک متن را در نظر بگیرید. برای مشخص کردن ناحیه حاوی متن ، پیش پردازش تصویر حاصل ، استخراج کاراکترهای مختلف و تشخیص آن کاراکترها در حیطه بحث پردازش تصویر قرار دارند استنباط مفهوم و محتوای متن مورد نظر در حیطه بینایی ماشین یا آنالیز تصویر قرار میگیرد.



تصویر دیجیتالی چیست؟

یک تصویر را میتوان توسط تابع دو بعدی $f(x,y)$ که در آن x و y را مختصات مکانی و مقدار f در هر نقطه را شدت روشنایی تصویر در آن نقطه می نامند اصطلاح سطح خاکستری نیز به شدت روشنایی تصاویر مونو کروم اطلاق میشود.

تصاویر رنگی نیز از تعدادی تصویر دو بعدی تشکیل میشود. زمانیکه مقادیر x و y و مقدار f (x,y) با مقادیر گسسته و محدود بیان شوند ، تصویر را یک تصویر دیجیتالی می نامند. دیجیتال کردن مقادیر x و y را SAMPLING و دیجیتال کردن مقدار $f(x,y)$ را quantization گویند.

یک تصویر دیجیتالی از تعدادی عناصر محدود با مقدار و موقعیت مختلف تشکیل شده است. این عناصر ، عناصر تصویر (picture element) یا پیکسل (pixel) نامیده میشوند. پردازش تصاویر دیجیتالی به معنی اعمال پردازشهای مختلف روی یک تصویر دیجیتالی با استفاده از کامپیوتر دیجیتالی است.

انواع تصوبر

تصاویر در matlab شامل یک ماتریس داده میباشد و معمولاً به ماتریس جعبه رنگ وابسته هستند. سه نوع ماتریس داده برای تصاویر وجود دارد که هر کدام به صورتهای مختلفی تفسیر میشوند.

- تصاویر اندیس گذاری شده (indexed)

- تصاویر با شدت رنگ (intensity)

- تصاویر RGB

تصاویر اندیس گذاری شده :

در واقع یک تصویر اندیس گذاری شده آرایه دو بعدی است که در هر خانه آن شماره رنگ پیکسل مورد نظر ذخیره میشود. یک تصویر اندیس گذاری شده به یک جعبه رنگ نیاز دارد و داده های تصویر را به صورت اندیس هایی در ماتریس جعبه رنگ تفسیر میکند. ماتریس جعبه رنگ یک جعبه رنگ استاندارد است که اندیس های آن $3 * M$ بوده و شامل مقادیر قابل قبول RGB است. با ارائه داده ای $X(I,J)$ که مربوط به تصویر است و آرایه جعبه رنگ

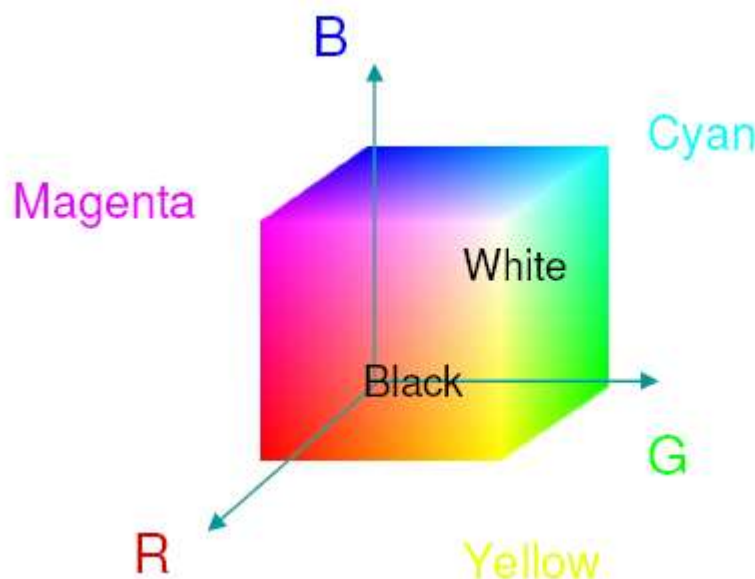
CAMP، رنگ هر پیکسل از تصویر با $\text{camp}(x(i,j),:)$ مشخص میشود که این رابطه نشانگر این است که مقادیر X ، اعداد صحیحی در بازه $[1 \text{ } \text{lenth}(\text{camp})]$ است. این تصویر را میتوان با استفاده از دستور زیر نمایش داد :

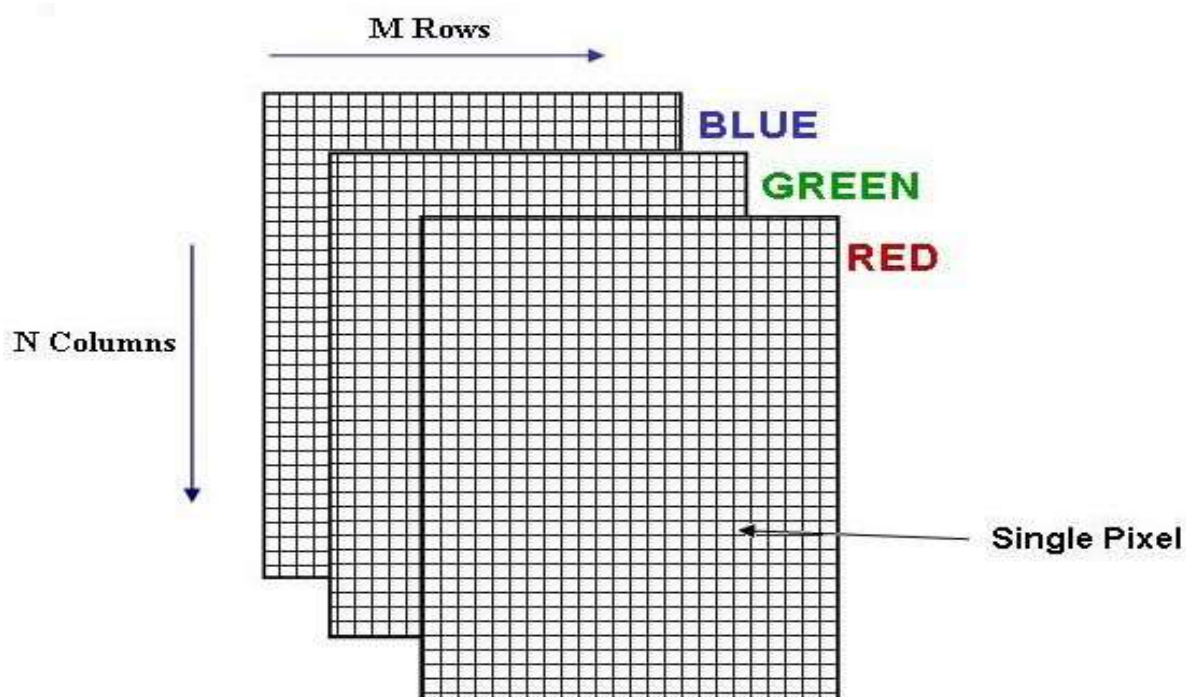
`Image(x);`

`Colormap(cmap)`

تصاویر RGB

تصاویر رنگی از ترکیب چند تصویر دو بعدی تشکیل شده اند بطور مثال در سیستم رنگی RGB هر تصویر از سه مولفه تصویر قرمز ، سبز و آبی تشکیل شده است.





تصاویر با شدت رنگ

همانطور که مشاهده شد در حات عادی ۳ درجه رنگی از ۳ رنگ اصلی آبی و سبز و قرمز برای هر نقطه وجود دارد که با کم و زیاد شدن شدت هر مولفه اصلی رنگ حاصل تغییر میکند. اما در پردازش تصویر این فرمت اصلا مناسب نیست، علت این است که در پردازش تصویر معمولا نیاز به تشخیص یک محدوده رنگی داریم ، مثلا رنگ نارنجی، در این حالت یافتن این محدوده رنگی با ترکیب ۳ رنگ اصلی ممکن نیست یا اینکه کاری بسیار دشوار و وقت گیر میباشد. سیستم با روش دیگری که در این زمینه وجود دارد بر اساس قاعده زیر است.

برای هر رنگ سه مشخصه وجود دارد :

۱- شدت رنگ (saturation) هر رنگ میتواند پر رنگ یا کم رنگ

باشد اما ماهیت ذاتی آن ثابت است و فقط کم رنگ تر و پر رنگ تر شده است.

۲- نام رنگ (HUE) که مشخص کننده نوع رنگ میباشد به عنوان

مثال آبی کم رنگ یا قرمز متمایل به نارنجی یا ...

۳- روشنایی یا تیرگی رنگ (intensity) این مولفه شدت رنگ را

تعیین میکند که با نور تابیده شده به آن تغییر میکند بعنوان مثال

اگر نور تابیده شده کم باشد ، رنگ روبه تیرگی رفته و تقریباً

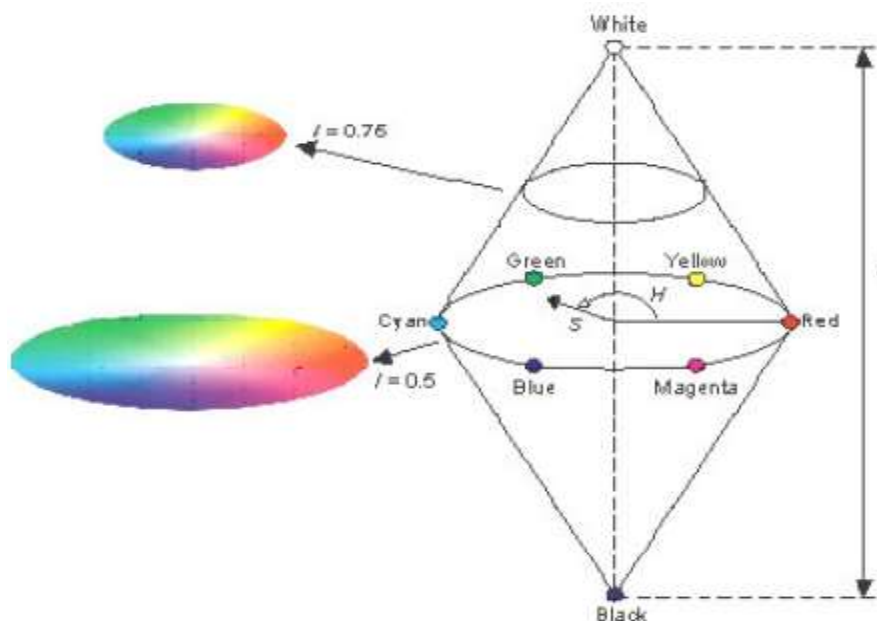
متمایل به سیاه می شود بنا بر این حوضه جدیدی از رنگ که الهام

گرفته از چشم انسان می باشد قابل تعریف است.

به این حوضه رنگی اصطلاحاً HIS گفته می شود که مخفف ۳ کلمه بالاست که در شکل زیر

قابل مشاهده

است :



H : بین ۰ تا ۳۶۰ قابل تغییر است یعنی از قرمز تا سبز ۱۲۰ درجه و از سبز تا آبی ۱۲۰ درجه و از آبی تا قرمز ۱۲۰ درجه در خلاف جهت عقربه های ساعت.

S : بین ۰ تا ۱۰۰ قابل تغییر است که از کم رنگ (0) تا پر رنگ (100).

I : از ۰ تا ۱۰۰ تغییر میکند یعنی از تاریک (0) تا روشن (100).

چند نمونه از رنگها در ۲ حوزه مذکور :

رنگ	RGB مقادیر	HIS
سیاه	(255,255,255)	(0,0,0)
سفید	(255,255,255)	(0,0,100)
قرمز	(255,0,0)	(0,100,100)
سبز	(0,255,0)	(120,100,100)
آبی	(0,0,255)	(240,100,100)
قهوه ای	(64,128,128)	(180,50,50)

فصل دوم

سخت افزار پروژه

در این فصل به بحث درمورد سخت افزار این پروژه پرداخته شده است. این فصل شامل چهار بخش میباشد که برای آشنایی شما توضیح مختصری در مورد هر بخش میدهیم. در بخش اول با عنوان معرفی روبات ، سعی کرده ایم که به کلیات روبات بپردازیم این بخش در دو قسمت ارائه شده است. به این صورت که ابتدا به معرفی قسمتهای مختلف روبات چه از نظر سخت افزاری و چه از نظر نرم افزاری پرداخته ایم. در ادامه سعی شده است که با توضیح مختصری ، عملکرد کلی مدار برای شما آشکار شود تا با به دست آوردن دید کلی از عملکرد مدار در فهم قسمتهای بعدی کمک قابل توجهی شود. در بخش بعدی به عنوان بدنه روبات به معرفی فیزیکی روبات پرداخته ایم و سپس به نکات ریزی اشاره کرده ایم. در قسمت بعدی این بخش که به نکات فنی اختصاص دارد سعی بر آن شده است که به نکات ریزولی قابل توجه و تأمل پرداخته شود که این نکات فقط در اثر کار عملی و تجربی به دست میآید و در کار تحقیقاتی و تئوری خود را نشان نمیدهند. در بخش سوم این پروژه به عنوان تشریح کار مدار، سعی بر این شده است که دیدی کلی از نحوه عملکرد روبات ارائه داده شود. در نهایت ، بخش انتهائی این فصل که به تشریح مدار اصلی روبات اختصاص دارد و با عنوان شرح مدار اصلی روبات معرفی گردیده است. در ادامه تلاش کرده ایم که توضیح نسبتاً کاملی از نحوه عملکرد مدار از جمله ارسال اطلاعات و دریافت دستورات لازم جهت راه اندازی موتورها داده شود.

معرفی روبات :

این روبات شامل دو بخش سخت افزاری و نرم افزاری میباشد که هر یک شامل قسمتهای جداگانه ای است.

بخش سخت افزاری :

۱- دوربین

۲- (RF) فرستنده و گیرنده

۳- مدار شارژ

۴- درایور موتور DC (میکرو کنترلر و آی سی های مربوطه)

۵- باتریها

۶- موتورهای DC

۷- بدنه اصلی روبات

که در ذیل مختصرا به آنها میپردازیم و توضیحات کاملتر در بخشهای دیگر ارائه میگردد.

۱- دوربین:

از نوع انالوگ و بی سیم (WIRELESS) میباشد که اطلاعات تصویر (IMAGE) را بصورت انالوگ و روی باند UHF ارسال میکند. این دوربین دارای یک گیرنده میباشد که سیگنالهای تصویر را به صورت انالوگ (AV) در اختیار ما قرار میدهد

۲- فرستنده- گیرنده (RF)

وظیفه این دستگاه ارسال و دریافت اطلاعات به صورت دیجیتال از طریق پورت سریال (SERIAL PORT) به گیرنده RF بر روی بدنه روبات میباشد. این اطلاعات شامل دستورالعملهای اجرائی برای موتورهای میباشد. مدار RF از نوع مدولاسیون فرکانس یا FSK (Frequency Shift Keying) میباشد.

۳- درایور موتورهای DC (DC MOTOR DRIVERS)

شامل کنترلر از نوع AVR و سری ATMEGA 16 و ای سی راه انداز موتور DC با نام LM298 میباشد.

۴- AVR

قسمت اصلی در مدار قسمت متحرک روبات میکروکنترلر از نوع AVR به شماره ATMEGA16 میباشد که توضیح مختصری در مورد آن میدهیم.

خصوصیات ATMEGA16 :

- دارای ۱۳۱ دستور قدرتمند است که اکثر آنها در یک سیکل ساعت انجام میشوند.

- از معماری AVR RISC استفاده میکند.

- کارایی بالا و توان مصرفی کم.

• حافظه

- 16K بایت حافظه فلش قابل برنامه ریزی.
- مجهز به قسمت Boot Loader.
- ۵۱۲ بایت حافظه EEPROM داخلی.
- 1K بایت حافظه SRAM داخلی.
- قفل قابل برنامه ریزی برای امنیت نرم افزار.

• امکانات جانبی

- ارتباط JTAG شامل اسکن کردن امکانات جانبی، پشتیبانی از دیباگ کردن تراشه و برنامه ریزی حافظه های EEPROM، FLASH، فیزها و بیت‌های قفل.
- دو تایمر/کانتر ۸ بیتی با تقسیم کننده فرکانسی مجزا و دارای مد Compare.
- یک تایمر/کانتر ۱۶ بیتی با تقسیم کننده فرکانسی مجزا و دارای مدهای Capture و Compare.
- دارای RTC (real-time Clock).
- چهار کانال PWM.
- ۸ کانال ADC ۱۰ بیتی.
- ۸ کانال تک پایه.
- ۷ کانال تفاضلی در بسته بندی TQFP.
- ۲ کانال تفاضلی با بهره قابل تنظیم 1x ، 10x و 200x.

- ارتباط سریال دو سیمه (Two Wire).
- USART سریال قابل برنامه ریزی.
- ارتباط سریال SPI به صورت Slave/Master
- Watchdog قابل برنامه ریزی با اسیلاتور مجزای داخلی.
- مقایسه کننده کننده آنالوگ داخلی.
- مقایسه کننده آنالوگ داخلی.
- دارای شش مد Idle:sleep , ADC noise Reduction , Power-save ,
Power-down , standby و Extended standby

• ولتاژهای عملیاتی

- 2.7v تا 5.7v برای Atmega16L
- 4.5v تا 5.5v برای Atmega16

• فرکانس کاری

- 0MHz تا 8MHz برای Atmega16L
- 0MHz تا 16MHz برای Atmega16

• خطوط I/O و انواع بسته بندی

- ۳۲ خط ورودی/خروجی قابل برنامه ریزی
- ۴۰ پایه در نوع PDIP و ۴۴ پایه در انواع TQFP و MLF.

۵- باتریها:

این قسمت از دو باتری شارژی با ولتاژهای گوناگون ساخته شده است. در یکی از آنها شش عدد باتری ۱,۵ ولتی با هم سری شده و باتری ۹ ولتی را تشکیل داده اند و دیگری یک باتری کتابی 8.4 V استفاده شده است. از باتری اول جهت تغذیه میکروکنترلر , مدار و موتورهای DC میباشد. و از باتری دوم جهت تغذیه دوربین استفاده شده است.

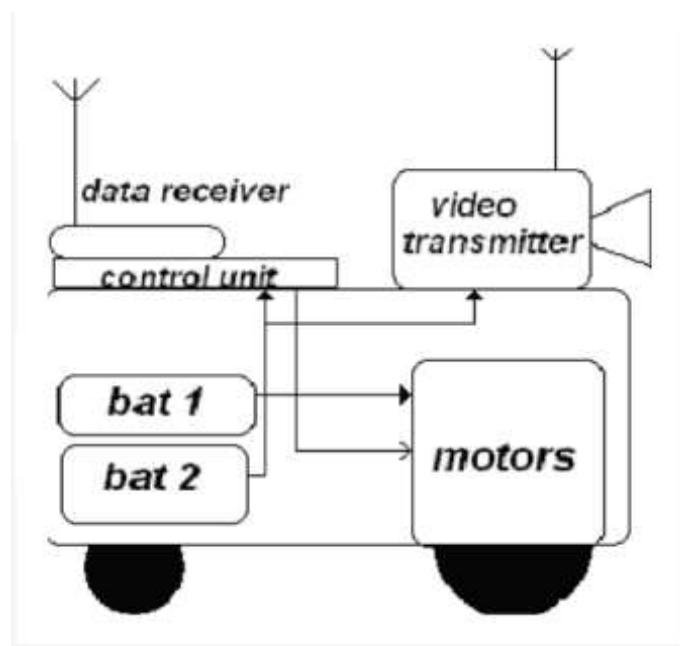
۶- موتورهای DC

۷- بدنه اصلی روبات

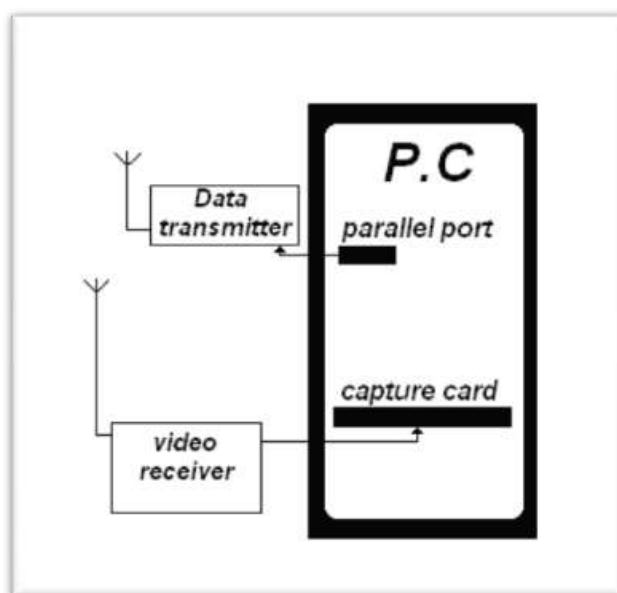


دیاگرام بلوکی کل سیستم

دیاگرام بلوکی روبات:



بخشهای ارتباطی با قسمت پردازنده مرکزی:



خلاصه کار سخت افزار:

ابتدا دوربین ما که از نوع بی سیم (Wire Less) نیز میباشد از محیط تصویر گرفته و اطلاعات تصویری را به صورت آنالوگ مدوله کرده و به گیرنده دوربین میفرستد. در گیرنده دوربین سیگنالهای آنالوگ رسیده به صورت AV در اختیار USER قرار میگیرد. این User در اختیار AV در گیرنده دوربین سیگنالهای آنالوگ رسیده به صورت این اطلاعات AV به کامپیوتر فرستاده میشود و از آنجا که این اطلاعات به صورت مستقیم و آنالوگ برای کامپیوتر قابل فهم و تجزیه و تحلیل نمیباشد لذا نیاز به واسطی جهت تبدیل این اطلاعات به کدهای دیجیتال (DIGITAL CODE) احساس میشود که این کار توسط ابزاری به نام کارت کپچر (CAPTURE CARD) و یا کارت تی وی (TV CARD) انجام میگیرد. پس از تبدیل سیگنالهای آنالوگ به کدهای دیجیتال این داده ها در محیط نرم افزار MATLAB پردازش میشوند که البته جهت پردازش میتوان از محیط برنامه نویسی دیگر همچون VB ، VC نیز استفاده کرد. پس از پردازش بر روی فریمهای تصویر نتایج این پردازش به صورت دستورالعملهایی به درگاه (Serial Port) کامپیوتر ارسال میگردد. پس از این مرحله اطلاعات روی پورت ، از طریق فرستنده FSK به گیرنده آن بر روی روبات ارسال میگردد و میکرو با توجه به ماهیت اطلاعات دریافتی سیگنالهای لازم جهت حرکت موتورها را فراهم میآورد. این امر به صورت مداوم انجام میگیرد به طوری که با حرکت روبات تصاویر بعدی ارسال میگردد و مجدداً دستورالعملهای مورد نیاز تهیه میشوند. باید خاطر نشان کنیم که حرکات روبات به سمت جلو، عقب ، چپ ، راست و ایستا میباشد.

نکات فنی قابل توجه:

در ساخت و استفاده از این روبات نکات فنی قابل توجهی وجود دارد که توجه شما را به چندی از آنها جلب میکنیم.

۱- به دلیل مصرف نسبتاً بالای دوربین ، باتریها حدوداً در مدت ۲ ساعت دشارژ میشوند و نیاز به شارژ مجدد دارند. البته اگر دوربین را با باتری مجزایی تغذیه کنیم مشکل افت ولتاژ بهبود میابد.

۲- با تغذیه دوربین با باتری جداگانه از اختلال در کار مدار RF نیز تا حد بسیار زیادی کاسته میشود همانطور که میدانیم مدارات RF به تغذیه بدون نویز دارند و کوچکترین خللی در این مورد موجب اختلال در کار مدار میگردد.

۳- از نظر فرکانسی باید فرکانس فرستنده دوربین با فرکانس دریافتی RF متفاوت باشد. در صورت تداخل فرکانس این دو بخش کار هر دو بخش مختل میگردد. بدین منظور فرکانس فرستنده RF (415) مگا هرتز و فرکانس کاری دوربین بیسیم (۱,۲) گیگا هرتز میباشد.

فصل سوم

شرح کلی عملکرد روبات :

این روبات دارای دوربینی است که مانند سیستم بینایی عمل میکند. به این ترتیب که اطلاعات تصویر از محیط دریافت میشود و در بخش بعد این data به صورت code به سیستم پردازنده ارسال میگردد. این در سطحی بسیار نازلتر شبیه همان اتفاقی است که در سیستم بینایی انسان رخ میدهد. با دقت به این نکته که حسگر بینایی در انسان همان چشم است و پردازنده نیز مغز آدمی است ، که البته از دقت و پیچیدگی بسیار بالاتری برخوردار است. البته تفاوت برجسته دیگری که کاملاً مشهود میباشد این است که این روبات دارای یک حسگر بینایی است ، لذا قادر به دیدن اجسام به صورت سه بعدی و تشخیص عمق نمیشد و در انسان با تعبیه دو حسگر بینایی یا به عبارتی دو چشم این مشکل حل شده است. پس از ارسال اطلاعات توسط دوربین به کامپیوتر (PC) در این بخش پردازشهای لازم نسبت به محتوی اطلاعات تصویر انجام شده و نتیجه به صورت دستورالعمل به موتورهای جهت حرکت دستگاه ارسال میگردد. عمل پردازش تصویر در بدن آدمی توسط مغز انجام میگردد و پایانه ارتباطی در بدن انسان همان اعصاب میباشد و در سیستمهای شبیه سازی شده میتواند از طریق سیگنالهای الکتریکی یا الکترومغناطیسی انجام پذیرد.

خلاصه عملکرد نرم افزاری روبات :

کار نرم افزاری که در این پروژه ارائه شده است بر پایه رنگهای محیط میباشد (Color Base) بدین صورت که وظیفه روبات تعقیب یک شیء (Object) رنگی است که ما رنگ قرمز را برای این منظور انتخاب کرده ایم. البته مشخص است که برای تعیین دقیق موقعیت obj نباید رنگهای محیط آزمایش به رنگ قرمز نزدیک باشد تا obj مورد نظر برای روبات کاملاً مشخص باشد.

با توجه به موقعیت پیکسلهای رنگی در تصویر، روبات دارای این توانایی است که در جهات مختلف حرکت کند و در صورت استقرار obj در مرکز افقی تصویر به سمت جلو رفته و پس نزدیک شدن به جسم مورد نظر درصد نور قرمز افزایش یافته و به این طریق روبات از حرکت می ایستد.

رایانه

بخش نرم افزاری قسمت اصلی جهت انجام فعالیتهای و در واقع تصمیم گیریهای روبات میباشد. با توجه به این مطلب روشن است که با هرچه قدرتمندتر کردن نرم افزار اجرایی سیستم میتوانیم فعالیتهای هوشمندانه تری برای روبات در نظر بگیریم. برای دستیابی به مقاصد ذکر شده نیاز به ابزاری جهت اجرای نرم افزار سیستم و در نهایت نیل به اهداف مشخص خود داریم.

از جمله ابزارهای مورد نیاز میتوان به موارد زیر اشاره نمود :

۱- کامپیوتر (PC) با قابلیت سرعت بالا و پردازش سریع

۲- محیط برنامه نویسی قدرتمند

۴- سخت افزارهایی که با کامپیوتر مرتبطند

۵- الگوریتمی مشخص بر اساس اهداف روبات

مشخصات رایانه مورد نیاز برای انجام عمل پردازش

به منظور پردازش هر چه سریعتر و با دقت بالاتر داده های ورودی توسط کامپیوتر (داده های مربوط به تصویر) نیاز به سیستمی نسبتاً سریع داریم. باید توجه داشت که علاوه بر دریافت اطلاعات می بایست اعمال ریاضی و منطقی در مورد این داده ها صورت گیرد. همچنین سیستم مذکور باید این توانایی را داشته باشد که بتوان نرم افزارها و محیط های برنامه نویسی نیرومندی نظیر MATLAB را بر روی سیستم نصب و اجرا کرد. همچنین باید دارای ورودی و خروجی های مورد نظر ما باشد. برای اجرای مناسب برنامه اجرائی ما سیستمی با مشخصات ذیل معرفی میکنیم

CPU : P4 1800 Mhz

RAM : 256MB DDR or SDRAM

OS : Windows XP

بخش نرم افزاری:

این بخش شامل دو قسمت است:

- نرم افزار کامپیوتر
- نرم افزار میکروکنترلر

که در ذیل به توضیح مختصری در مورد آنها میپردازیم:

۱- نرم افزار کامپیوتر:

نرم افزاری که روی کامپیوتر عمل پردازش را انجام میدهد تحت محیط MATLAB نوشته شده است و به عنوان مغز پروژه عمل میکند. که از آن برای پردازش تصویر (IMAGE PROCESSING) و ارسال دستور عملهای مورد نظر (COMMANDS) جهت حرکت موتورهای DC استفاده میشود.

۲- نرم افزار میکروکنترلر:

از این نرم افزار جهت فرمان دادن به موتورهای DC استفاده میشود که تحت محیط CodeVision avr نوشته شده است.

محیط برنامه نویسی:

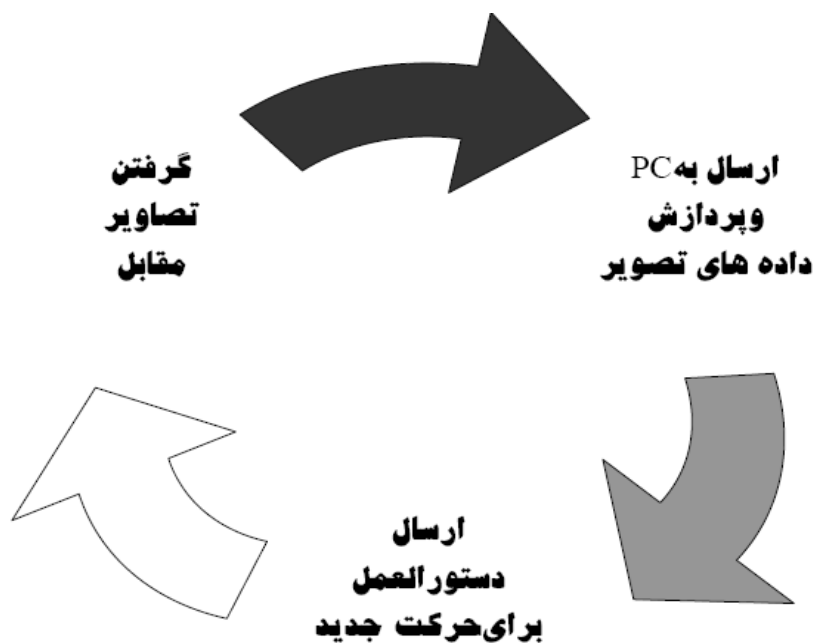
انتخاب یک محیط مناسب برای نوشتن و اجرای برنامه و پردازش تصاویر در این بخش بسیار حائز اهمیت است. محیطی که به این منظور در نظر گرفته میشود باید دارای قابلیت های مناسبی در زمینه های مرتبط با فعالیت ها و سخت افزارهای موجود در سیستم باشد. این بدین معناست که نرم افزار مزبور باید بتواند با ابزارهای موجود اعم از ابزارهای انتقال تصاویر و همچنین ورودی و خروجی های موجود ارتباط برقرار کند که البته این مورد در اکثر محیطهای برنامه نویسی در نظر گرفته شده است. با توجه به طرز عملکرد سیستم که مبتنی بر پردازش تصویر و شناسایی موقعیت ها در تصویر است لذا نرم افزار ذکر شده باید امکانات ویژه ای جهت پردازش تصویر در اختیار ما قرار دهد. از میان همه محیطهای برنامه نویسی با توجه به نکات ذکر شده MATLAB را برگزیده ایم. این نرم افزار به خاطر دارا بودن جعبه ابزارهای (Toolbox) توانمند در زمینه موضوعات مورد نظر ما گزینه بسیار خوبی میباشد.

سخت افزارهای مورد استفاده :

در مورد اجزای سخت افزاری مرتبط با کامپیوتر در بخشهای قبل توضیحات مبسوطی داده شد. در این بخش فقط میبایست به این نکته اشاره کرد که سخت افزاری که انتخاب میشود ممکن است با Motherboard کامپیوتر همخوانی نداشته باشد لذا در این مورد توجهات خاصی لازم است. همچنین ممکن است این سخت افزار توسط نرم افزارهای اجرائی سیستم عامل شناسایی نشود که این هم خود نقصی بزرگ به شمار می رود ؛ لذا در انتخاب سخت افزارها و بخصوص کارت Capture یا TV باید دقت لازم به عمل آید.

الگوریتم و برنامه اصلی :

در این قسمت مهمترین فاکتوری که پیش رو داریم و میبایست درباره آن تصمیم اصلی را اتخاذ کرد هدف یا اهداف روبات میباشد. با توجه به عمومیت سخت افزار برای روبات میتوان کارایی و اهداف متعددی را تصور کرد. بعنوان مثال میتوان به روبات مسیریاب یا سیستمی که به تحرکات عکس العمل نشان میدهد و مواردی از این دست اشاره کرد. با نظر به مطالب ذکر شده لازم است عملکردی خاص برای روبات در نظر بگیریم. این انتخاب تمامی تصمیم گیری های آتی را تحت الشعاع قرار میدهد. در این مرحله با تفکر در مورد نوع عملکرد روبات می بایست الگوریتمی در همان زمینه در نظر گرفت. البته تمامی این الگوریتم ها باید از چارت کلی سیستم که به صورت زیر است تبعیت کند.



با دقت به دیاگرام بالا بررسی سخت افزار موجود در این روبات و با عنایت به اینکه کارایی مکانیکی و تحرک این روبات دارای محدودیت هاییست باید سعی شود هدف در نظر گرفته

شده معقول و قابل اجرا روی این سیستم باشد. با توجه به دو موتوری که بر روی این روبات تعبیه شده است صرفاً روبات قادر به حرکت بر سطوح مسطح میباشد ، لذا در مورد فعالیتهایی که روبات نیاز به تحرک دیگری دارد این سخت افزار به کار نمیآید از جمله فعالیتهایی که روبات قادر به انجام آنهاست میتوان به چند مورد اشاره کرد:

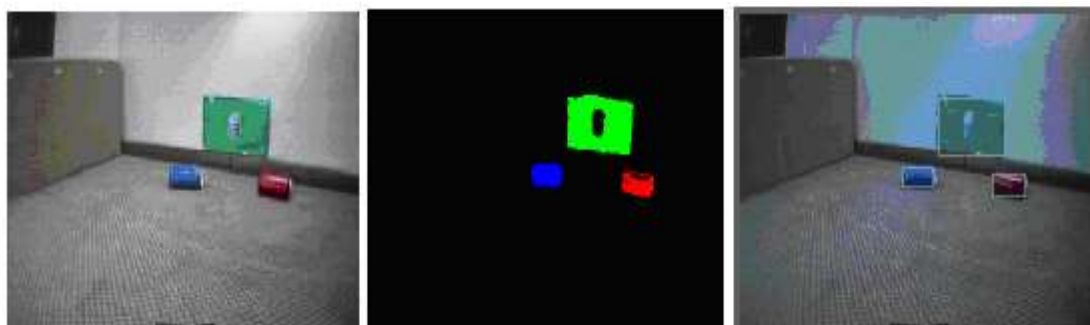
- استفاده بعنوان روبات مسیریاب
 - شناسایی و دنبال کردن اشیایی خاص (بر اساس رنگ یا شکل)
 - طی کردن مسیرهای از پیش تعیین شده
 - عبور از مسیرهایی با موانعی در اطراف
 - شناسایی و تعیین موقعیت یک نقطه یا شی متحرک و تعقیب آن
- برای شروع کار می بایست مراحل طی گردد که مرحله اول انتخاب یکی از اهداف بالاست. در این مورد به مراحل که بطور مشروح در زیر بدان اشاره شده توجه کنید :

۱- تعیین هدف و عملکرد :

در ابتدا می بایست هدف اصلی روبات تعیین گردد. همانطور که در بالا دیده شد میتوان راههای مختلفی را برگزید. در این پروژه و برنامه نمونه مورد نظرها هدف بدست آوردن موقعیت یک شی با رنگ قرمز است که میتواند یک توپ قرمز یا شی قرمز دیگری باشد. پس از مشخص شدن موقعیت جسم کامپیوتر به سرعت دستورات در نظر گرفته شده و اطلاعات لازم را در اختیار روبات قرار میدهد و روبات شروع به حرکت بسوی هدف میکند. سپس با نزدیک شدن به جسم روبات متوقف شده و از برنامه خارج میشود.

۲- تهیه الگوریتم برای هدف مورد نظر :

برای نیل به هدف مورد نظر لازم است بخشهای اشاره شده را به صورت الگوریتمی مناسب برای برنامه نویسی تبدیل کرد. از نکات مهم و قابل توجه در این بخش طریقه تعیین موقعیت شیء مورد نظر است که در اینجا بر پایه رنگ (Color Bas) و بر اساس اطلاعات حاصله از ماتریسهای RGB میباشد. در مرحله بعد و پس از تعیین موقعیت جسم ، وضعیت موتورهای به گونه ای تغییر میابد که تصویر شیء مزبور به سمت مرکز تصویر متمایل گردد این فرآیند به صورت زیر انجام می پذیرد. با توجه به موقعیت جسم نسبت به مرکز تصویر که امکان دارد در طرفین آن باشد ، فرمانی برای موتورهای ارسال میگردد که اگر جسم سمت راست باشد روبات به سمت راست دور بزند و بالعکس اگر سمت چپ باشد به سمت چپ دور بزند و اگر در مرکز تصویر باشد به سمت مقابل حرکت کند. تنها حالتی که در این قسمت باقی مانده حالیست که جسم در کادر تصویر دوربین نباشد. برای رفع این شرایط راهی که پیشنهاد میشود چرخش و جستجوی روبات برای یافتن شیء میباشد. توجه شما را به تصاویر زیر که در مورد تعیین و جدا سازی رنگهای اصلی از یک فریم است جلب میکنیم.

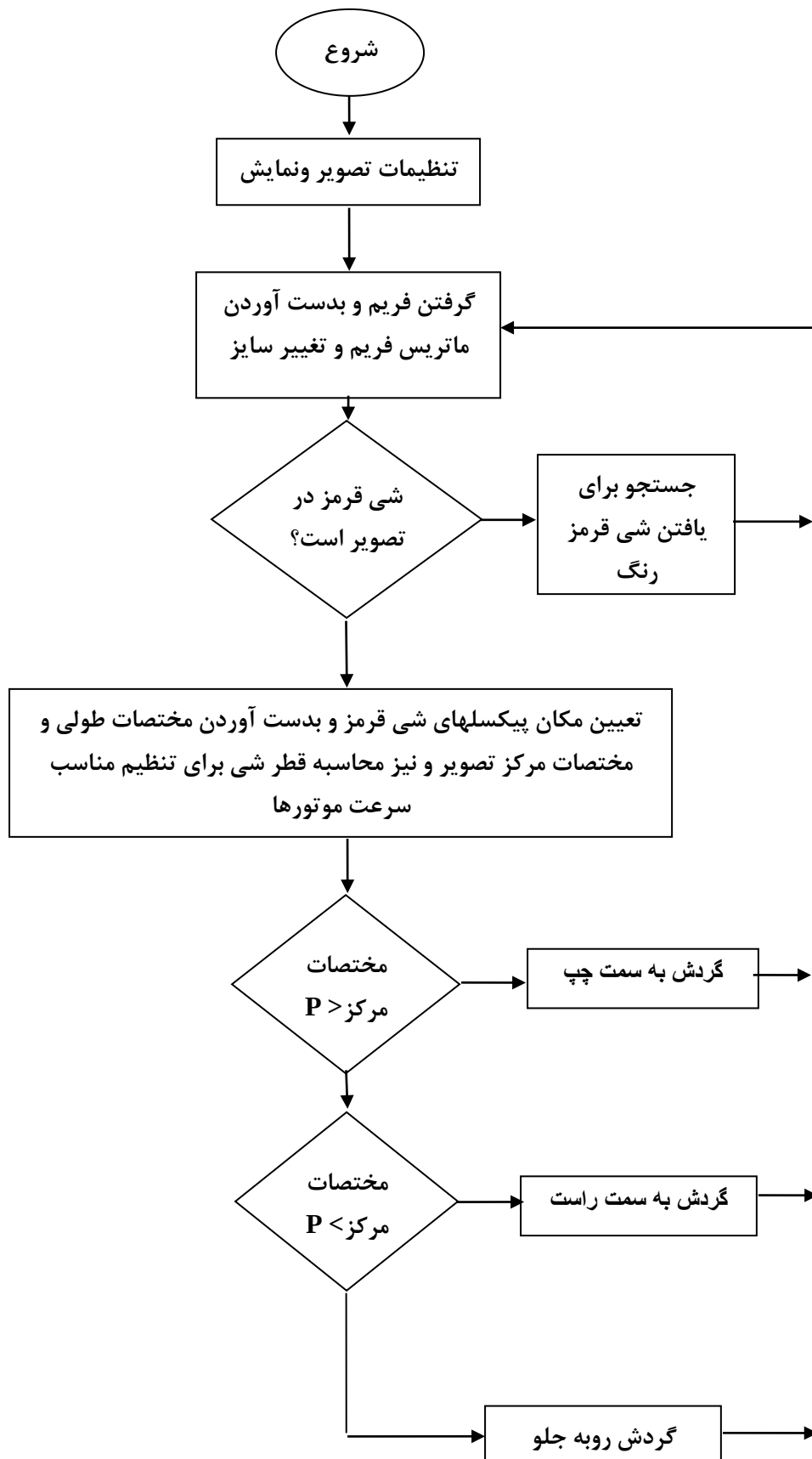


الف - تصویر

ب - جدا سازی رنگها

ج - مرز بندی برای رنگها و تعیین موقعیت

با در نظر داشتن مطالب ذکر شده اکنون به فلو چارت کلی برنامه توجه کنید.



۳- تهیه نرم افزار از روی فلوچارت :

در این بخش نوبت به نوشتن برنامه با توجه به فلوچارت عملکرد برنامه است. به این منظور همانطور که گفته شد از نرم افزار Matlab و محیط کد نویسی آن استفاده شده است. در اینجا به شرح بخشهای مختلف برنامه پرداخته و در مورد هر قسمت توضیحات لازم را متذکر میشویم.

قدم اول در شروع برنامه شناسائی و تنظیم بخش دریافت تصویر است. در این مرحله ابتدا نوع ابزار ویدئویی و مبدأ آن مشخص میشود ، سپس تنظیمات مربوط به سیستم رنگ ، شدت نور (brightness) ؛ تفکیک تصویر (contrast) و و سطح اشباع رنگ (saturation) انجام شده است. در نهایت تصویر دوربین در پنجره ای نمایش داده میشود. دستورات اولیه این بخش را در ذیل ملاحظه میکنید.

```
vid=videoinput ('winvideo',1,'YUY2_720x576') ;
```

```
vid.selectedsource='composit' ;
```

```
source=vid.source;
```

```
src=source(1) ;
```

```
src.brightness=1000;
```

```
src.contrast=3500;
```

```
src.saturation=7000;
```

```
src.analogvideoformat='pal_b' ;
```

```
set(vid,'ReturnedColorSpace','rgb') ;
```

```
preview(vid)
```

شرح تفصیلی برنامه :

قدم بعدی بدست آوردن ماتریسهای رنگ تصویر و انجام اعمال محاسباتی روی آنها و ارسال نتیجه به قسمتهای بعدیست. در این گام میبایست اولاً سائز تصویر کوچکتر شود (SCALE مورد استفاده برای کوچک کردن تصویر ۰,۱ است) و سپس ماتریس رنگ قرمز را از رنگهای دیگر تفکیک کرد. کوچکتر کردن ابعاد تصویر چند مزیت دارد که از جمله این مزایا بررسی سریعتر و ساده تر آن میباشد. همانگونه که مطلعید رنگهای موجود در تصویر را میتوان به سه رنگ اصلی تجزیه کرد.

در این قسمت از برنامه ما ماتریس رنگ قرمز را بدست می آوریم و سپس برای دستیابی به نقاطی که صرفاً قرمزند ماتریس تبدیل شده سیاه و سفید را از ماتریس قرمز کم میکنیم. بدین ترتیب کلیه نقاطی که دارای رنگ سفیدند که در ماتریس قرمز نیز موجودند حذف شده و فقط پیکسلهای قرمز باقی میمانند. این بخش داخل حلقه اصلی برنامه قرار گرفته و بمنظور ماندن در داخل حلقه درصد رنگ قرمز در سطرها سنجیده میگردد. در اینجا اگر میزان رنگ قرمز تشخیص داده شده که متناسب با فاصله از جسم میباشد به میزان مطلوبی رسیده بود برنامه از حلقه اصلی خارج شده و به اتمام میرسد. دستورات مربوط به توضیحات بالا به صورت زیر می باشند.

```
a=getsnapshot(vid) ;
```

```
b=imresize(a,scale) ; % scale = 0.1 %
```

```
c=rgb2gray(b) ;
```

```
d=b(:, :, 1) - c ;
```

پس از بدست آوردن ماتریس رنگ قرمز نوبت به یافتن موقعیت تجمع پیکسل‌های قرمز میرسد این بخش نیز در ادامه بخش قبل از دستورات داخل حلقه اصلی میباشد. این قسمت شامل چندین حلقه تو در تو میگردد. باید دقت داشت که حلقه ها با توجه به ابعاد تغییر یافته تصویر در نظر گرفته شده اند. باید ذکر کرد که در درون این حلقه ها پس از تعیین ناحیه مشخص ، اطلاعات مشخصی برای حرکت روبات به خروجی پورت سریال فرستاده میشود. به دستورات زیر که مربوط به توضیحات این بخش میباشند توجه کنید :

```
[n,m]=size(d);           % n= ROWS , m= COLUMNS %
```

```
for j=1:m
```

```
    for i=1:n
```

```
        if d(i,j)>mincolor, i1=i ; j1=j ; break
```

```
    end
```

```
end
```

```
if d(i,j)>mincolor ,break , end
```

```
end
```

```
for j=m:-1:1
```

```
    for i=n:-1:1
```

```
        if d(i,j)>mincolor, i2=i; j2=j; break
```

```
    end
```

```
end
```

```
if d(i,j)>mincolor ,break , end
```

```
end
```

برنامه فوق را میتوان به اینگونه شرح داد که پس از بدست آوردن ماتریس رنگ قرمز و متمایز کردن رنگ قرمز از سایر رنگ های موجود در تصویر نوبت به محاسبات مورد نیاز برای تشخیص مکان فعلی شی مورد نظر میرسد به این ترتیب که اگر شی مورد نظر ما که دارای مشخصات فیزیکی تعریف شده (رنگ قرمز) داخل تصویر دوربین نباشد به موتورهای روبات دستور داده میشود که یک دور کامل با سرعت ثابت و تعریف شده برای جستجوی شی مورد نظر بچرخد تا اینکه شی قرمز رنگ در میدان دید دوربین قرار بگیرد. حال پس از پیدا کردن شی قرمز رنگ و متمایز کردن پیکسلهای آن از سایر رنگها باید محاسبات مربوط به بدست آوردن مرکز تصویر و نیز مرکز شی (توپ قرمز رنگ) را انجام بدهیم. حال این سوال مطرح است که انجام این محاسبات چگونه ما را در رسیدن به هدف مورد نظر یاری می کنند. ما با بدست آوردن مرکز تصویر و توپ قرمز رنگ و محاسبه اختلاف بین این دو عدد می توانیم تشخیص بدهیم که شی در کجای تصویر قرار دارد. به این صورت که اگر مختصات طولی مرکز توپ از مختصات طولی مرکز تصویر کمتر باشد به این نتیجه میرسیم که شی مد نظر سمت چپ تصویر قرار دارد و باید موتور سمت راست به سمت جلو و موتور سمت چپ در جهت مخالف به حرکت در آیند که در نتیجه آن روبات به سمت چپ حرکت میکند در غیر اینصورت اگر مختصات طولی مرکز توپ از مختصات طولی مرکز تصویر بیشتر باشد با این نتیجه میرسیم که شی مورد نظر سمت راست تصویر قرار دارد و باید موتور سمت راست به سمت جلو و موتور سمت چپ در جهت مخالف به حرکت در آیند و باید موتور سمت راست به سمت عقب و موتور سمت چپ در جهت مخالف (رو به جلو) به حرکت در آیند که در نتیجه آن روبات به سمت راست حرکت میکند.

ما در محاسبات خود این را نیز لحاظ کردیم که موتورها با چه سرعتی به سمت جسم حرکت کنند از این رو برای نیل به این مقصود نیاز به محاسبه پارامتر دیگری اهمیت پیدا کرد ما برای حل این مسأله تصمیم گرفتیم که از قطر توپ برای اعمال سرعت مناسب به روبات برای حرکت استفاده کنیم. به این صورت که هرچه توپ به روبات نزدیکتر باشد مقدار پیکسلهای قرمز که در تصویر هستند افزایش پیدا میکنند و میتوان تشخیص داد که روبات با جسم فاصله کمی دارد به همین صورت اگر توپ در فاصله بیشتری نسبت به روبات قرار داشته باشد روبات میفهمد که با جسم مورد نظر فاصله زیادی دارد.

بر اساس این نظریه و تعریف کردن سه نقطه ایده ال برای چشم روبات و بدست آوردن معادله خط های مورد نظر برای بدست آوردن یک رابطه خطی برای سرعت روبات به معادله تقریبی زیر دست پیدا کردیم.

$$Y = - 6.75 (X-25)$$

که در این معادله (Y) همان سرعت مورد نظر است و متغیر (X) قطر شی قرمز رنگ است. عدد ۲۵ که در معادله فوق مشاهده میکنید همان قطر شی قرمز است در حالتیکه شی در فاصله مناسب از روبات قرار دارد که بر حسب نیاز می تواند تغییر کند که در اینصورت معادله سرعت تغییر میکند. سرعت بدست آمده از معادله فوق یک مقدار مثبت و حقیقی است که به هر دو موتور روبات اعمال می شود یعنی در این حالت روبات با اعمال سرعت برابر به هر دو موتور به سمت جسم حرکت میکند. اما این برای حالتی است که شی تقریباً در مرکز روبات قرار داشته باشد و نیاز به تغییر جهت نداشته باشد.

برای حالتی که شی مد نظر در مرکز تصویر نباشد و ما نیاز به تغییر جهت روبات داشته باشیم ما پارامتر دیگری با نام (dv) تعریف کردیم که این بار با اعمال این مقدار به موتورها، میتوانیم جهت حرکت روبات را نیز مشخص کنیم. این همان مقداریست که از اختلاف مختصات طولی مرکز تصویر و مرکز شی بدست می آید که در فوق شرح مبسوطی در مورد آن تقدیم شد.

معادله (dv) به صورت روبرو بدست می آید :

$$dv = k (oj - m / 2)$$

که در آن متغیر dv اختلاف سرعت بین موتورهای سمت چپ و راست است. مقدار oj مختصات طولی بدست آمده برای مرکز شی مورد نظر (توپ قرمز) است و m مختصات طولی کل تصویر و $m / 2$ مختصات طولی مرکز تصویر گرفته شده از دوربین است.

حال معادله کلی سرعت ، که به موتورهای روبات اعمال میشود به صورت زیر است :

$RV = (V - dv)$: سرعت اعمالی به موتور سمت راست

$LV = (V - dv)$: سرعت اعمالی به موتور سمت چپ

فصل در حساب

توضیحات تکمیلی

در این بخش ابتدا مختصری درباره سیستم بینایی ماشین توضیح می‌دهیم و سپس سعی شده است که برای قسمت‌های جانبی روبات که نه تنها در این پروژه بلکه در طرح‌های دیگری هم میتواند مورد استفاده قرار گیرد توضیحات مشروحی داده شود.

این بخش، قسمت‌های زیر را شامل میشود :

۱- سیستم بینایی ماشین

۲- دوربین

۳- مدار فرستنده و گیرنده (RF)

توضیح در مورد سیستم‌های بینایی و فرمان ماشین :

کارکرد ماشین‌های اتوماتیک و هوشمند معمولاً به این صورت است که سنسورها یا حسگرها در خروجی وضعیت کنونی سیستم را به صورت فیدبک در اختیار قسمت هوشمند سیستم قرار میدهند. شبیه این عملکرد در بدن انسان نیز وجود دارد به طوری که حواس پنجگانه یا همان حسگرها از طریق اعصاب با مغز و دیگر قسمت‌های فرمان دهنده بدن در ارتباط هستند. در سیستم‌های هوشمند نیز حسگرها نقش اساسی را ایفا میکنند. مثلاً در کوره‌های صنعتی حسگرهای حرارتی دمای محیط کوره را گرفته و این اطلاعات را به بخش CPU می‌فرستد و واحد پردازنده با توجه به اطلاعات دریافتی دمای کوره را کاهش یا افزایش میدهد. البته این یک مثال و کاربرد بسیار ساده و در عین حال دقیق از سیستم‌های هوشمند بود. سیستم‌های هوشمند کاربردهای به مراتب گسترده‌تر و پیچیده‌تری دارند که برای مثال

میتوان به سیستمهای CNC ، تشخیص هویت و یا اکتشافات فضایی اشاره کرد. برای توضیح بیشتر به سیستمهای CNC اشاره میکنیم. در این سیستمها نیاز به دقت و ظرافت از ضروریات میباشد. بنابراین باید از سنسورهای بسیار دقیق و سیستم هوشمندی با توانایی نسبتاً بالا بهره برد. کار کلی این مجموعه به این صورت است که محورهایی در چندین جهت قطعه مورد نظر برای تراشکاری را جابجا میکنند و یا اینکه تیغه های تراشکاری جابجا میشوند. برای این منظور نیاز به سنسورهای اندازه گیری طول داریم. این سنسورها میتوانند در این سیستم دقتی در حدود میکرو متر را ارائه کنند. به این صورت که اطلاعات دریافت شده در مورد جابجا شدن محورها در اختیار قسمت پردازنده قرار میگیرد و با توجه به برنامه موجود برای تراش قطعه ، فرمانهای لازم برای جابجایی محورها به سروو موتورهای یا موتورهای DC اعمال میشود. این روال ادامه مییابد تا اینکه مجدداً اطلاعات حرکت محورها در اختیار پردازنده قرار میگیرد و دوباره دستورات لازم به موتورهای ارسال میگردد. البته این یک بیان کلی و ساده از طرز عملکرد سیستمهای CNC بود.

امروزه سیستمهای هوشمند نقش بسزایی در تمام شاخه های علمی ، صنعتی و پزشکی دارند و دقت این سیستمها و اعتماد به آنها با توجه به پیشرفتهایی که در پردازنده ها و سنسورها حاصل شده است به حدی رسیده است که در زمینه های بسیار حساس و ظریف مانند پزشکی و هسته ای مورد استفاده قرار میگیرند.

مثلا امروزه روباتهای هوشمندی طراحی شده است که قادر هستند اطلاعات را از طریق سیگنالهای الکتریکی اعصاب بدن تهیه کنند و برای مثال ، اعضاء بدن را در انجام حرکات یاری کنند. برای این روباتها سنسورها در واقع همان حواس پنجگانه انسان است. امروزه شاهد

پیشرفتهای بیشتری در این زمینه هستیم و با به وجود آمدن رشته هایی از جمله هوش مصنوعی و پیشرفت رشته هایی از جمله مهندسی پزشکی امیدهای تازه ای در برابر دانشمندان قرار دارد و ما نیز امیدواریم با توجه به هوش و استعداد دانشمندان و محققان ایران زمین ما نیز نقش به سزایی در پیشرفت و پیشبرد اهداف این رشته که در نهایت خدمت به نوع بشر و حتی جانداران است را داشته باشیم. سیستمی که ارائه شده است یک روبات نیمه هوشمند است که قادر به تشخیص و تفکیک تصاویر میباشد. در شکل زیر نمونه ای از کاربردهای سیستم بینایی هوشمند مشاهده میشود. این سیستم که در خودرو تعبیه شده در مواقعی که انسان یا موجود متحرک دیگری را در محدوده عبور ماشین ببیند آنرا تشخیص داده و با کم کردن سرعت ، مانع از بروز تصادف میگردد. این حالت در شب هنگام بسیار مفید و موثر واقع میشود و از بروز بسیاری از تصادفات جلوگیری میکند. بدینوسیله میبینیم که اینگونه سیستمها در نجات جان انسانها نیز نقش دارند.



سخت افزار دوربین:

در این بخش به جزئیاتی راجع به دوربین میپردازیم. همانطور که در بخشهای قبلی گفته شد، دوربین استفاده شده در این روبات از نوع بی سیم میباشد. فرکانس کاری مدار RF دوربین

1.2GHz است. فرستنده درون خود دوربین می باشد. دوربین توسط ولتاژ ۸,۴ تغذیه میشود. این ولتاژ از طریق یک باتری ۸,۴ ولتی تأمین میشود. مقدار جریان مصرفی دوربین در حدود 250 mA است. که با وجود باتریهای مناسب این جریان به خوبی تأمین میشود. شکل ظاهری این دوربین به صورت زیر میباشد.



شرح گیرنده :

امواج فرستاده شده توسط دوربین که حاوی اطلاعات صوتی و تصویری میباشد ، در بخش گیرنده تبدیل به سیگنال آنالوگ AV میشوند. روی این گیرنده پیچ تنظیمی تعبیه شده که این امکان را میدهد که فرکانس گیرنده را تنظیم کرد ، بدین ترتیب میتوان دوربین مورد نظر را با توجه به فرکانس آن شناسائی و تصویر آن را دریافت کرد. همچنین روی این گیرنده دو خروجی وجود دارد که سیگنالهای صوتی و تصویری را میدهند. تغذیه این بخش توسط یک آداپتور 12V تأمین میشود. به این منظور یک ورودی روی گیرنده قرار دارد.

طریقه دستیابی به تصویر و تنظیمات در نرم افزار MATLAB :

همانطور که قبلاً اشاره شد ، سیگنال ویدئویی توسط بخش واسط کارت کپچر به اطلاعات مورد نظر ما تبدیل میشود. از این مرحله به بعد کار با نرم افزار شروع میشود هدف در این مرحله شناسائی کارت کپچر یا TV است. پس از این نوبت به گرفتن اطلاعات تصویر از کارت و پردازش نرم افزاری روی آن میباشد. در نرم افزار MATLAB یک جعبه ابزار نسبتاً قدرتمند برای کار با ابزارهای تصویری در نظر گرفته شده است. این جعبه ابزار IMAGEACQUISITION TOOLBOX میباشد. جزئیات دقیقتر Toolbox تهیه تصویر (Image Acquisition Toolbox) در matlab help موجود است. که ما در زیر مختصراً به قسمتهای استفاده شده در این پروژه اشاره میکنیم.

۱- نصب سخت افزار مورد نظر :

که در این پروژه کارت TV TUNER میباشد. در اینجا لازم است این نکته را یادآور شویم که تمامی ابزارهای ویدئویی قابل شناسایی توسط MATLAB نمیشوند ؛ به این منظور راه کارهایی وجود دارد. از جمله این راه کارها استفاده از یک نرم افزار اجرایی است. در صورتیکه ابزار مورد نظر با این نرم افزار شناخته شود ، و بتوانیم از طریق این نرم افزار تصویری دریافت کنیم ؛ میتوان گفت که این ابزار از طریق MATLAB قابل شناسایی و استفاده میباشد. آدرس این فایل در زیر آمده است

MATLAB\toolbox\imaq\imaq\AMCap.exe

البته میبایست قبل از اجرای نرم افزار بالا، درایورهای ابزار با دقت نصب شود. که البته به این مورد در بخش دوم اشاره شده است.

۲- شناسایی ابزار (کارت کپچر) وبدست آوردن مشخصات آن در MATLAB:

در این مرحله چندین قسمت از اطلاعات مورد نیاز این Toolbox در مورد سخت افزار بدست آورده میشود که این اطلاعات وسیله مورد نظر را به MATLAB معرفی میکند. این اطلاعات را وقتی در حال تعریف یک Object تهیه تصویر مورد استفاده قرار میگیرد. در زیر لیست اطلاعات برای این قسمت با کمی توضیح آورده شده است:

:Adaptor Name

یک adaptor نرم افزاری است که toolbox برای ارتباط با دوربین از طریق درایور دوربین آن را مورد استفاده قرار میدهد.

: Device ID

Device id شماره ای است که نرم افزار matlab در صورت بکار بردن چند دوربین به هرکدام از آنها اختصاص میدهد و در هر دستور با ذکر آن شماره مشخص میشود که با کدامیک از آن دوربینها در حال برقراری ارتباط هستیم. توجه شود که استفاده کردن از Device ID اختیاری میباشد و در حالت پیش فرض toolbox از اولین Device ID در دسترس استفاده میکند.

: Video Format

مشخص کننده دقت تصویر (طول و عرض بر حسب تعداد پیکسل وتعیین تعداد بیت هر پیکسل) است. لازم به ذکر است که عموما هر وسیله تهیه تصویر چند نوع فرمت تصویر را پشتیبانی میکند. توجه شود که مشخص کردن فرمت تصویر اختیاری است و در صورت

مشخص نکردن آن Toolbox از حالت پیش فرض دوربین استفاده میکند. برای مشخص کردن این آرگومانها به صورت زیر عمل میکنیم :

برای پیدا کردن adaptor name تابع imaqhwinfo را در محیط اصلی matlab بدون هیچ آرگومانی اجرا میکنیم ، خواهیم داشت :

```
>> imaqhwinfo
```

```
ans =
```

```
InstalledAdaptors: {'coreco' 'dcam' 'winvideo'}
```

```
MATLABVersion: '7.6 (R2008a) '
```

```
ToolboxName: 'Image Acquisition Toolbox'
```

```
ToolboxVersion: '3.1 (R2008a)
```

اطلاعات بازگشتی بوسیله تابع imaqhwinfo لیست adaptor هایی که بر روی کامپیوتر موجود میباشد را نشان میدهد. در این مثال imaqhwinfo یک adaptor که در این پروژه 'winvideo' (adaptor مربوط به tv tuner) میباشد و روی کامپیوتر موجود است را پیدا میکند در صورت بکار رفتن چند Adaptor امکان دسترسی به هر کدام از دوربین ها را پیدا میکند.

به جدول زیر که تعیین کننده Adaptor های وسایل مختلف است توجه کنید :

Adaptor name	Adaptors Included with the Image Acquisition Toolbox Description
'coreco'	Adaptor for image acquisition devices produced by Coreco Imaging, Inc.
'dcam'	Adaptor for IEEE 1394 (FireWire) image acquisition devices that comply with the IIDC 1394-based Digital Camera Specification (DCAM), developed by the Digital Camera subgroup of the 1394 Trade Association's Instrumentation and Industrial Control working group.
'dt'	Adaptor for image acquisition devices produced by Data Translation , Inc.
'matrox'	Adaptor for image acquisition devices produced by Matrox Electronic Systems, Ltd.
'winvideo'	Adaptor for devices that provide a Windows Driver Model (WDM) or Video for Windows (VFW) driver, including USB and IEEE 1394 (FireWire) cameras.

برای پیدا کردن Device ID تابع `imaqhwinfo` را همراه با `Adaptor Name` بدست آمده از قسمت قبل بصورت آرگومان آن در محیط اصلی اجرا میکنیم ، خواهیم داشت :

```
>> imaqhwinfo('winvideo')
ans =
AdaptorDllName: [1x61 char]
AdaptorDllVersion: '1.5 (R14)'
AdaptorName: 'winvideo'
DeviceIDs: {[1]}
```

این اطلاعات بازگشتی Device ID دوربین ، Adaptor Name ، و چند مورد دیگر را نشان میدهد. برای بدست آوردن Video Format، تابع imaqhwinfo همراه با آرگومانهای Adaptor Name و Device ID مشخص شده در قسمتهای قبل در محیط اصلی MATLAB اجرا میکنیم ، در اینصورت خواهیم داشت :

```
>> imaqhwinfo('winvideo',1)

ans =

DefaultFormat: 'RGB555_768x576'

DeviceFileSupported: 0

DeviceName: 'Bt878 TV Card'

DeviceID: 1

ObjectConstructor: 'videoinput('winvideo', 1) '

SupportedFormats: {1x5 cell}
```

اطلاعات بازگشتی شامل نام دوربین و فرمت پیش فرض آن و تکرار اطلاعات قبلی داده شده در مورد دوربین میباشد. جهت بدست آوردن دیگر فرمتهای پشتیبانی توسط دوربین بصورت زیر عمل میکنیم :

```
>>dev_info = imaqhwinfo('winvideo',1)

dev_info =

DefaultFormat: 'RGB555_768x576'

DeviceFileSupported: 0

DeviceName: 'Bt878 TV Card'

DeviceID: 1

ObjectConstructor: 'videoinput('winvideo', 1)'

SupportedFormats: {1x5 cell}

>> celldisp(dev_info.SupportedFormats)

ans{1} =

RGB555_768x576

ans{2} =

RGB24_768x576

ans{3} =

YUY2_768x576

ans{4} =

YVU9_768x576

ans{5} =

YV12_768x576
```

همانطور که مشاهده میشود فرمت‌های موجود بدست می‌آیند.

۳- ساختن یک video input object :

در این قدم بوسیله ساختن یک Video Input Object ارتباط میان MATLAB و وسیله تهیه تصویر برقرار میشود.

در ساده ترین حالت برای ساختن یک Video Input Object از تابع `videoinput` تنها به همراه آرگومان `Adaptor` مشخص شده در قدم چهارم بصورت زیر استفاده میشود :

```
>> vid=videoinput('winvideo',1)
```

Summary of Video Input Object Using 'Bt878 TV Card'.

Acquisition Source(s): composite, svideo, and tuner are available.

Acquisition Parameters: 'composite' is the current selected source.

10 frames per trigger using the selected source.

'RGB555_768x576' video data to be logged upon START.

Grabbing first of every 1 frame(s).

Log data to 'memory' on trigger.

Trigger Parameters: 1 'immediate' trigger(s) on START.

Status: Waiting for START.

0 frames acquired since starting.

0 frames available for GETDATA.

برای انتخاب یک فرمت مشخص از بین فرمتهایی که در بخش قبل به آن اشاره شد به طریقه زیر عمل کرده و مجدداً `vid` را با فرمت `RGB555_768x576` تعریف میکنیم.

```
>> vid=videoinput('winvideo',1,'RGB555_768x576')
```

Summary of Video Input Object Using 'Bt878 TV Card'.

Acquisition Source(s): composite, svideo, and tuner are available.

Acquisition Parameters: 'composite' is the current selected source.

10 frames per trigger using the selected source.

'RGB555_768x576' video data to be logged upon

START.

Grabbing first of every 1 frame(s).

Log data to 'memory' on trigger.

Trigger Parameters: 1 'immediate' trigger(s) on START.

Status: Waiting for START.

0 frames acquired since starting.

0 frames available for GETDATA.

۴-نمایش preview :

بعد از ساختن matlab، Video Input Object قادر است که دوربین را برای بدست

آوردن تصویر بکار برد ، اما قبل از شروع تصویر برداری ممکن است بخواهیم که یک

Preview از رشته فریمهایی که دوربین تهیه میکند را ببینیم ، تا از وضعیت مکانی تصویر و

شدت نور و وضوح و ... آن آگاه شویم.

برای این منظور تابع Preview را همراه با آرگومان Video Input Object مثلا بصورت preview(vid) بکار میبریم پس از اجرای این تابع ، پنجره ای باز میشود که رشته فریمهای تهیه شده توسط دوربین را بصورت زنده نمایش میدهد.

تهیه فریم در یک ماتریس :

دو دستور اصلی تهیه فریم در یک ماتریس به صورتهای زیر میباشد :

: Getsnapshot (a

قالب کلی این دستور به صورت $a = \text{getsnapshot}(\text{vid})$ میباشد. پس از اجرای این دستور یک فریم در ماتریس a ذخیره میشود. که سرعت اجرای آن در حالت باز بودن preview بیشتر است.

: Getdata (b

قبل از اجرای این دستور میبایست که Video Input Object توسط دستور start(vid) ، start شده باشد. سپس دستور getdata قادر است که تعداد فریمهای متغیری را بصورت $a = \text{getdata}(\text{vid}, n)$ (که n تعداد فریمهای مد نظر است) تا حداکثر مقدار پیش فرض ۱۰ فریم در ماتریس a تهیه کند. به اینصورت که $a(:, :, 1)$ اولین فریم و $a(:, :, 2)$ دومین فریم و ... را در خود در بر دارد.

این تعداد را میتوان از آن توسط دستور `set(vid,'framespertrigger', m)` تغییر داد که `m` تعداد کل فریمهایی است که بعد از `start` شدن میتوان توسط دستور `getdata` چه در یک بار و یا چند بار اجرای آن تهیه کرد. بعد از اتمام تهیه تصویر توسط این دستور میبایست که `Video Input Object` توسط دستور `stop(vid)` ، `stop` شود.

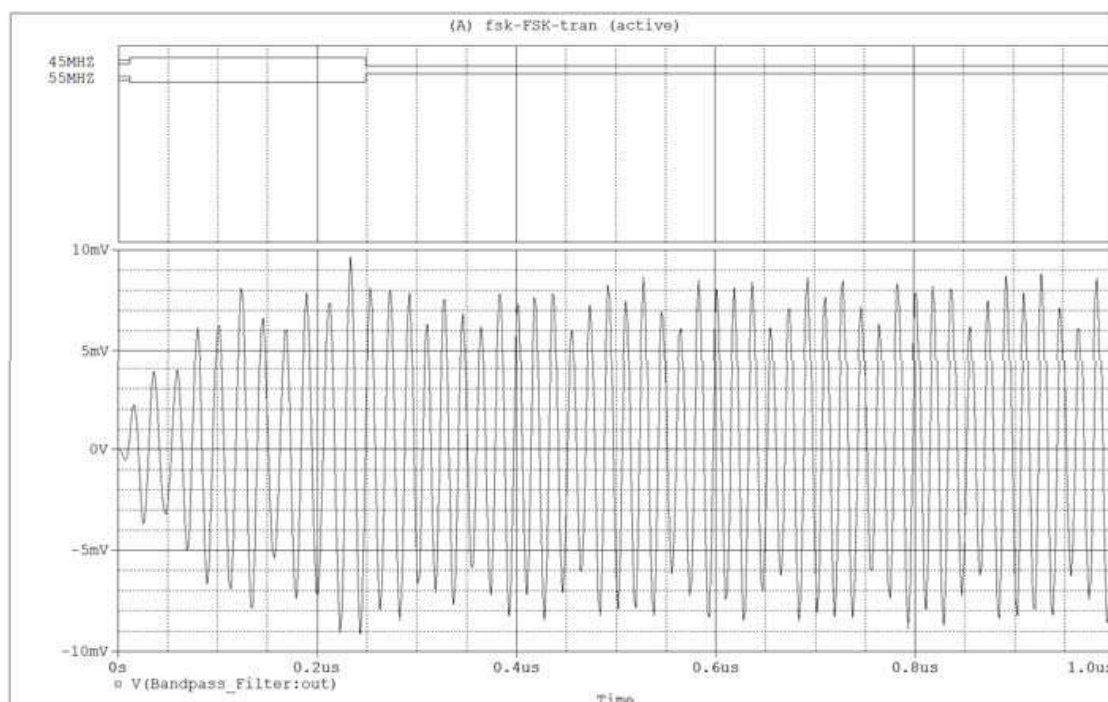
پس از طی کردن این مراحل همانطور که مشاهده شد اطلاعات مربوط به فریمهای تصویر به صورت ماتریس در اختیار ما قرار گرفته است. باتوجه به اطلاعات موجود حال میتوان پروسه پردازش اطلاعات تصویر را در `metlab` آغاز کرد.

ارسال و دریافت داده :

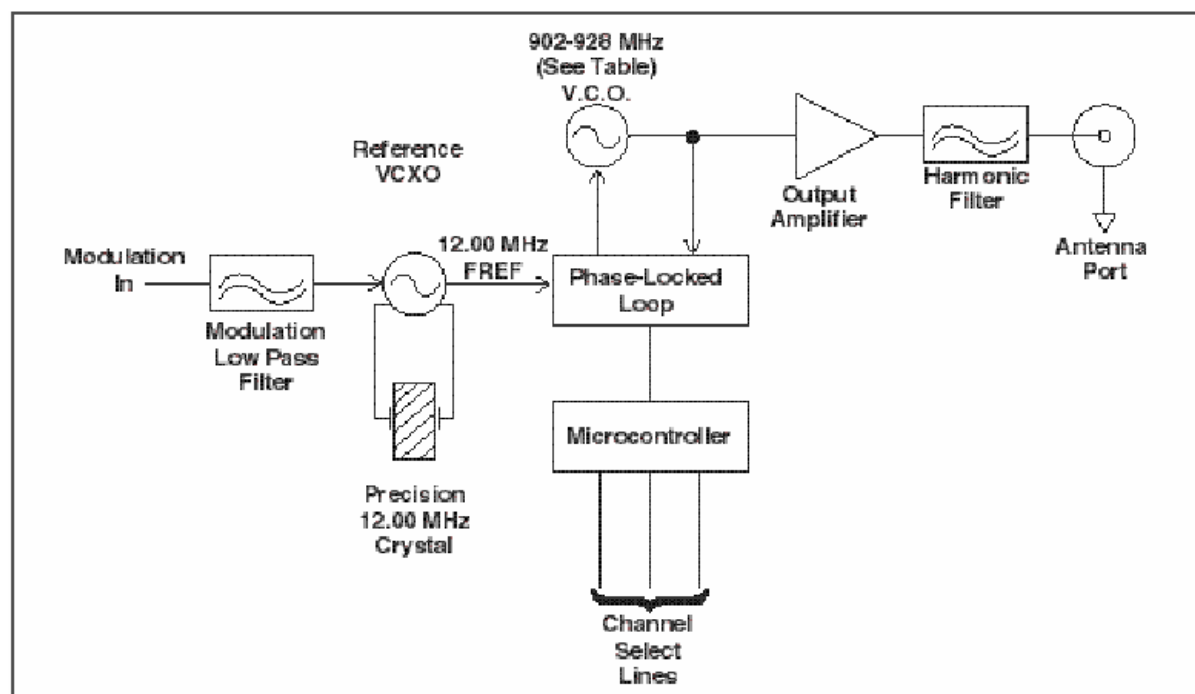
انتقال داده های دیجیتال همواره یکی از موضوعات مهم در مباحث مخابراتی بوده و میباشد. روشهای مختلفی برای انتقال دیتا وجود دارد. از جمله این روشها ارتباط رادیویی میباشد. انتقال اطلاعات در این روش و روشهایی از این دست به صورت سریال صورت میپذیرد. در اینجا به دو نوع مدولاسیون که در این گونه مدارات به کاربرده شده میپردازیم.

مدولاسیون FSK (Frequency Shift keying):

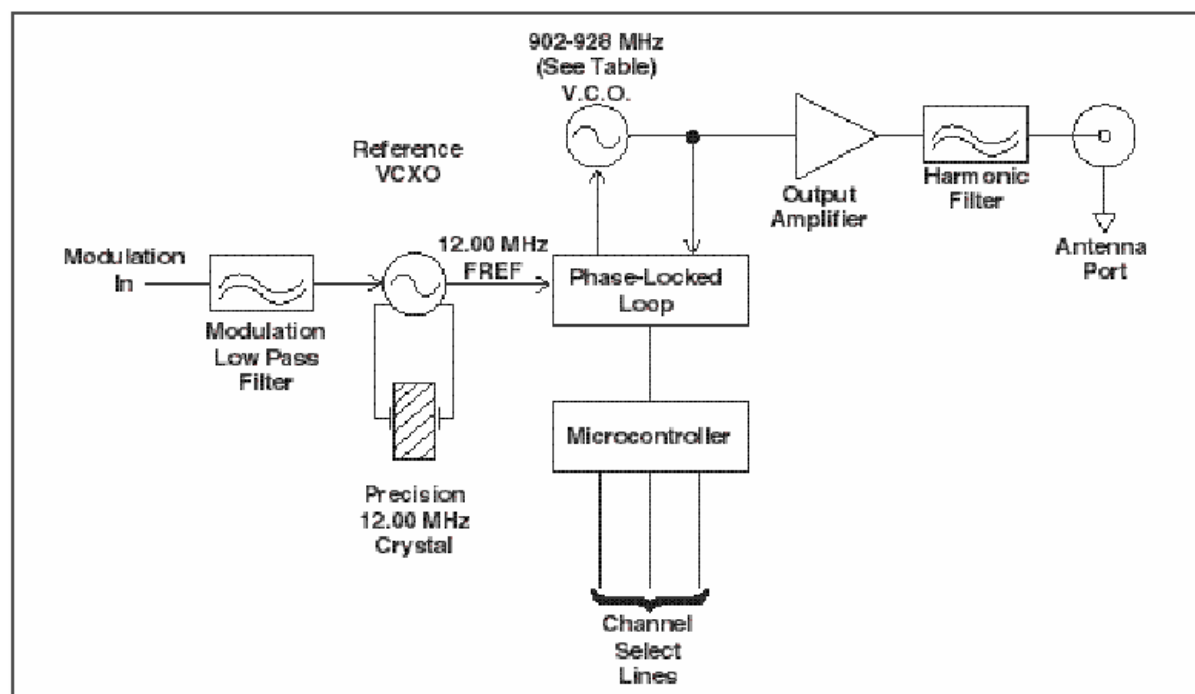
نوع ماژول استفاده شده در این مدار برای ارتباط بیسیم، از سری مدارات انتقال دیتا مبتنی بر FSK میباشند. مدولاسیون در این دسته مدارات بر اساس شیفت فرکانسی برقرار است. به این صورت که به ازای هر بیت با ارزش صفر در فرستنده نوسان مشخصی ایجاد میشود برای مثال فرکانس 45MHz برای صفر و همچنین فرکانس مشخصی برای بیت، با ارزش 1 در نظر گرفته میشود. مثلاً 50MHz.



بلوک دیاگرام بخش فرستنده :



بلوک دیاگرام بخش گیرنده :



از مزایای مدارات نوع FSK نسبت به ASK سرعت بالاتر انتقال در آن و نویز پذیری کمتر در آن می باشد همانگونه که مشاهده میشود که مدارات FSK کمی پیچیده تر از ASK می باشد. همینطور دارای قابلیت های بیشتری می باشد.

در بلوک دیاگرام مربوط به فرستنده بخش های مختلفی دیده میشود.

۱- بخش مدولاتور که از PLL تشکیل شده

۲- بخش کنترل دیجیتال که اطلاعات ورودی را گرفته و به PLL برای ایجاد فرکانس در خروجی فرمان میدهد این بخش شامل یک میکرو کنترلر می باشد .

۳- بخش تقویت کننده RF

۴- فیلتر هارمونیک و آنتن در خروجی

بخش های مختلف گیرنده به صورت زیر می باشد.

۱- بخش تقویت کننده و فیلتر RF در ورودی

۲- دو طبقه میکسر

۳- بخش اسیلاتور که شامل سینتی سایزر است و به صورت نرم افزاری میتوان آن را کنترل نمود و کانالی را انتخاب کرد

۴- در بخش خروجی نیز دمدولاتور FM قرار دارد که عموماً این مرحله به وسیله PLL انجام میگیرد در مورد مدارات FSK این موضوع را باید ذکر کرد همانطور که گفته شد این مدارات دارای کیفیتی به مراتب بیشتر از ASK می باشد.

ماژول استفاده شده در این پروژه hm-tr است که دیتا شیت آن در پیوست ارائه شده است.

ارتباط سریال

ارتباط سریال USART یکی از پروتکل‌هایی است که توسط انواع کامپیوترها نیز حمایت میشود و لذا برای برقراری ارتباط بین میکروکنترلر و کامپیوتر غالبا از این روش استفاده میشود

ارتباط سریال USART در AVR :

بخش ارتباط سریال USART در میکروکنترلرهای AVR قابلیت‌های متنوعی دارد که از جمله آنها میتوان به موارد زیر اشاره کرد:

- عملکرد full duplex (رجیسترهای سریال مستقل برای دریافت و ارسال)
- عملکرد سنکرون و آسنکرون
- عمل به صورت master و slave در حالت سنکرون
- تولید کننده نرخ ارسال (baudrate) دقیق
- حمایت از فریم‌های سریال با ۵،۶،۷،۸ یا ۹ بیت داده و ۱ یا ۲ بیت توقف
- تولید parity به صورت زوج یا فرد و امکان چک کردن سخت افزاری آن
- تشخیص خطای سر ریز و نوع فریم
- فیلتر پایین گذر دیجیتال
- تولید سه وقفه مجزا برای اتمام TX ، خالی شدن رجیستر داده TX و اتمام RX
- کار در حالت ارتباط چند پردازنده
- امکان دو برابر کردن سرعت در حالت آسنکرون

اتصال AVR به RS232

از انجایی که برای برقراری ارتباط سریال با کامپیوتر ناچاریم تا از RS232 استفاده کنیم. لازم است تا به نحوی بتونیم سطوح TTL ایجاد شده توسط میکرو و RS232 را به یکدیگر تبدیل کنیم. عموماً برای تبدیل این سطوح ولتاژ به یکدیگر از تراشه های MAX232 یا MAX233 استفاده میشود که در اینجا از تراشه MAX232 استفاده شده است. در میکرو کنترلرهای AVR دو پایه با نامهای TXD و RXD وجود دارند که از پایه TXD برای ارسال داده ها و از پایه RXD برای دریافت آنها استفاده میشود.

ارتباط با پورت سریال در MATLAB:

در نسخه های جدید نرم افزار MATLAB ، DAQ toolbox (data acquisition) بیش از پیش توسعه یافته است ، که به کمک آن میتوان به راحتی با بسیاری از مدارات جانبی ارتباط برقرار کرد که نتیجه آن ارائه مجموعه دستوراتی شئ گرا (objective) می باشد و لذا کار با آنها در مقایسه با زبانهای برنامه نویسی C، basic و... ساده تر است.

در نرم افزار matlab برای ارتباط با پورت سریال باید یک شئ تعریف گردد که وظیفه آن برقراری ارتباط است پس از تعریف شئ و تنظیم مشخصات آن با باز کردن شئ میتوان عمل رد و بدل کردن اطلاعات را آغاز نمود و در پایان نیز لازم است تا شئ مربوطه بسته شود و در صورت نیاز از محیط matlab حذف گردد.

تابع fopen (s) شئ سریال را باز میکند که در این صورت امکان دریافت و ارسال اطلاعات از طریق ارتباط سریال فراهم می گردد.

تابع `fprint()` مشابه تابع `print` در برنامه میکرو کنترلر است و یک رشته کاراکتری را از طریق ارتباط سریال ارسال میکند.

تابع `fget()` نیز شبیه تابع `gets()` در برنامه میکرو کنترلر است و یک رشته را از طریق ارتباط سریال دریافت میکند.

تابع `str2num()` یک رشته کاراکتری را به عنوان ورودی میگیرد و آن را به یک عدد تبدیل میکند.

توابع `fclose()` و `delete()` نیز به ترتیب شیء سریال را بسته و آنرا حذف میکنند.

برای ارسال و دریافت اطلاعات بر روی پورت سریال از دستور `fwrite (serial)` و `fread(serial)` استفاده میشود که این دو دستور داده ها را بصورت باینری بر روی پورت گذاشته و یا از پورت میخوانند.

شکل کامل دستور به صورت زیر است :

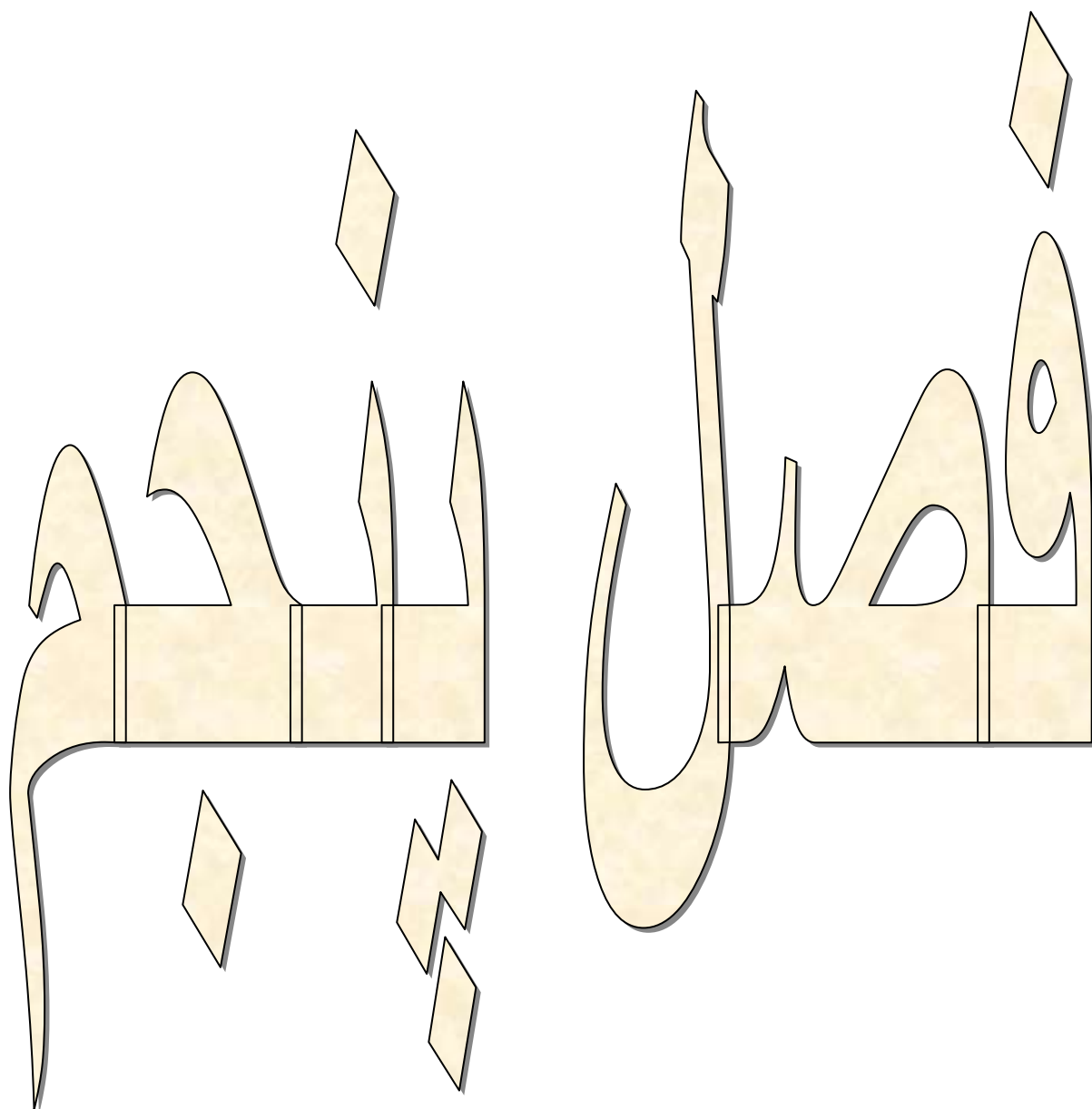
`fwrite(obj,A,'precision','mode' )`

`obj` : A serial port object.

`A` : The binary data written to the device.

`'precision'` : The number of bits written for each value, and the interpretation of the bits as character, integer, or floating-point values.

`'mode'` : Specifies whether data is written synchronously or asynchronously.



نتیجه گیری کلی و کارهای آینده

امروزه با پیشرفت روز افزون صنایع و تکنولوژی های جدید عرصه رباتیک و ماشین آلات اتوماتیک که برای انجام کارها نیازی به انسان ندارند روز به روز گسترش میابد. در این زمینه کارهای زیادی انجام گرفته است و روز به روز شاهد پیشرفت رباتیک و ماشین ابزار اتوماتیک هستیم.

خواب حضور رباتها و سیستمهای هوشمند در زندگی انسان و انجام امور توسط این ابزار آلات بی روح در حال تعبیر است. میتوان این ادعا را داشت که اگر علوم مکانیک نیز به اندازه الکترونیک پیشرفت میکرد امروزه شاهد تحول بسیار عظیمتری در این زمینه بودیم. کارهای تحقیقاتی بسیار زیادی در زمینه رباتیک در اقصا نقاط دنیا انجام گرفته و هر که سعی کرده که با ارائه مدلی بهتر و کارآمدتر گوی سبقت را از بقیه بر باید و امیدواریم که کشور عزیز ما نیز از این قافله جا نماند.

بطور حتم سیستمهای مبتنی بر بینایی ماشین در چند سال آینده بطور قابل ملاحظه ای بر زندگی ما اثر خواهند گذاشت و نقش مهمی ایفا خواهند کرد. ما تصمیم گرفتیم سیستمی طراحی و پیاده سازی کنیم تا در ابتدا خود در این زمینه تجربه ای کسب کنیم و سپس این تجربه هرچند کوچک را نیز در اختیار دیگران قرار دهیم.

در این پروژه نیز سعی شده است قدمی هرچند کوچک در این زمینه برداشته شود. سیستم ارائه شده با وجود بخشهای مختلف در زمینه مدارات الکترونیکی و سخت افزارهای مکانیکی دارای نقاط ضعف و قوت میباشد که در ذیل به برخی از آنها میپردازیم. یکی از نکات حائز اهمیت در این پروژه همان بخشهای متنوع آن میباشد که هر بخش حاوی اطلاعات مفید و

کاربردی در زمینه خود میباشد.

برای مثال در بخش ارتباط رادیویی نکات فنی و علمی بسیار زیادی وجود دارد که از آن جمله آشنایی با روشهای انتقال دیتا و همچنین مدولاسیون های FSK و ASK بود و در واقع آشنایی با این مباحث از قسمتهای مفید پروژه بود.

نقاط قوت پروژه :

۱ - طراحی روباتهایی که مبتنی بر پردازش PC میباشند برای ارتباط با کامپیوتر روشهای مختلفی وجود دارد که در تمامی آنها باید پایانه های ارتباطی به نوعی با هم متصل شوند. در سیستمهای مشابه یا باید کامپیوتر در کنار روبات باشد و یا اینکه روی آن متصل شده باشد و ارتباط این دو (Wire Less) توسط سیم برقرار میگردد. اما در سیستم ارائه شده ارتباط به صورت بی سیم انجام میگردد. با این کار سیم کشی و احتیاج به حمل کامپیوتر از ما سلب شد و این امر باعث شد که اولاً روبات سبکتر شود و در ضمن دامنه حرکت و سرعت عمل آن بیشتر شود که این موضوع بسیار حائز اهمیت است.

۲ - تغذیه این دستگاه توسط سه باتری (قابل شارژ) تأمین میگردد که به منظور شارژ مجدد آن میتوان تدبیری بر روی خود دستگاه اندیشید و میتوان آن را بارها شارژ کرد و مورد استفاده قرار داد.

3 - استفاده از روش ISP برای پروگرام کردن میکرو کنترلر که به این طریق به سادگی میتوان برنامه میکرو را چندین بار عوض کرد و امکانات جدیدی به آن افزود.

۴ - از مهمترین مزایای این دستگاه میتوان به این نکته اشاره کرد که این روبات قابلیت ارتقاء و تغییرات را به خوبی داراست. به طوری که میتوان با تغییر دادن نرم افزاری که روی کامپیوتر و تحت محیط MATLAB نوشته شده نوع کارکرد روبات را عوض کرد و انواع روباتهای مسیریاب و یا لایبرنت را طراحی کرد و همچنین میتوان با الگوریتمهای پیچیده تر اجسام با اشکال مختلف را شناسایی کرد و یا حتی الگوریتمی ایجاد کرد که بوسیله آن قادر به شناسایی چهره باشد و با توجه به این اطلاعات جهت حرکت روبات را تعیین کرد.

۵ - از دیگر نقاط قوت در این پروژه استفاده از ماژول HM-TR است که ما را از نویزهای ناشی از منابع تغذیه و امواج الکترومغناطیس مصون می دارد و این به نوبه خود مزیت خوبی به شمار میاید.

نقاط ضعف :

۱ - ضعف مکانیکی : یکی از ضعفهای موجود ضعف مکانیکی روبات میباشد. به این دلیل که از نظر مکانیکی و سخت افزاری کار جدی و تخصصی انجام نگرفته است ؛ لذا در این زمینه دارای ضعفهای قابل اصلاح میباشد. از جمله این ضعفها میتوان به اصطکاک نامناسب چرخها با زمین اشاره کرد و یا هم محور بودن چرخها با موتور که میشود با به کار گیری چرخ دنده های ترکیبی بر قدرت دستگاه افزود که خوب به دلیل کمبود امکانات و نداشتن اطلاعات لازم در این زمینه ما مجبور شدیم که چرخها را مستقیماً به شفت موتورهای متصل کنیم که این امر در برخی مواقع باعث به وجود آمدن اختلالاتی در حرکت روبات میشود.

۲ - ابراز تصویربرداری : این بخش یک دوربین به همراه فرستنده AV وجود دارد. دوربین بکار رفته از انواعی میباشد که کیفیت تصویر و رنگ در آن پایین میباشد همچنین زاویه دید دوربین در حدود ۱۶۰ درجه است که این امر محدودیت هایی مربوط به خود را ایجاد میکند. در مورد اجسام با عمقهای مختلف در تصویر نیز باید گفت به دلیل عدم Optical Zoom نمیتوان اجسام دورتر با وضوح خوب دید و این امر باعث خلل در عملکرد صحیح میشود.

۳ - عدم تشخیص فواصل به صورت مستقیم توسط تصویر دریافتی : به دلیل این که پردازش روی یک تصویر انجام میشود تصویر تنها در آن واحد از یک زاویه دریافت میشود. بنابراین نمیتوان فاصله از اجسام را به صورت طبیعی تشخیص داد که این باعث خلل در حرکت میشود.

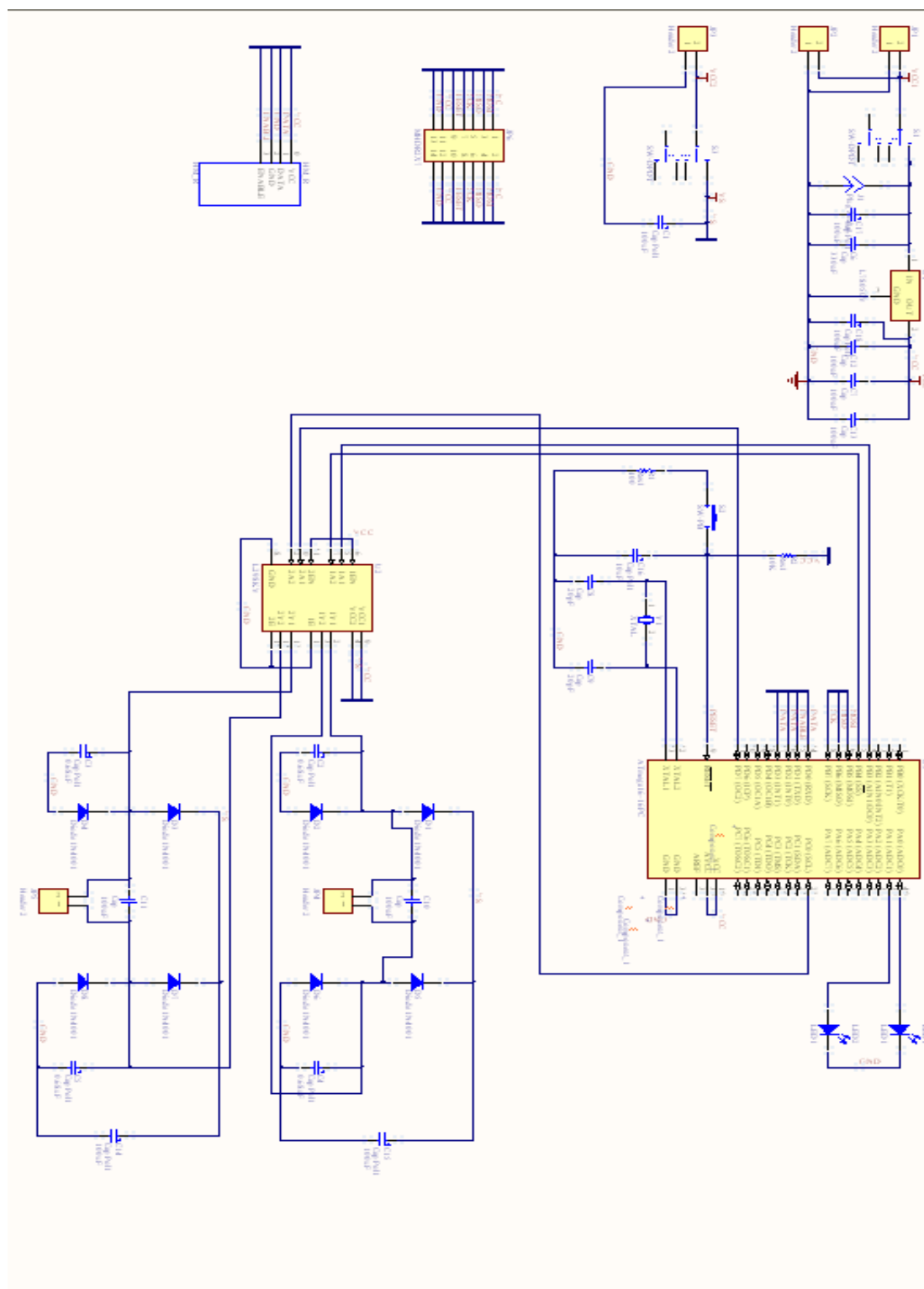
بهینه سازی و ارتقاء روبات :

با برشمردن نقاط قوت و ضعف اکنون میتوان گفت احاطه خوبی بمنظور رشد و ارتقاء پروژه بر آن یافته ایم. همچنین این موضوع را نیز قبول داریم که این طرح بمانند تمامی پروژه ها و طرحهایی از این دست دارای نواقص متعددی است. همچنین به عنوان اولین مورد میتوان به قسمتهای مکانیکی دستگاه اشاره کرد. همانطور که مشخص است و در روبات نیز مشهود است بخشهای راه انداز که شامل چرخها و بخصوص موتورها میشوند قادر به حرکت در تمامی سطوح نمیباشند. راه حلی که ما در این زمینه پیشنهاد میکنیم اینست که با توجه به عدم نیاز ما به دقت بالایی که موتورهای DC دارند میتوان از موتورهای DC (Gear Box) به همراه جعبه دنده استفاده کرد.

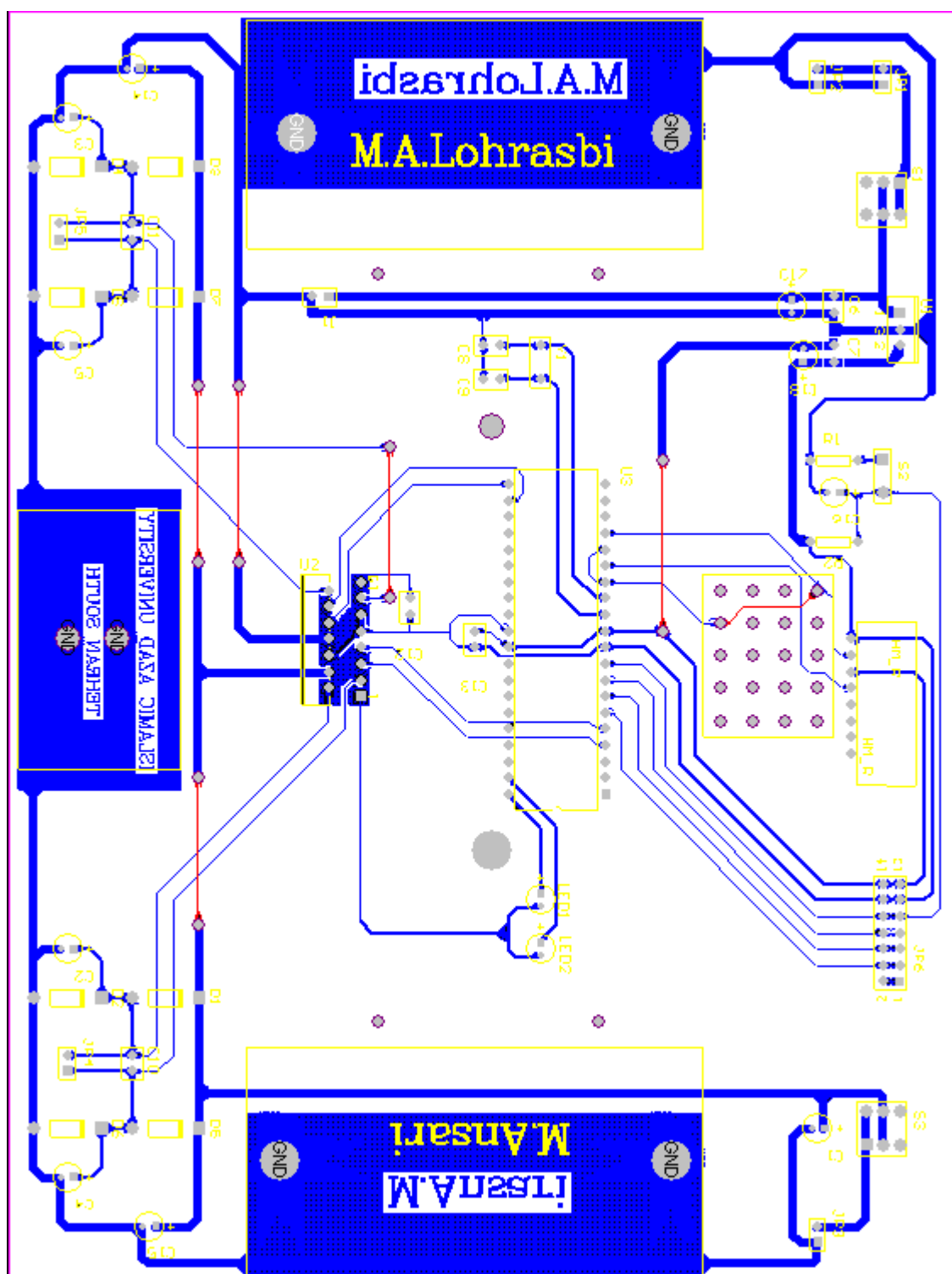
در بخش نرم افزاری میتوان با نوشتن الگوریتمهای دیگر کیفیت کار را به صورت مطلوبی بالا برد. استفاده از شبکه های عصبی یکی از این راه هاست. به کمک شبکه های عصبی میتوان برای مثال اشکال و لبه ها را تشخیص داد و همچنین هدفهای متنوعتری را دنبال کرد و بدین ترتیب کاربردهای متفاوت و پیچیده تری را برای روبات میتوان متصور شد. به کمک تغییرات نرم افزاری میتوان از این روبات ، یک روبات مسیریاب یا تعقیبگر و ... ساخت.

برای تکمیل کار و بهبود عملکرد میتوان به مطالب دیگری هم اشاره کرد که ما به طور مختصر به آن میپردازیم. میتوان با اضافه کردن بخش مکانیکی دیگری به این سیستم کارایی آنرا افزایش داد. به عنوان مثال میتوان یک بازو برای برداشتن اشیاء برای آن در نظر گرفت یا یک بخش ضربه زننده برای استفاده به عنوان روبات فوتبالیست روی آن تعبیه کرد. همچنین برای ایجاد دید سه بعدی و تشخیص عمق میتوان از دو دوربین روی روبات استفاده کرد. از موارد دیگری که میتوان نام برد پردازش سیگنال صوتی دریافتی از روبات است. میتوان برای بعضی حالات خاص از سنسورهای دیگری نیز بر روی روبات استفاده کرد. بفرض میتوان از سنسورهای التراسونیک برای تعیین فاصله از اجسام استفاده کرد.

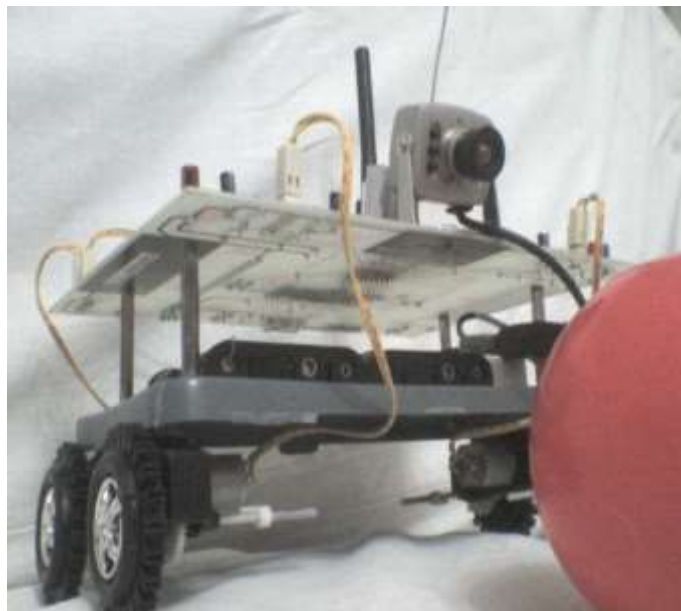
پیوست الف : طرح شماتیک مدارات استفاده شده در روبات



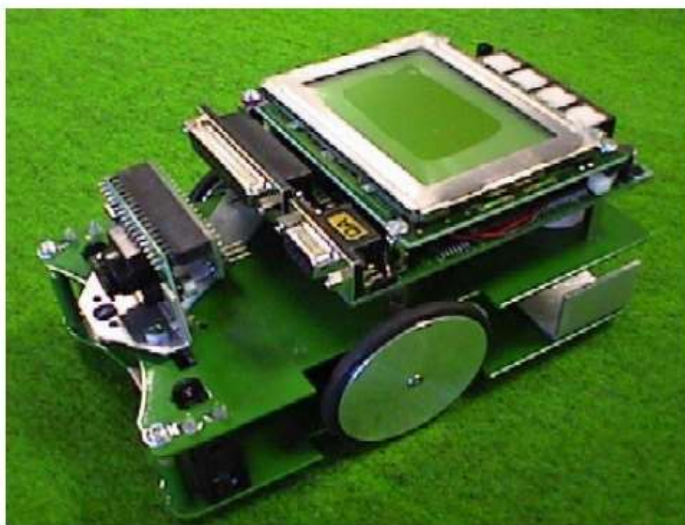
پیوست ب : مدار pcb



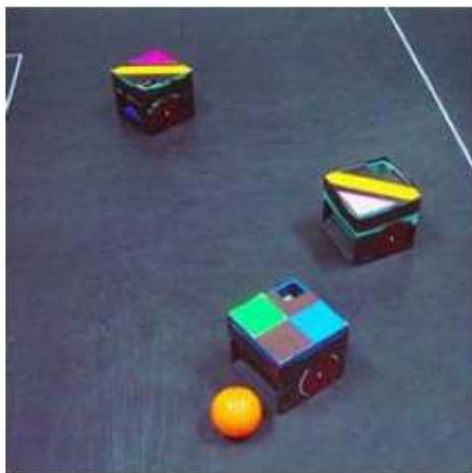
پیوست ج: نمائی از روبات



پیوست د : اشکال مختلفی از روباتهای هوشمند



الف- روبات فوتبالیست مجهز به دوربین

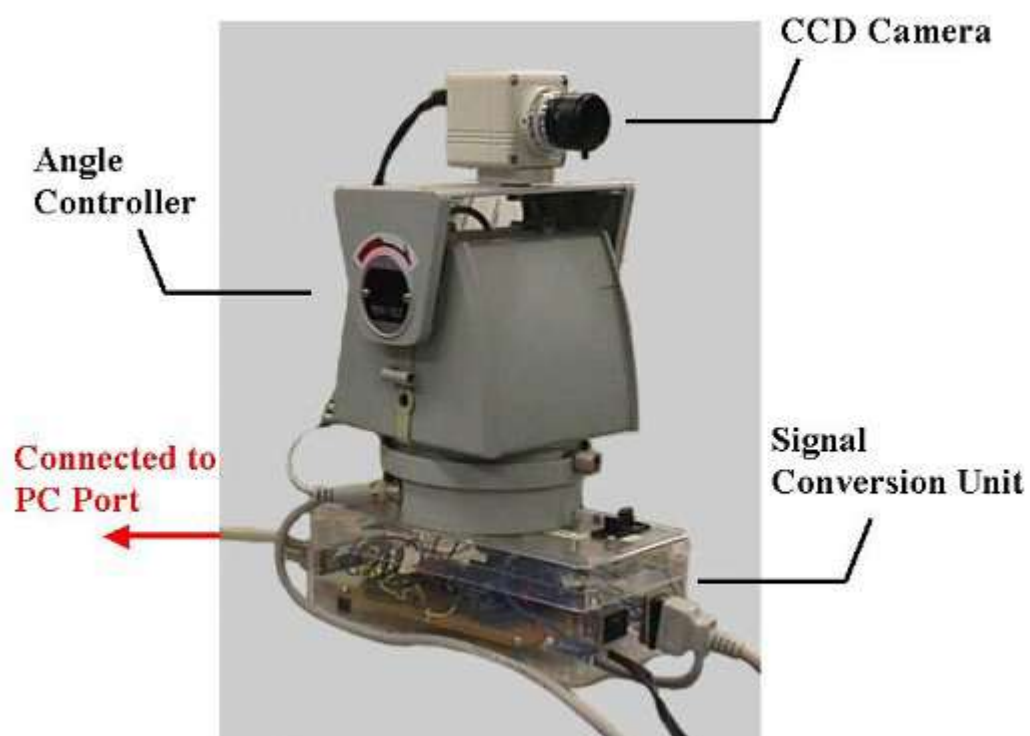


ج- مینی روباتهای فوتبالیست



ب- روبات متحرک اتوماتیک

از دیگر نمونه ها میتوان از روباتهای شبیه ساز حرکت چشم و دوربینهای هوشمند نام برد.



پیوست ۵: برنامه های استفاد شده در پروژه

- برنامه نوشته شده برای انجام عمل پردازش تحت محیط Matlab

```
close all;
clear all;
delete all;
scale=0.1;
i1=0; i2=0; j1=0 ; j2=0;
v=0; dv=0; lv=0; rv=0;
loop=0;
mincolor=uint8(30);           %0<minicolor<255   % red pixel color %
ID =int8(25);                  %0<ideal Diameter<72  ideal diameter in the fixed point
LES=-6.75;                     % line equation slope %
k1=3;                          % k2 %
s=serial('com5','Terminator',[]);
fopen(s);
vid=videoinput('winvideo',1,'YUY2_720x576');
vid.selectedsource='composit';
source=vid.source;
src=source(1);
src.brightness=1000;
src.contrast=3500;
src.saturation=7000;
src.analogvideoformat='pal_b';
set(vid,'ReturnedColorSpace','rgb');
preview(vid)
while 1
a=getsnapshot(vid);           % 3*3 MATRICE %
b=imresize(a,scale);         % 3*3 MATRICE %
```

```

c=rgb2gray(b);           % 2*2 MATRICE %
d=(b(:,1)-c);           % RED SCALE 2*2 MATRICE %
[n,m]=size(d);          % n= ROWS , m= COLUMNS %
i1=0; i2=0; j1=0 ; j2=0;
for j=1:m
    for i=1:n
        if d(i,j)>mincolor
            i1=i;
            j1=j;
            break
        end
    end
    if d(i,j)>mincolor ,break , end
end
for j=m:-1:1
    for i=n:-1:1
        if d(i,j)>mincolor
            i2=i;
            j2=j;
            break
        end
    end
    if d(i,j)>mincolor ,break , end
end
oi=int8((i1+i2)./2);
oj=int8((j1+j2)./2);

if (oi==0), oi=1; end
D=0;
for j=1:m

```

```

if d(oi,j)>mincolor
D=D+1;
end
end
if (oj==0)                                % IF OBJECT IN THE FRAME %
    lv=50;
    rv=-50;
    loop=loop+1;
    if (loop>10)                            %FIND OBJECT BY TURN AROUND %
        lv=0;
        rv=0;
    end
elseif (j1<5 )                            %IF OBJECT IN THE LEFT SIDE OF THE FRAME %
    lv=50;
    rv=-50;
    loop=0;
elseif (m-j2<5)                            %IF OBJECT IN THE RIGHT SIDE OF THE FRAME%
    lv=-50;
    rv=50;
    loop=0;
else                                        %OBJECT COMPLETELY IN THE FRAME %
    v=LES*(D - ID);
    dv=oj-m./2;
    lv=v+k2*dv;
    rv=v-k2*dv;
    loop=0;
end
fwrite(s,lv,'schar');                      % 'schar' BINARY STRING BETWEEN -128 ta +127 %
fwrite(s,rv,'schar');
end

```

• برنامه جهت پروگرام کردن میکرو کنترلر AVR :

Project : ALMA robotic

Chip type : ATmega16

Program type : Application

Clock frequency : 11.059200 MHz

Memory model : Small

External SRAM size : 0

Data Stack size : 256

```
#include <mega16.h>
```

```
#define RXB8 1
```

```
#define TXB8 0
```

```
#define UPE 2
```

```
#define OVR 3
```

```
#define FE 4
```

```
#define UDRE 5
```

```
#define RXC 7
```

```
#define FRAMING_ERROR (1<<FE)
```

```
#define PARITY_ERROR (1<<UPE)
```

```
#define DATA_OVERRUN (1<<OVR)
```

```
#define DATA_REGISTER_EMPTY (1<<UDRE)
```

```
#define RX_COMPLETE (1<<RXC)
```

```
// USART Receiver buffer
```

```
#define RX_BUFFER_SIZE 8
```

```
char rx_buffer[RX_BUFFER_SIZE];
```

```
unsigned char rx_wr_index,rx_rd_index,rx_counter;
```

```
// This flag is set on USART Receiver buffer overflow
```

```
bit rx_buffer_overflow;
```

```
Bit rdxok=0, left_or_right=0;
```

```
// USART Receiver interrupt service routine
```

```

#pragma savereg-
interrupt [USART_RXC] void uart_rx_isr(void)
{
char status,data;
#asm
    push r26
    push r27
    push r30
    push r31
    in  r26,sreg
    push r26
#endasm
status=UCSRA;
data=UDR;
if ((status & (FRAMING_ERROR | PARITY_ERROR | DATA_OVERRUN))==0)
{
    rx_buffer[rx_wr_index]=data;
    if (++rx_wr_index == RX_BUFFER_SIZE) rx_wr_index=0;
    if (++rx_counter == RX_BUFFER_SIZE)
    {
        rx_counter=0;
        rx_buffer_overflow=1;
    };
};
#asm
    pop r26
    out sreg,r26
    pop r31
    pop r30
    pop r27
    pop r26
#endasm

```

```

rdxok=1;
}

#pragma savereg+
#ifndef _DEBUG_TERMINAL_IO_
// Get a character from the USART Receiver buffer
#define _ALTERNATE_GETCHAR_
#pragma used+
char getchar(void)
{
char data;
while (rx_counter==0);
data=rx_buffer[rx_rd_index];
if (++rx_rd_index == RX_BUFFER_SIZE) rx_rd_index=0;
asm("cli")
--rx_counter;
asm("sei")
return data;}
#pragma used-
#endif

// Standard Input/Output functions
#include <stdio.h>
#include <delay.h>
// Declare your global variables here


void main(void)
{
// Declare your local variables here
signed char temp=0;
// Input/Output Ports initialization
// Port A initialization
// Func0=Out Func1=Out Func2=In Func3=In Func4=In Func5=In Func6=In Func7=In
// State0=0 State1=0 State2=T State3=T State4=T State5=T State6=T State7=T
PORTA=0x00;

```

```
DDRA=0x03;
```

```
// Port B initialization
```

```
// Func0=In Func1=In Func2=In Func3=Out Func4=Out Func5=In Func6=In Func7=In
```

```
// State0=T State1=T State2=T State3=0 State4=0 State5=T State6=T State7=T
```

```
PORTB=0x00;
```

```
DDRB=0x18;
```

```
// Port C initialization
```

```
// Func0=Out Func1=In Func2=In Func3=In Func4=In Func5=In Func6=In Func7=In
```

```
// State0=0 State1=T State2=T State3=T State4=T State5=T State6=T State7=T
```

```
PORTC=0x00;
```

```
DDRC=0x01;
```

```
// Port D initialization
```

```
// Func0=In Func1=In Func2=In Func3=In Func4=In Func5=In Func6=In Func7=Out
```

```
// State0=T State1=T State2=T State3=T State4=T State5=T State6=T State7=0
```

```
PORTD=0x00;
```

```
DDRD=0x80;
```

```
// Timer/Counter 0 initialization
```

```
// Clock source: System Clock
```

```
// Clock value: 11059.200 kHz
```

```
// Mode: Fast PWM top=FFh
```

```
// OC0 output: Non-Inverted PWM
```

```
TCCR0=0x69;
```

```
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;
OCR1AH=0x00;
OCR1AL=0x00;
OCR1BH=0x00;
OCR1BL=0x00;

// Timer/Counter 2 initialization
// Clock source: System Clock
// Clock value: 11059.200 kHz
// Mode: Fast PWM top=FFh
// OC2 output: Non-Inverted PWM
ASSR=0x00;
TCCR2=0x69;
TCNT2=0x00;
OCR2=0x00;

// External Interrupt(s) initialization
// INT0: Off
// INT1: Off
// INT2: Off
GICR|=0x00;
MCUCR=0x00;
MCUCSR=0x00;

// Timer(s)/Counter(s) Interrupt(s) initialization
TIMSK=0x00;

```

```

// USART initialization
// Communication Parameters: 8 Data, 1 Stop, No Parity
// USART Receiver: On
// USART Transmitter: Off
// USART Mode: Asynchronous
// USART Baud rate: 9600
UCSRA=0x00;
UCSRB=0x90;
UCSRC=0x86;
UBRRH=0x00;
UBRRL=0x47;

// Analog Comparator initialization
// Analog Comparator: Off
// Analog Comparator Input Capture by Timer/Counter 1: Off
// Analog Comparator Output: Off
ACSR=0x80;
SFIOR=0x00;

// Global enable interrupts
#asm("sei")

while (1)
{
    // Place your code here
    PORTA.0=1;
    PORTB.4=0;
    OCR0=255;
    PORTC.0=0;
    OCR2=255;
    PORTA.1=1;
    delay_ms(400);

```

```

PORTB.4=1;
OCR0=0;
PORTC.0=1;
OCR2=0;
PORTA.1=0;
delay_ms(400);
PORTB.4=0;
PORTC.0=0;

while (1) {
if (rdxok==1){
PORTA.1=1;
temp=getchar();
if (temp==--128) {temp=-127;}
if (temp>-30 & temp<30) { temp=0;}
if (left_or_right==0) { if (temp>=0) {TCCR0=0x69; PORTB.4=0;
OCR0=temp*2; left_or_right=1;}
else { temp=temp*-1; TCCR0=0x79; PORTB.4=1;
OCR0=temp*2; left_or_right=1;}
}

else { if (temp>=0) { TCCR2=0x69; PORTC.0=0;
OCR2=temp*2; left_or_right=0;}
else {temp=temp*-1; TCCR2=0x79; PORTC.0=1;
OCR2=temp*2; left_or_right=0;}
}

PORTA.1=0;
rdxok=0;
}
}
//
};
}

```

پیوست ی : دیتا شیت قطعات مورد استفاده

ATMEGA16L •

L298 •

HM-TR •

L7805 •

منابع :

- 1- digital image processing by gonzales
- 2- matlab image acquisition help
- 3- matlab image processing help

۴- میکرو کنترلرهای AVR . نویسنده امیر ره افروز

- 5- internet ebooks