هو الحق

پروژه ربات مسیریاب

- سید امیررضا علوی
- سبحان امینی
- استاد راهنما : دکتر شهبازی

گروه مکانیک دانشگاه اصفهان

زمستان ۹۶

مقدمه

درطول تدریس درس مبانی برق ۱ ما با مبانی مدارهای دیجیتال و بوردهایی برنامه پذیر آ شنا شدیم که قابلیت کنترل سیستمهای مختلف را دارا هستند؛ مدار هایی که مبتنی بر میکرو کنترلر های AVR , PIC , ARM هستند. طراحی سخت افزاری و نرم افزاری این سیستم های کنترل کننده مورد بررسی قرار گرفت. یکی از بردهای برنامه پذیر که بر مبنای میکروکنترلر های متنوع در حال حاضر با قابلیت های مختلف و در حال توسعه مورد استفاده است Arduinoنام دارد. به دلیل راحتی گسترده بودن ابزار های مورد نیاز این بورد ، کار با دردسر کمتری برای استفاده کنندگان میسر است. در ادامه مثال هایی از کار با Arduino فرا گرفته شد و تصمیم به ساخت یک **ربات مسیریاب** گرفته شد.

مقدمات طراحی و ساخت

برای ساخت و طراحی این سیستم مکاترونیکی می توان از مدار ها و سنسور ها و عملگرهای مختلف بهره برد. برای ساخت مدار کنترل کننده میتوان از میکروکنترلر های AVRو ARM یا مدار های منطقی متشکل از گیتهای منطقی و یا بورد های Arduino استفاده کرد. برای سنسور های تشخیص خط میتوان از سنسورهای اشعه مادون قرمز و یا تشخیص رنگ و یا حتی دوربین و پردازش تصویر بهره برد. برای سیستم حرکتی نیز میتوان از چرخ یا مکانیزم های دیگر حرکتی و موتور های DC ,Servo , Stepper استفاده کرد. در کنار این اجزای اصلی نیز می توان قابلیت های ارتباط با گجت های دیگر و ارسال اطلاعات خط و مسیر را به سیستم افزود.

ما در طراحی خود تصمیم به استفاده از بوردهای Arduino وموتورهای ساده DC و سنسورهای گیرنده مادون قرمز به همراه شیلد درایو موتور و یک راه ارتباط نزدیک Bluetooth گرفتیم. در ادامه قطعات الکترونیکی و مکانیکی سیستم آورده شده است:

- Arduino UNO
- L۲۹۳D motor control shield
- HC۰۵ Bluetooth module
- پنج عدد فرستنده و گیرنده مادون قرمز
- دو عدد موتور DCبه همراه گیربکس
- چهار عدد چرخ
- یک عدد برد بورد

- یک عدد بلندگو ۳۲ اهم
- سیم های رابط و تعدادی مقاومت ۲۰
- یک عدد شاسی

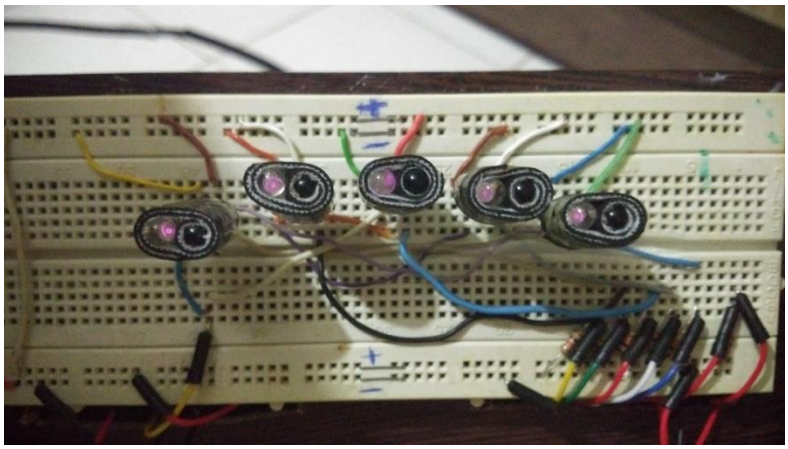
برای بخش نرم افزاری نیز برنامه Arduino نیاز است که خود بخش مهمی از ساخت است و به عنوان هوش مصنوعی و تصمیم گیرنده سیستم به کار گرفته می شود. از برنامه رسمی Arduinoبرای برنامه نویسی و programming استفاده شده است. در ادامه کار برای ارتباط سیستم با موبایل(برای مثال اندروید) از طریق Bluetooth از برنامه های کاربردی مبتنی بر ارتباط سریال در میکروکنترلر AVR استفاده شده است.

طراحی و ساخت بدنه و سخت افزار

وظیفه ا صلی دریافت داده های سنسور و تحلیل و بعد د ستور به موتور ها بر عهده Arduino و برنامه پروگرم شده بر روی آن است. اکنون به بررسی کلی مدار الکترونیکی می پردازیم.

برد Arduinoشامل پورت های ورودی آنالوگ به دیجیتال برای کاربرد سنسورها و پورت های مشخص PWM برای کاربرد خروجی دیجیتال به آنالوگ برای ارتباط با درایو موتور است.

پورتهای A۱تا A۵ برای پنج گیرنده مادون قرمز هستند. در کنار هر گیرنده مادون قرمز یک فرستنده مادون قرمز نیز تعبیه شده ا ست که در حالت نزدیک به زمین در صورت اح ساس خط و رنگ سیاه با مقاومت زیاد و فرستادن ولتاژ پایین اطلاعات گرفته می شود و در صورت احساس رنگ سفید با مقاومت پایین و فرستادن ولتاژ بالا عمل میکنند. پنج عدد جفت فرستنده و گیرنده IR در جلوی سیستم نصب میشوند.



بیشتر پایه های دیجیتال Arduinoبرای استفاده شیلد درایو که کاملاً بر روی Arduino UNO قرار میگیرند اشغال شده است که این امر سبب میشود دسترسی به آن پورتها سخت یا غیر ممکن شود. شیلد درایو L۲۹۳D یک درایو موتور با قابلیت کنترل چهار موتور DC یا دو سروو یا دو استپر موتور است که توسط کتابخانه adafruit motor به نام AFMotor.h در برنامه نویسی قابل استفاده است. این درایو به همراه کتابخانه مخصوص برای کنترل دو موتور در برنامه در سرعتهای مختلف و جهت جلو و عقب قابل استفاده است. برای نمونه از پینهای دیجیتال ۱۱ و ۳ و ۵ و ۶ برای کنترل سرعت DC Motor و ۴و۷و۸و۱۲ برای تعیین جهت موتورها ا ستفاده میشوند. تمام ارتباط با درایو توسط د ستوزاتی ساده شده در کتابخانه مخصوص موجود ا ست. روی درایو چهار عدد خروجی موتور موجود است که ما از موتورهای ۳و۴ استفاده کردیم.

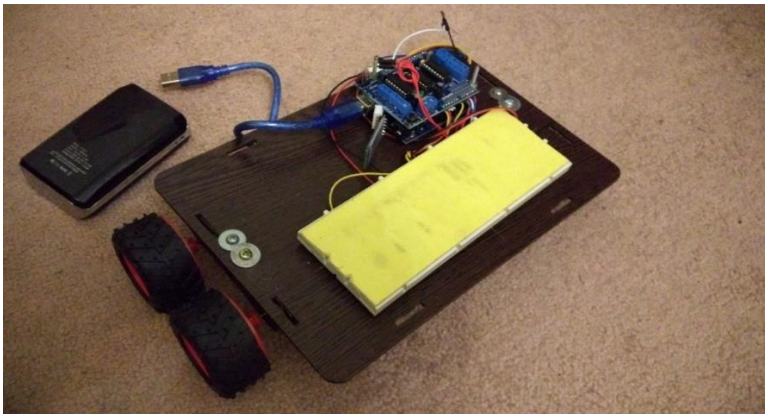
از پینهای دیجیتال ۰ و ۱ به عنوان RX , TX تحت ارتباط سریال ا ستفاده شده ا ست. این دو پین به دو پین RX , TX مدول بلوتوث مت صل می شود که میتوان از طریق ارتباط سریال(تنظیم شده روی ۹۶۰۰ bps) تحت یک نرم افزار اندروید مبتنی بر ارتباط سریال به آن دستوراتی داده شود. از این طریق میتوان با ارتباط

نزدیک دستورات حرکتی یا تنظیم حالت یا کالیبراسیون سنسورها و یا ارسال اطلاعات ذخیره شده مسیر بعد از طی مسافت به گجت اندروید را فراهم نمود.

یک عدد بلندگوی ۳۲ اهم نیز به پین۱۱ متصل شده است که برای مشخص کردن حالت های حرکتی و کمک به کنترل کننده و اعلام رویداد ها با صدای کم به کار میرود. از صداهای مختلفی برای زمانهای گوناگون استفاده شده است. لازم به ذکر است که در برنامه کامپیوتری از کتابخانه های PCM.h و pitches.h استفاده شده است.

برای باتری مدار نیز از دو باتری ۵و ۱۲ و یا ۵و ۵ استفاده میکنیم. یکی برای استفاده سنسور ها و Arduino و یکی برای استفاده درایو موتور. میدانیم که درایو موتور ما ولتاژ های ۴,۵ تا ۳۶ را برای استفاده موتور پشتیبانی میکند ولی از آنجا که موتور های مورد استفاده ۵ هستند ما از دو منبع تغذیه مجزای ۵ استفاده کردیم. مجزا به این دلیل که باعث ایجاد نویز در مدار نشود و همچنین در آینده امکان اتصال باتری قویتر با موتور قویتر وجود داشته باشد.

برای سوار کردن قطعات به روی شاسی نیز برد Arduino را روی شاسی نصب کردیم به نحوی که پوسته درایو بر آن کاملاً سوار شد.ماژول بلوتوث به همراه تغذیه آن از مدار جاسازی شد. جفت های فرستنده و گیرنده IR را روی بردبورد با سیم های تغذیه و مقاومت ها قرار دادیم و بردبورد را زیر شاسی چسباندیم تا به سطح زمین نزدیکتر باشد. در ادامه موتورها به دو سمت شاسی متصل شدند. بلندگو به پین ۱۱ و زمین متصل شد. دیواره های شاسی را برای استحکام و پایداری باتری داخل آن متصل کردیم. در آخر کار از یک عدد پاور بانک ساده با دو عدد خروجی ۵ برای تغذیه کل استفاده شد.



لازم به ذکر است در پیوست شماتیک مدار الکترونیکی آورده شده است.

نرم افزار و نوشتن برنامه کامپیوتری

از برنامه رسمی Arduino برای برنامه نویسی استفاده شده است. این محیط به ما امکان برنامه نویسی شی گرا و استفاده از کتابخانه های مختلف را برای راحتی کار میدهد.

در برنامه از تعدادی تابع ساخته شده توسط خود و کتابخانه ها استفاده شده است. تابع هایی نظیر دستورات جلو و عقب و چپ و راست و چرخش به راست و چرخش به چپ برای عملگر و تابع هایی برای خواندن حالت سنسورها و تابعهایی برای چاپ اطلاعات در سریال و تابع هایی برای ایجاد صدا و ملودی.

همچنین دو شی موتور تحت کتابخانه AFMotor ساخته شده است که کار با موتورها را ساده میکند.

در طول برنامه سعی شده با دادن توضیحات(comment) دستور ها و متغیر ها و نوشتن مرتب برای درک بهتر شرطها و حلقه ها و قسمتهای مختلف و همچنین استفاده از تابع های ساده کننده برای کاهش حجم کدنویسی این بستر فراهم شود تا در آینده امکان توسعه برنامه به کاربردهای پیشرفته تر یا دقیق تر وجود داشته باشد.

در برنامه روشهای تسهیل کار درایو موتور و بهینه سازی ذخیره سازی اطلاعات مسیر به کار گرفته شده است که کار های اصلی را بهبود میبخشند. برای مثال وقتی سرعت یک موتور از ساعتگرد به پادساعتگرد عوض شود یک ولتاژ القایی در یک لحظه کوتاه رد موتور پدید می آید که نیاز به مکث کوتاه (فقط در آن لحظه) است. این مکثهای لحاظ شده در برنامه (soft working) نامیده شده است. در ادامه به بررسی نحوه عملکرد این روشها می پردازیم.

برنامه کامپیوتری در بیش از ۷۷۰ خط نوشته شده است(با توجه به این که از تابع های زیادی برای کاهش حجم برنامه در IDE استفاده شده است این مقدار در صورت استفاده نکردن از تابعها به بیش از ۱۰۰۰ خط میرسید که این زیادی خود به دلایل زیادی دردسر زا است). این برنامه حدود ۹۹ درصد حافظه فلش میکروکنترلر را اشغال میکند(البته برای پخش صدایی با کمک کتابخانه PCM در بلندگو حدود ۶۰ درصد حافظه اشغال میشود که اگر این بخش تقریبا فانتزی(!) را که شامل یک دستور و یک متغیر حجیم حاوی صدا است حذف کنیم حدود ۴۰ در صد حافظه فلش پر میشود). این برنامه از ٪۷۳ حافظه دینامیک برای متغیر ها استفاده میکند. بخش اعظم این حافظه را متغیر voyage میگیرد که شامل دو آرایه ۱۷۰ تایی از حرکتهای ذخیره شده و زمان آن در طول تعقیب خط است. این اطلاعات ذخیره شده را میتوان با یک دستور طراحی شده در سریال چاپ کرد و میتوان با طراحی یک برنامه موبایل این اطلاعات را تبدیل به **نمود گرافیکی شکل مسیر** کرد.

از حافظه EEPROMنیز استفاده شده است. با تعریف دو تابع setdef و getdef امکان ذخیره و بازخوانی مقدار دیده شده سفید برای پیشفرض زمینه فراهم شده است. با توجه به این که امکان writeوerase تنها ۱۰۰۰۰ بار درArduino امکان پذیر است این دستورات باید کمتر مورد استفاده قرار گیرند.

از تعدادی define نیز در برنامه برای ساده سازی دسترسی به اعداد پینها و کالیبراسیون دیدن خط سیاه(مانند rrnes) و حالتها(تحت عنوان متغیر mood) استفاده شده است.

برنامه نوشته شده شامل سه بخش کلی است؛ یکی بخش تعریف متغیر ها و تعریف توابع و یکی بخش setup() و یکی بخش loop().

در **بخش اول** بعد از include های کتابخانه های مورد استفاده و define ها به تعریف متغیر ها میرسیم. در کنار اکثر متغیر ها کامنت آورده شده است. برای نمونه چند متغیر پرکاربرد در برنامه را می آوریم:

- دو شی موتور با نامهای mol و mor چپ و راست ایجاد شده اند .
- moodنام متغیری است که پس از هربار حرکت جلو یا چرخش به راست یا .. مقداری برابر با آن حرکت به خود میگیرد و برای ذخیره سازی اطلاعات مسیر (voyage)و همچنین برای تسهیل در کار درایو و موتور ها (soft working) به کار میرود. متغیر prvmood نیز حالت قبلی تر را ذخیره میکند.
- Voyage۱[voyagelimit] , voyage۲[voyagelimit] ذخیره کننده اطلاعات مسیر طی شده هستند. مولفه ۱ تعداد هر بار اجرا شدن تابع حرکت به جلو یا ... را ذخیره میکند و مولفه دوم نام آن حرکت خاص را و وقتی حرکت عوض شد مقدار voyagehouse یک مقدار اضافه میشود تا به خانه ی بعدی آرایه ها رویم. برای تبدیل کردن تعداد اجرا شدن هر حرکت یعنی مولفه اول به زمان بر حسب میلی ثانیه از متغیرهای setstartوsetstop استفاده شده است.
- ثابت sample برای ذخیره سازی ملودی حجیم پخش شونده از بلندگو ایجاد شده است.
- متغیر mode حالت سیستم را مشخص میکند. در صورت داشتن مقدار صفر دستگاه در حالت drive mode میرود و با مقدار یک در حالت line following mode میرود.
- متغیر seri برای ذخیره سازی کد ASCII کاراکتر فر ستاده شده به کار میرود تا د ستور دریافتی اجرا شود.

تابع های تعریف شده در این بخش نیز به این قرار اند:

- تابع های setbeep() , racebeep() , music() برای ایجاد صداهای اعلام کننده تعریف شده اند.

- تابع های getdef , setdef برای ذخیره سازی مقدار پیشفرض دیدن سفید که توضیح داده شد.
- تابع ()getsens که شماره پین آنالوگ را میگیرد و مقدار ولتاژ حس شده را میدهد .
- توابع حرکتی ()driveFو()driveStopو()driveB و ()driveR و ()deriveRR و ()driveL و ()driveLL و ()driveTurnR و ()driveTurnL که هرکدام با شـــدت و جهت خاصـــی موتورها را به حرکت در می آورند.
- توابع ()drive و ()drive۱ که با کمک همدیگر در تعقیب خط دستورات حرکت را با توجه به دیدن یا ندیدن خط توسط سنسورها به موتورها میدهند. نمونه ای از دستورات دیدن یا ندیدن خط توسط پنج سنسور را در ادامه میبینید.

```
if(!rr && !r && m && !l && !ll) {                // forward
  mode=۲;
  ret=۰;
}
 if(rr && !r && !m && !l && !ll){                // right
  mode=۰;
  ret=۰;  //   number of previous same modes RESET
 }
 else if(!rr && !r && !m && !l && ll) {                // left
  mode=٤;
  ret=۰;
 }
 else if (!rr && r && !m && !l && !ll){                // right
  mode=۱;
  ret=۰;
 }
 else if(!rr && !r && !m && l && !ll) {                //left
  mode=۳;
  ret=۰;
 }
```

```
else if(rr && r && m && l && ll) {                    // in AIR !
  driveB();
  driveStop();
  ret=۰;
  mode=-۱;
  delay(۱۰); }
else {                  //see nothing!  Go on
  delay(۱);
  ret++;
}
```

در **بخش ()setup** دســتوراتی نظیر تعیین ورودی و خروجی پینها و دســتوراتی که یکبار اجرا میشــوند مانند اجرای صدای بوت و تست اولیه موتور و بررسی زده شدن کلید مکانیکی در ابتدای کار آمده اند.

در **بخش ()loop** دستوراتی که مکرر اجرا میشوند و مقادیری که مجدد اندازه گیری میشوند آورده شده است.

در هربار اگر در سریال چیزی ار سال شده بود برر سی میکند و به آن د ستور عمل میکند. بسته به مقدار یک و صفر متغیر mode که توضیح داده شد کاری صورت میپذیرد. در ادامه لیست دستورات ارسالی در سریال آورده شده است.

کــاراکــتــر ارسالی	u	d	l	r	s
کد ASCII	۱۱۷	۱۰۰	۱۰۸	۱۱۴	۱۱۵
دستور	جلو	عقب	چپ	راست	ایست

کــاراکــتــر ارسالی	c	i	k	b	a	t
کد ASCII	۹۹	۱۰۵	۱۰۷	۹۸	۹۷	۱۱۶
دستور	رفتن به حالت رانندگی	نــمــایــش اطــلاعــات مسافت طی شده	رفــتــن بــه حالت تعقیب خط	بوق	وضــــعــیــت سنسورها	تعیین حد سنسورها

هزینه تمام شده

هزینه تمام شده برای همه اجزا غیر از باتری و شاسی ۹۰ هزار تومان شد.

Arduino	L۲۹۳D	HC۰۵	دو موتور	چهار چرخ	گیرنده و فرستنده IR	سیمها و مقاومتها	بردبورد	بلندگو
۲۳	۱۲	۱۸	۱۰	۱۰	۵	۱۰	۳	۲

در صورت اضافه کردن شاسی و باتری(پاوربانک □□□۷۰۰۰) هزینه تا ۱۳۰هزار تومان برآورد میگردد.

نتیجه گیری

پس از پروگرم کردن برنامه کامپیوتری روی Arduino و رفع اشکالهای پیاپی در تست عملکرد حالتهای مختلف مخصوصاً تعقیب خط، به جمع بندی و رفع اشکالهای جزئی پرداختیم.

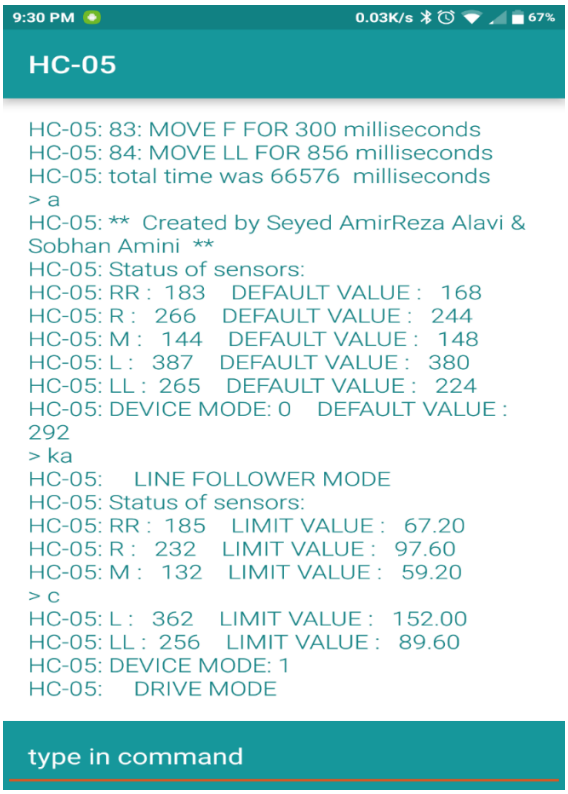
به عنوان حاصل کار یک سیستم مکاترونیکی طراحی و ساخته شد که به عنوان ماشین کنترل شونده و یا ماشین تعقیب خط به انتخاب کاربر میتواند حرکت کند. این تعویض بین حالت ها توسط دستورات ارسالی به ماژول بلوتوث و یا توسط کلید ساده تعبیه شده در سیستم امکان پذیر است.

قابلیت های ارسال اطلاعات مسیر و انطباق با محیط های تعقیب خط به سیستم افزون شده است.

از مهمترین دستاورد های این پروژه در درس مبانی برق، آشنایی با کاربردهای سیستم های کنترلی برنامه پذیر و مهارت های برنامه نویسی است.

ایده هایی برای توسعه

برای برنامه کاربردی ارسال دستورات به بلوتوث از برنامه های موجود در وب استفاده کردیم. نمونه ای از محیط ارتباط سریال در برنامه کاربردی موبایل آورده شده است.



میتوان با کار بیشتر با استفاده از سورس برنامه های موجود و کمک از کتابخانه های موجود، برنامه ای مخصوص این سیستم ساخت تا روی موبایل نصب شود و قابلیت های نمایش ساده تر وگرافیکی تر اطلاعات دریافتی از سیستم را دارا باشد. زیرا قابلیتهای ارسال اطلاعات مسیر و سنسورها در برنامه پروگرم شده Arduino موجود است. برای مثال میتوان قابلیت **نمود گرافیکی شکل مسیر** را به برنامه کاربردی افزود. اگرچه این کار نیازمند دانش برنامه نویسی موبایل است اما زمینه ای برای انجام یک پروژه بین رشته ای جذاب است.