

به نام خدا

عنوان:

گزارش آزمایش نهایی - ربات تعقیب خط

اعضای گروه:

فرید خدادادی

عرفان صبحایی روی

استاد آزمایشگاه:

استاد محمد لالی



دانشگاه خوارزمی

چکیده

در این آزمایش قصد داریم با استفاده از دانش کسب شده در آزمایشات گذشته و ترکیب این اطلاعات، یک ربات تعقیب خط را با استفاده از آردوینو طراحی کرده و سپس با استفاده از تجهیزات مورد نیاز به صورت عملی پیاده‌سازی کنیم. در نهایت با استفاده از ترکیب داده‌های دریافتی از سنسورها و استفاده از آنها جهت تنظیم سرعت موتورها توسط برنامه‌ی آردوینو، می‌توانیم ربات را طراحی و هدایت کنیم. در نهایت ما موفق شدیم رباتی طراحی کنیم که مسیر مسابقه را در زمان ۱۰.۵۰ ثانیه طی کند.

مقدمه و معرفی

این آزمایش از چهار مرحله مختلف تشکیل شده است.

در بخش اول لازم است تا شاسی ربات را سر هم کنیم. برای این کار از کیت ربات مدل ZK-2 استفاده شده است. در این کیت مجموعه‌ای از وسایل مورد نیاز جهت سر هم بستن و راه‌اندازی ربات به همراه آموزش اتصال قطعات وجود دارد.

پس از اتمام بخش اول لازم است با استفاده از بردبورد و یک قطعه L293D، سیمکشی‌های لازم برای عملکرد چرخ‌های ربات را انجام دهیم. این بخش شامل اتصالات مربوط به L293D می‌باشد و نیاز به وصل کردن باتری و سیم متصل به موتورها به پین‌های مربوطه روی L293D وجود دارد.

در مرحله سوم لازم است که آردوینو نیز وارد مدار شود. در این مرحله پین‌های مربوط به کنترل حرکت موتورها از آردوینو به L293D متصل می‌شوند. همچنین در این مرحله سنسورها را به جلوی ربات متصل می‌کنیم و آنها را نیز وارد مدار می‌کنیم. اتصال سنسورها به آردوینو نیز در این مرحله انجام می‌شود.

در مرحله آخر هنگامی که اتصالات کامل شد، کافیت برنامه‌ی مناسب ربات تعقیب خط نوشته شده و روی آردوینو بارگزاری شود.

تجهیزات مورد استفاده

برای آزمایشات از ابزارهای زیر استفاده شده است:

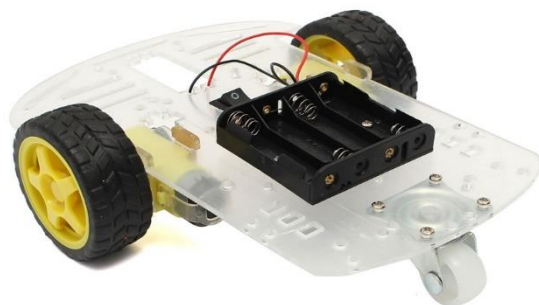
- یک عدد Arduino Uno
- نرم افزار Arduino IDE
- کیت شاسی ربات ZK-2

- سه عدد سنسور IR
- دو عدد باتری کتابی به همراه جای باتری کتابی
- دو عدد موتور به همراه چرخ
- یک عدد L293D
- یک عدد breadboard
- چسب دو طرفه
- سیم به تعداد کافی

روش آزمایش و نتایج

بخش اول

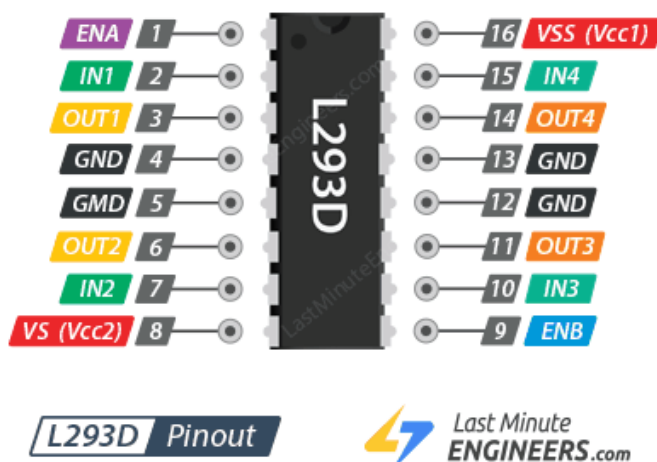
در این بخش می‌خواهیم به سر هم کردن شاسی ربات بپردازیم. همانطور که قبلاً اشاره کردیم، برای این پروژه از کیت ربات ZK-2 استفاده شده است. در این کیت دو عدد موتور و دو عدد چرخ نیز وجود دارد که در دو طرف ربات قرار می‌گیرند. همچنین یک چرخ با قابلیت چرخش آزاد و ۳۶۰ درجه برای حفظ تعادل و حرکت بهتر ربات قرار در قسمت عقب ربات قرار داده شد. همچنین برای منبع تغذیه یک عدد جای باتری برای ۴ عدد باتری قلمی ۱.۵ ولتی قرار داده شده است. یک کلید برای قطع و وصل کردن منبع تغذیه نیز در این کیت وجود دارد. با استفاده از دفترچه راهنمای موجود در بسته ربات، قطعات مربوطه رو به هم متصل می‌کنیم. لازم به ذکر است که در این پروژه برای فراهم کردن ولتاژ بالاتر برای چرخ‌ها و همچنین ایجاد فضای بیشتر، از منبع تغذیه درون کیت استفاده نشده است و به جای آن از یک عدد باتری کتابی ۹ ولتی استفاده کردیم. شکل ۱ ربات کامل شده را نشان می‌دهد.



شکل ۱- شاسی ربات

بخش دوم

در این بخش لازم است که با کاربرد و پین‌های L293D آشنا باشیم. این قطعه‌ی الکترونیکی وظیفه‌ی کنترل سرعت، جهت حرکت و انتقال و افزایش ولتاژ موتورها را بر عهده دارد. پین‌های L293D را در شکل ۲ مشاهده می‌کنید.



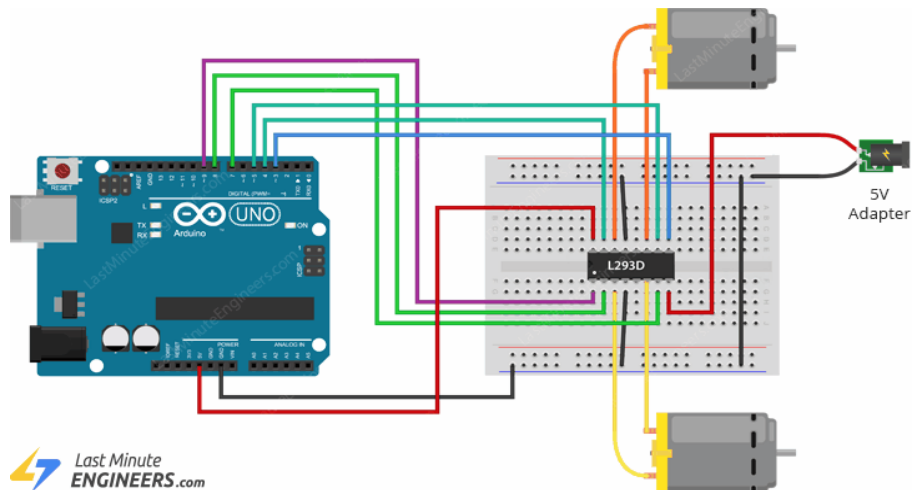
شکل ۲- پین‌های L293D

حالا به بررسی این پین‌ها می‌پردازیم:

- پین‌های ۴، ۵، ۱۳ و ۱۲: این‌ها پین‌های Ground هستند و باید به منبع تغذیه در مدار متصل شوند. علت وجود این پین‌ها کنترل دمای قطعه است.
- پین‌های ۱ و ۹: این پین‌ها جهت فعالسازی موتورها استفاده می‌شوند و می‌توان از آن‌ها با استفاده از دادن ورودی PWM سرعت موتورها را نیز کنترل کرد. از آنجایی که ما قصد داریم که موتورها همواره فعال باشند و قصد محدود کردن سرعت آن‌ها را نداریم، این دو پین به سیم مثبت منبع تغذیه وصل شده‌اند تا همواره ولتاژ ۵ ولت دریافت کنند و مقدار ۱ دیجیتالی داشته باشند.
- پین‌های ۳، ۶، ۱۱ و ۱۴: این پین‌ها باید به سیم‌های موتورها متصل شوند. سیم‌های یک موتور به پین‌های ۳ و ۶، و سیم‌های موتور دیگر باید به پین‌های ۱۱ و ۱۴ متصل شوند.
- پین‌های ۲، ۷، ۱۰ و ۱۵: این پین‌ها به Arduino متصل می‌شوند و برای فعالسازی و جهت حرکت موتورها استفاده می‌شوند.
- پین ۱۶: این پین باید مقدار ۵ ولت دریافت کند و در واقع برای کار کردن و فعال سازی L293D استفاده می‌شود.

- پین ۸: این پین می‌تواند مقداری بین ۴.۵ تا ۳۶ ولت دریافت کند و این ولتاژ را به موتورهای جهت حرکت آن‌ها انتقال می‌دهد. سرعت حرکت ربات بسیار وابسته به میزان ولتاژ دریافتی در این پین است. به همین جهت پیشنهاد می‌شود از یک منبع تغذیه خارجی و مستقل از مدار برای این پین استفاده شود.

در این مرحله طبق شکل ۳ سیم‌ها را متصل می‌کنیم.



شکل ۳- اتصالات L293D

ما در این پروژه از پین ۵ ولت و GND آردوینو به عنوان منبع تغذیه برای L293D استفاده کردیم. برای افزایش سرعت موتورهای نیز از یک باتری کتابی ۹ ولت استفاده کردیم و آن را به پین ۸ L293D متصل کردیم. لازم به ذکر است که برای روشن کردن خود آردوینو نیز از یک باتری کتابی ۹ ولت دیگر استفاده شده است. بدین منظور کافی است که سیم مثبت باتری را به پین Vin آردوینو و سیم منفی آن را به یکی از پین‌های GND وصل کنیم.

بخش سوم

در این مرحله کفایت که سنسورها را به ربات وصل کرده و آن‌ها را به مدار و آردوینو اضافه کنیم. برای اتصال سنسورها از چسب دو طرفه در بخش جلویی ربات استفاده شده است. سه سنسور باید با فاصله‌ی نسبتاً کم به یکدیگر در جلوی ربات قرار بگیرند. یک سنسور دقیقاً در وسط و رو به جلو، و دو سنسور دیگر در دو طرف سنسور اولی قرار می‌گیرند. توجه داشته باشید که فاصله سنسورها از یکدیگر بسیار در عملکرد و سرعت حرکت ربات تاثیرگذار است و لازم است این فاصله کم باشد. پس از اتصال سنسورها به بدنه‌ی ربات، باید پین‌های VCC و gnd آن‌ها را در مدار به منبع تغذیه متصل کنیم. همچنین پین A0 سنسورها را به پین‌های A0 تا A2 در

آردوینو متصل می‌کنیم. در نهایت باید با تست روی خط سیاه و با یک پیچ گوشتی چهار سو، حساسیت سنسورها را تنظیم کنیم تا خط را به خوبی تشخیص دهند.

بخش چهارم

این مرحله در عین سادگی، به نوعی مهمترین مرحله‌ی این پروژه محسوب می‌شود و توانایی ربات در تشخیص مسیر حرکت را مشخص می‌کند. در این بخش به بررسی کد ربات تعقیب خط به صورت مرحله به مرحله می‌پردازیم.

```
int right_wheel = 9;
int in2 = 10;
int left_wheel = 3;
int in4 = 5;

int sensor_right = A0;
int sensor_left = A1;
int sensor_mid = A2;
```

در ابتدای برنامه لازم است تا ثابت‌ها و درواقع شماره پین‌های ورودی و خروجی آردوینو را مشخص کنیم. همانطور که در کد مشخص است، چرخ راست به پین شماره ۹ و چرخ چپ به پین شماره ۳ متصل است. دقت کنید که این پین‌ها در واقع به پین‌های IN در L293D متصل هستند و حرکت موتورها به سمت جلو را کنترل می‌کنند. پین ۹ و ۱۰ آردوینو هم به همین صورت به L293D متصل هستند، لکن به خاطر اینکه حرکت موتورها را در جهت عقب کنترل می‌کنند و ما کلاً نیازی نداریم که چرخ‌ها به عقب بچرخند، نیازی به استفاده از آن‌ها در برنامه نیست. در ادامه پین‌های مربوط به سنسورها نیز تعریف شده است.

```
void setup() {
  pinMode(sensor_right, INPUT);
  pinMode(sensor_left, INPUT);
  pinMode(sensor_mid, INPUT);
  pinMode(right_wheel, OUTPUT);
  pinMode(in2, OUTPUT);
  pinMode(left_wheel, OUTPUT);
  pinMode(in4, OUTPUT);
  delay(500);
}
```

در بخش **setup** کد ورودی و خروجی‌ها را مشخص می‌کنیم. سنسورها در واقع ورودی ما هستند و خروجی ما مقادیری هستند که برای حرکت چرخ‌ها تعیین می‌کنیم. در پایان بخش **setup** یک تاخیر کوچک قرار گرفته است تا از اختلالاتی که ممکن است به علت تاخیر در روشن شدن سنسورها به وجود می‌آید جلوگیری شود.

(مثلا اگر سنسور سمت چپ حتی چند میلی ثانیه زودتر از دیگر سنسورها روشن شود، ربات به اشتباه تشخیص می‌دهد که باید به سمت راست بچرخد در صورتی که ممکن است ربات روی خط صاف قرار گرفته باشد)

```
void loop() {  
  
    int status_rs = 255 - analogRead(sensor_right);    موقعیت سنسور راست  
    int status_ls = 255 - analogRead(sensor_left);     موقعیت سنسور چپ  
    int status_ms = 255 - analogRead(sensor_mid);      موقعیت سنسور وسط
```

حالا وارد بخش loop برنامه می‌شویم. در این بخش در ابتدا مقادیر ورودی را دریافت می‌کنیم. مقدار status نشان می‌دهد که یک سنسور دارد خط (رنگ سیاه) می‌بیند یا نه. اگر مقدار status بیشتر از ۱۲۸ باشد، یعنی سنسور خطی مشاهده نمی‌کند و اگر این مقدار کمتر از ۱۲۸ باشد یعنی خط سیاه توسط سنسور دیده شده است.

```
if (status_rs >= 128 && status_ls >= 128 && status_ms >= 128)  
{  
    analogWrite(right_wheel, 255);  
    analogWrite(left_wheel, 255);  
}  
else if (status_rs >= 128 && status_ls >= 128 && status_ms <= 128) {  
    analogWrite(right_wheel, 255);  
    analogWrite(left_wheel, 255);  
}  
else if (status_rs <= 128 && status_ls >= 128) {  
    while (status_ms >= 128){  
        analogWrite(right_wheel, 0);  
        analogWrite(left_wheel, 255);  
        status_ms = 255 - analogRead(sensor_mid);  
    }  
}  
else if (status_rs >= 128 && status_ls <= 128){  
    while (status_ms >= 128){  
        analogWrite(right_wheel, 255);  
        analogWrite(left_wheel, 0);  
        status_ms = 255 - analogRead(sensor_mid);  
    }  
}  
else {  
    analogWrite(right_wheel, 0);  
    analogWrite(left_wheel, 0);  
}  
}
```

در این بخش از کد به کنترل جهت حرکت ربات بر اساس مقادیر دریافتی از سنسورها میپردازیم. در اینجا ۵ شرط را چک می‌کنیم و بر اساس آن‌ها حرکت چرخ‌ها را کنترل می‌کنیم.

- شرط اول: اگر هر سه سنسور خطی مشاهده نکردند (مقدار status هر سه سنسور بیشتر از ۱۲۸ بود) هر دو چرخ به سمت جلو حرکت کنند. این حالتی است که خطی وجود ندارد یا قطعی در مسیر بوجود آمده است.
- شرط دوم: اگر سنسورهای چپ و راست خط را مشاهده نکردند و سنسور وسط مشاهده کرد (مقدار status سنسور چپ و راست بیشتر از ۱۲۸ و برای سنسور وسط کمتر از ۱۲۸ بود)، هر دو چرخ با هم حرکت کنند. این شرطی است که ربات را روی خط صاف پیش می‌برد.
- شرط سوم: سنسور راست خط را مشاهده کرده و سمت چپ مشاهده نکرده است. در این صورت از یک حلقه‌ی while استفاده می‌کنیم و چک می‌کنیم که تا وقتی سنسور وسط خط سیاه را مشاهده نکرده است، چرخ چپ بچرخد و چرخ راست ثابت بایستد. با این کار ما مطمئن می‌شویم که وقتی سنسور راست خط را دید، ربات با چرخاندن چرخ چپ به تنهایی، به سمت راست می‌چرخد و در هنگام تغییر جهت چرخ به سمت راست درست عمل می‌کند. علت وجود حلقه while این است که عمل چرخش به راست تا زمانی ادامه داشته باشد که سنسور وسط خط را ببیند. این یعنی ربات کامل چرخیده است و حالا باید مسیر مستقیم را طی کند.
- شرط چهار: این شرط مشابه شرط قبلی است با این تفاوت که چرخش به سمت چپ را کنترل می‌کند.
- شرط پنجم: در صورتی که حالتی غیر از حالات بالا وجود داشت، ربات بایستد و چرخ‌ها نچرخند. (مثلاً اگر در پایان خط مسابقه یک خط افقی سیاه وجود داشته باشد، هر سه سنسور همزمان خط سیاه را می‌بینند و ربات می‌ایستد).

منابع

- [1] "Last Minute Engineers," [Online]. Available: <https://lastminuteengineers.com/l293d-dc-motor-arduino-tutorial/>. [Accessed 11 12 2022].