

بسم الله الرحمن الرحيم



نام و نام خانوادگی : امیر محمد شفیعی

استاد مربوطه : جناب آقای دکتر عسگری

گزارش سمینار ساخت ربات مسیر یاب

درس : رباتیک

تاریخ ارسال : 1403/11/05

چکیده

در این پروژه، ساخت یک ربات مسیر یاب با استفاده از پلتفرم **Arduino** انجام شد. هدف اصلی این پروژه طراحی و پیاده‌سازی یک ربات خودران قادر به حرکت در مسیرهای از پیش تعیین شده با استفاده از سنسورهای مادون قرمز برای شناسایی خطوط و موتورهای گیربکس برای حرکت است. ربات مسیر یاب با استفاده از یک برد **Arduino Uno SMD CH340** و ماژول درایور **L298N** طراحی و کدنویسی شده است. این پروژه شامل مراحل همچون طراحی سخت‌افزار ربات، پیاده‌سازی سیستم‌های تشخیص خط، کنترل حرکت موتور و نوشتن کد برنامه‌نویسی با استفاده از محیط **Arduino IDE** است. در این گزارش، ابتدا به معرفی اجزای مختلف ربات و عملکرد آن پرداخته می‌شود، سپس الگوریتم‌ها و روش‌های مورد استفاده در کنترل حرکت ربات شرح داده می‌شود. در نهایت، نتایج آزمایش‌ها و ارزیابی عملکرد ربات در محیط‌های مختلف توضیح داده شده و چالش‌های موجود در روند پیاده‌سازی بررسی می‌شود.

سپاسگذاری

در ابتدا، از خداوند بزرگ و مهربان به خاطر توفیق و فرصتی که برای انجام این پروژه فراهم آورد، سپاسگزارم. از جناب آقای دکتر عسگری، استاد محترم این درس، که با راهنمایی‌های علمی و حمایت‌های بی‌دریغ خود، بنده را در انجام این پروژه یاری رساندند، صمیمانه قدردانی می‌کنم. نکات و توصیه‌های ارزشمند ایشان در تمام مراحل پروژه، به ویژه در زمینه طراحی و پیاده‌سازی ربات، نقش بسزایی در پیشبرد این کار داشت. همچنین، از دستیار آموزشی عزیز، جناب آقای مهندس محسنی، برای راهنمایی‌ها و کمک‌های فنی‌شان در این پروژه کمال تشکر را دارم. تلاش‌های بی‌وقفه ایشان در ارائه توضیحات دقیق و پاسخ به سوالات، باعث شد تا به سرعت موانع پیش رو را برطرف کرده و پروژه به نحو احسن تکمیل شود. از تمامی کسانی که در طول این مسیر به من کمک کرده‌اند، چه به صورت مستقیم و چه به طور غیرمستقیم، سپاسگزارم. با احترام و امید به آینده‌ای روشن و موفق تر

امیرمحمد شفیعی

فهرست مطالب

5	مقدمه تعریف مسئله.....
6	شرح روش‌های کنترل حرکت ربات.....
7	تشریح قطعات و ساخت ربات.....
9	نحوه ساخت ربات.....
13	کد نویسی
15	نحوا کار در محیط آردوینو
17	بررسی کد اجرایی
21	منابع.....

مقدمه

هدف این پروژه ساخت یک ربات مسیر یاب است که بتواند مسیر مشخصی را با استفاده از سنسورهای مادون قرمز دنبال کند. این ربات بر پایه **Arduino** طراحی شده و از ماژول درایور **L298N** برای کنترل موتورهای استفاده می‌کند. عملکرد آن بر مبنای تشخیص خطوط تیره و روشن است که به کمک سنسورهای تعبیه شده، موقعیت خود را شناسایی کرده و حرکت می‌کند.

این پروژه شامل مراحل طراحی، ساخت و برنامه‌نویسی است. در بخش سخت‌افزاری، انتخاب قطعات مانند سنسورها، موتورهای و برد اصلی بر اساس نیاز عملکردی انجام شد. در بخش نرم‌افزاری، الگوریتم‌هایی برای شناسایی مسیر و کنترل حرکت توسعه داده شد. این ربات می‌تواند در محیط‌های مختلف آزمایش شود و برای کاربردهای صنعتی یا آموزشی توسعه یابد.

تعریف مسئله

ربات‌های مسیر یاب برای حرکت در مسیرهای مشخص و تعیین شده طراحی می‌شوند. این ربات‌ها معمولاً برای شناسایی و دنبال کردن خطوطی که مسیر را مشخص می‌کنند، از سنسورهای نوری یا مادون قرمز استفاده می‌کنند. هدف اصلی در این پروژه، طراحی و ساخت یک ربات مسیر یاب است که بتواند با استفاده از سنسورهای مادون قرمز خطوط تیره روی زمینه روشن را تشخیص دهد و در طول مسیر حرکت کند. برای حل این مسئله، از سخت‌افزارهایی نظیر برد **Arduino Uno SMD CH340**، ماژول درایور **L298N**، سنسورهای مادون قرمز و موتورهای گیربکس استفاده شده است. مهم‌ترین بخش این ربات شناسایی دقیق خطوط و خارج نشدن از مسیر هست.

شرح روش‌های کنترل حرکت ربات

در این پروژه‌ها، ربات مسیر یاب از الگوریتم‌های ساده اما کارآمد برای کنترل حرکت خود استفاده می‌کند. این الگوریتم‌ها بر اساس داده‌هایی هستند که از سنسورهای مادون قرمز برای شناسایی خطوط مسیر دریافت می‌شود.

استفاده از سنسورهای مادون قرمز برای تشخیص خط: ربات شما از سنسورهای مادون قرمز برای شناسایی تفاوت بین سطح تیره (خط) و سطح روشن (زمینه) استفاده می‌کند.

الگوریتم کنترل حرکت بر اساس سنسورها: در این حالت، فرض بر این است که ربات شما حداقل دو سنسور مادون قرمز (یکی برای سمت چپ و دیگری برای سمت راست) دارد:

حالت 1: ربات روی خط قرار دارد: اگر هر دو سنسور خط را تشخیص دهند، ربات به حرکت مستقیم ادامه می‌دهد.

حالت 2: ربات از خط خارج شده است: اگر یکی از سنسورها خط را تشخیص ندهد، ربات به سمت همان سنسور که خط را از دست داده، می‌چرخد تا دوباره به خط برگردد.

حالت 3: ربات در پیچ یا انحراف است: اگر یکی از سنسورها در یک سمت خط را تشخیص دهد و دیگری نه، ربات به سمت سنسوری که خط را شناسایی کرده، می‌چرخد تا مسیر خود را ادامه دهد.

تشریح قطعات و ساخت ربات

برد: Arduino Uno SMD CH340

- نقش: این برد به عنوان پردازشگر مرکزی ربات عمل می‌کند و وظیفه اجرای کدهای برنامه‌نویسی و کنترل سایر قطعات را بر عهده دارد.
- اتصال: برد Arduino به سایر قطعات مانند سنسورها، موتورها و ماژول درایور از طریق پایه‌های دیجیتال و آنالوگ متصل می‌شود.

ماژول درایور: L298N

- نقش: این ماژول برای کنترل حرکت موتورها استفاده می‌شود L298N. قادر به کنترل جهت و سرعت دو موتور DC است.
- اتصال: پایه‌های IN1، IN2، IN3 و IN4 به پایه‌های دیجیتال برد Arduino وصل می‌شوند تا جهت حرکت موتورها را کنترل کنند. همچنین، پایه‌های VCC و GND به برق و زمین متصل می‌شوند. پایه‌های OUT1 و OUT2 و OUT3 و OUT4 به موتورها متصل می‌شوند.

موتور DC و گیربکس:

- نقش: موتورها برای حرکت دادن چرخ‌های ربات و انجام عملیات چرخش و حرکت در مسیر استفاده می‌شوند. گیربکس به موتور متصل است تا سرعت و قدرت چرخش بهینه‌تر باشد.
- اتصال: دو موتور DC به ماژول درایور L298N متصل می‌شوند و این موتورها به چرخ‌های ربات متصل هستند.

سنسورهای مادون قرمز: (IR)

- **نقش:** سنسورهای مادون قرمز برای شناسایی خطوط مسیر (معمولاً خطوط تیره روی زمینه روشن) استفاده می‌شوند. این سنسورها تغییرات شدت نور را از سطح زیر ربات دریافت کرده و به پردازنده ارسال می‌کنند.
- **اتصال:** سنسورهای IR به پایه‌های آنالوگ یا دیجیتال برد Arduino متصل می‌شوند و اطلاعات مربوط به تشخیص خط را به برد ارسال می‌کنند.
منبع تغذیه (باتری):
- **نقش:** منبع تغذیه برای تأمین انرژی مورد نیاز برای عملکرد برد Arduino، موتورها و سنسورها استفاده می‌شود.
- **اتصال:** باتری به ورودی VCC برد Arduino و همچنین به مازول درایور L298N برای تغذیه موتورها متصل می‌شود.
اتصالات و کابل‌ها:
- **نقش:** برای اتصال تمامی قطعات به یکدیگر از سیم‌ها و کابل‌های مختلف استفاده می‌شود. این اتصالات به انتقال سیگنال‌ها و تغذیه برق بین قطعات کمک می‌کنند.
- **اتصال:** سیم‌ها از طریق پایه‌های مناسب در Arduino، مازول درایور L298N، سنسورها و موتورها به یکدیگر متصل می‌شوند.

نحوه ساخت ربات

در ابتدا ما تمام وسائل های مورد نیاز که جناب آقای محسنی لیست کرده بودند تهیه شد و روز ساخت ربات همگی در دانشگاه حضور پیدا کردیم و ربات مورد نظر را ساختیم.

در ابتدا با آنالیز بخش های مختلف ربات تصمیم گرفتیم که قطعات باید به گونه ای تنظیم شود که ربات دچار مشکل نشود و دینامیک ربات برقرار باشد.

قبل از شروع ساخت ربات به دیدن چند ویدیو و خواندن چند مقاله درباره ربات مسیر یاب پرداختیم که با عملکرد بهتری به ساخت این ربات بپردازیم.

بعد از مشخص شدن جایگاه های قطعات شروع به اسمبل کردن آن کردیم. در ابتدا موتور های گیربکسی را از طریق پیچ و مهره (یا از طریق چسب حرارتی) به بدنه متصل کردیم. در مرحله بعد تایرها را به موتور گیربکسی وصل میکنیم. از موتور های گیربکسی (از هر کدام) 2 تا سیم داریم که باید به پایه های 1, 2, 3, 4 out درایور وصل کنیم.

3 ماژول مادون قرمز رو به جلوی ربات پیچ میکنیم که فرستنده ها توانایی تشخیص مسیر را داشته باشند. پشت آن درایور را قرار میدهیم و بعد از برد آردوینو. بعد از برای برقراری تعادل و حرکت بهتر ربات یک یا دو هرزگرد به بدنه پیچ میکنیم. (تعداد هرزگرد بستگی به نوع طراحی و جایگذاری قطعات دارد). در این مرحله که بخش مهم ساخت هست، به اتصالات میپردازیم.

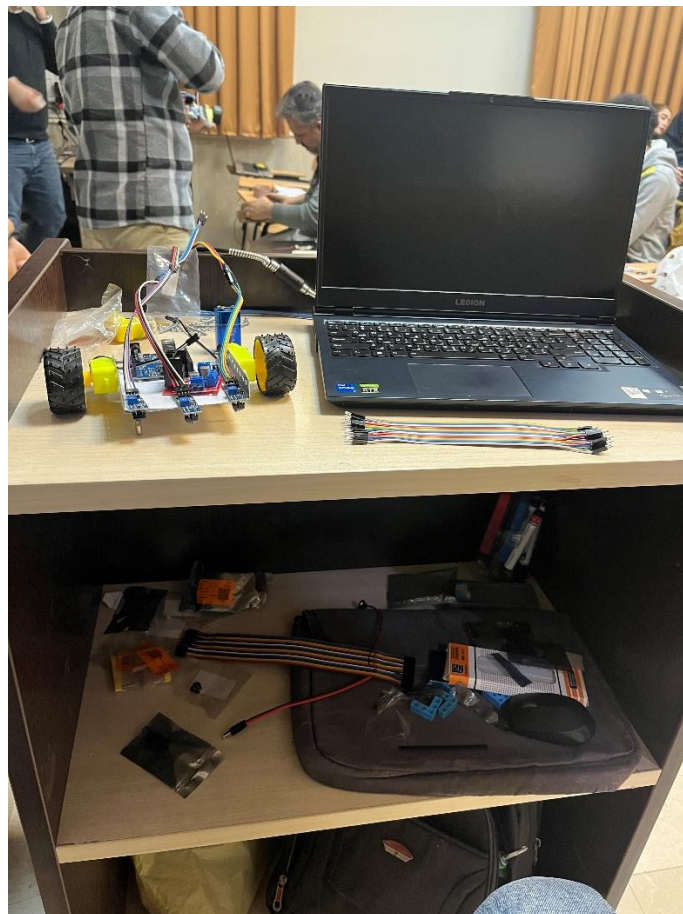
هر ماژول مادون قرمز یک پایه A0 دارد. که این پایه ها را باید به قسمت آنالوگ مادربرد وصل کنیم. در این پروژه بنده A0 مایول سمت راست را به A2، وسطی را به A1 و سمت چپی را به A0 وصل کردم. یکی دیگر از پایه های ماژول مادون قرمز GND هست. که این پایه ها را باید با سیم به هم وصل کنیم،

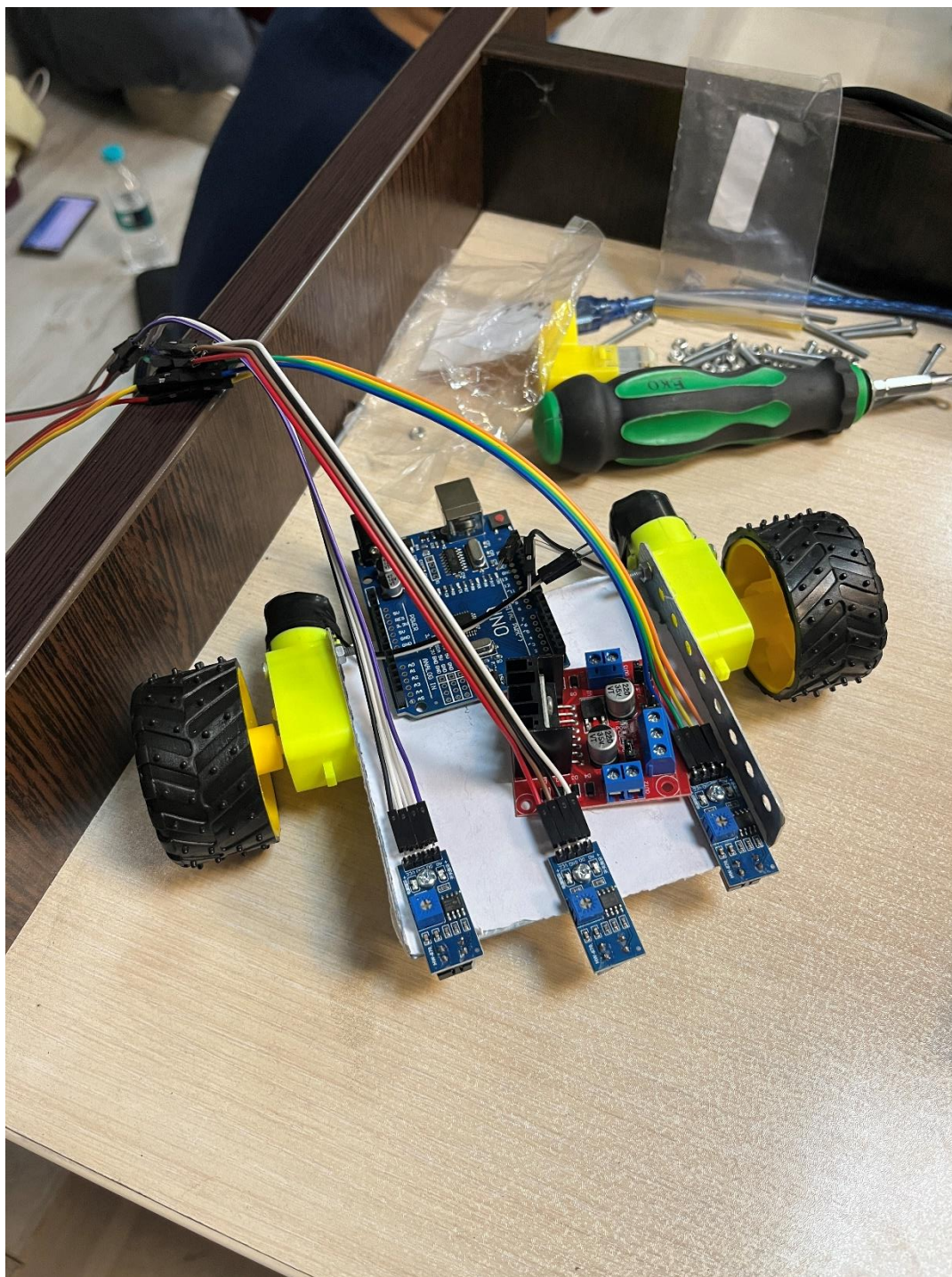
بعد قسمت GND درایور هم با آن ها یکی میشود (تمام این سیم ها را با لحیم به هم وصل میکنیم) و در آخر همه را باید به GND بر آردوینو وصل کنیم.

پایه دیگر VCC هست که باید به هم وصل شود. 3 تا VCC از سمت 3 واژول مادون قرمز ، V5 از درایور و برد آردوینو به هم وصل میکنم و به هم لحیم کردیم.

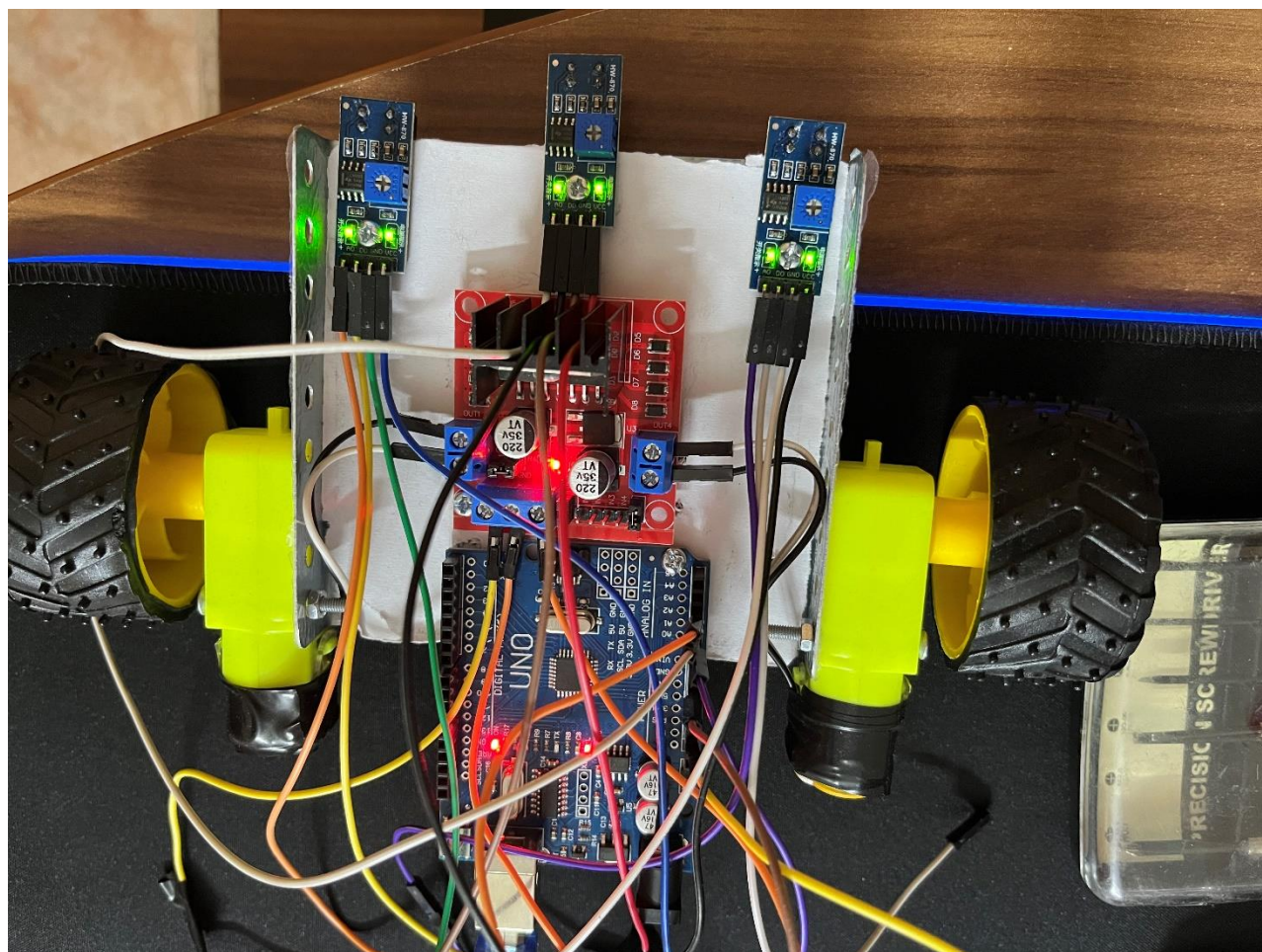
در آخر باید منبع تغذیه را تامین کنیم. با استفاده از پایه باطری که باطری به آن وصل میشود منبع را تغذیه ربات را تامین میکنیم. یک سر سیم باطری به V12 درایور وصل میکنیم و سر دیگر سیم را با سیم های GND لحیم میکنیم. مراحل اتصال کامل شده و با اتصال منبع تغذیه تمام چراغ های درایور، مادربرد، سنسور ها باید روشن شود. در آخر پین های In1 تا In4 را باید به پایه های 1 2 3 4 برد به ترتیب وصل کنیم.

عکس مراحل ساخت ربات :





روشن شدن چراغ های قطعات نشان دهنده ی اتصال صحیح است



کد نویسی

بعد از اسمبل کردن ربات و تست کردن اتصالات به مرحله کد نویسی میرسیم. در این مرحله باید منطق الگوریتم، روش و اجرا ربات را تعریف کنیم. بخش کد نویسی مربوط به بخش برد آردوینو میشود. لازم به ذکر است که قبل از شروع توضیحات کد ها بدانیم آردوینو چیست، چه زبانی دارد و چه کار میکند.

آردوینو چیست؟

به زبان ساده، آردوینو ابزاری برای کنترل الکترونیکی است. شما تعیین میکنید آردوینو برای شما چه کاری را انجام بدهد. با این برد شما میتوانید ربات های مختلف و دستگاه های متنوع (حضور و غیاب، کنترل تردد باشگاه ها، پروژه های امنیتی) بسازید. همچنین میتوانید هوشمند سازی ساختمان (کنترل از طریق اینترنت، بلوتوث، وای فای و...) را انجام بدهید.

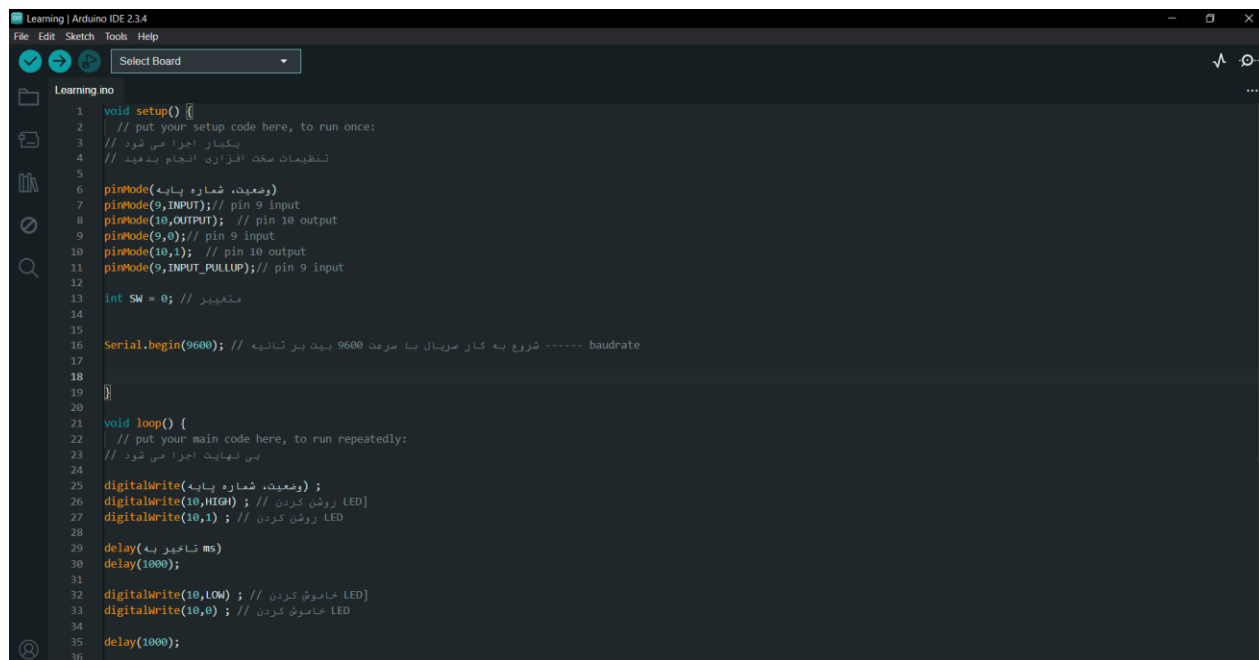
این برد میتواند ورودی ها را از سایر برد ها یا ماژول های الکترونیکی بگیرد. سپس با برنامه ای که به آن میدهید برای آن تعیین میکنید که چه عملکردی را در برابر ورودی ها مختلف انجام دهد.

آردوینو فقط سخت افزار نیست، بلکه نرم افزار بسیار محبوبی نیز دارد. چیزی به نام Arduino IDE (محیط توسعه یکپارچه) برای برنامه نویسی برد های آردوینو وجود دارد. این یک برنامه نرم افزاری است که بر روی کامپیوتر خود دانلود و نصب می کنید و سپس از آن برای برنامه ریزی بردهای آردوینو استفاده می کنید.

کد آردوینو

کد آردوینو بخش بسیار مهمی است. کدی که در Arduino IDE مینویسید در نهایت همان چیزی است که روی برد شما آپلود میشود و عملکرد برد شما را تعیین میکند. به کد آردوینو اسکچ (Sketch) گفته میشود. زبان برنامه نویسی آردوینو بر پایه زبان C و ++C است. بنابراین اگر شما برنامه نویسی آردوینو را یاد بگیرید، در حقیقت برنامه نویس C هم هستید. این کد های آردوینو هستند که نحوه کار برد را تعیین میکنند.

محیط Arduino IDE



```
1 void setup() {  
2   // put your setup code here, to run once:  
3   // یکبار اجرا می شود  
4   // تنظیمات سخت افزاری انجام بدیم  
5  
6   pinMode(شماره پایه,وضعیت);  
7   pinMode(9,INPUT); // pin 9 input  
8   pinMode(10,OUTPUT); // pin 10 output  
9   pinMode(9,0); // pin 9 input  
10  pinMode(10,1); // pin 10 output  
11  pinMode(9,INPUT_PULLUP); // pin 9 input  
12  
13  int SW = 0; // متغیر  
14  
15  
16  Serial.begin(9600); // شروع به کار سریال با سرعت 9600 بیت بر ثانیه ----- baudrate  
17  
18  
19 }  
20  
21 void loop() {  
22   // put your main code here, to run repeatedly:  
23   // بی نهایت اجرا می شود  
24  
25   digitalWrite(شماره پایه,وضعیت);  
26   digitalWrite(10,HIGH); // روشن کردن LED  
27   digitalWrite(10,1); // روشن کردن LED  
28  
29   delay(تایمر به ms)  
30   delay(1000);  
31  
32   digitalWrite(10,LOW); // خاموش کردن LED  
33   digitalWrite(10,0); // خاموش کردن LED  
34  
35   delay(1000);  
36 }
```

ما در این محیط کد های مربوطه رو نوشته، صحت و درستی سینتکس کد را بررسی میکنیم، کد را روی برد

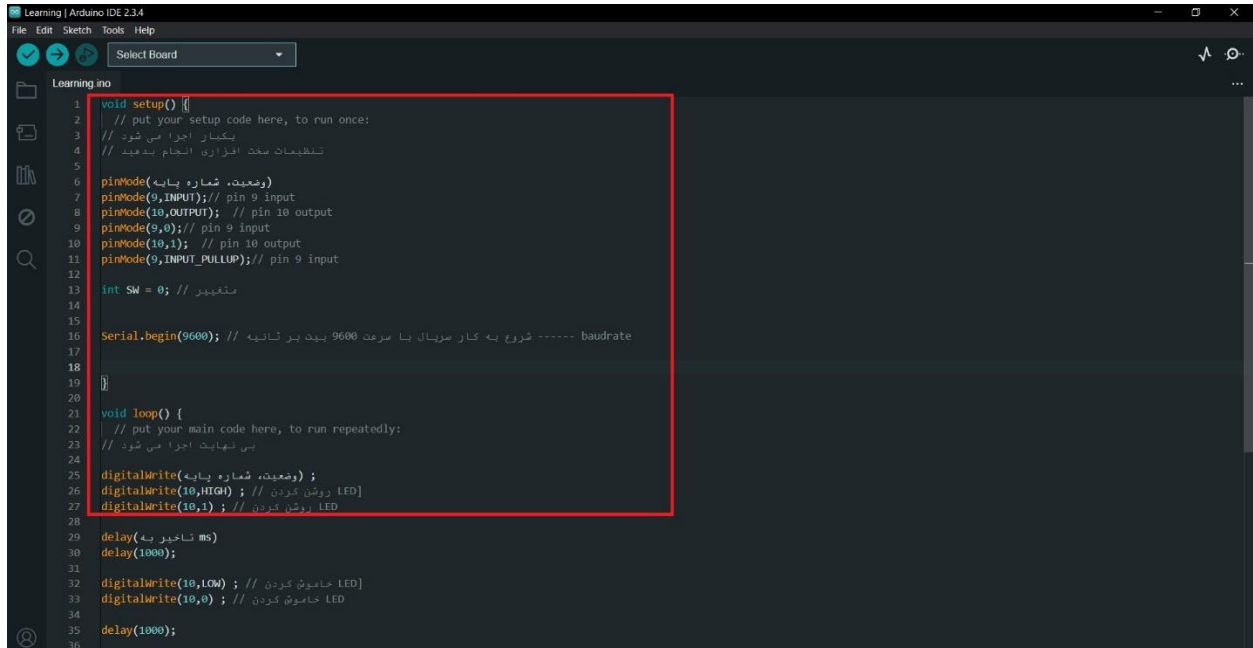
میریزم و از طریق serial monitor قسمت های پیرینت یا خروجی نوشته شده برای تست را مشاهده

میکنیم.

در ادامه به کد های نوشته شده و نحوه ی کار با این محیط می پردازیم.

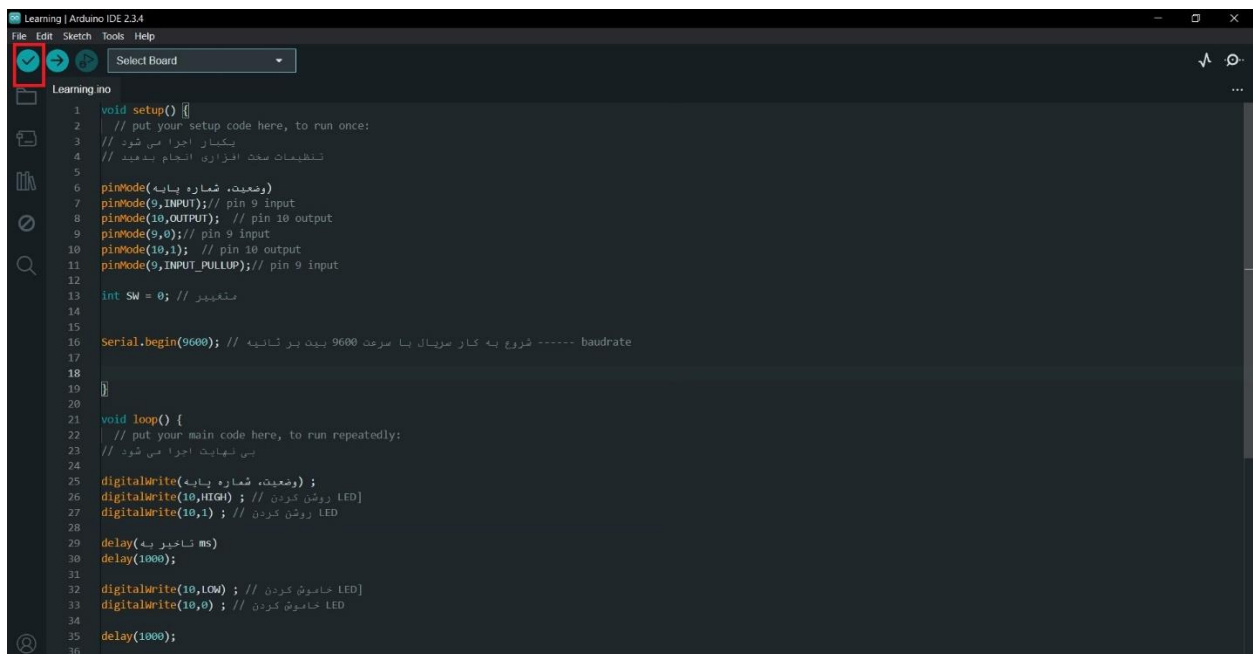
نحوا کار در محیط آردوینو

در این قسمت ما کد های مورد نظر را مینویسیم



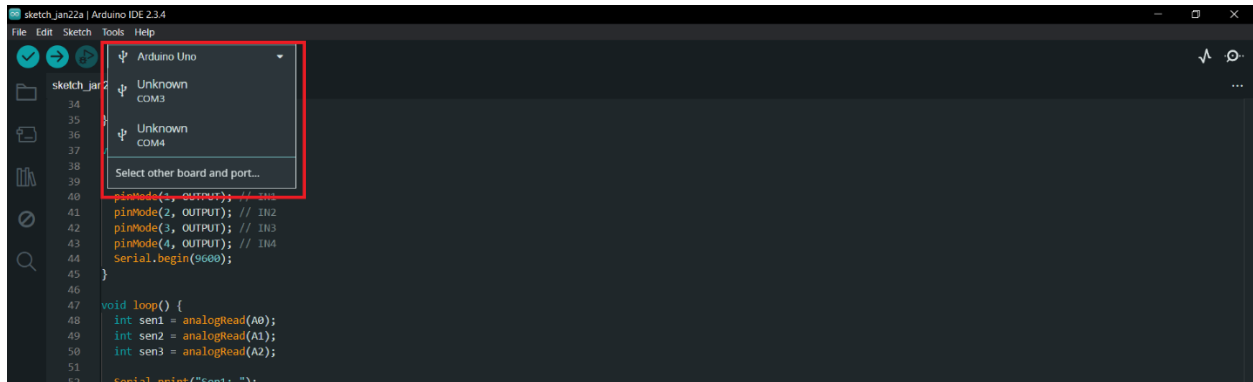
```
1 void setup() {  
2   // put your setup code here, to run once:  
3   // یکبار اجرا می شود  
4   // تنظیمات سخت افزاری انجام میگیرد  
5  
6   pinMode(شماره پایه)  
7   pinMode(9,INPUT); // pin 9 input  
8   pinMode(10,OUTPUT); // pin 10 output  
9   pinMode(9,0); // pin 9 input  
10  pinMode(10,1); // pin 10 output  
11  pinMode(9,INPUT_PULLUP); // pin 9 input  
12  
13  int SW = 0; // متغیر  
14  
15  
16  Serial.begin(9600); // ----- baudrate شروع به کار سریال با سرعت 9600 بیت بر ثانیه  
17  
18  
19  
20  
21  void loop() {  
22    // put your main code here, to run repeatedly:  
23    // بی نهایت اجرا می شود  
24  
25    digitalWrite(شماره پایه,وضعیت);  
26    digitalWrite(10,HIGH); // روشن کردن LED  
27    digitalWrite(10,1); // روشن کردن LED  
28  
29    delay(تاخیر به ms)  
30    delay(1000);  
31  
32    digitalWrite(10,LOW); // خاموش کردن LED  
33    digitalWrite(10,0); // خاموش کردن LED  
34  
35    delay(1000);  
36  }
```

بعد از وارد کردن کد ها نیاز است که آن ها را **verify** کنیم که زدن این تیک کد ما بررسی میشود که آیا از نظر سیستمکس درست است یا نه.

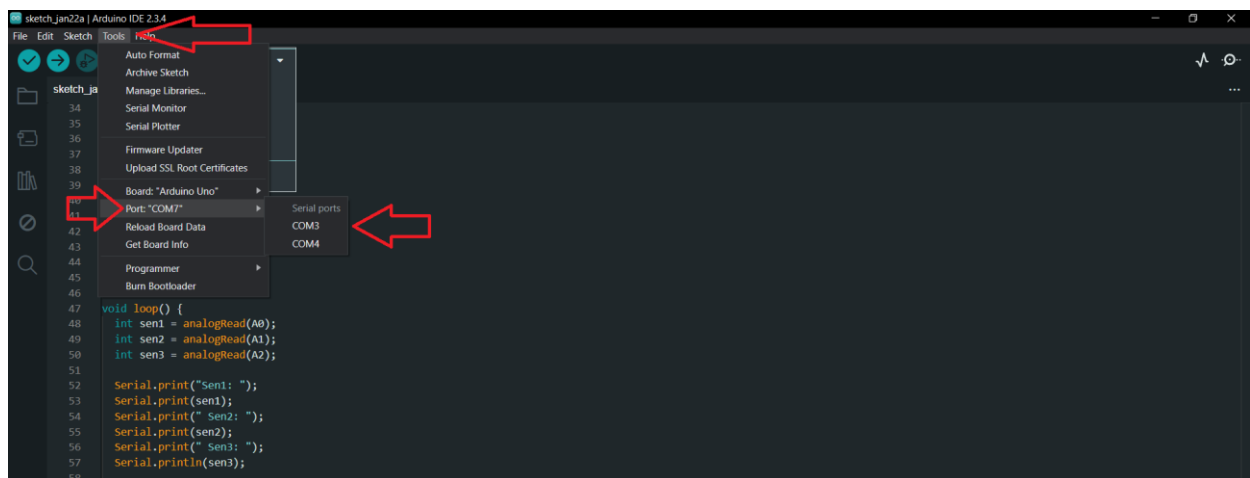


```
1 void setup() {  
2   // put your setup code here, to run once:  
3   // یکبار اجرا می شود  
4   // تنظیمات سخت افزاری انجام میگیرد  
5  
6   pinMode(شماره پایه)  
7   pinMode(9,INPUT); // pin 9 input  
8   pinMode(10,OUTPUT); // pin 10 output  
9   pinMode(9,0); // pin 9 input  
10  pinMode(10,1); // pin 10 output  
11  pinMode(9,INPUT_PULLUP); // pin 9 input  
12  
13  int SW = 0; // متغیر  
14  
15  
16  Serial.begin(9600); // ----- baudrate شروع به کار سریال با سرعت 9600 بیت بر ثانیه  
17  
18  
19  
20  
21  void loop() {  
22    // put your main code here, to run repeatedly:  
23    // بی نهایت اجرا می شود  
24  
25    digitalWrite(شماره پایه,وضعیت);  
26    digitalWrite(10,HIGH); // روشن کردن LED  
27    digitalWrite(10,1); // روشن کردن LED  
28  
29    delay(تاخیر به ms)  
30    delay(1000);  
31  
32    digitalWrite(10,LOW); // خاموش کردن LED  
33    digitalWrite(10,0); // خاموش کردن LED  
34  
35    delay(1000);  
36  }
```

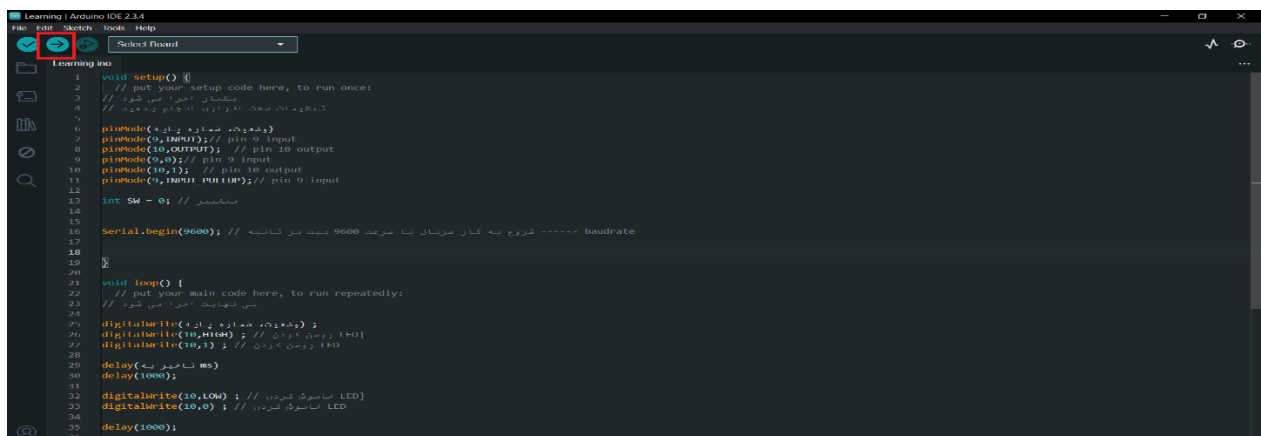
در این قسمت ، زمانی که برد به کامپیوتر وصل میشود باید برد رو انتخاب کنیم و درایور آن به صورت اتوماتیک دانلود کنیم.



حال بعد از انتخاب نوع برد باید پورتهای که به کامپیوتر هست را انتخاب کنیم. طبق روال عکس پیش میرویم. در بخش tools قسمت port انتخاب میکنیم و پورت احتمالی را انتخاب میکنیم.



زمانی که کد نوشته شد و تست شد. باید آن را بر روی برد upload کنیم.



بررسی کد اجرایی :

در این قسمت به آنالیز کد نوشته شده برای ربات میپردازیم.

این بخش پایه‌های مرتبط با درایور موتور L298N را تعریف می‌کند:

- **IN1 و IN2** برای کنترل جهت موتور سمت راست.
- **IN3 و IN4** برای کنترل جهت موتور سمت چپ.
- **ENA و ENB** برای کنترل سرعت موتورهای سمت راست و چپ به کمک PWM.

```
2
3 #define IN1 7
4 #define IN2 6
5 #define IN3 5
6 #define IN4 4
7 #define ENA 9
8 #define ENB 10
9
```

این بخش پایه‌های مرتبط با سنسورهای مادون قرمز را تعریف می‌کند:

- **LEFT_SENSOR** برای سنسور سمت چپ.
- **CENTER_SENSOR** برای سنسور وسط.
- **RIGHT_SENSOR** برای سنسور سمت راست.

```
9
0 #define LEFT_SENSOR A2
1 #define CENTER_SENSOR A1
2 #define RIGHT_SENSOR A0
3
```

مقدار آستانه‌ای برای تشخیص خط توسط سنسورها تعریف می‌شود. اگر مقدار خوانده‌شده از یک سنسور بزرگ‌تر از این مقدار باشد، به معنی تشخیص خط است.

```
13  
14 const int threshold = 500;  
15
```

پایه‌های موتور به عنوان خروجی تنظیم می‌شوند.

پایه‌های سنسور به عنوان ورودی تنظیم می‌شوند.

ارتباط سریال با سرعت 9600 بیت بر ثانیه برای ارسال داده‌ها جهت دیباگ شروع می‌شود.

```
void setup() {  
    pinMode(IN1, OUTPUT);  
    pinMode(IN2, OUTPUT);  
    pinMode(IN3, OUTPUT);  
    pinMode(IN4, OUTPUT);  
    pinMode(ENA, OUTPUT);  
    pinMode(ENB, OUTPUT);  
  
    pinMode(LEFT_SENSOR, INPUT);  
    pinMode(CENTER_SENSOR, INPUT);  
    pinMode(RIGHT_SENSOR, INPUT);  
  
    Serial.begin(9600);  
}
```

حرکت به جلو: موتورهای هر دو چرخ به سمت جلو حرکت می‌کنند. سرعت هر دو موتور با مقدار PWM برابر 200 تنظیم می‌شود.

```
void Forward() {  
    digitalWrite(IN1, HIGH);  
    digitalWrite(IN2, LOW);  
    analogWrite(ENA, 200);  
  
    digitalWrite(IN3, HIGH);  
    digitalWrite(IN4, LOW);  
    analogWrite(ENB, 200);  
}
```

حرکت به عقب : موتورهای هر دو چرخ به سمت عقب حرکت می کنند.

```
void Backward() {  
    digitalWrite(IN1, LOW);  
    digitalWrite(IN2, HIGH);  
    analogWrite(ENA, 200);  
  
    digitalWrite(IN3, LOW);  
    digitalWrite(IN4, HIGH);  
    analogWrite(ENB, 200);  
}
```

چرخش به چپ : موتور سمت چپ متوقف می شود (PWM برابر صفر).

موتور سمت راست به جلو حرکت می کند تا ربات به سمت چپ بچرخد.

```
void TurnLeft() {  
    digitalWrite(IN1, LOW);  
    digitalWrite(IN2, LOW);  
    analogWrite(ENA, 0);  
  
    digitalWrite(IN3, HIGH);  
    digitalWrite(IN4, LOW);  
    analogWrite(ENB, 200);  
}
```

چرخش به راست : موتور سمت چپ متوقف می شود (PWM برابر صفر). موتور سمت راست به جلو حرکت

می کند تا ربات به سمت راست بچرخد.

```
void TurnRight() {  
    digitalWrite(IN1, HIGH);  
    digitalWrite(IN2, LOW);  
    analogWrite(ENA, 200);  
  
    digitalWrite(IN3, LOW);  
    digitalWrite(IN4, LOW);  
    analogWrite(ENB, 0);  
}
```

توقف : هر دو موتور متوقف می شوند (PWM برابر صفر).

```
void Stop() {  
    digitalWrite(IN1, LOW);  
    digitalWrite(IN2, LOW);  
    analogWrite(ENA, 0);  
  
    digitalWrite(IN3, LOW);  
    digitalWrite(IN4, LOW);  
    analogWrite(ENB, 0);  
}
```

حلقه اصلی (loop)

خواندن مقادیر سنسورها

مقادیر آنالوگ سنسورهای سمت چپ، وسط و راست خوانده می شوند.

و قسمت دیباگ (نمایش مقادیر سنسورها) مقادیر خوانده شده از سنسورها از طریق سریال چاپ می شوند تا برای دیباگ استفاده شوند.

```
void loop() {  
    int leftValue = analogRead(LEFT_SENSOR);  
    int centerValue = analogRead(CENTER_SENSOR);  
    int rightValue = analogRead(RIGHT_SENSOR);  
  
    Serial.print("Left: ");  
    Serial.print(leftValue);  
    Serial.print(" | Center: ");  
    Serial.print(centerValue);  
    Serial.print(" | Right: ");  
    Serial.println(rightValue);  
}
```

تصمیم‌گیری :

اگر مقدار سنسور وسط بیشتر از آستانه باشد:

- ربات به جلو حرکت می‌کند.

اگر مقدار سنسور چپ بیشتر از آستانه باشد:

- ربات به سمت چپ می‌چرخد.

اگر مقدار سنسور راست بیشتر از آستانه باشد:

- ربات به سمت راست می‌چرخد.

اگر هیچ‌کدام از سنسورها خط را تشخیص ندهند:

- ربات متوقف می‌شود.

و در آخر تأخیر 100 میلی‌ثانیه برای پایداری و جلوگیری از تغییرات ناگهانی اعمال می‌شود.

```
if (centerValue > threshold) {  
    Forward();  
} else if (leftValue > threshold) {  
    TurnLeft();  
} else if (rightValue > threshold) {  
    TurnRight();  
} else {  
    Stop();  
}  
  
delay(100);  
}
```

با تشکر ویژه از جناب آقای دکتر عسگری برای راهنمایی‌های ارزشمند و حمایت‌های بی‌دریغ ایشان در مسیر یادگیری و اجرای این پروژه، و همچنین مهندس مهسنی برای مشاوره‌های دقیق و کمک‌های فنی که بدون آن‌ها این پروژه به نتیجه مطلوب نمی‌رسید. قدردان زحمات شما هستیم.

1. <https://www.aparat.com/v/d221q9i>
2. <https://nikushop.com/>
3. <https://roboeq.ir/>
4. <https://behnamrobotic.com/arduino/make-line-follower-robot-using-arduino/>
5. <https://chalik.net/build-a-line-tracking-robot/>
6. <https://alero.ir/others/line-follower-robot/>
7. <https://projecthub.arduino.cc/lightthedreams/line-following-robot-34b1d3>
8. <https://circuitdigest.com/microcontroller-projects/arduino-uno-line-follower-robot>
9. <https://robu.in/how-to-make-a-line-follower-robot-using-arduino-connection-code/>