

گزارش پروژه کاربرد کامپیوتر در کنترل

موضوع پروژه: کنترل و هدایت ربات با دوربین

افراد گروه: فاطمه چنگیزیان 9332682

سپیده نصراللهی 9332673

تابستان 1397

مقدمه

در این پروژه هدف کنترل ربات در یک محیط با کمک و نظارت یک دوربین می باشد. در اینجا دوربین به صورت ناظر در بالای محیط قرار گرفته می شود. نرم افزار باید به کمک تصویر دریافتی از دوربین محل ربات START را تخمین زده (تعیین کند) و با توجه به محل نهایی END یا مقصد ربات که به عنوان اطلاعات ورودی به نرم افزار داده می شود ربات را به سمت مقصد هدایت می کند. نکته قابل توجه این است که در صورت وجود موانع بین نقطه start و end سیستم باید آن ها را تشخیص داده و ربات را از بین موانع عبور دهد و به نقطه نهایی برساند.

ماژول NRF24L01

چیپ NRF24L01 نمونه تصحیح شده با امکانات بیشتر و دیتا ریت بالاتر چیپ NRF2401 است که توسط شرکت NORDIC در اوایل سال 2008 معرفی شد.

این چیپ در واقع یک ماژول بسیار عالی برای ارسال و دریافت اطلاعات بدون خطا است چیزی که در ماژول های HMTR یا حتی در RFX24R1 ها یا به کلی وجود نداشت یا این که مشکلات خاص خود را داشت.

مدولاسیون ارتباطی این ماژول به صورت GFSK است , همان مدولاسیونی که در تکنولوژی BLUETOOTH استفاده شده و به صورت انحصاری در دست چند شرکت بزرگ مثل BLUETOOTH و NORDIC SEMICONDUCTOR و TEXAS INSTRUMENT و چند شرکت دیگر قرار دارد.

فرکانس ارتباطی این چیپ 2.4 گیگا هرتز است که این خاصیت آن ویژگی های زیادی را برای ما به ارمغان می آورد از جمله کوچک شدن سایز آنت که حتی میتوان از خود PCB به صورت یک آنتن استفاده کرد , مورد دیگر هم به دلیل فرکانس بالا بسیار راحت تر از دیوار یا اجسام دیگر عبور می کند و با عث می شود که برد بیشتری هم به ما بدهد , می توانید فرکانس RFX24R1 یا HMTR را با این چیپ مقایسه کنید.

این چیپ به صورت دو طرفه کار می کند و در کل شما فقط به 2 عدد از این چیپ ها برای ارتباط لازم دارید

دیتا ریت این چیپ حداکثر 2 مگابیت بر ثانیه است که می توان از آن برای انتقال اطلاعات سنگینی مشابه صوت و یا حتی ویدئو استفاده کرد.

کارکرد پایه ها

IRQ ماژول توسط این پایه به میکرو وقفه میدهد (در حالت عادی پایه high است) و low شدن IRQ یکی از این حالت ها را می رساند:

-در گیرنده یک پکت دریافت شده

-در فرستنده پکت بدرستی ارسال شده (ACK تصدیق دریافت شد)

-در فرستنده یک پکت چندبار ارسال شده (retransmission) ولی تصدیقی دریافت نکرده پس عمل فرستادن اطلاعات به درستی انجام نشده است.

CE همان chip enable می باشد. در حالت عادی low است وقتی بخواهیم اطلاعات با ماژول بفرستیم یا دریافت کنیم باید این پین را high کنیم تا ماژول از مد استندبای (1) به active tx یا active rx برود (میتونید این پین را به VCC وصل کنید تا همیشه در حالت high باشد).

CSN در ارتباط spi به منظور پین CS یا ss استفاده میشود . در حالت عادی باید 1 باشد و وقتی با spi می خواهیم چیزی به ماژول بفرستیم باید آن را 0 کنیم

MOSI, MISO, CLK, CS برای ارتباط SPI هستند

VCC و GND هم تغذیه ماژول می باشد ماژول به ولتاژ معکوس و ولتاژ زیاد روی vcc و gnd حساس می باشد همچنین دمای زیاد روی پایه ها بوسیله هویه (چون پایه ها مستقیم به آیزی وصل می باشد) .

ماژول درایور موتور L298N

ماژول درایور L298N با استفاده از چیپ ST L298N به طور مستقیم می تواند 2 عدد موتور DC یا یک استپر موتور را راه اندازی کند.

- جریان پیوسته: 2 آمپر
- جریان لحظه ای: 3 آمپر
- ولتاژ ورودی: 3 الی 30 ولت
- توان: 25 وات
- سطح ولتاژ کنترلی: 0 و 5 ولت
- ابعاد: 43mm*27mm*43mm
- برای به حرکت درآوردن موتورها، از درایور L298n استفاده کرده ایم.
- دو سر مثبت و منفی موتور سمت چپ به out1 و out2 روی درایور وصل می شوند. سرعت این موتور توسط پایه ی EnA و جهت حرکت آن با پایه های in1 و in2 تنظیم می شود. به این صورت که پایه ی EnA به پینی روی آردوینو وصل می شود که سیگنال pwm تولید می کند. با عوض کردن duty cycle این سیگنال، ولتاژ داده شده به دو سر موتور را تغییر داده (pulse width modulation) و در نتیجه سرعت حرکت موتور را کنترل میکنیم.
- عوض کردن جهت حرکت موتور از طریق عکس کردن جهت جریان ورودی به آن اتفاق می افتد. این کار را با فرستادن صفر و یک به پایه های in1 و in2 انجام می دهیم. به عنوان مثال، یک low به in1 و یک high به in2 باعث می شود که موتور چپ، پادساعتگرد بچرخد. دو پایه ی یاد شده به هر کدام از 14 پین دیجیتال روی آردوینو می توانند وصل شوند.
- به طور متناظر اتصالات برای موتور دوم انجام میشود. یعنی مثبت و منفی موتور سمت راست به out3 و out4 وصل می شوند. EnB به یک پین روی آردوینو که سیگنال pwm تولید کند و in3 و in4 به هر کدام از پین های دیجیتال روی آردوینو وصل می شوند.

سنسور Lm35

معروفترین Sensor تشخیص دما موجود در بازار Lm35 می باشد ، این سنسور تغییرات دمای مورد نظر را به ولتاژ آنالوگ تبدیل می کند .

این سنسور دارای سه پایه می باشد در صورتی که سنسور روربروی ما قرار گیرد (بتوانیم نوشته هایش را ببینیم).

اولین پایه ، سمت چپ (VCC) می باشد که به ۵ ولت وصل می شود.

پایه وسط ، ولتاژ خروجی (Vout) است که به میکروکنترلر متصل می شود.

پایه سوم ، زمین (GND) سنسور است.

محدوده دمایی که این سنسور قادر به اندازه گیری آن می باشد بین ۵۵- تا ۱۵۰+ درجه سانتیگراد است و این سنسور به ازای هر درجه سانتیگراد ۱۰ میلی ولت ولتاژ خروجی را تغییر میدهد . یعنی به ازای دمای ۱ درجه ، ولتاژ خروجی سنسور ۱۰ میلی ولت و به ازای ۱۰۰ درجه خروجی سنسور ۱۰۰۰ میلی ولت می باشد.

همچنین به ازای دمای ۲۰- درجه خروجی سنسور ۲۰۰- میلی ولت می باشد .

ولتاژ تغذیه این سنسور ۵ ولت می باشد ، همچنین بدنه آن قابلیت تحمل دما تا ۲۰۰ درجه سانتیگراد را دارد.

از آنجا که مبدل آنالوگ به دیجیتال داخل میکروکنترلر مانند میکرو AVR ده بیتی است و ولتاژ مرجع آن بین صفر تا ۵ ولت است در نتیجه ولتاژ اندازه گیری شده را به ۱۰۲۴ قسمت تقسیم می کند پس می تواند سنسور دمای مورد نظر ما را به راحتی با دقت ۰,۵ درجه سانتیگراد بخواند .

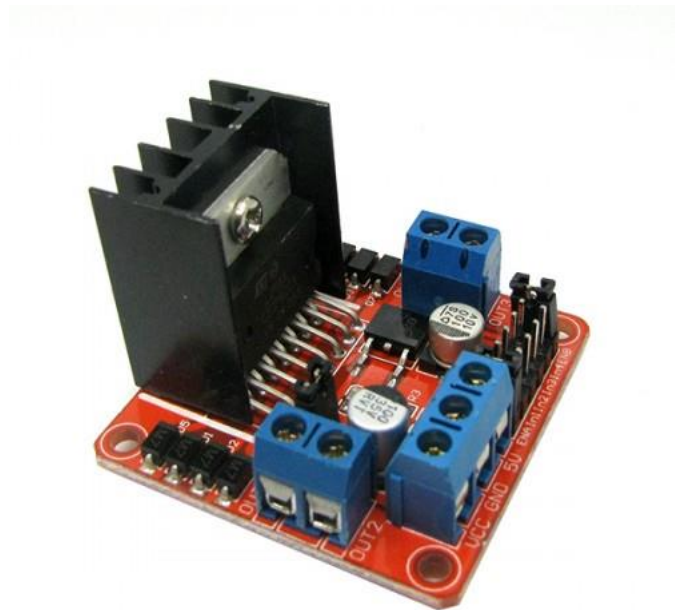
گام اول: ابزار و قطعات مورد نیاز

-استفاده از Arduino uno به عنوان پردازنده اصلی

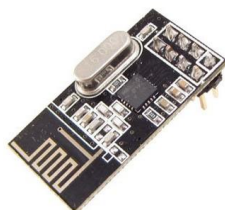


2 *

-ماژول راه انداز (درایور) موتور L298



-ماژول Wireless NRF24L01 به همراه آداپتور

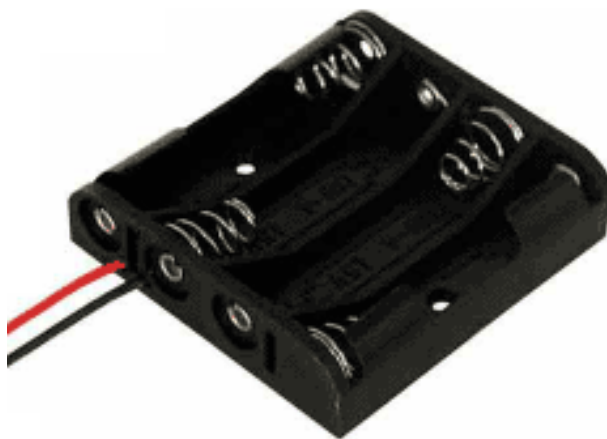


-برد برد

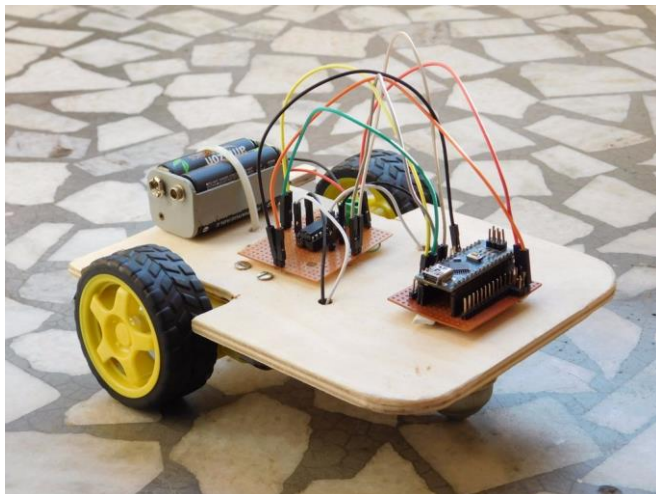
-سیم جامپر برد برد

-چهار عدد باتری 1.5 ولتی

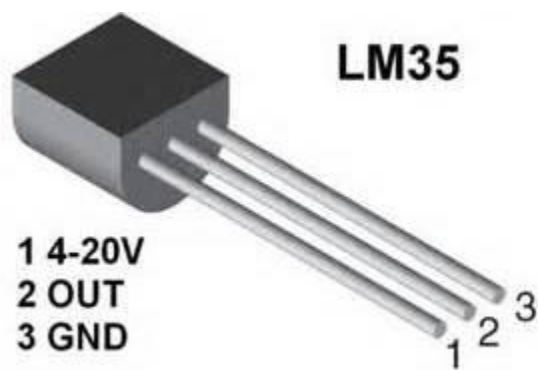
-جا باتری چهارتایی تخت



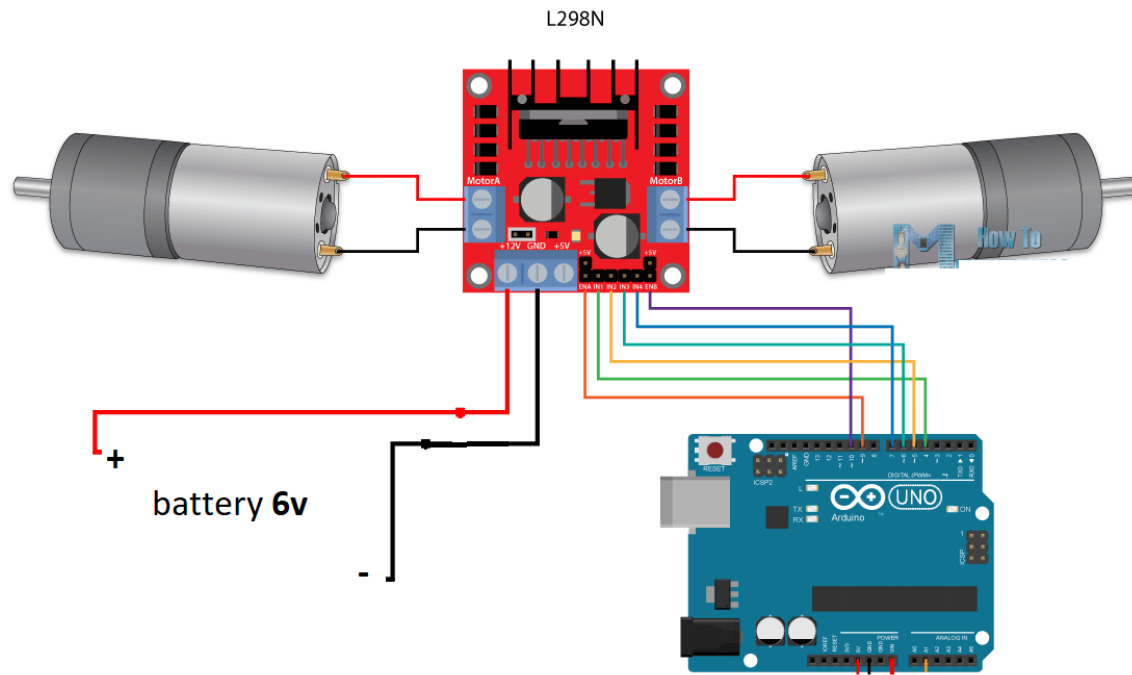
-ربات قابل هدایت به همراه دو عدد موتور دنده دار

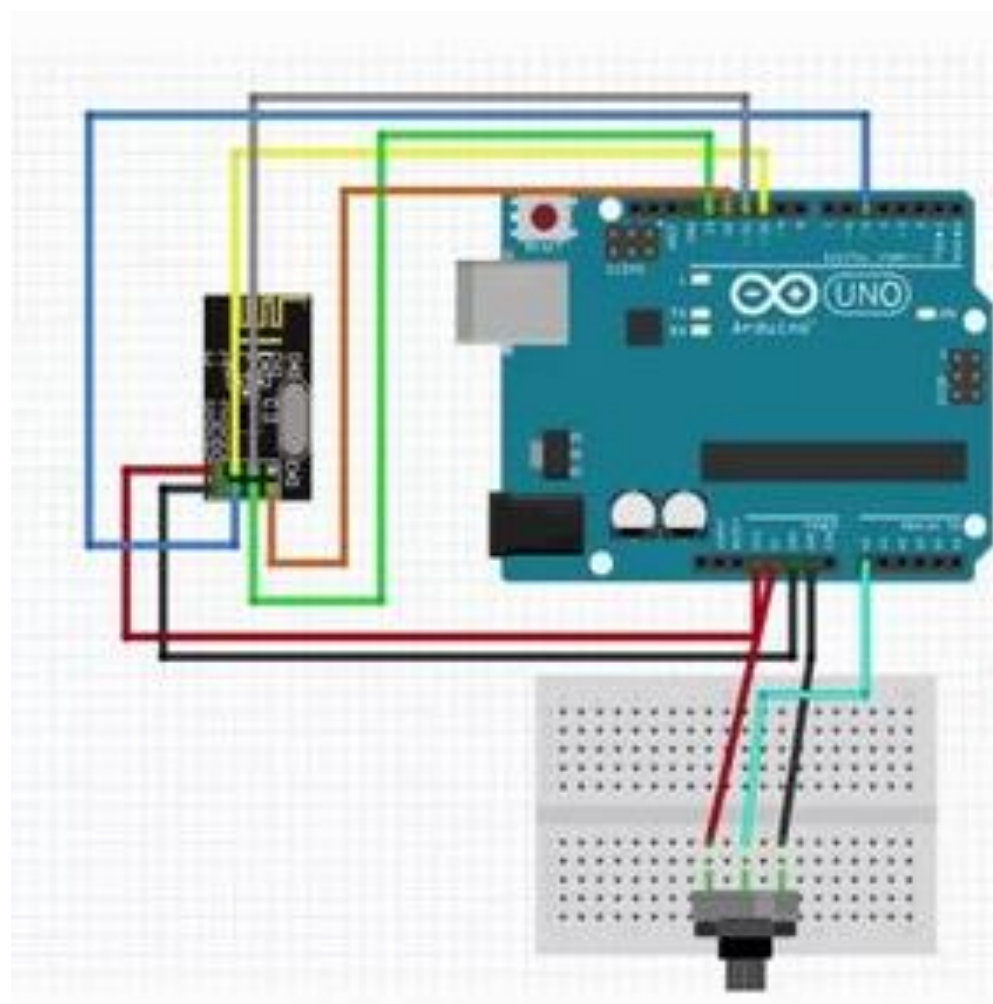


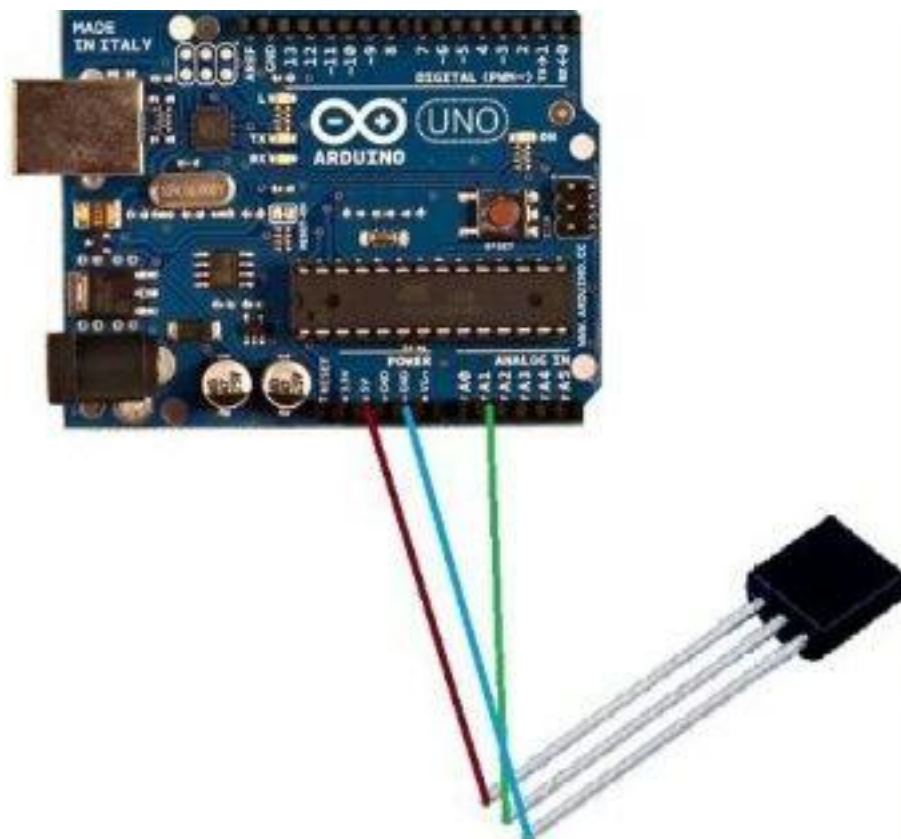
-سنسور lm35 برای اندازه گیری دما درایور



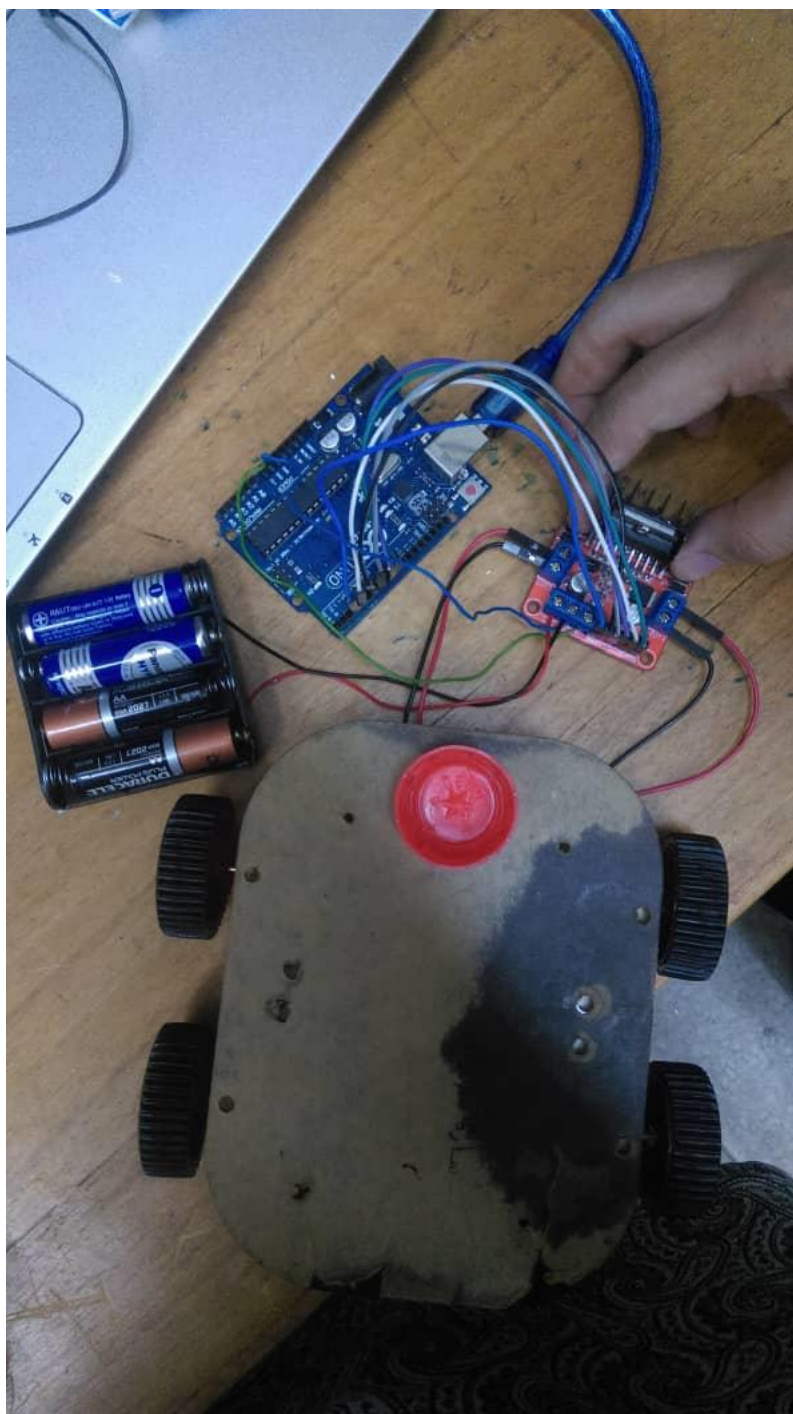
گام دوم : اتصالات سخت افزار







در زیر تصویر واقعی از مدار را مشاهده میکنید



گام سوم: کد آردوینو

برای اینکه موتور رو به جلو (forward) حرکت کند، بایستی موتور سمت راست ساعتگرد و سمت چپ پادساعتگرد بچرخند. کد آردوینوی این حالت در زیر آورده شده است.

```
8 int enA = 3;
9 int in1 = 6;
10 int in2 = 7;
11 int enB = 9;
12 int in3 = 8;
13 int in4 = 4;
14
15 void setup()
16 {
17     // set all the motor control pins to outputs
18     pinMode(enA, OUTPUT);
19     pinMode(enB, OUTPUT);
20     pinMode(in1, OUTPUT);
21     pinMode(in2, OUTPUT);
22     pinMode(in3, OUTPUT);
23     pinMode(in4, OUTPUT);
24 }
25
26 void demoOne()
27 {
28     // this function will run the motors in both directions at a fixed speed
29     // turn on motor A
30     digitalWrite(in1, HIGH);
31     digitalWrite(in2, LOW);
32     // set speed to 200 out of possible range 0-255
33     analogWrite(enA, 80);
34
35     // turn on motor B
36
37     digitalWrite(in3, LOW);
38     digitalWrite(in4, HIGH);
39     // set speed to 200 out of possible range 0-255
40     analogWrite(enB, 80);
41 }
```

برای اینکه موتور رو به عقب حرکت کند (backward) بایستی جای low و high را برای هر موتور عوض کرد.

```
void demoOne()
{
    // this function will run the motors in both directions at a fixed speed
    // turn on motor A
    digitalWrite(in1,LOW);
    digitalWrite(in2,HIGH);
    // set speed to 200 out of possible range 0~255
    analogWrite(enA, 200);

    // turn on motor B

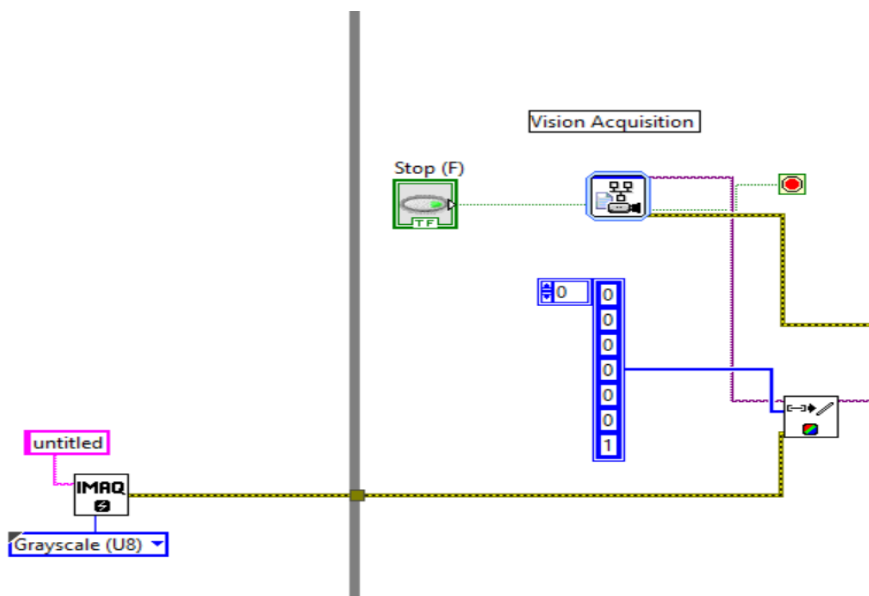
    digitalWrite(in3,HIGH);
    digitalWrite(in4,LOW);
    // set speed to 200 out of possible range 0~255
    analogWrite(enB, 200);
}
```

کد lm35

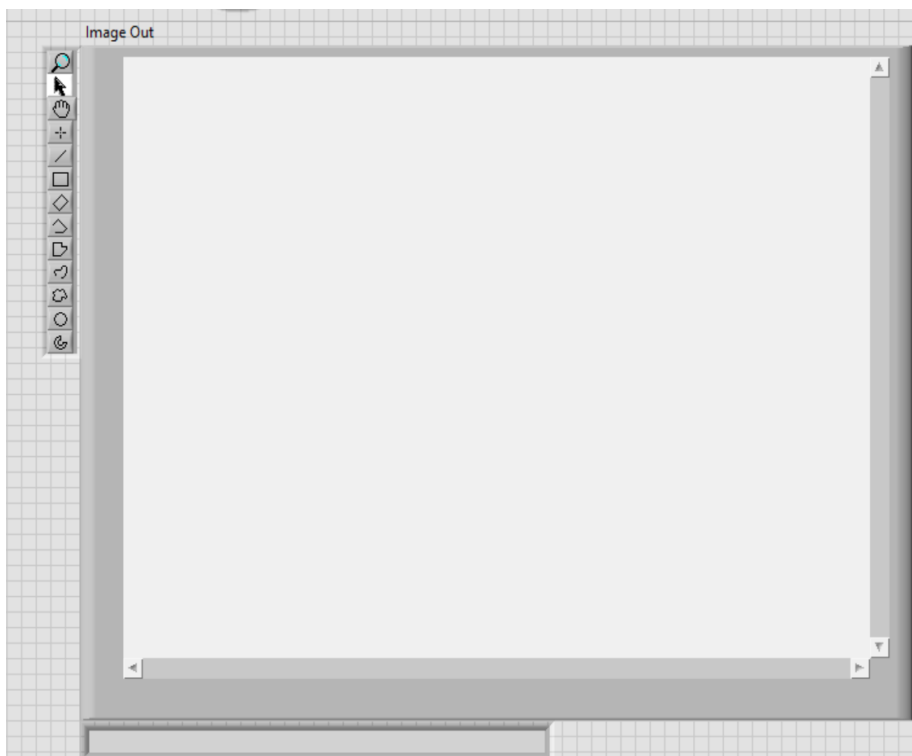
```
1 float temp;
2 int tempPin = 1;
3
4 void setup() {
5   Serial.begin(9600);
6 }
7
8 void loop() {
9   temp = analogRead(tempPin);
10  // read analog volt from sensor and save to variable temp
11  temp = temp * 0.1074;
12  // convert the analog volt to its temperature equivalent
13  Serial.print("TEMPERATURE = ");
14  Serial.print(temp); // display temperature value
15  Serial.print("C");
16  Serial.println();
17  delay(1000); // update sensor reading each one second
18 }
```

گام چهارم: کد لب ویو (labview)

در اولین مرحله از برنامه دوربین مورد نظر ما در برنامه لود شده و تصویر دوربین به صورت آنلاین قابل مشاهده می باشد.



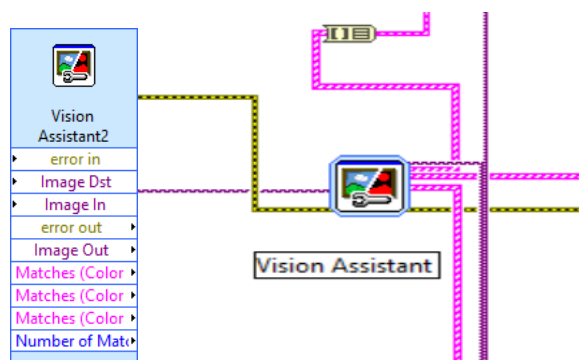
تصویر در بلوک زیر نمایش داده میشود



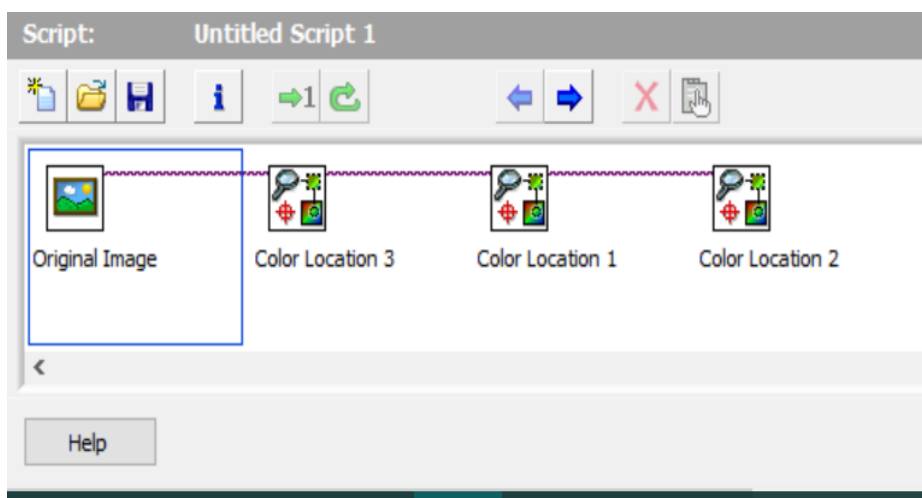
حال برای تشخیص ربات ،هدف و موانع از ایده تشخیص رنگ استفاده شده است
 بدین صورت که رنگ قرمز معرف ربات ،رنگ آبی معرف هدف و رنگ
 ____ معرف موانع می باشد

حال با استفاده از vision assistant برای هر مورد در قسمت color location یک نمونه از رنگ مورد نظر ساخته و به برنامه دستور میدهیم که به طور آنلاین آن را تشخیص دهد(دقت شود که بهتر است template از روی خود تصویر دریافتی از دوربین ساخته شود زیرا روند تشخیص بهتر و دقیقتر عمل می کند)

در این برنامه سه رنگ مختلف در color location به طور مجزا ساخته شده است



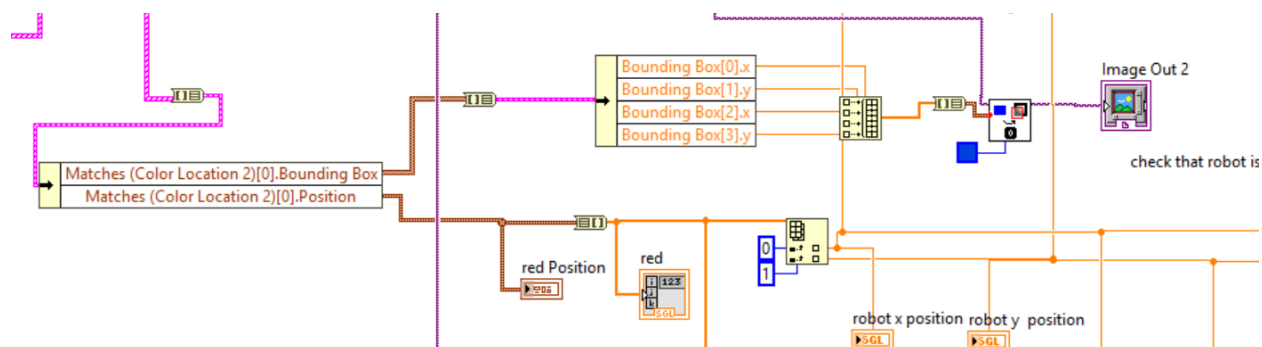
تصویر زیر مربوط به خود برنامه ویژن میباشد



حال در تصویر زیر با استفاده از اطلاعات خروجی ویژن در مورد تشخیص مکان هر رنگ اطلاعات را به صورت عددی و همچنین آرایه نمایش می دهیم

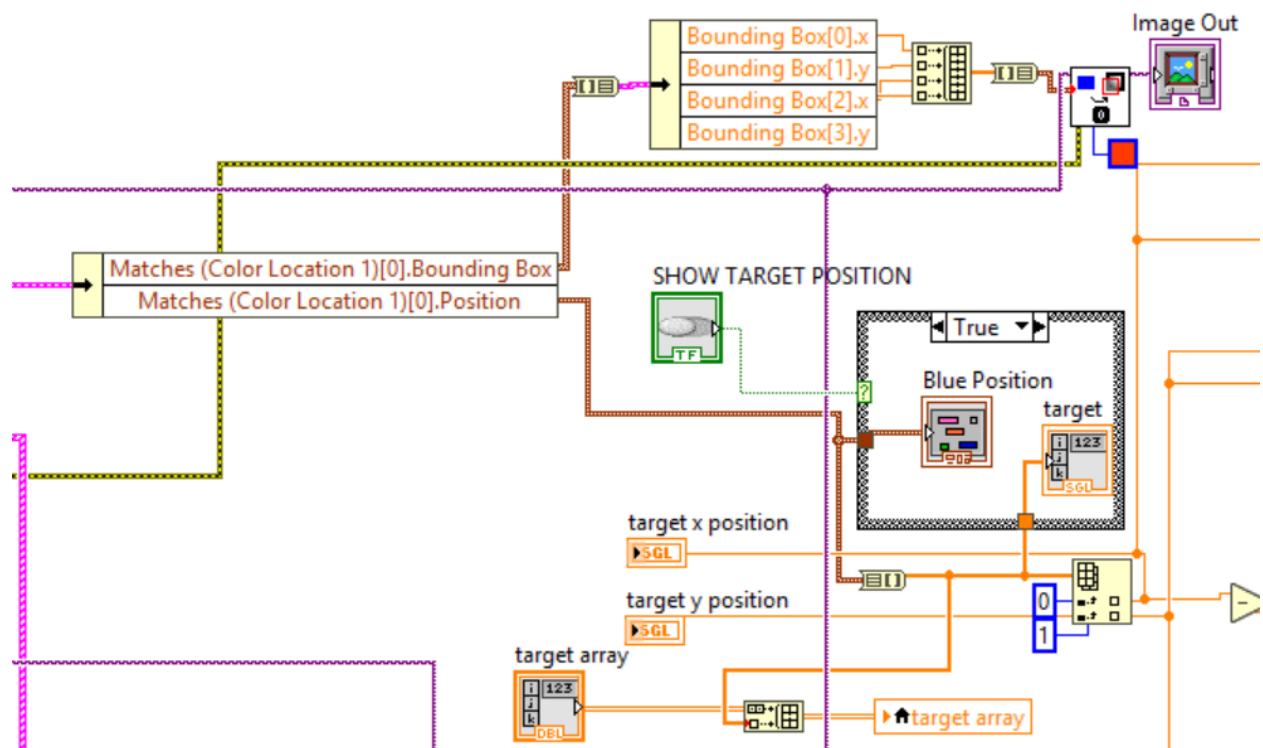
این قسمت همان نمایش آنلاین مکان ربات (رنگ قرمز) میباشد

دقت شود که در این قسمت یک باکس مربعی (bounding box) برای ایجاد یک کادر رنگی دور تصویر تشخیص داده شده استفاده شده است.



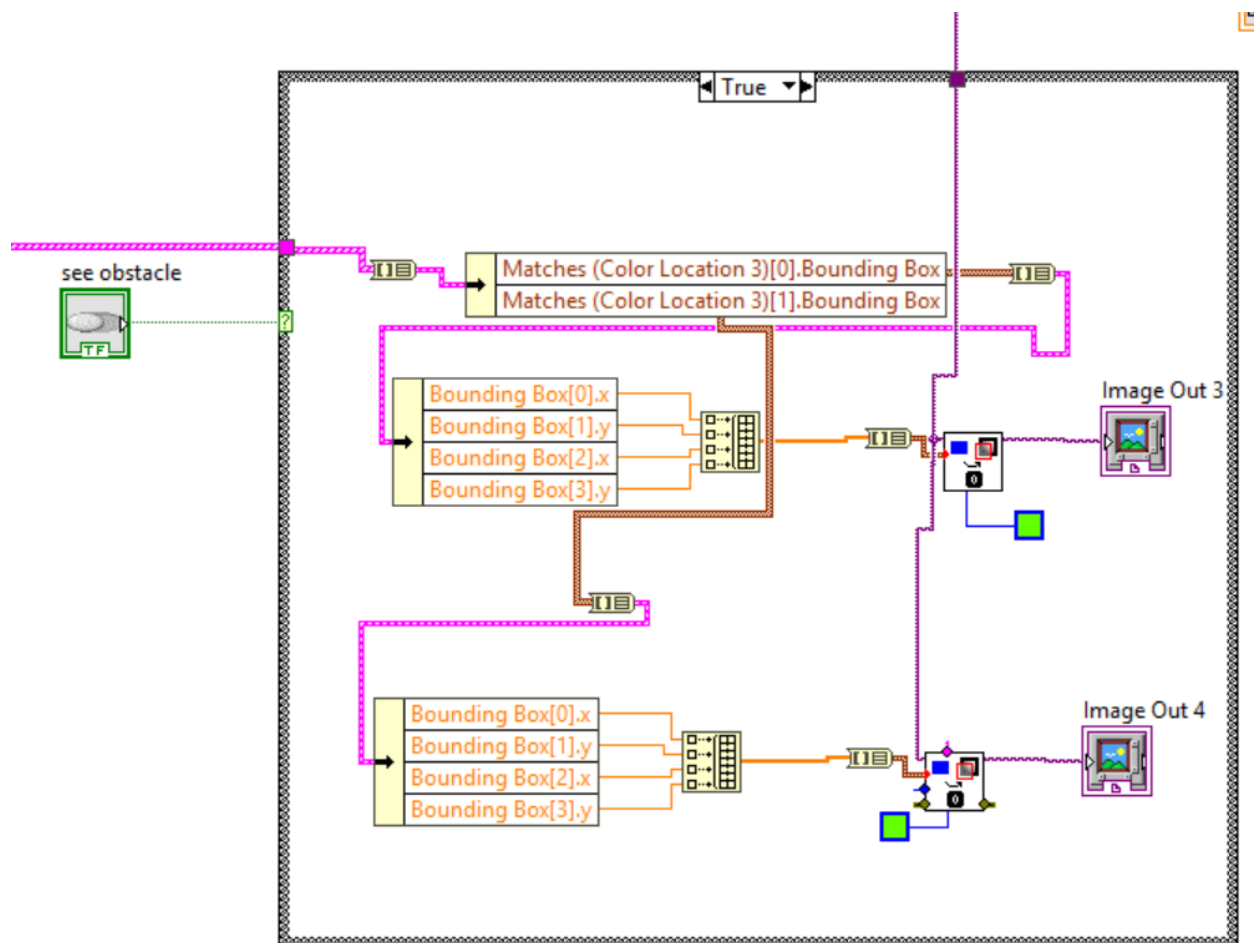
در کد بالا میتوانید مختصات ربات را به صورت آنلاین با مختصات پیکسلی مشاهده کنید

کد زیر مشابه کد بالا برای هدف و موانع می باشد:



دقت شود در کد بالا با فشردن کلید show target position میتوانيد موقعيت هدف را به طور آنلاين مشاهده كنيد در غير اين صورت مختصات هدف نمايش داده نميشود.

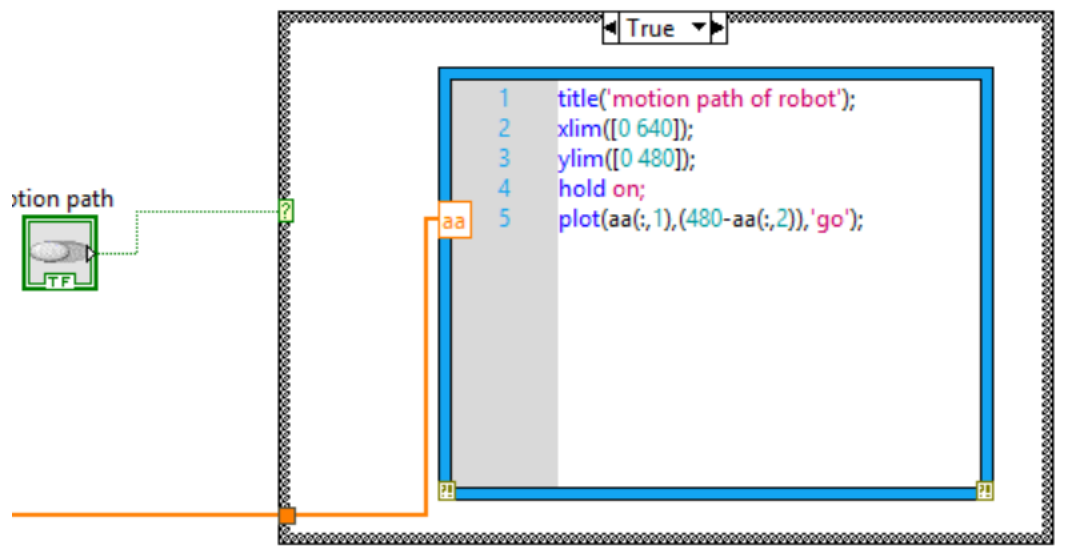
در کد زیر می‌توانید با فشردن کلید `see obstacles` قابلیت تشخیص و مشاهده موانع را فراهم کنید در غیر این صورت موانع نشان داده نمی‌شوند



S

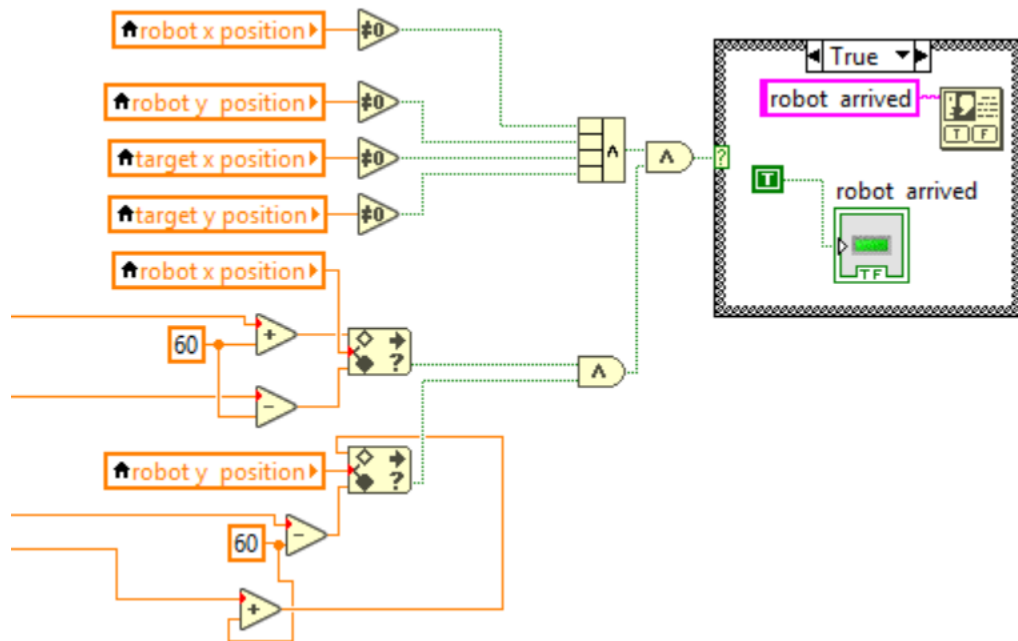
برای مشاهده مسیر حرکت ربات به صورت آنلاین و به کمک نقطه از mathscript tool box استفاده کرده و کد متلب مورد نظر را در mathscript node مینویسیم دقت شود که ورودی آن همان بردار مختصات ربات در هر لحظه می باشد.

به کمک کلید motion path امکان فعال سازی و غیر فعال سازی را فراهم میکنیم



در کد زیر قصد داریم به کمک مقایسه مختصات ربات و هدف قابلیت اعلام رسیدن ربات به نقطه مورد نظر را ایجاد کنیم

از آنجایی که اگر ربات خیلی خیلی به هدف نزدیک شود نقطه هدف دیگر برای دوربین قابل تشخیص نیست و مختصات آنرا صفر وارد میکند (مختصات صفر در این برنامه عموماً به معنی عدم تشخیص جسم توسط دوربین میباشد) پس تلاش ما این است که ربات تا حد ممکن به هدف نزدیک باشد اما همچنان دوربین قابلیت تشخیص هدف را داشته باشد پس یک محدوده در اطراف هدف در نظر میگیریم که اگر ربات در آن محدوده قرار گرفت پیام رسیدن به کاربر اعلام میشد و یک چراغ نیز روشن میشود



حال به سراغ اندازه گیری دمای درایور توسط سنسور lm35 میرویم در کد زیر ابتدا اطلاعات روی پورت com که آردوینو به آن وصل است خوانده می شود.

ای کار در سه مرحله انجام می شود: ابتدا لب ویو پورت مورد نظر را شناسایی میکند که چون این مرحله فقط یک بار انجام می شود بیرون از حلقه while قرار می گیرد. سپس اطلاعات از پورت خوانده شده و بعد در انتها نیز بیرون حلقه بایستی پورت را ببندیم تا دیگر توسط ما اشغال نشده باشد.

The diagram illustrates a state machine for a robot. It is divided into two states by a vertical line. The left state, labeled "robot is not in the picture", contains a "Time has Elapse" timer block, a "Display Message to User" block, and a "robot is not in the picture" indicator. The right state, labeled "robot is in the picture", contains a "Display Message to User2" block. The diagram includes various control elements like "True" buttons, "A" blocks, and comparison blocks.

ابتدا اطلاعاتی که روی پورت com قرار گرفته توسط لب ویو خوانده میشود و داده حاصل از string به numeric تبدیل میشود. دقت شود که به علت تقسیم مقاومتی انجام داده شده این داده در واقع نصف شارژ کنونی باتری می باشد پس از آن از این مقدار در صد گیری شده اگر این درصد برابر صد باشد چراغ سبز میشود و مقدار full در battery charge condition ریخته میشود اگر درصد بالاتر از 75 باشد چراغ نارنجی میشود و حالت باتری high است اگر بین 75

15 بود چراغ زرد میشود و حال باتری middle state است اگر کمتر از 15 بود چراغ قرمز میشود و حالت باتری low است.

