

گزارش فنی تیم Nemesis (Junior Soccer Light Weight)

طاها علیخانی^۱، محمد مهدی قاسمی^۲، امیر حسین کشاورز^۳، سیدامیرعلی نواده^۴

تهران، میدان فردوسی، خیابان ایرانشهر، دبیرستان پسرانه تیزهوشان علامه حلی^۵

NemesisteamH5@gmail.com

چکیده

در این مقاله به شرح مراحل طراحی و ساخت رباتهای فوتبالیست تیم Nemesis پرداخته شده است. ربات های این تیم متشکل از سه بخش اصلی مکانیک، الکترونیک، الگوریتم و برنامه نویسی است. ما با استفاده از چهار موتور DC که سرعتی حدود پانصد و پنجاه و پنج دور در دقیقه و چرخ های خورشیدی (trans) که امکان حرکت در تمامی جهات را با هدف دنبال کردن توپ برای ربات فراهم می کند استفاده کرده ایم. سخت افزار طراحی شده توانایی تشخیص پالس مادون قرمز ارسالی از توپ را به کمک سنسور های IRM8601 دارد و پس از تشخیص توپ، دستور ها و فرمان کنترلی لازم توسط میکروکنترلر STM32 به درایور موتور L6203 ارسال می شود. در هر کدام از این بخش ها زیر مجموعه هایی وجود دارد که در ادامه به طور کامل به آن ها اشاره می کنیم.

واژه های کلیدی: موتور DC ، STM32F103C8T6 ، L6203 ، IRM8601

1. مقدمه

ربات ها ، بازیگران عصر تکنولوژی:

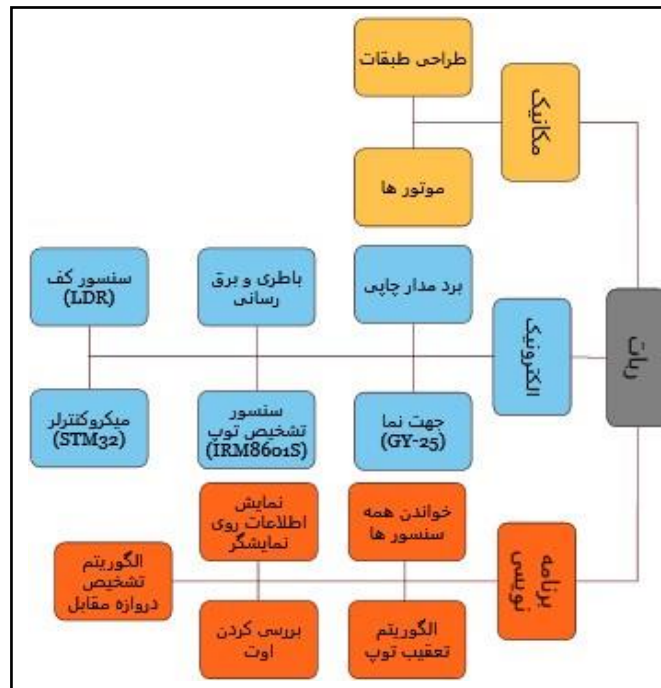
انجام بعضی از امور برای ما انسان ها شاید بسیار خسته کننده و عذاب آور باشد. اما بعضی از امور حتما باید انجام شوند. پس ما دو راه بیشتر نداریم اول اینکه کار های خود را با همان سختی انجام دهیم یا اینکه با استفاده از ذهن خلاق خود امور را به بازیگران عصر تکنولوژی بسپاریم. ربات ها همان بازیگران عصر تکنولوژی هستند بازیگرانی که روز به روز بیشتر خود را در معرض دید عموم میگذارند. از ارسال بسته های پستی گرفته تا حتی ورزش مهیج و پرطرفدار فوتبال که مورد توجه تیم ما قرار گرفت و به فکر تشکیل تیم ربات فوتبالیست سبک وزن و شرکت در مسابقات ایران این پرداختیم.

1- برنامه نویسی

2- الکترونیک و مکانیک

3- الکترونیک و مکانیک

4- برنامه نویسی



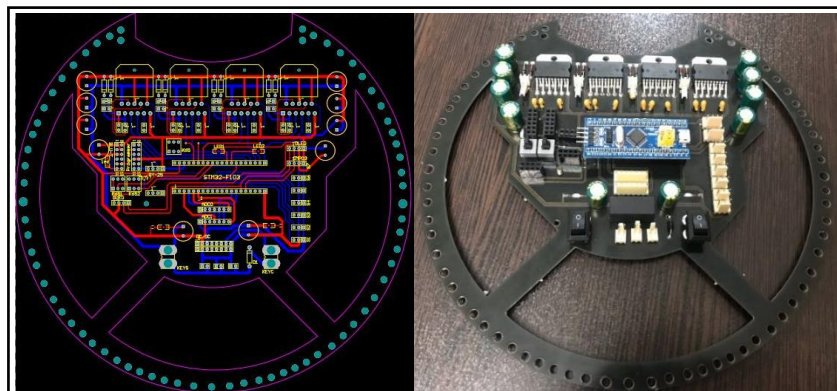
شکل 1: صفر تا صد ربات در یک نگاه

2. سیستم حرکتی:

2.1. مکانیک سیستم حرکتی:

2.1.1. بدنه

- بدنه ربات به طور کلی شامل سه طبقه می باشد که هر طبقه از جنس مدار چاپی و در برنامه ALTUM DESIGNER طراحی شده است:
- شاسی یا کفی ربات از چهار موتور و چرخ ها و مدار LDR تشکیل شده است.
- بخش دوم محل قرارگیری سنسور های مادون قرمز (IRM8601) است. که کار تشخیص توپ را انجام می دهند.
- بخش سوم محل قرار گیری باقی قطعات الکترونیکی است.



شکل 2: نمای روی ربات

2.1.2.

موتور ها و چرخ ها:

سیستم دو موتور و دو چرخ: با توجه به اینکه هر چه تعداد موتور ها کمتر باشد وزن نهایی ربات نیز کمتر میشود، پس بهتر است که از دو موتور قویتر با امکانات بیشتر استفاده کرد، به این منظور می بایست دو موتور با زاویه 180 درجه نسبت به هم و 90 درجه نسبت به جلوی ربات قرار گیرند. در نتیجه طبق این سیستم، ربات می تواند در دو جهت اصلی حرکت کند. اما در این صورت سرعت عکس العمل و قدرت مانور ربات کمتر می شود. در نتیجه زمان زیادی نسبت به دیگر سیستم ها برای به دست آوردن توپ، بازگشت به دروازه و ... تلف می شود. به این دلایل ما از این سیستم استفاده نکرده ایم.

سیستم سه موتور و سه چرخ: برای داشتن سرعت مناسب جهت مانور و عکس العمل میتوان از سیستم سه موتور و سه چرخ استفاده نمود. برای استفاده از این سیستم می بایست موتور ها با زاویه 120 درجه نسبت به هم قرار گیرند. این سیستم به 6 جهت اصلی و 6 جهت فرعی توان حرکتی دارد. این سیستم به طور نسبی از سرعت، مانور بالا و همچنین وزن کم برخوردار است اما سرعت مطلوب مورد نیاز ما را ندارد.



شکل 3: حالت های مختلف قرار گیری چرخ ها

سیستم چهار موتور و چهار چرخ: در این سیستم به دلیل استفاده از چهار موتور، ربات توانایی مانور و عکس العمل سریع را دارد. در این سیستم 4 موتور با زاویه 45 درجه نسبت به هم استفاده شده است و ربات می تواند در 8 جهت اصلی و 8 جهت فرعی حرکت کند. در این سیستم تراز بودن هر چهار چرخ اهمیت بالایی دارد و همچنین با توجه به تعداد موتور ها و محدودیت وزنی مشخص شده از سوی کمیته مسابقه باید از موتور های سبک تر و ضعیف تر استفاده کنیم. ما از موتور های تایوانی DC GEAR MOTOR با ولتاژ کاری 15-9 ولت و 555 دور در دقیقه استفاده کرده ایم که بیشتر شاخص های مورد نیاز ما را دارند.



شماره مدل	KG2102JB-0138S
ابعاد	شعاع: 25 میلی متر طول: 45 میلی متر
بازه ولتاژ کاری	9-15 ولت
تعداد دور در دقیقه (rpm)	555 دور در دقیقه
وزن	75 گرم

شکل 4: مشخصات موتور

چرخهای خورشیدی:

چرخ های ترنس (خورشیدی) چرخ هایی هستند که از یک چرخ ساده و تعدادی چرخ هرزگرد متصل به چرخ اصلی تشکیل شده اند. ساختار این چرخ ها طوری طراحی شده است که قابلیت حرکت در دو جهت را داراست و این قابلیت، امکان حرکت ربات در جهات مختلف را آسان تر و سریع تر می کند. سبک بودن نیز از ویژگی های مثبت این چرخ هاست.



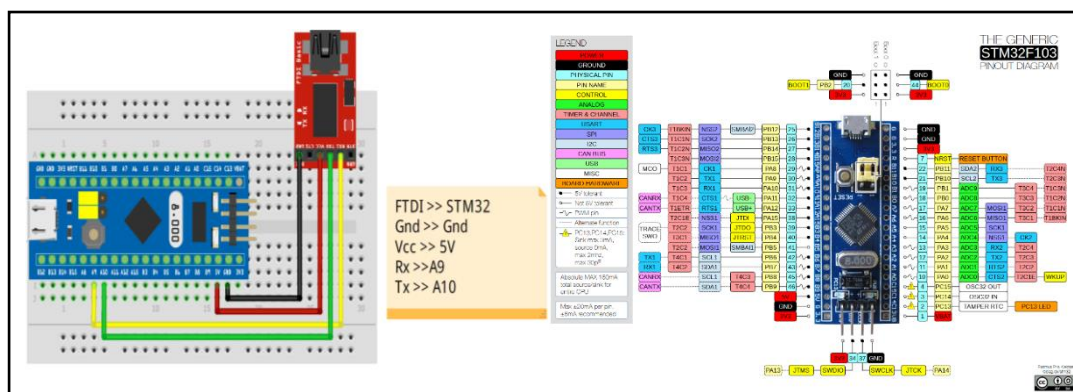
شکل 5: چرخ خورشیدی

2.2. مدارات:

میکروکنترلر STM32F103:

این میکروکنترلر های 32 بیتی از خانواده میکروکنترلر های ARM هستند که به چندین روش قابل برنامه ریزی و استفاده هستند. یکی از این روش ها استفاده از نرم افزار Keil MDK uvision است که بسیار قدرتمند بوده و قابلیت های بسیاری دارد. روش دیگر استفاده از نرم افزار Arduino IDE است که بسیار ساده و قابل فهم میباشد. ما با استفاده از نرم افزار Arduino IDE کد خود را بر روی میکروکنترلر پیاده میکردیم. بوردی که ما از آن در ربات استفاده کردیم بورد STM32F103C8T6 است با فرکانس 72 MHz با حافظه فلش KB64 که شامل یک میکروکنترلر STM32، مدار پروگرامر، مدار تغذیه، درگاه miniUSB و... میباشد. میکروکنترلر های STM را میتوان از چندین طریق پروگرام کرد. ما با استفاده از یک مبدل USB به سریال (FT232RL) و همچنین با استفاده از درگاه miniUSB در نرم افزار Arduino IDE آن را پروگرام میکنیم.

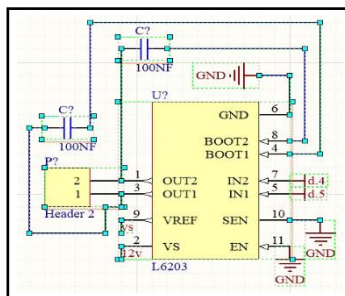
میکروکنترلرهای ARM نسبت به میکروکنترلرهای دیگر مانند AVR مزیت‌های بسیاری دارند از جمله فرکانس بسیار بالاتر، 32 بیتی بودن، طول عمر بسیار بالا، قابل اطمینان بودن و همچنین نسبت به خانواده‌ی AVR کامپایلرهای پیشرفته‌ای برای برنامه‌نویسی این خانواده از میکروکنترلرها در دسترس است.



شکل 6: پایه های STM32F103C8T6 و نحوه اتصال آن به FT232RL

درایور موتور:

L298 و L6203 قطعاتی هستند که برای درایو کردن موتور های DC از آن ها استفاده می شود. درایور L298 با قابلیت درایو کردن هم زمان 2 موتور، درایور بسیار مناسبی از لحاظ وزنی است اما به دلیل جریان دهی پایین امکان از کار افتادن این درایور وجود دارد. در این حال درایور L6203 با قابلیت درایو یک موتور و توان آمپر کشی تا 4 آمپر مورد استفاده تیم ما قرار گرفت.



شکل 7: بایاس درایور L6203

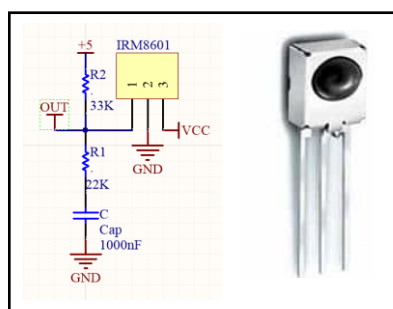
2.3. الگوریتم برنامه نویسی شده و فرمان به موتور ها:

در کد برای حرکت ربات تابع خاصی در نظر گرفته شده که در این تابع عدد سرعت و جهت چرخ موتور وارد میکنیم و ربات با دریافت این اطلاعات با توجه به قطعات استفاده شده در ساخت آن با کم زیاد کردن ولتاژ سرعت موتورها و با جابه جایی بار مثبت و منفی وارد به موتور جهت چرخ را عوض میکند. ما کسیم عدد سرعت 65535 و مینیم آن 0 است اما به دلیل فرکانس کاری حداقل مقدار سرعت جهت حرکت 10000 است که این عدد به صورت تجربی به دست آمده است و جهت چرخ با مثبت و یا منفی بودن عدد تعیین میشود. به طوری که اگر عدد مثبت باشد موتور پادساعت گرد و اگر منفی باشد ساعت گرد می چرخد.

3. تشخیص توپ:

3.1. سنسورها:

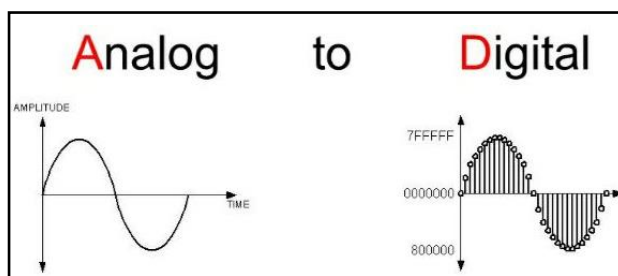
برای تشخیص توپ از 16 سنسور تشخیص دهنده پالس مادون قرمز IRM8601S استفاده کرده ایم. نحوه چیدمان این سنسور ها در 8 جهت حرکت اصلی و 8 جهت فرعی ربات قرار دارد. ما از این سنسور برای تشخیص فاصله توپ تا ربات استفاده کرده ایم تا ربات دروازه بان داشته باشیم. بایاس این سنسور با دو مقاومت 33k و 22k و یک خازن 1 میکرو فاراد به صورت زیر است:



شکل 8: سنسور IRM8601S و بایاس آن

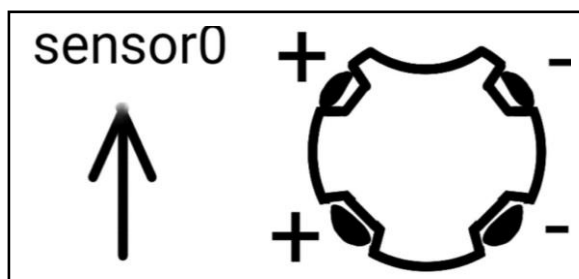
3.2. برنامه نویسی:

برای خواندن سنسور ها از پایه ی A1 در میکروکنترلر STM32 استفاده شده که این پایه دارای ADC می‌باشد.
ADC (Analog To Digital Converter) مبدل آنالوگ به دیجیتال است و کار آن این است که سیگنال های آنالوگ را از سنسور ها دریافت کرده و به دیجیتال تبدیل میکند.



شکل 9: تبدیل پالس آنالوگ به دیجیتال

سنسور ها هنگامی که اشعه ساطع شده از توپ را دریافت میکنند یک موج آنالوگ و هنگامی که اشعه ای دریافت نمیکند یک موج (مانند نمودار) میدهد که ما این تغییرات را به صورت عدد دریافت میکنیم و با توجه به اعداد دریافت شده آگاه میشویم که کدام سنسور توپ را مشاهده میکند سپس با دادن جهت چرخش به موتور ربات را در جهت آن سنسور هدایت می کنیم.



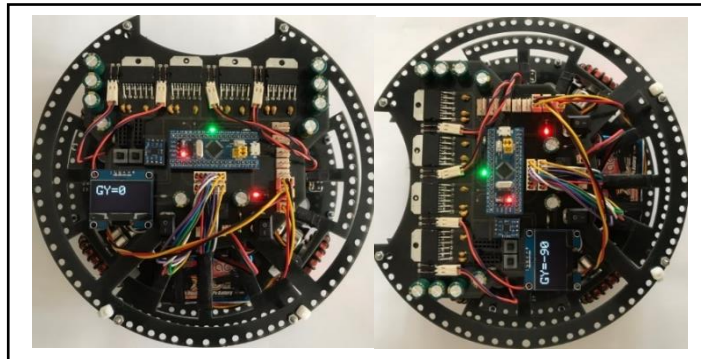
شکل 10: با دادن مقادیر مناسب حرکت سنسور 0 را (به سمت جلو) اجرا میکند.

4. تشخیص دروازه ها:

4.1. سنسور ها:

برای تشخیص اهداف از ماژول ژيروسکوپ GY-25 استفاده کردیم. دلیل استفاده از این ماژول پایین تر بودن میزان نویز آن نسبت به ماژول قطب نمای CMP03 است. این ماژول ژيروسکوپ دارای تراشه MPU-6050 و MCU است و ولتاژ ورودی این ماژول 3 تا 5 ولت است و با دو پروتکل I2C و UART قابل استفاده است که ما از طریق پروتکل UART بین ماژول ژيروسکوپ و میکروکنترلر ارتباط برقرار می کنیم برای برقراری این ارتباط دو پایه Rx و Tx را به میکروکنترلر وصل می کنیم.

ما نیز برای کامپس یا قطب نمای دیجیتال CMPS03 نیز جایی گذاشتیم که اگر احیاناً در هنگام بازی مشکلی برای ماژول GY-25 پیش آمد بتوانیم سریع آن را جایگزین کرده و از آن استفاده کنیم این ماژول نیز از طریق ارتباط I2C با میکروکنترلر ارتباط برقرار می‌کند. همچنین این ماژول با توجه به مغناطیس زمین بازه تغییرات بین 0 تا 255 می‌باشد که بر خلاف GY-25 با پایه های SDA و SCL میکروکنترلر ارتباط برقرار می‌کند.



شکل 12: مقادیر مختلف GY-25 برای جهات متفاوت

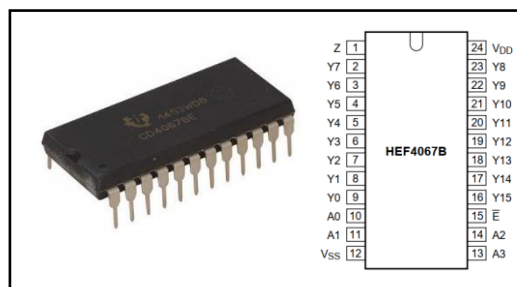
4.2. الگوریتم ها:

با فرستادن کد های اختصاصی ماژول (موجود در دیتا شیت آن) با هر چرخش ربات از سنسور اعدادی را دریافت می‌کنیم که این اعداد را به کمک کد برنامه نویسی شده از $+180$ تا -180 تنظیم می‌کنیم. 0 این بازه هنگامی است که دهانه ربات همراستا با جهتی قرار گیرد که هنگام روشن شدن در آن جهت بوده باشد. ما برای اینکه ربات دروازه حریف را از خودی تشخیص دهد، آن را رو به دروازه حریف قرار می‌دهیم. همچنین برای اینکه ربات در حین حرکت جهت خود را رو به دروازه حفظ کند و توپ را به سمت آن هدایت کند در بازه‌های مختلف اعدادی که از ماژول می‌گیریم، اعدادی را به سرعت موتور ها اضافه می‌کنیم که این اعداد حاصل آزمایش های بسیار هستند.

5. تشخیص خط اوت:

5.1. سنسورها:

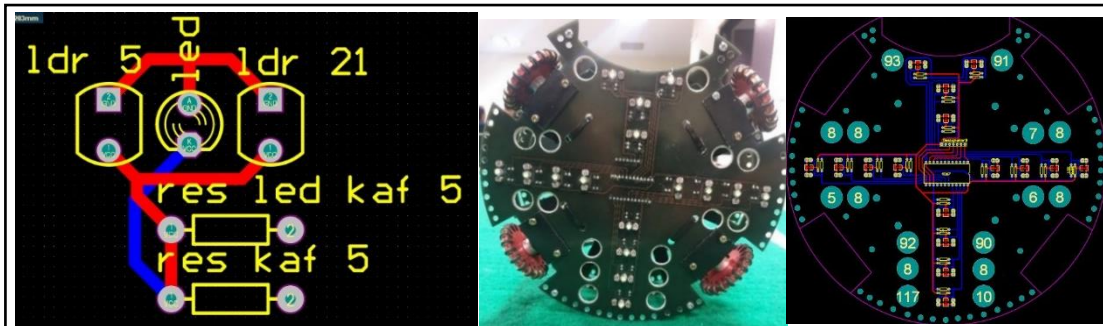
ما با استفاده از 4 عدد سنسور SHARP (در جهت های چپ، عقب و راست و جلوی ربات) و 32 عدد LDR فاصله تا اوت و خط اوت را تشخیص می‌دهیم. ما می‌توانستیم از سنسور های ULTRASONIC هم استفاده کنیم اما برای استفاده از هر کدام از این سنسور ها در کد باید از دو تأخیر 50 میلی ثانیه استفاده کرد که در نتیجه به ازای چهار سنسور ، باید 400 میلی ثانیه تأخیر داد که در کار ما تاثیر منفی زیادی دارد. از طرفی نیز گران بودن این سنسور و نداشتن بودجه کافی عامل دیگر استفاده نکردن از این سنسور است. سنسور های SHARP به پایه های ADC میکروکنترلر STM32 متصل اند. به دلیل کمبود پایه برای اتصال 32 LDR موجود در بخش کفی و 16 سنسور IRM8601 در طبقه دوم ربات، ما با استفاده از دو مولتی پلکسر 4067 در هر دو طبقه ، مقادیر مختلف سنسور های دو طبقه را با پایه Z مولتی پلکسر دریافت می‌کنیم. این مولتی پلکسر 24 پایه دارای 4 پایه آدرس برای باینری کردن اعداد، 16 پایه برای دریافت اطلاعات از سنسور ها و 4 پایه VDD ، VSS ، ENABLE و یک پایه Z است.



شکل 12: پایه های آی سی 4067

5.2. مدارات و چیدمان:

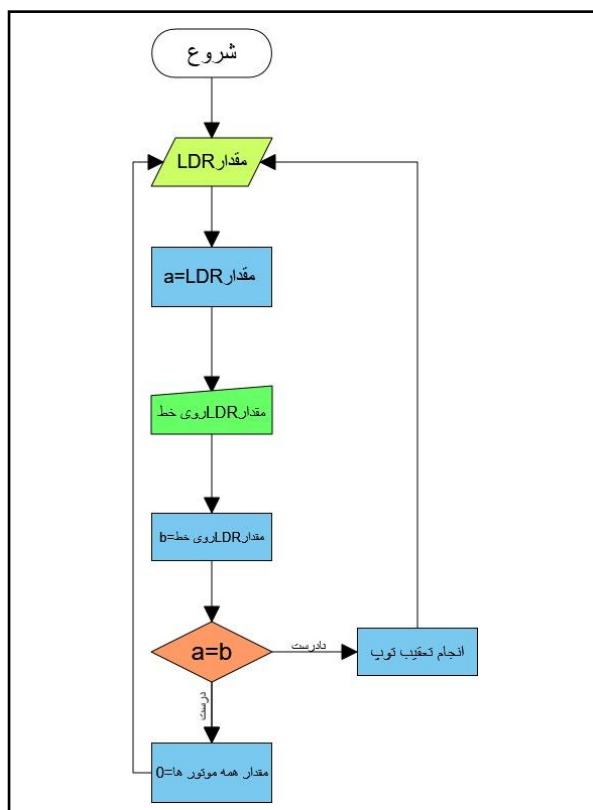
مدار LDR ما شامل دو مقاومت نوری، یک LED و دو مقاومت می‌باشد. 4 عدد LDR در نیمه راست، 4 عدد در نیمه چپ، 4 عدد در نیمه بالا و 4 عدد در نیمه پایینی تعبیه شده است. LED بر زمین نور می‌تاباند و می‌دانیم که بازتاب نور هر رنگ با دیگری متفاوت است. در نتیجه ما می‌توانیم با استفاده از این مدار تفاوت میان رنگ سبز (زمین) و رنگ سفید (خط اوت) را تشخیص داده و مانع عبور ربات از خط اوت شویم.



شکل 13: بایاس و نحوه قرارگیری LDR

5.3. الگوریتم‌ها:

ربات همواره مقدار LDR را می‌گیرد. ما هم مقدار LDR روی خط سفید را به صورت دستی در کد وارد می‌کنیم. با استفاده از داده‌ها، ربات را برنامه‌ریزی می‌کنیم که هنگامی که دو مقدار برابر شدند، ربات در همان جا بایستد و یا به عبارتی مقدار همه موتورها صفر شود. اگر دو مقدار برابر نبودند، ربات به تعقیب توپ پردازد.



شکل 14: الگوریتم تشخیص توپ با استفاده از LDR

6. مهاجم:

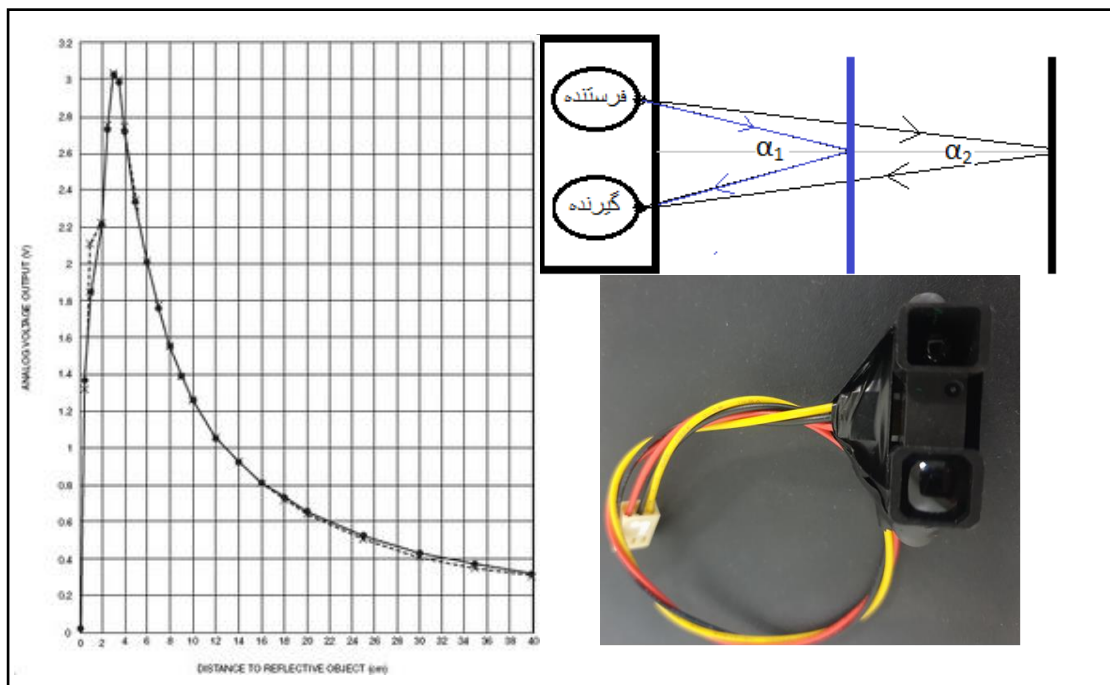
6.1. الگوریتم‌ها:

ربات مهاجم به گونه‌ای برنامه ریزی شده است که در صورت نبود توپ به عقب برنگردد و با استفاده از شارپ‌های چپ و راست در بازه‌ای در عرض زمین قرار بگیرد. ما با استفاده از روش‌های خاصی سعی کردیم که ربات همواره توپ را به دروازه حریف هدایت کند. اطلاعات تکمیلی در نوآوری توضیح داده شده است.

7. دروازه‌بان:

7.1. سنسورها:

سنسور SHARP (GP2D120) دارای دو بخش فرستنده و گیرنده می‌باشد. فرستنده پالس‌های مادون قرمز را ایجاد میکند و به خارج سنسور ارسال می‌کند و بخش گیرنده دارای لنز مخصوصی است که زاویه دریافت پالس را تشخیص داده و با توجه به آن مقداری آنالوگ را به میکروکنترلر STM32 گزارش می‌دهد. این سنسور با ولتاژ کاری 4-5 ولت با خروجی آنالوگ فاصله 4 تا 30 سانتی متری را گزارش می‌دهد. شاید سوال باشد که چرا محدوده کارکرد این سنسور از 4 سانتی متر به بعد است. با توجه به نمودار می‌توان پی برد که این سنسور تا چهار سانتی متری نمی‌تواند خروجی درست بدهد و داده پرت می‌دهد. سنسورهای SHARP به پایه‌های ADC میکروکنترلر STM32 متصل‌اند. هرچه زاویه بیشتر باشد، ربات به دیواره نزدیک‌تر است. نسبت تغییرات زاویه به تغییرات فاصله به صورت یک ولتاژ متغیر در خروجی داده میشود.



شکل 15: نحوه عملکرد سنسور شارپ و نمودار مقادیر مختلف به ازای فواصل متفاوت

7.2. الگوریتم‌ها:

یکی از ربات‌ها به گونه ای برنامه ریزی شده است که از نیمه‌ی زمین جلوتر نرود تا در هنگام حمله‌های غافلگیرانه حریف مانع از گل خوردن شود. این الگوریتم با استفاده از شارپ‌ها پیاده سازی می‌شود که در بخش نوآوری اطلاعات بیشتری در این باره ارائه شده است.

8. نوآوری‌ها:

8.1. طراحی مکانیکی

ما با استفاده از برد های چاپی با کم کردن مقدار سیم های استفاده شده در ربات باعث کاهش وزن بسزایی شدیم.

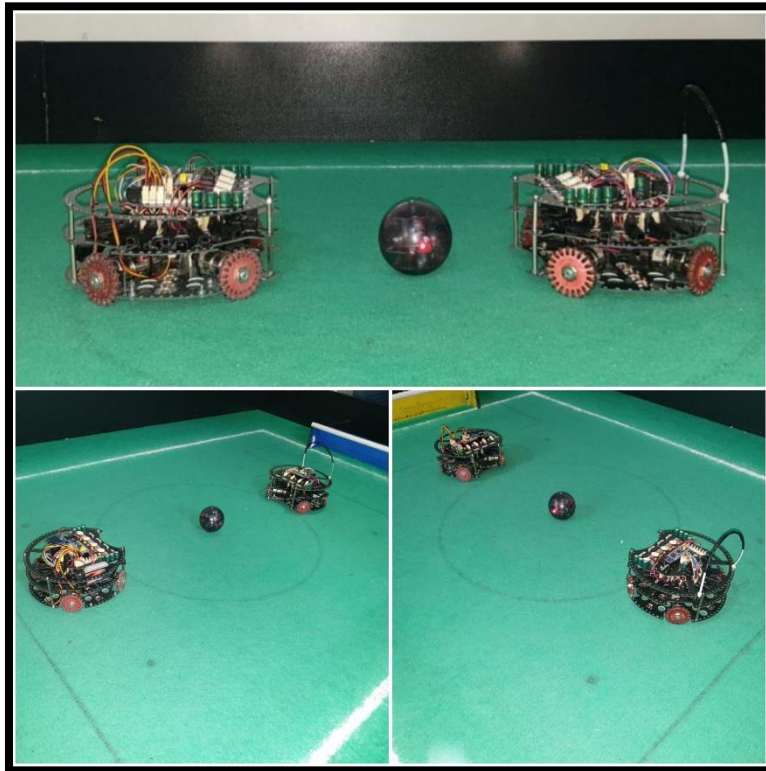
8.2. طراحی مدارات و سنسور ها

ما در کف ربات از حد معمول تعداد بیشتری LDR به کار برده ایم که سبب میشود دقت ربات بیشتر شده و احتمال اوت رفتن آن کمتر شود. همچنین برای هر طبقه از یک آیسی 4067 استفاده کرده ایم که سبب میشود تعداد کمتری سیم و اتصالات از طبقات پایینی به طبقه فوقانی بیاید. برای تعقیب توپ از سنسور های IRM8601S استفاده کردیم که افزون بر میدان دید بیشتر، به ازای هر مقدار فاصله توپ از ربات مقدار معینی میدهد و ما میتوانیم به ازای هر مقدار آن برنامه و الگوریتمی مشخص کنیم. استفاده از برد STM32F103C8T6 سبب میشود تا سرعت و دقت پردازش افزایش یابد. همان طور که می دانید قطب نماهای دیجیتالی براساس میدان مغناطیسی زمین عمل می کنند ولی مشکل اینجاست که هر میدان مغناطیسی دیگری بر خروجی آنها تاثیر می گذارد که این به معنای خطای زیاد آنها است. پس ما در ربات خود از یک ماژول غیر وابسته به میدان مغناطیسی (GY-25) استفاده کردیم اما برای استفاده از یک قطب نمای دیجیتالی نیز مکانی اختصاص دادیم. به سه روش می توان میکروکنترلر را پروگرام کرد و این یکی از مزایای مهم این میکروکنترلر است. ما بجای استفاده از LCD های کاراکتری رایج در بین بقیه تیم ها که دارای محدودیت های زیادی هست، از OLED DISPLAY استفاده کردیم که میتوان داده های مورد نیاز را در همه جای آن و با اندازه مورد نیاز نشان داد و همچنین پایه‌های کمتری هم نسبت به LCD های کاراکتری اشغال میکند.

8.3. برنامه نویسی و الگوریتم‌ها

ربات‌ها به گونه ای برنامه ریزی شده اند که هنگامی که توپ توسط سنسورها تشخیص داده نشد(توپ در زمین نیست) و با استفاده از سنسور های فاصله سنج (SHARP)، فاصله‌ی خود را از دیواره های کناری و پشت زمین اندازه میگیرد و به صورت عدد به میکروکنترلر میفرستد. ما با استفاده از این اعداد، بازه هایی را برای ربات تعریف میکنیم که وقتی ربات در آن بازه ها نبود با جابه جایی در زمین، خود را وارد منطقه مورد نظر میکند. با استفاده از این قابلیت یکی از ربات‌ها نسبت به دیگری مقداری عقب تر جایگیری میکند به طوری که میتواند گفت نقش دروازه بان را ایفا میکند. این قابلیت برای آن است که اگر یکی از ربات های حریف از یکی از ربات‌های ما پیشی گرفت ربات دیگر در عقب آن باشد و از گل خوردن جلوگیری کند. در ارتباط با ربات مهاجم باید به این نکته اشاره کرد که وقتی داور توپ را از زمین خارج میکند ربات با توجه به توضیحات داده شده در بخش 6.1 به عقب بر نمیگردد و در میانه زمین(نسبت به عرض) جایگیری میکند و این قابلیت به ربات مهاجم این اجازه را میدهد تا با قرار دادن توپ توسط داور در نقاط خنثی زمین نسبت به ربات حریف بسیار سریع تر مالکیت توپ را در دست بگیرد؛ علاوه بر آن اگر ربات در فاصله مشخصی از دیواره روبرویی و دیواره چپ و راست قرار بگیرد دهانه خود را چرخانده و رو در رو با دروازه حریف قرار میدهد و توپ را از کناره های زمین به سمت دروازه حریف هدایت میکند. در نتیجه در هنگام حمله احتمال هدایت توپ به اوت پشت دروازه حریف ناچیز میشود و این باعث افزایش احتمال گل زدن میشود.

9. روبات نهایی:



شکل ۱۶: نمایی از ربات

10. تشکر و قدردانی:

با تشکر از مسئولین دبیرستان علامه حلی 5 (دوره دوم) و همچنین با تشکر از مسئولان برگزاری مسابقات که همکاری لازم را با ما داشتند و همه کسانی که مارا در این پروژه یاری کرده‌اند.

11. مراجع:

1- وب سایت رسمی شرکت میکروکنترلر های ST و خلاصه ای از میکرو کنترلر های سری STM32 <https://www.st.com/>

2- وب سایت adafruit <https://www.adafruit.com/>

3- وب سایت رسمی Arduino <https://www.arduino.cc/>

4- <https://www.roboeq.com/>

5- <https://www.alldatasheet.com/>

6- <https://www.eca.ir/>