WRO 2025 | IRAN GILAN

FUTURE ENGINEERS

Qazvin ProBot Documents

فهرست مطالب :

1. مقدمه ..................................................................................................... 3
2. معرفی تیم .............................................................................................. 4
3. مرور کلی پروژه ..................................................................................... 5
4. طراحی ربات .......................................................................................... 6
5. برنامه نویسی ......................................................................................... 10
6. شکل گیری ایده .................................................................................... 13
7. وظایف گروهی ...................................................................................... 16
8. چالش ها و حل مسائل ........................................................................... 17
9. چشم انداز ارتقاء ..................................................................................... 19

مقدمه :

هدف اصلی پروژه ما، طراحی و ساخت یک ربات پیشرفته برای شرکت در مسابقات جهانی رباتیک WRO در دسته Future Engineers است. این رقابت‌ها بستری برای به‌کارگیری خلاقیت، تفکر مهندسی و نوآوری در حل چالش‌های واقعی فراهم می‌کنند، و تیم ما با اشتیاق کامل وارد این مسیر شده است. ما در ابتدای پروژه، تحقیقاتی گسترده در زمینه چالش‌های جهانی انجام دادیم تا بتوانیم ایده‌ای را انتخاب کنیم که هم از نظر فنی چالش‌برانگیز باشد و هم بتواند کاربردی واقعی و تأثیرگذار داشته باشد.

در نهایت، ما پروژه‌ای را انتخاب کردیم که با استانداردهای WRO هم‌راستا بوده و قابلیت رقابت در سطح ملی و بین‌المللی را دارد

تمام مراحل طراحی، ساخت، و برنامه‌نویسی ربات با هدف آماده‌سازی برای رقابت در این مسابقات انجام شده است. ربات ما باید بتواند وظایف پیچیده‌ای را به‌صورت خودکار و با دقت بالا انجام دهد، که این امر مستلزم تلفیق مؤثر بین طراحی مکانیکی، استفاده از حسگرها، و الگوریتم‌های هوشمند بوده است.

در این مستند، مسیر ما از مرحله ایده‌پردازی تا ساخت نهایی ربات به‌طور کامل شرح داده شده است. همچنین توضیحاتی درباره طراحی فنی، چالش‌های تیمی، راه‌حل‌ها و تجربه‌های به‌دست‌آمده از این پروژه آورده شده تا نمایانگر تلاش ما برای رسیدن به موفقیت در مسابقات WRO باشد.

معرفی تیم :

* نام تیم : ProBot|Qazvin
* مربی:

جناب آقای مهدی زاجکانی (سرمربی و راهنمای فنی)

آقای علی صادقی (کمک مربی و مشاور تیم)

* اعضای گروه :

**مصطفی کوچک پور :**

متولد 1381 ، دانشجو کارشناسی مهندسی عمران دانشگاه رجاء قزوین

**مهراد میرمحمدعلی :**

متولد 1382 ، دانشجو کارشناسی مهندسی عمران دانشگاه آزاد اسلامی واحد علوم و تحقیقات تهران



مرور کلی پروژه :

پروژه‌ی ما طراحی و ساخت یک ماشین خودران است که با هدف شبیه‌سازی عملکرد اتومبیل خودران هوشمند انجام شده است. این ربات طوری طراحی شده که بتواند حرکت کاملاً خودمختار (Autonomous Driving) را در شرایط شبیه‌سازی‌شده اجرا کند، و در عین حال، پایه‌گذار مفاهیم کاربردی برای سیستم‌های حمل‌ونقل هوشمند در دنیای واقعی باشد.

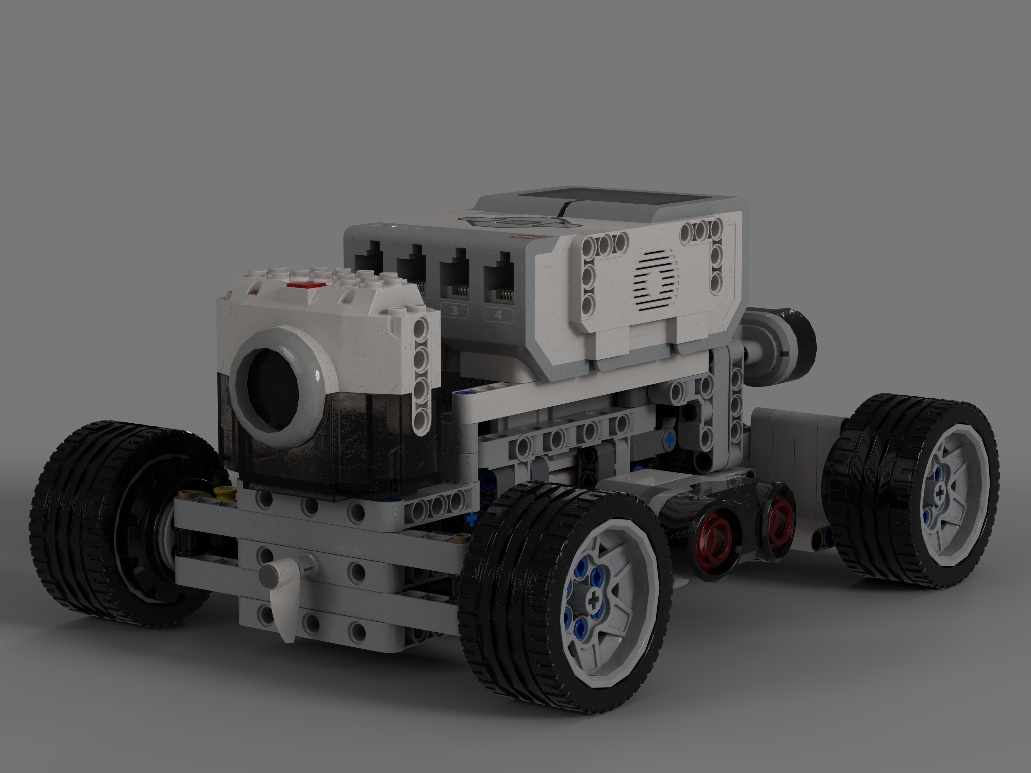
در مراحل اولیه‌ی طراحی، تیم ما با مطالعه‌ی دقیق بر روی ویژگی‌های مهندسی خودروهای واقعی از جمله نوع سیستم حرکتی، طول محور و نحوه‌ی تقسیم وزن، ایده‌پردازی پروژه را آغاز کرد. همچنین در ادامه برای جمع آوری اطلاعات کاربردی در ابعاد شبیه سازی ، با تحلیل ساختار ماشین‌های کنترل از راه دور (RC)، توانستیم اصولی مانند پایداری، توزیع وزن، و چابکی در حرکت را در طراحی خود لحاظ کنیم.

ربات ما نه‌تنها از نظر ظاهری و مهندسی الهام‌گرفته از نمونه های دنیای واقعی است، بلکه عملکرد آن نیز شامل قابلیت‌هایی مشابه با ماشین‌های خودران است؛ مانند تشخیص مسیر و تقاطع و موانع ، تصمیم‌گیری خودکار، و حرکت دقیق در محیط. این پروژه به ما این امکان را داد تا مفاهیم پیشرفته‌تری از مهندسی مکانیک، حسگرها، و الگوریتم‌های تصمیم‌گیری را در عمل به‌کار ببریم.

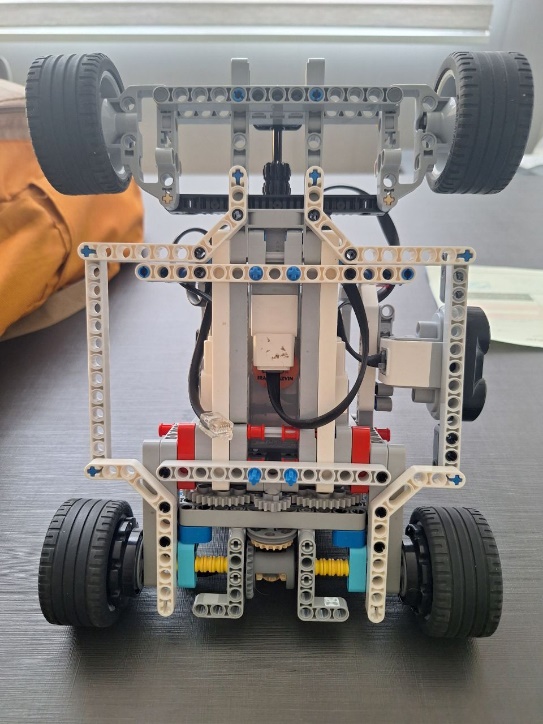
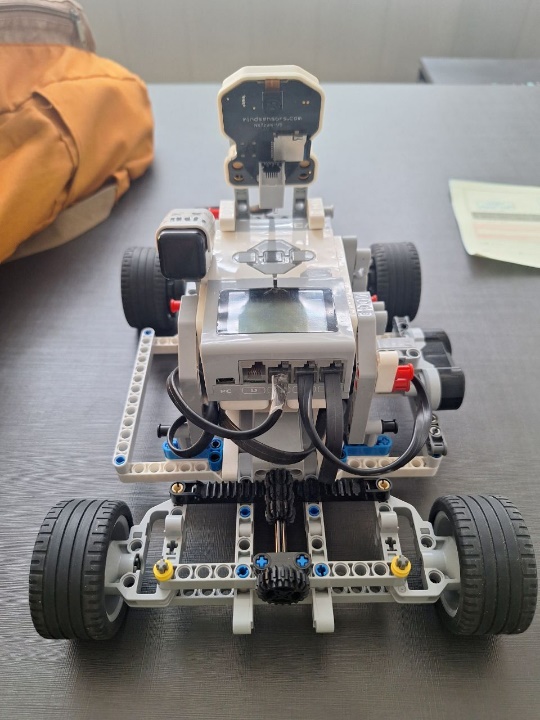
طراحی ربات :

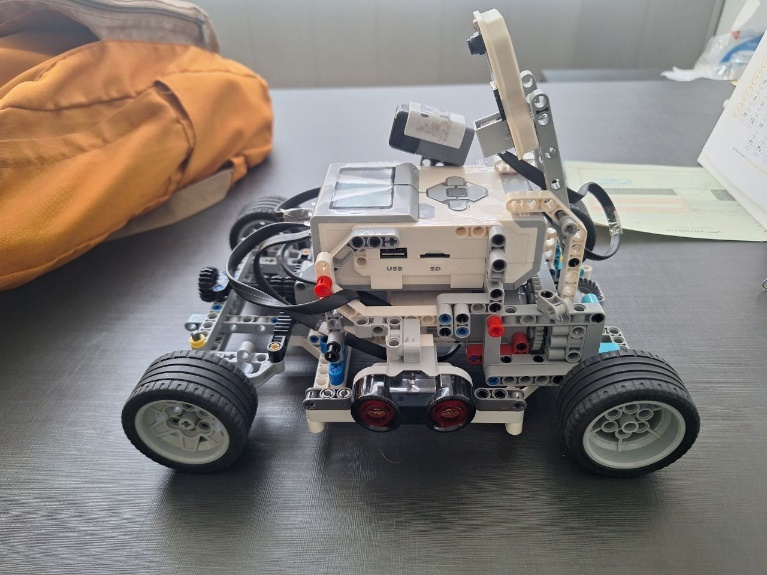
طراحی ربات ما با هدف شبیه‌سازی یک خودروی خودران آغاز شد و طی مراحل مختلف با بازبینی‌های مداوم، به شکل نهایی خود رسید. از همان ابتدا تلاش کردیم تا بین ساختار مکانیکی، انتخاب قطعات، و نیازهای عملکردی ربات تعادل برقرار کنیم.

در مراحل اولیه، از ساختار ماشین‌های RC به‌عنوان پایه طراحی استفاده کردیم. این ساختار به دلیل پایداری بالا، مرکز ثقل مناسب، و قابلیت مانور خوب در فضای محدود، گزینه مناسبی برای شروع بود.



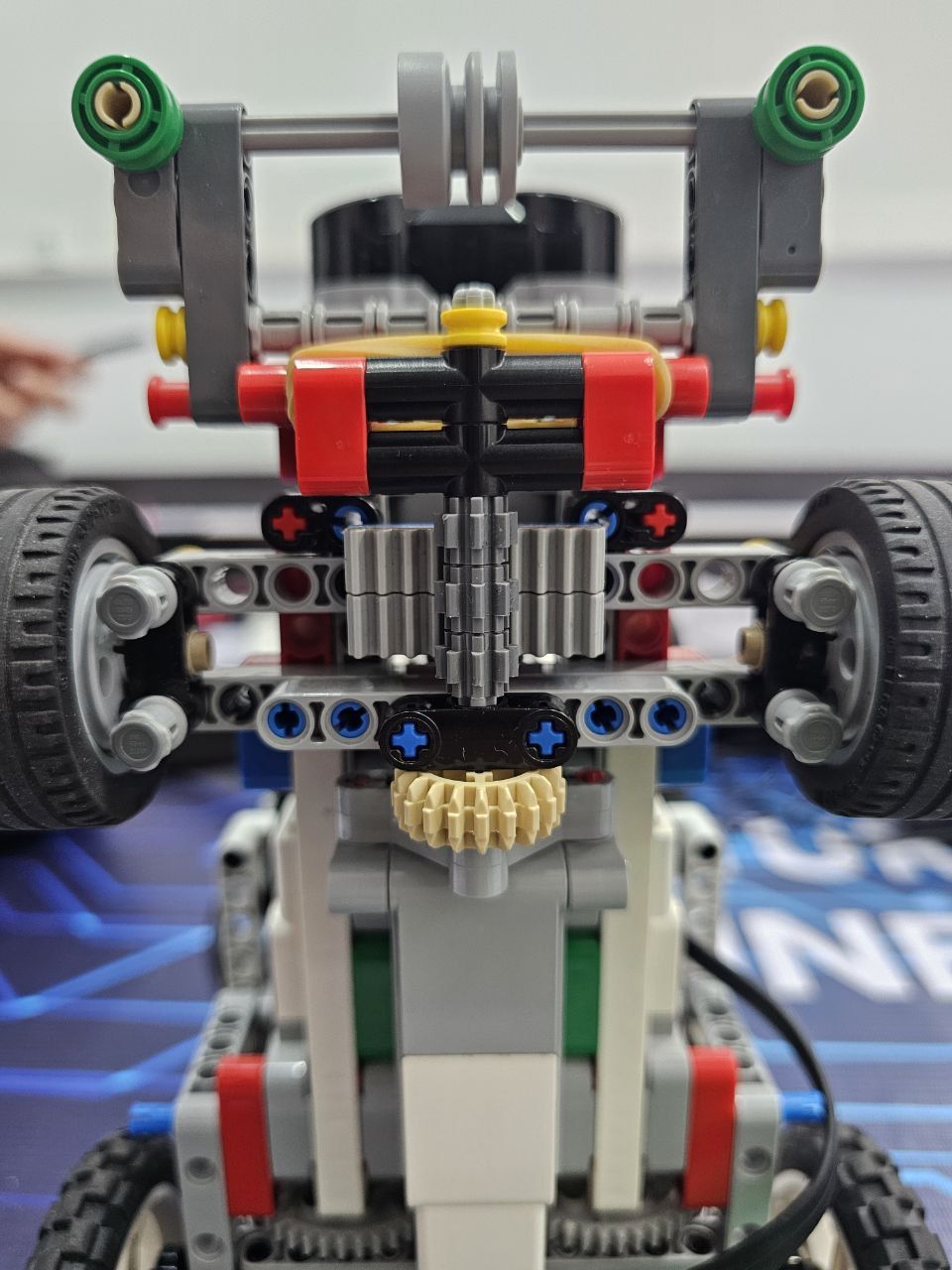
با این حال، در ادامه مسیر و با شناسایی نیازهای دقیق‌تر پروژه، متوجه شدیم که باید تغییراتی اساسی در سازه اعمال کنیم تا پاسخگوی اهداف فنی و رقابتی ما باشد.

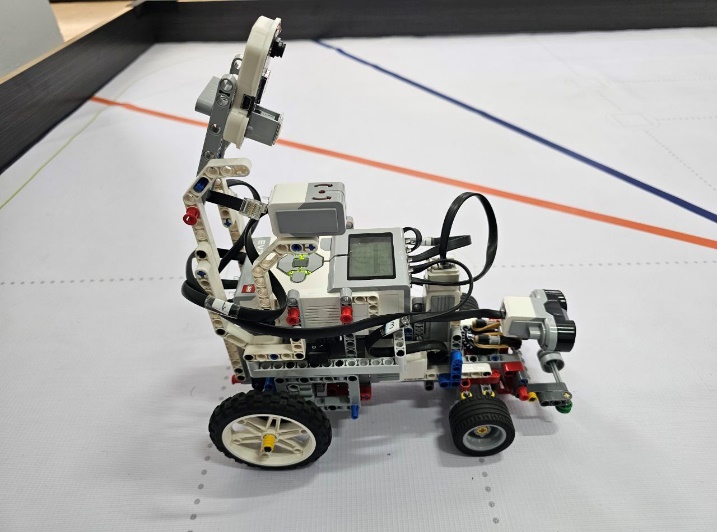
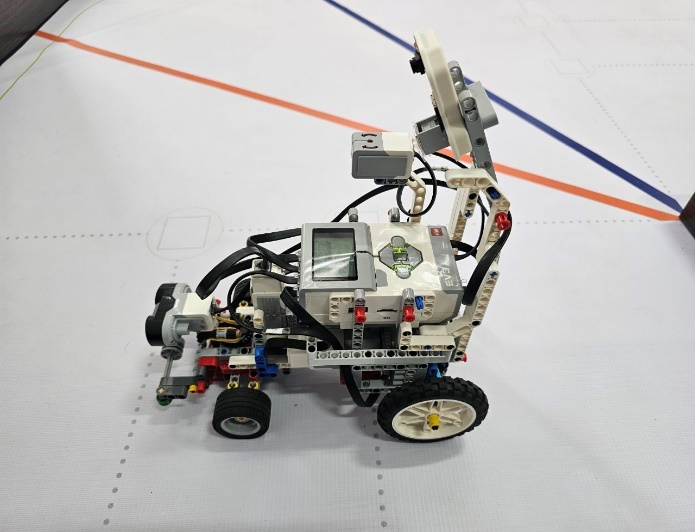


در نهایت، ساختار ربات با توجه به موارد زیر بازطراحی شد:

* بهینه‌سازی طول محور چرخ‌ها برای کنترل بهتر در سرعت بالا



* بازتوزیع وزن برای تعادل بیشتر هنگام پیچ‌ها
* طراحی دوباره شاسی برای نصب دقیق سنسورها و ماژول دوربین



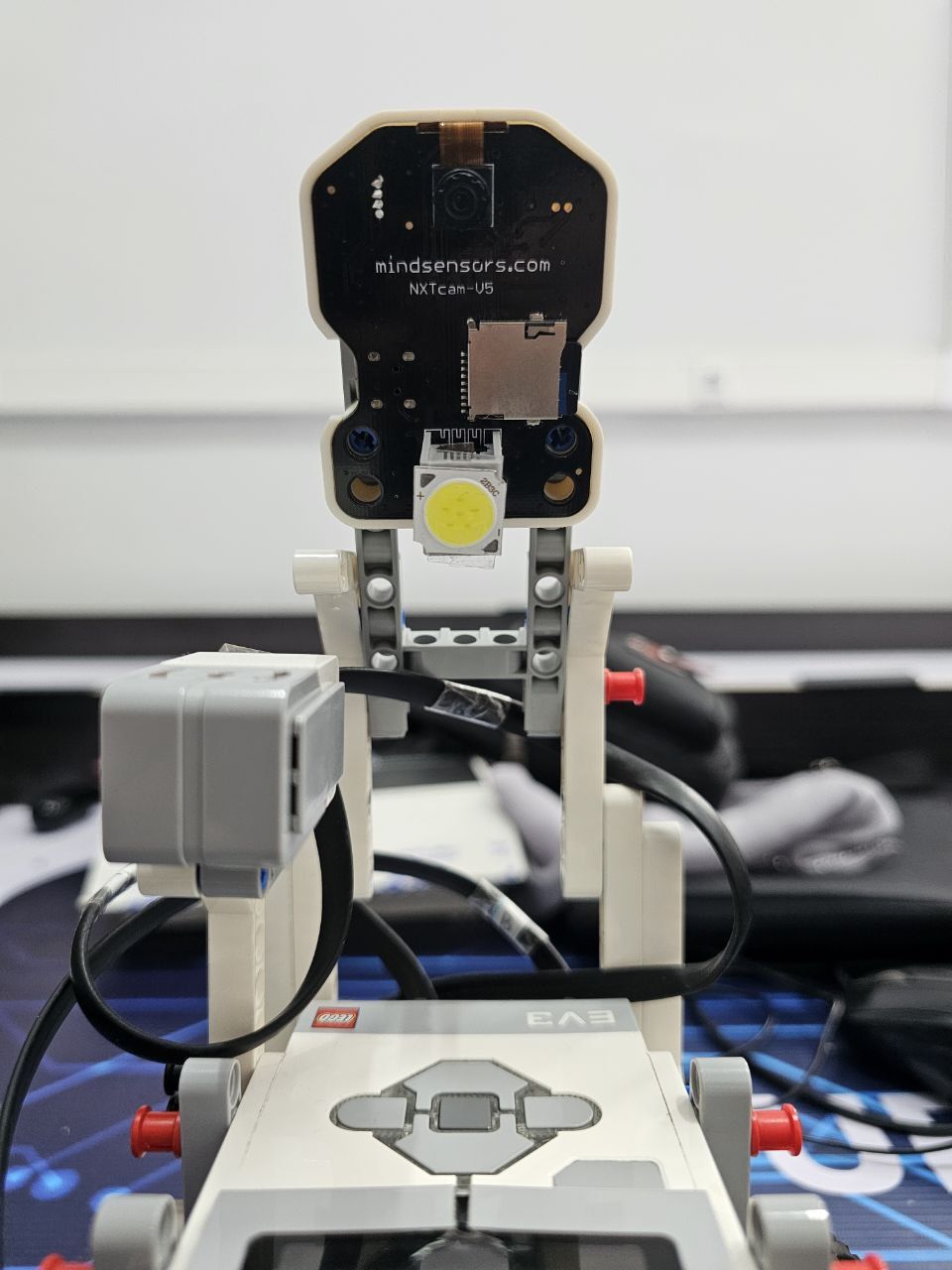
برای هوشمندسازی و اجرای پردازش‌های اصلی، از پلتفرم EV3 شامل اجزای زیر استفاده کردیم:

سنسورهای EV3 برای تشخیص فاصله، موانع، و دنبال‌کردن مسیر

پردازنده EV3 Brick به‌عنوان مغز ربات، برای اجرای الگوریتم‌های برنامه‌ریزی‌شده

استفاده از قطعات مکانیکی موجود در پکیج رسمی EV3 LEGO Mindstorms برای ساخت شاسی و نگهدارنده‌ها

به‌منظور ارتقاء توانایی ربات در درک محیط، از دوربین NXT-CAM نیز استفاده کردیم. این دوربین امکان جمع‌آوری داده‌های تصویری و تحلیل بصری را فراهم کرد، که در ترکیب با اطلاعات سنسورها، به ربات اجازه می‌دهد تصمیم‌گیری‌های دقیق‌تری در طول حرکت داشته باشد.



فرآیند طراحی، یک مسیر تکرارشونده از آزمون و خطا بود. بارها سازه را اصلاح کردیم تا به شکلی برسیم که هم عملکرد واقعی ماشین‌های خودران را شبیه‌سازی کند و هم با محدودیت‌های زمان، فضا و تجهیزات سازگار باشد. ساختار نهایی ربات، نتیجه بررسی‌های فنی، تجربه عملی، و همکاری تیمی در طراحی یک سیستم مکانیکی منسجم و هوشمند است.

برنامه نویسی :

برای برنامه‌نویسی ربات، تصمیم گرفتیم از محیط توسعه‌ی LEGO Mindstorms EV3 استفاده کنیم. انتخاب این زبان و پلتفرم به دلایل مختلفی انجام شد که مهم‌ترین آن‌ها سازگاری کامل با سخت‌افزار EV3 ، پشتیبانی کامل NXT-CAM5 و تسلط کامل اعضای تیم به این محیط برنامه‌نویسی بود.

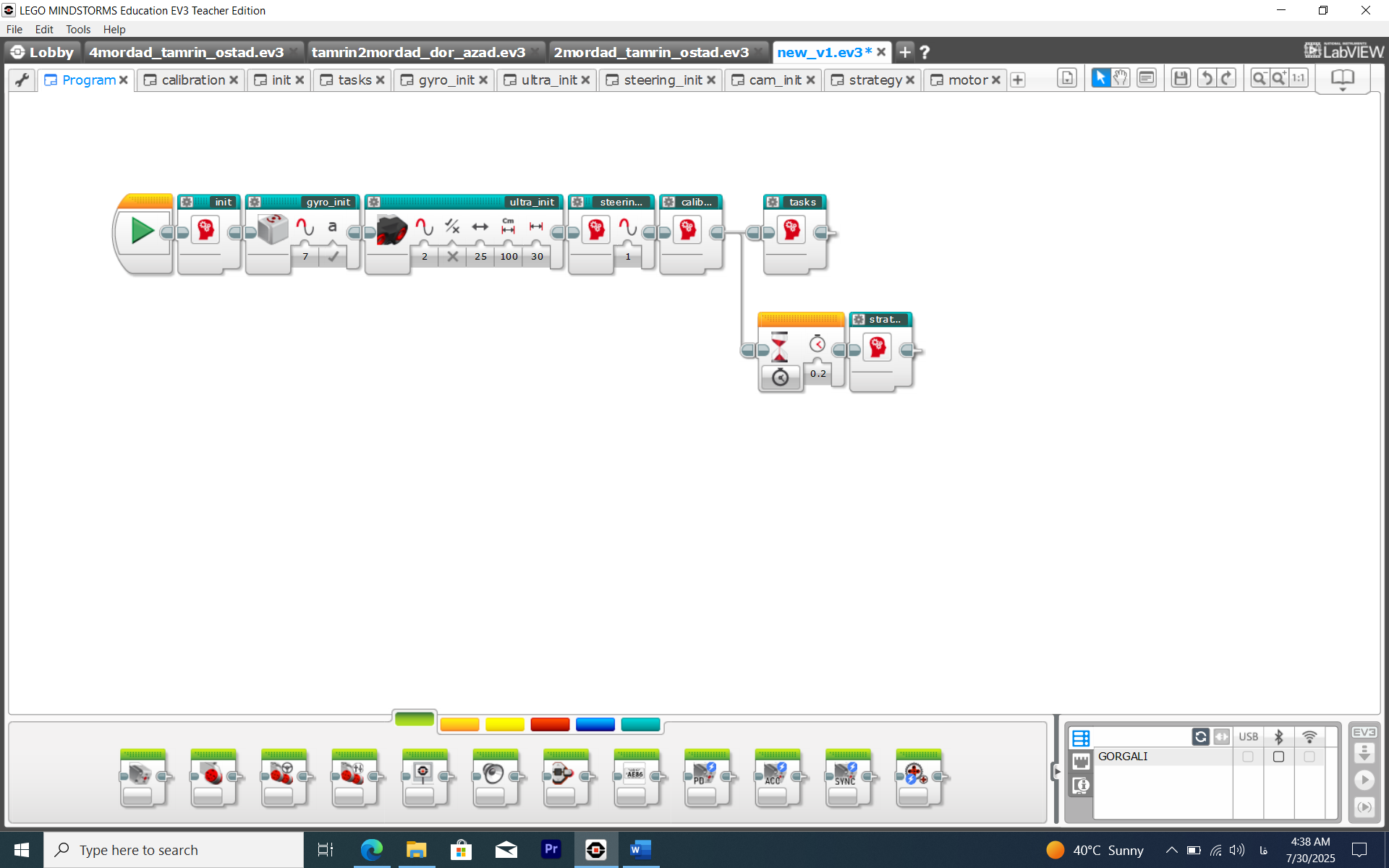
محیط Mindstorms به ما این امکان را می‌دهد تا با استفاده از بلوک‌های گرافیکی، منطق‌های پیچیده‌ی کنترلی را به‌صورت دقیق و قابل درک پیاده‌سازی کنیم. با وجود ظاهر گرافیکی، تیم ما توانست الگوریتم‌هایی با رفتارهای پویا و تعاملی طراحی کند که در عملکرد خودران ربات نقش اساسی دارند.

با توجه به اینکه زبان Mindstorms کاملاً برای سخت‌افزار EV3 بهینه‌سازی شده است، ما توانستیم بدون دغدغه‌ی ناسازگاری نرم‌افزاری، روی منطق کنترل و تست‌های مکرر تمرکز کنیم. این موضوع نقش مهمی در پیشرفت سریع‌تر پروژه و افزایش پایداری عملکرد ربات داشت.

در صورت نیاز به توسعه‌ی آتی یا انتقال پروژه به یک سطح پیشرفته‌تر، امکان بازنویسی منطق‌ها در زبان‌های سطح پایین‌تر مانند Python یا C نیز در نظر گرفته شده است.

ساختار کلی برنامه از نظر منطقی به سه بخش اصلی تقسیم شده است .

بخش ابتدایی (init) مختص به مقدار دهیه اولیه ، بخش دوم و اصلی تابع strategy (main) و بخش آخر تابع task است که خود شامل توابع محاسبه‌گر جزئی تری است .



**Strategy Function (تابع استراتژی)**این بخش نقش اصلی در تصمیم‌گیری و اجرای مأموریت‌ها را بر عهده دارد.  
پس از مقداردهی‌های اولیه (init) ، تابع استراتژی اجرا می‌شود.  
تابع Strategy به‌صورت مداوم از داده‌های محاسبه‌شده توسط سایر توابع (task/loops) استفاده می‌کند تا تصمیم بگیرد که ربات در هر لحظه چه کاری انجام دهد (مثلاً: حرکت، توقف، چرخش، واکنش به مانع).  
این تابع حکم «مغز تصمیم‌گیرنده» را دارد و همه رفتارهای حرکتی از طریق آن هدایت می‌شوند.

**Task function (تابع تحلیل)**   
در کنار تابع استراتژی، یک یا چند تابع مستقل به‌صورت Loop (حلقه‌های موازی) اجرا می‌شوند که وظیفه‌ی آن‌ها خواندن و تحلیل مکرر داده‌های سنسورها و دوربین است.  
این توابع با نام‌هایی مانند task / Loop مشخص شده‌اند.  
آن‌ها به‌طور مداوم، اطلاعات محیطی (مثلاً فاصله، رنگ، موقعیت) را از سنسورهای EV3 و دوربین NXT-CAM دریافت کرده و پردازش می‌کنند.  
خروجی این بخش به شکل مقادیر عددی یا شرایط منطقی در اختیار تابع Strategy قرار می‌گیرد.

**فرایند اجرای برنامه :**  
1. مقداردهی اولیه (init)  
تعریف مقادیر اولیه متغیرها  
آماده‌سازی سخت‌افزارها (موتورها، سنسورها، دوربین)  
2. اجرای هم‌زمان Strategy و Taskها  
Strategy: هدایت کلی و تصمیم‌گیری  
Task/Loop: تحلیل داده های سنسورها  
3. تعامل بین ماژول‌ها  
داده‌ها توسط Task به‌صورت مستقیم درStrategy مورد استفاده قرار می‌گیرند.  
این معماری باعث می‌شود ربات بتواند به‌صورت واکنشی، هوشمند، و سریع در محیط عمل کند.

**شکل‌گیری ایده (Idea Formation)**

فرآیند شکل‌گیری ایده‌ی ربات ما نتیجه‌ی ترکیبی از تحلیل مهندسی، الگوبرداری از دنیای واقعی، آزمون و خطا، و بهینه‌سازی مستمر بود. هدف اصلی، ساخت یک ماشین خودران با رفتار واقعی، کنترل دقیق، و توانایی انجام مانورهای پیچیده در محیط مسابقه بود.

۱. الهام از خودروهای واقعی

در مرحله نخست، تمرکز ما بر دستیابی به فرمی مناسب برای حرکت دقیق، پارک، و چرخش در پیچ‌ها بود. برای رسیدن به یک طراحی واقع‌گرایانه، ما از خودروهای واقعی الگوبرداری کردیم. پس از بررسی چندین نمونه، در نهایت Peugeot 206 را به‌عنوان مدل پایه انتخاب کردیم.

از این خودرو، موارد زیر استخراج و در مقیاس کوچک پیاده‌سازی شد:

نسبت فاصله بین دو محور چرخ‌ها (Wheelbase)

زاویه چرخش چرخ‌ها هنگام پیچیدن (Steering Geometry)

۲. تطبیق در مقیاس کوچک با ماشین‌های RC

با توجه به شرایط شبیه سازی و محیط مسابقه، لازم بود طراحی‌ را در ابعاد کوچک‌تر اجرا کنیم. بنابراین، از ماشین‌های RC (کنترل از راه دور) نیز الهام گرفتیم. آن‌ها ویژگی‌هایی مانند وزن سبک، فرمان‌پذیری بالا، و قدرت مانور خوب در فضاهای کوچک دارند. صنعت سرگرمی ماشین های RC با داشتن انجمن های اینترنتی فعال باعث میشود اطلاعات گسترده، ایده های کاربردی و طرح های متن باز زیادی برای جمع آوری، تحلیل و ایده پردازی فراهم باشد.

از ماشین‌های RC موارد زیر را بررسی و در ربات خود اعمال کردیم:

سیستم فرمان چرخ‌ها (steering linkage): تنظیم دقیق زاویه چرخ‌ها برای چرخش نرم و دقیق در پیچ‌ها

تقسیم وزن در ربات: جای‌گذاری موتور و باتری برای ایجاد تعادل مرکز ثقل

استفاده از چرخ‌های بزرگ‌تر در عقب (چرخ‌های محرک) برای افزایش طول حرکت

کاهش سایز چرخ‌های جلو برای کاهش سطح تماس و چرخش روان‌تر

در نهایت، طراحی ما ترکیبی از این دو الهام شد

ساختاری شبیه به یک خودرو سبک، با ویژگی‌های چرخشی و دینامیکی نزدیک به یک تراکتور کوچک.

۳. مسیر بهینه‌سازی سیستم تشخیص دیواره‌ها

برای تشخیص دیواره‌ها در محیط مسابقه، در ابتدا از ۳ سنسور اولتراسونیک استفاده کردیم: در قسمت جلو، چپ، و راست ربات. این طراحی در مراحل اولیه به ما کمک کرد تا در هر لحظه فاصله ربات از محیط را در سه جهت بررسی کنیم.

اما پس از تست‌های عملی و تحلیل عملکرد، دریافتیم که این سیستم پیچیده و دارای خطاهایی از جمله تداخل فرکانسی در سنسور ها است. در نتیجه تصمیم گرفتیم آن را بهینه‌سازی کنیم:

نسخه نهایی شامل فقط یک سنسور اولتراسونیک است

این سنسور بر روی یک محور چرخان (rotating mount) قرار گرفت

با چرخاندن سنسور به اطراف، داده‌های لازم از زوایای مختلف به‌دست می‌آید

این راهکار، دقت کافی را فراهم کرد، در عین حال وزن، فضای مصرفی، و پیچیدگی کدنویسی را کاهش داد

هدف از این سنسور چرخان این بود که ربات بتواند:

در زمان های متفاوت بتواند زوایای مختلفی را پوشش دهد

مسیر خود را به سمت دیواره‌ها اصلاح کند

تا رسیدن به دیواره‌ی انتهایی ادامه دهد

در نقطه مناسب، گردش کرده و مسیر بازگشت یا چرخش را انجام دهد

**وظایف گروهی :**

در این پروژه، وظایف میان اعضای تیم به‌گونه‌ای تقسیم شد که هر فرد با توجه به توانمندی‌ها و علاقه‌مندی‌هایش نقش مؤثری در پیشبرد پروژه داشته باشد. تمرکز ما بر همکاری مؤثر، مسئولیت‌پذیری، و یادگیری متقابل بود.

مصطفی کوچک پور – مسئول برنامه‌نویسی ربات :

طراحی ساختار اصلی نرم‌افزار شامل بخش‌های Strategy و Task

توسعه منطق کنترل ربات بر اساس داده‌های حسگرها و شرایط محیطی

پیاده‌سازی کدهای مربوط به تشخیص موانع، مسیر، و تنظیم سرعت

بهینه‌سازی حلقه‌های لوپ و مدیریت ورودی‌های حسگرها برای افزایش دقت تصمیم‌گیری

تست مکرر عملکرد نرم‌افزار و رفع باگ‌های احتمالی در مراحل مختلف پروژه

مهراد میرمحمدعلی – مسئول طراحی و ساخت مکانیکی ربات :

طراحی اولیه شاسی با الهام از خودروهای واقعی و RC

محاسبه دقیق نسبت طول محور، تقسیم وزن، و پایداری در پیچ‌ها

انتخاب و نصب مناسب موتورها، چرخ‌ها، و قطعات مکانیکی

طراحی سازه برای نصب سنسورها و ماژول دوربین به شکلی کارآمد و پایدار

تست عملکرد مکانیکی در سناریوهای واقعی و اصلاح ساختار با توجه به نیازهای پروژه

چالش ها و حل مسائل :

چالش‌ها و حل آن‌ها (Challenges and Problem Solving)

در طول مسیر طراحی، ساخت و برنامه‌نویسی ربات، با چالش‌های متعددی روبه‌رو شدیم. این چالش‌ها هم در حوزه مکانیکی و هم در حوزه نرم‌افزاری بودند و نیاز به تحلیل دقیق، آزمایش‌های مکرر و بازنگری در طراحی داشتند. در ادامه به مهم‌ترین چالش‌ها و راه‌حل‌های اتخاذ شده اشاره می‌کنیم:

۱. برخورد ربات با موانع در پیچ‌ها

یکی از چالش‌های اساسی ما، برخورد ربات با موانع حین چرخش یا عبور از پیچ‌ها بود. در این شرایط، فاصله‌ی ربات با موانع به‌درستی کنترل نمی‌شد و موجب خطا در عملکرد و برخورد با موانع می‌گردید.

راه‌حل:

برای حل این مشکل، در برنامه‌نویسی ربات یک ماژول کنترلی اضافه کردیم که در زمان نزدیک شدن به موانع، سرعت ربات را کاهش داده و در صورتی که فاصله به مقدار بحرانی برسد، دستور عقب‌گرد (Reverse) صادر می‌کند. این کار باعث افزایش دقت و ایمنی در حرکت‌های نزدیک به موانع شد.

۲. خطای سنسور رنگ (Color Sensor)

در مراحل اولیه، از سنسور رنگ برای تشخیص خطوط رنگی گوشه‌های زمین استفاده می‌کردیم. اما به دلیل تغییرات شرایط نوری محیط و فرکانس کاری سنسور، مقادیر عددی دریافتی از سنسور هر بار متفاوت بود و تشخیص دقیق رنگ‌ها دچار مشکل می‌شد.

راه‌حل:

با توجه به عدم پایداری در خوانش داده‌ها، تصمیم گرفتیم سنسور رنگ را حذف کرده و به‌جای آن از سنسور اولتراسونیک برای تعیین موقعیت و فاصله از دیواره‌ها استفاده کنیم. این کار باعث تغییر کلی و بهینگی الگوریتم، افزایش دقت و ثبات عملکرد در شرایط نوری مختلف شد.

۳. خطای تشخیص دوربین (Object Detection Errors)

در بخش بینایی ماشین (با استفاده از دوربین NXT-CAM یکی از چالش‌های غیرقابل اجتناب، اشتباه در شناسایی آبجکت‌ها توسط دوربین بود. در برخی موارد، حتی خطوط قرمز روی زمین به اشتباه به‌عنوان مانع قرمز شناسایی می‌شدند که موجب اختلال در عملکرد ربات می‌شد که هنوز راه حلی برای آن نداریم .

**چشم انداز ارتقاء (Future Improvements) :**

با بررسی عملکرد ربات و چالش‌هایی که در مراحل تست و اجرای مأموریت با آن‌ها مواجه شدیم، ایده‌هایی برای ارتقاء پروژه در آینده شکل گرفت. برخی از مهم‌ترین مسیرهای بهبود عبارت‌اند از:

۱. پیاده‌سازی الگوریتم‌های خودیادگیر(Self-Learning Algorithms)

یکی از اهداف بلندمدت ، توسعه یک سیستم هوشمند با قابلیت یادگیری از تجربه است. در این روش، ربات با هر بار طی کردن مسیر، داده‌هایی از عملکرد خود (مثل زمان، فاصله تا مانع، انحراف از مسیر) ذخیره می‌کند و در تکرارهای بعدی از این اطلاعات برای بهبود تصمیم‌گیری استفاده می‌کند.

راهکار پیشنهادی:

استفاده از الگوریتم‌های Reinforcement Learning یا Q-Learning برای یادگیری رفتار بهینه در محیط.

ثبت و تحلیل داده‌ها از سنسورها در هر اجرا و استفاده از آن برای Behavioral Adaptation.

طراحی ساختار داده‌ای برای ذخیره تجارب گذشته و به‌روزرسانی استراتژی بر اساس آن.

این الگوریتم می‌تواند به‌صورت آنلاین (Online Learning) یا آفلاین (Offline Training) پیاده‌سازی شود، بسته به محدودیت‌های سخت‌افزاری.

۲. بهینه‌سازی ساختار برنامه‌نویسی(Programming Architecture Refinement)

در ساختار فعلی برنامه، تقسیم‌بندی بین Strategy و Task انجام شده که ساختار خوبی برای کنترل وظایف مختلف است. اما در آینده می‌توان:

از State Machines استفاده کرد برای داشتن حالات مشخص (مثل حرکت مستقیم، تشخیص مانع، چرخش).

تعریف ماژول‌های مستقل برای هر حسگر یا عملکرد، برای افزایش Modularity و Debugging ساده‌تر.

افزودن Logging System برای ذخیره رخدادها و عملکرد ربات، جهت تحلیل دقیق‌تر پس از تست‌ها.

۳. تست‌های خودکار و محیط‌های شبیه‌سازی(Auto-Testing & Simulation)

برای کاهش نیاز به تست‌های فیزیکی (که زمان‌بر و پرهزینه هستند)، استفاده از شبیه‌سازهایی مثل Webots یا Gazebo پیشنهاد می‌شود

پیاده‌سازی محیط مجازی برای تمرین و آزمایش رفتار ربات

تست سریع الگوریتم‌ها بدون آسیب به ربات فیزیکی

استفاده از داده‌های مجازی برای آموزش الگوریتم‌های یادگیرنده