

موضوع پروژه: ربات دیفرانسیلی خود تعادل شونده

مقدمه:

ربات‌های دیفرانسیلی خود تعادل شونده، ربات‌هایی هستند که از الگوریتم‌های بلادرنگ (Real-Time) استفاده می‌کنند و به صورت مداوم، خطاهای موجود را نمونه‌برداری و پردازش کرده و همواره تلاش می‌کنند این مشکلات را رفع کنند. این ربات‌ها در صنایع پزشکی، عکاسی (gimbal)، ربات‌های خانگی و خدمتکار، ربات‌های انسان‌نما، ربات‌های حمل‌کننده (مانند ربات‌های شرکت بوستون داینامیکس) و غیره استفاده می‌شوند.

قطعات استفاده شده:

فناوری‌های استفاده شده در قسمت نرم‌افزار	قطعات الکترونیکی	قطعات مکانیکی
PWM	L298 درایور موتور	برد برد
Interrupt	و برد چاپ شده Atmega32	اسپیسر
ADC	های آبی و سفید LED	دی سی موتور براشلس ۱۲ ولت
I/O	IR LED TX and RX	
Timer	Resistor	
Serial	Potensiometer	

چالش‌ها:

چالش‌هایی که زیاد با آن‌ها درگیر شدم و باعث شد کمی روند پروژه و رسیدن به نتیجه مطلوب طولانی شود به شرح زیر است:

تأمین برق مورد نیاز: با توجه به کمبود امکاناتی مانند رگولاتور، منبع تغذیه DC، عدم وجود فیوز برای ساخت مدار تأمین کننده متغیر و همچنین نیاز به ولتاژهای مختلف با توجه به استفاده از موتورهای ۱۲ ولت و کمبود آمپر خروجی مستقیم از میکرو و نیاز به استفاده از درایور موتور که باعث می‌شود مدار، جریان و ولتاژکشی محسوسی داشته باشد، در وهله اول مجبور به تفکیک ولتاژهای موتور و مدار اصلی شدم و سپس با آزمون و خطا و دیباگ کردن‌های متعدد توسط مولتی‌متر، در نهایت با گذشت زمان زیاد، توانستم قسمت‌های مربوط به تغذیه مدارها را پایدار کنم.

دقت پایین سنسورهای IR (مادون قرمز) برای تشخیص زمین و دشوار بودن مانیتور کردن مقدارهای خوانده شده توسط میکرو. **حساسیت به نور:** با توجه به اینکه از سنسورهای مادون قرمز برای تشخیص فاصله استفاده شده است، وابستگی محیطی پیدا می‌کنیم که ممکن است بر اساس رنگ و یا جنس زمین مورد استفاده، عددها و عملکردهای متفاوت داشته باشیم. برای حل این چالش، با استفاده از مقاومت‌های متغیر (پتانسیومتر)، استفاده از LED برای نوردهی به کف زمین و همچنین بهره‌گیری از روش‌های نرم‌افزاری مانند **setup** اولیه سنسورها با زمین مورد نظر، تمام این سه مشکل را تا حد مناسبی کنترل کرده و توانستم وضعیت نسبتاً ثباتی برای این متغیرها پیدا کنم که نیازمند آزمون و خطای زیادی بود.

تعادل مکانیکی ربات: از آنجایی که هدف ربات تعادل خود مختار در وضعیت‌های متفاوت است، متقارن بودن ربات و همچنین هرچه سبک‌تر بودن آن، پارامترهای مهمی برای ایجاد وضعیتی ثابت و رسیدن به هدف نهایی که عملکردی مطلوب است، بودند که با آزمون و خطاهای فراوان و امتحان کردن ساختارهای مختلف، در نهایت شاسی مکانیکی با عملکردی مطلوب تست و ساخته شد.

اتصالات! : شاید بتوان گفت مهم‌ترین و وقت‌گیرترین چالش در تمام مدت پروژه که همیشه با آن درگیر بودم، در کیفیت بسته شدن اتصالات و همچنین کیفیت ساخت کابل‌های موجود خلاصه می‌شود که پس از خطاهای مختلف در نهایت طراحی تمیز و مناسب فراگرفته و سپس پیاده‌سازی شد که شاید در نگاه کلی به مسئله، معیار گرفته نشود.

روش‌های تست شده:

ابتدا توسط برد آردوینو Uno R3 و برای تست الگوریتم‌ها من از سنسور شتاب‌سنج ۶ محوره MPU6050 که در بالاتر اشاره شد استفاده کردم و با بهره‌گیری از الگوریتم PID، خطای ایجاد شده در هر مرحله را به حداقل رساندم. سپس به سراغ روش تست بوسیله سنسورهای IR رفتم که چالش‌هایی در این مسیر وجود داشت که در بالاتر نیز اشاره شد. در این مرحله پس از تست کردن عملکرد، به سراغ پیاده‌سازی این روش توسط میکروکنترلر و زبان AVR رفتم و در این مسیر، شاسی ربات را از اول ساخته و محصول نهایی‌تری ساختم.

الگوریتم اصلی:

دو موتور در کنار هم در زیر ربات ثابت هستند که وظیفه سیستم حرکتی ربات و ایجاد تحرک و تعادل را در محصول دارند. در کنار این دو موتور و در وسط ربات، دو دسته سنسور فرستنده و گیرنده مادون‌قرمز قرار گرفته‌اند که وظیفه تشخیص وجود خطا و معین کردن زاویه تغییر یافته ربات را دارند که توسط میکروکنترلر و توسط پایه‌های ADC، دیجیتالی‌ز شده‌اند و در میکرو به صورت اعداد ۸ بیتی خوانده شده و سپس با یک آستانه مشخص، خطا را محاسبه کرده و متناسب با آن، موتورها را در خلاف جهت خطای ایجاد شده به حرکت در می‌آورد که باعث رفع شدن خطا می‌شود که پس از مشخص شدن جهت حرکت موتورها و با توجه به زاویه خطا، از الگوریتم PID استفاده شده تا بتوان سرعت مناسب و پیوسته‌ای برای موتورها مشخص نمود. لازم به ذکر است که در این مرحله، برای تشخیص آستانه مناسب و همچنین حذف نویزهای محیط، مدتی در ابتدای شروع به کار ربات، مقداری آنالوگ از سنسورها خوانده شده و با نمونه‌برداری در زمانی مشخص، میانگینی برای معین کردن آستانه مناسب، استفاده شده است. سپس با توجه به زاویه خطای ایجاد شده و اینکه در چه سطحی از خطا هستیم، موج PWM ای با Duty Cycle‌هایی مشخص (تست شده با آزمون و خطا) و با توجه به جهت مورد نیاز چرخش ربات ساخته شده و در جهت رفع خطای احتمالی، ربات شروع به حرکت می‌کند.

در اینجا برای درگیر نبودن همیشگی میکرو، وقفه‌ای نوشته شده است که زمانی که زاویه از آستانه‌ی مشخصی بیشتر شد و نیاز به عملیات ذکر شده بود، وقفه فعال شده و کنترل‌کننده (Handler) مربوطه اجرا می‌شود.