

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

پروژه ربات جارو برقی هوشمند

درس : کنترل خطی

استاد : هادی عزمی

ارائه دهندگان : آرین حسن پور ، سعید غلامی

تیرماه ۱۴۰۳

فهرست مطالب

۱	خلاصه ای کلی از اهداف پروژه
۲	مقدمه
۳	بررسی راهکارهای موجود جهت انجام پروژه
۳	سنسور ها
۴	الگوریتم های مسیریابی
۷	پردازنده ها و میکروکنترلر ها
۸	سیستم های شارژ و مدیریت انرژی
۹	ساختار فیزیکی پروژه
۹	سنسور ها
۱۰	سنسور Bumper
۱۱	LCD 16*2
۱۲	موتور گیربکس N20
۱۳	چرخ هرز گرد
۱۳	درایور LN298N
۱۴	Arduino mega 2560
۱۵	چالش های پروژه و راهکارهای استفاده شده
۱۵	دقت پایین سنسور GY271
۱۶	گیج شدن ربات در هنگام افتادن در مسیر ۰ یا ۳۶۰ درجه
۱۶	چالش تعیین میزان چرخش در سطوح مختلف
۱۷	الگوریتم و ساختار نرم افزاری

مزایا و معایب ۱۷

فلوچارت ۱۸

راهنمای استفاده از ربات برای کاربران ۲۰

خلاصه ای کلی از اهداف پروژه

پروژه تعریف شده جهت شبیه سازی یک جارو برقی هو شمند میبا شد که بتواند از نقطه ای کار خود را آغاز نموده و ناحیه مورد نظر که قرار است عمل جارو کشیدن روی آن انجام شود را پوشش دهد و همچنین قابلیت تشخیص مانع داشته باشد و بتواند طبق شرایط پیش آمده تصمیم بگیرد که به کدام سمت حرکت کند .

مقدمه

در دنیای امروزی ، با افزایش مشغله های روزمره و نیاز به بهره‌وری بیشتر ، استفاده از ابزارهای هوشمند برای انجام کار های خانگی به طور چشمگیری افزایش یافته است.

یکی از این ابزارهای هوشمند ، جاروبرقی های رباتیک هستند که به منظور سهولت در نظافت منازل طراحی شده است.

این دستگاه ها با بهره‌گیری از فناوری های پیشرفته مانند هوش مصنوعی ، سنسورهای متنوع و الگوریتم های تصمیم گیری ، قادر به شناسایی محیط ، مسیریابی بهینه و اجرای تمیزکاری به صورت خودکار هستند.

هدف از پروژه جاروبرقی هوشمند، توسعه و پیاده سازی یک بات خودکار است که بتواند با حداقل دخالت انسانی نظافت منازل را به صورت کارآمد و موثر انجام دهد. این پروژه به منظور افزایش کیفیت زندگی ، کاهش زمان صرف شده برای نظافت و بهبود بهره‌وری از زمان ، طراحی و اجرا شده است.

در این گزارش به بررسی جزئیات فنی و علمی پروژه ، مراحل طراحی و پیاده سازی ، نتایج حاصل از آزمایش ها و ارزیابی عملکرد جاروبرقی هوشمند پرداخته شده است . همچنین ، چالش ها و مشکلات مواجه شده در طول پروژه و راه حل های ارائه شده نیز مورد بحث قرار خواهند گرفت . این گزارش با هدف ارائه یک دیدگاه جامع و کامل از پروژه و نتایج آن تهیه شده است.

بررسی راهکارهای موجود جهت انجام پروژه

برای توسعه یک جاروبرقی هوشمند ، بررسی و تحلیل راهکارهای موجود بسیار حائز اهمیت است. در این بخش به بررسی تکنولوژی‌ها ، الگوریتم‌ها و روش های مختلفی که میتوانند در طراحی و پیاده سازی جاروبرقی هوشمند مورد استفاده قرار گیرند، پرداخته میشود.

سنسورها

سنسورها نقش مهمی در تشخیص محیط و موانع و تصمیم گیری برای حرکت ربات را ایفا میکنند. چند نوع سنسور با توجه به تحقیقات انجام شده که بیشتر استفاده میشود بدین صورت میباشد.

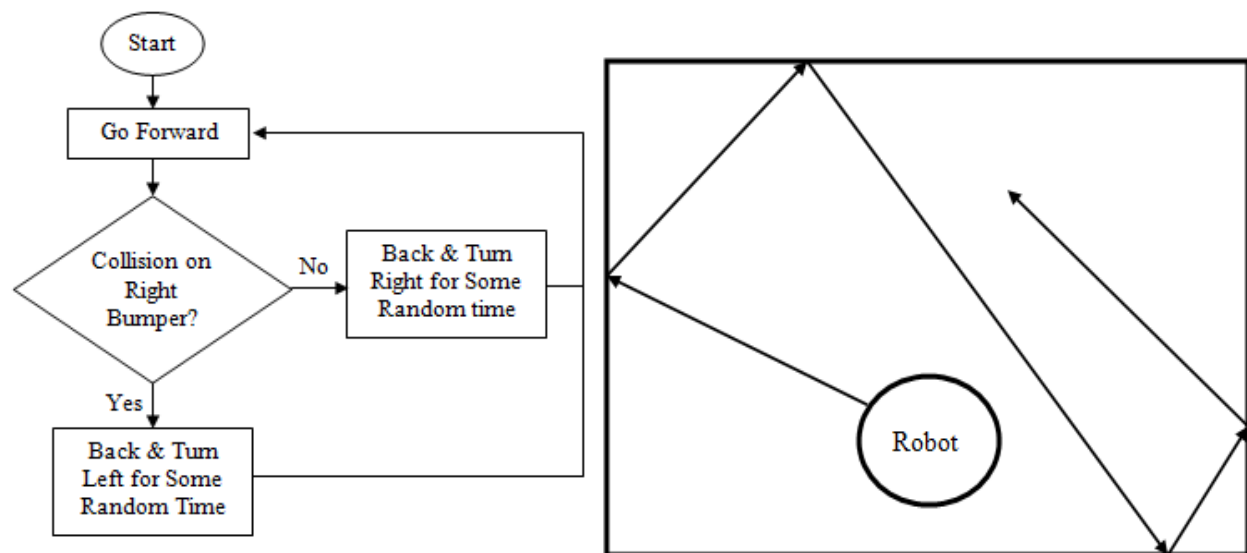
سنسور های فاصله سنج :

- سنسور زمان پرواز (TOF) پر استفاده ترین سنسور استفاده شده در نمونه های بازاری جارو های رباتیک هوشمند میباشد کار آن بدین صورت است که عمق و فاصله را با استفاده از مفهوم زمان پرواز تشخیص می دهد . با استفاده از زمان لازم برای حرکت فوتون ها از فرستنده سنسور به جسم و بازگشت به گیرنده، به محاسبه فاصله کمک میکند.
- سنسور اولتراسونیک ، اولتراسونیک به معنای مافوق صوت است. در واقع امواج اولتراسونیک امواج مکانیکی هستند که فرکانس آنها از شنوایی انسان بیشتر است. اصطلاح اولتراسوند برای امواج الاستیک با فرکانس بالای ۲۰ کیلوهرتز استفاده می شود. سنسور اولتراسونیک اغلب دارای فرستنده و گیرنده امواج اولتراسونیک است که پس از برخورد با مانع منعکس شده و به سنسور باز می گردد. سپس با توجه به زمان برگشت و کیفیت امواج منعکس شده عواملی مانند فاصله تا مانع، نوع مانع و سرعت مانع بدست می آید.
- سنسور IR (مادون قرمز) که بر اساس ارسال و دریافت امواج مادون قرمز کار میکند .

- Bumper سنسوری فیزیکی است که تنها پس از برخورد با مانع سوئیچی را فعال میکند. این بخش در فصل‌های آینده مورد بحث قرار خواهد گرفت.

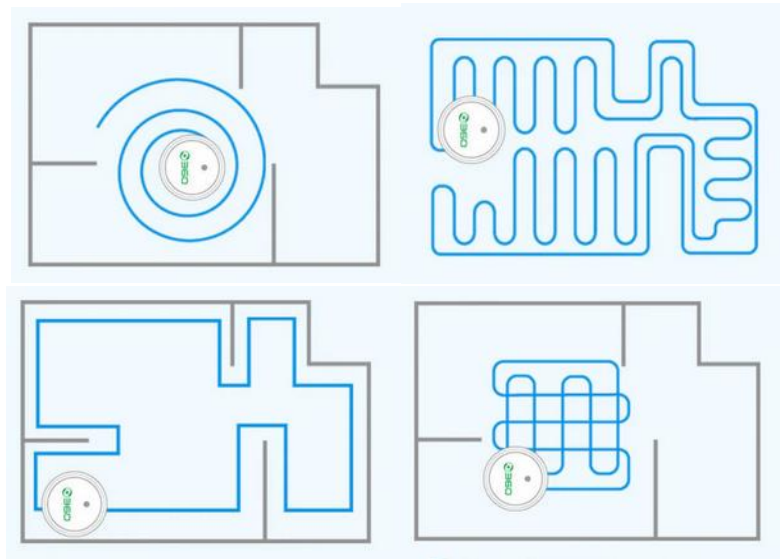
الگوریتم‌های مسیریابی

- الگوریتم تصادفی (Random Walk) ساده‌ترین نوع الگوریتم که ربات به صورت تصادفی حرکت میکند و با برخورد به مانع جهت حرکت خود را تغییر میدهد و نیاز به تصمیم‌گیری‌های پیچیده ندارد.

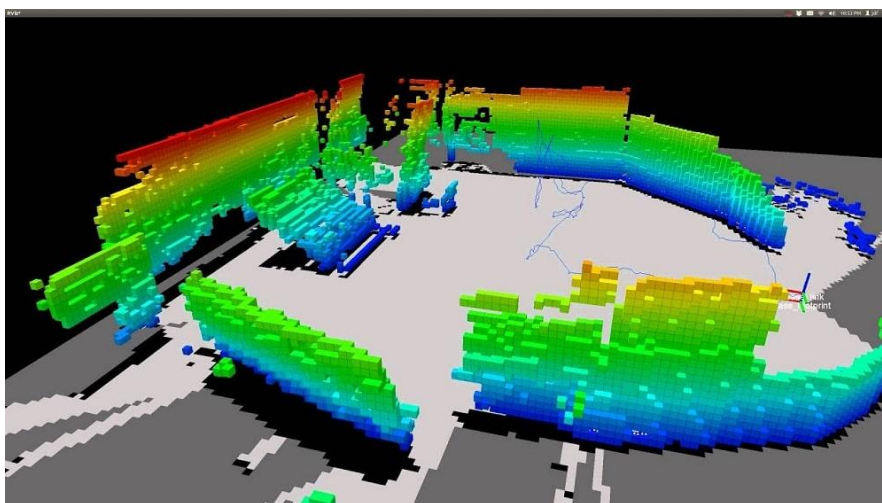


- الگوریتم سیستمی (Systematic Cleaning)

در این الگوریتم شیوه کار بر اساس حرکت های زیگزاگی و یا مارپیچی برای پوشش کامل محیط میباشد. این الگوریتم به طور مفصل در بخش های آینده مورد بحث قرار میگیرد.



- الگوریتم مبتنی بر نقشه سازی و مسیر یابی به صورت همزمان (SLAM)
این الگوریتم با استفاده از سنسورها نقشه‌ای از محیط را میسازد و مسیر بهینه را تعیین میکند.



- الگوریتم های یادگیری ماشین :
استفاده از شبکه‌های عصبی و یادگیری تقویت برای بهبود عملکرد جارو برقی با گذر زمان.

پردازنده ها و میکروکنترلر ها

- Arduino

میکروکنترلر ساده و کارآمد برای ایجاد پروژه های کوچک و متوسط مناسب ترین گزینه هستند.

- Raspberry pi

پردازنده قوی تر جهت اجرای الگوریتم های پیچیده تر به همراه پردازش تصویر مناسب ترین گزینه هستند.

سیستم های شارژ و مدیریت انرژی

یکی از چالش های مهم درخصوص طراحی ربات های هوشمند سیستم شارژ و مدیریت انرژی و سیستم شارژ خودکار است.

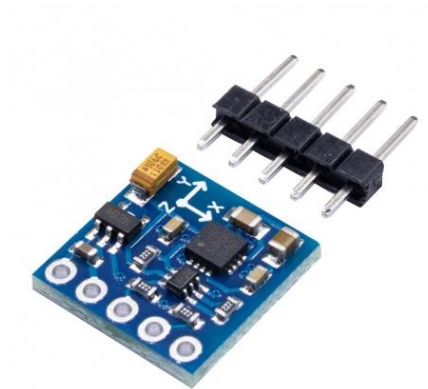
راهکار های موجود

- باتری های لیتیوم یون
- سیستم های شارژ خودکار که شامل ایستگاه های شارژ جارو برقی هستند که ربات به صورت خودکار به آنها باز میگردد.

ساختار فیزیکی پروژه

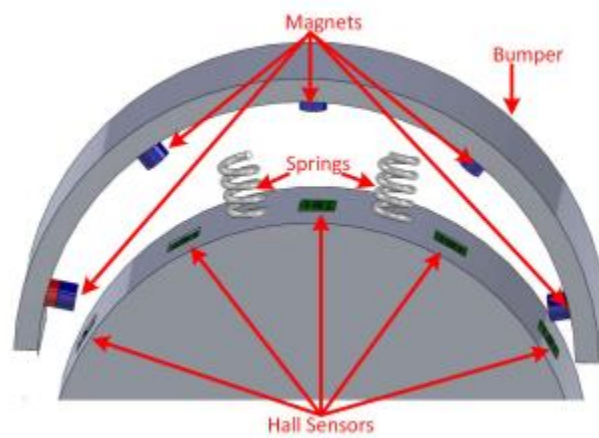
سنسورها

- سنسور قطب نمای GY271 جهت تشخیص جهت



سنسور Bumper

البته این سنسور در دسترس نبود و از چند میکرو سوئیچ جهت شبیه سازی این سنسور استفاده کرده ایم. طرز کار آن بدین شکل است که با برخورد به یک مانع میکروسوئیچ فعال شده و پالسی را به میکرو میفرستد. دلیل استفاده از این سنسور ساختار مکانیکی ساده آن و قابل اعتماد بودن آن نسبت به دیگر سنسورهای تشخیص مانع می باشد و همینطور از دیگر مزایای آن میتوان به هزینه بسیار کم آن اشاره نمود. از مشکلات آن هم میتوان به حساسیت به آسیب در برخوردهای مکرر اشاره کرد که برای حل این مشکل نیازمند استفاده از ساختار فیزیکی مستحکم تری میباشیم.



LCD 16*2

جهت نمایش گرافیکی و ارتباط ربات با کاربر و نشان دادن وضعیتی که در آن قرار دارد.



موتور گیربکس N20

اولین دلیل استفاده از این نوع موتور گیربکس اندازه کوچک و جمع و جور بودن آن می باشد . از دیگر مزایای این موتور میتوان به کارایی و توان بالا جهت تحمل وزن جارو برقی و به حرکت درآوردن آن در سرعت های پایین می باشد . همینطور دارای ساختار مقاوم قابل اطمینان تر هستند و عمر طولانی تری نسبت به دیگر موتور گیربکس های بازاری دارند.



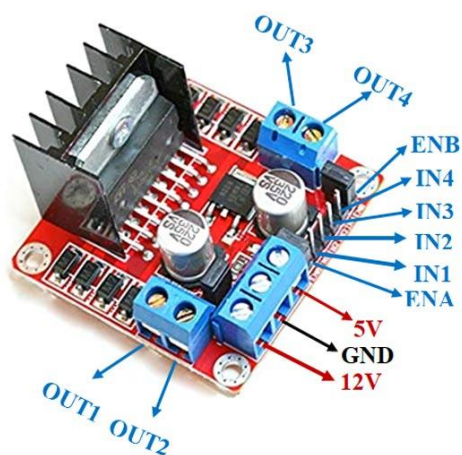
چرخ هرز گرد

جهت حفظ تعادل و حرکت ربات در تمامی جهت ها



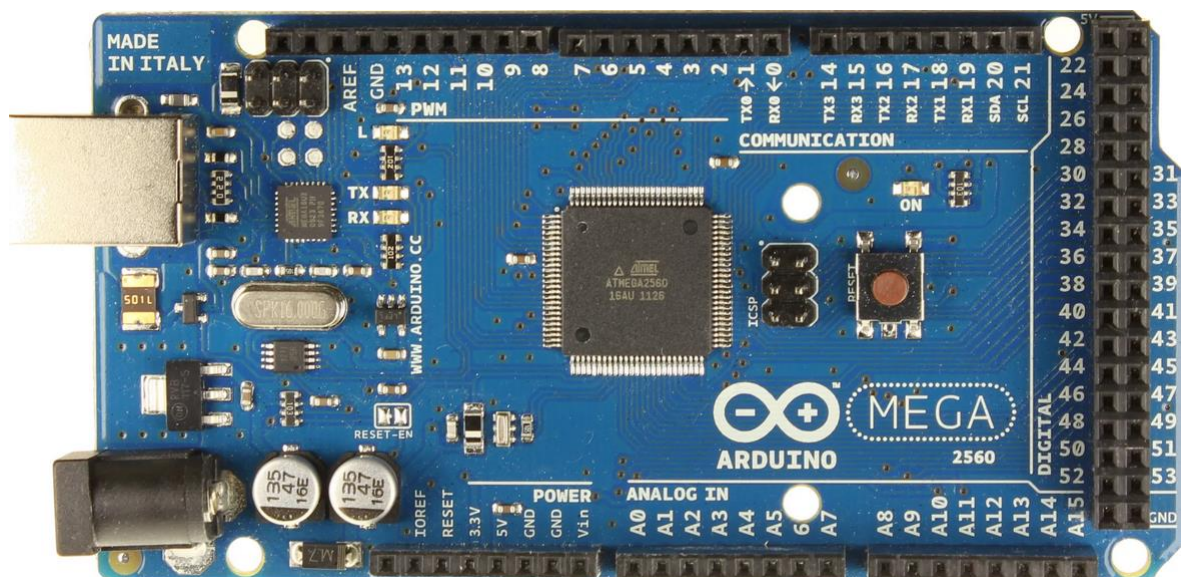
درایور LN298N

جهت درایو کردن موتور گیربکس های استفاده شده.



Arduino mega 2560

به دلیل استفاده از چهار میکروسوئیچ و یک عدد LCD و دو موتور گیربکس و سنسور GY271 تعداد پایه‌های Arduino uno به اندازه مورد نیاز نبود و برای حل این مشکل از این مدل Arduino استفاده کردیم .



چالش‌های پروژه و راهکارهای استفاده شده

دقت پایین سنسور GY271

۱. تداخل الکترومغناطیسی

سنسورهای مغناطیس سنج به شدت به تداخلات الکترومغناطیسی حساس هستند. وجود تجهیزات الکترونیکی نزدیک به سنسور میتوانند باعث ایجاد نویز و اختلال در عملکرد سنسور می‌شود. راه‌حل آن شیلدینگ مناسب و دور نگه داشتن مدار از سنسور می‌باشد.

۲. خطای کالیبراسیون

عدم کالیبراسیون صحیح سنسور میتواند منجر به خطاهای اندازه‌گیری دقیق درجه شود. راه‌حل اجرا شده در پروژه کالیبراسیون دستی سیستم توسط کاربر می‌باشد، بدین صورت که در ابتدای شروع کار ربات توسط کاربر با استفاده از سوئیچ تعبیه شده در سمت راست به صورت ساعت گرد چهار جهت اصلی به ربات شناسانده میشود و میزان درجه‌ها ثبت می‌شود.

۳. نویز محیطی و وجود اجسام مغناطیسی در محیط

حضور اجسام مغناطیسی مانند آهن‌رباها و فلزات در نزدیکی سنسور میتواند میدان مغناطیسی آنرا تغییر دهد و دقت سنسور را کاهش دهد. راه‌حل برای این مشکل پیدا نکردیم ولی تاثیر خیلی زیادی هم در عملکرد ربات نداشت و فقط سعی شد با استفاده از الگوریتم برنامه این خطا پوشش داده شود.

گیج شدن ربات در هنگام افتادن در مسیر ۰ یا ۳۶۰ درجه

زمانی که سنسور به این درجات می‌رسد ربات نمی‌تواند جهت صحیح حرکت را تعیین کند و ربات مسیر بی‌هدفی را طی می‌کند که برای حل این مشکل دو الگوریتم بجای یک الگوریتم برای ربات نوشته شد که طرز کار آن به طور مفصل در بخش توضیحات الگوریتم ارائه خواهد شد.

چالش تعیین میزان چرخش در سطوح مختلف

چرخش ربات مستقل از سنسور قطب نما تعریف شده است تا خطای کمتری داشته باشد و سنسور فقط در هنگام جلو رفتن ربات و در مسیر صاف حرکت کردن به کمک ربات می‌آید. برای ایجاد چرخش‌های نود درجه ای به چپ یا راست امکان استفاده از شیوه آزمون و خطا برای بدست آوردن تایم چرخش تا رسیدن به نود درجه مناسب نمی‌باشد، چرا که این تایم متناسب با میزان اصطکاک سطح می‌باشد.

از این رو برای رفع این چالش ربات بعد از کالیبراسیون یک دور به طور کامل به دور خود می‌چرخد تا به درجه ای که ابتدا در آن شروع به حرکت نموده برسد و تایم این مدت زمان چرخش را در خود ذخیره کرده و به چهار تقسیم میکند تا بتواند متناسب با سطحی که بر روی آن قرار گرفته تایم مورد نیاز برای چرخش ربع دایره را با دقت بسیار بالا و قابل قبول بدست بیاورد و در چرخش‌های خود از آن استفاده کند.

الگوریتم و ساختار نرم‌افزاری

الگوریتم استفاده شده در پروژه الگوریتم سیستمی می‌باشد. در ادامه به مزایا و معایب آن اشاره می‌شود.

مزایا و معایب

• مزایا

۱. پوشش کامل و کارآمد فضا
این الگوریتم تضمین میکند که تمام فضای اتاق به صورت یکنواخت و بدون نواحی تمیز نشده پوشش داده شود. این الگوریتم از الگوهای حرکتی منظم حرکت زیگزاگی استفاده میکند.
۲. ساده و قابل اجرا
این الگوریتم ساده و آسان برای پیاده سازی میباشد و نیاز به محاسبات پیچیده ندارد. این سادگی باعث میشود که با سخت افزارهای ساده مانند Arduino قابل پیاده سازی باشد.
۳. افزایش کارایی باتری
با داشتن یک الگوی حرکتی منظم و سیستماتیک، ربات مسیرهای بی‌هدف کمتری را طی میکند و از باتری به صورت بهینه‌تری استفاده میکند.
۴. پیش‌بینی پذیری
حرکت‌های سیستماتیک قابل پیش‌بینی هستند و میتوانند برنامه ریزی‌های زمانی بهتری برای تمیزکاری انجام دهند.

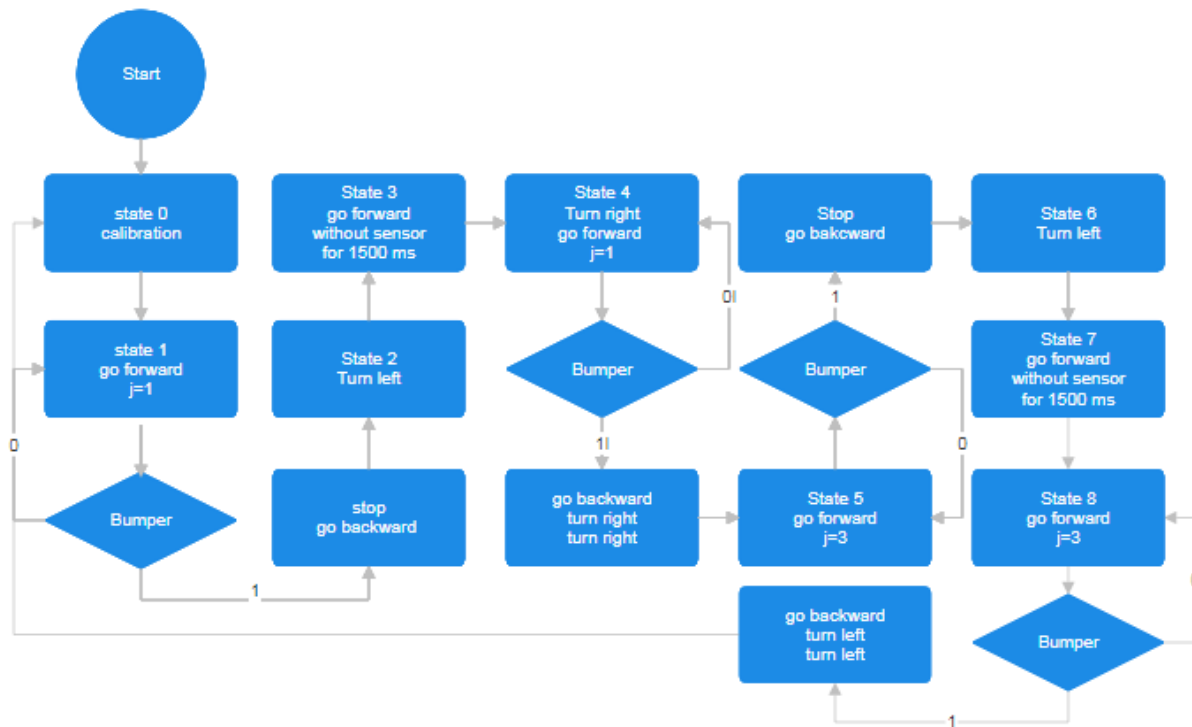
• معایب

۱. عدم انعطاف پذیری
این الگوریتم انعطاف پذیری کمتری در مواجهه با محیط‌های پیچیده و موانع غیر منتظره دارد. اگر موانع زیادی در مسیر باشد الگوریتم ممکن است نتواند به درستی کار کند.
۲. وابستگی به محیط ساده

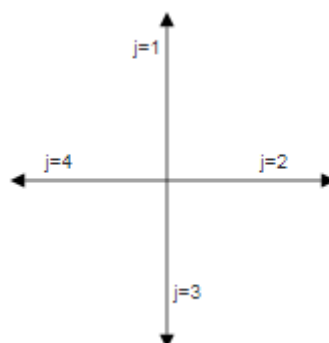
در محیط های با موانع پیچیده و غیرمنظم ، الگوریتم سیستمی ممکن است کارایی کمتری داشته باشد و نتواند محیط را به خوبی پوشش دهد.

فلوچارت

فلوچارت الگوریتم در شکل زیر به طور کاملا واضح رسم شده است.



جهت های کالیبره شده :



الگوریتم دوم نیز دقیقاً مانند الگوریتم اول میباشد با این تفاوت که تغییر جهت ها برعکس میباشد.

پس از کالیبره شدن ربات توسط کاربر ابتدا ربات چک میکند که در کدام یک از چهار جهت اصلی نقطه صفر یا سیصد و شصت درجه وجود دارد ، اگر این نقطه در جهت یک یا سه باشد ارایه ذخیره شده یکبار به سمت راست شیفت خورده و درجه های ذخیره شده برای الگوریتم دوم تنظیم میشود و ربات هم با یکبار چرخش به سمت راست با الگوریتم شماره دو شروع به کار میکند.

در غیر این صورت ربات با الگوریتم شماره یک کار خواهد کرد.

این بررسی شرط منجر به این میشود که نقطه صفر یا سیصد و شصت درجه در اندیسی از ارایه قرار بگیرد که ربات در آن مسیر ها بدون سنسور در حد بیست الی سی سانتی متر طی میکند و سپس چرخش کرده و تا زمان رسیدن به یک مانع حرکت خود را به صورت مستقیم ادامه میدهد.

راهنمای استفاده از ربات برای کاربران

۱. دستگاه را روشن کنید .
۲. ربات را در گوشه خانه قرار به طوری که پشت و سمت چپ آن دیوار باشد.
۳. جهت اولیه حرکت ربات را با دوبار فشردن کلید کالیبراسون تعیین شده در سمت راست ربات را فشار دهید.
۴. به سمت ساعت گرد جهت راست ، پایین و چپ را با هر بار فشردن کلید کالیبراسیون به ربات نشان دهید.
۵. در همان جهت که کلید را برای آخرین بار فشردید ربات را به زمین بگذارید .
۶. اکنون میتوانید به کارهای دیگر خود پردازید و از تماشای هوشمندی آن لذت ببرید.

پایان