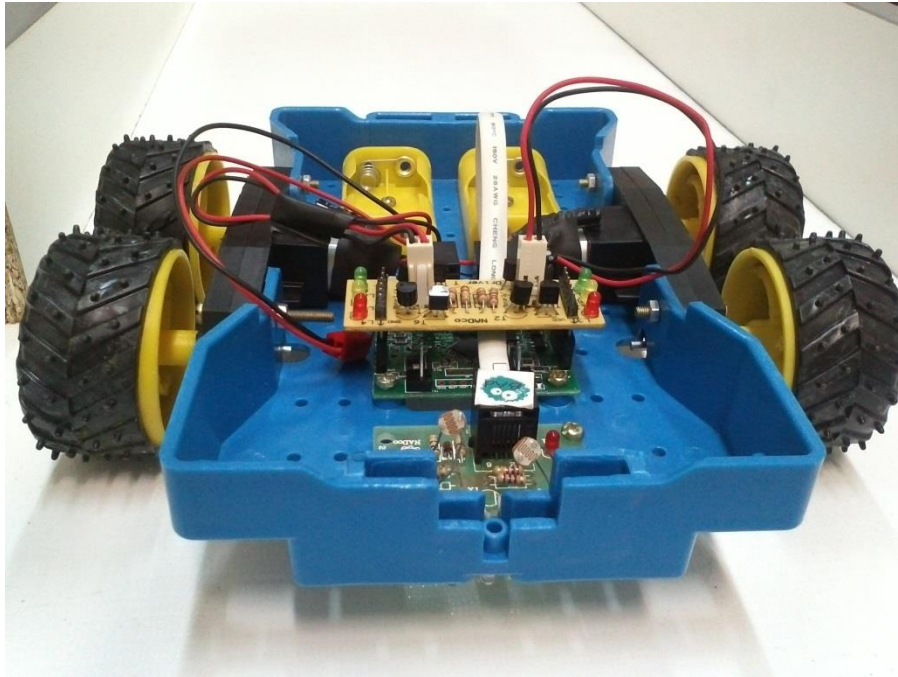


بسمه تعالی

مجموعه روبرو ۱

تابستان ۱۳۹۶



مقدمه:

روبورو یک کیت آموزشی است که برای دانش آموزان سالهای آخر ابتدایی و مقطع متوسطه دوره ی اول و دوم، طراحی شده است. این کیت از اجزای جدا از هم تشکیل شده و دانش آموز با سر هم کردن اجزای مختلف آن را کامل می کند، بعد از ساخت سخت افزار، دانش آموز برنامه روبورو را روی سیستم خود نصب می کند و برنامه های مختلف را طراحی می کند و بوسیله کابل یا ارتباط بلوتوثی انتقال آن را به حافظه ربات منتقل می نماید. بعد از اینکه برنامه به روی حافظه ذخیره شد، ربات شروع به تصمیم گیری می کند. کیت روبورو به سه کیت (روبورو ۱ ، روبورو ۲ ، روبورو ۳) تقسیم بندی می شود، که اساس هر سه کیت یکسان است اما با توجه به سن دانش آموز و دوره هایی که گذرانده نوع ماژول ها و برنامه نویسی متفاوت می شود.

در واقع روبورو مجموعه سخت افزاری/ نرم افزاری است که دانش آموز می تواند هم به زبان فارسی و هم به زبان انگلیسی آن را برنامه ریزی کند.

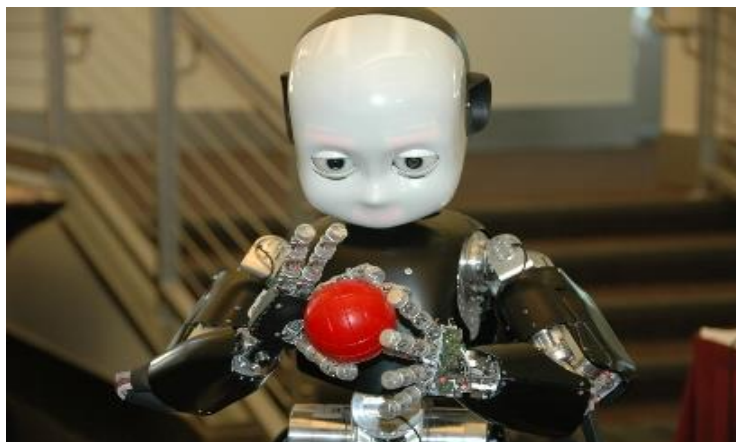
در جدیدترین نمونه روبورو، امکان ارتباط بلوتوثی برای بارگزاری برنامه فراهم شده است، و دانش آموزان می تواند به راحتی و بدون نیاز به مبدل، ربات را به لپ تاپ متصل نماید.

فهرست:

- فصل اول: آشنایی با علم رباتیک و انواع ربات ها..... ۴
- فصل دوم: شیوه نصب برنامه روبرو و معرفی سخت افزار آن..... ۱۲
- فصل سوم: آموزش برنامه نویسی و شیوه انتقال برنامه از کامپیوتر به ربات..... ۲۴
- فصل چهارم: سنسور تماسی و برنامه حل ماز..... ۵۹
- فصل پنجم: سنسور مادون قرمز و برنامه تعقیب خط..... ۶۵
- فصل ششم: سنسور فوتوسل و برنامه تعقیب و گریز..... ۷۱

فصل اول

آشنایی با علم روباتیک و انواع ربات‌ها



۱-۱ روبات چیست؟

کلمه روبات به معنی کارگر است. سابقه ساخت روبات به 270 سال قبل از میلاد مسیح برمی گردد. زمانی که یونانیان به ساخت مجسمه های متحرک می پرداختند. صاحب نظران روباتیک و فعالان در دانشگاه بر اساس تولیدات و با توجه به توسعه ای که روبات ها در تکنولوژی، صنعت و زندگی انسان دارند، چند تعریف عینی و قابل فهم از روبات ارائه کردند:

۱- یک دستگاه یا وسیله ای خودکار که قادر به انجام اعمالی است که معمولا به انسانها نسبت داده می شود و یا مجهز به قابلیت است که شبیه هوش بشری باشد.

۲- یک روبات هوشمند، ماشین خودکار چند منظوره ای است که طیف وسیعی از وظایف متفاوت را تحت شرایطی که حتی ممکن است نسبت به آن شناخت کافی نداشته باشد، همانند انسان انجام دهد.

موسسه روباتیک (صنعتی) آمریکا که یکی از معتبرترین و با سابقه ترین کمپانیهای صنعت روباتیک و هوش مصنوعی است، روبات را اینگونه تعریف می کند:

"روبات یک جابجا کننده چند وظیفه ای است که برای حرکت دادن مواد، قطعات، ابزارها و یا وسایلی خاص، با استفاده از حرکات برنامه ریزی شده، قابل تغییر برای تحقق فرامین طراحی شده است."

تصور مردم از روبات ماشینی است که اعمال هوشمندانه همانند انسان یا اعمالی مافوق تصور انسان را انجام می دهد. جالب تر اینکه آنها فکر می کنند روبات ها حتما باید شبیه انسان باشند. اگر روبات را بخواهیم عمومی و کلی معرفی کنیم، روبات یک ماشین

الکترومکانیکی هوشمند با خصوصیات زیر است:

۱. چندکار را می تواند انجام دهد.

۲. می توان آن را به صورت مکرر برنامه ریزی کرد.

۳. کارآمد و مناسب با محیط است و توانایی هماهنگ کردن خود با محیط را دارد.

در حال حاضر هدف نهایی علم رباتیک خلق رباتهایی است که همانند انسان خصوصیات برجسته ای در رفتار، حرکت، هوش و ارتباط داشته باشند.

دسته بندی ربات ها

روبات ها در سطوح مختلف دارای دو خاصیت " تنوع در عملکرد و قابلیت تطبیق خودکار با محیط " می باشند. بر اساس این دو خاصیت، دسته بندی ربات ها انجام می گیرد. اتحادیه ربات های ژاپنی ربات ها را اینگونه دسته بندی می کند:

روبات برای کارهای متوالی بدون تغییر استفاده شود.

روبات برای کارهای متوالی متغیر استفاده شود.

روبات تقلید کننده

روبات باهوش

روبات (وسیله ایی) که توسط دست کنترل می شود.

۱-۳- آناتومی روبات

اجزا و قسمت های اصلی یک روبات شامل شش قسمت است:

۱. **سیستم مکانیکی مفصل بندی شده:** سیستم مکانیکی مفصل بندی شده شامل بدنه و قسمت های

مختلف مکانیکی روبات است که روبات توسط آنها بر محیط اطراف خود تاثیر فیزیکی می گذارد.

۲. **عملگرها:** عملگرها تولید کنندگان نیروی محرکه روبات هستند.

۳. **سیستم انتقال قدرت(نیرو):** سیستم انتقال نیرو، وظیفه انتقال نیروی محرکه توسط عملگرها به قسمت های مکانیکی روبات را بر عهده دارند.

۴. **سنسورها:** سنسورها وظیفه ی سنجش و اندازه گیری کمیت های فیزیکی و تبدیل آنها به کمیت های الکتریکی قابل اندازه گیری را بر عهده دارند.

۵. **کنترلر:** کنترلر مغز متفکر و فرمان دهنده روبات است که بر اساس برنامه ی داده شده به آن و در نظر گرفتن کمیت های ورودی و خروجی تصمیم گیری کرده و نتیجه را به قسمت های مختلف روبات از جمله عملگرها اعمال می کند.

۶. منبع تغذیه: منبع تغذیه وظیفه تامین انرژی قسمت های مختلف روبات را بر عهده دارد.

۴-۱- انواع روبات ها

روبات های امروزی شامل قطعات الکترونیکی و مکانیکی زیادی هستند. برخی روبات ها در ابتدا به صورت بازوهای مکانیکی برای جابجایی قطعات و یا کارهای ساده و تکراری که موجب خستگی و عدم تمرکز کارگر و افت بازده می شد بوجود آمدند اینگونه روباتها به روبات های جابجاگر معروف هستند. جابجاگرها معمولا در نقطه ای ثابت و فضایی محدود نصب می شوند و کار می کنند و جز کاری که برای آن طراحی شده اند قادر به انجام کار یا دستورالعمل دیگری نیستند.

۵-۱- معرفی چند نمونه روبات

۱-۵-۱- ربات فضانورد

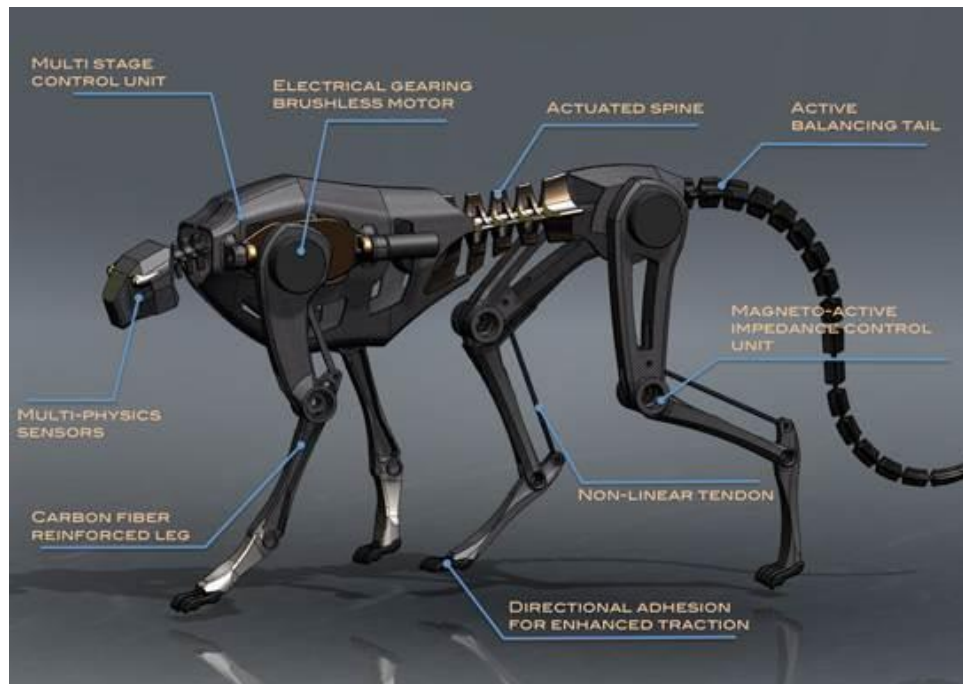


جالب است بدانید که از زمانی که ایده ارسال ربات به فضا مورد مطالعه قرار گرفته است این عرصه دائما در حال گسترش و پیشرفت بوده است. نمود بارز آن ربات پیشرفته ROBONAUT2 است که این ربات توسط شرکت

جنرال موتورز و سازمان فضایی آمریکا ناسا ساخته شد و در سال ۲۰۱۲ به ایستگاه بین المللی فضایی ارسال شده است.

هدف این ربات انسان نما کمک به فضانوردان ایستگاه بین المللی در رفع مشکلات فنی موجود است. این ربات قادر است به راحتی با ایستگاه های زمینی ارتباط برقرار کرده و نوع مشکل را اطلاع رسانی نماید.

۱-۵-۲- چیتا ربات



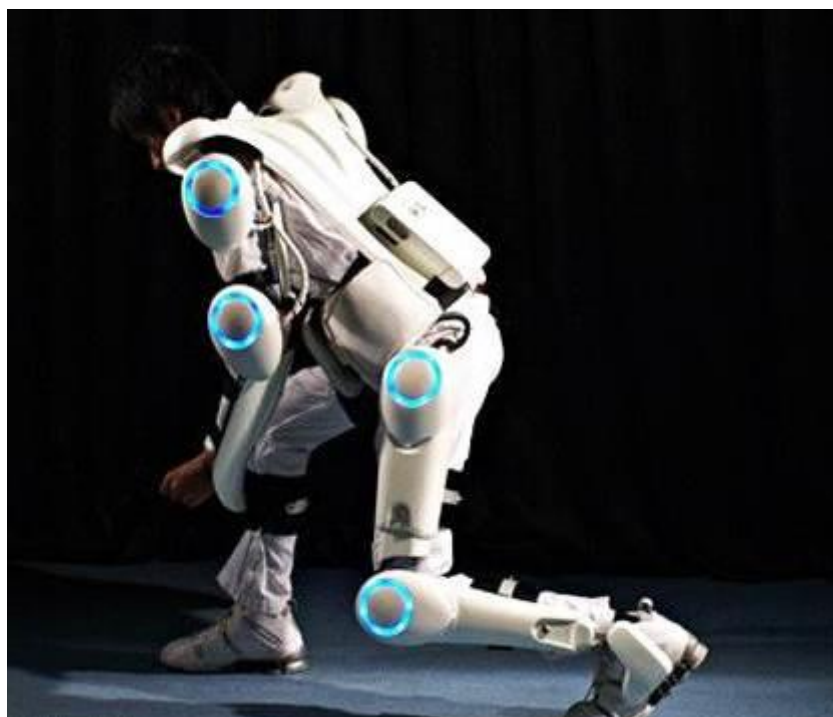
شرکت بوستون دینامیک موفق شده است که رباتی را بسازد که نام یوزپلنگ را به خود اختصاص داده است. این ربات سریع ترین ربات ساخته شده در جهان است. اگرچه شاید این ربات به سرعت واقعی یوزپلنگ یعنی ۱۰۰ کیلومتر بر ساعت نرسد اما به لطف پمپ های هیدرولیکی و پایداری بسیار خوب اجزای مکانیکی این ربات قادر است که تا سرعت ۴۵ کیلومتر بر ساعت نیز حرکت نماید. سازندگان در تلاشند که سرعت آن را حتی از خود حیوان واقعی نیز بیشتر نمایند!

۱-۵-۳- ربات ۴ پا:



این ربات نیز یکی دیگر از شاهکارهای شرکت بوستون دینامیک است. این ربات ۴ پا اسب مانند به منظور حرکت بر روی نقاط کوهستانی طراحی شده است و همانند یک ماشین باربر می تواند به کمک کوه نوردان و اعضای کمپ های کوه نوردی یا موارد نظامی بیاید. این ربات قادر خواهد بود تا وزن ۱۸۲ کیلوگرم را حمل نماید. سرعت این ربات تا ۳۲ کیلومتر بر ساعت افزایش می باید.

۱-۵-۴- ربات همکار:



هرچند بسیاری از سوانح طبیعی قابل پیش بینی و یا جلوگیری از خسارت های بوجود آمده تا حد کامل نیست، اما ذهن تیز دانشمندان در دانشگاه تسوکوبا ژاپن رباتی را ساخته اند که بتوانند از آن در سوانح طبیعی مانند سونامی و سیل و زلزله استفاده نمایند. ایده این ربات برای کمک به انسان برای حضور در مناطق خطرناکی است که امکان حضور انسان نیست. به عنوان مثال در انفجارهای اتمی یا تشعشعات بسیار مواد اتمی امکان حضور انسان نیست و این ربات می تواند تمامی کارهای تمیزکردن و بررسی و پوشش مناطق تابش هسته ای را انجام بدهد.

۱-۵-۵- ربات صنعتی



این ربات که نامش (باکستر) همانند حیوان خانگی است، یکی از پیشرفته ترین ربات های موجود در صنعت است. این ربات عموماً در صنایع باتری سازی و بستن بطری ها استفاده می شود و بهبود هزینه ای که صورت می گیرد به لطف این ربات بسیار به سود صاحبان صنعت است. این ربات به غیر از بستن در و کنترل بطری ها می تواند در بسیاری از صنایع نظیر خودروسازی با توجه به پیشرفته تر بودنش نسبت به سایر ربات ها انجام وظیفه بکند!

۱-۵-۶- ربات پرستار



معمولاً مراکز درمانی و بیمارستان ها مکان هایی دشوار هم برای بیماران و هم برای پرستاران است و در این میان ممکن است در وقت های بسیاری به دلیل برخی محدودیت ها امکان رسیدگی به بیماران به صورت مناسب برقرار نباشد. اما به لطف اختراع این ربات توسط شرکت تویوتا این مشکل برطرف شده است. این ربات قادر خواهد بود که نیازهای یک بیمار را به عنوان یک پرستار مرتفع نماید.

فصل دوم

شیوه نصب برنامه روبرو و معرفی سخت افزار آن



۱-۲- معرفی نرم افزار Roboro

نرم افزار روبورو یک نرم افزار بسیار جذاب و کاربردی برای برنامه نویسی رباتهای مختلف می باشد. ما توسط این برنامه می توانیم رباتهای تعقیب خط، شکارچی نور، و ... را طراحی کنیم. تمام دستورات لازم برای برنامه نویسی در منویی فرا روی شما قرار می گیرد.

۲-۲- آموزش نصب نرم افزار روبورو

مرحله اول: ابتدا لوح فشرده ی روبورو را داخل سیستم خود قرار دهید، در این مرحله صفحه ای مطابق صفحه زیر باز می شود.



همانطور که ملاحظه می کنید، چندین گزینه در این صفحه وجود دارد که هر کدام اطلاعاتی را در اختیار شما قرار می دهند.

روی گزینه محصولات کلیک نمایید تا وارد صفحه زیر شوید:

مکاترو	مکاترو	مکاترو	مکاترو
۱ مکاترو	۱ مکاترو	۱ روبرو	۰ مکاترو
۲ مکاترو	۲ مکاترو	۲ روبرو	۱ مکاترو
۳ مکاترو	۳ مکاترو	۳ روبرو	۲ مکاترو
نرم افزار	نرم افزارها	نرم افزارها	۳ مکاترو
			۴ مکاترو
			۵ مکاترو
			۶ مکاترو
			۷ مکاترو
			۸ مکاترو

Back

برای ادامه کار در قسمت روبو روی گزینه ی نرم افزارها کلیک کنید.

با کلیک بروی منوی فوق، صفحه زیر باز می شود:

نرم افزارها

روبو فارسی برای ویندوز ۷

روبو انگلیسی برای ویندوز ۷

DotNet

روبو بلوتوثی

راهنمای راه اندازی روبو بلوتوثی

Back

در این صفحه ۵ گزینه مشاهده می کنید:

۱- نرم افزار روبروی فارسی برای ویندوز ۷

۲- نرم افزار روبروی انگلیسی برای ویندوز ۷

۳- فایل کمکی DotNet

۴- نرم افزار روبروی بلوتوثی

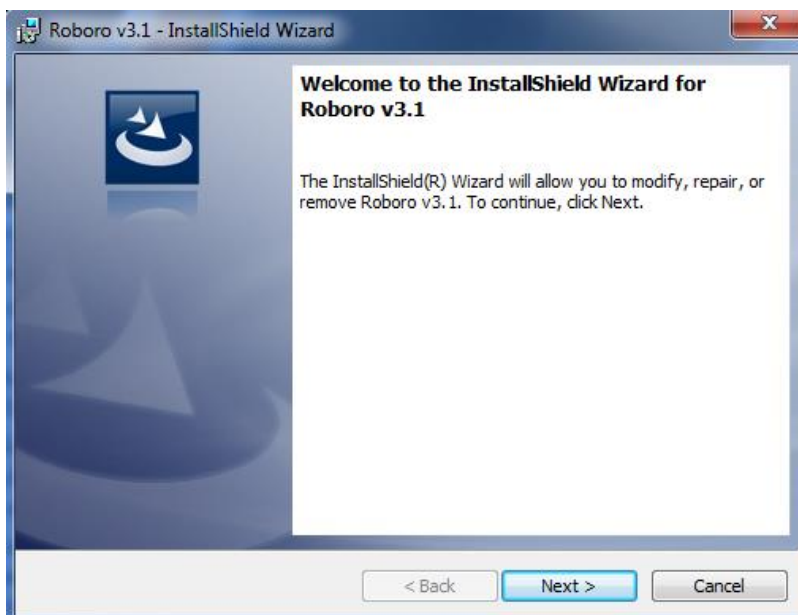
۵- راهنمای راه اندازی (شیوه اتصال ربات به لپ تاپ) روبروی بلوتوثی

در مورد نصب نرم افزار، انتخاب با خودتان است، هر کدام را که بخواهید می توانید نصب نمایید. برای نصب نرم افزار کافی ست روی آن کلیک کنید، تا پنجره زیر باز شود.

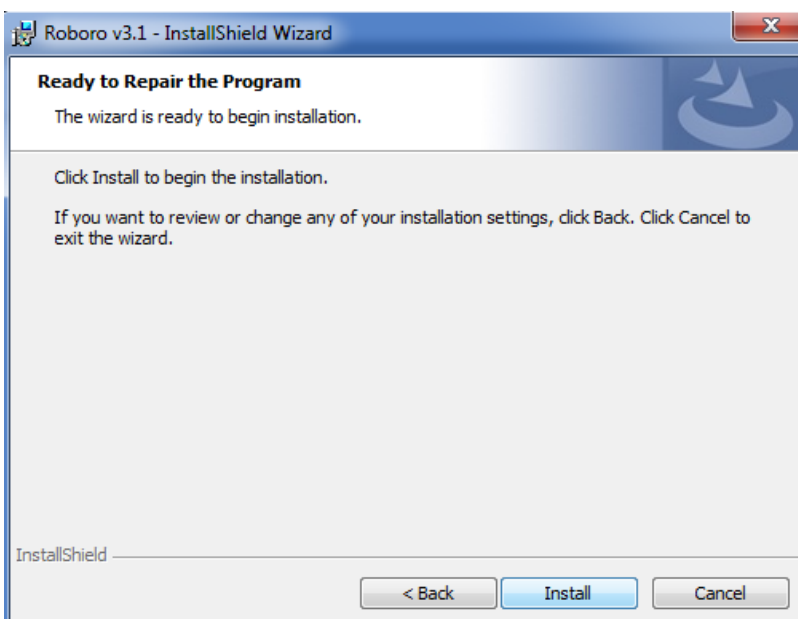


با کلیک روی گزینه **Run** نصب نرم افزار را ادامه می دهیم.

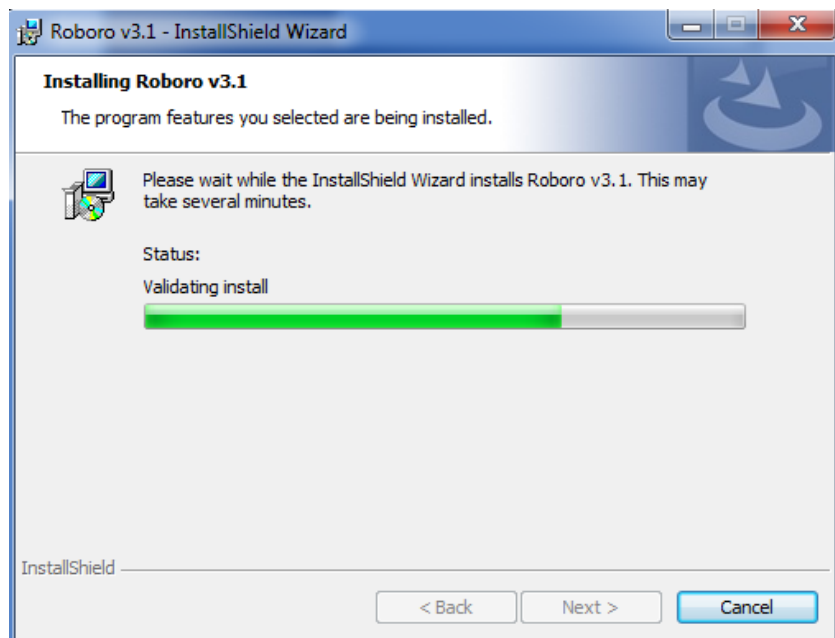
همان طور که می بینید صفحه ی زیر باز می شود با کلیک روی گزینه **Next** نصب نرم افزار را ادامه می دهیم.



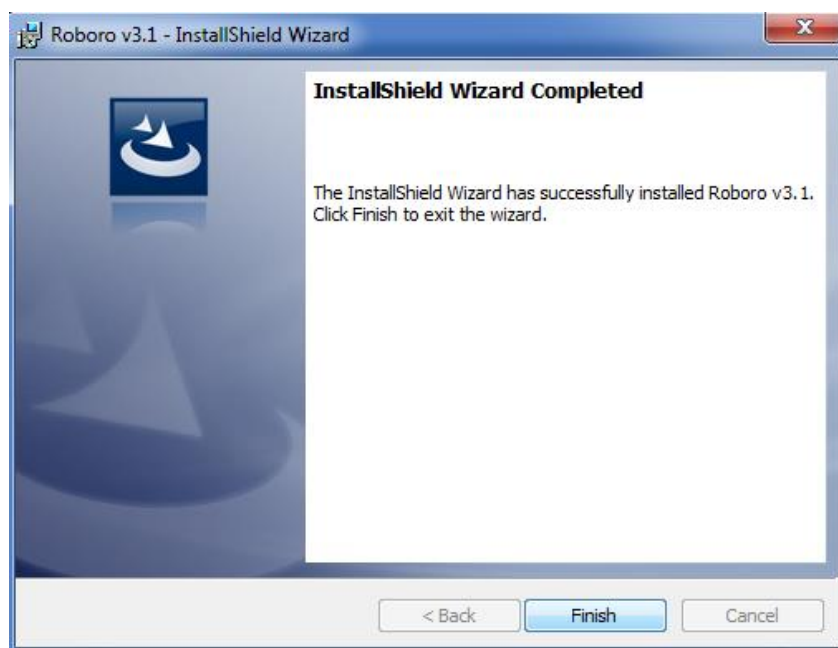
با کلیک روی گزینه Install نصب نرم افزار به راحتی انجام می شود.



ادامه نصب را در تصویر زیر مشاهده می کنید:



با ظاهر شدن پنجره زیر، مراحل نصب برنامه خاتمه می پذیرد:



همانطور که ملاحظه می کنید نصب نرم افزارها بسیار ساده است و شما می توانید با انجام مراحل فوق به راحتی هر دو نرم افزار فارسی و انگلیسی را روی سیستم خود نصب نمایید. با نصب نرم افزارها، روی دسکتاپ خود آیکن مربوط به هر برنامه را مشاهده می کنید.

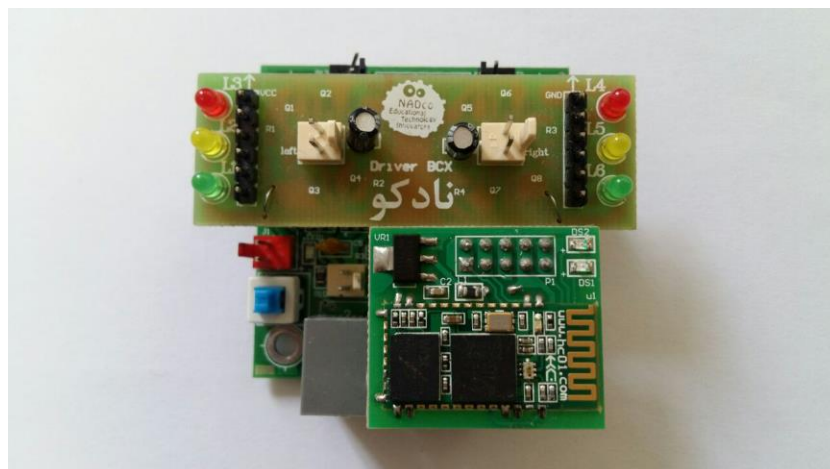
در جدیدترین نسخه ی کیتی که در اختیار دارید از ارتباط بلوتوثی برای بارگزاری برنامه بهره گرفته شده، به همین علت سیستم شما می بایست مجهز به ارتباط بلوتوثی باشد.

اگر سیستم شما مجهز به این قابلیت نیست با تهیه دانگل بلوتوث این قابلیت را فراهم کنید. این وسیله بسیار ارزان است و برای تهیه آن می توانید به نمایندگان ما و یا تمامی مراکز تجهیزات کامپیوتری مراجعه کنید.

۲-۳- معرفی اجزای روبرو

روبرو مجهز به قطعاتی می باشد که در ذیل مشاهده می کنید:

پردازنده بلوتوثی روبرو: این برد، در حقیقت مغز ربات شما می باشد، و تمام دستورات پس از اینکه در کامپیوتر نوشته شد، به این برد منتقل می شود. روی این برد یک درایور ترانزیستوری قرار گرفته است. به علت اینکه کنترل کننده ها توان راه اندازی موتور را ندارند برای به حرکت در آوردن موتور از یک مدار راه انداز جداگانه به نام درایور استفاده می شود. این مدار به ازای اطلاعاتی که از کنترل کننده دریافت می کند می تواند ولتاژ مورد نیاز موتور را تامین کند.



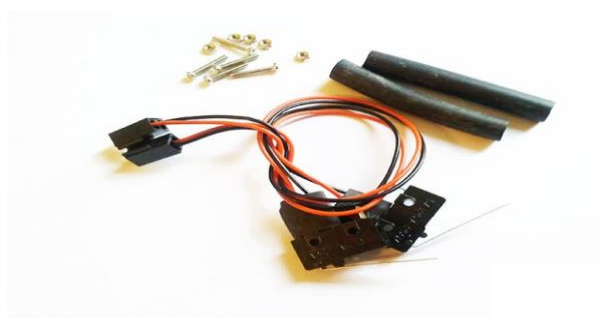
موتور و گیربکس نوع اول: گیربکس یکی از اعضای سیستم انتقال قدرت می باشد. گیربکس در حقیقت برای انتقال توان مکانیکی از یک منبع تولید توان به یک مصرف کننده و همچنین برآورده ساختن گشتاور و سرعت دورانی مورد نیاز مصرف کننده به کار می رود. گیربکس در واقع یک واسطه بین منبع توان و مصرف کننده توان می باشد که بین منبع توان و مصرف کننده توان یک انعطاف پذیری برقرار می کند.



چرخ: چرخ ها می توانند اندازه ها و شکل ظاهری متفاوتی داشته باشند. با توجه به نوع ربات، چرخ و نوع روکش آن انتخاب می شود.

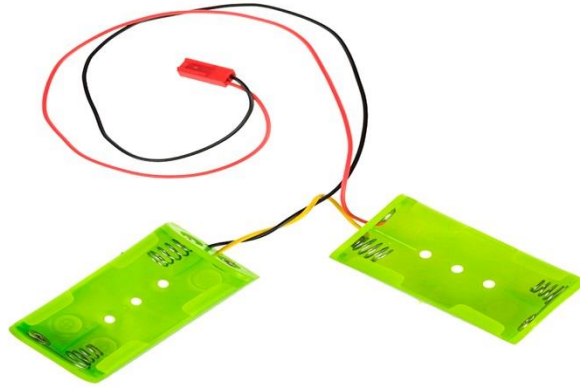


سنسور تماسی: یک نوع سنسور دیجیتالی است، که از آن برای تشخیص مانع می توان استفاده کرد.



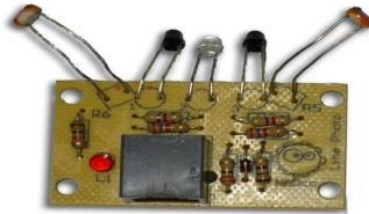
جباتری:

ولتاژ کاری ربات روبورو ۶ ولت است، که برای تامین این انرژی به ۴ عدد باتری قلمی نیاز داریم.



ماژول دو خط دو نور:

این ماژول مجهز به دو سنسور مادون قرمز و فوتوسل می باشد. از یک فرستنده و دو گیرنده مادون قرمز برای تشخیص خط و از دو فوتوسل برای شناسایی منبع نور استفاده می کنیم.



نکته

ماژول به بردی گفته می شود که مجموعه ایی از قطعات الکترونیکی (شامل انواع سنسورها، خازن، مقاومت، ترانزیستور و ...) روی آن نصب شده باشد و برای هدف معینی به کار رود.

پیچ و مهره و اسپیسر:



پیچ گوشتی:



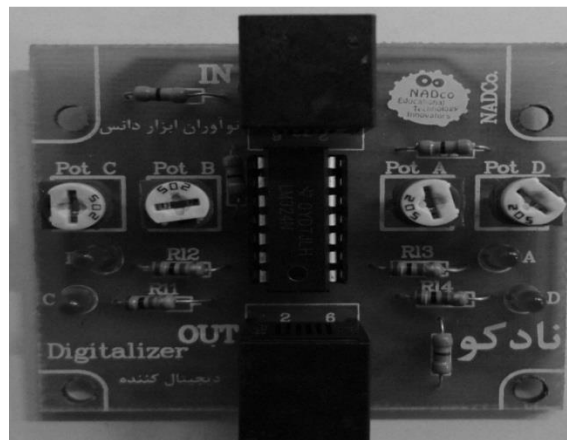
کفی روبرو:



کابل سوکتی ۶ پین کوتاه: از این کابل برای اتصال ماژول به برد پردازنده استفاده می کنیم.



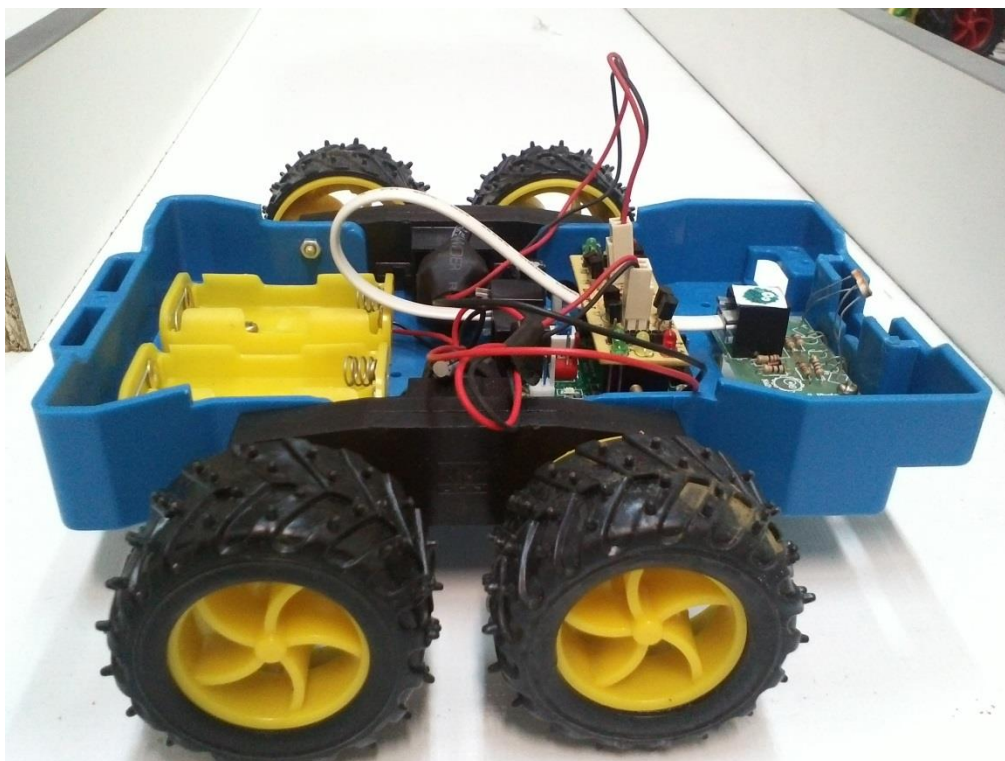
دیجیتالایزر:



جعبه روبیک، کتاب و دی وی دی: این ۳ وسیله هم جز قطعات روبرو ۱ محسوب می شوند.

۲-۴- شیوه اتصال قطعات:

برای اسمبل کردن ربات، دقت کنید که تمام قطعات مطابق زیر روی کفی روبرو قرار بگیرد:



فصل سوم

آموزش برنامه نویسی و شیوه انتقال برنامه از کامپیوتر به ربات



۳-۱- محیط نرم افزار

نرم افزار روبرو یک نرم افزار با کارایی ساده و قابلیت های زیادی ست که برای دانش آموزان طراحی شده است. شما به راحتی می توانید با این نرم افزار ربات های جالبی را طراحی کنید.

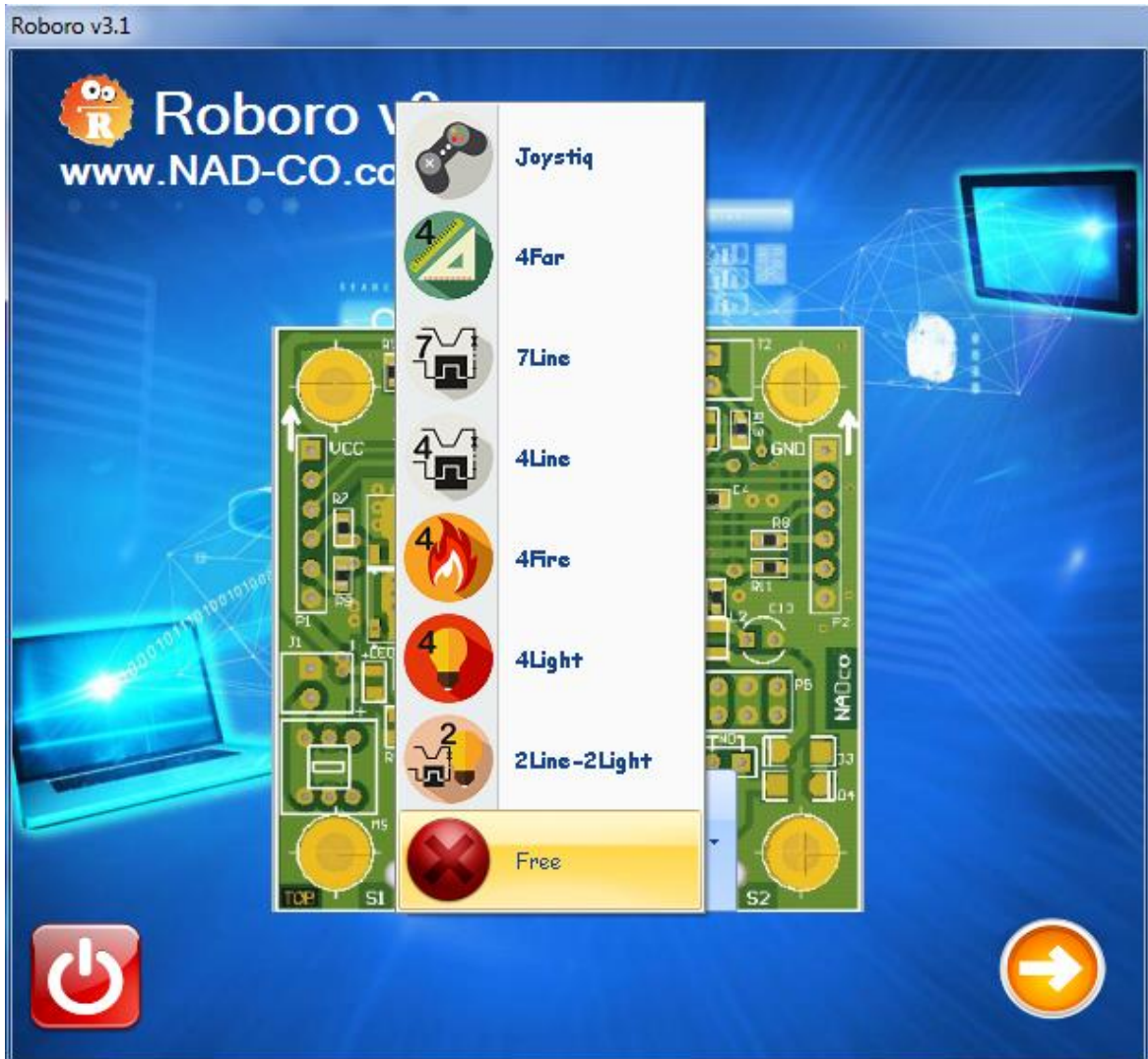


نمایی از صفحه اول نرم افزار روبرو

در بالا تصویر صفحه اول برنامه را می بینید. در این صفحه شما باید نوع ماژولی که می خواهید با آن کار کنید را انتخاب نمایید. در غیر اینصورت برای شما امکان کار با آن حسگر وجود ندارد. همانطور که در تصویر هم مشاهده می کنید دو سوکت شش پین بر روی برد قرار دارد (سوکت ۱ و سوکت ۲).

مسئله فوق به شما این امکان می دهد به طور همزمان از دو ماژول استفاده نمایید. اما نکته مهم این است که وقتی شما ماژولی را (در سخت افزار ربات) برای مثال به سوکت شماره یک متصل می نمایید ، در این صفحه هم باید به همان سوکت متصل نمایید.

برای این کار کفایت روی منوی کشویی کنار هر سوکت کلیک کنید تا لیست ماژول ها فرا روی شما قرار گیرد.



نحوه انتخاب ماژول مورد نظر و اتصال آن به سوکت

بررسی کارکرد هر ماژول:



چهار آتش: از این ماژول که متشکل از گیرنده ی مادون قرمز می باشد، برای تشخیص آتش در ربات آشنشان استفاده می کنیم.



دسته کنترل بی سیم: برای هدایت ربات به صورت بی سیم از این ماژول استفاده می کنیم.



کلپ سنسور: از این ماژول که متشکل از یک میکروفن خازنی و پتانسیومتری برای تنظیم حساسیت است، برای تشخیص صدا استفاده می کنیم.



چهار نور: از این ماژول که متشکل از ۴ سنسور نوری ست، برای شناسایی منبع نور استفاده می کنیم.



دوخط دو نور: این ماژول مجهز به یک فرستنده و دو گیرنده مادون قرمز و دو سنسور فوتوسل می باشد. از دو گیرنده مادون قرمز برای تشخیص خط و از دو فوتوسل برای شناسایی منبع نور استفاده می کنیم.



چهار خط: از این ماژول که مجهز به ۴ سنسور مادون قرمز است برای تشخیص خط در ربات مسیریاب استفاده می کنیم.



هفت خط: از این ماژول که مجهز به ۷ سنسور مادون قرمز است برای تشخیص خط در ربات مسیریاب استفاده می کنیم. (این ماژول دقت بیشتری نسب به دو ماژول ۲ خط و ۴ خط برای یافتن مسیر دارد).



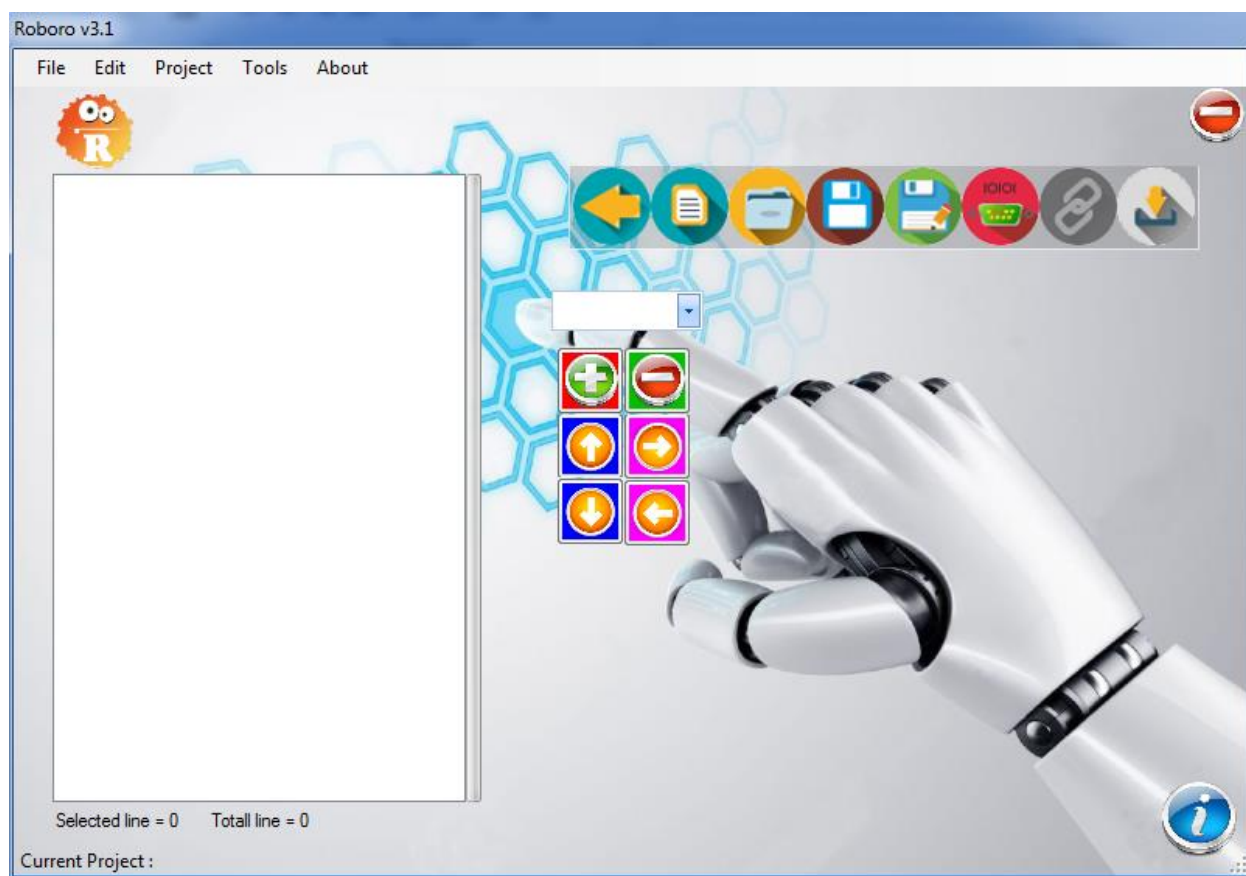
چهار فاصله: از این ماژول که مجهز به ۴ سنسور مادون قرمز است برای تشخیص فاصله استفاده می کنیم.

عدد آمده در هر ماژول نشان دهنده تعداد حسگر آن ماژول می باشد

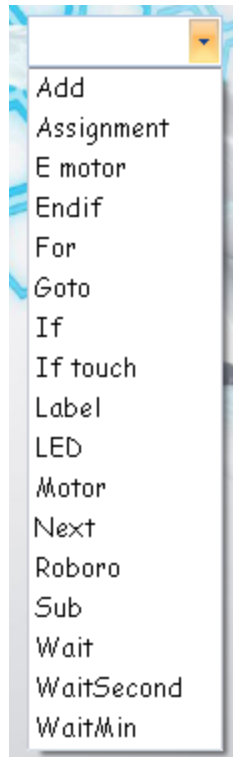
حال به وسیله کلید نارنجی که پایین سمت راست قرار دارد ، می توانید وارد صفحه اصلی برنامه نویسی شوید.



در تصویر زیر صفحه ی دوم نرم افزار روبرو را مشاهده می کنید.



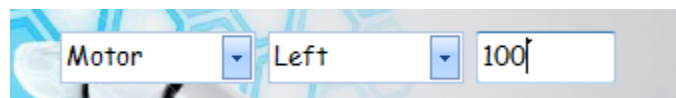
در تصویر فوق صفحه اصلی برنامه نویسی را مشاهده می کنید. در سمت چپ و در کادر سفیدرنگ، دستورات برنامه قرار می گیرد . دستوراتی که شما می توانید با آنها کار کنید ، در کادر سفید کوچک قرار دارند.



نحوه انتخاب دستورات از منوی مربوطه

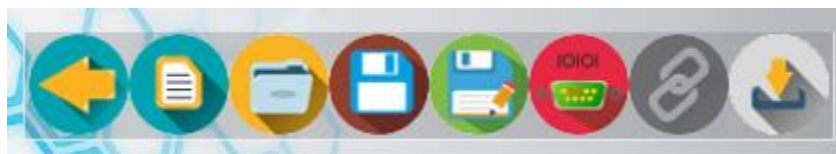
دستورات به ترتیب الفبای انگلیسی در این لیست قرار دارند.

بعضی از دستورات هستند که باید با جزئیات بیشتری به ربات منتقل شوند. مثلا هنگامی که شما دستور موتور را به ربات می دهید، باید دقیقا مشخص کنید، کدام موتور منظور شماست، موتور سمت راست ربات یا موتور سمت چپ، و نیز باید قدرت آن را هم مشخص کنید.



دستور motor و پارامترهای آن

در بالای صفحه به ترتیب از چپ به راست ، امکانات بدین صورت قرار دارند:



امکانات نرم افزار



Back: از این گزینه برای بازگشت به صفحه اول برنامه استفاده می کنیم.



New: از این گزینه برای ایجاد صفحه جدید استفاده می کنیم.



Open: از این گزینه برای باز نمودن برنامه ذخیره شده استفاده می کنیم.



Save: از این گزینه برای ذخیره نمودن برنامه استفاده می کنیم.



Save as: از این گزینه برای ذخیره نمودن برنامه با نام جدید و در جای جدید استفاده می کنیم.



Select comport: از این گزینه برای انتخاب پورت جهت اتصال سیستم به ربات استفاده می کنیم،

در ادامه نحوه ی پیدا کردن شماره پورت را توضیح خواهیم داد.



Connect: بعد از انتخاب پورت روی این گزینه کلیک می کنیم تا ربات به سیستم متصل شود و

چراغ روی برد پردازنده از قرمز به سبز تغییر کند.

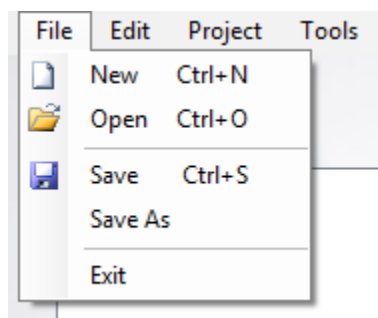


Program: از این گزینه برای انتقال برنامه به ربات استفاده می کنیم.

معرفی منوهای بالای صفحه:

File Edit Project Tools

۱- File: با فشردن این گزینه، منوهای زیر باز می شود:



New: برای باز کردن یک صفحه جدید برای برنامه نویسی

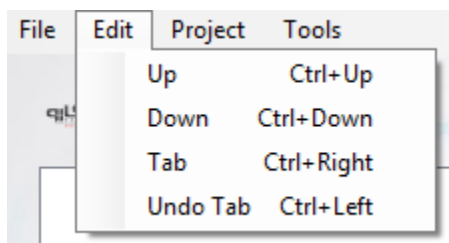
Open: از این گزینه برای باز کردن برنامه ایی که قبلاً ذخیره کرده اید، به کار می رود.

Save: از این گزینه برای ذخیره کردن برنامه استفاده می کنیم.

Save as: از این گزینه برای ذخیره کردن برنامه با یک نام جدید، در جایی جدید استفاده می شود.

Exit: برای رفتن به صفحه اول

۲- Edit: با باز کردن این گزینه، منوهای زیر در اختیار شما قرار خواهد گرفت:



Undo: برای ویرایش دستور

Redo: برای ویرایش دستور

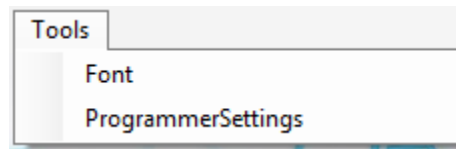
Cut: برای برش برنامه

Copy: برای کپی کردن برنامه

Paste: برای چسباندن برنامه

۳- Project ← **check syntax** ← برای کنترل و هماهنگی قسمت‌های مختلف

۴- Tools: با باز کردن این گزینه، منوهای زیر مورد باز می شوند:



Font: برای تنظیمات فونت برنامه نویسی

Programmer Settings: برای تنظیم کردن پروگرامر

۳-۲- دستور LED

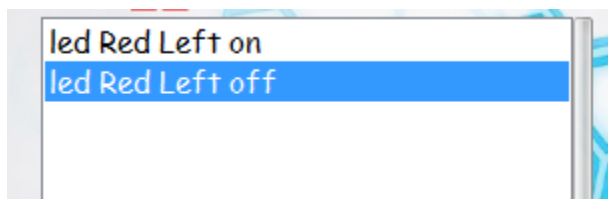
در روبروی فارسی این دستور معادل چراغ است.

به عنوان اولین دستور ، از کار با ال ای دی ها شروع می کنیم. در نرم افزار روبرو شما می‌توانید با ۴ چراغ کار کنید. این دستور در حقیقت یک نوع نشانگر است. برای روشن کردن یک ال ای دی لازم است که رنگ و جهت ال ای دی مورد نظر را انتخاب کنیم. برای انجام تنظیمات لازم است رنگ و جهت مورد نظر و وضعیت را تعیین نمایید.



دستور led و پارامترهای آن

با دستور بالا می‌توانیم ال ای دی قرمز سمت چپ را روشن کنیم . برای تایید این دستور لازم است آن را به کادر اصلی برنامه بفرستیم در غیر اینصورت دستور انجام نخواهد شد. برای این کار بر روی علامت + (add) سبز رنگی که در کنار کادر قرار دارد کلیک می‌کنیم. حالا اگر بخواهیم دستور را حذف کنیم ، بعد از کلیک کردن بر روی خط مورد نظر از برنامه ، علامت - (delete) را انتخاب می‌نماییم . برای خاموش نمودن ال ای دی نیز کافی است وضعیت خاموش (off) را انتخاب کنید و به برنامه اضافه کنید.



خاموش و روشن کردن یک led

برای جا به جا نمودن دستورات می‌توانید از دو فلش نارنجی کنار صفحه استفاده کنید و دستورات را خط به خط بالا یا پایین ببرید.

۳-۳- دستور wait

در روبروی فارسی این دستور معادل صبر است.

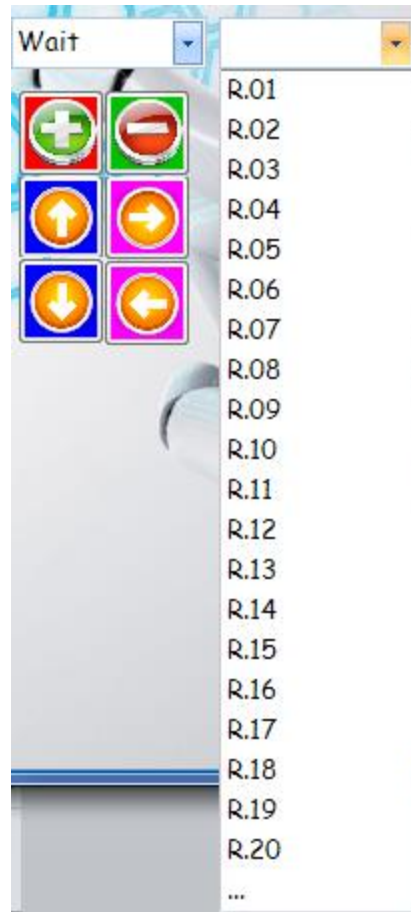
این دستور برای ایجاد وقفه در میان سطور برنامه استفاده می شود.

دستور wait ایجاد توقف برحسب میلی ثانیه را در برنامه امکان پذیر می کند.

همانطور که می دانید هر ثانیه، ۱۰۰۰ میلی ثانیه است.



این دستور عملاً در فرآیند اجرای دستورات به اندازه زمانی که ما می خواهیم ایجاد وقفه می کند و باعث ادامه کار در طول آن مدت می گردد. با انتخاب این دستور مطابق تصویر زیر، لیستی شامل رجیسترها و گزینه سه نقطه باز می شود.



دستور wait

در شکل فوق لیست رجیسترها و گزینه سه نقطه برای وارد کردن عدد دلخواه توسط کاربر را مشاهده می کنید. رجیستر یا Assignment (ثبات) واحدهایی از حافظه هستند که شما می توانید از آنها برای قرار دادن اعدادی که در طول برنامه به آنها نیاز دارید استفاده کنید. این ثباتها در حکم حافظه های کمکی هستند و تعداد آنها در برنامه روبرو ۲۰ عدد می باشد، یعنی شما ۲۰ عدد خانه حافظه دارید که از آنها می توانید برای قرار دادن اطلاعات استفاده کنید. (در مورد رجیسترها در ادامه به صورت کامل توضیح خواهیم داد)

نکته:

برای wait عدد ما می تواند از ۱ تا ۴۰۵۹ باشد.

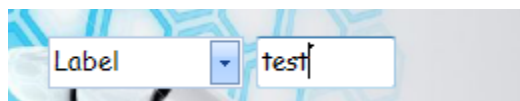
۳-۴- ساخت حلقه دائمی یا حلقه نامتناهی (دستور goto, label)

در روبروی فارسی این دستور معادل برچسب، برو به است.

در مبحث قبل شیوه ی روشن و خاموش کردن ال ای دی ها را یادگرفتید اما نکته مهم اینست که این دستورات تنها یک بار انجام می شوند ، حالا اگر بخواهیم کاری تکرار شود و برای مثال ال ای دی ها چشمک بزنند چه کار باید بکنیم؟

برای این کار از دستوری برای پرش به ابتدای برنامه و تکرار آن استفاده می کنیم.

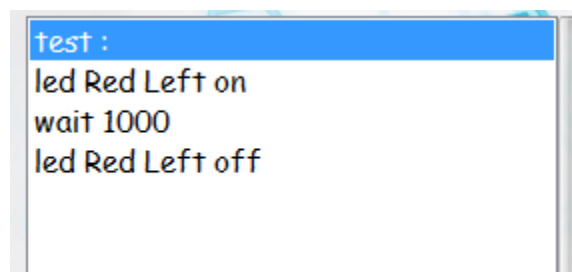
ابتدا لازم است نقطه ای که می خواهیم از آنجا کار تکرار شود و یا عملاً جایی که می خواهیم به آن پرش داشته باشیم را مشخص نماییم و نشانه گذاری کنیم . برای این کار از دستور label استفاده می کنیم.



دستور label و انتخاب یک نام برای آن

در این کادر می توانیم هر نامی شامل اعداد یا حروف را بنویسیم و آن را در محل مورد نظر قرار دهیم.

بعد از اضافه شدن به برنامه در مقابل اسم مورد نظر دو نقطه قرار می گیرد.



حالا به دستوری برای انجام پرش نیاز داریم . برای این کار از دستور goto استفاده می کنیم و مجدداً در کادر، نام برچسب مورد نظر را می نویسیم و دستور را در محلی که می خواهیم از آن جا پرش و یا تکرار اتفاق بیفتد قرار می دهیم.

```
test :
led Red Left on
wait 1000
led Red Left off
wait 1000
goto test
```

برنامه ی چشمک زدن یک لامپ

در برنامه فوق، چراغ قرمز سمت چپ به مدت ۱ ثانیه (۱۰۰۰ میلی ثانیه) روشن است و سپس به مدت ۱ ثانیه خاموش می‌شود و این کار تا زمانی که باتری شما انرژی داشته باشد یا کسی ربات را خاموش نکند، ادامه پیدا خواهد کرد.

نکته ایی که در برنامه بالا حائز اهمیت است، قرار گرفتن دستور صبر بعد از خاموش کردن چراغ می باشد. یک بار برنامه را بدون این دستور بنویسید و روی ربات بارگذاری کنید. در این لحظه چه اتفاقی می افتد!!!!؟؟؟

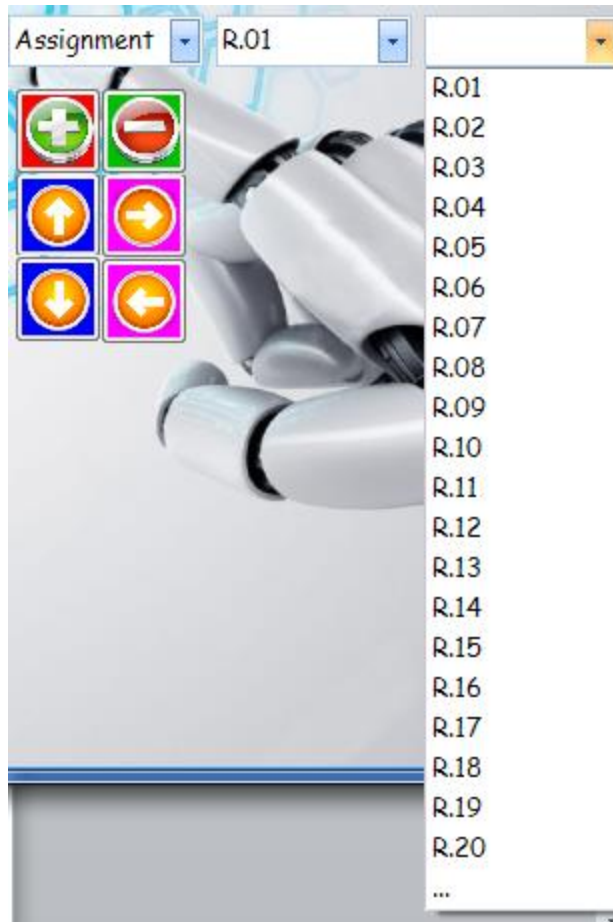
اگر بعد از خاموش کردن چراغ، دستور صبر حذف شود، شما خاموش شدن آن را به دلیل سرعت بالای اجرای دستورات در پردازنده، با چشم نمی بینید و آنچه که مشاهده می شود این است که همواره چراغ روشن است.

۳-۵- مقدار دهی به رجیسترها با دستور assignment

در روبروی فارسی این دستور معادل دستور انتساب است.

برای مقدار دهی به رجیسترها از دستور assignment، استفاده می کنیم. رجیسترها، همانطور که در ابتدای فصل بیان شد، واحدهایی از حافظه هستند که در اختیار برنامه نویس قرار می گیرند تا هر عدد دلخواهی را داخل آنها بگذارد و هر زمان که به آن مقادیر نیاز داشت، آدرس جایی که آن عدد قرار گرفته است را بدهد. قبلا گفتیم که روبرو ۲۰ خانه ی حافظه دارد، در واقع این شماره ها ی ۱، ۲، ۳، تا ۲۰ در حکم آدرس آن خانه ی حافظه محسوب می شود.

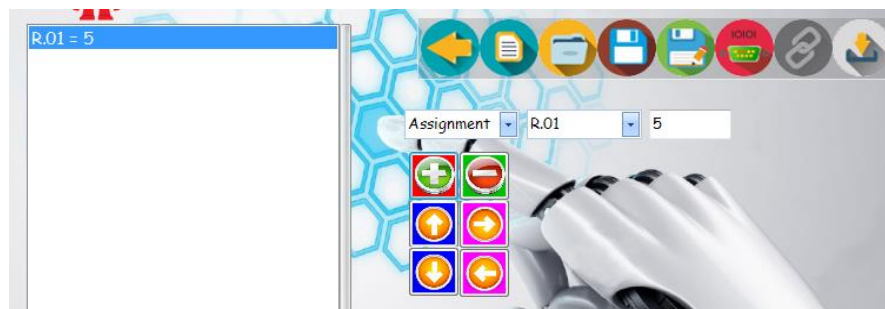
با انتخاب این دستور، دو پنجره مطابق شکل زیر باز می شود.



دستور assignment و پارامترهای آن

در پارامتر اول شماره رجیستری را که می خواهیم مقدار دهی نماییم انتخاب می کنیم در پارامتر دوم با انتخاب سه نقطه می توانیم هر عددی را انتخاب کنیم و یا می توانیم مقدار رجیستر دیگری را در رجیستر مورد نظر بریزیم .

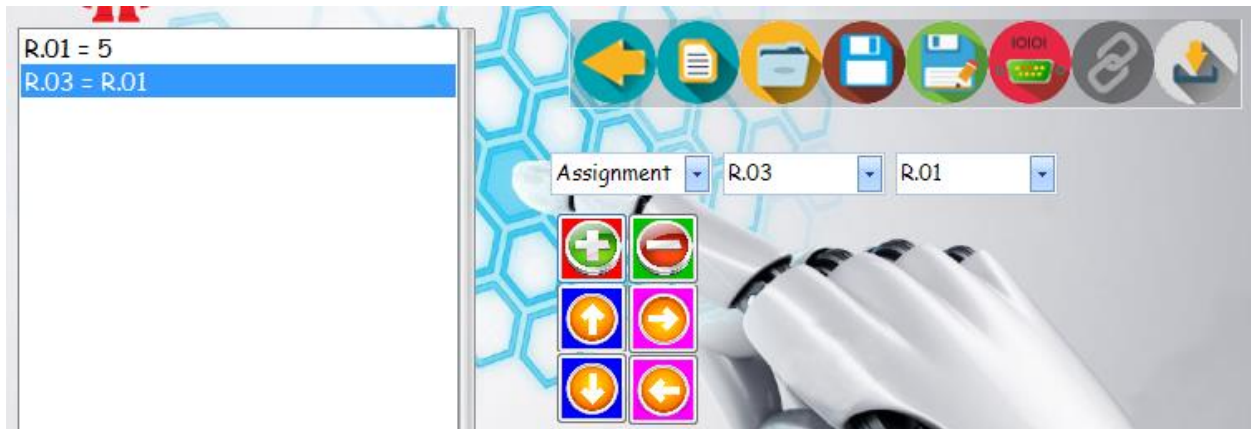
برای مثال، فرض کنید که می خواهیم عدد ۵ را درون رجیستر شماره ۱ قرار دهیم. به شکل زیر عمل می کنیم.



مقداردهی به رجیستر

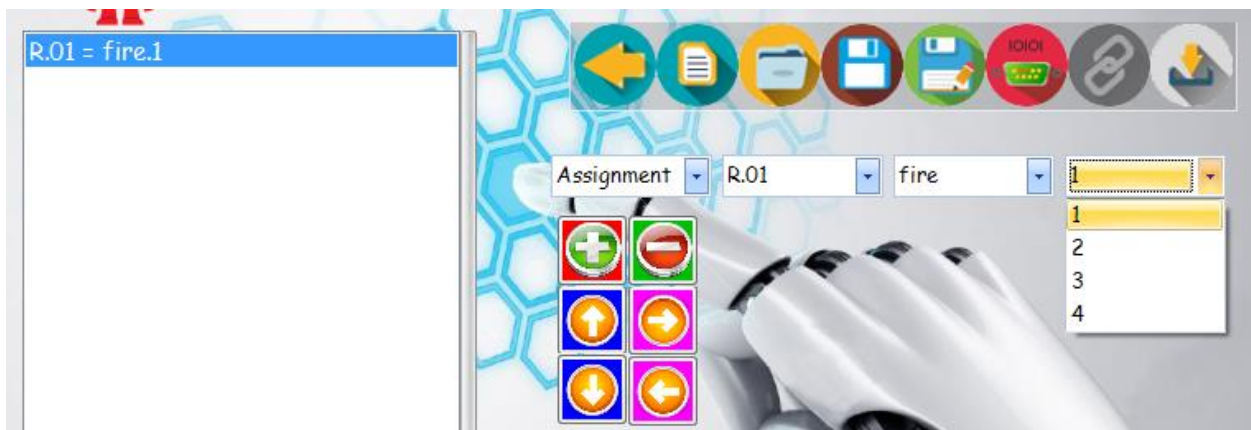
در اینجا مقدار رجیستر شماره ۱، پنج است. فرض کنید ما در طول برنامه به عدد ۵ نیاز داریم، برای فراخوانی عدد ۵، دیگر نیازی به وارد کردن این عدد نداریم، بلکه مستقیماً آدرس جایی که این عدد در آنجا ذخیره شده، یعنی رجیستر شماره ۱ را می‌دهیم.

در تصویر همانطور که می‌بینید، مقدار رجیستر ۳ نیز، به اندازه رجیستر ۱ باشد. در نتیجه مقدار رجیستر ۳ نیز پنج است.



مقداردهی به رجیستر

ما می‌توانیم مقادیر آنالوگ سنسورها را نیز درون رجیسترها بریزیم، به تصویر زیر دقت کنید:



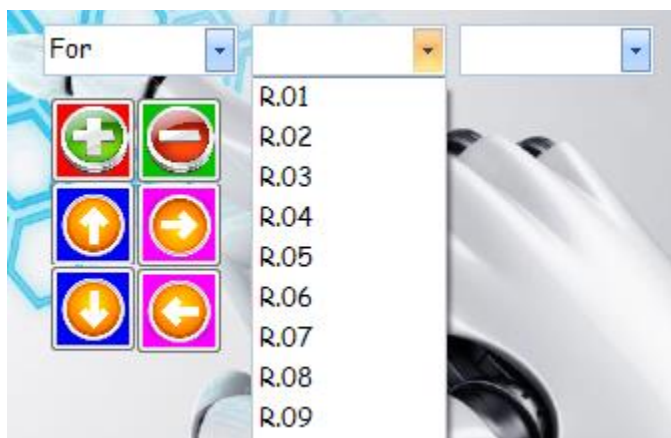
مقدار دهی به رجیستر

۳-۶- ساخت حلقه متناهی (دستور for , next)

در روبروی فارسی این دستور معادل دستور حلقه و تکرار حلقه است.

در مبحث قبل با ساخت حلقه نامتناهی یا دائمی آشنا شدید. حالا اگر بخواهیم کاری برای تعداد مشخص مثلاً ۱۰ بار یا ۱۰۰ بار یا ۱۰۰۰ بار تکرار شود چه کار باید بکنیم؟

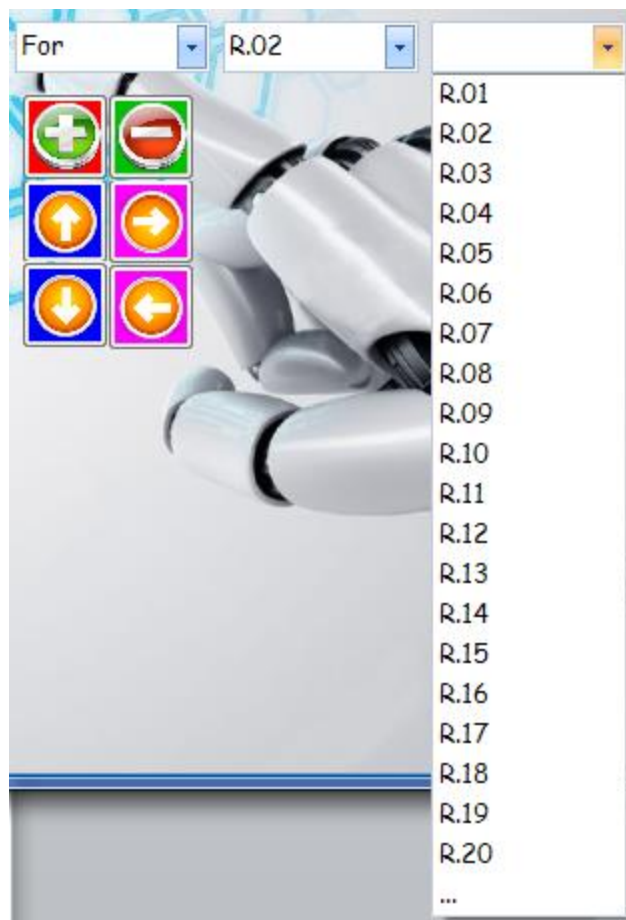
برای این کار از دستور for استفاده می کنیم . با انتخاب دستور for در مقابل شما لیستی باز می شود . این لیست شامل رجیسترهاست.



دستور for و پارامتر اول آن

در پارامتر اول این دستور شما شماره رجیستر را مشخص می نمایید، همان طور که می بینید ۲۰ رجیستر در اختیار شما قرار دارد. (هر رجیستری را می توانید انتخاب کنید)

در پارامتر دوم شما باید مقدار را مشخص نمایید . برای این کار هم می توانید با استفاده از گزینه سه نقطه ، هر عدد دلخواهی را استفاده نمایید و یا از خانه های دیگر حافظه کمک بگیرید. برای مثال شما قبلاً رجیستر شماره ۵ را مقدار دهی نموده اید ، حالا می توانید به جای مقدار این رجیستر، بگویید به میزان رجیستر شماره ۵ .



دستور for و پارامتر دوم آن

در زیر دستور for ، کاری که می خواهیم تکرار گردد (مثلاً چشمک زدن یک ال ای دی) را قرار می دهیم. اما ما نیاز به دستوری داریم که انتهای این کار را با آن مشخص کنیم. برای این منظور از دستور next استفاده می کنیم . این دستور هیچ پارامتری ندارد.

```

for R.01 = 5
led Yellow Left on
wait 1000
led Yellow Left off
wait 500
next
  
```

برنامه چشمک زدن لامپ زرد با تعداد متناهی

۳-۷- دستور Add

این دستور در روبروی فارسی معادل دستور جمع می باشد.
این دستور جزو دستوراتی است که می تواند عمل جمع را اجرا کند.
این دستور مقدار یک متغیر را به میزان دلخواه افزایش می دهد و یا دو متغیر را باهم جمع می کند.
در این دستور هر دو پارامتر اول رجیستر هستند. پارامتر سوم، هم می تواند رجیستر باشد و هم عدد. این دستور مقادیر پارامترهای دوم و سوم را با هم جمع می کند و در پارامتر اول قرار می دهد. برای مثال، شما پول هایی که در دو کشو دارید را روی هم می گذارید (با هم جمع می کنید) و در کشوی سوم قرار می دهید.
در دستور زیر مقدار رجیستر های ۲ و ۳ را با هم جمع می کنیم و در رجیستر شماره ۵ قرار می دهیم.



دستور add و پارامترهای آن (جمع مقدار ۲ رجیستر)

و یا در مثال زیر مقدار رجیستر ۶ را با عدد ۱۰ جمع می کنیم و در رجیستر ۹ قرار می دهیم.



دستور add و پارامترهای آن
(جمع مقدار یک رجیستر با یک عدد دلخواه)

۳-۸- دستور Sub

این دستور در روبروی فارسی معادل دستور تفریق می باشد.
دستور محاسباتی دیگری که در اینجا به آن اشاره کوتاهی می کنیم دستور sub است. این دستور عکس دستور add می باشد. ما با کمک این دستور می توانیم مقادیر دو رجیستر، یا یک رجیستر و یک عدد را از هم کم و حاصل را در یک رجیستر ذخیره کنیم.



دستور sub و پارامترهای آن
(تفریق مقدار ۲ رجیستر)

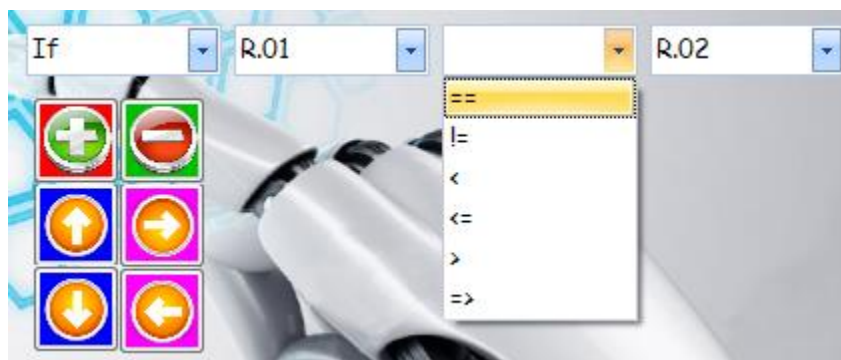


دستور sub و پارامترهای آن
(تفریق مقدار یک رجیستر با یک عدد دلخواه)

۳-۹- دستور If , Endif

این دستور در روبروی فارسی معادل دستور اگر می باشد.

دستورات شرطی در برنامه نویسی از اهمیت خاصی برخوردار هستند. دستور شرطی if کار مقایسه مقدار دو رجیستر و یا مقایسه مقدار یک رجیستر با یک عدد را توسط عملگرهایی که در زیر تعریف می شوند، انجام می دهد. این مقایسه میتواند به صورت مساوی بودن، نامساوی بودن، کوچکتر یا کوچکتر مساوی بودن و بزرگتر یا بزرگتر مساوی بودن بیان شود.



دستور if و پارامترهای آن

مفهوم	علامت
مساوي	==
نامساوي	!=
کوچکتر	<
کوچکتر مساوي	<=
بزرگتر	>
بزرگتر مساوي	>=

هرگاه در برنامه از دستورات شرطی استفاده کردیم، در انتهای دستورات مربوط به آن شرط می بایست از دستور **Endif** استفاده کنیم. این دستور در روکاری فارسی معادل **پایان شرط** است.

۳-۱۰- دستور motor

این دستور در روبروی فارسی معادل دستور **موتور** می باشد.

با انتخاب دستور motor ، شما دو پارامتر دارید. اول باید موتور مورد نظر (متصل به سمت راست یا چپ) را مشخص نمایید.



دستور motor و پارامتر اول آن

در پارامتر دوم ما باید pwm مخفف pulse width modulation یا همان توان موتورها را تعیین نماییم . در این قسمت ما هم می توانیم خود عدد مورد نظر را بدهیم و هم می توانیم از رجیسترها استفاده کنیم.

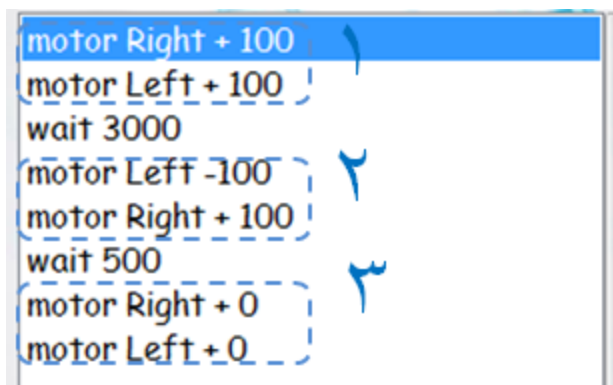
این عدد حداکثر می تواند ۱۰۰ باشد. ما می توانیم هر عددی بین ۱۰۰- تا ۱۰۰ ، قرار دهیم . یعنی زمانی که عدد ۱۰۰ را قرار داده ایم از موتور مورد نظر می خواهیم با ۱۰۰٪ توان به سمت جلو حرکت کند و در حالتی که علامت - را استفاده کنیم به معنای حرکت به سمت عقب است.

عدد صفر نیز به معنای توقف ربات می باشد. برای چرخش به هر سمت لازم است ، موتور آن سمت سرعت کمتری نسبت به طرف دیگر داشته باشد. یعنی یا صفر باشد یا منفی باشد و یا عددی کوچکتر از عدد موتور جهت مخالف باشد.

موتور چپ	ربات	موتور راست	حرکت ربات
رو به جلو		رو به جلو	رو به جلو
رو به عقب		رو به عقب	رو به عقب
خاموش		رو به جلو	چرخش به چپ
رو به جلو		خاموش	چرخش به راست
رو به جلو		رو به عقب	دور درجا به راست
رو به عقب		رو به جلو	دور درجا به چپ

جدول تعیین نحوه حرکت موتورها

اما برای چرخش به میزان مشخصی لازم است که با دستور `wait` زمان چرخش را تعیین کنیم. با کمی تمرین می توانید مقدار این `wait`ها را برای مقادیر چرخشی مشخص به دست آورید. برای مثال می خواهیم رباتمان به اندازه ۳ ثانیه به سمت جلو رفته و سپس برای ۵۰۰ میلی ثانیه به سمت چپ بچرخد و بایستد. برای این کار برنامه زیر را می نویسیم.



یک نمونه برنامه حرکتی

در حالت اول: ربات با حداکثر توان به سمت جلو می رود.

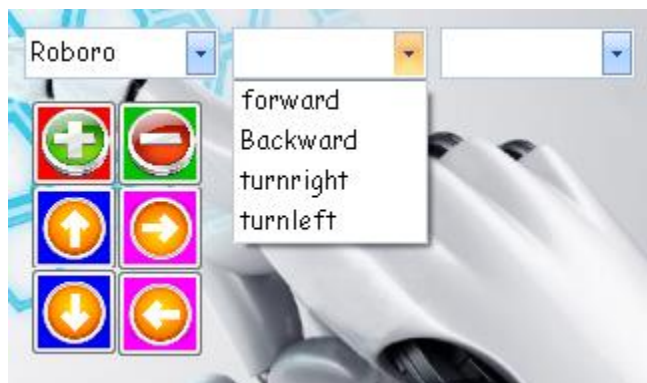
در حالت دوم: ربات با حداکثر توان به سمت چپ می چرخد.

در حالت سوم: ربات می ایستد.

۱۱-۳- دستور Roboro

این دستور در برنامه روکاروی فارسی معادل دستور ربات می باشد.

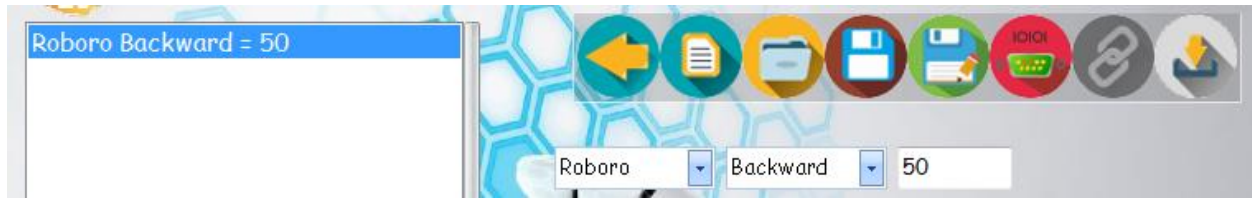
دستور حرکتی بعدی ، دستور roborobot است. این دستور را می توان شکل ساده دستور motor دانست. با این دستور که تنها چهار حالت دارد ، شما به راحتی می توانید به ربات دستور حرکت به جلو ، عقب ، چرخش به راست و یا چپ را بدهید.



دستور roborobot و پارامترهای آن

همچنین شما می توانید توان موتورها را هم تنظیم نمایید. یعنی تعیین نمایید ربات با چند درصد توان به جلو یا راست و... حرکت کند. این مقدار را در پارامتر دوم تعیین نمایید.

در برنامه زیر ربات با ۵۰ درصد توان به عقب حرکت می کند.



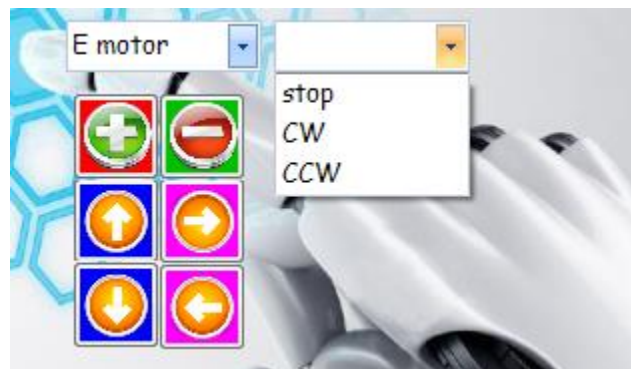
یک نمونه برنامه حرکتی با دستور roboro

۳-۱۲- دستور Emotor

این دستور در برنامه روبروی فارسی معادل دستور موتور اضافی می باشد.

این دستور برای حرکت دادن موتور اضافی استفاده می شود. منظور از موتور اضافی ، موتوری است که در حرکت ربات و به عبارتی راه رفتن آن نقشی ندارد. این موتور بر روی برد درایور قرار ندارد. بلکه روی برد اصلی قرار می گیرد.

این موتور می تواند به عنوان شوتر (وسیله ایی برای پرتاب توپ)، وسیله ایی برای اطفای حریق، وسیله ای برای کنار زدن مانع ومورد استفاده قرار گیرد.



دستور emotor و پارامتر آن

این موتور می تواند سه وضعیت داشته باشد و شما نمی توانید برای آن توان موتور را تنظیم کنید.

مفهوم	دستور
توقف موتور	stop
چرخش موتور در جهت عقربه های ساعت	CW
چرخش موتور در جهت خلاف عقربه های ساعت	CCW

۳-۱۳- سنسورها

قبل از آنکه به مطالعه‌ی دستورات مربوط به سنسورها بپردازیم، لازم است پیرامون سنسورها نکاتی را بیان کنیم.

۳-۱۳-۱- سنسور چیست؟

سنسور وسیله‌ایست که پدیده‌های فیزیکی را به الکتریکی تبدیل می‌کند. پدیده‌های فیزیکی از قبیل: نور، دما، حرکت، فاصله، تماس و ... می‌باشند. کار سنسور ها این است که این پدیده‌ها را به مؤلفه‌های الکتریکی مثل ولتاژ، جریان، مقاومت و ... تبدیل کنند هر چند که مقادیر آنها کم باشد. به وسیله تقویت کردن این مؤلفه‌های الکتریکی می‌توانیم از آنها استفاده کنیم و تغییرات آنها را ثبت و از آنها استفاده کنیم. انسانها توسط حسهای پنجگانه خود محیط اطراف خود را می‌شناسند. ربات‌ها هم برای اینکه بتوانند محیط خود را بشناسند مجبورند دارای حواسی باشند. سنسور یا معادل فارسی آن حسگر وسیله‌ایست که به عنوان وسیله احساسی ربات به کار می‌رود.

۳-۱۳-۲- تقسیم بندی سنسورها

می‌توان در یک دسته بندی کلی سنسورها را در دو بخش سنسورهای آنالوگ و سنسورهای دیجیتال تقسیم نمود.

۳-۱۳-۳- مفهوم آنالوگ و دیجیتال

طبیعت اطراف ما یک دنیای آنالوگ است. کلمه آنالوگ را می‌توان در مقابل کلمه دیجیتال یا عبارت پیوسته ترجمه کرد و همچنین دیجیتال را می‌توانیم گسسته ترجمه کنیم. در کامپیوتر و وسایل الکترونیکی اطراف خودمان همه چیز به صورت هست یا نیست وجود دارد. کامپیوتر و وسایل الکترونیکی همه چیز را به زبان صفر و یک می‌شناسند. سنسور تحریک شده است یا نه، دیسک در درایور وجود دارد یا نه، led قرمز روشن شود یا خاموش. بلندگو روشن شود یا خاموش. اما در طبیعت اینگونه نیست. مزرعه گندم و یا برنج که با تاخیر چند روزه کنار هم کاشته شده اند با اینکه همگی سبز هستند اما هر یک شدتی از نور سبز را دارند. از دید یک وسیله دیجیتال صرفاً همه آنها سبز هستند و هیچ تفاوتی بین آنها وجود ندارد.

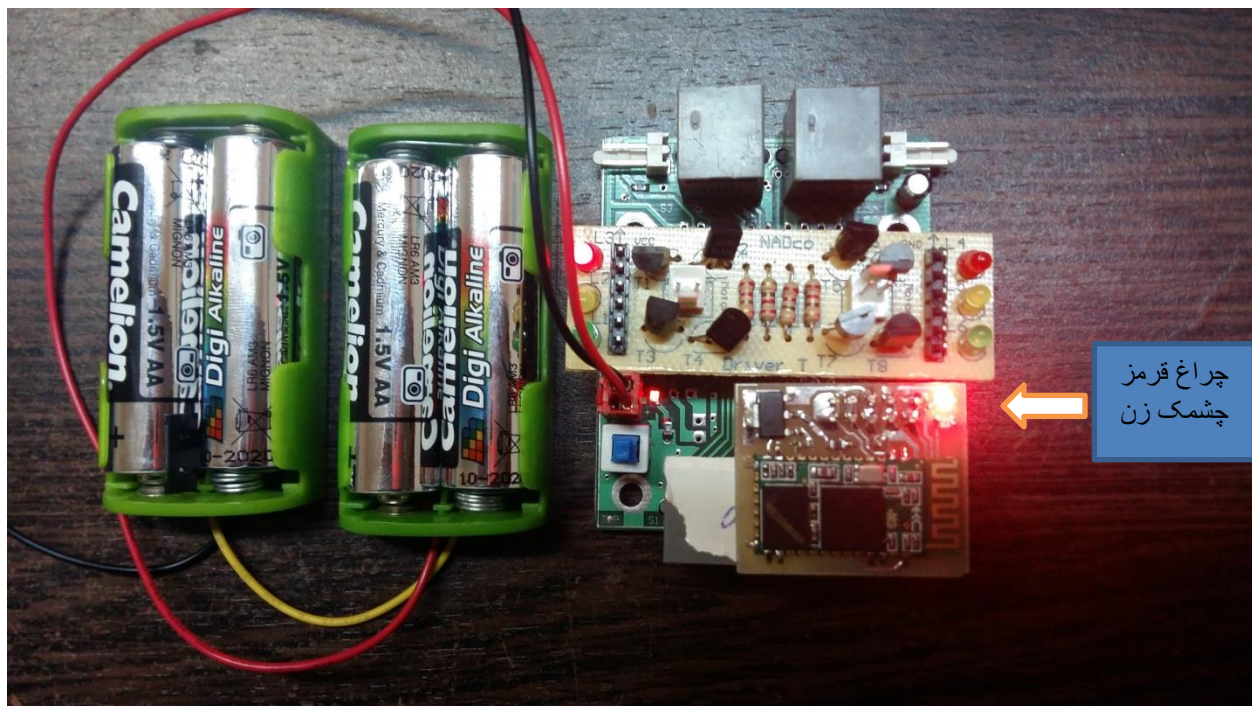
دیجیتال در مفهوم گسترده تری به معنای مقادیر متفاوت اما همچنان گسسته است. در مفهوم دیجیتال به شکل صفر و یک یا قطع و وصل ۲ سطح وجود دارد. اما می‌توان این مفهوم را گسترده تر کرد و ۴ یا ۸ سطح را برای یک متغیر تعریف کرد. مقدار متغیر باید در یکی از این سطوح قرار گیرد و نمی‌تواند در خارج از این سطوح باشد. به عنوان مثال یک دما سنچ معمولی دماهای مثل ۴۵/۸، ۲۵، ۲۰/۵ را نشان می‌دهد. اما یک

دماسنج دیجیتال با دقت یک درجه این اعداد را به صورت ۲۵، ۴۶ و ۲۱ نشان می دهد. در واقع در دنیای دیجیتال با دقت ۱، عددی مابین ۲۲ و ۲۳ وجود ندارد اما می دانیم که بی نهایت عدد در بین هر دو عددی وجود دارد.

در دوره ی روبرو ۱ شما شیوه ی کارکرد ۳ سنسور (تماسی، مادون قرمز و فوتوسل) را به صورت کامل فرا می گیرید و می توانید برنامه های مرتبط با هر یک را بیاموزید.

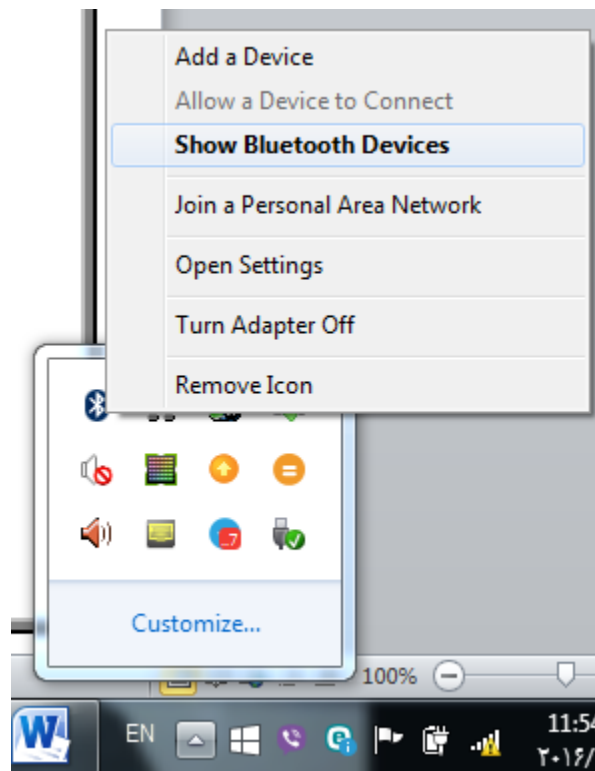
۳-۱۴- چگونگی اتصال ربات به لپ تاپ:

هنگامی که برد را روشن می کنید، LED قرمز رنگ مشخص شده در تصویر چشمک می زند. این چشمک زدن چراغ قرمز به معنی عدم اتصال برد با سیستم شماست.

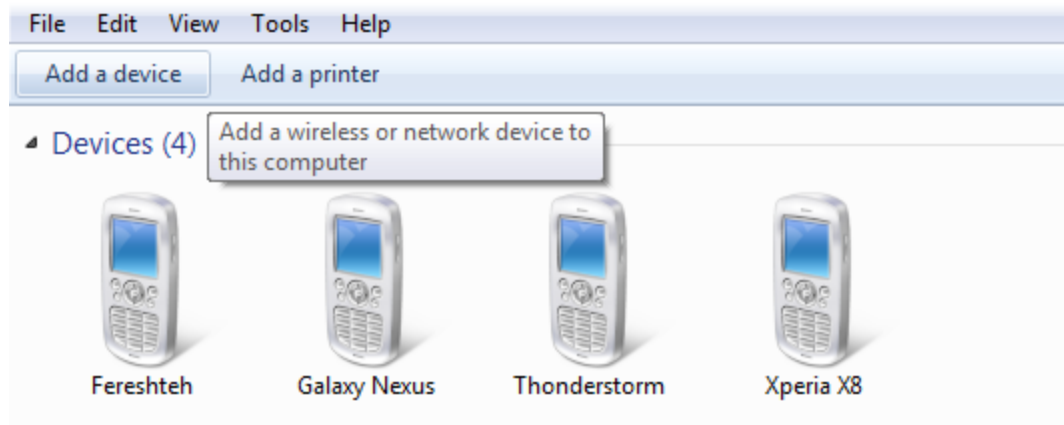


برای اتصال برد به لپ تاپ مراحل زیر را دنبال کنید:

ابتدا گزینه Bluetooth Devices و سپس گزینه show Bluetooth Devices را انتخاب کنید.



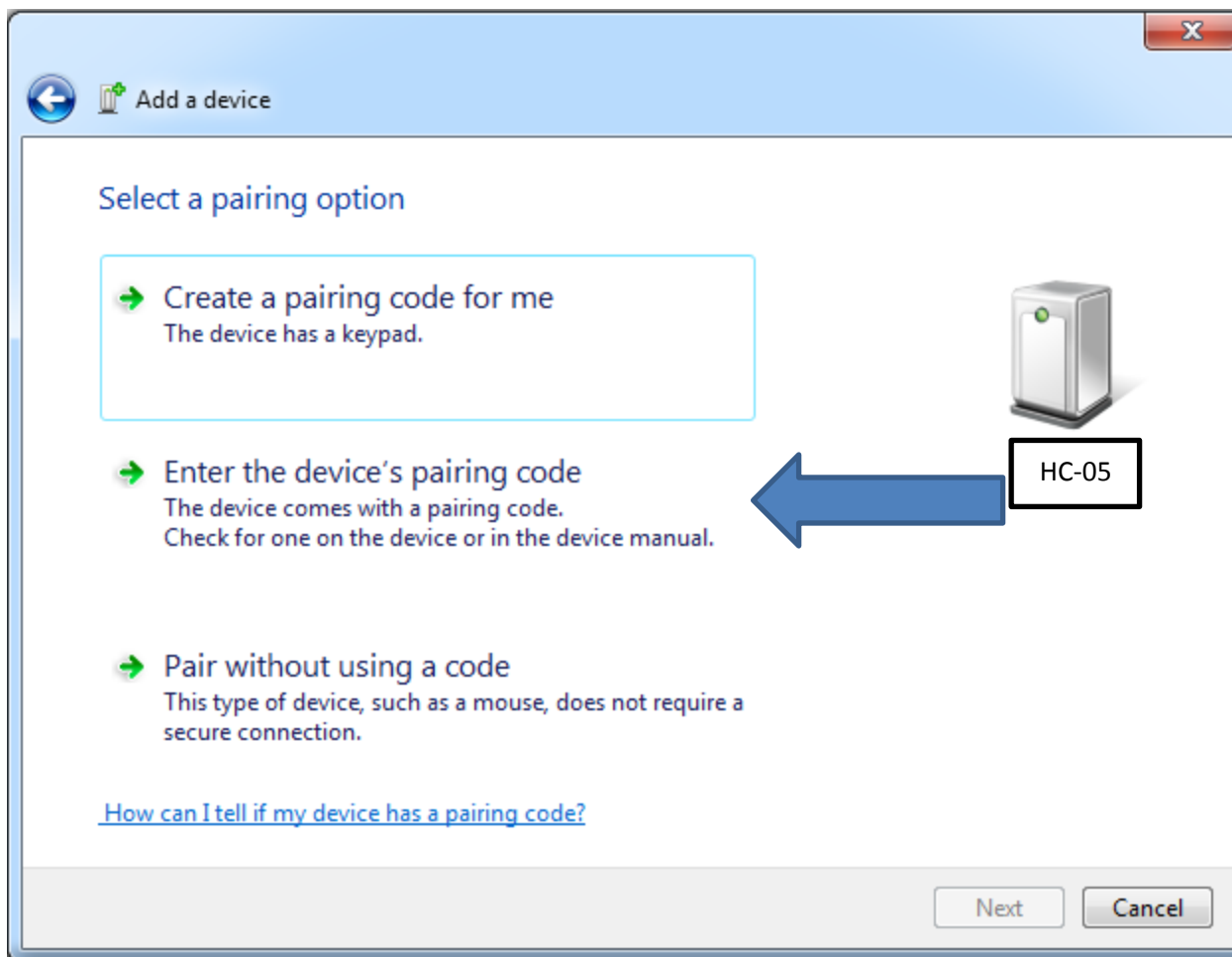
پنجره ایی که باز می شود، کلیه ی سیستم هایی که از طریق بلوتوث به لپ تاپ شما متصل شده اند را نشان می دهد، برای اضافه کرد برد پردازنده روکارو روی گزینه Add a device کلیک کنید.



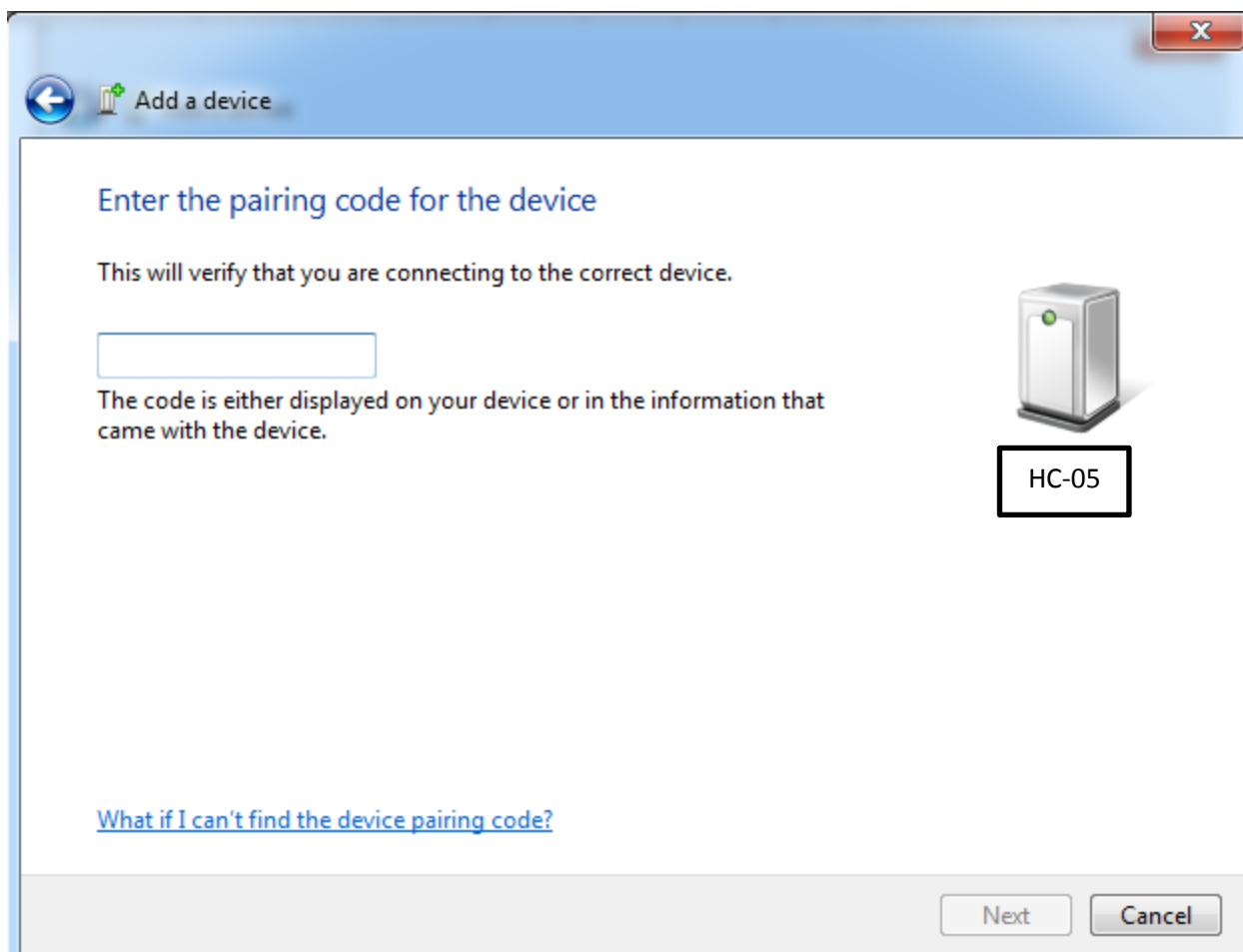
با کلیک روی گزینه **Add a device** پنجره ایی مطابق شکل زیر باز می شود، آیکن **HC-05** را انتخاب کنید و گزینه **Next** را کلیک کنید تا وارد مرحله بعدی شوید.

یک روش دیگر برای رسیدن به این مرحله آن است که وارد **control panel** شوید و روی **Devices and Printers** کلیک کنید و سپس با انتخاب **Add a device** منتظر بمانید تا سیستم شما **HC-05** را بیابد، پس از یافتن آن، آیکن **HC-05** را انتخاب کنید و روی **Next** کلیک کنید تا وارد مرحله بعدی شوید.

بعد از انجام مراحل قبل، پنجره ی زیر باز می شود:



روی گزینه "Enter the device's pairing code" کلیک کنید تا پنجره ی زیر باز شود.



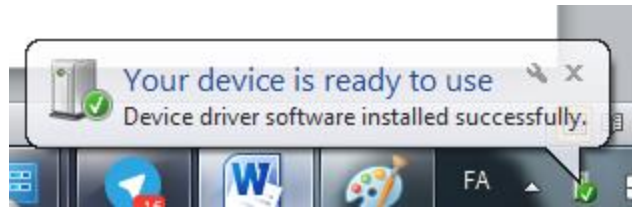
کد زیر را وارد کنید و گزینه Next را کلیک کنید.

Enter the pairing code for the device

This will verify that you are connecting to the correct device.

The code is either displayed on your device or in the information that came with the device.

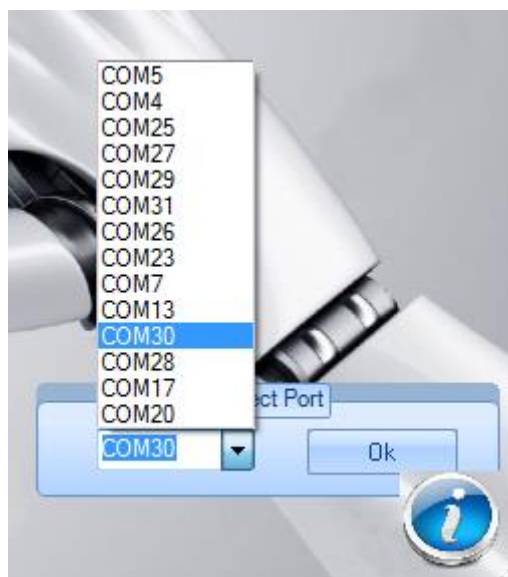
با دیدن پیام زیر، مراحل اتصال تمام می شود.



برای ادامه کار، نرم افزار روبرو بلوتوثی را باز کنید و در صفحه مربوط به کد نویسی روی گزینه **select comport** کلیک کنید.



با کلیک روی گزینه **select comport**، منوی زیر باز می شود، که باید پورت صحیح را وارد کنید.

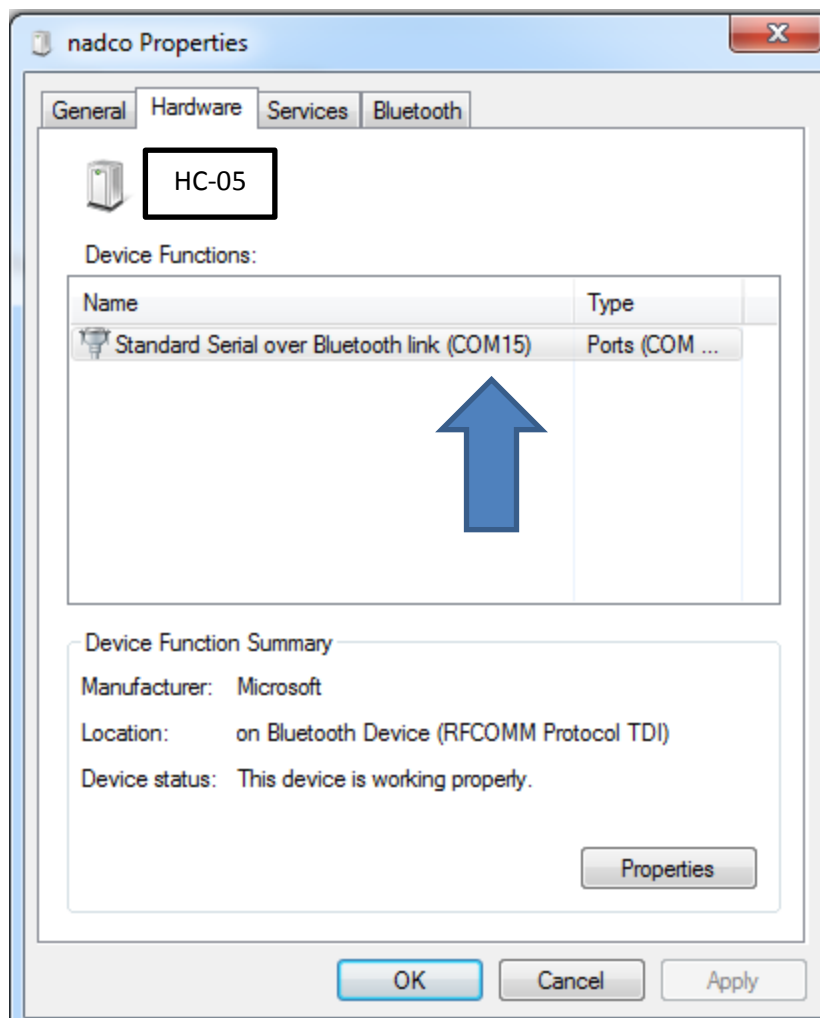


برای یافتن com صحیح ، تک تک مراحل زیر را دنبال کنید:

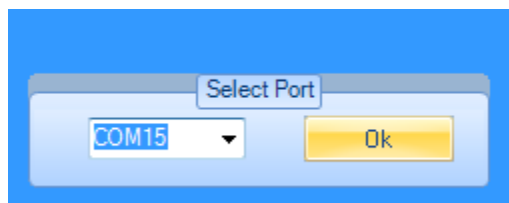
(۱) مجددا روی گزینه Bluetooth Devices کلیک کنید.



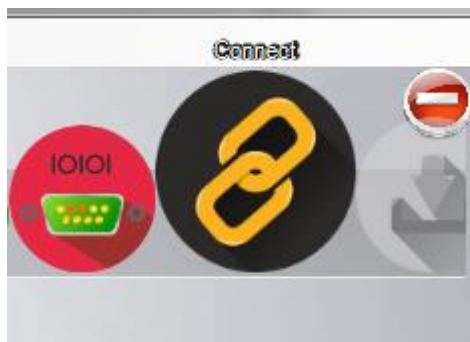
(۱) مجددا گزینه show Bluetooth Devices کلیک کنید. تصویری مطابق قبل می بینید، با این تفاوت که منوی HC-05 به سایر گزینه های شما اضافه شده است. روی HC-05 کلیک کنید تا گزینه ی زیر باز شود.



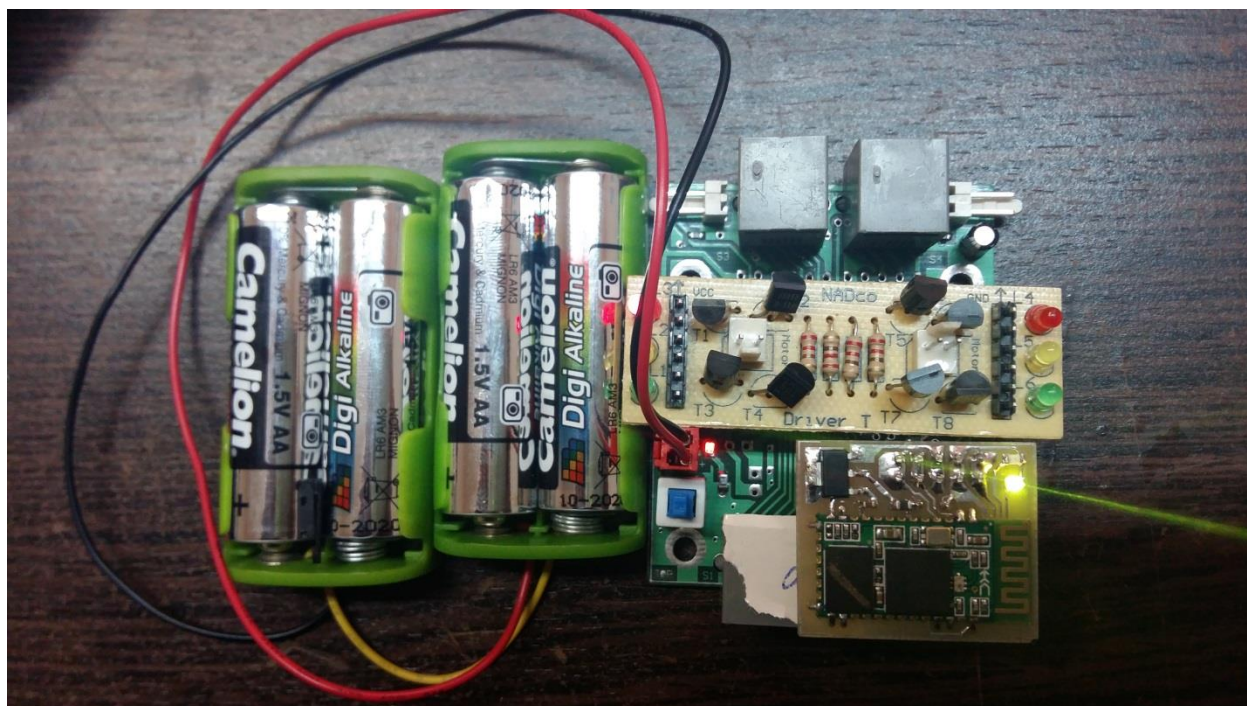
همانطور که در شکل فوق می بینید، روی زبانه ی Hardware رفته و شماره com را مشاهده کنید. حالا به محیط برنامه نویسی باز می گردیم و بعد از انتخاب com صحیح، ok را انتخاب کنید.



و سپس روی منوی connect مطابق زیر کلیک کنید.



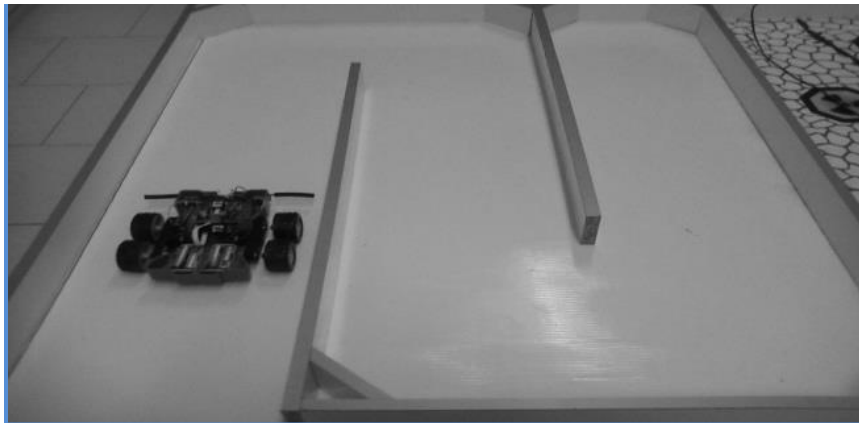
در این مرحله، چراغ سبز رنگ روی برد روشن می شود، به این معنی که برد به درستی به لپ تاپ متصل شده است.



حالا که برین ربات به لپ تاپ متصل شد ، می توانید برنامه ی دلخواه را نوشته و با کلیک رو منوی program برنامه را به ربات منتقل کنید.

فصل چهارم

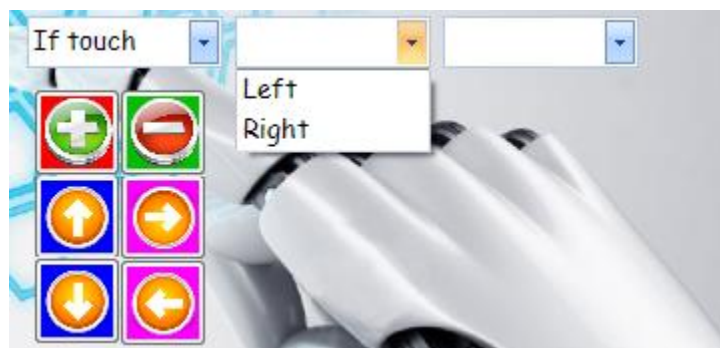
سنسور تماسی و برنامه ی حل ماز



۴-۱- سنسور تماسی با دستور iftouch

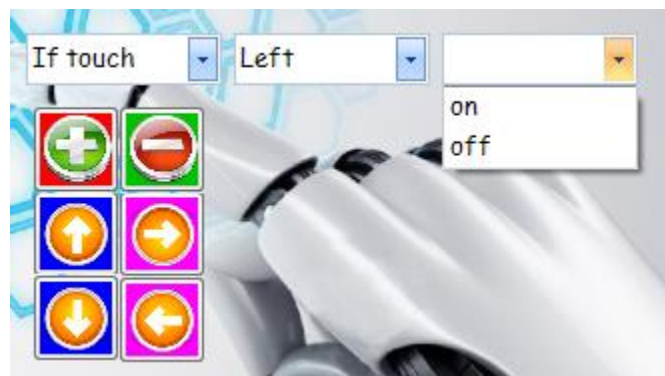
این دستور در برنامه روبروی فارسی معادل دستور حسگر تماسی می باشد.

سنسور تماسی جز سنسورهای دیجیتال است. ما در برنامه نویسی تنها باید سنسورها و اطلاعات آن ها را بررسی نماییم و در واقع برای این کار ، شرط می گذاریم. برای کار با سنسور تماسی باید از دستور iftouch استفاده کنیم و به وسیله آن وضعیت سنسور را چک نماییم. نکته ای که در اینجا وجود دارد اینست که سنسور تماسی از نوع دیجیتال است . یعنی دو حالت است . یا روشن است و یا خاموش. زمانی که سنسور ما با نقطه ای برخورد می کند ، روشن می شود و در حالت عدم برخورد خاموش است. با انتخاب دستور iftouch ، دو پارامتر داریم . پارامتر اول سنسور مورد نظر (سمت راست یا چپ) است.



دستور if touch و پارامتر اول تعیین جهت آن

پارامتر دوم ، وضعیت سنسور است ، on یا off



دستور if touch و پارامتر دوم تعیین وضعیت آن

ما وضعیت سنسور را چک می کنیم و برای آن شرط می گذاریم ، اگر شرط ما برقرار شد دستوراتی که زیر شرط نوشته ایم اجرا می شود. ما برای مشخص نمودن حدود شرط نیاز به دستور endif داریم تا به وسیله آن پایان

شرط را مشخص نماییم و در حالتی که شرط برقرار نگردد ، دستوراتی که بعد از endif قرار دارند اجرا می شوند.

حالا می خواهیم برنامه ای بنویسیم که در آن در صورت برخورد از سمت راست ال ای دی قرمز راست روشن شود و در غیر آن خاموش شود.

```
iftouch. Right on then  
led Red Right on  
endif  
led Red Right off
```

یک نمونه برنامه

در برنامه بالا در صورت برقراری شرط (برخورد از سمت راست) ال ای دی روشن می شود و در صورت عدم برخورد ، دستور بعد از endif یعنی خاموشی ال ای دی رخ می دهد.

توجه داشته باشید با لود کردن برنامه فوق ، هیچ اتفاقی رخ نمی دهد. یعنی هرچه شما سنسور تماسی را بزنید ، ال ای دی روشن نمی شود. این مسئله به این دلیل رخ می دهد که ، سرعت خواندن و اجرا شدن دستورات در هر خط در حد چند میلی ثانیه است به همین دلیل به محض روشن نمودن ربات تمامی دستورات خوانده می شود و چون در آن لحظه شرط برقرار نیست اتفاقی رخ نمی دهد. برای روشن شدن این مسئله یکبار زمانی که سنسور تماسی را زده اید نگه دارید و بعد ربات را روشن کنید ، می بینید که ال ای دی روشن است ، اما با برداشتن دست ، خاموش می شود و با زدن سنسور دیگر روشن نمی شود.

برای رفع این مشکل باید یک حلقه داشته باشیم تا این کار مدام تکرار شود و زمانی که شرط برقرار شد ، نتیجه را ببینیم.

توجه

به یاد داشته باشید در تمام برنامه هایی که از شرط و دستورات شرطی استفاده می کنیم نیاز به حلقه دائمی داریم در غیر اینصورت اتفاقی رخ نمی دهد

```
test :  
if touch. Right on then  
led Red Right on  
endif  
led Red Right off  
goto test
```

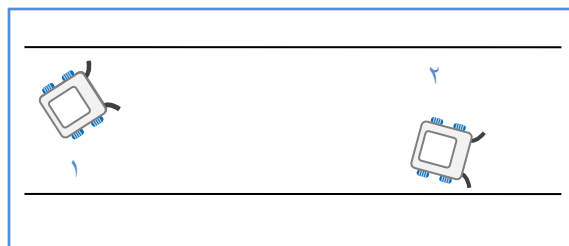
یک نمونه برنامه کامل با سنسور تماسی

۴-۲- برنامه ی حل ماز

ماز، به این راههای تو در تو می گویند که باید از یک جا وارد شویم و از طرف دیگر خارج ، یعنی دارای یک نقطه آغاز و یک نقطه پایان است. رومیان باستان به چنین مسیری لایرنث (Labyrinth) می گفتند. می دانستید مازها از نظر ریاضی، قابل مطالعه هستند؟ یکی از روبات هایی که در زمینه طرح ریزی حرکت، جهت تشخیص مسیرهای صحیح مورد توجه متخصصان قرار گرفته اند روبات های ماز هستند. این روبات ها وظیفه پیدا کردن راه صحیح از بین بی راهه های فراوان در یک ماز را بر عهده دارند.



در قسمت قبل پیرامون کار با سنسور تماسی را توضیح دادیم و یک برنامه نمونه هم نوشتیم، حالا با هم الگوریتم مسیر یابی را بررسی می کنیم. فرض کنید که ربات در یک مسیر که ممکن است گاهی مستقیم و گاهی همراه با پیچ و خم هایی هم باشد، قرار گرفته است اما ربات مستقیم حرکت نمی کند و به دیواره ها برخورد خواهد کرد. باید به نحوی ربات را برنامه ریزی کنیم که بتواند در صورت برخورد با دیواره ها با چرخش مناسب راه خود را ادامه دهد، در این مساله ۲ حالت وجود دارد که در تصویر زیر مشاهده می کنید:



حالت ۱: حالتی که سنسور سمت چپ به دیوار برخورد خواهد کرد.

حالت ۲: حالتی که سنسور سمت راست به دیوار برخورد خواهد کرد.

در حالت ۱ ربات از حالت مستقیم خارج شده و به دیوار برخورد کرده است برای برگشت به مسیر اصلی باید به راست بپیچد. در حالت ۲ ربات برای اینکه به مسیر اصلی برگردد باید به سمت چپ حرکت کند. فرض کنید می‌خواهیم برنامه ای بنویسیم که در آن ربات زیر در زمین نمایش داده شده، مسیر حرکت خود را پیدا کند.

برنامه:

Maz:

Motor left +100
Motor right +100

با این دستور ربات به صورت مستقیم حرکت می‌کند.

Iftouch. Left on then

Iftouch. right off then

Motor left +100
Motor right -100

Wait 400

Endif

Endif

اگر سنسور تماسی سمت چپ به دیوار برخورد کند، یعنی مانع سمت چپ قرار دارد، ربات باید به مدت ۴۰۰ میلی ثانیه به سمت راست بچرخد

Iftouch. right on then

Iftouch. Left off then

Motor left -100
Motor right +100

Wait 400

Endif

Endif

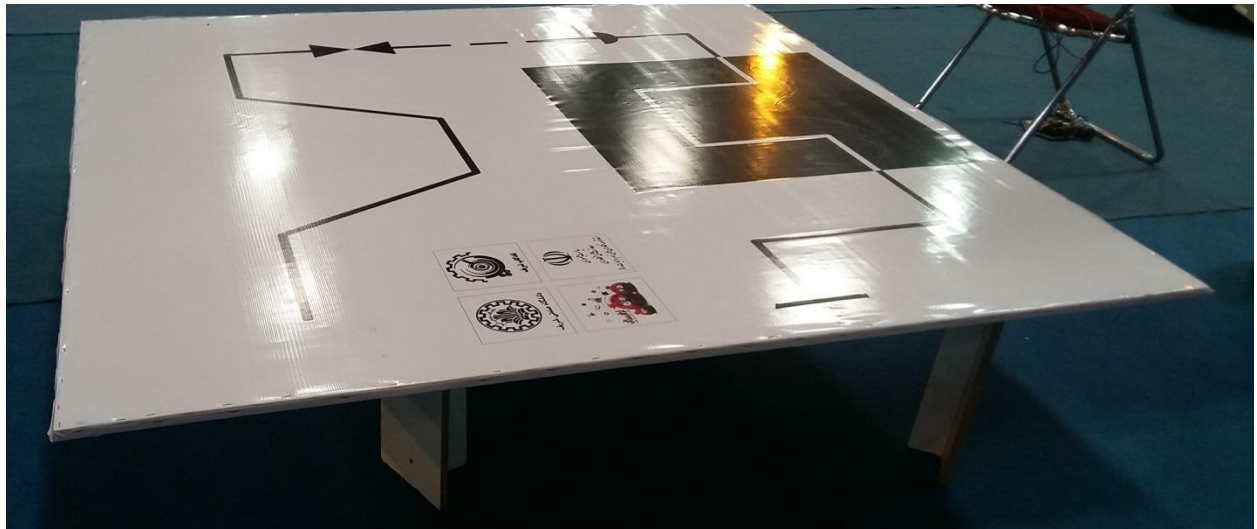
Goto maz

اگر سنسور تماسی سمت راست به دیوار برخورد کند، یعنی مانع سمت راست قرار دارد، ربات باید به مدت ۴۰۰ میلی ثانیه به سمت چپ بچرخد.

میزان **wait** بهینه را باید برای ربات های متفاوت با تست به دست آورد ، این میزان به موتورها و نوع و قدرت باطری ها بستگی دارد. مقداری که ما در برنامه داده ایم تنها برای مثال بوده است . در کل باید میزان چرخش داشته باشیم که هم نه خیلی کم باشد و نه خیلی زیاد.

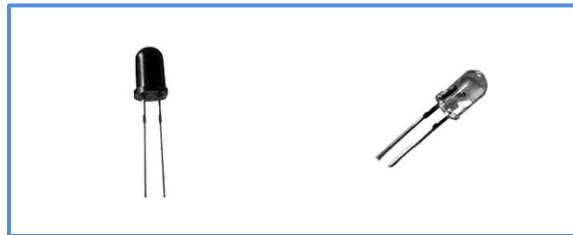
فصل پنجم

سنسور مادون قرمز و برنامه ی تعقیب خط



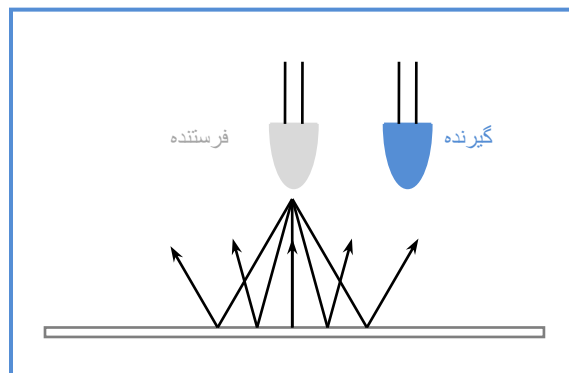
۵-۱- سنسور مادون قرمز

ما از فرستنده و گیرنده مادون قرمز برای تشخیص خط استفاده می کنیم. نور مادون قرمز در طیفی قرار دارد که قابل رویت با چشم نیست.

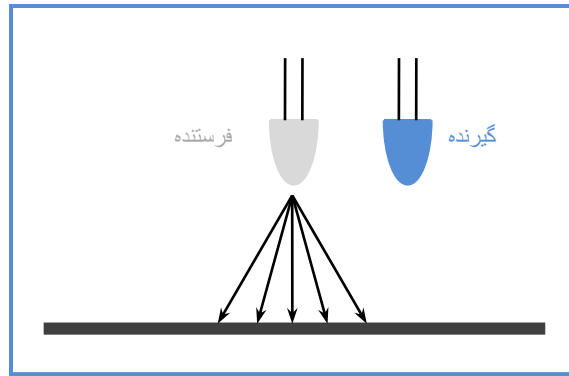


از راست به چپ فرستنده مادون قرمز و گیرنده مادون قرمز

زمانی که ما روی خط مشکی قرار داریم نور مادون قرمز تابیده شده با میزان کمتری بازتاب می شود و زمانی که روی سطح سفید هستیم بازتاب نور بیشتر است.



زمینه سفید



زمینه سیاه

ما از سیاهی مطلق تا سفیدی مطلق طیف گسترده ای داریم شامل توسی ، خاکستری و ... و در همه این رنگها بازتاب کمتر یا بیشتر می شود. به همین دلیل ما با کمیتی پیوسته روبه رو هستیم . سنسور خط، سنسوری آنالوگ است و برخلاف سنسور تماسی که دیجیتال بود، نیست . بلکه این سنسور اطلاعات و وضعیت را به صورت عددی ما بین ۰ تا ۱۰۲۳ می دهد. یعنی زمانی که سطح روشن تر است و بازتاب بیشتر است عدد بزرگتر و در حالت عکس نیز عددی کوچکتر می دهد.

معمولاً یک مرز وجود دارد که حسگر روی سطوح تیره عددی کوچکتر از آن عدد و روی سطوح روشن عددی بزرگتر از آن عدد به ما می دهد. قاعده خاصی برای به دست آوردن این عدد وجود ندارد. و تنها از طریق تست به دست می آید. این عدد بستگی به گیرندگی گیرنده ها دارد و برای همه حسگرها یکی نیست . برای دریافت نتیجه بهتر ، توصیه می شود با حوصله این عدد را به دست آورید.

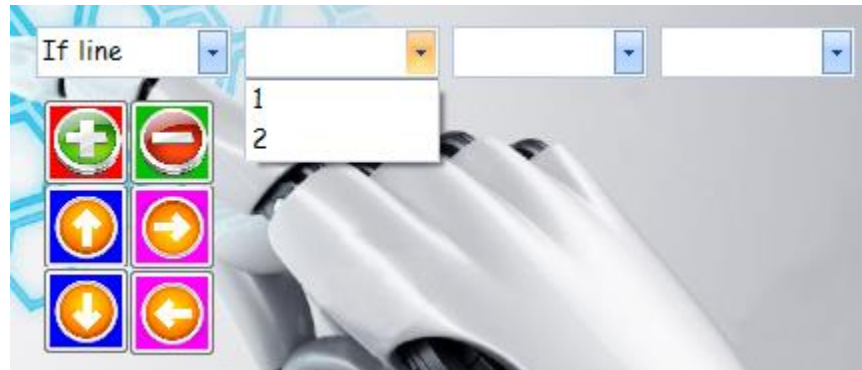
برای تست سالم بودن فرستنده ، زمانی که ربات روشن است و ماژول به برد اصلی متصل است ، با دوربین به فرستنده نگاه کنید در این حالت باید سفید باشد ولی بدون دوربین هیچ نوری ندارد.

ماژولهای متعددی با مادون قرمز برای شما طراحی شده است. ماژول دو خط دو نور، ماژول چهار خط و ماژول هفت خط. شیوه کار کردن با هر سه همانند یکدیگر می باشد. در اینجا برای نمونه شیوه برنامه نویسی با سنسور دو خط را آموزش می بینید. خودتان به همین ترتیب می توانید برای سایر ماژول ها از همین شیوه استفاده کنید.

نکته

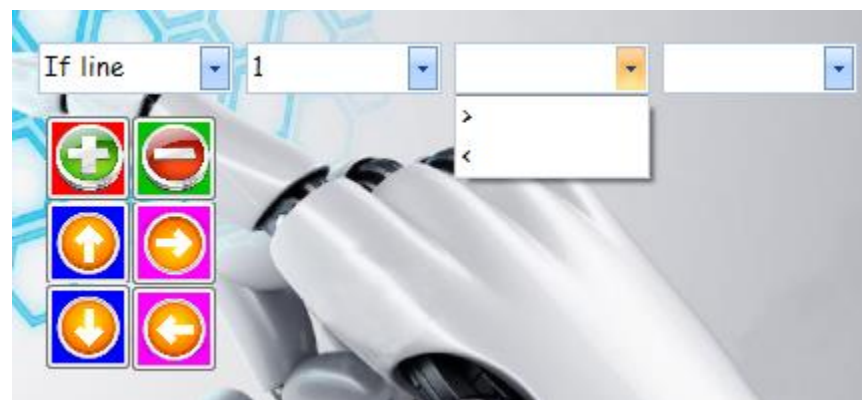
برای کار با ماژول دوخط دو نور لازم است در صفحه اول نرم افزار این ماژول را روی سوکت ۱ یا ۲ انتخاب نمایید در غیر اینصورت دستورات لازم برای این سنسور در لیست دستورات قرار نمی گیرد.

با انتخاب دستور **If line** که در انتهای لیست اضافه شده است . شما سه پارامتر دارید . پارامتر اول شماره سنسور است . شما ۲ سنسور دارید که به ترتیب از چپ به راست شماره گذاری می کنید . در اینجا شماره سنسوری که می خواهید بررسی نمایید را انتخاب می کنید.



دستور **if line** و پارامتر اول تعیین شماره سنسور

پارامتر دوم ، علامت مقایسه است . زیرا شما باید مقدار این سنسور را با عددی که آن را رمز می نامیم مقایسه کنید و پارامتر سوم نیز عدد مرز است که هم می توانید از رجیسترها استفاده کنید و هم خودتان عدد بدهید.



دستور **if line** و پارامتر دوم تعیین علامت بزرگتری، کوچکتری

برای درک بهتر، فرض کنید که مدیر مدرسه شرط می گذارد نمره بالای ۱۸ را به اردو ببرد. برای این کار نمرات شما با عدد ۱۸ مقایسه می شود و اگر بیشتر از ۱۸ بود، شما به اردو می روید در غیر اینصورت، خیر. پارامتر دوم، علامت مقایسه است. زیرا شما باید مقدار این سنسور را با عددی که آن را رمز می نامیم مقایسه کنید و پارامتر سوم نیز عدد مرز است که هم می توانید از رجیسترها استفاده کنید و هم خودتان عدد بدهید.

برنامه زیر هم به درک بهتر کمک می کند و هم برنامه ای برای به دست آوردن عدد مرز است.

```
test :
ifline.1 < 500 then
led Red Right on
endif
led Red Right off
goto test
```

یک نمونه برنامه برای تعیین عدد مرز

در برنامه فوق ما تنها سنسور ۱ را بررسی نموده ایم و لازم است برای سنسور شماره ۲ نیز، برنامه ای مشابه بنویسید.

در برنامه بالا ما عدد ۵۰۰ را به عنوان مرز فرض نموده ایم. این انتخاب به دلیل اینست که ۵۰۰ وسط اعداد است و برای شروع گزینه مناسبی می باشد. در این حالت با بردن سنسور به روی سطح سیاه باید ال ای دی قرمز راست روشن شود و روی سطح سفید نیز خاموش شود. اگر خلاف این رخ دهد باید عدد را عوض نمایید. برای مثال اگر در همه حالات روشن بود، یعنی سنسور ما به سطح سفید هم عدد کوچکتري اختصاص می دهد و ما باید عدد کوچکتري را تست کنیم.

دقت داشته باشید که ماژول را به همان سوکتی متصل کرده باشید که در صفحه اول متصل نموده اید.

عدد مرز بر روی سطوح مختلف متفاوت است و جنس سطح و میزان نور محیط می تواند بر آن تاثیر گذارد.

۵-۲- برنامه ربات تعقیب خط با دو ماژول دوخط دو نور

ما می خواهیم رباتی طراحی کنیم که بتواند خط را تعقیب کند. در اینجا ما دو سنسور داریم ابتدا باید حالاتی که سنسورها می توانند روی خط قرار بگیرند را بررسی نماییم.

۱. سنسور شماره یک روی خط و سنسور شماره دو روی سطح سفید باشد : در این حالت ربات باید به سمت چپ بپیچد.

Test:

```
ifline.1 < 500 then
```

```
ifline.2 > 500 then
```

```
motor right +100
```

```
motor left -100
```

```
endif
```

```
endif
```

۲. سنسور شماره دو روی خط و سنسور شماره یک روی سطح سفید باشد : در این حالت ربات باید به سمت راست بپیچد.

```
ifline.1 > 500 then
```

```
ifline.2 < 500 then
```

```
motor right -100
```

```
motor left +100
```

```
endif
```

```
endif
```

۳. هر دو سنسور روی خط قرار گرفته باشند: در این حالت ربات باید مستقیم حرکت کند.

```
ifline.1 < 500 then
```

```
ifline.2 < 500 then
```

```
motor right +100
```

```
motor left +100
```

```
endif
```

```
endif
```

```
goto test
```

فصل ششم

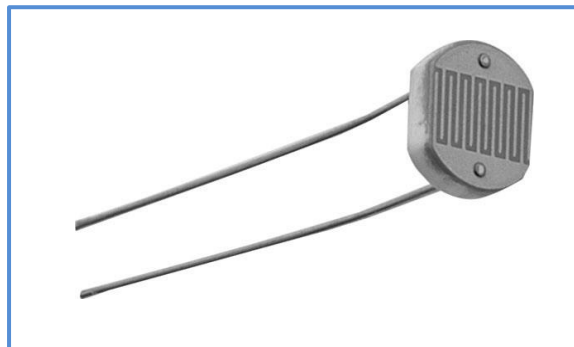
سنسور فوتوسل و برنامه ی تعقیب نور



*لیگ تعقیب و گریز در مسابقات نادکاپ

۱-۶- سنسور نوری

به سنسورهایی اطلاق می شود که در هنگام وجود نور مرئی یا نامرئی مشخصات الکتریکی خود را تغییر می دهند. در واقع کمیت فیزیکی نور را توسط کمیت الکتریکی (سیگنال الکتریکی) اندازه گیری می کنند.

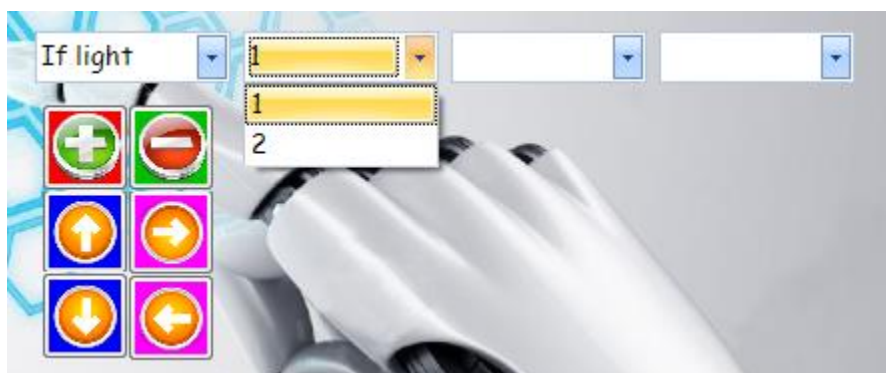


سنسور نوری

با سنسور نوری (فوتوسل) ما دو ماژول دو خط دو نور و ماژول چهار نور را طراحی کردیم، که در اینجا طرز کار کرد ماژول ۲ خط ۲ نور را می توانید یاد بگیرید.

با هر کدام از این ماژول ها که خواستید، برنامه بنویسید، به صفحه نخست برنامه روبرو رفته و آن را به سوکت یک یا دو متصل کنید (فقط حواستان باشد این شماره سوکت در برنامه با شماره سوکت روی رباتتان باید یکسان باشد)

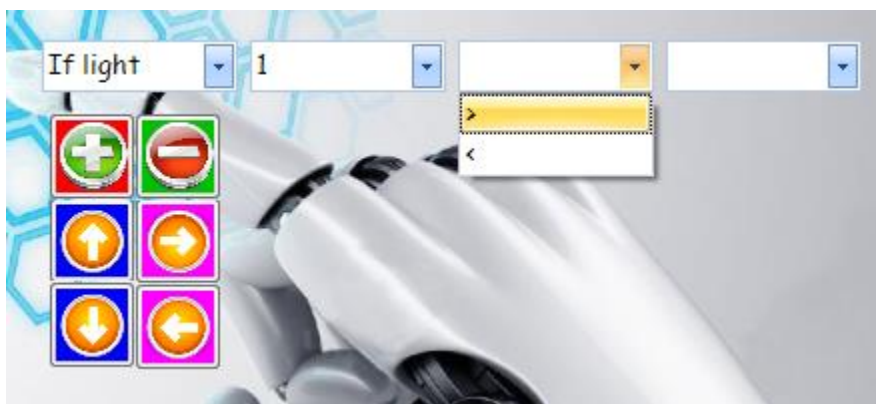
اگر ماژول دو خط دو نور را انتخاب کردید، دستور خط و نور در انتهای لیست دستورات ظاهر خواهد شد، دستور خط را در فصل گذشته بیان کردیم، حالا نوبت به سنسور نوری می رسد. با انتخاب این دستور کادر زیر باز می شود.



انتخاب شماره سنسور دستور if light

در کادر اول شماره ی سنسور را مشاهده می کنید، که به ترتیب از چپ به راست شماره گذاری می کنید . در اینجا شماره سنسوری که می خواهید بررسی نمایید را انتخاب می کنید.

پارامتر دوم ، علامت مقایسه است . زیرا شما باید مقدار این سنسور را با عددی که آن را مرز می نامیم مقایسه کنید و پارامتر سوم نیز عدد مرز است که هم می توانید از رجیسترها استفاده کنید و هم خودتان عدد بدهید.



انتخاب علامت بزرگتر، کوچکتری در دستور if light

برنامه زیر هم به درک بهتر کمک می کند و هم برنامه ای برای به دست آوردن عدد مرز است.

```
test :  
iflight. 1 > 500 then  
led Red Left on  
endif  
led Red Left off  
goto test
```

برنامه پیدا کردن عدد مرز در سنسور نوری

۲-۶- نوشتن برنامه ربات تعقیب نور

برای نوشتن برنامه ربات تعقیب نور لازم است ۲ نکته را در نظر داشته باشید:

اول اینکه مراقب باشید تا پایه های سنسور نوری به هم نچسبیده باشند (می توانید از چسب برق برای جلوگیری از اتصال پایه ها استفاده کنید) و دوم اینکه سنسورها هر یک به درستی در جهت های چپ و راست قرار گرفته باشند.

برای نوشتن برنامه ۳ حالت را نظر می گیریم:

حالت اول: نور از سمت چپ به سنسور شماره ۱ بتابد.

حالت دوم: نور از سمت راست به سنسور شماره ۲ بتابد.

حالت سوم: منبع نور جلوی ربات قرار گرفته باشد و هم به سنسور شماره ۱ و هم ۲ بتابد.

نکته: شماره گذاری ها در ماژول های نادکو از چپ به راست می باشد.



```
nadco :
iflight. 1 > 500 then
iflight. 2 < 500 then
motor Right + 100
motor Left -100
endif
endif
iflight. 1 < 500 then
iflight. 2 > 500 then
motor Left + 100
motor Right -100
endif
endif
iflight. 1 > 500 then
iflight. 2 > 500 then
motor Left + 100
motor Right + 100
endif
endif
goto nadco
```

Selected line = 20 Total line = 20

Motor: Right -100

Control panel icons: Home, Back, Forward, Stop, Play, Run, Connect, Help.

Robot hand icon: Stop, Play, Run, Connect, Help.

