

گزارش پروژه ساختار و زبان کامپیوتر

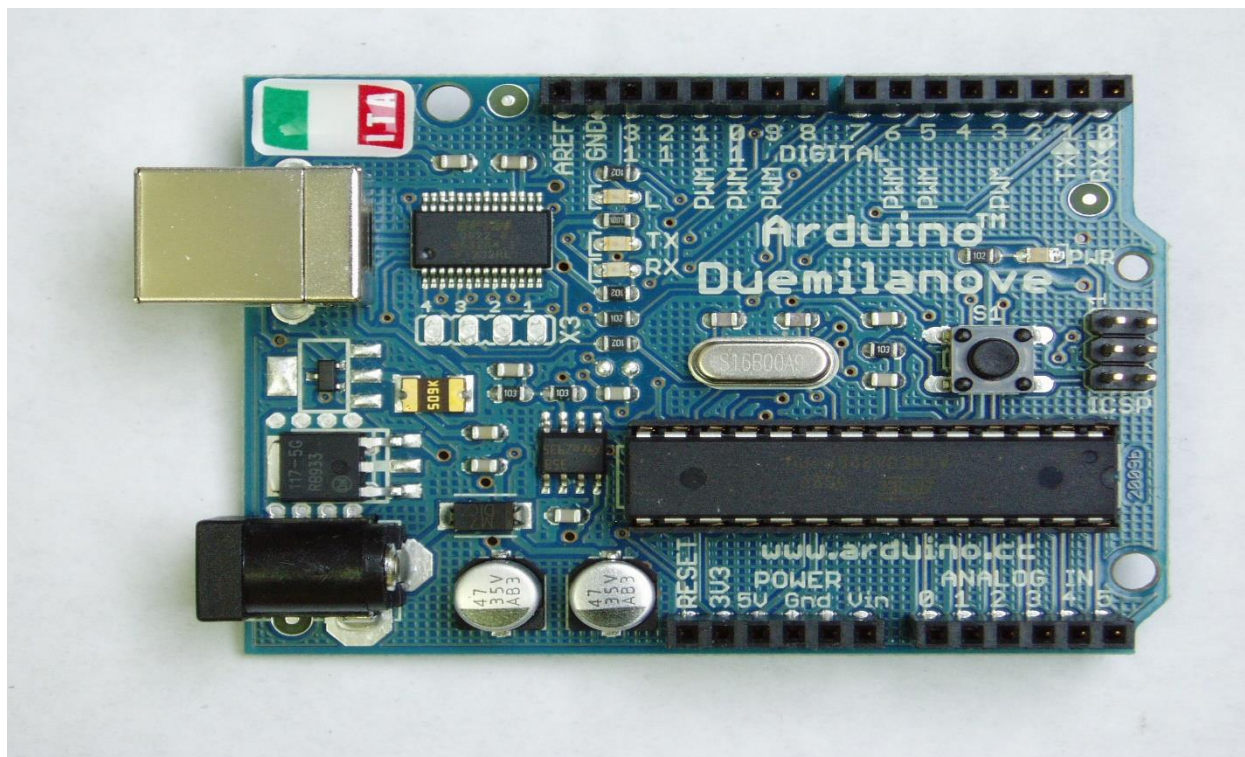
اشکان سلیمانی ، سروش باسلی زاده

پروژه اول: بررسی ماژول joystick با استفاده از برد Arduino

- معرفی قطعات سخت افزاری به کار رفته:

- برد آردواینو

یک میکروکنترلر تک بردی است که به منظور تولید راحت تر برنامه هایی که با اشیاء یا محیط تعامل داشته باشند طراحی شده است. سخت افزار آن شامل یک برد سخت افزار متن باز که حول میکروکنترلر ای وی آر اتمل یا یک آرم اتمل طراحی شده است. بعنوان مثال مدل Arduino UNO Rev3 دارای رابط یواس بی، ۶ پین ورودی آنالوگ و همچنین ۱۴ پین ورودی/خروجی دیجیتال است که اجازه اتصال بردهای توسعه مختلفی را فراهم می آورد. هرچند در مدلهای دیگر تعداد این درگاه های آنالوگ و دیجیتال متفاوت (بیشتر یا کمتر) است. آردوینو می تواند جهت ایجاد اشیای تعاملی، گرفتن ورودی از تعداد زیادی سویچ و حسگر، و کنترل تنوعی از لامپ ها، موتور ها و سایر خروجی های فیزیکی به کار گرفته شود. آردوینو در سال ۲۰۰۵ به منظور ایجاد راهی ارزان و ساده برای برنامه نویسی اشیای تعاملی ایجاد شد. آردوینو به همراه یک محیط یکپارچه توسعه نرم افزار (IDE) ساده ارائه می شود که در رایانه های عادی قابل اجرا است که اجازه برنامه نویسی به کمک سی یا سی++ را برای آردوینو می دهد. آردوینو می تواند پارامترهایی مانند نور محیط، کلیدها یا حتی یک ایمیل را به عنوان ورودی دریافت نماید و بعد از پردازش، خروجی هایی مانند روشن کردن یک وسیله برقی، تغییر رنگ LED ها یا ارسال یک ایمیل یا نظیر آن را ارائه دهد.

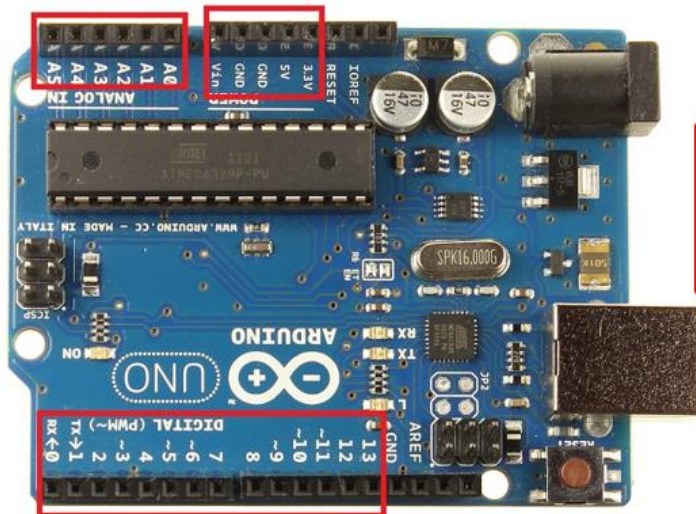


• پین های خروجی و ورودی برد آردواینو

پین های آردواینو به سه بخش تقسیم می شوند:

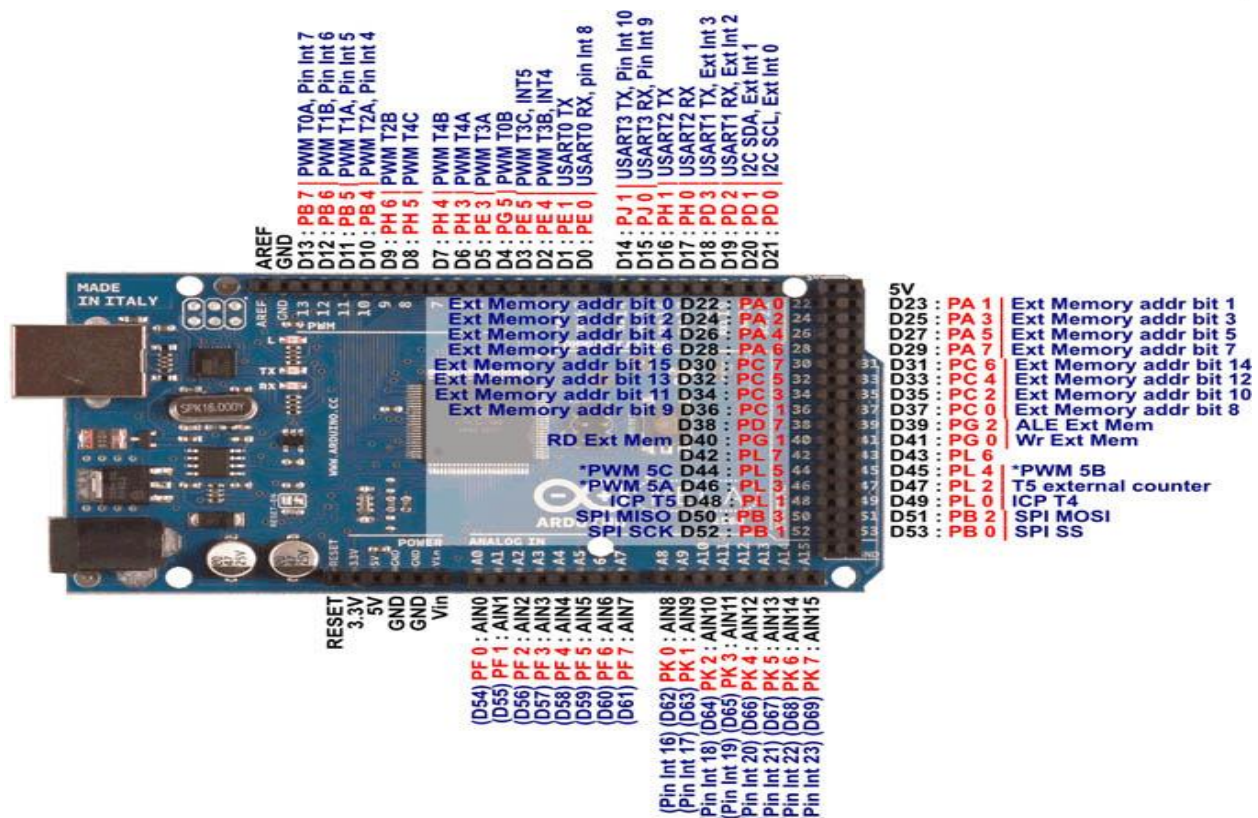
۱. پین های دیجیتال
 ۲. پین های آنالوگ
 ۳. پین های تغذیه
۱. پین های دیجیتال : تعداد این پین ها ۱۴ تاست که از ۰ تا ۱۳ شماره گذاری شده اند و هم به عنوان ورودی و هم به عنوان خروجی قابل استفاده اند.
۲. پین های آنالوگ:
- پین های آنالوگ ورودی : A0 تا A5 و مقدار آن ها بین ۰ تا ۱۰۲۳ خواهد بود.
- پین های آنالوگ خروجی : در واقع ۶ تا از همان پایه های دیجیتال است که (۳، ۵، ۶، ۹، ۱۱) است.
۳. پایه های تغذیه : می توان با استفاده از پورت USB و یا جک منبع نیرو را به آن متصل کرد.

پایه های تغذیه پایه های آنالوگ ورودی



نکته: پایه های شماره ۳ و ۵ و ۶ و ۹ و ۱۰ و ۱۱ پایه های PWM هستند که به پایه های آنالوگ خروجی معروف می باشند.

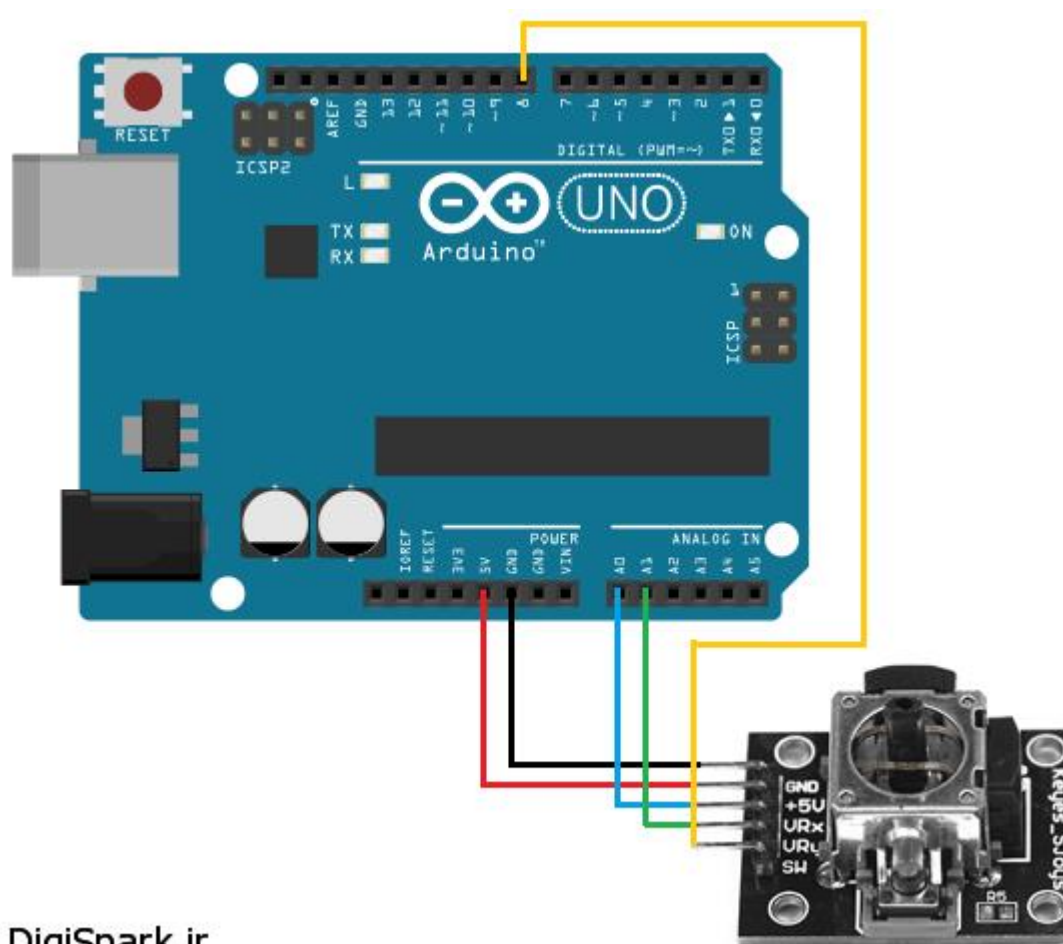
پایه های دیجیتال



• ماژول JoyStick

اهرمی متشکل شده از دسته‌ای است که در انتهای خود بر حول یک مقطه حرکت می‌کند و اطلاعات زاویه خود را به صورت دو بعدی یا سه بعدی به رایانه و یا ریزپردازنده می‌فرستد.

این ماژول یک ماژول بسیار ساده با خروجی آنالوگ و قابلیت اندازه‌گیری مختصات اهرم جوی استیک در راستای دو محور X و Y می‌باشد. این جوی استیک قابلیت حرکت در تمامی زوایا را داشته و همچنین دارای یک کلید فشاری نیز می‌باشد. از این جوی استیک می‌توان در ساخت انواع کنترلرهای ربات و کوادکوپتر و... استفاده کرد. بر روی این ماژول دو عدد پتانسیومتر ۱۰ کلیواهم جهت اندازه‌گیری حرکت اهرم جوی استیک در راستای دو محور X و Y وجود دارد. مکانیزم کار به این صورت است که با تغییر وضعیت اهرم میزان ولتاژ خروجی ماژول تغییر می‌کند و ما از طریق ADC متوجه می‌شویم که ماژول در چه وضعیتی قرار دارد



پین GND و +5V که مربوط به تغذیه ماژول می‌باشد و باید به ترتیب به GND آردوینو و مثبت ۵ ولت DC متصل شود.

پین SW که مربوط به کلید فشاری قرار گرفته بر روی جوی استیک می باشد. در حالت عادی خروجی این پین +۵ ولت DC است و در صورت فشار دادن آن ولتاژ خروجی آن به ۰ ولت تغییر می کند. (این پین به پین شماره ۸ آردوینو متصل می شود)

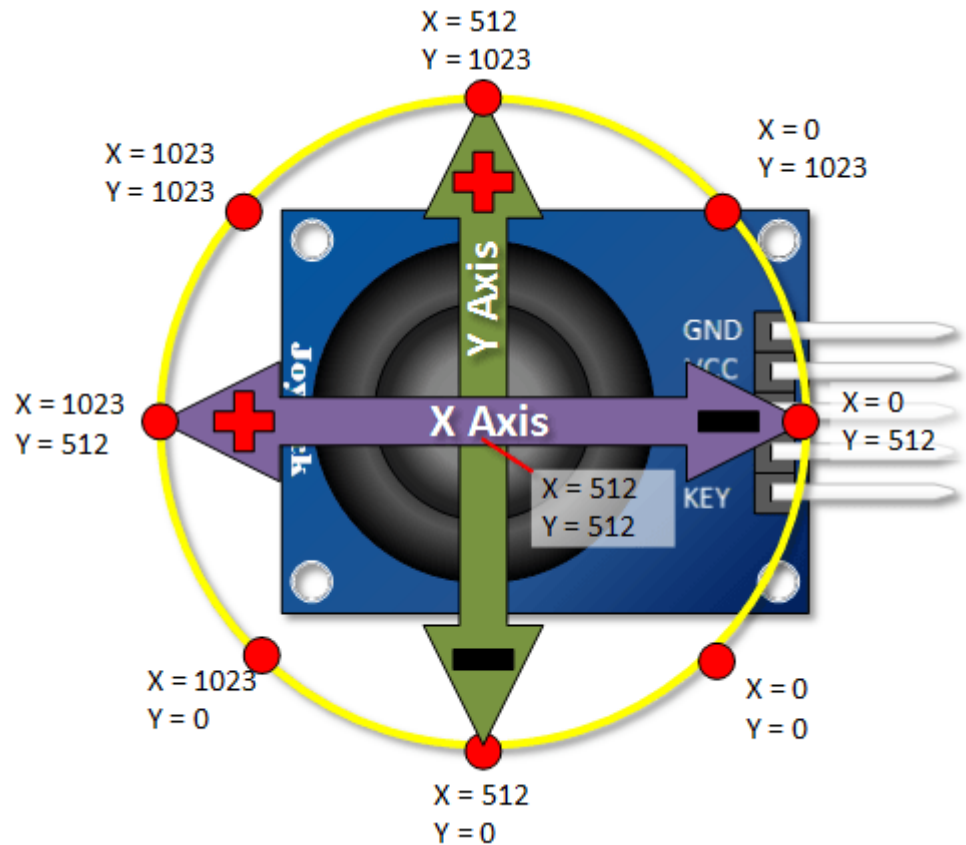
پین VRx: این پین مربوط به تغییرات اهرم جوی استیک در راستای محور X (محور افقی) می باشد. در حالت عادی ولتاژ خروجی این پین برابر ۲,۵ ولت می باشد و برحسب تغییر وضعیت اهرم از راست به چپ به تریب ولتاژ خروجی این پین میان دو بازه ۵ الی ۰ ولت تغییر می کند. (این پین به پین A0 برد آردوینو متصل می شود)

پین VRy: این پین مربوط به تغییرات اهرم جوی استیک در راستای محور Y (محور عمودی) می باشد. در حالت عادی ولتاژ خروجی این پین برابر ۲,۵ ولت می باشد و برحسب تغییر وضعیت اهرم از بالا به پایین به تریب ولتاژ خروجی این پین میان دو بازه ۵ الی ۰ ولت تغییر می کند. (این پین به پین A1 برد آردوینو متصل می شود)

برای برنامه نویسی این ماژول ما باید به وسیله ی مبدل آنالوگ به دیجیتال (ADC) ولتاژهای خروجی پین های VRx و VRy را اندازه گیری کنیم. همانطور که گفته شد ولتاژهای این دو پین میان بازه ی ۰ الی ۵ ولت تغییر می کند پس هنگامی که این ولتاژ آنالوگ توسط ADC به مقادیر دیجیتال تبدیل می شود، مقادیر بدست آمده بین ۰ الی ۱۰۲۳ می باشد.

VRy: در هنگام تبدیل به مقادیر دیجیتال، در بالاترین نقطه دارای مقدار ۱۰۲۳ و در پایین ترین نقطه برابر ۰ می باشد.

VRx: در هنگام تبدیل به مقادیر دیجیتال، در راست ترین نقطه دارای مقدار ۱۰۲۳ و در چپ ترین نقطه برابر ۰ می باشد.



• سایر

LED ها (برای هر وضعیت)، سیم های اتصال، مقاومت ها و کابل اتصال به کامپیوتر برد آردوینو.

• شرح برنامه:

برای نوشتن برنامه ای برای برد آردوینو از زبان C، C++ و یا اسمبلی AVR می توان استفاده کرد. برای محیط برنامه نویسی از Arduino Genuino استفاده می کنیم و با وصل کردن کامپیوتر به برد کد را در آن آپلود کرده و اجرا می شود.

برای نوشتن برنامه دو تابع `setup()` و `loop()` تعریف می کنیم تابع اول یک بار اجرا شده و تابع دوم مرتباً تکرار می شود.

ابتدا پین های Led های مطلوب را مشخص کرده

```
int ledY = 11;  
int ledX = 10;  
int ledXY = 3;  
int led_XY = 9;  
int ledPressed = 4;  
int isKeyPressed = 8;
```

سپس پین های ورودی joystick را مشخص می کنیم:

```
int input_X = 0;  
int input_Y = 1;
```

سپس یک آرایه و تعدادی متغیر برای رفع noise می گیریم.

روند کار به این صورت است که در ابتدا داده را از joystick می گیریم و در قدیمی ترین خانه آرایه ذخیره می کنیم هر بار برای قضاوت از محل مازول از میانگین این ده خانه آرایه بهره می بریم. برای قضاوت درباره محل حضور مازول هر بار معادله خط مربوطه را با مقداری error در نظر می گیریم که برای حضور بر روی دو محور errorAxis و برای حضور بر روی دو محور کج از error استفاده می کنیم. برای نمونه :

(errorAxis - مختصات مبدا > avgX && errorAxis + مختصات مبدا < avgX)

نشان دهنده حضور مازول بر روی محور X است.

برای خواندن مقدار ورودی های آنالوگ از joystick از دستور analogRead(input_X) و برای نوشتن مقدار خروجی بر روی خروجی های آنالوگ از analogWrite(ledX, 0) استفاده می کنیم (مقدار نور باید بین ۰ تا ۲۵۵ باشد)

هم چنین برای خواندن مقدار دیجیتال از joystick از دستور digitalRead(isKeyPressed) استفاده می کنیم که دو مقدار LOW یا HIGH خواهد داشت.

برای درخاطر نگه داشتن وضع سابق فشار داده شدن دکمه SW از یک int استفاده می کنیم که ۰ یا ۱ بودن آن که در کدام وضعیت فعلی هستیم را نشان خواهد داد.

برای اجرای برنامه آن را کامپایل کرده پورت خروجی را مشخص کرده و آپلود می کنیم.

برای بخش اسمبلی باید از اسمبلی inline استفاده کرد که بر دو نوع است :

۱- Volatile

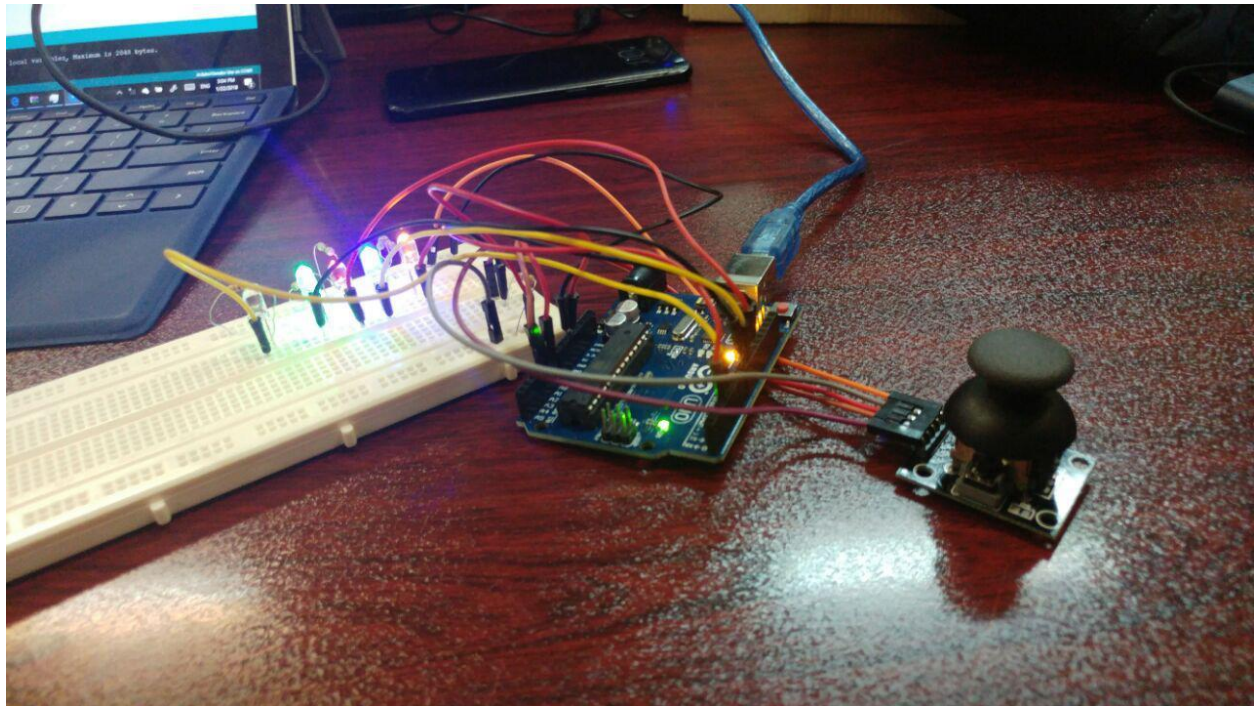
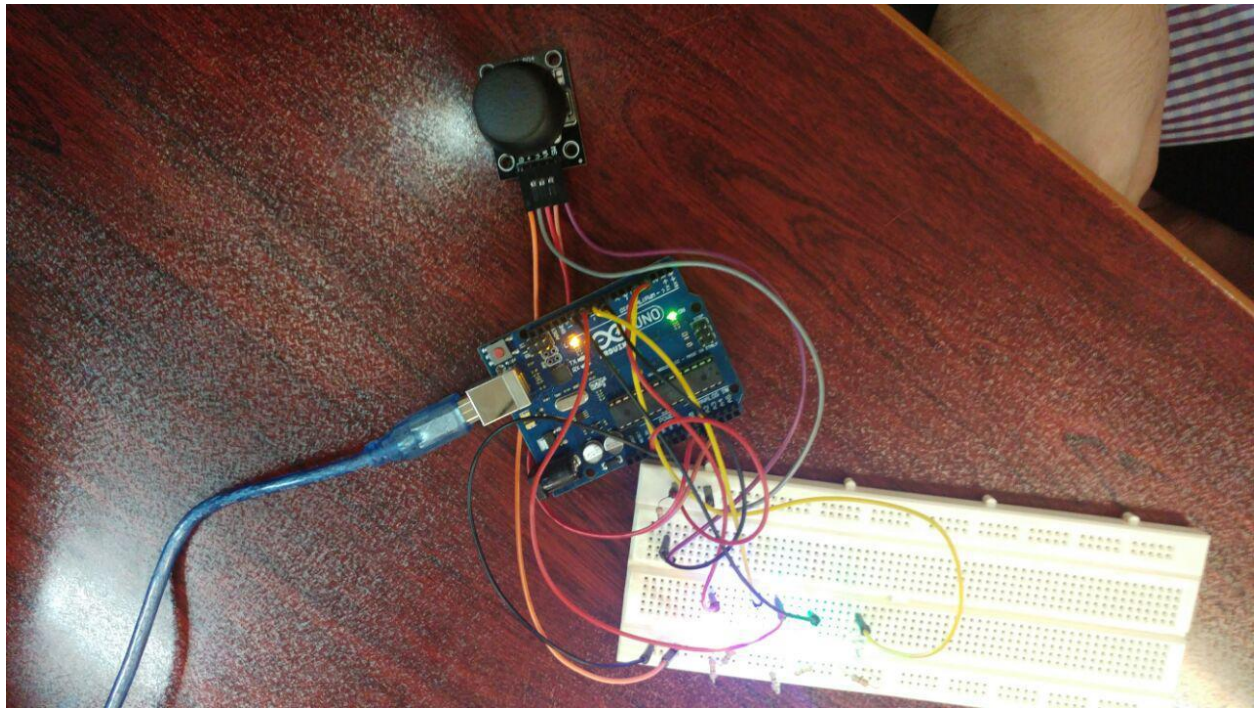
۲- Optimized

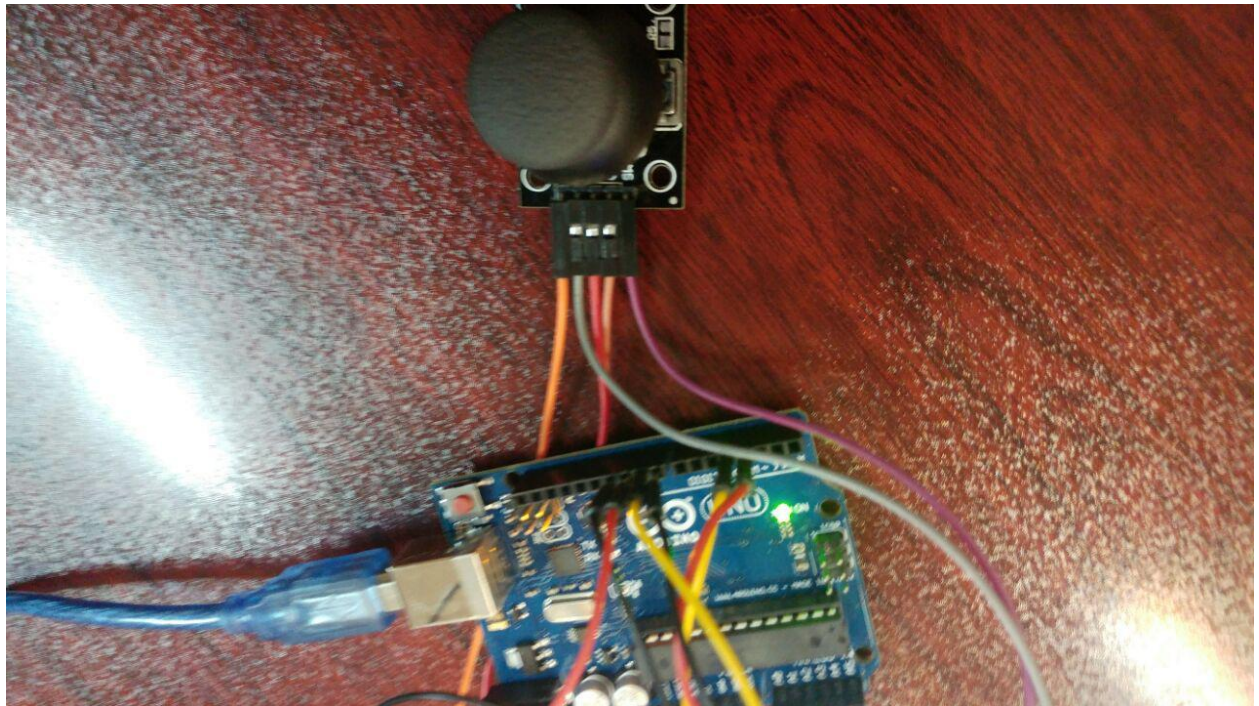
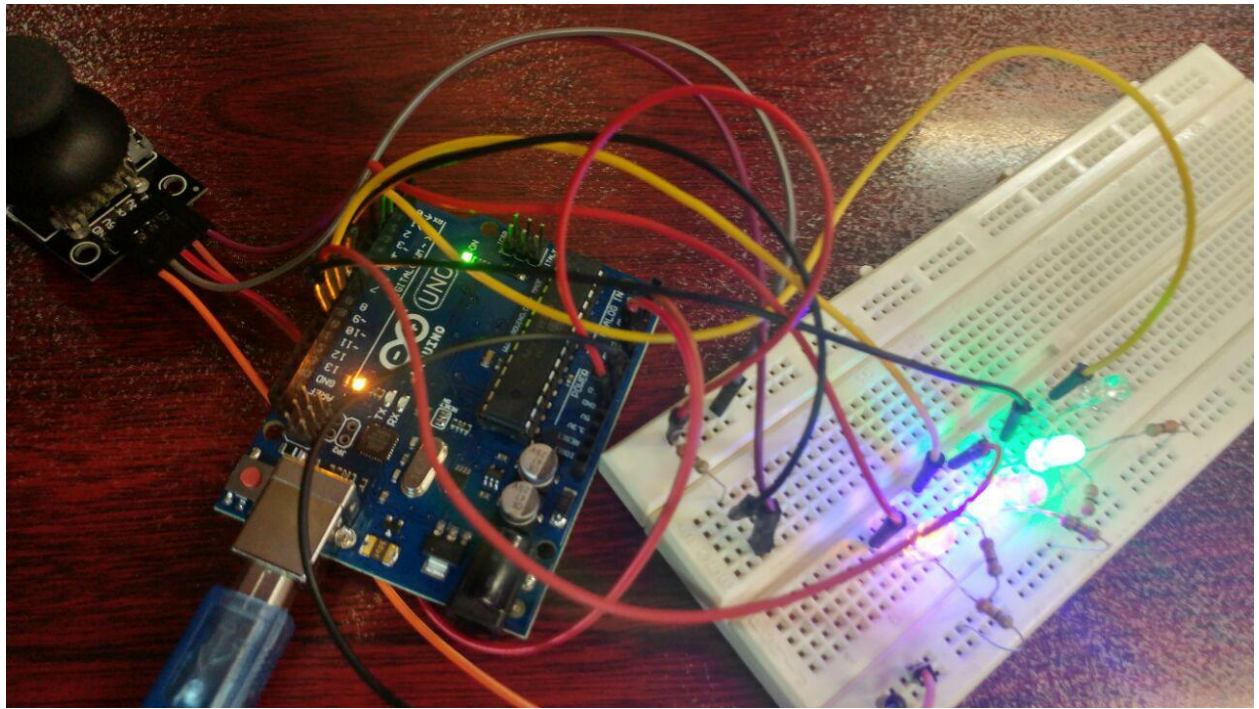
در این جا چون نمی خواهیم بهینه سازی ناخواسته ای از سوی کامپایلر صورت پذیرد از اولی استفاده می کنیم. در اسمبلی گفته شده باید ابتدا مشخص کنیم متغیر مورد نظر ما به چه دسته ای از ثبات ها نسبت داده شود و با این ثبات بتوان نوشتن و خواندن یا صرفا دومی را پیاده کرد. ثبات های مخصوص خود را رزرو باید کنیم (جداگانه).

شمارنده خود را ابتدا با یک حلقه for اسمبلی (برنج و ...) باقیمانده با ۸ می گیریم سپس مقادیر آرایه را یک به یک افزوده و میانگین این ۸ داده را می گیریم. (این کار به این صورت است که هر بار باید carry را به یک رجیستر دیگر اضافه کنیم و در نهایت با استفاده از شیفت ها آن را هندل نمائیم)

• بستن مدار :

برای بستن مدار led ها را با یک مقاومت به پایه اختصاص یافته و و به زمین وصل می کنیم (بدون مقاومت led ها می سوزند)، پایه SW را با pullUp کردن به پایه مخصوص تعریف شده آن وصل می کنیم ورودی های ماژول را هم به ۰ و ۱ یا همان A0 و A1 وصل می کنیم. ماژول joystick را جز برای SW که pull up می شود به مقاومت متصل نمی کنیم زیرا اندازه گیری آن براساس جریان و مقاومت خود آن است و این کار روند کاری آن را نادقیق می سازد.





همکاری :

اکثریت قریب به اتفاق پروژه با همکاری همراهی و هم فکری همدیگر و در کنار هم زده شده است اما به طور ریزتر بخش های پیاده سازی کدهای C ارتباط با led ها و ... با فروش باسلی زاده و بخش های assembly با اشکان سلیمانی بوده است.

چالش ها:

به نظر می رسد زدن کد اسمبلی این پروژه توجیهی ندارد سختی ها و ایرادات بسیاری به این موضوع وارد است این اسمبلی بسیار متفاوت تر از اسمبلی های مطرح شده در کلاس درس است این اسمبلی تقسیم (که احتمالا در سایر اسمبلی های آموخته شده در کلاس درس) از ابتدائی ترین دستورها بوده است را پیاده سازی نمی کند، این اسمبلی رجیسترهای ۸ بیتی دارد و فرآیند انجام عملیات بر روی اعداد با داده های بزرگتر بسیار سخت و طاقت فرسا است به دلیل آن چه که به نظر می رسد چندان community کثیری در دنیا از این اسمبلی استفاده نمی کنند برای همین tutorial ها، منابع و ... در این مورد بسیار اندک است و یافتن چنین مواردی بسیار سخت است.

زمان پروژه :

این پروژه بدون بخش اسمبلی در یک روز تمام خواهد شد و با آن بسیار سنگین و زمان بر است.

مراجع :

[JoyStick](#)

[آردوینو](#)