

28.08.1399

نوشاد شکوه فر

سنسور Gyro

18.11.2020

Noshad Shokouhfar

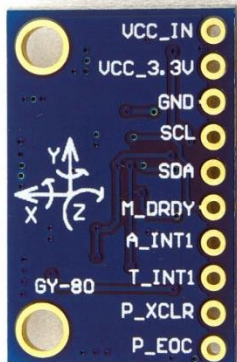
Gyro Sensor

### فهرست :

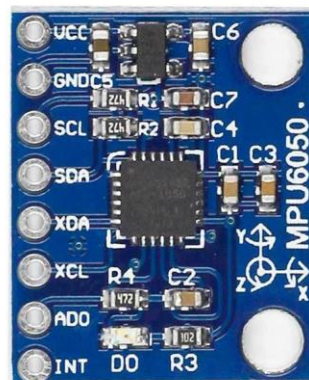
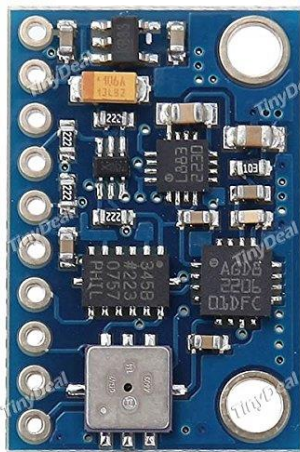
1. توضیحات
2. IMU
3. MEMS
4. GY-521
5. شتاب سنج | Accelerometer
6. ژایروسکوپ | Gyroscope
7. سنسورهای خارجی | External Sensors
8. پردازشگر حرکت دیجیتال | Digital Motion Processor
9. راه اندازی ماژول GY-521 به کمک برد آردینو
  - پایه ها
  - اتصال به برد
  - پیدا کردن آدرس ماژول
  - کالیبره کردن ماژول
  - راه اندازی سنسور
  - خروجی ماژول با کمک Processing
10. منابع

### 1. توضیحات

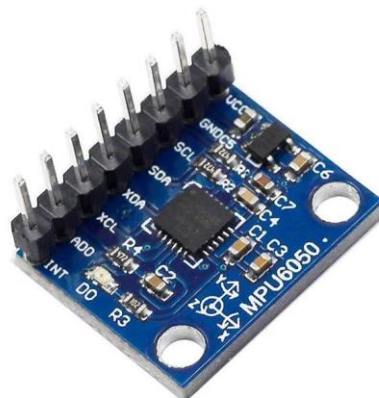
سنسور ژایرو (Gyro) برای محاسبه شتاب ، موقعیت ، زاویه و ... ربات بکار می رود .  
دو نوع از ماژول های آن در تصویر 1 آمده :



GY-80

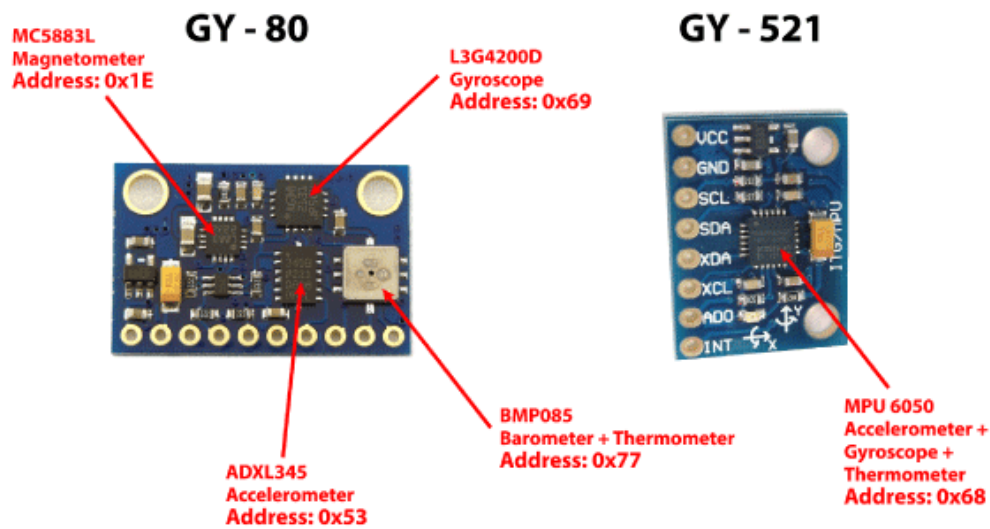


GY-521



تصویر 1

همینطور که در تصویر 1 و 2 می بینید ، این ماژول با کمک تراشه هایی که روی آن قرار گرفته شتاب و موقعیت ربات را محاسبه میکند.



تصویر 2

تفاوت اصلی این دو ماژول تو اینه که روی GY-80 از چند تراشه برای محاسبه شتاب ، موقعیت زاویه ای ، دمای محیط و ... استفاده شده ، در حالیکه ماژول GY-521 با کمک یک تراشه تمام این کمیت ها رو محاسبه میکنه . در نتیجه از این به بعد فقط درمورد ماژول GY-521 و تراشه ای که روش قرار گرفته صحبت میکنیم .

خوبه قبل از اینکه وارد بحث راه اندازی و چگونگی کارکرد سنسور بشیم ، یه سری توضیحات پایه رو بدونیم . مثلا :

### 2. IMU

تراشه ای که روی ماژول مورد نظر ما قرار گرفته (طبق تصویر 2) ، بنام MPU-6050 IMU شناخته شده . در حقیقت این تراشه جزو دسته IMU ها هست . IMU مخفف Inertial Measurement Units و به معنی واحدهای اندازه گیری اینرسی هست . در واقع ، هر جا لازم باشه سرعت ، شتاب ، اینرسی و هر کمیتی مربوط به موقعیت وسایل یا ربات محاسبه بشه ، کاربرد دارن .

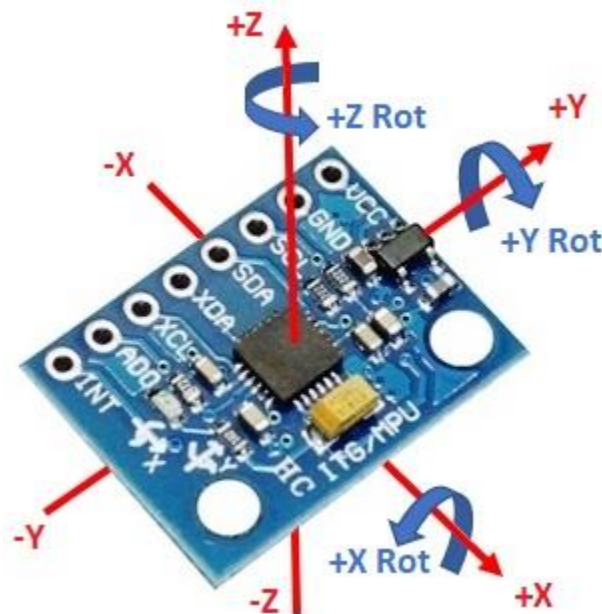
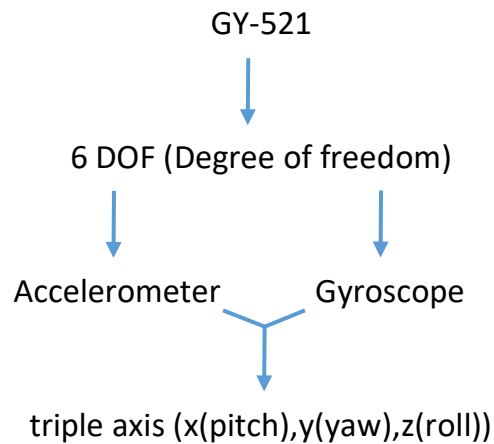
### 3. MEMS

اگر بخوایم خیلی ساده بگیم ، یکی از زیرمجموعه های دسته IMU ، همین MEMS ها هستند . MEMS از مخفف کلمات Micro-Electro-Mechanical Systems گرفته شده که در ابعاد بین 0.1 میلی متر تا 0.001 میلی متر داخل تراشه ها قرار میگیرن و از جنس های مختلفی مثل سیلیکون ، پلیمر ، سرامیک یا حتی متال ساخته میشن و برای تکمیل سنسور ، یک میکروکنترلر هم دارن .

تا همینجا اطلاعات پایه کافیه ، بر می گردیم به خود سنسور ☺

### 4. GY-521

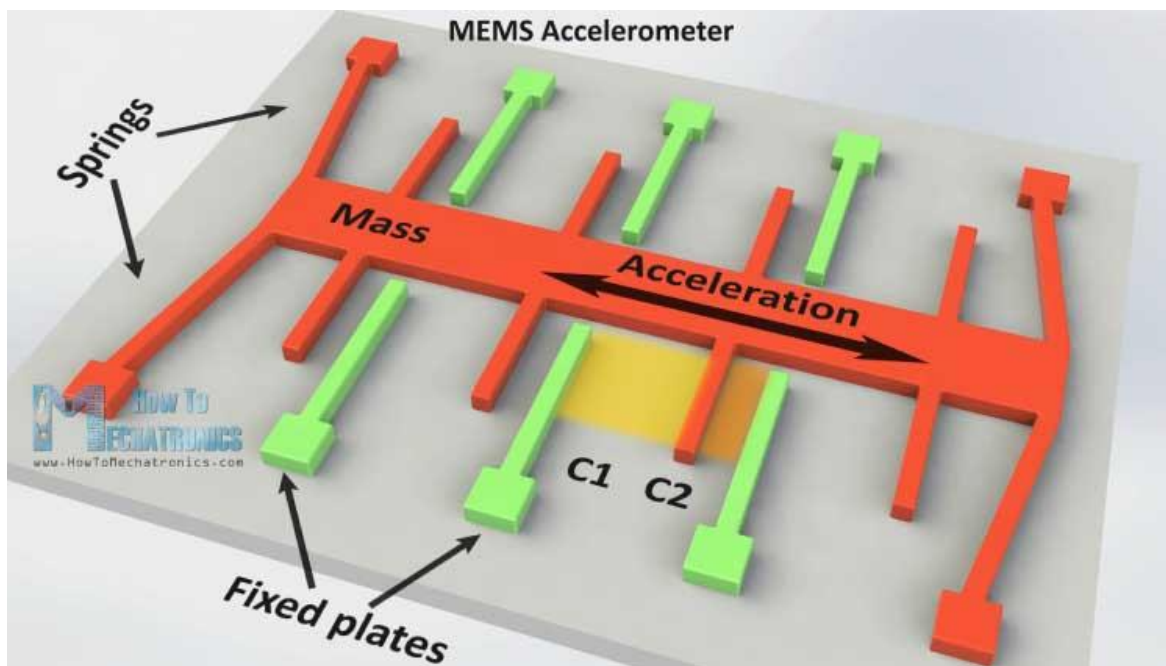
این ماژول 6 درجه آزادی دارد؛ به عبارتی می‌گیم ((6-DOF (Degree Of Freedom)). دلیل اینکه 6 درجه آزادی دارد، اینست که نه فقط سنسور شتاب سنج، بلکه سنسور ژایروسکوپ، هم حول 3 محور (pitch), y(yaw), z(roll) تغییرات رو محاسبه می‌کنند.



تصویر 3

### 5. شتاب سنج | Accelerometer

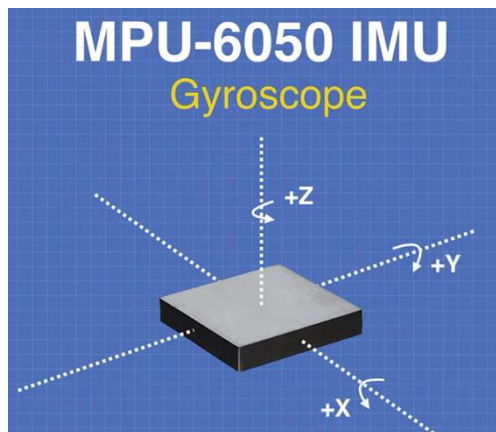
همونطور که بالا توضیح دادیم ، شتاب ربات حول سه محور  $x, y, z$  محاسبه میشه . برای اینکه بفهمیم سنسور چطور شتاب رو حول این سه محور محاسبه میکنه ، به تصویر 4 توجه کنید. این تصویر ، داخل سنسور شتاب سنج رو نشون میده که یه توده / جسم یا Mass (نارنجی) داخلش قرار داره و یه سری صفحات خازنی (سبز) هم دیده میشه . زمانی که شتابی به ربات ما داده بشه ، این توده ، در همون جهت جابجا میشه و ظرفیت بین صفحات خازنی و توده تغییر میکنن . سنسور با محاسبه مقدار تغییر ظرفیت صفحات خازنی و پردازش آن ((ADC (Analog to Digital Converter)) ، شتاب ربات رو برامون چاپ میکنه .



تصویر 4

### 6. ژایروسکوپ | Gyroscope

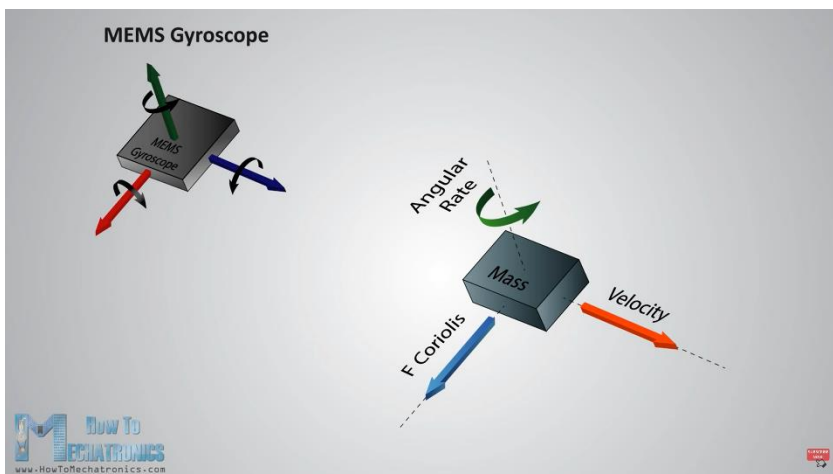
این سنسور هم طبق توضیحات بالا و تصویر 5 ، موقعیت زاویه ای ربات رو محاسبه میکنه .



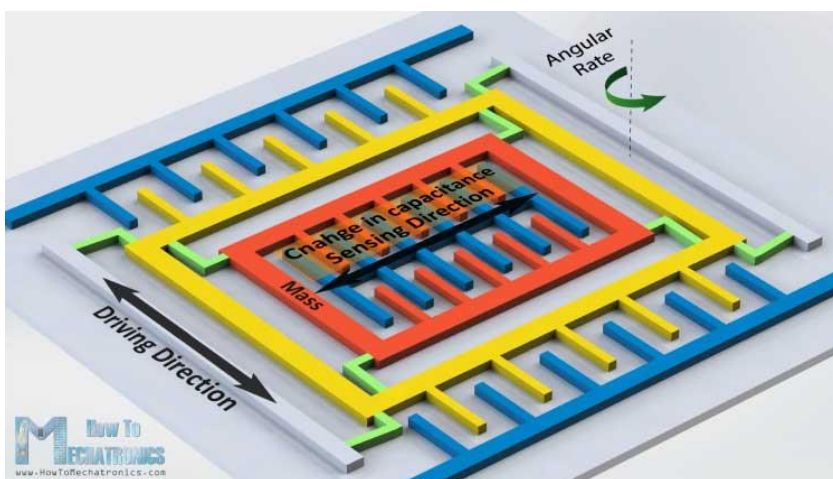
تصویر 5



به تصویر 6 و 7 توجه کنید. زمانی که به جسم، نرخ زاویه ای داده بشه، به جسم نیرو وارد میشه و سپس توده ای که درون سنسور قرار گرفته، عمود بر جهت زاویه وارد شده حرکت میکند. طرز محاسبه زاویه ربات هم مثل شتاب سنجه. به این صورت که با محاسبه تغییر ظرفیت بین صفحات خازنی و توده و پردازش آن، خروجی به عنوان زاویه ی ربات برای ما نمایش داده میشه. این نکته رو هم خوبه بدونید که روی این تراشه، در اصل برای محاسبه زاویه حول هر محور، یک سنسور داریم یا به عبارتی سه سنسور ژایرو داریم ☺



تصویر 6



تصویر 7

### 7. سنسورهای خارجی | External Sensors

یکی از قابلیت های تراشه MPU6050 اینه که میشه سنسورهای دیگه ای هم بهش اضافه کرد . یکی از متداول ترین سنسورها Magnetometer هست که میدان مغناطیسی زمین رو حول سه محور تعیین میکنه . در نتیجه اگر این سنسور هم اضافه کنیم ، ماژول ما از 6 درجه آزادی به 9 درجه آزادی تغییر می کنه .

یادمون نره که خود تراشه ، یه سنسور هم برای محاسبه دمای محیط (temperature sensor) داره .

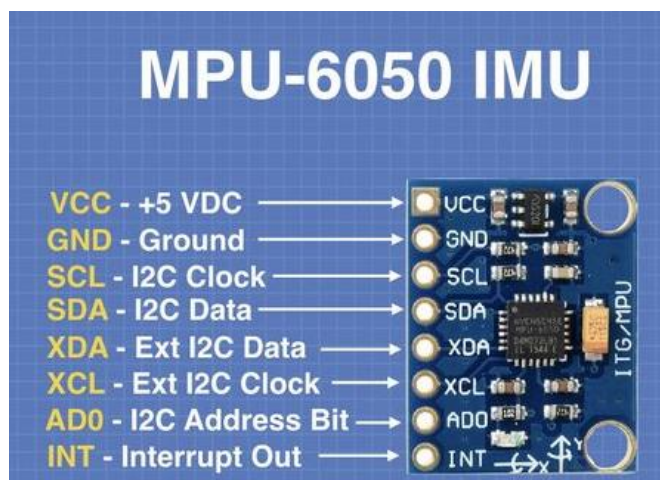
### 8. پردازشگر حرکت دیجیتال | Digital Motion Processor

یکی دیگه از قابلیت های این سنسور اینه که میتونیم به صورت سه بعدی با کمک برنامه Processing Software حرکاتش رو شبیه سازی کنیم . در واقع ، سنسور به شکل یه هواپیما بر روی صفحه مانیتور دیده میشه که ، وقتی در جهت های مختلف قرارش بدیم ، میتونیم حرکاتش رو تحلیل کنیم .

### 9. راه اندازی ماژول Gy-521 با کمک برد آردینو

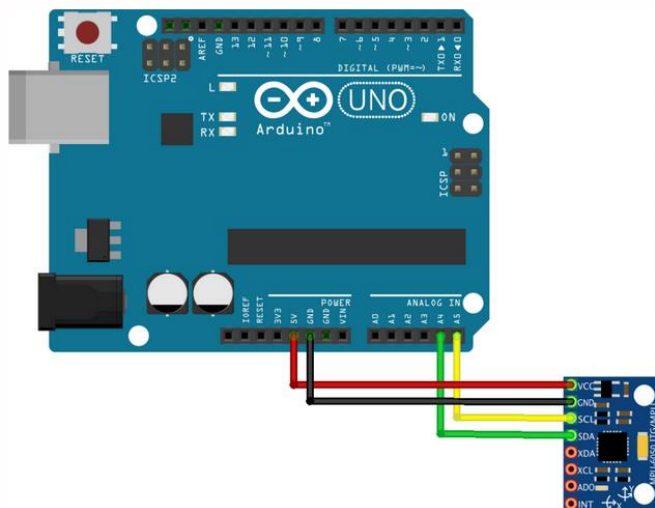
قبل از راه اندازی لازمه بدونید که ، این ماژول با کمک پروتکل I2C (I-squared-C) به آردینو متصل میشه .

— پایه ها :



تصویر 8

– اتصال به برد آردینو :



تصویر 9

Module pins	Arduino Board
VCC	5V
GND	GND
SCL	SCL-A5
SDA	SDA-A4

– پیدا کردن آدرس ماژول :

پایه AD0 آدرس I2C رو کنترل می کنه. اگر به 0 وصل شه ، آدرسش 0X68 و اگر به 1 وصل شه ، آدرسش 0X69 نوشته می شه. اگر هم به جایی وصل نباشه ، آدرسش همون 0X68 هست. برای پیدا کردن آدرس ماژول هایی که با I2C به آردوینو متصل شده اند باید مطابق زیر عمل کنیم :

می دونیم که 127 دستگاه از طریق این پروتکل می تونند به آردوینو متصل بشن؛ پس یک شمارنده میذاریم که تمام این 127 آدرس موجود رو بررسی کنه ؛ بدین صورت که اول با دستور Wire.beginTransmission(address) با آن آدرس ارتباط برقرار میکنه ، حال ارتباط را با دستور Wire.endTransmission قطع میکند. اگر دستگاهی در آن آدرس وجود داشت، خروجی این دستور 0 می شه و در غیر این حالت دستگاهی متصل نبوده یا اروری رخ داده.

کد مربوط به توضیحات بالا ، در زیر آمده :



```
nDevices = 0;
for(address = 1; address < 127; address++ )
{
    // This uses the return value of the Write.endTransmission
    // to see if a device did acknowledge to the address.
    Wire.beginTransmission(address);
    error = Wire.endTransmission();
    if (error == 0) //success
    {
        Serial.print("I2C device found at address 0x");
        if (address<16)
            Serial.print("0");
        Serial.print(address,HEX);
        Serial.println("  !");

        nDevices++;
    }
    else if (error==4)
    {
        Serial.print("Unknow error at address 0x");
        if (address<16)
            Serial.print("0");
        Serial.println(address,HEX);
    }
}
if (nDevices == 0)
    Serial.println("No I2C devices found\n");
else
    Serial.println("done\n");
```

– کالیبره کردن ماژول :

خب برای راه اندازی ماژول ، اول باید یکسری offset ها رو پیدا کنیم و تو کد جاگذاری کنیم. این آفست ها میانگین 1000 دیتا هستند که بعد از محاسبه تو کد اصلی جای داده می شوند. برای بدست آوردن این offset ها باید کد کالیبریشن را اجرا کرد و MPU را به صورت افقی و ثابت قرار داد تا کار خوندن دیتای سنسور ها و میانگین گیری از آنها تمام شود.

بعد از پیدا کردن آدرس ماژول با کد i2cScanner، و اطمینان از اینکه MPU می تونه با میکرو ارتباط برقرار کنه، نوبت به پیدا کردن این offset ها میرسه. پیدا کردن آفست ها ، همون کالیبره کردنه که باعث می شه دیتای خروجی ثابت باشه. ولی قبل از این کار نیازه که دو کتابخانه I2Cdev و MPU6050 را به کتابخانه هامون اضافه کنیم.

برای این کار ، دو فایل با این نام ها رو کپی کنید و در آدرسی که برنامه آردوینو رو نصب کردید ،

(مثلا: D:\Program Files\Arduino\libraries ) برید و دو فایل کتابخانه رو پیست (Paste) کنید.

به این ترتیب این دو کتابخانه ، به کتابخانه های آردوینوی شما اضافه میشن.

حالا برنامه MPU6050\_calibration رو باز کنید و کد رو روی میکرو آپلود کنید. برنامه شروع می کنه و به تعداد 1000 بار دیتا می گیره و از بعد از میانگین گیری ، به ما 6 تا عدد می ده. این 6 عدد ، همونطور که در زیر می بینید ، آفست های ژایرو و شتاب سنج در سه جهت X,Y,Z هستند .

```
mpu.setXGyroOffset(292);
mpu.setYGyroOffset(-20);
mpu.setZGyroOffset(-13);
mpu.setXAccelOffset(-3055);
mpu.setYAccelOffset(257);
mpu.setZAccelOffset(1249);
```

– راه اندازی سنسور :

حالا که کار کالیبره کردنمون تموم شد ، نوبت به کد اصلی می رسه. کد GY-521 رو باز کنید و اعداد حاصل از خروجی کد کالیبره ( 6 عددی که به عنوان آفست گرفتید ) رو تو کد اصلی جای گذاری کنید. وقتی کد رو آپلود کنید ، خروجی که در سه جهت yaw, pitch, roll هست رو میتونید روی سریال مانیتور ببینید.

\*نکته 1 : توجه کنید ، اگر دیدید که دیتا ثابت نیست و در حال تغییر است ، با دوباره کالیبره کردن سنسور، مشکل حل می شود.

\*نکته 2 : مشکل دیگری که ممکنه بهش بر بخورید ، اینه که وقتی که موتور ها روشن می شوند، ژایرو دیگه عدد نمی دهد و میکرو هنگ می کند. این مشکل رو با کوتاه و استیبل (Stable) کردن سیم های I2C و همچنین گذاشتن مقاومت های پول آپ (Pull-up) میشه حل کرد. اگر حل نشد ، با گذاشتن یک خازن 10uF دو سر VCC و GND. ماژول و یا دور کردن ماژول از موتور ها میشه مشکل رو حل کرد.

بعضی مواقع می بینیم که دو پایه I2C با مقاومت Pull-up می شوند، علت چیست؟

مقاومت های پول آپ یک مقدار پیش فرضی رو در خطوط سیگنال قرار می دن. عموماً مقدارشون بین 1K – 10K اهم هست . این مقدار مقاومت ، تضمین می کنه که شما جریان زیادی رو از طریق مقاومت ها به سیستم خود وارد نکنید.

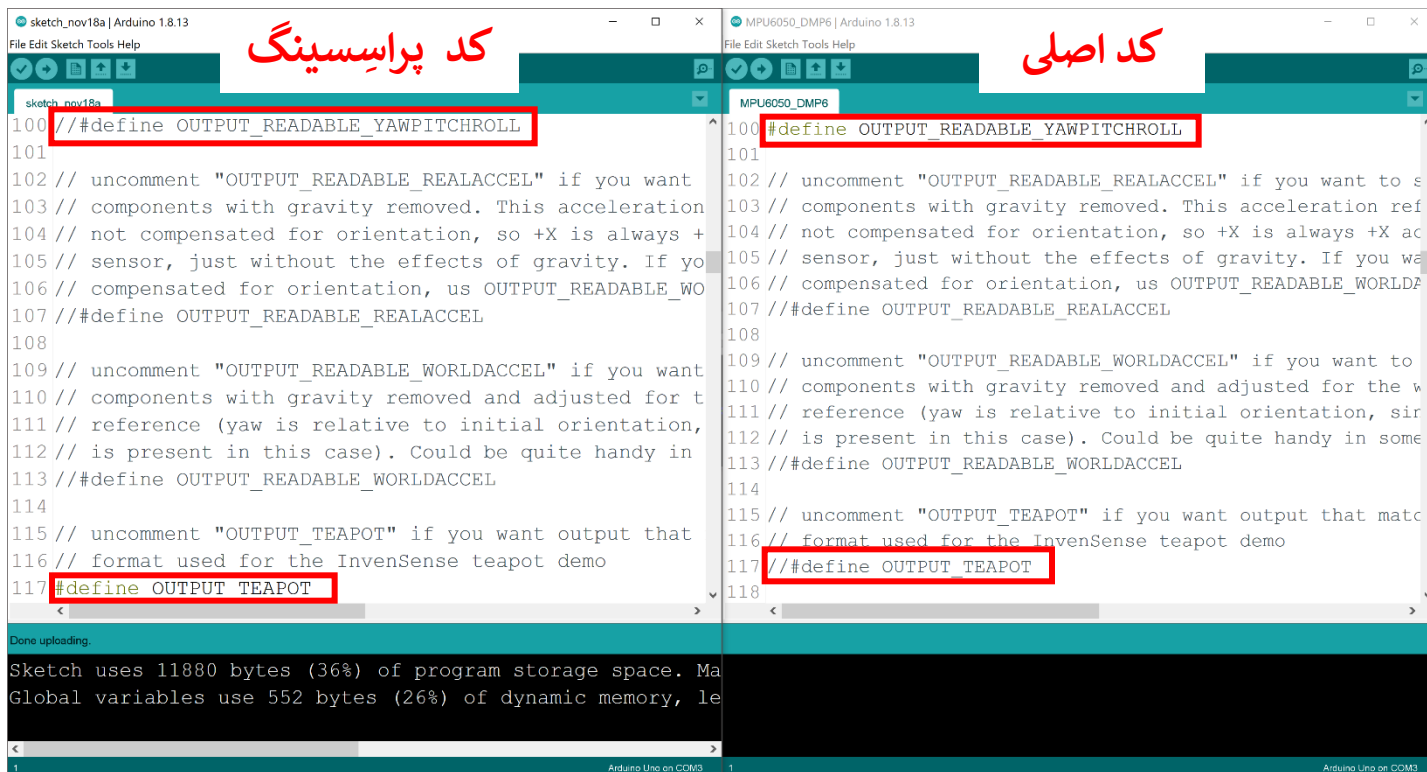
$$(5V V_{dd} / 10000\Omega = 0.5mA \text{ current})$$

تو لینک زیر می تونید بیشتر در این مورد بخونید:

<https://rheingoldheavy.com/i2c-pull-resistors/>

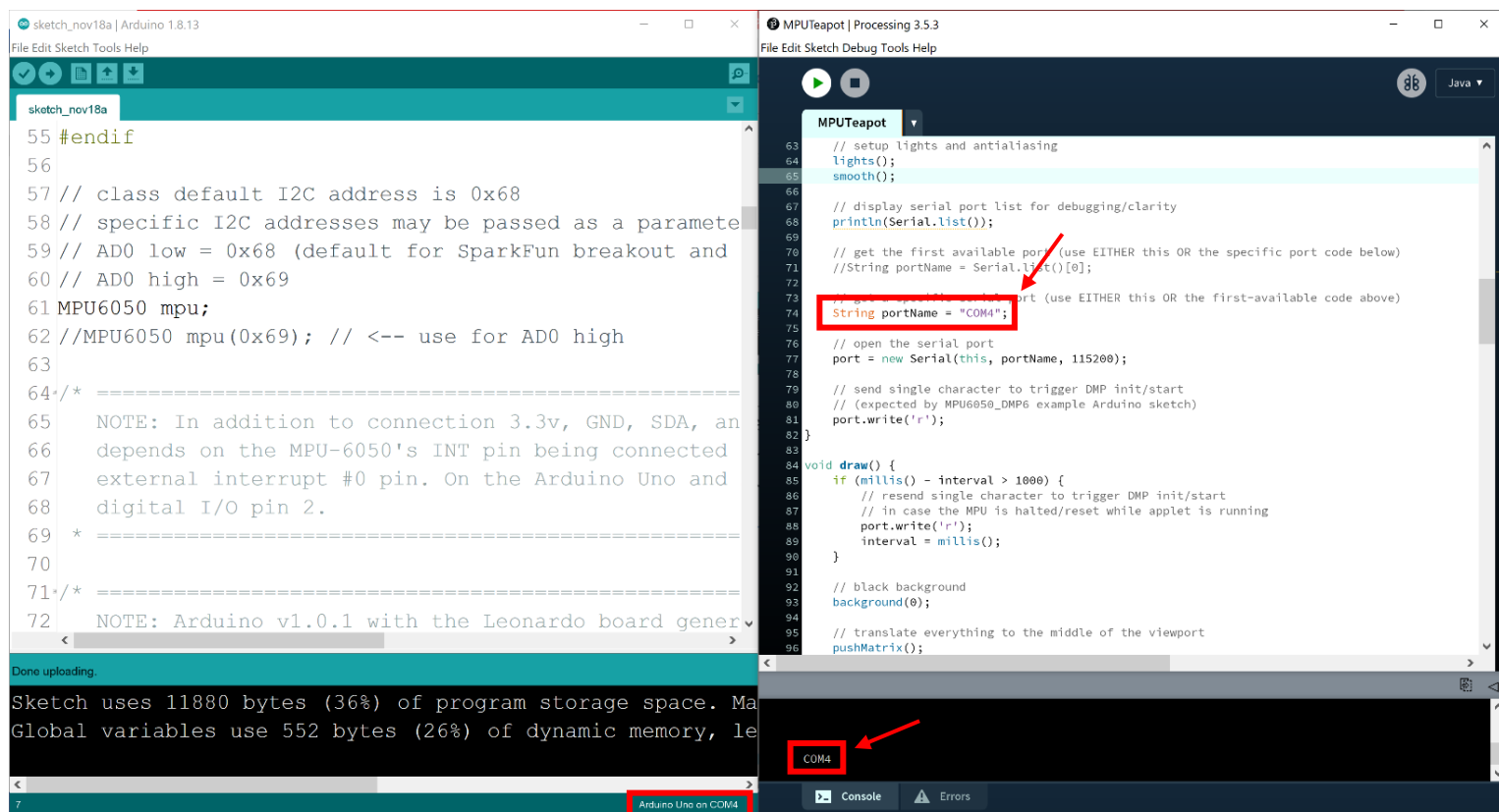
– خروجی ماژول با کمک Processing:

برای اینکه از DMP ماژول استفاده کنیم ، لازمه یه تغییراتی رو کد اصلی بدیم و بعد از آپلود کد جدید رو برد ، کد مربوط به پراسسینگ رو باز و اجرا کنیم . فقط لازمه به چند نکته دقت کنید .



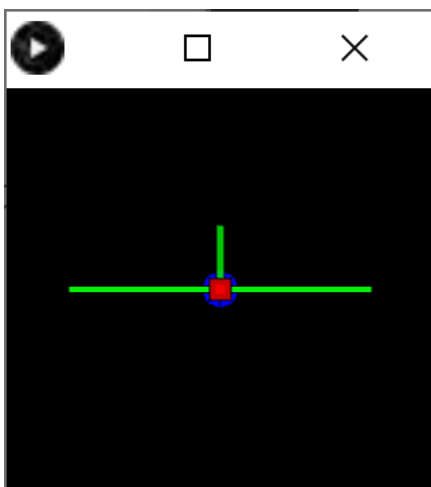
تصویر 10

بعد از اینکه تغییرات لازم رو روی کدتون دادید و آپلودش کردید ، برنامه MPUteapot.pde رو داخل Processing باز کنید و طبق تصویر 11 ، با توجه به COM بردتون ، برنامه رو run کنید .



تصویر 11

خروجی به صورت زیر خواهد بود .



تصویر 12

10. منابع

برای توضیحات بیشتر و دقیق‌تر به زبان انگلیسی، می‌تونید به لینک‌های زیر سر بزنین ☺

1. <https://mjwhite8119.github.io/Robots/mpu6050>
2. <https://dronebotworkshop.com/mpu-6050-level/>
3. <https://howtomechatronics.com/how-it-works/electrical-engineering/mems-accelerometer-gyrocope-magnetometer-arduino/>
4. [https://www.epsondevice.com/en/information/technical\\_info/gyro/#:~:text=Gyro%20sensors%2C%20also%20known%20as,s%20\(degrees%20per%20second\).](https://www.epsondevice.com/en/information/technical_info/gyro/#:~:text=Gyro%20sensors%2C%20also%20known%20as,s%20(degrees%20per%20second).)