ADXL-202

سنسورهای شتاب از مهمترین و پر کاربرد ترین سنسورها هستند.از جمله کاربردهای انها که می تون نام برد :

در رباتیک :برای فهمیدن شیب,شتاب,سرعت و فاصله ربات نسبت به مبدا حرکت.

در لب تاب : برای محافظت از هد هارد دیسک در مواقع حرکت ناگهانی دستگاه

در تراز های دیجیال : برای پیدا کردن زاویه استاتیک

و موارد فراوان دیگر که اهمیت این سنسور رامشخص می سازد.به همین خاطر ما بر ان شدیم تا نحوه به کار گیری یکی از این سنسورها به نام Adxl-202 را بررسی کنیم.

این سنسور دارای برد اندازه گیری ±2 g است, فقط بصورت SMD در بازار موجود می باشد و از قیمتی حدود 200,000 ریال است.

1.طراحی PCB برای adxl-202

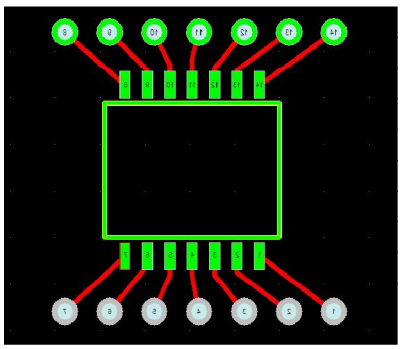
Ic مورد استفاده ما adxl-202 می باشد این ic, smd می باشد و امکان استفاده ی ان روی برد بورد نیست پس بر ان شدیم تا با یک تبدیل بتوانیم اتصال پایه

مجید امانی
amani.majid69@yahoo.com

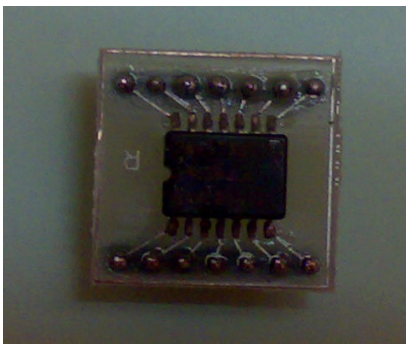
RoboPardaz.com
مرجع دیتاشیت و اطلاعات روباتیک

های ان را به پین هایی انتقال دهیم که قابل نصب روی برد برد برای ازمایش باشند.

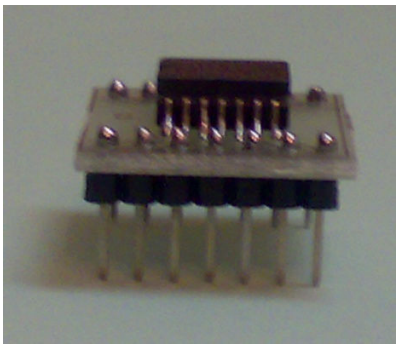
بوسیله نرم افزار قدرتمند Protel DXP توانستیم نقشه زیر را ترسیم کنیم و سپس برد مربوط به ان را اماده کرده و ic و پین های ارتباطی را روی ان لحیم کردیم.



نقشه ی تهیه شده توسط protel DXP



نمای بالای برد تهیه شده



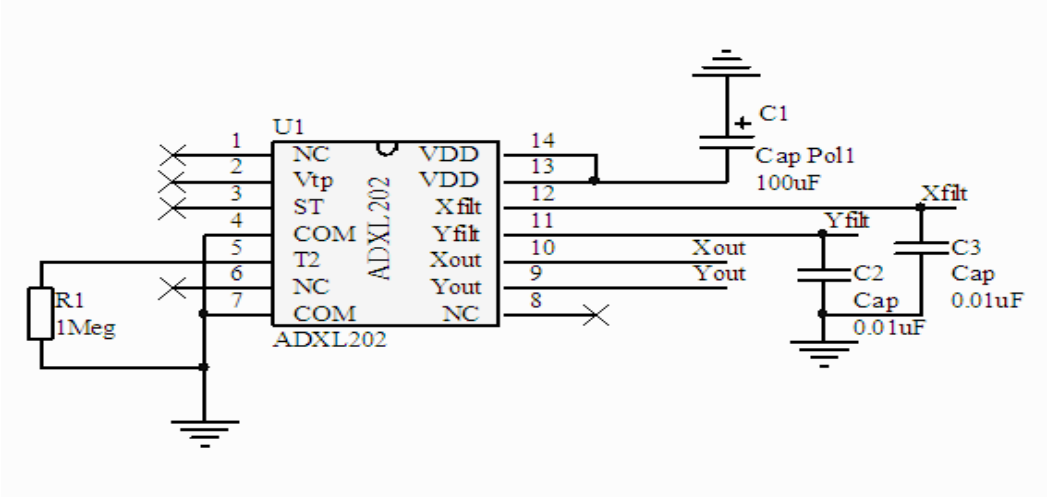
نمای کناری برد تهیه شده

2.بایاس کردن adxl-202

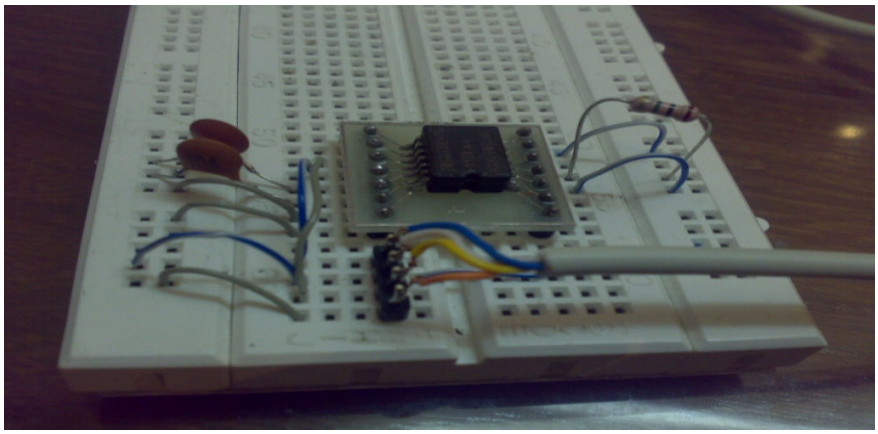
این سنسور نیز مانند تمامی سنسورها نیاز به مداری دارد تا بوسیله ان بتوان سنسور را به کار انداخت و خروجی های ان را مشاهده کرد تا بتوان به مرحله ی کالیبره کردن سنسور رفت.

شماتیک زیر مدار درایو این سنسور را نشان می دهد مقادیر مقاومت ها و خازن ها به روشی که گفته می شود محاسبه می شوند.

مجید امانی
amani.majid69@yahoo.com



نقشه شماتیک مدار درایور سنسور



مدار درایور سنسور که روی برد بورد بسته شده است

X,Yout خروجی های PWM و X,Yfilt خروجی های انالوگ هستند.

خازن c1 برای صافی ولتاژ است که مقدار ان هر چه بزرگتر باشد بهتر است. ما در این طراحی خازن 100uF را انتخاب کردیم.

فرمول محاسبه خازن های c2 و c3 به صورت زیر است :

مجید امانی
amani.majid69@yahoo.com

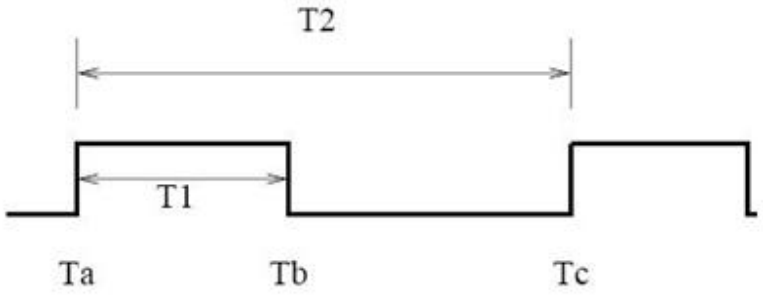
و در این طراحی ما پهنای باند 500Hz را انتخاب کردیم که در این صورت خازن ها برابر 0.01 uF خواهند شد.

مقاومت R1 پریود PWM را تنظیم می کند که مقدار این پریود از رابطه زیر محاسبه می شود.

در این طراحی مقدار مقامت را $1^{M\Omega}$ انتخاب کردیم که پریود 8^{ms} را تولید می کند.

3.تبدیل خروجی به عدد

می دانیم که این سنسور هم دارای خروجی انالوگ و هم PWM می باشد.
خروجی انالوگ را می توان به A/D داد و سپس روی اعداد بدست امده عمل کالیبره کردن را انجام داد و یا می توان ازخروجی PWM استفاده کرده و روی اعداد T1 و T2 بدست امده عمل کالیبره کردن را انجام بدهیم.



شکل موج PWM متناسب با شتاب

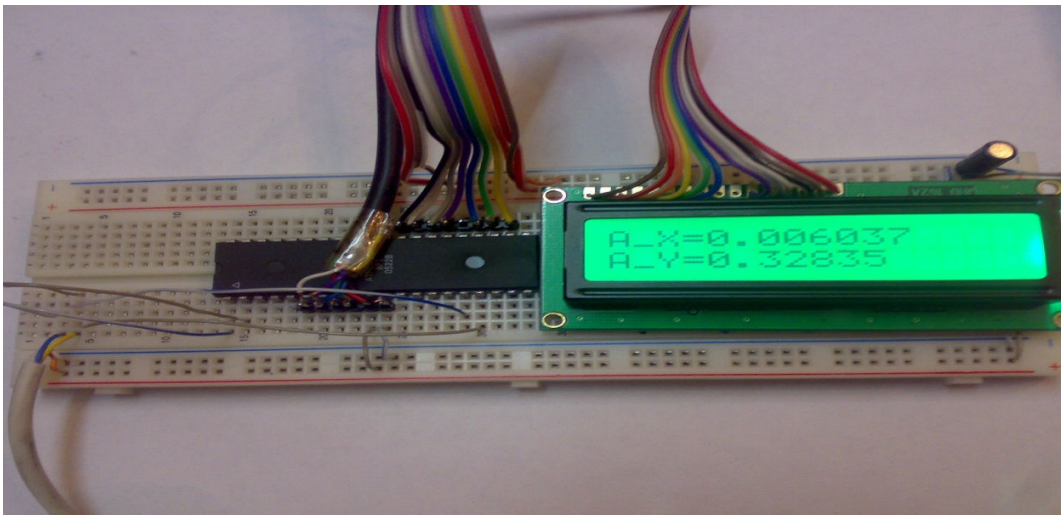
اما استفاده از PWM دارای دقت بیشتری است و ما از این روش استفاده می کنیم.

ما T1 و T2 را با کمک میکروکنترلر Atmega32 از خوانواده AVR با روش زیر بدست می اوریم.

روش کلی به این صورت است که خروجی x را هم به یک پایه میکرو و هم به پایه وقفه خارجی 0 متصل می کنیم و وقفه خارجی 0 را حساس به هر دو لبه بالا و پایین قرار می دهیم.با این عمل می توانیم بین اولین تغییر لبه تا تغییر لبه بعدی را اندازه گرفته و به کمک پایه میکرو بفهمیم که این بازه زمانی در چه سطح منطقی قرار دارد.و برای خروجی Y نیز همین عمل را انجام می دهیم.کد این قسمت برنامه که مربوط به X می باشد را در زیر مشاهده ی کنید.

```
interrupt [EXT_INT0] void ext_int0_isr(void)
{
if(flagx_0){
        if(flagx_2&&flagx_1){
                if(xread)                       //xread : پایه کنترلی میکرو برای تعیین سطح منطقی
                 twx=x;          //twx : T1              &           x : مقدار تایمر مجازی
                else
                 tpx=x;              ////tpx : T2
       flagx_2=0;
       flagx_1=0;
       flagx_0=0;
       tpx=tpx+twx;
       }
        if(flagx_1){
              if(xread)
               twx=x;
              else
               tpx=x;
       flagx_2=1;
       x=0;
```

```
        }
        if(!((flagx_2)&&(flagx_1))){
        x=0;
        flagx_1=1;
        }
}
}
```



مدار میکروکنترولر Atmega32 به همراه LCD برای نمایش

مشکلی که شاید در انجام این روش باشد این است که برای اندازه گیری X و Y به طور هم زمان نیاز به دو تایمر 16 بیتی داریم ولی Atmega32 فقط یک تایمر 16 بیتی دارد پس با روشی به نام تایمر مجازی که ان را نیز توضیح می دهم دو تایمر 16 بیتی برای ان می سازیم.

روش تایمر مجازی به این صورت است که با فعال کردن وقفه تایمر 16 بیتی بصورتی که با هر کلاک داخلی یک وقفه بدهد و سپس با قراردادن هر تعداد متغییر (در این پروژه دو متغییر) و افزایش یک واحدی به انها به ازای اجرای هر بار وقفه می توان به تعداد متغیرها, تایمر 16 بیتی داشت.برنامه تایمر مجازی را میتوانید در زیر مشاهده کنید.

```
interrupt [TIM1_COMPA] void timer1_compa_isr(void){
```

```
x++;
y++;
TCNT1=0;
}
```

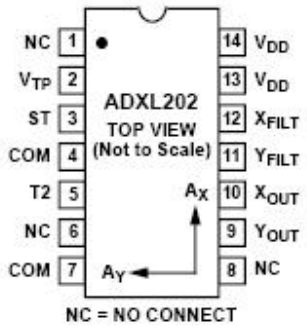
بعد از این که T1 و T2 را بدست اوردیم می توانیم به مرحله کالیبره کردن سنسور رفته و شتاب را از این داده ها استخراج کنیم.

4.1.کالیبره کردن و استخراج شتاب

این سنسور هم شتاب استاتیک و هم شتاب دینامیک را اندازه می گیرد. اما برای کالیبره کردن ان ما از شتاب استاتیک استفاده می کنیم.

ابتدا سنسور را روی سطحی کاملا مسطح قرار می دهیم تا سنسور در راستای X و Y هیچ مولفه شتابی را احساس نکند تا بتوانیم اعداد را روی صفر به گونه ای که مشاهده می کنید تنظیم کنیم.

T2	T1
1843	865
1832	856
1843	864
1844	864
1843	862
1844	867
1834	858
1835	860
1838	860
1836	861

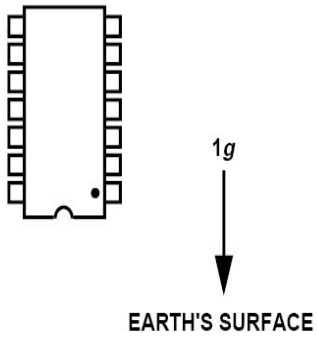


جهت های شتاب در راستای محورهای X و Y

با بدست اوردن متوسط T1وT2 و سپس تقسیم انها عددی بدست می اید(برای این پروژه عدد 0.468518 بدست امد) که باید مقدار $\frac{T1}{T2}$ از ان کم شود.و اگر ما مقدار 0.468518- $\frac{T1}{T2}$ را برای ده نمونه دیگر در حالت استاتیک, که شتاب جازبه زمین بر راستای X وارد می شود

محاسبه کنیم می توانیم ثابتی که عدد 0.468518- $\frac{T1}{T2}$ را به شتاب تبدیل می کند را بدست اوریم.

$\frac{T1}{T2}$ -0.468518
0.124453
0.12251
0.12389
0.12335
0.12379
0.12335
0.12325
0.12303
0.12391
0.12303



راستای قرار گیری سنسور

و متوسط مقادیر روبرو عدد 0.123456 است که با قرار گرفتن در رابطه زیر ثابت نهایی را بدست می دهد.

$$9.81=K*0.123456$$

مقدار K برابر 79.461509 می شود و معادله نهایی بصورت زیر است

$$A_x = (\frac{T1}{T2} \text{-0.468518})* 79.461509$$

A_y نیز به همین صورت محاسبه می شود.دقت این روش کالیبراسیون در حدود $\frac{g}{17}$ است.حال سنسور کالیبره بوده و می تواند شتاب را اندازه گیری کند. کد برنامه مربوط به این کالیبراسیون را در زیر مشاهده می کنید.

```
float accelerometer_X(void){
float sumx=0;
A_X=0;
for(j=0;j<10;j++){
flagx_1=0;
```

مجید امانی
amani.majid69@yahoo.com

```
flagx_0=1;

delay_ms(16);

sumx=(((twx*1.000000)/(tpx*1.000000))-0.468518  )*  79.461509

A_X+=sumx;

}

A_X=(A_X/(10.000000));

}
```

4.2.کالیبره کردن و استخراج شتاب

روشی که در بالا برای کالیبره کردن این سنسور به کار بردیم بسیار مقدماتی بوده و شامل پارامترهای اساسی تغییر شتاب نیست.

دما مهمترین عاملی است که باعث تغییرات عظیمی در خروجی این سنسور می شود.نمودارهای تغییرات پارامترهای مختلف که به دما وابسته هستند را در زیرمشاهده می کنید.

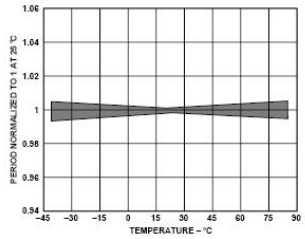


Figure 2. Normalized DCM Period (T2) vs. Temperature

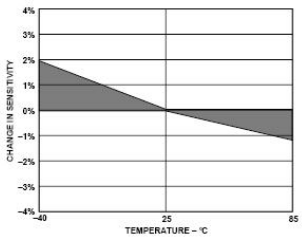


Figure 5. Typical X Axis Sensitivity Drift Due to Temperature

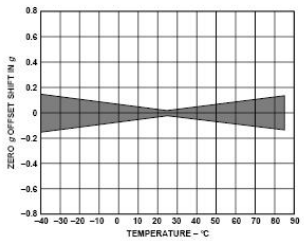


Figure 3. Typical Zero g Offset vs. Temperature

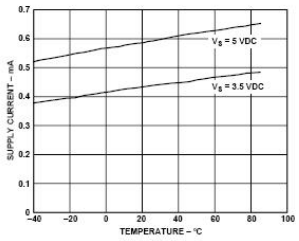


Figure 4. Typical Supply Current vs. Temperature

اگر سنسور ما با روش 4.1 کالیبره شده باشد عملا دقت بالایی ندارد(در حدود $\frac{g}{17}$).و در شرایطی که تغییرات دمایی شدید باشد سنسور عملا به کار نمی اید.

ما با روش 4.2 دقت را به $\frac{g}{100}$ می رسانیم اما همچنان مشکل دما وجود دارد. در روش بعد مشکل دما نیز حل می شود.

در اینجا میخواهیم فقط تئوری این روش را بگوییم چون اولا شما با روش اجرای این تئوری اشنا هستید(در روش قبل یک تئوری را عملا به اجرا در اوردیم) و دوما روش بعدی روشی بسیار کامل تری است و ما بیشتر به ان خواهیم پرداخت.

1.با یک نمونه گیری T2 در شتاب 0 را بدست اورده و نام ان را $T2_{cal}$ می گذاریم.

2.با نمونه گیری از T1 در شتاب 0 مقدار بدست امده را Z_{cal} نام گذاری می کنیم.

3. $$S = \frac{T2_{cal} * \mathrm{k}}{T1max - T1min}$$

4. $$Z_{act} = \frac{T2}{T2_{cal}} * Z_{\mathrm{cal}}$$

5. $$A = S * \frac{T1 - Z_{act}}{T2}$$

همانطور که می بینید Z_{act} از خروجی T2 فیدبک می گیرد و تا حدودی میتواند تغییرات دمایی را نیز خنثی کند.همانطور که گفته شد دقت این روش $\frac{g}{100}$ است. ودر شرایط تغییرات دمایی شدید دقت ان به شدت افت می کند اما کارایی خود را از دست نمی دهد.

4.3.کالیبره کردن و استخراج شتاب

روشی که گفته می شود بسیار کارامد وقدرتمند است.دقتی در حدود $\frac{g}{450}$ دارد. و پایداری حرارتی بسیار بالایی دارد.علاوه بر ان هر اختلالی که باعث تغییر خروجی از حالت درست به نادرست شود در این روش از بین می رود.

مدولاتور تولید کننده پالس سنسور به هر دو خروجی x و y متصل است و خطایی که در اثر تغییرات دما در پالس تولید می شود به هر دو خروجی به طور یکسان اعمال می شود . تایمر میکرو هم که پالس را اندازه می گیرد برای هر دو خروجی یکسان است(روش تایمر مجازی از یک تایمر استفاده می کند).پس ما این روش را برای یکی از خروجی ها اعمال می کنیم و مطمئنیم که برای خروجی دیگر نیز صادق است.

اما ما روابط زیر را برای محاسبه T1 و T2 داشتیم:

$$T1=twx \qquad T2=tpx+twx$$

پس با مشتق گیری داریم: $dT1=dtwx \quad dT2=dtpx+dtwx$

اما از انجا که هر دو اینها با یک تایمر اندازه گیری شده $dtwx=dtpx$ پس اگر رابطه(st) رو به رو راداشته باشیم: $subt\_time=T2-2*T1$

انگاه با مشتق گیریداریم: $dsubt\_time=dT2-2*dT1$

$$dsubt\_time= dtpx+dtwx -2* dtw \qquad (st\_1)$$

پس طبق برابری (dtwx=dtpx) رابطه (st_1) متحد با صفر است

و در نتیجه رابط (st) مستقل از هر گونه خطا می باشد.و همچنین می تواند معیار خوبی برای اندازه گیری شتاب به ما بدهد.با نمونه گیری از رابطه st درشتاب صفر و قرار دادن در متغییر zcal رابطه زیر شتاب را به ما می دهد.

$$A = (subt\_time - Zcal) * K$$

.کد این قسمت را در زیر مشاهده می کنید.

```
for(j=0;j<avg;j++){
 flagx_1=0;
 flagx_0=1;
delay_ms(7);
subtract_timex=tpx*1.0-2.0*twx;
sumx+=subtract_timex;
}
```

مجید امانی
amani.majid69@yahoo.com

```
subtract_timex=sumx/avg;

zactx=subtract_timex-zcalx;

A_X=zactx*kx;
```

5.اندازه گیری شتاب در حالت استاتیک

اگر سنسور در حالت ثابت نسبت به راستای عمود زاویه ای داشته باشد طبق روابط زیر این زاویه قابل محاسبه است.

$$G*\cos(\alpha)=A_x$$

$$g*\sin(\alpha)=A_y$$

$$\alpha=\tan^{-1}\frac{A_y}{A_x}$$

6.اندازه گیری سرعت و فاصله در حالت دینامیک

می دانیم که با انتگرال گیری از شتاب سرعت و با انتگرال گیری از سرعت فاصله به دست می اید.پس ما با در نظر گرفتن سرعت اولیه صفر و مکان اولیه صفر می توانیم سرعت و مکان سنسور را در هر لحظه بدست اوریم.

برای این که رابطه $V_{x2}-V_{x1}=A_x*(t_2-t_1)$ به فرم

در اید و ما بتوانیم سرعت لحظه ای را محاسبه کنیم باید در فواصل زمانی بسیار کوتاه رابطه $V_{x2}-V_{x1}=A_x*T$ را حساب کنیم که برای این طراحی مقدار T را برابر 1.024^{ms} قرار دادیم.روش کلی به این صورت است که یه تایمر را در حالت شمارش قرار داده و مقدار اورفلو ان را طوری تنظیم میکنیم که به ازای یک زمان مشخص عمل اورفلو انجام شده وبرنامه وارد زیر برنامه وقفه ی اورفلو تایمر میشود و در ان زیر برنامه عمل انتگرال روی شتاب و سرعت انجام می گیرد.

بدست اوردن فاصله نیز روشی کاملا مشابه شرعت دارد فقط جای شتاب با سرعت عوض می شود.در زیر کد مربوط به این قسمت برنامه را مشاهده می کنید.

مجید امانی
amani.majid69@yahoo.com

```
Interrupt [TIM1_OVF] void timer1_ovf_isr(void)
{
// Reinitialize Timer 1 value
TCNT1H=0xFF;
TCNT1L=0xEF;
//Place your code here
Vx+=Ax*(1.024e-3);
Vy+=Ay*(1.024e-3);
x+=Vx*(1.024e-3);
y+=Vy*(1.024e-3);
}
```

لازم به ذکر است که حتی در بهترین حالت یعنی با کمترین خطا در محاسبه شتاب و با دقت فوق العاده ($\frac{g}{450}$) در محاسبه سرعت و فاصله به خاطر انتگرال گیری و ضرب شدن T در ان , این خطای کوچک در مدت یک دقیقه به خطایی معادل $1.308\frac{m}{s}$ در محاسبه سرعت و 39.24m در محاسبه فاصله می رسد و در دقیقه دوم به خطایی معادل $2.616\frac{m}{s}$ در محاسبه سرعت و 117.72m در محسبه فاصله می رسد.

7.نتیجه گیری

این سنسور می تواند با روش های گفته شده زاویه را با دقت فوق العاده ای محسبه کند.برای محاسبه شتاب برای ربات های بالانس بسیار مناسب است.

و خطای فاحش ان در محاسبه سرعت و فاصله بدان معنا نیست که بدست اوردن سرعت و فاصله در عمل با این سنسور ممکن نیست بلکه برای تحقق این هدف نیاز به داده هایی از انکودرها ویا دیگر سنسورهای تشخیص حرکت است تا با کمک هم بتوانند دیتای سالم و بدون خطایی را در محاسبه سرعت و فاصله استخراج کنند.

مجید امانی
amani.majid69@yahoo.com