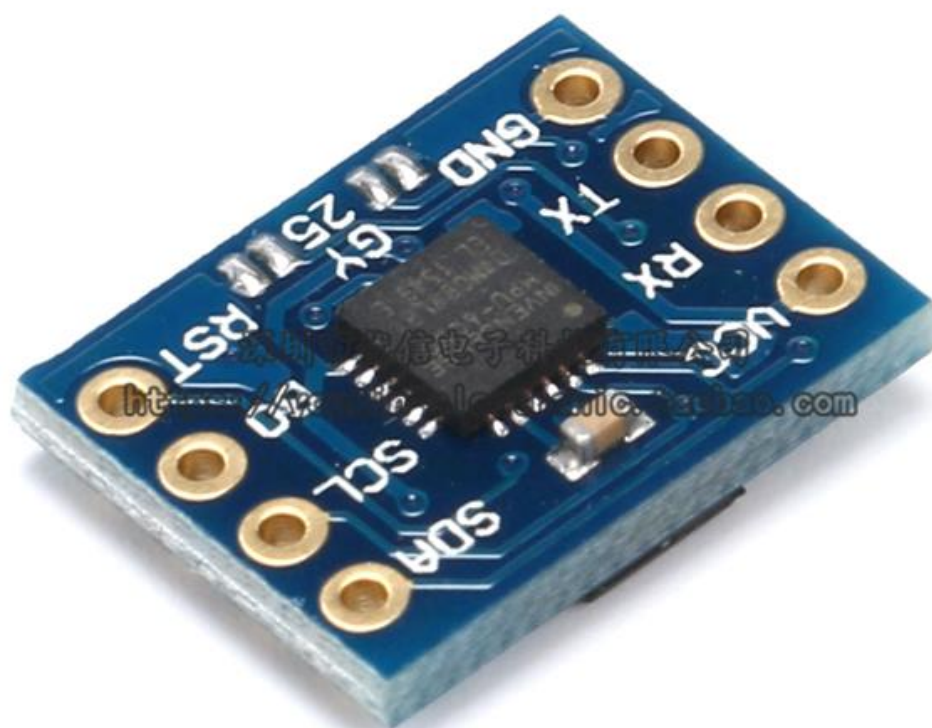


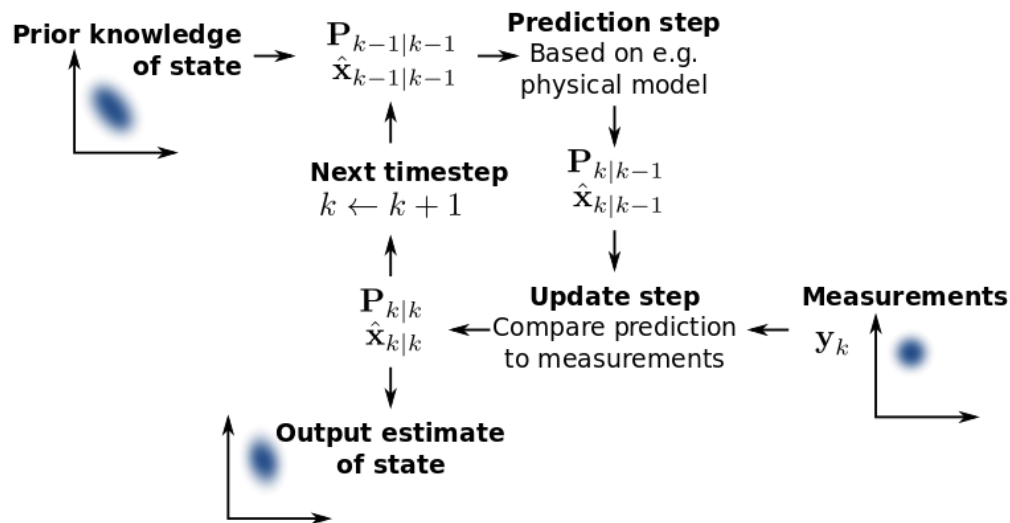


## راه اندازی سنسور GY-25 شتاب سنج ۳ محوره با فیلتر کالمن



## معرفی فیلتر کالمن:

فیلتر کالمن که رودولف کالمن پایه گذار آن بود یک الگوریتم است که طبق آن پارامتری از یک سیستم را با استفاده از مجموعه ای از اندازه گیری ها برآورد می کنند. این اندازه گیری ها ممکن است به تنهایی دارای خطاهایی باشند اما هنگامی که با یکدیگر دیده شوند تخمین دقیق تری از واقعیت را به ما می دهند.



به عنوان مثال فرض کنید قصد داریم سرعت حرکت و مکان یک خودرو را اندازه بگیریم. سنسور هایی که برای انجام این کار وجود دارد از جمله سنسور های شتاب سنج، دور سنج چرخ خودرو و ماهواره های GPS.



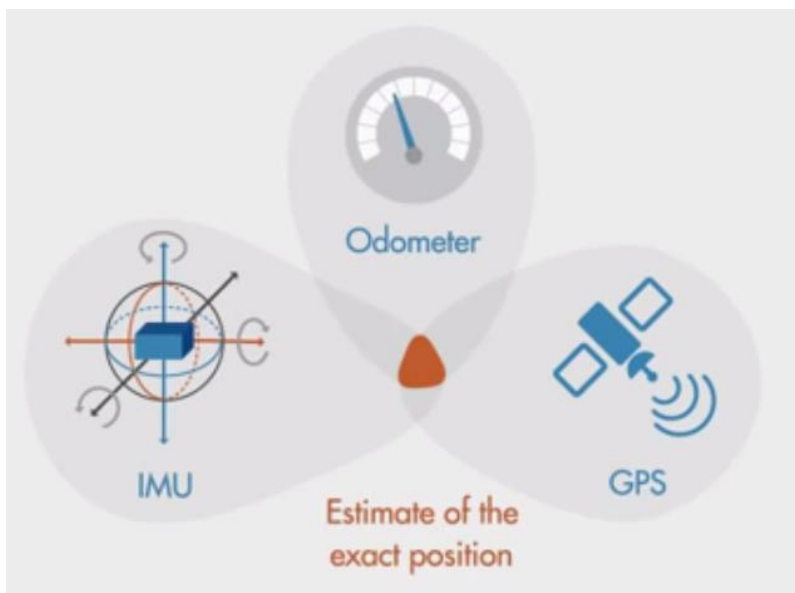
شتاب سنج

دورسنج چرخ خودرو

ماهواره GPS

هر کدام از این اندازه گیری ها دارای خطاهایی هستند. برای مثال هنگامی که داخل یک تونل در حال رانندگی هستیم امکان استفاده از GPS را نخواهیم داشت. در این صورت مجبور می شویم از سنسور شتاب سنج استفاده کنیم اما شتاب سنج در مورد مکان خودرو اطلاعاتی به ما نمی دهد بلکه شتاب حرکت را اندازه می گیرد. برای محاسبه مکان از روی شتاب نیاز است دو بار از آن بر حسب زمان انتگرال بگیریم (مساحت زیر نمودار a-t) اما این کار نیز دارای خطاهایی است که مربوط به زمان می شود. برای تخمین بهتر موقعیت خودرو می توان از دورسنج

استفاده کرد اما حتی این سنسور هم به فشار تایر های خودرو و وضعیت آسفالت جاده بستگی دارد و خطاهایی دارد. در اینجا با استفاده از فیلتر کالمن می توانیم از هر سه این اندازه گیری ها استفاده کنیم تا به تخمین دقیق تری برسیم.



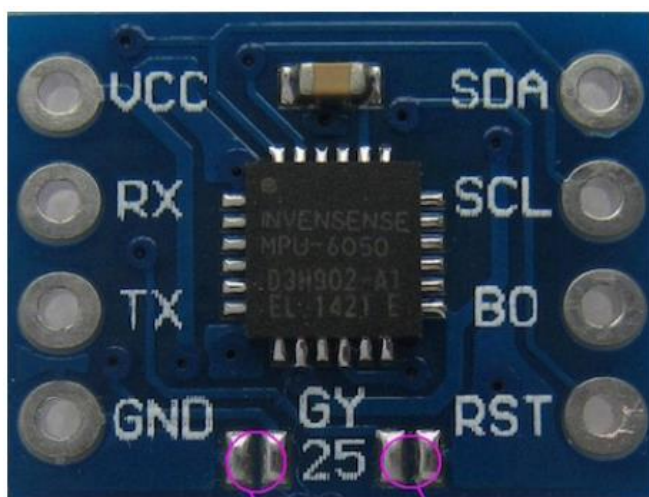
**ماژول شتاب سنج سه محوره:**

ماژول GY-25 دارای سنسور های شتاب سنج در سه محور  $x$ ،  $y$  و  $z$  است که به یک میکروکنترلر متصل شده و اطلاعاتی را که از سنسورها دریافت می کند پس از محاسبات و اعمال فیلتر هایی نظیر فیلتر کالمن برای تخمین بهتر و دقیق تر اندازه گیری، خروجی حاصل را از طریق ارتباط Serial و یا i2c به میکروکنترلر های دیگر منتقل می کند.

**مشخصات:**

- دقت اندازه گیری: 0.01 درجه
- فرکانس پاسخ دهی: 100Hz (115200bps)
- ولتاژ کاری: 5v
- جریان کاری: 15mA
- ابعاد: 11.5mm \* 15.5mm
- نحوه ارتباط: Serial و i2c

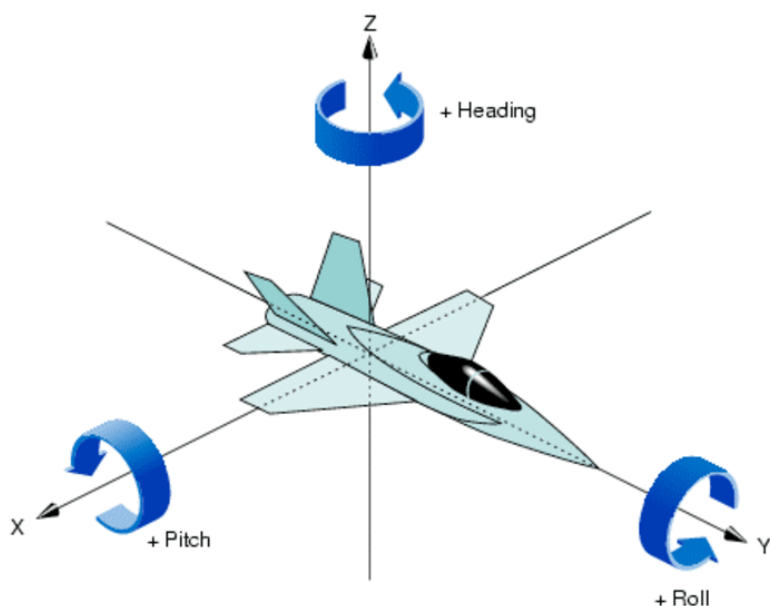
نام پایه	کاربرد	نام پایه	کاربرد
VCC	ولتاژ ورودی (۳ تا ۵ ولت)	SDA	i2c clock
RX	دریافت کننده (ارتباط Serial)	SCL	i2c data
TX	ارسال کننده (ارتباط Serial)	B0	استفاده داخلی، بدون اتصال
GND	منفی	RST	استفاده داخلی، بدون اتصال



open : I2C vers MPU6050  
close : I2C vers sorties

open : 115200  
close : 9600

## محور های دورانی:



این شتاب سنج زاویه را در سه جهت اندازه می گیرد که به این سه محور Roll، Heading و pitch گفته می شود. همانند تصویر روبرو این سه محور بر هم عمود هستند و جهت مثبت چرخش در هر کدام از آن ها نشان داده شده است.

## نحوه ارسال دیتا:

هر بسته ارسالی از ماژول GY-25 شامل ۸ بایت مبنای ۱۶ (Hex) می شود که عبارتند از:

۱. 0xAA ← بیت شروع
۲. 0x00 تا 0xFF ← رقم پرارزش زاویه Heading
۳. 0x00 تا 0xFF ← رقم کم ارزش زاویه Heading
۴. 0x00 تا 0xFF ← رقم پرارزش زاویه Pitch
۵. 0x00 تا 0xFF ← رقم کم ارزش زاویه Pitch
۶. 0x00 تا 0xFF ← رقم پرارزش زاویه Roll
۷. 0x00 تا 0xFF ← رقم کم ارزش زاویه Roll
۸. 0x55 ← بیت پایان

**نکته:** نحوه محاسبه هر زاویه با استفاده از رقم کم ارزش (LOW) و رقم پر ارزش (HIGH) آن در زبان برنامه نویسی C به صورت زیر است:

$$\text{Angle} = ((\text{HIGH} \ll 8) | \text{LOW}) / 100;$$

## دستورات:

- 1) **0xA5 + 0x51:** query mode, return directly to the angle value, to be sent each read.
- 2) **0xA5 + 0x52:** Automatic mode, send a direct return angle, only initialization.
- 3) **0xA5 + 0x53:** Automatic mode, ASCII code output, serial port for direct computer assistant View.
- 4) **0xA5 + 0x54:** correction mode, the pitch correction roll angle of 0 degrees, need to stay level when sending.
- 5) **0xA5 + 0x55:** correction mode, 0-degree course correction, heading cleared at any angle.

## نکات مهم:

۱. از آنجایی که این ماژول دارای Self-correction در هنگام روشن شدن است، (یعنی هنگامی که ماژول روشن می‌شود خودش را ست می‌کند) می‌بایست حداقل ۳ ثانیه ماژول را ثابت گذاشت.
۲. زاویه Heading این ماژول در زمان های طولانی کمی می‌چرخد.
۳. تمام پایه های این ماژول تلورانس ۵ ولت دارند. به همین خاطر می‌توان آن را هم با 3.3v و هم با 5v راه اندازی کرد.

## راه اندازی ماژول با آردوینو:

```
int buff[8];
int counter = 0;
float Roll,Pitch,Heading;

void setup() {
    delay(1000);
    Serial.begin(115200);
    Serial.write(0XA5);
    Serial.write(0X54);
    delay(1000);
    Serial.write(0XA5);
    Serial.write(0X51);
}

void loop() {
    Serial.write(0XA5);
    Serial.write(0X51);
    while (true) {
        buff[counter] = Serial.read();
        if(counter == 0 && buff[0] != 0xAA) break;
        counter++;
        if(counter==8)
        {
            counter=0;
            if(buff[0]==0xAA && buff[7]==0x55)
            {
                Heading=(int16_t) (buff[1]<<8|buff[2])/100.00;
                Pitch=(int16_t) (buff[3]<<8|buff[4])/100.00;
                Roll=(int16_t) (buff[5]<<8|buff[6])/100.00;
            }
        }
    }
}
```