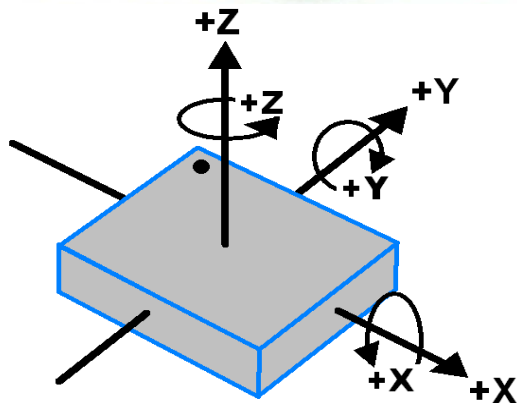
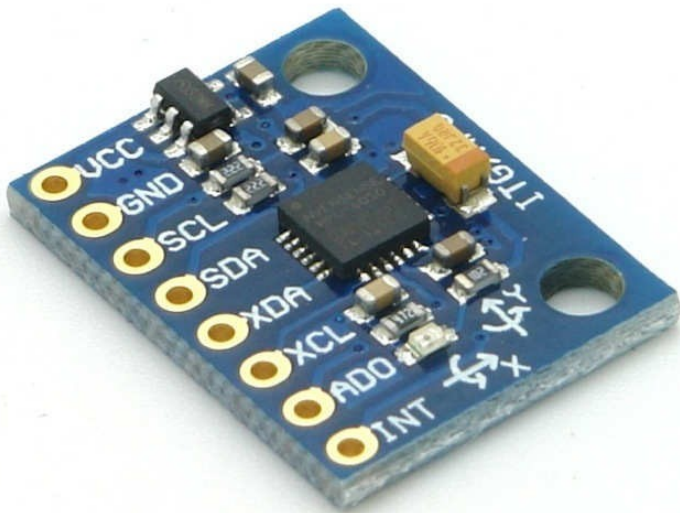
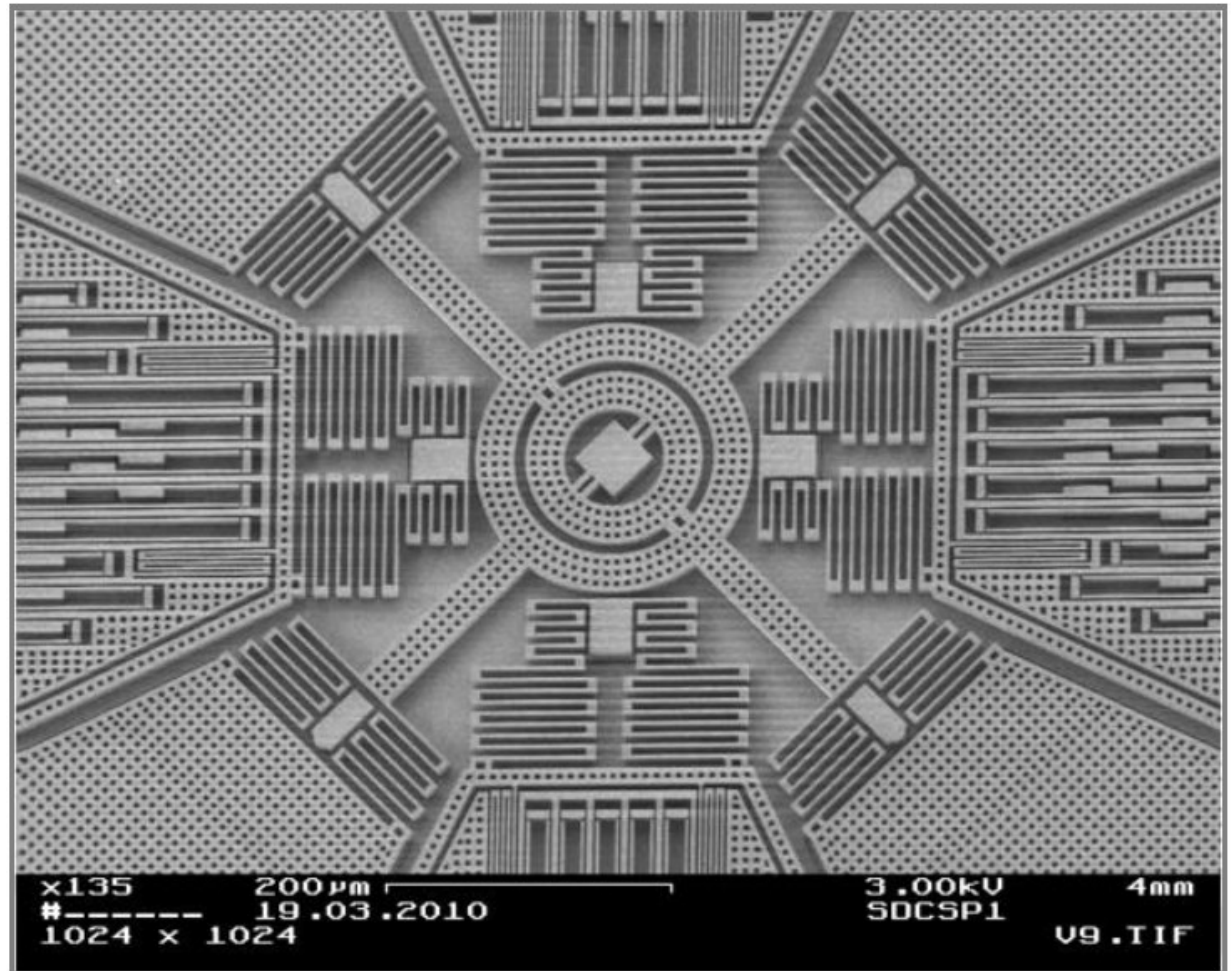


# Vegyes témakörök



MPU-6050  
Orientation & Polarity of Rotation



## 12. Az MPU6050 gyorsulás- és szögsebességmérő szenzor

# Felhasznált források

---

## Magyar nyelvű irodalom:

- ÉRZÉKELŐK ÉS BEAVATKOZÓK I. /2. ÉRZÉKELÉS ÉS MÉRÉS (Dr. Soumelidis Alexandros, BME KJIT)
- Multicopterek, nulláról az UAV-kig III. - IMU szenzorok (Topor Zoltán, Hobbielektronika)

## Angol nyelvű irodalom:

- An Overview of MEMS Sensors (Bal S. Sandhu, ARM)
- MPU6050 sensor module (Electronic Wings)
- Fundamentals of Attitude Estimation (CH Robotics)
- Lets work with MPU6050 (GY-521)—Part1 (Kavindu Gimhan Zoysa)
- MPU6050 Interfacing With Arduino UNO (Electronic Wings)
- I2Cdevlib (Jeff Rowberg, GitHUB)

# Hely- és helyzetérzékelés

- Egy test **helye** és **helyzete** egy inerciarendszerben relatív mennyiségek. (Inerciarendszer: olyan vonatkoztatási rendszer, amelyhez viszonyítva egy test mozgására érvényes Newton tehetetlenségi törvénye)
- Abszolút mennyiségek:
  - ❖ Gyorsulások –  $a_x, a_y, a_z$
  - ❖ Szögsebességek –  $\omega_x, \omega_y, \omega_z$
- Ezekből pozíció, sebesség és szöghelyzet számítható:

$$s(t) = s_0 + v_0 t + \int_{t_0}^t \int_{t_0}^t a(\tau) d\tau$$
$$v(t) = v_0 + \int_{t_0}^t a(\tau) d\tau$$
$$\varphi(t) = \varphi_0 + \int_{t_0}^t \omega(\tau) d\tau$$

- ❖ Határozatlan kezdeti feltételeket tartalmaznak
- ❖ Meghatározásuk csak halmozódó hibával lehetséges

# Giroszkópok és gyorsulásmérők

---

- Alkalmazásuk pl. hajózásban, repülésben, űrrepülésben:
  - ❖ A jármű mozgásának stabilizálása
  - ❖ Manőverek irányítása
  - ❖ Navigáció
- Eszközök:
  - ❖ Pörgettyűs giroszkópok
  - ❖ Piezoelektromos rezgőelemes giroszkóp
  - ❖ Lézer alapú giroszkópok
  - ❖ Gyorsulásmérők
  - ❖ MEMS szenzorok (Micro-electromechanical systems)
  - ❖ Inerciális Navigációs Rendszerek

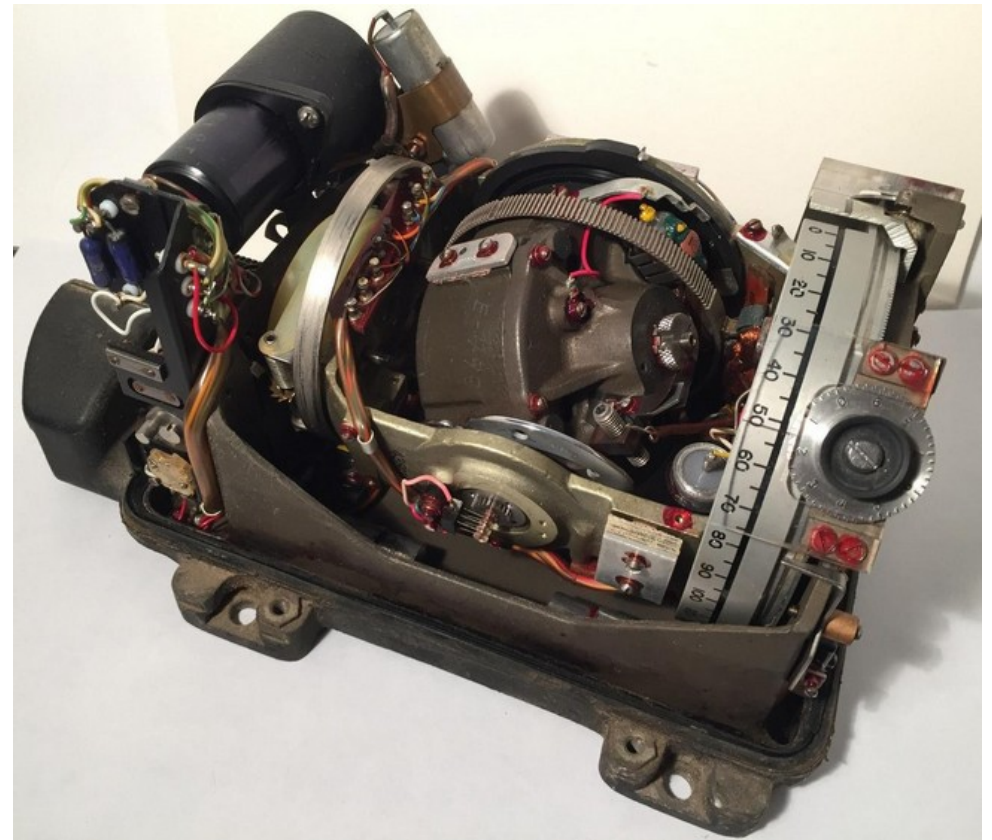
# Pörgettyűs giroszkópok

- Fizikai elv:
  - ❖ Newton axiómák
  - ❖ Impulzusmegmaradás törvénye
- Egy forgásban levő merev test forgástengelye stabil egyensúlyi helyzetben van
- Nem feltétlenül áll, hanem egy meghatározott irány körül egy kúp mentén forog – precesszál
- Ha az eszközt kardán felfüggesztéssel látjuk el, akkor a pörgő kerék megőrzi forgási tengelyének eredeti irányát, akkor is, ha a kerete elfordul



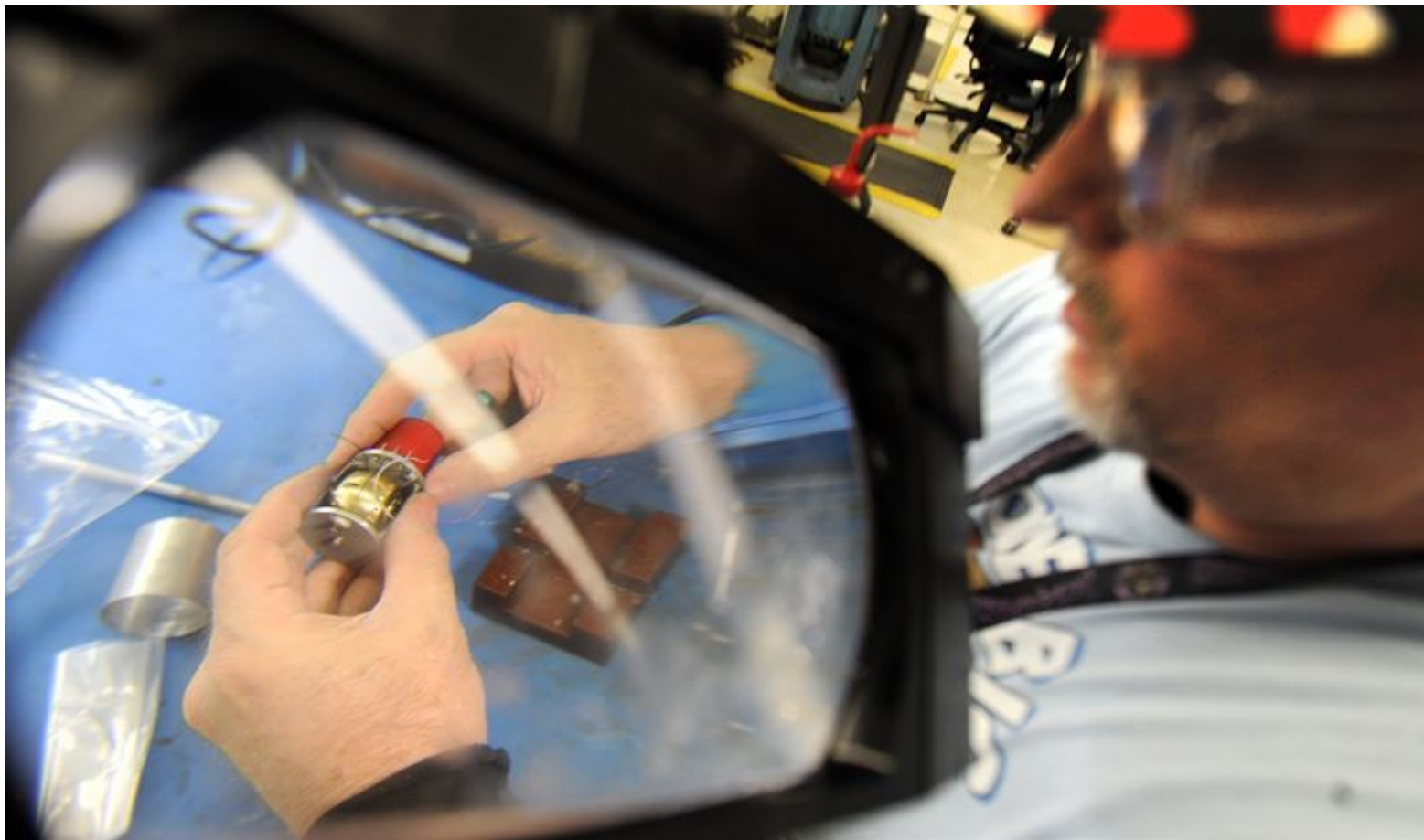
# Kéttengelyű pörgettyűs giroszkóp

- A Honeywell JG7044N giroszkóp egy függőleges tengelyű pörgettyűből áll, amelyet két gyűrű (gimbal) tart.
- A függőlegest egy gravitációs szenzor jelöli ki.
- A két gyűrű elfordulása a bólintó (pitch) és orsozó (roll) mozgás érzékelésére szolgál (pl. műhorizont)
- Az eszközt az 1950-es években fejlesztették ki
- A képen látható példányt az USA Légierő georgiai Robins Légibázisán használták



# Még mindig pörög a giroszkóp...

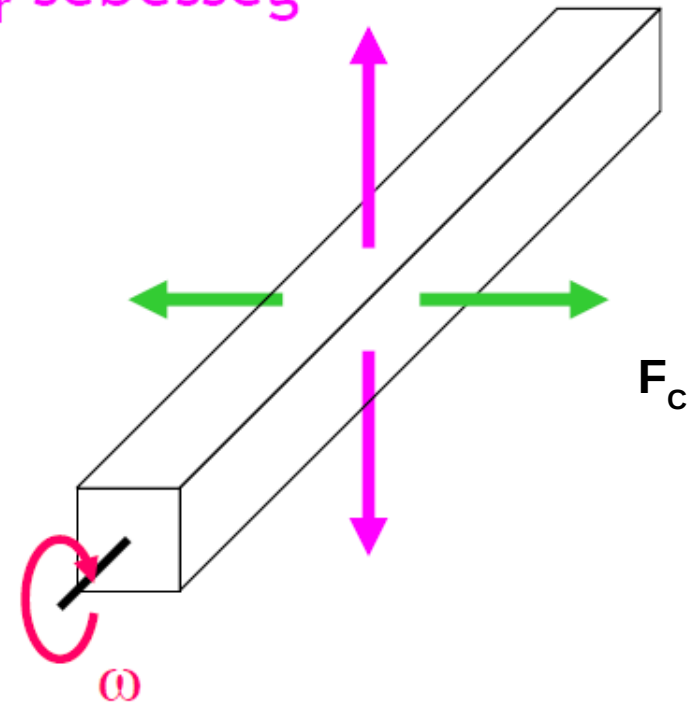
- 2013-ban probléma merült fel az US Army UH-60 típusú Black Hawk helikoptereinek duál száloptikai lézer giroszkóp rendszerével ezért hagyományos, mechanikus giroszkópok (lásd a képen) beszerelése mellett döntöttek [[link](#)]



# Rezgőelemes giroszkóp

- Alapelv: rugalmas rúd kényszerrezgése, Coriolis erő
- $\omega$  szögsebességű forgás esetén a fellépő  $F_c$  Coriolis erő a rákényszerített rezgésre merőleges deformációt okoz  
→ Mérhető
- Mérés elve szerint:
  - ❖ Piezoelektromos hatás
  - ❖ Kapacitív elvű elmozdulás-mérés
- Az alkalmazott technológia szerint:
  - ❖ Piezokeramikus kristály
  - ❖ MEMS – Micro ElectroMechanical System

Rákényszerített rezgés  
 $v_r$  sebesség

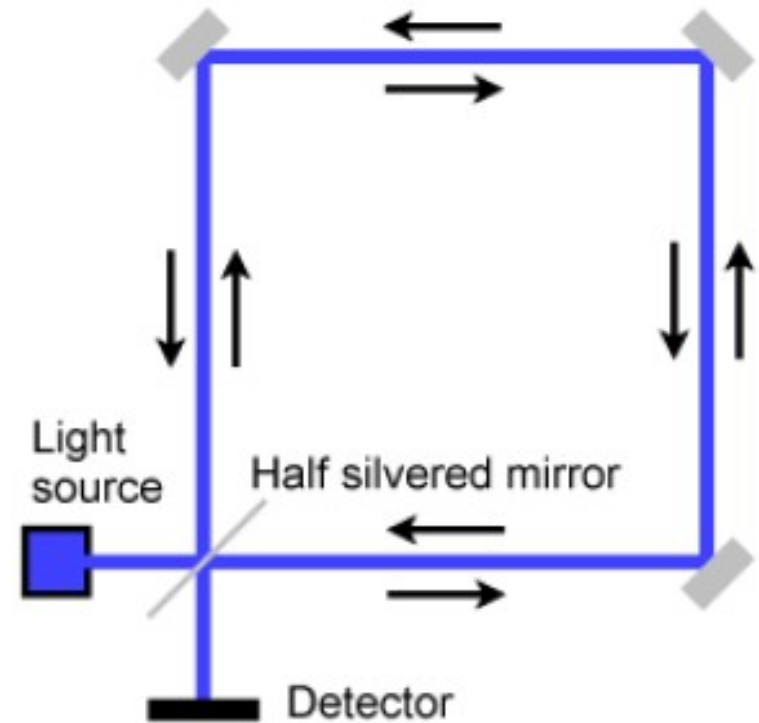


$$F_c = -m(\omega \times v_r \cdot L)$$



# Lézer giroszkóp

- Fizikai elv: **Sagnac effektus**  
Georges Sagnac (1869-1928) francia fizikus
- A két ellentétes irányú fénysugár interferál egymással
- Ha a rendszer forgásban van, fáziseltérés lép fel, megváltozik az interferenciakép
- Típusok:
  - ❖ Ring Laser Gyroscope (RLG)
  - ❖ Fiber Optic Gyroscope (FOG)

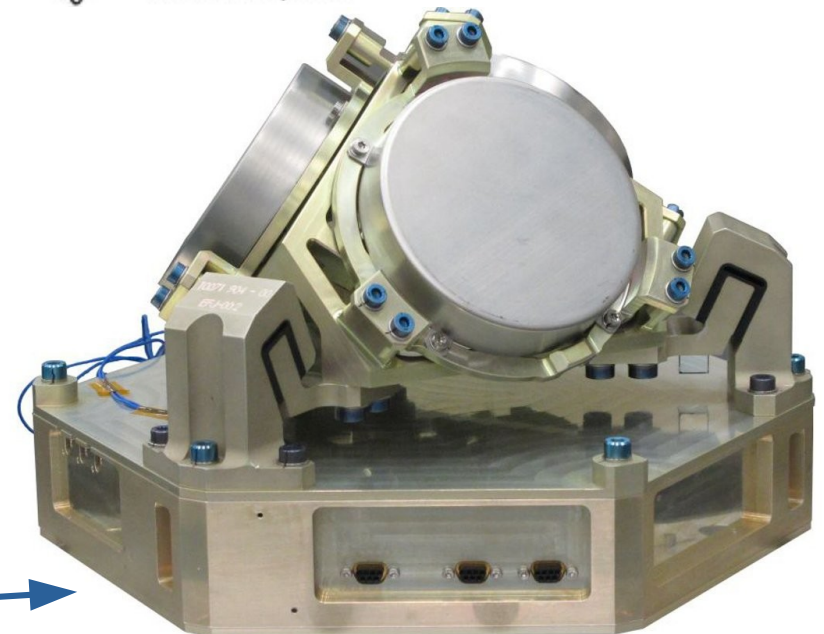
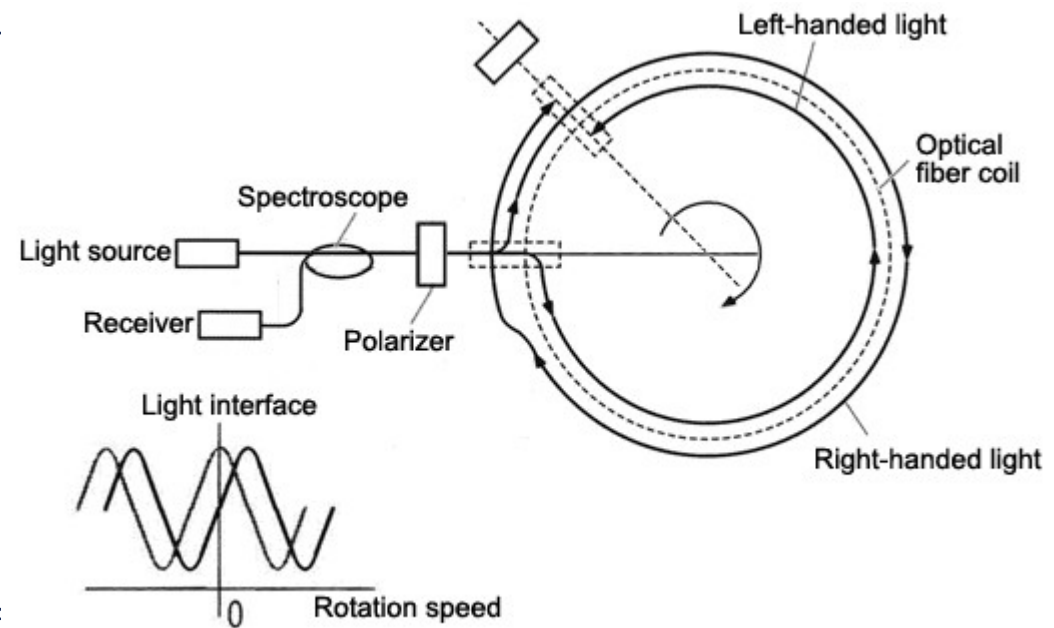


Sagnac interferométer

- Előnyök: nagy pontosság, és érzékenység, kis zaj
- Hátrányok: költséges és bonyolult megoldás

# Száloptikai lézer giroszkóp

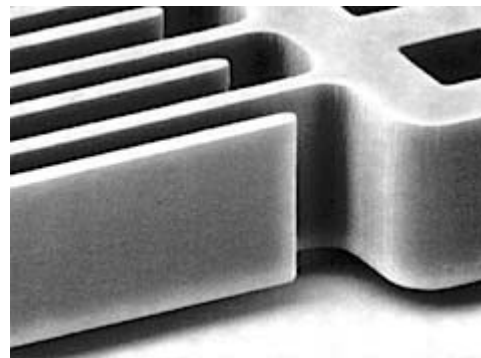
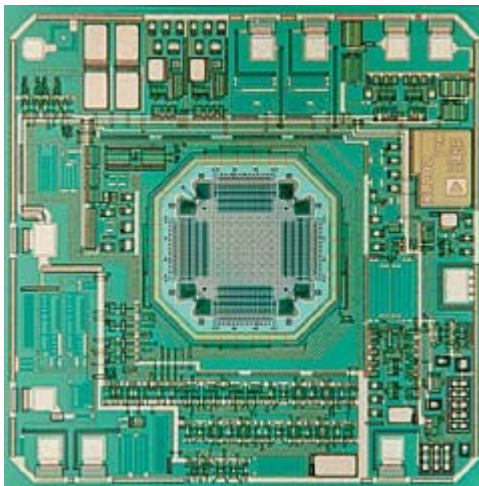
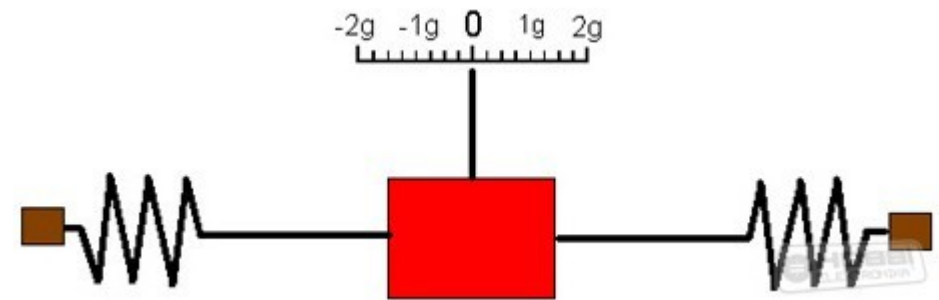
- A Fiber Optic Gyroscope (FOG) nem rendelkezik mozgó alkatrészsel, és nincs tehetetlensége
- Feltekercselt (akár 5 km-es) optikai szálból áll, és a fény interferenciáját használja a mechanikus forgatás észlelésére (a Sagnac-effektus alapján)
- A FOG rendkívül pontos forgási sebességadatokat nyújt, mivel érzéketlen a rezgésre, gyorsulásra és rázkódásra



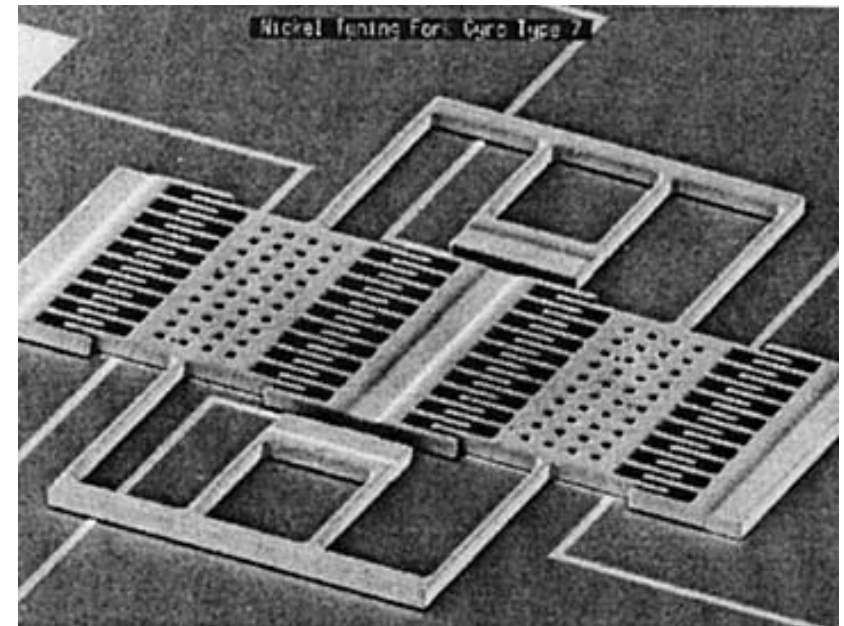
New **Astrix 1090 Fibre-Optic Gyro (FOG)**  
from Airbus Defense and Space

# Gyorsulás érzékelők

- Fizikai elv: rugó – tömeg együttes
- Érzékelés elve
  - ❖ Piezokeramikus  
Érzékelés piezoelektromos elven
  - ❖ MEMS
    - Kapacitív
    - Termikus



20  $\mu\text{m}$ -es Si fésű  
struktúra (Draper Lab)



A Draper Lab fésű struktúrával  
kialakított szenzora

Analog Devices ADXL 50 gyorsulásmérő

# Gyorsulás érzékelők

- A gyorsulás vektoriális egyenlete Newton törvénye alapján:

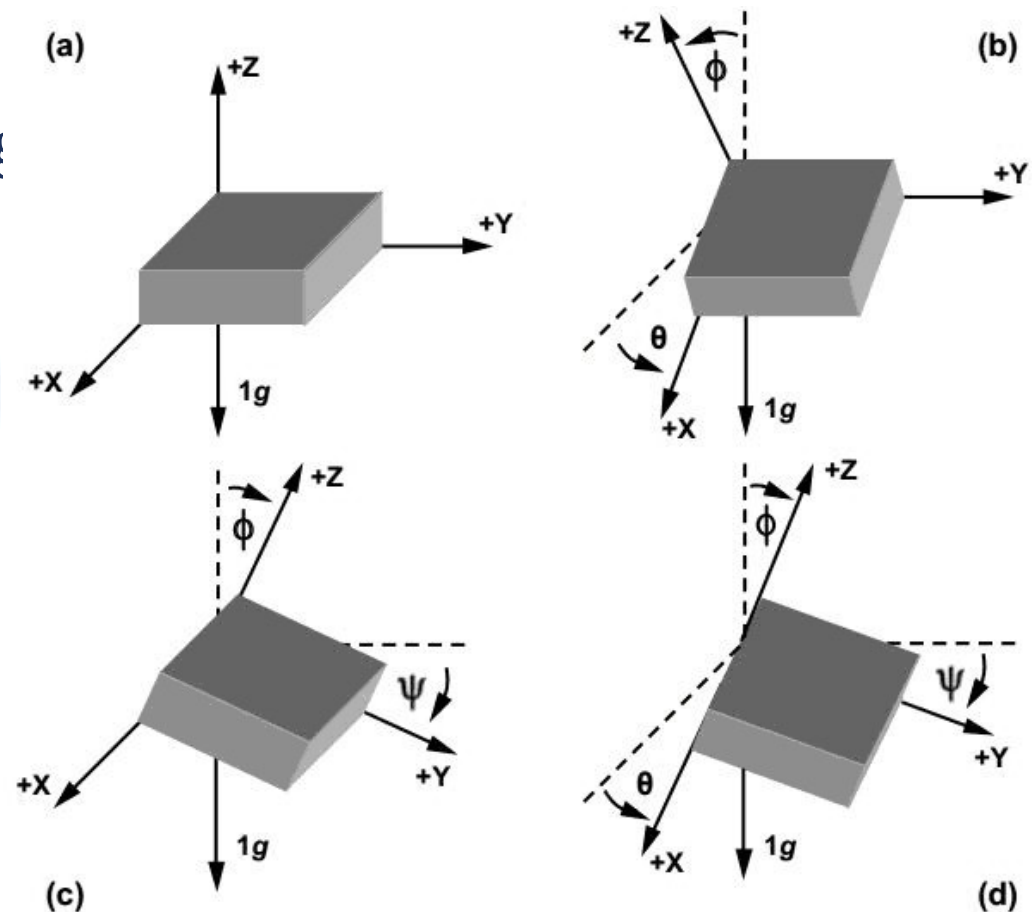
$$\mathbf{a}_m = \frac{1}{m}(\mathbf{F} - \mathbf{F}_g), \quad \text{ideális esetben } \mathbf{F} = 0 \quad (\text{csak a gravitáció hat})$$

ahol  $\mathbf{a}_m$  a gyorsulás,  $m$  a tömeg,  $\mathbf{F}$  a tömegre ható erők eredője,  $\mathbf{F}_g$  pedig a gravitációs erő

- A nehézségi erő a szenzor elforgatási rendszerében

$$\mathbf{F}_g = R_I^B \begin{pmatrix} 0 \\ 0 \\ mg \end{pmatrix} = \begin{pmatrix} -mg \sin(\theta) \\ mg \cos(\theta) \sin(\phi) \\ mg \cos(\phi) \cos(\theta) \end{pmatrix}$$

$$\mathbf{a}_m = \begin{pmatrix} g \sin(\theta) \\ -g \cos(\theta) \sin(\phi) \\ -g \cos(\phi) \cos(\theta) \end{pmatrix}$$



# Gyorsulás érzékelők

- A gyorsulás mért értékeiből meghatározhatjuk a  $\theta$  emelkedési szöget (pitch) és a  $\phi$  orsózási szöget (roll)

$$\hat{\theta}_{accel} = \arcsin\left(\frac{a_{m,x}}{g}\right),$$
$$\hat{\phi}_{accel} = \arctan\left(\frac{a_{m,y}}{a_{m,z}}\right)$$

A ^jel itt azt jelzi, hogy becsült értékről van szó, melynek meghatározásába beleszól minden zavar, rázkódás, zaj

- A fenti képletekkel meghatározott értékek segítségével korrigálhatjuk a giroszkóp adataiból számolt szögeket az elcsúszás (drift) kordában tartására
- Az Inerciális Mérőegységek (IMU) egy tokban giroszkópot és gyorsulásmérőt is tartalmaznak

# Szögek becslése giroszkóppal

- A MEMS giroszkópunk nem szöget, hanem szögsebességet mér
- Legyen  $p$ ,  $q$  és  $r$  a giroszkóp  $x$ ,  $y$ , és  $z$  irányú mérési eredménye, ekkor:

$$\begin{pmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{pmatrix} = \begin{pmatrix} p + q \sin(\phi) \tan(\theta) + r \cos(\phi) \tan(\theta) \\ q \cos(\phi) - r \sin(\phi) \\ q \sin(\phi) / \cos(\theta) + r \cos(\phi) / \cos(\theta) \end{pmatrix}$$

- Ebből az Euler szögek értékét idő szerinti integrálással lehet meghatározni. Legegyszerűbben téglalap közelítéssel

$$\hat{\theta} = \hat{\theta} + \dot{\theta} \cdot T \quad \text{Pitch}$$

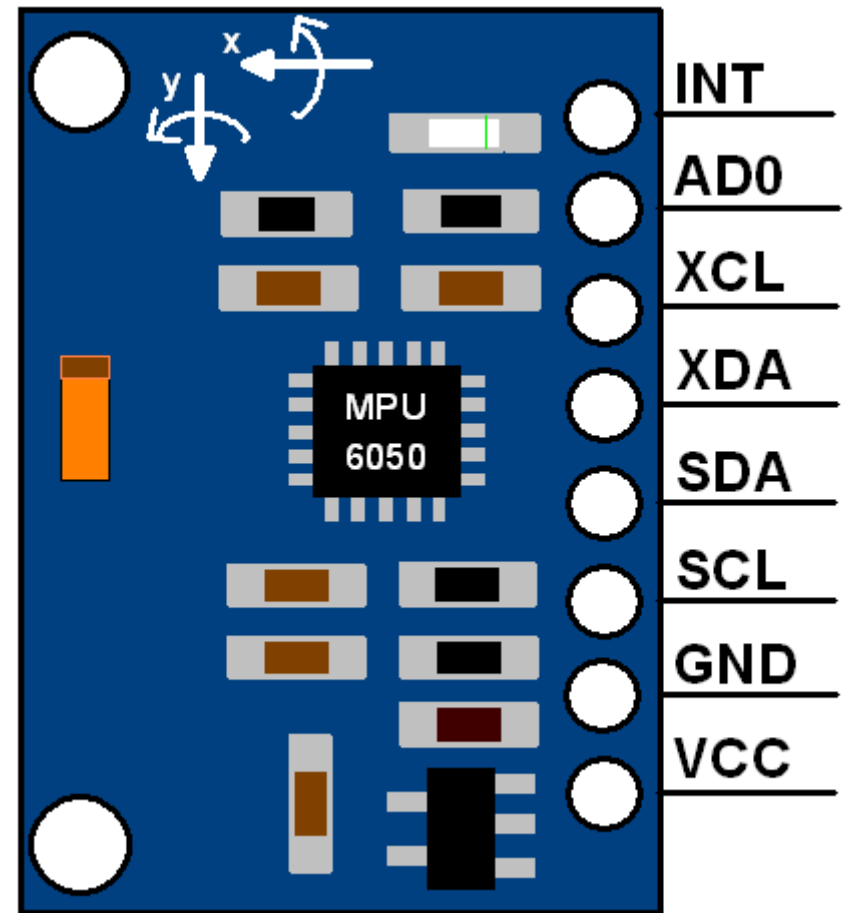
$$\hat{\phi} = \hat{\phi} + \dot{\phi} \cdot T \quad \text{Roll}$$

$$\hat{\psi} = \hat{\psi} + \dot{\psi} \cdot T \quad \text{Yaw}$$

$\Theta = 90^\circ$  esetén nullával való osztás miatt matematikai „gimbal lock” problémába ütközünk!

# Az MPU6050 modul

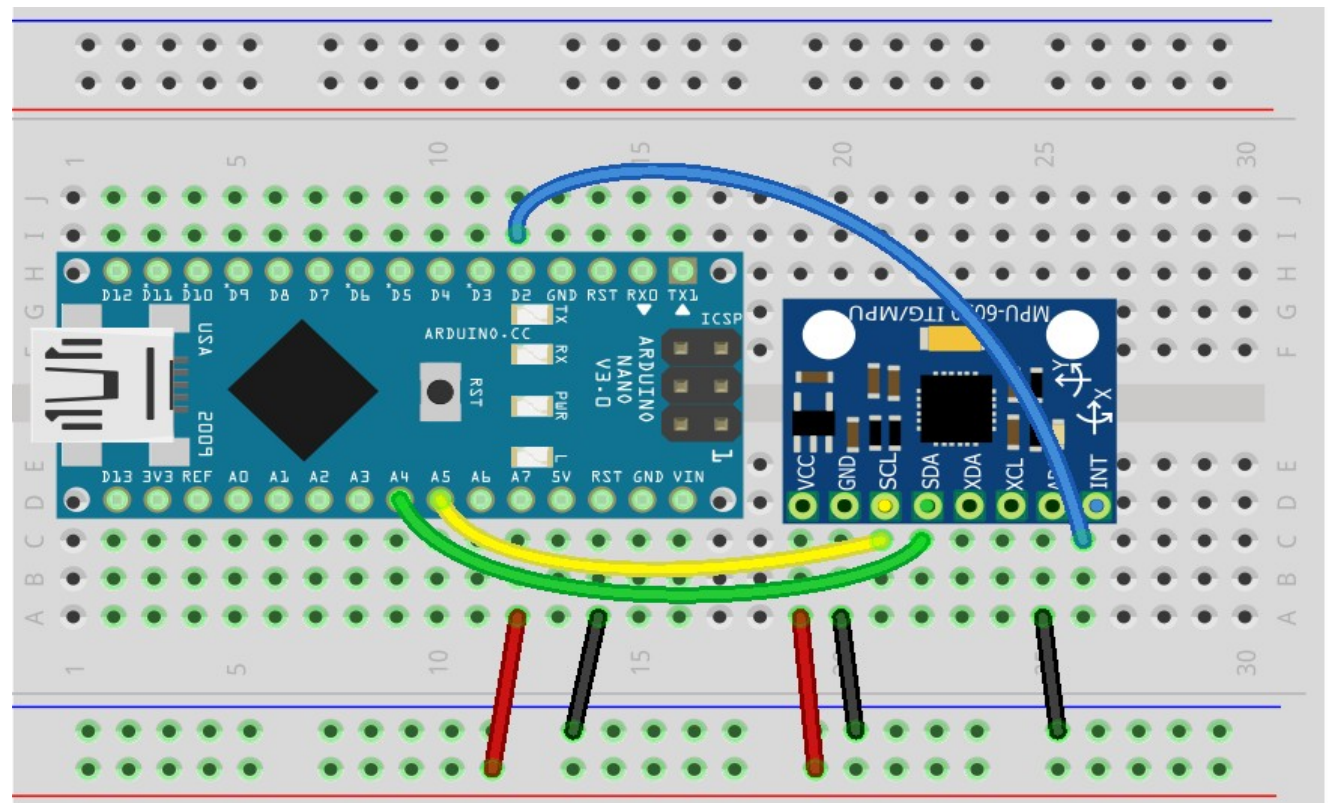
- Gyorsulásmérőt és szögsebességmérőt tartalmaz (6-tengelyű)
  - 16 bites regiszterek, I2C illesztő
  - VCC = 5V, GND = közös pont
  - SCL, SDA = I2C órajel és adat
  - AD0 címvonál (0 vagy 1)
  - INT megszakításkérő jel (jelez, ha van kiolvasható adat)
  - XCL és XDA a szenzor kiterjesztéséhez kell egy mágneses szenzor csatlakoztatásával
- Az adatprocesszor azonmabn nem dokumentált...



# Kapcsolási vázlat

- Az **SDA** és **SCL** lábakat az **A4**, illetve **A5** lábakra kötjük. Az **AD0** legyen **GND**-re kötve, az **INT** kimenetet pedig az Arduino **INT0/D2** bemenetére kötjük. A **VCC** bemenetre +5V-ot kötünk, a kártya saját feszültségstabilizátora állít elő belőle 3,3 V-ot.

Arduino nano v3.0  
és GY-521 kártya  
(MPU6050)  
összekötése



fritzing



# raw\_data.ino

- A nyers adatok kiolvasásához csak bekapcsoljuk az eszközt és kiolvassuk az adatregisztereket
- Az I2C cím 0x68, ha **AD0** 0-ra van kötve, egyébként pedig 0x69

```
#include<Wire.h>
const int MPU6050_addr = 0x68;      // I2C cím
int16_t AccX, AccY, AccZ, Temp, GyroX, GyroY, GyroZ;

void setup() {
  Wire.begin();
  Wire.beginTransmission(MPU6050_addr);
  Wire.write(0x6B);                 // Power Management regiszter címe
  Wire.write(0);                    // 0 beírása törli a tiltó biteket,
  Wire.endTransmission(true);       // azaz bekapcsolja az eszközt
  Serial.begin(9600);               // Soros kommunikáció 9600 bit/s
}
```

Folytatás a következő oldalon ...

# raw\_data.ino

```
void loop() {
  Wire.beginTransmission(MPU6050_addr);
  Wire.write(0x3B);
  Wire.endTransmission(false);
  Wire.requestFrom(MPU6050_addr, 14, true);
  AccX = Wire.read() << 8 | Wire.read();
  AccY = Wire.read() << 8 | Wire.read();
  AccZ = Wire.read() << 8 | Wire.read();
  Temp = Wire.read() << 8 | Wire.read();
  GyroX = Wire.read() << 8 | Wire.read();
  GyroY = Wire.read() << 8 | Wire.read();
  GyroZ = Wire.read() << 8 | Wire.read();
  Serial.print("AccX = "); Serial.print(AccX);
  Serial.print(" || AccY = "); Serial.print(AccY);
  Serial.print(" || AccZ = "); Serial.print(AccZ);
  Serial.print(" || Temp = "); Serial.print(Temp/340.00+36.53);
  Serial.print(" || GyroX = "); Serial.print(GyroX);
  Serial.print(" || GyroY = "); Serial.print(GyroY);
  Serial.print(" || GyroZ = "); Serial.println(GyroZ);
  delay(100);
}
```

# raw\_data.ino futási eredménye

- A gyorsulásmérő, a hőmérő és a szögsebességmérő nyers adatai

```
COM4  
Send  
AccX = 16 || AccY = 32 || AccZ = 16336 || Temp = 28.91 || GyroX = -27 || GyroY = 29 || GyroZ = 0  
AccX = -100 || AccY = 40 || AccZ = 16392 || Temp = 28.81 || GyroX = -3 || GyroY = 26 || GyroZ = 23  
AccX = -28 || AccY = -32 || AccZ = 16356 || Temp = 28.81 || GyroX = -12 || GyroY = 11 || GyroZ = 1  
AccX = -40 || AccY = 32 || AccZ = 16384 || Temp = 28.86 || GyroX = -20 || GyroY = 4 || GyroZ = 21  
AccX = 24 || AccY = -40 || AccZ = 16372 || Temp = 28.81 || GyroX = -20 || GyroY = 27 || GyroZ = -12  
AccX = -4 || AccY = 20 || AccZ = 16388 || Temp = 28.81 || GyroX = -24 || GyroY = 38 || GyroZ = -12  
AccX = 80 || AccY = -8 || AccZ = 16164 || Temp = 28.86 || GyroX = -11 || GyroY = 18 || GyroZ = 2  
AccX = 0 || AccY = 32 || AccZ = 16380 || Temp = 28.81 || GyroX = -4 || GyroY = 17 || GyroZ = 3  
AccX = -20 || AccY = -92 || AccZ = 16176 || Temp = 28.91 || GyroX = -14 || GyroY = 22 || GyroZ = -24  
AccX = 0 || AccY = 20 || AccZ = 16404 || Temp = 28.81 || GyroX = 8 || GyroY = 0 || GyroZ = -14  
AccX = 8 || AccY = -100 || AccZ = 16448 || Temp = 28.81 || GyroX = -22 || GyroY = 14 || GyroZ = 12  
AccX = -60 || AccY = -44 || AccZ = 16300 || Temp = 28.86 || GyroX = -13 || GyroY = 32 || GyroZ = -9  
AccX = 20 || AccY = -24 || AccZ = 16356 || Temp = 28.81 || GyroX = -5 || GyroY = 25 || GyroZ = 10  
AccX = 20 || AccY = 0 || AccZ = 16360 || Temp = 28.72 || GyroX = -11 || GyroY = 27 || GyroZ = 0  
AccX = 28 || AccY = -24 || AccZ = 16388 || Temp = 28.81 || GyroX = 3 || GyroY = 12 || GyroZ = 26  
AccX = -44 || AccY = -88 || AccZ = 16304 || Temp = 28.86 || GyroX = -36 || GyroY = 13 || GyroZ = 4  
AccX = 72 || AccY = 8 || AccZ = 16356 || Temp = 28.91 || GyroX = -13 || GyroY = 36 || GyroZ = 8  
AccX = 132 || AccY = 4 || AccZ = 16336 || Temp = 28.81 || GyroX = -34 || GyroY = -10 || GyroZ = -11  
AccX = -96 || AccY = -132 || AccZ = 16184 || Temp = 28.81 || GyroX = -12 || GyroY = -1 || GyroZ = -6  
 Autoscroll  Show timestamp Newline 9600 baud Clear output
```

# Power management register

- Írjunk nullát a regiszterbe!

## 4.28 Register 107 – Power Management 1 PWR\_MGMT\_1

Type: Read/Write

Register (Hex)	Register (Decimal)	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
6B	107	DEVICE_RESET	SLEEP	CYCLE	-	TEMP_DIS	CLKSEL[2:0]		

### Description:

This register allows the user to configure the power mode and clock source. It also provides a bit for resetting the entire device, and a bit for disabling the temperature sensor.

```
// power management
Wire.beginTransmission(0b1101000); // Start the communication by using address of MPU
Wire.write(0x6B); // Access the power management register
Wire.write(0b00000000); // Set sleep = 0
Wire.endTransmission(); // End the communication
```

# Giroszkóp konfigurációs regiszter

- Írjunk nullát a regiszterbe!

A méréshatár így  $\pm 250$  °/s lesz

FS_SEL	Full Scale Range
0	$\pm 250$ °/s
1	$\pm 500$ °/s
2	$\pm 1000$ °/s
3	$\pm 2000$ °/s

## 4.4 Register 27 – Gyroscope Configuration

### GYRO\_CONFIG

Type: Read/Write

Register (Hex)	Register (Decimal)	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
1B	27	XG_ST	YG_ST	ZG_ST	FS_SEL[1:0]		-	-	-

```
// configure gyro
Wire.beginTransmission(0b1101000);
Wire.write(0x1B); // Access the gyro configuration register
Wire.write(0b00000000);
Wire.endTransmission();
```

# Gyorsulásmérő konfigurációs regiszter

- Írjunk nullát a regiszterbe!

## 4.5 Register 28 – Accelerometer Configuration ACCEL\_CONFIG

Type: Read/Write

Register (Hex)	Register (Decimal)	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
1C	28	XA_ST	YA_ST	ZA_ST	AFS_SEL[1:0]				-

*AFS\_SEL* selects the full scale range of the accelerometer outputs according to the following table.

AFS_SEL	Full Scale Range
0	$\pm 2g$
1	$\pm 4g$
2	$\pm 8g$
3	$\pm 16g$

```
// configure accelerometer
Wire.beginTransmission(0b1101000);
Wire.write(0x1C);      // Access the accelerometer configuration register
Wire.write(0b00000000);
Wire.endTransmission();
```

# A giroszkóp adatok kiolvasása

## ■ A giroszkóp kiolvasása

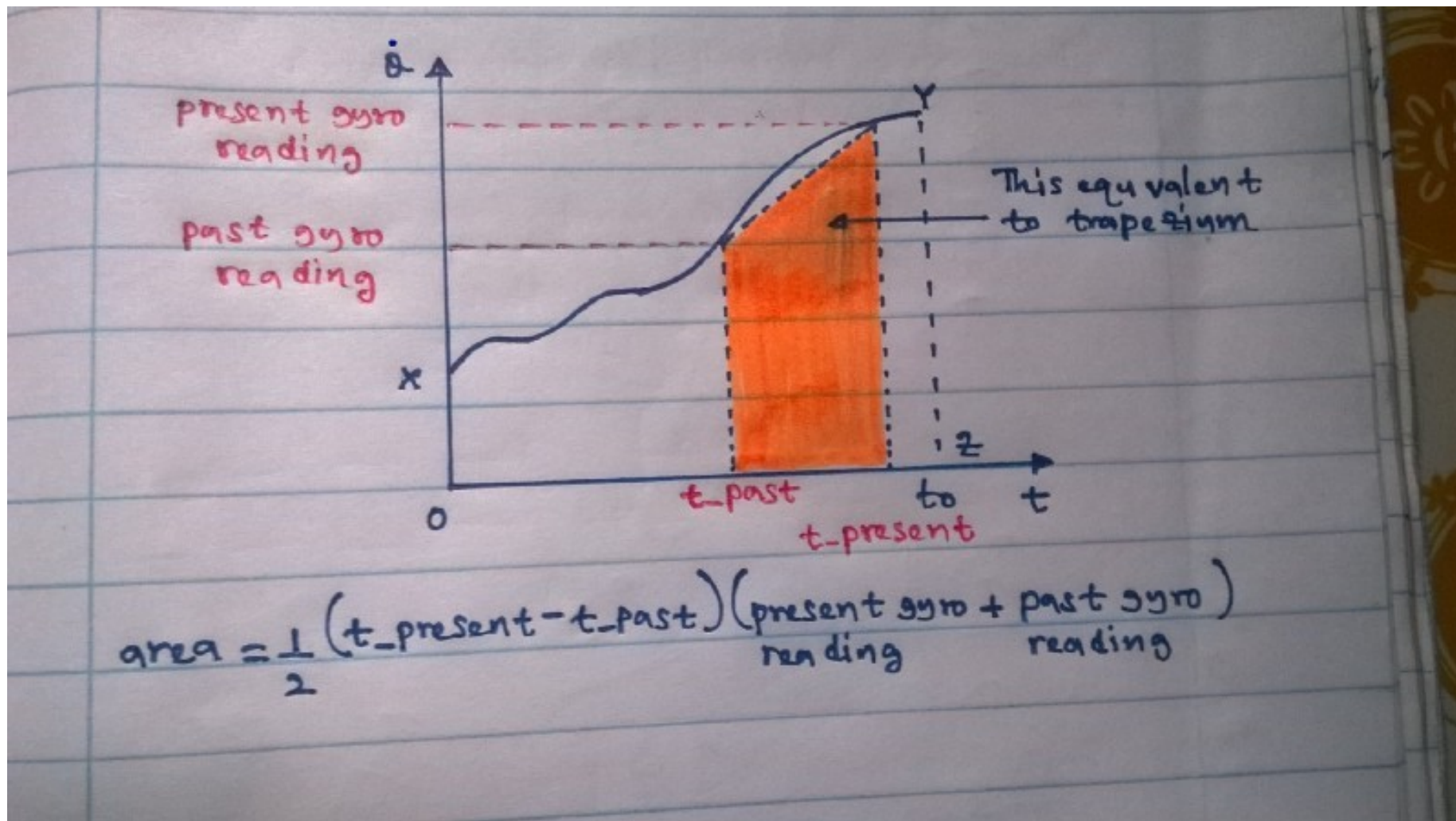
```
void getGyroValues() {  
Wire.beginTransmission(0b1101000); // Start the communication by using address of MPU  
Wire.write(0x43); // Access the starting register of gyro readings  
Wire.endTransmission();  
Wire.requestFrom(0b1101000,6); // Request for 6 bytes from gyro registers (43-48)  
while(Wire.available() < 6); // Wait until all 6 bytes are available  
gyroXPresent = Wire.read()<<8|Wire.read(); // Store first two bytes into gyroXPresent  
gyroYPresent = Wire.read()<<8|Wire.read(); // Store next two bytes into gyroYPresent  
gyroZPresent = Wire.read()<<8|Wire.read(); //Store last two bytes into gyroZPresent  
}
```

- Ezeket a nyers adatokat beosztjuk az érzékenységi faktoral (131), így megkapjuk a szögsebességek adatait:

```
void getAngularVelocity() {  
    rotX = gyroXPresent / 131.0;  
    rotY = gyroYPresent / 131.0;  
    rotZ = gyroZPresent / 131.0;  
}
```

# A szögsebesség integrálása trapéz képlettel

- Mivel a szögsebesség nem állandó, az integrálásnál pontosabb közelítést ad a trapéz képlet



<https://medium.com/@kavindugimhanzoysa/lets-work-with-mpu6050-gy-521-part1-6db0d47a35e6>



# A szögsebesség integrálása trapéz képlettel

- Ha az előző mérés timePast időpontban, az új mérés timePresent időpontban történt, akkor az előző és az új szögsebesség értékekből így kapjuk meg a szögeket:

```
void calculateAngle() {  
  // same equation can be written as  
  // angelZ = angelZ+((timePresentZ-timePastZ)*(gyroZPresent+gyroZPast-2*gyroZCalli))/(2*1000*131);  
  // 1/(1000*2*131) = 0.00000382  
  // 1000 --> convert milli seconds into seconds  
  // 2 --> comes when calculation area of trapezium  
  // substracted the callibated result two times because there are two gyro readings  
  angelX = angelX + ((timePresent - timePast)*(gyroXPresent + gyroXPast - 2*gyroXCalli)) * 0.00000382;  
  angelY = angelY + ((timePresent - timePast)*(gyroYPresent + gyroYPast - 2*gyroYCalli)) * 0.00000382;  
  angelZ = angelZ + ((timePresent - timePast)*(gyroZPresent + gyroZPast - 2*gyroZCalli)) * 0.00000382;  
}
```

- A teljes program **MPU\_6050.ino** néven található  
GitHub link: [MPU\\_6050.ino](#)

# MPU\_6050.ino futási eredmény

- A szenzort a Z tengely mentén jobbra forgattuk kb 90 fokkal

```
COM5
Send
Angular displacement wrt started position (deg)
angle of X axis=-15.37 angle of Y axis=28.56 angle of Z axis=89.86
Acceleration (g) Ax=-0.02 Ay=-0.01 Az=1.02
-----
Gyro (deg/sec) X=9.19 Y=-0.17 Z=-77.47
Angular displacement wrt started position (deg)
angle of X axis=-13.86 angle of Y axis=28.52 angle of Z axis=76.70
Acceleration (g) Ax=0.00 Ay=0.02 Az=1.00
-----
Gyro (deg/sec) X=20.73 Y=1.48 Z=-198.50
Angular displacement wrt started position (deg)
angle of X axis=-9.97 angle of Y axis=28.71 angle of Z axis=40.79
Acceleration (g) Ax=-0.04 Ay=0.03 Az=1.01
-----
Gyro (deg/sec) X=8.66 Y=2.36 Z=-85.10
Angular displacement wrt started position (deg)
angle of X axis=-6.13 angle of Y axis=29.22 angle of Z axis=3.75
Acceleration (g) Ax=-0.00 Ay=-0.04 Az=1.00
-----
Gyro (deg/sec) X=0.24 Y=-0.51 Z=-1.40
Angular displacement wrt started position (deg)
angle of X axis=-4.98 angle of Y axis=29.47 angle of Z axis=-7.46
Acceleration (g) Ax=-0.01 Ay=-0.01 Az=1.00
-----
Gyro (deg/sec) X=-0.47 Y=-0.49 Z=14.44
Angular displacement wrt started position (deg)
angle of X axis=-5.01 angle of Y axis=29.35 angle of Z axis=-5.76
Acceleration (g) Ax=0.02 Ay=-0.02 Az=0.99
```