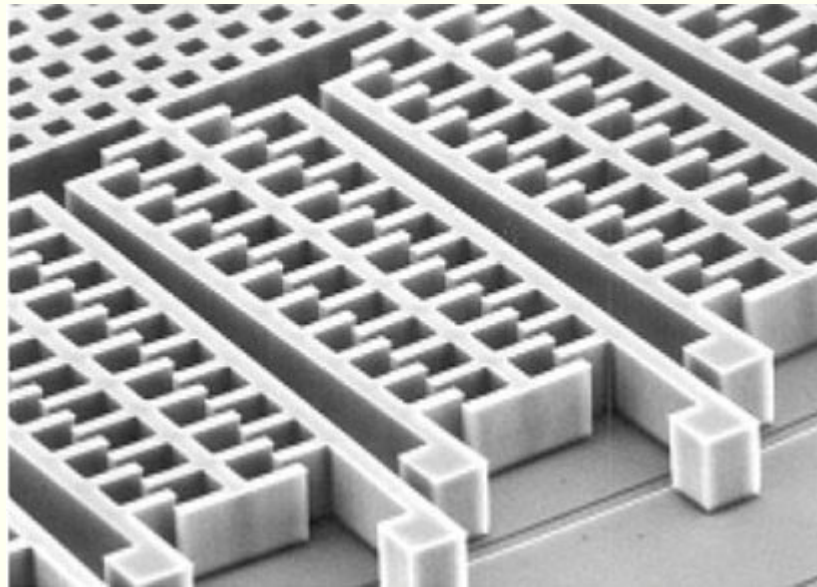


MEMS érzékelők, szenzorok, méréstechnika...

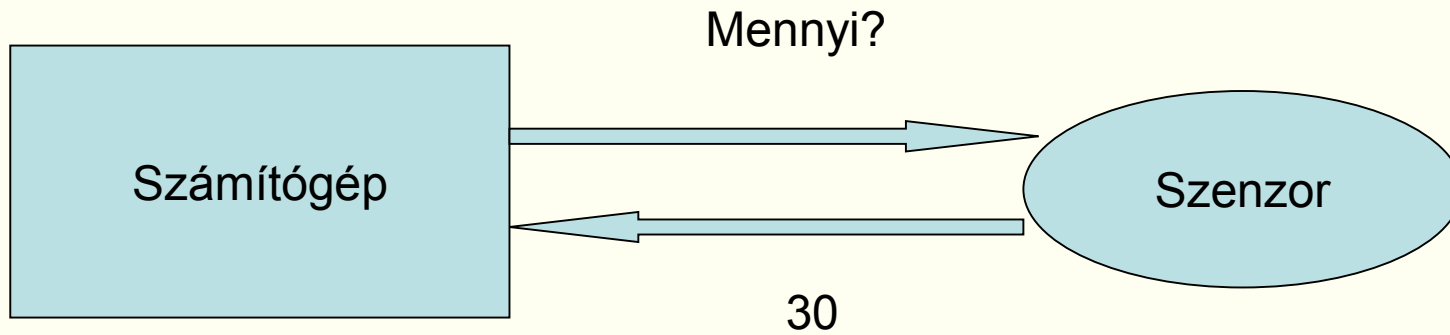
Horváth Gábor

ELTE / Informatikai Igazgatóság



MEMS érzékelők, szenzorok, mérés technika...

Mérni a legegyszerűbb, a rendszerhez kapcsolunk egy szenzort:



-Mi 30?

-Mennyire pontosan 30?

Minden döntés, beavatkozás annyira pontos, amennyire a kiinduló adatok.



MEMS érzékelők, szenzorok, mérés technika...

A legegyszerűbb eset: hőmérséklet mérés

(Egy hőmérséklet-függő ellenálláson áramgenerátorral ismert erősségű egyenáramot folyatok át és mérem a rajta eső feszültséget - pl. AD konverterrel, amit már jól tudok a rendszeremhez illeszteni.)

6 gyártó 8 terméke: 22-től 26 fokig minden volt a szobában...

Nagyon olcsó szenzor: a mért érték monoton nő a fizikai mennyiséggel

Olcsó szenzor: az arányosság lineáris, de 0-ban nem biztos, hogy 0-t ad vissza: kalibrálni kell

Drágább szenzor: a kalibrációs értékek egyből betölthetők

Még drágább szenzor: a kalibrálást gyárilag elvégezték.



MEMS érzékelők, szenzorok, méréstechnika...

Milyen hibákkal találkozhatunk:

- Mérési pontosság (pl. 16 bitet ad vissza, de abból csak 12 értékes)
- Nullponti hiba: 0-s mennyiségnél nem 0 a kapott érték
- Nemlinearitás: egy korrekciós függvényt kell alkalmazni (ezt minőségi szenzorokba gyárilag beépítik)
- Hőmérséklet függés (ezt mindig illik a gyártónak beépíteni)
- Memória effektus: a mért érték hibája függ az előzőleg mért értéktől
- Orientációs hiba: a mért érték függ a szenzor helyzetétől is
- ... a szenzor és a mért mennyiség függvényében egyéb hibák



MEMS érzékelők, szenzorok, méréstechnika...

Mit tudunk tenni a hibákkal?

1. Nézzünk utána, hogy mi a szenzorunk gyártó által megadott pontossága és a többi bitet dobjuk el, ne próbáljuk meg feldolgozni!
2. Ha van lehetőségünk, akkor átlagoljunk (aluláteresztő szűrő)
3. Nézzünk utána, hogy kell-e kalibrálnunk! Ha igen, akkor kaptunk-e autokalibrációs eljárást? Hívjuk meg. Ha nem kaptunk, van-e kézi kalibráció, mit kell csinálnunk a szenzorral? El tudjuk-e menteni a kalibrációs adatokat?



MEMS érzékelők, szenzorok, méréstechnika...

Mi a helyzet a mérés idejével?

1, A mérés időbe telik. Ha sok szenzorral sokat mérünk, akkor ezt már okosan kell csinálni, nem várhatjuk végig egymás után, amíg a mérés zajlik. Elindítjuk a méréseket párhuzamosan és figyeljük (pl. interrupt) hogy rendelkezésre áll-e már valamelyik szenzorban az eredmény.

2, Összetartozó mennyiségeket illik egyszerre mérni és nem egymás után – mert akkor már nagyon különböző időpillanatokhoz tartoznak.

3, Mi történik, ha gyorsabban olvasom ki az eredményt, mint ahogy a szenzor elkészül a méréssel?

a, az előző értéket kapom

b, 0 értéket kapok

c, véletlenszerű értéket kapok

d, a mért értéket kapom, de kisebb pontossággal



MEMS érzékelők, szenzorok, méréstechnika...

Hogyan illesztem a szenzort? I.

1. ötlet: úgyis mindig feszültséget mérek az AD konverteremmel: a mért mennyiséget valahogy analóg egyenfeszültséggé alakítom és azt vezetem oda a mikrovezérlőmhöz.

Rossz ötlet: a tapasztalat szerint egy 10 bit pontos értékből legalább 1 bitet bukok a 20 centi drót által összeszedett zavaron és azon, hogy külső AD átalakítót használok.

Jó ötlet: a fizikai szenzorhoz minél közelebb kell digitális jellé alakítani a mennyiséget és digitális interface-en vezetni a mikrovezérlőhöz.

Gyakori megoldások: I2C, SPI, UART

Kivétel: 4-20mA hurok (vezetékhozz független, de nem gyors és pontos)



MEMS érzékelők, szenzorok, mérés technika...

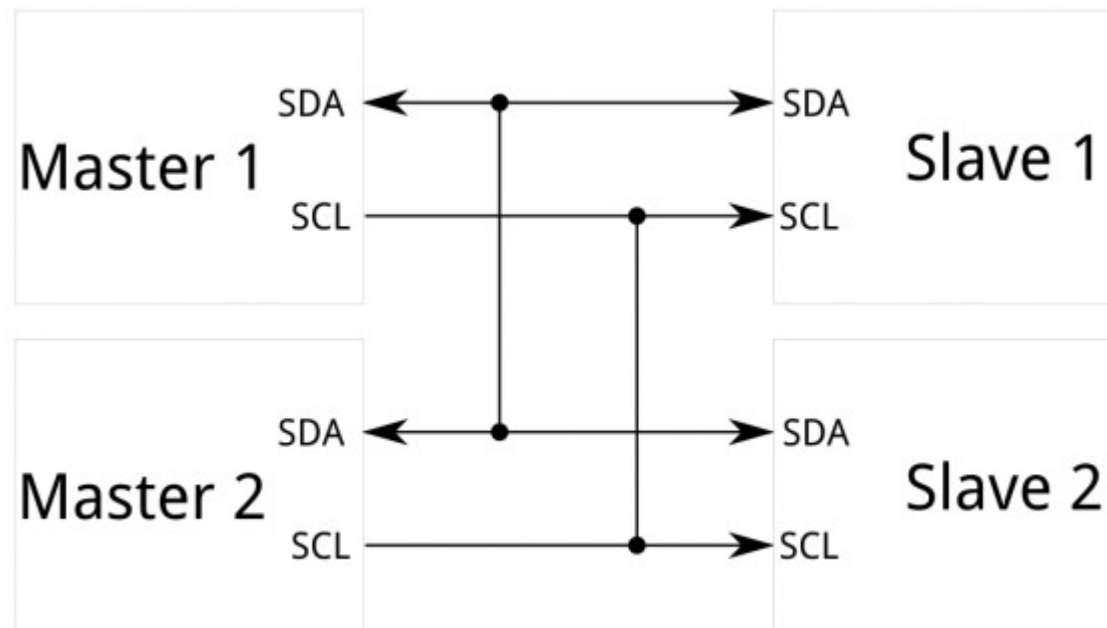
I2C: Inter-Integrated Circuit Protocol

I2C: busz topológia, egyszerre 1 aktív Master, sok Slave

2 vezetékünk van (a tápfeszültségen és a földvezetéken kívül):

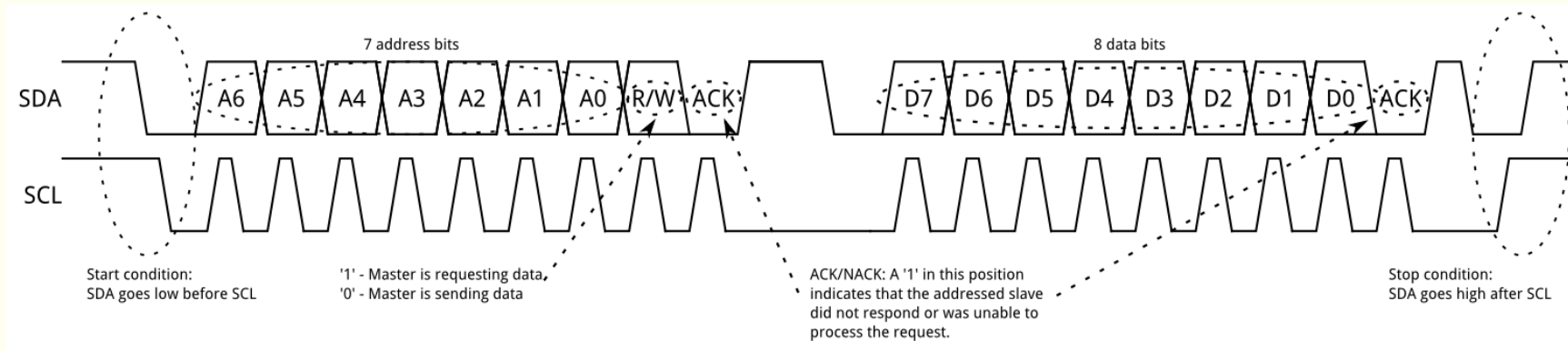
-SDA: ezen megy az adat bitenként

-SCL: ezen megy az órajel (mindig a Master egység adja)



MEMS érzékelők, szenzorok, mérés technika...

I2C tranzakció (cím kiküldése és adat kiküldése):



- mindig 8 bitet küldünk egy keretben, az első keretet mindig a Master küldi
- az első keret első 7 bitje a cím, az utolsó a kommunikációs irány
- az adat jelentése a Slave típusától függ (felsőbb szint)

Van ACK/NACK bit a keret végén: mindig a receiver „billenti be”
(pl. tudja a Master, hogy a Slave él azon a címen)

Nem illik 32-nél több egységet használni egy fizikai buszon.

Általában a Slave-ek címe jumperekkel állítható, lehet több azonos típusú IC.
Half-duplex átvitel, 100 kHz vagy 400 kHz SCL frekvencia (5 MHz?)



MEMS érzékelők, szenzorok, mérés technika...

I2C tranzakció programkódban (Arduino környezet):

```
#include <Wire.h>    (Wire/Wire1 objektum)
```

```
Setup(): Wire.begin();
```

1 byte kiírása:

```
Wire.beginTransaction(i2c_address);  
Wire.write(data)  
if (Wire.endTransaction()) ...
```

N byte beolvasása:

```
Wire.requestFrom(i2c_address, no_of_bytes);  
Wire.readBytes(buffer, no_of_bytes)
```



MEMS érzékelők, szenzorok, mérés technika...

I2C tranzakció programkódban
(Linux környezet):

```
#include <linux/i2c.h>
#include <linux/i2c-dev.h>
#include <sys/ioctl.h>
```

```
uint8_t data, addr = 0x76, reg = 0x0d;
const char *path = "/dev/i2c-3";
int file, rc;
```

```
file = open(path, O_RDWR);
rc = ioctl(file, I2C_SLAVE, addr);
```

```
data = i2c_read_byte_data(file, reg);
```

```
static inline __s32 i2c_access(int file, char
read_write, __u8 command, int size, union
i2c_smbus_data *data)
```

```
{
struct i2c_smbus_ioctl_data args;
```

```
args.read_write = read_write;
args.command = command;
args.size = size;
args.data = data;
return ioctl(file, I2C_SMBUS, &args);
}
```

```
static inline __s32 i2c_read_byte_data
(int file, __u8 command) {
union i2c_smbus_data data;
if (i2c_access(file, I2C_SMBUS_READ,
command, I2C_SMBUS_BYTE_DATA, &data))
return -1; else return 0xFF & data.byte;
}
```

MEMS érzékelők, szenzorok, méréstechnika...

SPI: Serial Peripheral Interface: csillag topológia, 1 master, több slave

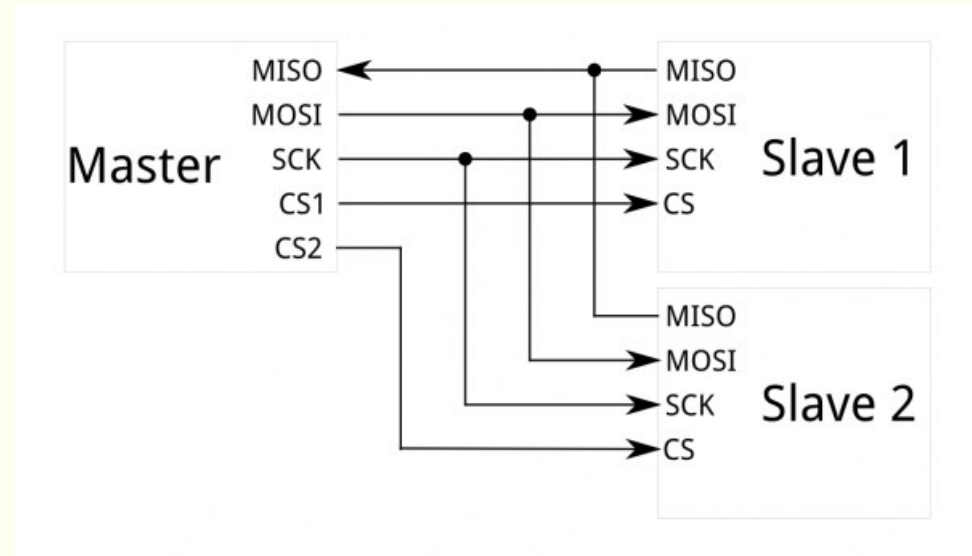
4 vezetékünk van:

-MOSI: master out, slave in
(közös minden slave esetén)

-MISO: master in, slave out
(közös minden slave esetén)

-SCK: mindig a master adja

-CS: chip select: minden slave-hez külön vezeték, ezzel választja ki a master, hogy melyik slave-vel beszél.



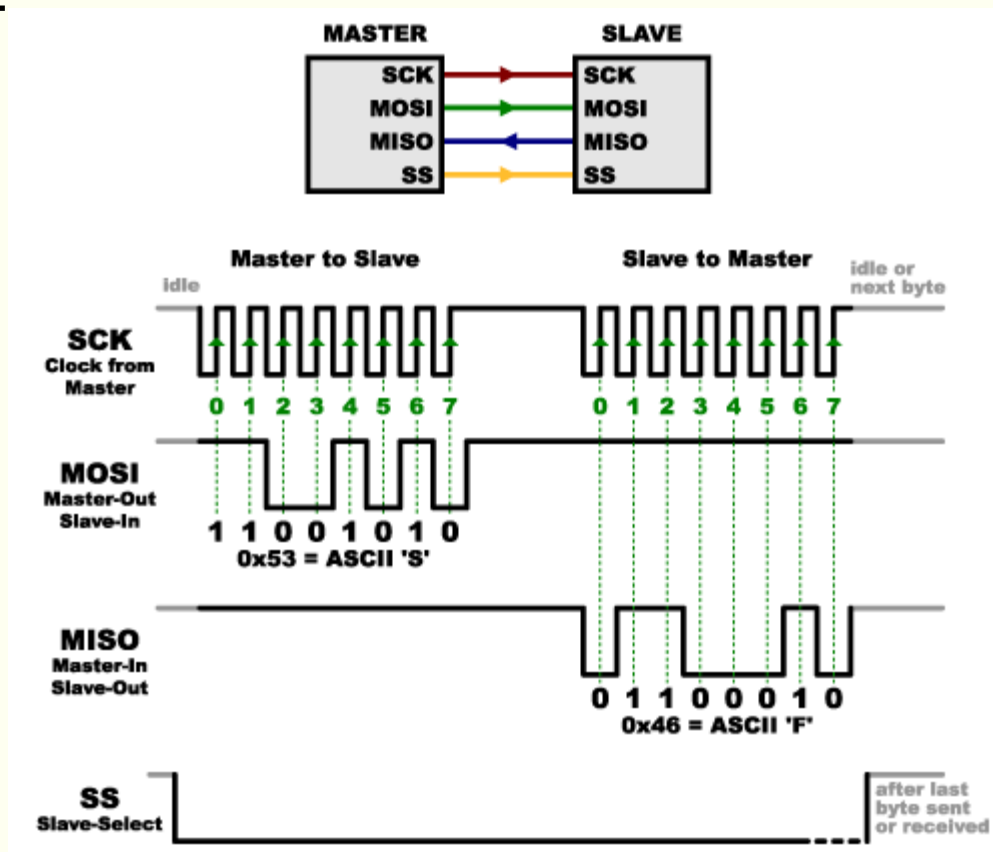
hogy melyik slave-vel beszél. A kapcsolatot full-duplex.

Az SCK fix értékek közül választható, a 4 Mhz-t mindenkinek támogatnia kell, egy korszerű mikrovezérlő 16-20 Mhz-s SPI-t is tud használni.



MEMS érzékelők, szenzorok, mérés technika...

SPI tranzakció:





MEMS érzékelők, szenzorok, méréstechnika...

SPI tranzakció programkódban (Arduino platform):

```
#include <SPI.h>          (lett SPI objektum)
SPI.begin();
```

```
digitalWrite(SSPin, LOW);
result = SPI.transfer(data);
digitalWrite(SSPin, HIGH);
```

Az ARM platform már másképp kezeli az SS lábakat:

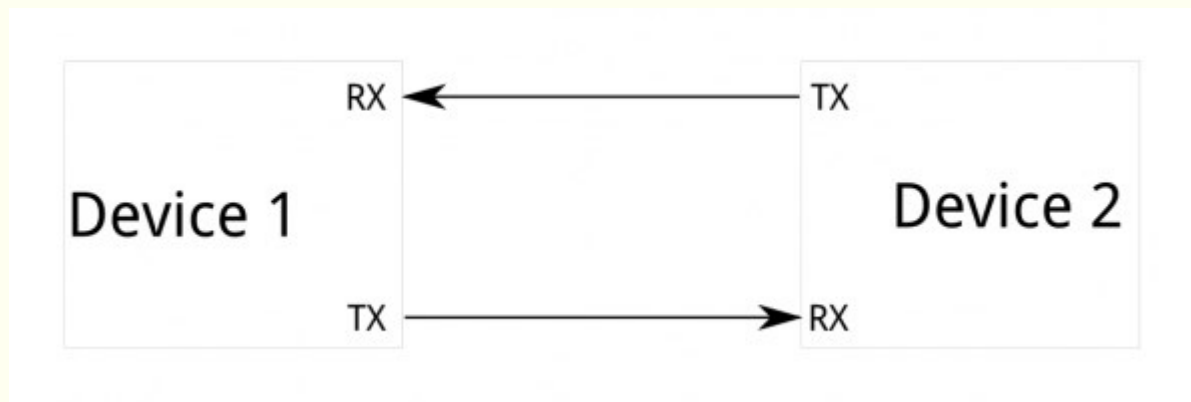
```
SPI.begin(SSPin);
SPI.setClockDivider(SSPin, 21);
byte receive_data = SPI.transfer(SSPin, send_data, SPI_CONTINUE);
```

(Nincs 3. argumentum, ha az utolsó tranzakciót csináljuk.)

MEMS érzékelők, szenzorok, mérés technika...

UART: Universal Asynchronous Receiver/Transmitter

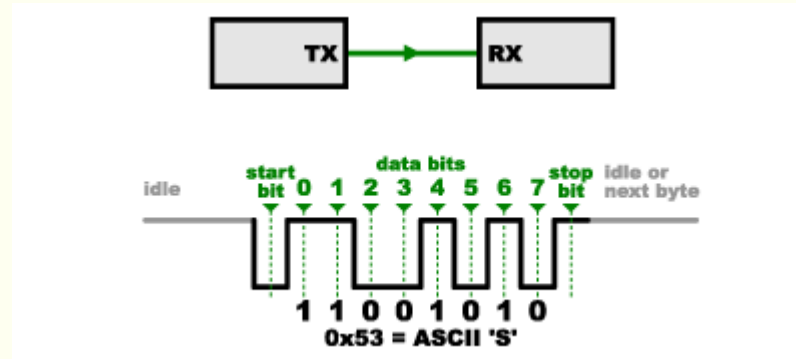
Órajel nélküli soros kapcsolat, TX és RX (keresztbe kapcsolva), duplex átvitel előre beállított átviteli sebességgel – csak két egység között.



- A két oldalt azonos átviteli sebességre kell állítani (max 230400 bps)
- A bufferelés általában minimális (RingBuffer – 128 byte a default) tehát tartósan egyik oldalt sem szabad gyorsabban „tömni” adattal, mint ahogy az UART kapcsolatba tudja tölteni, vagy a mikrovezérlő felé át tudja adni.

MEMS érzékelők, szenzorok, mérés technika...

UART tranzakció:



Az átvitel aszinkron: a két oldal órájának pontosan együtt kell járnia. A szinkronizálás miatt start és stop bit kerül a keretbe, ez plusz 25% overhead az átvitelben és csak 300K-ig működik gazdaságosan.

Az átvitel full-duplex: mindkét irányban egyszerre történhet.

- Mind a két oldalon UART chip
- Mind a két oldalon TX és RX buffer (állítható)
- 1 byte adása és vétele hardware funkció (de lehet szoftveresen is...)



MEMS érzékelők, szenzorok, mérés technika...

UART tranzakció programkódban:

Arduino: Serial objektum

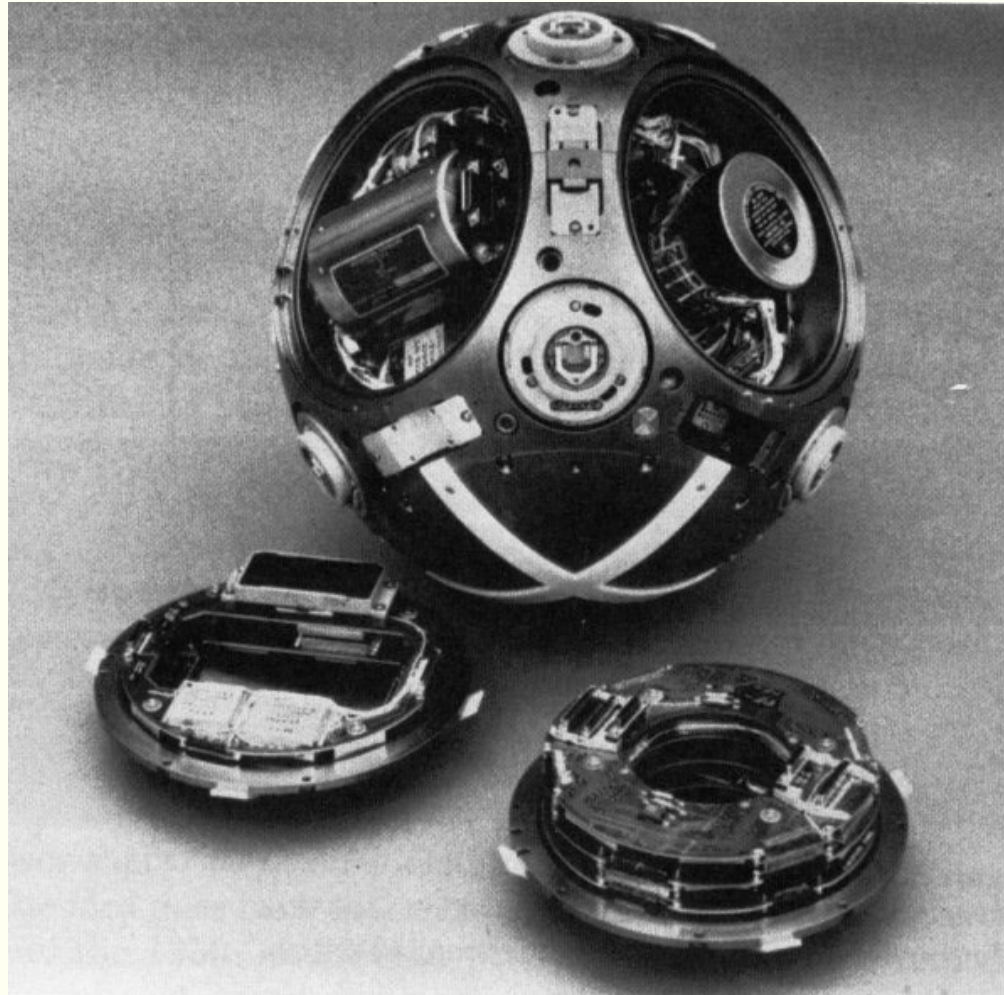
```
Serial.begin(38400);
```

```
Serial.write(data);
```

```
If (Serial.available()) data = Serial.read();
```

MEMS érzékelők, szenzorok, mérés technika...

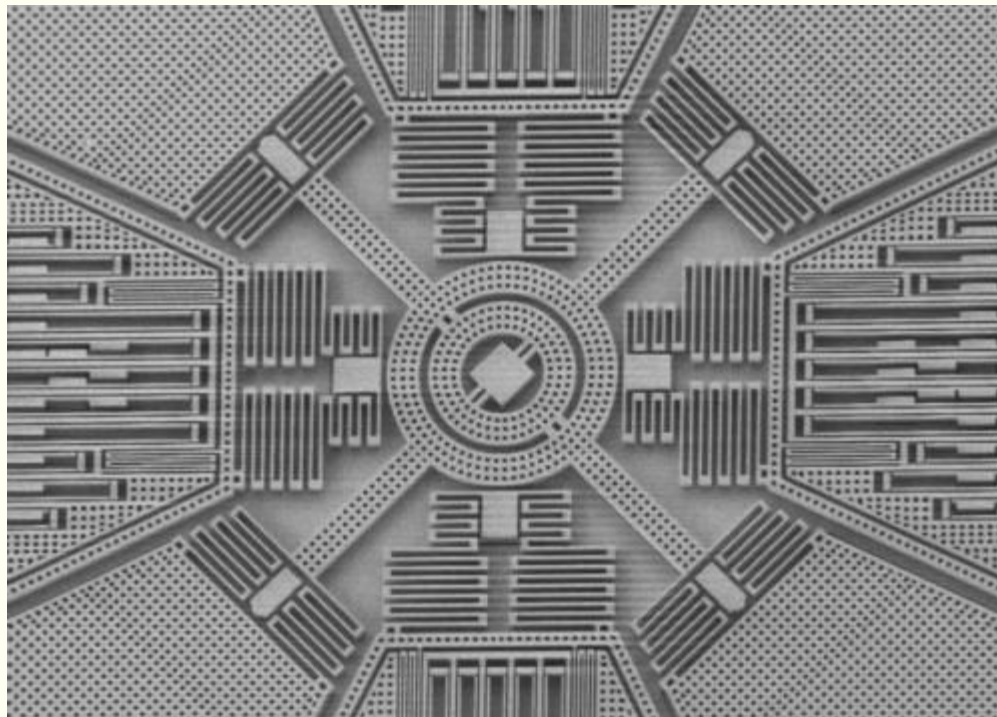
A szenzorok múltja – pörgettyűs giroszkóp



MEMS érzékelők, szenzorok, méréstechnika...

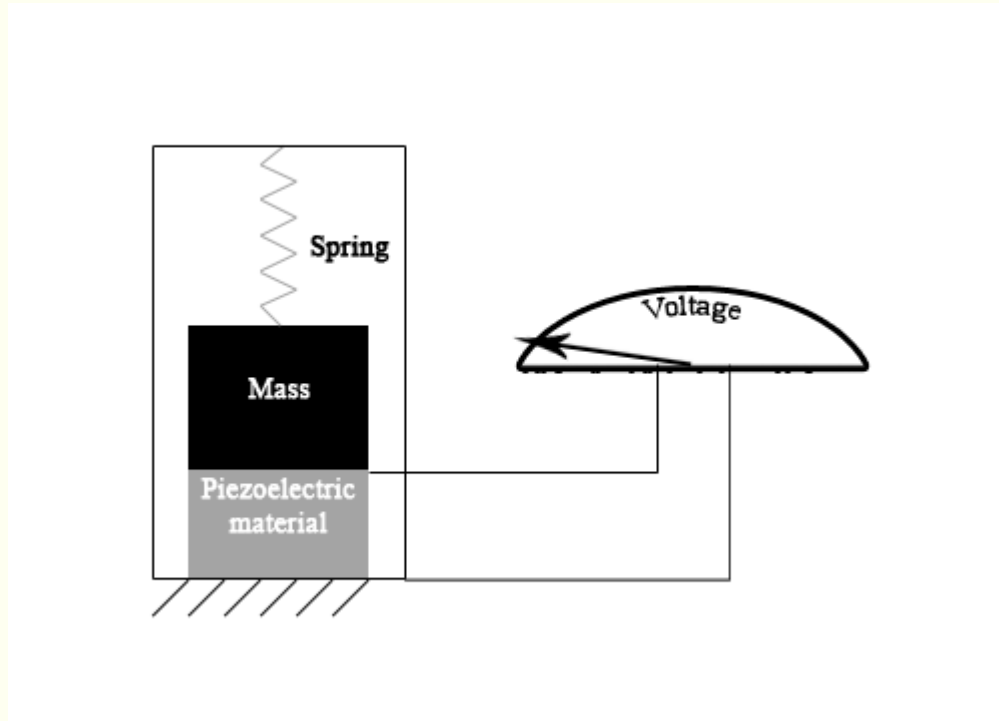
Szenzorok ma: MEMS – Micro Electro Mechanic System

A fizikai érzékelők (sokszor mozgó elemekkel) a félvezető lapon vannak megvalósítva olyan építőelemekből és technológiával, mint a félvezető rétegek és kapcsolatok.



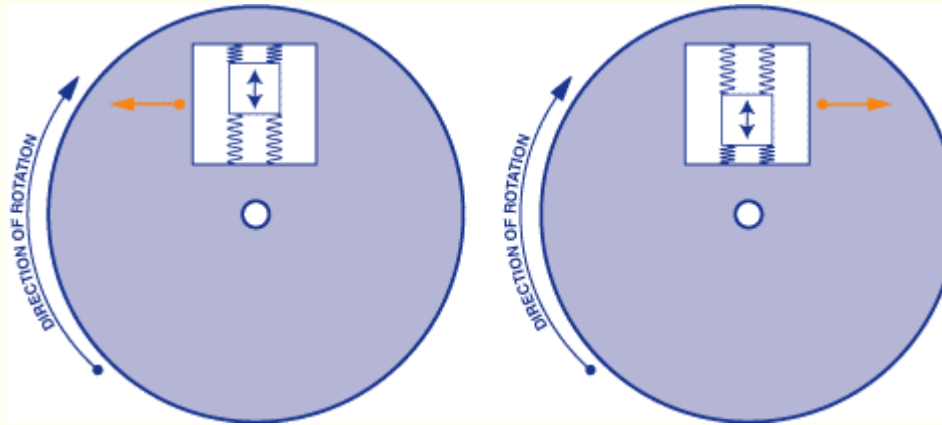
MEMS érzékelők, szenzorok, mérés technika...

Gyorsulásmérő szenzor felépítése (valójában erőt mérünk)



MEMS érzékelők, szenzorok, mérés technika...

Giroszkóp felépítése:

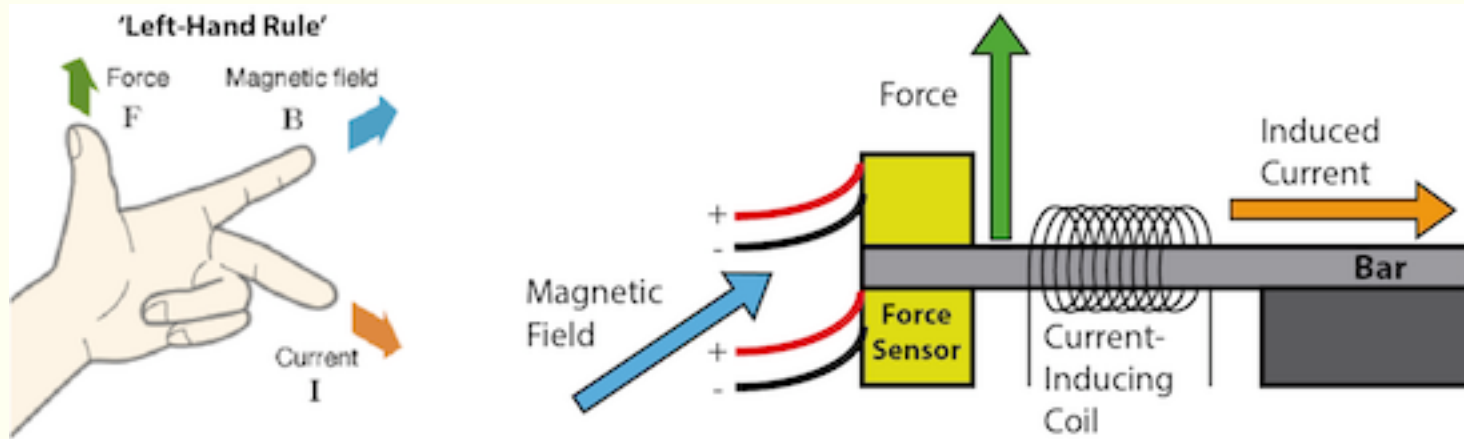


Centrifugális erőt mérünk – lineáris erőmérő egy tárcsára szerelve.

Párban vannak telepítve, hogy a lineáris gyorsulást ne mérjék bele – a tengely két oldalán elhelyezkedők jeleit kivonják egymásból.

MEMS érzékelők, szenzorok, mérés technika...

Magnetométer működési elve



Lorentz-erő: a mágneses térre merőleges folyó áram vezetőjére a síkra merőleges erő hat, aminek mértéke arányos a mágneses térrel.



MEMS érzékelők, szenzorok, méréstechnika...

Az orientáció problémája: össze-vissza áll a kezünkben a műszer

Abszolút koordináta rendszer:

- a gravitációs vektorra merőleges sík a vízszintes síkja
- ebben van egy kitüntetett irány, a mágneses északi irány

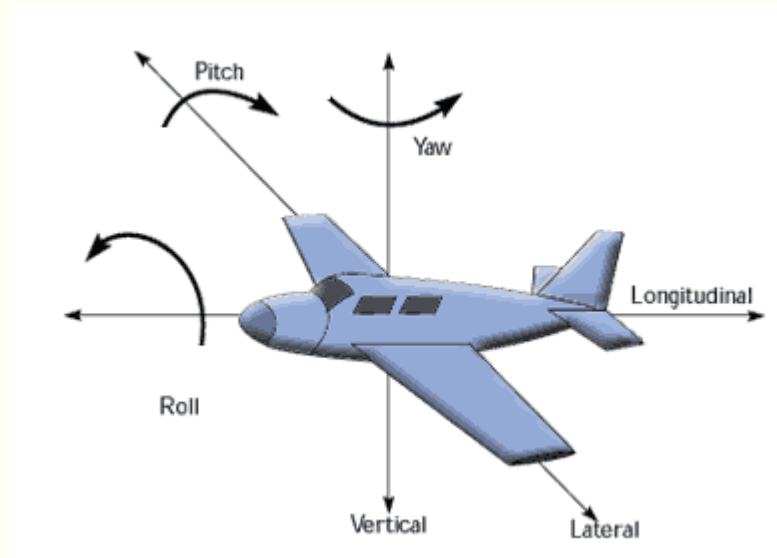
A gravitációs vektor irányát gyorsulásmérővel (axelerometer) tudjuk meghatározni.

A mágneses északi irányt magnetométerrel tudjuk meghatározni.

Aztán minden erőmérő szenzor adatából ki kell vonni a gravitáció hatását és be kell forgatni az abszolút koordinátarendszerünkbe.

MEMS érzékelők, szenzorok, mérés technika...

Orientáció: Euler szögekkel szokták megadni:



A rotáció nem kommutatív művelet: az egy lépésben történő, erőforrástakarékos transzformációhoz quaternio-kat (4 dimenziós komplex vektorokat) használunk. Ahol a hardware ezt meg tudja csinálni:

MPU – Motion Processor Unit

IMU – Inertial Measurement Unit



MEMS érzékelők, szenzorok, méréstechnika...

Végül a hálózobatitkok (milyen szenzorokat használok):

Mérendő fizikai mennyiség: -----	IC neve/száma: -----	Csatlakozás: -----
Hőmérséklet	LM75A	I2C (8 cím!)
Hőmérséklet, páratartalom	HDC1000	I2C (4 cím!)
Légnyomás	BMP180	I2C
Légnyomás, pára, hőmérséklet	BME280	I2C
DC áramerősség	INA219	I2C
Orientáció (9 DOF)	BNO055	I2C
Mágneses tér (iránytű)	LSM303DLHC	I2C
Gyorsulás	ADXL345	I2C
Gyorsulás	LIS3DSH	I2C
Megvilágítás	MLX75305	analóg
Veszélyes gáz	MQ135	analóg
Gyors AD konverzió	MCP320x	SPI



MEMS érzékelők, szenzorok, mérés technika...

Köszönöm a figyelmet!

hg@elte.hu