

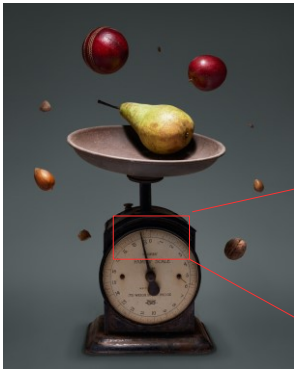
MEMS Intelligent Sensors and Actuators

# (MEMS) Inteligentné senzory a aktuátory

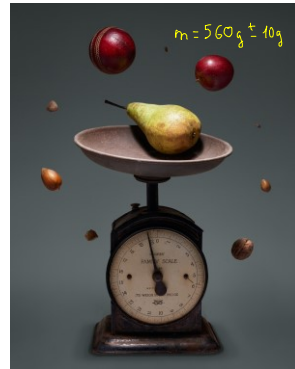
Ing. Richard Balogh



Základy merania



20  
delos  
19,45 ? jednotky? ⇒ 19,45oz  
(uncia)  
  
m = 560 gramov



## Meranie

-- je činnosť, ktorej cieľom je stanoviť hodnotu skúmanej (fyzikálnej) veličiny.

Výsledok merania:

**hodnota + jednotka ± presnosť**  
↳ 560 g ± 1g

Výsledok merania nie je správny, ak nie je určená aj jeho chyba! Pri jej určení vychádzame z metódy merania

## Medzinárodná sústava jednotiek SI Système International (d'Unités)

skr.	názov	pôvod	hodnota	hodnota	názov
Y	yotta	lat. yota = cosen	10 <sup>24</sup> (1000 <sup>8</sup> )	1 000 000 000 000 000 000 000 000	kvadrilion
Z	zetta	lat. zetta = sedem	10 <sup>21</sup> (1000 <sup>7</sup> )	1 000 000 000 000 000 000 000	triliona
E	etta	gr. εβήτα = hexadeca = šestnásť	10 <sup>18</sup> (1000 <sup>6</sup> )	1 000 000 000 000 000 000	bilión
P	petta	gr. πεντάκις = päťkrát	10 <sup>15</sup> (1000 <sup>5</sup> )	1 000 000 000 000 000	stardard
T	tera	gr. τέρας, téras = terakia = štyrikrát	10 <sup>12</sup> (1000 <sup>4</sup> )	1 000 000 000 000	bilión
G	giga	gr. γίγας, gígas = obrovský	10 <sup>9</sup> (1000 <sup>3</sup> )	1 000 000 000	milióna
M	mega	gr. μέγας, mégas = veľký	10 <sup>6</sup> (1000 <sup>2</sup> )	1 000 000	stísa
k	kilo	gr. χίλιος, chílios = tisíc	10 <sup>3</sup>	1 000	tisíc
h	hecto	gr. ἑκατόν, hekátón = sto	10 <sup>2</sup>	100	sto
da	deka	gr. δέκα, déka = desať	10 <sup>1</sup>	10	desať
—	—	—	10 <sup>0</sup>	1	jedni
d	deci	lat. decimus = desiatina	10 <sup>-1</sup>	0,1	desiatina
c	centi	lat. centesimus = stotina	10 <sup>-2</sup>	0,01	stotina
m	milli	lat. millesimus = tisícina	10 <sup>-3</sup>	0,001	tisícina
μ	micro	gr. μικρός, mikros = malý	10 <sup>-6</sup> (1000 <sup>-3</sup> )	0,000 001	miliónina
n	nano	gr. νᾶνος, nános = nánik	10 <sup>-9</sup> (1000 <sup>-3</sup> )	0,000 000 001	biliónina
p	piko	lat. picus = malý	10 <sup>-12</sup> (1000 <sup>-4</sup> )	0,000 000 000 001	biliónina
f	femto	skand. femton = päťdesiat	10 <sup>-15</sup> (1000 <sup>-5</sup> )	0,000 000 000 000 001	biliónina
a	atto	skand. atton = osemdesiat	10 <sup>-18</sup> (1000 <sup>-6</sup> )	0,000 000 000 000 000 001	biliónina
z	zepto	lat. septem = sedem	10 <sup>-21</sup> (1000 <sup>-7</sup> )	0,000 000 000 000 000 001	biliónina
y	yotta	lat. yota = cosen	10 <sup>24</sup> (1000 <sup>8</sup> )	1 000 000 000 000 000 000 000 001	kvadriliona



## Metódy merania

**absolútne** – poskytuje priamo údaj o veličine so známou jednotkou  
napr. 7,5 kV 12,4 cm 1,2 μA

**relatívne** – porovnáme skúmaný objekt s tzv. etalónom  
(normál - objekt so známou hodnotou veličiny).  
napr. meranie závažími

## Metódy merania

**subjektívne** – bez použitia meradla, založené na zmyslovom vnímaní  
napr. kontrola farby, škrabancov...

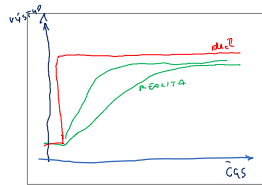
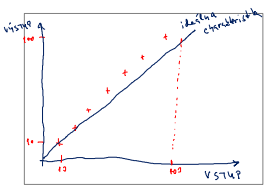
**objektívne** – pomocou meradla  
napr. meranie odporu multimetrom



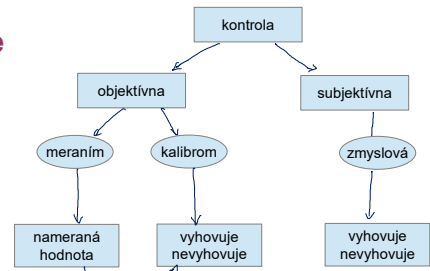
## Metódy merania

**statické** – meria sa ustálená hodnota, počkáme na ustálenie

**dynamické** – záznam meniacej sa veličiny v čase



## Postup pri kontrole výrobkov



## Chyby merania

**systematické** (presnosť meracieho prístroja a metódy merania)

**náhodné** (náhodné poruchové vplyvy - teplota, tlak; zmysly...)

**hrubé** (nepozornosť, porucha prístroja)

5,999V ± 1% do 600 + 2 číslice

Presnosť merania napätia DC ±(1% + 2 číslice)

Ak zvolíme vhodnejší rozsah 6V:

$$5,999V \pm \left( \frac{1\% \cdot 600V}{10} + 2 \text{ číslice} \right) = \pm (0,06V + 0,002) \Rightarrow 5,997 \pm 0,061V$$

## Kalibrácia meradiel

postupná strata presnosti

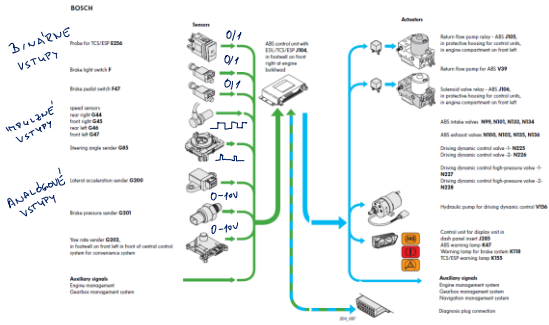
-- napr. časom, opotrebovaním, únavou materiálu, deformáciou, starnutím...

potreba pravidelnej kontroly a prípadne kalibrácie  
(= naviazanie na etalón)

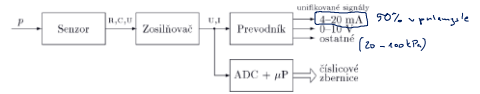
Slovenský metrologický ústav,  
Slovenská legálna metrologia  
ale i podnikové laboratóriá a pod.



System overview ESP - Electronic Stability Programme BOSCH



Všeobecná schéma snímača.



Senzory

- Aktívne** - pôsobením meranej fyzikálnej veličiny generujú elektrický signál, ktorý stačí len zosilniť, pričom nepotrebuje zvláštny zdroj energie. Využívajú javy: elektromagnetická indukcia, Hallov jav, termoelektrický jav (Peltierov), piezoelektrický jav, fotoelektrický jav.
- Pasívne** - musíme vybudovať externý zdrojom, až potom sa mení nejaký parameter, ktorý vieme upraviť a zosilniť ( $\Delta x \rightarrow \Delta R, \Delta L, \Delta C, \dots$ ).

V anglosamerickej literatúre sa označujú presne naopak – pasívny senzor je preto pasívny, lebo nepotrebuje aktívne budenie.



Aktívne snímače – napätie

Príklad: termočlánok, pH sondy, ORP sondy, fotočlánky.

Napätia bývajú v rádu  $\mu V$ , resp.  $mV$ .

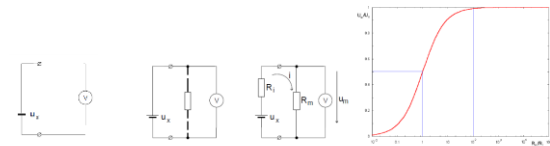
Problém: zafaržiteľnosť.  $R_{vst} > 10^9 \Omega$



Prístroj	$R_{vst}$	$I_{vst}$
Deprézsky voltmeter	$2 \cdot 10^5 \Omega$	$50 \mu A$
Avomet	$5 \cdot 10^5 \Omega$	$20 \mu A$
Elektronický voltmeter	$10^7 \Omega$	
Elektronkový voltmeter	$10^7 \Omega$	
Digitálny multimeter	$10^7 \Omega$	
pH meter	$10^{12} \Omega$	
Prístrojový OZ	$10^9 \Omega$	$10 nA$
FET OZ	$10^{12} \Omega$	$40 pA$

Požiadavky na zosilňovač

- diferenciálny zosilňovač
- symetria
- vyšoký  $R_{vst}$
- linearita
- vyšoké CMRR
- nezávislé nastavenie  $K, q$



Senzory

Podľa výstupu

- aktívne** - U, I, f..
- pasívne** - zmena parametrov (R, L, C,...)

Pozn.: v anglosaskom svete je pasívny snímač taký, ktorý nepotrebuje budenie, tj. presne naopak.

Senzory

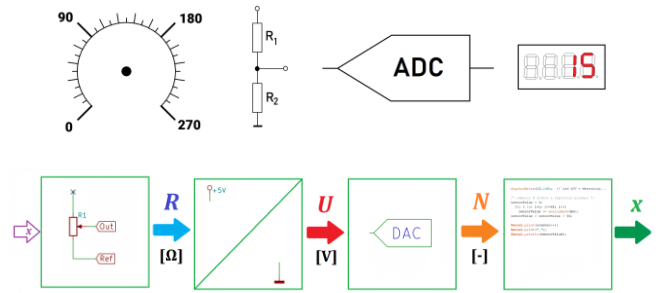
Podľa nosiča informácie

- elektrické**
- neelektrické** (pneumatický, mechanický, optický, magnetický...)

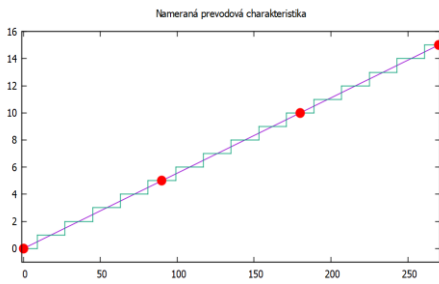
## Senzory

Podľa meranej veličiny:

- mechanické - poloha, otáčky...
- tepelné
- elektrické - U, I, P, ..
- magnetické - B, H,  $\Phi$
- radiačné - svetlo (IR, UV),  $\alpha$ ,  $\beta$ ,  $\gamma$ , kozmické ...
- chemické - pH, analýza ..



### Namerané hodnoty

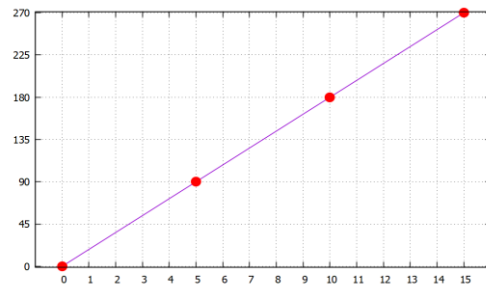


Uhol [°]	N [-]
0	0
90	5
180	10
270	15

### Ideálny prípad

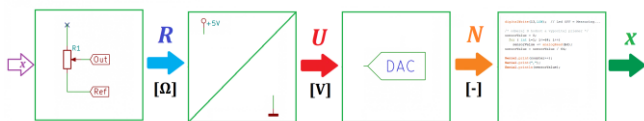
Inverzná charakteristika

```
int N = analogRead(vstup4);
Uhol = f(N);
```

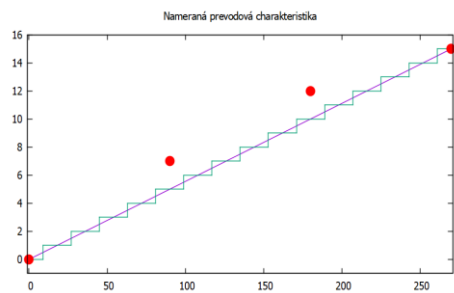


Uhol [°]	N [-]
0	0
90	5
180	10
270	15

### Príklad 2: nelineárna inverzná charakteristika

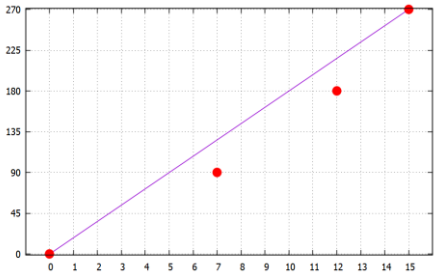


### Namerané hodnoty



Uhol [°]	Lin [-]	N [-]
0	0	0
90	5	7
180	10	12
270	15	15

### Inverzná charakteristika

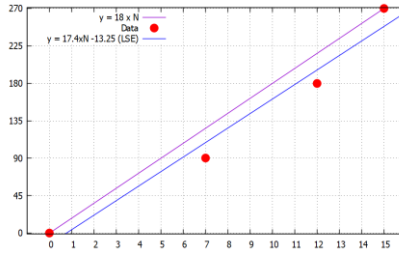


Lin	N	Uhol
[-]	[-]	[°]
0	0	0
5	7	90
10	12	180
15	15	270

```
> N = [0 15]
> Uhol = [0 270]
> p = polyfit(N,Uhol,1)
p = ??? ???
> polyval(p,[0 12 15])
    0    90   180
    1.1e+01  1.3e+02  1.0e+03
```

### Inverzná charakteristika

metóda najmenších štvorcov 1. stupňa  
least squares estimation

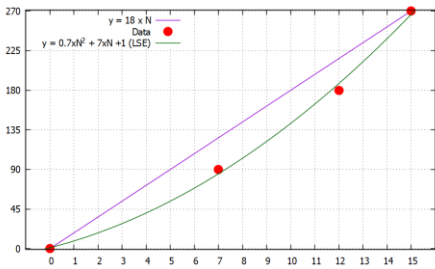


Lin	N	Uhol
[-]	[-]	[°]
0	0	0
5	7	90
10	12	180
15	15	270

```
N = [0 7 12 15]
Uhol = [0 90 180 270]
p = polyfit(N,Uhol,1)
p = 17.4419 -13.2558
polyval(p,N) - Uhol
-13.2558 18.8372 16.0465 -21.6279
```

### Inverzná charakteristika

metóda najmenších štvorcov 2. stupňa  
least squares estimation

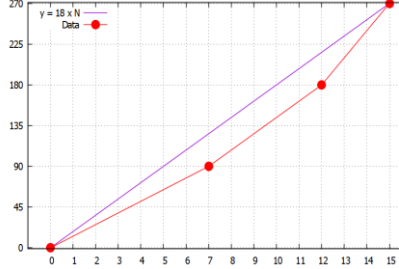


Lin	N	Uhol
[-]	[-]	[°]
0	0	0
5	7	90
10	12	180
15	15	270

```
> N = [0 7 12 15]
> Uhol = [0 90 180 270]
> p = polyfit(N,Uhol,2)
p = 0.7091 7.0364 1.0909
> polyval(p,N) - Uhol
    1.09 -4.98 7.63 -3.81
```

### Inverzná charakteristika

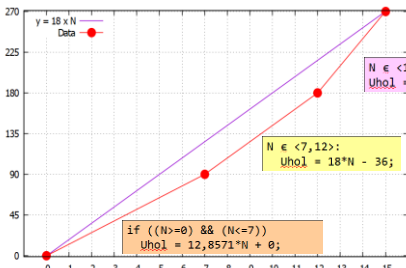
po častiach lineárna náhrada  
PWL - piecewise linear approx.



Lin	N	Uhol
[-]	[-]	[°]
0	0	0
5	7	90
10	12	180
15	15	270

### Inverzná charakteristika

po častiach lineárna náhrada  
PWL - piecewise linear approx.



Lin	N	Uhol
[-]	[-]	[°]
0	0	0
5	7	90
10	12	180
15	15	270

```
N ∈ <12,15>:
Uhol = 30*N - 180;

N ∈ <7,12>:
Uhol = 18*N - 36;

if ((N)=0) && (N<=7)
    Uhol = 12,8571*N + 0;
```

### Inverzná charakteristika

Tabuľka  
lookup table

```
N = [0 7]
Uhol = [0 90]
p = polyfit(N,Uhol,1)
p = 12.8571 0
polyval(p,[0 1 2 3 4 5 6 7])
    0    12.8571    25.7143    38.5714
    51.4286    64.2857    77.1429    90.0000

N = [7 12]
Uhol = [90 180]
p = polyfit(N,Uhol,1)
polyval(p,[7 8 9 10 11 12])

N = [12 15]
Uhol = [180 270]
p = polyfit(N,Uhol,1)
polyval(p,[12 13 14 15])
```

N	Uhol
0	0
1	13
2	26
3	38
4	51
5	64
6	77
7	90
8	108
9	126
10	144
11	162
12	180
13	210
14	240
15	270

Lin	N	Uhol
[-]	[-]	[°]
0	0	0
5	7	90
10	12	180
15	15	270

## Inverzná charakteristika

Tabuľka – implementácia  
lookup table

```
#include <avr/pgmapace.h>
//          0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
const PROGMEM int table[] = {0,13,26,38,51,64,77,90,106,126,144,162,180,210,240,270};
...
int N = analogRead(vratup4);
thol = table[N];
```

```
Tabuľka pamiatk
  analog(4) 2, = 26(10(1) 32)
  analog(4) 2, = 10(4) 28(1)
```

N	Uhol
[-]	[°]
0	0
1	13
2	26
3	38
4	51
5	64
6	77
7	90
8	108
9	126
10	144
11	162
12	180
13	210
14	240
15	270