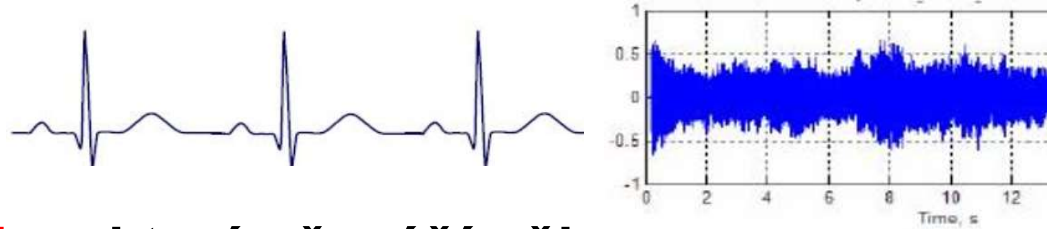


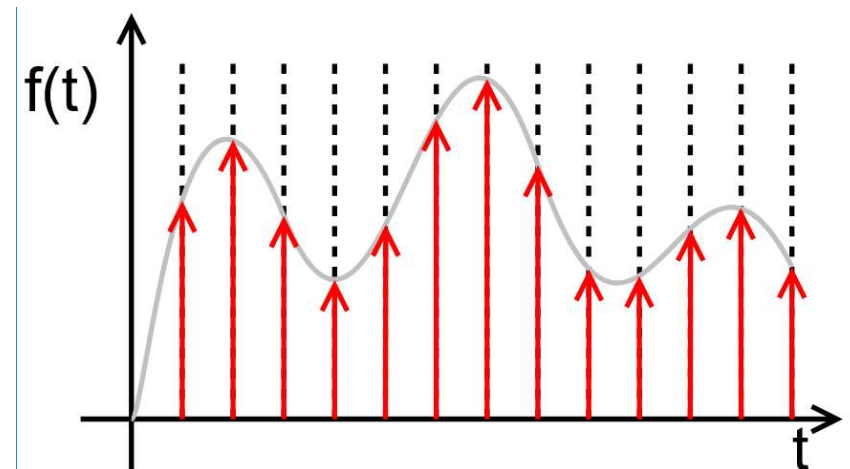
# Měření signálů

- signály a jejich charakteristiky
- úprava vstupního signálu
- analogově-digitální převodníky
- základy vzorkování dat
  - snímání stavu potenciometru
  - měření osvětlení fotoodporem
  - teploměr jinak
- digitálně-analogový převodník a jeho náhrada

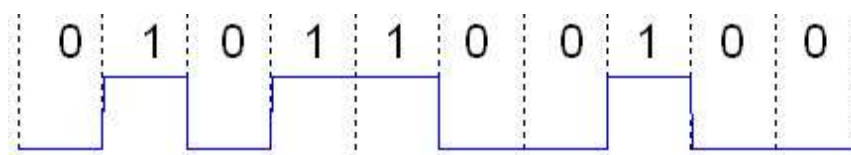
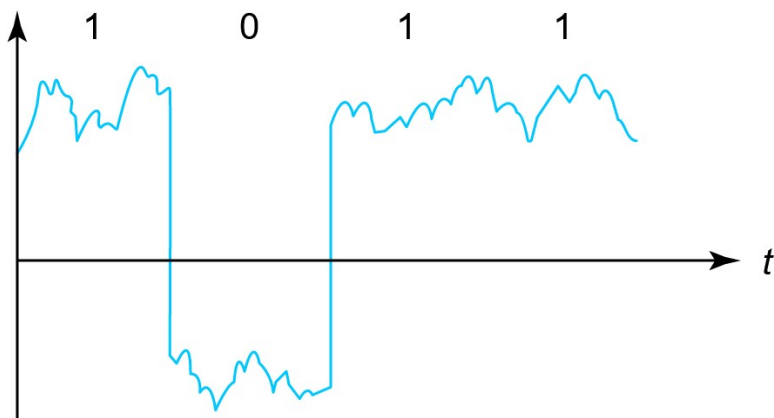
# Signál



- **fyzikální veličina**, která přenáší nějakou zprávu o nějakém jevu, obsahující určité množství (to často závisí i na příjemci) informace
- reprezentován popisem závislosti jednoho parametru (**závislá proměnná**) na parametru jiném (**nezávislá proměnná**)
  - často je nezávislá proměnná čas, může to být ale cokoliv jiného (poloha)
  - signály mohou být optické, elektrické, elektromagnetické, akustické, mechanické, pneumatické, nebo hydraulické
  - závislou proměnnou bude v rámci kurzu obvykle napětí, mnohdy ale reprezentující jinou veličinu
- **spojitý** signál existuje ve všech okamžicích určitého intervalu,
- **diskrétní** pouze v určitých časech
  - například určených vzorkováním
  - obdobně to platí i pro amplitudu signálu

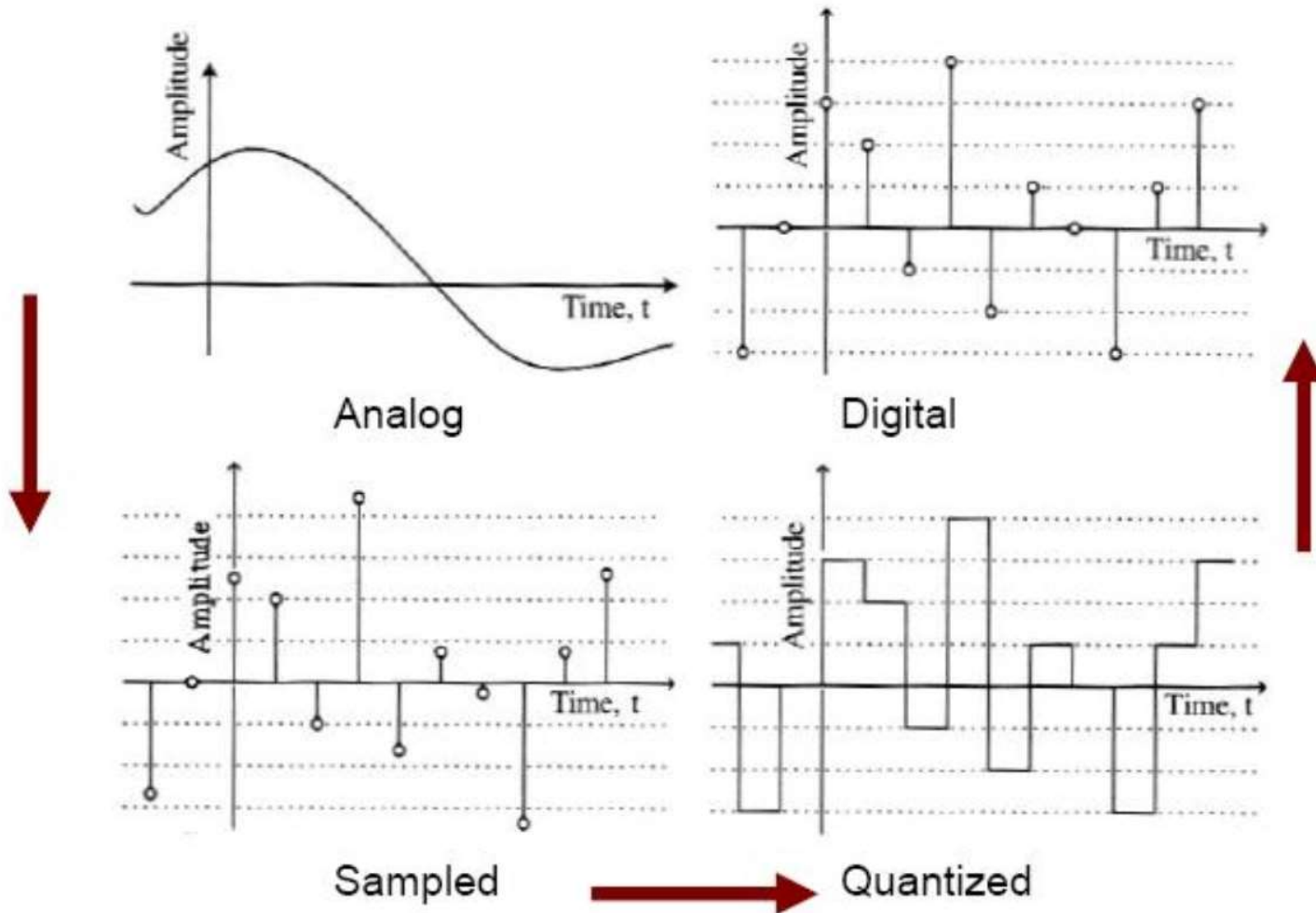


- **determinovaný** - lze určit hodnotu v jakýkoliv okamžik s absolutní jistotou, patří sem **periodický** – signál je definovaný pro  $t \in (-\infty ; \infty)$ 
  - podmnožiny harmonické a neharmonické;                      x **neperiodický**
- **stochastický** – velikost signálu v libovolném okamžiku dovedeme určit pouze s nějakou pravděpodobností
  - stacionární – nezávislé na počátku časové osy,    x nestacionární
  - existují signály, které nejsou deterministické ani stochastické
- **analogový** signál je dán spojitou (nebo po částech spojitou) funkcí spojitého času
  - podle média, kterým jsou přenášeny: akustické, elektrické, optické
- **diskrétní** - je dán funkcí definovanou pouze v diskrétních okamžicích
  - tvoří tak posloupnosti hodnot



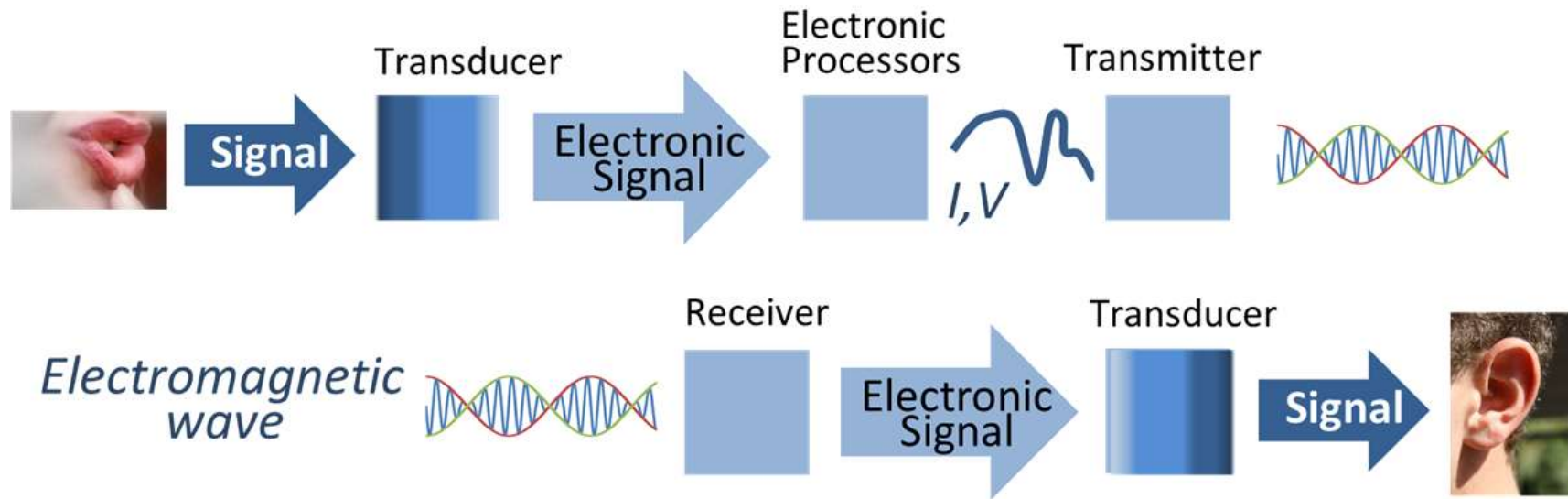
# spojitý vs. diskrétní

- spojité v čase i amplitudě: analogový
- spojité v čase a diskrétní v amplitudě: kvantizovaný
- diskrétní v čase a spojité v amplitudě: vzorkovaný
- diskrétní v čase i amplitudě : digitální



# souvislosti ...

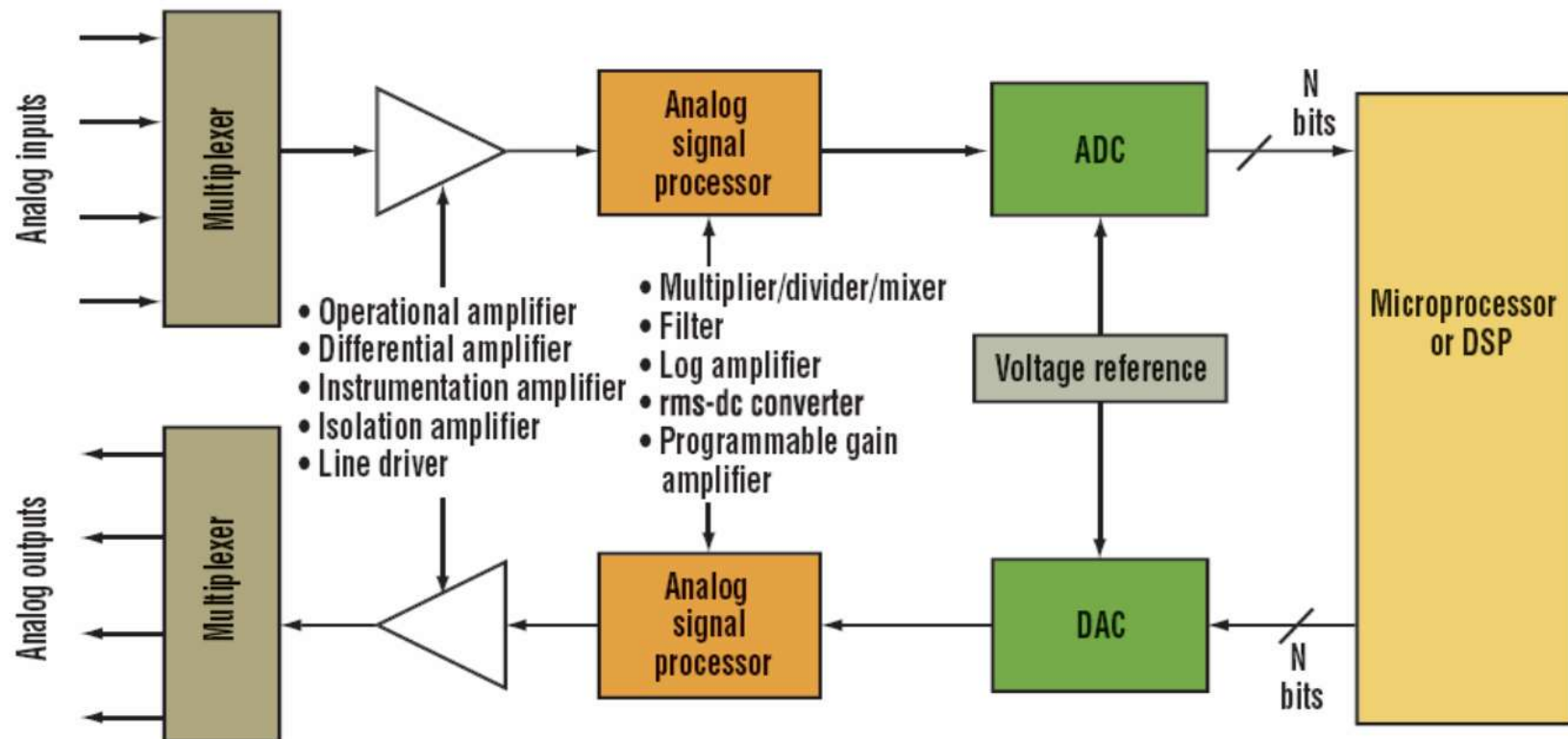
- signály, resp. reprezentované veličiny, často v průběhu přenosu zprávy mění svoji formu



- jeden a tentýž signál může u různých příjemců vyvolat různou reakci
- maximum informace nese přenášená zpráva tehdy, když příjemce předem neví, co bude obsahovat
- zprávu a informaci můžeme obdržet sledováním vybraných parametrů signálu
- signály využíváme zejména ke zjištění stavu různých systémů (živých i neživých)

# DSP digital signal processing

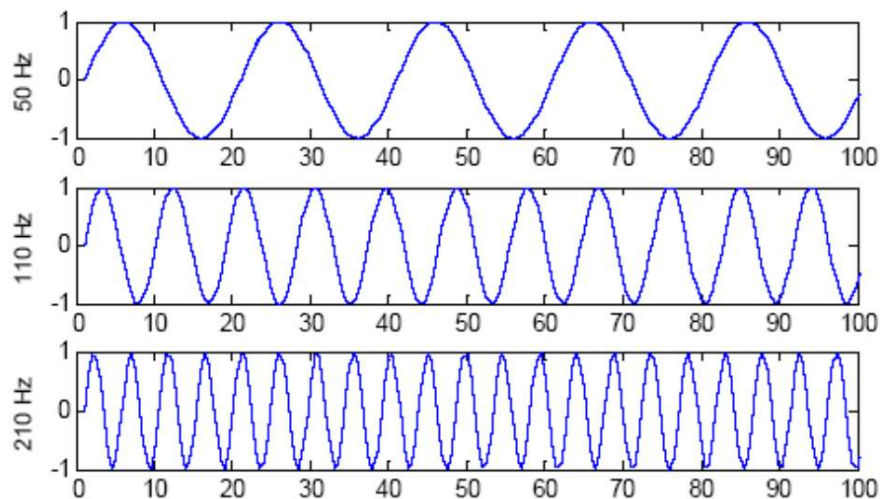
- digitalizace signálu (digital = číslicový) je převod analogového (spojitého) signálu (např. zvuk) do nespojité posloupnosti digitálních údajů, obvykle kódovaných v binární soustavě
- provádí **analogově-digitální převodník** (A/D nebo ADC)
- zahrnuje ale i další kroky - úprava úrovně vstupního signálu, filtrace nežádoucích frekvenčních složek, apod.
- opačný proces pak z digitální informace zpětně vytváří analogový signál – zde se uplatní **digitálně-analogový převodník** (D/A nebo DAC)



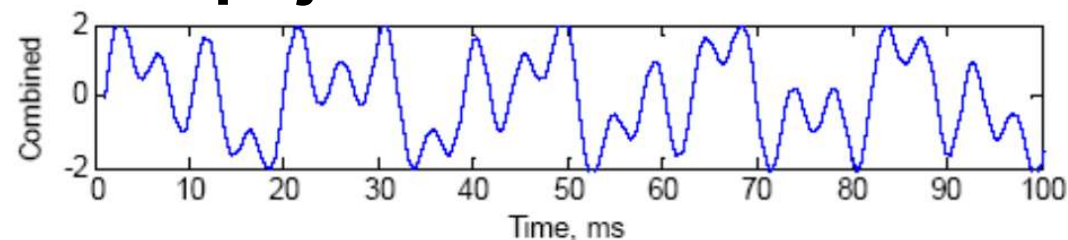
# Filtrace signálu

- **nejčastěji používaná operace v DSP, cílem je:**
  - změna frekvenčního obsahu signálu např. odstraněním některých rušivých frekvencí
  - odstranění šumu v signálu
  - výběr pouze určitých frekvencí, které přenáší informaci
- **typy filtrů:**
  - **dolní propust** (lowpass LPF) – odstraňuje vysoké frekvence ze signálu
  - **horní propust** (highpass HPF) – odstraňuje nízké frekvence ze signálu
  - **pásmová propust** (bandpass BPF) – propouští dál pouze frekvence z určitého frekvenčního pásma
  - pásmová zadrž (bandstop BSF) – zadržuje frekvence určitého pásma
  - notch filter odstraňuje specifickou frekvenci

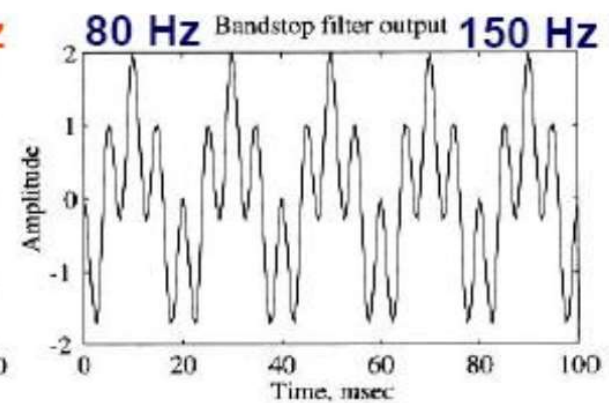
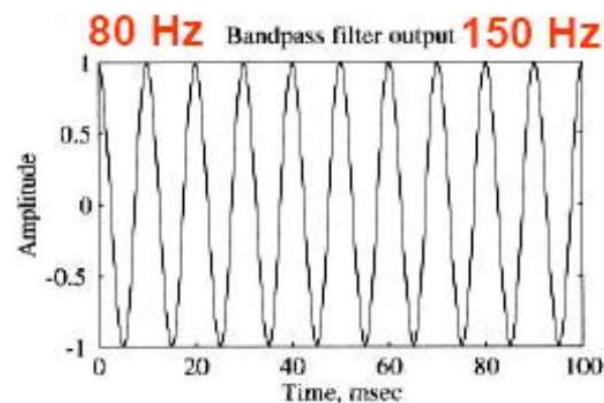
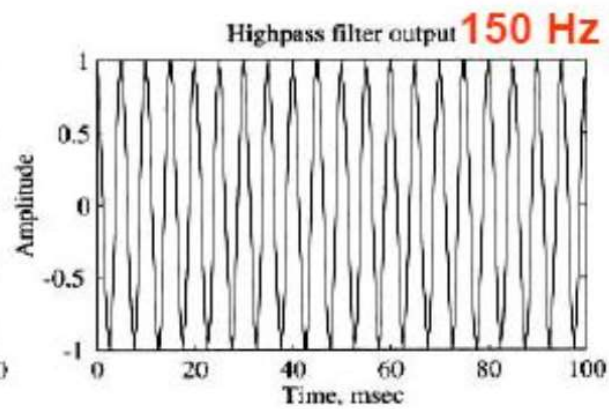
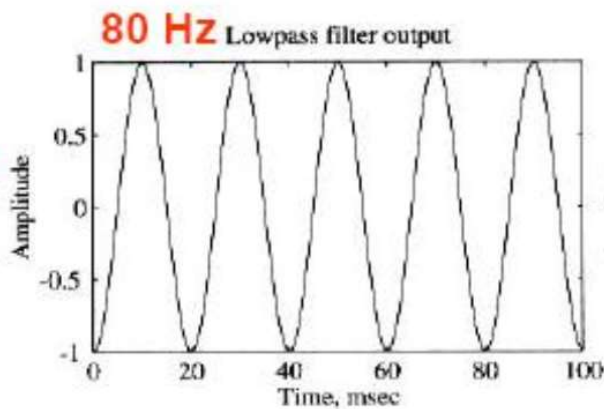
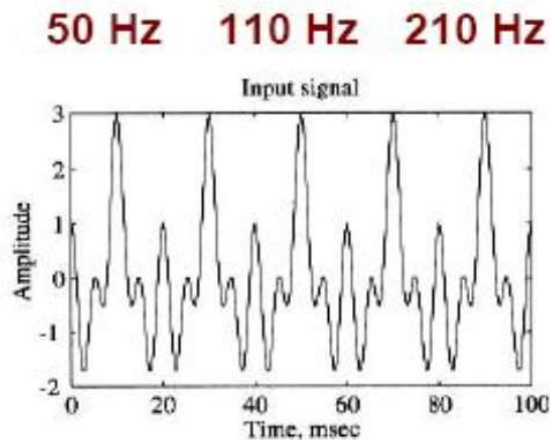
# kombinovaný signál a jeho filtrace



tři periodické složky po spojení:



- aplikace různých typů filtrů:



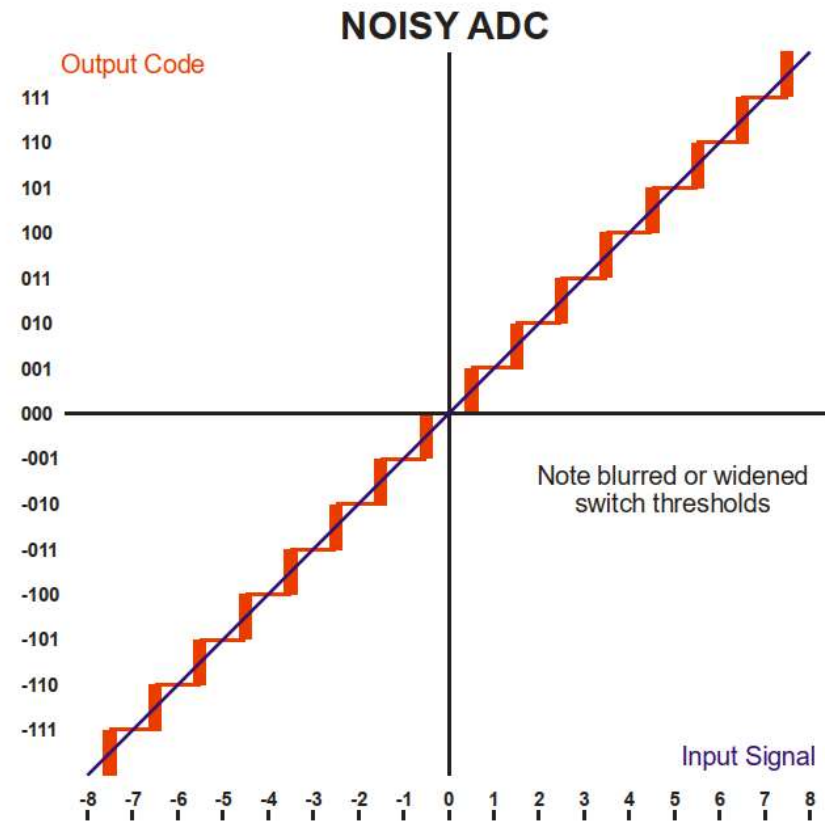
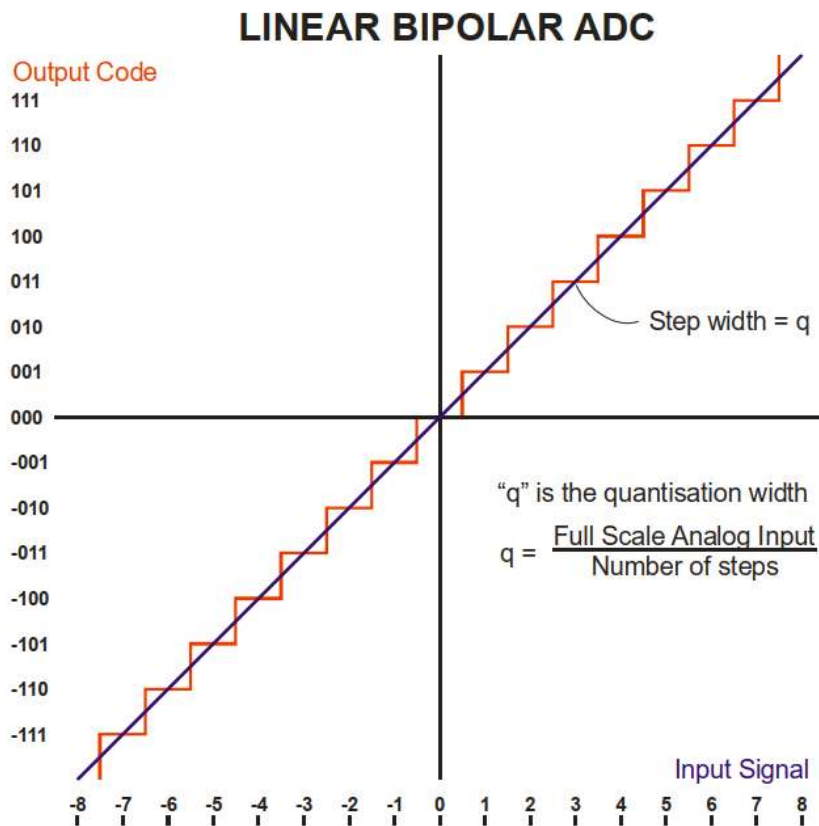


# Zpracování signálů podle:

- **linearity**
  - lineární (platí princip superpozice) – lze je matematicky poměrně snadno modelovat a analyzovat
  - nelineární – nelze je matematicky dobře charakterizovat a analyzovat, zřídka
- **setrvačnosti**
- **systemy bez paměti** – využívají pouze okamžité hodnoty vstupů
- **systemy s pamětí** – obsahují zpoždovací členy (paměťové registry)
  - nerekurzivní využívají pro výpočet výstupu jen vstupních hodnot (i zpožděných)
  - rekurzivní využívají pro výpočet výstupu jak vstupních hodnot, tak zpožděných výstupních hodnot
- **počtu proměnných**
  - jednorozměrné – pracují se signály, závisujícími pouze na jediné proměnné (čas)
  - vícerozměrné - pracují se signály, závisujícími na dvou nebo více proměnných
- **proměnnosti realizovaného operátoru v čase**
  - invariantní v čase (běžné typy zpracování)
  - proměnné v čase (adaptivní filtrace)
- **typu realizace**
  - číslicové – vzorky jsou reprezentovány čísli, zpravidla binárně kódovanými
  - diskrétní analogové – vzorky jsou reprezentovány analogovými veličinami, výpočet v čase je diskrétní

# ADC: ideální převodník

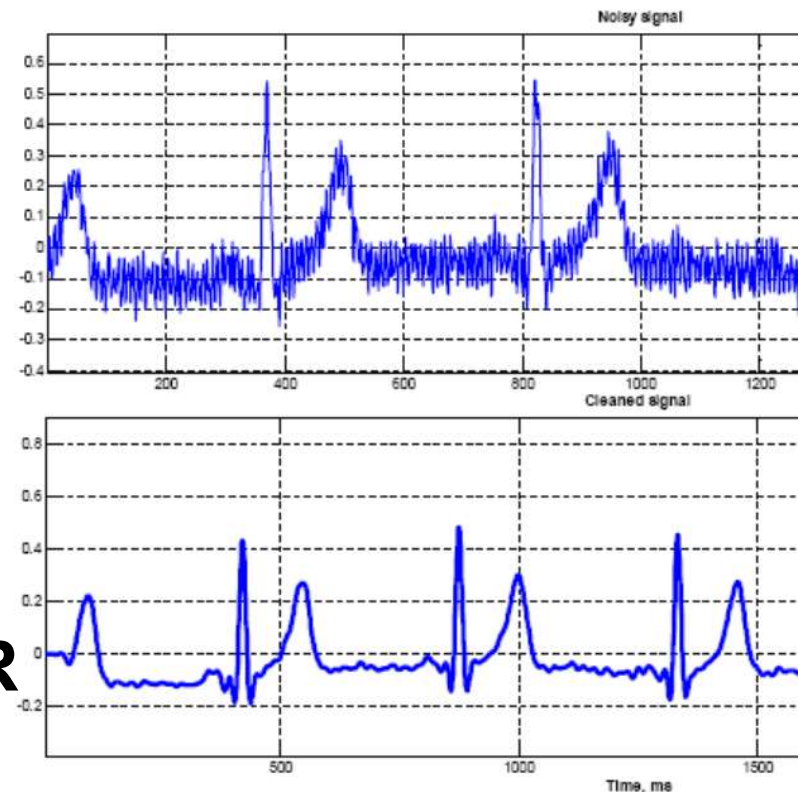
- předpokládá se lineární chování – každý bit má stejnou váhu a reprezentuje stejnou změnu napětí  $q$  (kvantizace) či obecně signálu
- příklad – 7-bitový bipolární převodník, x osa je měřený analogový signál, y osa generovaný výstupní kód:



- bitová změna nastává ideálně při přírůstku (nebo poklesu) o  $q/2$ , ale v reálném případě vpravo jsou tyto oblasti vlivem náhodného šumu zdrojem nestabilit či nejistot v přechodových bodech

# Rozlišení a šum ADC

- **rozlišení** (resolution) je nejmenší detekovatelná změna analogového signálu
  - nominální hodnota vychází z kvantizačního kroku, ale v realitě to je samozřejmě často podstatně horší
- **šum** (noise) je kombinací různých vlivů, nemusí být nezbytně náhodný (white noise)
  - může být odvozen od špičkových (**peak**) nebo „průměrné“ hodnoty **RMS** (viz dále) nebo jako **procento** z měřícího rozsahu
  - tím pádem se hodnoty mohou až 6x lišit
  - zdrojem šumu 50 Hz je elektrická síť
  - to, co nakonec měříme, je kombinací signálu a šumu
  - šum je „neužitečný“ signál ...
- **poměr signál / šum S/N, resp. SNR**



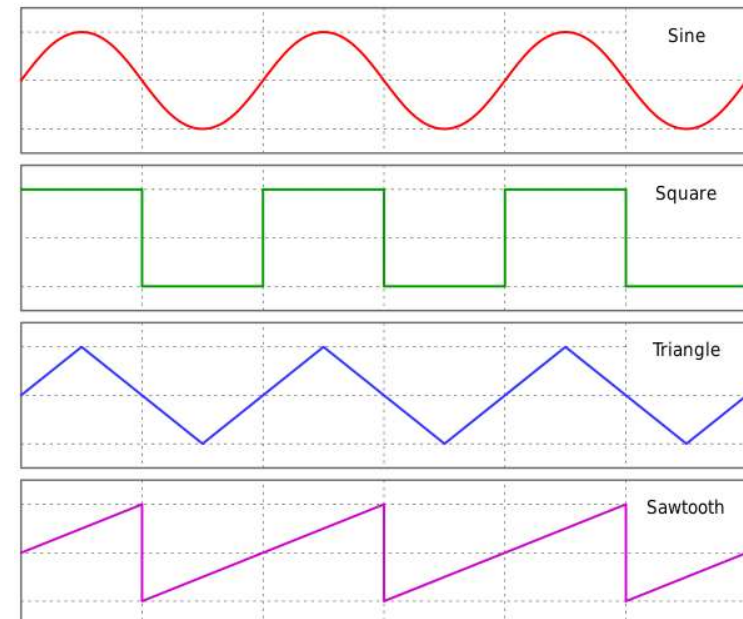
# RMS (rms, root mean square)

- **střední kvadratickou hodnota** (kvadratický průměr) je statistická veličina představující druhou odmocninu aritmetického průměru druhých mocnin daných hodnot:

$$\text{RMS} = \sqrt{x^2} = \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2} = \sqrt{\frac{X_1^2 + x_2^2 + \dots + x_n^2}{n}}$$

- je vždy nezáporné a větší nebo roven aritmetickému průměru
- umocnění hodnot na druhou - větší váha hodnot vzdálenějších od nuly; vzdáleně to připomíná výpočet váženého průměru
- např. při výpočtu střední kvadratické odchylky
- při výpočtu **efektivní hodnoty střídavého napětí** (nebo střídavého proudu)

Waveform	Variables and operators	RMS
DC	$y = A_0$	$A_0$
Sine wave	$y = A_1 \sin(2\pi ft)$	$\frac{A_1}{\sqrt{2}}$
Square wave	$y = \begin{cases} A_1 & \text{frac}(ft) < 0.5 \\ -A_1 & \text{frac}(ft) > 0.5 \end{cases}$	$A_1$
Triangle wave	$y =  2A_1 \text{frac}(ft) - A_1 $	$\frac{A_1}{\sqrt{3}}$
Sawtooth wave	$y = 2A_1 \text{frac}(ft) - A_1$	$\frac{A_1}{\sqrt{3}}$



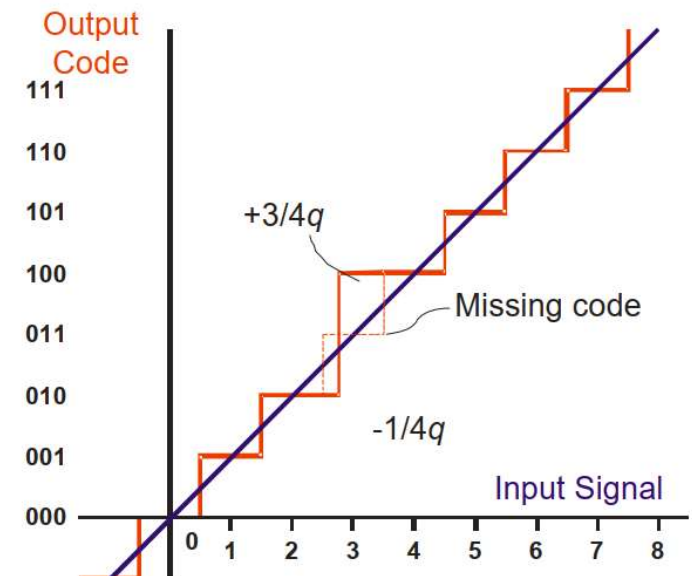
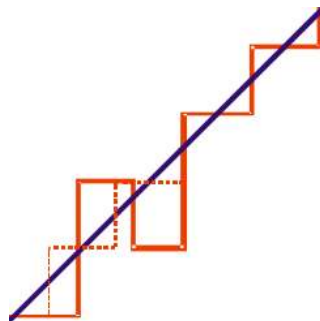
$A_1$  pozitivní peak,  $A_2$  neg. peak, střed je v 0

## další detaily A/D převodu

- **kvantizační chyba** – rozdíl mezi reálnou a převedenou hodnotou, dáno konečným počtem kvantizačních kroků v daném místě převodu, vyjadřuje se jako  $Q_n$  šum
  - analogie zaokrouhlování reálného čísla na celé číslo

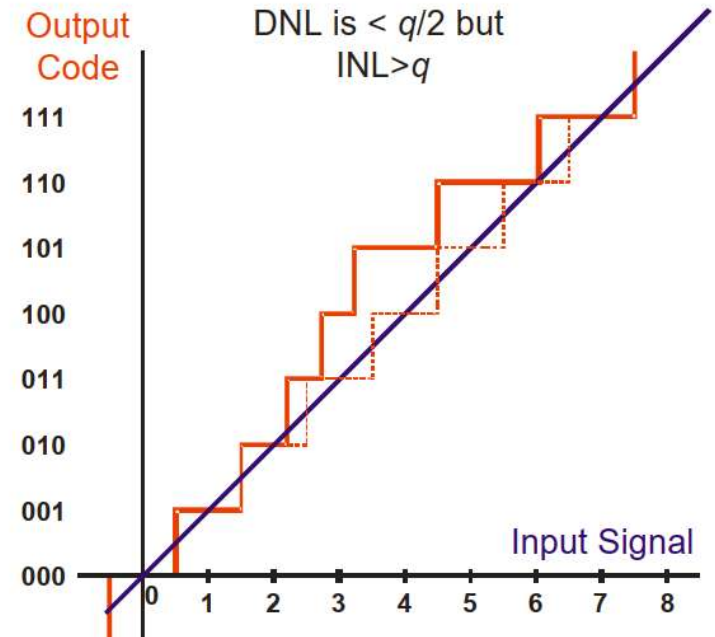


- **diferenciální nelinearita (DNL)** – míra chyby u individuálních převodových kroků, dobrý ADC to má blízko  $q/2$  hodnoty
- pokud to je někde větší, projeví se to jakoby scházejícím údajem (**missing code**)
- převodník přeskakočí mezi nesousedními výstupními údaji
- **nemonotónnost**
  - dvě vzdálené hodnoty poskytnou shodný údaj



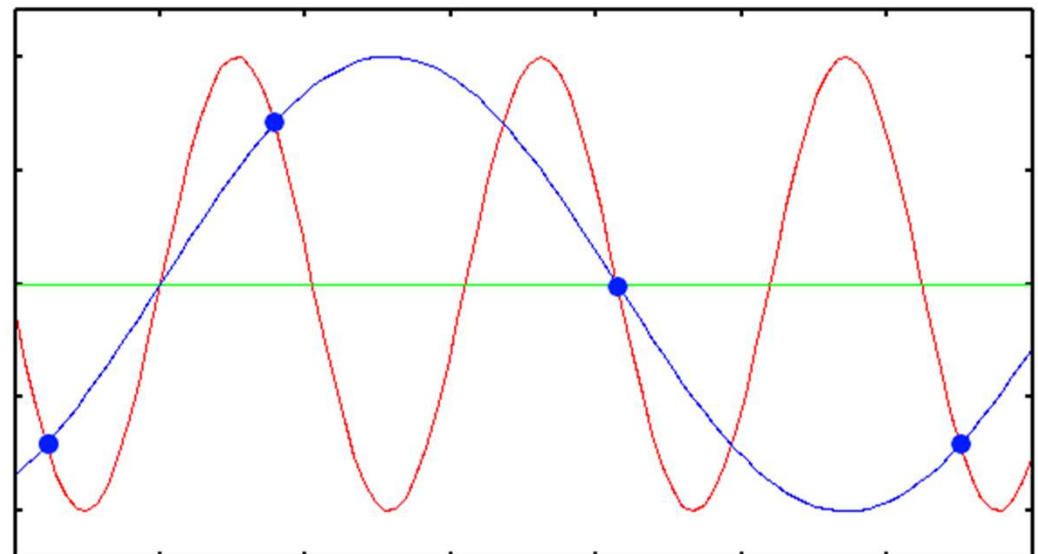
# a ještě více komplikací

- **integrální nelinearita (INL)** – rozdíl mezi ideálním a reálným převodem v daném místě, je větší než  $q$
- **efektivní počet bitů ( $N_{ef}$ , ENOB)** odpovídá degradaci maximálního rozlišení převodníku v reálných podmínkách měření
  - např. 24-bitový převodník poskytne pouze 20 reálných bitů, ostatní je šum...
- „**oversampling**“ – vzorkování signálu s frekvencí podstatně vyšší než **Nyquistova** frekvence = dvojnásobek frekvenčního rozsahu měřeného signálu (šířka pásma, **bandwith** je rozdíl mezi nejnižší a nejvyšší frekvenční složkou daného signálu, to první se často bere jako nula – stejnosměrná **DC** složka)
  - kombinování mnoha opakovaných měření obvykle poskytne mnohem lepší rozlišení
  - střídavý signál se značí AC (alternating component, DC je direct component)



# Rychlost vzorkování

- **vzorkovací frekvence** (sampling rate), definuje počet vzorků za čas (obvykle za 1 s) načítaných ze spojitého analogového signálu při jeho přeměně na diskrétní data
- **Nyquistův teorém** říká, že dokonalá rekonstrukce signálu ze zaznamenaných diskrétních dat je možná pouze tehdy, když byla vzorkovací frekvence větší než dvojnásobek maximální frekvence přítomné v signálu
- při použití nižší vzorkovací frekvence se po převodu zpátky na analogový signál mohou v důsledku tzv. **aliasingu** („zfalšování“) objevit ve výsledném signálu frekvence, které v tom původním nebyly
  - před ADC se zařazuje tzv. antialiasingový filtr, který má za úkol odfiltrovat vyšší frekvence - dolní propust
- pomalé vzorkování měřené červené sinusoidy poskytne data, která odpovídají zdánlivé modré sinusoidě
- jiný příklad – moiré v obrazech



# Digitální zpracování -

## ■ omezení:

- dle zpracovávaného rozsahu kmitočtu – vhodné pouze pro nižší
- frekvenční pásma – převodníky pro vyšší frekvence jsou obvykle drahé, použitelnost DSP je řádově do stovek MHz
- degradace signálu vlivem A/D převodu
- cena zařízení

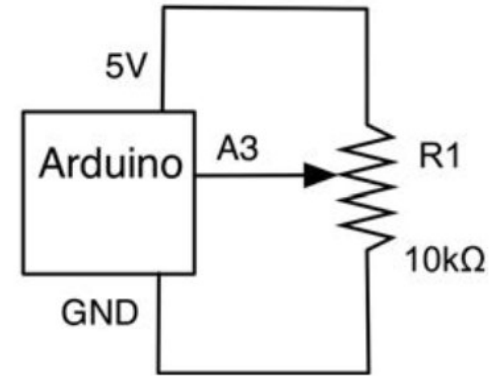
## ■ výhody:

- **pružnost** – charakteristiky číslicového systému jsou dány konstantami / parametry, které lze snadno měnit (lze změnit sw)
- časová **stálost** – vlastnost systému je daná konstantami a programem – pokud systém funguje, nemůže dojít ke změně charakteristik např. vlivem teploty popř. okolního prostředí
- dokonale **reprodukovatelné**
- časově **neomezená paměť** – možnost zpracovávat i pomalé signály, číslicové prvky jsou v tomto případě spolehlivější a jednodušší než objemné a nákladné analogové systémy
- odpadají problémy vzájemného ovlivňování soustav – impedanční přizpůsobení, vliv elmag. pole, ...
- možnost spolehlivé realizace komplikovaných systémů
- slučitelnost s **informačními systémy** (data do cloudu, na server, ...)
- možnost **multiplexního** provozu - u výkonných systémů lze zároveň zpracovávat více než jeden signál



# Měření s Arduinem

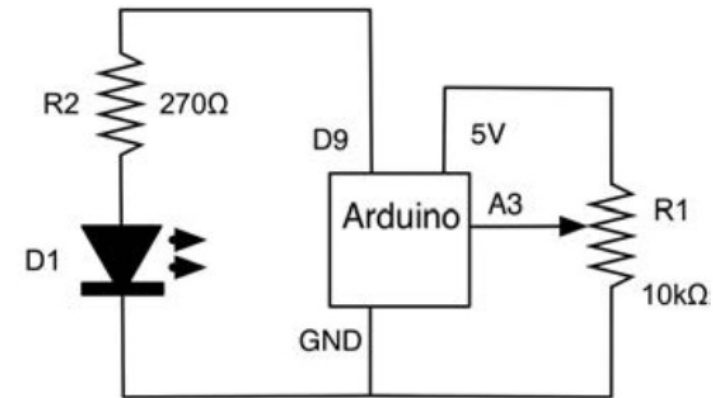
- nejjednodušší je využít **vnitřní ADC** – je možné přepínat několik analogových vstupů – A0, A1, ...
- pokud chceme jednoduše generovat proměnlivý signál, stačí použít potenciometr připojený mezi napájecí napětí (u XIAO je to 3.3 V) a zem GND
  - ve spojení s LED je tak možné třeba regulovat dobu jejího svitu:



```
const int voltsInPin = A3;
const int ledPin = 9;

void setup() { pinMode(ledPin, OUTPUT); }

void loop() {
  int rawReading = analogRead(voltsInPin);
  int period = map(rawReading, 0, 1023, 100, 500);
  digitalWrite(ledPin, HIGH); delay(period);
  digitalWrite(ledPin, LOW); delay(period);
}
```

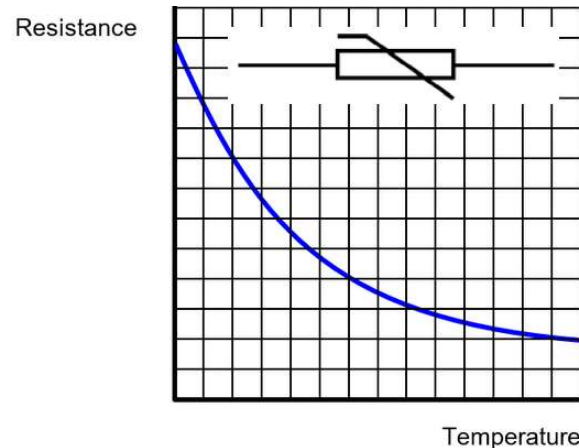


- drobnou úpravou sw lze modifikovat intenzitu svitu:
  - použití PWM modulace
  - jen dig. piny se ~ značkou

```
void loop() {
  int rawReading = analogRead(voltsInPin);
  int brightness = rawReading / 4;
  analogWrite(ledPin, brightness);
}
```

# Sensory s proměnným odporem

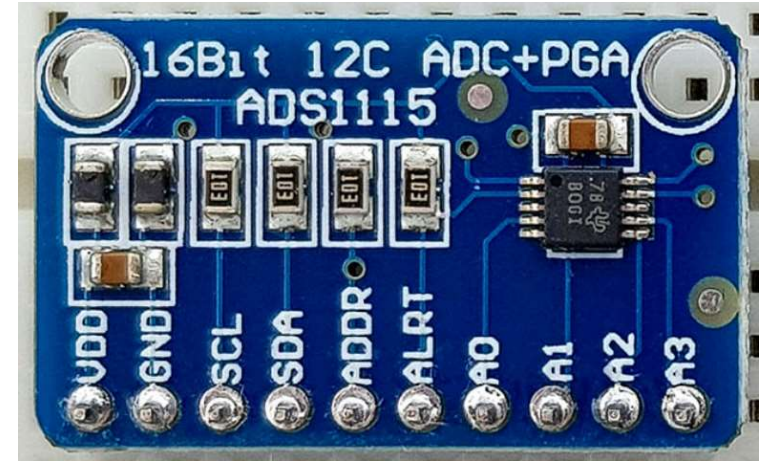
- sledování napětí regulovaného potenciometrem je vlastně základem pro měření sensorů, které se chovají jako odpory závislé na nějaké fyzikální veličině
- využije se princip **odporového děliče** napětí a snímá se změna napětí odpovídající změně světla – **fotoodpor**
  - u fotorezistorů se často udává nejčastěji odpor při osvětlení 10 luxů a teplotě 25 °C (označuje se  $R_{10lx}$ ) – bývá to kolem 10 k $\Omega$
  - zapojíme ho tedy třeba do serie s normálním 10 k $\Omega$  odporem
  - závislost na intenzitě světla je bohužel nelineární, při různých regulacích typu světlo / tma to příliš nevádí
- teploty – **termistor**
  - dvojího typu – běžnější fungují tak, že s rostoucí teplotou jejich odpor klesá (značí se NTC) → pro regulaci stačí, pro měření teploty nepohodlné
  - druhé (PTC) svůj odpor s rostoucí teplotou zvyšují (např. u Pt100 skoro lineárně)



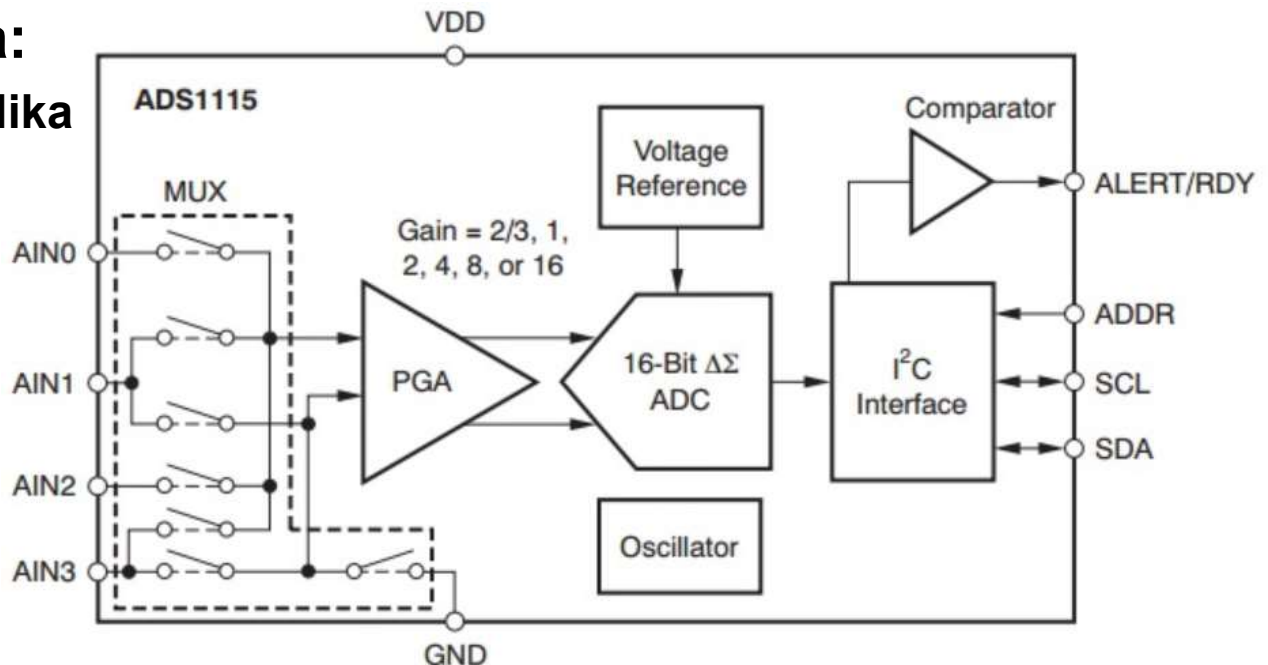


# Přesnější ADC?

- modul XIAO a další řady desek na bázi SAMD procesorů má lepší vnitřní ADC s rozlišením 12 bitů
- pokud to nestačí, je třeba externí přídavný ADC modul nebo čip, např. **ADS1115**
  - nabízí rozlišení 16 bit, 4-kanálové měření (4x **single-ended** – měří se proti GND, resp. 2 **diferencial** vstupy), PGA k předzesílení vstupního napětí, I2C komunikaci
  - protože je to hodně malý obvod, je rozumné si ho koupit připájený na destičce jako modul, rychlost je až 860 vzorků/s (SPS, samples per second), připojovací piny mají standardní vzdálenost 2.54 mm (0.1“)

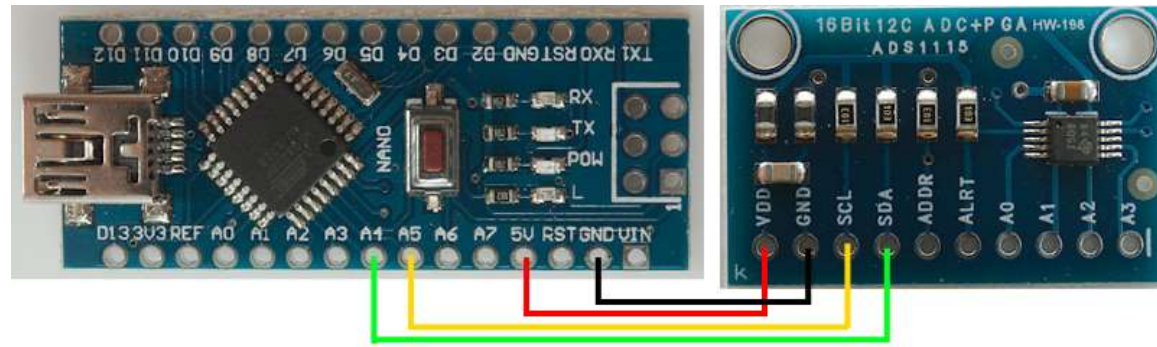


- **blokové funkční schema:**
  - složitější struktura z několika submodulů



- k propojení s Arduino modulem stačí 4 vodiče

- napájecí napětí, GND
- komunikace I2C – SDA, SCL
- zde deska Arduino Nano, ta



má SDA (data) vyvedeno na pin A4, SCL (clock) pak na A5, často jsou SDA a SCL vyvedeny na samostatné piny, a nebo projdeme dokumentaci

- ke komunikaci potřebujeme nainstalovat příslušnou knihovnu

- menu Sketch – Include Library – Manage Libraries a hledat ADS, vybrat lze třeba Adafruit ADS1X15, a pak z Examples – vybrat třeba demo single-ended, upraveno:

```
#include <Adafruit_ADS1X15.h> // single-ended, signál mezi GND a AIN0

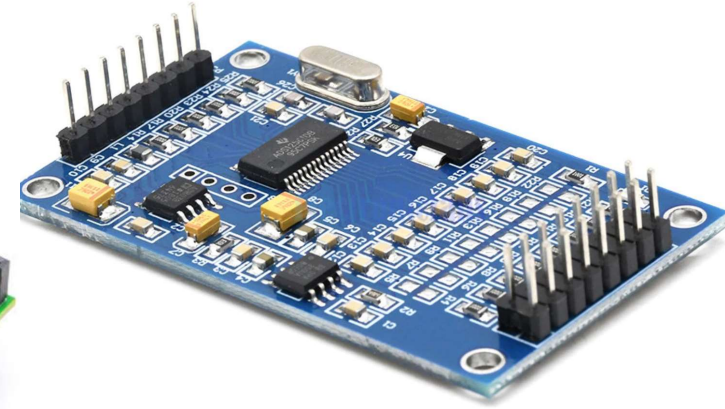
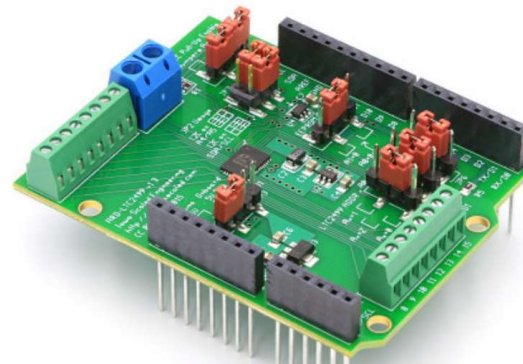
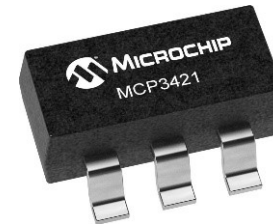
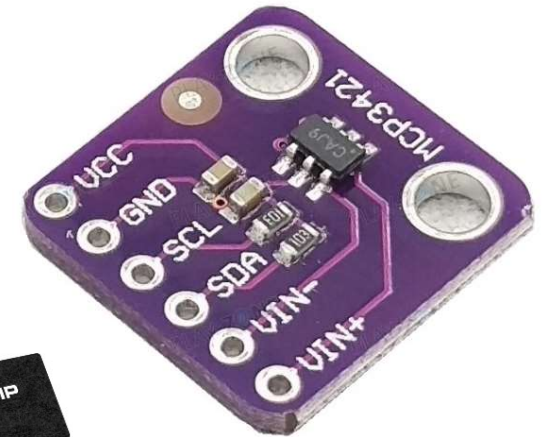
Adafruit_ADS1115 ads;

void setup(void) {
  Serial.begin(9600); Serial.println("Hello!");
  Serial.println("Getting single-ended readings from AIN0");
  Serial.println("ADC Range: +/- 6.144V (1 bit = 0.1875mV/ADS1115)");
  if (!ads.begin()) { Serial.println("Failed to initialize ADS."); while (1); }
}

void loop(void) {
  int16_t adc0; float volts0;
  adc0 = ads.readADC_SingleEnded(0); volts0 = ads.computeVolts(adc0);
  Serial.print("AIN0: "); Serial.print(adc0);
  Serial.print("  "); Serial.print(volts0); Serial.println("V");
  delay(1000);
}
```

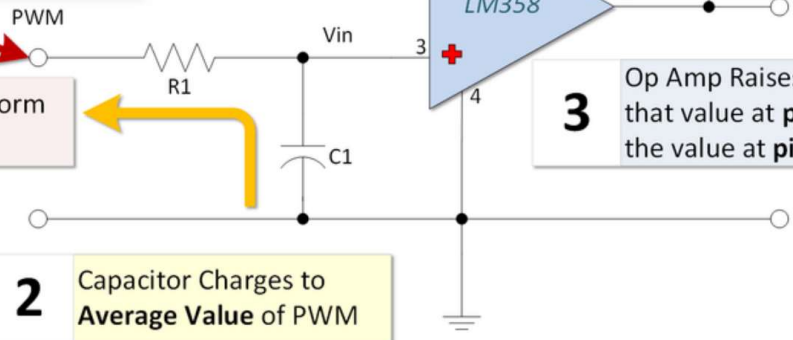
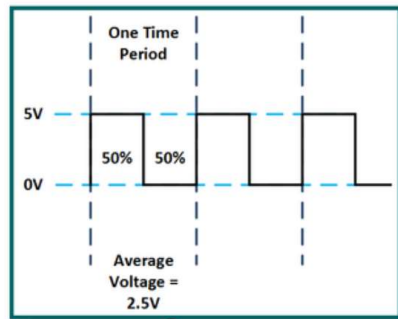
# Lepší rozlišení?

- **MCP3421 (Microchip)**
  - 18-bitový sigma-delta převodník, I2C komunikace
  - interní reference  $2.048V \pm 0.05\%$ , drift  $15 \text{ ppm}/^\circ\text{C}$
  - na vstupu PGA - zesílení 1, 2, 4, 8x
  - diferenciální vstup, rozsah  $\pm 2.048V$  / PGA
  - programovatelné rozlišení 12, 14, 16, 18 bitů
  - rychlost vzorkování 240, 60, 15, 3.75 vzorků/s
  - když si pořídíme modul, nemusíme pájet SMD
- **ADS1232 (TI)**
  - 24-bitový sigma-delta převodník, I2C komunikace
  - PGA: 1, 2, 64, 128x
  - pro váhy, odporové sensory (Wheatst. Můstek)
- **ADS1256 (TI)**
  - 24-bit, 8 SE / 4 DIF vstupy, SPI komunikace
- **LTC2499 (Analog Devices)**
  - 24-bit, 16/8 kan., I2C
  - „shield“ formát
- **AD7124 (Analog Devices)**



# Analogový výstup

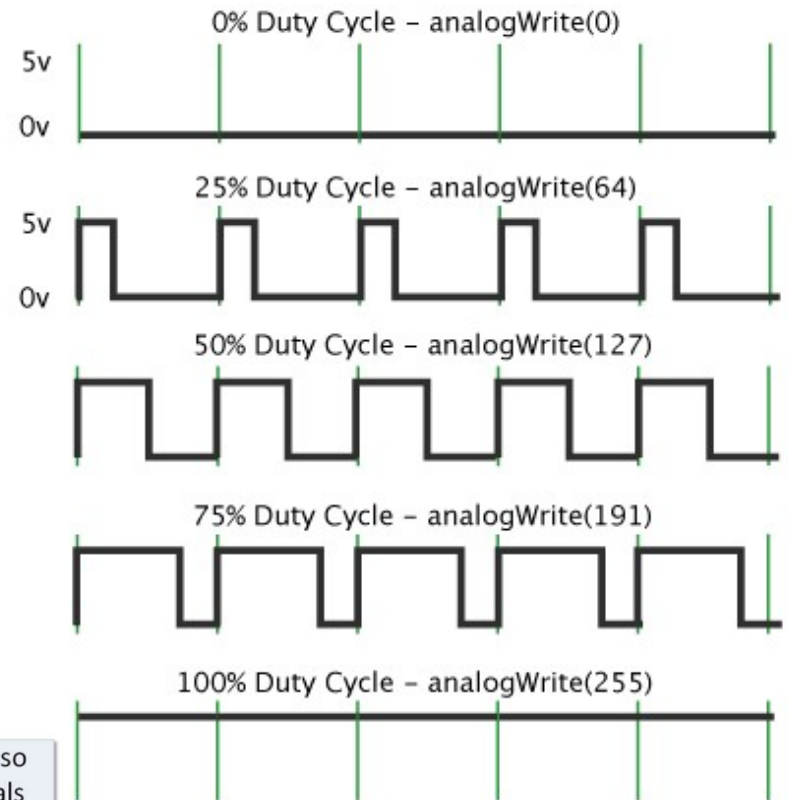
- DAC („digital to analog converter“) je někdy dostupný na lepších či novějších arduino a alternativních modulech (i na Xiao)
- funkce AnalogWrite a PWM („pulse width modulation“)
- po průchodu filtrem získáme proměnlivé napětí:



**1** Apply a PWM Waveform to the circuit input

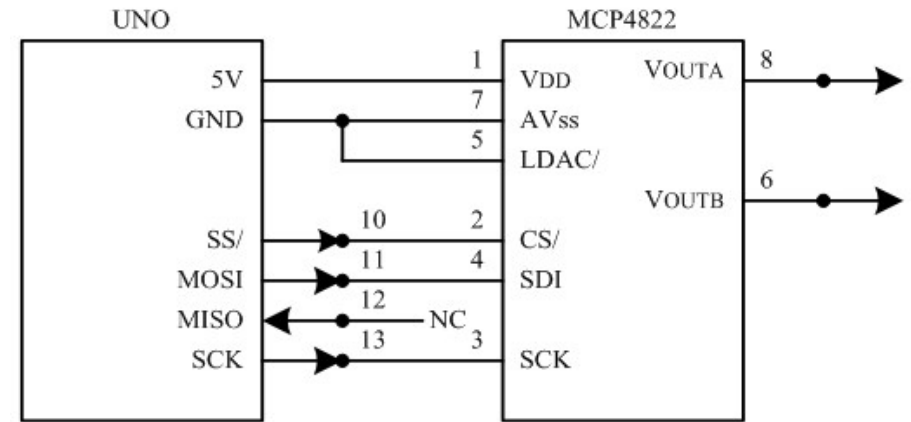
**2** Capacitor Charges to Average Value of PWM

**3** Op Amp Raises Output so that value at pin 2 equals the value at pin 3.



# Lze použít externí DAC

- třeba MCP4822 (Microchip)
- 12 bit, 2-kanálový, SPI



# Střídavé napětí?

- AD9833
  - programovatelný **generátor** signálu DDS pro sinusové, trojúhelníkové a čtvercové vlny, SPI rozhraní
  - výstupní frekvenční rozsah 0 MHz až 12,5 MHz
  - 28-bitové rozlišení, tj. 0,1 Hz při 25 MHz hodinách
  - lepší modul výstup generuje přes OZ a digitální potenciometr – pohodlné nastavení velikosti výstupní amplitudy

