IoT Internet of Things

Hartmut Seitter

Agenda – Topics which should be covered in this course

- Introduction
  - IoT Definition
  - What are the disciplines of IoT
  - A brief history about IoT
- Application scenarios for IoT
  - Cool applications and a lot of opportunities

- **The IoT Value Stack a real-world example**

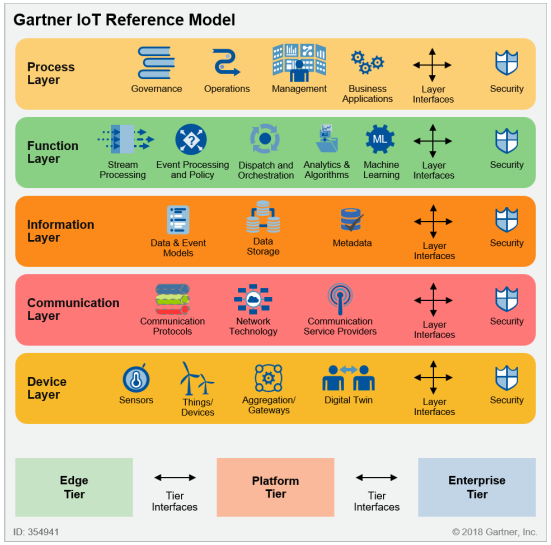
- **Sensors**
  - **Wireless, sensor, networks and IoT**
  - **From Sensors and Actuators to ….**
  - **… embedded systems and computation**
- **Circuits**
  - **Energy and Wireless**
  - **Digital & Analog**

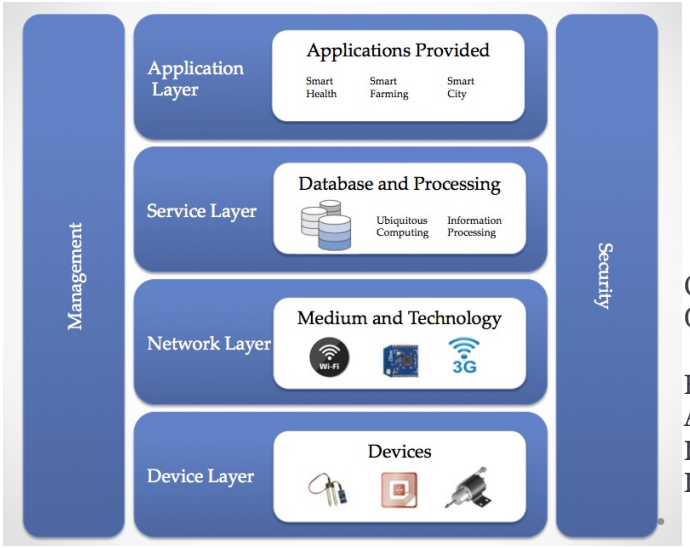
- **Embedded Systems**
  - **Technology Drivers,**
  - **Energy,**
  - **Microcontroller,**
  - **Software**

IoT Value stack

- **Attention: There are several definitions and interpretations of the IoT value stack and the architecture layers of an IoT solution.**

- **A typical IoT solution compromises 5…7 layers – here two samples of IoT Layer structure (more samples are available, and we will discuss it in future sessions)**





Copyright O'Reilly

Hands-On Artificial Intelligence for IoT

IoT Value Stack - The Device Layer

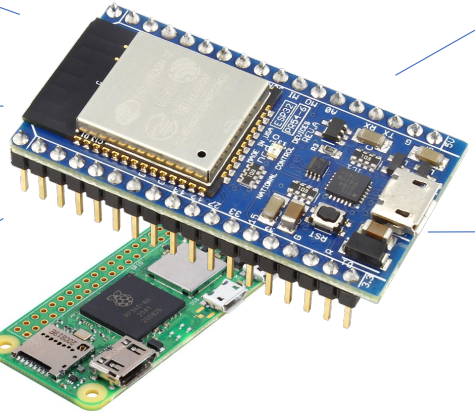
- **The device layer may consist of several components**
  - **Sensors**
    - **Collecting data from the environment (physical characteristics) such as Temperature, Humidity, Motion, Images, ….. and transform it into electrical characteristics (Voltage, Current, Frequency, …..**
  - **Actuators**
    - **Perform actions controlled by commands. Switching things On and Off, control speed of Motors (PWM), control Light intensity and Light color …..**
  - **Embedded Systems**
    - **More and more microcontroller are used to process data locally and control the Sensors and Actuators. Also collecting data from the Sensors aggregate the Data and prepare it for transmission**
  - **Connectivity Modules**
    - **Very often part of the Embedded System**
    - **Enable communication with other devices and the outside of the ‘device layer’**
    - **Attention – in this lecture we distinguish between communication to the sensors/actuators and the central system**
  - **Firmeware**
    - **Software embedded in the device**
    - **Here we also distinguish between firmware of the microcontroller itself, and software developed fo the device application > e.g. your Arduino sketch for esp32**

IoT Value Stack - The Device Layer

Different Types of Interfaces

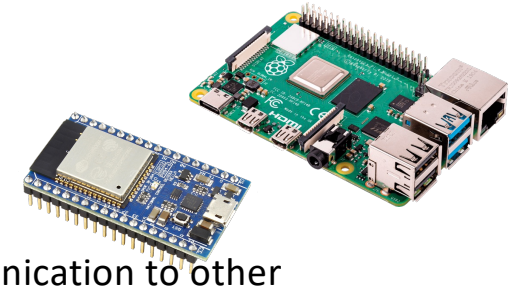
Different types of communication to sensors and actuators

Sensors and Actuators

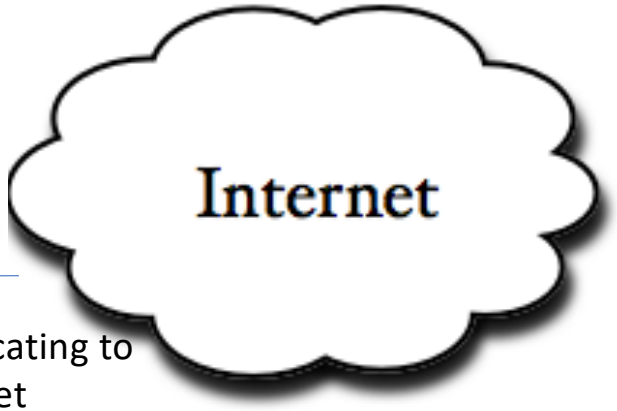


Embedded Systems
Microcontoller or spec. HW

Communication Modules
e.g. placed on a development board

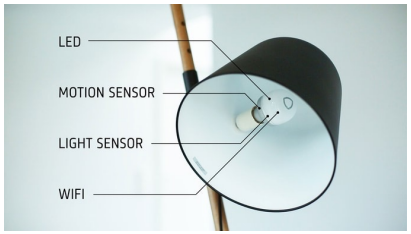


Communication to other devices

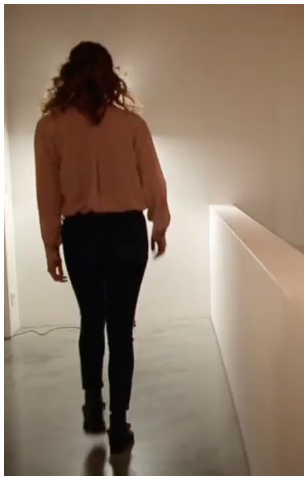


Communicating to the Internet

Another IoT solution – from Sensors to a IoT solution

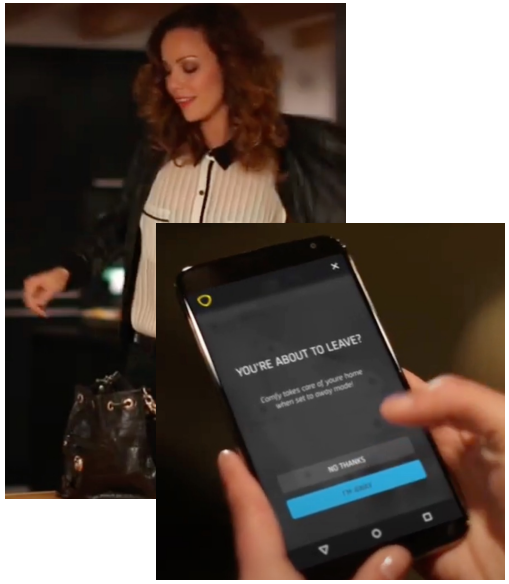


The device with sensor actuator, control unit and communication

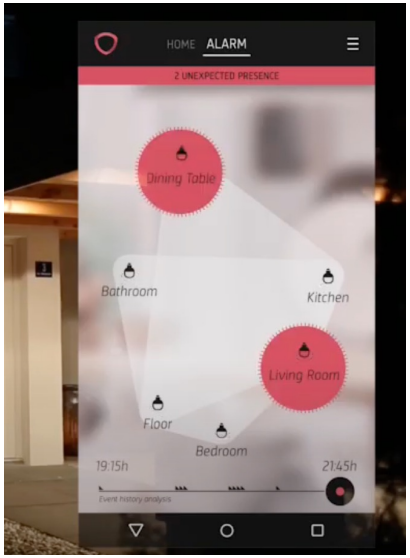


When you are at home
Provide comfort and convenience

Self learning light behaviour

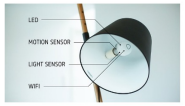


When you leave home – activate secure mode

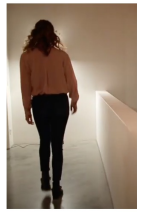


When movement is detected when secure mode is switched on – send an ALARM

Another IoT solution – from Sensors to a IoT solution

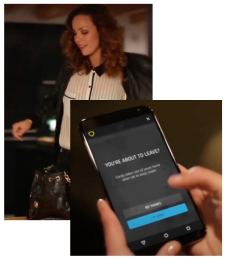


The device with sensor actuator, control unit and communication

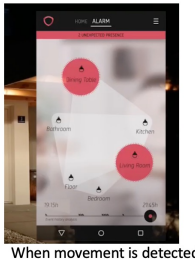


When you are at home
Provide comfort and convenience

Self learning light behaviour



When you leave home – activate secure mode

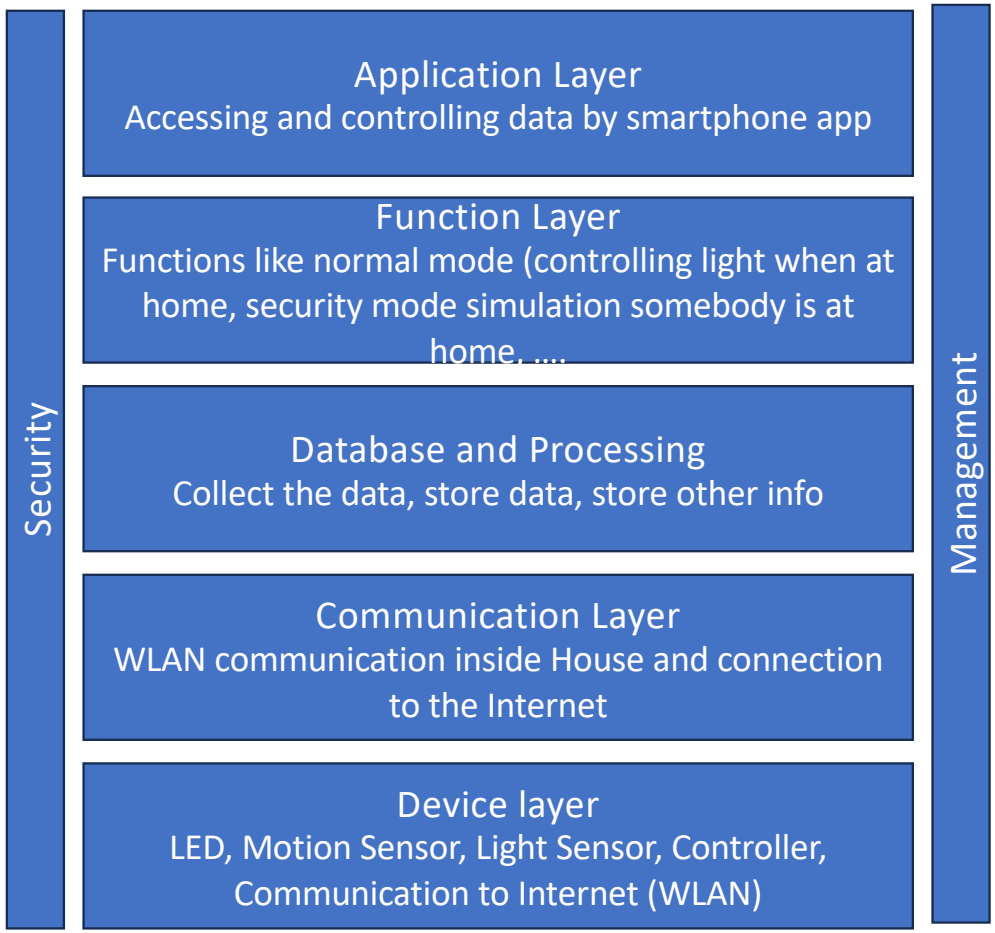


When movement is detected when secure mode is switched on – send an ALARM

Each layer will have special requests and requirements

e.g. Device Layer

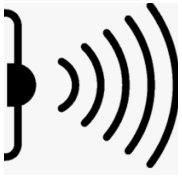
- Form factor, Voltage, Light color, communication type, ….
- Each requirement influence the solution and the price



IoT Value Stack – let's split it into different layers

- **LED, Housing, Industrial design**

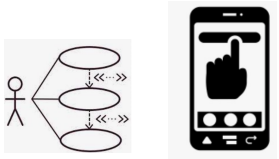
- **Microwave sensor, microprocessor**



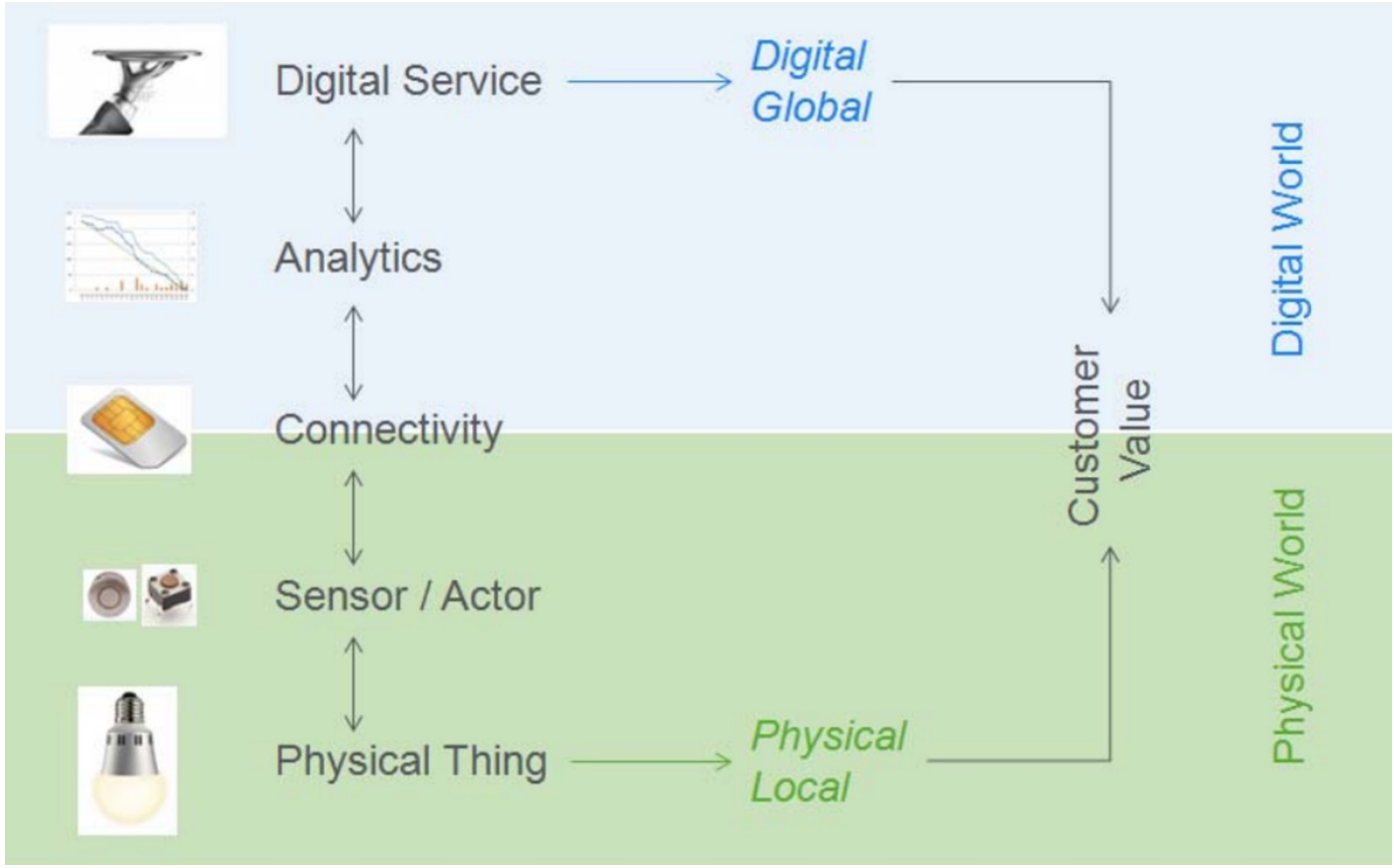
- **Bluetooth, Wifi, Zigbee, ….**

- **User Admin, DBs, Messaging**

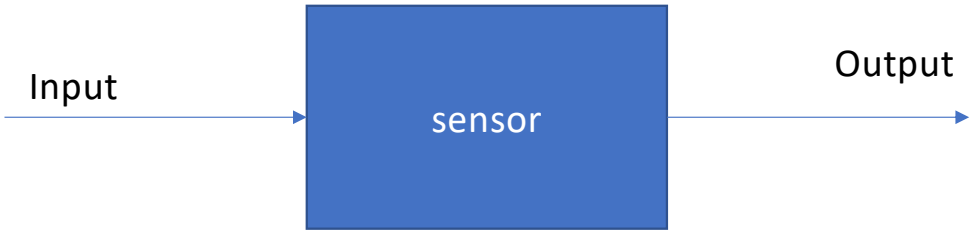
- **Use Cases, Apps, Business model**



Internet of Things value stack



Let’s dive into Sensors a little bit



A sensor transfers an input signal into an output signal

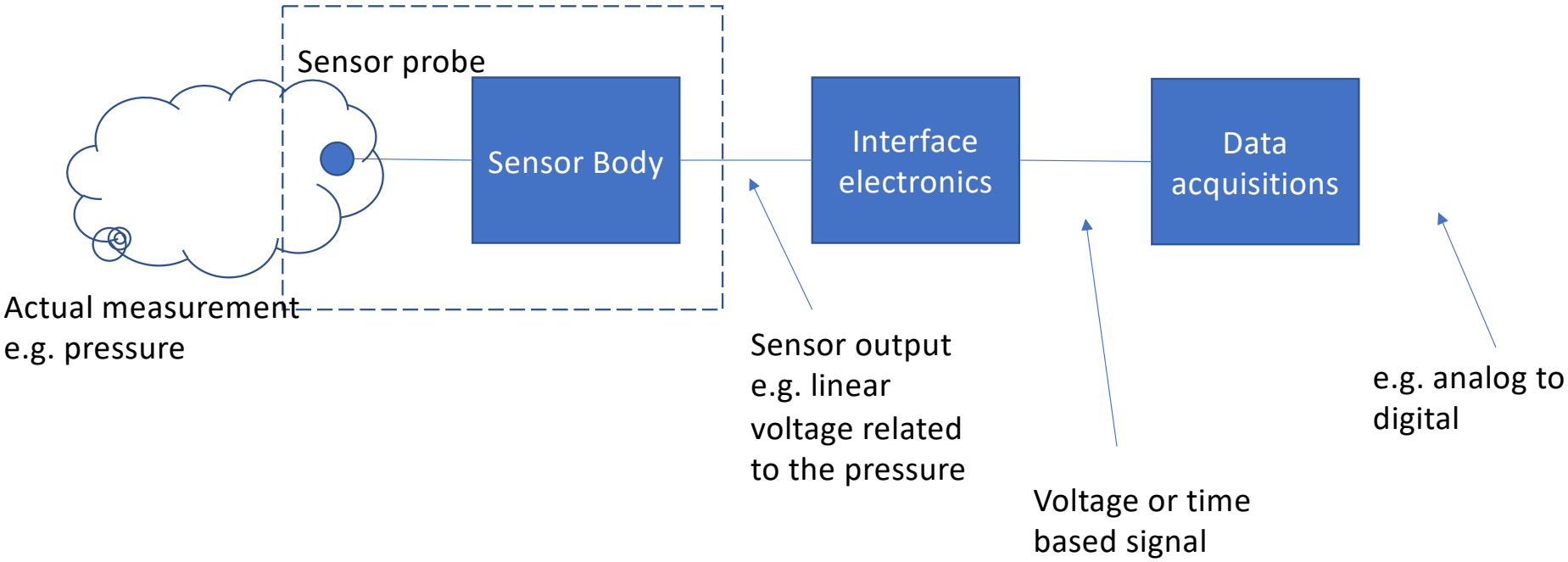
Input signals: pressure, movement, humidity, temperature, ……

Output signals: voltage, frequency, ….

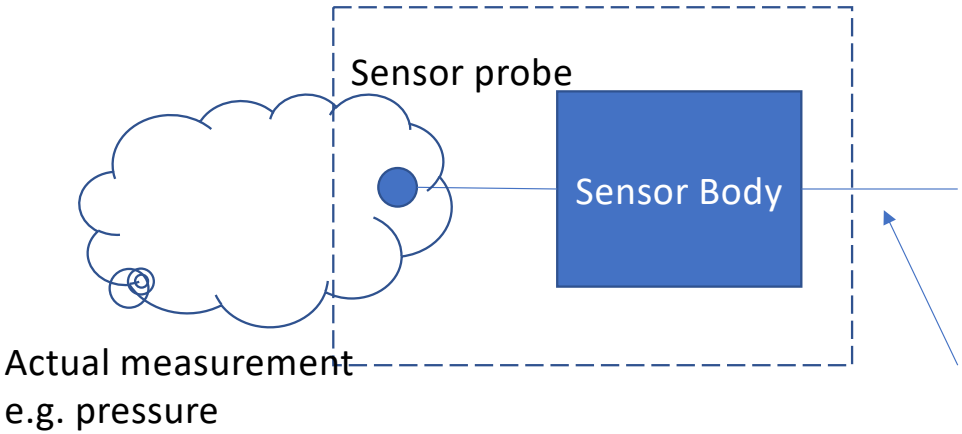
Devices which convert one energy form to another are called - Transducer

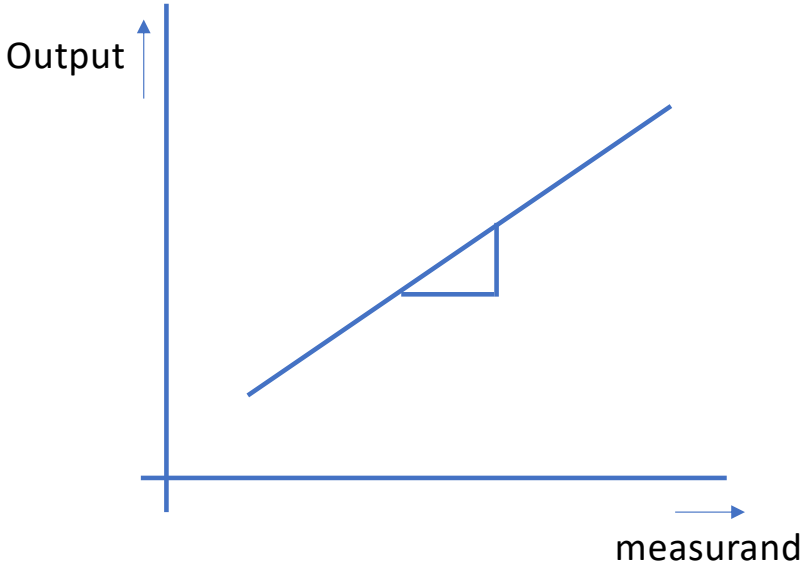
- A transducer is a device that converts energy from one form to another.
  - Usually, a transducer converts a signal in one form of energy to a signal in another form of energy.
- There are two type of transducer
  - Input transducer (Sensors) and output transducer (Actuators)
- Transducer, any device by which variations in one physical quantity (e.g. pressure, brightness) are quantitatively converted into variations in another physical quantity (e.g. voltage, position).
- OR...a device for which changes in input quantity produce corresponding, predictable changes in output quantity B
- For our purposes, Sensors convert physical energy to electrical energy (Sensor)
- Or convert electrical to mechanical (Actuator)

Generic Sensing Application



The ideal sensor



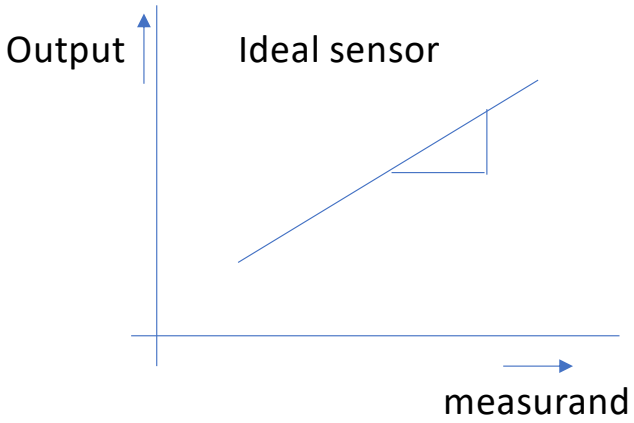


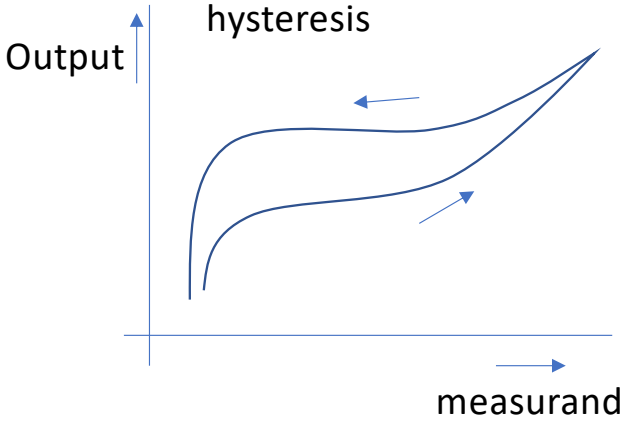
e.g. linear dependency

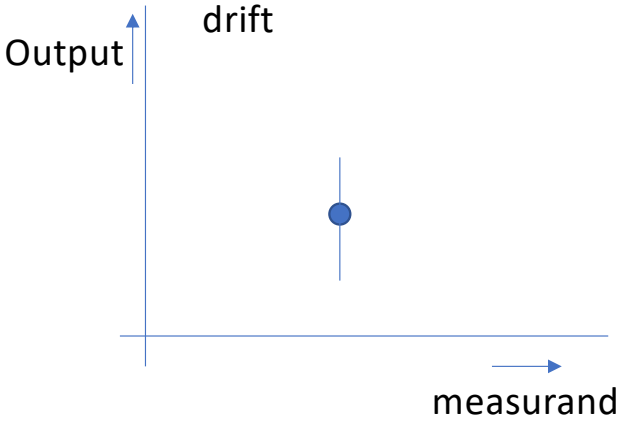
and no other influence

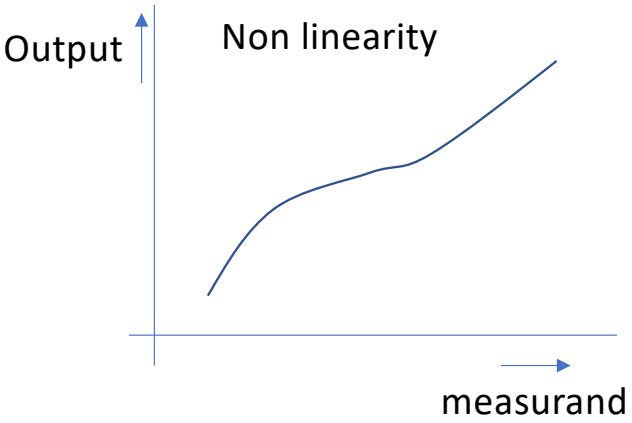
(e.g. pressure measurement is independent from temperature)

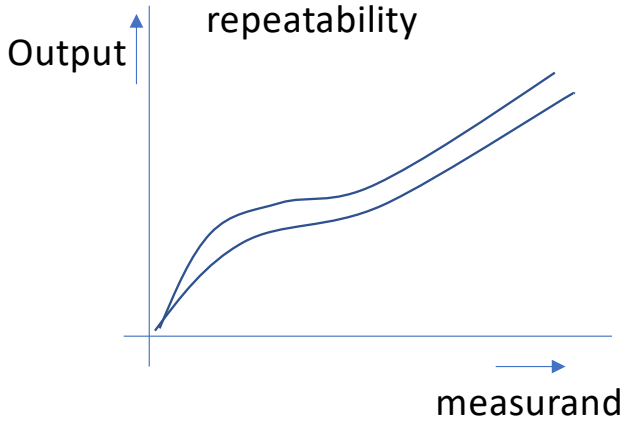
In reality we don't have ideal sensors available

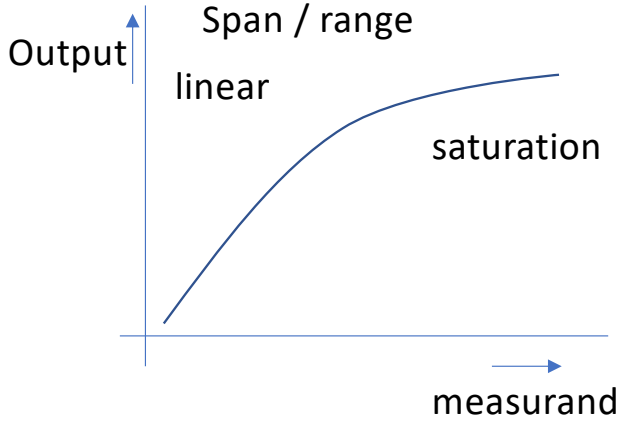








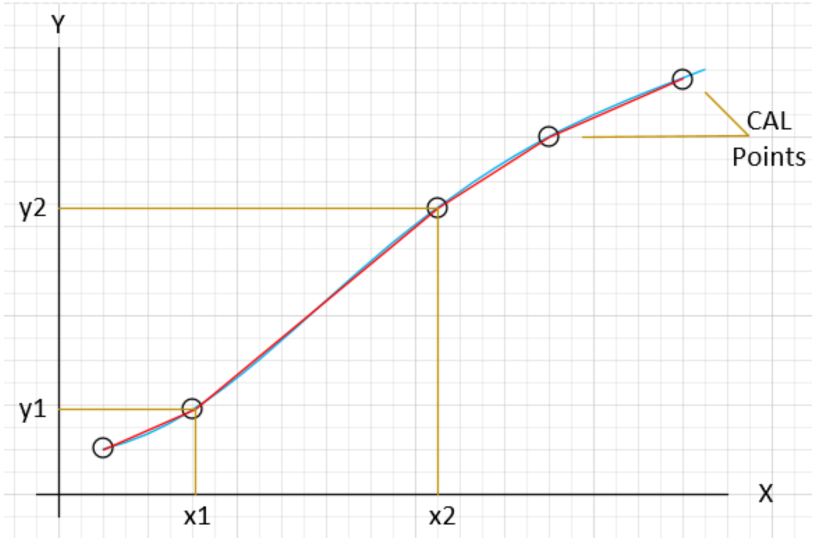




Linearization of a non-linear sensor

- **The specification (e.g. in the data sheet) of a sensor show the following diagram**
- **The data sheet provides some CAL point values**

- **We can use the linear interpolation between two known CAL points to get the corresponding output value**



$$y = \frac{y2 - y1}{x2 - x1}(x - x2) + y2$$

Example:
You are using a temperature sensor, and you get a measurement a current value of 5 mA.
The nonlinear diagram of the 'nonlinear sensor' is known
What temperature correspond to the 5 mA.

Test the formular with the value of y1=10^0C, y2=10^0C, x1 = 1mA, x2=10mA

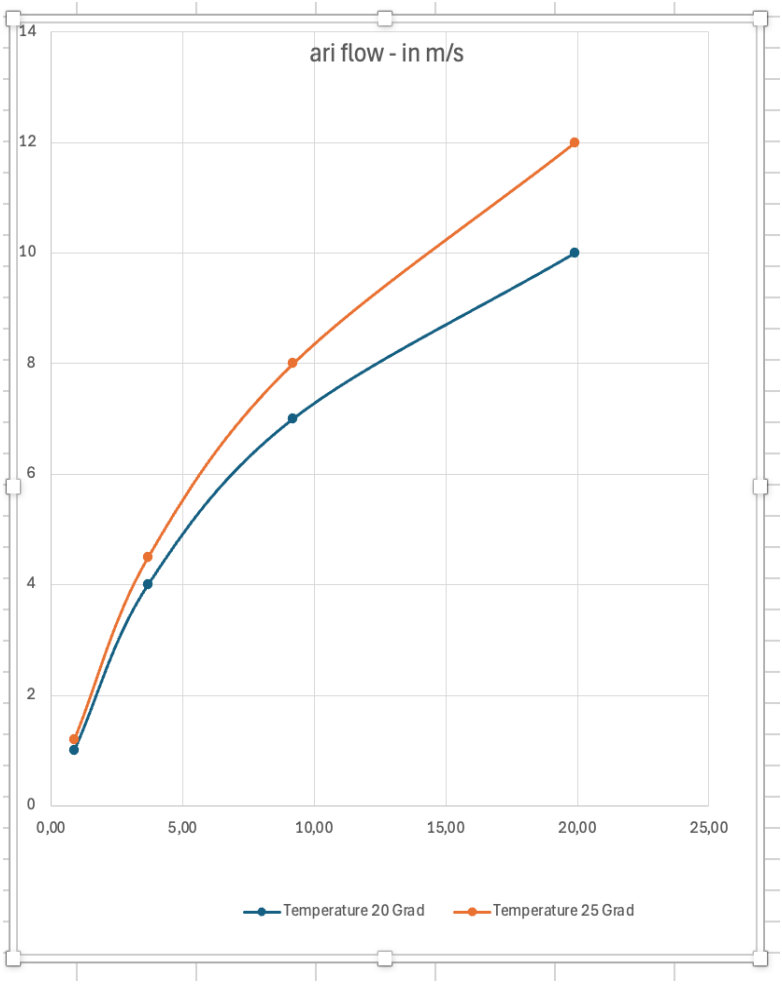
Actual measurement x=5 mA

Another example - Air Flowmeter

CAL Points

x in Volt	m/sec at T=20Gra	m/sec at T= 25 Grad
0,90	1	1,20
3,70	4	4,50
9,20	7	8,00
19,90	10	12,00

Which output value correspond to x= 4,5V at a Temperature of 20^0 ?



How to handle nonlinearity of sensor in IoT

- **It is an important topic in IoT, because most of the sensors show nonlinearity!!**
    - **Result in inaccuracies in data**
        - **Need to corrected**
- **Correction methods**
    - **Calibration**
        - **Compare the sensor data to reference values**
    - **Lookup Tables**
        - **Precomputed values of the sensor for specific input**
    - **Linear Approximation**
        - **Sensor curve is divided into small segments and for each segment a linear interpolation is applied**
    - **Polynomial Regression**
        - **Extending a linear regression to a polynomial regression often better fit to complex relations**
    - **Machine learning Models**
    - **Filtering and Smoothing**
    - **Sensor Fusion**
        - **Combine data from multiple sensors and get a more accurate overview**
    - **…..**

Short python program to deal with nonlinearity

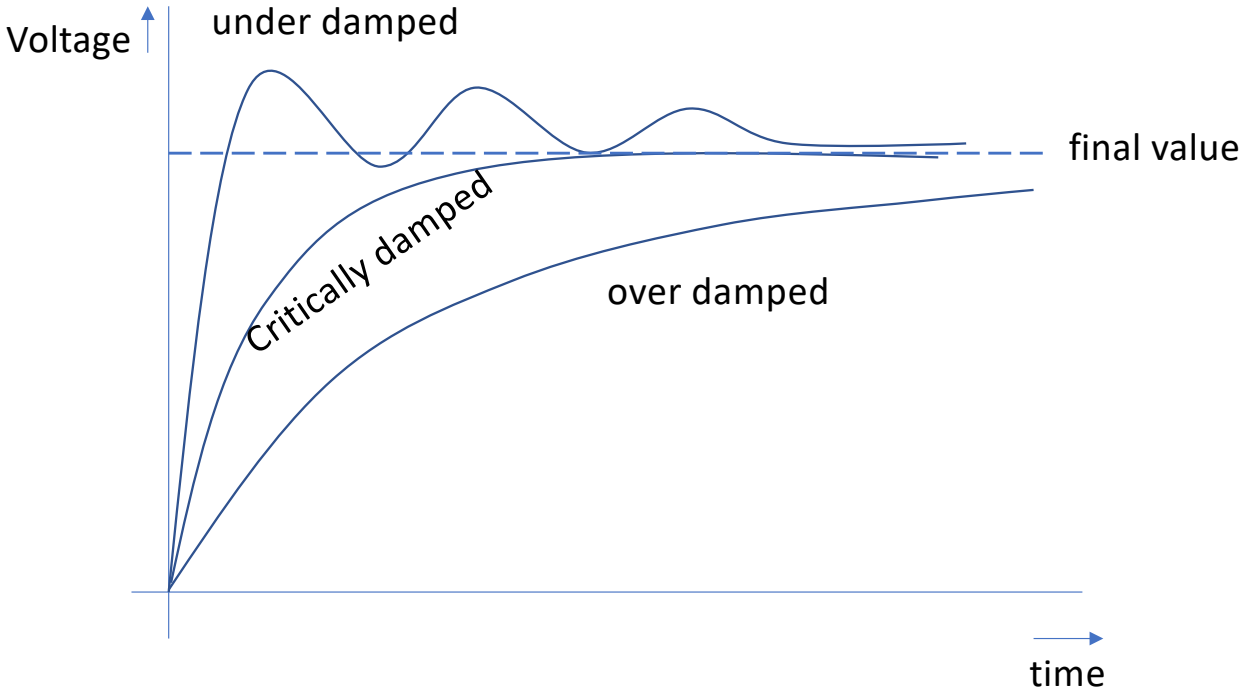
```
import numpy as np

# Example calibration data: raw sensor reading vs actual value
raw_readings = np.array([0, 1, 2, 3, 4, 5]) # Raw sensor outputs
actual_values = np.array([0, 0.9, 2.1, 3.05, 4.1, 5.2]) # True values

# Fit a polynomial (e.g., 2nd degree) to calibration data
coefficients = np.polyfit(raw_readings, actual_values, 2)

def correct_sensor_reading(raw_value, coeffs):
# Polynomial correction
corrected_value = np.polyval(coeffs, raw_value)
return corrected_value

# Example usage:
raw_sensor_value = 2.5
corrected_value = correct_sensor_reading(raw_sensor_value, coefficients)

print(f"Raw sensor value: {raw_sensor_value}")
print(f"Corrected sensor value: {corrected_value}")
```

You can test it using a jupyter notebook

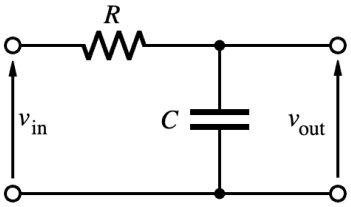
```
Raw sensor value: 2.5
Corrected sensor value: 2.537500000000001
```

Sensor dynamics

- **Real sensors need time to collect signals or to respond to a signal!!**
- **This will be influenced by:**
  - **Signal conditioning**
  - **Filtering**
  - **Sampling rate**
- ***Sensor dynamics influence how fast we can do measuring with the sensor***
- **Damping is used to stabilize the sensor output by reducing oscillation, noise reduction …. (see diagram)**
- **Damping can be achieved by electronic and software algorithm**
  - **See low pass filter implemented in (python, java, …..),**
  - **In electronic – RC circuit**



Sensor dynamics (e.g. Accelerator sensors show such behaviour)



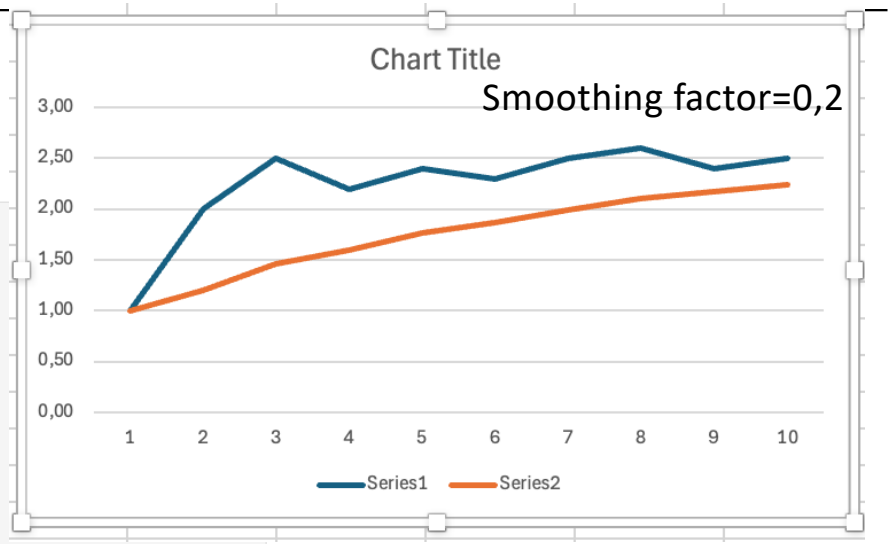
RC electronic circuits will be used to avoid over dumping (1st order)

There are more sophisticated circuits available to handle over dumping 2nd 3rd … order

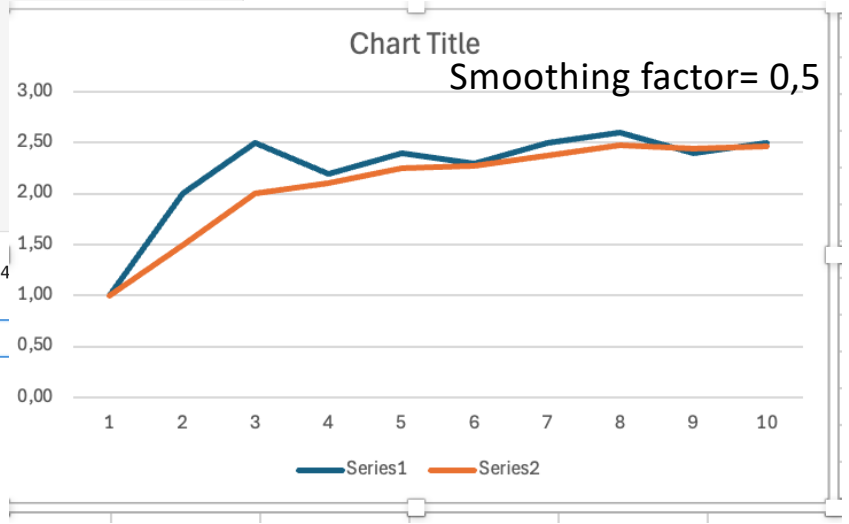
- **Temperature Sensors (e.g., thermistors, RTDs):**
  When placed in environments with rapid temperature changes or poor contact, they may oscillate around a stable value.
- **pH Sensors:**
  These sensors can oscillate due to unstable chemical conditions, electrolyte depletion, or improper calibration.
- **Pressure Sensors:**
  In dynamic environments or with improper mounting, pressure sensors can exhibit oscillation or noise in their readings.
- **Accelerometers and Gyroscopes:**
  When mounted loosely, subjected to vibrations, or in unstable environments, these can produce oscillatory signals.

Python low pass filter

```
[3]: def low_pass_filter(new_sample, previous_value, alpha):
         """
         Apply an exponential smoothing low-pass filter.

         Parameters:
             new_sample (float): The latest sensor reading.
             previous_value (float): The previous filtered value.
             alpha (float): Smoothing factor between 0 and 1 (closer to 0 for more smoothing).

         Returns:
             float: The new filtered value.
         """
         return alpha * new_sample + (1 - alpha) * previous_value

     # Example usage
     sensor_readings = [1.0, 2.0, 2.5, 2.2, 2.4, 2.3, 2.5, 2.6, 2.4, 2.5]
     alpha = 0.2  # Smoothing factor
     filtered_value = sensor_readings[0]  # Initialize with first reading

     filtered_readings = [filtered_value]
     for reading in sensor_readings[1:]:
         filtered_value = low_pass_filter(reading, filtered_value, alpha)
         filtered_readings.append(filtered_value)

     print("Original readings:", sensor_readings)
     print("Filtered readings:", filtered_readings)

     Original readings: [1.0, 2.0, 2.5, 2.2, 2.4, 2.3, 2.5, 2.6, 2.4, 2.5]
     Filtered readings: [1.0, 1.2000000000000002, 1.4600000000000002, 1.608, 1.7664000000000002, 1.8731200000000001, 1.9984
     8, 2.17503744, 2.2400299520000004]

[ ]:
```

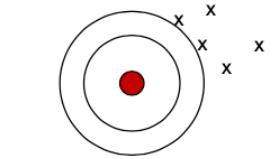


You can try it by using different smoothing factors

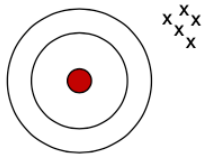


All sensor are not ideal …. but they are useful

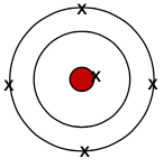
- **Accuracy vs Precision (Treffergenauigkeit vs Richtigkeit)**
  - **Accurate**
    - **Average of sampled outputs is close to the real value**
  - **Precise**
    - **Sampled out is consistently tightly grouped with consistent offset from real value**



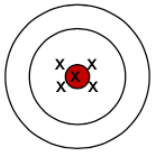
Imprecise and Inaccurate



Precise but Inaccurate



Accurate but Imprecise



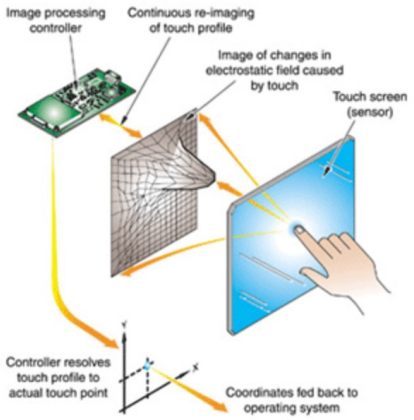
Precise and Accurate

Characteristics you should be aware of to select the right sensor for your app

- Linearity
- Noise
- Sample rate
- Resolution

The physical principles of sensing

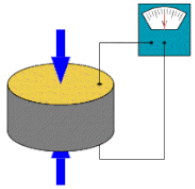
- **Electrical charges, fields and potentials**
- **Capacitive sensing**
- **Magnetic sensing**
- **Electromagnetic sensing**
- **Induction sensing**
- **Resistive sensing**



Infineon's 3D magnetic sensors are the perfect fit for control elements, joysticks and E-meters (anti-tampering), furthermore it can be use in **smart home** and **industrial control applications**

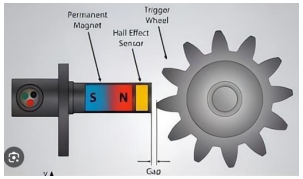
The physical principles of sensing

- **Piezoelectric effect**
  - **E.g. used in:**
    - **vibration and shock monitoring**
    - **Structural health monitoring (bridges, buildings, pipelines to detect stress, strain and crack over time**
    - **Pressure sensing**
    - **Energy harvesting (think on 'Feuerzeuge')**
    - **Medical health (heart rate, muscle activation, ….)**
    - **…**



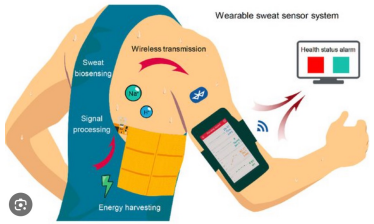
- **Photoelectric sensing**
  - **E.g use in:**
    - **Object detection and proximity sensing in industrial automation, manufacturing, robotics**
    - **Smart parking systems**
    - **Automated doors and gates**
    - **Smart lightning and presence detection**
    - **Level sensing container**
    - **….**

- **Hall effect**
  - **E.g. used in:**
    - **Detect the position of moving parts**
    - **Speed and rotation monitoring**
    - **Current sensing**
    - **Linear and rotary motion detection**
    - **Tacho- and Speedometer**
    - **Smart Meter and Utility Monitoring**
    - **Energy Harvesting**
    - **Water flow**



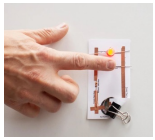
The physical principles of sensing

- **Thermoelectric effect**
  - **E.g. used in:**
    - **wearable devices to monitor body temperature**
    - **generate energy from body heat for low power devices**
    - **smart agriculture – generate energy**
    - **smart building – monitor temperature differences in heating, ventilation, air conditioning**
    - **automotive app – capture waste heat from engines, ….**
    - **home app – smart refrigerators, fire detection, ….**



- **Sound waves**
  - **E.g. used in:**
    - **Monitor sounds and vibrations emitted by machines / motors etc.**
    - **Glass break detection**

- **Light sensing**
  - **…..**



Nearly everybody uses a smartphone – which sensors are used in a smartphone

- **What do you think?**

- **Demo phyphox (sensor values on my iPhone)**

The top 10 sensor types used in IoT solutions

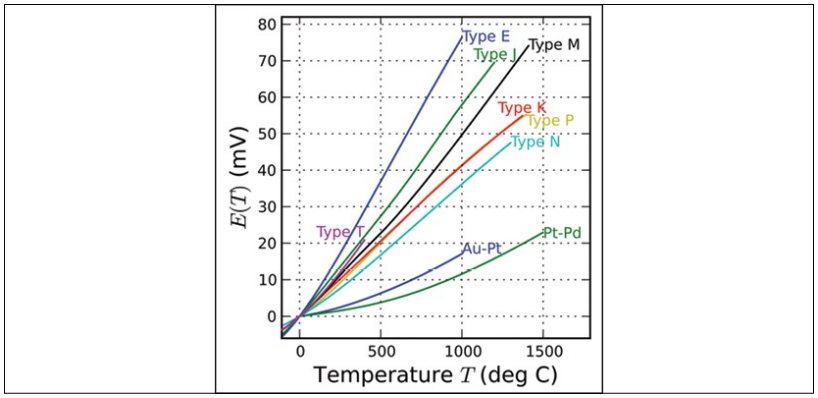
- **Temperature sensors**
  - **Measure the heat energy in a source**
- **Humidity sensors**
  - **Measure amount of water vapor in the air or gases or …..**
- **Pressure sensors**
  - **Sense changes in gases and liquids**
- **Proximity sensors**
  - **Non contact detection of objects**
- **Level sensors**
  - **Detect the level of liquids, powders and granular materials.**

The top 10 sensor types used in IoT solutions

- **Accelerometers**
  - **Measure the acceleration of an object or the change of the velocity in a timeframe**
- **Gyroscope**
  - **Measure the angular rate of velocity around an axis**
- **Gas detection sensors**
  - **Monitor and detect changes in air quality, including the presence of toxic, combustible and hazardous gasses**
- **Infrared sensors**
  - **Sense characteristics in their surroundings by either emitting or detecting infrared radiation**
- **Optical sensors**
  - **Convert light into electrical signals**

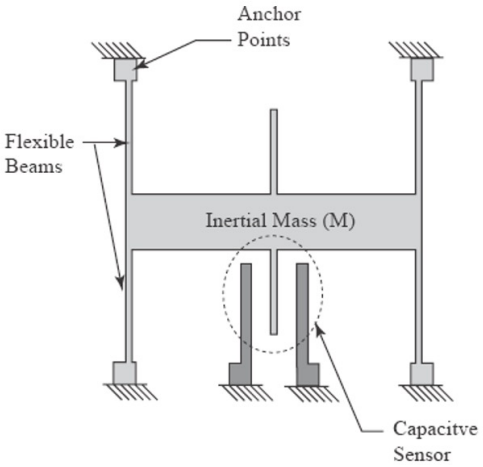
Temperature sensing and Thermocouples

- **Temperature sensor exist nearly everywhere**
  - **Could storage logistics, refrigerators to industrial machinery,**
- **Thermocouples**
  - **Produce a very small signal – two wires are connected at the end, a small voltage signal is produced on heating. They are good for wide temperature ranges.**

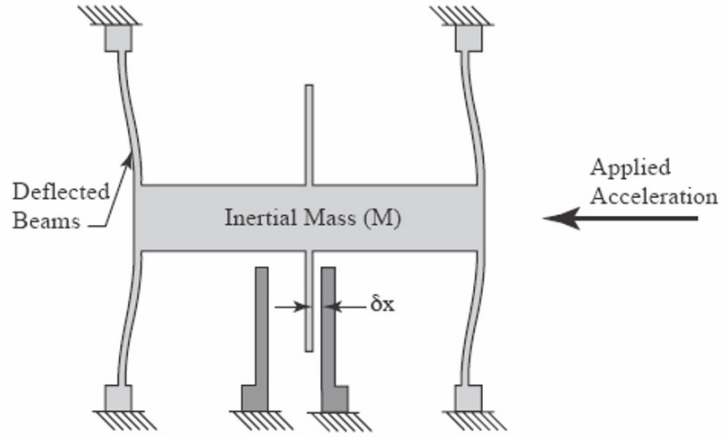


Category	Thermocouples	RTDs	Thermistors
Temperature range (degrees Celsius)	-180 to 2,320	-200 to 500	-90 to 130
Response time	Fast (microseconds)	Slow (seconds)	Slow (seconds)
Size	Large (~1 mm)	Small (5 mm)	Small (5 mm)
Accuracy	Low	Medium	Very high

Accelerator

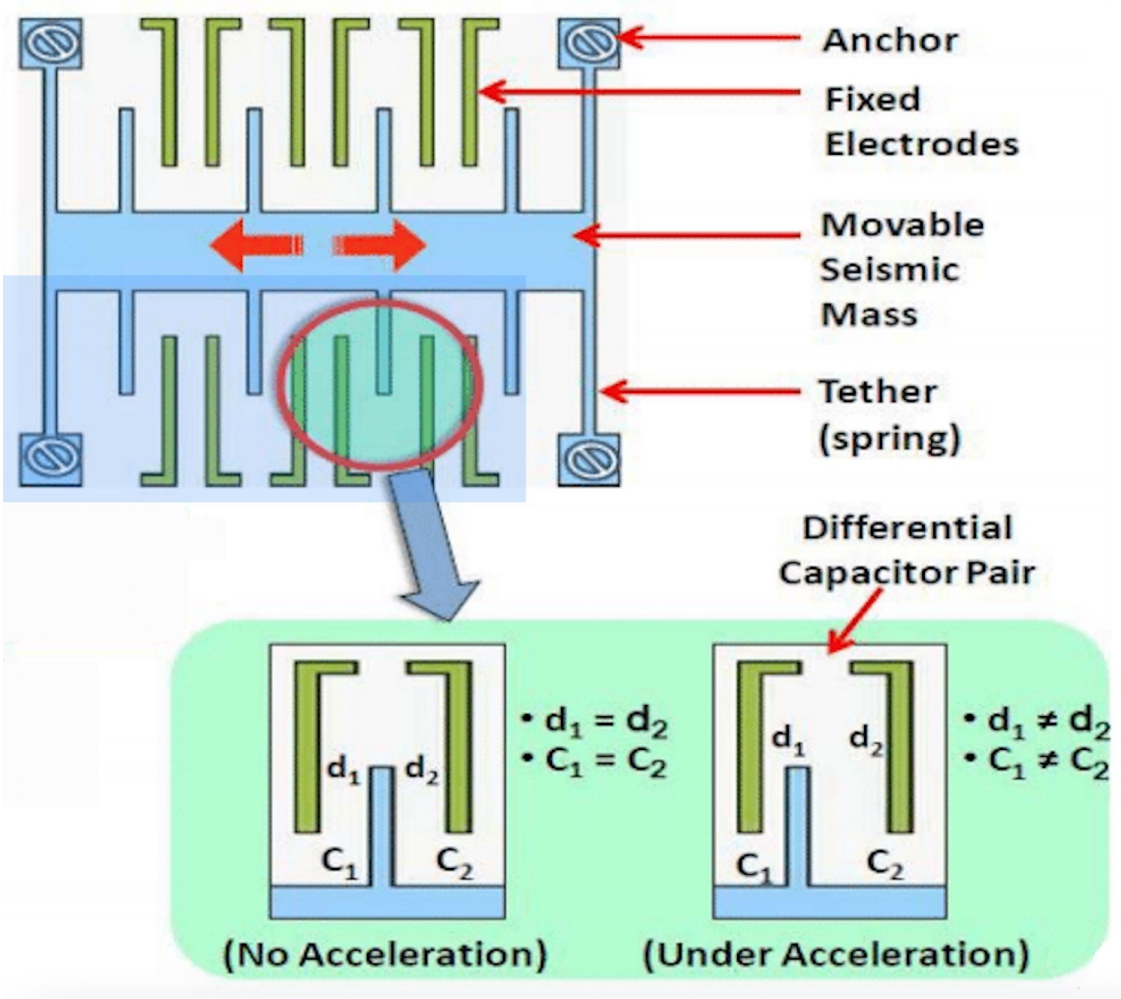


It consists of beams and a capacitive sensor with some anchor points



On applying the acceleration, the beams deflect and cause the change in capacitance.

Accelerator



Sample Product Specification The KSC7-1050

PROPRIETARY TECHNOLOGY

These high-performance silicon micromachined linear accelerometers and inclinometers consist of a sensor element and an ASIC packaged in a 3x3x0.9mm Land Grid Array Dual (LGA). The sensor element is fabricated from single-crystal silicon with proprietary Deep Reactive Ion Etching (DRIE) processes, and is protected from the environment by a hermetically-sealed silicon cap at the wafer level.

The KXSC7 series is designed to provide a high signal-to-noise ratio with excellent performance over temperature. These sensors can accept supply voltages between 1.8V and 3.6V. Sensitivity is factory programmable allowing customization for applications requiring from ±1.5g to ±6g ranges. Several pre-set internal low pass filters can eliminate the need for external filter capacitors. If the pre-set values are not optimal for an application, the sensor bandwidth is user-definable with the use of external capacitors.

The sensor element functions on the principle of differential capacitance. Acceleration causes displacement of a silicon structure resulting in a change in capacitance. An ASIC, using a standard CMOS manufacturing process, detects and transforms changes in capacitance into an analog output voltage, which is proportional to acceleration. The sense element design utilizes common mode cancellation to decrease errors from process variation and environmental stress.

One of the 1st accelerometer sensor for smart phone

Sample Product Specification

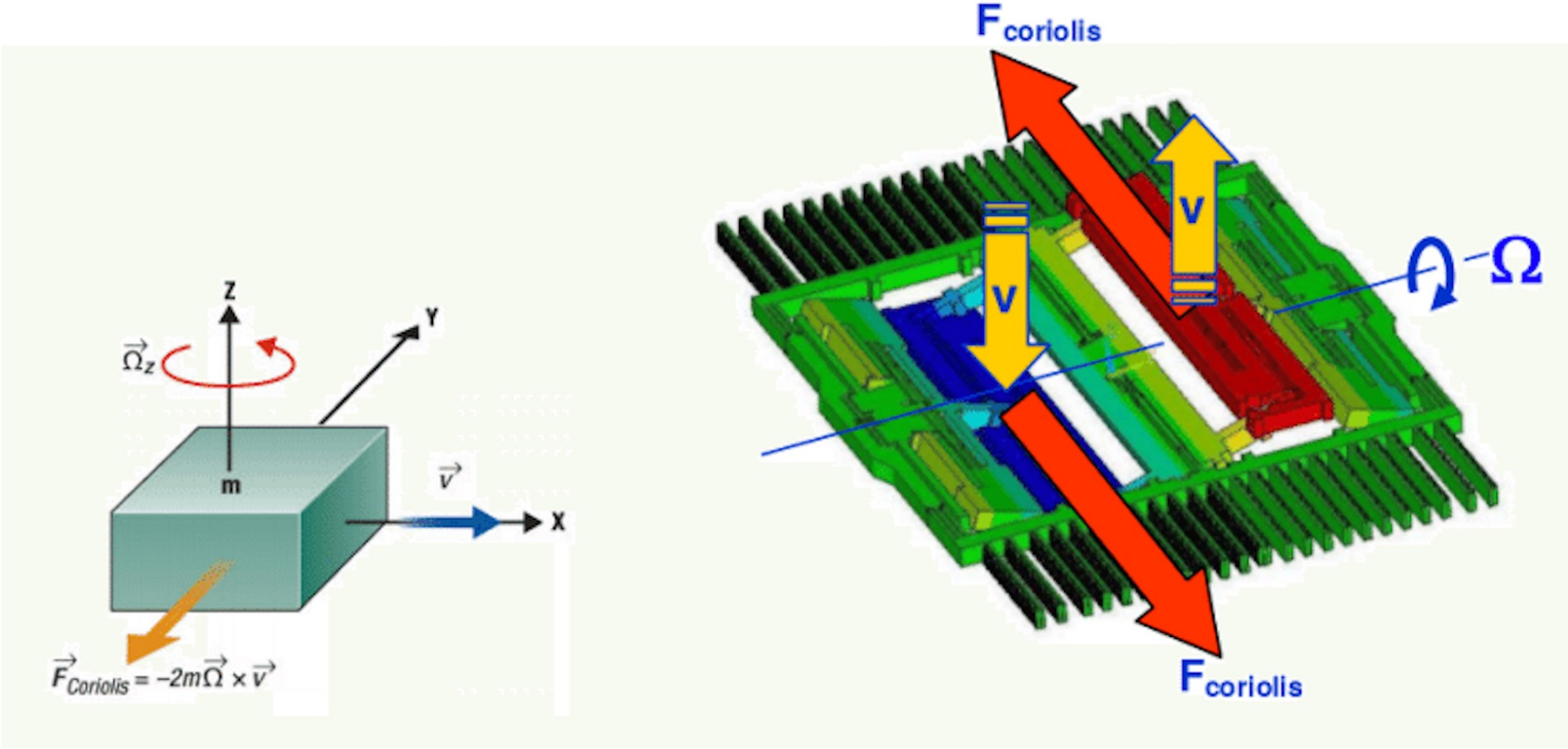
Product Specifications

Table 1. Mechanical

(specifications are for operation at 2.8V and T = 25C unless stated otherwise)

Parameters	Units	Min	Typical	Max
Operating Temperature Range	ºC	-40	-	85
Zero-g Offset	V	1.26	1.4	1.54
Zero-g Offset Variation from RT over Temp.	mg/ºC		0.5 (xy) 3 (z)	
Sensitivity	mV/g	543	560	577
Sensitivity Variation from RT over Temp.	%/ºC		0.01 (xy) 0.04 (z)	
Offset Ratiometric Error (V_{dd} = 2.8V ± 5%)	%		0.3	
Sensitivity Ratiometric Error (V_{dd} = 2.8V ± 5%)	%		1 (xy) 0.6 (z)	
Self Test Output change on Activation	g	0.7 0.6 0.02	0.9 (x) 0.8 (y) 0.2 (z)	1.2 1 0.7
Mechanical Resonance (-3dB)[1]	Hz		4000 (xy) 2000 (z)	
Non-Linearity	% of FS	0	0.1	0.2
Cross Axis Sensitivity	%		2	
Noise Density (on filter pins)	µg / √Hz		125	
Motion Interrupt Threshold	g		1.5	

Gyroscope


Z
Y
X
m
$\vec{\Omega}_z$
$\vec{v}$
$\vec{F}_{Coriolis} = -2m\vec{\Omega} \times \vec{v}$
F coriolis
v
v
Ω
F coriolis

Bosch BNO055 Intelligent 9 axis absolute orientation sensor

Bosch BNO055 Intelligent 9 axis absolute orientation sensor

OUTPUT SIGNAL ACCELEROMETER (ACCELEROMETER ONLY MODE)

Parameter	Symbol	Condition	Min	Typ	Max	Units
Sensitivity	S	All g_{FSXg} Values, T_A=25°C		1		LSB/mg
Sensitivity tolerance	S_{tol}	Ta=25°C, g_{FS2g}		±1	±4	%
Sensitivity Temperature Drift	TCS	g_{FS2g}, Nominal V_{DD} supplies, Temp operating conditions		±0.03		%/K
Sensitivity Supply Volt. Drift	S_{VDD}	g_{FS2g}, T_A=25°C, $V_{DD\_min} \le V_{DD} \le V_{DD\_max}$		0.065	0.2	%/V
Zero-g Offset (x,y.z)	Off_{xyz}	g_{FS2g}, T_A=25°C, nominal V_{DD} supplies, over life-time	-150	±80	+150	mg
Zero-g Offset Temperature Drift	TCO	g_{FS2g}, Nominal V_{DD} supplies		±1	+/-3.5	mg/K
Zero-g Offset Supply Volt. Drift	Off_{VDD}	g_{FS2g}, T_A=25°C, $V_{DD\_min} \le V_{DD} \le V_{DD\_max}$		1.5	2.5	mg/V
Bandwidth	bw_8	2nd order filter, bandwidth programmable		8		Hz
	bw_{16}			16		Hz
	bw_{31}			31		Hz
	bw_{63}			63		Hz
	bw_{125}			125		Hz
	bw_{250}			250		Hz
	bw_{500}			500		Hz
	bw_{1000}			1,000		Hz

Onboard A/D unit

BST-BNO055-DS000-18 | Revision 1.8 | October 2021 Bosch Sensortec

High performance IoT endpoints

- **Beside the ‘simple sensors’ there exist also high performance combined with sustainable processing power sensors**

- **E.g. Vision Systems**

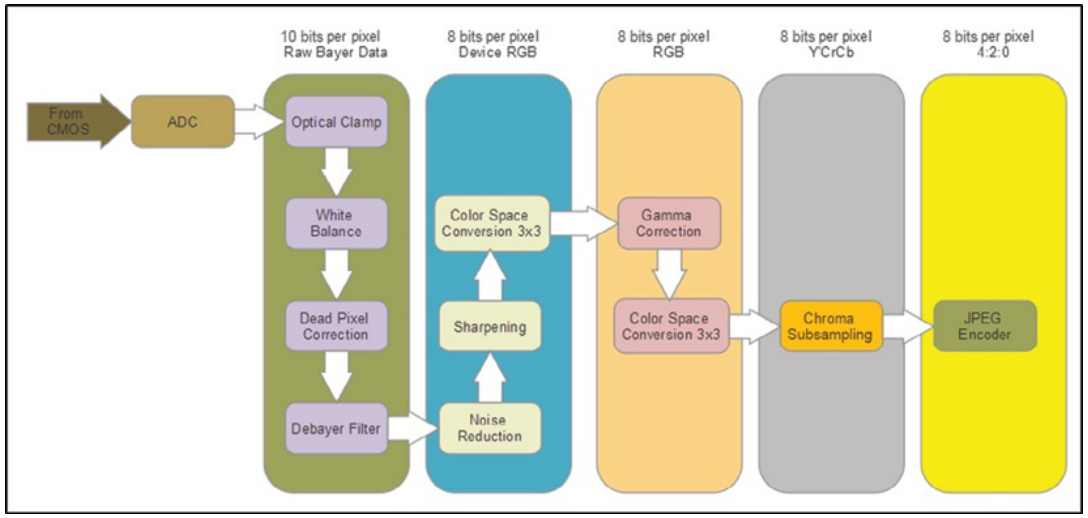


Image sensor: Typical image signal processor pipeline for color video

Analog-to-digital conversion:.
Optical clamp: Removes sensor biasing effects
Dead pixel correction: Dead pixels are replaced with the average of neighboring pixels.
Debayer filtering and demosaicing: Debayer filtering separates RGB data
Noise reduction:
Sharpening: Applies a blur to an image using a matrix multiplication, and then combines the blur with detail in content regions to create a sharpening effect.
Color space conversion 3 x 3: Color space conversion to RGB data for RGB-particular treatments.
Gamma correction: Corrects for CMOS image sensor nonlinear response on RGB data to different irradiance.
Color space conversion 3 x 3: Additional color space conversion from RGB to Y'CbCr format.
Chroma subsampling: Due to nonlinearities in RGB tones, corrects images to mimic other media such as film for tonal matching and quality.
JPEG encoder: Standard JPEG compression algorithm.

Output device - Actors

- **Output devices in the IoT ecosphere can be just about anything, from a simple LED to a full video system.**

- **Stepper motors**
- **Speaker**
- **Audio and Video Systems (needs output hardware and buffering capabilities)**
- **Relay**
- **Valves**