Machine Learning for IoT

Part-I: Technologies for the IoT

(HW perspective)

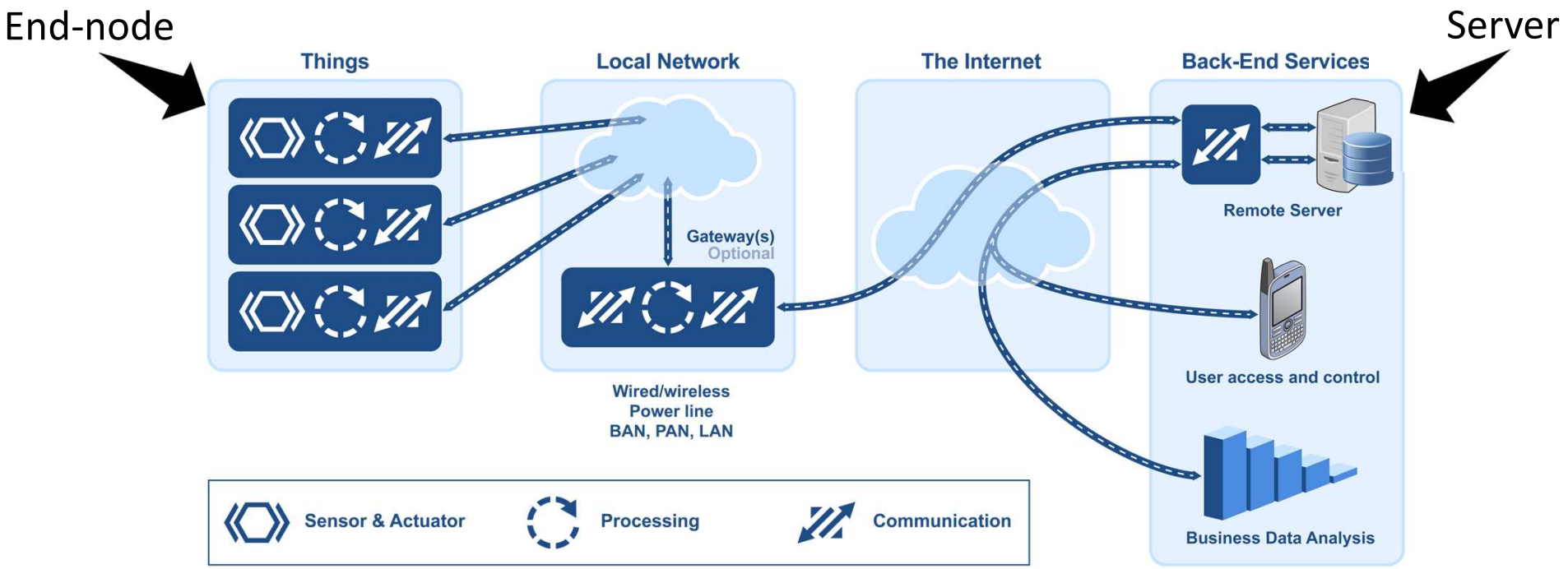
Objectives

- An overview of the HW devices in a typical IoT environment
- Taxonomy of HW components and their functionality
  - The ***functionality*** you can accommodate in your system
  - The computational power you need for a given ***functionality***
  - The ***power/energy*** sizing of your system
- Specifically, assuming you are not a “HW expert”

Contents

- Architectural overview
- Sensors (and actuators)
- Signal conditioning
- Processing units: MCU vs MPU
- Radio communication
- Beyond functionality: Energy path

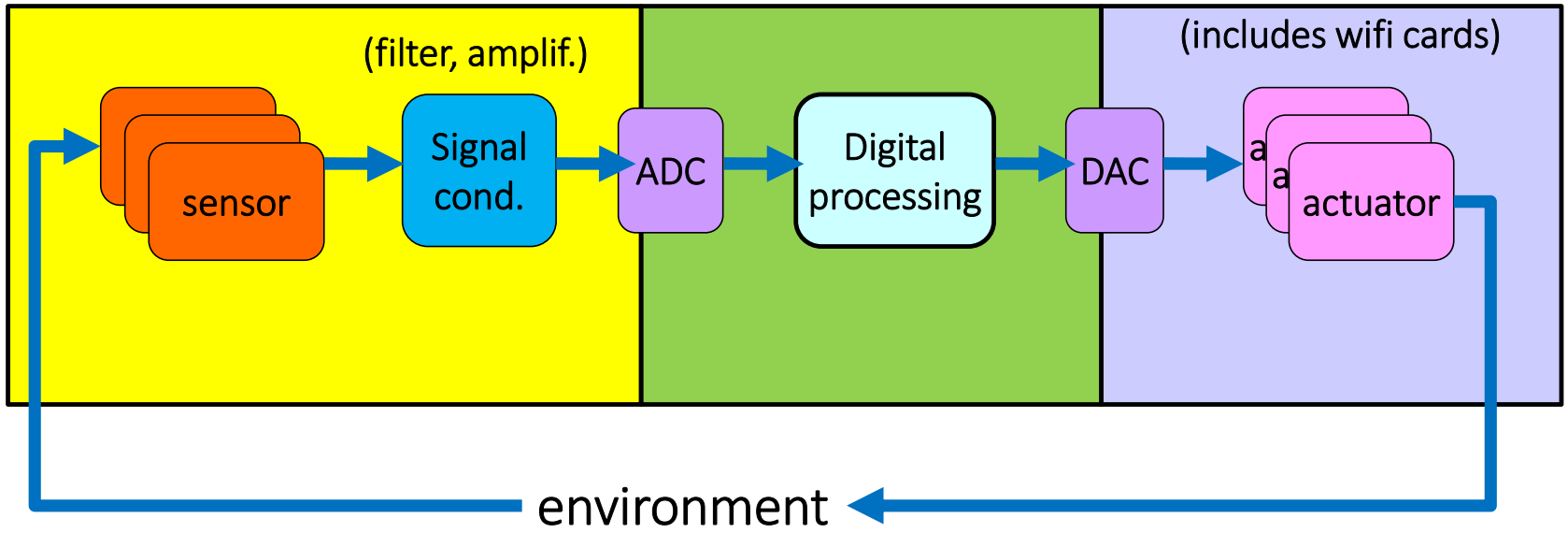
IoT infrastructure



End-nodes vs server/cloud nodes

- Form Factor
  - End-nodes: cm^2/mm^2
  - Server/Cloud: $(10^2)m^2$
- Functionality
  - End-nodes: Cyber-physical-system, with sensing, processing, acting capability, wifi connectivity
  - Server/cloud: (mostly) processing and storage, wired connectivity
- Performance
  - End-nodes: MHz/MFLOPs, MIPs
  - Server/cloud: GHz/many GFLOPs to TFLOPs
- Power/Energy budget
  - End-nodes: mw/W (<<7W)
  - Server/cloud: kW/MW

Functional view



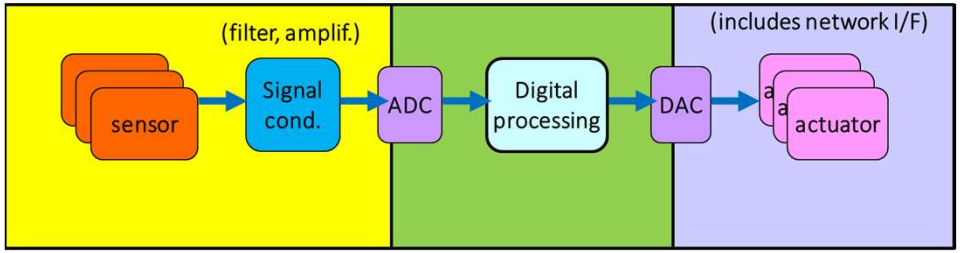
Sensing subsystem

Processing subsystem

Actuation (communication) subsystem

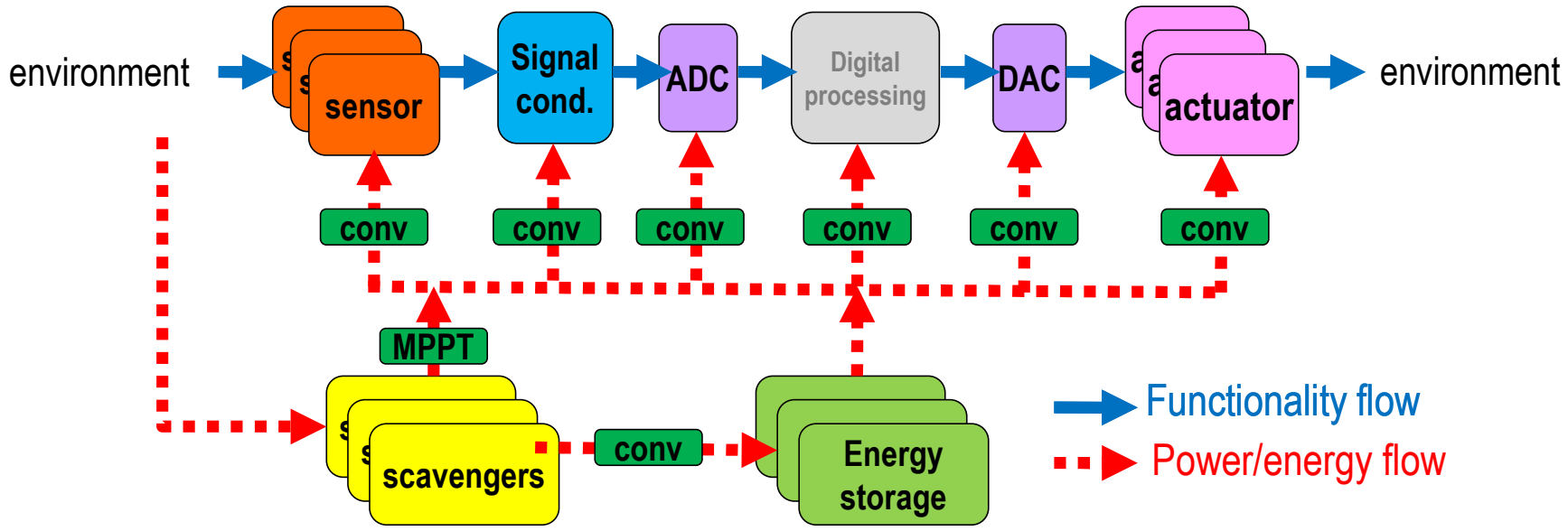
(Extra) Functionality

- This is a functional view
  - What the system does in terms of functionality
  - Connections represent flow of data



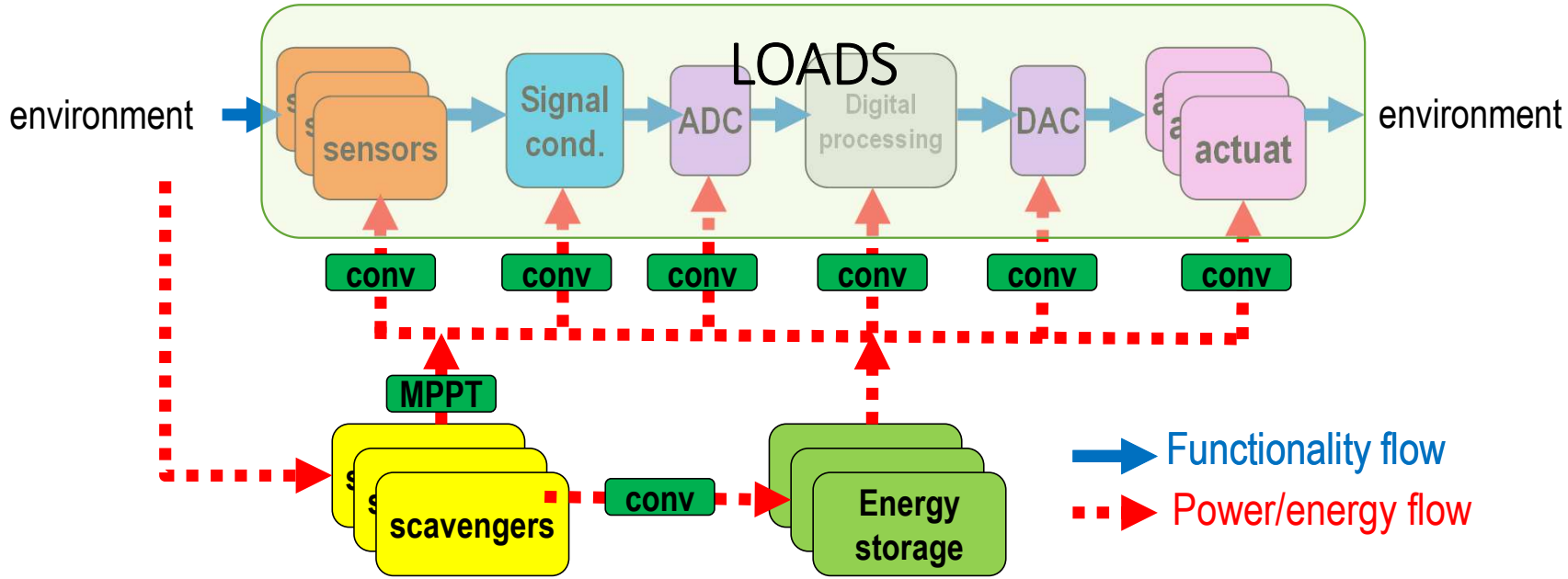
- What is missing?
- A Non-functional view
  - Most important non-functional information is the flow of energy/power
  - Implies adding other components

Functional view: info vs energy/power flows

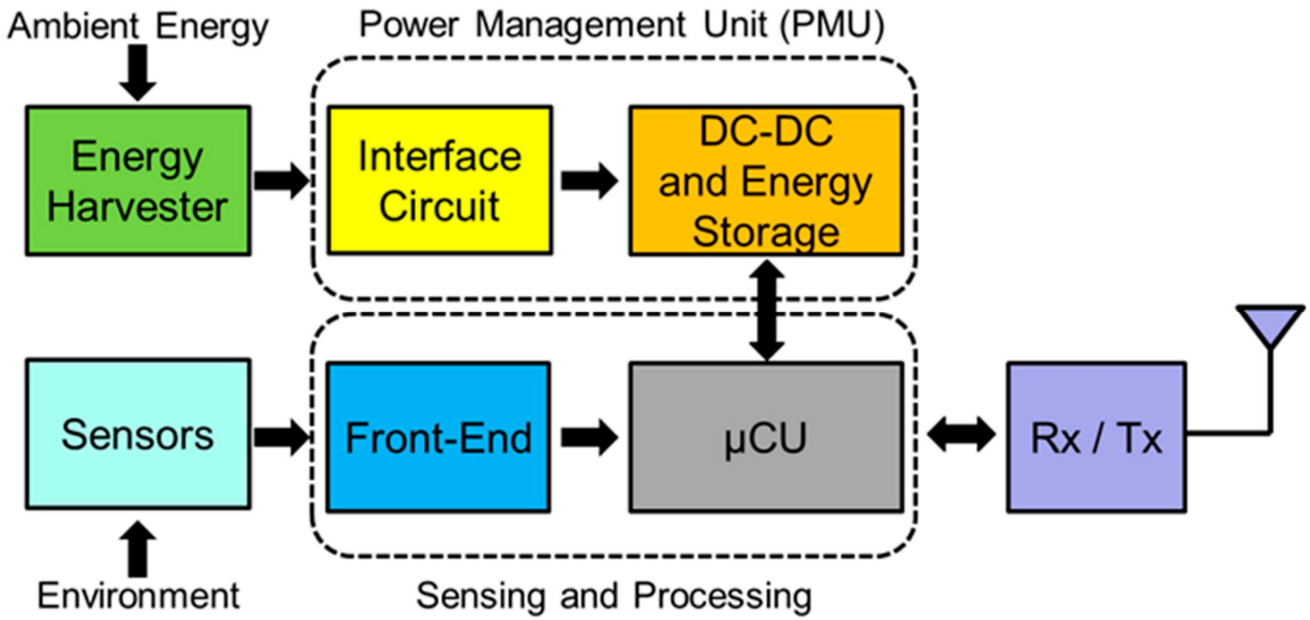


Functional view: info vs energy/power flows

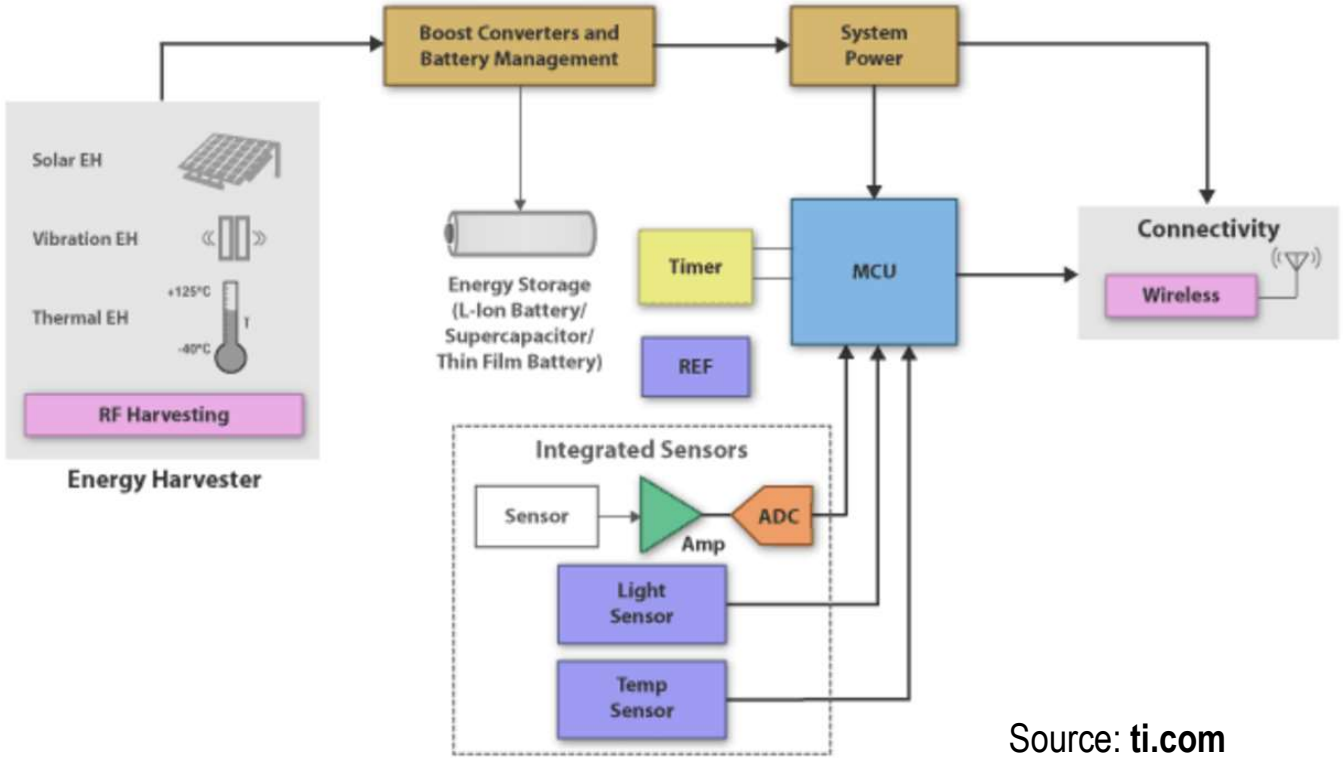
- Functional components are generic 'loads' from the power flow perspective!!!



Structural view



Integrated solutions



Source: **ti.com**

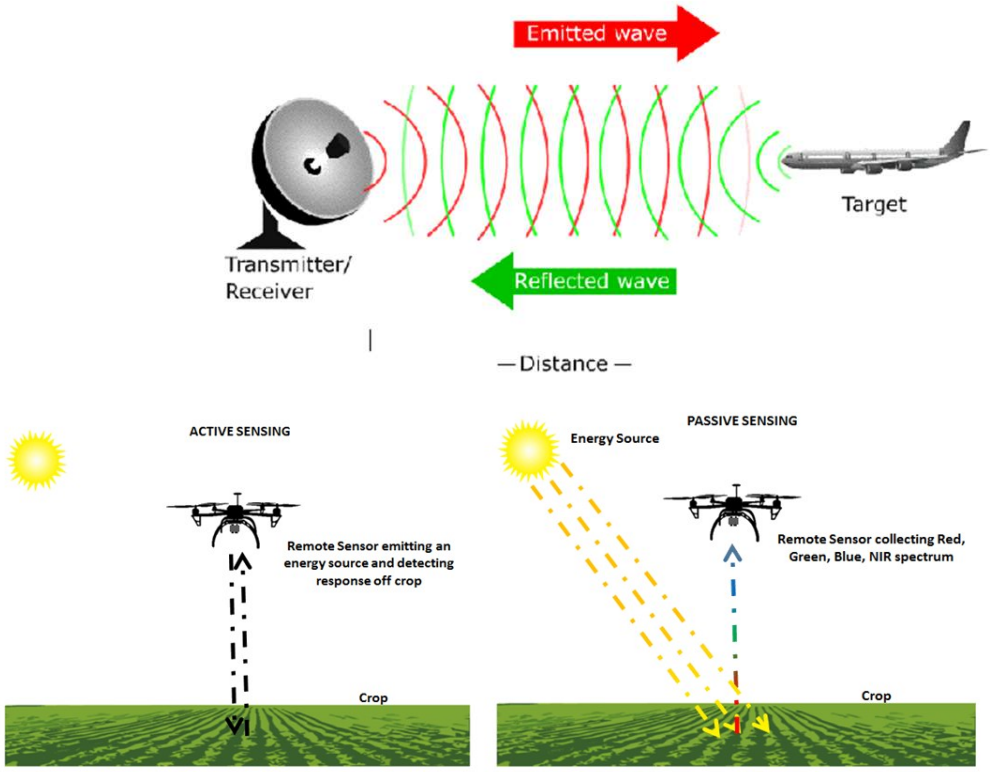
Lecture plan

- First explore the functional domain
  - Sensing subsystem
  - Processing subsystem
  - Communication subsystem
- Then the extra-functional (power) domain
  - Understand the issues involving energy
    - Consuming (as little as possible) energy
    - Generating energy
    - Storing energy
    - Convert energy
- Dive into computers architectures (as separate section)
  - Understand how commercial cores evolved over time
  - Taxonomy of existing programming/executions models and their characteristics
  - What are the options available today for complex ML-powered IoT

Sensors and acquisition

Active vs passive sensors

- Active = require an external power supply to operate, called an excitation signal which is used by the sensor to produce the output signal.
  - Example:
    - Radar
    - Lidar
- Passive = does not need any additional power source or excitation voltage.
  - They generates an output signal in response to some external stimulus
  - Example:
    - a thermocouple which generates its own voltage output when exposed to heat
    - Image sensor (camera)



https://towardsdatascience.com/why-tesla-wont-use-lidar-57c325ae2ed5

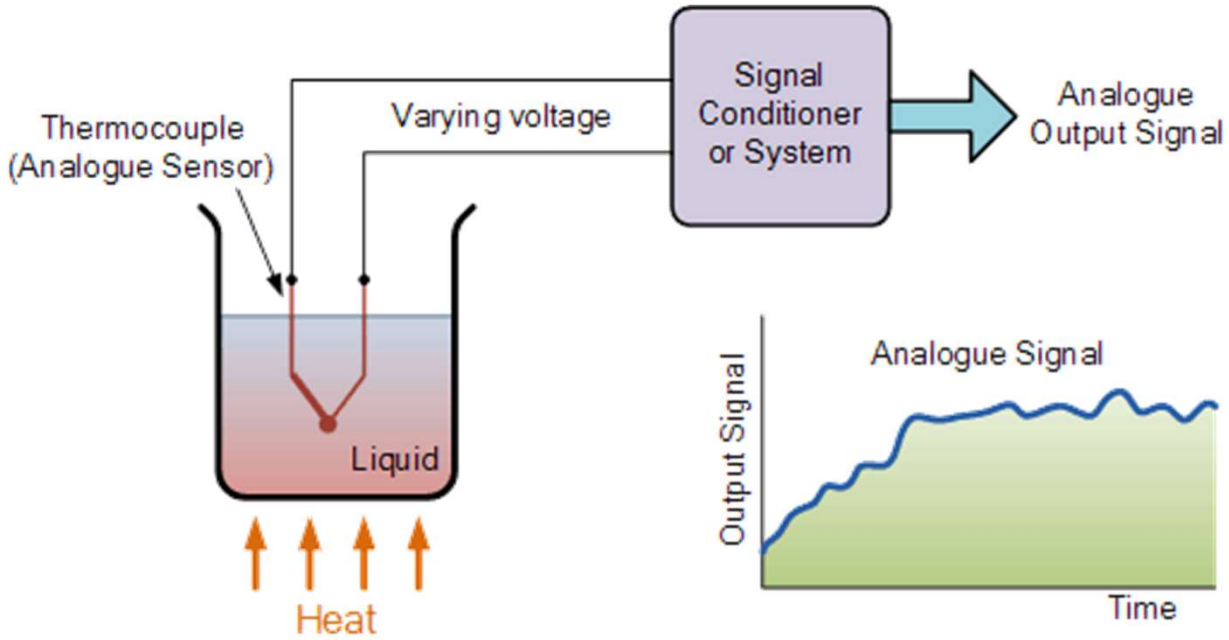
Analog vs Digital sensors

- Analog: produce a continuous output signal or voltage which is generally proportional to the quantity being measured (most common)

- Digital (or discrete): produce a discrete digital output signals or voltages that are a digital representation of the quantity being measured.

Analog sensor

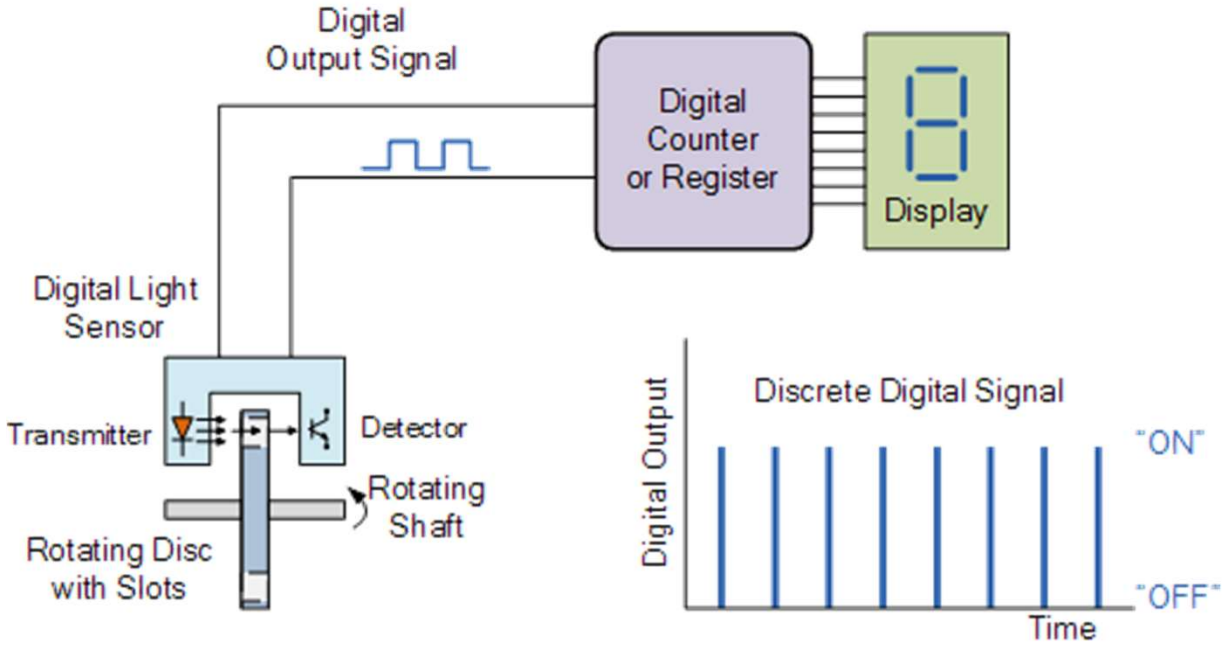
- Analog: Thermocouple used to produce an Analog Signal



[Source: electronics-tutorial.com]

Digital

- Digital: Light Sensor used to produce a Digital Signal



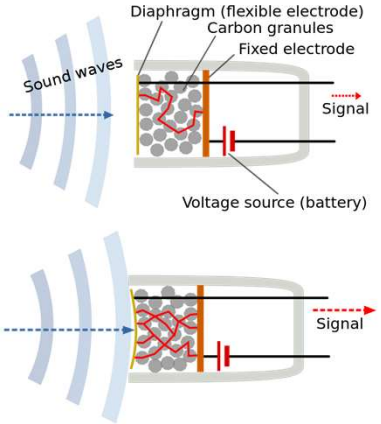
[Source: electronics-tutorial.com]

Sensing what?

- Basically everything (see list on Wikipedia…)
  - Acoustic, sound, vibration
  - Chemical (includes all biomedical ones)
  - Electric current, electric potential, magnetic, radio
  - Environmental
  - Flow, fluid velocity
  - Ionizing radiation, subatomic particles
  - Navigation
  - Position, angle, displacement, distance, speed, acceleration
  - Optical, light, imaging, photon
  - Pressure, Force, density, level
  - Thermal, heat, temperature
  - Proximity, presence

Some examples

- A simple one: a microphone
  - Can be very technological (e.g., a MEMS microphone)



RHLGA metal cap 4-lead
3.35 x 2.5 x 0.98 mm

Features

- Single supply voltage operation
- Fully differential output
- Omnidirectional sensitivity
- High signal-to-noise ratio
- High bandwidth
- Package compliant with reflow soldering
- High RF immunity
- ECOPACK®, RoHS, and "Green" compliant

Description

The MP23AB01DH is a compact, low-power microphone built with a capacitive sensing element and an IC interface.

The sensing element, capable of detecting acoustic waves, is manufactured using a specialized silicon micromachining process to produce audio sensors.

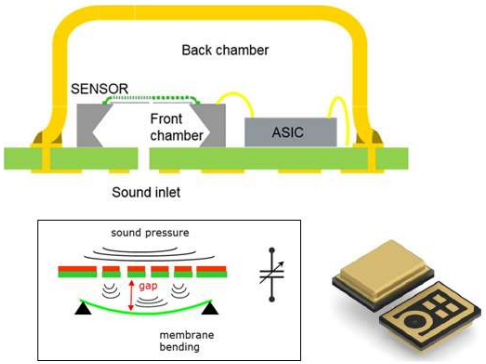
The MP23AB01DH has sensitivity of 38 dB ±1 dB, an acoustic overload point of 135 dBSPL with minimum 65 dB signal-to-noise ratio.

The MP23AB01DH has fully differential output in order to minimize common mode noise.

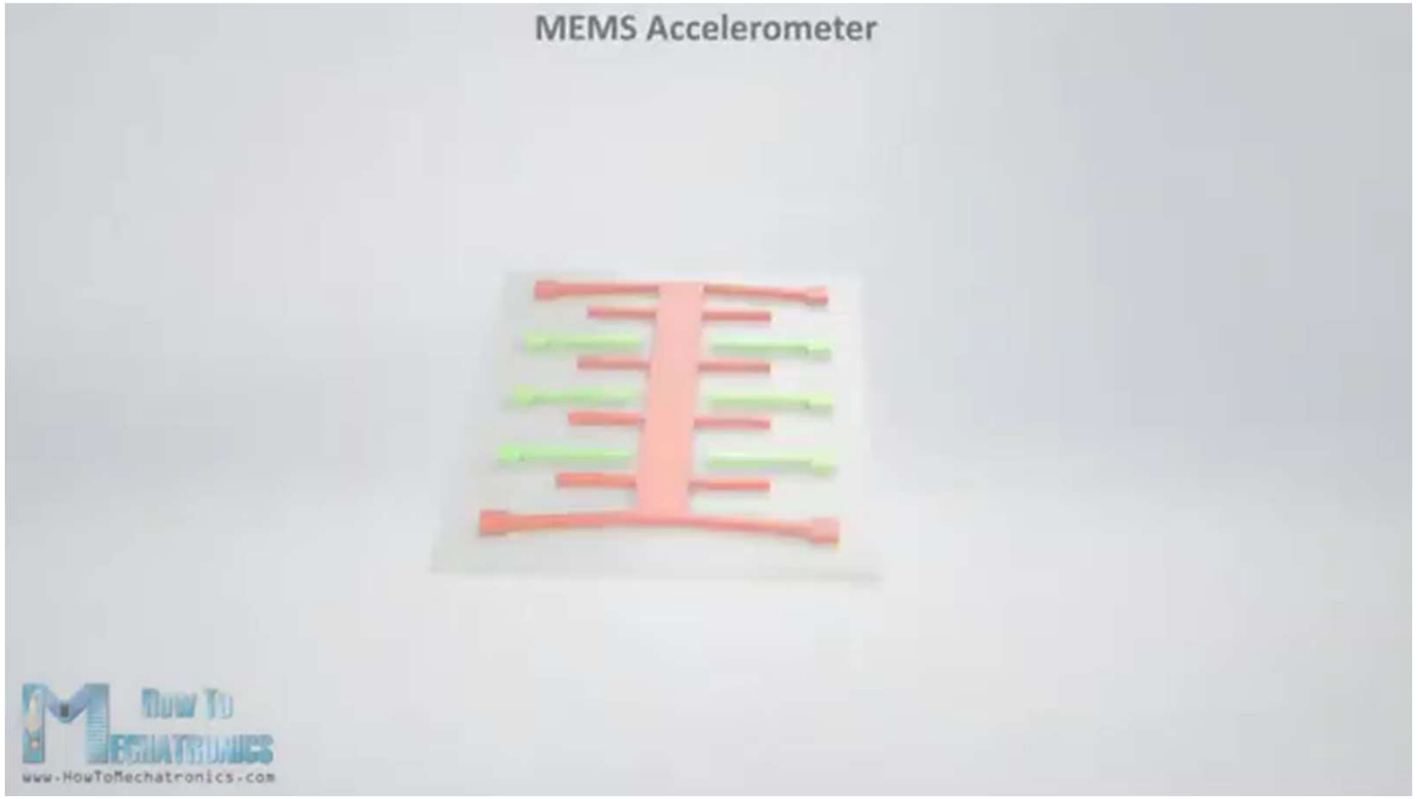
The MP23AB01DH is available in a package compliant with reflow soldering and is guaranteed to operate over an extended temperature range from -40 °C to +85 °C.

Table 1: Device summary

Order code	Temp. range (°C)	Package	Packing
MP23AB01DH	-40 to +85	(3.35 x 2.5 x 0.98) mm	Tray
MP23AB01DHTR	-40 to +85	(3.35 x 2.5 x 0.98) mm	Tape and reel

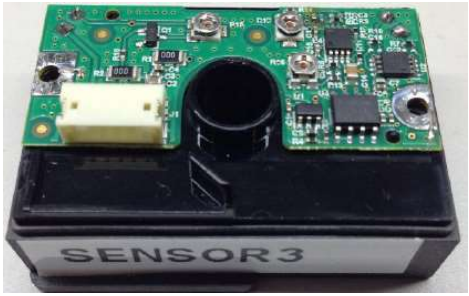


MEMS: MicroElectricalMechanicalSystems

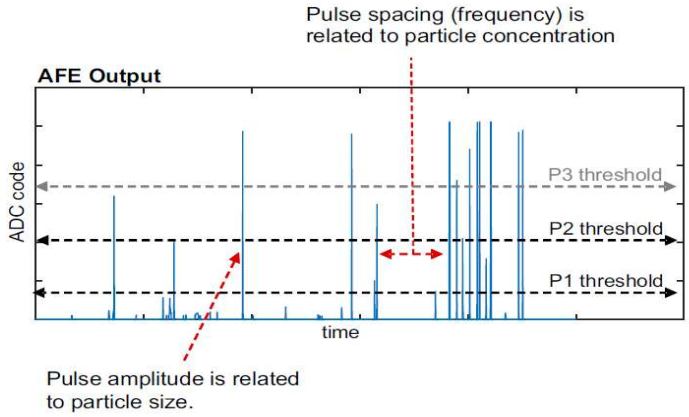


Some examples

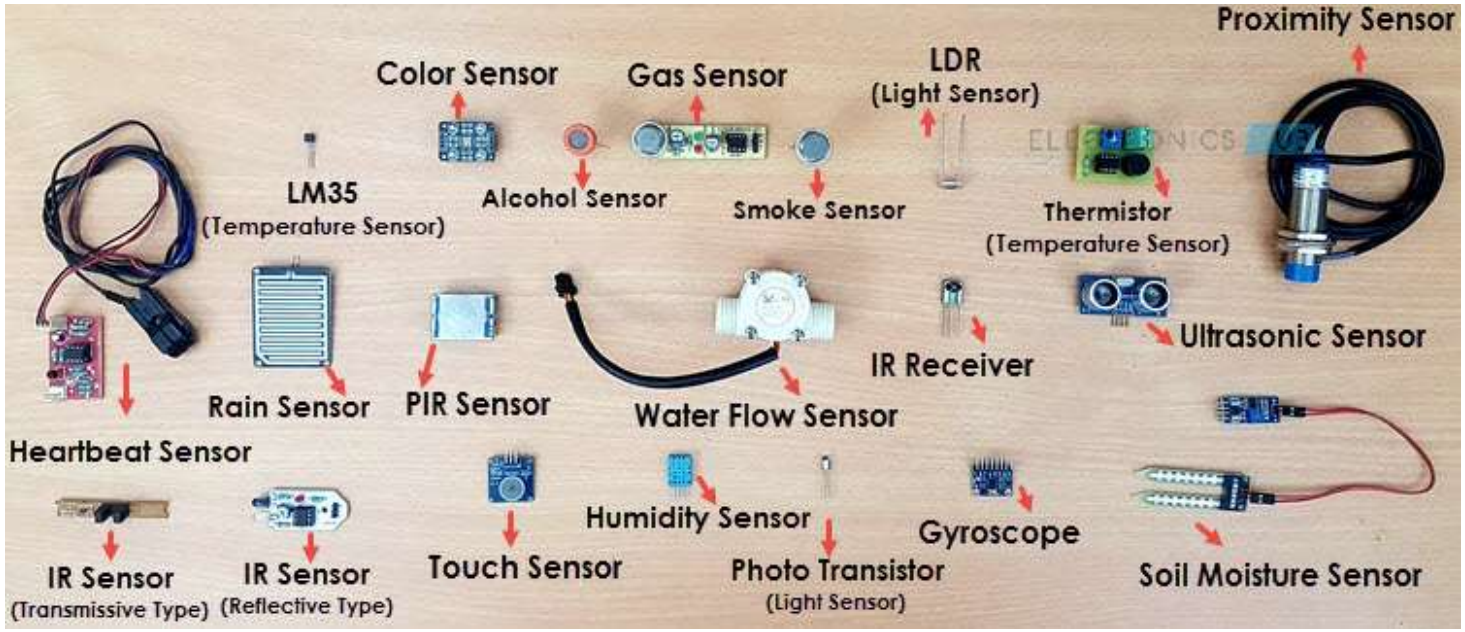
- A more sophisticated one: PM 2.5-10 sensor
  - E.g., for smoke, dust, pollen
  - Works by detecting with a photodiode the light scattered by particles suspended in air in a chamber
  - A circuit generates a series of pulses that are directly related to the light scattered by the particles
  - Needs a lot of processing!



TIDA-00378 PM2.5 and PM10



And many others…



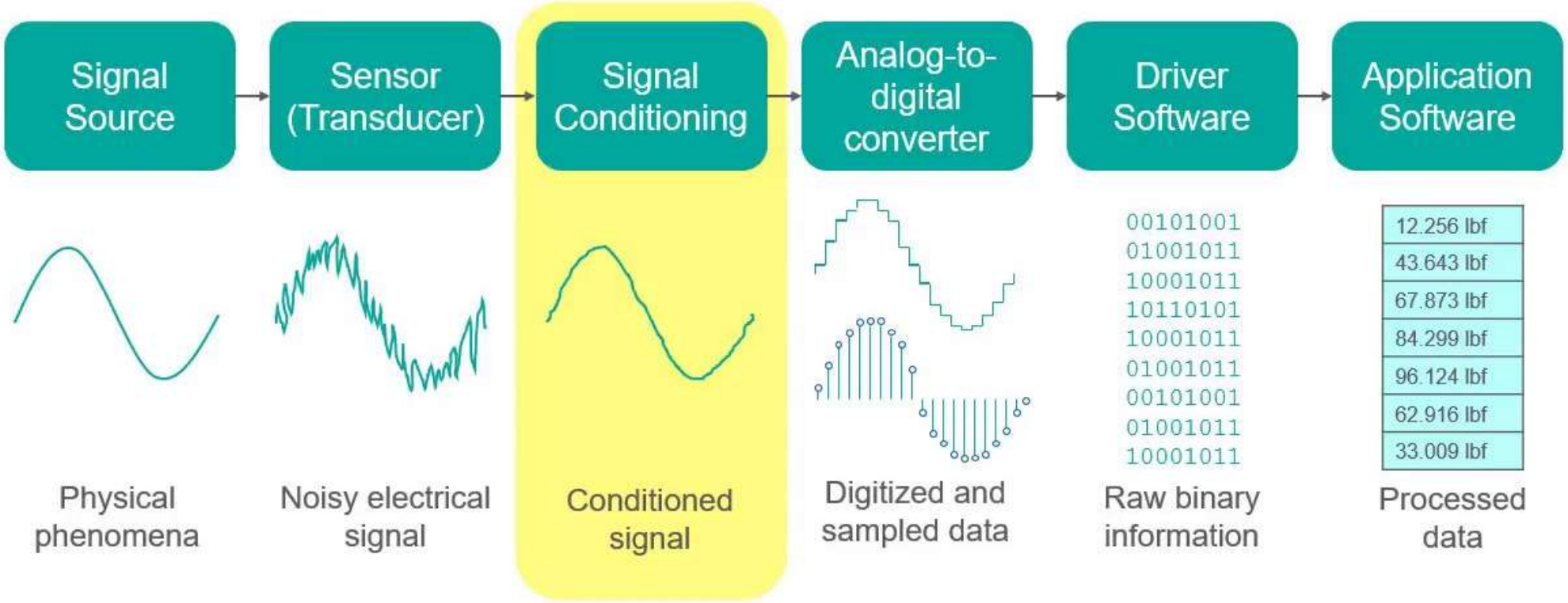
Concept generalization

- Key element is the transducer
- Transducer = element that converts energy from one domain to another

  - Typically we are interested into transducers that have electrical domain as output

- A sensor (as well as an actuator) includes a transducer
  - Sometimes the same actuator con serve in both directions (e.g., an antenna)

From the physical world to a digital domain

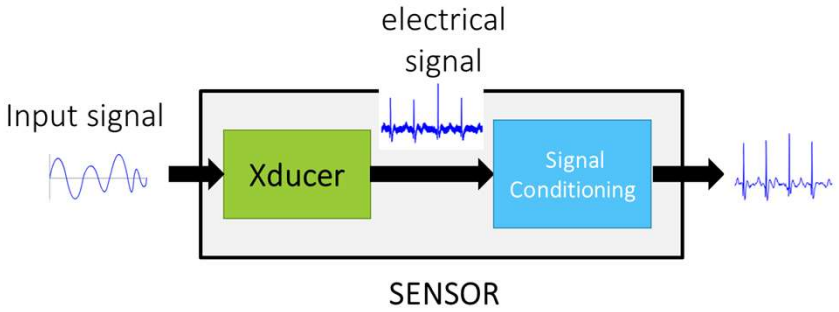


Signal conditioning

- Adjust the dynamic of signals generated by the transducer in order to match the desired electrical levels

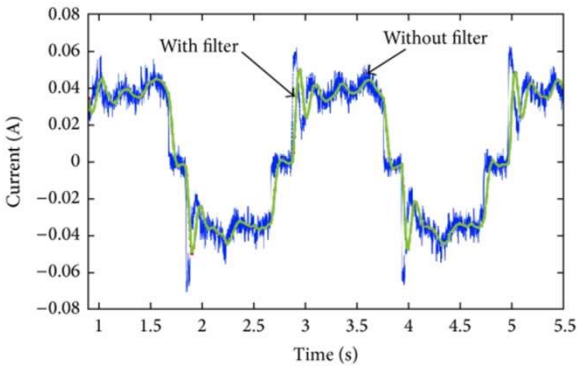
- Still in the analog domain
  - Analog circuits

- Part of any off-the-shelf sensor
  - Active/Passive
  - Digital/analog

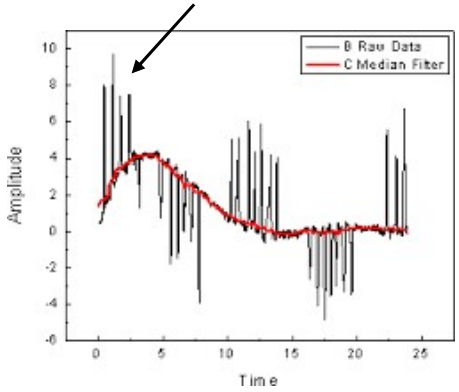


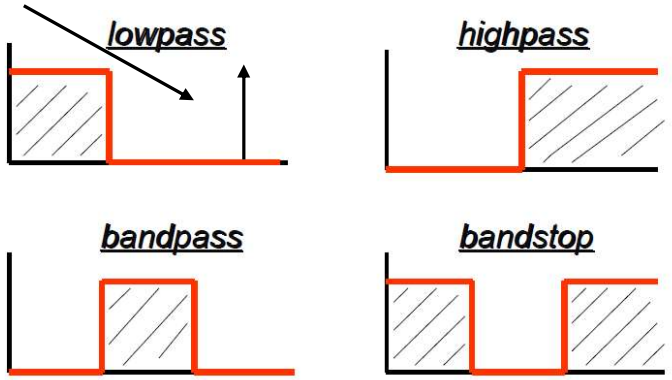
Signal conditioning

- Filtering:
  - the most common signal conditioning function, as usually not all the signal frequency spectrum contains valid data.
  - For noise suppression and/or frequency selection (need some specific band of the signal only)



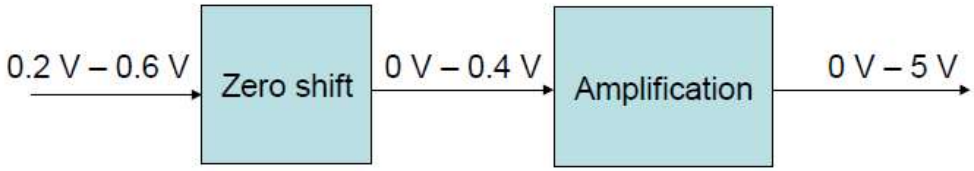
High-frequency spikes suppressed





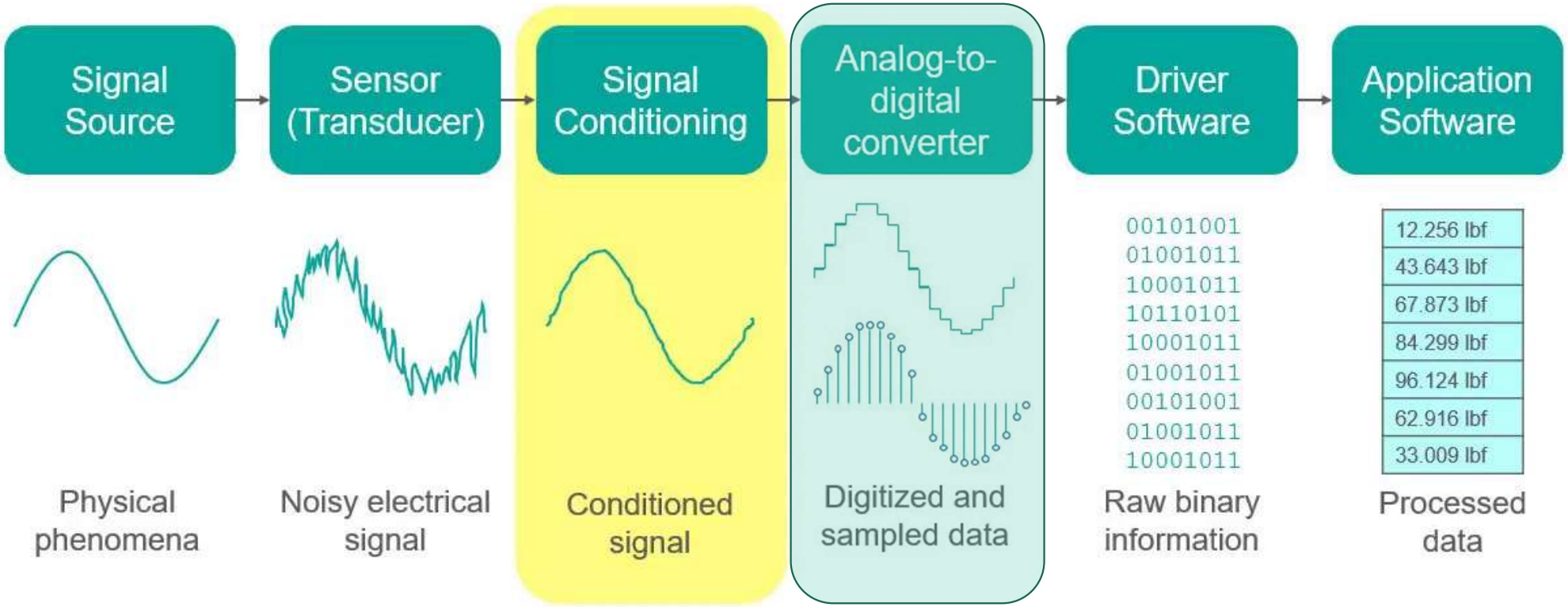
Signal conditioning

- Level shift
  - adjust the bias (zero value)
  - Better exploit the dynamic of ADCs



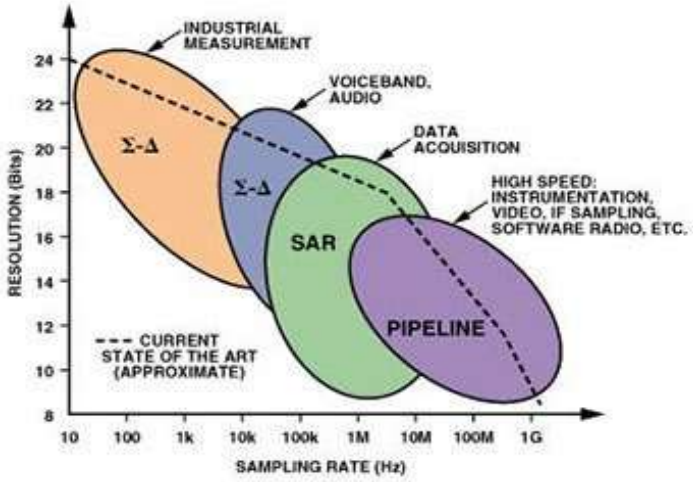
- Amplification:
  - Signal amplification performs two important functions:
    - increases the resolution of the input signal
    - and increases its signal-to-noise ratio
  - Signal levels are usually in the μVs to mVs range!
    - For example, the output of an electronic temperature sensor, which is probably in the millivolts range is probably too low for an analog-to-digital converter (ADC) to process directly. In this case it is necessary to bring the voltage level up to that required by the ADC. [wiki]

Converting to digital



Converting to digital

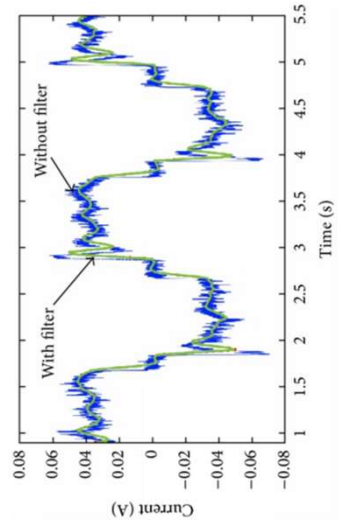
- Regardless of the sensor type the conversion into the digital domain is mandatory
  - Processing IS digital!!!
- A/D conversion implements sampling + quantization
- Key parameter is the resolution, i.e., # of bits of the quantization
- Various circuit implementations!

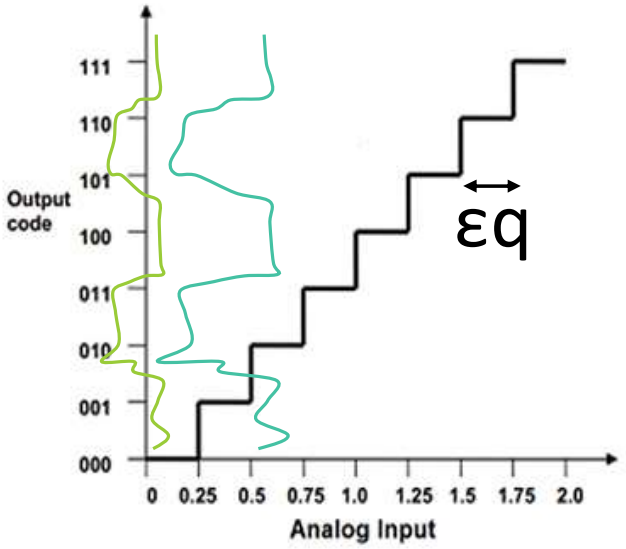


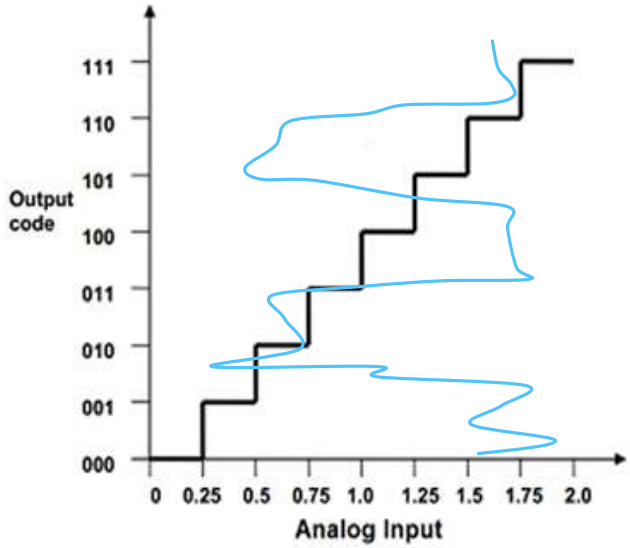
Source: Analog Devices

ADC operations

- Conceptually, and ADC implements a step-wise transfer function
  - Example: 3-bit conversion
- cover the whole input dynamic = minimize the quantization error
  - Conditioning is key!







ADC Example

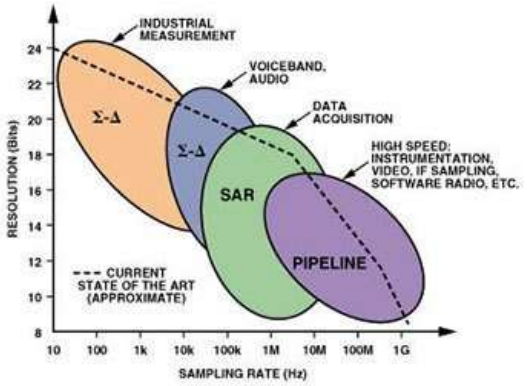
- 10-bit ADC
  - Input voltage x [0-5V]
  - 0V is 0, 5V is 1023, i.e. 2^{10}-1
  - 1023:5 = Ybin : x
- An analog voltage *x* [Volts] will be reported by the ADC as

$$Ybin = ceil(x * 1023/5)$$

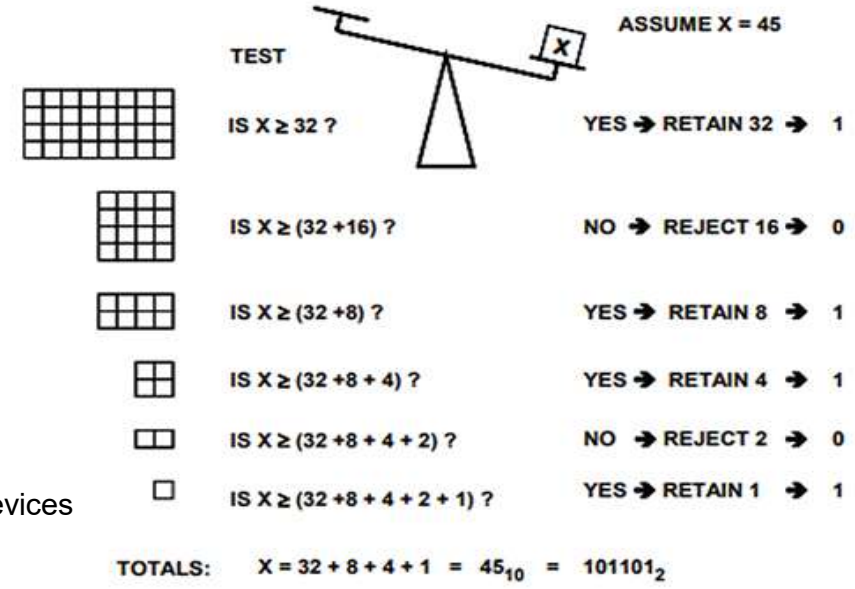
- Example: input V=2.12V
  - Ybin=ceil(433.752) → reading is 434
  - ε = 0.00121 V (0.06%)

ADC operations

- Many implementations do exist
- Successive Approximation Registers (SAR) ADCs
  - Basic principle of operations (iterative!):

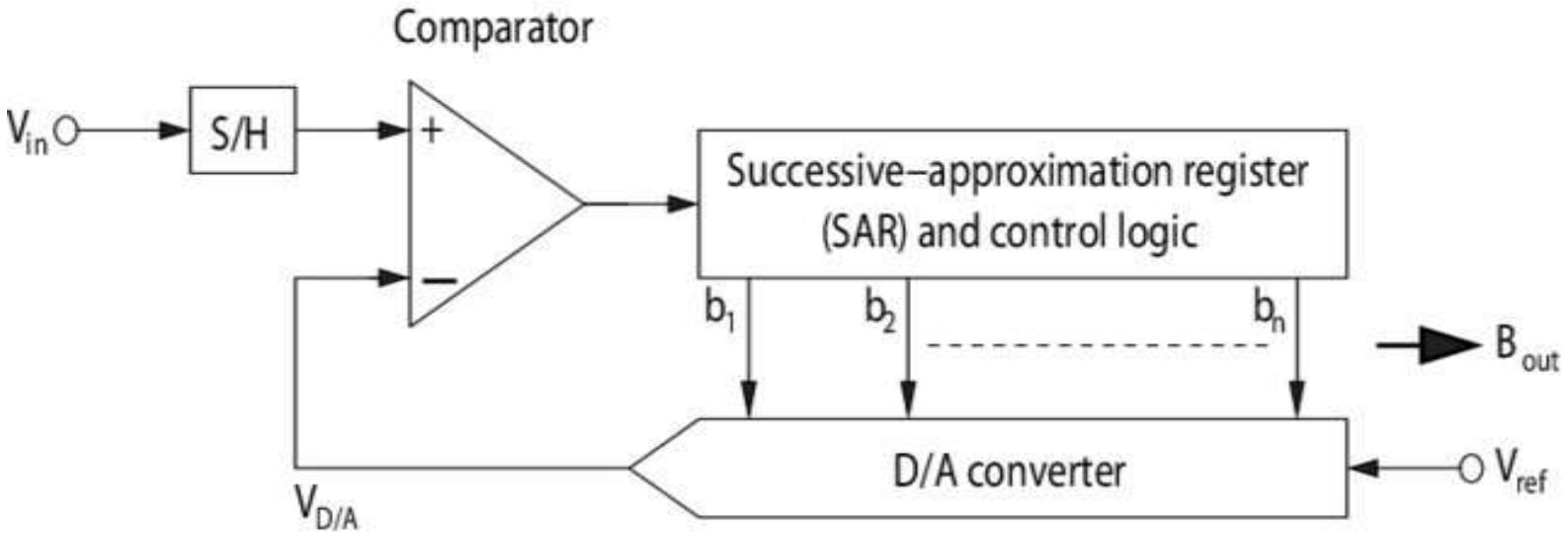


Source: Analog Devices

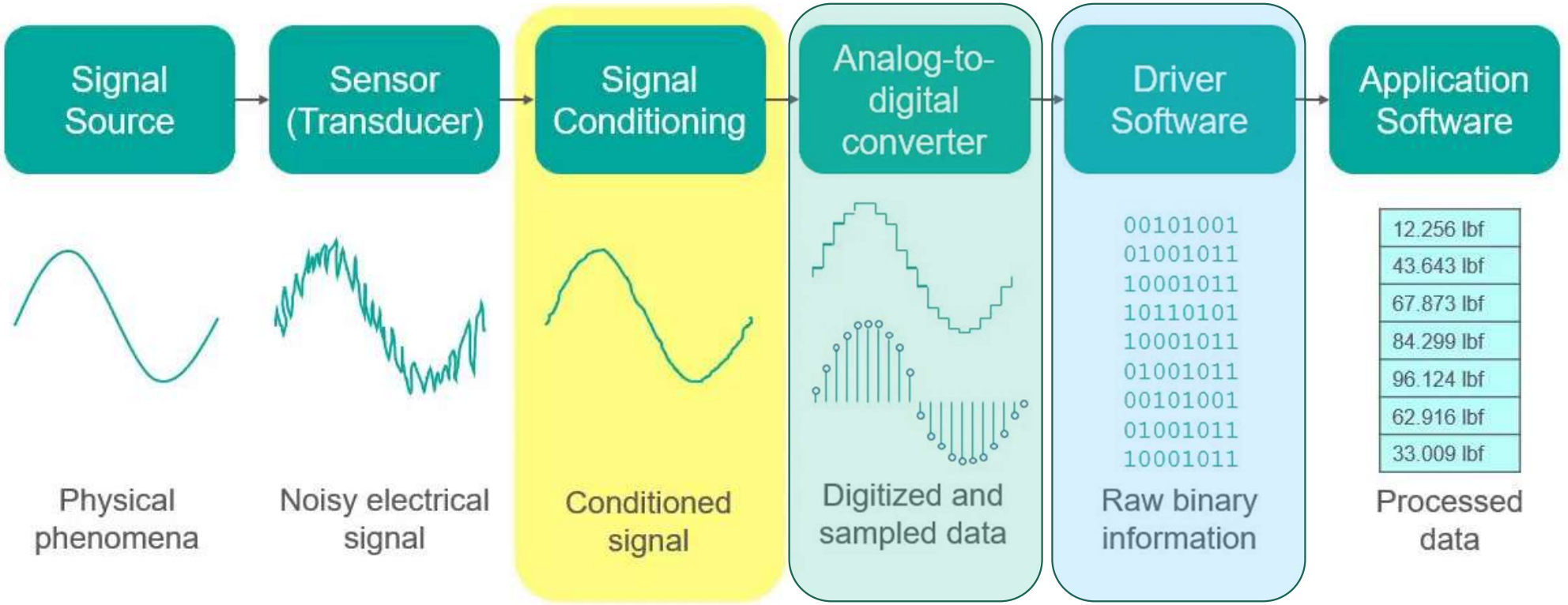


Source: Analog Devices

(very) Basic SAR architecture

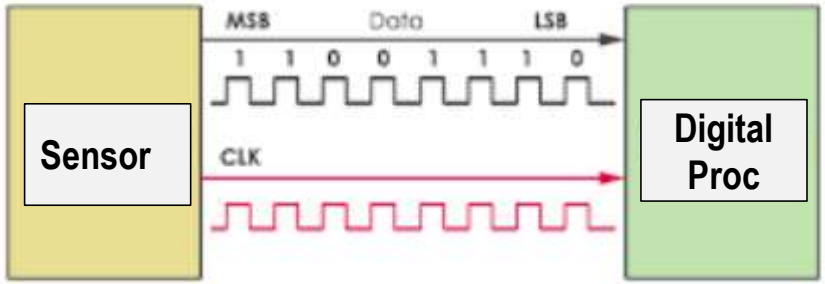


Readout


Signal Source
Sensor (Transducer)
Signal Conditioning
Analog-to-digital converter
Driver Software
Application Software
00101001
01001011
10001011
10110101
10001011
01001011
00101001
01001011
10001011
12.256 lbf
43.643 lbf
67.873 lbf
84.299 lbf
96.124 lbf
62.916 lbf
33.009 lbf
Physical phenomena
Noisy electrical signal
Conditioned signal
Digitized and sampled data
Raw binary information
Processed data

Reading

- Sensors (i.e., ADCs) typically outputs bits **serially** (one at a time):
  - Fewer pins
  - Better signal robustness (no interference)
  - Faster speed at iso-noise



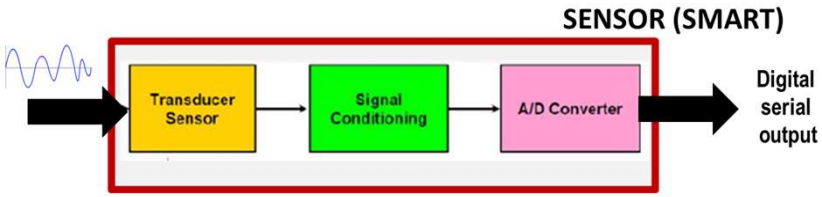
- In the digital word most interfaces are serial!
  - Most popular: USB, SPI, I2C
- This process is transparent to the programmer
  - Drivers manage the transfer protocol
  - A simple read function provides the full word

Serial Interface

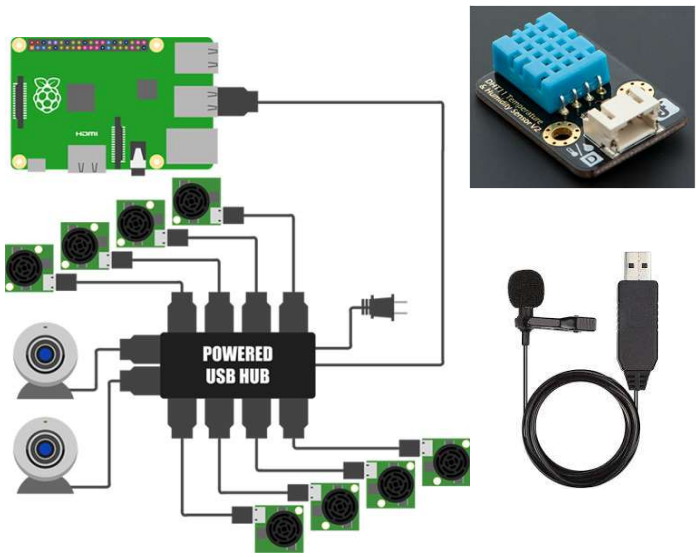
- USB
  - Vast majority of modern devices (less popular for sensors)
  - A (host/master) B (peripheral/slave)
    - OTG: "On The Go": Accepts A (make me a host) or B (make me a peripheral)
  - Often used as power source
- I2C (Inter-Integrated Circuit)
  - Allows several "slave" (sensor) devices to be accessed by same "master“
  - Data (SDA) and clock (SCL) lines (2 wires)
- SPI (Serial Peripheral Interface)
  - Multiple slaves, bidirectional
  - 4 wires (Clock, select, MISO/MOSI)

Practical considerations

- How can a non-EE engineer truly use sensors?
  - Too much technicalities, interfaces, etc.
- Commercial (smart) sensors for system designers include all the components and hide all the details to the designer
  - also preventing possible optimizations…
- Smartness often include
  - Full analog-to-digital pipeline
  - Programmability (signals to set some parameters)
  - Storage (e.g. for config parameters or buffering)

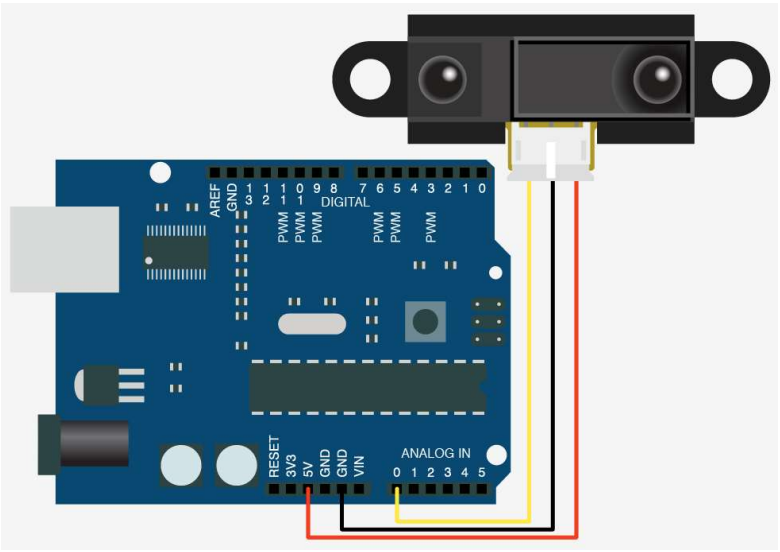



```
while 1:
    x = Serial.read()
    print x
```



Practical considerations

- Simpler sensors may demand the ADC conversion to “someone else”
  - done on-board and not in-sensor



Coordinating multiple sensors

- Devices seldom have only one type of sensor
- Multiple sensors can be used to sense independent quantities for
  - uncorrelated usage/multiple functions
    - E.g., a camera and microphone serve different purposes...
  - correlated usage
    - To improve the quality of the sensing process
      - E.g., gyroscope and accelerometer to improve an estimate of motion
    - Called sensor fusion
- Sensor fusion: combining sensory data so that the resulting information has less uncertainty than would be possible when these sources were used individually
  - Information integration problem

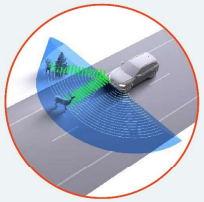
Why sensor fusion?

- Accuracy & reliability

- Fused data from multiple sensors provides several advantages over data from a single sensor:
  - Statistical advantage:
    - adding *N independent observations the estimate of the target measure is improved by a factor proportional to* $N^{1/2}$ *, assuming the data are combined in an optimal manner*
    - NOTE: The same result could also be obtained by combining N observations from an individual sensor
  - Better observability:
    - Unlike the case of a single sensor, where some quantities cannot be accurately estimated

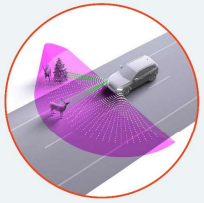
Self-driving

RADAR



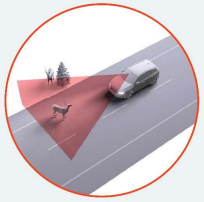
- Long-range sensing
- Object movement
- All-weather performance

LIDAR



- Precise 3D object detection
- Range accuracy
- Free-space detection

CAMERA



- Object classification
- Object angular position
- Scene context

	RADAR	LIDAR	CAMERA	FUSION
Object detection	+	+	○	+
Pedestrian detection	—	○	+	+
Weather conditions	+	○	—	+
Lighting conditions	+	+	—	+
Dirt	+	○	—	+
Velocity	+	○	○	+
Distance - accuracy	+	+	○	+
Distance - range	+	○	○	+
Data density	—	○	+	+
Classification	—	○	+	+
Packaging	+	—	○	+

+ = Strength O = Capability — = Weakness

Sensor Fusion

- There are three fundamental paradigms

  A. Measurement fusion: Direct fusion of sensor data

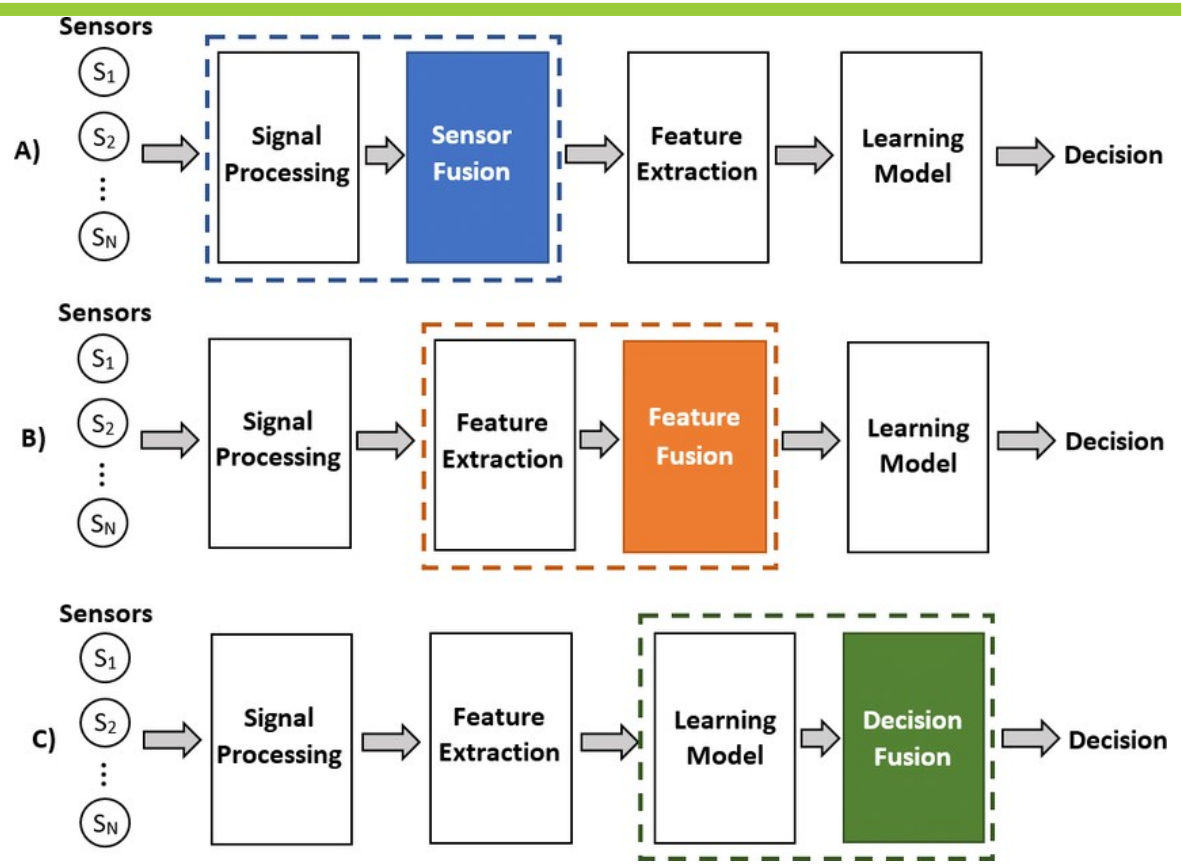
  B. Feature fusion: Representation of sensor data via feature vectors and fusion of the feature vectors

  C. Decision fusion: Processing of each sensor to achieve high-level inferences and fusion of these inferences

Sensor Fusion

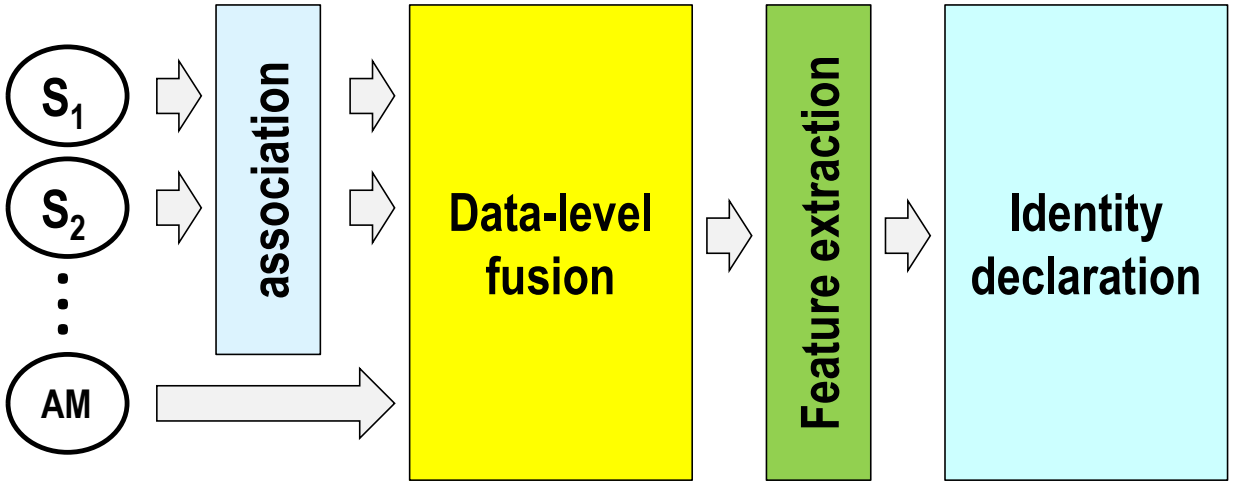
DATA

INFORMATION



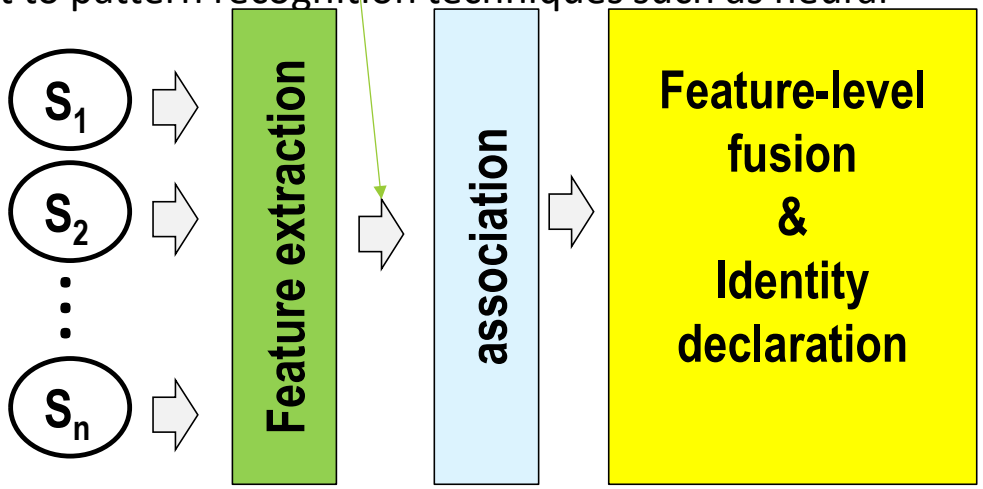
A. Measurement fusion

- Suitable when data are commensurate
  - sensors are observing the same physical phenomena, eventually measuring different physical quantities
- Classical estimation models (control theory)
  - Kalman filter



B. Feature fusion

- For non-commensurate sensor data, data **must** be fused at the **feature** level
  - Features are combined into a single concatenated feature vector
- Feature: a sort of signature/digest representative of the sensed quantity
  - Works as a common ground
  - features are extracted from multiple sensor observations and combined into a single concatenated feature vector that is input to pattern recognition techniques such as neural networks, clustering algorithms

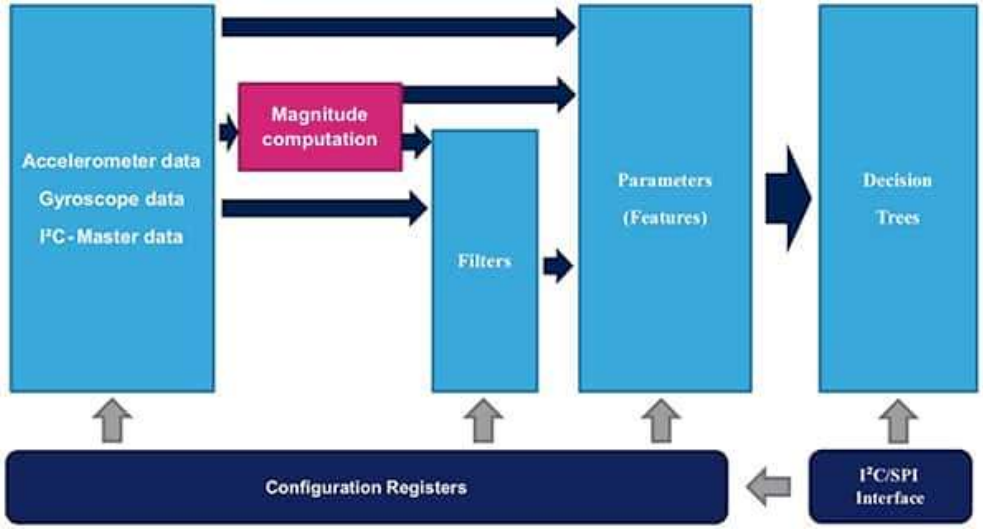


More about features

- Features are strongly sensor- and application-dependent!
- Since they must be combined, they are often normalized quantities
- Some examples:
  - Sound features
    - Pitch, Centroid, Spectral variance, Zero-crossing rate (ZCR). Root Mean Square (RMS) (Volume), Signal-to-noise ratio (SNR), …
    - Suitable also for generic signals
  - Visual (image) features
    - Edges, corners, ridges, color, hue, blobs, Spectral properties, special filters, etc

A practical example: Smart Motion from STM

- Motion sensor



- the device generates a large number of features calculated from primary data within a sliding time window including:
  - mean
  - variance
  - energy
  - peak to peak
  - zero crossing
  - positive zero crossing
  - negative zero crossing
  - peak detector
  - positive peak detector
  - negative peak detector
  - minimum
  - maximum

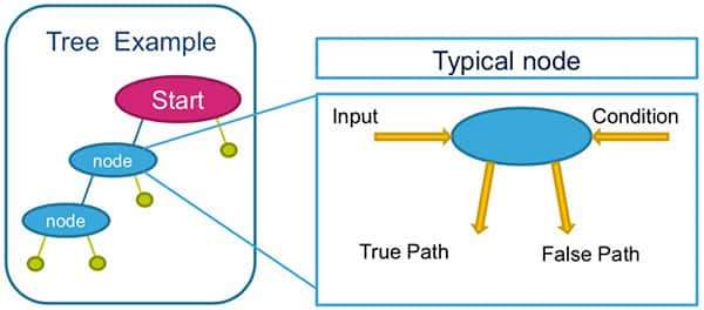
A practical example: Smart Motion from STM

- at each update cycle, the decision tree would be invoked to work through its nodes to determine if the data available – with that update – represented no movement, forward movement, or some other movement as follows:

1. test the magnitude of an accelerometer measurement
    1.1. terminate if the value is below some predetermined value (the condition)
    1.2. otherwise, branch to a child node to test gyroscope measurements taken in the same time window
        1.2.1. terminate if gyroscope measurements are below some predetermined value or
        1.2.2. continue to a deeper child node to test other attributes measured in the same time window or test the same attribute against another condition.

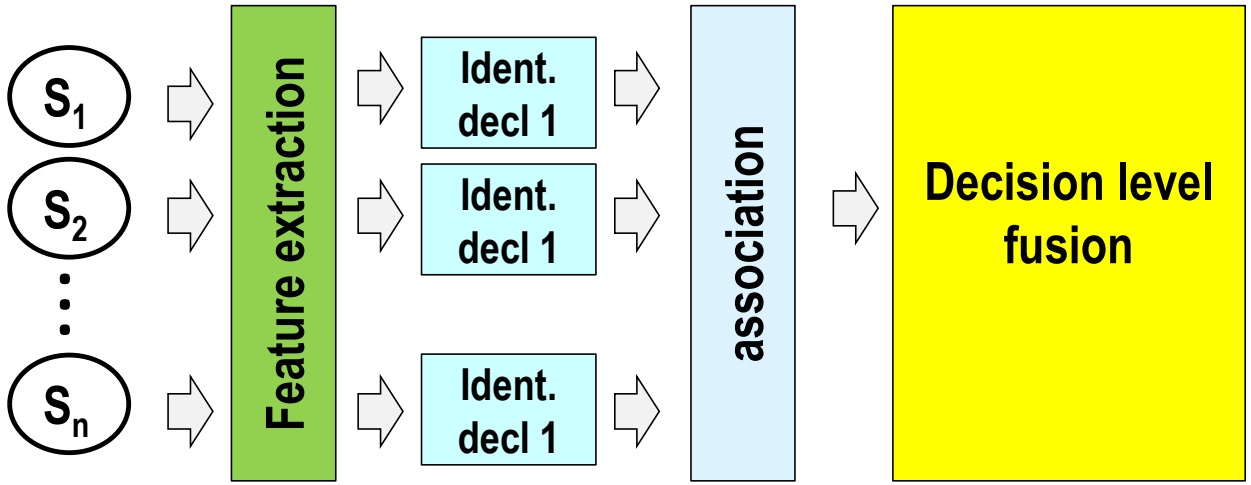
- This process repeats until the test reaches a terminal node, which in this context corresponds to a particular complex motion event, or class. In this simple example:
    - terminal node 1.1 might indicate that the data, or feature set, should be classified as “no movement”
    - terminal node 1.2.1 might indicate that the feature set should be classified as “forward movement”
    - terminal nodes below node 1.2.2 might be indicative of a moving turn or more complex change in orientation

Feature fusion using decision tree

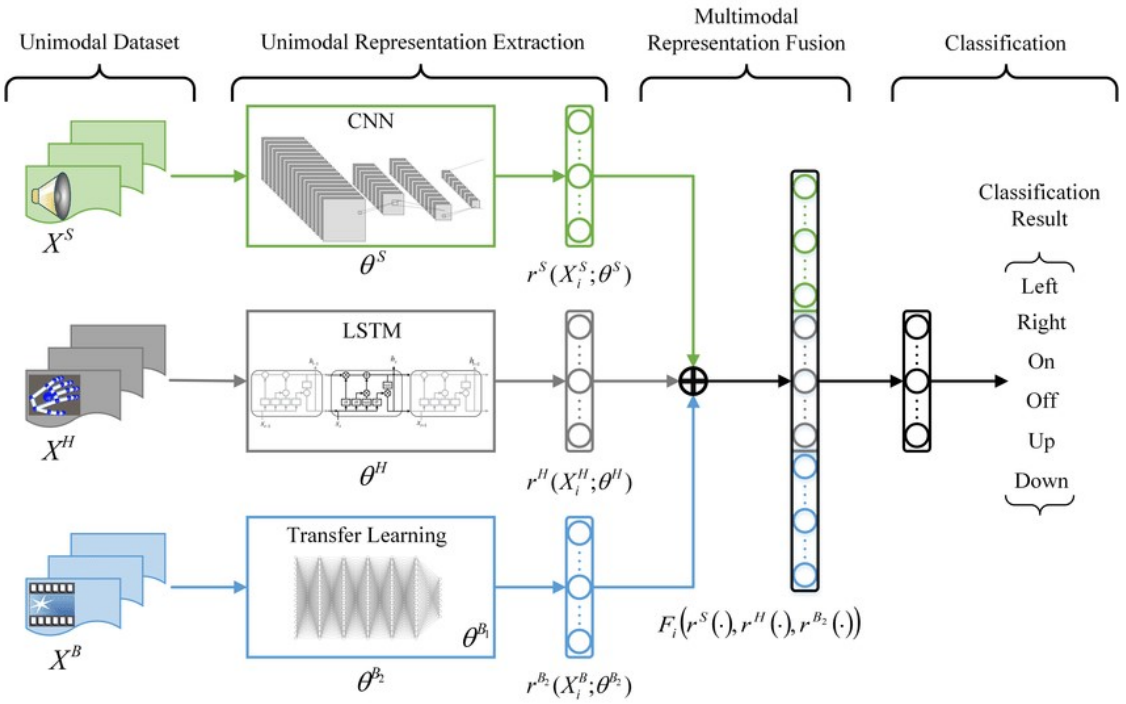


C. Decision fusion

- Each sensor has made a preliminary determination of an entity's location, attributes and identity before combining
- Decision-level fusion algorithms are used, such as
  - Voting/weighted decision or more complex algo (Bayesian inference and Dempster-Shafer's method)



A practical example



Other considerations

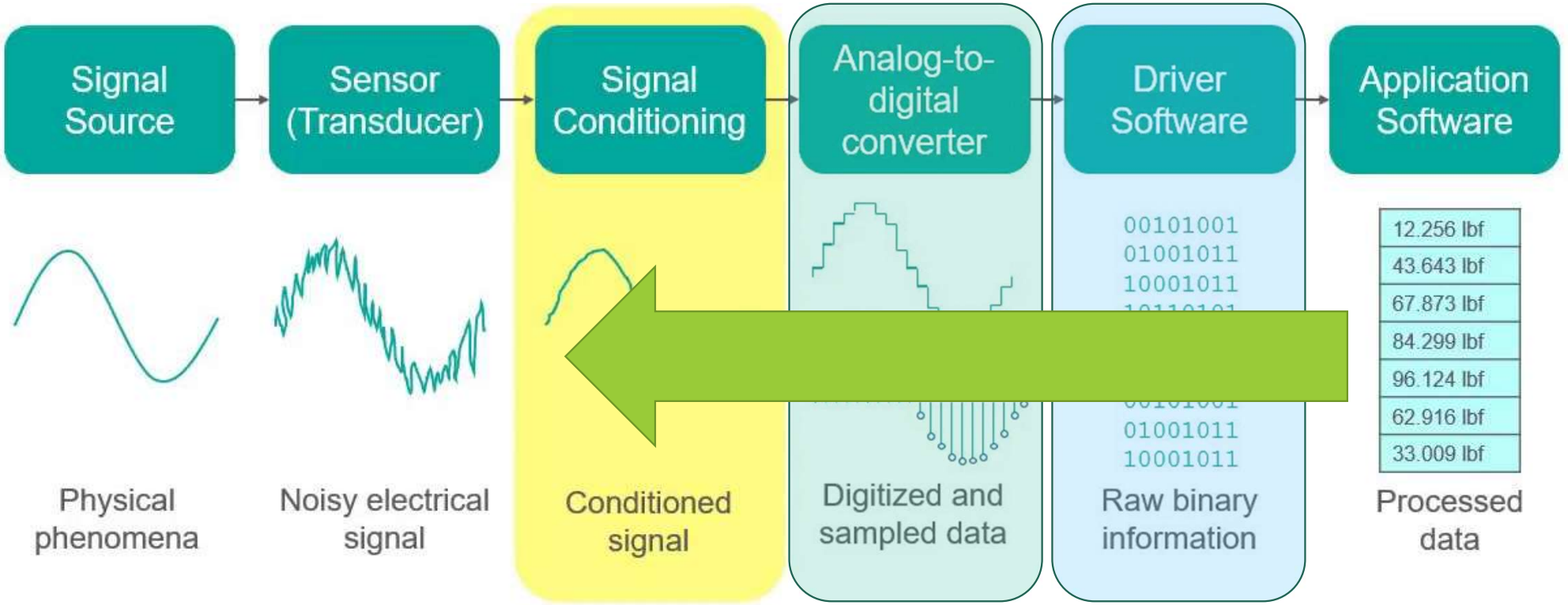
- Sensor fusion is a computational issue
  - Consider it as a part of the “software” part of an IoT device
  - Future smart sensors will integrate it as an embedded feature
- It requires significant computation complexity
  - Feature extraction and in particular classification are complex algorithms!
  - Tradeoff between quality of final classification and complexity of the method
- In-sensor computing will soon become a cutting-edge technology

Actuators

Actuators: back to the analog world

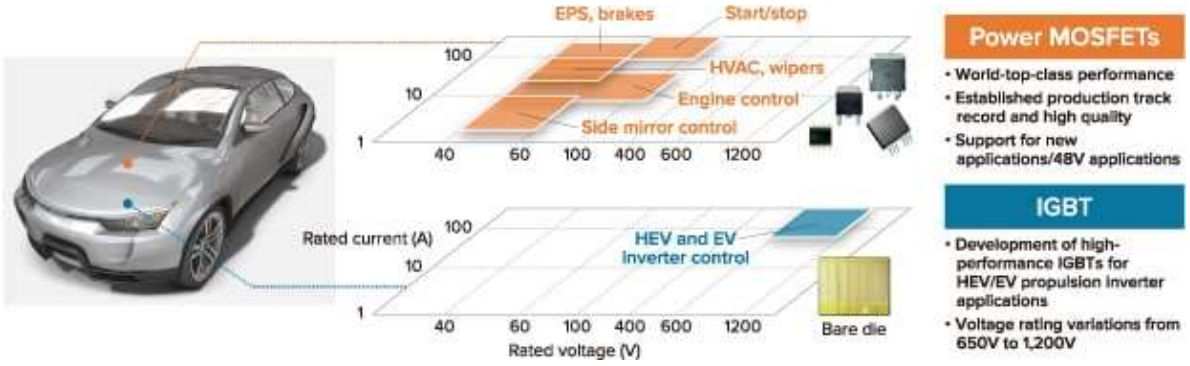
- Actuator = devices that alter a physical quantity
- The symmetric action of sensing
  - DAC + transducer
  - Transducer : electrical domain is in input…
- As for sensors, a number of possible devices
  - Electric motors
  - Comb drives
  - Piezoelectric actuator
  - Servomechanisms
  - Pneumatic actuators
  - Hydraulic actuators
  - Solenoids
  - Stepper motor
  - Speakers?

Backward flow



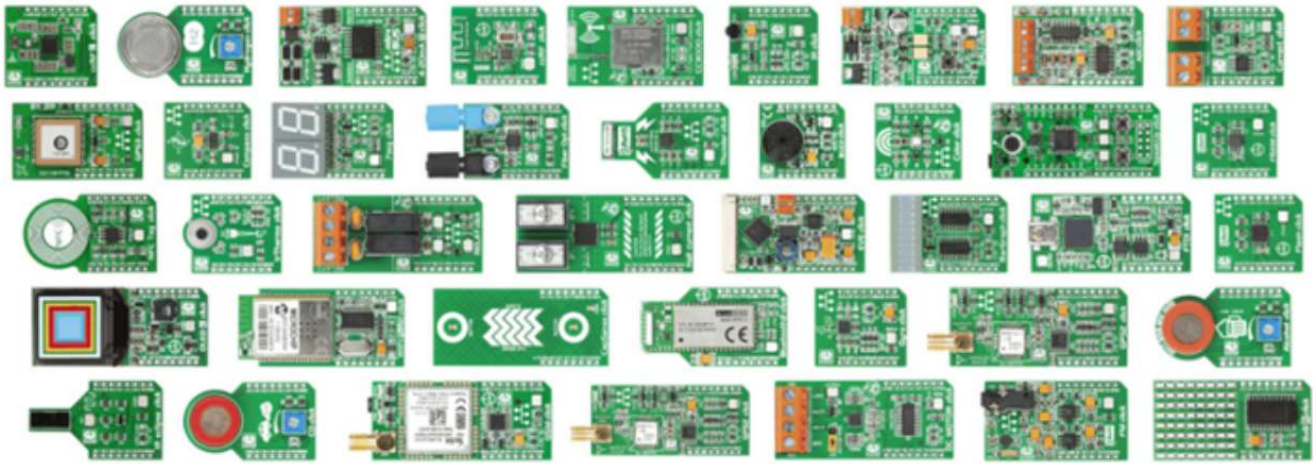
Actuators need power!

- Mechanical parts (actuators in general) need more power to be “driven”
  - Digital cores work in the range of hundreds of mV and uA (at best)
- Example: driving a speaker
  - Speaker essentially a resistor (typical 8Ω)
  - Power rating e.g. 1W ➡ current $P=RI^2$ ➡ $I = \sqrt{P/R} = 350mA!!!$
  - Assuming MCU has analog output, output current in the range of tens of mA!
    - Need an amplifier (sensor conditioning)…
- Think about electrical motors
  - Power rating KW
  - Need high-power electronics



Custom interfaces

- Apart from very common cases, actuators need custom circuitry for the interface
- Some integrated solutions do exist…
  - Rely on a common, standardized system architecture that allows seamless plug&play of sensors
  - Example: **MikroBUS™ Adapter click boards™**

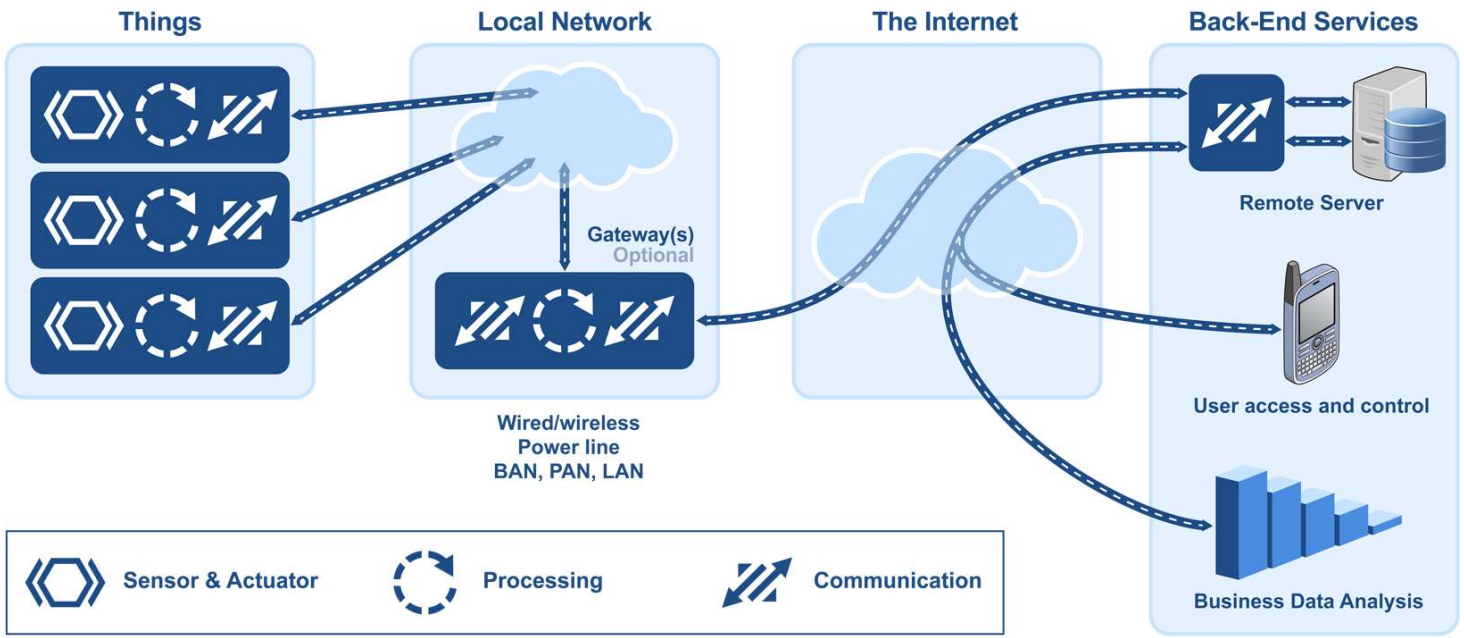


Processing

Different kind of “processing”

MCU: (Micro-)controllers

MPU: (micro-)processors



MCU vs MPU

- MPUs contain only a CPU (possibly with some cache) and interfaces
  - They require added peripherals to perform tasks.
- MPUs are quite **powerful**
  - Computational power (multiple execution units (multicore))
  - Support for special operations
  - Accelerators
- MPUs are '**modular**'
  - Resources (memory, disk, etc) are external,
  - Can be added as needed
- Expensive!!!

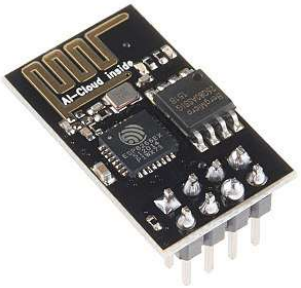
Intel Core i7-7820X

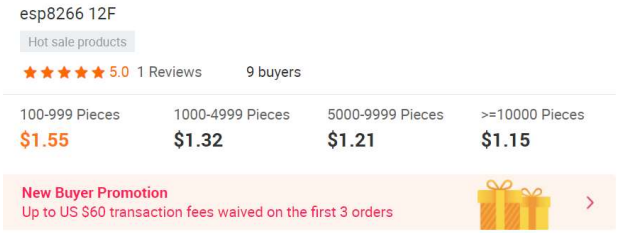
8 offerte:	**€ 475,10 – € 641,46** *
0 opinioni:	Lascia un'opinione!
Opinione degli esperti:	★★★★☆
3 Recensioni:	**Voto medio 8/10**

CPU 8 Core · Socket Socket 2066 · Numero di Threads 16 · Frequenza di clock 3,6 GHz · Frequenza turbo massima 4,5 GHz ·

MCU vs MPU

- MCUs are “complete systems” on a single board or chip
  - They contain RAM, ROM, and similar peripherals
  - A number of interfaces typical of sensors or other devices
  - Often include one or more ADC for analog inputs!
- MCUs have limited functionalities and modest computational power
  - things are changing…
- They are meant for application-specific systems
  - things are changing…
- Cheap!!



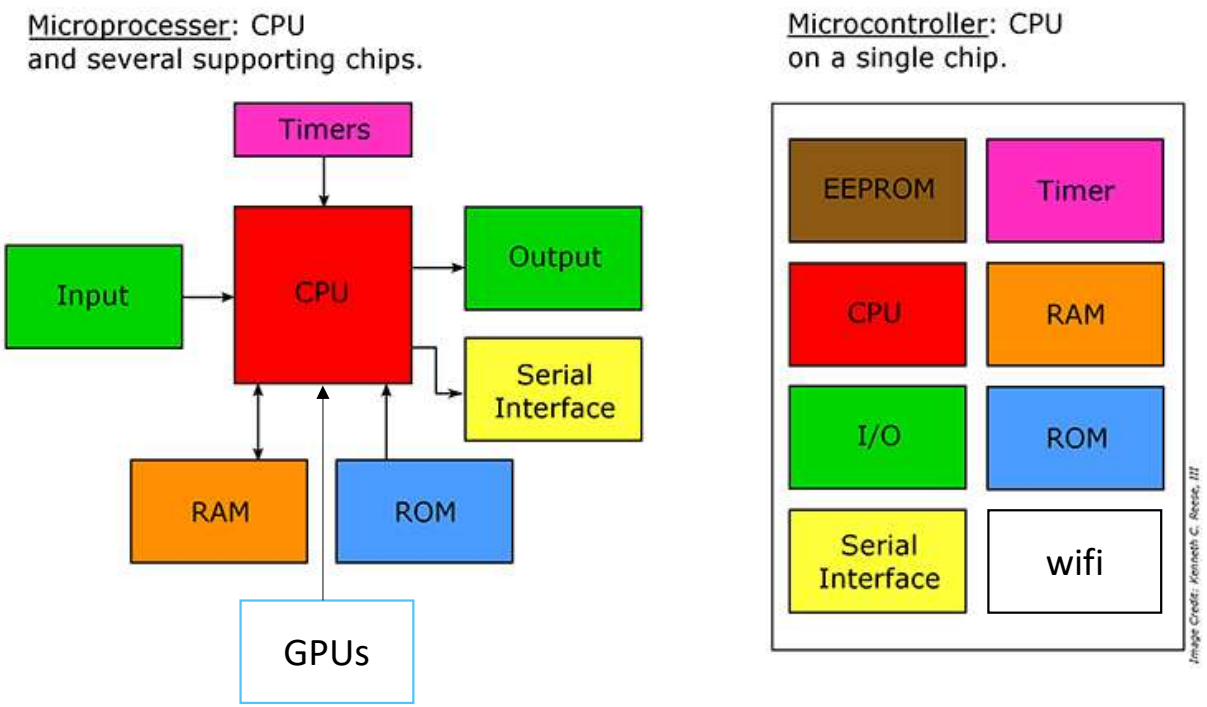


Processor: L106 32-bit RISC running at 80 MHz
Memory: 32 KiB instruction RAM, 32 KiB instruction, cache RAM, 80 KiB user-data RAM
External QSPI flash: up to 16 MiB is supported
IEEE 802.11 b/g/n Wi-Fi Integrated
16 GPIO pins
SPI: I^2C (software implementation), I^2S interfaces with DMA (sharing pins with GPIO), UART on dedicated pins, 10-bit ADC (successive approximation ADC)

Some figures

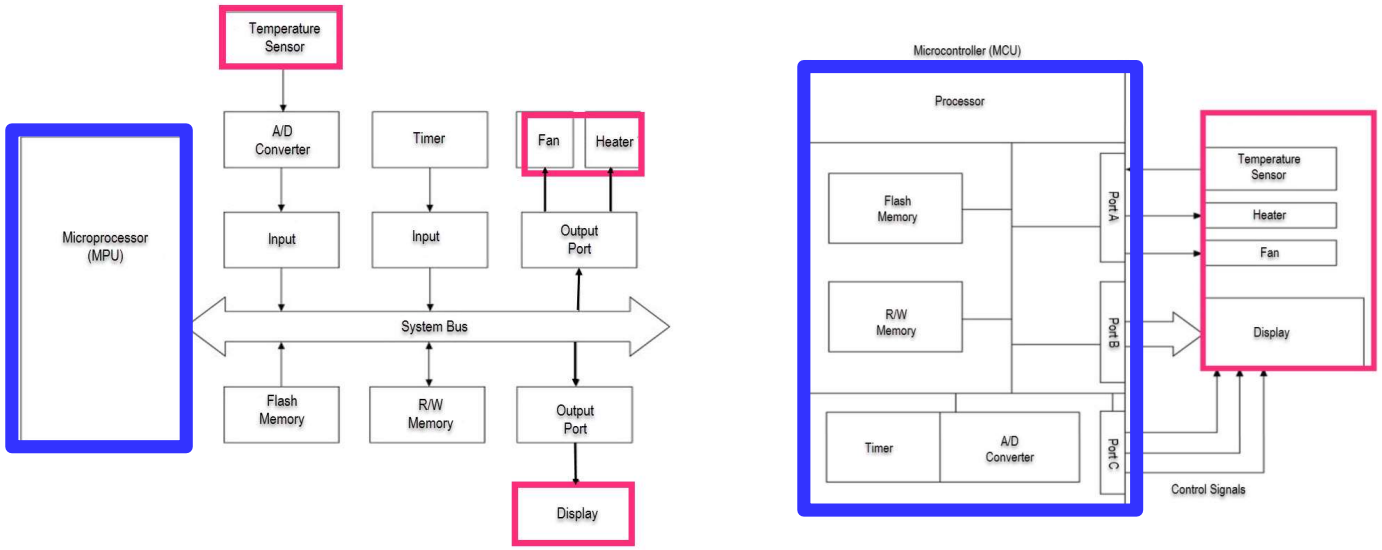
	MPU	MCU
Cost	100's $	1-10 $
Memory (RAM)	A few GBs	A few MB
Persistent Storage	Disk, TBs	Can be absent, otherwise FLASH or ROM (kB)
Clock	GHz	KHz/MHz
Cores	2-16	1
Parallelism	32-64	4-32
Speed	100-200 GOPS	1-10 MOPS
Power	100s of W	10-100 uA/MHz (a few mWs @ max f)

Conceptual Block diagram



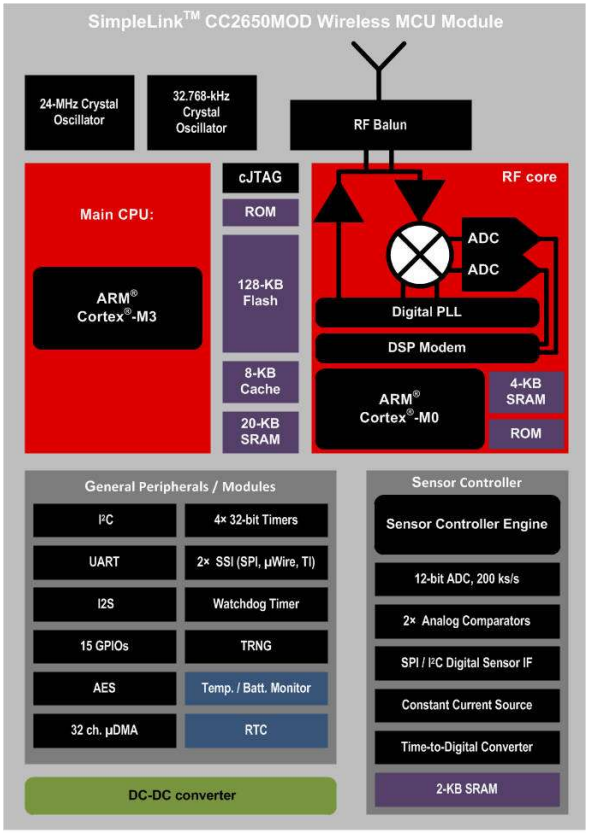
Implications on design

- A temperature control system using an MPU and MCU
  - Integration costs?



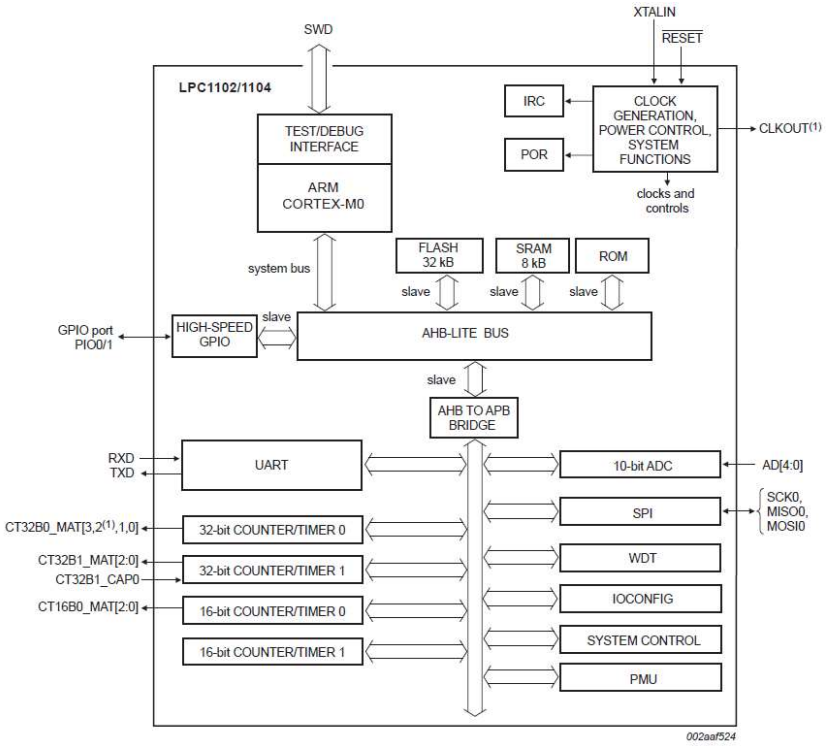
Some other commercial example

- TI's CC2650
  - "High-end" ultra low-power MCU
  - M3 core (16 bit) (<48MHz)
  - Includes wireless Xceiver
  - 6mA Rx/Tx
  - 60uA/MHz!
    - 2mA @max speed
  - 2.4-GHz RF Xceiver Compatible With BLE and IEEE 802.15.4



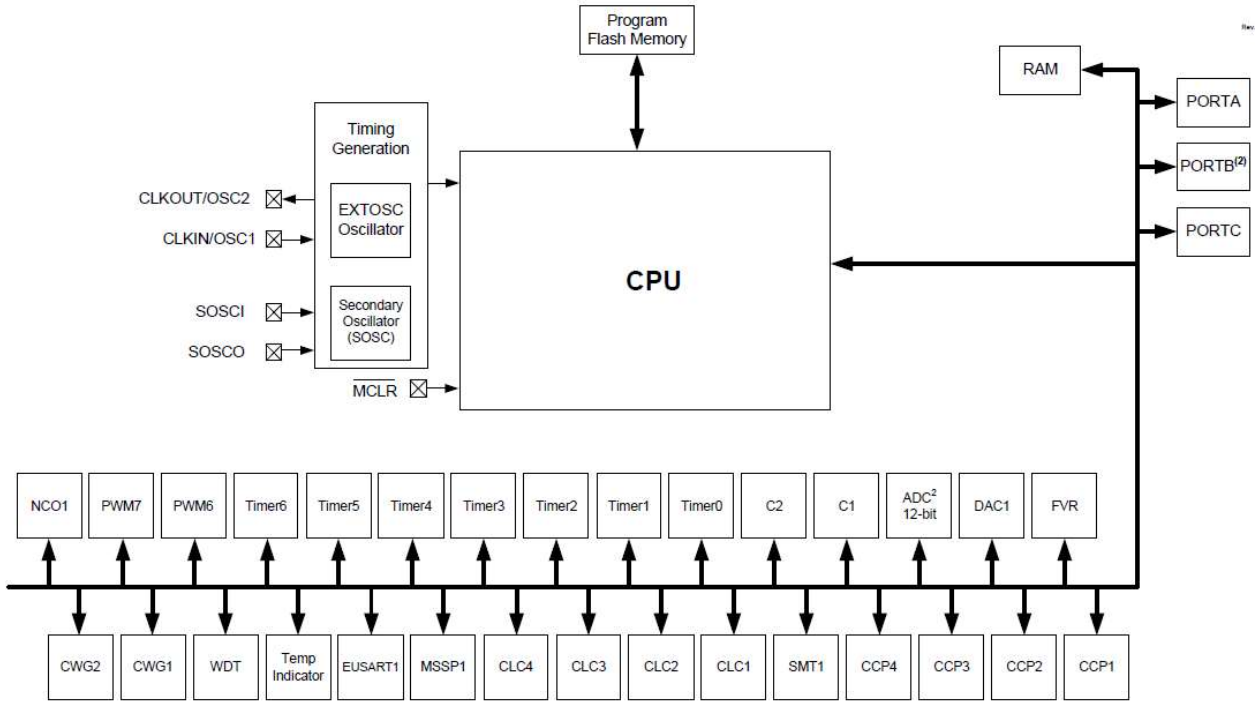
Some other commercial example

- NXP's LPC1102/1104 (32-bit)
  - M0 core
  - 12MHz-50MHz Clock
  - Vsupply = 3.3V
  - On current = 2-7mA!



Some other commercial example

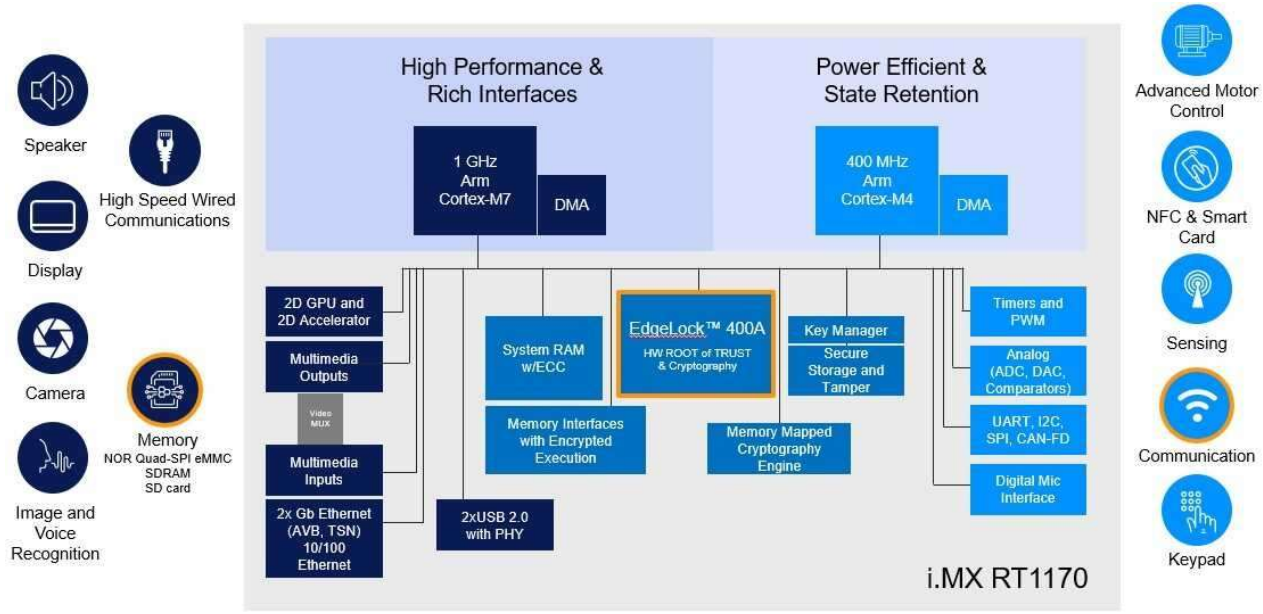
- Microchip's PIC MCU
  - 7KB RAM!
  - 16 bit
  - On-current: 2-4mA @3V



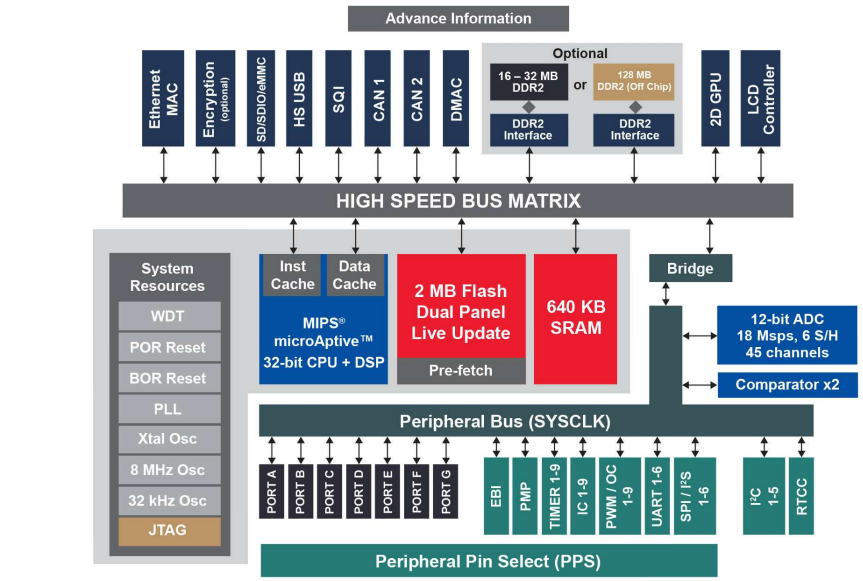
CCP = Capture, Compare & PWM
CWG= Complementary Waveform Generator
SMT = Signal Measurement Timer
CLC = Configurable Logic Cell
WDT = Watchdog Timer

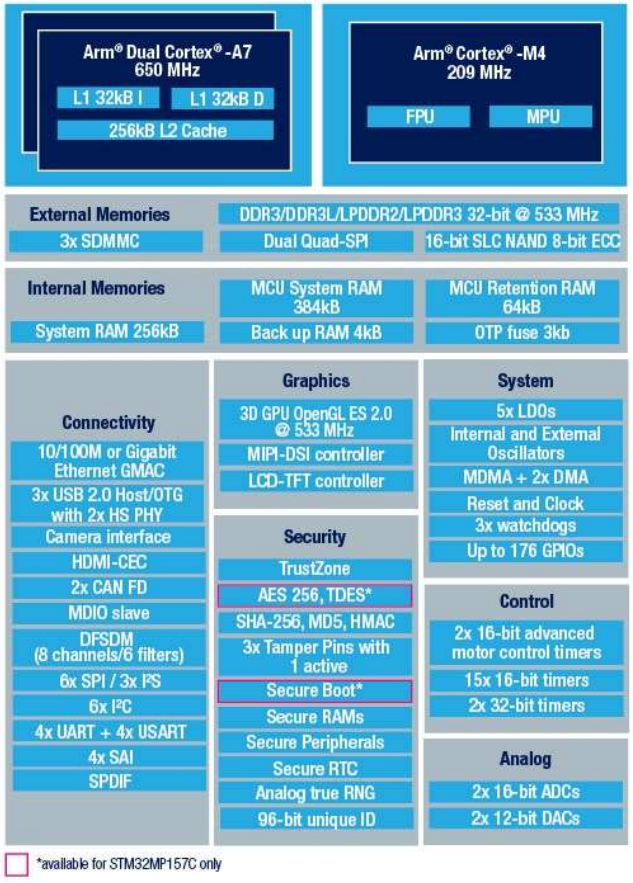
Data-analytics is changing the rules

- MCUs meet high performance
- Process data at the source



MCUs breaks GHz barrier





Other issues you must be aware

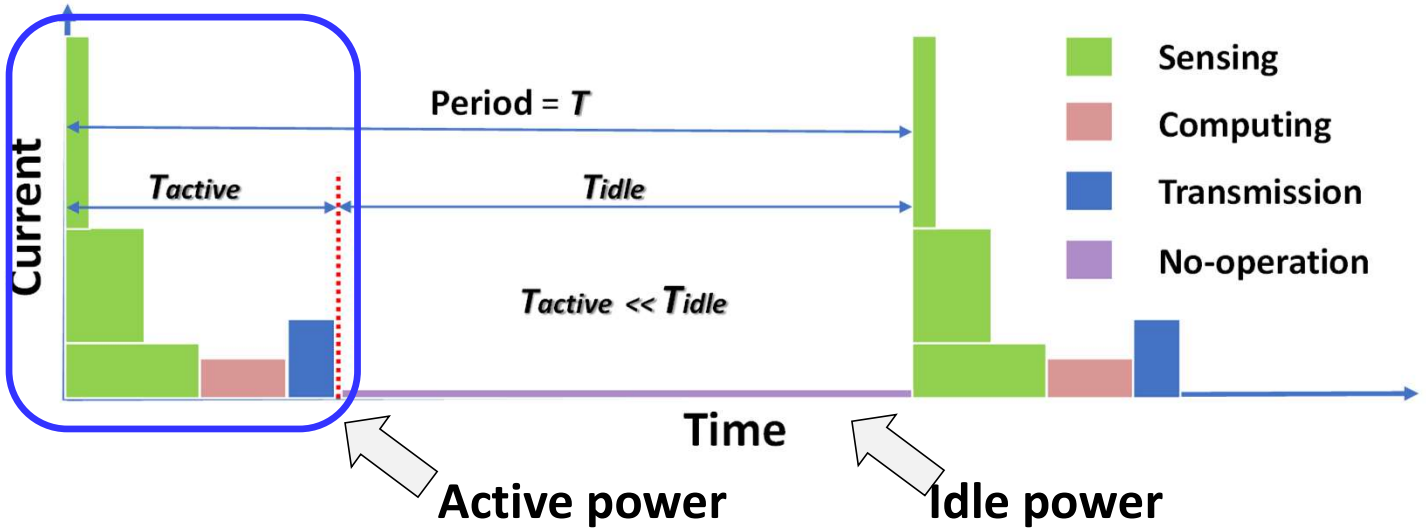
- In general, MCU vs MPU is a clear dichotomy
    - MCUs: run control algorithms
    - MPUs: run computationally intensive algorithms
    - You clearly have in mind what you need
        - MCU selection for your application not a difficult task
- To choose is getting less trivial for modern applications
    - Edge-computing: analytics-on-sensors
        - MCUs with MPU capability: heterogenous HW
    - The complexity of SW to be run is the discriminant factor
        - Not just functionality but also performance and power budget become important
            - Volume of data to process
            - kind of data-processing, e.g. deep-learning at the edge
            - Performance and real-time applications

Other issues you must be aware

- For classical (old-style) MCU-based applications, it is also important to consider
  - Programming models/support
    - Forget about fancy IDE with nice GUIs and support for virtually infinite programming languages
      - Simple IDEs are however available…
      - Need to interact with low-level details
  - SW support/IDE/toolchain and programming models and ready-to-use libraries are an important element for the choice of an MCU!
    - Reduce time-to-design
- Idle power consumption

Idle power consumption

- For MPUs active power matters
  - Active = when the MPU computes
- For MCUs idle power is at least as important
  - Idle = when the MCU is waiting for events



MCU typical workload

- MCU workloads are dominated by idle time…!
  - Small computational tasks (sensing, processing, actuating)
  - Often long duty cycles
- Duty cycle D = Tactive/T
  - No larger than 10-2, often 10-4/10-5
- MCU idle power is a fundamental parameter!!!
  - Notice that there are different types of ‘idleness’, depending on what is disabled
  - Ex: CC2650
    - Idle/standby/shutdown currents = 0.5mA/1uA/0.15uA
    - The lower, the longer the time to resume to active!
      (14 µs /151 µs / 1015 µs)

Example: STM32L4R5

life.augmented

STM32L4R5xx

Ultra-low-power Arm® Cortex®-M4 32-bit MCU
up to 2MB Flash, 640KB SRAM, LCD-TFT & M

Features

- Ultra-low-power with FlexPowerControl
  - 1.71 V to 3.6 V power supply
  - -40 °C to 85/125 °C temperature range
  - Batch acquisition mode (BAM)
  - 305 nA in VBAT mode: supply for RTC and 32x32-bit backup registers
  - 33 nA Shutdown mode (5 wakeup pins)
  - 125 nA Standby mode (5 wakeup pins)
  - 420 nA Standby mode with RTC
  - 2.8 µA Stop 2 with RTC
  - 110 µA/MHz Run mode (LDO mode)
  - 43 µA/MHz Run mode (@ 3.3 V SMPS mode)
  - 5 µs wakeup from Stop mode

LQFP144 (20 × 20) UF
LQFP100 (14 x 14) UF
U

  - Internal multi oscillator, aut ±0.25 % accu
  - Internal 48 M
  - 3 PLLs for sy
- RTC with HW ca
- Up to 24 capacit touchkey, linear
- Advanced graph
  - Chrom-ART A enhanced gra

Power and Energy: avoid confusion

- Power is an instantaneous metric
  - Energy is cumulative

    $\mathbf{E(t) = \int P(t)\, dt}$

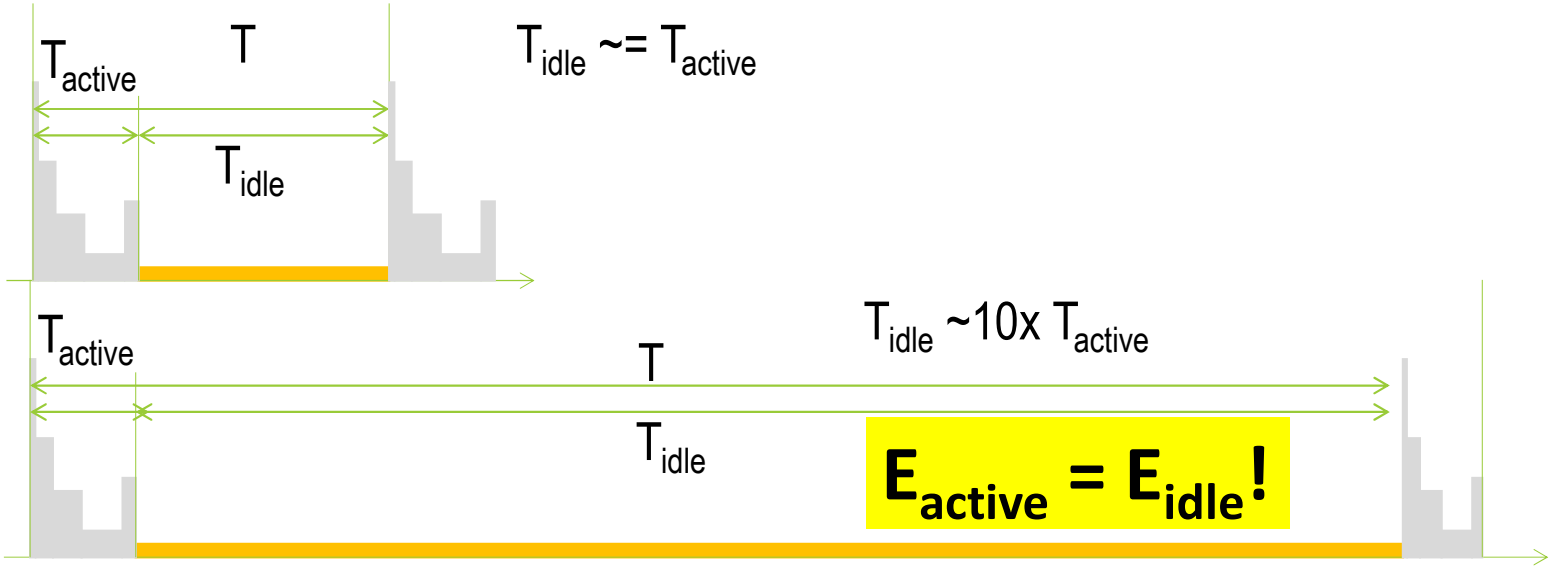
    (area of power waveform)
- Energy is what you consume from a battery!
  - Depends on power AND time

Importance of duty cycle

- Relevance of static power $\propto$ the ratio T_{idle}/T_{active}
- Example: 1/100 duty cycle (idle=99%)

P_{active} = 10mW

P_{idle}= 10μW (1/100)

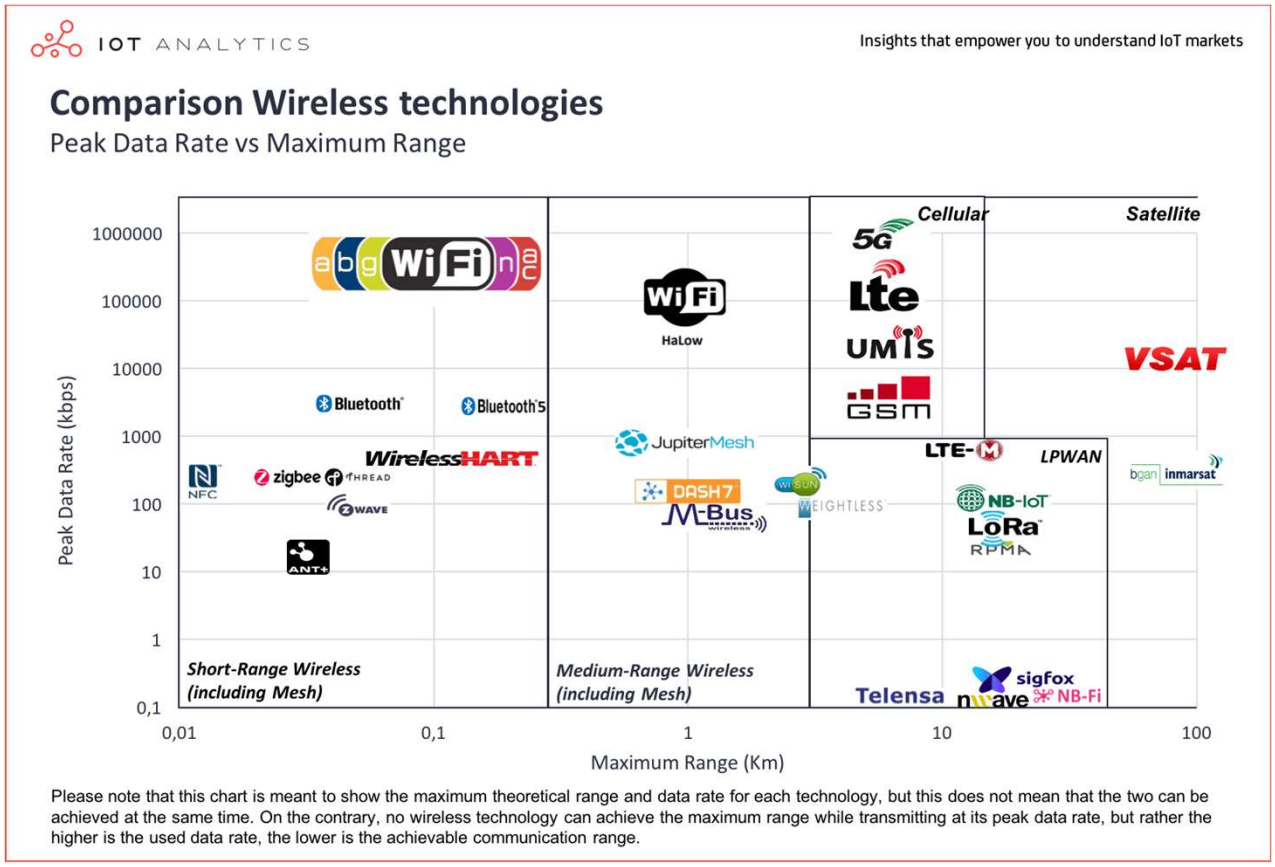


Radio communication

Wireless interfaces

- Can be seen as actuators: Antenna as a transducer
- Standardization helps!
  - Wireless interfaces allow
- Many processing cores include radios and wireless I/F
  - But we need to understand which is the wireless communication needed by our application
- FOMs
  - Bandwidth/bitrate: correlated to data volume
  - Power consumption
  - Range
  - Cost
- Other important characteristics
  - Architecture: Point-to-point vs. broadcast, # of supported nodes

Options available



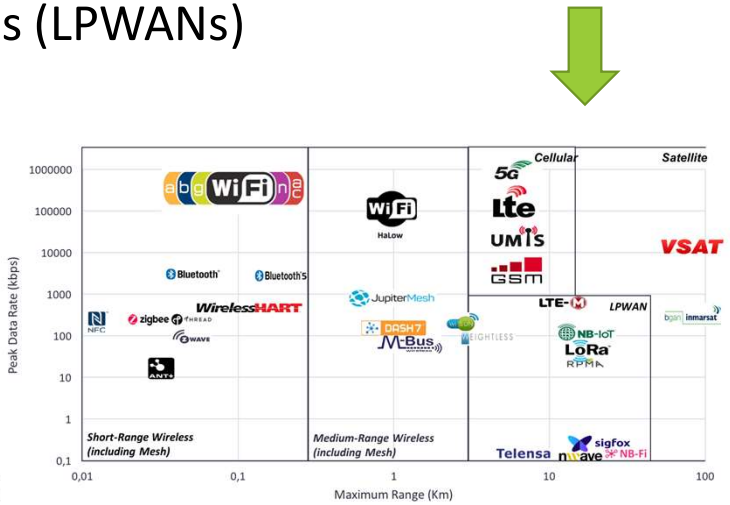
Please note that this chart is meant to show the maximum theoretical range and data rate for each technology, but this does not mean that the two can be achieved at the same time. On the contrary, no wireless technology can achieve the maximum range while transmitting at its peak data rate, but rather the higher is the used data rate, the lower is the achievable communication range.

Comparison chart

	Range	Standard	Max nodes	Tx Speed	Power/Energy	Price	Frequency
ANT	< 30m	Propriet.	∞ (2^{32})	1Mb/s	**~100 uW [700 nJ/bit]**	5$	2.4 GHz
Bluetooth (LE)	10m (100m)	Open	7+1	O(1 Mb/s)	**10mW, <15mA [70nJ/bit]**	5$	2.4 GHz
NFC	< 30cm	Open (multiple)	1 (M/S)	106 to 424 kbps	**<15mA**	10c	13.56MHz
UWB	< 10m	Open	7+1	110/480 Mb/s	**~200 uW**	$$$	3.1 to 6 GHz
Wifi	< 100m	Open	32	54 Mb/s	**250 mW @ 1Mbps [5.25nJ/bit]**	>10$	2.4 and 5 GHz
ZigBee	<100m	Open	∞ (2^{64})	250 Kbps	**O(10) mW [360 nJ/bit]**	<5$	2.4 GHz

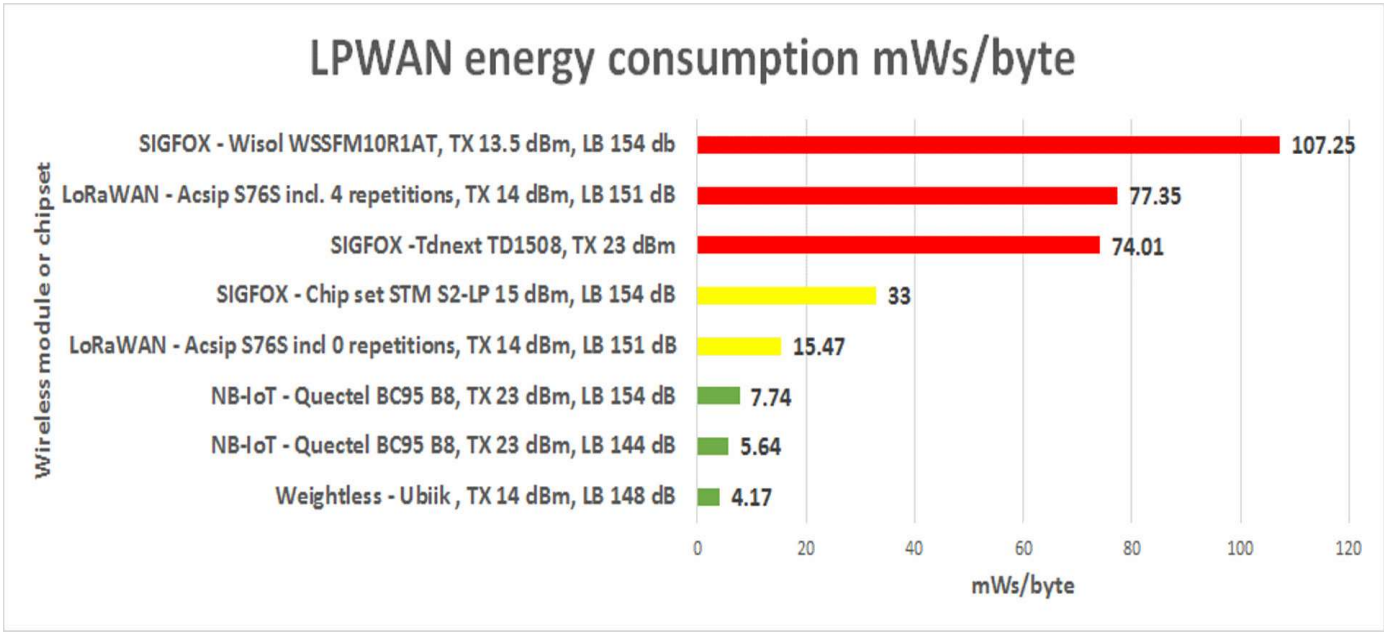
Long-range interfaces (distributed systems)

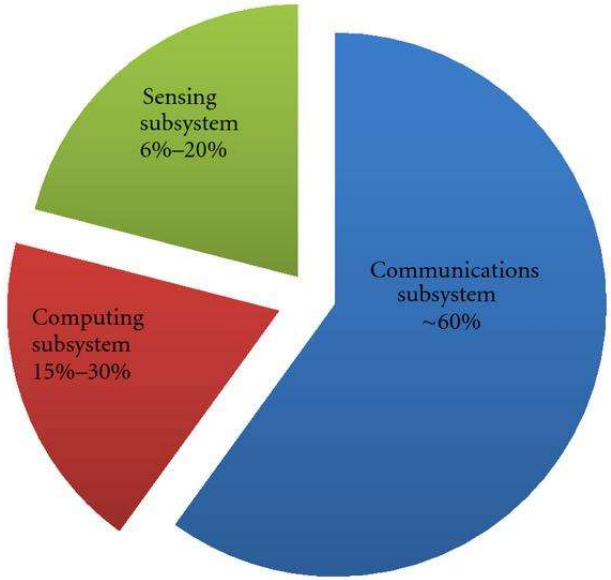
- An important segment for IoT are long-range radios (LPWANs)
  - **Tens of Km range, low-power, low-bitrate**



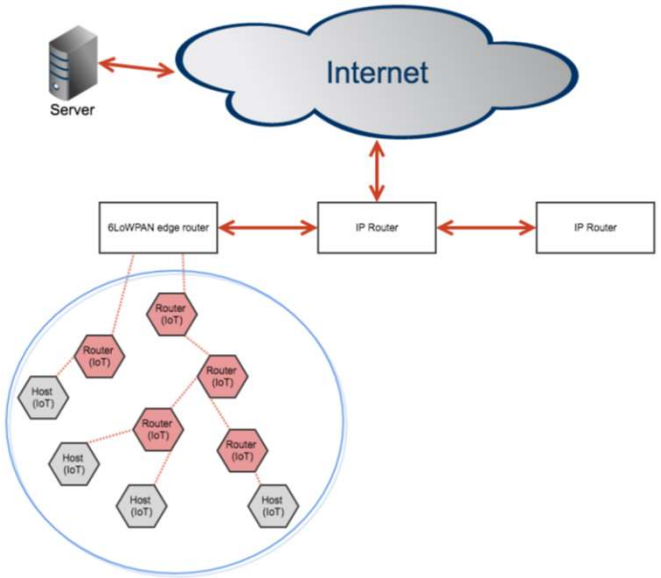
- Key protocols
  - SigFox, LoRaWAN, NB-IoT
- Operate in sub-GHz bands with very narrow bands
  - At those frequencies, signals penetrate obstacles and travel long distances while drawing relatively little power
- Bitrate from **100 bps** (SigFox) to **200 Kbps** (Nb-IoT)

Energy consumption

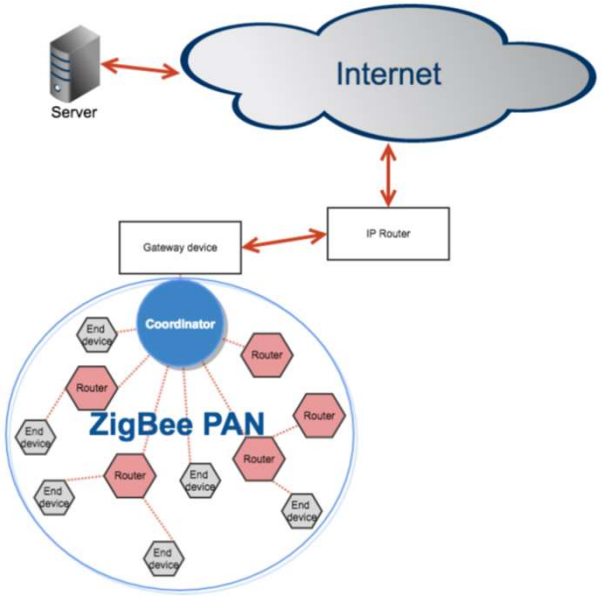




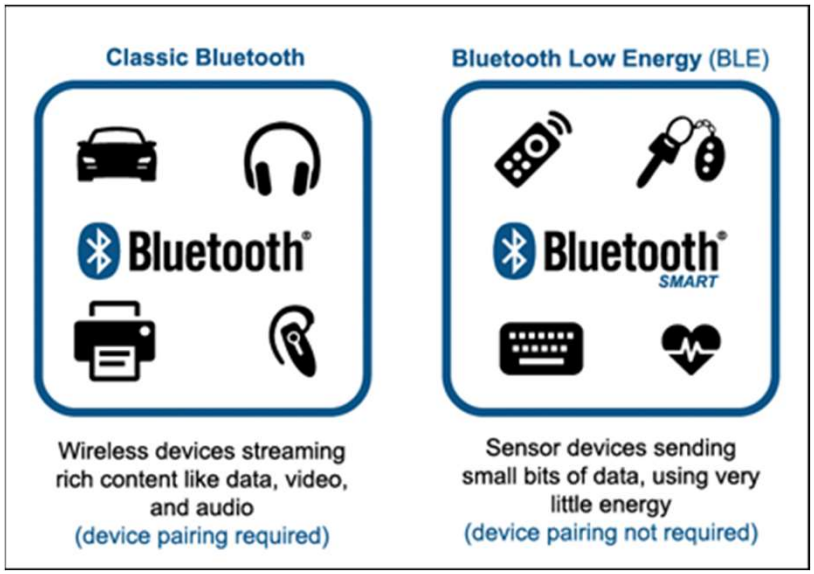
Mid-range: Wireless Local Area Networks



- Need a coordinator device that initiates network formation.
- More reliable and secure
- Less power
- Slower



Near, p2p communication

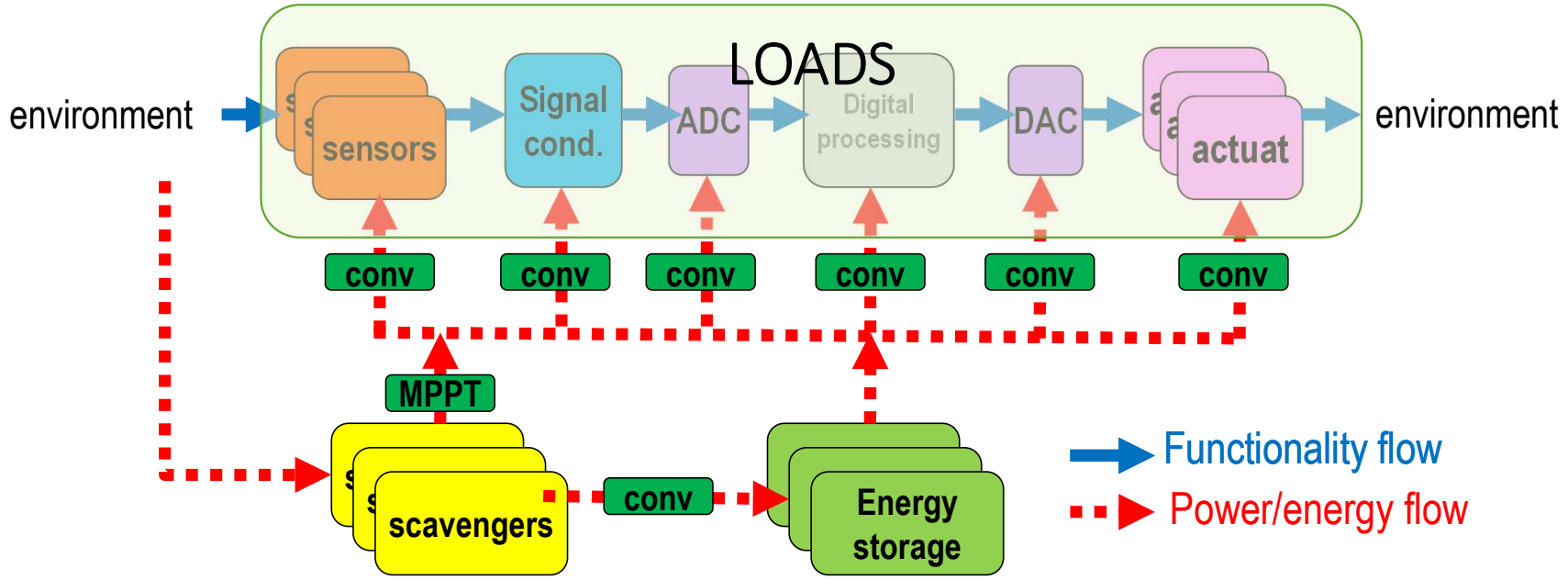


Energy

Why bother

- Many drivers for optimizing/controlling power consumption
  - Technological
    - Difficult to manufacture devices with high power consumption
    - Batteries do not catch up
- Market
  - How many devices you normally use are wired…?
- Economical
  - Costly to manufacture devices with high power consumption
- Environmental
  - Green computing….
  - Energy is a limited resource
  - Total Energy of Milky Way Galaxy: 1059 J !

Functional and energy/power flows

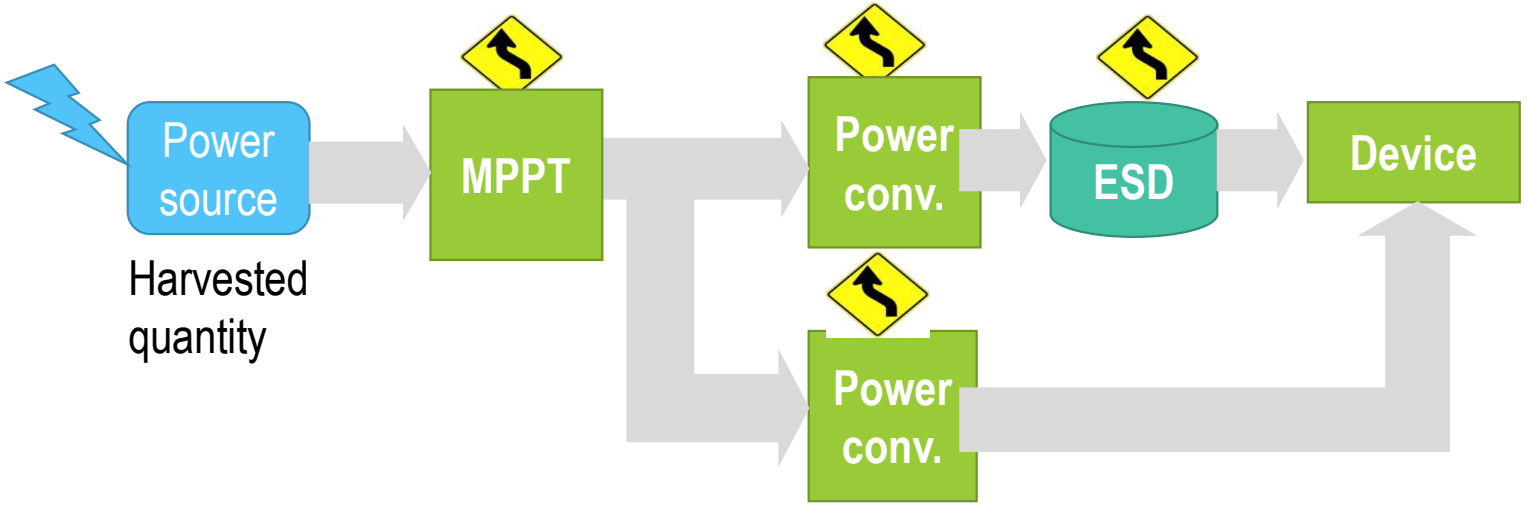


Energy is a resource to manage

- Functionality (@ low power) as commodity!
- Digital and non-digital functional parts (MCU, accelerators, memories, RF, I/Fs) are assumed to be power-optimized
  - Designer picks components, protocols, etc. based on needs and based on their power scores
  - System designer does not design the components but only decides the way they are used (SW)
- Important to reduce energy consumed, but (at least as important as) to manage how energy is:
  - Generated through various types of energy harvesters (power sources)
  - Distributed/converted to the various units
  - Stored in appropriate energy storage devices (ESDs) to guarantee sustained service

Energy management systems

- Accurate power/energy planning is non-trivial
  - Many points of losses
  - Sometimes contrasting…
  - Simple models can do the job of a quick assessment to avoid gross design mistakes!

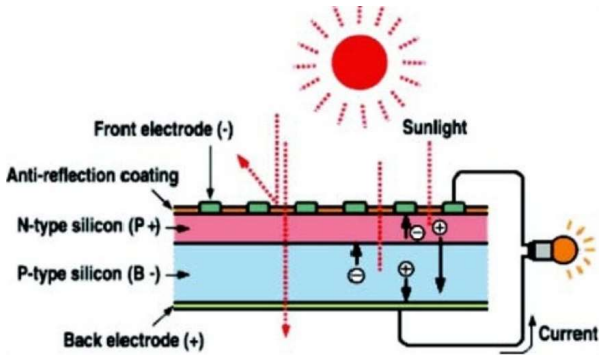


Energy harvesting

- = extracting energy from the environment
- We need electrical energy, so any harvester embeds some kind of transducer
- A large variety of physical effects can be exploited
  - Piezoelectric, photoelectric,triboelectric, thermoelectric,...
- As users, we are more interested in the “sources” from which we can extract energy…
- (solar) radiation
  - Vibration
  - temperature
  - friction
  - Ambient radio frequency/noise
  - Ambient airflow

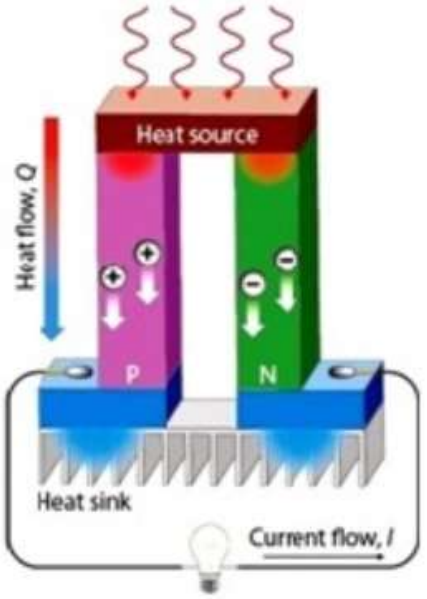
Harvesting from solar radiation

- Photelectric effect
  - Photelectric materials when exposed to light (photons) generate current
  - i.e., photons force electrons in the conduction band
- Silicon (P-n junctions) is one such material
  - A PV-cell is basically a diode and obey the basic diode rules
  - But reference current is determined by solar irradiance



Harvesting from temperature

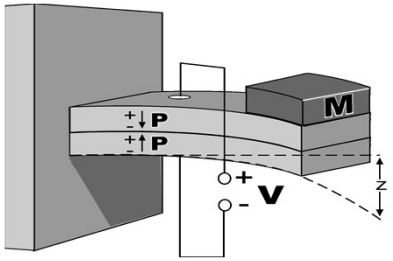
- Thermoelectric effect a.k.a. Seebeck effect
  - A potential is generated when the junctions of two dissimilar metals experience a temperature difference
  - Potential proportional to ΔT
- Inverse of Peltier effect (thermocouple)

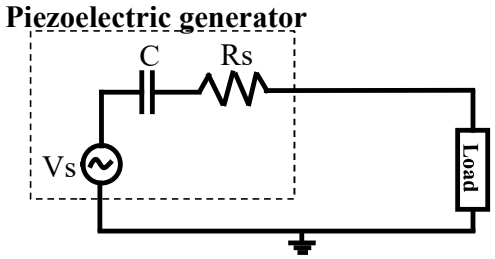


Harvesting from vibrations

Piezoelectric

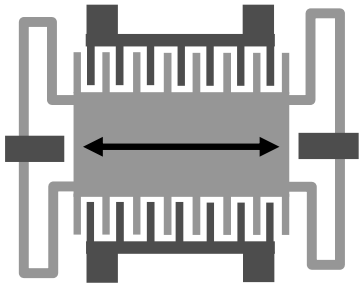
Strain in piezoelectric material causes a charge separation (voltage across capacitor)

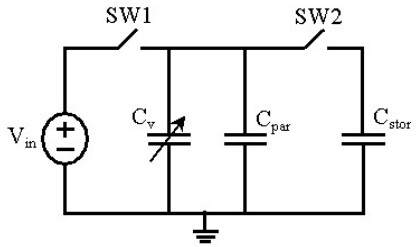




Capacitive

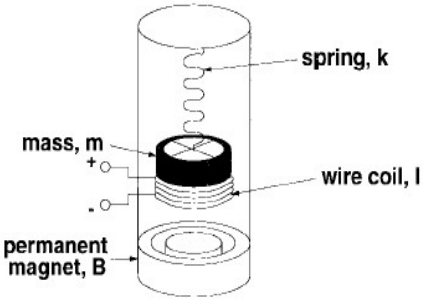
Change in capacitance causes either voltage or charge increase.





Inductive

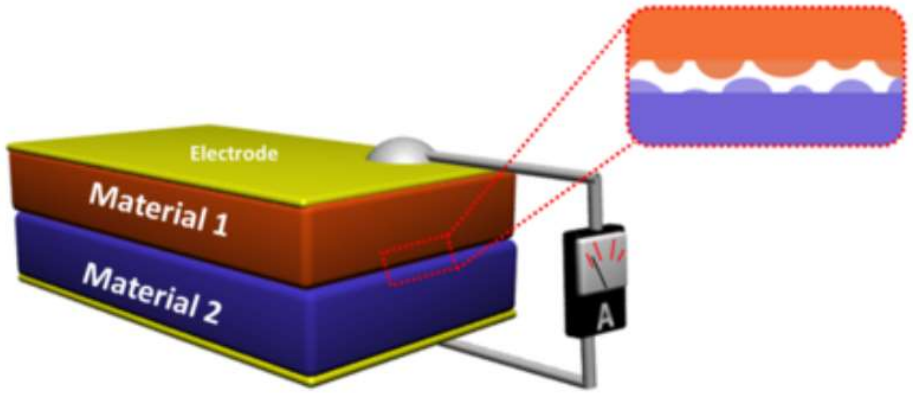
Coil moves through magnetic field causing current in the wire.



Amirtharajah et. al., 1998

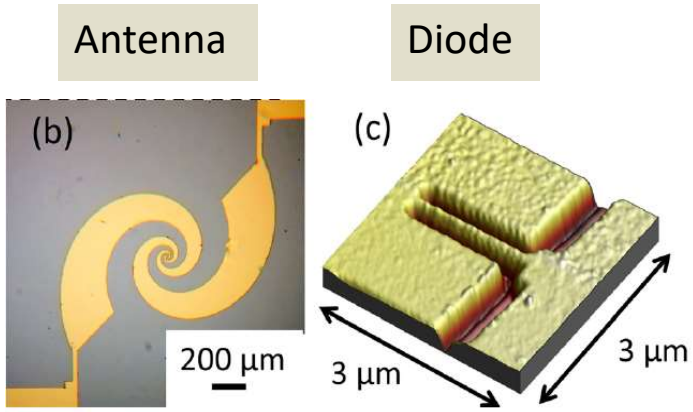
Harvesting from friction

- Triboelectric effect
  - Friction between materials can generate a potential
  - A particular instance of electrostatics…



Harvesting EM noise

- EM energy easy to convert into just DC electrical power
  - Typically, an antenna + rectifier will do the job!
- **Rectenna** (rectifying antenna)
  - Micro-scale rectennas are of interest!



Issues with energy harvesting

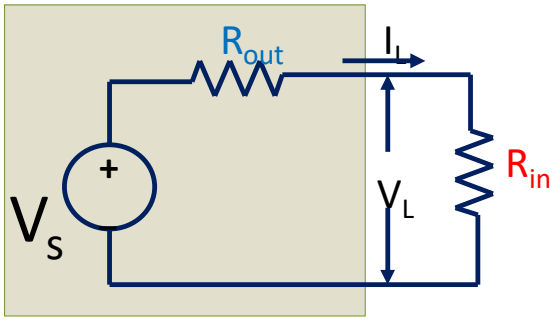
1. Generated power depends on the value of some environmental quantity
   - Q: How do we guarantee as much extracted power as possible under different environmental conditions?
   - A: The "maximum power extraction" problem
   - Q: How can we sustain computing under intermittent energy?
   - A: Intermittent computers
2. Extracted power is almost invariably very low…
   - Q: How can we derive acceptable power levels for typical computing blocks?
   - A: Appropriate power conversion, Scavenger "packing"
3. Extracted power might not be needed "right now"
   - Q: How can we accumulate excess energy?
   - A: Energy storage design & management

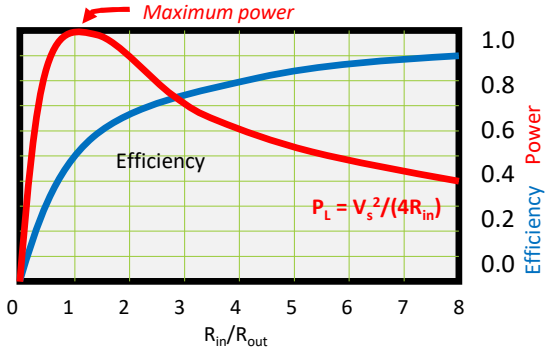
Harvesting method	Power density
Solar cells (outdoor at noon)	**15 mW/cm^2**
Solar cells (illuminated office)	**100 µW/cm^2**
Piezoelectric (shoe inserts)	**330 µW/cm^3**
Thermoelectric (10 °C gradient)	**40 µW/cm^3**
Acoustic noise (100 dB)	**1 µW/cm^3**

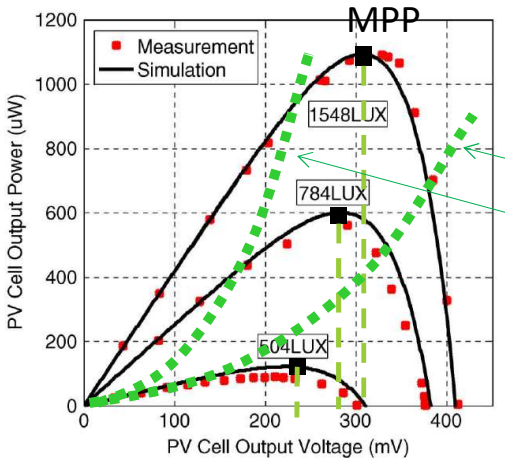
A typical battery cell has 100-1000 mW/cm^3 !!!

Extracting power

- Harvester has low efficiency if storage device is not impedance-matched to source !!!
  - Source impedance is variable
  - Maximum power point tracking (MPPT) problem
- Need proper circuits
  - Tradeoff between energy extracted and consumed by MPPT circuit







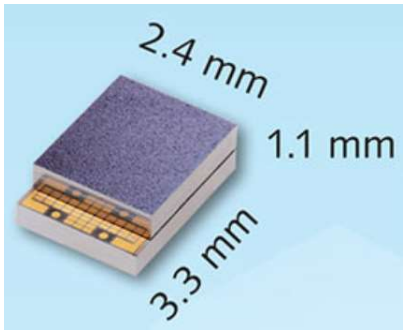
Resistive load curves (slope $P=V^2/R_{out}$)

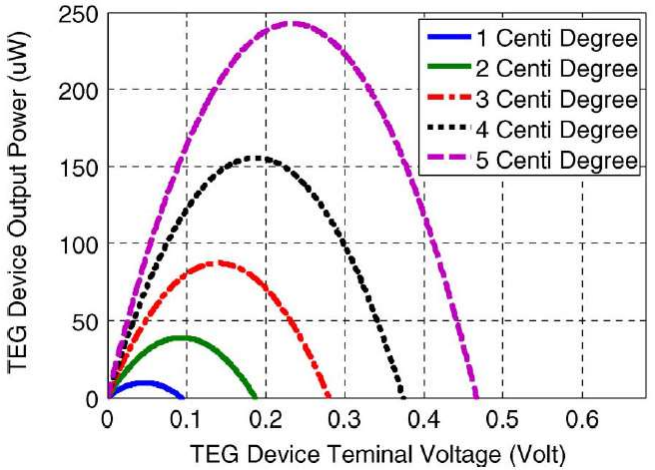
Tracking the MPP

- Done by ad-hoc HW call MPP tracker (MPPT)
- Basically a switching DC-DC converter that tries to match Rin and Rout by adapting the working point
- Double source of inefficiency !
  - Converter efficiency
    - Vmpp is not the level you need
  - Failure to perfectly track MPPT

Extracting from ultra-low power sources

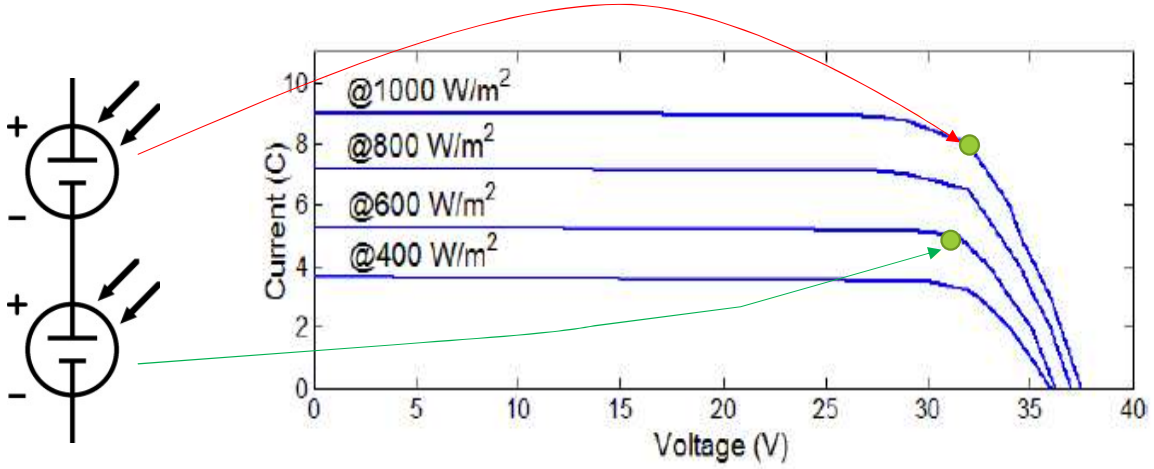
- Example:
  - TEG device Micropelt MPG-D751
  - MPPT for DT=5oC is only 50 uW!!! (@0.25V!)
- How to make such a low power level usable for a typical load (e.g., 10mA @ 1V = 1mW?)
  - Use appropriate converters
- If possible, combine power sources into “modules”
  - scale up both voltage and curretn
    - Series connection → increase voltage
    - Parallel connection → increase current





Packing power sources

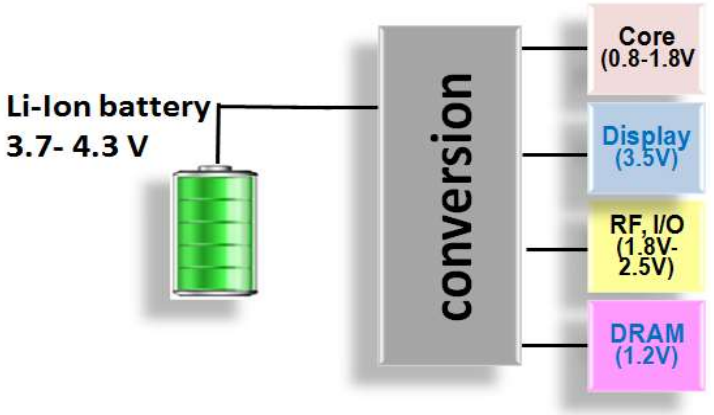
- Obviously, space constraints should allow this
- Issues with unmatched devices !!!!
  - i.e., exposed to different environmental conditions
  - Example:



Power Converters

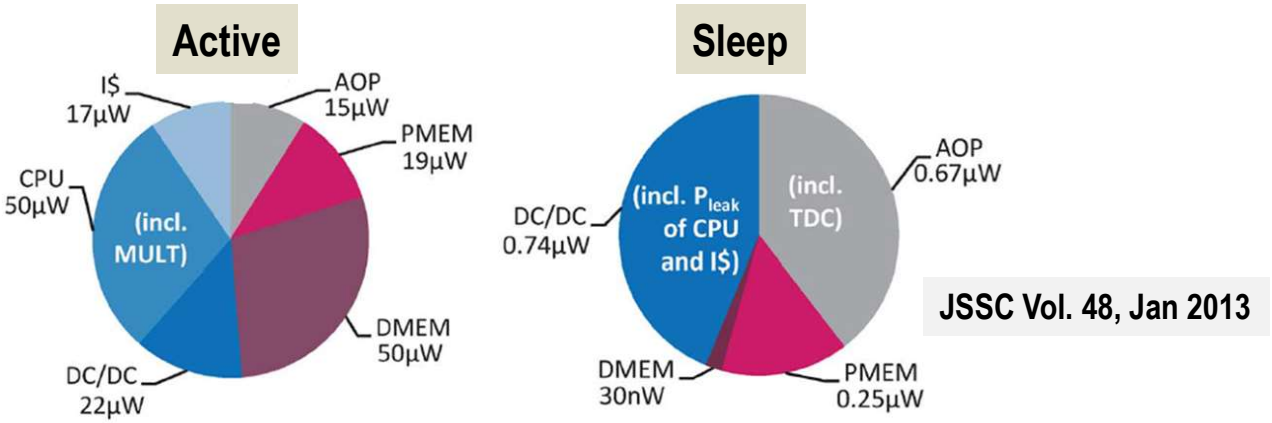
- a.k.a. Voltage regulators
  - The equivalent of transformers in DC!
- INPUT : an unregulated DC voltage Vin.
- OUTPUT: a regulated output voltage Vout, having a magnitude (and possibly polarity) that differs from Vin

- Conversion is ubiquitous!!!
  - Need it anytime you need to interface two power levels



Example figures

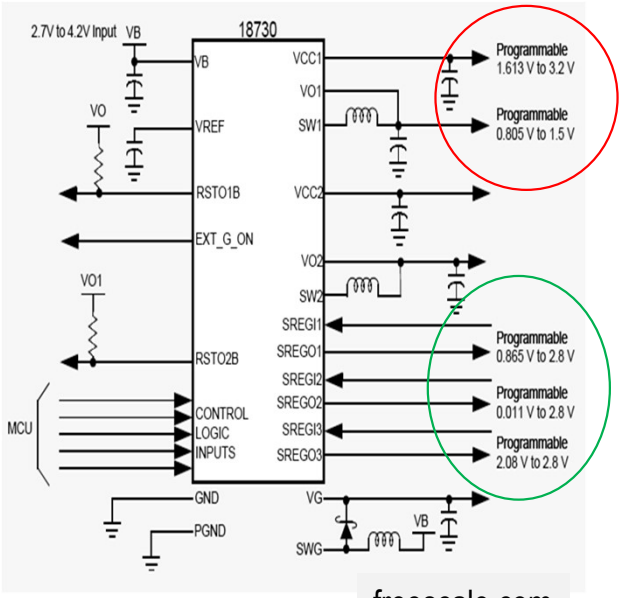
- SleepWalker node from IMEC
  - Actually a ULP µcontroller (7uW/MHz) including a DC/DC converter
- DC/DC losses take from about 12% to 40%
  - Small but even more significant in sleep state…
  - And this is just one converter…



JSSC Vol. 48, Jan 2013

Example figure

- Freescale's MPC18730 Multi Channel DC/DC Converter
  - 2 DC/DC Converters + 3 LDO (low-dropout) Regulators
- Example:
  - input from Li-Ion Battery 4.0V
  - DC/DC
    - Vout= 1.2V @ 50mA = 60mW
    - Vout= 2.5V @100 mA = 250mW
  - LDOs
    - Vout=1.8V @ 50mA = 90mW x 3
- Power loss in conversion
  - DC/DC = 85% efficiency
  - LDO = Vout/Vin= 45% efficiency

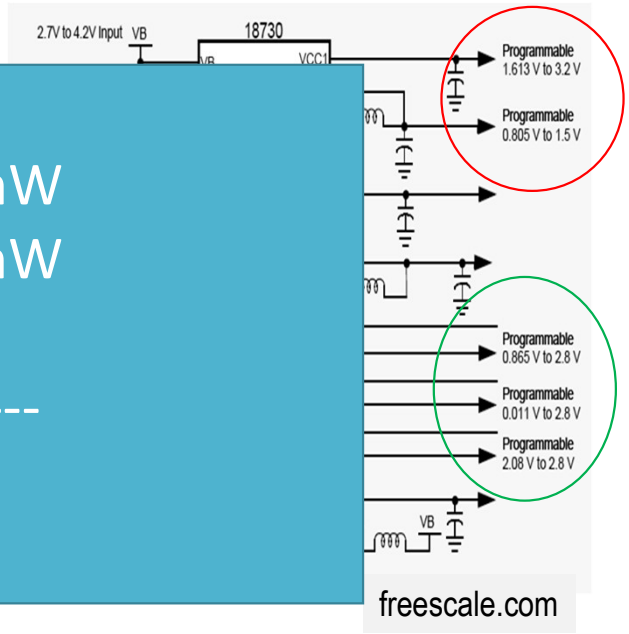


Example figure

- Freescale's MPC18730 Multi Channel DC/DC Converter
  - 2 DC/DC Converters + 3 LDO Regulators
- Example:
  - in
  - D
  - LD
- Powe
  - D
  - LD

Total loss:
DC/DC1 = 44mW
DC/DC2 = 10mW
LDOs = 110mW x 3 = 330mW

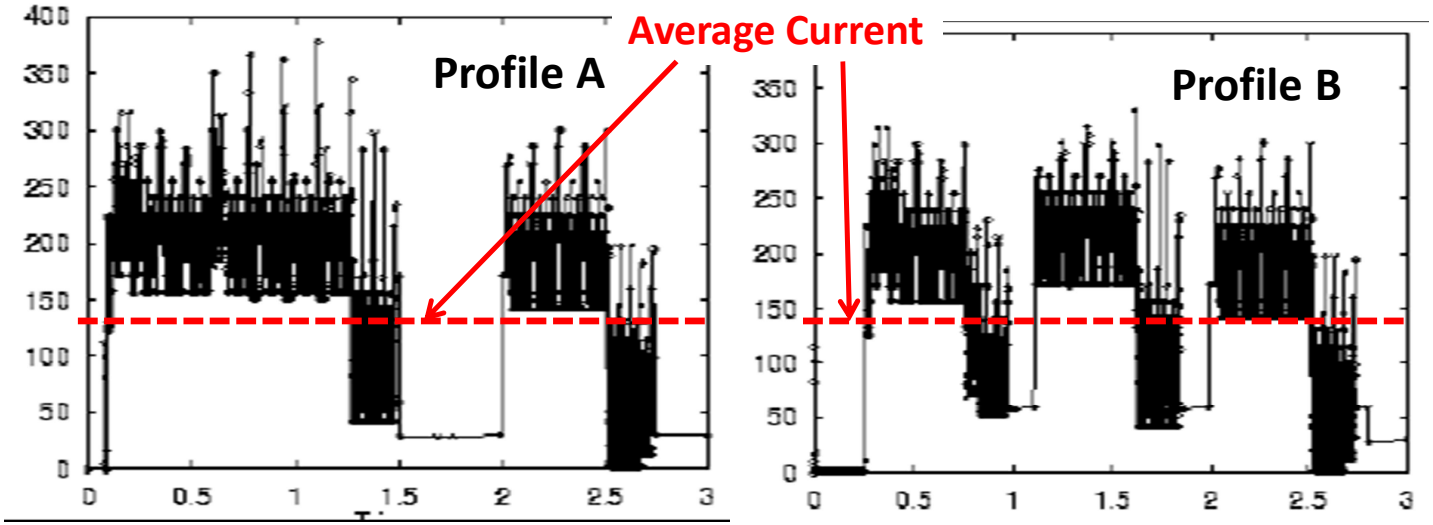
Total losses 384mW
(66% of consumed power !!!)



Energy Storage Devices (ESDs)

- Most typical embodiment of ESD is a battery
- Many possible alternatives (even at mscale)…
  - Ultra-capacitors
  - Fuel cells
  - Flywheels
  - …

Battery discharge path



[Source: Intel]

Profile	Average Current(mA)	Battery Life (ms)	Specific Energy(Wh/Kg)
A	123.8	357053	15.12
B	124.2	536484	18.58

Super- or Ultra-capacitors

- Supercapacitors (also known as ultracapacitor or double-layer capacitor) differs from a regular capacitor in that it has a very high capacitance
- But it is indeed a capacitor….
  - It stores energy by means of a static charge (as opposed to an electrochemical reaction)
  - Applying a voltage differential on the two plates will charge the capacitor

120F, 2.5V supercap – 20US$

Supercaps vs batteries: hybrid ESD

Figure of merit	Supercaps	Li-Ion (generic)
Charge time	Seconds	Hours
Cycle life	10^6	<1000
Cell voltage	2.3 – 2.75 V	3.6 – 4.3 V
Specific Energy [Wh/kg]	5-10	100-200
Specific Energy [W/kg]	Up to 10,000	1000 to 3000
Cost per Wh	$10-20	$1-2
Charge temperature	-40 to 65 ^oC	-0 to 45 ^oC
Discharge temperature	-40 to 65 ^oC	-20 to 60 ^oC