Field Devices for IIoT

Hardware and Hardware Abstractions

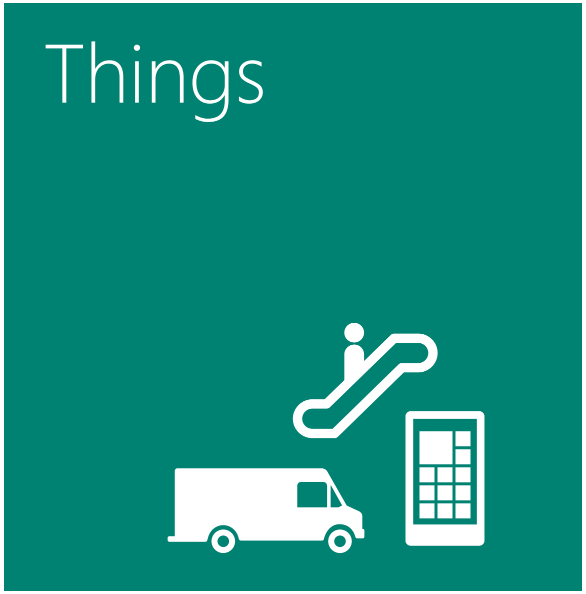
Design Challenges/Guidelines/Opportunities

Let's start From the edge..

- A Sensor Node (or *mote*) is a device with the following capabilities:
  - Sensing external phenomena
  - Processing information
  - Storing information
  - Communicating with other motes or devices
- The Actuator's tasks
  - Receiving input signals from control devices
  - Processing/storing information
  - Acting on the industrial process

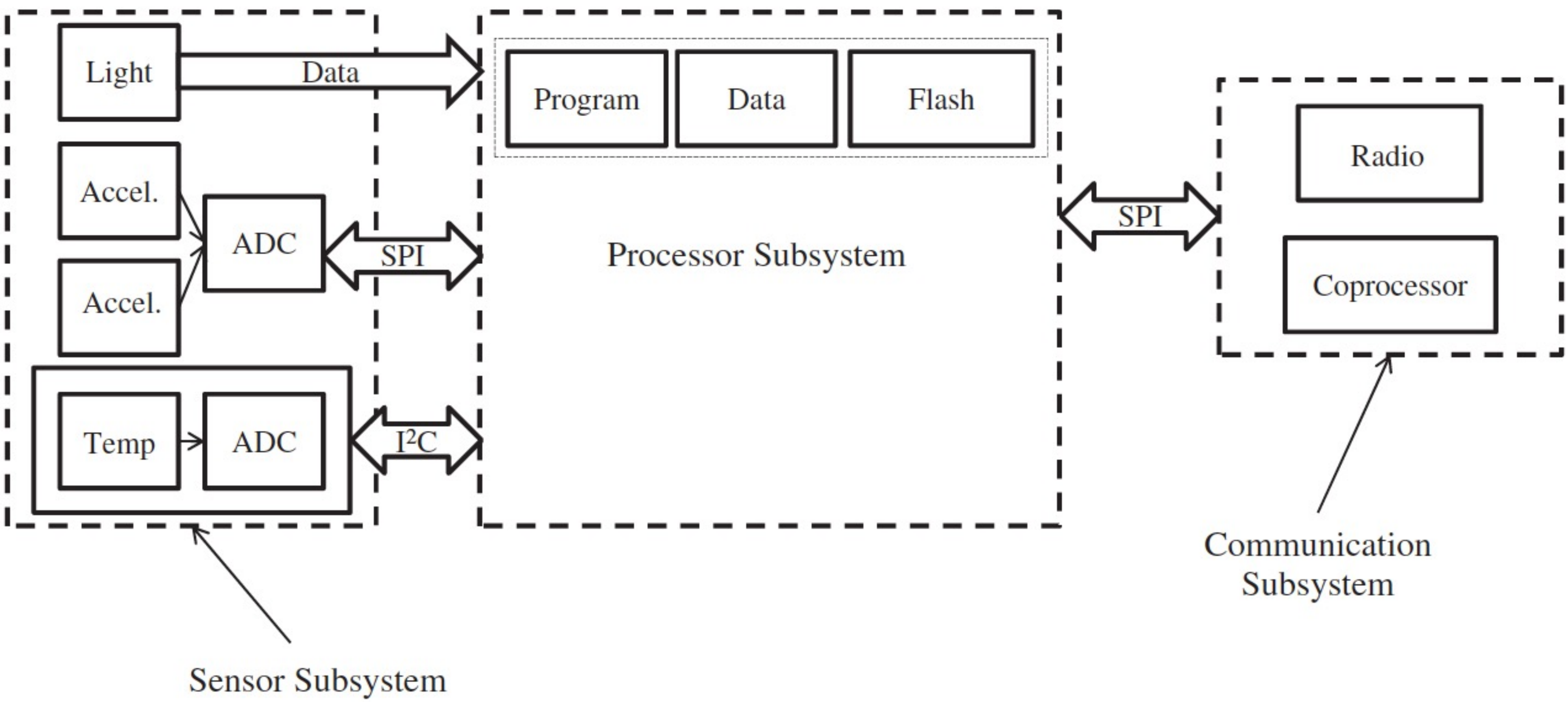
Any examplels??

The "Things"



- IT/OT check list:
  - What do you need to sense?
  - What constraint you have on:
    - Wired/Wireless
    - Flexibility/Programmability
    - Integration/size/design
    - Budget
    - H&M interfaces

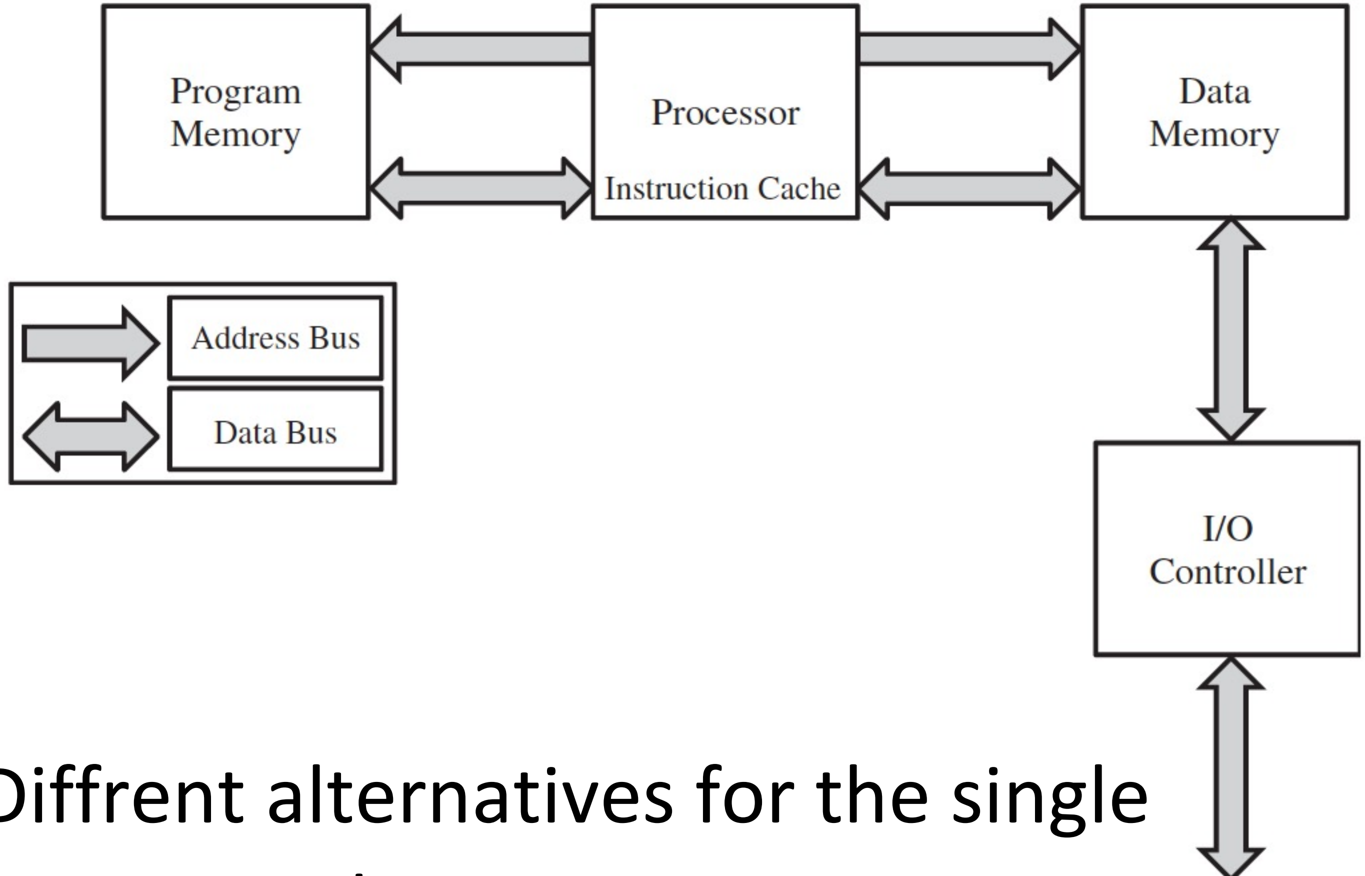
Mote Architecture



Fundamentals of Wireless Sensor Networks, W. Dargie

Processor Subsystem

- Often designed according to SHARC



Diffrent alternatives for the single components

Microcontroller

a single Integrated Circuit (IC) used for a specific application

- Processing
  - Central Processing Unit (CPU) – 4-bit to 64-bit processors
  - Clock generator (oscillator with quarz timimg crystals)
- Storage
  - Random Access Memory (RAM): volatile, easy-to-access memory
  - FLASH memory: non-volatile memory, block access only
  - Electrical Erasable Programmable Read-Only Memory (EEPROM): non volatile memory, byte access allowed
- Connectivity
  - Serial BUS: intra-IC connectivity (SPI, I2C)
  - Input/Output interfaces
  - Analog Digital Converters/Digital Analog Converters

Flexibility, low-cost ↑ Speed ↓

Digital Signal Processors(DSPs)

A DSP is a specialized microprocessor designed to process discrete signals thorugh digital filters

- DSPs can carry out complex mathematical operations at an extremely high efficiency, processing hundreds of millions of samples, every second and providing real-time performance

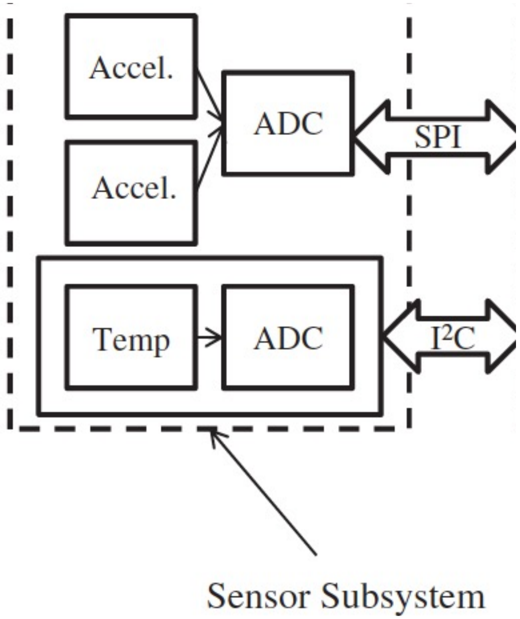
Speed, fine for data intensive operations (WMNs) ↑

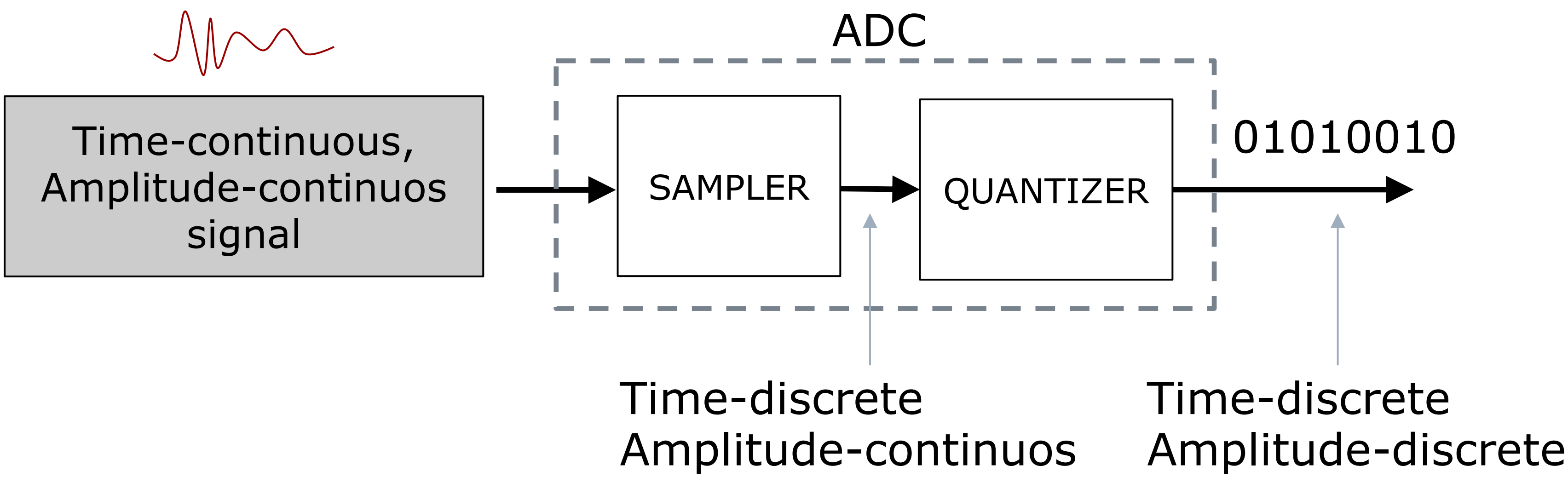
flexibility ↓

Application Specific IC (ASIC) and Field Programmable Gate Arrays (FPGA)

- ASIC: *integrated circuit (IC) that can be customized for a specific application*
  - High speed and customization
  - High development cost
  - Scarce flexibility
- FPGA: similar high-level architecture as ASICs
  - High speed (parallel programming allowed)
  - Moderate reconfigurability
  - High Complexity and cost

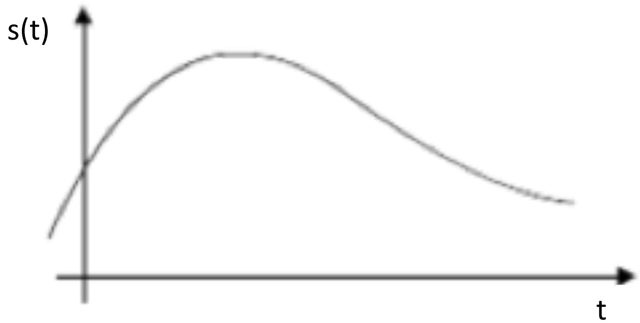
Sensor Subsystem - (Ideal) ADC Basics

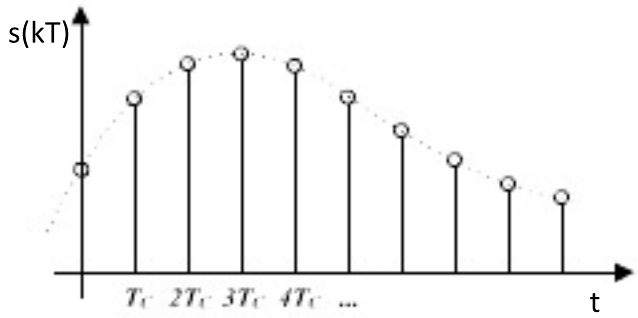




Sampling – Nyquist ADC

Reading the time-continuous signal at given points in time





Bandwidth (sampling rate): inverse of the sampling interval, $f_s=1/T$

Some magic: *if sampling frequency is properly set, then the initial signal can be lossless reconstructed from its samples*

Nyquist theorem

$f_s=2B$, B signal bandwidth

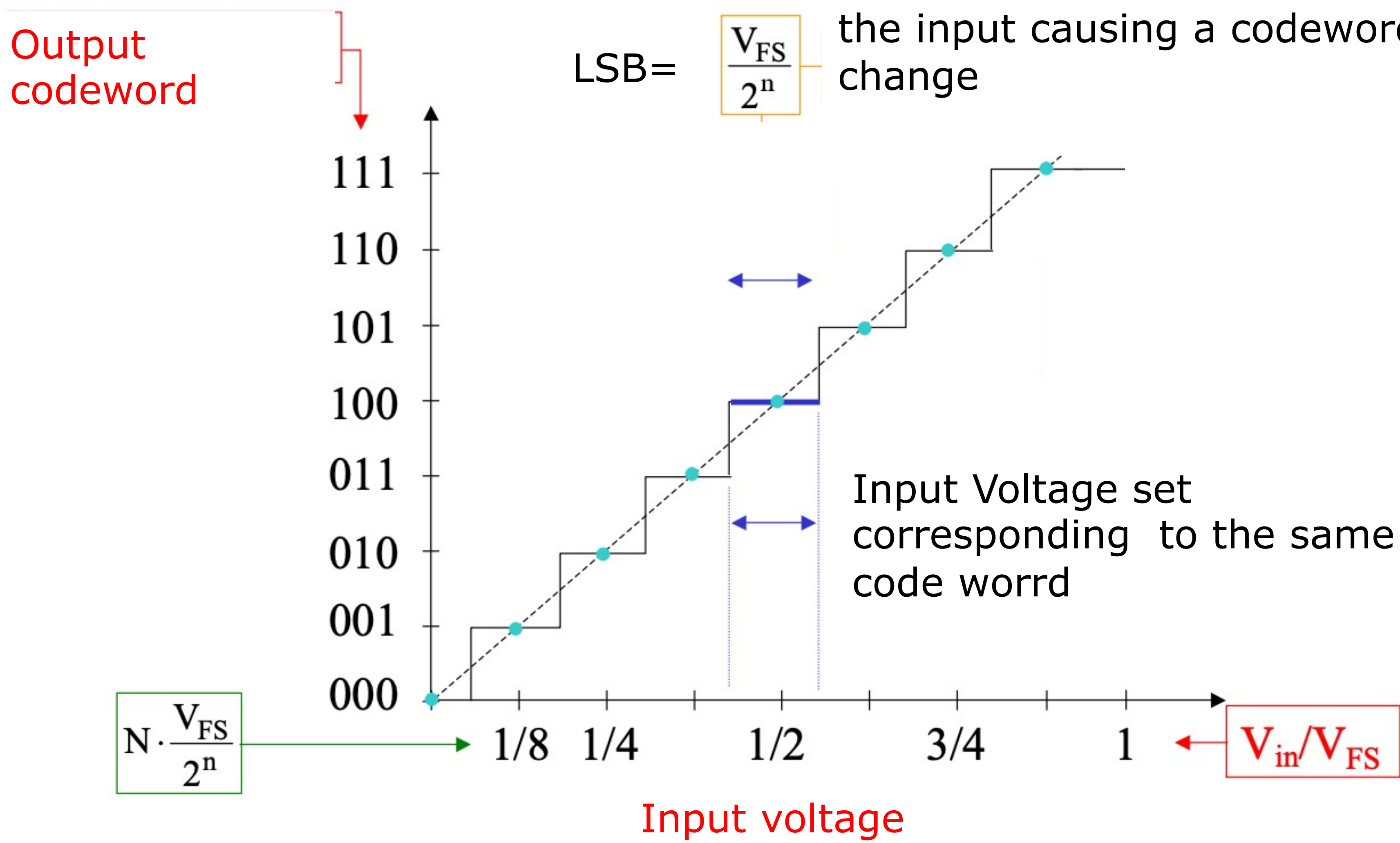
Quantization

- V_{in} is approximated by a digital codeword

Digital output: D_{n-1} D_{n-2} ......D_0

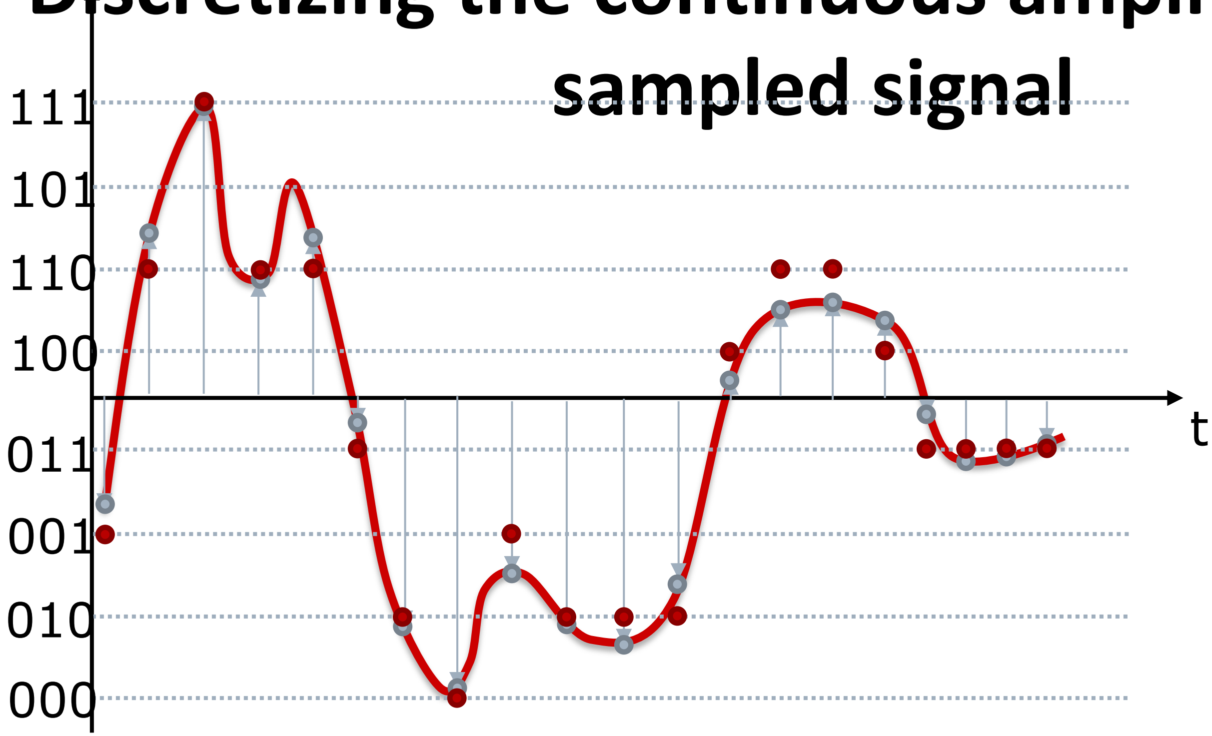
Ideal Quantizer – Input/Output

Resolution = lowest variation in the input causing a codeword change



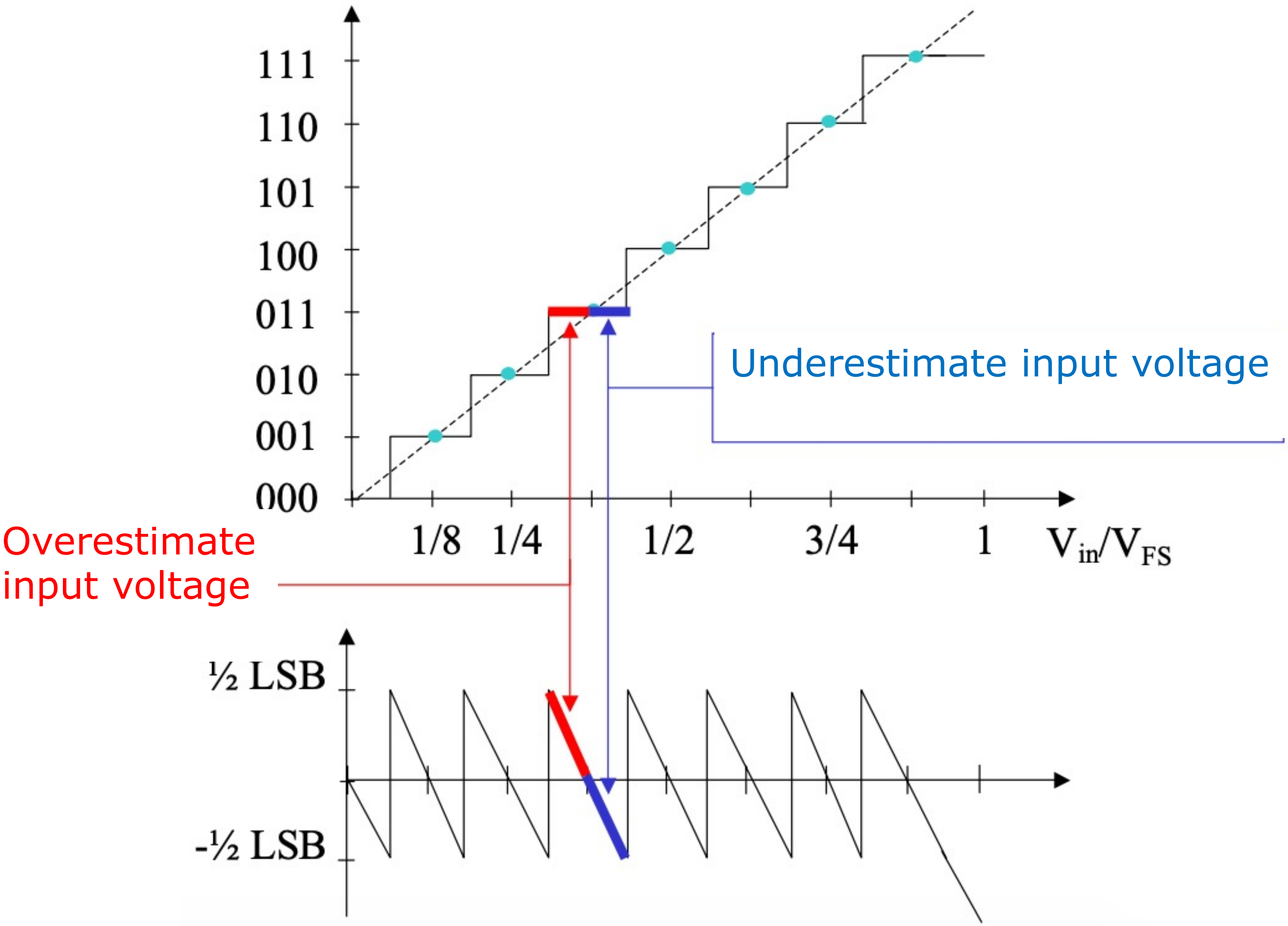
Quantization

Discretizing the continuous amplitude of the sampled signal

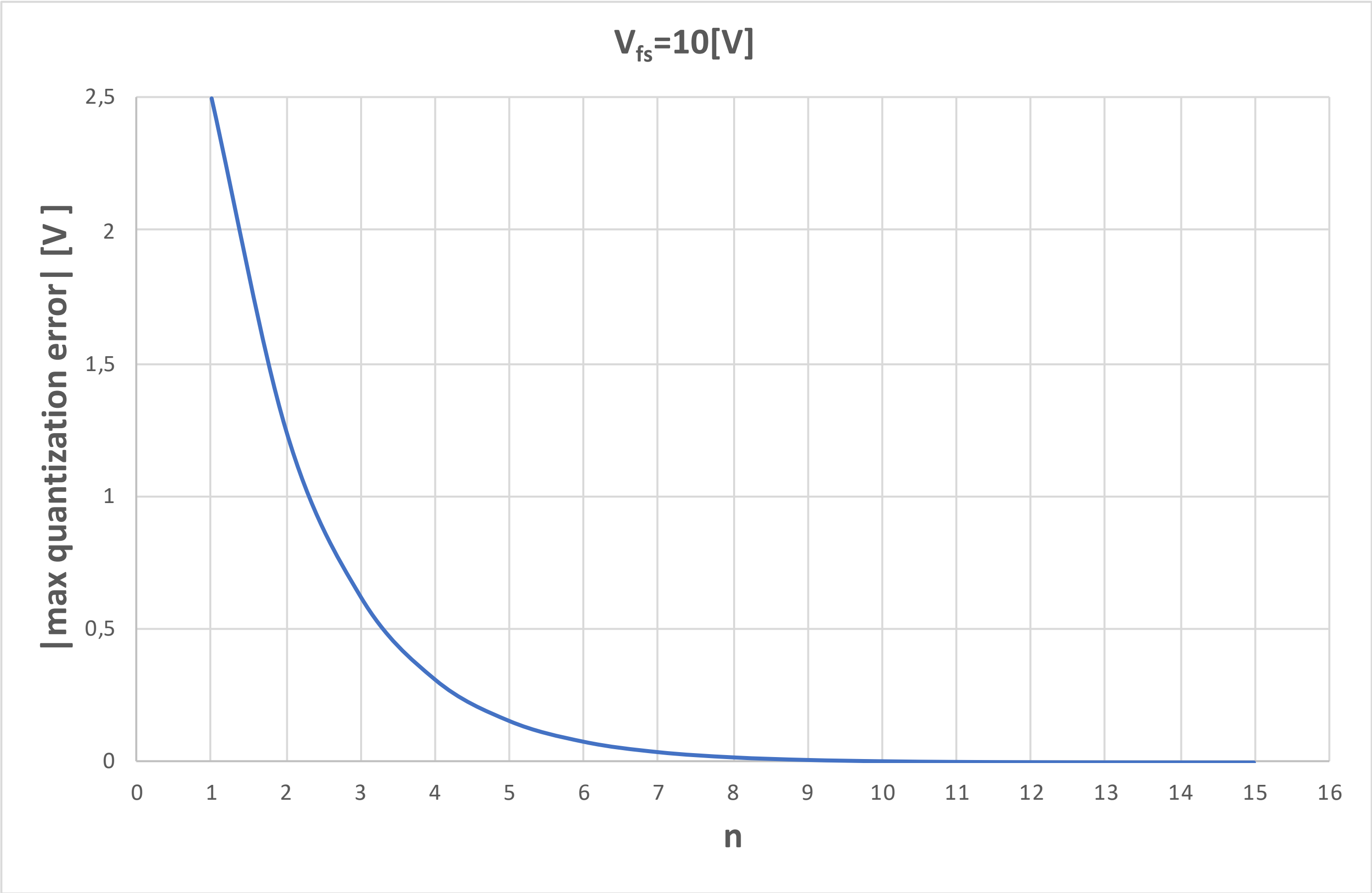


Sample 1 001
Sample 2 110
Sample 3 111
Sample 4 110
Sample 5 110
Sample 6 011
Sample 7 010
Sample 8 000
Sample 9 001

Quantization Error



Quantization Error



Quantization Error

7-bit Quantization

5-bit Quantization

3-bit Quantization

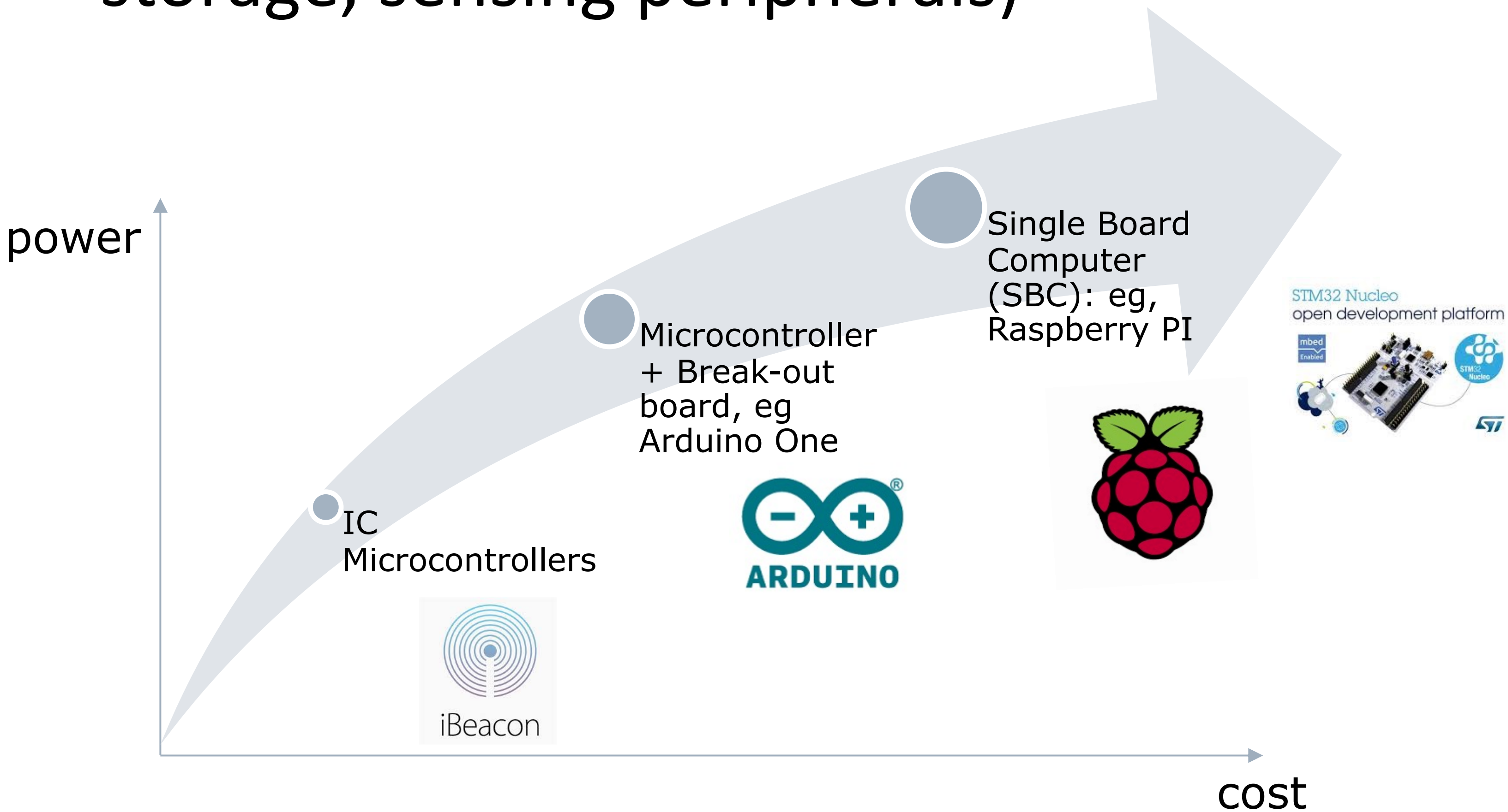
1-bit Quantization

Examples

	COM	CPU	BATTERY	STORAGE	SENSORS
	GPS, cellular, WiFi, BLE	S5, 64 bit dual core	Li-Po 296 mAh	32GB	Optical heart, accelerometer, gyroscope, ambient light, microphone, speaker, gymkit
	WiFi, BLE	32-bit ARM Cortex M3	Li-Ion 560 mAh	1MB Flash, 128 kB RAM	Accelerometer, Gyroscope, MFS, Light, Temperature, Pressure, Humidity

IoT Hardware Breakdown

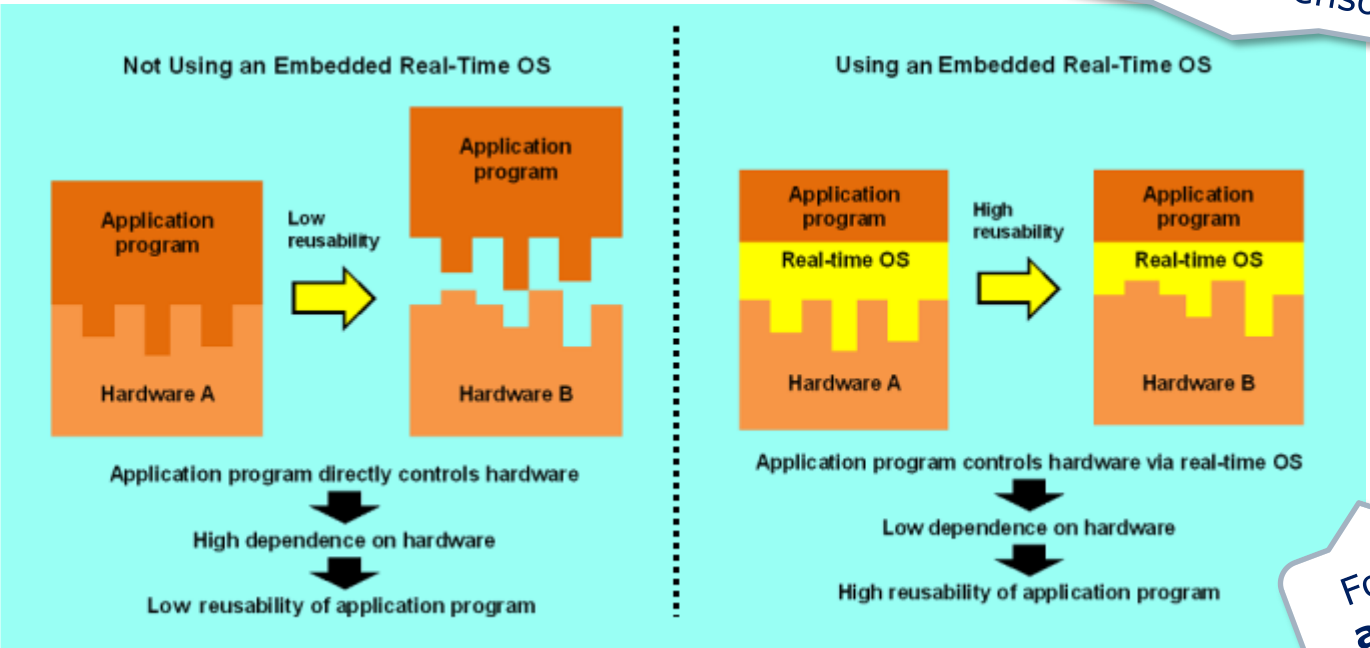
- IoT Hardware offer is vast, fragmented and heterogeneous (type of CPU, connectivity, storage, sensing peripherals)



Single Board Computer + OS

- End-devices/sensors become capable to run Operating Systems
- Why is this important?

"Outsource" generic tasks to the OS (ex: connectivity, security, sensors, …)



Source: http://www.tron.org/seminar/on-the-web-seminar/chap-2/

Focus on the **application**!

Separation of Application and Hardware Control

Sensors and boards evolution

Operating systems

- Battle of the Operating Systems
  - Commercial RTOS, open source RTOS, non-RT OS, ...

Open Source (RT)OS

RIOT OS

Linux

freeRTOS

ARM mbed

Offers from the big players

Samsung Tizen

Windows 10 IoT

Android Things / Weave

Embedded Apple iOS

Commercial RTOS

WindRiver VxWorks

Nucleus RTOS

Green Hills Integrity

What if Sensors/Actuators are Wireless?

No cables means no power and no wired connectivity

Wire-free Sensor: Energy efficiency becomes a MUST

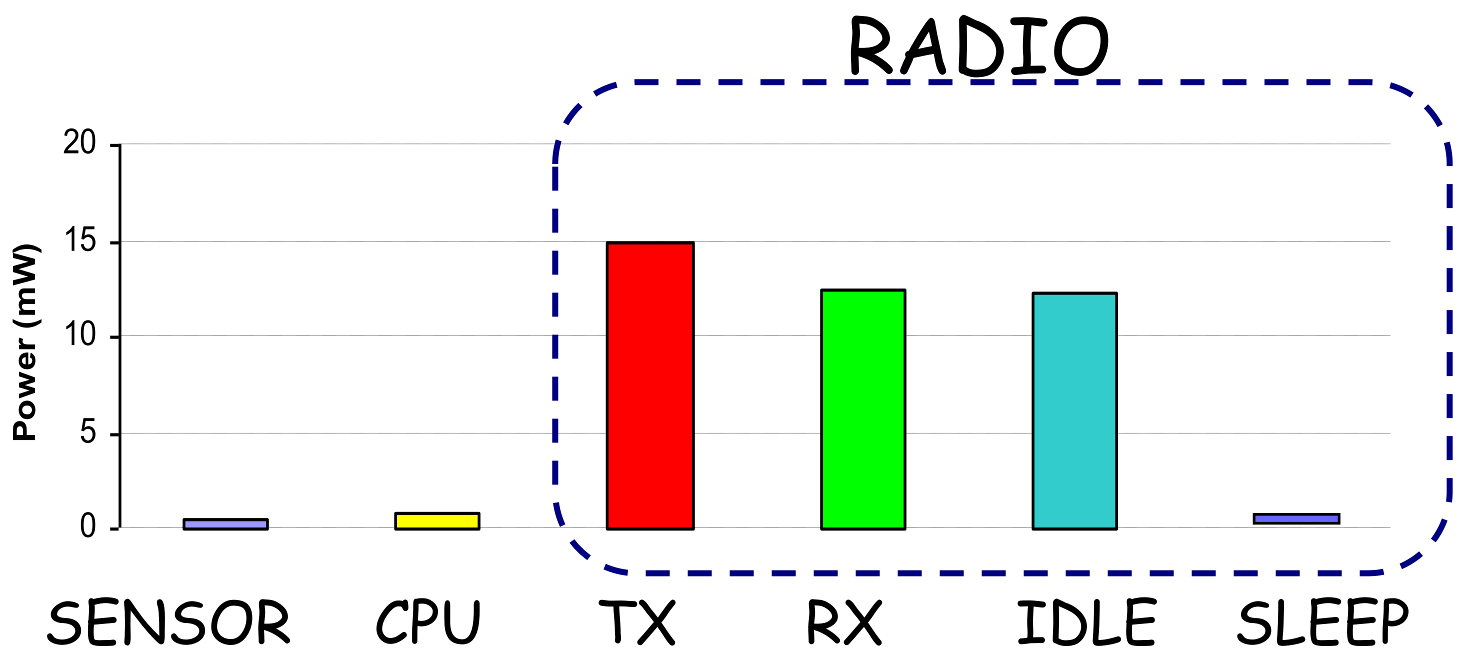
- Sensor node has limited power source
- Sensor node LIFETIME depends on BATTERY lifetime

- Goal: Provide as much energy as possible at smallest cost/volume/weight/recharge

- Problem: recharging and/or battery replacement may be immaterial or too expensive

- Options
  - Primary batteries – not rechargeable
  - Secondary batteries – rechargeable, only makes sense in combination with some form of energy harvesting

Power Consumption Dissected



General Design Guideline:

To switch off the Radio (TX/RX/IDLE) "as soon as possible"

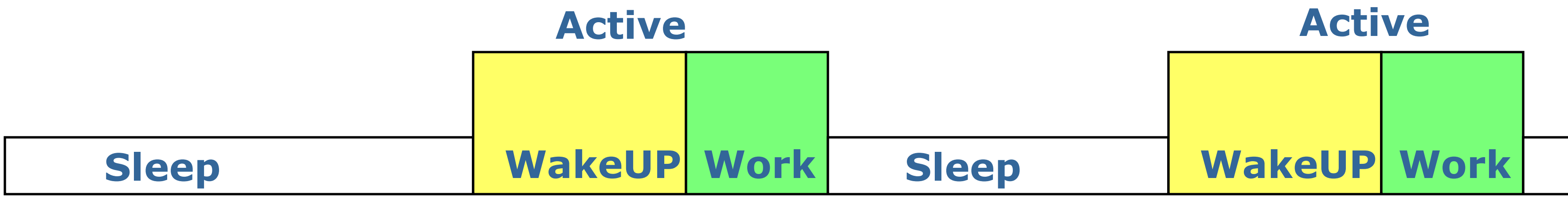
Source: Lecture slides of "Sensor Networks", Prof. I. Akyildiz

The "Idle Listening" Problem

- The power consumption of "short range" wireless communications devices is roughly the same whether the radio is transmitting, receiving, or simply ON, "listening" for potential reception (IEEE 802.15.4, Zwave, Bluetooth, WiFi)
- Circuit power dominated by core, rather than large amplifiers
- Radio must be ON (listening) in order receive anything.
  - Transmission is infrequent.
  - Listening (potentially) happens all the time

⇒ Total energy consumption dominated by *idle listening*

Power: Model of operation



- Sleep – Active [Wakeup / Work]
- Peak Power
  - MW in supercomputer, kW in server, Watts in PDA
  - milliwatts in "mote" class device
- Sleep power
  - Minimal running components + leakage
  - Microwatts in mote-class
- Average power
  - $P_{ave} = = (1-f_{active})*P_{sleep} + \boxed{f_{active}}*P_{active}$ Duty Cycle
  - $P_{ave} = f_{sleep}*P_{sleep} + f_{wakeup}*P_{wakeup} + f_{work}*P_{work}$
- Lifetime
  - EnergyStore / (P_{ave} - P_{gen})

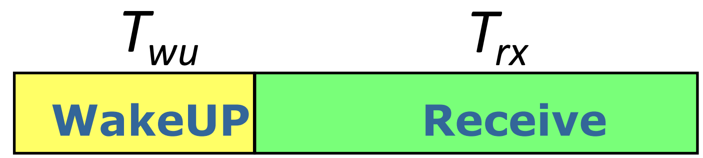
Source: Lecture slides of "Wireless Embedded Internetworking", Prof. D. Culler

Energy Consumption for Communication

TRANSMISSION

$$E_{tx} = P_{tx}(T_{wu} + T_{tx}) + P_o T_{tx}$$

RECEPTION



$$E_{rx} = P_{rx}(T_{wu} + T_{rx})$$

where

P_{tx} is power consumed by transmitter

P_{rx} is power consumed by receiver

P_O is output power of transmitter

T_{tx} time to transmit a packet

T_{rx} time to receive a packet

T_{wu} is start-up time for transmitter

Wake-Up Overhead

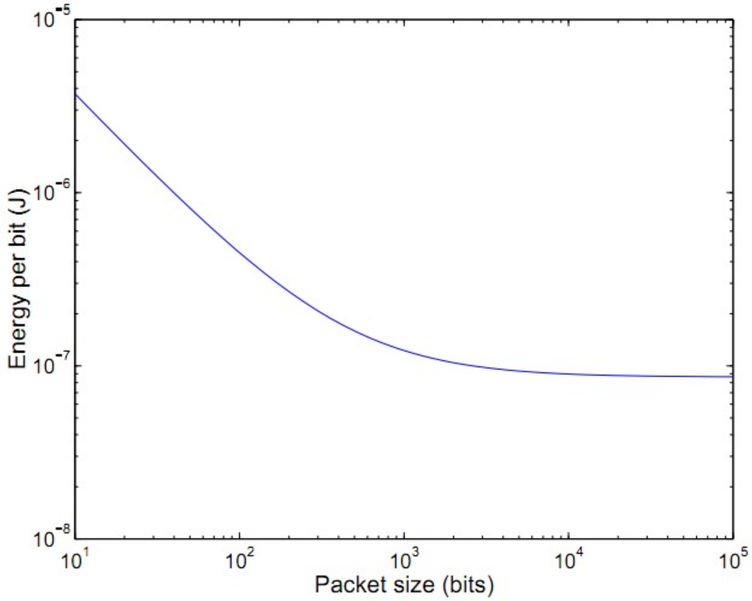
- Wake-Up comes with "energy overhead"

- Question:
  - What is the consumed energy per bit for transmitting a packet of *L* [bits]?
  - Energy spent in emission: $E_o=P_oT_L$
  - Energy spent during wake-up: $E_{wu}=P_{tx}T_{wu}$
  - Energy spent for TX circuitry: $E_{tx}=P_{tx}T_L$
  - Energy per bit: $(E_{wu}+E_o+E_{tx})/L$

[1] *E. Shih et al.,"Physical Layer Driven Protocols and Algorithm Design for Energy-Efficient Wireless Sensor Networks", ACM MobiCom, Rome, July 2001*

Wasted Energy

- Parameters: R=1 Mbps; T_{WU} ~ 450 msec, P_{tx}~81mW; P_{out} = 0 dBm



On the emitted power

- The emitted power is often a tunable parameter
- Good practice is to set it to the lowest value which allows for "good reception"
- The quality of the reception process is "measured" in terms of
  - Bit Error Rate (BER): fraction of bit not correctly received ("1" for a "0" or viceversa)
  - Packet Error Rate (PER): fraction of packet not correctly received
  - PER/BER relation (packet of length *l*, independent errors):

$$PER = 1 - (1 - BER)^l$$

Signal to noise and Interference Ratio

- BER (and PER in turn) depends on the "level of noise" in the TX/RX channel, which, in turn, depends on the transmitted/received power

$$\mathsf{SINR} = 10\log_{10}\left(\frac{P_{\mathsf{recv}}}{N_0 + \sum_{i=1}^{k} I_i}\right)$$

Thermal noise: *KTB*

- BER can be computed once given the specific TX/RX channel (modulation) and the specific SINR

Receiver Sensitivity

- Each receiver is characterized by a *sensitivity* parameter (e.g. P_{min}=-95dBm),
  The minimum input signal power needed at receiver input to provide adequate SNR at receiver output to do data demodulation

- Example: IEEE 802.15.4
  - Receiver sensitivity (packet error rate < 1%)
    - $P_{min} > -85$ dBm @ 2.4 GHz band
    - $P_{min} > -92$ dBm @ 868/915 MHz band
- Knowing such parameter, one can find the required emitted power at the transmitter by inverting the propagation law of the channel to get to required emitted power

Emitted power - Example

- A wireless receiver is characterized by a sensitivity P_r=-0.1[uW]; the transmitter is d=10[m] from the receiver; the TX-RX is performed at a carrier frequency f=2.4GHz; the propagation on the channel is characterized by the following model

$$^{*}P_r = P_t g_t g_r \left(\frac{\lambda}{4\pi d}\right)^2$$

- Further assuming the antenna gains equal to 1, we get a required emitted power

$$P_t = P_r \left(\frac{4\pi d}{\lambda}\right)^2 \approx 100[mW]$$

*If you have no clue on where this model comes from you may want to have a look at the primer on wireless propagation available at the course web site

Processing Power Consumption

- CPU power dissipation due to:

$$P_p = P_{dyn} + P_{sc} + P_{leak}$$

Job done — Short circuits — leakage

  - Where

$$P_{dyn} = CfV^2$$

  - C: capacitance (~ 0.67nF)
  - f: frequency
  - V: voltage

Processing Power Consumptions

- Rough Comparison:
  - Energy cost of transmitting 1 KB a distance of 100 m is approx. equal to executing 3 Million instructions by a 100 million instructions per second processor.

- Local data processing (if possible) is crucial in minimizing power consumption in a multi-hop network

Source: Lecture slides of "Sensor Networks", Prof. I. Akyildiz

Memory and Sensing Power Consumption

- Power consumption due to memory access
  - Crucial part is FLASH memory, Power for RAM almost negligible
  - FLASH writing/erasing is expensive (e.g., on Mica motes - Reading: 1.1 nAh per byte, writing: 83.3 nAh per byte)
- Power consumption due to sensing
  - Highly dependent on the sensor
  - Rough model for ADC:

$$P_s \sim f_s 2^n$$

Source: Lcture slides of "Sensor Networks", Prof. I. Akyildiz

Design Guidelines

- Do not run motes at full operation all the time
  - If nothing to do, switch to power safe mode
  - Question: When to throttle down? How to wake up again?
- Typical modes
  - Controller: Active, idle, sleep
  - Radio mode: Turn on/off, transmitter/receiver, both

Source: Lecture slides of "Sensor Networks", Prof. I. Akyildiz

Optimize Power Consumption

- Energy aware software
  - Power aware OS: dim displays, sleep on idle times, power aware scheduling
- Energy aware packet forwarding
  - Radio automatically forwards packets at a lower level, while the rest of the node is asleep
- Energy aware wireless communication
  - Exploit performance energy tradeoffs of the communication subsystem, better neighbor coordination, choice of modulation schemes