Power Efficiency in the Design of IoT Devices

Enrico Macii, Politecnico di Torino
ST-Polito s.c.ar.l. – Torino, Italy

Massimo Poncino, Politecnico di Torino – Torino, Italy

Michelangelo Grosso, ST-Polito s.c.ar.l. – Torino, Italy

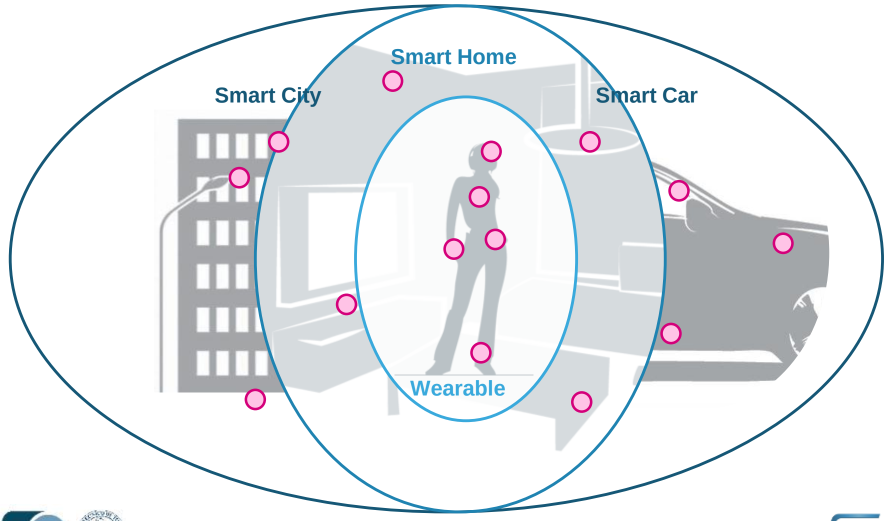
1st International Forum on Research and Technologies for Society and Industry
Torino, Italy – September 16-18 2015

Agenda

- Internet of Things scenario and main challenges
  - E. Macii, Politecnico di Torino
    ST-Polito s.c.ar.l. – Torino, Italy
- Principles of Power Optimization:
  Computation, Communication and Power Delivery
  - M. Poncino, Politecnico di Torino – Torino, Italy
- Energy optimization & system-level modeling – Case studies
  - M. Grosso, ST-Polito s.c.ar.l. – Torino, Italy

IoT: Expanding to make things smarter

Beyond the Smartphone



IoT: Expanding to make things smarter

Main ideas

1. Collecting data from sensors (things) much more cost effectively than ever before because sensors are ubiquitous and easy to install
2. Interpreting this data strategically using big data analytics and other techniques to turn the data into valuable information
3. Delivering new services or performance improvements
4. Presenting added-value information to the user or other person, e.g., medical doctors or plant personnel or remote experts, at the right time

Internet of Things: a definition

- The development of the IoT market is based on two main pillars that evolve in parallel
  - "Big data" analysis
  - Internet-connected "smart" devices

Smart System = Sensor + Brain + Communication (RF)

IoT = Smart Systems + @

Smart Systems for IoT

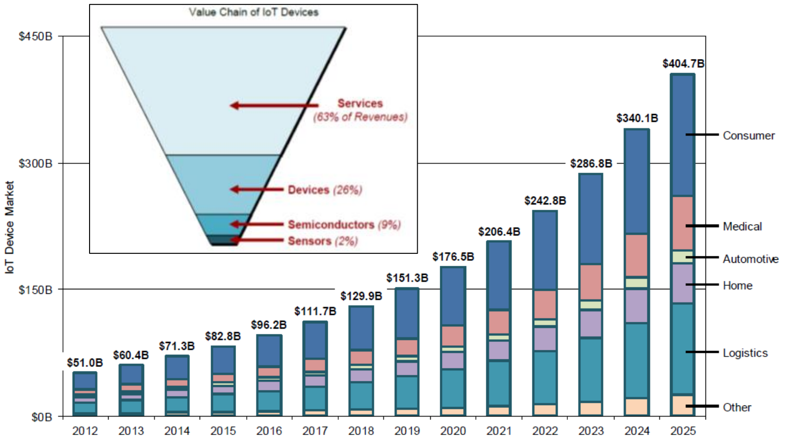
Application domains

- Increasingly wide range of contexts and environments
  - From everyday tasks to more complex and critical missions
- Market segments
  - Consumer
  - Medical
  - Automotive
  - Smart home
  - Logistics
  - Other (aerospace, instrumentation, robotics, drones, military, …)

Smart Systems for IoT

Device market

- Hardware, device software and service of IoT device vendors

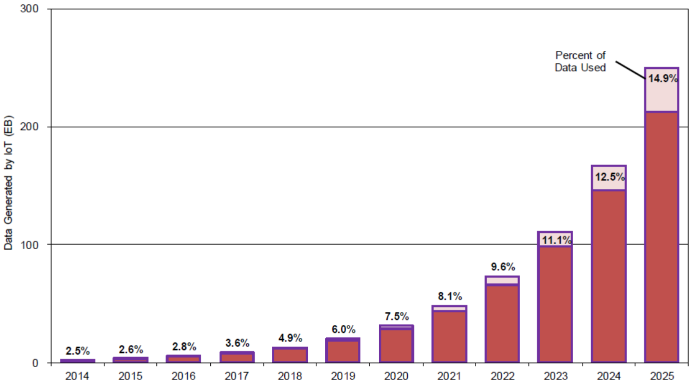


© 2015 International Business Strategies, Inc.

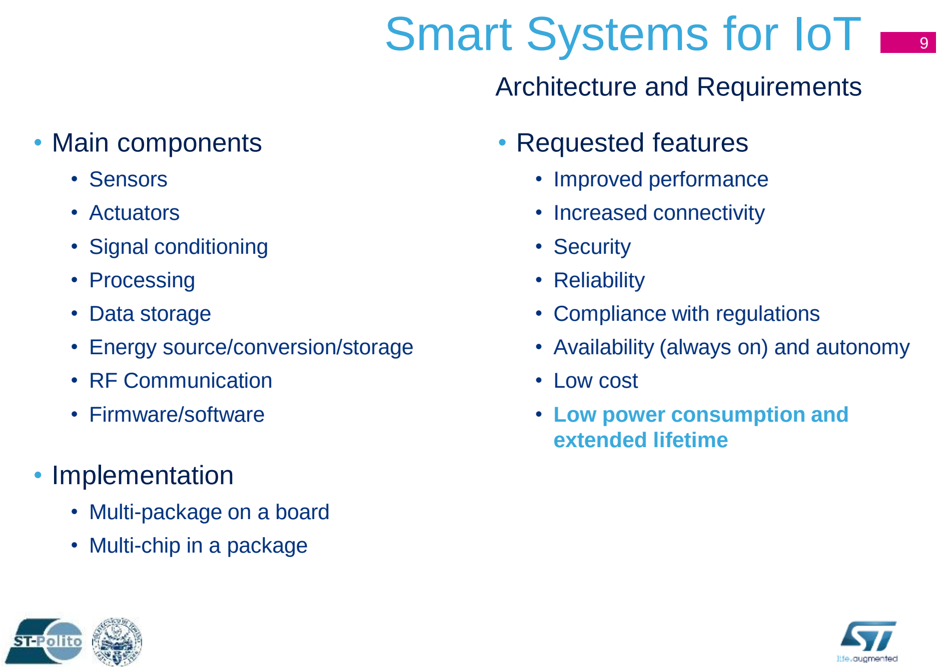
Smart Systems for IoT

Production and usage of data

- The adoption and spread of devices strongly depends on the use that we can make of data



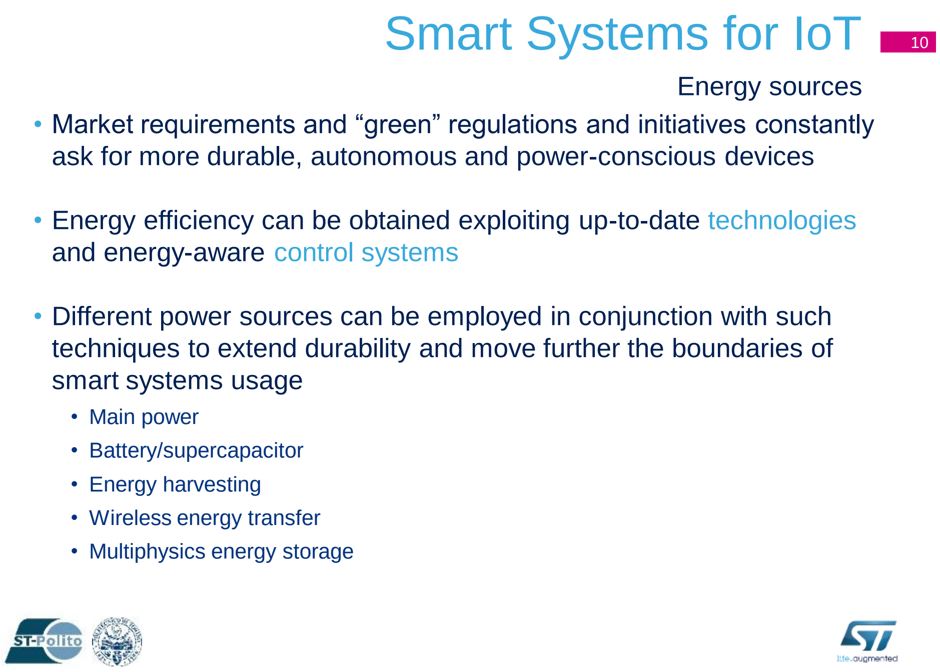
© 2015 International Business Strategies, Inc.



Smart Systems for IoT

Architecture and Requirements

- Main components
  - Sensors
  - Actuators
  - Signal conditioning
  - Processing
  - Data storage
  - Energy source/conversion/storage
  - RF Communication
  - Firmware/software
- Implementation
  - Multi-package on a board
  - Multi-chip in a package
- Requested features
  - Improved performance
  - Increased connectivity
  - Security
  - Reliability
  - Compliance with regulations
  - Availability (always on) and autonomy
  - Low cost
  - **Low power consumption and extended lifetime**

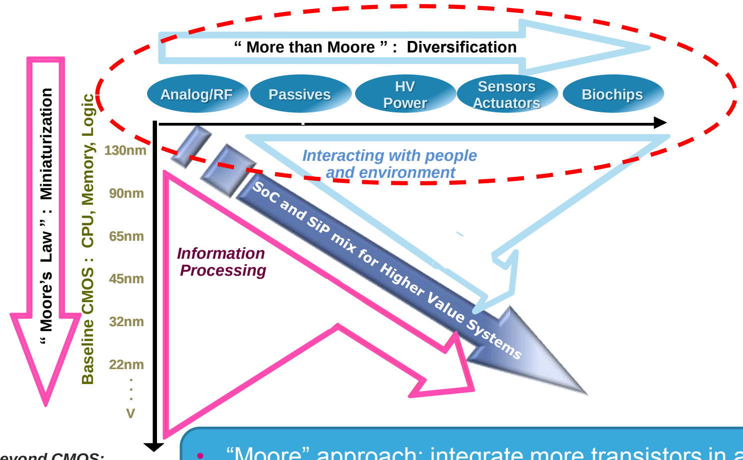


Smart Systems for IoT

Energy sources

- Market requirements and “green” regulations and initiatives constantly ask for more durable, autonomous and power-conscious devices
- Energy efficiency can be obtained exploiting up-to-date technologies and energy-aware control systems
- Different power sources can be employed in conjunction with such techniques to extend durability and move further the boundaries of smart systems usage
  - Main power
  - Battery/supercapacitor
  - Energy harvesting
  - Wireless energy transfer
  - Multiphysics energy storage

Emerging applications require Smart Integration



Beyond CMOS:
Quantum Computing,
Molecular Electronics
Spintronics

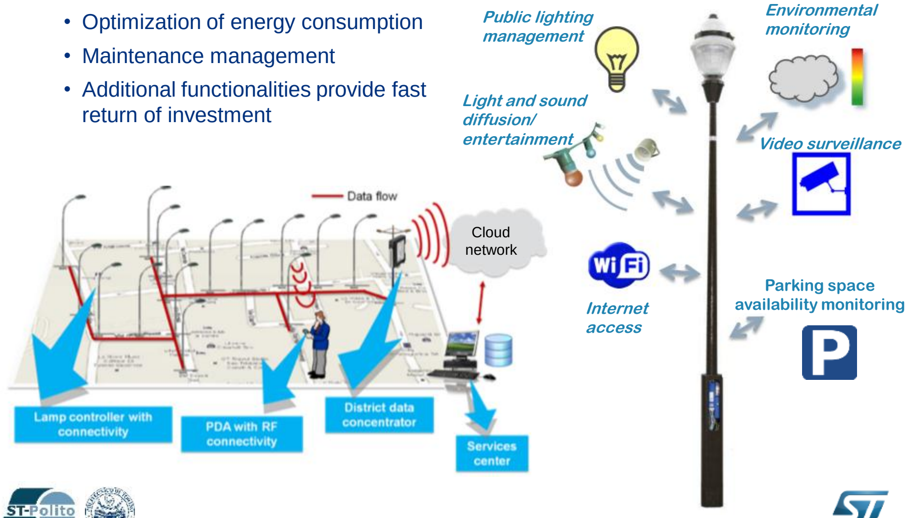
- “Moore” approach: integrate more transistors in a chip
- “More than Moore”: integrate functions in a Smart System
- Innovation in More than Moore comes in disruptive steps

Smart system integration issues

- Technology enables high level of integration and miniaturization of devices made of heterogeneous components
- Non-negligible multiphysical interactions between components (functional and non-functional)
  - Heat
  - Mechanical coupling
  - EM radiations…
- System-level design approaches are not yet available and standardized
  - Virtual prototyping
  - Design space exploration
  - Application optimization
- Modeling can reduce development time and costs (including validation, verification and testing)

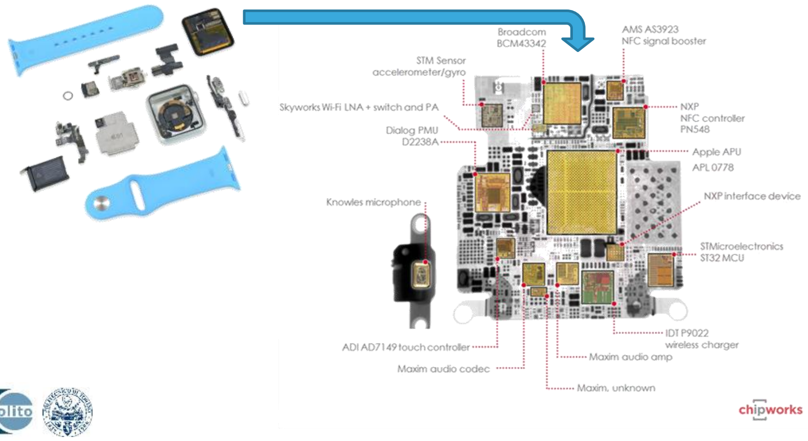
Some examples

- Smart Street lighting
  - Optimization of energy consumption
  - Maintenance management
  - Additional functionalities provide fast return of investment



Some examples [cont.]

- *Apple* Watch
  - Consumer smart system, sensor/actuator (haptics), networked, battery-operated
  - Autonomy: normal use 18 hours, "power reserve" 72 hours



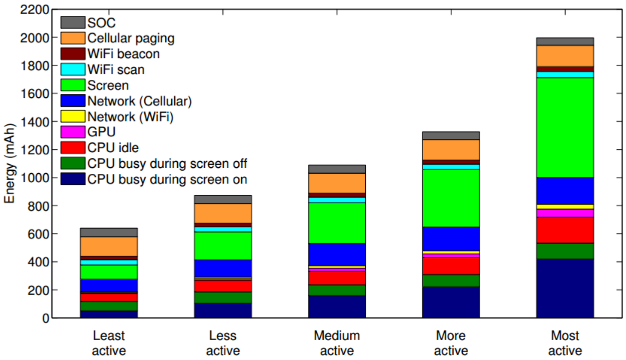
Some examples [cont.]

- Steam trap
  - Device used to discharge condensate and non condensable gases with a negligible consumption or loss of live steam
  - The simplest form of steam trap is a disc or short solid pipe nipple with a small hole drilled through it installed at the lowest point of the equipment. However, the vast majority of steam traps in current operation are of the mechanical or thermostatically operated design
- *Wireless steam trap monitor*
  - Steam trap monitoring via wireless acoustic sensors is a leading IIoT application
  - When traps fail to open, high-pressure steam leaks out, so more steam has to be produced by boilers
  - Depending on the price of steam at a facility, a single failed-open steam trap can waste $30,000 worth of steam each year
  - Usually battery-operated, lifetime ≥ 3 years

Typical energy breakdown

Smartphone

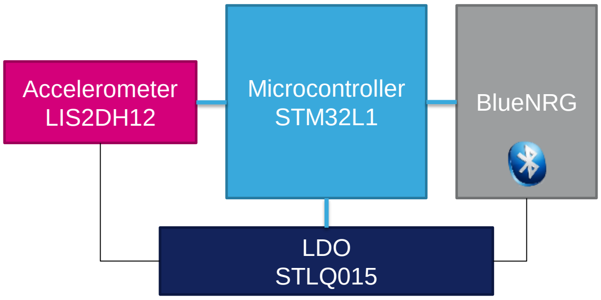
- Samsung Galaxy S3 – S4
  - Average energy drain breakdown of 1520 users (classified by activity rate)
  - X. Chen et al., “Smartphone Energy Drain in the Wild: Analysis and Implications”, ACM SIGMETRICS 2015



Typical energy breakdown [cont.]

Wireless sensor node by STMicroelectronics

- Microcontroller (STM32L1): 1μA (stop mode w/RTC), 195 μA/MHz Run mode (6mA @32MHz)
- Accelerometer (LIS2DH12): 2μA @1Hz, 11μA @50Hz, 0.5μA power-down
- Bluetooth LE network processor (BlueNRG): down to 1.7 μA with active BLE stack, 8.2 mA maximum TX current (@0 dBm, 3.0 V)
- LDO regulator (STLQ015): quiescent current ~1.4 μA, 1 nA in OFF mode



Accelerometer
LIS2DH12

Microcontroller
STM32L1

BlueNRG

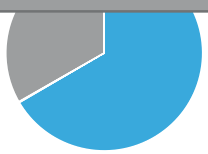
LDO
STLQ015

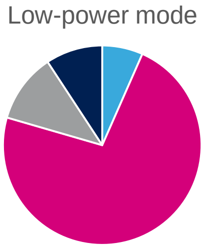
Typical energy breakdown [cont.]

Wireless sensor node by STMicroelectronics

	Normal mode (~10mA)	Low power mode (~8μA)
🟦	CPU running @32MHz	CPU in stop mode w/RTC
🟥	Accelerometer sampling @100Hz	Accelerometer sampling @50Hz,
▨	Bluetooth LE transmitting @50Hz	Low-power mode, active BLE stack
⬛	LDO on	LDO on

The system is in passive mode and is periodically woken up by clock or by external event (e.g., free fall detection)



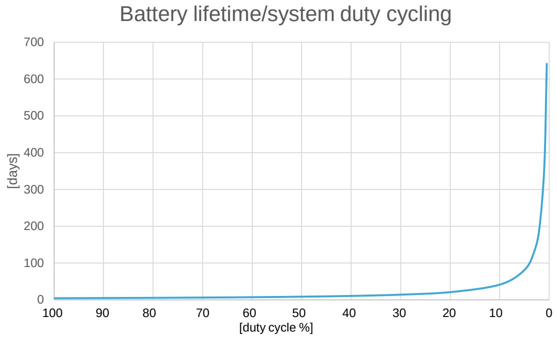


Low-power mode

Typical energy breakdown [cont.]

Wireless sensor node by STMicroelectronics

- Lifetime expectation system powered by 2xAAA batteries (1000mAh)
  - 100% active mode → 100 hours
  - 50% active → 200 hours
  - 1% active → 363 days - quite typical for IoT devices
  - (100% low-power mode → 7+ years!)



IoT devices power challenges

…and opportunities

- Evaluate system consumption taking into account mission profile
  - CPU profiling
  - Multiphysical interactions and device non-ideality (simulation and evaluation)
- Define low-power system management policies
  - Low-resource algorithms (e.g., sensor data or image processing)
  - Event-driven programming
  - Compressed sensing
  - Network optimization
  - …
- Optimize power consumption of components
  - Technology scaling
  - Architecture customization
  - DFVS, power gating, asynchronous devices, …
- Optimize energy storage, conversion and harvesting

Principles of Power Optimization: Computation, Communication and Power Delivery

Massimo Poncino, Politecnico di Torino – Torino, Italy

1st International Forum on Research and Technologies for Society and Industry
Torino, Italy – September 16-18 2015

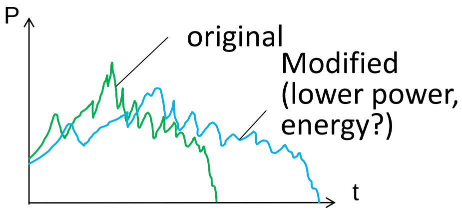
Outline

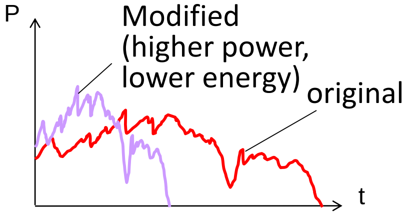
- Power models
  - What affects power consumption?
- Basic principles
- Solutions for energy efficiency
  - Computation
  - Communication
  - Power delivery

Power models

Power vs. Energy

- Often a misconception about these two metrics…
- *Power* is the **rate** at which energy is consumed
- *Energy* is the **amount** of power consumed
- You can use a transformation that reduces power but not energy or viceversa.





Power models

- An electronic system consumes power
  - When "operating"
    - This is the "true", intrinsic power consumption
    - **Active power**
  - When "idle"
    - Sort of "parasitic" power
    - Due to non-idealities in the technologies this component is non negligible!!!
    - **Standby or leakage power**

Power models (2)

- **Active** power
  - Dominated by "switching" component

$$P_{switching} = \alpha \cdot C \cdot V_{DD}^2 \cdot f_{clk}$$

  - α = Switching activity factor
  - C = switched capacitance.
  - V_{dd} = supply voltage
  - f_{CLK} = Clock frequency.

NOTE:
Assumes a specific (CMOS) technology of digital circuits…

Power models (3)

- **Standby** power

$$P_{stdby} = V_{DD} \cdot I_{stdby}$$

  - I_{stdby} = sum of all standby currents
- Emerging issue in deeply scaled technologies!

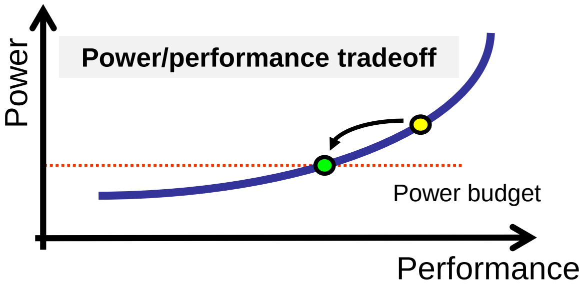
Standby power vs. energy

- **MYTH: Leakage is an issue only for deeply scaled technologies**
- **TRUTH: Don't underestimate idle power even for older technology nodes!**
  - Duty cycle of operations is essential when evaluating energy!!!!
- Example: 1/1000 duty cycle (idle=99.9%)
  - **P_{active} = 100mW**
  - **P_{idle}=100µW (1/1000)**

$$E_{active} = E_{idle}!$$

Power/performance tradeoff

- Invariably, **power and performance are contrasting metrics**
  - Lower voltage ➡ lower signal strength
  - Lower frequency ➡ lower speed
  - Lower activity ➡ fewer operations/unit time
  - Lower capacitance ➡ smaller (and slower) devices



Basic principles of energy-efficient design

1) Act on the variables that define power

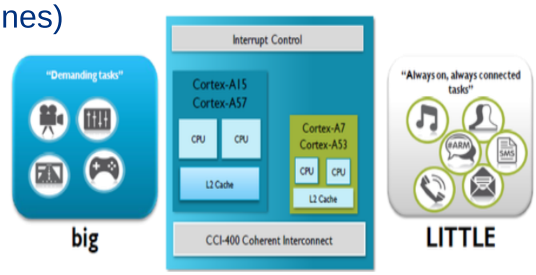
- **Speed** (voltage, frequency), **functionality** (activity), **complexity** (capacitance), **technology** (device parameters)
- Many embodiments at different abstraction levels
- Some of them can impact other metrics (typically, performance)
- Example
  - Speed scaling, i.e., voltage/frequency scaling to reduce power during non-critical computations (low workload)

2) Exploit Idleness

- i.e., **avoid useless work** ➔ turn off components
- Choice of actual implementation of standby dictated by technology (leakage, tech node, noise…)
  - Partially overlaps with 1?
- Example:
  - standby of idle components

3) Exploit common case

- Power and energy are cumulative (average) quantities
- Workload is not constant ➔ **match workload to resources**
  - Use what you need
  - High-workload to heavier resources, small workloads to lighter ones
- Example: ARM Big/Little architecture (e.g., in Samsung’s smartphones)



Solutions for energy efficiency

Improving the Energy Efficiency

- In **computation**
  - **Dynamic power management**
- In **communication**
  - **Energy-efficient data transfer**
- In the **power delivery**
  - **Power generation, conversion and storage**

Dynamic Power Management

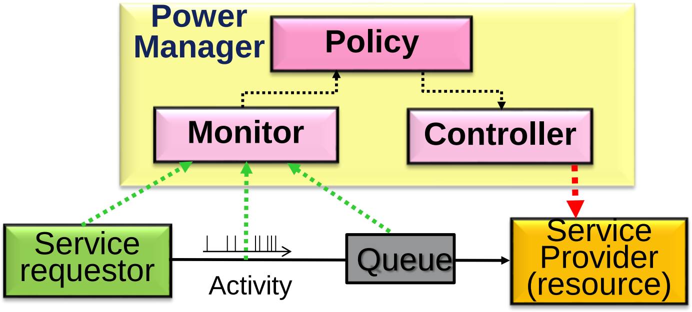
- Systems and components are designed to deliver **peak performance**
  - … but they do not need peak performance all the times (actually, they seldom need it…)
- The general idea of DPM is to reduce power by turning **resources** into a **low-power "state"** *when under-utilized*
  - **What resources ?**
  - **What low-power state?**

Dynamic Power Management

- **Resources...**
  - Processors, memories, devices (e.g., disks, displays,...)
  - **MUST BE POWER MANAGEABLE**
- **Low-power “state”....**
  - Various possible implementations depending also on
    - The type of resource
    - The available technology
  - Examples: clock gating/throttling, voltage scaling

DPM - overview

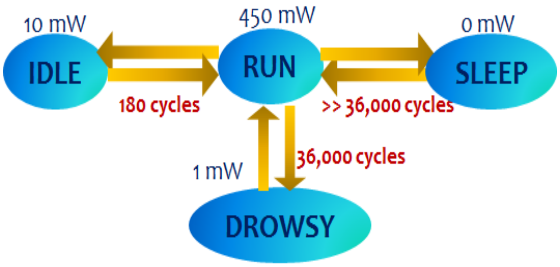
- Model:



- Power manager (PM):
  - monitors requestor’s activity and sets state of provider according to some **policy** (implemented inside the PM)

Model of the Resource

- **Power state machine**
  - **States = operative modes**
  - **Transitions have power/perfomance costs**
- **Example (Intel Xscale core)**



Model of the workload

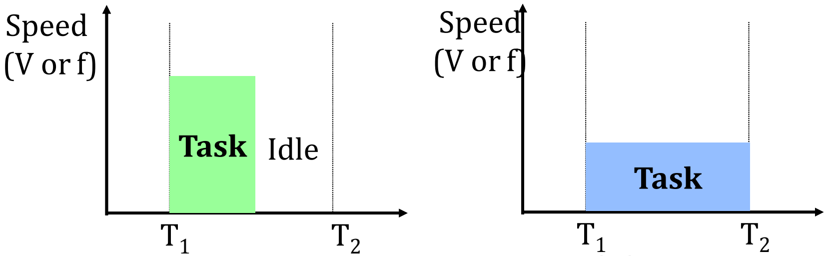
- Various options with variable degree of accuracy
  - **Waveforms**
    - power vs. time
  - **Statistical models**
    - **distribution of idle/active times**
  - **Stochastic models**
    - Random processes/Markov models
- **PM and its applicability is a property of a system-workload pair**

DPM: Embodiments

- **Duty-cycling**
  - Resource can execute only in one way (one “ON” state)
  - Possibly many “OFF” states
- **Speed scaling**
  - **Resource can execute with different power/performance points (multiple “ON” states)**
  - Possibly many “OFF” states

Duty-cycling vs. Speed scaling

- Speed scaling is always more effective

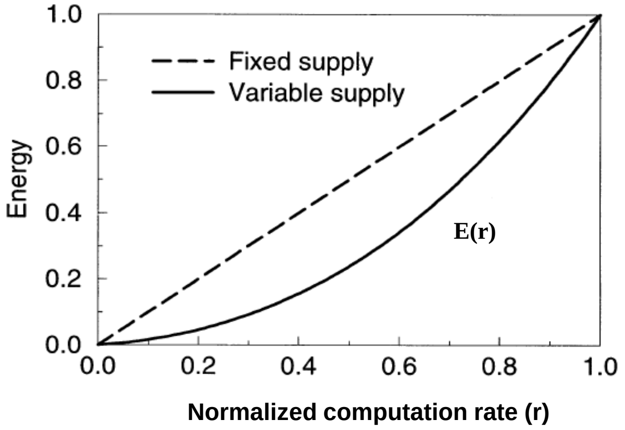


Same work, less energy

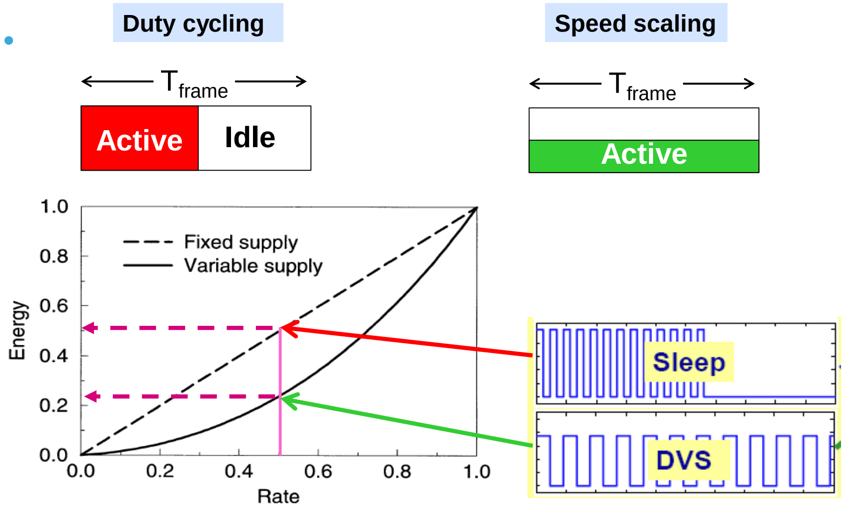
Why?

Duty-cycling vs. Speed scaling

- Reason lies in the **convex relation between energy and speed**

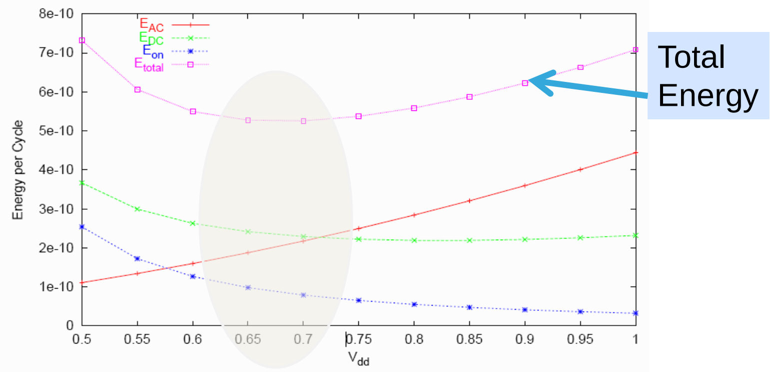


Energy vs. processing rate: example



But… beware of standby energy

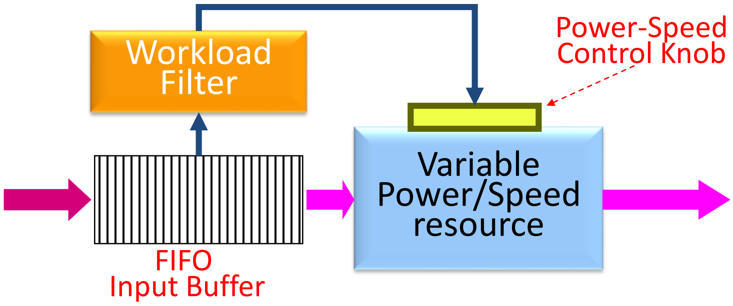
- If we lower speed (V/f) too much, standby energy becomes dominating
  - There exists an optimal speed!



Total Energy

DPM: practical implementations

- Typically in SW in the OS



Workload Filter

Power-Speed Control Knob

Variable Power/Speed resource

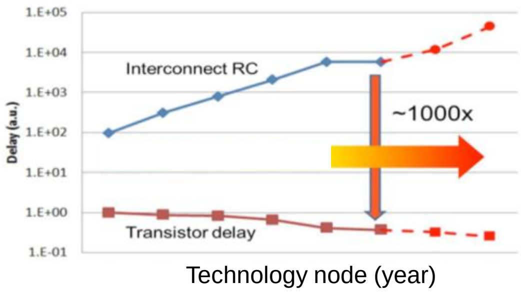
FIFO Input Buffer

Energy efficiency in communication

- Wired communication
- Wireless communication

Wired communication

- Whatever the scales (microns to meters) wires are used to transfer data
  - Wires are **not ideal** connections
    - Resistive, capacitive and inductive effects
    - Capacitve power component is dominant for typical interconnects
  - Why do we care?



Energy Efficient Wired Communication

- Power consumed when a transition over a wire occurs (capacitive switching)
- Example:

time

	# of bit toggles
00101010	
00111011	2
11010100	7
11110100	1
00001101	6
01110110	6
00010001	5
10000100	4

➡ **Reduce number of transitions**

Energy Efficient Wired Communication

- An encoding problem!
- An example: the **bus-invert code**
  - *Transmit pattern only if # of transitions ≤n/2, otherwise transmit the complement*
  - *Use 1 extra bus line to signal at the receiving end whether pattern is inverted or not!*

00101010		00101010	0	
00111011	2	00111011	0	2
11010100	7	00101011	1	2
11110100	1	00001011	1	1
00001101	6	00001101	0	3
01110110	6	10001001	1	3
00010001	5	00010001	0	4
10000100	4	10000100	0	4
Original: **31 Transitions**		**Bus-invert encoding:** **19 Transitions**		

Wireless communication

- In IoT everything will be connected
- Various types of “networks” available
  - Wifi, 3G, ad-hoc, short-range, …
  - And many devices support multiple types of connectivity !
- Important to have a qualitative idea of the suitable protocols (bandwidth, etc.) and the relative power demand…
- **…. Considering also that energy required for transmission typically exceed that for computation….**

Computation vs. Communication Energy Cost: Example

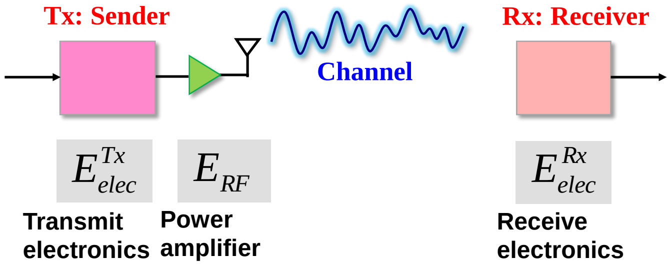
- Energy/bit >> Energy/op even for short ranges!

WSN-class Node	Transmit	**720 nJ/bit**	Processor	**4 nJ/op**
	Receive	**110 nJ/bit**	**~ 200 ops/bit**	

Microserver-class Node	Transmit	**6600 nJ/bit**	Processor	**1.6 nJ/op**
	Receive	**3300 nJ/bit**	**~ 6000 ops/bit**	

Computation vs. Communication Energy

- Communication always dominates, but communication energy is not just “communication”…



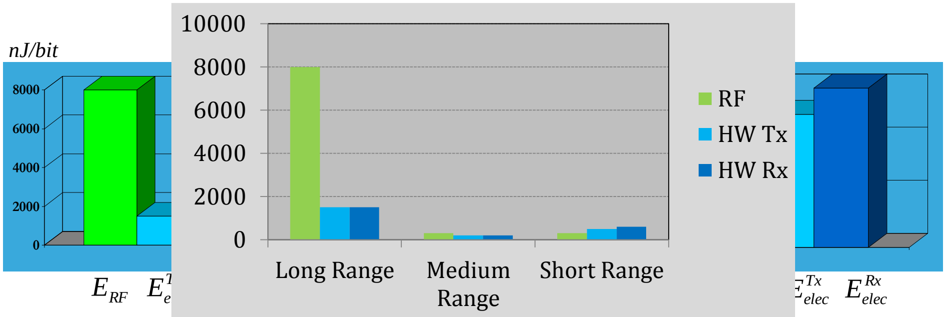
E_{elec}^{Tx} **Transmit electronics**

E_{RF} **Power amplifier**

E_{elec}^{Rx} **Receive electronics**

- **Range matters!!!**

Examples

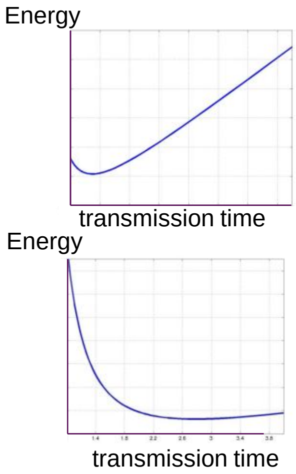


~ 1 km (GSM) **~ 50 m (WLAN)** **~ 10 m (ad -hoc)**

- RF energy increases with transmission range

Computation vs. Communication Energy tradeoff

- **Short-range links**
  - Shutdown based
  - Turn-off sender and receiver
- **Long-range links**
  - Scaling based
  - Slow down transmissions

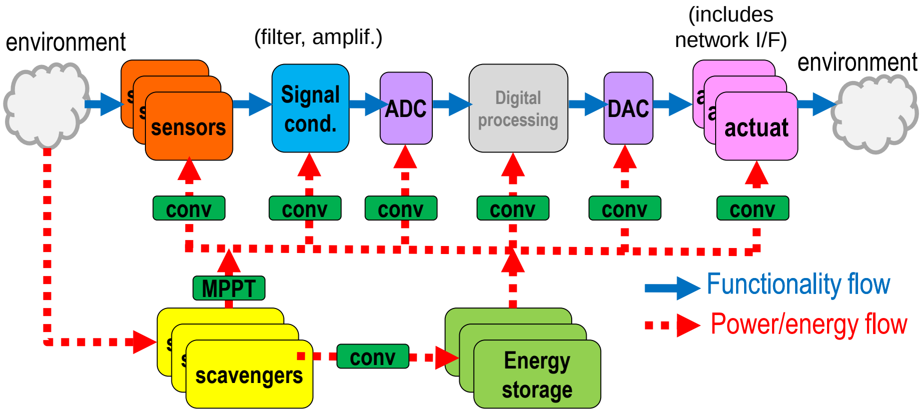


Energy efficient power/energy delivery

- Energy storage
- Power conversion
- Power generation

Functional and power flows in an autonomous device

- Not just functionality…



environment

(filter, amplif.)

(includes network I/F)

environment

sensors

Signal cond.

ADC

Digital processing

DAC

actuat

conv

conv

conv

conv

conv

conv

MPPT

scavengers

conv

Energy storage

Functionality flow

Power/energy flow

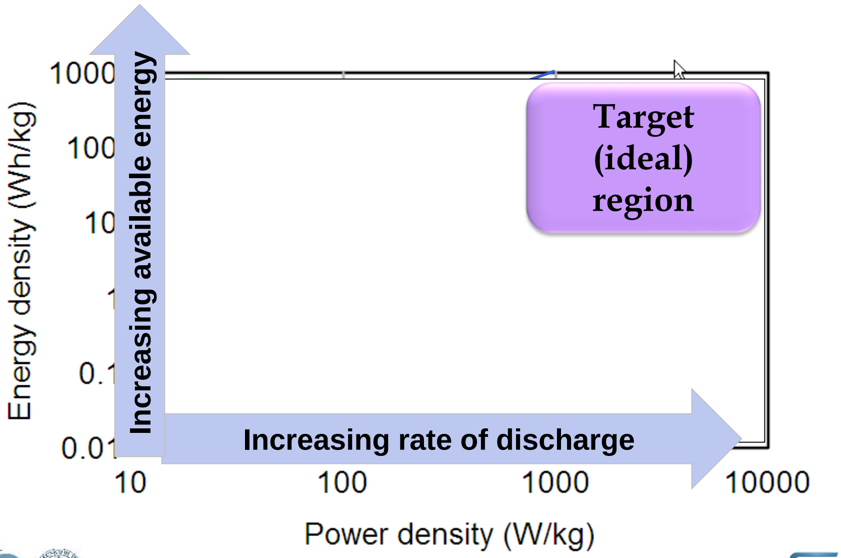
Energy storage

- Relevant issues:
  - **Which type of energy storage device (ESD)? (storage selection)**
    - A battery is the obvious choice, but is it the best?
  - **How much storage? (storage sizing)**
    - Apparently simple but ESD are not exactly ideal storage elements….

Storage Selection

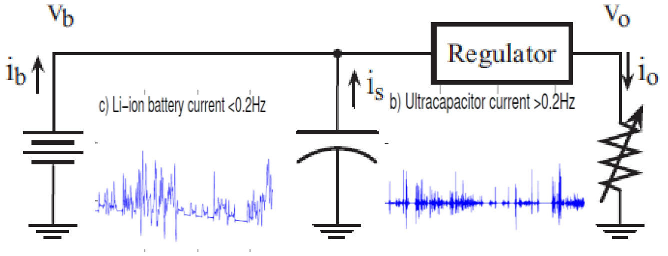
- Qualitative figures of merit of ESDs:
  - Amount of stored energy (per unit weight/volume)
    - **Energy density!**
  - How fast this energy can be drawn (per unit weight/volume)
    - **Power density!**
- **Ideally, we want both to be as large as possible!**
- ESDs are typically compared with a **Ragone plot**, which draws energy density vs. power density

Ragone plot



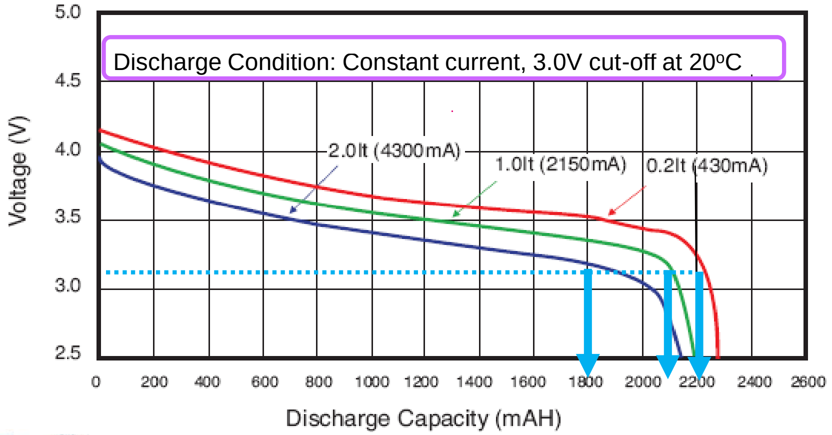
Hybrid ESD

- Similar situation as in digital memories…(the **nemory hierarchy)**
- In ES, rather a hybrid ESD than a hierarchy
  - Example:
    - Use battery for sustained (high ED) low freq loads (low PD)
    - Use a supercapacitor for short (low ED) high freq loads (high PD)

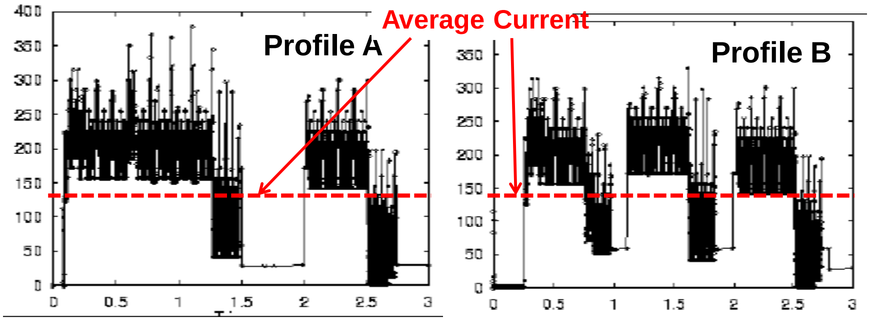


Non-ideal ESDs: Batteries (1)

The amount of energy a battery can provide depends on the current drawn from the battery itself



Non-ideal ESDs: Batteries (2)



[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

Storage sizing

- A simple calculation like

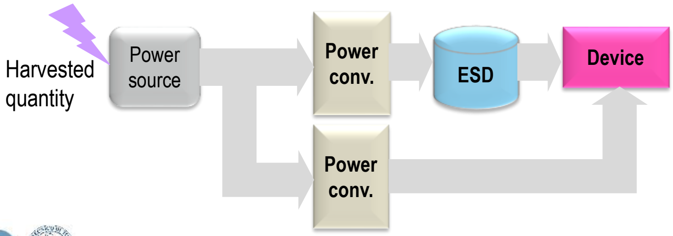
Capacity = *Average current* X *Desired lifetime*

May strongly underestimate the requirement!!!
  - Average current is not representative…
  - Rated capacity effect is not accounted for

- **Need model-based simulation to do a correct assessment!!!**

Power generation: scavenging

- Many scavenging mechanisms involving different x-to-electrical transduction
- Key parameter obviously **output power**
- **Issues:**
  - Can my power source directly power my device?
  - If not, how efficiently can it be "adapted" to an ESD?
- Scenario:

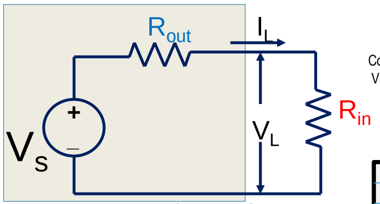


Extracting power

- Almost invariably, storage voltage (e.g., battery, load) is **not** the ideal value for harvesting the maximum energy available….
  - Need to find an optimal point **(maximum power point – MPP)**
- Harvester has low efficiency if storage device is not **impedance-matched** to source !!!
  - Source impedance is variable
- Optimal **matching** (MPP) circuits/algos have to track the source
  - Can consume too much energy themselves…
- MPP tracking is an essential component
  - Will discuss technicalities with reference to one specific type of scavenger (solar cell)

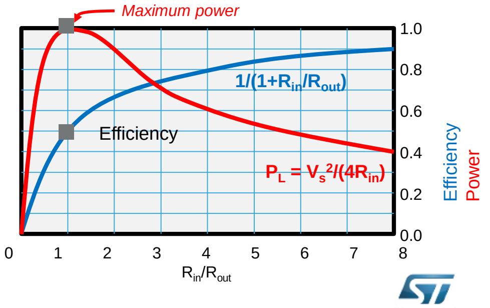
Impedance matching and maximum power: recap

- **Maximum power theorem:**
  - Power transferred to the load is maximum when output and load resistance are the same ($R_{in}=R_{out}$)



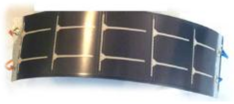
  - Power transferred only 50% (efficiency)

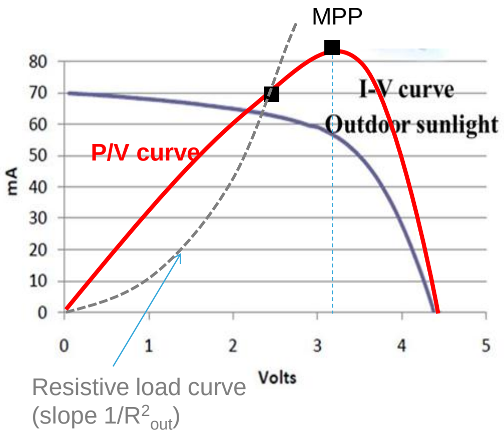
Courtesy:
VD Agrawal



MPP: an example

- Source: solar microcell





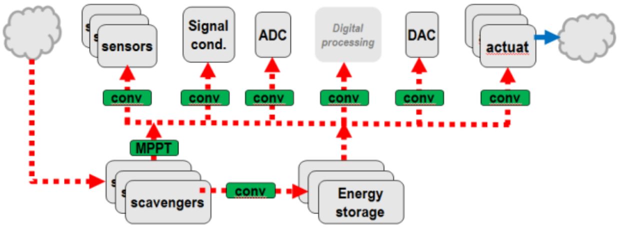
- Operating Voltage: 3
- Wattage: .15
- Current: 50
- Typ Voc: 4.1V
- Typ Isc: 60mA
- Typ Output @ AM 1.5: 55mA @ 3V
- Width (mm): 36.5 mm
- Length (mm): 114 mm
- Width (in): 1.45 in
- Length (in): 4.49 in
- Weight (kg): 1.2 g
- Weight (lb): .04 oz
- Total Thickness (mm): 0.22mm

Tracking the MPP

- Done by ad-hoc HW call MPP tracker (MPPT)
- Basically a switching DC-DC converter that tries to match R_{in} and R_{out} by adapting the working point
    - **Double source of inefficiency !**
        - **Converter efficiency…**
        - **Failure to perfectly track MPPT**

Energy conversion

- Why worry about conversion?
- **There is conversions between any two different voltage levels!**



- **Conversion involves (large) energy losses**
- **Converters cannot be (easily) put in standby**

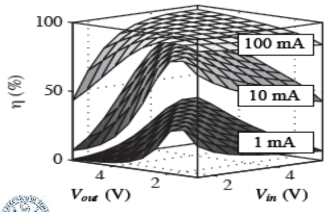
Characterization: efficiency

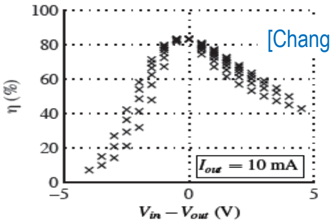
- The conversion process is not “perfect”
- Efficiency = output power/input power

$$\eta = \frac{P_{out}}{P_{in}} = \frac{P_{in} - P_{losses}}{P_{in}} = 1 - \frac{P_{losses}}{P_{in}}$$

$$P_{in} = V_{in} I_{in}$$
$$P_{out} = V_{out} I_{out}$$

Regulator type	Efficiency affected by
Switching	Output current, ΔV, input and output voltages
Linear	ΔV





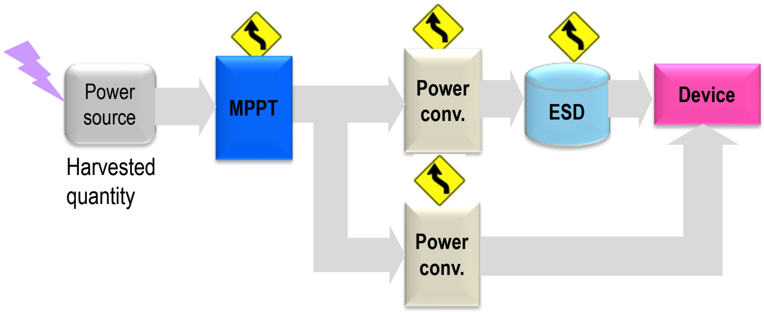
[Chang et al. 2011]

Reducing power wasted in conversion

- Converter “**binning**”
  - Group similar voltage and share conversion domains
  - Tradeoff between # of regulators and efficiency
    - *Many small losses vs. few larger ones?*
- Converter **gating**
  - Depending on architecture reduce quiescent current (circuit-level design problem) [Uzun and Kose, 2014]
- Incorporate **conversion awareness** in DPM
  - Converter-aware DPM [Choi et al., 2007]

Conclusion: Power is wasted in every step…

- And not just in the device
  - Device-side power optimization does not suffice
- **Losses, losses, losses** (and non-idealities)
  - Can exceed what you save in the device



Energy optimization & System-level modeling – Case studies

Michelangelo Grosso, ST-Polito s.c.ar.l.

1st International Forum on Research and Technologies for Society and Industry
Torino, Italy – September 16-18 2015

Energy optimization of IoT devices

- Good practices for low-power design:

1. Use low-power components
2. Use low-power modes
(and turn devices off for most of the time)
3. Evaluate power consumption and reduce energy waste
   - Optimization of network management
   - Definition of application management policies and scheduling
   - Identification of energy-optimal data analysis algorithms
4. Evaluate and optimize performance/power trade-off

- Extremely wide range of implementation possibilities!

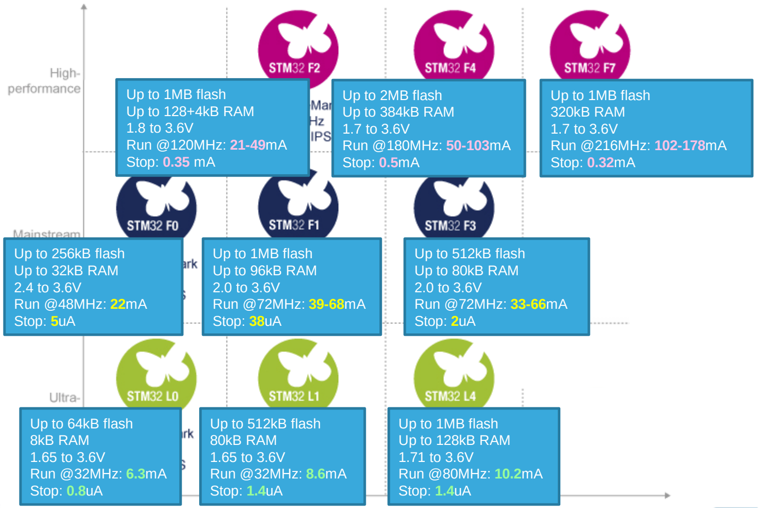
---

1. Microcontroller selection

	Cortex-M0 / -M0+	Cortex-M3	Cortex-M4	Cortex-M7
High-performance		STM32 F2 398 CoreMark 120 MHz 150 DMIPS	STM32 F4 608 CoreMark 180 MHz 225 DMIPS	STM32 F7 1 082 CoreMark 216 MHz 462 DMIPS
Mainstream	STM32 F0 106 CoreMark 48 MHz 38 DMIPS	STM32 F1 177 CoreMark 72 MHz 61 DMIPS	STM32 F3 245 CoreMark* 72 MHz 90 DMIPS*	
Ultra-low-power	STM32 L0 75 CoreMark 32 MHz 26 DMIPS	STM32 L1 93 CoreMark 32 MHz 33 DMIPS	STM32 L4 273 CoreMark 80 MHz 100 DMIPS	

* from CCM-SRAM

1. Microcontroller selection



STM32 F2:
Up to 1MB flash
Up to 128+4kB RAM
1.8 to 3.6V
Run @120MHz: 21-49mA
Stop: 0.35 mA

STM32 F4:
Up to 2MB flash
Up to 384kB RAM
1.7 to 3.6V
Run @180MHz: 50-103mA
Stop: 0.5mA

STM32 F7:
Up to 1MB flash
320kB RAM
1.7 to 3.6V
Run @216MHz: 102-178mA
Stop: 0.32mA

STM32 F0:
Up to 256kB flash
Up to 32kB RAM
2.4 to 3.6V
Run @48MHz: 22mA
Stop: 5uA

STM32 F1:
Up to 1MB flash
Up to 96kB RAM
2.0 to 3.6V
Run @72MHz: 39-68mA
Stop: 38uA

STM32 F3:
Up to 512kB flash
Up to 80kB RAM
2.0 to 3.6V
Run @72MHz: 33-66mA
Stop: 2uA

STM32 L0:
Up to 64kB flash
8kB RAM
1.65 to 3.6V
Run @32MHz: 6.3mA
Stop: 0.8uA

STM32 L1:
Up to 512kB flash
80kB RAM
1.65 to 3.6V
Run @32MHz: 8.6mA
Stop: 1.4uA

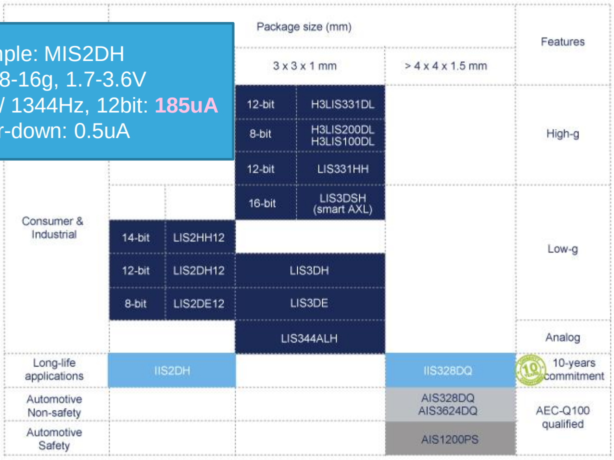
STM32 L4:
Up to 1MB flash
Up to 128kB RAM
1.71 to 3.6V
Run @80MHz: 10.2mA
Stop: 1.4uA

* from CCM-SRAM

Peripheral selection

- Sensor (accelerometer)

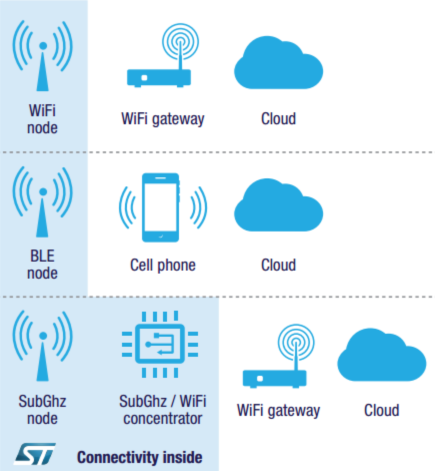
Example: MIS2DH
FS 2-4-8-16g, 1.7-3.6V
1Hz, 8bit: 2uA / 1344Hz, 12bit: 185uA
Power-down: 0.5uA



			Package size (mm)			Features
			3 x 3 x 1 mm		> 4 x 4 x 1.5 mm	
Consumer & Industrial			12-bit	H3LIS331DL		High-g
			8-bit	H3LIS200DL H3LIS100DL		
			12-bit	LIS331HH		
			16-bit	LIS3DSH (smart AXL)		Low-g
	14-bit	LIS2HH12				
	12-bit	LIS2DH12	LIS3DH			
	8-bit	LIS2DE12	LIS3DE			
			LIS344ALH			Analog
Long-life applications	IIS2DH				IIS328DQ	10-years commitment
Automotive Non-safety					AIS328DQ AIS3624DQ	AEC-Q100 qualified
Automotive Safety					AIS1200PS	

Peripheral selection [cont.]

- RF solutions

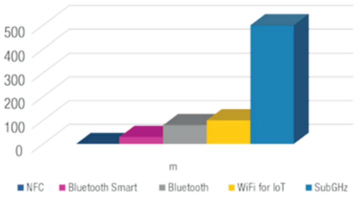


Advantages	Disadvantages
• Existing infrastructure • Customer familiarity	• Power consumption • Limited range • Large stack • Costly radio • Difficult set-up
• Connection to smartphone • Low power • Easy set-up	• Limited range • Royalties
• Light stack • Multi radio support • Multi P/S support • Good range • Very low power	• Additional concentrator required

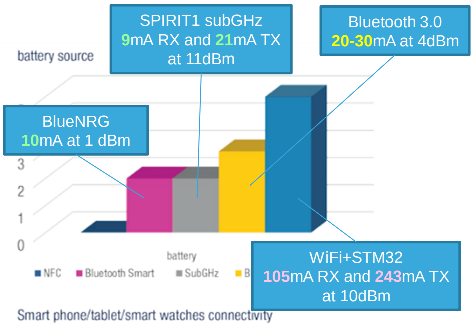
Peripheral selection [cont.]

- RF solutions performance

typical distance range [m]



battery source



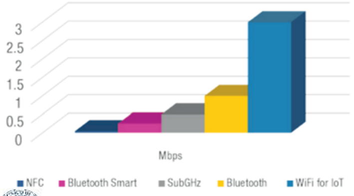
SPIRIT1 subGHz
9mA RX and 21mA TX
at 11dBm

Bluetooth 3.0
20-30mA at 4dBm

BlueNRG
10mA at 1 dBm

WiFi+STM32
105mA RX and 243mA TX
at 10dBm

Maximum throughput [Mbps]

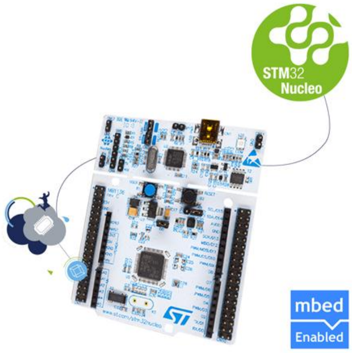


Smart phone/tablet/smart watches connectivity

YES	NO
NFC **Bluetooth Smart** **Bluetooth** **Wi-Fi for IoT**	SubGHz

2. Low-power modes

- An example: **STM32L152RE microcontroller**
  - 32MHz CPU, 512kB flash, 80kB RAM, 1.65-3.6V
  - Peripherals: 11x timers, 2x OPAMP, 2x COMP, 12-bit ADC and 2-ch. 12-bit DAC, 1x USB 2.0, 5x USART, 3x SPI, 2x I2C, CRC computation, LCD driver
- Run mode with Dynamic Voltage and Frequency Scaling (DVFS)
  - Range 1: V_{DD} 1.71-3.6V, up to 32MHz
  - Range 2: V_{DD} full range, up to 16MHz
  - Range 3: V_{DD} full range, up to 4MHz (internal oscillator)
- Sleep mode
  - CPU off, peripherals ON
- Low-power run mode
  - Internal clock up to 131kHz, regulator in low-power mode
- Low-power sleep mode
- Stop mode with/without RTC
  - Registers and RAM retention, peripherals off
- Standby mode with/without RTC

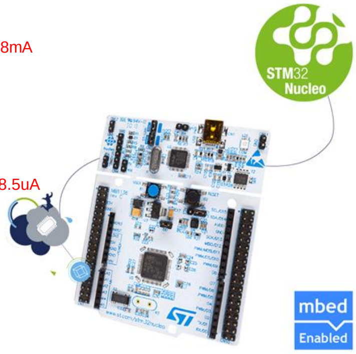


mbed Enabled

life.augmented

2. Low-power modes

- An example: **STM32L152RE microcontroller**
  - 32MHz CPU, 512kB flash, 80kB RAM, 1.65-3.6V
  - Peripherals: 11x timers, 2x OPAMP, 2x COMP, 12-bit ADC and 2-ch. 12-bit DAC, 1x USB 2.0, 5x USART, 3x SPI, 2x I2C, CRC computation, LCD driver
- Run mode with Dynamic Voltage and Frequency Scaling (DVFS)
  - Range 1: V_{DD} 1.71-3.6V, up to 32MHz 2.2-8.6mA
  - Range 2: V_{DD} full range, up to 16MHz 0.98-3.6mA
  - Range 3: V_{DD} full range, up to 4MHz (internal oscillator) 0.23-0.78mA
- Sleep mode
  - CPU off, peripherals ON 0.063-1.750mA
- Low-power run mode
  - Internal clock up to 131kHz, regulator in low-power mode 26.0-48.5uA
- Low-power sleep mode 5.5-21.5uA
- Stop mode with/without RTC 1.65-0.56uA
  - Registers and RAM retention, peripherals off
- Standby mode with/without RTC 0.97-0.29uA

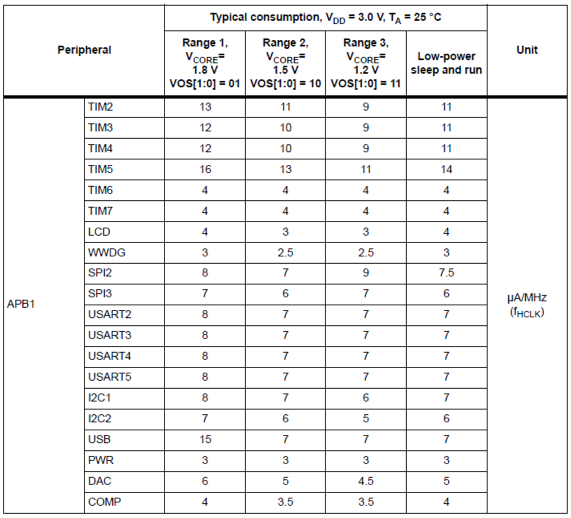


mbed Enabled

life.augmented

Low-power modes [cont.]

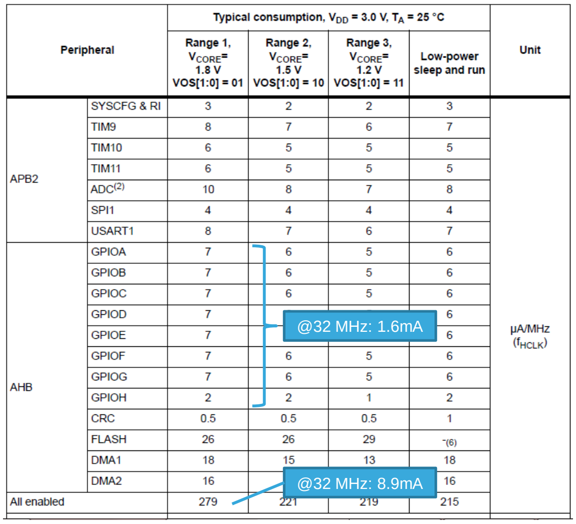
- STM32L152 peripheral power consumption



Peripheral		Typical consumption, V_{DD} = 3.0 V, T_A = 25 °C				Unit
		Range 1, V_{CORE}= 1.8 V VOS[1:0] = 01	Range 2, V_{CORE}= 1.5 V VOS[1:0] = 10	Range 3, V_{CORE}= 1.2 V VOS[1:0] = 11	Low-power sleep and run	
APB1	TIM2	13	11	9	11	µA/MHz (f_{HCLK})
	TIM3	12	10	9	11	
	TIM4	12	10	9	11	
	TIM5	16	13	11	14	
	TIM6	4	4	4	4	
	TIM7	4	4	4	4	
	LCD	4	3	3	4	
	WWDG	3	2.5	2.5	3	
	SPI2	8	7	9	7.5	
	SPI3	7	6	7	6	
	USART2	8	7	7	7	
	USART3	8	7	7	7	
	USART4	8	7	7	7	
	USART5	8	7	7	7	
	I2C1	8	7	6	7	
	I2C2	7	6	5	6	
	USB	15	7	7	7	
	PWR	3	3	3	3	
	DAC	6	5	4.5	5	
	COMP	4	3.5	3.5	4	

Low-power modes [cont.]

- STM32L152 peripheral power consumption [cont.]



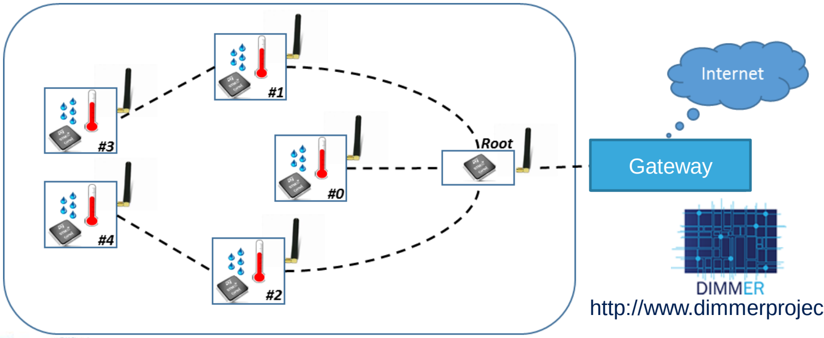
Peripheral		Typical consumption, V_{DD} = 3.0 V, T_A = 25 °C				Unit
		Range 1, V_{CORE}= 1.8 V VOS[1:0] = 01	Range 2, V_{CORE}= 1.5 V VOS[1:0] = 10	Range 3, V_{CORE}= 1.2 V VOS[1:0] = 11	Low-power sleep and run	
APB2	SYSCFG & RI	3	2	2	3	µA/MHz (f_{HCLK})
	TIM9	8	7	6	7	
	TIM10	6	5	5	5	
	TIM11	6	5	5	5	
	ADC(2)	10	8	7	8	
	SPI1	4	4	4	4	
	USART1	8	7	6	7	
AHB	GPIOA	7	6	5	6	
	GPIOB	7	6	5	6	
	GPIOC	7	6	5	6	
	GPIOD	7			6	
	GPIOE	7			6	
	GPIOF	7	6	5	6	
	GPIOG	7	6	5	6	
	GPIOH	2	2	1	2	
	CRC	0.5	0.5	0.5	1	
	FLASH	26	26	29	-(6)	
	DMA1	18	15	13	18	
	DMA2	16			16	
All enabled		279	221	219	215	

@32 MHz: 1.6mA

@32 MHz: 8.9mA

3. Energy evaluation and optimization

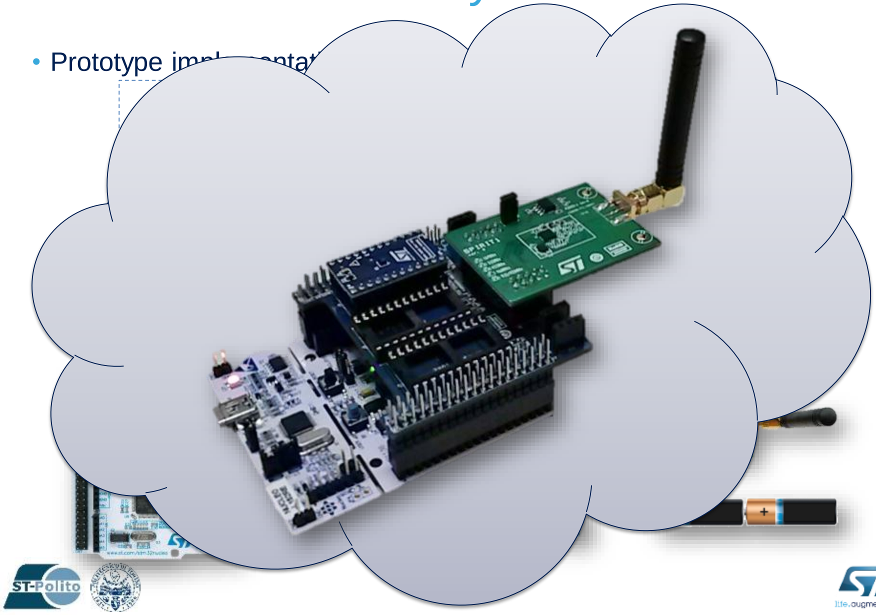
- Case study: Indoor monitoring of temperature and humidity
  - Autonomous, self-configurable and self-repairing multi-hop (mesh) network
  - Adaptable to residential and commercial buildings
  - Extended battery lifetime
- Application supported by the EU-funded FP7 Project DIMMER



http://www.dimmerproject.eu/

System architecture

- Prototype imp ntat



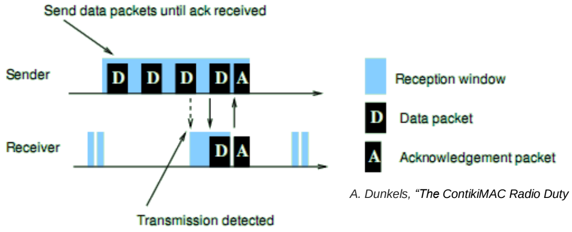
Network configuration

- Contiki open source operating system
  - IPv4 and IPv6, 6lowpan, RPL, CoAP
  - Low-power mode available (ContikiMAC)
  - Live community of users
- Thingsquare Mist open-source porting for STM32 and SPIRIT1
- Customized application layer
  - First, the network is built (handshaking mechanism between nodes and gateway)
  - Then, each node starts periodically transmitting data to gateway (OS services handle routing and packet forwarding)

Application	**Sensor node application**		
RTOS	**Contiki OS / Thingsquare**		
Platform	**NUCLEO-L152RE drivers**	**IKR002V4D drivers**	**MKI141V2 drivers**
Devices	**STM32L libs**	**SPIRIT1 libs**	**HTS221 libs**

Network operation

- ContikiMAC principle
  - Radio duty cycling to reduce power consumption
  - Useful when network needs to be “always active”
  - Small amount of latency introduced in communication

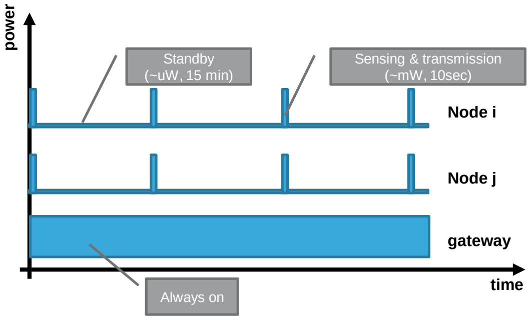


A. Dunkels, *“The ContikiMAC Radio Duty Cycling Protocol”*

- The specific application has relaxed constraints
  - Temperature and humidity change slowly in time
  - **This can be used to further reduce power!**

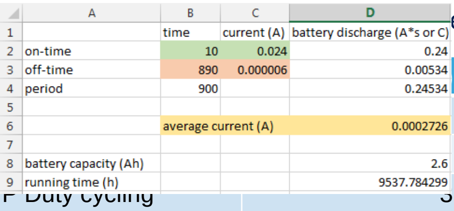
Low-power network operation

- STP system-level duty cycling
  - Periodic intermittent network behavior
  - Requires synchronization between nodes (periodic messages broadcast by gateway)
  - Standby mode: microcontroller stopped w/RTC and data retention, SPIRIT1 in standby, HTS221 powered-down



Power consumption and lifetime estimation

- *Always on*:
  - 10mA microcontroller + 12mA SPIRIT1 + 2μA HTS221= 24mA average
- *ContikiMAC* (best case, 20% radio duty cycle):
  - 10mA microcontroller + 2.4mA SPIRIT1 +2μA HTS221 = 12.4mA average
- *STP Duty cycling* (10s on, 890s off)
  - On-time 24mA, off-time 6μA ➔ **0.25mA average**
- Lifetime (syste eries):



	A	B	C	D
1		time	current (A)	battery discharge (A*s or C)
2	on-time	10	0.024	0.24
3	off-time	890	0.000006	0.00534
4	period	900		0.24534
5				
6		average current (A)		0.0002726
7				
8	battery capacity (Ah)			2.6
9	running time (h)			9537.784299

Alw	5
Co	9
STP Duty cycling	397
STP DC + ContikiMAC	753

4. Energy/performance evaluation

- System-level modeling enables the **evaluation** of

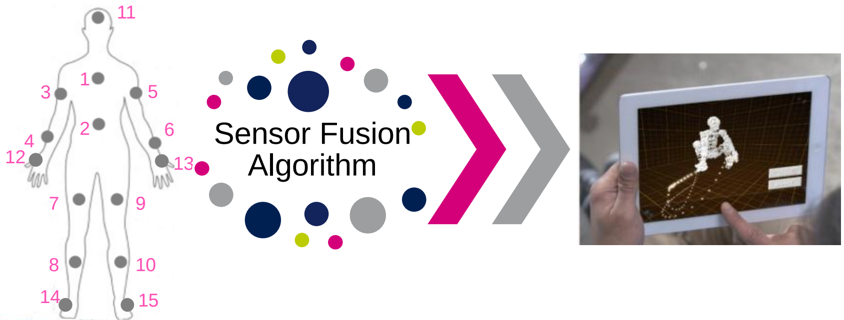
Performance

Energy consumption

- Energy optimization can thus be obtained by finding the most suitable **trade-off** for the target application
- *How much of a performance downgrade is associated with a power reduction strategy?*

Wearable sensing equipment for reliable drift-free limb tracking

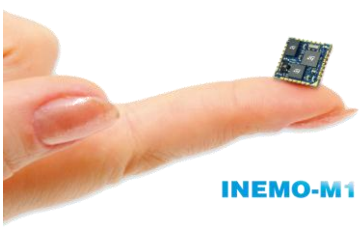
- Inertial body motion reconstruction (IBMR)
- Set of sensor nodes placed on limbs
  - Based on the ST iNEMO-M1 system-on-board
- Application supported by the EU-funded FP7 Project SMAC

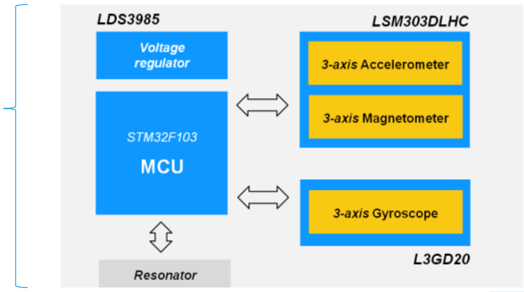


Sensor Fusion Algorithm

Sensor node

- The INEMO-M1 is the first 9-axis motion sensing system on board (SoB) of the iNEMO® module family
- It integrates multiple ST MEMS sensors (a 6-axis digital e-compass, a 3-axis digital gyroscope) with a powerful 32-bit computational core
- Power consumption ~ 150mW @100Hz AHRS estimation



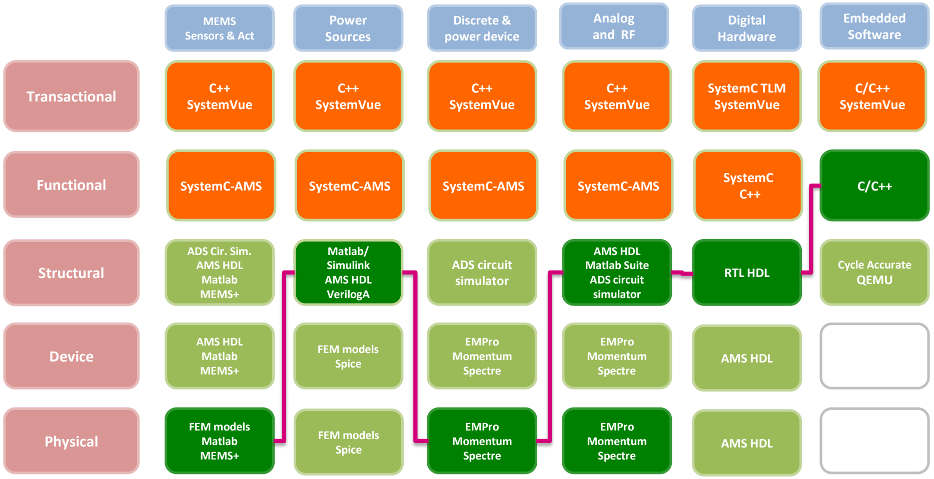


Goals of system-level modeling

- Reducing the costs of application-level performance evaluation
  - Avoid the use of camera-based motion capture systems (e.g., Vicon), which are expensive and need fixed laboratory setup
  - Anticipate application development for time-to-market reduction (no prototype needed)
- Optimizing application algorithms
  - Sensor fusion
  - Disturbance minimization in specific environment (e.g., magnetic field, temperature)
- Estimating and optimizing the trade-off between system performance and energy consumption

www.fp7-smac.org

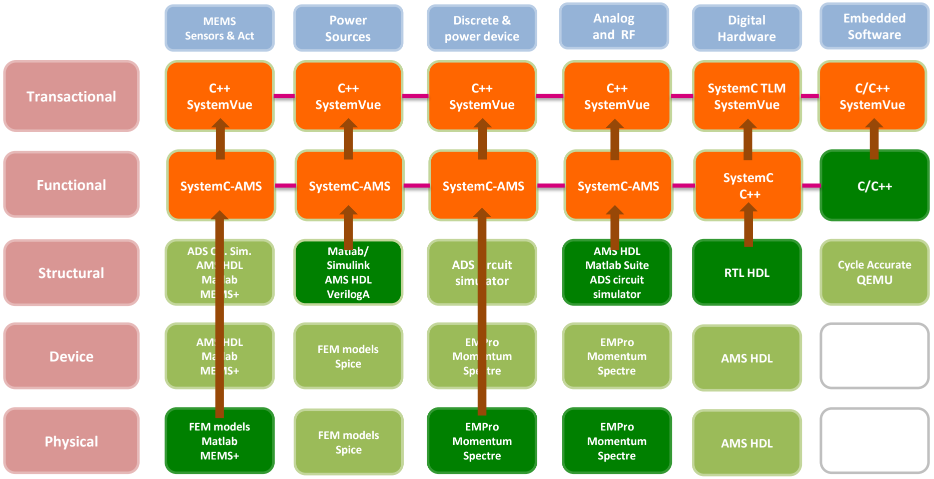
From Heterogeneous co-simulation…



	MEMS Sensors & Act	Power Sources	Discrete & power device	Analog and RF	Digital Hardware	Embedded Software
Transactional	C++ SystemVue	C++ SystemVue	C++ SystemVue	C++ SystemVue	SystemC TLM SystemVue	C/C++ SystemVue
Functional	SystemC-AMS	SystemC-AMS	SystemC-AMS	SystemC-AMS	SystemC C++	C/C++
Structural	ADS Cir. Sim. AMS HDL Matlab MEMS+	Matlab/ Simulink AMS HDL VerilogA	ADS circuit simulator	AMS HDL Matlab Suite ADS circuit simulator	RTL HDL	Cycle Accurate QEMU
Device	AMS HDL Matlab MEMS+	FEM models Spice	EMPro Momentum Spectre	EMPro Momentum Spectre	AMS HDL	
Physical	FEM models Matlab MEMS+	FEM models Spice	EMPro Momentum Spectre	EMPro Momentum Spectre	AMS HDL	

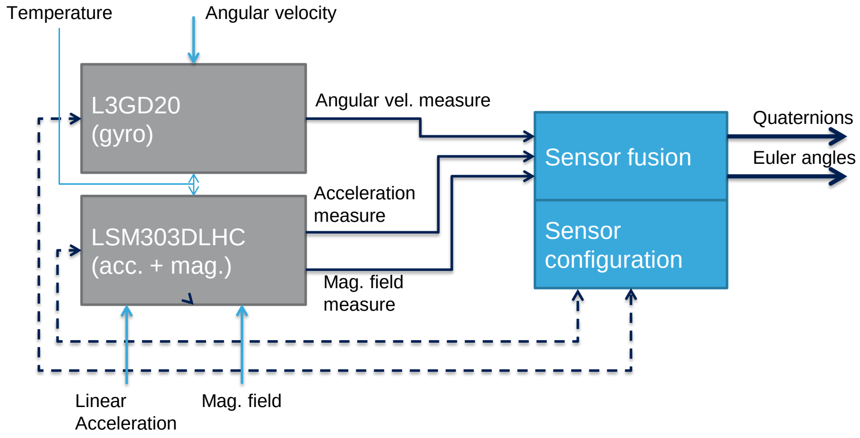
www.fp7-smac.org

…to homogeneous simulation



	MEMS Sensors & Act	Power Sources	Discrete & power device	Analog and RF	Digital Hardware	Embedded Software
Transactional	C++ SystemVue	C++ SystemVue	C++ SystemVue	C++ SystemVue	SystemC TLM SystemVue	C/C++ SystemVue
Functional	SystemC-AMS	SystemC-AMS	SystemC-AMS	SystemC-AMS	SystemC C++	C/C++
Structural	ADS C . Sim. AM HDL Ma ab ME S+	Matlab/ Simulink AMS HDL VerilogA	ADS rcuit simu ator	AMS HDL Matlab Suite ADS circuit simulator	RTL HDL	Cycle Accurate QEMU
Device	AM HDL Ma ab ME S+	FEM models Spice	EM ro Mom ntum Sp tre	EMPro Momentum Spectre	AMS HDL	
Physical	FEM models Matlab MEMS+	FEM models Spice	EMPro Momentum Spectre	EMPro Momentum Spectre	AMS HDL	

iNEMO-M1 Functional model



Temperature

Angular velocity

L3GD20
(gyro)

Angular vel. measure

LSM303DLHC
(acc. + mag.)

Acceleration
measure

Mag. field
measure

Sensor fusion

Sensor
configuration

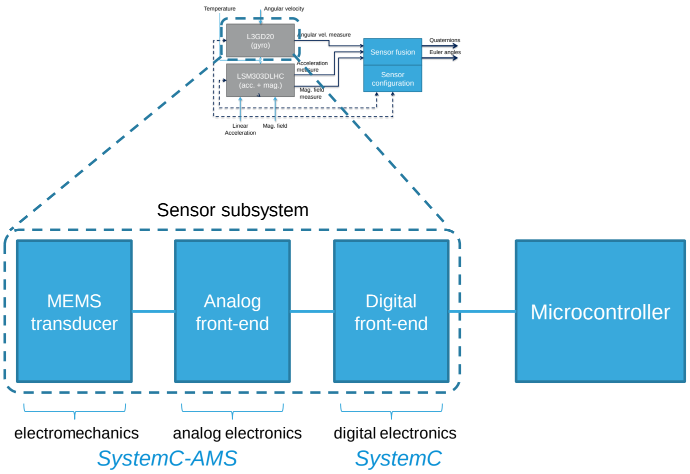
Quaternions

Euler angles

Linear
Acceleration

Mag. field

iNEMO-M1 Functional model [cont.]



Sensor subsystem

MEMS
transducer

Analog
front-end

Digital
front-end

Microcontroller

electromechanics

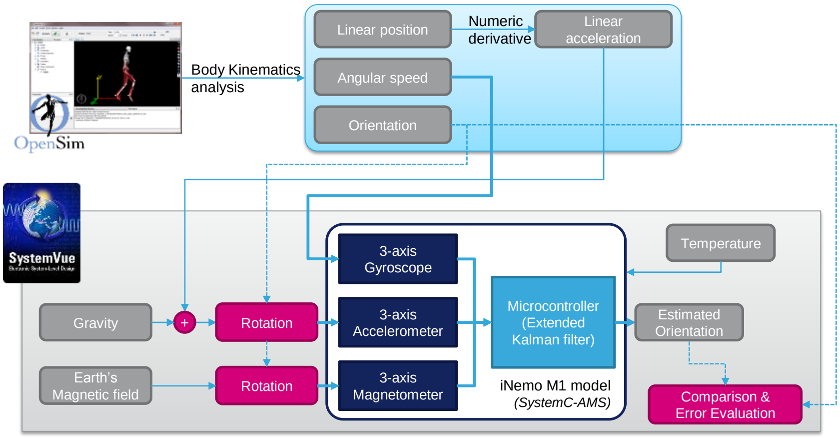
analog electronics

digital electronics

SystemC-AMS

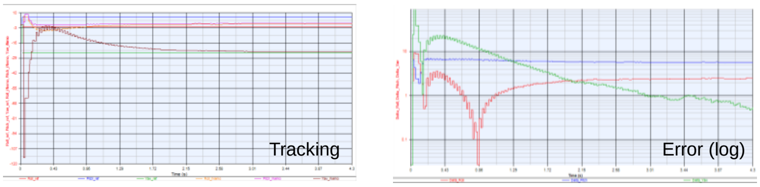
SystemC

System-level simulation platform

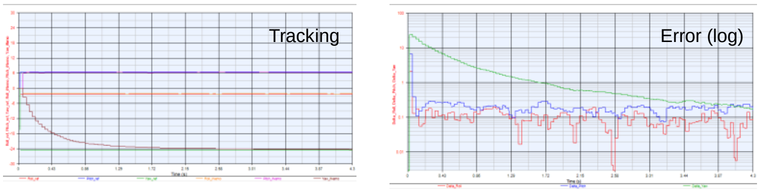


Functional optimization

- Evaluation in static condition: roll, pitch, yaw
  - Original iNEMO algorithm → maximum error > 8° after initial setup

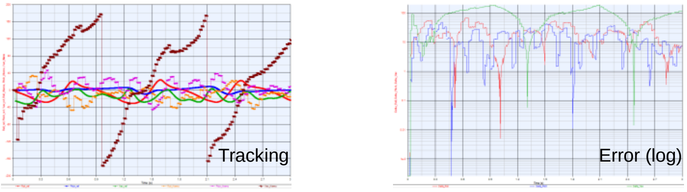


  - EKF improvement (magnetic field model) → maximum error < 0.5° after initial setup

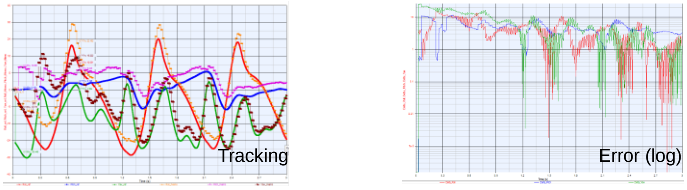


Functional optimization [cont.]

- Evaluation in dynamic condition (simulated walk): roll, pitch, yaw
  - Original iNEMO algorithm → no convergence in 3s

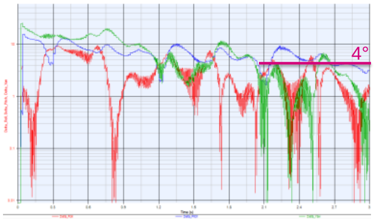


  - EKF improvement (magnetic field model) → max. error < 4°after initial setup

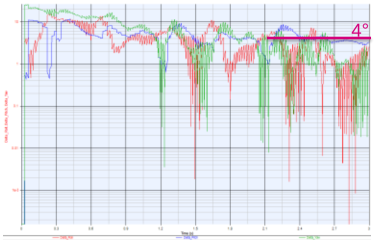


Energy estimation and optimization

- Evaluation in dynamic condition (simulated walk): roll, pitch, yaw
  - Angle estimation update error with 100Hz data sampling
    - Required processor frequency: 72MHz
    - **150mW**



  - Angle estimation update error with 50Hz data sampling
    - Comparable error magnitude
    - Processor frequency: 18MHz
    - Power consumption 35% lower: **98mW**

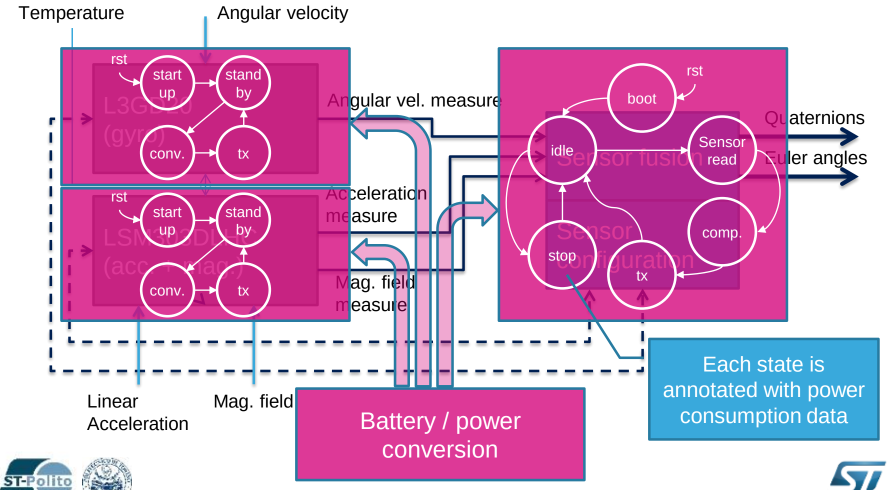


https://contrex.offis.de/

iNEMO-M1 Functional model

Energy modeling layer

- Activity supported by the EU-funded FP7 Project CONTREX

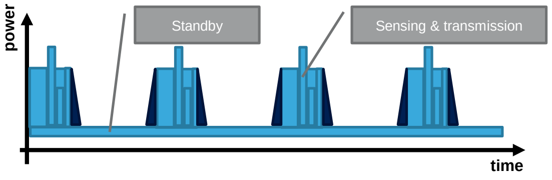


Battery / power conversion

Each state is annotated with power consumption data

System-level energy modelling summary

- Coarse modelling
  - Active and low-power modes, duty cycling…
  - Data from datasheet and/or measurements
  - Manageable with spreadsheet
- Finer-grained modelling
  - Transients (e.g., wake-up time), code profiling
  - Requires simulation and detailed device characterization
- System-level simulation-based energy estimation for WSNs
  - Network traffic evaluation (e.g., TOSSIM, J-SIM, SCNSL, Cooja, …)
  - Node-level extra-functional simulation (e.g., SWAT, N2Sim, …)



System level energy modelling summary

- Coarse
  - Active
  - Data
  - Mana
- Fin
  -
  -
- Sys
  -
  -