IoT Intro

Olav Solgaard

Bio

Olav Solgaard earned his Ph.D. degree from Stanford University in 1992. His doctoral dissertation: "Integrated Semiconductor Light Modulators for Fiber-optic and Display Applications" was the basis for the establishment of a Silicon Valley firm Silicon Light Machines (SLM), co-founded by Dr. Solgaard in 1994. From 1992 to 1995 he carried out research on optical MEMS as a Postdoctoral Fellow at the University of California, Berkeley, and in 1995, he joined the Electrical Engineering faculty of the University of California, Davis. His work at UC Davis led to the invention of the multi-wavelength, fiber-optical switch, which has been developed into commercial products by several companies. In 1999 he joined Stanford University where he is now a Professor of Electrical Engineering and the Director of Graduate Studies in the Department of Electrical Engineering.

Professor Solgaard's research interests include optical MEMS, Photonic Crystals, optical sensors, microendoscopy, atomic force microscopy, and solar energy conversion. He has authored more than 350 technical publications and holds 70 patents. Professor Solgaard is a Fellow of the IEEE, the Optical Society of America, the Royal Norwegian Society of Sciences and Letters, and the Norwegian Academy of Technological Sciences.

IoT Overview

- What are the disciplines of IoT?
  - Cool applications and fantastic opportunities – Ada Poon
  - Sensors – Beth Pruitt
    - Small, inexpensive and very, very good
  - Circuits – Boris Murmann
    - This is where it started
  - Embedded systems – Phil Levis
    - Sensors, circuits and communication links playing together
  - Networks – Ayfer Özgür-Aydin
  - How does the internet support IoT?
- Big data and machine learning
  - Other certificates focus on these aspects of IoT

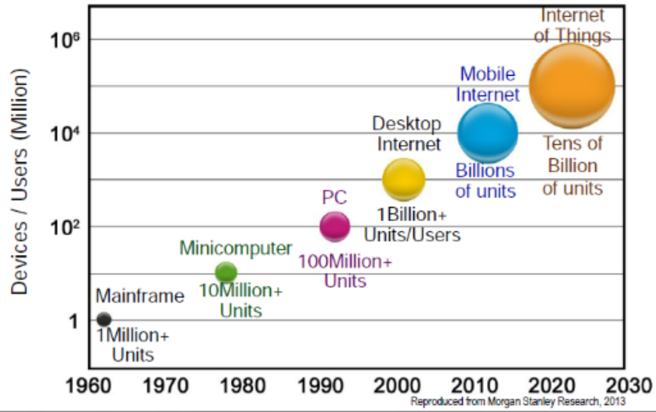
Why is it important?

- Fourth industrial revolution
- IoT will touch everything
- Everything will have IoT

Ideal IoT

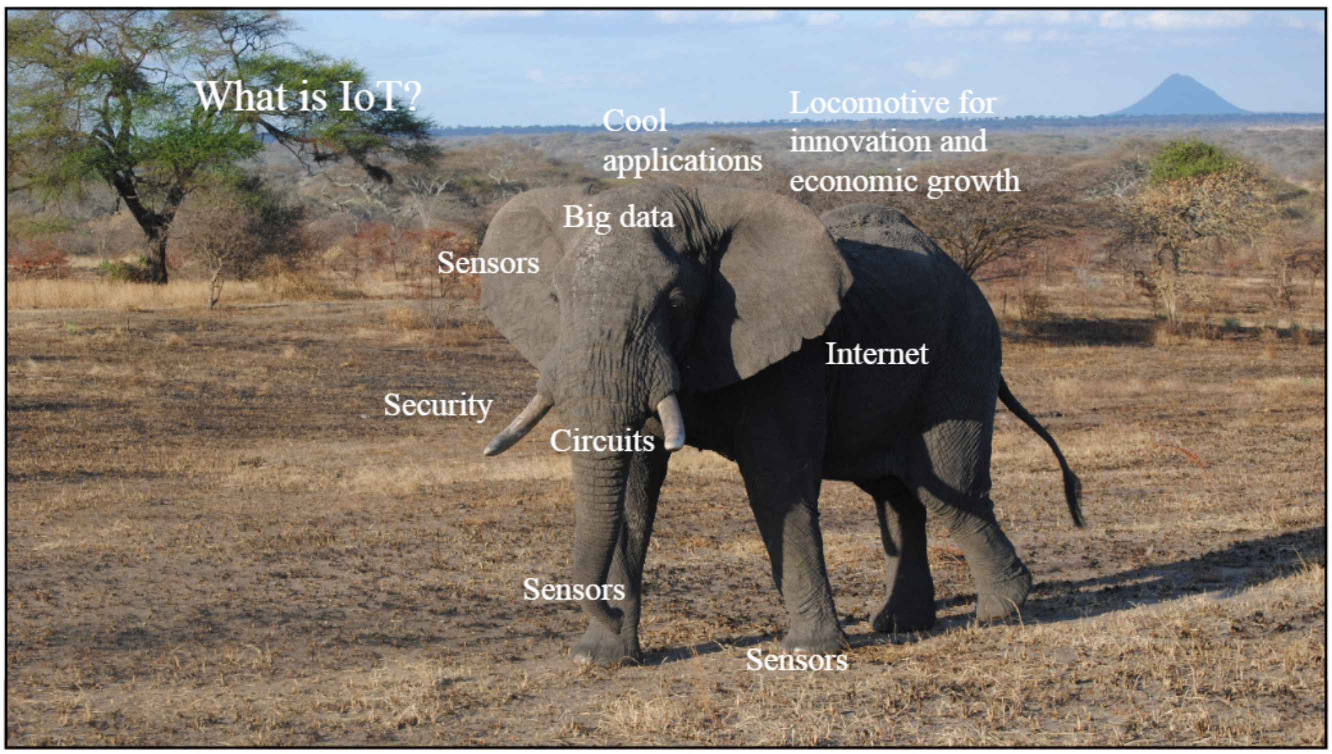
- Ubiquitous
- Smart
- Agile
- On demand
- Blend into background
- Secure
- Low maintenance
- Fast
- Upgradable
- Growing
- Adaptable

IoT will be the "next big thing"



Major Technology Cycles = 10x More Users & Devices
Driven by: 1) Lower Price, 2) Improved Functionality & Services

Tsai, 2014 International Solid-State Circuits Conference (ISSCC), Keynote Talk



IoT Overview

- What will you learn in the short course?
- These are the questions we will help you answer:
  - What is your perspective on IoT?
  - How does the disciplines of IoT overlap with your expertise and interests?
  - Where are the holes in your knowledge that you want to fill?
  - What IoT opportunities do you want to pursue?
- And ultimately
  - How to get the most out of your IoT certificate

Cool Applications – Professor Ada Poon

- Sustain (Smart cities)
- Move (Self driving cars)
- Heal (Healthcare)
- Feed (Agriculture)
- Make (Manufacturing and packaging)

- The purpose of this section is two fold:
  - Get you excited about IoT potential
  - And more importantly, have you thinking of better ideas than the ones already out there

Smart Waste Management

http://ecubelabs.com/case-studies/dublin-airport/

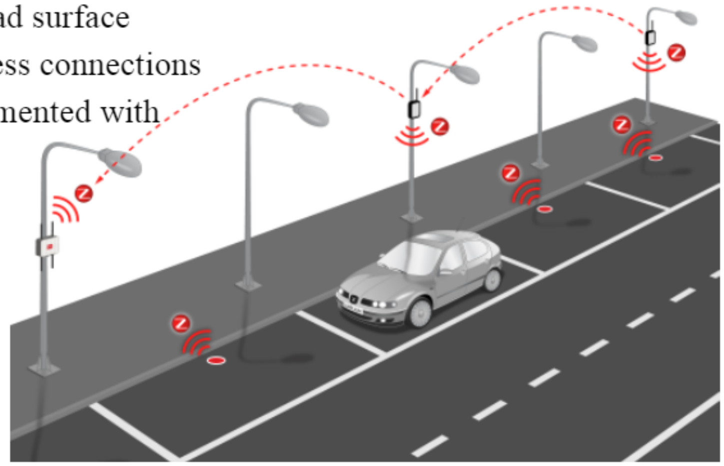
- Go from collecting 840 containers 4 times a day to collecting 80 containers a day.
- Increase waste collection efficiency by 90%.

Smart Street Lights

http://www.tvilight.com/

Smart Street Parking

- Infrared- and magnetic-based vehicle detection sensor mounted on the road surface
- Zigbee, LoRaWAN wireless connections
- Mesh networks are implemented with in street lights.
- Apps to direct drivers to empty spaces
- Dynamic parking pricess

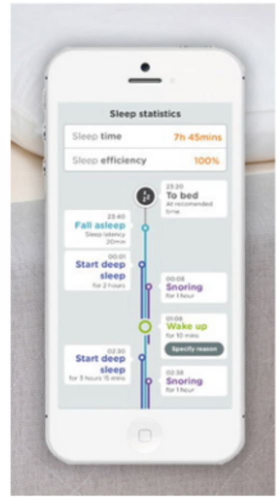


http://www.libelium.com/smart_parking/

Connected Vehicles – Combine private and collective transportation: The best of both

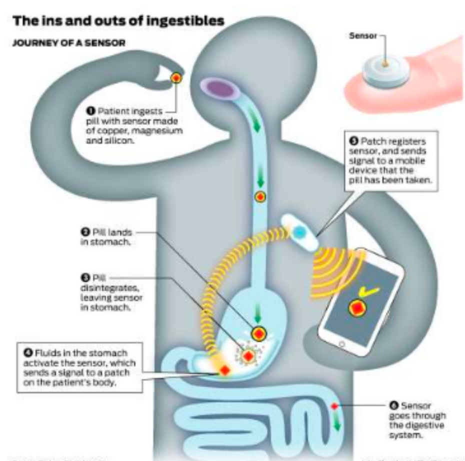
http://government-2020.dupress.com/trend/connected-vehicles/

Mood Enhancing – Sleep Monitoring



https://www.beddit.com/

Enhance Adherence – Ingestible Sensors



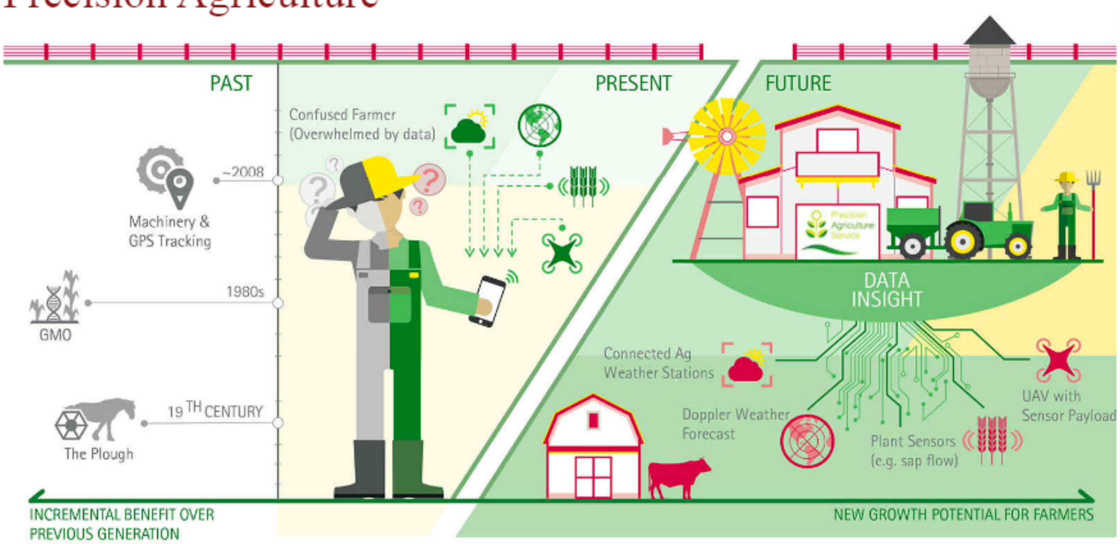
http://www.proteus.com/
http://www.sfchronicle.com/business/article/Did-you-take-your-pill-Ingestible-sensors-can-11206980.php

Challenges

- A lack of Electronic Health Record (EHR) integration and concerns about data security may prevent healthcare from fully adopting the IoT technology.
- The need to adopt an integration-first mindset instead of keep building interesting/fun gadgets. Sometimes, a dumb gadget can be as useful if it could integrate seamlessly with the EHR.

- False positives
  - Doctor's offices will be filled with perfectly healthy, but anxious patients

Precision Agriculture



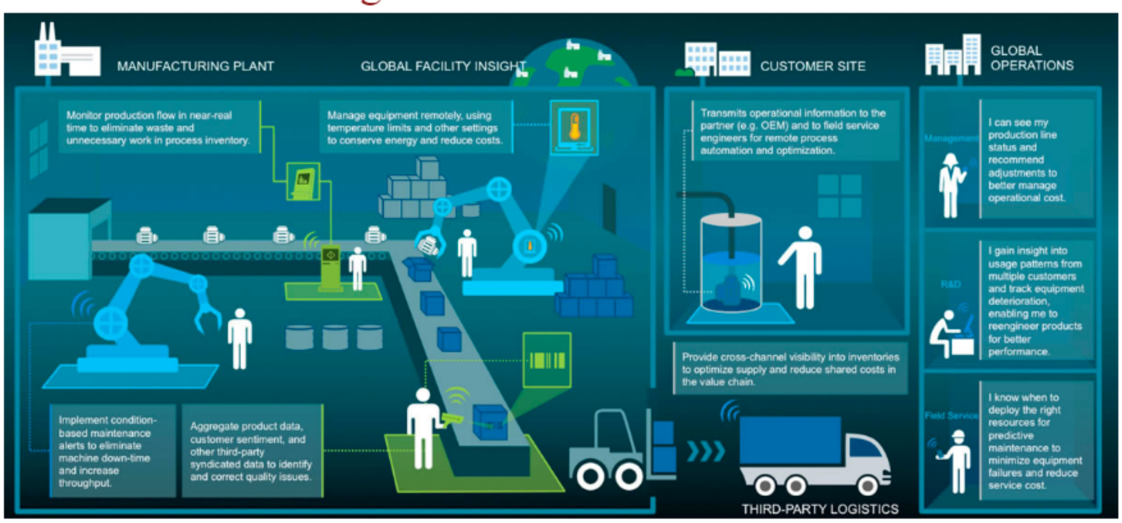
https://www.accenture.com/cn-en/insight-accenture-digital-agriculture-solutions

Food Safety & Storage Efficiency

- Wifi or cellular connectivity
- When a produce recall is initiated, the juice machine will check the packs and prevent the machine from pressing affected packs.

https://www.juicero.com/

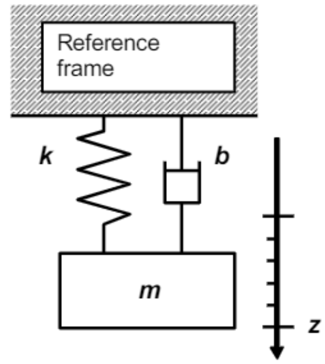
Smart Manufacturing



https://www.slideshare.net/andrejt/ntk-2015-internet-of-things-track-iot-smart-home

Scaling of sensors

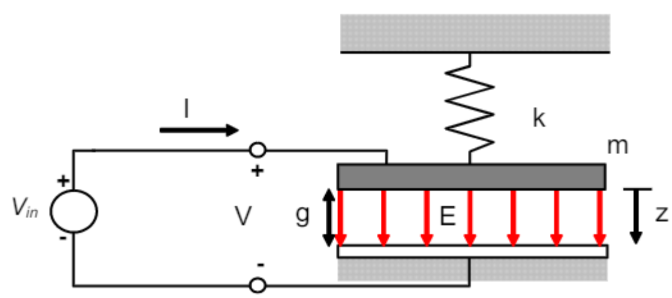
- Many sensors can be modeled as simple mechanical harmonic oscillators
- The figure shows an accelerometer
  - The proof mass (m) is accelerated and displaced
  - The displacement (Δz) of m is measured
  - The acceleration is calculated as $a = k \cdot \Delta z / m$ (The spring constant, k, is known)
  - The highest frequency that this accelerometer can measure is $f = \frac{1}{2\pi}\sqrt{\frac{k}{m}}$



Accelerometer modeled as mechanical harmonic oscillator

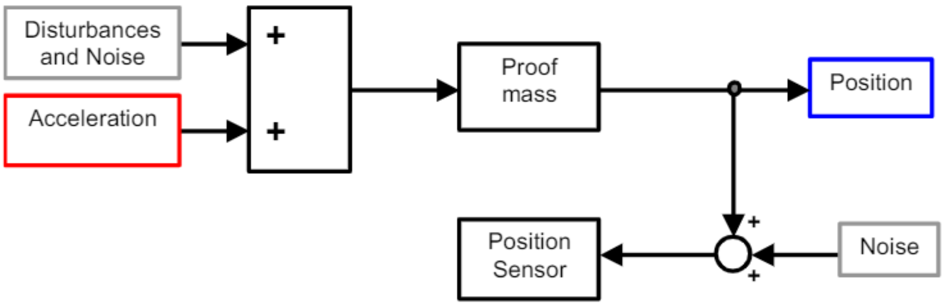
Capacitive measurement of displacement

- Most IoT sensors measure changes in electrical properties:
  - Resistance, capacitance, inductance
- Our accelerometer is completed by capacitive measurement of displacement
  - Capacitors are great at measuring short distance with very good absolute accuracy
- Measurements of large displacements are better done by inductors (magnetic fields)



Block diagram of MEMS accelerometer

Noise and disturbances enter the system both as an equivalent acceleration noise, and as an equivalent signal noise in the position sensor.

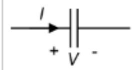
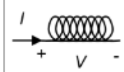


MEMS sensors: Small is beautiful

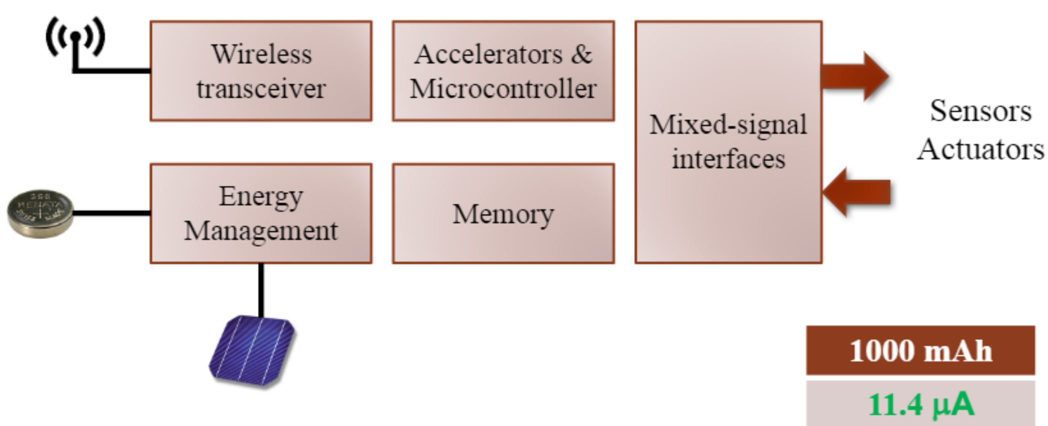
$$a = \frac{k}{m}\Delta z = (2\pi \cdot f)^2 \cdot \Delta z \qquad f = \frac{1}{2\pi}\sqrt{\frac{k}{m}}$$

- Design objective: Measure the smallest possible a up to a given f
- Looking at the first part of the equation, we might think that we should use a low k and large m
  - => low f
  - Given a constant f, only the ratio $k/m \sim f^2$ is important
  - A miniaturized accelerometer must have a small proof mass and a weak spring, but it is the ratio of the two that is important!
  - To measure small accelerations up to a given f, we therefore have to measure Δz very accurately
  - For application specific minimum detectable acceleration and bandwidth, there is a minimum detectable displacement that our position sensor must be able to measure
- Small structures have position sensitivity and can be designed to have any resonance frequency
- => MEMS sensors!

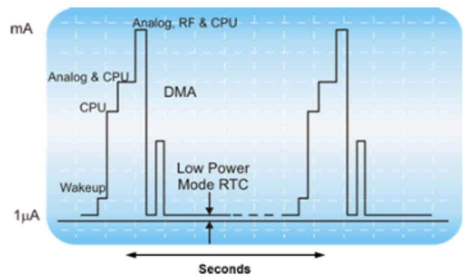
Electrical components used in miniaturized sensors

	Resistor	Capacitor	Inductor
Circuit Symbol			
I-V	$V = R \cdot I$	$I = C \cdot dV/dt$	$V = L \cdot dI/dt$
Units	Ohm: $\left(\frac{Vm}{A}\frac{m}{m^2} = \frac{V}{A} = \Omega\right)$	Farad: $F = \left(\frac{C}{V}\right)$	Henry: $\left(H = \frac{V \cdot s^2}{C}\right)$
Impedance	$Z = R$	$Z_C = 1/j2\pi f \cdot C$	$Z_L = j2\pi f \cdot L$
Power Loss	$P = V \cdot I = R \cdot I^2 = \frac{V^2}{R}$	0	0
Energy Storage	0	$W = \frac{C \cdot V^2}{2}$	$W = \frac{L \cdot I^2}{2}$

Generic block diagram



Key: Duty-cycling



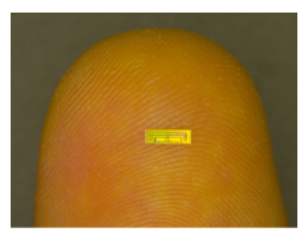
- IoT transceivers are “mostly” off
- Example: Receive/transmit within 1 ms, then go to sleep for 1 sec

http://www.mouser.com/applications/low-power-ewc-design/

Power/energy cost of wireless transmission

	Bluetooth LE	Nike+	ZigBee	WIFI
Power (mW)	0.147	0.675	35.7	**210**
Bits/sec	960	272	192	40M
Energy/bit (nJ)	153	2480	186,000	**5.25**

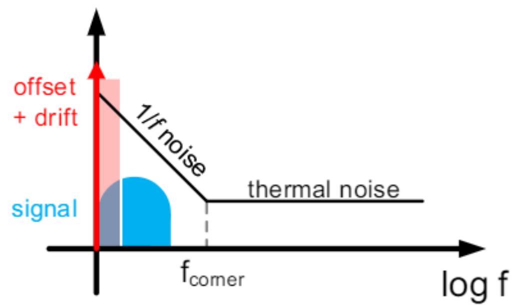
Ant-sized radio



- Numbers are a strong function of reach
- Energy per bit tends to be lower for high-data rate links → Invested power amortizes better

https://www.digikey.com/en/articles/techzone/2011/aug/comparing-low-power-wireless-technologies

Key issue for sensors: 1/f noise



- Analog MOS circuits come with a variety of low frequency artifacts that can easily ″swamp out″ DC or low-frequency signals of interest

What is an embedded system?

> "An embedded system is a computerized system that is purpose-built for its application."
>
> Elicia White
> Making Embedded Systems, O'Reilly

This has deep implications for how they are designed and why.
Typically, want to minimize cost/maximize lifetime for a given expected workload.
Optimized, custom software that uses as few resources as possible.

Two basic cost considerations dominate

Energy

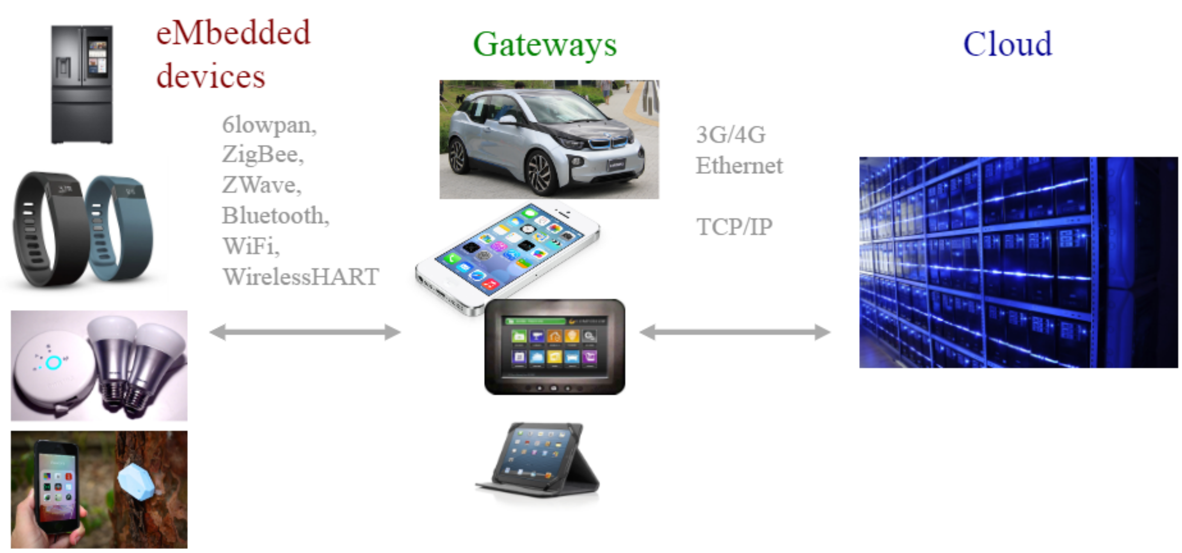
- Embedded systems have an expected workload – want to minimize energy required to handle the workload (longer battery life, more efficient)
- More powerful, highly featured MCUs draw more power: can be more *power efficient*, but consume more energy for a given workload

Money

- Many embedded systems markets have tight margins (cars, appliances, etc.)
- More powerful, highly featured MCUs cost more

Both costs push designs to pick minimal MCU and optimize code for it

IoT: MGC Architecture

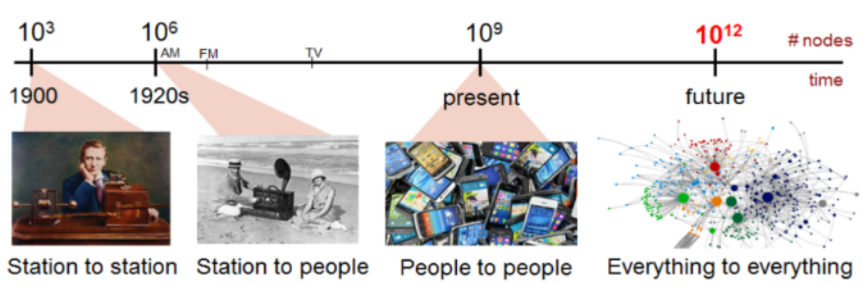


Two Game-Changers

ARM Cortex M series
- First released 2004
- Ultra-low power 32-bit processor
- 8-96kB of RAM, 64-512kB code flash
- Sleep currents recently dropped <1μA

Bluetooth Low Energy
- First released in 2006
- Send a 30 byte packet once per second, last for a year on a coin cell battery
- Support was weak until Apple incorporated into iBeacon, now all major smartphones include it

Historical Perspective on Wireless Systems

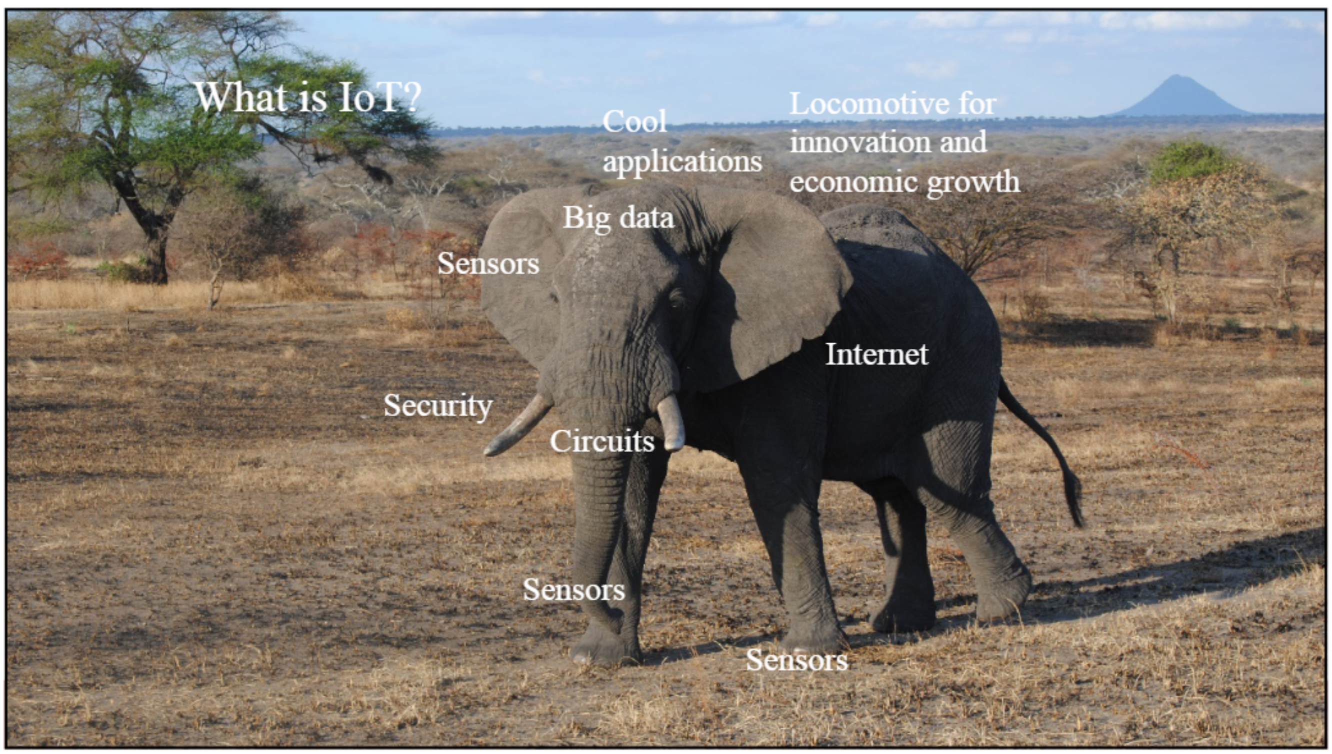


Paradigm shift in principles, technology, systems and applications.

Massive Multiple Access Networks

Tomorrow's IoT Networks:

- Massive number of devices, only few active at a time.
- Bursty traffic, few bits to transmit.
- Complexity and energy constrained.
- Node identity may not be relevant.



IoT Overview

- What will you learn in the short course?
- These are the questions we will help you answer:
  - What is your perspective on IoT?
  - How does the disciplines of IoT overlap with your expertise and interests?
  - Where are the holes in your knowledge that you want to fill?
  - What IoT opportunities do you want to pursue?
- And ultimately
  - How to get the most out of your IoT certificate

Cool Applications

Professor Ada Poon

Outline

- Smart cities
- Healthcare
- Agriculture
- Manufacturing and logistics

INTEGRATED REPORTING AND ANALYTICS: ACTIONABLE INSIGHT

CITY INTELLIGENCE | SMART OPERATIONS | CITIZEN RELATIONSHIP MANAGEMENT

ENERGY
- SMART BUILDINGS
- CONDITION BASED MAINTENANCE
- REMOTE OUTAGE NOTIFICATION
- SMART WASTE MANAGEMENT

UTILITY
- WATER TREATMENT
- WATER MANAGEMENT
- EQUIPMENT MONITORING/ CONTROL
- HAZARDOUS MATERIALS EMERGENCY RESPONSE

VEHICLE
- SMART PARKING
- PARKING ENFORCEMENT
- VEHICLE DETECTION
- MOBILE PAYMENTS
- EV CHARGING

TRANSIT
- INTELLIGENT RAIL AND TRANSIT SOLUTIONS
- FLEET MANAGEMENT
- ASSET TRACKING
- MOBILE PAYMENTS
- SMART ROADS

PUBLIC SAFETY
- VIDEO SURVEILLANCE
- REMOTE SECURITY MONITORING
- EMERGENCY RESPONSE
- SMART STREET LIGHTS
- MASS NOTIFICATIONS

Source: Verizon. Smart Cities Solutions, 2014

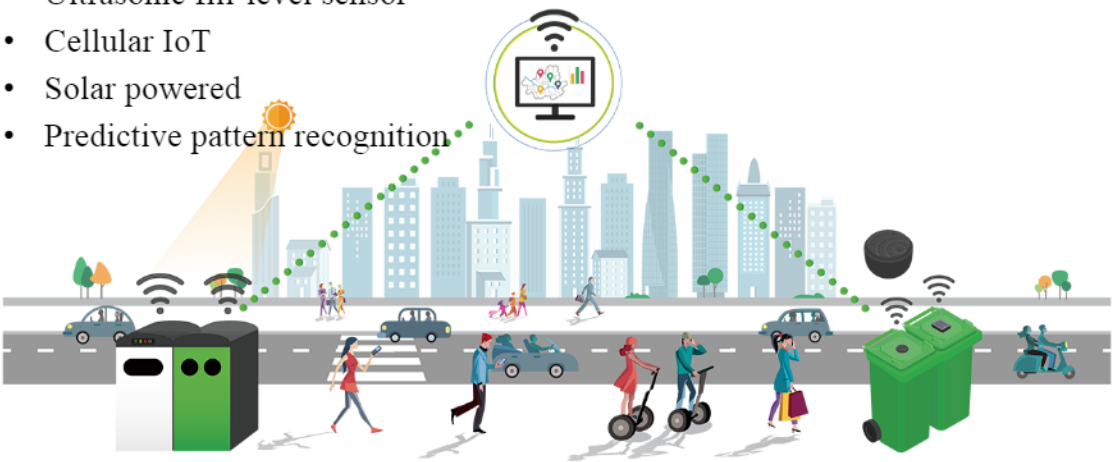
Smart Waste Management

http://ecubelabs.com/

http://bigbelly.com/

Smart Waste Management

- Ultrasonic fill-level sensor
- Cellular IoT
- Solar powered
- Predictive pattern recognition



http://ecubelabs.com/integrated-waste-management/

Smart Waste Management

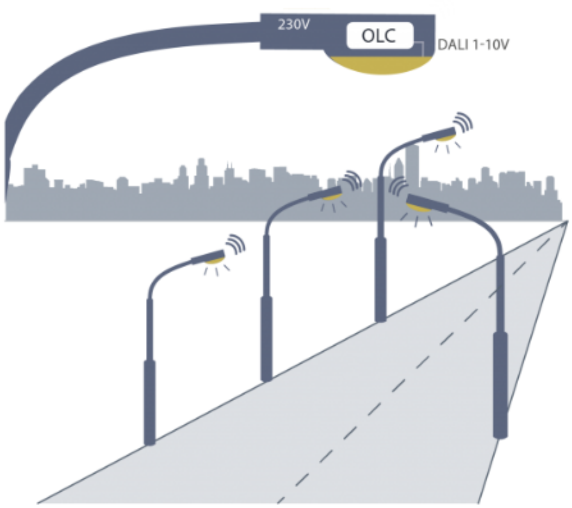
http://ecubelabs.com/case-studies/dublin-airport/

- Go from collecting 840 containers 4 times a day to collecting 80 containers a day.
- Increase waste collection efficiency by 90%.

Smart Street Lights

http://www.tvilight.com/

Smart Street Lights



- Light sensor, motion sensor
- Cellular IoT
- Real-time mesh network

https://chess-wise.eu/en/smart-street-lighting/

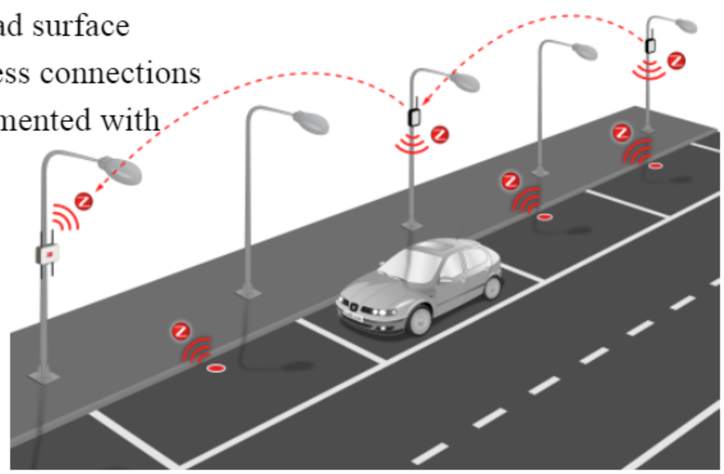
Smart Street Parking

- In the past year, how many times did you give up when looking for a parking space in cities? How many times did you argue with someone about a parking spot? On average, how long did you take to look for a parking spot?
- More than 30% of a city's traffic is caused by drivers searching for a parking spot.
- In New York City, 29% of commuters said that they spent 20 minutes on average looking for a parking spot and 10% spent more than 40 minutes.

http://www.govtech.com/transportation/Smart-Parking-Tech-US-Cities.html

Smart Street Parking

- Infrared- and magnetic-based vehicle detection sensor mounted on the road surface
- Zigbee, LoRaWAN wireless connections
- Mesh networks are implemented with in street lights.
- Apps to direct drivers to empty spaces
- Dynamic parking prices

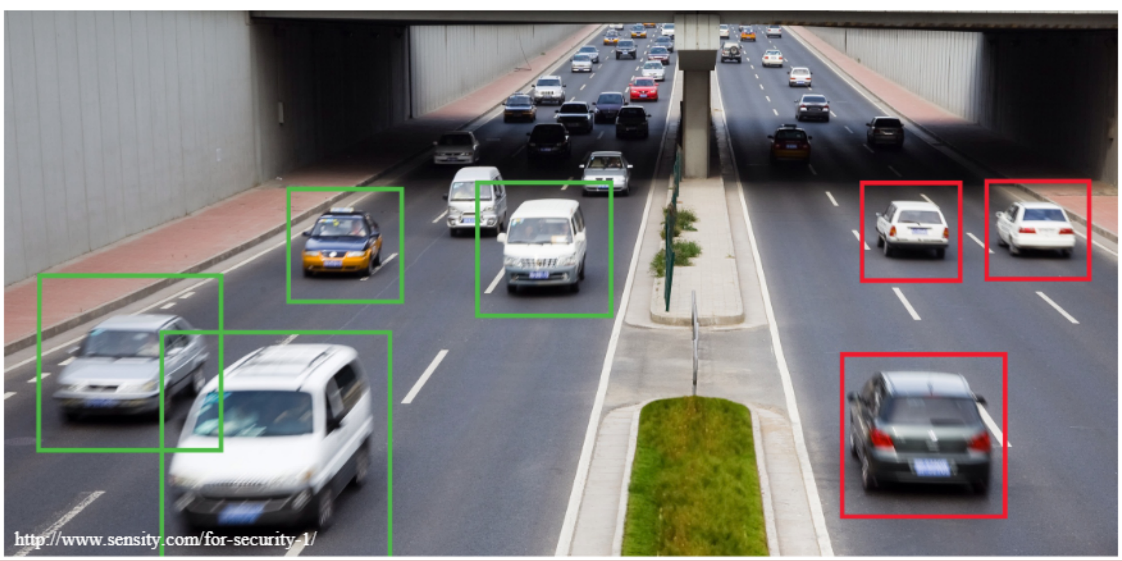


http://www.libelium.com/smart_parking/

Smart Street Parking in Poland

http://www.worldsensing.com/success-story/gliwice-smart-parking-solution/#

Security without Surveillance



http://www.sensity.com/for-security-1/

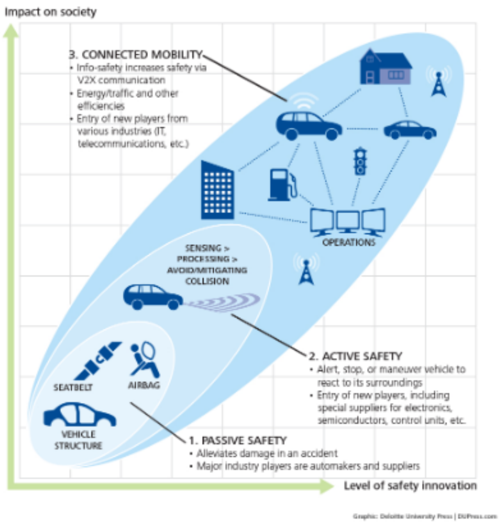
Security without Surveillance

- Real-time analytics rather than human-monitored surveillance
- Edge-based analytics rather than cloud-based analytics
  - All videos are stored locally.
  - This also reduces the requirement on datarate.
- Resultant analytics sent to central cloud database for issuing alerts

Connected Vehicles

http://government-2020.dupress.com/trend/connected-vehicles/

Connected Vehicles – Stages of Safety Innovation



- Passive vehicle safety uses sensors to take the vehicle's immediate surroundings into consideration.
- Recent efforts enable the sharing of information gathered by the sensors between vehicles, and between vehicles and their surroundings to increase safety further.
  - V2X (vehicle-to-X, where X represents other vehicles, infrastructure, roads, and so on)
  - A step towards autonomous driving

https://dupress.deloitte.com/dup-us-en/deloitte-review/issue-12/the-rise-of-safety-innovations-in-intelligent-mobility.html

Let's Recap

Sensors

- Ultrasonic fill-level sensor
- Light and motion sensors
- Infrared- and magnetic-based vehicle detection sensor
- Camera

Communication Protocols

- Cellular IoT (LTE Cat-M1 and NB-IOT)
- Zigbee, LoRaWAN
- V2X

Networking Protocols

- Mesh network

Outline

- Baby monitoring
- Elderly monitoring
- Mood enhancement
- Disease treatment and progression monitoring
- Enhance adherence

Baby Monitoring – Activity Tracking

- Body position
- Breathing
- Oxygen level
- Skin temp
- Wake/sleep pattern
- ECG

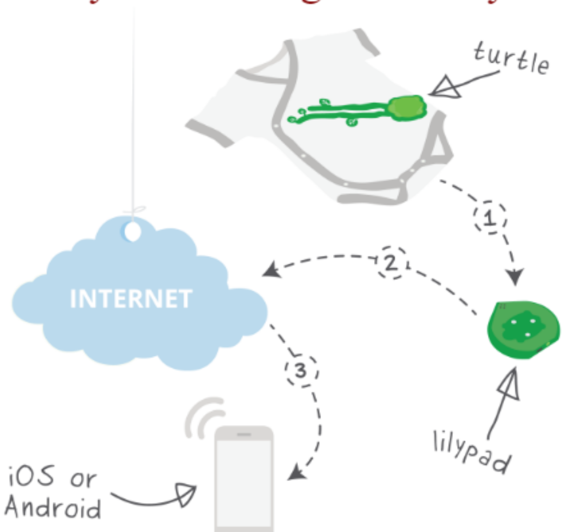
Can be integrated with other IoT devices such as thermostat and camera to “close the loop”. For example, if the baby is too warm, the thermostat will automatically adjust.

Baby Monitoring – Activity Tracking

http://mimobaby.com/

http://www.owletcare.com/smart-sock-2/

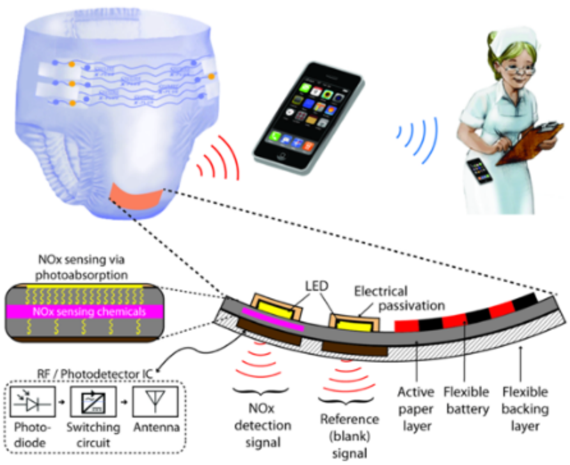
Baby Monitoring – Activity Tracking



1. The turtle sends information about the baby's breathing, body position, sleep activity, and skin temp to the Lilypad via Bluetooth LE.
2. The Lilypad streams data and live audio to the cloud via WiFi.
3. Parents receive real-time insight about their baby on their smartphone.

http://mimobaby.com/

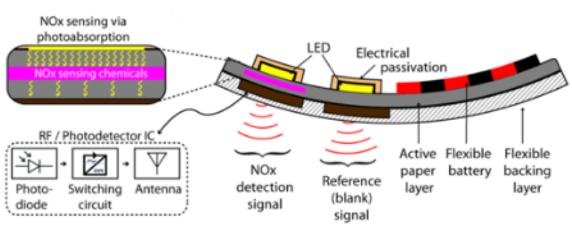
Baby Monitoring – Urinary Tract Infection Monitoring



- Urinary tract infection (UTI) is the second most common infection in the US accounting for 7M hospital visits and 100,000 hospitalization per year.
- It is easy to cure if detected and treated in early stage.
- Urine culture test is accurate but time consuming. Dip stick test is fast but high false alarm rate.

https://sites.google.com/site/jagpurdue/projects/catalyst

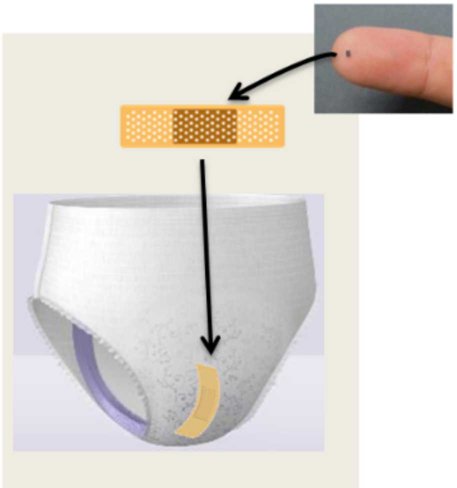
Baby Monitoring – Urinary Tract Infection Monitoring



- Urine-activated paper battery (self-powered)
- Paper-based colorimetric nitrite sensor consisting of an LED, a urine-absorbing strip, a reagent strip, an active photodiode, and a reference photodiode.
- Sensor signal is converted into a PWM waveform.
- BLE module transmits the PWM signal to the caregiver.

https://sites.google.com/site/jagpurdue/projects/catalyst

Elderly Monitoring – Incontinence Management



http://rfmicron.com/health-care/

- In most nursing homes, between 40% and 60% of residents suffer from urinary incontinence.
- Smart diaper allows caregivers to remotely detect if an incontinence event has occurred.
- Improved quality and dignity of care by not having to disturb the elderly.

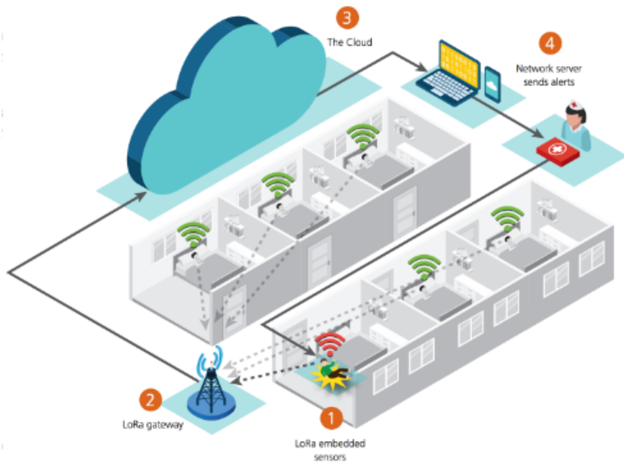
http://www.medisens.com/news/press-release-1

Elderly Monitoring – Fall Detection

- One-fourth of Americans aged 65+ falls each year.
- Every 11 seconds, an older adult is treated in the emergency room for a fall; every 19 minutes, an older adult dies from a fall.
- Falls are the leading cause of fatal injury and the most common cause of nonfatal trauma-related hospital admissions among older adults.
- Falls result in more than 2.8 million injuries treated in emergency departments annually, including over 800,000 hospitalizations and more than 27,000 deaths.
- In 2013, the total cost of fall injuries was $34 billion.
- The financial toll for older adult falls is expected to increase as the population ages and may reach $67.7 billion by 2020. (The medicare budget is $584 billion in 2016.)

https://www.ncoa.org/news/resources-for-reporters/get-the-facts/falls-prevention-facts/

Elderly Monitoring – Fall Detection



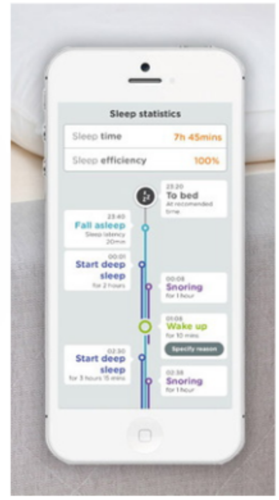
1. Fall/movement data collected by sensors embedded with LoRa Technology
2. Data from all sensors is sent to a LoRa gateway as person moves
3. Gateway sends information to the Cloud where the data is analyzed by an application to determine what is normal and what is a fall
4. Application server sends reports and alerts on the fall and location of the person to a computer or mobile device

http://www.semtech.com/wireless-rf/internet-of-things/downloads/Semtech_Health_FallDetection_AppBrief-FINAL.pdf

Mood Enhancement

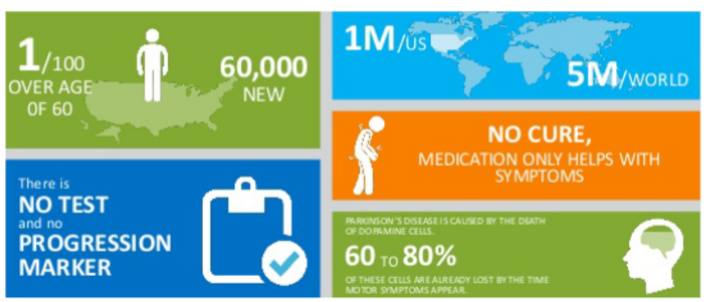
- Relaxing music could be cued to ease stress.
- Window shades could be programmed to let in the maximum amount of natural light.
- Use IoT to encourage healthy behaviors.
  - Automatically dim the lights in the home at a recommended bedtime.
  - Automatically turn off the TV to encourage exercise.

Mood Enhancing – Sleep Monitoring



https://www.beddit.com/

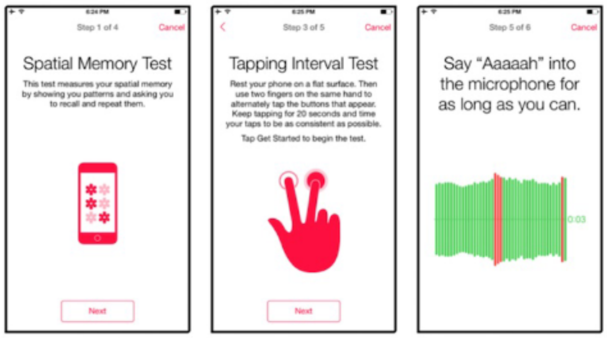
Disease Treatment and Progression Monitoring – Parkinson's



Patients with Parkinson's disease must be continually assessed in order to keep up with their symptoms. This becomes potentially problematic as symptoms fluctuate on a constant basis, and a monthly check in with their doctor may not be representative of their experience.

https://www.slideshare.net/rcnossen/healthcare-iot-and-analytics-to-treat-parkinsons

Disease Treatment and Progression Monitoring – Parkinson’s



Instead of patients actively performing certain tasks, could we monitor disease progression passively in the background?

http://parkinsonmpower.org/

Disease Treatment and Progression Monitoring – Parkinson’s

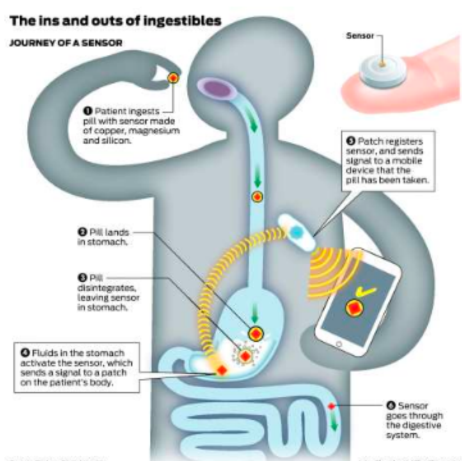
- IoT opens up new possibilities for disease treatment through remote monitoring. Using sensors, mobile devices, and advanced machine learning capabilities, a patient could keep track of a host of valuable data from mobility to sleep patterns all in real time. This information will give practitioners a more complete look into the progression of their patient’s disease states.
- Intel and Michael J. Fox Foundation for Parkinson’s Research, and Pfizer and IBM are individually collaborating on this idea. The collaboration involves planned clinical trial.

Enhance Adherence

- 84% of U.S. healthcare spending is on patients with chronic conditions.
- More than 50% of prescribed medications are not taken as directed.
- Reasons why people are not able to take their medication as directed:
  - They may forget.
  - They may not be convinced of the medication's effectiveness or be unsure that it is working.
  - They may fear the side effects or have difficulty taking the medication.
  - The rising cost of prescription medications is a barrier for many.

Anderson G. Chronic conditions: making the case for ongoing care. Baltimore, MD: Johns Hopkins University, 2010.
Sabate E. Adherence to long-term therapies: evidence for action. World Health Organization, 2003.

Enhance Adherence – Ingestible Sensors



http://www.proteus.com/
http://www.sfchronicle.com/business/article/Did-you-take-your-pill-Ingestible-sensors-can-11206980.php

Challenges

- A lack of Electronic Health Record (EHR) integration and concerns about data security may prevent healthcare from fully adopting the IoT technology.
- The need to adopt an integration-first mindset instead of keep building interesting/fun gadgets. Sometimes, a dump gadget can be as useful if it could integrate seamlessly with the EHR.

Outline

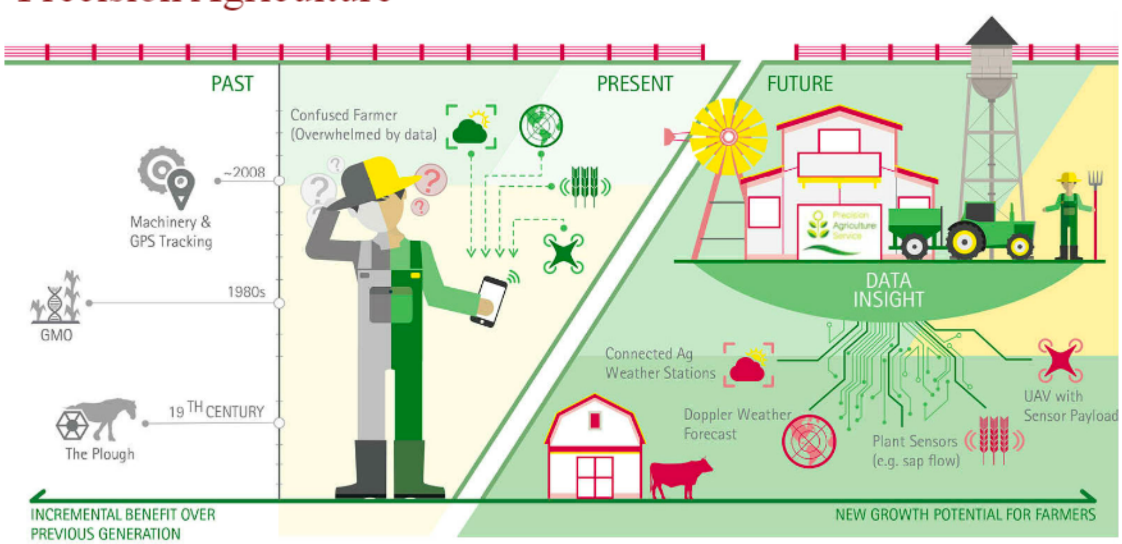
- Precision agriculture
- Connected livestock
- Food safety

Precision Agriculture

- A farming management concept based on observing, measuring and responding to inter and intra-field variability in crops.
- In the past, precision agriculture technology was implemented by big agribusinesses due to high costs.
- IoT technologies – which include everything from GPS services, sensors, and big data calculation – have made precision agriculture affordable by many farmers.
- Farmers don’t have to rely as much on their gut. Instead, they can make decisions based on detailed information about water, climate changes, soil quality, the health of their crops and livestock, and the conditions of their machinery.

https://flex.com/intelligence/iot/old-macdonalds-new-connected-farm

Precision Agriculture



https://www.accenture.com/cn-en/insight-accenture-digital-agriculture-solutions

Precision Agriculture

Opportunities for vertical integration

https://www.cbinsights.com/research/agriculture-tech-market-map-company-list/

Connected Livestock

- Around 1.4 billion cattle around the world
- Animals can't tell you when they first get sick. It can be hard for humans to tell a cow is ill until there are visible signs of sickness.
- IoT sensors cannot diagnose an illness but it will let the farmer know when something needs attention.

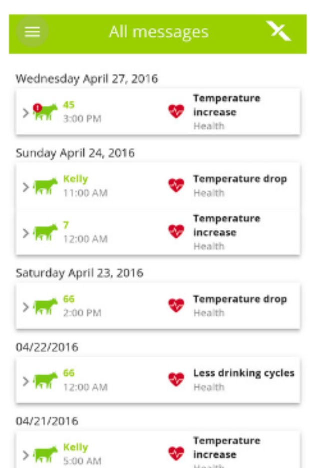
https://www.smaxtec.com/en/

Connected Livestock

- pH measurement
  - Early detection of fermentation disorders
- Activity level measurement
  - Early, automatic oestrus detection
  - Onset illness
- Temperature measurement
  - Early detection of onset of diseases such as feverish disorders, metabolic disorders, post-calving disorders
  - Early detection of start of calving

https://www.smaxtec.com/en/

Connected Livestock



http://uk.smaxtec.com/smaxtec-system/

Food Safety

- Wifi or cellular connectivity
- When a produce recall is initiated, the juice machine will check the packs and prevent the machine from pressing affected packs.

https://www.juicero.com/

Outline

- Smart manufacturing
- Smart packaging

Smart Manufacturing

- The use of IoT devices to improve efficiency and productivity of manufacturing operations. Typically, it involves retrofitting sensors to existing manufacturing equipment. But new manufacturing equipment often comes with IoT sensors pre-installed.
- According to IDC data, published early 2017, the manufacturing industry was good for a total IoT spending of $178 billion in 2016, which is more than twice as much the second largest vertical market, transportation.
- Manufacturing operations accounts for 57.5% of the total IoT spending on manufacturing.

http://www.businesswire.com/news/home/20170104005270/en/Internet-Spending-Forecast-Grow-17.9-2016-Led

Smart Packaging

- Packaging systems used with food and pharmaceutical that help extend shelf life, monitor freshness, display information on quality, improve safety, and improve convenience.
- Usually involve active functions beyond the inert passive containment, for example, the ability to sense or measure an attribute of the product, the inner atmosphere of the package, or the shipping environment. This information can be communicated to users or can trigger other active packaging functions.

Conclusion

http://www.eetimes.com/author.asp?doc_id=1328602

Introduction to Sensors

Professor Beth Pruitt

Stanford University

This modules provides an **Introduction to Sensors** and includes an introduction to sensors as transducers from physical parameters to signals; a review of sensor terminology; and discusses sensor selection including how to read a specification sheet. If you want more detail, check out the course **ENGR/ME220** which covers enough about all kinds of sensors so that you can make good decisions about recommending, selecting and using sensors in projects, products and research.

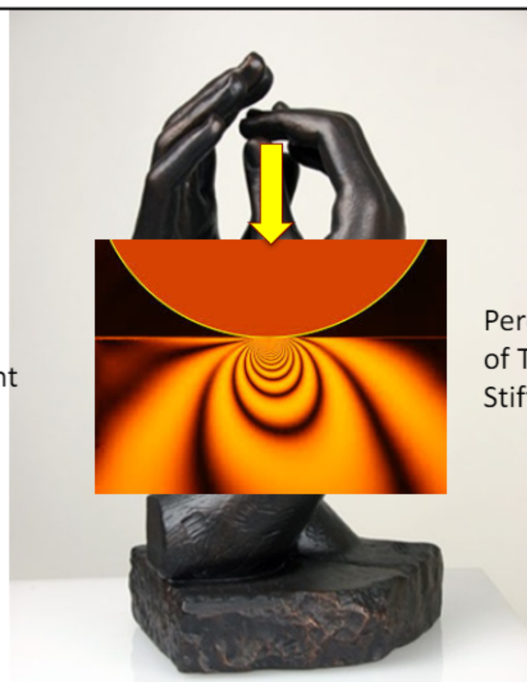
Learning Objectives

- Describe and define performance criteria for sensors (e.g., linearity, sensitivity, resolution, noise)
- Explain the operating mode for some common IoT transducers and sensors (strain gage, accelerometer, gyros, temperature, pressure sensors…)
- Interpret a specification sheet and extrapolate missing performance data

What is a sensor?

input -> Sensor -> output

Force or
Displacement



Perception
of Topography or
Stiffness

Rodin's Cathedral

external load

skin mechanics

stress/strain at neuron

neuron mechanics,
channel id & biophysics,
cell physiology

electrochemical response

neural circuit
past/current stimuli

behavioral response

SKIN: your largest and fastest growing organ

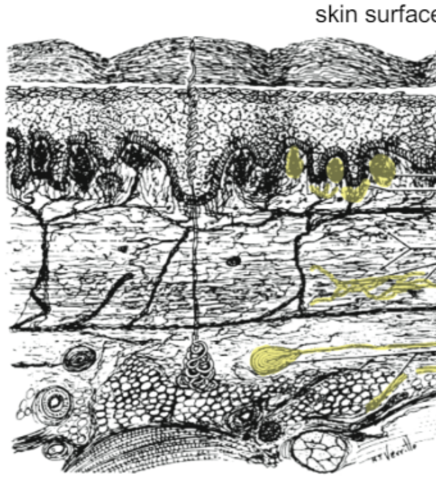
Barrier Layer:

- Respiration/perspiration
- Cooling

Sensory Function:

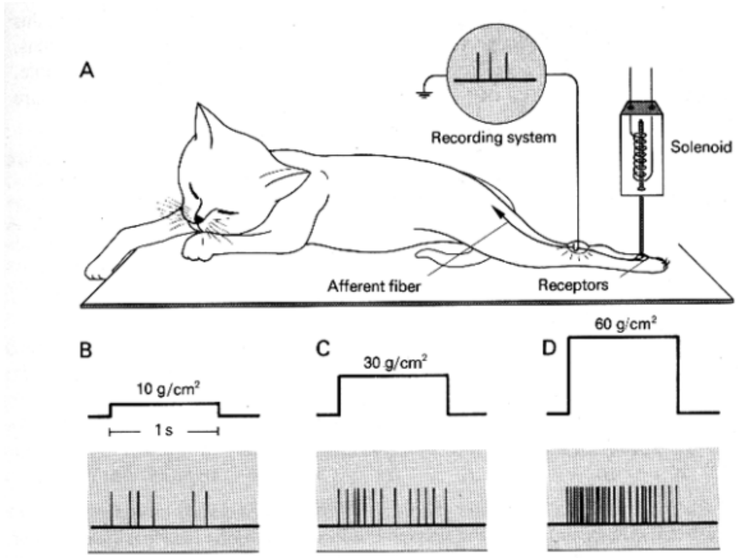
- Static Forces
- Dynamic Forces
- Temperature

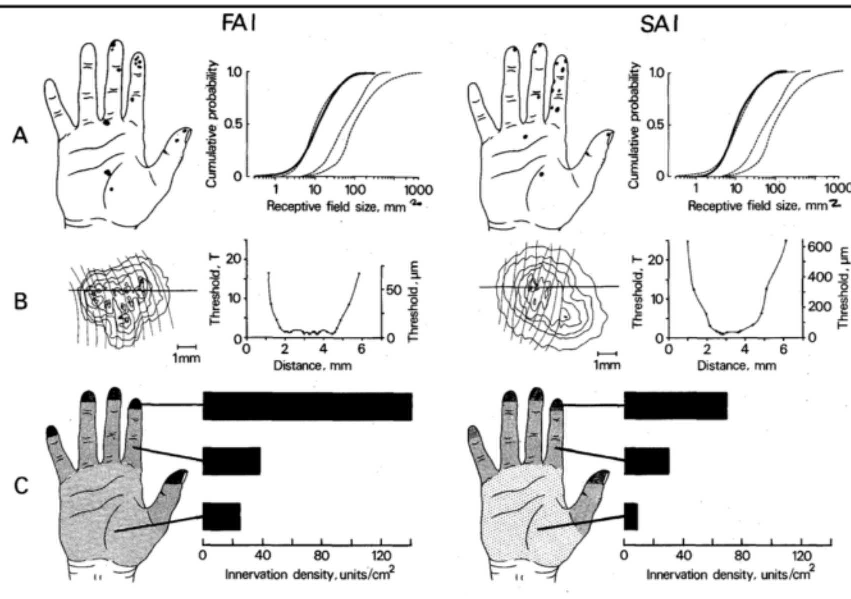
skin surface



Digital Signal Encoding : Amplitude coded by Pulse Density

Adaptation : Pulse density decays over time





Properties of cutaneous mechanoreceptors in the human hand related to touch sensation

Human Neurobiol (1984) 3:3–14

Å. B. Vallbo[1] and R. S. Johansson[2]*

[1]Nobel Institute for Neurophysiology, Karolinska Institutet, S-104 01 Stockholm, and [2]Department of Physiology, University of Umeå, S-901 87 Umeå, Sweden

Quantifiying touch sensation

- Dynamic Range – what range of force or deformation can you feel?
- Bandwidth – what frequency or on/off rate can you detect?
- Sensitivity – how strong is the output signal relative to the input? Is it the relationship linear?
- Resolution – what is the smallest feature or smallest force you can detect?

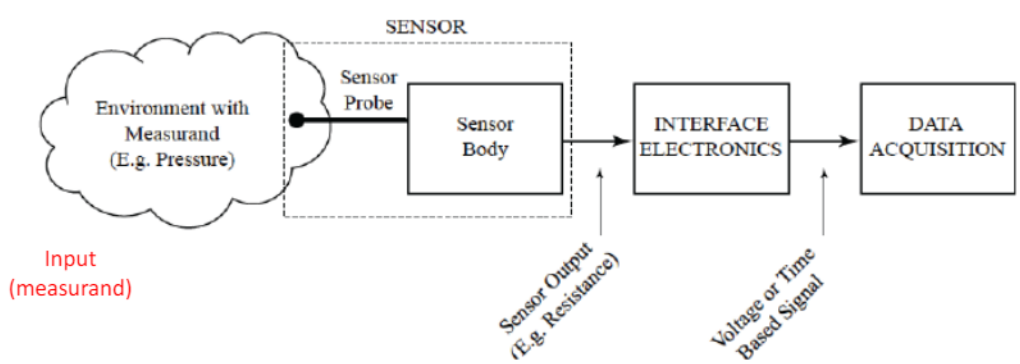
Sensor Terminology

Transducer, *n.* Any device by which variations in one physical quantity (e.g. pressure, brightness) are quantitatively converted into variations in another (e.g. voltage, position).

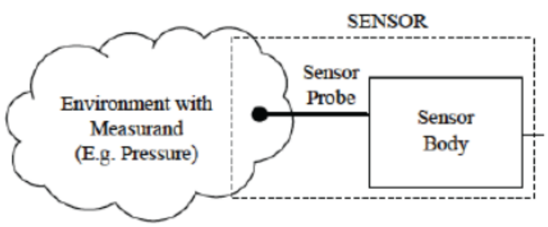
OR…a device for which changes in **input** quantity A produce corresponding, predictable changes in **output** quantity B

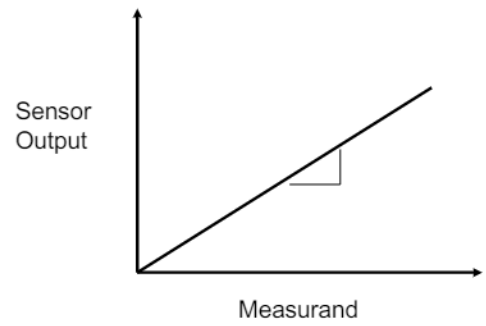
For our purposes, Sensors convert **physical** to **electrical** signals

Generic Sensing Application



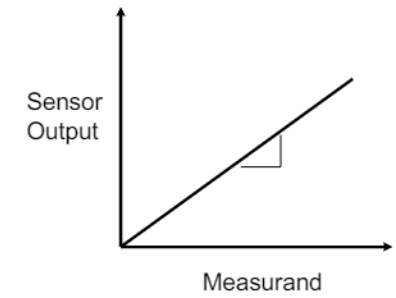
The Ideal Sensor

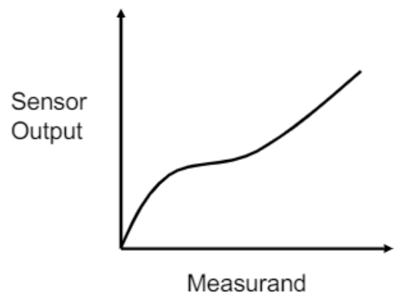




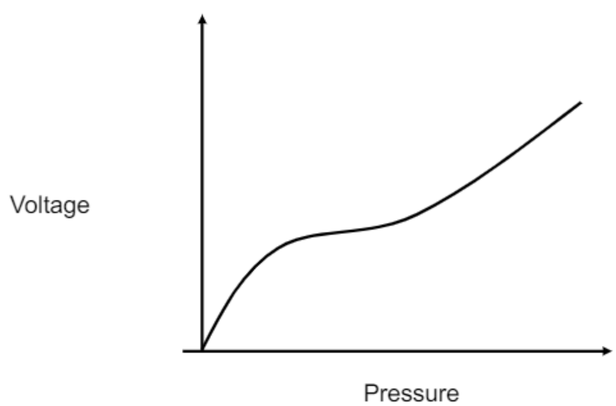
Sensor Specifications

Sensitivity

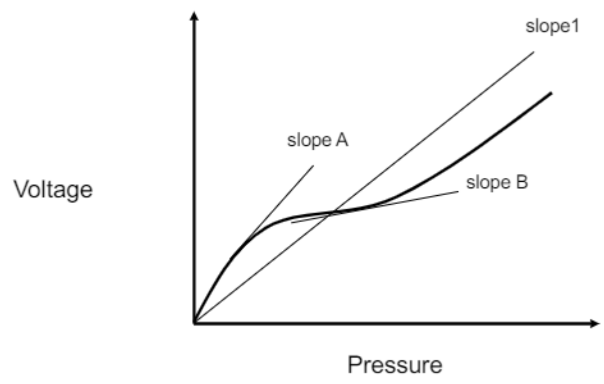




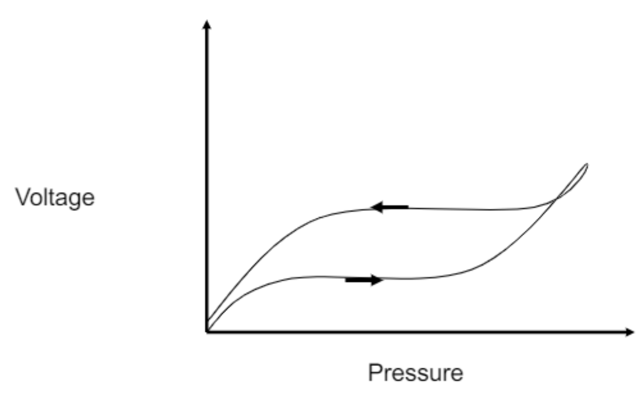
Non-Linearity



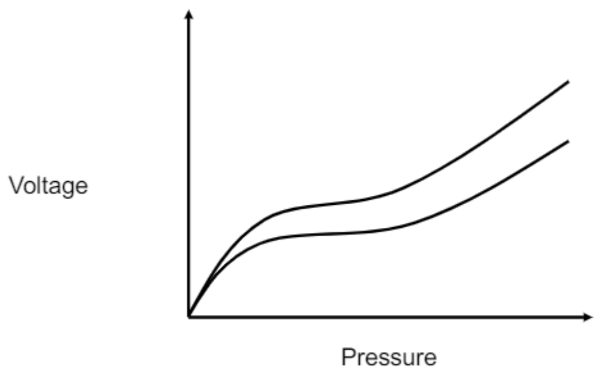
Non-Linearity



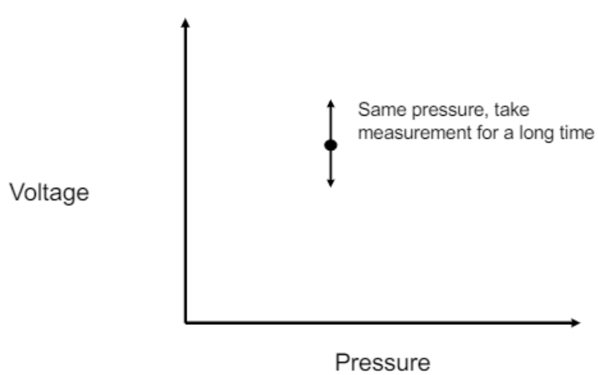
Hysteresis



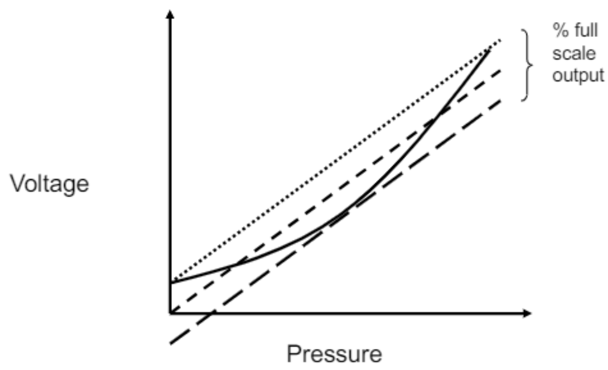
Repeatability



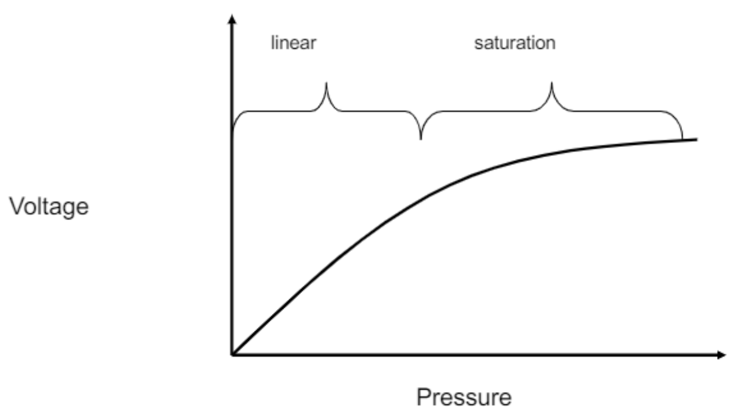
Drift



Specifying Non-Linearity



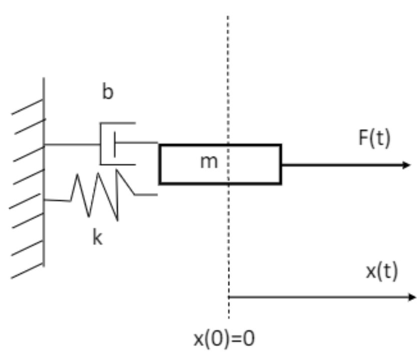
Span / Range



Sensor Dynamics

2^{nd} order Mechanical system

- Mass: F=ma
- Damper: F=bv
- Spring: F=kx



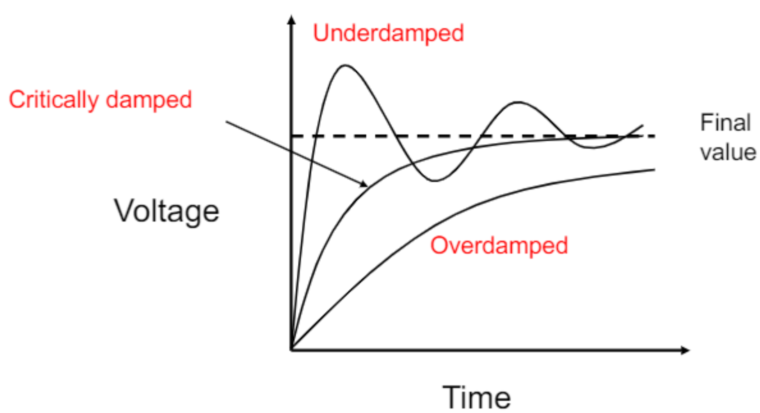
Transfer function & characteristic eqn

$$f(t) = m\frac{d^2x}{dt^2} + b\frac{dx}{dt} + kx$$

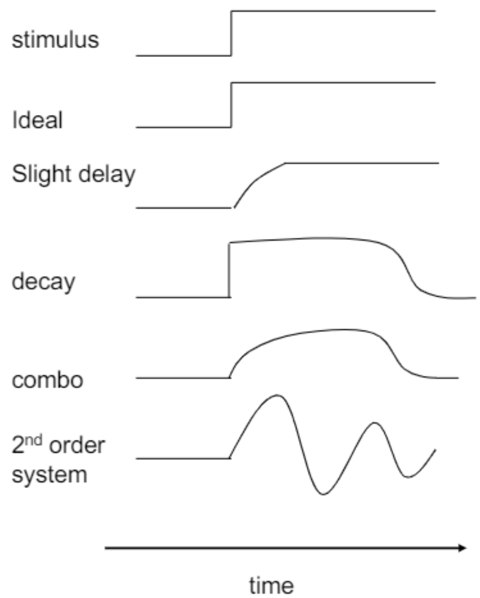
$$\frac{X(s)}{F(s)} = G(s) = \frac{1/m}{s^2 + {}^{b}\!/_{m}\, s + {}^{k}\!/_{m}}$$

$$\frac{X(s)}{F(s)} = G(s) = \frac{K\omega_n{}^2}{s^2 + 2\zeta\omega_n s + \omega_n{}^2}$$

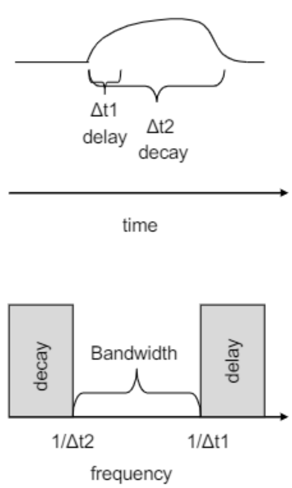
Step Response



Sensor Dynamics



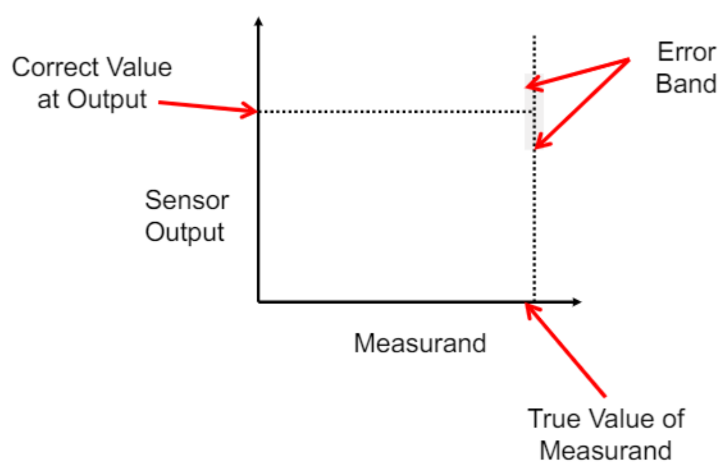
Bandwidth



Linearization and Error

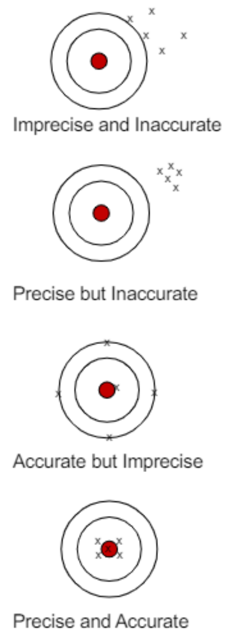
All models are wrong, some are useful.
- George Box

Overall Error (Error Band)

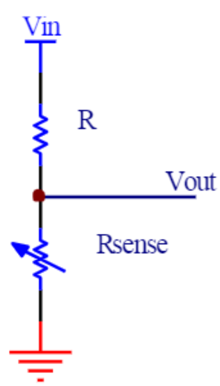


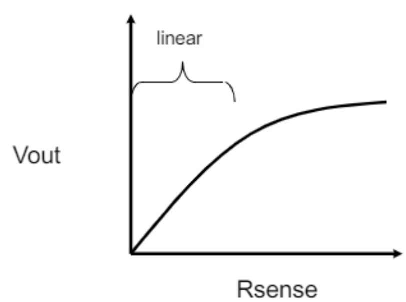
Accuracy vs. Precision

- Accurate
  - Average of sampled output is close to real value (AC errors and noise)
- Precise
  - Sampled output is consistently tightly grouped with consistent offset from real value (DC errors)



Linearity of a resistive sensor in a half bridge





$$Vout = Vin\left(\frac{Rsense}{Rsense + R}\right)$$

Linearity details

$$Vout = Vin\left(\frac{Rsense}{Rsense + R}\right) = V(R_s)$$

We're usually interested in the behavior for relatively small changes in Rs.

$$R_s = R_o + \Delta R$$

Linearity analysis using a Taylor expansion

$$Vout = Vin\left(\frac{Rsense}{Rsense + R}\right) = V(R_s)$$

We're usually interested in the behavior for relatively small changes in R_s so evaluate the linear (sensitivity) and non-linear (error) terms at an operating point – relative to starting value R_0

$$R_s = R_o + \Delta R$$

Recall that the Taylor series approximation of f(x) about operating point a is defined as:

$$f(x) = \underbrace{f(a)}_{\text{offset}} + \underbrace{f'(a)(x-a)}_{\text{Linear term}} + \underbrace{\frac{f''(a)}{2!}(x-a)^2 + \frac{f^{(3)}(a)}{3!}(x-a)^3 + \ldots + \frac{f^{(n)}(a)}{n!}(x-a)^n + \ldots}_{\text{Non-linear error terms}}$$

Linearity analysis using Taylor expansion

$$Vout = Vin\left(\frac{Rsense}{Rsense + R}\right) = V(R_s)$$

$$R_s = R_o + \Delta R$$

$$V(R_s) = V(R_o) + \Delta R\left[\frac{dV(R_s)}{dR_s}\right]_{Rs=Ro} + \frac{(\Delta R)^2}{2}\left[\frac{d^2V(R_s)}{dR_s^{\,2}}\right]_{Rs=Ro} + \frac{(\Delta R)^3}{6}\left[\frac{d^3V(R_s)}{dR_s^{\,3}}\right]_{Rs=Ro} \cdots$$

$$V(R_s) = V_{in}\underbrace{\frac{R_o}{R+R_o}}_{\text{offset}} + \underbrace{\Delta R\frac{R_o}{(R+R_o)^2}}_{\text{Linear term}} \underbrace{- \frac{(\Delta R)^2}{4}\frac{R_o}{(R+R_o)^3} + \frac{(\Delta R)^3}{36}\frac{R_o}{(R+R_o)^4}}_{\text{Non-linear error terms}}$$

offset

Linear term

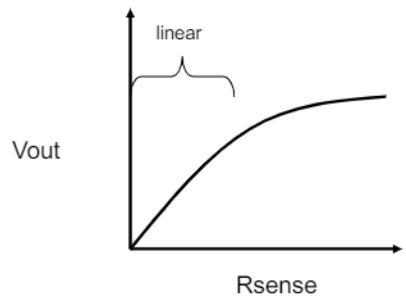
"Sensitivity"

Non-linear error terms

Decreasing weight with increasing exponent

Quadratic/linear = $\dfrac{\dfrac{(\Delta R)^2}{4}\dfrac{R_o}{(R+R_o)^3}}{\Delta R\dfrac{R_o}{(R+R_o)^2}} = \dfrac{\Delta R}{R + R_0}$

Why do we care so much about Linearity?



$$\iint v(a)\, dt^2 = \Delta x$$

$v = Sa + Ea^2$ (sensitivity and error terms)

$a = a_1 + a_2 \sin wt$ (accel signal plus accel noise)

$v = S(a_1 + a_2 \sin wt) + E(a_1^2 + a_1 a_2 \sin wt + a_2 \sin^2 wt)$

$1 - (\cos^2 wt)^2$

Vibration rectification – offset errors that add up and don't average out to zero

Noise

Electronic Noise Sources

- EMF - capacitive & inductive pickup
- Johnson noise
  - All resistors and dissipative systems
  - Thermal/Brownian random molecular interactions

$$V_J = \sqrt{4k_B TR}$$

- 1/f noise (shot, flicker, Hooge)
  - Semiconductor based electronics, amplifiers, instruments
  - Seminconductor resistors, Hooge noise

$$V_H = \sqrt{\frac{\alpha V_R^2}{Nf}}$$

- Drift
  - Accumulated offset errors
  - Very low frequency fluctuations?

Other electronic noise

- Shot noise
  - Associated with pn junctions

$$S_n(f) = 2q_e I_{DC}$$

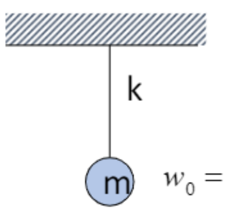
- Flicker noise (also 1/f)
  - Associated with trap charge states in diodes and FETs

$$S_n(f) = 2q_e I_{DC} + \frac{KI_{DC}}{f}$$

- Amplifier noise
  - Multiple factors, depends on op-amp type, see Senturia
  - Example: AD624 instrumentation amplifiers have
    - 4nV/rtHz voltage noise at low frequencies
    - 200fA/rtHz current noise above 10Hz

Intrinsic thermal noise

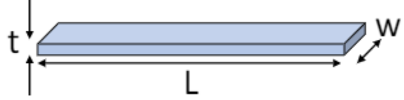
mechanical harmonic oscillator



$$\ddot{x} + \frac{w_0}{Q}\dot{x} + w_0^{\,2}x = \frac{1}{m}[F_{signal}(t) + F_{noise}(t)]$$

$$w_0 = \sqrt{\frac{k}{m}} \qquad S_{n\,force}(f) = \frac{4k_B Tk}{\omega_0 Q} \quad \left[m^2/Hz\right]$$

Cantilever

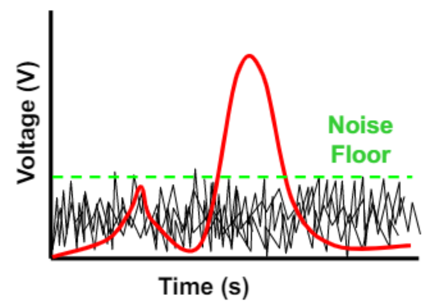


$$S_{n\,position}(f) = \frac{\dfrac{4k_B T}{k\omega_0 Q}}{\left(1 - \dfrac{\omega^2}{\omega_0^{\,2}}\right)^2 + \dfrac{\omega^2}{Q^2\omega_0^{\,2}}} \quad \left[m^2/Hz\right]$$

Quality factor, Q =(mk)$^{1/2}$/b, for a cantilever ~30-50 in air

Resolution

- Smallest signal the transducer can resolve or produce for a particular set of operating conditions
- Noise has some frequency distribution
- Signal to noise ratio (SNR)
- Depends on bandwidth



$$RESOLUTION = \frac{Noise\_Density}{Sensitivity}$$

$$MIN = \frac{Integrated\_Voltage\_Noise}{Sensitivity}$$

Scaling Issues for Force Sensors

- Noise generally does not improve with miniaturization
- Electrical Noise will be about same
- Energy Fluctuations ($k_B T$) will result in larger errors
  - Thermomechanical noise scales as $1/\sqrt{m}$
- Some 1/f noise becomes more significant
  - Increased surface/volume

Reading a Spec Sheet

Kionix®	± 2g Tri-axis Analog Accelerometer Specifications	PART NUMBER: KXSC7-1050 Rev. 6 Jul-2009

Product Description

The KXSC7-1050 is a Tri-axis, silicon micromachined accelerometer with a full-scale output range of +/-2g (19.6 m/s/s). The sense element is fabricated using Kionix's proprietary plasma micromachining process technology. Acceleration sensing is based on the principle of a differential capacitance arising from acceleration-induced motion of the sense element, which further utilizes common mode cancellation to decrease errors from process variation, temperature, and environmental stress. The sense element is hermetically sealed at the wafer level by bonding a second silicon lid wafer to the device using a glass frit. A separate ASIC device packaged with the sense element provides signal conditioning and self-test. The accelerometer is delivered in a 3 x 3 x 0.9mm Land Grid Array (LGA) plastic package operating from a 1.8 – 3.6V DC supply. The KXSC7 features a factory programmable low pass filter and an inertial wakeup interrupt.

Product Specifications

Table 1. Mechanical

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

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

Table 2. Electrical

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

Parameters		Units	Min	Typical	Max
Supply Voltage (V_{dd})	Operating	V	1.8	2.8	3.6
Current Consumption	Operating (full power)	µA	160	230	300
	Operating (low power)	µA		50	
	Standby	µA	-	-	5
Analog Output Resistance(R_{out})		kΩ	24	32	40
Power Up Time[1]		ms	-	5*R_{out}*C	-
Bandwidth (-3dB)[2]		Hz	40	50	60

Table 3. Environmental

Parameters		Units	Min	Typical	Max
Supply Voltage (V_{dd})	Absolute Limits	V	-0.3	-	6.0
Operating Temperature Range		ºC	-40	-	85
Storage Temperature Range		ºC	-55	-	150
Mech. Shock (powered and unpowered)		g	-	-	5000 for 0.5ms 10000 for 0.2ms
ESD	HBM	V	-	-	2000

0.1 mW to 1 mW power – also way too high for a phone

Data sheet

BNO055

Intelligent 9-axis absolute orientation sensor

Bosch Sensortec

BOSCH

Invented for life

Accelerometer features

- Programmable functionality — Acceleration ranges ±2g/±4g/±8g/±16g
  Low-pass filter bandwidths 1kHz - <8Hz
  Operation modes:
  - Normal
  - Suspend
  - Low power
  - Standby
  - Deep suspend
- On-chip interrupt controller — Motion-triggered interrupt-signal generation for
  - any-motion (slope) detection
  - slow or no motion recognition
  - high-g detection

Output Signal Accelerometer (Accelerometer Only Mode)

Parameter	Symbol	Condition	Min	Typ	Max	Units
Sensitivity	S	All g_{FSXg} Values, T_A=25°C		1		LSB/mg
Sensitivity tolerance	S_{tol}	Ta=25°C, g_{FS2g}		±1	±4	%
Sensitivity Temperature Drift	TCS	g_{FS2g}, Nominal V_{DD} supplies, Temp operating conditions		±0.03		%/K
Sensitivity Supply Volt. Drift	S_{VDD}	g_{FS2g}, T_A=25°C, $V_{DD\_min} \le V_{DD} \le V_{DD\_max}$		0.065	0.2	%/V
Zero-g Offset (x,y,z)	Off_{xyz}	g_{FS2g}, T_A=25°C, nominal V_{DD} supplies, over life-time	-150	±80	+150	mg
Zero-g Offset Temperature Drift	TCO	g_{FS2g}, Nominal V_{DD} supplies		±1	+/-3.5	mg/K
Zero-g Offset Supply Volt. Drift	Off_{VDD}	g_{FS2g}, T_A=25°C, $V_{DD\_min} \le V_{DD} \le V_{DD\_max}$		1.5	2.5	mg/V
Bandwidth	bw_8	2nd order filter, bandwidth programmable		8		Hz
	bw_{16}			16		Hz
	bw_{31}			31		Hz
	bw_{63}			63		Hz
	bw_{125}			125		Hz
	bw_{250}			250		Hz
	bw_{1000}			1,000		Hz
Nonlinearity	NL	best fit straight line, g_{FS2g}		0.5	2	%FS
Output Noise Density	n_{rms}	g_{FS2g}, T_A=25°C Nominal V_{DD} supplies Normal mode		150	190	μg/√Hz

Variable
Using onboard
Filters – choose
smalles BW
possible for app

Digital from onboard A/D

Similar offset to kionix

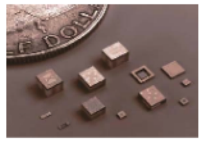
Reported as %Full scale output

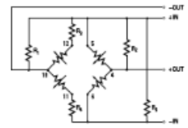
Note: spec sheets use definitions that give the best numbers!

Mechanical Characteristics Accelerometer

Parameter	Symbol	Condition	Min	Typ	Max	Units
Cross Axis Sensitivity	CAS	relative contribution between any two of the three axes		1	2	%
Alignment Error	E_A	relative to package outline		0.5	2	°

NPH Series Solid State Pressure Sensor (Low Pressure)





Features

- Solid State, High Reliability
- Standard TO-8 Package Suitable for PC Board Mount
- Low Cost, Small Size
- Available in Gage, Absolute, and Differential Pressure Versions
- Media Compatible with Noncorrosive Gases and Dry air
- Output Signal of 100mV @ 1.5mA
- Thermal Accuracy FSO 0.5% Typical
- Overpressure Capability to 5 Times Maximum Rated Pressure
- Three Standard Ranges: 0–10" H2O, 0–1 psi, and 0–5psi
- Nonlinearity 0.05% FSO Typical
- Standard 3/16" OD Pressure Port
- Ceramic Substrate with Temperature Compensation Resistors

Parameter	Min.	Typ.	Max.	Units	Notes
		2.5 kPa			
Performance Parameters (7), Compensated (1)					
Offset	–8	± 2	8	mV	
Full Scale Output					
2.5 kPa	25	50	90	mV	2
7 kPa				mV	2
30 kPa				mV	2
Linearity	–1.0	0.1	1.0	%FSO	3
Hysteresis and					
Repeatability	–0.2	0.05	0.2	%FSO	
Thermal					
Accuracy of Offset	–3	0.5	3	%FSO	4
Thermal					
Accuracy of FSO	–3	–1	3	%FSO	4
Thermal Hysteresis	–0.75	±0.5	0.75	%FSO	5
Short-Term Stability of Offset		5		µV/V	6, 11
Short-Term Stability of FSO		5		µV/V	6, 11

Notes: 1. Performance with offset, thermal accuracy of offset, and thermal accuracy of FSO compensation resistors. 2. FSO with 1.5mA input excitation. 3. Best fit straight line. 4. 0 to +70°C with reference to 25°C. 5. 0 to +70°C, by design. 6. Normalized offset/bridge voltage —100 hrs, typical value, not tested in production. 7. All values measured at 25°C and at 1.5mA, unless otherwise noted. 8. Reduced performance outside compensation range. 9. Backside differential tube is nickel or Kovar. 10. Top side pressure. 11. Typical specifications are for reference only; absolute values may vary.

measurement SPECIALTIES

30K6A1W Series III & IV Thermistor

- Thermally conductive epoxy coating
- Choice of 30 AWG or 28 AWG tin plated copper leads
- Available in custom probe assemblies
- Available in custom tolerances
- RoHS Compliant

DESCRIPTION

The BetaCURVE Chip is Soldered to Tin Plated Copper Leads and Encapsulated in Stycast Epoxy Resin

FEATURES

- Interchangeability
- Proven Stability and Reliability
- Rapid time response
- Alloy lead wires for reduced thermal conductivity ("stem effect").
- Thermally Conductive Epoxy Coating
- Temperature range -40 ºC to +125ºC

APPLICATIONS

- Temperature sensing, control and compensation
- Tight Tolerance Instrumentation
- Assembly into probes for a wide variety of applications
- General Instrumentation applications

PERFORMANCE SPECS

Parameters	Units	Value
Resistance @ +25°C	Ohms	30,000
30K6A1W2 Resistance tolerance from 0°C to +70°C	°C	0.2
30K6A1W3 Resistance tolerance from 0°C to +70°C	°C	0.2
Alpha Value @ 25°C	%/°C	4.68
Beta Value 25/85	K	4261
Tolerance on Beta Value 25/85	%	0.5
Time response in Liquids	Seconds	<1.5
30K6A1W2 Dissipation Constant in still air	mW/°C	2
30K6A1W3 Dissipation Constant in still air	mW/°C	3

RESISTANCE V TEMPERATURE TABLE

Temp. °C	Ohms	Temp. °C	Ohms	Temp. °C	Ohms	Temp. °C	Ohms
-40	1219114	-20	331871	0	105305	20	38019
-39	1137677	-19	312291	1	99787	21	36240
-38	1062161	-18	293977	2	94588	22	34554
-37	992104	-17	276841	3	89688	23	32955
-36	927079	-16	260801	4	85069	24	31439
-35	866698	-15	245781	5	80712	25	**30000**
-34	810604	-14	231711	6	76603	26	28635
-33	758469	-13	218526	7	72725	27	27339
-32	709991	-12	206164	8	69064	28	26108
-31	664894	-11	194572	9	65608	29	24939
-30	622924	-10	183696	10	62343	30	23828
-29	583847	-9	173489	11	59258	31	22773
-28	547448	-8	163905	12	56342	32	21770
-27	513528	-7	154905	13	53585	33	20816
-26	481906	-6	146449	14	50978	34	19909
-25	452413	-5	138501	15	48512	35	19047
-24	424895	-4	131028	16	46178	36	18225
-23	399208	-3	124000	17	43969	37	17444
-22	375221	-2	117388	18	41877	38	16700
-21	352813	-1	111164	19	39896	39	15992

Picking sensors

Automotive Applications:

- What might you want to measure for each?
- What sensors might you use?
- How will you use the data?

1. Seat occupancy
2. Airbag deployment
3. Tire pressure monitoring system
4. Cruise control

1. Seat occupancy

2. Airbag deployment

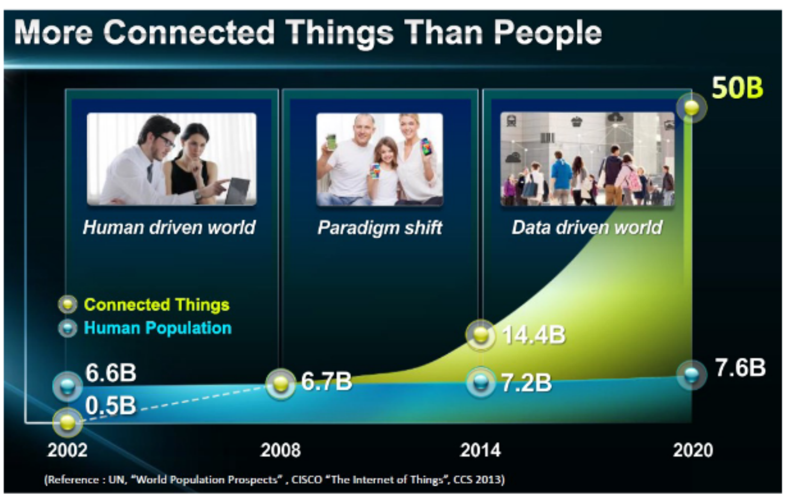
3. Tire pressure monitoring system

4. Cruise control

IoT Circuits

Boris Murmann
murmann@Stanford.edu

The data-driven world



Kim, 2015 International Solid-State Circuits Conference (ISSCC), Keynote Talk

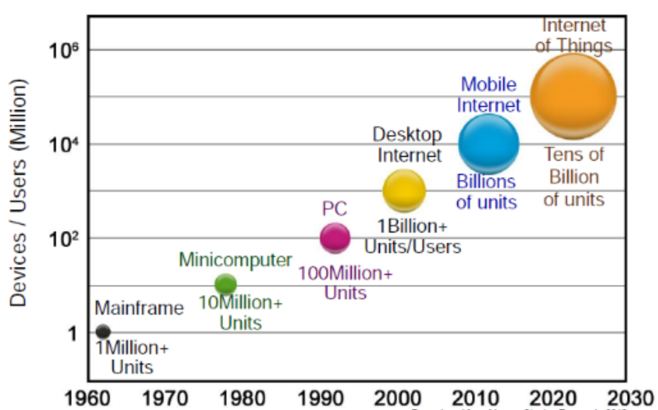
Explosion in wireless connectivity

TODAY:
10 Radios/Person
70 Billion Radios

2030:
100 Radios/Person
800 Billion Radios
$100B+ Silicon opportunity

Tsai, 2014 International Solid-State Circuits Conference (ISSCC), Keynote Talk

IoT will be the "next big thing"



Major Technology Cycles = 10x More Users & Devices
Driven by: 1) Lower Price, 2) Improved Functionality & Services

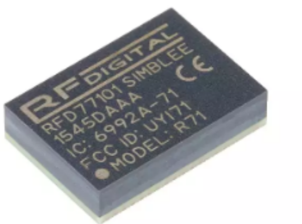
Tsai, 2014 International Solid-State Circuits Conference (ISSCC), Keynote Talk

Today's hardware: Bluetooth/ZigBee/WIFI/… modules

Simblee™

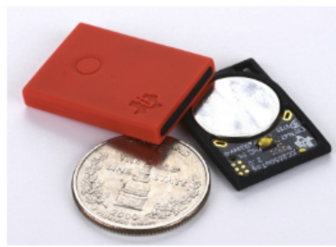
A subsidiary of HEPTAGON™

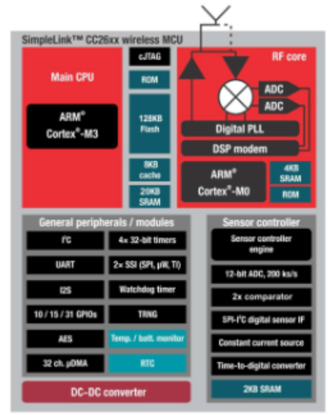
IoT for connecting
Everyone and Everything
IoT4EE



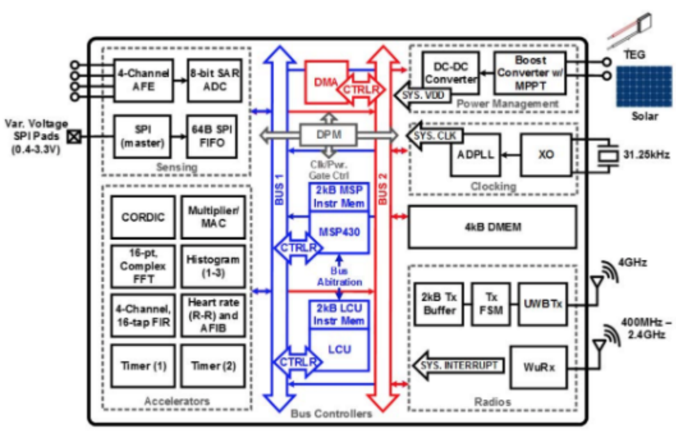
TI Designs
SimpleLink™ CC2650 uTag–Ultra-Compact Bluetooth® Smart Reference Design

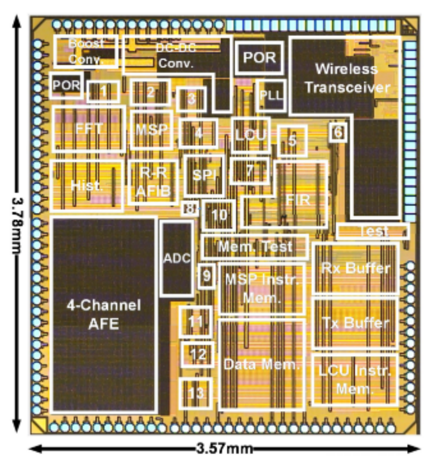
TEXAS INSTRUMENTS





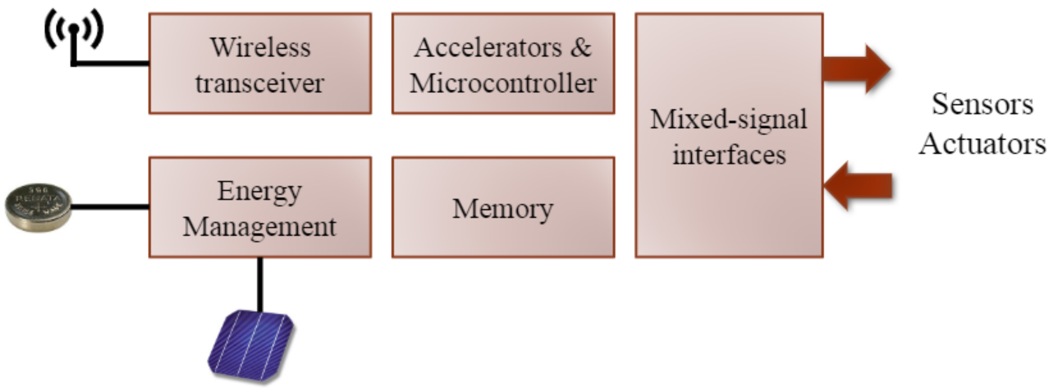
Under the hood of an IoT chip





A. Klinefelter et al., "A 6.45µW self-powered IoT SoC with integrated energy-harvesting power management and ULP asymmetric radios," 2015 IEEE International Solid-State Circuits, pp. 384-385, Feb. 2015.

Generic block diagram



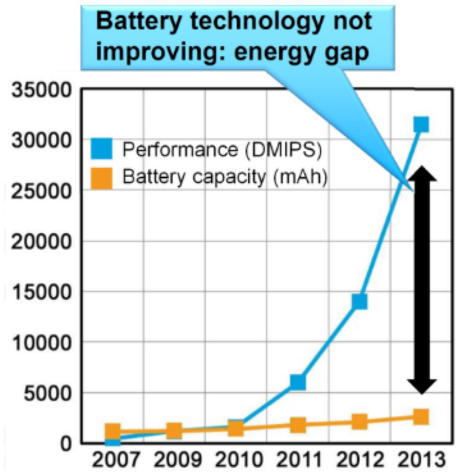
Batteries

Type	2x AAA	CR2032	CR123A	CR2
Material	Alkaline	LiMnO2*	Lithium	Lithium
Voltage	3 V	3 V	3 V	3 V
Capacity	1000 mAh	225 mAh	1500 mAh	800 mAh
Diameter	10.5 mm (x2)	20 mm	17 mm	15.6 mm
Height	45 mm	3.2 mm	34 mm	27 mm
Weight	24 g	3 g	17 g	11 g

Lithium Manganese Dioxide

http://www.silabs.com/whitepapers/battery-life-in-connected-wireless-iot-devices

Batteries improve slowly



- Semiconductor chip performance tends to improve rapidly
  - See e.g. the blue curve on the left for microprocessors
  - DMIPS = Dhrystone Million Instructions per Second
- Battery capacity improves slowly
  - About 5-8% per year
  - Doubling in ~10 years
- **Key challenge: How to fill this gap via creative circuit & system design?**

Tsai, 2014 International Solid-State Circuits Conference (ISSCC), Keynote Talk

Module outline

- Battery capacity (exercise)
- Energy management
- Wireless link
- Digital computing
- Analog-digital interfaces

Exercise

- Estimate the average current that be sustained for a battery life of 10 years

Battery	2 x AA	CR2032
Capacity	1000 mAh	225 mAh
Average current (for 10-year lifetime)	?	?

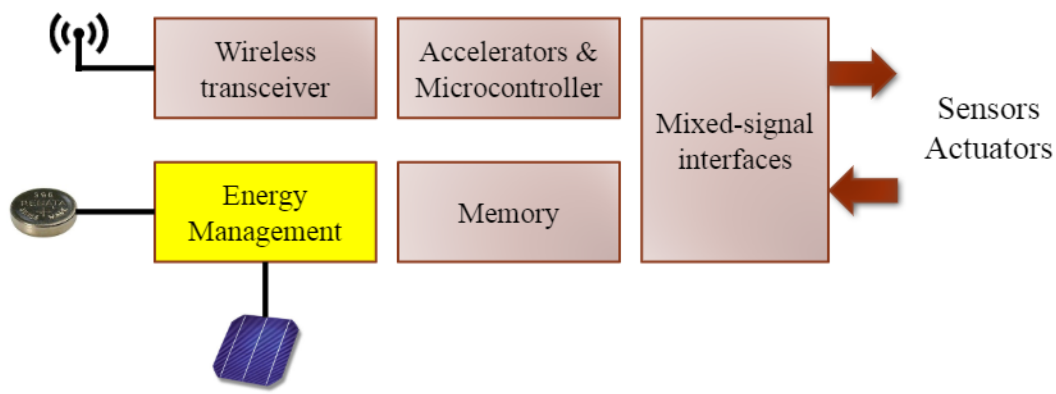
Solution

- 10 years = 87,600 hours

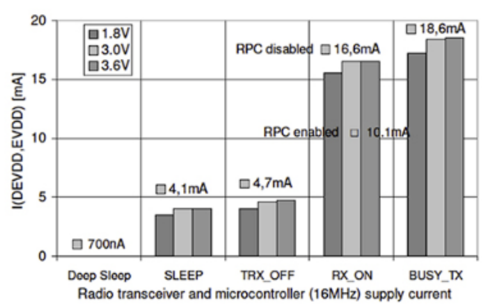
Battery	2 x AA	CR2032
Capacity	1000 mAh	225 mAh
Average current (for 10-year lifetime)	**11.4 μA**	**2.6 μA**

- Only a very small amount of current available!

Energy management



Typical current consumption across modes

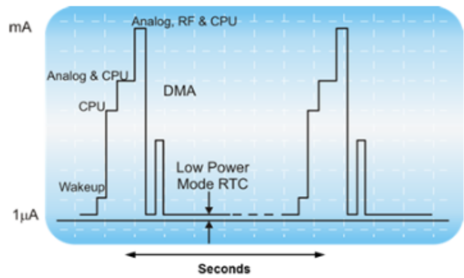


Atmel ATmega2564RFR2 wireless MCU power/sleep mode profiles

- How to achieve average current consumption of ~10 μA (or less)?

https://www.digikey.com/en/articles/techzone/2016/apr/achieving-wireless-connectivity-for-iot-applications-in-a-power-budget-efficient-way

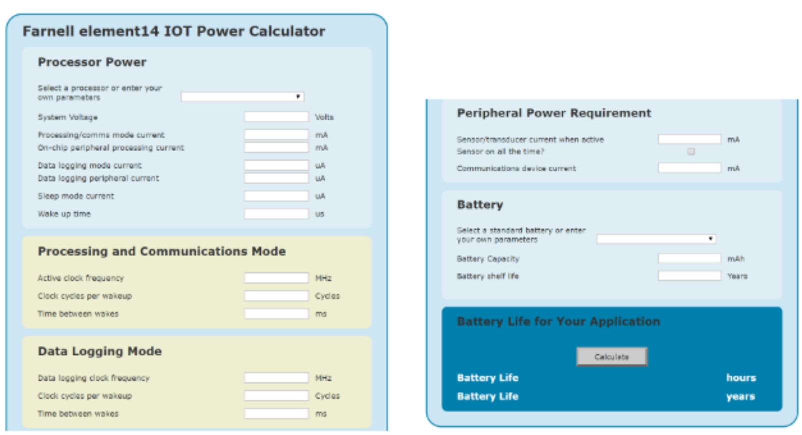
Key: Duty-cycling



- IoT transceivers are "mostly" off
- Example: Receive/transmit within 1 ms, then go to sleep for 1 sec

http://www.mouser.com/applications/low-power-ewc-design/

Battery lifetime calculation becomes complicated….



http://uk.farnell.com/calculating-battery-life-in-iot-applications

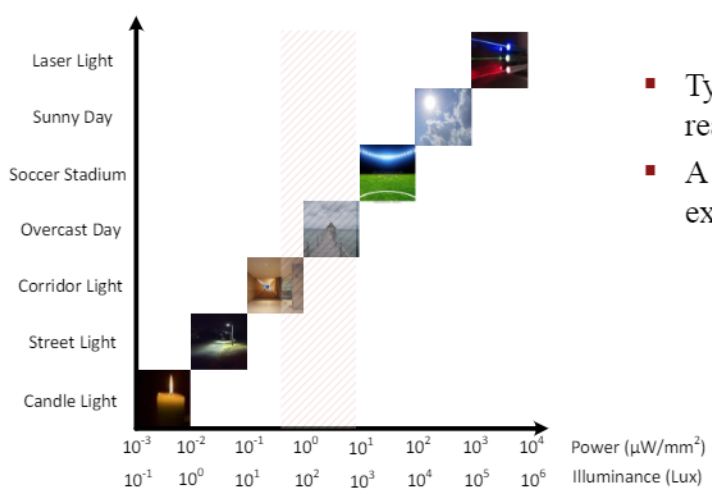
Extending battery life (or eliminate batteries!) through harvesting

Source	Limitation	Power Density
Inductive Coupling	Short range (cm), inefficient	∝ D, Q, $1/d^3$
Far-field RF	Base station (a few m), safety	∝ $1/d^2$
Solar (Indoor)	Available artificial lighting	10 μW/cm²
Solar (Outdoor)	Direct sunlight	10,000 μW/cm²
Vibration	Relatively constant movement	4 μW/cm²
Thermoelectric	Thermal gradient	25 μW/cm²

Most popular: Solar (Indoor), Solar (Outdoor)

A. Klinefelter et al., "A 6.45μW self-powered IoT SoC with integrated energy-harvesting power management and ULP asymmetric radios," 2015 IEEE International Solid-State Circuits, pp. 384-385, Feb. 2015.

How much solar power can we pick up?



- Typically expect 1 $\mu W/mm^2$ under reasonable lighting conditions
- A few cm^2 of solar cells may help extend the lifetime to "infinity"

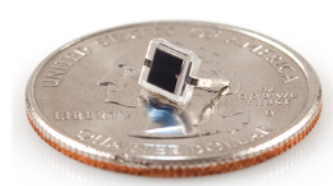
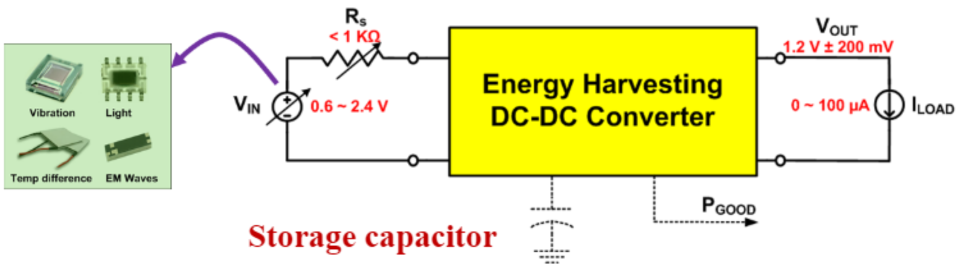


Chart by Nishit Shah, Stanford

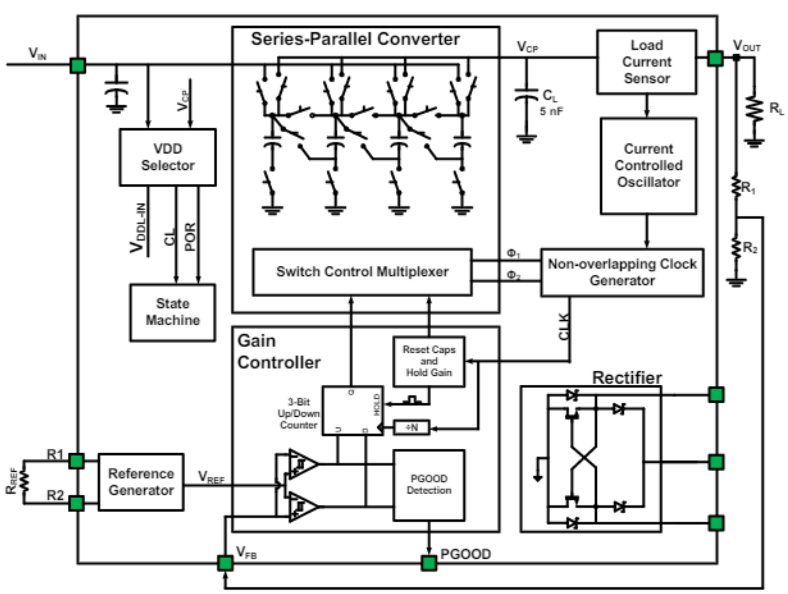
DC-DC converters for energy harvesting



Storage capacitor

- Want to cover a variety of harvesting sources → Large input voltage range
- Want relatively stable output voltage
- Need efficient management of storage capacitor

M. Saadat and B. Murmann, "A 0.6V - 2.4V Input, Fully Integrated Reconfigurable Switched-Capacitor DC-DC Converter for Energy Harvesting Sensor Tags," in Proc. IEEE Asian Solid-State Circuits Conf., Nov. 2015.

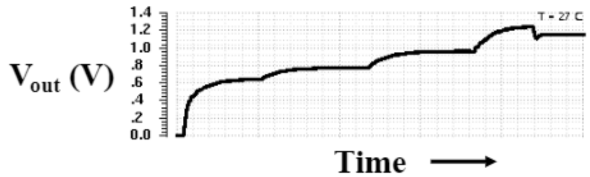


- Example design
- Switched-capacitor series-parallel converter provides 8 gain rations
- Controller adjusts frequency for optimum efficiency
- Provides step charging of storage capacitance

M. Saadat and B. Murmann, "A 0.6V - 2.4V Input, Fully Integrated Reconfigurable Switched-Capacitor DC-DC Converter for Energy Harvesting Sensor Tags," in Proc. IEEE Asian Solid-State Circuits Conf., Nov. 2015.

Step charging example

$$C_{store} = 100nF$$
$$V_{in} = 0.8V$$
$$V_{out} = 1.3V$$



Energy loss without step charging:

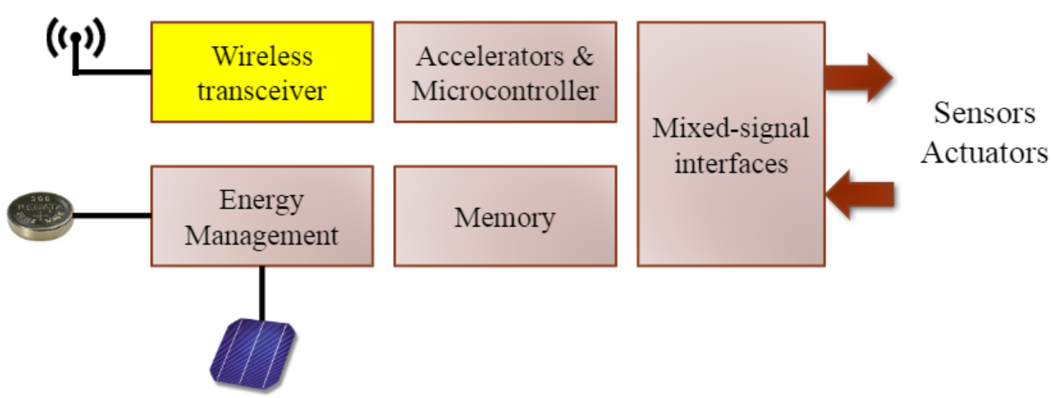
$$E_{loss} = \frac{1}{2} C_{store} V_{out}^2 = 192\ nJ$$

Energy loss with step charging:

$$E_{loss} = \sum_k \frac{1}{2} C_{store} \left(V_{out,k} - V_{out,k-1}\right)^2 = 18\ nJ$$

M. Saadat and B. Murmann, "A 0.6V - 2.4V Input, Fully Integrated Reconfigurable Switched-Capacitor DC-DC Converter for Energy Harvesting Sensor Tags," in Proc. IEEE Asian Solid-State Circuits Conf., Nov. 2015.

Wireless transceiver



Power/energy cost of wireless transmission

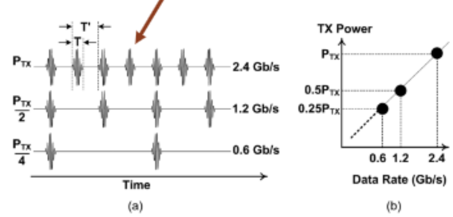
	Bluetooth LE	Nike+	ZigBee	WIFI
Power (mW)	0.147	0.675	35.7	210
Bits/sec	960	272	192	40M
Energy/bit (nJ)	153	2480	186,000	5.25

- Numbers are a strong function of reach
- Energy per bit tends to be lower for high-data rate links → Invested power amortizes better

https://www.digikey.com/en/articles/techzone/2011/aug/comparing-low-power-wireless-technologies

Pushing the limits

60 GHz pulses

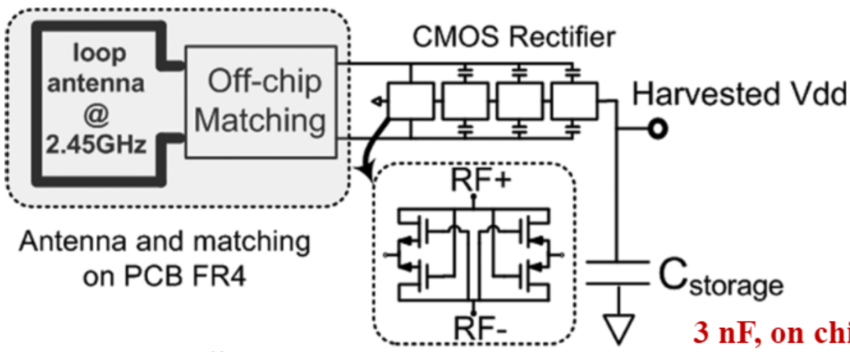


Power proportional to data rate

- A variety of non-standard, short-range radios are being investigated for much lower energy/bit numbers
- This open the door for RF powering (no batteries, no solar cells)!

M. Taghivand, K. Aggarwal, Y. Rajavi and A. S. Y. Poon, "An Energy Harvesting 2x60 GHz Transceiver With Scalable Data Rate of 38–2450 Mb/s for Near-Range Communication," in IEEE Journal of Solid-State Circuits, vol. 50, no. 8, pp. 1889-1902, Aug. 2015.

RF harvesting circuit



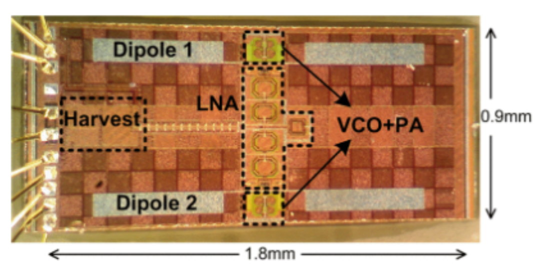
3 nF, on chip

- Supplies up to 1 mA at 5-10 cm distance
- RF power can be supplied by cell phone

M. Taghivand, K. Aggarwal, Y. Rajavi and A. S. Y. Poon, "An Energy Harvesting 2x60 GHz Transceiver With Scalable Data Rate of 38–2450 Mb/s for Near-Range Communication," in IEEE Journal of Solid-State Circuits, vol. 50, no. 8, pp. 1889-1902, Aug. 2015.

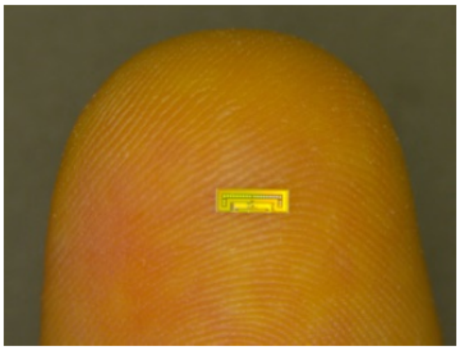
Comparison of recent works

	[41] VLSI 2010	[6] ISSCC 2012	[22] JSSCC 2010	[20] RFIC 2013	This Work	
Technology	90nm	65nm	90nm	65nm SOI	40nm GP	
Carrier Frequency (GHz)	60	60	60	60	60	
Modulation	FSK	QPSK	OOK-NRZ	OOK-RZ	OOK-RZ	
Antenna	On-board, folded dipole	Bond wire	On-board, folded dipole	On-chip, folded dipole	On-chip, dipole	
Data Rate (Mb/s)	2000	2620	3000	500 to 2200	38 to 2450	
Distance for max Data Rate (cm)	41	5	4.5	4	5	10
BER	$<10^{-12}$	NA	10^{-3}	10^{-5}	5×10^{-4}	5×10^{-5}
Power Consumption (mW)	TX: 280 RX 150	TX: 160 RX: 233	TX: 183 RX: 103	TX+RX: 98	TX: 0.10-6.26 RX: 74	TX: 0.26-11.9 RX: 74
Energy/bit for max Data Rate (pJ/bit)	215	150	95	45	32.8	35.1
Tx Energy/bit (pJ/bit)	140	61	61	NA	2.6-2.56	6.8- 4.9
Die Area (mm^2)	1.26	2.86	1.1	5.89	1.62	
TX×RX /Shared Antenna	1×1 / No	1×1 / Yes	1×1 / No	1×1 / No	2×2 / Yes	
Energy Harvesting	No	No	No	No	Yes	
ScalableTX Power	No	No	No	Yes	Yes	
RX/TX Non-Coherence Tolerant	Yes	No	Yes	Yes	Yes	



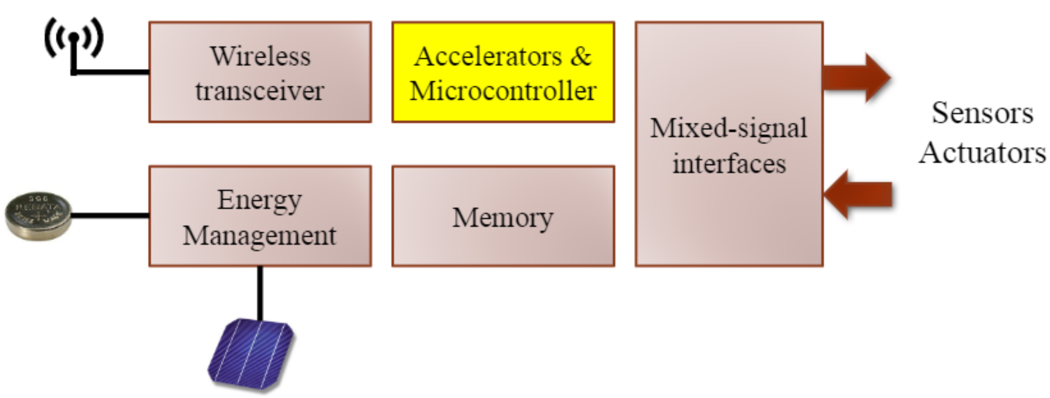
M. Taghivand, K. Aggarwal, Y. Rajavi and A. S. Y. Poon, "An Energy Harvesting 2x60 GHz Transceiver With Scalable Data Rate of 38–2450 Mb/s for Near-Range Communication," in IEEE Journal of Solid-State Circuits, vol. 50, no. 8, pp. 1889-1902, Aug. 2015.

Ant-sized radio



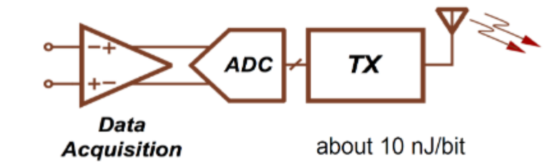
https://web.stanford.edu/~arbabian/Home/IoT_Radio.html

Digital circuits

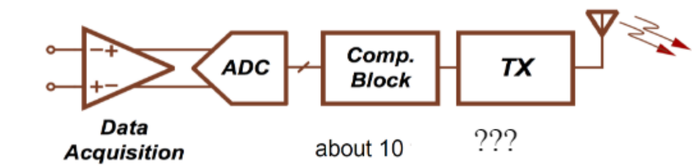


To transmit or not to transmit?

Sense → Communicate:



Sense → Compute → Communicate:



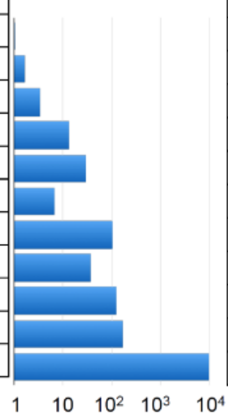
- Sending bits is expensive
- How about adding some signal processing to reduce number of bits?
- For example, how about data compression?

Drawing by Peter Kairouz, Stanford

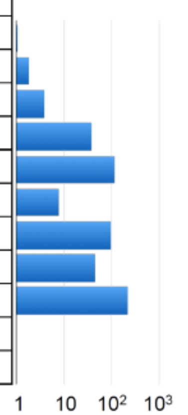
Typical energy of digital operations (45nm CMOS)

Operation:	Energy (pJ)	Area (μm^2)
8b Add	0.03	36
16b Add	0.05	67
32b Add	0.1	137
16b FP Add	0.4	1360
32b FP Add	0.9	4184
8b Mult	0.2	282
32b Mult	3.1	3495
16b FP Mult	1.1	1640
32b FP Mult	3.7	7700
32b SRAM Read (8KB)	5	N/A
32b DRAM Read	640	N/A

Relative Energy Cost

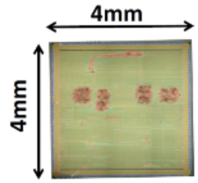


Relative Area Cost

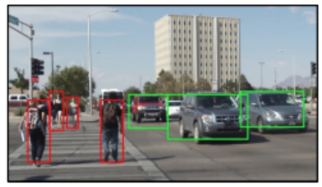


V. Sze, Y.-H. Chen, J. Emer, A. Suleiman, Z. Zhang, "Hardware for Machine Learning: Challenges and Opportunities," IEEE Custom Integrated Circuits Conference (CICC), Invited Paper, May 2017.

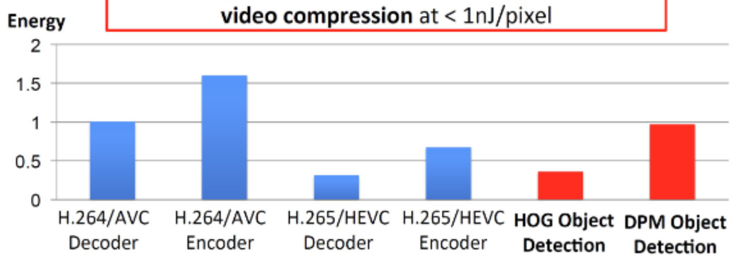
Energy of image processing accelerators



MIT Object
Detection Chip
[VLSI 2016]



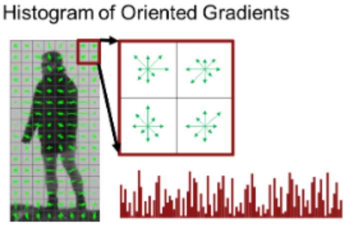
Enable **object detection** to be as **energy-efficient** as **video compression** at < 1nJ/pixel

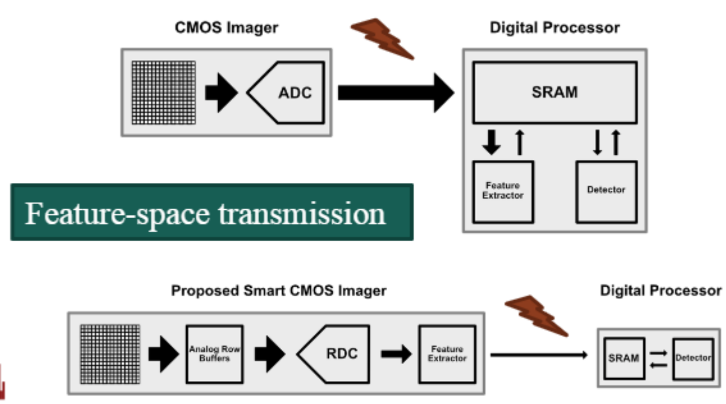


V. Sze, Y.-H. Chen, J. Emer, A. Suleiman, Z. Zhang, "Hardware for Machine Learning: Challenges and Opportunities," IEEE Custom Integrated Circuits Conference (CICC), Invited Paper, May 2017.

Research: Always-on object detection

Histogram of Oriented Gradients





20x less data

A. Omid-Zohoor, C. Young, D. Ta and B. Murmann, "Towards Always-On Mobile Object Detection: Energy vs. Performance Tradeoffs for Embedded HOG Feature Extraction," to appear, IEEE Trans. Circuits and Systems for Video Technology.

Microcontroller power modes

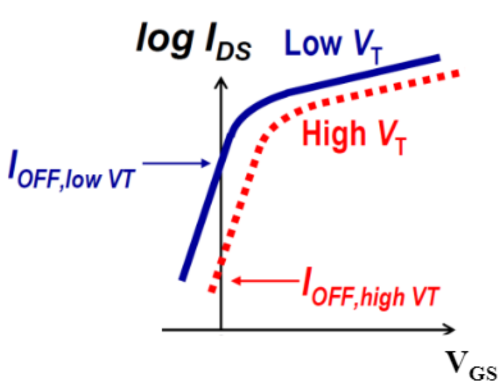
PARAMETER		TEST CONDITIONS	MIN	TYP	MAX	UNIT
I_{core}	Core current consumption	Reset. RESET_N pin asserted or VDD below Power-on-Reset threshold		100		nA
		Shutdown. No clocks running, no retention		150		
		Standby. With RTC, CPU, RAM and (partial) register retention. RCOSC_LF		1		µA
		Standby. With RTC, CPU, RAM and (partial) register retention. XOSC_LF		1.2		
		Standby. With Cache, RTC, CPU, RAM and (partial) register retention. RCOSC_LF		2.5		
		Standby. With Cache, RTC, CPU, RAM and (partial) register retention. XOSC_LF		2.7		
		Idle. Supply systems and RAM powered.		550		
		Active. Core running CoreMark		1.45 mA + 31 µA/MHz		
		Radio RX		6.2		mA
		Radio TX, 0-dBm output power		6.8		
		Radio TX, 5-dBm output power		9.4		

Critical for long battery life

Core Power Consumption of TI CC2650MODA

http://uk.farnell.com/calculating-battery-life-in-iot-applications

Why can't we reduce leakage current to zero?

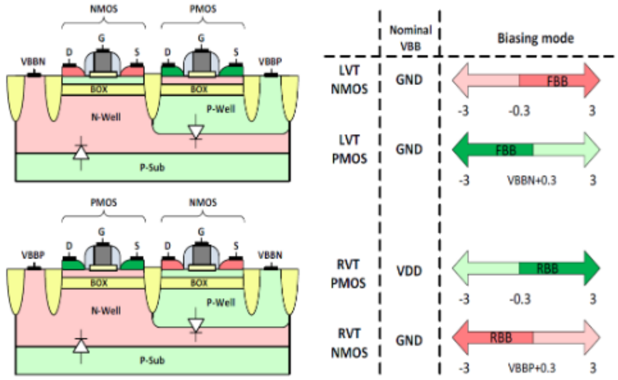


- MOS transistors turn off gradually
- Current drops exponentially below threshold voltage (V_T)

- Two options for reducing leakage
  - Increase V_T → But this reduces on-current
  - Increase subthreshold slope → very difficult; bounded by 60 mV/decade (kT/q)

Body biasing in fully depleted silicon-on-insulator (FD-SOI)

Low VT (LVT) CMOS in FD-SOI; flipped-well



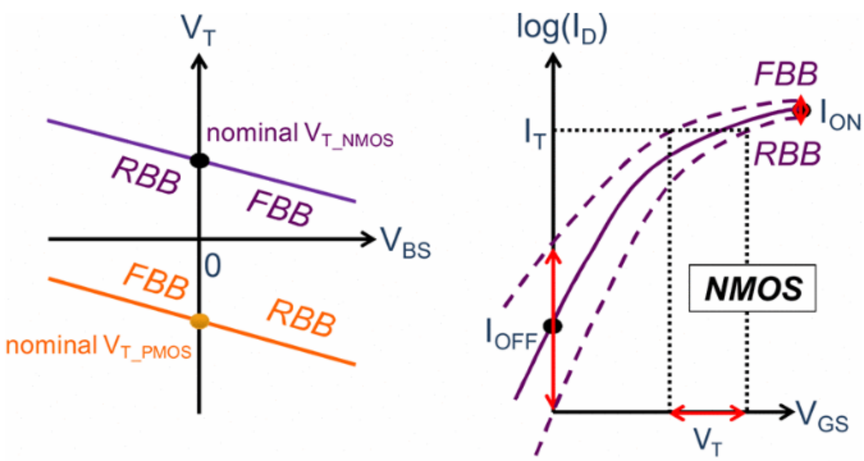
FBB = Forward body biasing

RBB = Reverse body biasing

Regular VT (RVT) CMOS in FD-SOI

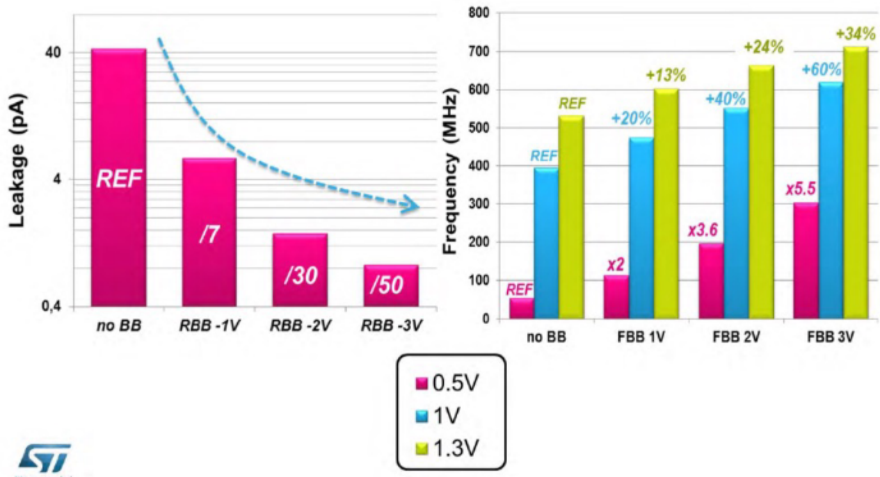
Courtesy Andreia Cathelin, STMicroelectronics

V_T changes with body biasing



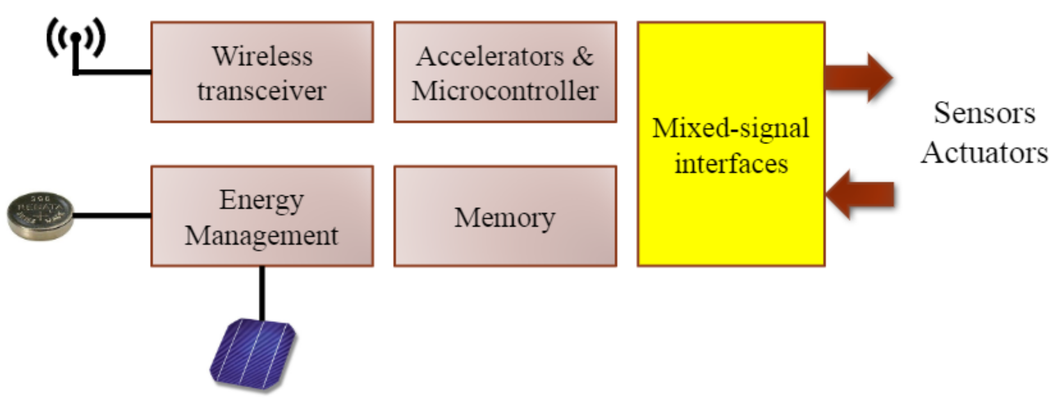
E. Beigne et al., "Ultra-Wide Voltage Range designs in Fully-Depleted Silicon-On-Insulator FETs," 2013 Design, Automation & Test in Europe Conference & Exhibition (DATE), Grenoble, France, 2013, pp. 613-618.

Impact on leakage and drive strength (frequency)

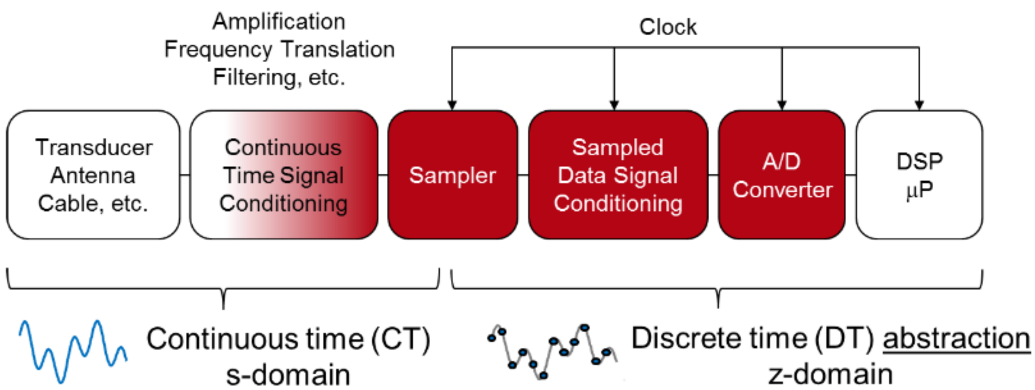


Courtesy Philippe Flatresse, STMicroelectronics

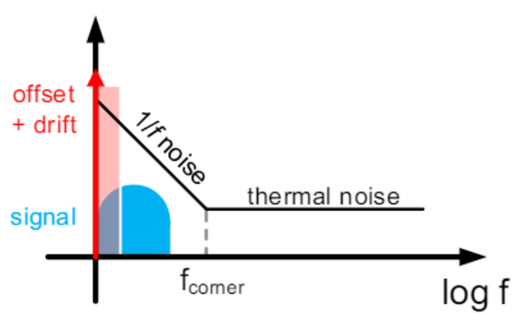
Mixed-signal interfaces



Anatomy of an analog-digital interface

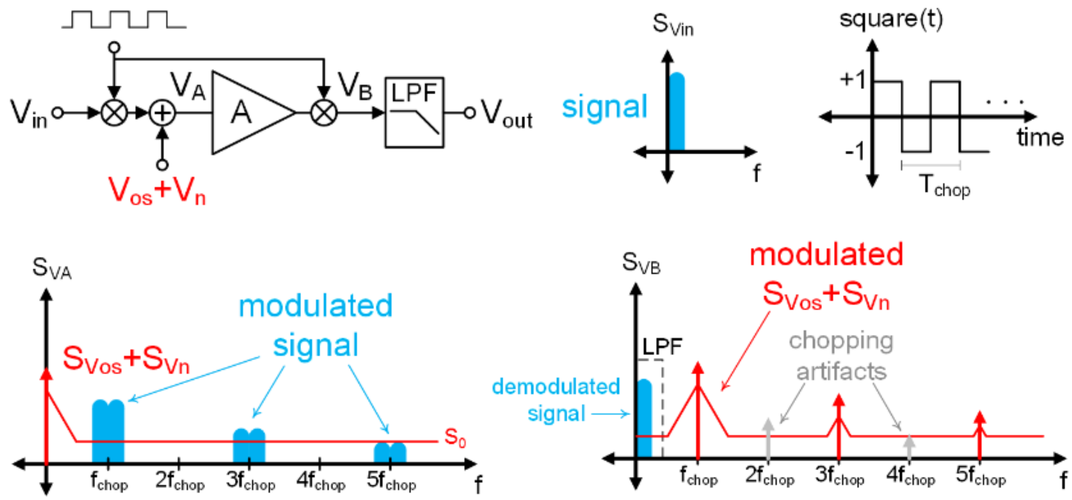


Key issue for sensors: 1/f noise

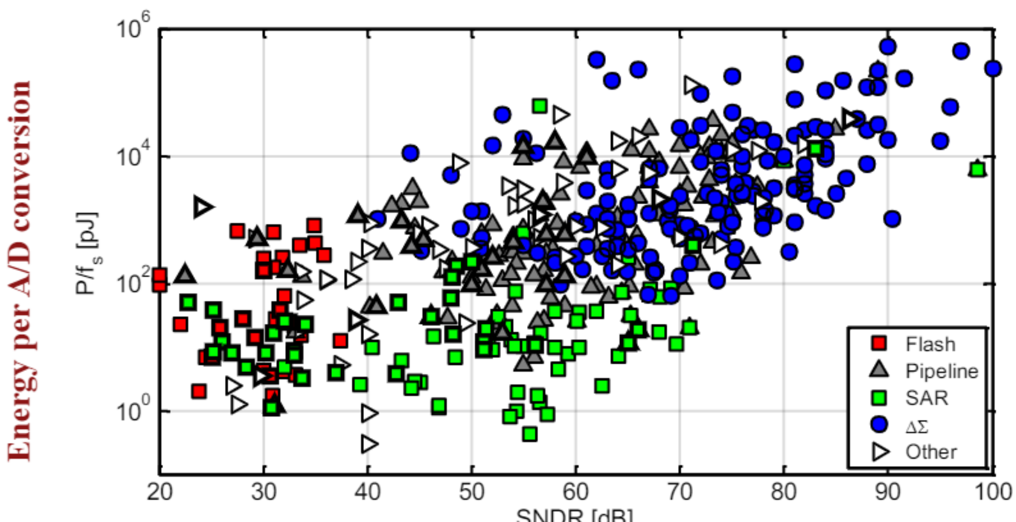


- Analog MOS circuits come with a variety of low frequency artifacts that can easily "swamp out" DC or low-frequency signals of interest

Chopper amplification

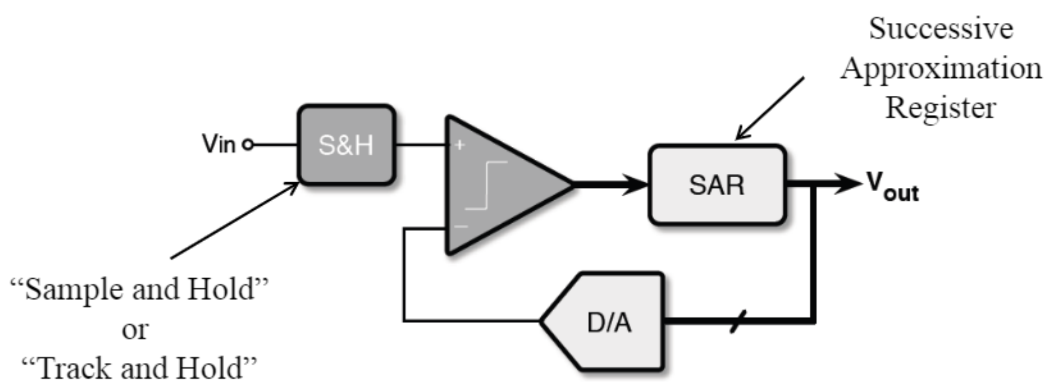


Analog-to-digital conversion



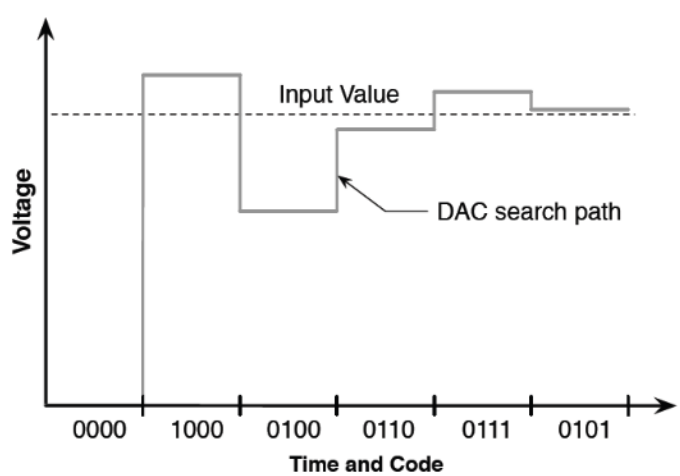
B. Murmann, "ADC Performance Survey 1997-2016," [Online]. Available: http://web.stanford.edu/~murmann/adcsurvey.html.

The most efficient architecture for IoT applications: SAR ADC



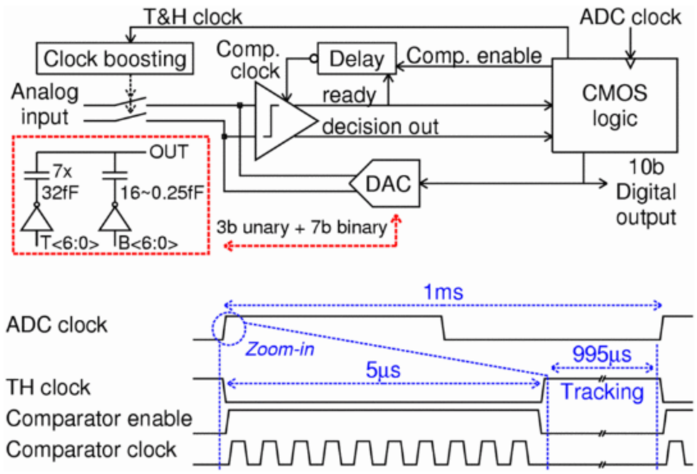
Maloberti, Understanding Microelectronics: A Top-Down Approach, Wiley, 2011

The most efficient architecture for IoT applications: SAR ADC



Maloberti, Understanding Microelectronics: A Top-Down Approach, Wiley, 2011

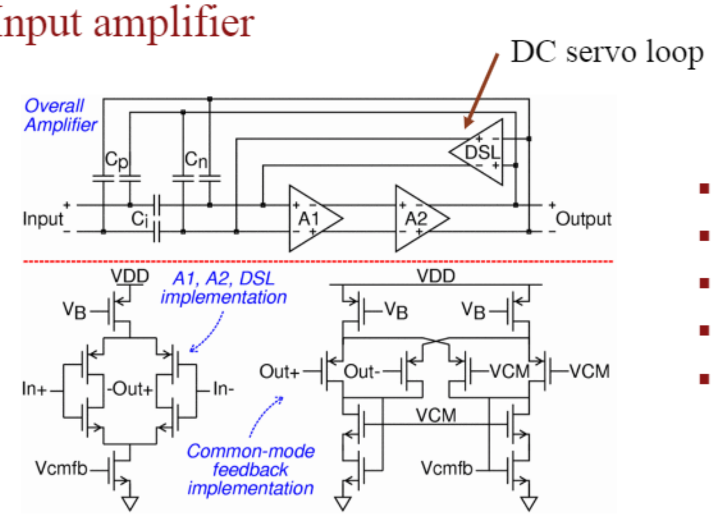
Implementation example



- Consumes only 1 nW at 1 kS/s!
  - Often negligible compared to RF and digital energy
- Conversion occurs within 5 µs of 1 ms period
  - Converter is off/idle most of the time

P. Harpe, et al. "A 3nW signal-acquisition IC integrating an amplifier with 2.1 NEF and a 1.5fJ/conv-step ADC," 2015 IEEE International Solid-State Circuits Conference, Feb. 2015.

Input amplifier



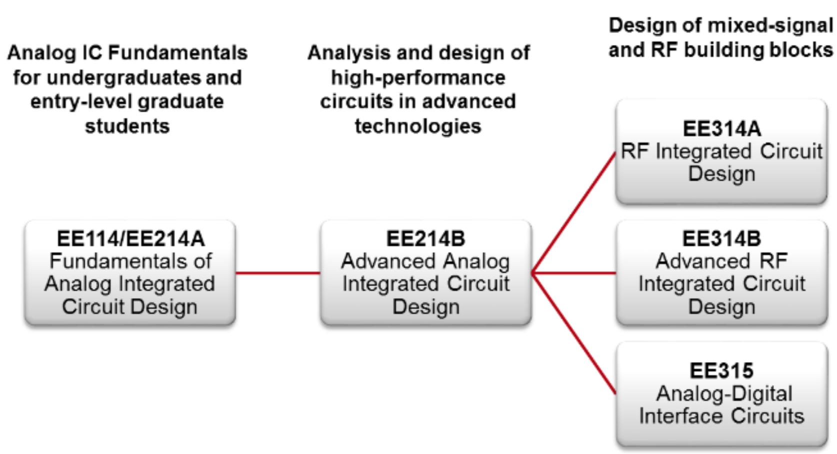
- 32 dB gain
- AC coupled input
- 370 Hz bandwidth
- Input noise = 26 μV_{rms}
- Total current = 1 nA!

P. Harpe, et al. "A 3nW signal-acquisition IC integrating an amplifier with 2.1 NEF and a 1.5fJ/conv-step ADC," 2015 IEEE International Solid-State Circuits Conference, Feb. 2015.

Summary

- The IoT has become a significant driver for the semiconductor industry
- The biggest circuit challenges are related to energy supply
  - Batteries won't improve significantly
  - We may not want to deal with batteries in billions of sensors!
- Take-homes on functional blocks
  - Energy management requires handcrafting; must often fuse multiple sources
  - RF transmission is most power hungry; must be heavily duty cycled
  - Large transmit energy warrants efforts toward increased local computing
  - Local compute strongly benefits from optimal supply & V_T control
  - Analog interfaces are non-trivial (1/f noise, etc.), but not a bottleneck
- Future applications will decide in which direction we will take the next wave of circuit & system design innovations

Related Stanford coursework on analog circuits



The Internet of Things

Embedded Systems

Philip Levis
Associate Professor
Computer Science and Electrical Engineering
Stanford University

Embedded Systems

The "Things" in "Internet of Things"

Computing systems designed for a particular application

- smart lightbulb
- medical implant
- aircraft engine controller/sensor
- personal fitness tracker

Cost and energy are paramount and greatly define design

Hardware and software often selected together, driven by application requirements, and software is often highly customized

This module

You'll learn what embedded systems are and how they differ from traditional computing systems such as servers, phones, and laptops

You'll learn the network and software architectures they typically use and why

You'll learn about the technology drivers behind the recent growth of the IoT

You'll learn about energy budgets and how to design an embedded system

You'll learn about common software techniques and tradeoffs

The Internet of *Things*

What is an embedded system?

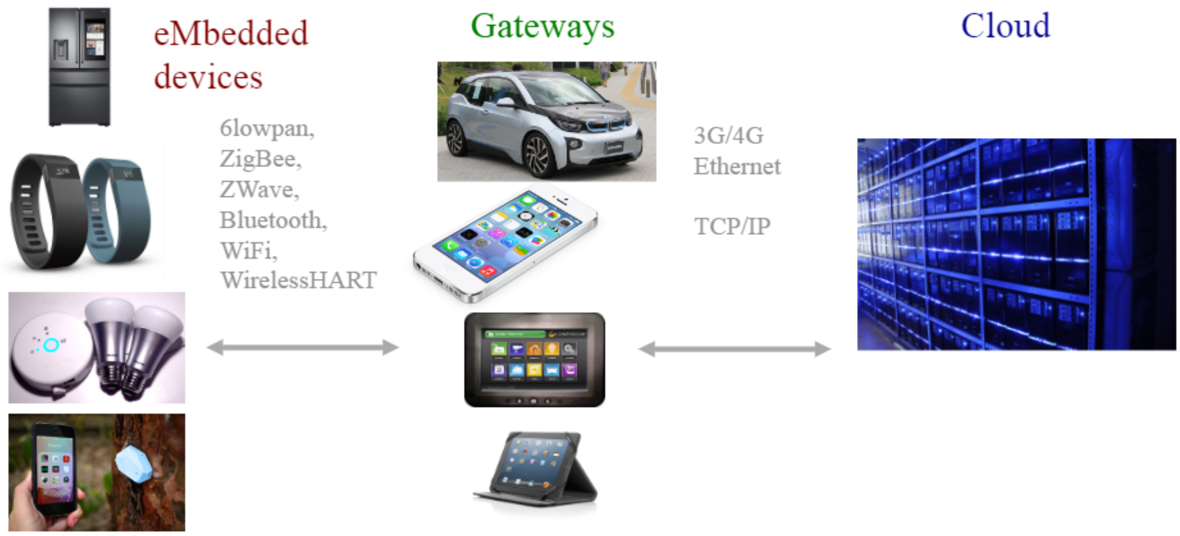
> "An embedded system is a computerized system that is purpose-built for its application."
>
> Elicia White
> Making Embedded Systems, O'Reilly

This has deep implications for how they are designed and why.
Typically, want to minimize cost/maximize lifetime for a given expected workload.
Optimized, custom software that uses as few resources as possible.

Internet(s) of Things

Industrial Automation	Home Area Networks	Personal Area Networks	Networked Devices
Thousands/person	Hundreds/person	Tens/person	Tens/person
Controlled environment	Uncontrolled environment	Personal environment	Uncontrolled environment
High reliability	Unlicensed spectrum	Unlicensed spectrum	Unlicensed spectrum
Control networks	Convenience	Instrumentation	Convenience
Industrial requirements	Consumer requirements	Fashion vs. function	Powered
WirelessHART, 802.15.4	ZigBee, ZWave, Thread	Bluetooth/BLE	WiFi/802.11
6tsch, RPL	6lowpan, Thread	3G/LTE	TCP/IP
IEEE/IIC/IETF	IETF/ZigBee/Thread/Private	3GPP/IEEE	IEEE/IETF

IoT: MGC Architecture



IoT Systems and Applications

Wide range of applications and application requirements

- Marketing uses the same term (IoT), but very different

Commonly a three-tier architecture: embedded, gateway, and cloud

Each tier has different computing systems

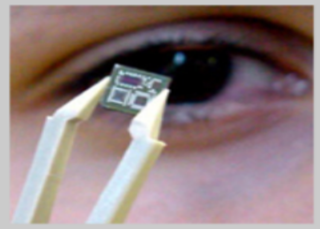
- Embedded: microcontrollers, custom software, energy
- Gateway: often ARM, Android, Linux, iOS
- Cloud: x86_64, virtualized servers (Linux, etc.)

Application is spread across tiers

- Software at each tier receives, processes, and displays data

The EmNets Vision

- "Information technology (IT) is on the verge of another revolution… The use of EmNets [embedded networks] throughout society could well dwarf previous milestones." [1]
- "The motes [EmNet nodes] preview a future pervaded by networks of wireless battery-powered sensors that monitor our environment, our machines, and even *us.*" [2]



[1] National Research Council. *Embedded, Everywhere,* 2001.
[2] MIT Technology Review. *10 Technologies That Will Change the World,* 2003.

15.iii.2005 Stanford Interview Talk 2

Two Game-Changers

ARM Cortex M series

- First released 2004
- Ultra-low power 32-bit processor
- 8-96kB of RAM, 64-512kB code flash
- Sleep currents recently dropped <1µA

Bluetooth Low Energy

- First released in 2006
- Send a 30 byte packet once per second, last for a year on a coin cell battery
- Support was weak until Apple incorporated into iBeacon, now all major smartphones include it

Cortex
Intelligent Processors by ARM

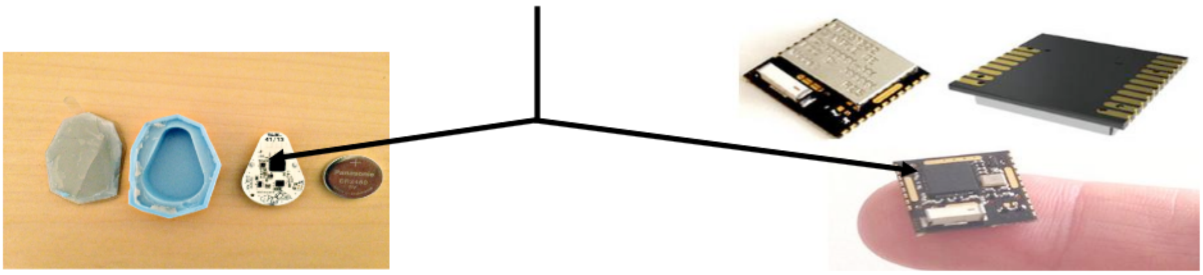
iBeacon

Example Part: nRF51822

Cortex M0+ with integrated 2.4GHz transceiver

- Supports Bluetooth Low Energy
- Two models: 32kB/256kB or 16kB/128kB

DigiKey cost for 25,000: $1.99



6lowpan

We can now inexpensively build powerful networked embedded devices

Early networking: proprietary, vertical silos

- ZigBee defines link up to application layer
- Zwave, etc.
- Connect to Internet through application gateway (lock-in)

6lowpan: IETF standard for IPv6 over low-power link layers

- Just defines IP packet formats/compression (plus UDP)
- Routing/MAC layer independent
- Allows interoperability and software flexiblity
- RFC4944 (basic protocol) and RFC6282 (additional header compression)

Why Today?

Wireless embedded sensors have been in the market for > 15 years

- Built around 8-bit or 16-bit microcontrollers, proprietary protocols

Two major technological shifts

- 32-bit CortexM microcontrollers became efficient enough to use
- Smartphone support for Bluetooth Low Energy allows mobile devices to interact with ubiquitous ones

Specification of 6lowpan opened low-power wireless devices ("things") up to the Internet, hence the "Internet of Things"

Two Game-Changers

ARM Cortex M series

- First released 2004
- Ultra-low power 32-bit processor
- 8-96kB of RAM, 64-512kB code flash
- Sleep currents recently dropped <1µA

Bluetooth Low Energy

- First released in 2006
- Send a 30 byte packet once per second, last for a year on a coin cell battery
- Support was weak until Apple incorporated into iBeacon, now all major smartphones include it

Two Game-Changers

ARM Cortex M series

- First released 2004
- Ultra-low power 32-bit processor
- 8-96kB of RAM, 64-512kB code flash
- Sleep currents recently dropped <1µA

Bluetooth Low Energy

- First released in 2006
- Send a 30 byte packet once per second, last for a year on a coin cell battery
- Support was weak until Apple incorporated into iBeacon, now all major smartphones include it

CortexM microcontrollers

32-bit microcontrollers, use Thumb-1 and Thumb-2 instruction sets

Many types: M0, M0+, M1, M3, M4, M7, M23, M33

- Optional floating point support: M4F single precision, M7F double precision

Three major subfamilies/architectures

- ARMv6-M: Cortex-M0, Cortex-M0+, Cortex-M1
- ARMv7-M: Cortex-M3
- ARMv7E-M: Cortex-M4, Cortex-M7
- Upwards compatible

Wikipedia has excellent details on all of the options/features

- https://en.wikipedia.org/wiki/ARM_Cortex-M

CortexM

CortexM defines an instruction set and core processor functionality

Several pieces of optional functionality

- Memory protection unit: used to protect firmware from applications
- SysTick: standard timer
- Bit banding: spread a word across 32 words, so bits can be individually read/written: useful for GPIO, other single-bit writes

On-chip features/components modeled as memory-mapped *peripherals*

- Each peripheral has a block of addresses for its registers

One huge benefit: compiler/tool support

CortexM has excellent open-source compiler support

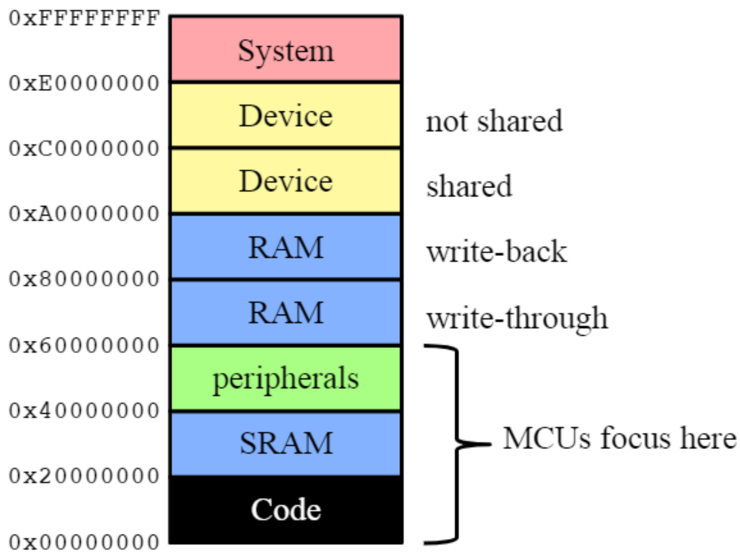
- GCC
- llvm
- Leverages importance of general ARM architecture

msp430-gcc recently taken in-house my TI (mspgcc)

avr-gcc is incorporated into Atmel studio

One issue: integrated development environments (AVR studio, code composer studio) tend to minimize activation energy but eventually become restrictive

CortexM memory map (ARMv6-M)



CortexM

On-chip features/components modeled as memory-mapped *peripherals*

- Each peripheral has a block of addresses for its registers

Every vendor has completely different peripheral interfaces, different chips by the same vendor can differ significant too

- CortexM makes it possible to use the same compiler, boot code, interrupt handling code
- Everything else needs to be implemented per-chip
- Switching from one MCU family to another is a lot of engineering effort
- E.g., switching from Atmel SAM4L to NXP K66

Two Game-Changers

ARM Cortex M series

- First released 2004
- Ultra-low power 32-bit processor
- 8-96kB of RAM, 64-512kB code flash
- Sleep currents recently dropped <1μA

Bluetooth Low Energy

- First released in 2006
- Send a 30 byte packet once per second, last for a year on a coin cell battery
- Support was weak until Apple incorporated into iBeacon, now all major smartphones include it

Bluetooth Low Energy

Not backwards compatible with Bluetooth: additional protocol that sits alongside traditional Bluetooth

- Uses similar physical layer (modulation, channel hopping, etc.)
- Allows transcievers to easily support both

Intended for low data-rate, ubiquitous computing applications

- Much lower energy consumption
- Much lower throughput

BLE basics: link layer

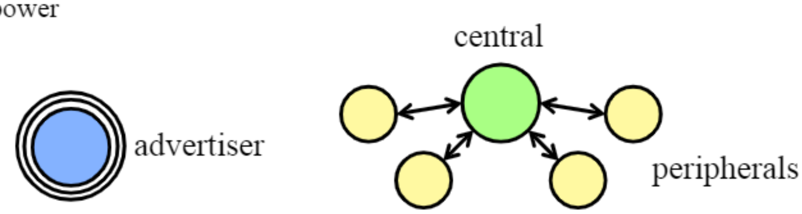
BLE nodes can send advertisements (short broadcast packets) freely

- iBeacon, other beacon technologies use this

Connections use a star topology: a *central* node can have multiple *peripherals*, all connections are between a central and a peripheral

- Centrals cannot connect to other centrals, peripherals cannot connect to other peripheral, peripheral can connect to only one central
- A single device can be both, or alternate: FitBit -> iPhone -> iPad

Central node sets the communication schedule, issues connection request to peripheral: low power



BLE basics: application layer

Generic ATTributes (GATT) are how BLE devices exchange data

Attributes have a handle, a type, and data

Attributes are *characteristics*, *services*, or *profiles*

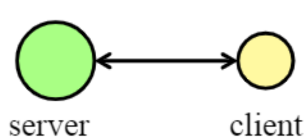
- Form a data hierarchy for querying/discovery

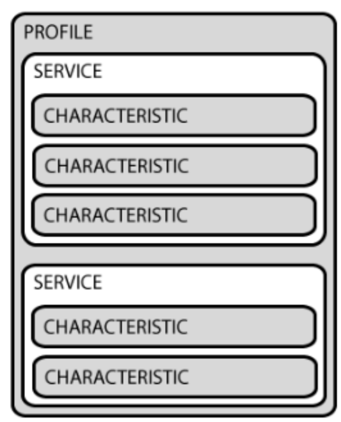
Many standard characteristics, services, profiles

- Many devices just invent their own

A GATT server provides data to a GATT client

- Server/client independent of central/peripheral





Overview

CortexM microcontrollers have many sub-families

- Increasingly powerful instruction sets
- Common memory layout, interrupt handling, etc.
- Each vendor has its own peripherals as memory-mapped I/O
- Leads to excellent compiler support

Bluetooth Low Energy is an ultra-low power link layer

- Advertisements (beacons) for simple data transmission
- Central node controls the schedule for low power operation
- Connections can use GATT, a hierarchical data representation for requesting particular data

Microcontrollers and Systems-on-Chip (SoCs)

Microcontroller executes code, has on-chip peripherals

- More powerful costs more, higher power
- Cortex-M0, CortexM4, Cortex-M4F, …
- On-chip peripherals (SPI, ADC, CAN

System-on-chip (SoC) combines a microcontroller core with more complex subsystems/peripherals

- Radios, Digital signal processors (DSPs)
- Exact line between "microcontroller" and "SoC" is very blurry (don't worry about it)

This module: learn about an example MCU and SoC

Atmel SAM4L series: functionality

Cortex M4 core

128-512kB flash (program) memory, 32-64kB RAM

Operates at up to 48MHz at 1.68 to 3.6V

4 USART (UART or SPI) busesm 4 I2C (two-wire) buses

Hardware AES encryption support

USB hardware support

8-bit or 12-bit ADC (3-15 channels depending on form factor)

15 DMA channels for I/O processing offload

1.5-3µA sleep, 1.5µs wakeup, as low as 90µA/MHz active (4.3mA @ 48MHz)

http://www.atmel.com/products/microcontrollers/ARM/SAM4L.aspx

Nordic nRF51 SoC

Cortex-M0 core

Integrated Bluetooth Low Energy transciever

128-256kB flash (program) memory, 16-32kB RAM

Operates at 16MHz at 1.68 to 3.6V

1 UART, 1 SPI, one I2C bus

Hardware AES encryption support (limited)

8-bit, 9-bit, 10-bit ADC

2.6µAsleep, 4.2µs wakeup, 2.4-4.1mA active (RAM vs. flash)

16mA TX (@ +4dBm), 13.4mA RX (but radio is usually off)

Takeaways

Huge variations in what different MCUs/SoCs can do

Orders of magnitude differences in active vs. sleep current

RF is much more expensive than compute

Two basic cost considerations dominate

Energy

- Embedded systems have an expected workload – want to minimize energy required to handle the workload (longer battery life, more efficient)
- More powerful, highly featured MCUs draw more power: can be more *power efficient*, but consume more energy for a given workload

Money

- Many embedded systems markets have tight margins (cars, appliances, etc.)
- More powerful, highly featured MCUs cost more

Both costs push designs to pick minimal MCU and optimize code for it

Energy examples

Cortex M4 core

128-512kB flash (program) memory, 32-64kB RAM

Operates at up to 48MHz at 1.68 to 3.6V

4 USART (UART or SPI) busesm 4 I2C (two-wire) buses

Hardware AES encryption support

USB hardware support

8-bit or 12-bit ADC (3-15 channels depending on form factor)

15 DMA channels for I/O processing offload

1.5-3µA sleep, 1.5µs wakeup, as low as 90µA/MHz active (4.3mA @ 48MHz)

http://www.atmel.com/products/microcontrollers/ARM/SAM4L.aspx

Nordic nRF51 SoC

Cortex-M0 core

Integrated Bluetooth Low Energy transciever

128-256kB flash (program) memory, 16-32kB RAM

Operates at 16MHz at 1.68 to 3.6V

1 UART, 1 SPI, one I2C bus

Hardware AES encryption support (limited)

8-bit, 9-bit, 10-bit ADC

2.6µAsleep, 4.2µs wakeup, 2.4-4.1mA active (RAM vs. flash)

16mA TX (@ +4dBm), 13.4mA RX (but radio is usually off)

Cost examples

Part	Family	Flash	RAM	Cost	Notes
ATSAMD20E15A	Cortex-M0+	16kB	2kB	$1.37	
ATSAMD21E16B	Cortex-M0+	64kB	8kB	$1.70	LIN, USB
NRF51422	Cortex-M0	256kB	32kB	$2.44	BLE
ATSAMD20J18A	Cortex-M0+	256kB	32kB	$2.69	20 12-bit ADC
ATSAM4S2BA	Cortex-M4	128kB	64kB	$2.80	SSC, USB
ATSAMD21G18A	Cortex-M0+	256kB	32kB	$3.15	LIN, USB
ATSAM4E8EA	Cortex-M4	512kB	128kB	$7.62	CAN, Ethernet, USB, IrDA

Data from DigiKey, July 2017, for ~1,000 units ; at scale costs are much lower

Takeaways

If energy is important, sleep as much as possible and avoid using radio

- Orders of magnitude differences in active vs. sleep current
- RF is much more expensive than compute

Choose the smallest MCU you need for the application

- Huge variations in what different MCUs/SoCs can do
- Cost increases with features

Energy is the limiting resource

Many embedded systems are battery-powered

- Consuming less energy means longer lifetime or fewer recharges

Calculating an energy budget allows you to reason about design tradeoffs

$$E = P_s \cdot t_s + P_a \cdot t_a$$

$$E = \text{sleep energy} + \text{active energy}$$

E: energy
P: power
t: time
s: sleep
a: active

Example energy budget

nRF51422 that wakes up at 1Hz and transmits a single BLE advertisement

$$E = P_s \cdot t_s \qquad + \qquad P_a \cdot t_a$$

E = sleep power * sleep time + active power * active time
E = 4.8µA * sleep time + 14.6mA * active time
E = 4.8µA * 999.7ms + 14.6mA * 0.3ms
E = 4.8uAs + 4.38µAs

E: energy
P: power
t: time
s: sleep
a: active

CR2032 is 225mAh – in theory can last 3 years (in practice much less)

Sleep current assumes all RAM retained; active current assumes 0dB transmit plus CPU at 16MHz
Active time is 37 bytes @ 1Mbps = 300µS

Alarm bells should be going off

Energy is power * time, not current * time

Often use current when calculating microcontroller power budgets

- Batteries do not provide constant voltage
- Don't use voltage regulators because of significant (energy) costs
- Take advantage of MCU tolerances (e.g. 1.8-3.6V)

Complication

We assumed that system instantaneously transitioned from sleep to wake

- MCU takes time to wake up
- Transceiver takes time to power up

Transition times – high power but no work – can be significant

- Wake up less often, for longer, to amortize over wake period

$$E = P_s \cdot t_s + P_a \cdot t_a + P_T \cdot t_T$$

$$E = \text{sleep energy + active energy + transition energy}$$

Example

It takes the nRF51422 140µS to transition to TX state, draws 7mA

$$E = P_s \cdot t_s \quad + \quad P_a \cdot t_a \; + \; P_T \cdot t_T$$

E: energy
P: power
t: time
s: sleep
a: active
T: transition

E = 4.8µA * 999.6ms + 14.6mA * 0.3ms + 7.0mA * 0.14ms
E = 4.8uAs + 4.38µAs + 1µAs

Not considering transition times is off by 10%

Takeaways

Energy consumed is the sum of active and sleep energy
For ultra-low power applications, sleep and active energy can be nearly equal
Transition costs can also be significant
Based on energy budget, pick battery with desired capacity

Minimizing Energy Consumption

To minimize sleep energy, put microcontroller into lowest possible state

- Microcontrollers have different many low-power states and power-saving features
- Complex implications to software

To minimize active energy, minimize time peripherals and MCU are active

- Perform operations in parallel to minimize active time
- Cluster/batch operations to minimize transition times
- Minimize clock rate

Sleep states: SAM4L

Four basic running modes

- Run – MCU executes instructions, everything can be active
- Sleep – no instructions, any clocks/peripherals can be active
- Wait – no instructions, only 32kHz clock active for peripherals
- Retention – no instructions, only 32kHz clock, no peripherals

Mode	**Current**	**Wakeup latency**
Run (@48MHz)	14.5mA	-
Sleep	50µA	0.25µs
Wait	6µA	1.5µs
Retention	3µA	1.5µs

Sleep state: SAM4L clocks

SAM4L has many different clocks: 32khZ, 1, 4, 8, 12, 80MHz
Higher speeds draw more power: use slowest clock possible (dividers)
But, if a fast enough clock is already on, can subdivide for slower rates

E.g., if need 1MHz and If RCFAST is on, use a clock divider of 12

Clock	Speed	Current
OSC32K	32kHz	350nA
RCSYS	116kHz	2µA
RC1M	1MHz	35µA
RCFAST	12MHz	180µA

Parallelism

Parallelism allows fixed overheads to amortize over multiple operations

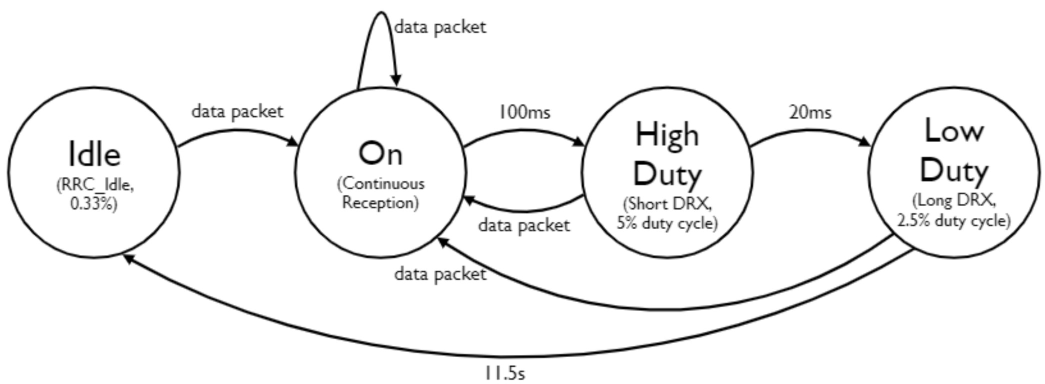
```
loop {
    sample_sensor();
    radio_on();
    send_value();
    sleep();
}
```

High-cost clocks are on longer,
in SLEEP (50µA) state longer.

```
loop {
    parallel {
      sample_sensor();
      radio_on();
    }
    send_value();
    sleep();
}
```

When could this be more efficient?

Batching: LTE as an example



LTE Batching

Using LTE, compare

- Sending one small packet every 30s
- Sending two small packets every 60s

One packet every 30s

- (0.1s * 100%) + (0.02 * 5%) + (11.5 * 2.5%) + ((30-11.62) * 0.33%)
- 0.1 + 0.001 + .2875 + .06 = .3576s/30s
- = 1.2%

Two packets every 60s

- (0.101s * 100%) + (0.02s * 5%) + (11.5s * 2.5%) + ((60-11.62)s * 0.33%)
- 0.101 + 0.001 + .2875 + .160 = .4577s/60s
- = 0.76%

Minimize clock rate

Most peripherals draw current based on clock rate

- UART: 8.5µA/MHz
- SPI: 1.9µA/MHz

Lowering the clock rate reduces power drawn; if a fast enough clock is already on, can subdivide for slower rates

E.g., if need 1MHz and If RCFAST is on, use a clock divider of 12

Clock	Speed	Current
O0SC32K	32kHz	350nA
RCSYS	116kHz	2µA
RC1M	1MHz	35µA
RCFAST	12MHz	180µA

Minimizing Energy Consumption

To minimize sleep energy, put microcontroller into lowest possible state

- Microcontrollers have different many low-power states and power-saving features
- Complex implications to software

To minimize active energy, minimize time peripherals and MCU are active

- Perform operations in parallel to minimize active time
- Cluster/batch operations to minimize transition times
- Minimize clock rate

CortexM

CortexM defines an instruction set and core processor functionality

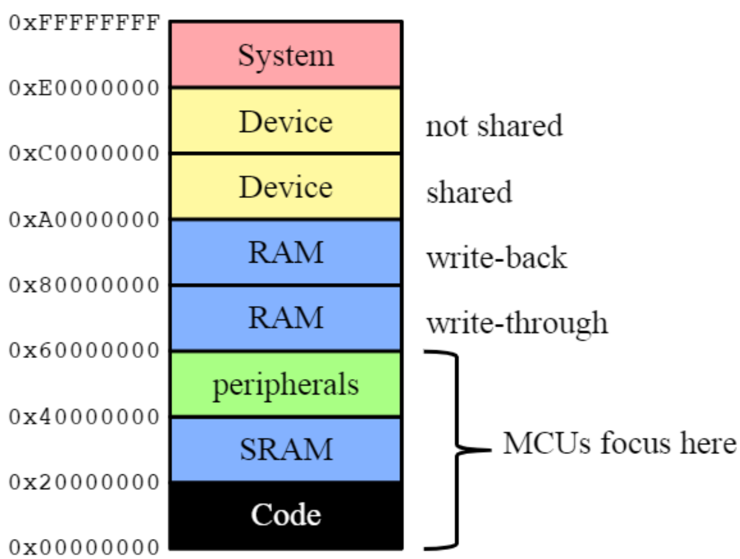
Several pieces of optional functionality

- Memory protection unit: used to protect firmware from applications
- SysTick: standard timer
- Bit banding: spread a word across 32 words, so bits can be individually read/written: useful for GPIO, other single-bit writes

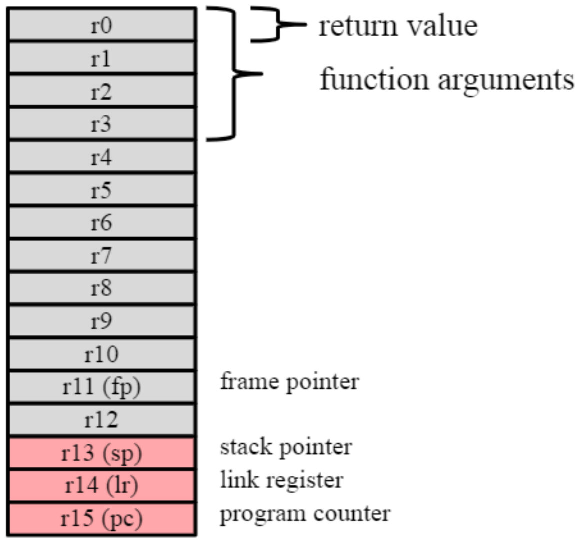
On-chip features/components modeled as memory-mapped *peripherals*

- Each peripheral has a block of addresses for its registers

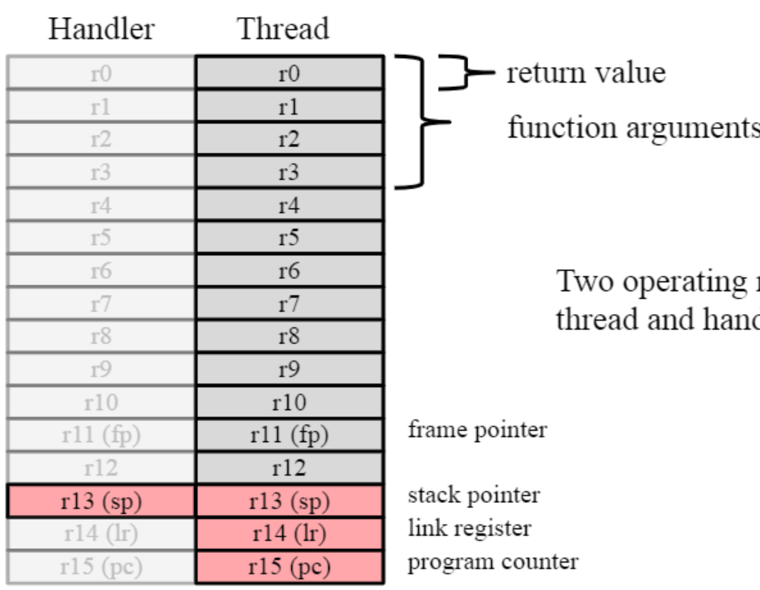
CortexM memory map (ARMv6-M, ARMv7-M)



Core registers

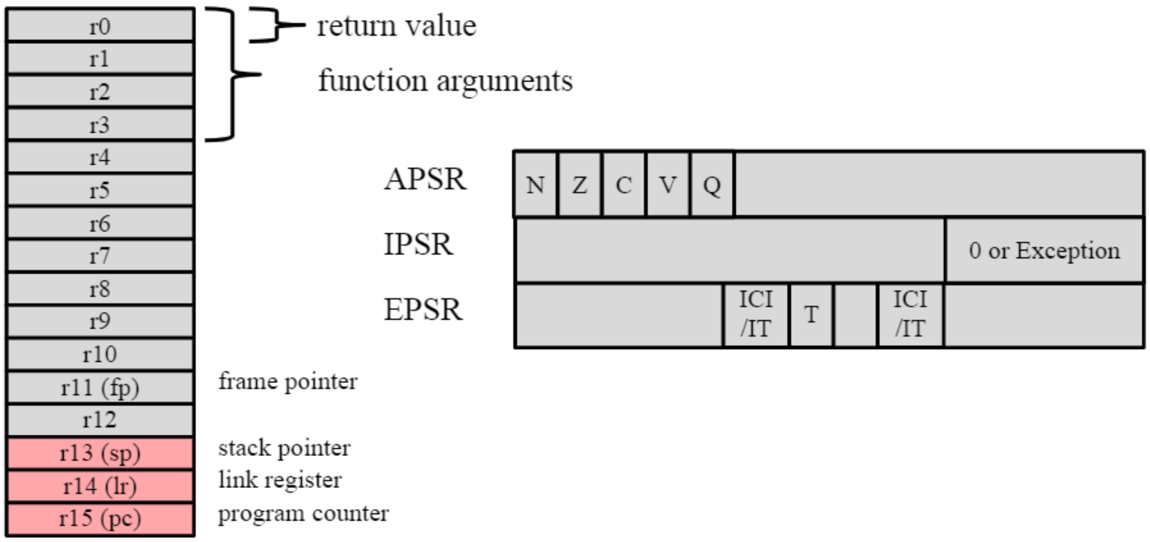


Core registers



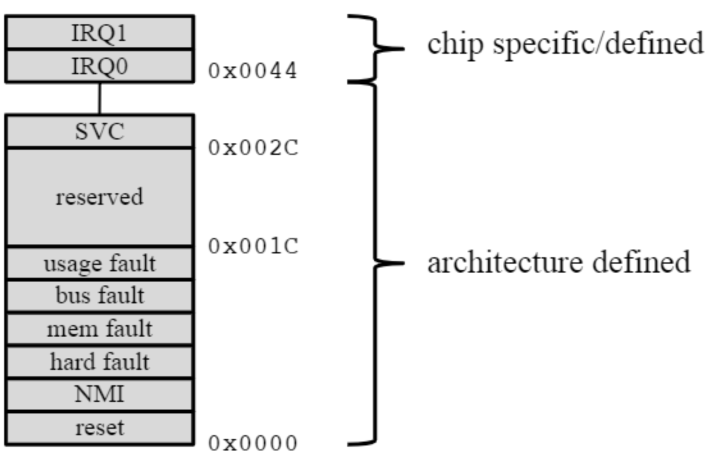
Two operating modes:
thread and handler

Control Registers (M4)



Interrupt Vector table (M4)

At address 0x0 (can be relocated with VTOR register)



CortexM

CortexM defines an instruction set and core processor functionality

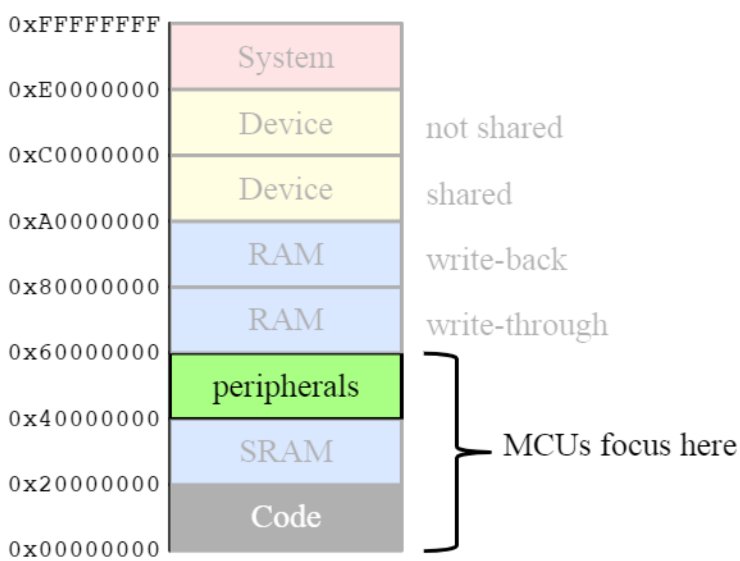
Several pieces of optional functionality

- Memory protection unit: used to protect firmware from applications
- SysTick: standard timer
- Bit banding: spread a word across 32 words, so bits can be individually read/written: useful for GPIO, other single-bit writes

On-chip features/components modeled as memory-mapped *peripherals*

- Each peripheral has a block of addresses for its registers

CortexM memory map (ARMv6-M, ARMv7-M)



Peripherals

Each peripheral is mapped to a region (0x1000 in size) of memory

Registers laid out in that memory (completely up to chip)

Most associated with an interrupt (NVIC index/IRQ#)

SAM4L AESA

AES accelerator

- AES-128
- ECB
- CBC
- CF
- OF
- CTR

Offset	Register	Name	Access	Reset
0x00	Control	CTRL	RW	0x00000000
0x04	Mode	MODE	RW	0x000F0000
0x08	Data buffer pointer	DATABUFPTR	RW	0x00000000
0x0C	Status	SR	R	0x00010000
0x10	Interrupt enable	IER	W	0x00000000
0x14	Interrupt disable	IDR	W	0x00000000
0x18	Interrupt mask	IMR	R	0x00000000
0x20	Key 0	KEY0	W	0x00000000
0x3C	Key 7	KEY7	W	0x00000000
0x40	Initialization vector 0	INITVECT0	W	0x00000000
0x4C	Initialization vector 3	INITVECT3	W	0x00000000
0x50	Input data	IDATA	W	-
0X60	Output data	ODATA	R	-
0x70	DRNG register	DRNGSEED	W	0x00000000

General operation

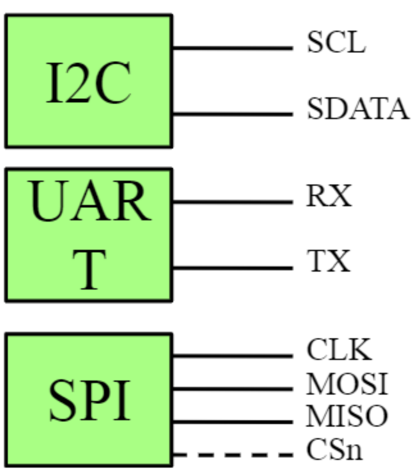
Configure peripheral for operation (e.g., AES input data, key)

Set bit in control register to start operation

Return data on completion

- Spin loop on interrupt mask for very fast operations
- Handle interrupt for longer operations
- Read data out of data register (if input)

Buses



Larger-scale I/O

Some operations transfer a small amount of data

- Single byte UART write
- Single ADC sample

Others transfer more data, sometimes very fast

- Transfer USB data at 8MHz
- High frequency ADC sampling (100 of kHz)

For larger transfers, per-byte interrupts can be expensive

- Wakeup, active power cost energy
- Software may not handle interrupts fast enough

DMA: Direct Memory Access

Set up a series of operations as a direct memory access operation

- Provide a buffer to read into and/or write from
- Tell system how many operations
- Start operation

Receive DMA interrupt when whole series of operations complete

Amortize interrupt cost across many operations

- Consumes less energy (fewer CPU cycles)
- Can run faster (not limited by interrupt rate)

Peripherals

On-chip accelerators for common operations

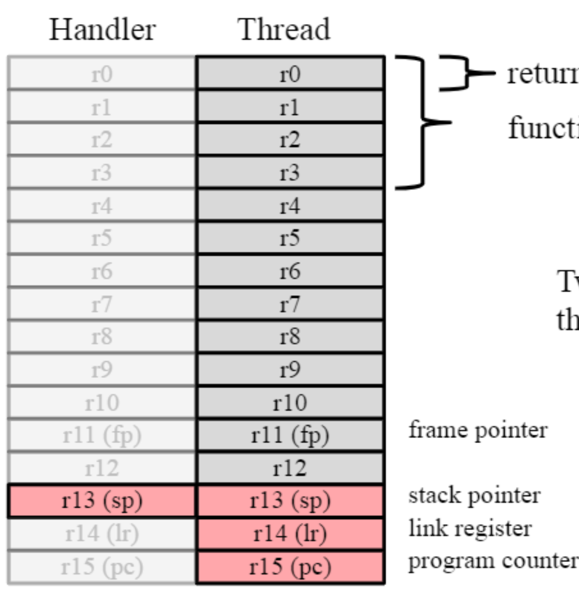
- Huge variety, depending on application focus on MCU

Interact/use with memory-mapped registers

Larger operations can use DMA to save energy

Fast operations can use DMA to run faster

Handling Interrupts



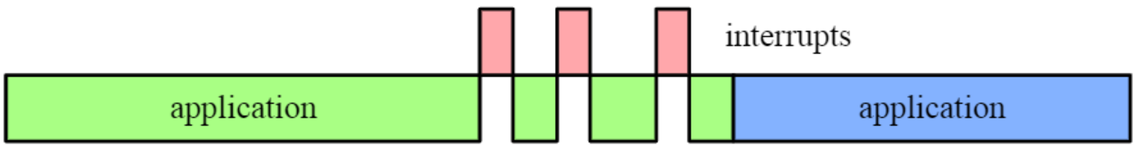
Two operating modes:
thread and handler

Execution model

Thread mode executes main loop (application code, etc.)

Handler mode executes interrupts

- Interrupts can happen at any time
- Can be disabled, but doing this for a long time is dangerous



The danger of interrupts

Thread mode

```
extern int a;
void inc() {
  a = a + 1;
}

00008000 <inc>:
    8000:    e52db004    push   {fp}
    8004:    e28db000    add    fp, sp, #0
    8008:    e59f3018    ldr    r3, [pc, #24]
    800c:    e5933000    ldr    r3, [r3]
    8010:    e2832001    add    r2, r3, #1
    8014:    e59f300c    ldr    r3, [pc, #12]
    8018:    e5832000    str    r2, [r3]
    801c:    e24bd000    sub    sp, fp, #0
    8020:    e49db004    pop    {fp}
    8024:    e12fff1e    bx     lr
    8028:    00010070    .word  0x00010070
```

Handler mode

```
extern int a;
void dec() {
    a = a - 1;
}

0000802c <dec>:
    802c:    e52db004    push   {fp}
    8030:    e28db000    add    fp, sp, #0
    8034:    e59f3018    ldr    r3, [pc, #24]
    8038:    e5933000    ldr    r3, [r3]
    803c:    e2432001    sub    r2, r3, #1
    8040:    e59f300c    ldr    r3, [pc, #12]
    8044:    e5832000    str    r2, [r3]
    8048:    e24bd000    sub    sp, fp, #0
    804c:    e49db004    pop    {fp}
    8050:    e12fff1e    bx     lr
    8054:    00010070    .word  0x00010070
```

The danger of interrupts

Thread mode

```
extern int a;
void inc() {
  a = a + 1;
}

00008000 <inc>:
    8000:    e52db004    push   {fp}
    8004:    e28db000    add    fp, sp, #0
    8008:    e59f3018    ldr    r3, [pc, #24]
    800c:    e5933000    ldr    r3, [r3]
    8010:    e2832001    add    r2, r3, #1   <- danger
    8014:    e59f300c    ldr    r3, [pc, #12]
    8018:    e5832000    str    r2, [r3]
    801c:    e24bd000    sub    sp, fp, #0
    8020:    e49db004    pop    {fp}
    8024:    e12fff1e    bx     lr
    8028:    00010070    .word  0x00010070
```

Handler mode

```
extern int a;
void dec() {
    a = a - 1;
}

0000802c <dec>:
    802c:    e52db004    push   {fp}
    8030:    e28db000    add    fp, sp, #0
    8034:    e59f3018    ldr    r3, [pc, #24]
    8038:    e5933000    ldr    r3, [r3]
    803c:    e2432001    sub    r2, r3, #1
    8040:    e59f300c    ldr    r3, [pc, #12]
    8044:    e5832000    str    r2, [r3]
    8048:    e24bd000    sub    sp, fp, #0
    804c:    e49db004    pop    {fp}
    8050:    e12fff1e    bx     lr
    8054:    00010070    .word  0x00010070
```

If dec() runs when inc() is at 0x8010, then result of dec() will be lost: it has copied `a` into `r3`.

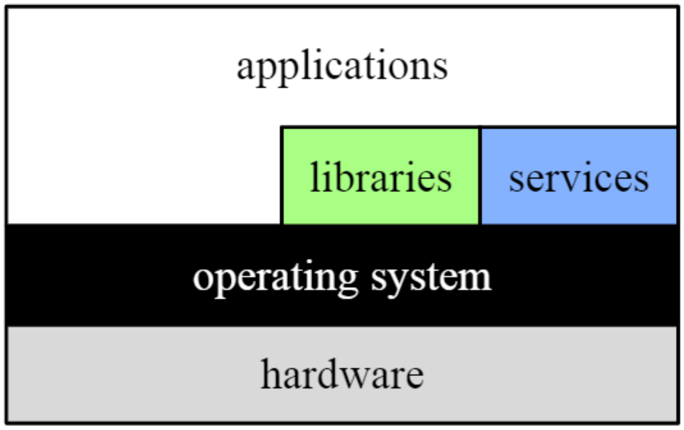
Dealing with interrupts

Option 1: spin loop to wait (check pending flag)
- Pro: no latency, no race conditions
- Con: wasted cycles, no concurrency

Option 2: disable interrupts in critical sections
- Pro: only small latency, full concurrency
- Con: critical sections must be small, easy to make mistakes

Option 3: minimize code in interrupt (delay processing to sequential code)
- Pro: simple code, full concurrency
- Con: interrupts can be delayed significantly

Software Architecture



Operating System

Kernel: lowest level of software

- Directly accesses hardware
- Has complete control of system
- Example: Linux, Windows, OSX

Libraries and services on top of kernel

- Networking, storage, other utilities

General goal: minimize what's in the kernel

- Challenge: crossing between kernel and libraries is expensive
- If abstract requires a lot of crossings, move inside kernel

Embedded Operating Systems

Many MCUs do not distinguish kernel and user code

- No hardware memory protection
- Application code can crash entire system
- Requires diligence and care in software design

Trend is changing, with CortexM memory protection unit (MPU)

- Embedded Oses haven't caught up

Embedded OSes have to walk a fine line

- Provide useful software abstractions for applications
- Allow applications to break through abstractions when necessary

Key OS responsibility: execution model

How do you write code that responds to interrupts?

How do you write code that runs in the background?

When can code preempt other code?

Interrupt code

Code in interrupt handler can block entire system (no interrupts handled)

- Preemptive interrupts are especially challenging (re-use of handler context)
- Goal: minimize code in direct handler

Use deferred procedure call (interrupt bottom half)

- Interrupt handler does core work (e.g., pull data out of register into queue)
- Deferred procedure call executes after interrupt handler returns
- Interrupts can preempt deferred procedure call/bottom half

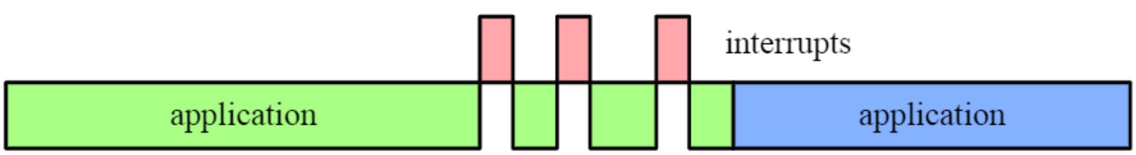
Interrupt code

Code in interrupt handler can block entire system (no interrupts handled)

- Preemptive interrupts are especially challenging (re-use of handler context)
- Goal: minimize code in direct handler

Use deferred procedure call (interrupt bottom half)

- Interrupt handler does core work (e.g., pull data out of register into queue)
- Deferred procedure call executes after interrupt handler returns
- Interrupts can preempt deferred procedure call/bottom half



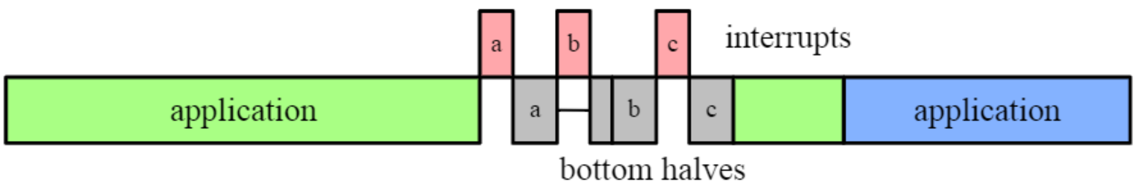
Interrupt code

Code in interrupt handler can block entire system (no interrupts handled)

- Preemptive interrupts are especially challenging (re-use of handler context)
- Goal: minimize code in direct handler

Use deferred procedure call (interrupt bottom half)

- Interrupt handler does core work (e.g., pull data out of register into queue)
- Deferred procedure call executes after interrupt handler returns
- Interrupts can preempt deferred procedure call/bottom half



Background code

Application code: preempted by interrupts and bottom halves

Less time critical (although may still have deadlines)

Option 1: event-driven

- Function that executes in response to an event (timer, packet reception)
- Cross-event state stored in global variables

Option 2: threaded

- Function that never returns: calls functions that block (resume thread on completion)
- Cross-event state stored on stack (higher RAM cost)

Example: event-driven

```
receive_callback(receive_cb);

#define MAX_LEN 128
uint8_t send_buffer[MAX_LEN];

void receive_cb(void* buf, uint8_t len) {
  if (!busy) {
    uint8_t safe_len= (len > MAX_LEN)? MAX_LEN: len);
    memcpy(buf, send_buffer, safe_len);
    send(send_buffer, safe_len);
    busy = true;
  }
}
```

Example: threaded

```
#define MAX_LEN 128

void main() {
  while (1) {
    uint8_t buffer[MAX_LEN];
    len = receive(buffer, MAX_LEN);
    send(buffer, len);
  }
}
```

Note: packets received while sending are dropped

FreeRTOS

Intended for real-time use (has priorities, deadlines, etc.)

Thin loop on top of hardware events

Extremely flexible, supports many chipsets

Allows (expects) you to write your own interrupt handlers

Also supports threads (called "tasks")

- Every task has a priority, higher is more important (idle task is 0)
- Tasks scheduled preemptively
- Interrupts can make tasks runnable

ARM Mbed OS

ARM Mbed OS provides

- Standard APIs for chip peripherals (SPI, I2C, etc.)
- Libraries on top of standard APIs

Supports both event-driven and threaded programming

- Threaded programming has easier power management
- Can write interrupt handlers, etc.

Generally speaking, APIs are richer and cleaner than FreeRTOS

But, actual Mbed implementations are by vendors so be careful

Operating Systems Review

Lowest level of software for applications to build on

Define concurrency model, APIs

Event-driven vs. threaded execution

This module

You've learned what embedded systems are and how they differ from traditional computing systems such as servers, phones, and laptops

You've learned the network and software architectures they typically use and why

You've learned about the technology drivers behind the recent growth of the IoT

You've learned about energy budgets and how to design an embedded system

You've learned about common software techniques and tradeoffs

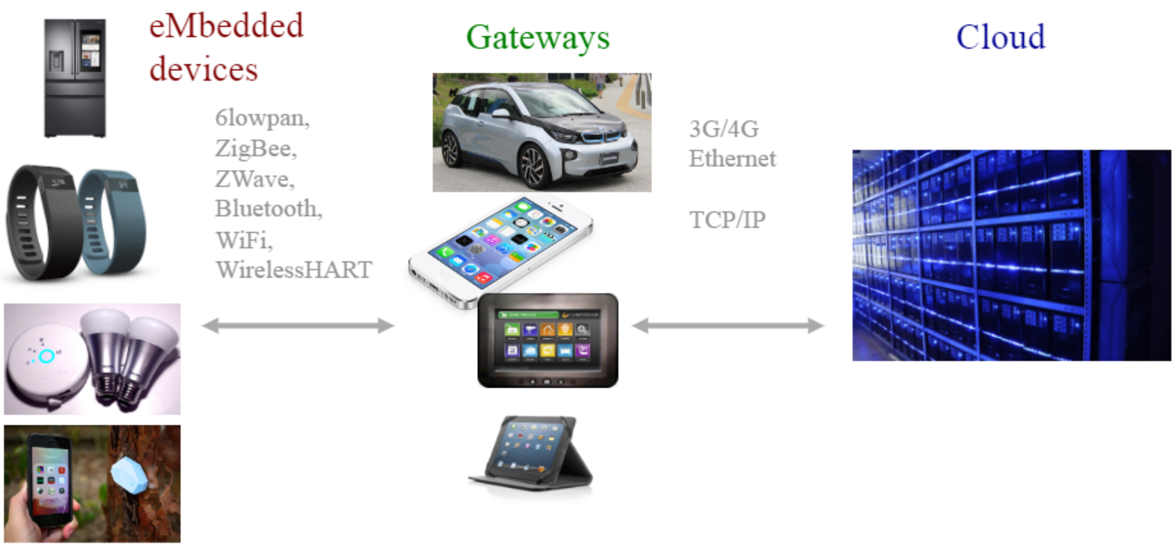
What is an embedded system?

"An embedded system is a computerized system that is purpose-built for its application."

Elicia White
Making Embedded Systems, O'Reilly

This has deep implications for how they are designed and why.
Typically, want to minimize cost/maximize lifetime for a given expected workload.
Optimized, custom software that uses as few resources as possible.

IoT: MGC Architecture



Two Game-Changers

ARM Cortex M series
- First released 2004
- Ultra-low power 32-bit processor
- 8-96kB of RAM, 64-512kB code flash
- Sleep currents recently dropped <1μA

Bluetooth Low Energy
- First released in 2006
- Send a 30 byte packet once per second, last for a year on a coin cell battery
- Support was weak until Apple incorporated into iBeacon, now all major smartphones include it

Cortex
Intelligent Processors by ARM®

iBeacon

Computing an energy budget

Many embedded systems are battery-powered
- Consuming less energy means longer lifetime or fewer recharges

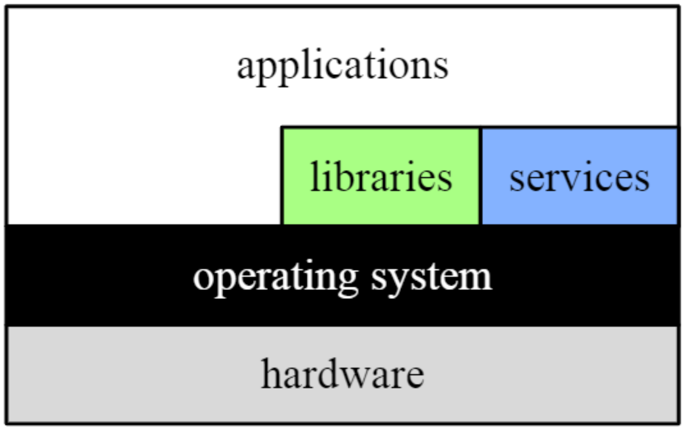
Calculating an energy budget allows you to reason about design tradeoffs

Transition times – high power but no work – can be significant
- Wake up less often, for longer, to amortize over wake period

$$E = P_s \cdot t_s + P_a \cdot t_a + P_T \cdot t_T$$

$$E = \text{sleep energy + active energy + transition energy}$$

Software Architecture



Minimizing Energy Consumption

To minimize sleep energy, put microcontroller into lowest possible state

- Microcontrollers have different many low-power states and power-saving features
- Complex implications to software

To minimize active energy, minimize time peripherals and MCU are active

- Perform operations in parallel to minimize active time
- Cluster/batch operations to minimize transition times
- Minimize clock rate

IoT Connectivity and Networking

Ayfer Özgür
Stanford University

Historical Perspective on Wireless Systems

- First Wireless Age: Station to Station (Kilo Scale)

GBC Marconi- Maritime Communication

Stanford | SystemX Alliance
IoE Focus Area

Historical Perspective on Wireless Systems

- First Wireless Age: Station to Station (Kilo Scale)
- Second Wireless Age: Station to People (Mega Scale)

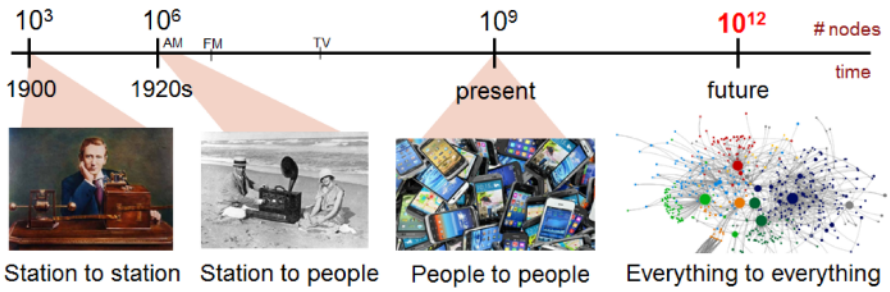
Stanford | SystemX Alliance
IoE Focus Area

Historical Perspective on Wireless Systems

- First Wireless Age: Station to Station (Kilo Scale)
- Second Wireless Age: Station to People (Mega Scale)
- Third Wireless Age: People to People (Giga Scale)

Stanford | SystemX Alliance
IoE Focus Area

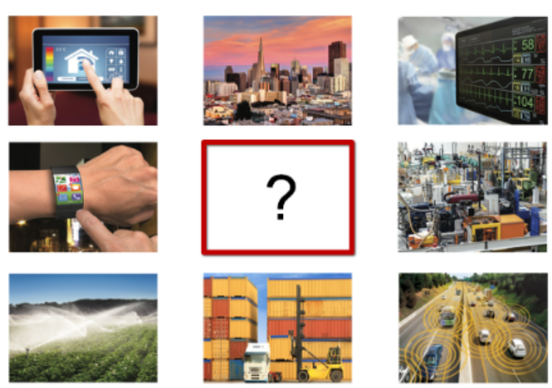
Historical Perspective on Wireless Systems



1000x increase in connectivity in each cycle.

Stanford | SystemX Alliance
IoE Focus Area

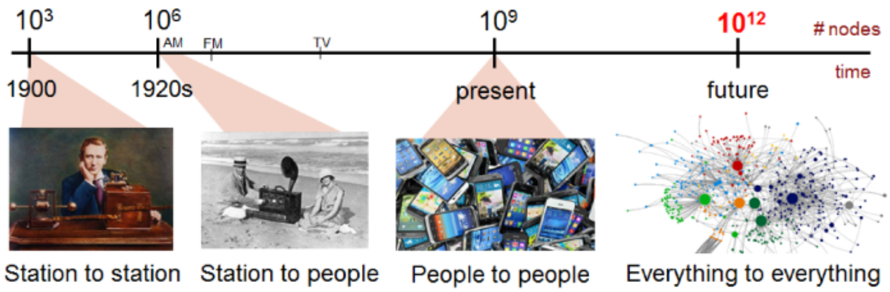
Historical Perspective on Wireless Systems



Connecting various devices and sensors to each other and the Internet.

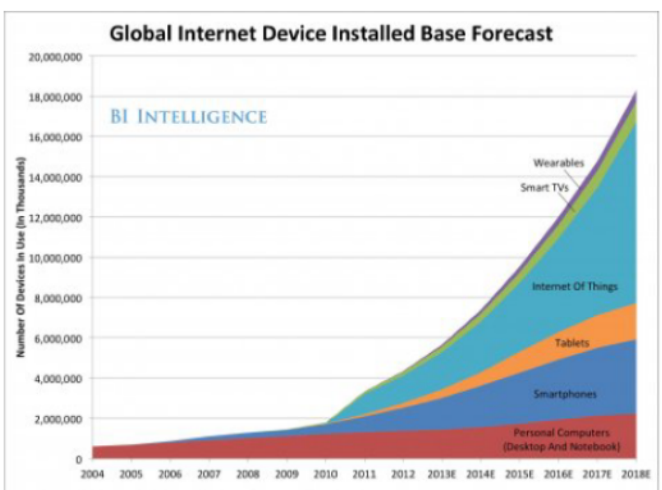
Stanford | SystemX Alliance
IoE Focus Area

Historical Perspective on Wireless Systems



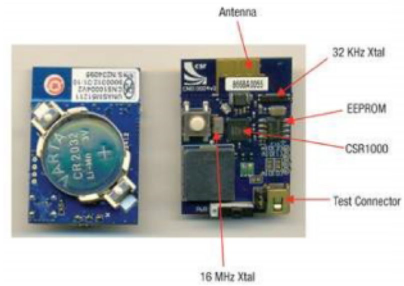
Paradigm shift in principles, technology, systems and applications.

Stanford | SystemX Alliance
IoE Focus Area



Cost, size, maintenance-free operation critical for scalability.

Eliminating the Battery



Batteries are bulky and relatively expensive, require maintenance.

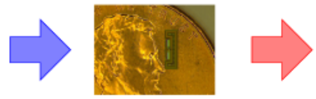
Energy Harvesting Remotely Powered Devices

- New deployment models and applications.
- True mobility.
- Maintenance-free perpetual operation.

Source:wonderfulenginneering.com

Communication Under Random Energy Dynamics

energy in



energy out

- Energy:
  - Static
  - Deterministic

- Energy:
  - Dynamic
  - Random

Communication system design under random energy dynamics.

Massive Multiple Access Networks

Today's Wireless Systems:
- Moderate system size.
- Large payloads.
- Tightly managed and synchronized.
- Node identity and message.

Massive Multiple Access Networks

Tomorrow's IoT Networks:

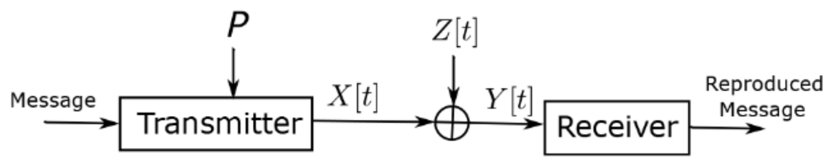
- Massive number of devices, only few active at a time.
- Bursty traffic, few bits to transmit.
- Complexity and energy constrained.
- Node identity may not be relevant.

Source:neoswarm.com

Module Outline

- Energy Harvesting Transmitters for IoT
- Wirelessly Powered Transmitters for IoT
- Massive Multiple Access
- Related Coursework
- Quiz

Basic Model for Communication



$$\frac{1}{n}\sum_{t=1}^{n}|X(t)|^2 \le P$$

Capacity of the AWGN Channel

The most famous formula (Shannon)

$$C = W \log\left(1 + \frac{P}{N_0 W}\right) \quad \text{bits/s}, \qquad \text{SNR} := \frac{P}{N_0 W}$$

Capacity of the AWGN Channel

The most famous formula (Shannon)

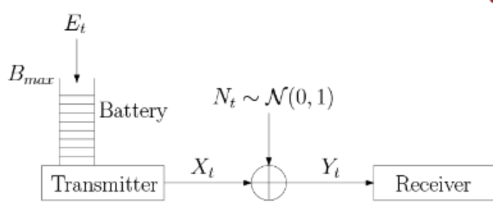
$$C = W \log\left(1 + \frac{P}{N_0 W}\right) \quad \text{bits/s}, \qquad \mathsf{SNR} := \frac{P}{N_0 W}$$

Operating Regimes:

$$C \propto \begin{cases} W & \mathsf{SNR} \ll 0 \text{ dB} \quad \text{Bandwidth-limited} \\ P/N_0 & \mathsf{SNR} \gg 0 \text{ dB} \quad \text{Power-limited} \end{cases}$$

Communication system design is dictated by the operating regime.

Communication Under Random Energy Dynamics



$$|X_t|^2 \le B_t$$

$$B_{t+1} = \min\left(B_t - |X_t|^2 + E_{t+1}, B_{max}\right).$$

E_t: stochastic process known causally at the transmitter and not at the receiver.

$B_{max} = \infty$

Capacity equal to that of a classical AWGN channel with $P = \mathbb{E}[E_t]$:

$$C = W \log\left(1 + \frac{\mathbb{E}[E_t]}{N_0 W}\right)$$

$B_{max} = \infty$

Capacity equal to that of a classical AWGN channel with $P = \mathbb{E}[E_t]$:

$$C = W \log\left(1 + \frac{\mathbb{E}[E_t]}{N_0 W}\right)$$

Insights:

- Use standard communication and coding techniques for the AWGN channel.
- Only relevant property in determining capacity is the average energy harvesting rate.

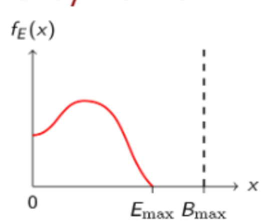
Ref: O. Ozel, S. Ulukus, Achieving AWGN capacity under stochastic energy harvesting, IEEE Transactions on Information theory, 2012.

B_{max} finite

Foremost engineering questions:

- How does the capacity of the energy harvesting AWGN channel depend on system parameters such as B_{max} and E_t?
- What are the properties of E_t most relevant to capacity? What are more favorable and less favorable energy profiles?
- Are there different operating regimes where the dependence to Bmax and E_t is qualitatively different?
- For a given E_t , how can we "optimally" choose B_{max}?
- What is the optimal power control policy?

Capacity for an Energy Harvesting Transmitter

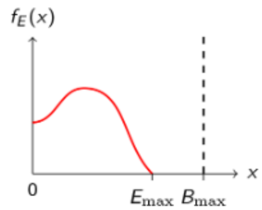


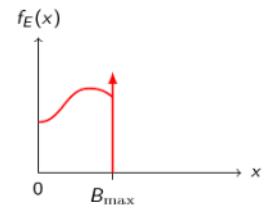
$$B_{\max} > E_{\max}$$

$$C \approx W \log\left(1 + \frac{\mathbb{E}[E_t]}{N_0 W}\right)$$

Ref: D. Shaviv, P.M. Nguyen, A. Ozgur, Capacity of the Energy Harvesting Channel with a Finite Battery, IEEE Transactions on Information theory, 2016.

Capacity for an Energy Harvesting Transmitter





$B_{\max} > E_{\max}$

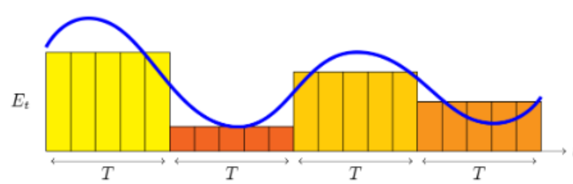
$$C \approx W \log\left(1 + \frac{\mathbb{E}[E_t]}{N_0 W}\right)$$

$B_{\max} < E_{\max}$

$$C \approx W \log\left(1 + \frac{\mathbb{E}[\min\{E_t, B_{\max}\}]}{N_0 W}\right)$$

Ref: D. Shaviv, P.M. Nguyen, A. Ozgur, Capacity of the Energy Harvesting Channel with a Finite Battery, IEEE Transactions on Information theory, 2016.

Correlations in Energy Arrivals



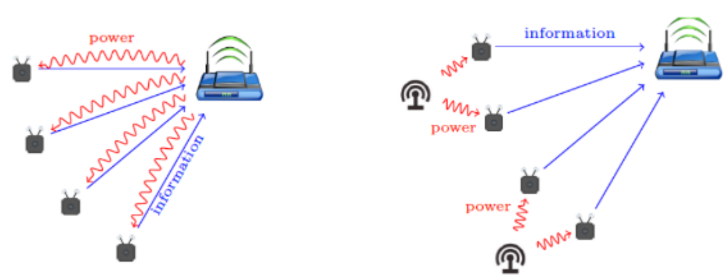
If $B_{max} > \mathbb{E}[E_t] + (E_{max} - \mathbb{E}[E_t])T$,

$$C \approx W \log\left(1 + \frac{\mathbb{E}[E_t]}{N_0 W}\right).$$

Capacity decreases with coherence time T in general.

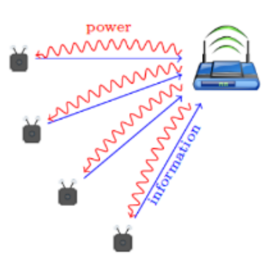
Remotely Powered Communication

Two Topologies for Home IoT



Current practice:
- Transfer energy at a constant rate.
- Periodically charge transmitter’s battery.

Exploit Side Information

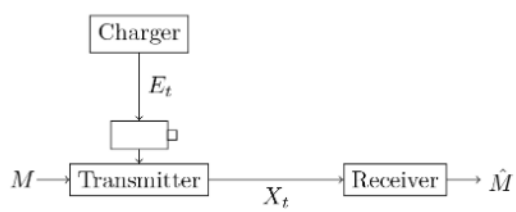


Charger observes the output of the channel.

Charger observes the input to the channel.

Binary Example

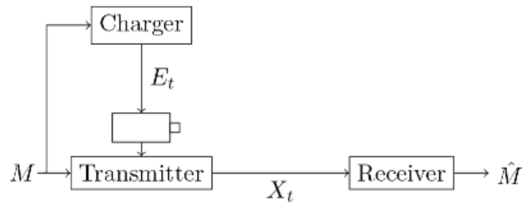
- Charger has no side information:
  - $E_t = 1$, $\forall t$: $C_{\emptyset} = 1$ bits/channel use, $\Gamma = 1$ unit/channel use.



$Y_t = X_t \qquad X_t \in \{0, 1\} \qquad E_t \in \{0, 1\}$

Binary Example

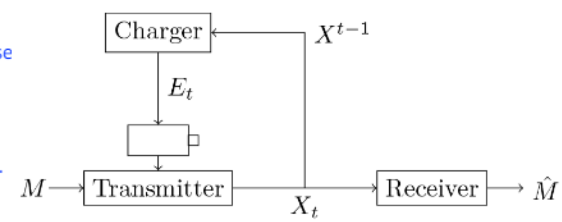
- Charger has no side information:
  - $E_t = 1, \forall t$: $C_{\emptyset} = 1$ bits/channel use, $\Gamma = 1$ unit/channel use.
- Charger knows the message:
  - Charge when the transmitter intends to send a 1:
    - $C_M = 1$ bits/channel use, $\Gamma = 1/2$ units/channel use.



$$Y_t = X_t \qquad X_t \in \{0,1\} \qquad E_t \in \{0,1\}$$

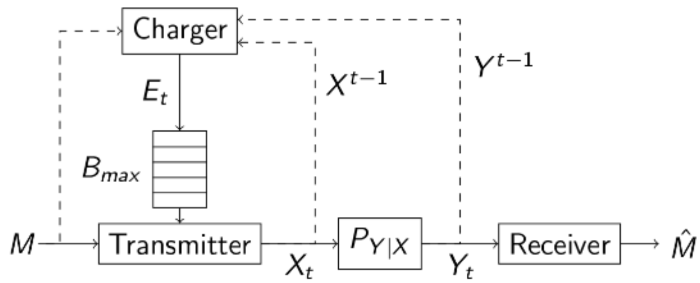
Binary Example

- Charger has no side information:
  - $E_t = 1, \forall t$: $C_{\emptyset} = 1$ bits/channel use, $\Gamma = 1$ unit/channel use
- Charger knows the message:
  - Charge when the transmitter intends to send a 1:
    - $C_M = 1$ bits/channel use, $\Gamma = 1/2$ units/channel use.
- Charger can observe the transmitted signal X_{t-1}:
  - Charge when battery is empty:
    - $C_X = 1$ bits/channel use, $\Gamma = 1/2$ units/channel use.



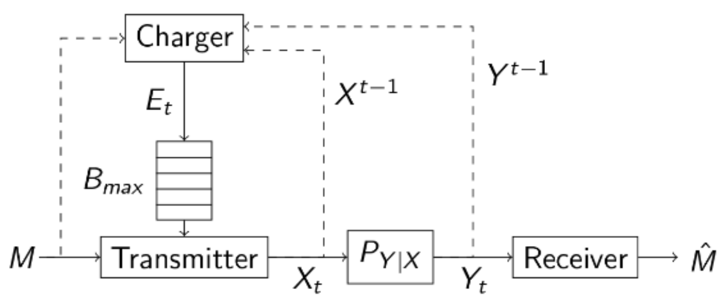
$$Y_t = X_t \qquad X_t \in \{0,1\} \qquad E_t \in \{0,1\}$$

Remotely Powered Communications



- Charger: Dynamically decide how much energy to transfer to the receiver based on its side information regarding the transmission (subject to an average power constraint Γ).
- Transmitter: Dynamically adapt its transmission scheme based on its instantaneous battery level.

Remotely Powered Communications

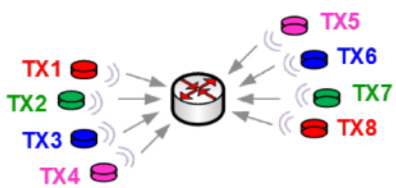


Exploiting side information at the charger can enable performance close to the centralized case.

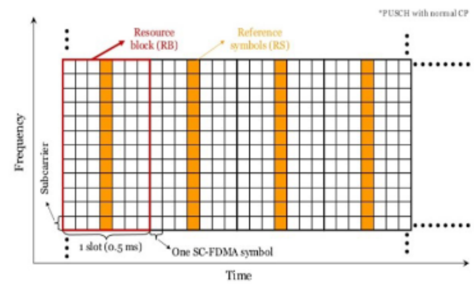
Ref: D. Shaviv, A. Ozgur, H. Permuter, Capacity of Remotely Powered Communication, IEEE Transactions on Information theory, 2016.

Reservation Systems

- LTE, GSM etc.
- Narrowband IoT



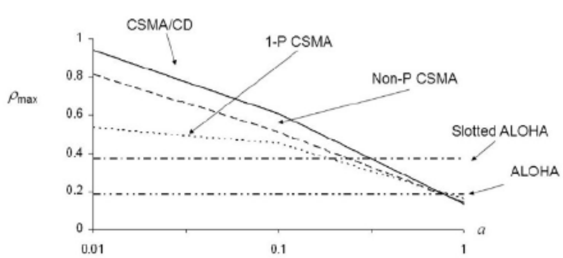
frequency and time resources are tightly allocated across active devices



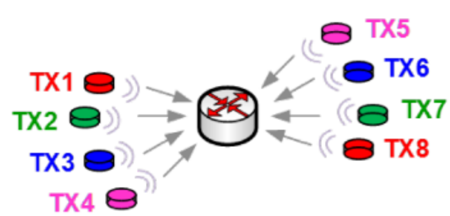
Overhead for coordination becomes the bottleneck when the number of devices is large and payload is small.

Random Access Systems

CSMA, CDMA, Aloha

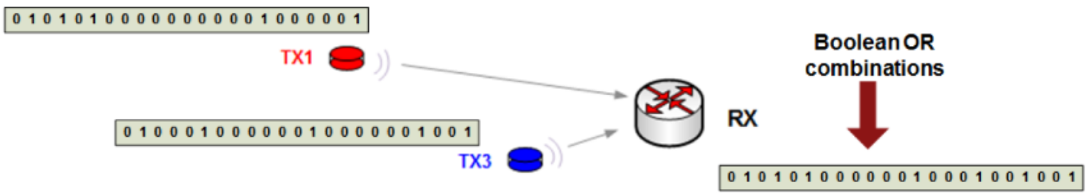


Wifi, Lora, ZigBee, BLE, etc.



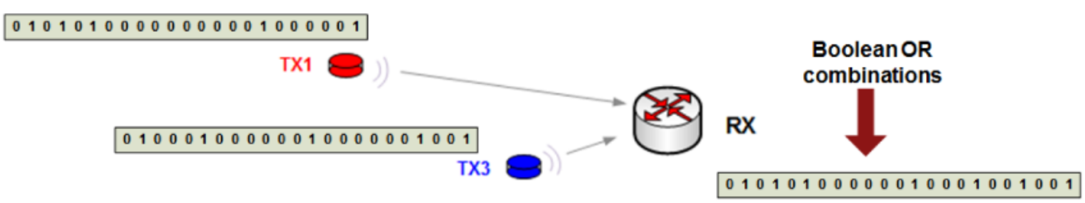
- Carrier-sensing is energy-consuming.
- Aloha leads to high collision rates.

Embracing Collisions



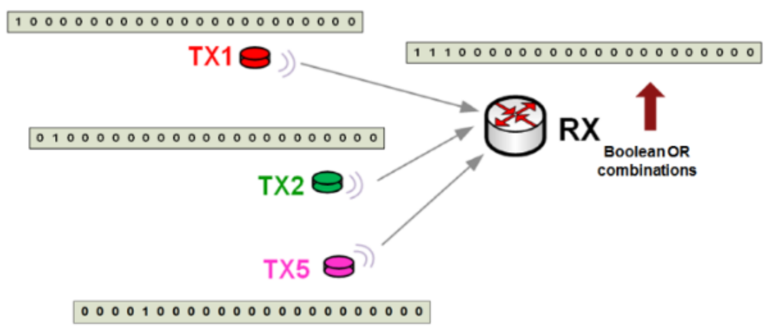
Design codes that are resilient up to a given number of collisions.

Node Identification



- On-off keying used for transmission.
- Energy detection at the receiver.
- At most d out of N users active at a time where $d \ll N$.

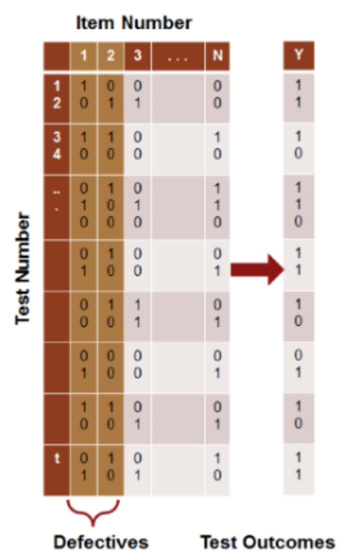
Node Identification



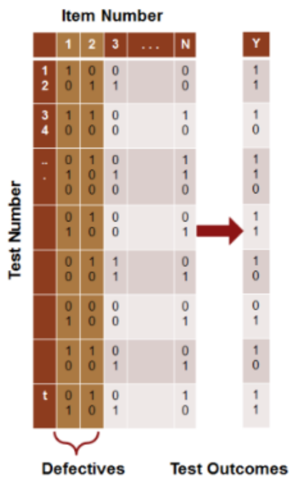
Trivial Solution: $t = N$.

Group Testing

- N items (soldiers).
- d defective (infected items).
- Figure out the defective group.
- Group items together and apply t tests.



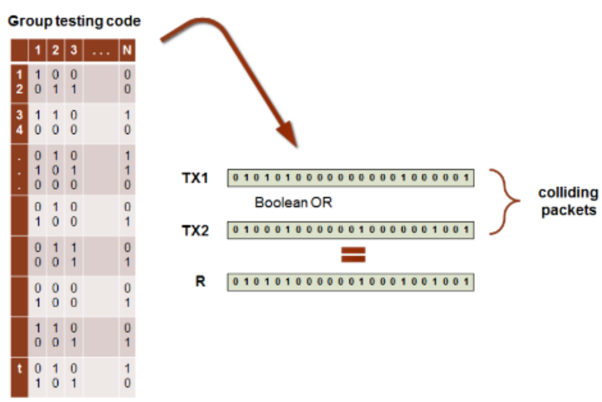
Group Testing



Group testing literature:

$$t = \Theta(d \log N)$$

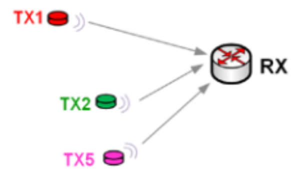
Collision Resolving Codes via Group Testing



Node identification: assign a single distinct column to each users.

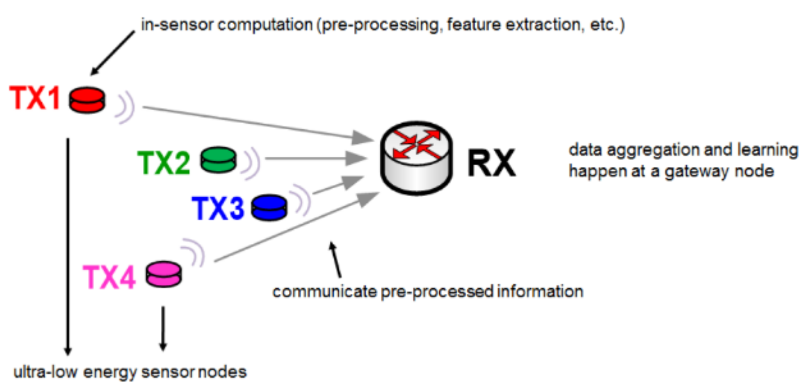
Collision Resolving Codes vis Group Testing

	1	2	3	4	5	6	7	8	. . .	N
1	1	0	0	1	1	0	1	0		0
2	0	1	1	0	1	0	1	0		0
3	1	1	0	1	0	1	0	1		1
4	0	0	0	1	0	0	0	1		0
⋮	0	1	0	0	1	0	0	1		1
	1	0	1	0	0	0	0	1		1
	0	0	0	0	0	1	0	1		0
	0	1	0	1	0	1	1	0		0
	1	0	0	0	0	1	0	0		1
	0	1	1	0	0	1	0	1		0
	0	0	1	0	0	0	0	0		1
	0	0	0	1	1	0	1	0		0
	1	0	0	0	1	1	1	0		1
	1	1	0	1	0	1	0	1		0
	0	0	1	0	0	1	0	1		1
t	0	1	0	1	1	0	1	0		1
	1	0	1	1	0	0	1	1		0



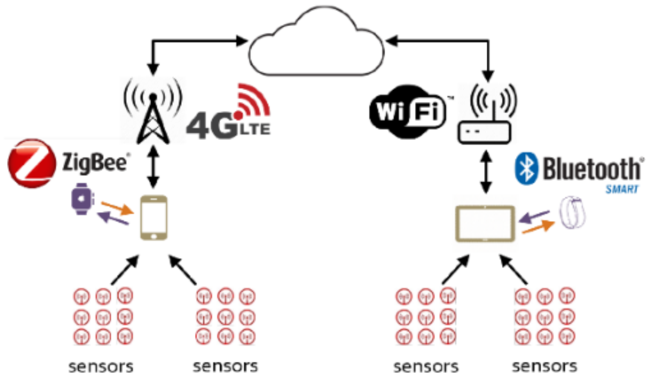
Joint Identification and Information Transmission: assign multiple distinct columns to each user.

Communication vs Computation



How to trade-off communication vs computation.

Communication vs Computation



Compute and communicate at multiple stages.

Related Coursework on Communication