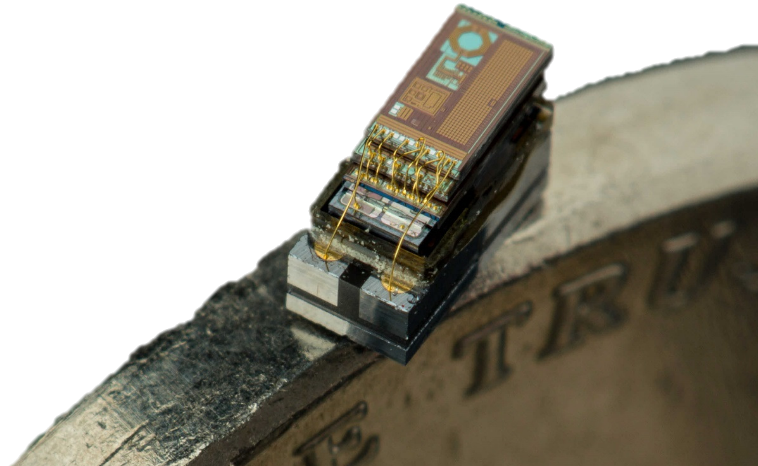
The Recent Past and Distant Future of [Micro-Scale] Embedded Systems

Pat Pannuto, University of Michigan
ppannuto@umich.edu, patpannuto.com

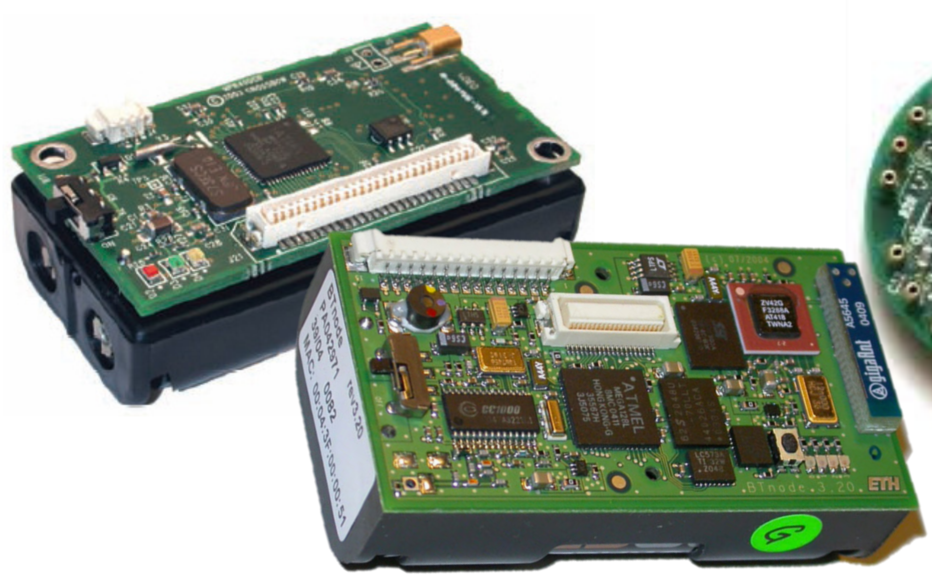
NextMote Workshop @ EWSN'16, Graz, Austria
February 15, 2016

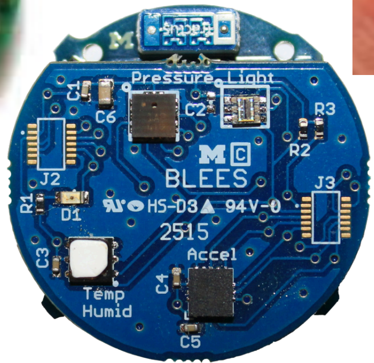


“Micro-Scale” are systems whose principle node dimension is measured on the order of millimeters

- Today’s motes are decimeter-scale…

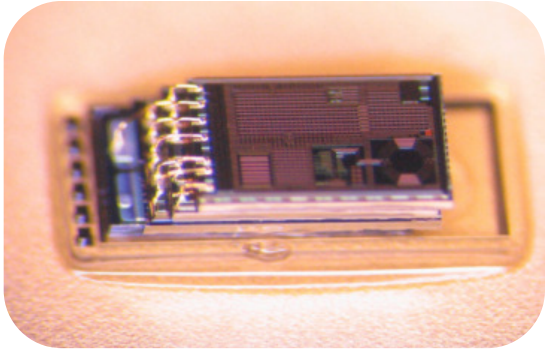
…and pushing towards centimeter-scale



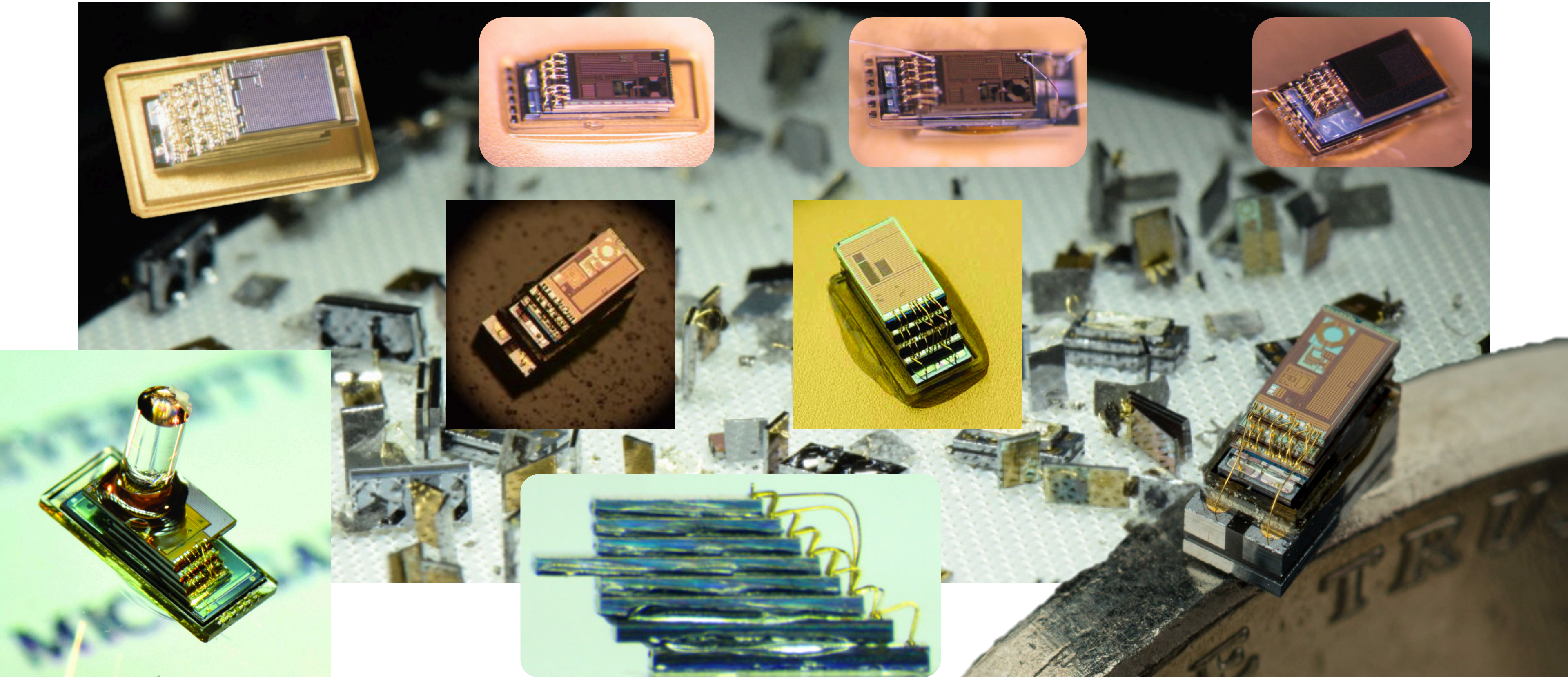


A millimeter-sized system is a 1000x reduction in volume

- In practice, several nodes fit on a period.



We have developed dozens of millimeter-scale motes as part of the Michigan Micro Mote (M3) project



This talk: A tour of micro-scale mote challenges

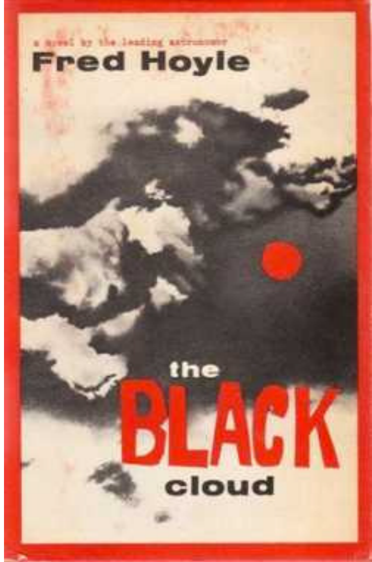
- A Brief History of “Smart Dust”
- Micro-Scale Composition Architecture
- Energy
- Communication
- Debugging

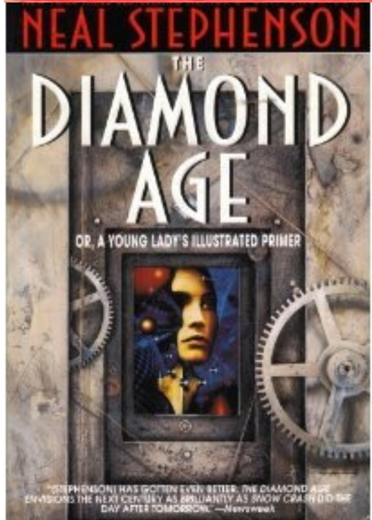
This talk: A tour of micro-scale mote challenges

- A Brief History of "Smart Dust"
- Micro-Scale Composition Architecture
- Energy
- Communication
- Debugging

“Smart Dust” is not new…

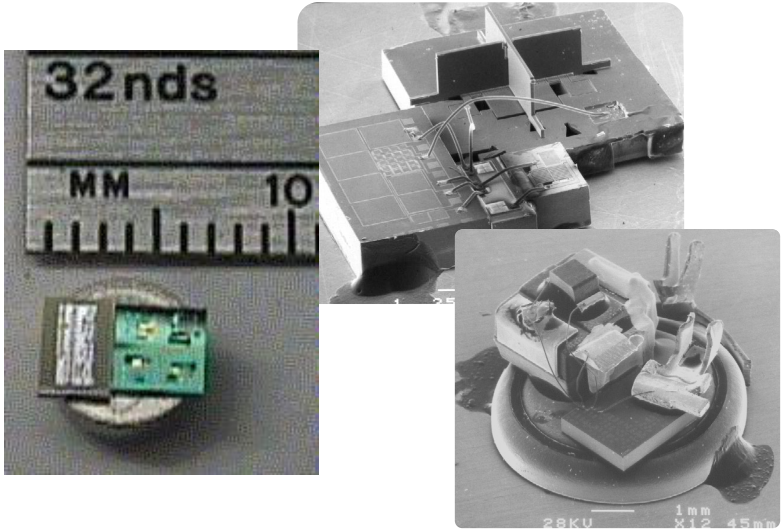
- 1957: Fred Hoyle’s *Black Cloud*
  - Sentient space-faring cloud organism
- Early-mid 90’s: Viable?
  - First serious discussions at RAND and ISAT workshops
- 1995: Neal Stephenson’s *Diamond Age*
  - Popularized a modern vision of smart dust
- 1998: Kris Pister coins the term
  - And starts making smart dust a reality!





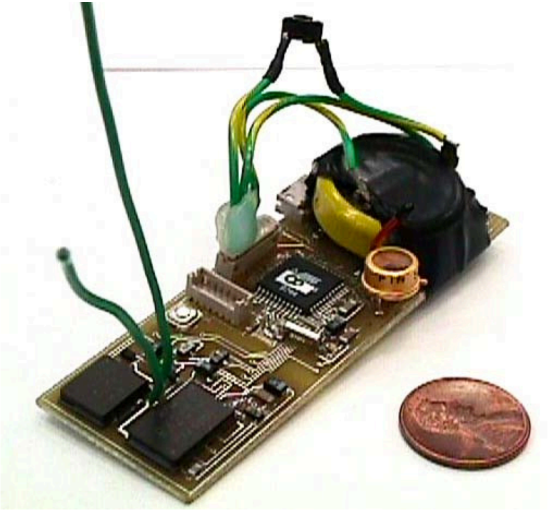
First-generation smart dust missed its 1mm^3 goal, but their so-called “macro motes” were perhaps the most important

- Ad-hoc composition lessened efficiency of volumetric space



http://robotics.eecs.berkeley.edu/~pister/SmartDust/

- Early “macro motes” dovetailed and helped drive early “mote” development



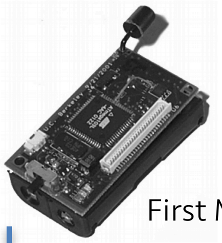
1999

- 900 MHz radio
- Magnetometer
- Accelerometer
- Light
- Temperature
- Pressure

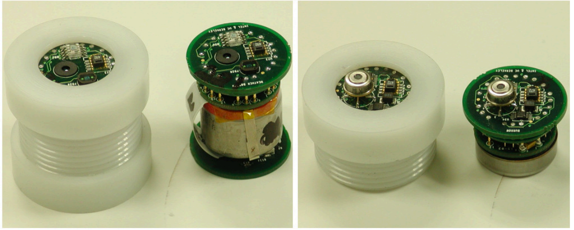
One week lifetime in continuous operation, 2 years w/ 1% duty cycle

The history of modern mote research picks up around this time

CSP Workshop @ Xerox PARC [soon to be IPSN]



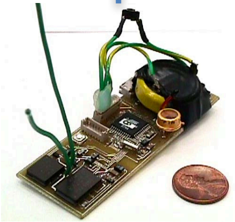
First MICA



Great Duck Island

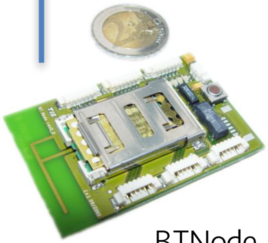
TelosB (Tmote Sky)

Epic Core (modular platforms)

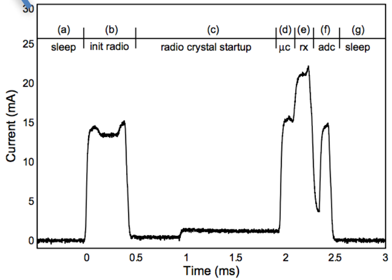


Pister & Hollar's "Macro Motes"

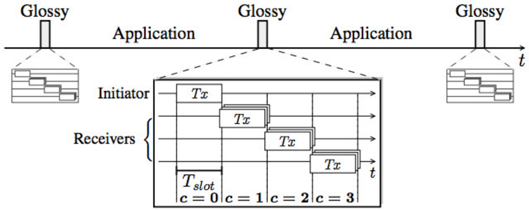
Workshop on Wireless Sensor Networks and Applications [soon to be SenSys]



BTNode



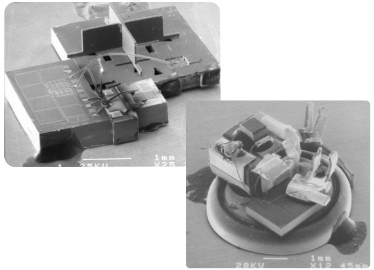
B-MAC



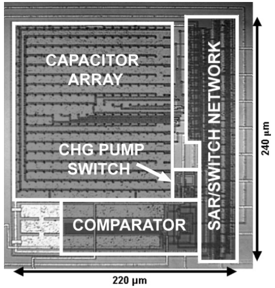
Glossy

But what happened to research pushing towards a 1mm^3 system?

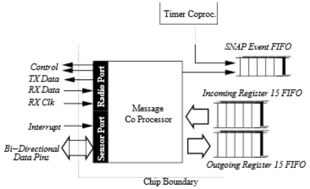
?



We stopped building mm-scale *systems*, and focused on mm-scale *circuits* – at least for a little while

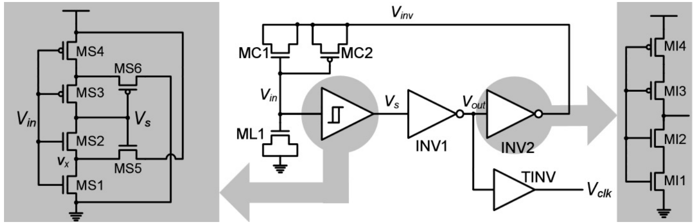


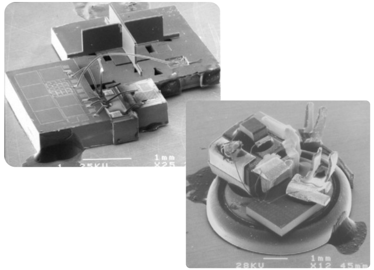
M. Scott '03

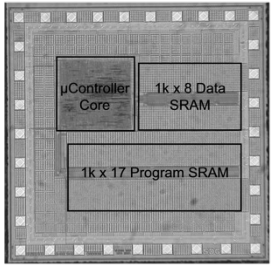


V. Ekanayake '04

Y.-S. Lin '07

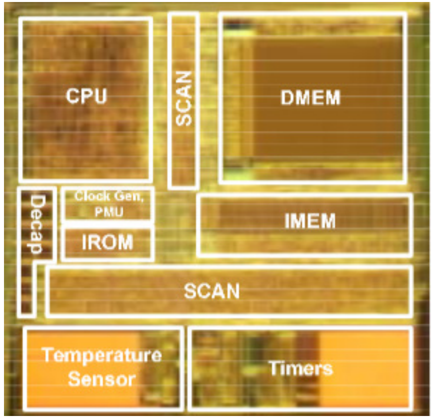






B. Warneke '04

Y.-S. Lin '08



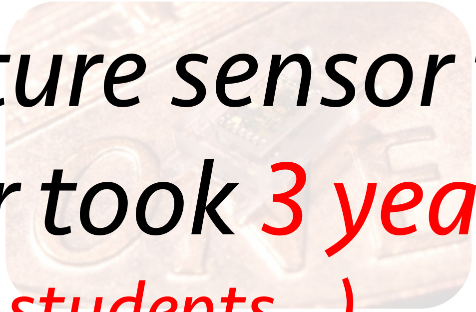
2008 The Phoenix Project

Monolithic designs let us push the envelope, but modularity must return to grow beyond a niche space

- Phoenix
  - World's lowest power computer (30 pW / 300 nW)
  - Basically a temperature sensor
- Intraocular Pressure
  - Collaboration for glaucoma health
  - A pressure sensor

2008

2011



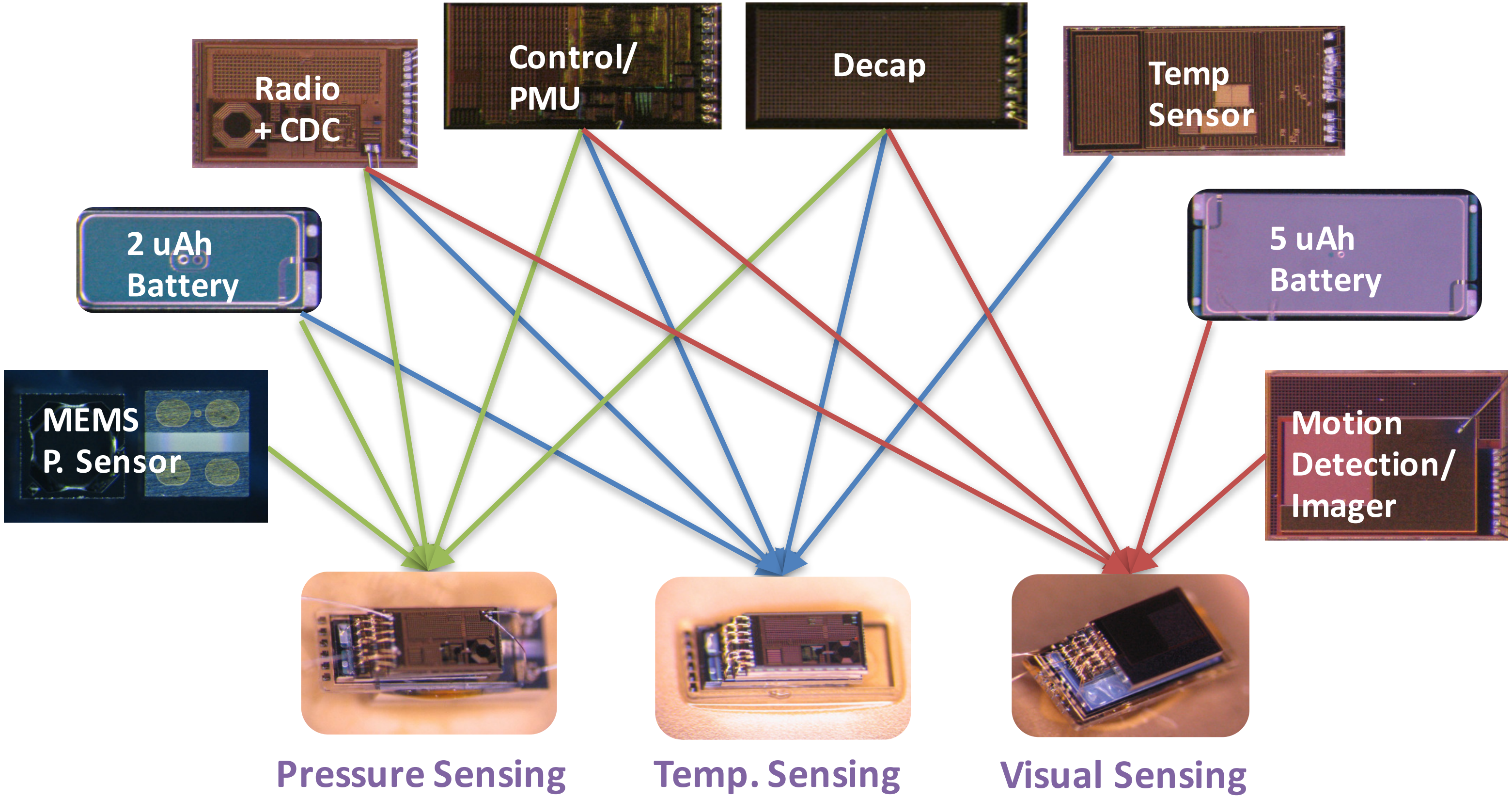
From mm-scale temperature sensor to mm-scale pressure sensor took 3 years

(and another half-dozen PhD students…)

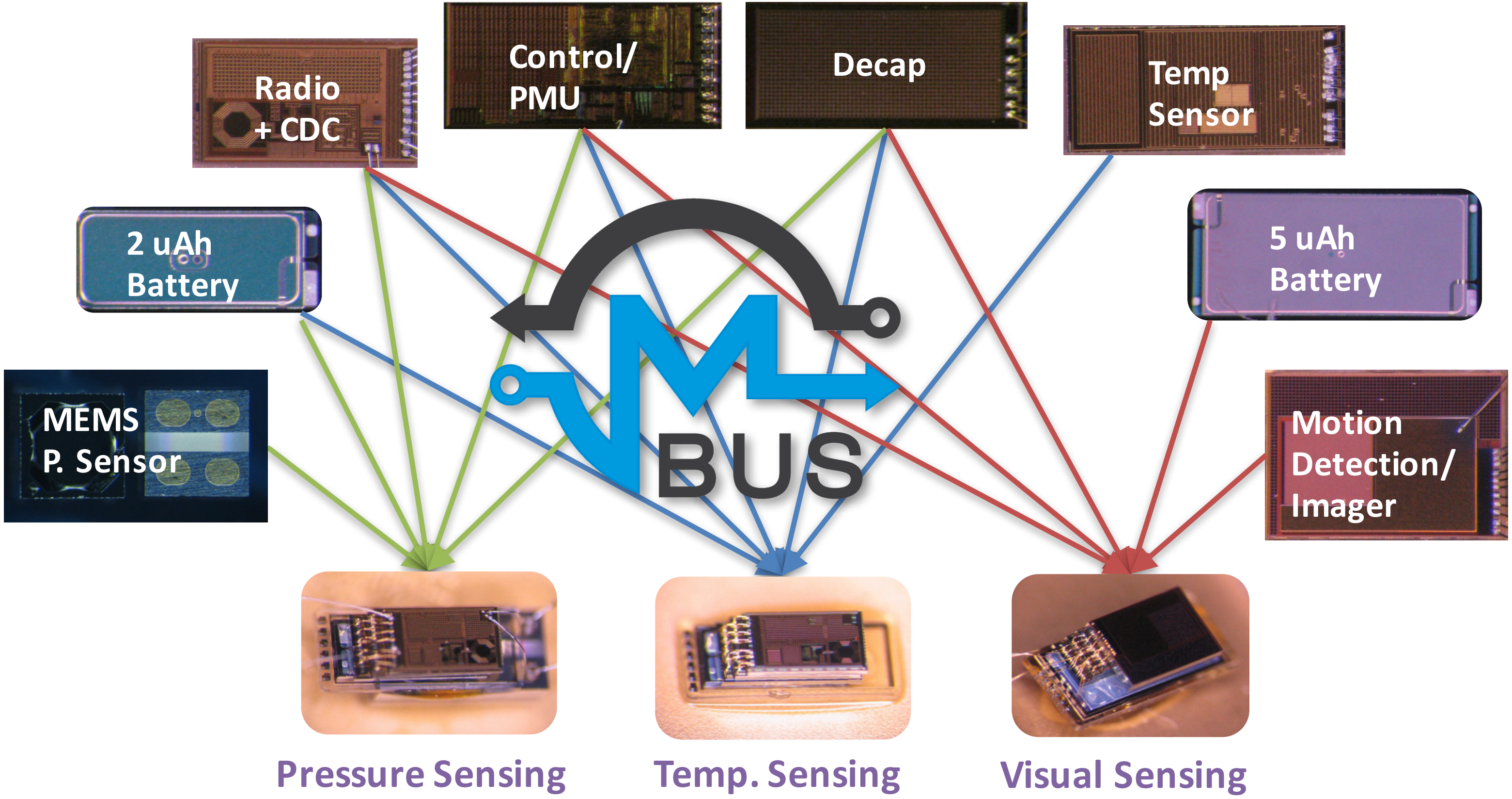
The Phoenix Processor: A 30pW Platform for Sensor Applications, Mingoo Seok, Scott Hanson, Yu-Shiang Lin, Zhiyoong Foo, Daeyeon Kim, Yoonmyung Lee, Nurrachman Liu, Dennis Sylvester, David Blaauw, VLSI '08

A Cubic-Millimeter Energy-Autonomous Wireless Intraocular Pressure Monitor, Gregory Chen, Hassan Ghaed, Razi-ul Haque, Michael Wieckowski, Yejoong Kim, Gyouho Kim, David Fick, Daeyeon Kim, Mingoo Seok, Kensall Wise, David Blaauw, Dennis Sylvester, ISSCC '11

Today, we are able to mix and match components to synthesize systems



However, we had to invent a new system interconnect to be able to do it



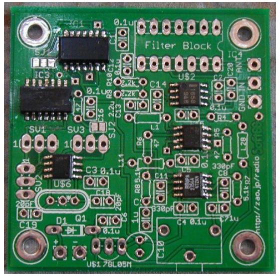
- A Brief History of “Smart Dust”
- Micro-Scale Composition Architecture
- Energy
- Communication
- Debugging

Comparing system composition strategies between traditional motes and mm-scale motes

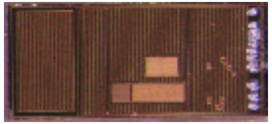
- How to build a mote to sense temperature today?
  - Processor

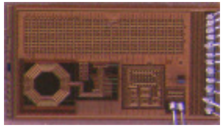
  - Temperature Sensor

  - Radio



- How to build a mm-scale mote to sense temperature?





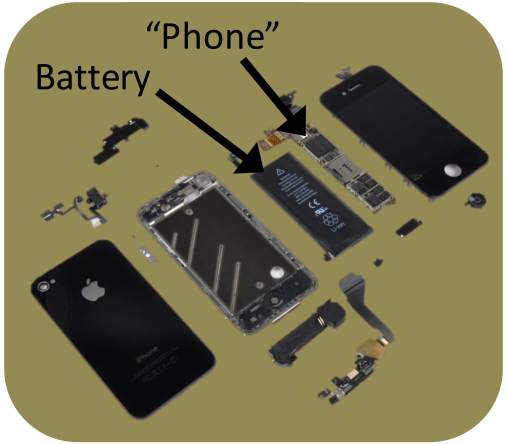
```
temp_sense t1 (
        …
        .sample_valid_out (to_proc_t1_sample_valid),
        .sample_out (to_proc_t1_sample),
        );
processor p1 (
        …
        .temp_valid_in (to_proc_t1_sample_valid),
        .temp_in (to_proc_t1_sample),
        );
```

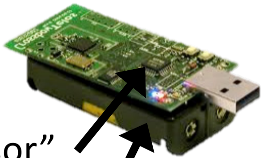
We have solutions for composing embedded systems... SPI and I^2C

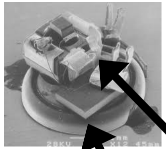
- Nearly every microcontroller has both
- Nearly every peripheral has one or the other
- Very few use anything else
  - (except maybe UART)
- *So why don't they work for mm-scale systems?*

Millimeter-scale systems are *small*

Node volume is dominated by
energy storage





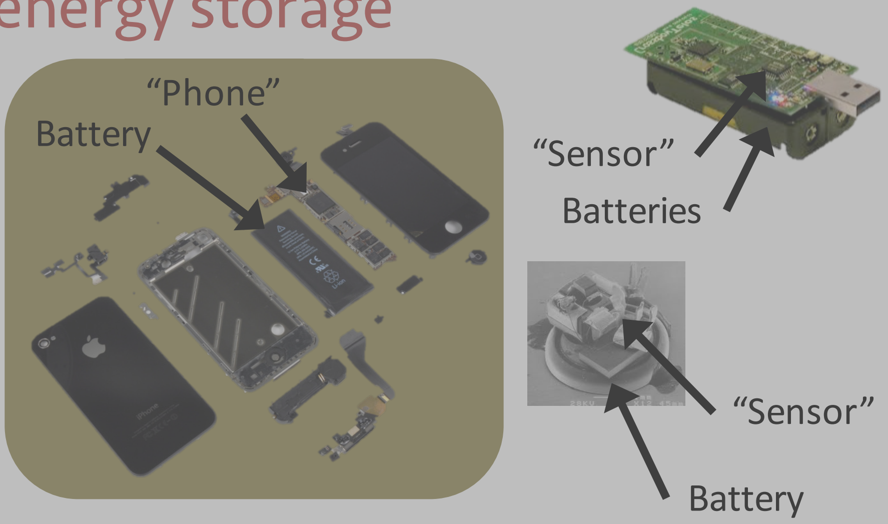


And volume is **shrinking cubically**

Budget 10's µW active, 10's nW sleep, DC 0.1%

Millimeter-scale systems are *small*

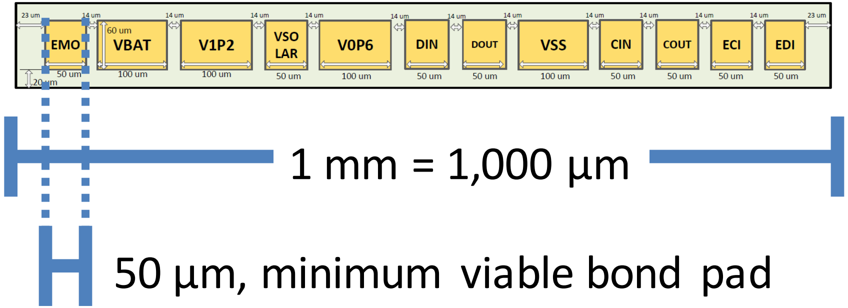
Node volume is dominated by energy storage



And volume is **shrinking cubically**

Budget 10's µW active, 10's nW sleep, DC 0.1%

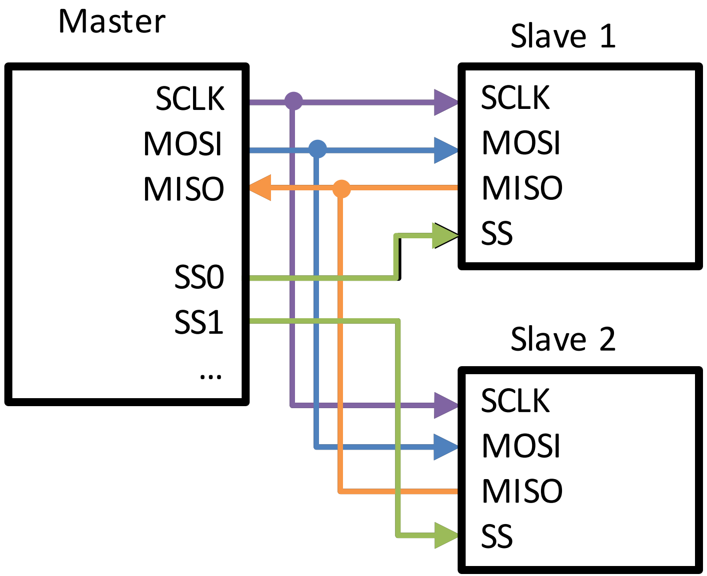
I/O pads begin to account for non-trivial percentage of node **surface area**



16-20 maximum I/O pins for 3D stacking

What is wrong with how are systems composed today?

- SPI, invented by Motorola in ~1979

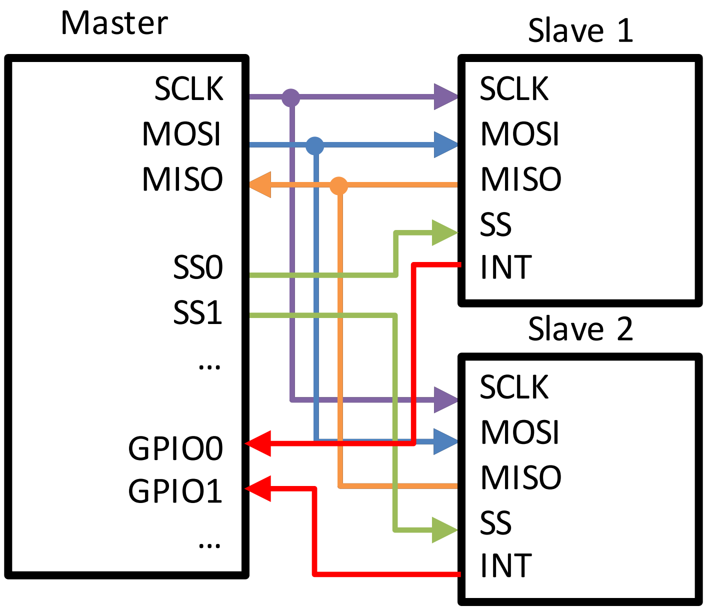


- One master, N slaves
- Shared clock: **SCLK**
- Shared data bus: **MOSI**
- Shared data bus: **MISO**
- One **Slave Select** line per slave

- **Key Properties**
  - One dedicated I/O line **per slave**
  - Master controls all communication

What is wrong with how are systems composed today?

- SPI, invented by Motorola in ~1979

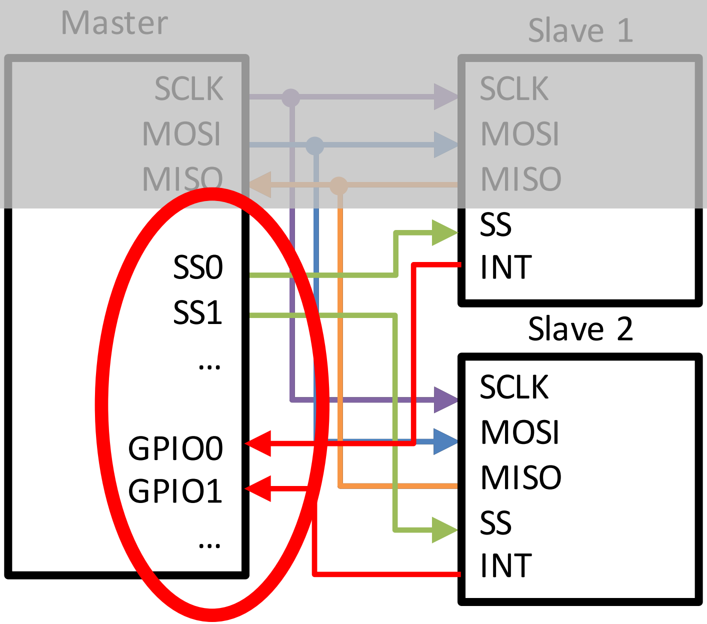


- One master, N slaves
- Shared clock: **SCLK**
- Shared data bus: **MOSI**
- Shared data bus: **MISO**
- One **Slave Select** line per slave
- One **interrupt line** per slave

- **Key Properties**
  - ~~One~~ Two dedicated I/O lines **per slave**
  - Master controls all communication
    - Interrupts must be out-of-band

SPI's I/O overhead and centralized architecture **do not scale** to mm-scale systems

- SPI, invented by Motorola in ~1979

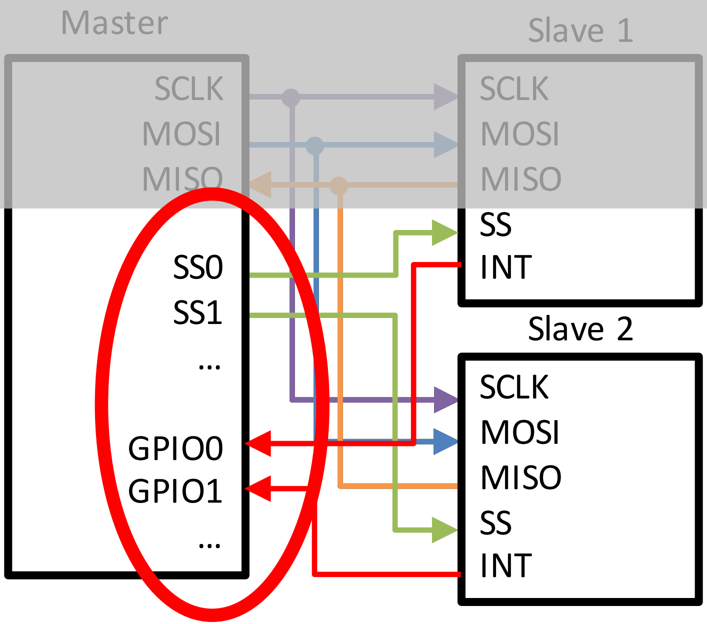


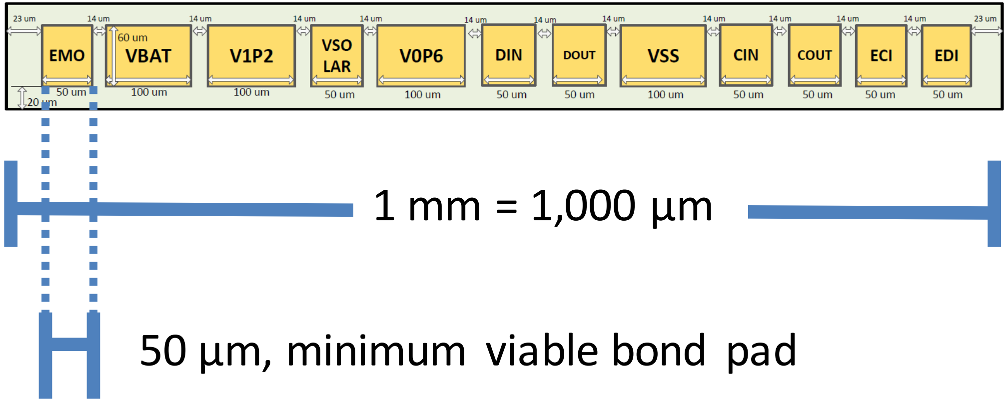
- One master, N slaves
- Shared clock: **SCLK**
- Shared data bus: **MOSI**
- Shared data bus: **MISO**
- One **Slave Select** line per slave
- One **interrupt line** per slave

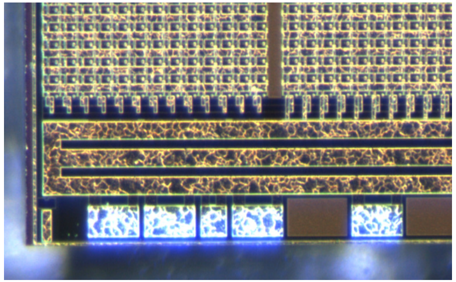
- **Key Properties**
  - ~~One~~ Two dedicated I/O lines **per slave**
  - Master controls all communication
    - Interrupts must be out-of-band

SPI's I/O overhead and centralized architecture **do not scale** to mm-scale systems

- SPI, invented by Motorola in ~1979



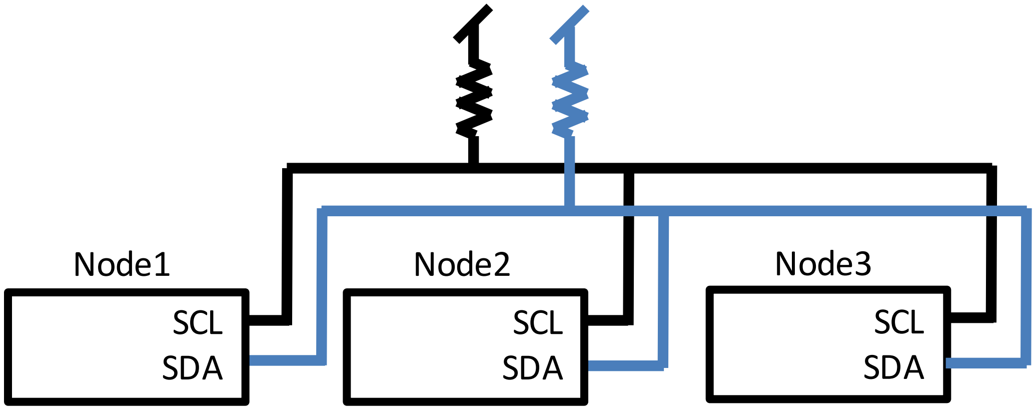




I^2C has fixed I/O requirements and a decentralized architecture

- I^2C, invented by Phillips in 1982
  - Any-to-(m)any on one shared bus

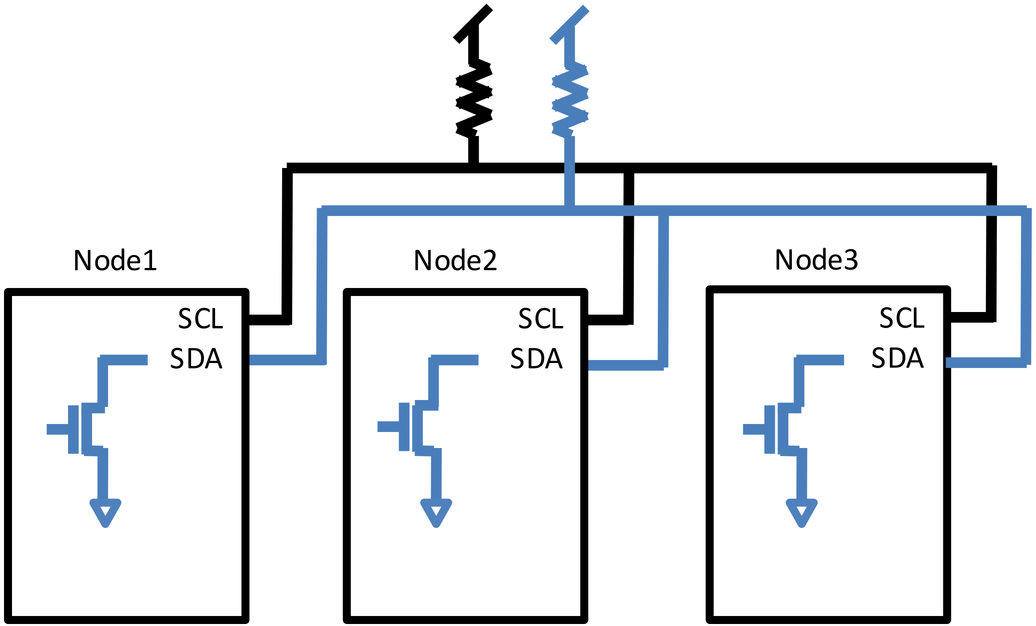
- **Key Properties**
  - Fixed wire count (2)



I^{2}C has fixed I/O requirements and a decentralized architecture

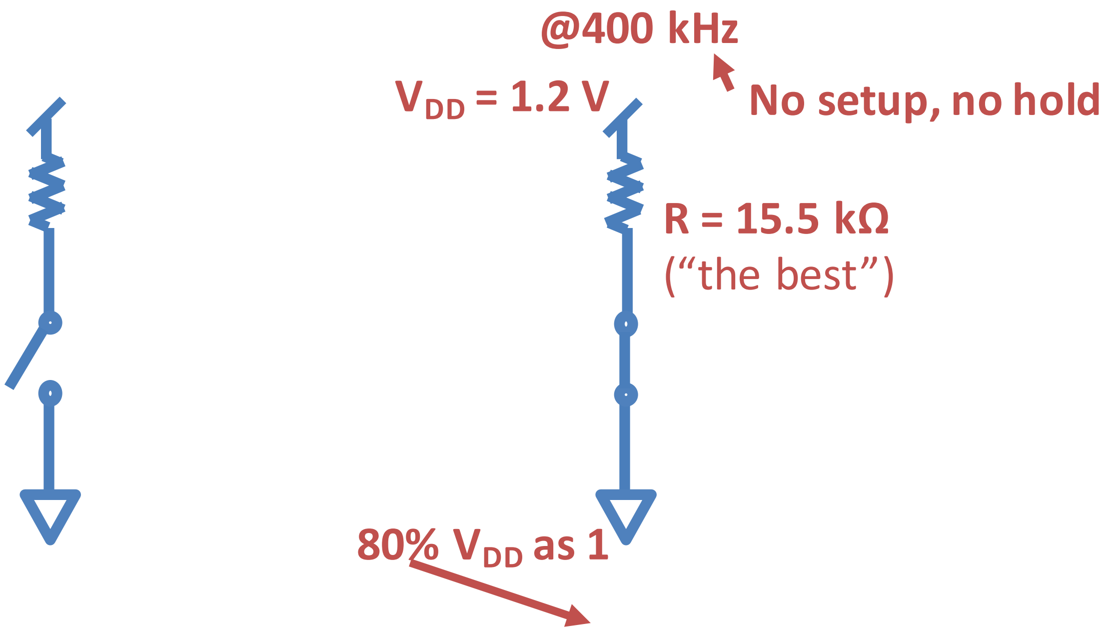
- I^{2}C, invented by Phillips in 1982
  - Any-to-(m)any on one shared bus

- **Key Properties**
  - Fixed wire count (2)
  - Open-collector
    - Multi-master
    - Flow Control



Open-collector (aka wired-AND)

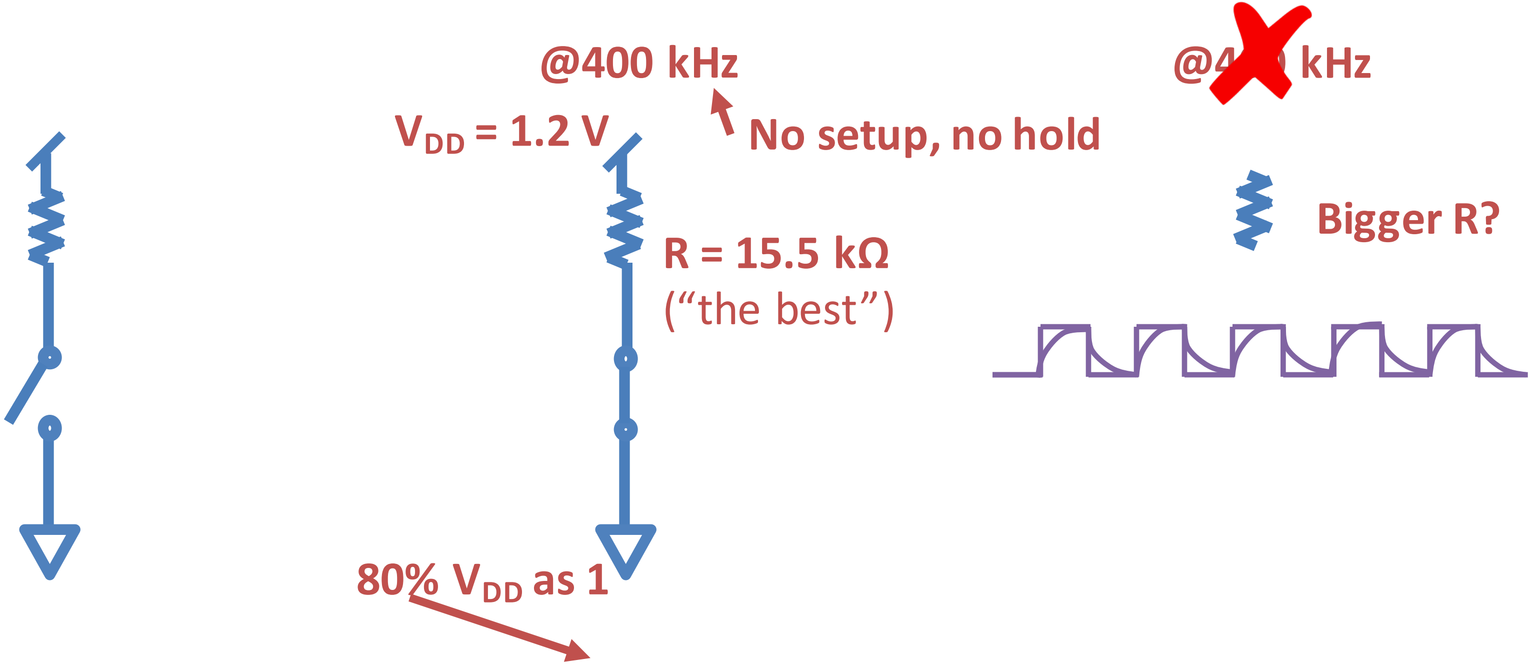
The problem is the energy costs of running an open-collector



Send a "1": 0 J

Send a "0": 174 pJ

The problem is the energy costs of running an open-collector

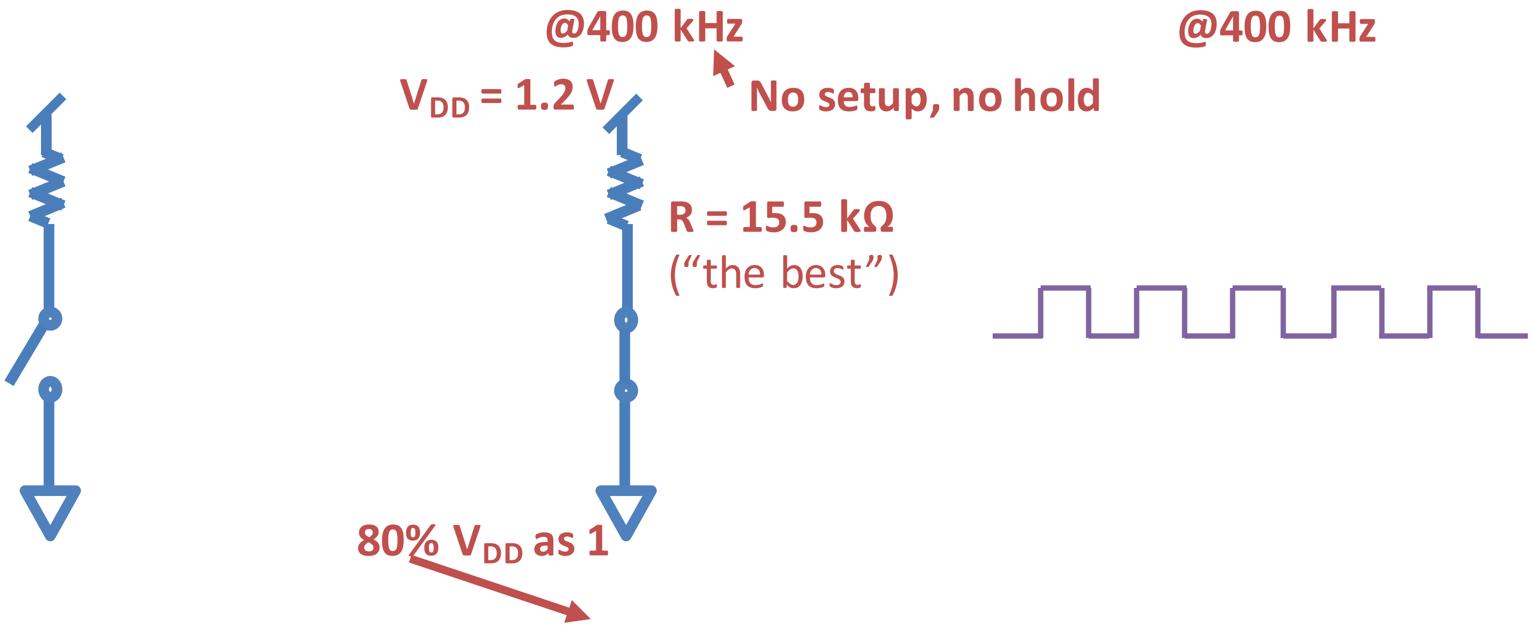


Send a “1”: 0 J

Send a “0”: 174 pJ

SCL Alone: **70 µW**

The problem is the energy costs of running an open-collector



Send a "1": 0 J

Send a "0": 174 pJ

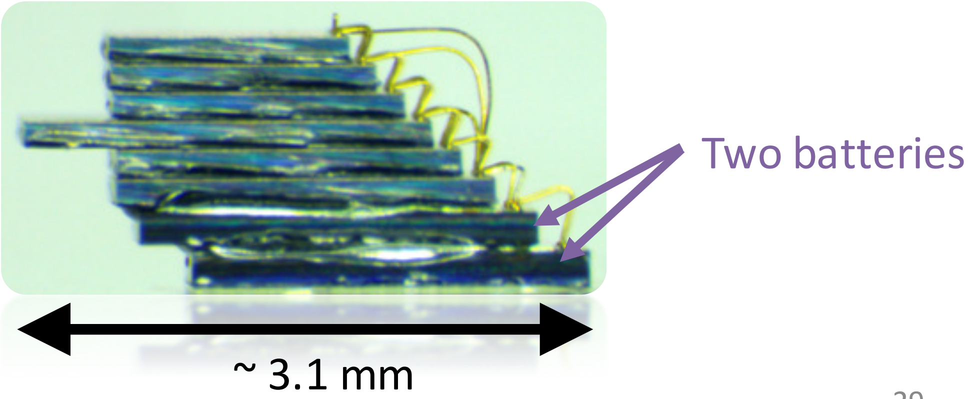
SCL Alone: **70 µW**

The energy demands of open-collectors make them unsuitable for mm-scale systems

@400 kHz

SCL Alone: **70 µW**

- Active Energy Budget: **20 µW**
- Not an arbitrary number:
  - Volume Target
  - Lifetime Target



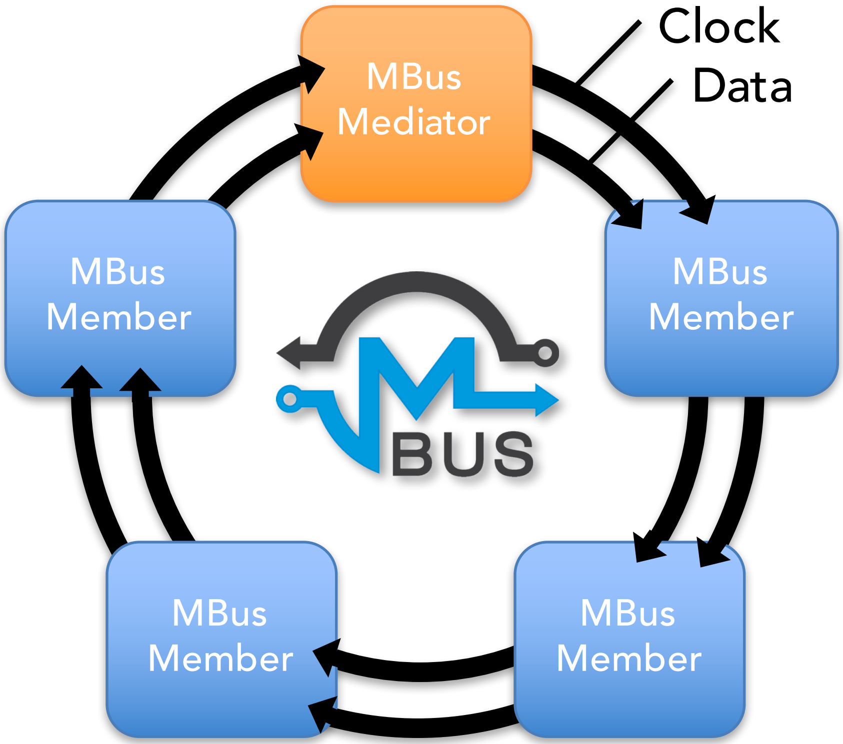
The design and synthesis of modular, millimeter-scale systems requires a new embedded interconnect

MBus is a clean-slate design, built to satisfy interconnect requirements for this and the next generation of motes

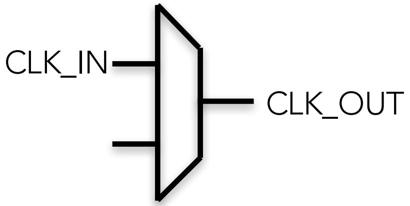
- Fixed pin count (4)
- Minimal standby and active power
- Large global address space
- Multi-master design
- Broadcast message support
- Hardware acknowledgements
- Data independent behavior
- Modest protocol overhead
- Power aware design

MBus: An Ultra-Low Power Interconnect Bus for Next Generation Nanopower Systems
Pat Pannuto, Yoonmyung Lee, Ye-Sheng Kuo, ZhiYoong Foo, Benjamin Kempke, Gyouho Kim, Ronald G Dreslinski, David Blaauw, and Prabal Dutta
Proceedings of the 42nd International Symposium on Computer Architecture (ISCA '15)

MBus in a nutshell



- Ring Topology
- 2 lines – 4 I/O per node
  - Clock
  - Data
- Transaction oriented
  - Arbitration
  - Address Transmission
  - Data Transmission
  - Interjection
  - Control (ACK/NAK)
- "Shoot-Through"

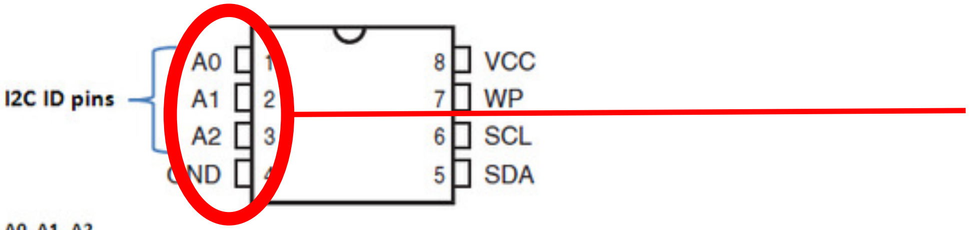


Embedded interconnect technology has not changed in over 30 years

- If we re-examine…

Embedded interconnect technology has not changed in over 30 years

- If we re-examine…
  - Addressing



A0, A1, A2
The A0, A1 and A2 are the device address inputs that are hardwired or left not connected for hardware compatibility with the 24C16. When pins are hardwired, as many as eight 32K/64K devices may be addressed on a single bus system. When the pins are not hardwired, the default values of A0, A1, and A2 are zero.

I2C slave address

BIT	7	6	5	4	3	2	1	0
	1	0	1	0	A2	A1	A0	R/$\overline{W}$

- Requires I/O not available
- Makes packaging assumptions
  - Systems will not always be PCBs
  - Routing may not be easy
    - 3D stack
    - Flip-chip + TSVs

Embedded interconnect technology has not changed in over 30 years

- If we re-examine…
  - Addressing

3 Options

Short static addresses and allow device conflicts

Long static addresses to avoid device conflicts

Non-static addresses

MBus does all 3

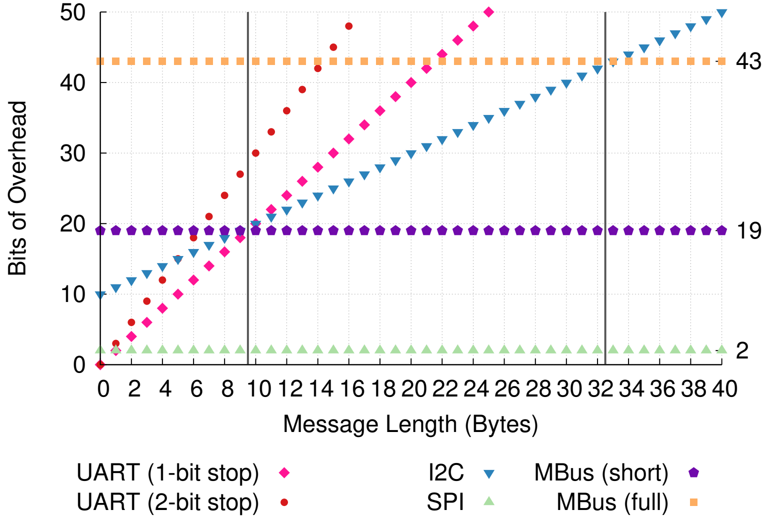
4-bit: Static short prefixes (device class)

24-bit: Static long prefixes (unique device ID)

4-bit: Runtime enumeration protocol (replaces short prefix)

Embedded interconnect technology has not changed in over 30 years

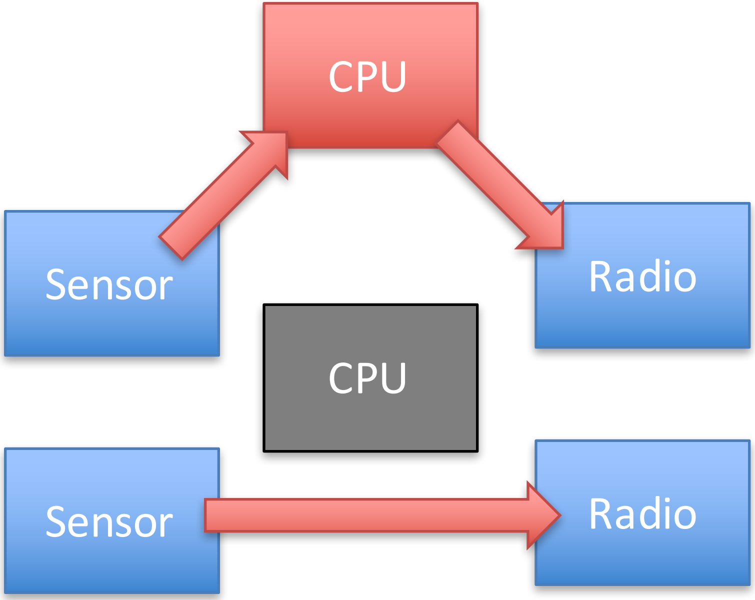
- If we re-examine…
  - Addressing
  - Acknowledgements



- I^2C acknowledges every byte
  - How often do NAKs happen?
    - To a random byte?
  - 12.5% overhead
- MBus ACKs transactions
  - Receiver can interject message

Embedded interconnect technology has not changed in over 30 years

- If we re-examine…
  - Addressing
  - Acknowledgements
  - Centralized vs Decentralized



Over 2x energy consumption with centralized

7% improvement in lifetime for a prototypical “sense & send” application

Existing interconnect technology cannot service forthcoming motes, and may be hindering current motes

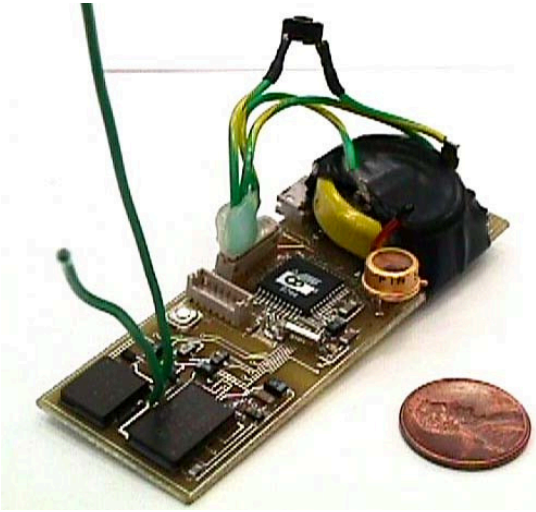
- Leading embedded interconnects are showing flaws
  - Unbounded I/O requirements
  - Burdensome runtime energy demands
  - Limited addressing schemes
  - Inefficient or nonexistent acknowledgments
  - Restrictive centralized architecture
- MBus addresses all of these issues and adds new features

Energy is a looming system synthesis problem

- A Brief History of “Smart Dust”
- Micro-Scale Composition Architecture
- Energy
- Communication
- Debugging

Motes embraced "dark silicon" first

- Powering off regions of a chip has become commonplace
- Motes power off regions of a *system*
  - Requires coordination strategy
  - Historically the CPU
  - Putting peripherals into standby mode was good enough for early, battery-backed motes…



1999

- 900 MHz radio
- Magnetometer
- Accelerometer
- Light
- Temperature
- Pressure

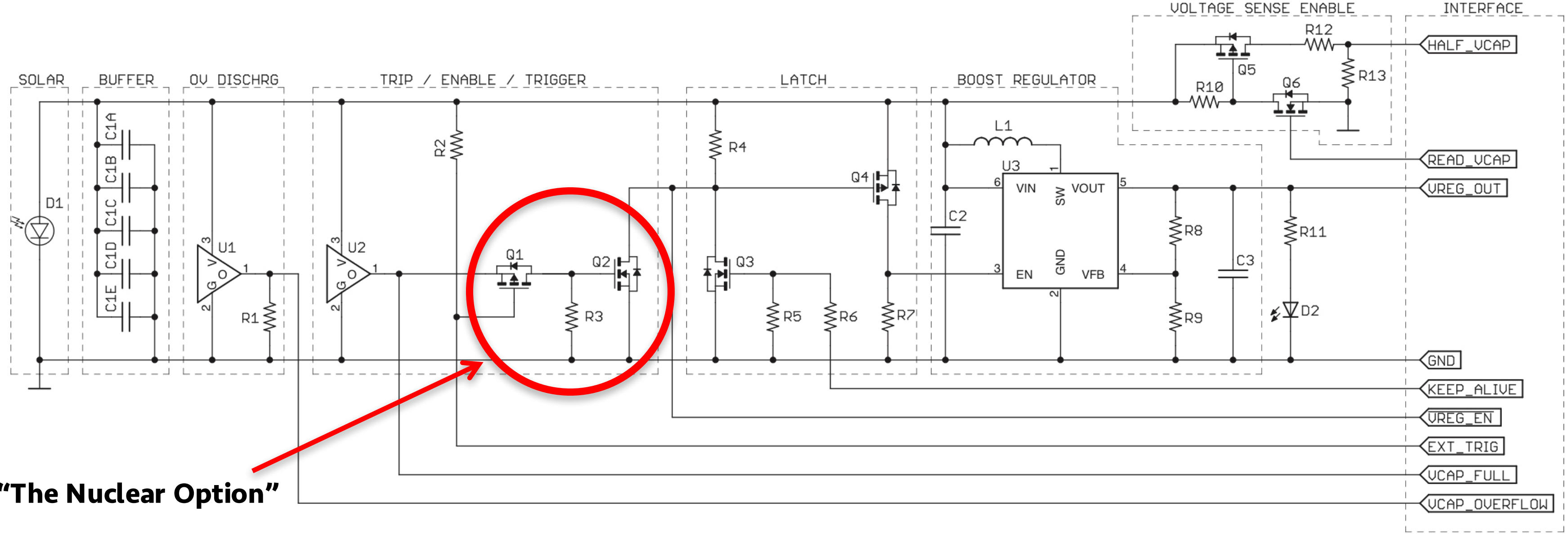
One week lifetime in continuous operation, 2 years w/ 1% duty cycle

Energy-critical systems push from “dark silicon” to “pitch black” silicon

- Clock-gating
  - Stop driving the clock tree of regions of a chip
  - Eliminates switching power, but not static leakage
  - “Dark Silicon”
- Power-gating
  - Switch off power to regions of a chip
  - Eliminates all chip leakage
  - “Pitch Black Silicon”
- M3 systems aggressively power-gate to reach energy budget

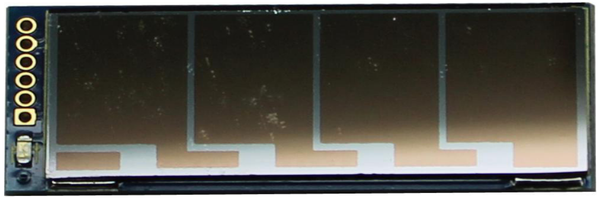
Energy-harvesting motes embrace "pitch black" silicon

Can we build a (useful) battery-free mote? Mote platform?



"The Nuclear Option"

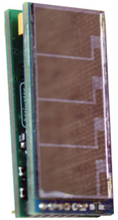
- Zero static leakage
- Very little control

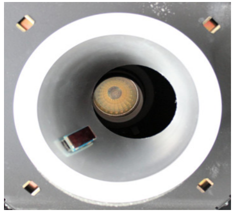


L. Yerva, B. Campbell, A. Bansal, T. Schmid, and P. Dutta, "Grafting Energy-Harvesting Leaves onto the Sensornet Tree," In *Proceedings of the 11th International Conference on Information Processing in Sensor Networks* (IPSN'12), Beijing, China, Apr. 16-20, 2012.

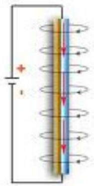
Slide borrowed from Prabal Dutta

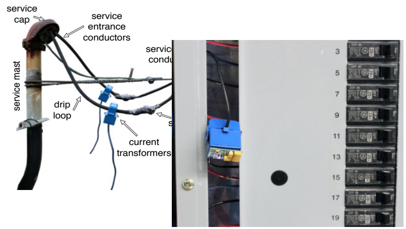
Monjolo sensors—the harvester *is* the sensor





L. Yerva, B. Campbell, A. Bansal, T. Schmid, and P. Dutta, "Grafting Energy-Harvesting Leaves onto the Sensornet Tree," In *Proceedings of the 11th International Conference on Information Processing in Sensor Networks* (IPSN'12), Beijing, China, Apr. 16-20, 2012.

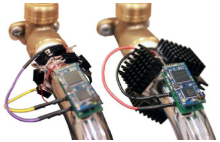




Samuel DeBruin, Bradford Campbell, and Prabal Dutta, "Monjolo: An Energy-Harvesting Energy Meter Architecture," In *Proceedings of the 11th ACM Conference on Embedded Networked Sensor Systems* (Sensys'13), Rome, Italy, Nov. 11-14, 2013.

Monjolo: A Portuguese water hammer

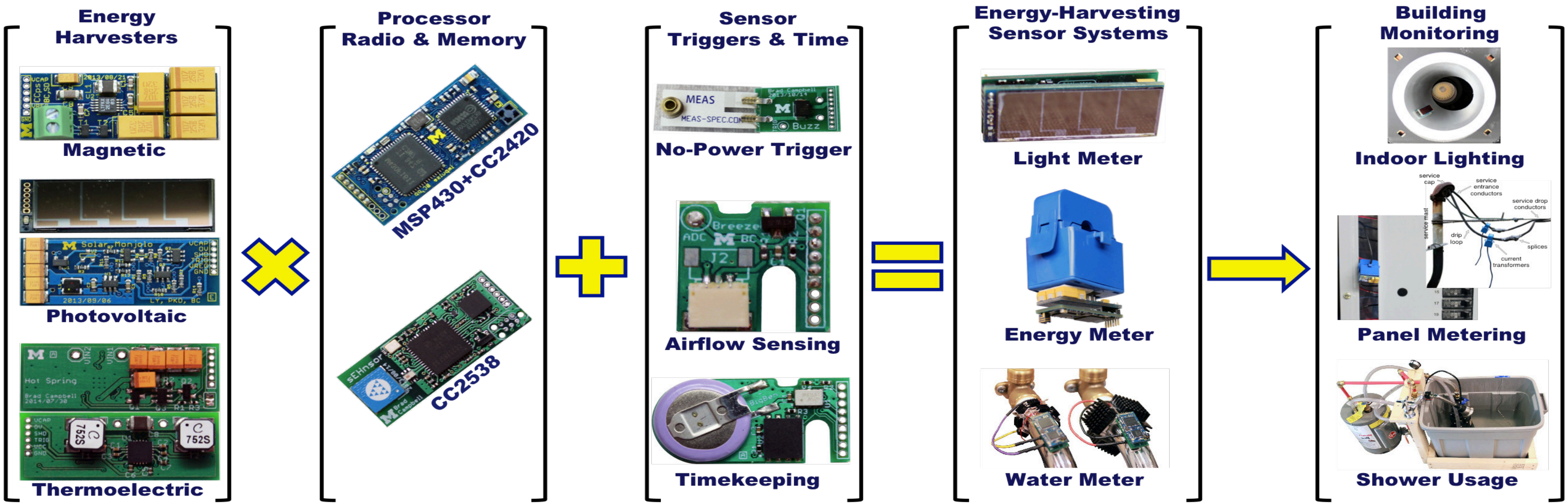
- Hammer rate ⬄ flow rate
- Fixed water quanta / strike
- Non-linearity with high flow rate



B. Campbell, B. Ghena, and P. Dutta, "Energy-Harvesting Thermoelectric Sensing for Unobtrusive Water and Appliance Metering," In *Proceedings of the 2nd International Workshop on Energy Neutral Sensing Systems* (ENSsys'14), Memphis, TN, Nov. 6, 2014.

Slide borrowed from Prabal Dutta

The Monjolo principle decomposes into an array of components that build a diverse set of energy-harvesting sensors



HW/SW designs available online:

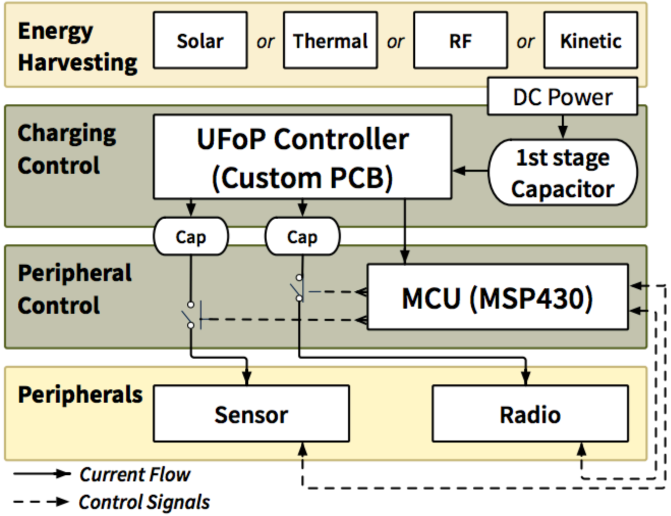
[1] http://lab11.eecs.umich.edu/pcb.html

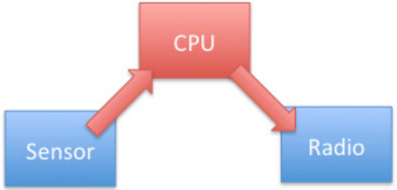
[2] https://github.com/lab11

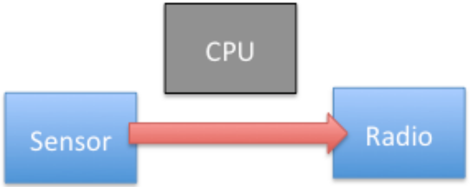
Slide borrowed from Prabal Dutta

Systems are not required to be all or none, however

- "Federated Energy"
  - Harvest an independent store for each component
  - Completely cut power when component is not in use
  - But this system cannot cut power to the MCU
  - MCU-Federated energy control







Josiah Hester, Lanny Sitanayah, and Jacob Sorber. 2015. Tragedy of the Coulombs: Federating Energy Storage for Tiny, Intermittently-Powered Sensors. In Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems (SenSys '15)

Embedded interconnect technology has not changed in over 30 years

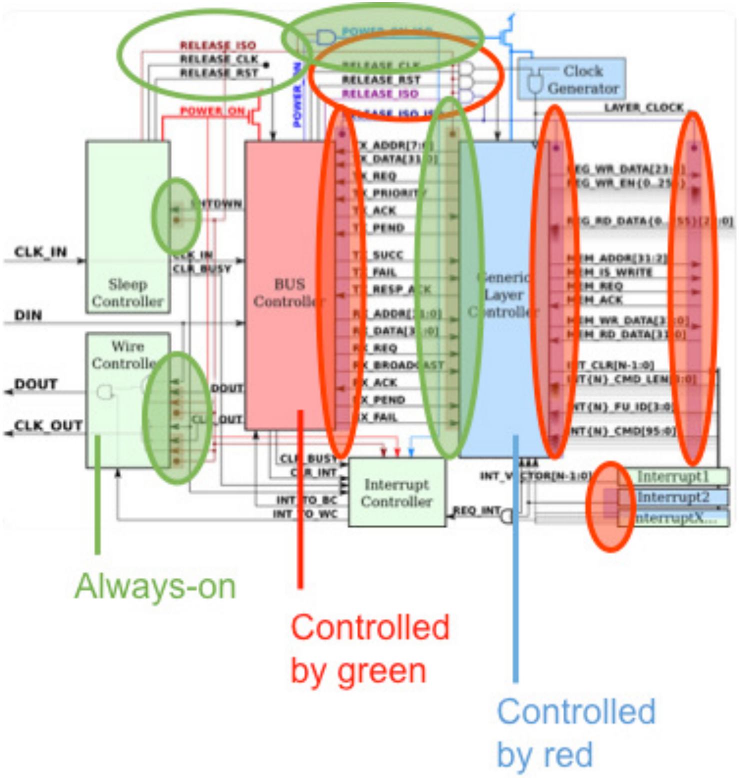
- If we re-examine…
  - Addressing
  - Acknowledgements
  - Centralized vs Decentralized
  - **Peripheral power state is an interconnect concern**

Our mm-scale systems leverage decentralized, autonomous power management via MBus

- Key Challenges
  - Energy efficient multi-master systems are a distributed state problem
  - Cold-booting silicon is a hard circuits problem
    - Because there is no stable clock
- MBus Key Observation
  - A component is either already on, or can leverage an incoming message to wake itself

MBus introduces three, transparent, hierarchical power domains to maximize efficiency

- Provide an "always-on" abstraction
  - Interconnect fabric powers on components upon receipt of a message
  - Transparent to callers and chip designers



As motes continue to push the boundaries of lifetime, harvesting, and volume, every microjoule is precious

- Circuits advancements will continue to provide benefits
  - Non-volatile RAM will fall in price and area
  - Accelerators will replace compute-intensive tasks
  - Finer-grained internal power-gating
  - Diminishing returns...
- But there are system-level constructs wasting energy today
  - Leveraging peripheral power-gating
  - Coalescing wakeups
  - Clock domain controls
  - ...

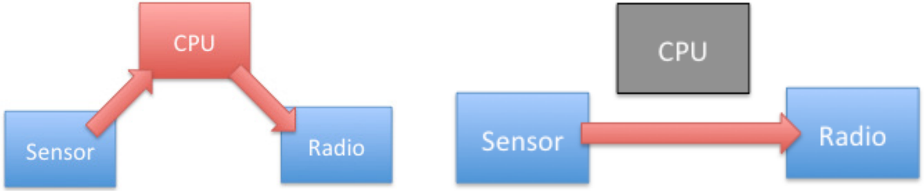
- A Brief History of “Smart Dust”
- System Composition Architecture
- Energy
- Communication
  - “Internal”
  - “External”
- Debugging

Embedded interconnect protocols have not changed in 30 years either

- In fact, they do not exist at all
- Q: How do you read a register on a sensor?
- Q: How do you write to a radio’s packet buffer?

MPQ: A (mostly) PHY-independent protocol to standardize reading and writing registers and memory

- **Motivations**
  - Simplicity across the ecosystem
  - Enabling *reliable* direct peripheral communication



  - Transition CPU from active overseer to passive coordinator
    - Configure all peripherals at boot, then power off self
    - Distributed equivalent of many COTS chips (Sleepwalking, uDMA, etc)

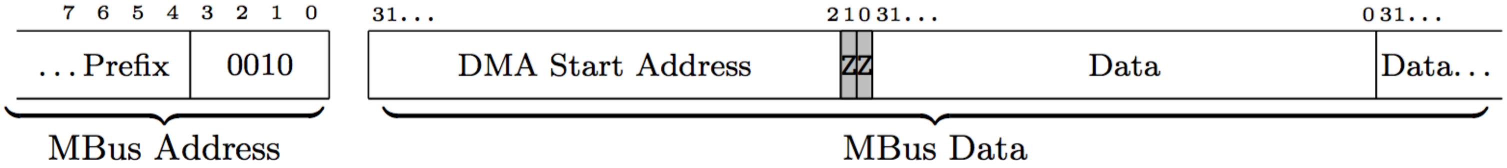
Primitives

- Read/Write Register
- Block DMA transfer
- "Streaming" memory

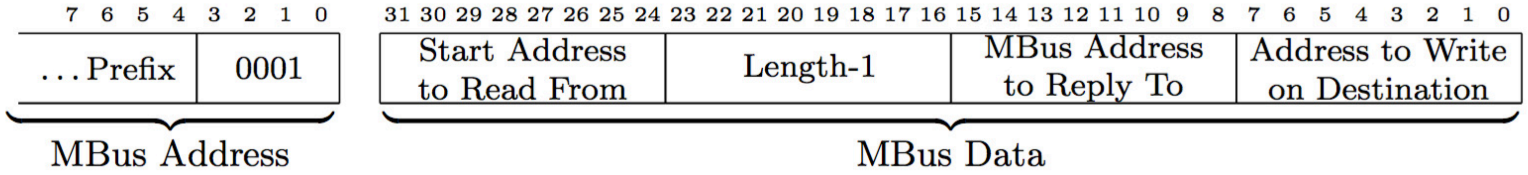
MPQ repurposes the 4 LSB of the address as a function identifier

- Could also use the first 4 bits of data if address bits are unavailable
- Embedded write addresses for read commands enable powerful command synthesis

6.3.1 Memory Bulk Write



6.2.2 Register Read



An interesting quirk begets an interesting question

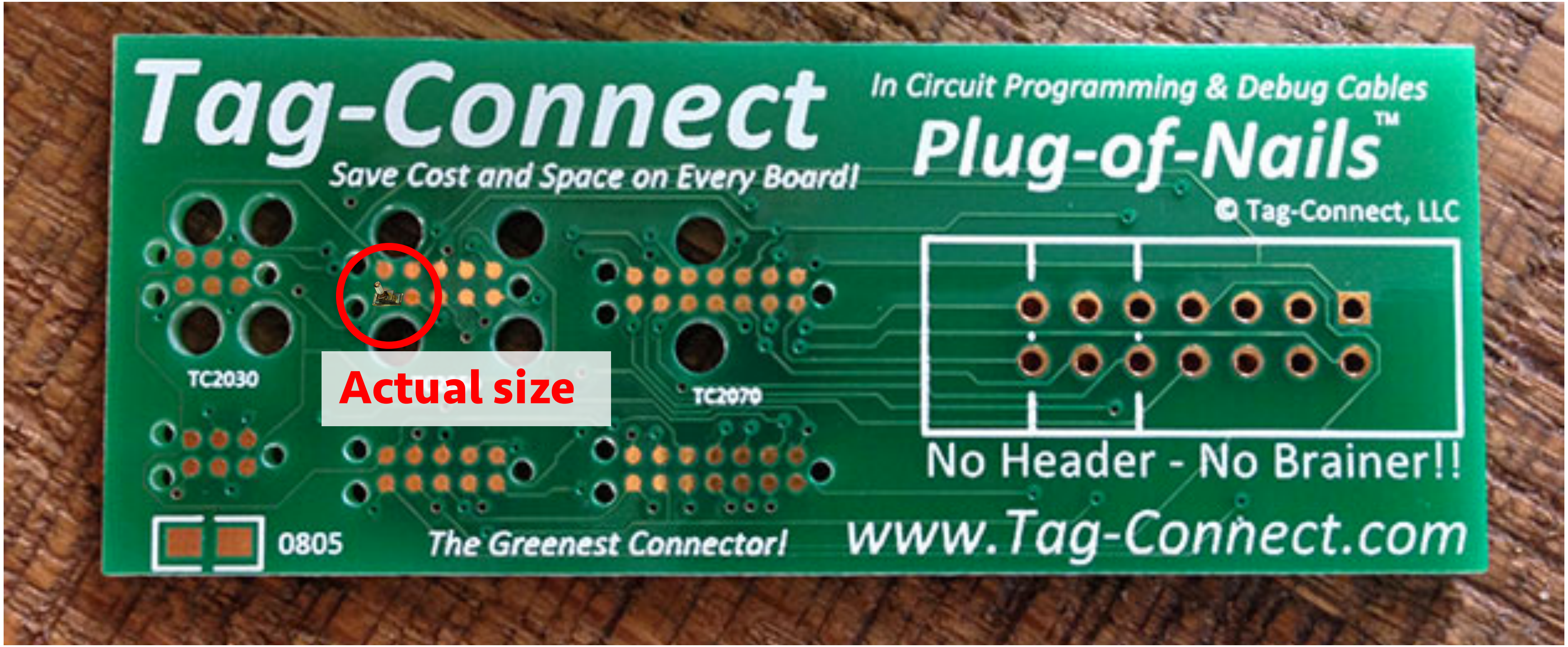
- M3 CPU chip has only SRAM
  - Loses state (and programming) on complete power loss
- Separate flash chip programs CPU on boot
  - Actually, it is a state machine issuing arbitrary DMA messages
- No CPU needed for a sense, send, and forget application
  - As peripherals become more capable, what is the role of the CPU?

Perhaps it is time for a generic protocol on top of major embedded buses

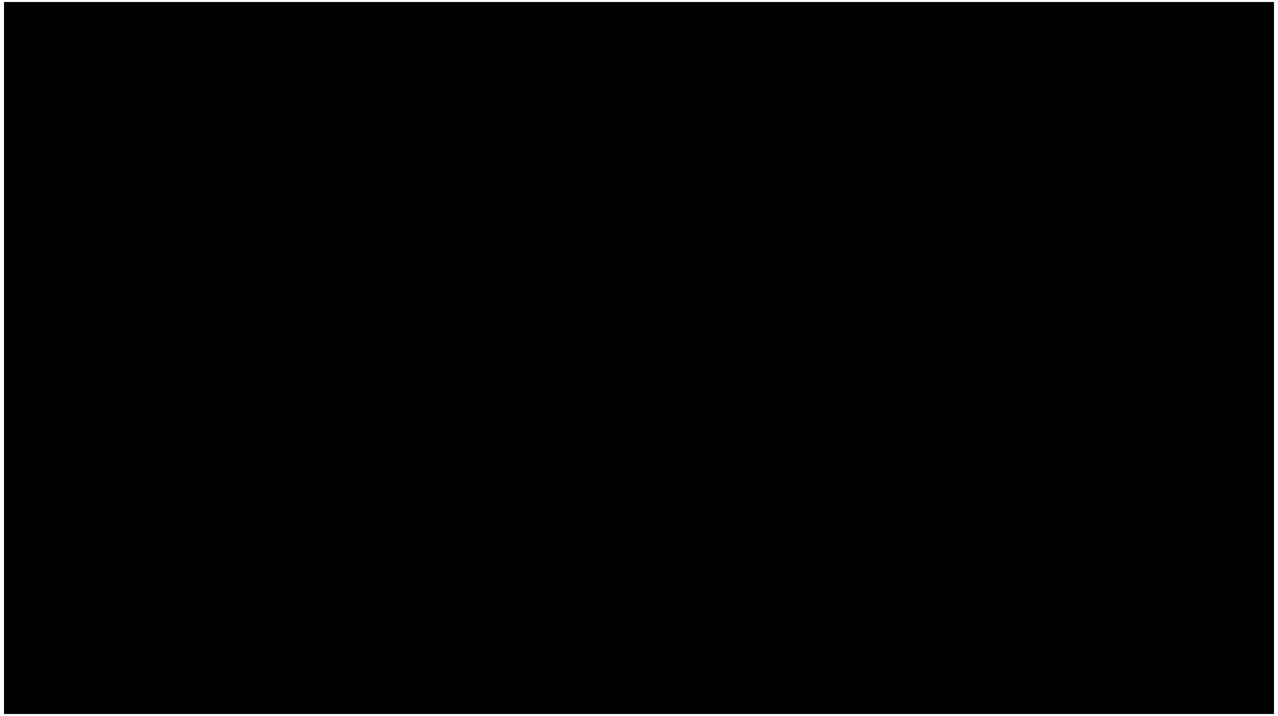
- Improve the programmability and utility of mostly-fixed-function peripheral frontend
- Proving powerful for synthesis in M3 ecosystem
- Easy to phase-in
  - No backward compatibility because there's nothing today!

- A Brief History of “Smart Dust”
- System Composition Architecture
- Energy
- Communication
  - “Internal”
  - “External”
- Debugging

Q: How do you program a 1mm^3 mote?

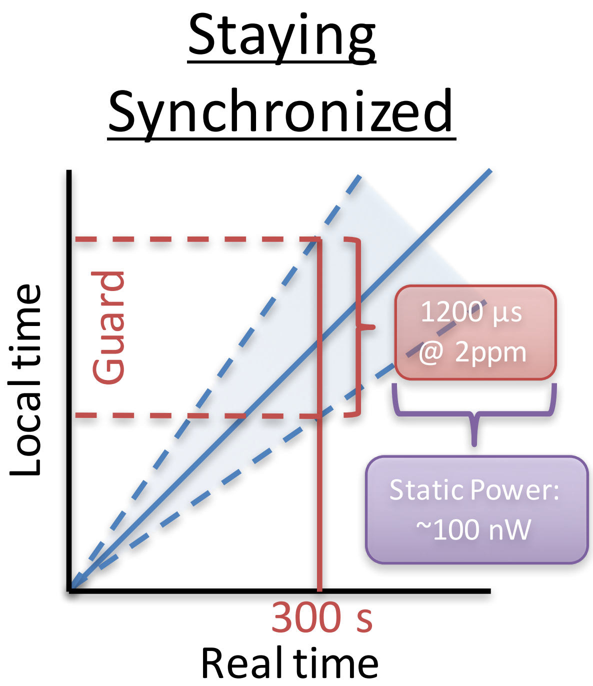


A 695 pW, always-on visible light communication receiver

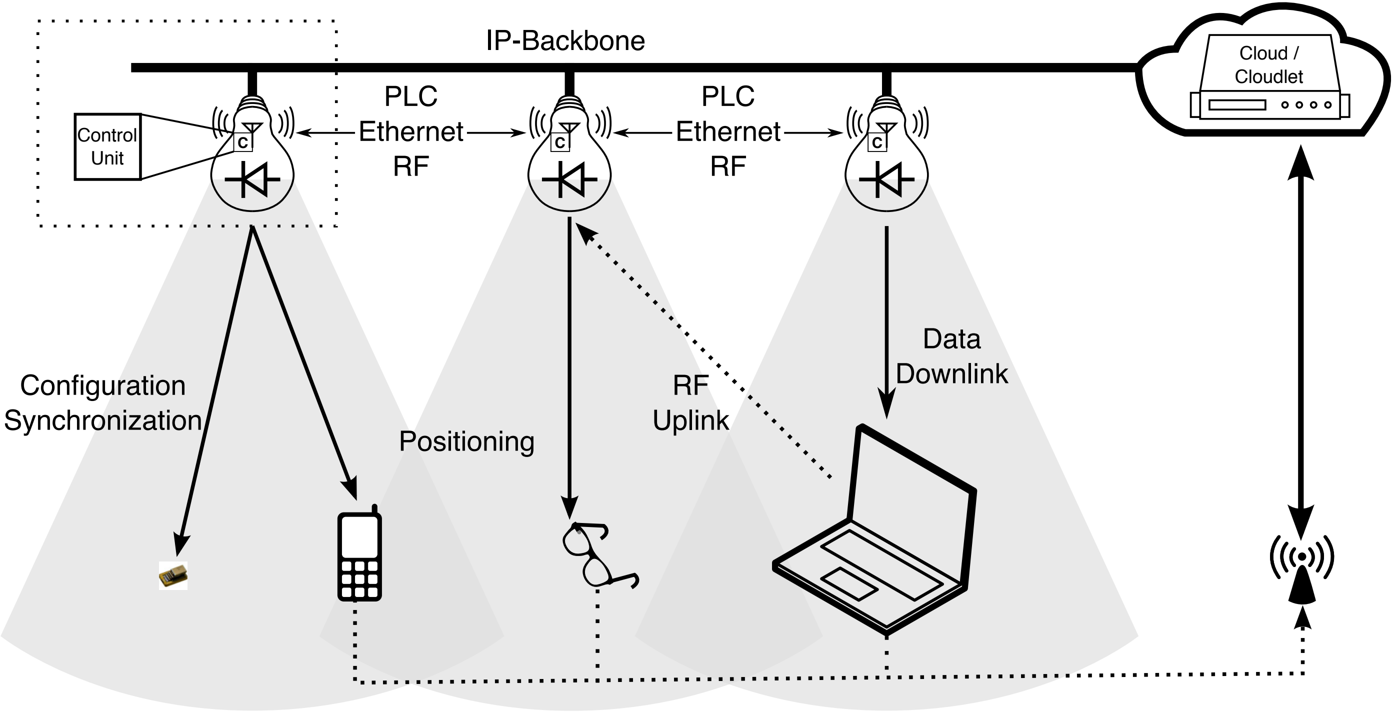


A sub-nanowatt receiver opens interesting possibilities for traditional motes

- Idle listening dominates energy budget of many MACs
- Can we decouple synchronization from communication?

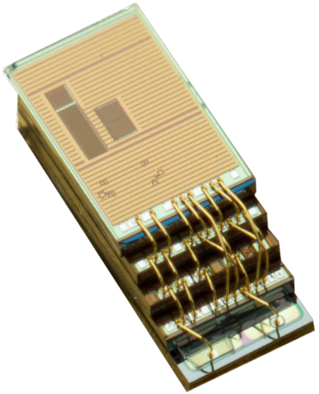


All this begs the question, what else can we do with an intelligent lighting infrastructure?



System Architecture Directions for a Software-Defined Lighting Infrastructure
Ye-Sheng Kuo, Pat Pannuto, and Prabal Dutta
1st ACM Workshop on Visible Light Communication Systems (VLCS '14)

The capabilities of VLC receivers vary greatly



Diffusing (single photodiode)

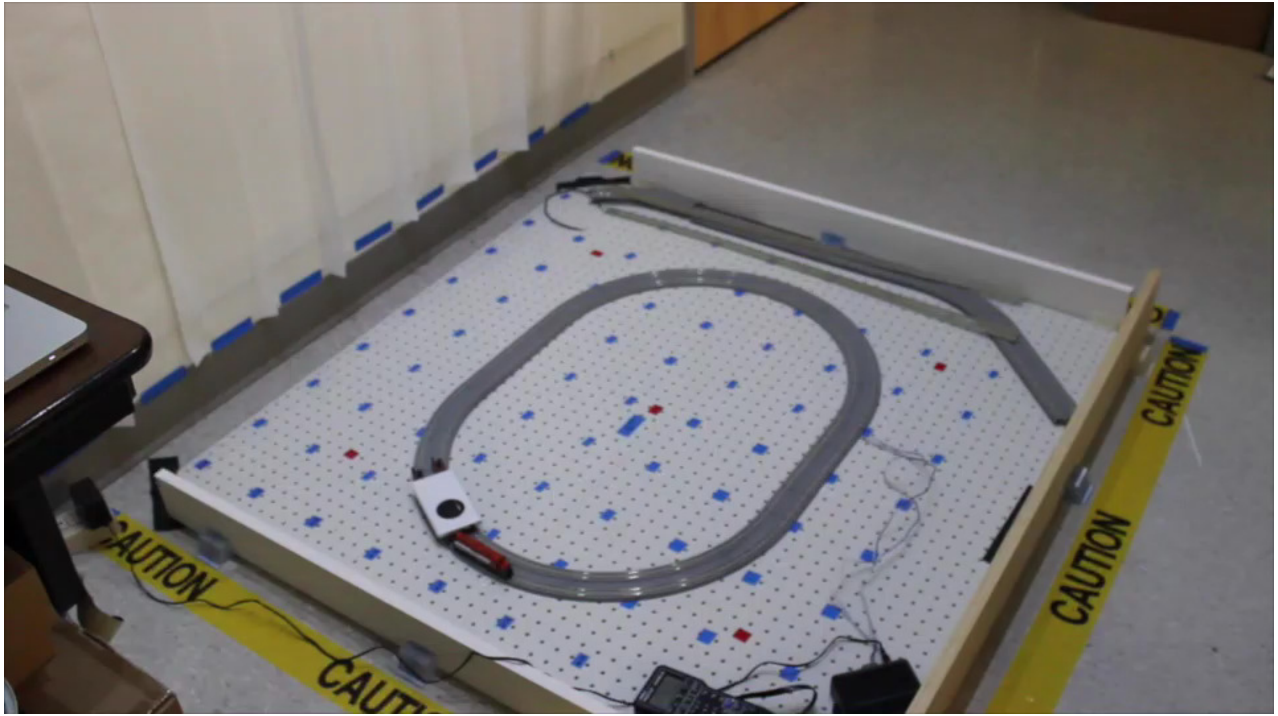
- Simpler devices

  → Less computation capability
- Cannot distinguish transmitters
- Requires temporal signal diversity

Array (multiple photocell)

- Can distinguish transmitters
- Temporal or spatial diversity

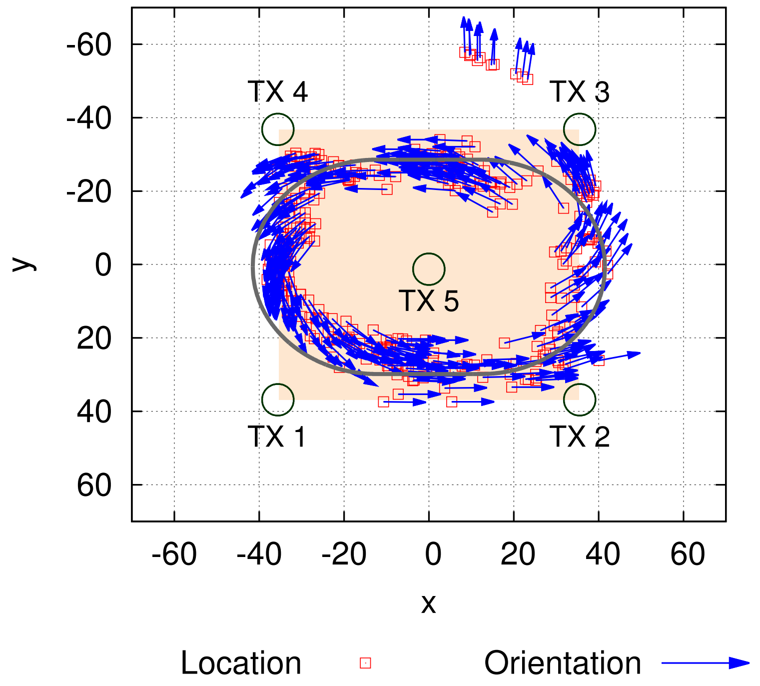
Luxapose:
Astral navigation with smart phones and smart lights

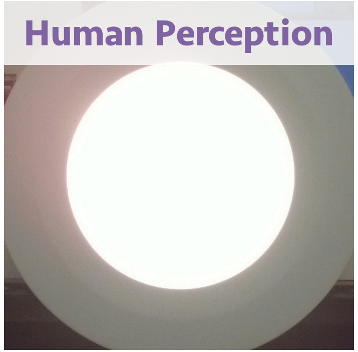


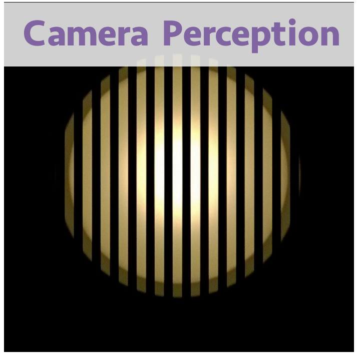
Luxapose: Indoor Positioning with Mobile Phones and Visible Light (MobiCom'14)

Ye-Sheng Kuo, Pat Pannuto, Ko-Jen Hsiao, and Prabal Dutta

Luxapose enables high-fidelity indoor localization with impending infrastructure





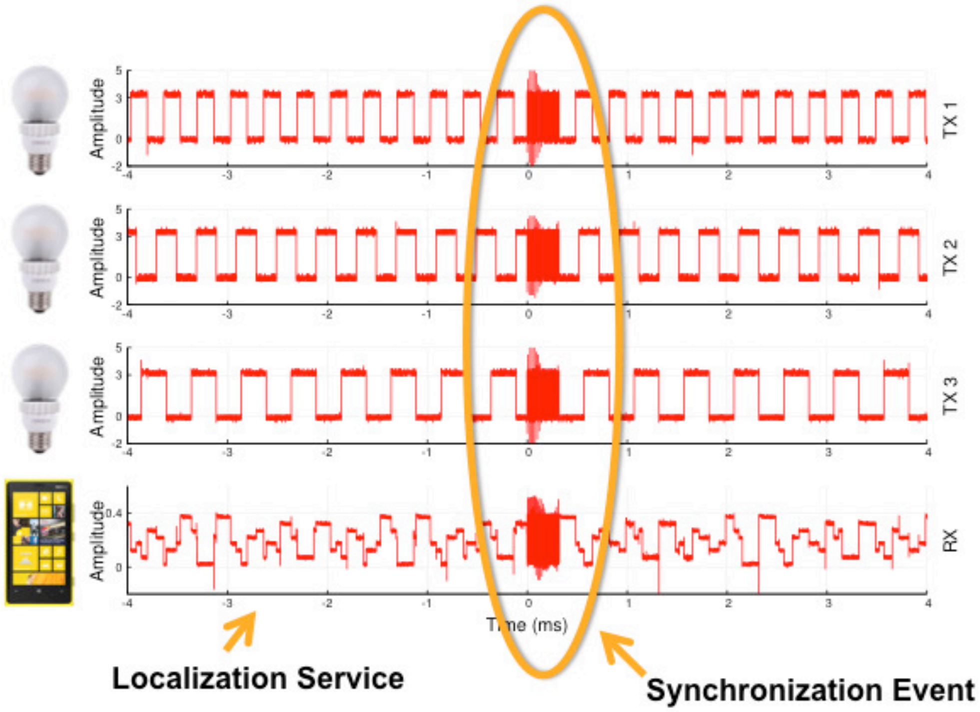


- Decimeter-level accuracy
- Unmodified smartphones
- Trivially modified LED lights

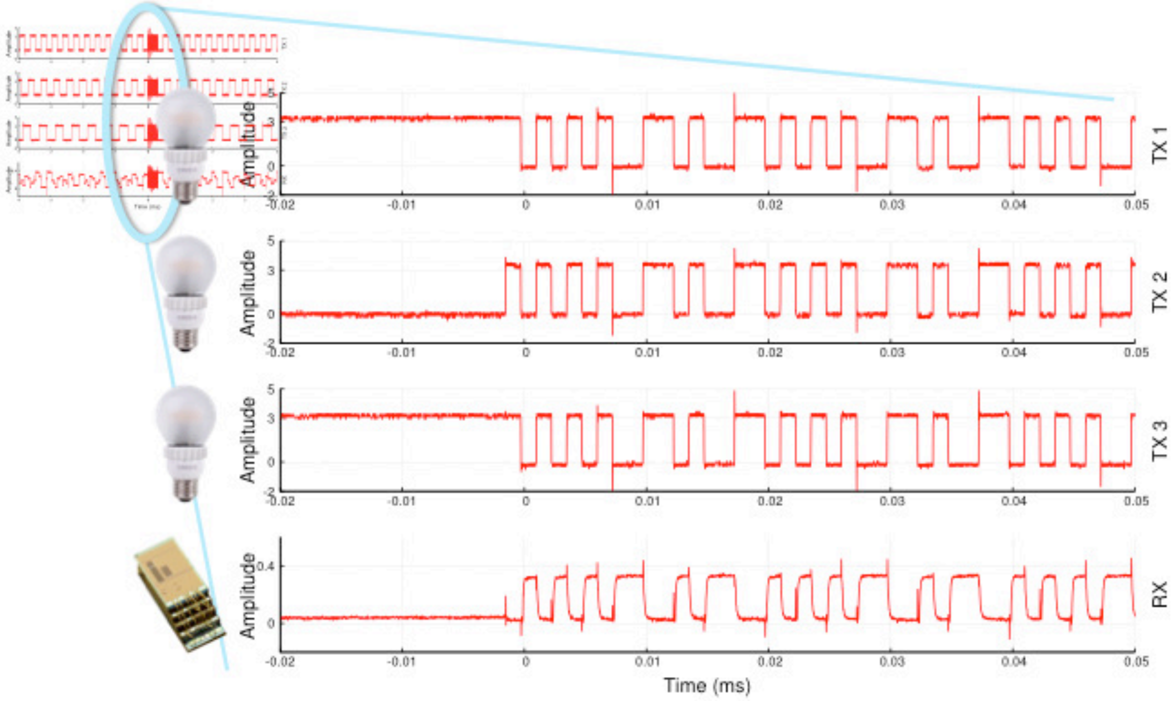
Luxapose: Indoor Positioning with Mobile Phones and Visible Light (MobiCom'14)

Ye-Sheng Kuo, Pat Pannuto, Ko-Jen Hsiao, and Prabal Dutta

Can smart dust and smart phones co-exist on the same lighting infrastructure?



With sufficient synchronization, transmitted signals can constructively interfere



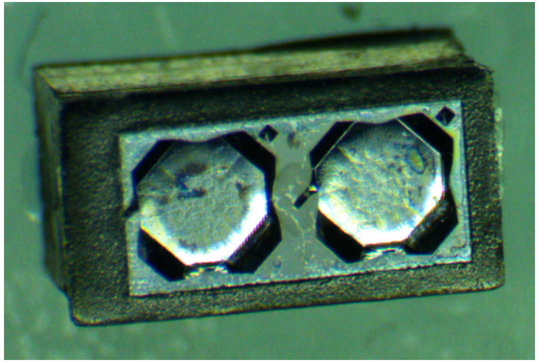
Emerging non-traditional (non-narrowband RF) communication technologies are just beginning exploration

- Sub-nanowatt VLC receivers
- VLC retroreflectors for bi-directional communication
- Acoustic?
- Vibratory?
- Backscatter communication
- High-fidelity localization with UWB
  - For almost no power with backscattered UWB?

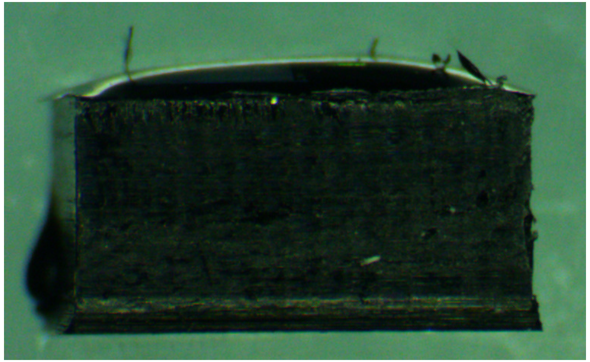
- A Brief History of “Smart Dust”
- System Composition Architecture
- Energy
- Communication
- Debugging

An open question: How to effectively debug a system you cannot touch?

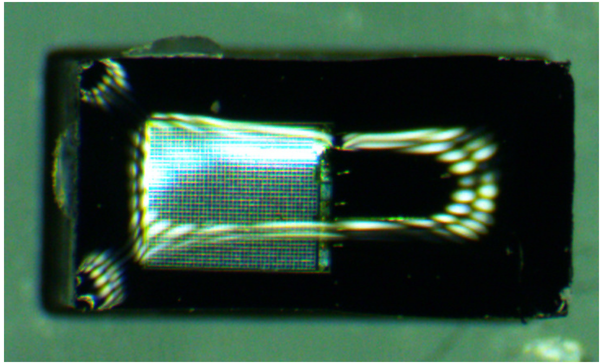
- Debug chips provide some introspection
  - Snoop (or inject) on system bus
  - Power traces remarkably effective
- Encapsulated systems remain a big issue



Bottom View: MEMS pressure sensor



Side View: Light-blocking epoxy



Top View: Clear epoxy for harvesting

Many further questions in this space for future motes

- When is an energy harvesting node dead or just harvesting?
- How to debug 1,000 nodes?
  - How to find "the" bad node among 1,000?
  - How to collect and recover failed nodes?
    - Or is it really just dust? ☺

Everyone's favorite debugging answer: Write something well enough that it won't crash!

- A teaser for Tock
  - New embedded operating system
    - Collaboration with Stanford and UC Berkeley
  - Defense-in-depth: language (Rust) and hardware (MPU) protections
  - It's time for proper kernel/application separation
    - Even viable for micro-scale processor (currently M0, but M0+ has MPU)
- And a safe, accessible user-space runtime (Lua)

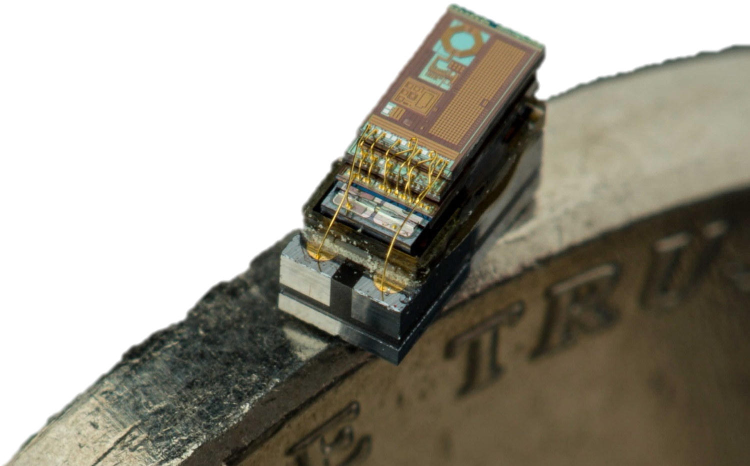
What have we learned from the development of millimeter-scale systems?

- Existing system synthesis technologies cannot scale
- Revisiting long-standing architectural components can yield unforeseen benefits
- To maximize energy efficiency, power-awareness must permeate every aspect of the system design
- New communication technologies are setting up major architectural shifts
- There is lots of work yet to do!

http://mbus.io
http://github.com/mbus

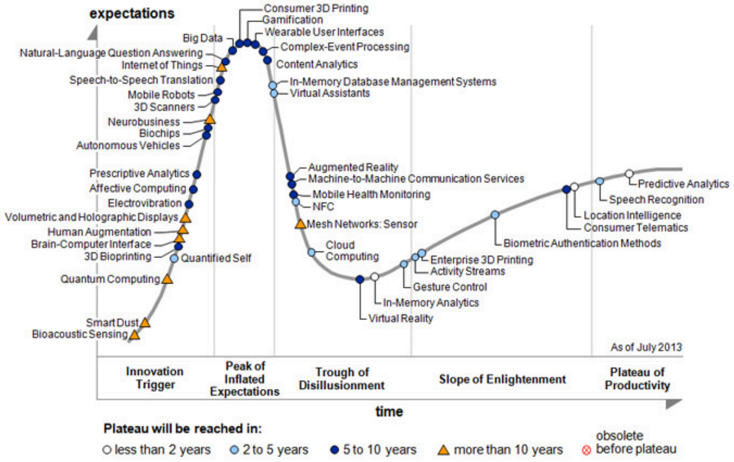
http://lab11.eecs.umich.edu
http://github.com/lab11

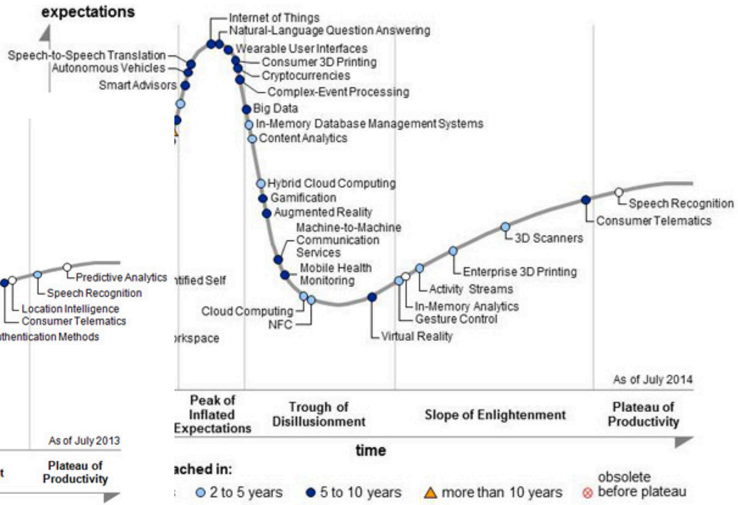
Pat Pannuto, University of Michigan
ppannuto@umich.edu, patpannuto.com

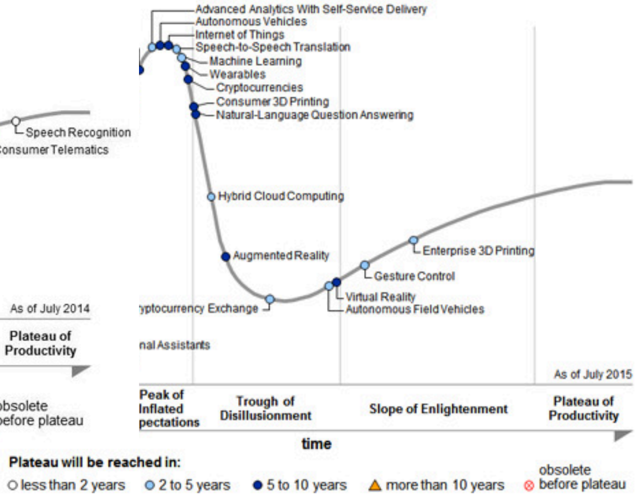


Gartner Hype Cycle

- Appeared 2013, fell off 2014, back in 2015







[2013]: http://www.gartner.com/newsroom/id/2575515
[2014]: http://www.gartner.com/newsroom/id/2819918
[2015]: http://www.gartner.com/newsroom/id/3114217