Northwestern University

Wireless Sensor Networks and RFIDs

Instructor: Robert Dick

Slides from Randall Berry

Motivations

Two basic motivation for any application:

1. Can do something you already do cheaper/better.
   - Wiring costs
     - indust. apps $300-400/sensor vs. < $100.
   - More frequent/cheaper than human monitoring.
2. Can do something you could not do before.
   - Non-invasive habitat monitoring.
   - Chemical/biohazard detection.

Wide Ranging Requirements

- **“Interaction pattern”**
  - Event detection, periodic measurements, function approximation, tracking
- **“Lifetime” requirements**
  - Hours to years.
- **Deployment options**
  - Planned vs. ad hoc.
  - Mobile vs. stationary.
- **Maintenance options/Fault tolerance**
  - Programmability/replace-ability/redundancy
- **Options for Energy supply**
  - Wired,battery, renewable,passive.

Wide Ranging Requirements

- **All requirements will not likely be met by any single type of sensor network.**
  - **Application dependent sensor nets.**

- **Role of standards?**

Related technologies

- **RFID (radio frequency identification)**
  - Use active or passive tags for monitoring locations.
  - Can be viewed as a relatively simple type of sensor network.
  - Usually no multi-hop communication/limited sensing capability.
- **Wireless mobile ad hoc networks (MANETS)**
  - E.g. 802.11 in ad hoc mode.
  - Support multi-hop communication.
  - Designed to provide Internet-like connectivity to mobile users (not sensing).

Outline:

- Introduction
- Enabling technology trends.
- History
- Single Sensor Node Architecture.
  - Hardware
  - Software
- Design considerations

Enabling trends

- Moore’s law.
- Convergence of communications and computation.
- MEMS technology.

Moore's Law

- Transistor density doubles ≈ every 18-24 months.
  - stated in 1965 by Gordon Moore.
- Results:
  - Cheaper cost/transistor
    - (today ≈ same as one printed newspaper character).
  - Given chip size ⇒ more functionality.
  - Given functionality ⇒ smaller chip size.
  - Empirical observation – not scientific fact
    - "self-fulfilling" prophesy
    - "Expected" to continue for at least next 5-10 years

Moore's Law

- Key insight is computation is "cheap" and can be expected to get cheaper.
- For sensor nodes – energy needed per computation is also important.
- Moore's law also drives this down:
  - Power consumption for a given operation drops roughly as the fourth power of the feature size.
  - but power consumption increases (linearly) with clock speed.

Moore’s Law

- Moore’s law type of improvements occur in other areas too
  - E.g. hard disk capacity.
- But can not be expected in all relevant technologies:
  - performance of many sensors and energy technology are limited by laws of physics.
  - Communication rates are limited by Shannon.

"Convergence" of Communication and Computing

- Convergence occurs when two previously independent areas become competitive.
  - Firms enter each others markets.
  - Devices merge – e.g. smart phones.
- Recent history:
  - Stand alone computers (PC's) and communications have converged $\Rightarrow$ the Internet.
  - Today networked computers are as much communication devices as computational.
- Key driver – digital revolution:
  - Both communications and computation deal with bits.

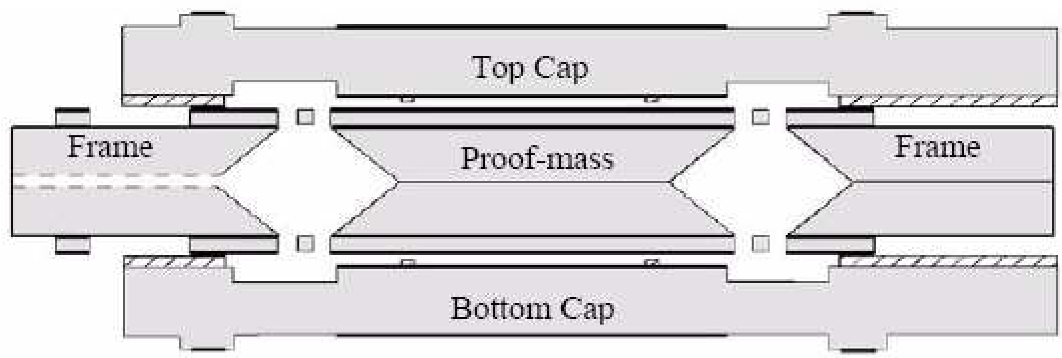
"Convergence" of Communication and Computing

- Next wave?
  - Rapid growth in wireless networking.
  - Rapid growth in embedded processors.
    - 98% of all processors sold are embedded.
  - Sensor networks represent convergence of these two areas.

MEMS technology

- MEMS = Micro-Electro-Mechanical Systems
- Integration of mechanical elements, sensors, actuators, and electronics on one chip through microfabrication technology.
  - Allows sensors to also "ride Moore's law"
- Currently used to develop some small sensing devices (e.g. accelerometers).
- In the future seen as promising way to build even smaller sensors.
  - "system on a chip" design.

MEMS technology



Example MEMS accelerometer:

Proof-mass is a moving structure suspended by silicon springs.

As it moves it changes the differential capacitance between the upper and lower fixed plates.

Entire structure can be made as part of an ASIC that is smaller than a postage stamp.

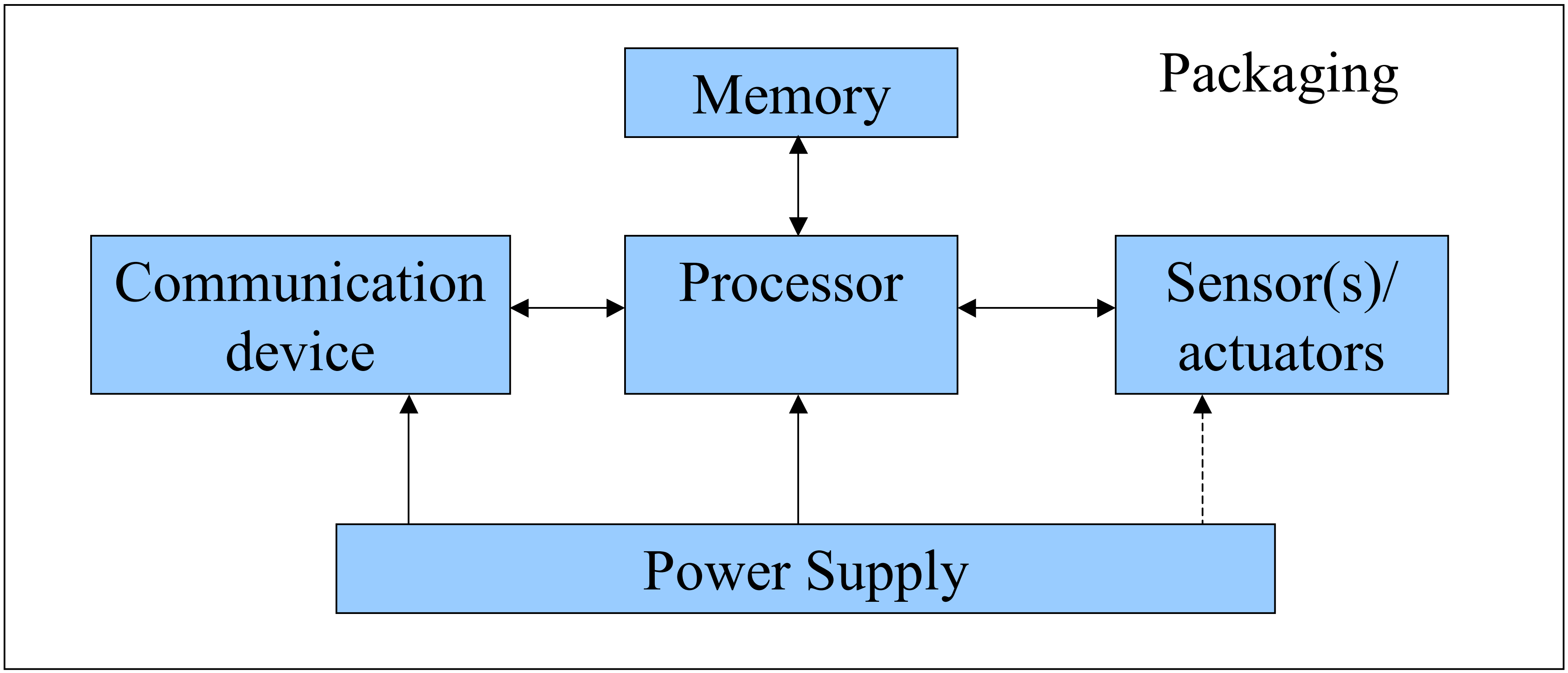
Brief Sensor Net History

- Long history of Military research dating back to 1980's.
- Recent excitement traced back to DARPA "Smart Dust" project at UC Berkeley (1998-2001).
  - Lead investigator: Kris Pister.
  - Goal to build sensor nodes using COTS technology.
  - Resulted in Berkeley Motes, TinyOS
    - Contracted with Crossbow to build Motes, with backing from Intel.
    - Hardware design made public.
- 2001 Pister founded Dust Networks.
  - In 2004 received $7 million from In-Q-tel (CIA venture arm).
  - Around same time several other start-ups founded (Ember, Millennial Net)
  - Together in 2004 about $53 million in venture funding.

Outline:

- Introduction
  - Applications
- Enabling technology trends.
- History
- Single Sensor Node Architecture.
  - Hardware
  - Software
- Design considerations

Basic Sensor Node Architecture



- All may not be present in all designs.
- Several options for each module.
  - Design driven by application constraints.

Processor (controller)

- Brains of the node - used to process data/execute code.
  - Communication protocols, signal processing, etc.
- Most common option is to use a microcontroller
  - Special processors geared towards embedded applications.
  - Less processing power than general-purpose processors but much lower energy consumption.
  - Usually some on chip RAM
  - high flexibility in connecting other devises
- Other options (less flexibility/better energy efficiency)
  - Field-programmable gate arrays (FPGAs)
  - Application specific integrated circuits (ASICs)
    - More expensive unless large volumes.

Memory

- Usually need some memory for storing sensor data & packets from other nodes.
- Options:
  - RAM (random access memory) fast but volatile – rarely used except for on-chip.
  - Flash memory - non-volatile but slower to read/write and higher required energy required.

Communication device

- Typically RF between 433 Mhz and 2.4 Ghz
  - Most operate in public ISM bands
  - Optical/Infrared & ultrasound also considered for some applications.
- Choice of radio system depends in part on deployment
  - E.g. large sparse deployment may need longer-rang comm.
  - Most apps. require comparably low data rates O(100Kbps)
- Also must balance complexity/power consumption
  - *m-arry* modulation schemes require more processing than binary.
  - Decoding complex channel codes requires significant processing.

Communication device

- Number of off-the-shelf single chip radio transceivers are available
  - Some based on 802.15.4 (Zigbee) Phy-layer
  - Others based on non-standardized platforms.
  - Sometimes include on-chip support for MAC/link layer protocols.

Sensors

- Wide range of different sensing options are available including:
  - Pressure sensors
  - thermometers
  - Humidity sensors
  - Accelerometers
  - Gyroscopes (angular rotation)
  - Gas phase composition sensors (e.g. automotive EGO sesnors)
  - Microphones
  - Strain/force sensors
  - Magnetometers
  - Light intensity sensors
  - Digital cameras

Sensors

- Generally focus is on passive sensing
  - Do not actively probe environment (e.g. radar)
- Also most technologies are omnidirectional within some "area of coverage"

- Most technologies measure analog quantities – must be converted to digital (A/D conversion).

- All sensors are imperfect:
  - Noise in environment/electronics.
  - Quantization noise in A/D conversion.
  - "Cross talk" from other physical quantities.
  - Drift (calibration)

Sensors

- Standard figures of merit:

$$\text{Responsivity} = \frac{\text{output signal (volts)}}{\text{input physical signal amplitude}}$$

$$\text{Sensitivity} = \frac{\text{output signal in absence of input}}{\text{responsivity}}$$

Sensitivity indicates how well sensor is at separating known signal from noise.

These are usually frequency dependent.

Sensors

- Needed performance highly depends on applications.
  - E.g. airbag deployment vs. inertial navigation.
- Performance per sensor can often traded off with number of sensors.

Power supplies

- Batteries are primary means of energy storage.
  - Non-rechargeable or chargeable.
- Energy scavenging techniques are also receiving a lot of interest:
  - Solar cells
  - Also techniques to harvest energy from vibrations, temp. gradients, etc.
- In some cases, external power can be used.

Power supplies

- Key parameters:
  - (volume) Energy density (Joules/cm^3)
  - Mass energy density (Joules/kg)
  - Maximum discharge current
  - Self-discharge
  - Relaxation
    - Self recharging when no current drawn
  - For energy scavenging, power density (W/cm^2) is also important.

Power supplies

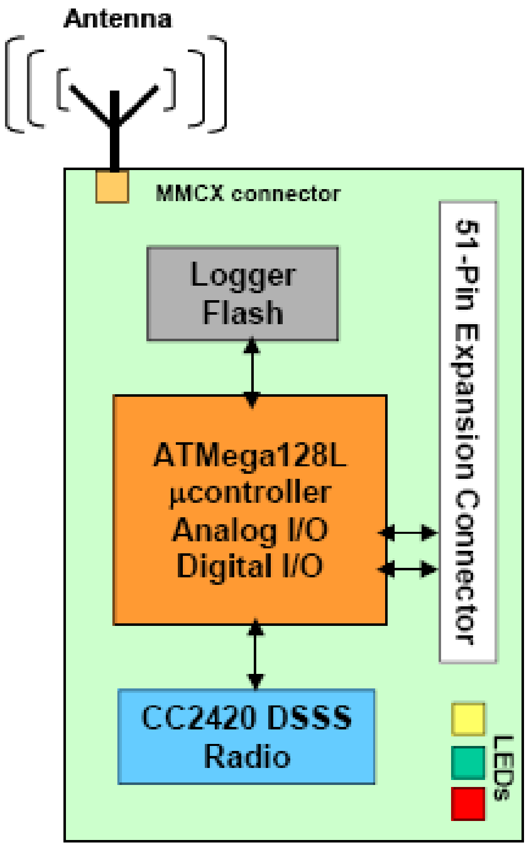
Example:

Li-ion batteries have an energy density of ≈ 2880 Joules/cm^3.

Given a 1cm^3 battery if a sensor network requires a peak power of 0.1 W, how long can it operate at peak power?

Example Mote Design

- Crossbow MICAz Mote.
- 2.25”x1.25”x0.25”
- “sensor platform” – actual sensors added, depending on application.



Example (Crossbow MICAz)

- **Processor:** 7.37 MHz/8 bit
  - Program memory 128kB
  - SRAM 4kB
- **I/O:** 10 bit ADC
- **RF:** 2400MHz, Max data rate 250 kbit/sec
- **Flash memory:** 512 kB
- **Power source:** 2 AA batteries.

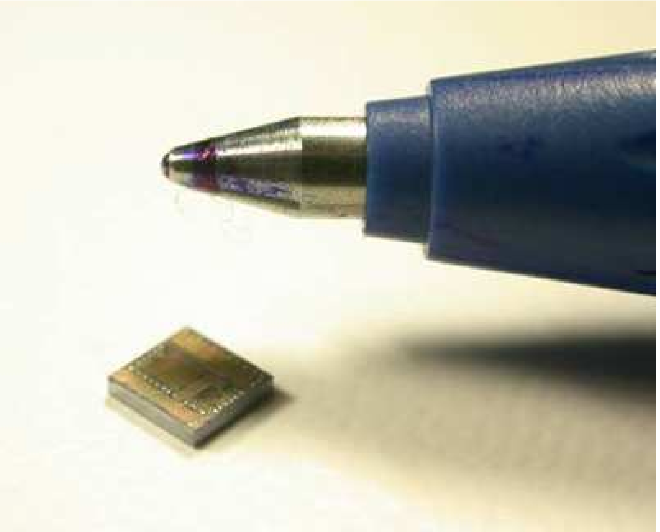
Model MPR 2040 – Apr. 2005

Example Sensor Module

- Crossbow MICA2 Multi-Sensor Module
  - Light, Temperature, Microphone, Sounder,Tone Detection Circuit, 2-Axis Accelerometer, 2-Axis Magnetometer
  - Complete sensor with housing

Smaller Example

- "Spec" Mote
  - Developed in 2003 @ Berkeley.
  - Contains Processor, 3K memory, 8 bit ADC & FSK transmitter on single chip.
  - Requires external antenna, power supply, & sensors.
  - Communicated 40ft indoors at 19.2Kbps.



Software

- Operating system
  - e.g. TinyOS, Linux
- Database and/or application specific software
  - e.g. TinyDB
- Networking Firmware
  - e.g. ZigBee specified protocols
    (more on this next time.)

(other proprietary software also used).

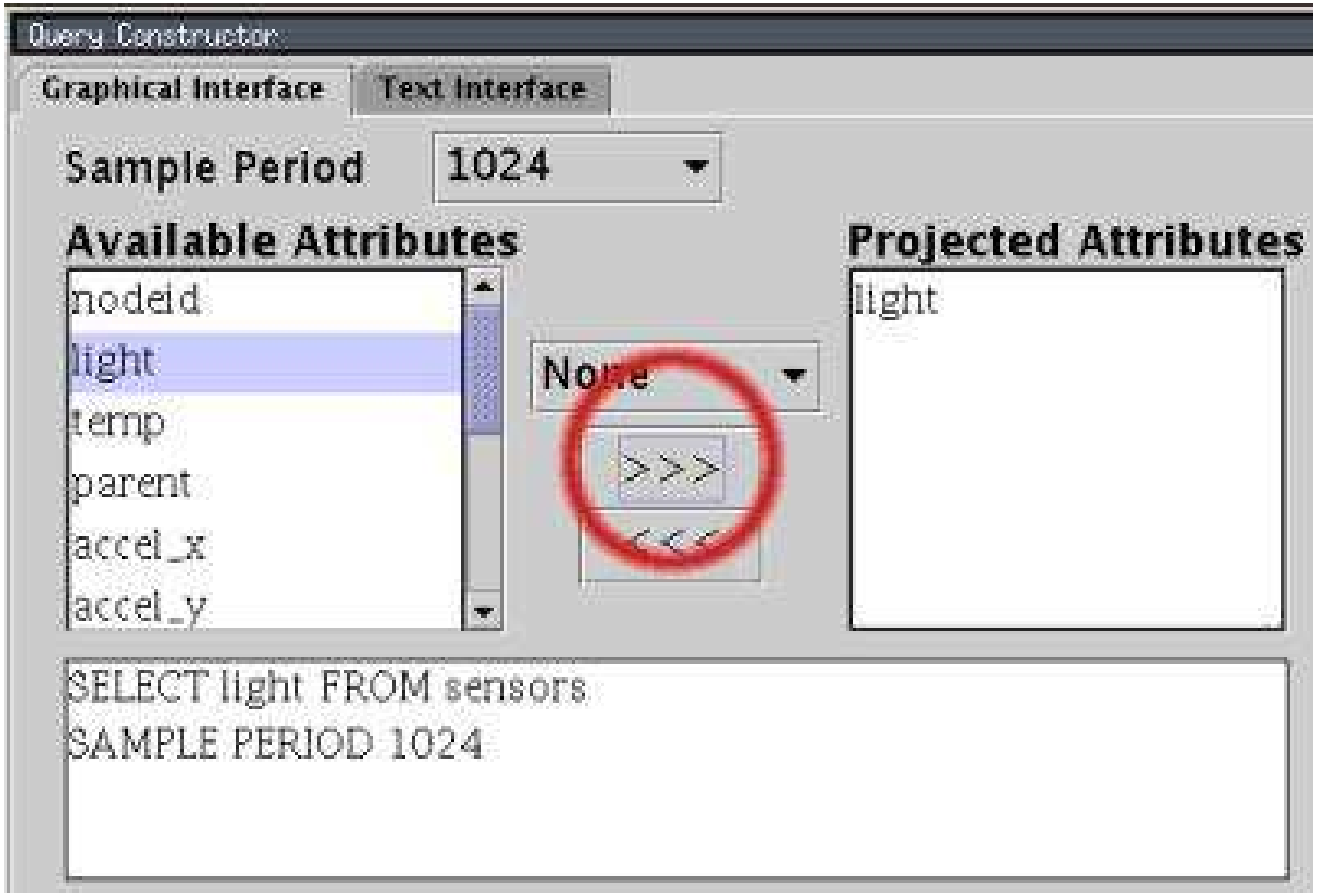
TinyOS

- Most standard OS's (e.g. MS windows, Unix) are too large and processor intensive.
- TinyOS is open source OS for sensor networks
  - Originally developed at UC-Berkeley.
  - Highly modular – if only certain features needed, rest can be removed to free up memory for data.
  - Core requires 400 bytes memory (typically uses 4KB).
  - Energy aware - forces programs to shut down except when certain events occur that warrant action.
  - Event-based

TinyDB

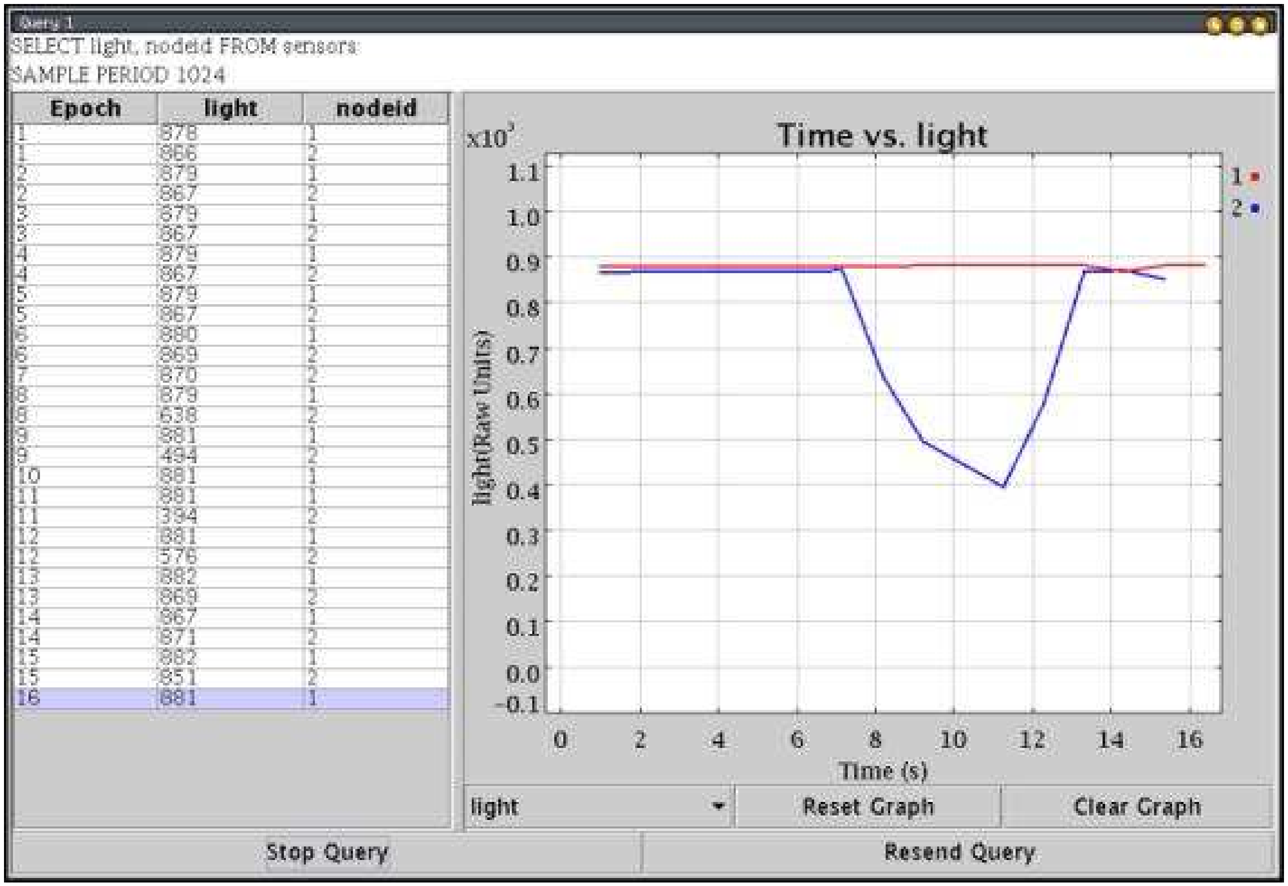
- Developed by Berkeley/Intel.
  - Runs on top of TinyOS
- Allows sensor net to function as a data-base.
  - Provides GUI/SQL-like queries/responses to application designer.
  - Hides complexity of network from end-user
- Queries entered at base-station (e.g. wired PC).
  - Some optimization performed before disseminating queries to save power.

TinyDB



- Specifies a new light reading delivered every 1024 *msec*.

TinyDB



Epoch	light	nodeid
1	878	1
1	866	2
2	879	1
2	867	2
3	879	1
3	867	2
4	879	1
4	867	2
5	879	1
5	867	2
6	880	1
6	869	2
7	870	2
8	879	1
8	638	2
9	881	1
9	494	2
10	881	1
11	881	1
11	394	2
12	881	1
12	576	2
13	882	1
13	869	2
14	867	1
14	871	2
15	882	1
15	851	2
16	881	1

- Result of previous query.