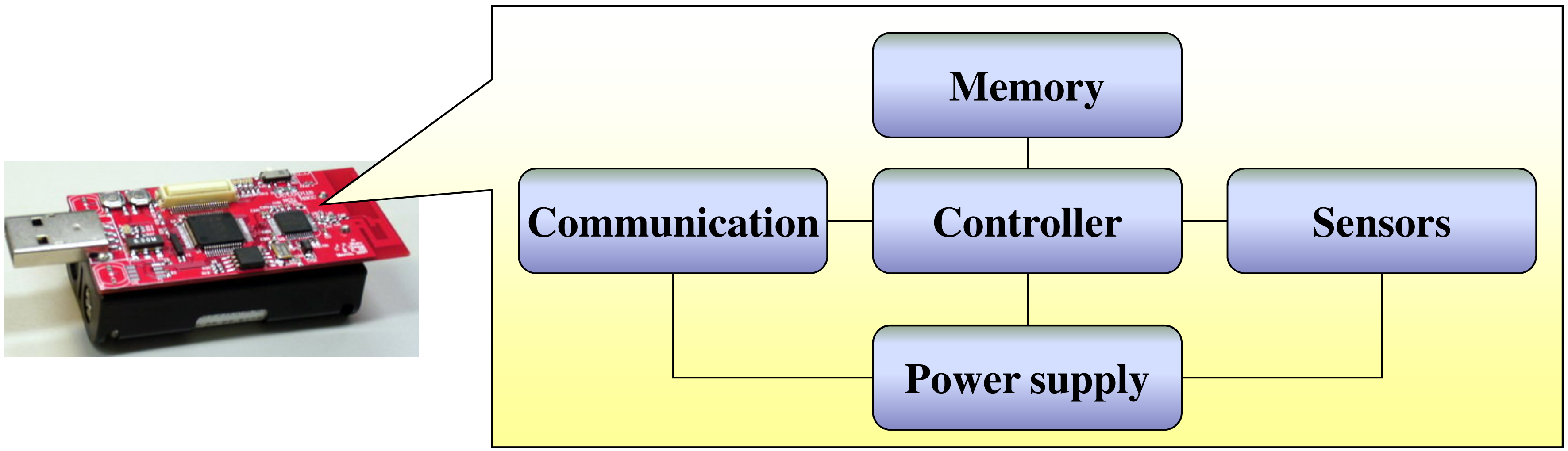
Chapter 2
Single-node Architecture

Outline

2.1. Sensor Node Architecture

Main Architecture of Sensor Node

- The main architecture of sensor node includes following components:
  - Controller module
  - Memory module
  - Communication module
  - Sensing modules
  - Power supply module



Main Components of a Sensor Node ： Controller module

- Main options:
  - MCUs (Microcontrollers)
    - The processor for general purposes
    - Optimized for embedded applications
    - Low energy consumption
  - DSPs (Digital Signal Processors)
    - Optimized for signal processing
    - Low cost
    - High processing speed
    - Not suitable for sensor node
  - FPGAs (Field Programmable Gate Arrays)
    - Suitable for product development and testing
    - Cost higher than DSPs
    - High energy consumption
    - Processing speed lower than ASICs
  - ASICs (Application-Specific Integrated Circuits)
    - Only when peak performance is needed
    - For special purpose
    - Not flexable

Main Components of a Sensor Node ： Controller module

- Example of microcontrollers are recently used in Senor Node
  - ATMega128L, Atmel
    - 8-bit controller
    - 128KB program memory (flash)
    - 512KB additional data flash memory
    - larger memory than MSP430
    - slower
  - MSP430, TI (Texas Instruments)
    - 16-bit RISC core
    - 8MHz
    - 48KB Flash
    - 10KB RAM
    - several DACs
    - RT clock
  - 8051 in CC2430 & CC2431, TI (Texas Instruments)
    - 8-bit MCU
    - 32/64/128 KB program memory
    - 8 KB RAM
    - 21I/O

Main Components of a Sensor Node ：Communication module

- The communication module of a sensor node is called “Radio Transceiver”
- The essentially tasks of transceiver is to “transmit” and “receive” data between a pair of nodes
- Which characteristics of the transceiver should be consider for sensor nodes?
  - Capabilities
  - Energy characteristics
  - Radio performance

Main Components of a Sensor Node ： Communication module

- Transceiver characteristics
  - Capabilities
    - Interface to upper layers (most notably to the MAC layer)
      - bit, byte or packet?
    - Supported frequency range?
      - Typically, somewhere in 433 MHz – 2.4 GHz, ISM band
    - Supported multiple channels?
    - Transmission data rates?
    - Communication range?

Main Components of a Sensor Node ： Communication module

- Transceiver characteristics
  - Energy characteristics
    - Power consumption to send/receive data?
    - Time and energy consumption to change between different states?
    - Supported transmission power control?
    - Power efficiency (which percentage of consumed power is radiated?)

Main Components of a Sensor Node ： Communication module

- Radio performance
  - Modulation?
    - ASK, FSK, PSK, QPSK…
  - Noise figure?
  - Gain?
  - Carrier sensing and RSSI characteristics
  - Frequency stability (Ex: towards temperature changes)
  - Voltage range

Main Components of a Sensor Node ： Communication module

- Transceivers typically has several different **states/modes** :
  - **Transmit** mode
    - Transmitting data
  - **Receive** mode
    - Receiving data
  - **Idle** mode
    - Ready to receive, but not doing so
    - Some functions in hardware can be switched off
    - Reducing energy consumption a little
  - **Sleep** mode
    - Significant parts of the transceiver are switched off
    - Not able to immediately receive something
    - Recovery time and startup energy to leave sleep state can be significant

Main Components of a Sensor Node ： Communication module

- Example of transceivers are recently used in Senor Node
  - RFM TR1000 family
    - 916 or 868 MHz
    - 400 kHz bandwidth
    - Up to 115,2 kbps
    - On/off keying or ASK
    - Dynamically tuneable output power
    - Maximum power about 1.4 mW
    - Low power consumption
  - Chipcon CC1000
    - Range 300 to 1000 MHz, programmable in 250 Hz steps
    - FSK modulation
    - Provides RSSI
  - Chipcon CC 2400
    - Ex: TI CC2420
    - Implements 802.15.4
    - 2.4 GHz, DSSS modem
    - 250 kbps
    - Higher power consumption than above transceivers
  - Infineon TDA 525x family
    - E.g., 5250: 868 MHz
    - ASK or FSK modulation
    - RSSI, highly efficient power amplifier
    - Intelligent power down, "self-polling" mechanism
    - Excellent blocking performance

Main Components of a Sensor Node ： Communication module

- TI CC 2431
  - 8051 MCU core
  - 128KB in-system programmable flash
  - 8KB RAM, 4KB with data retention in all power mode
  - Powerful DMA
  - One IEEE 802.15.4 MAC timer
  - 2.4GHz IEEE 802.15.4 compliant RF
  - RX (27mA), TX (27mA), MCU running at 32MHz
  - 0.5uA current consumption in powerdown mode
  - 0.3uA current consumption in stand-by mode
  - Wide supply voltage range (2.0V-3.6V)
  - CSMA/CA hardware support
  - Digital RSSI/LQI support
  - 12-bit ADC with up to eight inputs and configuration resolution
  - Two USARTs with support for several serial protocols

Main Components of a Sensor Node ： Sensing module

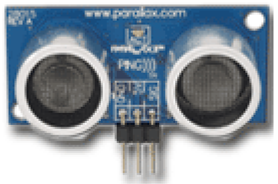
- Sensor’s main categories [1]
  - Passive vs. Active
  - Directional vs. Omidirectional

- Some sensor examples
  - Passive, omnidirectional
    - light, thermometer, microphones, hygrometer, …
  - Passive, directional
    - electronic compass, gyroscope , …
  - Passive, narrow-beam
    - CCD Camera, triple axis accelerometer, infrared sensor …
  - Active sensors
    - Radar, Ultrasonic, …

Main Components of a Sensor Node ： Sensing module

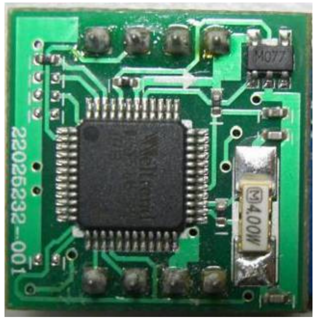
- Example of sensors are integrated with Senor Node

Infar sensor



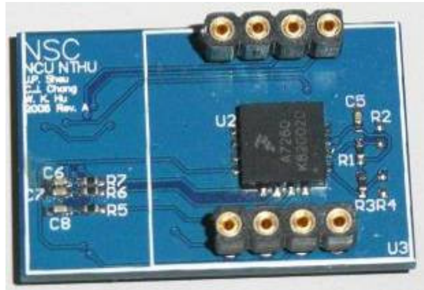
Ultrasonic

Gyroscope

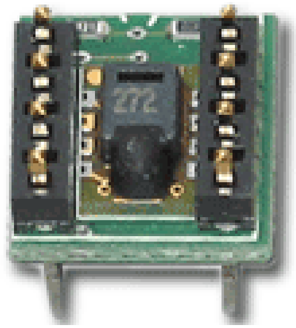


Electronic compass

Pressure Sensor



Triple axis accelerometer



Temperature and Humidity Sensor

Main Components of a Sensor Node ：Power supply module

- Power supply module
  - Provides as much energy as possible and includes following requirements
    - Longevity (long shelf live)
    - Low self-discharge
    - Voltage stability
    - Smallest cost
    - High capacity/volume
    - Efficient recharging at low current
    - Shorter recharge time
- Options of power supply module
  - Primary batteries
    - not rechargeable
  - Secondary batteries
    - rechargeable
    - In WSN, recharging may or may not be an option

Main Components of a Sensor Node ： Memory module

- The memory module of a sensor node has two major tasks
  - To store intermediate sensor readings, packets from other nodes, and so on.
  - To store program code
- For the first task
  - Random Access Memory (RAM) is suitable
  - The advantage of RAM is fast
  - The main disadvantage is that it loses its content if power supply is interrupted

Main Components of a Sensor Node ： Memory module

- For the second task
  - Read-Only Memory (ROM)
  - Electrically Erasable Programmable Read-Only Memory (EEPROM)
  - Flash memory (allowing data to be erased or written in blocks)
    - can also serve as intermediate storage of data in case RAM is insufficient or when the power supply of RAM should be shut down for some time
    - long read and write access delays
    - high required energy

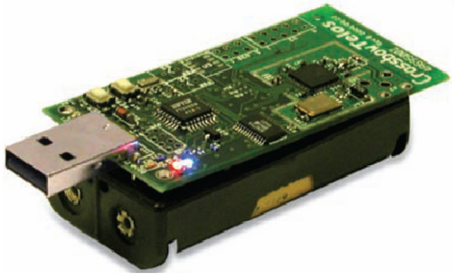
2.2. Introduction of Sensor Hardware Platform

Overview of Sensor Node Platforms

- Some modules developed by U.C. Berkeley & Crossbow Tech.
  - MICA2
    - 8-bit Atmel ATmega128L microcontroller
    - (4 KB SRAM + 128 KB Flash)
    - RF: CC1000 (data rate: 38.4kbits/s)
  - MICAz
    - 8-bit Atmel ATmega128L microcontroller
    - RF: CC2420 (data rate: 250kbits/s)
  - TelosB
    - 16-bit MSP430 microcontroller
    - (10 KB RAM + 48KB Flash) + 1MB Flash
    - RF: CC2420 (data rate: 250kbits/s)
  - IRIS
    - 8-bit Atmel ATmega1281 microcontroller
    - (8 KB RAM + 128KB Flash) + 512KB Flash
    - RF: RF230, data rate: 250kbits/s

MICA2

MICAz



TelosB

IRIS

Overview of Sensor Node Platforms

- Octopus modules were developed by NTHU
  - Octopus I (Compatible with MICAz)
    - 8-bit Atmel ATmega128L microcontroller
    - RF: CC2420 (data rate: 250kbits/s)
  - Octopus II
    - 16-bit MSP430 microcontroller
    - 10 KB RAM + 48KB Flash) + 1MB Flash
    - RF: CC2420 (data rate: 250kbits/s)
  - Octopus X
    - 8-bit 8051 microcontroller
    - 128KB in-system programmable flash
    - 8KB RAM + 4KB EEPROM
    - RF: CC2430, EEE 802.15.4 compliant RF transceiver

Octopus I

Octopus II

Octopus X

Introduction of Octopus X Hardware Platform

- Octopus X includes three models
  - Octopus X-A
    - CC2431 + Inverted F Antenna
  - Octopus X-B
    - CC2431 + SMA Type Antenna
  - Octopus X-C
    - CC2431 + Inverted F and SMA Type Antenna + USB interface

Octopus X-A

Octopus X-B

Octopus X-C

- Peripherals of Octopus X
  - Octopus X-USB dongle
  - Octopus X-Sensor board
    - Temperature sensor
    - Gyroscope
    - Three axis accelerometer
    - Electronic Compass

USB dongle

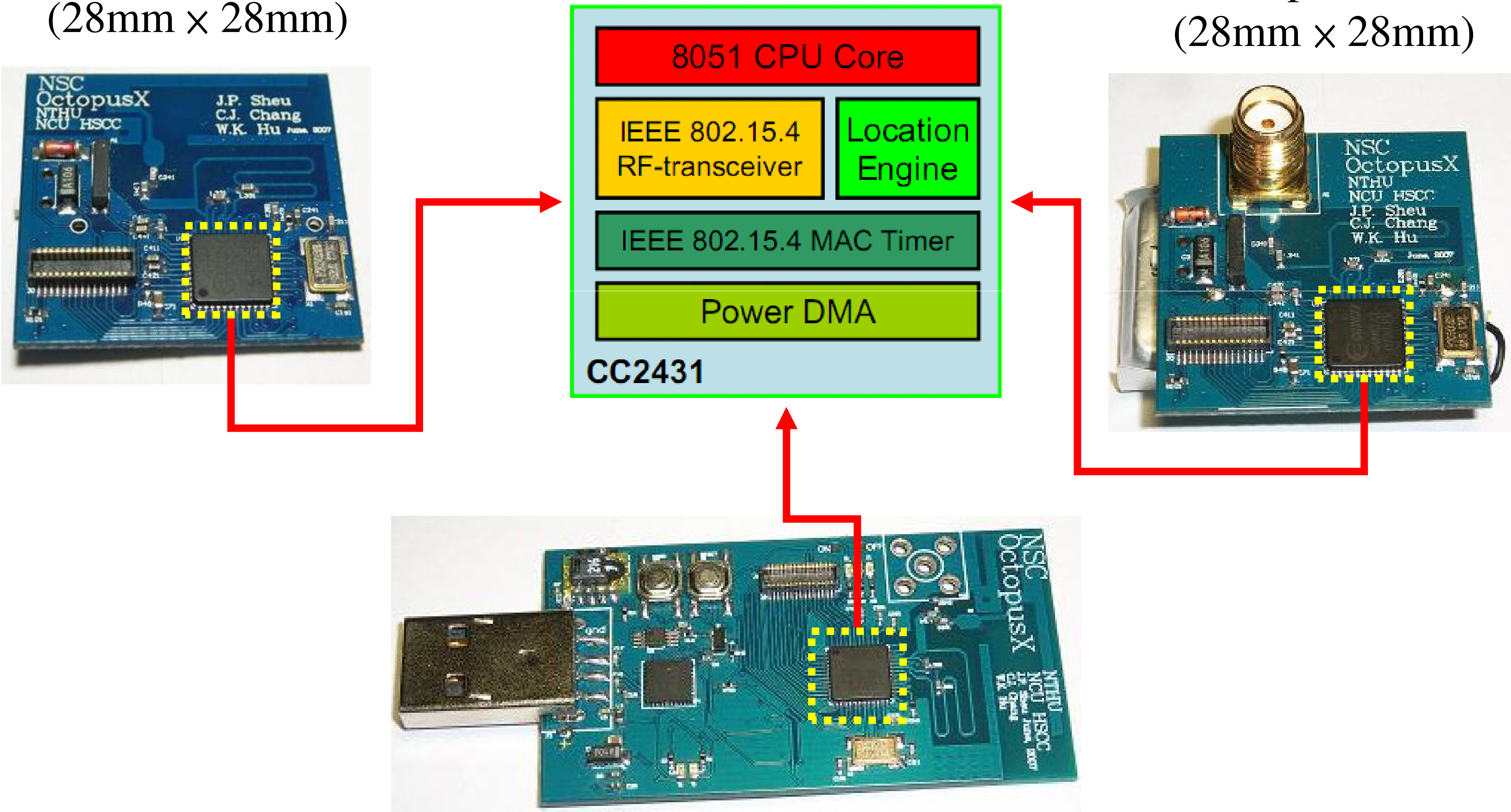
Temperature sensor

Three axis accelerometer

Introduction of Octopus X Hardware Platform

Octopus X-A
(28mm x 28mm)

Octopus X-B
(28mm x 28mm)

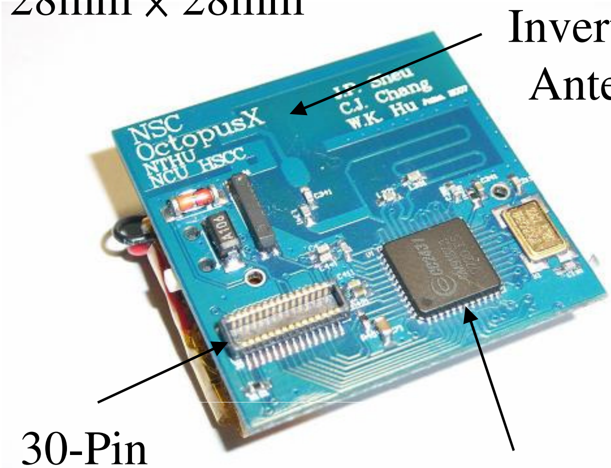


Octopus X-C
(57mm x 31mm)

Features of Octopus X-A

Size: 28mm × 28mm

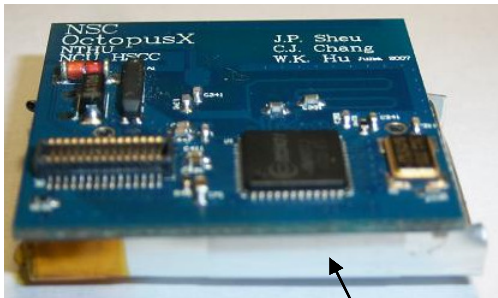
Inverted-F Antenna



30-Pin expansion connector

CC2431(MCU+RF)

- MCU (CC2431)
- Inverted-F antenna
- RF transmission range ≒ 100m
- External crystal (32MHz+32.768KHz)
- 30-Pin expansion connector
- Polymer batter (3.7V 300mAh)



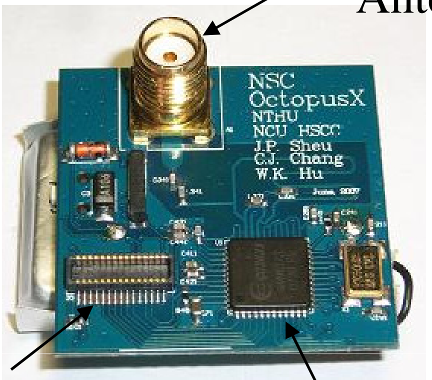
Height: 7mm

Polymer battery

Features of Octopus X-B

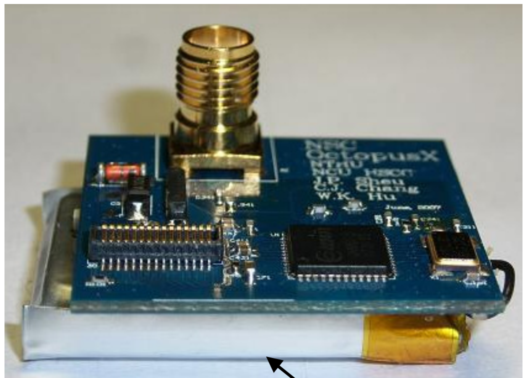
Size: 28mm × 28mm

SMA Type Antenna



30-Pin expansion connector

CC2431(MCU+RF)



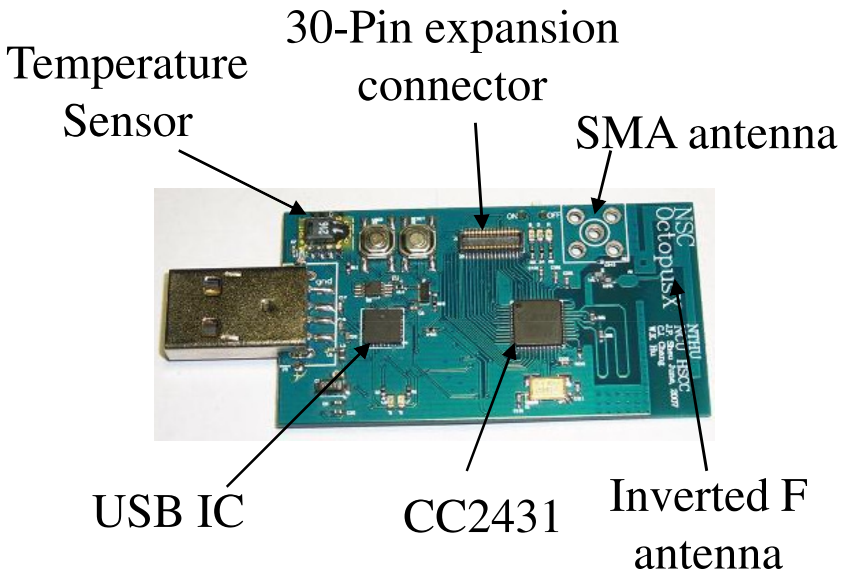
Height: 7mm

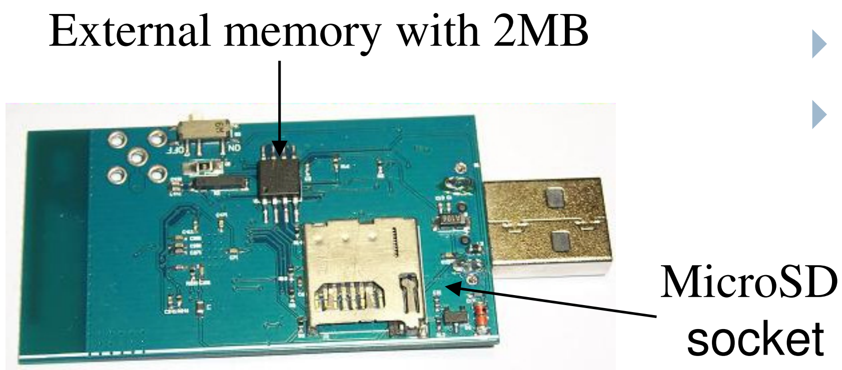
Polymer battery

- MCU (CC2431)
- SMA type antenna
- RF transmission range ≒ 150m
- External crystal (32MHz+32.768KHz)
- 30-Pin expansion connector
- Polymer batter (3.7V 300mAh)

Features of Octopus X-C

Size: 57mm × 31mm

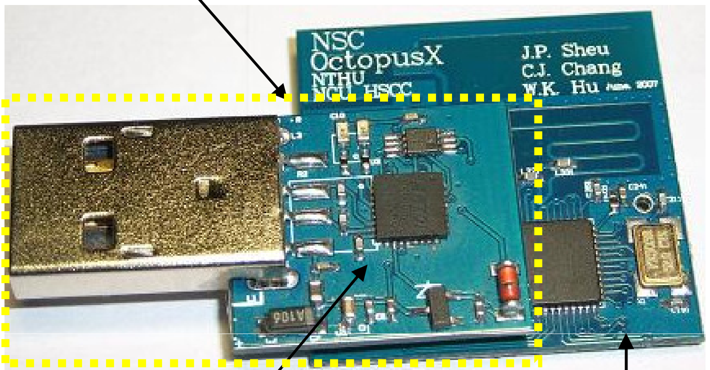




- MCU (CC2431)
- SMA type and Inverted-F antenna
- Humidity & Temperature sensor
  - Humidity 0~100%RH (0.03%RH)
  - Temperature -40°C~120°C (0.01°C)
- External flash memory (2MB)
- MicroSD socket (up to 8GB)
- USB Interface
  - Programming
  - Debugging
  - Data collection

Features of Octopus X - USB Dongle

USB Dongle



USB IC

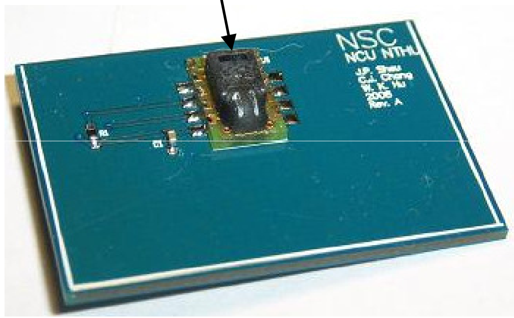
Octopus X-A

- Octopus X-USB dongle provides an easy-to-use USB protocol for
  - Programming
  - Debugging
  - Data collections

Features of Octopus X - Sensor Boards

Size: 28mm × 18mm

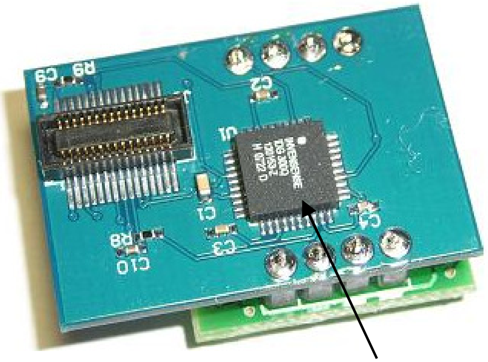
Temperature sensor



Front view of Octopus X-sensor board

Electronic Compass

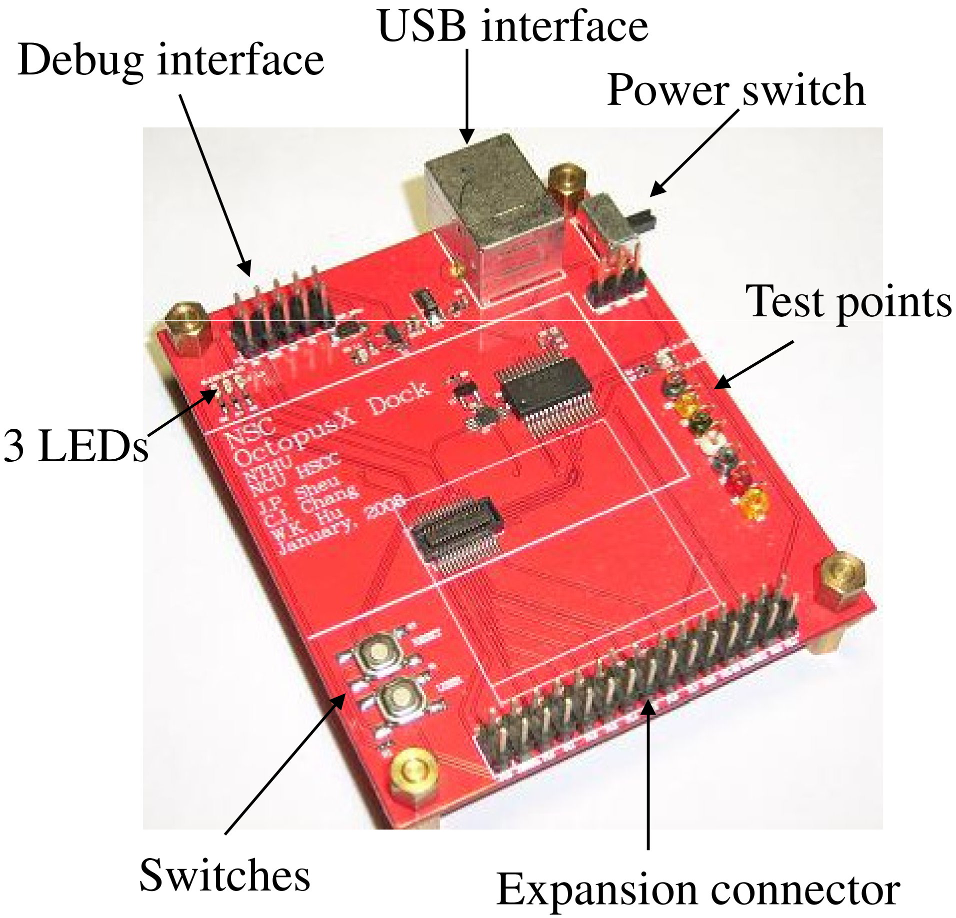
Back view of Octopus X-sensor board



Sensor board
(Gyroscope + Triple axis accelerometer)

Features of Octopus X - Dock

Size: 60mm × 71mm



- USB interface
  - Programming with our flash programmer
  - Data collections
- Debug interface
  - Programming with TI SmartRF04EB
- 30-Pin expansion connector
- User switch and reset switch
- Test points
- DC power switch
- 3 LEDs

Summary of Octopus X

- Octopus X is not only compatible with IAR embedded workbench but also “Keil C ” software
- Octopus X is of 2-Layer design to reduce production cost
- Octopus X can be not only programmed from USB interface but also TI programming board
- RF transmission range of Octopus X is up to 150m
- Expansion connector design on Octopus X provides a user interface for sensor boards and dock

Introduction of Octopus II Hardware Platform

- Octopus II includes two models
  - Octopus II-A
    - MSP430F1611 + USB Interface + Inverted F and SMA Type Antenna
  - Octopus II-B
    - Octopus II-A + External Power Amplifier
- Peripherals of Octopus II
  - Octopus II-Sensor board
    - Temperature sensor
    - Light sensors
    - Gyroscope
    - Three axis accelerometer

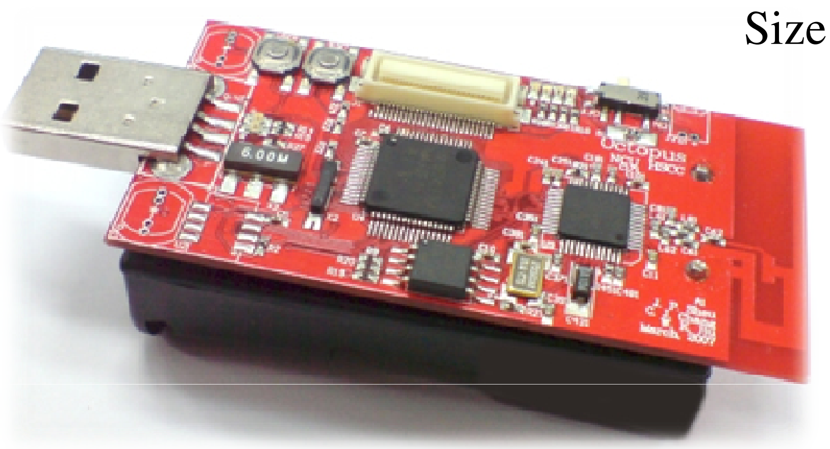
Octopus II-A

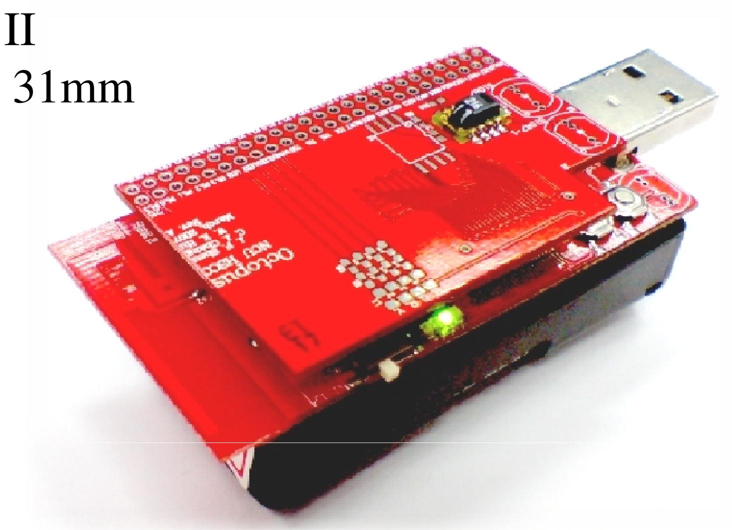
Octopus II-B

Octopus II-Sensor board

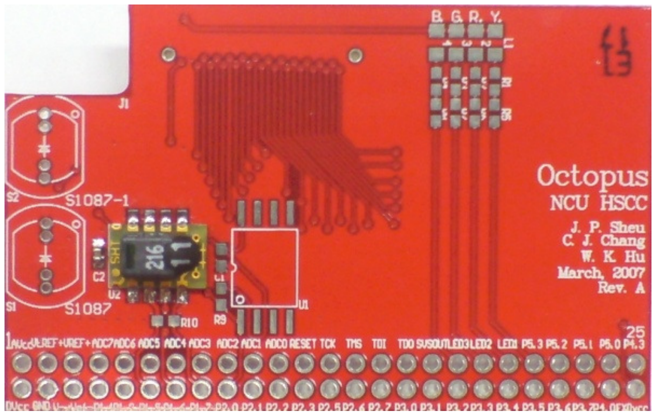
Introduction of Octopus II Hardware Platform

Octopus II
Size: 65mm × 31mm



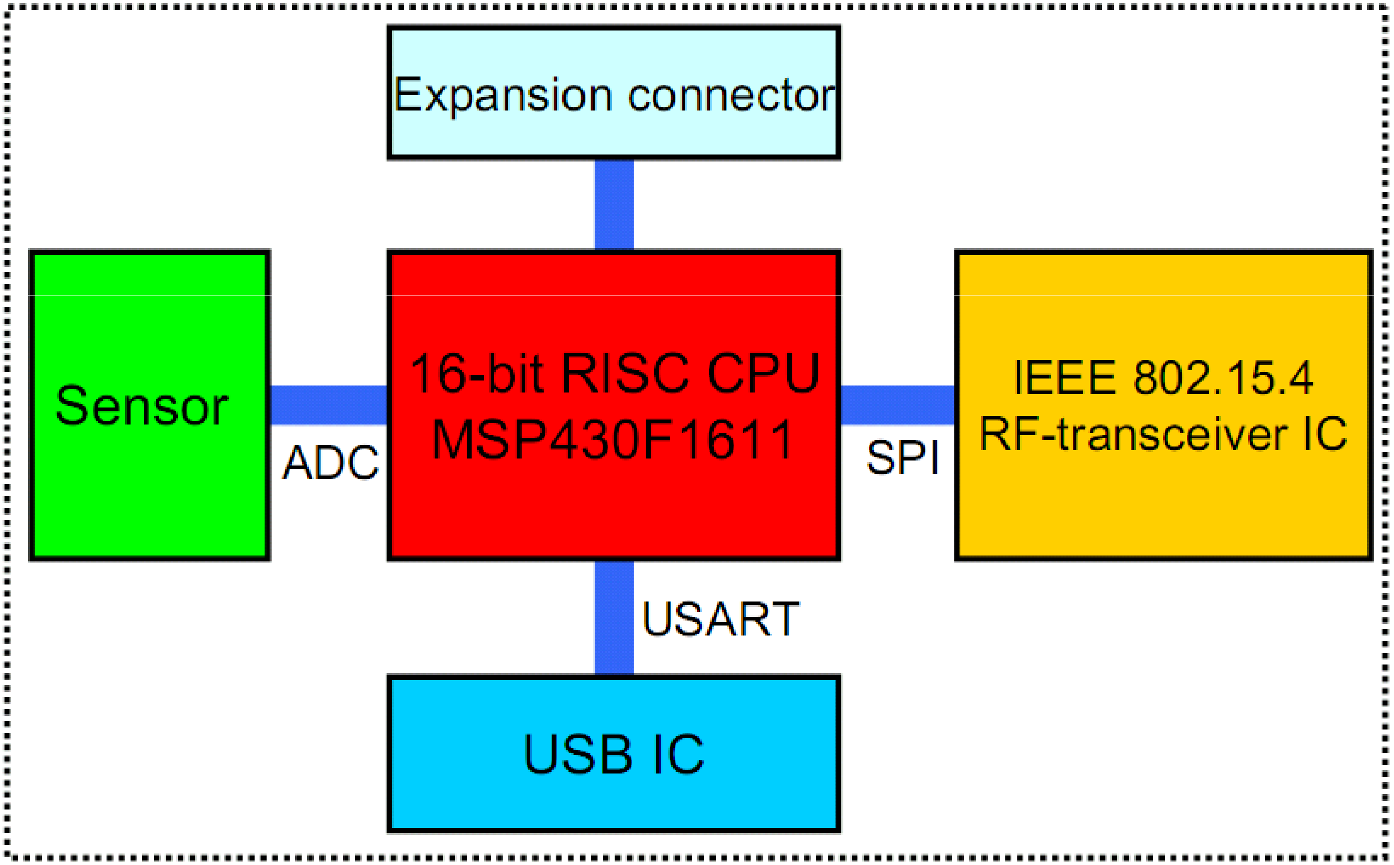


Sensor Board
Size: 50mm × 31mm



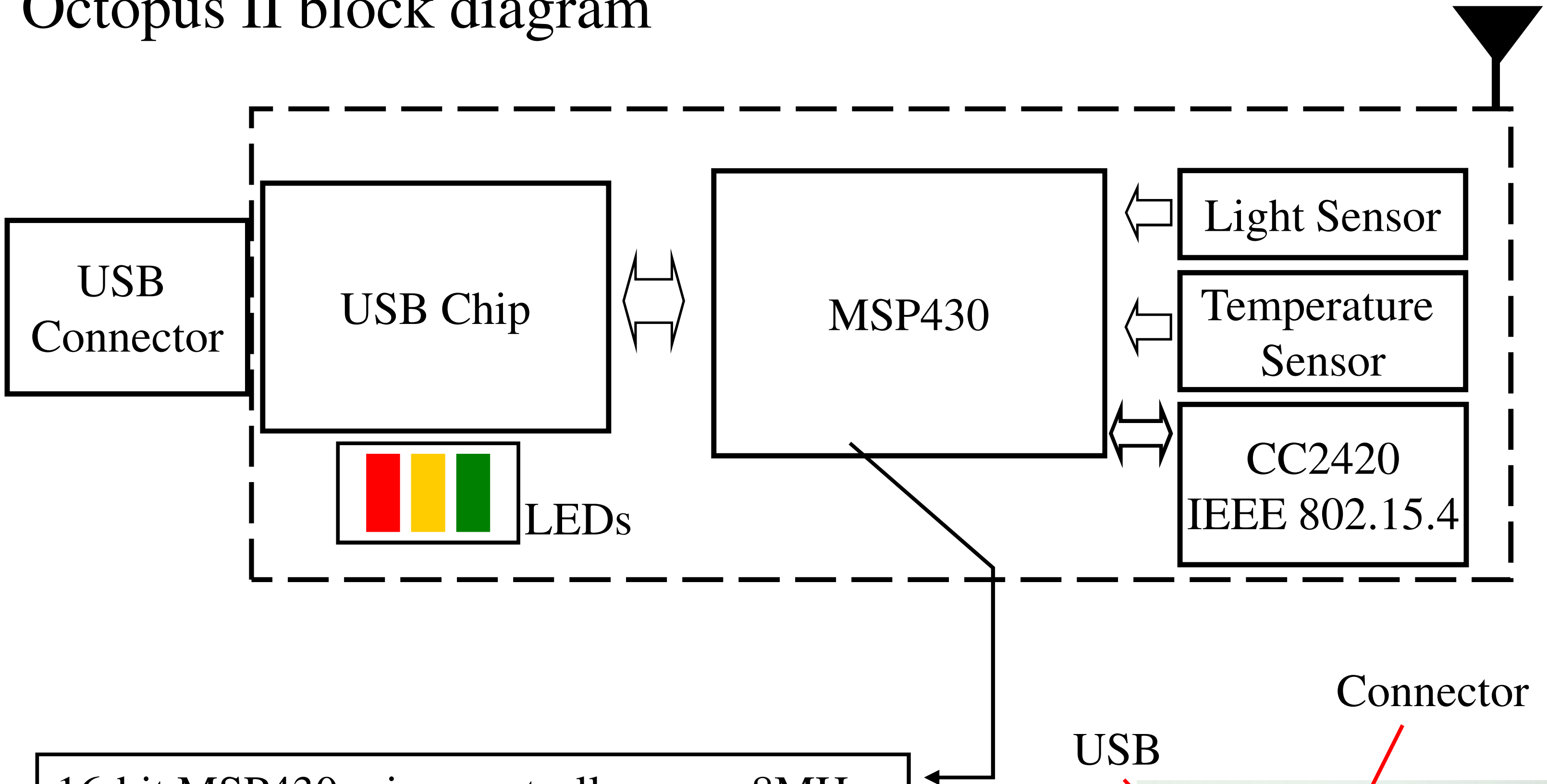
Introduction of Octopus II Hardware Platform

- Octopus II block diagram

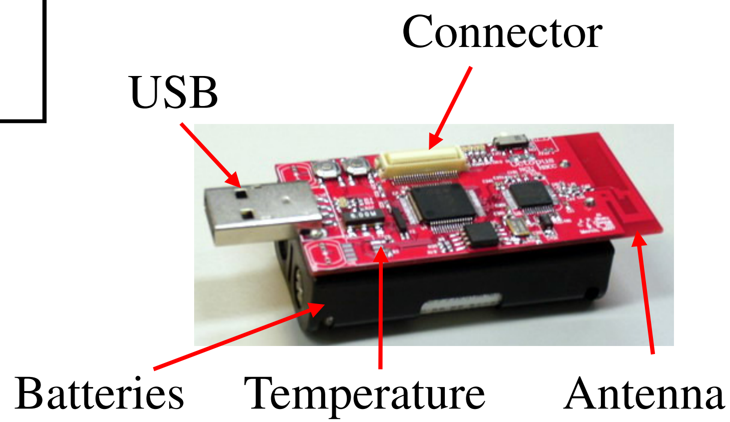


Introduction of Octopus II Hardware Platform

- Octopus II block diagram



16-bit MSP430 microcontroller core 8MHz
48 KB in-system programmable flash
10 KB RAM
ADC 12-Bit 8 Channels



Features of Octopus II-A

- MCU (MSP430F1611)
  - Flash Memory (48 KB + 256 KB)
  - RAM (10 KB)
  - External Flash (1 MB)
  - External Crystal (4 MHz + 32.768 KHz)
  - Serial Communication Interface (USART, SPI or I^2C)
  - Low Supply-Voltage Range (1.8V ~ 3.6V)
  - Five Power-Saving Modes

- Sensors
  - Humidity & Temperature sensor
    - Humidity 0 ~ 100%RH (0.03%RH)
    - Temperature -40^{o}C ~ 120^{o}C (0.01^{o}C)
  - Light sensors

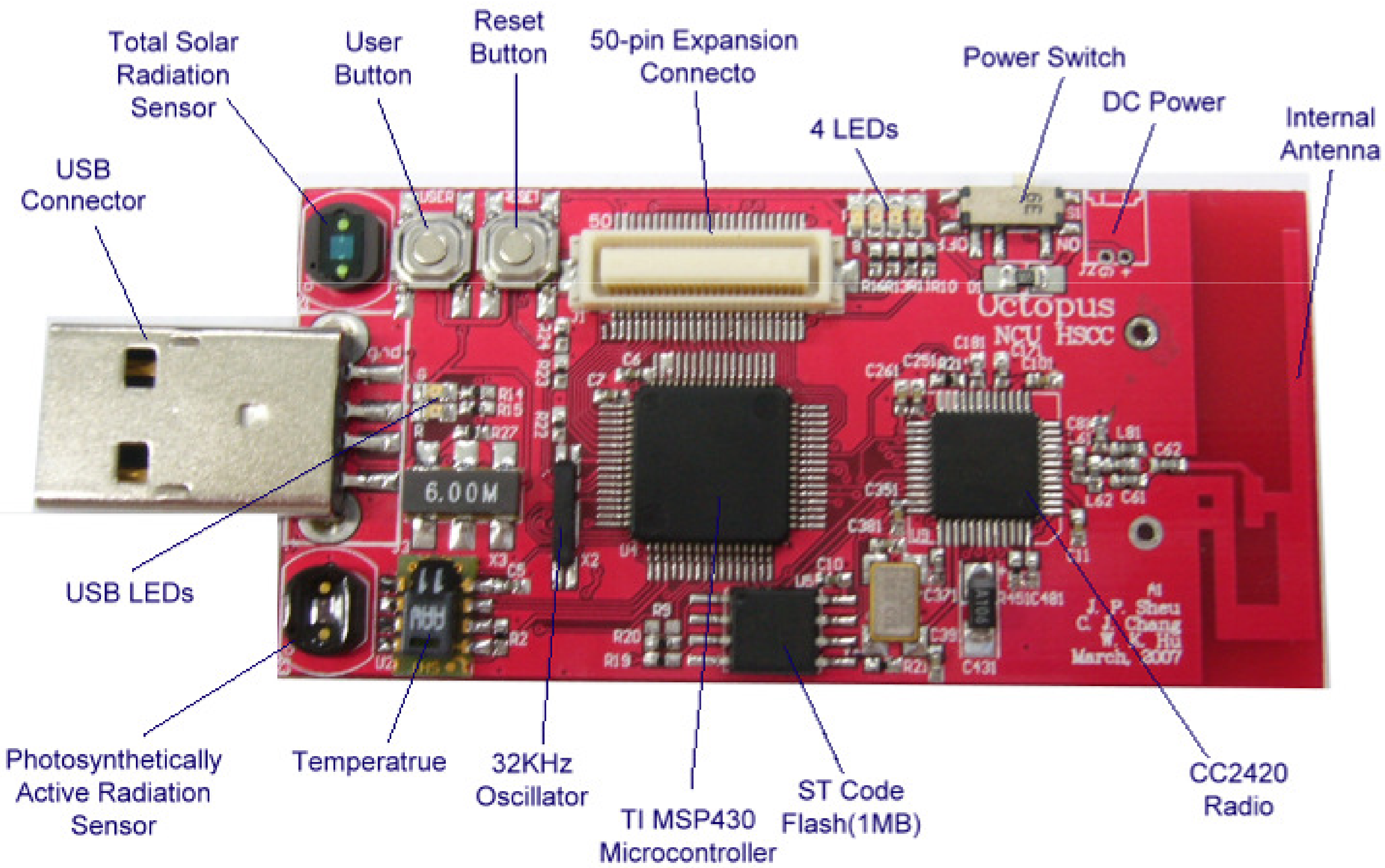
Features of Octopus II-A

- Radio (CC2420)
  - 2.4GHz IEEE 802.15.4 compliant RF
  - Data rate (250 Kbps)
  - Rx (18.8 mA), Tx (17.4 mA)
  - Programmable output power
  - Digital RSSI/LQI support
  - Hardware MAC encryption
  - Battery monitor
  - RF transmission range ≒ 250m
- Serial number ID
- 50-Pin expansion connector
- External DC power connector

Features of Octopus II-A

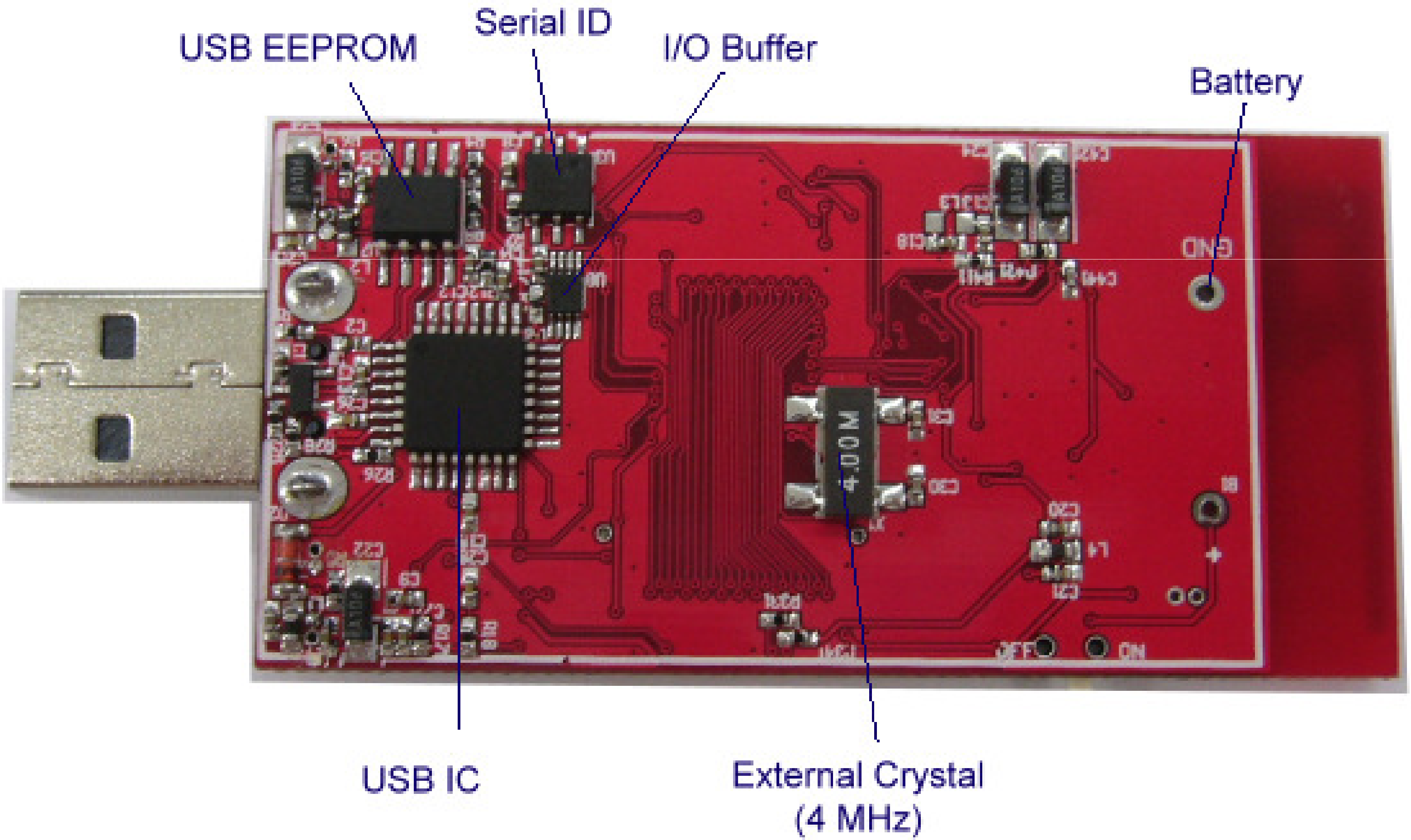
- Front view of Octopus II-A

Size: 65mm × 31mm



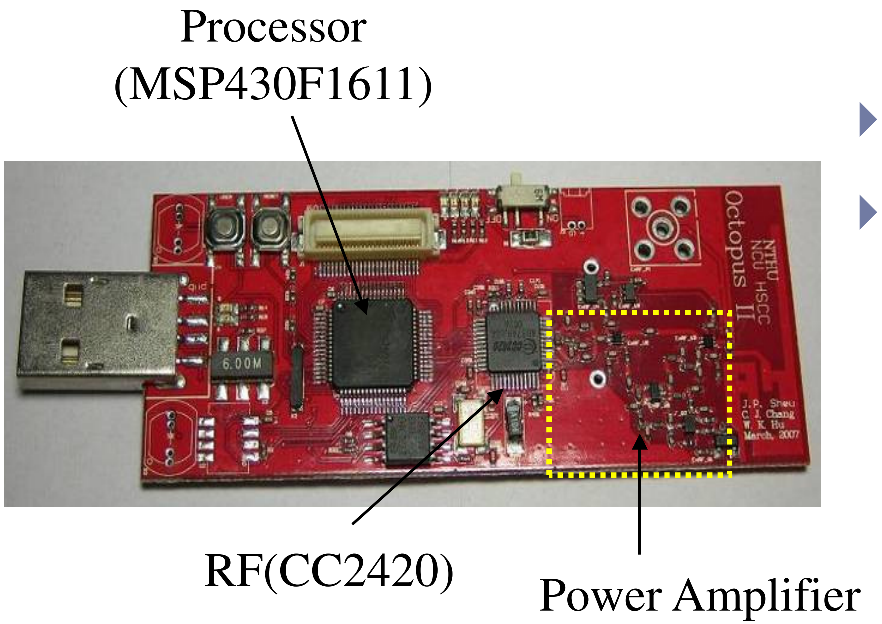
Features of Octopus II-A

- Back view of Octopus II-A



Features of Octopus II-B

Size: 80mm × 31mm

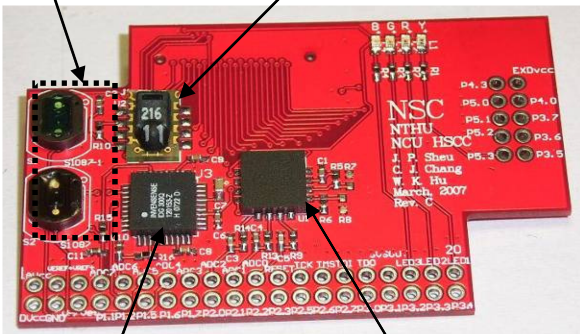


- RF transmission range ≒ 450m
- CC2420 with external power amplifier
- Maximum output power: ~10dBm
- Compliance with IEEE 802.15.4 (ZigBee)

Features of Octopus II - Sensor board

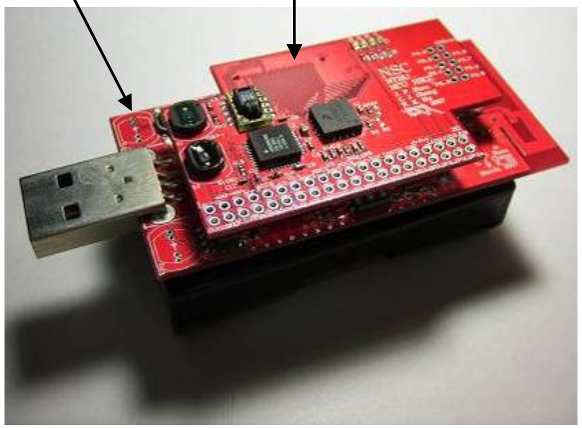
Size: 50mm × 31mm

Light sensors Temperature sensor



Gyroscope Three axis accelerometer

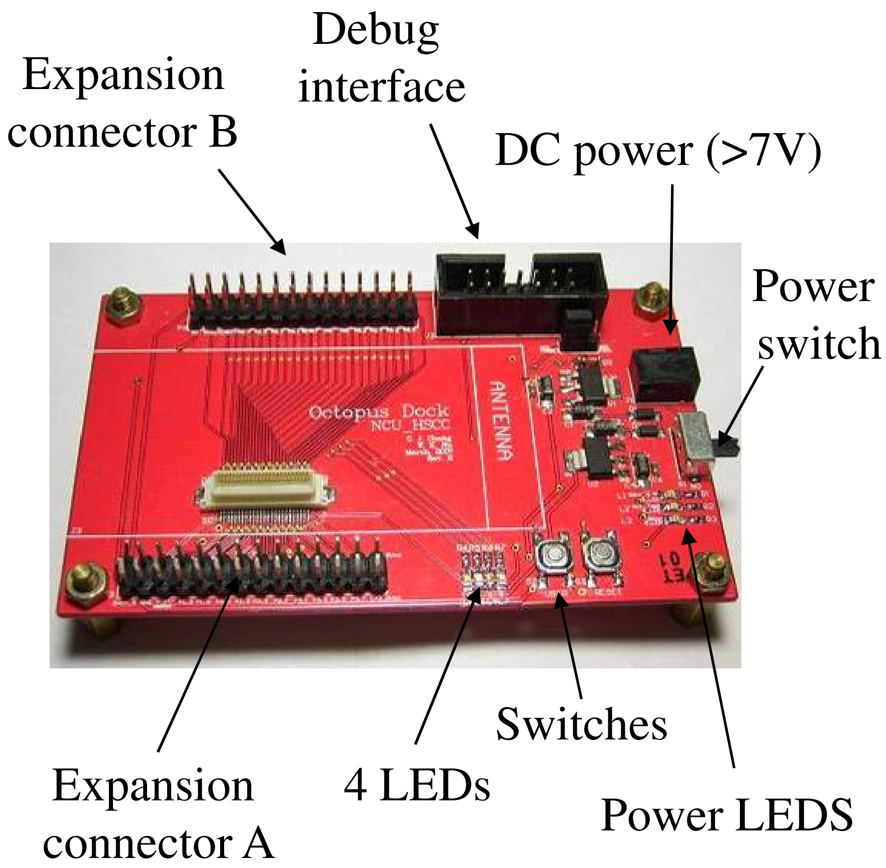
Octopus II Sensor board



- Sensors
  - Humidity & Temperature sensor
    - Humidity 0-100%RH (0.03%RH)
    - Temperature -40°C~120°C (0.01°C)
  - Light sensors
  - Gyroscope
    - Integrated X and Y-axis gyro
  - Three axis accelerometer
    - Selectable sensitivity (1.5g/2g/4g/6g)
    - Low current consumption (600uA)
    - Sleep mode (3uA)
    - Low voltage operation (2.2V-3.6V)
    - High sensitivity (800mV/g @ 1.5g)

Features of Octopus II - Dock

Size: 90mm × 54mm



- Easy-to-develop WSN applications
- Debug interface
  - Programming with TI flash programmer
- DC power input
- Power switch
- 3 power LEDs
- 4 user LEDs
- User switch and reset switch
- 2 row expansion connectors

Summary of Octopus II

- Octopus II is not only compatible with TinyOS but also standard C programming
- Octopus II is of 2-Layer design to reduce production cost
- Octopus II can be programmed from USB interface
- Octopus II has two kinds of antennas, SMA type and inverted F type
- RF transmission range of Octopus II is up to 450m
- Expansion connector design on Octopus II provides a user interface for sensor boards and dock

2.3. Energy Consumption of Sensor Node

The Main Consumers of Energy

- Microcontroller
- Radio front ends
  - RF transceiver IC
  - RF antenna
- Degree of Memory
  - RAM
  - EEPROM
  - Flash memory
- Depending on the type of sensors
  - Temperature sensor
  - Humidity sensor
- Other components
  - LED
  - External Crystal
  - USB IC

Energy consumption of Microcontroller

- A "back of the envelope" estimation for energy consumption
  - It means "energy consumption" is easily to estimate

- Number of instructions
  - Energy per instruction: 1 *nJ* [4]
  - Small battery ("smart dust"): 1 *J* = 1 *Ws*
  - Corresponds: 10^9 instructions!

- Lifetime
  - Require a single day operational lifetime
    = 24hr × 60mins × 60secs = 86400 secs
  - 1 Ws / 86400s ≒ **11.5 µW** as max. sustained power consumption!

- Not feasible!
  - Most of the time a wireless sensor node has nothing to do
  - Hence, it is best to turn it off

Multiple power consumption modes

- Way out: Do not run sensor node at full operation all the time
  - If nothing to do, switch to ***power safe mode***
  - Question: When to throttle down? How to wake up again?
- Typical modes
  - Microcontroller
    - Active, Idle, Sleep
  - Radio mode
    - Turn on/off transmitter/receiver or Both
- Multiple modes possible, “deeper” sleep modes
  - Strongly depends on hardware
  - Ex: TI MSP 430
    - Four different sleep modes
  - Atmel ATMega
    - Six different modes

Some Energy Consumption Figures

- Microcontroller power consumption
  - TI MSP 430 (@ 1 MHz, 3V) [6]
    - Fully operation : 1.2 mW
    - Deepest sleep mode : 0.3 µW
      - Only woken up by external interrupts (not even timer is running any more)
  - Atmel ATMega128L [7]
    - Operational mode:
      - Active : 15 mW
      - Idle : 6 mW
    - Sleep mode : 75 µW

Some Energy Consumption Figures

- TI CC2430[8] & 2431 [9]
  - MCU Active Mode, static : 492 μA
    - No radio, crystals, or peripherals
  - MCU Active Mode, dynamic : 210μA/MHz
    - No radio, crystals, or peripherals
  - MCU Active Mode, highest speed : 7.0 mA
    - MCU running at full speed (32MHz)
    - No peripherals
  - Power mode 1 : 296μA
    - RAM retention
  - Power mode 2 : 0.9 μA
    - RAM retention
  - Power mode 3: 0.6μA
    - No clocks, RAM retention

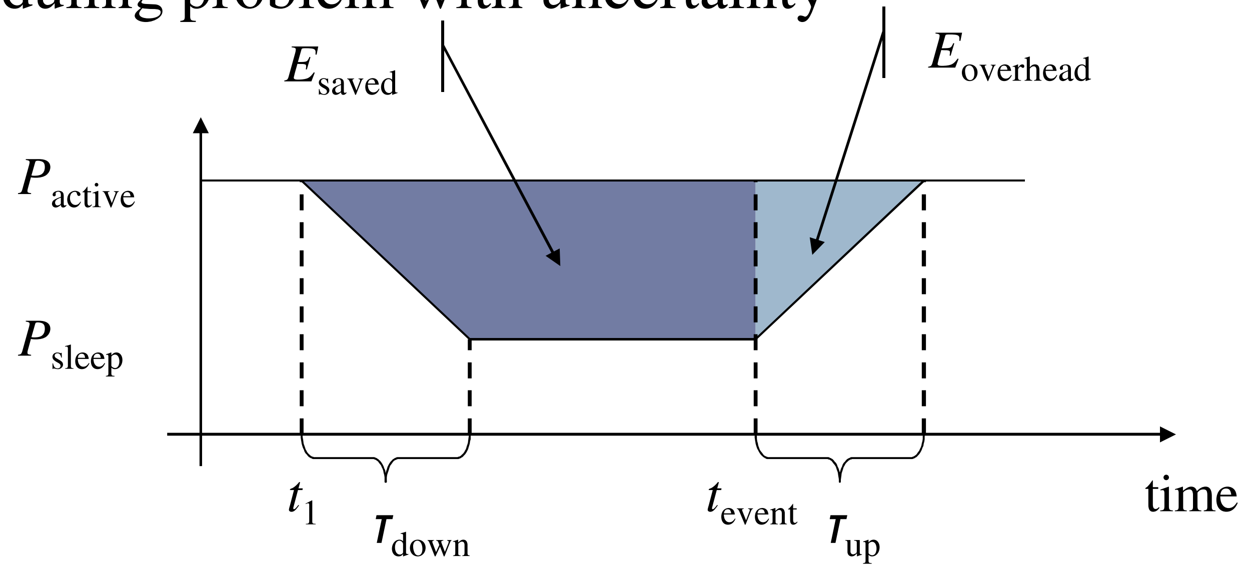
Some Energy Consumption Figures

- Memory power consumption
  - Power for RAM almost negligible
  - FLASH memory is crucial part

  - FLASH writing/erasing is expensive
    - Example: FLASH on Mica motes
    - Reading: $\doteqdot$ 1.1 *nAh* per byte
    - Writing: $\doteqdot$ 83.3 *nAh* per byte

Switching between Modes

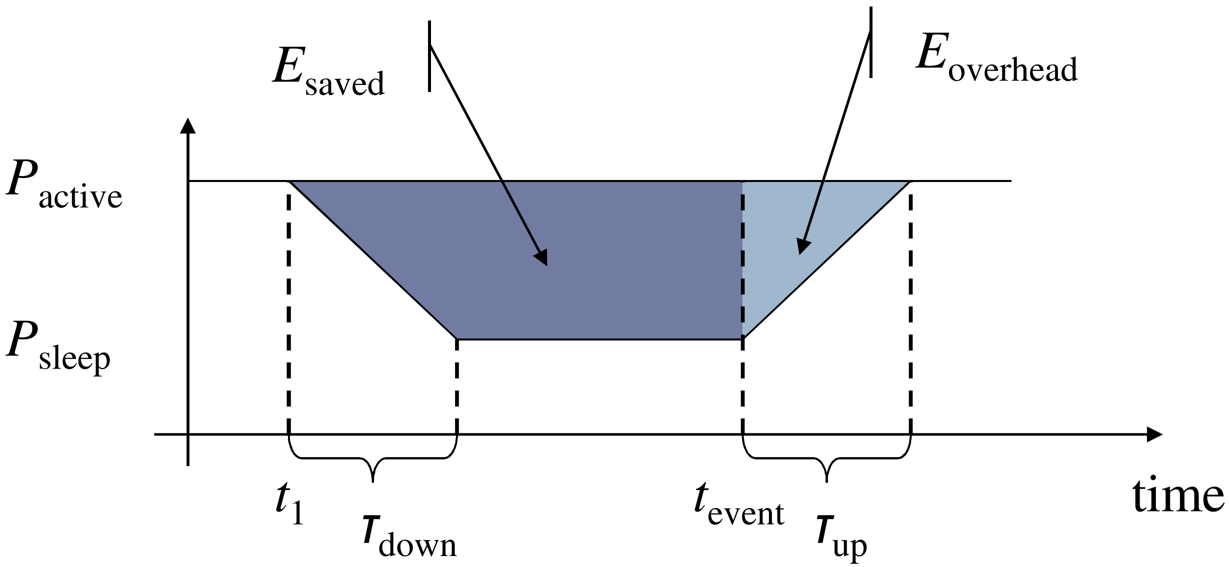
- Simplest idea: Greedily switch to lower mode whenever possible
- Problem: Time and power consumption required to reach higher modes not negligible
  - Introduces overhead
  - Switching only pays off if $E_{saved} > E_{overhead}$
- Example:
  Event-triggered wake up from sleep mode
- Scheduling problem with uncertainty



Switching between Modes

- $E_{\text{saved}} = (t_{\text{event}} - t_1) \times P_{\text{active}} - (\tau_{\text{down}} \times (P_{\text{active}} + P_{\text{sleep}}) / 2 + (t_{\text{event}} - t_1 - \tau_{\text{down}}) \times P_{\text{sleep}})$

- $E_{\text{overhead}} = \tau_{\text{up}} \times (P_{\text{active}} + P_{\text{sleep}}) / 2$



Power Consumption vs. Transmission Distance

- Free space loss: direct-path signal

$$P_r = P_t G_r G_t \frac{\lambda^2}{(4\pi)^2 (d)^2} = P_t \frac{A_r A_t}{(\lambda d)^2}$$

  - d = distance between transmitter and receiver
  - P_t = transmitting power
  - P_r = receiving power
  - G_t = gain of transmitting antenna
  - G_r = gain of receiving antenna
  - A_t = effective area of transmitting antenna
  - A_r = effective area of receiving antenna

Power Consumption vs. Transmission Distance

- Two-path model

$$P_r = P_t G_r G_t \left(\frac{h_t h_r}{d^2}\right)^2$$

- h_t and h_r are the height of the transmitter and receiver

- The general form

$$P_r = P_t G_r G_t \left(\frac{\lambda}{4\pi}\right)^2 \frac{1}{d^\gamma}$$

  - γ is the propagation coefficient that varies 2 ~ 5

Computation vs. Communication Energy Cost

- Tradeoff ?
  - It's not possible to directly compare computation/communication energy cost
  - Energy ratio of "sending one bit" vs. "computing one instruction"
  - Communicate (send & receive) 1 KB ≒ Computing 3,000,000 (3 million) instructions [10]

- Hence
  - Try to compute instead of communicate whenever possible

- Key technique in WSN
  - In-network processing
  - Exploit data centric/aggregation, data compression, intelligent coding, signal processing …

2.4. Network Architecture

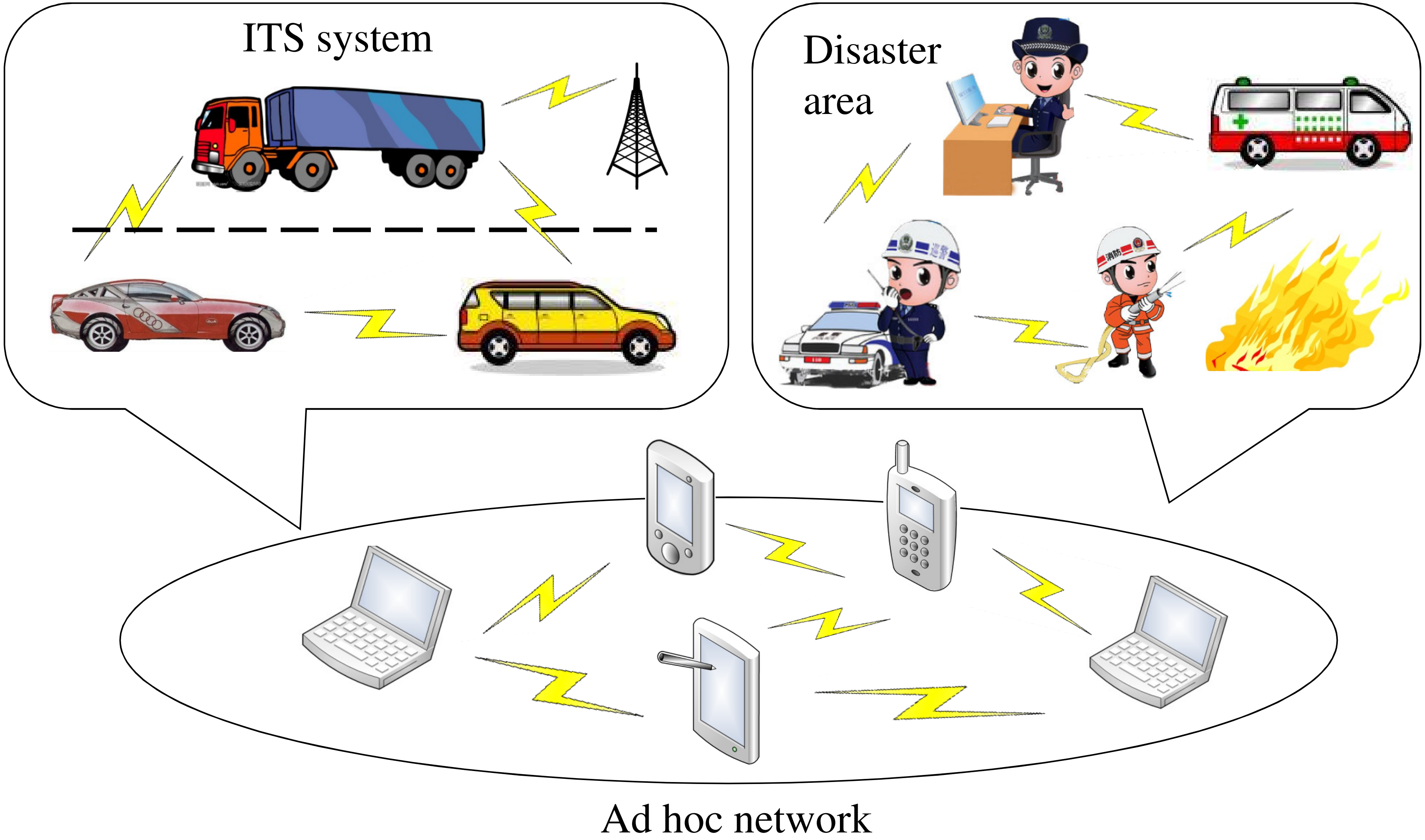
Difference between Ad hoc and Sensor Networks

- (Mobile) Ad hoc Scenarios
  - Nodes communicate with each other
    - That means each node can be a source node or destination node
  - Nodes can communicate "some" node in another network
    - Ex: Access to Web/Mail/DNS server on the Internet
    - Typically requires some connection to the fixed network

- Applications of Ad hoc networks
  - Traditional data (http, ftp, collaborative apps, …)
  - Multimedia (voice, video)

Difference between Ad hoc and Sensor Networks

- (Mobile) Ad hoc Scenarios



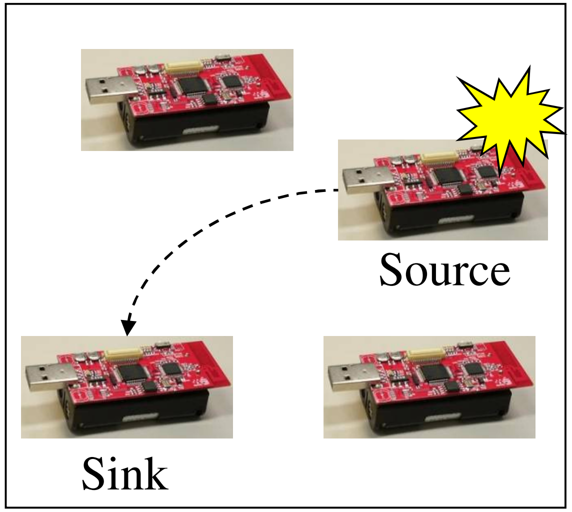
Difference between Ad hoc and Sensor Networks

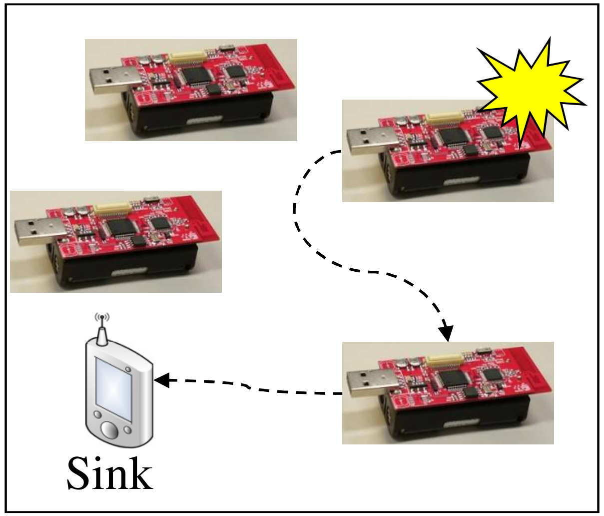
- Sensor Network Scenarios
  - **Sources**: Any sensor node that provides sensing data/measurements
  - **Sinks**: Sensor nodes where information is required
    - Belongs to the sensor network
    - Could be the same sensor node or an external entity such PDA/NB/Table PC
    - Is part of an external network (e.g., internet), somehow connected to the WSN

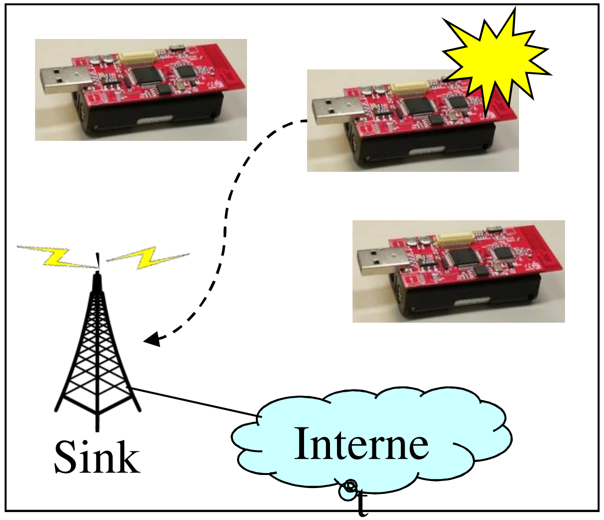
- Applications of Sensor Network
  - Usually, machine to machine
  - Often limited amounts of data
  - Different notions of importance

Difference between Ad hoc and Sensor Network

- Sensor Network Scenarios



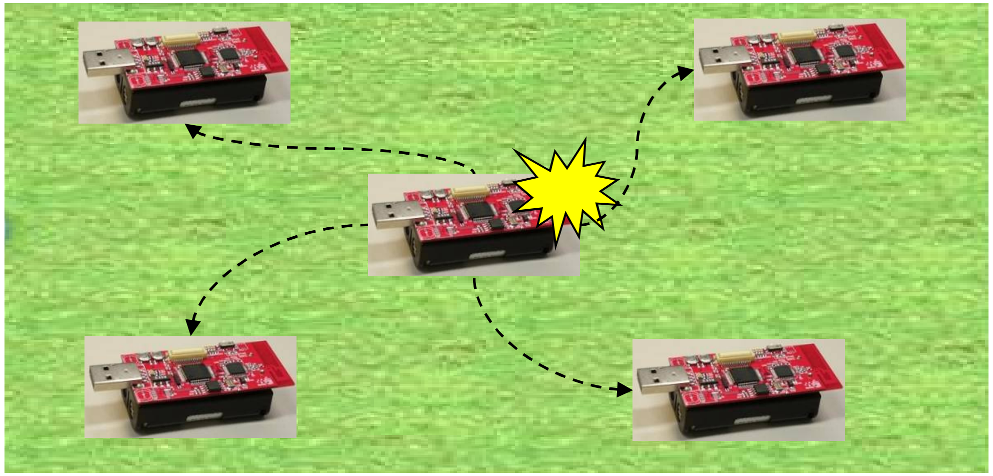




Single-hop vs. Multi-hop Networks

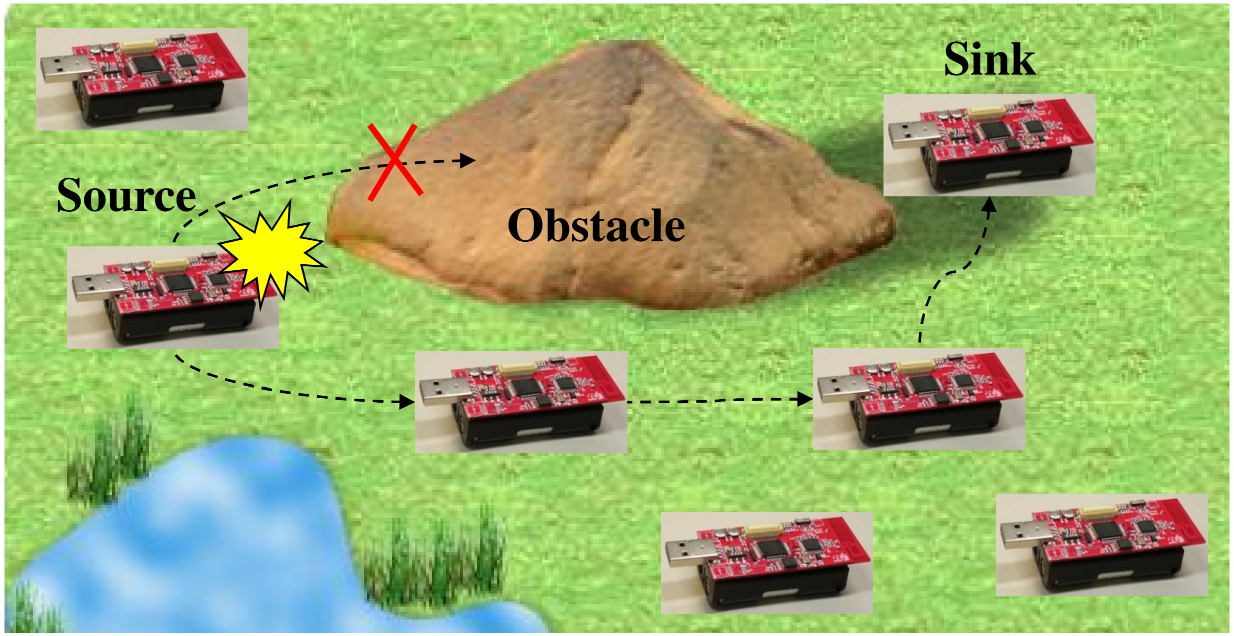
- One common problem: limited range of wireless communication
  - Limited transmission power
  - Path loss
  - Obstacles
- Solution: multi-hop networks
  - Send packets to an intermediate node
  - Intermediate node forwards packet to its destination
  - **Store-and-forward** multi-hop network
- Basic technique applies to both WSN and MANET
- Note:
  - Store-and-forward multi-hopping NOT the only possible solution
    - Ex: Collaborative networking, Network coding [11] [12]….

Single-hop vs. Multi-hop Networks

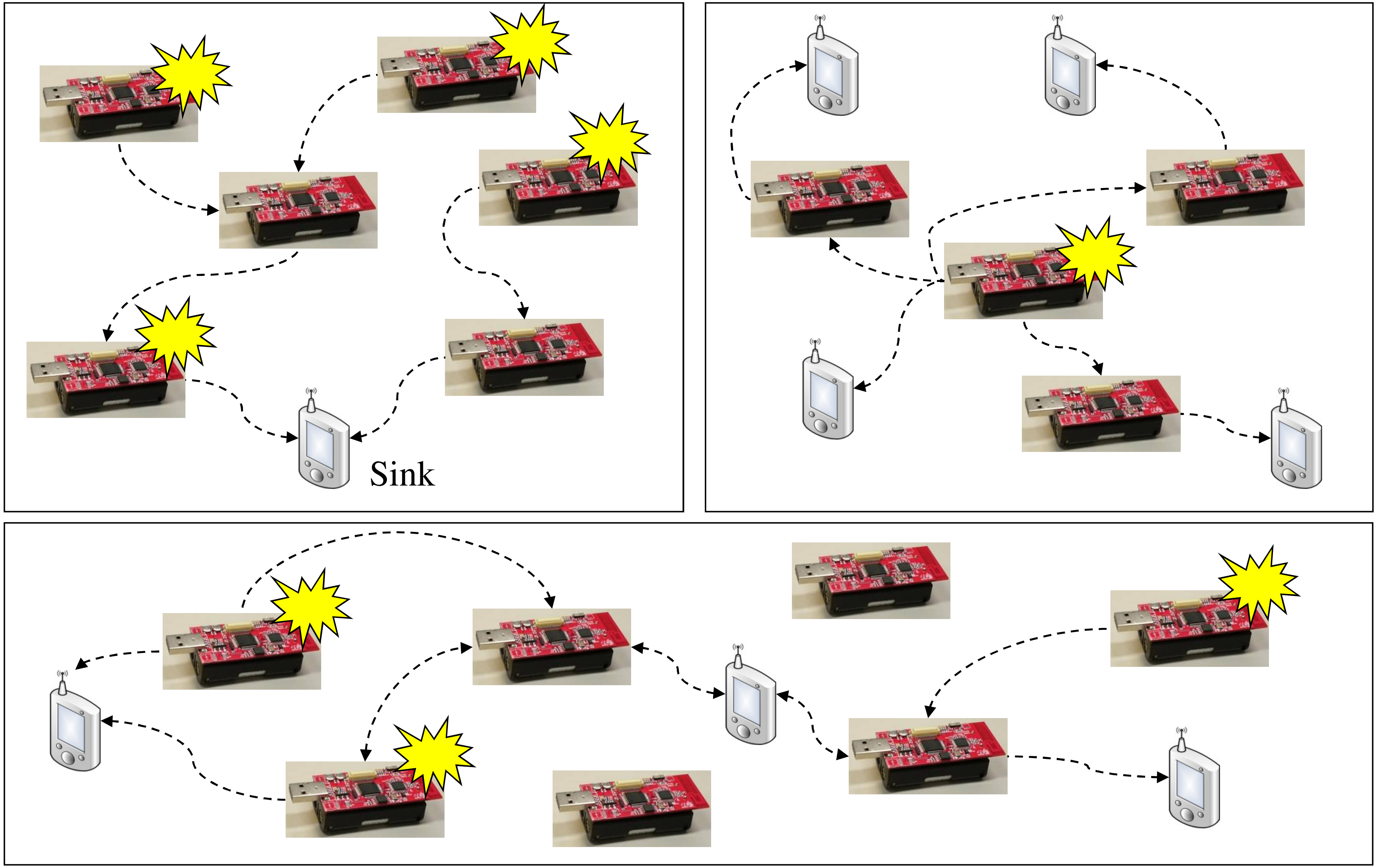


Single-hop networks

Multi-hop networks



Multiple Sinks, Multiple Sources WSN

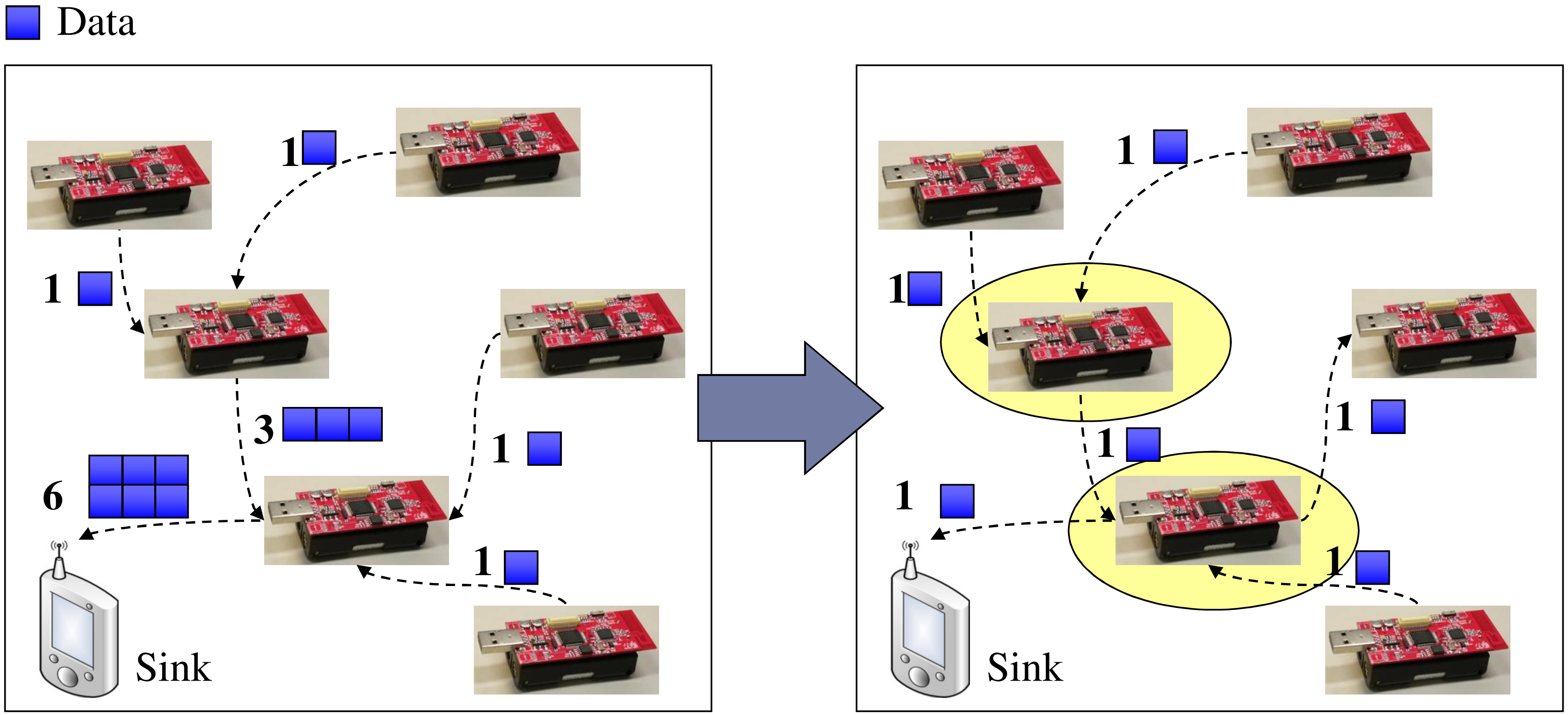


In-network Processing

- MANETs are supposed to deliver bits from one end to the other
- WSNs, on the other end, are expected to provide information, not necessarily original bits
  - Ex: ***manipulate*** or ***process*** the data in the network
- Main example: aggregation
  - Apply composable **[13]** aggregation functions to a convergecast tree in a network
  - Typical functions: minimum, maximum, average, sum, …

In-network Processing

- Processing Aggregation example
  - The simplest in-network processing technique
  - Reduce number of transmitted bits/packets by applying an aggregation function in the network

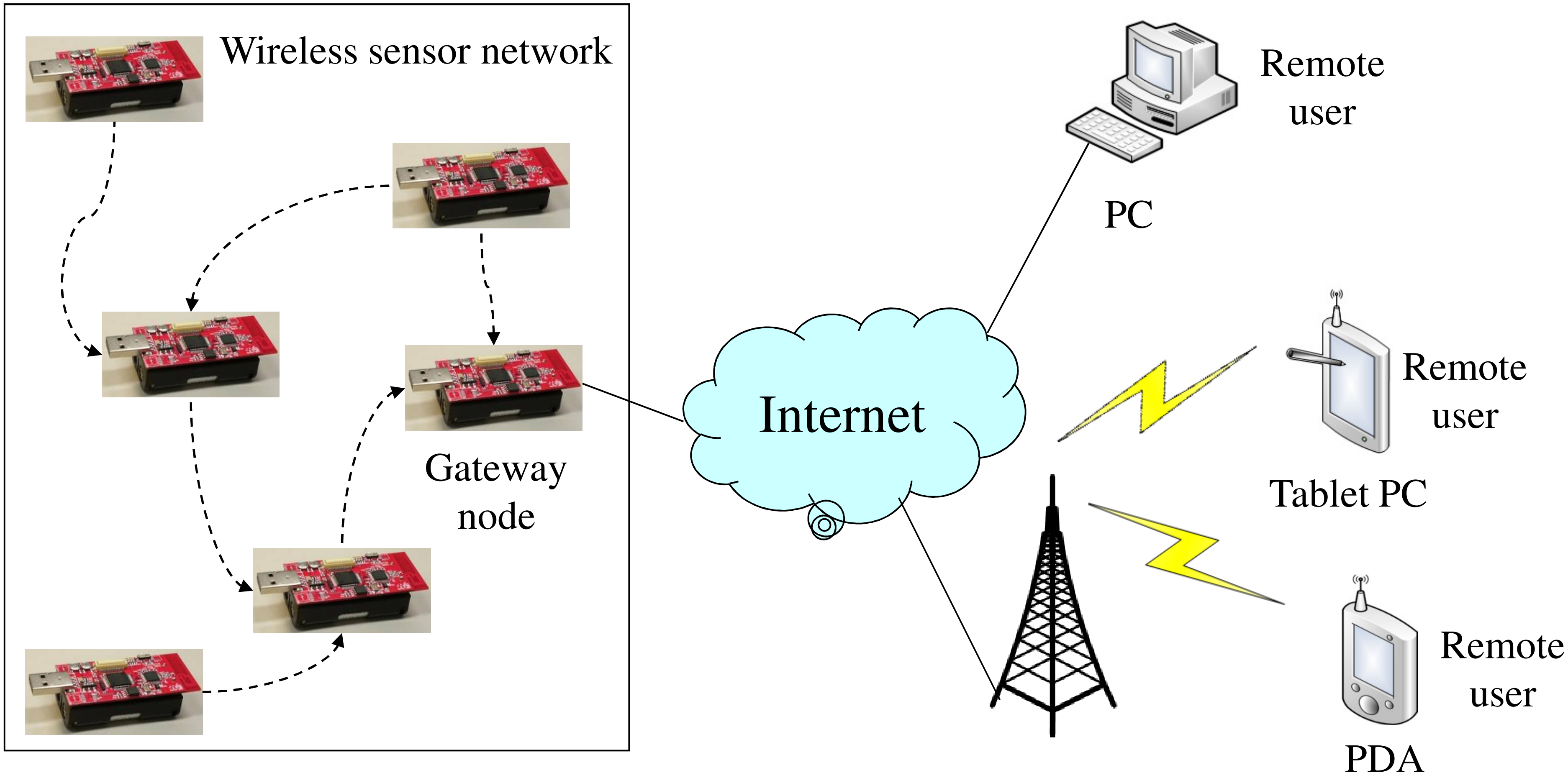


Gateway concepts for WSN/MANET

- Gateways are necessary to the Internet for remote access to/from the WSN
  - For ad hoc networks
    - Additional complications due to mobility
      - Ex: Change route to the gateway, use different gateways
  - For WSN
    - Additionally bridge the gap between different interaction semantics in the gateway

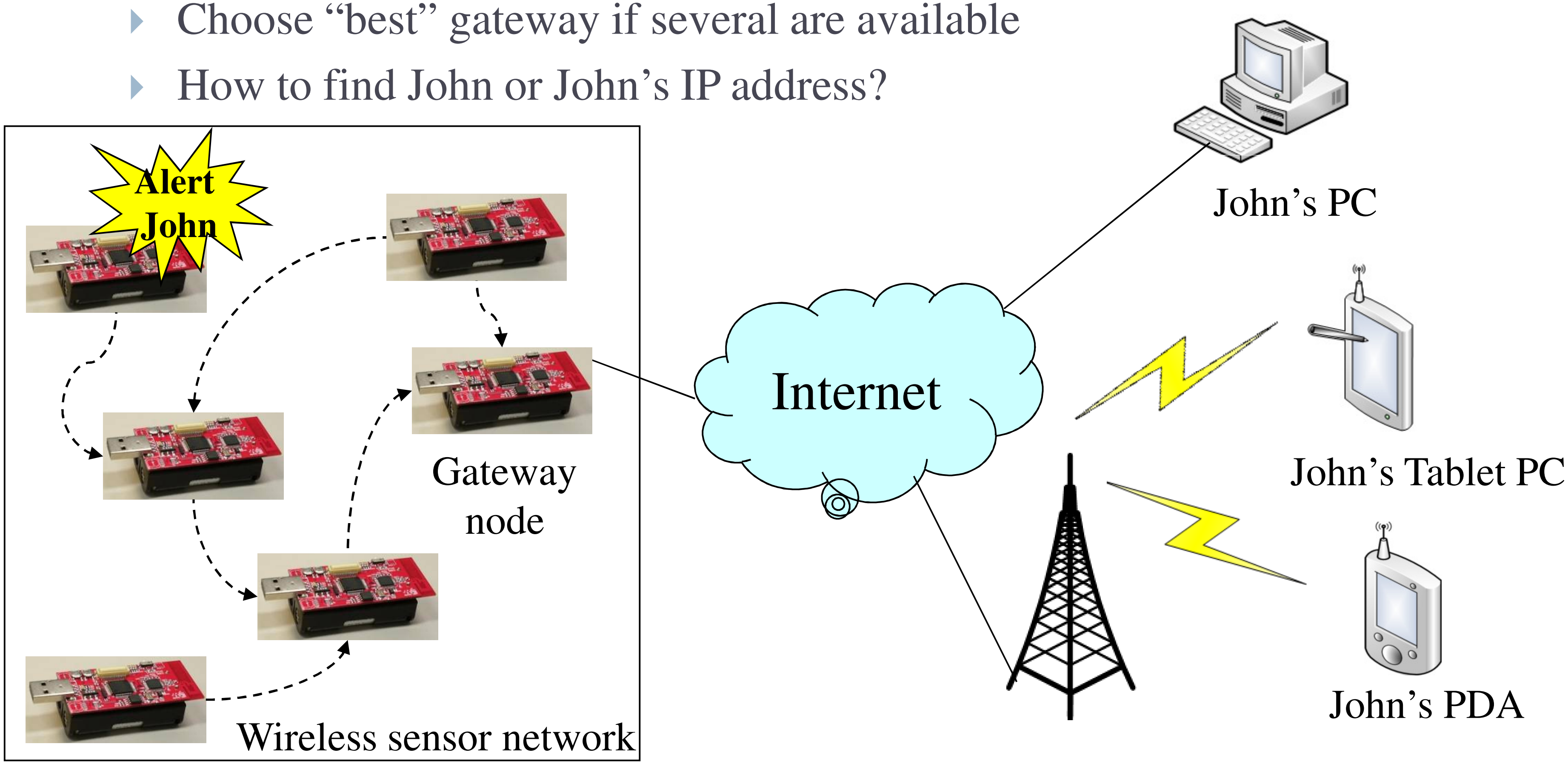
Gateway concepts for WSN/MANET

- Gateway support for different radios/protocols, …



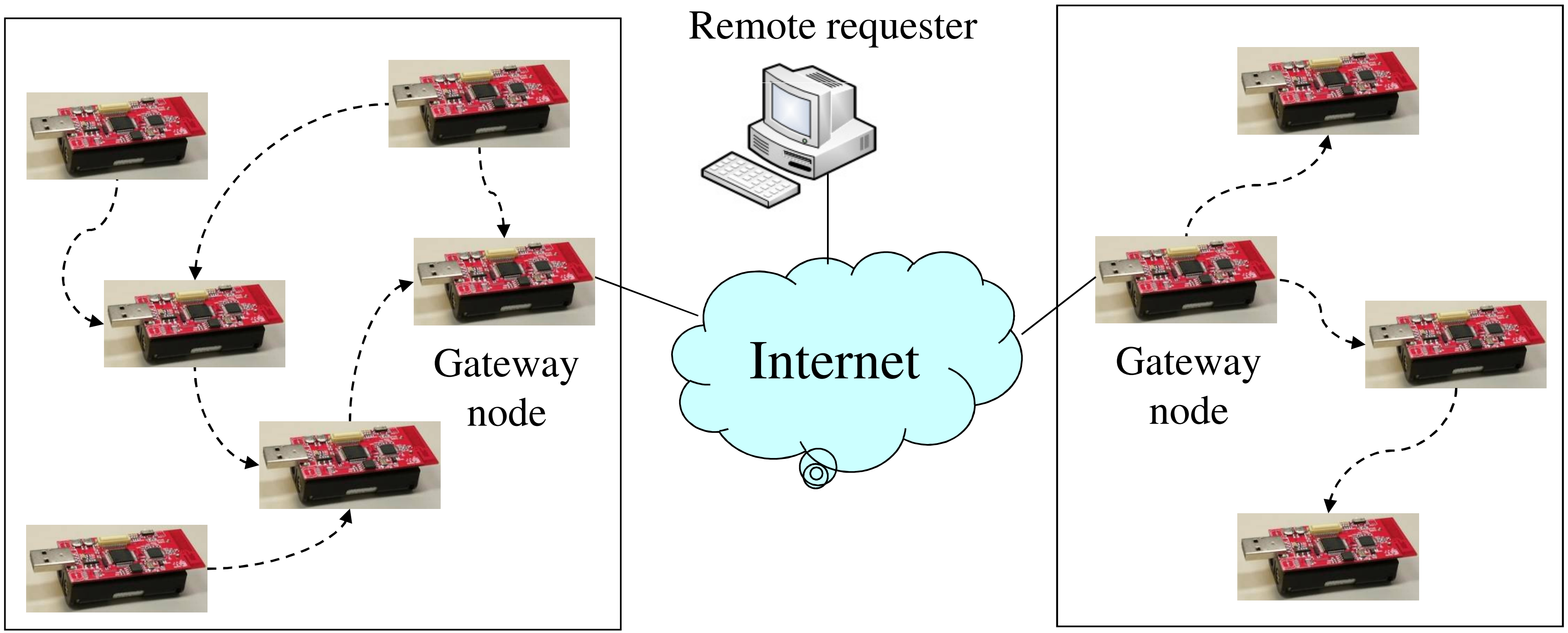
WSN to Internet communication

- Scenario: Deliver an alarm message to an Internet host
- Problems
  - Need to find a gateway (integrates routing & service discovery)
  - Choose "best" gateway if several are available
  - How to find John or John's IP address?



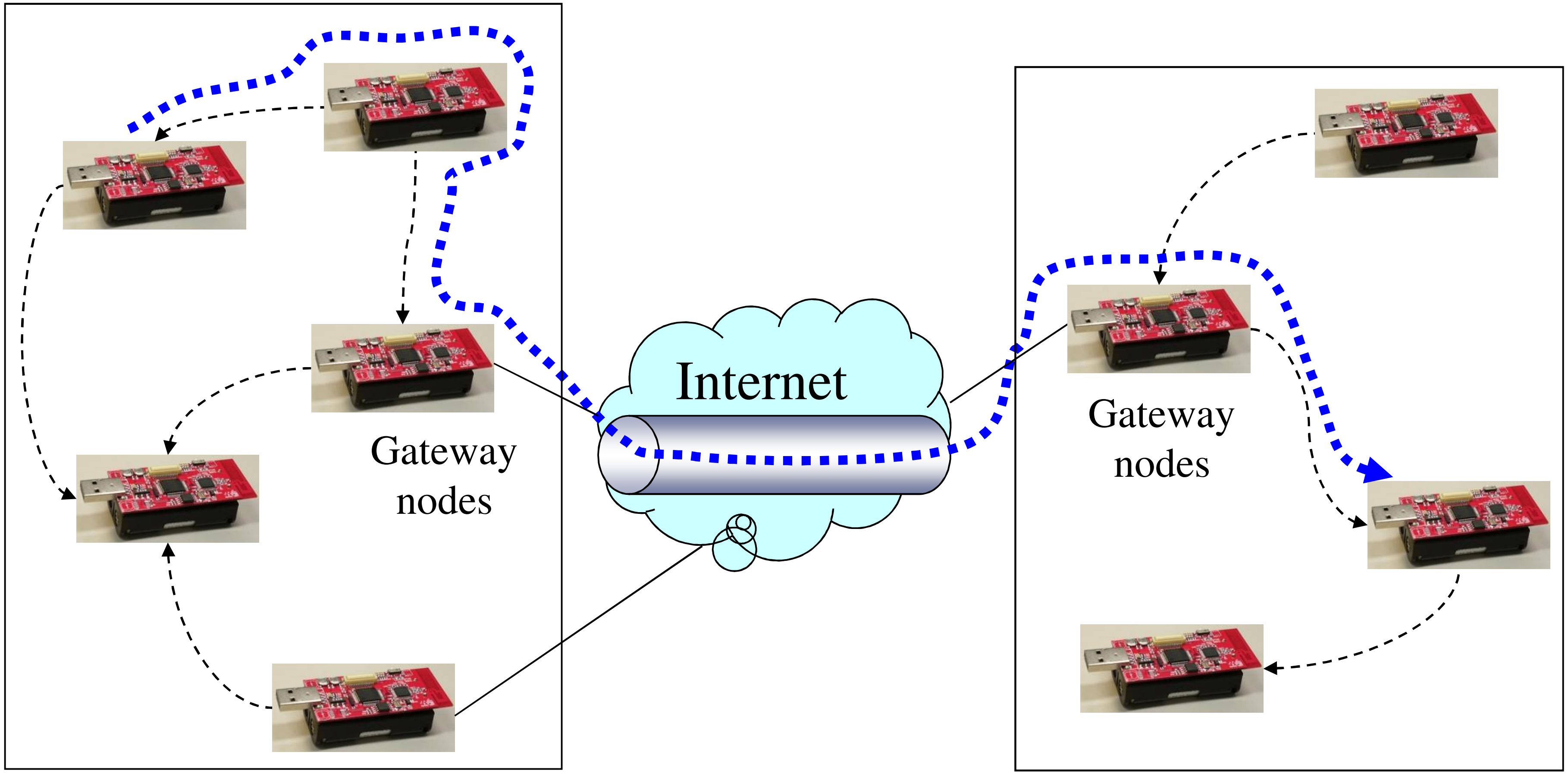
Internet to WSN communication

- How to find the right WSN to answer a need?
- How to translate from IP protocols to WSN protocols, semantics?



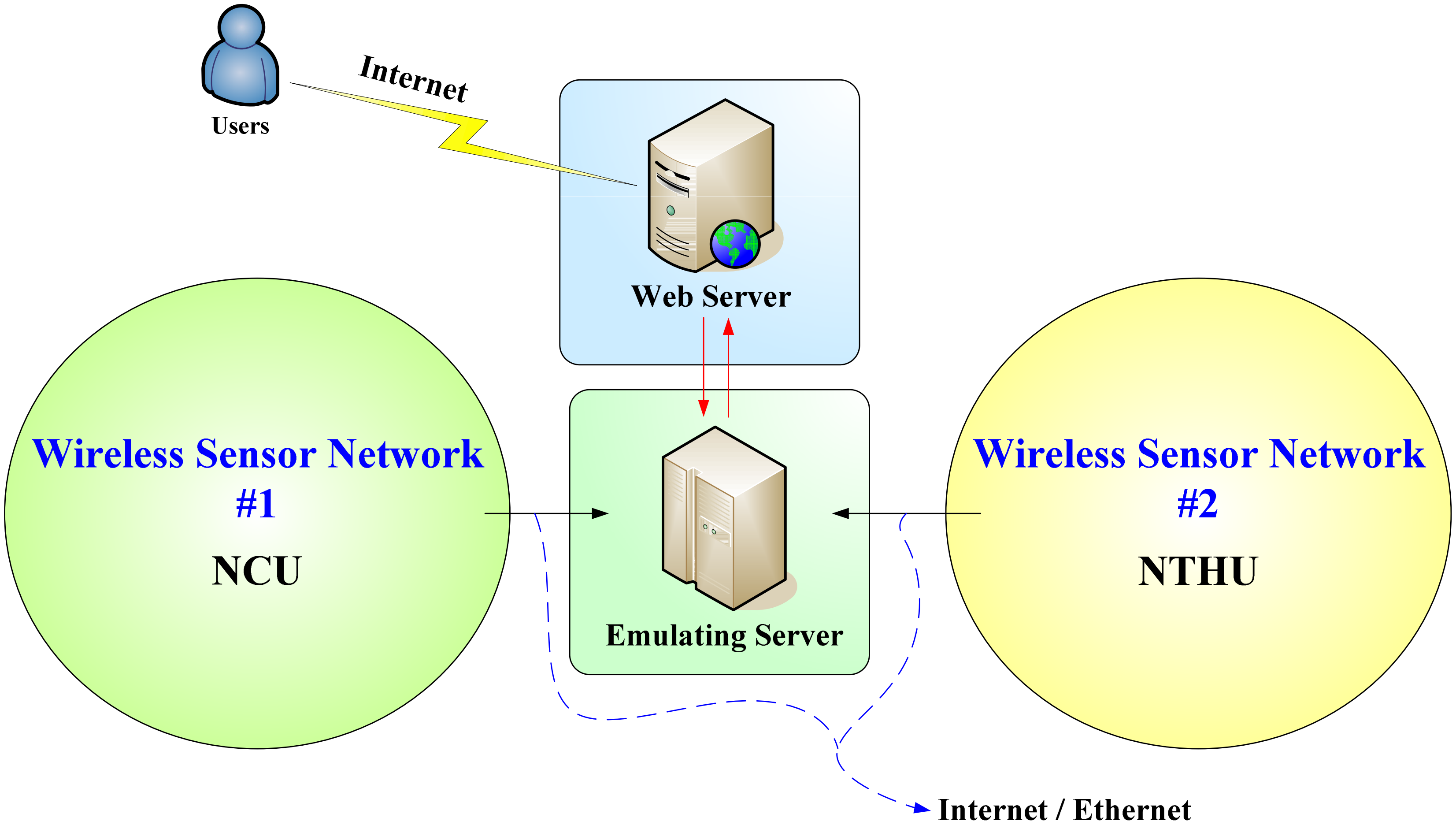
WSN Tunneling

- The idea is to build a larger, "Virtual" WSN
- Use the Internet to "tunnel" WSN packets between two remote WSNs



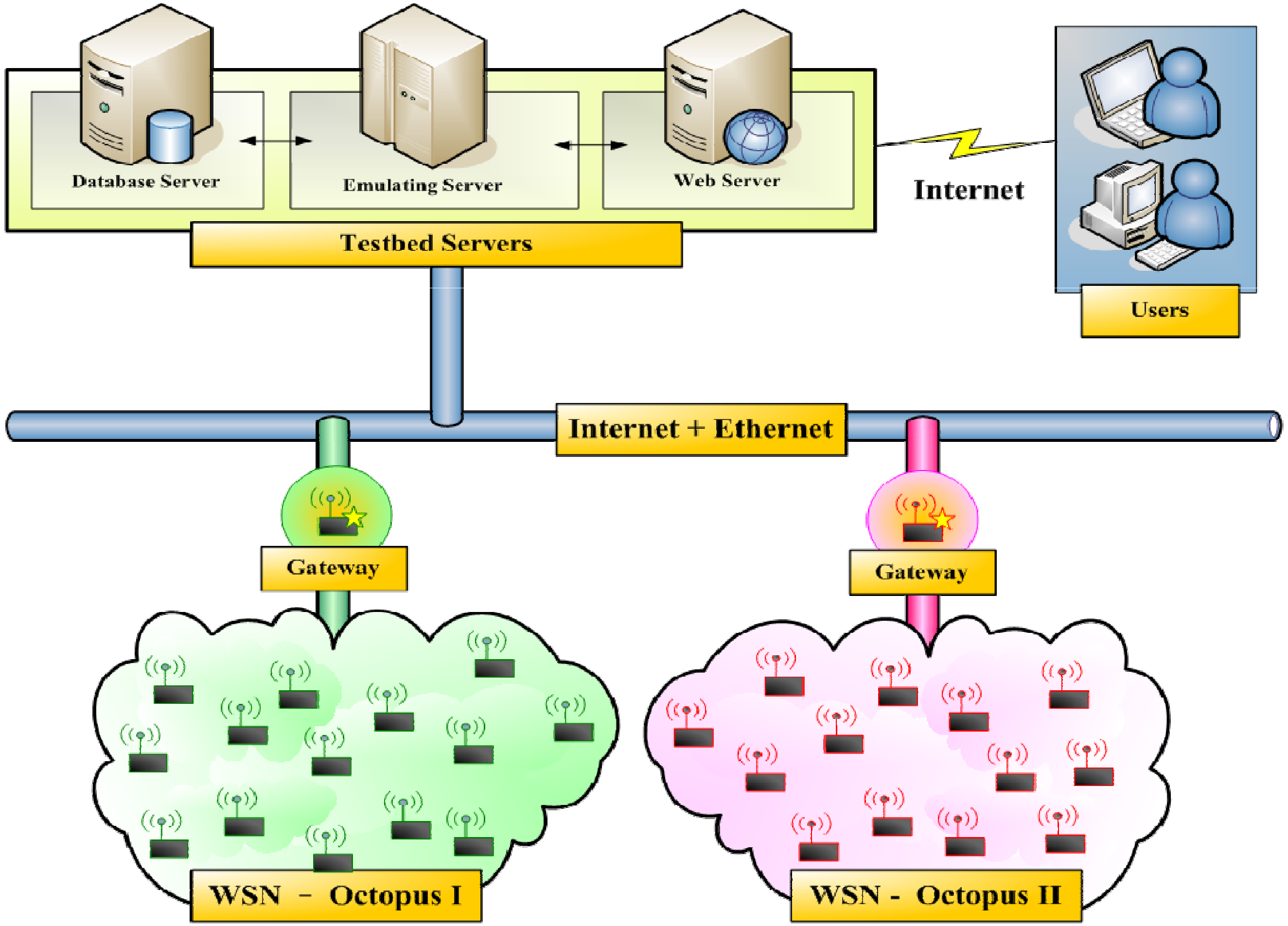
WSN Tunneling

- Example of WSN tunneling
  - WSNs Testbed



WSN tunneling

- Example of WSN tunneling
  - Testbed scenario



2.5. Challenges of Sensor Nodes

Challenges of Wireless Sensor Node

- More energy-efficient
  - Self-sufficiency in power supply such as the installation of solar collector panels
  - Design more energy-efficient of the circuit, or to adopt more energy-efficient electronic components
- Integrating more sensors
  - For multiple purposes such as detecting human's motion, temperature, blood pressure and heartbeat at the same time
- Higher processing performance
  - In future, more complex application need more powerful computation

Challenges of Wireless Sensor Node

- More Robust and Secure
  - Not easy damaged or be destroyed
  - Secure transmission of sensing data and not easy being tapped

- Easy to buy and deployment
  - Low price and easy to use

2.6. Summary

Summary

- For WSN, the need to build cheap, low-energy, (small) devices has various consequences for system design
  - Radio front ends and controllers are much simpler than in conventional mobile networks
  - Energy supply and scavenging are still (and for the foreseeable future) a premium resource
  - Power management (switching off or throttling down devices) crucial

- Unique programming challenges of embedded systems
  - Concurrency without support, protection
  - Lack of standard

References

- [1] V. Raghunathan, C. Schurgers, S. Park, and M. B. Srivastava. Energy-Aware Wireless Microsensor Networks. *IEEE Signal Processing Magazine*, 19: 40–50, 2002.
- [2] S. Roundy, D. Steingart, L. Frechette, P. Wright, and J. Rabaey. Power Sources for Wireless Sensor Networks. In H. Karl, A. Willig, and A. Wolisz, editors, *Proceedings of 1st European Workshop on Wireless Sensor Networks (EWSN)*, pp. 1-17. LNCS, Springer, Berlin, Germany, Vol. 2920, Jan. 2004.
- [3] J. M. Rabaey, M. J. Ammer, J. L. da Silva, D. Patel, and S. Roundy. PicoRadio Supports Ad Hoc Ultra-Low Power Wireless Networking. *IEEE Computer*, 33(7): 42–48, 2000.
- [4] J. M. Kahn, R. H. Katz, and K. S. J. Pister. Emerging Challenges: Mobile Networking for Smart Dust. *Journal of Communications and Networks*, 2(3): 188–196, 2000.
- [5] J. M. Kahn, R. H. Katz, and K. S. J. Pister. Next Century Challenges: Mobile Networking for "Smart Dust". In *Proceedings of ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom 99)*, Seattle, WA, Aug. 1999.
- [6] MSP430x1xx Family User's Guide. Texas Instruments product documentation. 2004.

References

- [7] ATmega 128(L) Preliminary Complete. ATmel product documentation, 2004.
- [8] TI CC2430, http://focus.ti.com/docs/prod/folders/print/cc2430.html
- [9] TI CC2431, http://focus.ti.com/docs/prod/folders/print/cc2431.html
- [10] G. J. Pottie and W. J. Kaiser. Embedding the Internet: Wireless Integrated Network Sensors. *Communications of the ACM*, 43(5): 51–58, 2000.
- [11] R. Ahlswede, N. Cai, S.-Y. R. Li, and R. W. Yeung. Network Information Flow. *IEEE Transaction on Information Theory*, 46(4): 1204–1216, 2000.
- [12] S.-Y. R. Li, R. W. Yeung, and N. Cai. Linear Network Coding. *IEEE Transactions on Information Theory*, 49(2): 371–381, 2003.
- [13] I. Gupta, R. van Renesse, and K. P. Birman. Scalable Fault-Tolerant Aggregation in Large Process Groups. In *Proceedings of the International Conference on Dependable Systems and Networks*, Goteborg, Sweden, July 2001. http://www.cs.cornell.edu/gupta/gupta_aggregn_dsn01.ps.

Recommend Reading

- Wireless sensor node concept
  - G.J. Pottie and W.J. Kaiser, Wireless Integrated Network Sensors, *Communication of the ACM*, Vol.43, No.3, pp. 121-133, 2001.
- Network coding
  - R. Ahlswede, N. Cai, S.-Y. R. Li, and R. W. Yeung. Network Information Flow. *IEEE Transaction on Information Theory*, 46(4): 1204–1216, 2000.
- WSN Testbed
  - J.-P. Sheu, C.-C. Chang, and W.-S. Yang, "A Distributed Wireless Sensor Network Testbed with Energy Consumption Estimation," *International Journal of Ad Hoc and Ubiquitous Computing* (accepted).

http://hscc.cs.nthu.edu.tw/~sheujp/module.php?page=publication