ECE 471 – Embedded Systems
Lecture 24

Vince Weaver

http://www.eece.maine.edu/∼vweaver

vincent.weaver@maine.edu

4 December 2014

Announcements

- Project Update
- Course Evals

Final Exam Review

- Definition of embedded systems; be able to explain why a system is or isn't based on the characteristics given in class
- Hard/Firm/Soft Realtime, know the definitions
- Operating System and Security Related Question
- Know the benefits of code density
- Bus tradeoff question (use cases for i2c/SPI/USB/1-

wire)

- Be sure to review the homeworks and the questions asked in them

- Power/Performance question; see sample HW10 posted to the website

Wireless Sensor Networks

- Low-power, scatter about and forget
- Report stats to central server
- Need to last long time and not use much energy
- Science Fiction element. For example, see *A Deepness in the Sky* by V. Vinge.

Uses for large sensor networks?

- military
- precision agriculture
- civil engineering: traffic / bridge sensing / industrial
- research: biology / environmental

Sensor Types

- *Survey of Hardware Systems for Wireless Sensor Networks*, Hempstead, Lyons, Brooks, and Wei, 2008.

- Very Low Frequency
  Temperature, Atmos Pressure – 0.17 - 1Hz

- Low Frequency
  Heart Rate – 0.8 - 3Hz
  Volcano Infrasound – 20 - 80 Hz
  Seismic – 0.2 - 100Hz

- Mid Frequency
  Earthquake Vibrations – 100 - 160Hz
  ECG (Heart) – 100-250Hz

- High Frequency
  Breathing – 100 - 5kHz
  Industrial – 40kHz
  Audio (human speaking) – 15-44kHz
  Audio (muzzle shock) – 1MHz
  Video (digital television) – 10MHz

Sensor Deployments

- Great Duck Island Maine – looking for rare seabirds
- Industrial – listen for failures in chip fab
- Volcano Monitoring
- Countersniper

Commodity Systems

- Off-the shelf
- if sampling is low and have good sleep behavior, can be competitive

Event Driven

- Handle events in hardware

Application Acceleration

- Hardware assist of detection, lowering energy use (General Purpose computing often wastes energy)

Some Implementations

- Berkeley: Motes, Renee, Mica, Dot
- Rockwell: WINS – ARM 32bit
- MIT: uAMPS, Cricket
- UCLA: iBadge, Medusa – 40MHz ARM
- U-Washington: SpotON
- U Tokyo U**3

- USC: ROBOMOTE
- Berkeley: Smart Dust?

Energy Saving Schemes

- Same as we've seen on larger systems
- DVFS, leakage mitigation, etc, as seen on bigger
- *Improving Energy Efficiency of Personal Sensing Applications with Heterogeneous Multi-Processors* (Heterogeneous)
- Concentration on low-power transmitters

MICA

- MICA: A Wireless Platform for Deeply Embedded Networks, Hill and Culler 2002

- peer-to-peer networking

- microamps of power. Last years on AA or coin cell. 100x less than cellphone

- not depend on outside architecture. Cell phones depend on high-powered base stations with fast connectivity

- network protocol... IEEE 802.11b might only take 25-100mW to transmit but processing the protocol can take up to 2000mW for high data rates
- Bluetooth maybe more suitable, but still take 115mW to communicate with a master node
- cell phone protocols optimize for latency. deep embedded can tradeoff latency and bandwidth. If only sampling every few minutes as is, delaying transfer by a few seconds not as critical
- have highly configurable software radio

- 1.25x2.25 inch, size of two double A batteries
- standby current on level of microamps
- thermal, barometric, magnetic, light, infrared, acceleration, vibration, acoustics
- 8-bit ATMEGA128, 4MHz, 128K flash rom, 4kB static RAM, 48GPIOS, SPI bus, 3 timers
- radio transmitter max outputs 0.75mW, (1/1000 of cell phone) while drawing 21mW. can transmitting up to 115kBPS

- 200 feet range, receive takes 15mW (whether receiving or not)

- 4Mbit flash chip for holding data

- Maxim 1678 DC/DC converter. Constant 3.3V supply. Can boost up from 1.1V, good as 50% of alkaline batteries below 1.2V. Boost converter can be disabled in ultra-low-power mode

- 51 pin expansion bus

Component	Active	Idle
CPU	16.5mW	30uW
Radio	21/15 mW	0
Silicon ID	0.015mW	0
External Flash	45mW	30uW
LED	10mW	0

- pooled operating. Radio packet start requires 3MIPS. Sensor reading 2MIPS. Partitioned system would require provisioning for both (5MIPS). Shared system get by with 3MIPS.

- RAW radio interface allow varied interfaces. Bluetooth or other have no control, can only queue up packets.
- can detect noise and auto-adjust transmit power
- provide some accelerators for the radio, based on existing microcontroller functionality
- low-power wakeup
- time synchronization
- estimate distance based on signal strength

- timed notify, say will turn back on in 5min to update report. master node has to turn on early and wait longer to account for clock drift

Other Implementations

- MICA2 – 7MHz
- TI MSP340 – 16bit, 2mA at 8MHz and 3.0V, few uamps at sleep, 750pJ/instruction
- Smart Dust – 0.25um process (low leakage) 1V and 500kHz, 12pJ/insn

Operating Systems

- Not run Linux
- Why need to run any OS at all? Why not program direct to hardware?

TinyOS

- Initially ran with only 512 bytes of RAM
- special language, nesC
- Not use threads, as threads need stack per process, waste RAM
- Components, state machine.
- Tasks can run to completion, but OS low-level tasks can preempt

Asynchronous Systems

- Necchi, Asynch 2006.
- Asynchronous ATMel compatible
- No clock, so no need to wait for PLL to spin up
- 14pJ/Insn at 1.2V
- 2.7pJ/Insn at 0.51V

WiseNet

- *WiseNET: An Ultralow-Power Wireless Sensor Network Solution* Enz, El-Hoiydi, Decotignie, Peiris, 2004.

- Issues that waste energy:
  - idle listening – waiting when no one is transmitting
  - over-emitting – transmitting and no one listening
  - over hearing – decoding a message not meant for you
  - collision

- Single 1.5V battery

- Lower frequency bands. 2.4GHz band requires more energy 2.4GHz at .18 micrometer process require 1.8V

SNAP/LE

- *An Ultra Low-Power Processor for Sensor Networks*, Ekanayake et. al., 2004.
- MEMS – self powered circuits (vibration), or solar powered, or RF powered
- SNAP/LE
- 16-bit processor, designed for sensor work
- 218pJ/ins at 1.8V and as little as 23pJ/ins at 0.6V

- chart of Energy per instruction