Embedded Internet Systems

Concepts and Applications

Introduction

Java for Embedded Systems

Internet enabled Embedded Devices
Dallas TINI and Imsys SNAP

Simple demonstrations.

Target Applications.

Claro Noda
IMRE-Physics Faculty.
University of Havana.
October, 2004

Introduction

Embedded Systems are ubiquitous

- Toys, telephones, televisions, VCRs, DVD players, stereos. Almost anything that plugs into the wall.
- Cars produced today, more than 80 Microcontrollers, millions of lines of code.
- There is a growing number of cell phones and PDAs
- Although many homes has a PC, almost everyone has a computer embedded into things.

IBM Terminology

- **embedded system**
  A computer system that is a component of a larger machine or system. Embedded systems can respond to events in real time. Hosts of embedded systems include watches, household appliances, cars, and cellular phones.

Introduction
Commodities based on silicon

Similar to PCs Embedded Systems are comprised of:

- Microcontroller
- Memory RAM/EPROM/Flash + I/O ports.
- Interface LCD, Keyboard, serial terminal, e-mail, Web
- Use languages like Assembler, C and **Java**.
- And some runs **multithreaded, preemptive RTOS**.

But differentiate in:

- Small form factor
- Low power consumption
- Broad based I/O
- Limited computing resources.
  Most systems in the market today
  has **2Kb-32Mb** Mem **8-bit/32-bit**
  Architectures operating below **100 MHz**.

Java for Embedded Systems

Embedded Systems meet the net

- TCP/IP is a “heavy” communication protocol
  (ping, telnet, ssh ***but also*** smtp, pop3, ftp, http)
- Bulky code pieces implies high cost to implement from scratch.
- Most small microcontrollers lack a way to connect to the
  network as opposed to PCs that can use Ethernet, USB DSL, modems, etc.

Java to the rescue

- Java superficially resemble C++,
  but Java differs in that it has a special relationship with the Internet.
- Portability simplifies programming (cross platform development)
- There’re new **Native Java Microcontrollers** available
- Lower engineering time ➔ cuts development cost
- Lower time to market/deployment.
- Still difficult to justify 4 big production volumes

Internet enabled Embedded Devices

Dallas Semiconductor **T**iny **I**nternet **N**etwork **I**nterface, TINI

- Based on the DS80C390 (8051 compatible, 40 MHz, 4x8bit I/O)
- 512Kbyte Flash ROM + 1Mbyte NV SRAM

$50 USD Retail

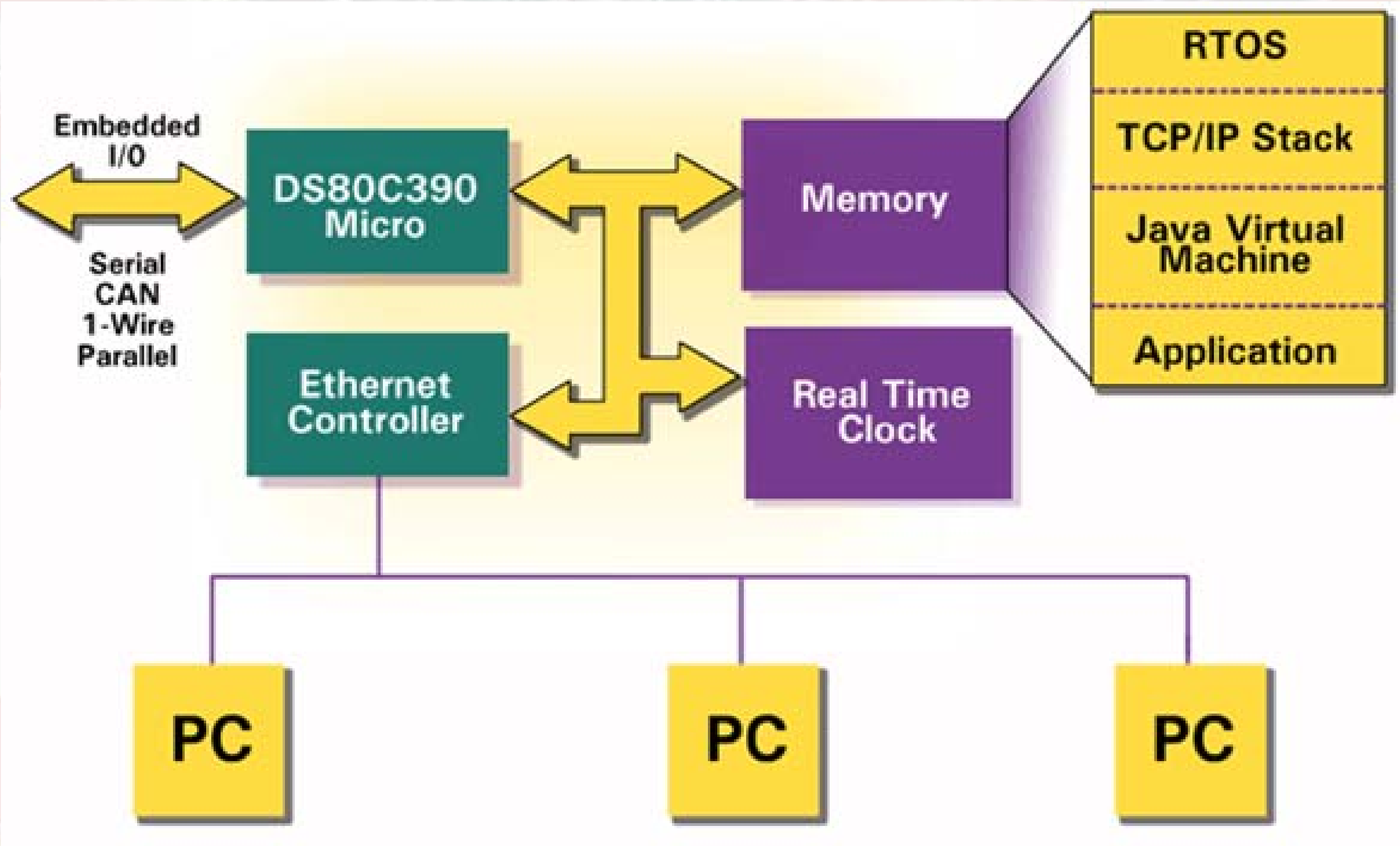
Imsys **S**imple **N**etwork **A**pplication **P**latform, SNAP

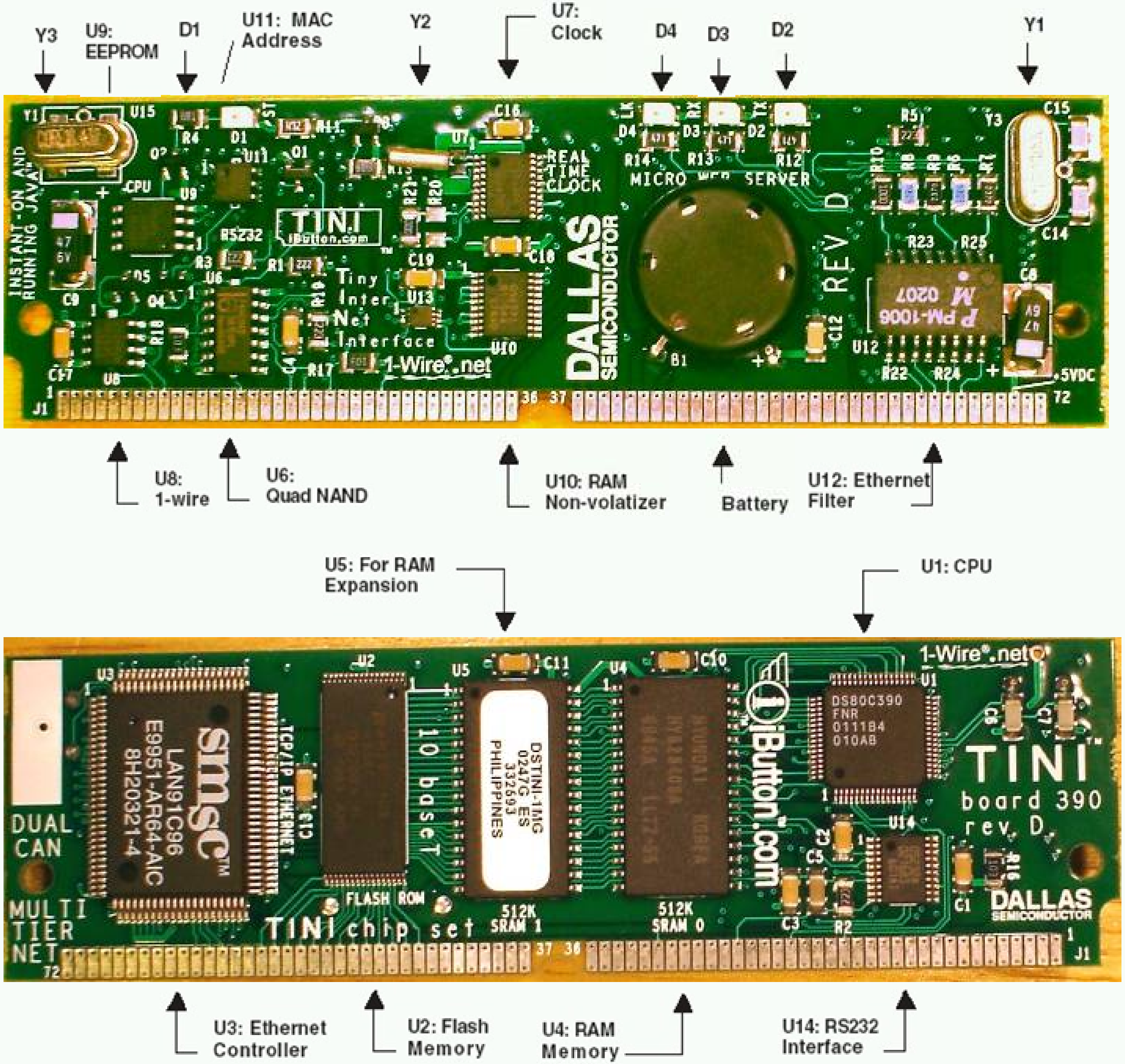
- Based on the Imsys Cjip processor (Native Java, CISC/WISC 66MHz, 24bit I/O)
- **2 Mbyte Flash + 8 Mbyte DRAM**

122 € Retail

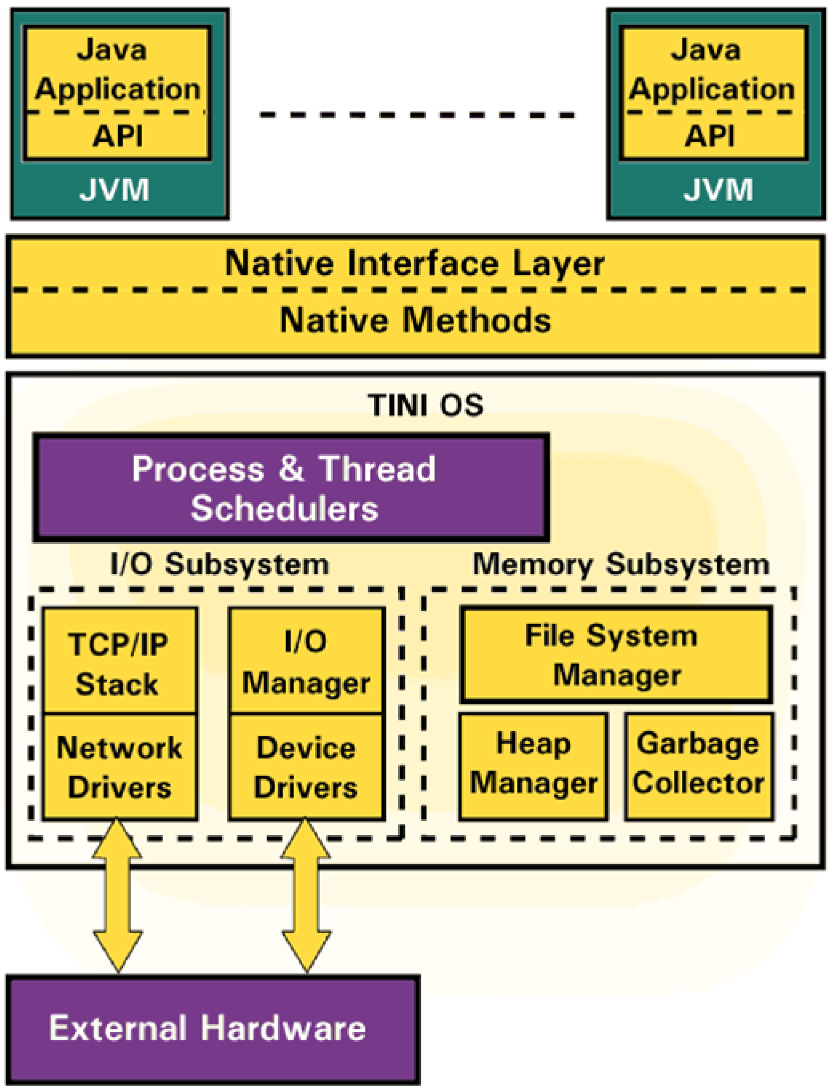
- 72-pin SIMM board (31.8 mm x 102.9 mm)
- reference implementation (design details made public)
- system component (fully specified, heavily tested)
- gives sensors and other devices a voice in the network allowing them to be monitored, controlled, and managed remotely
- on board **CAN**, **1-Wire**, **I^{2}C**, **SPI** and 10/100 Base-T Ethernet

TINI Platform Hardware




Y3
U9: EEPROM
D1
U11: MAC Address
Y2
U7: Clock
D4
D3
D2
Y1
U8: 1-wire
U6: Quad NAND
U10: RAM Non-volatizer
Battery
U12: Ethernet Filter
U5: For RAM Expansion
U1: CPU
U3: Ethernet Controller
U2: Flash Memory
U4: RAM Memory
U14: RS232 Interface
TINI board 390 rev. D
DALLAS SEMICONDUCTOR
1-Wire®.net
TINI chip set
FLASH ROM
512K SRAM 1
512K SRAM 0
DUAL CAN
MULTI TIER NET
REAL TIME CLOCK
MICRO WEB SERVER
REV D
Tiny InterNet Interface
INSTANT-ON AND RUNNING JAVA

Dallas TINI Software



JVM

- small footprint less than 40 Kb
- threads, all primitives and strings
- java.lang, java.io, java.net and java.util
- specific classes com.dalsemi.*

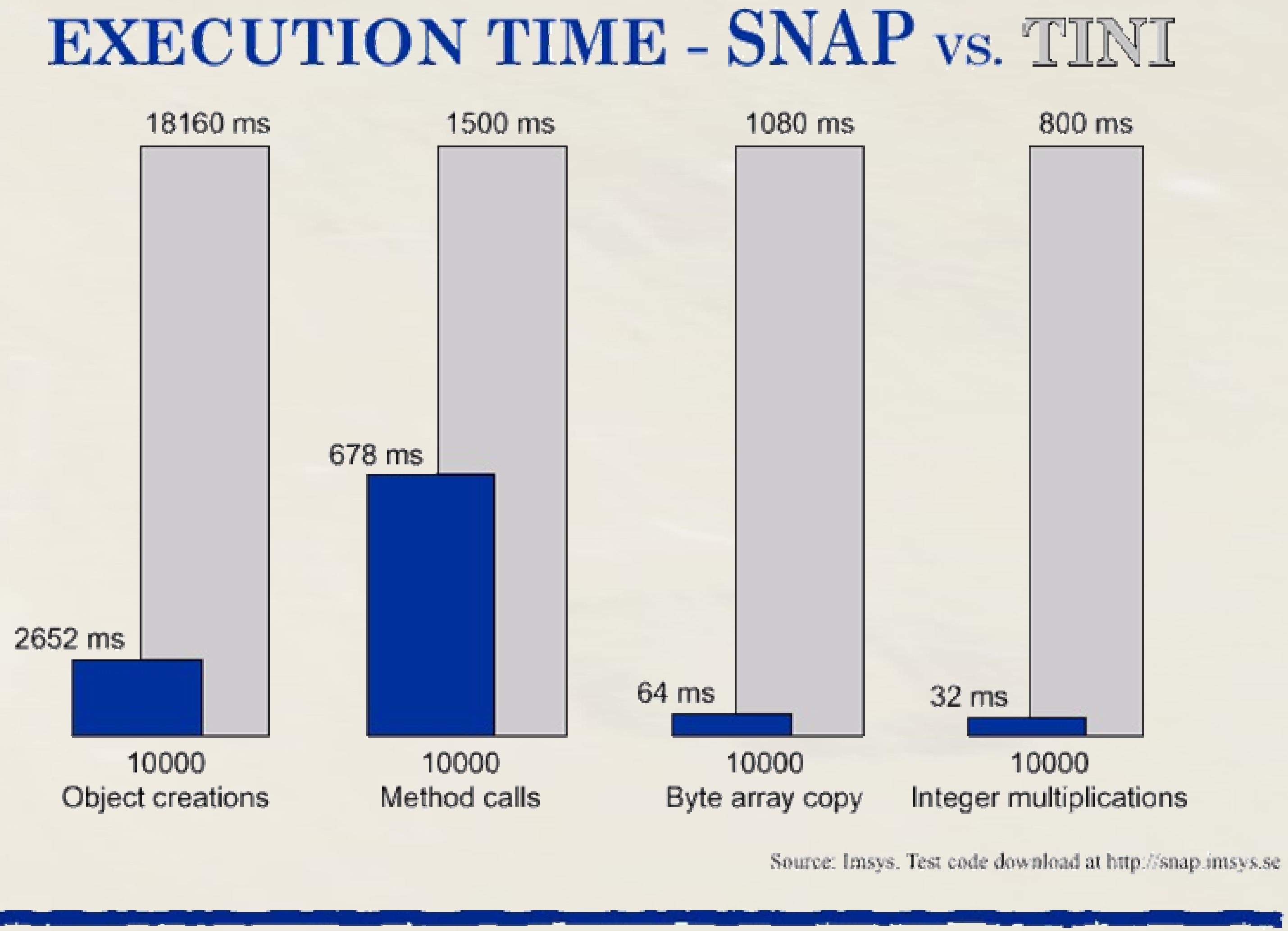
TINI OS

- task scheduling, a file system, memory and I/O managers
- optimized to switch between multiple executing instances of a Java bytecode interpreter
- 8-ms time slices, round robin scheduler

TINI SDK

- includes the JRE + Mics tools
- Sun JDK for cross platform dev

Native Java Execution



Web Server Demo

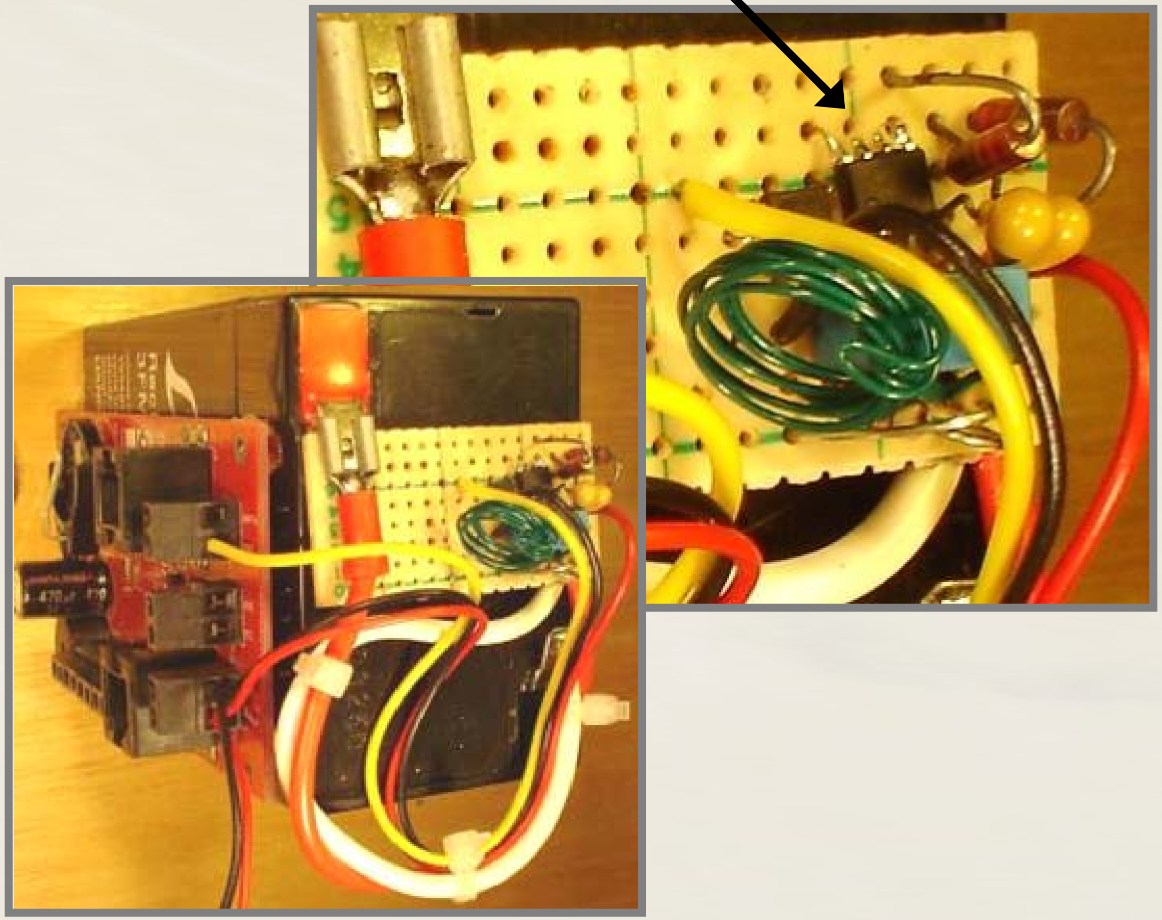
TINIWebserver

- simple multithreaded web server listen to port 80
- demonstrates the use of the 1-Wire net for remote monitoring
- reports time, date and current temperature
- logs every access attempt to web.log in TINI file system

Battery Management Demo

The 1-Wire net

• DS2438 Smart Battery Monitor



• Lead-Acid battery 6V 4Ah
• TINI runs at 150 mA
• Batt-pack lasts approx 20h
• 5% Fuel-Gauging accuracy
• Monitors voltage, current and temperature
• Uniquely addressable ID for bus sharing
• 40 Byte user EEPROM for pack-specific data

Target Applications

Energy Management

Generation and Demand Side Management. Load balancing, efficiency and peak hours.

Meteorology

Distributed weather stations would provide a stream of real-time surface data to feed hurricanes forecast models on-the-fly.

Habitat Monitoring in Social Insects

Dynamics. Experiments and modeling.

Energy Management

Motivation

- power grid -- the internet -- quality of power
  two networks that overlap
- nature of power consumption, stochastic or predictable?
- system limitations, self-adjustments (via power failure)

problem

- thermoelectric efficiency, operational point & inertia
- peak hours generation & quality of electricity

a smart workaround

- distributed intelligence
- utilities load percentage been published
- DSM an approach to smooth the peak → system become more stable

Weather Stations

TINI + 1-wire weather station: the "brute force" approach

Wide-spread weather stations concurrently populate the “weather database” over the network in real time.

Interest

- Surface data feeds hurricane models on-the-fly
  the higher the accuracy of data fed into the model the closer the forecast

- micro-weather remote monitoring
  fine-tuned (sustainable?) agriculture management

- geo catching/mapping automation

Habitat monitoring in Social Insects

Optimization algorithms
• social insects show self organization (SO)
• swarm-intelligence systems are hard to “program”
  paths to problem solving aren’t specified but *emergent*
• study SO in natural system → modeling behavior → use model for artificial device

Foraging activity models
• random search for foods
• communication through dancing (bees)
• negative feedback helps stabilization
• SO relies in amp of fluctuations
  randomness key to new solutions
  seed for nucleation and grows
  structures emerge despite randomness

Ants dynamics

Ants nest experiments
- nest occupies a surface area up to 600 m^2
- time correlation of activity between different entries
- excellent context for TINI
- new tools → might new info come out?

Experimental setup
- activity sensor on every entry
- enviromental data
- TINI collects and forward data to servers

Interpretation
- Fc [pi, pj] (t = 0) = 1 → enviromental role
- Fc [pi,pj] (t = τ) = 1 → certain kind of SO

Next step would be modeling

Ants activity sensor

Ideas for a prototype

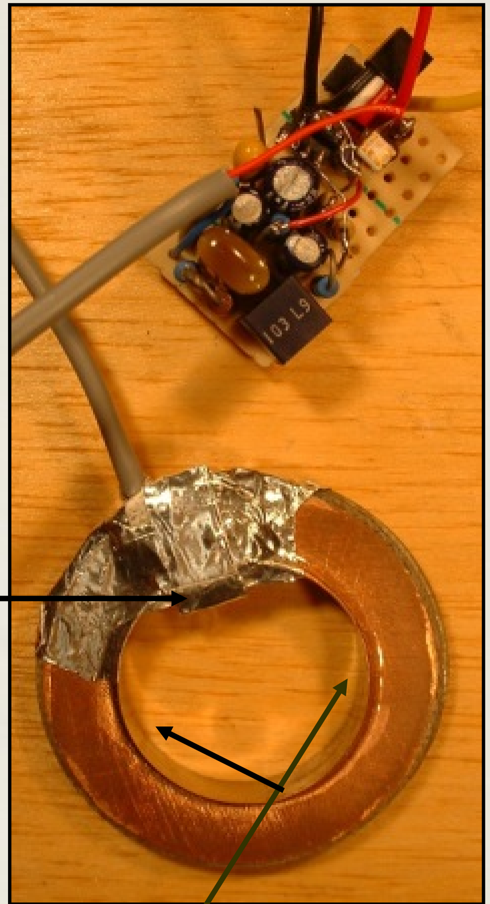
- detect activity at the nest entry
- low power consumption
- low count drift
- robustness for field use
- low maintenance

our proposal

- an infrared approach (non-intrusive?)
- 1-wire interface for scalability
- 25 KHz synchronic demodulation for reliability

additional data

- daylight intensity
- temperature
- humidity



It's more than fun . . .

The exciting part is that this technology is nascent - we are starting to see what it means the luxury of having inexpensive ubiquitous connectivity between embedded systems, PCs and humans.

The bottom line:

There is plenty of room for innovation.

Great opportunity for innovation

Are we up to the challenge?

Application development:

- Application specific knowledge
- Hardware
  Digital & Analog Electronics
- Sensor design science
- Java Programming, Algorithms, Data Structures
- Web Technologies
  low and high level protocols