More than just a few chips...

Building Wireless Sensor Networks in 2005

Jan Beutel
Outline

- Our device: BTnode rev3
- Recent developments @ TIK
  - BTnut System Software & Embedded Development
  - Deployment-Support Network
  - TinyOS on the BTnode rev3
- Experiences
New BTnode rev3: a lightweight dual radio platform

BTnode success story
- Running both, TinyOS and the BTnut system software
- Prototyping Wireless Sensor Networks with BTnodes [EWSN2004]
- 3rd generation node commercialized with industrial partners AoT and Iftest
- Open-source policy has led to commercial replicas (Cobalt Blue by Vitronics)
Poor WSN development reality

It is very hard to deploy anywhere beyond 10-20 nodes.

Coordinated methods and tools are missing today.
BTnut Software & Embedded Development

BTnodes are not targeted at ultra low-power...

... but target versatile and flexible fast-prototyping.

Multi-threaded OS frame in C
  - Standard open-source tools
  - Lightweight software distribution (8.2 MB binary, 27.3 MB source)

Rapid prototyping
  - HW emulation on Linux PC

Demo applications and tutorial
  - Lab for embedded systems lecture
Application: Deployment-Support Networks

Challenge
Development and Deployment of Sensor Networks.

Solution
Self-organizing backbone network with deployment-support services:
- Virtual connections to nodes
- Remote reprogramming, debugging, monitoring

[SenSys2004], [IPSN2005]
Application: JAWS Prototype

host controller

DSN nodes

targets

virtual connections
Application: JAWS Experiments

70+ node experiments

Main obstacles

- Software complexity and stability
- Predictable operation and performance guarantees

Ongoing work

- JAWS permanent demo setup
- XTC routing implementation [Wattenhofer2001]
- Distributed time synchronization [Blum2004]
- Online runtime analysis
- BTnode power profiling [Negri – exchange PhD]
Related Work – Sensor Network Deployment

Simulation
- TOSSIM [Levis2003 – Berkeley]
- PowerTOSSIM [Shnayder2004 – Harvard]

Testbeds
- Gnomes [Welsh2003 – Harvard]
- MoteLab [Wernerallen2005 – Harvard]
- Kansai – eXtreme Scaling Mote [OhioState]
- Mirage [Chun2005 – Intel]

Embedded emulation
- EmStar [Girod2004 – UCLA]
- BEE [Chang2003, Kuusilinna2003 – Berkeley]

Software distribution
- Deluge [Hui2004 - Berkeley]
TinyOS on the BTnode rev3

**Basics are working:** BTnode rev3 is a Mica2 replacement

- Cooperation with Uni Copenhagen/ P. Bonnet
- BTnode3 platform definition available in `contrib/tinybt`
- Hard to “just port” without an application requirement because of hardware dependencies in the software
- `tinyos-1.x` – 162 MB CVS nightmare
- nesC makes debugging harder, complexity is hidden within
Experiences 1 – where we stand today

Heterogeneous and large design space

- No formal definition of a sensor network
- No single hardware/software platform for all/most needs
- Minimum required resource set for correct, reliable and predictable function is unclear

Underestimated complexity

- Specialized chips are highly complex
- Applications need many support functions, software systems grow

Keep it simple with the right level of abstraction

- Different people speak different languages
- Technology is only accessible to highly trained personnel, e.g. not to an application-domain expert
Experiences 2 – beyond the proof-of-concept

Realistic field testing is key to functional validation

- Real-world interaction is necessary
- Sporadic race conditions do not show on a lab bench
- Standard embedded and real-time system issues are dominant
- In-situ testing, monitoring, calibration, validation and verification have to be designed in

Emerging (open) standards?

- nesC 1.1 -> 1.2 major revision with new features will break compatibility
- TinyOS 2.x will be completely new, looks promising, currently a lot of “back to the roots” discussion in the 2.0 WG – component based – platform design – compatibility – reuse – many new platforms
To probe further...

http://www.btnode.ethz.ch
Backup: BTnode rev3 architecture details

System core
- Atmel ATmega128, AVR RISC @ 8 MIPS, 128 kB Flash, 64 kB SRAM, 180 kB data cache, 4 kB EEPROM
- 4 LEDs, reset button
- External DC supply or 2 AA cells with on/off switch
- Generic sensor interfaces

Bluetooth radio
- Zeevo ZV4002, supporting AFH/SFH Scatternets with max. 4 Piconets/7 Slaves, BT v1.2 compatible

Low-power radio
- Chipcon CC1000 operating in ISM Band 433-915 MHz
Backup: BTnode rev3 additional information

Power Budget on typical AA cells (2x1.2V, 2500 mAh)

- 0.5 – 150 mA current consumption
- ~80% efficiency in up-conversion to 3.3V
- Non power-aware apps can last a few days on batteries
- Power optimized apps can last many weeks depending on duty cycle

<table>
<thead>
<tr>
<th></th>
<th>I_{out}@3.3V</th>
<th>I_{in}@2.4V</th>
<th>I_{in}@2.2V</th>
<th>I_{in}@2.1V</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.0</td>
<td>16.5</td>
<td>18.0</td>
<td>18.9</td>
<td></td>
</tr>
<tr>
<td>50.0</td>
<td>82.5</td>
<td>90.0</td>
<td>94.3</td>
<td></td>
</tr>
<tr>
<td>100.0</td>
<td>165.0</td>
<td>180.0</td>
<td>188.6</td>
<td></td>
</tr>
</tbody>
</table>

Commercialization

- **Availability** samples now, volume Q1/2005
- **Pricing** USD 215/EUR165/CHF255 for samples, larger quantities upon request
- **Contract Manufacturer** Art of Technology, Zurich, Switzerland
Developing with Eclipse and AVR Simulator