BTnodes

Prototyping Wireless Sensor Network Applications

Jan Beutel, Oliver Kasten, Friedemann Mattern, Kay Römer, Frank Siegemund, Lothar Thiele
Overview

Prototyping Wireless Sensor Network applications on real devices in a heterogeneous environment

BTnode – Ad hoc networking prototyping platform

Event driven OS/application integration

System services for heterogeneous sensor environments

Example application
Prototyping Wireless Sensor Networks

A myriad of interacting devices
- sensor node heterogeneity
- sensing and actuation
- user interaction

Smart everyday objects
by attaching sensor nodes:
- self aware
- context sensitive
- cooperative
- integration into computing environment

NCCR-MICS Terminodes

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

January 21, 2004
Backend connectivity

Infrastructure integration

Local applications

Mobile network gateways
– GSM
– SMS
– Wireless LAN
– interfacing to other sensor networks

Debugging and monitoring
– development
– quantitative analysis

Internet
Local Application
BTnodes
A look at platform competitors

**UC Berkeley motes** (Culler et al.)
- Motes and IPAQ’s (Estrin et al.)
- Mote-on-a-chip “spec” (Hill)

**Bluetooth devices**
- Intel research motes (Kling)
- Blue Wand (Zitterbart)

**Testbeds**
- Microsoft wake-on wireless (Bahl)
- UCLA i-Badge (Srivastava)
- UCLA/WINS/Rockwell nodes (Srivastava)
- PicoRadio Testbed (Rabaey)

**Other**
- TeCo Smart It’s (Beigl)
- PicoBeacon (Rabaey)
- WiseNet (Enz)
- ParkTab (Weiser)
- Active Bat (Hopper et al.)

- full custom radio
- proprietary
- limited availability
- complex and low integration
- low power
- reduced features
- full custom
BTnode architecture

Lightweight wireless communication and computing platform based on a Bluetooth radio module and a microcontroller.

Bluetooth has the advantage of
- availability today for experimentation
- compatibility to interface to consumer appliances
- an abstract, standardized high level digital interface
**BTnode hardware details**

**Integrated hardware features**
- 8-Bit RISC, max. 8 MIPS
- 128 kB Flash
- 64 kB SRAM, 180 kB data cache
- operating from 3 cell batteries
- integrated antenna
- power management
- generic sensor I/O

<table>
<thead>
<tr>
<th>Current bill of material</th>
<th>50 parts</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parts</td>
<td>60 USD</td>
</tr>
<tr>
<td>Assembly</td>
<td>5 USD</td>
</tr>
<tr>
<td>Bluetooth</td>
<td>45 USD</td>
</tr>
<tr>
<td><strong>Unit cost @ 200 units</strong></td>
<td>110 USD</td>
</tr>
</tbody>
</table>

January 21, 2004

Slide 7
BTnode System Software

Lightweight OS
- event-driven application model
- cooperative multithreading
- device drivers (UART, RTC, ADC, ...)
- static memory allocation
- minimum memory footprint

Programming
- standard C language
- high-level Bluetooth interface
- system software available as library
- emulation environment on Linux
Event driven OS/application integration

Approach common to embedded systems

Geared towards processing of external events
– sensor values, data packets, state changes
– only one handler active at a time

One application per system at a time
– application resident in device
– no dynamic process model
– events triggered by OS/driver functions and applications

No real-time OS knowledge necessary for application design
Event handling in the dispatcher

```c
void handler( /* ... */ ) {}  
void main() {  
  btn_disp_ev_reg( RECEIVE_EV, handler, 0 );  
  btn_disp_run();  
}
```

**Diagram:**
- **Application:** `btn_disp_run()`, `read()`, `handler()`, `read_buffer`
- **Driver:** `received_irq()`, `read_buffer`
- **Hardware:** `received_irq()`
- **Dispatcher:** `btn_disp_run()`, `event queue`, `EVENT, handler func`, `RECEIVE_EV, handler()`, `OTHER_EV, func2()`
Rapid prototyping – Linux emulation

Native compilation and execution on Linux
– using adapted drivers to match the host system
– with a serial Bluetooth device on a PC or iPAQ

Bluetooth PC
+ use unlimited resources of host
+ bridging networks
+ comfortable application debugging

BTnode
+ deployment platform
− slow upload necessary
System services – Network access

Mobile infrastructure integration
- Bluetooth enabled mobile phones or PDAs as access points
- GSM, SMS or WLAN access
- session/connection oriented access
- asynchronous access
Collaborative processing of sensor data
- local cooperation to solve complex tasks
- data replicated in distributed tuple space

Node operations
- sensor data written into tuples
- data fusion from multiple tuples
- results are written back into tuple space

Related work
- SensorWare [Boulis2003]
- DSWare [Li2003]
- TinyDB [Madden2002]
System services – Outsourcing computation

Nodes have limited computing capabilities

Outsourcing computation on nearby mobile devices
- BTnodes transport and store precompiled Java classes
- tuple space data and events are replicated
- *Smoblets* are executed on mobile devices (iPAQ, GSM, Laptop…)

![Diagram showing the interaction between sensor nodes and handheld devices via Smoblets and Bluetooth technology](image-url)
Bluetooth enabled appliances

Communication with other Bluetooth enabled devices
– standard Bluetooth profiles for SMS, object push and RFCOMM

BTnode enabled egg carton
SMS from egg carton
Interactive dialog
Power consumption example

Good energy-per-bit ratio due to high throughput [Leopold2003]
Sensor network example: 10% duty cycle

<table>
<thead>
<tr>
<th>Operation</th>
<th>Power consumption [mW]</th>
<th>Lifetime [h]*</th>
</tr>
</thead>
<tbody>
<tr>
<td>4 sec sensing</td>
<td>12</td>
<td>252</td>
</tr>
<tr>
<td>2 sec communication</td>
<td>160</td>
<td>19</td>
</tr>
<tr>
<td>54 sec idle</td>
<td>0.5</td>
<td>6048</td>
</tr>
<tr>
<td>Total duty cycle</td>
<td>~ 6.5 mW</td>
<td>421</td>
</tr>
</tbody>
</table>

*on 840 mAh Li-ion
BTnodes - A Distributed Environment for Prototyping Ad Hoc Networks

Welcome to the BTnode Platform!

Overview

The BTnode is an autonomous wireless communication and computing platform based on a Bluetooth radio and a microcontroller. It serves as a demonstration platform for research in mobile and ad hoc connected networks (MANETS) and distributed sensor networks. The BTnode has been jointly developed by the Computer Engineering and Networks Laboratory (TIK) and the Research Group for Distributed Systems at ETH Zurich. Currently the BTnode is primarily used in two major research projects: NCCR-MICS and SmartBits.

BTnode features at a glance

- Microcontroller: Atmel ATmega 128L (8 MHz @ 8 MIPS)
- Memories: 64 kbyte RAM, 128 kbyte FLASH ROM, 4 kbyte EEPROM
- Bluetooth radio module: Ericsson ROK 101 007
- External Interfaces (ISP, UART, SPI, I2C, GPIO, ADC,...)
- 4 LEDs
- Standard C Programming

Quickstart

To get going is quite straightforward. Before you can start off developing applications for the BTnode you need to:

- Download and install the development tools (compilers). See the tools section.
- Download and install the BTnode System Software. See Installing the BTnode System Software.
- Buy a hardware programmer. We recommend the Atmel STK 500 programmer.
- Build your own programming cable (one per programmer). See Hardware.
- Get BTnodes or serial Bluetooth devices that can be used in emulation mode.
- Compile and download your first example application.

http://www.btnode.ethz.ch