BTnodes

Directions in Hard- and Software Architecture

Jan Beutel, Oliver Kasten, Matthias Ringwald
Initial Projects

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

**Ad hoc networking scenarios**
- integrated application protocols
- scalable multi-hop routing
Bluetooth Piconets

Communication organized in piconets
- controlled by one master
- up to 7 active slaves
- 255 inactive (parked) slaves

Master-Slave
- implements centralized control
- synchronization of all slaves
- only master-slave communication

Multiple piconets
- separate channels
- no coordination
Bluetooth v1.1 Scatternets

- Using gateway nodes
- No coordination
- Role switch necessary

Problems
- synching to two FH piconets
- connected Slaves are invisible
- limited flexibility
- Slave/Slave not possible

- Networking has not yet been defined and is up to the application
Future Bluetooth v1.2 Scatternets

- Draft v1.2 in final voting phase
- Participating in different Piconets on TDMA base
- Supporting best effort traffic

Problems
  - distributed scheduling not fully defined
  - networking not included
  - hardware not available
Bluetooth Host Controller Interface

- standard interface for protocol software
- providing access to lower levels of the protocol stack

HCI_COMMAND

HCI_EVENT

OGF  OCF  PARAMETERS

Applications

SDP  RFCOMM  ...

L2CAP

Host Controller Interface

Audio  Link Manager  Baseband  RF  RF

Host processor

Physical interface

Bluetooth module
Bluetooth Connections

Managed by the host controller

Statemachine for each connection

Link Layer Control & Adaptation (L2CAP)
- connection-oriented
- connectionless data
- protocol multiplexing for a single “air interface”
- packet segmentation and reassembly
- channel abstraction
- encryption
- Security
- …
Hardware Requirements

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

Requirements
- small form factor, low component count
- standardized wireless interface
- flexible and cost effective deployment of large quantities of networking nodes
Hardware Details

- **Integrated PIFA Antenna**
- **Communication**
  - Ericsson Bluetooth Module
- **LEDs, Reset, Clocks**
- **Memory**
  - 128 kB Flash
  - 244 kB SRAM
  - 4 kB EEPROM
- **CPU**
  - Atmel ATmega 128L MCU
  - 8-Bit RISC (max. 8 MHz ~8MIPS)
- **Power Management**

Generic Sensor Interfaces

UART and I2C Data Interfaces

61 mm

40 mm

61 mm
Designing for Power Aware Operation

Features

- optional switchable power supply for Bluetooth module
- MCU with 6 power down modes
- frequency scaling: 7.3 MHz - 57 kHz
- low idle/sleep current

- single power supply (3.6 – 16 V), single internal voltage (3.3 V)
- battery charge indicator
- direct current access shunts for all components
- internal Vcc available at every connector to power external sensor modules

Power consumption @ 3.3V, 7.3 MHz [mW]

<table>
<thead>
<tr>
<th></th>
<th>max.</th>
<th>typ.</th>
<th>Lifetime [h]*</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bluetooth Connected/CPU On</td>
<td>250</td>
<td>160</td>
<td>12-19</td>
</tr>
<tr>
<td>Bluetooth Idle/CPU On</td>
<td>95</td>
<td>67</td>
<td>32-45</td>
</tr>
<tr>
<td>Bluetooth Off/CPU Idle</td>
<td>15</td>
<td>12</td>
<td>202-252</td>
</tr>
<tr>
<td>Bluetooth Off/CPU Sleep</td>
<td>6</td>
<td>&lt;0.5</td>
<td>504-6048</td>
</tr>
</tbody>
</table>

*on 840 mAh Li-ion
Power Consumption Details

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>
</tbody>
</table>

Total duty cycle ~ 6.5 mW 421

*on 840 mAh Li-ion
System Software

Lightweight OS
- event-driven application model
- cooperative multithreading
- device drivers (UART, RTC, ADC, ...)

Programming
- standard C language
- high-level Bluetooth interface
- system software available as library
- emulation environment on Linux
void handler( /* ... */ ) {}
void main() {
    btn_disp_ev_reg( RECEIVE_EV, handler, 0 );
    btn Disp_run();
}
Getting going on BTnodes …

Documentation available on the Web
– design documents
– API documentation
– mailing list

Software development tools
– Gnu toolchain
– avr-gcc, avr-libc, downloader, gdb
– available for Linux, Windows and Mac OS X

BTnode System Software
– snapshot packages and CVS repository
– Linux emulation on x86 and iPAQ
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
Better Avalanche Rescue through Sensors

- Oximeter
- Oxygen sensor
- Accelerometer, inclination
- Heart rate, oxygen saturation
- Air pocket detection
- Orientation, mobility
Physical Activity Detection Network

Idea: Many sensors reveal “more context“
- Use multiple motion sensors for context awareness
- Architecture required to combine those sensors
- Map hierarchical topology to human body
WURM - Wearable Unit with RC Modules

WURM hardware architecture

CPU for:
- legacy C-code, binary only code
- low-intensity, background tasks

Reconfigurable unit for:
- high-performance tasks
- low-power tasks

Case study: Bluetooth/Ethernet-Bridge

- IP access point for WURM modules using Btnodes
- Soft CPU (LEON, 32bit SPARC)
  - HW minimal TCP/IP stack
  - HW Ethernet MAC
XHOP/R-DSR Multihop Network

Bluetooth multihop source routing prototype
- integrated scalable application protocol
- based on Dynamic Source Routing (CMU)
- routing across piconet borders to support >8 nodes

Remote topology discovery
- script like command language in the payload

Performance
- 1-2 sec per hop, depending on inquiries
Over-the-air Software Updates

Bootloader support of the MCU
- new firmware in XHOP payload
- Flash reprogramming from pre-cached data
- reprogramming of single node and by network flooding

Per-hop performance
- transmitting 10 kB: ~0.8 sec
- writing 10kB to Flash: ~0.2 sec

Overall performance
- ~10s to transmit, write and reboot with 80 kB firmware
Other Projects using BTnodes

200 units with 16 research groups

- smart objects
- routing
- wearable computing
- perceptual computing
- operating systems

Bill of material

<table>
<thead>
<tr>
<th>Item</th>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parts</td>
<td>90 CHF</td>
</tr>
<tr>
<td>Assembly</td>
<td>15 CHF</td>
</tr>
<tr>
<td>Bluetooth</td>
<td>65 CHF</td>
</tr>
</tbody>
</table>

Unit cost @ 200 units: 170 CHF

VTT, FI
DSG, ETH Zurich, CH
PCCV, ETH Zurich, CH
TecO, University of Karlsruhe, GE
PLAY, Interactive Institute, SE
TIK, ETH Zurich, CH
IFE Wearable Lab, ETH Zurich, CH
NTT DoCoMo, Munich, GE
Ptolemy Group, UC Berkeley, USA
Art of Technology, Zurich, CH
DistLab, Diku, Copenhagen, DK
LAP, EPF Lausanne, CH
CS Department, Lancaster University, UK
LSL, EPF Lausanne, CH
TinyOS Group, UC Berkeley, USA
University of Uppsala, SE

May 27, 2003
Slide 21
The road to rev3

- Current Bluetooth is an End-of-Life product
  - Evaluate other BT solution
  - BT with embedded ARM core???

- Extension with **multiple radio frontends** (BT, WiseNet, Motes…)

- **Shrinking the design** by advanced packaging technology
  - next generation packaging saves 30-50% on component area
  - I/O and antenna become the size constraints
  - better battery integration and mounting holes

- Power – drop Vcc to 3.0 and beyond
- Implement **adaptive power/frequency control** in software

- Dispatcher needs rethinking
- Remote **analysis, debugging and management functions**

- Rewrite **system software** for stability, especially Bluetooth portions