Hardware/Software Platforms for Heterogeneous Wireless Sensor Networks

Ack: Slides from Xbow, UCLA, OSU Xscale Project, UCB, Intel Presentations
**Heterogeneous Sensor Networks**

- Explore hierarchical organization of a network of heterogeneous nodes
- Dense deployment of resource constrained micro-nodes (Mote)
- Sparse deployment of powerful macro-nodes (StarGate)
Stargate

- A single board, wireless computing platform
- Developed by the Ubiquity Computing group @ Intel Research licensed to Xbow

- Runs Linux 2.4.19
System Architecture

Processor Board
- MICA2/GPIO Connector
- Flash 32 MB
- CF
- PCMCIA

Daughter Card
- WDT
- LED
- Gas Gauge
- PXA255
- I2C Connector
- SDRAM 64 MB
- JTAG Connector
- MAX 3244
- Ethernet
- USB Master
- RS-232 (DB-9)
- RJ 45
CPU

- **PXA255 Xscale Processor**
  - Successor to the StrongARM family
  - Variable clock (100 - 400 MHz), less than 500 mW power
  - Several sleep modes, rich set of peripherals
Communication

- PCMCIA or Compact Flash slot available for 802.11 radios
- Wireless Software Support
  - Wireless Extensions API
  - HostAP driver
Stargate vs Mote

• Good at bulk computation
  – CRC32 on 1024 byte message
    • Stargate: 24.8 us, 12.89 uJ
    • Mica2: 5.15 ms, 367 uJ
    • Stargate is 200x faster and uses 28x less energy
  – 48-tap FIR filter on 256 samples of 32-bit data
    • Stargate: 325 us, 166.9 uJ
    • Mica2: 168 ms, 11.8 mJ
    • Stargate is 500x faster and uses 70x less energy
  – 128-point FFT on 16 bit samples
    • Stargate: 94.5 us, 45.8 uJ
    • Mica2: 14.5 ms, 934 uJ
    • Stargate is 150x faster and uses 20x less energy

• Mediocre Power Cycling Performance
  – Slow to resume operation
    • Suspend enter: 177 us, 63 uJ
    • Suspend resume: 10.6 ms, 4.17 mJ
  – High power leakage during sleep
Stargate Sensing

- Mote Sensor Access through Serial Port
- Webcam (Logitech Pro 4000)
- EE 682 ?
Stargate Development

- GNU cross development tools for Linux PC host
- Platform: Cross compilers for ARM platform
- Linux Kernel 2.4.19 based on familiar distribution
- Develop in C, or Java (open_wonka JVM)
- Emstar libraries for Sensor Networks
  http://cvs.cens.ucla.edu/emstar/
Stargate Ports

6 channels, 8 KHz, 10 bits - 480 Kb/sec

- RS 232 115.2 Kb/sec
- I2C 100 Kb/sec
- USB 12 Mb/sec
- JTAG
- PCMCIA
- PXA Serial Ports

Buffer or Stream?
# Mote HW Platforms

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Microcontroller</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Type</td>
<td>AT90LS8535</td>
<td>ATmega163</td>
<td>ATmega128</td>
<td>TI MSP430</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Program memory (KB)</td>
<td>8</td>
<td>16</td>
<td>128</td>
<td>60</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RAM (KB)</td>
<td>0.5</td>
<td>1</td>
<td>4</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Active Power (mW)</td>
<td>15</td>
<td>15</td>
<td>8</td>
<td>33</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sleep Power (µW)</td>
<td>45</td>
<td>45</td>
<td>75</td>
<td>75</td>
<td>6</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Wakeup Time (µs)</td>
<td>1000</td>
<td>36</td>
<td>180</td>
<td>180</td>
<td>6</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Nonvolatile storage</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Chip</td>
<td>24LC256</td>
<td>AT45DB041B</td>
<td>ST M24M01S</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Connection type</td>
<td>I²C</td>
<td>SPI</td>
<td>I²C</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Size (KB)</td>
<td>32</td>
<td>512</td>
<td>128</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Communication</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Radio</td>
<td>TR1000</td>
<td>TR1000</td>
<td>CC1000</td>
<td>CC2420</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Data rate (kbps)</td>
<td>10</td>
<td>40</td>
<td>38.4</td>
<td>250</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Modulation type</td>
<td>OOK</td>
<td>ASK</td>
<td>FSK</td>
<td>O-QPSK</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Receive Power (mW)</td>
<td>9</td>
<td>12</td>
<td>29</td>
<td>38</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Transmit Power at 0dBm (mW)</td>
<td>36</td>
<td>36</td>
<td>42</td>
<td>35</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Power Consumption</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Minimum Operation (V)</td>
<td>2.7</td>
<td>2.7</td>
<td>2.7</td>
<td>1.8</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total Active Power (mW)</td>
<td>24</td>
<td>27</td>
<td>44</td>
<td>89</td>
<td>41</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Programming and Sensor Interface</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Expansion</td>
<td>none</td>
<td>51-pin</td>
<td>51-pin</td>
<td>none</td>
<td>51-pin</td>
<td>19-pin</td>
<td>51-pin</td>
<td>10-pin</td>
</tr>
<tr>
<td>Communication</td>
<td>IEEE 1284 (programming) and RS232 (requires additional hardware)</td>
<td>USB</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Integrated Sensors</td>
<td>no</td>
<td>no</td>
<td>yes</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>yes</td>
<td></td>
</tr>
</tbody>
</table>
Extreme Scaling Mote

- Mica 2 with Integrated Sensor Board
- Xbow and OSU design
- Four 90deg PIR sensors
- Microphone with Analog BP Filtering
- 2-Axis Magnetometer
- Photosensor
- Buzzer
Challenges for Scalable Sensor Networks

- Highly Constrained resources
  - processing, storage, bandwidth, battery

- High Level of Concurrency is required, no hardware support for parallel execution
  - Streams of sensor data and network traffic

- Diverse set of Applications / Platforms
  - Modular and Reusable Software

- Robust
  - Cannot touch each node
TinyOS solution

- Event-driven architecture
- Support Concurrency: Two-level scheduling Events Tasks
- Modularity: application = scheduler + graph of components
- Efficiency: Get done quickly and sleep
- Reuse: Narrow Interfaces (Event/command)
- No kernel/application boundary
Components are wired together to build the application, need to supply module implementations (code) & interface definitions and configuration (wiring)

Applications are layers of components where higher level components issue commands to lower level components and lower level components signal events to higher level components
Two Level Scheduling

- **Events**
  - Time Critical
  - Interrupts cause Events (timer, ADC)
  - Short Execution

- **Tasks**
  - Time Flexible
  - Run Sequentially by TinyOS Scheduler
  - Interruptable
TinyOS components

- Component:
  - Interface
    - Commands,
    - Event Handlers
  - Frame (storage)
  - Tasks (threads of comp)

- Constrained Storage Model
  - frame per component, shared stack, no heap

- HW like interface
  - No global variables btw components
  - HPL and Logical HW components