Simulating the Power Consumption of Large-Scale Sensor Network Applications

Victor Shnayder, Mark Hempstead, Bor-rong Chen, Geoff Werner Allen, and Matt Welsh

Harvard University

shnayder@eecs.harvard.edu

http://www.eecs.harvard.edu/~shnayder/ptossim/
Power Management is Important

Sensors often deployed in hard to reach locations
Replacing batteries is difficult and expensive

Need to simulate energy usage before deployment
Our Contribution: PowerTOSSIM

Simulates the energy consumption of each node in a sensor network

- Easy to use
- Scalable
- Accurate
- Integrated with TinyOS

Measured power model of Mica2 mote
Outline

Background
  • Approaches to power simulation
  • TinyOS, TOSSIM architecture
Measurements
  • Mica2 energy model
Implementation
Evaluation
  • Does it work? (yes)
Approaches to Power Simulation

Count high level events
- Number of radio messages most common
  - Pro: Very fast, easy, simple
  - Con: Can be very inaccurate

Simulate at the bit/cycle level
- Keep track of exactly what's happening in system
  - Pro: Extremely accurate
  - Con: Extremely slow-impractical for large scale simulation

We want to be somewhere in between
- \textit{Get best of both worlds}
TOSSIM Architecture

- Component based
- TOSSIM provides PC versions of hardware components
- Event driven runtime
- Compile to a native PC binary

Good structure for monitoring individual components
PowerTOSSIM Architecture

- Add module for tracking power state
- Minor modifications to other modules to report transitions
- Decouple trace gathering from processing

APPLICATION COMPONENTS

- TEMP
- PHOTO
- AM
- CRC
- PHOTOTEMP
- LED
- ADC
- CLOCK
- CC1000

Power State Tracking

Power State Transition Messages

TinyViz Plugin

Post Processor

Power Model
Power Model

Many device components

• CPU
• Radio
• Sensor devices
• LEDs
• ADC
• EEPROM

All can be turned on/off independently

Need to derive an accurate model of power consumption
Measurements

Agilent Infiniium 54832B scope
Digital multimeter for very small currents
Designed microbenchmarks to isolate each component's energy use

Mica2 mote
Microbenchmark Example

Radio transmission at max power
# Mica2 Power Model

<table>
<thead>
<tr>
<th>Component</th>
<th>Current (mA)</th>
<th>Component</th>
<th>Current (mA)</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU</td>
<td></td>
<td>Radio</td>
<td></td>
</tr>
<tr>
<td>Active</td>
<td>8.00</td>
<td>Rx</td>
<td>7.00</td>
</tr>
<tr>
<td>Idle</td>
<td><strong>3.20</strong></td>
<td>Tx (dBm)</td>
<td><strong>3.70</strong></td>
</tr>
<tr>
<td>ADC NR</td>
<td>1.00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Power-down</td>
<td>0.10</td>
<td>-19</td>
<td>5.20</td>
</tr>
<tr>
<td>Power-save</td>
<td><strong>0.11</strong></td>
<td>-15</td>
<td>5.40</td>
</tr>
<tr>
<td>Standby</td>
<td>0.22</td>
<td>-8</td>
<td>6.50</td>
</tr>
<tr>
<td>Extended Standby</td>
<td>0.22</td>
<td>-5</td>
<td>7.10</td>
</tr>
<tr>
<td>Internal Oscillator</td>
<td>0.93</td>
<td>0</td>
<td>8.50</td>
</tr>
<tr>
<td>Sensor board</td>
<td>0.70</td>
<td>+4</td>
<td>11.60</td>
</tr>
<tr>
<td>LEDs</td>
<td><strong>2.20</strong></td>
<td>+6</td>
<td>13.80</td>
</tr>
<tr>
<td>EEPROM</td>
<td></td>
<td>+8</td>
<td>17.40</td>
</tr>
<tr>
<td>Read</td>
<td>6.20</td>
<td>+10</td>
<td><strong>21.50</strong></td>
</tr>
<tr>
<td>Write</td>
<td>18.40</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Visualization Plugin
Estimating CPU energy usage

TOSSIM compiles application code into a native PC binary. Therefore, it is difficult to determine how many CPU cycles are used on the mote.

Could simulate at the AVR instruction level:

- Very slow

Instead, record runtime basic block execution counts, map basic blocks to cycles used on AVR.

- This has low overhead
Cycle count estimation

Insert per-mote counters into each basic block
Cycle count estimation

### App Code

```c
if (x > 0) {
    t = x + 42;
    v = t / pi;
} else {
    v = -1;
}
```

### Mote Binary

```c
if (x > 0) {  →  2 cycles
    t = x + 42; → 21 cycles
    v = t / pi;
} else {
    v = -1; → 1 cycle
}
```

Analyze Mica2 assembly code:

- Compute number of CPU cycles executed for each basic block

---

### Basic Block

<table>
<thead>
<tr>
<th>Basic Block</th>
<th>Cycles</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>21</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
</tr>
</tbody>
</table>
Cycle count estimation (2)

<table>
<thead>
<tr>
<th>Basic Block</th>
<th>Cycles</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>21</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Mote</th>
<th>BB 1</th>
<th>BB 2</th>
<th>BB 3</th>
<th>Cycles</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>10</td>
<td>8</td>
<td>2</td>
<td>10<em>2+8</em>21+2*1=190</td>
</tr>
<tr>
<td>2</td>
<td>15</td>
<td>11</td>
<td>4</td>
<td>15<em>2+11</em>21+4*1=265</td>
</tr>
<tr>
<td>3</td>
<td>12</td>
<td>3</td>
<td>9</td>
<td>12<em>2+3</em>21+9*1=96</td>
</tr>
</tbody>
</table>

This is fast

Potential inaccuracies:

- Need accurate mapping from C basic blocks to binary
- Some low level components have no mapping

In many cases, active CPU cycles very small

- Negligible effect on total power
Efficiency

OscilloscopeRF, 300 simulated seconds
Accuracy

Measured

Simulated

CntToLedsAndRfm
Accuracy

Measured

Simulated

Low power beacon program

Victor Shnayder, Harvard University
Other benchmarks

PowerTOSSIM is accurate: less than 15% error for all tests
Future Work

Add battery model
  • Shouldn't require simulator modifications

Extend to new node platforms
  • CC2420 radio
  • Different sensors

Provide energy usage information at runtime
  • Could help nodes make better resource management decisions
Simulating the Power Consumption of Large-Scale Sensor Network Applications

Victor Shnayder, Mark Hempstead, Bor-rong Chen, Geoff Werner Allen, and Matt Welsh

Harvard University

shnayder@eecs.harvard.edu

http://www.eecs.harvard.edu/~shnayder/ptossim/

PowerTOSSIM is integrated into TOSSIM in the TinyOS CVS tree, and will be part of the 1.1.9 release.