Using the WiseNodes as low power wireless communication and wake-up modules

A. El-Hoiydi, J. Rousselot, J.-D. Decotignie

NCCR MICS WG2 Meeting, Zürich, 23.02.2006
WiseNodes as a low-power radio module

• What are the WiseNodes?
  • Hardware architecture
  • Software architecture
• What performance can we get from the WiseNodes?
  • Choosing the right wake-up period
  • Comparing to other protocols
• What can we do with these modules?
  • Using the WiseNodes as a platform
  • Using the WiseNodes as a wake-up module
  • Using the WiseNodes as a low-power radio module
WiseNode Hardware

- **RADIO**
- **CPU**
- **EEPROM**
- **AD**
WiseNode Software

WiseNode module

Optional host controller

Serial interface

WiseNode module

Embedded application

WiseMAC

Routing

WiseNode module

Hardware Abstraction Layer

Sensors, microcontroller, radio hardware

Host application

HCI

HCI

SW

HW
WiseMAC (Wireless Sensor MAC)

Sampling schedule information

Arrival, wait for right moment

If medium idle, transmit

Wake up, medium idle

Wake up, medium idle

Wake up, medium busy, receive message

T_W

T_P

T_D

T_A

T_C

Waiting for right moment

Transmission if medium idle

Receive message if medium busy

Doze state

RX

TX

ACK

TX

RX

ACK
Sampling, packet reception and forwarding
Performance

CR2 3V Lithium 950 mAh

10 bytes payload/packet
0.08 0.8 8 bits/s 80 bits/s
Performance

Forwarding a 32 bytes packet every 30 seconds in a multi-hop network
Same wake-up period ($T_w = 250$ ms), same hardware (XE1203).

**WiseMAC**

- Total 278 uW (92 uA)
- 14 months

**S-MAC**

- Total 2462 uW (x 8)

**IEEE 802.15.4 MAC Protocol**

- Total 7836 uW (x 28)
Memory usage

- 183 instructions left from 8 k instructions program memory
- About 10 bytes left from 512 bytes data memory
- Not much space for MICS partners to experiment with routing and applications directly on the on-board CPU
Proposal: Using WiseNodes as a Low Power Communication and/or Wake-up Module

Application board with sensors and CPU

Wake-up pin

HCI over UART

Data Len
Host Controller Interface

- **Command (From Controller to MAC)**
  For example, send a data packet of 10 bytes to nodes 0x0002

<table>
<thead>
<tr>
<th>‘c’</th>
<th>‘md’</th>
<th>0x0D</th>
<th>0x0002</th>
<th>‘t’</th>
<th>data to send</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 byte</td>
<td>2 bytes</td>
<td>1 byte</td>
<td>2 bytes</td>
<td>1 byte</td>
<td>10 bytes</td>
</tr>
</tbody>
</table>

```
Length  Protocol
DstAddr
```

- **Event (From MAC to Controller)**
  For example, receive a data packet of 10 bytes sent by node 0x0003

<table>
<thead>
<tr>
<th>‘e’</th>
<th>‘e’</th>
<th>0x0D</th>
<th>‘t’</th>
<th>0x0002</th>
<th>data received</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 byte</td>
<td>1 byte</td>
<td>1 byte</td>
<td>1 byte</td>
<td>2 bytes</td>
<td>10 bytes</td>
</tr>
</tbody>
</table>

```
Length  Protocol
SrcAddr
```
Thank you for your attention.