Battery-less sensor nodes for 802.15.4/ZigBee wireless networks

(Presented at Wireless Congress Conference Munich, 9th November 2011)

Dipl. Ing. Marcel da Silva

Prof. Dr. Marcel Meli

Contact: Marcel.meli@zhaw.ch
Outline

- A short presentation of our activities in low power wireless
- Introduction and motivation
- Some challenges while implementing battery-less for 802.15.4 networks
- Overcoming the challenges: examples
- It can only get better: Working with FRAM
- Conclusions
- Questions
Our activities

• Institute of Embedded Systems, Winterthur, Switzerland
  • Part of Zurich University of Applied Sciences
• Involved in teaching, applied research projects
  • Wired: Industrial Real Time Communication (Ethernet, 1588...)
  • Wireless: WPAN, RFID, UWB, ...
  • Energy harvesting, very Low power applications

• Examples of Low power developments:
  • Passive, Intelligent RFID Tag
    ▪ Battery-less: powered only by the RFID reader
    ▪ Uses a 32-Bit microcontroller to emulate protocols
    ▪ Successful emulation of LF, UHF, Gen2
    ▪ Identification, sensor, security, ... (RFID sensor network)
    ▪ Range of several meters
Our activities

- 6WLoPAN-Ethernet Bridge

- CAN, JTAG
- Cortex M3 with 1588 PTP support
- USB
- PoE HW
- 802.15.4 module
- Ethernet 100 Mb/S
- 6WLoPAN-Ethernet Bridge

Diagram:
- Layer 1: 802.15.4, Ethernet
- Layer 2: IEEE 802.15.4, IEEE 802.3
- Upper Layer: Apps, TCP, UDP
- Bridging Function: z.B.: Webserver
Our activities

- Wireless automation with different energy harvesting sources

The receiver controls a lamp

The sender can use different power sources

Works with 802.15.4/Zigbee and other wireless protocols

Electro-dynamic  Seebeck  Solar
Introduction and motivation

- Battery-free wireless sensor nodes are very convenient in many applications.
  - They eliminate the costs associated with installing batteries and
  - Reduce maintenance (and logistics) headaches.
- For these reasons, they are gaining in popularity.
- There are limitations to the kind of application and wireless protocol that can successfully be implemented, especially when energy is produced intermittently.
  - Can the application live the small amount of energy?
- The interest in 802.15.4-based wireless systems such as ZigBee has also led to its consideration for battery-less nodes.
  - 802.15.4/Zigbee/... are known to have an important overhead (in communication and associated software)
Introduction and motivation

• In this paper:
  • We briefly discuss some of the challenges faced when using 802.15.4/ZigBee in applications where little energy is available.
  • We contend these protocols can be used in certain cases, and that things will only get better (in terms of energy), considering the developments in the semiconductor industry.
    ▪ Tendency to lower energy budgets
    ▪ Better microcontrollers, ... etc
  • We illustrate our point by showing (for the first time) that some of the recently introduced microcontrollers featuring FRAM memory allow interesting gain in energy
    • Making it even easier to overcome the challenges mentioned above.
    • Or allowing new applications
  • We will not be covering ZigBee Green power
    • needs less bytes
    • What woks here will work even better with Green Power
Challenges

• There are several challenges, but we will look at 2 of them which we consider the most important.

• We will not go into much details, since you can find the information in presentations given at the European ZigBee Developer’s Conference in the previous years.
  • 2008 First battery-less switch using the ZigBee protocol
  • 2009 Pairing issues discussed
  • 2010 Using tiny solar cells for ZigBee
  • 2011 Monitoring dynamic current consumption to optimize energy needs
  • 2012 Come and see (we cannot go yet back to the future 😊)
  • We will draw on those previous works in the first part of this presentation

• The challenges are:
  • The problems associated with the large overhead in communication
  • The problems associated with joining the network and binding “end points”
Challenges: overhead of ZB is too large → more energy

- **Switch example:** The overhead is important

<table>
<thead>
<tr>
<th>Octets</th>
<th>2</th>
<th>1</th>
<th>0/2</th>
<th>0/2/8</th>
<th>0/2</th>
<th>0/2/8</th>
<th>0/5/6/10/14</th>
<th>Variable</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fields</td>
<td>MAC</td>
<td>Frame control</td>
<td>Sequenc e number</td>
<td>Dest. PAN</td>
<td>Dest. Addr.</td>
<td>Source PAN</td>
<td>Source Addr.</td>
<td>Security Header</td>
<td>Payload</td>
</tr>
<tr>
<td>Example</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>Variable</td>
<td>2</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Octets</th>
<th>2</th>
<th>2</th>
<th>2</th>
<th>1</th>
<th>1</th>
<th>0/8</th>
<th>0/8</th>
<th>0/1</th>
<th>var</th>
<th>var</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Octets</th>
<th>1</th>
<th>0/1</th>
<th>0/2</th>
<th>0/2</th>
<th>0/2</th>
<th>0/1</th>
<th>1</th>
<th>Variable</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fields</td>
<td>APS</td>
<td>Frame control</td>
<td>Dest. endpoin t</td>
<td>Group Addr.</td>
<td>Cluster Id.</td>
<td>Profile Id.</td>
<td>Source endpoint</td>
<td>APS counter</td>
</tr>
</tbody>
</table>

- The **green** is actually what you want to get through
- Lots of other “companions” going with (+ 6 bytes at Phy level)
Challenges: overhead of ZB is too large → more energy

- To be able to send switch commands for ZigBee, the following is needed:
  - Clear channel assessment (switch on the receiver to verify channel activity)
  - Transmit $\sim 30 + 2 + 6 = 38$ bytes. (30 bytes of data + 2 bytes (FCS) on MAC layer, 6 Bytes for PHY overhead)
    - All these layers will also mean more time/energy in software
  - Receive ACK and react.
    - Resent is data was not received
    - Do nothing (→ you can also ignore the ACK)

- The overhead is important → you “waste” most of the energy
  - This is not really wasted because many “extra” bytes contribute to the flexibility of ZigBee
  - As energy needs are reduced in other areas, this flexibility could keep you going.
    - If you design too just, there is little room for growth.
Challenge: Joining, binding\(\rightarrow\) more energy

- We will just say pairing.
- With pairing we refer to all what is needed to set up the communication so that 2 end devices can properly and "securely" exchange information:
  - Proper frequency channel, joining a network, binding end points, security parameters, ...
- Pairing involves the exchange of many parameters (channel number, src/dest/PAN addr, endpoint ID... )
Challenge: Joining, pairing → more energy.
Possible solution

- Use “Out of band channel for pairing”
  - How do I get the needed parameters from Master to Slave without having to use energy for the radio?
  - Possible solution: Use RFID pairing
  - Dual ported Memory

Channel number
PAN ID
SRC ID
DEST ID
End device

- Low power micro + 802.15.4 transceiver
- Power management
- Serial link
- Non volatile array
- Analog front-end
- RFID

PCB
RFID antenna
Challenge: Joining, pairing $\rightarrow$ more energy.
Possible solution

- Use RFID for pairing

- 1 ZB device + RFID tag+ with 2 end points.
  ON/OFF lamp. Dimmer

- 1 mobile with RFID reader and software for pairing

2 Battery-less switches + RFID tag sending ZB compatible packets
Design

- Components are getting better → they need less energy
  - Some of the problems associated with energy will become easier to solve
    - Competition for market shares
    - Better radios and better microcontrollers
    - Better hardware and software engineers 😊

- One example is the use of the new FRAM micro
  - Battery switches were revisited, 2 non-volatile technologies used,
  - Results were compared

- Block diagram of the node to be tested

![Block diagram of the switch wireless node](image)
Design

• FRAM Microcontroller
  • MSP430FR5739, Texas Instruments. Features an FRAM as NV memory.
  • The use of this FRAM opens the doors to very interesting concepts.
  • Parameters that need to be updated and kept while the wireless node is active can easily be stored in the FRAM, with little energy.
  • The state of the microcontroller can be saved, and then restored later when enough power is available.
  • This is very significant for devices powered intermittently.
  • There are not many microcontrollers with FRAM on the market. In some cases, an external serial FRAM memory can be attached to the microprocessor. The firm RAMTRON produces serial FRAM.
Design

- **Microcontroller**
  - For comparisons, we used a second microcontroller
  - EEPROM as non-volatile memory. The device is also good enough to allow ZigBee compatible frames to be sent, and update of the APS to be done.
  - We will not name this device, in order to avoid misunderstandings.
  - The main point in this work is simply to show the difference between using an EEPROM and using an FRAM. It is not the comparison of microcontrollers. We do not pretend here that the MSP430 is a better core

- **Transceiver from Atmel**
  - AT86RF231
    - Used already in our first work (no longer the best in terms of energy, but widely available)
    - >12 mA Tx/Rx current
Results (FRAM used, stable voltage source)

- FRAM is used to update APS in non-volatile counter
  - Stable power supply, @ 2volts
  - Less than 40μJ is enough to complete the task
    - Start up of systems, init, sending
    - Looking only at the processing, sending, → 30μJ will be enough

![Current measurement while sending 1 ZigBee message. APS incremented MSP430FR5739 (FRAM micro) + Atmel AT86RF231 + stable 2V source.](image)
Results (EEPROM used, stable voltage source)

- EEPROM is used to update APS in non-volatile counter
  - Stable power supply, @ 2volts
  - NOTE: EEPROM write should be at 20μJ (radio not in proper saving mode here)
  - EEPROM update (read, erase, write) leads to tens of extra microjoules

---

**Current measurement while sending 2 ZigBee messages. APS incremented Micro1 (with internal EEPROM) + Atmel AT86RF231 + stable 2V source.**

<table>
<thead>
<tr>
<th>Event</th>
<th>Energy (μJ)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Startup</td>
<td>5.4</td>
</tr>
<tr>
<td>Init</td>
<td>17.0</td>
</tr>
<tr>
<td>Processing/sending + EEPROM erase</td>
<td>45.5</td>
</tr>
<tr>
<td>EEPROM write</td>
<td>40.5</td>
</tr>
<tr>
<td>Goto Sleep</td>
<td>3.3</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>111.7</strong></td>
</tr>
</tbody>
</table>
Results (FRAM is used)

- FRAM is used to update APS in non-volatile counter
  - EH is used
  - Voltage still high enough after the whole operation is completed
  - Less than 50\(\mu\)J is enough

![Current measurement while sending 1 ZigBee message. APS incremented MSP430FR5739 (FRAM micro) + Atmel AT86RF231 + Energy Harvester.](image)
Results (More energy when EEPROM used)

- EEPROM used to update APS in non-volatile counter
  - EH is used
  - Voltage gets dangerously low

![Graph showing energy consumption and voltage levels during ZigBee message transmission](image-url)

- Current measurement while sending 1 ZigBee message. APS incremented using Micro1 (with internal EEPROM) + Atmel AT86RF231 + EH source.

- EH voltage: Source: 1V/div
- System current: 5mA/div

Startup: 9.4 μJ
Init: 22.5 μJ
Processing/Sending + EEPROM erase: 39.3 μJ
EEPROM write: 16.3 μJ
Total: 87.5 μJ
Results (1 activation, 2 messages, FRAM used, APS updated once)

- The EH is activated once, 2 messages are sent (radio at 0dBm)
  - ZB messages to control lamp. FRAM updated. Radio works down to 1.8 V
  - There is enough energy to do more

![Graph showing EH voltage, system current, and energy usage](image)

**Current measurement while sending 2 ZigBee messages. APS incremented once**

- TI MSP430FR5739 (Micro with FRAM) + Atmel AT86RF231 + EH source.

- **EH voltage**
  - Source: 1V/div

- **System current**
  - 5mA/div

- **Startup:** 7.8 μJ
- **Init:** 9.4 μJ
- **Processing, APS inc., sending:** 30.1 μJ
- **Init:** 4.7 μJ
- **Processing, sending:** 28.7 μJ
- **Total:** 80.7 μJ
Tests (1 activation, 2 frames sent, FRAM used)

Packet index: 1  Length: 32
Raw data (hex): 41 88 01 32 00 01 00 6F 79 48 00 00 00 6F 79 0A 01 00 0A 06 00 04 01 10
07 11 53 02 46 C9
RSSI [dBm]: -52  Correlation value: 108  CRC OK: 1

Packet index: 2  Length: 32
Raw data (hex): 41 88 01 32 00 01 00 6F 79 48 00 00 00 6F 79 0A 01 00 0A 06 00 04 01 10
07 11 53 02 46 C9
RSSI [dBm]: -53  Correlation value: 108  CRC OK: 1

Packet index: 3  Length: 32
Raw data (hex): 41 88 01 32 00 01 00 6F 79 48 00 00 00 6F 79 0A 01 00 0A 06 00 04 01 10
08 11 53 02 46 C9
RSSI [dBm]: -55  Correlation value: 108  CRC OK: 1

Packet index: 4  Length: 32
Raw data (hex): 41 88 01 32 00 01 00 6F 79 48 00 00 00 6F 79 0A 01 00 0A 06 00 04 01 10
08 11 53 02 46 C9
RSSI [dBm]: -56  Correlation value: 108  CRC OK: 1
Conclusions

• It is possible to design sensors, switches, powered on EH that will send data using 802.15.4 / ZigBee compatible frames.
  • This will work with EEPROM
  • Works even better with FRAM
    ▪ More energy available for retransmit or other activities
    ▪ Even better when Green Power addition to the standard is ready

• The introduction of ZigBee “Green Power” will make it even easier to use battery-less devices in ZigBee networks, since there is less data to send → less energy required

• New Harvesters are coming on the market that deliver more energy or are more robust → devices will

• Devices with integrated energy management are available, making it possible to live on small solar cells, energy harvested from temperature difference, ... etc.

• Competition means better devices → the next generation of transceivers will need even less energy.
Thanks, more information

- Thanks to Texas Instruments for making FRAM devices available for our work

- For more information:
  - Marcel Da Silva, Bsc in Computing Sciences
  - Prof. Dr. Marcel Meli, Head of Wireless Systems Group

- Zurich University of Applied Sciences (ZHAW)
  Institute of Embedded Systems (InES)
  Technikumstr. 9
  CH-8401 Winterthur
  Phone: +41 58 934 75 25
Question time

??????
Our activities (New)

- Dynapic Wireless (Battery-less, wireless switch >10 million cycles)

- Battery-less, wireless piezo-switch (patent filed)
- > 10 million cycles
- Silent (no unwanted clicks), thin, flat, fits on many surfaces