

Energy modelling in sensor networks

D. Schmidt¹, M. Krämer², T. Kuhn², and N. Wehn¹

¹Microelectronic System Design Research Group, University of Kaiserslautern, Germany

²Networked Systems Group, University of Kaiserslautern, Germany

Abstract. Wireless sensor networks are one of the key enabling technologies for the vision of ambient intelligence. Energy resources for sensor nodes are very scarce. A key challenge is the design of energy efficient communication protocols. Models of the energy consumption are needed to accurately simulate the efficiency of a protocol or application design, and can also be used for automatic energy optimizations in a model driven design process. We propose a novel methodology to create models for sensor nodes based on few simple measurements. In a case study the methodology was used to create models for MICAz nodes. The models were integrated in a simulation environment as well as in a SDL runtime framework of a model driven design process. Measurements on a test application that was created automatically from an SDL specification showed an 80% reduction in energy consumption compared to an implementation without power saving strategies.

1 Introduction

In the vision of a world of ambient intelligence in which people are constantly interacting with an adaptive, proactive, and user friendly environment, wireless sensor networks are a vital component, because of their inherent redundancy, their robustness, and their transparency to the user. As the nodes in AmI sensor networks have very limited energy resources and computing power, a key challenge is the development of energy efficient application specific communication protocols. In this paper, we present a component based modelling methodology for energy modelling of sensor nodes for these kinds of networks. The proposed approach is applicable for a wide range of wireless sensor nodes and applications. The models serve two purposes:

- At design time the models are used to estimate or simulate the energy consumption of the designed application and communication protocols.

Correspondence to: D. Schmidt
(schmidt@eit.uni-kl.de)

- As the models are very lightweight they can be used to estimate the energy consumption at runtime. This way the node can estimate the remaining lifetime of its energy source and adjust its quality of service accordingly.

We show how the models can be integrated into a simulation environment that is capable of simulating complete networks of nodes accurately regarding their behavior and energy consumption. This allows for optimization of network behavior and network lifetime rather than single node lifetime.

As a side-effect of the modelling methodology, the model's semantics can be adapted to reflect basic energy-optimal strategies, without having to add energy optimization strategies explicitly to the model. This is done in the model runtime environment as well as in the code patterns that are used for transforming the model into code. Using this approach, the code for the hardware nodes can be generated automatically and validated using the integrated simulation framework.

The remaining parts of this paper is structured as following: Sect. 2 provides an overview over different energy modelling approaches and introduces our proposed energy modelling methodology. The simulation environment and the underlying integration methodology is presented in Sect. 3. An exemplary energy model for the MICAz (MICAz, 2006), a state-of-the-art sensor node, that was obtained using the proposed energy modelling methodology is shown in Sect. 4 before Sect. 5 shows in a case study how these models can be used in a MDD process for energy aware communication protocols. Section 6 summarizes our conclusions and points out areas of future research.

2 Energy modelling methodology

To model the energy consumption of microelectronic devices several different approaches can be taken. Highly accurate estimates can be obtained by simulation of a microelectronic circuit on transistor level or on even lower abstraction levels, for example with SPICE (SPICE, 2006). While these simulations cover all effects including leakage, switching energy,

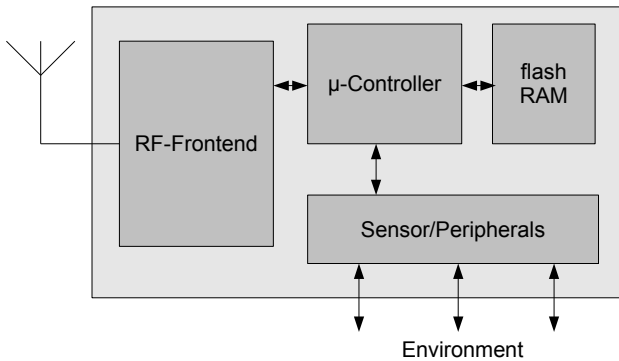


Fig. 1. Architecture of a typical sensor node.

etc., it is very tedious to create the models, as a detailed in-depth knowledge of the hardware is mandatory. Also, simulation on circuit level is very time consuming and is not suitable to simulate whole networks with a large number of individual nodes.

Hence, for many microprocessors energy models that simulate the hardware on the instruction level have been created. This is usually done by measurement of synthetic benchmarks (Senger et al., 2005). These contain loops that execute only one kind of instruction, so that the energy consumption of every single instruction can be calculated from the measurements. Even more measurements have to be done to model inter-instruction dependencies or the impact of operands on energy consumption. These models still allow for relatively accurate energy simulations but require a lot less knowledge of the hardware circuit. Another benefit over circuit simulation is the improved simulation runtime. However, the cost of model creation is relatively high and the resulting model is still too complex to be used at runtime, as sensor nodes typically have very limited computational power.

To overcome the mentioned problems we propose a component based high level modelling approach. A typical sensor node's architecture is depicted in Fig. 1. It is made up of several components like a microcontroller, a transceiver chip, sensor electronics, and various other devices like LEDs, flash RAMs, etc. Each of these components can be operated in different states, e.g. power down, idle, transmitting, and receiving in case of a transceiver chip. Based on the assumption that instruction level effects are negligible for the components used on today's sensor nodes we consider the energy consumption in each state to be constant. Thus, we can model the energy consumption of each individual component using a finite state machine (FSM) in the following way:

- Every operational state of a component is modeled as one state in the FSM.
- Every possible change from one operational state to another is modeled as a state transition in the FSM.

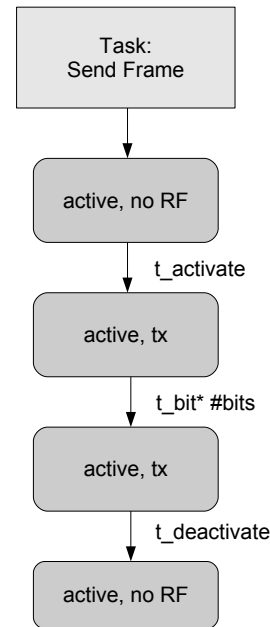


Fig. 2. Parametric flowchart for frame transmission.

- Every state in the FSM is associated with an energy consumed per time.
- Every transition in the FSM is associated with a time needed to switch between two operational states.
- The FSM has a well-defined initial state that corresponds to the stable state a component reaches after power up.

All the times and energy consumptions associated with states and state transitions can be obtained by relatively simple measurements or from data sheets.

Once every component of a sensor node has been modeled as a FSM a similar model for the whole sensor node can be constructed. Basically it is the product automaton of all component models. As usually only the microcontroller can trigger state changes the product automaton can be greatly reduced, because only in states with an active microcontroller other components can change their states.

The resulting FSM only represents the possible states the system can be in, but does not reflect constraints on state changes that appear in normal operation. Wireless communication is a good example as after the transceiver has begun frame transmission it can not be interrupted. This is not represented in the FSM model. Hence, in a second step dynamic behavior of the sensor node in typical scenarios is analyzed to model these constraints. A node's dynamic behavior is defined by the SDL model, the runtime environment implementing the SDL semantics, and the code transformation patterns. Power saving strategies can also be integrated anywhere, ranging from explicit power saving strategies that are part of the application model to implicit strategies that

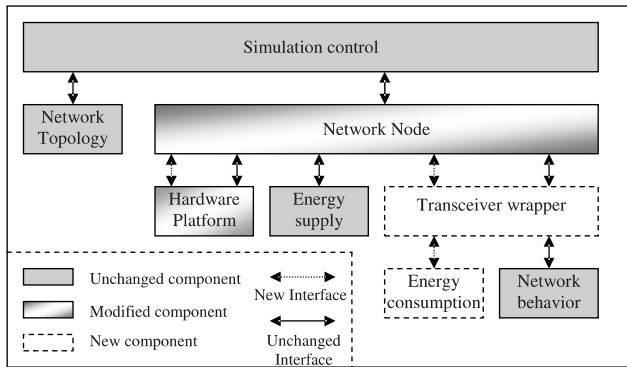


Fig. 3. Structure of the simulator integration framework.

are part of the runtime environment. The complete dynamic behavior of a node can be formalized as a set of communicating state machines. This holds for the SDL model, which is based on state machines, as well as the runtime environment that executes the SDL model. These state machines have common actions with the state machine describing the nodes hardware. An example for a part of a nodes dynamic behavior is given in Fig. 2. It represents the task of sending one frame of a given number of data via the wireless interface to a distant node. It starts in the state where the micro-controller is the only active component of the sensor node. It then triggers the transceiver to transmit one frame. The number of data bytes to be transmitted is the parameter in this case and determines the amount of time spent in transmit mode. After transmission the transceiver is deactivated again to save energy and the initial state is reached again.

From these flowcharts it is very easy to calculate the energy consumed by the specified task, conforming to Eq. (1). Here P_{state} is the power consumed in the state “state”, t_{state} is the time spent in this state. P_{trans} and t_{trans} are the power and time for the transitions between to states.

$$E = \sum_{state} P_{state} \cdot t_{state} + \sum_{trans} P_{trans} \cdot t_{trans} \quad (1)$$

This analytic model can be integrated in a simulator, offering the possibility to perform simulations that predict the energy consumption of a system very accurately.

3 Simulator integration methodology

To simulate the energy consumption of an array of sensor nodes accurately, it is necessary to capture all relevant simulation artifacts that affect the behavior of the sensor node array, i.e. platform resources, network resources, and energy resources. This is due to the fact that one simulated artifact can significantly affect all of the other simulated artifacts, and therefore change the outcome of the simulation. For example, the energy consumption of a sensor node participating

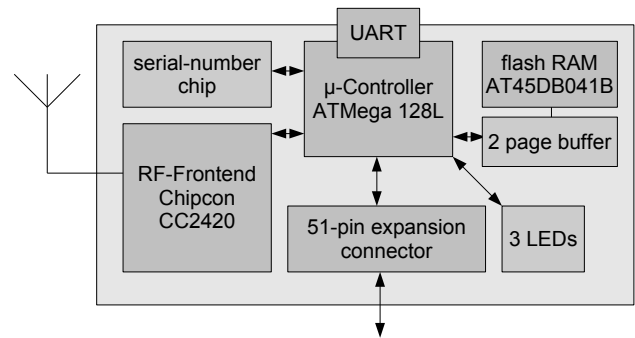


Fig. 4. Architecture of MICAz.

in a large sensor network is significantly affected by its wireless transceiver chip. The transceiver chip is used for communicating with other network nodes. Due to network collisions and bandwidth limitations, it is possible that a large number of transmission errors occur, changing the communication pattern of every network node, and thus affecting the energy consumption. Furthermore, platform limitations like inaccurate timers can make this effect worse by introducing jitter into time-synchronized networks. To capture all of these effects, a number of highly specialized simulators are necessary (SAM, 2006).

In (SAM, 2006), it has been shown that simulators for simulating network behavior are already available, that these specialized simulators can be turned into simulation components, and that they can be interconnected using a message based interface to form a system simulator. To integrate the simulation of energy consumption, two steps must be performed: The state machines describing a nodes energy consumption must be integrated, and the message based interfaces for coupling simulation of energy consumption with the remaining parts must be integrated into the system simulator. Therefore, two methodologies are supported:

- Integration of energy consumption into already existing simulators is feasible in the case, that a simulation of the energy consumption of a hardware component is already included in a simulation component. In this case, only an interface must be implemented into the simulation component. The Avrora simulator (Avrora, 2005) includes already the simulation of energy consumption.
- Integration of energy consumption as new simulation components is used when it is not feasible to add another interface to a simulator component. Using this approach, the simulated artifacts are clearly separated, i.e. one simulation component simulates the behavior of a hardware component, and another component simulates its energy consumption by implementing the energy model.

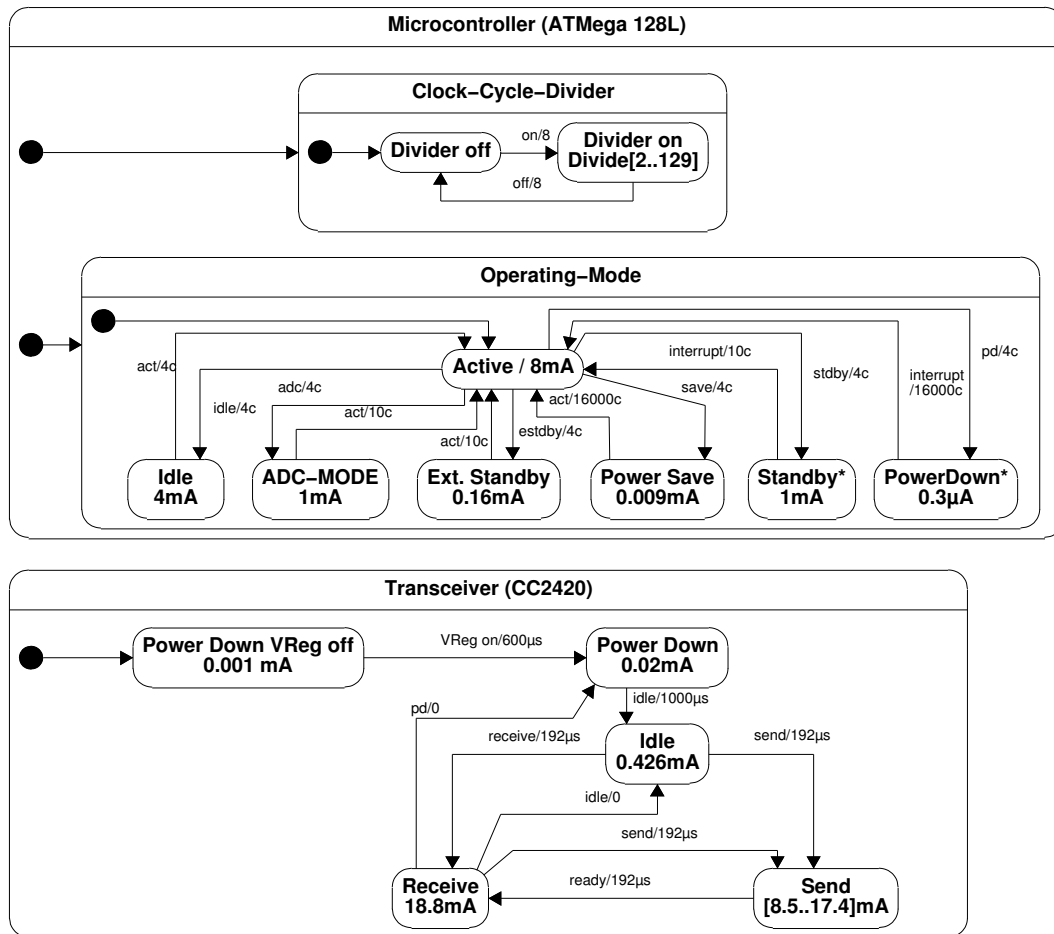


Fig. 5. Energy model of MICAz (only microcontroller and transceiver).

Both methodologies can easily be integrated into the simulator framework. Figure 3 shows the integration of both methodologies for the simulated hardware platform, and for the simulated network. It is shown, that the central component “Network Node”, which is part of the simulator framework needs to be changed to integrate the energy consumption into the simulation. This is necessary, because this component controls all simulators that simulate one node. For the simulated hardware, it was feasible to integrate the energy consumption into the already available simulator. For the simulation of the wireless network, a new component implementing the energy model was created, and the simulation component for the network behavior was replaced with a wrapper, distributing simulator messages among the original simulation component for network behavior and the added component that tracks the energy consumption.

4 Energy model for MICAz

The MICAz from Crossbow Technology is a typical wireless sensor node. It features an 8-bit Atmel microcontroller

with RISC architecture, clocked at 7.3728 MHz, with 4 kB of internal SRAM and 4 kB data EEPROM as well as 128 kB of internal flash memory. The ZigBee-compliant wireless transceiver chip operates at data rates up to 250 kbit/s. A 512 kB flash memory can be accessed via two SRAM page buffers of 264 bytes each. Three LEDs can be used to show the operational status of the device and each node is equipped with a serial-number chip that gives a node its unique ID. The MICAz has a 51-pin expansion connector as an interface to arbitrary sensors. Figure 4 shows the overall architecture.

For our energy model we only considered the microcontroller, the transceiver chip, and the flash memory. The LEDs can be turned off to reduce energy consumption and the energy consumption of the serial-number chip is negligible. Two component models for the transceiver and the microcontroller of the MICAz are shown in Fig. 5. Here, transition times are given in seconds for the transceiver and in clock cycles for the microcontroller. The microcontroller can be in different power reduced states that differ in the number of timers running and interrupt sources active.

As the MICAz operates at a constant input voltage of 3 V, the energy consumption for every state is given in mA. For the transceiver send state, however, there is not a single value given, but a range of energy consumption. The actual consumption in this state depends on the chosen output power of the transceiver chip. It is remarkable that even at the highest output power the energy consumption is lower than in receive mode.

5 Energy-optimization using energy-models

Energy optimization can be performed on several levels. It can be optimal for the node, for a few nodes or for a whole network. With the simulators described above it is possible to analyze and optimize the consumed energy on each level. For the optimization of a network, special protocols like MacZ (MacZ, 2006), which is currently under development, are needed. The first attempt is the reduction of consumed energy for a single node. For this task accurate models for the used hardware are needed. With our models of the MICAz we have extended the scheduler of the specification language SDL to optimize energy. SDL is a reactive language with additional timers. This enables the language to react on external events, like interrupts or even after a specified time period. SDL expects a continuous timebase and therefore the microcontroller has to wake up in regular intervals to increase the timebase. If no timer is expired and no external signal has arrived the microcontroller can enter a lower energy state. The entered energy state is chosen from the model and is based on the active components. In some cases the operating mode with the lowest consumption cannot be entered, e.g. due to an active transfer via the serial (UART) interface or because the wake up-time is longer than to the next scheduled timer.

To verify the results of the simulator and test the scheduler, a small example application was specified. The application gets temperature measurements every 10 s via serial communication link and send an average value every minute to another station of the same kind.

The measurement of the consumed energy for a time of 10 min shows that the consumption using the scheduler is reduced from 16.2 J to 3.09 J. Using the scheduler for this example application saves 80% energy. This consumption can be reduced in a future version if the precision of the timers is not so important. In the current implementation of the scheduler the precision of the timebase is set to $\frac{1}{16384}$ s, which is the same as without the new scheduler. Since the microcontroller has to wake up at least every 7 ms, it is planned to reduce this precision, if energy is more important.

6 Conclusions and future work

In this paper we presented a simple yet effective methodology to create energy models for wireless sensor nodes.

These models consist of two parts: a FSM modelling energy consumption in the different states a node can be in, and flowcharts that represent power saving best-practice-strategies for the dynamic behavior of the node in typical scenarios. We have shown how the energy models can be integrated in platform simulators to simulate a complete network of sensor nodes taking platform resources, network resources, and energy resources into account.

In a case study, we have used the proposed methodology to obtain an energy model for a MICAz sensor node and have used the modified SDL scheduler for an automated generation process for energy optimized code. This way we could show that a MDD process can generate energy optimized code directly from an SDL specification. Compared to an implementation that does not facilitate power saving strategies we measured an improvement in energy consumption of 80%.

Future work will focus on the modelling of energy sources enabling the nodes to estimate their remaining run time. Quality of service degradation or changing roles in a network can benefit from this.

Acknowledgements. This work was financed in context of the BelAmI-Project (BelAmI, 2006).

References

- Avrora: Avrora Simulator, <http://compilers.cs.ucla.edu/avrora>, 2005.
- BelAmI: Bilateral German-Hungarian Collaboration Project on Ambient Intelligence Systems, <http://www.belami-project.org>, 2006.
- International Telecommunications Union: Specification and Description Language (SDL), ITU-T Recommendation Z.100, August 2002.
- Kuhn, T.: MacZ – a QoS MAC Layer for Ambient Intelligence Systems, in: *Advances in Pervasive Computing 2006*, edited by: Pfeifer, T., Schmidt, A., Woo, W., et. al., Adjunct Proceedings of the 4th International Conference on Pervasive Computing, books@ocg.at, Vol. 207, Austrian Computer Society (OCG): Vienna, 244 pp., ISBN 3-85403-207-2, 2006.
- MICAz: MICAz ZigBee Series (MPR2400), <http://www.xbow.com/Products/productsdetails.aspx?sid=101>, 2006.
- SAM06: Kuhn, T. and Becker, P.: A Simulator Interconnection Framework for the Accurate Performance Simulation of SDL Models, Springer LNCS 4320, ISBN 3-540-68371-2, 2006.
- Senger, R., Marsman, E., and Brown, R.: Methodology for Instruction Level Power Estimation in Pipelined Microsystems, [https://domino.research.ibm.com/acas/w3www_acas.nsf/images/conf05/\\$FILE/senger.pdf](https://domino.research.ibm.com/acas/w3www_acas.nsf/images/conf05/$FILE/senger.pdf), 2005.
- SPICE: Simulation Program with Integrated Circuit Emphasis, <http://bwrc.eecs.berkeley.edu/Classes/IcBook/SPICE/>, 2006.