Dipl.-Ing. Leander Bernd Hörmann, BSc.

Hardware-Software Interactions in Energy Harvesting Wireless Sensor Networks

---

Dissertation

vorgelegt an der
Technischen Universität Graz

zur Erlangung des akademischen Grades
Doktor der Technischen Wissenschaften
(Dr.techn.)

durchgeführt am Institut für Technische Informatik
Technische Universität Graz

Graz, im Mai 2013

EIDESSTATTLICHE ERKLÄRUNG

Ich erkläre an Eides statt, dass ich die vorliegende Arbeit selbstständig verfasst, andere als die angegebenen Quellen/Hilfsmittel nicht benutzt und die den benutzten Quellen wörtlich und inhaltlich entnommenen Stellen als solche kenntlich gemacht habe.

Graz,am ....
(Unterschrift)

STATUTORY DECLARATION

I declare that I have authored this thesis independently, that I have not used other than the declared sources / resources, and that I have explicitly marked all material which has been quoted either literally or by content from the used sources.

....
date (signature)

Abstract

Wireless sensor networks (WSNs) are typically used in application areas without wired infrastructure and so they are energy-critical systems. These networks consist of a large number of sensor nodes. The sensor nodes exchange information via radio technology and each of them comes with its own energy source. As energy sources, batteries or energy harvesting systems (EHSs) can be used. The latter convert environmental energy into electrical energy and enable a continuous and autarkic operation. However, the harvestable power of the environment and also the capacity of the batteries are limited. Therefore, an optimization concerning the energy consumption is necessary.

Sensor nodes feature hardware and software which interact with each other. Hence, the hardware state can influence the behavior of the software and vice versa. Particularly energy harvesting WSNs come with software components which adapt their behavior to the energy state of the sensor node. To be able to optimize such WSNs, an analysis of the interactions is necessary.

This thesis deals primarily with the analysis of hardware-software interactions in energy harvesting wireless sensor networks. For this purpose, a simulation environment was implemented specifically for a concurrent simulation of hardware and software. The energy harvesting hardware is modeled at component level to ensure realistic results. Two case studies illustrate that the analysis of the interactions actually enhances the optimization of WSNs.

Kurzfassung

Drahtlose Sensornetzwerke werden typischerweise in Anwendungsbereichen ohne kabelgebundene Infrastruktur eingesetzt und sind deshalb energiekritische Systeme. Diese Sensornetzwerke bestehen aus einer Vielzahl von Sensorknoten, welche mittels Funktechnik Informationen übermitteln, wobei jeder über eine eigene Energiequelle verfügt. Als Energiequellen können Batterien oder Energiegewinnungssysteme verwendet werden. Letztere wandeln Energie aus der Umgebung in elektrische Energie um und ermöglichen somit einen ununterbrochenen und autarken Betrieb. Da jedoch die gewinnbare Leistung aus der Umgebung und auch die Speicherkapazität der Batterien begrenzt sind, ist eine Optimierung hinsichtlich des Energieverbrauches notwendig.

Sensorknoten weisen eine Hardware und eine Software auf, welche miteinander wechselwirken. Daher kann der Hardwarezustand das Verhalten der Software beeinflussen und umgekehrt. Vor allem Sensornetzwerke mit Energiegewinnungssystemen verfügen häufig über Softwarekomponenten, welche ihr Verhalten an den Energiezustand der Sensorknoten anpassen. Um solche Sensornetzwerke optimieren zu können, ist eine Analyse der Wechselwirkungen notwendig.

Diese Arbeit beschäftigt sich also im Wesentlichen mit der Analyse von Hardware-Software-Wechselwirkungen in Sensornetzwerken mit Energiegewinnungssystemen. Dazu wurde eine Simulationsumgebung implementiert, und zwar für die gleichzeitige Simulation von Hardware und Software. Die Energiegewinnungs-Hardware im Speziellen ist auf Bauteilebene modelliert und gewährleistet somit realistische Ergebnisse. Zwei Fallstudien zeigen, dass die Analyse der Wechselwirkungen tatsächlich die Optimierung der Sensornetzwerke verbessert.

Danksagung

Das Doktorat ist für mich besonders wichtig, da ich während dieser Zeit reichhaltige Erfahrungen in Forschung und Lehre sammeln konnte. Deshalb möchte ich mich bei allen bedanken, die mir das Studium an der Technischen Universität Graz am Institut für Technische Informatik ermöglicht haben.

Als erstes möchte ich mich bei Em.Univ.-Prof. Dipl.-Ing. Dr.techn. Reinhold Weiß bedanken, ohne den diese Dissertation nicht das wäre, was sie nun ist. Während unzähligen Gesprächen lenkte mein Betreuer die wissenschaftliche Arbeit immer wieder in die richtige Richtung. Weiters bedanke ich mich bei Ass.Prof. Dipl.-Ing. Dr.techn. Christian Steger und Dipl.-Ing. Dr.techn. Christian Kreiner, durch die ich in interessanten Projekten arbeiten konnte. Sie standen mir immer mit Rat und Tat zur Seite. Besonderer Dank geht an meine Kollegen Dipl.-Ing. Dr.techn. Philipp Glatz und Dipl.-Ing. Karima Hein, die mein Schaffen inspirierten und mir in Forschung und Lehre sehr geholfen haben. Bei Silvia Reiter bedanke ich mich für die sehr gute Organisation und Hilfestellungen bei administrativen Fragen. Bei Dipl.-Ing. (FH) Engelbert Meissl bedanke ich mich für die sehr gute Zusammenarbeit und Hilfestellungen bei technischen Realisierungen und einer Vielzahl von Versuchsaufbauten. Weiters möchte ich mich bei den Studenten bedanken, mit denen ich erfolgreich interessante Projekte umgesetzt habe: Marcus Dietl, Dipl.-Ing. Stephan Gether, Michael Kalcher, Alexei Kruglov, Johannes Leibetseder, Philipp Maierl, Dipl.-Ing. Hannes Mock, Paul Rudelstorfer, Wolfgang Staudacher und Dipl.-Ing. (FH) Dipl.-Ing. Michael Steinberger.

Ich bedanke mich auch bei meiner Familie und meinen Freunden für ihre Zeit und unaufhörliche Unterstützung. Vor allem meine Mutter, Hildegard, und mein Vater, Alfred, ermutigten mich immer wieder und ohne ihre Hilfe wäre meine Ausbildung nicht möglich gewesen. Besonders bedanke ich mich auch bei meiner Verlobten, Sabine Zarl, die mir unermessliche Geduld und viel Verständnis während meines gesamten Studiums entgegen brachte.

Graz, Mai 2013 — Leander Bernd Hörmann

Extended Abstract

Wireless sensor networks (WSNs) provide a digital interface with their environment to enable remote monitoring and tracking. WSNs are a key technology for ubiquitous computing. WSNs consist of small and networked devices, so called sensor nodes. These devices measure physical quantities, process the measurement data, and transmit the gained information to one or more recipients in a multi-hop manner. Precision agriculture, wildlife monitoring, forest fire and flood detection, human health care, and structural health monitoring are only a few examples which have one thing in common: the absence of wired infrastructure. This fact is typical for all WSNs and causes the necessity of a wireless communication and an autarkic energy supply of each sensor node.

The autarkic energy supply raises the problem of limited energy and power constraints. Basically, there are two possibilities to supply sensor nodes. First, primary cells (non-rechargeable batteries) can be used. Their advantage is the predictability of the sensor nodes' period of operation, assuming that the average power consumption is known. A disadvantage is the limited capacity of the batteries due to the limited physical size of a sensor node given by its application area. Second, energy harvesting systems (EHSs) can be employed for supplying sensor nodes. They consist of an energy harvesting device (e.g. solar cells), an energy storage component, and a circuitry which connects them. EHSs convert environmental energy into electrical energy and store it for periods with insufficient harvestable energy. The advantage is the possibility of a perpetual operation under certain constraints. However, the already mentioned restriction of the physical size of a sensor node for a specific application also causes a limitation of the harvestable power.

In the case of supplying the sensor nodes by primary batteries, the limitation of the stored energy is the reason for a preferably low average power consumption of the sensor nodes to extend the period of operation as long as possible. In the case of supplying the sensor nodes by EHSs, the limitation of the harvestable power causes the necessity of well-dimensioned energy storage components and also a limitation of the sensor node's average power consumption. However, in almost all application areas of WSNs, a manual exchange or recharging of the batteries is disproportionately time-consuming and expensive. The reasons are the typically high amount of sensor nodes and their location difficult to access. Therefore, each sensor node of the WSN has to be as energy efficient as possible.

The development of energy-efficient WSNs is difficult because hardware and software always interact. This means that the behavior of the software may change the state of the hardware, and the state of the hardware may alter the behavior of the software. Energy harvesting WSNs imply further interactions. On the one hand, they interact with their environment by means of measurements and energy harvesting. The harvestable energy depends on the available energy of the environment. On the other hand, their EHSs require energy management to enable a long or perpetual operation which itself causes

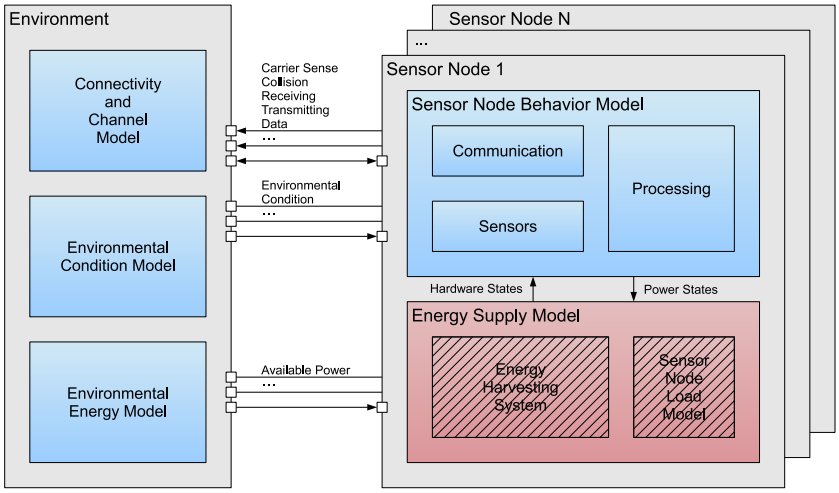


Figure 1: Overview of the implemented simulation environment. The non-shaded modules are implemented in SystemC and the shaded modules in SystemC-AMS.

local interactions of the sensor node's units. Therefore, an analysis of the hardware-software interaction in energy harvesting WSNs improves their optimization. Thus, this analysis enhances the development and operation of such WSNs.

The analysis of hardware-software interactions requires a methodology to find out how the subsystems interact with each other. Basically, there are three different possibilities: the analytical methodology, the empirical methodology, and the simulation methodology. Although all three of them can be utilized for analyzing hardware-software interactions, the simulation methodology provides the highest flexibility. Therefore, this methodology has been selected.

This work presents a simulation environment developed in SystemC-AMS. The simulation environment supports a functional simulation of the software and an analog simulation of the energy supply [1, 2] in order to analyze energy-related interactions between software and hardware. Figure 1 gives an overview of the simulation environment. It comprises an environment model which connects the nodes with each other and provides them with location-specific information, and a sensor node model consisting of a sensor node behavior model and an energy supply model.

The sensor node behavior model is defined by its application software and the access to its hardware modules. Here, the hardware modules are represented by high-level models which reduce the computational complexity. This means that the hardware is modeled functionally, especially the communication module, the sensor modules, and the processing module. The current consumption of these hardware modules is modeled by using power states inside the sensor node behavior module to ensure a realistic power trace.

The energy supply model consists of the energy source model and the load model which simulates the current consumption of a sensor node according to the power state provided by the sensor node behavior model. Different types of energy sources are supported by exchanging the energy source model. In this work, a battery model and an EHS model have been implemented. The hardware of an EHS can be structured into different tiers with special functionalities: the measurement tier, the power control and conditioning

tier, the storage access tier, and the energy storage tier [3]. These tiers can be adapted to each other to enhance the overall efficiency. Also, individual tiers can be easily exchanged to modify the behavior of the EHS. In order to enhance the accuracy of the simulation results, realistic models of the energy harvesting devices are used which are generated by using an on-site characterization instrument [4]. These realistic models also enhance the design process of energy harvesting WSN since the energy supply can be adapted to specific application scenarios.

The interaction between the software and hardware is ensured by the connection of the sensor node behavior model and the energy supply model. Two case studies, which are described in the following, show the effect of this interaction.

First, a low-power technique called component-aware dynamic voltage scaling (CADVS) is evaluated [5, 6] and simulated [2]. It combines the power-down of unused components (e.g. sensors and communication module) and the minimization of the supply voltage. Typically, each component of the hardware has its own supply voltage range. Therefore, the minimum allowed supply voltage depends on the activated components. These components and consequently the minimum allowed supply voltage vary over time. CADVS uses variable voltage converters to adjust the supply voltage of the hardware to save as much energy as possible. It has been shown that the energy savings depend on the leakage and quiescent currents of the voltage conversion circuit with respect to the supply current of the sensor node. Typically, the leakage and quiescent currents of the converter circuits strongly influence the energy savings at a low supply current. Therefore, the best converter circuit depends on the application running on the mobile device or WSN node. This means that the behavior of the software is a decisive factor for the selection of the optimal hardware.

Second, a measurement error of a sensor node's state-of-charge may affect the whole network behavior and reduce the network lifetime [7, 8]. Todays protocols and applications for WSNs are often energy aware. However, the state-of-charge (SoC) estimation of the energy storage component (e.g. rechargeable battery) influences the decisions of the energy-aware software. This may cause a wrong distribution of the workload or network traffic to specific nodes with a measurement error. Thus, the average power consumption of these nodes is increased which results in the reduction of the network lifetime. This means that the hardware influences the behavior of the software.

These two case studies illustrate that the analysis of the hardware-software interactions actually enhances the optimization of WSNs. Due to this optimization, the period of operation can be extended and the maintenance can be reduced. The implemented simulation environment for WSNs enables a concurrent simulation of the software and energy harvesting hardware at circuit level by using a single modeling language. This new approach of simulating energy harvesting WSNs enables a realistic analysis of hardware-software interactions.

Contents

List of Figures

List of Tables

Acronyms

ADC	Analog-to-Digital Converter
AMS	Analog-Mixed-Signal
AODV	Ad-Hoc On-Demand Distance Vector
CADVS	Component-Aware Dynamic Voltage Scaling
CPU	Central Processing Unit
DC	Duty Cycle
DES	Discrete Event-Based Simulation
DLC	Double Layer Capacitor
DSDV	Destination-Sequenced Distance-Vector
DSR	Dynamic Source Routing
DVS	Dynamic Voltage Scaling
EEPROM	Electrically Erasable Programmable Read-Only Memory
EH-WSN	Energy Harvesting Wireless Sensor Network
EHS	Energy Harvesting System
ICT	Information and Communication Technology
ID	Identification Number
LiIon	Lithium Ion
LSB	Least Significant Bit
MAC	Medium Access Control
MPPT	Maximum Power Point Tracker
NiMH	Nickel-Metal Hydride
OS	Operating System
RF	Radio Frequency
RFID	Radio Frequency Identification
SMAC	Sensor Medium Access Control
SoC	State-of-Charge
SPICE	Simulation Program with Integrated Circuit Emphasis
TMAC	Timeout Medium Access Control
WSN	Wireless Sensor Network

1 Introduction

Acquiring information, information processing, and communication are key developments of our modern society which are known as information and communication technology (ICT). There are two basic components of ICT. The first one is a node which collects, processes, and stores data. The second one is a connection that enables an exchange of information between the nodes. The connected nodes establish networks for distributed information processing and storage. On a large scale, global networks are used for connecting computers and electronic devices across the globe to satisfy our need for information. Due to the increasing amount of data, these large networks are equipped with fast broadband connections and energy-demanding server nodes.

On a smaller scale, wireless sensor networks (WSNs) provide a digital interface with the environment and enable ubiquitous computing in application areas without wired infrastructure. Examples are environmental monitoring [9], precision agriculture [10, 11], wildlife monitoring and tracking [12, 13], human health care [14, 15], structural health monitoring [16, 17, 18], and building automation [19, 20]. In such application areas, an installation of infrastructure is too time-consuming, expensive and in some cases completely impossible. WSNs are a typically cheap and an easy solution to address this problem.

1.1 Wireless Sensor Networks

WSNs consist of small sensor nodes, each of them equipped with sensors to measure physical quantities of their environment. The measured data can be preprocessed and stored at these nodes. Each node is additionally equipped with a wireless communication module to transmit or receive data, status, or command messages. The sensor nodes form a meshed network topology, and within the network, most messages are routed in a multihop manner to a base station which collects the data of all sensor nodes. The advantage of such a network structure is that the covered area can be easily extended by adding more sensor nodes. Thus, WSNs are suitable for covering large areas with minimal effort. However, the connectivity of the network and the network lifetime may depend on the operational capability of individual nodes and a sufficient wireless connectivity between them. In some application areas, especially in industrial applications, interferences cause sudden communication problems. Therefore, reliable WSNs must be fault tolerant in case of a spatially and temporally limited loss of communication links or node failures.

The absence of wired infrastructure causes another problem: the energy supply. Each sensor node needs a dedicated energy supply, which is discussed in the following section.

1.1.1 Energy Supply of Wireless Sensor Nodes

A sufficient energy supply is a fundamental requirement for the operability of a sensor node. Therefore, and due to the meshed network structure, the network lifetime may

depend on the sufficient energy supply of individual sensor nodes [21]. Basically, there are two different possibilities to supply a sensor node. First, it can already be powered with energy when being installed. Primary cells (non-rechargeable batteries) are typically used in such cases. Second, the energy of the environment can be utilized to power a sensor node. So called energy harvesting systems (EHSs) are developed for this purpose. Both methods including their advantages and disadvantages are described in the following.

1.1.1.1 Battery-Powered Wireless Sensor Nodes

Primary cells provide a certain amount of initial energy. They are used for sensor nodes with an a priori determined period of operation. This amount of time is mainly defined by the capacity of the primary cell and the average power consumption of the sensor node. However, the period of operation cannot be increased indefinitely by reducing the average power consumption of the sensor node. The reason is the leakage current of the primary cells [22]. If the average current drained to supply the sensor node is smaller than the leakage current, the period of operation is mainly defined by the latter.

The advantage of primary cells is the continuous energy supply. If the period of operation defined a priori is sufficient, the supply of sensor nodes using primary cells guarantees a continuous operation. For this, however, the knowledge of the expected average power consumption of the sensor node is required.

However, if the primary cells have to be replaced to extend the period of operation, the maintenance effort of the WSN increases significantly. The reason is that a WSN may consist of thousands of nodes [23] and some nodes may be installed at hard-to-access locations [24]. This is called the battery replacement problem [25].

1.1.1.2 Energy Harvesting Wireless Sensor Nodes

Energy harvesting WSNs (EH-WSNs) use EHSs to supply the sensor nodes. The EHSs convert environmental energy into electrical energy. Depending on the application area, different energy sources can be used. Solar radiation, thermal gradients, or vibrations are often utilized to supply sensor nodes [26, 27, 28]. A well-designed EHS extends the period of operation or even enables a perpetual operation. However, the temporal variation of the harvestable energy is typically significant. Therefore, a certain amount of the harvested energy must be stored to allow a continuous supply of the sensor node [29]. Secondary cells (rechargeable batteries) or supercapacitors – or a combination of both – store energy for periods with insufficient harvestable energy [27, 30].

1.1.1.3 Limitations of Available Power and Energy

Depending on the application area of the WSN, the size of the sensor nodes is often limited. Originally developed for military use, WSNs should be deployed on enemy territory to detect the opponent's movements. Therefore, the sensor nodes are intended to be as small as possible, ideally as small as a mote. Thus, a sensor node is sometimes called "mote". Other applications also require rather small sensor nodes. One of the smallest developed energy harvesting wireless sensors is presented in [31]. It is intended to measure the intraocular pressure of the human eye. Therefore, the volume of the sensor is limited

to 1.5 mm^3. It is powered by two integrated solar cells and consumes an average power of 5.3 nW at an activation interval of 15 minutes.

Due to the restricted size of a sensor node, the capacity of its primary cell is also limited. This restricts the available energy for supplying a sensor node. That is why it must be as energy efficient as possible to optimize the period of operation. In the case of EH-WSNs, the limited size restricts the harvestable power and the storable energy of EHSs. Here, the energy efficiency of the sensor node is important to enable a perpetual operation without interruptions. Therefore, energy management and awareness are important for WSNs, which is discussed in the following section.

1.1.2 Energy-Aware Services and Applications

Energy awareness is important for energy-efficient WSNs. The behavior of these WSNs is adapted to energy-related parameters, for instance the remaining stored energy. Therefore, the nodes are able to omit unnecessary tasks in order to extend the period of operation or enable perpetual operation.

Energy-aware routing [21, 32, 33, 34, 35] distributes the network traffic between different sensor nodes. Otherwise, the sensor nodes along the best route will be loaded with too much traffic and the lifetime of these sensor nodes will be reduced. The excessive network traffic at nodes near a base station compared with marginal nodes is called the energy hole problem [36, 37]. This difficulty can be mitigated with energy-aware routing.

Energy-aware workload distribution [38, 39] is also used for extending a WSN's period of operation. Specific tasks are assigned to sensor nodes with sufficient available energy. If the tasks are location specific, the distribution is only possible to nodes inside a region of interest with similar environmental conditions.

In the case of EH-WSNs, it is also possible to adapt the behavior to the harvestable energy [38]. For example, high power peaks can be used for performing energy-demanding tasks.

1.2 RiverMote - A Motivating Example

The RiverMote project [40, 41, 42, 43] aims at monitoring a river's water level by using EH-WSNs. The sensor nodes are fixed to the inside of a moored buoy, and each sensor node can be equipped with a GPS receiver, an ultrasonic sensor, and a pressure sensor to measure the water level. The purpose of the WSN is to use it as an early warning system for dangerous floods. The structure is shown in Figure 1.1. Such a WSN is a good example for installing the sensor nodes spread across a wide area in hard-to-access locations. A replacement of the batteries is time consuming and expensive. Therefore, each sensor node is equipped with a solar cell and two supercapacitors to supply it continuously. The project has not been finished yet and only one prototype has been implemented so far. However, measurements from this prototype show the necessity of energy-aware implementation because of bad weather conditions during winter. Hardware and software were adapted to each other to optimize the operation.

The analysis of the hardware-software interaction in this EH-WSN would further enhance the optimization process. By using a simulation, the operating mode of a multitude of sensor nodes can be estimated without their implementation and deployment. Possible

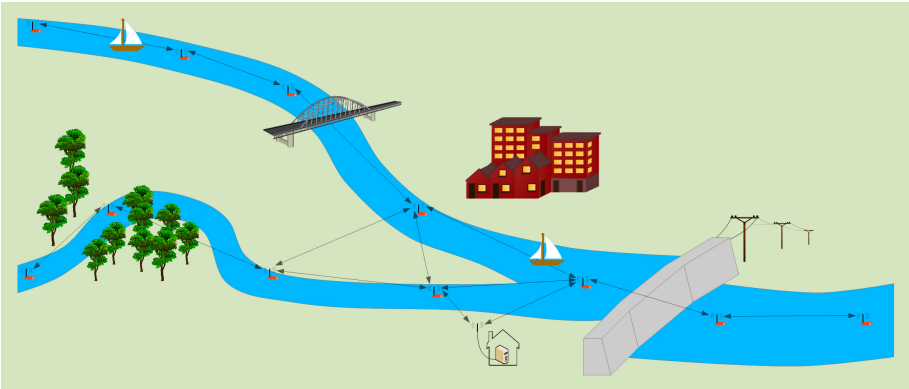


Figure 1.1: RiverMote WSN application scenario for early flood warning. Energy harvesting sensor nodes are fixed to the inside of a moored buoy. The information of the water level is transmitted to the base station in a multi-hop manner.

problems can be detected prior to the deployment, which avoids extensive maintenance. Therefore, the analysis and simulation of hardware-software interactions may save a lot of time and costs.

1.3 Wireless Sensor Networks as Interacting Systems

To be able to discuss WSNs as interacting systems, the terms system and interactions have to be explained. As described in [44], systems are defined by their internal structure and external behavior. The latter is defined by the input and the corresponding output of the system. Furthermore, systems theory deals with the combination of subsystems to larger systems and with decomposition of systems into individual components. Therefore, each system can be composed of individual elements interacting with each other. Such systems can be used for describing real systems at a certain abstraction level. The models of real systems are important to predict the behavior by simulation.

System descriptions are used in different application areas. Biology [45], economics [46], environmental simulation [47], home automation modeling [48], network analysis [49], and road traffic analysis [50] are only a few examples. WSNs can also be described as systems [51, 52]. Each sensor node of a WSN consists of different components interacting with each other to perform certain tasks. The sensor nodes also interact with their environment by measurements and energy harvesting as well as with each other by an exchange of information.

1.3.1 Hardware-Software Interactions

Each sensor node consists of hardware components defining the physical structure, and software which defines the behavior. The combination of hardware and software leads to interactions between them. Basically, the hardware-software interactions can be divided into two types which are described in the following.

1.3.1.1 Effects of Software on Hardware

Software accesses different hardware components while being executed, for instance registers, memory, timers, sensors, or communication modules. These accesses cause a temporal variation of the power consumption of the sensor node. Therefore, the execution of software changes the state of the hardware. The software can actively conserve energy by using efficient energy management strategies, and it can also change the way how a sensor node interacts with its environment, for instance by activating or deactivating sensors or energy harvesting.

However, the active control of hardware components may also be a disadvantage in case of software failures or problems. If critical control operations cannot be performed, hardware components may be damaged. For example, overcharge or undercharge of sensitive energy storage components may be triggered by faulty software.

1.3.1.2 Effects of Hardware on Software

The hardware itself cannot change the behavior of the software since the first mentioned usually does not change during the operation. However, the internal state of the hardware can influence the behavior of the software. As mentioned before, energy-aware services and applications adapt their behavior to the information of the internal state of the hardware. Each energy-aware sensor node requires sensors to measure the energy-related internal states of the hardware. An example is the state-of-charge (SoC) of the energy storage component.

Hardware failures may disrupt a continuous operation of a sensor node which, therefore, is required to detect failures of specific hardware components itself. Self-healing sensor nodes [53] are able to solve uncritical hardware failures, for instance by deactivating software components that use the faulty hardware components. Information about the health status influences the behavior of the sensor node's software in order to deactivate the relevant components if possible.

1.3.2 Interactions in Energy Harvesting Wireless Sensor Networks

A correct operation of an EH-WSN depends on the available environmental energy. Therefore, strong interactions exist between the environment and the sensor nodes. The behavior of the WSN must be adapted to the local energy conditions to guarantee a perpetual operation. Energy-aware services and applications are important components of these WSNs.

EHSs are complex systems which implicate further energy-related interactions. Measurements of the input power, the output power, or the stored energy are a part of such interactions. Typically, voltage and current measurements are used for determining the power values. As shown in [7] and [8], measurement errors influence the behavior and the lifetime of a WSN, which is discussed later in Section 4.2.

1.4 Simulation of Wireless Sensor Networks

Simulation is typically used for analyzing the behavior of WSNs, and numerous survey papers are available on this matter [54, 55, 56, 57, 58, 59, 60, 61, 62]. It is an easy way to

study different software implementations without programming and deploying the sensor nodes. Due to the fact that the application areas are very different, the simulation of a sensor node must be adaptable to them. Also the simulated hardware and software modules have to be exchangeable. Then, a flexible simulation of different hardware configurations with different software at various environmental conditions is possible.

The simulation of the energy consumption of a sensor node is necessary to estimate the network lifetime. Each sensor node is assumed to have a certain amount of initial energy. First simulation environments aimed at the analysis of different routing protocols regarding this lifetime [21, 63]. A method often applied for simulating the energy consumption is the use of power states which are defined by the active hardware components at a particular time [64, 63, 65].

The simulation of a WSN can be used for its optimization during the following phases of development:

Design: A rough simulation can be employed to estimate system-critical parameters, for instance network lifetime or reliability. The results can be used for a feasibility study.

Implementation: Simulation results can be utilized to optimize the implementation of the WSN's hardware and software.

Verification: Testing of the system and of individual modules can be supported by using a simulation environment. A final test with real sensor nodes only needs to be executed if the implementation proves successful in the simulation.

Maintenance: A simulation parallel to the deployed WSN may show anomalies or problems during the operation.

1.4.1 Combined Simulation of Software and Hardware

As mentioned before, an energy-related analysis is important for WSNs and even more significant for EH-WSNs. Due to the energy-related interactions in WSNs, a separated simulation is not possible. Therefore, a combined simulation environment is necessary to analyze the energy-related behavior. It is important that the results of the hardware simulation at a particular time are immediately forwarded to the software simulation which may adapt their behavior. Also the results of the software simulation have to be passed on to the hardware simulation instantaneously to correctly change their state. Ideally, it happens in the same simulation step.

1.4.2 Simulation-Based Enhancement of Energy Harvesting Wireless Sensor Networks

EH-WSNs suffer from being strongly dependent on the harvestable energy from their environment. A meaningful simulation can be used for optimizing the sensor nodes according to their application. However, a good knowledge and models of the harvestable energy are necessary. The hardware as well as the software can be adapted to the environmental conditions. Different modules can be simulated by using the same harvestable energy; therefore, the results are comparable and the best modules can be selected.

1.5 Thesis Organization

The remainder of this thesis is organized as follows: Chapter 2 outlines related work concerning WSN applications, WSN architecture, simulation, and interactions. Finally, it lists the main contribution of this thesis. Chapter 3 discusses interactions, especially hardware-software interactions in EH-WSNs. Furthermore, it introduces the architecture of the simulation environment which is used for analyzing these interactions. Chapter 4 shows two case studies as an application of the introduced simulation environment. First, component-aware dynamic voltage scaling illustrates the dependency of the optimal hardware selection on the behavior of the software. Second, a state-of-charge measurement error analysis shows the effects of hardware state changes on the behavior of the WSN. Chapter 5 concludes this thesis and gives directions of future work. Chapter 6 finally lists the contributed publications of this thesis.

2 Related Work

This chapter introduces related work which is important for this thesis. WSNs were investigated a lot in the past and they are objects of research in the present. First, deployed EH-WSNs of different application areas are shown. Second, the common WSN architecture which consists of the sensor node's hardware and the local and distributed software is explained. Third, existing WSN simulators and models are investigated. Fourth, related work of hardware-software interactions in EH-WSNs is discussed. At the end of this chapter, the contribution of this thesis is outlined.

2.1 Application Areas of Energy Harvesting Wireless Sensor Networks

As discussed in Chapter 1, there are numerous different application areas of WSNs. Generally, they can be divided into two main groups: tracking and monitoring [66]. Tracking is used for capturing the temporal change of an object's position, and the application areas range from product to animal and human tracking. Monitoring is used for apprehending the temporal change of an object's physical characteristic. Here, the objects are typically static, and the application areas range from environmental to machine and structural monitoring. However, there are also application areas which combine monitoring and tracking.

Most of them are characterized by the lack of wired infrastructure. One of the first EH-WSN is the ZebraNet [12]. The sensor nodes are put on zebras, each of them equipped with a GPS receiver to track the animals. Solar cells are used for charging a lithium-ion (LiIon) battery which is needed to supply the sensor node during night and for peak power demand (during data transmission and GPS position fix). A supply of the sensor nodes using primary cells is not possible due to the high energy demand of the GPS receiver. As mentioned in the paper, a 10 kg battery would be needed to supply the nodes for one year without energy harvesting. This project uses a customized hardware specially developed to fulfill the specific needs.

The Heliomote project [27] introduces a solar EHS which can be installed on a Mica2 hardware platform. This EHS stores the harvested energy in nickel-metal hydride (NiMH) rechargeable batteries. A direct connection of the solar cells to the batteries ensures an operation point near the maximum power point of the solar cells. A diode prevents a discharging of the battery through the solar cell during periods with insufficient light. The inquiry of the EHS status information was integrated into the operating system of the sensor node. Therefore, an application program can easily retrieve this information.

The Prometheus project [30] enhances the previously introduced EHS by using a supercapacitor in combination with a LiIon rechargeable battery. This combination prevents the LiIon rechargeable battery from short-term charge and discharge cycles. Only if the

harvestable energy is too low for a longer period of time, the battery is discharged. This enhances the lifetime of the LiIon battery. The authors of this paper have also integrated a temperature-aware charging algorithm because the LiIon battery's terminal voltage at a certain SoC varies with the temperature. The EHS is designed to be used with the TelosB sensor node.

The AmbiMax platform [67] combines different energy harvesting sources to maximize the harvested energy. In the paper, the authors present the combination of a wind generator and a solar panel. Each of the input channels has a maximum power point tracker (MPPT) which enhances the harvestable energy. The MPPT operates autonomously and is independent of digital control, which is preferable for a cold start (battery not charged). This energy harvesting platform is intended to supply different types of sensor nodes.

The authors of [68] evaluated the possibility of supplying sensor nodes inside a traffic tunnel by using vibration energy harvesting. They show that it is possible to supply a transmitter that sends a message every time a train passes the sensor node. The system is intended to monitor the structural health of the road tunnel.

The EH-WSN presented in [69] is meant to monitor overhead power lines. It harvests the energy from the electrostatic field caused by the high voltage of the power lines. The authors of the paper achieved a power output of 16.3 mW which is enough to supply a sensor node. These can be included into a system to protect the overhead power lines from destruction resulting from overload.

The authors of [70] present an EH-WSN to monitor aquatic environmental parameters. Underwater luminosity and temperature should be tracked over a longer period of time. To enable solar energy harvesting and radio communication, the electronic is fixed to the inside of a moored buoy. Intelligent energy storage architecture prevents the batteries from early degradation. Two identical batteries are alternately charged and discharged; while one battery is used for supplying the sensor node, the other battery is charged by the solar cells. Once the supplying battery is empty, it will be charged again and the other battery supplies the sensor node. This process enhances the lifetime of the battery. The authors of the paper demonstrate a continuous operation of charging and discharging over a span of four days.

These examples and application scenarios show the successful supplying of wireless sensor nodes by environmental energy. EH-WSNs reduce maintenance effort by extending the period of operation and may avoid the battery replacement problem.

2.2 Wireless Sensor Network Architectures

As already introduced in Chapter 1, a WSN consists of a certain number of sensor nodes, and each WSN can be classified into homogeneous and heterogeneous networks [23]. Every single sensor node of homogeneous WSNs has the same hardware structure and software and can only be distinguished by an identification number (ID). The advantage of these WSNs is the simplified software development since only one version has to be implemented. Heterogeneous WSNs consists of sensor nodes with different software and hardware, each sensor node being designed to fulfill a certain task, for instance the sensor nodes are equipped with different sensors. Since interactions in such WSNs are more difficult to analyze, these WSNs are not discussed in this thesis.

Another differentiating factor is the number of base stations or gateway nodes. Most WSNs use only one base station to collect the data from the network. However, there are also solutions with multiple base station deployments for redundancy or better energy efficiency [71, 72, 73].

Based on the location of the base station(s) and sensor nodes, and the transmission range of the communication modules, a certain network topology is defined. Without multi-hopping, merely a star topology is possible where every sensor node is in transmission range of the base station [74]. Here, an extension of the covered area is only achievable by adding further base stations. However, this topology is very simple because no routing protocol is necessary. Most WSN implementations use multi-hop routing protocols to forward messages to the base station(s). Therefore, the covered area can be extended by adding further sensor nodes to form a mesh topology.

Only if a mesh topology is given, individual nodes interact with each other. This fact implies energy-related interactions in EH-WSNs. Thus, only these topologies are interesting for an investigation. As mentioned in the introduction, energy-aware routing protocols change the routes to the base station according to the energy state of individual sensor nodes. In large-scale WSNs, clustering is used for optimizing communication. The clustering can be performed in an energy-aware manner [75] to extend the network lifetime.

The sensor node architecture consists of its hardware and software and is described in Section 2.2.1 and Section 2.2.2 respectively.

2.2.1 Wireless Sensor Node Hardware

A sensor node typically consists of four main units which are the sensor unit, the processing unit, the communication unit, and the energy supply unit [76]. A generic structure of such a sensor node is shown in Figure 2.1.

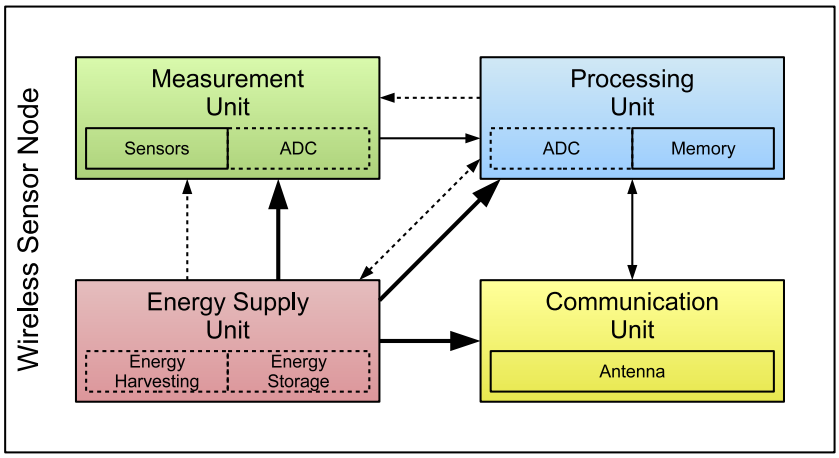


Figure 2.1: Generic structure of a wireless sensor node showing the four main building units adapted from [76]. The dashed subunits and the dashed arrows are optional. The thicker arrows represent the typical power flow and the thinner arrows represent the typical information flow.

2.2.1.1 Sensor Unit

The sensor unit consists of different sensors which are necessary to measure physical quantities of the environment. Furthermore, specific sensors can be used for determining the internal state of the sensor node which is important for energy-aware software. Some of the sensors have an integrated digital data processing system, for instance an integrated analog-to-digital converter (ADC) for an easy access of the measurement data.

2.2.1.2 Processing Unit

The processing unit is responsible for executing the programmed software. During the execution, measurement data is captured from the sensor, processed, and stored or transmitted by using the communication module. This unit typically consists of a microcontroller which is the central processing unit (CPU) of a sensor node. Modern microcontrollers have integrated peripherals, for instance a memory for software (usually an electrically erasable programmable read-only memory (EEPROM)) or an ADC.

2.2.1.3 Communication Unit

The communication unit enables an information exchange between the sensor nodes and is required to form a sensor network. Many sensor nodes such as the Telos sensor node [77] have integrated antennas to minimize the form factor. This unit is responsible for a correct access of the transmission medium (medium access control (MAC)). Typically, radio frequency (RF) communication is used. However, there are other types; for example, underwater WSNs often utilize acoustic communication [78, 79, 80].

2.2.1.4 Energy Supply Unit

The energy supply unit is responsible for the supply of the entire sensor node with electrical energy. It usually consists of an energy reservoir to store the energy. Primary cells (non-rechargeable batteries), secondary cells (rechargeable batteries), or supercapacitors can be used. Furthermore, the energy supply unit should guarantee safe supply conditions for the sensor node. This means that the energy storage component is prevented from overvoltage, undervoltage, or other improper operating conditions. Some sensor nodes are constructed for a constant supply voltage which necessitates voltage converters for stabilization.

2.2.2 Local Wireless Sensor Node Software

Due to the fact that a sensor node is resource constraint, only light-weight software can be executed. The software is often designed especially for a certain application to be efficient in terms of memory, communication, and energy. However, an abstraction of the hardware is meaningful to enhance the software development process. Hardware abstraction is commonly used in computer architecture to enable an easy access to hardware devices [81]. In [82], the authors present a hardware abstraction architecture which is applied to WSNs. They divide the abstraction into three sublayers. First, the hardware presentation layer defines interfaces with hardware by using memory, registers, or ports. Second, the hardware abstraction layer introduces new functionalities by using the first layer. It is needed for providing a common third layer for the application software: the hardware interface

layer. This layer is used by the application software to access hardware devices and provides a general interface to be able to execute the same application software on different hardware platforms.

Such hardware abstraction layers can also be utilized for simulation environments [83, 84]. A defined interface which is provided by the real sensor node as well as by the simulation environment can execute the same application software. Therefore, the results are comparable and the simulation environment can be used efficiently to estimate the real behavior of the software.

2.2.2.1 Operating Systems for Wireless Sensor Networks

In addition to hardware abstraction, operating systems (OSs) provide further functionalities, for instance interrupt handling or task scheduling. As mentioned before, sensor nodes have limited resources and need light-weight OSs. Furthermore, heterogeneous WSNs suffer from incompatible software [85]. An OS for WSNs should also be able to handle different hardware platforms.

TinyOS [86] is such an operating system. The software architecture is event driven and the hardware is abstracted by using the three layers mentioned above. This OS can be used for different hardware platforms. Therefore, application software can be compiled for the supported platforms. To reduce the size of the executable, only the needed OS components are included. The software is written in nesC [87] which is a nested and component-oriented programming language. Further OSs for WSNs are Contiky [88] or Mantis [89].

Energy-aware mechanisms are essential for WSN software because of the limited resources, here in terms of energy. Duty cycling is a common method applied to save energy. Mostly, the sensor node remains in energy-saving sleep state. It is only activated for measurement, processing and communication. However, during sleep state, no events can be detected by the sensor node, and the real-time behavior is affected. Therefore, the applicability depends on the application area, but most environmental monitoring applications use this mechanism. Such energy-aware mechanisms are typically integrated into the OS and can be easily accessed by the application software.

2.2.2.2 Local Power Management with Energy Harvesting

Energy harvesting changes the energy constraints of sensor nodes. Depending on the size of the energy storage component, the available energy is given by the history of the harvestable energy. Therefore, the behavior of the software should be changed accordingly [38]. For example, the duty cycle (DC) of the sensor node can be adapted to the harvestable energy. Such mechanisms related to energy harvesting imply energy-related hardware-software interactions which are important for this thesis.

2.2.3 Distributed Wireless Sensor Network Software

In contrast to the local WSN software, the distributed software is necessary for communication inside and management of the WSN. Communication is important to extract information from the WSN. Routing algorithms [33] find the way through the network.

To guarantee a certain robustness in harsh environments, the routing protocols for WSN must be fault tolerant.

The management of a WSN consists of different tasks, the most important ones being described in the following. First, synchronization is used for preventing collisions inside the network and to keep the communication link between duty cycling sensor nodes [90]. Duty cycling implies a temporal limited activation of the communication unit. Only if all sensor nodes activate their communication unit simultaneously, messages can be forwarded through the entire WSN. Second, task distribution is necessary if a certain task can be executed by different sensor nodes. Therefore, one sensor node is selected according to a metric, for instance available energy, sensor node location or environmental conditions. Third, topology control is used for reducing the energy consumption by keeping the connectivity of the entire sensor network [91]. The transmission power of the communication unit is minimized according to the distance of a certain number of neighbors. Fourth, as already mentioned, clustering structures the network into subregions to optimize the communication. All of these mechanisms introduce interactions between sensor nodes. In particular, the duty cycling is analyzed in a paper of this thesis [8].

2.2.3.1 Middleware for Wireless Sensor Networks

Middleware is a collective term for different distributed software architectures. It abstracts not only the hardware of one node but also the entire sensor network. The previously mentioned methods of communication and management of WSNs are combined to a single middleware solution. The application software can use this middleware to access the sensor node without being aware of single nodes. There are different types of middleware which are used for programming support [92]. The authors divide middleware into five subclasses which are virtual machines, database approaches, modular (agent-based) approaches, application-driven and message-oriented middleware. The PhD thesis of Glatz [43] introduces a network coding middleware for EH-WSNs. Network coding is integrated into the distributed software to optimize wireless communication.

Although all of these middleware architectures enhance the development of application software for WSNs, they introduce many interactions. Therefore, the analysis of specific interactions is more complicated. Furthermore, the simulation of these frameworks is only possible with an instruction-level simulator or a detailed model of the entire middleware, both being rather complex tasks. However, to analyze energy-related interactions, no complex frameworks are necessary; small programs are enough for analyzing specific interactions as done in this thesis.

2.2.3.2 Distributed Energy Management

Distributed energy management is important for resource-constraint WSNs, especially EH-WSNs. Distributed energy-aware mechanisms are typically integrated into the middleware of WSNs. However, they can also be integrated into small WSN applications without a middleware layer. As introduced in Section 1.1.2, energy-aware routing and workload distribution are key mechanisms for energy management. One goal of the distributed software is to keep the network alive as long as possible. Therefore, the sensor nodes exchange information about their energy states. These mechanisms implicate important energy-related interactions. Therefore, they are of major significance for this thesis.

2.3 Simulation of Wireless Sensor Networks

As mentioned in [59], simulation is an effective tool for evaluating the behavior of WSNs during the development. Other methods are possible but they are too complex, time consuming, or expensive, which is discussed later in Section 3.1.3. There are five basic entities for simulation as described in [44]: the source system, the behavior database, the experimental frame, the model, and the simulator. The model and the simulator are used for estimating the behavior of the source system under certain constraints given by the experimental frame and the behavior database. They are the key parts for simulation and described in Section 2.3.1 and Section 2.3.2 respectively.

2.3.1 Simulators for Wireless Sensor Networks

Simulators estimate the behavior of WSNs in order to analyze different characteristics. The requirements are manifold and depend on the focus of the simulation. The key requirements for a simulation of WSNs as listed in [54] are (i) *reusability and availability*, (ii) *performance and scalability*, (iii) *support for rich-semantics scripting languages to define experiments and process results*, and (iv) *graphical, debug and trace support*. [60] also lists *fidelity*, *energy awareness*, and *extensibility*.

Similar to [60], the following subsections try to classify WSN simulators according to their focus of simulation into four main classes: network system simulators, node system simulators, instruction-level simulators, and hardware-software co-simulators.

2.3.1.1 Network System Simulators

Network system simulators focus on the network communication of the evaluated WSN. The network is treated as a system consisting of nodes and the communication between them. Typically, they are based on discrete event-based simulation (DES) [93]. As the name implies, the activity inside the WSN is modeled by using events, for instance the exchange of messages [57]. Examples for this type of simulator are ns-2 [94] or Omnet++ [95].

Such simulators have a good scalability and performance. Although energy consumption is simulated, electronic circuit models for enhancing accuracy are hard to integrate.

2.3.1.2 Node System Simulators

Node system simulators put the focus on node simulation and integrate the network support. TOSSIM [96] is able to simulate the same software which can be executed on a real sensor node. The event-driven structure of the emulated operating system TinyOS [86] is used for an event-driven simulation. Further node system simulators are written in SystemC [97, 98, 99, 100, 101] which is based on DES and often used for modeling entire systems consisting of software and hardware. However, the focus of the related work lies mainly on communication or power consumption without energy harvesting. The simulation framework presented in [102] uses SystemC to evaluate different power management approaches for WSNs. The authors also envision a simulation of energy harvesting. However, they modeled the hardware components by using SystemC. Therefore, no circuit

level simulation is possible. Furthermore, they do not present any results, and further work by them was not found prior the publication of this thesis.

An analog-mixed-signal (AMS) extension to SystemC is SystemC-AMS. Systems can be composed of hardware components, including capacitors and resistors, which are connected in order to form electrical circuits. Therefore, it is possible to model complex hardware systems such as EHSs. SystemC-AMS is used for modeling and simulating WSNs, usually by focusing on transceiver circuits and communication [103, 104, 105], or ADCs [106].

The advantage of using SystemC and the extension AMS is the flexibility of the detailedness of the model. Therefore, a specific focus of the simulation (e.g. energy consumption) can be achieved easily. The detailedness of other parts can be reduced to enhance the simulation performance.

2.3.1.3 Instruction-Level Simulators

Instruction-level simulators focus on an accurate simulation of a sensor node's software. Such simulators can execute the same software as a real sensor node. Therefore, they provide high simulation accuracy at a typically lower performance compared to network system simulators. Examples are the ATEMU [107] and the Avrora [108].

2.3.1.4 Hardware-Software Co-Simulators

Co-simulation frameworks combine different modeling and simulation paradigms. For example, the authors of [109] combine SystemC and SPICE (Simulation Program with Integrated Circuit Emphasis). The software is simulated in SystemC and the electrical circuits in SPICE. A disadvantage is the high effort to implement the interface between the different simulators concerning data exchange and synchronization. Another co-simulation framework is presented in the PhD thesis of Janek [110] which is used for evaluating the period of operation of energy harvesting RFID (radio frequency identification) tags. This framework uses a two-step simulation execution. First, the simulation at hardware level is executed to get the power traces of the tag. Second, the simulation at system level evaluates the long-time performance of the tag. However, this prevents the evaluation of interactions between the separated simulation elements.

2.3.2 Simulation Models for Wireless Sensor Networks

Most simulation environments and frameworks include numerous models for communication protocols, energy supply, sensors, and other components. The comparative surveys of [56] and [58] also list the implemented models. The common supported models for communication and energy supply are described in the following sections.

2.3.2.1 Communication Models

Network system simulators focus on this kind of models since the communication should be as realistic as possible. Starting at the physical layer, the wireless channel can be simulated by using different models. Most common wireless channel models are free space, two-ray ground reflection and shadowing [56]. The interface between the wireless channel and the sensor node is the radio chip. Common physical radio chips are emulated which are often

used for sensor nodes, for instance the CC1100 or the CC2420, both being from Texas Instruments [56]. Different MAC strategies are included in the simulator frameworks, for instance 802.11, 802.15.4, SMAC (sensor MAC), or TMAC (timeout MAC) [56], and numerous MAC protocols for WSNs are surveyed in [111]. Different routing protocols are typically included in network system simulators. Therefore, an easy comparison of the protocols is possible. Examples are destination-sequenced distance-vector (DSDV), dynamic source routing (DSR), or ad-hoc on-demand distance vector (AODV) [58].

The other classes of simulators usually only implement the wireless channel model and the emulation of the radio chip. By selecting a specific radio chip, the MAC strategy is typically given, and the higher level communication (routing) is integrated in the simulated software.

2.3.2.2 Energy Supply Models

The energy supply models of sensor nodes consist of a consumption (or load) model and an energy storage model. The first one is used for modeling the temporal run of the power consumed by the hardware components of the sensor node. The second one is necessary to model the remaining energy. Most network system simulators implement only primitive energy models. However, there are extensions with better models [58, 112]. Avrora and TOSSIM use extensions to enable the simulation of energy consumption, these extensions being AEON [113] and PowerTOSSIM [64].

Nevertheless, these plugins suffer from the fact that it is hard to integrate customized hardware for the supply of the sensor nodes. The authors of [112] mention that the focus of most simulation environments and frameworks lies on communication. Energy-related simulation is often neglected or only modeled with insufficient detailedness. The authors present an energy-aware structure for WSN simulation which consists of energy source models, energy consumer models, and energy store models. The latter are presented in [114]. However, they do not present the support of circuit level simulation.

Discussion: Simulation of WSNs

To summarize the related work on simulation of WSNs, it can be said that there are numerous simulators which focus on communication between the sensor nodes. The small number of simulators which focus on energy-related simulation suffers from non-circuit level simulation of the electrical components. A simulation environment combining software simulation and hardware simulation at circuit level can enhance the fidelity of a WSN simulation regarding energy-related properties.

2.4 Hardware-Software Interactions in Energy Harvesting Wireless Sensor Networks

When talking about interactions in WSNs, the interactions between the sensor nodes are spoken of in general [115]. The authors of [116] describe three different interaction patterns between sensor nodes. First, hierarchical data collection is used for monitoring environmental parameters at which data is sent to a base station. Second, localized interactions are necessary to detect local events at which neighboring sensor nodes are involved in

decision-making. Third, actuation driven by sensing uses wireless sensor nodes to modify environmental parameters with actuators. Control loops are introduced and cause a strong interaction with the environment.

The authors of [117] present a framework to simulate virtual sensor nodes interacting with real sensor nodes. They also mention the interaction between sensor nodes and their environment.

However, there are also other interactions related to WSNs. This thesis classifies them as follows: external interaction, network level interaction, and device level interaction, as described in detail in Section 3.2.3. The following sections give an overview of related work regarding the different classes.

2.4.1 External Interactions

The authors of [118] describe interactions of the WSN with external networks and devices. They also mention that the middleware of the WSN should support these interactions. A WSN for ubiquitous health monitoring is presented in [119] where the physician can recommend further exercises based on the collected health data of the patient.

2.4.2 Network Level Interactions

There are many interactions at network level. Each time a sensor node transmits a message, it interacts with a certain number of neighboring nodes. Therefore, all kinds of distributed software (as discussed previously) are important at this level. For example, the WSN presented in [120] adapts the information coding rate to the current network conditions.

2.4.3 Device Level Interactions

Device level interactions between the four units of a sensor node enable the data gathering, processing, and transmission. Furthermore, the behavior of the sensor node can be adapted to environmental parameters or a sensor node status. As introduced previously, the authors of [112] present a simulation framework for an energy-aware analysis of WSN including energy harvesting. They also show the interactions between the models and distinguish two types: interactions between the environment and the sensor node, and between components of the sensor node. As already mentioned, the disadvantage of this framework is that the hardware cannot be simulated at circuit level. Hence, the interaction depending on specific circuit components cannot be simulated either. Furthermore, the simulated software consists of an energy stack and a sensing stack, and it is not clear if software not using these stacks can be simulated.

Discussion: Hardware-Software Interactions in EH-WSNs

Usually, all three classes of interactions are essential for the operation of a WSN. Although hardware-software interactions were not analyzed in hardly any of the related works, all of them used them implicitly. Especially energy-related interactions are important for EH-WSNs. The reason is that operability depends on the state of the energy harvesting

circuitry. Therefore, it is necessary to analyze hardware-software interactions including the circuit level of EH-WSNs. This analysis is a contribution of this thesis.

2.5 Contribution of this Thesis

The major contribution of this thesis is the analysis of hardware-software interactions. This examination is enabled by a developed simulation environment for a combined simulation of hardware and software. The contribution is supported by two case studies on hardware-software interactions which show the advantage of the analysis. The following list describes the contribution of this thesis in detail.

Hardware-Software Interactions

A classification for WSN-related interactions is introduced and different hardware-software interactions are presented. Furthermore, the analysis of hardware-software interactions including the circuit level is a novel approach. This analysis improves the design and development of EH-WSNs without a physical implementation.

Simulation Environment for Wireless Sensor Networks

A hardware-software simulation environment is designed and developed to analyze the interactions in EH-WSNs. The use of SystemC and the AMS extension enables a combined simulation of hardware and software components. The novelty of the simulation environment is the simulation of entire WSNs, including the simulation of the energy supply at circuit level, by using a single modeling language.

Case-Studies for Hardware-Software Interactions

Two novel case studies underline the advantage of analyzing hardware-software interactions. First, the component-aware dynamic voltage scaling shows that the optimal selection of voltage converters to supply a sensor node depends on the behavior of the software (and the resulted average power consumption). Second, a state-of-charge measurement error analysis reveals the impact of measurement errors on the behavior of the software, especially energy-aware tasking and routing. Furthermore, a simple probabilistic routing protocol is introduced to address this problem.

3 Hardware-Software Interactions in Energy Harvesting Wireless Sensor Networks

This chapter presents the methodology for analyzing hardware-software interactions, their classification, and the implemented simulation environment. The latter is designed to simulate EH-WSNs including software and hardware components as well as the environment of the sensor nodes. The results of the simulation can be used for evaluating the hardware-software interactions in EH-WSNs.

3.1 Analysis of Hardware-Software Interactions

Almost all electronic devices consist of hardware and software subsystems. This combination causes an interaction between them that should be analyzed to optimize the device. This section describes the hardware and software subsystem as well as different methodologies for their analysis.

3.1.1 Hardware and Software

Hardware is the collectivity of physical and interconnected components of an electronic device. Here, only hardware that can execute software will be considered. The main components involved in executing software are the processing unit, the memory, and the input/output. The processing unit loads and executes the software stored in the memory. The input/output is used for interacting with other hardware components. These three main components are used together as processing unit which is equipped with different interfaces to be able to communicate with other components, including sensors, external memory, or human-control interfaces. Furthermore, an energy supply is necessary to operate the hardware.

Software is stored in the memory and defines a certain behavior of the software-controlled hardware. The software consists of instructions and data, and the execution of these instructions processes the data and causes a change of the internal state of the hardware. Therefore, the execution of software also changes the power consumed by the hardware.

3.1.2 Interactions

If two subsystems are connected with each other, they will start to interact. Each system consisting of software and hardware is such an interacting system. This is shown in Figure 3.1. Embedded computers are systems that consist of a hardware subsystem including an embedded processor and a software subsystem. All of these systems have one thing in common: the execution of the software influences the hardware state. However, the hardware state does not always influence the software behavior. The precondition for this

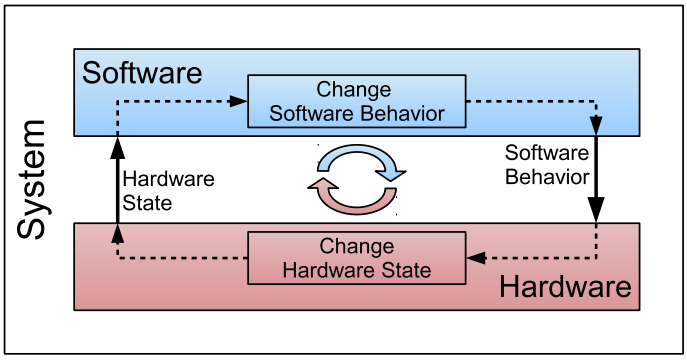


Figure 3.1: Principle structure of a system with hardware-software interactions.

influence is the existence of sensors. The software can access these sensors and change the behavior according to the measurement data. Energy-aware wireless sensor nodes are embedded systems with an entire interaction cycle. One example for an entire interaction cycle of a sensor node is the energy consumption variation: Energy-aware sensor nodes measure the state-of-charge of their energy storage components and adapt the behavior of the software (e.g. the duty-cycle). This changes the energy consumption of the sensor node; the new energy consumption alters the state-of-charge of the sensor node and the cycle starts again. This example shows the importance of the analysis of hardware-software interactions to predict the behavior as accurately as possible. The results of this analysis can be used for improving the system and reveal unwanted effects which would decrease the performance of the system.

3.1.3 Analysis Methodology

The analysis of hardware-software interactions requires a methodology to find out how the subsystems interact with each other. Basically, there are three different possibilities which are listed in the following.

Analytical Methodology: This approach uses equations to describe the system. Interactions can be deduced by solving these equations and calculating the results. Typically, it is hard to form the equations because their number increases with the detailedness of the described system.

Empirical Methodology: This approach uses observations of real systems to determine the interactions. The results of measuring the running (and interacting) system are analyzed to find out the interactions. These measurements are performed on the hardware (e.g. measurement of voltages, currents or timings) and software (tracking of the software's behavior).

Simulation Methodology: This approach uses a model of the system and a simulation environment. The results of the simulated models can be examined to deduce the interactions. It is similar to the empirical approach but with the advantage that no real system has to be implemented. The model of the system can be changed easily to analyze different kinds of interactions.

Although all three approaches can be used for examining hardware-software interactions, the simulation approach provides the highest flexibility. Therefore, this approach has been selected in this work.

3.2 Interactions in Energy Harvesting Wireless Sensor Networks

EH-WSNs are systems with strong interactions. On the one hand, they interact with their environment by means of measurements and energy harvesting (the harvestable energy depends on the available energy of the environment). On the other hand, their EHS requires an energy management to enable a long or perpetual operation which itself causes local interactions between the sensor node's units. This section describes the different interactions in EH-WSNs and shows the analyzed units and interactions.

3.2.1 The Sensor Node as Interacting System

As previously described, sensor nodes are embedded systems with hardware-software interactions between the four main hardware units described in Section 2.2.1. These interactions are summarized in the following.

Sensor Unit: This unit interacts with the processing unit by activating or deactivating the sensors and by the measurement itself. Furthermore, it may interact with the energy supply unit by measuring the energy-related parameters and by consuming energy.

Processing Unit: This unit interacts with all other units.

Communication Unit: This unit interacts with the processing unit by controlling communication and exchanging data. Furthermore, as the most power-consuming unit, it strongly interacts with the energy supply unit.

Energy Supply Unit: This unit interacts with all other units, for instance with the processing unit by using a controlled energy supply (e.g. voltage level, energy harvesting process).

These interactions are necessary in order to make a sensor node perform its tasks. An overview of the interactions is given in Table 3.1.

In this work, the focus lies on energy-related interactions inside and between sensor nodes. All other units influence the energy supply unit during their operation because of the energy consumption. As already mentioned, sensor nodes are often energy-aware systems that measure the available energy in order to adapt their behavior for extending the period of operation. Therefore, the sensor unit not only consumes energy but also measures the current state of the energy supply unit. This fact enables a full interaction cycle as previously described.

Energy harvesting causes an important interaction between the sensor node and its environment. Therefore, the available energy depends not only on the energy consumption of the sensor node but also on the harvestable energy of the environment.

Due to the importance of energy in EH-WSNs, the focus of this work lies on the analysis of the energy-related interactions. Therefore, the energy supply unit is the most important

Unit	Sensor (active)	Energy Supply (active)	Processing (active)	Communication (active)
Measurement (passive)		• Energy supply	• Data gathering • Data storage • Data processing	
Energy Supply (passive)	• Determination of energy state		• Dynamic voltage • Component-selective supply	
Processing (passive)	• Measurement notifications	• Energy supply • Low energy notification		• Communication notifications (RX, TX, collision, ...)
Communication (passive)		• Energy supply	• Exchange of messages • Communication control	

Table 3.1: Overview of interactions between the four main units of a sensor node.

one. In EH-WSNs, this unit may be very complex and consists of many different components. The task of the unit may range from energy harvesting to safe energy storage and a guaranteed continuous supply of the sensor node. These tasks have to be trimmed in order to enable an optimized operation of the energy supply unit. However, the other units are also important to guarantee a correct functionality of the sensor node. None of them can be totally neglected and each is needed for a comprehensive analysis of the WSN.

3.2.2 The Wireless Sensor Network as Interacting System

Apart from the local interactions inside the sensor nodes, there are also interactions at network level. The sensor nodes exchange information between each other which may include data about the current state of the sensor node. At network level, these data can be used for adapting the behavior of the WSN. In combination with the local interactions, these network interactions also cause interaction cycles. In energy-aware WSNs, the workload distribution and routing is adapted according to the state-of-charge of individual nodes. Therefore, the interaction cycles are used for controlling and optimizing the WSN behavior. The typical optimization goal is the maximization of the WSN's lifetime.

EH-WSNs imply further energy-related interactions at network level. As discussed in the introduction, the behavior of the EH-WSN can be adapted to the harvestable energy of individual sensor nodes. This energy-aware behavior can enable a perpetual operation of the EH-WSN.

3.2.3 Classification of WSN Interactions

This thesis divides interactions related to WSNs into three classes which are described in the following sections: device level interactions, network level interactions, and external interaction. Figure 3.2 summarizes this classification.

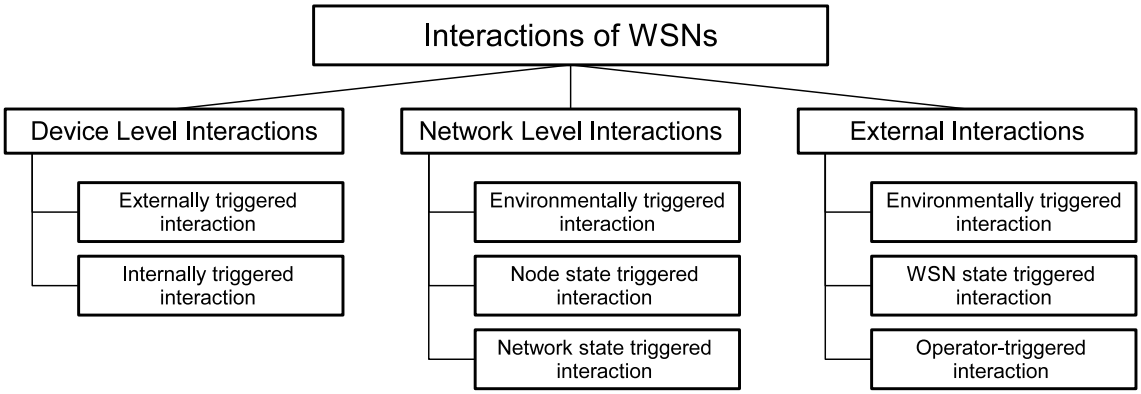


Figure 3.2: Classification of WSNs' interaction by means of influenced system level and cause.

3.2.3.1 Device Level Interactions

Device level interactions are caused by information exchange between the four main units of a sensor node and between their subcomponents. Typically, the behavior of the sensor node's local software is directly influenced by these interactions. Two different types can be distinguished:

Externally triggered interaction: Interactions between the sensor node's units are initiated by external events, including environmental changes, message receiving, and energy harvesting.

Internally triggered interaction: Interactions between the sensor node's units are triggered by internal events, for instance status changes or timer events.

All interactions between hardware and local software are part of this class. Therefore, energy-related interactions also belong to it. These interactions are important for EH-WSNs and hence investigated in this thesis.

3.2.3.2 Network Level Interactions

Network level interactions are caused by information exchange between individual sensor nodes. Typically, the behavior of the distributed software is directly influenced by these interactions. Three different types can be distinguished:

Environmentally triggered interaction: Interactions between sensor nodes are initiated by environmental changes or periodic monitoring tasks. Typically, the sensor nodes capture the environmental parameters and transmit them to a base station.

Node state triggered interaction: Interactions between individual sensor nodes are initiated by changes of their internal state. Examples are a monitoring of the battery's SoC which can be used for energy-aware routing or hardware failure indication which, in turn, can be utilized for task distribution.

Network state triggered interaction: Interactions between sensor nodes are initiated by changes of network parameters. Examples are the connectivity between individual nodes that is used by every routing algorithm, or the available bandwidth which can be utilized for network traffic distribution.

All interactions between hardware and distributed software are part of this class. The interactions concerning energy management are important for EH-WSNs and therefore investigated in this thesis.

3.2.3.3 External Interactions

External interactions are caused by an information exchange between the WSN and external networks or servers [118]. These interactions are often based on internal interactions in the WSN. Three different types can be distinguished:

Environmentally triggered interaction: Interactions with external networks are initiated by environmental changes or periodic monitoring tasks. Examples are external data logging of environmental parameters or event detection and visualization.

WSN state triggered interaction: Interactions with external networks are initiated by changes of the internal state of the network or individual sensor nodes. Examples are low battery indication or network connectivity monitoring and visualization.

Operator-triggered interaction: Interactions with external networks are initiated by the WSN operator, for instance the reconfiguration of one or more sensor nodes.

Due to the fact that the external interactions not only depend on the WSN, they are hard to analyze. Furthermore, they do not enhance the insight into energy-related hardware-software interactions in EH-WSNs. Therefore, they are not considered in this thesis.

3.3 Hardware-Software Simulation Environment and Its Architecture

This section introduces the implemented hardware-software simulation environment and shows its architecture. The most important requirements on a simulation environment are stated. Moreover, it describes the concept of the simulation environment, which is presented in [1], and the division into single modules. The principle of operation of these modules is explained in detail, and the design and implementation including an application scenario are described in [2].

3.3.1 Requirements

The following list shows the most important requirements on a simulation environment for an analysis of hardware-software interactions.

Combined Simulation of Software and Hardware: This is a precondition for the simulation of interactions between software and hardware.

Simulation Accuracy: The accuracy of the simulation results is important for predicting the behavior of WSN under certain conditions. The overall simulation accuracy implies the accuracy of the software simulation, of the hardware simulation, and of the environmental simulation.

Simulation Speed: The simulation time for a given amount of simulated time should be low in order to be able to analyze different configurations of the WSN. Especially by using EH-WSNs, long periods of time must be simulated (compared with typical simulation periods of WSNs).

Flexibility: The flexibility is important for analyzing different configuration of the WSN. Examples are the exchange of the software and the replacement of the energy storage components of a sensor node. Such modifications should be as easy as possible.

3.3.2 Concept

The choice for a self-implementation of the simulation environment was made because of different factors. First, the programming language can be selected without any restrictions. Second, there is no need to integrate functionality into an existing simulator for which it was not originally designed. As mentioned in Section 2, most existing WSN simulators only provide behavioral simulation of a WSN. It would be difficult to integrate hardware simulation support into the existing WSN simulators or to set up a co-simulation framework. Third, the simulation focus can be adjusted to receive the best results with least computational time. Finally, a self-implemented simulation environment can be trimmed to meet the most important requirements.

To be able to implement the simulation environment as easily as possible, the programming language should support software simulation as well as hardware simulation. The simulation library SystemC-AMS has been selected since it combines both. This library enables the implementation and simulation of systems which are composed of modules. It is written in the programming language C++ which entails an implementation of the simulation environment in C++. The advantage of this programming language is the execution speed of the compiled programs. Besides, a large number of libraries enhance the development process.

3.3.3 Design and Implementation

This section presents the design and implementation of the hardware-software simulation environment for EH-WSNs. Figure 3.3 shows the structure of the simulation environment with its submodules. The partitioning of the simulation environment and the submodules are described in the following.

3.3.3.1 Partitioning

The main modules are the environmental model and the sensor node model which, again, can be separated into the behavior model and the energy supply model. The first two models are implemented in standard SystemC. The execution of the sensor node's software is simulated in the sensor node's behavior model, and the sensor node's energy supply model is implemented in SystemC-AMS. This allows modeling linear hardware components

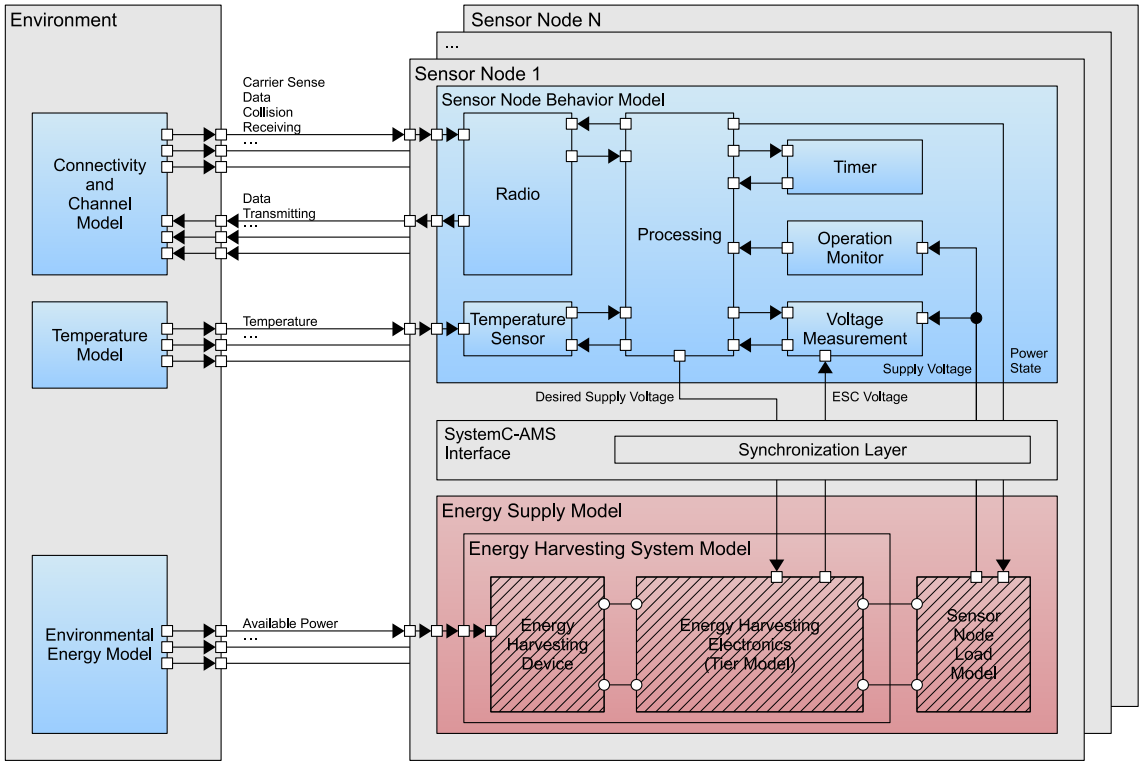


Figure 3.3: Structure of the implemented hardware-software simulation environment for EH-WSNs [2]. The non-shaded modules represent conventional SystemC modules and the shaded modules represent SystemC-AMS modules. The square connectors are SystemC ports and the round connectors show electrical terminals.

(e.g. resistors, capacitors). These components are connected to form an electrical network with a certain purpose. The SystemC-AMS library provides an interface between standard SystemC models and SystemC-AMS models. This interface automatically synchronizes the data exchange between the models by using the synchronization layer. A linear solver is used for simulating the electrical networks. Timed data flow models (also a class of SystemC-AMS models) are utilized for functional descriptions by using equations inside the models. Non-linear electrical components such as a diode are modeled in this way. The following sections describe the main modules of the simulation environment and their functionality.

3.3.3.2 Environmental Model

The environmental model is responsible for the interaction between the individual sensor nodes and for providing the nodes with location-specific environmental parameters (e.g. temperature, harvestable energy). Each transmission of a radio message is performed via the connectivity and channel model. At the beginning of a transmission, it checks if other sensor nodes are in transmission range and sends the radio message only to these nodes. The module is also responsible for detecting collisions during communication.

The magnitude of the environmental parameters changes over time and location. Therefore, the provided environmental parameters change during the simulation according to the time of day and other influences. Furthermore, this information may also vary between the individual sensor nodes depending on their location.

Network level interactions between the sensor nodes are influenced by this module and depend on the connectivity of the nodes and also on the environmental parameters.

3.3.3.3 Sensor Node Behavior Model

The sensor node behavior model includes the simulation of the sensor node's software. This software defines the behavior of the sensor nodes. The influence of the software on the power consumption of the sensor node is modeled by using power states. A change of the sensor node's power state causes a change of the power consumption inside the energy supply model.

The simulation of the software is event driven. The application software provides interfaces which are called by the processing module in case of an event and uses interfaces that are provided by the processing module to access and configure hardware modules. These interfaces are combined to a simulation abstraction layer, and if they are also provided by a real sensor node's operating system, the same application software can be executed. This would increase the comparability of the simulation results with a real implementation. It also enables an easy exchange of different application software.

The sensors of the nodes are also located inside this module and are used for interacting with the environment and also for measuring the internal state of the sensor node, especially the energy supply module. Therefore, they are important for interactions at device level and also at network level.

3.3.3.4 Energy Supply Model

The energy supply model is implemented in SystemC-AMS. It consists of different hardware components which are connected with each other to form the energy source model and the energy load model. The latter is implemented as a variable resistor that changes its value according to the power state provided by the sensor node behavior model. The energy source model is more complex. Depending on the application area of the EH-WSN, it may include different energy harvesting devices, different energy storage components, and different circuits to connect them. A detailed description of structuring a complex EHS can be found in the following section. The internal state of the energy supply module is important for EH-WSNs because the sensor nodes adapt their behavior according to the state. This adaption may include adaption at device level (e.g. activation or deactivation of specific sensors) and at network level (e.g. energy-aware routing). So, the energy supply model is important for the interactions not only at device level but also at network level.

3.3.4 Energy Harvesting System Model

As mentioned before, the EHS can be very complex in order to enable a continuous supply of the sensor node. Therefore, it is useful to divide the EHS into different blocks with special functionalities. One possibility to do so is to structure the EHS into five different tiers [3]. Each tier holds a special functionality and is connected to an inner and an outer

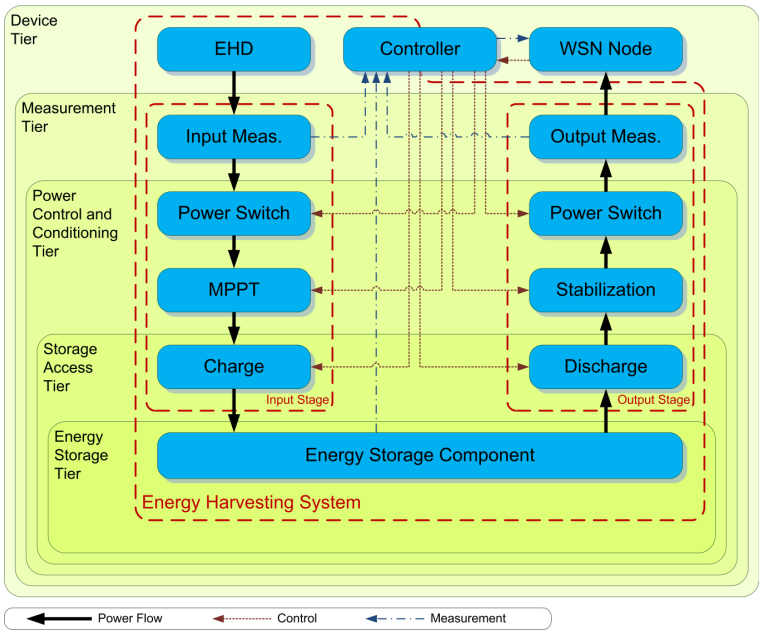


Figure 3.4: Standard EHS structured into five tiers including the power flow and the control and measurement possibilities [3].

tier (except the innermost and outermost tiers). These tiers can be exchanged easily to support a flexible simulation of different EHS configurations. Furthermore, each tier can be adapted to the other ones to enable an efficient flow of energy.

Not every tier (or part of a tier) is needed for every EHS. The simplest EHSs consist of an energy harvesting device which is directly connected to the sensor node. However, no control and no measurement are possible, and no energy can be stored. Typical EHSs are composed of at least an energy harvesting device, an energy storage component, and a circuitry to connect them. A standard EHS structured into the five tiers is shown in Figure 3.4. The measurement and control possibilities are important for interactions at device as well as network level. Each tier enables specific interactions which are described in the following sections.

3.3.4.1 Device Tier

The device tier contains the energy harvesting device, the sensor node, and the EHS controller. This controller may be included in the sensor node as a software module. Apart from controlling EHS, the device tier is responsible for energy harvesting and energy consumption. The tier enables the interaction between the sensor node and the environment concerning the harvestable energy.

3.3.4.2 Measurement Tier

The measurement tier is responsible for measuring the input and/or the output power. Typically, a voltage and a current measurement are performed to determine the power. These measurements which can be modeled in this tier are not lossless.

The measurement of the input and output energy is important for EH-WSNs. In enhanced applications, the information can be used for predicting the harvestable energy of the environment and the consumed energy of the sensor node to select the best strategy in order to avoid an interruption of the energy supply.

This tier enables interactions between the EHS and the sensor node's software based on its harvested and consumed energy. These interactions may influence the behavior at device level and network level.

3.3.4.3 Power Control and Conditioning Tier

The power control and conditioning tier is necessary to manipulate the power flow of the input and output stage. At the input stage for example, a maximum power point tracker increases the harvestable energy by an adaption of the power point of the energy harvesting device. It decouples the voltage level of the energy harvesting device and the energy storage component. At the output stage for instance, a voltage converter provides a stable supply voltage for the sensor node, independently of the energy storage component's voltage level. The components of this tier enable an active control of the power flowing through the EHS. Therefore, this tier influences energy-related interactions of the EHS.

3.3.4.4 Storage Access Tier

The storage access tier is responsible for a save charging and discharging of the energy storage component. The energy storage components are often very sensitive to overcharging, undercharging, or too high currents. Therefore, the necessity of this tier depends on the energy storage component. This tier influences the energy flow through the EHS if the energy storage component is at risk.

3.3.4.5 Energy Storage Tier

The energy storage tier consists of only one or more energy storage components. A part of the harvested energy is stored in this tier to enable a continuous operation of the sensor node. The necessary capacity of the energy storage component strongly depends on the variance of the harvestable energy and on the average power consumption of the sensor node.

The current energy level is an important value for EH-WSNs. It is used for adapting the behavior at device as well as network level. Therefore, the measurement of this value is very important for energy-related interactions.

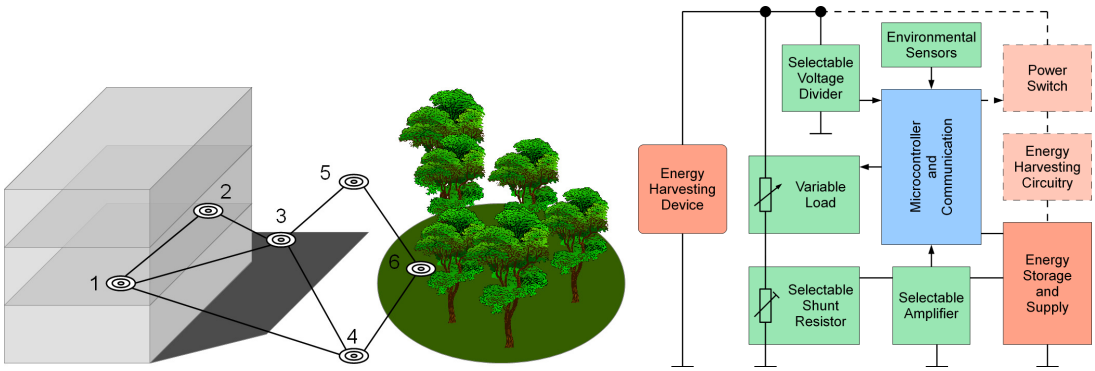
3.3.5 Enhanced Energy Harvesting Device Models

As already mentioned, the harvestable energy depends on the location and on the time of day. It varies between zero and a maximum power point which is determined by the energy harvesting device and the environmental conditions. Also the kind of the variation is dependent on different factors, including the weather. A realistic model of the energy harvesting device is important for good simulation results. The simplest way to obtain realistic data is to put the energy harvesting device at a specific location and characterize

it there. Therefore, an on-site characterization instrument [4], which is described in the following, was developed.

3.3.5.1 Characterization Concept

The advantage of the on-site characterization concept is the accuracy of the measured data. The energy harvesting device under characterization is placed at a specific location and connected to the characterization instrument. The instrument periodically performs a characterization and transmits the data. The characterization data consists of the measured environmental quantities (temperature, light) and a current-voltage characteristic diagram of the energy harvesting device. Typically, the difference of the harvestable energy at different locations with different environmental parameters is interesting. Therefore, several characterization instruments can be used for collecting temporally correlated and spatially distributed measurements. These instruments compose a WSN to extend the range and transmit the measured data to a central base station. An application scenario and the hardware structure of the characterization instrument are shown in Figure 3.5.



(a) (b)

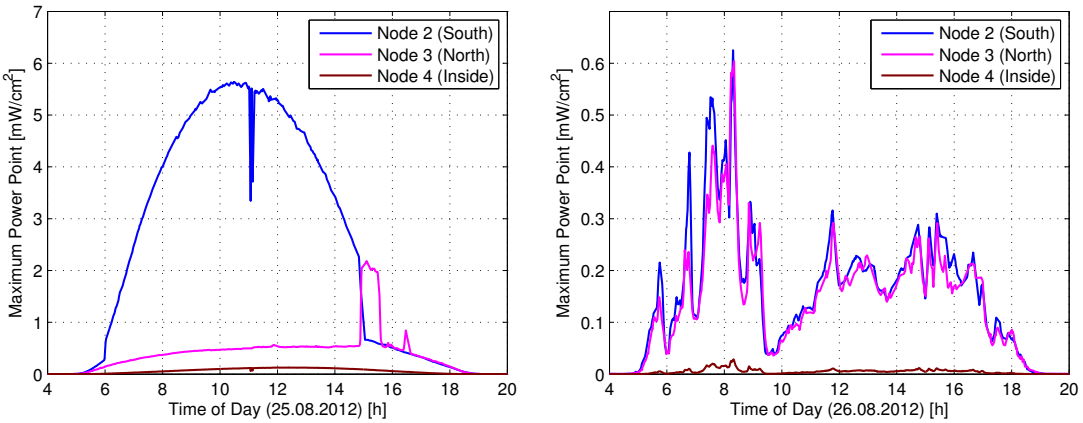
Figure 3.5: Application scenario (a) and hardware structure (b) of the characterization instrument for energy harvesting devices [4].

3.3.5.2 Solar Cell Characterization

Three of the characterization instruments are equipped with an amorphous silicon solar cell. Each of the sensor nodes is installed on the institute building (Graz, Austria, 47 °N latitude) at different locations. The first node is placed on the southern balcony which is sometimes shadowed when the sun is low (in winter). The second node is placed on the northern balcony that is shaded most of the day. The third node is placed inside the building and is also used as relay node. The first node was placed on the building in March 2012 and the measurements are running up to now with interruptions during service intervals. The sensor nodes installed outside are supplied by an EHS using the solar cell itself. Supercapacitors are used as energy storage components. Due to the limited capacity, only a period of one day with insufficient harvestable energy can be bridged. Therefore, the characterization interval is a trade-off between a guaranteed period of operation and a

temporal resolution. It is set to 100 s during winter when days with bad weather conditions are more likely.

Figure 3.6 shows the traces of the maximum power point of the three characterized solar cells on two different days. On the first day, the weather conditions were very good; it was sunny. Therefore, the southern solar cell harvests most energy. The difference to the other solar cells is significant. On the second day, the weather conditions were bad; it was cloudy. The clouds disperse the light of the sun and the solar cells installed outside harvest nearly the same amount of energy. Only the solar cell fixed inside harvests significantly less energy. The second day also shows a high variation of the maximum power point during the day. This variation has a maximum on days when the sun and clouds are changing.



(a) Good weather in Graz.

(b) Bad weather in Graz.

Figure 3.6: Maximum power point of three amorphous silicon solar cells on two days with different weather conditions. Note the scales of the diagrams. The solar cells are installed on the institute building in Graz, Austria (47 °N latitude). The tilt of the solar cells is 45 °.

Figure 3.7 shows the complete records of the characterized solar cells of more than one year. The data gaps are caused by updating and improving the characterization instruments' software and hardware. The plot reveals the significant difference of the harvestable energy during summer and winter. Furthermore, it shows the variation between consecutive days. Therefore, these data are used for enhancing the accuracy of the simulation environment.

3.4 Summary

This chapter introduced hardware-software interactions in EH-WSNs and showed the necessity of their analysis. Three different methodologies for analyzing EH-WSNs were explained and the simulation methodology was selected because of the high flexibility. To be able to analyze the interactions by using a simulator, the sensor nodes and their environment have to be mapped to models. Therefore, the main units of a sensor node and their interactions at device level were explained. Also interactions at network level

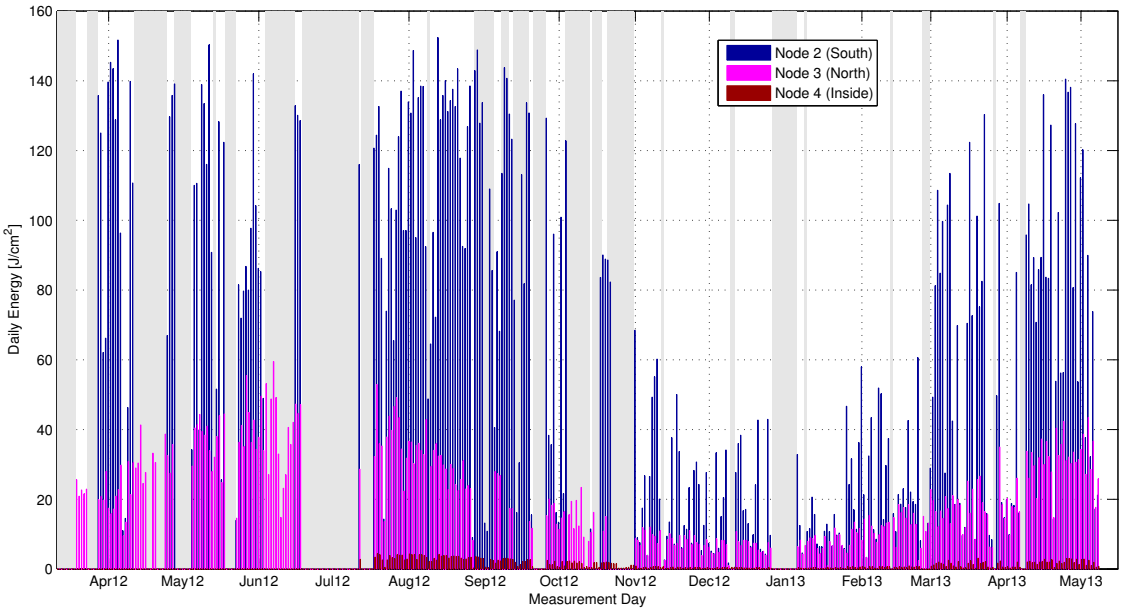


Figure 3.7: Daily harvestable energy of the three solar cells. The gray areas indicate days without a full record of all three solar cells.

and their impact on the behavior of the sensor network were listed. These and further interactions were summarized in a comprehensive classification. The architecture of the simulation environment enables an analysis of the interactions between the sensor nodes' hardware and software, between the sensor nodes and their environment, and between individual sensor nodes at network level. The focus of the simulation environment lies on the detailed modeling of the EHS. It was structured into five tiers consisting of hardware components which were simulated by using SystemC-AMS. Due to the flexibility of the tier structure, the composition of the EHS can be easily changed. Furthermore, an on-site characterization of solar cells was developed to improve the accuracy of the simulation results under real environmental constraints.

The flexibility of the simulation environment is ensured by its encapsulated modules. Therefore, it can be easily adapted to different application scenarios. This is shown on the basis of two examples in the following chapter.

4 Case Studies: Evaluation and Results

The theoretical part is needed to understand the issues of hardware-software interactions in EH-WSNs. However, a practical part is essential to support the technical claim of this thesis. Therefore, two case studies were developed to show the importance of analyzing hardware-software interactions. First, Section 4.1 demonstrates the influence of software on hardware by component-aware dynamic voltage scaling. Second, Section 4.2 shows the influence of hardware on software by state-of-charge measurement errors. Both case studies deal with the optimization of WSNs by analyzing the hardware-software interactions.

4.1 Case Study 1: Component-Aware Dynamic Voltage Scaling

Dynamic voltage scaling (DVS) is often used for decreasing the power consumption of CPUs by reducing the supply voltage. In combination with frequency scaling, the power consumption of a CPU can be adapted according to the workload during runtime. The influence of frequency and voltage on CMOS (complementary metal oxide semiconductor) circuits is the dynamic factor of their power consumption [121] and is shown in (4.1).

$$P_{Dynamic} = C \cdot f \cdot V_{Supply}^2 \tag{4.1}$$

As it can be seen, the supply voltage has a quadratic influence on the dynamic power consumption of the CMOS circuit. Therefore, a reduction can save much energy.

As introduced in [5], component-aware dynamic voltage scaling (CADVS) is an advancement of DVS. Typically, electronic devices consist of different hardware components which are needed to perform certain tasks. These components may operate at different supply voltages defined by the supply voltage range. During operation, the software may access different hardware components. Therefore, certain components can be switched off during certain intervals. The remaining activated hardware components define the lowest possible supply voltage. Due to the fact that the activated components change over time, also the lowest possible supply voltage may be altered.

4.1.1 Concept

CADVS adapts the supply voltage to the activated hardware components in order to save energy. Therefore, the supply voltage must be controllable during runtime. The chronology of the activated components is defined by the software executed on the microcontroller which has to set the correct supply voltage. In order to be able to do this, a variable voltage converter is needed. Furthermore, the power supply of the deactivated hardware components should be switchable for two reasons. First, in some cases it is possible to save more energy by a complete deactivation of the energy supply. Second, different components may not have overlapping supply voltage ranges. Therefore, the adaption of the supply voltage to the currently accessed hardware component may violate the supply voltage range of

another component. By a complete deactivation of the unused component's energy supply, the adaption of the supply voltage is possible without a violation.

The structure of a sensor node including the switchable energy supply and the variable voltage converter supporting CADVS is shown in Figure 4.1.

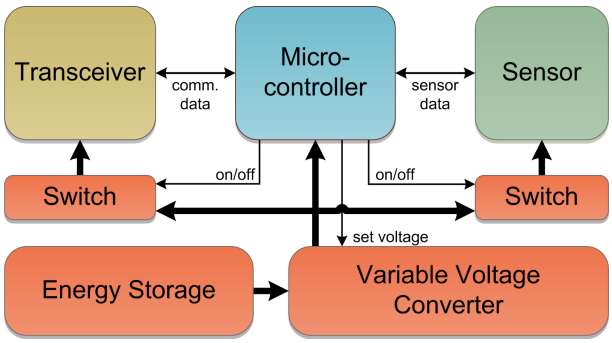


Figure 4.1: Structure of the sensor node (extended from [76]) [5, 6]. CADVS is enabled by the switchable energy supply of the sensor and of the transceiver, and by the variable voltage converter.

4.1.2 Prototype Implementation and Measurement

A prototype of a sensor node has been implemented according to the structure shown in Figure 4.1. The main hardware components and their supply voltage range are depicted in Table 4.1.

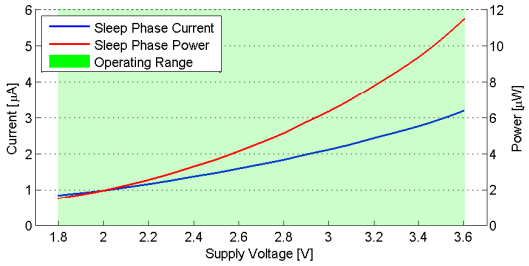
Hardware Component	Type	Manufacturer	Supply Voltage Range
Microcontroller	MSP430F1611	Texas Instruments	1.8 V to 3.6 V
Temperature Sensor	TMP05B	Analog Device	3.3 V to 3.3 V (best accuracy)
Transceiver	MRF24J40MB	Microchip	2.4 V to 3.6 V

Table 4.1: Main hardware components and their supply voltage range [5, 6].

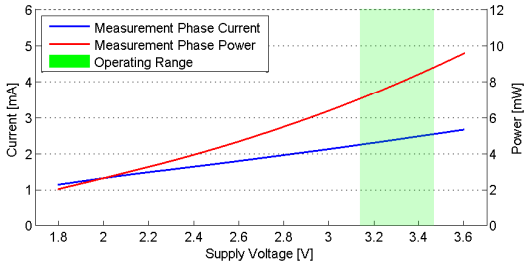
The task of the application scenario is to measure the temperature, preprocess the data and transmit them. The sensor node is in sleep state between the active phases. An overview of the different phases is given in Table 4.2. The sleep time is variable in order

Phase	Active Components	Minimal Supply Voltage	Duration
Sleep	Microcontroller	1.8 V	variable
Measurement	Microcontroller, Temperature Sensor	3.3 V	0.5 s
Computation	Microcontroller	1.8 V	0.5 s
Communication	Microcontroller, Transceiver	2.4 V	0.5 s

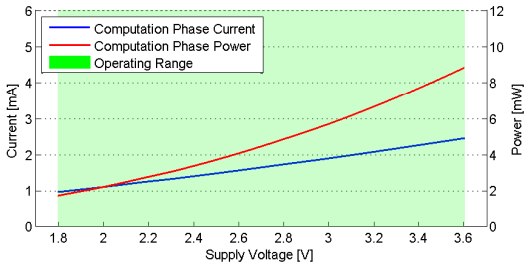
Table 4.2: Active hardware components, minimum supply voltage, and the duration of the four different phases [5, 6].



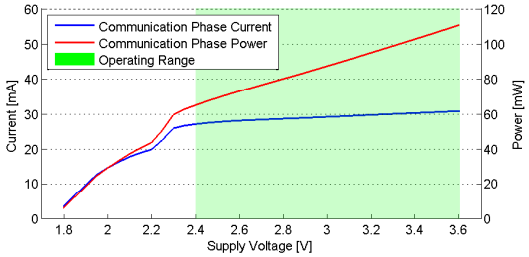
(a) Sleep phase.



(b) Measurement phase (best accuracy at 3.3 V).



(c) Computation phase.



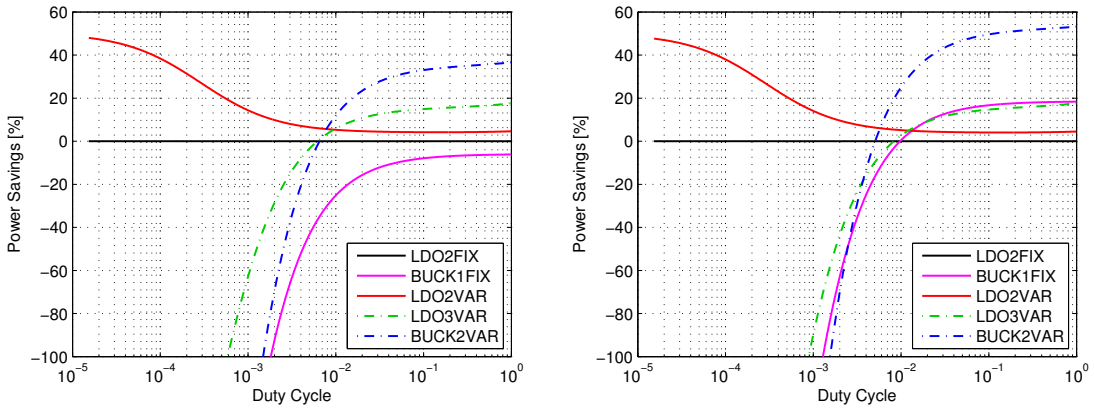
(d) Communication phase.

Figure 4.2: Current and power consumption of the sensor node during the four different phases depending on the supply voltage [6]. The shaded area indicates the supply voltage range.

to analyze the energy savings at different DCs which depend on the software of the sensor node. During these four phases, the power consumption of the sensor node has been measured at different supply voltages to illustrate the possible energy savings, as shown in Figure 4.2. The shaded area indicates the supply voltage range of the components. Using a constant supply voltage, 3.3 V is necessary to operate the temperature sensor with the utmost accuracy. In this case, CADVS could save a lot of energy (viewed proportionally) during the sleep and the computation phase. Due to the fact that the absolute value of the energy savings depends on the DC of the sensor node, the savings are analyzed as a function of the DC. Furthermore, the real energy savings are also dependent on the quiescent and leakage currents of the voltage converter. Five different supply circuits have been implemented to analyze the real energy savings. These circuits are explained in detail in [6]. Three linear dropout regulators and two switching regulators show the difference between these converter types. A detailed description showing the output voltage range is given in Table 4.3. Another influencing factor is the input voltage of the converters because

Converter	LDO2FIX	BUCK1FIX	LDO2VAR	LDO3VAR	BUCK2VAR
Conversion Mode	linear	switching	linear	linear	switching
Variable Voltage	no	no	limited	yes	yes
Voltage Range	3.3 V	3.3 V	2.2 V or 3.3 V	1.8 V – 3.3 V	1.8 V – 3.3 V

Table 4.3: Five different voltage converters and their characteristics which are implemented to analyze CADVS [6].



(a) Input Voltage 3.6 V. (b) Input Voltage 4.9 V.

Figure 4.3: Power savings of the five different voltage converter circuits as a function of the duty cycle at two different input voltages of the converters [6].

of their different conversion efficiencies. Therefore, the measurements at the conversion circuits were done at two different voltages and the results are illustrated in Figure 4.3. Figure 4.3(a) shows the energy savings as a function of the DC at an input voltage of 3.6 V. The five different traces show the five different voltage conversion circuits, the linear dropout regulator (LDO2FIX) being used as reference circuit. The main result is that the optimal voltage converter depends on the DC. At low DCs (which means long sleep phases), the linear dropout regulator (LDO2VAR) is more efficient although this circuit supports only two different voltage levels. The reason is the proportionally high influence of the quiescent and leakage currents on the other voltage converters. At high DCs (which means short sleep phases), the switching regulator with a variable voltage (BUCK2VAR) provides the highest efficiency due to the high conversion efficiency. Figure 4.3(b) shows the energy savings as a function of the DC at an input voltage of 4.9 V. Two main differences can be observed compared to Figure 4.3(a). The first one is that the energy savings of the switching regulators increase because of the nearly equal conversion efficiency at different input voltages. The second observation is implied by the first one. The changeover of the most energy-efficient converters moves towards lower DCs at higher input voltages. These results show the importance of analyzing the effects of software on hardware to develop energy-efficient WSNs.

4.1.3 Simulation and Results

Simulating CADVS is only possible when using a combined simulation of hardware and software due to their strong interactions. The implemented simulation environment introduced in Section 3.3 supports this as described in [2]. A model of a variable voltage converter is implemented to enable CADVS. This converter is controlled by the executed software that depends on the accessed hardware components.

The simulation scenario is a WSN consisting of a base station and five sensor nodes; this is shown in Figure 4.4. The sensor nodes are supplied by an EHS consisting of a solar cell and two double layer capacitors (DLCs) connected in series. Node 2 harvests

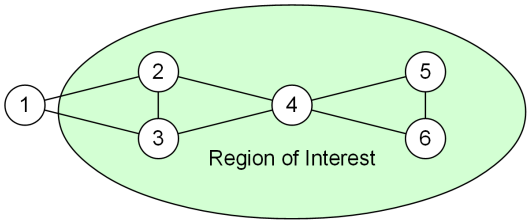
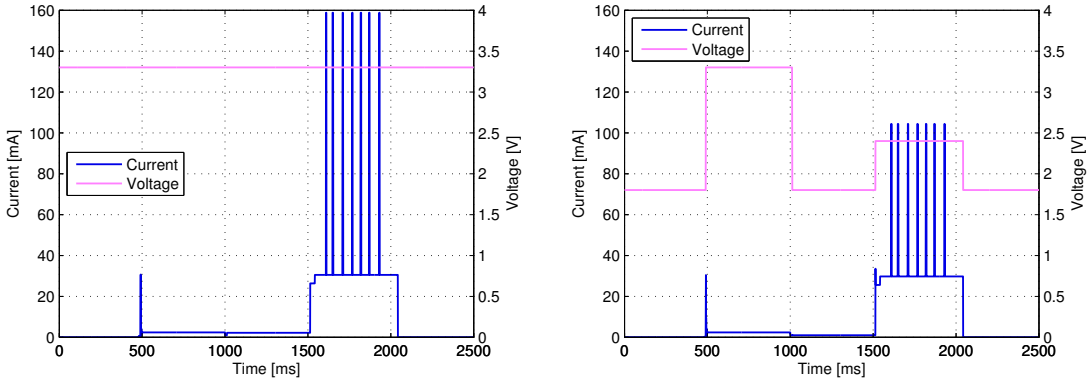


Figure 4.4: Structure of the WSN consisting of five sensor nodes supporting CADVS [2]. Node 1 is the base station, and nodes 2 to 6 are the sensor nodes.

most and node 6 least energy. The energy-aware application of the base station calculates the DC of the network which is dependent on the lowest measured SoC of the sensor node's energy storage components. This DC is transmitted to all sensor nodes to enable a synchronized communication. The routing protocol is also energy aware by selecting the next hop according to the receiver's SoC. The sensor nodes are intended to periodically transmit the data of the temperature measurements to the base station. Two different voltage conversion circuits are analyzed: first, a linear dropout regulator with a constant output voltage of 3.3 V; second, a linear dropout regulator with a variable output voltage supporting CADVS.

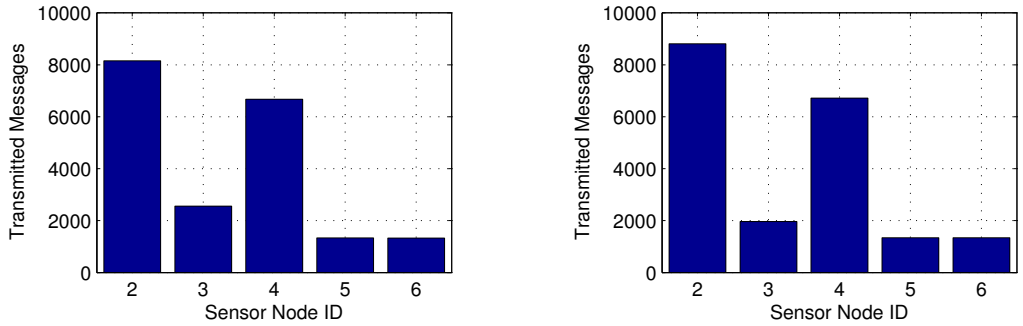
The simulation results of the supply voltage and current are shown in Figure 4.5. It can be stated that the peak current consumption during the transmission of the messages is significantly reduced. The detailed results can be found in [2]. The energy savings during the active phases (measurement, computation, and communication) amount to 34%. Here, only the energy consumed by the sensor node is considered. By analyzing the input energy of the voltage converter, the energy savings are reduced by 11.4%. The reason is the type of voltage converter. In both cases, linear dropout regulators are used. Therefore, the energy savings result from the reduced current consumption of the sensor node.



(a) Constant supply voltage of 3.3 V. (b) Variable supply voltage supporting CADVS.

Figure 4.5: Simulation of the supply voltage and current with a constant (a) and variable (b) supply voltage [2].

Finally, the message statistics of the five sensor nodes can be found in Figure 4.6. Node 4 has to forward all messages of node 5 and node 6. The forwarding of the messages from



(a) Constant supply voltage of 3.3 V. (b) Variable supply voltage supporting CADVS.

Figure 4.6: Transmitted message statistics of the five simulated sensor nodes [2].

the base station to node 4 and vice versa is performed by node 2 and node 3. Due to the fact that node 2 harvests more energy than node 3, it forwards more messages, as shown in Figure 4.6(a). The increased number of transmitted messages of node 2 (Figure 4.6(b)) is caused by the better energy efficiency by using CADVS.

This case study shows the importance of considering interactions between hardware and software in order to optimize the sensor node's energy efficiency. Especially the effects of software on hardware have to be considered here, for instance the influence of the DC. Furthermore, the simulation shows the effects of hardware on energy-aware software (the behavior of the WSN at network level) by the use of different hardware components. Therefore, this case study demonstrates both directions of interaction.

4.2 Case Study 2: State-of-Charge Measurement Error Analysis

Measurement values may influence the behavior of WSNs, especially energy-related ones affecting EH-WSNs. Energy-aware applications and services use these values to determine their behavior. Measurement errors may influence the correctness of decisions; wrong decisions lead to a degradation of the WSN's operability. Therefore, a measurement error analysis can unveil an unexpected and erroneous behavior of WSNs. Such an analysis was conducted by means of SoC measurement errors. The simulation concept and results are shown in the following.

4.2.1 Concept

One of the most important parameters for energy-aware applications and services is the SoC. This parameter indicates the remaining energy of the energy storage component. As described in [7], an erroneous determination of this residual energy may influence the behavior of the applications and services. There are different methods for measuring the SoC. One of the easiest is to quantify the terminal voltage of the energy storage component. This method is suitable for demonstrating the effects of measurement errors. Most microcontrollers have integrated ADCs to measure voltages. However, every ADC has a certain measuring inaccuracy. Here, a MSP430F1611 microcontroller is used with an inaccuracy of ± 5 LSB (least significant bit). Furthermore, the inaccuracy of the supply

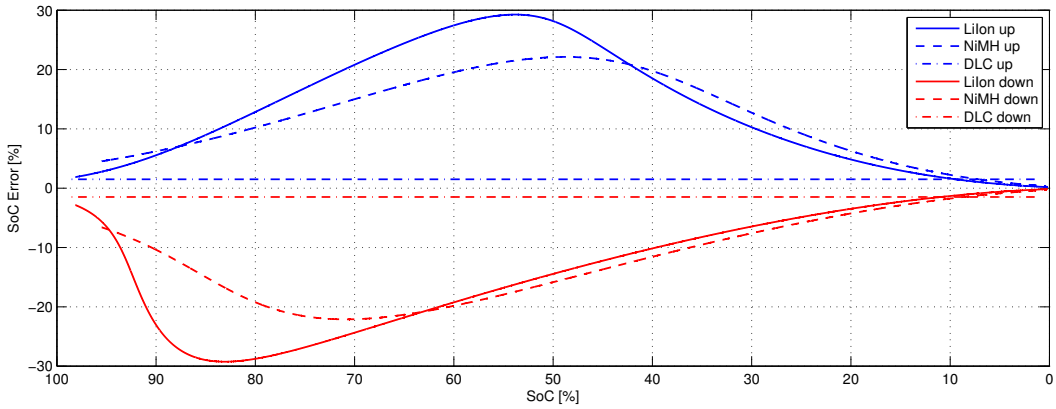


Figure 4.7: SoC measurement error limits as a function of the three different energy storage component's SoC [7]. The three curves in the upper half represent the error upwards, and the three curves in the lower half represent the error downwards.

voltage is assumed to be ±33 mV as provided by a TPS63031 voltage regulator. The overall measuring inaccuracy is ±37.03 mV which is used for a first analysis.

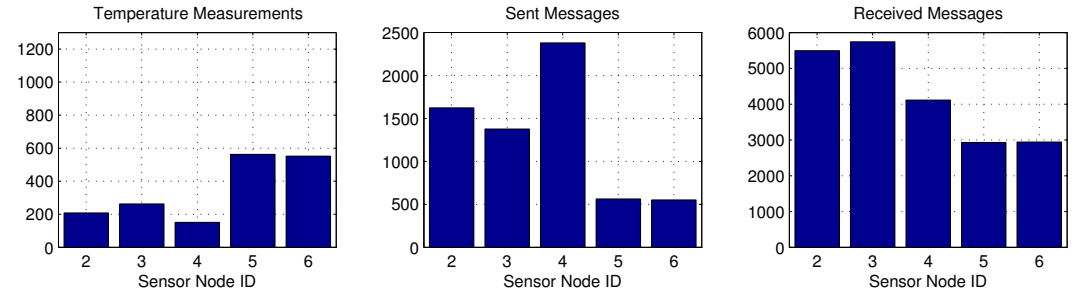
Three different energy storage components are analyzed: a LiIon rechargeable battery, a NiMH rechargeable battery, and a DLC. Figure 4.7 shows the SoC measurement error limits as a function of the energy storage component's SoC. The discharge traces needed for this analysis are simulated by using Matlab/Simulink and a constant discharge current. The error upwards is the significant value since an overrated SoC may result in an overload (network traffic or tasks) of this sensor node. Therefore, the sensor node's energy storage component may discharge faster than that of the others. The DLC shows a constant error because of the constant current discharge. The error is small due to the faster drop of the terminal voltage. Both rechargeable batteries show a critical zone in the middle of discharge. The reason is the slow drop of the terminal voltage in this zone. The determination of the SoC shows the highest inaccuracy here. Such an overload of a sensor node was simulated as described in the following section.

4.2.2 Simulation and Results

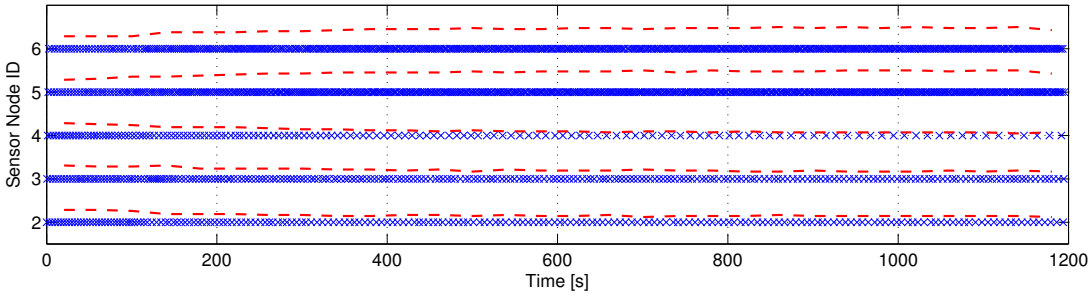
The SoC measurement error analysis is simulated by using the introduced simulation environment, as published in [8]. The network consists of a base station (node 1) and five sensor nodes (nodes 2 to 6). The structure is the same as in the previous case study shown in Figure 4.4. However, the application scenario is slightly different. Only one sensor node is selected to measure the temperature inside the region of interest. This process is reiterated until one sensor node runs out of energy. An energy-aware algorithm is used for selecting the routes of the messages as well as the measuring sensor node. Two different algorithms are compared; first, the normal algorithm always selects the sensor node with the highest SoC. Second, the enhanced algorithm randomly selects one from the two sensor nodes with the highest SoC. Furthermore, two different measurement error settings are simulated. First, all sensor nodes properly work without measurement errors. Second, node 4 is considered to have a measurement error after some time.

Normal Energy-Aware Algorithm without Measurement Errors

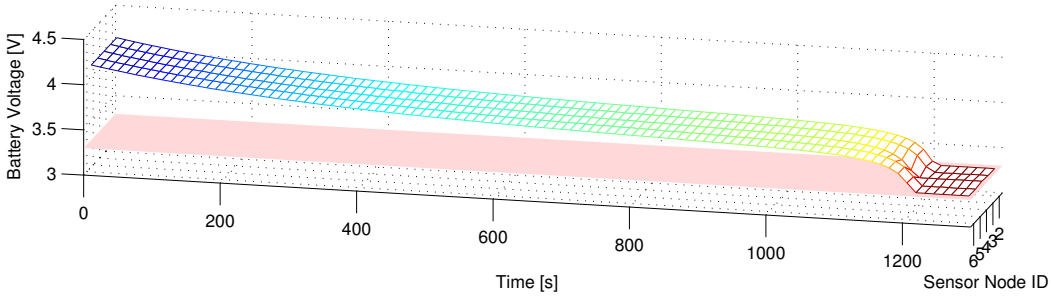
First, the normal energy-aware algorithm without measurement errors is simulated and is used as reference for further simulations. Figure 4.8 presents the simulation results. The statistics of the temperature measurement and network traffic are shown in Figure 4.8(a). Most temperature measurements are performed by node 5 and 6. The reason is that these sensor nodes do not forward any messages. Node 4, however, has to pass on most messages. Therefore, it performs the fewest temperature measurements. Figure 4.8(b) depicts the chronological sequence of the temperature measurements. It can be seen that all sensor nodes start with the same measurement density. The density of node 5 and 6 increases, the density of node 4 decreases, and the density of node 2 and 3 decreases only slightly. Figure 4.8(c) shows the discharge traces of the sensor nodes' energy storage components.



(a) Network traffic and temperature measurement statistics.



(b) Temperature measurement distribution.

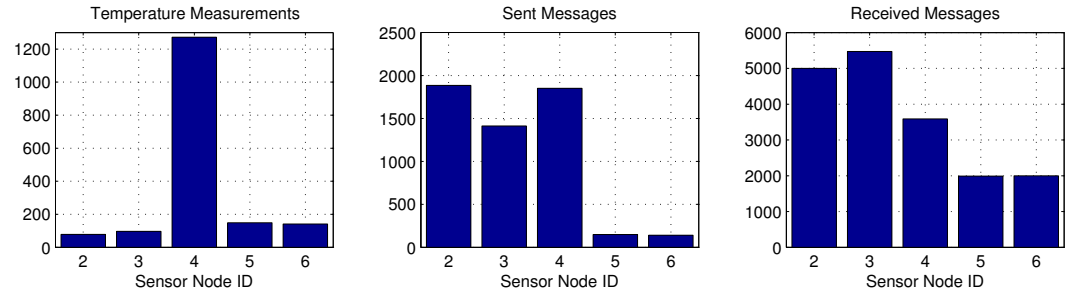


(c) Voltage trace of the sensor node's energy storage components.

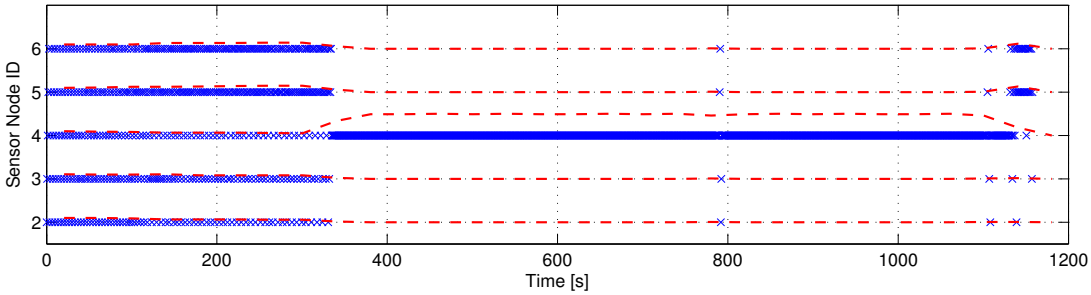
Figure 4.8: Simulation results of the normal energy-aware algorithm for routing and workload distribution [8]. No SoC measurement errors occur at any sensor node during the entire simulation.

Normal Energy-Aware Algorithm with Measurement Errors

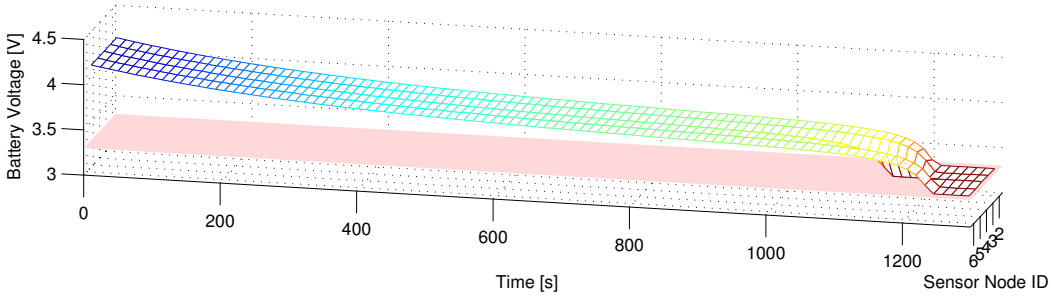
Second, the normal energy-aware algorithm and a WSN with a faulty sensor node are simulated. Figure 4.9 presents the results. The determination of the SoC of node 4 gets erroneous after 350 s. For this simulation, the measurement error of the energy storage component's terminal voltage is assumed to be +4%, which results in a maximum SoC determination error of 45%. The energy-aware algorithm loads node 4 with almost all temperature measurements, which is illustrated in Figure 4.9(a) and 4.9(b). The other sensor nodes hardly ever produce measurements after the error occurrence. The result is that the real SoC of node 4 decreases faster than that of the other sensor nodes. The discharge traces of the sensor nodes' energy storage components illustrate this effect as shown in Figure 4.9(c). Compared to the first simulation, the increased energy consumption of



(a) Network traffic and temperature measurement statistics.



(b) Temperature measurement distribution.



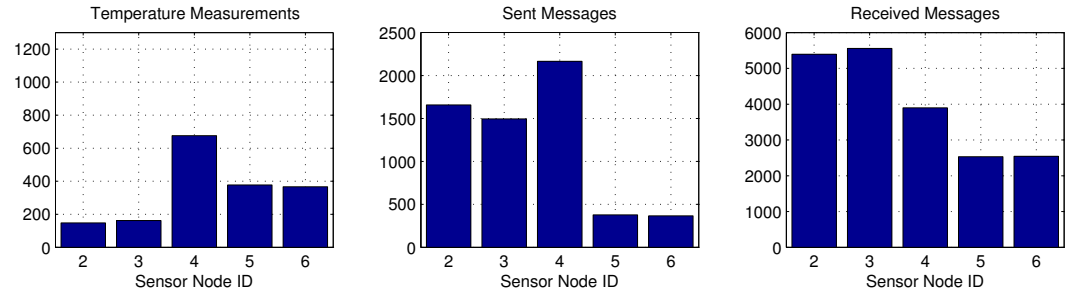
(c) Voltage trace of the sensor node's energy storage components.

Figure 4.9: Simulation results of the normal energy-aware algorithm for routing and workload distribution including SoC measurement errors at node 4 after a certain period of time [8].

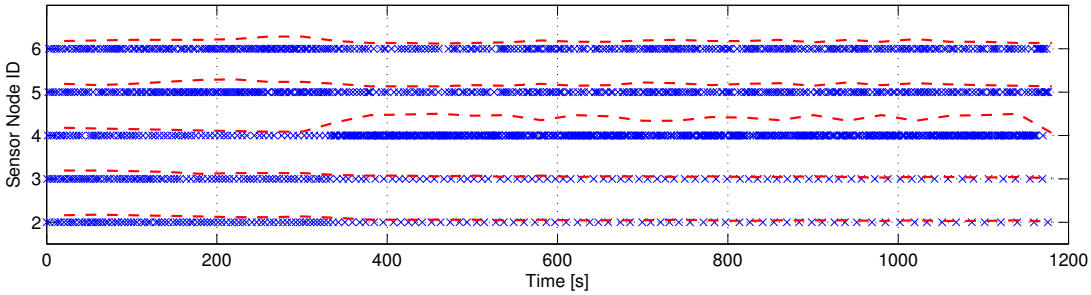
node 4 causes a reduction of the network lifetime by 3.13%. Although this reduction does not seem to be serious, in other application scenarios it may be more significant. For instance, it would be a severe problem if parts of the network would be disconnected from the base station due to an early failure of a sensor node.

Enhanced Energy-Aware Algorithm with Measurement Errors

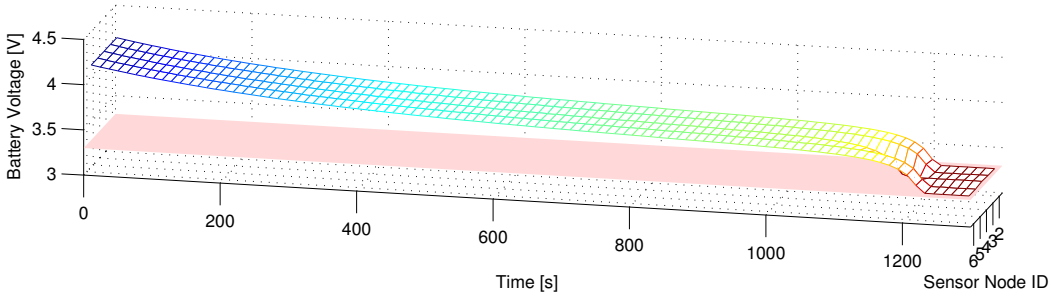
Third, the enhanced energy-aware algorithm and a WSN with a faulty sensor node are simulated. Figure 4.10 shows the simulation results. The temperature measurement and the network traffic statistics as shown in Figure 4.10(a) depict that also here, most measurements are performed by node 4. However, the other sensor nodes also produce measurements after the error occurrence, as illustrated by the chronological sequence of the



(a) Network traffic and temperature measurement statistics.



(b) Temperature measurement distribution.



(c) Voltage trace of the sensor node's energy storage components.

Figure 4.10: Simulation results of the enhanced energy-aware algorithm for routing and workload distribution including SoC measurement errors at node 4 after a certain period of time [8].

measurements (Figure 4.10(b)). This is the main difference to the previous simulation. The reason is the random selection of the two sensor nodes with the highest SoC. The measurement densities of nodes 2, 3, 5, and 6 are reduced compared to the first simulation but they are not nearly completely skipped as in the second simulation. This is the main advantage of the enhanced energy-aware algorithm. The network lifetime is increased by 1.66% compared to the second simulation.

Table 4.4 summarizes the simulation results and compares the network lifetime of the different simulation settings. It can be seen that the network lifetime is reduced by the faulty sensor node. This is a reduction of 3.13% compared to the error-free simulation. The enhanced energy-aware algorithm increases the network lifetime by 1.66%. These results show the advantage of simulating hardware-software interactions in WSN. The effects of hardware on the behavior of the software are highlighted. Therefore, this simulation stresses the importance of a combined simulation of hardware and software to optimize WSNs.

Simulation Settings	First Failing Sensor Node	Network Lifetime
All sensor nodes OK, normal routing	4	1195.10 s
Node 4 faulty, normal routing	4	1157.75 s
All sensor nodes OK, enhanced routing	4	1195.31 s
Node 4 faulty, enhanced routing	4	1176.96 s

Table 4.4: Summary of simulation results of the four different measurement setups [8]. The network lifetime is given by the first failing sensor node which runs out of energy.

5 Conclusion and Future Work

The thesis concludes with this chapter that gives direction for future work. The first section summarizes the work and shows the importance of the contribution. The second section identifies and explains possible improvements.

5.1 Summary and Conclusion

WSNs are used in application areas without wired infrastructure. Therefore, each sensor node of a WSN has its own energy supply, resulting in power and energy constraints. EHSs increase the period of operation or enable perpetual operation.

It has been shown that EH-WSNs are interacting systems which have to be analyzed in their entirety. Typically, a sensor node of an EH-WSN consists of hardware and energy-aware software, both optimized for low power consumption. These two main parts strongly interact with each other. Furthermore, the sensor nodes are interacting with each other by exchanging information about their environment and their state. The interactions have been classified into three main groups: device level interactions, network level interactions, and external interactions. The first two were analyzed in detail and are considered as important for this thesis. The main interactions at device level between the four main units of a sensor node were listed. The interactions at network level were also discussed but the focus was put on energy-related interactions at both levels.

A simulation environment which enables a combined simulation of hardware and software was implemented. Entire EH-WSNs including their environment can be simulated and the interactions can be analyzed. A detailed EHS model was implemented to get realistic results. Furthermore, the on-site characterization of energy harvesting devices enables an accurate simulation of EH-WSNs at different locations. It was shown that the relative difference of harvestable energy of the solar cells at different locations strongly depends on the weather.

Two case studies have been conducted to illustrate the importance of hardware-software interactions in EH-WSNs. First, CADVS illustrated the dependency of the optimal hardware selection on the behavior of the software. Second, the state-of-charge measurement error analysis showed the effects of hardware state changes on the behavior of the WSN.

The three contributions of this thesis were examined in Section 3 and Section 4, the analysis of hardware-software interactions being the major one. A detailed analysis of the interactions, which was conducted for energy-related interactions at the device and network level, is supported by their classification. The second contribution is a self-developed simulation environment for EH-WSNs including hardware-software interactions, as summarized above. Finally, the third contribution consists of two case studies, both using the analysis of hardware-software interactions to optimize WSNs.

To conclude this thesis, the importance of the contributions is stated. The analysis of the hardware-software interactions enhances the optimization of WSNs regarding the

energy efficiency during all development steps and operation. Due to the increased energy efficiency, the period of operation can be extended. Especially EH-WSNs imply strong interactions with their environment. The optimization of these WSNs can achieve a perpetual operation which minimizes the maintenance. The implemented simulation environment enables a concurrent simulation of WSNs' software and energy harvesting hardware at circuit level by using a single modeling language. This new approach of simulating EH-WSNs enables a realistic analysis of hardware-software interactions. Finally, two case studies demonstrate that the analysis of the interactions actually enhances the optimization of WSNs.

5.2 Future Work

This thesis discussed the analysis of hardware-software interactions in EH-WSNs by using a self-developed simulation environment and two case studies. Based on the three contributions, the future work can be split into three categories which are discussed in the following.

Analysis of Interactions

The focus of this thesis was put on the analysis of energy-aware hardware-software interactions which can be extended to more general interactions between the units and components of sensor nodes. Furthermore, external interaction can be included to provide a system-wide analysis of interactions.

Simulation environment for EH-WSNs

The simulation environment can be improved in a variety of ways. First, the simulation of the node-to-node communication can be enhanced by including realistic channel models and more accurate models of the radio unit of the sensor node. Second, mobile sensor nodes can be supported by including different movement patterns. Third, the simulation speed can be improved by optimizing the models. Another possibility of simulation speed enhancement is to reduce the temporal resolution of the analog circuit simulation by averaging the power consumption of different power states. This possibility can be implemented and the accuracy can be analyzed. Fourth, the usability of the simulation environment can be improved, for instance by developing a graphical user interface.

Case Studies for Hardware-Software Interactions

First, the case study of CADVS can be extended by including different application scenarios. More complex application scenarios would also mean to add further sensor and hardware component models. Second, the case study of the state-of-charge measurement error analysis can be improved by simulating more general network topologies with an increased number of sensor nodes. This would improve the validity of the simulation results. Furthermore, the enhanced routing algorithm can be analyzed for different numbers of randomly selected nodes with the highest SoC.

6 Contributed Publications

Eight publications at international conferences and workshops contribute to this thesis. These publications can be divided into three categories. First, the category "Hardware-Software Simulation Environment Architecture" covers the publications concerning the developed simulation environment. Second, the category "Case Study 1: Component-Aware Dynamic Voltage Scaling" contains publications related to the first case study. One paper of this category is also included in the first category. And third, the publications of the category "Case Study 2: State-of-Charge Measurement Error Analysis" deals with the second case study. There is no category for interactions in WSNs since they are grouped with regard to their main focus, but all of the publications implicitly deal with interactions. Figure 6.1 gives an overview of the publications and the following paragraphs summarize them rather briefly. Thereafter, all publications are reprinted in their published version.

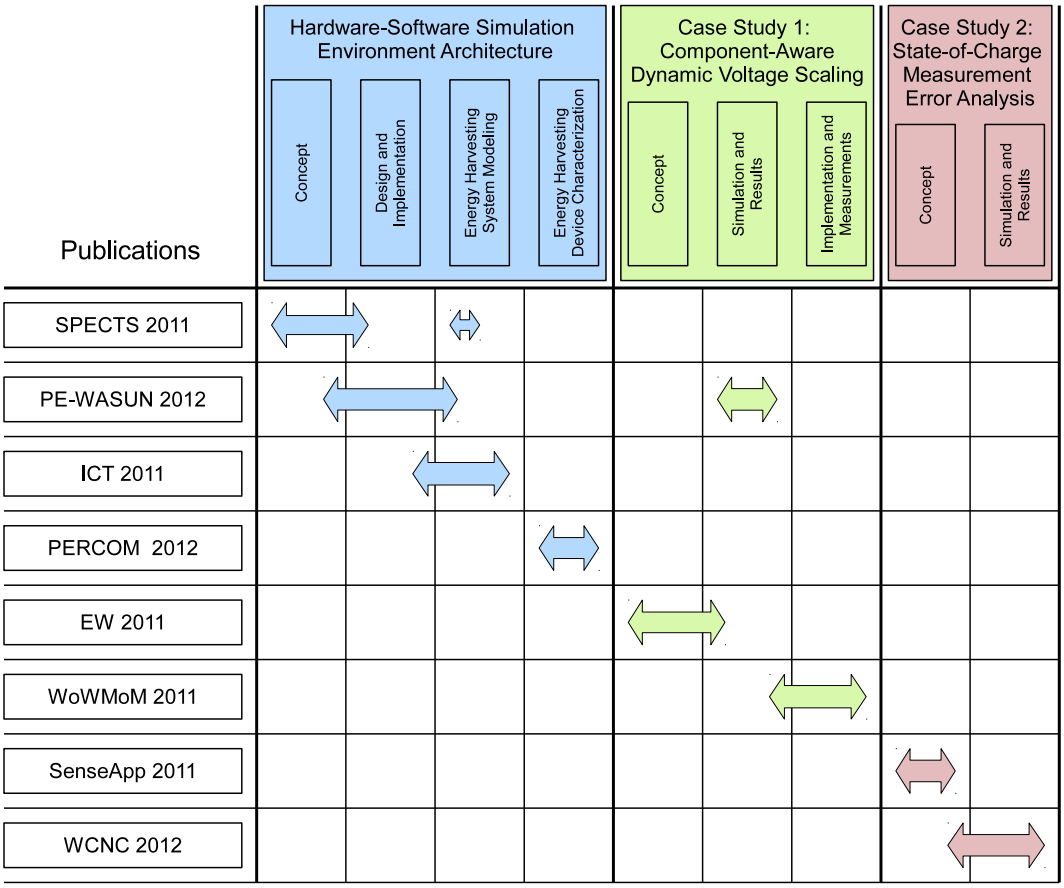


Figure 6.1: Overview of the eight publications contributing to this thesis.

The category "Hardware-Software Simulation Environment Architecture" contains four publications which are listed in the following.

- The publication **A SystemC-AMS Simulation Environment for the Evaluation of Energy Harvesting Wireless Sensor Networks** [1] was presented at the *International Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS 2011)* (reprinted from page 53). This publication is about the concept of the simulation environment and contains first simulation results of an energy harvesting system.

- The publication **A Hardware/Software Simulation Environment for Energy Harvesting Wireless Sensor Networks** [2] was presented at the *9th ACM symposium on Performance evaluation of wireless ad hoc, sensor, and ubiquitous networks (PE-WASUN 2012)* (reprinted from page 59). It illustrates the implementation of the simulation environment and contains simulation results of an EH-WSN. Furthermore, it presents a simulation scenario containing CADVS enhanced sensor nodes.

- The publication **Designing of Efficient Energy Harvesting Systems for Autonomous WSNs Using a Tier Model** [3] was presented at the *18th International Conference on Telecommunications (ICT 2011)* (reprinted from page 67). This publication introduces a tier model of the EHS's hardware. By using this model during the development, the energy efficiency of EHSs can be optimized. Furthermore, it groups hardware units into layers which can be used for a simulation model of EHSs. This model is featured by its modular structure and an easy exchangeability of the layers.

- The publication **Towards an On-Site Characterization of Energy Harvesting Devices for Wireless Sensor Networks** [4] was presented at the *IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops 2012)* (reprinted from page 73). It introduces the concept of the on-site characterization of energy harvesting devices. By integrating the characterization data in the simulation environment, it can be used for enhancing the accuracy of the simulation results.

The three publications of the category "Case Study 1: Component-Aware Dynamic Voltage Scaling" are related to the first case study and are listed in the following.

- The publication **Energy Efficient Supply of WSN Nodes using Component-Aware Dynamic Voltage Scaling** [5] was presented at the *European Wireless Conference (EW 2011)* (reprinted from page 77). It describes the concept of CADVS and gives a first estimation of the energy savings by using CADVS at different operating conditions.

- The publication **Evaluation of Component-Aware Dynamic Voltage Scaling for Mobile Devices and Wireless Sensor Networks** [6] was presented at the *IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM 2011)* (reprinted from page 85). It evaluates CADVS by using five implemented supply circuits.

- The publication **A Hardware/Software Simulation Environment for Energy Harvesting Wireless Sensor Networks** presents the results of the implemented simulation environment simulating a WSN enhanced by CADVS. It is also part of the first category (reprinted in the first category from page 59).

The category "Case Study 2: State-of-Charge Measurement Error Analysis" contains two publications concerning the second case study which are listed in the following.

- The publication **Measuring the State-of-Charge - Analysis and Impact on Wireless Sensor Networks** [7] was presented at the *Sixth IEEE International Workshop on Practical Issues in Building Sensor Network Applications (SenseApp 2011)* (reprinted from page 95). It discusses the impact of measurement errors of a sensor node's SoC on WSNs.
- The publication **State-of-Charge Measurement Error Simulation for Power-Aware Wireless Sensor Networks** [8] was presented at the *IEEE Wireless Communications and Networking Conference (WCNC 2012)* (reprinted from page 99). It discusses the results of the implemented simulation environment simulating the SoC measurement error.

A SystemC-AMS Simulation Environment for the Evaluation of Energy Harvesting Wireless Sensor Networks

Leander B. Hörmann, Philipp M. Glatz, Christian Steger and Reinhold Weiss
Institute for Technical Informatics
Graz University of Technology
Inffeldgasse 16, 8010 Graz, Austria
Email: {Leander.Hoermann@, Philipp.Glatz@, Steger@, RWeiss@}TUGraz.at

***Abstract*—Wireless sensor networks (WSNs) are very complex systems. They are often used in application areas with poor infrastructure and mobility, so each sensor node of a WSN needs its own power supply. Energy harvesting systems (EHSs) can be used to extend the operational lifetime of sensor nodes or even enable perpetual operation. However, the efficiency of the power supply depends not only on the used hardware itself but also on the application running on the sensor node. Therefore, a simulation environment including the complete WSN is needed to be able to understand all dependencies.**

This work presents the concept and first results of a simulation environment written in SystemC-AMS for the simulation of complete WSN. It includes energy harvesting, communication, power-awareness, location-awareness and dynamic power management. Each node is composed of single modules which can be exchanged easily to test different hardware or software components. The results show the ability to simulate energy harvesting systems using SystemC-AMS.

***Keywords*-wireless sensor networks; energy harvesting; modeling; SystemC-AMS; energy efficiency**

I. Introduction

Wireless sensor networks (WSNs) consist of a high number of nodes. These sensor nodes measure physical quantities and transmit the information of their environment through the wireless network. Typically, each sensor node needs a dedicated power supply, because of mobility or poor or no wired infrastructure at the given application area. A few examples are: precision agriculture [1], wildlife monitoring [2], human health care [3], and structural health monitoring [4]. Often, the sensor nodes are powered by batteries or so called energy harvesting systems (EHSs). The EHSs convert the energy supplied by the environment into electrical energy using energy harvesting devices, e.g. solar cells. Due to the fact that the size of a sensor node is often limited, the size of the battery or EHS is also limited. Therefore, the available power is limited. The power supply and the sensor node have to be as energy efficient as possible to extend the lifetime (in case of batteries) or enable perpetual operation (in case of EHSs).

Today's WSNs are complex systems which use many different sensors to capture the state of the environment around them. The nodes of these networks are mostly designed to be as efficient as possible for a specific application area. However, the complexity of the WSN makes it very hard to select the most efficient hardware for a given application. Therefore, we present the concept of a simulation environment which can be used to simulate a complete WSN including energy harvesting, communication, power-awareness, location-awareness, and dynamic power management. Each node is composed of single modules which can be exchanged easily to determine the appropriate hardware circuit.

The paper is organized as follows: Section II discusses a lot of important topics concerning the implementation of a simulation environment for energy harvesting WSNs. It describes different energy harvesting technologies and energy storage components. Furthermore, it shows related simulation environments which can be used to simulate WSNs. Section III describes briefly a tier model which can be used to model energy harvesting systems. Section IV shows the design of the simulation environment and explains its modules. Section V explains the simulation setup of a first simulation and shows the initial results. Finally, Section VI concludes the paper and shows directions for future work.

II. Related Work

As already mentioned, EHSs convert environmental energy into electrical energy. The amount of converted energy depends on the type of the environmental energy source and on the efficiency of the energy harvesting device. Table I shows a list of commonly used energy harvesting technologies and their power densities [5], [6]. There is a high variation of the power densities between the different technologies (factor 15000). Furthermore, there is a high variation of the power densities of solar cells depending on the location. Deployed outdoor, a solar cell has a 1000 times higher power density compared to an indoor deployment at noon.

A disadvantage of environmental energy sources is the discontinuous power output. An energy storage component is needed to bridge periods with insufficient power output. The capacity of the energy storage component must be sufficient to enable perpetual operation of the sensor node. It depends

Table I
POWER DENSITIES OF DIFFERENT ENERGY HARVESTING TECHNOLOGIES [5], [6].

Harvesting technology	Power density
Solar cells (outdoor at noon)	10-15 mW/cm^2
Solar cells (indoor) [7]	$\approx 10\,\mu W/cm^2$
Piezoelectric (shoe inserts)	$\approx 300\,\mu W/cm^3$
Vibration (small microwave oven)	$\approx 100\,\mu W/cm^3$
Thermoelectric ($10°C$ gradient)	$\approx 40\,\mu W/cm^3$
Acoustic noise (100dB)	$\approx 1\,\mu W/cm^3$

on the power consumption of the sensor node and the power output profile of the energy harvesting device. Therefore, knowledge about the chronological trend of environmental energy source is necessary to be able to dimension the energy storage component.

There are different types of energy storage components. They range from rechargeable batteries, e.g. Nickel-Metal hydride (NiMH), Nickel-Cadmium (NiCd), Lithium-Ion (Li-Ion), and Lithium-Ion Polymer (LiPo), to double layer capacitors (DLCs). Table II shows the most common types and their characteristics. The non-rechargeable alkaline battery is listed for completeness.

Compared to rechargeable batteries, DLCs have a very low energy density. However, the number of charge-discharge cycles which are possible till the usable capacity is reduced to $80\,\%$ of its nominal value is much higher. A typical number of charge-discharge cycles of DLCs is 500000. In contrast, the number of charge-discharge cycles of rechargeable batteries typically ranges from 200 to 500.

Table II shows also the maximum and minimum voltage of a discharge process. The terminal voltage of the energy storage device is equal to the maximum voltage if it is fully charged. The terminal voltage of it drops during discharge down to the minimum voltage. A voltage conversion circuit is necessary, if a sensor node needs a constant supply voltage or the terminal voltage of the energy storage component exceeds the operating range. The supply voltage of the sensor node should be as low as possible in order to save energy which is explained in the following. Most sensor nodes contain a microcontroller. Typically, this microcontroller is a digital CMOS (complementary metal oxide semiconductor) circuit. Such a circuit has a static and a dynamic power consumption [11]. Leakage and bias currents are the reason for the static power consumption. Typically, the static power consumption is small and can be neglected in systems with a total power consumption of more than $1\,\mathrm{mW}$. The dynamic power consumption of a CMOS circuit can be calculated as shown in (1). The switching elements (the gates) of the CMOS circuit have a common switching capacity C.

$$P_{Dynamic} = C \cdot f \cdot V_{Supply}^2 \tag{1}$$

The clock frequency f of the CMOS circuit has linear influence and the supply voltage V_{Supply} has quadratic influence on the dynamic power consumption $P_{Dynamic}$. This quadratic influence is the reason why energy can be saved if the supply voltage of the sensor node is reduced.

Furthermore, energy can be saved if the inactive hardware components are switched off completely. We have described this idea in [12]. A power switch for each component is necessary. These switches are controlled by the software running on the microcontroller. For example, if a measurement should be done, the software activates the power supply of the sensor, waits until the sensor has finished the measurement, reads the data from the sensor, and deactivates the power supply of the sensor. Therefore, the hardware components are only powered if they are needed by the software and the stand-by power is saved.

The combination of the controllable power switches of each component and the minimization of the needed supply voltage results in component-aware dynamic voltage scaling (CADVS). In [13] we described the theory of this low-power principle in detail and in [14] we evaluated six different voltage supply circuits with and without applying CADVS. The results showed that there is no optimal voltage supply circuit which fits for all applications. The duty-cycle of the application determines the ratio of the active time to the active time and the sleep time. During the active time, the application (of a sensor node) measures, preprocesses, and transmits the data. The sleep time is used to save as much energy as possible. During the sleep time, the hardware of a sensor node is in a low-power state and the functionality is reduced to a minimum. During this low-power state the power consumption of the node is lower than the quiescent power of some voltage converter circuits. Therefore, the duty-cycle of the application is significant to select the appropriate converter circuit. However, the determination of the duty-cycle is not always easy, because it depends on the measurement rate of the node itself, the network traffic, and in power-aware WSNs also on the power state of the energy storage component.

We present a simulation environment written in SystemC-AMS which is used to simulate the whole WSN system from hardware to software. SystemC is a language which is used to model and simulate complex systems consisting of hardware and software components. SystemC has been used many times to model WSN systems [15]–[20]. However, these focus mainly on communication or power consumption without energy harvesting and do not simulate analog-mixed-signal (AMS) systems. SystemC-AMS has been developed to be able to model such systems. It is possible to compose a system out of single hardware components (e.g. resistor, capacitor). Therefore, it is possible to model real hardware systems. A disadvantage of such a model is the high computational complexity. The trade-off between the computational complexity and the model's level of detail results in a mixture of AMS-Modules and higher level modules. SystemC-AMS has already been used to model

Table II
DIFFERENT TYPES OF ENERGY STORAGE COMPONENTS AND THEIR CHARACTERISTICS TO SUPPLY SENSOR NODES [8], [9].

Characteristic	Alkaline	NiCd	NiMH	Lead Acid	LiIon	LiPo	DLC [10]
Energy density [Wh/kg]	≈ 145	45-80	60-120	30-50	110-160	110-130	4.5
Charge-discharge cycles	—	1500	300-500	200-300	500-1000	300-500	500000
Typical fast charge time	—	1h	2-4h	8-16h	2-4h	2-4h	≈ 1 min
Overcharge tolerance	—	moderate	low	high	very low	low	low
Maximum cell voltage [V]	1.5	1.48	1.38	2.1	4.1	4.1	2.7
Minimum cell voltage [V]	0.7	0.8	0.8	1.75	2.5	2.8	0.0
Operating Temp. [°C]	-30 to 55	-40 to 40	-20 to 60	-20 to 60	-20 to 60	0 to 60	-40 to 65

WSNs. Most of them focus on the transceiver circuits and communication [21]–[23]. The work in [24] describes the design methodology of the AMS path of embedded systems, especially the analog-to-digital converter.

A very interesting proposal is presented in [25]. They describe a framework for evaluating different power-management approaches in WSNs using SystemC. They also envisioned including energy harvesting technologies into the simulation framework. However, they haven't presented any results. Furthermore, they do not use the AMS extension of SystemC to model down to hardware level.

There are also combined methodologies which use SystemC models and circuit simulators. In [26] they combine a SystemC simulation and a circuit simulation to get accurate simulation results. The disadvantage of this solution is that the circuit models are independent to the SystemC models and two different languages are used.

III. ENERGY HARVESTING SYSTEM MODEL

This section briefly describes a tier model for EHSs. We have introduced this model in [27]. It consists of five different tiers. Each tier is interacting with the bordering tiers and provides a specific functionality. It encapsulates the functionality and ensures a safe power flow through the whole model. The tiers consist of elements which represents different hardware blocks of the EHS. Typically, two different elements form a complete tier.

Fig. 1 shows the model of a simple EHS. The input measurement, the power switch, the MPPT, and the charge element form the input stage. It is shown on the left side of the figure. This stage is responsible for the measurement, control, and optimal storage of the input power which is provided by the solar cell. The combination of the discharge, the stabilization, the power switch, and the output measurement element form the output stage. It is shown on the right side of the figure. This stage is responsible for the stabilization, control, and measurement of the output power which powers the sensor node.

The dotted and the dash-dotted lines indicate the information flow in the model. The dash-dotted lines show the flow of measurement information concerning the state of the

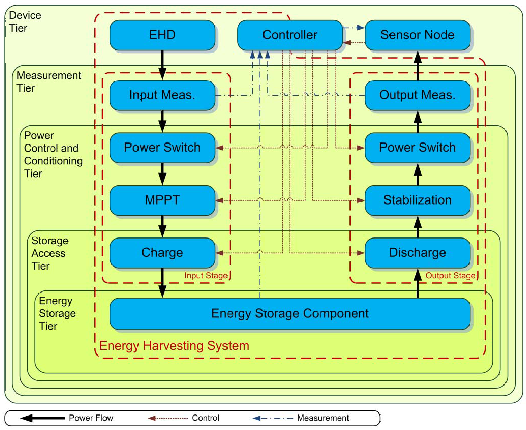


Figure 1. Tier model of a simple EHS.

EHS. The dotted lines show the flow of control information. Therefore, the behavior of the elements of the tiers can be changed and the performance of the EHS can be enhanced.

IV. SYSTEM DESIGN

This section shows the design of the implemented simulation environment. It consists of single modules which are composed to more complex modules. Figure 2 shows the basic structure of it. It can be seen that a node consist of five different modules. Each of these modules consit of lower-level modules or components.

The transceiver transmits and receives the messages. The focus of it is the timely correct transmission and receiving of the messages. It is planned to use different propagation models to be able to compare them.

The sensor module is used to measure the physical quantities which are specified by the application area of the sensor node, e.g. temperature, humidity, light. The inclusion of the sensor is important if the transmission rate depends on the measured data.

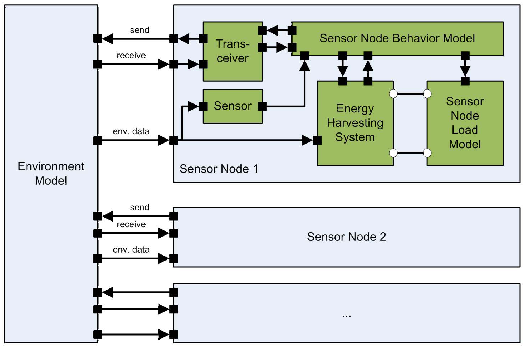


Figure 2. Simulation environment of an energy harvesting WSN.

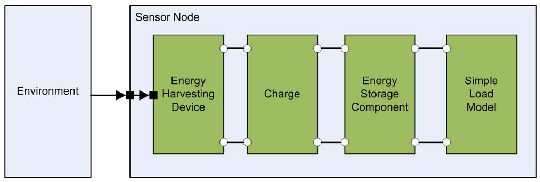


Figure 3. Simulation setup of the energy harvesting system of a simple sensor node.

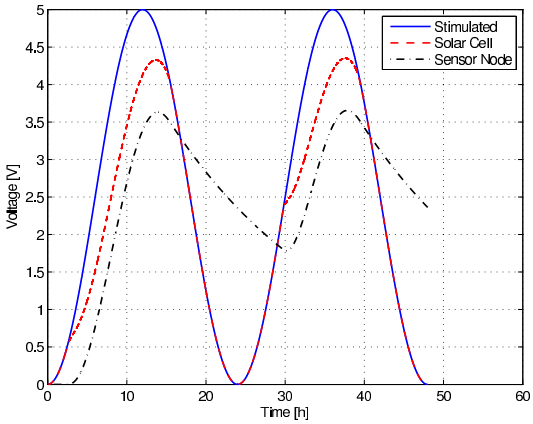


Figure 4. Simulation of the power supply of a simple sensor node showing the system voltages.

The energy harvesting module is composed out of several sub-modules. These modules are connected using electrical linear network (ELN) terminals and nodes. The module uses the structure of the tier model for energy harvesting systems as explained in Section III. Using this structure, it is very easy to exchange single modules e.g. the energy storage component or the stabilization module. Therefore, it can be evaluated which components or modules fit best for a given application.

The sensor node load model simulates the load of the sensor node by adapting its internal resistance to a given value. This value depends on the state of the sensor node behavior model.

Finally, the sensor node behavior model includes the software part of a sensor node. The software is simulated by this module. The simulation includes an interaction with the transceiver, the sensors, and the EHS.

Each sensor node module is connected to the environment model. It connects each node to the other ones. It is responsible for the wireless communication including collision detection and handling. Furthermore, the environmental data, e.g. solar irradiation, temperature, and humidity, are forwarded to the nodes. It should be mentioned that these data can be dependent on the location of the node in the environment (simulation of location-aware behavior).

This structure of the simulation environment enables the simulation of a complete WSN system including energy harvesting, communication, power-awareness, location-awareness, and dynamic power management.

V. First Results

This section shows first results of the simulation of a simple sensor node. The simulation setup focuses on the energy harvesting system of this node. Figure 3 shows the modules of the simulation. The transceiver, sensor, and the sensor node behavior model are excluded completely, because this simulation should only show the feasibility of the low-level simulation of the energy harvesting system.

The energy harvesting device consists of an ideal voltage source combined with an internal resistor. It is stimulated by the environment using a sinusoidal trace with a frequency of one day. It is a very simple model of the sunlight.

The charge module is also kept very simple. It consists of a diode that prevents the energy storage component from discharging via the energy harvesting device.

A DLC is used as energy storage component. It is modeled with an ideal capacitor and a series resistor and a parallel resistor to model the leakage current.

The simple load model consists of a resistor simulating an ohm resistive load. It is clear that a real sensor node cannot be modeled by a constant resistive load. However, this simulation setup should mainly show the feasibility of the simulation of low-level components using the SystemC-AMS language.

Figure 4 shows the voltage traces at different components. The solid blue trace shows the stimulated voltage of the energy harvesting device. In this case, it is the open circuit voltage of the solar cell. The dashed red trace shows the output voltage of the energy harvesting device. It can be seen that this voltage is lower than the open circuit voltage if

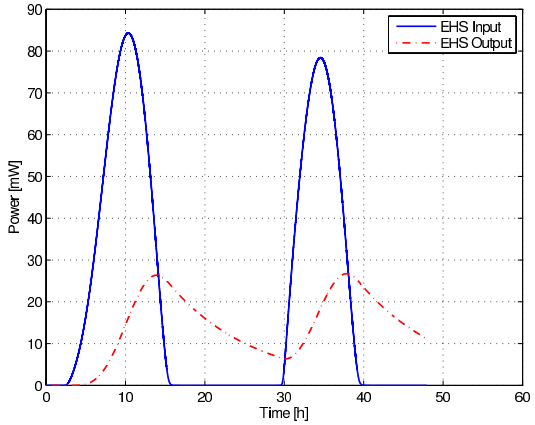


Figure 5. Simulation of the power supply of a simple sensor node showing the power flow.

the solar cell powers the EHS and therefore, when the DLC is charged. The dash-dotted black trace shows the output voltage of the EHS. In this case it is equal to the voltage of the DLC. It can be seen that the voltage of the DLC drops during the night according to a discharge trace of a capacitor.

Figure 5 shows the output power of the energy harvesting device and the power consumption of the simple sensor node. The solid blue trace shows the power generated by the energy harvesting device. The dashed red trace shows the power consumption of the simple sensor node. It can be seen that the energy harvesting device can only charge the DLC if the output voltage of the energy harvesting device is higher than the voltage of the DLC and the forward voltage of the diode. The second peak of the input power of the EHS is lower, because of the already partly charged DLC.

VI. Conclusion and Future Work

This work presents a simulation environment written in SystemC-AMS to simulate complete WSNs from the application down to the hardware. Lot of other work has been analyzed to be able to implement a detailed model of such WSN networks. In previous work we have shown that the selection of the optimal hardware depends also on the software running on the sensor node. Therefore, this simulation environment can be used to be able to design better WSN hardware and software. The simulation of a simple sensor node including an EHS has shown that the simulation structure allows a simulation of low-level hardware consisting of single components.

Future work will focus on the implementation of the sensor node behavior model, the communication module and the environment. Also different types of EHS should be implemented and simulated. Then, the simulation environment will have the ability to simulate complete WSNs using energy harvesting.

References

[1] K. Langendoen, A. Baggio, and O. Visser, "Murphy loves potatoes: experiences from a pilot sensor network deployment in precision agriculture," *Parallel and Distributed Processing Symposium, 2006. IPDPS 2006. 20th International*, p. 8 pp., apr. 2006.

[2] A. Lindgren, C. Mascolo, M. Lonergan, and B. McConnell, "Seal-2-seal: A delay-tolerant protocol for contact logging in wildlife monitoring sensor networks," *Mobile Ad Hoc and Sensor Systems, 2008. MASS 2008. 5th IEEE International Conference on*, pp. 321 –327, sep. 2008.

[3] K. Lorincz, B.-r. Chen, G. W. Challen, A. R. Chowdhury, S. Patel, P. Bonato, and M. Welsh, "Mercury: a wearable sensor network platform for high-fidelity motion analysis," in *SenSys '09: Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems*. New York, NY, USA: ACM, 2009, pp. 183–196.

[4] N. Xu, S. Rangwala, K. K. Chintalapudi, D. Ganesan, A. Broad, R. Govindan, and D. Estrin, "A wireless sensor network for structural monitoring," in *SenSys '04: Proceedings of the 2nd international conference on Embedded networked sensor systems*. New York, NY, USA: ACM, 2004, pp. 13–24.

[5] V. Raghunathan, A. Kansal, J. Hsu, J. Friedman, and M. Srivastava, "Design considerations for solar energy harvesting wireless embedded systems," in *Proceedings of the 4th international symposium on Information processing in sensor networks*. IEEE Press, 2005, p. 64.

[6] P. He, Q. Cui, and X. Guo, "Efficient solar power scavenging and utilization in mobile electronics system," *Green Circuits and Systems (ICGCS), 2010 International Conference on*, pp. 641 –645, jun. 2010.

[7] S. Roundy, D. Steingart, L. Frechette, P. Wright, and J. Rabaey, "Power sources for wireless sensor networks," in *Wireless Sensor Networks*, ser. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 2004, vol. 2920, pp. 1–17.

[8] D. Linden and T. B. Reddy, *Handbook of Batteries*, 3rd ed. New York: McGraw-Hill, 2002.

[9] C. Kompis and S. Aliwell, "Energy harvesting technologies to enable remote and wireless sensing," http://host.quid5.net/ koumpis/pubs/pdf/energyharvesting08.pdf, June 2008.

[10] Maxwell Technologies, "Bcap0310 p270 t10 - datasheet - bc power series radial d cell 310f ultracapacitor," http://www.maxwell.com/docs/DATASHEET_DCELL_POWER_1014625.PDF, 2011 February, 1014625.1.

[11] J. Pouwelse, K. Langendoen, and H. Sips, "Dynamic voltage scaling on a low-power microprocessor," in *Proceedings of the 7th annual international conference on Mobile computing and networking*, ser. MobiCom '01. New York, NY, USA: ACM, 2001, pp. 251–259.

[12] L. B. Hörmann, P. M. Glatz, C. Steger, and R. Weiss, "A wireless sensor node for river monitoring using msp430 and energy harvesting," in *European DSP in Education and Research Conference. Proceedings.* Texas Instruments, 2010, pp. 140–144.

[13] ——, "Energy efficient supply of wsn nodes using component-aware dynamic voltage scaling," in *Wireless Conference (EW), 2011 European*, April 2011, pp. 147–154.

[14] ——, "Evaluation of component-aware dynamic voltage scaling for mobile devices and wireless sensor networks," in *World of Wireless Mobile and Multimedia Networks (WoWMoM), 2011 IEEE International Symposium on a*, june 2011, p. to be published.

[15] J. Haase, M. Damm, J. Glaser, J. Moreno, and C. Grimm, "Systemc-based power simulation of wireless sensor networks," in *Specification Design Languages, 2009. FDL 2009. Forum on*, sept. 2009, pp. 1 –4.

[16] W. Du, F. Mieyeville, and D. Navarro, "Idea1: A systemc-based system-level simulator for wireless sensor networks," in *Wireless Communications, Networking and Information Security (WCNIS), 2010 IEEE International Conference on*, june 2010, pp. 618 –622.

[17] W. Du, D. Navarro, and F. Mieyeville, "A simulation study of ieee 802.15.4 sensor networks in industrial applications by system-level modeling," in *Sensor Technologies and Applications (SENSORCOMM), 2010 Fourth International Conference on*, july 2010, pp. 311 –316.

[18] W. Du, D. Navarro, F. Mieyeville, and F. Gaffiot, "Towards a taxonomy of simulation tools for wireless sensor networks," in *Proceedings of the 3rd International ICST Conference on Simulation Tools and Techniques*, ser. SIMUTools '10, 2010, pp. 52:1–52:7.

[19] D. Navarro, W. Du, F. Mieyeville, and L. Carrel, "Hardware and software system-level simulator for wireless sensor networks," *Procedia Engineering*, vol. 5, pp. 228 – 231, 2010, eurosensor XXIV Conference, Eurosensor XXIV Conference.

[20] W. Du, F. Mieyeville, and D. Navarro, "Modeling energy consumption of wireless sensor networks by systemc," in *Systems and Networks Communications (ICSNC), 2010 Fifth International Conference on*, aug. 2010, pp. 94 –98.

[21] M. Vasilevski, N. Beilleau, H. Aboushady, and F. Pecheux, "Efficient and refined modeling of wireless sensor network nodes using systemc-ams," in *Research in Microelectronics and Electronics, 2008. PRIME 2008. Ph.D.*, 22 2008-april 25 2008, pp. 81 –84.

[22] M. Vasilevski, F. Pecheux, N. Beilleau, H. Aboushady, and K. Einwich, "Modeling and refining heterogeneous systems with systemc-ams: Application to wsn," in *Design, Automation and Test in Europe, 2008. DATE '08*, march 2008, pp. 134 –139.

[23] M. Vasilevski, H. Aboushady, F. Pecheux, and L. de Lamarre, "Modeling wireless sensor network nodes using systemc-ams," in *Microelectronics, 2007. ICM 2007. Internatonal Conference on*, dec. 2007, pp. 53 –56.

[24] M. Farooq, S. Adhikari, J. Haase, and C. Grimm, "Modeling methodology in systemc-ams for embedded analog mixed signal systems," in *Proceedings of the 8th International Conference on Frontiers of Information Technology*, ser. FIT '10, 2010, pp. 27:1–27:6.

[25] R. Thomasius and S. Guttowski, "A systemc based framework for the evaluation of proactive power-management approaches in distributed energy harvesting systems," in *Sensor Technologies and Applications (SENSORCOMM), 2010 Fourth International Conference on*, july 2010, pp. 348 –351.

[26] T. Kirchner, N. Bannow, and C. Grimm, "Analogue mixed signal simulation using spice and systemc," in *Design, Automation Test in Europe Conference Exhibition, 2009. DATE '09.*, april 2009, pp. 284 –287.

[27] L. B. Hörmann, P. M. Glatz, C. Steger, and R. Weiss, "Designing of efficient energy harvesting systems for autonomous wsns using a tier model," *Telecommunications(ICT), 2011 18th IEEE International Conference on*, pp. 185–190, may 2011.

A Hardware/Software Simulation Environment for Energy Harvesting Wireless Sensor Networks

Leander B. Hörmann
Inst. for Technical Informatics
Graz University of Technology
Inffeldgasse 16, 8010 Graz
Austria
leander.hoermann@tugraz.at

Philipp M. Glatz
Embedded Software Group
Delft University of Technology
Mekelweg 4, 2628 CD Delft
The Netherlands
p.m.glatz@tudelft.nl

Karima B. Hein
Inst. for Technical Informatics
Graz University of Technology
Inffeldgasse 16, 8010 Graz
Austria
hein@tugraz.at

Christian Steger
Inst. for Technical Informatics
Graz University of Technology
Inffeldgasse 16, 8010 Graz
Austria
steger@tugraz.at

Reinhold Weiss
Inst. for Technical Informatics
Graz University of Technology
Inffeldgasse 16, 8010 Graz
Austria
rweiss@tugraz.at

ABSTRACT

Wireless sensor networks (WSNs) consist of wirelessly communicating nodes with an autarkic power supply for each node. Typically, the consumable energy of these nodes is very limited. Energy harvesting systems (EHSs) can be used to extend the lifetime or even enable perpetual operation of the sensor nodes. Applicable energy-aware WSN protocols and applications usually raise the complexity such that rough calculations are not sufficient any more. Simulation-based analysis is needed to cope with the complexity of hardware/software interaction and its implications.

This work presents a simulation environment which enables combined simulation and performance evaluation of complete WSNs including the sensor nodes' application software, the energy harvesting enhanced hardware, the wireless network communication, and the environment of the sensor nodes.

Categories and Subject Descriptors

C.4 [**Perfomance of Systems**]: Modeling techniques; I.6.8 [**Simulation and Modeling**]: Types of Simulation—*combined*

General Terms

Design, Performance

Keywords

Energy harvesting, simulation environment, wireless sensor networks

PE-WASUN'12, October 24–25, 2012, Paphos, Cyprus.

1. INTRODUCTION

Wireless sensor networks (WSNs) are used to collect information about their environment if no wired infrastructure is available for communication and power supply. Application areas are manifold: Environmental monitoring [1], precision agriculture [15], wildlife monitoring [17], human health-care [18], or structural health monitoring [26] are only a few examples.

Typically, each sensor node has a dedicated power supply. Basically, there are two possibilities to supply a sensor node. First, conventional batteries can be used which are cheap but cannot be recharged. This means, that the lifetime of the battery powered sensor nodes is limited. Furthermore, the leakage current of a battery makes it impossible to extend the operating time at will by reducing the average power consumption of the nodes [5]. Second, so called energy harvesting systems (EHSs) can be used to supply the sensor nodes. The EHSs convert environmental energy into electrical energy and perpetual operation is possible. An energy storage component (ESC) is needed to supply the mote during periods with insufficient harvestable energy. Typically, such EHSs are used for long-term deployments of sensor nodes at hard-to-access locations. However, the power density of environmental energy sources is often very low and the resulting available power and energy is limited.

The sensor node has to be energy efficient in order to extend the lifetime (conventional battery supply) or enable perpetual operation (EHS supply). Especially in case of using an EHS supply, the difference between harvested energy and consumed energy is often very small. Duty cycling can be used to conserve additional energy, but this depends on the application area. The sensor node is in sleep state most of the time. It is only activated for measurement, computation and transmission of the results.

Today's WSNs are complex systems and the selection of the most energy efficient hardware as well as the design of energy efficient software is difficult. The energy efficiency depends not only on hardware, but also on software. The main problem is to get a realistic simulation of both, software of the sensor nodes and the energy harvesting enhanced hardware. Therefore, we present a simulation environment

which enables performance evaluation of complete WSNs including energy harvesting, sensing, communication, energy-awareness, and dynamic energy management. It implements the concept that we have proposed in [12].

The rest of the paper is organized as follows: Section 2 discusses related work. Section 3 describes the simulation environment and Section 4 shows the simulation results. Section 4.3 describes a simulation scenario of a low-power technique which needs combined hardware and software simulation. Finally, Section 5 concludes the paper and gives directions for future work.

2. RELATED WORK

The simulation environment is written in SystemC-AMS which is used to simulate the overall WSN system including hardware and software. Complex systems consisting of software and hardware components can be modeled and simulated with the SystemC language. As mentioned before, WSNs are complex systems and SystemC has already been used to model and simulate them [3, 4, 9, 19]. However, work in this field usually focuses on communication or power consumption without energy harvesting. Also simulation of analog-mixed-signal (AMS) systems is not considered.

The SystemC-AMS extension enables the combined simulation of system-level based models and analog models. Therefore, models can be composed from single hardware components (e.g. resistors, capacitors ...) and high-level blocks (e.g. processor, memory ...). They can be used to model real hardware systems. However, a disadvantage of low-level models is the high computational complexity. This trade-off between the computational complexity and the model's level of detail often causes a mixture of AMS-modules and high-level models. SystemC-AMS has also been used to model and simulate WSNs. The focus is mainly on the transceiver circuits and communication [23, 24, 25]. The work in [7] describes the design methodology of the AMS part of embedded systems. In particular, it describes the analog-to-digital converter.

The authors of [22] propose a framework for the evaluation of different power-management approaches in WSNs using SystemC. They also suggest the integration of EHSs. However, they present no results and we did not find any other work of them presenting results prior the publication of this work.

3. SIMULATION ENVIRONMENT

The main parts of the simulation environment are the environmental model and the sensor node model. The latter also has two main parts which are the sensor node behavior model and the power supply model. These three main models are described in the following subsection. An overview of the model structure is shown in Figure 2.

3.1 Environmental Model

The environmental model establishes connections between all nodes of the WSN and provides them with location specific information, for example temperature and harvestable energy. Hence, the location of each node has to be stored in this module. The provided information changes during the simulation according to the time of day and other influences. Furthermore, this information may also vary between the sensor nodes depending on the location. To get realistic simulation inputs, we presented an on-site characterization instrument for energy harvesting devices [11]. First results of a solar cell characterization illustrate the importance of such characterizations. Figure 1 shows the trace of the maximum power point (MPP) of two sensor nodes placed south and north of the author's department (northern hemisphere) during a day with good and a day with bad weather conditions. It shows a significant difference of the harvestable energy only at days with good weather conditions.

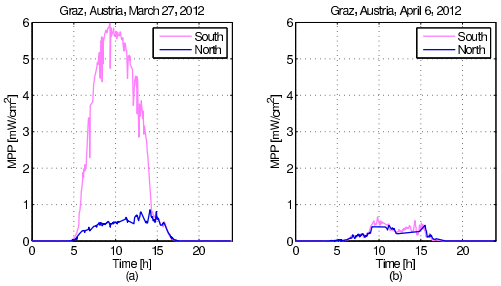


Figure 1: Maximum power point (MPP) of two solar cells placed at different locations. The traces are measured during a day with (a) good weather conditions and (b) bad weather conditions.

The wireless network communication is modeled by the connectivity and channel model. The model provides a carrier sense signal to each node which indicates an ongoing transmission. Depending on the distance between the nodes and the transmission range, not every transmission can be detected by each sensor node. A collision signal indicates a broken transmission to each sensor node in range. A transmitting signal is used to model the temporal behavior of the transmission which is activated during an ongoing transmission. Each sensor node in range of the transmission recognizes it via the receiving signal. The packet data is provided to and from the sensor nodes using the data signal.

3.2 Sensor Node Behavior Model

The behavior of a sensor node is defined by its application software and the access to its hardware modules. Here, the hardware modules are represented by high-level models which reduce the computational complexity. However, the increased current consumption caused by activating these hardware modules is modeled using power states inside the processing module to ensure a realistic power trace.

The software is simulated in an event-driven way similar to TinyOS [16]. This means that the software provides predefined interfaces and uses itself interfaces from the processing module. The supported interfaces are combined to a simulation abstraction layer (SAL). The advantage of such an interface layer is the possibility to execute the application software by the simulator as well as by the sensor node itself if the layer is also provided by the real sensor node. This would enhance the comparability of the results. Furthermore, the emulation of the sensor node's processing hardware increases the simulation speed dramatically. The abstraction principle is shown in Figure 3. The following list shows the functions which have to be implemented by the

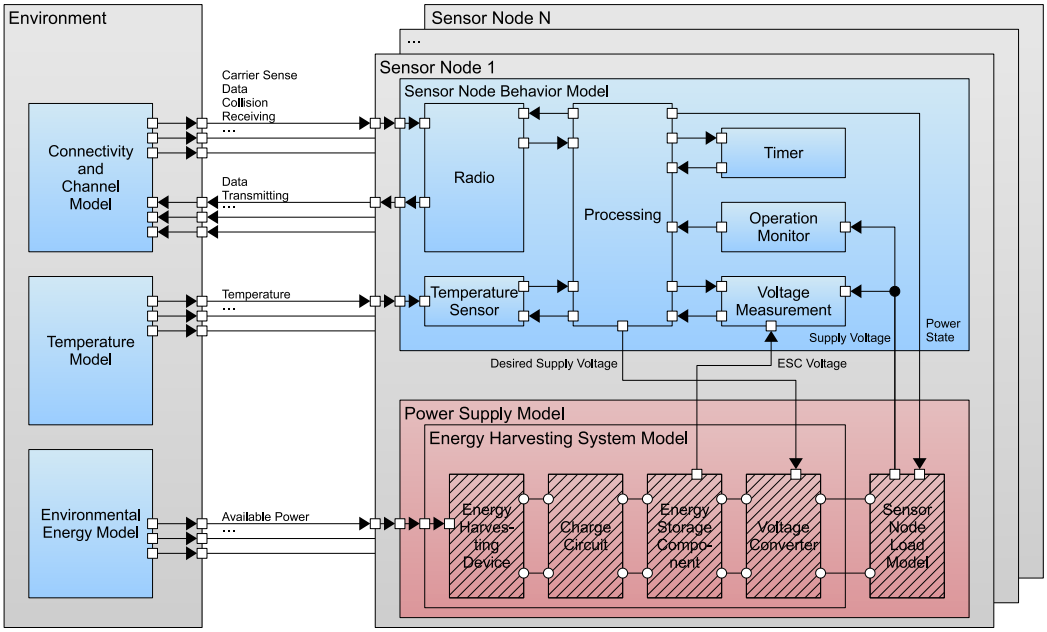


Figure 2: SystemC-AMS structure of the simulation environment including the environmental model and one wireless sensor node in detail. The non-shaded modules indicate conventional SystemC modules and the shaded modules indicate SystemC-AMS modules. The square connectors show SystemC ports and the round connectors show electrical terminals.

application of the sensor node. They are called at certain events.

- `mp_init()`: This function is called at the initialization event after the startup of the microcontroller.
- `mp_timer_fired()`: This function is called at the timer fired event after a preconfigured time has elapsed.
- `mp_tempsens_finished(uint16 temp)`: This function is called in response to the finish of a temperature measurement.
- `mp_radio_received(Message msg)`: This function is called at the event after successfully receiving a message. The source and the destination are encapsulated in the message.
- `mp_radio_transmitted()`: This function is called at the event after the transmission of a message.

The following list shows the functions which can be used by the application of the sensor node (commands).

- `hm_start_timer(uint16 msec)`: This command configures and starts the timer.
- `hm_radio_power_on()`: This command energizes the radio module.
- `hm_radio_power_off()`: This command de-energizes the radio module.
- `hm_radio_send(Message msg)`: This command transmits the message. The source and the destination are encapsulated in the message.
- `hm_tempsens_power_on()`: This command is used to energize the temperature sensor.
- `hm_tempsens_power_off()`: This command de-energizes the temperature sensor.
- `hm_tempsens_start()`: This command starts a temperature measurement.

The radio module and the temperature sensor module are the interface of the sensor node to its environment. The timer module is used to generate periodic events. The operation monitor module supervises the supply voltage and starts or stops the application software in case of exceeding or falling below the minimum supply voltage of the microcontroller. Finally, the voltage measurement module simulates the analog-to-digital conversion of the microprocessor's supply voltage.

3.3 Power Supply Model

The power supply model consists of the energy source model and the load model. Here, the energy source is an EHS using a solar cell as energy harvesting device and double layer capacitors (DLCs) as ESC. The power supply model is similar to the EHS we have already implemented and evaluated in [8]. This EHS is now used to supply a WSN for characterizing different types of energy harvesting devices

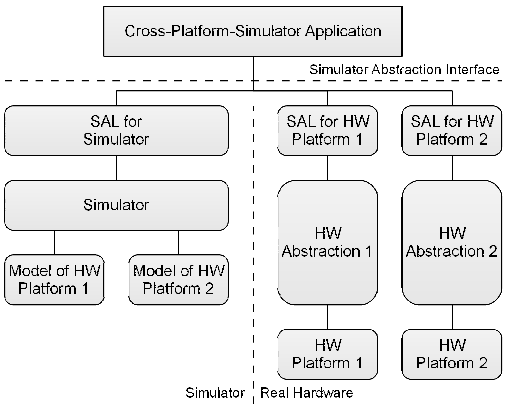


Figure 3: Cross-platform-simulator application support using a simulation abstraction layer (extended from [10]).

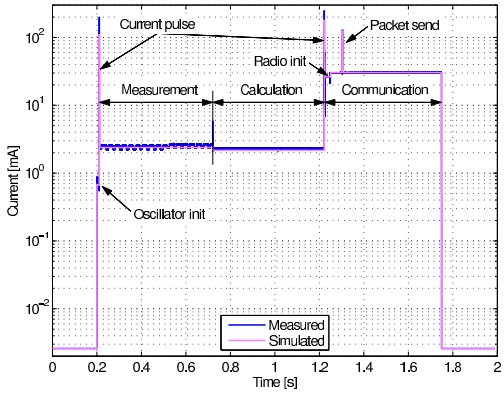


Figure 4: Comparison of the simulated current and the measured current. The supply voltage is set to $3.3\,V$.

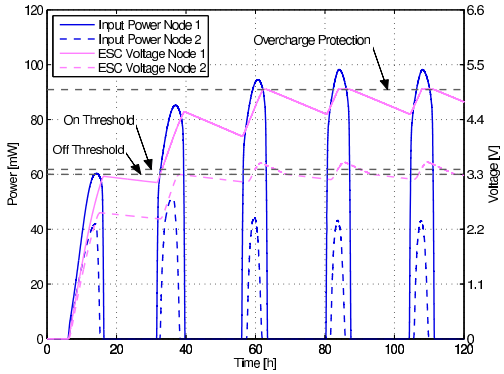


Figure 5: Energy harvesting simulation of two nodes for five days. The solar radiation at the location of node 2 is lower.

as shown in Section 2 [11]. The modules are connected with electrical terminals. The charge circuit is implemented using a simple diode which prevents discharging via the solar cell. However, the implementation of the non-linear hardware components (e.g. diode) was difficult, because the SystemC-AMS extension only provides a linear simulator. Therefore, the diode was implemented using a variable resistor combined with a look-up table. The resistance of the diode depends on its voltage drop and is adapted at each simulation step. Furthermore, the charge circuit contains an overcharge protection using a voltage dependent shunt resistor. The ESC consists of two DLCs connected in series. Each of them has a capacity of $350\,F$ and a maximum voltage of $2.5\,V$. We have chosen and adapted the DLC model from [6] to be able to use it in SystemC-AMS. The voltage converter is described by a model of a linear dropout regulator with a constant output voltage of $3.3\,V$. The load model uses a variable resistor to adapt the power consumption of the sensor node on the current power state. Therefore, the power state information is provided to this module.

4. RESULTS

This section shows the simulation results of a sensor node's detailed current consumption, the simulation results of two sensor nodes supplied for five days by energy harvesting, and the simulation results of component-aware dynamic voltage scaling applied at two energy harvesting sensor nodes. The results show the ability of the simulation environment to simulate detailed and long-term application scenarios.

4.1 Detailed Simulation Results

The simulation shows the current consumption of a sensor node during an activation interval. Figure 4 compares the simulated current and the measured current. The different power states during the activation interval are shown. The main phases of the activation interval are the sleep phase, the measurement phase, the computation phase, and the communication phase. The sleep phase takes $8.5\,s$. The other main phases take $0.5\,s$ each. One packet is sent and one is received during the communication phase. The highest differences occur during the power-on of the temperature sensor and the radio module. However, the mean relative error of the current consumption during the active phases is $-1.92\,\%$. The sleep phase is not considered here, because the constant current is modeled accurately.

4.2 Five-Day Simulation

The long-term simulation shows two sensor nodes supplied by solar cells. The ESCs of both sensor nodes are empty at simulation start. Sensor node 1 is virtually placed at a location with better irradiation constraints. The open circuit voltage of its solar cell is $8.34\,V$. The open circuit voltage of sensor node 2's solar cell is only $4.17\,V$. Figure 5 shows the input powers and the voltages of the ESC of both sensor nodes. The input power of node 1 increases with each day, because of the increasing voltage of its ESC, the DLCs. The solar cell is operated closer and closer to the MPP of $7.4\,V$. However, the input power of node 2 decreases after the second day, because the MPP is already exceeded. The

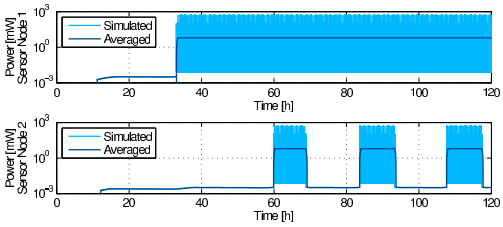


Figure 6: Energy harvesting simulation of two sensor nodes for five days showing the normal and the averaged power consumption of node 1 and 2.

MPP of node 2's solar cell is at $3.3\,V$, because of the lower solar radiation. The reason for these two effects is the direct connection of the solar cell and the DLCs. No MPP tracker is used. This reduces the efficiency of the EHS, but simplifies the hardware significantly. Only a simple diode is used as discharge protection circuit. Otherwise, if no discharge protection is used, the DLCs would be discharged via the solar cell during periods of insufficient available energy.

Both sensor nodes are turned on at a DLCs voltage of $3.4\,V$ and turned off at $3.3\,V$. Sensor node 1 reaches the power-on threshold at the second day. The DLCs voltage doesn't fall below the power-off threshold during the rest of the simulation. Sensor node 2 reaches the power-on threshold at the third day. However, the energy stored in the DLCs is not enough to supply node 2 during the complete night. It can be clearly seen in Figure 6. It shows the power consumption of both sensor nodes. Also, it depicts the average power consumption.

4.3 Simulation of Component-Aware Dynamic Voltage Scaling

This simulation scenario shows the ability of adapting the simulation environment on specific hardware and software features.

DVS is commonly used to adapt the supply voltage of a processor depending on its clock frequency. The clock frequency can be reduced if the workload of the processor is low. The authors of [2] and [21] describe the design issues for DVS systems and various DVS algorithms, respectively. A DVS platform using an MSP430 ultra-low power microcontroller is introduced by the authors of [20]. They use only components of the shelf to vary the supply voltage of the microcontroller. However, they do not consider the efficiency and the leakage currents of the voltage converter.

A technique to save energy is to turn on and off the hardware components (e.g. sensors, radio module) according to their usage. The method can be enhanced by adapting the supply voltage of the sensor node to the minimum supply voltage of the activated hardware components. This is called component-aware dynamic voltage scaling (CADVS) which we have introduced in [13] and evaluated with real hardware in [14]. We have shown that the energy savings also depends on the leakage and quiescent current of the voltage conversion circuit. CADVS adapts the supply voltage of the complete wireless sensor node. A block diagram of a simple CADVS enhanced wireless sensor node's hardware is shown in Figure 7.

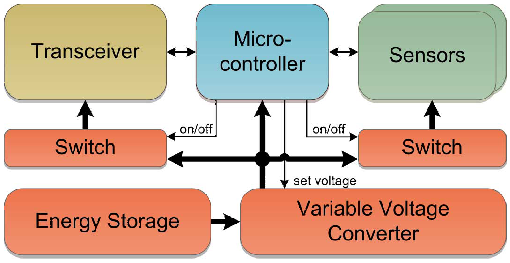


Figure 7: Structure of a CADVS enhanced wireless sensor node [14]. The power supply of the sensors and of the transceiver can be switched on and off by the microcontroller. Furthermore, the variable voltage converter is adjusted by the microcontroller.

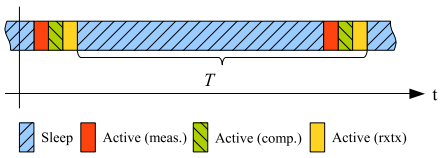


Figure 8: Chronological sequence of an interval T.

4.3.1 *Sensor Node*

The application scenario of the simulated WSN is to measure the environmental temperature at a specified region and transmits the information to a base station. Each sensor node consists of three main components which are the temperature sensor, the radio module, and the microcontroller. A detailed description of the hardware can be found in [14]. A duty cycle is applied which is shown in Figure 8. One interval consists of a sleep phase, a measurement phase, a computation phase, and a transmission phase. The sensor node has been implemented and the average current consumption of the different phases has been measured. This is shown in Table 1.

The sensor node is supplied by solar cells and two DLCs connected in series are used as ESC. The charge circuit is a simple diode which prevents discharging via the solar cell. Each DLC has a capacity of $350\,F$ and a maximum voltage of $2.5\,V$. The joint capacitance is $175\,F$ and the joint maximum voltage is $5.0\,V$. A voltage converter is needed to supply the node with an adjustable voltage and to prevent it from destruction. Here, we simulate and discuss two linear voltage regulators, one with a constant output voltage of $3.3\,V$ and one with an adjustable output voltage supporting CADVS.

4.3.2 *Network*

The simulated network consists of five sensor nodes and one base station. The five sensor nodes are placed in the region of interest. The structure of the network is shown in Figure 9. Node 1 represents the base station. Each node except the base station is supplied by solar cells. The harvestable energy of each node is different. Node 2 gets most energy and node 6 gets least energy which can be seen in Section 4.3.4.

Table 1: Characteristic of the four different phases of the sensor node.

Phase	Minimum Supply Voltage	Supply Current @ 3.3 V	Supply Current using CADVS	Duration
Sleep	1.8 V	2.59 μA	0.84 μA	variable
Temperature measurement	3.3 V	2.39 mA	2.39 mA	0.51 s
Calculation	1.8 V	2.17 mA	0.97 mA	0.501 s
Communication (receive)	2.4 V	29.93 mA	27.02 mA	0.527 s
Communication (transmit)	2.4 V	156.6 mA	103 mA	4.4 ms

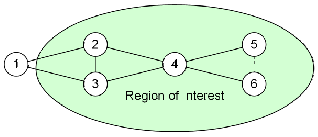


Figure 9: Structure of the simulated network. Node 1 is the base station.

4.3.3 Energy-Aware Application

The energy-aware application is used to measure the temperature of each sensor node in the region of interest by adapting the duty cycle of the whole network to the stored energy of the sensor nodes. At the beginning of the communication phase, the base station transmits a synchronization beacon to all sensor nodes which includes the network duty cycle. This information is used by the sensor nodes to adapt their sleeping schedule. Therefore, the nodes are kept synchronized. At the end of the communication phase, the nodes transmit the previous measured temperature to the base station. Each time a temperature measurement is performed, the sensor node measures also the voltage of its ESC and determines the state-of-charge (SoC). Each neighboring node updates the SoC of the source node. This information is also transmitted to the base station. The base station stores the SoC of each sensor node, calculates the next duty cycle, and transmits it at the next synchronization message. To calculate the next duty cycle, the base station uses the lowest SoC value of all sensor nodes. The SoC is calculated using the measured voltage of the ESC as mentioned before. The energy stored in a capacitor can be calculated as shown in (1).

$$E_{Capacitor} = \frac{1}{2} \cdot C \cdot V_C^2 \tag{1}$$

The SoC varies between 0 and 100 % and it is mapped according to (2). Due to the fact that the minimum input voltage of the linear voltage regulators is 3.3 V, an SoC of 0 % is mapped to this voltage. The maximum voltage of the ESC is 5.0 V which is mapped to 100 %.

$$SoC = \frac{E}{E_{max}} = \frac{E - E_{3.3V}}{E_{5.0V} - E_{3.3V}} = \frac{V_C^2 - 10.89\,V^2}{14.11\,V^2} \tag{2}$$

The duty cycle is adjusted by adapting the sleep time of the interval. The sleep time is calculated according to (3).

$$t_{sleep} = \max\left(500\,s - \frac{2\,s - 500\,s}{50\,\%} \cdot SoC\ ,\ 2\,s\right) \tag{3}$$

The SoC depending on the ESC voltage and the sleep time depending on the SoC are shown in Figure 10.

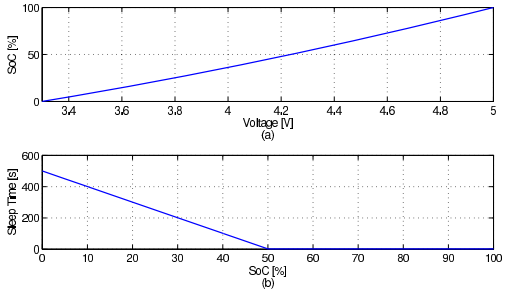


Figure 10: Dependency of (a) the SoC on the ESC voltage and (b) the sleep time on the SoC.

Messages are also forwarded in an energy-aware manner. When transmitting a message to a specific node, the next hop is selected according to the SoC. Therefore, each node has to store the SoC of the neighbors. Due to the fact that each message contains the SoC of the transmitting node, each receiving node gets the SoC information.

4.3.4 Application Scenario Results

This section shows the results of the WSN simulation using the described simulation environment. Figure 11 shows the simulation results using a constant supply voltage and Figure 12 illustrates the use of CADVS. The active part of the interval starts at about 500 ms and ends after 2000 ms. During the first third, the measurement is performed and the sensor node must be supplied with 3.3 V. During the second third, the measurement results are post processed. The voltage of the sensor node can be reduced to 1.8 V. During the last third, the communication is done. Each peak shows a transmission of a message. The voltage can be set to 2.4 V. It can be seen clearly, that the current during the computation phase and during the transmissions is reduced. Therefore, much energy can be saved. The energy consumed during the active period using the constant supply voltage is 71.21 mJ. The energy consumption using the variable voltage is only 46.99 mJ. This is a reduction of 34 %. However, the energy savings comparing the input energy of both voltage converters are lower. The input energy of the constant voltage converter is 92.08 mJ and the input energy of the variable converter is 81.59 mJ. This corresponds to an energy saving of 11.4 %. The reason is that the input current and output current of linear converters are equal when the leakage and quiescent currents are neglected. The reduction of the voltage at the variable voltage converter causes a higher voltage drop at the converter. Therefore, the sav-

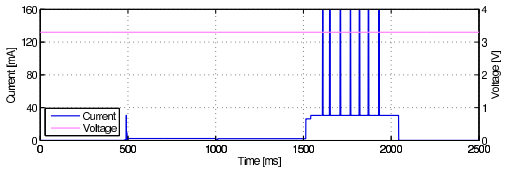


Figure 11: Simulated voltage and current trace using a constant supply voltage of $3.3\,V$.

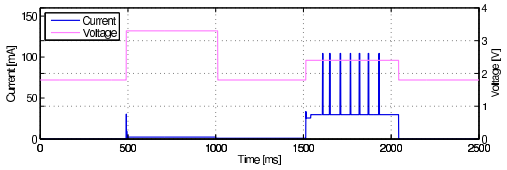
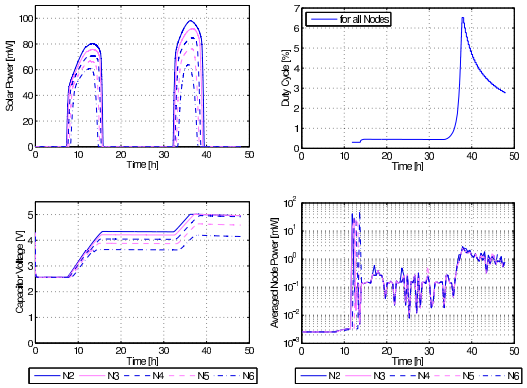


Figure 12: Simulated voltage and current trace using a variable supply voltage supporting CADVS.



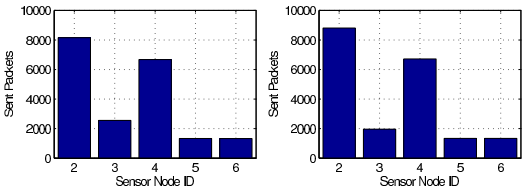
(a) Solar power and capacitor voltage.

(b) Duty cycle and averaged power consumption.

Figure 13: Simulation results of five sensor nodes supplied with energy harvesting during two days. CADVS is applied at all nodes.

ings depend only on the lower current consumption of the wireless sensor node at the lower supply voltages.

The two-day evaluation of the simulation scenario is discussed in the following. Figure 13 shows the input power provided by the solar cells, the voltage trace of the DLCs, the duty cycle of the nodes, and the averaged power consumption of the nodes applying CADVS. It can be seen that the DLCs have an initial voltage of $2.56\,V$ which is too low to start up the sensor node. The sensor node is configured to start up if the voltage of the ESC reaches $3.6\,V$. Figure 14 shows the sent packet statistics of the five sensor nodes. Node 4 has to forward all packets to and from node 5 and 6. The forwarding of the packets to and from node 4 has to be done by node 2 or 3. Most packets are forwarded by node 2, because it harvests more energy than node 3. The significant difference between the simulation using the



(a) Constant supply voltage at $3.3\,V$

(b) Variable supply voltage using CADVS

Figure 14: Sent packet statistics of five sensor nodes.

constant and CADVS is that node 2 forwards more packets using CADVS. In case of the constant supply voltage, 6811 packets are forwarded by node 2. In case of CADVS, 7462 packets are forwarded by node 2 which is an increase of 9.6 %. This means that the node operates more efficiently and more energy is available for network communication.

5. CONCLUSIONS

This work presented a simulation environment for energy harvesting WSNs implemented in SystemC-AMS. It enables a combined simulation and performance evaluation of software and hardware of WSNs. The interactions between software and hardware can be analyzed which can help to optimize them. Furthermore, the simulation scenario showed the ability of evaluating a low-power technique which needs the combined simulation of software and hardware.

Future work will target an enhancement of the simulation environment. Especially, the analog power supply and load model will be optimized for simulating a higher number of nodes and longer periods.

6. ACKNOWLEDGMENTS

This work was supported by the Austrian Research Promotion Agency (FFG) under KIRAS PL3.3 project grant GeoWSN (832344).

7. REFERENCES

[1] G. Barrenetxea, F. Ingelrest, G. Schaefer, M. Vetterli, O. Couach, and M. Parlange. SensorScope: Out-of-the-Box Environmental Monitoring. In *Proceedings of the International Conference on Information Processing in Sensor Networks (IPSN 2008)*, pages 332–343, 2008.

[2] T. D. Burd and R. W. Brodersen. Design Issues for Dynamic Voltage Scaling. In *Proceedings of the 2000 international symposium on Low power electronics and design (ISLPED 2000)*, pages 9–14, 2000.

[3] W. Du, F. Mieyeville, and D. Navarro. Modeling Energy Consumption of Wireless Sensor Networks by SystemC. In *Fifth International Conference on Systems and Networks Communications (ICSNC 2010)*, pages 94–98, 2010.

[4] W. Du, D. Navarro, and F. Mieyeville. A Simulation Study of IEEE 802.15.4 Sensor Networks in Industrial Applications by System-Level Modeling. In *Fourth International Conference on Sensor Technologies and Applications (SENSORCOMM 2010)*, pages 311–316, 2010.

[5] C. Enz, N. Scolari, and U. Yodprasit. Ultra Low-Power Radio Design for Wireless Sensor Networks. In *IEEE Int. Workshop on Radio-Frequency Integration Technology*, pages 1–17, 2005.

[6] R. Faranda, M. Gallina, and D. Son. A New Simplified Model of Double-Layer Capacitors. In *International Conference on Clean Electrical Power (ICCEP 2007)*, pages 706–710, 2007.

[7] M. Farooq, S. Adhikari, J. Haase, and C. Grimm. Modeling Methodology in SystemC-AMS for Embedded Analog Mixed Signal Systems. In *Proceedings of the 8th International Conference on Frontiers of Information Technology (FIT 2010*, pages 27:1–27:6, 2010.

[8] P. M. Glatz, L. B. Hörmann, C. Steger, and R. Weiss. Designing perpetual energy harvesting systems explained with rivermote: A wireless sensor network platform for river monitoring. *Electronic Journal of Structural Engineering, Special Issue: Sensor Network on Building Monitoring: from Theory to Real Application*, pages 55–66, 2010.

[9] J. Haase, M. Damm, J. Glaser, J. Moreno, and C. Grimm. SystemC-Based Power Simulation of Wireless Sensor Networks. In *Forum on Specification Design Languages (FDL 2009)*, pages 1–4, 2009.

[10] V. Handziski, J. Polastre, J.-H. Hauer, C. Sharp, A. Wolisz, and D. Culler. Flexible Hardware Abstraction for Wireless Sensor Networks. In *Proceedings of the 2nd European Workshop on Wireless Sensor Networks*, pages 145–157, 2005.

[11] L. B. Hörmann, P. M. Glatz, K. B. Hein, M. Steinberger, C. Steger, and R. Weiss. Towards an On-Site Characterization of Energy Harvesting Devices for Wireless Sensor Networks. In *IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops 2012)*, pages 415–418, 2012.

[12] L. B. Hörmann, P. M. Glatz, C. Steger, and R. Weiss. A SystemC-AMS Simulation Environment for the Evaluation of Energy Harvesting Wireless Sensor Networks. In *International Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS 2011)*, pages 247–252, 2011.

[13] L. B. Hörmann, P. M. Glatz, C. Steger, and R. Weiss. Energy Efficient Supply of WSN nodes using Component-Aware Dynamic Voltage Scaling. In *European Wireless Conference (EW 2011)*, pages 147–154, 2011.

[14] L. B. Hörmann, P. M. Glatz, C. Steger, and R. Weiss. Evaluation of Component-Aware Dynamic Voltage Scaling for Mobile Devices and Wireless Sensor Networks. In *IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM 2011)*, pages 1–9, 2011.

[15] K. Langendoen, A. Baggio, and O. Visser. Murphy Loves Potatoes: Experiences from a Pilot Sensor Network Deployment in Precision Agriculture. *20th International Parallel and Distributed Processing Symposium (IPDPS 2006)*, page 8 pp., 2006.

[16] P. Levis, S. Madden, J. Polastre, R. Szewczyk, K. Whitehouse, A. Woo, D. Gay, J. Hill, M. Welsh, E. Brewer, and D. Culler. TinyOS: An Operating System for Sensor Networks. In W. Weber, J. M. Rabaey, and E. Aarts, editors, *Ambient Intelligence*, pages 115–148. Springer Berlin Heidelberg, 2005.

[17] A. Lindgren, C. Mascolo, M. Lonergan, and B. McConnell. Seal-2-Seal: A Delay-Tolerant Protocol for Contact Logging in Wildlife Monitoring Sensor Networks. *5th IEEE International Conference on Mobile Ad Hoc and Sensor Systems (MASS 2008)*, pages 321–327, 2008.

[18] K. Lorincz, B.-r. Chen, G. W. Challen, A. R. Chowdhury, S. Patel, P. Bonato, and M. Welsh. Mercury: A Wearable Sensor Network Platform for High-Fidelity Motion Analysis. In *Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems (SenSys 2009)*, pages 183–196, 2009.

[19] D. Navarro, W. Du, F. Mieyeville, and L. Carrel. Hardware and Software System-Level Simulator for Wireless Sensor Networks. *Procedia Engineering*, 5:228–231, 2010. Eurosensor XXIV Conference.

[20] H. C. Powell, A. T. Barth, and J. Lach. Dynamic Voltage-Frequency Scaling in Body Area Sensor Networks Using COTS Components. In *Proceedings of the Fourth International Conference on Body Area Networks (BodyNets 2009)*, pages 15:1–15:8, 2009.

[21] T. Simunic, L. Benini, A. Acquaviva, P. Glynn, and G. De Micheli. Dynamic Voltage Scaling and Power Management for Portable Systems. In *Proceedings of the 38th annual Design Automation Conference (DAC 2001)*, pages 524–529, 2001.

[22] R. Thomasius and S. Guttowski. A SystemC Based Framework for the Evaluation of Proactive Power-Management Approaches in Distributed Energy Harvesting Systems. In *Fourth International Conference on Sensor Technologies and Applications (SENSORCOMM 2010)*, pages 348–351, 2010.

[23] M. Vasilevski, H. Aboushady, F. Pecheux, and L. de Lamarre. Modeling Wireless Sensor Network Nodes Using SystemC-AMS. In *Internatonal Conference on Microelectronics (ICM 2007)*, pages 53–56, 2007.

[24] M. Vasilevski, N. Beilleau, H. Aboushady, and F. Pecheux. Efficient and Refined Modeling of Wireless Sensor Network Nodes Using SystemC-AMS. In *Research in Microelectronics and Electronics (PRIME 2008)*, pages 81–84, 2008.

[25] M. Vasilevski, F. Pecheux, N. Beilleau, H. Aboushady, and K. Einwich. Modeling and Refining Heterogeneous Systems with SystemC-AMS: Application to WSN. In *Design, Automation and Test in Europe (DATE 2008)*, pages 134–139, 2008.

[26] N. Xu, S. Rangwala, K. K. Chintalapudi, D. Ganesan, A. Broad, R. Govindan, and D. Estrin. A Wireless Sensor Network for Structural Monitoring. In *Proceedings of the 2nd international conference on Embedded networked sensor systems (SenSys 2004)*, pages 13–24, 2004.

2011 18th International Conference on Telecommunications

Designing of Efficient Energy Harvesting Systems for Autonomous WSNs Using a Tier Model

Leander B. Hörmann, Philipp M. Glatz, Christian Steger and Reinhold Weiss
Institute for Technical Informatics
Graz University of Technology
Graz, Austria
Email: {Leander.Hoermann@, Philipp.Glatz@, Steger@, RWeiss@}TUGraz.at

***Abstract*—Energy harvesting systems (EHSs) are the key to perpetual operation of electronic devices in application areas with bad infrastructure or mobility. Wireless sensor networks (WSNs) are often used in such areas. Normal WSN nodes are powered by batteries. Therefore, the lifetime is limited and the batteries have to be replaced manually after a certain period of time. This problem can be solved by EHSs. They exploit energy sources of the environment and store the harvested energy in energy buffers. The EHS supplies the electronic device and ensures a continuous operation. WSNs can benefit from these developments, because the lifetime can be enhanced dramatically. However, the EHS have to be adapted to the requirements of the application area and of the supplied device. This enhances the overall efficiency of EHS. To be able to do that, the fundamental mode of operation of an EHS has to be well-understood.**

We introduce a novel tier model for EHSs. It structures the EHS into tiers with special functions. This enhances the design process of an EHS, because tiers can be adapted to each other and the overall efficiency of can be increased.

The tier model is applied to RiverMote, a WSN node for in-river water level monitoring. Each node is supplied by solar cells and the energy is stored in double layer capacitors (DLCs). The hardware of the EHS of RiverMote is divided into the tiers of the model. These tiers are adapted to each other carefully. Although no maximum power point tracker has been implemented, it has been shown that the available power of the solar cell is greater than 80 % of the maximum power point if the energy level of the DLCs is between 42 % and 100 %. This result was only possible by a careful design and an adaption of the tiers.

I. Introduction

Wireless sensor networks (WSNs) provide sensing and communication capabilities in application areas with mobility or bad infrastructure. The use of WSNs supersedes the wiring of each node. Due to the fact that there is no wired infrastructure, each node of the WSN needs its own energy source. However, the requirements on a WSN depend strongly on its application area, which are manifold. Precision agriculture [1], [2], wildlife monitoring [3], [4], human health-care [5] or structural health monitoring [6] are only few of different application areas of WSNs. Their diversity causes an adaptation of a WSN to a specific application area. Today, there is no all-in-one WSN that can be used in every application area. Therefore, the WSN is specialized and it is able to perform its task in an optimal way.

As mentioned before, each node needs its own energy source. Very often, batteries are used. They are cheap and provide a reliable supply. However, the lifetime of each node supplied by batteries is limited. Typically, the current consumption of the node determines the lifetime. If the average current approximates to the leakage current of the battery, the lifetime depends also on the leakage current [7]. Therefore, the lifetime cannot be extended in any order by reducing the average current consumption. This results in the battery replacement problem as mentioned in [8]. The batteries have to be replaced manually after a certain period of time. However, the replacement is sometimes very difficult, because the WSN nodes are deployed at hard-to-access locations. Hence, the replacement should be avoided if possible.

The solution is the use of energy harvesting systems (EHS) [9]–[11]. These systems exploit environmental energy sources to enhance the lifetime of WSN nodes. They convert the available energy into electrical energy and store it in energy storage components (ESCs). The storage is needed, because *energy from the environment is generally unpredictable, discontinuous, and unstable* [12]. Rechargeable batteries or double layer capacitors (DLCs) can be used as ESCs. There are also approaches to combine different types of ESCs to benefit from the advantages of each type [13]. The energy is converted by energy harvesting devices (EHDs), e.g. solar cells or thermoelectric generators. The selection of the type of the EHD and the ESC depends on the application area. Each application has special requirements on the WSN node and hence on the EHS. Every EHD and ESC has special characteristics which are called implicit properties here. It is useful to consider these properties during the design of the EHS to enhance the overall efficiency of the system.

It can be seen that an EHS is a complex hardware unit and efficient mode of operations depends on the adaptation of the EHD, the ESC and other components to each other. Therefore, we have developed a tier model which breaks down the EHS into five different tiers and simplifies understanding of it. Each tier consists of one or more components, which are necessary to fulfill the functionality of the tier. It is also possible that a tier can be omitted completely if the functionality is not required. Therefore, it is necessary to adapt the tiers and the components of the tiers to each other in order to optimize the efficiency of the EHS.

This paper is organized as follows: Section II discusses related work. The tier model is introduced in Section III and

978-1-4577-0024-8/11/$26.00 © 2011 IEEE

TABLE I
POWER DENSITIES OF DIFFERENT ENERGY HARVESTING TECHNOLOGIES [10], [15].

Harvesting technology	Power density
Solar cells (outdoor at noon)	10-15 mW/cm^2
Solar cells (indoor) [16]	$\approx 10\,\mu W/cm^2$
Piezoelectric (shoe inserts)	$\approx 300\,\mu W/cm^3$
Vibration (small microwave oven)	$\approx 100\,\mu W/cm^3$
Thermoelectric ($10°C$ gradient)	$\approx 40\,\mu W/cm^3$
Acoustic noise (100dB)	$\approx 1\,\mu W/cm^3$

describes each tier in detail. Section IV applies the tier model to existing EHSs and presents the results. Finally, Section V concludes the paper and shows directions for future work.

II. Background and Related Work

For understanding the special characteristics of EHSs, their constraints and their opportunities, we briefly survey properties of well-known EHSs and design methodologies.

A theoretical energy harvesting model is introduced in [14]. The model deals with the description of environmental energy sources and the energy sink (consumer).

However, the model does not describe characteristics of the energy storage unit or consider the power loss of the EHS. Furthermore, it is a model of the input and output power and not of the EHS itself. The tier model of this paper describes the hardware of the EHS.

The amount of harvestable energy depends on the type of the environmental energy source, on the power of the energy source and on the efficiency of the EHD. Table I shows a list of commonly used energy harvesting technologies and their power densities [10], [15]. It can be seen that the power density of the solar cell depends dramatically on the location. Therefore, if the EHS is deployed outdoor, solar cells are suited well. However, indoor EHSs have lower input power and the supplied node should have a low power consumption, too.

The properties of different types of ESCs are shown in Table II. Nickel-cadmium (NiCd), nickel-metal hydride (NiMH), lead-acid, lithium-ion (LiIon) and lithium-polymer (LiPo) batteries and DLCs are given. The properties vary between the single types. Especially the nominal cell voltage and the overcharge tolerance have a high influence on the design of the EHS. Therefore, the selection of the ESC type is one of the first steps during EHS design. A great difference is also between the DLC and the other batteries. The energy density of the batteries is much higher compared to the DLC. However, the cycle life of the DLC is much higher compared to the batteries. This difference must be considered at the selection of the ESC, too.

There are a lot of existing WSN nodes and EHSs for these nodes. The Heliomote [10] module is an EHS using solar cells and NiMH batteries for Mica2 nodes. The Prometheus [13] module is an EHS using solar cells, DLCs and LiPo batteries for Telos nodes. It uses two different types of ESCs to store the harvested energy. In [19] an EHS is described, which supports different types of EHDs and stores the energy in a DLC. It can measure the input and output power.

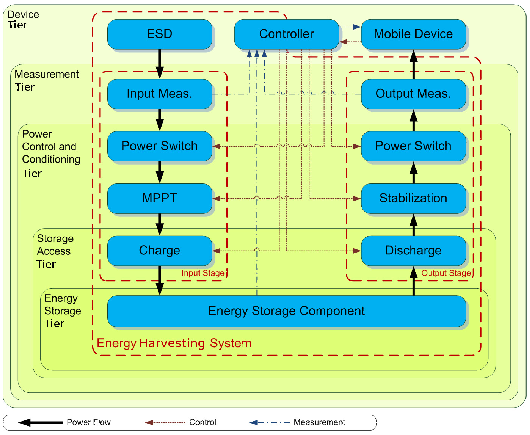


Fig. 1. Tier model of a simple EHS.

Prometheus and Heliomote do not use so called maximum power point trackers (MPPTs). A MPPT can be used to maximize the output power of an EHD. Such elements or multiple ESCs increase the complexity of the EHS. Therefore, it is suitable to use a model which can help to design an efficient EHS.

III. Energy Harvesting System Model

This section introduces a tier model to describe EHSs. The model consists of five different tiers. Each tier provides a specific functionality and is interacting with the inner and outer tier. Interacting means a flow of energy, here. Therefore, a tier encapsulates the functionality and guarantees a safe energy flow through the whole model. It simplifies the real EHS by combining different hardware blocks to single elements of a tier. Typically two different elements are needed to form a complete tier. One element is used as an input element and the other one is used as output element. Therefore, energy can flow from an outer tier through the input element of the current tier to an inner tier, and vice versa. An element of a tier can measure, transform, store, and consume energy.

Fig. 1 shows the model of a simple EHS. The combination of the input measurement, the power switch, the MPPT, and the charge element forms the input stage. It is shown on the left side of the figure. The functionality of the stage is the measurement, control, and optimal storage of the energy provided by the EHD. The combination of the discharge, the stabilization, the power switch, and the output measurement element forms the output stage. It is shown on the right side of the figure. The functionality of the stage is the stabilization, control, and measurement of the energy needed by the WSN node.

TABLE II
PROPERTIES OF DIFFERENT TYPES OF ESCS [17]. THE VALUES OF THE DLC COME FROM MAXWELL BOOSTCAP DLC [18].

Property	NiCd	NiMH	Lead Acid	LiIon	LiPo	DLC
Energy density [Wh/kg]	45-80	60-120	30-50	110-160	110-130	4.5
Cycle life (to 80% of initial capacity)	1500	300-500	200-300	500-1000	300-500	500000
Typical fast charge time	1h	2-4h	8-16h	2-4h	2-4h	$\approx$ 1 min
Overcharge tolerance	moderate	low	high	very low	low	low
Nominal cell voltage [V]	1.25	1.25	2	3.6	3.6	2.5
Operating Temp. [°C]	-40 to 40	-20 to 60	-20 to 60	-20 to 60	0 to 60	-40 to 65

The dotted and the dash-dotted lines indicate the information flow in the model. The dash-dotted lines represent the measurement lines that are used to measure the current state of the EHS. The dashed lines represent the control lines. They are used to vary their behavior in order to enhance the performance.

If there are no control lines from the controller to any elements, the behavior of the elements cannot be adjusted and it is a fixed EHS. This reduces the complexity of the EHS. However, the system must be well designed, because the behavior cannot be changed afterwards. If there are no measurement lines from the elements to the controller, the state of the EHS cannot be determined. Therefore, it is a blind EHS. The disadvantage is that the end device does not know how much energy is stored and it cannot estimate directly the remaining supply time.

A. Tiers of the Model

The model consists of five nested tiers. The following sections describe each tier and typical elements of it.

1) Device Tier: This tier contains the EHD as well as the supplied WSN node. The EHD provides the energy to the whole system. The task of the tier is the generation and consumption of the energy and a control of the EHS.

The controller is placed on the boarder of the EHS. It can be a part of the EHS or it can be a part of the WSN node. For example the controller can be implemented as a separate component of the EHS or it can be implemented as a software module of the WSN node. There are many possibilities and it should be adapted to the application to be as efficient as possible.

2) Measurement Tier: The measurement task is important to predict the available energy. In [20], different forecasting methods are compared and evaluated. This tier is placed directly after the device tier, because so the energy losses of the EHS can be measured and evaluated during operation. Typically, the measurement element determines the energy by measuring the power over a certain period of time. The power can be calculated with the input voltage and the input current. The current can be measured with a small shunt resistor and this resistor causes a loss of energy. Therefore, it is realistic to define the measurement element as a lossy element. This means that the output power (e.g. the power to the power switch) is smaller than the input power (e.g. the power from the EHD).

The elements of this tier have measurement outputs that are used to detect the current state of the EHS. This information can be used by the controller to predict the available energy of the EHS.

If no elements of this tier are implemented by the EHS, the input or output energy transfer cannot be measured. However, the controller is able to calculate a rough estimation by measuring the energy level of the ESC. With this value and an estimation of the output power, the input power can be calculated. Therefore, a model of the ESC must be known. The output power can be defined as constant or can be estimated at runtime [21], [22]. It can be seen clearly that it is a rough estimation only. If a more detailed determination of the current state of the EHS is needed, measurement elements of this tier are necessary.

3) Power Control and Conditioning Tier: This tier controls the power flow of the energy harvesting system and enables an optimal power flow between the EHD and the ESC and between the ESC and the WSN node, respectively.

Power switches are used to control the power flow. These elements can be used to actively control the energy level of the ESC. They can also be used to deactivate energy harvesting or to switch off the WSN node. Typically, the controller or the mote controls these elements to optimize the energy harvesting process.

In more complex EHSs the power control tier can also be used for the power flow between input stage and the output stage. This bypasses the power and the ESC is not stressed. Therefore, the lifetime of the ESC can be enhanced.

The conditioning element of the input stage is a MPPT. It ensures an operation of the EHD at the MPP. The behavior of the EHD depends on the type of it (solar cell, TEG, ...). Therefore, it is possible that the MPPT is designed for a single type of EHDs.

The conditioning element of the output stage is the stabilization element. It is necessary, because the WSN node needs a stabilized voltage to operate well. Typically, the voltage level of the ESC depends on the energy stored. Therefore, the stabilization is used to transform the irregular voltage of the ESC to a constant output voltage.

There are two possibilities to control the MPPT. First, it can be controlled by itself. This means the MPPT is able to

detect the MPP and to adjust the power flow. Second, it can be controlled by the controller. Hence, the complexity of the MPPT is reduced. However, the controller has to continuously track the input power and adjust the MPPT.

The stabilization element can also be adjusted by the controller if it is supported by the element. Therefore, the output voltage can be varied. This leads to dynamic voltage scaling (DVS) that is used to reduce the power consumption of the supplied device [23]. Component-aware DVS [24] minimizes the power consumption of the mote by setting the supply voltage to the minimal possible supply voltage defined by the currently active hardware components.

4) Storage Access Tier: This tier is used to access the ESC. It controls the charging and discharging of the ESC. This is very important, because some ESC types are very sensitive to overvoltage, undervoltage, or too high currents (e.g. LiIon batteries). Therefore, this tier is responsible for a proper and safe access. The charge element uses an overcharge protection to prevent a destruction of the ESC. This can be done in three different ways. First, the charging process can be interrupted by disconnecting the input lines. Second, only the needed power is conducted to the ESC and the excess power is converted into heat. Third, the whole input power can be drained off by shorting it. The solution for a specific EHS depends on the EHD and the ESC, because for example not every EHD can bear a short. A good solution is the second approach, because the ESC is kept at a nearly constant energy level. Therefore, the lifetime of the ESC can be enhanced.

The discharge element is responsible for a controlled discharging process. It prevents an undervoltage of the ESC and too high discharge currents. However, some WSN nodes are able to detect a low power level of the ESC and reduce their duty cycle to save power. This dynamic energy management is typically implemented in software. However, a undervoltage protection implemented in hardware is necessary if the ESC is sensitive to undervoltage, because of unexpected software faults or long interruption time of the EHD. An example of an unexpected long interruption time is the coverage of the solar cell by leaves or waste. If the interruption time is too long, the software cannot balance this drop of available energy.

The controller can adjust the charging and discharging elements to adapt it to environmental changes. For example it can adjust the maximum charging current dependent on the temperature.

5) Energy Storage Tier: The innermost tier of an EHS is the energy storage tier. It contains the ESC itself. An access (charging or discharging) is only allowed through the elements of the outer tier. A direct and uncontrolled access may damage the ESC, because of overvoltage or undervoltage.

If the energy level of the ESC cannot be determined, the EHS is blind in terms of its energy level. The measurement of the input and output level cannot compensate this lack of information. The input and output power can be integrated, but small errors will accumulate and the calculated energy level will drift away from the real one. Therefore, the detection of the energy level of the ESC is essential for the correct determination of the energy level of the EHS.

B. DLC, LiIon and LiPo Battery Requirements.

Every type of ESC has specific properties and leads to other requirements on the energy flow. This section describes the requirements of a LiIon or LiPo battery and a DLC. However, there are many other types that are not discussed in detail, because it would go beyond the scope of this paper. Only the most important two types are described.

A DLC is only sensitive to overvoltage. Therefore, if it is used as ESC, the charging element of the storage access tier has to disrupt the charging process. One of the three types mentioned in Section III-A4 can be used. Undervoltage and typical charge and discharge currents (up to 1 A) are unproblematic. Therefore, the discharging element can be omitted.

As already mentioned, LiIon and LiPo batteries are very sensitive to undervoltage, overvoltage and too high currents. Therefore, both types lead to detailed requirements on the whole system if they are used as ESC. A consistent implementation of these requirements enhances significantly the lifetime. The charge element of the storage access tier is very important to prevent both from overvoltage. One of the three types mentioned in Section III-A4 can be used. It is also important to limit the maximum charge current to the maximum allowed current. The discharge element has to avoid an undervoltage of both battery types. Therefore, it has to switch off the WSN node if the voltage is too low.

C. Selection of the Needed Tiers and Elements

The measurement tier is only needed if it is defined by the application requirements. For EHS purpose it is not really necessary. The measurement of the input and output power can be used to forecast the available energy in the next hours or days. However, to supply the WSN node continuously with energy, the EHS cannot rely on forecasts. All harvestable energy should be stored. Therefore, the input power switch is not always needed. Also the output power switch is not always necessary. Therefore, the measurement tier and the power switches can be omitted in many cases.

It is possible to exploit the special properties of elements to reduce the need of some elements. The simplest possible EHS consists of only three elements: the EHD, a charge element, and the ESC. In this case, the EHD must be selected very carefully. The only task of the charge element is the prevention of discharging the ESC by the ESD during intervals with no harvestable power, for example in the case of solar cells during the night. The ESC must be robust against undervoltage and the peak input and output current. However, the disadvantage of such a system is the lower efficiency compared to more complex systems. The tradeoff between high efficiency and low complexity has to be taken at the design phase of the EHS.

D. Multiple Energy Storage Units

As mentioned before, the Prometheus module uses a DLC and a LiPo battery. The DLC is used as primary ESC and

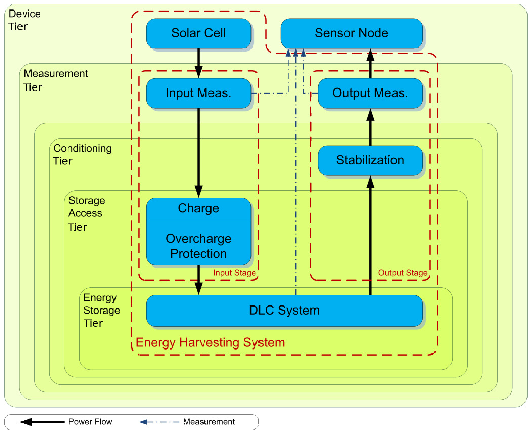


Fig. 2. Tier model of a RiverMote's EHS.

buffers the energy from the solar cell. The LiPo is used as secondary ESC. If the energy level of the primary buffer is high enough, the secondary buffer is charged. If it is too low, the end device is supplied by the secondary buffer. Usually, the WSN node is supplied by the primary buffer. The advantage of this structure is that the LiPo is only used if the energy level of the primary buffer is too low. Therefore, the number of charge discharge cycles of the LiPo is kept low and the lifetime can be enhanced. The power flow is controlled by a software module of the mote.

It is also possible to integrate non rechargeable batteries as secondary ESC. This can be done to guarantee a minimum lifetime if no periodical environmental power is available. A primary ESC is needed to store the energy for short periods without harvestable energy. Only if these periods get longer and longer and the primary ESC runs out of energy, the secondary ESC is used. However, it cannot be loaded anymore and it must be replaced manually. Therefore, it is only a less-than-ideal solution. It is better to exhaust the available environmental energy sources and to do without non rechargeable batteries.

IV. Practical Example and Evaluation

The second example shows the design and implementation of RiverMote [25], a wireless sensor node designed to perform in-river measurements of the current water level. An overview of the EHS system is shown in Fig. 2. The EHS was designed to use two DLCs connected in series as ESC and a solar cell as EHD. The solar cell transforms the solar radiation in electrical energy. The power of the solar cell is measured and conducted to an overcharge protection and a charging circuit. Then, the DLC system is directly charged. The output stage consists only of a stabilizing element and a measurement element to measure the output power.

The EHS is reduced to a minimum of the needed elements. One requirement was the measurement of the input and output power. Therefore, the measurement tier is implemented. The EHS harvests and stores energy until the maximum allowed voltage of the DLCs is reached. There are no control lines from the sensor node to any element. This means that the EHS cannot be controlled by the node and it is a fixed EHS as mentioned before.

A MPPT is not implemented in hardware. But a careful selection of the solar cell and the DLCs ensures a maximum power with minimum effort. The open circuit voltage of the solar cell is slightly above the nominal voltage of the DLCs. Measurements have shown that if the voltage of the DLC system is between 63 % and 100 %, the harvestable power of the solar cell is greater than 80 % of the MPP. The relation between the voltage and the stored energy of a DLC is:

$$W = \frac{1}{2} \cdot C \cdot U^2 \tag{1}$$

Due to the fact that the voltage is squared, the energy efficiency is even better. The measurements have shown that if the energy level of the DLC system is between 42 % and 100 %, the harvestable power of the solar cell is greater than 80 % of the MPP. Fig. 3 shows the trace of the solar efficiency vs. the DLC system voltage and the solar efficiency vs. the energy level of the DLC system.

Also the discharge element of the storage access tier is omitted, because the DLCs are not sensitive to an undervoltage. If the input power is smaller than the output power, the voltage drops until the stabilizing element stops working.

It can be seen that the EHS of RiverMote is modeled with the introduced tier model. Exhausting the implicit properties of the elements, the complexity has been reduced and the efficiency has been enhanced. This was only possible by the knowledge of the interaction of the tiers and the adaption of each element to the others.

V. Conclusion

We presented a model for EHS, based on nested tiers. The key elements of an EHS are identified and assigned to the tiers. Each tier fulfills a special functionality of the EHS. Furthermore, typical requirements on an EHS are discussed and the implementation of them using the tier model. Our model allows a structured and clear approach of the design of the EHS. It has been shown that the complexity of the EHS can be reduced and the efficiency enhanced if the elements of different tiers are adapted to each other. The EHS of the RiverMote has an efficiency of greater than 80 % if the energy level of the ESC is greater than 42 % referring to the MPP. The model can also be used to develop EHS simulators and emulators. Each tier or even each element can be implemented by a single software module. This enhances the exchangeability of the modules and different types of elements can be evaluated easily. Altogether the model improves the understanding of EHSs and their functionality.

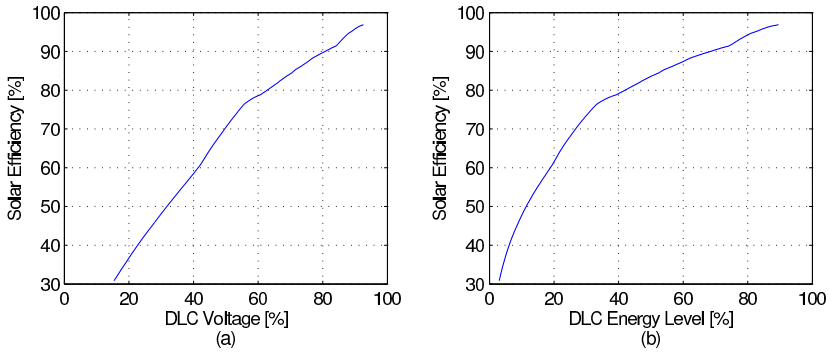


Fig. 3. Dependency of the solar efficiency on (a) the voltage and (b) the energy level of the DLC system.

REFERENCES

[1] K. Langendoen, A. Baggio, and O. Visser, "Murphy loves potatoes: experiences from a pilot sensor network deployment in precision agriculture," *Parallel and Distributed Processing Symposium, 2006. IPDPS 2006. 20th International*, p. 8 pp., apr. 2006.

[2] N. Watthanawisuth, A. Tuantranont, and T. Kerdcharoen, "Microclimate real-time monitoring based on zigbee sensor network," *Sensors, 2009 IEEE*, pp. 1814 –1818, oct. 2009.

[3] P. Juang, H. Oki, Y. Wang, M. Martonosi, L. S. Peh, and D. Rubenstein, "Energy-efficient computing for wildlife tracking: design tradeoffs and early experiences with zebranet," in *ASPLOS-X: Proceedings of the 10th international conference on Architectural support for programming languages and operating systems*. New York, NY, USA: ACM, 2002, pp. 96–107.

[4] A. Lindgren, C. Mascolo, M. Lonergan, and B. McConnell, "Seal-2-seal: A delay-tolerant protocol for contact logging in wildlife monitoring sensor networks," *Mobile Ad Hoc and Sensor Systems, 2008. MASS 2008. 5th IEEE International Conference on*, pp. 321 –327, sep. 2008.

[5] K. Lorincz, B.-r. Chen, G. W. Challen, A. R. Chowdhury, S. Patel, P. Bonato, and M. Welsh, "Mercury: a wearable sensor network platform for high-fidelity motion analysis," in *SenSys '09: Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems*. New York, NY, USA: ACM, 2009, pp. 183–196.

[6] N. Xu, S. Rangwala, K. K. Chintalapudi, D. Ganesan, A. Broad, R. Govindan, and D. Estrin, "A wireless sensor network for structural monitoring," in *SenSys '04: Proceedings of the 2nd international conference on Embedded networked sensor systems*. New York, NY, USA: ACM, 2004, pp. 13–24.

[7] C. Enz, N. Scolari, and U. Yodprasit, "Ultra low-power radio design for wireless sensor networks," *Radio-Frequency Integration Technology: Integrated Circuits for Wideband Communication and Wireless Sensor Networks, 2005. Proceedings. 2005 IEEE International Workshop on*, pp. 1 – 17, nov. 2005.

[8] M. Minami, T. Morito, H. Morikawa, and T. Aoyama, "Solar biscuit: A battery-less wireless sensor network system for environmental monitoring applications," *The 2nd International Workshop on Networked Sensing Systems*, 2005.

[9] R. Min, M. Bhardwaj, S.-H. Cho, N. Ickes, E. Shih, A. Sinha, A. Wang, and A. Chandrakasan, "Energy-centric enabling tecumologies for wireless sensor networks," *Wireless Communications, IEEE*, vol. 9, no. 4, pp. 28 – 39, aug. 2002.

[10] V. Raghunathan, A. Kansal, J. Hsu, J. Friedman, and M. Srivastava, "Design considerations for solar energy harvesting wireless embedded systems," in *Proceedings of the 4th international symposium on Information processing in sensor networks*. IEEE Press, 2005, p. 64.

[11] M. Rahimi, H. Shah, G. Sukhatme, J. Heideman, and D. Estrin, "Studying the feasibility of energy harvesting in a mobile sensor network," *Robotics and Automation, 2003. Proceedings. ICRA '03. IEEE International Conference on*, vol. 1, pp. 19 – 24 vol.1, sep. 2003.

[12] A. Janek, C. Trummer, C. Steger, R. Weiss, J. Preishuber-Pfluegl, and M. Pistauer, "Simulation based verification of energy storage architectures for higher class tags supported by energy harvesting devices," *Microprocessors and Microsystems*, vol. 32, no. 5-6, pp. 330–339, 2008, dependability and Testing of Modern Digital Systems.

[13] X. Jiang, J. Polastre, and D. Culler, "Perpetual environmentally powered sensor networks," *Information Processing in Sensor Networks, 2005. IPSN 2005. Fourth International Symposium on*, pp. 463 – 468, apr. 2005.

[14] A. Kansal, D. Potter, and M. B. Srivastava, "Performance aware tasking for environmentally powered sensor networks," in *SIGMETRICS '04/Performance '04: Proceedings of the joint international conference on Measurement and modeling of computer systems*. New York, NY, USA: ACM, 2004, pp. 223–234.

[15] P. He, Q. Cui, and X. Guo, "Efficient solar power scavenging and utilization in mobile electronics system," *Green Circuits and Systems (ICGCS), 2010 International Conference on*, pp. 641 –645, jun. 2010.

[16] S. Roundy, D. Steingart, L. Frechette, P. Wright, and J. Rabaey, "Power sources for wireless sensor networks," in *Wireless Sensor Networks*, ser. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 2004, vol. 2920, pp. 1–17.

[17] C. Kompis and S. Aliwell, "Energy harvesting technologies to enable remote and wireless sensing," http://host.quid5.net/ koumpis/pubs/pdf/energyharvesting08.pdf, June 2008.

[18] MaxwellTechnologies, "Datasheet bcap0310 p250," http://www.maxwell.com, Document number: 1009473.5, November 2009.

[19] P. Glatz, P. Meyer, A. Janek, T. Trathnigg, C. Steger, and R. Weiss, "A measurement platform for energy harvesting and software characterization in wsns," *Wireless Days, 2008. 1st IFIP*, pp. 1 –5, nov. 2008.

[20] C. Bergonzini, D. Brunelli, and L. Benini, "Algorithms for harvested energy prediction in batteryless wireless sensor networks," in *3rd International Workshop on Advances in sensors and Interfaces*, June 2009, pp. 144–149.

[21] A. Dunkels, F. Osterlind, N. Tsiftes, and Z. He, "Software-based on-line energy estimation for sensor nodes," in *EmNets '07: Proceedings of the 4th workshop on Embedded networked sensors*. New York, NY, USA: ACM, 2007, pp. 28–32.

[22] P. M. Glatz, C. Steger, and R. Weiss, "Tospie2: tiny operating system plug-in for energy estimation," in *IPSN '10: Proceedings of the 9th ACM/IEEE International Conference on Information Processing in Sensor Networks*. New York, NY, USA: ACM, 2010, pp. 410–411.

[23] A. Sinha and A. Chandrakasan, "Dynamic power management in wireless sensor networks," *Design Test of Computers, IEEE*, vol. 18, no. 2, pp. 62 –74, mar. 2001.

[24] L. B. Hörmann, P. M. Glatz, C. Steger, and R. Weiss, "Energy efficient supply of wsn nodes using component-aware dynamic voltage scaling," in *Wireless Conference (EW), 2011 European*, April 2011, accepted for publication.

[25] L. B. Hörmann, "Design and implementation of a wireless sensor platform for river monitoring based on energy harvesting," Master's thesis, Graz University of Technology, Institute for Technical Informatics, 2010.

Work in Progress session at PerCom 2012, Lugano (19-23 March 2012)

Towards an On-Site Characterization of Energy Harvesting Devices for Wireless Sensor Networks

Leander B. Hörmann*, Philipp M. Glatz†, Karima B. Hein*,
Michael Steinberger*, Christian Steger* and Reinhold Weiss*
*Institute for Technical Informatics, Graz University of Technology, Inffeldgasse 16, 8010 Graz, Austria
Email: {Leander.Hoermann@, Hein@, Michael.Steinberger@student., Steger@, RWeiss@}TUGraz.at
†Embedded Software Group, Delft University of Technology, Mekelweg 4, 2628 CD Delft, The Netherlands
Email: P.M.Glatz@TUDelft.nl

***Abstract*—Wireless sensor networks (WSNs) suffer from the lack of wired infrastructure. Each node needs its own power supply, e.g. batteries or energy harvesting systems (EHSs). Typically, EHSs can extend the lifetime of a sensor node or even enable perpetual operation. Due to the high variation of harvestable energy of the environment, the design of the EHSs has to be done very carefully. The design process can be enhanced by using simulation of the WSN including energy harvesting. However, a realistic simulation needs accurate data of the harvestable energy of the environment. This paper presents the concept of an on-site characterization instrument for different types of energy harvesting devices. These instruments can be connected like a WSN.**

***Index Terms*—Wireless sensor network, energy harvesting, energy harvesting device, on-site characterization.**

I. Introduction and Related Work

Wireless sensor networks (WSNs) are often used in application areas without wired infrastructure. Examples are: precision agriculture [1], [2], wildlife monitoring [3], human health care [4], and structural health monitoring [5]. Typically, each sensor node is powered by batteries or energy harvesting systems (EHSs). Batteries have to be replaced or recharged after a certain period of time depending on the energy consumption of the node and the capacity of the batteries. This fact is well known as the battery replacement problem [6]. EHSs convert the energy of the environment into electrical energy to supply the sensor node. This can enable perpetual operation. Table I shows typical energy harvesting technologies and their power densities [7], [8]. It can be seen that the energy densities are very low except for solar cells used outside. A part of the harvested energy has to be stored in order to supply the node during periods with insufficient harvestable energy. Rechargeable batteries or double layer capacitors (DLCs) are commonly used to store the energy. There are a lot of existing energy harvesting enhanced WSNs. Examples are the EHS module of the Heliomote project [7] using solar cells and Nickel-Metal hydride (NiMH) rechargeable batteries and the EHS module of the Prometheus project [9] using solar cells, DLCs and Lithium-Polymer (LiPo) rechargeable batteries. The latter benefits from the advantages of both energy storage components: the high number of supported charge-discharge cycles of the DLC and the high capacity of the LiPo battery.

TABLE I
Power densities of different energy harvesting technologies [7], [8].

Harvesting technology	Power density
Solar cells (outdoor at noon)	10-15 mW/cm^2
Solar cells (indoor) [11]	$\approx 10\,\mu W/cm^2$
Piezoelectric (shoe inserts)	$\approx 300\,\mu W/cm^3$
Vibration (small microwave oven)	$\approx 100\,\mu W/cm^3$
Thermoelectric ($10°C$ gradient)	$\approx 40\,\mu W/cm^3$
Acoustic noise (100dB)	$\approx 1\,\mu W/cm^3$

However, these modules cannot be used to characterize energy harvesting devices.

Some EHSs have an integrated maximum power point tracker (MPPT). The MPPT operates the energy harvesting device always at the maximum power point, therefore ensuring efficient harvesting. A possible solution of an energy harvesting enhanced WSN using an MPPT is shown in [10]. They used the open-circuit voltage of a second and smaller solar cell to determine the maximum power point of the first one. However, an MPPT increases the complexity of the hardware. On the other side, EHSs without an MPPT have lower efficiency. Their design must be done more careful to ensure good efficiency. This means there is a tradeoff between the efficiency of the EHS and its complexity. Therefore, it depends on the application whether an MPPT should be used or not.

Due to the fact that a WSN is often placed at hard-to-access locations, a replacement or reconfiguration is cumbersome and expensive. Maintenance should be avoided after installation. Therefore, simulations are used to determine the behavior of a WSN before the installation or even before the designing and implementing of a sensor node. An accurate simulation can enhance the design process of a WSN including software and hardware and reduces the probability of faults during operation.

We have introduced a simulation environment written in SystemC-AMS in [12] which is used to simulate whole WSNs including energy harvesting. The EHS is modeled in a detailed way using the model we have introduced in [13]. It is composed of single components, e.g. resistors, diodes, voltage converters. This guarantees realistic simulation results.

978-1-4673-0907-3/12/$31.00 ©2012 IEEE

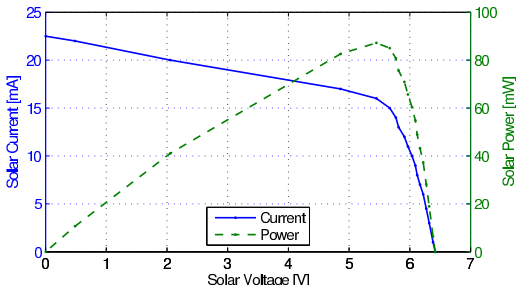


Fig. 1. Characteristic of the ASI3Oo05/162/192FAmod solar sell from Schott Solar.

The only thing that is missing is a detailed model of the energy harvesting device. Therefore, this work presents the concept of an on-site characterization instrument for different energy harvesting devices. In the first step we will consider solar cells and thermogenerators.

Also, available energy that can be harvested varies with location and time. For example solar cells are sometimes shaded by a tree or something else. In [7] the characteristic of a solar cell at two different points in time is shown. Therefore, it is necessary to be able to determine the characteristic of the energy harvesting device at different locations and over time. It is obvious that a WSN is used for data gathering and configuration. Each sensor node has the ability to determine the characteristic of the used energy harvesting device at a given time.

The rest of the paper is organized as follows: Section II shows the characteristics of a solar cell and a thermogenerator which are considered in this work. Section III introduces the concept of the characterization instrument and Section IV shows first results. Finally, Section V concludes the paper and gives directions for future work.

II. Energy Harvesting Device Evaluation

This section shows the characteristics of two different types of energy harvesting devices. The first one is a solar cell which has been already used to supply a WSN node for river monitoring [14]. The second one is a thermogenerator which is operated with very low temperature differences.

A. Solar Cell

The characteristic of the ASI3Oo05/162/192FAmod solar cell from Schott Solar is shown in Fig. 1. Its size is 162 mm by 192 mm. The characterization of the solar cell has been done with bad lighting conditions at noon on a rainy and foggy day. This is the reason for the rather low current. The short-circuit current at direct sunlight conditions is about 355 mA. The important insight of this characteristic is the non-linear shape. Therefore, it is not sufficient to measure for example only the open-circuit voltage of the solar cell. To find the maximum power point, it is necessary to measure the current at different voltages. A full characterization is also important for

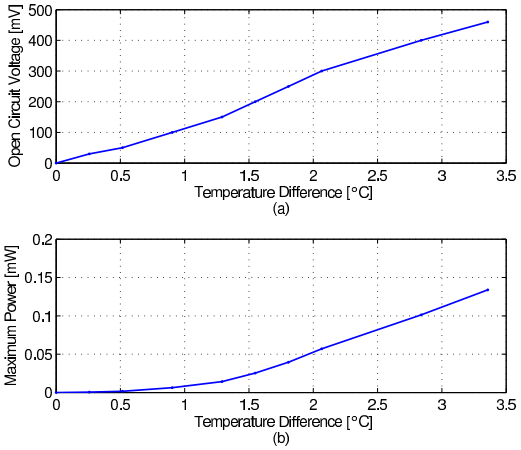


Fig. 2. Characteristic of the MPG-D751 thermogenerator from Micropelt.

simulations using an EHS without an MPPT. The reason is that the solar cell is operated at any point of the characteristic. A full characterization fosters highly accurate simulation results.

B. Thermogenerator

The characteristic of the MPG-D751 thermogenerator from Micropelt is shown in Fig. 2. Fig. 2a illustrates the open-circuit voltage versus the temperature difference of the two sides of the thermogenerator. According to the datasheet [15], the voltage-current characteristic is linear. Therefore, it is possible to calculate the maximum power point of the thermogenerator from the open-circuit voltage and the internal resistance as shown in (1). The internal resistance has been measured to be $R_i = 395\,\Omega$.

$$P_{MPP} = \frac{U^2_{opencircuit}}{4 \cdot R_i} \tag{1}$$

The calculated maximum harvestable power versus the temperature is shown in Fig. 2b. It can be seen that the power is much lower than the maximum output power of the solar cell. Also the open-circuit voltage is much lower compared to the solar cell.

III. Concept

This section introduces the concept of an instrument for on-site characterization of energy harvesting devices. The first part introduces the concept of the hardware and the second part shows a typical application scenario and the associated WSN structure.

A. Hardware Concept

As shown in Section II, the characteristics of different energy harvesting devices vary. The proposed hardware should be as simple as possible while assuring good accuracy of the measurements. We decided to use the MSP430F1611 from

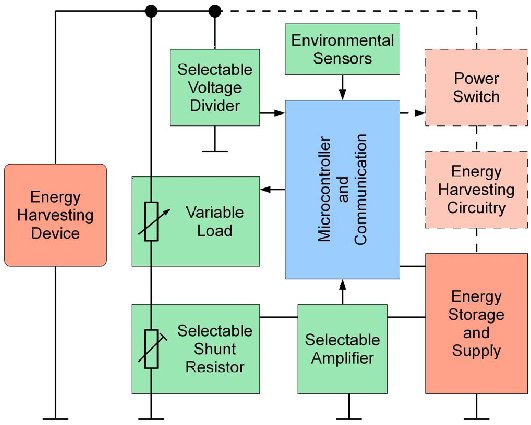


Fig. 3. Structure of the instrument for characterization of energy harvesting devices. The dashed part is only necessary if energy harvesting should be supported.

Texas Instruments for measuring and processing. This microcontroller has an integrated analog-to-digital converter which is used to perform all measurements. To be able to characterize different energy harvesting devices, different selectable shunt resistors for current measurement and different selectable voltage dividers are implemented. Therefore, it is possible to adapt the measurement circuit on the energy harvesting device which should be characterized. This has to be done prior to the installation using a specific energy harvesting device.

Fig. 3 shows the hardware structure of the characterization instrument. The measurement principle is described in the following: The terminal voltage of the energy harvesting device is measured by the microcontroller using a selectable voltage divider. This is necessary for measuring voltages (up to 10 V) higher than the supply voltage of the microcontroller (3.3 V). The ratio can be selected prior to the installation of the instrument (manual switches). A shunt resistor is used to measure the current flowing through the variable load. Due to the fact that the voltage drop at the shunt resistor should be low to ensure an accurate result, an amplifier is being used. To be as flexible as possible, the value of the shunt resistor and the gain of the amplifier can be selected prior to the installation.

The variable load is controlled by the microcontroller. It enables a characterization of the energy harvesting device. The voltage and the current can be measured at different load resistances to get a full characterization trace similar to the characteristics shown in Section II. The measured voltage and current trace are stored representing a full characterization. Such a characterization is repeated in a certain interval depending on the variation of the environmental conditions. The result is a set of different characteristics of an energy harvesting device at different environmental conditions. The environmental condition is measured by the environmental sensors. This information is linked with the corresponding characterization data.

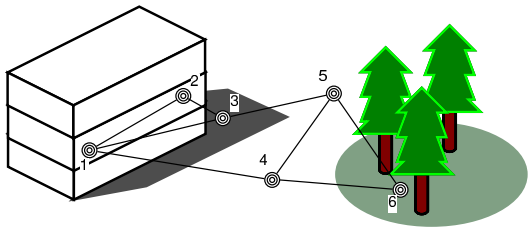


Fig. 4. Typical application scenario for solar cell characterization. Six nodes are placed at different locations with different lighting conditions.

The dashed structure represents the energy harvesting feature of the instrument. It is possible to use the energy harvesting device to recharge the energy storage component between two full characterizations. A power switch and an energy harvesting circuitry are needed for that.

B. Application Scenario

A typical application scenario is shown in Fig. 4. A certain number of characterization instruments are connected to a WSN. All the characterization and environmental data is transmitted to the base station. The advantage of such a scenario is the possibility to correlate the measured data among different placements. In the scenario, six characterization instruments are connected to a WSN. Solar cells should be characterized at different locations with different lighting conditions. Node 1 represents the base station. It is connected to a computer which stores the measurement data of all nodes. Node 2 is placed inside a building. Typically, the indoor lighting conditions are very bad. Node 3 is placed in the shadow of a building. Often, sensor nodes are placed at the north side of buildings (in the northern hemisphere) to prevent measurement influences from direct sunlight. Node 4 and 5 are placed at direct sunlight to characterize the solar cell at maximum power output. Finally, Node 6 is placed inside a forest. This scenario ensures a characterization of a solar cell at various lighting conditions.

The introduced design of the instrument ensures a high flexibility to characterize different types of energy harvesting devices at different locations in space. A long-term operation over a full year could provide a detailed characteristic of an energy harvesting device at nearly all possible weather conditions.

IV. First Results

A key element of the introduced characterization instrument is the variable load. It is used to operate the energy harvesting device at different operating points. This part has been simulated using LTspice IV. The schematic is shown in Fig. 5. The dynamic load is implemented using an operational amplifier U_2 with a feedback loop. The desired voltage can be applied to the U_{set} input. The current is varied till the terminal voltage of the energy harvesting device meets the condition shown in (2).

$$U_{V1} = U_{set} \cdot \frac{R_4 + R_5}{R_5} \tag{2}$$

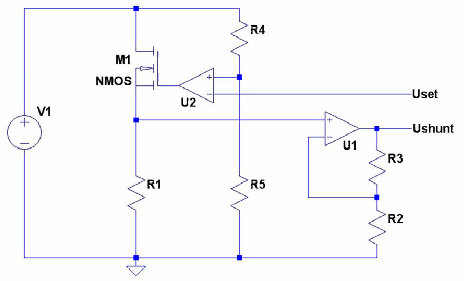


Fig. 5. Schematic of the simulated dynamic load. V_1 represents the energy harvesting device.

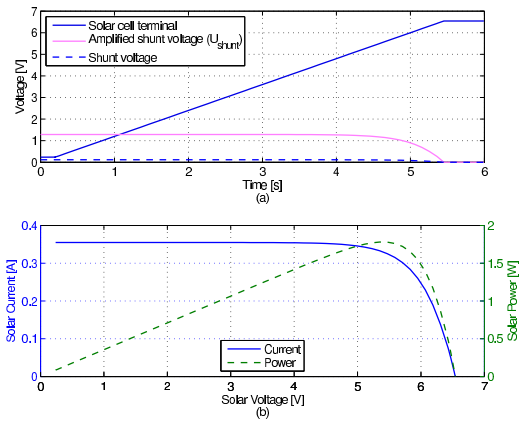


Fig. 6. Simulation results of the dynamic load and the solar cell model.

The voltage drop of the shunt resistor R_1 is amplified by the non-inverting operational amplifier U_1. The amplified shunt resistor voltage is calculated according to (3).

$$U_{shunt} = U_{R_1} \cdot \left(1 + \frac{R_3}{R_2}\right) \tag{3}$$

The simulation results are shown in Fig. 6. Fig. 6a illustrates the time trace of the solar cell terminal voltage, the direct shunt voltage, and the amplified shunt voltage. The increase of the terminal voltage causes a decrease of the solar current. Fig. 6b shows the extracted characteristic of the measured solar cell. The simulation approved the measurement concept and the ability to characterize a solar cell.

V. Conclusion and Future Work

This work presented the concept of a characterization instrument prototype for different types of energy harvesting devices. Multiple instruments can be connected to a WSN enabling a spatially distributed time-correlated characterization. The characterization data can be used to implement or improve realistic simulation environments for energy harvesting enhanced WSNs. The selectable components enable a customization of the measurement circuits. Because of this high flexibility, different energy harvesting device can be characterized.

The simulation results have shown the basic mode of operation of the dynamic load. Using this dynamic load, it is possible to characterize a solar cell.

Future work will target the completion of the instrument prototype and evaluation. It is intended for characterizing solar cells and thermogenerators. Finally, the gathered characterization information will be used in our simulation environment to enable a detailed energy harvesting simulation in order to improve the simulation results.

References

[1] K. Langendoen, A. Baggio, and O. Visser, "Murphy loves potatoes: experiences from a pilot sensor network deployment in precision agriculture," *20th International Parallel and Distributed Processing Symposium (IPDPS 2006)*, p. 8 pp., 2006.

[2] N. Watthanawisuth, A. Tuantranont, and T. Kerdcharoen, "Microclimate real-time monitoring based on zigbee sensor network," *IEEE Sensors*, pp. 1814–1818, 2009.

[3] A. Lindgren, C. Mascolo, M. Lonergan, and B. McConnell, "Seal-2-seal: A delay-tolerant protocol for contact logging in wildlife monitoring sensor networks," *5th IEEE International Conference on Mobile Ad Hoc and Sensor Systems (MASS 2008)*, pp. 321–327, 2008.

[4] K. Lorincz, B.-r. Chen, G. W. Challen, A. R. Chowdhury, S. Patel, P. Bonato, and M. Welsh, "Mercury: a wearable sensor network platform for high-fidelity motion analysis," in *Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems (SenSys 2009)*, 2009, pp. 183–196.

[5] N. Xu, S. Rangwala, K. K. Chintalapudi, D. Ganesan, A. Broad, R. Govindan, and D. Estrin, "A wireless sensor network for structural monitoring," in *Proceedings of the 2nd international conference on Embedded networked sensor systems (SenSys 2004)*, 2004, pp. 13–24.

[6] M. Minami, T. Morito, H. Morikawa, and T. Aoyama, "Solar biscuit: A battery-less wireless sensor network system for environmental monitoring applications," *The 2nd International Workshop on Networked Sensing Systems*, 2005.

[7] V. Raghunathan, A. Kansal, J. Hsu, J. Friedman, and M. Srivastava, "Design considerations for solar energy harvesting wireless embedded systems," in *Proceedings of the 4th international symposium on Information processing in sensor networks (IPSN 2005)*, 2005, pp. 457–462.

[8] P. He, Q. Cui, and X. Guo, "Efficient solar power scavenging and utilization in mobile electronics system," *International Conference on Green Circuits and Systems (ICGCS 2010)*, pp. 641–645, 2010.

[9] X. Jiang, J. Polastre, and D. Culler, "Perpetual environmentally powered sensor networks," *Fourth International Symposium on Information Processing in Sensor Networks (IPSN 2005)*, pp. 463–468, 2005.

[10] D. Brunelli, L. Benini, C. Moser, and L. Thiele, "An efficient solar energy harvester for wireless sensor nodes," *Design, Automation and Test in Europe Conference and Exhibition*, vol. 0, pp. 104–109, 2008.

[11] S. Roundy, D. Steingart, L. Frechette, P. Wright, and J. Rabaey, "Power sources for wireless sensor networks," in *Wireless Sensor Networks*, ser. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 2004, vol. 2920, pp. 1–17.

[12] L. B. Hörmann, P. M. Glatz, C. Steger, and R. Weiss, "A systemc-ams simulation environment for the evaluation of energy harvesting wireless sensor networks," in *International Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS 2011)*, 2011, pp. 247–252.

[13] ——, "Designing of efficient energy harvesting systems for autonomous wsns using a tier model," *18th IEEE International Conference on Telecommunications (ICT 2011)*, pp. 185–190, 2011.

[14] ——, "A wireless sensor node for river monitoring using msp430 and energy harvesting," in *Proceedings of the European DSP in Education and Research Conference (EDERC 2010)*. Texas Instruments, 2010, pp. 140–144.

[15] MicropeltGmbH, "Mpg-d651 mpg-d751 - thin film thermogenerators and sensing devices," http://www.micropelt.com/down/datasheet_mpg_d651_d751.pdf, November 2011.

European Wireless 2011, April 27-29, 2011, Vienna, Austria ISBN 978-3-8007-3343-9 © VDE VERLAG GMBH

Energy Efficient Supply of WSN Nodes using Component-Aware Dynamic Voltage Scaling

Leander B. Hörmann, Philipp M. Glatz, Christian Steger and Reinhold Weiss
Institute for Technical Informatics
Graz University of Technology
Graz, Austria
Email: {Leander.Hoermann@, Philipp.Glatz@, Steger@, RWeiss@}TUGraz.at

***Abstract*—Energy efficiency is very important for wireless sensor networks (WSNs), because the consumable energy is limited. Each WSN node has its own power supply. The lifetime of the WSN node depends basically on its average power consumption. Therefore, an efficient supply of the WSN node can enhance the lifetime of it. Typically, the various components of a WSN node (microcontroller, transceiver, sensors) have different supply voltage ranges. To save as much energy as possible, the supply voltage of the node should be as low as possible. Therefore, a voltage converter is needed to reduce the voltage of a battery. The minimum allowed supply voltage depends on the components that are active. The active components and consequently the minimum allowed supply voltage vary over time. Component-aware dynamic voltage scaling (CADVS) can be used to adapt the supply voltage of the node. This work presents the possible energy savings using four different voltage conversion techniques. It has been shown that CADVS can be used to save up to 31.5 % of the energy compared to a constant voltage supply using the introduced scenario while achieving the same end-user performance.**

I. Introduction

Wireless sensor networks (WSNs) are very power critical systems. They are used to measure physical quantities and transmit the information of its environment through the network in application areas with mobility or bad infrastructure. A WSN typically consists of a lot of sensor nodes. Each sensor node is an intelligent device which is responsible to forward messages through the network and to prepare the measured data for the transmission. These tasks are typically fulfilled by a microcontroller. The missing wired infrastructure causes the need of dedicated energy sources for each sensor node. Conventional batteries are cheap and provide a reliable supply for a certain period of time, because each battery has a limited capacity. Therefore, the lifetime of battery powered sensor nodes is also limited. Energy harvesting systems (EHSs) can be used to extend the lifetime of the sensor nodes [1]–[3]. They utilize environmental energy sources. Energy harvesting devices (EHDs) convert the energy of the environmental sources into electrical energy. Typical EHDs are solar cells and thermogenerators. The converted energy must be stored, because *energy from the environment is generally unpredictable, discontinuous, and unstable* [4]. The available energy is often very low compared to the energy consumption of the sensor node. These are the reasons of the strict power constraints for WSNs.

The application areas of WSNs are manifold. Precision agriculture [5], [6], wildlife monitoring [7], [8], human health care [9] and structural health monitoring [10] are only a few examples. Typically, a sensor node is adapted to its special application area. Therefore, only the necessary sensors and components are embedded into the sensor node and significant energy can be saved. The needed sensors depend strongly on the application area and are often predetermined. Each sensor has its own power requirements and its own supply voltage range. Therefore, the components (especially the microcontroller) have to be adapted to the sensors to ensure proper functionality. In many cases, the supply voltage of the sensor node is set to a fixed high value, because only one component of the node needs this high voltage. The other components do not need such a high supply voltage. Therefore, lot of energy is wasted, because of a single component.

Another case of wasting energy is the direct connection of the sensor node to the energy storage unit (ESU), e.g. battery. The voltage of almost all ESUs depends on its energy level. Typically, the voltage decreases with decreasing energy level of the ESU. Each sensor node has a specific supply voltage range wherein a proper functionality is guaranteed. Below the lower limit the sensor node cuts out. Therefore, the voltage of the fully charged ESU must be higher than this lower limit. The emerging difference is wasted energy. This difference depends on the lower limit of the supply range and the type of the ESU.

Component-aware dynamic voltage scaling (CADVS) can be used to close the gap between the different minimal supply voltages of the different sensors and components. The rest of the paper is organized as follows: Section II discusses related work. Section III describes the principals of energy efficient supply of WSN nodes and CADVS applied to the nodes. Section IV shows the possible gains using DVS. Finally, Section V concludes the paper and shows directions for future work.

II. Related Work

As mentioned before, WSN nodes are often supplied by conventional batteries. The problem is the limited lifetime. It can be extended by reducing the power consumption of the sensor node. However, this extension is limited by the leakage currents of each battery. Figure 1 shows the expected

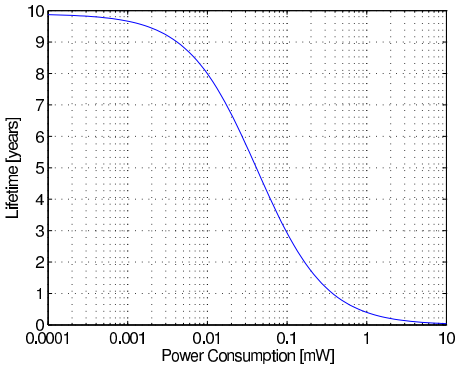


Fig. 1. Lifetime of a battery-powered system versus average power consumption using a simple battery model of a single AA cell (NiMH) with 2.6 Ah and a leakage current of 30 μA [11].

lifetime of a single AA cell (NiMH) with a capacity of 2.6 Ah depending on the current consumption of the sensor node. A typical leakage current of such a cell is 30 μA [11]. Therefore, the batteries have to be replaced after a certain period of time. This is called battery replacement problem as mentioned in [12]. The replacement is sometimes very difficult, because the WSN nodes are deployed at hard-to-access locations. Hence, the replacement should be avoided if possible.

An EHS can extend the lifetime of a WSN node by utilizing environmental energy sources. A theoretical energy harvesting model is introduced in [13]. The model deals with the description of environmental energy sources and the energy sink (consumer). It consists of a so called (ρ, σ_1, σ_2) - source and a (ρ', σ) - consumer. The (ρ, σ_1, σ_2) - source is used to describe environmental energy sources. $P(t)$ is the continuous power output of the source at the time t. ρ is the average power output of this source. σ_1 and σ_2 describe the changes of the power output. The power consumption is also characterized with the average power consumption ρ' and the maximum power consumption σ. It is called (ρ', σ) - consumer. The following theorem has been proven in [13]:
Sustainable Performance at Eternity (Variable Consumption Profile): If a (ρ', σ) - consumer device is powered by a (ρ, σ_1, σ_2) - source model, has an energy storage capacity of $\sigma+\sigma_1+\sigma_2$, and $\rho' < \rho$, then the device can operate forever.

Two conclusions can be derived from this model. First, the EHS needs a buffer (ESU) to bridge the discontinuous power output of the environmental energy source. There are different types of ESUs that can be used, e.g. rechargeable batteries (nickel-cadmium NiCd, nickel-metal hydride NiMH, lithium ion LiIon or lithium-ion polymer LiPo) or double layer capacitors (DLCs). Second, if the average power consumption of the WSN node is lower than the average harvested power of the environmental energy source and the ESU is well dimensioned, then a perpetual operation is possible. This solves the battery replacement problem.

TABLE I
POWER DENSITIES OF DIFFERENT HARVESTING TECHNOLOGIES [2], [14].

Harvesting technology	Power density
Solar cells (outdoor at noon)	10-15 mW/cm^2
Solar cells (indoor) [15]	$\approx 10\,\mu W/cm^2$
Piezoelectric (shoe inserts)	$\approx 300\,\mu W/cm^3$
Vibration (small microwave oven)	$\approx 100\,\mu W/cm^3$
Thermoelectric (10°C gradient)	$\approx 40\,\mu W/cm^3$
Acoustic noise (100dB)	$\approx 1\,\mu W/cm^3$

The amount of harvestable energy depends on the type of the environmental energy source, on the power of the energy source and on the efficiency of the EHD. Table I shows a list of commonly used energy harvesting technologies and their power densities [2], [14]. It can be seen that the power density of the solar cell depends dramatically on the location. If the EHS is deployed outdoor, solar cells have the best power density. Therefore, the power consumption of the WSN node must be lower or the size of the EHD must be increased.

Dynamic voltage scaling (DVS) is commonly used to adapt the supply voltage of a processor depending on its clock frequency. The clock frequency can be reduced if the workload of the processor is low. In [16] the design issues for DVS are explained. In [17] they describe various algorithms that can be used for DVS. These algorithms *reduce energy consumption by changing processor speed and voltage at run-time depending on the needs of the applications running* [17]. In [18] DVS is applied to low power microprocessors. They use a low power StrongARM embedded Linux platform running a multimedia application for evaluation. In [19] they describe the energy savings using dynamic frequency and voltage scaling. The reduction of the frequency causes linear energy savings. Additionally, the adaption of the supply voltage to the lower clock frequency of the processor causes quadratic energy savings. However, all the works only consider the power consumption of the processor core itself. Other components are not included in their calculations. In [20] they have implemented a dynamic voltage and frequency scaling platform using an MSP430 ultra-low power microcontroller from Texas Instruments and other components of the shelf (COTS). This approach is very interesting, because they can vary the supply voltage of the MSP430 microcontroller. However, they do not consider the efficiency and the leakage currents of the voltage converter. The following section describes CADVS applied on the whole sensor node and the influence of the voltage converters.

III. Energy Efficient Supply of WSN Nodes

This section starts with the basics of the energy efficient supply of WSN nodes. Then, the scenario that is used for the calculations is explained in detail. Afterwards, two different voltage converters and their theoretical savings are discussed. Finally, CADVS and the possible energy savings are explained.

As mentioned in the introduction, an energy efficient supply of a WSN node is very important and can extend the lifetime of such a node. Each sensor node typically consists of a

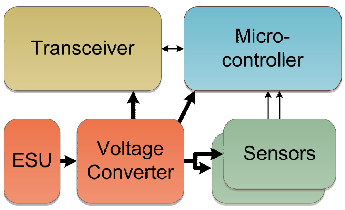


Fig. 2. Structure of a wireless sensor node (extended from [21]).

microcontroller and some sensors. To guarantee an efficient supply of the WSN node a conversion circuit is needed. Typically, this converter reduces the battery voltage to a constant value. Figure 2 shows the structure of a wireless sensor node. The converter is placed between the ESU (e.g. a battery or a DLC) and the rest of the sensor node. Usually, the whole sensor node is supplied by the same voltage. Multiple supply voltages are possible, but dramatically increase the complexity of the hardware, because each voltage level needs its own converter and between each voltage level some level shifter hardware is needed. Therefore, it is not useful to supply a sensor node with different voltage levels simultaneously in general.

Energy can be saved by switching off the components which are not needed currently by the application. This means the power supply of each component should be controllable by the microcontroller. Then, the driver software of a component or the application software itself can enable or disable the specific components. Therefore, a program or a part of a program has a list of the used hardware components L. The only component which is supplied all the time is the microcontroller. Hence it is always on the list L. The rest of the list depends on the components that are currently needed by the application (transceiver, sensors). Each of the components has its own supply voltage range with a lower limit $V_{sup,min,component}$. This limit is the lowest possible supply voltage (LPSV) of a component. The list L and the LPSVs $V_{sup,min,component}$ of the components result in a list of the LPSVs L_{LPSV}. Due to the fact that all hardware components of a sensor node are supplied by the same voltage, the minimum allowed voltage can be determined as shown in Equation 1.

$$V_{sup,min,node} = max(L_{LPSV}) \quad (1)$$

The list of the needed hardware components L depends on the application or even on the part of the application which is currently executed by the microcontroller. Therefore, the list changes over the time and consequently the list L_{LPSV} and the minimum allowed supply voltage $V_{sup,min,node}$. To be as efficient as possible, the supply voltage of the node should be equal to the minimum allowed supply voltage: $V_{CC} = V_{sup,min,node}$. This variation depends on the application and on the embedded hardware components of the mote. The following section describes the assumptions for further considerations.

TABLE II
HARDWARE COMPONENTS AND THEIR SUPPLY VOLTAGE RANGE.

Hardware Component	Supply Range	$V_{sup,min,component}$
Microcontroller	1.8 V to 3.6 V	1.8 V
Temperature Sensor	3.3 V to 3.3 V	3.3 V
Transceiver	2.4 V to 3.6 V	2.4 V

A. Scenario

A typical energy saving method used in WSNs is duty cycling (DC). The sensor nodes are only active for a short period of time t_{active}. During this time, the node collects the information of its environment, preprocesses the data and transmits these data. During the rest of the time t_{sleep}, the sensor node is placed into a sleep state to save as much energy as possible. Thus, one interval T consists of the active period t_{active} and the sleep period t_{sleep}.

An MSP430F1611 from Texas Instruments is used as microcontroller. This microcontroller has a supply voltage range from 1.8 V to 3.6 V [22]. To be able to use the full supply range the clock frequency of the microcontroller has to be limited to 4 MHz. The active mode current I_{AM} is calculated according to the datasheet. First, the current consumption at a clock frequency of 1 MHz and the given supply voltage V_{CC} is calculated as shown in Equation 2. Therefore, the current consumption at 1 MHz and 3 V is used ($I_{AM,1MHz,3V} = 500\,\mu$A). Then, the current consumption at the given clock frequency f can be calculated as shown in Equation 3).

$$I_{AM,1MHz}(V_{CC}) = I_{AM,1MHz,3V} + 210\,\mu\text{A/V} \cdot (V_{CC} - 3\,\text{V}) \quad (2)$$

$$I_{AM}(V_{CC}, f) = I_{AM,1MHz}(V_{CC}) \cdot f[\text{MHz}] \quad (3)$$

The sleep mode current I_{sleep} (low power mode 3, only the 32 kHz timer is active) is interpolated using the two values given in the datasheet [22] (Equation 4).

$$I_{sleep}(V_{CC}) = 1.3\,\mu\text{A} \cdot (V_{CC} - 2.2\,\text{V}) \cdot \frac{2.6\,\mu\text{A} - 1.3\,\mu\text{A}}{3\,\text{V} - 2.2\,\text{V}} \quad (4)$$

The WSN node should be used to measure the temperature of the environment. Therefore, the precise temperature sensor TMP05B from Analog Devices is embedded. It has a typical accuracy of 0.2 % if the supply voltage is in the range of 3.135 V to 3.465 V [23]. The full supply voltage range is from 3.0 V to 5.5 V. However, to guarantee the best accuracy, the supply voltage of the temperature sensor should be 3.3 V. Furthermore, the sensor node uses the MRF24J40MB 2.4 GHz IEEE802.15.4 transceiver module from Microchip. It has a supply voltage range from 2.4 V to 3.6 V [24]. Table II summarizes the embedded components and their supply voltage ranges.

Furthermore, it is assumed that the sensor node is in a sleep state (t_{sleep}) during 85 % of the interval T. Then the sensor node wakes up and performs some measurements (t_{meas}) during 5 %. After that, the measured data is preprocessed (t_{comp}) during 5 % of the interval. Therefore, no other hardware components are needed. Finally, the sensor node sends the

TABLE III
SUBINTERVALS AND THE NEEDED HARDWARE COMPONENTS.

Subinterval	Needed Components	$V_{sup,min,node}$
Sleep	Microcontroller	1.8 V
Active, Measurement	Microcontroller Temperature Sensor	3.3 V
Active, Computation	Microcontroller	1.8 V
Active, Communication	Microcontroller Transceiver	2.4 V

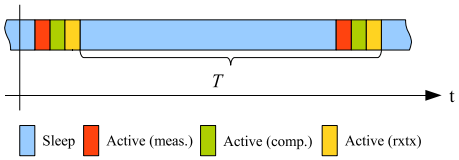


Fig. 3. Chronological sequence of an interval T.

results through the network (t_{comm}) during the last 5 % of the interval. The DC can be calculated according to Equation 5.

$$DC = \frac{t_{active}}{T} = \frac{t_{meas} + t_{comp} + t_{comm}}{T} \quad (5)$$

$$= \frac{15}{100} = 15\,\% \quad (6)$$

Table III shows the needed components of each part of the interval and the resulting minimum supply voltage of the node $V_{sup,min,node}$. Figure 3 shows the chronological sequence of the interval.

As shown in Fig. 2, a voltage converter is placed between the ESU and the components of the sensor node. The minimum supply voltage of the WSN node is 3.3 V and the maximum supply voltage is 3.6 V. This is a very small range and there is no battery type that operates well in this range. It would be possible to use a battery, which is not fully loaded but this would impair dramatically the usable capacity of the battery. Therefore, it is not possible to supply the node directly. The voltage converter reduces the battery voltage to a constant value. However, a little voltage drop $U_{drop,min}$ remains at the converter. The relation is shown in Equation 7.

$$U_{battery,min} = U_{drop,min} + U_{sup,node} \quad (7)$$

Therefore, the minimum voltage of battery $U_{battery,min}$ must be higher than the supply voltage of the node $U_{sup,node} = 3.3\,\text{V}$. For further calculations a minimum voltage drop of $U_{drop,min} = 0.2\,\text{V}$ of the converter is assumed [25], [26]. Table IV shows five possible battery types to supply the WSN node. All values are from [27]. The first column shows the type of the battery and the second column shows the number of needed cells which have to be connected in series to achieve the minimum supply voltage of the node. The third column lists the voltage values of the fully charged cell (or cells) and the fourth column shows the average voltage during a full discharge process. Finally, the last column shows the estimated usable capacity of the battery concerning the supply voltage

TABLE IV
POSSIBLE BATTERY TYPES FOR A SUPPLY OF THE WSN NODE [27].

Battery Type	Needed Cells	Start Voltage	Avg. Discharge Voltage	Usable Capacity
Alkaline	3	4.5 V	≈ 3.6 V	≈ 65 %
NiMH	3	4.14 V	≈ 3.75 V	≈ 89 %
NiCd	3	4.44 V	≈ 3.78 V	≈ 92 %
LiIon	1	4.1 V	≈ 3.76 V	≈ 95 %
LiPo	1	4.1 V	≈ 3.8 V	≈ 95 %

of the node and the minimum voltage drop of the converter. The minimum voltage of the battery is $U_{battery,min} = 3.5\,\text{V}$. The table shows that the LiIon and the LiPo rechargeable batteries have the best usable capacity. The alkaline battery is not rechargeable and is only given as reference. The low usable capacity is caused by the faster drop of the cell voltage compared to the other types.

B. Voltage Conversion Techniques

As mentioned before, energy can be saved if the voltage of the battery is reduced using a voltage converter. Basically, there are two different types of voltage converters which are explained in the following two sections.

1) Linear Voltage Regulators: This type provides a constant output voltage by adapting its internal resistance. So called low-dropout (LDO) regulators are linear voltage regulators which can operate with a low difference of the input and the output voltage.

Figure 4a shows the dependency of the current (blue trace) and the power consumption (red trace) of the microcontroller on the supply voltage during active mode at 4 MHz. The values are simulated using Equation 2. It is a linear dependency of the current consumption on the supply voltage. When using an LDO regulator, the difference between the battery voltage V_{Bat} and the supply voltage of the microcontroller V_{CC} drops at the regulator V_{LDO} (see Equation 8).

$$V_{Bat} = V_{LDO} + V_{CC} \quad (8)$$

The current through the microcontroller is equal to the current through the LDO (Equation 9).

$$I_{LDO} = I_{\mu C} \quad (9)$$

The power consumption of the microcontroller and the LDO can be calculated as shown in Equation 10.

$$P(V_{CC}) = I_{LDO} \cdot V_{Bat} \quad (10)$$

$$= I_{AM}(V_{CC}) \cdot V_{Bat}$$

It can be seen that the power consumption depends linearly on the supply voltage. The reason is that the input current of the LDO regulator is the same as the output current and therefore the current of the microcontroller.

Figure 4b shows the dependency of the current (blue trace) and the power consumption (red trace) of the microcontroller depending on the supply voltage during sleep mode (low power mode 3, only the 32 kHz timer is active). Due to the fact

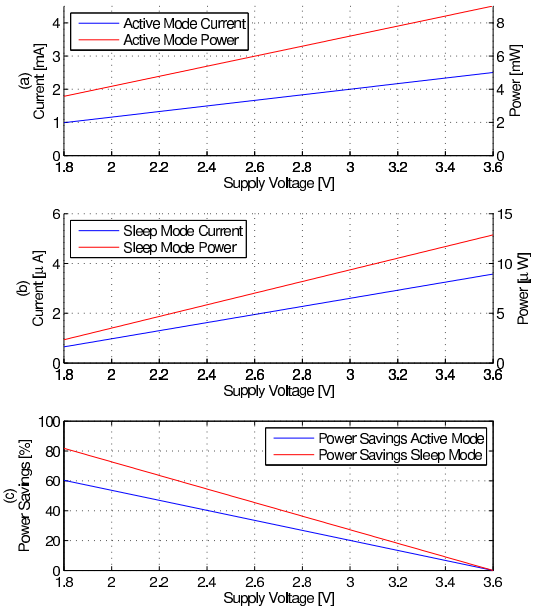


Fig. 4. Current and power consumption during active mode (a) and sleep mode (b) using a linear voltage regulator; (c) shows the theoretical power savings in active and sleep mode reffering to a supply voltage of 3.6 V.

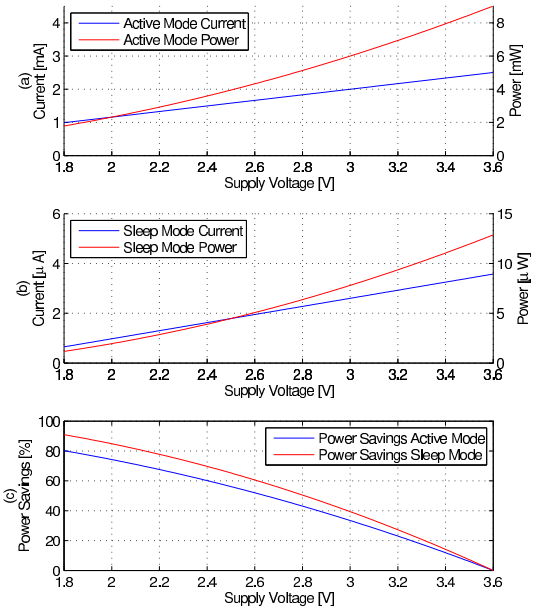


Fig. 5. Current and power consumption during active mode (a) and sleep mode (b) using a step down converter. (c) shows the theoretical power savings in active and sleep mode reffering to a supply voltage of 3.6 V.

that the sleep current depends also linear on the supply voltage the resulting power consumption has a linear dependency, too.

Finally, 4c shows the theoretical power savings using an LDO regulator. No losses are assumed here.

2) Step Down Converter: A step down converter, also called buck converter, provides a constant output voltage by using internal switching elements. Typically a coil or a capacitor is used in combination with the switches. The output voltage can be controlled by varying the timing of the switching elements (pulse-width modulation).

Figure 5a shows the dependency of the current (blue trace) and the power consumption (red trace) of the microcontroller depending on the supply voltage during active mode at 4 MHz. The values are simulated using Equation 2. When using a step down converter, the difference between the battery voltage V_{Bat} and the supply voltage of the microcontroller V_{CC} drops at the converter V_{Buck} (see Equation 11).

$$V_{Bat} = V_{StepDown} + V_{CC} \tag{11}$$

However, the average input current of the converter is unequal to the output current and therefore, it is different form the current through the microcontroller (Equation 12).

$$I_{Buck,input} \neq I_{Buck,output} = I_{\mu C} \tag{12}$$

The ideal step down converter has no losses and therefore, the input power of it is equal to the output power. This means that the input power depends only on the power consumption of the microcontroller as shown in Equation 13.

$$\begin{aligned} P_{IN}(V_{CC}) &= P_{OUT}(VCC) \\ &= I_{AM}(V_{CC}) \cdot V_{CC} \end{aligned} \tag{13}$$

It can be seen that the power consumption depends quadratically on the supply voltage. The reason is that the input current of the step down converter is lower than the output current.

Figure 5b shows the dependency of the current (blue trace) and the power consumption (red trace) of the microcontroller depending on the supply voltage during sleep mode (low power mode 3, only the 32 kHz timer is active). Due to the fact that the sleep current has the same influence on the power consumption, the power consumption depends quadratically on the supply voltage, too.

Finally, 5c shows the theoretical power savings using a step down converter. No losses are assumed here.

C. Component-Aware Dynamic Voltage Scaling

CADVS can be applied to reduce the power consumption of the hardware. As shown in Table III the minimum allowed supply voltage varies over the time. It is assumed that only a few messages are sent. Therefore, the energy of the transmissions is neglected at the following calculations. This can be done, because the main focus of this work is on the possible energy savings using CADVS and not on a correct energy estimation of an application running on the sensor node. Furthermore, it is

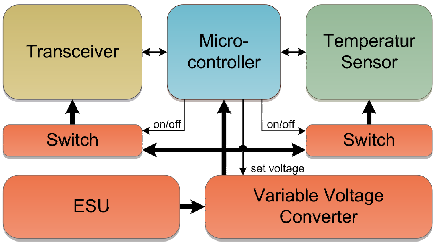


Fig. 6. Structure of the WSN node (extended from [21]). The power supply of the temperature sensor and of the transceiver can be switched off by the microcontroller. Furthermore, the variable voltage converter enables CADVS.

TABLE V
CURRENT CONSUMPTION OF THE COMPONENTS OF THE SENSOR NODE.

Component	Sleep	Active Measure	Active Computation	Active Commun.
V_{CC}	1.8 V	3.3 V	1.8 V	2.4 V
Microcontroller	650 nA	2.25 mA	0.992 mA	1.5 mA
Transceiver	0	0	0	25 mA
Temp. Sensor	0	0.37 mA	0	0
Power Switches	170 nA	402 nA	170 nA	259 nA
Sum	820 nA	2.62 mA	0.99 mA	26.5 mA

TABLE VI
POWER CONSUMPTION OF THE COMPONENTS OF THE SENSOR NODE.

Component	Sleep	Active Measure	Active Computation	Active Commun.
V_{CC}	1.8 V	3.3 V	1.8 V	2.4 V
Microcontroller	1170 nW	7.43 mW	1.786 mW	3.6 mW
Transceiver	0	0	0	60 mW
Temp. Sensor	0	1.221 mW	0	0
Power Switches	306 nW	1327 nW	306 nW	621 nW
Sum	1476 nW	8.65 mA	1.79 mW	63.6 mW

assumed that the transceiver and the temperature sensor can be switched off completely. This is necessary, because they have very high sleep currents (compared to the microcontroller). Fig. 6 shows the resulting block diagram of the sensor node. TPS22921 from Texas Instruments are assumed as power switches [28].

Table V and Table VI shows an overview of the current consumption and power consumption of the single components during the different subintervals.

As mentioned before, two different techniques can be used to convert the supply voltage of the WSN node. The following figures and tables show the possible power savings using an LDO regulator with fixed output voltage (3.3V), an LDO regulator with variable output voltage (CADVS), a step down converter with fixed output voltage (3.3 V) and a step down converter with variable output voltage (CADVS).

The voltage of the battery is assumed to be $V_{bat} = 3.77$ V. This is the mean of the average discharge voltages of the rechargeable batteries (see Table IV). The power savings are calculated using the LDO regulator with fixed output voltage as reference, because it is the simplest supply configuration. Table VII shows the power consumption and the theoretical

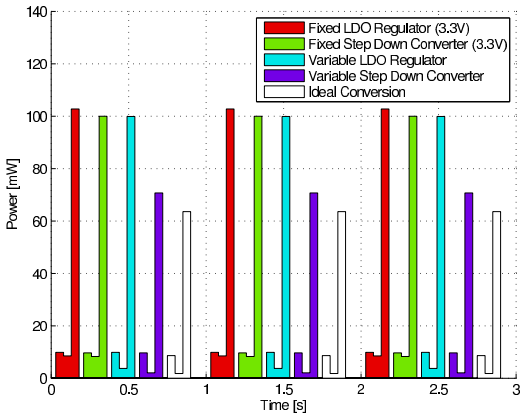


Fig. 7. Power consumption of the WSN node with four different supply configurations and ideal conversion.

power savings of the different subintervals using the four different supply configurations.

It is also important to consider the quiescent current of the voltage converters. The LDO regulator TPS78001 from Texas Instruments has a quiescent current of $I_{q,LDO} = 500$ nA [26]. The quiescent current of the step down converter TPS62120 also from Texas Instruments is $I_{q,stepdown} = 11\mu$A [25]. Furthermore, the efficiency of the TPS62120 is about 90 %. To adjust the output voltage a digital potentiometer is needed. The quiescent current of the ISL22313 from Intersil is $I_{q,dpot} = 1\,\mu$A [29]. The results of this calculation are shown in Table VIII.

Fig. 7 shows the power trace of the WSN node using the four different supply configurations. Three intervals are plotted. The white trace shows the minimum possiple supply power using ideal components without conversion losses and leakage currents.

The comparison of Table VII and Table VIII is very interesting. The best supply configuration of theoretical calculation (Table VII) is the variable step down converter. It is also the best supply configuration including the efficiency and the quiescent currents of the voltage converter. However, the energy savings of the variable step down converter drops significantly. The reason is the energy wasting of the step down converter during the sleep subinterval. Analyzing the savings during each subinterval, it can be seen that the step down converter has much higher power savings at higher power consumptions and lower supply voltages. The average power savings depends on the DC of the WSN node. This is shown in Fig. 8. Therefore, the sleep duration is varied and the durations of the active subintervals are kept constant. The shorter the sleep duration, the higher is the DC. It can be seen that at a higher DC the step down converter using CADVS provides the best results. At a lower DC, the LDO regulator using CADVS provides the best results. However,

TABLE VII
POWER CONSUMPTION AND THEORETICAL POWER SAVINGS OF THE FOUR DIFFERENT SUPPLY CONFIGURATIONS.

Description	Fixed LDO Reg. 3.3 V	Fixed Step Down Conv. 3.3 V	Variable LDO Reg.	Variable Step Down Conv.
Sleep Power	13.32 µW	11.66 µW	3.09 µW	1.48 µW
Measurement Power	9.89 mW	8.65 mW	9.89 mW	8.65 mW
Computation Power	8.49 mW	7.43 mW	3.74 mW	1.79 mW
Communication Power	102.74 mW	89.93 mW	99.89 mW	63.59 mW
Average Power	6.07 mW	5.31 mW	5.68 mW	3.70 mW
Sleep Power Savings	0.0 %	12.5 %	76.8 %	88.9 %
Measurement Power Savings	0.0 %	12.5 %	0.0 %	12.5 %
Computation Power Savings	0.0 %	12.5 %	56.0 %	79.0 %
Communication Power Savings	0.0 %	12.5 %	2.8 %	38.1 %
Average Power Savings	0.0 %	12.5 %	6.4 %	39.0 %

TABLE VIII
POWER CONSUMPTION AND POWER SAVINGS OF THE FOUR DIFFERENT SUPPLY CONFIGURATIONS INCLUDING THE QUIESCENT CURRENT OF THE VOLTAGE CONVERTER.

Description	Fixed LDO Reg. 3.3 V	Fixed Step Down Conv. 3.3 V	Variable LDO Reg.	Variable Step Down Conv.
Quiescent Current Voltage Conv.	0.5 µA	11.0 µA	1.5 µA	12.0 µA
Quiescent Power Voltage Conv.	1.9 µW	41.5 µW	5.7 µW	45.2 µW
Sleep Power	15.20 µW	54.42 µW	8.75 µW	46.88 µW
Measurement Power	9.89 mW	9.66 mW	9.89 mW	9.66 mW
Computation Power	8.49 mW	8.30 mW	3.75 mW	2.03 mW
Communication Power	102.74 mW	99.97 mW	99.90 mW	70.70 mW
Average Power	6.07 mW	5.94 mW	5.68 mW	4.16 mW
Sleep Power Savings	0.0 %	−258.0 %	42.5 %	−208.4 %
Measurement Power Savings	0.0 %	2.3 %	−0.0 %	2.3 %
Computation Power Savings	0.0 %	2.3 %	55.9 %	76.1 %
Communication Power Savings	0.0 %	2.7 %	2.8 %	31.2 %
Average Power Savings	0.0 %	2.1 %	6.3 %	31.5 %

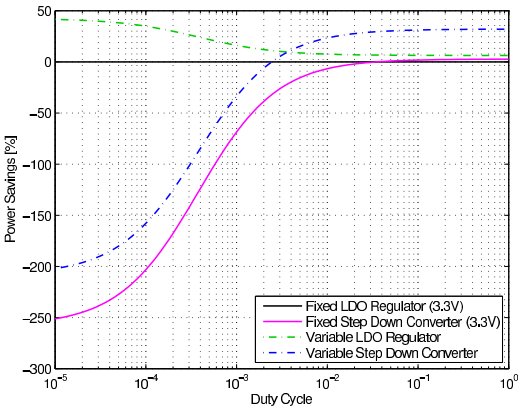


Fig. 8. Power savings of the WSN node depending on the DC.

there are applications with a dynamic DC. This means the DC is adapted to the current circumstances. The best solution is to combine an LDO regulator and a step down converter. Only the voltage converter with the better power savings is active.

IV. EVALUATION

This section evaluates the results of the previous sections. Unfortunately, no fully functional prototype was available to perform a complete evaluation. However, the current consumption of the transceiver and of the microprocessor was measured. Fig. 9 shows the current consumption of the transceiver depending on the supply voltage during receiving.

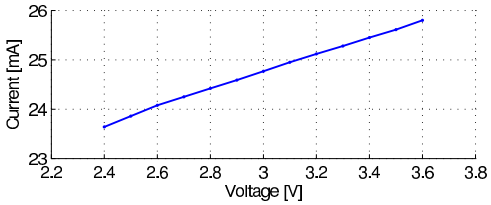


Fig. 9. Current consumption of the transceiver depending on the supply voltage.

Fig. 10 shows the active current consumption of the MSP430F1611 microprocessor. The active mode current is similar to the current specified in the datasheet. It can be seen that the difference of the lowest and the highest current of the transceiver is smaller than the difference of the microcontroller. Furthermore, the measurements have shown that the assumption of the linear dependency of the current consumption on the supply voltage is acceptable.

V. CONCLUSION AND FUTURE WORK

This work demonstrated different possibilities of the supply of WSN nodes. It has shown different voltage conversion

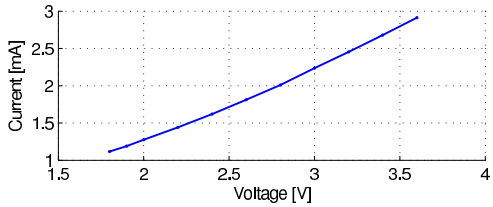


Fig. 10. Active mode current consumption of the MSP430F1611 depending on the supply voltage.

techniques and their advantages and disadvantages concerning the energy savings. A realistic scenario has been developed to show how DVS can be used to save up to 31.5 % of the energy compared to a constant voltage supply using the introduced scenario while achieving the same end-user performance. Furthermore, it demonstrated the energy savings dependent on the DC of a WSN node considering the leakage currents and the conversion efficiency of the four different voltage conversion techniques. Finally, two measurements have shown that the assumptions of a linear dependency of the current consumption on the supply voltage are acceptable. Future activities will target an implementation of a prototype to evaluate the different voltage conversion techniques with real components.

References

[1] R. Min, M. Bhardwaj, S.-H. Cho, N. Ickes, E. Shih, A. Sinha, A. Wang, and A. Chandrakasan, "Energy-centric enabling tecumologies for wireless sensor networks," *Wireless Communications, IEEE*, vol. 9, no. 4, pp. 28 – 39, aug. 2002.

[2] V. Raghunathan, A. Kansal, J. Hsu, J. Friedman, and M. Srivastava, "Design considerations for solar energy harvesting wireless embedded systems," in *Proceedings of the 4th international symposium on Information processing in sensor networks*. IEEE Press, 2005, p. 64.

[3] M. Rahimi, H. Shah, G. Sukhatme, J. Heideman, and D. Estrin, "Studying the feasibility of energy harvesting in a mobile sensor network," *Robotics and Automation, 2003. Proceedings. ICRA '03. IEEE International Conference on*, vol. 1, pp. 19 – 24 vol.1, sep. 2003.

[4] A. Janek, C. Trummer, C. Steger, R. Weiss, J. Preishuber-Pfluegl, and M. Pistauer, "Simulation based verification of energy storage architectures for higher class tags supported by energy harvesting devices," *Microprocessors and Microsystems*, vol. 32, no. 5-6, pp. 330–339, 2008, dependability and Testing of Modern Digital Systems. [Online]. Available: http://www.sciencedirect.com/science/article/B6V0X-4S80XHN-2/2/22459f6e78fe4b7c4111c8af88fa7fb4

[5] K. Langendoen, A. Baggio, and O. Visser, "Murphy loves potatoes: experiences from a pilot sensor network deployment in precision agriculture," *Parallel and Distributed Processing Symposium, 2006. IPDPS 2006. 20th International*, p. 8 pp., apr. 2006.

[6] N. Watthanawisuth, A. Tuantranont, and T. Kerdcharoen, "Microclimate real-time monitoring based on zigbee sensor network," *Sensors, 2009 IEEE*, pp. 1814 –1818, oct. 2009.

[7] P. Juang, H. Oki, Y. Wang, M. Martonosi, L. S. Peh, and D. Rubenstein, "Energy-efficient computing for wildlife tracking: design tradeoffs and early experiences with zebranet," in *ASPLOS-X: Proceedings of the 10th international conference on Architectural support for programming languages and operating systems*. New York, NY, USA: ACM, 2002, pp. 96–107.

[8] A. Lindgren, C. Mascolo, M. Lonergan, and B. McConnell, "Seal-2-seal: A delay-tolerant protocol for contact logging in wildlife monitoring sensor networks," *Mobile Ad Hoc and Sensor Systems, 2008. MASS 2008. 5th IEEE International Conference on*, pp. 321 –327, sep. 2008.

[9] K. Lorincz, B.-r. Chen, G. W. Challen, A. R. Chowdhury, S. Patel, P. Bonato, and M. Welsh, "Mercury: a wearable sensor network platform for high-fidelity motion analysis," in *SenSys '09: Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems*. New York, NY, USA: ACM, 2009, pp. 183–196.

[10] N. Xu, S. Rangwala, K. K. Chintalapudi, D. Ganesan, A. Broad, R. Govindan, and D. Estrin, "A wireless sensor network for structural monitoring," in *SenSys '04: Proceedings of the 2nd international conference on Embedded networked sensor systems*. New York, NY, USA: ACM, 2004, pp. 13–24.

[11] C. Enz, N. Scolari, and U. Yodprasit, "Ultra Low-Power Radio Design for Wireless Sensor Networks," in *IEEE Int. Workshop on Radio-Frequency Integration Technology*, 2005, pp. 1–17, (Invited Keynote).

[12] M. Minami, T. Morito, H. Morikawa, and T. Aoyama, "Solar biscuit: A battery-less wireless sensor network system for environmental monitoring applications," *The 2nd International Workshop on Networked Sensing Systems*, 2005.

[13] A. Kansal, D. Potter, and M. B. Srivastava, "Performance aware tasking for environmentally powered sensor networks," in *SIGMETRICS '04/Performance '04: Proceedings of the joint international conference on Measurement and modeling of computer systems*. New York, NY, USA: ACM, 2004, pp. 223–234.

[14] P. He, Q. Cui, and X. Guo, "Efficient solar power scavenging and utilization in mobile electronics system," *Green Circuits and Systems (ICGCS), 2010 International Conference on*, pp. 641 –645, jun. 2010.

[15] S. Roundy, D. Steingart, L. Frechette, P. Wright, and J. Rabaey, "Power sources for wireless sensor networks," in *Wireless Sensor Networks*, ser. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 2004, vol. 2920, pp. 1–17.

[16] T. D. Burd and R. W. Brodersen, "Design issues for dynamic voltage scaling," in *Proceedings of the 2000 international symposium on Low power electronics and design*, ser. ISLPED '00. New York, NY, USA: ACM, 2000, pp. 9–14. [Online]. Available: http://doi.acm.org/10.1145/344166.344181

[17] T. Simunic, L. Benini, A. Acquaviva, P. Glynn, and G. De Micheli, "Dynamic voltage scaling and power management for portable systems," in *Proceedings of the 38th annual Design Automation Conference*, ser. DAC '01. New York, NY, USA: ACM, 2001, pp. 524–529. [Online]. Available: http://doi.acm.org/10.1145/378239.379016

[18] J. Pouwelse, K. Langendoen, and H. Sips, "Dynamic voltage scaling on a low-power microprocessor," in *Proceedings of the 7th annual international conference on Mobile computing and networking*, ser. MobiCom '01. New York, NY, USA: ACM, 2001, pp. 251–259. [Online]. Available: http://doi.acm.org/10.1145/381677.381701

[19] A. Sinha and A. Chandrakasan, "Dynamic power management in wireless sensor networks," *Design Test of Computers, IEEE*, vol. 18, no. 2, pp. 62 –74, Mar./Apr. 2001.

[20] H. C. Powell, A. T. Barth, and J. Lach, "Dynamic voltage-frequency scaling in body area sensor networks using cots components," in *Proceedings of the Fourth International Conference on Body Area Networks*, ser. BodyNets '09. ICST, Brussels, Belgium, Belgium: ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2009, pp. 15:1–15:8. [Online]. Available: http://dx.doi.org/10.4108/ICST.BODYNETS2009.6015

[21] D. Puccinelli and M. Haenggi, "Wireless sensor networks: applications and challenges of ubiquitous sensing," *Circuits and Systems Magazine, IEEE*, vol. 5, no. 3, pp. 19–31, 2005.

[22] TexasInstruments, "Msp430f15x, msp430f16x, msp430f161x mixed signal microcontroller," www.focus-ti.com, SLAS368F, 2009.

[23] AnalogDevices, "±0.5°C accurate pwm temperature sensor in 5-lead sc-70," www.analog.com, D03340 Rev.B, 2006.

[24] Microchip, "Mrf24j40mb data sheet - 2.4 ghz ieee std. 802.15.4 20 dbm rf transceiver module," www.microchip.com, DS70599B, 2009.

[25] TexasInstruments, "Tps62120 - 15V, 75mA high efficient buck converter," www.focus-ti.com, SLVSAD5, July 2010.

[26] ——, "Tps780 series - 150mA, low-dropout regulator, ultralow-power, i_q 500nA with pin-selectable, dual-level output voltage," www.focus-ti.com, SLVSAD5, May 2008.

[27] D. Linden and T. B. Reddy, *Handbook of Batteries*, 3rd ed. New York: McGraw-Hill, 2002.

[28] TexasInstruments, "Tps22921 - low input voltage, ultra-low r_{ON} load switches," www.focus-ti.com, SLVS749A, December 2008.

[29] Intersil, "Isl22313 - single digitally controlled potentiometer (xdcp(tm))," www.intersil.com, FN6421.0, July 2007.

Evaluation of Component-Aware Dynamic Voltage Scaling for Mobile Devices and Wireless Sensor Networks

Leander B. Hörmann, Philipp M. Glatz, Christian Steger and Reinhold Weiss
Institute for Technical Informatics
Graz University of Technology, Austria
Email: {Leander.Hoermann@, Philipp.Glatz@, Steger@, RWeiss@}TUGraz.at

***Abstract*—Energy efficiency is very important for mobile devices and wireless sensor networks (WSNs), because the consumable energy is limited. Therefore, the operating time of such devices depends mainly on the capacity of the energy storage component and on the average power consumption of the device. The power consumption depends on the supply voltage and on the activated components of the hardware. This work presents the evaluation of component-aware dynamic voltage scaling (CADVS). This low power technique combines the power-down of unused components and the minimization of the supply voltage. Typically, each component of the hardware (microcontroller, transceiver, sensors) has its own supply voltage range. Therefore, the minimum allowed supply voltage depends on the activated components. However, the activated components and consequently the minimum allowed supply voltage varies over time. CADVS uses voltage converter to adjusts the supply voltage of the hardware to save as much energy as possible. This work presents the evaluation of six different voltage converters. It has been shown that CADVS can be used to save up to 38.7 % of the energy compared to a constant voltage supply using the introduced scenario while achieving the same end-user performance.**

***Keywords*-component aware dynamic voltage scaling; dynamic voltage scaling; energy efficient supply; mobile devices; perpetual operation; wireless sensor networks**

I. Introduction and Related Work

Mobile devices and wireless sensor network (WSN) nodes need to be as energy efficient as possible, because their available energy is limited. Mobile devices need their own power source to supply the device everywhere. Typically, they are powered by normal (Alkaline) or rechargeable batteries (nickel-cadmium (NiCd), nickel-metal hydride (NiMH), lithium-ion (LiIon) or lithium-ion polymer (LiPo)). The operating time depends on the power consumption of the device and on the capacity of the batteries. Due to the fact that the devices are portable, the size of the batteries and consequently the available energy is limited.

WSNs are used to measure physical quantities of the environment in application areas with mobility or bad infrastructure. A few examples of application areas are precision agriculture [1], wildlife monitoring [2], human health care [3] and structural health monitoring [4]. Due to the lack of wired infrastructure, each sensor node needs its own energy source. Typically, sensor nodes are powered by batteries or energy harvesting systems (EHSs) [5]–[7]. EHSs use energy harvesting devices to convert environmental energy into electrical energy. The type of the environmental energy source and the efficiency of the energy harvesting device have a high influence on the amount of harvestable energy. Table I shows typical energy harvesting technologies and their power densities [7], [8]. The size of a sensor node is often limited. Consequently, the available power consumption of the WSN node is also limited. Due to the fact, that *energy from the environment is generally unpredictable, discontinuous, and unstable* [10], an energy storage component is needed to bridge periods with insufficient harvestable energy.

Table I
Power densities of different energy harvesting technologies [7], [8].

Harvesting technology	Power density
Solar cells (outdoor at noon)	10-15 mW/cm^2
Solar cells (indoor) [9]	$\approx 10\,\mu W/cm^2$
Piezoelectric (shoe inserts)	$\approx 300\,\mu W/cm^3$
Vibration (small microwave oven)	$\approx 100\,\mu W/cm^3$
Thermoelectric ($10°C$ gradient)	$\approx 40\,\mu W/cm^3$
Acoustic noise (100dB)	$\approx 1\,\mu W/cm^3$

Typical energy storage components are rechargeable batteries and double layer capacitors (DLCs). Table II shows the most common types and their characteristics. It can be seen that the DLC has very low energy density compared to the battery types, but the lifetime of the DLC is much higher. The lifetime is measured in charge-discharge cycles. Here, it indicates the number of charge-discharge cycles which are necessary to reduce the usable capacity of the battery to 80 % of its nominal value. All the mentioned rechargeable batteries have a lifetime of 500 to 1000 cycles. However, the DLC has a lifetime of up to 500000 cycles.

Table II shows also the start and end voltage of a discharge phase. The terminal voltage of an energy storage component starts at a high voltage (start voltage) and drops down to the end voltage during discharge. If the hardware is designed properly, it is possible to supply the hardware directly. This means the energy storage component is connected to the mobile device or the WSN node directly. Then, the

978-1-4577-0351-5/11/$26.00 ©2011 IEEE

Table II
POSSIBLE BATTERY TYPES AND THEIR CHARACTERISTICS TO SUPPLY MOBILE DEVICES AND WSN NODES [11].

Battery Type	Start Voltage	End Voltage	Average Discharge Voltage	Energy Density
Alkaline	1.5 V	0.7 V	≈ 1.2 V	≈ 145 Wh/kg
NiMH	1.38 V	0.8 V	≈ 1.25 V	≈ 75 Wh/kg
NiCd	1.48 V	0.8 V	≈ 1.26 V	≈ 35 Wh/kg
LiIon	4.1 V	2.5 V	≈ 3.76 V	100 – 158 Wh/kg
LiPo	4.1 V	2.8 V	≈ 3.8 V	136 – 190 Wh/kg
DLC [12]	2.7 V	0.0 V	1.35 V	5 Wh/kg

supply voltage of the hardware is equal to the terminal voltage of the energy storage component. Therefore, the voltage of the fully charged energy storage component must be higher than the minimum allowed supply voltage. The emerging difference is wasted energy which is explained in the following.

The heart of almost all electronic devices is a microprocessor or microcontroller. Typically, it is a digital CMOS (complementary metal oxide semiconductor) circuit. Such a circuit has a static and a dynamic power consumption [13]. The static power consumption results from leakage and bias currents. It can be neglected in systems with a power consumption of more than 1 mW. The dynamic power consumption of a CMOS circuit can be calculated as shown in (1). It is assumed that the switching elements (the gates) of the CMOS circuit have a common switching capacity C.

$$P_{Dynamic} = C \cdot f \cdot V_{Supply}^2 \quad (1)$$

It can be seen that the clock frequency f of the circuit has linear influence and the supply voltage V_{Supply} has quadratic influence on the dynamic power consumption $P_{Dynamic}$. Therefore, much energy can be saved if the supply voltage of devices using CMOS circuits is as low as possible. Voltage conversion circuits are used to convert the terminal voltage of the energy storage components to the desired supply voltage of the hardware. There are different types of such voltage conversion circuits and two of them are explained in Section III.

Another method to save a lot of energy is to switch off the unused hardware components completely. We have described this idea in [14]. The microcontroller of the hardware controls the power supply of the components of a mobile device or a WSN node, e.g. sensors, communication module. They are only switched on if they are needed by the microcontroller.

The combination of the controllable power switches of each component and the minimization of the needed supply voltage results in component-aware dynamic voltage scaling (CADVS). Our previous work [15] describes the theory of this low-power principle in detail.

The rest of the paper is organized as follows: Section II presents the application scenario of a WSN node which is used for further discussion. Section III shows the principles of energy efficient supply of mobile devices and WSN nodes. Therefore, different battery types and different hardware components are considered. Section IV explains CADVS applied to mobile devices and WSN nodes. Section V shows the measurement setup which is used to evaluate the different voltage conversion circuits. Section VI presents and discusses the measurement results of the different circuits with and without CADVS. Finally, Section VII concludes the paper and shows directions for future work.

Table III
HARDWARE COMPONENTS AND THEIR SUPPLY VOLTAGE RANGE.

Hardware Component	Supply Range	$V_{sup,min,component}$
Microcontroller	1.8 V to 3.6 V	1.8 V
Temperature Sensor	3.3 V to 3.3 V	3.3 V
Transceiver	2.4 V to 3.6 V	2.4 V

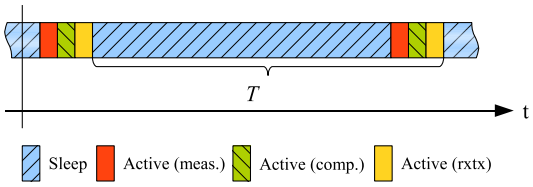


Figure 1. Chronological sequence of an interval T.

II. APPLICATION SCENARIO

This section describes the application scenario and the used hardware components. The goal is to design a low-power WSN node which can be used to measure the temperature of the environment.

The WSN node consists of three main components. The first one is an MSP430F1611 microcontroller from Texas Instruments. It has a supply voltage range from 1.8 V to 3.6 V [16]. The microcontroller is operated at 4 MHz clock frequency to be able to use the full supply range.

The second main component is the precise temperature sensor TMP05B from Analog Devices to measure the temperature of the environment. To achieve an accuracy of 0.2 %, the supply voltage has to be in the range of 3.135 V to 3.465 V [17]. To get the best accuracy, the supply voltage should be 3.3 V.

The third main component is the 2.4 GHz IEEE802.15.4 transceiver module MRF24J40MB from Microchip. Its supply voltage range is from 2.4 V to 3.6 V [18]. Table III lists the main components and their supply voltage ranges.

The chronological sequence of an interval is shown in Figure 1. It is assumed that the sleep state (t_{sleep}) lasts for 85 % of the interval T. The measurement state (t_{meas}), the computation state (t_{comp}) and the communication state (t_{comm}) last for 5 % of the interval T each.

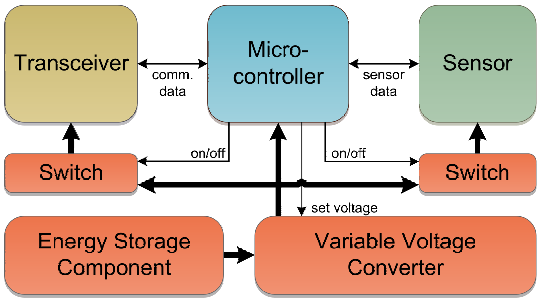


Figure 2. Structure of the WSN node (extended from [19]). The power supply of the sensor and of the transceiver can be switched off by the microcontroller. Furthermore, the variable voltage converter enables CADVS.

III. Energy Efficient Design of Mobile Devices and WSN Nodes

This section describes three low-power principles for an energy efficient design of mobile devices or WSN nodes. It also provides an overview of two common voltage conversion techniques.

The first principle is duty cycling. The microcontroller remains in a power saving sleep state during periods without workload. Therefore, no special hardware is needed, because the MSP430 family supports this feature. This method is often used in low-power mobile devices and WSNs. One interval T is split into an active phase t_{active} and a sleep phase t_{sleep}. The duty cycle can be calculated according to (2).

$$DutyCycle = \frac{t_{active}}{T} \quad (2)$$

The calculation of the duty cycle of the scenario is shown in (3).

$$DutyCycle_{Scenario} = \frac{t_{active}}{T} = \frac{t_{meas} + t_{comp} + t_{comm}}{T} = 15\,\% \quad (3)$$

The second principle is power supply switching. The power supply of the unused components is switched off completely. The hardware must support this feature. The sensor node in Figure 2 uses two power switches controlled by the microcontroller to enable or disable the power supply of the transceiver and the sensor.

The third principle is the minimization of the supply voltage. The supply voltage is set to the minimum allowed value depending on the currently active components. However, the hardware must support this feature. The microcontroller controls the output voltage of variable voltage converter as shown in Figure 2.

Due to the fact that the terminal voltage of the LiIon and the LiPo battery is higher than the maximum allowed supply voltage of the node, a voltage converter is needed to reduce the battery voltage. We discuss only converter circuits which reduce the terminal voltage.

Basically, there are two different types of voltage converters which are explained in the following two sections.

A. Linear Voltage Regulators

Linear voltage regulators provide a constant output voltage by changing its internal resistance depending on the load current. A low-dropout (LDO) regulator is a special type which can operate with a low difference of the input and the output voltage. The voltage drop at the regulator can be calculated using the battery voltage V_{Bat} and the supply voltage of the microcontroller V_{CC} (4).

$$V_{regulator} = V_{Bat} - V_{CC} \quad (4)$$

The average input current of the microcontroller is equal to the average input current of the regulator (5). Quiescent and leakage currents of the regulator are not considered here.

$$I_{regulator} = I_{\mu C} \quad (5)$$

The power consumption can be calculated as shown in (6).

$$P(V_{CC}) = I_{regulator} \cdot V_{Bat} = I_{\mu C}(V_{CC}) \cdot V_{Bat} \quad (6)$$

B. Step Down Converter

Step down converters, also called buck converters, provide a constant output voltage by changing the timing of internal switching elements. Typically, inductors or capacitors are used in combination with these switches. The voltage drop at the converter can be calculated using the battery voltage V_{Bat} and the supply voltage of the microcontroller V_{CC} (7).

$$V_{Buck} = V_{Bat} - V_{CC} \quad (7)$$

However, the average input current of the microcontroller is not equal to the average input current of the regulator (8).

$$I_{Buck,input} \neq I_{Buck,output} = I_{\mu C} \quad (8)$$

The input power is equal to the output power (ideal step down converter). Therefore, the input power depends only on the power consumption of the microcontroller (9).

$$P_{IN}(V_{CC}) = P_{OUT}(VCC) = I_{\mu C}(V_{CC}) \cdot V_{CC} \quad (9)$$

IV. Component Aware Dynamic Voltage Scaling

As mentioned in the introduction, the combination of the controllable power switches of each hardware component and the minimization of the needed supply voltage results in CADVS. The application running on the hardware enables only the needed hardware components. The information of these active components is stored in a list L. The microcontroller is always on the list, because it is supplied all the time in this example.

Each hardware component has its own supply voltage range with a lowest possible supply voltage (LPSV)

Table IV
SUBINTERVALS AND THE NEEDED HARDWARE COMPONENTS.

Phase	Needed Comp. L	Min. allowed volt. list L_{LPSV}	V_{CC}
Sleep	[Microcontroller]	[1.8 V]	1.8 V
Measurement	[Microcontroller, Temp. Sensor]	[1.8 V, 3.3 V]	3.3 V
Computation	[Microcontroller]	[1.8 V]	1.8 V
Communication	[Microcontroller, Transceiver]	[1.8 V, 2.4 V]	2.4 V

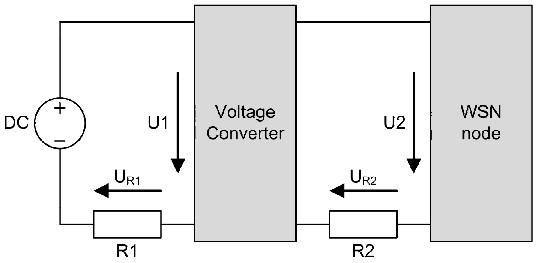


Figure 3. Measurment setup to evaluate CADVS.

$V_{sup,min,component}$. The combination of the list of the active components and the their LPSV results in a list of LPSVs of the active components L_{LPSV}. The optimal supply voltage can be determined as shwon in (10).

$$V_{sup,min,node} = max(L_{LPSV}) \tag{10}$$

The active components change during program execution and therefore, the lists L and L_{LPSV}. Consequently, the optimal supply voltage $V_{sup,min,node}$ changes also during program execution. To save as much energy as possible the supply voltage of the node should be set to this value (11).

$$V_{CC} = V_{sup,min,node} \tag{11}$$

Table IV shows the list of the needed components L of each phase, the list of the minimum allowed supply voltage L_{LPSV} and the resulting supply voltage of the node V_{CC} of the scenario.

V. Measurement Setup and Voltage Converters

This sections shows the measurement setup and the five different voltage converter circuits which are used to evaluate CADVS. The measurement setup is similar to the setup which we have used to measure EHSs in [20]. It is shown in Figure 3. The NI PXI-6221 DAQ measurement device from National Instruments is used to sample the voltage drop U_{R1} and U_{R2} at the shunt resistors $R1$ and $R2$ as well as the input voltage $U1$ of the voltage converter and the input voltage $U2$ of the WSN node. The value of both shunt resistors is $4\,\Omega$. Therefore, the input power can be calculated as shown in Equation 12.

$$P_{conv,input} = U1 \cdot \frac{U_{R1}}{R1} \qquad P_{node,input} = U2 \cdot \frac{U_{R2}}{R2} \tag{12}$$

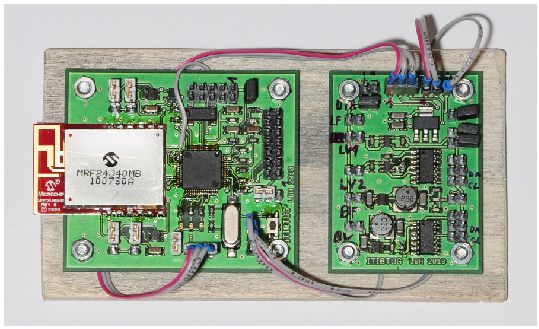


Figure 4. Prototype of the WSN node (left board) and the five different converter circuits (right board).

The sleep current is too low to be measured with this setup. Therefore, a precise multimeter (Fluke 289) with an direct current resolution of $0.01\,\mu$A and an accuracy of $\pm 0.075\,\%+ 0.2\,\mu$A is used. An image of the implemented prototype is provided in Figure 4.

A. Voltage Converter Circuits

The five different voltage converter circuits are shown in Figure 5. The following list describes the different voltage converter circuits.

LDO1FIX: The first circuit (Figure 5(a)) is only implemented to show a bad voltage converter. This converter has a very high quiescent current of 5 mA according to the datasheet [21]. This current is higher than the supply current of the MSP430 at full power mode.

LDO2FIX, LDO2VAR: The second circuit (Figure 5(b)) is used for two measurements. First, it is used to evaluate the LDO regulator circuit with a constant output voltage of 3.3 V (LDO2FIX). Second it is used in a two-step mode (LDO2VAR). This regulator supports the possibility to change the output voltage between two fix values. The selected version of the converter supports 2.2 V and 3.3 V output voltage. Therefore, it can be used for CADVS in a slightly limited way.

BUCK1FIX: The third circuit (Figure 5(c)) shows the implementation of a step down converter with a constant output voltage of 3.3 V.

LDO3VAR: The fourth circuit (Figure 5(d)) is one of the two circuits which enables real CADVS. The output voltage of the LDO regulator can be adjusted by changing the resistance of the digital potentiometer AD5241 between 1.8 V and 3.3 V.

BUCK2VAR: The fifth circuit (Figure 5(e)) is the second circuit which enables real CADVS. The output voltage of the step down converter can be adjusted by changing the resistance of the digital potentiometer AD5241 between 1.8 V and 3.3 V.

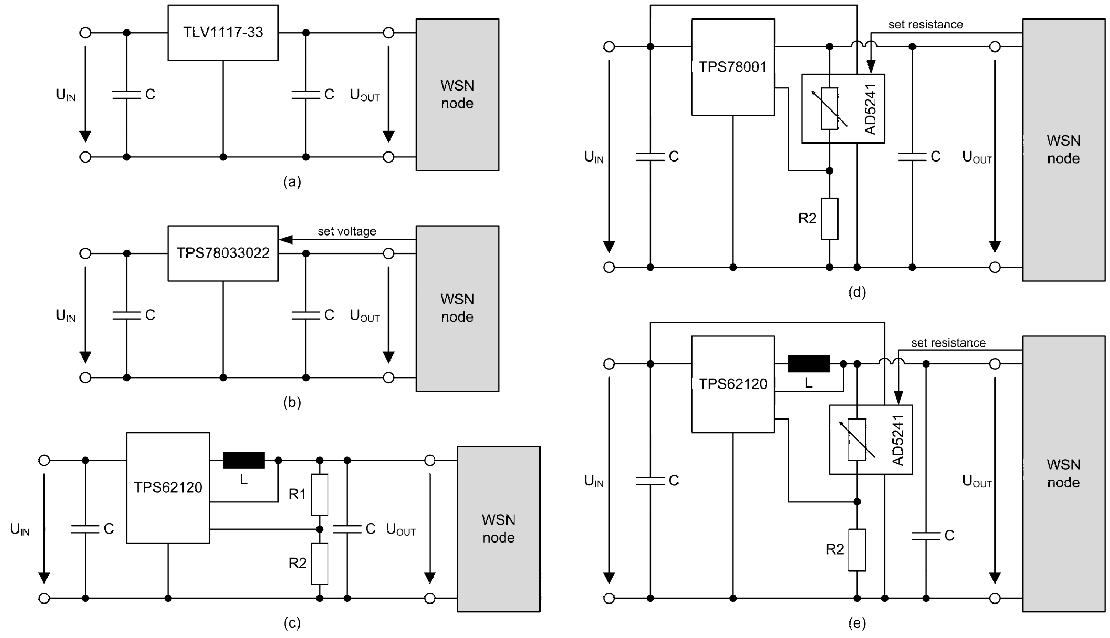


Figure 5. Five different voltage converter circuits which are used to evaluate CADVS. (a) shows the converter circuit using the TLV1117-33 LDO regulator with a constant output voltage of 3.3 V. (b) shows the converter circuit using the TPS78033022 LDO regulator. It is used with a constant output voltage of 3.3 V and with variable output voltage (2.2 V and 3.3 V). (c) shows the converter circuit using the TPS62120 step down converter with a constant output voltage of 3.3 V. (d) shows the converter circuit using the TPS78001 LDO regulator with a variable output voltage (1.8 V to 3.6 V). (e) shows the converter circuit using the TPS62120 step down converter with a variable output voltage (1.8 V to 3.6 V).

VI. Results

This section presents the measurement results of this work. First, the input current and the input power of the WSN node is measured without a voltage converter.

A. Direct Supply

The WSN node is connected to the power supply without using a voltage converter. The supply voltage is varied from 1.8 V to 3.6 V and the input current and power are plotted. This is done for each phase of the interval.

Figure 6 shows the current and power consumption of the WSN node depending on the supply voltage during the sleep phase 6(a), the measurement phase 6(b), the computation phase 6(c) and the communication phase 6(d).

It can be seen that the current consumption has a close-to-dependency on the supply voltage during the measurement and the computation phase. Due to the very low current consumption during the sleep phase, other influences as mentioned in the introduction cause the non- dependency.

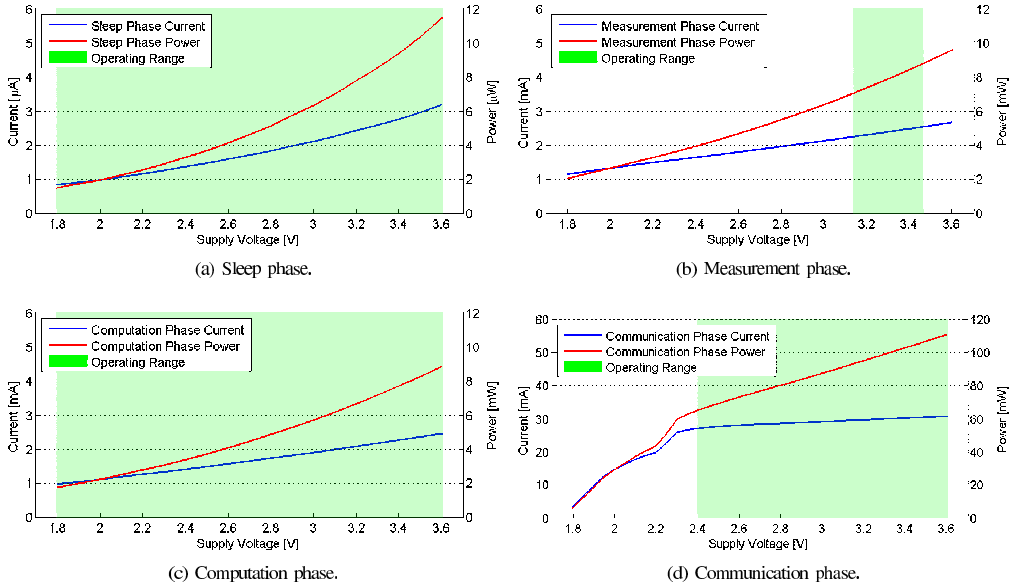
B. Converter Output Voltages

The measurement shows the output voltages of the six different converters depending on the input voltage. The results are shown in Figure 7. It can be seen that the LDO1FIX converter has a very high drop voltage. The terminal voltage of the battery must be higher than at the other converters to get an output voltage of 3.3 V. A lot of energy is wasted when using this converter. Therefore, it is unsuitable to supply the WSN node and it is not considered in further results.

The minimum supply voltage at the sleep phase and the computation phase is 1.8 V. Figures 7(a) and 7(c) shows that the LDO3VAR and the BUCK2VAR converter can set this voltage. The LDO2VAR converter cannot set this low voltage. It is only possible to set 2.2 V. The other converters (LDO2FIX and BUCK1FIX) supply the node with a voltage of 3.3 V.

The minimum supply voltage during the measurement phase must be 3.3 V. Therefore, all converter circuits try to set this output voltage.

During the communication phase (Figure 7) the minimum supply voltage is 2.4 V. LDO3VAR and the BUCK2VAR converter can set this voltage. The lower voltage (2.2 V) of the LDO2VAR converter is too low for this phase. Therefore, the output voltage has to be set to 3.3 V to ensure proper functionality. The other converters (LDO2FIX and BUCK1FIX) supply the node with a voltage of 3.3 V.



(a) Sleep phase.

(b) Measurement phase.

(c) Computation phase.

(d) Communication phase.

Figure 6. Current and power consumption of the WSN node during the four different phases of the interval depending on the supply voltage of the WSN node.

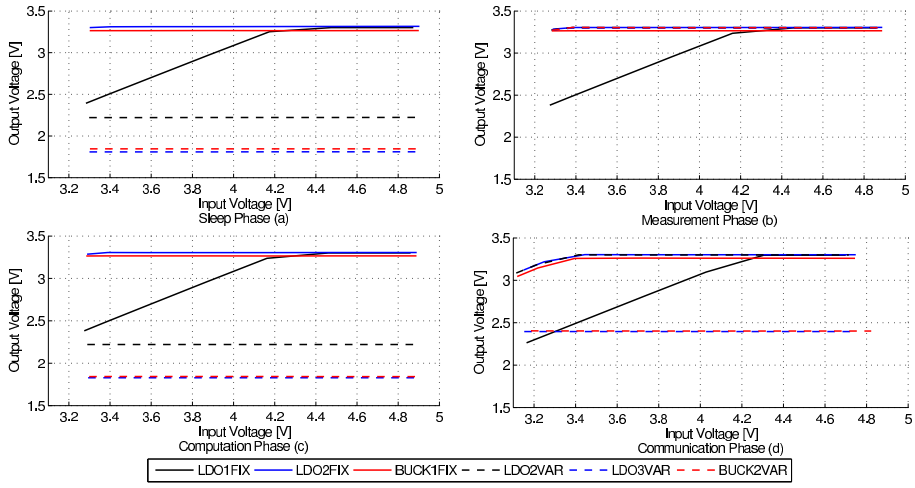


Figure 7. Output voltages of the six different converter circuits depending on the input voltage. (a) shows the output voltages during the sleep phase. (b) shows the output voltages during the measurement phase. (c) shows the output voltages during the computation phase. (d) shows the output voltages during the communication phase.

C. CADVS Results

following. Table V shows the quiescent power of the voltage converter, the overall power consumption (input power of the voltage converter) during the different phases, the efficiency of the voltage converter during the different phases and the power savings compared to the LDO2FIX converter circuit. This converter circuit is used as reference,

Table V
POWER CONSUMPTION, EFFICIENCY AND POWER SAVINGS OF THE FIVE DIFFERENT VOLTAGE CONVERTERS INCLUDING THE QUIESCENT POWER OF THEM. THE AVERAGE VALUES REFER TO THE INTERVAL OF THE SCENARIO. THE INPUT VOLTAGE IS 3.9 V.

Voltage Converter	LDO2FIX	BUCK1FIX	LDO2VAR	LDO3VAR	BUCK2VAR
Quiescent Power of Voltage Conv.	0.5 μW	73.2 μW	0.8 μW	45.7 μW	93.9 μW
Power during Sleep	10.73 μW	83.13 μW	5.38 μW	49.07 μW	97.38 μW
Power during Measurement	9.37 mW	8.25 mW	9.38 mW	9.78 mW	8.90 mW
Power during Computation	8.44 mW	7.42 mW	4.65 mW	3.59 mW	1.63 mW
Power during Communication	95.26 mW	99.45 mW	94.35 mW	81.24 mW	56.89 mW
Average Power	5.71 mW	5.87 mW	5.47 mW	4.83 mW	3.50 mW
Efficiency during Sleep	80.6 %	10.0 %	48.7 %	3.2 %	1.7 %
Efficiency during Measurement	84.9 %	93.2 %	84.6 %	84.3 %	91.9 %
Efficiency during Computation	85.0 %	93.4 %	56.7 %	45.3 %	103.3 %
Efficiency during Communication	87.9 %	94.9 %	88.6 %	63.8 %	91.6 %
Average Efficiency	87.4 %	93.6 %	86.9 %	64.7 %	89.8 %
Power Savings during Sleep	0.0 %	−674.7 %	49.8 %	−357.3 %	−807.5 %
Power Savings during Measurement	0.0 %	12.0 %	−0.1 %	−4.3 %	5.0 %
Power Savings during Computation	0.0 %	12.1 %	44.9 %	57.5 %	80.7 %
Power Savings during Communication	0.0 %	−4.4 %	1.0 %	14.7 %	40.3 %
Average Power Savings	0.0 %	−2.8 %	4.1 %	15.3 %	38.7 %

because it is one of the simplest and cheapest converter circuits.

The results are very interesting. First of all, it can be seen that the quiescent power of the LDO2FIX and the LDO2VAR are much lower than that of the other converters. The high quiescent power of the BUCK1FIX and the BUCK2VAR was expected. The high quiescent power of the LDO3VAR was not expected. It is caused by the supply current of the digital potentiometer and its circuit.

The power consumption during the sleep phase ranges from 5.38 μW to 97.38 μW and this is a big difference. The result is a power saving of −807.5 % during the sleep phase of the BUCK2VAR. The supply power of the WSN node itself ranges from 2 μW at 1.8 V to 8.5 μW at 3.3 V. Therefore, the overall power consumption depends mainly on the quiescent power of the voltage converters BUCK1FIX, LDO3VAR and BUCK2VAR. The two other converters (LDO2FIX and LDO2VAR) have a very low quiescent current. The crucial factor here is the supply voltage of the WSN node. It is 3.3 V at the LDO2FIX converter and 2.2 V at the LDO2VAR converter. This is the reason for the halving of the overall power consumption.

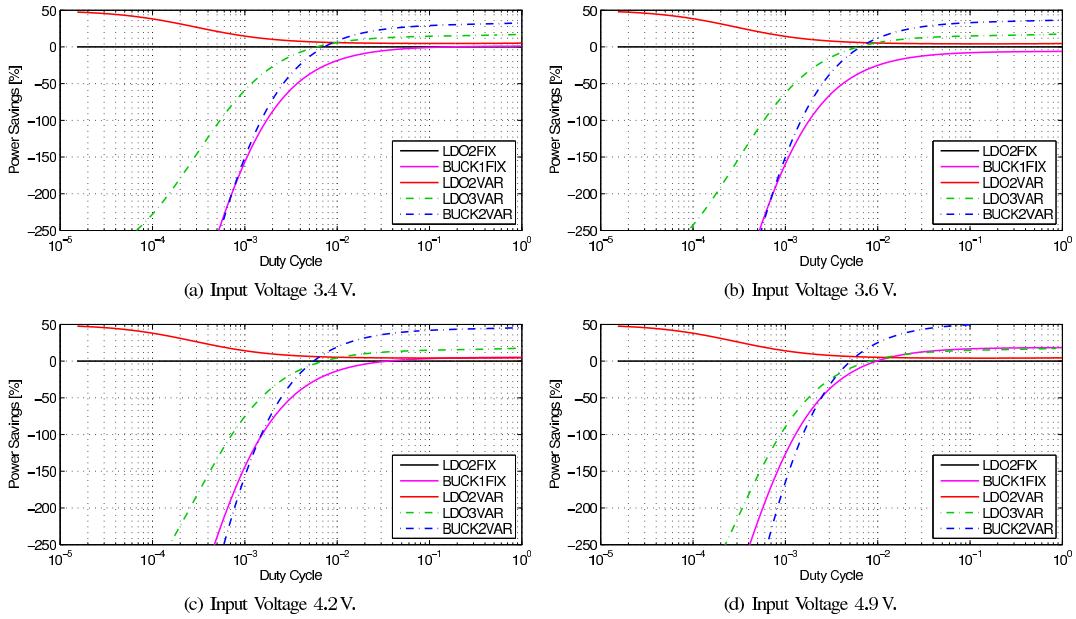
The bad performance of the converter circuits with high quiescent current can also be seen at the efficiency during the sleep phase. During all other phases, the step down converters have the best performance. The regulators cannot achieve such a high efficiency, because they are converting the excess power into heat. However, the 103.3 % is caused by measurement errors. The measurement setup has an error of about 2 % for each power measurement channel as derived in [20].

The power savings of the different voltage converters and the different phases vary very much. As already mentioned, a high quiescent current causes a much higher overall power consumption and consequently a wasting of energy during the sleep phase. During the measurement phase, only the step down converters are able to save power compared to the LDO2FIX converter (reference converter). During the computation phase, the LDO2VAR converter and the LDO3VAR converter perform better than the BUCK1FIX converter, because of the reduced supply voltage of the WSN node. The BUCK2VAR converter performs best, because of the reduced supply voltage and the high conversion efficiency. During the communication phase, only the LDO3VAR converter and the BUCK2VAR converter was able to reduce the supply voltage of the WSN node. Therefore, only these two were able to significantly save power. The bad performance of the BUCK1FIX converter was not expected and can be caused by measurement errors or bad performance of the converter at higher currents. The average power savings during the whole interval clearly show that the BUCK2VAR performs best.

However, the overall power savings heavily depend on the segmentation of the interval. The results are completely different if the sleep phase is much longer than in the scenario. This can be seen in Figure 8. All four plots are based on the measurement results as discussed before. The duration of the sleep phase is varied by keeping the durations of the other phases.

The main result of these four plots is that the lower the duty cycle the better is the LDO2VAR converter. As discussed before, this converter has the lowest quiescent current of the variable voltage converters. The drawback of the only two voltage steps is not important at low duty cycles. The input voltage has no significant influence on the power savings of this converter.

On the other hand the higher the duty cycle is the better performs the BUCK2VAR converter. It has the highest



(a) Input Voltage 3.4 V.

(b) Input Voltage 3.6 V.

(c) Input Voltage 4.2 V.

(d) Input Voltage 4.9 V.

Figure 8. Power savings of the five different voltage converter circuits depending on the duty cycle at four differ input voltages of the converters.

efficiency and a variable output voltage. Furthermore, the power savings decrease with decreasing input voltage to over 50 % at 4.9 V.

All these results are also valid for all kinds of mobile devices with low power techniques like duty cycling.

Finally, to achieve best performance on all duty cycles, a combination of the two voltage converters LDO2VAR and BUCK2VAR would perform best. However, the drawback of this solution is the need of extra hardware to switch between these two converters, but it is the best solution for applications with a high variation of the duty cycle. Therefore, the selection of the right voltage converter mainly depends on the specifications of the application running on the mobile device or WSN node.

VII. Conclusion

This work has evaluated different possibilities of the supply of mobile devices or WSN nodes. A realistic scenario has been introduced to get real world results. A prototype has been implemented to evaluate six different voltage converter circuits. It has been shown, that it is possible to save 38.7 % of the energy compared to a constant supply using the introduced scenario while achieving the same end-user performance. In addition, it demonstrated the energy savings depending on the duty cycle of the application. It has been shown, that the leakage currents of the different converter circuits have a high influence at low duty cycles. Therefore, the best converter circuit depends on the application running on the mobile device or WSN node. Future activities will target an improvement of the prototype and a combination of two converters to achieve the best performance over all duty cycles.

References

[1] K. Langendoen, A. Baggio, and O. Visser, "Murphy loves potatoes: experiences from a pilot sensor network deployment in precision agriculture," *Parallel and Distributed Processing Symposium, 2006. IPDPS 2006. 20th International*, p. 8 pp., apr. 2006.

[2] A. Lindgren, C. Mascolo, M. Lonergan, and B. McConnell, "Seal-2-seal: A delay-tolerant protocol for contact logging in wildlife monitoring sensor networks," *Mobile Ad Hoc and Sensor Systems, 2008. MASS 2008. 5th IEEE International Conference on*, pp. 321 –327, sep. 2008.

[3] K. Lorincz, B.-r. Chen, G. W. Challen, A. R. Chowdhury, S. Patel, P. Bonato, and M. Welsh, "Mercury: a wearable sensor network platform for high-fidelity motion analysis," in *SenSys '09: Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems*. New York, NY, USA: ACM, 2009, pp. 183–196.

[4] N. Xu, S. Rangwala, K. K. Chintalapudi, D. Ganesan, A. Broad, R. Govindan, and D. Estrin, "A wireless sensor network for structural monitoring," in *SenSys '04: Proceedings of the 2nd international conference on Embedded networked sensor systems*. New York, NY, USA: ACM, 2004, pp. 13–24.

[5] R. Min, M. Bhardwaj, S.-H. Cho, N. Ickes, E. Shih, A. Sinha, A. Wang, and A. Chandrakasan, "Energy-centric enabling technologies for wireless sensor networks," *Wireless Communications, IEEE*, vol. 9, no. 4, pp. 28 – 39, aug. 2002.

[6] M. Rahimi, H. Shah, G. Sukhatme, J. Heideman, and D. Estrin, "Studying the feasibility of energy harvesting in a mobile sensor network," *Robotics and Automation, 2003. Proceedings. ICRA '03. IEEE International Conference on*, vol. 1, pp. 19 – 24 vol.1, sep. 2003.

[7] V. Raghunathan, A. Kansal, J. Hsu, J. Friedman, and M. Srivastava, "Design considerations for solar energy harvesting wireless embedded systems," in *Proceedings of the 4th international symposium on Information processing in sensor networks*. IEEE Press, 2005, p. 64.

[8] P. He, Q. Cui, and X. Guo, "Efficient solar power scavenging and utilization in mobile electronics system," *Green Circuits and Systems (ICGCS), 2010 International Conference on*, pp. 641 –645, jun. 2010.

[9] S. Roundy, D. Steingart, L. Frechette, P. Wright, and J. Rabaey, "Power sources for wireless sensor networks," in *Wireless Sensor Networks*, ser. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 2004, vol. 2920, pp. 1–17.

[10] A. Janek, C. Trummer, C. Steger, R. Weiss, J. Preishuber-Pfluegl, and M. Pistauer, "Simulation based verification of energy storage architectures for higher class tags supported by energy harvesting devices," *Microprocessors and Microsystems*, vol. 32, no. 5-6, pp. 330–339, 2008.

[11] D. Linden and T. B. Reddy, *Handbook of Batteries*, 3rd ed. New York: McGraw-Hill, 2002.

[12] Maxwell Technologies, "Bcap0310 p270 t10 - datasheet - bc power series radial d cell 310f ultracapacitor," http://www.maxwell.com/docs/DATASHEET_DCELL_POWER_1014625.PDF, 2011 February, 1014625.1.

[13] J. Pouwelse, K. Langendoen, and H. Sips, "Dynamic voltage scaling on a low-power microprocessor," in *Proceedings of the 7th annual international conference on Mobile computing and networking*, ser. MobiCom '01. New York, NY, USA: ACM, 2001, pp. 251–259.

[14] L. B. Hörmann, P. M. Glatz, C. Steger, and R. Weiss, "A wireless sensor node for river monitoring using msp430 and energy harvesting," in *European DSP in Education and Research Conference. Proceedings*. Texas Instruments, 2010, pp. 140–144.

[15] ——, "Energy efficient supply of wsn nodes using component-aware dynamic voltage scaling," in *Wireless Conference (EW), 2011 European*, April 2011, pp. 147–154.

[16] Texas Instruments, "Msp430f15x, msp430f16x, msp430f161x mixed signal microcontroller," www.focus-ti.com, SLAS368F, 2009.

[17] Analog Devices, "±0.5°C accurate pwm temperature sensor in 5-lead sc-70," www.analog.com, D03340 Rev.B, 2006.

[18] Microchip, "Mrf24j40mb data sheet - 2.4 ghz ieee std. 802.15.4 20 dbm rf transceiver module," www.microchip.com, DS70599B, 2009.

[19] D. Puccinelli and M. Haenggi, "Wireless sensor networks: applications and challenges of ubiquitous sensing," *Circuits and Systems Magazine, IEEE*, vol. 5, no. 3, pp. 19–31, 2005.

[20] P. Glatz, L. B. Hörmann, C. Steger, and R. Weiss, "A system for accurate characterization of wireless sensor networks with power states and energy harvesting system efficiency," *Pervasive Computing and Communications Workshops (PERCOM Workshops), 2010 8th IEEE International Conference on*, pp. 468 –473, mar. 2010.

[21] Texas Instruments, "Tlv1117 - adjustable and fixed low-dropout voltage regulator," www.focus-ti.com, SLVS561J, 2004.

6th IEEE International Workshop on Practical Issues in Building Sensor Network Applications SenseApp 2011, Bonn

Measuring the State-of-Charge - Analysis and Impact on Wireless Sensor Networks

Leander B. Hörmann, Philipp M. Glatz, Christian Steger and Reinhold Weiss
Institute for Technical Informatics
Graz University of Technology
Inffeldgasse 16, 8010 Graz, Austria
Email: {Leander.Hoermann@, Philipp.Glatz@, Steger@, RWeiss@}TUGraz.at

***Abstract*—Wireless sensor networks (WSNs) are typically used in application areas without wired infrastructure or mobility. Therefore, each sensor node needs its own energy supply unit. Sustainable WSNs are powered by energy harvesting systems (EHSs). These systems harvest and buffer the energy from the environment into rechargeable batteries or double layer capacitors.**

Due to the fact that the connectivity of the WSN depends on every single sensor node, it is important to know the state-of-charge (SoC) of the buffer to determine the remaining operating time. Therefore, each node has to measure its SoC, calculate the remaining operating time and populate this information through the network. The measurement is usually done by an integrated analog-to-digital converter (ADC) of the microcontroller. Each ADC has a specified error that has to be taken into account at the calculations of the remaining operating time. This work presents the error analysis of the SoC measurement using different energy storage components.

***Keywords*-wireless sensor networks; state-of-charge measurement; state-of-charge simulation; error analysis**

I. Introduction and Related Work

Wireless sensor networks (WSNs) are often used in application areas without wired infrastructure or mobility. Precision agriculture [1], [2], wildlife monitoring [3], [4], human health care [5] and structural health monitoring [6] are only a few examples. A WSN consists of a large number of sensor nodes, also called motes. These motes collect information of their environment and transmit the processed information through the sensor network. Typically, the packets are transmitted in a multi-hop manner, i.e. the packets are sent from one mote to the next till the destination is reached. Therefore, the link between two nodes depends also on the operability of other motes.

The lack of wired infrastructure causes the need of dedicated energy sources for each mote. Basically, there are two different possibilities of energy storage components (ESCs). First, conventional batteries can be used. These batteries are cheap but cannot be recharged. Therefore, the operating time of the mote is limited to the lifetime of the battery. Furthermore, the leakage current of a battery makes it impossible to extend the operating time in any order by reducing the power consumption of the mote [7] as shown in Figure 1. Often, the motes are applied at hard-to-access locations and the replacement is very time-consuming and expensive which is also known as battery

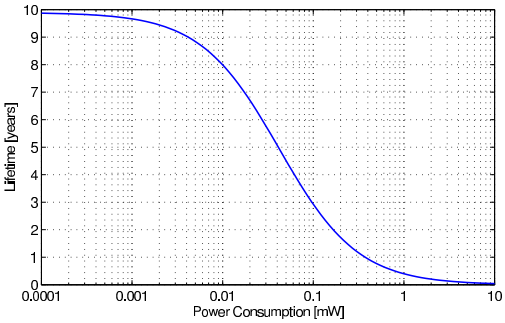


Fig. 1. Lifetime of a battery-powered system versus average power consumption using a simple battery model of a single AA cell (NiMH) with 2.6 Ah and a leakage current of 30 μA [7].

replacement problem [8]. Therefore, these types of ESCs are not considered in further discussion.

The second possibility is to use rechargeable ESCs and so called energy harvesting systems (EHSs) [9]–[11]. They convert the environmental energy into electrical energy and a perpetual operation is possible. An ESC is needed to supply the mote during periods with insufficient harvestable energy. A commonly used environmental energy source is solar radiation. The radiation is converted into electrical energy by solar cells. A part of the harvested energy has to be stored to enable a continuous operation during phases without light (e.g. night).

There are many different types of rechargeable ESCs, e.g. nickel-metal hydride (NiMH) rechargeable batteries, lithium-ion (LiIon) rechargeable batteries and double layer capacitors (DLCs). The Heliomote module [10] is an EHS using NiMH rechargeable batteries as ESC. It is also possible to combine different ESC types as shown in [12]. The Prometheus module uses a LiIon rechargeable battery as a long-term storage and a DLC as short-term storage. Recent advances in DLC technology enabled the use of DLCs without other ESCs to store the energy. In previous work we have shown the perpetual supply of mote using an EHS consisting of a solar cell and two powerful DLCs [13], [14].

As mentioned before, the connectivity of a WSN depends on the connectivity between the single motes and thus the

978-1-61284-928-7/11/$26.00 ©2011 IEEE

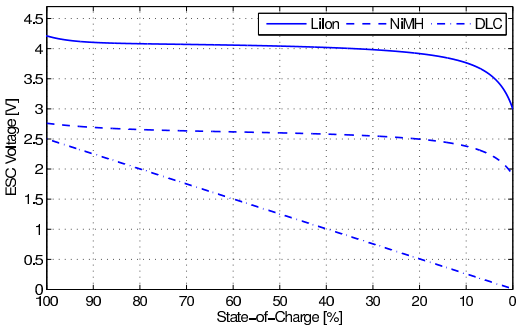


Fig. 2. ESC voltage dependent on the state-of-charge of three different ESCs.

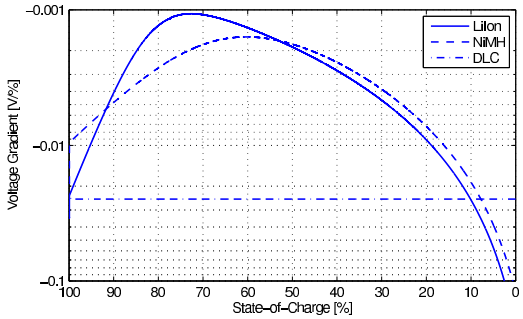


Fig. 3. Gradient of the ESC voltage dependent on the state-of-charge of three different ESCs.

operational availability of them. Therefore, it is necessary to know the current operational state of each mote. Especially, the current state-of-charge (SoC) of the ESC is important to estimate the remaining operating time. Furthermore, power-aware routing [15] relieves single nodes with low SoC of network traffic.

There are many different possibilities to determine the SoC of an ESC as listed in [16]. Most accurate methods need the measurement of the current or the charge running out of the ESC. These are complex measurements and a lot of additional components are required. However, a simple method is to measure the open circuit voltage (OCV). Due to the fact that the load currents of motes are very low (about 2 mA in active mode without transmission), the OCV can be estimated by measuring the supply voltage of the mote. Furthermore, most microcontrollers have integrated analog-to-digital converters (ADCs), which can be used to measure the voltage. It is a cheap and simple solution.

This work presents an error analysis of the SoC determination using off-the-shelf components. The rest of the paper is organized as follows: Section II describes the simulation setup. Section III shows the results of the error analysis and the impact on WSNs. Section IV briefly evaluates the battery model. Finally, Section V concludes the paper and shows directions for future work.

II. Concept and Model

This work is based on simulation results of different ESCs. Therefore, the analysis is as flexible as possible and parameters can be changed easily. The simulation uses the SimPowerSystems toolbox of Matlab/Simulink. A ready-to-use battery model of this toolbox is employed to simulate a LiIon rechargeable battery and two NiMH rechargeable batteries connected in series. The DLC uses a simple capacitor model. The rated capacity of both batteries is set to 1000 mAh, which is typical value. The capacitor of the DLC is set to 310 F and the initial voltage to 2.5 V. The dependency of the voltage on the SoC of different ESCs is shown in Figure 2. All three elements are discharged by a constant current load of only 1 mA.

As mentioned before, the ADC of a microcontroller is used to measure the voltage of the ESC. Such an ADC has a specific resolution and accuracy. As reference component the MSP430F1611 microcontroller has been chosen. It is an ultra-low power 16-bit RISC microcontroller, which is often used in motes (e.g. TelosB [17]). It has an integrated 12-bit ADC with a maximum unadjusted error of ±5 LSB. Furthermore, it is assumed that the microcontroller is supplied through a buck-boost voltage converter to ensure a constant supply voltage of $V_{sup} = 3.3\,\mathrm{V}$ with high precision. An example of such a converter is the TPS63031 from Texas Instruments with a precision of ±33 mV [18]. The supply voltage is used as reference voltage by the ADC of the microcontroller.

The terminal voltage of the LiIon battery cannot be measured directly, because it is higher than the supply voltage. Therefore, an ideal voltage divider is assumed. This makes the results comparable to each other. The corresponding voltage of one LSB can be calculated according to (1).

$$V_{LSB} = \frac{V_{reference}}{2^{resolution}} = \frac{V_{sup}}{2^{12}} = 0.806\,\mathrm{mV} \qquad (1)$$

The accuracy of the supply voltage is assumed to be $\Delta V_{sup} = \pm 33\,\mathrm{mV}$. Equation (2) shows the calculation of the total accuracy of the voltage measurement.

$$\Delta V_{err} = \Delta V_{sup} + 5 \cdot V_{LSB} = 37.03\,\mathrm{mV} \qquad (2)$$

The simulated terminal voltage of the ESCs and the calculated measurement error is used to determine the error of the SoC measurement which is shown in the next section.

III. Error Analysis of SoC Measurement

As shown in Figure 2, the start and the end of the trace of both batteries are steeper than the middle part of the trace. The DLC has a linear drop of the voltage, because of the constant current load. This fact has direct influence on the error of the SoC measurement.

Figure 3 shows the gradient of the terminal voltage of the ESC (V_{ESC}) during the discharge process. The gradient can

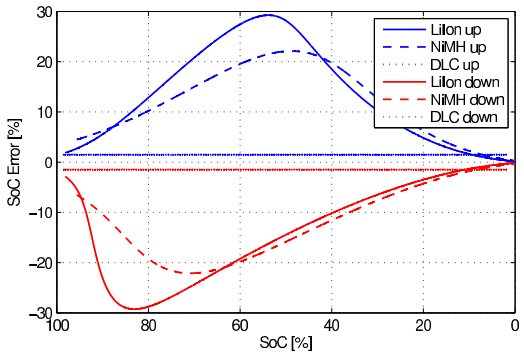


Fig. 4. Errors of the ESC voltage measurement dependent on the state-of-charge of three different ESCs. The blue lines represent the error upwards and the red lines represent the error downwards.

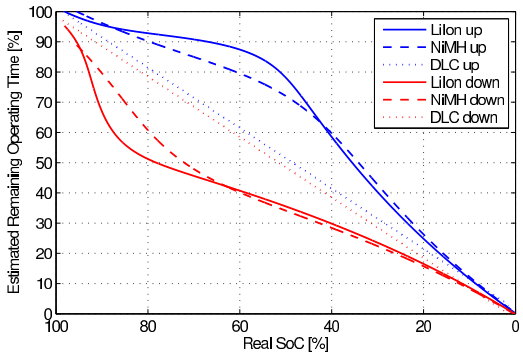


Fig. 5. Upper and lower limit of the remaining operating time dependent on the real SoC of three different ESCs.

be calculated according to (3). It can be used to determine the error of the SoC measurement.

$$\text{grad}(V_{ESC}) = \frac{\partial V_{ESC}(SoC)}{\partial SoC} \approx \frac{\Delta V_{ESC}(SoC)}{\Delta SoC} \quad (3)$$

However, the gradient is only an upper bound. The curvature of the terminal voltage causes a containment of the error. Furthermore, the error upwards (real value plus measurement error) is different to the error downwards (real value minus measurement error). The calculation of these errors is shown in (4) and (5).

$$SoC_{err,up}(i) = SoC(V_{ESC}(i) + \Delta V_{err}) - SoC(V_{ESC}(i)) \quad (4)$$

$$SoC_{err,down}(i) = SoC(V_{ESC}(i) - \Delta V_{err}) - SoC(V_{ESC}(i)) \quad (5)$$

The calculated errors are shown in Figure 4. It can be seen that the error upwards of batteries' SoC measurements has a maximum at a high SoC and decreases with decreasing SoC. Therefore, the measurement becomes more accurate at a lower SoC.

The error upwards is important to estimate the remaining operating time. The worst case is if the measured voltage is higher than the true voltage, because then the calculation of the remaining operating time will result in a too high value. To be able to compare the results of the three different ESCs the remaining operating time is normalized. The results are shown in Figure 5.

It can be seen that the error of the SoC measurement at the DLC is very low compared to the other ESCs. The reason is the constant gradient of the terminal voltage. Therefore, the determination of the SoC using only a voltage measurement is sufficient. The determination of the batteries' SoC could be enhanced by further measurements, e.g. current and charge measurement. However, this would increase the complexity of the mote and also the power consumption. A very simple energy measurement solution is presented in [19]. They use the switching frequency of the voltage regulator to gain an accuracy of $\pm 20\,\%$. However, the final implementation depends on the application area of the mote.

A. Impact on the Network

The network has to deal with measurement errors of the current SoC. All applications, services and protocols that use the current SoC should consider it. For example a power-aware routing algorithm should not have a hard decision boarder to determine the route with the highest remaining operating time. It should rank the possible routes and choose not only the best route but also other routes according to their SoCs. Also a power-aware load balancing algorithm should spread the load to a group of motes with the highest SoC and not only using the mote with the highest SoC.

Furthermore, the knowledge of the measurement error depending on the current SoC should be stored at each mote. It is possible that heterogeneous networks consist of different motes using different ESCs. Therefore, it is easier to determine the error of the SoC measurement or of the estimation of the remaining operating time locally and transmit it over the network. In addition, it is possible that each mote is able to adapt the error according to local conditions (e.g. temperature, age of battery).

The results can also be considered in WSN simulation environments which supports hardware simulation. A measurement error can be introduced to investigate different errors on the network behavior. In [20] we have introduced a simulation environment written in SystemC-AMS which supports a simulation of hardware and software of energy harvesting WSNs.

IV. Model Evaluation and Results

A 2600 mAh LiIon rechargeable battery cell is used to evaluate the battery model. Figure 6 shows the measured terminal voltage and the simulated voltages using the standard LiIon model and an adapted model. The adapted model is fit to the real LiIon battery. It can be seen that the standard model is insufficient to describe this battery. The adapted LiIon model

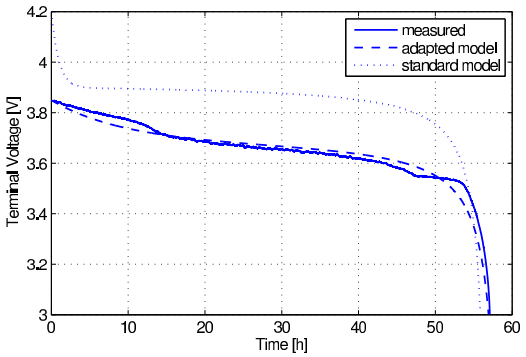


Fig. 6. Evaluation of the LiIon battery model.

uses modified parameters. The resulting maximum error is lower than 4 % and even lower than 1 % if the SoC is greater than 18 %. The battery is connected to a TPS63031 buck-boost converter supplying a mote with a constant input current of 31.9 mA. Therefore, the input current of the converter increases with a decreasing battery terminal voltage. This input current has been measured and it is applied to the simulation to be able to compare the results.

Furthermore, the measurement shows the need for adapting a standard battery model to a specific battery type. This enhances the accuracy of a simulation using such battery models.

V. Conclusion and Future Work

This work discussed the measurement error of the SoC of three different ESCs and the impact on the determination of the remaining operating time. It has been shown that the type of the ESC and the current SoC have a high influence on the measurement error. These results could be considered in applications, services and protocols.

Finally, a measurement has shown that the simulation error of the terminal voltage of a LiIon battery is lower than 4 % if the model uses modified parameters.

Future work will target the integration of this error analysis in WSN simulation environments. Furthermore, the charging process of the ESCs and the transitions between charging and discharging will be analyzed in detail.

References

[1] N. Watthanawisuth, A. Tuantranont, and T. Kerdcharoen, "Microclimate real-time monitoring based on zigbee sensor network," *Sensors, 2009 IEEE*, pp. 1814 –1818, oct. 2009.

[2] K. Langendoen, A. Baggio, and O. Visser, "Murphy loves potatoes: experiences from a pilot sensor network deployment in precision agriculture," *20th International Parallel and Distributed Processing Symposium (IPDPS 2006)*, p. 8 pp., apr. 2006.

[3] P. Juang, H. Oki, Y. Wang, M. Martonosi, L. S. Peh, and D. Rubenstein, "Energy-efficient computing for wildlife tracking: design tradeoffs and early experiences with zebranet," in *ASPLOS-X: Proceedings of the 10th int. conference on Architectural support for programming languages and operating systems*, 2002.

[4] A. Lindgren, C. Mascolo, M. Lonergan, and B. McConnell, "Seal-2-seal: A delay-tolerant protocol for contact logging in wildlife monitoring sensor networks," *5th IEEE International Conference on Mobile Ad Hoc and Sensor Systems (MASS 2008)*, pp. 321 –327, sep. 2008.

[5] K. Lorincz, B.-r. Chen, G. W. Challen, A. R. Chowdhury, S. Patel, P. Bonato, and M. Welsh, "Mercury: a wearable sensor network platform for high-fidelity motion analysis," in *Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems (SenSys 2009)*, 2009, pp. 183–196.

[6] N. Xu, S. Rangwala, K. K. Chintalapudi, D. Ganesan, A. Broad, R. Govindan, and D. Estrin, "A wireless sensor network for structural monitoring," in *Proceedings of the 2nd international conference on Embedded networked sensor systems (SenSys 2004)*, 2004, pp. 13–24.

[7] C. Enz, N. Scolari, and U. Yodprasit, "Ultra Low-Power Radio Design for Wireless Sensor Networks," in *IEEE Int. Workshop on Radio-Frequency Integration Technology*, 2005, pp. 1–17.

[8] M. Minami, T. Morito, H. Morikawa, and T. Aoyama, "Solar biscuit: A battery-less wireless sensor network system for environmental monitoring applications," *The 2nd International Workshop on Networked Sensing Systems*, 2005.

[9] M. Rahimi, H. Shah, G. Sukhatme, J. Heideman, and D. Estrin, "Studying the feasibility of energy harvesting in a mobile sensor network," *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA 2003)*, sep. 2003.

[10] V. Raghunathan, A. Kansal, J. Hsu, J. Friedman, and M. Srivastava, "Design considerations for solar energy harvesting wireless embedded systems," in *Proceedings of the 4th international symposium on Information processing in sensor networks*, 2005, p. 64.

[11] R. Min, M. Bhardwaj, S.-H. Cho, N. Ickes, E. Shih, A. Sinha, A. Wang, and A. Chandrakasan, "Energy-centric enabling technologies for wireless sensor networks," *IEEE Wireless Communications*, vol. 9, no. 4, pp. 28 – 39, aug. 2002.

[12] X. Jiang, J. Polastre, and D. Culler, "Perpetual environmentally powered sensor networks," *Fourth International Symposium on Information Processing in Sensor Networks (IPSN 2005)*, pp. 463 – 468, apr. 2005.

[13] L. B. Hörmann, P. M. Glatz, C. Steger, and R. Weiss, "A wireless sensor node for river monitoring using msp430 and energy harvesting," in *Proceedings of the European DSP in Education and Research Conference (EDERC 2010)*. Texas Instruments, 2010, pp. 140–144.

[14] P. M. Glatz, L. B. Hörmann, C. Steger, and R. Weiss, "Designing perpetual energy harvesting systems explained with rivermote: A wireless sensor network platform for river monitoring," *Electronic Journal of Structural Engineering, Special Issue: Sensor Network on Building Monitoring: from Theory to Real Application*, pp. 55–66, 2010.

[15] S. Yoon, R. Dutta, and M. Sichitiu, "Power aware routing algorithms for wireless sensor networks," in *Third International Conference on Wireless and Mobile Communications (ICWMC 2007)*, March 2007, pp. 15 –15.

[16] V. Pop, H. J. Bergveld, P. H. L. Notten, and P. P. L. Regtien, "State-of-the-art of battery state-of-charge determination," *Measurement Science and Technology*, vol. 16, p. R93, 2005.

[17] J. Polastre, R. Szewczyk, and D. Culler, "Telos: enabling ultra-low power wireless research," in *Fourth International Symposium on Information Processing in Sensor Networks (IPSN 2005)*, April 2005, pp. 364 – 369.

[18] Texas Instruments, "Tps63031 - high efficiency single inductor buck-boost converter with 1-a switches," www.focus-ti.com, SLVS696A, Oktober 2008.

[19] P. Dutta, M. Feldmeier, J. Paradiso, and D. Culler, "Energy metering for free: Augmenting switching regulators for real-time monitoring," in *International Conference on Information Processing in Sensor Networks (IPSN 2008)*, april 2008, pp. 283 –294.

[20] L. B. Hörmann, P. M. Glatz, C. Steger, and R. Weiss, "A systemc-ams simulation environment for the evaluation of energy harvesting wireless sensor networks," in *2011 International Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS 2011)*, june 2011, pp. 247 – 252.

2012 IEEE Wireless Communications and Networking Conference: Mobile and Wireless Networks

State-of-Charge Measurement Error Simulation for Power-Aware Wireless Sensor Networks

Leander B. Hörmann*, Philipp M. Glatz†, Karima B. Hein* and Reinhold Weiss*
*Institute for Technical Informatics, Graz University of Technology, Inffeldgasse 16, 8010 Graz, Austria
Email: {Leander.Hoermann@, Hein@, RWeiss@}TUGraz.at
†Embedded Software Group, Delft University of Technology, Mekelweg 4, 2628 CD Delft, The Netherlands
Email: P.M.Glatz@TUDelft.nl

***Abstract*—Wireless sensor networks (WSNs) are power critical systems, because they are used in application areas without wired infrastructure. Each sensor node needs a dedicated power supply. Today's protocols and applications for WSNs are often power-aware. However, the state-of-charge (SoC) estimation of the energy storage component (e.g. rechargeable battery) influences the decisions of the power-aware software. A measurement error may cause wrong decisions which would reduce the lifetime of the network. This work presents the simulation results of the impact of an SoC measurement error on the network lifetime running a simple and an enhanced power-aware WSN application. A simulation environment written in SystemC-AMS is used, which enables the combined simulation of the sensor node's application software and its hardware. The simulation shows a reduction of the network lifetime of 3.13% caused by an erroneous measurement. However, the enhanced software is able to increase the network lifetime by 1.66%.**

I. Introduction

Typically, wireless sensor networks (WSNs) are power critical systems, because they are used in application areas with mobility or no wired infrastructure. Examples are: precision agriculture [1], wildlife monitoring [2], human health care [3], and structural health monitoring [4]. The sensor nodes measure physical quantities and transmit the information about their environment through the wireless network in a multi-hop manner. Therefore, the connectivity of the network depends on the operability of single sensor nodes.

Each sensor node needs a dedicated energy source because of the lack of wired infrastructure. Basically, two different types of energy storage components (ESCs) exist. First, primary cells are cheap but cannot be recharged. Therefore, the operating time of the sensor node depends on the lifetime of the batteries. Due to the leakage current of each battery, the operating time cannot be extended in any order by reducing the average power consumption of the node [5]. Furthermore, the replacement of the batteries is very time-consuming and expensive, especially if the nodes are installed at hard-to-access locations. This is also known as battery replacement problem [6]. Therefore, these types of ESCs are not considered in further discussion.

Second, rechargeable ESCs and so called energy harvesting systems (EHSs) [7], [8] can be used. EHSs convert environmental energy into electrical energy and perpetual operation is possible. An ESC supplies the sensor node during periods with insufficient harvestable energy. There are many different types of rechargeable ESCs which can be used for that, e.g. Lithium-Ion (LiIon) rechargeable batteries, Nickel-Metal-Hydride (NiMH) rechargeable batteries, and double layer capacitors (DLCs).

As mentioned before, the connectivity of a WSN depends on the operability of single nodes along the route of the connection. To be aware of the future connectivity of the network, it is necessary to know the current operational state of each node. A key factor for estimating the remaining operating time is the state-of-charge (SoC) of the ESC. The information of the node's SoC is also used by power-aware routing algorithms [9] to relieve single nodes with low SoC of network traffic.

However, measurement errors can seriously influence SoC estimation. Furthermore, wrong SoC estimation impacts power-aware networking. Single sensor nodes with a wrong estimation of their SoC can run out of energy earlier than others. This reduces the lifetime of the WSN. For evaluating this impact, we use a modified version of the simulation environment we have introduced in [10]. The simulation environment simulates both software and hardware.

This work presents the simulation results of the impact of the SoC measurement error on a power-aware WSN application. Additionally, we show a modified power-aware application including a possible solution of the problem which extends the lifetime of the WSN.

The rest of the paper is organized as follows: Section II shows related work and Section III describes the simulation environment. Section IV introduces the evaluated network and the implemented power-aware routing algorithm and Section V shows the simulation results. Finally, Section VI concludes the paper and gives directions for future work.

II. Related Work

The SoC can be estimated in many different ways, as listed in [11]. Most accurate methods are using the information of the current or the charge running out of the ESC. These methods are difficult and complex hardware is required. One of the simplest method is to determine the open circuit voltage (OCV) of the ESC. The OCV can be estimated by measuring the terminal voltage of the ESC. This method is considered in further discussion and it is used for the SoC estimation.

978-1-4673-0437-5/12/$31.00 ©2012 IEEE

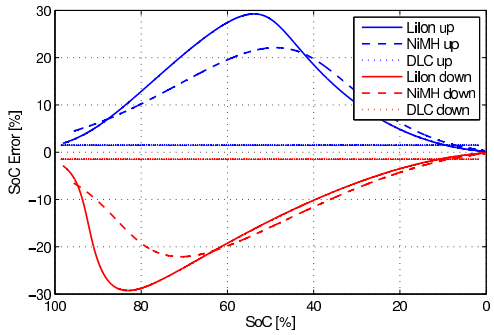


Fig. 1. Errors of the ESC voltage measurement dependent on the state-of-charge of three different ESCs. A measurement error of ±37.03 mV is assumed.

The error of the SoC estimation depends not only on the measurement error, but also on the current SoC and the type of the ESC. We have analyzed these dependencies in [12]. Fig. 1 shows the error of the SoC estimation of three different ESC types. The error upwards is important, because a too high estimated SoC may result in a higher work load of this sensor node and the battery runs out of energy earlier.

The simulation environment is written in SystemC-AMS which is used to simulate the whole WSN system including hardware and software.

The SystemC language can be used to model and simulate complex systems including hardware and software components. SystemC has already been used to simulate WSN systems [13]–[16]. The focus of these works is mainly on communication or power consumption of normal nodes (without EHS). The simulation of analog-mixed-signal (AMS) systems is not considered. SystemC-AMS is able to model such systems. A complex system can be composed out of single hardware components (e.g. resistor, capacitor) and higher level models. SystemC-AMS has also been used to model WSNs. However, the focus is mainly on the transceiver circuits and communication [17]–[19] and not on the power supply of nodes. The work presented in [20] describes a framework for testing different power-management approaches in WSNs implemented in SystemC. Energy harvesting technologies are also envisioned to be included into the simulation framework. However, the authors haven't presented any results and the power supply system is not modeled using the AMS extension of SystemC.

III. Simulation Environment

The simulation environment consists of two main parts. The first part is the environment model. It is responsible for the communication between the sensor nodes and for the location specific information (e.g. temperature). The second part is the sensor node model which itself is also divided into two parts describing the behavior and the power consumption of the sensor node. The sensor node model is a modified version of the sensor node we introduced and implemented in [21]. Each instance of the sensor node model is connected to the environment model. Fig. 2 shows the structure of the simulation environment.

A. Environment Model

The environment module connects the sensor nodes with each other and supplies them with location specific information. Therefore, the location of the sensor nodes is also stored in this module. The information of the local temperature is provided to the sensor nodes.

The communication between the sensor nodes is simulated by the connectivity and channel model. In this work, the communication is modeled at a high level (similar to unslotted IEEE 802.15.4 above MAC layer). The channel model provides a carrier sense signal to each sensor node. During a transmission, this signal is activated for all nodes in range of the transmission. The collision signal indicates a collision during the transmission of a packet. The transmitting signal models the temporal behavior of the sending of a packet. This means that the signal is activated when the transmission starts and it is deactivated at the end of the transmission. The channel model provides this signal as receiving signal to all sensor nodes in range of the sending node. The packet data is provided by by means of data signals.

B. Sensor Node Behavior Model

This model describes the behavior of the sensor node. The behavior is defined by the application software and the accessed hardware modules of the sensor node. These hardware modules (e.g. temperature sensor, transceiver) are modeled at a high level. This means that the module is described by a functional model. The hardware behind these modules is not described (except the current consumption). This enables a fast and quantitative simulation.

The software is simulated in the processing module. This is done in an event-driven way similar to TinyOS [22]. This means the application software of the sensor node provides predefined interfaces and uses itself interfaces from the processing module.

The other modules are behavioral descriptions of the respective hardware modules. The transceiver module simulates the transmission of the messages. The temperature sensor module simulates the measurement of the ambient temperature including possible measurement errors. The voltage measurement module simulates the determination of the battery's terminal voltage including possible measurement errors. The timer module can be used to get periodic events. Finally, the operation monitor module starts up the microcontroller and initializes the application software if the supply voltage is high enough and stops the execution if the supply voltage is too low.

C. Power Supply Model

The power supply model consists of the energy storage component model, the voltage converter model, and the load model. The energy storage component consists of a model

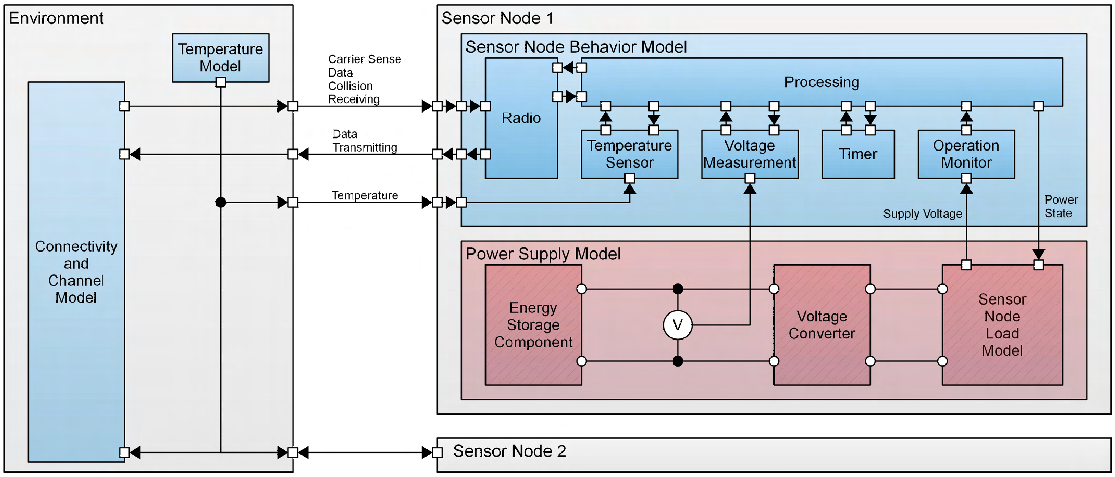


Fig. 2. SystemC-AMS structure of the simulation environment including the environmental model and one wireless sensor node in detail. The non-shaded modules indicate conventional SystemC modules and the shaded modules indicate SystemC-AMS modules. The square connectors show SystemC ports and the round connectors show electrical terminals.

TABLE I
MOST IMPORTANT POWER STATES OF A SENSOR NODE.

Power state	Supply current	Duration
Sleep (RX enabled)	28.34 mA	—
Message reception	30.51 mA	2.8 ms
Temperature measurement	30.54 mA	510 ms
Message transmission	158.8 mA	4.4 ms

of a LiIon rechargeable battery. The authors in [23] presented a generic battery model which we have implemented in SystemC-AMS. We have modified the parameters of the evaluated LiIon battery to reduce the capacity to 0.01 Ah. As the shape of the discharge characteristic does not change, the simulation time is reduced without changing the insight.

The voltage converter is described by a model of a linear dropout regulator with a constant output voltage of 3.3V. However, a detailed description of the EHS model would go beyond the scope of this paper. The load model is implemented using a variable resistor. The resistor changes its value depending on the power states provided by the processing module. The current consumption during each power state has been measured using the implemented sensor node as already mentioned using the measurement setup described in [24]. Different supply voltages have been considered to get a realistic model of the node.

The most important power states and their average current consumption and duration are shown in Table I.

IV. POWER-AWARE ALGORITHMS AND NETWORK STRUCTURE

The power-aware algorithm, the enhanced power-aware algorithm, the determination of the SoC, and the network structure including the region of interest are explained in the following subsections.

A. Power-Aware Algorithm

The power-aware algorithm is used to select the next sensor node to measure the temperature in the region of interest and to select the next hop of the packet's route. We assume that the temperature measurement of one node is sufficient in one interval for the whole region of interest. The temperature reading command is generated by the base station. It is sent to the node with highest SoC or to a node without any local SoC information. The message is routed to the next hop node. The next hop node is the neighboring node with the minimal hop count to the destination and the maximum SoC. The message is forwarded to the next hop node until the destination is reached. The destination node performs the temperature measurement and determines its SoC. Then, the measured temperature and the SoC are transmitted back to the base station. The next hop node is selected according to the already mentioned rules. When the message reaches the base station, the temperature is stored. Every time a sensor node receives a message, the SoC of the source node is stored locally. This is also done even if the receiving sensor node is not the destination.

B. Enhanced Power-Aware Algorithm

The enhanced algorithm is similar to the original one. The difference is at the selection of the next node and the next hop node. In the original algorithm, the node with the highest SoC is selected. Here, the two nodes with the highest SoC are preselected. Then, one of the two nodes is selected randomly. This causes a better distribution of the workload.

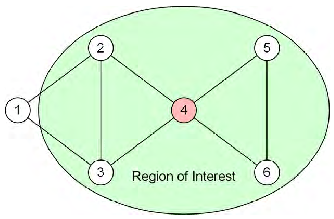


Fig. 3. Structure of the simulated network. Node 1 is the base station and node 4 is the critical node.

C. SoC Determination

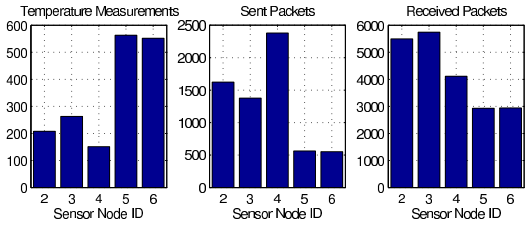
The SoC of the sensor node's ESC is determined with the measured terminal voltage of the ESC. This dependency is stored locally as a lookup table. It would be possible to use an analytic function including an exponential function. However, a sensor node is a resource-constrained device and an analytic function is typically more time and space consuming (depending on the implementation) compared to a lookup table. Another advantage of the lookup table is that it is possible to adapt it easily on the measured discharge trace of a real rechargeable battery. The SoC is calculated interpolating the nearest two points of the lookup table. We assume that the ESC of each sensor node has the same maximum charge voltage. Therefore, no adjustment of the lookup table is necessary. In practice, the maximum charge voltage may vary and an adjustment of the lookup table would increase its accuracy. However, this is omitted here, because it would not enhance the insight into the simulation.

D. Network Structure

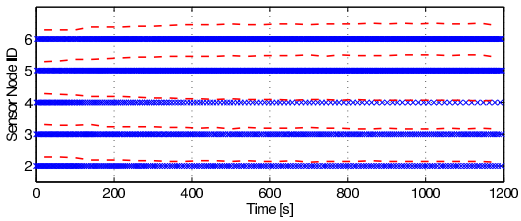
The simulated network consists of five sensor nodes and one base station. The five sensor nodes are placed in the region of interest. The structure of the network is shown in Fig. 3. Node 1 represents the base station. The critical node is node 4. All traffic from the base station to nodes 5 and 6 has to be forwarded from node 4. Node 4 is also the node which is considered to have a measurement error in the respective simulation. Furthermore, the error occurs after a certain period of time. At the beginning of the simulation it is not present. Therefore, no adjustment to the maximum charge voltage is possible. Despite the simple structure, the SoC measurement error's impact can be observed.

V. Simulation Results

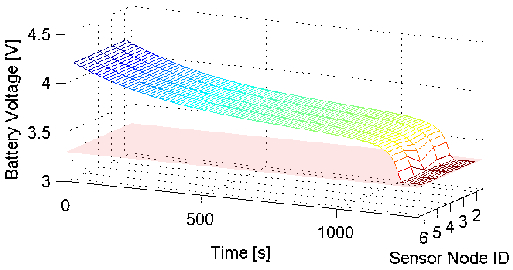
This section shows the simulation results of the WSN. The temporal resolution of the analog circuit simulation is $100\,\mu$s. Fig. 4 shows the results of the network simulation using the normal routing protocol. All sensor nodes estimate the SoC correctly. The statistics of the network traffic and the temperature measurements are illustrated in Fig. 4(a). It can be seen that the rear sensor nodes (5 and 6) perform most temperature measurements. They need not forward any messages and more energy is available. Node 4 has to forward most of the messages. Therefore, it performs the fewest



(a) Network traffic and temperature measurement statistics.



(b) Temperature measurement distribution.

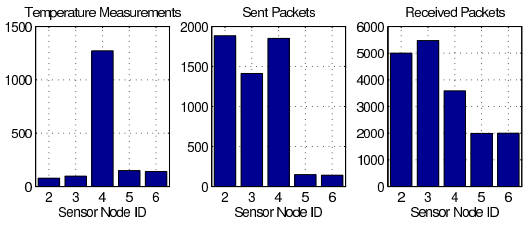


(c) Voltage trace of the node's ESC.

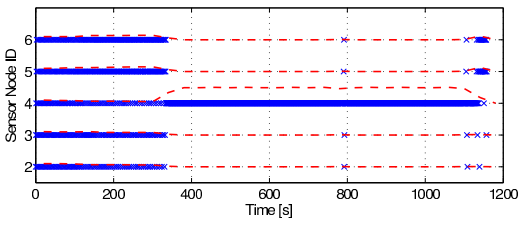
Fig. 4. Simulation results of the five-nodes-network using the normal routing protocol. All nodes are working properly.

measurements. Fig. 4(b) shows the chronological sequence of the temperature measurements. The dashed line shows the moving average of the temperature measurement rate. The traces of the ESC voltages are plotted in Fig. 4(c). They show the typical discharge characteristic of a LiIon rechargeable battery. It can be seen that the voltage of node 4 drops a little bit faster. The reason is the lack of information at the base station about the real SoC of node 4: The SoC information of a specific node at the base station is only updated after a temperature measurement command for that node. After a certain period of time, the information of node 4 is not up to date and it wrongly indicates the highest SoC of all nodes.

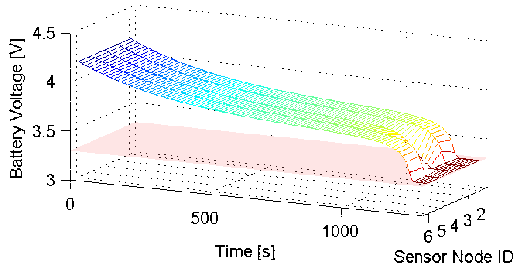
Fig. 5 shows the results of the network simulation using the normal routing protocol. The measurement of the ESC terminal voltage of node 4 becomes erroneous after 350 s. Then, the measurement error is $+4\,\%$. This causes a maximum SoC estimation error of $45\,\%$. The statistics of the network traffic and the temperature measurements are illustrated in Fig. 5(a). It can be seen that node 4 performs most temperature



(a) Network traffic and temperature measurement statistics.

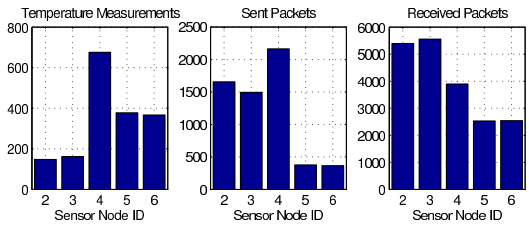


(b) Temperature measurement distribution.

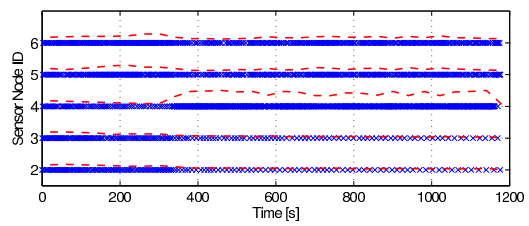


(c) Voltage trace of the node's ESC.

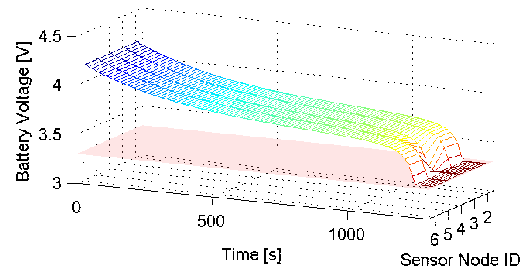
Fig. 5. Simulation results of the five-nodes-network using the normal routing protocol. An SoC estimation error occurs at node number 4.



(a) Network traffic and temperature measurement statistics.



(b) Temperature measurement distribution.



(c) Voltage trace of the node's ESC.

Fig. 6. Simulation results of the five-nodes-network using the enhanced routing protocol. An SoC estimation error occurs at node number 4.

measurements. The reason is the wrong estimation of the SoC. This can be seen clearly in the chronological sequence of the temperature measurements shown in Fig. 5(b). Almost all temperature measurements are performed by node 4 after 350 s. Thus, the real SoC of node 4 decreases faster than the SoC of the other nodes. This can be seen in Fig. 5(c). The increased consumed energy is lacking at the end of the simulation. Then, node 5 and node 6 have energy reserves but node 4 fails and it cannot forward messages. The simulation statistics including the network lifetime is shown in Table II.

Fig. 6 shows the results of the network simulation using the enhanced routing protocol. The measurement of the ESC terminal voltage of node 4 becomes also erroneous after 350 s here. The statistics of the network traffic and the temperature measurements are illustrated in Fig. 6(a). It can be seen that node 4 performs most temperature measurements. However, the difference between node 4 and the other nodes is smaller than in the previous simulation. The reason is the random selection of the next node from the two nodes with the highest SoC. This can be seen clearly in the chronological sequence of the temperature measurements shown in Fig. 6(b). Despite the measurement error of node 4, temperature measurement commands are also sent to the other nodes. The voltage of node's 4 ESC does not decrease as fast as in the previous simulation. This can be seen in Fig. 6(c). This is why the lifetime of the network is enhanced.

Table II lists the statistics of all four performed simulations. The simulation setting without any measurement error using the enhanced routing is listed for completeness. The shortest network lifetime is caused by the simulation setting with the faulty node and the normal routing. This is a reduction of 3.13 % related to the error-free simulation setting. The enhanced routing causes an increase of 1.66 % related to the reduced network lifetime. The number of the temperature measurements is interesting, because it is equal with and without the faulty node. The reason is that the base station transmits the next temperature measurement command immediately after receiving a measured temperature. The measurement at the

TABLE II
SUMMARY OF SIMULATION RESULTS.

Simulation settings	First failing node	Network lifetime	Number of temperature measurements	Measurement rate
All nodes OK normal routing	4	1195.10 s	1737	1.453 $\frac{1}{s}$
Node 4 faulty normal routing	4	1157.75 s	1737	1.500 $\frac{1}{s}$
All nodes OK enhanced routing	4	1195.31 s	1736	1.452 $\frac{1}{s}$
Node 4 faulty enhanced routing	4	1176.96 s	1731	1.471 $\frac{1}{s}$

rear nodes needs more time, because the messages must be forwarded more often. Due to the fact that most temperature measurements are performed by node 4, the number of temperature measurements does not change despite the reduced network lifetime. This can be seen in the last column of the table showing the average temperature measurement rate.

VI. Conclusion and Future Work

This work presented the simulation results of the impact of the SoC measurement error on a power-aware WSN application. We used a simulation environment written in SystemC-AMS which enables a combined simulation of hardware and software of a wireless sensor node.

Using a simple power-aware routing protocol showed the reduction of the network lifetime by 3.13 % simulating an erroneous SoC estimation at one node. A simple enhancement of the routing increased the network lifetime by 1.66 %. Furthermore, we showed that a simulation environment supporting a combined simulation of software and hardware is necessary to optimize power-aware algorithms.

The environment allows the integration of various relevant hardware characteristics, paving the way for the evaluation of different power-aware algorithms.

References

[1] K. Langendoen, A. Baggio, and O. Visser, "Murphy loves potatoes: experiences from a pilot sensor network deployment in precision agriculture," *20th International Parallel and Distributed Processing Symposium (IPDPS 2006)*, p. 8 pp., 2006.

[2] A. Lindgren, C. Mascolo, M. Lonergan, and B. McConnell, "Seal-2-seal: A delay-tolerant protocol for contact logging in wildlife monitoring sensor networks," *5th IEEE International Conference on Mobile Ad Hoc and Sensor Systems (MASS 2008)*, pp. 321–327, 2008.

[3] K. Lorincz, B.-r. Chen, G. W. Challen, A. R. Chowdhury, S. Patel, P. Bonato, and M. Welsh, "Mercury: a wearable sensor network platform for high-fidelity motion analysis," in *Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems (SenSys 2009)*, 2009, pp. 183–196.

[4] N. Xu, S. Rangwala, K. K. Chintalapudi, D. Ganesan, A. Broad, R. Govindan, and D. Estrin, "A wireless sensor network for structural monitoring," in *Proceedings of the 2nd international conference on Embedded networked sensor systems (SenSys 2004)*, 2004, pp. 13–24.

[5] C. Enz, N. Scolari, and U. Yodprasit, "Ultra Low-Power Radio Design for Wireless Sensor Networks," in *IEEE Int. Workshop on Radio-Frequency Integration Technology*, 2005, pp. 1–17.

[6] M. Minami, T. Morito, H. Morikawa, and T. Aoyama, "Solar biscuit: A battery-less wireless sensor network system for environmental monitoring applications," *The 2nd International Workshop on Networked Sensing Systems*, 2005.

[7] M. Rahimi, H. Shah, G. Sukhatme, J. Heideman, and D. Estrin, "Studying the feasibility of energy harvesting in a mobile sensor network," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA 2003)*, vol. 1, sept. 2003, pp. 19–24.

[8] V. Raghunathan, A. Kansal, J. Hsu, J. Friedman, and M. Srivastava, "Design considerations for solar energy harvesting wireless embedded systems," in *Proceedings of the 4th international symposium on Information processing in sensor networks (IPSN 2005)*, 2005, pp. 457–462.

[9] S. Yoon, R. Dutta, and M. Sichitiu, "Power aware routing algorithms for wireless sensor networks," in *Third International Conference on Wireless and Mobile Communications (ICWMC 2007)*, 2007, p. 15 pp.

[10] L. B. Hörmann, P. M. Glatz, C. Steger, and R. Weiss, "A systemc-ams simulation environment for the evaluation of energy harvesting wireless sensor networks," in *International Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS 2011)*, 2011, pp. 247–252.

[11] V. Pop, H. J. Bergveld, P. H. L. Notten, and P. P. L. Regtien, "State-of-the-art of battery state-of-charge determination," *Measurement Science and Technology*, vol. 16, p. R93, 2005.

[12] L. B. Hörmann, P. M. Glatz, C. Steger, and R. Weiss, "Measuring the state-of-charge - analysis and impact on wireless sensor networks," in *36thIEEE Conference on Local Computer Networks (LCN 2011)*, 2011, pp. 982–985.

[13] J. Haase, M. Damm, J. Glaser, J. Moreno, and C. Grimm, "Systemc-based power simulation of wireless sensor networks," in *Forum on Specification Design Languages (FDL 2009)*, sept. 2009, pp. 1–4.

[14] W. Du, D. Navarro, and F. Mieyeville, "A simulation study of ieee 802.15.4 sensor networks in industrial applications by system-level modeling," in *Fourth International Conference on Sensor Technologies and Applications (SENSORCOMM 2010)*, 2010, pp. 311–316.

[15] D. Navarro, W. Du, F. Mieyeville, and L. Carrel, "Hardware and software system-level simulator for wireless sensor networks," *Procedia Engineering*, vol. 5, pp. 228–231, 2010, eurosensor XXIV Conference.

[16] W. Du, F. Mieyeville, and D. Navarro, "Modeling energy consumption of wireless sensor networks by systemc," in *Fifth International Conference on Systems and Networks Communications (ICSNC 2010)*, aug. 2010, pp. 94–98.

[17] M. Vasilevski, N. Beilleau, H. Aboushady, and F. Pecheux, "Efficient and refined modeling of wireless sensor network nodes using systemc-ams," in *Ph.D. Research in Microelectronics and Electronics (PRIME 2008)*, 2008, pp. 81–84.

[18] M. Vasilevski, F. Pecheux, N. Beilleau, H. Aboushady, and K. Einwich, "Modeling and refining heterogeneous systems with systemc-ams: Application to wsn," in *Design, Automation and Test in Europe (DATE 2008)*, 2008, pp. 134–139.

[19] M. Vasilevski, H. Aboushady, F. Pecheux, and L. de Lamarre, "Modeling wireless sensor network nodes using systemc-ams," in *Internatonal Conference on Microelectronics (ICM 2007)*, 2007, pp. 53–56.

[20] R. Thomasius and S. Guttowski, "A systemc based framework for the evaluation of proactive power-management approaches in distributed energy harvesting systems," in *Fourth International Conference on Sensor Technologies and Applications (SENSORCOMM 2010)*, 2010, pp. 348–351.

[21] L. Hörmann, P. Glatz, C. Steger, and R. Weiss, "Evaluation of component-aware dynamic voltage scaling for mobile devices and wireless sensor networks," in *IEEE International Symposium on a World of Wireless Mobile and Multimedia Networks (WoWMoM 2011)*, june 2011, pp. 1–9.

[22] P. Levis, S. Madden, J. Polastre, R. Szewczyk, K. Whitehouse, A. Woo, D. Gay, J. Hill, M. Welsh, E. Brewer, and D. Culler, "Tinyos: An operating system for sensor networks," in *Ambient Intelligence*, W. Weber, J. M. Rabaey, and E. Aarts, Eds. Springer Berlin Heidelberg, 2005, pp. 115–148.

[23] O. Tremblay, L.-A. Dessaint, and A.-I. Dekkiche, "A generic battery model for the dynamic simulation of hybrid electric vehicles," in *IEEE Vehicle Power and Propulsion Conference*, 2007, pp. 284–289.

[24] P. Glatz, L. B. Hörmann, C. Steger, and R. Weiss, "A system for accurate characterization of wireless sensor networks with power states and energy harvesting system efficiency," *8th IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM 2010)*, pp. 468–473, 2010.

Bibliography

[1] L. B. Hörmann, P. M. Glatz, C. Steger, and R. Weiss, "A SystemC-AMS Simulation Environment for the Evaluation of Energy Harvesting Wireless Sensor Networks," in *International Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS 2011)*, 2011, pp. 247–252.

[2] L. B. Hörmann, P. M. Glatz, K. B. Hein, C. Steger, and R. Weiss, "A Hardware/Software Simulation Environment for Energy Harvesting Wireless Sensor Networks," in *Proceedings of the 9th ACM symposium on Performance evaluation of wireless ad hoc, sensor, and ubiquitous networks (PE-WASUN 2012)*, 2012, pp. 61–68.

[3] L. B. Hörmann, P. M. Glatz, C. Steger, and R. Weiss, "Designing of Efficient Energy Harvesting Systems for Autonomous WSNs Using a Tier Model," in *18th International Conference on Telecommunications (ICT 2011)*, 2011, pp. 185–190.

[4] L. B. Hörmann, P. M. Glatz, K. B. Hein, M. Steinberger, C. Steger, and R. Weiss, "Towards an On-Site Characterization of Energy Harvesting Devices for Wireless Sensor Networks," in *IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops 2012)*, 2012, pp. 415–418.

[5] L. B. Hörmann, P. M. Glatz, C. Steger, and R. Weiss, "Energy Efficient Supply of WSN nodes using Component-Aware Dynamic Voltage Scaling," in *European Wireless Conference (EW 2011)*. Berlin u. Offenbach, Germany: VDE VERLAG GMBH, 2011, pp. 147–154.

[6] L. B. Hörmann, P. M. Glatz, C. Steger, and R. Weiss, "Evaluation of Component-Aware Dynamic Voltage Scaling for Mobile Devices and Wireless Sensor Networks," in *IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM 2011)*, 2011, pp. 1–9.

[7] L. B. Hörmann, P. M. Glatz, C. Steger, and R. Weiss, "Measuring the State-of-Charge - Analysis and Impact on Wireless Sensor Networks," in *IEEE International Workshop on Issues in Building Sensor Network Applications (SenseApp 2011)*, 2011, pp. 982–985.

[8] L. B. Hörmann, P. M. Glatz, K. B. Hein, and R. Weiss., "State-of-Charge Measurement Error Simulation for Power-Aware Wireless Sensor Networks," in *IEEE Wireless Communications and Networking Conference (WCNC 2012)*, 2012, pp. 2209–2214.

[9] G. Barrenetxea, F. Ingelrest, G. Schaefer, M. Vetterli, O. Couach, and M. Parlange, "SensorScope: Out-of-the-Box Environmental Monitoring," in *Proceedings of the In-*

ternational Conference on Information Processing in Sensor Networks (IPSN 2008), 2008, pp. 332–343.

[10] K. Langendoen, A. Baggio, and O. Visser, "Murphy Loves Potatoes: Experiences from a Pilot Sensor Network Deployment in Precision Agriculture," *20th International Parallel and Distributed Processing Symposium (IPDPS 2006)*, p. 8 pp., 2006.

[11] N. Watthanawisuth, A. Tuantranont, and T. Kerdcharoen, "Microclimate real-time monitoring based on ZigBee sensor network," *Sensors, 2009 IEEE*, pp. 1814–1818, 2009.

[12] P. Zhang, C. M. Sadler, S. A. Lyon, and M. Martonosi, "Hardware Design Experiences in ZebraNet," in *Proceedings of the 2nd international conference on Embedded networked sensor systems*, ser. SenSys '04, 2004, pp. 227–238.

[13] A. Lindgren, C. Mascolo, M. Lonergan, and B. McConnell, "Seal-2-Seal: A Delay-Tolerant Protocol for Contact Logging in Wildlife Monitoring Sensor Networks," *5th IEEE International Conference on Mobile Ad Hoc and Sensor Systems (MASS 2008)*, pp. 321–327, 2008.

[14] K. Lorincz, B.-r. Chen, G. W. Challen, A. R. Chowdhury, S. Patel, P. Bonato, and M. Welsh, "Mercury: A Wearable Sensor Network Platform for High-Fidelity Motion Analysis," in *Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems (SenSys 2009)*, 2009, pp. 183–196.

[15] J. Corchado, J. Bajo, D. Tapia, and A. Abraham, "Using Heterogeneous Wireless Sensor Networks in a Telemonitoring System for Healthcare," *Information Technology in Biomedicine, IEEE Transactions on*, vol. 14, no. 2, pp. 234–240, 2010.

[16] N. Xu, S. Rangwala, K. K. Chintalapudi, D. Ganesan, A. Broad, R. Govindan, and D. Estrin, "A Wireless Sensor Network for Structural Monitoring," in *Proceedings of the 2nd international conference on Embedded networked sensor systems (SenSys 2004)*, 2004, pp. 13–24.

[17] M. Ceriotti, L. Mottola, G. P. Picco, A. L. Murphy, S. Guna, M. Corra, M. Pozzi, D. Zonta, and P. Zanon, "Monitoring Heritage Buildings with Wireless Sensor Networks: The Torre Aquila Deployment," in *Proceedings of the 2009 International Conference on Information Processing in Sensor Networks*, ser. IPSN '09, 2009, pp. 277–288.

[18] J. Wu, S. Yuan, S. Ji, G. Zhou, Y. Wang, and Z. Wang, "Multi-agent system design and evaluation for collaborative wireless sensor network in large structure health monitoring," *Expert Systems with Applications*, vol. 37, no. 3, pp. 2028–2036, 2010.

[19] F. Osterlind, E. Pramsten, D. Roberthson, J. Eriksson, N. Finne, and T. Voigt, "Integrating Building Automation Systems and Wireless Sensor Networks," in *Emerging Technologies and Factory Automation, 2007. ETFA. IEEE Conference on*, 2007, pp. 1376–1379.

[20] M. Ceriotti, M. Corra, L. d'Orazio, R. Doriguzzi, D. Facchin, S. Guna, G. Jesi, R. Cigno, L. Mottola, A. Murphy, M. Pescalli, G. Picco, D. Pregnolato, and C. Torghele, "Is There Light at the Ends of the Tunnel? Wireless Sensor Networks for Adaptive Lighting in Road Tunnels," in *10th International Conference on Information Processing in Sensor Networks (IPSN 2011)*, 2011, pp. 187–198.

[21] M. Maleki, K. Dantu, and M. Pedram, "Lifetime Prediction Routing in Mobile Ad Hoc Networks," in *Wireless Communications and Networking, 2003. WCNC 2003. 2003 IEEE*, vol. 2, 2003, pp. 1185–1190.

[22] C. Enz, N. Scolari, and U. Yodprasit, "Ultra Low-Power Radio Design for Wireless Sensor Networks," in *IEEE Int. Workshop on Radio-Frequency Integration Technology*, 2005, pp. 1–17.

[23] K. Römer and F. Mattern, "The Design Space of Wireless Sensor Networks," *IEEE Wireless Communications*, vol. 11, no. 6, pp. 54–61, 2004.

[24] M. Afzal, W. Mahmood, and A. Akbar, "A Battery Recharge Model for WSNs using Free-Space Optics (FSO)," in *IEEE International Multitopic Conference (INMIC 2008)*, 2008, pp. 272–277.

[25] M. Minami, T. Morito, H. Morikawa, and T. Aoyama, "Solar Biscuit: A Battery-Less Wireless Sensor Network System for Environmental Monitoring Applications," *The 2nd International Workshop on Networked Sensing Systems*, 2005.

[26] S. Roundy, D. Steingart, L. Frechette, P. Wright, and J. Rabaey, "Power Sources for Wireless Sensor Networks," in *Wireless Sensor Networks*, ser. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 2004, vol. 2920, pp. 1–17.

[27] V. Raghunathan, A. Kansal, J. Hsu, J. Friedman, and M. Srivastava, "Design Considerations for Solar Energy Harvesting Wireless Embedded Systems," in *Proceedings of the 4th international symposium on Information processing in sensor networks (IPSN 2005)*, 2005, pp. 457–462.

[28] P. He, Q. Cui, and X. Guo, "Efficient solar power scavenging and utilization in mobile electronics system," *International Conference on Green Circuits and Systems (ICGCS 2010)*, pp. 641–645, 2010.

[29] A. Janek, C. Trummer, C. Steger, R. Weiss, J. Preishuber-Pfluegl, and M. Pistauer, "Simulation based Verification of Energy Storage Architectures for Higher Class Tags Supported by Energy Harvesting Devices," *Microprocessors and Microsystems*, vol. 32, no. 5-6, pp. 330–339, 2008.

[30] X. Jiang, J. Polastre, and D. Culler, "Perpetual Environmentally Powered Sensor Networks," *Fourth International Symposium on Information Processing in Sensor Networks (IPSN 2005)*, pp. 463–468, 2005.

[31] G. Chen, H. Ghaed, R. Haque, M. Wieckowski, Y. Kim, G. Kim, D. Fick, D. Kim, M. Seok, K. Wise, D. Blaauw, and D. Sylvester, "A Cubic-Millimeter Energy-Autonomous Wireless Intraocular Pressure Monitor," in *IEEE International Solid-*

State Circuits Conference Digest of Technical Papers (ISSCC 2011), 2011, pp. 310–312.

[32] S. Singh, M. Woo, and C. S. Raghavendra, "Power-Aware Routing in Mobile Ad Hoc Networks," in *Proceedings of the 4th annual ACM/IEEE international conference on Mobile computing and networking*, ser. MobiCom '98, 1998, pp. 181–190.

[33] J. Al-Karaki and A. Kamal, "Routing Techniques in Wireless Sensor Networks: A Survey," *IEEE Wireless Communications*, vol. 11, no. 6, pp. 6–28, 2004.

[34] J. Heo, J. Hong, and Y. Cho, "EARQ: Energy Aware Routing for Real-Time and Reliable Communication in Wireless Industrial Sensor Networks," *Industrial Informatics, IEEE Transactions on*, vol. 5, no. 1, pp. 3–11, 2009.

[35] L. Mottola and G. Picco, "MUSTER: Adaptive Energy-Aware Multisink Routing in Wireless Sensor Networks," *Mobile Computing, IEEE Transactions on*, vol. 10, no. 12, pp. 1694–1709, 2011.

[36] J. Li and P. Mohapatra, "Analytical modeling and mitigation techniques for the energy hole problem in sensor networks," *Pervasive and Mobile Computing*, vol. 3, no. 3, pp. 233–254, 2007.

[37] X. Wu, G. Chen, and S. Das, "Avoiding Energy Holes in Wireless Sensor Networks with Nonuniform Node Distribution," *IEEE Transactions on Parallel and Distributed Systems*, vol. 19, no. 5, pp. 710–720, 2008.

[38] A. Kansal, D. Potter, and M. B. Srivastava, "Performance Aware Tasking for Environmentally Powered Sensor Networks," in *Proceedings of the joint international conference on Measurement and modeling of computer systems*, ser. SIGMETRICS '04/Performance '04, 2004, pp. 223–234.

[39] C.-C. Shen, W. L. Plishker, D.-I. Ko, S. S. Bhattacharyya, and N. Goldsman, "Energy-driven Distribution of Signal Processing Applications across Wireless Sensor Networks," *ACM Trans. Sen. Netw.*, vol. 6, no. 3, pp. 24:1–24:32, 2010.

[40] L. B. Hörmann, "Design and Implementation of a Wireless Sensor Platform for River Monitoring Based on Energy Harvesting," Master's thesis, Graz University of Technology, Institute for Technical Informatics, 2010.

[41] L. Hörmann, P. Glatz, C. Steger, and R. Weiss, "A wireless sensor node for river monitoring using MSP430 and energy harvesting," in *4th European Education and Research Conference (EDERC 2010)*, 2010, pp. 140–144.

[42] P. M. Glatz, L. B. Hörmann, C. Steger, and R. Weiss, "Designing Perpetual Energy Harvesting Systems explained with RiverMote: A Wireless Sensor Network Platform for River Monitoring," *Electronic Journal of Structural Engineering, Special Issue: Sensor Network on Building Monitoring: from Theory to Real Application*, pp. 55–66, 2010.

[43] P. M. Glatz, "Efficient Network Coding Middleware for Energy Harvesting Wireless Sensor Networks," Ph.D. dissertation, Graz University of Technology, Institute for Technical Informatics, 2011.

[44] B. P. Zeigler, H. Praehofer, and T. G. Kim, *Theory of Modeling and Simulation - Integrating Discrete Event and Continuous Complex Dynamic Systems*, 2nd ed. San Diego, CA: Academic Press, 2000.

[45] P. Wellstead, E. Bullinger, D. Kalamatianos, O. Mason, and M. Verwoerd, "The role of control and system theory in systems biology," *Annual Reviews in Control*, vol. 32, no. 1, pp. 33–47, 2008.

[46] D. Xu and W. Wang, "Research on Applications of Linear System Theory in Economics," in *Control and Automation (ICCA), 2010 8th IEEE International Conference on*, june 2010, pp. 1843–1847.

[47] H. Millan, A. Kalauzi, G. Llerena, J. Sucoshanay, and D. Piedra, "Meteorological complexity in the Amazonian area of Ecuador: An approach based on dynamical system theory," *Ecological Complexity*, vol. 6, no. 3, pp. 278–285, 2009.

[48] G. Morganti, A. Perdon, G. Conte, and D. Scaradozzi, "Multi-Agent System Theory for Modelling a Home Automation System," in *Bio-Inspired Systems: Computational and Ambient Intelligence*, ser. Lecture Notes in Computer Science, J. Cabestany, F. Sandoval, A. Prieto, and J. Corchado, Eds. Springer Berlin Heidelberg, 2009, vol. 5517, pp. 585–593.

[49] J. Huang, Y. Feng, and S. Zhang, "Research of Complex System Theory Application on Reliability Analysis of Network System," in *Reliability, Maintainability and Safety, 2009. ICRMS 2009. 8th International Conference on*, july 2009, pp. 1141–1145.

[50] J. Zhao, "Road Traffic Safety Evaluation Index System Based on Complex System-Entropy Theory," in *Electric Technology and Civil Engineering (ICETCE), 2011 International Conference on*, april 2011, pp. 1521–1524.

[51] J. L. Hill, "System Architecture for Wireless Sensor Networks," Ph.D. dissertation, University of California at Berkeley, 2003.

[52] M. Ilyas and I. Mahgoub, Eds., *Handbook of Sensor Networks: Compact Wireless and Wired Sensing Systems*. CRC Press LLC, 2005.

[53] T. Liu and M. Martonosi, "Impala: A Middleware System for Managing Autonomic, Parallel Sensor Systems," in *Proceedings of the ninth ACM SIGPLAN symposium on Principles and practice of parallel programming*, ser. PPoPP '03, 2003, pp. 107–118.

[54] E. Egea-Lopez, J. Vales-Alonso, A. S. Martinez-Sala, P. Pavon-Mario, and J. Garcia-Haro, "Simulation Tools for Wireless Sensor Networks," in *Proceedings of the International Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS 2005)*, 2005, pp. 559–566.

[55] X. Xian, W. Shi, and H. Huang, "Comparison of OMNET++ and Other Simulator for WSN Simulation," in *3rd IEEE Conference on Industrial Electronics and Applications (ICIEA 2008)*, june 2008, pp. 1439–1443.

[56] M. Jevtic, N. Zogovic, and G. Dimic, "Evaluation of Wireless Sensor Network Simulators," in *Proceedings of the 17th Telecommunications Forum (TELFOR 2009)*, 2009.

[57] E. Weingartner, H. vom Lehn, and K. Wehrle, "A Performance Comparison of Recent Network Simulators," in *IEEE International Conference on Communications (ICC 2009)*, 2009, pp. 1–5.

[58] M. Korkalainen, M. Sallinen, N. Karkkainen, and P. Tukeva, "Survey of Wireless Sensor Networks Simulation Tools for Demanding Applications," in *Fifth International Conference on Networking and Services (ICNS 2009)*, 2009, pp. 102–106.

[59] M. Imran, A. Said, and H. Hasbullah, "A Survey of Simulators, Emulators and Testbeds for Wireless Sensor Networks," in *International Symposium in Information Technology (ITSim 2010)*, vol. 2, 2010, pp. 897–902.

[60] W. Du, D. Navarro, F. Mieyeville, and F. Gaffiot, "Towards a Taxonomy of Simulation Tools for Wireless Sensor Networks," in *Proceedings of the 3rd International ICST Conference on Simulation Tools and Techniques*, ser. SIMUTools '10. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2010, pp. 52:1–52:7.

[61] M. Khan, B. Askwith, F. Bouhafs, and M. Asim, "Limitations of Simulation Tools for Large-Scale Wireless Sensor Networks," in *Advanced Information Networking and Applications (WAINA), 2011 IEEE Workshops of International Conference on*, 2011, pp. 820–825.

[62] H. Sundani, H. Li, V. K. Devabhaktuni, M. Alam, and P. Bhattacharya, "Wireless Sensor Network Simulators: A Survey and Comparisons," *International Journal of Computer Networks (IJCN)*, vol. 2, pp. 249–265, 2011.

[63] D. Jung, T. Teixeira, and A. Savvides, "Sensor Node Lifetime Analysis: Models and Tools," *ACM Trans. Sen. Netw.*, vol. 5, no. 1, pp. 3:1–3:33, 2009.

[64] V. Shnayder, M. Hempstead, B.-r. Chen, G. W. Allen, and M. Welsh, "Simulating the Power Consumption of LargeScale Sensor Network Applications," in *Proceedings of the 2nd international conference on Embedded networked sensor systems*, ser. SenSys '04, 2004, pp. 188–200.

[65] P. Glatz, L. B. Hörmann, C. Steger, and R. Weiss, "A System for Accurate Characterization of Wireless Sensor Networks with Power States and Energy Harvesting System Efficiency," *8th IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops 2010)*, pp. 468–473, 2010.

[66] J. Yick, B. Mukherjee, and D. Ghosal, "Wireless sensor network survey," *Computer Networks*, vol. 52, no. 12, pp. 2292–2330, 2008.

[67] C. Park and P. Chou, "AmbiMax: Autonomous Energy Harvesting Platform for Multi-Supply Wireless Sensor Nodes," in *3rd Annual IEEE Communications Society on Sensor and Ad Hoc Communications and Networks (SECON 2006)*, vol. 1, 2006, pp. 168–177.

[68] M. Wischke, M. Masur, M. Kroener, and P. Woias, "Vibration harvesting in traffic tunnels to power wireless sensor nodes," *Smart Materials and Structures*, vol. 20, no. 8, pp. 1–8, 2011.

[69] X. Zhao, T. Keutel, M. Baldauf, and O. Kanoun, "Energy harvesting for overhead power line monitoring," in *Systems, Signals and Devices (SSD), 2012 9th International Multi-Conference on*, 2012, pp. 1–5.

[70] C. Alippi, R. Camplani, C. Galperti, and M. Roveri, "A Robust, Adaptive, Solar-Powered WSN Framework for Aquatic Environmental Monitoring," *Sensors Journal, IEEE*, vol. 11, no. 1, pp. 45–55, 2011.

[71] S. Gandham, M. Dawande, R. Prakash, and S. Venkatesan, "Energy Efficient Schemes for Wireless Sensor Networks with Multiple Mobile Base Stations," in *IEEE Global Telecommunications Conference (GLOBECOM 2003)*, vol. 1, 2003, pp. 377–381.

[72] Z. Vincze, R. Vida, and A. Vidacs, "Deploying Multiple Sinks in Multi-hop Wireless Sensor Networks," in *IEEE International Conference on Pervasive Services*, 2007, pp. 55–63.

[73] V. Shah-Mansouri, A.-H. Mohsenian-Rad, and V. Wong, "Lexicographically Optimal Routing for Wireless Sensor Networks With Multiple Sinks," *IEEE Transactions on Vehicular Technology*, vol. 58, no. 3, pp. 1490–1500, 2009.

[74] A. Shrestha and L. Xing, "A Performance Comparison of Different Topologies for Wireless Sensor Networks," in *IEEE Conference on Technologies for Homeland Security*, 2007, pp. 280–285.

[75] A. A. Abbasi and M. Younis, "A survey on clustering algorithms for wireless sensor networks," *Computer Communications*, vol. 30, no. 1415, pp. 2826–2841, 2007.

[76] D. Puccinelli and M. Haenggi, "Wireless Sensor Networks: Applications and Challenges of Ubiquitous Sensing," *Circuits and Systems Magazine, IEEE*, vol. 5, no. 3, pp. 19–31, 2005.

[77] J. Polastre, R. Szewczyk, and D. Culler, "Telos: Enabling Ultra-Low Power Wireless Research," in *Fourth International Symposium on Information Processing in Sensor Networks (IPSN 2005)*, 2005, pp. 364–369.

[78] E. Sozer, M. Stojanovic, and J. Proakis, "Underwater Acoustic Networks," *IEEE Journal of Oceanic Engineering*, vol. 25, no. 1, pp. 72–83, 2000.

[79] I. F. Akyildiz, D. Pompili, and T. Melodia, "Challenges for Efficient Communication in Underwater Acoustic Sensor Networks," *SIGBED Rev.*, vol. 1, no. 2, pp. 3–8, Jul. 2004.

[80] S. Pandya, J. Engel, J. Chen, Z. Fan, and C. Liu, "CORAL: Miniature Acoustic Communication Subsystem Architecture for Underwater Wireless Sensor Networks," in *IEEESensors*, 2005, p. 4 pp.

[81] A. S. Tanenbaum, *Moderne Betriebssysteme*, 3rd ed. Pearson Studium, 2009.

[82] V. Handziski, J. Polastre, J.-H. Hauer, C. Sharp, A. Wolisz, and D. Culler, "Flexible Hardware Abstraction for Wireless Sensor Networks," in *Proceedings of the 2nd European Workshop on Wireless Sensor Networks*, 2005, pp. 145–157.

[83] S. Yoo, I. Bacivarov, A. Bouchhima, Y. Paviot, and A. A. Jerraya, "Building Fast and Accurate SW Simulation Models Based on Hardware Abstraction Layer and Simulation Environment Abstraction Layer," in *Proceedings of the conference on Design, Automation and Test in Europe - Volume 1*, ser. DATE '03, 2003, pp. 10 550–.

[84] C. Buschmann and D. Pfisterer, "iSense: A Modular Hardware and Software Platform for Wireless Sensor Networks," 6. Fachgespräch Drahtlose Sensornetze der GI/ITG-Fachgruppe Kommunikation und Verteilte Systeme, Tech. Rep., 2007.

[85] L. Wanner, A. Junior, F. Polpeta, and A. Frohlich, "Operating System Support for Handling Heterogeneity in Wireless Sensor Networks," in *10th IEEE Conference on Emerging Technologies and Factory Automation (ETFA 2005)*, vol. 2, 2005, pp. 6 pp. –518.

[86] P. Levis, S. Madden, J. Polastre, R. Szewczyk, K. Whitehouse, A. Woo, D. Gay, J. Hill, M. Welsh, E. Brewer, and D. Culler, "TinyOS: An Operating System for Sensor Networks," in *Ambient Intelligence*, W. Weber, J. M. Rabaey, and E. Aarts, Eds. Springer Berlin Heidelberg, 2005, pp. 115–148.

[87] D. Gay, P. Levis, R. von Behren, M. Welsh, E. Brewer, and D. Culler, "The nesC Language: A Holistic Approach to Networked Embedded Systems," *SIGPLAN Not.*, vol. 38, no. 5, pp. 1–11, 2003.

[88] A. Dunkels, B. Gronvall, and T. Voigt, "Contiki - a Lightweight and Flexible Operating System for Tiny Networked Sensors," in *29th Annual IEEE International Conference on Local Computer Networks*, 2004, pp. 455–462.

[89] S. Bhatti, J. Carlson, H. Dai, J. Deng, J. Rose, A. Sheth, B. Shucker, C. Gruenwald, A. Torgerson, and R. Han, "MANTIS OS: An Embedded Multithreaded Operating System for Wireless Micro Sensor Platforms," *Mob. Netw. Appl.*, vol. 10, no. 4, pp. 563–579, 2005.

[90] B. Sadler and A. Swami, "Synchronization in Sensor Networks: an Overview," in *IEEE Military Communications Conference (MILCOM 2006)*, 2006, pp. 1–6.

[91] P. Santi, "Topology Control in Wireless Ad Hoc and Sensor Networks," *ACM Comput. Surv.*, vol. 37, no. 2, pp. 164–194, 2005.

[92] S. Hadim and N. Mohamed, "Middleware: Middleware Challenges and Approaches for Wireless Sensor Networks," *IEEE Distributed Systems Online*, vol. 7, no. 3, p. 1, 2006.

[93] G. S. Fishman, *Principles of Discrete Event Simulation.* New York, NY, USA: John Wiley & Sons, Inc., 1978.

[94] "The network simulator NS-2," http://nsnam.isi.edu/nsnam/index.php/Main_Page, (last access on 10th of January 2013).

[95] A. Varga and R. Hornig, "An overview of the OMNeT++ simulation environment," in *Proceedings of the 1st international conference on Simulation tools and techniques for communications, networks and systems & workshops*, ser. Simutools '08, 2008, pp. 60:1–60:10.

[96] P. Levis, N. Lee, M. Welsh, and D. Culler, "TOSSIM: Accurate and Scalable Simulation of Entire TinyOS Applications," in *Proceedings of the 1st international conference on Embedded networked sensor systems*, ser. SenSys '03, 2003, pp. 126–137.

[97] J. Haase, M. Damm, J. Glaser, J. Moreno, and C. Grimm, "SystemC-Based Power Simulation of Wireless Sensor Networks," in *Forum on Specification Design Languages (FDL 2009)*, 2009, pp. 1–4.

[98] W. Du, F. Mieyeville, and D. Navarro, "IDEAl: A SystemC-based System-level Simulator for Wireless Sensor Networks," in *Wireless Communications, Networking and Information Security (WCNIS), 2010 IEEE International Conference on*, june 2010, pp. 618 –622.

[99] W. Du, D. Navarro, and F. Mieyeville, "A Simulation Study of IEEE 802.15.4 Sensor Networks in Industrial Applications by System-Level Modeling," in *Fourth International Conference on Sensor Technologies and Applications (SENSORCOMM 2010)*, 2010, pp. 311–316.

[100] D. Navarro, W. Du, F. Mieyeville, and L. Carrel, "Hardware and Software System-Level Simulator for Wireless Sensor Networks," *Procedia Engineering*, vol. 5, pp. 228–231, 2010, eurosensor XXIV Conference.

[101] W. Du, F. Mieyeville, and D. Navarro, "Modeling Energy Consumption of Wireless Sensor Networks by SystemC," in *Fifth International Conference on Systems and Networks Communications (ICSNC 2010)*, 2010, pp. 94–98.

[102] R. Thomasius and S. Guttowski, "A SystemC Based Framework for the Evaluation of Proactive Power-Management Approaches in Distributed Energy Harvesting Systems," in *Fourth International Conference on Sensor Technologies and Applications (SENSORCOMM 2010)*, 2010, pp. 348–351.

[103] M. Vasilevski, H. Aboushady, F. Pecheux, and L. de Lamarre, "Modeling Wireless Sensor Network Nodes Using SystemC-AMS," in *Internatonal Conference on Microelectronics (ICM 2007)*, 2007, pp. 53–56.

[104] M. Vasilevski, N. Beilleau, H. Aboushady, and F. Pecheux, "Efficient and Refined Modeling of Wireless Sensor Network Nodes Using SystemC-AMS," in *Research in Microelectronics and Electronics (PRIME 2008)*, 2008, pp. 81–84.

[105] M. Vasilevski, F. Pecheux, N. Beilleau, H. Aboushady, and K. Einwich, "Modeling and Refining Heterogeneous Systems with SystemC-AMS: Application to WSN," in *Design, Automation and Test in Europe (DATE 2008)*, 2008, pp. 134–139.

[106] M. Farooq, S. Adhikari, J. Haase, and C. Grimm, "Modeling Methodology in SystemC-AMS for Embedded Analog Mixed Signal Systems," in *Proceedings of the 8th International Conference on Frontiers of Information Technology (FIT 2010*, 2010, pp. 27:1–27:6.

[107] J. Polley, D. Blazakis, J. McGee, D. Rusk, and J. Baras, "ATEMU: A Fine-grained Sensor Network Simulator," in *First Annual IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks(SECON 2004)*, 2004, pp. 145–152.

[108] B. Titzer, D. Lee, and J. Palsberg, "Avrora: Scalable Sensor Network Simulation with Precise Timing," in *Fourth International Symposium on Information Processing in Sensor Networks (IPSN 2005)*, 2005, pp. 477–482.

[109] T. Kirchner, N. Bannow, and C. Grimm, "Analogue mixed signal simulation using spice and SystemC," in *Design, Automation Test in Europe Conference Exhibition, 2009. DATE '09.*, april 2009, pp. 284 –287.

[110] A. Janek, "Architecture Design and Simulation of Energy Harvesting Sensor," Ph.D. dissertation, Graz University of Technology, Institute for Technical Informatics, 2008.

[111] A. Bachir, M. Dohler, T. Watteyne, and K. Leung, "MAC Essentials for Wireless Sensor Networks," *IEEE Communications Surveys Tutorials*, vol. 12, no. 2, pp. 222–248, 2010.

[112] G. Merrett, N. White, N. Harris, and B. Al-Hashimi, "Energy-Aware Simulation for Wireless Sensor Networks," in *6th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON 2009)*, 2009, pp. 1–8.

[113] O. Landsiedel, K. Wehrle, and S. Gotz, "Accurate Prediction of Power Consumption in Sensor Networks," in *The Second IEEE Workshop on Embedded Networked Sensors (EmNetS-II 2005)*, 2005, pp. 37–44.

[114] G. Merrett, A. Weddell, A. Lewis, N. Harris, B. Al-Hashimi, and N. White, "An Empirical Energy Model for Supercapacitor Powered Wireless Sensor Nodes," in *Proceedings of 17th International Conference on Computer Communications and Networks (ICCCN 2008)*, 2008, pp. 1–6.

[115] D. Estrin, R. Govindan, J. Heidemann, and S. Kumar, "Next Century Challenges: Scalable Coordination in Sensor Networks," in *Proceedings of the 5th annual*

ACM/IEEE international conference on Mobile computing and networking, ser. MobiCom '99, 1999, pp. 263–270.

[116] A. Pathak, L. Mottola, A. Bakshi, V. K. Prasanna, and G. P. Picco, "Expressing Sensor Network Interaction Patterns Using Data-Driven Macroprogramming," in *Fifth Annual IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops 2007)*, 2007, pp. 255–260.

[117] A. Lalomia, G. L. Re, and M. Ortolani, "A Hybrid Framework for Soft Real-Time WSN Simulation," in *Proceedings of the 2009 13th IEEE/ACM International Symposium on Distributed Simulation and Real Time Applications*, ser. DS-RT '09, 2009, pp. 201–207.

[118] K. Römer, O. Kasten, and F. Mattern, "Middleware Challenges for Wireless Sensor Networks," *SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 6, no. 4, pp. 59–61, 2002.

[119] C. Otto, A. Milenković, C. Sanders, and E. Jovanov, "System architecture of a wireless body area sensor network for ubiquitous health monitoring," *J. Mob. Multimed.*, vol. 1, no. 4, pp. 307–326, 2005.

[120] W. Wang, D. Peng, H. Wang, H. Sharif, and H.-H. Chen, "Energy Efficient Multirate Interaction in Distributed Source Coding and Wireless Sensor Network," in *IEEE Wireless Communications and Networking Conference (WCNC 2007)*, 2007, pp. 4091–4095.

[121] J. Pouwelse, K. Langendoen, and H. Sips, "Dynamic Voltage Scaling on a Low-Power Microprocessor," in *Proceedings of the 7th annual international conference on Mobile computing and networking*, ser. MobiCom '01, 2001, pp. 251–259.