Towards a Comprehensive Power Consumption Model for Wireless Sensor Nodes

Marc Hesse, Michael Adams, Timm Hörmann and Ulrich Rückert

Abstract—Energy efficiency is the most outstanding design criterion for wireless sensor nodes and especially wireless body sensors. Because a detailed measurement of the system's power consumption is not possible during the design process and often too complex for already manufactured devices, the power consumption has to be estimated. This leads to the need for a comprehensive and modular model for the power consumption of WSNs, which is proposed in this work. Due to the modular structure of the model the user is able to get a first estimate in an early stage of the design process (e.g. choose components) and to get a more accurate estimation later in the design process by lowering the abstraction level. This tackles the demanding trade-off between accuracy and usability in modeling.

I. Introduction

With the dissemination of wearable electronic devices and Internet of Things (IoT), the need for energy efficient wireless sensor nodes (WSN) and especially wireless body sensors (WBS) is increasing. Besides functionality, runtime and a small form factor are the most important aspects. Both are mainly influenced by the power consumption of the WSN, because an energy efficient operation allows to use a smaller battery pack without reducing the runtime of the system. Due to the increasing complexity of WSNs, the estimation of the system's runtime is challenging. This leads to the need of an appropriate model for the design of WSNs. The model should enable the theoretical estimation of a WSN's runtime before the hardware design process and without manual determination of a multitude of parameters. Because this requirement is often contrary to the model's accuracy, the system's behavior has to be abstracted. Due to different requirements, the model's level of abstraction needs to be adaptable. Therefore, it should be possible to get a first estimation of the system's runtime without having to determine multiple parameters beforehand. Additionally, the model should allow an accurate estimation in a later stage of the design process when it is easier to identify additional parameters of the system. In this paper, we propose a model for the power consumption estimation of WSNs, which includes the components microcontroller, sensor, transceiver, memory, interface, DC-DC converter and battery as shown in fig. 1. Furthermore, an abstracted description of the WSN's behavior (algorithm) is established. Every component is modeled by its modes of operation and can be used to form a WSN on system level.

Authors are with the Department of Cognitronics and Sensor Systems, CITEC, Bielefeld University, 33619 Bielefeld. E-mail: mhesse@techfak.uni-bielefeld.de
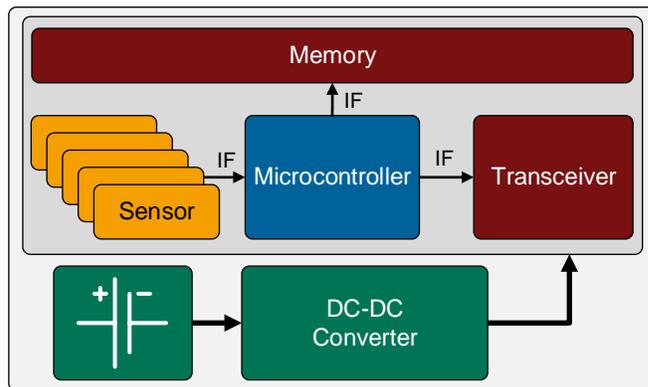
Fig. 1. Components of a WSN

The paper is structured as follows: Section II summarizes related work, before we introduce our power consumption model in section III. To demonstrate the usage of our proposed model, an example architecture of a WBS is analyzed in section IV. Afterwards, the results are presented in section V, and subsequently discussed in section VI. Finally, a prospect on our future work is given in section VII.

II. Related Work

A common method to determine the runtime of a system is by measuring its power consumption. Although there are several methods available [1], the current shunt method is often utilized, which is due to the rather easy handling. But the major drawback of this method is its smoothening character. For cycle accurate measurements it is necessary to place the shunt resistor as close as possible to the analyzed component (e.g. the microcontroller), which is sometimes not feasible in embedded systems. Additionally, the number of components and their modes of operation increases constantly. Therefore, only the current consumption for the whole system can be measured, which prevents a detailed analysis. This leads to the need of modeling and simulating the system.

Due to the increasing demand for WSN modeling there are several approaches discussed in literature. A detailed model of an embedded system was proposed by Simunic et al. [2] in 2001. The authors combined a cycle-accurate simulation of an ARM processor with an energy model. Their model included the core, memory, cache, IO lines, DC-DC converter, and the battery. The energy consumed per cycle is determined by each component's equivalent capacitance, the supply voltage and the clock rate, which

makes the model highly scalable and very fine-grained. But it is only applicable, if the application algorithm is already known and implemented for the used instruction set simulator. Additionally, their energy values are solely based on the manufacturer's specification in the data sheet, which was also stated by Steinke et al. [3], who extended the model for an energy-driven compiler. These models are capable of precisely estimating the power consumption, but they require a deep insight in the processor's architecture which is often not the case during the design process of a WSN.

Several detailed studies are presented by Tobola et al. in which they compare different microcontrollers for WSNs [4], quantify the impact of different sampling rates for various algorithms [5], and optimize the hard- and software implementation of an ECG sensor [6]. Likewise, Berlin et al. [7] increased the runtime of a wearable data logger by varying the sampling rates, the low-power modes, the type of used SD-card and by applying a run length encoding. Besides the fact that all these studies are rather specific, the components DC-DC converter and battery are not considered, which are significant for the system's energy efficiency. This is underlined by Abdallah et al. [8] who distinguish between the minimum energy operation point of the core (C-MEOP) and of the complete system (S-MEOP) including DC-DC converter and battery.

A generic approach is described by Benini et al. [9]. In their model, the system consists of power manageable components (PMC) whose modes of operations are modeled as power state machines (PSM). Additionally, the transitions between different modes cause costs (power, delay), which is in contrast to [2]. In our approach, we will follow the concept of manageable components with independent power state machines and will adapt it to the requirements of WSNs.

III. POWER CONSUMPTION MODEL

Based on the requirements analysis, a comprehensive model of the power consumption of WSNs should include the following features:

- System and components modeling
- Transition costs
- DC-DC converter and battery efficiency (S-MEOP)
- Adaptable level of abstraction

Every WSN consists of at least one sensor, a microcontroller and a wireless transceiver, which are connected via serial interfaces. The power supply includes the battery and, in most cases, a DC-DC converter. Due to the modular structure of WSNs the proposed model needs to be designed modular as well. In the following sections the components' models (sec. III-A) and the system model (sec. III-B) are described. The system model also includes the task definition, which specifies the active time for each component included in the system. These tasks can be used to model and simulate a large variety of system duty cycles, as described in sec. III-C. The model is implemented using MATLAB [10].
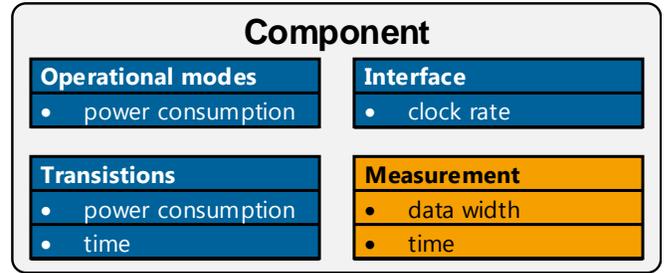
Fig. 2. The components' model and its parameters. The user forms the modeled system by selecting and configuring several components. Afterwards, different tasks for the system can be defined and simulated.

A. Components

The components are partitioned into the functional groups: sensor, transceiver, memory, interfaces, microcontroller, DC-DC converter and battery. Every component in each functional group can be defined by the same parameters, which are shown in fig. 2. After defining a device by its parameters it is possible to store it in a database, which leads to a growing device database for future analyses. As described in [9], each component is modeled by a power state machine.

1) Sensors: A sensor is defined by its power consumption in every operational mode (e.g. different measurements, idle, standby), the power and delay for every transition between the modes and the active time needed for every possible measurement. Additionally, the available interfaces, their clock rates and the device specific protocol overhead are noted. The data width depends on the chosen measurement and determines the payload for the interface operation.

2) Transceiver: Similar to a sensor, a transceiver is defined by its power consumption in every mode (e.g. TX, RX, active, idle, standby) and the power and delay for every transition. The available interfaces, their clock rates and the protocol overhead are noted in the model. Based on these parameters, the time needed for sending a specific number of bytes (including payload and protocol overhead) is calculated by a device specific function. Only the payload is needed as an input parameter.

3) Memory: A memory device is defined by the power consumption in every mode (e.g. read, write, idle, standby), the power and delay for every transition, and the time needed for writing or reading a specific number of bytes which includes payload and overhead. Available interfaces, their clock rates and protocol overhead are noted.

4) Interfaces: An interface specific function calculates the length of the data packet, which consists of the interface overhead, the device's protocol overhead and the device's payload. The overheads for the most common interfaces (SPI, I2C, UART) are depicted in [11]. The time needed for the interface communication is added to the microcontroller's active time.

5) Microcontroller: A microcontroller is defined by the power consumption in every mode (e.g. active at different clock rates, idle, standby) and the power and delay for

every transition. Additionally, the power consumption for the active interfaces is added to the microcontroller's active mode power consumption. The active time is specified by the user at task level and depends on the implemented algorithm. An algorithm's runtime can be determined by estimation, measurement, or the usage of an instruction set simulator. The active time of the sensors, including the time needed for the serial interface communication, is added to the microcontroller's active time.

6) Converter: The efficiency of the DC-DC converter depends on the input voltage, output current and output voltage. In our work it is modeled by a lookup table and linear interpolation, which is a commonly used method [12].

7) Battery: Our battery model is based on Peukert's law [13], which is an empirical model with very low computational complexity [14]. In our implementation (eqn. 1), a battery specific Peukert constant k is only applied, if the load current exceeds the specified maximal continuous discharge current $I_{BAT,cont.}$ of the battery. Otherwise, $I_{BAT}(t)$ equals $I_{DC,ave}(t)$.

$$I_{BAT}(t) = I_{DC,ave}(t) \times \left(\frac{I_{DC,ave}(t)}{I_{BAT,cont.}} \right)^{k-1} \quad (1)$$

where $I_{DC,ave}(t)$ is the average current drawn by the DC-DC converter over the past 1000 time steps [2].

B. System Configuration and Task Modeling

In the next step, the modeled components can be used to define multiple system configurations. A configuration selects the components for the system and defines the inactive mode for each device. The system's inactive power consumption is given by the sum of all devices' power consumptions in their inactive modes. A task definition selects the active mode of every component and therefore its active time and the transition costs. The active time of the sensors and the wireless transceiver are based on the chosen measurements of the sensors, as described in sec. III-A. The tasks are modeled with 1 μs time steps. The energy consumed by every defined task is given by the active times of the components and their power consumption in the selected active modes.

C. Duty Cycle

The system's duty cycle can be modeled by the execution frequency of the different tasks (sampling rate). The possibility to define the sampling rate independently for each task allows the user to flexible model the system's behavior. Therefore, the overall power consumption is given by the inactive power consumption and the type and number of executed tasks.

IV. EXAMPLE WSN

An example architecture of a WSN was defined to demonstrate the capabilities of the proposed model. First of all, the components and their chosen operational modes are selected in sec. IV-A. Afterwards, two different tasks were defined in sec. IV-B and used for the duty cycle modeling in sec. IV-C.
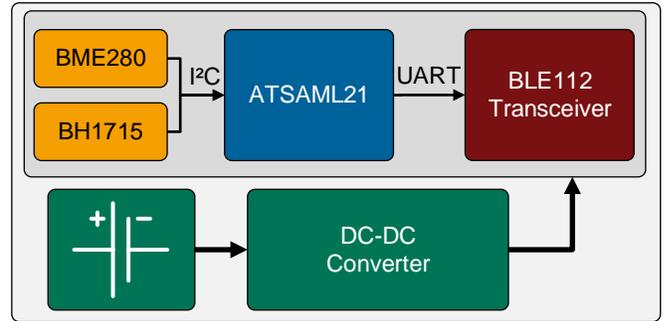
Fig. 3. Example architecture of the analyzed WSN

A. Configurations

The chosen example architecture of the WSN can be seen in fig. 3, which could serve as a wireless environmental sensor node. The Atmel SAML21 with ARM Cortex M0+ core is used as the main microcontroller. A Rohm BH1715 ambient light sensor and a Bosch Sensortec BME280 combined humidity and pressure sensor are connected via I2C to the microcontroller. The Bluegiga BLE112 Bluetooth Low Energy transceiver is connected via an UART interface. To demonstrate the flexibility of the model and to underline the importance of an efficient power supply for the system, two different step down converters and five different, commonly used, battery types were analyzed. In active mode, the SAML21 microcontroller uses the *PL0* mode at 12 MHz. For the inactive mode, in configuration *C-IDL* and *C-STB* the *Idle PL0* mode or the *Standby* mode is selected, respectively. All other devices use the same operational modes in configuration *C-IDL* and *C-STB*. An overview of the devices and their operational modes is given in tbl. I.

TABLE I
EXAMPLE WSN: COMPONENTS AND MODES

Component	Active mode	Inactive mode
SAML21	PL0 at 12MHz *0.3515 mA*	**C-IDL:** Idle PL0 at 12MHz *0.1968 mA*
		C-STB: Standby (PD0-PD2 in retention) *0.0060 mA*
BME280	Measurement Temp./Hum./Press. *0.468 mA*	Standby *0.0005 mA*
BH1715	Measurement Low-Resolution Mode *0.190 mA*	Powerdown *0.0010 mA*
BLE112	Send Notification *13 mA*	PM2 *0.0809 mA*
DC-DC converter	TI TPS62743, TI TPS62730	
Batteries	CR2025, CR2032, LIR2032, IPC402025, CR123	

The denoted current consumptions for the devices are either measured (Atmel SAML21, Bluegiga BLE112) or

taken from the data sheet specifications (Bosch BME280, Rohm BH1715).

B. Tasks

The usage and therefore the active time of the devices is defined by the tasks. Each sensor's active time depends on the selected measurement and can be found in its data sheet [15,16]. The packet size of the interface is, as described in section III-A, set by the sensor's payload (BME280: 48 Bit, BH1715: 16 Bit), the device's protocol overhead and the interface overhead. The active time for the interface equals packet size multiplied with the interface's clock rate. Likewise, the active time for the wireless transceiver is given by the time to send a notification with 48 Bit + 16 Bit payload, which was measured by applying variable payload sizes. Finally, the active time for the microcontroller is the sum of the transceiver's and the sensors' active times including their interfaces' active times and the algorithm's runtime. In our example, the microcontroller's active time needed to check the sensor values and to determine, if a wireless transmission is necessary is set to $1\,ms$. This follows our environmental sensor node example application. An overview for the components' active times in task *T-MST* and their sequence is given in fig. 4. The visualization was generated using TORSCHE toolset [17].
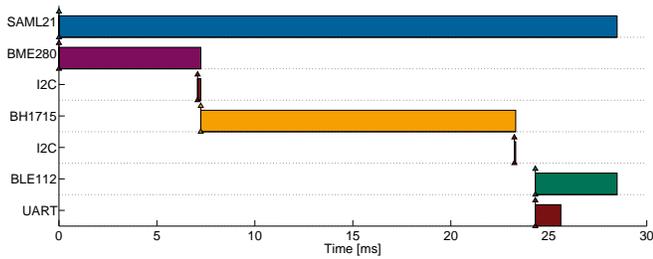
Fig. 4.   Example WSN: Task *T-MST* sequence

To enable an optional wireless transmission, the task *T-MS* was defined, in which the payload for the wireless transceiver, and therefore its active time, is set to zero. The overall durations for both tasks, including the transition times for both configurations, are shown in tbl. II.

TABLE II
EXAMPLE WSN: TASK DURATIONS

	Task *T-MST*	Task *T-MS*
Conf. *C-IDL*	28489 μs	24314 μs
Conf. *C-STB*	28493 μs	24318 μs

C. Duty Cycle

The example system's duty cycle is modeled for a wide sampling rate range of the defined tasks. Task *T-MS* (microcontroller and sensors active), as the default task, is executed with a rate of $0.0001\,Hz$ to $2\,Hz$. Task *T-MST*, which additionally uses the wireless transceiver, is executed

with a predefined ratio of the sampling rate of task *T-MS*. Applied values for the ratio were 0.1, 0.5 and 1.0. For example, if task *T-MS* is executed at $1\,Hz$ and the chosen wireless ratio is 0.5, task *T-MST* will be executed at $0.5\,Hz$. So the microcontroller will be active once every second and both tasks are alternatingly executed. The energy consumption of the tasks is determined by the devices' active times (fig. 4 and tbl. II) and their power consumption during that time (tbl. I). During the inactive period, the system's power consumption is defined as the sum of all devices' inactive power consumption (tbl. I).

V. RESULTS

The following section points out the capabilities of our proposed modeling technique. First, the energy consumption for the two defined tasks and the inactive power consumption for the two different configurations are analyzed (sec. V-A). Based on that results, the optimal power supply is identified and used for the duty cycle simulations in sec. V-B.

A. Tasks and Inactive Mode

The energy consumed by the tasks and the inactive power consumption for the WSN is shown in fig. 5. Additionally, the devices' share in the overall values is emphasized. Note, that these values do not include the components DC-DC converter and battery. Because the wireless transceiver is inactive in task *T-MS*, the energy consumption is reduced by the transceivers share and the microcontroller's active time is accordingly decreased. The interfaces' energy consumptions are relatively low and therefore not visible in fig. 5. Since the configurations only differ by the chosen inactive mode and their transition costs, task energy consumption is nearly equal for both configurations. But the impact of using the deeper inactive mode in configuration *C-STB* can clearly be seen in the inactive modes' power consumption. The choice of a suitable configuration depends on the duty cycle, which will be discussed in sec. V-B.
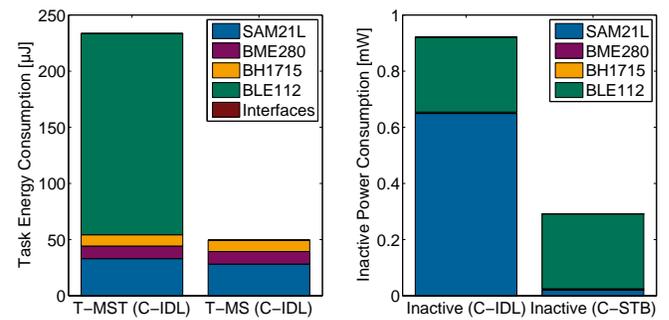
Fig. 5.   Task energy and inactive power consumption

The effective task energy including DC-DC converter and battery is shown in fig. 6. While the influence of the different DC-DC converters is rather small, the coin cell batteries are not capable of efficiently supplying the high peak currents especially needed for the wireless transmission. This effect is reduced in task *T-MS*, but shows the same pattern. The inactive power consumption for configuration

C-STB, in dependence of the modeled power supplies, is shown in fig. 7. Every battery type is able to deliver the low standby current efficiently. However, a rather big deviation in the converter's performance occurs, which is due to a significant difference in the converters' efficiencies in the specific operation point.
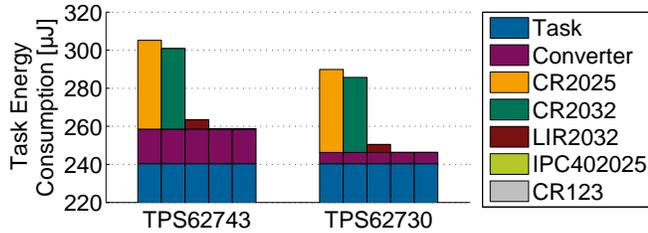
Fig. 6.   Task *T-MST* energy consumption (*C-IDL*)
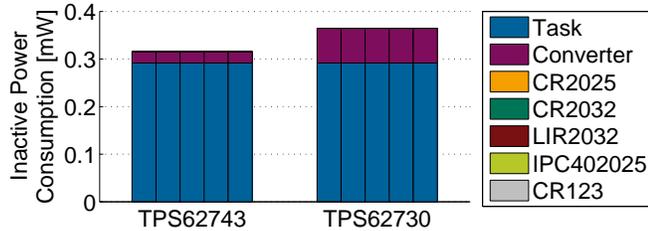
Fig. 7.   Inactive power consumption (*C-STB*)

Based on that results, it is possible to determine the most efficient power supply (consisting of converter and battery) for every configuration and task combination. The most efficient power supply is equal for both configurations and both tasks, but differs for task execution and inactive mode. For task execution, the combination of *TPS62730* converter and *IPC402025* battery is optimal. In inactive mode, the *TPS62743* converter with any battery is more efficient. Therefore, the optimal power supply for the WSN depends on the system's duty cycle and will be determined in the following section.

B. Duty Cycle

The defined configurations and tasks were used to simulate a wide range of different duty cycles. The sampling rate of task *T-MS* is in the range from 0.0001 Hz to 2 Hz, while task *T-MST* is executed with a predefined ratio of 0.1, 0.5 and 1.0. Because the optimal power supplies for task execution and inactive mode are not equal, the simulations were done for both optimal converter and battery combinations. Due to the lower inactive power consumption in configuration *C-STB*, it should be preferred. However, choosing a deeper standby mode results in higher transitions costs in terms of power and delay. Therefore, the saved energy in the lower mode has to exceed the energy needed for the transitions [9]. Because both configurations fulfill this requirement in our sampling rate range, *C-STB* is chosen. The system's power consumption as a function of the sampling rate and the wireless ratio (WL) is shown in fig. 8. Because the inactive

power comsumption is equal for every battery, the ideal task execution battery *IPC402025* is chosen in both cases. The results are shown for the optimal task power supply (*TPS62730+IPC402025*) and the optimal inactive power supply (*TPS62743+IPC402025*).
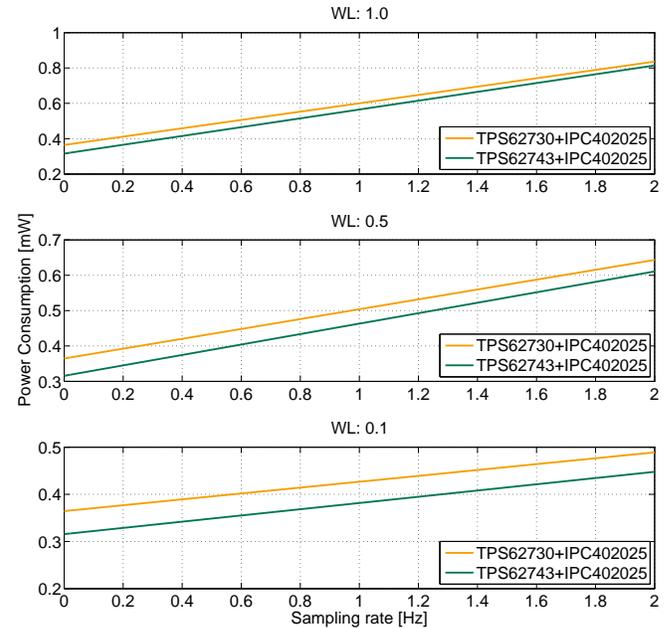
Fig. 8.   Overall WSN power consumption (C-STB)

The power consumption increases linearly with the sampling rate. A lower wireless ratio and therefore a decreased use of the wireless transceiver results in a significantly lower power consumption. The simulation of the task execution and inactive power supplies enables the determination of a break-even sampling rate, at which the optimal power supply for the WSN changes. Below that break-even sampling rate, the optimal inactive power supply is overall more efficient than the optimal task power supply. Above that break-even sampling rate, the optimal task power supply is more efficient. This is denoted in tbl. III. The break-even sampling rate in configuration *C-STB* is significantly higher compared to *C-IDL*, which is due to the lower inactive power consumption and therefore the increased influence of the task power supply efficiency.

TABLE III
BREAK-EVEN SAMPLING RATE

	WL: 1.0	WL: 0.5	WL: 0.1
Conf. *C-IDL*	0.18 Hz	0.32 Hz	0.80 Hz
Conf. *C-STB*	3.60 Hz	6.00 Hz	12.70 Hz

VI. DISCUSSION AND CONCLUSION

We proposed a comprehensive and easy to use model, which enables a rapid estimation of the effective power consumption of future WSNs. By analyzing an example

architecture of a WSN we were able to quantify task energy consumption and inactive power consumption. After identifying the optimal power supply for task execution and inactive mode we simulated a wide range of duty cycles and optimized the power supply in dependence of the sampling rate. The determined break-even sampling rate gives the designer an estimation of the system's optimal configuration and the possibility to adapt the system's hardware design to the intended use. Based on that knowledge, it is possible to guarantee a desired runtime of the system by defining limits for the sampling range or the ratio of wireless transmissions. For our example of the wireless environmental sensor node, the optimal inactive power supply should be chosen, because those systems usually operate below the identified break-even sampling rates. A limitation of our model is the simplified battery model. Therefore, we estimated only the system's power consumption instead of the runtime based on the battery discharge. Additionally, the model needs the active time of the microcontroller as user input, which is sometimes hard to determine. But the modular structure and the adaptable abstraction level gives the opportunity to either estimate, measure, or simulate a algorithm's runtime.

VII. FUTURE WORK

In future work we want to implement an extensive battery model, which handles rate-dependent capacity, temperature effects and capacity fading [14]. Furthermore, we want to define the active time of the microcontroller as a function of its clock rate in order to extrapolate a known runtime of an algorithm for a variety of configurations. Additionally, important features like interrupt handling and DMA operation have to be modeled. Currently, we are designing the succeeding version of our wireless body sensor *BG-V4.2* [18], which will be used to improve and validate the proposed model on the basis of a manufactured WBS. Furthermore, we want to evaluate different configurations by the outcome of various applications. For instance, the correlation between the accuracy of machine learning algorithms [19]–[21] and the power consumption would enable the assessment of different configurations. This could establish a correlation between power consumption and accuracy, which is already a rising topic of recent research [22].

ACKNOWLEDGMENT

REFERENCES

[1] Z. Nakutis, "Embedded systems power consumption measurement methods overview," *MATAVIMAI*, vol. 2, no. 44, pp. 29–35, 2009.

[2] T. Simunic, L. Benini, and G. De Micheli, "Energy-efficient design of battery-powered embedded systems," *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, vol. 9, no. 1, pp. 15–28, Feb 2001.

[3] S. Steinke, M. Knauer, L. Wehmeyer, and P. Marwedel, "An accurate and fine grain instruction-level energy model supporting software optimizations," in *Proc. of PATMOS*. Citeseer, 2001.

[4] A. Tobola, O. Korpok, H. Leutheuser, B. Schmitz *et al.*, "System design impacts on battery runtime of wearable medical sensors," in *The proceedings from the mobilemed 2014 conference*, 2014.

[5] A. Tobola, F. J. Streit, C. Espig, O. Korpok *et al.*, "Sampling rate impact on energy consumption of biomedical signal processing systems," in *Wearable and Implantable Body Sensor Networks (BSN), 2015 IEEE 12th International Conference on*. IEEE, 2015, pp. 1–6.

[6] A. Tobola, C. Espig, F. J. Streit, O. Korpok *et al.*, "Scalable ecg hardware and algorithms for extended runtime of wearable sensors," in *Medical Measurements and Applications (MeMeA), 2015 IEEE International Symposium on*. IEEE, 2015, pp. 255–260.

[7] E. Berlin, M. Zittel, M. Braunlein, and K. Van Laerhoven, "Low-power lessons from designing a wearable logger for long-term deployments," in *Sensors Applications Symposium (SAS), 2015 IEEE*. IEEE, 2015, pp. 1–6.

[8] R. A. Abdallah, P. S. Shenoy, N. R. Shanbhag, and P. T. Krein, "System energy minimization via joint optimization of the dc-dc converter and the core," in *Proceedings of the 17th IEEE/ACM international symposium on Low-power electronics and design*. IEEE Press, 2011, pp. 97–102.

[9] L. Benini, A. Bogliolo, and G. De Micheli, "A survey of design techniques for system-level dynamic power management," *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, vol. 8, no. 3, pp. 299–316, 2000.

[10] MATLAB, *Version 8.1.0 (R2013a)*. Natick, Massachusetts: The MathWorks Inc., 2013.

[11] K. Mikhaylov and J. Tervonen, "Evaluation of power efficiency for digital serial interfaces of microcontrollers," in *New Technologies, Mobility and Security (NTMS), 2012 5th International Conference on*. IEEE, 2012, pp. 1–5.

[12] L. Benini, G. Castelli, A. Macii, E. Macii *et al.*, "Discrete-time battery models for system-level low-power design," *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, vol. 9, no. 5, pp. 630–640, 2001.

[13] W. Peukert, "Über die abhängigkeit der kapazität von der entladestromstärke bei bleiakkumulatoren," *Elektrotechnische Zeitschrift*, vol. 20, pp. 20–21, 1897.

[14] R. Rao, S. Vrudhula, and D. N. Rakhmatov, "Battery modeling for energy aware system design," *Computer*, vol. 36, no. 12, pp. 77–87, 2003.

[15] *BH1715FVC Ambient Light Sensor IC*, Rev. c ed., Rohm Semiconductor, November 2011.

[16] *BME280 Combined humidity and pressure sensor*, Rev. 1.2 ed., Bosch Sensortec, October 2015.

[17] P. Sucha, M. Kutil, M. Sojka, and Z. Hanzalek, "Torsche scheduling toolbox for matlab," in *Computer Aided Control System Design, 2006 IEEE International Conference on Control Applications, 2006 IEEE International Symposium on Intelligent Control, 2006 IEEE*. IEEE, 2006, pp. 1181–1186.

[18] M. Hesse, P. Christ, T. Hörmann, and U. Rückert, "A respiration sensor for a chest-strap based wireless body sensor," in *SENSORS, 2014 IEEE*, Nov 2014, pp. 490–493.

[19] P. Christ, G. Sievers, J. Einhaus, T. Jungeblut *et al.*, "Pareto-optimal signal processing on low-power microprocessors," in *Proc. of the 12th IEEE Int. Conf. on SENSORS*, Nov. 2013, pp. 1843–1846.

[20] T. Hörmann, P. Christ, M. Hesse, and U. Rückert, "Robust estimation of physical activity by adaptively fusing multiple parameters," in *Body Sensor Networks*. IEEE, Jun. 2015.

[21] T. Hörmann, M. Hesse, P. Christ, M. Adams *et al.*, "Fine-grained prediction of cognitive workload in a modern working environment by utilizing short-term physiological parameters," in *Proc. of the 9th Int. Conf. on Bio-Inspired Systems and Signal Processing*, 2016.

[22] U. Jensen, P. Kugler, M. Ring, and B. M. Eskofier, "Approaching the accuracy–cost conflict in embedded classification system design," *Pattern Analysis and Applications*, pp. 1–17, 2015.