

Center for Cognitive Interaction Technology

Kognitronik und Sensorik

Prof. Dr.-Ing. U. Rückert

Modellbasierte Entwurfsmethodik für drahtlose Sensorknoten

zur Erlangung des akademischen Grades eines

DOKTOR-INGENIEUR (Dr.-Ing.)

der Technischen Fakultät

der Universität Bielefeld

genehmigte Dissertation

von

Marc Hesse

Referent: Prof. Dr.-Ing. Ulrich Rückert

Korreferent: Prof. Dr. rer. nat. Axel Schneider

Tag der mündlichen Prüfung: 25.06.2021

Bielefeld / Juni 2021

DISS KS / 20

Dieses Werk ist lizenziert unter einer Creative Commons Namensnennung 4.0 International Lizenz (CC BY 4.0): <https://creativecommons.org/licenses/by/4.0/deed.de>

Gedruckt auf alterungsbeständigem Papier (gemäß DIN EN ISO 9706).

Kurzzusammenfassung

Die stetig steigende Komplexität von eingebetteten mikroelektronischen Systemen, die sich durch eine Vielzahl von Komponenten mit jeweils verschiedenen möglichen Zustände jeder einzelnen Komponente auszeichnet, führt dazu, dass eine ganzheitliche Optimierung nicht mehr, oder nur sehr zeitaufwändig möglich ist. Dieses zeigt sich auch im Bereich der drahtlosen Sensorknoten, welche sich durch mehrere Sensoren, ein Bauteil zur Informationsverarbeitung, eine Funkschnittstelle und eine Energieversorgung mittels Batterie auszeichnen. Dessen Optimierung besteht in der Maximierung der Energieeffizienz, welche als die Anpassung der Leistungsfähigkeit an die Anforderungen der Anwendung bei gleichzeitiger Minimierung der Leistungsaufnahme definiert ist. Ziel der Arbeit ist daher die Entwicklung einer Entwurfsmethodik für drahtlose Sensorknoten, welche sowohl die Energieeffizienz des Systems optimiert, als auch den Entwurfsprozess systematisiert.

Einleitend wird zunächst der Stand der Technik drahtloser Sensorknoten hinsichtlich Systemstruktur, Komponenten, Messgrößen und Parametern sowie der verfügbaren Systeme analysiert. Anschließend wird die bisher übliche Methodik zur Messung der Leistungsaufnahme von eingebetteten Systemen dargestellt und das Vorgehen anhand von zwei Praxisbeispielen verdeutlicht, wodurch sich die Komplexität der Konfigurationsmöglichkeiten hinsichtlich der Optimierung bestätigt. Vergleichend dazu wird der Stand der Technik für die Modellierung von drahtlosen Sensorknoten analysiert.

Auf dieser Grundlage wird die Methodik für den Entwurf von drahtlosen Sensorknoten beschrieben. Zunächst erfolgt die Einordnung in den bisherigen Entwurfsprozess drahtloser Sensorknoten und im Anschluss daran die Ableitung des Handlungsbedarfs. Darauf folgt die theoretische Betrachtung des aufgestellten Modells anhand des grundlegenden System- und Komponentenmodells. Anschließend werden die spezifischen Modelle für die einzelnen Komponenten und deren Zusammenschluss zu einem Modell für das Gesamtsystem beschrieben. Darauf aufbauend wird das Systemverhalten mittels der Definition von Tasks, dem Tastgrad und dem Datenfluss abgebildet. Somit wird eine Entwurfsraumexploration hinsichtlich Leistungsfähigkeit und Leistungsaufnahme ermöglicht.

Anschließend wird die Methodik des modellgestützten Entwurfs von drahtlosen Sensorknoten exemplarisch angewendet, wobei der Entwurfsraum in jedem Schritt des Entwurfsprozesses eingeschränkt und schlussendlich auf einen kleinen Lösungsraum fokussiert wird. Somit wird demonstriert, dass das Verfahren in der Lage ist, den Entwurf zu systematisieren und durch die Simulation der verschiedenen Entwurfsschritte zielführender zu gestalten. Das Vorgehen erhöht somit zum einen die Effizienz des Sensorknotens und zum anderen die Effizienz der Entwicklung von drahtlosen Sensorknoten oder auch eingebetteten Systemen im weiteren Sinne.

Inhaltsverzeichnis

1	Einleitung	1
2	Drahtlose Sensorknoten	5
2.1	Systemstruktur	5
2.2	Komponenten	6
2.2.1	Sensoren	6
2.2.2	Informationsverarbeitung	6
2.2.3	Aktoren	7
2.2.4	Spannungswandler	9
2.2.5	Energiespeicher	10
2.2.6	System on a Chip	11
2.3	Messgrößen und Parameter	11
2.3.1	Physiologische Größen	12
2.3.2	Biokinematische Größen	13
2.3.3	Umgebungsgrößen	14
2.3.4	Sensorfusion	16
2.4	Verfügbare Systeme	16
2.4.1	Umgebungssensorknoten	17
2.4.2	Körpersensorknoten	18
2.5	Zusammenfassung	21
3	Vermessung drahtloser Sensorknoten	23
3.1	Methodik	23
3.2	Arbeiten in der Literatur	25
3.3	Vermessung Körpersensorknoten BG-V4.2	25
3.4	Vermessung Umgebungssensorknoten TI CC2560STK	27
3.5	Zusammenfassung und Bewertung der Methode	31
4	Modellierung drahtloser Sensorknoten	33
4.1	Mikrocontroller	33
4.2	Sensoren	38
4.3	Aktoren	38
4.3.1	Funktransceiver	38
4.3.2	Speicher	39
4.4	Indikation	39
4.5	Schnittstellen	39
4.6	Spannungswandler	39
4.7	Energiespeicher	41
4.7.1	Mathematische Modelle	42

4.7.2	Elektrische Modelle	44
4.7.3	Generierung der Modelle	50
4.8	System	56
4.9	Energiemanagement	59
4.10	Zusammenfassung und Bewertung der Methode	60
5	Modellgestützte Entwurfsmethodik	63
5.1	Einordnung in den Entwurfsprozess	63
5.2	Handlungsbedarf	67
5.2.1	Abzubildendes System	67
5.2.2	Skalierbarkeit des Modells	68
5.3	Grundlegendes Systemmodell	70
5.4	Grundlegendes Komponentenmodell	71
5.5	Spezifische Komponentenmodelle	73
5.5.1	Komponente Mikrocontroller	74
5.5.2	Komponente Sensor	76
5.5.3	Komponente Funktransceiver	77
5.5.4	Komponente Speicher	78
5.5.5	Komponente Spannungswandler	78
5.5.6	Komponente Energiespeicher	80
5.5.7	Schnittstellen zwischen den Komponenten	84
5.6	Datenfluss	85
5.7	Systemkonfiguration	86
5.8	Taskkonfiguration	89
5.9	Tastgrad	91
5.10	Zusammenfassung	93
6	Modellgestützter Entwurf eines Sensorknotens	95
6.1	Bedarfsanalyse	95
6.2	Bauteilauswahl PMC	95
6.2.1	Komponente SoC	96
6.2.2	Komponente Umgebungslichtsensor	101
6.2.3	Komponente Umgebungssensor	104
6.2.4	Komponente Beschleunigungssensor	109
6.3	Komponenten der Spannungsversorgung	112
6.3.1	Spannungswandler	112
6.3.2	Energiespeicher	117
6.4	Entwurfsraum der Systemkonfiguration	122
6.5	Task	129
6.6	Tastgrad	136
6.7	Bewertung des Systementwurfs	144

6.8 Zusammenfassung	149
7 Zusammenfassung und Ausblick	151
7.1 Zusammenfassung	151
7.2 Abgleich mit dem Handlungsbedarfs	152
7.2.1 Abzubildendes System	153
7.2.2 Skalierbarkeit des Modells	155
7.3 Mögliche Einschränkungen	156
7.3.1 Validität	156
7.3.2 Energieeffizienz	157
7.3.3 Generierung der Daten	157
7.3.4 Generische Anwendbarkeit	157
7.4 Ausblick	158
7.4.1 Modellgestützter Entwurf der Software	158
7.4.2 Skalierung	158
7.4.3 Anwendungen	159
7.4.4 Antizipation	160
Abkürzungsverzeichnis	169
Symbolverzeichnis	173
Literaturverzeichnis	177
Eigene Veröffentlichungen	195
Betreute Arbeiten	199
A Übersicht drahtloser Sensorknoten	201

1 Einleitung

Der Begriff *Internet der Dinge* (*Internet of Things* (IoT)) beschreibt den aktuell vorherrschenden Trend der Vernetzung technischer Systeme. Während bisher vernetzte technische Geräte wie beispielsweise ein PC, ein Tablet-Computer oder ein Smartphone immer mit menschlicher Interaktion verbunden waren, so werden zunehmend auch rein technische Systeme untereinander vernetzt. Das bedeutet, dass diese Geräte ohne menschliches Handeln in der Lage sind, Informationen auszutauschen und somit eine umfassende und stetig wachsende Datenbasis zu generieren (*Big Data*). Der Schlüssel dazu liegt in der Verbindung verschiedener aktueller Technologien: Technische Geräte, die ihre Umgebung wahrnehmen können, die Vernetzung dieser Geräte und die Analyse großer Datenmengen zur Wissensgenerierung [13]. Die Anwendung dieses Konzepts findet sich in vielen Bereichen, wie beispielsweise der Industrie zur Überwachung von Produktionsanlagen und Fertigungsabläufen [231], dem Smart-Home Bereich zur Automatisierung von technischen Alltagsgegenständen in der eigenen Wohnung [111], aber auch im Bereich der tragbaren Elektronik (*Wearables*) zur Überwachung der eigenen Gesundheit (*Quantified Self*) [194]. Bei den auf unterster Ebene notwendigen technischen Geräten zur Wahrnehmung der Umgebung handelt es sich zumeist um kleine Sensorknoten, die im Wesentlichen aus einem oder mehreren Sensoren, einer Einheit zur Informationsverarbeitung und einer Schnittstelle zur Vernetzung bestehen [5]. Erst die technische Weiterentwicklung im Bereich der Mikroelektronik und die daraus resultierende Miniaturisierung der Sensorknoten bei gleichzeitig sinkenden Kosten ermöglicht die Integration von Sensorknoten in bereits bestehende technische Systeme und deren Vernetzung.

Die Relevanz der IoT-Geräte hat in letzter Zeit stark zugenommen. So steigt die in Abbildung 1.1 dargestellte Anzahl der jährlichen Publikationen, bei denen sich die Begriffe *Internet of Things* oder *IoT* in der Kurzzusammenfassung finden, stetig. Jedoch müssen die

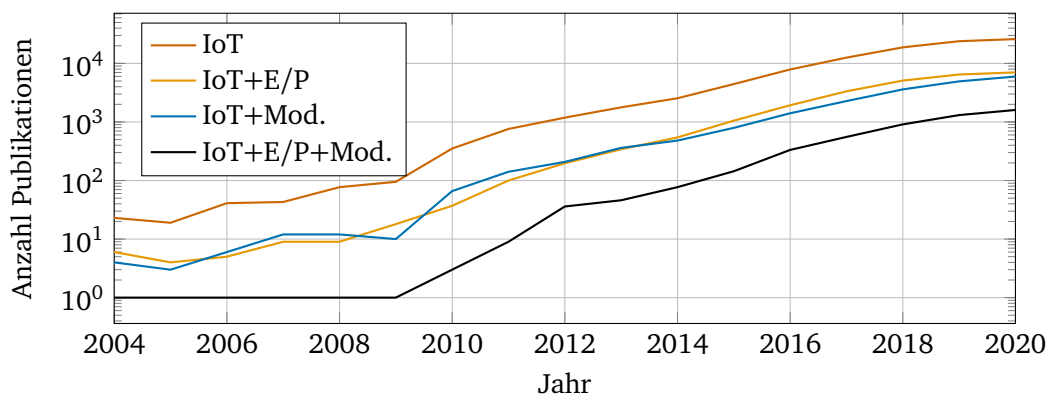


Abbildung 1.1.: Entwicklung der Anzahl der jährlichen Publikationen für die Begriffe *Internet of Things* und *IoT* zusammen mit verschiedenen Schlüsselwörtern. Die Daten wurden mit *Dimensions* [52] ermittelt.

dafür notwendigen Sensorknoten die Ansprüche des Marktes erfüllen. Neben möglichst geringen Kosten werden besonders hohe Ansprüche an die Laufzeit der Geräte gestellt. Diese kann entweder über die Verwendung eines Energiespeichers mit großer Kapazität oder über eine geringe Energieaufnahme der elektronischen Komponenten erreicht werden. Da ein größerer Energiespeicher, bei gleicher Technologie, jedoch auch erhöhte Kosten, mehr Gewicht und mehr Volumen bedeutet, ist die Reduzierung der Energieaufnahme ein wichtiger Aspekt aktueller Forschung. Dieses spiegelt sich auch in der steigenden Anzahl der jährlichen Publikationen, die neben den Begriffen *Internet of Things* bzw. *IoT* auch die Begriffe *Energy* bzw. *Power* (E/P) in der Kurzzusammenfassung aufführen (vgl. Abbildung 1.1). Eine geringe Energieaufnahme ermöglicht somit eine lange Laufzeit der Geräte bei gleichzeitig kompakter Bauform. Besonders wichtig ist eine geringe Energieaufnahme bei tragbaren, also körpernahen Sensorknoten, da diese zum einen die nutzende Person nicht einschränken dürfen und zum anderen die Laufzeit entscheidend für die Benutzungsakzeptanz ist. Bei Sensorknoten in den Bereichen Smart-Home oder Industrie 4.0 spielt hingegen die Wartbarkeit häufig eine große Rolle. Da es sich hier um Sensornetze aus einzelnen verteilten Sensorknoten handelt, ist ein Laden oder Wechseln der Batterien in einigen Fällen nicht vorgesehen oder sollte so selten wie möglich nötig sein. Da die Vernetzung von technischen Geräten ein wichtiges Themenfeld mit stetig steigendem Bedarf bei gleichzeitigem Handlungsbedarf darstellt, fokussiert sich diese Arbeit auf die Reduzierung der Energieaufnahme von Sensorknoten, welche im IoT-Umfeld eingesetzt werden können. Die erforderlichen Sensorknoten bestehen aus immer mehr Komponenten, wobei jede Komponente über eine Vielzahl von Parametern und möglichen Betriebszuständen verfügt. Zusammen mit dem vielfältigen Angebot verschiedenster Elektronikhersteller ergibt sich somit eine in vielen Fällen nicht mehr überschaubare Anzahl an Entwurfsmöglichkeiten für einen Sensorknoten. Da der Entwurf eines Sensorknotens jedoch immer bedarfsgerecht für eine spezifische Anwendung erfolgen sollte, wird im Rahmen dieser Arbeit ein modellgestützter Entwurfsablauf vorgestellt. Im Folgenden wird zunächst eine Begriffsklärung gegeben und das Ziel der Arbeit formuliert.

Drahtloser Sensorknoten Ein drahtloser Sensorknoten zeichnet sich durch eine Funkschnittstelle zur Datenübertragung und durch eine Energieversorgung mittels Batterie aus. Auch wenn der Einsatzzweck und die Leistungsfähigkeit der Systeme stark variiert, bestehen diese im Wesentlichen aus einem oder mehreren Sensoren zur Erfassung von Messgrößen, einem Bauteil zur Informationsverarbeitung (Mikrocontroller) und einem Funktransceiver. Jeder Sensorknoten kann somit als ein Sensor-Aktor-System aufgefasst werden. Eine Unterscheidung erfolgt hier, unabhängig nach den Einsatzbereichen (Industrie 4.0, Smart-Home, Wearables), anhand der erfassten Messgrößen, also dem Einsatzzweck der Sensoren. Hier werden Umgebungssensoren und Körpersensoren unterschieden:

- **Umgebungssensoren** nehmen ihre Umgebung wahr und verfügen über Größen wie beispielsweise die Lufttemperatur und die Luftfeuchtigkeit

-
- **Körpersensoren** werden am Körper getragen und nehmen physiologische Größen auf, wie beispielsweise die Herzrate oder die Atemtätigkeit

Bedarfsgerechter Entwurf Ein bedarfsgerechter Entwurf ist bestrebt, die Effizienz des Systems zu maximieren, in dem ein Abgleich zwischen benötigter Leistungsfähigkeit und der dafür notwendigen Energieaufnahme erfolgt. Eingebettete Systeme, also informationsverarbeitende Systeme, die in Produkte eingebettet sind [125], werden hinsichtlich ihrer Leistungsfähigkeit häufig sehr großzügig dimensioniert, so dass große Leistungsreserven für die jeweilige Anwendung vorhanden sind. Auf diese Weise soll gewährleistet werden, dass die geforderte Anwendung sicher und mit ausreichenden Reserven ausgeführt werden kann. Jedoch führt dieser Ansatz häufig zu einem Entwurf von Systemen, die weit leistungsfähiger sind, als die Anwendung es erfordert. Dieses ist darauf zurück zu führen, dass Anforderung und Leistungsfähigkeit nicht hinreichend präzise oder nur mit nicht vertretbarem Aufwand quantifiziert werden können. Der Überschuss an Leistungsfähigkeit resultiert somit in einer gesteigerten Energieaufnahme. Der bedarfsgerechte Entwurf zielt somit auf eine möglichst präzise Anpassung der Leistungsfähigkeit an die Leistungsanforderung ab und führt schlussendlich zu einer für die spezifische Anwendung minimalen Leistungsaufnahme.

Modellgestützter Entwurf Ein modellgestützter Entwurf unterstützt bei der Dimensionierung des Systems, in dem das Verhalten der einzelnen Komponenten und somit auch des Gesamtsystems in einem Modell abgebildet wird. Das Modell sollte möglichst vollständig sein, so dass sämtliche Zustände abgebildet werden können. Somit kann eine Entwurfsraumexploration durchgeführt werden, die das Verhalten des Systems unter der Variation der relevanten Parameter untersucht. Entscheidend ist hier, dass auf Basis des Modells auch eine Simulation des Systems ermöglicht wird. Das steigende wissenschaftliche Interesse bezüglich des modellbasierten Entwurfs zeigt sich auch in der steigenden Anzahl der jährlichen Publikationen, die neben den Begriffen *Internet of Things* bzw. *IoT* auch den Begriff *Model* (Mod.) für sich oder in Kombination mit den Begriffen *Energy* bzw. *Power* in der Kurzzusammenfassung aufführen (vgl. Abbildung 1.1).

Ziel der Arbeit Ziel der Arbeit ist die Entwicklung einer Entwurfsmethodik für drahtlose Sensorknoten, welche sowohl die Energieeffizienz des Systems optimiert, als auch den Entwurfsprozess systematisiert. Dieses setzt ein parametrierbares und modulares Modell der Komponenten von drahtlosen Sensorknoten und deren Zusammenschluss zu einem Gesamtsystem voraus, so dass verschiedene Systemkonfigurationen simuliert und hinsichtlich ihrer Leistungsaufnahme und Leistungsfähigkeit verglichen werden können.

Die Arbeit gliedert sich wie folgt:

- **Kapitel 2** fasst den Stand der Technik bezüglich drahtloser Sensorknoten zusammen und gibt eine Übersicht der Komponenten eines drahtlosen Sensorknotens. Die am

Markt verfügbaren Systeme werden unterteilt nach Umgebungssensorknoten und Körpersensorknoten vorgestellt. Zudem werden typische Messgrößen und Parameter der Sensorknoten eingeführt.

- **Kapitel 3** stellt die Methodik und aktuelle Arbeiten in der Literatur zur Messung der Leistungsaufnahme von eingebetteten Systemen vor und verdeutlicht diese anhand von zwei Praxisbeispielen. Die praktische Vermessung von Sensorknoten umfasst dabei den kommerziellen *CC2560STK* Umgebungssensorknoten der Firma *Texas Instruments* und den Körpersensorknoten *BG-V4.2* der Arbeitsgruppe Kognitronik & Sensorik (AG K&S). Abschließend wird die Optimierung durch Vermessung diskutiert und deren Grenzen aufgezeigt.
- **Kapitel 4** beschreibt den Stand der Technik für die Modellierung der einzelnen Komponenten eines Sensorsystems. Diese umfassen die Komponenten Mikrocontroller, Sensoren, Funktransceiver, Speicher, Bauteile zur Indikation, Schnittstellen, Spannungswandler und Energiespeicher. Zudem wird ein Überblick über die Ansätze zur Modellierung des Gesamtsystems gegeben. Abschließend wird eine Zusammenfassung und Bewertung des Stands der Technik bezüglich der Modellierung drahtloser Sensorknoten gegeben.
- **Kapitel 5** beschreibt die entwickelte Entwurfsmethodik. Zunächst wird die Methodik in den bisherigen Entwurfsprozess eingeordnet. Anschließend wird, auf Basis des Stands der Technik drahtloser Sensorknoten, der derzeitigen Ansätze hinsichtlich Vermessung und Modellierung von Sensorknoten und dem Entwurfsprozess mikroelektronischer Systeme, der Handlungsbedarfs abgeleitet. Auf Basis des identifizierten Handlungsbedarfs wird im Folgenden das entwickelte Modell beschrieben. Es werden zunächst das grundlegende Systemmodell und das grundlegende Komponentenmodell beschrieben. Anschließend werden die spezifischen Modelle für die einzelnen Komponenten und deren Zusammenschluss zu einem Modell für das Gesamtsystem beschrieben. Diese bilden die Basis für die abschließende Abbildung des Systemverhaltens.
- **Kapitel 6** beschreibt exemplarisch den modellgestützten Entwurf anhand des in Kapitel 5 aufgestellten Systemmodells für einen multimodalen Umgebungssensorknoten. Es wird aufgezeigt, wie der Entwurfsraum in jedem Schritt eingeschränkt und schlussendlich auf einen kleinen Lösungsraum fokussiert wird. Abschließend erfolgt eine Bewertung des Entwurfs, so dass das Potential der Methodik verdeutlicht wird.
- **Kapitel 7** fasst die Arbeit zusammen, gleicht die umgesetzte modellbasierte Entwurfsmethodik mit dem identifizierten Handlungsbedarf ab und gibt einen Ausblick.

2 Drahtlose Sensorknoten

Drahtlose Sensorknoten können im Allgemeinen als Sensor-Aktor-Systeme aufgefasst werden, da die im System enthaltenen Sensoren spezifische Messgrößen aufnehmen, diese mittels einer Einheit zur Informationsverarbeitung verarbeitet werden und anschließend Einfluss auf einen Aktor nehmen. Im Folgenden werden zunächst die Systemstruktur eines drahtlosen Sensorknotens und dessen typische Komponenten eingeführt. Im Anschluss wird auf die erfassten Messgrößen und berechneten Parameter eingegangen. Abschließend wird ein Überblick über die am Markt verfügbaren Systeme gegeben.

2.1 Systemstruktur

Eine schematische Darstellung der Architektur eines drahtlosen Sensorknotens ist in Abbildung 2.1 dargestellt. Die grundlegenden aktiven Komponenten sind Sensoren, eine Einheit zur Informationsverarbeitung und Aktoren. Zur Informationsverarbeitung wird typischerweise ein Mikrocontroller verwendet. Als Aktoren können Komponenten wie ein Funktransceiver, ein Speicher oder Bauteile zur Indikation enthalten sein. Zur Verbindung dieser Komponenten werden verschiedene Typen von Schnittstellen verwendet. Die Energieversorgung eines solchen Systems besteht typischerweise aus einem Spannungswandler und einem Energiespeicher.

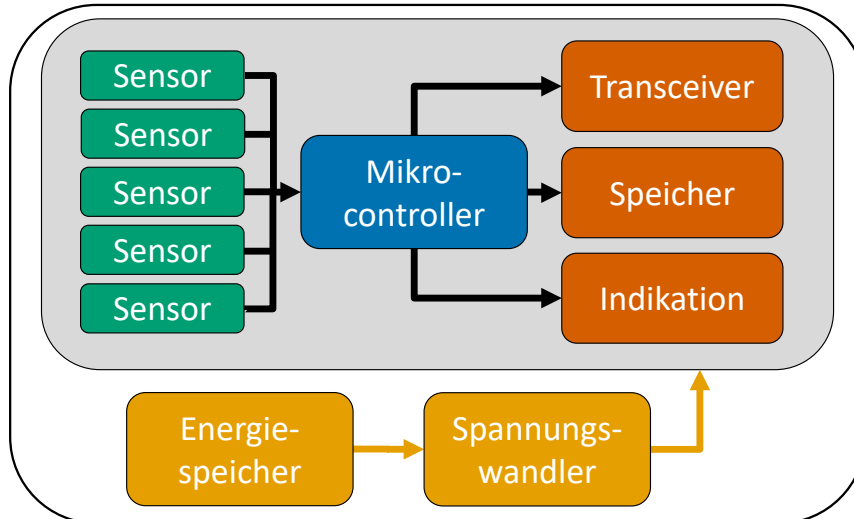


Abbildung 2.1.: Schematische Darstellung der Komponenten eines drahtlosen Sensorknotens

2.2 Komponenten

Im Folgenden werden die verschiedenen Komponenten exemplarisch vorgestellt und somit ein Überblick über die möglichen Ausprägungen eines drahtlosen Sensorknotens gegeben.

2.2.1 Sensoren

Sensoren sind spezielle Elektronikbausteine, die ihre Umgebung wahrnehmen und dabei beispielsweise physikalische, chemische, biologische und physiologische Größen in elektrische Signale umwandeln [217]. Die Eingruppierung eines drahtlosen Sensorknotens wird häufig anhand der zu erfassenden Messgrößen und somit anhand der enthaltenen Sensoren vorgenommen. Hier kann zwischen Umgebungssensorknoten und Körpersensorknoten unterschieden werden. So wird der Körpersensor direkt am Körper getragen und kann sowohl physiologische (z.B. Herzrate), als auch biokinematische Messgrößen und Parameter (z.B. Postur) der nutzenden Person erfassen. Die Daten sind somit personenbezogen. Ein Umgebungssensor hingegen wird in der zu erfassenden Umgebung installiert und nimmt Größen wie die Lufttemperatur und die Luftfeuchtigkeit auf.

Eine detaillierte Betrachtung der gebräuchlichen Sensoren, der zu erfassenden Messgrößen und der zu berechnenden Parameter erfolgt in Abschnitt 2.3.

2.2.2 Informationsverarbeitung

Die Einheit zur Informationsverarbeitung besteht auf einem drahtlosen Sensorknoten in den meisten Fällen aus einem Mikrocontroller. Diese Recheneinheit verfügt, im Gegensatz zu einem Mikroprozessor, nur über einen eingeschränkten Instruktionssatz, welcher jedoch für die auszuführenden Kontrollaufgaben ausreichend ist. Zudem verfügt ein Mikrocontroller bereits über internen Speicher, so dass dieser nicht mehr extern angebunden werden muss. Eine Übersicht der Bestandteile eines Mikrocontrollers wird von Gridling und Weiss [72] gegeben. So sind neben dem eigentlichen Prozessorkern und dem benötigten Speicher für Daten und Instruktionen auch direkt die Module der Schnittstellen mit integriert. Hier finden sich neben den digitalen I/O auch serielle Schnittstellen (z.B. I2C, SPI, UART) und analoge Schnittstellen (*Analog Digital Converter* (ADC), *Digital Analog Converter* (DAC)). Somit kann ein solches System kompakt aufgebaut und dementsprechend in vielfältige Objekte integriert werden. Der Markt verfügbarer Mikrocontroller ist sehr breit, so dass ein geeignetes Modell entsprechend der Anwendung ausgewählt werden kann. Vorherrschend im Bereich der eingebetteten Systeme sind die *Cortex* Modelle der Firma *ARM*, wobei verschiedene Typen (z.B. *M0*, *M3*, *M4*) angeboten werden [35]. Diese unterscheiden sich in ihrer Leistungsfähigkeit, Speichergröße, Art und Anzahl der Schnittstellen und Größe des Gehäuses. Weitere ausschlaggebende Kriterien für die Auswahl eines Mikrocontrollers sind Programmierbarkeit, Verfügbarkeit und nicht zuletzt der Preis.

2.2.3 Aktoren

Anders als in beispielsweise technischen Systemen der Automatisierungstechnik ist ein Aktor in einem drahtlosen Sensorknoten nicht zwangsläufig als Antrieb zu verstehen. Vielmehr kann ein Aktor hier ein Funktransceiver, ein Datenspeicher oder auch ein Bauteil zur Indikation sein.

2.2.3.1 Funktransceiver

Ein Funktransceiver ist bereits laut Definition zwingender Bestandteil eines drahtlosen Sensorknotens, wobei der Markt der verfügbaren Funktechnologien sehr vielfältig ist. Von den Standards, welche häufig für drahtlose Sensorknoten verwendet werden, basieren einige auf dem Standard 802[®] [88] des *Institute of Electrical and Electronics Engineering* (IEEE) Berufsverbandes. Eine Übersicht der Funkstandards wurde zum Teil in einer betreuten Masterarbeit [275] recherchiert und wird hier zusammengefasst.

- Der auch als **Wireless Local Area Network (WLAN)** bezeichnete IEEE Standard 802.11[®] [89] hat durch die persönlichen Heimnetzwerke und die Verwendung in mobilen Endgeräten wie Smartphones eine weite Verbreitung erfahren. Neben der großen Interoperabilität sind vergleichsweise hohe Reichweite und hoher Durchsatz die Vorteile der Technologie. Jedoch ist auch der Ressourcenbedarf entsprechend hoch.
- Der **Bluetooth** Standard (IEEE 802.15.1[™] [87]) ermöglicht eine kabellose Datenübertragung zwischen Geräten über kurze Distanzen. Bereits 2009 wurde die Bluetooth v3.0+HS Spezifikation vorgestellt, welche eine theoretische Datenrate von 24 Mbit/s bietet und bis heute Anwendung in Autos, Bluetooth Headsets und weiteren Geräten findet, die von einer hohen Datenrate, z.B. zur Übertragung von Audiosignalen, profitieren. Die Reichweite von Bluetooth beträgt ca. 100 m.
- Der neuere **Bluetooth Low Energy (BLE)** Standard ist eine Funktechnologie für kurze Distanzen. Im Gegensatz zu klassischem Bluetooth wurde BLE als **Ultra Low Power (ULP)** Lösung entwickelt, da es hauptsächlich für einfache Kontrollanwendungen genutzt werden soll und daher der periphere BLE-Sender sehr energieeffizient arbeiten muss. Die Reichweite beträgt 50 m und der maximale, theoretische Datendurchsatz 1 Mbit/s [82].
- Die 2004 von der Firma Dynastream entwickelte, proprietäre Funktechnologie **ANT** [54] und dessen 2010 vorgestellte Erweiterung ANT+ arbeitet im 2,4 GHz *Industrial Scientific Medical (Band)* (ISM). Die Reichweite beträgt ca. 30 m, der theoretische Datendurchsatz liegt bei 1 Mbit/s. ANT wurde von der Firma Garmin¹ mitentwickelt, so dass das primäre Ziel dieser Funktechnologie im Sport- und Fitnessbereich liegt.

¹Garmin Deutschland GmbH: <https://www.garmin.com/de-DE>

- Eine weitere ULP Funktechnologie stellt **ZigBee** dar, das auf dem IEEE Standard 802.15.4™ [86] basiert. Diese Funktechnologie wurde für den Zweck der Heimautomatisierung entworfen. Die maximale Reichweite beträgt 100 m und der theoretische maximale Datendurchsatz liegt bei 250 kBit/s.
- Das **IPv6 over Low power Wireless Personal Area Network (6LoWPAN)** Protokoll basiert auf dem IEEE 802.15.4™ Standard [86] und definiert eine reduzierte, IP-basierte Kommunikation für drahtlose Sensorknoten. Zu Adaption des IP-Protokollstapels wurde dieser hinsichtlich Größe des Codes, Komplexität, Konfigurationsmöglichkeiten und Overhead an die beschränkten Ressourcen eines Mikrocontrollers angepasst [135].
- Die **Z-Wave** Technologie² wurde für einfache Kontrollaufgaben [233] in der Heimautomatisierung entwickelt und verfügt dort über eine entsprechende Verbreitung. Die Z-Wave Technologie verfügt über eine maximale Reichweite von 30 m und einen maximalen Durchsatz von 40 kbit/s.
- Zusätzlich setzen manche Hersteller auf **proprietäre Protokolle**, um sich durch ein geschlossenes System von der Konkurrenz abzugrenzen. Ein Beispiel ist die pulskodierte Funkübertragung im 5 kHz Bereich, die Polar in manchen Modellen [160] zur Übertragung der Herzfrequenz verwendet. Jedoch setzt auch Polar in aktuellen Modellen [158] zunehmend auf die Funktechnologie BLE.

2.2.3.2 Speicher

Für den Datenspeichers eines drahtlosen Sensorknotens werden nicht-flüchtige Speicher benötigt, die ihre Daten auch ohne Spannungsversorgung halten. Die Anforderung einer hohen Flexibilität (feingranulares Beschreiben und Löschen) bei gleichzeitig niedrigen Kosten (geringe Größe der Speicherzelle) wird derzeit vom Flash-Speicher erfüllt, da dieser vielfach elektrisch schreib- und löschar ist und nur einen Transistor pro Speicherzelle benötigt [20]. Als Standard haben sich hier die beiden Typen NAND- und NOR-Flash etabliert. NOR-Flash verfügt über ein einfaches paralleles Interface, weist gute Lesegeschwindigkeit und ist auch als Code-Speicher geeignet. Jedoch weist NAND-Flash eine höhere Dichte auf und ist somit auch bei höheren Kapazitäten kosteneffizient. Zusätzlich können NAND-Flash-Speicher schneller beschrieben und gelöscht werden und eignen sich somit als Massenspeicher [195]. Aufgrund des Kosteneffizienz findet NAND-Flash zunehmend Verbreitung in eingebetteten Systemen [152].

2.2.3.3 Indikation

Da oftmals eine direkt Ablesbarkeit des Systemzustandes gewünscht ist, wird oftmals mindestens ein Bauteil zur Indikation vorgesehen, wobei hier verschiedene Technologien zur Verfügung stehen, die hier kurz eingeführt werden.

²Z-Wave Alliance: <http://z-wavealliance.org/>

- Die Integration einer **Light Emitting Diode (LED)** in einen Sensorknoten ist eine einfache und kostengünstige Möglichkeit verschiedene Betriebszustände anzuzeigen. Entweder wird diese fest an die Spannungsversorgung angeschlossen, um anzuzeigen, dass der Knoten eingeschaltet ist, oder die LED wird mit einem IO-Pin des Mikrocontrollers verbunden. Auf diese Weise können verschiedene Ereignisse dynamisch angezeigt werden. Mit der Verwendung einer **Red Green Blue (RGB) LED**, welche drei LEDs in einem Gehäuse vereint, können zusätzlich viele verschiedene Farbabstufungen angezeigt werden. Nachteile einer Indikation mittels LED ist die hohe Leistungsaufnahme und möglicherweise eine Verdeckung der LED, so dass diese für die nutzende Person nicht sichtbar ist.
- Die bereits in am Körper getragenen Geräten wie Smartphones üblichen **Vibrationsmotoren** [162] bestehen aus einem kleinen Gleichstrommotor mit einer Unwucht auf der Antriebsachse und sind somit in der Lage, eine direkte haptische Rückmeldung an die nutzende Person zu geben. Somit kann auch dann eine Rückmeldung gegeben werden, wenn das System für die nutzende Person nicht sichtbar ist. Potentielle Nachteile sind eine hohe Leistungsaufnahme, notwendige Zusatzbeschaltung und eine Beeinträchtigung von Messungen durch die Vibrationen.
- Bei einem **elektromagnetisch-akustischen Wandler** wird durch das Anlegen einer Wechselspannung an eine Spule ein wechselndes Magnetfeld erzeugt, welches wiederum eine Membran zum Schwingen anregt [147]. Der so erzeugte Ton weist die selbe Frequenz auf wie das Eingangssignal. Somit wird eine direkte akustische Rückmeldung an die nutzende Person ermöglicht. Nachteile sind die potentiell hohe Leistungsaufnahme, notwendige Zusatzbeschaltung und der Einfluss von erzeugtem Magnetfeld und Schalldruck auf das System.
- Ein **piezoelektrisches Element** wird durch das Anlegen einer Wechselspannung in Schwingung versetzt, wodurch ein Ton erzeugt wird [36]. Im Vergleich zu einem elektromagnetisch-akustischen Wandler weist der Piezo-Summer eine geringere Leistungsaufnahme auf und arbeitet in einem höherem Frequenzbereich.

2.2.4 Spannungswandler

Der Spannungswandler eines drahtlosen Sensorknotens stellt sicher, dass die zuvor vorgestellten Komponenten mit der korrekten Spannung versorgt werden. Gegebenenfalls können somit auch mehrere Spannungswandler auf einem Sensorknoten notwendig sein, so dass verschiedene Spannungsdomänen versorgt werden können. Die Eingangsspannung wird vom Energiespeicher des Systems zur Verfügung gestellt. Grundlegend sind zwei verschiedene Typen von Spannungswandlern zu unterscheiden [225]:

- Ein **Linearregler** arbeitet nach dem Prinzip eines Spannungsteiler und liefert eine rauscharme Ausgangsspannung. Jedoch fällt die Differenz zwischen Eingangs- und

Ausgangsspannung über einen ohmschen Widerstand ab und wird in Wärme umgesetzt. Somit wird im Betrieb nur eine niedrige Effizienz erreicht. Zudem muss die Eingangsspannung immer größer als die Ausgangsspannung sein. Vorteile liegen in den geringen Kosten und der kompakten Bauform, da keine Induktivitäten oder Kapazitäten benötigt werden.

- Ein **Schaltregler** hingegen lädt periodisch eine Kapazität, Induktivität oder eine Kombination aus beiden Elementen. Die Ausgangsspannung kann entweder niedriger (Abwärtswandler, engl.: *Buck-Converter*) oder höher (Aufwärtswandler, engl.: *Boost-Converter*) sein als die Eingangsspannung. Verglichen mit einem Linearregler ist die Effizienz deutlich höher und liegt theoretisch bei nahezu 100 %. Jedoch resultiert aus den Schaltvorgängen eine vergleichsweise rauschbehaftete Ausgangsspannung (*voltage ripple*).

2.2.5 Energiespeicher

Die Wahl des Energiespeicher ist bei drahtlosen Sensorknoten von entscheidender Bedeutung, da dieser bezüglich des Volumens häufig den größten Anteil am Gesamtsystem hat und dessen Kapazität die Laufzeit des Systems bestimmt. Es kommt somit zu einem Zielkonflikt zwischen Volumen und Kapazität des Energiespeichers.

Ein Überblick zum Stand der Technik wird von Linden und Reddy [118] gegeben. So ist ein Energiespeicher laut Definition eine Zelle, die gespeicherte chemische Energie mittels einer elektrochemischen Reaktion in elektrische Energie umwandelt und diese somit zur Verfügung stellt. Eine Batterie besteht aus mehreren dieser Zellen, die entweder seriell oder parallel verschaltet sind. Jedoch hat sich auch für eine einzelne Zelle der Begriff Batterie durchgesetzt. Der generelle Aufbau einer Zelle besteht aus der negativ geladenen Elektrode (Anode) und einer positiv geladenen Elektrode (Kathode). Zwischen den Elektroden befindet sich ein Separator, welcher die Elektroden mechanisch voneinander trennt und ein Elektrolyt, welches als Medium für den Ladungstransport zwischen beiden Elektroden dient. Der Separator ist für das Elektrolyt durchlässig. Bei der Entladung wandern Lithium-Ionen von der Anode zur Kathode und der positive Strom in der versorgten Schaltung fließt in entgegengesetzter Richtung. Das Laden der Zelle kehrt diese Richtung um [153]. Kann die Zelle nicht wieder aufgeladen werden, wird sie als Primärzelle bezeichnet. Wiederaufladbare Zellen werden als Sekundärzellen oder auch Akkumulator (kurz: Akku) bezeichnet. Die folgende Auflistung gibt einen Überblick über die wichtigsten Eigenschaften einer Zelle [14]:

- Kapazität C_{BAT} in Wh
- Ladezustand in % (engl.: *State-of-Charge* (SOC))
- Volumen in l
- Masse in kg
- Leerlaufspannung U_{OC} in V (engl.: *open circuit voltage*)

- Klemmenspannung U_{BAT} in V (engl.: *terminal voltage*)
- Entladeschlussspannung U_{CUT} in V (engl.: *cut-off voltage*)
- Ladeschlussspannung in V
- Innenwiderstand R_0 in Ω
- Gravimetrische Energiedichte in Wh/kg
- Volumetrische Energiedichte in Wh/l
- Lade- und Entladerate in Vielfachem der Kapazität
- Lade- und Entladeleistungsdichte in W/kg
- Zyklusanzahl
- Langlebigkeit in % der Nennkapazität nach N Ladezyklen
- Effizienz als Anteil der Wärmeverluste in %
- Kosten in €/Wh

Die Laufzeit der Batterie ist eine super-lineare Funktion des Entladestroms, wobei der Betrag und die Verteilung der Stromentnahme entscheidend sind. Die maximale Laufzeit wird bei minimaler Varianz der Entladestromverteilung erreicht [153]. Die verwendeten Materialien von Anode, Kathode, Elektrolyt und Separator bestimmen grundlegend die Eigenschaften der jeweiligen Batterie. Neben der Auswahl der Technologie ist für den Einsatz in drahtlosen Sensorknoten auch die Bauform ein wichtiges Kriterium. So sind bereits flexible Batterien am Markt erhältlich [34], die beispielsweise in Armbänder integriert werden können. Jedoch weisen diese nur eine geringe Kapazität und einen vergleichsweise hohen Innenwiderstand auf, so dass deren Einsatzzweck zur Zeit noch begrenzt ist. Neben der Weiterentwicklung flexibler Batterien [73] betreffen die neuesten Entwicklungen im Bereich der Energiespeicher den 3D-Druck von Lithium-Ionen Batterien [193] und die textile Fertigung [140].

2.2.6 System on a Chip

Der Trend zu einer Integration mehrerer Elemente in einer Komponente geht mittlerweile über den reinen Mikrocontroller mit Prozessorkern, Speicher, Schnittstellen hinaus. Bedingt durch den Fortschritt in der Mikroelektronik und der daraus resultierenden Miniaturisierung gehen viele Hersteller dazu über, weitere Funktionalitäten auf einem Halbleiter-Substrat (*Chip*) zu vereinen. Hier spricht man von einem *System-on-a-Chip* (SoC). Zwar werden die Chips immer spezialisierter, jedoch kann auf diese Weise eine Reduktion der Größe und der Kosten erreicht werden. Aktuelle Beispiele finden sich in der Kombination aus Mikrocontroller und Funktransceiver. Während der zusätzlich integrierte Mikrocontroller zunächst nur den Protokoll-Stack beinhaltete und nur sehr eingeschränkt programmiert werden konnte [22], verfügen aktuelle Modelle über einen sehr leistungsfähigen Mikrocontroller, welcher frei programmiert werden kann [139].

2.3 Messgrößen und Parameter

Das ausschlaggebende Merkmal drahtloser Sensorknoten ist die Art der erfassten Daten, so dass die Sensormodule anhand ihrer enthaltenden Sensoren klassifiziert werden können.

Im Folgenden wird ein Überblick über die Messgrößen und Parameter von drahtlosen Sensorknoten gegeben. Der Begriff Messgröße bezeichnet hier eine Größe, die von einem Sensor direkt erfasst und ausgegeben werden kann. Dem gegenüber können die sogenannten Parameter nicht direkt gemessen werden, sondern werden auf Basis der erfassten Messgrößen berechnet. Je nach Komplexität werden die Parameter direkt auf dem Sensorknoten oder auf einem externen System berechnet. Die Einführung der Messgrößen und Parameter erfolgt entsprechend der Definition von Christ [49] in den Kategorien der physiologischen und biokinematischen Größen. Diese Größe nehmen Informationen über die nutzende Person auf und sind somit personenspezifisch. Zusätzlich wird die Kategorie der Umgebungsgrößen betrachtet. Abschließend wird ein Überblick über komplexe Parameter gegeben, welche auf der Fusion von mehreren Parametern basieren, also auf einer Vielzahl von Messgrößen und Sensoren. Die Berechnung dieser Parameter wird als Sensorfusion bezeichnet.

2.3.1 Physiologische Größen

Physiologische Messgrößen und Parameter beziehen sich auf die Erfassung der physiologischen Vorgänge im menschlichen Körper [49]. Die wichtigsten Größen in Bezug auf die Erfassung mit drahtlosen Sensorknoten werden im Folgenden eingeführt.

- Eine wichtige Größe für die **Herzaktivität** ist das Elektrokardiogramm (EKG), welches die Aufzeichnung der elektrischen Aktivität des Herzens bezeichnet. Das wichtigste Merkmal für nicht-klinische Anwendungen ist der QRS-Komplex, welcher die Erregungsausbreitung in den Herzkammern darstellt [108]. Besonders die R-Zacke wird, aufgrund ihrer vergleichsweise großen Amplitude, für die Berechnung der physiologischen Parameter Herzrate (HR) und Herzratenvariabilität (HRV) genutzt. Die HR wird aus der Häufigkeit der R-Zacken in einem bestimmten Zeitabschnitt ermittelt und in Herzschlägen pro Minute angegeben. Die HRV gibt die Variation der zeitlichen Abstände zwischen aufeinander folgenden R-Zacken an und kann sowohl im Zeit- als auch im Frequenzbereich berechnet werden [123]. Der Vergleich mit Normwerten [141] erlaubt Rückschlüsse auf die Aktivität des autonomen Nervensystems und somit auf den pathologischen Zustand der nutzenden Person [291]. Eine weitere Methode die Aktivität des Herzens zu messen stellt die Photoplethysmographie (PPG) dar. Mit dieser Methode wird die Haut an einer geeigneten Stelle (z.B. Ohr-läppchen, Finger, Handgelenk) mittels einer LED durchleuchtet und das reflektierte Licht mittels eines Photodetektors ausgewertet [6]. Bei Verwendung von LEDs mit verschiedenen Wellenlängen kann, neben der HR, auch auf die Sauerstoffsättigung (SpO_2) des Blutes geschlossen werden.
- Eine weitere physiologische Größe ist die **Atmung** der nutzenden Person. Zur Erfassung der Atemtätigkeit existieren zahlreiche Methoden, wobei zwischen direkten und indirekten Verfahren unterschieden wird. Bei der direkten Messung wird der

orale und nasale Luftstrom der Atmung über die Strömung (Anemometer), Druckschwankungen, Lufttemperatur, Luftfeuchtigkeit oder den Gehalt an Kohlendioxid gemessen [143]. Mit diesen Verfahren kann, neben der Atemfrequenz, auch häufig das Atemzugvolumen bestimmt werden. Indirekte Methoden nähern den komplexen physiologischen Atemvorgang mit Hilfe der Volumenänderung in Brustkorb und Abdomen an [109]. Die aus der Atmung resultierende Bewegung kann beispielsweise mittels Respiratorische induktive Plethysmografie (RIP), Impedance Plethysmography (IPG), Magnetfeldmessung, Dehnungsmessstreifen oder Kraftsensoren erfasst werden [143]. Zusätzlich kann die Atmung, bei geeignetem Messaufbau, auch aus der Analyse des EKG oder dem PPG Signalverlauf [128] abgeleitet werden. Bei den indirekten Verfahren kann in den meisten Fällen nur die Atemfrequenz sicher bestimmt werden. Die Erfassung des Atemzugvolumens mit indirekten Verfahren ist derzeit noch Gegenstand der Forschung [117]. Während die direkten Verfahren häufig in klinischen Anwendungen oder der stationären Leistungsdiagnostik angewendet werden, finden die indirekten Verfahren zunehmend Verwendung in mobilen Systemen. Von Mitchell u. a. [132] wurde ein textiles piezoresistives Material in ein T-Shirt integriert, so dass aus der Bewegung von Brustkorb und Abdomen die Atemfrequenz bestimmt werden kann. Das System von Moraveji u. a. [133] verwendet zu dem gleichen Zweck einen Dehnungssensor. Lücken u. a. [121], Ramos-Garcia u. a. [167] und Tobola u. a. [210–212] benutzen ein leitfähiges Garn. Die Eigenentwicklung der AG K&S verwendet einen Kraftsensor [253].

Weitere physiologische Größen und deren Messverfahren sind die Körper- bzw. Hauttemperatur (mittels Temperatursensor), der Blutdruck (mittels Blutdruckmanschette am Arm), Schweißbildung (Elektrodermale Aktivität (EDA) oder Durchflussmessung [10]), Blutzucker (mittels Blutentnahme und Verwendung von Messstreifen), Muskelaktivität mittels Elektromyogramm (EMG) (Elektroden auf der Haut) und die Hirnaktivität mittels Elektroenzephalogramm (EEG) (Elektroden auf der Kopfhaut) [151].

2.3.2 Biokinematische Größen

Biokinematische Parameter beschreiben laut Definition von Christ [49] die ausgeführten Körperbewegungen (Biokinematik). Zu Erfassung der zugrunde liegenden Messgrößen eignen sich die sogenannten Inertialsensoren, welche ihre Eigenbewegung in Bezug auf ein Referenzsystem wahrnehmen. Zu der Klasse der Inertialsensoren zählen der Beschleunigungssensor und der Drehratensensor, welche im Folgenden, wie bereits in [79] beschrieben, vorgestellt werden. Darüber hinaus wird auch der Magnetfeldsensor eingeführt, welcher in einigen Anwendungen Stützinformationen liefert.

- Ein **Beschleunigungssensor** ist in der Lage, die lineare Beschleunigung auf einer Achse zu messen, wobei die Maßeinheit m/s^2 beträgt. Beschleunigungssensoren können kostengünstig in der Mikroelektromechanische Systeme (MEMS)-Technologie [232]

gefertigt werden und basieren auf einem Feder-Masse System. Heutige Sensoren vereinen bereits drei sensitive und orthogonal zueinander angeordnete Achsen (xyz) in einem Gehäuse [177].

- Ein **Drehratensensor**, auch Gyroskop genannt, ist in der Lage, die Drehgeschwindigkeit auf einer Achse in $^{\circ}/s$ zu messen. Da ein solcher Sensor auch in der MEMS-Technologie gefertigt werden kann, ist es möglich, mechanische und elektrische Komponenten kostengünstig direkt auf der Chipfläche zu fertigen. Derzeitig verfügbare und kostengünstige MEMS Gyroskope basieren auf dem Prinzip einer vibrierenden Masse, welche von elektrostatischen Aktuatoren zum Schwingen angeregt wird. Auch hier sind drei sensitive Achsen in einem Gehäuse erhältlich [95].
- Ein **Magnetfeldsensor** arbeitet oftmals nach dem Prinzip der anisotropen Magnetoresistenz, welche in eisenhaltigen Materialien auftritt [190]. Dieses bedeutet, dass sich der Widerstand eines solchen Materials ändert, wenn ein Magnetfeld angelegt wird, welches senkrecht zum Stromfluss steht. Dieser Effekt kann zur Orientierungsbestimmung genutzt werden, wenn vier solcher Elemente in Form einer Wheatstoneschen Brücke angeordnet werden. Werden drei solcher Brücken orthogonal zueinander angeordnet, ist es möglich, das Magnetfeld für alle drei Raumachsen zu bestimmen.

Bereits auf Basis der erfassten Beschleunigungen können zahlreiche biokinematische Parameter berechnet werden. So kann bereits der Betrag des Beschleunigungsvektors Informationen darüber liefern, ob die nutzende Person des Sensorknotens körperlich aktiv ist. Darüber hinaus kann mit Hilfe von einfachen trigonometrischen Funktionen die Oberkörperneigung [164] der nutzenden Person bestimmt werden und auch eine Berechnung der Anzahl zurückgelegter Schritte ist möglich [238]. Komplexere Parameter werden häufig aus der Fusion mehrerer Messgrößen berechnet und in Unterabschnitt 2.3.4 vorgestellt.

2.3.3 Umgebungsgrößen

Die Umgebungsgrößen bezeichnen Informationen, welche drahtlose Sensorknoten über ihre Umgebung sammeln. Anders als die physiologischen und biokinematischen Größen beziehen sich diese Größen nicht auf die nutzende Person und sind somit nicht personenspezifisch. Im Folgenden wird ein Überblick der Umgebungsgrößen und -parameter gegeben.

- Zur Messung der **Umgebungstemperatur** können verschiedene Typen von Temperatursensoren verwendet werden. Üblich sind beispielsweise einfache Schaltungen mit einem *Negative Temperature Coefficient* (NTC) Thermistor [65, S.532] oder berührungslose Verfahren mittels Pyrometer [65, S.481]. Auch sind integrierte digitale Sensoren [183] erhältlich, die beispielsweise nach dem Prinzip eines

Silizium-Bandlücken-Thermometers arbeiten und in MEMS-Technologie gefertigt werden [185].

- Die relative **Luftfeuchtigkeit** bezeichnet den Wassergehalt der Umgebungsluft und kann mit einem Hygrometer gemessen werden [65, S.445]. Auch hier sind integrierte Lösungen am Markt erhältlich [184].
- Die **Luftqualität** wird maßgeblich von dem Gehalt an Kohlenstoffdioxid bestimmt (CO_2). Zur Messung der Konzentration können Festelektrolyt-Sensoren [74] oder nichtdispersive Infrarotsensoren verwendet werden. Ein nichtdispersiver Infrarotsensor durchleuchtet das zu untersuchende Gas und bestimmt die Gaskonzentration auf Basis der absorbierten spezifischen Wellenlängen [150]. Dieses Messprinzip liefert sehr präzise Ergebnisse und wird zumeist unter Laborbedingungen angewendet. Jedoch sind auch hier erste kompakte Sensoren erhältlich [8].
- Der barometrische **Luftdruck** kann mit einem piezoresistiven Sensor gemessen werden, wobei die Kraft, die auf eine Membran mit bekannter Fläche wirkt, gemessen wird [65, S.381]. Kommerzielle Lösungen sind mit digitaler Schnittstelle und in kleiner Bauform erhältlich [30].
- Die **Helligkeit** der Umgebung wird über die Intensität des einstrahlendes Lichts bestimmt und beispielsweise mit Hilfe von Phototransistoren in elektrische Signale umgewandelt [65, S.471]. Auch hier sind kompakte integrierte Lösungen erhältlich [202].
- Der in der Umgebung herrschende **Geräuschpegel** kann mit einem Mikrofon aufgenommen werden. Es werden dabei die Druckwellen des Schalls in elektrische Signale gewandelt [65, S.341]. Mittlerweile können auch Mikrofone in MEMS Technologie gefertigt werden und sind somit in kompakter Bauform erhältlich [94].

Die Umgebungsgrößen sind für sich genommen einfache und alltägliche Informationen. Jedoch kann vor allem aus der Kombination mehrerer Größen die Belastung des Menschen durch äußere Einflüsse ermittelt werden. So hat beispielsweise die Kombination aus Lufttemperatur und Luftfeuchtigkeit einen großen Einfluss auf das Wohlbefinden eines Menschen und auch erhöhte Werte des Geräuschpegels können schnell zu Unwohlsein bzw. Stress führen. Auch die Luftqualität ist entscheidend, wobei eine stark erhöhte Konzentration an CO_2 in der Atemluft lebensbedrohlich sein kann. Da die Anwesenheit von Menschen in einem Raum die Umgebungsgrößen beeinflussen kann, werden diese auch häufig für Präsenzmelder verwendet. Bei einer Vernetzung der Geräte und Analyse der gespeicherten Daten wie der Lufttemperatur und der Luftfeuchtigkeit kann zudem auf die Anwesenheit und das Verhalten von Personen geschlossen werden [134].

2.3.4 Sensorfusion

Durch die Fusion mehrerer Messgrößen verschiedener Kategorien zu komplexen Parametern können drahtlose Sensorknoten in vielfältigen Anwendungen zum Einsatz kommen.

- Frühere Ansätze zur Erfassung der physischen **Aktivität** finden sich in [43], wobei hier die Herzrate und das Beschleunigungssignal einbezogen wurden. Aktuelle Arbeiten zu diesem Thema finden sich detailliert in [255] bzw. [49] und wurden bereits in [252] zusammenfassend publiziert.
- Im Rahmen eines Forschungsprojektes konnte eine multimodale **Stresserkennung** auf Basis von maschinellen Lernverfahren realisiert werden, wobei die verwendeten Features auf den Parametern HR, HRV, EDA und den Eingaben der nutzenden Person basieren. Die Entwicklung dieser Stresserkennung für den Sensorknoten wurde in [258] und [257] publiziert.
- Die Entwicklung eines inertialen **Navigationssystem** (engl: *Inertial Measurement Unit* (IMU)) zur Bestimmung der Position und der Orientierung eines Systems musste 2010 noch mit diskreten Bauteilen [80] erfolgen, wobei die inertielle Sensorfusion zur Berechnung der Orientierung des Systems auf einem dedizierten Mikrocontroller [79] implementiert werden musste. Mittlerweile sind die drei Sensortypen Beschleunigungssensor, Drehratensensor und Magnetfeldsensor mit jeweils drei sensitiven Achsen in einem Gehäuse erhältlich. Durch die Integration eines Mikrocontrollers zusammen mit den Sensoren zu einem SoC muss entsprechend auch kein zusätzlicher Mikrocontroller vorgesehen werden. Die Fusion von Drehrate, Beschleunigung und den Daten des Magnetfeldsensors erfolgt bei aktuellen Modellen direkt auf dem Chip [31].
- Die **Bewegungserfassung** eines Menschen mit inertialen Sensoren (*Inertial Motion Capturing*) ist derzeit Gegenstand der Forschung [130] und wurde im Rahmen einer betreuten Masterarbeit [281] untersucht. Verwendet wurden die *BRIX₂* Module [234; 235] der Arbeitsgruppe *Ambient Intelligence*³, welche über eine *InvenSense MPU-9150* [96] IMU verfügen. Mit dem entwickelten inertialen *Motion Capturing* System konnte nachgewiesen werden, dass die aktuellen kostengünstigen Sensoren in der Lage sind, grundlegende menschliche Bewegungen zu erfassen. Jedoch konnte die Präzision des Referenzsystem noch nicht erreicht werden.

2.4 Verfügbare Systeme

Die derzeit am Markt verfügbaren oder in der Entwicklung befindlichen Systeme sind äußerst vielfältig, so dass für den folgenden Überblick der Systeme Unterkategorien

³<https://www.cit-ec.de/en/ami>

definiert wurden. Zunächst wird die Kategorie der Umgebungssensorknoten vorgestellt, welche entsprechend ihrer Definition Umgebungsgrößen aufnehmen. Die Kategorie der Körpersensorknoten, welche physiologische und biokinematische Größen aufnehmen, wurde weiter unterteilt, so dass hier zwischen Herzfrequenzsensorknoten, Armbändern und multimodalen Sensorknoten unterschieden wird. Zusammenfassend wird in Anhang A ein tabellarischer Überblick gegeben. Zu beachten ist, dass die Übersicht der Systeme aufgrund der enormen Vielfalt am Markt nicht vollständig ist, sondern exemplarisch einzelne System entsprechend ihrer Relevanz ausgewählt wurden.

2.4.1 Umgebungssensorknoten

Die am Markt erhältlichen Produkte zur Erfassung der Umgebung sind bisher vorwiegend kabelgebundene Systeme wie beispielsweise Luftgütesensoren [175] und Wetterstationen [137]. Jedoch befindet sich dieser Markt im Umbruch, so dass zunehmend drahtlose Systeme entwickelt werden. Erste kommerzielle drahtlose Umgebungssensorknoten sind die *Bluetooth Beacons*, welche zur Lokalisierung auf Basis der BLE-Funktechnologie genutzt werden können. Definiert wurde dieser Standard als *iBeacon* von der Firma *Apple* [11], wodurch sich eine entsprechende Anzahl verfügbarer Produkte wie die Modelle *Ruuvi-Tag* [241] und *Puck.js* [59] etabliert haben. Basis für zukünftige Sensorknoten sind die derzeit erhältlichen Evaluierungsplattformen verschiedener Unternehmen aus dem Bereich der Elektronikkomponenten. So bietet *Texas Instruments* mit dem *SimpleLink™ SensorTag* [203] eine Evaluierungsplattform für ihre aktuellen Funk-SoC und Sensoren an, wobei neben dem in Tabelle 2.1 aufgeführten *CC2650STK* zwei weitere Varianten mit anderer Funktechnologie angeboten werden (vgl. Tabelle A.4 in Anhang A). Eine weitere Kategorie sind die Forschungsplattformen. Die *TelosB* Plattform wurde ursprünglich von der *University of California, Berkeley*⁴ entwickelt und ist gemäß der *Open Source Initiative* frei verfügbar, wobei die Hardware auch kommerziell angeboten [4] wird. Auch die am *Cognitive Interaction Technology Excellence Cluster* (CITEC) entwickelte *BRIX₂* Plattform [234; 235] ist frei verfügbar und kann über zahlreiche Zusatzmodule erweitert werden.

⁴<http://www.berkeley.edu/>

2. Drahtlose Sensorknoten

Tabelle 2.1.: Auswahl verfügbarer Umgebungssensoren und deren Komponenten. Mit x markierte Positionen sagen aus, dass der entsprechende Sensor auf dem System zur Verfügung steht. Mit * markierte Positionen sind Komponenten, die über Zusatzmodule eingebunden werden können.

Hersteller Bezeichnung	TI CC2560STK	Ruuvi RuuviTag	Espruino Puck.js	Advanticsys TelosB	AG AMI BRIX ₂
Temperatur	x	x	-	x	*
Luftfeuchte	x	x	-	x	*
Druck	x	x	-	-	-
Lautstärke	x	-	-	-	-
Beschl.	x	x	-	-	x
Gyroskop	x	-	-	-	x
Magnetfeld	x	-	x	-	x
Helligkeit	x	-	-	x	*
Abstand	-	-	-	-	*
Mikrocon.	Cortex M3	Cortex M4F	Cortex M4F	MSP430	Atmega32U4
Funk	BLE 6LoWPAN ZigBee	BLE ANT 2,4 GHz	BLE ANT 2,4 GHz	802.15.4 ZigBee	868 MHz BLE*
Speicher	8Mbit Flash	-	-	8Mbit Flash	SD Karte*
Indikation	LED,Buzzer	LED	LED	LED	LED,Buzzer* Audio*,Vibr.*
Batterie	CR2032	CR2477	CR2032	2xAA	Akku (450 mAh)

2.4.2 Körpersensorknoten

Zur Übersicht und besseren Vergleichbarkeit der erhältlichen Körpersensoren wurden diese in verschiedene Kategorien eingeteilt. Die erste Kategorie sind die Fitnessarmbänder, welche bereits eine relativ hohe Verbreitung im Konsumentenmarkt haben. Die zweite Kategorie sind die Herzfrequenzsensoren, wobei hier bisher die Zielgruppe bei ambitionierten Sportlern liegt. Die dritte Kategorie sind die multimodalen Sensorknoten. Diese sind in ihrer Funktionalität den übrigen Kategorien überlegen und erfassen eine Vielzahl von physiologischen und biokinematischen Größen. Zu dieser Kategorie gehört auch der Körpersensorknoten *BG-V4.2* der AG K&S.

Armbänder In der Kategorie der kommerziellen Fitnessarmbänder sind zahlreiche sogenannte Aktivitätstracker erhältlich. Diese Geräte (vgl. Tabelle A.1 in Anhang A) verfügen in der einfachsten Variante nur über einen Beschleunigungssensor und einen Funktransceiver.

ver [64] und erfassen die zurückgelegten Schritte der nutzenden Person, so dass diese ein Maß für die tägliche Aktivität erhält. Fortgeschrittenere Modelle wie *Fitbit Alta HR* [63], *Nokia Steel HR* [138] (ehemals *Withings*) und *Xiaomi Mi Band 2* [230] erfassen zusätzlich die HR mittels Photoplethysmographie (PPG), welche in das Aktivitätsmaß und eine Analyse des Schlafs einfließt. Jedoch sind die von diesen Geräten berechneten Parameter häufig ungenau [61]. Aktuelle Systeme mit einem erhöhten Anspruch an die Validität der Daten sind das *Empatica E4 Wristband* [57] und das, noch in der Entwicklung befindliche, *Samsung SIMBAND* [179].

Herzfrequenzsensoren Die als Brustgurt getragenen Herzfrequenzsensoren sind vor allem im Bereich ambitionierter Sportler verbreitet und dienen der Überwachung der HR während des Trainings und im Wettkampf. Marktführer ist hier die Firma *Polar*, welche verschiedene Modelle anbietet. So verfügt der *Polar H7* [160], neben einem BLE-Funktransceiver, auch über eine Übertragung der Daten im 5 kHz Band, während der kompaktere *Polar H6* [159] nur über die BLE-Verbindung verfügt. Das neueste Produkt, der *Polar H10* [158], verfügt bei kompakter Bauform und erhöhter Laufzeit sowohl über BLE, als auch die 5 kHz Übertragung. Zusätzlich ist ein Speicher integriert, so dass die erfassten Daten gespeichert werden können und keine kontinuierliche Verbindung zu einem Empfänger (Smartwatch, Smartphone) mehr erforderlich ist. Das Modell *TICKRx* der Firma *Wahoo* [223] verfügt, neben einem Speicher, zusätzlich über einen Beschleunigungssensor, so dass neben der HR auch die Anzahl der Schritte und die Aktivität berechnet werden kann.

Tabelle 2.2.: Überblick der Körpersensoren: Herzfrequenzsensoren. Mit x markierte Positionen sagen aus, dass der entsprechende Sensor auf dem System zur Verfügung steht.

Hersteller Bezeichnung	Polar H6	Polar H7	Polar H10	Wahoo TICKRx
EKG	1-Kanal	1-Kanal	1-Kanal	1-Kanal
Beschleunigung	-	-	-	x
Funk	BLE	BLE, 5 kHz	BLE, 5 kHz	BLE, ANT
Speicher	-	-	x	x
Indikation	-	-	-	LED
Batterie	CR2025	CR2025	CR2025	CR2032
Rohdaten	-	-	-	-
HR	x	x	x	x
Schritte	-	-	-	x
Aktivität	-	-	-	x

Einige Hersteller versuchen zudem, den notwendigen Brustgurt durch ein T-Shirt mit integrierter Elektronik zu ersetzen. Beispiele finden sich im *VivoSense Hexoskin Shirt* [221], welches neben einem EKG auch die Atmung und auftretende Beschleunigungen erfasst. Das Produkt der Firma *Ambiotex* basiert auf dem *FitnessSHIRT* des Fraunhofer-Instituts für Integrierte Schaltungen IIS in Erlangen [66] und war zunächst auch mit integriertem Atemsensor angekündigt [67]. Jedoch verfügt das erhältliche Produkt, bestehend aus *TechUnit* und T-Shirt [7], nicht über einen Atemsensor, sondern zeichnet lediglich EKG und Beschleunigung auf. Die Firma *OMsignal* [142] bietet mit dem *Ombra* einen Sport-BH an, welcher EKG, Atmung und Beschleunigung aufnehmen kann. Eine Besonderheit stellt das *Athos Core Shirt* [12] dar, welches neben EKG und Beschleunigung auch ein EMG aufzeichnet. Das Shirt kann durch Shorts mit integrierten EMG Sensoren ergänzt werden. Die Tabelle A.2 in Anhang A enthält einen Überblick der als T-Shirt erhältlichen Herzfrequenzsensoren.

Multimodale Sensorknoten Multimodale Sensorknoten sind komplexe Systeme, die eine Vielzahl von physiologischen und biokinematischen Größen erfassen. Zusätzlich besteht ein erhöhter Anspruch an die Validität der erfassten Größen und berechneten Parameter, was mit entsprechenden wissenschaftlichen Publikationen belegt wird. Dementsprechend erfolgt die Anwendung dieser System zunächst nur in der Forschung. Derzeit kommerziell verfügbare Systeme sind der *MoviSens EcgMove 3* [240], die *SHIMMER3 ECG UNIT* [187] und der *Zephyr BioPatch HP* [236]. In die gleiche Kategorie fällt auch der multimodale Sensorknoten *BG-V4.2*. Dieser wurde zunächst in der Fachgruppe Schaltungstechnik des Heinz Nixdorf Instituts der Universität Paderborn entwickelt [227] und ab 2010 in der AG K&S der Universität Bielefeld weiter entwickelt [49]. Wesentliche Punkte der Weiterentwicklung sind die Berechnung physiologischer und biokinematischer Parameter [43; 45–48; 50], die Entwicklung eines Atemsensors [253], die Kalibrierung der Sensoren [256] und die Visualisierung der Daten für die nutzende Person. Ein Überblick der Systeme wird in Tabelle 2.3 gegeben. Die angegebenen Laufzeiten sind stark anwendungsabhängig und somit nicht unmittelbar vergleichbar.

Andere universitäre Forschungsplattformen sind beispielsweise das *IPANEMA* System [104] des Lehrstuhls für Medizinische Informationstechnik (MedIT) der RWTH-Aachen⁵. Dieses besteht aus mehreren Modulen, welche ein Funknetzwerk am Körper (433 MHz) aufspannen, wobei ein Master-Knoten die Daten mittels Bluetooth an eine Basisstation sendet. Hier liegt der Fokus jedoch nicht auf den erfassten Größen, sondern auf dem drahtlosen Netzwerk der Knoten. Das *ULPSEK* System [210] der Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU) verfügt, neben einer modularen Evaluierungsplattform, auch über einen mobilen Sensorknoten. Jedoch liegt hier der Fokus auf der verwendeten Entwurfsmethodik und nicht auf der Anwendung des Sensorknotens. Beide Forschungsplattformen sind, wie der *BG-V4.2* der AG K&S, nicht kommerziell erhältlich.

⁵<https://www.medit.hia.rwth-aachen.de/>

Tabelle 2.3.: Überblick der Körpersensoren: Multimodale Sensormodule. Mit x markierte Positionen sagen aus, dass der entsprechende Sensor auf dem System zur Verfügung steht.

Hersteller Bezeichnung	Zephyr BioPatch HP	MoviSens EcgMove 3	Shimmer SHIMMER3 ECG UNIT	AG KS BG-V4.2
EKG	x	x	x	x
Atmung	x	-	-	x
Beschleunigung	x	x	x	x
Gyroskop	-	-	x	-
Magnetometer	-	-	x	-
Temperatur	-	-	-	x
Barometer	-	x	x	-
Funk	BLE	BLE	BT	2,4 GHz
Speicher	x	x	8 GB SD	8 MB Flash
Laufzeit	500 h	336 h	n.b.	12 h
Indikation	LED	LED,Vibration	LED	LED
Batterie	LiPo	LiPo	LiPo	LiPo
Laufzeit	35 h	72 h	n.b.	12 h
Rohdaten	-	x	x	x
HR	x	x	x	x
HRV	x	x	-	x
Atemfrequenz	x	-	x	x
Postur	x	-	-	x
Schritte	x	-	-	x
Aktivität	x	x	-	x
Max. Beschl.	x	-	-	x

2.5 Zusammenfassung

Die Systeme zur Erfassung der Umgebung erfahren zur Zeit den Trend weg von kabelgebundenen Systemen hin zu drahtlosen Sensorknoten. Dieser Trend wird durch die zunehmende Vernetzung aller Geräte in den Bereichen IoT und SmartHome und den damit verbundenen Aufwand zur Vernetzung vorangetrieben.

Im Bereich der Körpersensoren sind, bedingt durch den körpernahen Einsatz dieser Systeme, drahtlose und mobile Sensorknoten bereits seit längerem Stand der Technik. Im Bereich der kommerziellen Produkte sind bisher einige einfachere Systeme mit eingeschränkter Validität, wie beispielsweise die Aktivitätstracker, für den Konsumenten erhältlich. Einzig die bei Sportlern verbreiteten Herzfrequenzsensoren auf EKG-Basis liefern verlässliche Ergebnisse in der Berechnung der HR. Während diese Systeme bisher

2. Drahtlose Sensorknoten

nur auf diese Funktionalität eingeschränkt waren, werden diese in letzter Zeit zunehmend komplexer, so dass die Lücke zu den in der Forschung eingesetzten multimodalen Sensorknoten schmaler wird.

3 Vermessung drahtloser Sensorknoten

Die Abschätzung der Leistungsaufnahme spielt bei der Entwicklung von Sensorknoten eine immer größere Rolle, da eine geringe Leistungsaufnahme eine hohe Laufzeit des Systems bei gleichzeitiger Verwendung kleiner Batterien erlaubt. Da beide Eigenschaften direkte Auswirkung auf die Benutzungsakzeptanz haben, ist eine geringe Leistungsaufnahme in vielen Fällen entscheidend für den Erfolg eines Produktes. Zur Auslegung der Batterie in Abhängigkeit von der geforderten Laufzeit ist es üblich, die Leistungsaufnahme des Systems zu messen. Im folgenden Kapitel werden daher die Methodik und aktuelle Arbeiten in der Literatur zur Messung der Leistungsaufnahme von eingebetteten Systemen vorgestellt und anhand von zwei Praxisbeispielen aufgezeigt. Die praktische Vermessung von Sensorknoten umfasst dabei den Körpersensorknoten *BG-V4.2* der AG K&S in Abschnitt 3.3 und den kommerziellen *CC2560STK* Umgebungssensorknoten der Firma *Texas Instruments* in Abschnitt 3.4. Abschließend wird die Methode der Messung diskutiert und deren Grenzen aufgezeigt.

3.1 Methodik

Für die Bestimmung der potentiellen Laufzeit eines Systems ist die Messung der durchschnittlichen Leistungsaufnahme notwendig. Gebräuchlich ist hier die Angabe von Mittelwerten für bestimmte Betriebszustände (z.B. im aktiven Zustand) oder auch als Mittelwert über alle Betriebszustände. Die gemessene mittlere Leistungsaufnahme kann für die Dimensionierung der Batterie verwendet werden, wobei die erforderliche Kapazität der Batterie C_{BAT} aus der mittleren Leistungsaufnahme P_{AVG} und der gewünschten Laufzeit des Systems T_{SYS} berechnet wird (vgl. Gleichung 3.1). Jedoch kann diese erste Abschätzung um bis zu 50 % von der realen Laufzeit abweichen [153], da diese Überschlagsrechnung von einer idealen Batterie ausgeht.

$$T_{\text{SYS}} = \frac{C_{\text{BAT}}}{P_{\text{AVG}}} \quad (3.1)$$

Die mittlere Leistungsaufnahme P_{AVG} ist die Summe aller Ströme des Systems multipliziert mit der Klemmenspannung der Batterie U_{BAT} . Die Ströme des Systems setzen sich aus den Reglerverlusten [225] und den einzelnen Verbrauchern zusammen. Als Verbraucher kann hier eine gesamte Spannungsdomäne gelten oder auch einzelne Systemkomponenten wie der Mikrocontroller oder der Funktransceiver (vgl. Abschnitt 2.2). Für eine feingranulare Bestimmung der Leistungsaufnahme und somit eine genauere Analyse des Systems ist es daher notwendig, die einzelnen Komponenten und deren Zustände zu vermessen. Hier kann die mittlere Leistungsaufnahme für jede Komponenten oder auch die Energieaufnahme (in Ws oder J) für einzelne Ereignisse bestimmt werden. Für einzelne Ereignisse ist die Angabe der Energieaufnahme gebräuchlich, da sowohl die benötigte Leistung und auch die Dauer entscheidend sind (z.B. für das Senden eines Datenpakets).

Die verschiedenen Methoden der Messung der Leistungsaufnahme in eingebetteten

Systemen werden von Nakutis [136] beschrieben und hier zusammen gefasst:

- **Stromzange:** Die Zuleitung der Spannungsversorgung wird mit einer Stromzange umschlossen, wobei der Stromfluss nach dem Prinzip der magnetischen Kopplung bestimmt wird. Jedoch ist diese Methode bei eingebetteten Systemen oft nicht möglich, da die Zuleitungen Teil der Leitplatte und die Ströme oft sehr klein sind.
- **Thermische Überwachung:** Die indirekte Messung der Stromaufnahme durch die Überwachung der Erwärmung des Systems. Jedoch muss der Zusammenhang zwischen Stromaufnahme und Erwärmung bekannt sein. Zudem ist das Verfahren nicht für kleine Ströme geeignet und ist nicht in der Lage dynamische Änderungen zu erfassen.
- **Ladungstransfer:** Messung der Restspannung oder der Entladezeit von Kapazitäten. Somit ist eine feingranulare Messung möglich, jedoch wird für die Charakterisierung auch eine präzise Messung mit hoher Messrate benötigt.
- **Shunt:** Bei der Messung mittels Shunt wird ein möglichst niederohmiger ($< 1\ \Omega$) und präziser ($< 1\%$ Abweichung) Widerstand in der Strompfad eingebracht. Bei Messung des Spannungsabfalls über diesen Shunt-Widerstand kann, bei bekanntem Widerstand, die Stromaufnahme des Systems bestimmt werden. Hier ist zu beachten, dass der Shunt für die Vermessung von dynamischen Ereignissen möglichst nah am Verbraucher platziert werden muss, da ansonsten eine Glättung des Signals durch die im System verbauten Kapazitäten auftritt.
- **Stromspiegel:** Da ein Shunt-Widerstand selbst die Messung beeinflusst kann zusätzlich ein Wilson-Stromspiegel verwendet werden. Die Messung mittels Shunt erfolgt dann im gespiegelten Pfad.

In der Praxis wird häufig die Shunt-Messung verwendet, da diese kostengünstig ist, in vielen Fällen eine ausreichende Genauigkeit liefert und der Markt vielfältige Lösungen anbietet. So können beispielsweise entsprechende Komponenten (z.B. *Texas Instruments INA219* [200]) direkt in eingebettete Systeme integriert und mit einem Mikrocontroller angesteuert werden, so dass eine kontinuierliche Überwachung der Leistungsaufnahme ermöglicht wird. Zudem sind auch kommerzielle Laborgeräte (z.B. *Keysight Power Analyzer* [103]) verfügbar. Jedoch kann eine Shunt-Messung auch manuell mit einem präzisen Widerstand erfolgen, wobei der Spannungsabfall über den Widerstand mit einem Oszilloskop gemessen wird. Da eine solche manuelle Messung jedoch häufig stark rauschbehaftet und entsprechend unpräzise ist wird das im Vergleich zu den Laborgeräten günstige *μCurrent Gold* [101] angeboten, welches eine relativ präzise Shunt-Messung anbietet und den gemessenen Strom als analogen Spannung ausgibt. Diese kann dann wiederum mit einem Oszilloskop gemessen werden.

3.2 Arbeiten in der Literatur

In der Literatur finden sich einige Arbeiten, welche die Leistungsaufnahme eines Systems für eine spezifische Anwendung abschätzen und versuchen diese zu optimieren. Die Arbeit von Berlin u. a. [19] hat das Ziel mittels verschiedener Messreihen den energetisch optimalen Arbeitspunkt für einen Sensorknoten zu ermitteln. Die Zielanwendung ist hier das Speichern von Beschleunigungsdaten mit 100 Hz über mehrere Tage, wobei das System einen dreiachsigen *ADXL345* Beschleunigungssensor, einen *PIC18* Mikrocontroller, eine microSD-Karte, Ladeelektronik und einen Akku mit einer Kapazität von 180 mAh umfasst. Die Messung erfolgt mittels Shunt und untersucht den Einfluss verschiedener Parameter auf die Leistungsaufnahme. So wird das Schreiben auf verschiedene SD-Karten, die Variation der Ausführungsfrequenz des Beschleunigungssensors, eine Datenkompression (K-RLE), der Einfluss von FIFO Burst Zugriffen und die Verwendung der verschiedenen inaktiven Modi des Mikrocontrollers untersucht. Basierend auf den Einzelmessungen wird die Konfiguration mit der potentiell niedrigsten Leistungsaufnahme bestimmt und somit eine Laufzeit des Systems von 10 Tagen erreicht. Jedoch erfolgt keine Bewertung der verschiedenen Konfigurationen hinsichtlich ihrer Leistungsfähigkeit. Die Arbeit von Tobola u. a. [216] untersucht die Eignung verschiedener Mikrocontroller für einen spezifischen Algorithmus. So wird ein *MSP430* Mikrocontroller mit verschiedenen *ARM* Architekturen (*ARM Cortex-M0+/-M3/-M4F*) verglichen, wobei alle Mikrocontroller mit einer Taktfrequenz von 14 MHz betrieben werden. Der Testalgorithmus detektiert Aussetzer in der Atmung. Die Strommessung mittels Shunt ergibt, dass der *Cortex-M4F* die geringste Leistungsaufnahme aufweist. Die Ergebnisse werden jedoch nicht umfassend diskutiert, da nur die Leistungsaufnahme der Mikrocontroller betrachtet wurde. Entscheidend ist jedoch die Ausführungszeit und somit die Energieaufnahme für den spezifischen Algorithmus. Für einen Vergleich der Leistungsaufnahme müsste zudem der Tastgrad des Systems spezifiziert werden. Auch hier findet keine Betrachtung der Leistungsfähigkeit statt. Die Effizienz eines Systems wird in der Literatur somit als die geringste Leistungsaufnahme für eine spezifische Anwendung betrachtet.

3.3 Vermessung Körpersensorknoten BG-V4.2

Der in Kapitel 2 eingeführte Körpersensor BG-V4.2 der AG K&S wurde im Rahmen einer betreuten Bachelorarbeit vermessen [283]. Hier wurde ein eigens entwickeltes Messboard (BGM-V4.2) verwendet, welches die Hardware des Körpersensors um einen übergeordneten Mikrocontroller und eine Shunt-Messung aller Komponenten mit integrierten Bauteilen zur Stromüberwachung (*Texas Instruments INA219B* [200]) erweitert.

Aktive und inaktive Modi der Komponenten In einem ersten Schritt wurden die einzelnen Komponenten und deren aktive sowie inaktive Betriebsmodi vermessen. Betrachtet wurden hier die folgenden Komponenten bei Variation der angegebenen Parameter:

3. Vermessung drahtloser Sensorknoten

- **Mikrocontroller MSP430:** *Active, Idle, PM0, PM2, PM3*
- **Flash Speicher S25FL064K:** *Idle, Löschen, Lesen (1/256/1024 Byte), Schreiben (1/256/1024 Byte)*
- **Funktransceiver nRF24L01:** *Senden (1/1024 Byte), Idle, PowerDown*
- **Mikrocontroller ATmega328P:** *Senden (1/1024 Byte), Idle, PowerDown*
- **Beschleunigungssensor LIS331HH:** *PowerDown, Low Power Sampling 0,5 - 10 Hz, Normal Sampling 50-1000 Hz*
- **Analoge Sensorteile (OPA2334, INA321, MSP430):** *Analog aus, Sampling Temperatur, Sampling Herzfrequenz, Alle aktiv*
- **Spannungswandler TPS61020:** *Effizienz bei Versorgung mit Knopfzelle und USB*

Gesamtsystem Die Konfiguration des Körpersensors BG-V4.2 folgt einem festen Ablauf. In einer Hauptschleife im Programmablauf werden zunächst die Daten des digitalen Beschleunigungssensors abgefragt. Anschließend werden die analogen Sensoren zur Messung der Herzfrequenz und Temperatur ausgelesen und verarbeitet. Eine Verteilung der Leistungsaufnahme des Gesamtablaufs auf die einzelnen Komponenten wurde im Rahmen der betreuten Bachelorarbeit [283] analysiert und bereits in der Arbeit von Christ [49] beschrieben.

Optimierte Konfigurationen Basierend auf dieser Untersuchung erfolgte abschließend eine Optimierung des Gesamtsystems, wobei eine Anpassung der Abtast- und Datenraten, sowie die Nutzung eines effizienteren inaktiven Modus des Mikrocontrollers erfolgte. Im Anschluss an die betreute Bachelorarbeit [283] wurden zwei verschiedene Speicherformate definiert und das System im Rahmen einer Hilfskrafttätigkeit des Studenten erneut vermessen. Die gemessene Leistungsaufnahme des BG-V4.2 Körpersensorknotens nach Optimierung ist in Tabelle 3.1 dargestellt. Versorgt wurde das System während der Messungen von einer CR2025 Knopfzelle.

Das Speicherformat *mem-l-1.0* umfasst alle Sensordaten und berechnete Größen in voller Auflösung und ist 13 Byte breit. Das Speicherformat *mem-s-1.0* beinhaltet die gleichen Größen in reduzierter Auflösung und ist 8 Byte breit. Das Funkformat *rf-l-1.0* mit einer Breite von 22 Byte umfasst, neben den Größen in voller Auflösung, weitere Informationen wie einen Paketzähler und eine *Cyclic Redundancy Check* (CRC)-Prüfsumme. Für die weiteren Arbeiten mit dem BG-V4.2 Körpersensor wurde, abhängig von der Anwendung, eine der erstellten Konfigurationen verwendet. Der Körpersensor verfügt somit über eine Laufzeit zwischen 2 und 12 Stunden [49].

Tabelle 3.1.: Gemessene Leistungsaufnahme des BG-V4.2 Körpersensorknotens nach Optimierung. Das System wurde während der Messungen von einer CR2025 Knopfzelle versorgt

Konfiguration	BG-K1	BG-K2	BG-K3	BG-K4
Speicherformat	mem-l-1.0	mem-l-1.0	mem-s-1.0	mem-s-1.0
Funkformat	rf-l-1.0	rf-l-1.0	rf-l-1.0	-
Abtastrate	200 Hz	200 Hz	125 Hz	125 Hz
Datenrate	100 Hz	100 Hz	25 Hz	25 Hz
Standby Modus	<i>Idle</i>	<i>LPM1</i>	<i>LPM1</i>	<i>LPM1</i>
Leistungsaufnahme	39,24 mW 100 %	24,21 mW 62 %	20,10 mW 51 %	13,94 mW 36 %

3.4 Vermessung Umgebungssensorknoten TI CC2560STK

Der Texas Instruments *SimpleLink SensorTag CC2650STK* [203] basiert auf dem *CC2650* SoC [198] von *Texas Instruments*, welcher Funktransceiver und Mikrocontroller in einem Gehäuse vereint. Dieser SoC verfügt insgesamt über drei Rechenkerne, wobei für den Anwendungscode ein *ARM Cortex-M3* zur Verfügung steht. Zusätzlich beinhaltet der SoC einen *Cortex-M0* für die Konfiguration und Ansteuerung der Funkschnittstelle, sowie einen speziellen *Reduced Instruction Set Computer* (RISC)-Rechenkern, welcher vom Hersteller als *Sensor Controller* bezeichnet wird. Während sich der *Cortex-M3* im inaktiven Modus befindet ist dieser in der Lage, die Sensoren des Systems abzufragen und einfache Berechnungen durchzuführen. Zudem kann dieser den *Cortex-M3* mittels Interrupt zurück in einen aktiven Zustand versetzen. Neben dem *CC2650* SoC verfügt der *Texas Instruments SimpleLink SensorTag CC2650STK* über fünf verschiedene Sensoren und einen Flash-Speicher:

- CC2650 SoC [198]:
 - 2,4 GHz Funktransceiver
 - *Cortex-M3* Applikationsmikrocontroller (48 MHz Taktrate)
 - *Cortex-M0 Radio Controller* (48 MHz Taktrate)
 - *RISC Sensor Controller* (24 MHz Taktrate)
- Sensoren:
 - Texas Instruments HDC1000 (Temperatur und Luftfeuchtigkeit) [199]
 - Texas Instruments OPT3001 (Umgebungs-Lichtintensität) [201]
 - Texas Instruments TMP007 (Infrarot- und Umgebungstemperatur) [204]
 - Bosch BMP280 (Luftdruck) [29]
 - Invensense MPU9250 (Drehrate, Beschleunigung, magnetische Flussdichte) [97]

- Winbond W25X40CLUXI SPI NOR-Flash (4 Mbit) [229]

Die Evaluierung des *TI SensorTags* bezüglich der Konfigurationsmöglichkeiten und der damit verbundenen Leistungsaufnahme wurde im Rahmen einer betreuten Bachelorarbeit durchgeführt [286]. Inhalt der Arbeit war die Entwicklung verschiedener Konfigurationen des Sensorknotens, die Entwicklung eines BLE-Profiles zur effizienten und adaptiven Übertragung der Sensordaten, sowie die Vermessung der Leistungsaufnahme. Vermessen wurde die Leistungsaufnahme der einzelnen Sensoren, der Funkübertragung mittels *BLE-Notification* bei veränderlicher Nutzlast, sowie die durchschnittliche Leistungsaufnahme des Gesamtsystems bei vier verschiedenen exemplarischen Konfigurationen. Bei der Messung handelt es sich um eine Shunt-Messung, wobei das *μCurrent Gold* [101] verwendet wurde.

Funkübertragung Die Vermessung der Funkübertragung mittels BLE diente als Voruntersuchung für die Umsetzung der Sensordatenübertragung. Untersucht wurde die Dauer und die durchschnittliche Leistungsaufnahme der Übertragung eines Funkpakets ohne Nutzdaten (leeres *Connection Event*) und von *BLE-Notifications* mit einer Nutzlast zwischen 1 und 20 Byte. Somit konnte die Energieaufnahme des gesamten Funkpakets und die Energieaufnahme pro Byte Nutzdaten bestimmt werden, welche in Tabelle 3.2 dargestellt ist. Die Energieaufnahme pro Byte Nutzdaten berechnet sich aus der Energieaufnahme des jeweiligen Funkpakets abzüglich der Energieaufnahme des leeren Pakets geteilt durch die Anzahl der übertragenen Bytes Nutzdaten. Die nicht stetig steigende Energieaufnahme bei steigender Breite der Nutzdaten ist hierbei potentiell auf rauschbehaftete Messungen zurück zu führen.

Tabelle 3.2.: Energieaufnahme der CC2560 BLE-Funkübertragung

Nutzdaten	Energie des Pakets	Energie pro Byte
0 Byte	31,00 μJ	-
1 Byte	36,99 μJ	5,990 $\mu\text{J}/\text{Byte}$
2 Byte	35,19 μJ	2,095 $\mu\text{J}/\text{Byte}$
3 Byte	37,61 μJ	2,203 $\mu\text{J}/\text{Byte}$
4 Byte	35,87 μJ	1,218 $\mu\text{J}/\text{Byte}$
5 Byte	36,08 μJ	1,016 $\mu\text{J}/\text{Byte}$
6 Byte	36,88 μJ	0,980 $\mu\text{J}/\text{Byte}$
20 Byte	40,26 μJ	0,463 $\mu\text{J}/\text{Byte}$

Aus der Voruntersuchung folgte, dass bei größerer Nutzlast weniger Energie pro Byte Nutzlast aufgewendet werden muss. Die Übertragung der Sensordaten wurde daher in

einem Wert (*BLE-Characteristic*) mit einer Datenbreite von 20 Byte zusammen gefasst. Es werden somit die Messdaten der Sensoren *BMP280* (6 Byte), *HDC1000* (4 Byte), *OPT3001* (2 Byte) und *TMP007* (4 Byte) zusammen mit einem 4 Byte breiten Paketzähler in einer *BLE-Notification* übertragen. Im BLE Profil wurde ein Kontrollpunkt implementiert, welcher es erlaubt, die Übertragung der Sensordaten zu aktivieren beziehungsweise zu deaktivieren. Falls die *Notification* aktiviert ist, kann deren Übertragungsfrequenz mit Hilfe eines weiteren Kontrollpunkts in 10 ms Schritten in einem Bereich von 100 ms bis 2550 ms eingestellt werden. Zusätzlich kann die Abtastfrequenz der Sensoren im gleichen Intervall konfiguriert werden. Bei Deaktivierung der *Notification* werden auch die Sensoren deaktiviert, also in den Standby-Modus versetzt. Die adaptive Übertragung der Sensordaten umfasst daher zum einen die Aktivierung der einzelnen Sensoren, als auch deren Konfiguration hinsichtlich Abtast- und Übertragungsfrequenz. Diese Parameter beeinflussen die Leistungsaufnahme des Sensorknotens, so dass dieser entsprechend der jeweiligen Anforderungen konfiguriert werden kann. Die auf dem Sensorknoten vorhandene *MPU-9150* IMU der Firma *Invensense* wurde nicht betrachtet, da der Fokus auf den Umgebungssensoren und -größen lag.

Inaktive Modi der Komponenten In einem nächsten Schritt wurde die Stromaufnahme des Systems im inaktiven Zustand vermessen und mit den Herstellerangaben verglichen. Die vom jeweiligen Hersteller angegebene Stromaufnahme I_{DAT} wurde dem entsprechenden Datenblatt entnommen [29; 198; 199; 201; 204; 229]. Die gemessene Stromaufnahme I_{MESS} konnte nur für das Gesamtsystem ermittelt werden, da für die Messung der Stromaufnahme der einzelnen Komponenten des Sensorknotens vom Hersteller keine entsprechenden Messpunkte (z.B. Brücken oder Jumper) vorgesehen wurden und ein Auftrennen der stromführenden Verbindungen auf der Leiterplatte nicht praktikabel war. Wie in Tabelle 3.3 zu sehen ist, liegt die gemessene Stromaufnahme im Bereich der Herstellerangaben. Auch unter Einbeziehung der vom *CC2650* ausgeführten Ladeimpulse für die Kapazität der Versorgungsspannung (*Recharge-Puls* [286]) liegt der Wert in diesem Bereich.

Aktive Modi der Komponenten Anschließend wurde die Stromaufnahme der Komponenten in den zur Verfügung stehenden aktiven Modi vermessen. Bei dem Funktransceiver handelt es sich hier um den Modus *Senden* und bei den Sensoren um die Erfassung der Messwerte. Der Sensor *BMP280* verfügt über zwei verschiedene Messmodi. Im *Normal Mode* misst der Sensor kontinuierlich und im *Forced Mode* geht der Sensor nach einer Messung in den Standby Modus. *Continuous Mode* und *Forced Mode* des *OPT3001* Sensors arbeiten in gleicher Weise. Die Sensoren *HDC1000* und *TMP007* verfügen nur über einen Messmodus. Eine Gegenüberstellung der gemessenen Stromaufnahme I_{MESS} mit der vom Hersteller im jeweiligen Datenblatt angegebenen Stromaufnahme I_{DAT} erfolgt in Tabelle 3.4. Die Komponenten liegen bezüglich der Stromaufnahme bis auf den *OPT3001* alle in

3. Vermessung drahtloser Sensorknoten

Tabelle 3.3.: Stromaufnahme der CC2560STK Komponenten im inaktiven Modus

Komponenten	I_{DAT}	I_{MESS}
BMP280	0,10-0,30 μA	-
HDC1000	0,11-0,20 μA	-
OPT3001	0,30-0,47 μA	-
TMP007	2,00-4,00 μA	-
CC2650	1,00-2,70 μA	-
W25X40CLUXI	1,00-5,00 μA	-
Gesamt	4,91-10,97 μA	6,92 μA
Gesamt (Recharge)	-	8,15 μA

der Nähe der Spezifikation. Entscheidend für eine Evaluierung der Sensoren ist, neben der Stromaufnahme, auch die Dauer der jeweiligen Aufgabe (senden oder messen), so dass mit der hier gemessenen Dauer t_{MESS} auch die Energieaufnahme ermittelt werden kann.

Optimierte Konfigurationen Basierend auf den Voruntersuchungen der einzelnen Komponenten und deren aktiven Modi wurden verschiedene Konfigurationen für den CC2650STK Sensorknoten implementiert und vermessen. In der Konfiguration TI-K1 wird der *Cortex-M3* Applikationsmikrocontroller mitsamt dediziertem *Inter-Integrated Circuit* (I2C)-Modul verwendet. Die Abfrage der Sensoren erfolgt für jeden Sensor in einem einzelnen *Real-Time Operating System* (RTOS)-Task. In Konfiguration TI-K2 wird ebenfalls der *Cortex-M3* Applikationsmikrocontroller mitsamt dediziertem I2C-Modul verwendet. Die

Tabelle 3.4.: Stromaufnahme der CC2560STK Komponenten in den aktiven Modi

Komponenten	Task	I_{DAT}	I_{MESS}	t_{MESS}
BMP280	Sampling (Normal Mode)	325-1120 μA	614 μA	99,68 ms
BMP280	Sampling (Forced Mode)	325-1120 μA	175 μA	82,93 ms
HDC1000	Sampling	155-220 μA	363 μA	17,26 ms
OPT3001	Sampling (Continuous Mode)	1,8-3,7 μA	2200 μA	1,40 ms
OPT3001	Sampling (Single Shot)	1,8-3,7 μA	1800 μA	1,53 ms
TMP007	Sampling	270-350 μA	230 μA	280,06 ms
CC2650	Senden 20 Byte Not. (0 dBm)	6100 μA	4500 μA	2,97 ms

Abfrage der Sensoren erfolgt jedoch in einem gemeinsamen RTOS-Task. Die Konfiguration TI-K3 verwendet statt dem *Cortex-M3* Applikationsmikrocontroller den *Sensor Controller*. Die Abfrage der Sensoren erfolgt in einem gemeinsamen RTOS-Task. Die Konfiguration TI-K4 verwendet auch den *Sensor Controller* und fragt die Sensoren in einem gemeinsamen RTOS-Task ab. Zusätzlich wird der *Forced Mode* des BMP280 Sensors und der *Single Shot Mode* des OPT3001 Sensors verwendet. Eine Übersicht der Konfigurationen und deren Stromaufnahme wird in Tabelle 3.5 gegeben. Die Ausführungsfrequenz der Sensoren und die Frequenz der Funkübertragung beträgt jeweils 1 Hz.

Tabelle 3.5.: Vergleich der CC2560STK Konfigurationen

Konfiguration	TI-K1	TI-K2	TI-K3	TI-K4
Ausführung	Cortex-M3	Cortex-M3	<i>Sensor Controller</i>	<i>Sensor Controller</i>
Sensorabfrage	Einzel	Gemeinsam	Gemeinsam	Gemeinsam
BMP280	<i>Normal Mode</i>	<i>Normal Mode</i>	<i>Normal Mode</i>	<i>Forced Mode</i>
OPT3001	<i>Cont. Mode</i>	<i>Cont. Mode</i>	<i>Cont. Mode</i>	<i>Single Shot</i>
TMP007	Sampling	Sampling	Sampling	Sampling
HDC1000	Sampling	Sampling	Sampling	Sampling
Stromaufnahme	175,59 μ A 100 %	162,63 μ A 93 %	131,09 μ A 75 %	66,59 μ A 38 %

Durch vorherige Vermessung der einzelnen Komponenten und die somit mögliche Bestimmung der effizientesten Modi konnte eine Optimierung der Konfiguration des Gesamtsystems durchgeführt werden. So war es in Konfiguration TI-K4 möglich, die mittlere Stromaufnahme im Vergleich zur unoptimierten Konfiguration TI-K1 auf 38 % zu senken (vgl. Tabelle 3.5). Neben der Ausführung durch den *Sensor Controller* und die Abfrage der Sensoren in einem gemeinsamen RTOS-Task hatte auch die Wahl der Messmodi der Sensoren einen großen Einfluss.

3.5 Zusammenfassung und Bewertung der Methode

Die Methode der Vermessung stellte bisher eine praktikable Vorgehensweise zur Bestimmung und Optimierung der Leistungsaufnahme bestehender Sensorknoten dar. Jedoch bieten heutzutage bereits vergleichsweise einfache Sensorknoten, bestehend aus einem Mikrocontroller und mehreren Sensoren, komplexe Konfigurationsmöglichkeiten. Anhand des Stands der Technik und der beispielhaften Vermessung des Umgebungssensorknotens *TI CC2560STK* und des Körpersensorknotens *BG-V4.2* konnte gezeigt werden, dass diese vielfältigen Konfigurationsmöglichkeiten dazu führen, dass Optimierungspotentiale hinsichtlich der Energieeffizienz nicht direkt ersichtlich sind, sondern experimentell ermittelt

3. Vermessung drahtloser Sensorknoten

werden müssen. Dieser Umstand ist oftmals auf die heterogene Informationsdarstellung in den Datenblätter der einzelnen Komponenten zurück zu führen. Die Leistungsaufnahme eines Sensorknotens im Realbetrieb mit den Angaben des Herstellers aus den Datenblättern der Komponenten abzuschätzen ist nicht direkt möglich, so dass sich die tatsächliche Leistungsaufnahme nur aus der Messung ergibt. Mit der Konfigurationsvielfalt des Gesamtsystems und der experimentellen Vorgehensweise kann jedoch nicht sicher gestellt werden, dass das Energieoptimum des Sensorknotens mit einer der getesteten Konfigurationen erreicht wird. Zudem besteht ein großes Optimierungspotential in der Software, welches zuvor nicht abgeschätzt werden kann. Für eine Betrachtung der Energieeffizienz muss zudem, neben der Leistungsaufnahme, auch die Leistungsfähigkeit eines Systems ermittelt werden. Jedoch erfordert eine solche Bewertung neben der Analyse sämtlicher Hardwarekomponenten auch eine Einbeziehung der Software. Mittels der Methode der Messung ist dieses nur sehr eingeschränkt möglich, so dass Bedarf an einer neuen Vorgehensweise besteht.

4 Modellierung drahtloser Sensorknoten

Mit zunehmender Komplexität der Systeme (Vielzahl der Komponenten und möglicher Betriebszustände) wird die Methode der Messung immer aufwändiger und liefert oftmals keine umfassende Abschätzung der Energieaufnahme des Systems im realen Betrieb. Daher wird angestrebt, die Systeme zu modellieren und mit Hilfe einer Simulation eine möglichst gute Abschätzung der Leistungsaufnahme zu erhalten. Hier müssen vor allem die möglichen Betriebszustände der Komponenten und die auszuführenden Algorithmen (*software related power*) berücksichtigt werden. Neben der Abschätzung der Leistungsaufnahme bestehender Sensorknoten besteht ein großer Vorteil der Modellierung in der möglichen Analyse des Systems bereits vor oder während des Entwurfs.

In der Literatur muss zwischen reiner Messung und tatsächlicher Modellierung und Simulation unterschieden werden. Eine Messung stellt meist nur eine Stichprobe einiger weniger Konfigurationen dar, so dass oftmals kein ganzheitliches Abbild des Verhaltens und somit der Leistungsaufnahme entsteht. Eine Modellierung hingegen nutzt zur Verfügung stehende Informationen und fügt diese zu einem umfassenden Abbild des Systemverhaltens zusammen. Mit der Inter- beziehungsweise Extrapolation dieser Informationen können dann Abschätzungen über den gesamten Entwurfsraum getroffen werden.

Basis für die Modellierung eines Sensorsystems ist zunächst die Modellierung der einzelnen Komponenten. So kann über die Leistungsaufnahme in den verschiedenen Betriebsmodi und die Dauer der jeweiligen Operation die Energieaufnahme bestimmt werden. In heutigen Systemen mit vielen Komponenten, welche wiederum jeweils verschiedene Betriebszustände einnehmen können, führt auch eine solche Betrachtung bereits zu einem großen Entwurfsraum. Somit ist oftmals eine Abstraktion des Systems notwendig. Zusätzlich ist die Energieaufnahme von Mikrocontrollern abhängig von der jeweiligen Anwendung und den auszuführenden Algorithmen, so dass auch eine Modellierung der Software notwendig ist.

Im Folgenden wird zunächst der Stand der Technik für die Modellierung der einzelnen Komponenten eines Sensorsystems beschrieben. Diese umfassen, wie in Abschnitt 2.2 vorgestellt, die Komponenten Mikrocontroller, Sensoren, Funktransceiver, Speicher, Bauteile zur Indikation, Schnittstellen, Spannungswandler und Energiespeicher. Zudem wird ein Überblick über die Ansätze zur Modellierung des Gesamtsystems und der Abbildung des Energiemanagements gegeben. Abschließend wird eine Zusammenfassung und Bewertung des Stands der Technik bezüglich der Modellierung drahtloser Sensorknoten gegeben.

4.1 Mikrocontroller

Ein Mikrocontroller stellt in den meisten drahtlosen Sensorknoten das zentrale Element der Informationsverarbeitung dar. Die Abschätzung der Leistungsaufnahme kann auf verschiedenen Ebenen erfolgen. So ermöglicht eine Betrachtung auf Gatter- [107] oder Architekturebene [114] zwar eine präzise Abschätzung, jedoch werden detaillierte Informationen über den schaltungstechnischen Aufbau eines Systems benötigt, welche in

den meisten Fällen nicht zur Verfügung stehen [174]. Entscheidend für die Energieaufnahme eines Mikrocontrollers ist der Tastgrad, also das Verhältnis von aktivem Modus zu inaktivem Modus, sowie die Leistungsaufnahme in diesen Betriebsmodi. Während die Leistungsaufnahme der Betriebsmodi als erste Näherung oftmals aus dem Datenblatt entnommen oder gemessen werden kann, ist die Ausführungszeit (Dauer des aktiven Modus) abhängig vom auszuführenden Algorithmus und der Architektur des Mikrocontrollers.

Um eine Vergleichbarkeit zwischen verschiedenen Systemen zu schaffen wird versucht, die Leistungsbewertung von Mikrocontrollern mit Hilfe von standardisierten Benchmarks, wie dem *CoreMark* und dem *ULPMark* des *Embedded Microprocessor Benchmark Consortium* (EEMBC) [55] abzuschätzen. Der *CoreMark* Benchmark generiert die Kennzahl *CoreMark/MHz* als ein auf die Taktfrequenz normiertes Maß für die Leistungsfähigkeit. Zusätzlich kann mit Hilfe der Angabe der auf die Taktfrequenz normierten Leistungsaufnahme in *mW/MHz* eine erste Abschätzung der Energieeffizienz erfolgen. In Kombination ergibt sich daraus die Kennzahl *CoreMark-Power-Ratio* mit der Einheit *CoreMark/mW*. Für den *ULPMark* wird die Kennzahl *ULPMark-CP* generiert, welche ein Maß für die für den Benchmark benötigte Energie ist. Jedoch stellt ein Benchmark stets eine synthetische Anwendung dar, so dass mit einer solchen Angabe keine Aussage über die Energieaufnahme anwendungsspezifischer Algorithmen möglich ist. Die Mikrocontroller können anhand ihrer erreichten Werte lediglich klassifiziert werden.

Eine präzise Abschätzung der Energieaufnahme eines Mikrocontrollers kann somit immer nur für die individuelle Kombination aus Zielplattform und auszuführendem Algorithmus erfolgen. Da eine Implementierung der Algorithmen für alle potentiell geeigneten Zielplattformen zur Abschätzung der Energieeffizienz nicht praktikabel ist, muss an dieser Stelle abstrahiert werden. In der Literatur findet sich der Ansatz, die Energieaufnahme für spezifische Algorithmen über die Art und Anzahl der auszuführenden Instruktionen abzuschätzen. So beschreiben Tiwari u. a. [209] bereits 1994 den Aufbau eines Modells für die Leistungsaufnahme eines Prozessors auf Basis der Instruktionen. Die Basiskosten für jede Instruktion, bestehend aus der benötigten Anzahl der Takte und der spezifischen Stromaufnahme, werden jeweils mittels Messung bestimmt. Zudem werden mögliche Variationen der Basiskosten betrachtet, die auf die Werte der Operanden (Abweichung < 5%), den Systemzustand zum Zeitpunkt der Ausführung der Instruktion (Abweichung < 4%), *Stalls* aufgrund von Ressourcenbeschränkungen und *Cache misses* zurück zu führen sind. Mit vorheriger Abschätzung der *Stalls* und *Misses* und entsprechender Korrekturfaktoren konnte eine Abweichung zwischen Modellierung und Messung von unter 3% erreicht werden.

Ein weiteres Modell auf Basis der Instruktionen wurde von Simunic u.a. [188; 189] vorgestellt, wobei die ermittelte Energieaufnahme als Information für einen *Instruktionssatzsimulator* (ISS) für ARM Prozessoren (*ARMulator* [1]) verwendet wird. Somit kann die Energieaufnahme des Systems unter Einbeziehung der Versorgungsspannung und der Taktfrequenz pro Takt berechnet werden. Es wird eine Abweichung von unter 5% erreicht. Die Arbeit von Steinke u. a. [192] greift die Arbeit von Simunic u. a. [188] mit der Kritik auf, dass

die Basiskosten der Instruktionen auf Angaben im Datenblatt beruhen und das Modell keine spezifischen Instruktionen unterscheidet. Somit eignet es sich nicht für einen energiegetriebenen Compiler. Die Autoren erweitern das Modell um die Berücksichtigung verschiedener Instruktionen, ein Scheduling der Instruktionen, den Einfluss der Speicher-Hierarchie, Bit-Toggeling auf den Bussen und vermessen die Variation dieser Parameter. Es wird eine Genauigkeit von 1,7 % erreicht. Mit der Begründung, dass die Arbeit von Steinke u. a. [192], wie andere Modelle auch, nur für eine einzelne Anwendung validiert wurde und zudem eine Vielzahl an Parametern zu bestimmen sind, stellen Bazzaz u. a. [16] ein eigenes Modell auf. Die Energieaufnahme eines Programms wird als die Summe einzelner Ereignisse berechnet, wobei der Prozessorkern und die internen SRAM- und Flash-Speicher abgebildet werden. So umfassen die Ereignisse, neben den einzelnen Instruktionen, das Lesen eines Bytes vom SRAM, das Schreiben eines Bytes auf den SRAM, das Lesen eines Bytes vom Flash, die Hamming-Distanz der Instruktionen, das Wortgewicht der Instruktionen, die Bit-Flips in den Registerbänken und auftretende Shift-Operationen. Zur Ermittlung der Energieaufnahme für jedes Ereignis wurde zunächst die Energieaufnahme von sechzig speziellen Testprogrammen gemessen, wobei jedes Testprogramm den Effekt eines spezifischen Ereignisses hervorhebt. Auf dieser Basis wurde anschließend eine Regressionsanalyse zur Ermittlung der Energieaufnahme der einzelnen Ereignisse durchgeführt. Zur Validierung wurde die Modellierung von sieben verschiedenen Algorithmen mit deren Messung verglichen, wobei in jedem Fall eine Abweichung von unter 6 % erreicht wurde.

Von Bogdanov [26] wurde ein Energiemodell für Mikrocontroller erstellt, in dem die Energieaufnahme für einzelne Instruktionen vermessen und in einem *Lookup table* (LUT) abgelegt wird. Die Vermessung erfolgt hier exemplarisch für einen *ARM Cortex-M4*. Jedoch erlaubt die Einführung eines abstrakten Instruktionssatzes theoretisch die generische Verwendung des Modell. Die vom Autor identifizierten Instruktionen werden unter Variation der Taktrate und des Speicherzugriffs (Flash, SRAM) vermessen und im Modell abgelegt. Es erfolgt keine Validierung des Ansatzes, jedoch wird laut Autor ein Fehler im Bereich von 20-30 % erwartet. In einer darauf folgenden Arbeit beschreibt Bogdanov [25] die Anwendung des Modells in einem eigenen ISS (*Powot Simulator v.2*). So wird aus dem Objektcode eines Programms mit Hilfe der *GNU Compiler Collection* die Art und Anzahl der enthaltenen Anweisungen ermittelt. Zusammen mit dem Energiemodell der Instruktionen wird die Energieaufnahme zunächst für jede Anweisung, anschließend für jede Funktion und schlussendlich für das gesamte Programm ermittelt. Dieses beinhaltet auch die Anpassung an enthaltene Anweisungen zur Änderungen der Taktrate und der Versorgungsspannung, so dass ein *Dynamic Voltage and Frequency Scaling* (DVFS) berücksichtigt werden kann. Die abschließende Validierung erfolgt auf Basis von zehn verschiedenen Algorithmen der *BEEBS Benchmark-Suite* [148]. Absolut gesehen wurden jedoch Abweichungen von mehr als 100 % zwischen Simulation und Messung ermittelt. Die Variation der Taktrate und des Speicherortes hingegen führte zu einer Änderung der Energieaufnahme, für die, relativ betrachtet, eine Übereinstimmung von unter 6,2 % zwischen Simulation und Messung

gezeigt werden konnte.

Die Arbeit von Roth u. a. [174] greift die Arbeiten von Tiwari u. a. [209] und Steinke u. a. [192] mit der Kritik auf, dass die Messungen mit einem handelsüblichen Strommessgerät durchgeführt wurden und daher nicht ausreichend präzise sind. Die Autoren entwickeln daher eine eigene Messplattform bestehend aus einem Mikrocontroller, einem Analog-Digital-Wandler (ADC) und einem Shunt-Widerstand, welches von einem Host-PC aus gesteuert werden kann. Jedoch erfolgt keine Evaluierung der Messgenauigkeit. Die Ergebnisse der Messungen werden in XML-Dateien abgelegt und berücksichtigen die Abhängigkeit zwischen den Instruktionen sowie den jeweiligen Speicherort. Der Hamming-Abstand zwischen einzelnen Instruktionen soll in einer späteren Version berücksichtigt werden. Diese Informationen werden einem eigens entwickelten Compiler zur Verfügung gestellt, welcher neben der maximalen Laufzeit eines Algorithmus (*Worst-Case Execution Time* (WCET)) auch dessen Energieaufnahme analysieren kann.

An Instruktionssatzsimulatoren ist, neben dem von Simunic u. a. [188] verwendeten *ARMulator* des *ARM DS-5 Development Studio* [1], der *Keil μ Vision Execution Profiler* [102] und der *Open Virtual Platform* (OVP)-Simulator [90] verfügbar. Diese liefern eine Aussage über den Typ und die Anzahl der ausgeführten Instruktionen sowie die zur Ausführung benötigten Takte.

Ansätze zur Modellierung der Energieaufnahme auf Basis der Instruktionen für Multiprozessoren finden sich beispielsweise in den Arbeiten von Purnaprajna [163], Kronic u. a. [113] und Klarhorst u. a. [106].

Während die Arbeiten von Tiwari u. a. [209] und Folgende als *Instructional Level Power Analysis* (ILPA) bezeichnet werden, basieren andere Arbeiten auf der *Functional Level Power Analysis* (FLPA)-Methode, die von Laurent u. a. [115] beschrieben wurde. So wird ein Modell für den Prozessor und ein Modell für den Algorithmus erstellt. Für das Modell des Prozessors werden die funktionalen Blöcke des Prozessors identifiziert und dessen Leistungsaufnahme per Messung abgeschätzt. Parameter der Architektur sind die Taktfrequenz F , der *memory mode* für Instruktionen MM, das *data mapping* DM, die *Direct Memory Access* (DMA) Datenbreite W und der *power management* Parameter PM. Dieses Modell stellt somit den Zusammenhang zwischen der Aktivität des Prozessor und dessen Leistungsaufnahme her. Für das Modell des Algorithmus werden einzelne Parameter aus dem Code gewonnen und mit dem Modell des Prozessors verbunden. Algorithmische Parameter sind die *parallelism rate* α , die *processing rate* β , die *cache miss rate* γ , die *external data memory access rate* τ , die Aktivität zwischen dem Speichercontroller und dem DMA ϵ sowie die *pipeline stall rate* PSR. Somit wird ermittelt, welche Aktivität der Algorithmus im Prozessor generiert. Für eine Abschätzung von Assembler-Code genügen diese beiden Modelle. Für eine Abschätzung von C-Code muss der Einfluss des Compilers mit Hilfe eines Vorhersagemodells abgeschätzt werden. Das Vorhersagemodell für den Compiler bildet die Compileroptionen (*Flags*) wie die Optimierung auf Performanz oder Speichergröße beziehungsweise die parallele oder sequentielle Verarbeitung ab. Anwendung findet die Methode im *SoftExplorer* [181] der Autoren, welcher es ermöglichen soll, die Leistungs-

und Energieaufnahme verschiedener Algorithmen (in Assembler oder C) für spezifische Zielarchitekturen abzuschätzen und somit den geeigneten Prozessor und dessen optimale Taktfrequenz zu bestimmen. Mittels *SoftExplorer* kann in einer ersten groben Abschätzung der C-Code geparkt und die Leistungsaufnahme des Algorithmus für den Bereich 0-100 % der Parameter *cache miss rate* γ und *pipeline stall rate* PSR ermittelt werden. Nach Festlegung des *data mappings* kann eine feinere Abschätzung erfolgen, welche mögliche Datenkonflikte für jede Schleife im Programmablauf ermittelt. Falls $\gamma = 0$ gilt kann für jeden Durchlauf die Anzahl der *pipeline stalls* und die Ausführungszeit ermittelt werden. Neben der Energieaufnahme können somit auch die Parameter *parallelism rate* α und die *processing rate* β bestimmt werden. Für eine exakte Abschätzung werden die Parameter Ausführungszeit, *cache miss rate* γ und PSR nicht vom *SoftExplorer* vorhergesagt, sondern vom Anwender bereit gestellt. Diesen können laut der Autoren mit der Werkzeugkette der jeweiligen Zielarchitektur bestimmt werden (z.B. mit einem Profiler). Die Autoren untersuchen in ihrer Arbeit für einen Beispiyalgorithmus (MPEG-1 Decoder) und drei verschiedene Digitaler Signalprozessor (DSP)-Zielarchitekturen (C62,C67,C55), bei welcher Taktfrequenz die geringste Energieaufnahme erreicht wird. Die Transitionen wurden hier nicht mit betrachtet, da diese laut der Autoren nur eine Offset-Verschiebung verursachen würden. Mit Hilfe des Modells konnten die Ausführungszeiten, die Leistungsaufnahme und somit auch die Energieaufnahme abgeschätzt werden. Diese wurden jedoch nicht validiert und die Generierung der für die Modellbildung benötigten Parameter wird nicht eindeutig beschrieben. Zudem sind die meisten Parameter nur für DSP-Architekturen relevant. Für eine untersuchte Mikrocontroller-Architektur (*ARM7TDMI*) lag die Variation für verschiedene Algorithmen unter 8 %. Somit können laut der Autoren die algorithmischen Parameter entfallen und die Architekturparameter reduzieren sich auf die Taktfrequenz F und den *memory mode* der Instruktionen [115]. Rethinagiri u. a. [172] verwenden die *Functional Level Power Analysis* (FLPA)-Methode [115], um die Leistungsaufnahme von spezifischen RISC-Prozessoren in einer generischen Funktion abzubilden. In diese Funktion gehen algorithmische Parameter wie *external memory access rate*, *cache miss rate* und *instruction per cycle rate* (IPC) sowie architekturenspezifische Parameter wie die Taktfrequenz des Prozessors und des Busses ein. Die Koeffizienten für jeden Prozessor werden entsprechend der FLPA-Methode durch die Vermessung verschiedener Basisprogramme und das anschließende *Curve Fitting* der Messdaten mittels Regression ermittelt. Die ermittelte Funktion für die Leistungsaufnahme eines *ARM Cortex-A8* lautet beispielsweise:

$$P = 0,79 f_{CORE} + 18,65 IPC + 0,26(\gamma_1 + \gamma_2) + 10,13 \quad (4.1)$$

Wobei γ_1 und γ_2 die *cache miss rates* für Level 1 und 2 Cache bezeichnen. In einem zweiten Schritt werden die algorithmischen Parameter wie die *miss rates* für spezifische Algorithmen ermittelt. Alle Informationen gehen in einen Simulator ein, der die OVP Bibliothek [91] nutzt. Auf diese Weise werden Fehler unter 3 % zwischen der Simulation und der Messung erreicht. Als Abgrenzung zum Stand der Technik führen die Autoren an, dass die Simulation, verglichen mit bestehenden Ansätzen, doppelt so schnell ist.

Jedoch erfolgt kein quantitativer Vergleich hinsichtlich der erreichten Genauigkeit. Da die benötigten Simulationszeiten bei allen Ansätzen im Bereich von Sekunden liegen, wäre zudem eine Betrachtung der erreichten Genauigkeit im Vergleich zu dem Aufwand für die Generierung des Modells relevanter.

4.2 Sensoren

Die Leistungsaufnahme der Betriebsmodi einzelner Sensoren und die Wahl der Abstrakte haben einen signifikanten Einfluss auf die Energieaufnahme der im System integrierten Sensorik. In Datenblättern wird oftmals mit einer sehr geringen Leistungsaufnahme des Sensors im aktiven Betrieb geworben, die im Bereich von wenigen μA liegt. Diese Angaben beziehen sich jedoch häufig auf eine durchschnittliche Leistungsaufnahme bei einer bestimmten Konfiguration. Beispielsweise die Messung eines einzelnen Wertes bei einer Ausführungsfrequenz von 1 Hz [201]. Die geringe Leistungsaufnahme entsteht daher durch den geringen Tastgrad des Sensors. Die eigentlich entscheidende Leistungsaufnahme während der Messung und die Dauer dieser Messung wird jedoch häufig nicht angegeben. Ausnahmen finden sich nur exemplarisch in den Datenblättern einiger Hersteller [29]. Dieses erschwert die Vergleichbarkeit verschiedener Sensoren anhand der Datenblätter. Zudem fehlt für eine Modellierung des Sensors eine einfach zugängliche Informationsquelle, so dass die Leistungsaufnahme und Dauer (und somit die Energieaufnahme) einer Messung oftmals vermessen werden müssen. Neben dem Tastgrad haben beispielsweise die Auflösung der Messung und der Messbereich des Sensors einen Einfluss auf die Energieaufnahme. Falls benötigte Informationen wie die Leistungsaufnahme der Modi und die Dauer der Operationen vorliegen, kann ein Sensor in der Modellierung entsprechend abgebildet werden.

4.3 Aktoren

Bei drahtlosen Sensorsystemen werden in den meisten Fällen keine Aktoren im klassischen Sinne (Antriebe) verwendet. Stattdessen finden sich hier Funktransceiver zur Übermittlung der Daten an externe System und integrierte Speichermodule zur direkten Speicherung der Daten auf dem mobilen System. Darüber hinaus finden sich häufig Komponenten zur Indikation (LEDs, Buzzer, Displays) von Betriebszuständen oder Ereignissen.

4.3.1 Funktransceiver

Die Modellierung der in Abschnitt 2.2 vorgestellten Funktechnologien umfasst in der Literatur im Wesentlichen die Leistungsaufnahme in den verschiedenen Betriebsmodi wie Senden, Empfangen, Verbindungsaufbau und inaktive Modi. Die Betrachtung für eine spezifische Funktechnologie oder auch für einen spezifischen Funktransceiver verfügt über einen höheren Detaillierungsgrad, da jeweils protokollspezifische Modi und Operationen berücksichtigt werden müssen, die sich darüber hinaus auch herstellerspezifisch unterscheiden können. Im Rahmen einer betreuten Masterarbeit [275] wurde beispielsweise

der *BLE112* BLE-Funktransceiver [22] der Firma *Bluegiga* vermessen und anschließend modelliert, so dass ein Zusammenhang zwischen der Sendefrequenz eines Datenpakets (*Notification*), der Breite der enthaltenen Nutzdaten und der mittleren Leistungsaufnahme des Funktransceivers für eine spezifische Sendestärke hergestellt werden konnte. Ein umfassendes und feingranulares Modell für diesen Transceiver findet sich darüber hinaus auch in der Arbeit von Kindt u. a. [105]. Der auf dem Körpersensor BG-V4.2 enthaltene *nRF24L01* Transceiver wurde detailliert von Christ u. a. [44] untersucht.

4.3.2 Speicher

Abzubildende Faktoren der Energieaufnahme von Speichermodulen sind die Leistungsaufnahme der Betriebsmodi Lesen, Schreiben, inaktive Modi und die Dauer der Operationen. Darüber hinaus muss auch der Einfluss der jeweiligen Architektur des Speichers und dessen Zugriffsschemata berücksichtigt werden. Signifikante Unterschiede finden sich in der Wahl der Speichertechnologie, so dass die in Abschnitt 2.2 beschriebenen gebräuchlichen NOR und NAND Flash-Module oder SD-Karten abzubilden sind. Eine Abschätzung der Energieaufnahme verschiedener SD-Karten findest du beispielsweise in [19].

4.4 Indikation

Für die Abbildung der zur Indikation verwendeten Komponenten wie LEDs, Buzzer (Piezo, magnetisch) und Vibrationsmotoren ist nur eine einfache Modellierung notwendig, da der eingeschaltete Zustand und ausgeschaltete Zustand mit der entsprechenden Leistungsaufnahme abgebildet werden können. Gegebenenfalls muss die Leistungsaufnahme im aktiven Modus abhängig von der Frequenz der Anregung betrachtet werden.

4.5 Schnittstellen

Die Leistungsaufnahme für sowohl analoge (ADC, DAC), als auch digitale Schnittstellen (I2C, *Serial Peripheral Interface* (SPI) und *Universal Asynchronous Receiver Transmitter* (UART)) kann oftmals dem Datenblatt des jeweiligen Mikrocontrollers entnommen werden. Eine Messung der Leistungsaufnahme der digitalen Schnittstellen I2C, SPI und UART mittels Shunt und eine anschließende Modellierung findet sich beispielsweise in [131]. Für jede Schnittstelle wird ein LUT generiert, der einen Zusammenhang zwischen der zu sendenden Datenbreite und der Energieaufnahme herstellt, wobei auch der jeweilige Protokoll-Overhead einbezogen wird.

4.6 Spannungswandler

Hinsichtlich der in Sensorsystemen gebräuchlichen Spannungswandler muss zwischen Schalt- und Linearreglern unterschieden werden, wobei die Modellierung als wichtigsten Parameter die Effizienz der Spannungswandlung abbilden muss. Die Effizienz eines Schaltreglers ist dabei von vielen Parametern abhängig [218] und ist bei einem Abwärtswandler beispielsweise auf die Verluste am Feldeffekttransistor (*MOSFET conduction loss* und *MOS-*

4. Modellierung drahtloser Sensorknoten

FET switching loss), der Diode (*body diode reverse recovery loss*), der Induktivität (*inductor loss*), der Kapazität (*capacitor loss*) und der Schaltlogik (*controller loss*) zurück zu führen. Der Einfluss der Verluste verschiebt sich mit der Last des Reglers, so dass bei niedriger Last die *MOSFET Switching* und *Gate Drive* Verluste bestimmend sind und bei hoher Last die *Inductor losses* und *MOSFET conduction losses* vorherrschend sind. Um dem entgegen zu wirken, wird in vielen Reglern bei niedriger Last vom Pulsweitenmodulation (PWM)- auf das Pulsfrequenzmodulation (PFM)-Verfahren umgestellt und somit die Schaltfrequenz reduziert. Die Umstellung erfolgt entweder automatisch oder mittels externem Eingang.

Die Modellierung eines Spannungswandler kann dabei auf verschiedenen Abstraktionsebenen erfolgen, welche von Benini u. a. [18] zusammen gefasst werden. Auf Schaltungsebene (*Transistor Level*) kann das Verhalten des Wandlers präzise nachgebildet werden, jedoch ist die Simulation auf dieser Ebene sehr zeit- und berechnungsaufwändig. Zur Reduktion der Komplexität kann das Verhalten des Wandlers mittels einem Ersatzschaltbild vereinfacht werden (*Behavioral White Box*). Eine deutliche Reduktion der Komplexität kann über eine Abstraktion des Verhaltens erreicht werden (*Behavioral Black Box*). In diesem Fall sind keine Informationen über die interne Struktur des Wandlers notwendig, sondern das Verhalten wird nur über die Effizienz im jeweiligen Arbeitspunkt bestehend aus Eingangsspannung, Ausgangsspannung und Ausgangsstrom abgebildet. Diese Informationen können häufig direkt dem jeweiligen Datenblatt entnommen werden, so dass das zu erwartende Verhalten des Wandlers für den Anwender einsehbar ist (vgl. Abbildung 4.1).

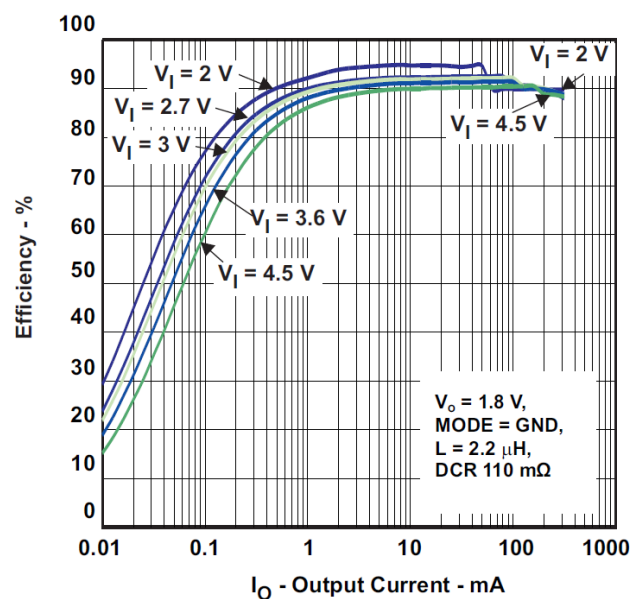


Abbildung 4.1.: Darstellung der Effizienz des *Texas Instruments TPS62243* Spannungswandlers im Datenblatt [206]. *Courtesy of Texas Instruments.*

Für einen Schaltregler kann der Zusammenhang zwischen der Effizienz η , der Eingangsspannung U_I , dem Eingangsstrom I_I , der Ausgangsspannung U_O und dem Ausgangsstrom I_O mittels der Gleichung 4.2 hergestellt werden [153].

$$\eta = \frac{U_O * I_O}{U_I * I_I} \quad (4.2)$$

Für die Modellierung eines Spannungswandlers kann die Effizienz im jeweiligen Arbeitspunkt als LUT abgebildet werden. Die Effizienz ist somit in einem mathematischen Modell als Funktion von Eingangsspannung und Ausgangsstrom (bei fester Ausgangsspannung) hinterlegt [18], wobei eine lineare Interpolation mit 20 Wertepaaren bereits ausreichend sein kann [188].

Dem gegenüber ist die Effizienz eines Linearreglers in Gleichung 4.3 dargestellt. Da der Eingangsstrom I_I dem Ausgangsstrom I_O entspricht [210], hängt die Effizienz η nur von dem Verhältnis der Eingangsspannung U_I zur Ausgangsspannung U_O ab.

$$\eta = \frac{U_O * I_O}{U_I * I_I} = \frac{U_O}{U_I} \quad (4.3)$$

4.7 Energiespeicher

Die Modellierung der in Abschnitt 2.2 beschriebenen Eigenschaften einer Batterie kann auf verschiedenen Abstraktionsebenen erfolgen. Zielgröße der Modellierung beziehungsweise der Simulation von Batterien ist in den meisten Fällen die Abschätzung der Entladung auf Basis eines bestimmten Entladeprofils und somit die Bestimmung der Laufzeit eines Systems. Bei idealer Batterie und konstanter Entladung kann die Laufzeit mit Hilfe von Gleichung 4.4 berechnet werden, wobei I_{BAT} dem Entladestrom der Batterie entspricht.

$$T_{\text{SYS}} = \frac{C_{\text{BAT}}}{I_{\text{BAT}}} \quad (4.4)$$

Eine konstante Entladung einer Batterie wird häufig mit Hilfe der Entladerate angegeben, die als Vielfaches der Kapazität angegeben wird. So würde eine Batterie mit einer Kapazität C_{BAT} von 120 mAh bei einer Entladerate von 0,5 C mit einem konstantem Entladestrom von 60 mA entladen werden und wäre, bei Annahme einer idealen Batterie, nach 2 Stunden entladen.

Jedoch kann die gespeicherte Energie einer Batterie nicht voll abgerufen und eine reale Batterie somit nicht als ideal angenommen werden. Die reale Laufzeit des Systems verringert sich dementsprechend. So hängt die Laufzeit vor allem von dem Entladestrom ab, wobei ein höherer Entladestrom die Effizienz der Entladung verringert. Ein Überblick über auftretende Batterieeffekte wird von Benini u. a. [18] gegeben, wobei zwischen Effekten erster und zweiter Ordnung unterschieden werden kann. Effekte erster Ordnung sind:

- Die nutzbare Kapazität hängt von der Entladung ab. Bei hohem Entladestrom kann die chemisch gespeicherte Energie nicht effizient in elektrische Energie umgesetzt werden und es kommt zu höheren Verlusten.
- Die Batteriespannung U_{BAT} hängt nicht-linear vom Ladezustand ab (SOC). Die Batteriespannung nimmt mit dem SOC ab.
- Die Frequenz der Entladung beeinflusst die nutzbare Kapazität. Die Batterie reagiert nur träge auf eine Änderung der Last. Dieses ist bedingt durch große Zeitkonstanten in den elektrochemischen Vorgängen.

Die Effekte zweiter Ordnung umfassen darüber hinaus:

- Die Temperaturabhängigkeit der Entladung. Die Erhitzung der Zelle führt zu einer Verschiebung der Ausgangsspannung, was vor allem bei hohen Entladeströmen relevant ist.
- Die Veränderung des Innenwiderstands der Batterie. So wird die Ausgangsspannung unter Last um den resistiven Spannungsabfall reduziert.

Hinsichtlich der Modellierungsebene kann zwischen elektrochemischen, mathematischen und elektrischen Modellen unterschieden werden [38]. Während elektrochemische Modelle in ihrer Anwendung sehr komplex sind und lange Simulationszeiten erfordern, sind mathematische Modelle hingegen oft sehr abstrakt und decken nicht alle Effekte ab. Für spezifische Anwendungen liegt der Fehler mathematischer Modelle im Bereich von 5-20 % [38]. Elektrische Modelle hingegen weisen einen Fehler im Bereich von 1-5 % auf [38] und liegen hinsichtlich ihrer Präzision zwischen mathematischen und elektrochemischen Modellen.

Im Folgenden wird der aktuelle Stand im Bereich der mathematischen und elektrischen Modell näher betrachtet. Chemische Modelle werden nicht näher betrachtet, da diese für eine Modellierung und Simulation des Systems nach derzeitigem Stand nicht praktikabel sind.

4.7.1 Mathematische Modelle

Einen einfachen Ansatz zur Abbildung des Zusammenhangs zwischen nutzbarer Kapazität und Entladestrom stellt die in Gleichung 4.5 dargestellte Peukert-Gleichung [155] dar, welche die Laufzeit der Batterie in Abhängigkeit vom Entladestrom beschreibt.

$$T_{\text{SYS}} = \frac{C_{\text{PK}}}{I_{\text{BAT}}^k} \quad (4.5)$$

Die Peukert-Kapazität C_{PK} ist die Nennkapazität bei einem Entladestrom von 1 A. Der Exponent k ist die dimensionslose Peukert-Zahl und stellt den Zusammenhang zwischen der zur Verfügung stehenden Kapazität und dem Entladestrom I_{BAT} her. Da die Nennkapazität

einer Batterie in den meisten Fällen nicht für einen Entladestrom von 1 A angegeben wird, kann die Peukert-Gleichung entsprechend angepasst werden, so dass sich Gleichung 4.6 ergibt:

$$T_{\text{SYS}} = \frac{C_{\text{BAT}}}{I_N} \cdot \left(\frac{I_N}{I_{\text{BAT}}} \right)^k \quad (4.6)$$

Hier entspricht C_{BAT} der vom Hersteller angegebenen Nennkapazität bei Nennentladestrom I_N . Somit kann mit der für jeden Batterietyp spezifischen Peukert-Zahl k bei gegebenen Entladestrom I_{BAT} die Laufzeit der Batterie T_{SYS} berechnet werden. Bei einer idealen Batterie ist $k = 1$, so dass sich Gleichung 4.4 ergibt. Jedoch wurde die Peukert-Gleichung ursprünglich für Bleiakkumulatoren definiert und ist für heutige Energiespeicher und deren Anwendungen oftmals nicht mehr ausreichend [53; 144]. Extremfälle wie sehr hohe oder niedrige Entladeströme werden im Modell nicht abgedeckt, so dass es außerhalb der spezifizierten Anwendung zu hohen Abweichungen kommen kann. Für zeitveränderliche Lasten kann die Peukert-Gleichung einen Fehler von bis zu 100 % aufweisen [166].

In der weiteren Entwicklung von Batteriemodellen wird daher angestrebt, alle relevanten Effekte abzubilden und die Präzision der Modellierung zu verbessern. So erweitert das Modell von Pedram und Wu [153] die Peukert-Gleichung um einen Effizienzfaktor, welcher aus der Wahrscheinlichkeitsdichtefunktion des Entladeprofils abgeleitet wurde und somit die Verteilung des Entladestroms berücksichtigt. So konnte gezeigt werden, dass Entladeprofile mit gleichem durchschnittlichen Entladestrom aber verschiedener Verteilungen zu unterschiedlichen Laufzeiten der Batterie führen. Die maximale Laufzeit wird bei minimaler Varianz, also konstanter Entladung, erreicht.

Aktuelle Arbeiten von Y. Chen u. a. [42] befassen sich mit der Charakterisierung von Batterien anhand ihres Frequenzverhaltens. Es wird zunächst die Generierung von Ragone-Diagrammen aus dem Datenblatt betrachtet, wobei ein Ragone-Diagramm den Zusammenhang zwischen der Energiedichte und der Leistungsdichte eines Energiespeichers zeigt. Zur Generierung werden Entladekurven für mindestens zwei verschiedene Entladeströme benötigt. Um das Verhalten der Batterien in Abhängigkeit von der Frequenz der Entladung zu betrachten, werden die Ragone-Diagramme in den Frequenzbereich überführt. Somit wird ein Zusammenhang zwischen der Kapazität der Batterie und der Frequenz der Entladung hergestellt. Die Betrachtung im Frequenzbereich erfolgt über die spektrale Leistungsdichte (*energy spectral density*) des Entladeprofils. Die Integration und Normalisierung des Spektrums liefert die kumulative Verteilung. In dieser Verteilung wird eine Grenzfrequenz bestimmt, wobei beispielsweise 90 % der Entladefrequenzen unter dieser Grenzfrequenz liegen. Als Anwendungsbeispiel werden zwei verschiedene Entladeprofile mit ähnlichem durchschnittlichen Entladestrom, aber unterschiedlichem dynamischen Verhalten betrachtet. Bei Auswahl eines Energiespeichers nur anhand des durchschnittlichen Entladestroms würde der gleiche Batterietyp ausgewählt werden. Die Auswahl anhand der Grenzfrequenz liefert zwei verschiedene Batterietypen, wobei beide von dem anhand

des durchschnittlichen Entladestroms ausgewählten Batterietyps verschieden sind. Jedoch erfolgt kein Nachweis, ob die anhand des Frequenzverhaltens ausgewählten Batterietypen tatsächlich besser geeignet sind. In einer folgenden Arbeit erweitern Y. Chen u. a. [40] das Modell von Pedram und Wu [153] um einen Korrekturterm für die Frequenzverteilung des Entladestroms, so dass Varianz und Frequenz der Last berücksichtigt werden. Das entwickelte Modell wird zur Auslegung der Batterie für ein System verwendet, wobei die zuvor publizierte Charakterisierung der Batterien anhand ihres Frequenzverhaltens [42] die Basis bildet. Die Batteriekapazität kann somit in Abhängigkeit von Varianz und Frequenz des Entladestroms betrachtet werden. Die Validierung dieses mathematischen Modells erfolgt mit einem selbst entwickelten elektrischen Modell [39], wobei drei künstliche und zwei reale Entladeprofile untersucht wurden. Jedoch wurde dieses elektrische Modell [39] bisher nicht validiert, so dass keine valide Grundlage für das publizierte mathematische Modell [40] vorliegt.

4.7.2 Elektrische Modelle

Effekte erster und zweiter Ordnung können als Ersatzschaltbilder bestehend aus Spannungsquellen, Widerständen und Kondensatoren abgebildet werden. Somit ist auch eine Ko-Simulation mit anderen Schaltungen möglich. Aufgrund der guten Anwendbarkeit bei gleichzeitig relativ geringem Fehler sind elektrische Modelle potentiell für eine Modellierung des Energiespeichers drahtloser Sensorknoten geeignet. Im Folgenden werden daher verbreitete Modelle näher vorgestellt und die Entwicklung in diesem Bereich aufgezeigt. Die Bandbreite der Modelle reicht dabei von einfachen Basismodellen, die keine oder nur wenige der beschriebenen Batterieeffekte abdecken bis hin zu komplexen Modellen, die eine Vielzahl von Effekten abdecken.

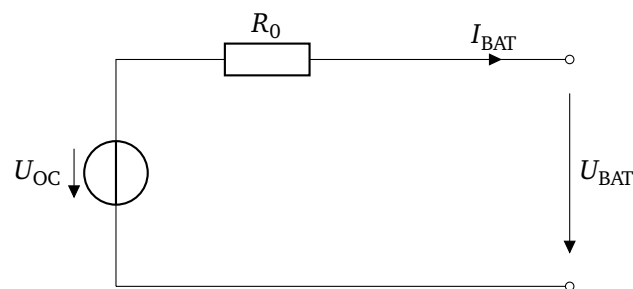


Abbildung 4.2.: Rint-Batteriemodell nach He u. a. [75]

Einfache Modelle Das in Abbildung 4.2 dargestellte einfache Rint-Batteriemodell [75] berücksichtigt nur den Innenwiderstand der Batterie und dessen Einfluss auf die Klemmenspannung. Hier bezeichnet U_{OC} die Leerlaufspannung der Batterie (*open circuit voltage*), U_{BAT} die Klemmenspannung und R_0 den Innenwiderstand. Die Klemmenspannung kann

demnach wie in Gleichung 4.7 dargestellt berechnet werden.

$$U_{\text{BAT}} = U_{\text{OC}} - I_{\text{BAT}} \cdot R_0 \quad (4.7)$$

Thevenin-basierte Modelle Während einfache Modelle nur den Innenwiderstand der Batterie berücksichtigen, zeichnen sich sogenannte Thevenin-Modelle durch die Berücksichtigung weiterer Parameter aus. So berücksichtigt das in Abbildung 4.3 dargestellte und in [14] und [120] beschriebene Modell neben dem Innenwiderstand und der Leerlaufspannung auch den Elektrolytwiderstand R_p und die Plattenkapazität C_p der Batterie.

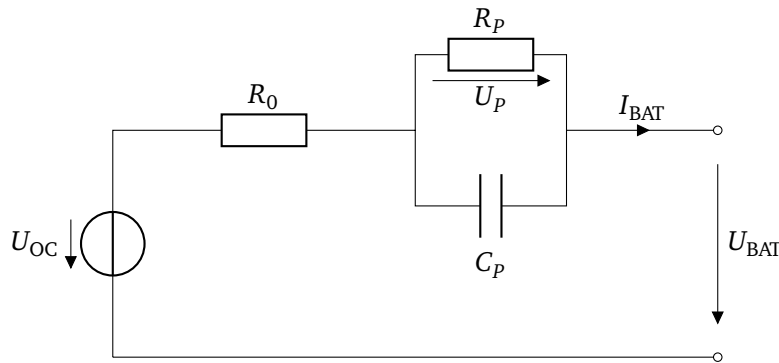


Abbildung 4.3.: Thevenin Batteriemodell 1. Ordnung nach Liu u. a. [120]

Diese beschreiben das Einschwingverhalten (*transient response*) der Batterie bei plötzlicher Änderung der Last [75]. Die über dem RC-Netzwerk abfallende Spannung U_p wird auch als Polarisierungsspannung bezeichnet. Sie bildet den durch Polarisierung bedingten Spannungsabfall beim Laden und Entladen in Bezug zur Leerlaufspannung U_{OC} ab [100]. Darüber hinaus sind Leerlaufspannung und Innenwiderstand abhängig vom Ladezustand der Batterie (SOC). So sinkt die Leerlaufspannung U_{OC} mit der in der Batterie gespeicherten Energie. Der Innenwiderstand R_0 hingegen nimmt während der Entladung zu. Somit ergibt sich für das Thevenin-Modell die Klemmenspannung der Batterie zu [75]:

$$U_{\text{BAT}} = U_{\text{OC}} - U_p - I_{\text{BAT}} \cdot R_0 \quad (4.8)$$

Wobei:

$$\dot{U}_p = \frac{I_{\text{BAT}}}{C_p} - \frac{U_p}{R_p C_p} \quad (4.9)$$

Der Elektrolytwiderstand R_p und die Plattenkapazität C_p sind für eine spezifische Batterie konstant. Das Modell wird von Badam u. a. [14] für konstante Entladungen mit einer Genauigkeit von 97,5 % validiert. Als Referenz dienten hier professionelle Batterietestsysteme, wobei die Parameter Leerlaufspannung, Innenwiderstand, Elektrolytwiderstand und Plattenkapazität zuvor mit den gleichen Testsystemen experimentell ermittelt wurden.

4. Modellierung drahtloser Sensorknoten

Von Liu u. a. [120] wird ein Vorgehen zur diskreten Berechnung der Parameter definiert, wobei diskrete Formeln für den SOC, die Spannung des RC-Netzwerks U_p und die Klemmenspannung U_{BAT} definiert werden. Der SOC in jedem Schritt berechnet sich aus dem SOC des vorherigen Schritts, der nominalen Kapazität C_{BAT} , dem Entladestrom I_{BAT} und dem Abtastintervall T_s . Die Temperaturabhängigkeit von R_0 und die Abhängigkeit der Klemmenspannung U_{BAT} vom SOC werden mit Hilfe von LUT abgebildet. Die Generierung der LUT wird in dieser Arbeit jedoch nicht beschrieben.

Eine Erweiterung des Thevenin-Modells stellt das von He u. a. [75] beschriebene und in Abbildung 4.4 dargestellte DP-Modell (*double polarization* oder auch *dual polarization* Modell) dar.

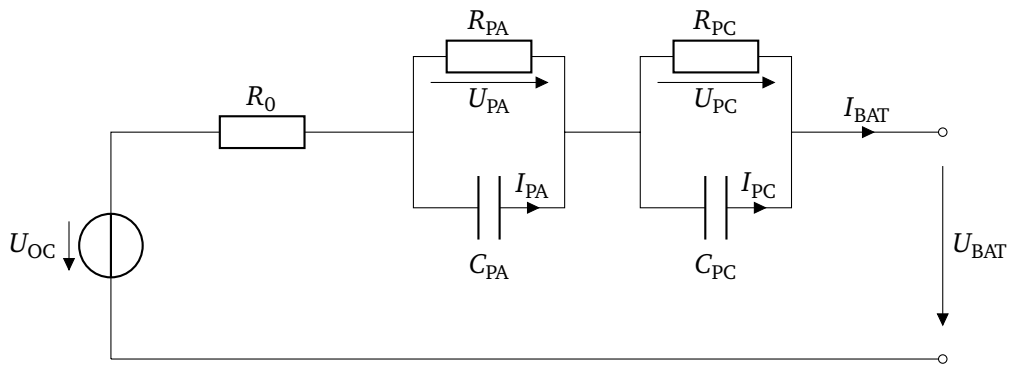


Abbildung 4.4.: DP-Modell nach He u. a. [75]

Dieses bildet, ebenso wie das Thevenin-Modell, das Einschwingverhalten der Batterie ab. Jedoch wird das Verhalten hier differenzierter betrachtet und mit Hilfe von zwei RC-Netzwerken abgebildet. Die Klemmenspannung der Batterie ergibt sich für das DP-Modell zu [76]:

$$U_{\text{BAT}} = U_{\text{OC}} - U_{\text{PA}} - U_{\text{PC}} - I_{\text{BAT}} \cdot R_0 \quad (4.10)$$

Wobei:

$$\dot{U}_{\text{PA}} = \frac{I_{\text{BAT}}}{C_{\text{PA}}} - \frac{U_{\text{PA}}}{R_{\text{PA}} C_{\text{PA}}} \quad (4.11)$$

Und:

$$\dot{U}_{\text{PC}} = \frac{I_{\text{BAT}}}{C_{\text{PC}}} - \frac{U_{\text{PC}}}{R_{\text{PC}} C_{\text{PC}}} \quad (4.12)$$

So charakterisieren R_{PA} und C_{PA} das Kurzzeitverhalten der Batteriespannung aufgrund der elektrochemischen Polarisation. Das RC-Netzwerk aus R_{PC} und C_{PC} hingegen bildet das Langzeitverhalten aufgrund der Konzentrationspolarisation im Elektrolyt [154] ab. Der Zeitraum bis $5 \cdot \tau_{\text{PA}} = 5 \cdot R_{\text{PA}} C_{\text{PA}}$ wird als Kurzzeit definiert. Das beschriebene Verhalten der Klemmenspannung bei einem Entladepuls ist schematisch in Abbildung 4.5 dargestellt. Die Bezeichnung τ_s in der Darstellung entspricht τ_{PA} und der Zeitraum τ_L entspricht τ_{PC} .

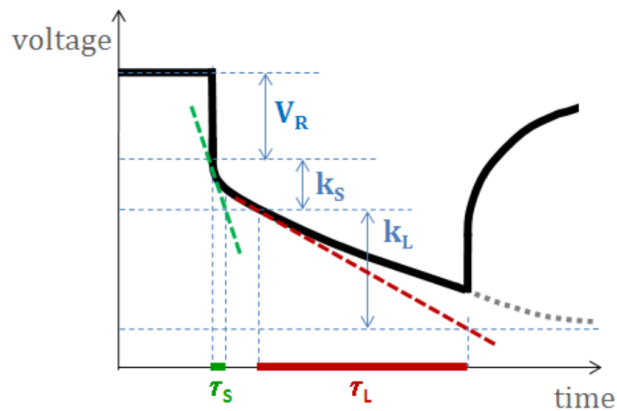


Abbildung 4.5.: Verlauf der Klemmenspannung einer Batterie in Folge eines Entladepulses [24]. ©2018 IEEE.

Laufzeit-basierte Modelle Laut M. Chen und Rincon-Mora [38] sind Thevenin-Modelle in der Lage, das Batterieverhalten bei veränderlicher Last (*transient load events*) für einen spezifischen SOC abzubilden. Jedoch kann somit die vom SOC abhängige Leerlaufspannung U_{OC} im Modell nur als konstant angenommen werden, was die Erfassung von Spannungsänderungen im stationären Zustand (*steady state voltage response, DC response*) und die Betrachtung der Laufzeit der Batterie verhindert. Demgegenüber sind sogenannte Laufzeit-Modelle (*Runtime-based model*) in der Lage, diese Aspekte mit Hilfe eines elektrischen Modells abzubilden.

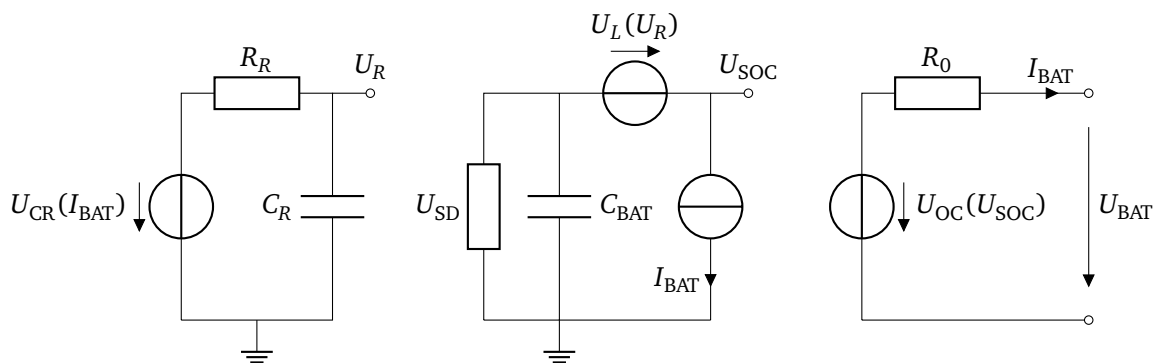


Abbildung 4.6.: Laufzeit-Batteriemodell nach M. Chen und Rincon-Mora [38]

Das in Abbildung 4.6 dargestellte Laufzeit-Modell ist in drei Schaltungsteile aufgeteilt. Der linke Teil bildet die Abhängigkeit des SOC von der Entladerate und der Entladefrequenz

4. Modellierung drahtloser Sensorknoten

ab. So bilden C_R und R_R einen Tiefpassfilter, wobei die Zeitkonstante $\tau = R_r \cdot C_r$ der elektrochemischen Verzögerung der Batterie entspricht. Der Entladestrom der Batterie I_{BAT} geht als Abhängigkeit in U_{CR} ein, so dass das Potential U_R von der Entladerate und der Entladefrequenz abhängt. U_R wiederum geht im mittleren Teil als Abhängigkeit in U_L ein. Zudem wird mit dem Widerstand U_{SD} die Selbstentladung der Batterie abgebildet. Das Potential U_{SOC} hängt zudem von der Entladerate und dem SOC der Batterie ab. U_{SOC} geht schlussendlich als Abhängigkeit in die Leerlaufspannung U_{OC} ein, so dass sich mit dem Innenwiderstand R_0 die Klemmenspannung U_{BAT} ergibt.

Eine weitere Beschreibung eines Laufzeit-Modells findet sich bei Benini u. a. [18], welches auf der Arbeit von Gold [71] basiert (vgl. Abbildung 4.7). Im Vergleich zu dem in Abbildung 4.6 dargestellten Modell wird der Widerstand der Selbstentladung U_{SD} und der Innenwiderstand R_0 nicht berücksichtigt. Diese Variante wird von den Autoren als Laufzeit-Modell erster Ordnung definiert. Die abgebildeten Effekte umfassen somit die Abhängigkeit der Klemmenspannung von der Entladerate, der Entladefrequenz und dem SOC. Zusätzlich beschreiben Benini u. a. [18] ein Laufzeitmodell zweiter Ordnung, welches darüber hinaus die externe Temperatur und die Änderung des Innenwiderstands der Batterie als die beiden wichtigsten Effekte zweiter Ordnung einbezieht.

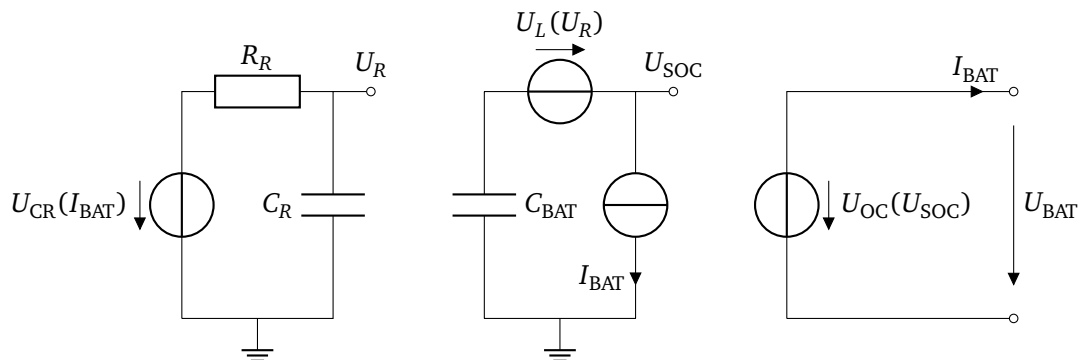


Abbildung 4.7.: Laufzeit-Batteriemodell 1. Ordnung nach Benini u. a. [18]

Diese Effekte sind vor allem für Batterien mit großer Kapazität entscheidend, da hohe Entladeraten einen hohen Spannungsabfall über den Innenwiderstand und dementsprechend eine signifikante Selbsterwärmung verursachen. Im Anschluss an die Beschreibung der grundlegenden Laufzeit-Modelle werden die zur Beschreibung des Batteriehaltens benötigten Differentialgleichungen diskretisiert und in *Very High Speed Integrated Circuit Hardware Description Language* (VHDL) implementiert. Die Validierung erfolgt mit dem zugrunde liegenden kontinuierlichen Laufzeit-Modell von Gold [71], welches als *Simulation Program with Integrated Circuit Emphasis* (SPICE) Modell implementiert wurde.

Kombinierte Modelle Das von M. Chen und Rincon-Mora [38] vorgestellte Modell (vgl. Abbildung 4.8) ist eine Kombination aus verschiedenen elektrischen Modellen. So orientiert sich der linke Teil am mittleren Teil des Laufzeit-Modells, welches den SOC und somit die Laufzeit der Batterie in Abhängigkeit vom Entladestrom abbildet. Der rechte Teil mit zwei RC-Netzwerken orientiert sich am DP-Modell und bildet das Einschwingverhalten ab. Alle Parameter im Modell sind multivariable Funktionen vom SOC, dem Entladestrom, der Temperatur und der Zyklenanzahl, so dass die Extraktion der benötigten Modellparameter komplex ist. Zudem müssen die Parameter durch Messung einzelner Batterien empirisch ermittelt werden, so dass eine Berücksichtigung aller Einflussfaktoren sehr zeitaufwändig wäre. Die Faktoren Zyklenanzahl und Temperatur werden daher von den Autoren vernachlässigt. Die Bestimmung der Parameter erfolgt von M. Chen und Rincon-Mora [38] für eine einzelne Batterie unter Variation des SOC und der Entladerate. Abschließend wird das modellierte Verhalten der Batterie für drei verschiedene Entladeprofile (kontinuierlich, konstante Pulse, veränderliche Pulse) mit der vermessenen realen Entladung verglichen, wobei bezüglich der Laufzeit der Batterie ein Fehler von unter 0,4 % erreicht wird.

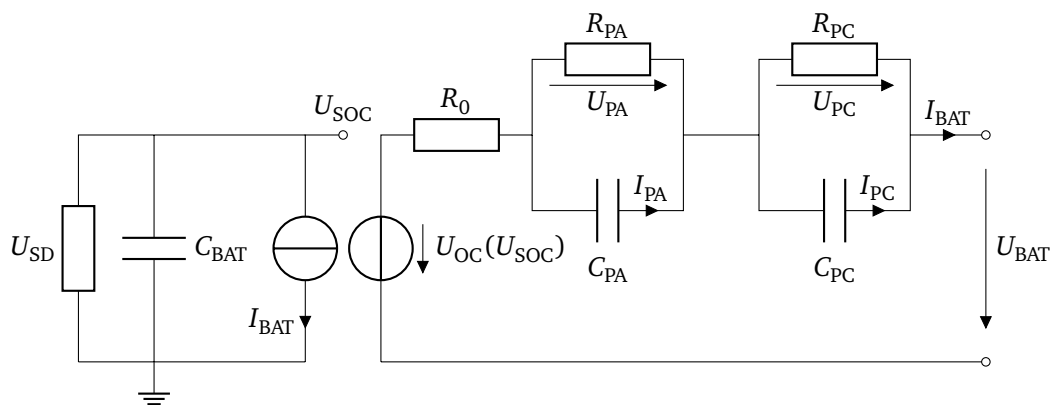


Abbildung 4.8.: Kombiniertes Modell nach M. Chen und Rincon-Mora [38]

Laut Y. Chen u. a. [39] können mit dem Modell von M. Chen und Rincon-Mora [38] zwar dynamische (z.B. gepulste) Entladeprofile abgebildet werden, jedoch berücksichtigt der Ansatz nur die Varianz des Entladeprofils und nicht dessen Frequenz. Verschiedene Entladeprofile mit gleichem mittlerem Entladestrom würden im Modell somit zur gleichen Laufzeit führen. Um eine Aussage über die Kapazität des Energiespeichers in Abhängigkeit von der Entladefrequenz zu ermöglichen wurde von Y. Chen u. a. [39] daher ein Modell vorgestellt, welches eine eigens entwickelte Methode zur Charakterisierung des Frequenzverhaltens [42] verwendet und mit diesem Ansatz das elektrische Modell von M. Chen und Rincon-Mora [38] erweitert. Das resultierende Modell ist in Abbildung 4.9 dargestellt.

Um die Varianz des Entladestroms abzubilden, wurde die im Laufzeitmodell von Benini

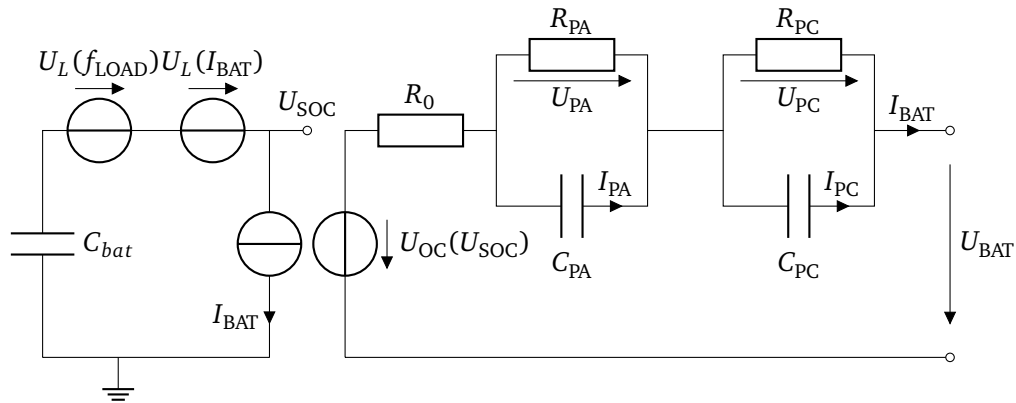


Abbildung 4.9.: Kombiniertes Modell nach Y. Chen u. a. [39]

u. a. [18] beschriebene Spannungsquelle U_L im linken Schaltungsteil eingefügt, wobei diese hier direkt vom Entladestrom I_{BAT} abhängt. Zusätzlich wurde eine weitere Spannungsquelle U_L in Abhängigkeit der Frequenz des Entladeprofils f_{LOAD} eingefügt, die einen frequenzabhängigen Spannungsabfall bewirkt. Somit kann sowohl die Varianz des Entladestroms, als auch dessen Frequenz abgebildet werden. Beide Spannungsquellen können entsprechend dem eigenen Verfahren [42] über die Abhängigkeit der Kapazität von der Varianz und Frequenz des Entladestroms bestimmt werden. Im Gegensatz zur vorherigen Arbeit fließen diese hier nicht in ein mathematisches Modell zur Dimensionierung der Batterie [40] (vgl. Unterabschnitt 4.7.1), sondern in ein elektrisches Modell ein. Mittels Simulation konnte gezeigt werden, dass eine höhere Frequenz des Entladeprofils zu einer reduzierten Laufzeit der Batterie führt. Jedoch wurden die Simulationen nicht durch reale Messungen validiert, so dass keine Aussage darüber getroffen werden kann, ob der Einfluss der Frequenz des Entladeprofils im Modell korrekt wieder gegeben wird. So könnte die elektrochemische Verzögerung der Batterie, die im Laufzeit-Modell [18; 71] als Tiefpass abgebildet wird und hier nicht berücksichtigt wird einen signifikanten Einfluss haben.

4.7.3 Generierung der Modelle

Die zuvor beschriebenen elektrischen Modelle sind zwar in die Lage, das Verhalten der Batterie abzubilden, jedoch müssen die benötigten Parameter aufwändig gemessen werden und sind daher oft nur für eine spezifische Batterie gültig. Eltoumi u. a. [56] ermitteln die für das DP-Modell benötigten Parameter R_0 , R_{PA} , R_{PC} , C_{PA} und C_{PC} experimentell. So wird mit Hilfe einer elektronischen Last für die zu modellierende Lithium-Ionen Zelle ein Zyklus aus einer Ladephase mit konstantem Ladestrom (*Constant Current* (CC)), einer Entspannungsphase, einer Entladephase mit konstantem Entladestrom und einer abschließenden Entspannungsphase gefahren und währenddessen die Klemmenspannung der Batterie gemessen. Ein zu beobachtender unmittelbarer Abfall der Klemmenspannung ist auf den

Innenwiderstand der Batterie zurück zu führen. Der interne elektrochemische Prozess einer Batterie führt zu einem darauf folgenden abnehmendem exponentiellen Verlauf (*transient response*). Der Gesamtverlauf hängt von der Höhe des Entladestroms [180] und dem SOC [154] der Batterie ab. Der nichtlineare Verlauf der Klemmenspannung wird mittels einer Exponentialfunktion abgebildet und die Parameter des Modells durch ein *Curve Fitting* bestimmt.

He u. a. [76] ermitteln die Parameter für sieben verschiedene Modelle (u.a. Rint-, Thevenin- und DP-Modell) mit Hilfe eines professionellen Batterieprüfsystems ebenfalls experimentell. Anschließend wird für jedes Modell die Klemmenspannung mittels *Recursive-Least-Squares-Algorithmus* ermittelt und mit der gemessenen Klemmenspannung verglichen.

Cacciato u. a. [37] verwenden für ihren Ansatz das Modell von M. Chen und Rincon-Mora [38], wobei das Modell zunächst auf Basis der Datenblätter initialisiert wird. Anschließend wird das Batteriemodell mit der gemessenen Spannung verglichen (*Proportional-Integral-Observer*) und das Batteriemodell zur Laufzeit verbessert.

Ziel der Arbeit von Tian u. a. [208] ist die Generierung aller Parameter eines DP-Modells, wobei nicht nur die Werte der Widerstände und Kapazitäten, sondern auch der Zusammenhang zwischen Klemmenspannung und SOC in einem einzigen Verfahren ermittelt werden soll. So werden die Gleichungen 4.10 bis 4.12 um die Änderung des SOC, die Klemmenspannung und den Innenwiderstand (in Abhängigkeit des SOC) sowie die Spannungen U_{PA} und U_{PC} (in Abhängigkeit der Entladerate) zu einer Gesamtgleichung für U_{BAT} verbunden. Basierend auf der Gesamtgleichung folgt zunächst ein Vortest, in dem ermittelt wird, ob U_{BAT} sensitiv genug ist, um alle Parameter eindeutig zu identifizieren (*identifiability analysis*). Anschließend wird ein Verfahren zur Bestimmung der Parameter des Modells mittels nichtlinearer Regressionsanalyse vorgestellt. Dieses Verfahren wird abschließend exemplarisch auf vier Batterien eines öffentlich verfügbaren Datensatzes angewendet, dessen Daten die vollständige Entladung der Batterien mit konstantem Entladestrom umfasst. Für die Regressionsanalyse wurden zuvor untere und obere Schranke sowie eine erste Schätzung der Parameter definiert, die auf typischen Werten basiert. Es konnte gezeigt werden, dass die generierten Modelle eine gute Übereinstimmung mit den Messwerten erreichen und somit für die vier exemplarischen betrachteten Batterien zuverlässig den Zusammenhang zwischen dem SOC und der Klemmenspannung geschätzt werden kann. Zhang u. a. [237] hingegen schätzen den Zusammenhang zwischen dem SOC und der Leerlaufspannung für ein Thevenin-Modell mittels Kalman-Filter.

Zusammenfassend besteht das übliche Vorgehen für die Generierung der Modellparameter zunächst in der Messung ausgewählter Batterien oder der Verwendung bereits vorliegender Datensätze. Anschließend werden die Parameter mittels verschiedener Verfahren aus den vorliegenden Messdaten ermittelt. Somit können in die Modellierung des Systems nur Batterien einbezogen werden, für die bereits geeignete Messdaten vorliegen.

Hier setzen aktuelle Arbeiten an, die bestrebt sind, das Batteriemodell aus den Informationen des Datenblatts zu generieren. So definieren Petricca u. a. [154] eine Vorgehens-

weise zur Modellgenerierung mit verschiedenen Modellebenen. In einem ersten Schritt wird überprüft, welche Informationen im Datenblatt vorliegen und die entsprechende Modellebene ausgewählt. Dieses ist notwendig, da nicht in jedem Datenblatt die benötigten Informationen für komplexe Modelle enthalten sind. Im nächsten Schritt müssen die Daten in eine vorgegebene Datenstruktur übertragen werden, woraufhin das Modell automatisch generiert wird. Abschließend kann das Verhalten des generierten Modells mit den Angaben im Datenblatt verglichen oder, falls möglich, mittels Messung validiert werden. Im Folgenden werden die definierten Modellebenen vorgestellt, wobei auf die für die jeweilige Ebene benötigten Angaben aus dem Datenblatt und die Generierung des Modells eingegangen wird:

- **Ebene 1:** Mathematisches Modell nach Peukert [155] (vgl. Unterabschnitt 4.7.1). Zur Generierung wird das Verhalten der Klemmenspannung über der Zeit für verschiedene konstante Entladeströme benötigt. Die Punkte, in welchen jeweils die Entladeschlussspannung erreicht wird, bestimmen die Laufzeit der Batterie, so dass sich mehrere Datenpaare aus Entladestrom und Laufzeit ergeben. Auf Basis dieser Datenpaare werden mittels *Curve Fitting* die für das Modell benötigten Parameter Peukert-Kapazität C_{PK} und Peukert-Zahl k generiert (vgl. Abbildung 4.10). Laut der Autoren sind für die Modellgenerierung bereits 4-5 Wertepaare ausreichend.

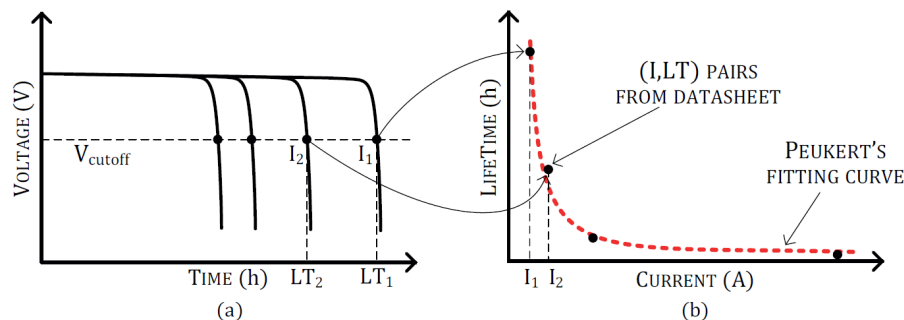


Abbildung 4.10.: Generierung eines Batteriemodells der Ebene 1 nach Petricca u. a. [154].
©2013 IEEE

- **Ebene 2 und 2.5:** Vereinfachtes Laufzeit-Modell basierend auf Benini u. a. [18] (vgl. Unterabschnitt 4.7.2). Zur Generierung des in Abbildung 4.11 dargestellten Modells wird das Verhalten der Klemmenspannung über dem SOC benötigt. Dieses wird aus dem Datenblatt entnommen und mittels *Curve Fitting* angenähert. Für den Fall, dass der Zusammenhang zwischen Klemmenspannung und SOC für verschiedene Entladeströme angegeben wird, können mehrere Funktionen generiert werden. Somit kann auch der Einfluss des Innenwiderstandes R_0 abgebildet werden

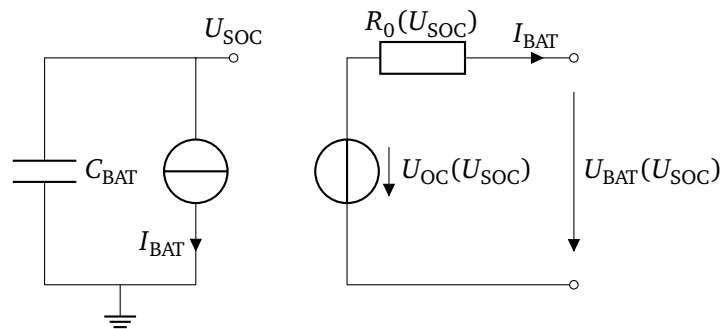


Abbildung 4.11.: Von Petricca u. a. [154] verwendetes vereinfachtes Laufzeit-Batteriemodell nach Benini u. a. [18]

(Ebene 2.5). Andernfalls kann der Innenwiderstand nicht berücksichtigt werden (Ebene 2). Die im Ersatzschaltbild enthaltene Kapazität C_{BAT} ergibt sich direkt aus der nominalen Kapazität der Batterie umgerechnet in Amperesekunden (As). Mit aus der Schaltung abgeleiteten Formeln für die Klemmenspannung und den Innenwiderstand kann schlussendlich auf das Verhalten der Leerlaufspannung U_{OC} in Abhängigkeit vom SOC geschlossen werden.

- **Ebene 3:** Kombiniertes Modell nach M. Chen und Rincon-Mora [38] (vgl. Unterabschnitt 4.7.2). Das in Abbildung 4.8 dargestellte kombinierte Modell verfügt im Vergleich zu dem in Abbildung 4.11 dargestellten Laufzeit-Modell zusätzlich über zwei RC-Netzwerke im rechten Teil, mit denen das dynamische Verhalten der Batterie (*transient response*) abgebildet werden kann. Neben den für die Modellebene 2,5 benötigten Angaben wird hier zusätzlich das Verhalten der Klemmenspannung über der Zeit für ein gepulstes Entladeprofil benötigt. Aus dem zuvor beschriebenen sofortigen Spannungsabfall durch den Innenwiderstand und dem exponentiellen Abfall durch elektrochemische Prozesse können die benötigten Parameter abgeleitet werden. So werden die Spannungsabfälle U_{RO} , U_{PA} und U_{PC} sowie die Zeitkonstanten τ_{PA} und τ_{PC} aus den Daten ermittelt und die Parameter R_{PA} , R_{PC} , C_{PA} und C_{PC} darauf basierend berechnet. Da das dynamische Verhalten auch vom SOC abhängt, sollte im Datenblatt eine Reihe von gepulsten Entladungen dargestellt sein. Somit könnten die Parameter für die RC-Netzwerke für verschiedene SOC ermittelt und darauf basierend eine Funktion der Klemmenspannung in Abhängigkeit des SOC definiert werden.

In der Arbeit von Petricca u. a. [154] wird exemplarisch eine *CGR18650CG* Batterie des Herstellers (*Panasonic*) mit einem Modell der Ebene 2.5 und eine Batterie (*Energizer CR2032*) mit einem Modell der Ebene 3 abgebildet und das simulierte Verhalten mit den Darstellungen im Datenblatt verglichen. Laut der Autoren stellt diese Arbeit einen ersten

4. Modellierung drahtloser Sensorknoten

Ansatz der automatisierten Modellgenerierung auf Basis der Datenblätter dar. Zudem soll ein Überblick gegeben werden, welche Informationen für welche Modellebene notwendig sind und somit eine Grundlage für standardisierte Herstellerangaben gelegt werden. Diese Notwendigkeit wird dadurch verdeutlicht, dass bei über 50 untersuchten Datenblättern nur in einem Datenblatt die notwendigen Informationen für ein Modell der Ebene 3 enthalten waren. Neben den von den Autoren selbst diskutierten Einschränkungen besteht eine weitere Herausforderung darin, dass die Hersteller oftmals nur grafische Darstellungen des Batterieverhaltens und keine tabellarischen Übersichten oder maschinenlesbare Datensätze zur Verfügung stellen. Dieser Umstand verhindert entsprechend eine automatisierte Modellgenerierung.

Die Weiterführung dieses Ansatzes wird durch Bocca u. a. [24] beschrieben. So wurde die Analyse der Datenblätter auf über 120 Datenblätter ausgeweitet (vgl. Tabelle 4.1), wobei die große Heterogenität in den Angaben bestätigt wurde. Da die realisierbare Modellstruktur von den verfügbaren Informationen in den Datenblättern abhängt, wird die Argumentation der notwendigen verschiedenen Modellebenen bestätigt. Wie in der

Tabelle 4.1.: In den Datenblättern enthaltene Daten für Lithium-Ionen Primär- und Sekundärzellen nach Bocca u. a. [24]

Parameter	Ausprägung	Anzahl	Primärzellen	Sekundärzellen
Leerlaufspannung vs. Laufzeit	Konstant	1	13 %	0 %
	Konstant	>1	79 %	8 %
Leerlaufspannung vs. SOC	Konstant	1	6 %	8 %
	Konstant	>1	13 %	89 %
	Gepulst	1	4 %	0 %
	Gepulst	>1	4 %	0 %
Innenwiderstand	Einzel	1	19 %	54 %
	Graph	>1	19 %	0 %

Arbeit von Petricca u. a. [154] werden hier nur Batterieeffekte erster Ordnung (*working condition*) betrachtet, wobei die Abhängigkeit der Kapazität vom Entladestrom als wichtigster Effekte betrachtet wird (*rated capacity effect*). Effekte zweiter Ordnung (*operating conditions*) werden nicht betrachtet. In dieser Arbeit werden die unten stehenden vier verschiedenen Modellebenen definiert. Es werden jeweils die benötigten Informationen aufgeführt und angegeben, in wie vielen Datenblättern der untersuchten Li-Ionen Primär- und Sekundärzellen diese Informationen zu finden waren. Neben Li-Ionen Zellen wurden auch die Datenblätter von Alkali-Mangan-Zellen und Bleiakkumulatoren untersucht.

- **Ebene 1:** Vereinfachtes Laufzeit-Modell basierend auf Benini u. a. [18] ohne Berücksichtigung des Innenwiderstandes (entspricht Ebene 2 bei Petricca u. a. [154]).

Es wird das Verhalten der Klemmenspannung in Abhängigkeit des SOC für einen konstanten Entladestrom benötigt.

- **Ebene 2:** Vereinfachtes Laufzeit-Modell basierend auf Benini u. a. [18] mit Berücksichtigung des Innenwiderstandes (entspricht Ebene 2,5 bei Petricca u. a. [154]). Es wird das Verhalten der Klemmenspannung in Abhängigkeit des SOC für mehrere konstante Entladeströme benötigt. Alternativ ist auch das Verhalten für einen Entladestrom und eine spezifische Angabe zum Innenwiderstand ausreichend, jedoch hängt dieser dann nicht vom SOC ab.
- **Ebene 3:** Kombiniertes Modell nach M. Chen und Rincon-Mora [38] (vgl. Abbildung 4.8) für einen SOC (entspricht Ebene 3 bei Petricca u. a. [154]). Neben den Angaben für Ebene 2 wird das Verhalten der Klemmenspannung über die Zeit für einen Entladepuls benötigt (vgl. Abbildung 4.5). Mit der Amplitude und dem zeitlichen Verlauf des Spannungsabfalls können die Parameter R_0 , R_{PA} , C_{PA} , R_{PC} und C_{PC} bestimmt werden.
- **Ebene 4:** Kombiniertes Modell nach M. Chen und Rincon-Mora [38] für mehrere SOC (entspricht Ebene 3 bei Petricca u. a. [154]). Neben den Angaben für Ebene 3 wird das Verhalten der Klemmenspannung über die Zeit für mehrere aufeinander folgende Entladepulse benötigt. Somit kann auf die Abhängigkeit zwischen den für Ebene 3 ermittelten Parameter und dem SOC geschlossen werden.

Es stehen somit vier verschiedene Ebenen der Modellbildung und eine Aufstellung darüber, welche Informationen für welche Ebene benötigt werden zur Verfügung. Der Ablauf der Modellgenerierung gliedert sich in drei Schritte:

- **Auswahl des Templates:** Anwendende Person prüft Angaben im Datenblatt und wählt geeignetes Template aus. Es stehen Templates für die zuvor beschriebenen vier verschiedenen Modellebenen zur Verfügung.
- **Extraktion der Daten:** Anwendende Person überträgt die Angaben aus dem Datenblatt in ein standardisiertes Format. In diesem Schritt müssen die Kurven im Datenblatt digitalisiert werden.
- **Modellidentifizierung:** Daten aus dem zweiten Schritt und Modell aus dem ersten Schritt werden verwendet, um mittels Skript die Parameter des Modells zu generieren. Es wird vollautomatisch ein simulierbares Modell generiert (MATLAB Skript).

Die ersten beiden Schritte müssen manuell erfolgen. Nur die Generierung des simulierbaren Modells im dritten Schritt erfolgt automatisch. Die Autoren bauen ein Modell der Ebene 2 für einen *Panasonic NCR18650E* Li-Ionen Akku und ein Modell der Ebene 3 für eine *Renata CR2032* Knopfzelle auf und vergleichen das simulierte Verhalten mit den Kurven

im Datenblatt. So werden die Modelle anhand des Verhaltens für die jeweils angegebene minimale und maximale Entladerate generiert, das Verhalten für eine dazwischen liegende Entladerate simuliert und mit dem Datenblatt verglichen. Als Fazit werden die manuelle Extraktion der Daten aus den Datenblättern und die fehlende Standardisierung der Herstellerangaben als Hindernisse einer automatischen Modellgenerierung identifiziert.

4.8 System

Ziel dieses Unterkapitels ist die Darstellung des Stands der Technik zur Modellierung der Energieaufnahme für das Gesamtsystem. Da jedoch keine der bekannten Arbeiten alle zuvor beschriebenen Komponenten berücksichtigt, werden in diesem Kapitel Ansätze beschrieben, die bei der Modellierung mindestens zwei verschiedene Komponenten betrachten.

So werden in der Arbeit von Simunic u. a. [188] neben der Modellierung des Mikrocontrollers (vgl. Abschnitt 4.1) auch der Einfluss des Spannungswandlers und der Batterie betrachtet. Die Modellierung des Spannungswandlers erfolgt mit dem in Abschnitt 4.6 beschriebenen Effizienzfaktor η . Das Verhalten der Batterie wird mit Hilfe der in Unterabschnitt 4.7.1 beschriebenen Peukert-Gleichung abgebildet. Das beschriebene Modell stellt einen guten Ansatz dar, jedoch muss für die zyklengenaue Betrachtung in jedem Takt bekannt sein, in welchem Zustand sich jede Komponente befindet und wie viel Leistung sie in diesem Zustand aufnimmt.

Konstantakos u. a. [110] messen und modellieren in ihrer Arbeit die Energieaufnahme eines eingebetteten Systems bestehend aus Mikrocontroller, analogem Temperatursensor, ADC und externem Speicher. Für den Mikrocontroller werden einzelne Instruktionen hinsichtlich der benötigten Takte und der Energieaufnahme pro Takt untersucht. Der ADC und der Speicher werden anhand der Energieaufnahme typischer Operationen (z.B. lesen und schreiben) charakterisiert. Abschließend wird die modellierte Energieaufnahme einer Beispielanwendung mittels Messung validiert, wobei eine Abweichung von unter 5 % erreicht wird.

Prayati u. a. [161] vermessen in ihrer Arbeit den verbreiteten *TelosB* Sensorknoten [4] und betrachten den *MSP430* Mikrocontroller (inaktive Modi und verschiedene Instruktionen), den internen Flash-Speicher (lesen, schreiben, löschen), den internen Timer, die RGB-LED und den Transceiver (verschiedene Sendeleistungen). Mikrocontroller, Flash-Speicher und Timer werden jeweils für zwei verschiedene Taktraten vermessen. Abschließend wird eine Formel für die Energieaufnahme des Gesamtsystems aufgestellt, wobei die Dauer der einzelnen Operationen jedoch vom Anwender eingegeben werden müssen.

In der Arbeit von Abdallah u. a. [3] werden Spannungswandler und Informationsverarbeitung parallel betrachtet. Ziel ist die Ermittlung des Minimum Energy Operation Point (MEOP) für das System, welcher als 3-er Tupel aus Energieaufnahme, Versorgungsspannung und Taktfrequenz ein Maß für die Energieeffizienz ist. Entscheidendes Ergebnis ist, dass der optimale Arbeitspunkt für den *Core* (Core Minimum Energy Operation Point (C-MEOP)) nicht dem optimalen Arbeitspunkt für das System (System Minimum Energy

Operation Point (S-MEOP)) entspricht. So hat die Effizienz des Spannungswandlers einen entscheidenden Einfluss und muss für eine energetische Optimierung mit betrachtet werden.

Die Arbeit von Martinez u. a. [124] zielt auf die Abschätzung der Leistungsaufnahme von IoT-Geräten auf Systemebene ab. Die Motivation ist, dass sich bisherige Arbeiten im Bereich der Abschätzung von drahtlosen Sensornetzen (*Wireless Sensor Network* (WSN)) nur auf die drahtlose Kommunikation beziehen und Ansätze auf Systemebene, wie die von Konstantakos u. a. [110], sich hingegen auf die Leistungsaufnahme des Prozessors beschränken. In dem von den Autoren beschriebenen Modell setzt sich die Leistungsaufnahme des Systems aus verschiedenen Blöcken zusammen und umfasst die Aspekte Daten aufnehmen, Daten verarbeiten, Kommunikation der Daten und die Leistungsaufnahme des zugrunde liegenden Systems (Betriebssystem, Transitionen zwischen verschiedenen Modi). Die durchschnittliche Leistungsaufnahme der Kommunikation basiert hier im Wesentlichen auf der Energie zur Übertragung der Daten und der Häufigkeit der Übertragung. Da der Fokus der Arbeit auf WSN liegt, werden zudem Faktoren wie Anzahl der Neuübertragungen, Sendeleistung und Spreizfaktor spezifischer betrachtet. Die Energieaufnahme zum Daten aufnehmen setzt sich aus der Energie für eine Messung und der Anzahl der Messungen zusammen. Im Gegensatz zu Systemen mit festen Intervallen wird die Datenerfassung in ereignisgetriebenen Anwendungen zudem von der Wahrscheinlichkeit des Ereignisses bestimmt. Die Energie für die Datenverarbeitung basiert auf der Art und Anzahl der ausgeführten arithmetischen Operationen. Abhängig von der Hardware werden für jede Operation eine bestimmte Anzahl Taktzyklen benötigt, die in Kombination mit der Energieaufnahme pro Takt die Energie der Datenverarbeitung bestimmen. Die Extraktion der Parameter erfolgt per Messung, wobei die verschiedenen Spannungsdomänen und die Effizienz des Spannungswandlers mit einbezogen werden. Um unabhängig von Spannungsdomänen zu sein, werden die jeweils benötigten Ladungen betrachtet. So beinhaltet die durchschnittliche Ladung für die Datenaufnahme die aktiven Anteile des Sensors und des Prozessors sowie die inaktive Stromaufnahme des Sensors. Für die Datenverarbeitung wird die Zeitkomplexität der Anwendung in Bachmann-Landau-Notation angegeben und die für jede Operation benötigte Ladung abgeschätzt. Die Einflüsse von Kommunikation, Datenaufnahme und Datenverarbeitung werden in einer Formel zusammen gefasst, wobei drei hardware-spezifische Konstanten (Ladung pro Sample, Ladung pro Instruktion, Ladung pro Kommunikation) und drei anwendungsspezifische Parameter (Anzahl der Samples, Datenmenge, Kommunikationsaktivität) definiert werden. Anschließend folgen zwei Anwendungsbeispiele des Modells. Die punktuelle Validierung zeigt eine gute Übereinstimmung, jedoch finden sich keine Angaben, wie die Stromaufnahme der realen Hardware für die Simulation ermittelt wurden.

Bag u. a. [15] stellen ein Modell vor, welches es ermöglichen soll, die Laufzeit eines drahtlosen Sensorknotens mit geringem Aufwand und ohne tieferes technisches Verständnis abzuschätzen. Jedoch erfolgt die Betrachtung sehr spezifisch für die Parameter des

Constrained Application (CoAP) Protokolls [186] und den *CC2560* SoC der Firma *Texas Instruments*. Zudem werden neben dem SoC keine weiteren Komponenten betrachtet und die Laufzeit des Systems lediglich anhand eines idealen Batteriemodells (vgl. Gleichung 4.4) abgeschätzt. Durch die spezifische Betrachtung, die notwendigen feingranularen Messungen und die unzureichende Betrachtung des System wird die Arbeit dem Ziel somit nicht gerecht.

Einen umfassenderen Ansatz für einen modellgestützten Entwurf stellen Tobola u. a. [210] vor. So besteht das System aus einer modularen Hardware für den Laborversuch und einem kompakten Sensormodul mit den identischen Komponenten. Auf dieser Basis konnten Parameter wie das Ein- und Ausschalten einzelner Komponenten sowie die Verwendung und Anpassung spezifischer Algorithmen zur Signalverarbeitung (ADC Auflösung, Samplingraten) vermessen und in einzelnen Modellen [214–216] abgebildet werden. Mit der Zusammenführung zu einem System [210] lassen sich alle Parameter komfortabel mittels einer *Graphical User Interface* (GUI) bedienen. Jedoch beschränkt sich die mögliche Auswahl der Komponenten auf die Batterie (10 Varianten), den Spannungswandler (5 Varianten) und das *Analog Front End* (AFE) (2 Varianten). Eine Betrachtung verschiedener Mikrocontroller, Sensoren und Transceiver ist nicht möglich.

Die Arbeit von Y. Chen u. a. [41] betrachtet das System ganzheitlich mit Mikrocontroller, verschiedenen Sensoren, Funktransceiver, Spannungswandler und Batterie. Motivation der Arbeit ist die signifikante Differenz zwischen der Leistungsaufnahme der Komponenten und der Leistungsentnahme aus der Batterie. Ziel ist ein optimiertes *Scheduling* von einzelnen Tasks. So versuchen bisherige Ansätze mitunter nur den Tastgrad zu minimieren, also alle Tasks in möglichst kurzer Zeit abzuarbeiten und somit eine lange inaktive Phase zu haben. Die Energieeffizienz hängt jedoch auch vom Arbeitspunkt der Spannungswandler und der Batterie ab, wobei vor allem große Entladeströme zu einem ineffizienten Betrieb der Batterie führen können. Die Autoren identifizieren die Tasks der Sensoren als größtes Optimierungspotential, da diese eine hohe Leistungsaufnahme aufweisen. Dieses bedingt sich jedoch zum einen durch die betrachteten stromintensiven Sensortypen sowie die veraltete Hardware. Die Sensor-Tasks haben jedoch den Vorteil, dass diese keine Abhängigkeiten aufweisen und somit beliebig parallel oder seriell angeordnet werden können. Für die Modellierung wird die Effizienz η des Spannungswandlers als Funktion aus Eingangsspannung, Ausgangsspannung und Strom definiert (vgl. Abschnitt 4.6). Als Batteriemodell wird das eigene elektrische Modell [39] verwendet, welches sowohl die Varianz als auch die Frequenz des Entladeprofils berücksichtigt (vgl. Absatz 4.7.2). Als Parameter der Komponenten werden für Sensoren die Leistungsaufnahme und die Dauer der Messung verwendet. Für den Mikrocontroller werden verschiedene aktive Modi eingetragen, die sich durch verschiedene Taktraten und davon abhängige Leistungsaufnahmen aufzeichnen. Die Taktrate bestimmt die benötigte Ausführungszeit. Freiheitsgrade in der Simulation sind die Anordnung der Sensor-Tasks und die aktiven Modi des Mikrocontrollers. Die optimale Konfiguration wird zunächst statisch ermittelt, wobei ein SOC der Batterie von 100 % angenommen wird. Eine optimale Konfiguration zeichnet sich durch

eine möglichst geringe Reduzierung des SOC aus. In der Simulation werden eine leichte Batterie mit geringem Nennstrom und eine schwerere Batterie mit höherem Nennstrom betrachtet. Für die statische Simulation betrug bei der Batterie mit hohem Nennstrom die Differenz zwischen längster und kürzester Laufzeit 4,53 %. Bei der Batterie mit geringem Nennstrom hingegen 18,48 %. Bei dynamischem *Scheduling*, also unter Einbeziehung des jeweils aktuellen SOC, konnte die Varianz der Lebensdauer zusätzlich um 3,27 % bzw. 6,37 % gesteigert werden. Für den dynamischen Ansatz, also der Anpassung der *Schedules* zur Laufzeit, sind die *Schedules* identisch für beide Batterien und werden während der Gesamtlaufzeit des System dreimal gewechselt. Für die Batterie mit geringerem Nennstrom wird der *Schedule* jedoch immer früher gewechselt. Das dynamische *Scheduling* konnte nur exemplarisch gezeigt werden, da laut der Autoren eine Berechnung zur Laufzeit zu aufwändig wäre. Die Arbeit von Chen u. a. [41] stellt den bisher umfassendsten Ansatz dar, jedoch wurden die simulierten Ergebnisse nicht validiert und die Varianz zwischen der kleinsten und größten Laufzeit ist als Kennzahl nur begrenzt aussagekräftig.

Bei den kommerziellen Werkzeugen erlaubt die Entwicklungsumgebung *CubeMX* [178] für die *STM32* Reihe des Herstellers *STMicroelectronics* eine Analyse der Leistungsaufnahme. Neben der grafischen Konfigurationen der Pin-Belegung, der benötigten Schnittstellen und darauf basierenden Generierung von Basis-Programmcode erlaubt der enthaltende *Power Consumption Generator* die Spezifikation der Spannungsversorgung (Versorgungsspannung, Batterie) und die Definition verschiedener Ausführungsschritte (*Steps*). Jeder Ausführungsschritt ermöglicht die Definition von spezifischem Modus, Taktfrequenz, Dauer, Peripherie und zusätzlichen Verbrauchern. Auf dieser Basis erfolgt die Bestimmung der Leistungsaufnahme für Prozessorkern und Peripherie in jedem Schritt und die Abschätzung der Batterielaufzeit. Jedoch müssen die Ausführungsschritte manuell hinzugefügt und konfiguriert werden, was einen zusätzlich Entwicklungsaufwand bedeutet. Die Batterie kann aus einer Datenbank ausgewählt werden, die für jede Batterie Angaben zur Kapazität (in mAh), Selbstentladung (in %/Monat), Nennspannung (in V) sowie maximalem kontinuierlichen und gepulsten Entladestrom (in mA) enthält. Weitere Batterien können mit diesen Angaben manuell hinzugefügt werden. Das zur Berechnung der Laufzeit verwendete Batteriemodell ist jedoch unklar. Falls die Leistungsaufnahme den spezifizierten maximalen Entladestrom der Batterie übersteigt, wird eine Warnung generiert. Weitere Komponenten, wie Sensoren, können als zusätzliche Stromaufnahme in einem Ausführungsschritt abgebildet werden.

4.9 Energiemanagement

Die Energieeffizienz, als Balance zwischen Leistungsaufnahme und Leistungsfähigkeit, eines drahtlosen Sensorsystems hängt, neben der Architektur und der Auswahl der Bauteile, auch vom Energiemanagement des Systems ab. Somit müssen die Methoden des Energiemanagements auch in der Modellierung eines System abgebildet werden können. Im Folgenden wird ein Überblick den gebräuchlichen Methoden gegeben. So steuert ein

dynamisches Energiemanagement (*Dynamic Power Management* (DPM)) die Zustände der einzelnen Komponenten in einem System [17]. Die Steuerung der einzelnen Komponenten kann dabei das vollständige Abschalten der Taktversorgung (*clock gating*) oder Spannungsversorgung (*power gating*), die dynamische Anpassung der Taktrate oder Versorgungsspannung (DVFS) sowie den Wechsel zwischen verschiedenen aktiven und inaktiven Systemzuständen umfassen. Die Komponenten können sowohl intern oder extern gesteuert werden. Die Leistungsaufnahme aller nicht steuerbaren Komponenten bestimmt die Grundlinie der Leistungsaufnahme des Systems. Zur Optimierung der Energieeffizienz eines System muss eine die Konfiguration der möglichen Zustände aller Komponenten optimiert werden. Für einen optimierten Wechsel zwischen einem aktiven und einem inaktiven Zustand einer Komponenten wird von Benini u. a. [17] die Kennzahl der Nutzenzeit (*break-even time*) T_{BE} formuliert. Diese gibt an, wie lange eine inaktive Phase der Komponente andauern muss, um aus energetischer Sicht den Wechsel in den inaktiven Zustand zu amortisieren. Neben der Leistungsaufnahme im inaktiven Zustand wird die Energieaufnahme der Transitionen zwischen den beiden Zuständen einbezogen. Die T_{BE} muss somit für jeden möglichen inaktiven Zustand einer Komponente berechnet werden. Die Einsparung der Leistungsaufnahme ergibt sich aus der Differenz zur Leistungsaufnahme im dauerhaft aktiven Zustand. Eine Modellierung muss somit den Betrieb der Komponenten in verschiedenen Zuständen (z.B. aktiv oder inaktiv) sowie die Variation von Taktrate und Versorgungsspannung abbilden können.

4.10 Zusammenfassung und Bewertung der Methode

Zusammenfassend weist der Stand der Technik in Bezug auf die Modellierung der einzelnen Komponenten Unterschiede hinsichtlich Komplexität und Anwendbarkeit auf. So kann das Verhalten von Sensoren, Transceiver, Speicher und Schnittstellen bei entsprechender Dokumentation durch den Hersteller (Energieaufnahme der jeweiligen Aktion) gut abgebildet werden. Der Einfluss des Spannungswandlers kann sowohl für Schaltregler als auch für Linearregler über die Effizienz η modelliert werden. Gegenwärtige Batteriemodelle sind jedoch technisch anspruchsvoll und die auf dem Mikrocontroller auszuführende Software eröffnet einen großen Entwurfsraum. Die Modelle für Mikrocontroller, Energiespeicher und das Gesamtsystem sind demnach noch Gegenstand aktueller Forschung. Die sich hier stellenden Herausforderungen werden im Folgenden zusammen gefasst.

Mikrocontroller Die beschriebene Abbildung der Energieaufnahme für einen spezifischen Algorithmus über die Art und Anzahl der benötigten Instruktionen stellt einen vielversprechenden Ansatz dar. Entscheidende Faktoren sind hier valide Daten für die Basiskosten jeder Instruktion (Leistungsaufnahme und Anzahl der benötigten Takte). Zudem muss gegebenenfalls die Varianz der Basiskosten aufgrund von Abhängigkeiten (Systemzustand, Operanden, Speicher) und bedingten Anweisungen im Programmablauf bestimmt werden, welche unter Umständen erst zur Laufzeit vorliegen. Die größte

Herausforderung für die Abschätzung der durch die Software bedingten Leistungsaufnahme eines Systems besteht in der generischen Anwendbarkeit eines Modells, welches die Abhängigkeit zwischen Algorithmus und Zielplattform systematisch beschreibt.

Energiespeicher Neben der Wahl des für die jeweilige Anwendung geeigneten Modells und dem damit verbundenen Kompromiss zwischen Genauigkeit und Anwendbarkeit, stellt die Generierung der benötigten Parameter eine große Herausforderung dar. So sind die für den Aufbau von komplexen und somit potentiell präzisen Modellen benötigten Daten oftmals nicht in den Datenblättern der Batteriehersteller vorhanden und müssen aufwändig experimentell ermittelt werden. Die beschriebenen Ansätze zur Generierung sind derzeit nur für einfachere Modell geeignet und durch die manuelle Extraktion der Daten zudem zeitaufwändig und fehleranfällig.

System Die in Tabelle 4.2 dargestellte Übersicht derzeitiger Systemmodelle verdeutlicht, dass bisher kein umfassendes und gleichzeitig praktikables Systemmodell verfügbar ist.

Tabelle 4.2.: Übersichtstabelle der Systemmodelle. Mit X markierte Komponenten werden von dem Systemmodell abgedeckt. Mit ~ markierte Komponenten werden nur rudimentär betrachtet. Für Spannungswandler und Energiespeicher wird das jeweils verwendete Modell angegeben.

Arbeit	Jahr	Core	Sens.	Akt.	Schnittst.	Spannungsr.	Energiesp.
Simunic u. a. [188]	2001	X	-	-	-	η	Peukert
Konstantakos u. a. [110]	2008	X	X	X	-	-	-
Prayati u. a. [161]	2010	X	-	X	-	-	-
Abdallah u. a. [3]	2011	X	-	-	-	η	-
Martinez u. a. [124]	2015	X	X	X	-	η	-
Bag u. a. [15]	2016	X	-	-	-	-	Ideal
Tobola u. a. [210]	2016	X	X	-	-	η	Ideal
Chen u. a. [41]	2018	X	X	X	-	η	El. Mod. [39]
STM32CubeMX [178]	2018	X	~	~	~	-	n.b.

So ist das Modell von Konstantakos u. a. [110] beispielsweise nur sehr spezifisch für die in der Arbeit beschriebene Anwendung anwendbar. Andere Arbeiten wie die Modelle von Simunic u. a. [188], Prayati u. a. [161], Abdallah u. a. [3] und Bag u. a. [15] betrachten zwar mehrere Komponenten eines drahtlosen Sensorknotens, jedoch werden jeweils wichtige Komponenten wie die Sensoren oder die Spannungsversorgung nicht mit einbezogen. Die Arbeit von Tobola u. a. [210] ist bereits sehr umfassend, jedoch nicht generisch anwendbar.

So ist es nicht möglich, verschiedene Mikrocontroller zu betrachten, die Algorithmen sind vorgegeben und der Einfluss des Transceivers wird nicht abgebildet. Das Modell von Y. Chen u. a. [41] erfüllt durch die umfassende Modellierung der Komponenten in Kombination mit aktuellen Modellen für den Spannungswandler und die Batterie bereits viele Anforderungen. Jedoch fokussiert sich die Arbeit auf das *Scheduling* einzelner Tasks und es wird keine Entwurfsraumexploration für die Auswahl der Komponenten ermöglicht. Es wird lediglich der Einfluss von zwei verschiedenen Batterien betrachtet, wobei alle übrigen Komponenten nicht variiert werden können. Zudem wird der Einfluss der Transitionen zwischen verschiedenen Zuständen und der Datenfluss zwischen den Komponenten nicht betrachtet.

Im Bereich kommerzieller Werkzeuge erlaubt *CubeMX* [178] die Betrachtung der Leistungsaufnahme für verschiedene Konfigurationen, jedoch müssen die Ausführungsschritte manuell eingetragen werden. Der Bezug zu spezifischen Algorithmen ist somit nicht gegeben. Zudem können zusätzliche Komponenten, wie Sensoren und Transceiver, nur sehr abstrakt über eine zusätzliche Leistungsaufnahme in einem Ausführungsschritt eingebunden werden. Durch die Auswahl einer Batterie kann die Laufzeit des Systems abgeschätzt werden, jedoch ist das zugrunde liegende Modell nicht bekannt.

5 Modellgestützte Entwurfsmethodik

In diesem Kapitel wird die entwickelte Methodik für den Entwurf von drahtlosen Sensorknoten beschrieben. Einleitend erfolgt in Abschnitt 5.1 zunächst die Einordnung dieses modellbasierten Vorgehens in den Entwurfsprozess drahtloser Sensorknoten. Im Anschluss wird in Abschnitt 5.2, basierend auf der zuvor beschriebenen derzeitigen Vermessung und Modellierung drahtloser Sensorknoten, der Handlungsbedarf abgeleitet. Darauf folgt die theoretische Betrachtung des aufgestellten Modells. Es werden zunächst das grundlegende Systemmodell und das grundlegende Komponentenmodell beschrieben. Anschließend werden die spezifischen Modelle für die einzelnen Komponenten (Mikrocontroller, Sensoren, Transceiver, Speicher, Schnittstellen, Spannungswandler, Energiespeicher) und deren Zusammenschluss zu einem Modell für das Gesamtsystem beschrieben (Systemkonfiguration). Darauf aufbauend wird das Systemverhalten mittels der Definition von Tasks, dem Tastgrad und dem Datenfluss abgebildet. Einzelne Aspekte des Modells und eine erste Auswertung konnten bereits publiziert werden [250].

Das gesamte Modell wurde in MATLAB [122] implementiert. Somit liegen sämtliche Funktionen als Skripte vor (*m-Files*).

5.1 Einordnung in den Entwurfsprozess

Der in dieser Arbeit angestrebte modellbasierte Entwurf von drahtlosen Sensorknoten ist als Teil des notwendigen Entwurfsprozesses für mikroelektronische Systeme zu verstehen. Zur Einordnung des eigenen Beitrags soll in diesem Kapitel der Stand der Technik bezüglich des Entwurfsprozesses beschrieben werden. Basis bildet die Dissertation von Herbrechtsmeier [78], in der er das Modell eines agilen Leiterplattenentwurfsprozesses beschreibt.

Als grundlegendes Entwurfsmodell für mikroelektronische Systeme beschreibt Herbrechtsmeier [78, S. 174], mit Verweis auf Gajski und Kuhn [70], das in Abbildung 5.1 dargestellte Y-Diagramm. Dieses beschreibt auf den Achsen verschiedene Sichten auf das System anhand der drei Repräsentationen Struktur, Verhalten und Gestalt. Die Kreise beschreiben den jeweiligen Abstraktionsgrad, wobei dieser in Richtung des Mittelpunktes immer feingranularer wird und somit eine detailliertere Spezifikation des Systems darstellt. Da der angestrebte modellbasierte Entwurf die Entwicklung drahtloser Sensorknoten fokussiert, wird dieser als Funktionseinheit betrachtet, die laut Herbrechtsmeier [78, S. 236] eine „nach Aufgabe oder Wirkung abgrenzbare Betrachtungseinheit“ beschreibt. Die möglichen Bauelemente dieser Funktionseinheit sind laut Herbrechtsmeier [78, S. 235] die „kleinsten, nicht weiter zerlegbaren logischen Einheiten einer Schaltung“. Diese können daher als die in Abschnitt 2.2 beschriebenen Komponenten angesehen werden. Zusätzlich müssten lediglich passive Komponenten wie Widerstände und Kondensatoren betrachtet werden. Der angestrebte modellbasierte Entwurf unterstützt somit die BauteilAuswahl, also die Detaillierung der für die Funktionseinheit des Sensorknotens benötigten Bauelemente (z.B. die Komponente Mikrocontroller) hin zu spezifischen Bauteilen (Hersteller

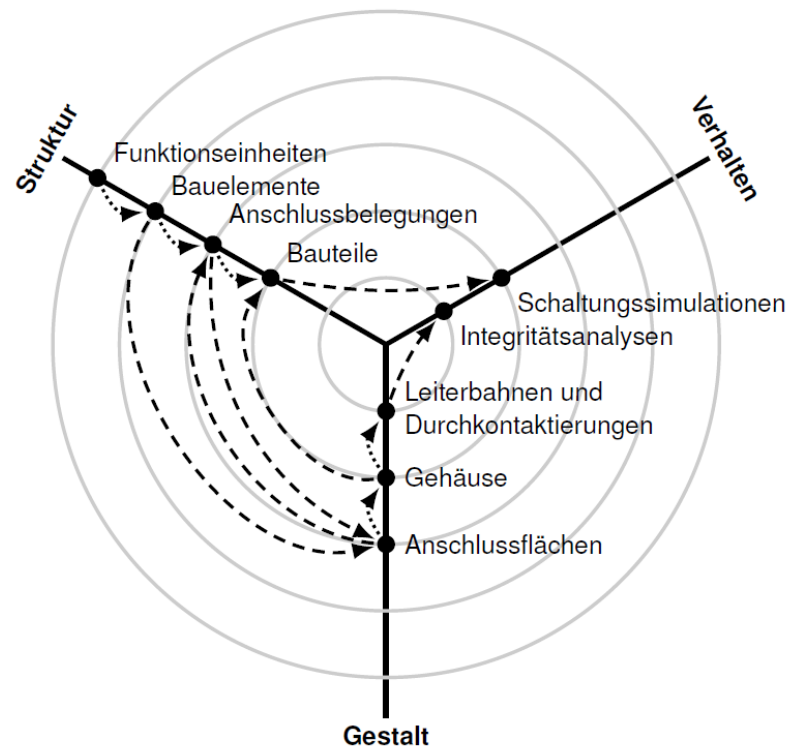


Abbildung 5.1.: Entwurfsmodell eines agilen Leiterplattenentwurfsprozesses nach Herbrechtsmeier [78]

und Typ). In einem klassischen Entwurfsablauf werden diese in einer Bauteilrecherche ermittelt, in dem verschiedene Bauteile anhand ihrer Datenblätter verglichen und relevante Eigenschaften in einer Tabelle gesammelt werden [78, S. 162]. Dieser Prozess ist jedoch nicht spezifiziert [78, S. 165]. Daraus folgt, dass die Auswahl der Bauteile oftmals nicht dokumentiert und somit der Entscheidungsprozess auch nicht nachvollziehbar ist [78, S. 164]. In dem von Herbrechtsmeier [78] beschriebenen agilen Entwurfsprozess wird dieser daher näher spezifiziert. Der beschriebene Prozessmodell für den gesamten agilen Leiterplattenentwurfsprozess ist in Abbildung 5.2 dargestellt. So werden im Architekturentwurf die Funktionen des Systems auf einzelne Funktionseinheiten abgebildet, welche wiederum aus einzelnen Bauelementen bestehen [78, S. 183]. Nach den für den Entwurf der Leiterplatte relevanten Schritten des Schaltungsentwurfs, der Anschlussflächenauswahl, der Anschlussbelegungsspezifikation und der Platzierung erfolgt die Gehäuseauswahl. Diese weist den Bauelementen spezifische Gehäuseformen zu [78, S. 188]. Anschließend werden durch den Schritt der Bauteilerauswahl (vgl. Abbildung 5.3) die Bauelemente mittels detaillierterer Spezifikation durch Bauteile ersetzt. Die Auswahl erfolgt durch den Vergleich der

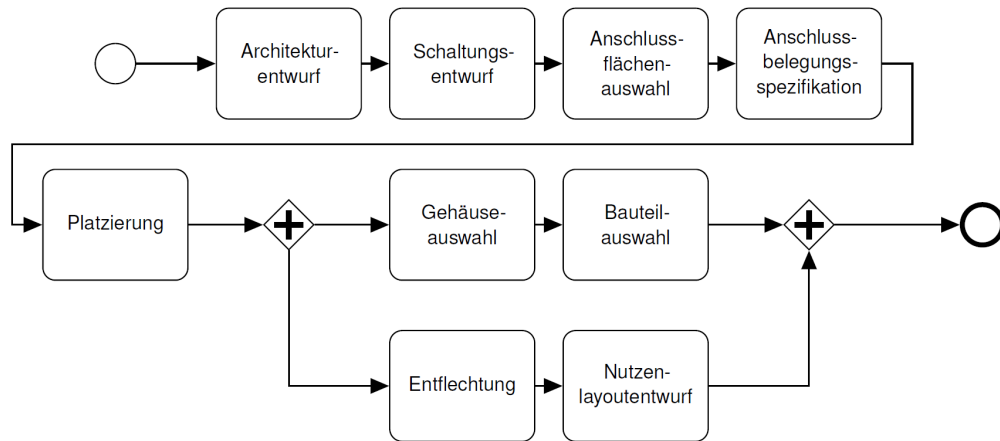


Abbildung 5.2.: Prozessmodell eines agilen Leiterplattenentwurfsprozesses nach Herbrechtsmeier [78]

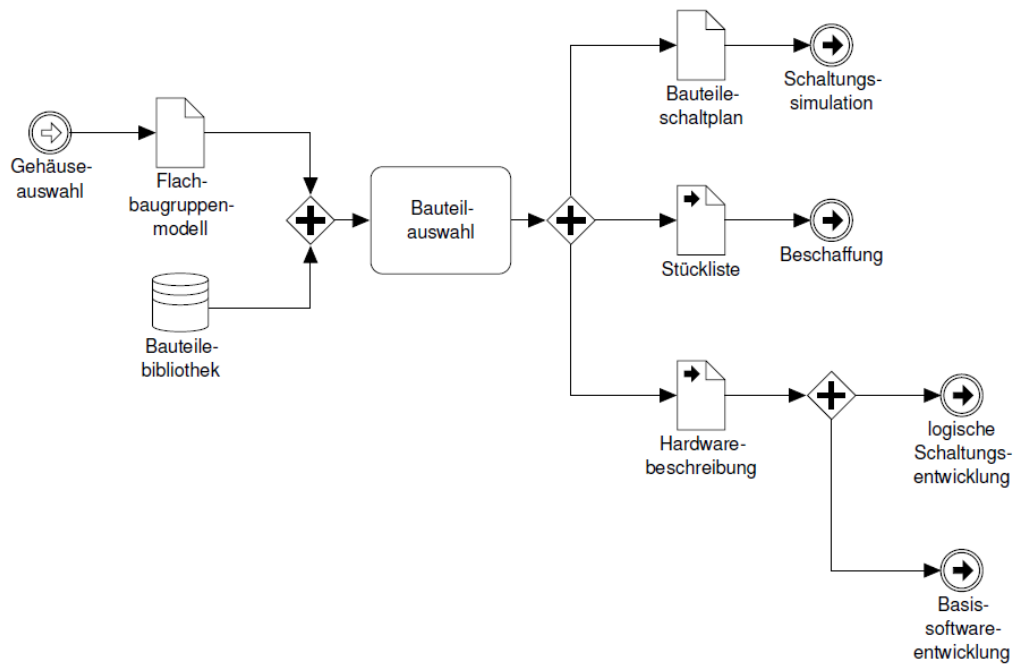


Abbildung 5.3.: Bauteilauswahl in einem agilen Leiterplattenentwurfsprozess nach Herbrechtsmeier [78]

geforderten Eigenschaften der benötigten Bauelemente mit den Eigenschaften der in einer Bauteilbibliothek enthaltenden Bauteile [78, S. 189]. Bezüglich der Eigenschaften der Bauteile in der Bauteilbibliothek unterscheidet Herbrechtsmeier [78] zwischen physikalischen (z.B. elektrotechnische Kennzahlen), physischen (z.B. Abmessungen) und wirtschaftlichen (z.B. Kosten) Eigenschaften. Bei den physikalischen Eigenschaften wird zudem zwischen unveränderlichen, konfigurierbaren und resultierenden Eigenschaften unterschieden [78, S. 203]. Unveränderliche Eigenschaften werden vom Bauteil vorgegeben, konfigurierbare Eigenschaften stellen mehrere Optionen zur Verfügung, die bei Auswahl zu resultierenden Eigenschaften führen [78, S. 204]. Somit wird eine Entwurfsraumexploration ermöglicht, die laut Herbrechtsmeier [78, S. 29] „eine Methode zur Sondierung alternativer Entwürfe und zur Ermittlung einer optimalen und ressourceneffizienten Lösung“ darstellt. Dieses umfasst eine Mehrzieloptimierung mit Entscheidungsfindung sowie die Gegenüberstellung verschiedener Bewertungsfunktionen.

Übliche Zielgrößen für die Bewertungsfunktionen der Entwurfsraumexploration sind beispielsweise die *Worst-Case Energy Consumption* (WCEC), die als maximale Energieaufnahme des Systems in allen möglichen Zuständen definiert ist und die *Worst-Case Power Consumption* (WCPC) als maximale Leistungsaufnahme des Systems [125]. Die WCEC ist dabei als Integral der WCPC über die WCET definiert [125], also die maximale Ausführungszeit:

$$\text{WCEC} \leq \int_0^{\text{WCET}} \text{WCPC} \, dt = \text{WCET} \cdot \text{WCPC} \quad (5.1)$$

Ansätze zur Abschätzung der WCEC von Software finden sich beispielsweise bei Jayaseelan u. a. [98], Pallister u. a. [149], Wägemann u. a. [222] und Trilla u. a. [219].

Analog zur WCEC wird für die Hardware häufig das *Power-delay Product* (PDP) oder auch das *Time-power Product* (TPP) betrachtet, also das Produkt aus Ausführungszeit und Leistungsaufnahme. In der Arbeit von Hübener [85] wird der Kehrwert des PDP als Energieeffizienz definiert. Diese Definition greift hier jedoch zu kurz, da die Energieeffizienz den Leistungsbedarf in Bezug zur Leistungsfähigkeit setzt. Für drahtlose Sensorknoten ergibt sich die Leistungsfähigkeit des Systems aus der Quantität und Qualität der Messwerte sowie dessen Datenverarbeitung. Der Begriff der Ressourceneffizienz ist im Vergleich zur Energieeffizienz weiter gefasst und schließt weitere Zielgrößen wie die Kosten ein.

Das modellbasierte Vorgehen bei der Systementwicklung (*Model-Based Systems Engineering* (MBSE)) [60] kann die Basis für einen *Digitalen Zwilling* des Systems bilden, welcher u.a. von Boschert und Rosen [32] beschrieben wird. Dieser beinhaltet umfassende physikalische und funktionale Beschreibungen des Systems, die in allen Phasen wie dem Entwurf (*Design*), der Entwicklung (*Engineering*), dem Betrieb (*Operation*) und der Wartung (*Service*) Mehrwerte generieren können. Wesentlicher Aspekt des *Digitalen Zwillings* ist, neben der Abbildung von Informationen und Abhängigkeiten in einem Modell (beispielsweise durch MBSE), die Ausführbarkeit der Modelle, so dass die Phasen des Systems in der jeweils benötigten Granularität durch Simulationen gestützt werden können. Dieses erfordert einen

modularen Aufbau mit standardisierten Schnittstellen. Zusätzlich müssen Algorithmen vorgesehen werden, die sowohl historische Daten als auch Echtzeitdaten einbeziehen und somit den *Digitalen Zwilling* verifizieren oder aktualisieren.

5.2 Handlungsbedarf

Nachdem der Stand der Technik drahtloser Sensorknoten hinsichtlich Komponenten, Messgrößen und Parameter sowie verfügbarer Systeme dargestellt (vgl. Kapitel 2) und die derzeitigen Ansätze hinsichtlich Vermessung (vgl. Kapitel 3) und Modellierung (vgl. Kapitel 4) von Sensorknoten beschrieben wurden, erfolgt hier auf dieser Basis die Ableitung des Handlungsbedarfs. Neben dem grundlegenden Bedarf für einen modellbasierten Entwurf drahtloser Sensorknoten werden die Anforderungen hinsichtlich des abzubildenden Systems und der Skalierbarkeit des Modells beschrieben.

Der grundlegende Bedarf leitet sich aus der stetig steigenden Komplexität der technischen Systeme ab. Die Vielzahl an verschiedenen Komponenten eines Systems und die verschiedenen möglichen Zustände jeder einzelnen Komponente führen dazu, dass die ganzheitliche Vermessung eines Sensorknotens nicht mehr, oder nur sehr zeitaufwändig möglich ist. So decken derzeitige Ansätze in der Literatur oftmals nur Spezialfälle ab, die exemplarisch vermessen werden. Somit ist für jedes zu entwickelnde System oder dessen Anwendung eine Einzelfallbetrachtung notwendig. Zudem ist eine Messung erst nach der Fertigung möglich, so dass gegebenenfalls mehrere zeit- und kostenaufwändige Iterationen nötig sind, bis ein energieeffizientes System zur Verfügung steht. Die Vermessung aktueller drahtloser Sensorknoten ist somit nicht mehr praktikabel. Jedoch ist noch kein generisches und ganzheitliches Modell für den Entwurf drahtloser Sensorknoten vorhanden. Bisherige Arbeiten befassen sich auch hier nur mit einzelnen Aspekten eines Systems. Zudem erfolgt kein Abgleich zwischen der Leistungsaufnahme und der Leistungsfähigkeit eines Systems, so dass die Energieeffizienz nicht bestimmt wird. Im Folgenden wird daher ein Überblick über die Anforderungen an das Modell gegeben.

5.2.1 Abzubildendes System

Ein generisches, ganzheitliches und gleichzeitig praktikables Modell zur Abschätzung der Energieaufnahme drahtloser Sensorknoten muss verschiedene Aspekte erfüllen, die hier anhand verschiedener Kategorien dargestellt werden.

Allgemeines Komponentenmodell Ein möglichst generisches Komponentenmodell sollte in der Lage sein, die für jede Komponente relevanten Parameter, wie die Leistungsaufnahme in verschiedenen Zuständen, abzubilden. Zudem sollte beachtet werden, dass auch die Zustandsübergänge energetische Kosten verursachen und somit einen Einfluss auf die Energieeffizienz haben [17]. Bei häufigen Zustandsübergängen können die Transitionskosten die Einsparung von Energiespartechniken überwiegen [25].

Komponente Mikrocontroller Die Modellierung eines Mikrocontrollers sollte, neben der Leistungsaufnahme der verschiedenen Betriebsmodi, auch einen Zusammenhang zwi-

schen dem auszuführenden Algorithmus und der für die Ausführung benötigten Energie herstellen. Die bestehenden Ansätze zur Erweiterung des Compilers um die Zielgröße der Energieaufnahme (*energy aware compiler* [165]) sind häufig nicht im Entwurfsablauf integriert und erfordern die zeitintensive Implementierung des Algorithmus für die jeweilige Zielplattform. Für eine Bewertung der energetisch optimalen Systemkonfiguration für verschiedene Algorithmen (*software-related power*) muss ein praktikabler Kompromiss zwischen Genauigkeit und Komplexität gefunden werden, so dass eine Evaluierung spezifischer Algorithmen auf verschiedenen *virtuellen* Systemen ermöglicht wird.

Komponente Transceiver Neben der Leistungsaufnahme in den verschiedenen Betriebsmodi sollte die Modellierung eines Funktransceivers auch die beeinflussenden Parameter wie die Breite der Nutzdaten, die Sendestärke und die Sensitivität des Empfangsteils enthalten.

Komponente Sensoren Die Modellierung eines Sensors sollte, neben der Leistungsaufnahme in den möglichen Modi, auch die benötigte Dauer für die spezifischen Messungen enthalten. Ziel sollte eine einheitliche Darstellung sein, so dass eine Vergleichbarkeit zwischen verschiedenen Sensoren gegeben ist. Somit wäre auch eine Bewertung der Sensoren anhand der Energieeffizienz (Präzision pro Energie) möglich.

Komponente Schnittstellen Die Modellierung von Schnittstellen muss, neben der Leistungsaufnahme der aktiven (Senden und Empfangen) und inaktiven Modi, auch die Taktfrequenz der Schnittstelle und den Protokolloverhead enthalten.

Spannungsversorgung Da die Komponenten der Spannungsversorgung potentiell einen entscheidenden Einfluss auf die Leistungsaufnahme des Gesamtsystems haben, müssen diese mit berücksichtigt werden. Relevant sind hier die Effizienz des Spannungswandlers und die verschiedenen Batterieeffekte.

Datenfluss Um das Systemverhalten abbilden zu können, muss das Modell, neben den direkten energetischen Aspekten wie den Betriebsmodi, auch funktionale Aspekte wie den Datenfluss zwischen den einzelnen Komponenten abbilden. Diese bestimmen maßgeblich die Verarbeitungszeiten und somit die Zeit in den aktiven Modi.

Gesamtsystem Entscheidend für die Abbildung des Gesamtsystems ist die ganzheitliche Modellierung des Systems anhand der enthaltenden Komponenten, so dass Wechselwirkungen untersucht werden können. So muss beispielsweise für eine qualifizierte Analyse der Laufzeit eines Systems auch die Effizienz des Spannungswandlers [3] und der verwendete Energiespeicher [188] betrachtet werden. Zudem müssen gängige Energiespartechniken und der Datenfluss abgebildet werden können.

5.2.2 Skalierbarkeit des Modells

Für eine praktische Verwendung eines modellbasierten Entwurfs muss der Ansatz skalierbar sein. Das heißt, dass das Modell in der Lage sein muss, eine Vielzahl möglicher

Systeme und auch Variationen innerhalb des Systems effizient abzubilden. Im Folgenden werden daher Aspekte zur Skalierbarkeit des Modells beschrieben.

Modularität Das gesamte Modell muss durchgehend modular gestaltet werden, so dass zum einen die Komplexität des Gesamtsystem beherrschbar wird und zum anderen einzelne Module wieder verwendet werden können. Durch einen modularen Ansatz könnten zudem abstrahierte Schnittstellen zwischen einzelnen Modulen geschaffen werden, die im Entwurfsablauf verschiedene Abstraktionsebenen ermöglichen.

Generierung der Daten Großer Handlungsbedarf besteht bei den für die Modellierung benötigten Daten. So ist vor allem die Generierung der Daten für die auszuführenden Algorithmen auf spezifischen Mikrocontrollern und für die Energiespeicher komplex. Zwar bestehen bereits Ansätze diese aus den Datenblättern zu entnehmen, jedoch weist die Dokumentation durch die Hersteller eine sehr heterogene Ausprägung auf, so dass ein generischer Ansatz in vielen Bereichen noch nicht, oder nur für sehr einfache Modelle möglich ist.

Datenbasis Durch die zahlreichen Arbeiten im Bereich der Vermessung und Modellierung besteht bereits eine stetig wachsende Informationsbasis. Diese ist jedoch bisher nicht überschaubar und muss im Zuge einer Literaturrecherche erarbeitet werden, was bei einer oft zeitkritischen Entwicklung neuer System nicht praktikabel ist. Die Informationsbasis der aktuellen Arbeiten sollte für die Gemeinschaft nutzbar werden und eine Plattform mit einheitlichem Format zur Sammlung der Informationen geschaffen werden. Da viele Arbeiten nur punktuell spezifische Aspekte einzelner Hardwarekomponenten oder Algorithmen untersuchen, muss die Information zum einen abstrahiert und zum anderen modular aufgebaut werden. Ziel in diesem Bereich sollte eine standardisierte Informationsbasis für den Entwurf drahtloser Sensorknoten sein, die beständig erweitert werden kann. Möglichkeiten zur Erweiterung könnten, neben der manuellen Eingabe der Daten, das automatisierte Einlesen (*parsen*) von Datenblättern oder vom Hersteller zur Verfügung gestellter maschinenlesbarer Daten sein. Somit könnte eine stetig wachsende Bauteilbibliothek entstehen, die die Effizienz des Bauteilwahlprozesses deutlich steigern könnte [78, S. 221].

Simulation Zur direkten Nutzbarkeit der Systembeschreibung und der damit verbundenen Praxisrelevanz muss der Ansatz über die reine Darstellung der Informationen hinausgehen. Ein wichtiger Aspekt des Modells ist daher die Simulation, so dass das Modell nicht nur der Darstellung bzw. der Übersicht dient, sondern funktionale Aspekte, die für den Entwurfsablauf relevant sind, auch direkt simuliert werden können. Die anwendende Person könnte somit das Modell parametrieren und das so beschriebene System für einen breiten Entwurfsraum simulieren. Es sollte eine Entwurfsraumexploration für drahtlose Sensorknoten geschaffen werden, welche die primären Ziele Energieaufnahme und Leistungsfähigkeit als Basis für die Energieeffizienz abdeckt. Sekundäre Größen wie

Bauform, Kosten könnten gegebenenfalls als eine weitere Ebene im Entwurfsablauf verstanden werden. Zusammenfassend sollte sich der für die Modellierung eines Systems benötigte Aufwand durch die Simulation und die sich daraus ergebenden Optimierungen amortisieren.

Auf Basis des identifizierten Handlungsbedarfs für den modellbasierten Entwurf drahtloser Sensorknoten wird im Folgenden das entwickelte Modell beschrieben.

5.3 Grundlegendes Systemmodell

Das Gesamtsystem eines drahtlosen Sensorknotens besteht aus einzelnen aktiven Komponenten wie dem Mikrocontroller, dem Funktransceiver, einem Speicher und einem oder mehreren Sensoren. Zusätzlich wird die Spannungsversorgung, bestehend aus Energiespeicher und Spannungswandler, berücksichtigt. In Abbildung 5.4 ist das System als Blockdefinitionsdiagramm dargestellt, wobei die einzelnen Blöcke die Komponenten des Systems repräsentieren. Über die Multiplizität wird die zulässige Anzahl der jeweiligen Komponente definiert. So besteht ein drahtloser Sensorknoten aus jeweils genau einem Mikrocontroller und Funktransceiver, mindestens jeweils einem Sensor, Spannungswandler und Energiespeicher ([1..*]), sowie optionalem Speicher ([0..*]). Die formale Darstellung der Komponenten, Eigenschaften und Parameter des Systems folgt der Syntax- und Semantik der Modellierungssprache *Systems Modeling Language* (SysML) [69; 224], die auch bereits von Herbrechtsmeier [78] für den agilen Leiterplattenentwurfprozess verwendet wurde.

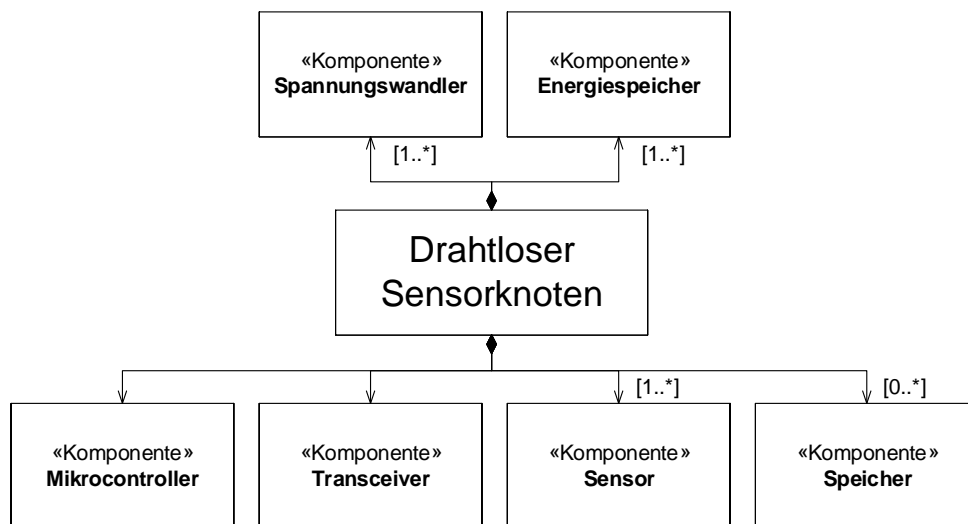


Abbildung 5.4.: Formale Darstellung der Komponenten eines drahtlosen Sensorknotens

Die Modellbildung erfolgt zunächst für jede einzelne Komponente des Sensorknotens, wobei dieser verschiedene Parameter (Zustände, Kosten, ...) zugeordnet werden. Auf Basis der Komponenten und deren abstrahierten Schnittstellen erfolgt anschließend, zur Abbildung des Systemverhaltens, die Modellierung des Gesamtsystems. Dabei muss das Modell verschiedene Detaillierungsgrade unterstützen, so dass eine schnelle Generierung grundlegender Modelle ermöglicht wird, die zu einem späteren Zeitpunkt mit komplexeren Modellen und Daten weiter detailliert werden können. Im Folgenden wird zunächst das grundlegende Komponentenmodell beschrieben.

5.4 Grundlegendes Komponentenmodell

Eine generische Beschreibung zur Modellierung sowohl auf Komponenten-, als auch auf Systemebene stellen Benini u. a. [17] vor. So besteht ein System aus steuerbaren Komponenten (*Power Manageable Component* (PMC)), wobei die Zustandsmodellierung einzelner Komponenten als Endlicher Automat (*Power State Machine* (PSM)) abgebildet wird (vgl. Abbildung 5.5). Die Ein- beziehungsweise Abschaltung und Anpassung einzelner Komponenten ist entscheidend für ein energieeffizientes Design. Zudem ist zu beachten, dass auch die Zustandsübergänge energetische Kosten verursachen und somit abgebildet werden müssen [17]. Der Betrieb einer Komponente im Zustand X verursacht die Leistungsaufnahme P_X . Die Transition vom Zustand X in den Zustand Y verursacht die Energieaufnahme $E_{X,Y}$. Die Dauer des Betriebs in einem spezifischen Zustands hängt vom jeweiligen Typ der Komponente und der Anwendung ab.

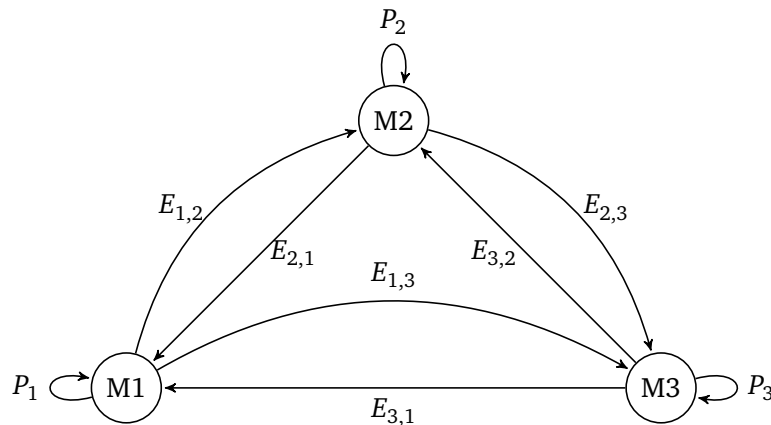


Abbildung 5.5.: Abbildung der Zustände und Zustandsübergänge einer Komponente anhand einer PSM (angelehnt an [17])

Im Komponentenmodell müssen daher die Leistungsaufnahmen der verschiedenen Modi N_M und die Energieaufnahme für die Transitionen zwischen diesen Modi hinterlegt

sein. Eine Übersicht der Parameter des grundlegenden Komponentenmodells wird durch Tabelle 5.1 gegeben.

Tabelle 5.1.: Parameter des grundlegenden Komponentenmodells

Beschreibung	Parameter	Einheit	Format	Definition
Bezeichnung	<i>description</i>	-	String	Manuell
Min. Versorgungsspannung	<i>voltage.min</i>	V	[1x1] double	Manuell
Typ. Versorgungsspannung	<i>voltage.typ</i>	V	[1x1] double	Manuell
Max. Versorgungsspannung	<i>voltage.max</i>	V	[1x1] double	Manuell
Modi	<i>mode</i>	-	[1x N_M] struct	Manuell
- Bezeichnung	<i>mode.description</i>	-	String	Manuell
- Aktiver Modus	<i>mode.active</i>	-	[1x1] double	Manuell
- Stromaufnahme	<i>mode.current</i>	mA	[1x1] double	Manuell
Transitionen	<i>transitions</i>	-	[1x1] struct	Manuell
- Dauer	<i>transitions.time</i>	μ s	[$N_M \times N_M$] double	Manuell
- Stromaufnahme	<i>transitions.current</i>	mA	[$N_M \times N_M$] double	Manuell
Schnittstellen	<i>interface</i>	-	[1x N_{IF}] struct	Manuell
- Bezeichnung	<i>interface.description</i>	-	String	Manuell
- Nummer	<i>interface.no</i>	-	[1x1] double	Manuell
- Datenrate	<i>interface.datarate</i>	kbps	[1x1] double	Manuell
- Stromaufnahme	<i>interface.current</i>	mA	[1x1] double	Manuell
Fläche des Gehäuses	<i>area</i>	mm ²	[1x1] double	Manuell

So wird für jeden Modus (*mode*) die Bezeichnung (*mode.description*) und die Stromaufnahme (*mode.current*) hinterlegt. Zudem wird über den Parameter *mode.active* definiert, ob es sich bei dem Modus um einen aktiven (*mode.active*=1) oder inaktiven (*mode.active*=0) Modus handelt. Die Transitionen (*transitions*) werden, getrennt nach Dauer (*transitions.time*) und Stromaufnahme (*transitions.current*), als Matrix der Dimension $N_M \times N_M$ hinterlegt (vgl. Gleichung 5.2).

$$TR_{M,N} = \begin{pmatrix} tr_{1,1} & tr_{1,2} & \cdots & tr_{1,N} \\ tr_{2,1} & tr_{2,2} & \cdots & tr_{2,N} \\ \vdots & \vdots & \ddots & \vdots \\ tr_{M,1} & tr_{M,2} & \cdots & tr_{M,N} \end{pmatrix} \quad (5.2)$$

Als weitere Parameter der Komponente werden die Bezeichnung des Bauteils (*description*) sowie die minimale (*voltage.min*), typische (*voltage.typ*) und maximale Versorgungsspannung (*voltage.max*) angegeben. Sämtliche Angaben zur Stromaufnahme der Komponente beziehen sich immer auf die typische Versorgungsspannung. Für die Schnitt-

stellen einer Komponente N_{IF} werden die Beschreibung, der Verweis auf den LUT aller möglichen Schnittstellen (vgl. Unterabschnitt 5.5.7), die Datenrate und die Stromaufnahme hinterlegt. Die Dimensionen des Bauteils gehen als Fläche des Gehäuses (*area*) in das Modell ein.

Diese Parameter bilden als grundlegendes Komponentenmodell die Basis für die spezifischen Komponentenmodelle, die durch weitere Parameter für die einzelnen PMC (Mikrocontroller, Sensoren, Transceiver, Speicher) ergänzt werden können. Alle Parameter werden bei Definition einer neuen Komponenten zunächst mit Null initialisiert. Die anwendende Person kann verfügbare Angaben entweder dem jeweiligen Datenblatt oder zuvor publizierten wissenschaftlichen Arbeiten entnehmen. Nicht verfügbare Angaben können gegebenenfalls durch Messung ermittelt werden. Die Quelle und Art jedes Wertes kann in dem Textfeld der Bezeichnung angegeben werden, so dass die Angaben für jede anwendende Person transparent und nachvollziehbar sind. Falls keine weiteren Angaben gemacht werden handelt es sich um typische Werte. Somit entsteht eine wiederverwendbare und definierte Informationsbasis aller bereits evaluierten Komponenten.

5.5 Spezifische Komponentenmodelle

Im Folgenden werden die spezifischen Modelle für die einzelnen Komponenten eines Sensorknotens vorgestellt. Die Komponenten umfassen den Mikrocontroller, die Sensoren, sowie die Aktoren Funktransceiver und Speicher. Darüber hinaus wird das Modell der Schnittstellen zwischen den Komponenten und die Spannungsversorgung bestehend aus Spannungswandler und Energiespeicher betrachtet. Zur besseren Übersicht werden die Beschreibungen der spezifischen Komponentenmodelle anhand folgender Unterpunkte strukturiert:

- **Parameter und Funktionen:** Beschreibt die zu spezifizierenden Parameter des Komponentenmodells, die von der anwendenden Person eingegeben werden können. Zudem werden die im Modell hinterlegten Funktionen, die aus den eingegebenen Parametern weitere Parameter für das Modell generieren, beschrieben.
- **Schnittstelle und Datenfluss:** Beschreibt die abstrahierte Schnittstelle der Komponenten zu weiteren Komponenten im Systemmodell. Zudem wird der Datenfluss beschrieben. Dieser beinhaltet, ob es sich bei der jeweiligen Komponenten um eine Datenquelle oder Datensenke handelt und ob durch eine Funktion der Komponente die Datenbreite verändert wird.
- **Abhängigkeiten:** Beschreibt die Abhängigkeiten einer Komponenten und gibt somit einen Überblick darüber, welche Parameter von anderen Komponenten die eigenen Parameter und Funktionen beeinflussen.

5.5.1 Komponente Mikrocontroller

Heutige Mikrocontroller sind komplexe Bauteile, die aufgrund zahlreicher zu setzender Parameter über eine hohe Anzahl an Konfigurationen verfügen. Zu unterscheiden sind in erster Instanz die möglichen Betriebszustände. So stehen sowohl für den aktiven Betrieb, als auch für den inaktiven Zustand verschiedene Modi zur Verfügung, so dass der jeweilige Leistungsanspruch abgedeckt werden kann. Im aktiven Zustand kann die Leistungsfähigkeit mit Hilfe der Taktrate eingestellt werden (*frequency scaling*). Somit skaliert die Anzahl der Instruktionen, die pro Zeit verarbeitet werden können. Im inaktiven Zustand stehen verschiedene Modi zur Verfügung, in denen ein variabler Teil des Mikrocontrollers aktiv beziehungsweise inaktiv geschaltet wird (*power gating* verschiedener Spannungsdomänen). Je mehr Spannungsdomänen abgeschaltet werden, desto geringer ist die Leistungsaufnahme. Jedoch muss beachtet werden, dass neben dem Wegfall der entsprechenden Funktionen auch die für einen Wechsel zurück in einen aktiven Zustand benötigte Energie ansteigt. Hierbei gibt es für jeden inaktiven Zustand einen *Break-Even* Punkt, welcher abhängig vom Tastgrad ist (Nutzenzeit T_{BE}). Darüber hinaus kann für das Verlassen des inaktiven Zustands gegebenenfalls ein externes Signal nötig sein. Programme auf einem Mikrocontroller bestehen aus einer Initialisierungsphase, in der einzelne Module und Variablen konfiguriert werden und einem Hauptteil (*main*), der in einer Endlosschleife ausgeführt wird. In der Modellierung kann die Initialisierung vernachlässigt werden, da diese nur einmalig durchlaufen und anschließend nur der Hauptteil ausgeführt wird [16].

Parameter und Funktionen Die verschiedenen Zustände eines Mikrocontrollers (aktive und inaktive Modi) sind entsprechend des grundlegenden Komponentenmodells als textuelle Beschreibung zusammen mit der jeweiligen Stromaufnahme hinterlegt. Der Wechsel zwischen den Zuständen wird mittels der hinterlegten Transitionen abgebildet, welche die Dauer und die Stromaufnahme für jeden möglichen Wechsel zwischen den verschiedenen Zuständen enthalten. Die typischerweise in einem Mikrocontroller enthaltenden Schnittstellen werden im Modell des Mikrocontrollers nur vermerkt und gesondert modelliert (vgl. Kapitel 5.5.7), da deren Leistungsaufnahme abhängig vom Datenfluss ist.

Als Funktion des spezifischen Komponentenmodells werden die Nutzenzeiten T_{BE} für den Wechsel zwischen den aktiven (A) und inaktiven Modi (I) nach Benini u. a. [17] berechnet. Als Eingangsdaten für die Nutzenzeiten werden die Dauer der Transitionen (*transitions.time*), die Stromaufnahme der Transitionen (*transitions.current*) und die Stromaufnahme der Modi (*mode.current*) benötigt.

$$t_T = t_{A,I} + t_{I,A} \quad (5.3)$$

$$P_T = \frac{t_{A,I} \cdot P_{A,I} + t_{I,A} \cdot P_{I,A}}{t_T} \quad (5.4)$$

t_T ist die Summe aus der Dauer der beiden Transitionen zum wechseln in den inaktiven Zustand ($t_{A,I}$) und aus dem inaktiven Zustand ($t_{I,A}$). P_T ist die durchschnittliche Leistungsaufnahme der beiden Transitionen. Für den Fall, dass die durchschnittliche Leistungsaufnahme der Transitionen P_T kleiner oder gleich der Leistungsaufnahme P_A der Komponente im aktiven Zustand ist, entspricht die Nutzenzeit T_{BE} der Dauer der Transitionen t_T . Dieses bedeutet, dass ein Wechsel der Zustände sinnvoll ist, sobald dieser zeitlich möglich ist. Für den Fall, dass die Leistungsaufnahme der Transitionen P_T größer ist als die Leistungsaufnahme im aktiven Zustand P_A vergrößert sich die Nutzenzeit T_{BE} um den in Gleichung 5.5 aufgeführten Faktor:

$$T_{BE} = \begin{cases} t_T + t_T \cdot \frac{P_T - P_A}{P_A - P_I} & \text{falls } P_T > P_A \\ t_T & \text{falls } P_T \leq P_A \end{cases} \quad (5.5)$$

Der Faktor beschreibt die minimale Zeit, die die Komponente in dem inaktiven Zustand I betrieben werden muss, um die Leistungsaufnahme P_T der Transitionen zu kompensieren. Die Nutzendauer wird für jeden möglichen Wechsel zwischen den aktiven und inaktiven Modi der Komponente berechnet und in einem LUT der Dimension $N_M \times N_M$ hinterlegt.

Neben der berechneten Nutzendauer für den Wechsel der Modi umfassen die spezifischen Parameter die Taktrate für jeden Modus des Mikrocontrollers (vgl. Tabelle 5.2). Alle übrigen relevanten Parameter der Komponente Mikrocontroller werden bereits durch das grundlegende Komponentenmodell abgedeckt.

Tabelle 5.2.: Spezifische Parameter des Mikrocontrollers

Beschreibung	Parameter	Einheit	Format	Definition
Modi	<i>mode</i>	-	[1x N_M] struct	Manuell
- Taktrate	<i>mode.f</i>	MHz	[1x1] double	Manuell
Nutzenzeit	<i>tbe</i>	μs	[$N_M \times N_M$]	Funktion generiert LUT

Abstrahierte Schnittstelle und Datenfluss Die Komponente Mikrocontroller ist gleichzeitig Datenquelle und Datensenke. So senden die Sensoren ihre Messdaten an den Mikrocontroller. Diese werden durch einen Algorithmus verarbeitet und im Anschluss gegebenenfalls an einen Aktor gesendet. Der Algorithmus ist dabei in der Lage, die Datenbreite zu verändern, so dass dieser auch als eine Funktion f_{ALG} auf den Eingangsdaten D_I gesehen werden kann und die Ausgangsdaten D_O generiert:

$$D_O = f_{ALG}(D_I) \quad (5.6)$$

Die Funktion der Datenbreite kann manuell definiert werden (vgl. Taskdefinition in Abschnitt 5.8) oder durch eine Modellierung des Algorithmus generiert werden. Neben dem

Datenfluss wird der Mikrocontroller im System durch die abstrahierte Schnittstelle der Dauer und Leistungsaufnahme der Modi dargestellt.

Abhängigkeiten Für den Mikrocontroller hängt die Dauer der aktiven Modi, neben der Taktrate des Modus, von dem auszuführenden Algorithmus ab. Dieser wird von der anwendenden Person im Zuge der Taskkonfiguration ausgewählt (vgl. Abschnitt 5.8).

5.5.2 Komponente Sensor

Ein Sensor ist, wie auch der Mikrocontroller, durch die verschiedenen Modi und die Transitionen zwischen diesen bestimmt. Da es sich bei den aktiven Modi um Messungen handelt, werden für das spezifische Komponentenmodell entsprechende weitere Parameter definiert.

Parameter und Funktionen So werden für jeden aktiven Modus, also jede mögliche Messung des Sensors, jeweils die Dauer der Messung (*mode.time*) und die erzeugte Datenbreite der Messung (*mode.datawidth*) angegeben. Für jeden aktiven Modus wird vom Modell aus der Dauer der Messung zusammen mit der im allgemeinen Komponentenmodell definierten Versorgungsspannung und der Stromaufnahme des Modus die Energieaufnahme für eine Messung berechnet (*mode.em eas*). Die Bandbreite der Schnittstelle wird durch die zu übertragende Datenmenge (*mode.datawidth*) bestimmt, welche von der ausgewählten Messung abhängt (*mode*). Über den Parameter *mode.acc* kann eine Kennzahl für die Qualität der Messung angegeben werden, wobei die Einheit vom spezifischen Sensortyp abhängt. Die Spezifikation der Anzahl der verfügbaren Schnittstellen N_{IF} und deren Typ ist bereits in dem grundlegenden Komponentenmodell enthalten. Ein Überblick über die spezifischen Parameter der Komponente Sensor wird in Tabelle 5.3 gegeben.

Tabelle 5.3.: Spezifische Parameter des Sensors

Beschreibung	Parameter	Einheit	Format	Definition
Modi	<i>mode</i>	-	[$1 \times N_{M,A}$] struct	Manuell
- Datenbreite	<i>mode.datawidth</i>	Bit	[1×1] double	Manuell
- Dauer	<i>mode.time</i>	μs	[1×1] double	Manuell
- Qualität	<i>mode.acc</i>	div.	[1×1] double	Manuell
- Energie pro Messung	<i>mode.em eas</i>	$\mu W s$	[1×1] double	Generiert

Abstrahierte Schnittstelle und Datenfluss Ein Sensor dient im System als Datenquelle, wobei der gewählte aktive Modus, also die Messung einer spezifischen Größe, die generierte Datenmenge (*mode.datawidth*) und somit den Datenfluss bestimmt.

Abhängigkeiten Für die Komponente Sensor ist die Dauer des aktiven Modus spezifisch für jeden Modus (Messung) und somit im spezifischen Komponentenmodell hinterlegt (*mode.time*).

5.5.3 Komponente Funktransceiver

Auch der Funktransceiver wird über die Betriebszustände, die Transitionen und die Schnittstellen definiert.

Parameter und Funktionen Die spezifischen Parameter des Transceiver umfassen die Datenbreite und die Dauer der aktiven Modi (Senden, Empfangen). Datenbreite und Dauer sind, im Gegensatz zur Komponente Sensor, nicht fest definiert sondern werden durch eine Funktion generiert. Von der anwendenden Person ist lediglich die maximale Nutzlast (*mode.maxpayload*) und die Datenrate der drahtlosen Schnittstelle anzugeben (*mode.datarate*).

Tabelle 5.4.: Spezifische Parameter des Transceivers

Beschreibung	Parameter	Einheit	Format	Definition
Modi	<i>mode</i>	-	[1xN _M] struct	Manuell
- Maximale Nutzlast	<i>mode.maxpayload</i>	Byte	[1x1] double	Manuell
- Datenrate	<i>mode.datarate</i>	kbps	[1x1] double	Manuell
- Nutzdaten	<i>mode.datawidth</i>	Bit	[1xN _{PL}] double	Funktion generiert LUT
- Dauer	<i>mode.time</i>	μs	[1xN _{PL}] double	Funktion generiert LUT

Die daraus resultierende Datenbreite der drahtlosen Übertragung (*mode.datawidth*) wird mittels einer hinterlegten Funktion für den Protokolloverhead für den Bereich von 0 bis *maxpayload* Byte über eine lineare Funktion der Form $a \cdot D_I + b$ generiert und in einem LUT abgelegt. Die Dauer der drahtlosen Datenübertragung (*mode.time*) wird aus der möglichen Eingangsdatenbreite inklusive Protokolloverhead (*mode.datawidth*) und der Datenrate der drahtlosen Übertragung (*mode.datarate*) berechnet und ebenfalls in einem LUT abgelegt. N_{PL} entspricht der Anzahl der möglichen Datenbreiten im Bereich 1-*maxpayload* Byte. Ein Überblick über die spezifischen Parameter der Komponente Transceiver wird in Tabelle 5.4 gegeben.

Abstrahierte Schnittstelle und Datenfluss Ein Funktransceiver dient im System als Datensinke und nimmt die vom Mikrocontroller verarbeiteten Daten auf. Durch die im Modell hinterlegten Funktionen kann der Protokolloverhead der drahtlosen Schnittstelle und die Dauer der Übertragung berechnet werden. Die hinterlegten Funktionen sind spezifisch für jede Funktechnologie.

Abhängigkeiten Die Datenmenge und Dauer der Übertragung wird durch die Breite der Eingangsdaten bestimmt.

5.5.4 Komponente Speicher

Der Speicher wird über die Betriebszustände, die Transitionen und die Schnittstellen definiert.

Parameter und Funktionen Hinsichtlich der spezifischen Parameter wird die Dauer der aktiven Modi (Schreiben) durch eine im Modell hinterlegte Funktion generiert und in einem LUT hinterlegt. Als Eingangsgröße dient hier die Datenrate des Modus (*mode.datarate*). Ein Überblick über die spezifischen Parameter der Komponente Speicher wird in Tabelle 5.5 gegeben.

Tabelle 5.5.: Spezifische Parameter des Speichers

Beschreibung	Parameter	Einheit	Format	Definition
Modi	<i>mode</i>	-	[1xN _M] struct	Manuell
- Taktrate	<i>mode.datarate</i>	kbps	[1x1] double	Manuell
- Dauer	<i>mode.time</i>	-	[1xN _{PL}] double	Funktion generiert LUT

Abstrahierte Schnittstelle und Datenfluss Ein Speicher dient im System als Daten Senke und nimmt die vom Mikrocontroller verarbeiteten Daten auf. Entsprechend der typischen Funktionalität in drahtlosen Sensorknoten wird die Verwendung als Datenquelle (Lesen vom Speicher) nicht betrachtet.

Abhängigkeiten Die Dauer der Modi wird durch die Breite der Eingangsdaten bestimmt.

5.5.5 Komponente Spannungswandler

Das Modell des Spannungswandlers umfasst die Effizienz η in verschiedenen Arbeitspunkten. Bestimmend sind hier die Eingangsspannung, die Ausgangsspannung und der Ausgangsstrom des Spannungswandlers. Die Summe der Stromaufnahmen aller PMC ergibt die Gesamtstromaufnahme des Systems I_{SYS} und somit den Ausgangsstrom des Spannungswandlers. Zusammen mit der Versorgungsspannung des Systems U_{SYS} und der Klemmenspannung der Batterie U_{BAT} ergibt sich der Arbeitspunkt des Spannungswandlers. Dieser bestimmt maßgeblich die Effizienz und somit den Eingangsstrom des Wandlers I_{DC} , welcher, bei einer idealen Batterie, dem Entladestrom der Batterie I_{BAT} entspricht [18]:

$$I_{BAT} = I_{DC} = \frac{U_{SYS} \cdot I_{SYS}}{\eta \cdot U_{BAT}} \quad (5.7)$$

Im Komponentenmodell wird der Eingangsstrom des Spannungswandlers als I_{IN} , der Ausgangsstrom als I_{OUT} , die Eingangsspannung als U_{IN} und die Ausgangsspannung als U_{OUT} bezeichnet:

$$I_{IN} = \frac{U_{OUT} \cdot I_{OUT}}{\eta \cdot U_{IN}} \quad (5.8)$$

Spezifische Parameter für einen Spannungswandler sind die in Tabelle 5.6 aufgeführte minimale (*vin.min*) und maximale Eingangsspannung (*vin.max*), die Ausgangsspannung (*vout*) und der maximale Ausgangsstrom (*iout.max*).

Tabelle 5.6.: Spezifische Parameter des Spannungswandlers

Beschreibung	Parameter	Einheit	Format	Definition
Bezeichnung	<i>description</i>	-	String	Manuell
Minimale Eingangsspannung	<i>vin.min</i>	V	[1x1] double	Manuell
Maximale Eingangsspannung	<i>vin.max</i>	V	[1x1] double	Manuell
Ausgangsspannung	<i>vout</i>	V	[1x1] double	Manuell
Maximaler Ausgangsstrom	<i>iout.max</i>	mA	[1x1] double	Manuell
Angegebener Bereich des Ausgangsstroms	<i>input.iout</i>	mA	[1x N_{IOUTp}] double	Manuell
Effizienz bei min. Eingangsspannung	<i>input.effvinmin</i>	%	[1x N_{IOUTp}] double	Manuell
Effizienz bei max. Eingangsspannung	<i>input.effvinmax</i>	%	[1x N_{IOUTp}] double	Manuell
Voller Bereich der Eingangsspannung	<i>vin.range</i>	V	[1x N_{VIN}] double	Generiert
Voller Bereich des Ausgangsstroms	<i>iout.range</i>	mA	[1x N_{IOUT}] double	Generiert
Effizienz η des Spannungswandlers	<i>eff</i>	%	[1x N_{VIN}] cell	Generiert

Die Effizienz, in Abhängigkeit vom Arbeitspunkt, wird vom Hersteller für verschiedene Eingangsspannungen spezifiziert und kann dem jeweiligen Datenblatt entnommen werden. Für eine Verwendung der Daten im Modell müssen die Werte in vielen Fällen manuell aus einer Grafik ausgelesen werden. Der Hersteller *Texas Instruments* bietet zudem die Möglichkeit, die Daten für einige Spannungswandler auch in maschinenlesbarer Form zu exportieren [207]. Jedoch liefert dieser Export nur einen eingeschränkten Bereich der möglichen Arbeitspunkte des Wandlers. So wird die Effizienz nur für die minimale und die maximale Eingangsspannung und jeweils nur für einen Bereich des Ausgangsstroms ange-

geben, welcher nicht den gesamten Bereich von 0 mA bis zum maximalen Ausgangsstrom $i_{out,max}$ abdeckt. Der angegebene Bereich des Ausgangsstroms ($input.i_{out}$) und die Effizienz bei minimaler ($input.eff_{vinmin}$) und maximaler Eingangsspannung ($input.eff_{vinmax}$) gehen als Parameter in das Modell ein (vgl. Tabelle 5.6). N_{IOUTp} entspricht der Anzahl der im Datenblatt angegebenen Ausgangsströme für die auch die Effizienz angegeben wird. Damit die Effizienz des Spannungswandlers für alle Arbeitspunkte vorliegt, müssen diese Daten entsprechend extra- und interpoliert werden. Für die Effizienz über den gesamten Bereich der möglichen Eingangsspannungen wird zwischen der Effizienz bei minimaler und maximaler Eingangsspannung in 0,1 V Schritten linear interpoliert. Zur Abdeckung des gesamten Bereichs des Ausgangsstroms erfolgt anschließend für jede mögliche Eingangsspannung ein *Curve Fitting* der Effizienzkurven über den gesamten Bereich des Ausgangsstroms mittels einer kubisch hermiteschen Spline (*Piecewise Cubic Hermite Interpolation Polynomial* (PCHIP)) und es wird für jede Eingangsspannung ein *cfi*-Objekt generiert. Diese Fitting-Objekte ermöglichen es, die Effizienz des Spannungswandlers für die feste Ausgangsspannung v_{out} für den gesamten Arbeitsbereich aus Eingangsspannung ($vin.range$) und Ausgangsstrom ($i_{out.range}$) zu berechnen.

Eine Übersicht der von der anwendenden Person einzugebenden und der generierten Parameter findet sich in Tabelle 5.6. Es wird der mögliche Bereich der Eingangsspannung ($vin.range$) zwischen der minimalen Eingangsspannung $vin.min$ und der maximalen Eingangsspannung $vin.max$ in 0,1 V Schritten im Modell abgelegt. N_{VIN} entspricht die Anzahl der möglichen Eingangsspannungen. Zudem wird der mögliche Bereich des Ausgangsstroms ($i_{out.range}$) zwischen 0 mA und dem maximalen Ausgangsstrom $i_{out,max}$ in 0,01 mA Schritten im Modell abgelegt. N_{IOUT} entspricht der Anzahl der möglichen Ausgangsströme. Für die Generierung der Fitting-Objekte muss die Effizienz bei minimaler und bei maximaler Eingangsspannung ($input.eff_{vinmin}$, $input.eff_{vinmax}$) über einen Bereich des Ausgangsstromes ($input.i_{out}$) vorliegen. Diese müssen entweder manuell eingegeben werden oder können vom Hersteller exportiert werden. Die Fitting-Objekte werden in dem Parameter *eff* abgelegt, wobei N_{VIN} der Anzahl der möglichen Eingangsspannungen über den gesamten Bereich in 0,1 V Schritten entspricht.

5.5.6 Komponente Energiespeicher

Der Eingangsstrom des Spannungswandlers entspricht dem Ausgangsstrom der Batterie, wobei dieser die Effizienz der Energieentnahme aus dem Energiespeicher bestimmt. Die Effizienz ist dabei abhängig von verschiedenen Batterieeffekten (vgl. Abschnitt 2.2) und spezifisch für jeden Energiespeicher. Zur Modellierung einer Batterie und somit zur Berücksichtigung dieser Effekte existieren in der Literatur zahlreiche Modelle (vgl. Abschnitt 4.7), die sich hinsichtlich Komplexität und Genauigkeit unterscheiden. Wie in Abschnitt 4.10 beschrieben, stellt die Generierung der benötigten Parameter eine große Herausforderung dar, da die für den Aufbau der Modelle benötigten Daten oftmals nicht in den Datenblättern der Batteriehersteller enthalten sind. Da die Angaben in den Datenblättern zudem sehr heterogen sind, werden von Petricca u. a. [154] und Bocca u. a. [24] verschiedene Ebenen

für ein zu generierendes Modell definiert. Dieses Konzept wird hier übernommen, da es der Anforderung des modularen Modells mit verschiedenen Abstraktionsebenen entspricht und somit der Skalierbarkeit des Modells dient (vgl. Unterabschnitt 5.2.2).

Im Folgenden wird zunächst das Basismodell für die Komponente Batterie beschrieben. Anschließend wird das erweiterte Modell erläutert.

Basismodell Als Basismodell für die Batterie wird das mathematische Modell nach Peukert [155] verwendet (vgl. Unterabschnitt 4.7.1), welches die Laufzeit eines Systems bei einer Entladung mit konstantem Strom empirisch ermittelt:

$$T_{\text{SYS}} = \frac{C_{\text{PK}}}{I_{\text{BAT}}^k} = \frac{C_{\text{BAT}}}{I_N} \cdot \left(\frac{I_N}{I_{\text{BAT}}} \right)^k \quad (5.9)$$

Hier bezeichnet C_{PK} die Peukert-Kapazität, k die dimensionslose Peukert-Zahl und T_{SYS} die Laufzeit des Systems. Die Peukert-Kapazität C_{PK} ist die Nennkapazität bei einem Entladestrom von 1 A. Da die Nennkapazität einer Batterie in den meisten Fällen nicht für einen Entladestrom von 1 A angegeben wird, kann die Peukert-Gleichung entsprechend angepasst werden. Hier entspricht dann C_{BAT} der vom Hersteller angegebenen Nennkapazität bei Nennentladestrom I_N . Das mathematische Modell nach Peukert wird auch von Petricca u. a. [154] als Basismodell (Modell der Ebene 1, vgl. Unterabschnitt 4.7.3) verwendet, wobei die Autoren auch die Generierung des Modells beschreiben, die hier übernommen wird.

Aus dem Datenblatt der abzubildenden Batterie wird das Verhalten der Klemmenspannung während der Entladezeit für verschiedene konstante Entladeströme benötigt. Die Punkte, in welchen jeweils die Entladeschlussspannung erreicht wird, bestimmen die Laufzeit der Batterie, so dass sich mehrere Datenpaare aus Entladestrom und Laufzeit ergeben. Auf Basis dieser Datenpaare können mittels *Curve Fitting* die für das Modell benötigten Parameter Peukert-Kapazität C_{PK} und Peukert-Zahl k ermittelt werden. Da laut Gleichung 5.9 für die Laufzeit des Systems $T_{\text{SYS}} = C_{\text{PK}} \cdot I_{\text{BAT}}^{-k}$ gilt, kann mittels Regression eine Kurve der Form $y = a \cdot x^{-b}$ angenähert werden. Somit können die Parameter Peukert-Kapazität C_{PK} und Peukert-Zahl k für spezifische Batterien ermittelt werden.

Der aktuelle SOC einer Batterie und somit die Laufzeit des Systems ergeben sich entsprechend aus dem Entladestrom, der Peukert-Kapazität C_{PK} und der Peukert-Zahl k . Entsprechend der elektrochemischen Verzögerung von Batterien wird der Entladestrom der Batterie mit einem gleitenden Mittelwert geglättet. In elektrischen Modellen wird dieses über die Zeitkonstante des RC-Netzwerks abgebildet, wobei die Fensterbreite typischerweise bei 1 s liegt [71; 188].

$$\text{SOC}(t) = \text{SOC}_0 - \frac{1}{C_{\text{PK}}} \int_0^{T_s} I_{\text{BAT}}(t)^k dt \quad (5.10)$$

Die spezifischen Parameter für das Basismodell der Komponente Batterie werden in Tabelle 5.7 aufgeführt. N_{RT} bezeichnet die Anzahl der verfügbaren Laufzeiten bei verschiedenen Entladeströmen, wobei jeweils Wertepaare hinterlegt werden. Diese sind in 79 % der von Bocca u. a. [24] untersuchten Datenblätter von Lithium-Ionen Primärzellen enthalten.

Tabelle 5.7.: Spezifische Parameter für das Basismodell der Batterie

Beschreibung	Parameter	Einheit	Format	Definition
Bezeichnung	<i>description</i>	-	String	Manuell
Nennspannung U_{BAT}	<i>voltage.nom</i>	V	[1x1] double	Manuell
Nennkapazität C_{BAT}	<i>cap.nom</i>	mAh	[1x1] double	Manuell
Typ. Entladestrom	<i>current.typ</i>	mA	[1x1] double	Manuell
Max. Entladestrom (kontinuierlich)	<i>current.maxcont</i>	mA	[1x1] double	Manuell
Max. Entladestrom (gepulst)	<i>current.maxpuls</i>	mA	[1x1] double	Manuell
Entladeschlussspannung U_{CUT}	<i>voltage.cut</i>	V	[1x1] double	Manuell
Laufzeit bei verschiedenen Entladeströmen	<i>voltage.rt</i>	mA,h	[$N_{RT} \times 2$] double	Manuell
Peukert-Kapazität C_{PK}	<i>cap.pk</i>	mAh	[1x1] double	Generiert
Peukert-Zahl k	<i>k</i>	-	[1x1] double	Generiert

Erweitertes Modell Das Basismodell ist in der Lage, die Entladung bei verschiedenen Ladeströmen abzubilden und stellt somit potentiell eine Verbesserung gegenüber der Verwendung eines idealen Batteriemodells dar. Jedoch kann das Basismodell nur als eine erste Näherung betrachtet werden, da es vor allem bei dynamischen Lasten hohe Abweichungen aufweisen kann (vgl. Unterabschnitt 4.7.1). Zudem ist der für das Basismodell benötigte Verlauf der Klemmenspannung über der Zeit für verschiedene konstante Entladeströme in den von Bocca u. a. [24] untersuchten Datenblättern für Sekundärzellen nur in 8 % der Fälle enthalten. Es wird daher ein elektrisches Laufzeit-Modell als erweitertes Modell für die Komponente Batterie definiert (vgl. Abbildung 5.6). Dieses bildet im linken Teil die nutzbare Kapazität C_{BAT} in Abhängigkeit vom Entladestrom I_{BAT} ab. Der rechte Teil stellt die Abhängigkeit der Batteriespannung U_{BAT} vom SOC und dem Entladestrom I_{BAT} dar. Die Abhängigkeit vom Entladestrom I_{BAT} ergibt sich durch den Innenwiderstand R_0 , welcher, neben dem Entladestrom, auch vom SOC abhängt. Das Vorgehen zur Generierung des Modells entspricht dem von Bocca u. a. [24]. Somit kann das Modell aus den Angaben des Datenblatts generiert werden.

Laut Bocca u. a. [24] enthalten 89 % der Datenblätter von Sekundärzellen den für die Generierung des Modells notwendigen Verlauf der Spannung in Abhängigkeit des

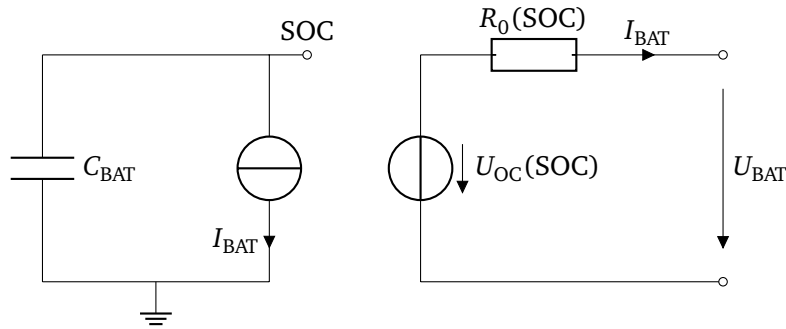


Abbildung 5.6.: Generiertes Laufzeit-Batteriemodell nach Bocca u. a. [24]

SOC für mindestens zwei verschiedene Entladeströme. Aus diesen Spannungsverläufen in Abhängigkeit vom SOC kann der Innenwiderstand der Batterie R_0 ermittelt werden [24; 154]:

$$R_0(\text{SOC}) = \frac{U_{\text{BAT},1}(\text{SOC}) - U_{\text{BAT},2}(\text{SOC})}{I_{\text{BAT},2} - I_{\text{BAT},1}} \quad (5.11)$$

Wobei $I_{\text{BAT},1}$ und $I_{\text{BAT},2}$ zwei konstante Entladeströme sind und $I_{\text{BAT},2} > I_{\text{BAT},1}$ gilt.

Somit ist es möglich, die Leerlaufspannung U_{OC} des elektrischen Modells zu berechnen, wodurch der Zusammenhang mit dem Entladestrom I_{BAT} und dem SOC charakterisiert wird [24; 154]:

$$U_{\text{OC}}(\text{SOC}) = U_{\text{BAT}} + R_0(\text{SOC}) \cdot I_{\text{BAT}} \quad (5.12)$$

Falls der Verlauf der Spannung in Abhängigkeit des SOC nur für einen Entladestrom angegeben ist, kann der Innenwiderstand R_0 nur statisch und somit nicht in Abhängigkeit vom SOC berücksichtigt werden. Für die statische Betrachtung muss der Innenwiderstand als Wert im Datenblatt spezifiziert sein, was laut Bocca u. a. [24] für Sekundärzellen in 54 % der Datenblätter der Fall ist.

Der SOC der Batterie wird entsprechend der Gleichung 5.13 berechnet, wobei der Entladestrom der Batterie zuvor wie bei dem Basismodell mit einem gleitenden Mittelwert geglättet wird.

$$\text{SOC}(t) = \text{SOC}_0 - \frac{1}{C_{\text{BAT}}} \int_0^{T_s} I_{\text{BAT}}(t) dt \quad (5.13)$$

Der jeweils aktuelle SOC und der Entladestrom I_{BAT} gehen in die generierten Funktionen für den Innenwiderstands R_0 (Gleichung 5.11) und die Leerlaufspannung U_{OC} (Gleichung 5.12) ein. Auf dieser Basis kann die aktuelle Klemmenspannung der Batterie U_{BAT} berechnet werden:

$$U_{\text{BAT}} = U_{\text{OC}}(\text{SOC}) - R_0(\text{SOC}) \cdot I_{\text{BAT}} \quad (5.14)$$

Die Parameter des erweiterten Batteriemodells sind in Tabelle 5.8 dargestellt. Zur Generierung des Modells wird der Verlauf der Klemmenspannung in Abhängigkeit vom SOC ($vsoc1$, $vsoc2$) für zwei verschiedene Entladeströme ($i1$, $i2$) benötigt. In $vsoc1$ und $vsoc2$ sind N_{SOC1} bzw. N_{SOC2} Datenpaare aus entladener Kapazität und Klemmenspannung hinterlegt. Die Eingabe der Daten kann durch manuelle Extraktion, automatisiertes Parsen der Daten aus dem Datenblatt oder durch eine Bereitstellung in digitaler Form durch den Hersteller erfolgen. Aus diesen Daten werden die Funktionen für den Innenwiderstand R_0 und die Leerlaufspannung U_{OC} generiert und als MATLAB Fitting-Objekte hinterlegt ($fitobjr0$, $fitobjuoc$).

Tabelle 5.8.: Spezifische Parameter für das erweiterte Modell der Batterie

Beschreibung	Parameter	Einheit	Format	Definition
Bezeichnung	<i>description</i>	-	String	Manuell
Nennspannung U_{BAT}	<i>voltage.nom</i>	V	[1x1] double	Manuell
Nennkapazität C_{BAT}	<i>cap.nom</i>	mAh	[1x1] double	Manuell
Entladeschlussspannung U_{CUT}	<i>voltage.cut</i>	V	[1x1] double	Manuell
Entladestrom $i1$	<i>i1</i>	mA	[1x1] double	Manuell
Entladestrom $i2$	<i>i2</i>	mA	[1x1] double	Manuell
Klemmenspannung vs. SOC bei $i1$	<i>vsoc1</i>	mAh,V	[$N_{SOC1} \times 2$] double	Manuell
Klemmenspannung vs. SOC bei $i2$	<i>vsoc2</i>	mAh,V	[$N_{SOC2} \times 2$] double	Manuell
Funktion des Innenwiderstands R_0	<i>fitobjr0</i>	Ω	[1x1] cfit	Generiert
Funktion der Leerlaufspannung U_{OC}	<i>fitobjuoc</i>	V	[1x1] cfit	Generiert

Somit liefert das Modell die Klemmenspannung in Abhängigkeit vom SOC und dem Entladestrom. Da der Entladestrom der Batterie dem Eingangsstrom des Spannungswandlers entspricht, führt das Absinken der Klemmenspannung während der Entladung dazu, dass sich der Arbeitspunkt des Spannungswandlers verschiebt und somit die Effizienz beeinflusst wird. Bei Verwendung des erweiterten Batteriemodells kann diese Wechselwirkung somit in die Modellbildung bzw. Simulation des Systems einbezogen werden.

5.5.7 Schnittstellen zwischen den Komponenten

Die Schnittstellen zwischen den Komponenten dienen der Datenübertragung im System. Im Modell wird der Einfluss der Schnittstelle mittels einer Funktion abgebildet. So berechnet die Funktion den Overhead der jeweiligen Schnittstelle, so dass sich ein Zusammenhang zwischen der zu übertragenden Datenmenge der Komponenten D_I (*Payload*) und der real über die Schnittstelle zu übertragenden Datenmenge D_O ergibt. Die in Tabelle 5.9 aufgeführten Funktionen bilden den von Mikhaylov und Tervonen [131] ermittelten jeweiligen Overhead ab.

Tabelle 5.9.: Modell der Schnittstellen. D_I und D_O werden in Bit angegeben.

Nummer	Beschreibung	Funktion f_{IF}
1	UART	$D_O = (1 + \frac{2}{8}) \cdot D_I$
2	SPI	$D_O = D_I$
3	I2C	$D_O = (1 + \frac{1}{8}) \cdot D_I + 11$

Die Nummer entspricht der im grundlegenden Komponentenmodell definierten Nummer der Schnittstelle *interface.no* (vgl. Abschnitt 5.4). Zusammen mit der im spezifischen Modell für die Komponenten Sensor definierten Datenrate der Schnittstelle *interface.datarate* (vgl. Unterabschnitt 5.5.1) ergibt sich die Bandbreite der jeweiligen Schnittstelle als Produkt aus Datenrate und Datenbreite. Die Datenbreite wird durch die im System enthaltenden Sensoren und die Informationsverarbeitung bestimmt und wird im folgenden Abschnitt 5.6 näher beschrieben.

5.6 Datenfluss

Der Datenfluss über die Schnittstellen im System wird als Datenflussdiagramm abgebildet. So stellen die im System enthaltenden Sensoren die Datenquelle dar, wobei jeder Sensor bei einer Messung Daten der Breite D_{Si} generiert. Die Datenbreite einer Messung ist im Komponentenmodell des jeweiligen Sensors definiert (*mode.datarate*). In Abbildung 5.7 ist der Datenfluss in einem System mit drei Sensoren exemplarisch dargestellt. Anhängig von der Schnittstelle des Sensors wird dieser Datenbreite der Overhead der Schnittstelle hinzugefügt:

$$D_{IFSi} = f_{IFi}(D_{Si}) \quad (5.15)$$

Die über die jeweiligen Schnittstellen übermittelten Messdaten der Sensoren der Breite D_{IFSi} bilden die Eingangsdaten des Mikrocontrollers. Da die auf dem Mikrocontroller auszuführenden Algorithmen in den meisten Fällen auch einen Einfluss auf die Datenbreite haben, wird dieser Einfluss als Funktion auf den Eingangsdaten der Sensoren D_{Si} abgebildet:

$$D_{UCi} = f_{ALGi} \left(\sum_{i=1}^{N_{SEN}} D_{Si} \right) \quad (5.16)$$

Die vom Mikrocontroller verarbeiteten Daten der Breite D_{IFUCi} werden über die i -te Schnittstelle des Mikrocontrollers auf einen Aktor gegeben, wobei es sich hier um einen Funktransceiver oder um einen Speicher handeln kann. Die Datenbreite erhöht sich entsprechend des Overheads der Schnittstelle:

$$D_{IFUCi} = f_{IFi}(D_{UCi}) \quad (5.17)$$

Die Datenbreiten für die verschiedenen Akteure können sich entsprechend der Anwendung unterscheiden. Beispielsweise könnte ein reduziertes Datenformat drahtlos übertragen und die Rohdaten auf dem Sensorknoten intern gespeichert werden. Dieses kann durch verschiedene Algorithmen auf dem Mikrocontroller abgebildet werden, welche entsprechend die Datenbreite beeinflussen. Die Akteure im System stellen die Datensenden dar.

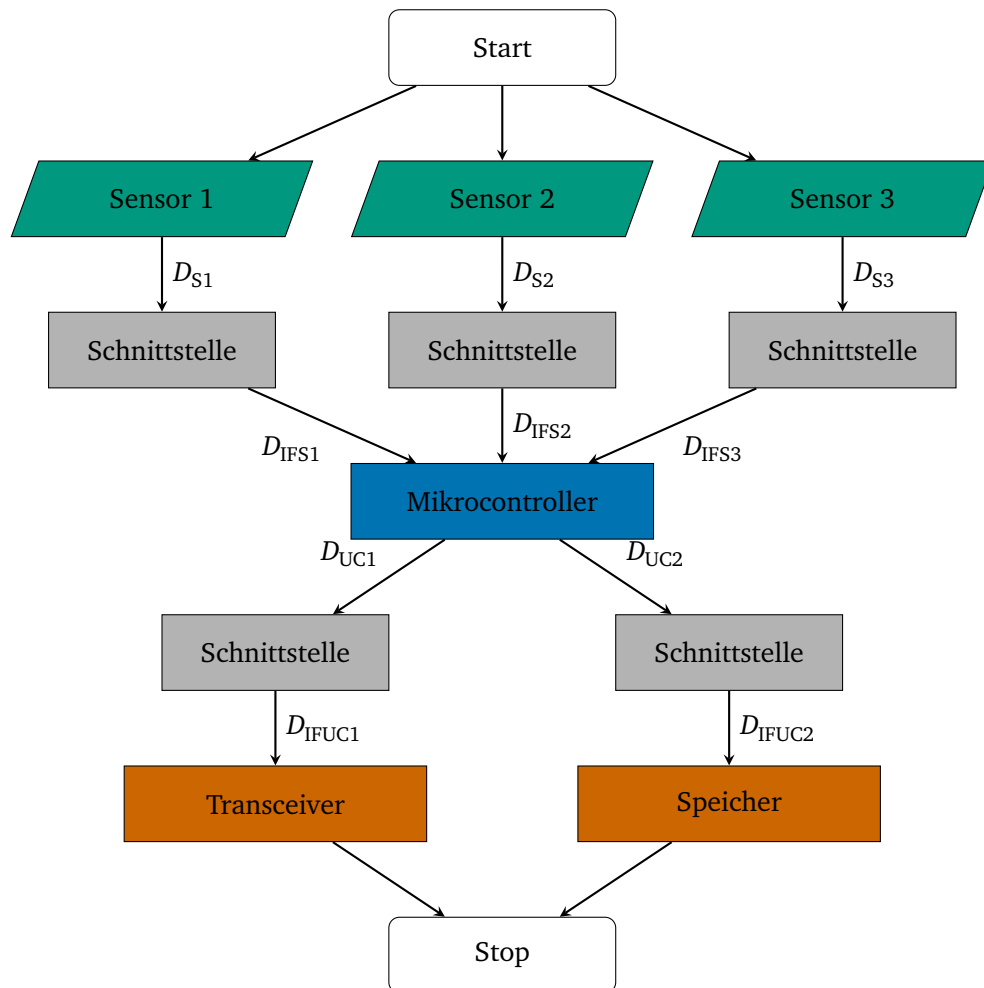


Abbildung 5.7.: Datenfluss im Systemmodell

5.7 Systemkonfiguration

Auf Basis der modellierten Komponenten können verschiedene Gesamtsysteme konfiguriert werden. Eine Systemkonfiguration legt dabei die Komponenten des Systems fest

und definiert die aktiven und inaktiven Modi jeder Komponente. Im Folgenden werden die Parameter der Systemkonfiguration erläutert und die generierten Kennzahlen für die Auswertung des Entwurfsraums auf Ebene der Systemkonfiguration beschrieben.

Parameter Die Parameter der Systemkonfiguration sind in Tabelle 5.10 aufgeführt. Im Rahmen der Systemkonfiguration werden die Komponenten für die Typen Mikrocontroller, Sensoren, Transceiver, Speicher, Spannungswandler und Energiespeicher ausgewählt. Für die aktiven Komponenten (PMC) Mikrocontroller, Sensoren, Transceiver und Speicher werden zudem die aktiven und inaktiven Modi definiert, die in den Entwurfsraum der Systemkonfigurationen eingehen sollen.

Tabelle 5.10.: *Parameter der Systemkonfiguration*

Beschreibung	Parameter	Einheit	Format	Definition
Mikrocontroller	<i>uc</i>	-	[1x1] struct	Manuell
- Typ	<i>uc.devices</i>	-	[1x1] struct	Manuell
- Aktiver Modus	<i>uc.mactive</i>	-	[1xN _{M,A}] double	Manuell
- Inaktiver Modus	<i>uc.midle</i>	-	[1xN _{M,I}] double	Manuell
- Interface	<i>uc.if</i>	-	[1x1] double	Manuell
Sensoren	<i>sensors</i>	-	[1x1] struct	Manuell
- Typ	<i>sensors.devices</i>	-	[1xN _{SEN}] struct	Manuell
- Aktiver Modus	<i>sensors.mactive</i>	-	[1xN _{M,A}] double	Manuell
- Inaktiver Modus	<i>sensors.midle</i>	-	[1xN _{M,I}] double	Manuell
- Interface	<i>sensors.if</i>	-	[1x1] double	Manuell
Transceiver	<i>tranceiver</i>	-	[1x1] struct	Manuell
- Typ	<i>tranceiver.devices</i>	-	[1x1] struct	Manuell
- Aktiver Modus	<i>tranceiver.mactive</i>	-	[1xN _{M,A}] double	Manuell
- Inaktiver Modus	<i>tranceiver.midle</i>	-	[1xN _{M,I}] double	Manuell
- Interface	<i>tranceiver.if</i>	-	[1x1] double	Manuell
Speicher	<i>memory</i>	-	[1x1] struct	Manuell
- Typ	<i>memory.devices</i>	-	[1x1] struct	Manuell
- Aktiver Modus	<i>memory.mactive</i>	-	[1xN _{M,A}] double	Manuell
- Inaktiver Modus	<i>memory.midle</i>	-	[1xN _{M,I}] double	Manuell
- Interface	<i>memory.if</i>	-	[1x1] double	Manuell
Spannungswandler	<i>converter</i>	-	[1x1] struct	Manuell
- Typ	<i>converter.devices</i>	-	[1xN _{DC}] struct	Manuell
Energiespeicher	<i>bat</i>	-	[1x1] struct	Manuell
- Typ	<i>bat.devices</i>	-	[1xN _{BAT}] struct	Manuell

Die in Tabelle 5.10 aufgeführten Struktur-Arrays für die verschiedenen Typen (*.devices) enthalten die zuvor generierten Modelle der jeweiligen Komponenten. Die Bildung des

Entwurfsraums wird über die Variation der Komponenten und ihrer Modi im Rahmen von verschiedenen Systemkonfigurationen ermöglicht. Zudem wird mittels (*.if) für jede PMC die Schnittstelle ausgewählt, so dass im System eine einheitliche Schnittstelle verwendet wird (z.B. I2C). Laut Definition besteht ein System aus genau einem Mikrocontroller, einem Funktransceiver, einem oder mehreren Sensoren, einem Spannungswandler, einem Energiespeicher und optional einem Speicher. Jede Systemkonfiguration umfasst somit zunächst einen Microcontroller, einen Transceiver, bis zu N_{SEN} Sensoren und optional einen Speicher. Diese Systemkonfiguration der PMC kann zudem für mehrere Spannungswandler (N_{DC}) und mehrere Energiespeicher (N_{BAT}) untersucht werden. Die Größe des Entwurfsraums für das System $N_{\text{EWR,SYS}}$ ergibt sich durch die Anzahl der verschiedenen Spannungswandler N_{DC} , die Anzahl der Energiespeicher N_{BAT} und die Größe des Entwurfsraums der aktiven Komponenten $N_{\text{EWR,PMC}}$.

$$N_{\text{EWR,SYS}} = N_{\text{BAT}} \cdot N_{\text{DC}} \cdot N_{\text{EWR,PMC}} \quad (5.18)$$

Die Größe des Entwurfsraums der aktiven Komponenten ergibt sich aus der Anzahl der verschiedenen Mikrocontroller N_{UC} , Sensoren N_{SEN} , Transceiver N_{TC} und Speicher N_{MEM} sowie der Anzahl der aktiven und inaktiven Modi ($N_{M,A}$ und $N_{M,I}$) dieser Komponenten.

$$N_{\text{EWR,PMC}} = \sum_{i=1}^{N_{\text{UC}}} N_{M,A,i} N_{M,I,i} \cdot \sum_{i=1}^{N_{\text{SEN}}} N_{M,A,i} N_{M,I,i} \cdot \sum_{i=1}^{N_{\text{TC}}} N_{M,A,i} N_{M,I,i} \cdot \sum_{i=1}^{N_{\text{MEM}}} N_{M,A,i} N_{M,I,i} \quad (5.19)$$

Der so entstehende Entwurfsraum aus verschiedenen Systemkonfigurationen wird im Modell als mat-Datei hinterlegt.

Auswertung Die auf Basis der Komponentenmodelle definierten Systemkonfigurationen ermöglichen die Berechnung der Leistungsaufnahme des Systems im inaktiven Zustand (vgl. Tabelle 5.11). Diese entspricht der Summe der Leistungsaufnahme aller Komponenten des Systems im inaktiven Zustand und wird für jede Systemkonfiguration der

Tabelle 5.11.: Kennzahlen des inaktiven Modus

Beschreibung	Parameter	Einheit	Format
Inaktiver Modus	<i>idle</i>	-	[1x1] struct
- Leistungsaufnahme PMC	<i>idle.pmcpower</i>	mW	[1x $N_{\text{EWR,PMC}}$] double
- Anteilige Leistungsaufnahme PMC	<i>idle.pmcpowerratio</i>	mW	[$N_{\text{PMC}} \times N_{\text{EWR,PMC}}$] double
- Leistungsaufnahme Spannungswandler	<i>idle.dcpower</i>	mW	[$N_{\text{DC}} \times N_{\text{EWR,PMC}}$] double
- Effizienz des Spannungswandlers	<i>idle.dceff</i>	mW	[$N_{\text{DC}} \times N_{\text{EWR,PMC}}$] double
- Optimaler Spannungswandler	<i>idle.optdc</i>	-	[1x $N_{\text{EWR,PMC}}$] double

PMC im Entwurfsraum berechnet (*idle.pmcpower*). Zudem wird der Anteil der einzelnen Komponenten an der Leistungsaufnahme für den inaktiven Modus (*idle.pmcpowerratio*) berechnet. Die Anzahl der Anteile entspricht der Anzahl der PMC. Zusätzlich wird für jede Kombination aus PMC und Spannungswandler die Leistungsaufnahme des inaktiven Zustands (*idle.dcpower*) und die Effizienz des Spannungswandlers in diesem Arbeitspunkt (*idle.dceff*) ermittelt. Somit wird für jede Konfiguration der PMC der effizienteste Spannungswandler (*idle.optdc*) bestimmt. Die generierten Kennzahlen werden der mat-Datei des untersuchten Entwurfsraums der Systemkonfigurationen hinzugefügt.

5.8 Taskkonfiguration

Für die zuvor definierten Systemkonfigurationen können anschließend verschiedene Tasks definiert werden. Es gehen die in der Systemkonfiguration ausgewählten Mikrocontroller (*uc*), Sensoren (*sensors*) und Aktoren (*transceiver*, *memory*) ein. Im Folgenden werden die Parameter der Taskkonfiguration erläutert und die generierten Kennzahlen für die Auswertung des Entwurfsraums auf Ebene der Tasks beschrieben.

Parameter Die in Tabelle 5.12 aufgeführten Parameter einer Taskkonfiguration legen fest, welche Komponenten des Systems in welchem Umfang aktiv sind.

Tabelle 5.12.: Parameter der Taskkonfiguration

Beschreibung	Parameter	Einheit	Format	Definition
Task	<i>taskconf</i>	-	[1xN _T] struct	-
- Aktivierung der Komponententypen	<i>taskconf.devtypesactive</i>	-	[1xN _{CT}] double	Manuell
- Aktive Zeit des Mikrocontrollers	<i>taskconf.uccycles</i>	Zyklen	[1x1] double	Man./Gen.
- Datenoperation des Mikrocontrollers	<i>taskconf.falg</i>	-	Funktion	Man./Gen.
- Wechsel der Modi	<i>taskconf.ucsleep</i>	-	[1x1] double	Manuell

Die Dauer der aktiven Zeit des Mikrocontrollers wird von dem auszuführenden Algorithmus bestimmt. Die aktive Zeit *taskconf.uccycles* in Taktzyklen kann manuell angegeben oder von einem Modell des auszuführenden Algorithmus generiert werden. Die manuelle Eingabe der für die Ausführung benötigten Anzahl Taktzyklen ermöglicht eine erste Abschätzung (z.B. auf Basis von Erfahrungswerten) des Systems oder auch die Einbindung der Ergebnisse einer bereits durchgeführten Analyse (z.B. durch Messung). Somit kann die Laufzeit abgeschätzt, gemessen oder mit Hilfe eines Modells bestimmt werden, wodurch sich verschiedene Abstraktionsebenen ergeben. Zudem kann der in Gleichung 5.16

beschriebene Einfluss des auszuführenden Algorithmus auf die Datenbreite mittels der Funktion *taskconf.falg* abgebildet werden. Diese kann ebenfalls manuell eingetragen oder von einem Modell des Algorithmus generiert werden. Für die Komponenten vom Typ Sensor, Transceiver und Speicher kann über die Parameter *taskconf.dectypesactive* definiert werden, ob diese in dem definierten Task aktiv (1) oder inaktiv (0) sind. Die aktiven Zeiten der Komponenten, die Dauer der Datenübertragung zwischen den Komponenten und die Datenbreite des Transceivers werden auf Basis der Komponentenmodelle und der Systemkonfiguration automatisch berechnet. Der Parameter *taskconf.ucsleep* legt fest, ob der Mikrocontroller während der Messung der Sensoren in den inaktiven Zustand wechselt. Der Wechsel erfolgt jedoch nur, wenn die Nutzenzeit T_{BE} kleiner ist als die Dauer des Messung.

Als Basis für die Berechnung der Dauer der Datenübertragung über die Schnittstellen des Systems dienen hier zunächst die im Komponentenmodell hinterlegten Nummern der Schnittstellen *interface.no* und die in Tabelle 5.9 aufgeführten Funktionen für den Overhead der jeweiligen Schnittstelle. Somit ergibt sich die Datenbreite inklusive Overhead D_{IFS_i} für jeden Sensor. Zusammen mit der im Komponentenmodell hinterlegten Datenrate der Schnittstelle *interface.datarate* wird die Dauer der Übertragung für jeden Sensor berechnet. Die aktive Zeit des Transceivers wird von der Breite der Daten vom Mikrocontroller D_{UC_i} und dem Overhead der Schnittstelle zwischen Mikrocontroller und Transceiver bestimmt (vgl. Gleichung 5.17). Die daraus resultierende Datenbreite D_{IFS_i} dient als Eingabeparameter für den im Komponentenmodell (vgl. Unterabschnitt 5.5.3) des Transceivers generierten LUT *transceivers.mode.time*.

Die Übertragung der Daten vom Mikrocontroller zum Transceiver ergibt sich aus der Datenbreite inklusive Overhead D_{IFUC_i} und der Datenrate der Schnittstelle *interface.datarate*. Die drahtlos zu übertragende Datenbreite des Transceivers basiert auf der Breite der Daten vom Mikrocontroller D_{UC_i} und dem Overhead des Funkprotokolls. Dieser Overhead ist im Komponentenmodell des Transceivers durch den LUT *transceivers.mode.datawidth* abgebildet (vgl. Unterabschnitt 5.5.3). Die aktive Zeit des Speichers und die Dauer der Datenübertragung ergibt sich analog zur Berechnung der aktiven Zeit für den Transceiver. Abhängig vom Parameter *taskconf.ucsleep* ist der Mikrocontroller während des gesamten Ablaufs im aktiven Modus oder wechselt, abhängig von der Nutzenzeit T_{BE} , in den inaktiven Modus. Die aktive Zeit der übrigen Komponenten und die Datenübertragung zwischen den Schnittstellen addieren sich zur gesamten aktiven Zeit des Mikrocontrollers und somit zur Gesamtdauer des Tasks.

Auswertung Auf Basis der Systemkonfigurationen und der Taskkonfigurationen können die Kennzahlen für den Entwurfsraum $N_{EWR,PMC}$ und die Spannungswandler generiert werden (vgl. Tabelle 5.13).

Diese umfassen die Dauer (*task.duration*), die Nutzlast (*task.payload*) sowie die Leistungsaufnahme der PMC (*task.pmcpower*). Die Leistungsaufnahme wird für den gesamten

Tabelle 5.13.: Kennzahlen der aktiven Modi

Beschreibung	Parameter	Einheit	Format
Aktive Modi	<i>task</i>	-	[1x N_T] struct
- Dauer	<i>task.duration</i>	μs	[1x1] double
- Nutzlast	<i>task.payload</i>	Bit	[1x1] double
- Leistungsaufnahme	<i>task.pmcpower</i>	mW	[1x N_{LT}] double
- Mittlere Leistungsaufnahme	<i>task.pmcmeanpower</i>	mW	[1x1] double
- Energieaufnahme	<i>task.pmcenergy</i>	mWs	[1x1] double
- Anteilige Energieaufnahme	<i>task.pmcenergyratio</i>	mWs	[1x N_{PMC}] double
- Leistungsaufnahme Spannungswandler	<i>task.dcpower</i>	mW	[N_{DC} x N_{LT}] double
- Mittl. Leistungsaufn. Spannungswandler	<i>task.dcmeanpower</i>	mW	[N_{DC} x1] double
- Energieaufnahme Spannungswandler	<i>task.denergy</i>	mWs	[N_{DC} x1] double
- Effizienz des Spannungswandlers	<i>task.dceff</i>	%	[N_{DC} x1] double
- Optimaler Spannungswandler	<i>task.optdc</i>	-	[1x1] double

Task in einer μs Auflösung berechnet. N_{LT} ist entsprechend die Anzahl der Datenpunkte für einen Task. Zudem wird die mittlere Leistungsaufnahme *task.pmcmeanpower* für den Task berechnet. Aus der Dauer des Tasks *task.duration* und der mittleren Leistungsaufnahme *task.pmcmeanpower* wird die Energieaufnahme des Task *task.pmcenergy* berechnet. Zudem wird der Anteil der einzelnen Komponenten an der Energieaufnahme (*task.pmcenergyratio*) berechnet. Darüber hinaus wird für jede Kombination aus PMC und Spannungswandler die Leistungsaufnahme (*task.dcpower*), mittlere Leistungsaufnahme (*task.dcmeanpower*), die Energieaufnahme (*task.denergy*) und die Effizienz des Spannungswandlers (*task.dceff*) für den Task berechnet. Neben den berechneten Werten wird der für die jeweilige Systemkonfiguration optimale Spannungswandler *task.optdc* abgelegt.

5.9 Tastgrad

Der Tastgrad des Systems (*Duty Cycle*) bestimmt über die Ausführungsfrequenz der zuvor definierten Tasks das Verhältnis von aktiven zu inaktiven Modi und ist somit ein entscheidender Faktor für die Leistungsfähigkeit und die Leistungsaufnahme des Systems. Im Folgenden werden zunächst die Parameter des Tastgrads erläutert und die generierten Kennzahlen für die Auswertung des Entwurfsraums auf Ebene des Tastgrads beschrieben.

Parameter Mittels der in Tabelle 5.14 aufgeführten Parameter kann der Tastgrad des Systems definiert werden. Die Basis bildet hier die Ausführungsfrequenz *samplef*, die bestimmt, mit welcher Frequenz ein Task ausgeführt wird. Welcher der zuvor definierten Tasks ausgeführt wird, wird durch das Ausführungsverhältnis *taskratio* bestimmt. Um ein System für einen spezifischen Entwurfsraum simulieren zu können ist es möglich, für beide Parameter einen Wertebereich anzugeben.

Tabelle 5.14.: Parameter des Tastgrads

Beschreibung	Parameter	Einheit	Format	Definition
Ausführungsfrequenz der Tasks	<i>samplef</i>	Hz	[1xN _{TF}] double	Manuell
Ausführungsverhältnisse der Tasks	<i>taskratio</i>	-	[N _T xN _{TR}] double	Manuell

Der Parameter *taskratio* definiert die Ausführungsverhältnisse der N_T Tasks, wobei jede Spalte eine Konfiguration darstellt, so dass die Ausführungsverhältnisse aller Tasks in Summe 1 ergeben müssen (vgl. Gleichung 5.20). Für die t -te Spalte muss somit $\sum_{tr=1}^{N_{TR}} \text{taskratio}(t, tr) = 1$ gelten. Über die Definition von mehreren Spalten können verschiedene Konfiguration für die Simulation des System erstellt werden.

$$\text{taskratio}_{t, tr} = \begin{pmatrix} \text{taskratio}_{1,1} & \text{taskratio}_{1,2} & \cdots & \text{taskratio}_{1,N_{TR}} \\ \text{taskratio}_{2,1} & \text{taskratio}_{2,2} & \cdots & \text{taskratio}_{2,N_{TR}} \\ \vdots & \vdots & \ddots & \vdots \\ \text{taskratio}_{N_T,1} & \text{taskratio}_{N_T,2} & \cdots & \text{taskratio}_{N_T,N_{TR}} \end{pmatrix} \quad (5.20)$$

Mit der Größe des Entwurfsraums der Systemkonfigurationen $N_{EWR, SYS}$, der Anzahl der definierten Tasks N_T , der Anzahl der Ausführungsfrequenzen N_{TF} und der Anzahl der Ausführungsverhältnisse N_{TR} ergibt sich die folgende Größe des Entwurfsraums nach der Definition des Tastgrads:

$$N_{EWR} = N_{EWR, SYS} \cdot N_T \cdot N_{TF} \cdot N_{TR} \quad (5.21)$$

Auswertung Mittels des Tastgrads wird das Verhältnis von aktiven zu inaktiven Modi und somit das Systemverhalten bestimmt. Der Einfluss des Tastgrads auf die Leistungsaufnahme der PMC wird über die in Tabelle 5.15 aufgeführte Kennzahl *cycle.pmcmeanpower* abgebildet. Diese Kennzahl wird für jeden Tastgrad des Systems berechnet, die sich aus den Ausführungsfrequenzen N_{TF} und den Ausführungsverhältnissen N_{TR} ergeben. Ebenso wird die mittlere Leistungsaufnahme für jeden Spannungswandler berechnet (*cycle.dcmeanpower*).

Die mittlere Leistungsaufnahme der Spannungswandler, also die Leistungsentnahme aus der Batterie dient als Basis für die Berechnung der Laufzeit des Systems. Diese ist spezifisch für jede Kombination aus Spannungswandler und Batterie und wird entsprechend für den gesamten Entwurfsraum berechnet. Somit wird die Leistungsentnahme des Systems aus der Batterie für alle Kombinationen aus Systemkonfigurationen inklusive verschiedener Spannungswandler und Batterien, der definierten Tasks und deren Verhalten, abgebildet durch die Ausführungsfrequenzen N_{TF} und Ausführungsverhältnisse N_{TR} , bestimmt und so eine ganzheitliche Analyse des Entwurfsraums ermöglicht.

Tabelle 5.15.: Kennzahlen des Tastgrads

Beschreibung	Parameter	Einheit	Format
Tastgrad	<i>cycle</i>	-	$[N_{TF} \times N_{TR}]$ struct
- Mittlere Leistungsaufnahme PMC	<i>cycle.pmcmeanpower</i>	mW	$[1 \times N_{EWR, SYS}]$ double
- Mittlere Leistungsaufnahme Spannungswandler	<i>cycle.dcmeanpower</i>	mW	$[N_{DC} \times N_{EWR, SYS}]$ double
- Laufzeit des Systems	<i>cycle.tsys</i>	mW	$[N_{BAT} \times N_{DC} \times N_{EWR, SYS}]$ double

5.10 Zusammenfassung

Anhand der erstellten ganzheitlichen Entwurfsmethodik können Systeme und ihre Komponenten in jeder Phase des Entwurfsprozesses auf ihre Eignung für zuvor spezifizierte Anwendungsfälle hin beurteilt werden, so dass ein agiler und iterativer Entwurfsprozess ermöglicht wird. Insbesondere kann anhand der Modellierung und Simulation die Laufzeit des Systems bei der Verwendung von verschiedenen Komponenten abgeschätzt werden. Dadurch können zum einen Grenzfälle betrachtet werden, aber auch die ideale Systemkonfiguration für angestrebte Einsatzszenarien ermittelt werden.

Da der initiale Aufbau des Modells für ein zu entwerfendes System Zeit erfordert, muss sich die aufgewendete Zeit im Laufe der Entwicklung amortisieren. Dieses kann über die Wiederverwendung einzelner Teile des Modells erreicht werden, so dass zukünftige Entwicklungen entsprechend beschleunigt werden. So steht der anwendenden Person eine stetig wachsende Komponentendatenbank zur Verfügung, mit der verschiedene Komponenten des gleichen Typs miteinander verglichen werden können. Beispielsweise wird ein Vergleich verschiedener Funktransceiver oder Beschleunigungssensoren anhand der in den Komponentenmodellen hinterlegten Parameter ermöglicht. Somit können unmittelbar alle hinterlegten Komponenten eines Typs miteinander verglichen werden, ohne dass die relevanten Parameter wie die Leistungsaufnahme manuell aus den Datenblättern extrahiert werden müssen. Gegebenenfalls neu erhältliche Komponenten können entsprechend mit eingepflegt und somit in die Entscheidung mit einbezogen werden. Somit entsteht eine strukturierte und wiederverwendbare Informationsbasis aller bereits evaluierten Komponenten.

Zudem werden auch die Ergebnisse jeder Simulation gespeichert. In Tabelle 5.16 ist die Struktur der Kennzahlen dargestellt. So werden für jede der definierten Konfigurationen die zuvor beschriebenen Kennzahlen für den inaktiven Modus (*idle*, vgl. Tabelle 5.11), die aktiven Modi (*task*, vgl. Tabelle 5.13) und den Tastgrad (*cycle*, vgl. Tabelle 5.15) gespeichert.

Diese umfassen die Leistungsaufnahme für den inaktiven Modus und die Energieauf-

Tabelle 5.16.: Kennzahlen des Modells

Beschreibung	Parameter	Einheit	Format
Inaktiver Modus	<i>idle</i>	-	$[1 \times N_{\text{EWR,SYS}}]$ struct
Aktive Modi	<i>task</i>	-	$[N_T \times N_{\text{EWR,SYS}}]$ struct
Tastgrad	<i>cycle</i>	-	$[N_{\text{TF}} \times N_{\text{TR}}]$ struct

nahme der aktiven Modi des Systems der PMC zuzüglich der Effizienz der Spannungsversorgung für einen breiten Entwurfsraum. Aus diesen Daten wird die effizienteste Kombination aus PMC und Spannungswandler für jeden Modus des Systems bestimmt werden. Zudem kann analysiert werden, wie sich die Leistungsaufnahme im inaktiven Zustand (*idle.pmcpowerratio*) und die Energieaufnahme im aktiven Zustand (*task.pmcenergyratio*) auf die einzelnen Komponenten aufteilt. Über den Tastgrad, bestehend aus Ausführungsfrequenz und Ausführungsverhältnis, wird das Verhältnis von aktiven zu inaktiven Modi definiert und für jeden Arbeitspunkt des Systems die mittlere Leistungsaufnahme bestimmt. Abschließend wird für jeden Tastgrad die Kombination aus PMC, Spannungswandler und Energiespeicher mit der höchsten Laufzeit bestimmt. Es wird somit eine Entwurfsraumexploration ermöglicht, die für verschiedene Leistungsfähigkeiten (Task, Tastgrad) die Leistungsaufnahme und die Laufzeit des Systems ermittelt, so dass ein bedarfsgerechter Entwurf ermöglicht wird, welcher die Effizienz des Systems maximiert.

6 Modellgestützter Entwurf eines Sensor-knotens

Nachdem im vorherigen Kapitel der Aufbau des Modells beschrieben und die einzelnen Aspekte beispielhaft verdeutlicht wurden, folgt in diesem Kapitel die Beschreibung des modellgestützten Entwurfs eines drahtlosen Sensorknotens. Im Entwurfsablauf erfolgt zunächst die Bedarfsanalyse. Anschließend wird die BauteilAuswahl der aktiven Komponenten (PMC) beschrieben und die Spannungsversorgung dimensioniert. Basierend darauf erfolgt die Systemkonfiguration sowie die Definition der Tasks und des Tastgrads des Systems. Erste Ansätze dieser Umsetzung konnten bereits auf einer Konferenz vorgestellt und publiziert werden [251].

6.1 Bedarfsanalyse

Die Bedarfsanalyse basiert auf den Anforderungen an einen drahtlosen Sensorknoten, die im Rahmen eines Forschungsprojekts systematisch umgesetzt wurden. Das Ziel war hier die Entwicklung eines multimodalen Umgebungssensorknotens für die Gebäudeautomation. Der Sensorknoten sollte batteriebetrieben sein und erfasste Daten mittels Funkkommunikation an eine Basisstation übertragen. Im Rahmen des Projektes sollte eine Auswahl geeigneter Systemkomponenten für den Sensorknoten erfolgen, sowie eine erste prototypische Realisierung umgesetzt werden. Für die Auswahl geeigneter Systemkomponenten war eine prototypische Abschätzung des Energiebedarfs möglicher Systeme für realistische Anwendungsszenarien notwendig. Insbesondere die Laufzeit des Sensorknotens entscheidet im praktischen Einsatz über die ökonomisch sinnvolle Verwendbarkeit von verteilten Sensorknoten, da diese ohne Wartungsaufwand möglichst lange betrieben werden sollen. Weitere Zielgrößen waren eine kompakte Bauform und geringe Kosten des Systems. Die priorisierte Zielgröße für die Bewertungsfunktionen der Entwurfsraumexploration ist hier jedoch die Laufzeit des Systems.

Laut Vorgabe sollte der Sensorknoten über Sensoren für typische Umgebungsmessgrößen wie Helligkeit, Temperatur und Luftfeuchtigkeit verfügen. Die Informationsverarbeitung des Sensorknotens sollte in der Lage sein, die Sensoren periodisch abzufragen, die Messwerte auf Änderung zu überprüfen und diese gegebenenfalls drahtlos an eine Basisstation zu senden.

6.2 BauteilAuswahl PMC

Im Folgenden wird die BauteilAuswahl der aktiven Komponenten (PMC) beschrieben. Diese umfassen die Komponenten zur Datenverarbeitung und Funkübertragung sowie die verschiedenen Sensoren zur Erfassung der Umgebung. Aus den beschriebenen Rahmenbedingungen wie der geringen Komplexität der Datenverarbeitung, der geforderten langen Laufzeit, der geringen Größe sowie den geringen Kosten, ergab sich vorab die Entwurfsentscheidung statt einem Mikrocontroller und einem Transceiver ein SoC zu

verwenden, welches diese Komponenten in einem Gehäuse vereint.

Im Folgenden wird für jede Komponenten aufgeführt, wie die für die Modellbildung benötigten Parameter ermittelt wurden und welche Parameter vom Modell generiert werden. Darauf folgend wird die Bewertung der Komponenten beschrieben. Das Modell jeder Komponente wird als MATLAB *mat*-Datei gespeichert. Somit wird eine Komponentendatenbank aufgebaut.

6.2.1 Komponente SoC

Laut Bedarfsanalyse war ein SoC mit Transceiver und integriertem Mikrocontroller vorgegeben. Zwecks Kompatibilität mit anderen Systemen sollte nach Vorgabe im Forschungsprojekt zur Datenübertragung an die Basisstation der BLE Funkstandard genutzt werden. Zum Zeitpunkt der Komponentenauswahl erfüllten der *Nordic Semiconductor nRF51822* [139] und der *Bluegiga BLE113* [23] diese Anforderungen. Mit Hilfe des modellbasierten Entwurfs erfolgte ein Vergleich dieser beiden Komponenten.

In einem ersten Schritt wurden die benötigten Daten gesammelt. Diesen umfassen die Parameter des grundlegenden Komponentenmodells wie Versorgungsspannung, Stromaufnahme der Modi, Dauer und Stromaufnahme der Transitionen sowie die Stromaufnahme der Schnittstellen (vgl. Tabelle 5.1). Zudem werden die Daten für die spezifischen Komponentenmodelle für Mikrocontroller und Transceiver benötigt. Für den Mikrocontroller ist dieses die Taktrate (vgl. Tabelle 5.2). Für den Transceiver wird, neben der Taktrate, die maximale Nutzlast der drahtlosen Schnittstelle benötigt (vgl. Tabelle 5.4). Die in den SoC integrierten Komponenten Mikrocontroller und Transceiver werden im Modell zunächst als eine Komponente abgebildet.

Ermittelte Parameter Die Parameter Versorgungsspannung und Stromaufnahme der Modi konnten sowohl für den *Nordic Semiconductor nRF51822* [139] als auch für den *Bluegiga BLE113* [23] den Datenblättern der Hersteller entnommen und in das Modell eingefügt werden. Eine Übersicht dieser Parameter findet sich in Tabelle 6.1.

Für den Modus 2 (Senden) müssen zudem die maximale Nutzlast (*mode.maxpayload*) und die Datenrate der drahtlosen Übertragung (*mode.datarate*) angegeben werden. Die maximale Nutzlast ergibt sich aus der Spezifikation des BLE-Protokolls [82]. Die Datenrate kann bei beiden Transceivern konfiguriert werden [23; 139] und wird hier hinsichtlich der Vergleichbarkeit auf 250 kbps gesetzt.

Die Parameter der Transitionen konnten für den *Nordic Semiconductor nRF51822* SoC dem Datenblatt des Herstellers [139] entnommen werden (Gleichung 6.1 und Gleichung 6.2). Zu beachten ist jedoch, dass einige Parameter auf Basis der im Datenblatt dokumentierten Parameter ermittelt werden müssen. So finden sich beispielsweise die Dauer der Transition zwischen den Modi 5 (*OFF*) und 4 (*ON*) und die Dauer der Transitionen zwischen den Modi 4 (*ON*) und 2 (*TX*) im Datenblatt. Für die Dauer der Transition zwischen dem Modus 5 (*OFF*) und dem Modus 2 (*TX*) muss entsprechend die Dauer der Transitionen zwischen den Modi 5 (*OFF*) und 4 (*ON*) und den Modi 4 (*ON*) und 2

Tabelle 6.1.: Definierte Parameter der SoC: Modi

Parameter	SoC 1	SoC 2	Einheit
<i>description</i>	Bluegiga BLE113	Nordic nRF51822	-
<i>voltage.min</i>	2,0	1,8	V
<i>voltage.typ</i>	3,0	3,0	V
<i>voltage.max</i>	3,6	3,6	V
<i>mode(1).description</i>	CPU from flash	CPU from flash	-
<i>mode(1).current</i>	6,7	4,1	mA
<i>mode(1).f</i>	32	16	MHz
<i>mode(1).active</i>	1	1	-
<i>mode(2).description</i>	TX 0 dB	TX 0 dB	-
<i>mode(2).current</i>	18,2	10,5	mA
<i>mode(2).maxpayload</i>	20	20	Byte
<i>mode(2).datarate</i>	250	250	kbps
<i>mode(2).active</i>	1	1	-
<i>mode(3).description</i>	RX normal mode	RX 250 kbps	-
<i>mode(3).current</i>	17,9	12,6	mA
<i>mode(3).active</i>	1	1	-
<i>mode(4).description</i>	Power Mode 1	ON	-
<i>mode(4).current</i>	0,27	0,0038	mA
<i>mode(4).active</i>	0	0	-
<i>mode(5).description</i>	Power Mode 2	OFF	-
<i>mode(5).current</i>	0,001	0,0006	mA
<i>mode(5).active</i>	0	0	-
<i>mode(6).description</i>	Power Mode 3	-	-
<i>mode(6).current</i>	0,0005	-	mA
<i>mode(6).active</i>	0	-	-

(TX) addiert werden. Für die Stromaufnahme muss der Mittelwert gebildet werden. Diese Berechnungen werden vom Modell selbst durchgeführt, so dass nur die grundlegenden Parameter angegeben werden müssen. Zu der Dauer und Stromaufnahme der Transitionen werden im Datenblatt des *Bluegiga BLE113* [23] keine Angaben gemacht. Da der *BLE113* auf einem *Texas Instruments CC2541* SoC [197] Transceiver basiert, konnten die Transitionszeiten dem *CC2541* Datenblatt [197] entnommen werden (Gleichung 6.3 und Gleichung 6.4). Angaben zur Stromaufnahme der Transitionen waren weder im Datenblatt des *BLE113* noch im Datenblatt des *CC2541* enthalten. Daher wurde für die Stromaufnahme der Transition zwischen zwei Modi jeweils die Stromaufnahme des Modus mit der höheren Stromaufnahme angenommen. Auch hier ist zu beachten, dass sich einige

Transitionen aus der Summe bzw. dem Mittelwert anderer Transitionen ergeben. Die Hauptachsen der Transitionen, also bei Verbleib in einem Zustand, weisen die Stromaufnahme des jeweiligen Modus auf.

$$nrf51822.transitions.current = \begin{pmatrix} 4,1 & 7,0 & 8,7 & 4,1 & 4,1 \\ 10,5 & 10,5 & 8,7 & 10,5 & 10,5 \\ 12,6 & 7,0 & 12,6 & 12,6 & 12,6 \\ 0,6 & 7,0 & 8,7 & 0,0038 & 0,0038 \\ 0,46 & 6,55 & 8,13 & 0,4 & 0,0006 \end{pmatrix} \text{ mA} \quad (6.1)$$

$$nrf51822.transitions.time = \begin{pmatrix} 0 & 130 & 130 & 0 & 0 \\ 0 & 0 & 130 & 0 & 0 \\ 0 & 130 & 0 & 0 & 0 \\ 3,8 & 130 & 130 & 0 & 0 \\ 13,4 & 139,6 & 139,6 & 9,6 & 0 \end{pmatrix} \mu\text{s} \quad (6.2)$$

$$ble113.transitions.current = \begin{pmatrix} 6,7 & 18,2 & 17,9 & 6,7 & 6,7 & 6,7 \\ 18,2 & 18,2 & 18,2 & 18,2 & 18,2 & 18,2 \\ 17,9 & 18,2 & 17,9 & 17,9 & 17,9 & 17,9 \\ 6,7 & 18,2 & 17,9 & 0,27 & 0,27 & 0,27 \\ 6,7 & 18,2 & 17,9 & 0,27 & 0,001 & 0,001 \\ 6,7 & 18,2 & 17,9 & 0,27 & 0,001 & 0,0005 \end{pmatrix} \text{ mA} \quad (6.3)$$

$$ble113.transitions.time = \begin{pmatrix} 0 & 180 & 180 & 0 & 0 & 0 \\ 0 & 0 & 150 & 0 & 0 & 0 \\ 0 & 150 & 0 & 0 & 0 & 0 \\ 4 & 184 & 184 & 0 & 0 & 0 \\ 120 & 300 & 300 & 0 & 0 & 0 \\ 120 & 300 & 300 & 0 & 0 & 0 \end{pmatrix} \mu\text{s} \quad (6.4)$$

Die Parameter der Schnittstellen konnten bei beiden SoC den Datenblättern der Hersteller [23; 139] entnommen werden. Diese umfassen die Stromaufnahmen und die Datenraten der verfügbaren Schnittstellen. Zudem wird die Fläche der Komponenten angegeben. Eine Übersicht der Parameter findet sich in Tabelle 6.2. Für den *Bluegiga BLE113* [23] waren die Stromaufnahmen der Schnittstellen nicht verfügbar.

Generierte Parameter Auf Basis der aus den Datenblättern ermittelten Daten werden vom Modell weitere Parameter für die Komponenten generiert. Für die Komponente Mikrocontroller ist dieses die Nutzenzeit T_{BE} (vgl. Tabelle 5.2). Für die Komponente Transceiver umfassen die generierten Parameter die möglichen Datenbreiten der Übertragung und die Dauer der möglichen Übertragungen (vgl. Tabelle 5.4).

Tabelle 6.2.: Ermittelte Parameter der SoC: Schnittstellen und Fläche

Parameter	SoC 1	SoC 2	Einheit
<i>description</i>	Bluegiga BLE113	Nordic nRF51822	-
<i>interface(1).no</i>	1	1	-
<i>interface(1).description</i>	UART	UART	-
<i>interface(1).current</i>	n.v.	0,22	mA
<i>interface(1).datarate</i>	115,2	115,2	kbps
<i>interface(2).no</i>	2	2	-
<i>interface(2).description</i>	SPI	SPI	-
<i>interface(2).current</i>	n.v.	0,18	mA
<i>interface(2).datarate</i>	n.v.	125	kbps
<i>interface(3).no</i>	3	3	-
<i>interface(3).description</i>	I2C	I2C	-
<i>interface(3).current</i>	n.v.	0,4	mA
<i>interface(3).datarate</i>	400	400	kbps
<i>area</i>	144,11	36	mm ²

Wie in Unterabschnitt 5.5.3 beschrieben, wird die Datenbreite der drahtlosen Übertragung (*mode.datawidth*) mittels einer hinterlegten Funktion für den Protokolloverhead für den Bereich von 0 bis *maxpayload* Byte über eine lineare Funktion der Form $a \cdot D_I + b$ generiert und in einem LUT abgelegt. Da beide Transceiver nach dem BLE-Standard arbeiten, ist diese Funktion für beide SoC gleich und kann der Literatur entnommen werden [82]:

$$\text{mode.datawidth} = 56 \text{ Bit} + D_I \cdot 8 \frac{\text{Bit}}{\text{Byte}} \quad (6.5)$$

So werden jedem zu übertragenden Datenpaket (Nutzlast D_i in Byte) die Daten des Protokolls hinzugefügt, welche bei BLE aus einer Präambel (8 Bit), der Zugangsadresse (32 Bit), dem Header (8 Bit) und der Länge des Paketes (8 Bit) bestehen. Somit beträgt die zu übertragende Datenbreite zwischen 64 Bit (für 1 Byte Nutzdaten) und 216 Bit (für die in *mode(2).maxpayload* definierten maximal möglichen 20 Byte Nutzdaten).

Die Dauer der drahtlosen Datenübertragung (*mode.time*) wird aus der zu übertragenden Datenbreite (*mode.datawidth*) und der Datenrate der drahtlosen Übertragung (*mode.datarate*) berechnet:

$$\text{mode.time} = \frac{\text{mode.datawidth}}{\text{mode.datarate}} \quad (6.6)$$

Da beide Transceiver laut *mode(2).datarate* eine Datenrate von 250 kbps aufweisen ergibt sich eine Dauer der drahtlosen Übertragung zwischen $256 \mu\text{s}$ (für 64 Bit Datenbreite) und $864 \mu\text{s}$ (für die maximale Datenbreite von 216 Bit Nutzdaten).

Für die Komponente Mikrocontroller wird die Nutzenzeit T_{BE} (vgl. Unterabschnitt 5.5.1) generiert. Diese beschreibt die minimale Zeit, die die Komponente in einem inaktiven Zustand betrieben werden muss, um die Leistungsaufnahme der Transitionen zu kompensieren. Die Nutzendauer wird für jeden möglichen Wechsel zwischen einem aktiven und einem inaktiven Modus der Komponente berechnet und in einem LUT hinterlegt.

Die vom Modell berechnete Nutzenzeit für den *Bluegiga BLE113* ist in Gleichung 6.7 dargestellt, wobei alle Angaben in μs sind. Die Modi 1, 2 und 3 sind aktiv, die Modi 4, 5 und 6 sind inaktive Modi (vgl. Tabelle 6.1). Der Wert $tbe_{1,4}$ mit $4000\ \mu\text{s}$ entspricht der Nutzenzeit des aktiven Modus 1 (*CPU from flash*) gegenüber dem inaktiven Modus 4 (*ON*). Der Wechsel vom Modus 1 in den Modus 4 lohnt sich aus energetischer Sicht nur, wenn die inaktive Phase in Modus 4 mindestens $4000\ \mu\text{s}$ dauert. Falls diese Phase kürzer ist, ist es energieeffizienter die Komponente im Modus 1 zu belassen. Andernfalls überwiegen die energetischen Kosten von Transition und inaktivem Modus gegenüber den energetischen Kosten des aktiven Modus. Für den Fall, dass die Leistungsaufnahme der Transitionen geringer ist als die Leistungsaufnahme im aktiven Modus ist es immer energieeffizienter in den inaktiven Modus zu wechseln.

$$\begin{aligned} ble113.tbe &= \begin{pmatrix} tbe_{1,1} & tbe_{1,2} & tbe_{1,3} & tbe_{1,4} & tbe_{1,5} & tbe_{1,6} \\ tbe_{2,1} & tbe_{2,2} & tbe_{2,3} & tbe_{2,4} & tbe_{2,5} & tbe_{2,6} \\ tbe_{3,1} & tbe_{3,2} & tbe_{3,3} & tbe_{3,4} & tbe_{3,5} & tbe_{3,6} \end{pmatrix} & (6.7) \\ &= \begin{pmatrix} - & - & - & 4.000 & 120.000 & 120.000 \\ - & - & - & 184.000 & 300.000 & 300.000 \\ - & - & - & 184.000 & 300.000 & 300.000 \end{pmatrix} \end{aligned}$$

Analog dazu ist in Gleichung 6.8 die vom Modell berechnete Nutzendauer in μs für den *Nordic Semiconductor nRF51822* aufgeführt. Dieser verfügt über 5 verschiedene Modi, wobei die Modi 1, 2 und 3 aktiv und die Modi 4 und 5 inaktiv sind (vgl. Tabelle 6.1).

$$\begin{aligned} nRF51822.tbe &= \begin{pmatrix} tbe_{1,1} & tbe_{1,2} & tbe_{1,3} & tbe_{1,4} & tbe_{1,5} \\ tbe_{2,1} & tbe_{2,2} & tbe_{2,3} & tbe_{2,4} & tbe_{2,5} \\ tbe_{3,1} & tbe_{3,2} & tbe_{3,3} & tbe_{3,4} & tbe_{3,5} \end{pmatrix} & (6.8) \\ &= \begin{pmatrix} - & - & - & 3.800 & 13.400 \\ - & - & - & 130.000 & 139.600 \\ - & - & - & 130.000 & 139.600 \end{pmatrix} \end{aligned}$$

Bewertung der Komponenten Im praktischen Einsatz wird die Leistungsaufnahme des SoC bestimmt durch die Leistungsaufnahme in den aktiven und inaktiven Modi und den Tastgrad des Systems, also dem Anteil der aktiven Modi gegenüber den inaktiven Modi. Jedoch hängt der Tastgrad des Systems von der Anwendung ab und ist noch nicht bekannt. Neben den Leistungsaufnahmen wird daher die Nutzenzeit T_{BE} als Kennzahl verwendet.

Diese wird mit der Leistungsaufnahme des jeweiligen inaktiven Modus multipliziert und bildet als $T_{BE} \cdot P_I$ -Produkt das Bewertungsmaß für die inaktiven Modi der Komponente (vgl. TPP [17]). Zur Auswahl der Komponente werden in Abbildung 6.1 für beide SoC die $T_{BE} \cdot P_I$ -Produkte der inaktiven Modi über den Leistungsaufnahmen der aktiven Modi CPU und TX aufgetragen. Für den *nRF51822* werden die inaktiven Modi 4 und 5 und für den *BLE113* die inaktiven Modi 5 und 6 dargestellt (vgl. Tabelle 6.1). Der inaktive Modus 4 des *BLE113* wird nicht dargestellt, da dieser weit abseits der übrigen Werte liegt.

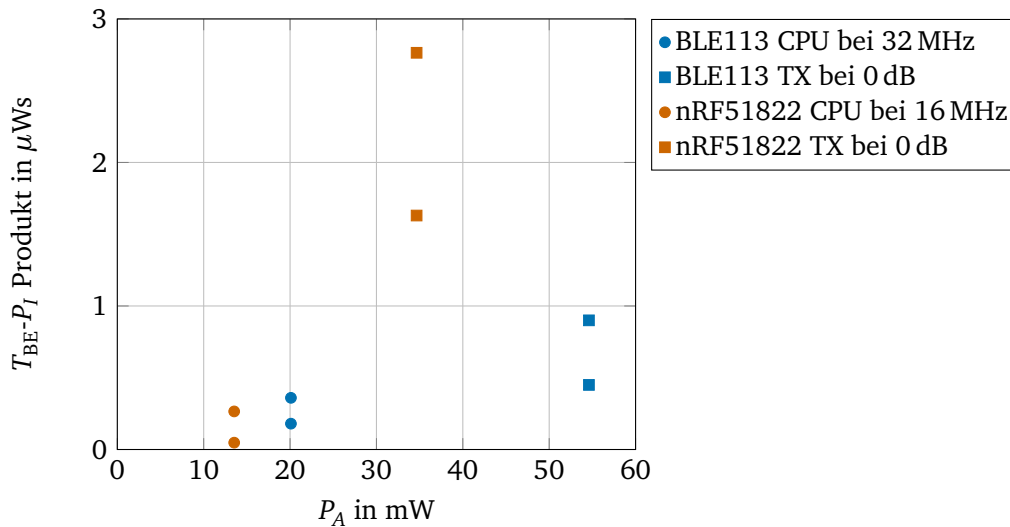


Abbildung 6.1.: Vergleich der SoC *nRF51822* und *BLE113*

In der grafischen Darstellung wird deutlich, dass für den Rechenbetrieb (Aktiver Modus CPU) des *nRF51822* sowohl das $T_{BE} \cdot P_I$ -Produkt als auch die Leistungsaufnahme im aktiven Modus P_A im Vergleich zum *BLE113* geringer sind. Somit ist der *nRF51822* im Rechenbetrieb effizienter. Im Sendebetrieb (Aktiver Modus TX) weist der *nRF51822* ein höheres $T_{BE} \cdot P_I$ -Produkt auf. Jedoch ist, im Vergleich zum *BLE113*, die Leistungsaufnahme beim Senden wesentlich geringer. Bei häufigem Senden von Daten wäre der *nRF51822* somit effizienter. Bei einem sehr geringen Tastgrad des Systems könnte der *BLE113* jedoch effizienter sein, da in diesem Fall der Betrieb im inaktiven Modus überwiegt. Eine Auswahl der Komponente SoC kann somit nicht auf der Komponentenebene, sondern muss zu einem späteren Zeitpunkt unter Berücksichtigung des Tastgrads des Systems erfolgen.

6.2.2 Komponente Umgebungslichtsensor

Entsprechend der Zielvorgabe des multimodalen Sensorknotens für die Gebäudeautomation muss dieser in der Lage sein, die Helligkeit der Umgebung zu erfassen. Zu diesem Zweck wurden die am Markt erhältlichen Umgebungslichtsensoren *Texas Instruments OPT3001* [201] und *Maxim Integrated MAX44009* [126] näher betrachtet.

Ermittelte Parameter Die für die Modellbildung benötigten Parameter konnten für beide Sensoren den Datenblättern der Hersteller [126; 201] entnommen werden und sind in Tabelle 6.4 aufgeführt. Der *OPT3001* verfügt über einen Messmodus für dunkle Umgebungen (*Dark*) und einen Messmodus für die gesamte Bandbreite (*Full-scale lux*). Dementsprechend gibt es auch zwei verschiedene inaktive Modi. Der *MAX44009* hingegen verfügt nur über einen aktiven Modus. Zudem finden sich im Datenblatt keine Informationen zu einem verfügbaren inaktiven Modus. Beide Sensoren unterstützen Messungen mit einer Dauer von 100 ms als Standardeinstellung und Messungen mit einer Dauer von 800 ms für geringe Lichtintensitäten. Im Sinne der Vergleichbarkeit wird für beide Sensoren eine Messdauer von 100 ms angenommen. Eine Kennzahl für die Qualität der Messungen (*mode.acc*) kann nicht angegeben werden, da sich in den Datenblättern keine vergleichbaren Angaben finden. Bezüglich der Dauer und Leistungsaufnahme der Transitionen finden sich für den *OPT3001* keine Angaben, so dass diese mit 0 angenommen werden. Da der *MAX44009* nur über einen Modus verfügt, existieren dementsprechend keine Transitionen. Beide Sensoren verfügen über eine I2C-Schnittstelle, jedoch werden keine Angaben zur Stromaufnahme gemacht.

Generierte Parameter Das Modell generiert für jeden aktiven Modus der Sensoren aus der Dauer der Messung, der Stromaufnahme des Modus und der Versorgungsspannung der Komponente die Energieaufnahme für eine Messung (vgl. Tabelle 6.3).

Tabelle 6.3.: *Generierte Parameter der Umgebungslichtsensoren*

Parameter	Sensor 1	Sensor 2	Einheit
<i>description</i>	OPT3001	MAX44009	-
<i>mode(1).description</i>	Active Full-scale lux	Active	-
<i>mode(1).emeas</i>	1221	216	μWs
<i>mode(2).description</i>	Active Dark	-	-
<i>mode(2).emeas</i>	594	-	μWs

Bewertung der Komponenten Zur Bewertung der Komponente wird die vom Modell generierte Energieaufnahme pro Messung über die Leistungsaufnahme im jeweiligen inaktiven Modus P_I aufgetragen. Da sich im Datenblatt des *MAX44009* keine Informationen zu einem verfügbaren inaktivem Modus finden, wird für die Leistungsaufnahme im inaktiven Modus auch die Leistungsaufnahme des aktiven Modus angenommen.

In der grafischen Darstellung zeigt sich, dass die Energieaufnahme pro Messung für den *OPT3001* für beide Messmodi höher ist als für den *MAX44009*. Jedoch ist für den *MAX44009* die Leistungsaufnahme im inaktiven Modus deutlich höher. Da keiner der Sensoren sowohl

Tabelle 6.4.: Ermittelte Parameter der Umgebungslichtsensoren

Parameter	Sensor 1	Sensor 2	Einheit
<i>description</i>	OPT3001	MAX44009	-
<i>voltage.min</i>	1,6	1,7	V
<i>voltage.typ</i>	3,3	1,8	V
<i>voltage.max</i>	3,6	3,6	V
<i>mode(1).description</i>	Active Full-scale lux	Active	-
<i>mode(1).current</i>	0,0037	0,0012	mA
<i>mode(1).datawidth</i>	16	16	Bit
<i>mode(1).time</i>	100.000	100.000	μs
<i>mode(1).acc</i>	-	-	-
<i>mode(1).active</i>	1	1	-
<i>mode(2).description</i>	Active Dark	-	-
<i>mode(2).current</i>	0,0018	-	mA
<i>mode(2).datawidth</i>	16	-	Bit
<i>mode(2).time</i>	100.000	-	μs
<i>mode(2).acc</i>	-	-	-
<i>mode(2).active</i>	1	-	-
<i>mode(3).description</i>	Shutdown Full-scale lux	-	-
<i>mode(3).current</i>	0,0004	-	mA
<i>mode(3).active</i>	0	-	-
<i>mode(4).description</i>	Shutdown Dark	-	-
<i>mode(4).current</i>	0,0003	-	mA
<i>mode(4).active</i>	0	-	-
<i>transitions.current</i>	$\begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$	-	mA
<i>transitions.time</i>	$\begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$	-	μs
<i>interface(1).no</i>	3	3	-
<i>interface(1).description</i>	I2C	I2C	-
<i>interface(1).current</i>	n.v.	n.v.	mA
<i>interface(1).datarate</i>	400	400	kbps
<i>area</i>	4	4	mm ²

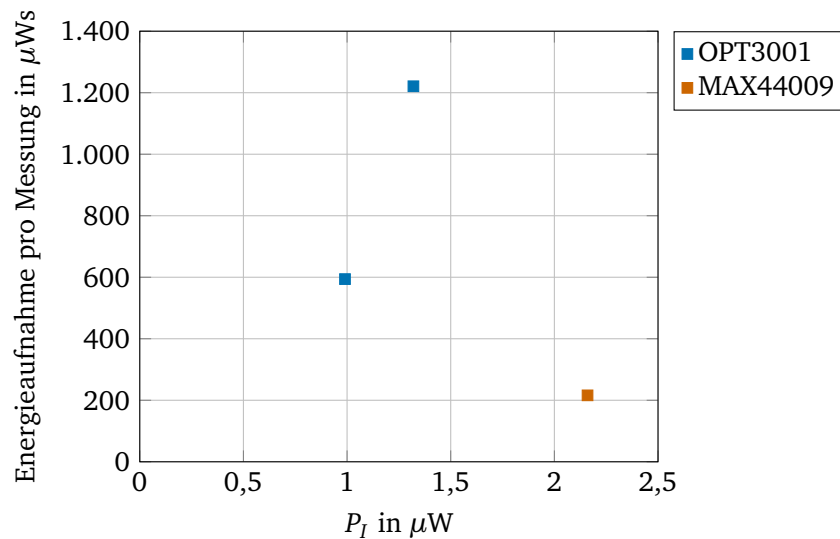


Abbildung 6.2.: Vergleich der Umgebungslichtsensoren *OPT3001* und *MAX44009*

die kleinste Energieaufnahme pro Messung, als auch die kleinste Leistungsaufnahme im inaktiven Modus aufweist, kann die Auswahl der Komponente Umgebungslichtsensor somit nicht auf der Komponentenebene, sondern muss zu einem späteren Zeitpunkt unter Berücksichtigung des Tastgrads des Systems erfolgen.

6.2.3 Komponente Umgebungssensor

Entsprechend der Zielvorgabe des multimodalen Sensorknotens für die Gebäudeautomation soll das System über Sensoren zur Erfassung des Raumklimas verfügen. Entsprechende Parameter sind, neben der Helligkeit, die Umgebungstemperatur, die Luftfeuchtigkeit und der Luftdruck (vgl. Abschnitt 2.3). Zu diesem Zweck wurden die am Markt erhältlichen Umgebungssensoren *Bosch Sensortec BME280* [28], *Texas Instruments HDC1000* [199], *Sensirion SHT20* [182] und *Measurement Specialties MS8607* [127] verglichen.

Ermittelte Parameter Alle Typen sind in der Lage, die Umgebungstemperatur und die Luftfeuchtigkeit zu messen. Die Komponenten *BME280* und *MS8607* können darüber hinaus auch den Luftdruck erfassen. Jede mögliche Messung wird im Modell als ein aktiver Modus abgebildet (vgl. Tabelle 6.5 und Tabelle 6.6). Bei dem *BME280* ist es zudem möglich, alle drei Parameter in einem Messzyklus zu erfassen (Modus HPT). Als Kennzahl für die Qualität der Messung wird für jeden aktiven Modus die in den Datenblättern angegebene *Accuracy* als Parameter in das Modell aufgenommen (*mode.acc*). Für den *BME280*, den *HDC1000* und den *SHT20* konnten die Parameter Stromaufnahme, Datenbreite und Dauer

Tabelle 6.5.: Ermittelte Parameter der Umgebungssensoren: Versorgungsspannung und Modi

Parameter	Sensor 1	Sensor 2	Sensor 3	Sensor 4	Einheit
<i>description</i>	BME280	HDC1000	SHT20	MS8607	-
<i>voltage.min</i>	1,71	2,7	2,1	1,5	V
<i>voltage.typ</i>	1,8	3,0	3,0	3,0	V
<i>voltage.am</i>	3,6	5,5	3,6	3,6	V
<i>mode(1).description</i>	Humidity	Humidity	Humidity	Humidity	-
<i>mode(1).current</i>	0,340	0,180	0,300	n.v.	mA
<i>mode(1).datawidth</i>	16	14	12	16	Bit
<i>mode(1).time</i>	3.500	6.500	22.000	n.v.	μ s
<i>mode(1).acc</i>	± 3	± 3	± 3	± 5	%RH
<i>mode(1).active</i>	1	1	1	1	-
<i>mode(2).description</i>	Temperature	Temperature	Temperature	Temperature	-
<i>mode(2).current</i>	0,350	0,155	0,300	n.v.	mA
<i>mode(2).datawidth</i>	20	14	14	24	Bit
<i>mode(2).time</i>	3.000	6.350	66.000	n.v.	μ s
<i>mode(2).acc</i>	± 1	$\pm 0,2$	$\pm 0,3$	± 2	$^{\circ}$ C
<i>mode(2).active</i>	1	1	1	1	-
<i>mode(3).description</i>	Pressure	Sleep	Sleep	Pressure	-
<i>mode(3).current</i>	0,714	0,00011	0,00015	n.v.	mA
<i>mode(3).datawidth</i>	20	-	-	24	Bit
<i>mode(3).time</i>	3.500	-	-	n.v.	μ s
<i>mode(3).acc</i>	± 1	-	-	± 4	hPa
<i>mode(3).active</i>	1	0	0	1	-
<i>mode(4).description</i>	HPT	-	-	Standby	-
<i>mode(4).current</i>	0,4425	-	-	0,00003	mA
<i>mode(4).datawidth</i>	56	-	-	-	Bit
<i>mode(4).time</i>	8.000	-	-	-	μ s
<i>mode(4).active</i>	1	-	-	0	-
<i>mode(5).description</i>	Standby	-	-	-	-
<i>mode(5).current</i>	0,0002	-	-	-	mA
<i>mode(5).active</i>	0	-	-	-	-
<i>mode(6).description</i>	Sleep	-	-	-	-
<i>mode(6).current</i>	0,0001	-	-	-	mA
<i>mode(6).active</i>	0	-	-	-	-

6. Modellgestützter Entwurf eines Sensorknotens

der Modi den Datenblättern entnommen werden [28; 182; 199]. Für den *MS8607* findet sich im Datenblatt [127] jedoch nur die durchschnittliche Stromaufnahme bei einem Tastgrad von 1 Hz, die Konvertierungszeit der ADC und die maximale Stromaufnahme in jedem Messmodus. Aus diesen Angaben ist es nicht möglich, die für das Modell benötigte Stromaufnahme und Dauer der Messung zu rekonstruieren. Alle Sensoren verfügen über mindestens einen inaktiven Modus und eine I2C Schnittstelle. Der *BME280* verfügt zusätzlich über eine SPI Schnittstelle. In keinem der Datenblätter finden sich Angaben zur Stromaufnahme der Schnittstellen und der Dauer und Stromaufnahme der Transitionen. Somit werden diese Angaben im Modell auf 0 gesetzt.

Tabelle 6.6.: Ermittelte Parameter der Umgebungssensoren: Schnittstellen und Fläche

Parameter	Sensor 1	Sensor 2	Sensor 3	Sensor 4	Einheit
<i>description</i>	BME280	HDC1000	SHT20	MS8607	-
<i>interface(1).no</i>	3	3	3	3	-
<i>interface(1).description</i>	I2C	I2C	I2C	I2C	-
<i>interface(1).current</i>	n.v.	n.v.	n.v.	n.v.	mA
<i>interface(1).datarate</i>	400	400	400	400	kbps
<i>interface(2).no</i>	2	-	-	-	-
<i>interface(2).description</i>	SPI	-	-	-	-
<i>interface(2).current</i>	n.v.	-	-	-	mA
<i>interface(2).datarate</i>	10.000	-	-	-	kbps
<i>area</i>	6,25	3,2	9	15	mm ²

Generierte Parameter Das Modell generiert für jeden aktiven Modus der Sensoren *BME280*, *HDC1000* und *SHT20* aus der Dauer der Messung, der Stromaufnahme des Modus und der Versorgungsspannung der Komponente die Energieaufnahme für eine Messung (vgl. Tabelle 6.7). Da für den *MS8607* die Dauer der Messung und die Stromaufnahme des Modus fehlen, kann entsprechend nicht die Energieaufnahme für eine Messung berechnet werden.

Bewertung der Komponenten Da für den *MS8607* keine Daten vorliegen kann dieser nicht in den modellbasierten Auswahlprozess einbezogen werden. Bei der Betrachtung der vom Modell berechneten Energieaufnahme in Tabelle 6.7 wird zudem ersichtlich, dass die Energieaufnahme des *SHT20* bei beiden Messgrößen weit über denen der anderen Sensoren liegt und somit nicht konkurrenzfähig ist. Es werden daher nur *BME280* und *HDC1000* näher betrachtet. Zur Auswahl der Komponente wird für beide Typen die vom Modell generierte Energieaufnahme pro Messung über die Leistungsaufnahme im jeweiligen inaktiven Modus P_I aufgetragen. In der grafischen Darstellung (vgl. Abbildung 6.3) zeigt

Tabelle 6.7.: Generierte Parameter der Umgebungssensoren

Parameter	Sensor 1	Sensor 2	Sensor 3	Sensor 4	Einheit
<i>description</i>	BME280	HDC1000	SHT20	MS8607	-
<i>mode(1).description</i>	Humidity	Humidity	Humidity	Humidity	-
<i>mode(1).emeas</i>	2.142	3.510	19.800	n.v.	μ Ws
<i>mode(2).description</i>	Temperature	Temperature	Temperature	Temperature	-
<i>mode(2).emeas</i>	1.890	2.953	59.400	n.v.	μ Ws
<i>mode(3).description</i>	Pressure	Sleep	Sleep	Pressure	-
<i>mode(3).emeas</i>	4.498	-	-	n.v.	μ Ws
<i>mode(4).description</i>	HPT	-	-	Standby	-
<i>mode(4).emeas</i>	6.372	-	-	-	μ Ws

sich, dass der *BME280* sowohl für die Messung der Umgebungstemperatur, als auch für die Messung der Luftfeuchtigkeit eine geringere Energieaufnahme aufweist. Zudem verfügt der *BME280* über einen inaktiven Modus mit geringerer Leistungsaufnahme.

Als zweites Auswahlkriterium wird die Energieaufnahme pro Messung über der Präzision der Messung aufgetragen (vgl. Abbildung 6.4). Hier zeigt sich, dass der *HDC1000* für beide Messungen, bei gleicher bzw. besserer Präzision, über eine höhere Energieaufnahme pro Messung verfügt. Der *BME280* verfügt hingegen über eine geringere Leistungsaufnahme im inaktiven Modus. Somit ist keiner der beiden Sensoren in allen Aspekten pareto-optimal und die Auswahl der Komponente muss zu einem späteren Zeitpunkt unter Berücksichtigung des Tastgrads des Systems erfolgen.

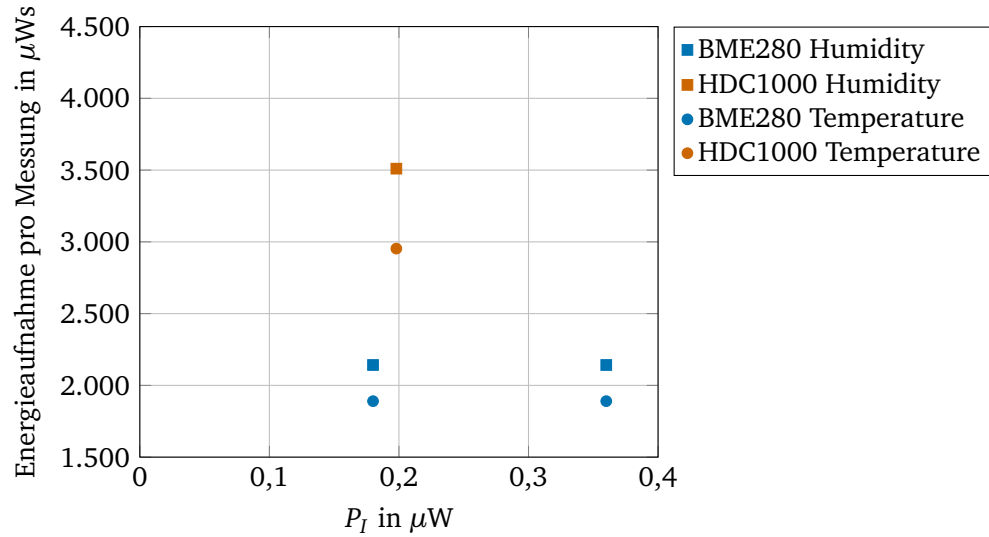


Abbildung 6.3.: Vergleich der Umgebungssensoren anhand der Leistungsaufnahme im inaktiven Modus P_I und der Energieaufnahme pro Messung

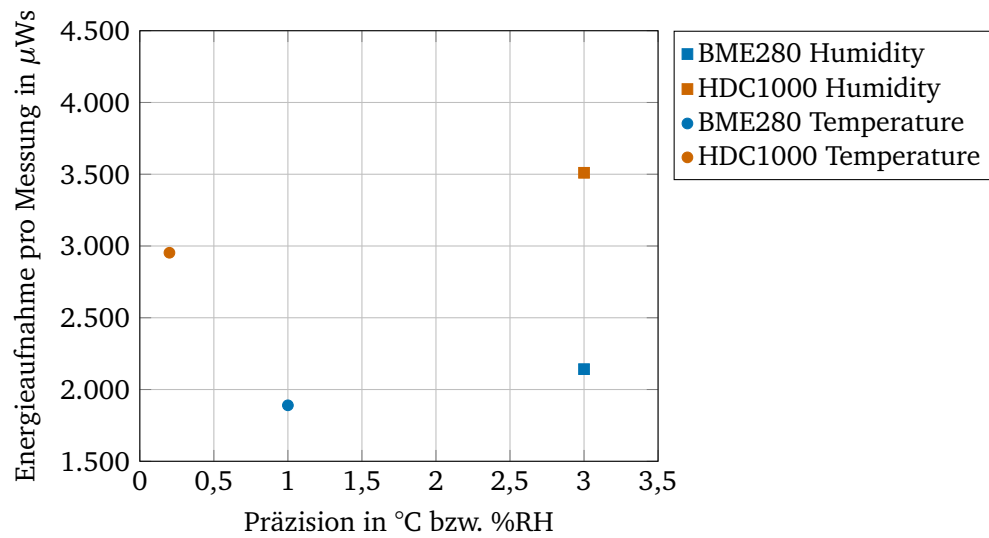


Abbildung 6.4.: Vergleich der Umgebungssensoren anhand der Präzision der Messung und der Energieaufnahme pro Messung

6.2.4 Komponente Beschleunigungssensor

Neben den zuvor beschriebenen Sensoren zur Erfassung der Umgebung soll der multimodale Sensorknoten für die Gebäudeautomation auch einen Beschleunigungssensor enthalten. Dieser könnte beispielsweise Erschütterungen in der Umgebung erfassen und somit Aktivität detektieren.

Ermittelte Parameter Für die Auswahl der Komponente wurden die am Markt erhältlichen Beschleunigungssensoren *STMicroelectronics LIS2DE* [176], *Bosch Sensortec BMA280* [27], *Analog Devices ADXL362* [9] und *Freescale Semiconductor MMA8451Q* [68] verglichen. Hier zeigt sich, dass die erforderlichen Werte nicht in den Datenblättern enthalten sind (vgl. Tabelle 6.8 und Tabelle 6.9). So finden sich für die Messmodi nur die mittlere Stromaufnahme der Komponente bei verschiedenen Ausführungsfrequenzen. Für den *BMA280* sind zudem die Stromaufnahme und die Dauer der Transitionen in den aktiven Modus angegeben. Alle Sensoren verfügen über mindestens einen inaktiven Modus für den auch die Stromaufnahme angegeben wird. Der *MMA8451Q* verfügt über eine I2C Schnittstelle, der *ADXL362* über eine SPI Schnittstelle. Der *LIS2DE* und der *BMA280* verfügen sowohl über eine I2C Schnittstelle, als auch eine SPI Schnittstelle. In keinem der Datenblätter finden sich Angaben zur Stromaufnahme der Schnittstellen. Somit werden diese Angaben im Modell auf 0 gesetzt.

Generierte Parameter Da die für das Modell benötigten Angaben nicht verfügbar sind, kann kein Vergleich der Komponenten anhand der Energieaufnahme pro Messung erfolgen. Als Vergleichswert wird daher der für alle Sensoren verfügbare Mittelwert der Strom- bzw. Leistungsaufnahme bei spezifischen Ausführungsfrequenzen verwendet. In Abbildung 6.5 ist die auf eine Messung normierte Energieaufnahme über dem Durchsatz aufgetragen. Die normierte Energieaufnahme ergibt sich aus der mittleren Leistungsaufnahme dividiert durch die entsprechende Ausführungsfrequenz. Es ist demnach die Energieaufnahme für einen Messzyklus bestehend aus der Messung im aktiven Modus und dem anschließenden Wechsel und Verbleib im inaktiven Modus. Der Durchsatz ergibt sich aus der Ausführungsfrequenz (in Hz) multipliziert mit der Datenbreite (in Bit). Der für den *BMA280* aufgeführte Modus *Low-power 1* entspricht laut Hersteller einem Wechsel zwischen dem *Normal mode* und dem *Suspend mode*. Der Modus *Low-power 2* einem Wechsel zwischen *Normal mode* und *Standby mode*.

Bewertung der Komponenten In der grafischen Darstellung in Abbildung 6.5 zeigt sich, dass der *ADXL362* im Vergleich die geringste Energieaufnahme pro Messzyklus aufweist. Jedoch hat die Komponente eine Gehäusefläche von $9,75 \text{ mm}^2$. Da eine kompakte Bauweise ein wichtiges Kriterium für den Sensorknoten ist und der *LIS2DE* im Vergleich nur eine geringfügig höhere Energieaufnahme aufweist, wird für das System die Komponente *LIS2DE* ausgewählt. Jedoch sind die für die Modellierung benötigten Daten nicht

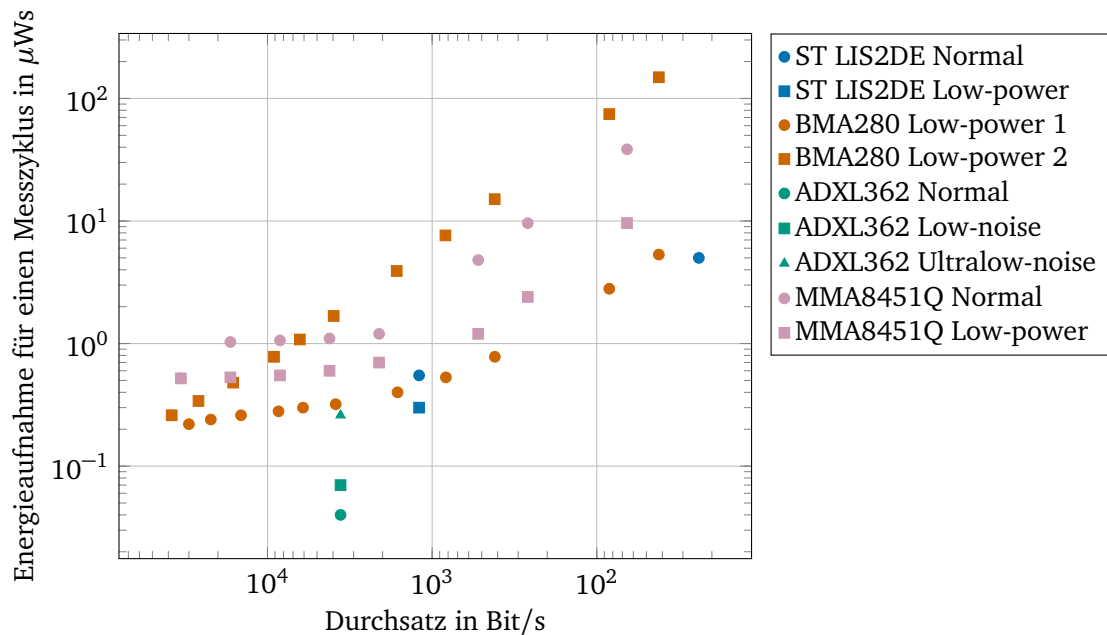


Abbildung 6.5.: Vergleich der Beschleunigungssensoren

verfügbar und müssen abgeschätzt werden. Die maximale Ausführungsfrequenz für den *LIS2DE* beträgt laut Datenblatt [176] 400 Hz, was einer Periode von 2,5 ms entspricht. Da keine höhere Ausführungsfrequenz möglich ist wird angenommen, dass der Sensor sich durchgehend im aktiven Modus befindet und somit die Periode von 2,5 ms der Messdauer entspricht. Im *Normal mode* weist der *LIS2DE* bei einer Ausführungsfrequenz von 50 Hz eine durchschnittliche Stromaufnahme von $11 \mu\text{A}$ auf. Im *Low-power mode* sind es bei einer Ausführungsfrequenz von 50 Hz $6 \mu\text{A}$. Eine Ausführungsfrequenz von 50 Hz entspricht einer Periode von 20 ms. Unter der Annahme, dass die Messdauer 2,5 ms beträgt, befindet sich der *LIS2DE* somit für 17,5 ms im inaktiven *Power-down mode* und für 2,5 ms im aktiven *Normal mode*. Da die Stromaufnahme für den *Power-down mode* bekannt ist, ergibt sich für den *Normal mode* eine Stromaufnahme von $84,5 \mu\text{A}$. Analog dazu kann die Stromaufnahme im *Low-power mode* berechnet werden. Diese beträgt $44,5 \mu\text{A}$. Auf Basis dieser abgeschätzten Werte können für den *LIS2DE* die Energieaufnahme für die Messungen ermittelt werden. Somit ergeben sich $528,125 \mu\text{Ws}$ für eine Messung im *Normal Modus* und $278,125 \mu\text{Ws}$ für eine Messung im *Low-power Modus*.

Tabelle 6.8.: Ermittelte Parameter der Beschleunigungssensoren: Versorgungsspannung und Modi

Parameter	Sensor 1	Sensor 2	Sensor 3	Sensor 4	Einheit
<i>description</i>	LIS2DE	BMA280	ADXL362	MMA8451Q	-
<i>voltage.min</i>	1,71	1,62	1,6	1,95	V
<i>voltage.typ</i>	2,5	2,4	2,0	2,5	V
<i>voltage.max</i>	3,6	3,6	3,5	3,6	V
<i>mode(1).description</i>	Normal	Normal	Normal	Normal	-
<i>mode(1).current</i>	n.v.	0,130	n.v.	n.v.	mA
<i>mode(1).datawidth</i>	24	42	36	42	Bit
<i>mode(1).time</i>	n.v.	n.v.	n.v.	n.v.	μs
<i>mode(1).active</i>	1	1	1	1	-
<i>mode(2).description</i>	Low-power	Stand-by	Low-noise	Low-power	-
<i>mode(2).current</i>	n.v.	0,062	n.v.	n.v.	mA
<i>mode(2).datawidth</i>	24	-	36	42	Bit
<i>mode(2).time</i>	n.v.	-	n.v.	n.v.	μs
<i>mode(2).active</i>	1	0	1	1	-
<i>mode(3).description</i>	Power-down	Suspend	Ultralow-noise	Stand-by	-
<i>mode(3).current</i>	0,0005	0,0021	n.v.	0,0018	mA
<i>mode(2).datawidth</i>	-	-	36	-	Bit
<i>mode(2).time</i>	-	-	n.v.	-	μs
<i>mode(3).active</i>	0	0	1	0	-
<i>mode(4).description</i>	-	Deep Suspend	Wake-up	-	-
<i>mode(4).current</i>	-	0,001	0,00027	-	mA
<i>mode(4).active</i>	-	0	0	-	-
<i>mode(5).description</i>	-	-	Standby	-	-
<i>mode(5).current</i>	-	-	0,00001	-	mA
<i>mode(5).active</i>	-	-	0	-	-
<i>transitions.current</i>	n.v.	$\begin{pmatrix} 0 & 0 & 0 & 0 \\ 0,130 & 0 & 0 & 0 \\ 0,130 & 0 & 0 & 0 \\ 0,130 & 0 & 0 & 0 \end{pmatrix}$	n.v.	n.v.	mA
<i>transitions.time</i>	n.v.	$\begin{pmatrix} 0 & 0 & 0 & 0 \\ 1.000 & 0 & 0 & 0 \\ 1.300 & 0 & 0 & 0 \\ 1.300 & 0 & 0 & 0 \end{pmatrix}$	n.v.	n.v.	μs

Tabelle 6.9.: Ermittelte Parameter der Beschleunigungssensoren: Schnittstellen und Fläche

Parameter	Sensor 1	Sensor 2	Sensor 3	Sensor 4	Einheit
<i>description</i>	LIS2DE	BMA280	ADXL362	MMA8451Q	-
<i>interface(1).no</i>	2	2	2	3	-
<i>interface(1).description</i>	SPI	SPI	SPI	I2C	-
<i>interface(1).current</i>	n.v.	n.v.	n.v.	n.v.	mA
<i>interface(1).datarate</i>	10.000	10.000	8.000	400	kbps
<i>interface(2).no</i>	3	3	-	-	-
<i>interface(2).description</i>	I2C	I2C	-	-	-
<i>interface(2).current</i>	n.v.	n.v.	-	-	mA
<i>interface(2).datarate</i>	400	400	-	-	kbps
<i>area</i>	4	4	9,75	9	mm ²

6.3 Komponenten der Spannungsversorgung

Die Spannungsversorgung eines Sensorknotens besteht aus Spannungswandler und Energiespeicher. Zur Auswahl der Komponenten muss die maximale Leistungsaufnahme (WCPC) und die maximale Energieaufnahme (WCEC) des Systems abgeschätzt werden. Die WCPC ist relevant für die Auswahl des Spannungswandlers und die WCEC für die Auswahl des Energiespeichers [125]. Im Folgenden werden, anhand der zuvor ausgewählten PMC des Systems, zunächst geeignete Spannungswandler bestimmt. Anschließend erfolgt die Auswahl geeigneter Energiespeicher.

6.3.1 Spannungswandler

Die zuvor beschriebene BauteilAuswahl hat den Entwurfsraum der PMC auf zwei verschiedene SoC (*BLE113* oder *nRF51822*), zwei verschiedene Umgebungslichtsensoren (*OPT3001* oder *MAX44009*), zwei verschiedene Umgebungssensoren (*BME280* oder *HDC1000*) und einen Beschleunigungssensor (*LIS2DE*) eingeschränkt. Zur Spannungsversorgung dieser Komponenten muss eine geeignete Versorgungsspannung bestimmt werden. Diese kann mit Hilfe der im Modell für jede Komponente hinterlegten minimalen (*voltage.min*) und maximalen Versorgungsspannung (*voltage.max*) ermittelt werden.

Der mögliche Bereich der Versorgungsspannung wird dabei von der höchsten minimalen Eingangsspannung und der niedrigsten maximalen Eingangsspannung der Komponenten bestimmt. Die höchste minimale Eingangsspannung liegt bei 2,7V (*HDC1000*). Die niedrigste maximale Eingangsspannung liegt bei 3,6V, da alle Komponenten, außer dem *HDC1000* (5V), diesen Wert aufweisen. Der Bereich der möglichen Versorgungsspannung liegt demnach bei 2,7 - 3,6V. Somit könnten alle Komponenten mit der gebräuchlichen Versorgungsspannung von 3,3V versorgt werden. Jedoch soll das System, laut Zielvorgabe

im Forschungsprojekt, mit einer Versorgungsspannung von 1,8 V arbeiten.

Da einige Komponenten eine höhere minimale Eingangsspannung aufweisen schränkt diese den zuvor bestimmten Entwurfsraum weiter ein. Dieser besteht somit aus einem SoC (*nRF51822*), zwei verschiedenen Umgebungslichtsensoren (*OPT3001* oder *MAX44009*), einem Umgebungssensor (*BME280*) und einem Beschleunigungssensor (*LIS2DE*). Für diese Komponenten liegt der gemeinsame Bereich der möglichen Versorgungsspannung bei 1,7 - 3,6 V. Somit ist eine Versorgungsspannung von 1,8 V möglich.

Zur Bestimmung des notwendigen Ausgangsstroms des Spannungswandlers wird die WCPC des System anhand der maximalen Leistungsaufnahme der Modi für alle PMC bestimmt. Die Leistungsaufnahme ergibt sich aus der Stromaufnahme (*mode.current*) und der typischen Versorgungsspannung (*voltage.typ*) für die die Stromaufnahme angegeben ist. Für den Transceiver *nRF51822* beträgt die maximale Leistungsaufnahme eines Modus 37,8 mW. Bei den Umgebungslichtsensoren beträgt die maximale Leistungsaufnahme des *OPT3001* 12,21 μW und die des *MAX44009* 2,16 μW . Die maximale Leistungsaufnahme des Umgebungssensors *BME280* beträgt 1,2852 mW, die des Beschleunigungssensors *LIS2DE* beträgt 211,25 μW . Da die maximale Leistungsaufnahme des *OPT3001* höher ist als die des *MAX44009* wird der *OPT3001* in die Abschätzung der WCPC einbezogen. Somit ergibt sich die Summe der maximalen Leistungsaufnahmen von *nRF51822*, *OPT3001*, *BME280* und *LIS2DE* mit 39,30855 mW. Bei einer Ausgangsspannung von 1,8 V muss der Spannungswandler mindestens einen Strom von 21,8381 mA zur Verfügung stellen können.

Als weiteres Kriterium für die Auswahl eines Spannungswandlers wird die minimale Leistungsaufnahme der Modi für alle Komponenten ermittelt. Für den Transceiver *nRF51822* beträgt die minimale Leistungsaufnahme 1,8 μW . Für die Umgebungslichtsensoren beträgt die minimale Leistungsaufnahme des *OPT3001* 0,99 μW , die des *MAX44009* 2,16 μW . Die des Umgebungssensors *BME280* beträgt 0,18 μW und für den Beschleunigungssensor *LIS2DE* beträgt die minimale Leistungsaufnahme 1,25 μW . Da die minimale Leistungsaufnahme des *OPT3001* niedriger ist als die des *MAX44009* wird der *OPT3001* in die Abschätzung einbezogen. Die Summe der minimalen Leistungsaufnahmen von *nRF51822*, *OPT3001*, *BME280* und *LIS2DE* ist 4,22 μW . Bei einer Ausgangsspannung von 1,8 V beträgt der minimale Ausgangsstrom somit 2,34 μA . Die Stromaufnahme des Systems liegt bei einer Ausgangsspannung von 1,8 V somit zwischen 2,34 μA und 21,8381 mA.

Ermittelte Parameter Entsprechend der ermittelten Anforderungen an Ausgangsspannung und Ausgangsstrom wurden die am Markt erhältlichen Spannungswandler *Texas Instruments TPS62202* [205] und *Texas Instruments TPS62243* [206] für einen näheren Vergleich ausgewählt. Die spezifischen Parameter minimale (*vin.min*) und maximale Eingangsspannung (*vin.max*), Ausgangsspannung (*vout*) und maximaler Ausgangsstrom (*iout.max*) konnten den Datenblättern entnommen werden und sind in Tabelle 6.10 aufgeführt.

Tabelle 6.10.: Ermittelte Parameter der Spannungswandler

Parameter	Spannungswandler 1	Spannungswandler 2	Einheit
<i>description</i>	TPS62202	TPS62243	-
<i>vin.min</i>	2,7	2,0	V
<i>vin.max</i>	5,0	6,0	V
<i>vout</i>	1,8	1,8	V
<i>iout.max</i>	300	300	mA
<i>input.iout</i>	0,05 - 300	22,12 - 300	mA
<i>input.effvinmin</i>	64,50 - 89,00	79,99 - 90,31	%
<i>input.effvinmax</i>	48,00 - 87,00	62,66 - 85,99	%

Zudem gehen die Effizienz bei minimaler (*input.effvinmin*) und maximaler Eingangsspannung (*input.effvinmax*) über den angegebenen Bereich des Ausgangsstroms (*input.iout*) als Parameter in das Modell ein. Für den *TPS62202* umfasst der Bereich des Ausgangsstroms 0,05 - 300 mA, wobei für die Effizienz bei minimaler und maximaler Eingangsspannung jeweils 35 Wertepaare angegeben werden. Für den *TPS62243* liegen jeweils 40 Wertepaare für den Bereich 22,12 - 300 mA vor. Die Werte konnten vom Hersteller in maschinenlesbarer Form exportiert werden [207]. Somit liegen alle für die Modellbildung benötigten Angaben vor. Die exportierten Werte der Effizienz sind in Abbildung 6.6 exemplarisch für den Spannungswandler *TPS62243* dargestellt.

Generierte Parameter Die generierten spezifischen Parameter des Spannungswandlers umfassen den vollen Bereich der Eingangsspannung (*vin.range*), den vollen Bereich des Ausgangsstrom (*iout.range*) und die Effizienz η über diese Bereiche (*eff*). Der Bereich der Eingangsspannung wird zwischen der minimalen Eingangsspannung *vin.min* und der maximalen Eingangsspannung *vin.max* in 0,1 V Schritten gebildet und im Modell abgelegt. Der volle Bereich des Ausgangsstroms wird zwischen 0 mA und dem maximalen Ausgangsstrom *iout.max* in 0,01 mA Schritten gebildet und im Modell abgelegt. Für die Effizienz über den vollen Bereich wird die vom Hersteller angegebene Effizienz bei minimaler und maximaler Eingangsspannung extra- bzw. interpoliert und auf dieser Basis für jede Eingangsspannung ein Fitting-Objekt generiert. Diese Fitting-Objekte ermöglichen es, die Effizienz des Spannungswandlers für die feste Ausgangsspannung *vout* für den gesamten Arbeitsbereich aus Eingangsspannung (*vin.range*) und Ausgangsstrom (*iout.range*) zu berechnen (vgl. Unterabschnitt 5.5.5). Die Fitting-Objekte werden im Modellparameter *eff* abgelegt. Somit konnten alle spezifischen Parameter der beiden Spannungswandler ermittelt bzw. generiert werden. Die für den gesamten Arbeitsbereich generierte Effizienz η des Texas Instruments *TPS62243* Spannungswandlers ist in Abbildung 6.7 dargestellt. Zur besseren Übersicht erfolgt die Darstellung nur für einige ausgewählte Eingangsspan-

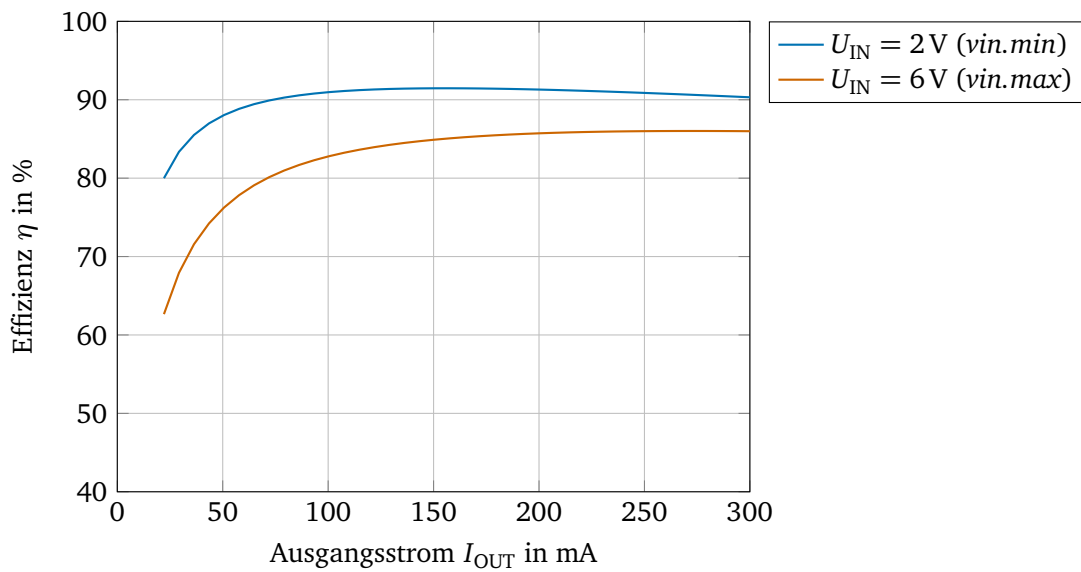


Abbildung 6.6.: Effizienz des *Texas Instruments TPS62243* Spannungswandlers auf Basis der vom Hersteller exportierten Daten

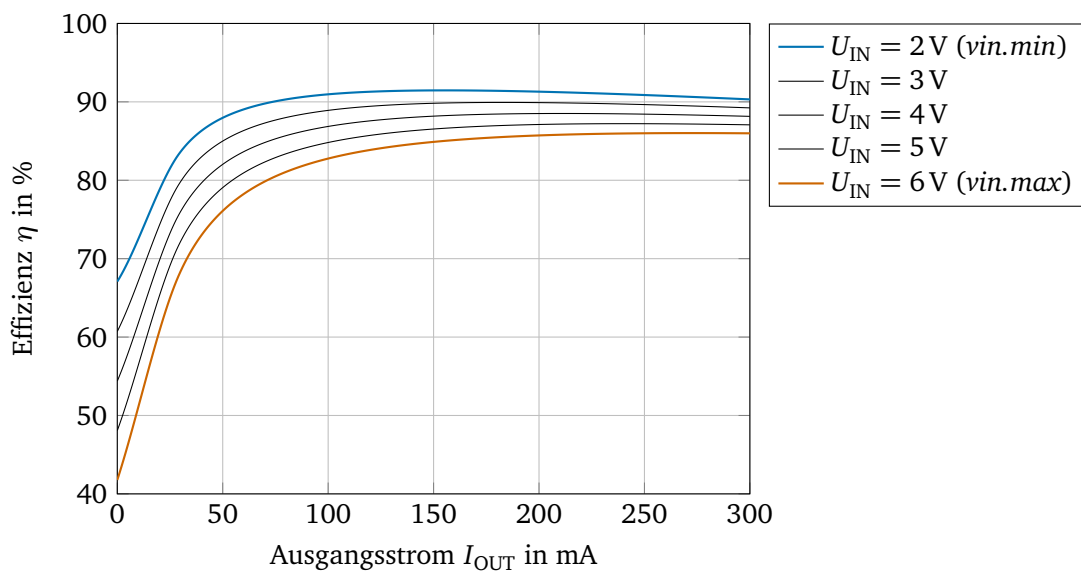


Abbildung 6.7.: Darstellung der inter- und extrapolierten Effizienz des *Texas Instruments TPS62243* Spannungswandlers auf Basis der vom Hersteller exportierten Daten

nungen. Analog dazu wurde die Effizienz des Spannungswandlers *TPS62202* für dessen Arbeitsbereich generiert und im Modell abgelegt.

Bewertung der Komponenten Beide Spannungswandler erfüllen mit einem maximalen Ausgangsstrom von 300 mA die Anforderung der WCPC. Zur Auswahl der Komponente wird daher die Effizienz für die Spannungsversorgung der PMC betrachtet. Wie zuvor ermittelt, liegt deren Stromaufnahme I_{SYS} , die dem Ausgangsstrom des Spannungswandlers entspricht, bei einer Ausgangsspannung von 1,8 V zwischen $2,34 \mu\text{A}$ und $21,8381 \text{ mA}$. Da die Stromaufnahme des Systems im Betrieb von der Anwendung abhängt, können hier zunächst nur die Grenzfälle betrachtet werden. Zudem hängt die Effizienz des Spannungswandlers von dessen Eingangsspannung, also der Klemmenspannung des versorgenden Energiespeichers ab, der noch nicht bestimmt ist. Somit erfolgt hier eine Betrachtung für den gesamten Bereich der möglichen Eingangsspannungen. In Abbildung 6.8 ist die Effizienz der beiden Spannungswandler *TPS62202* und *TPS62243* bei minimaler und maximaler Stromaufnahme des Systems für den jeweils möglichen Bereich der Eingangsspannungen dargestellt.

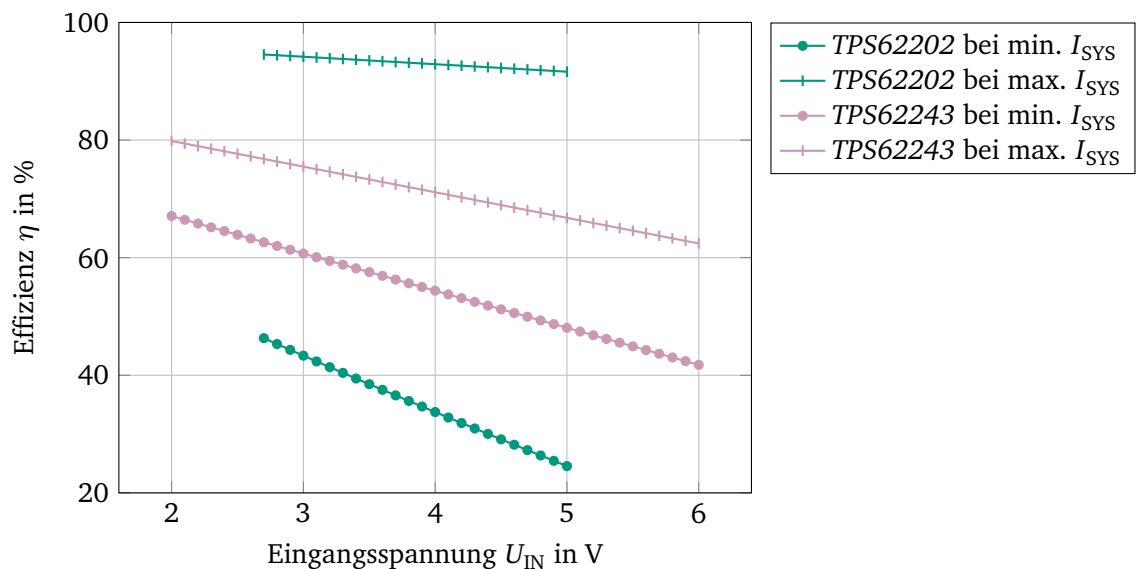


Abbildung 6.8.: Effizienz der Spannungswandler bei minimaler und maximaler Stromaufnahme des Systems für den jeweils möglichen Bereich der Eingangsspannungen

Hier zeigt sich, dass bei minimaler Stromaufnahme des Systems der *TPS62243* im Vergleich zum *TPS62202* eine durchgehend höhere Effizienz η aufweist. Bei maximaler Stromaufnahme ist hingegen der *TPS62202* für alle möglichen Eingangsspannungen

effizienter. Welcher Spannungswandler das System effizienter versorgen kann, hängt somit von der Stromaufnahme der anwendungsspezifischen Systemkonfiguration aus Taskdefinition und Tastgrad ab, die im weiteren Verlauf des Entwurfs betrachtet wird. Auf Basis der beiden Grenzbetrachtungen kann somit keine Auswahl der Komponente erfolgen.

6.3.2 Energiespeicher

Der Energiespeicher des Systems muss die Anforderungen der WCEC, also der maximalen Energieaufnahme, erfüllen [125, S. 258]. So muss sicher gestellt werden, dass die Kapazität der Batterie ausreichend ist, um die geplanten Tasks des Systems auszuführen [219]. Als obere Grenze für die WCEC kann das Produkt aus maximaler Leistungsaufnahme (WCPC) und maximaler Laufzeit (WCET) angenommen werden (vgl. Abschnitt 5.1). Im Modell findet sich die WCPC als die maximale Leistungsaufnahme eines Modus wieder. Die WCET ergibt sich entsprechend aus der maximalen Dauer in diesem Modus. Für alle Komponenten gilt, dass die Leistungsaufnahmen der aktiven Modi stets über denen der inaktiven Modi liegen. Für die PMC entspricht das Produkt aus WCPC und WCET somit der maximalen Energieaufnahme der aktiven Modi. Im Fall der Sensoren ist dieser Wert als die Energieaufnahme der Messungen (*mode.em eas*) im Komponentenmodell hinterlegt. Für die Komponenten Transceiver und Speicher kann die maximale Energieaufnahme als die Nutzung der aktiven Modi (z.B. Senden bzw. Speichern) mit der maximal möglichen Datenbreite als obere Schranke abgeschätzt werden. Beim Mikrocontroller hängt die Dauer des aktiven Modus vom auszuführenden Algorithmus ab, wobei der aktive Modus im energetisch ungünstigsten Fall durchgehend aktiv wäre. Die WCEC kann somit noch nicht hinreichend genau abgeschätzt werden. Die Abschätzung erfolgt daher im späteren Verlauf des Entwurfs.

Eine Vorwahl der Energiespeicher erfolgt zunächst anhand der Zielvorgaben im Forschungsprojekt. Demnach ist ein möglichst kompakter und kostengünstiger Energiespeicher auszuwählen. Zudem ist aus Gründen des Wartungsaufwandes ein Laden des Energiespeichers nicht vorgesehen. Bei vollständiger Entladung des Energiespeichers müsste demnach der Energiespeicher oder der gesamte Sensorknoten ausgetauscht werden. Entsprechend dieser Anforderungen soll eine kostengünstige und mit geringem Aufwand wechselbare Primärzelle verwendet werden. Daher werden am Markt erhältliche Knopfzellen untersucht.

Ermittelte Parameter Knopfzellen werden am Markt von vielen verschiedenen Herstellern in unterschiedlichen Bauformen angeboten. Entsprechend der Verfügbarkeit der Komponenten und deren Datenblätter wurden für eine genauere Betrachtung die Knopfzellen *Renata CR2032* [171], *Renata CR2025* [170], *Energizer CR2032* [58] und *Maxell CR2032H* [83] ausgewählt. Eine Übersicht der aus den Datenblättern ermittelten Parameter findet sich in Tabelle 6.11.

So finden sich in allen Datenblättern Angaben zu der Nennspannung U_{BAT} (*voltage.nom*),

Tabelle 6.11.: *Ermittelte Parameter der Energiespeicher für das Basismodell*

Parameter	Batterie 1	Batterie 2	Batterie 3	Batterie 4	Einheit
<i>description</i>	Renata CR2025	Renata CR2032	Energizer CR2032	Maxell CR2032H	- -
<i>voltage.nom</i>	3,0	3,0	3,0	3,0	V
<i>cap.nom</i>	165	235	235	220	mAh
<i>current.typ</i>	0,3	0,4	0,19	0,2	mA
<i>current.maxcont</i>	3,0	3,0	n.v.	n.v.	mA
<i>current.maxpuls</i>	n.v.	n.v.	n.v.	n.v.	mA
<i>voltage.cut</i>	2,0	2,0	2,0	2,0	V
<i>voltage.rt</i>	[5x2]	[5x2]	[1x2]	[6x2]	mA,h
<i>i1</i>	n.v.	n.v.	0,2	0,003	mA
<i>i2</i>	n.v.	n.v.	7,5	10,0	mA
<i>vsoc1</i>	n.v.	n.v.	[44x2]	[56x2]	mAh,V
<i>vsoc2</i>	n.v.	n.v.	[39x2]	[44x2]	mAh,V

der Nennkapazität C_{BAT} (*cap.nom*) und dem typischen Entladestrom (*current.typ*). Der maximale kontinuierliche Entladestrom (*current.maxcont*) ist nur bei zwei Typen angegeben. Angaben zum maximalen gepulsten Entladestrom (*current.maxpuls*) finden sich in keinem Datenblatt. Die Entladeschlussspannung U_{CUT} (*voltage.cut*) ist für alle Typen angegeben.

Zur Generierung des **Basismodells** der Komponente wird der Verlauf der Klemmenspannung über der Entladezeit für verschiedene konstante Entladeströme benötigt. Diese ist in den Datenblättern grafisch dargestellt. Der Zeitpunkt, an dem die Klemmenspannung auf die Entladeschlussspannung absinkt, markiert den Zeitpunkt der Entladung. Somit kann ein Zusammenhang zwischen dem Entladestrom und der daraus folgenden Dauer der Entladung hergestellt werden (vgl. Unterabschnitt 5.5.6). Der Parameter *voltage.rt* enthält diese Datenpaare aus Entladestrom und Laufzeit, wobei für den Typ *Renata CR2032* 5 Datenpaare, den Typ *Renata CR2025* 5 Datenpaare, den Typ *Energizer CR2032* 1 Datenpaar und für den Typ *Maxell CR2032H* 6 Datenpaare ermittelt werden konnten. In allen Fällen mussten die Werte manuell aus der grafischen Darstellung im Datenblatt abgelesen werden.

Für das **erweiterte Modell** wird der Verlauf der Klemmenspannung über dem SOC (*vsoc1*, *vsoc2*) für zwei verschiedene Ströme (*i1*, *i2*) benötigt. Diese Angaben finden sich als Graph in den Datenblättern von *Energizer CR2032* [58] und *Maxell CR2032H* [83]. Da hier der gesamte Verlauf der Klemmenspannung relevant ist, werden die Daten nicht manuell abgelesen sondern mittels MATLAB Skript [112] digitalisiert. Jedoch erfordert auch die Nutzung des Skripts manuelle Schritte. Für den Typ *Energizer CR2032* wurden

für den Verlauf der Klemmenspannung über dem SOC 44 Datenpunkte (*vsoc1*) bzw. 39 Datenpunkte (*vsoc2*) extrahiert. Für den Typ *Maxell CR2032H* sind es 56 Datenpunkte (*vsoc1*) bzw. 44 Datenpunkte (*vsoc2*). Die extrahierten Verläufe der Klemmenspannungen sind für beide Knopfzellen in Abbildung 6.9 dargestellt.

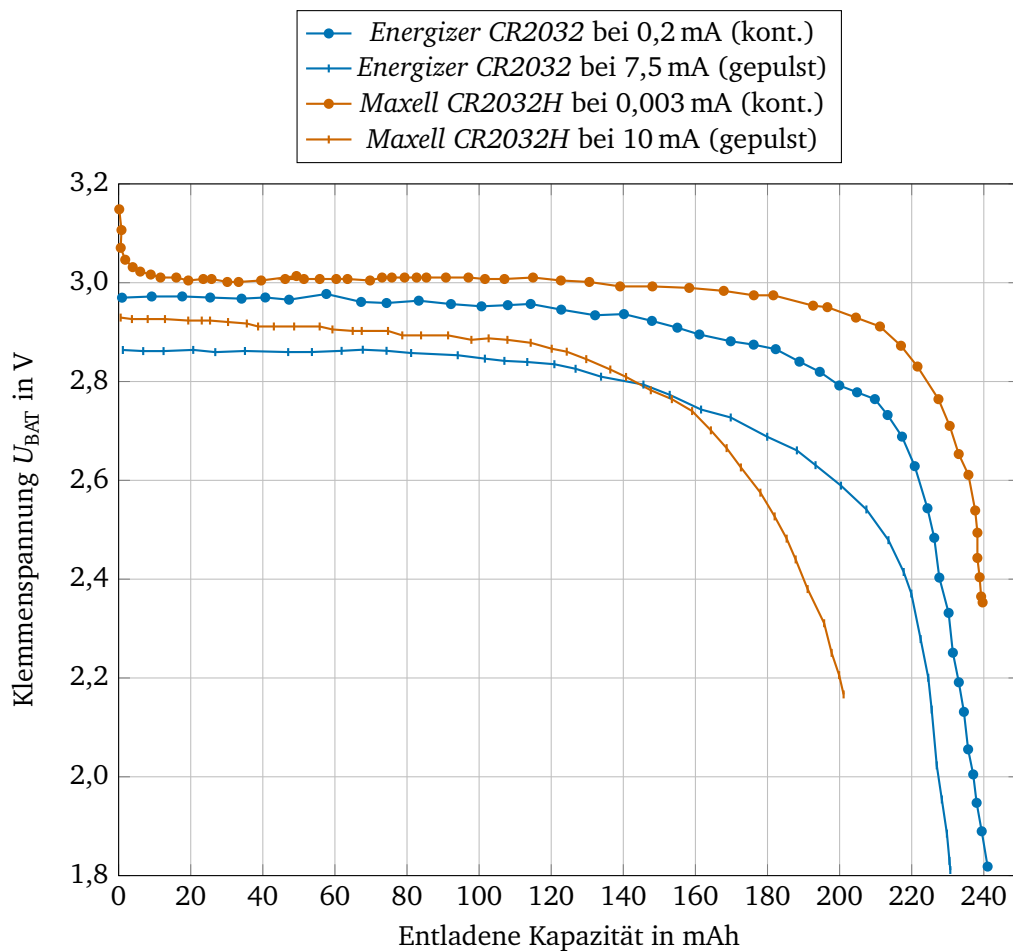


Abbildung 6.9.: Verlauf der Klemmenspannung in Abhängigkeit des SOC für jeweils zwei verschiedene Entladeströme

Generierte Parameter Auf Basis der aus den Datenblättern extrahierten Daten können die Modelle für die Komponenten generiert werden. Wie in Tabelle 6.11 dargestellt, liegt für die Knopfzellen *Renata CR2032* [171], *Renata CR2025* [170] und *Maxell CR2032H* [83] der für die Generierung des **Basismodells** benötigte Verlauf der Klemmenspannung über

6. Modellgestützter Entwurf eines Sensorknotens

der Entladezeit (*voltage.rt*) für mindestens zwei verschiedene Entladeströme vor. Für die Knopfzelle *Energizer CR2032* [58] liegt nur ein Datenpaar vor. In Abbildung 6.10 sind die Datenpaare und die darauf basierenden angenäherten Kurven dargestellt. Die Abweichung der Kurve für die Batterie *Maxell CR2032H* resultiert aus der höheren Anzahl der Datenpaare für sehr niedrige Ströme.

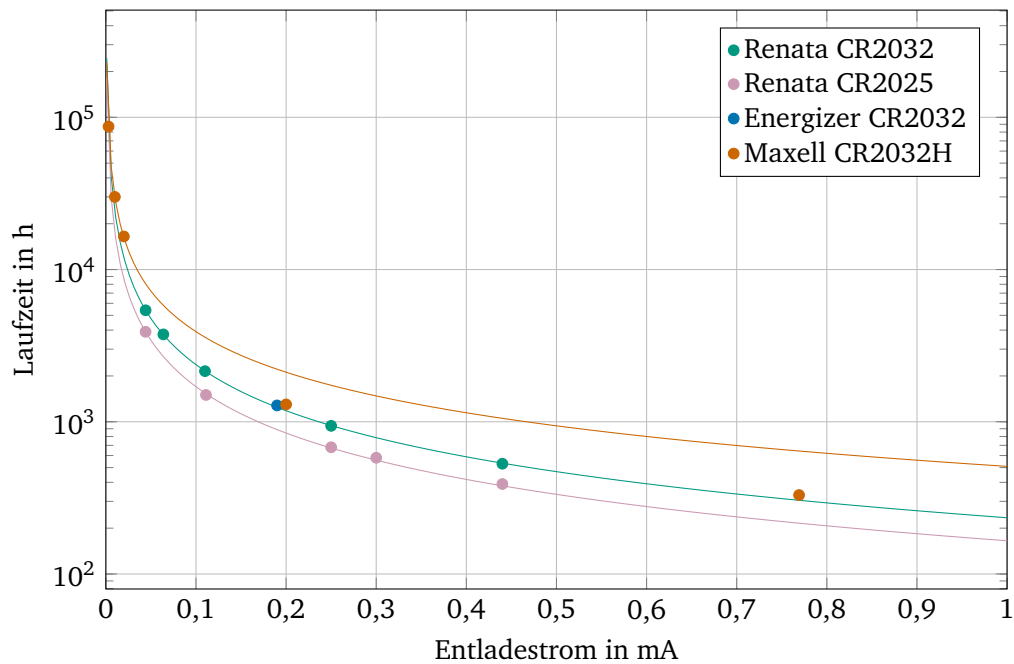


Abbildung 6.10.: Annäherung von Kurven zur Ermittlung der Peukert-Kapazität und der Peukert-Zahl

Daraus ergeben sich die in Tabelle 6.12 aufgeführten Werte für die Peukert-Kapazität C_{PK} und die Peukert-Zahl k , die im jeweiligen Komponentenmodell hinterlegt werden.

Tabelle 6.12.: Generierte Parameter der Energiespeicher für das Basismodell

Parameter	Batterie 1	Batterie 2	Batterie 3	Batterie 4	Einheit
<i>description</i>	Renata CR2025	Renata CR2032	Energizer CR2032	Maxell CR2032H	-
<i>cap.pk</i>	165,3198	234,1441	-	509,3095	mAh
<i>k</i>	1,01219	1,00597	-	0,88498	-

Der für das **erweiterte Modell** benötigte Verlauf der Klemmenspannung über dem SOC (v_{soc1} , v_{soc2}) für zwei verschiedene Ströme (i_1 , i_2) liegt für die Knopfzellen *Energizer CR2032* [58] und *Maxell CR2032H* [83] (vgl. Tabelle 6.11 und Abbildung 6.9) vor. Entsprechend dem in Unterabschnitt 5.5.6 beschriebenen Vorgehen kann aus der Differenz der beiden Spannungsverläufe der Innenwiderstand in Abhängigkeit vom SOC bestimmt werden. Der Verlauf des Innenwiderstands für beide Knopfzellen ist in Abbildung 6.11 dargestellt.

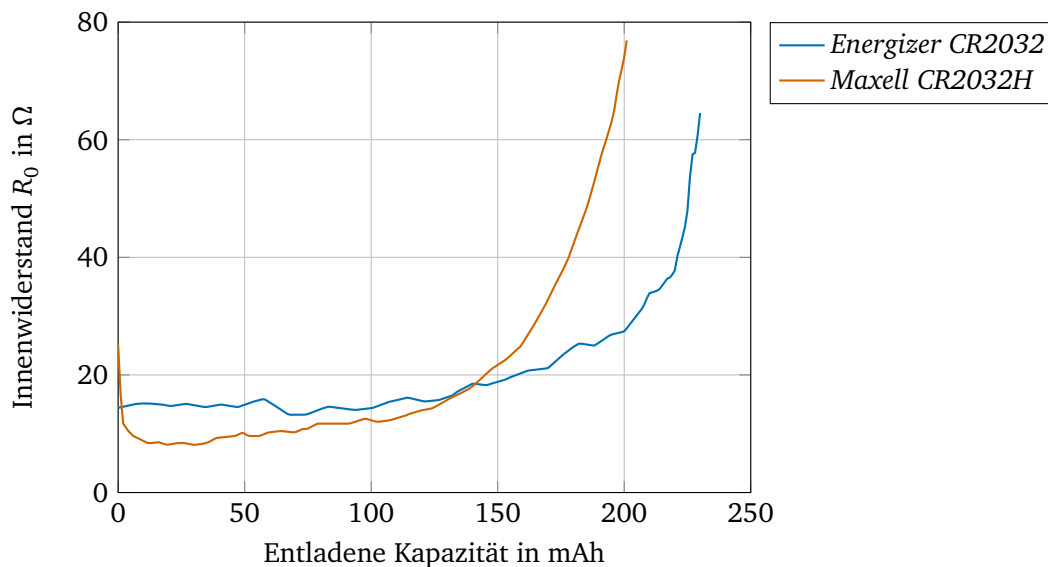


Abbildung 6.11.: Innenwiderstand in Abhängigkeit vom SOC

Nach Berechnung des Innenwiderstands kann anschließend für beide Knopfzellen der Verlauf der Leerlaufspannung U_{OC} in Abhängigkeit des SOC berechnet werden. Der Verlauf ist in Abbildung 6.12 dargestellt. Der Abfall der Leerlaufspannung bei beginnender Entladung der *Maxell CR2032H* ist auf den Abfall der aus dem Datenblatt extrahierten Klemmenspannung bei einer kontinuierlichen Entladung mit 0,003 mA zurückzuführen (vgl. Abbildung 6.9). Dieses entspricht dem resistiven Abfall der Klemmenspannung bei beginnender Entladung (vgl. Abbildung 4.5). Im Datenblatt der *Energizer CR2032* ist dieses nicht abgebildet. Aus diesen Daten werden MATLAB Fitting-Objekte für den Innenwiderstand R_0 und die Leerlaufspannung U_{OC} generiert und im Modell hinterlegt (*fitobjr0*, *fitobjuoc*). Diese Fitting-Objekte können als Funktionen verwendet werden und geben für den SOC als Eingabewert den entsprechenden Innenwiderstand bzw. die Leerlaufspannung zurück. Über den Zusammenhang zwischen Innenwiderstand, Leerlaufspannung und Klemmenspannung im elektrischen Modell kann somit die Klemmenspannung für jeden Arbeitspunkt aus SOC und Entladestrom ermittelt werden.

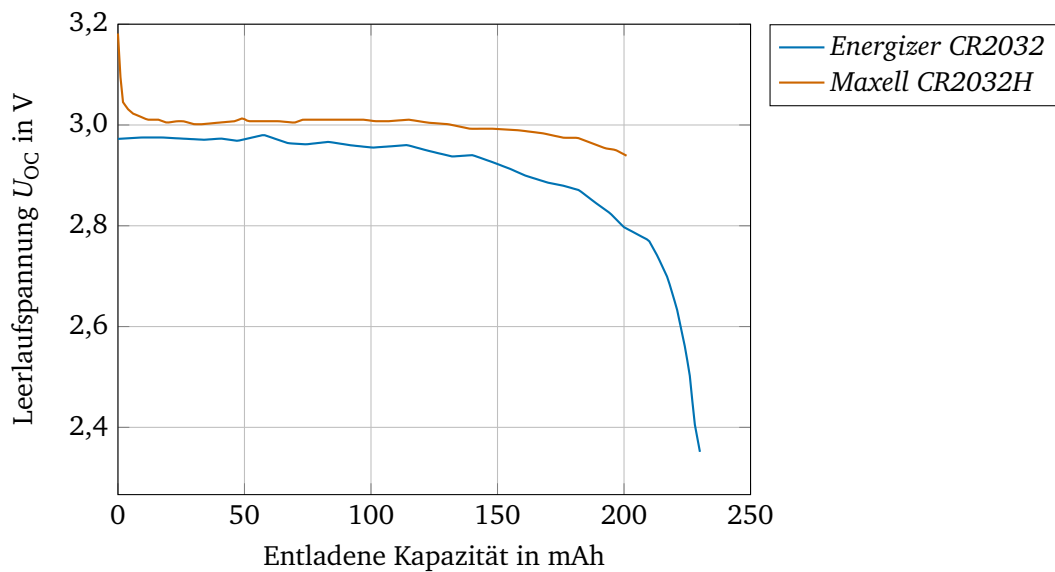


Abbildung 6.12.: Leerlaufspannung in Abhängigkeit vom SOC

Bewertung der Komponenten Welche Laufzeit des Systems mit welchem Energiespeicher erreicht werden kann hängt, neben den Parametern des Energiespeichers, von der anwendungsspezifischen Systemkonfiguration aus Taskdefinition und Tastgrad (und somit der WCEC) ab, die im weiteren Verlauf des Entwurfs betrachtet wird. Somit kann an dieser Stelle noch keine Bewertung der Komponenten erfolgen. So bestimmen lediglich die zur Verfügung stehenden Daten, ob eine Komponente in den Entwurfsraum aufgenommen werden kann.

6.4 Entwurfsraum der Systemkonfiguration

Wie in Abschnitt 5.7 beschrieben, legt eine Systemkonfiguration die Komponenten des Systems fest. Diese umfassen sowohl die aktiven Komponenten (PMC) der Typen Mikrocontroller, Sensoren, Transceiver und Speicher als auch die Komponenten der Spannungsversorgung, Spannungswandler und Energiespeicher. Der zu betrachtende Entwurfsraum der Systemkonfigurationen $N_{\text{EWR,SYS}}$ ergibt sich durch die Anzahl der verschiedenen Spannungswandler N_{DC} , die Anzahl der Energiespeicher N_{BAT} und die Größe des Entwurfsraums der aktiven Komponenten $N_{\text{EWR,PMC}}$ (vgl. Gleichung 5.18). Die Größe des Entwurfsraums $N_{\text{EWR,PMC}}$ ergibt sich wiederum aus der Anzahl der verschiedenen Mikrocontroller N_{UC} , Sensoren N_{SEN} , Transceiver N_{TC} und Speicher N_{MEM} sowie der Anzahl der aktiven und inaktiven Modi ($N_{\text{M,A}}$ und $N_{\text{M,I}}$) dieser Komponenten (vgl. Gleichung 5.19). Im Folgenden wird aufgeführt, wie der Entwurfsraum der Systemkonfigurationen aufgebaut wird und

welche Parameter vom Modell generiert werden. Darauf folgend wird die Bewertung der Systemkonfigurationen beschrieben.

Aufbau des Entwurfsraums In der zuvor beschriebenen Analyse der Komponenten wurden die im Folgenden aufgeführten zwei SoC, zwei Umgebungslichtsensoren, vier Umgebungssensoren, vier Beschleunigungssensoren, zwei Spannungswandler und vier Energiespeicher betrachtet:

- SoC: *BLE113*, *nRF51822*
- Umgebungslichtsensor: *OPT3001*, *MAX44009*
- Umgebungssensor: *BME280*, *HDC1000*, *SHT20*, *MS8607*
- Beschleunigungssensor: *LIS2DE*, *BMA280*, *ADXL362*, *MMA8451*
- Spannungswandler: *TPS62202*, *TPS62243*
- Energiespeicher: *Renata CR2032*, *Renata CR2025*, *Energizer CR2032*, *Maxell CR2032H*

Neben der Anzahl der verschiedenen Komponenten geht auch die Anzahl der aktiven und inaktiven Modi in den Entwurfsraum ein. Zur besseren Übersicht wurden diese in Tabelle 6.13 zusammen gefasst.

Tabelle 6.13.: Anzahl der Modi der PMC. Optionale Modi mit gleicher Funktionalität sind fett markiert.

Komponente	$N_{M,A}$	$N_{M,I}$
BLE113	3 (CPU,RX,TX)	3
nRF51822	3 (CPU,RX,TX)	2
OPT3001	2 (Full-scale, Dark)	2 (Full-scale, Dark)
MAX44009	1 (Active)	0
BME280	4 (H,T, P ,HPT)	2
HDC1000	2 (H,T)	1
SHT20	2 (H,T)	1
MS8607	3 (H,T,P)	1
LIS2DE	2 (Normal, Low-power)	1
BMA280	1 (Normal)	3
ADXL362	3 (Normal, Low-noise, Ultralow-noise)	2
MMA8451	2 (Normal, Low-power)	1

Bei der Anzahl der Modi ist zu beachten, dass die verschiedenen Modi echte Optionen für den Entwurfsraum darstellen müssen. So verfügen die SoC jeweils über drei verschiedene aktive Modi, jedoch stellen diese unterschiedliche Funktionalitäten dar und können somit nicht optional verwendet werden. Das Gleiche gilt für die aktiven Modi der Umgebungssensoren. Die aktiven Modi der Umgebungslichtsensoren und Beschleunigungssensoren

6. Modellgestützter Entwurf eines Sensorknotens

hingegen stellen optionale Modi der gleichen Funktionalität dar. Die jeweiligen optionalen Modi der PMC sind in Tabelle 6.13 fett markiert.

Dementsprechend ergibt sich die Größe des Entwurfsraums der PMC aus der Anzahl der SoC N_{SoC} , Umgebungslichtsensoren $N_{\text{SEN,LUX}}$, Umgebungssensoren $N_{\text{SEN,AMB}}$ und Beschleunigungssensoren $N_{\text{SEN,ACC}}$ sowie der Anzahl der aktiven und inaktiven Modi ($N_{M,A}$, $N_{M,I}$).

$$\begin{aligned}
 N_{\text{EWR1,PMC}} &= \sum_{i=1}^{N_{\text{SoC}}} N_{M,A,i} N_{M,I,i} \cdot \sum_{i=1}^{N_{\text{SEN,LUX}}} N_{M,A,i} N_{M,I,i} \cdot \sum_{i=1}^{N_{\text{SEN,AMB}}} N_{M,A,i} N_{M,I,i} \cdot \sum_{i=1}^{N_{\text{SEN,ACC}}} N_{M,A,i} N_{M,I,i} \\
 &= 5 \cdot 5 \cdot 5 \cdot 13 \\
 &= 1.625
 \end{aligned} \tag{6.9}$$

Im Rahmen der BauteilAuswahl der PMC (Abschnitt 6.2) erfolgte bereits eine Bewertung und eine daraus resultierende Vorauswahl der Komponenten. Nach der Vorauswahl ergeben sich die im Folgenden aufgeführten zwei SoC, zwei Umgebungslichtsensoren, zwei Umgebungssensoren und ein Beschleunigungssensor:

- SoC: *BLE113*, *nRF51822*
- Umgebungslichtsensor: *OPT3001*, *MAX44009*
- Umgebungssensor: *BME280*, *HDC1000*
- Beschleunigungssensor: *LIS2DE*

Dementsprechend ergibt sich der reduzierte Entwurfsraums *EWR2* der PMC:

$$\begin{aligned}
 N_{\text{EWR2,PMC}} &= \sum_{i=1}^{N_{\text{SoC}}} N_{M,A,i} N_{M,I,i} \cdot \sum_{i=1}^{N_{\text{SEN,LUX}}} N_{M,A,i} N_{M,I,i} \cdot \sum_{i=1}^{N_{\text{SEN,AMB}}} N_{M,A,i} N_{M,I,i} \cdot \sum_{i=1}^{N_{\text{SEN,ACC}}} N_{M,A,i} N_{M,I,i} \\
 &= 5 \cdot 5 \cdot 3 \cdot 2 \\
 &= 150
 \end{aligned} \tag{6.10}$$

Zudem hat die Auswahl der Spannungswandler in Unterabschnitt 6.3.1 und die damit einhergehende Festlegung der Versorgungsspannung des Systems auf 1,8 V eine weitere Reduzierung der PMC bewirkt:

- SoC: *nRF51822*
- Umgebungslichtsensor: *OPT3001*, *MAX44009*
- Umgebungssensor: *BME280*
- Beschleunigungssensor: *LIS2DE*

Dementsprechend ergibt sich der reduzierte Entwurfsraums *EWR3* der PMC:

$$\begin{aligned}
 N_{\text{EWR3,PMC}} &= \sum_{i=1}^{N_{\text{SoC}}} N_{M,A,i} N_{M,I,i} \cdot \sum_{i=1}^{N_{\text{SEN,LUX}}} N_{M,A,i} N_{M,I,i} \cdot \sum_{i=1}^{N_{\text{SEN,AMB}}} N_{M,A,i} N_{M,I,i} \cdot \sum_{i=1}^{N_{\text{SEN,ACC}}} N_{M,A,i} N_{M,I,i} \\
 &= 2 \cdot 5 \cdot 2 \cdot 2 \\
 &= 40
 \end{aligned} \tag{6.11}$$

Aus der Anzahl der aktiven Komponenten und deren Modi ergeben sich somit 150 mögliche Konfigurationen für den Entwurfsraum *EW2* und 40 mögliche Konfigurationen für den Entwurfsraum *EW3*. Zusammen mit den zwei verschiedenen Spannungswandlern und den vier verschiedenen Energiespeichern ergeben sich 1200 bzw. 320 mögliche Systemkonfigurationen.

$$\begin{aligned}
 N_{\text{EW3,SYS}} &= N_{\text{BAT}} \cdot N_{\text{DC}} \cdot N_{\text{EW3,PMC}} \\
 &= 4 \cdot 2 \cdot 40 \\
 &= 320
 \end{aligned} \tag{6.12}$$

Der Entwurfsraum der Systemkonfigurationen besteht somit aus den möglichen Komponenten und deren optionalen aktiven und inaktiven Modi (vgl. Tabelle 6.14). An dieser Stelle könnte optional eine weitere Reduktion des Entwurfsraums anhand von nicht-funktionalen Anforderungen wie Größe, Kosten oder Verfügbarkeit der Komponenten durchgeführt werden.

Tabelle 6.14.: Größe der Entwurfsräume

Entwurfsraum	Komponenten	Spannungsdomäne	$N_{\text{EW,PMC}}$	$N_{\text{EW,SYS}}$
EW1	Alle	Alle	1.625	13.000
EW2	Vorauswahl	Alle	150	1.200
EW3	Vorauswahl	1,8 V	40	320

Die Systemkonfigurationen werden, wie in Abschnitt 5.7 beschrieben, im Modell abgelegt. Durch die Auftrennung der Komponenten nach unterschiedlichen Funktionalitäten ergibt sich im Modell die Trennung des SoC in Mikrocontroller (*uc*) und Transceiver (*transceiver*) sowie die Trennung des Umgebungssensors in Feuchtigkeitssensor (*hum*) und Temperatursensor (*temp*). Somit wird sichergestellt, dass für jeden Komponententyp nur optional zu verwendende Modi enthalten sind. In der Systemkonfigurationen sind zudem die Spannungswandler und Batterien als *converter.devices* bzw. *bat.devices* hinterlegt.

Generierte Parameter Auf Basis der Komponentenmodelle wird zunächst die Leistungsaufnahme im inaktiven Modus für alle Varianten der PMC bestimmt. Die Leistungsaufnahme der einzelnen Komponenten berechnet sich jeweils aus der Stromaufnahme der Modi und der typischen Versorgungsspannung der Komponente. Die Leistungsaufnahme der 40 möglichen Systemkonfigurationen der PMC in *EW3* liegt zwischen $4,22 \mu\text{W}$ und $15,17 \mu\text{W}$ und ist in Abbildung 6.13 dargestellt.

Die höhere Leistungsaufnahme für die Konfigurationen K1 bis K20 sind auf den inaktiven Modus *ON* des *nRF51822* SoC zurückzuführen, welcher eine Stromaufnahme von $3,8 \mu\text{A}$ bei einer Versorgungsspannung von 3 V aufweist. Die Stromaufnahme des inaktiven Modus

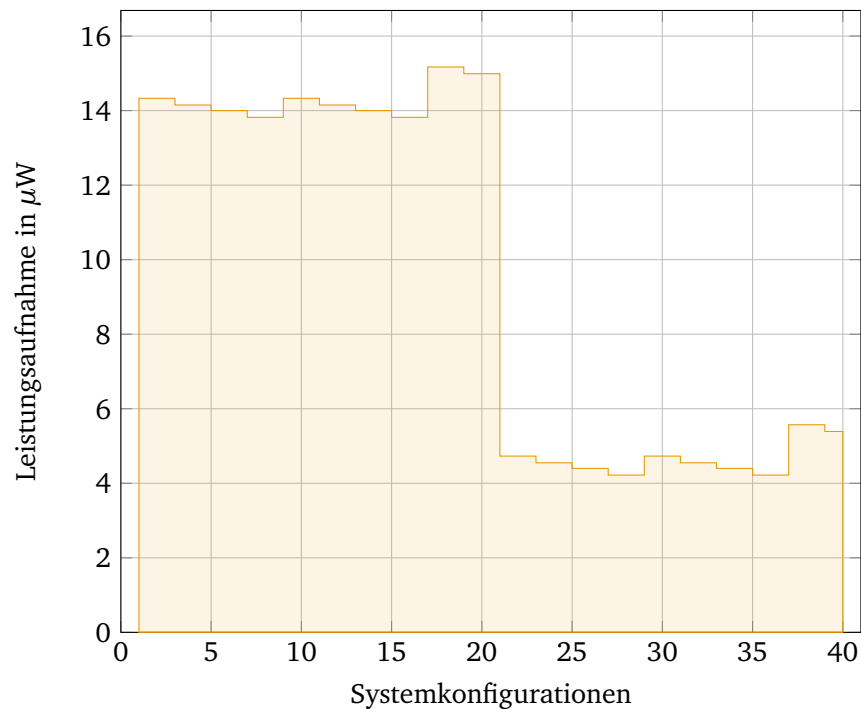


Abbildung 6.13.: Leistungsaufnahme im inaktiven Modus für alle Systemkonfigurationen im Entwurfsraum *EWR3*

OFF in den Konfigurationen K21 bis K40 liegt bei $0,6 \mu A$. Zur genaueren Betrachtung wird aus diesen 40 Varianten für jede Komponente jeweils der inaktive Modus mit der geringsten Leistungsaufnahme vorausgewählt. Die Optionen des Entwurfsraums werden somit auf die Auswahl der Komponenten eingeschränkt und es ergeben sich 2 mögliche Varianten. Die Komponenten dieser 2 Varianten und die daraus resultierende minimal mögliche Leistungsaufnahme des Systems im inaktiven Modus (*idle.pmcpower*) ist in Tabelle 6.15 aufgeführt.

Tabelle 6.15.: Minimale Leistungsaufnahme der PMC im inaktiven Modus

Bez.	Konf.	SoC	LUX	AMB	ACC	$P_{PMC,I}$
MIN1	K27,K28,K35,K36	nRF51822	OPT3001	BME280	LIS2DE	4,22 μW
MIN2	K39,K40	nRF51822	MAX44009	BME280	LIS2DE	5,39 μW

Diese Varianten werden im Folgenden als *MIN1* und *MIN2* bezeichnet. Diese minimalen

Leistungsaufnahmen können, wie in Tabelle 6.15 aufgeführt, für mehrere Konfiguration im Entwurfsraum gelten. Diese unterscheiden sich entsprechend nur in der Wahl der aktiven Modi und weisen die gleichen Komponenten mit den gleichen inaktiven Modi auf. Zur weiteren Analyse kann die Leistungsaufnahme der Systemkonfigurationen getrennt für die einzelnen Komponenten einer Systemkonfiguration betrachtet werden. Diese anteilige Leistungsaufnahme der PMC im inaktiven Modus (*idle.dmcpowerratio*) ist in Abbildung 6.14 dargestellt.

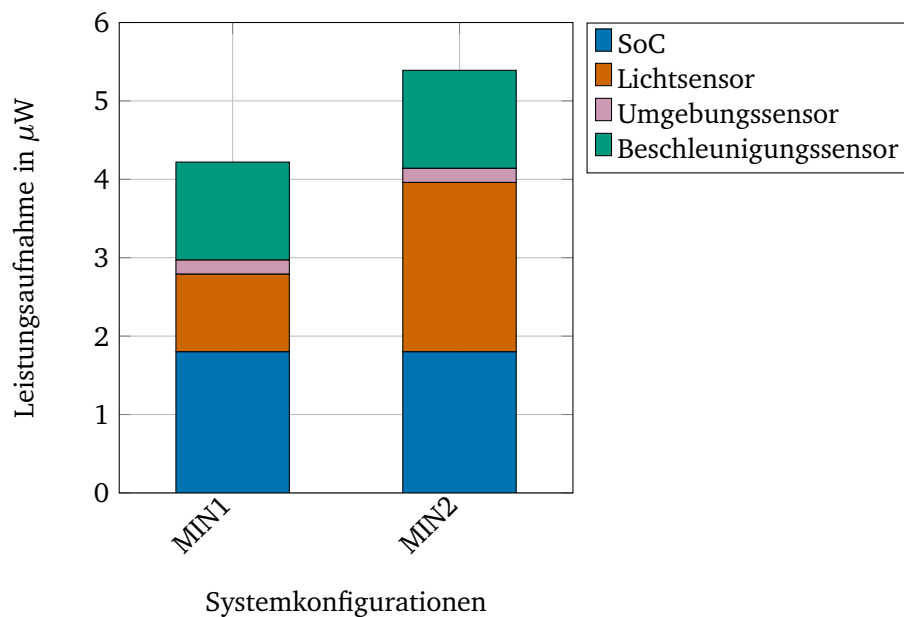


Abbildung 6.14.: Anteilige minimale Leistungsaufnahme der PMC im inaktiven Modus

Wie in Abbildung 6.14 zu erkennen ist, hat der Lichtsensor *OPT31001* (*MIN1*) eine geringere Leistungsaufnahme als der Lichtsensor *MAX44009* (*MIN2*). Die Konfiguration *MIN1* weist somit insgesamt die geringste Leistungsaufnahme auf.

Neben der Leistungsaufnahme der PMC können auch die im Entwurfsraum hinterlegten Spannungswandler einbezogen werden. Deren Effizienz hängt vom Arbeitspunkt aus Eingangsspannung, Ausgangsspannung und Ausgangsstrom ab (vgl. Unterabschnitt 5.5.5). Die Ausgangsspannung ist durch die in Unterabschnitt 6.3.1 definierte Versorgungsspannung des Systems von 1,8V gegeben. Der Ausgangsstrom der Spannungswandler entspricht der ermittelten Leistungsaufnahme der PMC bei dieser Versorgungsspannung. Die Eingangsspannung entspricht der Batteriespannung. Da diese vom Zustand der Batterie abhängt, wird diese zunächst mit der Nennspannung der Batterien angenommen. Diese ist für alle Batterien im Entwurfsraum gleich und liegt bei 3,0V. Auf dieser Basis ergibt sich eine

Effizienz des Spannungswandlers *TPS62202* von 43,33 % für die Leistungsaufnahme der Konfiguration *MIN1* und 43,59 % für die Leistungsaufnahme der Konfiguration *MIN2*. Für den Spannungswandler *TPS62243* liegt die Effizienz sowohl für *MIN1* als auch *MIN2* bei 60,72 %. Daraus resultieren die in Abbildung 6.15 dargestellten Leistungsaufnahmen der Spannungswandler (*idle.dcpower*) für die 2 Systemkonfigurationen der PMC. Die Konfigurationen *MIN1DC1* und *MIN2DC1* beinhalten den Spannungswandler *TPS62202*. Die Konfigurationen *MIN1DC2* und *MIN2DC2* hingegen den Spannungswandler *TPS62243*.

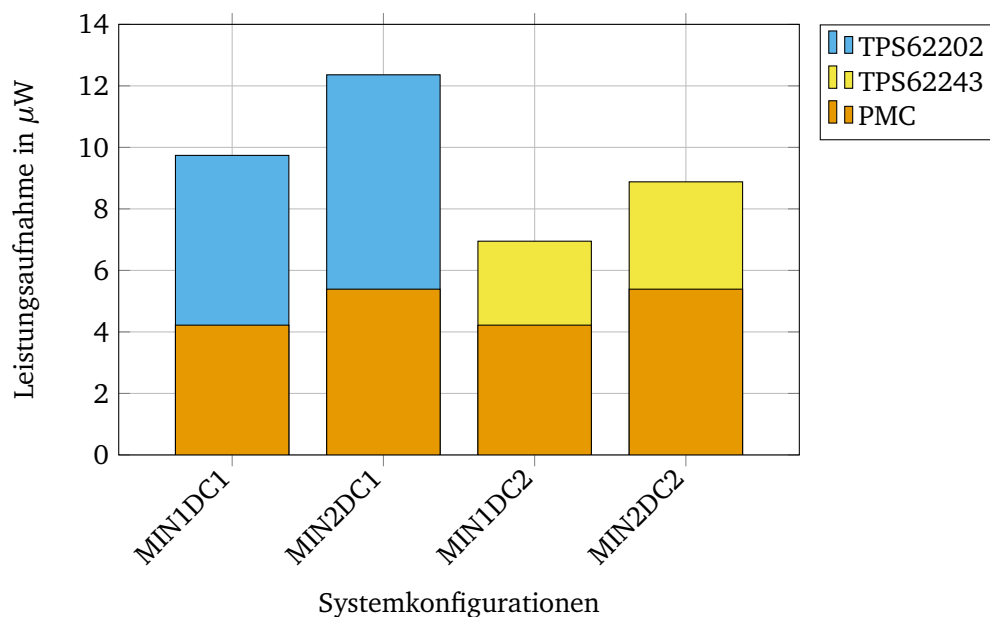


Abbildung 6.15.: Anteilige minimale Leistungsaufnahme der PMC und Spannungswandler im inaktiven Modus

Hier wird deutlich, dass die höhere Effizienz des *TPS62243* Spannungswandlers zu einer entsprechend geringeren Leistungsaufnahme führt. Somit ist der *TPS62243* für alle Systemkonfigurationen der effizientere Spannungswandler für den inaktiven Zustand (*idle.optdc*). Die Konfiguration *MIN1DC2*, bestehend aus diesem Spannungswandler und der PMC-Konfiguration *MIN1*, weist somit insgesamt die geringste Leistungsaufnahme von $6,95 \mu\text{W}$ auf.

Bewertung der Komponenten für den inaktiven Modus Auf Basis der Systemkonfigurationen im Entwurfsraum kann eine Bewertung des Systems aus PMC und Spannungswandler für den inaktiven Modus erfolgen. So können einzelne PMC gleicher Funktionalität hinsichtlich ihrer Leistungsaufnahme miteinander verglichen werden. Zudem können diese

Einzelkomponenten in verschiedenen Systemkonfigurationen kombiniert werden. Deren summierte Leistungsaufnahme wiederum dient als Basis für die Effizienzbetrachtung verschiedener Spannungswandler. Die daraus resultierenden Leistungsaufnahmen der Spannungswandler bei Versorgung der PMC entsprechen der Leistungsentnahme aus einer (zunächst idealen) Batterie. Hier wird deutlich, dass die Wahl des Spannungswandlers einen signifikanten Einfluss auf die Leistungsaufnahme des Systems hat. So hat in diesem Fall die Wahl des Spannungswandlers einen höheren Einfluss auf die Leistungsaufnahme als die Wahl der PMC. Auf Basis der Komponentenmodelle und dem in diesem Unterkapitel definierten Entwurfsraum der Systemkonfigurationen konnte die Konfiguration *MIN1DC2* bestehend aus dem SoC *nRF51822*, dem Lichtsensor *OPT3001*, dem Umgebungssensor *BME280*, dem Beschleunigungssensor *LIS2DE* und dem Spannungswandler *TPS62243* als die Konfiguration mit der geringsten Leistungsaufnahme im inaktiven Modus ermittelt werden.

Die Analyse für die aktiven Modi des Systems erfolgt im folgenden Abschnitt 6.5. Die Ermittlung der aktiven und inaktiven Modi sowie die Auswahl der Batterie hängt von der Anwendung des Systems ab und erfolgt erst nach der Definition des Tastgrads in Abschnitt 6.6.

6.5 Task

Die Tasks eines System bilden die Anwendung des Sensorknotens ab und bestimmen somit die aktiven Modi der Komponenten. Das System wird durch den zuvor definierten Entwurfsraum der Systemkonfigurationen abgebildet. Die zu definierenden Task müssen dabei die Anwendung eines multimodalen Umgebungssensorknotens abbilden. Eine solche Anwendung umfasst typischerweise die Abfrage der Sensoren durch den Mikrocontroller, eine Verarbeitung der Daten und das Senden der Daten über eine drahtlose Schnittstelle. Im Folgenden werden die Parameter der Taskkonfiguration und die Generierung der Tasks durch das Modell beschrieben. Anschließend erfolgt die Beschreibung der generierten Parameter und die darauf basierende Bewertung der Task- und Systemkonfigurationen für den aktiven Modus.

Definierte Parameter Der in Abbildung 6.16 schematisch dargestellte Ablauf eines Tasks ist entsprechend der typischen Anwendung eines Sensorknotens für die einzelnen Komponententypen vom Modell fest definiert und sieht vor, dass zunächst alle in der Taskkonfiguration aktivierten Sensoren abgefragt werden. So beginnt der Task mit der Transition des Mikrocontrollers aus dem inaktiven Modus in den aktiven Modus. Anschließend werden die Sensoren abgefragt. Jede Abfrage besteht aus dem Senden eines Kommandos an den Sensor zur Aktivierung der Messung, der Messung durch den Sensor und der Übertragung der Messdaten an den Mikrocontroller. Die Dauer der Übertragung wird dabei von der Datenbreite des Kommandos bzw. der Messdaten und der Schnittstelle zwischen dem Mikrocontroller und den Sensoren bestimmt. Da sich die Komponenten für

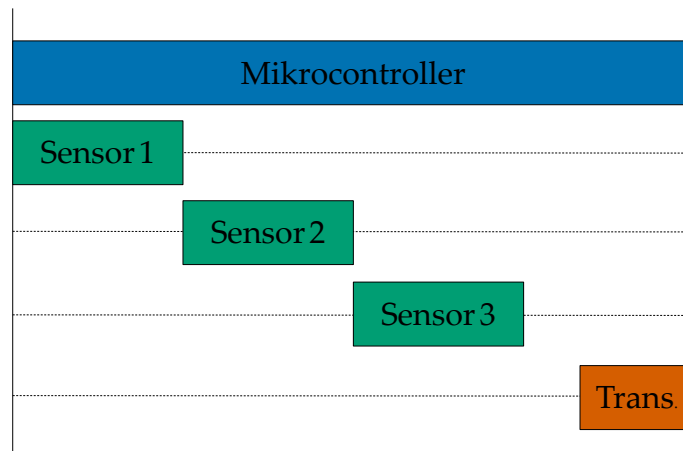


Abbildung 6.16.: Schematischer Ablauf eines Tasks

die Dauer der Datenübertragung in einem aktiven Zustand befinden müssen, addieren diese sich zu den aktiven Zeiten der Komponenten. Anschließend erfolgt die Datenverarbeitung auf dem Mikrocontroller und die Übertragung der Daten an den Transceiver. Dieser sendet die Daten, wobei die Dauer der Funkübertragung von der Breite der Eingangsdaten und dem Protokolloverhead abhängt. Abschließend erfolgt die Transition der Komponenten vom aktiven in den inaktiven Zustand und der Task ist beendet. Der Ablauf eines Tasks wird vom Modell automatisch für die in der Systemkonfiguration enthaltenen Komponententypen erstellt. Die Tasks können dann, entsprechend der Varianten der Systemkonfigurationen im Entwurfsraum, mit verschiedenen Komponenten und Modi simuliert werden.

Durch die in Tabelle 6.16 aufgeführten Taskkonfigurationen werden 4 mögliche Tasks des Systems definiert. Der Parameter *devtypesactive* legt dabei fest, welche Komponententypen im Task aktiviert werden. Die fünf Stellen entsprechen den Komponententypen Umgebungslichtsensor, Feuchtigkeitssensor, Temperatursensor, Beschleunigungssensor und Transceiver. Für den Komponententyp Mikrocontroller gibt es keine Option, da dieser in jedem Task aktiv sein muss. Der SoC wird durch die getrennten Komponententypen Mikrocontroller und Transceiver beschrieben. In den Tasks *T1* und *T3* sind alle Komponententypen aktiviert. In den Tasks *T2* und *T4* ist der Transceiver nicht aktiviert. Der Parameter *uccycles* legt fest, wie viele Taktzyklen der Mikrocontroller für die Ausführung eines Algorithmus benötigt. Typischerweise erfolgt eine Überprüfung, ob ein Messwert sich geändert hat und dementsprechend gesendet werden muss. Die aktive Zeit des Mikrocontrollers wird an dieser Stelle pauschal mit 160000 Taktzyklen abgeschätzt, da an diesem Punkt im Entwurfsablauf noch keine Implementierung für die Zielarchitektur vorliegt. Mit der im Komponentenmodell hinterlegten Taktrate des SoC von 16 MHz entspricht

Tabelle 6.16.: Taskkonfiguration

Parameter	Task T1	Task T2	Task T3	Task T4
<i>devtypesactive</i>	$\begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$	$\begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 0 \end{pmatrix}$	$\begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$	$\begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 0 \end{pmatrix}$
<i>uccycles</i>	160.000	160.000	160.000	160.000
<i>ucsleep</i>	1	1	0	0
<i>falg</i>	$\sum_{i=1}^{N_{SEN}} N_{Si}$	$\sum_{i=1}^{N_{SEN}} N_{Si}$	$\sum_{i=1}^{N_{SEN}} N_{Si}$	$\sum_{i=1}^{N_{SEN}} N_{Si}$

dieses einer Dauer von 10 ms. Die Funktion der Datenoperation des Mikrocontrollers *falg* ist so definiert, dass der Mikrocontroller die Messdaten der Sensoren N_{Si} sammelt und gegebenenfalls weitergibt. Ob die Messwerte gesendet werden, ist durch die Aktivierung des Transceivers in *devtypesactive* definiert. Als weiterer Parameter geht *ucsleep* in die Taskkonfiguration ein. Dieser legt fest, ob der Mikrocontroller während der Messung der Sensoren in den inaktiven Zustand wechselt. Der Wechsel erfolgt jedoch nur, wenn die im Komponentenmodell hinterlegte Nutzenzeit T_{BE} kleiner ist als die Dauer der Messung. Wenn der Parameter *ucsleep* nicht gesetzt wird, bleibt der Mikrocontroller für den gesamten Task im aktiven Modus.

Somit können 4 verschiedene Tasks untersucht werden, die sich in der Aktivierung des Transceivers und der Modi der SoC unterscheiden. Wie in Gleichung 6.13 aufgeführt ergeben sich daraus 320 Varianten im Entwurfsraum.

$$\begin{aligned}
 N_{EWR3,T} &= N_{EWR3,PMC} \cdot N_{DC} \cdot N_T & (6.13) \\
 &= 40 \cdot 2 \cdot 4 \\
 &= 320
 \end{aligned}$$

Generierte Parameter Aus den Taskkonfigurationen, dem im Modell definierten Ablauf der Tasks und dem Entwurfsraum der Systemkonfigurationen werden verschiedene Parameter zur Auswertung generiert. Diese umfassen die Dauer und Nutzlast des Tasks, die Leistungs- und Energieaufnahme der PMC und Spannungswandler, die anteilige Energieaufnahme der Komponententypen und der für den Task optimale Spannungswandler. Die Darstellung der Ergebnisse erfolgt aus Gründen der Übersicht exemplarisch.

In Abbildung 6.17 ist die Energieaufnahme der Tasks *T3* und *T4* (*task.pmcenergy*) über alle 40 Systemkonfigurationen des Entwurfsraums *EWR3* dargestellt. Entsprechend der Taskkonfigurationen überträgt Task *T3* die Messdaten und bei Task *T4* ist die Übertragung

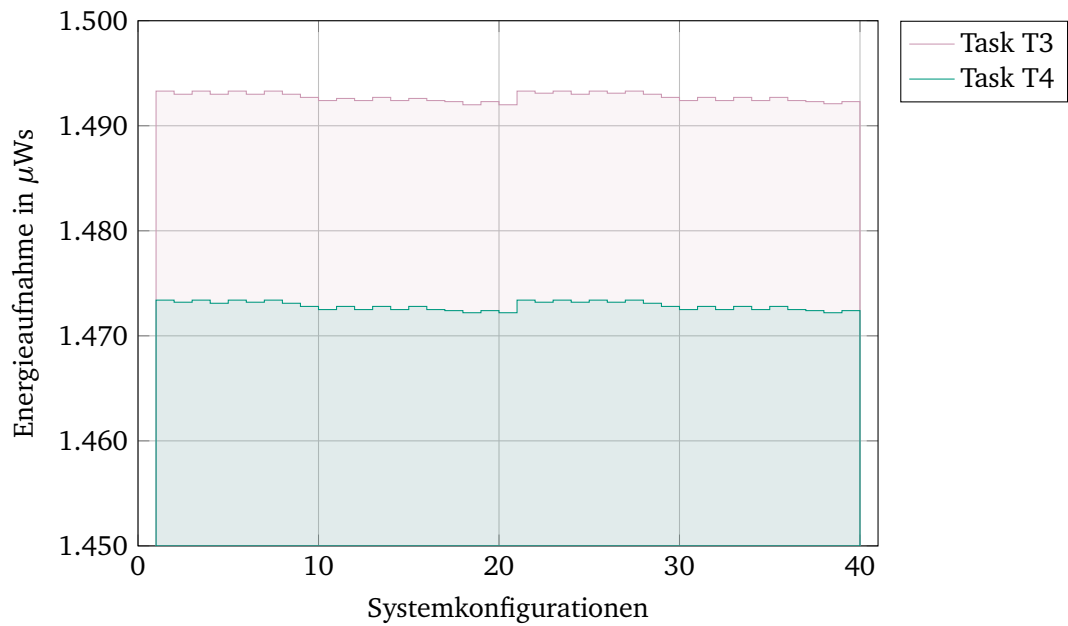


Abbildung 6.17.: Energieaufnahme der Tasks *T3* und *T4* für alle Systemkonfigurationen

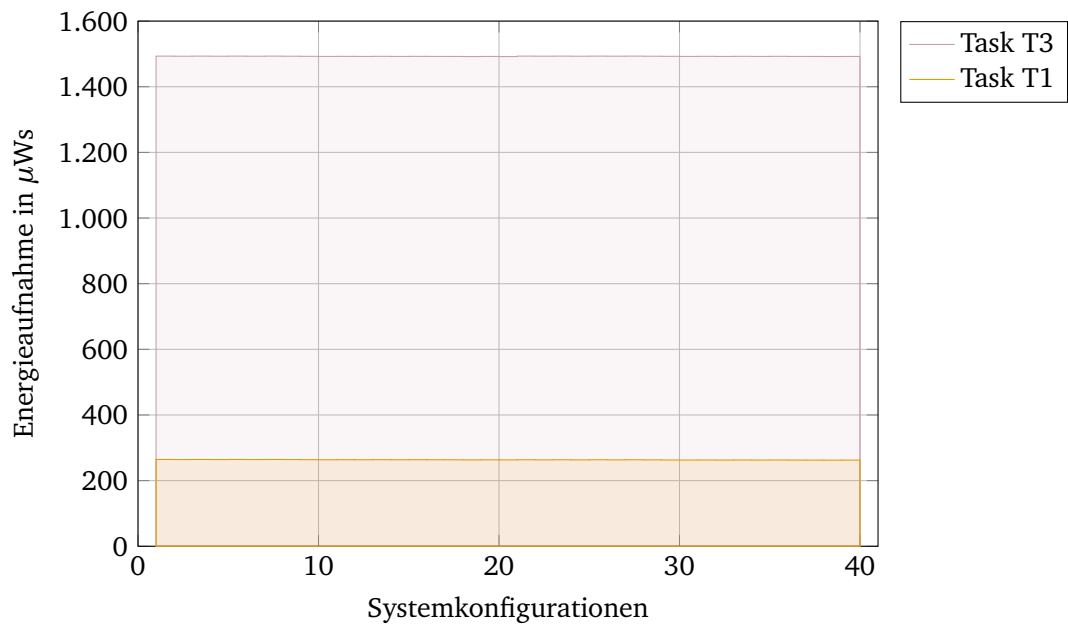


Abbildung 6.18.: Energieaufnahme der Tasks *T1* und *T3* für alle Systemkonfigurationen

deaktiviert. Dieses resultiert in einer etwas höheren Energieaufnahme von Task *T3*. Bei beiden Tasks ist der Mikrocontroller dauerhaft im aktiven Zustand. In Abbildung 6.18 ist die Energieaufnahme der Tasks *T1* und *T3* über alle 40 Systemkonfigurationen des Entwurfsraums *EWR3* dargestellt. Bei beiden Tasks ist die Übertragung der Messdaten aktiviert. Entsprechend der Definition von *ucsleep* ist bei Task *T3* der SoC dauerhaft im aktiven Zustand. Bei Task *T1* wechselt dieser während der Messungen der Sensoren in einen inaktiven Zustand, wenn die Nutzenzeit T_{BE} unterhalb der Messdauer liegt. Dieses führt bei Task *T1* zu einer deutlich reduzierten Energieaufnahme.

Zur genaueren Analyse wird für jeden Task ermittelt, bei welcher Systemkonfiguration dieser die kleinste Energieaufnahme aufweist. Diese sind in Tabelle 6.17 aufgeführt.

Tabelle 6.17.: *Kombinationen aus Task und Systemkonfigurationen mit der niedrigsten Energieaufnahme*

Task	Trans.	<i>ucsleep</i>	Dauer	Leistungsaufn.	Energieaufn.	bei Konf.
T1	1	1	120.016 μ s	2,18 mW	262,25 μ Ws	K40
T2	0	1	119.332 μ s	2,03 mW	242,36 μ Ws	K40
T3	1	0	119.982 μ s	12,44 mW	1.492,02 μ Ws	K20
T4	0	0	119.308 μ s	12,34 mW	1.472,15 μ Ws	K20

Tabelle 6.18.: *Kombinationen aus Task und Systemkonfigurationen mit der höchsten Energieaufnahme*

Task	Trans.	<i>ucsleep</i>	Dauer	Leistungsaufn.	Energieaufn.	bei Konf.
T1	1	1	119.986 μ s	2,20 mW	264,44 μ Ws	K01
T2	0	1	119.312 μ s	2,05 mW	244,57 μ Ws	K01
T3	1	0	120.002 μ s	12,44 mW	1.493,32 μ Ws	K21
T4	0	0	119.318 μ s	12,35 mW	1.473,43 μ Ws	K21

So liegt die minimale Energieaufnahme für die Tasks *T1* und *T2* (Mikrocontroller wechselt Zustände) bei Konfiguration *K40* vor. Für die Tasks *T3* und *T4* (Mikrocontroller dauerhaft aktiv) liegt die minimale Energieaufnahme bei Konfiguration *K20* vor. Demgegenüber ist in Tabelle 6.18 für jeden Task die Systemkonfiguration mit maximaler Energieaufnahme aufgeführt. Hier wird ersichtlich, dass die maximale Energieaufnahme für die Tasks *T1* und *T2* bei Konfiguration *K01* und für die Tasks *T3* und *T4* bei Konfiguration *K21* vorliegt. Zur genaueren Betrachtung werden die Konfigurationen *K01* und *K40* in Tabelle 6.19 hinsichtlich der Komponenten, deren Modi und der anteiligen Energieaufnahme aufgeschlüsselt.

6. Modellgestützter Entwurf eines Sensorknotens

Tabelle 6.19.: Systemkonfigurationen K01 und K40 im EWR3 mit maximaler und minimaler Energieaufnahme der PMC für den Task T1

Komptyp.		K01	K40
μC	Komp.	nRF51822	nRF51822
	M_A	CPU from flash	CPU from flash
	M_I	ON	OFF
	$E_{\mu\text{C}}$	258,32 μWs	257,41 μWs
LUX	Komp.	OPT3001	MAX44009
	M_A	Active Full-scale lux	Active
	M_I	Shutdown Full-scale lux	Active
	E_{LUX}	1,25 μWs	0,26 μWs
HUM	Komp.	BME280	BME280
	M_A	Humidity meas.	Humidity meas.
	M_I	Standby	Sleep
	E_{HUM}	4,17 μWs	4,16 μWs
TEMP	Komp.	BME280	BME280
	M_A	Temperature meas.	Temperature meas.
	M_I	Standby	Sleep
	E_{TEMP}	in E_{HUM}	in E_{HUM}
ACC	Komp.	LIS2DE	LIS2DE
	M_A	Normal	Low-power
	M_I	Power-down	Power-down
	E_{ACC}	0,70 μWs	0,44 μWs
Trans.	Komp.	nRF51822	nRF51822
	M_A	TX 0 dBm	TX 0dB
	M_I	ON	OFF
	E_{TX}	in $E_{\mu\text{C}}$	in $E_{\mu\text{C}}$
PMC	E_{T1}	264,44 μWs	262,25 μWs

Die Konfigurationen K40 mit der niedrigsten Energieaufnahme umfasst im Vergleich zur Konfiguration K1 den Umgebungslichtsensor *MAX44009* und nutzt inaktive Modi mit geringerer Leistungsaufnahme. Jedoch sind die Differenzen in der Energieaufnahme der einzelnen Komponenten (*task.pmcenergyratio*) relativ gering und es wird deutlich, dass der SoC den größten Anteil an der Energieaufnahme ausmacht.

Auf Basis der Leistungsaufnahmen der PMC für die Kombinationen aus Task und Systemkonfiguration kann die Effizienz der Spannungswandler ermittelt werden, die vom Arbeitspunkt aus Eingangsspannung, Ausgangsspannung und Ausgangsstrom abhängt (vgl. Unterabschnitt 5.5.5). Die Ausgangsspannung ist die Versorgungsspannung des Sys-

tems von 1,8 V und der Ausgangsstrom der Spannungswandler entspricht der ermittelten Leistungsaufnahme der PMC bei dieser Versorgungsspannung. Die Eingangsspannung entspricht der Batteriespannung. Da diese vom Zustand der Batterie abhängt, wird diese auch hier zunächst mit der Nennspannung der Batterien von 3,0 V angenommen. So wird die Eingangsleistung der Spannungswandler in μs Auslösung für jeden Task und jede Systemkonfiguration berechnet. Die darüber gemittelte durchschnittliche Leistungsaufnahme ($dc_{\text{meanpower}}$) ergibt, multipliziert mit der Dauer des Tasks, die Energieaufnahme (dc_{energy}) des Tasks. Diese ist beispielhaft für den Task *T1* für beide Spannungswandler und alle 40 Systemkonfigurationen in Abbildung 6.19 dargestellt. Hier zeigt sich, dass der Spannungswandler *TPS62202* für die Energieaufnahme von Task *T1* eine höhere Effizienz (93 %) aufweist, als der Spannungswandler *TPS62243* (65 %). Für die Tasks *T2*, *T3* und *T4* ergibt sich eine ähnliche Abstufung.

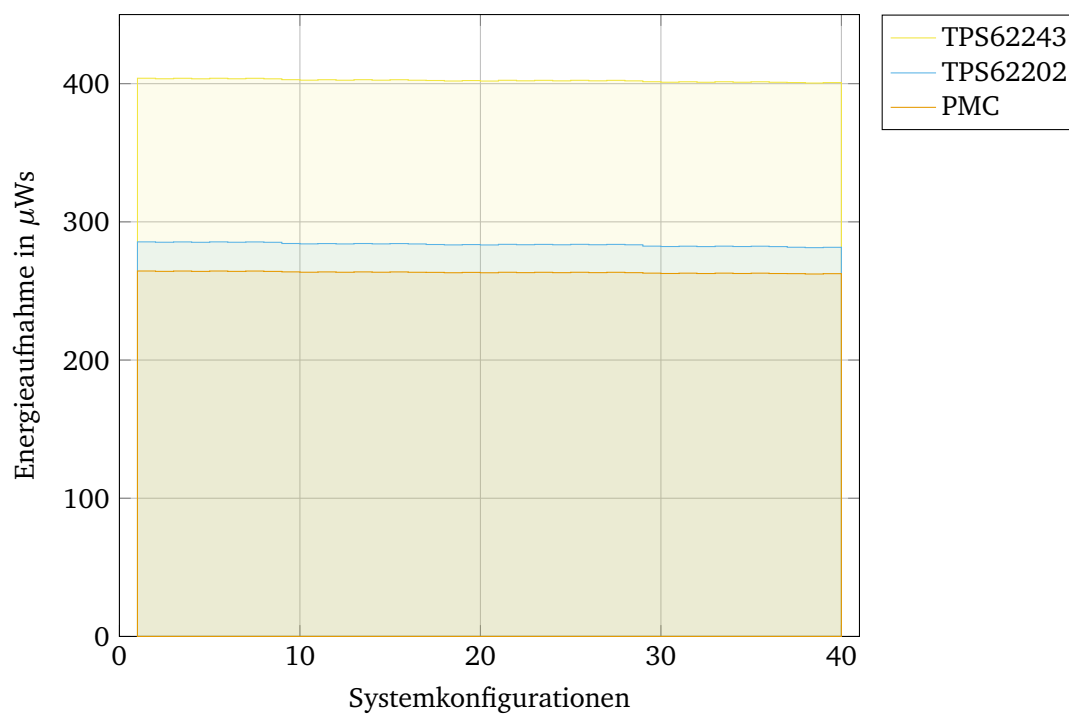


Abbildung 6.19.: Durchschnittliche Energieaufnahme von Task *T1* für die PMC und Spannungswandler für alle Systemkonfigurationen

Bewertung der Komponenten für den aktiven Modus Auf Basis der Taskkonfigurationen kann eine Bewertung des Systems aus PMC und Spannungswandler für den aktiven Modus und somit für die spätere Anwendung erfolgen. Es können einzelne

PMC gleicher Funktionalität und deren Modi hinsichtlich ihrer Energieaufnahme für den definierten Anwendungsfall verglichen werden. Neben der Wahl der Komponenten hat zudem die Wahl der Modi einen signifikanten Einfluss auf die Energieaufnahme, was durch den Parameter *ucsleep* und die Analyse der im Komponentenmodell hinterlegten Nutzenzeit T_{BE} für die Transitionen verdeutlicht wurde. Die Option *ucsleep* sollte daher in der Taskkonfiguration gesetzt werden (Tasks *T1* und *T2*), so dass das Modell den Modus abhängig von der T_{BE} wechselt. Für die Tasks *T1* und *T2* weist die Konfiguration *K40* die geringste Energieaufnahme auf. Diese besteht aus dem SoC *nRF51822*, dem Lichtsensor *MAX44009*, dem Umgebungssensor *BME280* und dem Beschleunigungssensor *LIS2DE*. Mit der ermittelten Leistungsaufnahme der PMC konnte zudem die Effizienz verschiedener Spannungswandler analysiert werden. Hier hat sich gezeigt, dass der Spannungswandler *TPS62202* für alle 4 Tasks und alle 40 Systemkonfigurationen effizienter ist als der Spannungswandler *TPS62243*. Im Vergleich zu der Bewertung für den inaktiven Modus in Abschnitt 6.4 zeigt sich, dass sowohl der Spannungswandler als auch die Komponente des Lichtsensors in den jeweiligen Systemkonfigurationen voneinander abweichen. Welche Konfiguration im simulierten Betrieb am effizientesten ist hängt somit von der Ausführungsfrequenz der Tasks ab, die im folgenden Abschnitt 6.6 untersucht wird.

6.6 Tastgrad

Der Tastgrad des Systems bestimmt über die Ausführungsfrequenz der zuvor definierten Tasks das Verhältnis von aktiven zu inaktiven Modi und ist somit ein entscheidender Faktor für die Leistungsfähigkeit und die Leistungsaufnahme des Systems. Für die Betrachtung des Tastgrads wird zunächst der Entwurfsraum einschränkt, da die zuvor erfolgte Bewertung der Tasks ergeben hat, dass es effizienter ist, wenn der Mikrocontroller während des Tasks in den inaktiven Modus wechselt (*ucsleep*=1). Somit werden die zuvor betrachteten vier Tasks auf die beiden Tasks *T1* und *T2* reduziert, welche diesen Wechsel erlauben. Im Folgenden wird zunächst die Ermittlung der Parameter beschrieben. Anschließend erfolgt die Beschreibung der generierten Parameter und abschließend die Bewertung des Systems.

Definierte Parameter Ein wesentlicher zu definierender Parameter ist die Ausführungsfrequenz der Tasks (*samplef*). Der zu simulierende Bereich der Ausführungsfrequenz wird in diesem Fall über die Anwendung des multimodalen Umgebungssensorknotens bestimmt. In Tabelle 6.20 ist die Auswertung von drei verschiedenen gemessenen Aktivitätsprofilen eines Präsenzsensors in Büroräumen ausgeführt. Jede aufgeführte Aktivität entspricht dem Senden von Messwerten bei einer Veränderung der Lichtintensität.

Über alle Messreihen ergibt sich im Mittel eine Aktivität alle 133 s mit einer Standardabweichung von 708 s. Das entspricht einer Ausführungsfrequenz von 0,0075 Hz bei einer Standardabweichung von 0,0014 Hz. Der minimale Abstand zwischen zwei Aktivitäten lag bei 0,63 s was einer Frequenz von 1,5873 Hz entspricht. Somit wird für die Ausführungsfrequenz des Systems ein Bereich zwischen 0 Hz und 10 Hz simuliert. Für das

Tabelle 6.20.: *Aktivität der Konstantlichtregelung*

Messung	1	2	3
Dauer der Aufzeichnung	6 h	21 h	6 h
Anzahl der Aktivitäten	311	425	175
Min. Intervall	1,97 s	1,98 s	0,63 s
Max. Intervall	7.224 s	7.261 s	11.966 s
Mean	70 s	181 s	132 s
Std.	436 s	667 s	1.089 s

Ausführungsverhältnis der beiden Tasks (*taskratio*) wird der Bereich von 1 zu 0 bis 0 zu 1 mit einer Schrittweite von 0,1 simuliert. Die definierten Parameter des Tastgrads sind in Tabelle 6.21 aufgeführt.

Für das Systemverhalten bedeutet dieses, dass beispielsweise bei einer Ausführungsfrequenz von 1 Hz einmal pro Sekunde ein Task ausgeführt wird. Bei einem Ausführungsverhältnis von jeweils 0,5 werden abwechselnd Task *T1* und Task *T2* ausgeführt. In einem Intervall von 2 s wird demnach jeder Task einmal ausgeführt. Beträgt das Verhältnis 0,9 zu 0,1 wird in einem Intervall von 10 s neunmal Task *T1* und einmal Task *T2* ausgeführt. Bei einem Verhältnis von 0 zu 1 wird nur Task *T2* ausgeführt. Bei einem Verhältnis von 1 zu 0 nur Task *T1*. In der Zeit im Intervall, in der kein Task ausgeführt wird, befinden sich die Komponenten jeweils im inaktiven Zustand. Der jeweils effizienteste inaktive Modus wird vom Modell, unabhängig vom Task und der Systemkonfiguration, durch einen Vergleich zwischen der Dauer im inaktiven Modus und der jeweiligen Nutzenzeit T_{BE} der Komponente automatisch ermittelt. Nach diesem Schema kann für jede Kombination aus Ausführungsfrequenz und Ausführungsverhältnis die mittlere Leistungsaufnahme bestimmt werden.

Tabelle 6.21.: *Parameter des Tastgrads*

Parameter	Wert
<i>samplef</i>	10^{-6} Hz bis 10 Hz
<i>taskratio</i>	$\begin{pmatrix} 0,0 & 0,1 & 0,2 & 0,3 & \dots & 1,0 \\ 1,0 & 0,9 & 0,8 & 0,7 & \dots & 0,0 \end{pmatrix}$

Generierte Parameter Laut der Konfiguration des Tastgrads in Tabelle 6.21 werden die Tasks mit einer Frequenz zwischen 10^{-6} Hz und 10 Hz ausgeführt. Als Eingangsgröße für das Modell wird für diesen Bereich eine logarithmische Skala erstellt. Anhand der in

Abschnitt 6.5 ermittelten Dauer der Tasks wird zuvor die maximal mögliche Ausführungsfrequenz ermittelt. Somit reduzieren sich die Ausführungsfrequenzen auf einen Bereich zwischen 10^{-6} Hz und 7,94 Hz, wobei die Skala 70 verschiedene Werte umfasst. Aus der Anzahl der Systemkonfigurationen $N_{\text{EWR,SYS}}$, den zwei Tasks sowie den verschiedenen Ausführungsverhältnissen N_{TR} und Ausführungsfrequenzen N_{TF} ergibt sich ein Entwurfsraum mit 61.600 Varianten. Für jede dieser Varianten kann die mittlere Leistungsaufnahme der PMC ermittelt werden. Zudem kann die mittlere Leistungsaufnahme für die beiden Spannungswandler bestimmt werden, wodurch sich der Entwurfsraum verdoppelt. Auf dieser Basis kann abschließend die Laufzeit des Systems für die vier verschiedenen Batterien im Entwurfsraum bestimmt werden. Somit beträgt die Anzahl der Varianten im Entwurfsraum *EWR3* 492.800.

$$\begin{aligned} N_{\text{EWR3}} &= N_{\text{EWR3,SYS}} \cdot N_{\text{T}} \cdot N_{\text{TF}} \cdot N_{\text{TR}} & (6.14) \\ &= 320 \cdot 2 \cdot 70 \cdot 11 \\ &= 492.800 \end{aligned}$$

Entsprechend der Größe des Entwurfsraumes kann nur eine selektive Darstellung der Ergebnisse erfolgen. Anhand der Simulationen wurde ermittelt, dass der Einfluss des Ausführungsverhältnisses der beiden Tasks im Vergleich zur Ausführungsfrequenz nicht signifikant ist. Somit wird im Folgenden nur das Ausführungsverhältnis 1 zu 0 für T1 und T2 betrachtet. Es wird daher nur der Task T1 ausgeführt. Der inaktive Modus wird für jede Ausführungsfrequenz automatisch anhand der Nutzenzeit T_{BE} ausgewählt. Dieses Vorgehen wird im Folgenden als *IdleOpt*-Option bezeichnet. In Abbildung 6.20 ist zum Vergleich die Leistungsaufnahme der PMC für eine Ausführungsfrequenz von 1 Hz jeweils mit und ohne *IdleOpt*-Option dargestellt. Ohne diese Option wird der inaktive Modus der Systemkonfiguration verwendet, wobei in den Konfigurationen *K01* bis *K20* für den SoC der inaktive Modus *ON* definiert ist und in den Konfigurationen *K21* bis *K40* der inaktive Modus *OFF*. Mit der *IdleOpt*-Option wird der inaktive Modus anhand der Nutzenzeit ausgewählt, so dass auch für die Konfigurationen *K01* bis *K20* der inaktive Modus *OFF* gesetzt wird. Die Differenzen in der Leistungsaufnahme sind bei aktivierter *IdleOpt*-Option somit nur auf die Tasks zurückzuführen. Der Einfluss der Ausführungsfrequenz auf die Leistungsaufnahme der PMC (*cycle.pmcmeanpower*) über alle 40 Systemkonfigurationen ist in Abbildung 6.21 exemplarisch für drei verschiedene Größenordnungen dargestellt.

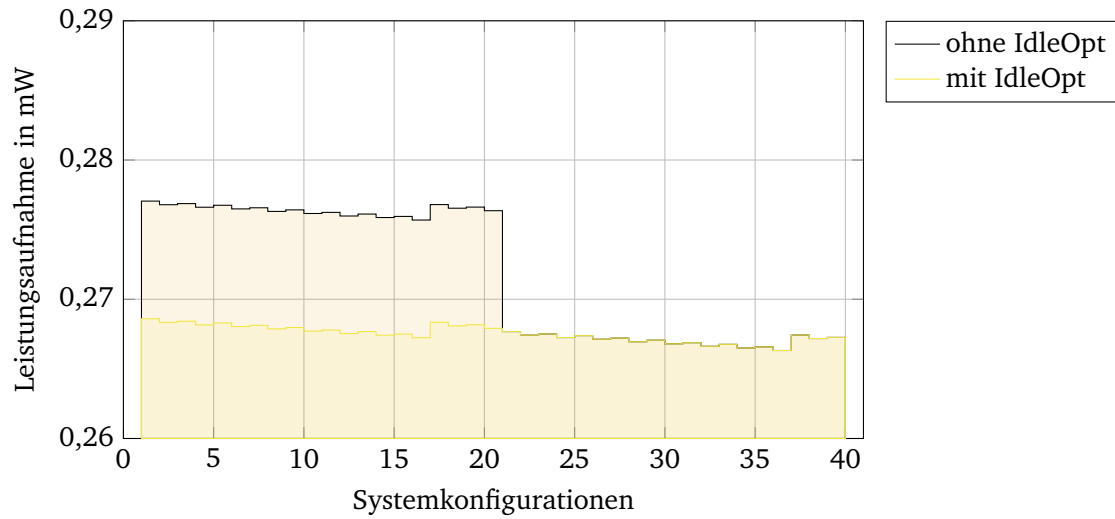


Abbildung 6.20.: Leistungsaufnahme der PMC bei Ausführung von Task T1 über alle Systemkonfigurationen für eine Ausführungsfrequenz von 1 Hz

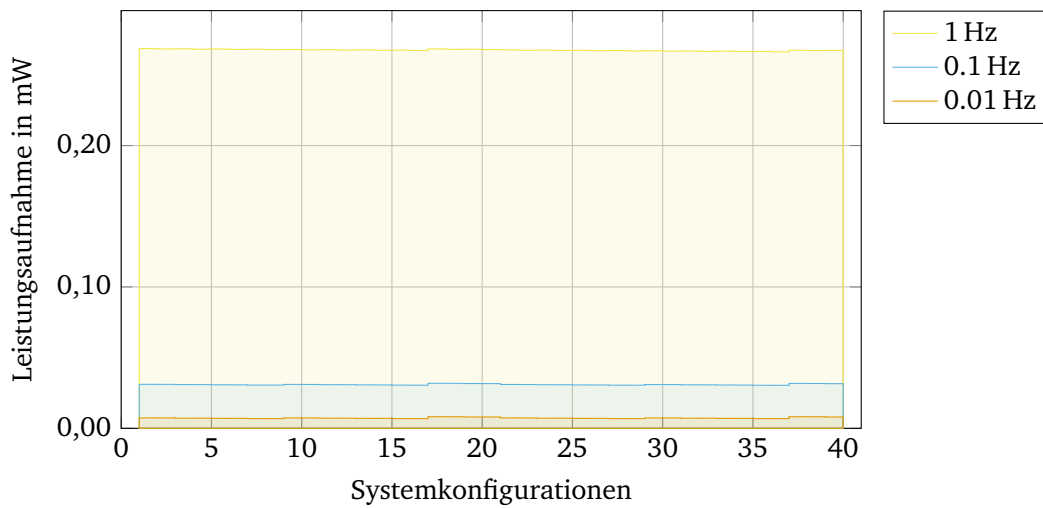


Abbildung 6.21.: Leistungsaufnahme der PMC bei Ausführung von Task T1 über alle Systemkonfigurationen für verschiedene Ausführungsfrequenzen

6. Modellgestützter Entwurf eines Sensorknotens

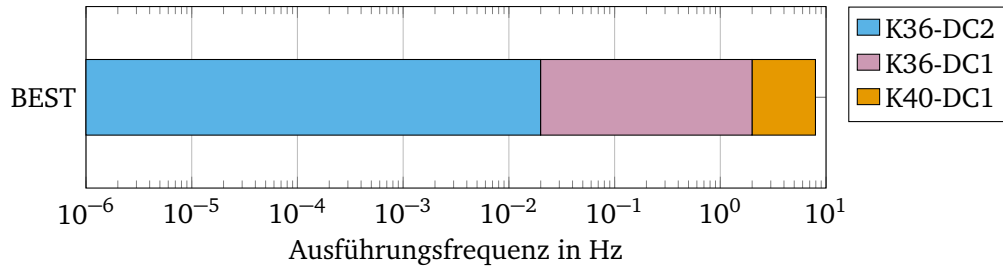


Abbildung 6.22.: Systemkonfigurationen aus PMC und Spannungswandler mit der niedrigsten Leistungsaufnahme.

Da mittels der Simulation die effizienteste Konfiguration für das System bestimmt werden soll, wurde für jede Ausführungsfrequenz die Kombination aus PMC und Spannungswandler mit der jeweils kleinsten Leistungsaufnahme ermittelt. Diese sind in Abbildung 6.22 dargestellt. Hier zeigt sich, dass bis zu einer Ausführungsfrequenz von ca. 0,02 Hz die Kombination aus Konfiguration K36 und dem Spannungswandler TPS62243 (DC2) die kleinste Leistungsaufnahme aufweist. Zwischen 0,02 Hz und ca. 2 Hz ist es die Kombination aus Konfiguration K36 und dem Spannungswandler TPS62202 (DC1). Zwischen 2 Hz und der maximalen Ausführungsfrequenz von 7,94 Hz hat die Kombination aus K40 und dem Spannungswandler TPS62202 (DC1) die kleinste Leistungsaufnahme. Da die Konfiguration K36 den Umgebungslichtsensor OPT3001 und die Konfiguration K40 hingegen den Umgebungslichtsensor MAX44009 beinhaltet, unterscheiden sich die Konfigurationen hinsichtlich ihrer Komponenten, so dass die Wahl der Konfiguration zur Entwurfszeit erfolgen muss.

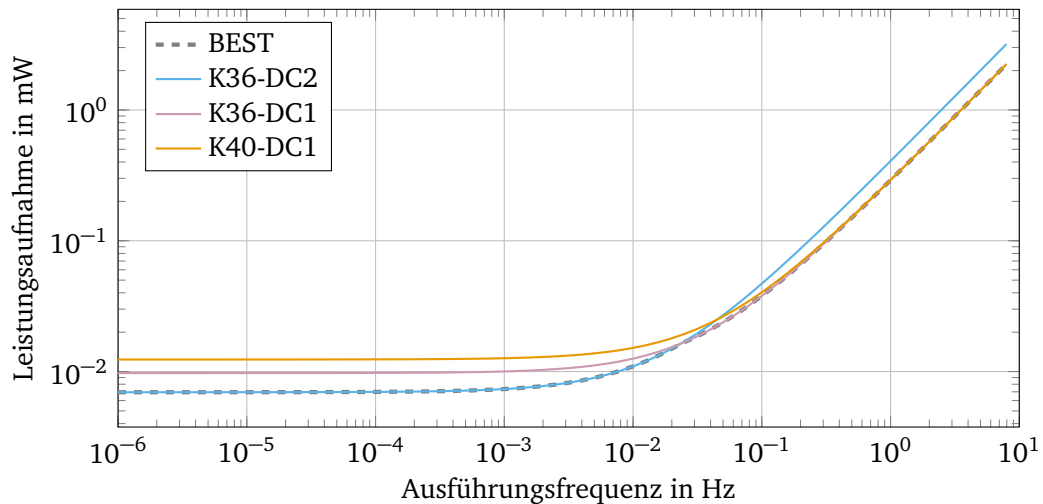


Abbildung 6.23.: Leistungsaufnahme des Systems für verschiedene Konfigurationen

Zur weiteren Einschränkung des Entwurfs- bzw. Lösungsraums werden die Kombinationen *K36-DC2*, *K36-DC1* und *K40-DC1* mit der theoretisch optimalen Konfiguration verglichen. In Abbildung 6.23 ist die Leistungsaufnahme der theoretisch optimalen Konfiguration im Vergleich zur Leistungsaufnahme der spezifischen Kombinationen dargestellt. Wie zuvor ermittelt (vgl. Abbildung 6.22) zeigt sich, dass die Kombinationen jeweils in verschiedenen Frequenzbereichen mit der optimalen Konfiguration übereinstimmen. Zur quantitativen Beurteilung wird der *Root Mean Square Error* (RMSE) betrachtet. Hier zeigt sich, dass die Kombination *K36-DC1* mit einem RMSE von 0,0023 mW die geringste Abweichung aufweist. Der RMSE von *K40-DC1* liegt etwas darüber bei 0,0043 mW. Der von *K36-DC2* liegt bei 0,1863 mW. Die Kombinationen mit dem Spannungswandler *DC1* (TPS62202) weisen somit einen kleineren RMSE auf.

Auf Basis der durchschnittlichen Leistungsaufnahme des Systems kann im Folgenden die **Laufzeit des Systems** für verschiedene Energiespeicher bestimmt werden. Dieses kann anhand des Basismodells und des erweiterten Modells der Batterie erfolgen. In Tabelle 6.22 ist dargestellt, für welche Batterien des Entwurfsraums welche Modelle generiert werden können.

Tabelle 6.22.: Mögliche Batteriemodelle

Typ	Basis	Erweitert
<i>Renata CR2032</i>	X	-
<i>Renata CR2025</i>	X	-
<i>Energizer CR2032</i>	-	X
<i>Maxell CR2032H</i>	X	X

In einem ersten Schritt wird anhand des **Basismodells** die Laufzeit für die Batterien *Renata CR2032*, *Renata CR2025* und *Maxell CR2032H* bestimmt. Die Laufzeit des Systems T_{SYS} berechnet sich aus dem Batteriestrom I_{BAT} und den im Komponentenmodell hinterlegten Parametern Peukert-Kapazität C_{PK} und Peukert Zahl k (vgl. Gleichung 5.9). Der Batteriestrom ergibt sich aus der mittleren Leistungsaufnahme des Spannungswandlers (*cycle.dcpower*) geteilt durch die im Komponentenmodell hinterlegte Nennspannung der Batterie. Die Laufzeiten des Systems für die minimale und maximale Ausführungsfrequenz sind in Tabelle 6.23 aufgeführt.

Tabelle 6.23.: Laufzeit des Systems bei maximaler und minimaler Ausführungsfrequenz

Typ	K36-DC2	K36-DC1	K40-DC1
<i>Renata CR2032</i>	220 - 104.786 h	314 - 74.645 h	315 - 58.704 h
<i>Renata CR2025</i>	156 - 76.830 h	222 - 54.616 h	223 - 42.888 h
<i>Maxell CR2032H</i>	483 - 109.388 h	659 - 81.168 h	661 - 65.705 h

Hier zeigt sich, dass mit der Batterie *Maxell CR2032H* die höchsten Laufzeiten erreicht werden. Im Vergleich der Konfigurationen zeigt sich, dass die Konfiguration *K36-DC2* die höchste Laufzeit bei minimaler Ausführungsfrequenz erreicht. Die Konfigurationen *K36-DC1* und *K40-DC1* hingegen weisen bei maximaler Ausführungsfrequenz die höchste Laufzeit auf.

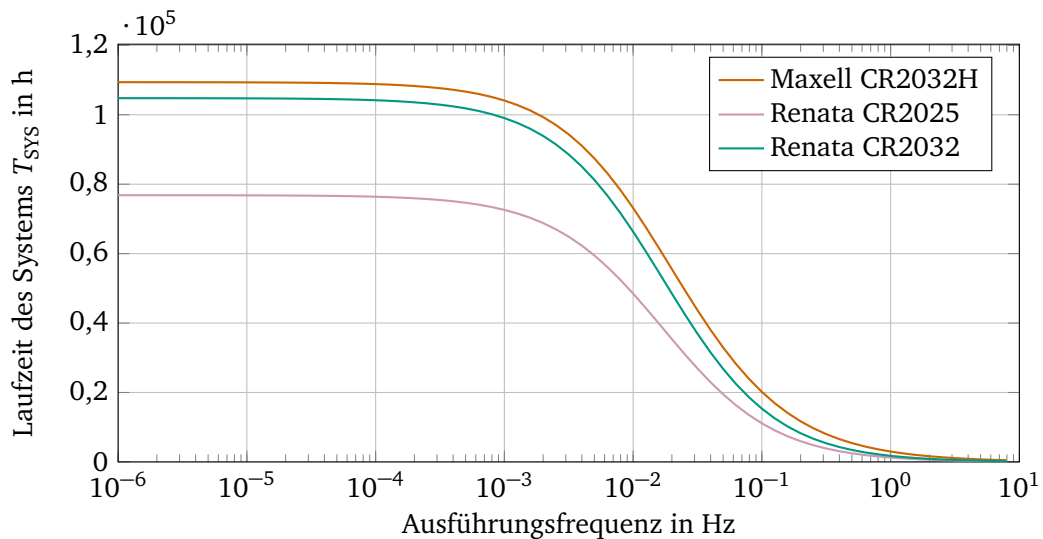


Abbildung 6.24.: Laufzeit des Systems anhand das Basismodells für die Kombination *K36-DC2* für verschiedene Batterien

Zum grafischen Vergleich der Batterien ist in Abbildung 6.25 die Laufzeit des Systems für die Batterien *Maxell CR2032H* (220 mAh), *Renata CR2025* (165 mAh) und *Renata CR2032* (235 mAh) über den gesamten Bereich der Ausführungsfrequenzen für die Kombination *K36-DC2* dargestellt. Hier zeigt sich, dass die *Renata CR2032* im Vergleich zur *Maxell CR2032H* trotz höherer Nennkapazität eine geringere Laufzeit aufweist. Laut Simulation kann die *Maxell CR2032H* also effizienter entladen werden. Die Batterie *Renata CR2025* ermöglicht, aufgrund der geringeren Nennkapazität, eine entsprechend geringere Laufzeit. Dieses Bild zeigt sich ebenso für die Konfiguration *K36-DC1* und *K40-DC1*. Im grafischen Vergleich der Konfigurationen für die Batterie *Maxell CR2032H* (Abbildung 6.26) bestätigt sich die in Abbildung 6.22 ermittelte Frequenzabhängigkeit der jeweils optimalen Konfiguration.

Für die Batterien *Maxell CR2032H* und *Energizer CR2032* kann zudem das **erweiterte Batteriemodell** generiert werden. Dieses bildet ab, dass die Klemmenspannung während der Entladung sinkt. Dieser Effekt sorgt zum einen dafür, dass bei gleicher Leistungsentnahme der Batteriestrom ansteigt und zum anderen, dass sich der Arbeitspunkt des Spannungswandlers verschiebt und sich somit die Effizienz η ändert. In der Simulation

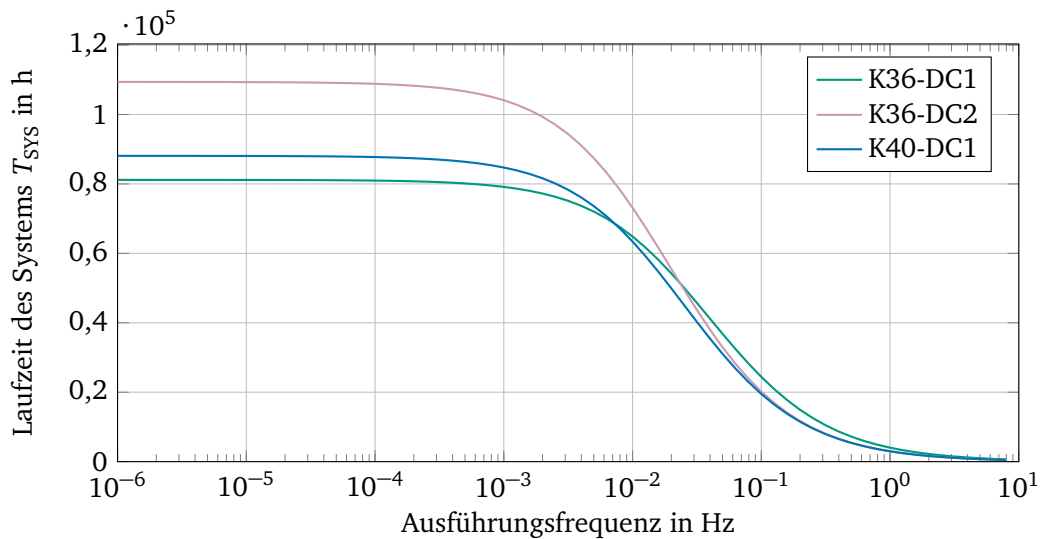


Abbildung 6.25.: Laufzeit des Systems anhand des Basismodells für die Batterie Maxell CR2032H für verschiedene Systemkonfigurationen

mit dem erweiterten Batteriemodell wird zunächst die momentane Klemmenspannung auf Basis der im Komponentenmodell hinterlegten Funktionen für den Innenwiderstand R_0 und die Leerlaufspannung U_{OC} in Abhängigkeit vom SOC berechnet (vgl. Gleichung 5.14). Mit der momentanen Klemmenspannung wird im nächsten Schritt über eine Periode dt die Entladung berechnet (vgl. Gleichung 5.13). Diese Entladung erfolgt in der Simulation so lange, wie die entladene Kapazität unterhalb der Nennkapazität liegt und die Klemmenspannung größer ist als die Entladeschlussspannung der Batterie und die minimale Eingangsspannung des Spannungswandlers. Sobald eine der drei Bedingungen nicht mehr erfüllt ist, gilt die Batterie als entladen und die maximale Laufzeit des Systems für die Konfiguration ist erreicht. In Abbildung 6.27 ist die Entladung mittels erweitertem Modell für die Kombination K36-DC2 und die Batterien Maxell CR2032H (220 mAh) und Energizer CR2032 (235 mAh) dargestellt. Die Periode dt beträgt für die Simulation 100 h. Zudem ist die Entladung der Batterie Maxell CR2032H mittels Basismodell dargestellt, so dass ein Vergleich der Modelle ermöglicht wird.

Erwartungsgemäß ermöglicht die höhere Kapazität der Energizer CR2032 im Vergleich zu der Maxell CR2032H eine etwas höhere Laufzeit des Systems. Im Vergleich der beiden Batteriemodelle für die Maxell CR2032H zeigt sich, dass die Simulation mit dem erweiterten Modell eine geringere Laufzeit ergibt. Dieses ist auf den Spannungsabfall der Batterie bei der Entladung zurück zu führen und bildet somit das reale Verhalten besser ab.

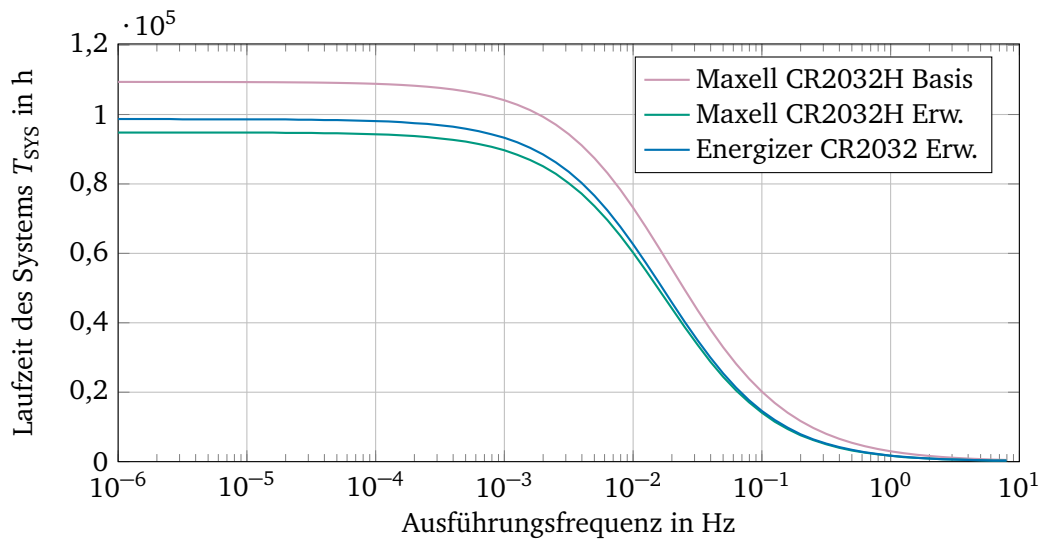


Abbildung 6.26.: Laufzeit des Systems anhand des Basismodells und des erweiterten Modells für verschiedene Batterien über den gesamten Bereich der Ausführungsfrequenzen für die Kombination *K36-DC2*

6.7 Bewertung des Systementwurfs

Die Simulation des System mit dem definierten Tastgrad ermöglicht die abschließende Evaluierung hinsichtlich der zu erwartenden Laufzeit des Systems. Da hier sämtliche Optionen hinsichtlich der Komponenten, der inaktiven Modi und der Tasks zusammen kommen, wird ein entsprechend großer Entwurfsraum generiert. Im Laufe der exemplarisch aufgeführten Auswertungen kann dieser Entwurfsraum in jedem Schritt eingeschränkt werden. Es findet daher die Fokussierung eines breiten Entwurfsraums auf einen kleinen Lösungsraum statt, welcher die laut Simulation effizienteste Implementierung für den betrachteten Anwendungsfall enthält. In diesem Fall ist dieses die Konfiguration der PMC *K36* in Kombination mit dem Spannungswandler *TPS62243 (DC2)* und der Batterie *Maxell CR2032H (B2)*. Die Effizienz einer Konfiguration definiert sich dabei als die Laufzeit eines Systems in Bezug auf dessen Leistungsfähigkeit. Wobei sich die Leistungsfähigkeit eines Sensorknotens aus der Quantität und Qualität der Messwerte sowie der Datenverarbeitung dieser Messwerte ergibt (vgl. Abschnitt 5.1). Hinsichtlich der Datenverarbeitung der Messwerte weisen alle Konfigurationen den gleichen Algorithmus (*taskconf:alg*) auf, so dass sich hier keine Unterschiede hinsichtlich der Leistungsfähigkeit ergeben.

Da im Entwurfsraum *EWR3* nur optionale PMC für den Typ des Umgebungslichtsensors enthalten sind, kann eine Variation von Quantität und Qualität der Messwerte bei Nutzung aller Sensortypen nur über die beiden Umgebungslichtsensoren *OPT3001* und *MAX44009*

erfolgen. Diese weisen jedoch beide eine Datenbreite von 16 Bit auf und es finden sich in den Datenblättern keine vergleichbaren Angaben zu der Qualität der Messung (*mode.acc*). Es bestehen somit keine Optionen hinsichtlich der Quantität und Qualität der Messwerte.

Die Leistungsfähigkeit des Systems definiert sich somit durch die Bandbreite der Messdaten, die sich aus der summierten Datenbreite der Sensoren und dem untersuchten Bereich der Ausführungsfrequenzen ergibt. Da die Datenbreite der Sensoren für den betrachteten Entwurfsraum konstant ist (76 Bit), wird die Ausführungsfrequenz als Maß für die Leistungsfähigkeit angegeben. Verschiedene Konfiguration weisen bei fester Ausführungsfrequenz somit die gleiche Leistungsfähigkeit auf und die Effizienz der Konfiguration ergibt sich aus der Laufzeit des Systems. Um die Varianz in der Effizienz zu verdeutlichen, wird im Folgenden die Spannbreite zwischen der ineffizientesten und der effizientesten Konfiguration untersucht.

Da die Konfiguration mit der niedrigsten Leistungsaufnahme bereits systematisch ermittelt wurde (vgl. Abbildung 6.22) erfolgt analog dazu die Ermittlung der in Abbildung 6.28 dargestellten Konfiguration mit der jeweils höchsten Leistungsaufnahme in Abhängigkeit von der Ausführungsfrequenz. Hier zeigt sich, dass bis zu einer Ausführungsfrequenz von 0,01 Hz die Kombination aus Konfiguration *K17* und dem Spannungswandler *TPS62202* (*DC1*) die größte Leistungsaufnahme aufweist. Zwischen 0,01 Hz und ca. 0,8 Hz ist es die Kombination aus Konfiguration *K17* und dem Spannungswandler *TPS62243* (*DC2*). Zwischen 0,8 Hz und der maximalen Ausführungsfrequenz von 7,94 Hz hat die Kombination aus der Konfiguration *K01* und dem Spannungswandler *TPS62243* (*DC2*) die größte Leistungsaufnahme.

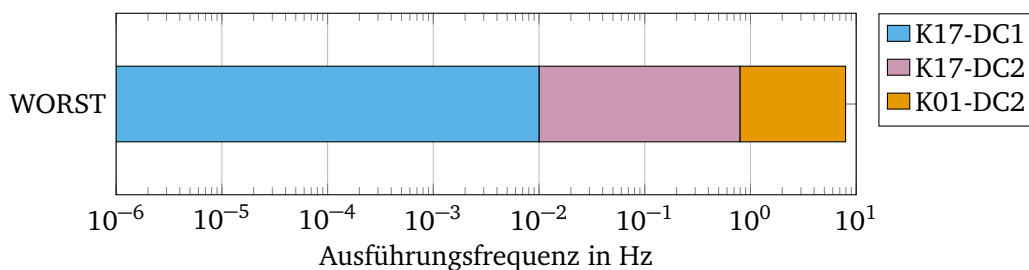


Abbildung 6.27.: Systemkonfigurationen aus PMC und Spannungswandler mit der höchsten Leistungsaufnahme.

In Abbildung 6.29 ist ein Vergleich für die Konfigurationen mit der hier ermittelten höchsten (*WORST*) und der zuvor niedrigsten Leistungsaufnahme (*BEST*, vgl. Abbildung 6.23) dargestellt. Hier zeigt sich die Spanne der möglichen Leistungsaufnahmen im Systementwurf. Diese Grenzfälle der Leistungsaufnahme für jede Ausführungsfrequenz bilden die obere und die untere Schranke für die in Abbildung 6.30 dargestellten Laufzeiten des Systems. Zur Analyse der Einflussfaktoren auf die Laufzeit des Systems werden exemplarische

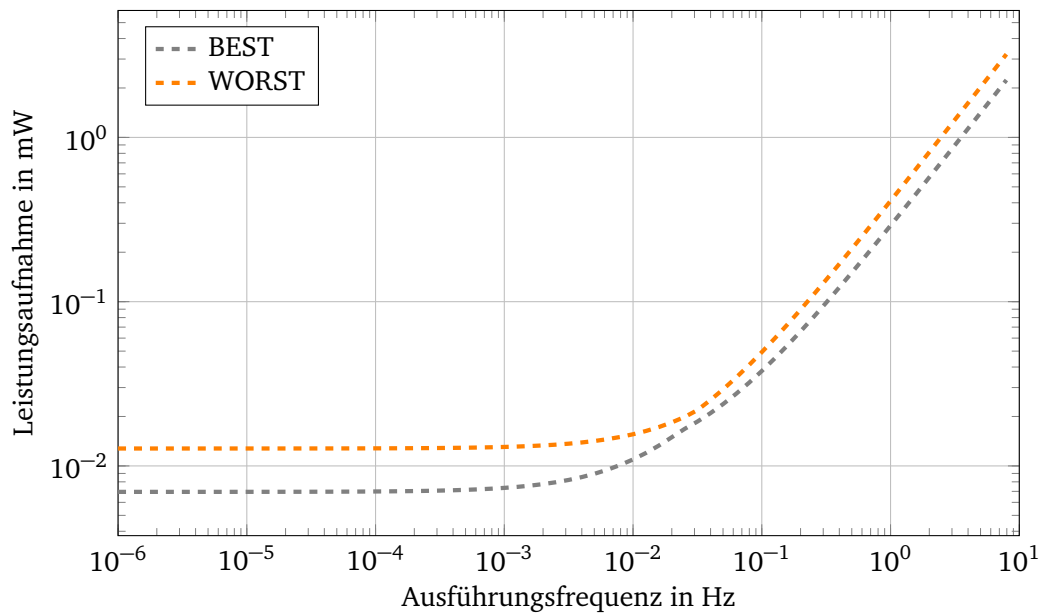


Abbildung 6.28.: Leistungsaufnahme des Systems für die Konfigurationen mit der höchsten und der niedrigsten Leistungsaufnahme

Konfigurationen für die Faktoren dargestellt und im Folgenden diskutiert.

Die über den gesamten Bereich der Ausführungsfrequenzen theoretisch optimale Systemkonfiguration *T1-BEST-B2 IdleOpt* weist die in Abbildung 6.22 aufgeführten Kombinationen aus PMC und Spannungswandler mit der jeweils kleinsten Leistungsaufnahme auf. Zudem verwendet die Konfiguration den Task *T1*, welcher innerhalb des Tasks den Wechsel des Mikrocontrollers in den inaktiven Modus der Konfiguration erlaubt. Zudem wird die Batterie *B2* (Maxell CR2032H) mit der höchsten Laufzeit verwendet. Darüber hinaus wird die *IdleOpt*-Option verwendet, welche den inaktiven Modus des Mikrocontrollers außerhalb des Tasks anhand der Nutzenzeit T_{BE} auswählt. Die Systemkonfiguration *T1-K36-DC2-B2 IdleOpt* verwendet die Kombination *K36-DC2* welche, wie zuvor ermittelt, die höchste Laufzeit für den niedrigen Frequenzbereich aufweist. Somit stimmt diese Systemkonfiguration über einen weiten Bereich der Ausführungsfrequenzen mit dem Optimum überein und es zeigt sich erst bei höheren Frequenzen eine Abweichung. Im Vergleich dazu verwendet die Systemkonfiguration *T3-K36-DC2-B2 IdleOpt* den Task *T3*, in dem der Mikrocontroller während des Tasks im aktiven Zustand verbleibt. Hier zeigt sich, dass diese Systemkonfiguration nur bei sehr niedrigen Ausführungsfrequenzen nahe am Optimum ist und bei höheren Frequenzen signifikante Abweichungen aufweist. Die Systemkonfiguration *T3-K36-DC1-B2 IdleOpt* beinhaltet den Spannungswandler *DC1* (*TPS62202*), welcher bei niedriger Leistungsaufnahme eine geringere Effizienz ausweist und somit in dem Bereich der niedrigen

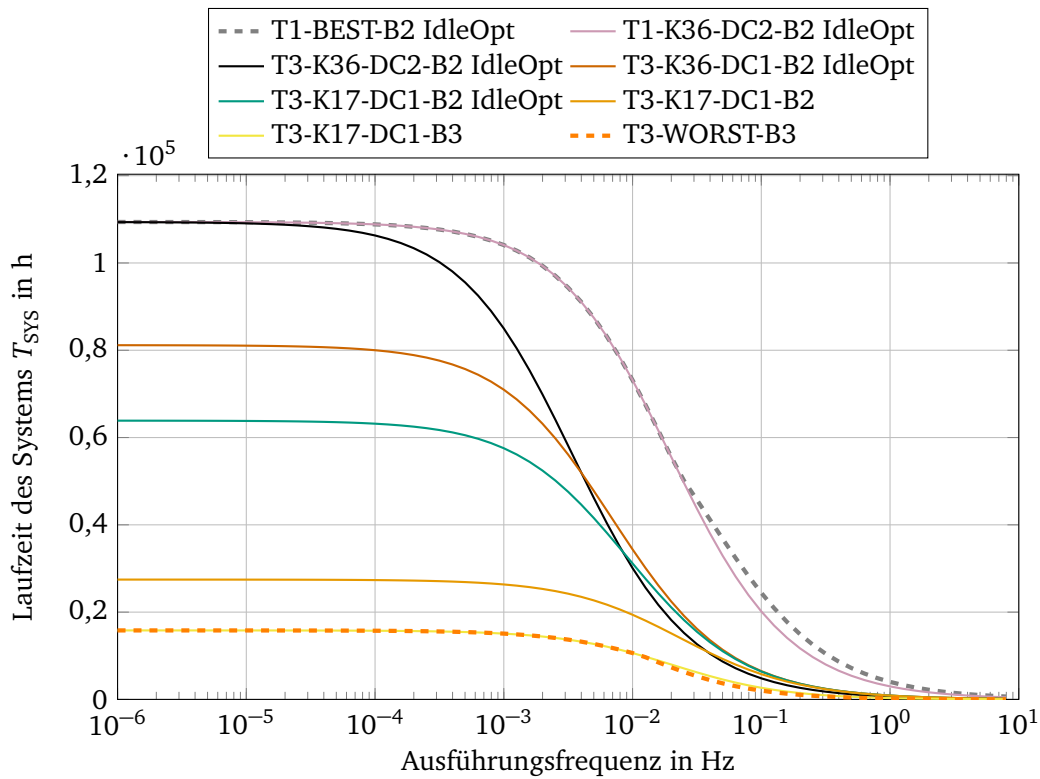


Abbildung 6.29.: Laufzeiten des Systems bei verschiedenen Systemkonfigurationen

Ausführungsfrequenzen die Laufzeit reduziert. Höhere Ausführungsfrequenzen führen zu höheren Leistungsaufnahmen, so dass der *TPS62202* im Vergleich zum *TPS62243 (DC2)* effizienter ist. Die Systemkonfiguration *T3-K17-DC1-B2 IdleOpt* weist die zuvor ermittelte (vgl. Abbildung 6.28) Kombination *K17-DC1* aus PMC und Spannungswandler mit der größten Leistungsaufnahme auf. Somit ergibt sich hier eine deutliche Reduzierung der Laufzeit. In der nächsten Abstufung verwendet die Systemkonfiguration *T3-K17-DC1-B2* nicht die *IdleOpt*-Option, so dass der inaktive Modus des Mikrocontroller außerhalb des Tasks nicht anhand der Nutzenzeit ermittelt, sondern anhand der Konfiguration der PMC ausgewählt wird. Hier zeigt sich entsprechend bei niedrigen Ausführungsfrequenzen eine deutliche Reduzierung der Laufzeit. Die Systemkonfiguration *T3-K17-DC1-B3* nutzt nicht die Batterie *B2 (Maxell CR2032H)* mit der höchsten Laufzeit, sondern die Batterie *B3 (Renata CR2025)* mit der niedrigsten Laufzeit. Somit stimmt die Laufzeit dieser Konfiguration zu weiten Teilen mit dem zuvor ermittelten Pessimium (*WORST*) überein. Diese nutzt darüber hinaus die, in Abhängigkeit von der Ausführungsfrequenz, Kombination mit der jeweils höchsten Leistungsaufnahme (vgl. Abbildung 6.28).

Zur quantitativen Beurteilung der Einflussfaktoren sind in Tabelle 6.24 die RMSE der einzelnen Systemkonfigurationen im Vergleich zur optimalen Konfiguration *T1-BEST-B2 IdleOpt* aufgeführt. Zudem wird die jeweilige Differenz des RMSE angegeben, die aus dem jeweiligen Einflussfaktor resultiert. Hier zeigt sich, dass der Wechsel des Mikrocontrollers im Task in einen inaktiven Zustand (Task *T3* gegenüber Task *T1*) und die Wahl des inaktiven Zustands außerhalb des Task (*IdleOpt*-Option) den größten Einfluss auf die Laufzeit haben. Die Wahl der Modi ist somit von entscheidender Bedeutung.

Tabelle 6.24.: *RMSE der Systemkonfigurationen im Vergleich zur Konfiguration T1-BEST-B2 IdleOpt*

Systemkonfiguration	RMSE	Δ	Einflussfaktor
<i>T1-K36-DC2-B2 IdleOpt</i>	1.480 h	1.480 h	feste Konf. <i>K36-DC</i>
<i>T3-K36-DC2-B2 IdleOpt</i>	19.169 h	17.689 h	Task <i>T3</i>
<i>T3-K36-DC1-B2 IdleOpt</i>	26.643 h	7.474 h	Spannungswandler <i>DC1</i>
<i>T3-K17-DC1-B2 IdleOpt</i>	36.916 h	10.273 h	Konf. <i>K17</i>
<i>T3-K17-DC1-B2</i>	61.329 h	24.413 h	<i>IdleOpt</i> -Option
<i>T3-K17-DC1-B3</i>	70.253 h	8.924 h	Batt. <i>B3</i>
<i>T3-WORST-B3</i>	70.311 h	58 h	variable Konf. <i>WORST</i>

6.8 Zusammenfassung

In diesem Kapitel wurde das im Rahmen dieser Arbeit aufgestellte Systemmodell für den modellgestützten Entwurf von drahtlosen Sensorknoten angewendet und so exemplarisch dessen Möglichkeiten aufgezeigt. Im Zuge der **Bauteil Auswahl der PMC** konnte zunächst eine Komponentendatenbank mit den benötigten Typen aufgebaut werden, wobei alle Parameter auf Basis der Informationen im Datenblatt ermittelt und im jeweiligen Komponentenmodell abgelegt werden konnten. Auf dieser Basis konnte modellgestützt eine Vorauswahl der Komponenten erfolgen, so dass der Entwurfsraum bereits auf dieser Ebene fokussiert werden konnte. Die **Bauteil Auswahl der Spannungswandler** erfolgte anhand der auf Basis der Komponentenmodelle ermittelten WCPC. Auf Basis des Modells für diese Komponenten konnte zudem die Effizienz der Spannungswandler bei zu erwartender minimaler und maximaler Leistungsaufnahme der PMC betrachtet werden. Auf Basis der beiden Grenz betrachtungen konnte jedoch noch keine Auswahl der Komponente erfolgen, da der *TPS62243* bei minimaler Stromaufnahme eine durchgehend höhere Effizienz aufwies, der *TPS62202* hingegen bei maximaler Stromaufnahme effizienter war. Bei der **Bauteil Auswahl der Energiespeicher** konnten auf Basis der Angaben im Datenblatt Modelle der Batterien generiert werden, die es erlauben, die Laufzeit eines Systems zu bestimmen. Da die Laufzeit, neben den Parametern des Energiespeichers, auch von der anwendungsspezifischen Systemkonfiguration aus Task und Tastgrad abhängt, konnte

nur auf Basis der Komponentenmodelle keine Vorauswahl erfolgen. Auf Basis der **Systemkonfigurationen** konnte im Entwurfsraum eine Bewertung des Systems aus PMC und Spannungswandler für den inaktiven Modus erfolgen. Hier konnten sowohl einzelne PMC gleicher Funktionalität verglichen, als auch zu verschiedenen Systemkonfigurationen kombiniert werden. Deren summierte Leistungsaufnahme wiederum diente als Basis für die Effizienzbetrachtung verschiedener Spannungswandler. Hier wurde deutlich, dass die Spannungswandler einen signifikanten Einfluss auf die Leistungsaufnahme des Systems haben. Auf Basis der **Taskkonfigurationen** konnte eine Bewertung des Systems aus PMC und Spannungswandler für den aktiven Modus und somit für die spätere Anwendung erfolgen. Es konnten einzelne PMC gleicher Funktionalität und deren Modi hinsichtlich ihrer Energieaufnahme für den definierten Anwendungsfall verglichen werden. Neben der Wahl der Komponenten hatte zudem die Wahl der Modi einen signifikanten Einfluss auf die Energieaufnahme, was durch den Parameter *ucsleep* und die Analyse der im Komponentenmodell hinterlegten Nutzenzeit T_{BE} für die Transitionen verdeutlicht wurde. Mit der ermittelten Leistungsaufnahme der PMC konnte zudem die Effizienz verschiedener Spannungswandler analysiert werden. Die Simulation des System mit dem definierten **Tastgrad** ermöglichte abschließend die Bestimmung der Laufzeit des Systems. Im Laufe der exemplarisch ausgeführten Auswertungen konnte der Entwurfsraum in jedem Schritt eingeschränkt werden, so dass dieser schlussendlich auf einen kleinen Lösungsraum fokussiert werden konnte. In der abschließenden **Bewertung des Systems** wurde die Spannbreite der möglichen Lösungen im Entwurfsraum hinsichtlich deren Effizienz aufgezeigt und die Einflussfaktoren analysiert. Hier zeigte sich, dass vor allem die Wahl der inaktiven Modi einen signifikanten Einfluss auf die Laufzeit eines Systems und somit auf die Effizienz einer Lösung hat.

Mit der exemplarischen Durchführung des modellgestützten Entwurfsablaufs konnte somit gezeigt werden, dass das Verfahren potentiell in der Lage ist, den Entwurf zu systematisieren und durch die Simulation der verschiedenen Entwurfsschritte zielführender zu gestalten. Offene Punkte wie die mögliche Validierung der Simulationen und weitere Anwendungsgebiete des Ablaufs werden im folgenden Kapitel 7 diskutiert.

7 Zusammenfassung und Ausblick

In diesem Kapitel wird die Arbeit zunächst zusammen gefasst. Anschließend erfolgt der Abgleich mit dem Handlungsbedarf und eine Diskussion der möglichen Einschränkungen. Abschließend wird ein Ausblick gegeben.

7.1 Zusammenfassung

Im Rahmen dieser Arbeit wurde in **Kapitel 2** zunächst der Stand der Technik drahtloser Sensorknoten hinsichtlich Systemstruktur, Komponenten, Messgrößen und Parametern sowie verfügbarer Systeme dargestellt. Somit wurde ein Überblick geschaffen und die Grundlagen für die weiteren Untersuchungen gelegt. Im folgenden **Kapitel 3** wurde die Methodik zur Vermessung drahtloser Sensorknoten eingeführt, aktuelle Arbeiten in der Literatur zur Messung der Leistungsaufnahme von eingebetteten Systemen vorgestellt und das Vorgehen anhand von zwei Praxisbeispielen verdeutlicht. Hier zeigte sich, dass heutzutage bereits vergleichsweise einfache Sensorknoten komplexe Konfigurationsmöglichkeiten bieten und somit Optimierungspotentiale hinsichtlich der Energieeffizienz nicht direkt ersichtlich sind, sondern experimentell ermittelt werden müssen. Bedingt durch die Konfigurationsvielfalt des Gesamtsystems und der experimentellen Vorgehensweise kann jedoch nicht sicher gestellt werden, dass mit einer der getesteten Konfigurationen das Energieoptimum des Sensorknotens erreicht wird. Somit besteht Bedarf an einer neuen Vorgehensweise. Im **Kapitel 4** wurde zunächst der Stand der Technik für die Modellierung der einzelnen Komponenten eines Sensorknoten beschrieben. Zudem wurde ein Überblick über die Ansätze zur Modellierung des Gesamtsystems und der Abbildung von gängigen Energiespartechniken gegeben. In der abschließenden Bewertung zum aktuellen Stand in der Modellierung zeigt sich, dass der Stand der Technik in Bezug auf die Modellierung der einzelnen Komponenten große Unterschiede hinsichtlich Komplexität und Anwendbarkeit aufweist. So kann das Verhalten von Sensoren, Transceiver, Speicher und Schnittstellen bei entsprechender Dokumentation durch den Hersteller hinreichend abgebildet werden, wohingegen die Modelle für Mikrocontroller, Energiespeicher und das Gesamtsystem aktuell noch Gegenstand der Forschung sind. Darauf folgend wurde im **Kapitel 5** die im Rahmen dieser Arbeit entwickelte Methodik für den Entwurf von drahtlosen Sensorknoten beschrieben. Einleitend erfolgte zunächst die Einordnung dieses modellbasierten Vorgehens in den bisherigen Entwurfsprozess drahtloser Sensorknoten. Im Anschluss wurde, basierend auf der zuvor untersuchten Vermessung und Modellierung drahtloser Sensorknoten, der Handlungsbedarf abgeleitet. Darauf folgte die theoretische Betrachtung des aufgestellten Modells, wobei zunächst das grundlegende System- und Komponentenmodell beschrieben wurde. Anschließend wurden die spezifischen Modelle für die einzelnen Komponenten (Mikrocontroller, Sensoren, Transceiver, Speicher, Schnittstellen, Spannungswandler, Energiespeicher) und deren Zusammenschluss zu einem Modell für das Gesamtsystem beschrieben (Systemkonfiguration). Darauf aufbauend wurde das Systemverhalten mittels der Definition von Tasks, dem Tastgrad und dem Datenfluss abgebildet. Somit wurde eine

Entwurfsraumexploration ermöglicht, die, der Definition von Herbrechtsmeier [78, S. 29] folgend, „eine Methode zur Sondierung alternativer Entwürfe und zur Ermittlung einer optimalen und ressourceneffizienten Lösung“ darstellt und „eine Mehrzieloptimierung sowie Entscheidungsfindung beinhaltet“. In dem darauf folgenden **Kapitel 6** wurde die im Rahmen dieser Arbeit aufgestellte Methodik für den modellgestützten Entwurf von drahtlosen Sensorknoten angewendet und so exemplarisch dessen Möglichkeiten aufgezeigt. Im Zuge der BauteilAuswahl konnte zunächst eine Komponentendatenbank mit den benötigten Typen aufgebaut werden. Auf dieser Basis konnte modellgestützt eine Vorauswahl der Komponenten erfolgen, so dass der Entwurfsraum der Systemkonfigurationen bereits auf diese Ebene fokussiert werden konnte. Auf Basis dieser Systemkonfigurationen konnte im Entwurfsraum eine Bewertung des Systems aus aktiven Komponenten (*Power Manageable Component* (PMC)) und Spannungswandler für den inaktiven Modus erfolgen. Hier konnten sowohl einzelne PMC gleicher Funktionalität verglichen, als auch zu verschiedenen Systemkonfigurationen kombiniert werden. Deren summierte Leistungsaufnahme wiederum diente als Basis für die Effizienzbetrachtung verschiedener Spannungswandler. Hier wurde deutlich, dass die Spannungswandler einen signifikanten Einfluss auf die Leistungsaufnahme des Systems haben. Auf Basis der Taskkonfigurationen, welche die Anwendung des Systems abbilden, konnte eine Bewertung des Systems aus PMC und Spannungswandler für den aktiven Modus erfolgen. Es konnten einzelne PMC gleicher Funktionalität und deren Modi hinsichtlich ihrer Energieaufnahme für den definierten Anwendungsfall verglichen werden. Neben der Wahl der Komponenten hatte zudem die Wahl der Modi einen signifikanten Einfluss auf die Energieaufnahme. Mit der ermittelten Leistungsaufnahme der PMC konnte zudem die Effizienz verschiedener Spannungswandler analysiert werden. Die Simulation des Systems mit dem definierten Tastgrad ermöglichte abschließend die Bestimmung der Laufzeit des Systems. Im Laufe der exemplarisch ausgeführten Auswertungen konnte der Entwurfsraum in jedem Schritt eingeschränkt werden, so dass dieser schlussendlich auf einen kleinen Lösungsraum fokussiert werden konnte. Mit der exemplarischen Durchführung des modellgestützten Entwurfsablaufs konnte daher gezeigt werden, dass das Verfahren in der Lage ist, den Entwurf zu systematisieren und durch die Simulation der verschiedenen Entwurfsschritte zielführender zu gestalten. Das Vorgehen erhöht somit zum einen die Effizienz des Sensorknotens und zum anderen die Effizienz der Entwicklung von drahtlosen Sensorknoten oder auch eingebetteten Systemen im weiteren Sinne.

7.2 Abgleich mit dem Handlungsbedarfs

In diesem Unterkapitel erfolgt ein Abgleich zwischen dem in Abschnitt 5.2 beschriebenen Handlungsbedarf und dem im Rahmen dieser Arbeit umgesetzten Vorgehen. Der grundlegende Bedarf eines modellbasierten Entwurfs leitete sich aus der stetig steigenden Komplexität der technischen Systeme ab, die sich in der Vielzahl der verschiedenen Komponenten eines Systems und deren möglicher Zustände begründete. Jedoch existierte noch

kein generisches und ganzheitliches Modell für den Entwurf drahtloser Sensorknoten, um diese Komplexität beherrschbar zu machen. Ob diese Anforderungen durch diese Arbeit erfüllt wurden, wird im Folgenden anhand der verschiedenen Anforderungen evaluiert.

7.2.1 Abzubildendes System

Ein generisches, ganzheitliches und gleichzeitig praktikables Modell zur Abschätzung der Energieaufnahme drahtloser Sensorknoten sollte verschiedene Aspekte erfüllen, die hier anhand der im Handlungsbedarf definierten Kategorien dargestellt werden.

Allgemeines Komponentenmodell Ein möglichst generisches Komponentenmodell sollte in der Lage sein, die für jede Komponente relevanten Parameter, wie die Leistungsaufnahme in verschiedenen Zuständen und die damit verbundenen Zustandsübergänge, abzubilden. Diese Anforderung ist erfüllt, da diese Informationen in strukturierter Form in den jeweiligen Komponentenmodellen hinterlegt sind und somit in den Entwurfsablauf einbezogen werden können.

Komponente Mikrocontroller Die Modellierung eines Mikrocontrollers sollte, neben der Leistungsaufnahme der verschiedenen Betriebsmodi, auch einen Zusammenhang zwischen dem auszuführenden Algorithmus und der für die Ausführung benötigten Energie herstellen. Diese Anforderung wurde bedingt erfüllt. In der exemplarischen Umsetzung wird die Dauer eines auszuführenden Algorithmus über die manuell gesetzte Anzahl der Taktzyklen und die im Komponentenmodell hinterlegte Taktfrequenz ein. Jedoch bestehen bereits Ansätze die Dauer einer Ausführung rein modellbasiert abzuschätzen (vgl. Ausblick in Abschnitt 7.4).

Komponente Transceiver Neben der Leistungsaufnahme in den verschiedenen Betriebsmodi sollte die Modellierung eines Funktransceivers auch die beeinflussenden Parameter wie die Breite der Nutzdaten, die Sendestärke und die Sensitivität des Empfangsteils enthalten. Diese Anforderung ist erfüllt, da alle benötigten Parameter im Komponentenmodell hinterlegt sind und als Funktion auf die zu sendenden Daten angewendet werden können. Parameter wie Sendestärke und die Sensitivität können als weitere aktive Modi in das Modell eingehen.

Komponente Sensoren Die Modellierung eines Sensors sollte, neben der Leistungsaufnahme in den möglichen Modi, auch die benötigte Dauer für die spezifischen Messungen enthalten. Ziel sollte eine einheitliche Darstellung sein, so dass eine Vergleichbarkeit zwischen verschiedenen Sensoren gegeben ist. Somit wäre auch eine Bewertung der Sensoren anhand der Energieeffizienz (Präzision pro Energie) möglich. Diese Anforderung ist erfüllt, da alle benötigten Parameter im Modell hinterlegt werden. Die Anforderung an die Energieeffizienz können jedoch nicht für jeden Typ erfüllt werden, da nicht in allen Fällen die entsprechenden Informationen im Datenblatt vorliegt.

Komponente Schnittstellen Die Modellierung von Schnittstellen sollte, neben der Leistungsaufnahme, auch die Taktfrequenz der Schnittstelle und den Protokolloverhead

enthalten. Diese Anforderung ist erfüllt, da alle benötigten Parameter im Komponentenmodell hinterlegt sind und als Funktion auf die zu sendenden Daten angewendet werden können.

Spannungsversorgung Da die Komponenten der Spannungsversorgung einen entscheidenden Einfluss auf die Leistungsaufnahme des Gesamtsystems haben, sollten die Effizienz des Spannungswandlers und die verschiedenen Batterieeffekte abgebildet werden. Für den Spannungswandler ist diese Anforderung voll erfüllt, da dessen Effizienz für alle Arbeitspunkte als Funktion im Komponentenmodell hinterlegt wird. Die Anforderung alle Batterieeffekte abzubilden ist bedingt erfüllt, da sowohl das Basismodell, als auch das erweiterte Modell der Batterie nicht alle Effekte abbilden. Jedoch werden durch das erweiterte Modell die wichtigsten primären Effekte wie die Abhängigkeit der nutzbaren Kapazität vom Entladestrom und der Zusammenhang zwischen Batteriespannung und Ladezustand abgebildet. Die Einbindung komplexer Modelle würde zum einen die Ermittlung weiterer Batterieparameter erfordern, die in den Datenblättern oftmals nicht enthalten sind. Zum anderen würde sich die Simulationszeit signifikant erhöhen. Somit wurde die Anforderung bedarfsgerecht erfüllt.

Datenfluss Um das Systemverhalten abbilden zu können, sollte das Modell, neben den direkten energetischen Aspekten wie den Betriebsmodi, auch funktionale Aspekte wie den Datenfluss zwischen den einzelnen Komponenten abbilden. Diese bestimmen maßgeblich die Verarbeitungszeiten und somit die Zeit in den aktiven Modi. Diese Anforderung ist erfüllt, da alle benötigten Parameter in den jeweiligen Komponentenmodell hinterlegt sind und als Funktion auf die zu übertragenden Daten angewendet werden können.

Gesamtsystem Entscheidend für die Abbildung des Gesamtsystems ist die ganzheitliche Modellierung des Systems anhand der enthaltenen Komponenten, so dass Wechselwirkungen untersucht werden können. Zudem müssen gängige Energiespartechniken und der Datenfluss abgebildet werden können. Diese Anforderungen sind erfüllt, da das Modell des Gesamtsystems sämtliche Aspekte einbezieht und somit auch die Analyse von Wechselwirkungen wie die Veränderung der Effizienz des Spannungswandlers bei absinkender Batteriespannung ermöglicht wird. Bezüglich der Einbeziehung von Energiespartechniken wurde die Wahl der effizientesten Modi anhand der Nutzenzeit demonstriert. Die Veränderung der Taktfrequenz und Versorgungsspannung wäre auch möglich, da diese als Parameter im Modell hinterlegt sind. Zur Einordnung des entwickelten Modells in den Stand der Technik erfolgt in Tabelle 7.1 der Abgleich. Hier wird deutlich, dass das im Rahmen dieser Arbeit entwickelte Modelle die Betrachtung aller Aspekte des Gesamtsystems ermöglicht.

Tabelle 7.1.: Vergleich der Systemmodells mit dem Stand der Technik. Mit X markierte Komponenten werden von dem Systemmodell abgedeckt. Mit ~ markierte Komponenten werden nur rudimentär betrachtet. Für Spannungswandler und Energiespeicher wird das jeweils verwendete Modell angegeben.

Arbeit	Jahr	Core	Sens.	Akt.	Schnittst.	Spannungsr.	Energiesp.
Simunic u. a. [188]	2001	X	-	-	-	η	Peukert
Konstantakos u. a. [110]	2008	X	X	X	-	-	-
Prayati u. a. [161]	2010	X	-	X	-	-	-
Abdallah u. a. [3]	2011	X	-	-	-	η	-
Martinez u. a. [124]	2015	X	X	X	-	η	-
Bag u. a. [15]	2016	X	-	-	-	-	Ideal
Tobola u. a. [210]	2016	X	X	-	-	η	Ideal
Chen u. a. [41]	2018	X	X	X	-	η	El. Mod. [39]
STM32CubeMX [178]	2018	X	~	~	~	-	n.b.
Diese Arbeit	2021	X	X	X	X	η	Peuk. + El. Mod.

7.2.2 Skalierbarkeit des Modells

Für eine praktische Verwendung eines modellbasierten Entwurfs sollte der Ansatz skalierbar sein. Das heißt, dass das Modell in der Lage sein muss, eine Vielzahl möglicher Systeme und auch Variationen innerhalb des Systems effizient abzubilden. Im Folgenden wird daher der Abgleich verschiedener Aspekte der Skalierbarkeit des Modells beschrieben.

Modularität Das gesamte Modell sollte durchgehend modular gestaltet werden, so dass zum einen die Komplexität des Gesamtsystems beherrschbar wird und zum anderen einzelne Module wieder verwendet werden können. Diese Anforderung ist erfüllt, da sämtliche Komponenten anhand von spezifischen Komponentenmodellen definiert werden und als Teile in die Systemkonfiguration eines Gesamtsystems eingehen können. Darauf aufbauend kann das Systemverhalten durch die Definition von verschiedenen Tasks und deren Tastgrad abgebildet werden. Sämtliche Komponentenmodelle, Konfigurationen und Simulationsergebnisse werden strukturiert abgelegt, so dass eine Wiederverwendbarkeit gegeben ist.

Generierung der Daten Großer Handlungsbedarf bestand bei den für die Modellierung benötigten Daten. Diese Anforderung ist bedingt erfüllt. Im Rahmen der Arbeit ist es zwar gelungen, für jeden Komponententyp auf Basis der Datenblattangaben ein Modell zu erstellen. Jedoch erfordert die Generierung der Modelle in vielen Fällen noch eine manuelle Eingabe dieser Angaben. Jedoch deutet sich hier in der technologischen Entwicklung bereits

ein Trend zu maschinenlesbaren Datenblättern an, der in Abschnitt 7.4 näher beschrieben wird.

Datenbasis Ziel in diesem Bereich sollte eine standardisierte Informationsbasis für den Entwurf drahtloser Sensorknoten sein, die beständig erweitert werden kann. Die Grundlagen für die Erfüllung dieser Anforderung wurden im Rahmen dieser Arbeit mit der strukturierten Ablage der Daten bereits geschaffen. In einem nächsten Schritt kann daher eine Publikation des Modells unter einer *Open Source* Lizenz geprüft werden.

Simulation Ein wichtiger Aspekt der Methodik ist die Simulation, so dass funktionale Aspekte, die für den Entwurfsablauf relevant sind, auch direkt simuliert werden können. Es sollte eine Entwurfsraumexploration für drahtlose Sensorknoten geschaffen werden, welche die primären Ziele Leistungsaufnahme und Leistungsfähigkeit als Basis für die Energieeffizienz abdeckt. Sekundäre Größen wie Bauform, Kosten könnten gegebenenfalls als eine weitere Ebene im Entwurfsablauf verstanden werden. Zusammenfassend sollte sich der für die Modellierung eines Systems benötigte Aufwand durch die Simulation und die sich daraus ergebenden Optimierungen amortisieren. Die Anforderung der Entwurfsraumexploration durch Simulation konnte erfüllt werden. So kann ein zunächst breiter Entwurfsraums durch die Simulation auf verschiedenen Ebenen auf einen kleinen Lösungsraum fokussiert werden. Für den Fall, dass Angaben zur Leistungsfähigkeit verfügbar sind, kann entsprechend auch die Energieeffizienz betrachtet werden. Dieses konnte auf Komponentenebene und über die Ausführungsfrequenz der Tasks demonstriert werden. Weitere Entwurfsziele wie Bauform und Kosten können als weitere Ebene mit einbezogen werden. Inwiefern sich der modellbasierte Entwurf amortisiert muss über den Verlauf von mehreren Entwurfsabläufen für verschiedene Systeme untersucht werden. Entscheidend ist hier die Verfügbarkeit der benötigten Daten in maschinenlesbarer Form, so dass der umgesetzte modellbasierte Entwurfsmethodik skaliert werden kann.

7.3 Mögliche Einschränkungen

Im Folgenden werden offene Punkte, welche die Anwendbarkeit des modellbasierter Entwurfsablaufs eventuell einschränken könnten, aufgeführt und diskutiert.

7.3.1 Validität

Ein entscheidender Punkt stellt die Validität der Simulationsergebnisse dar. Diese nachzuweisen ist in der Praxis komplex, da der Nachweis der untersuchten Kennzahlen wie die Leistungs- und Energieaufnahme bzw. die Laufzeit des Systems jeweils Prüfaufbauten für dessen Messung erfordern. Somit könnten einzelne Ergebnisse zumindest punktuell validiert werden. Eine Möglichkeit dieses Vorgehen effizient zu gestalten könnte der Aufbau von automatisierten Prüfumgebungen (*Testbeds*) sein. Diese könnten, zumindest auf Softwareebene, zu einer Validierung führen. So wäre es denkbar, in Bezug auf den auszuführenden Algorithmus und den Tastgrad verschiedene Varianten zu generieren, auf eine Evaluierungsplattform zu übertragen und dessen Leistungsaufnahme zu vermes-

sen. Diese Messungen könnten nachfolgend für eine iterative Optimierung des Modells genutzt werden. Um potentielle Unsicherheiten im Modell zu verdeutlichen, wäre es zudem denkbar, für alle Parameter und Kennzahlen Mittelwert und Varianz anzugeben (vgl. Martinez u. a. [124]). Die Einbeziehung der Varianz führt somit zur Bildung eines Unsicherheitsbereich für die Ergebnisse, wobei eine Präzisierung der Daten oder auch des Modells zu einer Reduzierung der Unsicherheit führt.

7.3.2 Energieeffizienz

Der für die Effizienz entscheidende Bezug zur Leistungsfähigkeit einer Lösung konnte im exemplarisch gezeigten Entwurfsablauf für den Umgebungssensorknoten anhand der Variation der Quantität der Messdaten (Ausführungsfrequenz der Tasks) gezeigt werden. Jedoch sollten zukünftig auch weitere Aspekte wie die Qualität der Messwerte und deren Datenverarbeitung eingehen. Vielfach entscheidend für die Leistungsfähigkeit ist hier der auszuführende Algorithmus, dessen Einbeziehung im Ausblick in Unterabschnitt 7.4.1 beschrieben wird.

7.3.3 Generierung der Daten

Ein wichtiger Aspekt bei einem modellbasierten Vorgehen ist der für die Modellierung aufzuwendende Mehraufwand. Dieser muss in einem praktikablen Verhältnis zum Nutzen stehen bzw. sollte idealerweise nicht vorhanden sein. Handlungsbedarf besteht hier bei der Ermittlung der für die Modellierung benötigten Daten. So liegen die Daten oftmals nicht in digitaler Form und zudem in heterogener Ausprägung vor, was die Skalierung des beschriebenen modellbasierten Entwurfs für eine breite Basis an Komponentenmodell einschränkt. Ansätze zur Lösung dieser Einschränkung werden im Ausblick in Unterabschnitt 7.4.2 beschrieben.

7.3.4 Generische Anwendbarkeit

Die generische Anwendbarkeit des Modells setzt voraus, dass sämtliche Aspekte im Entwurfsablauf und alle Eigenschaften der Komponenten und Systeme berücksichtigt werden. Perspektivisch könnte sich hier die Umsetzung des Modells in MATLAB als nicht ausreichend skalierbar erweisen, so dass die Arbeiten gegebenenfalls in ein anderes Softwareframework portiert werden müssen. Zudem müssen auch zukünftige Trends im Bereich IoT, wie die steigende Anzahl der Komponenten pro System [93], die lokale Ausführung von maschinellen Lernverfahren (*IoT Data as a Service* (IoTDaaS)) [81], die Versorgung mittels *Energy Harvesting* [81] und die Einbringung von Sicherheitsaspekten [169] abgebildet werden können. Somit muss sich die Entwurfsmethodik fortlaufend an die Anforderungen anpassen.

7.4 Ausblick

Im Folgenden wird dargelegt, wie der modellbasierte Entwurfsablauf zukünftig weiter entwickelt werden kann. Ansatzpunkte sind hier der modellgestützte Entwurf der Software, die Skalierbarkeit und weitere Anwendungen.

7.4.1 Modellgestützter Entwurf der Software

Ein entscheidender Punkt für die zukünftige Entwicklung ist der modellgestützte Entwurf der Software. Also der Zusammenhang zwischen dem auszuführenden Algorithmus und der für die Ausführung benötigten Energie. Hier besteht in der Arbeitsgruppe Kognitronik & Sensorik (AG K&S) bereits ein Ansatz den Quellcode des auszuführenden Algorithmus automatisch hinsichtlich der Anzahl und Typen der Instruktionen zu analysieren [289]. In einem nächsten Schritt soll die Verknüpfung der auszuführenden Instruktionen mit der Ausführungszeit auf verschiedenen Zielplattformen erfolgen. Denkbar wäre hier die Nutzung des OVP Emulators [90]. Perspektivisch wäre es auch interessant den Task, der in das Modell eingeht, direkt aus der Software des Mikrocontrollers zu generieren, da dieser bereits den Ablauf, wie die Kommunikation mit den Sensoren, beinhaltet. Zudem bestehen in der AG K&S bereits Ansätze zur Abschätzung der Energieaufnahme auf verschiedenen Zielplattformen. So arbeiten Klarhorst u. a. [106] an einem Energiemodell für einen Multi-Processor System-on-a-Chip (MPSoC), Ostrau u. a. [145; 146] an der Abbildung neuromorpher Hardware und Homburg u. a. [254] an der Generierung von neuronalen Netzen für spezifische Zielarchitekturen. Hier soll im nächsten Schritt geprüft werden, inwiefern diese Arbeiten verknüpft werden können. In diesem Rahmen wäre zudem die Entwicklung einer automatisierten Prüfumgebung denkbar.

7.4.2 Skalierung

Zur Skalierung der Arbeit könnte das Modell in einer Gemeinschaft (z.B. *Open Source*) weiter entwickelt werden, so dass sich interessierte Personen einbringen können (*Crowdsourcing*). Somit könnte eine umfassende und stetig wachsende Datenbank der Komponenten entstehen. Zudem hat eine einheitliche Ausprägung der Daten durch einen Standard oder eine Norm das Potential, das Vorgehen erheblich zu beschleunigen bzw. vollständig automatisierbar machen. Hier könnte perspektivisch das *Common Data Dictionary* [92] genutzt werden, welches unter *Electric/electronic components (IEC 61360-4)* einheitliche Parameter für elektronische Komponenten wie Sensoren und Mikrocontroller definiert. Somit wäre eine maschinenlesbare Form der Daten möglich. Jedoch waren zum Zeitpunkt dieser Arbeit weder von Herstellern noch von Distributoren Daten in dieser Form für spezifische Komponenten verfügbar. Weitere Standards stellen *eCl@ss* [239], *United Nations Standard Products and Services Code* (UNSPSC) und *Electro-Technical Information Model* (ETIM) dar. In der Industrie ist besonders der *eCl@ss* Standard verbreitet, welcher die Klassifizierung und Beschreibung von Produkten und Dienstleistungen durch standardisierte, ISO-konforme Merkmale erlaubt. So finden sich beispielsweise definierte

Merkmale für Mikrocontroller (27-26-17-03) und Sensoren (27-26-17-07). Demnach existieren bereits potentiell geeignete Standards für die einheitliche Beschreibung, jedoch haben diese bisher im Bereich der elektronischen Komponenten noch keine signifikante Verbreitung gefunden. Großes Potential verspricht derzeit das durch die Initiative *Plattform Industrie 4.0* definierte Modell der Verwaltungsschalen [157] (*Asset Administration Shell* (AAS)). Dieses definiert einheitliche Standards für die Beschreibung von Geräten in der Automatisierungstechnik und erfährt dort eine zunehmende Verbreitung. Somit entsteht dort ein Digitaler Zwilling [32] von Fertigungsanlagen. Hier könnte eine Adaption auf die Ebene der elektronischen Komponenten ein möglicher Weg sein, die Arbeit der Initiative auch für den modellbasierten Entwurf von eingebetteten Systemen zu nutzen.

7.4.3 Anwendungen

Im Folgenden werden mögliche Anwendungen des modellbasierten Entwurfsablaufs diskutiert. So ermöglicht die Modellierung der Batterie die zielgerichtete **Anpassung der Spannungsversorgung** an die Anwendung. Beispiele finden sich hier in dem Entwurf der Spannungsversorgung in Abhängigkeit von der Batterie, die Optimierung eines Scheduling zur Vermeidung von Spitzen in der Leistungsaufnahme und eine optimierte Lastverteilung bei hybriden Batteriesystemen [168]. Im Sinne der Vermeidung von Leistungsspitzen könnte auch ein kontinuierlich aktives System mit geringer Taktrate am effizientesten sein, da bei kleinen Strukturgrößen der Halbleiter ansonsten die Leckströme dominieren [77]. Interessant wäre zudem ein Vergleich der Versorgung mit verschiedenen Spannungsdomänen auf einem System. Somit ergäbe sich ein Vergleich zwischen Kosten bzw. Flächenbedarf und Effizienz der Spannungsversorgung. Neben der hier demonstrierten Analyse verschiedener Konfigurationen hinsichtlich ihrer Leistungsaufnahme wäre auch die Optimierung eines Systems auf ein vorgegebenes Energiebudget denkbar. Diese Anforderung findet sich beim *Energy Harvesting*, bei dem das System durch eine begrenzt verfügbare Energie aus der Umgebung versorgt wird. Hier finden sich in der Literatur zahlreiche Arbeiten (u.a. [62; 173; 213]), in denen drahtlose Sensorknoten mittels *Energy Harvesting* versorgt werden. Zudem gibt es in der *Arbeitsgruppe Kognitronik & Sensorik* (AG K&S) bereits Vorarbeiten [279; 285], in denen untersucht wurde, welches Energiebudget von welcher Quelle, in Abhängigkeit von der Fläche und Rahmenbedingungen wie dem Wetter [21], zur Verfügung steht. Zur Anwendung des Entwurfsablaufs auf ein festes Energiebudget könnten entsprechende Spannungsversorgungen für *Energy Harvesting* [191] in das Modell eingebunden werden. Das Potential hier zeigt sich zudem darin, dass die Publikation einer frühen Version des Modells [250] bereits von Arbeiten aus dem Bereich *Energy Harvesting* zitiert wurde [2; 129; 156]. **Weitere Applikationen** finden sich im Bereich des energiegetriebenen *Machine Learnings*, so dass ein passender Kompromiss zwischen benötigter Energie und zu erreichender Genauigkeit gefunden werden kann [99; 116]. Hier bestehenden auch bereits Vorarbeiten in der AG K&S [49; 84].

Weitere Möglichkeiten bestehen in der Robotik [272; 273; 226], der Integration von Sensorsystemen in Alltagsgegenstände [248; 249], im Bereich der Industriesensorik [245;

271; 274] und weiteren Forschungsfeldern der AG K&S [243–274], [275–296]. Denkbar wäre auch die Ausdehnung des Modell auf größere Systeme (*System of Systems*) wie Sensornetzwerke [269; 119] oder auch Rechenzentren [242]. Zudem könnten weitere Zielgrößen wie Größe, Gewicht, Kosten oder andere **nicht-funktionale Zielgrößen** [220] in das Modell eingebunden werden.

7.4.4 Antizipation

Wie bereits beschrieben hat die im Rahmen dieser Arbeit entwickelte modellbasierte Entwurfsmethodik das Potential, sowohl die Effizienz des Energieeinsatzes, als auch die Effizienz in der Entwicklung zu erhöhen. Die Steigerung der Effizienz des Energieeinsatzes zeigt sich unter anderem darin, dass mit Hilfe des modellbasierten Vorgehens systematisch die effizienteste Konfiguration ermittelt werden konnte. Demgegenüber stand bisher das Vorgehen der Optimierung durch Messung, wobei nur punktuelle Verbesserung vorgenommen werden konnten. Die Steigerung der Effizienz in der Entwicklung zeigt sich durch die entstehende Datenbank an Komponenten, Konfigurationen und Simulationsergebnissen. So wären die in einer Datenbank hinterlegten Bauteile direkt vergleichbar und könnten in eine Simulation des Gesamtsystems eingehen, welche potentiell zu weniger bzw. kürzeren Entwicklungszyklen führt und somit die Entwicklungszeit verkürzt.

Im bisherigen Entwurfsablauf ist es oftmals so, dass ausreichend dokumentierte und vorab evaluierte Bauteile bei Erstellung des Systems unter Umständen bereits einige Zeit auf dem Markt sind. Geht man von typischen Produktentwicklungszyklen aus, so kann die Definition der Anforderungen und der Architektur für ein eingebettetes System 6 bis 12 Monate beanspruchen [51]. In dessen Verlauf werden die Komponenten ausgewählt und zu einem Gesamtsystem verbunden. Diese Komponenten unterliegen jedoch dem Entwicklungszyklus der Halbleiterindustrie, welcher in Abbildung 7.1 schematisch dargestellt ist.

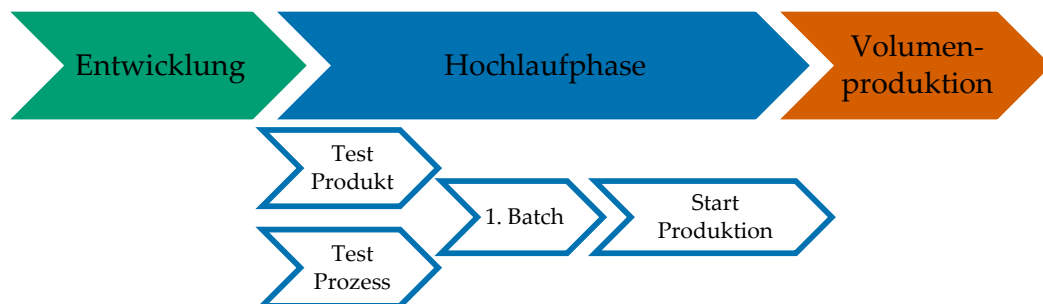


Abbildung 7.1.: Darstellung der Hochlaufphase eines Produktionsprozesses angelehnt an Willmann und Kastner [228]

Als erste Phase erfolgt die Entwicklung des neuen Produktes, z.B. eines Sensors. Hier können zwischen dem Beginn der Entwicklung und dem Start der Produktion 6-8 Jahre

liegen [33] (*time-to-market*). Jedoch liegt der Start der Produktion noch in der Hochlaufphase (*Ramp-up*), welche die Optimierung der Produktion bezeichnet, bis das neue Produkt in der gewünschten Qualität, Quantität und für den angestrebten Stückpreis produziert werden kann [228]. Sobald dieser Punkt erreicht ist, schließt sich die Phase der Volumenproduktion an, ab der das Produkt in hinreichender Stückzahl am Markt verfügbar ist [196] (*time-to-volume*). Die Dauer der Hochlaufphase hängt dabei von zahlreichen Faktoren wie den Fertigungsprozessen und der Zulieferkette ab und kann nicht pauschal abgeschätzt werden [228]. Wenn der Hersteller jedoch die, anhand der ersten gefertigten Produkte (*1. Batch*) ermittelten Parameter zur Verfügung stellt, können mit Hilfe der modellbasierten Entwurfsmethodik bereits Komponenten in die Entwurfsentscheidung einbezogen werden, die noch nicht auf dem Markt verfügbar sind. Diese können vorab in der Simulation evaluiert und bei Verfügbarkeit direkt integriert werden. Zusammengefasst ist das im Rahmen dieser Arbeit definierte Vorgehen daher in der Lage, das bisher nicht spezifizierte Vorgehen der Bauteilrecherche [78, S. 165] und nachfolgende Entwurfsschritte zu systematisieren und den Prozess zudem nachvollziehbar zu machen.

Zudem können die steigende Anzahl der Komponenten [93], die lokale Ausführung von maschinellen Lernverfahren [81] und die Versorgung mittels *Energy Harvesting* [81] als zukünftige Trends im Bereich IoT mittels der entwickelten Entwurfsmethodik antizipiert werden. Hier stellt sich schlussendlich die Frage, wie viele Ressourcen für welches Ergebnis investiert werden müssen.

Abbildungsverzeichnis

1.1	Entwicklung der Anzahl der jährlichen Publikationen für die Begriffe <i>Internet of Things</i> und <i>IoT</i> zusammen mit verschiedenen Schlüsselwörtern. Die Daten wurden mit <i>Dimensions</i> [52] ermittelt.	1
2.1	Schematische Darstellung der Komponenten eines drahtlosen Sensorknotens	5
4.1	Darstellung der Effizienz des <i>Texas Instruments TPS62243</i> Spannungswandlers im Datenblatt [206]. <i>Courtesy of Texas Instruments</i>	40
4.2	Rint-Batteriemodell nach He u. a. [75]	44
4.3	Thevenin Batteriemodell 1. Ordnung nach Liu u. a. [120]	45
4.4	DP-Modell nach He u. a. [75]	46
4.5	Verlauf der Klemmenspannung einer Batterie in Folge eines Entladepulses [24]. ©2018 IEEE.	47
4.6	Laufzeit-Batteriemodell nach M. Chen und Rincon-Mora [38]	47
4.7	Laufzeit-Batteriemodell 1. Ordnung nach Benini u. a. [18]	48
4.8	Kombiniertes Modell nach M. Chen und Rincon-Mora [38]	49
4.9	Kombiniertes Modell nach Y. Chen u. a. [39]	50
4.10	Generierung eines Batteriemodells der Ebene 1 nach Petricca u. a. [154]. ©2013 IEEE	52
4.11	Von Petricca u. a. [154] verwendetes vereinfachtes Laufzeit-Batteriemodell nach Benini u. a. [18]	53
5.1	Entwurfsmodell eines agilen Leiterplattenentwurfsprozesses nach Herbrechtsmeier [78]	64
5.2	Prozessmodell eines agilen Leiterplattenentwurfsprozesses nach Herbrechtsmeier [78]	65
5.3	BauteilAuswahl in einem agilen Leiterplattenentwurfsprozess nach Herbrechtsmeier [78]	65
5.4	Formale Darstellung der Komponenten eines drahtlosen Sensorknotens . .	70
5.5	Abbildung der Zustände und Zustandsübergänge einer Komponente anhand einer PSM (angelehnt an [17])	71
5.6	Generiertes Laufzeit-Batteriemodell nach Bocca u. a. [24]	83
5.7	Datenfluss im Systemmodell	86
6.1	Vergleich der SoC <i>nRF51822</i> und <i>BLE113</i>	101
6.2	Vergleich der Umgebunglichtsensoren <i>OPT3001</i> und <i>MAX44009</i>	104
6.3	Vergleich der Umgebungssensoren anhand der Leistungsaufnahme im inaktiven Modus P_I und der Energieaufnahme pro Messung	108
6.4	Vergleich der Umgebungssensoren anhand der Präzision der Messung und der Energieaufnahme pro Messung	108

6.5	Vergleich der Beschleunigungssensoren	110
6.6	Effizienz des <i>Texas Instruments TPS62243</i> Spannungswandlers auf Basis der vom Hersteller exportierten Daten	115
6.7	Darstellung der inter- und extrapolierten Effizienz des <i>Texas Instruments TPS62243</i> Spannungswandlers auf Basis der vom Hersteller exportierten Daten	115
6.8	Effizienz der Spannungswandler bei minimaler und maximaler Stromaufnahme des Systems für den jeweils möglichen Bereich der Eingangsspannungen	116
6.9	Verlauf der Klemmenspannung in Abhängigkeit des SOC für jeweils zwei verschiedene Entladeströme	119
6.10	Annäherung von Kurven zur Ermittlung der Peukert-Kapazität und der Peukert-Zahl	120
6.11	Innenwiderstand in Abhängigkeit vom SOC	121
6.12	Leerlaufspannung in Abhängigkeit vom SOC	122
6.13	Leistungsaufnahme im inaktiven Modus für alle Systemkonfigurationen im Entwurfsraum <i>EWR3</i>	126
6.14	Anteilige minimale Leistungsaufnahme der PMC im inaktiven Modus . . .	127
6.15	Anteilige minimale Leistungsaufnahme der PMC und Spannungswandler im inaktiven Modus	128
6.16	Schematischer Ablauf eines Tasks	130
6.17	Energieaufnahme der Tasks <i>T3</i> und <i>T4</i> für alle Systemkonfigurationen . .	132
6.18	Energieaufnahme der Tasks <i>T1</i> und <i>T3</i> für alle Systemkonfigurationen . .	132
6.19	Durchschnittliche Energieaufnahme von Task <i>T1</i> für die PMC und Spannungswandler für alle Systemkonfigurationen	135
6.20	Leistungsaufnahme der PMC bei Ausführung von Task <i>T1</i> über alle Systemkonfigurationen für eine Ausführungsfrequenz von 1 Hz	139
6.21	Leistungsaufnahme der PMC bei Ausführung von Task <i>T1</i> über alle Systemkonfigurationen für verschiedene Ausführungsfrequenzen	139
6.22	Systemkonfigurationen aus PMC und Spannungswandler mit der niedrigsten Leistungsaufnahme.	140
6.23	Leistungsaufnahme des Systems für verschiedene Konfigurationen	140
6.24	Systemkonfigurationen aus PMC und Spannungswandler mit der höchsten Leistungsaufnahme.	141
6.25	Laufzeit des Systems anhand das Basismodells für die Kombination <i>K36-DC2</i> für verschiedene Batterien	142
6.26	Laufzeit des Systems anhand des Basismodells für die Batterie Maxell <i>CR2032H</i> für verschiedene Systemkonfigurationen	143
6.27	Laufzeit des Systems anhand des Basismodells und des erweiterten Modells für verschiedene Batterien über den gesamten Bereich der Ausführungsfrequenzen für die Kombination <i>K36-DC2</i>	144

6.28 Systemkonfigurationen aus PMC und Spannungswandler mit der höchsten Leistungsaufnahme.	146
6.29 Leistungsaufnahme des Systems für die Konfigurationen mit der höchsten und der niedrigsten Leistungsaufnahme	146
6.30 Laufzeiten des Systems bei verschiedenen Systemkonfigurationen	148
7.1 Darstellung der Hochlaufphase eines Produktionsprozesses angelehnt an Willmann und Kastner [228]	160

Tabellenverzeichnis

2.1	Auswahl verfügbarer Umgebungssensoren	18
2.2	Überblick der Körpersensoren: Herzfrequenz-Gurte	19
2.3	Überblick der Körpersensoren: Multimodale Sensormodule	21
3.1	Gemessene Leistungsaufnahme des BG-V4.2 Körpersensorknotens nach Optimierung	27
3.2	Energieaufnahme der CC2560 BLE-Funkübertragung	28
3.3	Stromaufnahme der CC2560STK Komponenten im inaktiven Modus	30
3.4	Stromaufnahme der CC2560STK Komponenten in den aktiven Modi	30
3.5	Vergleich der CC2560STK Konfigurationen	31
4.1	In den Datenblättern enthaltene Daten für Lithium-Ionen Zellen nach Bocca u. a. [24]	54
4.2	Übersichtstabelle der Systemmodelle	61
5.1	Parameter des grundlegenden Komponentenmodells	72
5.2	Spezifische Parameter des Mikrocontrollers	75
5.3	Spezifische Parameter des Sensor	76
5.4	Spezifische Parameter des Transceivers	77
5.5	Spezifische Parameter des Speichers	78
5.6	Spezifische Parameter des Spannungswandlers	79
5.7	Spezifische Parameter für das Basismodell der Batterie	82
5.8	Spezifische Parameter für das erweiterte Modell der Batterie	84
5.9	Modell der Schnittstellen	85
5.10	Parameter der Systemkonfiguration	87
5.11	Kennzahlen des inaktiven Modus	88
5.12	Parameter der Taskkonfiguration	89
5.13	Kennzahlen der aktiven Modi	91
5.14	Parameter des Tastgrads	92
5.15	Kennzahlen des Tastgrads	93
5.16	Kennzahlen des Modells	94
6.1	Definierte Parameter der SoC: Modi	97
6.2	Ermittelte Parameter der SoC: Schnittstellen und Fläche	99
6.3	Generierte Parameter der Umgebungslichtsensoren	102
6.4	Ermittelte Parameter der Umgebungslichtsensoren	103
6.5	Ermittelte Parameter der Umgebungssensoren: Versorgungsspannung und Modi	105
6.6	Ermittelte Parameter der Umgebungssensoren: Schnittstellen und Fläche	106
6.7	Generierte Parameter der Umgebungssensoren	107

6.8	Ermittelte Parameter der Beschleunigungssensoren: Versorgungsspannung und Modi	111
6.9	Ermittelte Parameter der Beschleunigungssensoren: Schnittstellen und Fläche	112
6.10	Ermittelte Parameter der Spannungswandler	114
6.11	Ermittelte Parameter der Energiespeicher für das Basismodell	118
6.12	Generierte Parameter der Energiespeicher für das Basismodell	120
6.13	Anzahl der Modi der PMC	123
6.14	Größe der Entwurfsräume	125
6.15	Minimale Leistungsaufnahme der PMC im inaktiven Modus	126
6.16	Taskkonfiguration	131
6.17	Kombinationen aus Task und Systemkonfigurationen mit der niedrigsten Energieaufnahme	133
6.18	Kombinationen aus Task und Systemkonfigurationen mit der höchsten Energieaufnahme	133
6.19	Systemkonfigurationen <i>K01</i> und <i>K40</i> im <i>EWR3</i> mit maximaler und minimaler Energieaufnahme der PMC für den Task <i>T1</i>	134
6.20	Aktivität der Konstantlichregelung	137
6.21	Parameter des Tastgrads	137
6.22	Mögliche Batteriemodelle	141
6.23	Laufzeit des Systems	142
6.24	RMSE der Systemkonfigurationen im Vergleich zur Konfiguration <i>T1-BEST-B2 IdleOpt</i>	148
7.1	Vergleich der Systemmodells mit dem Stand der Technik	155
A.1	Überblick der Körpersensoren: Armbänder	202
A.2	Überblick der Körpersensoren: Herzfrequenzsensoren	203
A.3	Überblick der Körpersensoren: Multimodale Sensormodule	204
A.4	Überblick der Umgebungssensoren	205

Abkürzungsverzeichnis

6LoWPAN	<i>IPv6 over Low power Wireless Personal Area Network.</i>
AAS	<i>Asset Administration Shell.</i>
ADC	<i>Analog Digital Converter.</i>
AFE	<i>Analog Front End.</i>
AG K&S	<i>Arbeitsgruppe Kognitronik & Sensorik.</i>
BLE	<i>Bluetooth Low Energy.</i>
C-MEOP	<i>Core Minimum Energy Operation Point.</i>
CC	<i>Constant Current.</i>
CITEC	<i>Cognitive Interaction Technology Excellence Cluster.</i>
CoAP	<i>Constrained Application.</i>
CRC	<i>Cyclic Redundancy Check.</i>
DAC	<i>Digital Analog Converter.</i>
DMA	<i>Direct Memory Access.</i>
DPM	<i>Dynamic Power Management.</i>
DSP	<i>Digitaler Signalprozessor.</i>
DVFS	<i>Dynamic Voltage and Frequency Scaling.</i>
EDA	<i>Elektrodermale Aktivität.</i>
EEG	<i>Elektroenzephalogramm.</i>
EEMBC	<i>Embedded Microprocessor Benchmark Consortium.</i>
EKG	<i>Elektrokardiogramm.</i>
EMG	<i>Elektromyogramm.</i>
ETIM	<i>Electro-Technical Information Model.</i>
FLPA	<i>Functional Level Power Analysis.</i>
GUI	<i>Graphical User Interface.</i>
HR	<i>Herzrate.</i>
HRV	<i>Herzratenvariabilität.</i>
I2C	<i>Inter-Integrated Circuit.</i>
IEEE	<i>Institute of Electrical and Electronics Engineering.</i>
ILPA	<i>Instructional Level Power Analysis.</i>

IMU	<i><u>I</u>ner<u>t</u>ial <u>M</u>ea<u>s</u>ure<u>m</u>ent <u>U</u>nit.</i>
IoT	<i><u>I</u>nter<u>n</u>et of <u>T</u>hings.</i>
IoTDaaS	<i><u>I</u>o<u>T</u> <u>D</u>ata as a <u>S</u>ervice.</i>
IPG	<i><u>I</u>mpedance <u>P</u>le<u>t</u>hys<u>m</u>og<u>r</u>aphy.</i>
ISM	<i><u>I</u>ndustrial <u>S</u>cientific <u>M</u>edical (Band).</i>
ISS	<i><u>I</u>nstru<u>k</u>tion<u>s</u>sat<u>z</u>simu<u>l</u>ator.</i>
LED	<i><u>L</u>ight <u>E</u>mitting <u>D</u>iode.</i>
LUT	<i><u>L</u>ook<u>u</u>p <u>t</u>able.</i>
MBSE	<i><u>M</u>odel-<u>B</u>ased <u>S</u>ystems <u>E</u>ngineering.</i>
MEMS	<i><u>M</u>ikro<u>e</u>l<u>e</u>ktro<u>m</u>echanische <u>S</u>ysteme.</i>
MEOP	<i><u>M</u>inimum <u>E</u>nergy <u>O</u>peration <u>P</u>oint.</i>
MPSoC	<i><u>M</u>ulti-<u>P</u>rocessor <u>S</u>ystem-on-a-<u>C</u>hip.</i>
NTC	<i><u>N</u>egative <u>T</u>emperature <u>C</u>oefficient.</i>
OVP	<i><u>O</u>pen <u>V</u>irtual <u>P</u>latform.</i>
PCHIP	<i><u>P</u>iecewise <u>C</u>ubic <u>H</u>ermite <u>I</u>nterpolation <u>P</u>olynomial.</i>
PDP	<i><u>P</u>ower-<u>d</u>elay <u>P</u>roduct.</i>
PFM	<i><u>P</u>uls<u>f</u>requenz<u>m</u>odulation.</i>
PMC	<i><u>P</u>ower <u>M</u>anageable <u>C</u>omponent.</i>
PPG	<i><u>P</u>hoto<u>p</u>lethys<u>m</u>og<u>r</u>aphie.</i>
PSM	<i><u>P</u>ower <u>S</u>tate <u>M</u>achine.</i>
PWM	<i><u>P</u>uls<u>w</u>eiten<u>m</u>odulation.</i>
RGB	<i><u>R</u>ed <u>G</u>reen <u>B</u>lue.</i>
RIP	<i><u>R</u>espiratorische <u>i</u>nduktive <u>P</u>lethysmografie.</i>
RISC	<i><u>R</u>educed <u>I</u>nstruction <u>S</u>et <u>C</u>omputer.</i>
RMSE	<i><u>R</u>oot <u>M</u>ean <u>S</u>quare <u>E</u>rror.</i>
RTOS	<i><u>R</u>eal-<u>T</u>ime <u>O</u>perating <u>S</u>ystem.</i>
S-MEOP	<i><u>S</u>ystem <u>M</u>inimum <u>E</u>nergy <u>O</u>peration <u>P</u>oint.</i>
SOC	<i><u>S</u>tate-of <u>C</u>harge.</i>
SoC	<i><u>S</u>ystem-on-a-<u>C</u>hip.</i>
SPI	<i><u>S</u>erial <u>P</u>eripheral <u>I</u>nterface.</i>
SPICE	<i><u>S</u>imulation <u>P</u>rogram with <u>I</u>ntegrated <u>C</u>ircuit <u>E</u>mphasis.</i>
SysML	<i><u>S</u>ystems <u>M</u>odeling <u>L</u>anguage.</i>

TPP	<u>T</u> ime- <u>p</u> ower <u>P</u> roduct.
UART	<u>U</u> niversal <u>A</u> synchronous <u>R</u> eceiver <u>T</u> ransmitter.
ULP	<u>U</u> ltra <u>L</u> ow <u>P</u> ower.
UNSPSC	<u>U</u> nited <u>N</u> ations <u>S</u> tandard <u>P</u> roducts and <u>S</u> ervices <u>C</u> ode.
VHDL	<u>V</u> ery <u>H</u> igh <u>S</u> peed <u>I</u> ntegrated <u>C</u> ircuit <u>H</u> ardware <u>D</u> escription <u>L</u> anguage.
WCEC	<u>W</u> orst- <u>C</u> ase <u>E</u> nergy <u>C</u> onsumption.
WCET	<u>W</u> orst- <u>C</u> ase <u>E</u> xecution <u>T</u> ime.
WCPC	<u>W</u> orst- <u>C</u> ase <u>P</u> ower <u>C</u> onsumption.
WLAN	<u>W</u> ireless <u>L</u> ocal <u>A</u> rea <u>N</u> etwork.
WSN	<u>W</u> ireless <u>S</u> ensor <u>N</u> etwork.

Symbolverzeichnis

C_{BAT}	Nominale Kapazität der Batterie.
C_P	Plattenkapazität der Batterie.
C_{PA}	Effektive Kapazität der elektrochemischen Polarisierung im DP-Batteriemodell.
C_{PC}	Effektive Kapazität der Konzentrationspolarisation im DP-Batteriemodell.
C_{PK}	Peukert-Kapazität: Nennkapazität bei einem Entladestrom von 1 A.
C_R	Kapazität des Tiefpasses im Laufzeit-Modell.
D_I	Datenbreite der Eingangsdaten.
D_{IFSi}	Datenbreite des i -ten Sensors inklusive Schnittstelle.
D_{IFUCi}	Datenbreite des i -ten Ausgangs des Mikrocontrollers inklusive Schnittstelle.
D_O	Datenbreite der Ausgangsdaten.
D_{Si}	Datenbreite des i -ten Sensors.
D_{UCi}	Datenbreite des i -ten Ausgangs des Mikrocontrollers.
η	Effizienz des Spannungswandlers.
f_{LOAD}	Frequenz des Entladeprofils einer Batterie.
I_{BAT}	Entladestrom der Batterie.
I_{DC}	Eingangsstrom des Spannungswandlers.
I_N	Nennentladestrom der Batterie.
I_{PA}	Stromfluss der elektrochemischen Polarisierung im DP-Batteriemodell.
I_{PC}	Stromfluss der Konzentrationspolarisation im DP-Batteriemodell.
I_{SYS}	Stromaufnahme des Systems nach dem Spannungswandlers.
k	Dimensionslose Peukert-Zahl.
N_{BAT}	Anzahl der Batterien im Entwurfsraum.
N_{DC}	Anzahl der Spannungswandler im Entwurfsraum.
$N_{\text{EWR,SYS}}$	Größe des Entwurfsraums der Systemkonfigurationen.
N_{IF}	Anzahl der Schnittstellen im Komponentenmodell.
N_{IOUT}	Anzahl der Ausgangsströme im Komponentenmodell des Spannungswandlers.
N_{IOUTp}	Anzahl der Ausgangsströme im Datenblatt des Spannungswandlers.
N_{LT}	Anzahl der Datenpunkte in einem Task (in μs Auflösung).
N_M	Anzahl der Modi im Komponentenmodell.

$N_{M,A}$	Anzahl der aktiven Modi im Komponentenmodell.
N_{MEM}	Anzahl der Speicher im Entwurfsraum.
$N_{M,I}$	Anzahl der inaktiven Modi im Komponentenmodell.
N_{PL}	Anzahl der möglichen Datenbreiten im Komponentenmodell des Transceivers.
$N_{EWR,PMC}$	Größe des Entwurfsraum der PMC.
N_{SEN}	Anzahl der Sensoren im Entwurfsraum.
$N_{SEN,ACC}$	Anzahl der Beschleunigungssensoren im Entwurfsraum.
$N_{SEN,AMB}$	Anzahl der Umgebungssensoren im Entwurfsraum.
$N_{SEN,LUX}$	Anzahl der Umgebungslichtsensoren im Entwurfsraum.
N_{SoC}	Anzahl der SoC im Entwurfsraum.
N_T	Anzahl der Tasks im Entwurfsraum.
N_{TF}	Anzahl der Ausführungsfrequenzen im Entwurfsraum.
N_{TR}	Anzahl der Ausführungsverhältnisse im Entwurfsraum.
N_{UC}	Anzahl der Mikrocontroller im Entwurfsraum.
N_{VIN}	Anzahl der Eingangsspannungen im Komponentenmodell des Spannungswandlers.
P_{AVG}	Mittlere Leistungsaufnahme des Systems.
R_0	Innenwiderstand der Batterie.
R_P	Elektrolytwiderstand der Batterie.
R_{PA}	Effektiver Widerstand der elektrochemischen Polarisation im DP-Batteriemodell.
R_{PC}	Effektiver Widerstand der Konzentrationspolarisation im DP-Batteriemodell.
R_R	Widerstand des Tiefpasses im Laufzeit-Modell.
U_R	Von der Entladerate und der Entladefrequenz abhängige Spannung im Laufzeit-Modell.
τ_{PA}	Zeitkonstante des <i>short-time</i> RC-Netzwerks im DP-Batteriemodell.
τ_{PC}	Zeitkonstante des <i>long-time</i> RC-Netzwerks im DP-Batteriemodell.
T_{BE}	Nutzenzeit (<i>Break-even Time</i>).
N_{TC}	Anzahl der Transceiver im Entwurfsraum.
T_S	Abtastintervall.
T_{SYS}	Laufzeit des Systems.
U_{BAT}	Klemmenspannung der Batterie.
U_{CR}	Von der Entladerate abhängige Spannung im Laufzeit-Modell.
U_{CUT}	Entladeschlussspannung einer Batterie.
U_{SD}	Widerstand der Selbstentladung im Laufzeit-Modell.

U_L	Von der Entladerate abhängige Spannung im Laufzeit-Modell.
U_{OC}	Leerlaufspannung der Batterie, engl.: <i>open circuit voltage</i> .
U_P	Spannung des RC-Netzwerks im Thevenin-Modell, auch als Polarisierungsspannung bezeichnet.
U_{PA}	Spannung über die effektive Kapazität der elektrochemischen Polarisierung im DP-Batteriemodell.
U_{PC}	Spannung über die effektive Kapazität der Konzentrationspolarisation im DP-Batteriemodell.
U_{R0}	Spannungsabfall über den Innenwiderstand der Batterie.
U_{SOC}	Von der Entladerate und dem SOC abhängige Spannung im Laufzeit-Modell.
U_{SYS}	Versorgungsspannung des Systems.

Literaturverzeichnis

- [1] ARM Limited. *ARM Developer Suite - Debug Target Guide*. 2001.
- [2] A. M. Abdal-Kadhim und K. S. Leong. „Electrical power flow of typical wireless sensor node based on energy harvesting approach“. In: *International Journal of Electronics Letters* 8.1 (2020), S. 17–27.
- [3] R. A. Abdallah, P. S. Shenoy, N. R. Shanbhag und 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, S. 97–102.
- [4] Advanticsys. *MTM-CM5000-MSP TelosB mote*. Letzter Zugriff 21.08.2017. URL: <https://www.advanticsys.com/shop/mtmcm5000msp-p-14.html>.
- [5] I. Akyildiz, W. Su, Y. Sankarasubramaniam und E. Cayirci. „Wireless Sensor Networks: A Survey“. In: *Computer Networks* 38.4 (2002), S. 393–422. DOI: [http://dx.doi.org/10.1016/S1389-1286\(01\)00302-4](http://dx.doi.org/10.1016/S1389-1286(01)00302-4).
- [6] J. Allen. „Photoplethysmography and its application in clinical physiological measurement“. In: *Physiological Measurement* 28.3 (2007). URL: <http://stacks.iop.org/0967-3334/28/i=3/a=R01>.
- [7] Ambiotex. *Das smarte Shirt zum Erfassen deiner Vitaldaten*. Letzter Zugriff 25.08.2017. URL: <https://www.ambiotex.com/produkt/ambiotex-starter-package/>.
- [8] Amphenol Advanced Sensors. *Telaire T6713 Series CO₂ Module*. 2015.
- [9] Analog Devices. *ADXL362 - Micropower, 3-Axis, ±2 g/±4 g/±8 g Digital Output MEMS Accelerometer*. Rev. D. 2012.
- [10] S. Anastasova, B. Crewther, P. Bembnowicz, V. Curto, H. M. Ip, B. Rosa und G.-Z. Yang. „A wearable multisensing patch for continuous sweat monitoring“. In: *Biosensors and Bioelectronics* 93 (2017), S. 139–145.
- [11] Apple Inc. *iBeacon - Apple Developer*. Letzter Zugriff 28.08.2017. URL: <https://developer.apple.com/ibeacon/>.
- [12] Athos. *Athos Training System*. Letzter Zugriff 22.08.2017. MAD Apparel, Inc. URL: <https://www.liveathos.com/>.
- [13] L. Atzori, A. Iera und G. Morabito. „The Internet of Things: A Survey“. In: *Computer Networks* 54.15 (2010), S. 2787–2805.
- [14] A. Badam u. a. „Software defined batteries“. In: *Proceedings of the 25th Symposium on Operating Systems Principles*. ACM. 2015, S. 215–229.

- [15] G. Bag, Z. Pang, M. E. Johansson, X. Min und S. Zhu. „Engineering friendly tool to estimate battery life of a wireless sensor node“. In: *Journal of Industrial Information Integration* 4 (2016), S. 8–14. DOI: <https://doi.org/10.1016/j.jii.2016.11.001>.
- [16] M. Bazzaz, M. Salehi und A. Ejlali. „An Accurate Instruction-Level Energy Estimation Model and Tool for Embedded Systems“. In: *IEEE Transactions on Instrumentation and Measurement* 62.7 (Juli 2013), S. 1927–1934. DOI: [10.1109/TIM.2013.2248288](https://doi.org/10.1109/TIM.2013.2248288).
- [17] L. Benini, A. Bogliolo und G. De Micheli. „A survey of design techniques for system-level dynamic power management“. In: *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on* 8.3 (2000), S. 299–316.
- [18] L. Benini, G. Castelli, A. Macii, E. Macii, M. Poncino und R. Scarsi. „Discrete-time battery models for system-level low-power design“. In: *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on* 9.5 (2001), S. 630–640.
- [19] E. Berlin, M. Zittel, M. Braunlein und K. Van Laerhoven. „Low-power lessons from designing a wearable logger for long-term deployments“. In: *Sensors Applications Symposium (SAS), 2015 IEEE*. IEEE. 2015, S. 1–6.
- [20] R. Bez, E. Camerlenghi, A. Modelli und A. Visconti. „Introduction to flash memory“. In: *Proceedings of the IEEE* 91.4 (Apr. 2003), S. 489–502. DOI: [10.1109/JPROC.2003.811702](https://doi.org/10.1109/JPROC.2003.811702).
- [21] C. Bianchi, S. M. Lucich und A. D. Smith. „Influence of weather boundary conditions on building energy modeling“. In: *2015 IEEE Conference on Technologies for Sustainability (SusTech)*. 2015, S. 35–41. DOI: [10.1109/SusTech.2015.7314318](https://doi.org/10.1109/SusTech.2015.7314318).
- [22] Bluegiga Technologies. *BLE112 Data Sheet*. 1.4. 23. Mai 2013.
- [23] Bluegiga Technologies. *BLE113 Preliminary Data Sheet*. 0.4. 1. März 2013.
- [24] A. Bocca, A. Macii, E. Macii und M. Poncino. „Composable Battery Model Templates Based on Manufacturers’ Data“. In: *IEEE Design Test* PP.99 (2017), S. 1–1. DOI: [10.1109/MDAT.2017.2755642](https://doi.org/10.1109/MDAT.2017.2755642).
- [25] L. Bogdanov. „Statement-level energy simulation in embedded systems using GCC“. In: *2016 XXV International Scientific Conference Electronics (ET)*. 2016, S. 1–4. DOI: [10.1109/ET.2016.7753462](https://doi.org/10.1109/ET.2016.7753462).
- [26] L. Bogdanov. „Look-up Table-Based Microprocessor Energy Model“. In: *International Scientific Conference on Engineering, Technologies and Systems TECHSYS*. 2016.
- [27] Bosch Sensortec. *BMA280 - Digital, triaxial acceleration sensor*. BST-BMA280-DS000-11. 1.8. 1. Aug. 2014.

- [28] Bosch Sensortec. *BME280 Combined humidity and pressure sensor*. BST-BME280-DS001-11. 1.2. 26. Okt. 2015.
- [29] Bosch Sensortec. *BMP280 Digital Pressure Sensor*. 2. Nov. 2016.
- [30] Bosch Sensortec. *BMP380 Digital, barometric pressure sensor*. Version_1.0_122016.
- [31] Bosch Sensortec. *BNO055 - Intelligent 9-axis absolute orientation sensor*. 1.4. Juni 2016.
- [32] S. Boschert und R. Rosen. „Digital Twin—The Simulation Aspect“. In: *Mechatronic Futures: Challenges and Solutions for Mechatronic Systems and their Designers*. Hrsg. von P. Hehenberger und D. Bradley. Cham: Springer International Publishing, 2016, S. 59–74. DOI: 10.1007/978-3-319-32156-1_5.
- [33] S. B. Boyd. *Life-Cycle Assessment of Semiconductors*. Springer New York, 2012. DOI: 10.1007/978-1-4419-9988-7.
- [34] BrightVolt Batteries. *BrightVolt PME[®] (Polymer Matrix Electrolyte)*. White Paper. BrightVolt Batteries, 2017.
- [35] *CPUs and NPUs: Processor IP for the widest range of devices - from sensors to servers*. Arm Limited. URL: <https://www.arm.com/products/silicon-ip-cpu> (besucht am 28.01.2021).
- [36] CUI Inc. *Piezo and Magnetic Buzzers*. Letzter Zugriff 16.08.2017. URL: <http://www.cui.com/product-spotlight/piezo-and-magnetic-buzzers>.
- [37] M. Cacciato, G. Nobile, G. Scarcella und G. Scelba. „Real-time model-based estimation of SOC and SOH for energy storage systems“. In: *IEEE Transactions on Power Electronics* 32.1 (2017), S. 794–803.
- [38] M. Chen und G. A. Rincon-Mora. „Accurate electrical battery model capable of predicting runtime and I-V performance“. In: *IEEE Transactions on Energy Conversion* 21.2 (Juni 2006), S. 504–511. DOI: 10.1109/TEC.2006.874229.
- [39] Y. Chen, E. Macii und M. Poncino. „A circuit-equivalent battery model accounting for the dependency on load frequency“. In: *Design, Automation Test in Europe Conference Exhibition (DATE), 2017*. März 2017, S. 1177–1182. DOI: 10.23919/DATE.2017.7927167.
- [40] Y. Chen, E. Macii und M. Poncino. „Workload-driven frequency-aware battery sizing“. In: *2017 IEEE/ACM International Symposium on Low Power Electronics and Design (ISLPED)*. Juli 2017, S. 1–6. DOI: 10.1109/ISLPED.2017.8009196.
- [41] Y. Chen, D. Jahier Pagliari, E. Macii und M. Poncino. „Battery-aware Design Exploration of Scheduling Policies for Multi-sensor Devices“. In: *Proceedings of the 2018 on Great Lakes Symposium on VLSI*. GLSVLSI '18. Chicago, IL, USA: ACM, 2018, S. 201–206. DOI: 10.1145/3194554.3194588.

- [42] Y. Chen, E. Macii und M. Poncino. „Frequency domain characterization of batteries for the design of energy storage subsystems“. In: *2016 IFIP/IEEE International Conference on Very Large Scale Integration (VLSI-SoC)*. Sep. 2016, S. 1–6. DOI: 10.1109/VLSI-SoC.2016.7753548.
- [43] P. Christ, J. Mielebacher, M. Haag und U. Rückert. „Detection of Body Movement and Measurement of Physiological Stress with a Mobile Chest Module in Obesity Prevention“. In: *Proceedings of the 10th Australasian Conference on Mathematics and Computers in Sport*. Hrsg. von A. Bedford und M. Ovens. Juli 2010, S. 67–74. ISBN: 9780957862357.
- [44] P. Christ, B. Neuwinger, F. Werner und U. Rückert. „Performance Analysis of the nRF24L01 Ultra-Low-Power Transceiver in a Multi-Transmitter and Multi-Receiver Scenario“. In: *Proceedings of the 10th IEEE International Conference on SENSORS*. IEEE, Okt. 2011, S. 1205–1208. DOI: 10.1109/ICSENS.2011.6127100.
- [45] P. Christ und U. Rückert. „Ein ressourceneffizienter Körpersensor zur Erfassung physiologischer und kinematischer Parameter“. In: *Informations- und Kommunikationstechnologien in der angewandten Trainingswissenschaft - Abstracts zur 15. Frühjahrsschule des IAT*. Apr. 2013, S. 6–7.
- [46] P. Christ, G. Sievers, J. Einhaus, T. Jungeblut, M. Porrmann und U. Rückert. „Pareto-optimal Signal Processing on Low-Power Microprocessors“. In: *Proceedings of the 12th IEEE International Conference on SENSORS*. Nov. 2013, S. 1843–1846. DOI: 10.1109/ICSENS.2013.6688593.
- [47] P. Christ, F. Werner, U. Rückert und J. Mielebacher. „An Approach for Determining Linear Velocities of Athletes from Acceleration Measurements using a Neural Network“. In: *Proceedings of the 6th IASTED International Conference on Biomechanics*. Hrsg. von B. M. M. Hamza. ACTA Press. Nov. 2011, S. 105–112. DOI: 10.2316/P.2011.751-009.
- [48] P. Christ, F. Werner, U. Rückert und J. Mielebacher. „Athlete Identification using Acceleration and Electrocardiographic Measurements Recorded with a Wireless Body Sensor“. In: *Proceedings of the 6th International Conference on Bio-Inspired Systems and Signal Processing, International Joint Conference on Biomedical Engineering Systems and Technologies*. Hrsg. von A. F. Sergio Alvarez Jordi Solé-Casals und H. Gamboa. SciTePress, Feb. 2013, S. 11–19.
- [49] P. Christ. „Detektion und Analyse physiologischer und biokinematischer Parameter mit Körpersensoren“. Diss. Universität Bielefeld, 2016.
- [50] P. Christ und U. Rückert. „Identification of Athletes During Walking and Jogging Based on Gait and Electrocardiographic Patterns“. In: *Biomedical Engineering Systems and Technologies*. Hrsg. von M. Fernández-Chimeno, P. L. Fernandes, S. Alvarez, D. Stacey, J. Solé-Casals, A. Fred und H. Gamboa. Bd. 452. Communications

- in Computer and Information Science. Springer Berlin Heidelberg, Nov. 2014, S. 240–257. DOI: 10.1007/978-3-662-44485-6_17.
- [51] J. A. Debardelaben, V. K. Madiseti und A. J. Gadiant. „Incorporating cost modeling in embedded-system design“. In: *IEEE Design Test of Computers* 14.3 (1997), S. 24–35. DOI: 10.1109/54.605989.
- [52] *Dimensions*. Digital Science & Research Solutions. 25. Jan. 2021. URL: <https://app.dimensions.ai/discover/publication>.
- [53] D. Doerffel und S. A. Sharkh. „A critical review of using the Peukert equation for determining the remaining capacity of lead-acid and lithium-ion batteries“. In: *Journal of Power Sources* 155.2 (2006), S. 395–400.
- [54] Dynastream Innovations Inc. *ANT Message Protocol and Usage*. Rev 5.1. Apr. 2014.
- [55] EEMBC. *Embedded Microprocessor Benchmark Consortium*. Letzter Zugriff 03.11.2017. URL: <http://http://www.eembc.org>.
- [56] F. Eltoumi, A. Badji, M. Becherif und H. S. Ramadan. „Experimental Identification using Equivalent Circuit Model for Lithium-Ion Battery“. In: *International Journal of Emerging Electric Power Systems* 19.3 ().
- [57] Empatica Inc. *E4 wristband*. Letzter Zugriff 22.08.2017. URL: <https://www.empatica.com/e4-wristband>.
- [58] Energizer. *Energizer CR2032 Product Data Sheet*. Version CR2032MFR.05.
- [59] Espruino. *Puck.js - The ground-breaking bluetooth beacon - An Open Source JavaScript microcontroller you can program and debug wirelessly*. Pur3 Ltd. URL: <https://www.puck-js.com/>.
- [60] J. A. Estefan u. a. „Survey of model-based systems engineering (MBSE) methodologies“. In: *IncoSE MBSE Focus Group* 25.8 (2007), S. 1–12.
- [61] K. R. Evenson, M. M. Goto und R. D. Furberg. „Systematic review of the validity and reliability of consumer-wearable activity trackers“. In: *International Journal of Behavioral Nutrition and Physical Activity* 12.1 (Dez. 2015), S. 159. DOI: 10.1186/s12966-015-0314-1.
- [62] D. Fan, L. L. Ruiz und J. Lach. „Application-driven dynamic power management for self-powered vigilant monitoring“. In: *2018 IEEE 15th International Conference on Wearable and Implantable Body Sensor Networks (BSN)*. 2018, S. 210–213. DOI: 10.1109/BSN.2018.8329695.
- [63] Fitbit, Inc. *Alta HR*. Letzter Zugriff 22.08.2017. URL: <https://www.fitbit.com/de/altahr>.
- [64] Fitbit, Inc. *Flex 2*. Letzter Zugriff 29.08.2017. URL: <https://www.fitbit.com/de/flex2>.

- [65] J. Fraden. *Handbook of Modern Sensors: Physics, Designs, and Applications*. Fourth Edition. Springer New York, 2010.
- [66] Fraunhofer-Institut für Integrierte Schaltungen IIS. *FitnessSHIRT - Verbesserung der Sicherheit durch Telemonitoring*. Letzter Zugriff 25.08.2017. Fraunhofer-Gesellschaft. URL: <https://www.iis.fraunhofer.de/de/ff/sse/mks/prod/fitnessshirt.html>.
- [67] Fraunhofer Venture. *Gesundheitscheck per T-Shirt - Fraunhofer beteiligt sich an ambiotex*. Presseinformation 06.05.2015. Mai 2015.
- [68] Freescale Semiconductor. *MMA8451Q, 3-Axis, 14-bit/8-bit, Digital Accelerometer*. 9.1. 2015.
- [69] S. Friedenthal, A. Moore und R. Steiner. *A practical guide to SysML: the systems modeling language*. Morgan Kaufmann, 2014.
- [70] D. Gajski und R. Kuhn. „New VLSI Tools“. In: *Computer* 16 (Dez. 1983), S. 11–14. DOI: 10.1109/MC.1983.1654264.
- [71] S. Gold. „A PSPICE macromodel for lithium-ion batteries“. In: *The Twelfth Annual Battery Conference on Applications and Advances*. 1997, S. 215–222. DOI: 10.1109/BCAA.1997.574106.
- [72] G. Gridling und B. Weiss. „Introduction to microcontrollers“. In: *Vienna University of Technology Institute of Computer Engineering Embedded Computing Systems Group* (2007).
- [73] Z. Guo u. a. „Multi-functional Flexible Aqueous Sodium-Ion Batteries with High Safety“. In: *Chem* 3.2 (2017), S. 348–362.
- [74] U. Guth. „Solid Electrolytes“. In: *Encyclopedia of Applied Electrochemistry*. Hrsg. von G. Kreysa, K. Ota und R. F. Savinell. New York, NY: Springer New York, 2014, S. 1989–1993. DOI: 10.1007/978-1-4419-6996-5_317.
- [75] H. He, R. Xiong und J. Fan. „Evaluation of lithium-ion battery equivalent circuit models for state of charge estimation by an experimental approach“. In: *energies* 4.4 (2011), S. 582–598.
- [76] H. He, R. Xiong, H. Guo und S. Li. „Comparison study on the battery models used for the energy management of batteries in electric vehicles“. In: *Energy Conversion and Management* 64 (2012). IREC 2011, The International Renewable Energy Congress, S. 113 –121. DOI: <https://doi.org/10.1016/j.enconman.2012.04.014>.
- [77] M. Hempstead, N. Tripathi, P. Mauro, G.-Y. Wei und D. Brooks. „An ultra low power system architecture for sensor network applications“. In: *ACM SIGARCH Computer Architecture News* 33.2 (2005), S. 208–219.

-
- [78] S. Herbrechtsmeier. „Modell eines agilen Leiterplattenentwurfsprozesses basierend auf der interdisziplinären Entwicklung eines modularen autonomen Miniroboters“. Diss. Universität Bielefeld, 2017.
- [79] M. Hesse. „Analyse und Optimierung inertialer Navigationssysteme“. Diplomarbeit. Fachgruppe Schaltungstechnik, Universität Paderborn, Dez. 2010.
- [80] M. Hesse. „Entwicklung eines inertialen Navigationssystems für Miniroboter“. Studienarbeit. Fachgruppe Schaltungstechnik, Universität Paderborn, Apr. 2010.
- [81] *Heterogeneous Integration Roadmap 2019 Edition*. IEEE Electronics Packaging Society, Juli 2020. Kap. 3: Internet of Things (IoT).
- [82] R. Heydon. *Bluetooth low energy: the developer's handbook*. Prentice Hall, 2013.
- [83] Hitachi Maxell, Ltd. *Maxell CR2032H Coin Type Lithium Manganese Dioxide Battery Data Sheet*. Version CR2032MFR.05. 2016.
- [84] T. Hörmann. „Design and Application of Wireless Body Sensors“. Diss. Universität Bielefeld, 2019.
- [85] B. Hübener. „Analyse verschiedener Architekturvarianten des CoreVA-VLIW-Prozessors“. Diss. Universität Bielefeld, 2016.
- [86] IEEE Computer Society. *802.15.14™ IEEE Standard for Low-Rate Wireless Networks*. IEEE Std 802.15.4™-2015. Dez. 2015.
- [87] IEEE Computer Society. *802.15.1™ IEEE Standard for Information technology - Telecommunications and information exchange between systems - Local and metropolitan area networks - Specific requirements. Part 15.1: Wireless medium access control (MAC) and physical layer (PHY) specifications for wireless personal area networks (WPANs)*. IEEE Std 802.15.1™-2005. Juni 2005.
- [88] IEEE Computer Society. *802®- IEEE Standard for Local and Metropolitan Area Networks: Overview and Architecture*. IEEE Std 802®-2014. Juni 2014.
- [89] IEEE Computer Society. *IEEE Standard for Information technology - Telecommunications and information exchange between systems. Local and metropolitan area networks - Specific requirements. Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*. IEEE Std 802.11™-2016. Dez. 2016.
- [90] Imperas Software. *Open Virtual Platforms - OVPSim simulator*. URL: http://www.ovpworld.org/technology_ovpsim.
- [91] Imperas Software. *Open Virtual Platforms - the source of Fast Processor Models & Platforms*. Letzter Zugriff 21.11.2018. URL: <http://www.ovpworld.org/>.
- [92] International Electrotechnical Commission (IEC). *IEC 61360 - Common Data Dictionary (CDD - V2.0014.0016)*. Letzter Zugriff 06.12.2018. URL: <https://cdd.iec.ch/cdd/iec61360/iec61360.nsf>.

- [93] *International Roadmap for Devices and Systems (IRDS™) 2020 Edition*. IEEE, 2020. Kap. Systems and Architectures.
- [94] InvenSense. *ICS-52000 Low-Noise Microphone with TDM Digital Output*. 1.3. Apr. 2017.
- [95] InvenSense. *ITG-3050 Product Specification*. Revision 1.3. Aug. 2011.
- [96] InvenSense. *MPU-9150 Product Specification*. Revision 4.3. Sep. 2013.
- [97] InvenSense. *MPU-9250 Product Specification*. 1.1. 20. Juni 2016.
- [98] R. Jayaseelan, T. Mitra und X. Li. „Estimating the Worst-Case Energy Consumption of Embedded Software“. In: *12th IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS 2006)*. IEEE. DOI: 10.1109/rtas.2006.17.
- [99] U. Jensen, P. Kugler, M. Ring und B. M. Eskofier. „Approaching the accuracy-cost conflict in embedded classification system design“. In: *Pattern Analysis and Applications* (2015), S. 1–17.
- [100] Y. Jiang, C. Zhang, W. Zhang, W. Shi und Q. Liu. „Modeling charge polarization voltage for large lithium-ion batteries in electric vehicles“. In: *Journal of Industrial Engineering and Management* 6.2 (2013), S. 686.
- [101] D. L. Jones. *μCurrent GOLD Multimeter Adapter*. Letzter Zugriff 31.08.2017. EEV-Blog. URL: <http://www.eevblog.com/product/ucurrentgold/>.
- [102] KEIL. *Getting Started - Creating Applications with μVision 4 for 8-bit, 16-bit, and 32-bit Microcontrollers*.
- [103] Keysight Technologies. *N6700 Modular Power System Family*. Aug. 2017.
- [104] S. Kim, C. Brendle, H.-Y. Lee, M. Walter, S. Gloeggler, S. Krueger und S. Leonhardt. „Evaluation of a 433 MHz band body sensor network for biomedical applications“. In: *Sensors* 13.1 (2013), S. 898–917.
- [105] P. Kindt, D. Yunge, R. Diemer und S. Chakraborty. „Precise Energy Modeling for the Bluetooth Low Energy Protocol“. In: *ArXiv e-prints* (März 2014). arXiv: 1403.2919 [cs.NI].
- [106] C. Klarhorst, M. Flasskamp, J. Ax, T. Jungeblut, W. Kelly, M. Pormann und U. Rückert. „Development of Energy Models for Design Space Exploration of Embedded Many-Core Systems“. In: *arXiv preprint arXiv:1801.04242* (2018).
- [107] B. Klass, D. E. Thomas, H. Schmit und D. F. Nagle. „Modeling inter-instruction energy effects in a digital signal processor“. In: *Proceedings of the Power-Driven Microarchitecture Workshop*. 1998.
- [108] R. Klinge. *Das Elektrokardiogramm: Leitfaden für Ausbildung und Praxis*. Georg Thieme Verlag, 2015.

- [109] K. Konno und J. Mead. „Measurement of the separate volume changes of rib cage and abdomen during breathing“. In: *Journal of applied physiology* 22.3 (1967), S. 407–422.
- [110] V. Konstantakos, A. Chatzigeorgiou, S. Nikolaidis und T. Laopoulos. „Energy Consumption Estimation in Embedded Systems“. In: *IEEE Transactions on Instrumentation and Measurement* 57.4 (Apr. 2008), S. 797–804. DOI: 10.1109/TIM.2007.913724.
- [111] G. Kortuem, F. Kawsar, V. Sundramoorthy und D. Fitton. „Smart Objects as Building Blocks for the Internet of Things“. In: *IEEE Internet Computing* 14.1 (Jan. 2010), S. 44–51. DOI: 10.1109/MIC.2009.143.
- [112] H. E. Krogstad. *Digitizing Graphs*. MATLAB Central File Exchange. 3. Sep. 2020. URL: <https://www.mathworks.com/matlabcentral/fileexchange/11640-digitizing-graphs> (besucht am 03.09.2020).
- [113] M. Krunic, M. Popovic, V. Krunic und N. Cetic. „Energy consumption estimation for embedded applications“. In: *Elektronika ir Elektrotehnika* 22.3 (2016), S. 44–49.
- [114] D. Langen. „Abschätzung des Ressourcenbedarfs von hochintegrierten mikroelektronischen Systemen“. Diss. Universität Paderborn, 2005.
- [115] J. Laurent, N. Julien, E. Senn und E. Martin. „Functional level power analysis: An efficient approach for modeling the power consumption of complex processors“. In: *Proceedings of the conference on Design, automation and test in Europe-Volume 1*. IEEE Computer Society. 2004, S. 10666.
- [116] S. Lauwereins, K. Badami, W. Meert und M. Verhelst. „Optimal resource usage in ultra-low-power sensor interfaces through context- and resource-cost-aware machine learning“. In: *Neurocomputing* 169 (2015). Learning for Visual Semantic Understanding in Big Data ESANN 2014 Industrial Data Processing and Analysis, S. 236–245. DOI: <http://dx.doi.org/10.1016/j.neucom.2014.11.077>.
- [117] H. Leutheuser, C. Heyde, K. Roecker, A. Gollhofer und B. Eskofier. „Reference-Free Adjustment of Respiratory Inductance Plethysmography for Measurements during Physical Exercise“. In: *IEEE Transactions on Biomedical Engineering* PP.99 (2017), S. 1–1. DOI: 10.1109/TBME.2017.2675941.
- [118] D. Linden und T. Reddy. *Handbook of Batteries*. 3. Aufl. McGraw-Hill handbooks. McGraw-Hill Education, 2001. ISBN: 9780071414753.
- [119] J. Liu, P. Wang, J. Lin und C. H. Chu. „Model Based Energy Consumption Analysis of Wireless Cyber Physical Systems“. In: *2017 IEEE 3rd international conference on big data security on cloud (bigdatasecurity), IEEE international conference on high performance and smart computing (hpsc), and IEEE international conference on intelligent data and security (ids)*. Mai 2017, S. 219–224. DOI: 10.1109/BigDataSecurity.2017.53.

- [120] K. Liu, K. Li und C. Zhang. „Constrained generalized predictive control of battery charging process based on a coupled thermoelectric model“. In: *Journal of Power Sources* 347 (2017), S. 145–158.
- [121] M. Lüken, B. Penzlin, S. Leonhardt und B. J. E. Misgeld. „Quantification of respiratory sinus arrhythmia using the IPANEMA body sensor network“. In: *2016 IEEE 13th International Conference on Wearable and Implantable Body Sensor Networks (BSN)*. Juni 2016, S. 83–88. DOI: 10.1109/BSN.2016.7516237.
- [122] MATLAB. *version 9.4.0.813654 (R2016a)*. Natick, Massachusetts: The MathWorks Inc., 23. Feb. 2018.
- [123] M. Malik. „Heart rate variability“. In: *Annals of Noninvasive Electrocardiology* 1.2 (1996), S. 151–181.
- [124] B. Martinez, M. Monton, I. Vilajosana und J. D. Prades. „The Power of Models: Modeling Power Consumption for IoT Devices“. In: *IEEE Sensors Journal* 15.10 (Okt. 2015), S. 5777–5789. DOI: 10.1109/JSEN.2015.2445094.
- [125] P. Marwedel. *Embedded System Design: Embedded Systems Foundations of Cyber-Physical Systems, and the Internet of Things*. 3. Aufl. Springer, 2018. DOI: 10.1007/978-3-319-56045-8.
- [126] Maxim Integrated. *MAX44009 - Industry's Lowest-Power Ambient Light Sensor with ADC*. 2011.
- [127] Measurement Specialties. *MS8607-02BA01 PHT Combination Sensor*. 7. Juli 2015.
- [128] D. Meredith, D. Clifton, P. Charlton, J. Brooks, C. Pugh und L. Tarassenko. „Photoplethysmographic derivation of respiratory rate: a review of relevant physiology“. In: *Journal of medical engineering & technology* 36.1 (2012), S. 1–7.
- [129] J. Meyer, H. Meyer und G. von Cölln. „An Energy Measurement System for Characterization of Energy Harvesting Systems“. In: *2018 IEEE 23rd International Conference on Emerging Technologies and Factory Automation (ETFA)*. Bd. 1. IEEE. 2018, S. 1225–1228.
- [130] M. Miezal, B. Taetz und G. Bleser. „Real-time inertial lower body kinematics and ground contact estimation at anatomical foot points for agile human locomotion“. In: *International Conference on Robotics and Automation*. 2017.
- [131] K. Mikhaylov und J. Tervonen. „Evaluation of power efficiency for digital serial interfaces of microcontrollers“. In: *2012 5th International Conference on New Technologies, Mobility and Security (NTMS)*. IEEE. 2012, S. 1–5.
- [132] E. Mitchell, S. Coyle, N. E. O'Connor, D. Diamond und T. Ward. „Breathing feedback system with wearable textile sensors“. In: *2010 International Conference on Body Sensor Networks (BSN)*. IEEE. 2010, S. 56–61.

- [133] N. Moraveji, B. Olson, T. Nguyen, M. Saadat, Y. Khalighi, R. Pea und J. Heer. „Peripheral paced respiration: influencing user physiology during information work“. In: *Proceedings of the 24th annual ACM symposium on User interface software and technology*. ACM. 2011, S. 423–428.
- [134] P. Morgner, C. Müller, M. Ring, B. Eskofier, C. Riess, F. Armknecht und Z. Benenson. „Privacy Implications of Room Climate Data“. In: *European Symposium on Research in Computer Security*. Springer. 2017, S. 324–343.
- [135] G. Mulligan. „The 6LoWPAN architecture“. In: *Proceedings of the 4th workshop on Embedded networked sensors*. ACM. 2007, S. 78–82.
- [136] Z. Nakutis. „Embedded systems power consumption measurement methods overview“. In: *MATAVIMAI 2.44* (2009), S. 29–35.
- [137] Netatmo. *Netatmo Smarte Wetterstation*. Letzter Zugriff 30.08.2017. URL: <https://www.netatmo.com/de-DE/product/weather/weatherstation>.
- [138] Nokia Health. *Steel HR*. Letzter Zugriff 22.08.2017. URL: <https://health.nokia.com/de/de/steel-hr>.
- [139] Nordic Semiconductor ASA. *nRF51822 Product Specification*. v3.3. Juli 2016.
- [140] M. Normann, T. Grethe, A. Schwarz-Pfeiffer und A. Ehrmann. „Textile solutions for energy harvesting and storage“. In: *Technical Textiles International* 60(1):E38-E41 (März 2017).
- [141] D. Nunan, G. R. Sandercock und D. A. Brodie. „A quantitative systematic review of normal values for short-term heart rate variability in healthy adults“. In: *Pacing and Clinical Electrophysiology* 33.11 (2010), S. 1407–1417.
- [142] OMSignal. *OMSignal - Biosensing Clothing - Wearable Clothing*. Letzter Zugriff 29.08.2017. URL: <https://www.omsignal.com/>.
- [143] P. A. Oberg, T. Togawa und F. A. Spelman. *Sensors applications, sensors in medicine and health care*. Bd. 3. John Wiley & Sons, 2006.
- [144] N. Omar, P. V. d. Bossche, T. Coosemans und J. V. Mierlo. „Peukert Revisited - Critical Appraisal and Need for Modification for Lithium-Ion Batteries“. In: *Energies* 6.11 (2013), S. 5625–5641.
- [145] C. Ostrau, J. Homburg, C. Klarhorst, M. Thies und U. Rückert. „Benchmarking Deep Spiking Neural Networks on Neuromorphic Hardware“. In: *Artificial Neural Networks and Machine Learning – ICANN 2020*. Hrsg. von I. Farkaš, P. Masulli und S. Wermter. Cham: Springer International Publishing, 2020, S. 610–621. ISBN: 978-3-030-61616-8.
- [146] C. Ostrau, C. Klarhorst, M. Thies und U. Rückert. „Benchmarking of Neuromorphic Hardware Systems“. In: *Neuro-inspired Computational Elements Workshop (NICE'20), March 17–20, 2020, Heidelberg, Germany*. 2020.

- [147] PUI Audio. *Electromechanical transducers: General Overview and Driving Circuits*. Letzter Zugriff 16.08.2017. URL: <http://www.puiaudio.com/resources-whitepapers-electromechanical-transducers.aspx>.
- [148] J. Pallister, S. Hollis und J. Bennett. „BEEBS: Open benchmarks for energy measurements on embedded platforms“. In: *arXiv preprint arXiv:1308.5174v2* (2013).
- [149] J. Pallister, S. Kerrison, J. Morse und K. Eder. „Data dependent energy modeling for worst case energy consumption analysis“. In: *Proceedings of the 20th International Workshop on Software and Compilers for Embedded Systems*. 2017, S. 51–59.
- [150] S. K. Pandey und K.-H. Kim. „The Relative Performance of NDIR-based Sensors in the Near Real-time Analysis of CO₂ in Air“. In: *Sensors* 7.9 (2007), S. 1683–1696. DOI: 10.3390/s7091683.
- [151] A. Pantelopoulos und N. G. Bourbakis. „A survey on wearable sensor-based systems for health monitoring and prognosis“. In: *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on* 40.1 (2010), S. 1–12.
- [152] C. Park, J. Seo, D. Seo, S. Kim und B. Kim. „Cost-efficient memory architecture design of NAND flash memory embedded systems“. In: *Proceedings 21st International Conference on Computer Design*. IEEE. 2003, S. 474–480.
- [153] M. Pedram und Q. Wu. „Design Considerations for Battery-powered Electronics“. In: *Proceedings of the 36th Annual ACM/IEEE Design Automation Conference*. DAC '99. New Orleans, Louisiana, USA: ACM, 1999, S. 861–866. DOI: 10.1145/309847.310089.
- [154] M. Petricca, D. Shin, A. Bocca, A. Macii, E. Macii und M. Poncino. „An automated framework for generating variable-accuracy battery models from datasheet information“. In: *Proceedings of the 2013 International Symposium on Low Power Electronics and Design*. IEEE Press. 2013, S. 365–370.
- [155] W. Peukert. „Über die Abhängigkeit der Kapazität von der Entladestromstärke bei Bleiakкумуляtoren“. In: *Elektrotechnische Zeitschrift* 20 (1897), S. 20–21.
- [156] N. Piovesan, A. F. Gambin, M. Miozzo, M. Rossi und P. Dini. „Energy sustainable paradigms and methods for future mobile networks: A survey“. In: *Computer Communications* 119 (2018), S. 101–117.
- [157] Plattform Industrie 4.0. *Die Verwaltungsschale im Detail. Von der Idee zum implementierbaren Konzept*. 19. Dez. 2020. URL: https://www.plattform-i40.de/PI40/Redaktion/DE/Downloads/Publikation/verwaltungsschale-im-detail-pr%C3%A4sentation.pdf?__blob=publicationFile&v=14 (besucht am 19.12.2020).
- [158] Polar Electro GmbH Deutschland. *H10 Herzfrequenz-Sensor*. Letzter Zugriff 15.08.2017. URL: https://www.polar.com/de/produkte/accessoires/herzfrequenz_sensor_h10.

- [159] Polar Electro GmbH Deutschland. *H6 Herzfrequenz-Sensor*. Letzter Zugriff 15.08.2017. URL: <https://www.polar.com/de/produkte/accessoires/H6>.
- [160] Polar Electro GmbH Deutschland. *H7 Herzfrequenz-Sensor*. Letzter Zugriff 15.08.2017. URL: <https://www.polar.com/de/produkte/accessoires/H7>.
- [161] A. Prayati, C. Antonopoulos, T. Stoyanova, C. Koulamas und G. Papadopoulos. „A modeling approach on the TelosB WSN platform power consumption“. In: *Journal of Systems and Software* 83.8 (2010). Performance Evaluation and Optimization of Ubiquitous Computing and Networked Systems, S. 1355–1363. DOI: <http://dx.doi.org/10.1016/j.jss.2010.01.015>.
- [162] Precision Microdrives Limited. *Vibration Motors*. Letzter Zugriff 15.08.2017. URL: <https://www.precisionmicrodrives.com/vibration-motors>.
- [163] M. Purnaprajna. „Run-time Reconfigurable Multiprocessors“. Diss. Universität Paderborn, 2010.
- [164] T. Pycke. *Accelerometer to pitch and roll*. Letzter Zugriff 17.08.2017. MAV Blog. URL: <http://tom.pycke.be/mav/69/accelerometer-to-attitude>.
- [165] P. Raghavan, A. Lambrechts, J. Absar, M. Jayapala, F. Catthoor und D. Verkest. „COFFEE: compiler framework for energy-aware exploration“. In: *Lecture Notes in Computer Science* 4917 (2008), S. 193.
- [166] D. Rakhmatov, S. Vrudhula und D. A. Wallach. „A model for battery lifetime analysis for organizing applications on a pocket computer“. In: *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 11.6 (2003), S. 1019–1030. DOI: 10.1109/TVLSI.2003.819320.
- [167] R. I. Ramos-Garcia, F. Da Silva, Y. Kondi, E. Sazonov und L. E. Dunne. „Analysis of a coverstitched stretch sensor for monitoring of breathing“. In: *10th International Conference on Sensing Technology (ICST)*. Nov. 2016, S. 1–6. DOI: 10.1109/ICSensT.2016.7796258.
- [168] R. Rao, S. Vrudhula und D. N. Rakhmatov. „Battery modeling for energy aware system design“. In: *Computer* 36.12 (2003), S. 77–87.
- [169] D. Ratasich, F. Khalid, F. Geissler, R. Grosu, M. Shafique und E. Bartocci. „A Roadmap Toward the Resilient Internet of Things for Cyber-Physical Systems“. In: *IEEE Access* 7 (2019), S. 13260–13283. DOI: 10.1109/ACCESS.2019.2891969.
- [170] Renata SA. *CR2025 MFR Technical Data Sheet*. Version CR2025MFR.05. 2017.
- [171] Renata SA. *CR2032 MFR Technical Data Sheet*. Version CR2032MFR.05. 2017.
- [172] S. K. Rethinagiri, R. Ben Atitallah, J.-L. Dekeyser, E. Senn und S. Niar. „An efficient power estimation methodology for complex RISC processor-based platforms“. In: *Proceedings of the great lakes symposium on VLSI*. ACM. 2012, S. 239–244.

- [173] E. Rodriguez-Villegas, S. Iranmanesh und S. A. Imtiaz. „Wearable Medical Devices: High-Level System Design Considerations and Tradeoffs“. In: *IEEE Solid-State Circuits Magazine* 10.4 (2018), S. 43–52. DOI: 10.1109/MSSC.2018.2867247.
- [174] M. Roth, A. Luppold und H. Falk. „Measuring and Modeling Energy Consumption of Embedded Systems for Optimizing Compilers“. In: *Proceedings of the 21st International Workshop on Software and Compilers for Embedded Systems*. ACM. 2018, S. 86–89.
- [175] STEINEL Vertrieb GmbH. *Air Control PRO Signal*. Letzter Zugriff 30.08.2017. URL: <http://www.steinel-professional.de/de/sensoren/presenzmelder/air-control-pro-signal-ap.html>.
- [176] STMicroelectronics. *LIS2DE - MEMS digital output motion sensor: ultra-low-power high-performance 3-axis "femto" accelerometer*. Rev 2. 2014.
- [177] STMicroelectronics. *LIS331HH - MEMS Digital Output Motion Sensor Ultra Low-Power High Full-Scale 3-Axes Nano Accelerometer*. Okt. 2009.
- [178] STMicroelectronics. *UM1718 - User manual STM32CubeMX for STM32 configuration and initialization C code generation*. Version Rev. 26. 5. Sep. 2018.
- [179] Samsung. *SIMBAND documentation*. Letzter Zugriff 22.08.2017. URL: <https://www.simband.io/documentation/simband-documentation/>.
- [180] B. Schweighofer, K. M. Raab und G. Bresseur. „Modeling of high power automotive batteries by the use of an automated test system“. In: *IEEE Transactions on Instrumentation and Measurement* 52.4 (2003), S. 1087–1091. DOI: 10.1109/TIM.2003.814827.
- [181] E. Senn, J. Laurent, N. Julien und E. Martin. „SoftExplorer: estimation, characterization, and optimization of the power and energy consumption at the algorithmic level“. In: *International workshop on power and timing modeling, optimization and simulation*. Springer. 2004, S. 342–351.
- [182] Sensirion AG. *Datasheet SHT20 - Humidity and Temperature Sensor IC*. Version 4. 2014.
- [183] Sensirion AG. *Datasheet STS3x-DIS*. Version 1. Nov. 2016.
- [184] Sensirion. *Digitale Feuchtesensoren (RH/T)*. Letzter Zugriff 18.08.2017. URL: <https://www.sensirion.com/de/umweltsensoren/feuchtesensoren/>.
- [185] Sensirion. *Hochgenaue digitale Temperatursensoren*. Letzter Zugriff 18.08.2017. URL: <https://www.sensirion.com/de/umweltsensoren/temperatursensoren/>.
- [186] Z. Shelby, K. Hartke und C. Bormann. *The constrained application protocol (CoAP)*. Techn. Ber. 2014.
- [187] Shimmer. *Shimmer3 ECG Unit*. Letzter Zugriff 25.08.2017. URL: <http://www.shimmersensing.com/products/shimmer3-ecg-unit>.

- [188] T. Simunic, L. Benini und G. De Micheli. „Energy-efficient design of battery-powered embedded systems“. In: *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on* 9.1 (Feb. 2001), S. 15–28. DOI: 10.1109/92.920814.
- [189] T. Simunic, L. Benini und G. De Micheli. „Cycle-accurate simulation of energy consumption in embedded systems“. In: *Proceedings of the 36th annual ACM/IEEE Design Automation Conference*. ACM. 1999, S. 867–872.
- [190] Solid State Electronics Center. *Magnetic Sensor Products HMC / HMR Series*. Honeywell Inc.
- [191] P. Spies, M. Pollak und L. Mateu. *Handbook of energy harvesting power supplies and applications*. CRC Press, 2015.
- [192] S. Steinke, M. Knauer, L. Wehmeyer und P. Marwedel. „An accurate and fine grain instruction-level energy model supporting software optimizations“. In: *Proceedings of PATMOS*. Citeseer. 2001.
- [193] K. Sun, T.-S. Wei, B. Y. Ahn, J. Y. Seo, S. J. Dillon und J. A. Lewis. „3D printing of interdigitated Li-Ion microbattery architectures“. In: *Advanced Materials* 25.33 (2013), S. 4539–4543.
- [194] M. Swan. „Sensor Mania! The Internet of Things, Wearable Computing, Objective Metrics, and the Quantified Self 2.0“. In: *Journal of Sensor and Actuator Networks* 1.3 (2012), S. 217. DOI: 10.3390/jсан1030217.
- [195] A. Tal. *Two Technologies Compared: NOR vs. NAND*. White Paper 91-SR-012-04-8L, Rev 1.1. M-Systems, Juli 2003.
- [196] C. Terwiesch und Yi Xu. „The copy-exactly ramp-up strategy: trading-off learning with process change“. In: *IEEE Transactions on Engineering Management* 51.1 (2004), S. 70–84. DOI: 10.1109/TEM.2003.822465.
- [197] Texas Instruments. *CC2541 - 2.4-GHz Bluetooth low energy and Proprietary System-on-Chip*. SWRS158B. Juli 2013.
- [198] Texas Instruments. *CC2650 SimpleLink™ Multistandard Wireless MCU*. SWRS158B. Juli 2016.
- [199] Texas Instruments. *HDC1000 Low Power, High Accuracy Digital Humidity Sensor with Temperature Sensor*. SNAS643C. Jan. 2016.
- [200] Texas Instruments. *INA219 Zero-Drift, Bidirectional Current/Power Monitor With I2C Interface*. Dez. 2015.
- [201] Texas Instruments. *OPT3001 Ambient Light Sensor (ALS)*. SBOS681B. Dez. 2014.
- [202] Texas Instruments. *OPT3006 Ultra-Thin Ambient Light Sensor*. Initial release. Okt. 2016.

- [203] Texas Instruments. *Simplelink SensorTag - IoT made easy*. Letzter Zugriff 19.08.2017. URL: http://www.ti.com/ww/en/wireless_connectivity/sensortag/index.html.
- [204] Texas Instruments. *TMP007 Infrared Thermopile Sensor with Integrated Math Engine*. SBOS685C. Juli 2015.
- [205] Texas Instruments. *TPS6220x High-Efficiency, SOT23 Step-Down, DC-DC Converter*. SLVS417F. 2015.
- [206] Texas Instruments. *TPS6224x 2.25-MHz 300-mA Step-Down Converter in 2 x 2 WSON and SOT Package*. SLVS762C. 2018.
- [207] Texas Instruments. *WEBENCH[®] Power Designer*. Letzter Zugriff 08.11.2018. URL: <http://www.ti.com/tools-software/design-center/webench-power-designer.html>.
- [208] N. Tian, Y. Wang, J. Chen und H. Fang. „On parameter identification of an equivalent circuit model for lithium-ion batteries“. In: *2017 IEEE Conference on Control Technology and Applications (CCTA)*. 2017, S. 187–192. DOI: 10.1109/CCTA.2017.8062461.
- [209] V. Tiwari, S. Malik und A. Wolfe. „Power analysis of embedded software: a first step towards software power minimization“. In: *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 2.4 (1994), S. 437–445. DOI: 10.1109/92.335012.
- [210] A. Tobola, H. Leutheuser, B. Schmitz, C. Hofmann, M. Struck, C. Weigand, B. M. Eskofier und G. Fischer. „Battery runtime optimization toolbox for wearable biomedical sensors“. In: *2016 IEEE 13th International Conference on Wearable and Implantable Body Sensor Networks (BSN)*. Juni 2016, S. 199–204. DOI: 10.1109/BSN.2016.7516259.
- [211] A. Tobola und C. Weigand. *Garment for detecting respiratory movement*. US Patent App. 12/742,260. 2010. URL: <https://www.google.ch/patents/US20100286546>.
- [212] A. Tobola, S. Ershov, U. Wissendheit und R. Couronne. *Device and method for sensing respiration of a living being*. US Patent App. 12/574,603. Okt. 2009.
- [213] A. Tobola, H. Leutheuser, M. Pollak, P. Spies, C. Hofmann, C. Weigand, B. M. Eskofier und G. Fischer. „Self-powered Multiparameter Health Sensor“. In: *IEEE Journal of Biomedical and Health Informatics* (2017).
- [214] A. Tobola u. a. „Sampling rate impact on energy consumption of biomedical signal processing systems“. In: *2015 IEEE 12th International Conference on Wearable and Implantable Body Sensor Networks (BSN)*. IEEE. 2015, S. 1–6.
- [215] A. Tobola u. a. „Scalable ECG hardware and algorithms for extended runtime of wearable sensors“. In: *Medical Measurements and Applications (MeMeA), 2015 IEEE International Symposium on*. IEEE. 2015, S. 255–260.

- [216] A. Tobola u. a. „System Design Impacts on Battery Runtime of Wearable Medical Sensors“. In: *The proceedings from the mobilemed 2014 conference*. 2014.
- [217] H.-R. Tränkler und L. M. Reindl. *Sensortechnik: Handbuch für Praxis und Wissenschaft*. Springer-Verlag, 2015.
- [218] O. Trescases und Y. Wen. „A survey of light-load efficiency improvement techniques for low-power DC-DC converters“. In: *8th International Conference on Power Electronics - ECCE Asia*. Mai 2011, S. 326–333. DOI: 10.1109/ICPE.2011.5944617.
- [219] D. Trilla, C. Hernandez, J. Abella und F. J. Cazorla. „Worst-Case Energy Consumption: A New Challenge for Battery-Powered Critical Devices“. In: *IEEE Transactions on Sustainable Computing* (2019), S. 1–1. DOI: 10.1109/tsusc.2019.2943142.
- [220] S. Vinco, Y. Chen, F. Fummi, E. Macii und M. Poncino. „A Layered Methodology for the Simulation of Extra-Functional Properties in Smart Systems“. In: *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 36.10 (2017), S. 1702–1715. DOI: 10.1109/TCAD.2017.2650980.
- [221] VivoSense. *Hexoskin smart shirt*. Letzter Zugriff 25.08.2017. URL: <https://www.vivosense.com/portfolio/hexoskin/>.
- [222] P. Wägemann, T. Distler, T. Hönig, H. Janker, R. Kapitza und W. Schröder-Preikschat. „Worst-case energy consumption analysis for energy-constrained embedded systems“. In: *Real-Time Systems (ECRTS), 2015 27th Euromicro Conference on*. IEEE, 2015, S. 105–114.
- [223] Wahoo Fitness. *TICKR Herzfrequenzmesser*. Letzter Zugriff 22.08.2017. URL: <http://de-eu.wahoofitness.com/devices/heart-rate-monitors>.
- [224] T. Weilkens, J. G. Lamm, S. Roth und M. Walker. *Model-based system architecture*. John Wiley & Sons, 2015.
- [225] M. Wens und M. Steyaert. *Design and implementation of fully-integrated inductive DC-DC converters in standard CMOS*. Springer Science & Business Media, 2011.
- [226] J. Wienke. „Framework-level resource awareness in robotics and intelligent systems. Improving dependability by exploiting knowledge about system resources“. Diss. Universität Bielefeld, 2018.
- [227] P. Wilhelm. „IT-gestützte Leistungsmessung im Sport : System und Anwendung“. Diss. Universität Bielefeld, 2011.
- [228] R. Willmann und W. Kastner. „Product Ramp-up for Semiconductor Manufacturing Automated Recommendation of Control System Setup“. In: *Semantic Web Technologies for Intelligent Engineering Applications*. Springer International Publishing, 2016, S. 219–255. DOI: 10.1007/978-3-319-41490-4_9.
- [229] Winbond. *W25X40CL 2.5/3/3.3V 4M-Bit Serial Flash Memory with 4KB Sectors and Dual I/O SPI*. 15. Okt. 2012.

- [230] Xiaomi. <http://www.mi.com/en/miband2/>. Letzter Zugriff 22.08.2017. URL: <http://www.mi.com/en/miband2/>.
- [231] L. D. Xu, W. He und S. Li. „Internet of Things in Industries: A Survey“. In: *IEEE Transactions on Industrial Informatics* 10.4 (Nov. 2014), S. 2233–2243. DOI: 10.1109/TII.2014.2300753.
- [232] N. Yazdi, F. Ayazi und K. Najafi. „Micromachined inertial sensors“. In: *Proceedings of the IEEE* 86.8 (1998), S. 1640–1659.
- [233] Z-Wave Alliance. *Application Note: Z-Wave Networking Basics*. Aug. 2016.
- [234] S. Zehe. „BRIX₂ - A Versatile Toolkit for Rapid Prototyping and Education in Ubiquitous Computing“. Diss. Universität Bielefeld, 2018.
- [235] S. Zehe, T. Großhauser und T. Hermann. „BRIX - An Easy-to-Use Modular Sensor and Actuator Prototyping Toolkit“. In: *Tenth Annual IEEE International Conference on Pervasive Computing and Communications, Workshop Proceedings*. Lugano, Switzerland: IEEE, 2012, S. 817–822. DOI: 10.1109/PerComW.2012.6197624.
- [236] Zephyr. *Zephyr BioPatch Monitoring Device for Human Performance (HP)*. Medtronic. Nov. 2016.
- [237] C. Zhang, L. Y. Wang, X. Li, W. Chen, G. G. Yin und J. Jiang. „Robust and Adaptive Estimation of State of Charge for Lithium-Ion Batteries“. In: *IEEE Transactions on Industrial Electronics* 62.8 (2015), S. 4948–4957. DOI: 10.1109/TIE.2015.2403796.
- [238] N. Zhao. „Full-Featured Pedometer Design Realized with 3-Axis Digital Accelerometer“. In: *Analog Dialogue* 44.06 (Juni 2010), S. 1–5.
- [239] eCl@ss e.V. *eCl@ss Classification and Product Description*. 21. Juli 2020. URL: <https://www.eclassecontent.com/>.
- [240] movisens GmbH. *EcgMove 3 - EKG- und Aktivitätssensor*. Letzter Zugriff 25.08.2017. URL: <https://www.movisens.com/de/produkte/ekg-und-aktivitaetssensor-ecgmove-3/>.
- [241] ruuvi innovations LTD (OY). *RuuviTag*. Letzter Zugriff 28.08.2017. URL: <https://tag.ruuvi.com/>.
- [242] M. vor dem Berge, G. Da Costa, M. Jarus, A. Oleksiak, W. Piatek und E. Volk. „Modeling Data Center Building Blocks for Energy-Efficiency and Thermal Simulations“. In: *Energy-Efficient Data Centers: Second International Workshop, E²DC 2013, Berkeley, CA, USA, May 21, 2013. Revised Selected Papers*. Hrsg. von S. Klingert, X. Hesselbach-Serra, M. P. Ortega und G. Giuliani. Berlin, Heidelberg: Springer Berlin Heidelberg, 2014, S. 66–82. DOI: 10.1007/978-3-642-55149-9_6.

Eigene Veröffentlichungen

- [243] M. Adams, M. Hesse, T. Hörmann und U. Rückert. „Visuelles Sensorsystem für die Trainings- und Spielunterstützung im Leistungshandball“. In: *Informations- und Kommunikationstechnologien in der angewandten Trainingswissenschaft - 19. Frühjahrsschule des IAT*. 2018.
- [244] M. Adams u. a. „Towards an SSVEP-BCI Controlled Smart Home“. In: *2019 IEEE International Conference on Systems, Man and Cybernetics (SMC)*. Bari, Italien: IEEE, 2019. DOI: 10.1109/smc.2019.8914668.
- [245] U. Damerow, M. Borzykh, D. Tabakajew, W. Schaermann, M. Hesse, W. Homberg, A. Trächtler, T. Jungeblut und J. S. Michels. „Intelligente Biegeverfahren“. In: *wt Werkstattstechnik online* 2015.6 (2015), S. 427–432.
- [246] N. Giard, P. Hensen, M. Hesse, N. Löken, Z. Nouri, J. Reddehase, M. Reimann, K. Schlicher und J. Schulte. *Technical Report. Interdisziplinäre Befragung von Crowdworkern*. Techn. Ber. 2019, S. 36. DOI: 10.4119/unibi/2936990.
- [247] P. Hensen, M. Hesse, N. Löken und Z. Nouri. „Platform-independent Reputation and Qualification System for Crowdwork“. In: *2nd Crowdworking Symposium*. Universität Paderborn, 2020. URL: <https://pub.uni-bielefeld.de/record/2948587>.
- [248] M. Hesse, A. F. Krause, L. Vogel, B. Chamadiya, M. Schilling, T. Schack und T. Jungeblut. „A Connected Chair as Part of a Smart Home Environment“. In: *2017 IEEE 14th International Conference on Wearable and Implantable Body Sensor Networks (BSN)*. Mai 2017, S. 47–50. DOI: 10.1109/BSN.2017.7936004.
- [249] M. Hesse, A. F. Krause, L. Vogel, T. Schack und T. Jungeblut. „KogniCoach - Entwicklung und Evaluierung eines in das Wohnumfeld integrierten persönlichen Trainers mit vernetztem Sessel“. In: *Konferenzband der Dritten Transdisziplinären Konferenz Technische Unterstützungssysteme, die die Menschen wirklich wollen*. Hrsg. von R. Weidner und A. Karafllidis. 2018, S. 235–243. ISBN: 978-3-86818-246-3.
- [250] M. Hesse, M. Adams, T. Hörmann und U. Rückert. „Towards a Comprehensive Power Consumption Model for Wireless Sensor Nodes“. In: *2016 IEEE 13th International Conference on Wearable and Implantable Body Sensor Networks (BSN)*. Juni 2016, S. 390–395. DOI: 10.1109/BSN.2016.7516293.
- [251] M. Hesse, T. Born und T. Jungeblut. „Modellgestützter Entwurf eines Multisensormoduls für die Gebäudeautomation“. In: S.-O. Schulze. *Tag des Systems Engineering: Paderborn, 8. -10. November 2017*. Hrsg. von C. Tschirner, R. Kaffenberger und S. Ackva. Carl Hanser Verlag GmbH Co KG, 6. Nov. 2017. ISBN: 3446455469.

- [252] M. Hesse, P. Christ, T. Hörmann, M. Adams und U. Rückert. „Die Entwicklung zukünftiger körpernaher Sensorsysteme für die autarke und mobile Trainingsunterstützung“. In: *Informations- und Kommunikationstechnologien in der angewandten Trainingswissenschaft - 17. Frühjahrsschule des IAT*. 2015.
- [253] M. Hesse, P. Christ, T. Hörmann und U. Rückert. „A Respiration Sensor for a Chest-Strap Based Wireless Body Sensor“. In: *Proceedings of the 13th IEEE International Conference on SENSORS*. IEEE, Nov. 2014, S. 490–493. DOI: 10.1109/ICSENS.2014.6985042.
- [254] J. D. Homburg, M. Adams, M. Thies, T. Korthals, M. Hesse und U. Rückert. „Constraint Exploration of Convolutional Network Architectures with Neuroevolution“. In: *Advances in Computational Intelligence. IWANN 2019*. Hrsg. von I. Rojas, G. Joya und A. Catala. Bd. 11507. Gran Canaria, Spain: Springer, 2019, S. 735–746. DOI: 10.1007/978-3-030-20518-8_61.
- [255] T. Hörmann, P. Christ, M. Hesse und U. Rückert. „Robust Estimation of Physical Activity by Adaptively Fusing Multiple Parameters“. In: *2015 IEEE 12th International Conference on Wearable and Implantable Body Sensor Networks (BSN)*. Juni 2015, S. 1–6. DOI: 10.1109/BSN.2015.7299390.
- [256] T. Hörmann, M. Hesse, M. Adams und U. Rückert. „A Software Assistant for User-Centric Calibration of a Wireless Body Sensor“. In: *2016 IEEE 13th International Conference on Wearable and Implantable Body Sensor Networks (BSN)*. IEEE. 2016, S. 183–188.
- [257] T. Hörmann, M. Hesse, P. Christ, M. Adams, C. Menßen und U. Rückert. „Detailed Estimation of Cognitive Workload with Reference to a Modern Working Environment“. In: *Biomedical Engineering Systems and Technologies: 9th International Joint Conference, BIOSTEC 2016, Rome, Italy, February 21–23, 2016, Revised Selected Papers*. Hrsg. von A. Fred und H. Gamboa. Cham: Springer International Publishing, 2017, S. 205–223. DOI: 10.1007/978-3-319-54717-6_12.
- [258] T. Hörmann, M. Hesse, P. Christ, M. Adams, C. Menßen und U. Rückert. „Fine-Grained Prediction of Cognitive Workload in a Modern Working Environment by Utilizing Short-Term Physiological Parameters“. In: *Proceedings of the 9th International Conference on Bio-Inspired Systems and Signal Processing, International Joint Conference on Biomedical Engineering Systems and Technologies*. 2016.
- [259] T. Korthals, J. Exner, T. Schöpping und M. Hesse. „Path Evaluation via HMM on Semantical Occupancy Grid Maps“. In: *ICRA 2018 Workshop on Informative Path Planning and Adaptive Sampling*. 2018.
- [260] T. Korthals, J. Exner, T. Schöpping und M. Hesse. „Semantical Occupancy Grid Mapping Framework“. In: *European Conference on Mobile Robotics*. Paris, FR, 2017.

- [261] T. Korthals, M. Hesse, J. Leitner, A. Melnik und U. Rückert. „Jointly Trained Variational Autoencoder for Multi-Modal Sensor Fusion“. In: *22st International Conference on Information Fusion, (FUSION) 2019, Ottawa, CA, July 2-5, 2019*. 2019, S. 1–8. URL: <https://nbn-resolving.org/urn:nbn:de:0070-pub-29375217>, <https://pub.uni-bielefeld.de/record/2937521>.
- [262] T. Korthals, A. Melnik, M. Hesse und J. Leitner. „Multisensory Assisted In-hand Manipulation of Objects with a Dexterous Hand“. In: *2019 IEEE International Conference on Robotics and Automation Workshop on Integrating Vision and Touch for Multimodal and Cross-modal Perception, (ViTac) 2019, Montreal, CA, May 20-25, 2019* (2019), S. 1–2. URL: <https://nbn-resolving.org/urn:nbn:de:0070-pub-29375195>, <https://pub.uni-bielefeld.de/record/2937519>.
- [263] T. Korthals, D. Rudolph, J. Leitner, M. Hesse und U. Rückert. „Multi-Modal Generative Models for Learning Epistemic Active Sensing“. In: *2019 IEEE International Conference on Robotics and Automation*. Montreal, Canada: IEEE, 2019. URL: <https://pub.uni-bielefeld.de/record/2934957>.
- [264] T. Korthals, D. Wolf, D. Rudolph, M. Hesse und U. Rückert. „Fiducial Marker based Extrinsic Camera Calibration for a Robot Benchmarking Platform“. In: *European Conference on Mobile Robots, ECMR 2019, Prague, CZ, September 4-6, 2019*. 2019, S. 1–6. URL: <https://nbn-resolving.org/urn:nbn:de:0070-pub-29375116>, <https://pub.uni-bielefeld.de/record/2937511>.
- [265] C. Lian Sang, M. Adams, M. Hesse, T. Hörmann, T. Korthals und U. Rückert. „A Comparative Study of UWB-based True-Range Positioning Algorithms using Experimental Data“. In: Bremen, Germany: IEEE, 2019. DOI: 10.1109/WPNC47567.2019.8970249.
- [266] C. Lian Sang, M. Adams, T. Hörmann, M. Hesse, M. Porrman und U. Rückert. „An Analytical Study of Time of Flight Error Estimation in Two-Way Ranging Methods“. In: *Proceedings of the 9th International Conference on Indoor Positioning and Indoor Navigation (IPIN 2018)*. 2018.
- [267] C. Lian Sang, M. Adams, T. Hörmann, M. Hesse, M. Porrman und U. Rückert. „Numerical and Experimental Evaluation of Error Estimation for Two-Way Ranging Methods“. In: *Sensors* 19.3 (2019). DOI: 10.3390/s19030616.
- [268] C. Lian Sang, M. Adams, T. Korthals, T. Hörmann, M. Hesse und U. Rückert. „A Bidirectional Object Tracking and Navigation System using a True-Range Multilateration Method“. In: Pisa, Italy: IEEE, 2019. DOI: 10.1109/IPIN.2019.8911811.
- [269] C. Lian Sang, M. Hesse, S. Zehe, M. Adams, T. Hörmann und U. Rückert. „An Adaptive Acknowledgement On-Demand Protocol for Wireless Sensor Networks“. In: *Proceedings of the 6th International Conference on Sensor Networks*. Bd. 1. Porto, Portugal, 2017, S. 174–181.

- [270] C. Lian Sang, B. Steinhagen, J. D. Homburg, M. Adams, M. Hesse und U. Rückert. „Identification of NLOS and Multi-path Conditions in UWB Localization using Machine Learning Methods“. In: *Applied Sciences* 10.11 (2020). DOI: 10.3390/app10113980.
- [271] W. Schaermann, M. Borzykh, A. Trächtler, D. Tabakajew, U. Damerow, W. Homberg, M. Hesse und T. Jungeblut. „Selbstkorrigierende Biegeprozesse in der Umformtechnik“. In: *Automation 2014 Smart X - powered by automation*. Baden-Baden: VDI-Verlag, 2014.
- [272] T. Schöpping, T. Korthals, M. Hesse und U. Rückert. „AMiRo: A Mini Robot as Versatile Teaching Platform“. In: *Proceedings of the 9th International Conference on Robotics in Education. Advances in Intelligent Systems and Computing*. 2018.
- [273] T. Schöpping, T. Korthals, M. Hesse und U. Rückert. „Generic Architecture for Modular Real-Time Systems in Robotics“. In: *Proceedings of the 15th International Conference on Informatics in Control, Automation and Robotics (ICINCO)*. 2018.
- [274] F. Wittenfeld, M. Hesse und T. Jungeblut. „Optische Vermessung bewegter Rotationskörper in industriellen Fertigungsanlagen“. In: *Wissenschaftsforum Intelligente Technische Systeme (WinTeSys) 2017*. Hrsg. von E. Bodden, F. Dressler, R. Dumitrescu, J. Gausemeier, F. Meyer auf der Heide, C. Scheytt und A. Trächtler. Bd. 369. Heinz Nixdorf Institut, Universität Paderborn, 2017, S. 241–253.

Betreute Arbeiten

- [275] M. Adams. „Entwicklung und Evaluation eines BLE-Multitransmitterszenarios für ein Vitalparametermonitoring“. Masterarbeit. Arbeitsgruppe Kognitronik und Sensorik, Universität Bielefeld, Jan. 2015.
- [276] F. Bohr. „Textiler Atemsensor für einen drahtlosen Sensorknoten“. Projektarbeit. Arbeitsgruppe Kognitronik und Sensorik, Universität Bielefeld, 2017.
- [277] M. Bungenstock. „Entwicklung eines minimalistischen Sensorknotens“. Bachelorarbeit. Arbeitsgruppe Kognitronik und Sensorik, Universität Bielefeld, Sep. 2016.
- [278] N. Diller. „Evaluierung aktueller Inertialsensorik für den Einsatz in körpernahen Sensorsystemen“. Projektarbeit. Arbeitsgruppe Kognitronik und Sensorik, Universität Bielefeld, Jan. 2016.
- [279] H. Eser. „Energy Harvesting für drahtlose Sensorknoten“. Projektarbeit. Arbeitsgruppe Kognitronik und Sensorik, Universität Bielefeld, 2017.
- [280] R. Ewers. „Implementierung einer Dual Transceiver Lösung“. Bachelorarbeit. Arbeitsgruppe Kognitronik und Sensorik, Universität Bielefeld, 2017.
- [281] M. Heinz. „Ein inertiales Messsystem zur ganzkörperlichen Bewegungserfassung“. Masterarbeit. Arbeitsgruppe Kognitronik und Sensorik, Universität Bielefeld, Juni 2015.
- [282] F. Hermann. „Effiziente Signalverarbeitung für den BI-Vital Körpersensor“. Bachelorarbeit. Arbeitsgruppe Kognitronik und Sensorik, Universität Bielefeld, 2018.
- [283] T. Hörmann. „Experimentelle Leistungsanalyse des BG-V4 Sensorknotens“. Bachelorarbeit. Arbeitsgruppe Kognitronik und Sensorik, Universität Bielefeld, Sep. 2012.
- [284] T. Hörmann. „Realisierung eines multidimensionalen Aktivitätsmaßes mit Hilfe eines Körpersensors“. Masterarbeit. Arbeitsgruppe Kognitronik und Sensorik, Universität Bielefeld, Sep. 2014.
- [285] K. Hunke. „Thermoelektrisches Energy Harvesting“. Projektarbeit. Arbeitsgruppe Kognitronik und Sensorik, Universität Bielefeld, 2018.
- [286] R. Jasmann. „Aufbau eines adaptiven Sensornetzes“. Bachelorarbeit. Arbeitsgruppe Kognitronik und Sensorik, Universität Bielefeld, Juli 2016.
- [287] C. Jürgens. „Kooperative Infrarot-basierte Hindernisdetektion und Roboteridentifikation“. Masterarbeit. Arbeitsgruppe Kognitronik und Sensorik, Universität Bielefeld, Nov. 2012.
- [288] N. Kaden. „Entwicklung und Evaluation eines Systems zur sensorgestützten Zahnreinigung“. Masterarbeit. Arbeitsgruppe Kognitronik und Sensorik, Universität Bielefeld, Dez. 2019.

- [289] L. Kohlhase. „Automatic Prediction of Power Consumption of Microcontroller Algorithms“. Projektarbeit. Arbeitsgruppe Kognitronik und Sensorik, Universität Bielefeld, Dez. 2017.
- [290] T. Krahn. „Weiterentwicklung eines minimalen Sensorknotens“. Projektarbeit. Arbeitsgruppe Kognitronik und Sensorik, Universität Bielefeld, Dez. 2017.
- [291] S. Lemling. „Erfassung der Herzschlagvariabilität durch einen handelsüblichen Herzfrequenzsensor“. Projektarbeit der Lehrveranstaltung IT im Sport. Arbeitsgruppe Kognitronik und Sensorik, Universität Bielefeld, Mai 2015.
- [292] C. Menßen. „Multifaktorielle online Stresserkennung mit einem mobilen Endgerät“. Masterarbeit. Arbeitsgruppe Kognitronik und Sensorik, Universität Bielefeld, Okt. 2014.
- [293] J.-E. Peitzmeier. „Verfahren zur automatischen Konfiguration von Profilmantelungsmaschinen“. Masterarbeit. Arbeitsgruppe Kognitronik und Sensorik, Universität Bielefeld, Okt. 2011.
- [294] D. Stöber. „Konzeption und Entwicklung eines 3D Echtzeitlokalisierungssystems auf Basis der Funktechnologie Ultrabreitband“. Bachelorarbeit. Arbeitsgruppe Kognitronik und Sensorik, Universität Bielefeld, Apr. 2018.
- [295] L. Stüssel. „Konzeption und Entwicklung eines Echtzeitlokalisierungssystems zu Lokalisierung von Fahrzeugen auf sehr großen Geländen“. Bachelorarbeit. Arbeitsgruppe Kognitronik und Sensorik, Universität Bielefeld, Jan. 2017.
- [296] F. Wittenfeld. „Optische Vermessung bewegter Rotationskörper in industriellen Fertigungsanlagen“. Masterarbeit. Arbeitsgruppe Kognitronik und Sensorik, Universität Bielefeld, Mai 2016.

A Übersicht drahtloser Sensorknoten

Anhang A. Übersicht drahtloser Sensorknoten

Tabelle A.1.: Überblick der Körpersensoren: Armbänder. Mit x markierte Positionen sagen aus, dass der entsprechende Sensor bzw. der Parameter auf dem System zur Verfügung steht.

Hersteller Bezeichnung	Samsung SIMBAND	Empatica E4	Fitbit Flex 2	FitBit ALta HR	Nokia Steel HR	Xiaomi Mi Band 2
EKG	X	-	-	-	-	-
PPG	X	X	-	X	X	X
Atmung	-	-	-	-	-	-
Beschleunigung	X	X	X	X	X	X
Gyroskop	-	-	-	-	-	-
Magnetometer	-	-	-	-	-	-
Hautleitwert	X	X	-	-	-	-
Temperatur	X	X	-	-	-	-
Luftdruck	-	-	-	-	-	-
Luftfeuchtigkeit	-	-	-	-	-	-
EMG	-	-	-	-	-	-
Bioimpedanz	X	-	-	-	-	-
Funk	WLAN, BLE	BLE	BLE	BLE	BLE	BLE
Speicher	x	x	x	x	x	x
Laufzeit	n.b.	60 h	n.b.	720 h	n.b.	n.b.
Batterie	n.b.	LiPo	LiPo	LiPo	n.b.	LiPo
Laufzeit	30 h	36 h	120 h	168 h	600 h	480 h
Indikation	Display Vibration	LED	Display Vibration	Display Vibration	Display Vibration	Display Vibration
Rohdaten	X	X	-	-	-	-
HR	X	X	-	X	X	X
HRV	X	X	-	-	-	-
Atemfrequenz	-	-	-	-	-	-
Postur	-	-	-	-	-	-
Schritte	X	-	X	X	X	X
Ganganalyse	-	-	-	-	-	-
Aktivität	X	X	X	X	X	-
PAT	X	-	-	-	-	-
Muskelaktivität	-	-	-	-	-	-
Schlafanalyse	-	-	X	X	X	X
Max. Beschl.	-	-	-	-	-	-
Preis (19.08.2017)	n.v.	1436,16€	79,95€	149,95€	189,95€	17,95€

Tabelle A.2.: Überblick der Körpersensoren: Herzfrequenzsensoren. Mit x markierte Positionen sagen aus, dass der entsprechende Sensor auf dem System zur Verfügung steht.

Hersteller Bezeichnung	Polar H6	Polar H7	Polar H10	Wahoo TICKRx	Athos Core	VivoSense Hexoskin	Ambiotex TechUnit
Brustgurt Shirt	X -	X -	X -	X -	- X	- X	- X
EKG	1-Kanal	1-Kanal	1-Kanal	1-Kanal	1-Kanal	1-Kanal	1-Kanal
PPG	-	-	-	-	-	-	-
Atmung	-	-	-	-	-	X	-
Beschleunigung	-	-	-	X	X	X	X
Gyroskop	-	-	-	-	-	-	-
Magnetometer	-	-	-	-	-	-	-
Hautleitwert	-	-	-	-	-	-	-
Temperatur	-	-	-	-	-	-	-
Luftdruck	-	-	-	-	-	-	-
Luftfeuchtigkeit	-	-	-	-	-	-	-
EMG	-	-	-	-	X	-	-
Bioimpedanz	-	-	-	-	-	-	-
Funk	BLE	BLE, 5 kHz	BLE, 5 kHz	BLE, ANT	BLE	Bluetooth	BLE
Speicher	-	-	x	x	-	X	X
Laufzeit	-	-	40 h	16 h	-	n.b.	24 h
Batterie	CR2025	CR2025	CR2025	CR2032	n.b.	LiPo	LiPo
Laufzeit	300 h	200 h	400 h	n.b.	10 h	14 h	24 h
Indikation	-	-	-	LED	LED	n.b.	LED
Rohdaten	-	-	-	-	-	-	-
HR	X	X	X	X	X	X	X
HRV	-	-	-	-	-	X	X
Atemfrequenz	-	-	-	-	-	X	-
Postur	-	-	-	-	-	-	-
Schritte	-	-	-	X	-	-	-
Ganganalyse	-	-	-	-	-	-	-
Aktivität	-	-	-	X	X	-	-
PAT	-	-	-	-	-	-	-
Muskelaktivität	-	-	-	-	X	-	-
Schlafanalyse	-	-	-	-	-	-	-
Max. Beschl.	-	-	-	-	-	-	-
Preis (19.08.2017)	59,95€	69,95€	89,95€	79,99€	338,30€	a.A.	249,90€

Anhang A. Übersicht drahtloser Sensorknoten

Tabelle A.3.: Überblick der Körpersensoren: Multimodale Sensormodule. Mit x markierte Positionen sagen aus, dass der entsprechende Sensor auf dem System zur Verfügung steht.

Hersteller Bezeichnung	Zephyr BioPatch HP	MoviSens EcgMove 3	Shimmer SHIMMER3 ECG Unit	MedIT IPANEMA ECG	FAU ULPSEK	AG KS BG-V4.2
EKG	1-Kanal	1-Kanal	4-Kanal	1-Kanal	1-Kanal	1-Kanal
PPG	-	-	-	-	-	-
Atmung	X	-	-	X	X	X
Beschleunigung	X	X	X	-	X	X
Gyroskop	-	-	X	-	-	-
Magnetometer	-	-	X	-	-	-
Hautleitwert	-	-	-	-	-	-
Temperatur	-	-	-	-	X	X
Luftdruck	-	X	X	-	-	-
Luftfeuchtigkeit	-	-	-	-	-	-
EMG	-	-	-	-	-	-
Bioimpedanz	-	-	-	-	-	-
Funk	BLE 802.15.4	BLE	Bluetooth	CC1101 433 MHz	BLE	2,4 GHz
Speicher	x	x	8 GB SD	n.b.	X	8 MB Flash
Laufzeit	500 h	336 h	n.b.	n.b.	n.b.	12 h
Batterie	LiPo	LiPo	LiPo	LiPo	LiPo	CR2025
Laufzeit	35 h	72 h	n.b.	n.b.	n.b.	12 h
Indikation	LED	LED Vibration	LED	LED	Display	LED
Rohdaten	-	X	X	X	X	X
HR	X	X	X	X	X	X
HRV	X	X	-	-	X	X
Atemfrequenz	X	-	X ¹	X	X	X
Postur	X	-	-	-	-	X
Schritte	X	-	-	-	-	X
Ganganalyse	X	-	-	-	-	-
Aktivität	X	X	-	-	-	X
PAT	-	-	-	-	-	-
Muskelaktivität	-	-	-	-	-	-
Schlafanalyse	-	-	-	-	-	-
Max. Beschl.	X	-	-	-	-	X
Preis (19.08.2017)	449,70€	1700€	448,00€	-	-	-

Tabelle A.4.: Überblick der verfügbaren Umgebungssensoren. Mit * markierte Positionen sind Sensoren, die über Zusatzmodule eingebunden werden können.

Hersteller Bezeichnung	TI	TI	TI	Ruuvi	Espruino	Advanticsys	AG AMI
	CC2560STK	CC1350STK	CC3200STK	RuuviTag	Puck.js	TelosB	BRIX ₂
Temperatur	TMP007	TMP007	TMP007	BME 280	-	SHT11	HIH-6130*
Luftfeuchte	HDC1000	HDC1000	HDC1000	BME 280	-	SHT11	HIH-6130*
Druck	BMP280	BMP280	BMP280	BME 280	-	-	-
Lautstärke	SPH0641LU	SPH0641LU	SPH0641LU	-	-	-	-
Beschl.	MPU-9250	MPU-9250	MPU-9250	LIS2DH12	-	-	MPU-9150 BNO055*
Gyroskop	MPU-9250	MPU-9250	MPU-9250	-	-	-	MPU-9150 BNO055*
Magnetfeld	MPU-9250	MPU-9250	MPU-9250	-	MAG3110	-	MPU-9150 BNO055*
Helligkeit	OPT3001	OPT3001	OPT3001	-	-	S1087	TSL2561*
Abstand	-	-	-	-	-	-	VCNL4010*
Mikrocon.	CC2650: Cortex M3	CC1350: Cortex M3	CC3200: Cortex M4	nRF52832: Cortex M4F	nRF52832: Cortex M4F	MSP430 MSP430	Atmega32U4 Atmega32U4
Funk	CC2650: BLE 6LoWPAN ZigBee	CC1350: BLE Sub1 GHz	CC3200: WLAN	nRF52832: BLE ANT 2,4 GHz	nRF52832: BLE ANT 2,4 GHz	CC2420 802.15.4 ZigBee	A1101, BLE113* 868 MHz BLE*
Speicher	8Mbit Flash	8Mbit Flash	8Mbit Flash	-	-	8Mbit Flash	SD Karte*
Indikation	LED, Buzzer	LED, Buzzer	LED, Buzzer	LED	LED	LED	LED, Buzzer* Audio*, Vibr.*
Batterie	CR2032	CR2032	2xAAA	CR2477	CR2032	2xAAA	Akku (450 mAh)
Preis (19.08.2017)	28€	28€	39€	32€	33€	90€	-