Friedrich-Alexander-Universität
Faculty of Engineering

# Dissertation

Semester 2023

Andreas Tobola

***Energy-Aware Signal Processing for Wearable Biomedical Sensor Systems***

# Energy-Aware Signal Processing for Wearable Biomedical Sensor Systems

# Energieeffiziente Signalverarbeitung für tragbare biomedizinische Sensorsysteme

Der Technischen Fakultät
der Friedrich-Alexander-Universität
Erlangen-Nürnberg
zur
Erlangung des Doktorgrades

Doktor-Ingenieur

vorgelegt von

Andreas Tobola

aus Kattowitz

Als Dissertation genehmigt
von der Technischen Fakultät
der Friedrich-Alexander-Universität Erlangen-Nürnberg

Tag der mündlichen Prüfung: 18. Dezember 2023

Gutachter: Prof. Dr.-Ing. Georg Fischer
Prof. Dr.-Ing. Thomas Ußmüller

# Abstract

Wearable technology has been rapidly changing the overall picture of medical diagnosis. Highly integrated hardware and smart algorithms are enabling long-desired healthcare applications. Nevertheless, engineers still face major challenges. One such bottleneck is the limited energy of sensors worn on the body. Despite highly integrated hardware, the energy limitation requires larger batteries and eventually energy harvesting in addition.

This thesis gives a comprehensive insight into the development of portable sensor technology, especially for biomedical applications, with a particular focus on minimizing energy consumption. One of the central drivers of this work is the philosophy of taking a holistic view of the entire signal processing chain, from the sensor to the evaluation of biomedical information. For this purpose, all parts of a biomedical signal processing chain were examined. The examination included hardware and software components.

Specialized tools and methods were developed for this purpose. A new power profiling method was developed to evaluate the power consumption of microcontroller-based systems. The unique feature of this method is the automated profiling tool which analyzes the system's power consumption for a better understanding of the underlying processes. A further contribution of this thesis is new sensor hardware and new software. This sensor system was particularly designed for low-power evaluation. The sensor system simplifies low-power debugging with modularity and the comfort of dedicated power measurement terminals. Different variants of the sensor system hardware and software were developed and evaluated. The hardware variants investigated in this thesis were various sensor circuits, power supply circuits, and microcontroller types. Examined software variants were real-time operating systems, software-based power managers, and signal processing algorithms. The outstanding feature of the system is its scalability, achieved by designing software and hardware components configurable. The configuration determines the behavior of the sensor system. Also, the configuration influences power consumption. For the Ultra-low-power Sensor Evaluation Kit (ULPSEK), the average power consumption was determined between 50 µW and 10 mW, depending on the configuration.

The measurement results from the test bench, combined with theoretical principles, were merged into a general mathematical energy model to support the engineer in planning a new sensor system. Compared to state of the art, the model combines two standard sleep modes. Besides, the model was proposed in three stages, supporting tailoring for applications beyond this thesis. The energy model was then applied to a theory of information processing efficiency by examining and extending existing theories. In particular, a new metric was introduced to express signal processing efficiency: the Landauer-Decibel. The first advantage of Landauer-Decibel is the reference to a physical limit. The second advantage is the logarithmic representation, predestined to express technological progress over time and compare large efficiency quantities.

# Kurzfassung

Wearables haben das Gesamtbild der medizinischen Diagnostik rasch verändert. Hochintegrierte Hardware und intelligente Algorithmen ermöglichen langersehnte Anwendungen. Allerdings stehen die Ingenieure immer noch vor großen Herausforderungen. Eine dieser Herausforderungen ist die stark begrenzte Energie, die tragbaren Sensoren zur Verfügung steht. Demzufolge können Sensorsysteme nur bedingt miniaturisiert werden, da Batterien mit hoher Kapazität, die Größe des Sensorsystems bestimmen. Gegebenenfalls ist zudem Energy Harvesting zusätzlich erforderlich. Diese Arbeit gibt einen umfassenden Einblick in die Entwicklung der tragbaren Sensortechnologie, insbesondere für biomedizinische Anwendungen, mit besonderem Fokus auf die Minimierung des Energieverbrauchs. Ein zentrales Thema dieser Arbeit ist es, die gesamte Signalverarbeitungskette – vom Sensor bis zur Auswertung biomedizinischer Informationen – zu optimieren. Zu diesem Zweck wurden alle Teile einer biomedizinischen Signalverarbeitungskette untersucht. Die Untersuchung umfasste Hardware- und Softwarekomponenten. Zu diesem Zweck wurden spezielle Werkzeuge und Methoden entwickelt. Eine neue Power-Profiling-Methode wurde entwickelt, um den Stromverbrauch von mikrocontroller-basierten Systemen zu bewerten. Das Besondere an dieser Methode ist ein automatisiertes Profiling-Werkzeug, das den Stromverbrauch des Systems analysiert, um ein besseres Verständnis der zugrunde liegenden Prozesse zu ermöglichen. Ein weiterer Beitrag dieser Arbeit ist eine neue Sensor-Hardware und -Software. Dieses Sensorsystem wurde für eine stromsparende Evaluierung entwickelt. Das Sensorsystem vereinfacht das Low-Power-Debugging durch Modularität und bietet den Komfort von dedizierten Schnittstellen zur Messung der Leistungsaufnahme. Es wurden verschiedene Varianten der Hard- und Software des Sensorsystems entwickelt und miteinander verglichen. Die in dieser Arbeit untersuchten Hardwarevarianten waren Sensorschaltungen, Stromversorgungsschaltungen und Mikrocontrollertypen. Die untersuchten Software-Varianten waren Echtzeitbetriebssysteme, softwarebasierte Power-Manager und Signalverarbeitungsalgorithmen. Das herausragende Merkmal des Systems ist seine Skalierbarkeit, die durch das Hinzufügen mehrerer Konfigurationsmöglichkeiten erreicht wurde. Die Konfiguration bestimmt das Verhalten des Sensorsystems. Außerdem beeinflusst die Konfiguration den Stromverbrauch. Für das Ultra-low-power Sensor Evaluation Kit (ULPSEK) wurde abhängig von der Konfiguration ein mittlerer Stromverbrauch zwischen 50 µW and 10 mW ermittelt. Die Messergebnisse aus dem Prüfstand wurden in Kombination mit theoretischen Grundlagen zu einem allgemeinen mathematischen Energiemodell zusammengeführt, um den Ingenieur bei der Planung eines neuen Sensorsystems zu unterstützen. Die gewonnenen Daten und Modelle wurden anschließend in eine Theorie der energie-effizienten Informationsverarbeitung übergeführt. Dabei wurde eine neue Metrik eingeführt, um die Effizienz der Signalverarbeitung auszudrücken: das Landauer-Decibel.

# Abbreviations

| | |
|---|---|
| **AAMI** | Association for the Advancement of Medical Instrumentation |
| **ADC** | Analog-to-Digital Converter |
| **AHA** | American Heart Association |
| **ARM** | Advanced RISC Machines Ltd. |
| **ASIC** | Application-Specific Integrated Circuit |
| **ASSP** | Application-Specific Standard Part |
| **AWGN** | Additive White Gaussian Noise |
| **BIH** | Boston's Beth Israel Hospital |
| **BLE** | Bluetooth Low Energy |
| **bpm** | Beats Per Minute |
| **CMOS** | Complementary Metal–Oxide–Semiconductor |
| **CMSIS** | Cortex Microcontroller Software Interface Standard |
| **CoAP** | Constrained Application Protocol |
| **CPU** | Central Processing Unit |
| **CSS** | Cascading Style Sheets |
| **CVD** | Cardiovascular Disease |
| **DC** | Direct Current |
| **DIY** | Do It Yourself |
| **DMA** | direct memory access |
| **DRL** | Driven Right Leg |
| **DSP** | Digital Signal Processor |
| **ECG** | Electrocardiograph |
| **ECG** | Electrocardiogram |
| **EEMBC** | Embedded Microprocessor Benchmark Consortium |
| **EFM32** | Energy-Friendly, 32-bit Microcontroller by Silicon Labs Inc. |
| **EMG** | Electromyogram |
| **ESR** | Equivalent Series Resistance |
| **EU** | European Union |
| **FIR** | Finite Impulse Response |
| **FN** | False Negative |
| **FP** | False Positive |
| **FPGA** | Field-Programmable Gate Array |
| **Fraunhofer IIS** | Fraunhofer Institute for Integrated Circuits |
| **GCC** | GNU Compiler Collection |
| **GIF** | Graphics Interchange Format |
| **GPIO** | General Purpose Input/Output |

| | |
|---|---|
| **HMM** | Health Monitoring Manager |
| **HR** | Heart Rate |
| **HRV** | Heart Rate Variability |
| **HTML** | Hypertext Markup Language |
| **$I^2C$** | Inter-Integrated Circuit Bus |
| **IAR** | Ingenjörsfirman Anders Rundgren |
| **IC** | Integrated Circuit |
| **ICU** | Intensive Care Unit |
| **IEE** | Informational Energy Efficiency |
| **IEEE** | Institute of Electrical and Electronics Engineers |
| **IMEC** | Interuniversity Microelectronics Centre |
| **IoT** | Internet of Things |
| **LA** | Left Arm |
| **LED** | Light-emitting Diode |
| **LL** | Left Leg |
| **LoRaWAN** | Long Range Wide Area Network |
| **LPSAN** | Low-Power Short-Area Network |
| **LPWAN** | Low-Power Wide-Area Network |
| **LZW** | Lampel-Ziv-Welch |
| **MEMS** | Micro-Electro-Mechanical Systems |
| **METEAN** | Medizintechnisches Test- und Anwendungszentrum |
| **MIT** | Massachusetts Institute of Technology |
| **MOSFET** | Metal–Oxide–Semiconductor Field-Effect Transistor |
| **NN** | Normal to Normal |
| **OSI** | Open Systems Interconnection |
| **PC** | Personal Computer |
| **PCB** | Printed Circuit Board |
| **PEEK** | Proactive Energy-aware Evaluation Kit |
| **PLL** | Phase Locked Loop |
| **PPG** | Photoplethysmograph |
| **PPG** | Photoplethysmography |
| **PPV** | Positive Predictive Value |
| **PRS** | Peripheral Reflex Systems |
| **PSRR** | Power Supply Rejection Ratio |
| **PTT** | Pulse Transit Time |
| **Qi** | Open Interface standard that defines wireless power transfer from the Chinese word qi |
| **QRS** | Q, R, and S markers within an Electrocardiogram (ECG) |
| **RA** | Right Arm |
| **RAM** | Random Access Memory |
| **RC** | resistor-capacitor |
| **RIP** | Respiratory Inductance Plethysmography |
| **RLD** | Right Leg Drive |
| **RMS** | Root Mean Square |
| **RMSSD** | Root Mean Square of Successive Differences |
| **RSA** | Respiratory Sinus Arrhythmia |

| | |
|---|---|
| **RTOS** | Real-Time Operating System |
| **SIMD** | Single Instruction, Multiple Data |
| **SNR** | Signal-to-Noise Ratio |
| **SoC** | System on a Chip |
| **SPI** | Serial Peripheral Interface |
| **$SpO_2$** | Oxygen Saturation |
| **TFT** | Thin-Film Transistor |
| **TP** | True Positive |
| **TUM** | Technical University of Munich |
| **UI** | User Interface |
| **ULPSEK** | Ultra-low-power Sensor Evaluation Kit |
| **UML** | Unified Modeling Language |
| **UML2** | Unified Modeling Language in version 2 |
| **USB** | Universal Serial Bus |
| **WHO** | World Health Organization |

# List of Symbols

| | |
|---|---|
| $E_{STAR}$ | Energy required for the cycles Start-up, Acquisition, Transition, and Release. |
| $E_{STR}$ | Energy required for the cycles Start-up, Transition, and Release |
| $E_p$ | Energy required for processing a computing task. |
| $E_{hib,er}$ | Energy required to enter hibernation and return from hibernation. |
| $E_L$ | The absolute minimum for computing is known as Landauer's Principle or Landauer's Bound. |
| $P_{sys,M1}$ | Average power dissipation of the whole sensor system expressed by model 1, which is an always-on system.. |
| $P_{sys,M2}$ | Average power dissipation of the whole sensor system expressed by model 2, which is a system with automatic sleep on idle feature controlled by the operating system. |
| $P_{sys,M3}$ | Average power dissipation for a system with model M3 utilizing hibernation mode. |
| $\tilde{P}_{sys,M2}$ | Approximation for average power dissipation with model 2 without taking transitions between system states into account. |
| $\tilde{P}_{sys,M3}$ | Approximation for average power dissipation with model 3 without taking transitions between system states into account. |
| $P_{hib}$ | Static power dissipation in hibernation mode. |
| $P_{core,dyn}$ | Dynamic power dissipation of the microcontroller core. |
| $P_{peri,dyn}$ | Dynamic power dissipation of the microcontroller core. |
| $P_{stat}$ | Static power dissipation of the microcontroller core. |
| $P_{lp,dyn}$ | Dynamic power dissipation of the microcontroller core. |
| $P_{lp,stat}$ | Static power dissipation of the microcontroller core. |
| $P_{PDN}$ | Power consumption by the power distribution network (e.g. losses incurred by voltage regulators) |
| $P_{frontends}$ | Power consumption required for all frontends, e.g. ECGs amplifier. |
| $P_{UI}$ | Power consumption required for the user interface, e.g. display or Light-emitting Diodes (LEDs). |
| $P_{sensors}$ | Power consumption required for the sensors, e.g. the probe of a pulse oximeter. |

| | |
|---|---|
| $P_{radio}$ | Power consumption that is required for wireless transmission, e.g. the Bluetooth module. Therefore, an external model is required, likewise evaluated in section 6.2 . |
| $V_{dd}$ | Supply voltage for a digital circuit. |
| $f_c$ | Clock of a digital system. |
| $f_{core,M1}$ | Microcontroller core clock for the model type 1, which is an always-on system. |
| $f_{core,M2}$ | Microcontroller core clock for the model type 2, which is a system with automatic sleep on idle feature controlled by the operating system. |
| $f_{peri}$ | Microcontroller peripheral clock. |
| $f_{lp}$ | Microcontroller clock of the low-power domain of a system with automatic sleep on idle feature controlled by the operating system. |
| $r_c$ | Cycle rate or number of executions per second. |
| $r_h$ | Rate of the hibernation cycles. |
| $t_{STAR}$ | Time required for the for cycles Start-up, Acquisition, Transition, and Release. |
| $t_{hib,er}$ | Duration while entering hibernation and returning from hibernation. |
| $t_{hib}$ | Duration of staying in hibernation. |
| $t_{sens}$ | Duration while sensing and data processing is performed, includes the pre-sening. |
| $t_{sens,pre}$ | Duration of the pre-sensing after hibernation. Pre-sensing is required to fill buffers, initialize filters, and to wait for valid outputs. Presenting does not deliver output. |
| $t_{sens,eff}$ | Duration of the effective sensing while output is generated. |
| $t_h$ | Average time interval for the hibernation cycle. |
| $t_p$ | Average time estimation for processing a task. |
| $p_{act}$ | Probability for a system to be in the active state. |
| $p_{hib}$ | Probability for a system to be in the hibernation state. |
| $C$ | Shannon's Channel Capacity describes the maximum bit rate for a given bandwidth and a given Signal-to-Noise Ratio (SNR) of a transmission channel. |
| $B$ | Bandwidth of a communication channel or a signal processing unit. |
| $S$ | Signal power. |
| $N$ | Noise power. |
| $r_{service}$ | Service Rate. |
| $r_t$ | Throughput data rate. |
| $r_{out}$ | Total bitrate of all outputs of a signal processing block under investigation. |
| $r_{in}$ | Total bitrate of all inputs of a signal processing block under investigation. |
| $P_{sui}$ | Total power consumption of a signal processing block under investigation. |
| $\Gamma_{IEE}$ | Informational Energy Efficiency |
| $\Gamma_{dBLa}$ | Informational Energy Efficiency expressed in Decibel and normalized to Landauerś bound. |
| $\gamma_{decay}$ | Overhead Factor Decay is a measure for system design efficiency. |
| $\gamma_{hib}$ | Hibernation efficiency factor. |
| $k_B$ | Boltzmann constant. |

| | |
|---|---|
| $T$ | Temperature |
| $N_{cy}$ | An estimation for the average number of cycles for repetitive computing processes. |
| $C_L$ | Load capacitance in a digital system refers to the total capacitance that the output of a digital circuit must drive. |

# Author's Statement

This thesis is based on scientific work accomplished under the supervision of Prof. Dr.-Ing. Georg Fischer of the Institute for Electronics Engineering (LTE) of Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU). Further supervisors during the research work for this thesis were Prof. Dr. Björn Eskofier, Head of the Machine Learning and Data Analytics Lab (MaD Lab), Department of Artificial Intelligence in Biomedical Engineering (AIBE), Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU). The experiments and research were mainly executed at the Department for Image Processing and Medical Engineering at the Fraunhofer Institute for Integrated Circuits (IIS) in Erlangen, with Christian Weigand as Department Head and Christian Hofmann as Group Head. Most chapters of this thesis, including 1, 2, 3, 4, 7, 8, and 9, are written in an entirely monographic character. The experimental parts of this thesis were frequently reused as drafts for paper publications, then updated after paper publication, and further developed or reworked after publication. In particular, the experimental chapters 5 and 6 are correlated with the following author's research papers:

- Journal Paper: **Tobola, A.**, Leutheuser, H., Pollak, M., Spies, P., Hofmann, C., Weigand, C., Eskofier, B. M., Fischer, G., "Self-Powered Multiparameter Health Sensor," IEEE Journal of Biomedical and Health Informatics, vol. 22 (1), pp. 15–22, Jan. 2018.
  The author's contributions to this work encompassed a range of integral roles and tasks, including the conceptualization of the research goals and objectives, development of the overarching methodology for the project, including the demonstration of the sensor system powered by an energy harvester and the systematic collection of data, designing the experiments and investigation, formal analysis, writing additional software for ULPSEK, integration of energy harvesting hardware into the ULPSEK platform, complemented by supplementary hardware components for compensation of probe effects, validation of results, writing the original draft, writing essential parts of the corresponding publication, editing, and correspondence during the review process.

- Conference Paper: Kindt, P., Yunge, D., **Tobola, A.**, Fischer, G., Chakraborty, S., "Dynamic service switching for the medical IoT," IEEE 27th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC), pp. 1–7, 2016.
  The author's contribution was the conceptualization of the research goals together with the members of the Chair of Real-Time Computer Systems at the Technical University of Munich (TUM), developing the methodology for combining the model for wireless transmission developed by Kindt together with the model developed

for this thesis, writing the related parts about low-power ECG of the publication regarding the power consumption of a configurable biomedical sensor system, design and development of methodology for the energy model with exemplary data for this publication.

- Conference Paper: **Tobola, A.**, Leutheuser, H., Schmitz, B., Hofmann, C., Struck, M., Weigand, C., Eskofier, B. M., Fischer, G., "Battery Runtime Optimization Toolbox for Wearable Biomedical Sensors," Proc. IEEE 13th International Conference on Wearable and Implantable Body Sensor Networks (BSN), pp. 199–204, Jun. 2016.
  The author's contributions were the conceptualization of the research goals and objectives, the development of methodology for the energy model and the collection of data, the development of methodology for collection of data for exemplary values of model parameters, designing the experiments and investigation, design and formal analysis, writing software for ULPSEK, supervising students supporting the experimental execution, validation of results, writing the original draft, writing essential parts of the corresponding publication, project administration, correspondence during the review process, visualization of the results with a poster for the conference in San Francisco, USA.

- Conference Paper: **Tobola, A.**, Espig, C., Streit, F. J., Korpok, O., et al., "Scalable ECG Hardware and Algorithms for Extended Runtime of Wearable Sensors," Proc. IEEE International Symposium on Medical Measurements and Applications (MeMeA), 2015.
  The author's contributions were the conceptualization of the research goals and objectives, development of methodology for a novel system design allowing adaptation to a necessary energy level, development of methodology for collecting data, designing the experiments and investigation, formal analysis and verification of the results, optimizing the measurement procedure, software, and hardware iteratively according to the gained results, writing software for ULPSEK, supervising three students supporting the experimental execution, conducting research, validation of results, writing the original draft, writing essential parts of the corresponding publication, project administration, correspondence during the review process, visualization of the results, preparing and giving a talk for the conference in Turin, Italy.

- Conference Paper: **Tobola, A.**, Korpok, O., Leutheuser, H., Schmitz, B., Hofmann, C., Struck, M., Weigand, C., Eskofier, B., Heuberger, A., Fischer, G., "System Design Impacts on Battery Runtime of Wearable Medical Sensors," Proc. 2nd International Conference on Mobile and Information Technologies in Medicine (MobileMed), 2014.
  The author's contributions were the conceptualization of the research goals and objectives, in particular, the conceptualization of the objectives for investigating the impact regarding operating system, the impact regarding clock unit settings, the impact regarding microcontroller core, the design and development of methodology for high-resolution energy monitoring, the design and development of methodology for automation and data collection of energy monitoring data, designing the experiments and investigation, formal analysis, evaluating the results while developing and optimizing the profiling method iteratively, writing software for ULPSEK, conceptualization and writing of the automated power profiler software in MATLAB,

supervising three students supporting the experimental execution, conducting research, validation of results, writing the original draft, writing essential parts of the corresponding publication, project administration, correspondence during the review process, visualization of the results, and preparing and giving a talk at the conference in Prague, Czech Republic.

The following author's publications were additionally published within the context of this thesis topic without overlap in this thesis:

- Technical Report: **Tobola, A.**, Hofmann, C., "Schlussbericht HE2mT - High-Level Entwurfsmethoden energieoptimierter, mobiler Telemonitoringsysteme, Schlussbericht für das Forschungsvorhaben HE2mT - Teilbericht Fraunhofer IIS, Final report on public funded research project HE2mT," Tech. Rep., Frauhofer IIS, 2017.
  The author's contribution was conceptualization and methodology for the corresponding report, which was the final result presentation for one of the projects that financed the research in this thesis.

- Book Chapter: Leutheuser, H., Lang, N., Gradl, S., Struck, M., **Tobola, A.**, Hofmann, C., Anneken, Eskofier, B., "Book Chapter: Textile Wearable Technologies for Sports and Medical Applications," in Smart Textiles: Fundamentals, Design, and Interaction. Springer, 2017.
  The author's contribution was conceptualizing and writing one section of the corresponding book chapter and reviewing the book chapter.

- Technical Report: **Tobola, A.**, "Intelligente Sensorik : Schlussbericht, BMBF-Spitzencluster Medical Valley" Tech. Rep., 2015.
  The author's contribution was conceptualization and methodology for the corresponding report, including the description of the apnea algorithm, which was the final result presentation for one of the projects that financed the research in this thesis.

- Conference Paper: Schmitz, B., Hofmann, C., Maestre, R., Bleda, A. L., **Tobola, A.**, Melcher, V., Gent, J., van, "Continuous vital monitoring and automated alert message generation for motorbike riders," Computing in Cardiology Conference (CinC), IEEE, pp. 397–400, 2015.
  The author's contribution was writing parts of the publication, designing a low-power feature for the respiratory system, and supporting the primary author in the conceptualization of the research goals and objectives for his first publication.

- Conference Paper: **Tobola, A.**, Streit, F. J., Korpok, O., Espig, C., et al., "Sampling Rate Impact on Energy Consumption of Biomedical Signal Processing Systems," Proc. IEEE 12th International Conference on Wearable and Implantable Body Sensor Networks (BSN), 2015.
  This publication is entirely theoretical, giving quantitative insight into the interest in keeping the sampling rate as low as possible regarding power consumption.
  The author's contributions were the conceptualization of the research goals and objectives, design and development of the methodology for combining the O-notation with the data rate, deriving all mathematical terms about the impact of the sampling rate on power consumption, writing the original draft, writing of the corresponding

publication, project administration, correspondence during the review process, visualization of the results, preparing and presenting a poster for the conference in Boston, USA.

- Conference Publication: **Tobola, A.**, Rybalko, R., Korpok, O., Streit, F. J., Espig, C., Hofmann, C., Lang, N., Struck, M., Weigand, C., Fischer, G., "ULPSEK – Ultra-Low-Power Sensor Evaluation Kit," Proc. Biomedizinische Technik (BMT), 2015.
  The author's contributions were the conceptualization of a modular development board, methodology for gaining data with the development board with emphasis on low-power optimization, designing experiments, investigation for comparable systems, specifying the entire hardware, writing software for ULPSEK, validation of results, writing the original draft, writing essential parts of the corresponding publication, correspondence during the review process, visualization of the results, preparing and giving a talk about the challenges and solutions for low-power wearable biomedical sensor systems at the conference Biomedizinische Technik (BMT) 2015 in the FOCUS Session "Wearable Sensors" in Aachen.

Besides, three intellectual property applications were published within the context of this thesis topic.

- Patent filed: Digital Twin for Sensor System Energy Optimization and Supervision, Alternative Title: Controlling a Sensor System, (EP Application No. 22157742.2, **Tobola**, 2022)
  This invention disclosure describes a system with essential parts: (1) Sensor system extended by a self-monitoring unit for power consumption profiling chapter 4 with the technical solution presented in section 3.7, (2) a model for the power consumption of the sensor system as described in chapter 7, (3) a real-time communication between a real asset (sensor system) and the energy model to update model parameters and to identify anomalies. This solution aims to detect anomalies in power consumption which can be safety or security related. The author contributed 100% to this patent's conceptualization and methodology.

- Patent filed: Data-quality-aware Sensor Communication, (Scholl, **Tobola**, Ludwig, 2021)
  This invention disclosure describes a procedure to transmit data based on the sensor data quality to reduce power consumption to a necessary level. The author contributed 30% to this patent's conceptualization and methodology.

- Patent published EP4063795A1: Sensorsystem zur Messung der Variabilität der elektrischen Spannung einer Energieversorgung (**Tobola**, 2021)
  This patent describes a system for the quality monitoring of the sensor's supply voltage. In particular, a method was proposed to release the processing load from the microcontroller by preprocessing the quality metric with a proposed circuit. This circuit was proposed in a draft version and has been overworked meanwhile. However, the goal could be achieved by introducing a preprocessing stage to provide a metric, reducing the overall system energy consumption. The author contributed 100% to this patent's conceptualization and methodology.

For the publications with first authorship, the author of this thesis engaged in conceptualization, formulated overarching research goals and aims, provided the methodology for the experiments, and conducted the formal analysis to evaluate the results. This research was carried out with advisory support from Prof. Dr. Georg Fischer and Prof. Dr. Björn Eskofier. The roles of the co-authors grouped by their affiliation at the time of publication are declared below.

Members of the Fraunhofer IIS in Erlangen were contributing as follows: Korpok, Streit, Espig, and Schmitz contributed during their Bachelor's Thesis or Master's Thesis for this Doctoral Thesis by executing experiments, reworking hardware, writing parts of the software, and reviewing the wording of the conference papers. Rybalko was a hardware engineer contributing to the layout of printed circuit boards. Sauter supported one paper with valuable discussions from the perspective of computer science. Pollak and Spies contributed their hardware for the experiments with the energy harvester. Hofmann and Struck were group heads at the Fraunhofer IIS in Erlangen. Lang had a coordinating function in the Leistungszentrum Elektroniksysteme (LZE), Erlangen. Heuberger has been the head of the Fraunhofer IIS in Erlangen.

Members from the Chair of Real-Time Computer Systems at the Technical University of Munich (TUM) contributed as follows: Kindt contributed energy estimations for wireless transmission and was the main author of the publication "Dynamic Service Switching for the Medical IoT". Yunge contributed energy estimations for code interpreters. Chakraborty was the chair head of Real-Time Computer Systems.

Members from the Machine Learning and Data Analytics Lab (MaD Lab), Department of Artificial Intelligence in Biomedical Engineering (AIBE), Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU) contributed as follows: Eskofier was the supervising professor with feedback on the publications. Leutheuser was a valuable discussion partner for reviewing the above-stated publications. Further, Leutheuser was the corresponding author for the above-stated book chapter. Gradl contributed individual sections to the above-listed book chapter.

Anneken has been a member of the Department of Cardiology, University Hospital Erlangen, Germany, with individual sections to the above-listed book chapter.

Members from Siemens AG, in Erlangen contributed as follows: Ludwig has contributed to one patent as a project manager and senior engineer at Siemens AG. Scholl was a Ph.D. student at Siemens AG under the author's supervision in another research field with the idea of one patent listed above.

# Preface

In my exploration of portable sensor technology for biomedical applications, I took a holistic perspective on signal processing. A core objective was to minimize energy consumption, enabling longer battery-powered health monitoring and fostering possibilities for breakthroughs in disease healing. By developing scalable and energy-efficient solutions, I envision a future where wearable technology empowers longer, uninterrupted monitoring, opening new horizons for healthcare advancements.

Throughout this thesis, I have predominantly experienced positive work experiences and a sense of energy driving me forward. As I reach its conclusion, I am reminded of a fitting citation from Mihaly Csikszentmihalyi's book ”The Psychology of Optimal Experience,” published in 1990: “The best moments in our lives are not the passive, receptive, relaxing times. The best moments usually occur if a person's body or mind is stretched to its limits in a voluntary effort to accomplish something difficult and worthwhile.”

# Acknowledgment

I am deeply grateful to all those who have supported and contributed to the completion of this thesis. This work would not have been possible without their encouragement, guidance, and assistance.

First and foremost, I extend my heartfelt gratitude to my supervisor Prof. Dr.-Ing. Georg Fischer of the Institute for Electronics Engineering (LTE) of Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU), whose invaluable expertise, inspiring discussions, and unwavering support steered me through the entire research process. His insightful feedback and constructive criticism have significantly improved the quality of this thesis. Prof. Dr.-Ing. Björn Eskofier, Head of the Machine Learning and Data Analytics Lab (MaD Lab), Department of Artificial Intelligence in Biomedical Engineering (AIBE), Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU) has generously shared his knowledge and experience regarding publications, offering valuable suggestions to improve the clarity and impact of the findings. His constructive criticism and attention to detail have been pivotal in refining the published manuscripts. I am profoundly grateful to Prof. Dr.-Ing. Ußmüller, Head of the Microelectronics and Implantable Systems Group, Department of Mechatronics, Universität Innsbruck, for undertaking the review and dedicating thorough attention and expertise in evaluating this Ph.D. thesis. I feel incredibly privileged to have had the opportunity to be guided by such exceptional supervisors, and their guidance was always at the forefront of my awareness.

I am grateful for the resources and facilities provided by the Friedrich-Alexander-University Erlangen-Nürnberg (FAU), in general. In particular, I want to express my heartfelt thanks to Shira and Deirdre for helping me gain speed in writing and answering my specific questions. The FAU offered support in academic writing which was an essential brick on the path toward the final goal. Thank you both for the invaluable support in improving my academic English writing skills. Your guidance and feedback have been tremendously helpful, and I am truly grateful for your assistance.

I would like to express my sincere appreciation to the staff of the Fraunhofer Institute for Integrated Circuits (IIS), where I have executed the main part of my research, in particular for providing an environment and the necessary resources to conduct this research. Thanks to the group and department heads Christian H. and Christian W., who paved the way and made this work possible. Furthermore, I would like to acknowledge the support of all my former colleagues and peers at Fraunhofer IIS for their camaraderie and stimulating discussions, which have broadened my perspective on the subject matter. Their commitment to excellence has been a source of inspiration throughout my journey.

I want to express my sincere thanks to the numerous Bachelor's and Master's Theses Students, as well as student assistants under my supervision: Majdi, Oliver, Chris, Franz, Julia, Rainer, Björn, and Michael. Thank you all for your passion and endurance while conducting the research under my supervision. Your theses have added value to this research, and I have enjoyed spending time with you.

I extend my heartfelt thanks to the group head for Sensor System Integration at Siemens AG, Dr. Uwe Linnert, and the project leaders for the flexibility at the end run and for many of the team who supported me. I am grateful to be part of such an inspiring and powerful team.

My heartfelt thanks go to my family and friends for their unyielding encouragement, patience, and understanding. Their constant belief in my abilities has been the driving force behind my perseverance. In particular, I want to thank Carsten, Uli, and Thomas for the feedback during the final editing.

I give special thanks to Tobias for teaching me the GTD (Getting Things Done) method. Your guidance and insights have been truly transformative, and I am immensely thankful for introducing me to such a valuable productivity system.

I would like to extend my deepest gratitude to my mother for her unwavering support and dedication in taking care of the children during the course of this research. Her selflessness and love have been a constant source of inspiration and motivation for me. Without her tireless efforts and sacrifices, it would have been immensely challenging to focus on my academic pursuits. Her presence and care have provided me with peace of mind and the ability to devote myself to this thesis. I am truly blessed to have such an incredible and caring mother, and I am forever grateful for all that she has done.

With the same warmth and gratitude, I want to extend my heartfelt thanks to Lina and Robert for their exceptional care of our children. Your dedication and love have made a significant impact, and I genuinely appreciate all you do. Your support means the world to us, and we are blessed to have you as a part of our family.

I want to express my particular honor to my father, who profoundly impacted my life, inspiring my curiosity and igniting the fire for learning. The memories and lessons he shared with me will undoubtedly continue to shape my journey, and he will forever hold a special place in my heart and mind.

Dear Heike, I want to express my heartfelt thanks for being an incredible mother to our children and for being an excellent scientist with whom I have enjoyed exchanging ideas about research practice. Your unwavering support and dedication have been a source of strength for our family, and I am grateful for the love you provide to our children and me.

I would like to express my sincere appreciation to all my former project partners, who were many interesting individuals. Working with each of you during this thesis was an absolute pleasure, and I truly enjoyed the collaborative experience we shared. Together, we accomplished our goals, and I couldn't be more thankful for the dedication and effort each one of you put into the funded projects. Your contributions made a significant impact, and I'm proud of what we achieved as a team. Thank you all for the wonderful journey we had together! Two public funding projects supported this thesis: project "HE2mT - High-Level Entwurfsmethoden energieoptimierter, mobiler Telemonitoringsysteme", with the funding reference number BMBF 16M3203E, compound number 01133267, and the project "Intelligente Sensorik: Smart Sensors A" with the funding reference number BMBF 13EX1014D, compound number 01079021.

For all of you who believed in me, I sincerely appreciate your unwavering support throughout my scientific journey.

# Contents

# Chapter 1

# Introduction

## 1.1. Motivation

17.9 million people died from Cardiovascular Diseases (CVDs) in 2016 according to the World Health Organization (WHO) [45], representing 31% of all deaths globally. The European Health Network reported in the *European Cardiovascular Disease Statistics 2017* [56] that CVD is currently responsible for 3.9 million deaths per year in Europe and 1.8 million deaths in the European Union (EU). This is 45% of all deaths in Europe and 37% of all deaths in the EU. Further, almost 49 million people live with cardiovascular disease in the EU. Guidelines for early detection of CVDs were published by the European Heart Network [26] in 2019. They declare: The key is early diagnosis. For this, patients must learn to recognize the symptoms of heart failure. Also, medical personnel needs to be better trained to recognize non-specific symptoms. Further, on any suspicion of heart failure, patients should visit a physician for diagnosis with the support of medical instrumentation. Learning to recognize non-specific symptoms is as challenging as it sounds, and measurement by medical instrumentation still seems to be a rare commodity. Wearable biomedical sensor systems promise to fill this gap. Wearables are already available, and the market is growing.

The major advantage of wearable technology is the higher number of measurements compared to a rare visit to the physician. In general, measurements of biomedical parameters and especially their interpretation are rare. There are no incentives for frequent visits to the doctor either. The opposite is the case. The German health system does not cover frequent visits to a specialist. Additional visits are at the physician's expense. Wearables are one attempt to get the costs of an increasingly expensive healthcare system under control.

However, on the way toward novel wearable sensors, limited battery runtime is still one of the barriers for many long-desired applications. Some wearable products reach a few weeks of battery runtime, but most run at best in less than a day. Battery runtime and the size of the battery remain major concerns.

Therefore, research has been done in key areas for optimizing the battery runtime, such as efficient wireless connectivity technology, novel energy storage technology, and energy harvesting. Since the energy for wireless connectivity was identified as a significant issue, less attention has been paid to signal processing. On the contrary, Mammela and Anttonen [51] countered that the energy for signal processing would become the dominant consumer on a wireless sensor node due to physical limits.

The aim of this work is a holistic view of the signal processing chain of a biomedical wearable sensor system, intending to optimize energy consumption for the most extended possible battery life.

## 1.2. Contributions

This thesis contributes to methods for minimizing the power consumption required for signal processing in wearable biomedical sensor systems.

The development plan of this work is illustrated in Fig. 1.1 in four layers from the bottom to the top. The bottom is the broad field of fundamentals with a literature review and standard techniques regarding low-power signal processing. The second layer focuses on concrete tools for the experiments, particularly the new sensor system and the new power profiling method. The third layer represents the experimental stage with empirical studies. Finally, the fourth layer is dedicated to generalization based on the empirical observations by creating an energy model for the sensor system. The thesis closes up with a discussion about the best practices and the most effective decisions when designing a new wearable biomedical sensor system.

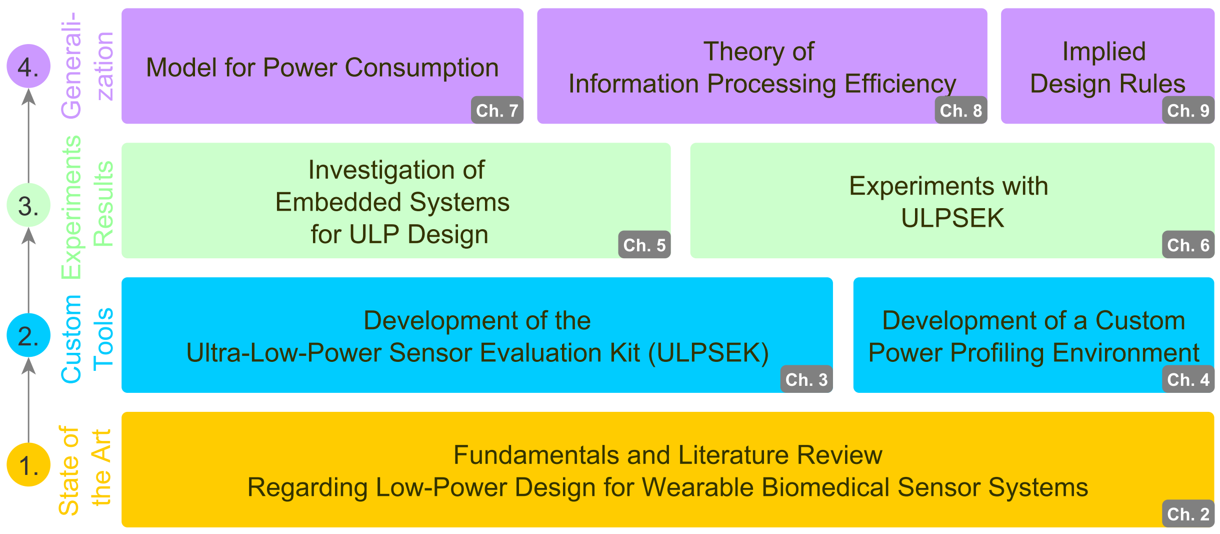

4.
Generali-zation
Model for Power Consumption
Ch. 7
Theory of Information Processing Efficiency
Ch. 8
Implied Design Rules
Ch. 9
3.
Experiments Results
Investigation of Embedded Systems for ULP Design
Ch. 5
Experiments with ULPSEK
Ch. 6
2.
Custom Tools
Development of the Ultra-Low-Power Sensor Evaluation Kit (ULPSEK)
Ch. 3
Development of a Custom Power Profiling Environment
Ch. 4
1.
State of the Art
Fundamentals and Literature Review Regarding Low-Power Design for Wearable Biomedical Sensor Systems
Ch. 2

**Figure 1.1.:** This figure illustrates the building blocks of this thesis. Each building block refers to a chapter in this thesis. The building blocks are ordered in four levels: from the bottom to the top level: The goal at the top of this thesis is the power consumption model.

### Contribution to the low-power sensor system hardware and software

ULPSEK is a sensor system with selected hardware and dedicated software to meet low-power requirements. The hardware and software of this platform are fully accessible, allowing insights into the possibilities of optimizing low-power sensor systems. Therefore, this platform has a central importance for this work, and its iterative development is one of the main contributions of this thesis. Further details can be found in chapter 3.

## Contribution to low-power profiling methods

The development of a custom profiling environment for power consumption measurements, including reference measurement techniques and automated profiling software, was developed for extracting numerous indicators from the measurements gained in this work (moreover in chapter 4). This tool was essential for this work by enabling further experiments to measure and evaluate the power consumption in dependency of several configurations of the system's hardware and software components.

## Contributions to energy consumption benchmarks

Benchmark results are essential when making design decisions. Furthermore, benchmark results were used to develop a model and, in particular, to fill it with concrete values derived from actual measurements. Therefore, in this thesis, benchmarks have been done for single parts of a sensor systems hardware and software and for the sensor system as a whole.

The findings influenced the product development the author was involved in parallel with this thesis. After all, the methods have changed the process of how software for microcontrollers has to be written and how components need to be selected for low-power applications.

The gained results through benchmarking influenced other teams, too. As a consequence of the results published in [100] the whole embedded developer team migrated to a new microcontroller family, including the recommended toolchain. This decision was already thought of but not expected to be in the quantity of 80 % wasted battery energy compared to the previously used microcontroller. The same decision was made by other research groups after reviewing the results.

Further, example applications for configurable systems and energy harvesting were set up using the ULPSEK, and the power consumption was measured, allowing detailed insight into hidden sources of unnecessary power loss. Details can be found in chapter 6.

## Contribution to energy-consumption modeling

Energy consumption modeling is an ongoing process that requires updates continuously in a fast-changing world. The model contributed in this thesis allows the prediction of the power consumption depending on various sensor system configurations giving other engineers a quantifiable effect on the overall power consumption depending on their decisions on hardware and software. The mathematical description of the model is given in chapter 7. For better convenience, a subset of the model was implemented as an interactive website at http://ulpsek.com.

## 1.3. Outline

This thesis is structured as follows:

**Chapter 2: Fundamentals** (p. 7)
This chapter gives an overview of biomedical wearable sensor systems and the challenges of designing them for long battery life.

**Chapter 3: Developing the Ultra-Low-Power Sensor Evaluation Kit (ULPSEK)** (p. 45)
ULPSEK is a sensor system designed to address the research questions in this work. The hardware and software of this platform are fully accessible, allowing insights into the possibilities of optimizing low-power sensor systems. Therefore, this platform has central importance for this work, and its iterative development is one of the main contributions of this thesis. This chapter describes the purpose, features and all components of ULPSEK.

**Chapter 4: Power Profiling Environment for Low-Power Circuits and Algorithms** (p. 75)
Since energy consumption measurements were essential for optimizing low-power systems, a measurement procedure was developed especially for this purpose. The outstanding features of this measurement procedure are the high time resolution and bandwidth of the recordings.

**Chapter 5: Experiments with Embedded Systems** (p. 89)
In this chapter, experiments were performed, which were essential for selecting the microcontroller for ULPSEK.

**Chapter 6: Experiments with ULPSEK** (p. 105)
The chapter describes experiments conducted with ULPSEK to gain insight into the energy model and determine model parameters.

**Chapter 7: Generalized Model for Power Consumption** (p. 121)
The content of this chapter is the formal description of an energy model for the biomedical signal processing of the wearable sensor system regarding the used hardware and software. The energy model was derived based on the theoretical and empirical work in the previous chapters.

**Chapter 8: Theory of Energy-efficient Information Processing** (p. 135)
This chapter utilizes methods from the research field of Information Theory for optimizing power consumption. The contributions to the state of the art are proposals for methodical extensions, in particular, the introduction of a new metric for signal processing efficiency: the Landauer-Decibel.

**Chapter 9: Implied Design Rules** (p. 149)
In this chapter, the most remarkable findings were mapped into a short list of design rules.

# Chapter 2

# Fundamentals

This chapter overviews biomedical wearable sensor systems and the challenges of designing them for extended battery life. The fundamentals begin with biomedical signal analysis in general, followed by the challenges and key areas, specifically in optimization for wearable biomedical sensor systems. After that, the fundamentals focus on wearable signal processing, which is the power optimization scope of this work.

## 2.1. Wearable Biomedical Sensor Systems

This section dedicates to Wearable Biomedical Sensor Systems and their essential parts. The term Wearable Biomedical Sensor Systems was defined as follows: A Sensor translates a property of the physical world into a representation, making it technically beneficial for further signal processing. A Sensor System contains at least one sensor and at least rudimentary signal processing e.g. amplification and band limitation. However, also Sensor Systems with advanced signal processing exist. A Biomedical Sensor System describes a subset of sensors with a particular focus on biomedical sensing and signal processing, e.g. clinical monitor. Finally, Wearable Biomedical Sensor Systems represent on-body accessories with the aim of biomedical sensing, processing, and monitoring. The term sensor is often used as a synonym or abbreviation for a Sensor System, such as by Gandhi and Wang [89]. However, Gandhi and Wang discussed the definition using the term Biosensing Wearables. The term used in this thesis is Wearable Biomedical Sensor Systems.

### 2.1.1. Exemplary Wearable Biomedical Sensor Systems

Wearables differ in body placement, sensor type, the type of information they provide, the quality of the information they provide, and of particular interest in this thesis, their battery runtime. In table 2.1 selected examples for typical wearable biosensors were given together with their battery runtime-related information. After the first column, which contains the name of the sensor system, a category was given. Further, the following two columns contain the used sensors and the parameters calculated from raw sensor data, e.g. using an ECG sensor to calculate Heart Rate (HR) as a parameter. Moreover, as things become connected, the wireless standard, which determines the boundaries of data rate and power consumption, was added to the next column. The next three columns contain the power consumption, the battery size, and the battery runtime. If available in the table, the data was mostly taken from the sources given in column one. In some cases, the missing data were calculated from other data in the product sheets or publications. Note that data about power consumption and battery runtime is rare in product sheets. Despite this, it is even more difficult to get data that is reliable in a typical usage scenario.

The first row in the table is the heart chest belt Polar H7. This chest belt uses two built-in electrodes, which are used to sense an ECG signal. This raw ECG signal is then used with onboard signal processing for computing the HR. The HR is then transmitted wirelessly via Bluetooth Low Energy. Chest belts, in general, were designed for long battery runtime and robustness regarding motion artifacts. The Polar H7 is commercially available and claims a battery runtime of 200 h in the user's manual [67]. In a lab scenario with a close distance to a smartphone that was receiving the data, a current was measured at 510 µA. Besides this, 430 µA were measured during advertising when the smartphone was not connected. Another and more advanced chest belt is the Zephyr Bioharness which was added to the table in two release versions in the next two rows of the table. This harness extends the ECG by a respiratory sensor, an accelerometer, and a body surface temperature sensor. The total of four heterogeneous sensors enables a variety of interesting parameters which can be potentially gained. The additional functionality comes with an additional need for energy which is approximately 200 times as the Polar H7. This has limited the Zephyr Bioharness in release version 1 from 6 h to 18 h of operation. The battery runtime was improved with the Zephyr Bioharness 3, ranging from 12 h to 24 h. Another chest belt with similar parameters is the QuadriCore and similar battery runtime. In contrast to the so far introduced chest belts, which were mainly designed to derive vital parameters HR and Heart Rate Variability (HRV) from ECG, the QuadriCore product page claimed to be the world's first ECG monitor designed to provide continuous medical grade data. The chest belt is a classic and robust biomedical sensor. With progress in sensor system integration, garments with built-in biomedical sensors were introduced. In the year 2016, a start-up company and corresponding product named Ambiotex were introduced, offering a shirt with integrated ECG electrodes. The Ambiotex shirt comes with attachable electronics for signal conditioning, signal processing, long-term recording, and wireless connectivity. Ambiotex was based on the technology platform FitnessSHIRT, which was developed by Fraunhofer IIS in Erlangen. The author of this thesis was involved as a software architect in the development of the embedded software. While Ambiotex has been introduced as a product, FitnessSHIRT [49] can be seen as a technology and research platform which has been in progress continuously. FitnessSHIRT has been designed in numerous variants. One of the remarkable prototypes was manufactured in additive manufacturing. Figure 2.2 illustrates a prototype for additive manufacturing of a sensor shirt with integrated fabric ECG electrodes and one fabric respiratory sensor. The textile was knitted in one manufacturing process without interruption using non-conductive and two different types of conductive materials, one for ECG electrodes and another conductive fabric for the respiratory sensor. After manufacturing, only two flat boxes with electronics were attached to the shirt. The system was an experimental prototype in a research project at Fraunhofer IIS back in 2009. Both Ambiotex and FitnessSHIRT achieved a battery runtime of approximately 24 h. A similar product from another company is Hexoskin. Hexoskin provides 14 parameters derived from Electrocardiograph (ECG), respiratory sensors, and accelerometer. According to the product data, the runtime of Hexoskin was improved from 14 h to 36 h [27] over two product versions.

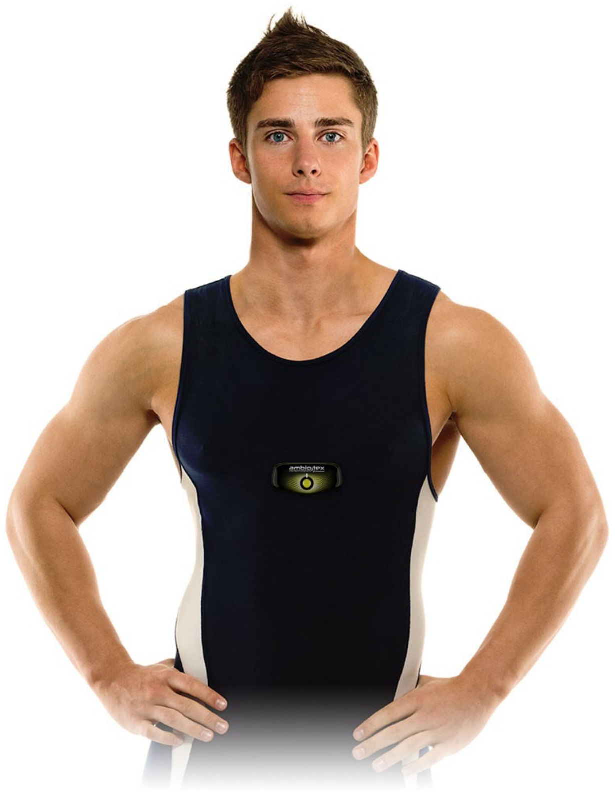


**Figure 2.1.:** Wearable product example: Ambiotex, a smart textile for training support by monitoring heart and stress parameters [43].

In contrast to chest-worn sensors, better battery runtime has been targeted with wrist-worn devices. As an example, in the table Jawbone UP3 enables monitoring of pulse rate and activity for at least 7 days with one battery charge. A newer product is the popular Fitbit Inspire 2, which was designed to track similar data for 10 days. Fitness trackers like Fitbit, along with chest belts like Polar H7, were leading the product spectrum in terms of extraordinary battery runtime. After fitness trackers, the first smartwatches were introduced at the market. However, the additional features of a smartwatch, compared to fitness trackers, require more energy. The Apple Smart Watch Series 4 was introduced in 2018, including an accelerometer, an optical pulse sensor and an ECG sensor. With 18 h of battery runtime, the Apple Smart Watch Series 4 was far behind fitness trackers. This was beaten by the Samsung Galaxy Watch 46 [39] in 2018 with a run time of 4 days while monitoring vital data. An extraordinary performance was achieved in 2019 with the HUAWEI Watch GT2 with specialized hardware with 14 days for monitoring vital data. The wrist is a comfortable place and people are used to wearing watches during daily activities, which makes vital monitoring predesignated at the wrist. Another body part of special interest for vital data monitoring is the ear. Hearables were introduced, which are smart headphones with additional features. The Munich start-up company Bargi introduced The Dash. The Dash was the first hearable product with integrated vital monitoring. The battery lasts for 5 h with vital data monitoring. This runtime has been beaten by the hearable Cosinus One [21] with 7 h on one battery charge.

So far, sensor systems have been placed on specific body parts. Other solutions address more freedom in body placement. Interuniversity Microelectronics Centre (IMEC)

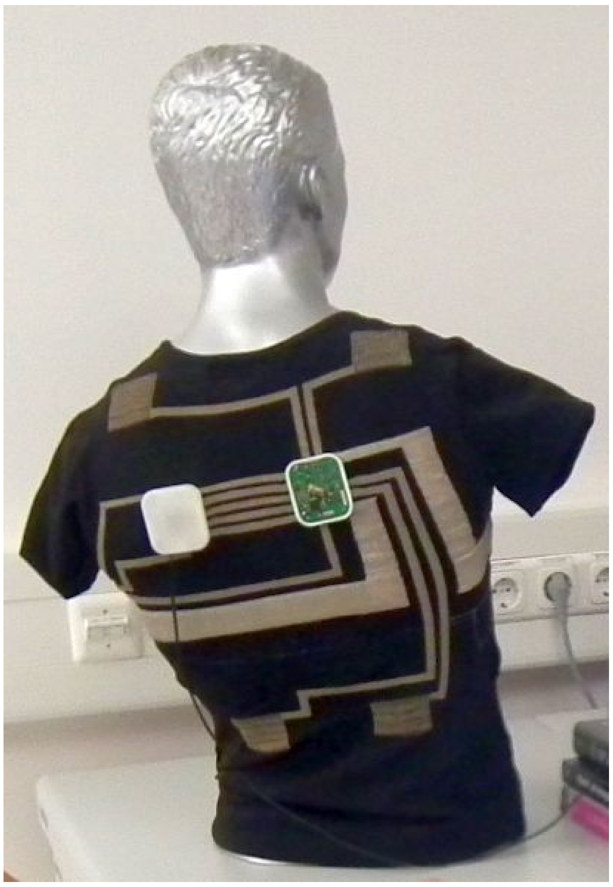


**Figure 2.2.:** Wearable research example: Additive manufactured wearable shirt with fabric ECG electrodes and fabric respiratory sensor developed at the Fraunhofer IIS in Erlangen [146]. This fabric was woven in one work cycle using an elastic non-conductive fabric and two different types of conductive fabrics for sensors and electric leads.

introduced a multi-sensor system based on a specially designed System on a Chip (SoC). Depending on the publication, different system settings were made, and thus the power consumption varies. Also, several battery sizes were investigated. According to Elfaramawy et al. [30] a battery runtime of three weeks should be possible. Another interesting solution was presented by Elfaramawy et al., where a respiratory patch network was created. The patch network runs on batteries for less than 8 h. The advantage of sensor patches is the freedom of placement at the body. Similarly, miniaturized sensor boxes exist that support free body placement in circumferences of practical sensor placement. The company Shimmer provides battery-powered sensors for data collection and monitoring tasks. Their sensor variety covers ECG, Electromyogram (EMG), plethysmography, accelerometers, gyroscopes, and magnetometers allowing computation of various biomedical parameters in the sensor data [14]. Shimmer is an acronym for Sensing Health with Intelligence, Modularity, Mobility, and Experimental Reusability. This sensing platform comes with open software, modular expansion capabilities, and software for data processing [25]. Al Disi et al. [31] have demonstrated how Shimmer can be used with custom software by implementing compressed sensing. In this work, a 6.4 times larger battery was used to achieve longer battery runtime in the range of 24 h. Therefore, for the data in the table, the power consumption was recalculated to Shimmer's standard battery capacity of 450 mAh with battery runtime data from [31]. With the Shimmer standard battery, a battery runtime of 1.6 h to 3.5 h would be possible, depending on the mode of operation in this publication. In conclusion, battery runtime ranges from a few hours to one week, depending on the system design and activated features.

Remarkable state-of-art reviews addressing wearable bio-sensing with respect to power consumption are Palumbo et al. (2001) [7] and Perez and Zeadally (2001) [8]. Perez and Zeadally published their work about recent advances in wearable sensing technologies. Palumbo et al. reviewed state-of-the-art of signal monitoring systems and sensors that are relevant to the field of telerehabilitation and health monitoring. Both papers identify energy consumption and battery runtime as major challenges that must be addressed at the system level.

**Table 2.1.:** Selected wearables and their technical properties regarding signal processing and energy budget.

| **Sensor System** | **Category** | **Sensors** | **Parameter Calculation** | **Radio** | **Power consumption** | **Battery** | **Runtime** |
|---|---|---|---|---|---|---|---|
| Polar H7 [67] | Chest belt | ECG | HR | Bluetooth 4.0 | 510 µA; and 430 µA when advertising | CR 2025, 150 mAh, 3 V | 200 h [67] |
| Zephyr Bioharness 1 | Chest belt | ECG, resp., accel., temp. | ECG, HR, HRV, resp. rate, resp. Amplitude, activity level, skin temp., posture | Bluetooth 2.0 | 97 mW | 150 mAh, Li-Ion | 6 h to 18 h |
| Zephyr Bioharness 3 | Chest belt | ECG, resp., accel., temp. | ECG, HR, HRV, resp. rate, resp. amplitude, activity level, skin temp., posture | Bluetooth 2.1 | | | 12 h to 24 h |
| Qardiocore [18, 52] | Chest belt | ECG, temp., accel., resp. | HR, HRV, resp. rate, skin temp., activity level | Bluetooth 4.0 | | Li-Ion | 1 day |
| Ambiotex [42], [54] | Shirt | ECG | HR, HRV, stress level | Bluetooth 4.0 | | | approx. 24 h |
| FitnessSHIRT [49] | Shirt | ECG, resp. | ECG, HR, HRV, resp. rate | Bluetooth 4.2 | Wireless 13 mA; recording to memory 10 mA | 300 mAh, LiPo, nominal 3.7 V | 20.8 h |
| Hexoskin Classic Device [27] | Shirt | ECG, resp., accel. | 14 parameters | Bluetooth 2.1 | | | 14 h |
| Hexoskin Smart Device [27] | Shirt | ECG, resp., accel. | same as classic | Bluetooth 4.1 | | | 36 h [27] |
| Jawbone UP3 | Fitness Tracker | accel. | | Bluetooth 4.0 | | | 7 to 8 days |
| Fitbit Inspire 2 | Fitness Tracker | PPG, accel. | HR, HRV, set of activity parameters | Bluetooth 4.2 | | | 10 days |
| Apple Watch Series 5 | Smart Watch | ECG, PPG | | Bluetooth 5.0 | 61 µW (calculated) | 296 mAh, nominal 3.7 V | 18 h after software bugfix |
| Samsung Galaxy Watch 46 [39] | Smart Watch | PPG, accel. | HR, activity classification | Bluetooth 4.2, WiFi | 18.2 mW | 472 mAh | 4 days |
| HUAWEI Watch GT2 (46 mm) | Smart Watch | PPG, accel. | $SpO_2$, HR, activity classification | Bluetooth 5.1, WiFi | | 455 mAh | 14 days monitor-ing, 30 h with GPS |
| Bargi The Dash | Hearables | PPG, accel. | $SpO_2$, HR | | | 100 mAh | 5 h |
| Cosinus One [21] | Hearables | PPG, temp., accel. | | Bluetooth 4.0, ANT+ | | 50 mAh, Li-Ion | 7 h |
| IMEC MultiSensor SOC [30] | Patch | ECG, PPG, resp., accel.and more | HR, resp. rate,$SpO_2$ | Bluetooth 4.2 | 2018 µW | 1260 mAh | 3 weeks |
| Respiratory Patch Network [22] | Patch Network | resp., accel., gyro. | QRS detection | nRF24L01 2.4 GHz | 12 mA to 16.2 mA | 100 mAh, Li-Ion | $<$ 8 h |
| Shimmer3 ECG/EMG [74] | Sensor Box | ECG; EMG, resp., accel., gyro., magnetometer, and altimeter | HR + custom algorithms | Bluetooth 4.0 | 129 mA to 276 mA | 450 mAh, Li-Ion | 1.6 h to 3.5 h |

### 2.1.2. Special Challenges of Wearable Development

Wearables differ from stationary devices in size and power supply. Wearables have to be lightweight, compact, and comfortable. On the contrary, stationary monitoring devices are taking advantage of the additional space and electrical power. Thus, the size constraints are limiting wearables to small battery sizes, small electronics, and less powerful digital signal processing. Unfortunately, wearables actually require better signal processing than stationary devices. Better signal processing would be necessary for handling the extra motion artifacts on wearable systems. However, the signal processing power on a wearable sensor system is strongly limited because of the limited size and limited energy. Therefore, energy-hungry signal processing would be necessary for advanced motion suppression algorithms, but computing power is highly limited on wearables.

Another challenging difference between wearables compared to stationary devices is a lack of control by professional operators. Wearables are made to generate long-term data, being used over days, weeks, and even months without professional supervision. In contrast, stationary devices are supervised by professionals. Professional personal can check the quality of the signals, the fitting of electrodes, correct sensor placement, and correct it if necessary. However, wearables are left alone with the wearer, and all the supervision should be replaced somehow technically by the smart capabilities of the product.

A summary of the problem can be formulated as follows: Mobile bio-sensing is an attempt to achieve reliable data from motion-sensitive sensors during daily life activities instead of stationary conditions with signal processing that runs on a highly restricted low-power device. This is the challenge that hardware and software engineers face when designing wearables compared to stationary devices.

### 2.1.3. Wearable System Architecture

All wearable sensor systems share a basic architecture on a very high level of abstraction. The IEEE Standard for Wearable Consumer Electronic Devices [1] defines fundamental components as parts of an architecture. Figure 2.3 is another view on similar architectural components, additionally highlighting the information and energy flow. This figure illustrates the elementary components at a high architectural level that all biomedical wearable sensor systems share. Most parts of the wearable system require energy to run a signal processing task. Consequently, there must be at least one part providing the energy. Designs are getting more advanced if there is not just one power source like a battery, but at least a second source like an energy harvester. Moreover, it is a good design idea to implement a power distribution network that enables control of energy distribution by providing energy to the parts which currently contribute to the signal processing task exclusively. All parts communicate with each other over some kind of analog or digital interface which are not necessary to pay attention to here in detail. Interfaces between the parts will be addressed later.

The signal processing task begins at the left side of the figure with the sensor. The sensor provides a representation of a physical property that can be technically processed further. The sensor is followed by a sensor front-end. The purpose of the sensor front-end is to amplify weak signals, limit the signal to the necessary bandwidth, and digitize the signal for further processing, optionally. Some sensor front-ends provide additional functionality for active sensors, which require controlling timing and intensity parameters.

However, many modern sensor front-ends are based on mixed design, which includes analog signal processing combined with digital processing. All sensor systems require some kind of system control. System control executes the start sequence, controls the entire sensing process, implements sensing applications, handles errors, and many security and communication-related tasks. In the illustration, the signal processing was logically separated from the system control, even if both are technically often implemented on one computing platform. The signal processing, mostly digitally implemented, takes signals from the sensor front-ends and implements more advanced processing. State-of-the-art sensor systems provide wireless connectivity for data transmission and remote configuration. Optionally, some devices combine wireless connectivity and data recording in one sensor system.

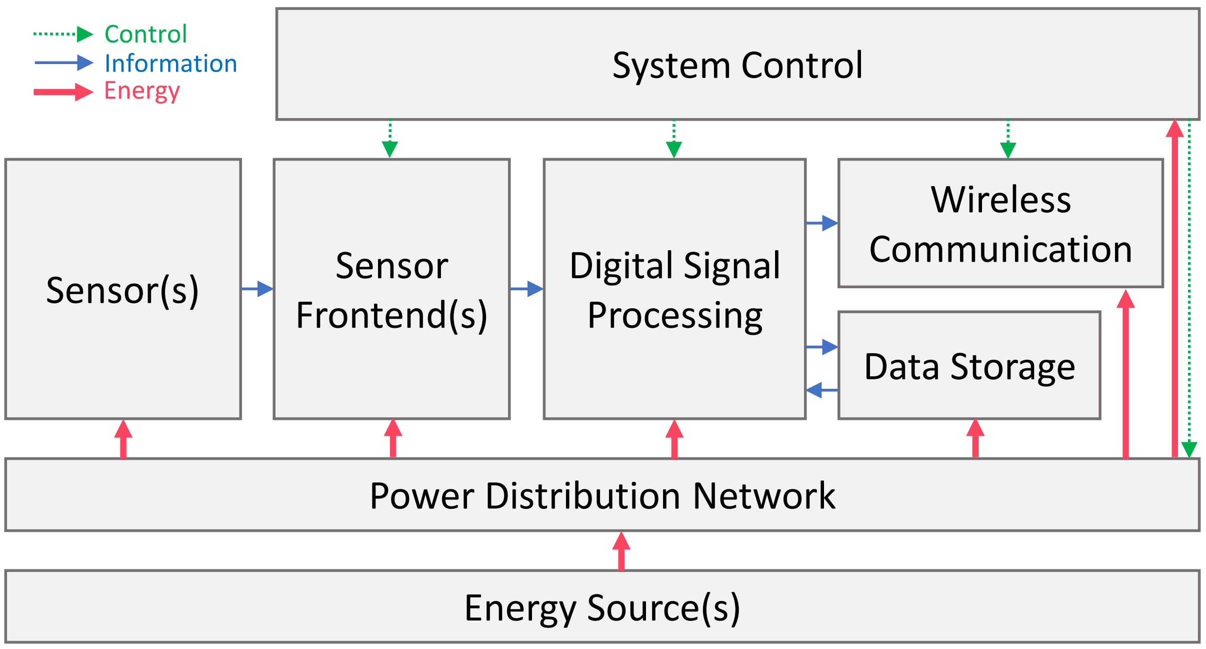


**Figure 2.3.:** This diagram shows the essential building blocks of a wearable sensor system at a high level of abstraction. The blocks are linked with arrows representing three types of flows: control (e.g. configuration and commands), communication (sensor data and status information), and energy.

The underlying component in figure 2.3 is the power distribution network. The power distribution network delivers energy to all of the components mentioned above. Each energy consumer has different requirements regarding power supply. The task of the power distribution network is to provide each participant with an appropriate energy supply in terms of voltage level, voltage noise level, maximum current, and power gating functionality. At the same time, it is the task of the power distribution network to optimally utilize the battery and protect it from overload. The last component in figure 2.3 is the energy source, which is often a battery supplemented by a charging circuit. Optionally, batteries can be supported or completely replaced by energy harvesting. These are the building blocks of wearable sensor systems.

### 2.1.4. Key Areas of Low-power Optimization

The improvement of battery runtime for wearable biomedical sensor systems addresses several research areas at the same time. These research areas are similar to the general field of application within the Internet of Things (IoT). Power optimization for IoT can be organized in four research areas:

1. Battery technology
2. Energy harvesting
3. Energy Efficient Wireless communication
4. Energy Efficient Sensing and Signal Processing (research focus in this thesis)

The latest area, low-power sensing and signal processing, is the focus of this thesis. Nevertheless, the other areas play an important role from a holistic point of view. A brief insight into the four areas is given hereafter.

#### Battery Technology

In recent years batteries were improved, which enabled many desired applications. Higher energy density enabled wearable applications. However, with higher energy density, the handling of batteries has become more complicated and dangerous. Therefore, Hahn and Reichl [177] reviewed power supplies for wearable and ubiquitous computing. Additionally, Rao et al. [169] reviewed battery models for energy-aware system design. Those models consider discharge rate, temperature effects, and capacity fading. In conclusion, according to Martin et al. [179], battery behavior must be considered to precisely make forecasts about runtime. Battery technology is not the focus of this thesis.

#### Energy Harvesting

Batteries are not the only energy source. Energy harvesting takes energy from the environment to supply a sensor system. Batteries were supplemented or, in some cases, replaced completely by energy harvesting. The first self-powered biomedical sensors were investigated by Leonov [111, 130]. An overview of different energy harvesting methods was given by Yeatman and Mitcheson [104]. Related work on energy harvesting for biomedical sensors was done by Voss et al. [101], Misra et al. [73], and Dionisi et al. [58] including harvesters integrated into a garment Brogan et al. [82] and Ruiz et al. [9]. An interesting energy source for wearable sensor systems is the heat generated by the human body. An overview of energy harvesting from the human body for biomedical applications was given by Selvarathinam and Anpalagan [68]. For this purpose, Mateu et al. [95] published work on efficient harvester circuits. Even though energy harvesting was outside the focus of this thesis, the sensor system developed here was tested in an experiment with an energy harvester from Fraunhofer IIS. This will be the content of section 6.3.

### Energy Efficient Wireless Communication

A popular research area was the improvement of power efficiency by optimizing wireless data transmission. This area was driven by the fact that old radio standards were the highest energy consumer on a wireless sensor system. Specialized standards addressing low-power applications are available for long-range applications such as Long Range Wide Area Network (LoRaWAN) and short-range applications such as Bluetooth Smart, also known as Bluetooth Low Energy which was introduced as part of the Bluetooth 4.0 standard introduced in 2009. These low-energy standards can be used with standard parameters or tuned for the best performance. Therefore, Kindt et al. [62, 92] investigated smart timing strategies and adaptive Bluetooth parameter optimization for improved power efficiency. The key to optimizing the transmission duty cycle is to reduce the power of the radio module according to [103, 121, 123]. Zanaj et al. [11] presented an extensive review of wireless communication technologies, including both Low-Power Short-Area Networks (LPSANs) and Low-Power Wide-Area Networks (LPWANs), from the perspective of energy efficiency and power consumption.

Furthermore, efficient data transmission also requires protocols with low data overhead. Therefore, Pandesswaran et al. [66] investigated Constrained Application Protocol (CoAP) for remote patient monitoring. Communication protocols like IEEE 11073 are specialized to medical applications [163]. Therefore, Santos et al. [98] estimated the overhead of IEEE 11073 combined with CoAP for usage on constrained devices. In the best-case configuration for a payload of 172 bytes, a total of 599 bytes was transmitted. Therefore, the choice of communication protocol on a constraint device can have a major influence on the required power for every transmitted bit.

As everybody can listen to the wireless transmission, further overhead is added through security-related features, e.g. encryption, hashing, and key exchange. Therefore, Akbas et al. [16] compared lightweight cryptography algorithms on resource-constrained microcontrollers, in particular on 8-bit microcontrollers. However, on several microcontrollers, hardware acceleration for encryption algorithms exists. According to investigations by Bota et al. [107], hardware encryption requires less energy than software implementation. With this, the overhead for wireless transmission is enhanced for every additional feature in a fully populated state-of-the-art product.

Wireless energy efficiency is a wide research field that is still ongoing. However, this research area is not the focus of this thesis.

### Energy Efficient Sensing and Signal Processing

Energy-efficient sensing and signal processing is the focus of this thesis. The main task of a sensor system is to acquire sensor data, followed by at least rudimentary signal processing. This research area addresses the following subcategories.

- Low-power sensors
- Low-power sensor front-ends
- Efficient hardware for running the algorithms
- Efficient algorithms

- Appropriate design tools (e.g. compiler)
- Efficient software libraries supporting the signal processing chain

Almost all silicon manufacturers followed the demand for low-power components by introducing specialized products for sensing, amplifying, filtering, and digital processing of sensor signals. For the analog signal processing part, low-power sensors and ECG front-ends were proposed with continuous power requirements of only 2.76 µW [137, 143] or 2.6 µW [145]. However, these parts were primarily investigated in research projects and are not all commercially available. Research results are ahead of commercial products. An example of a commercially available system is the ECG front-end AD8232. It requires 170 µW under the best conditions [12]. Nevertheless, a clear trend towards low power can be observed. The above front-end is partly analog and partly digital; in other words, it is a mixed-signal component. For the digital part, various microcontroller and software solutions emerge on the market, addressing particularly low-power applications. Optimized signal processing algorithms are required to process data efficiently. Beside this, digital signal processing can be optimized either for high performance or in case of wearable sensors for low power according to Oshana [120]. However, the algorithm's power efficiency depends on a particular implementation designed for underlying computing hardware. This implementation may be affected by a specific compiler that translates the algorithm representation from a high-level language to machine code that the hardware can execute. Moreover, the algorithm's power efficiency relies on supporting software components, like an operating system that provides required functionality, software abstraction layers, hardware accessing libraries, mathematical libraries, data buffers for interconnecting signal processing software, and many more or less power-efficient software components. All these things together impact power consumption along the signal processing path.

### 2.1.5. Monitoring Program on a Sensor System (Mode of Operation)

Every wearable sensor system has a monitoring program that implements a monitoring task with respect to the available resources (e.g. energy). Another synonym commonly used is Mode of Operation. However, in this thesis, the term monitoring program will be used as it fits the meaning more generally. It is important to note that this monitoring program may be fixed or configurable. If the monitoring program is configurable, its existence may appear more noticeable. However, such a monitoring program is an essential part of the application which implements the intended use of a smart sensor system. Moreover, the monitoring program is of special interest when optimizing for low-power. Four fundamental monitoring program groups were identified by reviewing state-of-the-art system designs and the corresponding literature.

1. Continuous monitoring
2. Interval monitoring
3. Episodic monitoring
4. Event driven monitoring

### Continuous monitoring

Continuous monitoring refers to an always-on operation: sampling and processing are performed without interruption. In continuous monitoring, either data is transmitted raw without processing or by reducing the data by transmitting only parameters. Continuous monitoring can also include event-based transmission after evaluation of the data on the node. However, the unique property of continuous monitoring is that the sensor system is always active in sensing the data.

### Interval monitoring

Interval monitoring refers to a fixed period of time a sensor system performs monitoring, followed by a fixed time for pausing the monitoring. Wearable sensor systems are driven by the motivation to measure more frequently than a regular visit to a physician's office. As an example, the Samsung Galaxy Watch [39] offers a mode of operation that takes a heart rate measurement every 15 minutes. Makikawa et al. [94] gives examples for monitoring frequencies based on application: 1) once a day for a health check; 2) a few minutes every day for blood pressure monitoring.

### Episodic monitoring

Episodic monitoring is almost identical to interval monitoring, except that the timing parameters are controlled adaptively. As an example, the control algorithm introduced by Au et al. [139] and utilized by Bonfiglio et al. [126] decreases and increases the pause duration depending on the measured data. Therefore, a task-specific algorithm was implemented which distinguishes between abnormal and normal data. Once the data was identified as abnormal, the duration of the pause deceases. On the contrary, once the monitored data returns to its normal state, the duration of the pause increases. The pause range was also limited by a configured minimum and maximum.

### Event driven monitoring

Event-driven monitoring is a method when sampling is started on a special event. The special event must be detected in some way while the actual sampling is deactivated. Thus, this is mostly implemented by using a second sensor which consumes less power. As an example, the Samsung Galaxy Watch [39] offers a mode of operation that takes a heart rate measurement only when the accelerometer detects low motion. This mode was introduced to improve accuracy and power efficiency. A similar example is the rest heart rate which was described by Nanchen [38]. Detecting the resting heart rate requires the detection of a prolonged rest condition before starting the measurement. Another example was presented by Sun et al. [132] increasing the power efficiency by a factor of 2.5 through event-based ECG monitoring controlled by activity recognition.

### Conclusion on Monitoring Programs

Consequently, sensor systems follow a predefined monitoring program. Depending on the application, it is not necessary to measure continuously. If the omission of a measurement can be legally justified, then it makes sense to use this period of inactivity to reduce energy consumption. A method for reducing energy consumption is using low-power modes if monitoring can be paused. Manufacturers of sensor hardware and signal processing hardware have been addressing this technique by providing different types of low-power modes. Therefore, low-power Integrated Circuits (ICs) provide a shutdown input pin intended to be controlled by another hardware component. Another way is provided by more complex devices like microcontrollers or mixed-signal sensor frontends, which provide multiple power modes controlled by software. Caples [83] defines *Hibernation* as the capability of a system to go offline to the degree that corresponds to the duration of the inactivity and restart time constraints. In general, for hibernation, all hardware parts require to be shut down on demand. Finally, in hibernation the sensor system must be capable of wake-up by itself, which requires some low-power hardware being left active while hibernating. Such as, hibernation is not equal to power-off. Therefore, a low-power timer or a low-power sensor serves for time-based or event-based wake-up generation.

In summary, the monitoring program on a sensor system determines whether a hibernation can be used at all, when is the optimal time for hibernation, and how long the hibernation should last. Thus, the monitoring program on a sensor system has a high impact on the energy budget.

## 2.2. Biomedical Signals

Biomedical signals are observations of the underlying processes of living organisms. Analysis of biomedical signals allows conclusions about the health of a living being. Checking the pulse is a simple way to get some insight into the health of the cardiovascular system. The pulse can be felt easily when an artery comes close to the skin's surface. This sensing can be done by placing the index and middle finger on the underside of the wrist. Finally, counting the beats for a known time allows measuring the pulse rate.

### 2.2.1. Electrocardigraphy

The electrocardiogram has been an effective instrumentation for understanding and treating heart diseases, particularly for fighting the main cause of death worldwide: CVD. Augustus D. Waller [222] published work on the electrical activity of the human heart in 1887. There he analyzes the heartbeat using measurement equipment with some disadvantages regarding the bandwidth. He named the distinctive points according to the alphabet A, B, C, D, and E. Willem Einthoven studied August D. Waller's work. Einthoven proposed to use another measurement equipment that allows recording faster changes. Einthoven published his work analyzing the shape of the ECG in the year 1895 [221]. In figure 2.4 Einthoven compares his high-bandwidth recordings with Waller's recordings. Einthoven introduced the letters P, Q, R, S, and T for the distinctive points in the signal. Einthoven proposed the string galvanometer, which was the first Electrocardiograph. The proposed method gained popularity in the medical field, and he received the Nobel Prize in Medicine for his development [148].

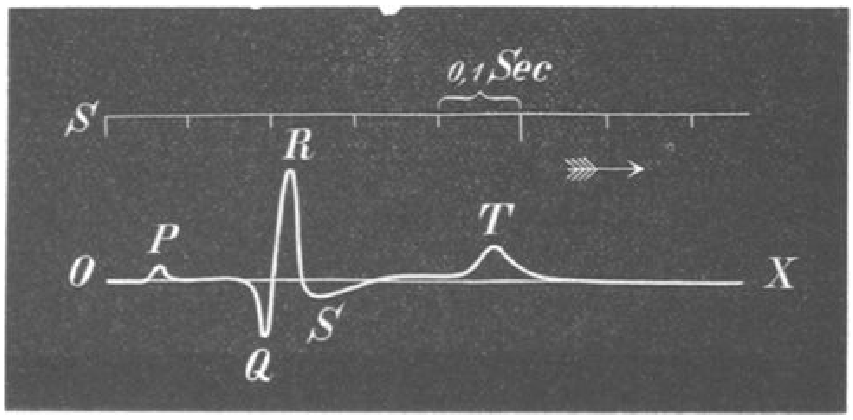


**Figure 2.4.:** Original ECG figure with annotation by Einthoven from his publications titled in German language “Über die Form des menschlichen Electrocardiogrammms”, 1895 [221]

## ECG lead systems

An ECG lead system defines the electrodes' placement and the leads, which are then derived from the electrical signals. Additionally, standard ECGs have an actively driven electrode at the right leg for common mode rejection: Driven Right Leg (DRL). This electrode does not play a role in the lead system – it will be postponed for later discussion. A lead is a combination of electropotentials at the passive electrodes. For example, leads I, II, and III, referred to as the Einthoven Lead System, are derived from the limbs by measuring the difference at each pair of electrodes. By default, the electrodes are placed at the Right Arm (RA), Left Arm (LA), and Left Leg (LL). The electrodes must not necessarily be placed at the arms. There are also alternative positions at the chest. With these three electrodes, three leads are generated: Lead I, for example, is the potential between the RA and the LA. Three additional leads can be generated with the same electrode setup: aVL, aVR, and aVF, also known as the Goldberger Lead System. The potentials between the electrodes span a vector field. The leads offer views of the heart's electrical activity from different angles. Further ECG lead systems exist, but the standard for clinical diagnosis is the 12-lead ECG system. This 12-lead system defines 10 electrodes at distinctive positions for limb electrodes RA, LA, and LL; the active DRL; and additional six frontal electrodes V1 to V6. Integrating 10 electrodes into a wearable system is difficult. An attempt for a 9-lead system integrated into an ECG is CardioTEXTIL by Hofmann et al. [48]. However, it is not always necessary to examine all leads. Some leads are required only for specific diagnostics. Many parameters can be derived even from a single lead ECG. Likewise, even a subset of arrhythmia can be detected with single lead ECG, in particular, lead I across RA and LA. A single lead system requires at least three electrodes in general: RA, LA, and Right Leg Drive (RLD). However, solutions exist to avoid the usage of the RLD electrode. The solution was already discussed in 1980 by Thakor and Webster [211] and further commented on by Towe [209]. They distinguished between monitoring and biotelemetry. Their solution was adapted for the designs of portable systems over the years. For example, the AD8232 [12] is an ECG front-end for portable devices. There, a circuit is given where the DRL output can drive a third electrode or be fed back over two resistors, each 1 MΩ, to the leads RA and LA alternatively. With this, a single lead system requires only two electrodes.

### Wearable Electrodes

Wearable electrodes, particularly the ECG electrodes, differ from clinically used electrodes. ECG electrodes can be classified into three types according to Makikawa et al. [94]: Skin electrode, capacitively-coupled electrode, and dry electrode. The skin electrode, also called a wet electrode, is the most popular and requires a contact gel that must be supplied separately or is an integral part of a disposable electrode. While skin electrodes provide the best signal quality and resistance to artifacts, the contact gel is impractical for permanent wearable integration. The capacitively-coupled electrode reached the practical level for integration in wearables. More common for integration in wearables were dry electrodes. Dry electrodes are primarily based on conductive polymers [81]. Recent development show integration of dry electrodes in wearables [5]. Even so, capacitively-coupled electrodes and dry electrodes are more suitable for wearables. Capacitively-coupled electrodes show relativity higher sensitivity to artifacts caused by motion compared to skin electrodes.

### Bandwidth

With Waller and Einthoven's contributions, it is known that bandwidth in ECG matters. Nowadays, bandwidth in the range of biosignals is not a technical problem. For a product development, standards define a bandwidth for diagnostic ECG. The European standard DIN EN ISO 60601-2-27 [159] claims a minimum frequency range from 0.67 to 40 Hz for diagnostic ECG. The Association for the Advancement of Medical Instrumentation (AAMI) confirms a bandwidth of 40 Hz for diagnosis and automated analysis of ECG as recommended in [170]. However, this is the minimum bandwidth and products may cover a wider bandwidth for diagnostic ECG. Unfortunately, motion artifacts and sensitivity to other noise sources increase with wider bandwidth. This sensitivity to motion is particularly a problem for wearable devices, comparing the relatively controlled conditions with stationary devices. This problem also applies to stationary devices, which therefore implement configurable bandwidth. For example, the Infinity M450 declares multiple frequency ranges in the manual [70]: the narrowest from 0.5 to 16 Hz and the widest from 0.05 to 150 Hz. In conclusion, the bandwidth is a tradeoff between artifact suppression and the realizable set of applications.

### Beat Detection

The task of a beat detector is to detect a QRS complex in an ECG time series. Beat detection in ECG is the starting point for many signal processing algorithms. Knowing the location of the QRS within the ECG signal is a precondition for many following steps in ECG processing. Once the QRS was located, other distinctive points can be found. For example, the R wave is the maximum within the QRS, the P wave is located to the left of the QRS, and the ST segment is located to the right of the QRS. Finally, calculating the heart rate requires a sequence of detected beats. These were just a few examples of possible signal processing tasks that require a beat detector within a signal processing chain.

In general, beat detection is an essential orientation for further analysis with specialized algorithms. Therefore, multiple beat detectors were designed. The detectors differ in their performance for detecting the beats. To evaluate beat detectors, databases with manually annotated ECG beats exist. An established procedure is to rate the performance of beat detectors by two indicators: sensitivity and Positive Predictive Value (PPV). PPV is defined as the fraction of True Positive (TP) divided by the sum of TP and False Positive (FP).

$$PPV = \frac{TP}{TP + FP} \tag{2.1}$$

Sensitivity is defined as the fraction of TP divided by the sum of TP and False Negative (FN).

$$Se = \frac{TP}{TP + FN} \tag{2.2}$$

The indicator PPV has a range between 0 and 1. The indicator PPV has the value 1 when there are no false positive beat detections. As algorithms are imperfect, beats will be detected at the wrong positions where no beats are marked in the data manually. With this PPV is $< 1$. Sensitivity has a range between 0 and 1. Sensitivity is 1 when there were no false negative beat detections at all. As algorithms are imperfect, beats will be missed at a position where beats were marked.

Besides performance, another criterion for beat detectors is the algorithm's computational complexity. The Pan-Tompkins algorithm published by Pan and Tompkins [205] in 1985 became a popular beat detector. In their publication, Pan and Tompkins implemented the algorithm on the at that time popular Z80 processor and estimated the accuracy with the MIT/BIH arrhythmia database. Meanwhile, many other implementations of the algorithm exist. As computers became faster also algorithms became more advanced. However, low computational complexity matters again because of the limited resources for wearable computing. Köhler et al. [172] published a review comparing algorithms for detecting QRS complexes. Elgendi et al. [88] reviewed QRS detection methodologies for portable, wearable, battery-operated, and wireless ECG systems. They declared three major criteria for QRS detector algorithms: 1) robustness to noise, 2) parameter choice, and 3) numerical efficiency. Both publications classify computational complexity in three classes: low, medium, and high. Liu et al. [35] defined computational complexity by computing time while estimating the performance for 10 beat detectors with six different databases. The rating there is more quantifiable. However, it was computed on a Personal Computer (PC) and does not consequently apply to devices with restricted resources. Elgendi et al. [88] analyzed the algorithms in their computational steps and their complexity assessment correlates with Köhler et al. [172].

Table 2.2 contains a collection of selected publications. The algorithm's computational cost classification was taken from Köhler et al. [172] and Elgendi et al. [88]. Additionally, the publications were rated in the table by performance. Therefore, the measure according to Köhler et al. [172] was adopted by taking the minimum of PPV and sensitivity. This rating was given in percent quantized in three classes: >90%, 95%–99%, and <95%.

Wearable computing benefits from good-performing algorithms with low computational complexity. Consequently, when algorithms exist in each of the nine groups presented in table 2.2, the best for wearable computing would be the upper left group. There, the performance of the beat detector is above 99% and the computational complexity is

**Table 2.2.:** The performance-complexity tradeoff for different beat detectors was verified with standard databases.

| **Perfor-mance** | **Algorithm Complexity** | | |
|---|---|---|---|
| | **Low** | **Medium** | **High** |
| **> 99%** | Afonso et al. [176]<br>Köhler et al. [168]<br>Ghaffari et al. [151] | Bahoura et al. [182]<br>Hamilton & Tompkins [202]<br>Hamilton [171]<br>Poli et al. [191]<br>Gritzali [201]<br>Ruha et al. [185]<br>Vijaya et al. [181]<br>Zidelmal [124] | Inoue and Miyazaki [184]<br>Li et al. [190]<br>Xue et al. [199]<br>Hu et al. [194]<br>Sahambi et al. [186]<br>Chiarugi et al. [155]<br>Christov [165]<br>Elgendi [109] |
| **95%–99%** | Suppappola & Sun [192, 198]<br>Choukri [127] | Pan & Tompkins [205]<br>Kadambe et al. [178, 197]<br>Elgendi et al. [150]<br>Arzeno et al. [147]<br>Deepu & Lian [87] | Coast et al. [200] |
| **< 95%** | Trahanias [195] | Ligtenberg & Kunt [207] | Papakonstantinou et al. [203] |

low. However, 99% is not the threshold for excellent beat detectors. Better-performing algorithms exist. For example, with sensitivity at 99% assuming a heart rate of 75 bpm there was an average rate of 45 missed beats per hour in the test data set. Additionally, the label High Complexity is subjective and does not prevent an algorithm from running efficiently on a low-power system. Although this table is a good starting point, a detailed investigation is required to determine whether or not an algorithm is suited for wearables.

### RR interval

Once QRS complexes were found, R peaks can be detected in ECG. Then, an RR interval is a time between two successive R peaks within an ECG. The computing of RR intervals is the baseline for many ECG processing algorithms.

### HR

HR is computed from RR intervals. By calculating the reciprocal value and scaling the time base to minutes, the HR is given in Beats Per Minute (bpm). Different algorithms exist for computing HR from RR intervals. HR can be computed from only two successive RR intervals. In fact, beat-to-beat HR computing exists. However, it is difficult to read beat-to-beat HR on a screen due to the physiological variations in the RR interval. Therefore, HR is computed as the average of multiple beats. For example, the manual for the monitor Infinity Acute Care System [47] declares how the parameter HR was computed. First, the manual specifies a range of 15 to 300 bpm. Then, the manual declares that the latest 10 seconds will be used for the following computations. Further, it declares that the calculation excludes outliers. Finally, the remaining RR intervals are averaged. In the above example, the averaging time was fixed. Some devices allow adjusting the averaging time. Other devices provide configuration for averaging the number of beats, instead of selecting the number of beats within an adjustable time interval.

There are standardized regulations for the algorithms. The standard DIN EN ISO 60601-2-27 [159] declares the minimum requirements for the European domestic market for heart rate monitors. There, the minimum range for heart rate is 30 to 200 bpm for monitoring adults, 15 to 300 bpm for children, and 15 to 350 bpm for neonates. The minimum accuracy is $\pm 10\,\%$ or $\pm 5\,$bpm.

To sum up, the computing of HR may differ in detail depending on the algorithm. In practical usage, the algorithm may be relevant for interpreting the HR parameter. Therefore, the algorithm was specified in the medical device manual in the example above.

### HRV

The heartbeat is irregular. A dedicated measure for the irregular behavior of the beats is the Heart Rate Variability (HRV). HRV is an easily measurable examination method of the autonomic nervous system. A disorder of the autonomic nervous system leads over time to cardiovascular disease, inflammation, mental vulnerability, and increased mortality, as summarized by Fouradoulas et al. [23]. Fenzl and Schlegel [133] summarized the literature on defining a normal range for HRV. HRV itself is not a parameter. HRV is a category for a group of parameters that all describe the variations of RR intervals. The definitions were taken from “Heart rate variability: Standards of measurement, physiological interpretation, and clinical use” [188] published by the Task Force of The European Society of Cardiology and The North American Society of Pacing and Electrophysiology. Before going forward, HRV can be calculated not only for RR intervals. Therefore, [188] defines NN intervals as a general term for two distinctive markers in a cardiac signal. N is mostly defined as R for an ECG. However, in the following definitions, the recommended definition will be used. Two HRV parameters will be introduced in this thesis: RMSSD and pNN50%. Root Mean Square of Successive Differences (RMSSD) is represented by the equation defined in [188].

$$RMSSD = \sqrt{\frac{1}{M}\sum_{i=1}^{M}\left(NN_i - NN_{i-1}\right)^2} \tag{2.3}$$

The second parameter, pNN50%, requires the calculation of NN50 first, defined in [188]. NN50 is calculated by counting adjacent pairs of NN intervals larger than 50 ms. pNN50 is the ratio of NN50 divided by the count of all NN intervals. And finally, pNN50% is given in percent.

## 2.2.2. Respiratory Plethysmography

Respiratory plethysmography, in particular the Respiratory Inductance Plethysmography (RIP), is a common technique for diagnosis of obstructive sleep apnea [63]. Therefore, algorithms exist for detecting obstructive sleep apnea events. Another application is to compute tidal volume using multiple respiratory bands around the chest, as proposed by Leutheuser [50]. Schmitz [99] showed similar results using seven resistive and inductance sensors around the thorax and abdomen. Another interesting parameter of respiratory signals is the respiration rate. However, the respiration signal is not always periodic – it has many shapes and facets, many more than other biosignals. The respiratory signal is almost sinusoidal during exercise, irregular in rest conditions, and when a person is talking it has a prolonged exponential decay followed by irregular fast inhales in between. With this, depending on the mentioned signal shape, it may be possible to gain information from the respiratory signal easily, or it may be almost impossible to compute parameters such as the signal frequency. Aliverti [44] gives further examples and a brief overview of the role of wearable technology for respiratory health monitoring. The solution focuses on the integration of respiratory sensors into garments. However, another solution exists such as a wireless respiratory monitoring system using a wearable patch sensor network, proposed by Elfaramawy [22]. Three sensing principles for respiratory plethysmography will be introduced: resistive, capacitive, and inductance plethysmography.

### Resistive Plethysmography

Resistive sensing was used in the FitnessSHIRT sensor shirt [49]. The sensing element is a polymer with metal particles that changes the resistance with the elongation of the material. The advantage is a simple sensing procedure using a current source and a transimpedance circuit. The drawbacks are a tight fit and a hysteresis effect which cause a difficult-to-evaluate signal.

### Capacitive Plethysmography

Capacitive sensing was used for a new generation of sensors at the Fraunhofer Society. A composite material of electrically conductive silicone and layers of insulating silicone was used to create a capacitor. The equation 2.4 describes the capacitance of a plate capacitor in dependency of the area $A$ and the plate distance $d$. With the stretching of the fabric material, the plate area $A$ of the silicone capacitor increases. At the same time, the insulating center layer gets thinner, decreasing the distance between the capacitor plates $d$. Using the formula for a plate capacitor, an increasing area and a decreased distance increase the capacitance $C$.

$$C = \epsilon_0 \epsilon_r \frac{A}{d} \tag{2.4}$$

The advantage of the capacitive sensor is a simple sensing procedure using a current source, capacitor discharge circuit, a comparator, and a timer for measurement of the time which is required to charge the variable capacitor. Thus, the equation 2.5 describes a capacitor voltage $u(t)$ charged by a current $i(t)$ in general.

$$u(t_1) = \frac{Q(t_1)}{C} = \frac{1}{C} \int_{t_0}^{t_1} i(t)\mathrm{dt} + u(t_0) \tag{2.5}$$

Using the capacitor discharge circuit before every measurement $u(t_0) = 0$, a constant current $i(t) = i_c$, assuming an average capacitance $(C_a)$, and a threshold voltage at comparator $(u_T)$ results in equation 2.6.

$$\Delta t_T = \frac{u_T i_c}{C_a} \tag{2.6}$$

This equation describes the charging time $(t_T)$ which was then measured by the microcontroller. The drawback of the capacitive sensor is a tight fit as the composite silicone sensor requires pretension.

### Inductance Plethysmography

Respiratory inductance sensors are known under the term RIP. RIP is the most used technique for portable respiratory plethysmography. The integration of RIP into a garment was claimed with a patent by Tobola and Weigand [154] in 2008. The primary electronics of this garment was a sensor module that provided two channels of digitally processed respiration signals. This module can be seen in Figure 2.5. In this garment, the sensor was a wire integrated around the thorax or abdomen, forming a coil around the area $A$ with a variable inductance $L$. The coil was integrated into a close-fitting garment. Because wires are not flexible, they were woven in a sinusoidal shape around the body, allowing the textile space for stretching. This sensor system uses an oscillator for generating a frequency $f$ at approximately 1 MHz according to equation 2.7.

$$f = \frac{1}{2\pi\sqrt{LC}} \tag{2.7}$$

The oscillator's frequency depends on a fixed capacitor $C$ and a variable inductance $L$. The oscillator's signal is provided to a counter at the microcontroller which counts the number of pulses. The counter was reset 16 times per second. This frequency leads to the sampling rate of 16 Sa/s. Such as the area $A$ increases on inhale. With $A$ also the inductance $L$ increases. And with $L$ the frequency $f$ decreases. The dependency is non-linear and can be approximated with a third-degree polynomial. In calibration experiments, the factor of the quadratic term was negligible compared to the linear coefficient of the linearization polynomial. Therefore, linearization was even neglected for applications like respiration frequency or event detection. The advantage of inductance is the simplicity of the sensor itself and the best comfort as the wire does not require the amount of pretension required by resistance and capacitance sensors. The drawback is an elaborate measurement circuit based on an oscillator. Besides, the power consumption of this circuit was 11 mW per channel. Additionally, 10 mW were consumed by the microcontroller.

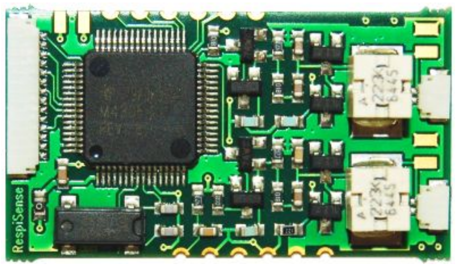


**Figure 2.5.:** A photograph of the RespiSENS module which was integrated into several respiratory monitoring systems including wearables. This module provided two channels of respiratory data including digital processing.

## 2.2.3. Signal Processing: from Sensor Signals to Vital Parameters

Signal processing can be seen as a chain of individual steps necessary to extract data of interest. The signal processing of a sensor system starts with at least one sensor which provides a sensor signal. Sensor signals require some preconditioning, usually amplification combined with filtering for frequency band limitation. Other sensors require more complex hardware or even actively controlled components e.g. (1) moving parts in gyroscopes, (2) driven feedback electrodes in ECG or (3) light sources in pulse oximeters. However, it became common to digitize a signal after the sensor signal was preconditioned. Numerous methods for digital signal processing exist. After digitalization, the sensor signal becomes a discrete time series processed further digitally. The aim of the signal processing is to extract the interesting part of the sensor signal. Pretending that the information of interest is contained in this sensor signal, there will be also some redundant information and mostly an over-proportionately large amount of irrelevant information. If it comes to efficiently transmitting data wirelessly, further efficient processing, and compact data storage, it is a good idea to reduce the data to the required minimum of interest. This extraction is done with specialized signal processing by removing redundancy and irrelevance from the signal until almost the interesting part remains. In biomedical engineering, the information of interest is often expressed as vital parameters.

Further, the vital parameters from heterogeneous sensors can be combined again into a new vital parameter. An example of how sensor parameters can be turned into vital signs by processing was created in figure 2.6. This figure illustrates a signal flow diagram that starts with five sensors on the left side: ECG, Photoplethysmograph (PPG), respiration sensor, accelerometer, and temperature sensor. The signals themselves are represented as hexagonal blocks. Signals can be either sensor time series or computed vital signs. Signal processing is represented in squared blocks. Each of these sensors delivers raw sensor signals. Body core temperature can be computed from the skin surface through a model. According to Anuar [17] and Atallah [33] body core temperature can be calculated by modeling from non-invasive surface thermometers. Further, 3-axis accelerometer data can be turned by an algorithm into posture information. It requires a second algorithm to generate activity recognition from accelerometer signals, e.g. office working, reading, writing, taking a rest, eating, cooking, dishwashing, walking, and running [6]. Going further, the respiratory sensor delivers a raw respiratory curve. This raw signal can be processed by an apnea detector algorithm [105]. Another algorithm could estimate breath states (inhale, rest and exhale). Further, with this breath states the breath rate can be calculated. A further algorithm would calculate the breath intensity from the same raw respiratory signal. From the raw ECG signal an algorithm extracts the cardiac beat events. With the cardiac beat events, further algorithms would compute HR and HRV. Additionally, arrhythmia-detection algorithms analyze the ECG signal shape and rhythm [212]. So far, sensor signals have been processed independently. More parameters can be gained by sensor data fusion. Therefore, the next parameter requires the raw data input of two heterogeneous sensors: ECG and respiratory sensor. Respiratory Sinus Arrhythmia (RSA) is the influence factor of the raw breath signal on the RR intervals of the ECG. According to [193], RSA magnitude is defined as the periodic cardiac interval change corresponding to the respiratory phases. In a co-authored paper [72] an application using RSA for arrhythmia detection was demonstrated. Thus, estimating RSA magnitude requires two sensor signals: respiratory sensor signal and cardiac data from ECG. A second example for data fusion is Pulse Transit Time (PTT) which requires input from a Photoplethysmography (PPG) and an ECG [149].

A third example for data fusion is the resting heart rate which requires input from an ECG and the activity detector. This example illustrates how sensor signals are transformed into vital signs and how useful it is to utilize signal flow diagrams.

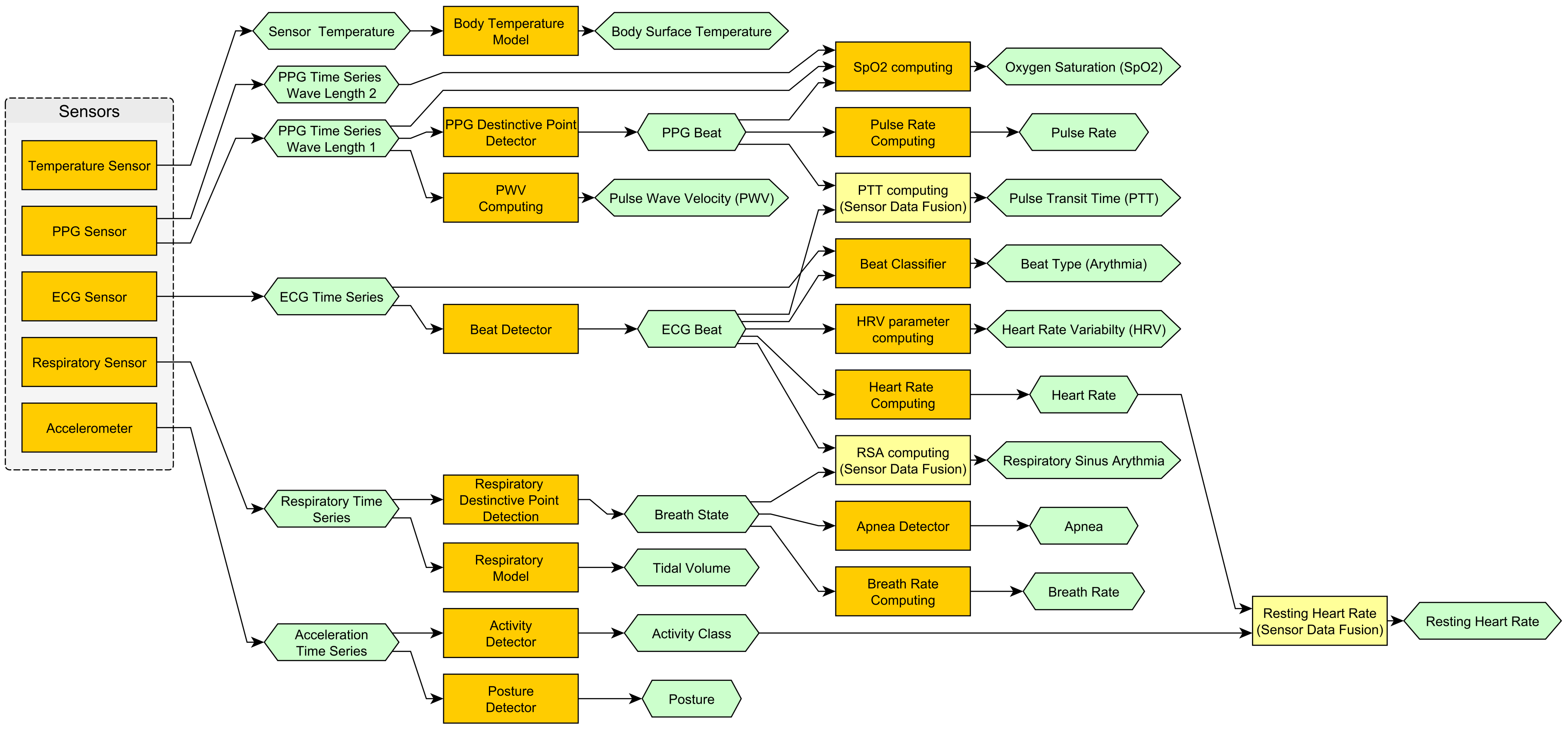


**Figure 2.6.:** A signal flow diagram with the sensors at the left side followed by their time series, algorithms, and computed data. Signal processing blocks are represented in squared blocks. Signals themselves are represented as hexagonal blocks. A detailed definition for this type of signal flow diagram is included in the appendix A.

### 2.2.4. Sensor Data Quality

The previous subsections focused on how sensor signals are obtained to be further processed into biomedical parameters. This process of sensing and signal processing always has weaknesses at every involved processing stage that may lead to reduced quality of the data provided. Two systems connected to one patient placed side by side will not provide perfectly equal data. Nissen et al. (2022) [3] has investigated this by comparing heart rate readings from two wearable systems. When it comes to differences between sensor systems, one major difference is the quality of the obtained values. Although the quality of sensor data is not the actual focus of this work, it is indirectly related to energy efficiency. Dhaimodker et al. [13] proposed a quality-driven energy optimization procedure for the IoT. Consequently, high data quality expectations will lead to a system design or configuration requiring more energy. However, the question of data quality is not easy to answer. The following summary is intended to reflect the state of the art.

Metrology is a classical and well-founded field of research addressing the quality of measurements. Metrology covers the definition of units of measurement, the realization of these units of measurement in practice, and the traceability of measurements made in practice to reference standards. This level of maturity could give the impression that metrology covers the entire topic of data quality. However, metrology does not cover all needs for the rapidly emerging problems of modern sensor systems and their complex data processing in data science. This gap has been filled in several attempts by data scientists, rather than meteorologists. Surveys on sensor data quality were published by Jagarlamudi et al. (2022) [2] and Teh et al. (2020) [15]. Jagarlamudi et al. used the term Quality of Context to compare the literature on sensor data quality. Teh et al. covered different literature sources on sensor data quality including an interpretation of the ideas originally punished by Kuka and Nicklas (2014) [93]. The original publication by Kuka and Nicklas addresses sensor data processing with quality semantics. In this work, multiple Quality Indicators were used to express the quality of sensor data. Similar ideas were previously published by Bisdikian et al. (2013) [108], and originally defined in their earlier work [141] and [140] back in 2009. Bisdikian et al. used the terms Quality of Information and term Value of Information. Quality of Information can be understood as a set of values that express a quality of data without an application context. Whereas, Value of Information expresses the value in a specific application context. Wu et al. [102] used the terms Quality of Data and Quality of Information for their proposed Cognitive Internet of Things. All reviewed publications use different wording for a similar solution: e.g. Quality of Information, Quality of Context, and Quality of Data. All researchers used so-called quality dimensions, which exist in different categories, naming, and meanings, although there are common overlaps here. The following four categories of quality dimensions are the most important and common in state of art: Accuracy, Timing, Completeness, and Consistency.

**Accuracy** has several meanings. The meaning depends on the domain of knowledge in which this expression is used. In the first place, the term Accuracy has two different meanings in metrology and data science. Moreover, if just focusing on metrology the term accuracy is defined differently depending on the standard. As if things were not complicated enough, the term accuracy is used several times with different meanings in metrology alone. Prenesti and Gosmaro [97] investigated this problem by reviewing multiple standards and guidelines defining the term Accuracy in the context of metrology. Consequently, one must decide on a metrology standard and declare how accuracy is defined there and used. In this work, accuracy will be used as a combination of precision and trueness, where precision defines the stochastic error and trueness a systematic error. Besides metrology, there is the domain of data science, where accuracy is defined completely differently. In data science, accuracy has been defined as a quotient of combinations of true positive, true negative, false positive, and false negative data points. However, accuracy can be seen as a category of all measures mentioned above defined in metrology and data science. Although it is not the only one, accuracy as a category is probably one that comes to mind when thinking about data quality.

**Timing** is the second category of data quality. Timing is an important category of data quality. The first dimension of timing is the so-called Timeliness. Timeliness is related to real-time conditions by defining when data arrives timely enough such as a processing unit that can consume the data. Timeliness is important because data can be useless if it is accurate but too late. Further, data usually comes with some time stamps. If time stamps are inaccurate data comparison with other data points in the past or data points of other sources leads to false interpretations. Multiple sources of errors can lead to wrong time stamps, e.g., errors in setting the time of the sensor, drift of the clock of a sensor, or damaged clock sources. Further, multiple time stamps can be given, e.g time stamp of data acquisition, timestamp of sending, and time stamp of arrival. All of them have different ascending places on a timeline. However, the meaning of the timestamp is not given because of a lack of semantics often. A further problem related to timing is the sampling rate. If the sampling rate is generated by precise clock sources, e.g., crystal oscillator, then there is a small error in the sampling rate which for most applications does not require special attention. Nevertheless, modern integrated circuits like accelerometers have integrated low-precision clock sources. The sampling rate can differ 10 % in the operating temperature range of the sensor. This dependency can cause several unwanted interpretation effects when assuming the sampling rate as a constant value. Another effect is jitter which is a variation between two sampling points in time. All these quality dimensions affect the data quality. Ignoring timing as a data quality indicator leads to false interpretations.

**Completeness** is the third category of data quality which declares if data is complete to achieve a certain task. Therefore, a data set may be required from several sources or a data sequence from a defined time interval. Data is incomplete if it is not obtainable, e.g., if the sensor missed an event and did not measure the value. Likewise, in a complex wireless data acquisition system, data can get lost during transportation to the consumer. The quality category Completeness is a set of quality dimensions that expresses how complete the data is according to a certain expectation.

**Consistency** is the last common category of data quality. Consistency compares the data from other similar data sources or in the context of other heterogeneous sources. For example, if multiple sensors are placed very close to each other, the sensors would be expected to provide similar values within some allowed boundaries. Bad consistency exists if one of the sensors would deliver values that differ from the other. A more general work published in this context was Scholl et al. [10] which allows verification of five capacitive sensor readings if the values are within a defined distribution. The distribution was previously determined and thus the values can be verified for consistency. The aim of this procedure is, only if the values are consistent, a machine learning algorithm that is prepared to process the values can provide reliable results. The consistency check allows machine learning to act more reliable concerning the provided data.

These four categories of data quality dimensions Accuracy, Timing, Completeness, and Consistency demonstrate how multifaceted data quality is. Even if the data quality is excellent in all categories data needs to be transformed into information. The heart rate is a good example here. Depending on age and sex the range for a normal pulse rate differs. Therefore, assumptions about the heart rate without considering the context may lead to completely wrong conclusions. Additionally, the heart rate will differ depending on the physical activity. A high heart rate during long-lasting physical exercise for a young man is completely harmless. On the contrary, the same heart rate history may be alarming during rest. Therefore, the literature distinguishes between different types of heart rates. Nanchen et al. [38] evaluated the resting heart rate. Danieli et al. [86] measured heart rate recovery after exercise. With the context, the heart rate parameter gets different meninges and can generate useful information. These were just two examples of how context plays a role. For this purpose, Park et al. [96] analyzed the role of wearable systems for big data applications. According to their conclusion, the gained data must be transformed properly within an information or knowledge processing ecosystem. With this, it depends on external information and the current context for data to be useful and relevant for the application. Analyzing data in context together with personal information, environmental conditions, past data, and data from a similar population is the key to generating reliable information from data.

## 2.3. Hardware for Signal Processing

In this section, hardware for signal processing will be discussed. Before diving deeper into the topic it should be clarified what signal processing is. In the IEEE signal processing magazine Chakravorty [34] (2018) was discussing the actual definition of a *signal*, finishing with the conclusion: "A signal, represented as a function of one or more variables, may be defined as an observable change in a quantifiable entity." Furthermore, Chakravorty also confirmed that a signal may or may not contain any useful information.

Then, *signal processing* can be defined as a set of operations modifying the properties of a signal. For signal processing, a variety of technical solutions exist. When talking about systems the boundaries of the system should be defined. In this work, a signal processing system – here in particular a sensor system – starts the signal inputs with the sensors and ends with the delivery of the processed signals. Further, regardless of the implementation, signal processing systems require energy. Several implementations exist. A distinction can be made between analog and digital signal processing at a very high level of abstraction.

### 2.3.1. Fundamental Properties of Analog Circuits

Physical sensors are fundamentally analog. Therefore, analog signal processing is required on a sensor system. In general, sensor signals are very small and require amplification. After amplification, some bandwidth limitation is required before digitizing a signal. Amplification and bandwidth limitations are required before a signal can be digitized. In times of digitization, everything becomes digital. This was already emphasized by Hosticka [196] in 1992. However, many signal processing tasks can still be realized analog. Many analog circuits are mixed circuits. Mixed circuits are analog and digital circuits combined. A good example of mixed circuits, is the ECG front-end AD8232. AD8232 contains analog amplifiers and filters controlled by digitally controlled analog switches. The actual signal processing path in the AD8232 is analog. Generally, power consumption increases with the required SNR and bandwidth [153]. Thus, there is a point of choice where signal processing is better realized analog than digital. For example, Thakor et al. [208] presented a QRS detector completely using standard components, most of them analog. The interesting question is: how does the power of analog circuits scale along with digital circuits to complete the same task regarding power consumption? Therefore, Enz and Vittoz [187] compared analog and digital. They calculated the absolute power per pole for circuits implementing filters. Assuming such ideal conditions, the power scales differently for analog and digital with SNR. Such as, according to Enz and Vittoz, for SNR below 25 dB analog is more energy efficient. Whereas, for SNR above 25 dB digital is more energy efficient. Similarly, Hosticka [204] derived more detailed models considering real-world conditions. Hosticka estimated this threshold between analog and digital for SNR at 44.5 dB. Besides fundamental energy limits, analog signal processing circuits have several practical properties. Analog signal processing suffers from aging, temperature dependency, and sensitivity to electromagnetic interference. This issue can be improved with high-quality components, in particular class 1 capacitors such as type C0G or NP0, or at least class 2, such as X7R or X5R. Moreover, digital circuits are easy to parametrize and configure during run-time. In comparison, analog circuits tend to be harder in adding and testing configuration features. Further, the variety of filter types available for digital designs is huge compared to analog filter designs. Consequently, analog signal processing has various challenges that must be adequately addressed. Regardless, analog signal processing is unavoidable in some cases.

## 2.3.2. Fundamental Properties of Digital Circuits

Complementary Metal–Oxide–Semiconductor (CMOS) technology is used for low-power integrated circuits. The average power consumption $P_{avg}$ in digital CMOS circuits can be approximated as the sum of three parts as refereed by Chandrakasan [189] and Sarpeshkar [137]: dynamic power dissipation $P_{dynamic}$, static power dissipation $P_{stat}$, and short-circuit power dissipation $P_{switching}$.

$$P_{avg} = P_{dynamic} + P_{stat} + P_{switching} \tag{2.8}$$

Strategies for reducing the average power dissipation in digital circuits target the minimization of all parameters. Therefore, a deeper insight into the three contributions is required. $P_{dynamic}$ is the power required for switching parasitic load capacitors $C_L$ at a frequency $f_c$ and the voltage of $V_{dd}$.

Since not all CMOS gates are active simultaneously, a transition probability factor $\alpha < 1$ is often added to the dynamic part of the equation.

$$P_{dynamic} = \alpha C_L V_{dd}^2 f_c \tag{2.9}$$

The second parameter is the static power $P_{stat}$ caused by constant current leakage $I_{leakage}$. This current is present as long as the sub-circuit is powered at $V_{dd}$.

$$P_{stat} = I_{static} V_{dd} \tag{2.10}$$

The third power leakage, $P_{switching}$, is caused by the short-circuit current which exists for a short finite rise and fall times of the input waveforms. It is a direct current path between $V_{dd}$ and ground which exist for a short time during switching together with $f_c$. The dependency on $f_c$ is nonlinear and requires a longer mathematical term. However, according to [137] for CMOS processes around 180 nm the switching power dissipation $P_{switching}$ is 5 % of the dynamic power dissipation $P_{dynamic}$. Thus, the focus will be on $P_{dynamic}$ and $P_{stat}$ for the following analysis. With this, the average power consumption $P_{avg}$ in digital CMOS circuits can be expressed with a dynamic and static part.

$$P_{avg} = \alpha C_L V_{dd}^2 f_c + I_{leakage} V_{dd} \tag{2.11}$$

Some parameters can be optimized on a CMOS technology level, others on an application level. $I_{leakage}$ and $C_L$, both depend on the CMOS technology. With descending process size $C_L$ falls while $I_{leakage}$ rises. The load capacitance $C_L$ and $I_{leakage}$ are determined by the choice of a product with a certain CMOS technology. Unlike, $f_c$ can be controlled on the application level to a technically possible minimum. A reduction by half of $f_c$ leads reduction by half of $P_{dynamic}$. Assuming a computing task requires a fixed number of clock cycles $N_{cy}$ for completion, the computing time can be given with $t_p$.

$$t_p = \frac{N_{cy}}{f_c} \tag{2.12}$$

The above power optimization strategy is based on automated clock scaling of $f_c$. However, clock scaling faces several problems, which require appropriate algorithms to meet real-time conditions [125]. Additionally, clock scaling requires a reserve factor to meet dynamic workloads.

Another approach is to set the clock $f_c$ to a constant frequency and to utilize clock and power gating. Therefore, the equation 2.11 was multiplied by $t_p$, which gives the energy portion per gate $E_p$ required for completing a computing task with $N_{cy}$ clock cycles.

$$E_p = \alpha C_L V_{dd}^2 N_{cy} + I_{leakage} V_{dd} \frac{N_{cy}}{f_c} \tag{2.13}$$

With this, $f_c$ should be maximized to the limit of the specific digital circuit, to minimize $E_p$. This example assumes instantaneous transitions for clock and power gating, which are not considered in equation 2.13. Further, the parameter $V_{dd}$ can be controlled on the application level. $V_{dd}$ has a quadratic relationship on dynamic power dissipation. With this, $V_{dd}$ may greatly impact power consumption. When minimizing $V_{dd}$ the maximum clock $f_c$ required for stable operation descends. Consequently, going down with $V_{dd}$, limits $f_c$, which causes longer processing periods, and thus higher static dissipation. Finding the best solution for the parameters is a typical minimization problem. This problem was theoretically addressed in Ha et al. [90] where the minimum energy point was illustrated as a function of $V_{dd}$. For practical usage, $I_{leakage}$ and $C_L$ will not be available in a datasheet. Instead, the power dissipation in low-power mode and power dissipation per cycle are expressed in the datasheet, giving an idea about $I_{leakage}$ and $C_L$. To sum up, with proper power gating and clock gating the equation 2.13 describes the energy portion per gate for one computation of a computing task. However, a system may have always-on components at different clocks. These always-on components can be described with the previously used model in equation 2.11.

### 2.3.3. Wearable Computing Hardware

In a modern wearable, the signal is being digitized at some point and hardware will be required for further digital signal processing. There are several options to choose from to perform digital signal processing.

- Application-Specific Integrated Circuit (ASIC), Application-Specific Standard Part (ASSP), and SoC
- Sensor System on a Chip
- Microcontroller
- Digital Signal Processor (DSP)
- Field-Programmable Gate Array (FPGA)

#### ASIC, ASSP, and SoC

ASICs are custom ICs typically designed for one company. ASSPs are application-specific parts designed and implemented in the same way as ASICs but intended to be used by multiple system design houses. Min et al. [113] compares implantable cardiac pacemakers with a total amount of power of 19 µW including advanced digital ECG processing. Thus, ASICs, ASSPs and SoC would be the best choice for energy-efficient biomedical systems. Unfortunately, an ASIC design is the most expensive way of designing the signal chain and is only possible in large-quantity applications. SoC designed and implemented in the same way as ASSPs combining a system on one silicon single die, while additionally containing multiple computing cores. A representative SoC for biomedical sensor systems is the Infineon MD8710 which contains analog front-ends for multiple sensors, an application microcontroller and an additional dedicated communication microcontroller, together with radio-frequency transceiver. In [85] a minimum battery current of 20 mA has been achieved during operation and 50 µA during shutdown. Van Helleputte et al. claimed in [78] using a specialized SoC for bio-impedance, 3-channel ECG, motion artifact reduction, and integrated Advanced RISC Machines Ltd. (ARM) Cortex-M0 processor with hardware accelerators for signal processing running at total 345 µW. This SoC is available in its third version.

### Sensor System on a Chip

Modern Micro-Electro-Mechanical Systems (MEMS) contain a whole sensor system on a single chip. MEMS with analog output already have rudimentary signal conditioning, e.g. temperature compensation and amplification, which makes them a sensor system. Advanced MEMS like the LSM6DSL [36] or the BMX055 [116] contain additionally an anti-aliasing filter, followed by an Analog-to-Digital Converter (ADC) for digitization. The digital signal is passed through a configurable signal processing chain: sample rate conversion, multiple digital filters, and several adjustable thresholds for rudimentary event detection. A more advanced MEMS, like the LSM6DSOX [28] extends the functionality of the LSM6DSL by on-chip machine learning with basic feature extraction followed by configurable state machines and decision trees. An equal task would require more energy to compute the same results using a microcontroller. Therefore, Gallizio claims in [24] for computing an activity recognition algorithm that detects (staying, walking, jogging) the LSM6DSOX requires additionally 7 µA while the microcontroller STM32L152RE running at 32 MHz which is an ARM Cortex-M3 requires additional 240 µA for continuous operation. However, computing is limited to certain kinds of features and certain kinds of machine learning algorithms.

### Microcontrollers

Microcontrollers consist of a processor, together with memory and peripherals. Microcontrollers are mass products embedded countless in our entire environment. A variety of so-called low-power microcontrollers were available as this work was started. Towards the end, numerous new microcontrollers came up and some more were announced promising new groundbreaking features for power savings. During this time an overview of potentially suitable microcontrollers for low-power applications was brought together in a mind map in figure 2.7. Microcontrollers are preferred candidates for implementing both, signal processing and overall system control of a wearable sensor system. Therefore, microcontrollers were investigated further in chapter 5.

### DSP

DSPs are specialized processors for effectively executing signal processing. Extended instruction sets, specialized addressing techniques, and parallel execution are just a few general techniques for the accelerated execution of signal processing algorithms.

In general, DSPs are mostly not designed to be power efficient. Unlike the TMS320C55-Series: An example was the project KonMeVit [144] where the DSP of type TMS320C5509 was used in a wearable biomedical sensor system. The DSP was used for the acceleration of biomedical signal processing. All other computing and application tasks were executed on a separate low-power microcontroller of Texas Instruments MSP430. The microcontroller was controlling the sensor system while serving the DSP with data and reviving the computed results.

In the multiprocessor system, MSP430 was the main controller with low-power capabilities while the TMS320C5509 was activated on demand taking the role of the DSP. Today, this functionality can be found on a single standard microcontroller. With rising requirements for signal processing DSP functionality became popular on microcontrollers.

Eyre and Bier [180] gave a brief introduction to DSPs and explain how DSP became the mainstream. Giving an example, ARM architecture, in particular the ARM Cortex-M4 series supports e.g. multiply and accumulate in one operation, Single Instruction, Multiple Data (SIMD), and saturation arithmetic, which are key DSP concepts for acceleration of signal processing algorithms [64]. A specialized DSP still beats the computing performance with additional features. Still, the boundaries between DSP and microcontroller became blurred.

#### FPGA

With FPGAs highly parallel signal processing can be realized. For a long time, FPGAs were optimized for high data throughput and fast response times, while low-power optimization played a secondary role. Qi et al. [53] compared the State of the Art with a custom ultra-low power FPGA with a supply voltage down to sub-threshold and by using clock-gating, which makes them interesting for wearable computing. Donnelly et al. [46] mentioned the system developed in this thesis with the proposal of using FPGA for digital interpolation filters.

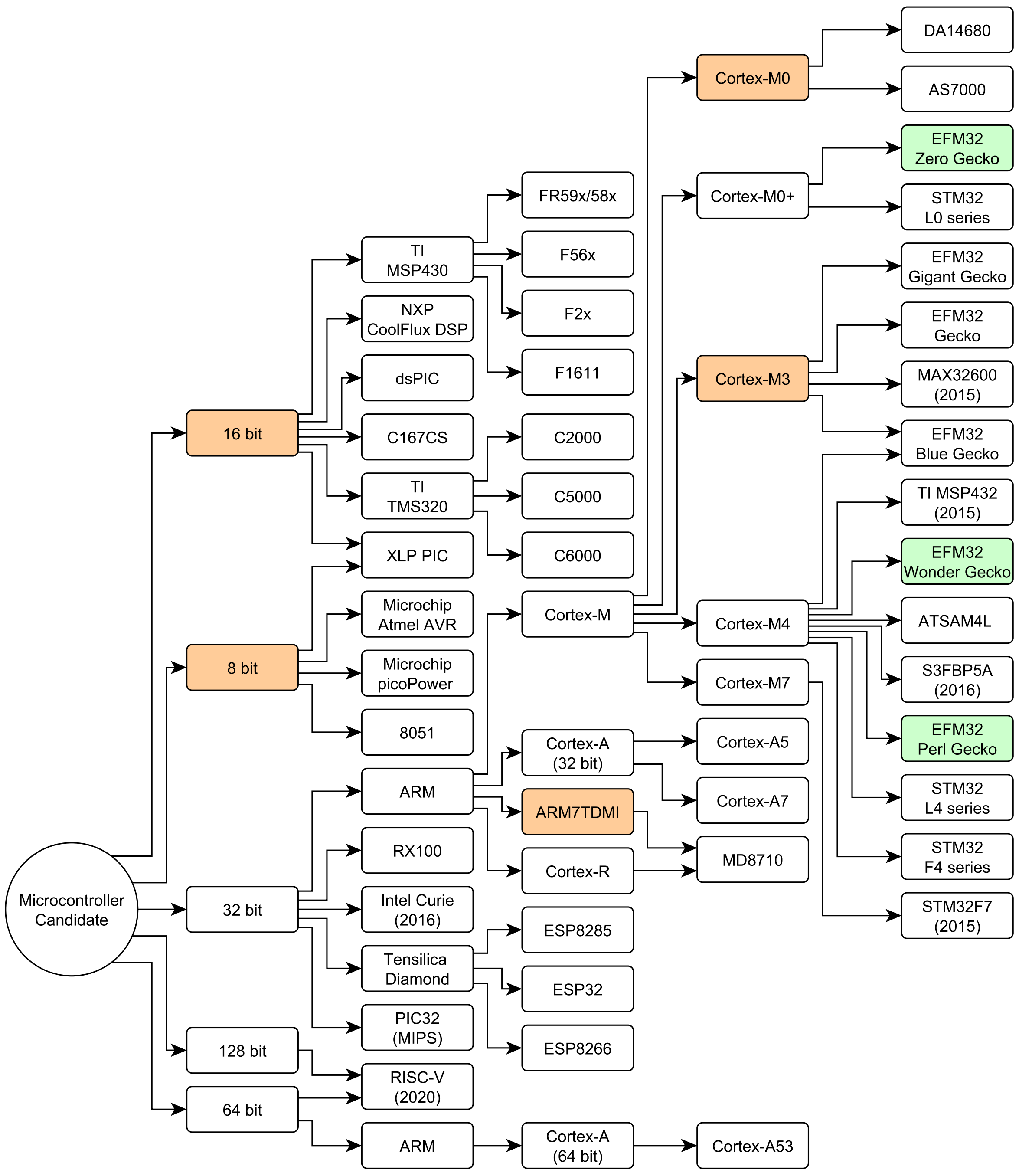


**Figure 2.7.:** Microcontrollers, which are considered in this thesis, were structured in this mind map. The first level of the tree branches into the native register size (8, 16 or 32 bit) of a microcontroller. From the second level on, the microcontroller families are classified in more and more detail, up to a certain microcontroller type. The favorite microcontroller types, such as the Energy-Friendly, 32-bit Microcontroller by Silicon Labs Inc. (EFM32) Wonder Gecko, have been used in the ULPSEK.

## 2.4. Wearable Software

The term, Wearable Software, was defined in this thesis as software that meets wearable computing requirements, mainly addressing restricted computing and memory resources. Wearable Software was split into three areas of special interest:

1. Data Rate Reduction Techniques, a key factor for low-power systems.
2. Real Time Operating System, which is a key factor for microcontroller-based low-power designs.
3. Wearable Algorithms, addressing several levels of low-power optimization techniques.

### 2.4.1. Data Rate Reduction Techniques

The amount of raw sensor data generated by a wearable sensor is low compared to other applications, e.g. image processing and audio processing. Neverless, an ECG signal at 1000 Sa/s and 16 bit resolution leads into a data production of 173 MByte per day. Therefore, to reduce the energy consumption for transmission requires reducing the data rate. Data reduction techniques can be classified in four general types of data reduction, often combined to gain maximum compression rates.

- Lossless compression
- Lossy compression
- Compressed sensing
- Parameter extraction

#### Lossless compression

Lossless compression, also known as entropy encoding, is used to maximize the *information entropy* of a data set without any knowledge about the data source's properties and the properties of its data. One drawback is that the compressed data can only be interpreted and processed further after decompression. Therefore, an appropriate decompression algorithm is required. An advantage is that entropy decompression allows perfect reconstruction of the original data without any loss of information. A popular algorithm is the Lampel-Ziv-Welch (LZW) compression algorithm published in 1978 [213], which has been used in the Graphics Interchange Format (GIF) for compressing bitmap graphics and many other file archives for compressing text and executable files. However, this method's compression ratio depends on the data's information entropy. The higher the information entropy, the lower the compression ratio. Unfortunately, medical sensor data is noisy and thus the information entropy is high, leading to compression ratios below 2:1. In a study from 2021, Ketshabetswe et al. [4] investigated data compression algorithms for wireless sensor networks. They achieved a power saving of 67.8 % to 73.2 %. In this work not only lossless but also lossy compression was investigated.

### Lossy compression

Lossy compression, as the name implies, removes unnecessary information to achieve a higher compression ratio as it can be done by lossless compression. Lossy compression algorithms are always application specific. The compression algorithm can only be applied to data where the properties of the source and the wanted information are known. Decompression returns the data where the wanted properties are conserved within a specified difference to the origin. Popular examples of lossy compressions are the audio coding standard MPEG Audio Layer III, better known by its file extension MP3 [183]. Specialized lossy compression algorithms exist for medical applications. Therefore, for lossy compression of ECG signals Cetin et al. [175] and Priyanka et al. [114] are claiming compression rates between 2:1 and 23:1. Finally, the compression ratio is an adjustable parameter of the compression algorithm. A general rule is the higher the compression ratio, the lower the data quality after decompression.

### Compressed sensing

Compressed sensing is comparable to lossy compression. The key difference is that instead of sampling and then compressing the data, compression becomes an integral part of the data acquisition process based on non-equidistant sampling. The motivation for compressed sensing is that the acquired data is compressed by the acquisition technique, saving storage for raw data and energy for sampling. Mamaghaninan et al. [131] compared the energy amount for raw data transmission, lossy compression using wavelets, and compressed sensing for ECG transmission using wearable sensors. They achieved 9.7% less power consumption using compressed sensing, while the conventional lossy compression method required more energy as sending the uncompressed data. However, the major disadvantage of compressed sensing is the lack of interpretable data on the device itself. This fact is irrelevant if data is not wanted to be interpreted at the sensor itself, but in the case of a smart sensor data is interpreted at the sensor itself. Thus, a system designer should consider using conventional sampling instead of compressed sensing. The data does not exist in an interpretable representation at any time on the system. Further, the decompression algorithm requires computing capacities beyond the energy required for sampling the data the conventional (Shannon) way. Thus, this method is perfect for streaming applications, but not appropriate when data needs to be interpreted on the sensor node itself.

### Parameter extraction

Parameter extraction may be declared as a subcategory of lossy compression. The difference to lossy compression is, decompression is not necessary as parameters can be interpreted as they are. Parameters e.g. frequency, amplitude, and Root Mean Square (RMS) are extracted form data. Further, parameter extraction provides the highest data reduction. Five seconds of ECG at 1 kS/s require 10 kByte of memory which can be reduced to the heart rate, allocating only 1 Byte of memory for storage or transmission. Parameter extraction is the most powerful technique for data reduction, as long as the parameters are calculated correctly.

### 2.4.2. Real Time Operating System

Real-Time Operating System (RTOS) are made for devices with constrained resources, low volatile memory and low processing capabilities. When setting up a new embedded project, a mission-critical decision is using an RTOS or building a so-called *bare metal* application. A *bare metal* application is useful for small sensors with less functionality. With increasing features, handling a *bare metal* application becomes more and more difficult. An RTOS is a software library for scheduling the execution of parallel pending tasks. Besides this, an RTOS provides standard mechanisms for synchronizing tasks execution and exchanging data between them. The unique feature of an RTOS is the capability of giving a task control back guaranteed at a certain point in time, which makes it capable of real-time applications.

While using an RTOS on a microcontroller has been very common in general, using an RTOS for low-power applications has become popular recently. For example, the so-called tickless idle mode was introduced by Martin Tverdal [138] in 2010. This work was an important contribution to FreeRTOS[1], which is a popular, suitable for professional usage and a free RTOS. Before this contribution using an RTOS for low-power applications was a waste of energy. When not using tickless idle mode the microcontroller will wake up from low-power mode every tick. The tick frequency is set 1000 times per second in FreeRTOS by default. Thus, in combination with a low-power mode, the microcontroller would wake up 1000 times per second, mostly unnecessarily. The solution for this waste of energy is tickless idle mode: With tickless idle mode, FreeRTOS calculates the duration for the next task to execute instead of checking each tick for pending tasks. The advantage of the tickless mode can be seen in figure 2.8.

However, a low-power microcontroller often serves for simple repeating tasks. With increasing requirements for microcontrollers, the demand for an RTOS grows. When using RTOS for a low-power application, tickless idle mode is a must-have. FreeRTOS and meanwhile other RTOS support this feature.

### 2.4.3. Wearable Algorithms

In general, an algorithm is a systematic procedure that produces – in a finite number of steps – the answer to a question or the solution to a problem [135]. Wearable devices have highly limited resources, in computing performance, memory, and energy. An algorithm must fit the constraints providing an energy-efficient solution for being wearable. Estimations of algorithmic energy efficiency require consideration of the underlying hardware and the surrounding software. Therefore, according to Chen et al. [84], wearable algorithms are a multi-disciplinary problem. Those considerations were done in different levels of detail. However, different approaches exist to rate algorithms in terms of computing efficiency. The following literature review attempts to categorize the algorithm efficiency in different detail levels concerning power consumption.

**Detail Level 1:** Estimating the number of mathematical operations:

- Elgendi et al. [88] reviewed QRS detection methodologies for portable, wearable, battery-operated, and wireless ECG systems. They declared three major criteria

[1]FreeRTOS: Real-time operating system for microcontrollers, home page at http://www.freertos.org

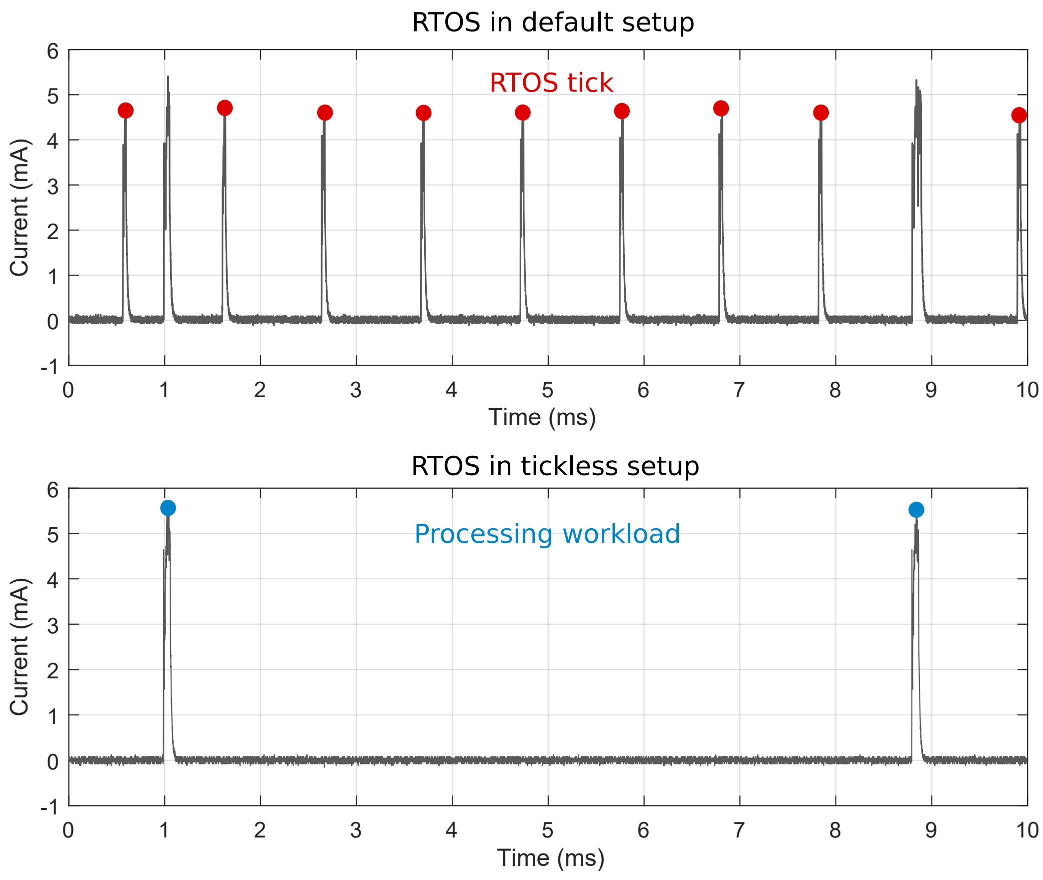


**Figure 2.8.:** This figure compares two microcontroller current recordings. The first plot shows a waste of energy by regularly waking the microcontroller from sleep to handle the RTOS system tick. This wake-up is unnecessary because, in most cases, the RTOS tick routine detects nothing to process, consequently sending the microcontroller back to sleep again. The second plot shows the tick suppression after an improved setup of the RTOS for tickless operation.

for QRS detector algorithms: 1) robustness to noise, 2) parameter choice, and 3) numerical efficiency.

- Bolz [106] reviewed several QRS detectors for computing efficiency.
- Ring et al. [122] evaluated machine learning algorithms regarding complexity analysis for the design of embedded classification systems.
- According to Moyer [174]: "Algorithmic-level power reduction techniques focus on minimizing the number of operations weighted by the cost of those operations."

**Detail Level 2:** Estimating the number of cycles for a specific processor:

- Jensen et al. [61] addressed the accuracy-cost conflict for popular classification algorithms on embedded systems.
- Johnston et al. [160] compared signal processing algorithms for a microcontroller-based wearable pulse oximeter.

**Detail Level 3:** Estimating energy consumption by simulating hardware:

- Sieh et al. [29] combined automated measurement-based cost modeling with energy-consumption analysis. Several algorithms were evaluated by providing a number of cycles. The evaluated hardware was a microcontroller with a core of type ARM Cortex-M0+.

- Zou et al. [115] presented a QRS complex detection algorithm based on wavelet transform. The energy consumption for the algorithm was simulated using a SystemC-based simulation framework for energy-efficiency evaluation [128].

- Wägemann et al. [79] proposed a method for worst-case energy consumption if detailed knowledge about the energy costs of instructions on the target platform is available.

- Sieh et al. [29] combined automated measurement-based cost modeling with energy-consumption analysis.

- Kugler et al. [110] simulated Shimmer sensor nodes for on-node signal processing algorithms, in particular algorithms for activity and fall detection.

- Eriksson et al. [142] proposed a power profiling of sensor nets with a simulator.

**Detail Level 4:** Executing and measuring algorithms on specific hardware at the application level.

- Chen et al. [84] compares literature for wearable algorithms ruining on low-power hardware.

- Chen et al. [78] The algorithm was verified using the MIT/BIH Arrhythmia Database.

- Morshedlou et al. [37] presented an implemented of a QRS detector algorithm completely in mixed-signal hardware claiming 71 nW for average power consummation. The algorithm was verified using the MIT/BIH Arrhythmia Database.

- Hamilton [171] published a modified open-source implementation of the famous Pan-Tompkins algorithm [205] for QRS complex detection. Hamilton added an optional beat classifier and compiled the code for the microcontroller PIC16F877. The algorithm was verified using the MIT/BIH Arrhythmia Database. However, power measures are not mentioned in this paper.

- Wägemann et al. [41] proposed SysWCEC, an approach that addresses these problems by enabling static worst-case energy consumption analysis for entire real-time systems, including internal as well as external devices. Anliker et al. [164] presented AMON: A Wearable Multiparameter Medical Monitoring and Alert System. The proposed device measured ECG, blood pressure, and Oxygen Saturation ($SpO_2$).

- Chen et al. [84] introduced whole-system worst-case energy-consumption analysis for energy-constrained real-time systems.

- Hönig et al. [91] presented Proactive Energy-aware Evaluation Kit (PEEK), a system approach to optimize software for energy efficiency. PEEK automates energy measurement tasks and suggests program-code improvements. In this paper, the authors also compare the influence of compilers on different algorithms.

The four categories above show how different algorithms were rated with respect to power consumption in literature. The first detail level was a power reduction on the algorithmic level, assuming that less number of operations contribute to less computing and thus less energy cost. However, beyond mathematical operations, multiple additional operations exist on a microcomputer, e.g. memory movement operations. Those operations are more cost-intensive than the computing itself. Thus the second detail level investigates the operations of a specific processor, including memory transport. The methods in the third level were estimating energy consumption by simulating hardware. Finally, the fourth detail level is measuring on specific hardware, which is the most precise rating regarding energy efficiency. However, with a rising detail level, the general algorithm turns into a specialized machine code. All detail levels have their advantages and disadvantages. Investigation on all levels is required in science. The first level allows a general and easy-to-apply metric, giving a general comparison for algorithms. However, hardware can execute mathematical operations very differently, depending on the data type, available DSP commands, and available SIMD support. Further, many operations are not mathematical and are costly. For example, the Xeon X5670 microprocessor requires more energy for data transport compared to actual computing. According to Molka et al. [136], the processor power efficiency was determined at 156 pJ/bit for data transport between Random Access Memory (RAM) and the Central Processing Unit (CPU), while the actual computing operations were determined in the range between 3.5 and 7.4 pJ/bit. Another example was given by Trevor Martin [112]. Martin gives the example where a 16 bit by 16 bit multiplication takes 48 clockcycles on a 8 bit processor, 8 clockcycles on a 16 bit processor and 1 clockcycle on an ARM Cortex-M. Consequently, the efficiency of algorithms depends essentially on the underlying hardware. Therefore, a high detail level is more target-oriented in terms of energy optimization. In conclusion, a favorite detail level does not exist. All of them have their purpose. According to Frantz [134], a signal processing system should address low-power consumption at all levels of abstraction. It should apply state-of-the-art power-efficient silicon technology. It should utilize the most advanced power management techniques at RTOS-level. The application should be trimmed to low-power consumption. Algorithmic transformations at the application level can be very power efficient. The exploration of the potential of the hardware and software leads to a low-power system. Therefore, designing a low-power signal processing system, like a biosensing wearable, is a combined hardware and software problem.

Chapter 3

# Developing the Ultra-Low-Power Sensor Evaluation Kit (ULPSEK)

ULPSEK is a sensor system designed to address the research questions in this work. The hardware and software of this platform are fully accessible, allowing insights into the possibilities of optimizing low-power sensor systems. Therefore, this platform has central importance for this work, and its iterative development is one of the main contributions of this thesis. This chapter describes the purpose, features and all components of ULPSEK.

## 3.1. Motivation for Creating ULPSEK

ULPSEK was created to fill a gap. Many systems were mentioned in the literature, but all of them were barely suitable for investigating low-power consumption strategies, as has been aimed in this thesis. The aim of this thesis was to find a model for signal processing regarding the energy consumption of a portable sensor system. Thus, the objective of this work required access to the hardware and software of an entire sensor system. In particular, all signal processing parts needed to be accessible, individually activated and deactivated, and modified. Unfortunately, the state-of-the-art hardware and software were restricted to certain circumferences, which would handicap this research work. Two categories of potential sensor systems for low-power investigations were available: close source products and open platforms.

**Wearable close source products:** The number of low-power vital monitoring products is increasing. Unfortunately, all of them were closed in hardware and software. With scientific interest, results gained with closed-source hardware are hard to analyze and almost impossible for others to reproduce.

**Open platforms:** Ciabattoni et al. [57] investigated open and modular hardware nodes for wireless sensor and body area networks. While the investigated DIY platforms were very open in hardware and software, the low-power design was not a high-ranked requirement. They are primarily made with respect for ease of use, which of course, matches the requirements of an open-source community perfectly.

Thus, neither the closed source hardware nor the open Do It Yourself (DIY) platforms could fulfill the requirements for the investigation planned in this work. Consequently, a way to get around this lack was to create a new system design: accessible in hardware and software without restrictions on disclosure of any parts of the sensor system. Thus, the idea of a ULPSEK was born.

## 3.2. ULPSEK Components: a Brief Overview

ULPSEK was developed for experimental purposes as part of this thesis to investigate the effects of a design decision on the power consumption of wearable sensor systems (figure 3.1). The hardware and software of ULPSEK were optimized for battery runtime. All modular system components take a dedicated place on the ULPSEK motherboard and will be introduced hereafter in the following order.

- Microcontroller module
- Sensors modules
  - PPG sensor module
  - Respiration sensor module
  - ECG sensor module
  - Accelerometer module
  - Temperature sensor module
- Radio module
- Voltage regulator modules (power distribution network)
- Battery

Further, there are several system components fixed at the ULPSEK motherboard:

- Battery charger
- Power switches (p-channel MOSFETs)
- Power measurement terminals
- Microcontroller debugger connector
- USB port
- Power measurement IC of type INA219 [71] for self monitoring purposes.
- User interface
  - Multifunction button (on, off, mode change)
  - Display
  - Multicolor LED
  - Speaker

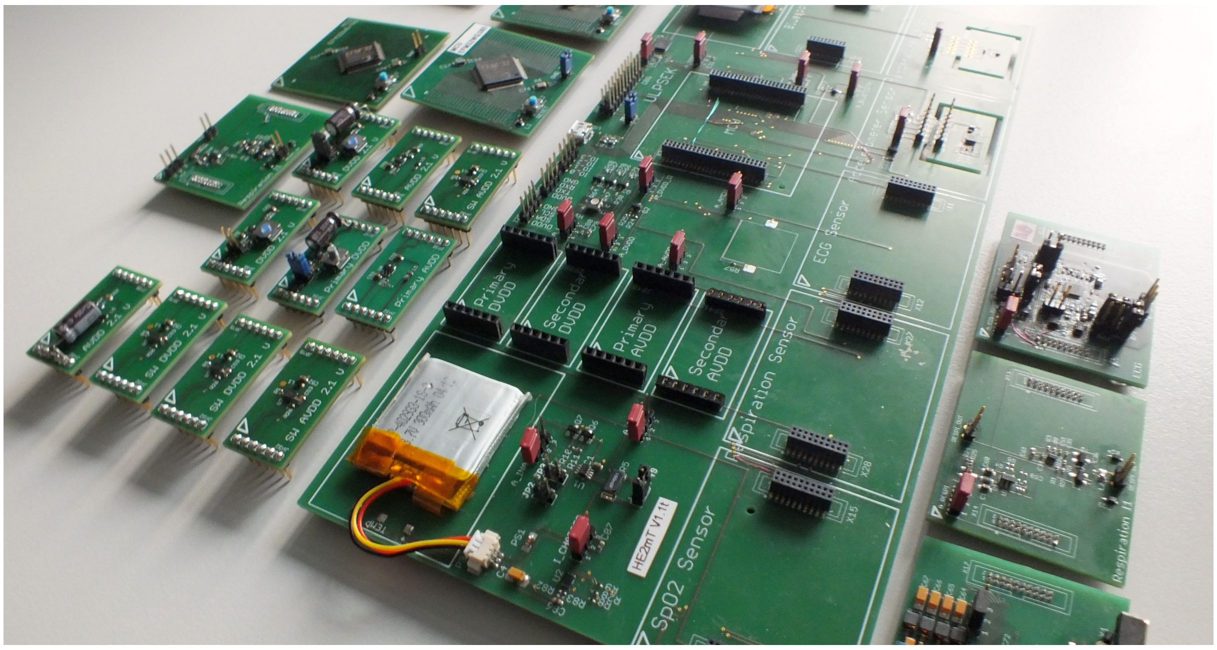


**Figure 3.1.:** A photograph of ULPSEK, which is a modular development kit for ultra-low-power wearable sensor systems. This photograph shows ULPSEK with unattached extension modules surrounding the main motherboard.

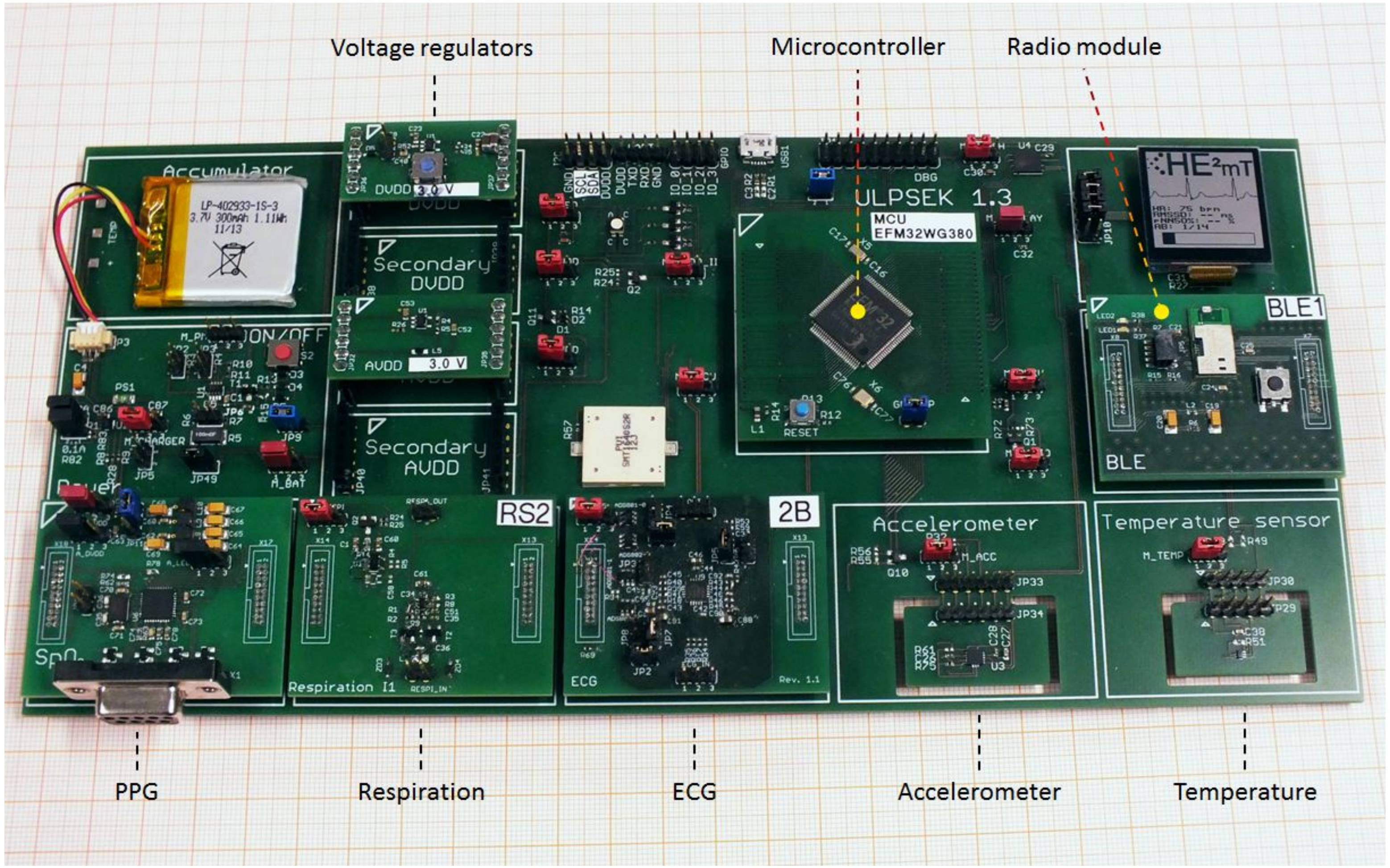


**Figure 3.2.:** ULPSEK – a modular development kit for the investigation of ultra-low-power wearable sensor systems. The dimensions of ULPSEK are 260 mm x 125 mm and 10.2 in x 4.9 in, respectively. Sensor modules are located at the bottom row: photoplethysmography (PPG), respiration, electrocardiography (ECG), acceleration, and temperature. Further parts like the microcontroller, power supply modules and radio module are exchangeable. The red jumpers mark the terminals for power measurement.

## 3.3. Microcontroller Module

Fig. 3.3 shows two different microcontroller modules. One such module can be placed almost in the center of the motherboard. This module contains the actual microcontroller and additional components such as capacitors, resistors and two crystals.

The microcontroller is connected to almost all components on the board using analog and digital interfaces. On the one hand, all sensor modules deliver their data over individual interfaces to the microcontroller. On the other hand, the microcontroller controls the configuration and, thus the behavior of all peripherals, including sensors, display, radio, and the power distribution network.

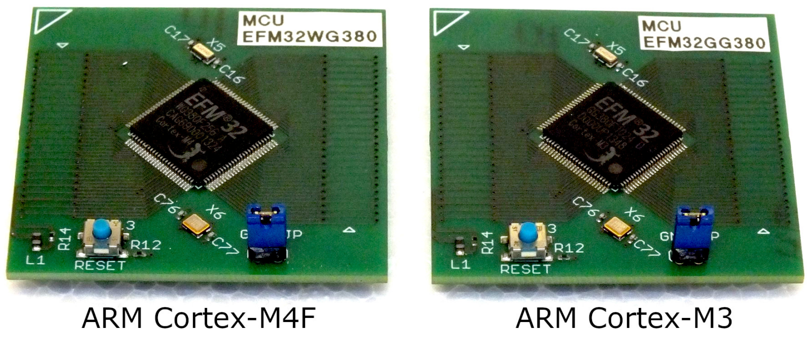


**Figure 3.3.:** Two similar-looking modules, each carrying differently energy-efficient microcontrollers. To the left is an ARM Cortex-M4F (EFM32WG380) and to the right is an ARM Cortex-M3 (EFM32GG380).

## 3.4. Sensor Modules

The ULPSEK base board hosts five sensor modules. The sensor modules are seen at the bottom in Figure 3.2, having their dedicated places from left to right:

1. PPG module
2. Respiration module
3. ECG module
4. Accelerometer module
5. Temperature module

Each sensor module has a corresponding low-power sensor driver, which is a piece of software written in the programming language C dedicated to the microcontroller. These software drivers allow to control and read sensor values from the modules. The five sensor modules (hardware) and their drivers (software) will be described hereafter.

In general, interfacing hardware should be done via Serial Peripheral Interface (SPI) rather than Inter-Integrated Circuit Bus ($I^2C$) if there is a choice. Mikhaylov and Tervonen [119] found $I^2C$ interface required more than two times more energy compared to SPI.

### 3.4.1. Photoplethysmography Module

The heart of the PPG module is the integrated front-end AFE4490 released by Texas Instruments (datasheet [80]). The AFE4490 hardware is capable of both: photoplethysmography and pulse oximetry. The AFE4490 sensor front end is configurable for a variety of optical probes. Therefore, the PPG module provides connectors for external probes. Common pulse oximetry probes use a D-sub 9-pin connector for the fingertip or ear lobe. This connector is visible at the bottom side in figure 3.4. Each probe connected to this port requires a special configuration at least for maximum LED current and photocurrent amplification.

On the software side, a new driver was written for the AFE4490 front-end specially designed with a focus on low-power for ULPSEK. The purpose of the driver was to set the LED timing parameters, LED currents, photocurrent amplification factors, and read the sensor data. The configuration of the analog front-end and the reading of the sensor data was done via SPI. Therefore, the driver allocates one SPI interface. This functionality is provided to the application programmer without the need to understand the hardware registers of the AFE4490.

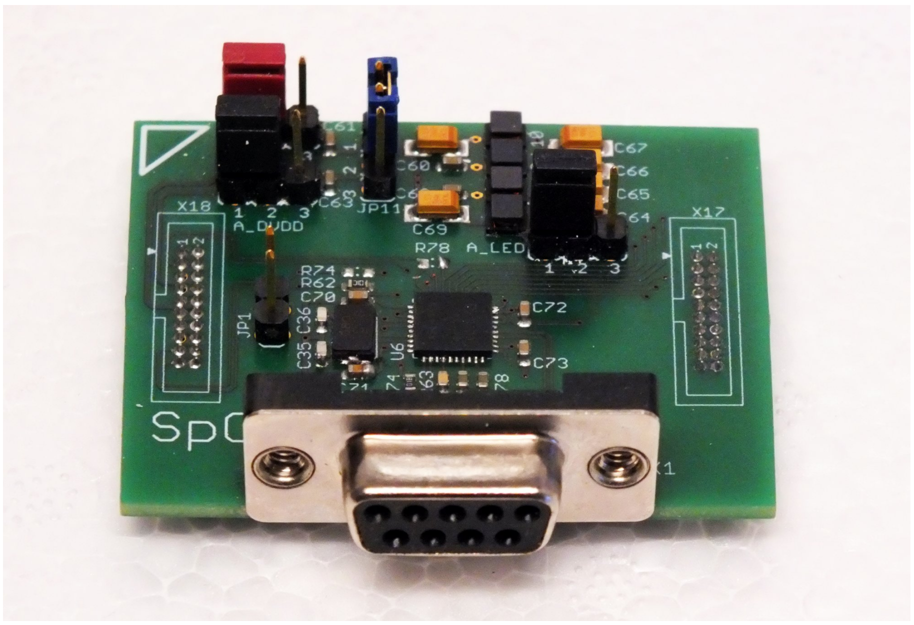


**Figure 3.4.:** ULPSEK sensor module for PPG acquisition. A pulse oximetry probe can be attached to the D-SUB-9 connector. The module has its dedicated connector at the ULPSEK board.

The driver configures the AFE4490 internal registers, which determine the behavior of the hardware. Figure 3.6 is showing an abstract representation of the receiver block, the LED driver and the SPI which interfaces to the microcontroller. In the center of this figure, there is a timing controller which requires setting a pattern of when to power on the probe LEDs and sample sensor data. The AFE4490 controls up to two LEDs using a digitally adjustable current source. An H-bridge reverses the current flow to drive a second LED by using only two wires.

The photodiode delivers a current proportional to the received light. This current is converted by the trans-impedance amplifier into a proportional voltage. Here, gain and

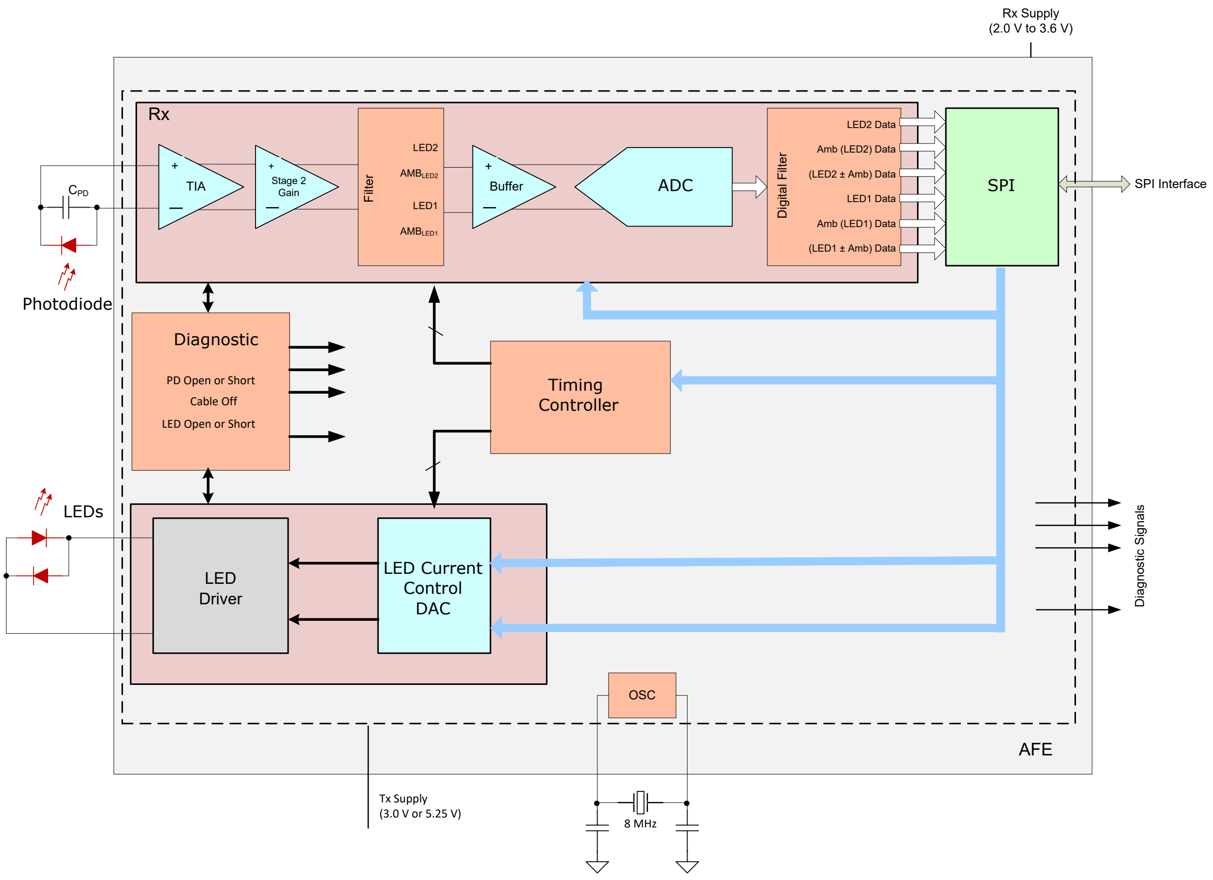


**Figure 3.5.:** Functional diagram of the integrated circuit AFE4490 derived from the AFE4490 datasheet [80]. The original figure was modified. The diagnostic module was removed and minor typographic modifications were made.

bandwidth can be adjusted digitally. This signal is forwarded to a second amplification stage. For environments with intensive static light, an analog ambient light subtraction offset can be activated for the second stage. The second amplification stage is followed by an analog demultiplexer. The demultiplexer forks the path into four sample-and-hold stages, which additionally act as a low-pass filter for suppressing aliasing. There are four channels in total: Two channels for each LED wavelength and two additional channels for the ambient light sampling. The filter group is followed by a multiplexer which joins the path into one buffer, followed by one 22-bit sigma-delta ADC. After the ADC, a basic digital signal processing can be set up optionally, e.g. averaging of incoming samples and digital ambient light subtraction. Thereafter, depending on the settings, up to six sampled curves can be picked up via SPI by the microcontroller:

1. Red LED channel
2. Ambient light for red channel
3. Red LED channel with subtracted ambient light
4. Infrared LED channel
5. Ambient light for infrared channel

6. Infrared LED channel with subtracted ambient light

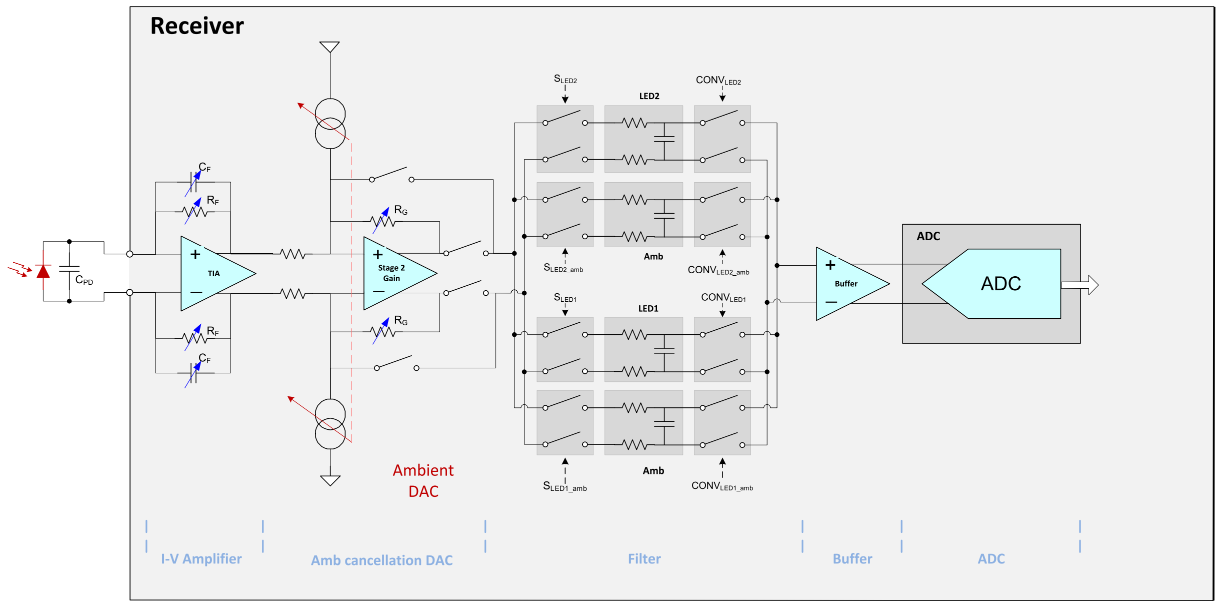


**Figure 3.6.:** This is a functional diagram of the receiver signal path in the AFE4490. This image was taken originally from the AFE4490 datasheet [80] and modified.

The power consumption of the PPG sensor module depends most of all on the sampling rate, the LED on-time, and the LED current. While the sampling rate was fixed, the LED on-time and the LED current were controlled dynamically to a minimum level. This was done to achieve low-power consumption.

AFE4490 provides low-power modes either via SPI command or via an external digital input [80]. In power down via SPI command, the AFE4490 still requires 111 µW. In the power down state, a power consumption of 21.3 µW at 3 V system voltage was measured. To reduce the power consumption further, three transistor power switches were added to the Printed Circuit Board (PCB) design of the PPG sensor module. This allows the shutdown of the module below 1 µW when not active.

## 3.4.2. Respiration Module

The second slot of the ULPSEK is dedicated to the respiratory module, which is illustrated in figure 3.7. The respiration module is based on the technology published previously in the patent by Weigand and Tobola in 2008 [154]. The patent is actually about two concepts: (1) a circuit for respiratory measurement and (2) the integration of respiratory measurement into a garment. An introduction to RIP was given in subsubsection 2.2.2. For ULPSEK this circuit was taken and extended by a low-power feature. Therefore, a high-side MOSFET switch was added to shut down the module.

On the software side, a driver was written for the ULPSEK in programming language C for providing respiratory signals at a sampling rate of 16 Sa/s. The driver uses a dedicated hardware timer which is integrated into the microcontroller. This timer counts the frequency of the digital signal from the respiratory module. This signal is approximately

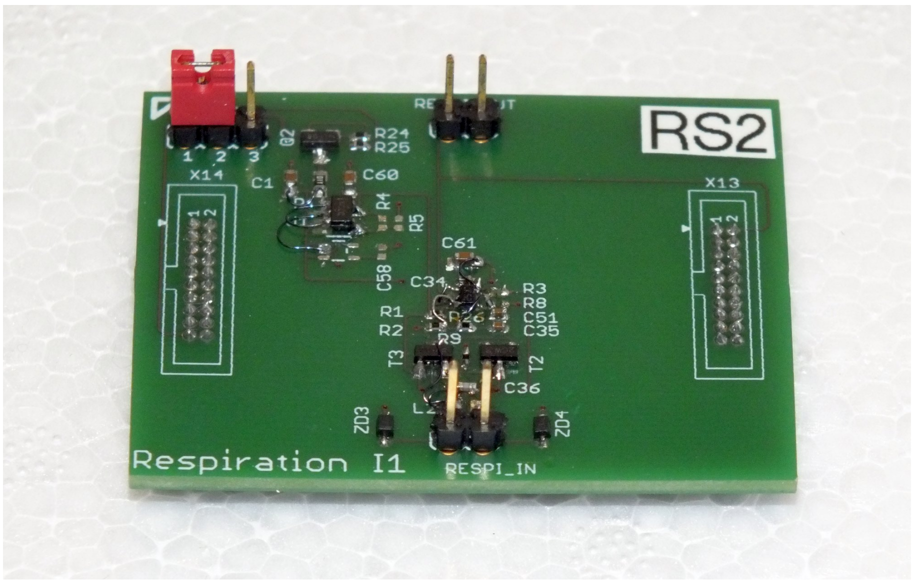


**Figure 3.7.:** ULPSEK sensor module for respiratory signal acquisition. A respiratory sensor coil or an integrated sensor garment can be connected to the module directly. The module has its dedicated connector at the ULPSEK board.

at 1 MHz and its precise frequency shifts with ±50 kHz along with the circumference of the chest. The timer counts this frequency and the driver provides this count as a respiratory signal to the application.

### 3.4.3. Electrocardiography Module

Figure 3.8 is a photograph of the ECG sensor module. In the center, the module carries the integrated analog front-end AD8232 released by Analog Devices (datasheet [12]). The AD8232 is a single lead integrated front end for signal conditioning of cardiac biopotentials for heart rate monitoring. It consists of an instrumentation amplifier, an operational amplifier, a DRL amplifier, and a voltage reference buffer. In addition, the AD8232 includes features like ECG electrode fall-off detection and a fast restore feature that accelerates filter settling after electrode contact loss or large motion artifacts. In this work, the two-electrode configuration was used instead of the three-electrode configuration. Therefore, the output of the DRL amplifier was connected through two 10 MΩ resistors to the electrode inputs. Further, the behavior of the AD8232 is intended to be modified with external passive components, e.g. resistors and capacitors. Therefore, Analog Devices provided a dedicated design tool for the calculation of filter and amplification properties. A static filter parameter configuration was designed for the first version of the hardware. In the subsequent three versions of this module, a configurable bandpass filter was designed. This bandpass can be switched from narrow band to wide band. This feature was achieved by adding analog switches to the circuit, similar to Hosticka [210]. The analog switches are used for adding and removing the passive components, which determine the filter corner frequencies. The analog switches are controlled digitally by the microcontroller. Thereby the analog bandpass filter are configurable by software. With this, the ECG module supports a narrow-band option from 4.8 to 25 Hz and a wide-band option from

0.35 to 130 Hz. The narrow-band option is better for artifact suppression, as refereed in [12]. The passband of the bandpass was selected between 4.8 to 25 Hz, keeping in mind that the following signal processing requires this bandwidth for QRS detection. When motion artifacts are low, the filter can be set to wide-band providing more details in the ECG signal.

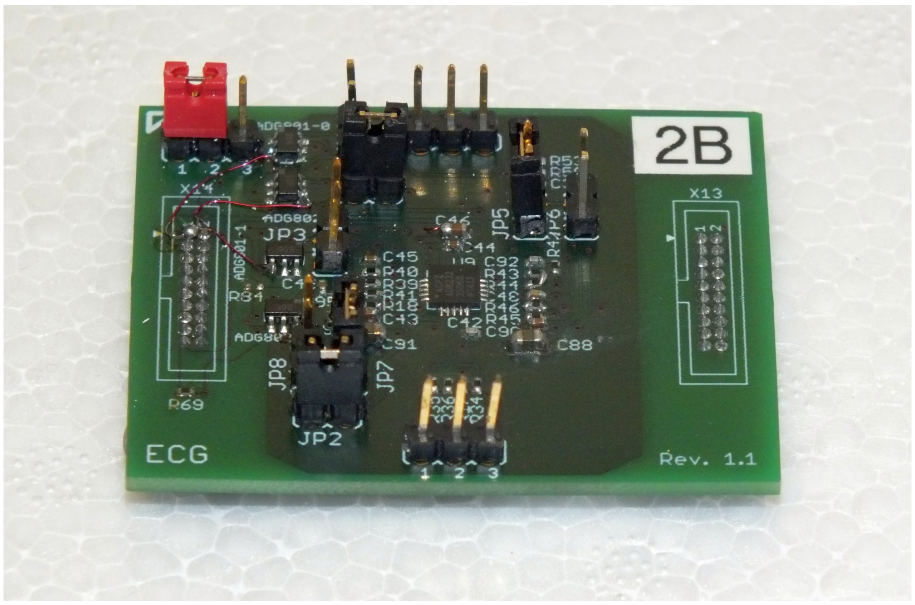


**Figure 3.8.:** The ULPSEK sensor module for ECG signal acquisition. A pair of ECG electrodes can be attached to the module.

After the bandpass filtering, the analog output of the AD8232 is connected to two inputs at the microcontroller. The first input is connected to the integrated ADC and the second to the integrated comparator. Further connections between the microcontroller and the AD8232 were established for setting the AD8232 to low-power mode, lead-off detection, and for controlling the passband of the bandpass filter.

On the software side, a driver in the C programming language was written dedicated for ULPSEK. The driver provides a single-wire ECG signal and interfaces for configuration. The driver controls the ECG module itself and three hardware peripherals of the microcontroller exclusively: ADC, comparator, and timer. The ADC uses the interrupt-driven sampling method. Additionally, the driver uses a hardware timer which is integrated into the microcontroller for timing the sampling process.

The total set of configurable features for the ECG sensor are:

1. The filter's passband is configurable between narrow band (better artifact suppression) and wide band option (more ECG details).
2. The ADC sample rates are configurable between fixed sampling rates at 100, 200 or 1000 Sa/s.
3. The ADC resolution is configurable to 8 or 12 bits
4. Instead of sampling the ECG and passing the signal through the beat detection algorithm, the ECG can be passed to a comparator where the threshold for R-peak detection is set dynamically.

5. The ECG sensor front-end can be powered down.

6. Reading the status of the lead-off detection of the AD8232.

### 3.4.4. Accelerometer Module

Figure 3.9 shows a photograph of the module hosting the accelerometer. This board consists of the integrated circuit MMA8452QT released by NXP Semiconductors N.V. (datasheet [65]). The MMA8452QT is a 3-axis accelerometer with digital signal processing. Data rates up to 800 Sa/s and selectable acceleration ranges from 2 g to 8 g are configurable. Data readout and configuration are handled via an $I^2C$ interface. Sampling rates, filters, and additional onboard processing are configurable via registers. The MMA8452QT provides configurable interrupt outputs on special events, e.g. when a defined acceleration force is exceeded. A sensor driver was written for ULPSEK for configuration and data readout in C programming language.

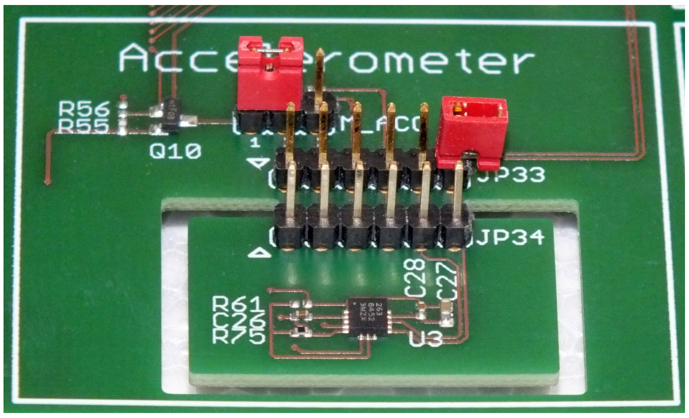


**Figure 3.9.:** ULPSEK sensor for 3-axis acceleration signal acquisition. This module is designed as a fixed part of the ULPSEK board with the ability to break out the board with the integrated circuit for testing purposes.

### 3.4.5. Temperature Module

The temperature module contains the integrated circuit TMP102 released by Texas Instruments (datasheet [158]). Data readout and configuration are handled via $I^2C$. The sensor supports sampling by an internal clock. Alternatively, the sensor supports so-called one-shot readings. A sensor driver was written for ULPSEK for configuration and data readout in C programming language.

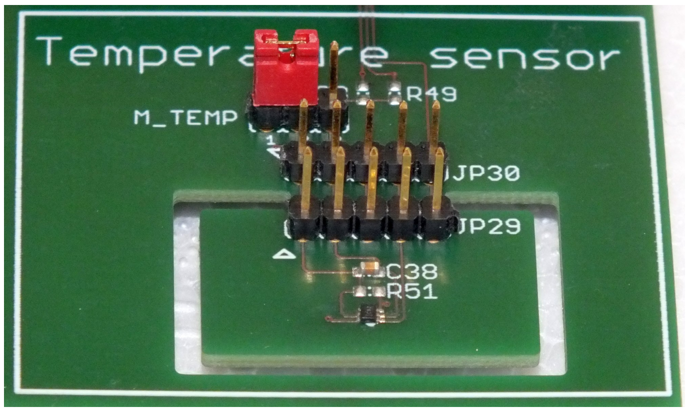


**Figure 3.10.:** ULPSEK sensor module for skin temperature. The IC is intended to be connected via heat sink to the body surface.

## 3.5. Radio Module

The radio module hosts itself the actual radio module of type PAN1721 released by Panasonic. The PAN1721 again contains the CC2541 [20] released by Texas Instruments for Bluetooth Low Energy (BLE) connectivity. This module allows ULPSEK transmission of biomedical data and for receiving control commands. A driver was written for the microcontroller in programming language C to provide this functionality on the abstract application level, e.g. passing biomedical data to a smartphone and receiving configuration commands from a smartphone application.

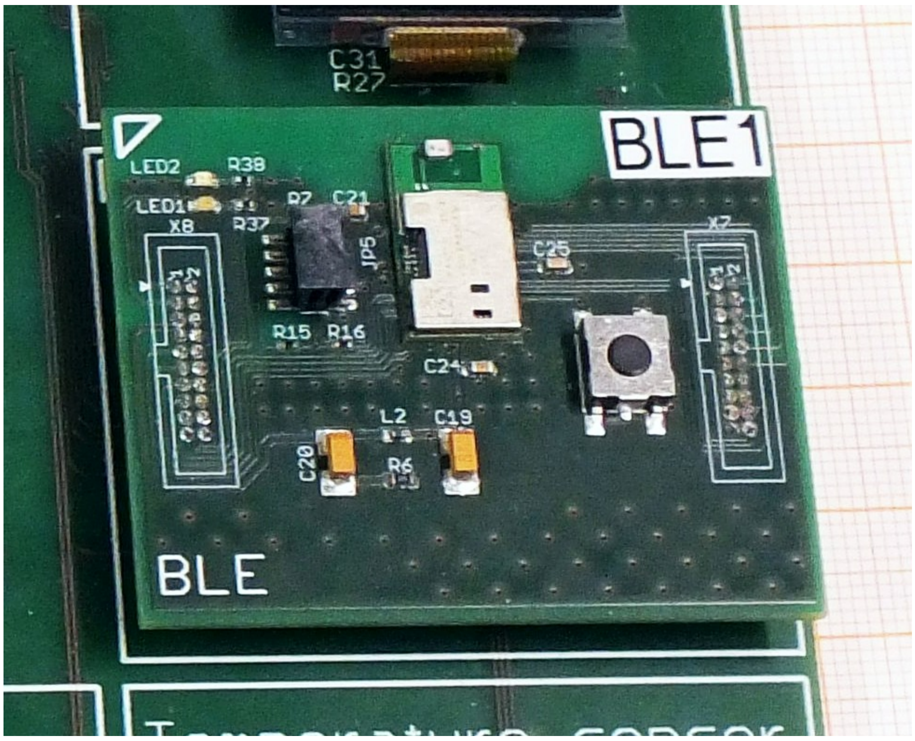


**Figure 3.11.:** The Panasonic PAN1721 is an ultra-low-power SoC for BLE applications. This module was used to transmit biomedical data and receive control commands. This photograph illustrates the ULPSEK communication module hosting the PAN1721.

## 3.6. Display

The ULPSEK hosts a Thin-Film Transistor (TFT) display model LS013B7DH03 released by Sharp [118]. This TFT display is capable of monochrome graphics and text with a resolution of 128 by 128 pixel. The purpose of this display was to visualize raw sensor data and vital parameters in real-time. A driver was written in C programming language for the microcontroller for real-time graphical representation of sensor data. The driver implements double buffering for writing data in the first buffer while displaying the content of the second buffer. Furthermore, a graphical library was written for elementary graphical shapes, lines, and text. The display driver supports three modes: (1) off, (2) parameter mode, which requires 130.5 µW, (3) raw data mode for visualizing ECG with 1699 µW for graphical computing. Visualizing the real-time ECG signals required sample rate conversion implemented using a decimation filter. The decimation is required to avoid aliasing when displaying the ECG on display with only 128 pixels in a row. Without decimation, the aliasing effect negatively influenced signal clarity on the display output. However, the decimation filter requires power for computing which will be measured and added to a model later.

## 3.7. Power Distribution Network

Figure 3.12 is an abstract representation of the power distribution network. This figure represents the power distribution network as a feed-forward graph, starting at the power sources to the left and ending at the peripherals and sensors to the right. There are three power sources:

- USB port for battery charging
- Optional wireless charging (Open Interface standard that defines wireless power transfer from the Chinese word qi (Qi)) via USB port
- Optional energy harvesting

The output of the wireless charger is connected directly to the USB port charger. Usually, the battery is recharged by either the USB port or wireless charging. After the battery, four slots for independent voltage regulators follow.

- Primary digital voltage regulator
- Secondary digital voltage regulator
- Primary analog voltage regulator
- Secondary analog voltage regulator

The primary regulators were linear voltage regulators in this work. The idea of the secondary regulators was to use more power-efficient switched regulators for high loads. However, most of the switched regulators have high quiescent currents. The idea was to switch back and forward depending on the load for the best power efficiency. However, the exciting idea of using switched regulators was never realized because of noise issues with the switching regulators. Therefore, two linear regulators were used: one to supply the digital circuitry part and the other for the analog side. In general, selecting linear regulators with low quiescent current is desirable for low-power systems. On the other side, when selecting linear regulators for the analog power domain, a high Power Supply Rejection Ratio (PSRR) is required to provide a clean power supply for the sensitive analog amplifiers and sensors. Unfortunately, a high PSRR correlates with a high quiescent current [162]. Therefore, the regulator for the digital power domain was selected with a lower quiescent current at 11 µA on the cost of lower PSRR. The regulator for the analog power domain was selected with a better PSRR on the cost of higher quiescent current at 26 µA. With this, the analog supply can better deal with power supply noise, particularly problematic through noise caused by the radio module. Consequently, the analog regulator's quiescent current is higher than the quiescent current of a digital regulator. When using low-power modes, analog sensors are shut down. Nevertheless, the microcontroller requires power at the analog power inputs. Therefore, an additional power-saving feature was implemented. The switch number 1, right after the digital regulator, can route the power from the digital domain to the analog domain. As a consequent advantage, the analog regulator can be switched off without powering down the analog domain. The common problem was that even if an IC has a shutdown mode, some of the IC still consume 10 times more current than the quiescent current of the linear regulators. This additional power consumption is not an issue when measuring and computing, but in low-power mode, particularly when hibernating for a long time, it would dominate the battery life. Therefore, eight p-channel MOSFET switches were inserted. One into each path of the energy consumers to shut down the power distributed to the peripherals completely. All eight switches are controlled by the microcontroller. However, after connecting the power to the board, the microcontroller needs time until the first lines of code are executed. During this boot sequence, the microcontroller can not control the power switches for the peripherals. Without any control, power switches would switch randomly. This effect could cause unnecessary energy waste, undefined conditions, and even hardware damage would be possible. Therefore, a secure state was realized by using pull-up and pull-down resistors. In general, values between 10 kΩ an 1 MΩ are typical when using pull-up and pull-down resistors. Unfortunately, this range is a problem in a low-power system. When using high resistor values, resistors together with the circuit capacitance cause long-time constants. Consequently, more extended switch on or off time is required. Timing and low-power optimization were addressed while parametrizing the hardware. However, in the first design eight 1 MΩ resistors were used to control the eight switches and caused another problem. Assuming eight switches with 1 MΩ resistors at 3 V the power consumption therefore is 24 µA. Again, this was the dominant parameter for battery life. The solution was to use 10 MΩ resistors in the design. This modification led to a satisfying power consumption of the default setup for the power distribution network.

The switches are the gates to the power domains of the systems. The power tree splits further until it reaches the leaves with its five-sensor frontends, multiple peripherals, and the microcontroller. When developing low-power systems, at least the total power consumption must be checked after every modification. The total power consumption often increases after hardware and software modifications. An increase in power consumption should be identified as early as possible. After several modifications, it is harder to find the cause of increased power consumption. An additional hint when searching for the cause of increased power consumption is the ability to measure not only the total power consumption but also at every separate leave of the power distribution network. Therefore, for measuring current in each domain separately, the ULPSEK design in version 1.3 was extended with 19 measurement terminals. Each measurement terminal has a three-pin connector for measuring current and voltage for each power domain separately. The three pins are: (1) current source, (2) current sink, and (3) ground. Those terminals were highlighted on ULPSEK by red colored jumpers, which are bridging pin 1 and 2 by default. When measuring at a specific terminal, the jumper was removed and replaced by a custom three-pin header. This header contained a 1 Ω resistor placed at pin 1 and 2. The header was also used for monitoring the voltage at pin 2 against the ground. In figure 3.12 equivalent power measuring terminals are marked with the prefix 'M_'.

Beside the monitoring with external equipment, a feature for self-monitoring was added to the ULPSEK. Therefore, the board hosts a power measurement IC, the INA219 [71], enabling the system to measure its total current and battery voltage. The idea behind this self-monitoring feature was to use a digital twin to optimize power and detect safety and security issues. Although this idea was not realized for this thesis, the concept of self-monitoring, optimization, and anomaly detection was published with the invention disclosure in 2022.

To conclude, the power distribution network was optimized iteratively before the development of the software started. Hardware design optimization was an essential precondition for further work. The microcontroller controls the power distribution network's power gating. Thus it is the responsibility of the software to control the sensor's system power efficiency and safety.

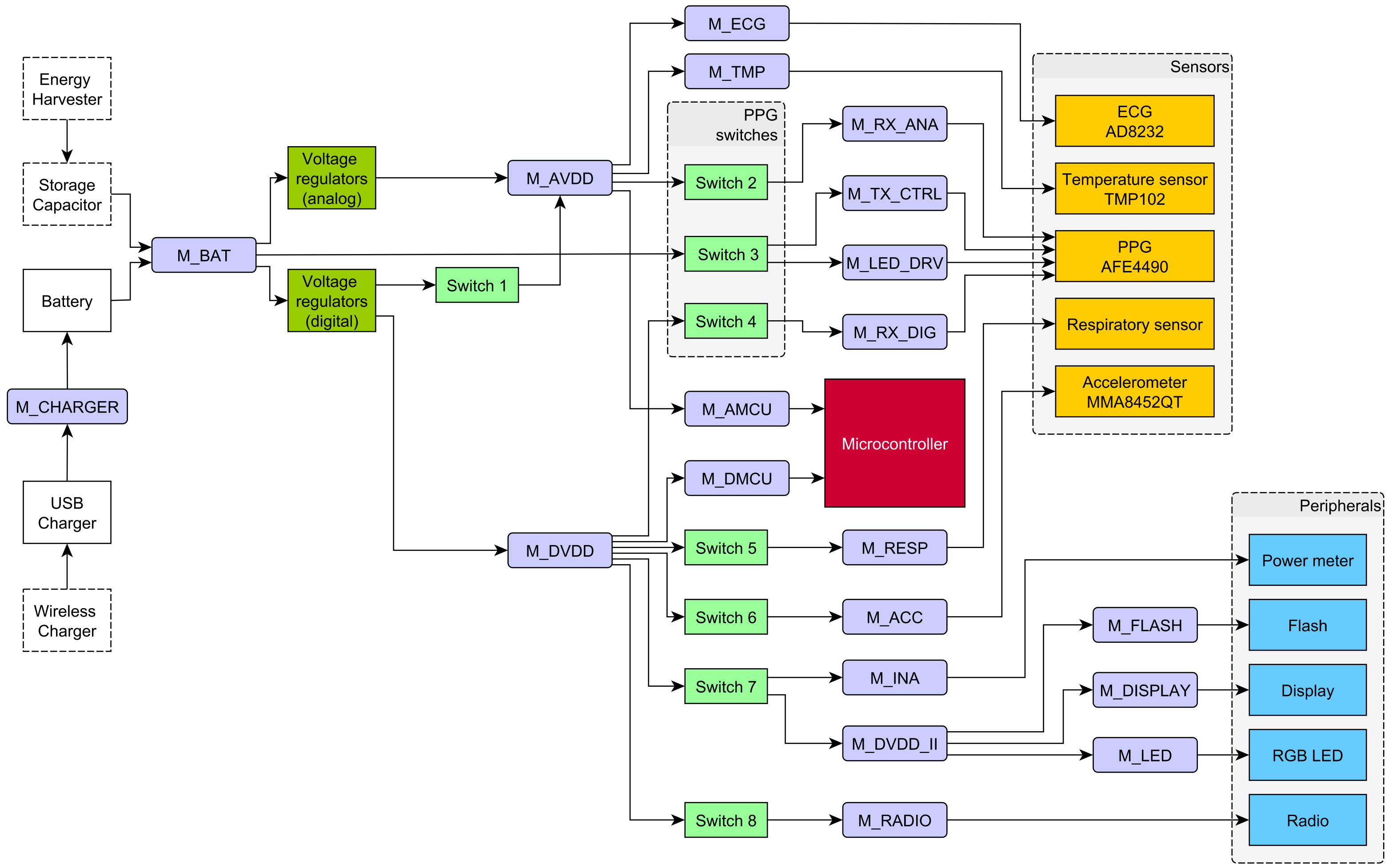


**Figure 3.12.:** This diagram is an abstract representation of the power distribution network of the ULPSEK. It was an essential tool for optimizing hardware and software properties during the development. Each power measuring point in this network has a label starting with M_. The power sources, such as the USB charger and battery, are located to the left. Analog and digital regulators follow the power sources with shutdown functionality. The output of the regulators is distributed to eight power switches before reaching the consumers, such as microcontrollers, sensors, and peripherals.

## 3.8. ULPSEK Software

The embedded software for ULPSEK was written in the programming language C.

### 3.8.1. Software Architecture

Architecture is the process of planning, designing, and constructing buildings or other structures. Likewise, software architecture is the process of planning, designing, and constructing software. For this purpose, several views on software architecture exist, and each view has a different perspective and purpose. All views together define the software architecture. The software architecture of ULPSEK consists of three diagrams.

- **Software Architecture Diagram**, a non-formal diagram, represents all existing software components (figure 3.13). The software modules in this diagram were organized in layers on top of the hardware for better software quality.
- **Class Diagram** according to the standard Unified Modeling Language (UML) 1.4. This diagram defines how software components depend on each other (figure 3.14). It aims to plan how software components can be reused, replaced, and reconfigured for better software quality.
- **Signal Flow Diagram** shows the path of the data through the software system. This diagram has been proposed by the author of this thesis to fill a gap regarding signal flow diagrams. A detailed specification for this kind of signal flow diagram was included in the appendix A.

The Software Architecture Diagram is an abstract representation of the software components categorized and ordered in functional groups and layers. Each abstract software component in the architecture diagram has a counterpart in the software. The purpose was to increase the code quality through cohesion. In software engineering, cohesion measures how the operations of software modules are related. A high cohesion, and thus good cohesion, is given when a software module provides a separable and specific functionality. McConnell [166] describes practical methods of designing software in terms of high cohesion. Thus, the purpose of this diagram was to give each component a place and individual responsibility within the software's entire system. This architectural view is comparable to an organizational chart where each department has its function and responsibility. The stack of layers indicates a hierarchy where software components take control down to the hardware at the bottom. For the software developer, this diagram was an orientation giving which components belong to which group or layer, which are closer to each other, and indicating which components were allowed to communicate directly. This architectural view contributed to a better software structure, re-usability, and overall software quality. The software architecture consists of multiple layers. The foundation of the architecture is the hardware layer. On top of it, the software provides drivers which access the hardware. Build on the drivers are two layers: control and communication. The control and communication layer does not access the hardware directly and requires a driver to communicate with the hardware. The components within these layers will be explained later in detail. The RTOS provides the functionality required by software layers. Therefore, the RTOS has a special place in the architecture diagram. The link layer has a unique function in this diagram which was done the first time this way. The link layer connects signal processing to the control elements during runtime. Finally, the signal processing components are located on the top of the architecture diagram. The following walk through the layers presents the involved modules and their responsibilities.

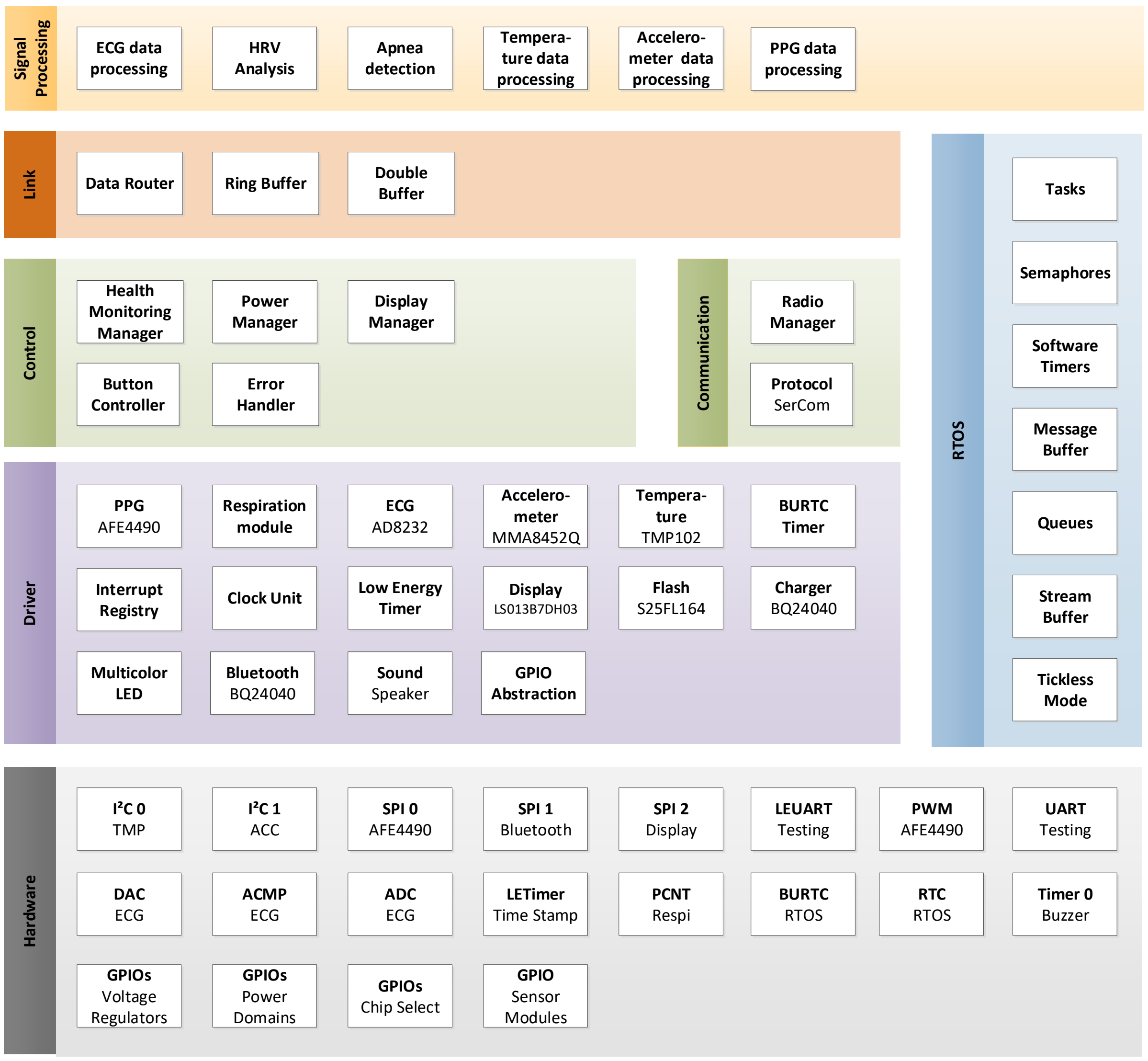


**Figure 3.13.:** The software architecture diagram shows the components of the ULPSEK embedded system, grouped in hardware, drivers, control, communication, link, signal processing and RTOS.

## 3.8.2. Operating System

Once the microcontroller has started, the first software module which gets the control is the operating system. The RTOS is used to set timers, handle events and tasks, and avoid wasting energy using the tickless mode. The RTOS is configured to use the energy mode 2 of the Energy-Friendly, 32-bit Microcontroller by Silicon Labs Inc. (EFM32) microcontroller to save power automatically during idle. After the start, the operating system gives the power manager highly prioritized control.

### 3.8.3. Control Layer

Five software modules are assigned to the control layer. The most relevant within the scope of this thesis are the Health Monitoring Manager (HMM) and the power manager. The HMM contains a high-level program for monitoring a person wearing the sensor. This program defines what and when data is collected. If required for the application, the program of the HMM can change the strategy based on the sensor data.

The next component is the power manager. The purpose of the power manager is to control hardware and software states concerning their power consumption. The power manager coordinates the hardware and software to execute the monitoring tasks correctly and at a minimum energy level. The power manager is the first system part initiated after powering on. After the power on button is pressed, the power manager walks through the power distribution network step by step, initializing the default states for the regulators and switches. After powering on, all subsystems are powered off by default. An additional responsibility of the power manager is to set the system only to valid states. Therefore, the power manager has some embedded knowledge about the system implemented as a set of logical rules, timing restrictions, and dependencies. The power manager receives commands which activate or deactivate hardware and software. There are two sources of commands for the power manager: via wireless remote or from the HMM.

The HMM starts after the power manager sets up the system. The purpose of the HMM is to execute a monitoring program. This program was introduced in the fundamentals subsection 2.1.5. The program defines the application. On ULPSEK, this program defines what to do and how to react to events optionally. For example, the HMM tells the power manager to activate some sensors at specific sampling rates, activates the related signal processing algorithms, the display, and the radio module. Immediately after activation, the program of the HMM sends a predefined series of commands to the power manager. The HMM can also send a hibernation command repeatedly. This is how interval sampling was implemented. Alternatively, the HMM can send a hibernation command based on events. With this, programming the sensor system on the abstraction level of the HMM is straightforward. The HMM does not need to be concerned about the order and the timing of the commands. The complexity of the hardware and software dependencies is hidden inside the power manager. Valid order of commands and timing restrictions are the power manager's responsibility. For example, the power manager knows that the activation of heart rate computing also requires activating the ECG front-end. Furthermore, the power manager knows that activating the ECG front-end also requires activating the analog power domain through the analog voltage regulator. As such, there is no need for the HMM to care about all the details. All the HMM has to do is communicate that the ECG is required at a specific rate. The simplicity of HMM can be shown with a code example. The code listing 3.1 shows a program for continuous data sampling and transmission. The job ends for the HMM by executing two commands.

**Listing 3.1:** HMM example code with a two-line configuration.

```
pM_event(PME_ECG_sampling_mode, S200); //ECG sampling at 200 Sa/s
pM_event(PME_ble_on, 0); //activate wireless transmission
```

The code listing 3.2 demonstrates another more detailed example for the HMM with multiparameter data acquisition, parameter computing, and interval monitoring. The HMM repeats a sequence endlessly in this configuration.

**Listing 3.2:** HMM example code with a more extensive configuration.

```
while(true) // Endless loop
{
    //*** Measure for 20 seconds ***
    pM_event(PME_ECG_hr, true); //activate heart rate algorithm
    pM_event(PME_ECG_hrv, true); //activate HRV algorithms
    pM_event(PME_ECG_sampling_mode, S1000); //ECG sampling at 1000 Sa/s
    pM_event(PME_temp_one_shot, 0); //execute one temperature measurement
    pM_event(PME_ECG_bdac, 0); //activate beat classification algorithm
    vTaskDelay(20*1000 / portTICK_PERIOD_MS); //wait 20 seconds

    //*** Measure, display and transmit for 10 seconds ***
    pM_event(PME_disp_graph, 0); //activate display in graphical mode
    pM_event(PME_ble_on, 0); //activate wireless transmission
    vTaskDelay(10*1000 / portTICK_PERIOD_MS); //wait 10 seconds

    //*** Prepare for hibernation ***
    pM_event(PME_ble_off, 0); //switch off wireless transmission
    pM_event(PME_ECG_off, 0); //switch off ECG
    pM_event(PME_disp_off, 0); //switch off display
    vTaskDelay(500 / portTICK_PERIOD_MS); //wait 500 ms

    //*** Hibernate for one minute ***
    LED_pulse(led_color_red); //short red LED signal
    pM_event(PME_hib_min, 1); //hibernate for one minute
    LED_pulse(led_color_green); //short green LED signal
    vTaskDelay(500 / portTICK_PERIOD_MS); //wait 500 ms
}
```

The two program examples show how applications can be defined using the HMM. The programming of the HMM is simplified relative to the software layers below the HMM. Within the scope of this work, the power manager became the software component with the most substantial importance for the realization of low-power applications. It was surprising because the initial role of the power manager was to switch on and off several domains of the power distribution network. However, as the project evolved, the power manager was given an extended responsibility. This responsibility is defined as that no matter what was requested by the application, the power manager will never allow an invalid or even dangerous state of the hardware and software. For this purpose, the power manager was given a comprehensive set of rules, which includes knowledge about the hardware and software modules. The rules include logical and timing restrictions. For example, a logical restriction would be being only allowed to switch off a specific power domain when any dependent subsystem no longer uses it. An example of a timing restriction would authorize the powering off of a specific submodule after waiting five milliseconds because of hardware restrictions. The number of rules added to the power manager increased quickly during the project. As the software matured, the number of rules added to the power manager grew rapidly, transforming it into the most complex software module. Over time, the power manager became crucial in efficiently scaling performance to a tailored minimum for each application under evaluation.

### 3.8.4. Link Layer

A link layer was introduced to the ULPSEK software architecture for connecting signal processing parts dynamically during runtime. This link layer is not to be confused with the Open Systems Interconnection (OSI) model. Most software modules for signal processing were coupled extremely hard. Hard coupling of software is a typical example of low cohesion. As mentioned above, cohesion in software engineering is a sign of bad code quality. Consequently, replacing, reusing, or modifying the signal processing components was time-consuming and complicated. This problem exists because every component depends on the other while passing data back and forward.

To overcome this issue, a link layer was introduced to the ULPSEK's software architecture. The idea here is to implement common interfaces for communication to each signal processing block while leaving the software modules unconnected. After compiling the code, the modules stay unconnected, which is excellent for replacing modules, isolated testing of modules, and creating configurable software. The connections are created once the application is launched. The architecture diagram in Figure 3.13 shows the location of the link layer within the software architecture. There, the link layer separates signal processing from control modules. A more detailed view is contained in the class diagram with figure 3.14. The class diagram is an object-oriented view of the software components according to UML in version 1.4. With the introduction of the link layer, software components are unbound and can therefore be reused, exchanged, and even attached at runtime. Links do not exist during software compile time. Instead, the links are established during the system's start. From a code perspective, this is done by providing standard interfaces for passing data between modules. The link is established by calling a function for interconnecting a data source with a sink. For example, the output of the ECG driver is connected to the input of the ECG processing during system start. Then, the output of the ECG processing is connected to HR processing. Furthermore, the output of the HR is connected to the display and radio manager. The advantage of doing the link during runtime is separating and reusing the modules. Therefore, the link layer defines a standard interface for data exchange within the sensor system. All parts must implement this standard interface for internal data exchange. There was an opportunity to improve the link layer introduced to ULPSEK further. At Siemens, a reusable software component could be introduced that implements a so-called DSP Blackboard. The base idea is the same as the link layer of ULPSEK. The DSP Blackboard was not integrated to ULPSEK. However, this software became a mandatory component for connecting signal-processing modules in several projects.

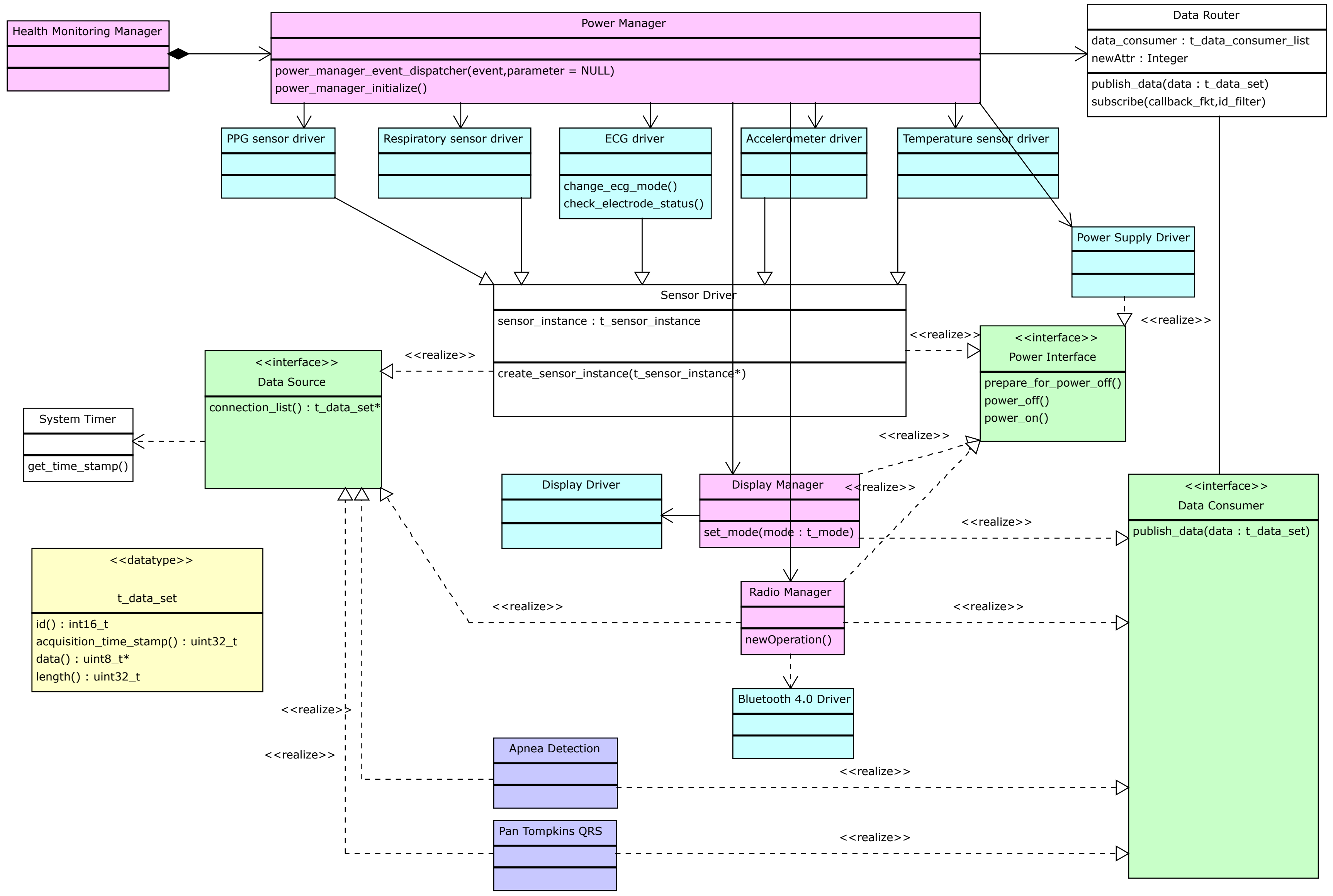


**Figure 3.14.:** Class diagram according to UML version 1.4 showing the dependencies and interfaces of the ULPSEK.

### 3.8.5. Signal Processing Layer

The signal processing layer contains software components for processing and analyzing data. Each signal processing component has data inputs and data outputs. Additionally, most of the signal processing components have interfaces for configuration. The signal processing layer interacts with the system over the link layer. The ECG processing was the most populated and investigated signal processing within this thesis. Therefore, it is described in more detail than the other implemented processing modules.

#### ECG Processing

Figure 3.15 shows the signal flow chart of ECG processing. The particular new type of signal flow diagram is described in appendix A. This chart has three sections: (1) The ECG sensor front-end, (2) the parameter computing, and (3) the data block where the derived data is accessible for further processing, showing up at the display, or transmission. The derived data consists of the ECG signal itself, a beat classification (normal or abnormal beat) [171], the Normal to Normal (NN) interval[1], HR, and the two HRV parameters (RMSSD and pNN50%).

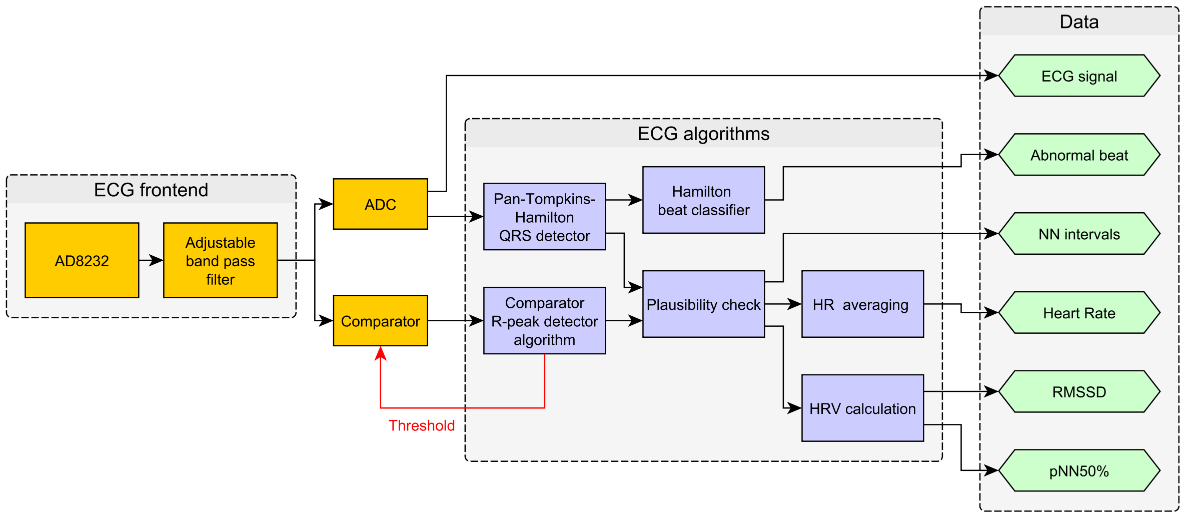


**Figure 3.15.:** This figure shows a signal flow diagram of the ECG signal processing path. The diagram is grouped into three areas: ECG sensor, parameter extraction, and data. Each arrow represents one signal pipeline from one block to another. Each block requires only one pipeline input to produce data at its output pipeline. The graph is oriented in one direction except for the feedback loop for the adaptive threshold. More detailed instructions for this type of signal flow diagram are included in the appendix A.

The signal processing starts with the ECG amplification using the AD8232. A configurable band-pass filter was added where the corner frequencies are adjustable. This band-pass filtered signal is then used twice. One path is leading to the ADC and the other

[1]NN interval: Normal-to-normal interval is the time between two adjacent points of the same type within an ECG or another cardiogram. Most often, two adjacent R-peaks.

to the comparator. However, only one path is active at the same time. Either the ADC produces samples or the comparator generates interrupts on each detected heartbeat. The ADC is connected to the QRS detection algorithm. There are two parallel implementations of the QRS detector. One variant is compiled for processing at 100 Sa/s, and the second variant is compiled for processing at 200 Sa/s. Depending on the provided sampling rate, the matching beat detector is activated by the software during runtime. Additionally, for other sampling rates than 100 Sa/s or 200 Sa/s, a decimation or interpolation filter is used to convert to the target sampling rate of 200 Sa/s. The used QRS detector is the Pan-Tompkins algorithm. The Pan-Tompkins algorithm [205] is probably the most popular QRS detector in science. This algorithm was designed in 1985 by Pan and Tompkins and implemented in assembly language to run on the Z80, a famous 8-bit microprocessor. Pan and Tompkins work was cited thousands of times by other scientists. Thus, various modifications of the original algorithm exist today. The original algorithm was improved one year later and rewritten in C-programming language by Hamilton and Tompkins [202]. Fortunately, Hamilton published an open-source implementation of the algorithm in 2002. He also added an optional beat classifier. Hamilton published this code together with a paper [171]. This paper contains the performance measurement of this algorithm according to the Massachusetts Institute of Technology (MIT)/Boston's Beth Israel Hospital (BIH) database. The QRS detector was designed primarily for 200 Sa/s and, as mentioned above, was also compiled for 100 Sa/s. The already evaluated and, in terms of scientific reproduction, freely available source code was predesignated for investigation regarding energy consumption in this thesis. Therefore, the Hamilton implementation of the QRS detector was integrated into the ULPSEK software. With this, ULPSEK implements ECG sampling and QRS detection.

Additionally, a hardware-based R-peak detection was implemented. This method is an alternative for sampling. Instead, a comparator is used with an adaptive threshold. The usage of the comparator was inspired by Thakor et al. [208]. The comparator continuously compares the ECG signal level with a digitally controlled threshold. This threshold must be set above any other waves than the R-peak. Therefore an algorithm increases or decreases the threshold. The threshold will be increased each time a second interrupt occurs within the blanking interval of 200 ms. The threshold will be decreased each time an R-peak should, but was not detected after an expected time. The expected time is calculated by the previous RR interval adding 50 % tolerance. The advantage of this method is a lower power consumption compared to the method utilizing sampling and QRS detection. Only one beat detector can be activated simultaneously: sampling-based or comparator-based beat detection. After that, every detected QRS complex is checked for physiological plausibility to occur between 30 and 240 bpm. The output of the plausibility check is an RR interval. After the check for plausibility, HR calculation is executed for each RR interval. The HR calculation includes smoothing over four data points using a moving average filter. Advanced Intensive Care Unit (ICU) monitors provide HR smoothing using configurable moving average filters for better readings at the display. In parallel, the RR intervals are processed with the HRV algorithms. The HRV software module implements RMSSD and pNN50% according to the formal description in [188]. The ECG processing software was designed to be configurable during runtime, which was a precondition for the experiments in section 6.1.

#### Respiratory Processing

A substantial part of the respiratory processing was the apnea algorithm. The algorithm classifies the signal strengths into two classes: *normal breathing* or *apnea*. The code for the algorithms is included in Appendix B. The algorithm was developed initially in the research project *Smart Sensors A* and published in the public project report [77]. This algorithm was also used for benchmarking the microcontrollers in the chapter 5.

#### PPG Processing

The PPG infrared channel raw data is forwarded directly to the display and radio module without processing. For the display, the sampling rate needed to be reduced to fit at least one complete pulse wave in the display with 128 dots in width. A decimation filter for rate conversion was implemented to avoid aliasing effects. Aliasing effects were seen when using subsampling instead of decimation. However, decimation is a processing task and processing requires energy.

#### Accelerometer Processing

The accelerometer processing is provided by the accelerometer IC. Therefore, the accelerometer IC is configured by the driver. The computed data is forwarded to the display and radio module without processing.

#### Temperature Processing

The temperature is forwarded directly to the display and radio module without processing.

## 3.9. Low-power Idle and Hibernation

ULPSEK implements two low-power modes: Low-power idle and hibernation. Table 3.1 gives an overview of the differences between the two low-power modes. Both modes set the microcontroller in sleep mode. The purpose is to conserve energy when computing is not necessary. Additionally, only hibernation switches off the sensor electronics. Each mode has a dedicated purpose. The main difference is the duration of sleep. Low-power idle was made for a short sleep, and hibernation was made for a long sleep. Low-power idle was implemented for short-term sleep under 5 s and it was tested down to 800 µs. Theoretically, low-power idle should be possible down to 100 µs with ULPSEK. *Idle* is a mandatory task on a RTOS executed when there is nothing to do. The RTOS was programmed to enter this mode when *idle* and wake up automatically for a planned job. Additionally, ULPSEK implements hibernation to support deeper sleep modes which conserve even more energy. During hibernation, the microcontroller is in a deeper sleep state and the sensors are powered down. A disadvantage of this mode is a longer time and higher energy cost for entering and returning from hibernation compared to the short sleep in idle. While low-power idle has a cost of $E_{STR} < 1\,\mu\text{J}$ for entering and returning from sleep, the hibernation was measured at $E_{hib,er} = 31.5\,\text{mJ}$. Accordingly, the time for entering and returning takes 29 times longer in hibernation than in idle sleep. Thus, hibernation is made for more prolonged and less frequent sleep periods. Therefore, a model is needed to decide which sleep state is more beneficial for energy conservation. The modeling will be addressed in chapter 7. Hibernation is the responsibility of the Power Manager, which was already introduced in subsection 3.8.3. Sensor applications can request hibernation by sending a command to the power manager. The application must also tell how long the system should hibernate. The power manager has information about all active components. Furthermore, the power manager knows the dependencies between the hardware and software modules. For example, the power manager prevents switching off an ECG sensor when ECG algorithms are running. First, the algorithms must be switched off, and then the sensor power supply can be switched off. Another example is the case where the power manager rejects a command for switching off a power domain when the power domain is a supply of an active ECG sensor. As such, the power manager powers down the active algorithms first, then the sensors, then the parts of the power distribution network. After that, the power manager adjusts a timer for wake-up as configured by the application and executes a deep sleep command. Then, only the wake-up timer clocked by a low-power crystal 32 kHz and the voltage regulators remain. With this, the system is powered down to a total of 22 µW.

**Table 3.1.:** ULPSEK implements two fundamental low-power modes: Low-power idle and hibernation. This table compares the essential differences.

| **Property** | **Low-power idle** | **Hibernation** |
|---|---|---|
| Purpose | short sleep | long sleep |
| Sleep duration | 100 µs to 5 s | 2 s to 255 h |
| Sensor power gating | always on | switched on/off |
| Continuous sampling possible | yes | no |
| Transition time per cycle | 40 µs | 1.15 s |
| Transition energy per cycle | <1 µJ | 31.5 mJ |
| Power consumption[1] | 250 µW | 22 µW |
| Controlled by | Operating system | Power manager |

Once the wake-up timer reaches the set wake-up time, the system wakes up from hibernation. After the microcontroller starts, the HMM (software module) sends commands to the power manager. Then, the power manager reconfigures all hardware and software modules while keeping the dependencies consistent. Hardware, control logic, state machines, data buffers, and algorithms need to be brought into a steady state. It takes ULPSEK between 1.15 s and 20 s to restore from hibernation. The 1.15 seconds are necessary to start software modules and warm-up hardware modules. Further, time is required to fill software buffers for signal processing with the latest signals. The filling of buffers is required to reach a valid computing state. For example, temperature measurement can be achieved quickly, but an ECG analysis requires 10 s until valid heart rate parameters are

[1]The power consumption of ULPSEK during sleep depends on the sleep mode. For more details, see experiments with ULPSEK in chapter 6.

provided. To be summarized, ULPSEK hardware and software implement hibernation, a required feature for interval-driven, episodic-driven, and event-driven monitoring.

## 3.10. Miniaturizing ULPSEK for being wearable

In addition to ULPSEK, a miniaturized and thus wearable sensor system was developed which is electrically equal and code-compatible to ULPSEK figure 3.16. Of course, the miniaturized sensor system has a fixed hardware setup without the advantage of modularity. The advantage of miniaturized hardware is the ability to be wearable. The disadvantage is that power measurements are almost impossible due to the lack of power-measuring terminals. Therefore both systems existed in parallel: The large flexible ULPSEK and the compact version. However, the miniaturized hardware is fully code-compatible to ULPSEK. Thus, the development was executed at the actual development kit. Furthermore, once power consumption was satisfying, the compiled firmware was copied to the miniaturized version.

The miniaturized version was integrated into an experimental chest sensor belt. With miniaturization, modularization was also abandoned. Instead, modules have been replaced by fixed subgroups as follows:

- Two fixed 3.0 V linear voltage regulators TPS78230DRVT for analog and digital power domain
- microcontroller ARM Cortex-M4F in particular EFM32WG390F256
- radio Bluetooth low-energy module PAN1721
- ECG analog frontend AD8232 in version B2; in detail published in Tobola et al. [100]
- inductive respiration circuit according to the patent by Tobola and Weigand [154]
- accelerometer MMA8452QT as body motion sensor
- TMP102 as a body temperature sensor
- memory TFT display LS013B7DH03

While ULPSEK has five sensor modules, the miniaturized version has four. The PPG sensor was dropped for the chest belt sensor. The power distribution network, including power gates, capacitors, battery charger circuit, other minor parts such as LEDs, and multifunction button, were kept equal as on ULPSEK.

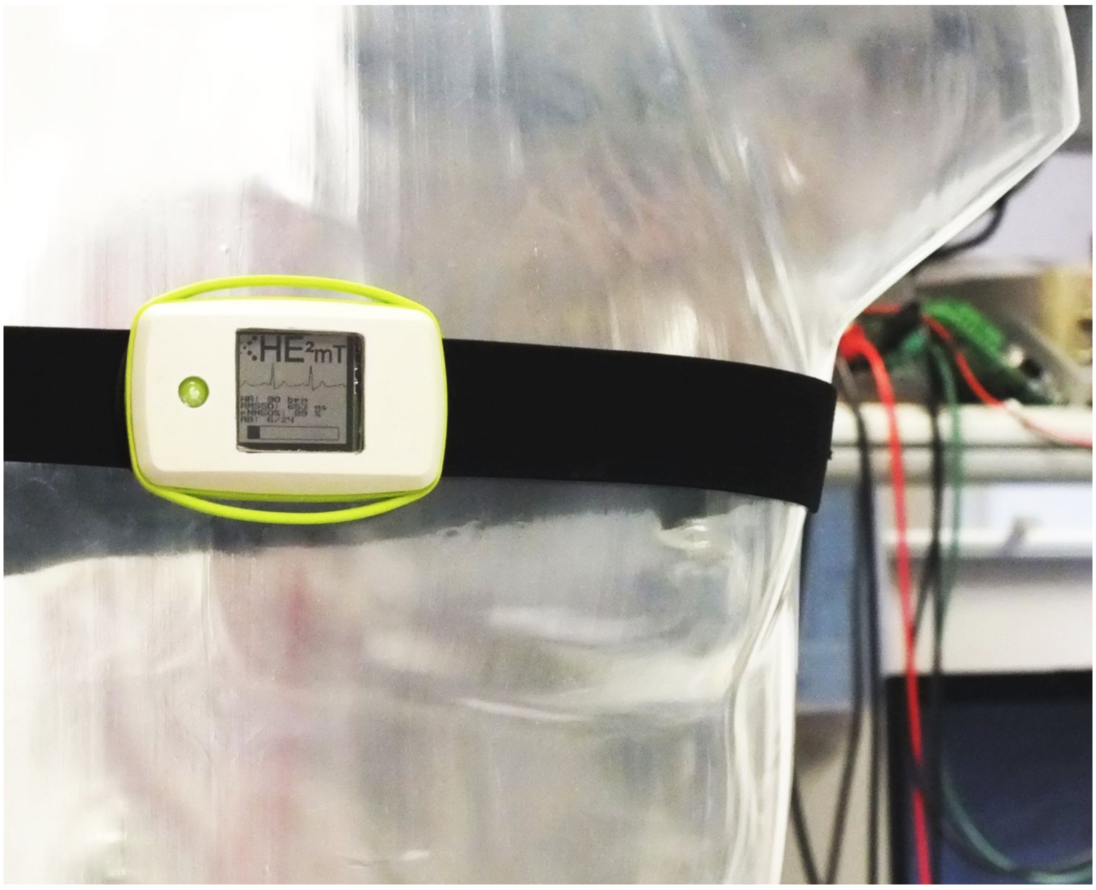


**Figure 3.16.:** Photograph of the miniaturized and code-compatible derivate of ULPSEK integrated into a wearable chest belt.

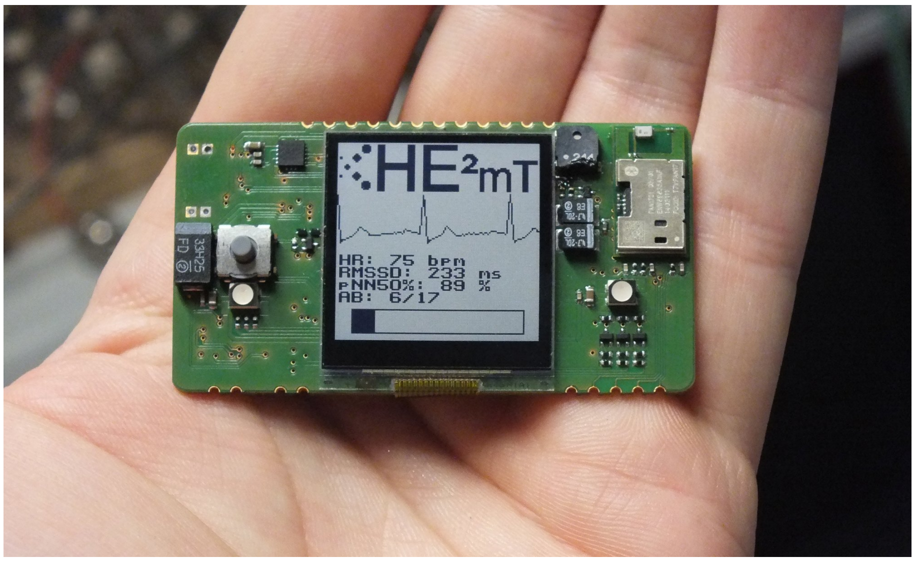


**Figure 3.17.:** Photograph of the miniaturized and code-compatible derivate of ULPSEK. The plain PCB without housing has a size of 60 mm x 32 mm, or 2.4 in x 1.3 in.

## 3.11. Chapter Summary

Creating an entire sensor system from scratch instead of using available sensor hardware and software was time and cost-intensive. As an advantage, full access to hardware and software was a precondition for this thesis's experiments, which finally paid off. The most challenging task was the interactive optimization of the system's power distribution network and the optimization of the power manager software. The learned lessons were: a careless written driver, a simple pull-up resistor, or some error in the software can dominate the power consumption. Thus, ULPSEK results from detailed planning and intense and long-lasting optimization. The tools for power measurement were the key to identifying dominant power consumers. Finally, ULPSEK implements hardware and software configuration with automated low power in idle and hibernation. All the efforts have made the sensor system scalable in energy consumption, which was the precondition for further investigation.

# Chapter 4

# Power Profiling Environment for Low-Power Circuits and Algorithms

Since energy consumption measurements are essential for optimizing low-power systems, a measurement procedure was developed especially for this purpose. The outstanding features of this measurement procedure are the high time resolution and bandwidth of the recordings. Therefore, the gained voltage and current signals and additional timing triggers were used to generate a set of parameters. This parameter set was given the name *power profile*. The parameters were utilized to analyze and compare the fast processes of low-power systems. The introduction of the power profiling tool is part of this chapter.

## 4.1. Introduction to Power Debugging

A typical workplace of an embedded software developer has the target system on the desk connected to the debugger. With the debugger, programs on a microcontroller run under controlled conditions. When working with microcontrollers, the debugger is a standard tool to analyze the code execution and find software bugs. Usually, developers need to pay more attention to power consumption while developing embedded software for microcontrollers.

However, this changes when developing a low-power system drastically. Monitoring power consumption while developing low-power systems is a crucial task. Every small change in the software can enormously impact the overall power consumption. The term *power bug* can be used when the system delivers wanted behavior while the power consumption increases unexpectedly. Power bugs are not just limited to software. Power bugs may likewise occur in hardware or, most likely, in the combination of hardware and software. Furthermore, the term *power debugging* expresses methods for developing hardware or software while searching for power bugs.

Using a standard multimeter for current measurement is a starting point for power debugging. However, low-power systems especially have high power consumption dynamics and irregular current changes. Reading the changing values using a standard multimeter from its display will likely be inaccurate. Some specialized companies have addressed this issue by extending debugging tools with power debugging features. These tools not only measure the average consumption, but they additionally record the dynamics of the current. However, many of those tools were not available at the beginning of this thesis. Therefore, this work contributed a method to address this gap of profiling power consumption.

## 4.2. Reviewing Power Debugging Tools

As power debugging gained popularity in the last years, new solutions were introduced on the market addressing the workflow of power debugging engineers. An overview of tools is given in table 4.1. Meanwhile, almost every provider of debugging tools for embedded systems has some power debugger in their product portfolio. The majority of these tools were introduced to the market during this work.

*Silicon Labs energyAware* was the first evaluated tool. It comes built into the evaluation board together with the available debugger. The built-in current probe has two amplifiers for dual range measurement: high and low currents. The *ARM ULINKplus* is a similar tool. It has a single range, but it also has multiple inputs for monitoring other signals. *IAR I-jet* is a debugger with a single-range power probe providing a higher sampling rate. This tool can be combined with *IAR I-scope* for additional power monitoring peripherals and multiple digital inputs. The Power Profiler Kit PCA63511 is a similar tool, which comes even with triple-range current amplifiers. So far, the tools above measured the current using one, two, or three ADCs. In contrast, the tool *MSP EnergyTrace Technology* has a built-in counter which counts energy portions. A similar principle was introduced with *FAU MeasureAlot* by Hönig et al. [91]. A current mirror creates two copies of the current flow using two capacitors. One capacitor is charged at a time while the other is discharging. Once a charge level is reached, a flip-flop circuit switches the charging to the other capacitor. The frequency of the flip-flop is proportional to the measured current. The number of counts is proportional to the energy. *FAU MeasureAlot* is not available on the market. Therefore, this tool had to be rebuilt from open-source documents.

This capacitor-charging principle was also implemented by Embedded Microprocessor Benchmark Consortium (EEMBC) an organization formed in 1997. Usually, EEMBC was known for defining *CoreMark*, giving a better measure for qualifying processor cores compared to *Dhrystone*. *Dhrystone* was introduced by Weicker in 1984 [206] and has been used until today for qualifying processor cores. *CoreMark* addressed some issues [55] along with *Dhrystone* benchmark allowing users to make quick comparisons between processor cores. However, the community questioned the applicability of *CoreMark* to low-power microcontrollers. A major argument was, *CoreMark* considered only the always-on operation of a microcontroller, while low-power microcontrollers take advantage of low-power modes and quick transitions between them. Thus, the EEMBC response to the embedded community was the *ULPBench* [19] – a new standard for ultra-low-power embedded systems. *EEMBC Energy Monitor* is a tool implementing *ULPBench* which provides the measure *ULPMark*. It defines software that must be programmed into the microcontroller under investigation. It also defines circumstances like ambient temperature and a fixed supply voltage of 3 V. Another recently introduced tool was the X-NUCLEO-LPM01A by STmicroelectronics, which also implements *ULPBench* with a dual range measurement principle. X-NUCLEO-LPM01A can be used as a power monitor and has built-in *ULPBench* simultaneously.

Most of the tools mentioned above were evaluated in depth. All these tools fit the requirements for general-purpose power debugging. The price was acceptable and the size was small enough to fit perfectly at a regular developer desk. However, these tools differ in detail in features, including the hardware properties and the respective software. Two main differentiation aspects are the bandwidth and the sampling rate. The bandwidth is directly related to the corner frequency of the current sensing probe, encompassing the probe itself as well as all analog filter and amplifier stages. While the actual bandwidth was specified in rare cases, the sampling rate was defined for all products. All sampling rates were given as a time resolution in table 4.1. A third differentiation aspect is measurement accuracy. The term accuracy is defined depending on the context. Prenesti and Gosmaro reviewed standards and guidelines defining measurement accuracy [97]. This thesis defines accuracy as a combination of precision and trueness. In table 4.1, the accuracy was given when available in the datasheet, other publications, or investigated by contacting product support. Besides these tools, which may be used for regular power debugging problems, some problems require specialized tools. Such a problem was the monitoring of fast changes in the measured current. Monitoring fast changes required a resolution in less than 1 µs for understanding the underlying processes. Besides this, the current signal analysis was previously time-consuming and repeated hundreds of times while optimizing the system under investigation. Therefore, the motivation was to automate the signal analysis of the gained data in Matlab. In this work, the procedure was named Power Profiling. Power profiling addresses a specialized domain of low-power problems which will be explained in the next section.

Before finishing this overview of the tools, it may be interesting to note what has changed since the measurements for this thesis were completed. If the measurements had been done previously and the equipment had been available earlier, the Keysight CX3324A, together with the CX1102A dual-range current probe, would have been used. Additionally, the probe CX1102A also has wide bandwidth for high-resolution power profiles. This equipment comes with evaluation software, which can not replace the capabilities developed in Matlab for this thesis. Tests have shown that this tool could potentially replace the oscilloscope and probe used for this thesis. However, the Keysight CX3324A and the probe CX1102A also have the highest price in the list of reviewed tools. A recent release and far more affordable product is the Qoitech Otii Arc. It comes with software for data evaluation and export functions. The bandwidth and the resolution of the Otii Arc can by far not reach the Keysight CX3324A and the setup used for measurements in this thesis. With recent tools, the Otii Arc is the tool of choice for the most power profiling tasks, while the Keysight high-end tool set should be chosen for high-resolution investigations as performed for this thesis.

**Table 4.1.:** Comparison of power debugging tools for embedded systems reviewed for this thesis. The tools were compared in time resolution, accuracy, and price. The category is an attempt to differentiate the measurement methods.

| Name of the Tool | Time Resolution | Approx. Accuracy | Approx. Price | Category |
|---|---:|---:|---:|---:|
| Power profile method developed for this thesis | 400 ns | 1% | € 12k | Multiple steps |
| ARM ULINKplus | 100 µs | 5% | € 200 | Single range |
| IAR I-jet | 5 µs | n.a. | € 500 | Single range |
| IAR I-scope with I-jet | 5 µs | n.a. | € 1k | Single range |
| Silicon Labs energyAware profiler | 160 µs | 1 µA | € 30 | Dual range |
| X-NUCLEO-LPM01A, ULPBench | 313 ns | ±2 % | € 65 | Dual range |
| Keysight CX3324A with CX1102A | 1 ns | 1.4%[1] | € 40k | Dual range |
| Power Profiler Kit PCA63511 | 13 µs | ±15 % | € 80 | Triple range |
| Qoitech Otii Arc | 250 µs | ±0.1 % | € 700 | Multirange |
| MSP EnergyTrace Technology | 100 µs | 1% | € 20 | Counter |
| FAU MeasureAlot [91] | 6 ns | n.a. | n.a. | Counter |
| EMBC Energy Monitor, ULPBench | 250 µs | 2% | € 100 | Counter |

## 4.3. Introducing the Power Profiling Method

The idea for developing this method came from the need to monitor fast dynamics in the process of the low-power systems investigated for this thesis. The main difference to other tools is a higher bandwidth to see more details in the voltage and current curves. The second part of this method is post-processing. The post-processing is executed in Matlab to automatically gain power and timing parameters from the recorded data. This method requires some preconditions, preparation of the investigated system, and additional measurements for systematic error correction.

## 4.4. Power Profiling Procedure

The procedure introduced here requires the following steps:

1. Set all settings of the measurement equipment according to a checklist.
2. Note the ambient temperature.
3. Note the system voltage.
4. Note the compiler version.
5. Note the compiler settings.
6. Set the microcontroller into permanent low-power mode by flashing a modified firmware.
7. Disconnect all cables where current could leak e.g. microcontroller programming tool (debugger).
8. During measurement, do not touch any part of the system to avoid current leakage, potential ESD events, and other effects that potentially destroy the measurement accuracy and, therefore, the measurement result.
9. Measure and document the current in low-power mode using a dedicated multimeter.
10. Prepare the software for giving additional timing information over free General Purpose Input/Outputs (GPIOs) at the microcontroller.
11. Prepare the microcontroller for repeating the process under investigation endlessly.
12. Record voltage, current, and two timing channels using an oscilloscope with the differential probe.
13. Measure and note the average current using a multimeter as a second reference measure.
14. Transfer the recorded oscilloscope data from the oscilloscope to a computer with Matlab.
15. Enter all additionally gained reference data into Matlab.

16. Start Matlab for deriving parameters from the recorded data by generating tables and plots with automatically set markers.

17. Change one parameter e.g. sampling rate, clock frequency, algorithm type and repeat the procedure.

Once the data was collected for different cases, interpretation of the gained results follows.

## 4.5. Power Profiling Method in Detail

A principal block diagram of the overall measurement setup is shown in figure 4.1. During the measurements, the power supply on the left was used as the system's battery replacement. Following the path from the power supply, it was possible to insert a multimeter, illustrated by the switches after the power supply. The current was measured using a shunt with a nominal resistance of 1 Ω. A metal film resistor was used as a shunt. The actual resistance was measured with a multimeter and noted for further calculations. A Tektronix ADA400 differential pre-amplifier was used to measure the voltage drop at the shunt caused by the proportional current through the shunt. The bandwidth of the ADA400 was set to the maximum, which was 1 MHz [167]. The differential amplifier ADA400 was set to Direct Current (DC) mode. The ADA400 has a potentiometer for offset adjustment. The offset was compensated by adjusting the ADA400 as far as possible to zero before the start of each measurement. The thermal noise current, according to Johnson [219] was determined to be at 41 nA at a resistance of 1 Ω, with bandwidth 1 MHz, at a temperature of 25 °C. However, the RMS noise of the ADA400 differential probe was more relevant. It was determined to be at 30 µV with a gain of 100 and bandwidth of 1 MHz according to the manual [167]. Thus, the total RMS noise of the current signal can be estimated with 30 µV/Ω. Finally, 30 µV/Ω noise voltage at a resistor with 1 Ω leads to 30 µA current noise.

The output of the differential amplifier was connected to a Tektronix TDS5054B oscilloscope. The oscilloscope was set to a sampling rate of 2.5 MSa/s. As the investigated process was forced to be repetitive, the advantage of the curve averaging function of the oscilloscope was taken. Averaging of $N_{avg} = 128$ waveforms increased the signal-to-noise ratio by a factor of $\sqrt{N_{avg}} = 11.3$, which is an increase of the SNR of $20log_{10}(\sqrt{N_{avg}}) =$ 21 dB. The TDS5054B contains an ADC with a resolution of 8 bit. Thus, the increase of SNR provides additional 3.5 bits according to equation 4.1. With this, the theoretical resolution due to averaging was increased to 11.5 bits, giving a resolution of 7 µA for 20 mA total range.

$$N_{bits} = \frac{1}{2} log_2(N_{avg}) \tag{4.1}$$

Reference measurements were made using a Metrahit 26 S multimeter. This multimeter was shipped yearly for calibration to an authorized laboratory to check the accuracy specified in the corresponding datasheet. The first measurement was taken by the multimeter to estimate the systematic error caused by the offset of the differential amplifier ADA400. Therefore, for a given range from 500 nA to 3 µA, the multimeter has an overall accuracy of ±35 nA according to the datasheet. This value was measured while the system was set permanently in low-power mode. This value was noted for later evaluation in Matlab.

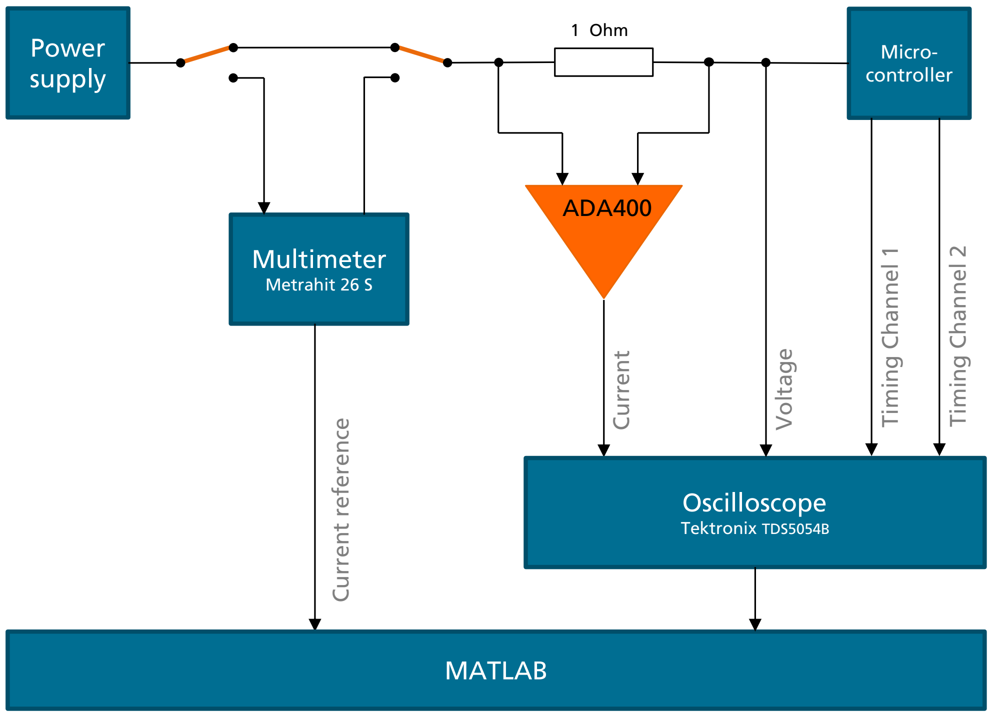


**Figure 4.1.:** Measurement setup for recording microcontroller current, voltage, and timing events at 2.5 MSa/s using an oscilloscope and a precision multimeter as reference.

A second reference measurement was taken for later verification of the average current allowing control of the readings gained by the oscilloscope. For a given range from 15 µA to 2 mA, the multimeter has an accuracy of ±2.5 µA. During the recording of the power curve, the multimeter was removed from the power supply to avoid the error caused by the current flow into the multimeter.

## 4.6. Bandwidth Optimization

The problem was that the current curve was smoothed. The bandwidth of the setup so far was estimated to 1 MHz limited by the ADA400 differential probe. A further limitation was the measurement resistor value, combined with the decoupling capacitors placed close to the microcontroller. Both together form a low-pass filter with a corner frequency of $f_c$.

$$f_c = \frac{1}{2\pi RC} \tag{4.2}$$

With R = 1 Ω and C = 1600 nF the bandwidth was approximately 100 kHz. Three solutions were proposed: (1) reducing the resistance (R); (2) measuring the current after the capacitors; (3) reducing the capacitance (C). The RMS noise of the measured value was previously estimated to 30 µV/Ω caused by the ADA400 differential probe. With this, decreasing R by a factor of 10 would increase the noise to 300 µA. Going further down with R, the battery's internal resistance becomes relevant. The first solution was dropped because of these two effects.

The second option was to measure between the capacitors and the microcontroller. In general, decoupling capacitors are required along with integrated circuits. Decoupling capacitors are local energy buffers for the integrated circuit. Manufacturers of integrated circuits specify for every input a capacitance required to provide this input with energy when needed for the worst case. A straightforward solution to handle this would be to measure behind the capacitor, directly right after the capacitor at the microcontroller power supply input. There, the full bandwidth can be seen. Unfortunately, a microcontroller has multiple power supply inputs. Therefore, a separate current measurement channel would be necessary for each input, including multiple differential amplifiers. Furthermore, there was only one amplifier of this kind. Thus, this solution was dropped.

The third solution to overcome this problem was to reduce the amount of decoupling capacitors. As mentioned, the capacitors by the manufacturer of the IC are recommended for the worst case. Under restricted circumstances, the number of total capacitors which allows stable operation should be less. Restricted circumstances can be given by loading a specific software with defined dynamics. Special software on the microcontroller executes a defined process during a measurement that repetitively changes low-power states. With this, the capacity was reduced from 1600 nF to 200 nF. With $C = 200\,\text{nF}$ in equation 4.2 the bandwidth was approximately 796 kHz. This was a practicable solution for gaining details in the current recordings while keeping the microcontroller stable. Figure 4.2 shows the effect, which brings more clarity of details to the recorded current curve compared to the smoothed recording in a usual setup.

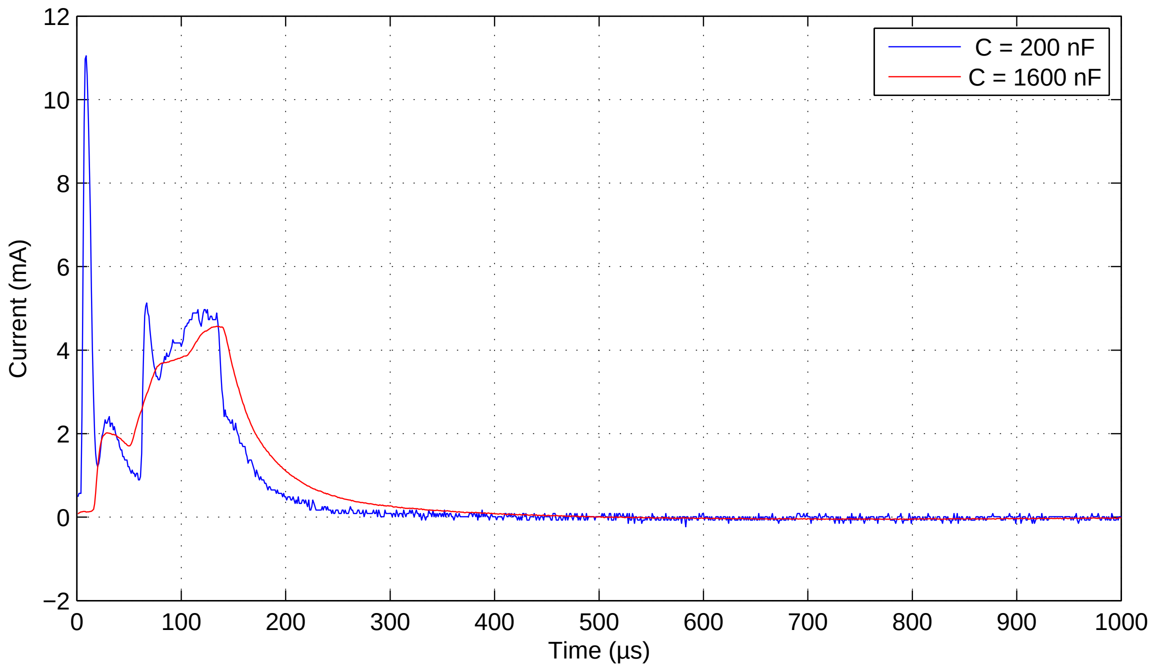


**Figure 4.2.:** This graph shows two recordings of the microcontroller current with the recommended 1600 nF and the reduced 200 nF set of decoupling capacitors. More details can be seen with 200 nF while the current curve is smoothed in the recommended setup when using a total of 1600 nF. The bandwidth was determined to be at 100 kHz or 796 kHz, respectively.

Additionally, to monitor the profiling process quality, the Matlab script evaluates the supply voltage's maximum drop, recorded along with the current curve at the same sampling rate.

## 4.7. Automated Data Analysis in Matlab

Once the data was imported into Matlab an analysis script was executed. For each dataset, power profiles are determined using two distinct methods. The first method involves generating a graphical representation, as shown in Figure 4.1, which includes automatically highlighted regions. The second method provides a table of performance parameters. This tabular view was utilized for comparing parameters during experiments. However, the table contains a large set of parameters. Therefore, this table will be shown only once exemplary with Figure 4.3 without discussing the exemplary select numbers of the parameters. This table includes groups of parameters as follows.

- Microcontroller identification
- Compiler version and configuration
- Core clock type and frequency
- Duration of each time segment
- The duty cycle between activity (the first five segments) and sleep mode
- The absolute energy cost for each time segment
- The relative (percentage) energy cost for each time segment
- Average current, voltage, and power over one sampling period without sleep mode
- Average current, voltage, and power over one sampling period, including sleep mode
- The calculated number of cycles to stabilize the clock source

The following additional parameters are for validation purposes of the measurement process.

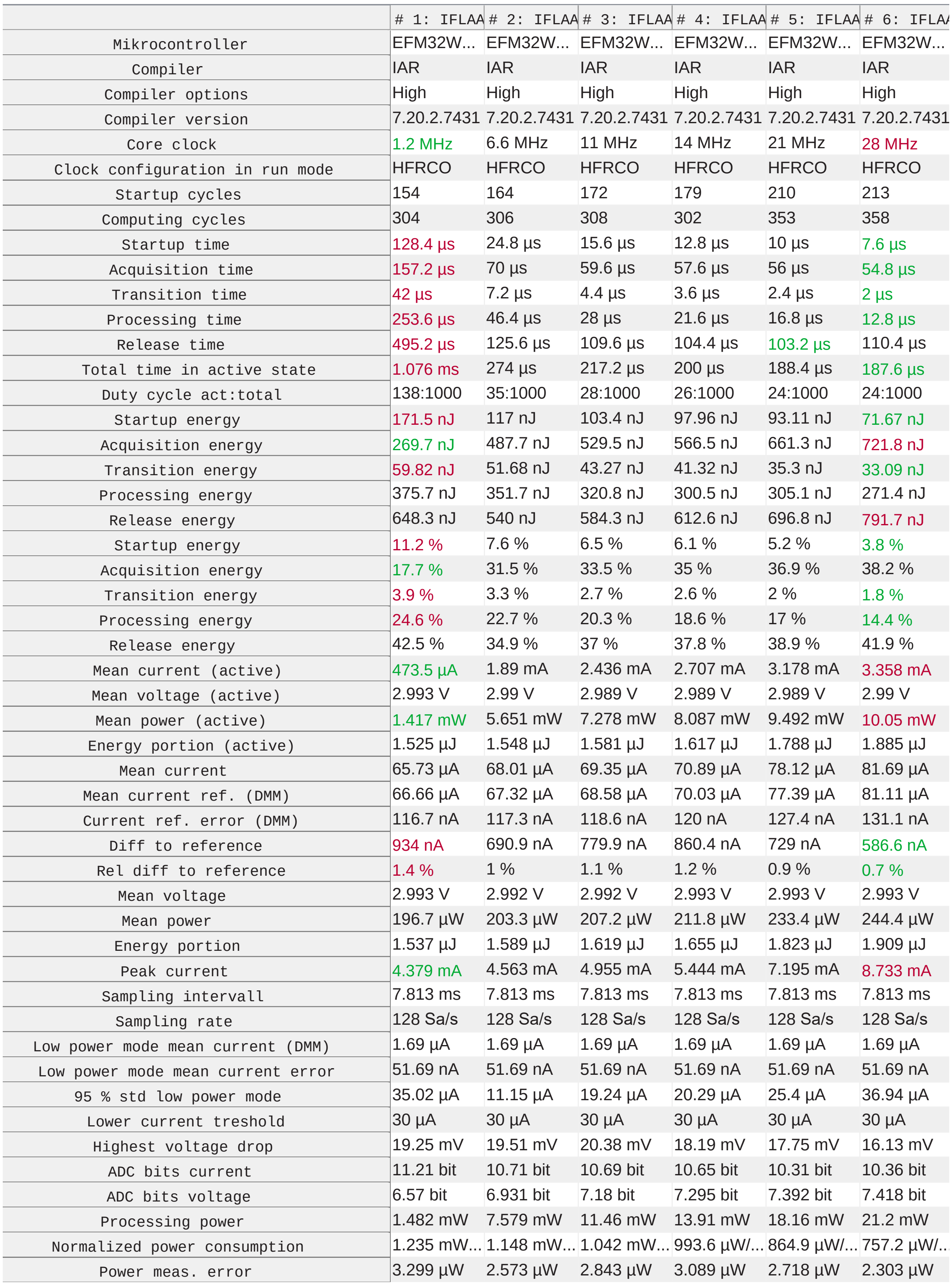
- Reference measurements and comparison to the estimated average current with alert function if the difference is out of range
- Peak current
- Sampling interval and rate
- Low power mode current reference and estimated error after correction due to floating point inaccuracy
- Highest voltage drop during activity with alert on high voltage drops
- Estimation of the resulting information in bits to check if the averaging mode of the oscilloscope has been used or not

- The mean power during processing
- The normalized processing power in Milliwatts per Megahertz
- The power measurement error with alert function on exceeded errors

Instead of showing the entire table for each experiment, selected parameters will be discussed later in this thesis in the experimental chapters 5 and 6.
The six time regions were defined for segmenting the data automatically:

1. **Start-up:** Start-up procedure after low-power mode; it ends when the core clock was stable and right after the first command was executed.
2. **Acquisition:** During data acquisition from sensors.
3. **Transition:** Period after the end of acquisition and until processing of sensor data (for ULPSEK negligible, in general relevant).
4. **Processing:** While the microcontroller is processing the acquired data.
5. **Release:** Time interval while going to sleep, right after the command for entering the low-power mode, lasts until consumption reaches the low-power level.
6. **Low-power:** While the microcontroller is in the low-power state.

Figure 4.4 shows an example recording of the current curve of a microcontroller with automatically highlighted regions. The segmenting was achieved by evaluating the timing trigger, which was recorded on two additional channels with voltage and current. The first trigger marks the acquisition's beginning and end, on rising and falling edge, respectively. The second trigger marks the processing's beginning and end on the rising and falling edge, respectively. The cycle's start and end can be determined by thresholding the current. With this, all six time segments can be determined and the plot with segmentation can be generated. This set of parameters was used as a fingerprint for the process cycle of the low-power modes of the evaluated system.



| | # 1: IFLAA | # 2: IFLAA | # 3: IFLAA | # 4: IFLAA | # 5: IFLAA | # 6: IFLA |
|---|---|---|---|---|---|---|
| Mikrocontroller | EFM32W... | EFM32W... | EFM32W... | EFM32W... | EFM32W... | EFM32W... |
| Compiler | IAR | IAR | IAR | IAR | IAR | IAR |
| Compiler options | High | High | High | High | High | High |
| Compiler version | 7.20.2.7431 | 7.20.2.7431 | 7.20.2.7431 | 7.20.2.7431 | 7.20.2.7431 | 7.20.2.7431 |
| Core clock | 1.2 MHz | 6.6 MHz | 11 MHz | 14 MHz | 21 MHz | 28 MHz |
| Clock configuration in run mode | HFRCO | HFRCO | HFRCO | HFRCO | HFRCO | HFRCO |
| Startup cycles | 154 | 164 | 172 | 179 | 210 | 213 |
| Computing cycles | 304 | 306 | 308 | 302 | 353 | 358 |
| Startup time | 128.4 µs | 24.8 µs | 15.6 µs | 12.8 µs | 10 µs | 7.6 µs |
| Acquisition time | 157.2 µs | 70 µs | 59.6 µs | 57.6 µs | 56 µs | 54.8 µs |
| Transition time | 42 µs | 7.2 µs | 4.4 µs | 3.6 µs | 2.4 µs | 2 µs |
| Processing time | 253.6 µs | 46.4 µs | 28 µs | 21.6 µs | 16.8 µs | 12.8 µs |
| Release time | 495.2 µs | 125.6 µs | 109.6 µs | 104.4 µs | 103.2 µs | 110.4 µs |
| Total time in active state | 1.076 ms | 274 µs | 217.2 µs | 200 µs | 188.4 µs | 187.6 µs |
| Duty cycle act:total | 138:1000 | 35:1000 | 28:1000 | 26:1000 | 24:1000 | 24:1000 |
| Startup energy | 171.5 nJ | 117 nJ | 103.4 nJ | 97.96 nJ | 93.11 nJ | 71.67 nJ |
| Acquisition energy | 269.7 nJ | 487.7 nJ | 529.5 nJ | 566.5 nJ | 661.3 nJ | 721.8 nJ |
| Transition energy | 59.82 nJ | 51.68 nJ | 43.27 nJ | 41.32 nJ | 35.3 nJ | 33.09 nJ |
| Processing energy | 375.7 nJ | 351.7 nJ | 320.8 nJ | 300.5 nJ | 305.1 nJ | 271.4 nJ |
| Release energy | 648.3 nJ | 540 nJ | 584.3 nJ | 612.6 nJ | 696.8 nJ | 791.7 nJ |
| Startup energy | 11.2 % | 7.6 % | 6.5 % | 6.1 % | 5.2 % | 3.8 % |
| Acquisition energy | 17.7 % | 31.5 % | 33.5 % | 35 % | 36.9 % | 38.2 % |
| Transition energy | 3.9 % | 3.3 % | 2.7 % | 2.6 % | 2 % | 1.8 % |
| Processing energy | 24.6 % | 22.7 % | 20.3 % | 18.6 % | 17 % | 14.4 % |
| Release energy | 42.5 % | 34.9 % | 37 % | 37.8 % | 38.9 % | 41.9 % |
| Mean current (active) | 473.5 µA | 1.89 mA | 2.436 mA | 2.707 mA | 3.178 mA | 3.358 mA |
| Mean voltage (active) | 2.993 V | 2.99 V | 2.989 V | 2.989 V | 2.989 V | 2.99 V |
| Mean power (active) | 1.417 mW | 5.651 mW | 7.278 mW | 8.087 mW | 9.492 mW | 10.05 mW |
| Energy portion (active) | 1.525 µJ | 1.548 µJ | 1.581 µJ | 1.617 µJ | 1.788 µJ | 1.885 µJ |
| Mean current | 65.73 µA | 68.01 µA | 69.35 µA | 70.89 µA | 78.12 µA | 81.69 µA |
| Mean current ref. (DMM) | 66.66 µA | 67.32 µA | 68.58 µA | 70.03 µA | 77.39 µA | 81.11 µA |
| Current ref. error (DMM) | 116.7 nA | 117.3 nA | 118.6 nA | 120 nA | 127.4 nA | 131.1 nA |
| Diff to reference | 934 nA | 690.9 nA | 779.9 nA | 860.4 nA | 729 nA | 586.6 nA |
| Rel diff to reference | 1.4 % | 1 % | 1.1 % | 1.2 % | 0.9 % | 0.7 % |
| Mean voltage | 2.993 V | 2.992 V | 2.992 V | 2.993 V | 2.993 V | 2.993 V |
| Mean power | 196.7 µW | 203.3 µW | 207.2 µW | 211.8 µW | 233.4 µW | 244.4 µW |
| Energy portion | 1.537 µJ | 1.589 µJ | 1.619 µJ | 1.655 µJ | 1.823 µJ | 1.909 µJ |
| Peak current | 4.379 mA | 4.563 mA | 4.955 mA | 5.444 mA | 7.195 mA | 8.733 mA |
| Sampling intervall | 7.813 ms | 7.813 ms | 7.813 ms | 7.813 ms | 7.813 ms | 7.813 ms |
| Sampling rate | 128 Sa/s | 128 Sa/s | 128 Sa/s | 128 Sa/s | 128 Sa/s | 128 Sa/s |
| Low power mode mean current (DMM) | 1.69 µA | 1.69 µA | 1.69 µA | 1.69 µA | 1.69 µA | 1.69 µA |
| Low power mode mean current error | 51.69 nA | 51.69 nA | 51.69 nA | 51.69 nA | 51.69 nA | 51.69 nA |
| 95 % std low power mode | 35.02 µA | 11.15 µA | 19.24 µA | 20.29 µA | 25.4 µA | 36.94 µA |
| Lower current treshold | 30 µA | 30 µA | 30 µA | 30 µA | 30 µA | 30 µA |
| Highest voltage drop | 19.25 mV | 19.51 mV | 20.38 mV | 18.19 mV | 17.75 mV | 16.13 mV |
| ADC bits current | 11.21 bit | 10.71 bit | 10.69 bit | 10.65 bit | 10.31 bit | 10.36 bit |
| ADC bits voltage | 6.57 bit | 6.931 bit | 7.18 bit | 7.295 bit | 7.392 bit | 7.418 bit |
| Processing power | 1.482 mW | 7.579 mW | 11.46 mW | 13.91 mW | 18.16 mW | 21.2 mW |
| Normalized power consumption | 1.235 mW... | 1.148 mW... | 1.042 mW... | 993.6 µW/... | 864.9 µW/... | 757.2 µW/.. |
| Power meas. error | 3.299 µW | 2.573 µW | 2.843 µW | 3.089 µW | 2.718 µW | 2.303 µW |

**Figure 4.3.:** This figure shows an example of six energy profiles collected in one parameter table gained from six recordings with different core clocks.

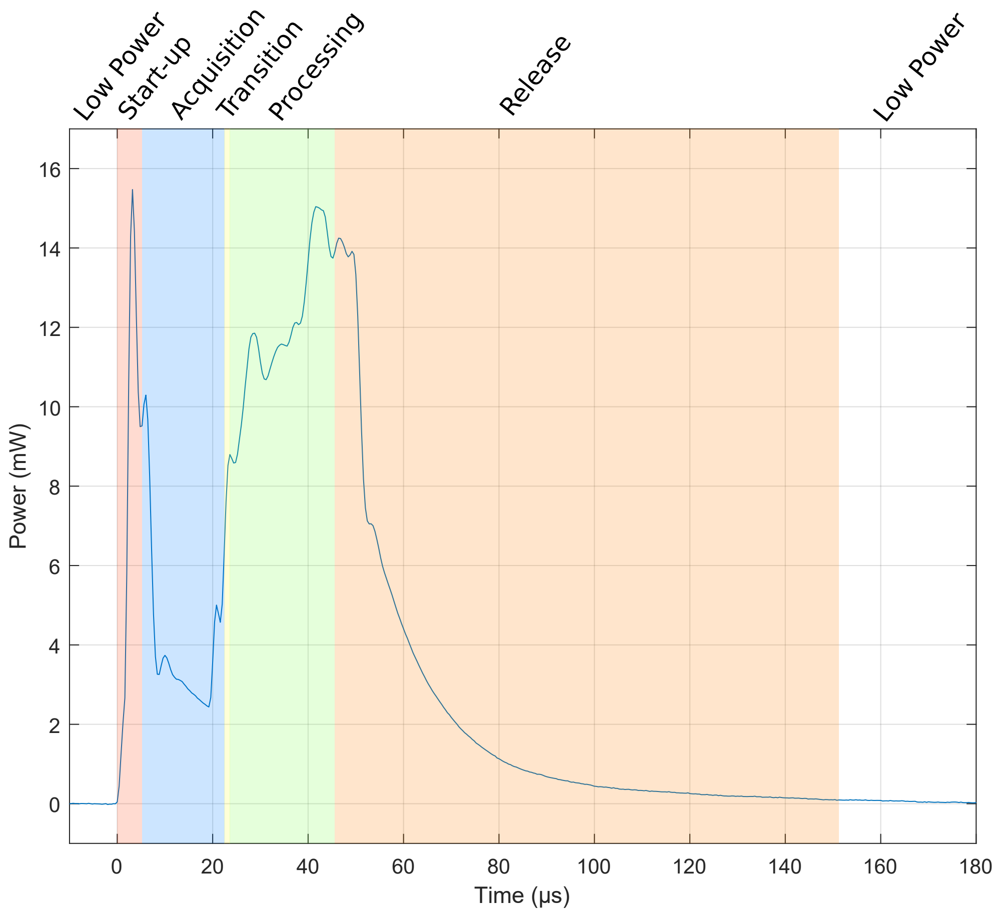


**Figure 4.4.:** Example graphical profile for the microcontroller EFM32WG990F256 at 14 MHz core clock using IAR compiler and the *Apnea* benchmark algorithm (Appendix B). Highlighted areas: 🟥 Start-up, 🟦 Acquisition, 🟨 Transition, 🟩 Processing, and 🟧 Release. The low-power state before and after a whole cycle is not highlighted. The x-axis is the time, starting with the acquisition. The y-axis shows the power in this example. Alternatively, the y-axis can be assigned to the current.

## 4.8. Chapter Summary

This power profiling method allows high-resolution recordings of the underlying process of the sensor system. The bandwidth was estimated to 700 kHz. Further, this method specializes in a subset of problems essential to ultra-low-power system design projects. As preconditions, the observed processes must be repetitive to take advantage of the average feature. For everyday power debugging, other tools with lower bandwidth and without automated analysis may be more practical. However, the power profiling method introduced here was the key to many findings in this thesis which helped optimize the sensor system's power consumption.

# Chapter 5

# Experiments with Embedded Systems

In this chapter, experiments were performed, which were essential for selecting the microcontroller for ULPSEK. This thesis used microcontrollers for signal processing and control of all system processes. Therefore, experiments were documented in this chapter, deciding for ULPSEK in selecting the microcontroller. Tobola et al. [100] is the related publication addressing the following research questions. The research questions in this chapter addressed gaps in the following domains:

1. Which microcontroller is best suitable for biomedical applications?
2. Are compilers and their settings for the microcontroller relevant to energy consumption?
3. How should the clock unit of the low-power microcontroller[1]be used optimally?
4. How does an RTOS affect the power consumption?

The challenge here was to answer all research questions experimentally by measurements and not by using datasheets.

## 5.1. Benchmark Application

The benchmark algorithm from the project "Smart Sensors A" was used for all experiments in this chapter. The algorithm was designed to detect apnea in respiratory signals and published in [77]. It qualifies for benchmarks of biomedical signal processing due to its diversity of computing steps. The algorithm covers linear and nonlinear filters, integer operations and floating point operations. Before the benchmark began, the algorithm was supplied with test data and verified for expected results to guarantee that the algorithm works appropriately with the tested microcontroller-, compiler-, and optimization-level combination.

## 5.2. Experiment: Microcontroller and Compiler Impact

In this experiment the microcontroller and compiler variations were evaluated with the benchmark algorithm.

[1]Low-power microcontrollers have an enhanced clock unit with extended configuration possibilities, which distinguishes them from mainstream microcontrollers.

### 5.2.1. Methods

In this experiment four microcontrollers were tested using the proposed benchmark algorithm. Each microcontroller was evaluated with different compilers. The proposed profiling method measured the consumption profiles for each test run. In this experiment, all core clocks were set to 14 MHz. The results are listed in Table 5.1.
Tested microcontrollers:

- MSP430F5659 (16 bit, low-power)
- EFM32ZG222F32 (ARM Cortex-M0+, 32 bit, low-power)
- EFM32GG990F1024 (ARM Cortex-M3, 32 bit)
- EFM32WG990F256 (ARM Cortex-M4F, 32 bit, DSP support, floating point unit)

ARM-C-compiler and version, followed by best-performed optimization levels:

- IAR 7.20.2.7431, High, Speed
- GCC 4.7.3, -O3
- Keil, High, Time

MSP430-C-compiler and version, followed by best-performed optimization levels:

- TI 4.3.2, -O4
- IAR 6.10.2, High, Speed
- GCC 4.8.0, -O3

Analog-to-digital converter configuration:

- MSP430: clock: 5 MHz, 2.5 V Ref., 12 Bit, sample hold time: 32 cycles
- EFM32: clock: 1, 5, or 7 MHz (constant per experiment), 2.5 V Ref., 12 Bit, acquisition time: 32 cycles

### 5.2.2. Results

Four different microcontroller architectures and four compilers were investigated. Table 5.1 shows a span between 78 µW and 2555 µW for processing an equal computing task. For the result sections two figures were selected: Figure 5.1 is showing the difference between ARM Cortex cores. Figure 5.2 is showing the difference between MSP430 compilers.

**Table 5.1.:** Average microcontroller power consumption in $\mu$W depending on microcontroller type and compiler, including individual absolute measurement error in $\mu$W.

| | TI | IAR | GCC | Keil |
|---|---|---|---|---|
| MSP430 | 443.4 $\pm$ 29.2 | 390.4 $\pm$ 8.1 | 2555 $\pm$ 29.1 | - |
| Cortex-M0+ | - | 100.5 $\pm$ 9.3 | 158.1 $\pm$ 10.1 | 94.2 $\pm$ 6.8 |
| Cortex-M3 | - | 157.1 $\pm$ 2.1 | 190.4 $\pm$ 1.1 | 152.0 $\pm$ 1.7 |
| Cortex-M4F | - | 79.0 $\pm$ 0.6 | 88.7 $\pm$ 1.8 | 78.4 $\pm$ 2.1 |

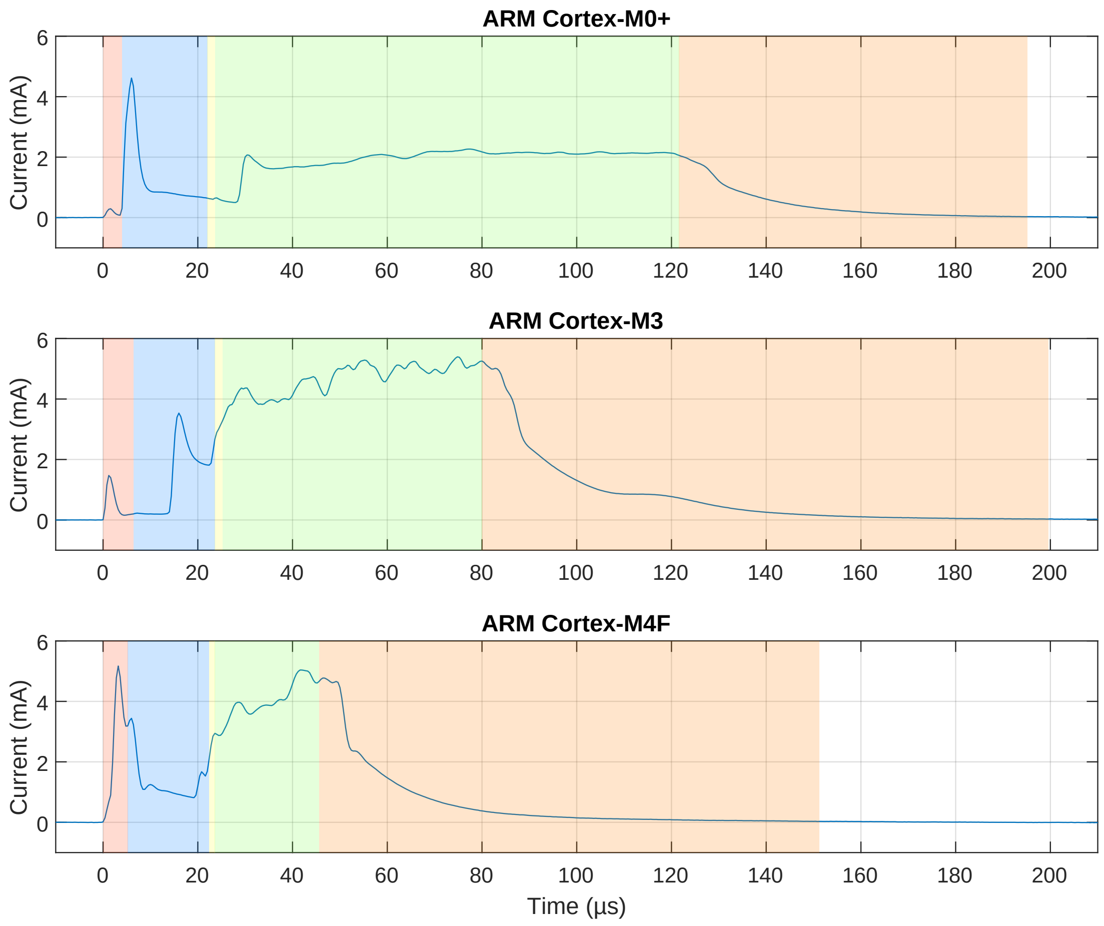


**Figure 5.1.:** This figure compares three microcontroller current recordings for processing of the benchmark task with three different ARM Cortex microcontroller cores: ARM Cortex-M0+, ARM Cortex-M3, and ARM Cortex-M4F. The IAR compiler was utilized for all three profiles presented exemplary in this figure.
Highlighted time slices according to the energy profile in chapter 4:
🟥 Start-up, 🟦 Acquisition, 🟨 Transition, 🟩 Processing, and 🟧 Release.

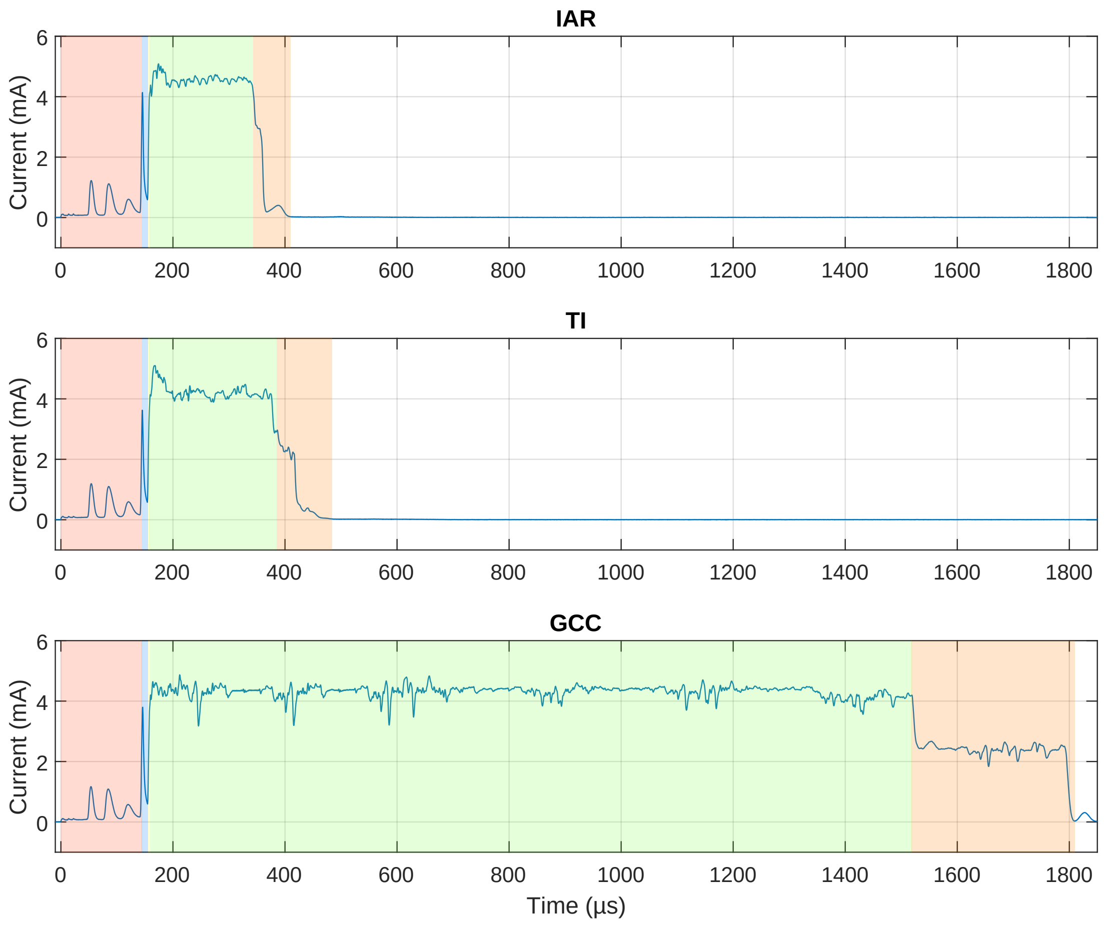


**Figure 5.2.:** This figure compares three microcontroller current recordings for processing the benchmark task with three different compilers IAR, TI, and Keil. The MSP430 microcontroller is compared with different compilers in all three selected cases. Highlighted time slices according to the energy profile in chapter 4:
🟥 Start-up, 🟦 Acquisition, 🟨 Transition, 🟩 Processing, and 🟧 Release.

### 5.2.3. Discussion

#### ARM Cortex-M4F (best case)

Of all twelve tested cases in Table 5.1 the most efficient microcontroller core was the ARM Cortex-M4F for the selected test application. For ARM Cortex-M4F, the compilers IAR and Keil delivered comparable results.

#### ARM Cortex-M0+ (for small systems)

The second best result was achieved with the ARM Cortex-M0+. Comparing only the CoreMark-Power-Ratio[2], the ARM Cortex-M0+ seemed to be the most efficient choice. However, since the floating point operations are part of the benchmark scenario, the ARM Cortex-M4F with a floating-point unit consumed significantly less power with the test application.

#### ARM Cortex-M3

ARM Cortex-M3 is not surprisingly the least efficient ARM core out of the selected range. The ARM Cortex-M3 has no DSP support and compared to ARM Cortex-M0+ less energy efficiency when comparing mW/MHz.

#### MSP430

It is worth noting that both the ARM Cortex-M core and the MSP430 microcontroller families are designed for low-power applications but stand apart in their architecture, instruction set, processing power, and typically the semiconductor process technology (measured in nanometers) used to manufacture them. The MSP430 microcontroller family, with its 16-bit architecture, is designed for ultra-low power applications, making it ideal for battery-operated and low-energy devices. In contrast, ARM Cortex-M cores are 32-bit and offer various options from low-power (M0+) to high-performance (M3, M4, M7) models, supporting more complex applications. Texas Instruments uses 130 nm ultra-low-leakage technology [117] for MSP430 to keep costs low and prioritize power savings over performance. ARM does not specify a particular manufacturing process, as ARM licenses its IP to manufacturers. The EFM32 Gecko microcontrollers with ARM Cortex-M cores (M0+, M3, M4) used for this investigation were built on a 180 nm process to optimize ultra-low-power consumption.

The MSP430 is an excellent microcontroller that has been in many professional designs for decades. However, in this particular test application with multiple sensors and signal processing, the MSP430 requires relatively more time and energy to compute and switch between the low-power states. Thus, the resulting energy consumption was higher than any of the ARM Cortex cores.

[2]CoreMark-Power-Ratio defined as CoreMark/MHz normalized by mW/MHz, which gives CoreMark/mW as core power efficiency scale.

### Microcontroler Summary

The Cortex-M4F performed best, followed by Cortex-M0+, with a large gap to Cortex-M3, leaving behind the MSP430. This order was initially different when considering standard performance parameters from datasheets regarding the power per clock cycle. The experiments revealed that computing biomedical signals on low-power systems with automatic sleep state controlled by RTOS benefits from faster computing by DSP functionality. The data has shown that for biomedical applications, the Cortex-M4F architecture is the best for multisensor and onboard computing applications. Exceptionally, for minimum computing on single sensor nodes, the Cortex-M0+ might perform better. Cortex-M3 was best in price when low power did not play a role.

### Compiler Performance

Surprisingly, the GCC compiler performed less efficiently in all table rows. The development of the GCC compiler for the MSP430 microcontroller was discontinued. The stop of the development may explain the extraordinarily long runtime for the combination MSP430 with GCC compiler. However, it confirms an exciting assumption that the impact of the compiler matters short runtime and low power, consequently. Unlike the GCC compiler for the MSP430, all compilers for the ARM microcontrollers were actively supported. So why did IAR and Keil perform better than GCC? The disassembly allows analysis of the machine code in assembly language. Hence, machine code binaries of all compilers and all ARM processors cores were disassembled and analyzed over numerous hours. Comparing the machine instructions line by line, it turned out that the commercial compilers IAR and Keil used the instruction set with relative addressing more excessively than GCC. However, this finding applies within the boundaries of the experimental setup, notably the compiler versions and the compiler settings, as declared previously. In conclusion, more frequent usage of relative addressing made this application example faster and consequently more energy efficient. All in all, the choice of the compiler matters.

## 5.3. Experiment: Operating System Impact

An RTOS provides scheduling mechanisms for handling multiple tasks. In general, handling multiple tasks on a microcontroller can be realized without having an RTOS. Also, low-power sleep modes can be implemented without an RTOS. However, if multiple tasks are planned to be handled by a microcontroller, then an RTOS should be used to schedule the workload. Therefore, the RTOS provides well-tested task scheduling mechanisms. The actual multitasking comes with the demand for synchronizing threads and handing over data from one thread to another. All of that can be implemented from scratch. However, many RTOS implementations for microcontrollers exist and all of them provide this commonly needed base functionality. Compared to operating systems known from general purpose computers and smartphones, most of the RTOS distributions are limited to this base functionality. Some RTOS distributions come with additional software packages, e.g., data protocols, encryption, and file systems. A significant difference between an RTOS and a standard operating system is minimal memory requirements and real-time capability. The real-time capability guarantees finishing a task in a defined time. Thus, many microcontroller applications take advantage of an RTOS. Consequently, there were the following questions.

- Does using an RTOS differ for low-power applications compared to general applications when not using low-power features?
- How does an RTOS impact power consumption in low-power applications?

### 5.3.1. Methods

The following experimental setup was prepared to answer the above research questions. The first measurement was prepared for interrupt-driven operation while sampling a test signal without using an RTOS. The second measurement utilized RTOS A in default mode, then in tickless mode, as described in [138]. A fourth experiment quantified power consumption when using a semaphore. Semaphore are used for synchronization between tasks of an RTOS. The semaphore triggered the execution of processing code in a task instead of doing this in the interrupt handler. Finally, the results were compared against another RTOS distribution, here RTOS B. The names of the used RTOS were anonymized because of a statement regarding benchmarking in the license. This statement should not apply to the following investigation because the measurements are not intended to benchmark RTOS A versus RTOS B. Instead, the data should show how an RTOS (A) affects power consumption in the proposed setup and confirm the findings with a second RTOS (B).

### 5.3.2. Results

The results from this experiment are listed in table 5.2. This table compares the transition and release times, which have the highest impact on the average power consumption in this experiment.

**Table 5.2.:** This table shows the parameters from the power profile with the main impact on the average power consumption.

| | Transition time | Release time | Average power consumption |
|---|---:|---:|---:|
| Without RTOS, interrupt driven | 1.2 µs | 54.8 µs | 76.8 µW |
| RTOS A, interrupt driven | 3.6 µs | 69.6 µs | 579.6 µW |
| RTOS A, tickless, interrupt driven | 1.2 µs | 85.6 µs | 132.0 µW |
| RTOS A, tickless, using semaphores | 34.4 µs | 133.6 µs | 304.6 µW |
| RTOS B, tickless, interrupt driven | 3.6 µs | 89.6 µs | 156.1 µW |

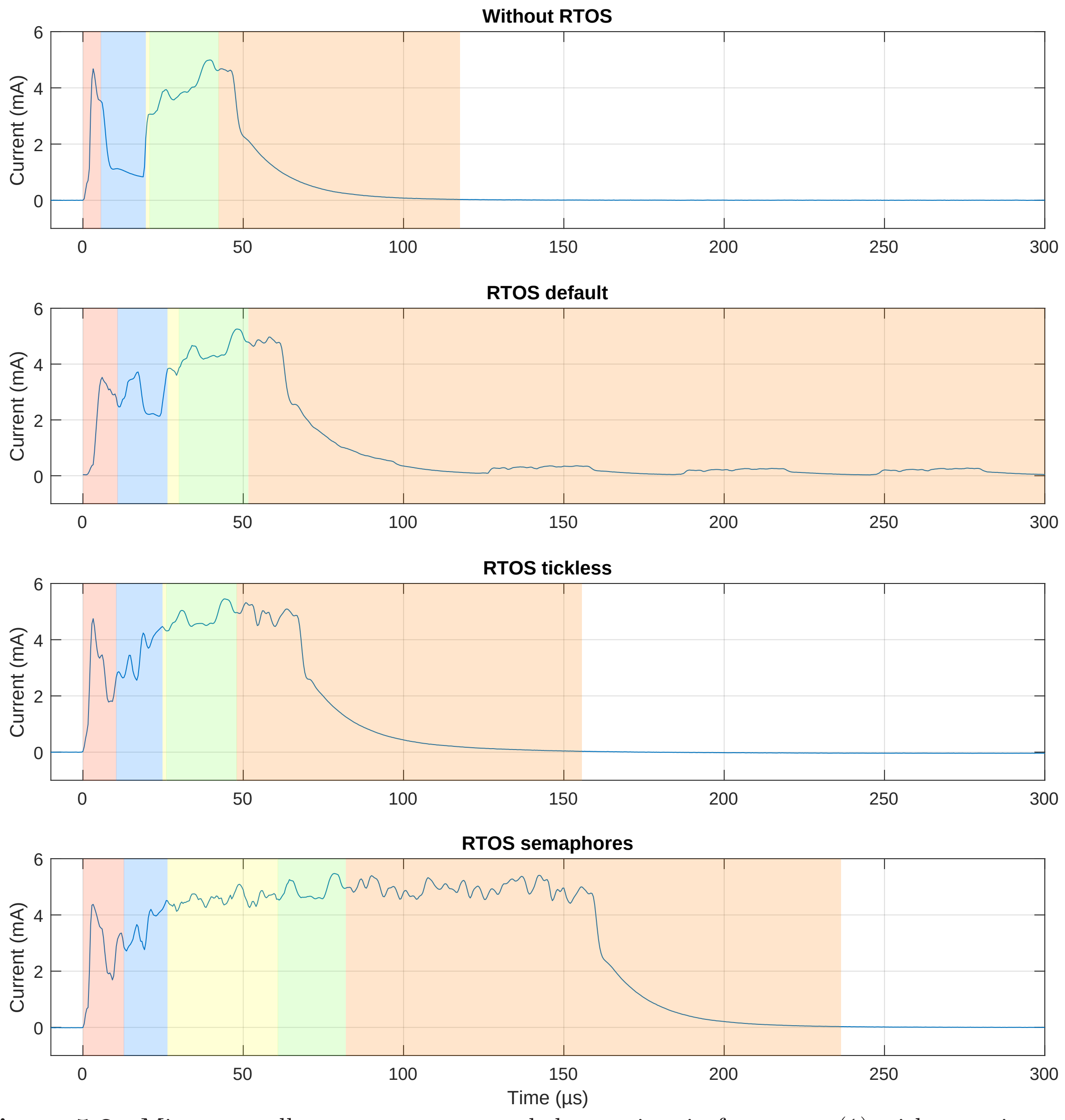


**Figure 5.3.:** Microcontroller current was recorded over time in four cases (1) without using an RTOS, (2) using RTOS A with default settings, (3) using an RTOS A in tickless mode, (4) using an RTOS A with semaphores. Since RTOS B performed similarly, the plots stand exemplary for RTOS A.
Highlighted time slices according to the energy profile in chapter 4:
🟥 Start-up, 🟦 Acquisition, 🟨 Transition, 🟩 Processing, and 🟧 Release.

### 5.3.3. Discussion

An RTOS is a critical software component of a microcontroller. When using semaphores, notable delays will be caused by the RTOS. These delays caused remarkable waste of energy while holding the microcontroller active. A solution was proposed here to overcome the waste of energy by directly computing smaller workloads in the interrupt handler. The system should be programmed to automatically switch to a low-power state when the core is not required to compute anything. In contrast to a regular RTOS usage, the system tick causes a waste of energy by waking up the microcontroller at every tick. The solution was activating a so-called tickles mode is mandatory on a low-power system. With this low-power setup, a second RTOS distribution was investigated. The second investigated RTOS confirmed almost equal power consumption results. Further, in figure 5.5 high release energy was noticed compared to the processing energy. Here is space left for optimization of the RTOS configuration and RTOS implementation.

To answer the research questions, the RTOS must be set up, particularly for low-power applications. In particular, the tickless mode is mandatory. Even so, the RTOS requires additional energy, which was 72 % more within the used benchmark.

## 5.4. Experiment: Clock System Impact

The clock system is the heart of a microcontroller. In particular low-power microcontrollers have a rich clock unit, including low-power clock routing options and clock gating.

### 5.4.1. Methods

In the last experiment, the EFM32WG passed with the best results. The microcontroller EFM32WG was reviewed for this experiment at different clock frequencies. The EFM32WG has an integrated resistor-capacitor (RC)-oscillator with six selectable frequencies between 1 MHz and 28 MHz. This clock was routed directly to the processor core. A crystal oscillator with an external crystal was also available at 48 MHz. This frequency was the maximum limit of the microcontroller. For this experiment, the profiling method was used to record and analyze the sampling, processing and low-power mode transitions.

### 5.4.2. Results

The results are listed in table 5.3. For every clock frequency, the start-up energy, processing energy, and the sum of both were listed. The energy for acquiring samples and the transition energy were irrelevant in this experiment. Figure 5.5 and figure 5.4 are the corresponding profiler plots for this experiment.

**Table 5.3.:** Energy portions to process one sample of data, start-up time, and transition time depending on the core clock frequency and oscillator type (RC: resistor-capacitor oscillator; XO: crystal oscillator).

| Core clock | Start-up energy | Processing energy | Sum |
|---|---|---|---|
| 1.2 MHz RC | 171 nJ | 367 nJ | 538 nJ |
| 6.6 MHz RC | 117 nJ | 352 nJ | 469 nJ |
| 11.0 MHz RC | 103 nJ | 321 nJ | 424 nJ |
| 14.0 MHz RC | 98 nJ | 300 nJ | 398 nJ |
| 21.0 MHz RC | 93 nJ | 305 nJ | 398 nJ |
| 28.0 MHz RC | 72 nJ | 271 nJ | 343 nJ |
| 48.0 MHz XO | 4768 nJ | 296 nJ | 5064 nJ |

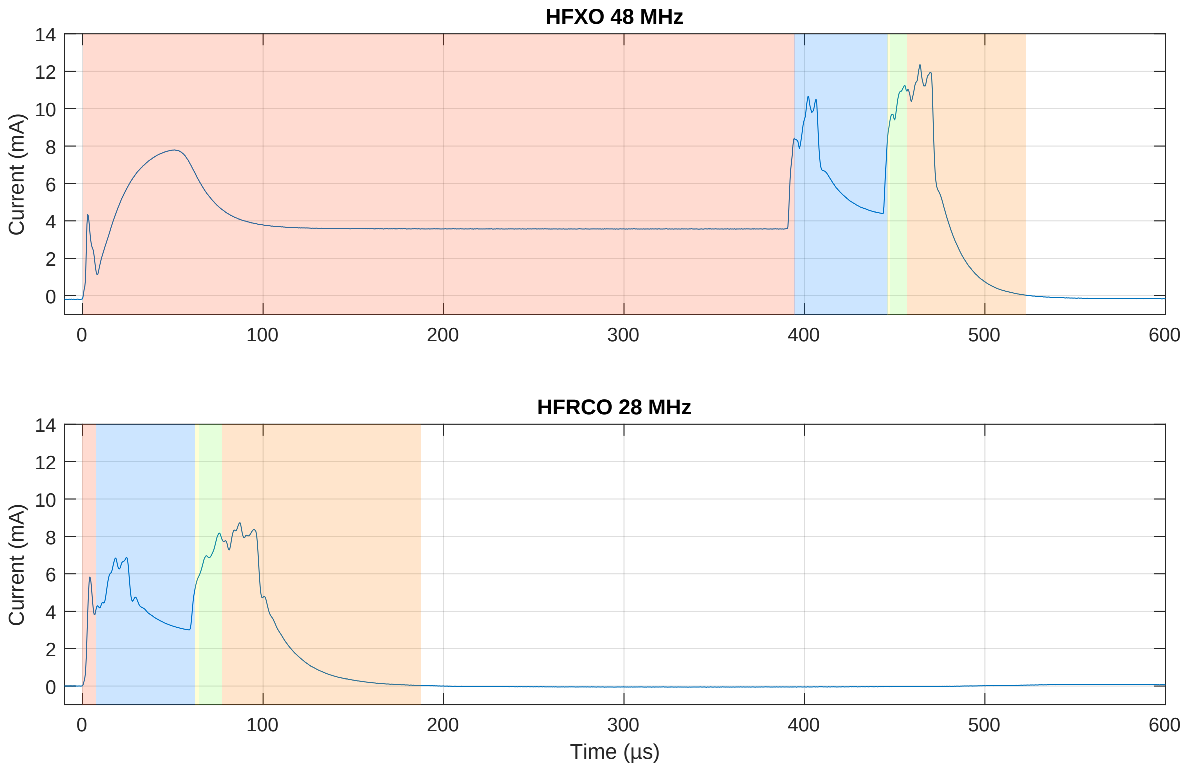


**Figure 5.4.:** Microcontroller current recorded over time at 48 MHz using the crystal oscillator compared to the highest clock using the RC-oscillator at 28 MHz.
Highlighted time slices according to the energy profile in chapter 4:
🟥 Start-up, 🟦 Acquisition, 🟨 Transition, 🟩 Processing, and 🟧 Release.

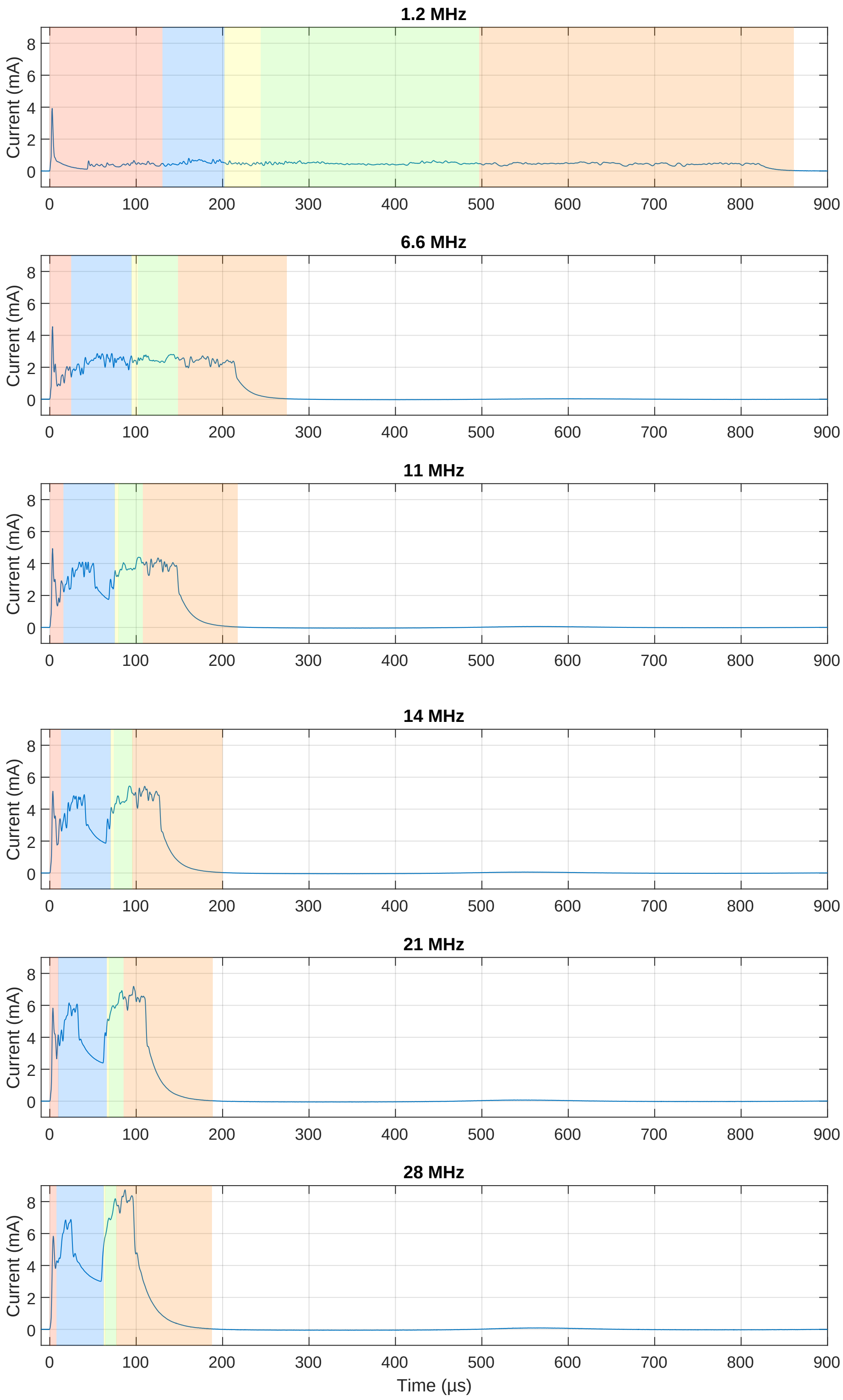


**Figure 5.5.:** Microcontroller current is recorded over time at different RC oscillator clocks for generating the microcontroller core clock starting at 1.2 MHz up to 28 MHz. Highlighted time slices according to the energy profile in chapter 4: 🟥 Start-up, 🟦 Acquisition, 🟨 Transition, 🟩 Processing, and 🟧 Release.

### 5.4.3. Discussion

#### Release Energy

In figure 5.5 one may notice the Release Energy was high compared to the Processing Energy. This finding was already mentioned in the previous experiments. The reason is the influence of the operating system, even if not using semaphores. The required amount of the Release Energy is an issue for further optimization. However, with the profiling method, the Release Energy can be separated from the actual values of interest.

#### Most Power-Efficient Clock

The actual values of interest are the energy required for processing and the energy required for start-up. In Table 5.3 both are listed together with their sum for every investigated clock setup. According to the table, the most efficient sum is at 28 MHz. Also, the processing energy itself is best at 28 MHz. While looking at the values for processing energy, the values started falling with the rising clock frequency. Then, this trend interrupts at 21 MHz. This relationship was explained in the manual for the EFM32 microcontroller. Reading from flash memory is limited to 14 MHz. Above, the processor must insert a wait state to give time to access the slow flash memory. A recommended solution to reduce this negative effect is activating the microcontroller's data and instruction cache. The EFM32 microcontroller has both and both were activated during the experiment. However, the cache can not replace fast memory access and thus, the energy consumption drops until 14 MHz. Starting from 21 MHz, two effects work against each other. The static consumption is applied shorter with faster processing speeds. On the contrary, the faster processing speed requires more wait states.

#### Crystal Oscillator

Nevertheless, the energy cost for a start-up using a crystal oscillator is much higher. The start-up oscillator time was set to 400 µs as recommended in the datasheet. According to figure 5.4 it might be possible to reduce the time further, but not less than 100 µs. However, having the 48 MHz clock as a temporary option for scalable computing increases the space of possible applications on this microcontroller.

## 5.5. Experiment: Sampling via DMA or Interrupt

The key to low-power systems is to keep the microcontroller as long as possible [112] in a low-power condition. Additionally, since multiple energy modes exist, a low-power design typically aims for the deepest low-energy mode possible. However, low-power modes are restricted when using DMA depending on the integrated circuit design. In general, DMA was made to transfer data between peripherals and memory directly without interaction of the CPU. The EFM32 microcontroller series additionally provides Peripheral Reflex Systems (PRS), which has the same intention. However, when using DMA or PRS the lowest energy modes are not usable. Therefore, an interrupt-driven technique can be used where the microcontroller needs to wake up to take each sample. DMA pays out at higher sampling rates. Therefore, DMA was recommended for sampling rates higher than 6 kSa/s in the application note [32]. Biomedical sensor systems have sampling rates up to 1 kSa/s [103]. Wearable sensors have low data rates, so low-power methods like DMA or the PRS require more power than the interrupt-driven sampling technique. Thus, for this thesis, an interrupt-driven method was identified as the best choice for biomedical signal processing. This decision allows using the deeper low-power mode of the microcontroller. Both were tested experimentally; the interrupt-driven sampling compared to DMA required 45 times less average power consumption.

## 5.6. Chapter Summary

In this chapter, microcontrollers were evaluated regarding their power consumption. The evaluation included diversity in (1) microcontroller architectures, (2) microcontroller compilers, (3) operating system settings, and (4) the clock unit settings. From this evaluation, the following best practices were derived.

A low power consumption per clock cycle is a remarkable feature of low-power microcontrollers. The ARM Cortex-M0+ has the best properties in terms of low power consumption per clock cycle. However, the ARM Cortex-M0+ is not power efficient for executing signal processing applications. For applications with multiple sensors attached to the microcontroller, with additional demand for signal processing, the ARM Cortex-M4F is the best choice. The ARM Cortex-M4F was the most powerful microcontroller, with the highest energy efficiency per clock cycle. The outstanding feature of the ARM Cortex-M4F is the fast computing of signal processing workload. It supports several features known from DSPs, including a co-processor for floating point operations. This benefit makes the microcontroller based on the ARM Cortex-M4F core faster in computing. Faster computing is equal to more time spent in low-power mode. More time spent in low-power is the key to low-power applications.

The toolchain, notably the compiler, impacts the machine code execution efficiency. The impact comes with fast code execution. The developer has to give the compiler instructions to optimize the code. Usually, it is common to optimize for code size, mainly when compiling for microcontrollers where code memory is limited. However, optimizing for code size is not recommended for low-power applications. Low-power applications have to be optimized for fast code execution. All brands of compilers provide compiler settings for fast code execution. The best-recommended settings were presented in this work above.

Using an RTOS is not necessary for simplistic applications. The most power-efficient option was the interrupt-based sampling without an RTOS. Larger sensor systems like ULPSEK require multiple threads for handling multiple sensor streams, signal processing tasks, power management, communication, and user interaction. These tasks require an RTOS. Unfortunately, an RTOS with default configuration will cause a relevant waste of energy. This waste of energy was verified experimentally by using two brands of popular RTOS. Therefore, the RTOS needs to be configured for low-power applications.

External high-frequency crystals, e.g., 48 MHz for core clock generation or using the integrated Phase Locked Loop (PLL), are not recommended for low-power applications. The energy for switching them on is too high. Instead, the clock unit achieved the best power efficiency by using a solution with two clock sources. One clock source is required for the core clock, and the second is for precision timing. The integrated RC-oscillator has to be set up at the highest possible rate and is used for the core clock. The highest rate was at 28 MHz for the EFM32. This clock was used to generate the core clock and was turned on and off frequently. The advantage of the RC-oscillator is the energy-efficient ability to switch the oscillator on and off. The disadvantage of the RC-oscillator is a lack of frequency precision. As compensation for this lack, a second clock source was an external crystal oscillator at 32 kHz. This clock was used for precise timing, such as sensor data sampling and system events, e.g., waking the microcontroller from a sleep state. This low-power strategy is not limited to microcontrollers of the EFM32 series. Thus it is a strategy that can be used for various microcontroller brands, such as ST microelectronics STM32 microcontroller series. However, the EFM32 microcontroller series provides outstanding clock rooting capabilities, giving remarkably more space for low-power optimization.

# Chapter 6

# Experiments with ULPSEK

The chapter describes experiments conducted with ULPSEK. The experiments aimed to prove the effectiveness of all low-power features added to hardware and software. Besides, the experiments were essential to estimate a theoretical model for the low-power system, including empirical values for the model parameters. Therefore, this chapter contains three experimental sections, each with an experimental topic. The first experiment was about the ability of ULPSEK to scale the power consumption depending on the configuration and number of features. In the second section, the energy of data reduction was compared with the energy for data transmission. The third experiment assessed ULPSEK for being powered by a body heat energy harvester. All experiments together rely on the configuration features of ULPSEK, which was the prerequisite for investigating these different system configurations.

## 6.1. Configurable Signal Processing

The ULPSEK was designed to configure analog and digital signal processing properties during runtime. Tobola et al. [75] was the related publication investigating this functionality experimentally. However, the target of the investigation was the signal processing of the ECG, as this signal chain had so far the most adjustable settings among the five sensors within ULPSEK.

### 6.1.1. Methods

The configurable properties along the signal processing path for this investigation were:

1. The ECG front-end had a configurable analog filter. The bandwidth of this filter was configurable to the narrowband (better artifact suppression) and wideband (more ECG details).
2. ADC sample rates were configurable to fixed sampling rates at 100, 200 or 1000 Sa/s.
3. ADC resolution was configurable to 8 or 12 bits.
4. Sample rate converters are implemented and activated automatically by the ECG module to meet the supported sample rates for algorithms.
5. The QRS complex detector algorithm was compiled for sample rates at 100 Sa/s and 200 Sa/s and could be chosen on demand by the system.

6. Vital parameters HR and two HRV can be activated on demand
7. Beat type detector algorithms can be activated on demand providing information about the abnormal QRS complex type
8. Instead of sampling the ECG and passing it through the beat detection algorithm, the ECG can be passed to a comparator where the threshold for R-peak detection is set dynamically.
9. The ECG sensor front-end can be powered down.

The above configuration space allows us 384 possible permutations. However, not all permutations make sense; they can even lead the system into an unacceptably invalid state. Therefore, the configuration space was reduced to 7 modes of operation for the ECG sensor:

- **ECG mode S1000V**: A sampling at 1000 Sa/s and 12 bit resolution for diagnosis. This mode requires ensuring a reduced activity of the monitored person because of moderate artifact suppression. For diagnosis purposes, the band-pass filter of the ECG fronted is switched to a wide band. The QRS detector is compiled for 200 Sa/s. Therefore a decimation Finite Impulse Response (FIR) filter reduces the sampling rate by a ratio of 5. All detected QRS complexes are checked for plausibility to calculate the HR and HRV.
- **ECG mode S200H**: Sampling mode at 200 Sa/s and 8 bit resolution. The band-pass filter is switched to a narrowband in this mode for better motion artifact suppression. The raw ECG signal is passed to the QRS detector compiled for the matched sampling rate of 200 Sa/s. HRV is not calculated.
- **ECG mode S200V**: Identical to S200H, but additionally with computation of HRV.
- **ECG mode S200B**: Same as S200V with additional beat classification software by Hamilton [171].
- **ECG mode S100U**: Sampling mode at 100 Sa/s using an interpolation FIR filter to convert the data rate from 100 Sa/s to 200 Sa/s for the QRS detector.
- **ECG mode S100C**: This version can run directly at 100 Sa/s without a sampling rate converter. The raw ECG signal is passed to the QRS detector compiled for the matched sampling rate of 100 Sa/s.
- **ECG mode COMP**: In this mode, the ECG signal is routed to an analog comparator which was inspired by Tompkins [212]. The actual comparator was a part of the microcontroller. In addition, an algorithm was added to adaptively adjust the threshold for R-peak detection.

All seven ECG modes were investigated for three monitoring modes

1. Continuous monitoring
2. Interval monitoring with short duty cycle: ECG over 2 minutes every 15 minutes

3. Interval monitoring with long duty cycle: Daily ECG (10-minute scan once a day)

The QRS detector algorithm was compiled for two sampling rates: 100 Sa/s and 200 Sa/s. Hamilton [171] verified the QRS detector with the MIT/BIH arrhythmia and American Heart Association (AHA) database at different sampling rates. Hamilton estimated a sensitivity above 0.9975 for both databases and a positive predictive value above 0.9979 at 200 Sa/s. For 100 Sa/s, Hamilton warns in his publication: “Performance differences only seem significant when the base sample rate is dropped as low as 100 or 125 Sa/s..”

A reproducible ECG signal must be established for the whole experiment. The ECG simulator MS410 [60] was used for generating a reproducible test signal. The simulator was set to Sinus Rhythm at 75 bpm for all measurements in this experiment. The correct heart rate calculation was controlled for every measurement allowing a precision of ± 1 bpm. Furthermore, the heart rate was also varied to investigate the dependency on power consumption.

### 6.1.2. Results

Table 6.1 gives the measured average power consumption for each mode, followed by the corresponding available information and the artifact suppression level. Figure 6.1 are representations of the data in Table 6.1 which was measured with continuous monitoring. Furthermore, it was determined that the beat detector algorithm requires 353 µJ per beat.

**Table 6.1.:** Power consumption in $\mu$W is depending on the mode of operation followed by the measuring uncertainty in $\mu$W for each mode. HR and NN are always calculated.

| ECG Mode | Power Consumption in $\mu$W | Sampling Rate | Bandwidth | Algorithm HRV | Algorithm Beat Type |
|---|---|---|---|---|---|
| S1000V | 1053 (± 2.1) | 1000 Sa/s | wide | on | off |
| S200H | 584 (± 0.7) | 200 Sa/s | narrow | off | off |
| S200V | 585 (± 0.7) | 200 Sa/s | narrow | on | off |
| S200B | 740 (± 10.9) | 200 Sa/s | narrow | on | on |
| S100U | 535 (± 0.6) | 100 Sa/s | narrow | off | off |
| S100C | 496 (± 0.6) | 100 Sa/s | narrow | off | off |
| COMP | 402 (± 0.5) | - | narrow | off | off |

### 6.1.3. Discussion

Seven modes of operation were introduced and investigated experimentally for a scalable ECG sensor system. Table 6.1 contains the results for power measurements in continuous monitoring. Figure 6.1 is an illustration of the power consumption from Table 6.1. Continuous sampling should be confused with the always-on operation. The system was

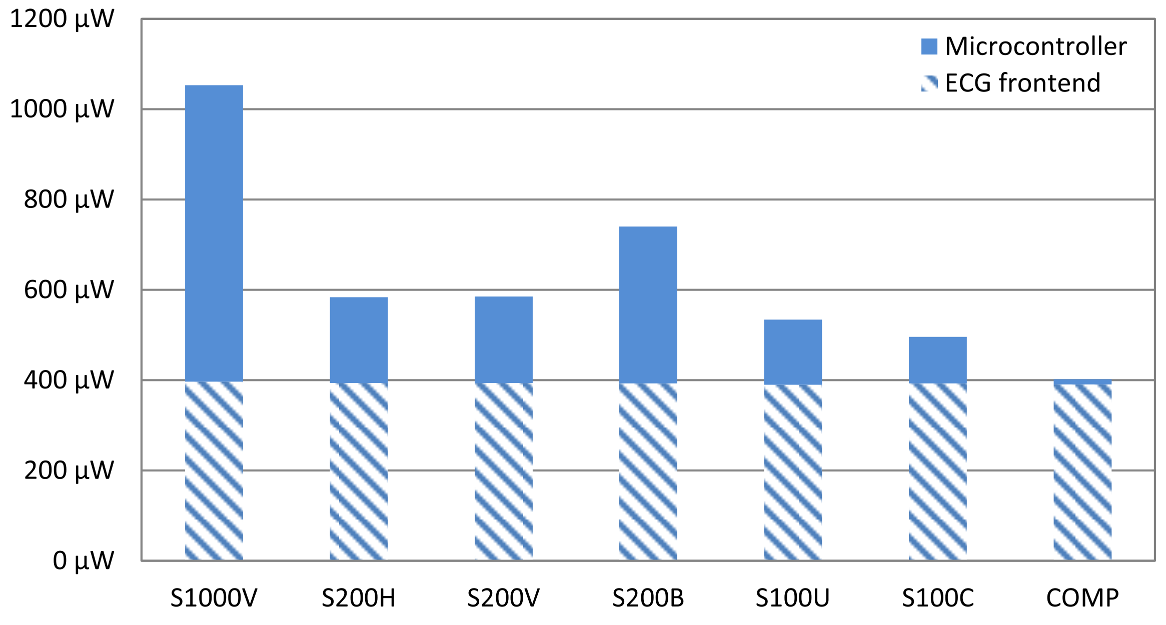


**Figure 6.1.:** This table shows the average power consumption in the continuous operation mode of the analog fronted and the microcontroller.

in a low-power mode after acquiring one sample, processing the sample, and storing the results. The sampling rate dominates the power consumption. With S1000V sampling was done at 1000 Sa/s. Therefore, the power consumption was the highest with 1053 µW of all measured values. S1000V is the only mode with the filter set to wide bandwidth. S200H and S200V were both around 584 µW. In S200V the HRV parameters were computed additionally. The algorithms for calculating the two HRV parameters are irrelevant compared to the overall power consumption. Unlike in S200B the computing-intensive algorithm for beat type classification required more energy. The algorithm classifies each QRS complex. This value is valid for the constant heart rate of 75 bpm. With S100U the power consumption for digital processing, including sampling, can be further reduced. It is optional to sample at 200 Sa/s. Therefore, in S100U, sampling was done at 100 Sa/s followed by digital upsampling using an interpolation filter to 200 Sa/s. The first stage in the QRS detector implementation by Hamilton was a band-pass filter with an upper cutoff frequency at 16 Sa/s. In S100C the algorithm was compiled for processing at 100 Sa/s directly. With this, the sampling rate converter is obsolete. Finally, COMP was a hardware implementation using a comparator with adaptive thresholds. COMP mode has a high sensitivity to artifacts. This mode works sufficiently for heart rate measurement of regular beats under rest conditions.

With this experiment, the system's scalability was demonstrated and quantified regarding energy. In Figure 6.1 a remarkable amount of power was needed for the analog front-end compared to the microcontroller. While the microcontroller spent most of the time in low-power mode, the analog front end was permanently running. The analog front end cannot be shut down when continuous sampling is required. This limitation exists because it takes the analog filters and the AD8232 up to 16 ms to get back into a steady state after shutdown. Filters generally need time after shutdown to reach a steady state, depending on the filter time constant. At that time, the AD8232 was available. The AD8232 required 390 µW including filters at 2.1 V power supply. Meanwhile, the successor AD8233 has been introduced promising 105 µW at 2.1 V. The AD8233 is a candidate for further improvements.

With the data from the experiments, the model in the next chapter was supplied with measured values for the model parameters. The beat detector algorithm works on every QRS complex. Therefore, the beat classification was determined to 353 µJ per beat.

## 6.2. Raw Data Transmission vs. Parameter Computing

The previous experiment focused on analog and digital signal processing in a real sensor system application. However, with this data, it is interesting to compare the energy cost for raw data transmission versus the energy cost for parameter calculation and parameter transmission. Kindt, Yunge, and Tobola et al. [62] was the related publication where this functionality was investigated in dynamic quality services switching context by combining the system model based on previous ECG measurements from this thesis with the system model for Bluetooth provided by P. Kindt from Technical University of Munich (TUM).

### 6.2.1. Methods

Two configurations of the ECG sensor system will be prepared for this experiment, similar to the previous experiments.
Individual configuration for high sampling rate and beat detection (S1000B):

1. The ECG front-end was set to wide-band.
2. ADC sample rates was set to 1000 Sa/s.
3. Sample rate converter from 1000 Sa/s to 200 Sa/s was activated.
4. Additionally the energy-hungry beat-type detector algorithms were activated.
5. The QRS complex detector algorithm was used at original sample rate 200 Sa/s.
6. Calculated parameters were HR and two HRV.
7. The ECG front-end was always on.

Individual configuration for lower sampling rate and no beat detection (S200V):

1. The ECG front-end was set to narrow-band.
2. ADC sample rates was set to 200 Sa/s.
3. ADC resolution was set 12 bits.
4. Sample rate converters were switched off.
5. The QRS complex detector algorithm was used at original sample rate 200 Sa/s.
6. Calculated parameters were HR and two HRV.

7. The ECG front-end was always on.

Each case mode was compared against raw data transmission. In the case of raw data transmission, the computing of parameters was deactivated.

### 6.2.2. Results

Power consumption is given in Table 6.1 for each mode, followed by the corresponding available information and the artifact suppression level. The artifact suppression is best in sampling mode using the filtering functionality of the QRS detector, combined with the narrow band-pass filter setting (NB).

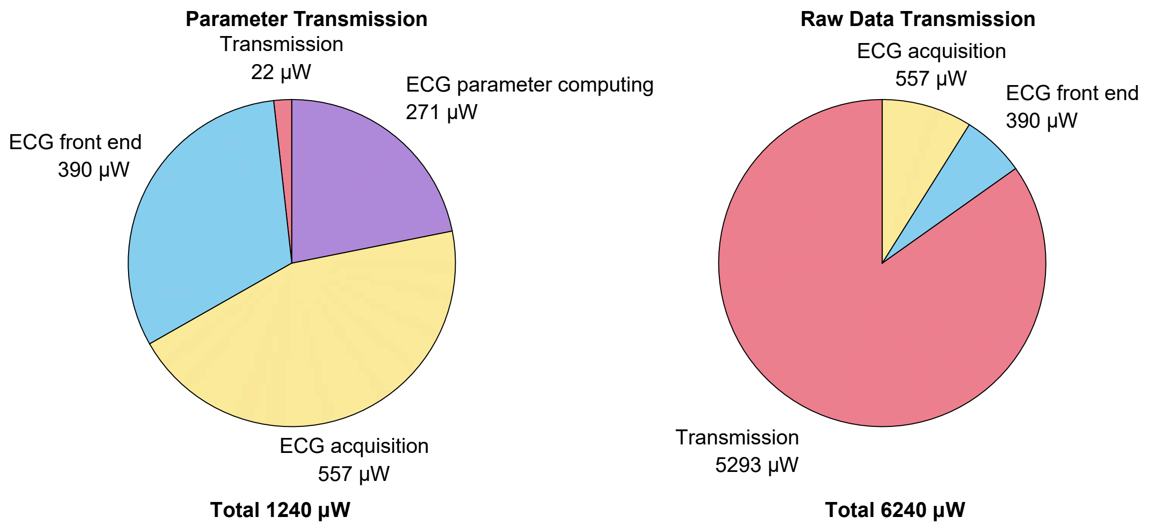


**Figure 6.2.:** Two pie charts comparing ECG sampling at 1000 Sa/s with beat classification algorithm: the left chart shows the average power for parameter computing and transmitting parameters; the chart to the right shows the average power required for transmission of raw ECG data.

### 6.2.3. Discussion

In figure 6.2 on the left, four parameters were computed: HR parameter, two HRV parameters, and the computing-intensive beat classifier. In this configuration, the computed parameters were transmitted via Bluetooth after computing. In figure 6.2 on the right, instead of computing the parameters, the whole continuous ECG was sent at the highest configurable rate of 1000 Sa/s. The raw ECG contains information about the parameters, such as the information that can be extracted elsewhere. Comparing 6240 µW for raw data transmission with 557 µW for parameter computing, the power can be reduced down to 9 % in this specific example.

For the following case, in figure 6.3 on the left, three parameters were computed: HR parameter, two HRV parameters. The computed parameters were also transmitted via

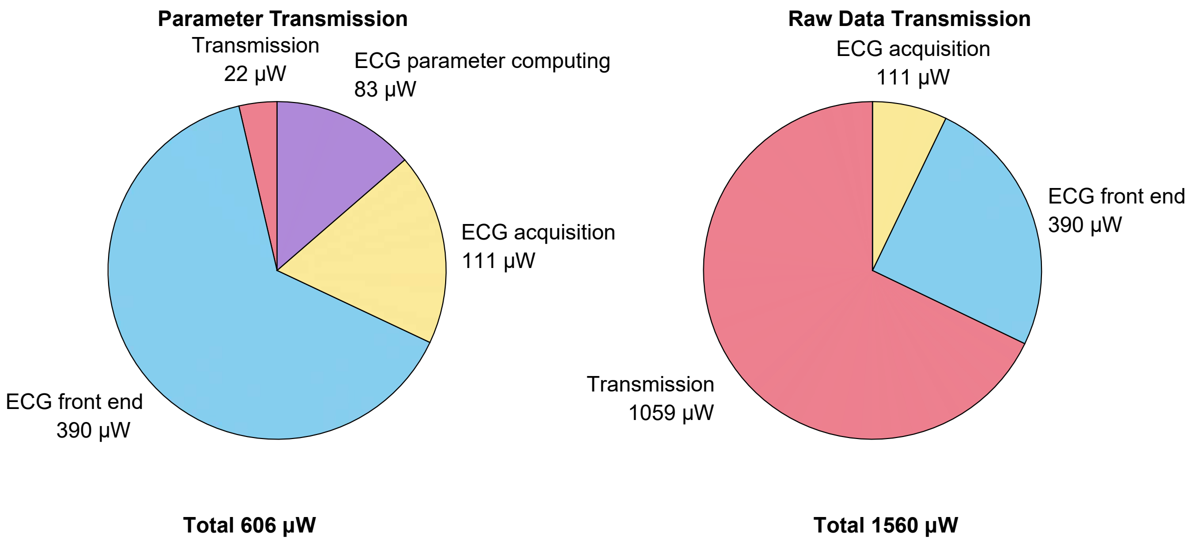


**Figure 6.3.:** Two pie charts comparing ECG sampling at 200 Sa/s: the left chart shows the average power for parameter computing and transmission of parameters; the chart to the right shows the average power required for transmitting raw ECG data. In both cases, the sampling rate was 200 Sa/s.

Bluetooth after computing in this configuration. In figure 6.3 on the right, instead of computing the parameters, the whole continuous ECG was sent at a lower rate of 200 Sa/s. Comparing 1560 µW for raw data transmission with 606 µW for parameter computing, the power can be reduced down to 39 % in this specific example.

The power savings in the second experiment are less because the data rate linearly dominates the radio's power consumption. As expected, reducing the data rate by calculating parameters was very effective.

## 6.3. Self-Powered Multiparameter Monitoring

By reviewing achievements in power harvesting technology, Mateu, Pollak, and Spies et al. [95, 152, 156, 157] attracted attention with an efficient harvester circuit. The team around Spies provided an energy harvester for experiments with ULPSEK. This harvester should theoretically provide enough energy for powering ULPSEK from the heat of the human body. ULPSEK was designed with several configurable properties which allow adapting the system to various low-power applications. The aim was to find a configuration of ULPSEK which allows powering the wearable sensor by the wearable energy harvester. The idea behind this experiment was to test the capabilities of ULPSEK. The exciting question behind this experiment was: assuming ULPSEK can be run by an energy harvester, how to configure the system?

In this experiment ULPSEK was powered by an energy harvester as proof of concept of the sensor system's low-power capabilities. This section relies on the experiments from the publications *Self-Powered Multiparameter Health Sensor*, Tobola et al. [40].

### 6.3.1. Methods

#### Experimental Setup

This experiment aimed to power the sensor system with an energy harvester. Therefore, one person was equipped with ULPSEK and the energy harvester while following daily activities. An independently powered data acquisition unit recorded voltages, currents, and four temperatures during the experiment.

The daily activities performed were sitting, talking, eating, or walking. The person was in different rooms at various temperatures: home office at 20.5 °C for 7 minutes, bedroom at 15.0 °C for 9 minutes, living room at 22.5 °C for 23 minutes, outside at −2.7 °C for 3 minutes, and back to the living room for 6 minutes.

#### Energy Converter

In this experiment, the body's thermal energy was converted into electrical energy. The critical component, therefore, was a heat sink attached to a Peltier element combined with an efficient converter circuit (Figure 6.4). The thermal harvester was worn at the forearm or the chest, as shown in Figure 6.5.

It includes a heat sink, a thermoelectric module, and a DC-DC converter circuit introduced in Pollak et al. [152], which can handle the typically small output voltages of thermoelectric generators around 50 mV/K. This power converter increases the voltage level to the system voltage level of 3 V with a high efficiency of up to 70% at 200 mV input.

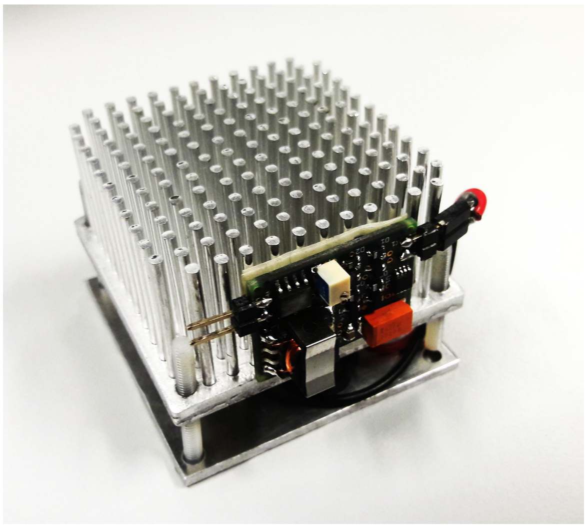


**Figure 6.4.:** The thermal energy harvester includes a heat sink, a Peltier element, and a converter circuit.

#### Monitoring Program

The ULPSEK implements a Health Monitoring Manager which implements a monitoring program according to subsection 2.1.5. For this experiment, the configuration was implementing a dedicated use case.

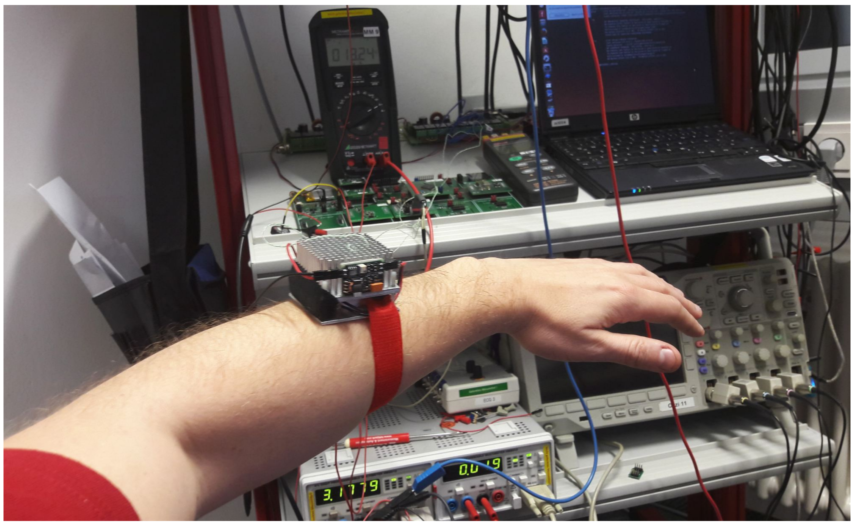


**Figure 6.5.:** Thermal energy harvester placement at the forearm fixed by a strip.

This configuration consisted of two phases that were repeated continuously: A data acquisition phase and a hibernation phase. The data acquisition phase lasted for 31 seconds. The pause lasted for 12 minutes. The acquisition phase was split into a 21 seconds period without parameter transmission. After 21 seconds of operation, the Bluetooth 4.0 module was activated, and data transmission to a tablet PC was initiated while monitoring and evaluation continued. The transmission period lasted 10 seconds. After 10 seconds, the sensor switched back to hibernate mode for 12 minutes. The procedure was repeated periodically.
For this experiment, the ULPSEK was configured to acquire data in a fixed configuration as the following:

1. The ECG sampling rate was at 200 Sa/s.
2. The QRS detection was using Pan-Tompkins algorithm.
3. Heart rate calculation was activated.
4. Calculation of two heart rate variability parameters was activated.
5. Cardiac beat classification was set up using Hamilton implementation according to [171].

The second individual configuration was at a lower sampling rate and no beat detection (S200V):

1. The ECG front-end was set to narrow-band.
2. ADC sample rate was set to 200 Sa/s.
3. ADC resolution was set 12 bits.
4. Sample rate converters were off.

5. The QRS complex detector algorithm was used at original sample rate 200 Sa/s.
6. Parameters were calculated: HR and two HRV.
7. Cardiac beat classification using Hamilton implementation was set up according to [171].
8. ECG parameters were transmitted.
9. Accelerometer data was transmitted.
10. Body surface temperature was transmitted.

#### Setup of the Power Distribution Network

The power distribution network of the ULPSEK was modified for this setup. The power distribution network was introduced with Figure 3.12 in section 3.7. External power sources and wireless charging were removed for this experiment. Furthermore, the battery was removed. Instead, a body heat harvester module followed by a capacitor was connected to the node M_BAT in Figure 3.12.

#### Capacitors for Energy Storage

An energy buffering capacitor is an essential part of the energy harvesting system. The stored energy was provided, while the sensor required energy for measurement, processing, and data transmission. Table 6.2 is a list of capacitors that were evaluated during this experiment. The most relevant parameters were the Equivalent Series Resistance (ESR) and the current leakage. Both parameters influence the power losses. A high ESR causes a dynamic energy leak each time energy is put into or out of the capacitor. Current leakage is defined as a static current. The parameters were taken from the datasheets. In theory, ESR and leakage should be taken as low as possible. In the experiments, all capacitors were investigated for usage as energy storage.

**Table 6.2.:** Evaluated capacitors for energy storage.

| **Type** | **Capacity** | **ESR** | **Leakage** |
|---|---:|---|---:|
| EECF5R5U105 | 1 F | <30 Ω | 5 µA |
| SCMR18C105MRBA0 | 1 F | 150 mΩ | 6 µA |
| SCMR14C474MRBA0 | 470 mF | 300 mΩ | 2 µA |
| PM-5R0V474-R | 470 mF | 420 mΩ | 8 µA |
| PB-5R0V104-R | 100 mF | 4 Ω | 3 µA |

### 6.3.2. Results

#### Sensor System Performance

The power consumption of the sensor during hibernation was measured at 40.9 µW. The sensor could wake up for the next sampling period or accelerometer events in this mode. The power consumption of the sensor during the activity of 31 seconds, including transmission, was on average 2.52 mW including 9.0 % power loss on voltage regulators. Over a complete cycle of 12.5 minutes, the power consumption was in average 137 µW including 9.2 % power loss on voltage regulators. Figure 6.6 shows the power consumption of the sensor during the 31 seconds of activity: waking up from hibernation at the beginning and returning to hibernation at the end. The data was received on a Nexus 7 Android tablet (Figure 6.7).

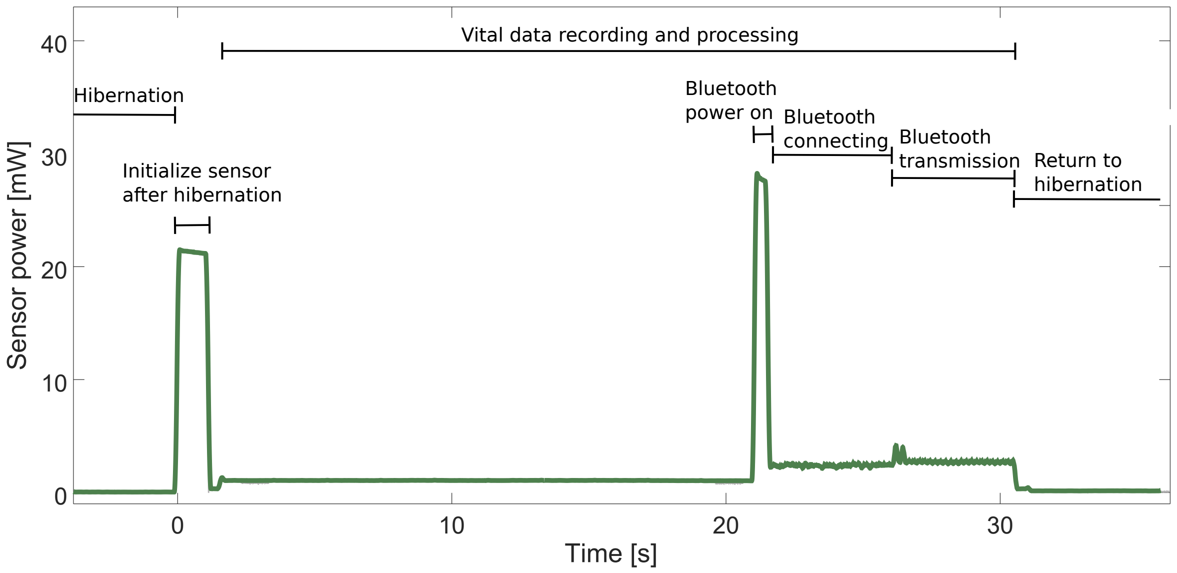


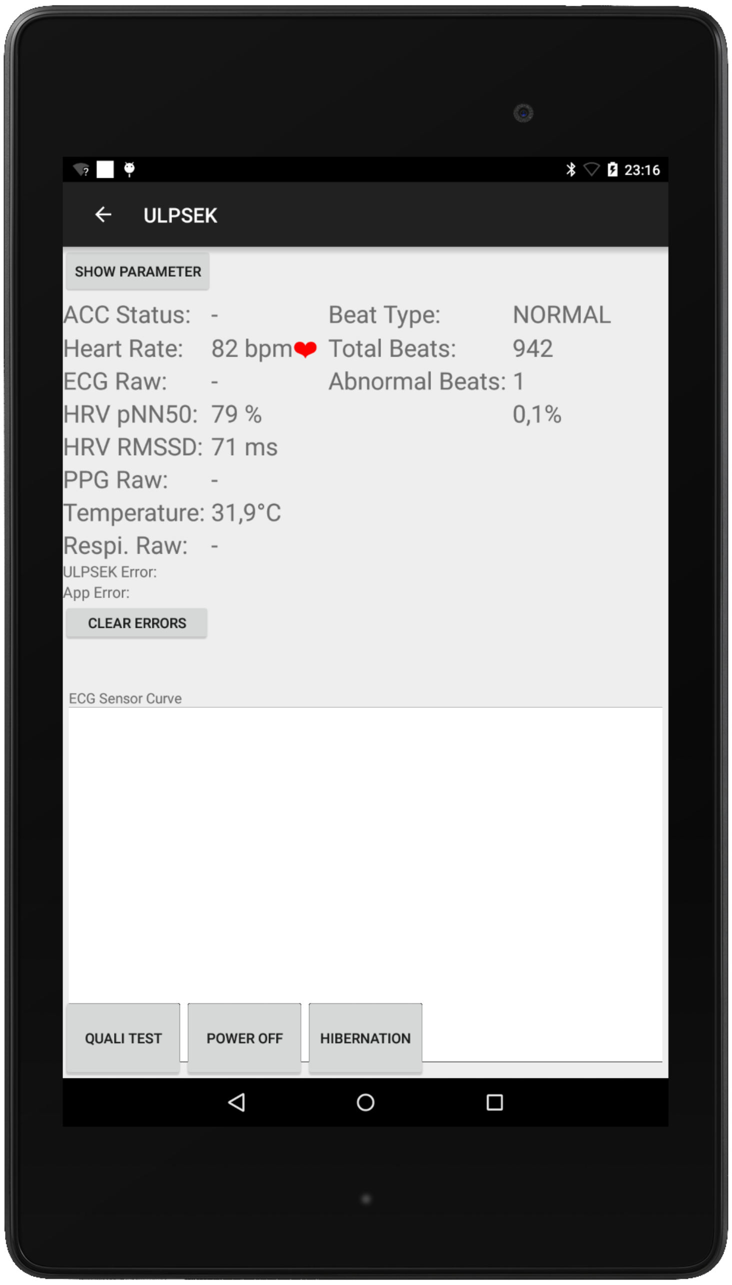
**Figure 6.6.:** The graph shows the power consumption while the sensor wakes up from hibernation for health monitoring and returns to hibernation. The sampled signal at 1000 Sa/s and for better interpretation filtered back and forward by moving average filer at a width of 100 ms. Starting at t=0 s the sensor wakes up from hibernation taking energy for wake up, followed by the start of vital data acquisition at t=1.5 s. Bluetooth starts and begins advertising at t=21 s followed by connection and beginning of transmission at t=26 s. The sensor changes into hibernate mode at t=31 s. This sequence was repeated every 12.5 minutes.

## Energy Harvester Performance

The energy harvester started generating power on a minimum temperature difference $\Delta T$ starting at 2.7 K. The average power generated by the harvester during the experiment was 171 µW (Figure 6.8). The maximum power of 600 µW was generated outdoors at an ambient temperature of −2.7 °C. The temperature difference $\Delta T$ and the generated power $P_h$ measured at the output of the converter circuit showed a linear relationship. The equation $P_h(\Delta T) = 114\,\mu\text{W/K} \cdot \Delta T - 306\,\mu\text{W}$ fits the data by a correlation coefficient 0.991 for input values $\Delta T \in [2.7\,\text{K}, 8.5\,\text{K}]$.

## Best Capacitor for Energy Buffering

Only the capacitor PB-5R0V104-R with 100 mF enabled operation without battery. The other capacitors listed in Table 6.2 discharged at a higher rate than the harvester could provide.



|  |  |  |  |
|---|---|---|---|
| ACC Status: | - | Beat Type: | NORMAL |
| Heart Rate: | 82 bpm | Total Beats: | 942 |
| ECG Raw: | - | Abnormal Beats: | 1 |
| HRV pNN50: | 79 % |  | 0,1% |
| HRV RMSSD: | 71 ms |  |  |
| PPG Raw: | - |  |  |
| Temperature: | 31,9°C |  |  |
| Respi. Raw: | - |  |  |

ULPSEK Error:
App Error:

**Figure 6.7.:** Real time data received vie Bluetooth at the Nexus 7 tablet.

### Energy Balance

The balance summarizes average power levels in Table 6.3. In this table, the power required for the sensor in the "on" state is composed of the power for data acquisition, processing, transmission, and restoring the system from hibernation. The data acquisition and processing power were in total 1.0 mW. The power for transmission and restoration was 1.5 mW. Furthermore, all measured values are listed in detail using four graphs in Figure 6.8, including (1) the total power taken by the sensor, (2) the power generated by the thermal harvester, (3) the voltage at the capacitor; and (4) the temperatures around the harvester.

**Table 6.3.:** Energy balance for one period of 12.5 minutes.

| Energy source/sink | Power | Duration | Energy per cycle |
|---|---:|---:|---:|
| Harvester | 171 µW | 761 s | 130 mJ |
| Sensor on | -2517 µW | 31 s | -79 mJ |
| Sensor hibernating | -41 µW | 730 s | -30 mJ |
| Capacitor losses | -16 µW | 761 s | -12 mJ |
| Probe effect | -4.3 µW | 761 s | -3 mJ |
| **Total** | | | **6 mJ** |

## 6.3.3. Discussion

The harvester powered the sensor system. Figure 6.8 shows the recording of the power, energy, and temperature levels while wearing the sensor daily and staying and walking indoors in different rooms at different temperatures. At t=40 s, 3 minutes were spent outside on a cold winter evening, giving the sensor temporarily a boost up to $P_h = 600\,\mu\text{W}$.

The power output $P_h$ of the harvester depends on the temperature difference $\Delta T$ linearly. This observation differs from the generally cubic relationship [95]. A cubic relationship is an advantage for higher temperature differences. However, we must deal with lower temperature differences between skin and air in body energy harvesting. Thus, the circuit introduced by Pollak et al. [152] was optimized for best performance at low input voltages using an efficiency factor depending on the input voltage.

At the end of the recording, the amount of energy stored in the capacitor was 24 mJ. Thus, the level at the end was higher than at the beginning of the experiment. The total balance is presented in Table 6.3, where energy is listed per cycle, giving an average increase of 6 mJ per cycle. Of course, the amount of power left in the capacitor at the end of every experiment depends mainly on two circumstances: First, it depends on the power consumed by the sensor, which was fixed in the setup of this experiment. Second, the power provided by the harvester depends on room temperature or weather conditions when being outside.

Additionally, the choice of the capacitor was a key factor for the experiment's success. In this thesis, the lowest losses were estimated at 16 µW for the capacitor PB-5R0V104-R in total. These 16 µW can be split into 15 µW for the leakage current of the capacitor and

1 µW for the losses caused by the internal series resistance of 4 Ω. The experiment failed for all other capacitors in Table 6.2 except PB-5R0V104-R.

Unlike the harvester, the average power consumption was predictable for the sensor. The ULPSEK was configured using its configuration abilities. For this experiment, several algorithms were used. The most power-intensive algorithm was the QRS beat classifier by Hamilton [171]. This algorithm takes 353 µJ for every QRS-complex classification. The beat classification requires 69 times more energy than the HR and HRV calculation. On the contrary, to lower the power consumption, a health check was performed every 12.5 minutes for a duration of 31 s and therefore resting most of the time in hibernation.

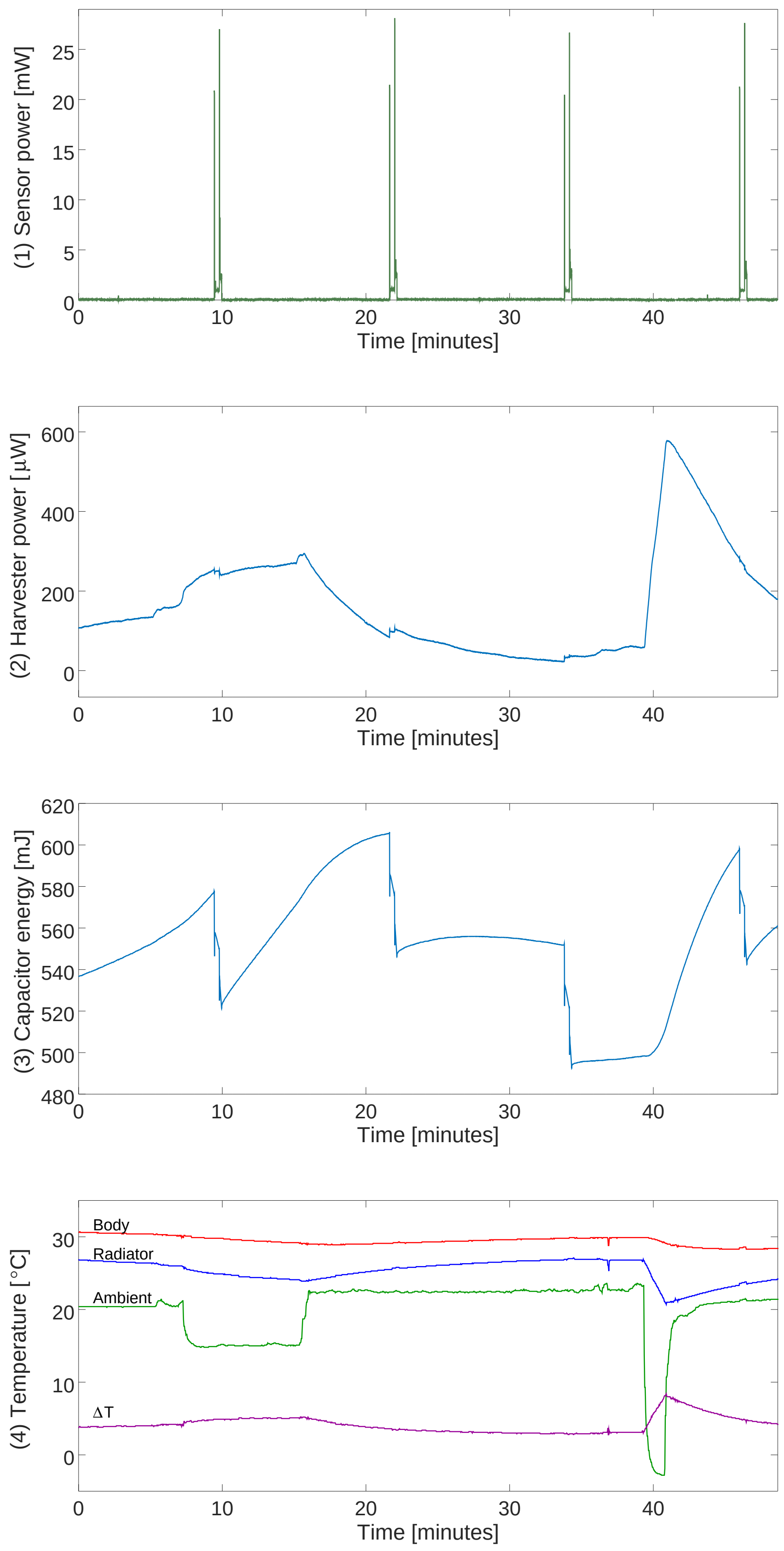


**Figure 6.8.:** Recorded data while powering a sensor with a body heat harvester showing (1) the total power taken by the sensor, (2) the power generated by the thermal harvester, (3) the voltage at the capacitor, and (4) the three temperatures around the harvester including the difference $\Delta T$ between body and radiator temperature at the harvester.

## 6.4. Chapter Summary

The experiments aimed to gain a proof of concept for the methods applied to the hardware and software of the ULPSEK. The first experiment confirmed that the power consumption scales with the configuration and number of features activated. The ULPSEK can vary the power consumption depending on the configuration, enabled by multiple low-power hardware and software features. The ability to scale the power consumption with the configuration is a precondition to achieve the lowest energy level for a specific application. The second experiment was the comparison of raw data transmission against the data rate reduction by computation of parameters. It was empirically confirmed in a practical application by configuring the ULPSEK for both cases: raw data and parameter calculation. The last experiment has shown how to configure the ULPSEK for energy-autonomous operation by utilizing energy harvesting. The results proved the impact of the low-power features of the hardware and software. During the experiments, some suboptimal design properties were detected. Thus, the system was optimized iteratively by executing the experiments multiple times until a satisfactory result was achieved. However, this optimization process uncovered some surprising facts. An interesting effect was the software's sensitivity to inappropriate design changes, which accidentally caused enormous changes in power consumption. The gained data and experience within this chapter were used for developing the model presented in the next chapter.

# Chapter 7

# Generalized Model for Power Consumption

The content of this chapter is the formal description of an energy model for the biomedical signal processing of the wearable sensor system regarding the used hardware and software. The purpose of this model is the ability to make predictions about energy consumption. Using such a model for predictions about energy consumption allows faster fitting of the system parameters to find the minimum power consumption possible with a specific system.

The energy model was derived based on the theoretical and empirical work in the previous chapters. Tobola et al. [69] was the related publication where the first version of the model was introduced. However, the model presented in this chapter was extended, modularized, and generalized to cover a broader range of applications.

## 7.1. Model Construction

In this section, a theoretical model for the power consumption of a system will be described that takes full advantage of all low-power features implemented with ULPSEK. The purpose of the formal model is to identify the critical parameters behind the processes of a microcontroller-based sensor system. The focus of this thesis was on the hardware and software components that are involved in processing signals. In contrast, parts that do not participate in processing signals, such as wireless transmission and communication, will be handled at a high abstraction level. This chapter will introduce the model to three expansion stages: M1, M2, and M3. The foundation of model M1 describes an always-on sensor system. On top of model M1, the second stage (M2) adds a low-power feature for short idle intervals controlled automatically by the sensor system. With the third stage (M3), the model was extended by another low-power feature for relatively long idle intervals. Long idle intervals were referred to as Hibernation in this thesis. A differentiating description of the low-power modes was given in section 3.9, in particular pointing out the differences between low-power in idle (M2) and Hibernation (M3) in table 3.1. Each of the three model expansion stages, M1, M2, and M3, introduces specific system properties stepwise. Finally, with the fully populated theoretical model M3 a system will be described that takes full advantage of all low-power features which were empirically investigated with ULPSEK.

### 7.1.1. Model M1: Always-On Sensor System

Model M1 describes an always-on sensor system as a foundation. Such a system does not use any sleep modes of the microcontroller and is not switching off unused sensors. Therefore, in such a system, the clock is always present and the microcontroller core is never powered down. The energy consumption for the sensor system was expressed as a set of equations containing timing and power parameters. As a starting point, the fundamental limits of the computing hardware were taken from the fundamentals in section 2.3. There, dynamic $P_{core,dyn}$ and static power $P_{stat}$ consumption was expressed for digital systems such as the microcontroller. $P_{core,dyn}$ was expressed as a function of the microcontroller's core clock frequency $f_{core,M1}$. Besides this, the microcontroller contains several peripheral subsystems, such as communication interfaces and timers. The power consumption for the peripherals was expressed by $P_{peri,dyn}$. $P_{peri,dyn}$ is a dynamic power consumption that depends on a peripheral clock $f_{peri}$. The clock $f_{peri}$ can be selected independently below or equal $f_{core,M1}$. As introduced in the fundamentals chapter, a sensor system consists further of a power distribution network $P_{PDN}$, a set of sensors $P_{sensors}$, their sensor front-ends $P_{frontends}$, the wireless communication module (radio) $P_{radio}$, and User Interface (UI) components $P_{UI}$, such as a display, a speaker and LEDs. $P_{PDN}$ is primarily caused by the voltage regulators. For wireless communication, several models exist depending on the communication technology, configuration parameters, and application. Therefore, an appropriate model needs to be utilized for $P_{radio}$. A model for $P_{radio}$ is not part of this thesis. For further details, a detailed model for $P_{radio}$ was proposed by Kindt et al. [92]. However, in the experiment section 6.2, this radio model was utilized as part of a collaboration with a research project. With this, the average sensor system power consumption $P_{sys,M1}$ for an always-on sensor system can be declared as the sum of all parts.

$$\begin{aligned} P_{sys,M1} = {} & P_{core,dyn}(f_{core,M1}) + P_{peri,dyn}(f_{peri}) + P_{stat} \\ & + P_{PDN} + P_{sensors} + P_{frontends} \\ & + P_{radio} + P_{UI} \end{aligned} \tag{7.1}$$

### 7.1.2. Model M2: Automatic Low-Power in Idle State

ULPSEK took advantage of the low-power microcontroller mode, which was utilized whenever the microcontroller was idle. Low-Power in Idle State is one of two sleep modes introduced in section 3.9 at p. 70. Entering the low-power mode was controlled by the operating system. In this low-power mode, the peripherals and sensors were kept active while the computing was shut down by the clock and power gating. Additionally, ULPSEK was set up to shut down the oscillator dynamically. For this model, an independent clock setting will be used by introducing $f_{core,M2}$. Further, in low-power mode, a subset of digital circuits was still powered and clocked at a lower clock frequency $f_{lp}$, expressed as a dynamic $P_{lp,dyn}$ and a static $P_{lp,stat}$ part. Therefore, a factor is required $p_{act} < 1$, which declares the probability of a system being active. On the contrary, with $1 - p_{act}$, the probability of the system taking advantage of the low-power mode can be expressed. With this, $P_{sys,M2}$ declares the power consumption for a sensor system that utilizes the low-power mode during idle.

$$\begin{aligned}\tilde{P}_{sys,M2} = &\, p_{act}\left(P_{core,dyn}(f_{core,M2}) + P_{peri,dyn}(f_{peri}) + P_{stat}\right)\\ &+ (1 - p_{act}) \cdot \left(P_{lp,dyn}(f_{lp}) + P_{lp,stat}\right)\\ &+ P_{PDN} + P_{sensors} + P_{frontends}\\ &+ P_{radio} + P_{UI}\end{aligned} \tag{7.2}$$

Equation 7.2 was further extended by the energy cost for starting-up, acquiring, transition, and release according to the definition in section 4.7. For this extension, $r_c$ was introduced, which is the rate in cycles per second. The system is sent to low-power mode and woken up at this average rate. Further, $E_{STAR}$ is the energy required in each cycle for Start-up, Acquisition, Transition, and Release, according to the definition section 4.7. Finally, $t_{STAR}$ is the time that passes during each cycle for Start-up, Acquisition, Transition, and Release.

$$\begin{aligned}P_{sys,M2} = &\, p_{act}\left(P_{core,dyn}(f_{core,M2}) + P_{peri,dyn}(f_{peri}) + P_{stat}\right)\\ &+ (1 - p_{act} - r_c t_{STAR}) \cdot \left(P_{lp,dyn}(f_{lp}) + P_{lp,stat}\right)\\ &+ r_c E_{STAR}\\ &+ P_{PDN} + P_{sensors} + P_{frontends}\\ &+ P_{radio} + P_{UI}\end{aligned} \tag{7.3}$$

Equation 7.3 implements the automatic sleep state, which adapts the power consumption to a minimum depending on the workload while sampling and processing sensor data. Optionally, the probability $p_{act}$ can be approximated with the $r_c$ and the number of clock cycles $N_{cy}$ for processing, with relation to the core clock frequency $f_{core,M2}$.

$$p_{act} = \frac{N_{cy} r_c}{f_{core,M2}} \tag{7.4}$$

### 7.1.3. Model M3: Hibernation

ULPSEK also took advantage of hibernation to conserve energy. Hibernation is one of two sleep modes introduced in section 3.9 at p. 70. Hibernation was declared as the ability of a system to go down to a deeper low-power mode by switching off peripherals and pausing the operation of the sensor system. If system parts are in hibernation, there is still power consumption remaining. This remaining power consumption can be measured for each part separately $P_{pdn,hib}, P_{sensors,hib}, P_{frontends,hib}, P_{radio,hib}, andP_{ui,hib}$. Note that the average hibernation power for each submodule should include an initiation after hibernation. In particular, for $P_{radio,hib}$, this may significantly increase depending on the used communication technology, wireless radio parameters, and usage frequency of the hibernation. Additionally, the microcontroller needs a minimum of energy to keep the low-power clock running to initiate the wake-up from hibernation. In equation 7.5, the power required for the low-power subsystem at low clock $P_{lp,dyn}(f_{lp}) + P_{lp,stat}$ remains in hibernation. Thus the power consumption of the system during hibernation is $P_{hib}$.

$$\begin{aligned} P_{hib} &= P_{lp,dyn}(f_{lp}) + P_{lp,stat} \\ &+ P_{pdn,hib} + P_{sensors,hib} + P_{frontends,hib} \\ &+ P_{radio,hib} + P_{ui,hib} \end{aligned} \tag{7.5}$$

Equation 7.5 describes a system in permanent hibernation. In hibernation, all sensors are deactivated and processing is stopped. The next equation adds a probability factor that declares the portion spent in hibernating $p_{hib} < 1$, and the remaining portion spent in mode M2. The power consumption in model M2 ($P_{sys,M2}$) was described with equation 7.3 previously.

$$\tilde{P}_{sys,M3} = p_{hib}P_{hib} + (1 - p_{hib})P_{sys,M2} \tag{7.6}$$

The always-on model (model M1) may be used together with (model M3) without utilizing (model M2). However, this model combines M1, M2, and M3, taking advantage of both: the operation in a low-power mode during sampling (model M2) and the hibernation feature (model M3). The preliminary equation 7.6 assumes that transitions between sensor activity and hibernation occur instantly. Thus, the next step is to add the cost for transitions with the time spent in hibernation $t_{hib}$, the total time required for entering and returning from hibernation $t_{hib,er}$, the energy required for entering and returning from hibernation $E_{hib,er}$, and the hibernation cycle rate $r_h$.

$$P_{sys,M3} = p_{hib}P_{hib} + (1 - p_{hib} - r_h t_{hib,er})P_{sys,M2} + r_h E_{hib,er} \tag{7.7}$$

The probability of spending time in hibernation $p_{hib}$ can be expressed with the average hibernation cycle rate $r_h$ .

$$p_{hib} = r_h t_{hib} \tag{7.8}$$

Equation 7.8 in equation 7.7 results in a model variant where an approximated average rate replaced the relative probability.

$$P_{sys,M3} = r_h(t_{hib}P_{hib} + E_{hib,er}) + (1 - r_h(t_{hib} + t_{hib,er}))P_{sys,M2} \tag{7.9}$$

The next step requires solving the dependency between $r_h$ and $t_{hib}$. A solution, therefore, is to introduce the hibernation interval $t_h$.

$$r_h = \frac{1}{t_h} \tag{7.10}$$

The hibernation interval $t_h$ can be seen as a sum of three-time intervals, where $t_{hib}$ is the already introduced time spent in hibernation, $t_{hib,er}$ is the time required to enter and return from hibernation, and $t_{sens}$ is the time while sensing and processing signals.

$$t_h = t_{hib} + t_{hib,er} + t_{sens} \tag{7.11}$$

In an application, $t_{sens}$ is often split into two parts, pre-sensing $t_{sens,pre}$ and $t_{sens,eff}$. Pre-sensing is required to fill buffers, initialize filters, and wait for valid outputs. Once pre-sensing is completed, a system can deliver output. Thus, after pre-sensing, effective sensing follows. Thus, the effective time to evaluate a signal after hibernation can be shorter if the value of $t_{sens,pre}$ is close to the value of $t_{sens,eff}$. This detail will not be used in the further model equation. However, it will be necessary for practical system estimations.

$$t_{sens} = t_{sens,pre} + t_{sens,eff} \tag{7.12}$$

Equation 7.10 and equation 7.11 combined in equation 7.9 result in an alternative model M3 definition.

$$P_{sys,M3} = \frac{E_{hib,er} + P_{hib}t_{hib} + P_{sys,M2}t_{sens}}{t_{hib,er} + t_{hib} + t_{sens}} \tag{7.13}$$

Equation 7.13 is the final model for the system implementing hibernation.

## 7.1.4. Precondition for reliable Hibernation

The model M3 was declared with equation 7.13. However, hibernation is not always beneficial. Hibernation can cause a higher average power consumption. Therefore, the preconditions for hibernation need to be checked before entering hibernation. Equation 7.14 is a result of the ratio $P_{sys,M3}$ divided by $P_{sys,M2}$.

$$\gamma_{hib} = \frac{P_{sys,M3}}{P_{sys,M2}} = \frac{E_{hib,er} + P_{hib}t_{hib} + P_{sys,M2}t_{sens}}{P_{sys,M2}(t_{hib,er} + t_{hib} + t_{sens})} \tag{7.14}$$

This ratio is a predictive metric of power savings by using hibernation. For power-efficient operation, $\gamma_{hib}$ should be below 1. Hibernation must not be initiated if $\gamma_{hib}$ is larger or equal to 1.

$$\gamma_{hib} < 1 \tag{7.15}$$

This power efficiency rule will be explained later in the discussion.

## 7.1.5. Core Clock Independence

A system that behaves according to model M2 has the advantage of automatically adapting the energy consumption according to the workload. This ability makes the model independent from the core clock frequency. In contrast, according to model M1, the system requires the scaling of core clock frequency. This effect can be seen in Figure 7.1 where an always-on system (according to model M1) was compared to automatic sleep in idle (model M2). The measurements were done at six discrete frequencies which were technically selectable at the EFM32 microcontroller. For all cases, the current of the microcontroller was measured. According to the theory in section 2.3, the microcontroller current increases with the core frequency for the always-on measurement. This behavior corresponds to the energy mode EM0 in the terminology of the EFM32 microcontroller. Additionally, the figure shows a microcontroller current measured in sleep mode, which is the energy mode EM2 in the terminology of the EFM32 microcontroller. Thus, the solid upper line (EM0) and the solid bottom line (EM2) are the boundaries in which the average power consumption of any application running automatic sleep in idle mode must take place. As an example in Figure 7.1, the dashed line shows ULPSEK operating in automatic sleep in idle according to model M2 while operating on an ECG processing using the previously introduced configuration S200H. In this example, the workload was 174000 computing cycles, which results in a relative workload of 14.5 % for 1.2 MHz or 0.6 % in case of 28 MHz. The microcontroller performs clock gating on demand, switching the core clock on (top line) and off (bottom line). It can be seen that the current consumption is independent of the core clock frequency. This independency is theoretically plausible because doubling the core clock frequency halves the processing time. As a result, the average power consumption remains almost constant over the core clock frequency.

The advantage of automatic sleep in idle allows for adapting to a necessary minimum while having the headroom for high processing loads. Thus, according to model M2, the highest core clock should be used for automatic sleep in idle. In contrast, according to model M1, the lowest clock is required for always-on systems, including headroom for transient higher loads, which is also required with automatic core clock regulation. Even so, M2 outperforms M1.

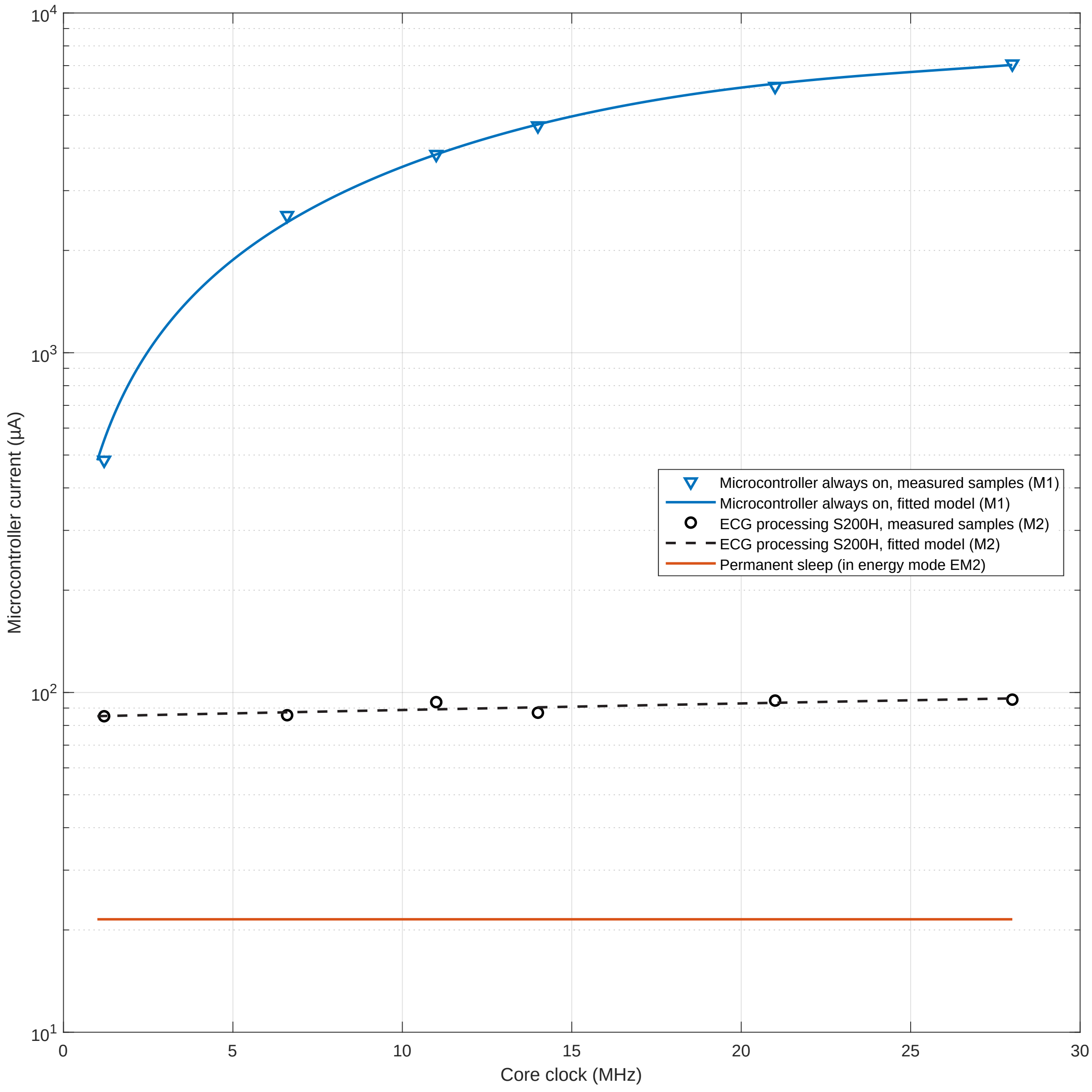


**Figure 7.1.:** Microcontroller current over the core clock in three cases: always-on operation (M1), microcontroller processing ECG with automatic sleep in idle state (M2), and permanent sleep (bottom line).

## 7.2. Model Analysis and Discussion

In the previous section, the model was declared in three expansion stages M1, M2, and M3. This section will discuss the model properties, critical parameters, and their essential boundaries.

### 7.2.1. Discussion of an Always-on System (Model M1)

Model M1 describes the average power consumption of a sensor system that is always powered on. Such a sensor system has several major advantages: The system design is remarkably simplified compared to the features that need to be implemented in hardware and software for low-power systems M2 or M3. Another advantage that makes data processing easier is seamless or continuous sensing. This feature will be addressed later in detail with model M3.

The disadvantage of an always-on system is the highest power consumption of all three models. However, optimization strategies exist for systems based on model M1. By analyzing equation 7.1 the critical parameters of M1 are the individual parts that contribute to the total average power consumption. Optimizing the individual parts require the selection of energy-efficient hardware components and the design of energy-efficient software. This optimization strategy regarding the hardware and software was addressed in the previous chapters, in particular 3, 5, and 6. All strategies address the reduction of the average power consumption for all parts that contribute to the average power in 7.1. However, in equation 7.1 the dynamic power of the microcontroller $P_{core,dyn}$ depends on the core frequency $f_{core,M1}$. The core clock $f_{core,M1}$ is the critical parameter in model M1. In section 2.3 the fundamentals of digital systems were introduced. This fundamental perspective shows that $f_{core,M1}$ linearly influences the dynamic power consumption. However, in detail, the microcontroller system, particularly low-power optimized systems, face clock limits, where the linearization does not apply. One reason is the flash memory datarate limit, which requires additional clock cycles. Once the limit is reached, rising the core clock frequency $f_{core,M1}$ would not contribute to increasing the processing speed. Thus, the power consumption would rise non-linearly. Therefore, modern microcontrollers have cashing techniques for machine instructions and data. The optimization technique for $f_{core,M1}$ is an optimization between two interests. On the one hand, keeping $f_{core,M1}$ as low as possible to conserve energy. On the other hand, keeping $f_{core,M1}$ at a level, the workload can be processed in a defined time. This time constraint is part of the real-time requirements. Consequently, a fixed core frequency $f_{core,M1}$ requires to be high enough to cover peak processing loads. Thus, an approach exists to vary $f_{core,M1}$ depending on the current workload: clock scaling adapts $f_{core,M1}$ to the current demand. As a result, the core clock $f_{core,M1}$ is set above the required minimum as a reserve for irregular computing loads. This reserve causes a waste of energy and can be addressed better by model M2.

### 7.2.2. Discussion of System Based on Model M2

ULPSEK enters low-power mode automatically controlled by the RTOS whenever the microcontroller is done with processing. This method's advantage is adapting the power consumption to a required minimum. The automation was possible by delegating the decision for entering the low-power to the RTOS. The role of the RTOS is predesignated to decide whenever a low-power state can be entered because the RTOS controls the entire task schedule. The cost of waking-up and the cost for transition to low-power mode is often missing in the literature. This cost was expressed as energy $E_{STAR}$ and time $t_{STAR}$.

Equation 7.3 contains the probability $p_{act}$. The wake-up and sleep states are activated irregularly for multitasking or multi-sensor systems. Thus, this probability is the universal definition for a system of this kind. In special cases, the probability can be simplified. If the sampling process repeats in fixed intervals, then the probability can be replaced by Equation 7.4.

$N_{cy}$ is the number of clock cycles executed in the active state. The competition depends on the processor core clock $f_{core,M2}$. And the regular cycle rate can be expressed by $r_c$. An example for $r_c$ is the sampling rate of the ECG. The sampling and computing process was repeated regularly at fixed intervals of 200 Sa/s. With equation 7.4, $p_{act}$ is specified for signal processing tasks using low-power mode.

The advantage of systems based on model M2 is the automatic adaption to the lowest energy level which can be achieved during the sampling and processing of sensor data. Additionally, the core clock runs at high speed, providing a reserve for high computing loads, such as more data analytics and complex algorithms. Low-power systems implemented according to M2 are remarkable in their flexibility regarding new software features. However, one relationship has to be kept in mind, as more features are activated, more power will be consumed automatically. This property leads to the conclusion that low power must not only be achieved with engineering but also from a product design perspective by managing the power consumption with mandatory features. On the other side, this model should answer how much energy each feature costs.

### 7.2.3. Discussion of a System with Hibernation (Model M3)

Model 3 extends the model by adding hibernation. The key benefit of hibernation is that the sensor system can conserve even more energy. The disadvantage is that during hibernation sensing is paused. The relevant amount of energy conservation was achieved by powering down all electronics, including the microcontroller: sensors, sensor frontends, UI components, a radio module, and even parts of the power distribution network itself. Several technical challenges require engineering solutions for implementing hibernation. The implementation including the challenges was elaborated in chapter 3. However, the benefit must be measured once the hardware and software implement hibernation. The criteria for hibernation being beneficial was given by inequation 7.15 which states that the efficiency factor $\gamma_{hib}$ needs to be below 1. Because of the intention of how $\gamma_{hib}$ was defined, an efficiency factor larger than 1 would lead to higher power consumption as it would be without hibernation. Therefore, the $\gamma_{hib}$ should be checked before using hibernation. In general, $\gamma_{hib}$ is a predictive metric for the efficiency of hibernation. The power efficiency factor was defined with Equation 7.14. In this equation, constants and variables determine the efficiency. The constants are system specific. In the case of ULPSEK these constants were determined by measurements: $P_{sys,M2}$ is the average power consumption required while sensing and processing according to model M2. $P_{hib}$ is the average power consumption while staying in hibernation. $E_{hib,er}$ is the energy required to enter and return from hibernation. $t_{hib,er}$ is the time required to enter and return from hibernation. Additionally, some variables can be controlled from an application perspective: $t_{hib}$ is the time in hibernation. $t_{sens}$ is the time spent with sensing and processing. With efficiency, factor answers can be given from an application and engineering perspective. From an application perspective, $t_{hib}$ and $t_{sens}$ can be proven for system-specific constants. If $t_{hib}$ and $t_{sens}$ are planed to be adaptive then the Equation 7.14 should be implemented in software to predict if hibernation will be beneficial. This reliability test is required each time before hibernating to avoid waste of energy. From an engineering perspective, the system parameters $P_{sys,M2}$, $P_{hib}$, $E_{hib,er}$, and $t_{hib,er}$ can be evaluated for a desired $t_{hib}$ and $t_{sens}$. This way an engineer can prove which part of the system should be optimized to achieve the highest benefit regarding power efficiency.

### 7.2.4. Power Consumption Depends on Physiological Parameters

An interesting finding is that the power consumption of a low-power system depends on the heart rate. This dependency becomes clearer with the following rule which is the key to low-power system design. A low-power system aims to reduce energy demand to a minimum. This functionality was implemented with model M2. Thus, if the heart rate is at rest condition, processing each heartbeat requires less processing load. If the heart rate is doubled or tripled during physical exercise, then the processing load for each beat is rising proportionally.

### 7.2.5. Supply Voltage for the Digital Circuits

It was previously shown theoretically that the supply voltage for digital circuits has a quadratic impact on power consumption in section 2.3. Therefore, the supply voltage ($Vdd$) should be set down as low as possible. The supply voltage was reduced down to 2.1 V in section 6.1 experimentally. Two major findings were experienced. First, with dropping supply voltages, the chance of system instability rises. Second, with dropping supply voltages, the chance to get appropriate hardware parts drops.

An example of system instability was the supply voltage for the microcontroller. 2.0 V was the absolute minimum rating, so if the voltage dropped below 2.0 V the microcontroller became unstable. The EFM32 microcontroller specifies a supply voltage down to 2.0 V volts. A drop is possible due to transient disturbances in the power supply. This instability was confirmed experimentally. Even though the EFM32 microcontroller specifies a supply voltage down to 2.0 V volts, a headroom of 100 mV was added to the supply voltage. Therefore, the voltage was set to 2.1 V.

Another problem a sensor system designer faces is the availability of low-power components. The availability and quality of hardware components drop with the system voltage. ULPSEK consists of many parts requiring a minimum voltage. As a drawback, searching for parts down to 2.1 V was limiting the number of available components e.g. sensor front-ends, especially those with excellent properties. Another solution to overcome these contradictory requirements is to introduce multiple power domains. Therefore, voltage shifters can serve for communication between power domains. However, the capability to shut down power domains independently must be conserved when using voltage shifters. However, ULPSEK design target was to avoid voltage shifters. Instead, enough integrated circuits fit the desired application at a common voltage level of 2.1 V.

### 7.2.6. Sampling Rate Impact

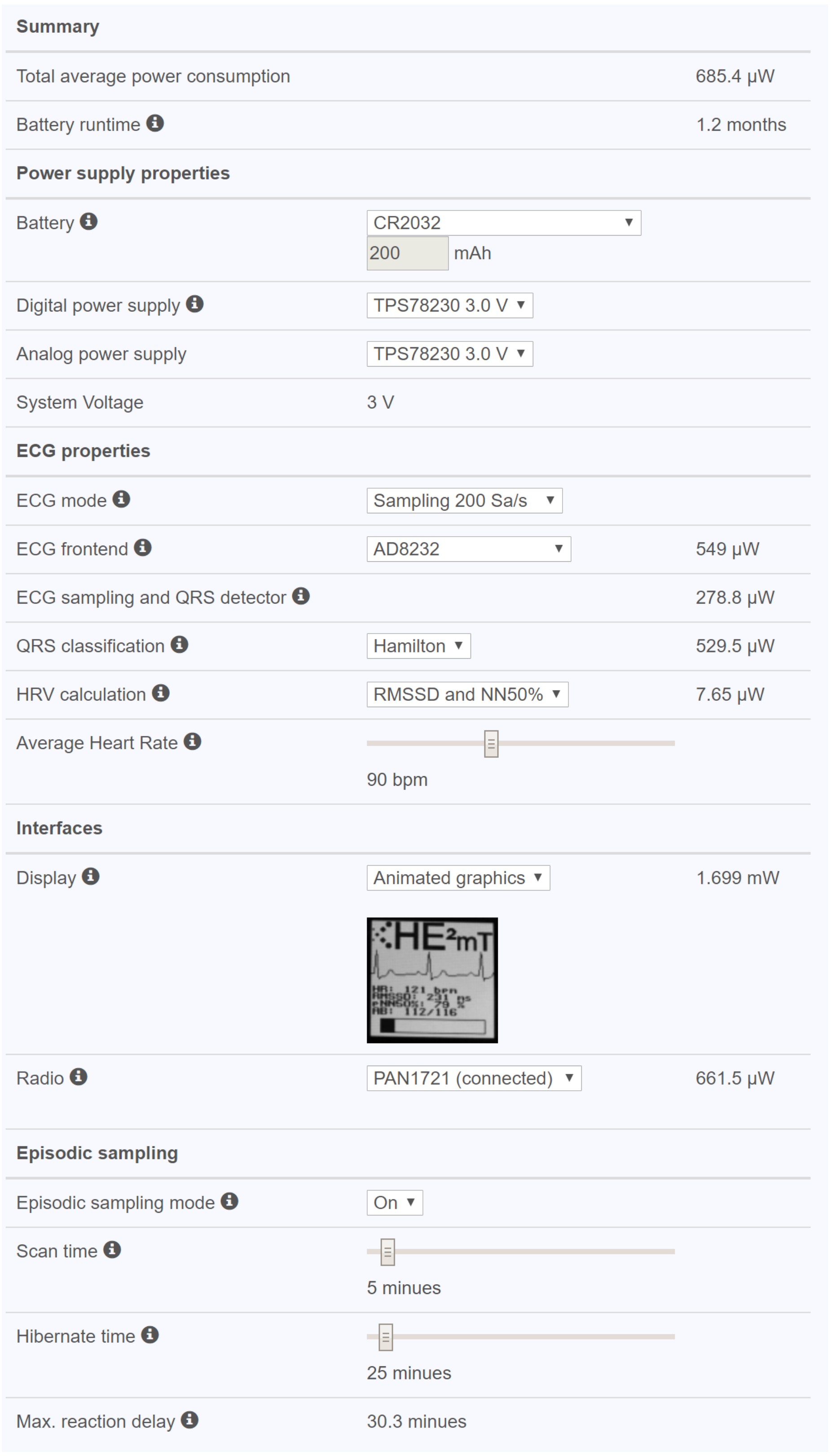
The choice of the sampling rate is a critical decision in a signal processing design. Unnecessary oversampling is a common cause of energy inefficiency. This topic was addressed theoretically in the paper by Tobola et al., "Sampling Rate Impact on Energy Consumption of Biomedical Signal Processing Systems" [76] in detail, which contributes to this thesis. The full motivation, mathematical derivation, and discussion are the content of this paper. An excerpt will be given here. This paper shows the dependency of the computational effort on the sampling rate. It should be obvious, that keeping the sampling rate as low as possible supports conserving energy. Each sample needs to be processed separately. However, the mathematical relationship between the sampling rate and the energy efficiency of algorithms is not always linear. Thus, energy wasting can be over-proportionally high when choosing a higher sampling rate as necessary. The idea of this paper was to use Bachmann-Landau notation in combination with the sampling rate.

In the literature, Bachmann-Landau notation is commonly used to investigate how algorithm complexity scales with the number of data samples. In contrast, this work, along with the corresponding paper, uses the sampling rate in combination with algorithm complexity to analyze scaling behavior. As the data rate increases, the computational complexity per unit time also grows. Some algorithms may additionally require reconfiguration in response to changes in the data rate. Consequently, it can be demonstrated that processing an algorithm at twice the rate may result in up to four times the computational effort. In the context of the M2 model introduction, the computational load in Equation 7.4 increases both the number of computing cycles $N_{cy}$ and the cycle rate $r_c$ based on the specific algorithmic details. The mentioned publication includes a derivation of the algorithms' complexity in relation to the sampling rate, along with a tabular comparison of the complexity of commonly used algorithms. Special consideration is necessary for algorithms that scale superlinearly, as they can lead to a rapid increase in power consumption with higher data rates. The mentioned paper provides an overview of all analyzed algorithms, detailing their computational complexity relative to the sampling rate.

## 7.3. Web-based Battery Runtime Calculator

The energy model was partly implemented as an interactive, web-based application. This application is based on Hypertext Markup Language (HTML), Cascading Style Sheets (CSS), Java-Script, Bootstrap[1] and Knockout[2]. Sensor system parameters such as battery type, voltage regulators, modules, and algorithms for vital parameter computation can be selected. The overall runtime will be shown based on the model above. An example screenshot can be seen in Fig. 7.2. The web-based implementation of the battery runtime calculator is publically available at http://ulpsek.com

[1]Web library Bootstrap, version 4, http://getbootstrap.com
[2]Web library Knockout, version 3.4.0, http://knockoutjs.com



| | | |
|---|---|---|
| **Summary** | | |
| Total average power consumption | | 685.4 µW |
| Battery runtime | | 1.2 months |
| **Power supply properties** | | |
| Battery | CR2032 ▾ 200 mAh | |
| Digital power supply | TPS78230 3.0 V ▾ | |
| Analog power supply | TPS78230 3.0 V ▾ | |
| System Voltage | 3 V | |
| **ECG properties** | | |
| ECG mode | Sampling 200 Sa/s ▾ | |
| ECG frontend | AD8232 ▾ | 549 µW |
| ECG sampling and QRS detector | | 278.8 µW |
| QRS classification | Hamilton ▾ | 529.5 µW |
| HRV calculation | RMSSD and NN50% ▾ | 7.65 µW |
| Average Heart Rate | 90 bpm | |
| **Interfaces** | | |
| Display | Animated graphics ▾ | 1.699 mW |
| Radio | PAN1721 (connected) ▾ | 661.5 µW |
| **Episodic sampling** | | |
| Episodic sampling mode | On ▾ | |
| Scan time | 5 minues | |
| Hibernate time | 25 minues | |
| Max. reaction delay | 30.3 minues | |

**Figure 7.2.:** Screen shot of the web-based power and battery runtime estimation tool available at http://ulpsek.com.

## 7.4. Chapter Summary

The model reflects three modes of operation M1, M2, and M3. Then the key parameters were discussed for power reduction. The next chapter deals with a general theory of information processing efficiency that goes beyond this model in terms of abstraction.

# Chapter 8

# Theory of Energy-efficient Information Processing

This chapter utilizes existing methods from information theory for energy efficiency. Section 8.1 and 8.2 applies state-of-the-art methods to the data gained in this work. Section 8.3 extends the state-of-the-art method in chapter 8.2 by proposing a new measure for the system design regarding energy. Section 8.4 and 8.5 propose a new scale for signal processing efficiency: the Landauer-Decibel. In section 8.8 and 8.9 the energy efficiency metrics were applied to different areas of interest. In section 8.10 the technology progress for digital circuits was rated with Landauer-Decibel. Section 8.11 introduces a metric for systems that can adapt their energy efficiency to the required minimum.

## 8.1. Shannon's Channel Capacity

Shannon's Channel Capacity will be the basis for the theories in this chapter. Therefore, a short introduction to this fundamental theorem will be given here. Shannon proposed his Channel Capacity in the publication "A Mathematical Theory of Communication" in 1948 [217]. Shannon's work extended Nyquist's [220] and Hartley's [218] definition of symbol rates. Shannon's Channel Capacity describes a transmission channel's maximum bit rate for a given bandwidth and a given SNR.

$$C = B \cdot log_2 \left(1 + \frac{S}{N}\right) \tag{8.1}$$

The above equation assumes equally distributed SNR over the bandwidth $B$. This distribution applies to an Additive White Gaussian Noise (AWGN) channel. Shannon's channel capacity has become a fundamental theorem in communication engineering. Likewise, Shannon's channel capacity will be the basis for the methods utilized in this chapter.

## 8.2. Overhead Factor

A system optimization based on Shannon's Channel Capacity was proposed by Fischer [161]. Fischer introduced this method for analyzing the complexity and cost of systems in mobile radio communications with the patent EP1521378B1. This patent introduced an indicator for system design efficiency: the overhead factor.

The overhead factor will be explained by applying the method to the system model from this patent. Therefore, the overhead factor has to be computed for every signal processing block. It is required to determine the Service Rate $r_{service}$ first in order to calculate the overhead factor. The service rate measures the data rate with the information required for the application. However, the service rate does not include the overhead caused by redundancy and irrelevant information. $r_{service}$ is then used for computing the overhead factor for every signal along the signal processing chain. Two different formulas were given for analog and digital signal processing.

$$O_i = \frac{C_i}{r_{service}} \tag{8.2}$$

For digital signal processing blocks, the overhead factor was defined as a product of sampling rate $S$, resolution $n_B$, and several signal streams $n_{sB}$ normalized by $r_{service}$.

$$O_i = \frac{S_i \cdot n_{B,i} \cdot n_{sB,i}}{r_{service}} \tag{8.3}$$

For analog signal processing blocks, the overhead factor is Shannon's channel capacity according to equation 8.1 normalized by $r_{service}$. Further, Fischer's publication proposes a graphical analysis method on a logarithmic scale, with analog signal blocks on the left side and digital signal blocks on the right side. The aim of this method is that the overhead has to decay as fast as possible with every signal processing stage. This method was designed and shown as an example in communications. This method will be applied to the system and the data from this work. More specifically, for the analysis with overhead factor, three configurations of ULPSEK were selected for investigation. Every configuration can be seen as a separate signal processing chain. The three selected signal processing chains were previously investigated in the experimental chapter 5. Fig. 8.1 represents the three signal processing chains prepared for the analysis with the overhead factor.

In Figure 8.2 the overhead factor was computed for three hardware and software configurations in this work: S1000V, S200V, and the comparator mode. Table 8.1, 8.2, and 8.3 contain the equations and parameters for each signal processing stage and each system configuration. The mode S1000V stands for sampling at 1000 Sa/s in the wideband mode for high resolution of raw data. In mode S1000V the analog ECG front-end was set to a bandwidth of 130 Hz. With 500 Hz, the Nyquist Frequency in this mode is far above 130 Hz to ensure enough space for the roll-off of the second-order low-pass filter. The oversampling was used in S1000V, ensuring clear raw ECG recordings with low aliasing. Similarly, the S200V mode also utilizes sampling, but at 200 Sa/s and with a narrowband filter at 20 Hz bandwidth.

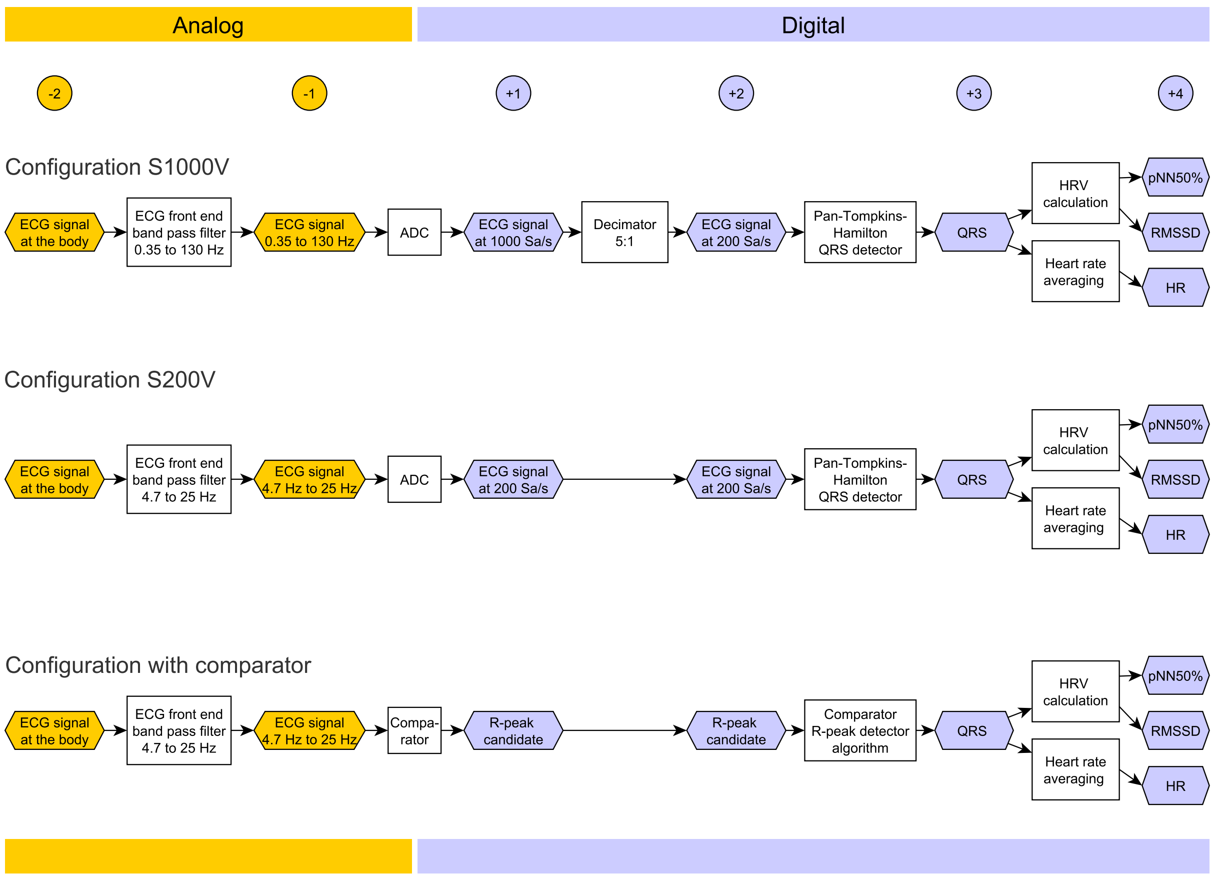


**Figure 8.1.:** This figure contains three signal flow diagrams. Each signal flow diagram represents a configuration for the ECG signal processing implemented in hardware and software using the ULPSEK. The configurations were mode S1000V, mode S200V, and the comparator mode. Detailed instructions for this type of signal flow diagram are included in the appendix A. On the top, a numbering system was added with negative numbers for analog signals and positive ones for digital signals according to the definition of the Overhead Factor Method [161].

**Table 8.1.:** The overhead factor ($O_i$) was computed for each signal ($i$) using the configuration S1000V with the equations and parameters shown in this table.

| Signal (i) | Equation | Parameters | Bitrate | $O_i$ |
|---|---|---|---:|---:|
| -2 | 8.2 | B=2 kHz, SNR=25 dB | 8462.7 bps | 2644.6 |
| -1 | 8.2 | B=130 Hz, SNR=35 dB | 759.0 bps | 237.2 |
| 1 | 8.3 | S = 1000 Sa/s, $n_B$ = 12 bit, $n_{sB}$ = 1 | 12 000.0 bps | 3750.0 |
| 2 | 8.3 | S = 200 Sa/s , $n_B$ = 16 bit, $n_{sB}$ = 1 | 3200.0 bps | 1000.0 |
| 3 | 8.3 | S = 2 Sa/s, $n_B$ = 8 bit, $n_{sB}$ = 1 | 16.0 bps | 5.0 |
| 4 | 8.3 | S = 0.2 Sa/s, $n_B$ = 8 bit, $n_{sB}$ = 2 | 3.2 bps | 1.0 |

**Table 8.2.:** The overhead factor ($O_i$) was computed for each signal ($i$) using the configuration S200V with the equations and parameters shown in this table. Signal number 2 does not exist in this configuration.

| Signal (i) | Equation | Parameters | Bitrate | $O_i$ |
|---|---|---|---:|---:|
| -2 | 8.2 | B=2 kHz, SNR=25 dB | 8462.7 bps | 2644.6 |
| -1 | 8.2 | B=20.3 Hz, SNR=37 dB | 125.2 bps | 39.1 |
| 1 | 8.3 | S = 200 Sa/s, $n_B$ = 8 bit, $n_{sB}$ = 1 | 1600.0 bps | 500.0 |
| 2 | - | - | - | - |
| 3 | 8.3 | S = 2 Sa/s, $n_B$ = 8 bit, $n_{sB}$ = 1 | 16.0 bps | 5.0 |
| 4 | 8.3 | S = 0.2 Sa/s, $n_B$ = 8 bit, $n_{sB}$ = 2 | 3.2 bps | 1.0 |

**Table 8.3.:** The overhead factor ($O_i$) was computed for each signal ($i$) using the configuration the comparator mode with the equations and parameters shown in this table. Signal number 2 does not exist in this configuration.

| Signal (i) | Equation | Parameters | Bitrate | $O_i$ |
|---|---|---|---:|---:|
| -2 | 8.2 | B=2 kHz, SNR=25 dB | 8462.7 bps | 2644.6 |
| -1 | 8.2 | B=20.3 Hz, SNR=37 dB | 125.2 bps | 39.1 |
| 1 | 8.3 | S = 2 Sa/s, $n_B$ = 8 bit, $n_{sB}$ = 1 | 16.0 bps | 5.0 |
| 2 | - | - | - | - |
| 3 | 8.3 | S = 2 Sa/s, $n_B$ = 8 bit, $n_{sB}$ = 1 | 16.0 bps | 5.0 |
| 4 | 8.3 | S = 0.2 Sa/s, $n_B$ = 8 bit, $n_{sB}$ = 2 | 3.2 bps | 1.0 |

It can be seen that the overhead factor increases in the critical transition from the analog to the digital domain in configurations S1000V and S200V. Unlikely, when using the comparator, the data rate reduces early. With Fischer's diagram, the overhead factor was visualized. The overhead factor starts at the same point and ends at the same point for every three of the selected solutions. This similarity is required for the comparison of the three use cases. This precondition was forced using three different implementations of a signal processing chain. All three implementations required equal input and output

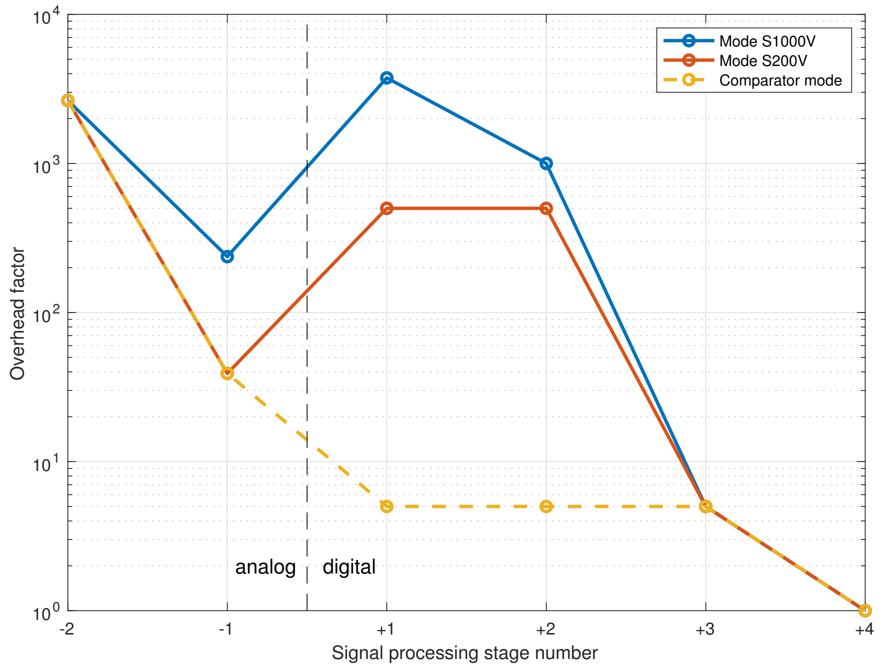


**Figure 8.2.:** The overhead factor, according to Fischer [161], was applied to the results of this work in three cases: Mode S1000V, S200V, and the comparator mode. The assumption is that the faster the decay of the overhead factor with every processing stage, the better for energy efficiency. Table 8.1, 8.2, and 8.3 contain the equations and parameters for each signal processing stage and each system configuration.

interfaces. Thus, in all three cases, the input is an ECG signal and the output is a heart rate parameter. All three implementations were tested with an ECG simulator for delivering the heart rate parameter for a given ECG pattern and different heart rates. However, this test was similar to rest conditions. In particular, the accuracy of the heart rate parameter is more robust in the sampling modes S1000V and S200V when testing with motion artifacts. For this experiment, rest conditions were assumed and the heart rate was set to 80 bpm. S1000V has the highest power consumption with 1053 µW, followed by S200 with 585 µW, and finally the comparator solution with 414 µW. Comparing the decay of the overhead factor in the graph with the average power consumption confirms the expectation that reducing the rate as early as possible leads to low-power designs.

## 8.3. Introducing Overhead Factor Design Efficiency

So far the overhead factor method was applied to the data gained with the system developed in this work. In figure 8.2 a correlation between the energy efficiency of the system and the decay of the overhead factor can be seen. This decay is also an expected relationship. Reducing the data rate in a signal processing chain as early as possible enables designing the following stages more energy efficiently. Thus, a fast decay of the overhead factor is wanted for energy efficiency.

With this, it would be interesting to define a quantitative indicator for the system design efficiency according to the overhead factor. The following definition is a proposal for extending the overhead factor method presented by Fischer. The Overhead Factor Design Efficiency is defined as the maximum of the overhead factor, normalized to the sum of the overhead factor.

$$\gamma_{decay} = \frac{max_i(O_i)}{\sum\limits_i O_i} \tag{8.4}$$

The interval of $\gamma_{decay}$ is within $]0,1[$. Towards 0 the system design solution is better regarding earlier rate reduction. On the contrary, towards 1 the system design solution is worse.

With this definition, it would be interesting to compute $\gamma_{decay}$ for every system configuration to compare it with the power consumption. Table 8.4 compares $\gamma_{decay}$ against the power consumption of ULPSEK. In these three cases, there is a linear relationship between power and $\gamma_{decay}$ . In this example, the $\gamma_{decay}$ correlates with the power linearly. Thus it can be stated that, the earlier the rate reduction, the lower the total system power. To conclude, the Overhead Factor Design Efficiency ($\gamma_{decay}$ ) appears to be a valuable indicator of system efficiency for system designs beyond this thesis.

**Table 8.4.:** The overhead factor design efficiency rating in relation to the power consumption for three signal processing chain configurations (S200V, S1000V, and Comparator).

| **Configuration** | $\gamma_{decay}$ | **Power** |
|---|---:|---|
| S1000V | 0.98 | 1053 µW |
| S200V | 0.78 | 585 µW |
| Comparator | 0.38 | 414 µW |

## 8.4. Introducing the Informational Energy Efficiency

Metrics are used for expressing energy efficiency in communication theory and communication engineering. Such metrics are given as the energy required to transmit one bit. This section proposes a metric for calculating energy per bit for a signal processing unit under investigation. An important step is to circumference or isolate the signal processing unit under investigation, so the parameters are determined correctly. This isolation can be done for every single subsystem or on a large scale for a whole system, depending on how the boundaries are chosen. Once the system under investigation is isolated, three values are required for every isolated block: (1) Input data rate $r_{in}$, (2) output data rate $r_{out}$, and (3) the power $P_{sui}$ required for the signal processing block. Figure 8.3 illustrates an isolated signal processing unit with a known data rate at the input and output and the power required for the signal processing unit.

Once the three values are known, the Informational Energy Efficiency $\Gamma_{IEE}$ can be computed. Equation 8.5 introduces the power efficiency metric $\Gamma_{IEE}$, which is calculated as the power of the system under investigation $P_{sui}$ divided by throughput bit rate $r_t$.

$$\Gamma_{IEE} = \frac{P_{sui}}{r_t}, \left[\frac{W \cdot s}{bit} = \frac{J}{bit}\right] \tag{8.5}$$

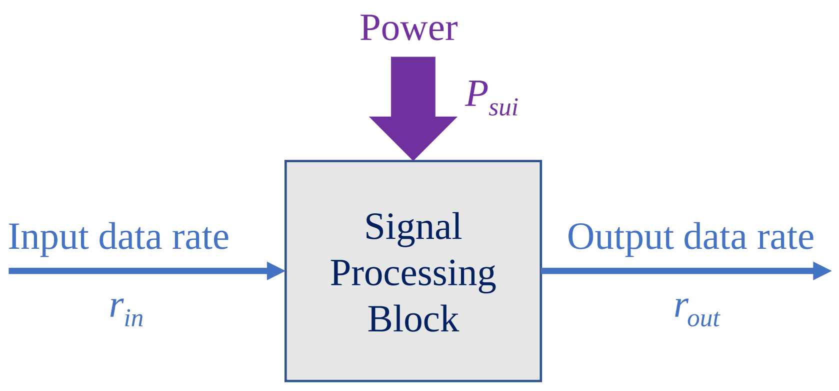


**Figure 8.3.:** To determine the Informational Energy Efficiency $\Gamma_{IEE}$ a boundary around a signal processing block must be defined. Once the boundary is defined, three values are required: (1) Input data rate $r_{in}$, (2) output data rate $r_{out}$, and (3) the power $P_{sui}$ required for the signal processing block.

The power of the system under investigation $P_{sui}$ has to be measured or estimated theoretically for each signal processing block. For the throughput bitrate $r_t$ in Equation 8.5, multiple solutions exist for combining input bitrate $r_{in}$ and the output bitrate $r_{out}$ to get a representable $r_t$. The final decision is Equation 8.6. It defines the throughput bitrate $r_t$ as the minimum of the input bitrate $r_{in}$ and the output bitrate $r_{out}$. Thus, the smaller number equals the throughput bitrate $r_t$.

$$r_t = \mathbf{min}\left(r_{in}, r_{out}\right) \tag{8.6}$$

Finding a practicable definition for the throughput rate was not trivial. Here is the complete argumentation to how Equation 8.6 became the favorite solution. The first solution was to use an average data rate. Unfortunately, the average did not represent the system very well for signals with high data rate reduction from $r_{in}$ to $r_{out}$. The second solution targeted a weighted integral over the signal processing inside the signal processing block. This solution was mathematically the best choice but not practicable because of its complexity. A third solution was to keep the input and output rates and calculate two efficiency values for both. Both input and output efficiency were interesting numbers to compare for different signal processing blocks. However, the output rate turned out to rate the signal processing better. The only drawback was that if a signal block increased the bitrate $r_{in}$ to $r_{out}$, which is technically required for some signal processing units, the efficiency improved. However, if a signal block increases the data rate, it does not generate any additional information, which should not be rated positively. The solution was to define a minimum of both $r_{in}$ to $r_{out}$ with the following argumentation: If the data rate decreases from input to output, which is the desired case for low-power systems, then the $r_{out}$ is the throughput bitrate of interest, with the assumption that the irrelevant and redundant information was removed. On the contrary, if the data rate increases from input to output, which is technically required sometimes, then the $r_{in}$ is the throughput bitrate of interest, with the assumption that irrelevant or redundant information was added. Thus, the best and fortunately trivial solution is Equation 8.6.

In conclusion, with Equation 8.5 the Informational Energy Efficiency ($\Gamma_{IEE}$) can be computed for any system under investigation with defined boundaries. The units of $\Gamma_{IEE}$

are given in Joule per bit. Thus, alternatively, a fraction of energy and total data transited might be used. However, using power and bitrate was decided to be more practical. Informational Energy Efficiency $\Gamma_{IEE}$ will be used further as defined above.

## 8.5. Introducing Landauer-Decibel

The Informational Energy Efficiency ($\Gamma_{IEE}$) was introduced in the previous section. $\Gamma_{IEE}$ was defined as energy per bit for a signal processing unit under investigation. This section introduces the Landauer-Decibel, an alternative representation of $\Gamma_{IEE}$. For transforming $\Gamma_{IEE}$ into Landauer-Decibel, two steps are required:

1. Normalizing $\Gamma_{IEE}$ to Landauer's energy limits
2. Using a logarithmic scale in Decibel (dBLa)

The idea of introducing Landauer-Decibel is to set the absolute minimum for computing to 0 dBLa and be able to express the distance to this absolute minimum. Further, the progress of technology can be expressed in Landauer-Decibel per year while approaching this fundamental physical minimum towards 0 dBLa. The absolute minimum for computing is known as Landauer's Principle. In 1961, Landauer [216] defined a theorem for the absolute limit for the energy required for inverting the state of a bit.

$$E_L = k_B T \cdot ln(2) \tag{8.7}$$

This bit inversion depends on the Temperature $T$, and $k_B$ is the Boltzmann constant. This portion of the energy is the absolute computational physical limit required for inverting a bit. Landauer's work was confirmed experimentally. Lastly, in the year 2016, Hong et al. [59] confirmed Landauer's principle underlying the relevance of this discovery with the words: “This result reinforces the connection between *information thermodynamics* and physical systems and also provides a foundation for the development of practical information processing technologies that approach the fundamental limit of energy dissipation.“ With this, Landauer's principle is an interesting and practical metric to normalize energy efficiency by the absolute limit.

The second step towards Landauer-Decibel is using a logarithmic scale. With this, the full definition of the Landauer-Decibel is defined as the $\Gamma_{IEE}$ normalized by Landauer's energy minimum $E_L$ on a logarithmic scale:

$$\Gamma_{dBLa} = 10 \cdot log_{10}\left(\frac{\Gamma_{IEE}}{E_L}\right) \tag{8.8}$$

With equation 8.5 for $\Gamma_{IEE}$ and equation 8.7 for $E_L$, Landauer-Decibel can be expressed as follows.

$$\Gamma_{dBLa} = 10 \cdot log_{10}\left(\frac{P_{sui}}{r_t k_B T \cdot ln(2)}\right) \tag{8.9}$$

Landauer-Decibel is a scale for expressing the energy efficiency of a signal processing unit relative to the absolute physical limit. In general, the lower the number, the more efficient the system. The absolute limit was defined at 0 dBLa.

## 8.6. Temperature Dependency of Landauer-Decibel

It should be noted that Landauer's principle depends on temperature. Therefore, when using the Landauer-Decibel, this dependency requires some attention. This section shows when temperature dependency is more or less relevant for using the Landauer-Decibel. Therefore, three cases were examined:

1. At first, using 0 K is not possible in the universe we know. Moreover, Landauer-Decibel is not defined for $T = 0\,\mathrm{K}$. Therefore, this approach was dropped.
2. A second approach was using the temperature of our known universe. Even if the background radiation is given as the universe's temperature with $T = 2.7\,\mathrm{K}$, there is no definition for temperature that can be assigned to space. Besides this, even if a temperature close to 0 K could be defined, a fundamental drawback of this approach would be that most systems are not specified for operating at a temperature close to 0 K.
3. Finally, a third approach was to define the Landauer-Decibel in the range of the system's operating temperature. This approach will be used for further investigations here in this work.

So the third approach assumes that each computing system has an operating temperature range. In this work, wearables were the systems of investigation. Wearables are devices worn at the surface of a human body's skin around 30 °C. Therefore, a wearable's electronic parts operate close to this temperature. Consequently, a homogenous temperature of 30 °C can be expected. However, self-heating and ambient temperature caused temperature gradients during the experiments in the system's signal processing parts.

Knowing the order of magnitude of the temperature gradient that causes a relevant effect on energy efficiency is essential. A concrete question could be as follows: How far must the operating temperature decrease or increase, inducing a change of 1 dBLa or −1 dBLa, respectively? This question can be answered theoretically by utilizing equation 8.9. As a result, an operating temperature of −36 °C would shift the efficiency by 1 dBLa. Conversely, an operating temperature of 106 °C would shift the efficiency by −1 dBLa. This example shows what temperature gradients are required to influence computing efficiency by 1 dBLa for a wearable system. If temperature gradients in the wearable system are expected within the estimated range above, then the efficiency may increase or drop by less than 1 dBLa. To conclude, for further usage of the Landauer-Decibel in this thesis, an average temperature of 30 °C was selected for wearable computing. With this decision, deviation by ±1 dBLa has to be expected within a temperature range between −36 °C and 106 °C.

## 8.7. Graphical Representation of the Energy Efficiency Metric

Inspired by the German energy pass, a proposal for a graphical representation of the throughput energy efficiency scale was designed. The graphical representation has two scales: The scale in the center shows the informational energy efficiency in Watts per bit. Additionally, on the top, the scale shows the corresponding Landauer-Decibel for the system's operating temperature. The results will be discussed in the next section.

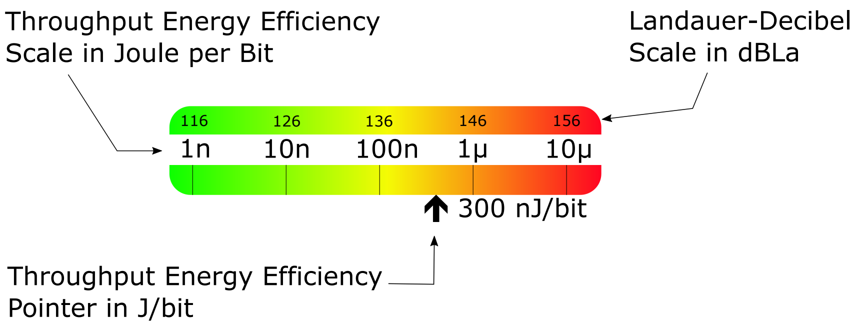


**Figure 8.4.:** An example graphical representation of the energy efficiency scale. This example scale has a range from 1 nW/bit to 10 µW/bit. This corresponds to a Landauer-Decibel scale from 116 dBLa to 156 dBLa.

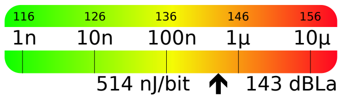
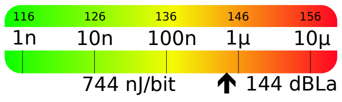
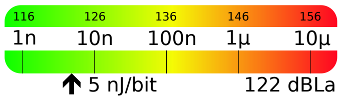
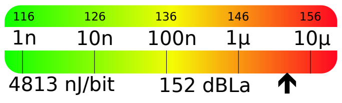
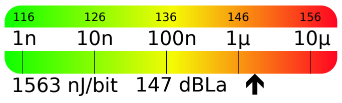
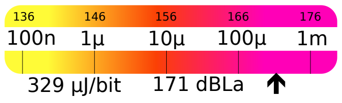
This scale representation will be used later to visualize the efficiency of individual system parts and the entire system. Later the boundaries of the scale will also be shifted depending on the magnitude of the efficiency under investigation.

## 8.8. Efficiency Metrics applied to ULSPEK

The efficiency metrics Informational Energy Efficiency (IEE) and Landauer-Decibel were defined above. These metrics were applied to ULPSEK with the known data and model from the previous chapters. The results are listed in table 8.5.

Table 8.5 illustrates that the QRS detector has the worst energy efficiency due to its multiple processing stages; despite being one of the more efficient detectors, its overall power consumption remains relatively high while operating at a low bit rate. The Pan-Tompkins algorithm for QRS detection involves five stages: bandpass filtering, differentiation, squaring, moving window integration, and thresholding with peak detection. This may explain the relatively poorer efficiency: the data rate reduction occurs late in the fifth processing stage, despite drastically reducing signal bandwidth in the first stage. In contrast, the decimator achieves the best energy efficiency by processing at a higher througput rate, resulting in a greater efficiency. This is accomplished through its FIR filtering, which reduces the sampling rate immidiatly.

**Table 8.5.:** This table shows the IEE for the signal processing chain in the configuration S1000V. The table contains the parts of the signal processing chain in each row, with the total in the last row. The columns start with the name of the signal processing part, followed by the input and output data rate, the throughput data rate according to equation 8.6, and the Informational Energy Efficiency according to section 8.4 visualized with the scale from section 8.7.

| **Part** | **Power** $P_{sui}$ | **Rate** $r_{in}$ **and** $r_{out}$ | **Throughput Rate** $r_t$ | **Efficiency Scale** |
|---|---|---|---|---|
| ECG front end | 390 µW | $r_{in}$=8462.7 bps<br>$r_{out}$=759 bps | 759 bps |  |
| ADC + SR | 565 µW | $r_{in}$=759 bps<br>$r_{out}$=12 000 bps | 759 bps |  |
| Decimator | 16 µW | $r_{in}$=12 000 bps<br>$r_{out}$=3200 bps | 3200 bps |  |
| QRS detector | 77 µW | $r_{in}$=3200 bps<br>$r_{out}$=16 bps | 16 bps |  |
| HR and HRV algorithm | 5 µW | $r_{in}$=16 bps<br>$r_{out}$=3.2 bps | 3.2 bps |  |
| Overall system | 1053 µW | $r_{in}$=8462.7 bps<br>$r_{out}$=3.2 bps | 3.2 bps |  |

## 8.9. Expressing Present Technology Efficiency with the Landauer-Decibel

In the previous section, the Landauer-Decibel was utilized to express the efficiency of several parts of a signal processing chain. All signal processing parts are based on some technology that can be expressed as an offset from the Efficiency-Minimum at 0 dBLa. Technology, analog and digital hardware, and software such as algorithms or an RTOS determine the overall energy efficiency according to the results of previous chapters. This section aims to find some limits of the present technology with data from literature and measurements. The best-performing microcontroller which was used in this thesis was the EFM32WG990F256. From a theoretical perspective, it was interesting to determine the perfect algorithm efficiency for this 32-bit microcontroller type. The core computes at 66.1 µA/MHz according to the data from the datasheet. This computation efficiency corresponds to a technology limit of 98 dBLa. Thus, algorithms can not be better as 98 dBLa with this microcontroller. With the data from the measurements, the efficiency of the algorithms with this microcontroller can be transformed to Landauer-Decibel. The efficiency of the algorithms with this microcontroller ranged between 111 dBLa and 148 dBLa. Consequently, from a relative point of view, the best algorithm was 13 dBLa above this technology limit. The worth algorithm was 50 dBLa above this technology limit. The wide range of energy efficiency in algorithm implementation comes through abstraction. Some algorithms were design abstract with the purpose of better reusability. The additional software layers require additional computing effort. Additional computing effort leads to less energy efficiency. Another efficiency gain can be achieved by utilizing the parallel computing capabilities of the microcontroller. Therefore, a software library exists for the comfortable usage of this technology. Optimization by using ARM Cortex Microcontroller Software Interface Standard (CMSIS)-DSP library has improved the energy efficiency up to −3 dBLa. This advantage comes with the limitation that algorithms have to use the 16 bit fixed point data type instead of 32 bit integer or floating point. Additional −3 dBLa may be gained in case the 8 bit fixed point data type would fit the application.

So far, the best-performing microcontroller has been taken into account. However, one of the results from chapter 5 was the impact of microcontroller and compiler combinations on energy efficiency. Applying the Landauer-Decibel to these results will lead to the following data. The investigated selection of different microcontrollers impacts the power consumption up to 7.0 dBLa for the IAR compiler. The choice of the microcontroller would have an even more intense impact if GCC were the compiler of choice with a gain of up to 14.6 dBLa. The impact of a compiler for the best-performing microcontroller would have an impact of 0.5 dBLa. The compiler impact for the Cortex-M3 is higher with 1.0 dBLa. The Cortex-M0+ compiler impacted the efficiency by 2.3 dBLa.

So far, digital hardware and algorithms have been rated. In this application also, analog hardware was utilized. Previously, the ECG frontend AD8232 was identified as an inefficient signal processing part for this application with 143 dBLa. Meanwhile, the manufacturer introduced a replacement part, the AD8233, with the same bandwidth and noise properties but less power consumption. The pin-compatible part reduces the power consumption from 390 µW to 111 µW. Consequently, the energy efficiency would improve by −5.7 dBLa by replacing the integrated circuit with a newer part.

## 8.10. Expressing Digital Technology Progress with the Landauer-Decibel

In the previous section, an example was given how a new release of an integrated circuit could contribute to better energy efficiency. This section will review the progress of digital technology in a general manner. Advances in digital technology improve non-linearity, which makes Landauer-Decibel predestinated for using it as a measure. Koomey et al. [129] published an exciting work on computational efficiency in 2001. They found that the number of computations per Watt doubled every 1.53 years. The exponential growth fits the data with a correlation coefficient $R^2$ =0.983 from 1975 to 2009. This growth can be expressed using the introduced scale: In Landauer-dB this technology improvement corresponds to −2 dBLa per year. Later in 2016, Koomey reexamined the data again. He found that since 2000 the number of computations per Watt only doubled every 2.6 years. This corresponds to −1.2 dBLa per year. The reason for the slowdown can be explained by Moore's law and Dennard's scaling. Moore observed in 1965 [215] that the number of transistors per area doubles every two years. While he extrapolated the data giving a forecast until 1975, the model holds so far until 2023. Later in 1974, Dennard et al. [214] published an analysis about the scaling of MOSFETs. He predicted that the energy density per area would stay constant. More importantly, he predicted that the energy consumption scales with the technology size $\kappa$ by $1/\kappa^2$. Combining both would lead to the following conclusion. Moore's doubling leads to twice as many transistors per area over two years. Assuming the doubling in two dimensions, the transistor size decreases by $\kappa = \sqrt{2}$ every two years. Moore's law in Dennard's scaling leads to half energy per transistor every two years. With this, the number of computations per Watt would double every two years. This rate is in the range of Koomey's prediction. Consequently, Moore's, Dennard's and Koomey's predictions align.

While the predictions were true for decades, the validity could not hold forever. Moors' law could hold until today. Conversely, Dennard's scaling came to a limit due to leakage currents. With this in mind, the scaling models for MOSFETs were updated by Frank and Dennard et al. [173] in 2001. And with the end of Dennard's scaling also Koomey's slowdown since 2000 can be explained. For instance, the leakage currents of MOSFETs are the reason for the slowdown from $-2\,$dBLa to $-1.2\,$dBLa per year. The leakage was addressed in ULPSEK by implementing low-power modes, by clock and power supply gating.

## 8.11. Ideal Adaptive Energy Efficient Design

This consideration is limited to signal processing parts that handle different data rates or at least two discrete or variable data rates. A property of a well-designed signal processing part of this kind is adapting the power consumption according to a variable data rate. Consequently, an ideal design has a constant energy efficiency $\Gamma_{dBLa}$ independent from the throughput data rate $r_t$. The following law could describe such an ideal system.

$$\frac{\delta\Gamma_{IEE}\left(r_t\right)}{\delta r_t} = 0 \tag{8.10}$$

Therefore, $\Gamma_{IEE}$ should exist as a differentiable function, and $r_t$ should be greater 0. ULPSEK achieved this behavior by implementing sleep in idle at least for the sampling and data pre-processing of the ECG. This almost ideal behavior was achieved by implementing the system according to model M2 in subsection 7.1.2. However, as a whole, ULPSEK does not show this behavior. For example, the analog ECG frontend had an adjustable throughput data rate $r_t$, achieved with adjustable filters. This system part consumes the same power regardless of the throughput data rate $r_t$, which makes it not adapt to the current situation from an energy efficiency perspective. Thus the deviation according to Equation 8.10 is negative for the ECG frontend. However, Equation 8.10 is proof for an ideal system design regarding adaptation, and thus another interesting energy efficiency test. It is recommended to use this test for individual parts of the signal processing chain rather than for whole sensor systems.

## 8.12. Chapter Summary

This chapter applied methods from communications engineering and information theory to low-power signal processing. Existing metrics were used, such as the Overhead Factor. It could be shown how this method can be applied beyond the communication system to a wearable system. Additionally, the Overhead Factor was extended by the proposed Overhead Factor Design Efficiency.

Further, two representations of a metric for energy efficiency were defined. First, the informational energy efficiency $\Gamma_{IEE}$ was introduced to express the energy required for processing one bit. Second, the Landauer-Decibel was defined by normalizing $\Gamma_{IEE}$ according Landauer's principle and mapping to a logarithmic scale.

Energy efficiency expressed by the Landauer-Decibel starts at the fundamental limit at 0 dBLa. From this limit, a technology overhead was determined at 98 dBLa for the best microcontroller used in this thesis. It could be shown that suboptimal microcontroller choice may impact up to 7.0 dBLa on energy efficiency. Also, suboptimal compiler choice may impact energy efficiency up to 2.3 dBLa. Using parallel computing on the microcontroller with 16 bit data type improved the energy efficiency by −3 dBLa in the best case. Additional efficiency overhead was caused by the design of the algorithms in the range of 13 dBLa and 50 dBLa. In general, energy efficiency improves with progress in technology. This progress was expressed as a negative rate, currently at −1.2 dBLa per year. With this in mind, it should be noted that wrong design decisions are comparable with setting back the technology decades in evolution.

To conclude, the highest impact on energy efficiency was caused by (1) the design of algorithms and the (2) the technology offset of the hardware used for the computing task. The first issue can be addressed by software design focusing on energy efficiency. The second issue improves along with state-of-the-art technology. As a general rule, only what can be measured can be improved. Landauer-Decibel can be an appropriate metric to express and compare energy efficiency.

# Chapter 9

# Implied Design Rules

In this chapter, the most remarkable findings were collected in a short list of design for low-power rules. These rules were selected carefully based on the research related to this work and supported by the experience in several projects in the industry. The list is a short selection of rules starting with the most effective first.

**Design Rule 1: Using Signal Flow Diagrams** A detailed Signal Flow Diagram is the most effective tool for energy optimization. A signal flow shows the path of the signals through a system. In this PhD work and also in industrial projects beyond this work, signal flow diagrams were used extensively. In particular, a special type of signal flow diagram was introduced. The specification for this type of diagram is contained in Appendix A.

However, even any kind of signal flow diagram is better than improving energy efficiency without it. Unfortunately, for several reasons engineers tend to simplify signal flow diagrams or even skip them completely. This is a fatal underestimation of the most powerful tool for energy optimization, often with tragic consequences for the energy budget and overall system performance. As an often seen consequence, even if a functional system may be delivered in time and within budget, the power requirements for the sensor system were not met by far. Therefore, an architect of the signal processing system requires a complete and detailed overview of the whole signal path. Moreover, the architect needs to communicate this signal path with an interdisciplinary team and the customer effectively. The signal flow diagram should be less seen as a document than as a communication tool. The purpose of this tool is to identify incorrect requirements, engineering errors, and unnecessary energy consumption by systematically matching each individual signal processing component to its corresponding context within the signal processing chain, step by step.

A complete and detailed signal flow diagram is mandatory for understanding and optimizing the entire signal processing chain for low-power applications. Optimally, this can be achieved by utilizing the proposed signal flow chart according to Appendix A.

**Design Rule 2: Developing from Back to Front** It may seem intuitive to develop a signal flow diagram from the sensors towards a point where the data is consumed. However, it is more efficient in terms of saving time and budget to do it the other way around, by starting at the consumer of the data and working towards the sensors. This is especially critical for low-power system design, as the point at which data is consumed determines the requirements for the overall design of the low-power optimized hardware and software design. Therefore, signal flow diagrams were used extensively to analyze and optimize the power consumption, developing them in reverse order.

**Design Rule 3: Data Rate Reduction to the Lowest Necessary Minimum** Data rate reduction was the key to low-power design in almost all chapters of this thesis. An early data rate reduction along the signal processing chain was the key to success. Therefore, in section 8.3, a measure for early data rate reduction was introduced. However, many problems in engineering actually do not have their root in engineering. Such a problem is the idea of adding low-power functionality to a system without accepting the disadvantages of low-power. A basic example therefore is a system for streaming sensor data at high data rates. Such a system may be optimized but it will never compete with true low-power systems which profit from data rate reduction. High energy optimization can be compared to ordering a tailored suit. The tailored suit will fit perfectly but a standard suit will fit more people. This is what happens when engineers optimize systems for power consumption. The systems are tailored to fit a specific task. This tailoring includes selecting electrical components, software, and algorithms to fit the application requirements. The signal quality of the components should not be better than required for every part of the signal flow. The data rate should be as low as possible in every part of the signal flow. The reaction time until an event can be detected should be as high as justifiable from an application perspective, including legal concerns. These parameters need to be decreased to an acceptable minimum quality to the advantage of low-power consumption. However, no one can be held responsible for reducing the quality of the data and the response time if there are no substantial reasons for doing so. Therefore, engineers need tools to quantify the energy conservation at cost of the data quality. Accordingly, in chapter 7 models for the system and general data rate effects were proposed which were used to motivate decisions to the advantage of low-power. To give an example of motivation for considering the reduction of data quality, the reduction of the data rate by factor of 4 leads to a reduction of the energy consumption by a factor of 16 for FIR filters. Finally, the developer team must identify if the system which is going to be built from scratch or optimized for low-power is actually meant to be low-power by the requirements.

**Design Rule 4: Dual system approach: data collection vs. low-power design** The following issue is not limited to low-power devices. However, this issue has a much higher impact on low-power designs. The issue is that the algorithm operating in the sensor system must be developed first. Developing algorithms means that test data for algorithms exist. Test data for algorithms consists of input data for the algorithms and expected output data. Unfortunately, to gain this test data, it is necessary to acquire real-world data. There are several solutions. First of all, the signal flow chart should exist in two versions: One for a system which is the final low-power optimized product version without raw data collection, and the second diagram shows the system for raw data collection. Consequently, there should be two systems: The first for data collection and the second for the final low-power optimized system. The two systems differ in software and hardware. As an example from a former project example, while the first system had a low-power radio, the second had another radio module supporting higher data rates for raw data acquisition. Alternatively, the system for data collection may have more extensive storage for raw data. Once the test data are acquired, the algorithms can be optimized and tested according to low-power requirements. The final product does not include any raw data collection features. Instead, the final product should be optimized for energy conservation and the benefit of a smaller housing. Thus, planning project resources should consider a data collection system and a final low-power system.

**Design Note 5: The Power Manager** Within the scope of this work, the power manager became the software component with the higher influence on the realization of low-power applications. The term power manager exists with different definitions in the literature. However, in this thesis, the power manager was given defined (1) responsibility and (2) knowledge. The responsibility was defined such that, no matter what was requested by the application, the power manager would never allow an invalid or even dangerous state of the hardware and software to occur. For this purpose, the power manager was given a comprehensive set of rules, which included knowledge about the hardware and software modules. This knowledge includes logical and time constraints. Further, the power manager enabled the scalability of ULPSEK through the dynamic activation and deactivation of system components. This scalability was investigated in section 6.1. Therefore, the power manager has proven to be the key to effectively scaling performance to a tailored minimum for each application under investigation. For further investigation, the power manager was introduced in subsection 3.8.3 and is part of the control layer within the software architecture.

**Design Note 6: Power Distribution Network** The power distribution network is a hardware component. The software component that controlled the Power Distribution Network, was the Power Manager, mentioned above. The power manager holds detailed attributes about the power distribution network properties to ensure safe operation. In this thesis, an abstract specification of the power distribution network was utilized to optimize the interaction of the hardware and software. This abstract specification document was used extensively for power optimization during the whole development of hardware and software. It was the second most used specification document. For more details, the power distribution network was introduced in section 3.7.

**Design Note 7: Choosing and Using Low-Power Microcontrollers** Choosing a microcontroller for an application is a challenging engineering task. While this task is generally challenging, it is even more challenging for low-power applications. The reason is that choosing a low-power microcontroller alone will not significantly help conserve energy. On the contrary, it could be demonstrated that choosing a low-power microcontroller can lead to worse results. Energy conservation requires extensive usage of the low-power features of low-power microcontrollers. The key low-power features are low-power modes, low-power peripherals, and a substantially expanded clock unit. The extended clock unit of low-power microcontrollers, including extra-clock sources and better routing capabilities, is critical for low-power performance. For this reason, the clock unit of a low-power microcontroller is at least twice as large as with the mainstream series, regardless of the microcontroller brand. Today, every microcontroller brand has its low-power microcontroller series. However, the brands differ in relevant details for power optimization. The brand of microcontroller used finally for this thesis still has outstanding clock routing capabilities and key concepts. Therefore, it is mandatory to understand and use the clock unit features which come with low-power microcontrollers in detail. Consequently, the additional clock-unit features are the key to low power optimization of microcontroller-based systems. This topic was investigated in detail in section 5.2.

**Design Note 8: Using and configuring an RTOS property** Highly minimalistic applications can benefit from a design without an RTOS. However, assuming a biomedical sensor would be a minimalistic application is a mistake with consequences for a project's success. Usually, microcontrollers for biomedical sensing are packed with multiple features. With this, an RTOS is a must-have in almost every wearable application. However, as one result of section 5.3, an RTOS comes with drawbacks regarding power efficiency. The losses of power efficiency can reach the order of seven times more power consumption than without RTOS. Luckily, solutions exist to set up an RTOS to profit from its existential advantages without having the drawback of substantial power losses. Once an RTOS is configured for low-power applications, it takes control by putting the system in idle mode whenever possible. This configuration is an automatic power conservation feature that allows scaling the energy requirements depending on the CPU workload according to the newly introduced theoretical ideal system equation in section 8.11. A model for power consumption regulation using an RTOS was presented in subsection 7.1.2.

**Design Note 9: Supposed Optimization Strategies** Some strategies seem energy-saving, but they cause even more energy losses. For example, saving CPU workload using DMA is usually an excellent idea. However, DMA should be considered with care for low-power systems at low sampling rates. Many features like DMA do not allow setting the microcontroller to low-power mode while activated. Therefore, features like DMA prevent the microcontroller from conserving energy. This effect was investigated and confirmed in section 5.5. Using DMA was a misleading optimization strategy because it pays out for high data throughput above 1000 Sa/s, which is not the case for most biomedical sensor signals.

A second example is code optimization. Compilers come with compiler settings for code optimization. Optimization for smaller code sizes is the typical compiler setting to shrink the code to the available memory space. Optimizing for size is generally a good idea. However, a small code size does not imply fast code execution. Therefore, algorithms should be compiled with optimization for speed instead of size. Optimization for speed may produce a larger code size, but it also produces faster execution. Again, longer idle times allow a power-optimized RTOS based system to automatically spend more time in low-power mode. Consequently, compiler optimization settings for speed lead to energy conservation.

As a third example, taking the most efficient microcontroller in power per clock may seem obvious for a low-power application. Unfortunately, mostly the opposite is the case. A larger low-power microcontroller with faster computing support strategies is the better option. This effect was investigated in section 5.2.

**Design Note 10: Power Consumption System Testing** In a complex sensor system, minor changes in hardware and software can easily cause a substantial increase in the average power consumption. This effect was seen in experiments numerous times and is supported by the model. For a low-power system, low power consumption is a high-ranked requirement that must be met to pass a release system test. Therefore, the system test setup must include power measurement in all possible software configurations. Test automation can be gained by automatic checkout from the code repository and automatic upload to the latest test sensor hardware. This automation enables finding drastic increases in power consumption earlier in the development process.

# Chapter 10

# Summary and Outlook

This Ph.D. thesis aimed to develop a systematic approach for the system design of energy-efficient, battery-powered biomedical sensor systems. A particular focus of this work was the optimization of components along the entire signal processing chain, which implicitly included analog and digital technologies. This dissertation opened with an introductory chapter, then moved to an empirical core, and concluded by inducing the experimental results into a generalized theory of energy efficiency in information processing. The chapters reflect this common thread in this Ph.D. thesis, starting with the theoretical foundations in chapter 2, continuing with the construction of specialized utilities in chapters 3 and 4, the subsequent experiments in chapters 5 and 6, to a generic model in chapter 7, a theoretical treatise on energy-efficient signal theory in chapter 8, and finally the implicit rules in chapter 9. The core contents and particular findings of each chapter can be briefly recapitulated: The introductory chapter elaborates on particular features of wearable biomedical sensors compared to stationary devices and the fundamental challenges in biomedical signal processing. It summarizes the fundamental characteristics of signal processing system components, both analog and digital, in terms of their energy efficiency.

Following the introductory part, two tools were developed that served as a basis for later experiments: (1) ULPSEK – a novel sensor system with a development board specifically designed for the needs of energy optimization, and (2) a novel method for energy profile recording. This hardware's advantages were the modularity of the development board and the ability to monitor the energy consumption of individual submodules. The modularity allowed different hardware versions to be exchanged more quickly and compared against each other in terms of energy efficiency. Thanks to the 19 measuring points, essential energy supply branches could be monitored independently and regularly. These advantages made it possible to identify and eliminate energy waste much faster than possible with already miniaturized hardware. The advantages and necessity of this hardware over state of the art have been illustrated in chapters 3. Several design documents exist for the design of ULPSEK. In addition to the schematic, one design document had a significant positive impact on development and bug finding: the design document is named "Power Distribution Network" (PDN) and was presented in section 3.7. The PDN abstracts the information about the power supply from the schematic. This document was used continuously, especially for finding errors and had a decisive advantage over the circuit diagram due to the overview. In addition to the ULPSEK hardware, special ULPSEK software with numerous software components was designed from scratch. The development of the entire software was aimed at implementing the software components energy-efficiently. For example, all sensor driver, control, and signal processing modules were developed to be deactivatable and configurable. The software design documents in section 3.8 include three views that served day-to-day development to achieve the energy efficiency goals: (1) the architecture diagram with its layer model, (2) the class diagram, (3) the novel signal flow diagram, declared in Appendix A. An important finding was especially the essential role of the Power Manager. This software component provided a large set of rules about hardware and software dependencies. These rules included permissible state permutations and timing constraints. This rule set enabled reliable operation. At the same time, the power consumption could be automatically adjusted to the current demand. All hardware and software was designed and implemented specifically for this thesis. ULPSEK was a prerequisite for gaining new knowledge and proving it with data from experiments. As a second tool, a novel energy profile acquisition was developed. Compared with state-of-the-art, these methods allowed high-resolution voltage and current signal waveforms to be acquired and evaluated, mainly automatically. The automatic evaluation provided many parameters that helped compare different hardware and software variants in terms of energy efficiency. Multiple details were discovered with the help of the energy profiles and served as a basis for new findings. ULPSEK and the energy profiling subsequently served as the basis for all subsequent experiments.

The experiments in the chapters 5 and 6 aimed to obtain concrete measurements about the behavior of the system along the signal processing chain. Since this thesis focused on microcontroller-based systems, dedicated individual experiments were performed with different permutations of common microcontrollers. The results made clear that adjusting screws are crucial for energy-efficient signal processing. For example, a method that usually saves resources can lead to significant energy waste on low-power microcontrollers. In particular, decisive differences in energy efficiency became apparent in the choice of the supposedly energy-saving microcontroller or even its configuration in the experiment. The influences of software such as the operating system as well as the design of signal processing algorithms, especially in the choice of sampling rate, were demonstrated with measurement results from the experiment. Finally, numerous experiments were performed with ULPSEK in various hardware and software configurations. Early experiments required disconnecting the power supply to replace the hardware modules and reprogramming with alternate software. Later, a runtime configurable signal processing chain was developed to support runtime configuration. For this purpose, selected software and hardware parts were reworked to be parameterized at runtime. For example, filter characteristics of the hardware or alternative algorithms could thus be switched at runtime. As an advantage, a significant energy saving could be demonstrated since the system could now be set to the required minimum data quality at runtime. However, this extra functionality was offset by a much higher effort in development and testing compared to a system that could not be parameterized. Finally, the experiments were concluded with challenging proof that ULPSEK was successfully operated in an energy-autonomous manner using only a thermal harvester. In this experiment, it was impressively demonstrated by accident how a slight change in software could significantly negatively impact energy efficiency. For this reason, energy-efficient systems require a regular measurement routine for energy consumption during development with appropriate tools, as suggested in Chapter 4. From the mentioned experiments, depending on the hardware configuration and software parameterization, numerous energy profiles with new findings and measurement data have been collected. Next, a model was derived from these energy profiles. This model was presented in three successive stages in chapter 7. The three stages contain generic system parameters whose values were determined in particular for ULPSEK with the data from the experiments. Independently, the model is generic and thus applicable to other systems. With the help of this model, one of the central goals of this thesis was achieved, to predict the energy demand and energy efficiency of a sensor system depending on the system configuration.

Chapter 8 is a high-level view of energy efficiency in information processing. The methods in this chapter reduce any system or signal processing component to the data rate throughput and the energy required for processing. This theoretical consideration aimed to generalize further and abstract the previous findings. The abstraction applies to arbitrary system boundaries in the global system and its subcomponents. For this purpose, a measure of the energy efficiency of a system component was defined. This energy efficiency was then normalized to a physical minimum. Subsequently, the measure was transformed to a nonlinear scale. This way, the Landauer-Decibel was introduced as a new energy efficiency measure. The energy efficiency in Landauer-Decibels with the newly introduced auxiliary unit of measurement dBLa has two advantages: (1) orders of magnitude of energy efficiency are often far apart and can therefore be better compared on a logarithmic scale, and (2) the end of technology progress is at 0 dBLa, which means that it is always apparent how far energy efficiency is from the physical limit. Henceforth, all theoretical considerations were performed using the newly introduced Landauer decibel. Empirical data from the experiments were applied to illustrate the measures. Subsequently, the energy efficiency of the technology used was determined and compared. The technological progress of energy efficiency was also expressed and discussed using Landauer-Decibel per year. In addition, a theoretical condition for an energy-optimal system design was formally defined based on observation in the experiment. This condition applies to variable data rate systems and states that a parameterizable system maintains constant energy efficiency regardless of its data rate. The approximation of this condition could be empirically demonstrated using individual system parts of ULPSEK and theoretically proven in the model.

Following this, the chapter 9 concluded briefly with ten derived rules that have a particular effect on energy efficiency in system design and implementation and could be helpful in the applying engineer. Fortunately, the methods from this work have already found practical application in numerous projects in the professional environment. The insights, models, novel signal flow diagrams, and empirically obtained data contribute decisively to better decisions regarding energy efficiency at the design stage.

Besides energy efficiency, various secondary but related topics were briefly investigated in this work. Consequently, many ideas for further questions have emerged. Two selected topics are given for the outlook: (1) energy optimization with a digital twin and (2) energy optimization based on data quality. The first topic is based on the model from this work from Chapter 7. The idea for this topic was published with an invention disclosure. The invention disclosure describes a Digital Twin that continuously determines an energetically optimal configuration for a sensor system based on the model and measured parameters. The Digital Twin thus optimizes the energy efficiency of the sensor system or an entire sensor network. This invention disclosure also describes additional features for detecting deviations from the expected energy consumption via the model so that a security- or safety-relevant event or an approaching defect can be detected via the changed model parameters. The second topic of interest for the outlook addresses questions around data quality. In addition to energy efficiency, data quality has already been considered sporadically as a side matter in this work. However, data quality represents a broad cross-domain research field with many facets. The introductory chapter gave a comprehensive overview of data quality's versatility in subsection 2.2.4. Especially around data quality, many questions remain open and are not covered by state of the art. Therefore, deepening the topic of data quality in combination with energy modeling from this thesis would be another exciting continuation of this work.

# Zusammenfassung und Ausblick

Ziel dieser Doktorarbeit war die Erarbeitung einer systematischen Herangehensweise für den Systementwurf energieeffizienter, batteriebetriebener, biomedizinischer Sensorsysteme. Ein besonderes Augenmerk dieser Arbeit lag auf der Optimierung der Komponenten entlang der gesamten Signalverarbeitungskette, die implizit sowohl Analog- als auch Digitaltechnik umfasste. Diese Doktorarbeit wurde mit einem Grundlagenkapitel eröffnet, ging anschließend zu einem empirischen Kern über und schloss durch Induktion der experimentellen Ergebnisse zu einer generalisierten Theorie über die Energieeffizienz in der Informationsverarbeitung ab. Die Kapitel spiegeln diesen roten Faden in der Doktorarbeit wieder, angefangen bei den theoretischen Grundlagen in Kapitel 2, über die Konstruktion von spezialisierten Hilfsmitteln in Kapiteln 3 und 4, die anschließenden Experimente in den Kapiteln 5 und 6, bis hin zu einem generischen Modell in Kapitel 7, einer theoretischen Abhandlung zur energieeffizienten Signaltheorie in Kapitel 8 und schließlich den abgeleiteten Entwurfsregeln in Kapitel 9. Die Kerninhalte und besonderen Erkenntnisse der einzelnen Kapitel können wir folgt in Kürze wiedergegeben werden. Das Grundlagenkapitel arbeitet spezielle Merkmale tragbarer biomedizinischer Sensoren gegenüber stationären Geräten heraus, sowie die grundlegenden Herausforderungen bei der biomedizinischen Signalverarbeitung, und fasst die fundamentalen Eigenschaften der signalverarbeitenden Systemkomponenten, sowohl in der Analog- als auch Digitaltechnik, hinsichtlich derer Energieeffizienz, zusammen.

Im Anschluss an den Grundlagenteil wurden zwei Hilfsmittel entwickelt, die als Grundlage für spätere Experimente dienten: (1) ULPSEK – ein neuartiges Sensorsystem mit einer speziell für die Belange der Energieoptimierung entwickelten Platine und (2) ein neuartiges Verfahren zur Energieprofilerfassung. Die Modularität der Entwicklungsplatine und die Möglichkeit die Energieaufnahme der Einzelbaugruppen zu überwachen, waren die herausstechenden Vorteile dieser Hardware. Durch die Modularität konnten schneller verschiedene Versionen der Hardware ausgetauscht und gegeneinander hinsichtlich Energieeffizienz verglichen werden. Dank den 19 Messpunkten konnten wichtige Zweige der Energieversorgung unabhängig und regelmäßig überwacht werden. Damit war es möglich, Energieverschwendung deutlich schneller zu identifizieren und zu eliminieren, als dass es mit einer bereits miniaturisierten Hardware möglich wäre. Die Vorteile und die Notwendigkeit dieser Hardware gegenüber dem Stand der Technik wurden in Kapiteln 3 verdeutlicht. Für den Entwurf von ULPSEK existieren mehrere Entwurfsdokumente. Neben dem Schaltplan hatte ein Entwurfsdokument einen maßgeblich positiven Einfluss auf die Entwicklung und das Auffinden von Fehlern: das Entwurfsdokument trägt den Namen „Power Distribution Network“ (PDN) und wurde in Abschnitt 3.7 vorgestellt. Das PDN abstrahiert die Information über die Energieversorgung aus dem Schaltplan auf einer übersichtlichen Seite. Es stelle sich heraus, dass dieses Dokument fortlaufend insbesondere zum Auffinden von Fehlern genutzt wurde und durch die Übersicht gegenüber dem Schaltplan einen entscheidenden Vorteil hatte. Neben der ULPSEK-Hardware wurde eine spezielle ULPSEK-Software mit zahlreichen Softwarekomponenten von Grund auf neu entworfen und neu entwickelt. Die Neuentwicklung der gesamten Software hatte zum Ziel, die Softwarekomponenten energieeffizient umzusetzen. So wurden beispielsweise alle Sensor-Treiber-, Steuerungs- und Signalverarbeitungs-Module abschaltbar und konfigurierbar entwickelt. Die Software-Entwurfsdokumente in Abschnitt 3.8 umfassen drei Ansichten, die tagtäglich bei der Entwicklung zur Erreichung der Energieeffizienzziele dienten: (1) das Architekturdiagramm mit seinem Schichtenmodell, (2) das Klassediagramm, (3) das neuartige Signalflussdiagramm. Eine wichtige Erkenntnis war insbesondere die Bedeutung der umfangreichen Softwarekomponente des Powermanagers. Der Powermanager wurde mit zahlreichen Informationen in Form von Regeln versorgt. Diese Regeln umfassten zulässige Zustandspermutationen und zeitliche Einschränkungen. Erst dadurch wurde ein zuverlässiger Betrieb ermöglicht und zugleich konnte die Energieaufnahme automatisiert an den tatsächlichen aktuellen Bedarf angepasst werden. Die gesamte Hardware und Software wurde speziell zum Zwecke dieser Doktorarbeit entworfen und umgesetzt. ULPSEK war eine Voraussetzung, um neue Erkenntnisse zu gewinnen und mit Zahlen aus dem Experiment zu belegen. Als zweites Hilfsmittel wurde eine neuartige Energieprofilerfassung entwickelt. Gegenüber dem Stand der Technik konnten mit diesen Verfahren hochaufgelöste Spannungs- und Stromsignalverläufe erfasst und vorwiegend automatisiert ausgewertet werden. Die automatische Auswertung lieferte eine Vielzahl von Kenngrößen, die dabei halfen verschiedene Hardware- und Software-Varianten hinsichtlich ihrer Energieeffizienz zu vergleichen. Zahlreiche Details wurden mit Hilfe der Energieprofile entdeckt und dienten als Grundlage für neue Erkenntnisse. ULPSEK und die Energieprofilerfassung dienten anschließend als Grundlage für alle nachfolgenden Experimente.

Die Experimente in den Kapiteln 5 und 6 hatten zum Ziel konkrete Messwerte über das Verhalten des Systems entlang der Signalverarbeitungskette zu gewinnen. Da der Schwerpunkt dieser Doktorarbeit auf mikrocontrollerbasierten Systemen lag, sind zahlreiche Einzeluntersuchungen mit verschiedenen Permutationen gängiger Mikrocontroller durchgeführt worden. Die Ergebnisse machten deutlich welche Entwurfsentscheidungen maßgeblich für eine energieeffiziente Signalverarbeitung sind. So kann eine Methode, die üblicherweise Ressourcen spart, zu einer maßgeblichen Energieverschwendung auf energiesparenden Mikrocontrollern führen. Insbesondere wurden entscheidende Unterschiede in der Energieeffizienz in der Wahl des vermeintlich energiesparenden Mikrocontrollers oder auch dessen Konfiguration im Experiment deutlich. Auch die Einflüsse der Software wie das Betriebssystem sowie die Auslegung der Signalverarbeitungsalgorithmen, insbesondere bei der Wahl der Abtastrate sind mit Zahlen aus dem Experiment belegt worden. Schließlich wurden zahlreiche Experimente mit ULPSEK in verschiedenen Konfigurationen von Hardware und Software durchgeführt. Jedes Experiment erforderte zunächst ein Trennen der Stromversorgung für den Austausch der Hardware-Module und ein Neuprogrammieren mit einer alternativen Software. Als Steigerung des Funktionsumfangs wurde eine zur Laufzeit konfigurierbare Signalverarbeitungskette entwickelt. Hierzu wurden Teile der Software und Hardware überarbeitet, so dass sie auch zur Laufzeit parametrisierbar wurden. Filtercharakteristiken der Hardware oder alternative Algorithmen konnten so zur Laufzeit umgeschaltet werden. Als Vorteil konnte eine deutliche Energieersparnis nachgewiesen werden, da das System nun auf das erforderliche Minimum der Datenqualität zur Laufzeit eingestellt werden konnte. Dem gegenüber stand allerdings ein vielfach höherer Aufwand in der Entwicklung und im Test im Vergleich zu einem nicht parametrisierbaren System. Schließlich wurden die Experimente mit einem Härtetest abgeschlossen, in dem ULPSEK mithilfe eines thermischen Harvesters erfolgreich energieautark betrieben wurde. Hierbei konnte immer wieder eindrucksvoll gezeigt werden, wie eine kleine Änderung in der Hardware oder Software einen erheblich negativen Einfluss auf die Energieeffizienz haben kann. Deshalb gehört bei energieeffizienten Systemen die regelmäßige Messung der Energieaufnahme schon während der Entwicklung dazu. Dies musste auch so in der Praxis gehandhabt werden, um sogenannte Energy-Bugs frühzeitig zu erkennen, da eine nachträgliche Suche deutlich auswendiger war.

Aus den erwähnten Experimenten sind in Abhängigkeit der Hardware-Konfiguration und Software-Parametrisierung zahlreiche Energieprofile mit neuen Erkenntnissen und Messdaten erfasst worden. Aus diesen Energieprofilen ist ein Modell abgeleitet worden. Dieses Modell wurde in drei aufeinander aufbauenden Stufen in Kapitel 7 vorgestellt. Die drei Ausbaustufen enthalten generische Systemparameter deren Werte im Speziellen für ULPSEK mit den Daten aus den Experimenten ermittelt wurden. Unabhängig davon ist das Modell generisch und somit auch auf andere Systeme anwendbar. Mithilfe dieses Modells wurde eines der zentralen Ziele dieser Arbeit erreicht, die Vorhersage des Energiebedarfs und der Energieeffizienz eines Sensorsystems in Abhängigkeit der Systemkonfiguration.

Darauf folgt Kapitel 8 über die Energieeffizienz in der Informationsverarbeitung. In diesem Kapitel wird ein beliebiges System oder eine beliebiges Signalverarbeitungskomponente auf den Datenratendurchsatz und die dafür erforderliche Energie reduziert. Ziel dieser theoretischen Betrachtung war eine weiterführende Generalisierung und Abstraktion der bisherigen Erkenntnisse. Die Abstraktion gilt für beliebige Systemgrenzen, sowohl im großen Ganzen als auch für Teilkomponenten. Hierfür wurde eine Maßzahl für die Energieeffizienz einer Systemkomponente festgelegt. Daraufhin wurde diese Energieeffizienz auf ein physikalisches Minimum normiert. Anschließend wurde die Maßzahl auf eine nichtlineare Skala transformiert. Auf diese Weise wurde das Landauer-Dezibel als neue Kennzahl für Energieeffizienz eingeführt. Die Energieeffizienz in Landauer-Dezibel mit der neu eingeführten Hilfsmaßeinheit dBLa hat zwei Vorteile: (1) Größenordnungen der Energieeffizienz liegen oft weit auseinander und lassen sich deshalb auf er logarithmischen Skala besser vergleichen, und (2) das Ende des Technologiefortschritts liegt bei 0 dBLa, wodurch stets ersichtlich ist, wie weit sich die Energieeffizienz vom physikalischen Limit befindet. Fortan wurden alle theoretischen Betrachtungen mit dem neu eingeführten Landauer-Dezibel durchgeführt. Zur Veranschaulichung der Maßzahlen wurden empirische Daten aus den Experimenten angewandt. Anschließend wurde die Energieeffizienz der verwendeten Technologie ermittelt und verglichen. Auch der technologische Fortschritt der Energieeffizienz wurde mit Landauer-Dezibel pro Jahr ausgedrückt und diskutiert. Darüber hinaus wurde auf Basis einer Beobachtung im Experiment eine theoretische Bedingung für einen energie-optimalen Systementwurf formal definiert. Diese Bedingung gilt für Systeme mit variabler Datenrate und besagt, dass ein parametrisierbares System unabhängig von seiner Datenrate eine konstante Energieeffizienz behält. Die Annäherung an diese Bedingung konnte anhand einzelner Systemteile von ULPSEK empirisch nachgewiesen und auch im Modell theoretisch belegt werden.

Darauf folgend wurden im Kapitel 9 abschließend in Kürze zehn abgeleitete Regeln aufgestellt, die einen besonderen Effekt auf die Energieeffizienz beim Systementwurf und der Umsetzung haben und für den anwendenden Ingenieur nützlich sein könnten. Erfreulicherweise fanden die Methoden aus dieser Arbeit bereits praktische Anwendung in zahlreichen Projekten im professionellen Umfeld. Die hier erarbeiteten Erkenntnisse, Modelle, neuartigen Signalflusspläne, und auch die empirisch gewonnen Daten trugen entscheidend dazu bei, bessere Entscheidungen hinsichtlich der Energieeffizienz schon im Entwurfsstadium zu treffen.

Abgesehen von der Energieeffizienz sind in dieser Arbeit zahlreiche Nebenthemen untersucht worden. Dabei sind Ideen für weiterführende Fragestellungen entstanden. Nachfolgen werden zwei dieser weiterführenden Themen für den Ausblick vorgestellt: (1) Digitaler Zwilling und (2) Datenqualität. Das erste Thema basiert auf dem Modell aus dieser Arbeit. Die Idee dazu wurde in einer Erfindungsmeldung veröffentlicht. Die Erfindungsmeldung beschreibt einen Digitalen Zwilling, der fortlaufend eine energetisch optimale Konfiguration für ein Sensorsystem auf Basis des Modells und gemessener Parameter ermittelt. Der Digitale Zwilling optimiert damit die Energieeffizienz des Sensorsystems oder, größer gedacht, eines ganzen Sensornetzes. Diese Erfindungsmeldung beinhaltet darüber hinaus eine zusätzliche praktische Erweiterung, die dazu dient, eine Abweichung der erwarteten Energieaufnahme über das Modell zu erkennen, so dass ein sicherheitsrelevantes Ereignis oder ein sich anbahnender Defekt über die geänderten Modellparameter erkannt werden kann. Als zweites für die Weiterführung interessantes Thema, ergaben sich weitere Fragestellungen rund um die Qualität der Daten. Neben der Energieeffizienz wurde bereits in dieser Arbeit die Datenqualität vereinzelt als Nebenthema betrachtet. Allerdings stellt Datenqualität für sich schon ein weites domänenübergreifendes Forschungsfeld mit vielen Facetten dar. Hierfür wurde ein umfassender Überblick über die Vielseitigkeit der Datenqualität im Grundlagenkapitel gegeben. Gerade rund um das Thema der Datenqualität sind viele Fragen weiterhin offen und nicht über den Stand der Technik abgedeckt. Deshalb wäre eine Vertiefung in das Thema der Datenqualität in Kombination mit der Energiemodellierung aus dieser Arbeit eine weitere interessante Fortführung dieser Arbeit.

# References

[1] “IEEE Standard for Wearable Consumer Electronic Devices--Overview and Architecture,” *IEEE Std 360-2022*, pp. 1–35, 2022. DOI: 10.1109/IEEESTD.2022.9762855.

[2] Jagarlamudi, K. S., Zaslavsky, A., Loke, S. W., Hassani, A., Medvedev, A., “Requirements, Limitations and Recommendations for Enabling End-to-End Quality of Context-Awareness in IoT Middleware,” *Sensors*, vol. 22 (4), 2022. DOI: 10.3390/S22041632.

[3] Nissen, M., Slim, S., Jaeger, K., Flaucher, M., Huebner, H., Danzberger, N., Fasching, P., Beckmann, M., Gradl, S., Eskofier, B., “Heart Rate Measurement Accuracy of Fitbit Charge 4 and Samsung Galaxy Watch Active2: Device Evaluation Study,” *JMIR Formative Research*, vol. 6, e33635, Mar. 2022. DOI: 10.2196/33635.

[4] Ketshabetswe, K. L., Zungeru, A. M., Mtengi, B., Lebekwe, C. K., Prabaharan, S. R. S., “Data Compression Algorithms for Wireless Sensor Networks: A Review and Comparison,” *IEEE Access*, vol. 9, pp. 136 872–136 891, 2021. DOI: 10.1109/ACCESS.2021.3116311.

[5] Masihi, S., Panahi, M., Maddipatla, D., Hanson, A. J., Fenech, S., Bonek, L., Sapoznik, N., Fleming, P. D., Bazuin, B. J., Atashbar, M. Z., “Development of a Flexible Wireless ECG Monitoring Device with Dry Fabric Electrodes for Wearable Applications,” *IEEE Sensors Journal*, pp. 1–1, 2021. DOI: 10.1109/JSEN.2021.3116215.

[6] Mekruksavanich, S., Jitpattanakul, A., “Sensor-based Complex Human Activity Recognition from Smartwatch Data using Hybrid Deep Learning Network,” in *2021 36th International Technical Conference on Circuits/Systems, Computers and Communications*, 2021, pp. 1–4. DOI: 10.1109/ITC-CSCC52171.2021.9501477.

[7] Palumbo, A., Vizza, P., Calabrese, B., Ielpo, N., “Biopotential Signal Monitoring Systems in Rehabilitation: A Review,” *Sensors*, vol. 21 (21), 2021. DOI: 10.3390/S21217172.

[8] Perez, A. J., Zeadally, S., “Recent Advances in Wearable Sensing Technologies,” *Sensors*, vol. 21 (20), 2021. DOI: 10.3390/S21206828.

[9] Ruiz, L. J. L., Ridder, M., Fan, D., Gong, J., Li, B. M., Mills, A. C., Cobarrubias, E., Strohmaier, J., Jur, J. S., Lach, J., “Self-Powered Cardiac Monitoring: Maintaining Vigilance With Multi-Modal Harvesting and E-Textiles,” *IEEE Sensors Journal*, vol. 21 (2), pp. 2263–2276, 2021. DOI: 10.1109/JSEN.2020.3017706.

[10] Scholl, C., **Tobola, A.,** Ludwig, K., Zanca, D., Eskofier, B. M., “A Smart Capacitive Sensor Skin with Embedded Data Quality Indication for Enhanced Safety in Human–Robot Interaction,” *Sensors*, vol. 21 (21), 2021. DOI: 10.3390/S21217210.

[11] Zanaj, E., Caso, G., De Nardis, L., Mohammadpour, A., Alay, Ö., Di Benedetto, M.-G., “Energy Efficiency in Short and Wide-Area IoT Technologies—A Survey,” *Technologies*, vol. 9 (1), 2021. DOI: 10.3390/TECHNOLOGIES9010022.

[12] "AD8232: Single-lead, heart rate monitor front end," version D, Analog Devices, 2020.

[13] Dhaimodker, V., Desai, R., Mini, S., Tosh, D. K., "Quality-driven Energy Optimization in Internet of Things," in *2020 29th International Conference on Computer Communications and Networks*, 2020, pp. 1–8. DOI: 10.1109/ICCCN49398.2020.9209666.

[14] Köckemann, U., Alirezaie, M., Renoux, J., Tsiftes, N., Ahmed, M. U., Morberg, D., Lindén, M., Loutfi, A., "Open-Source Data Collection and Data Sets for Activity Recognition in Smart Homes," *Sensors*, vol. 20 (3), 2020. DOI: 10.3390/S20030879.

[15] Teh, H. Y., Kempa-Liehr, A. W., Wang, K. I.-K., "Sensor data quality: a systematic review," *Journal of Big Data*, vol. 7 (1), p. 11, 2020.

[16] Akbas, A., "Comparative Analysis of Lightweight Cryptography Algorithms on Resource Constrained Microcontrollers," in *2019 1st International Informatics and Software Engineering Conference*, 2019, pp. 1–4.

[17] Anuar, H., Leow, P. L., "Non-invasive Core Body Temperature Sensor for Continuous Monitoring," in *2019 IEEE International Conference on Sensors and Nanotechnology*, 2019, pp. 1–4.

[18] C, B., "Comparison of Accuracy and Diagnostic Validity of a Novel Non-Invasive Electrocardiographic Monitoring Device with a Standard 3 Lead HolterMonitor and an ECG Patch over a 24 hours Period," *Journal of Cardiovascular Diseases and Diagnosis*, 2019.

[19] Carratù, M., Iacono, S. D., Long Hoang, M., Pietrosanto, A., "Energy characterization of attitude algorithms," in *2019 IEEE 17th International Conference on Industrial Informatics*, vol. 1, 2019, pp. 1585–1590.

[20] "CC2541 SimpleLink™ Bluetooth® low energy and proprietary wireless MCU," Datasheet, version 2.4, Panasonic, 2019.

[21] "Cosinus One, Users Manual," version 1.1, 2019.

[22] Elfaramawy, T., Fall, C. L., Arab, S., Morissette, M., Lellouche, F., Gosselin, B., "A Wireless Respiratory Monitoring System Using a Wearable Patch Sensor Network," *IEEE Sensors Journal*, vol. 19 (2), pp. 650–657, 2019.

[23] Fouradoulas, M., Känel, R., von, Schmid, J.-P., "Heart Rate Variability - State of Research and Clinical Applicability," *Praxis*, vol. 108, pp. 461–468, 7 2019.

[24] Gallizio, E., "A Breakthrough Innovation in MEMS Sensors, Introducing LSM6DSOX, iNEMO 6DoF InertialMeasurement Unit (IMU) with Machine Learning Core," STmicroelectronics, 2019.

[25] González-Ortega, D., Díaz-Pernas, F. J., Martínez-Zarzuela, M., Antón-Rodríguez, M., "A Physiological Sensor-Based Android Application Synchronized with a Driving Simulator for Driver Monitoring," *Sensors*, vol. 19 (2), 2019. DOI: 10.3390/S19020399.

[26] "Heart Failure and Cardiovascular Diseases -- A European HeartNetwork Paper," 2019.

[27] "Hexoskin Product Specifications," 2019.

[28] "LSM6DSOX: Machine Learning Core, Application note AN5259," version 1, STMicroelectronics, 2019.

[29] Sieh, V., Burlacu, R., Hönig, T., Janker, H., Raffeck, P., Wägemann, P., Schröder-Preikschat, W., "Combining Automated Measurement-Based Cost Modeling With Static Worst-Case Execution-Time and Energy-Consumption Analyses," *IEEE Embedded Systems Letters*, vol. 11 (2), pp. 38–41, 2019.

[30] Song, S., Konijnenburg, M., van Wegberg, R., Xu, J., *et al.*, "A 769  W Battery-Powered Single-Chip SoC With BLE for Multi-Modal Vital Sign Monitoring Health Patches," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 13 (6), pp. 1506–1517, 2019.

[31] Al Disi, M., Djelouat, H., Kotroni, C., Politis, E., Amira, A., Bensaali, F., Dimitrakopoulos, G., Alinier, G., "ECG Signal Reconstruction on the IoT-Gateway and Efficacy of Compressive Sensing Under Real-Time Constraints," *IEEE Access*, vol. 6, pp. 69 130–69 140, 2018. DOI: `10.1109/ACCESS.2018.2877679`.

[32] "AN0021: Analog to Digital Converter (ADC)," Application Note, version 1.11, Silicon Labs, 2018.

[33] Atallah, L., Ciuhu, C., Wang, C., Bongers, E., Blom, T., Paulussen, I., Noordergraaf, G., "An ergonomic wearable core body temperature sensor," in *2018 IEEE 15th International Conference on Wearable and Implantable Body Sensor Networks*, 2018, pp. 70–73.

[34] Chakravorty, P., "What Is a Signal? [Lecture Notes]," *IEEE Signal Processing Magazine*, vol. 35 (5), pp. 175–177, 2018.

[35] Liu, F., Liu, C., Jiang, X., Zhang, Z., Zhang, Y., Li, J., Wei, S., "Performance analysis of ten common QRS detectors on different ECG application cases," *Journal of healthcare engineering*, Hindawi, vol. 2018, 2018.

[36] "LSM6DSL: always-on 3D accelerometer and 3D gyroscope, Application note AN5040," version 3, STMicroelectronics, 2018.

[37] Morshedlou, F., Ravanshad, N., Rezaee-Dehsorkh, H., "A Low-Power Current-Mode Analog QRS-Detection Circuit for Wearable ECG Sensors," in *2018 25th National and 3rd International Iranian Conference on Biomedical Engineering*, 2018, pp. 1–6.

[38] Nanchen, D., "Resting heart rate: what is normal?" 2018.

[39] "Samsung Galaxy Watch," User manual, 2018.

[40] **Tobola, A.,** Leutheuser, H., Pollak, M., Spies, P., Hofmann, C., Weigand, C., Eskofier, B. M., Fischer, G., "Self-Powered Multiparameter Health Sensor," *IEEE Journal of Biomedical and Health Informatics*, vol. 22 (1), pp. 15–22, Jan. 2018. DOI: `10.1109/JBHI.2017.2708041`.

[41] Wägemann, P., Dietrich, C., Distler, T., Ulbrich, P., Schröder-Preikschat, W., "Whole-System Worst-Case Energy-Consumption Analysis for Energy-Constrained Real-Time Systems," in *30th Euromicro Conference on Real-Time Systems*, Altmeyer, S., (Ed.), ser. Leibniz International Proceedings in Informatics (LIPIcs), vol. 106, Dagstuhl, Germany: Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2018, 24:1–24:25. DOI: `10.4230/LIPICS.ECRTS.2018.24`.

[42] Walzer, T., Thies, C., Meier, K., Madrid, N. M., "Textile Sensor Platform (TSP)-development of a textile real-time electrocardiogram," in *International Conference on Bioinformatics and Biomedical Engineering*, Springer, 2018, pp. 359–370.

[43] Zellner, D., "Unternehmenszusammenschluss im Bereich Sportmanagement am Beispiel von Ambiotex durch Adidas." GRIN Verlag, 2018.

[44] Aliverti, A., "Wearable technology: role in respiratory health and disease," *Breathe*, Eur Respiratory Soc, vol. 13 (2), e27–e36, 2017.

[45] "Cardiovascular diseases (CVDs)," May 2017.

[46] Donnelly, T., Choi, J., Kildishev, A. V., Swabey, M., Johnson, M. C., “A comparative study for performance and power consumption of FPGA digital interpolation filters,” *International Journal of Advanced Computer Science and Applications (IJACSA)*, vol. 8, p. 477, 2017.

[47] Dräger, (Ed.), “Instructions for use Infinity Acute Care System,” version 4, Dräger, 2017.

[48] Hofmann, C., “CardioSHIRT -- Mobile textile-based multichannel ECG,” 2017.

[49] Leutheuser, H., Lang, N., Gradl, S., Struck, M., **Tobola, A.,** Hofmann, C., Anneken, Eskofier, B., “Book Chapter: Textile Wearable Technologies for Sports and Medical Applications,” in *Smart Textiles: Fundamentals, Design, and Interaction*. Springer, 2017.

[50] Leutheuser, H., Heyde, C., Roecker, K., Gollhofer, A., Eskofier, B. M., “Reference-free adjustment of respiratory inductance plethysmography for measurements during physical exercise,” *IEEE Transactions on Biomedical Engineering*, IEEE, vol. 64 (12), pp. 2836–2846, 2017.

[51] Mammela, A., Anttonen, A., “Why Will Computing Power Need Particular Attention in Future Wireless Devices?” *IEEE Circuits and Systems Magazine*, vol. 17 (1), pp. 12–26, 2017.

[52] “Qardio: Wireless ECG Monitor User Manual,” version 1.01, 2017.

[53] Qi, H., Ayorinde, O., Calhoun, B. H., “An ultra-low-power FPGA for IoT applications,” Burlingame, CA: IEEE, 2017, pp. 1–3. DOI: 10.1109/S3S.2017.8308753.

[54] “User’s Manual - ambiotex,” 2017.

[55] Vera, A., Love, J., Piovesan, J., Burke, B., Mee, J., “SHARP: A Space HARdened Procesor for Next Generation CubeSats,” 2017.

[56] Wilkins, E., Wilson, L., Wickramasinghe, K., Bhatnagar, P., Leal, J., Luengo-Fernandez, R., Burns, R., Rayner, M., Townsend, N., “European cardiovascular disease statistics 2017,” European Heart Network, 2017.

[57] Ciabattoni, L., Freddi, A., Longhi, S., Monteriù, A., Pepa, L., Prist, M., “An open and modular hardware node for wireless sensor and body area networks,” *Journal of Sensors*, Hindawi, vol. 2016, 2016.

[58] Dionisi, A., Marioli, D., Sardini, E., Serpelloni, M., “Autonomous Wearable System for Vital Signs Measurement With Energy-Harvesting Module,” *IEEE Transactions on Instrumentation and Measurement*, IEEE, vol. 65 (6), pp. 1423–1434, 2016.

[59] Hong, J., Lambson, B., Dhuey, S., Bokor, J., “Experimental test of Landauer’s principle in single-bit operations on nanomagnetic memory bits,” *Science advances*, American Association for the Advancement of Science, vol. 2 (3), e1501492, 2016.

[60] “Instruction Manual: MS410 ECG Simulator,” version 1.6, MedTec-Science GmbH, 2016.

[61] Jensen, U., Kugler, P., Ring, M., Eskofier, B. M., “Approaching the accuracy--cost conflict in embedded classification system design,” *Pattern Analysis and Applications*, Springer, vol. 19 (3), pp. 839–855, 2016.

[62] Kindt, P., Yunge, D., **Tobola, A.,** Fischer, G., Chakraborty, S., “Dynamic service switching for the medical IoT,” *IEEE 27th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, pp. 1–7, 2016.

[63] Kogan, D., Jain, A., Kimbro, S., Gutierrez, G., Jain, V., "Respiratory Inductance Plethysmography Improved Diagnostic Sensitivity and Specificity of Obstructive Sleep Apnea," *Respiratory Care*, Respiratory Care, vol. 61 (8), pp. 1033–1037, 2016. DOI: 10.4187/RESPCARE.04436.

[64] Lorenser, T., "The DSP capabilities of ARM Cortex -M4 and Cortex-M7 Processors, DSP feature set and benchmarks," White Paper, 2016.

[65] "MMA8452Q, 3-axis, 12-bit/8-bitdigital accelerometer," Datasheet, version 10, NXP, 2016.

[66] Pandesswaran, C., Surender, S., Karthik, K. V., "Remote patient monitoring system based coap in wireless sensor networks," *International Journal of Sensor Networks and Data Communications*, vol. 5 (3), 2016.

[67] "Polar H7, User Manual," 2016.

[68] Selvarathinam, J., Anpalagan, A., "Energy Harvesting From the Human Body for Biomedical Applications," *IEEE Potentials*, vol. 35 (6), pp. 6–12, 2016.

[69] **Tobola, A.,** Leutheuser, H., Schmitz, B., Hofmann, C., Struck, M., Weigand, C., Eskofier, B. M., Fischer, G., "Battery Runtime Optimization Toolbox for Wearable Biomedical Sensors," *Proc. IEEE 13th International Conference on Wearable and Implantable Body Sensor Networks (BSN)*, pp. 199–204, Jun. 2016. DOI: 10.1109/BSN.2016.7516259.

[70] Dräger, (Ed.), "Infinity®M540 monitor," version 4, Dräger, 2015.

[71] "INA219: Zerø-Drift, Bidirectional Current/Power Monitor With I2C Interface," version G, Texas instruments, 2015.

[72] Leutheuser, H., Gradl, S., Eskofier, B., **Tobola, A.,** Lang, N. R., Anneken, L., Arnold, M., Achenbach, S., "Arrhythmia classification using RR intervals: Improvement with sinusoidal regression feature," *IEEE 12th International Conference on Wearable and Implantable Body Sensor Networks (BSN)*, pp. 1–5, 2015.

[73] Misra, V., Bozkurt, A., Calhoun, B., Jackson, T., Jur, J. S., Lach, J., Lee, B., Muth, J., Oralkan, Ö., Öztürk, M., *et al.*, "Flexible technologies for self-powered wearable health and environmental sensing," *Proceedings of the IEEE*, IEEE, vol. 103 (4), pp. 665–681, 2015.

[74] "Shimmer 3 ECG / EMG Unit," version 1.5, Shimmer, 2015.

[75] **Tobola, A.,** Espig, C., Streit, F. J., Korpok, O., *et al.*, "Scalable ECG Hardware and Algorithms for Extended Runtime of Wearable Sensors," *Proc. IEEE International Symposium on Medical Measurements and Applications (MeMeA)*, 2015.

[76] **Tobola, A.,** Streit, F. J., Korpok, O., Espig, C., *et al.*, "Sampling Rate Impact on Energy Consumption of Biomedical Signal Processing Systems," *Proc. IEEE 12th International Conference on Wearable and Implantable Body Sensor Networks (BSN)*, 2015.

[77] **Tobola, A.,** "Intelligente Sensorik : Schlussbericht," Tech. Rep., 2015. DOI: 10.2314/GBV:868717789.

[78] Van Helleputte, N., Konijnenburg, M., Pettine, J., Jee, D., *et al.*, "A 345 µW Multi-Sensor Biomedical SoC With Bio-Impedance, 3-Channel ECG, Motion Artifact Reduction, and Integrated DSP," *IEEE Journal of Solid-State Circuits*, vol. 50 (1), pp. 230–244, 2015.

[79] Wägemann, P., Distler, T., Hönig, T., Janker, H., Kapitza, R., Schröder-Preikschat, W., “Worst-Case Energy Consumption Analysis for Energy-Constrained Embedded Systems,” in *2015 27th Euromicro Conference on Real-Time Systems*, 2015, pp. 105–114.

[80] “AFE4490 Integrated Analog Front-End for Pulse Oximeters,” Datasheet, Texas Instruments, 2014.

[81] Balint, R., Cassidy, N. J., Cartmell, S. H., “Conductive polymers: Towards a smart biomaterial for tissue engineering,” *Acta Biomaterialia*, vol. 10 (6), pp. 2341–2353, 2014. DOI: `HTTPS://DOI.ORG/10.1016/J.ACTBIO.2014.02.015`.

[82] Brogan, Q., O’Connor, T., Ha, D. S., “Solar and thermal energy harvesting with a wearable jacket,” in *Proc. IEEE ISCAS*, 2014, pp. 1412–1415.

[83] Caples, A., “How Effective Power management Increases the Lifetime Operation of Portable Medical Devices,” Tech. Rep., Mentor Graphics, 2014.

[84] Chen, G., Rodriguez-Villegas, E., Casson, A. J., “Wearable algorithms: An overview of a truly multi-disciplinary problem,” in *Wearable sensors*, Elsevier, 2014, pp. 353–382.

[85] Craciun, A. V., “Power modeling of complex circuits based on measurements; MD8710 chip analysis,” in *2014 International Conference on Optimization of Electrical and Electronic Equipment*, IEEE, 2014, pp. 833–838.

[86] Danieli, A., Lusa, L., Potočnik, N., Meglič, B., Grad, A., Bajrović, F. F., “Resting heart rate variability and heart rate recovery after submaximal exercise,” *Clinical Autonomic Research*, Springer, vol. 24 (2), pp. 53–61, 2014.

[87] Deepu, C., Lian, Y., “A Joint QRS Detection and Data Compression Scheme for Wearable sensors,” IEEE, 2014.

[88] Elgendi, M., Eskofier, B., Dokos, S., Abbott, D., “Revisiting QRS detection methodologies for portable, wearable, battery-operated, and wireless ECG systems,” *PLOS ONE*, Public Library of Science, vol. 9 (1), e84018, 2014.

[89] Gandhi, M., Wang, T., “The future of biosensing wearables,” *San Francisco: Rock Health*, 2014.

[90] Ha, S., Kim, C., Chi, Y. M., Cauwenberghs, G., “Low-Power Integrated Circuit Design for Wearable Biopotential Sensing,” in *Wearable Sensors*, Elsevier, 2014, pp. 323–352.

[91] Hönig, T., Janker, H., Eibel, C., Mihelic, O., Kapitza, R., “Proactive energy-aware programming with PEEK,” in *2014 Conference on Timely Results in Operating Systems*, 2014.

[92] Kindt, P., Yunge, D., Diemer, R., Chakraborty, S., “Precise energy modeling for the bluetooth low energy protocol, Wireless Radio Bluetooth Network Networks IoT,” 2014.

[93] Kuka, C., Nicklas, D., “Enriching sensor data processing with quality semantics,” in *2014 IEEE International Conference on Pervasive Computing and Communication Workshops*, 2014, pp. 437–442. DOI: `10.1109/PERCOMW.2014.6815246`.

[94] Makikawa, M., Shiozawa, N., Okada, S., “Fundamentals of wearable sensors for the monitoring of physical and physiological changes in daily life,” in *Wearable Sensors*. Elsevier, 2014, pp. 517–541.

[95] Mateu, L., Dräger, T., Mayordomo, I., Pollak, M., “Wearable Sensors: Fundamentals, Implementation and Applications,” in *Wearable Sensors: Fundamentals, Implemen-*

*tation and Applications*, Sazonov, N. M. R., E. (Ed.). Elsevier, 2014, ch. Energy Harvesting at the Human Body.

[96] Park, S., Jayaraman, S., "A transdisciplinary approach to wearables, big data and quality of life, Wearable Wearables," *In Proc: 36th Annual International Conference of the IEEE EMBC*, pp. 4155–4158, 2014.

[97] Prenesti, E., Gosmaro, F., "Trueness, precision and accuracy: a critical overview of the concepts as well as proposals for revision," *Accreditation and Quality Assurance*, vol. 20, pp. 33–40, 2014.

[98] Santos, D. F., Bublitz, F. M., Almeida, H. O., Perkusich, A., "Integrating IEEE 11073 and constrained application protocol for personal health devices," in *Proceedings of the 29th Annual ACM Symposium on Applied Computing*, 2014, pp. 466–467.

[99] Schmitz, B., "Entwicklung einer textilintegrierten Sensorik zur kontinuierlichen Atemaktivitätserfassung," Master's thesis, Fraunhofer IIS, 2014.

[100] **Tobola, A.,** Korpok, O., Leutheuser, H., Schmitz, B., Hofmann, C., Struck, M., Weigand, C., Eskofier, B., Heuberger, A., Fischer, G., "System Design Impacts on Battery Runtime of Wearable Medical Sensors," *Proc. 2nd International Conference on Mobile and Information Technologies in Medicine (MobileMed)*, 2014.

[101] Voss, T. J., Subbian, V., Beyette, F. R., "Feasibility of energy harvesting techniques for wearable medical devices," in *Proc. IEEE EMBC*, 2014, pp. 626–629.

[102] Wu, Q., Ding, G., Xu, Y., Feng, S., Du, Z., Wang, J., Long, K., "Cognitive Internet of Things: A New Paradigm Beyond Connection," *IEEE Internet of Things Journal*, vol. 1 (2), pp. 129–143, 2014. DOI: 10.1109/JIOT.2014.2311513.

[103] Yang, G.-Z., "Body sensor networks, Wearable," 2nd edition. Springer-Verlag London, 2014.

[104] Yeatman, E., Mitcheson, P., "Energy Harvesting and Power Delivery, Body Sensor Networks," in *Body Sensor Networks*, Yang, G.-Z., (Ed.), Springer-Verlag London, 2014.

[105] Almazaydeh, L., Elleithy, K., Faezipour, M., Abushakra, A., "Apnea Detection based on Respiratory Signal Classification," *Procedia Computer Science*, vol. 21, pp. 310–316, 2013. DOI: HTTPS://DOI.ORG/10.1016/J.PROCS.2013.09.041, The 4th International Conference on Emerging Ubiquitous Systems and Pervasive Networks (EUSPN-2013) and the 3rd International Conference on Current and Future Trends of Information and Communication Technologies in Healthcare (ICTH).

[106] Bolz, A., Urbaszek, W., "Technik in der Kardiologie: eine interdisziplinäre Darstellung für Ingenieure und Mediziner, ECG." Springer-Verlag, 2013.

[107] Botta, M., Simek, M., Mitton, N., "Comparison of hardware and software based encryption for secure communication in wireless sensor networks," in *2013 36th International Conference on Telecommunications and Signal Processing*, 2013, pp. 6–10.

[108] Chatschik Bisdikian, M. B. S., Lance M. Kaplan, "On the Quality and Value of Information in Sensor Networks," *ACM Transactions on Sensor Networks*, vol. 9, 4 2013. DOI: HTTPS://DL.ACM.ORG/DOI/10.1145/2489253.2489265.

[109] Elgendi, M., "Fast QRS detection with an optimized knowledge-based method: Evaluation on 11 standard ECG databases," *PloS one*, Public Library of Science, vol. 8 (9), e73557, 2013.

[110] Kugler, P., Nordhus, P., Eskofier, B., “Shimmer, Cooja and Contiki: A new toolset for the simulation of on-node signal processing algorithms,” in *2013 IEEE International Conference on Body Sensor Networks*, 2013, pp. 1–6.

[111] Leonov, V., “Thermoelectric energy harvesting of human body heat for wearable sensors,” *IEEE Sensors Journal*, IEEE, vol. 13 (6), pp. 2284–2291, 2013.

[112] Martin, T., “The designer’s guide to the Cortex-M processor family: A tutorial approach.” 2013.

[113] Min, Y.-J., Kim, H.-K., Kang, Y.-R., Kim, G.-S., Park, J., Kim, S.-W., “Design of wavelet-based ECG detector for implantable cardiac pacemakers,” *Biomedical Circuits and Systems, IEEE Transactions on*, IEEE, vol. 7 (4), pp. 426–436, 2013.

[114] Priyanka, I. S., “Analysis ECG data compression techniques-a survey approach,” *International Journal of Emerging Technology and Advanced Engineering*, vol. 3 (2), pp. 544–548, 2013.

[115] Zou, Y., Han, J., Weng, X., Zeng, X., “An Ultra-Low Power QRS Complex Detection Algorithm Based on Down-Sampling Wavelet Transform,” *IEEE Signal Processing Letters*, vol. 20 (5), pp. 515–518, 2013.

[116] “BMX055, Digital 9-axis sensor,” version 1, Bosch Sensortx GmbH, 2012.

[117] Borgeson, J., “Ultra-low-power pioneers: TI slashes total MCU power by 50 percent with new “Wolverine” MCU platform,” 2012.

[118] “LCD Module LS013B7DH03,” Datasheet, Sharp, 2012.

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

[120] Oshana, R., “DSP for Embedded and Real-Time Systems.” Elsevier, 2012.

[121] Reuter, L., Schwarzmeier, A., Weigel, R., Fischer, G., “Energy budget in mobile health assistance systems,” *Proc. Biomedizinische Technik (BMT)*, Walter de Gruyter, vol. 57 (SI-1 Track-N), pp. 1070–1070, 2012.

[122] Ring, M., Jensen, U., Kugler, P., Eskofier, B., “Software-based performance and complexity analysis for the design of embedded classification systems,” in *Proceedings of the 21st International Conference on Pattern Recognition*, IEEE, 2012, pp. 2266–2269.

[123] Schwarzmeier, A., Reuter, L., Mena-Carrillo, J., Weber, W., Fischer, G., Weigel, R., Kissinger, D., “A low-power embedded communication platform for medical applications,” *Proc. Biomedizinische Technik (BMT)*, Walter de Gruyter, vol. 57 (SI-1 Track-N), pp. 1069–1069, 2012.

[124] Zidelmal, Z., Amirou, A., Adnane, M., Belouchrani, A., “QRS detection based on wavelet coefficients,” *Computer methods and programs in biomedicine*, Elsevier, vol. 107 (3), pp. 490–496, 2012.

[125] Anghel, I., Cioara, T., Salomie, I., Copil, G., Moldovan, D., Pop, C., “Dynamic frequency scaling algorithms for improving the CPU’s energy efficiency,” in *2011 IEEE 7th International Conference on Intelligent Computer Communication and Processing*, 2011, pp. 485–491. DOI: 10.1109/ICCP.2011.6047920.

[126] Bonfiglio, A., Rossi, D. D., “Wearable Monitoring Systems.” 2011.

[127] Chouakri, S., Bereksi-Reguig, F., Taleb-Ahmed, A., “QRS complex detection based on multi wavelet packet decomposition,” *Applied Mathematics and Computation*, Elsevier, vol. 217 (23), pp. 9508–9525, 2011.

[128] Horvath, D., Trinh, T. A., “A SystemC-Based Simulation Framework for Energy-Efficiency Evaluation of Embedded Networking Devices,” in *Energy-Aware Communications*, Lehnert, R., (Ed.), Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 169–180.

[129] Koomey, J., Berard, S., Sanchez, M., Wong, H., “Implications of Historical Trends in the Electrical Efficiency of Computing,” *IEEE Annals of the History of Computing*, vol. 33 (3), pp. 46–54, 2011.

[130] Leonov, V., “Energy Harvesting for Self-Powered Wearable Devices,” in *Wearable Monitoring Systems*, *Wearable Monitoring Systems*. 2011.

[131] Mamaghanian, H., Khaled, N., Atienza, D., Vandergheynst, P., “Compressed Sensing for Real-Time Energy-Efficient ECG Compression on Wireless Body Sensor Nodes,” *Biomedical Engineering, IEEE Transactions on*, vol. 58 (9), pp. 2456–2466, Sep. 2011. DOI: 10.1109/TBME.2011.2156795.

[132] Sun, F.-T., Kuo, C., Griss, M., “Pear: Power efficiency through activity recognition (for ECG-based sensing), ECG, Accelerometer,” in *Proc. Pervasive Health*, 2011, pp. 115–122.

[133] Fenzl, M., Schlegel, C., “Herzratenvariabilität -- Diagnosemittel für die Gesundheit: altersbezogene Effektgrössen,” *Schweizerische Zeitschrift fur Sportmedizin und Sporttraumatologie*, vol. 58 (4), p. 134, 2010.

[134] Frantz, G., Henkel, J., Rabaey, J., Schneider, T., Wolf, M., Batur, U., “Ultra-Low Power Signal Processing [DSP Forum],” *IEEE Signal Processing Magazine*, vol. 27 (2), pp. 149–154, 2010.

[135] Hosch, W., *et al.*, “The Britannica guide to numbers and measurement.” Britannica Educational Publishing, 2010.

[136] Molka, D., Hackenberg, D., Schöne, R., Müller, M. S., “Characterizing the energy consumption of data transfers and arithmetic operations on x86—64 processors,” in *International Conference on Green Computing*, 2010, pp. 123–133.

[137] Sarpeshkar, R., “Ultra-Low Power Bioelectronics: Fundamentals, Biomedical Applications, and Bio-Inspired Systems.” Cambridge University Press, 2010.

[138] Tverdal, M., “Operating system directed power reduction on EFM32,” Master’s thesis, Norwegian University of Science, Technology, Department of Computer, and Information Science, 2010.

[139] Au, L. K., Batalin, M. A., Stathopoulos, T., Bui, A. A., Kaiser, W. J., “Episodic sampling: Towards energy-efficient patient monitoring with wearable sensors,” in *Proc. IEEE EMBC*, 2009, pp. 6901–6905.

[140] Bisdikian, C., Branch, J., Leung, K. K., Young, R. I., “A letter soup for the quality of information in sensor networks,” in *2009 IEEE International Conference on Pervasive Computing and Communications*, 2009, pp. 1–6. DOI: 10.1109/PERCOM.2009.4912835.

[141] Bisdikian, C., Kaplan, L. M., Srivastava, M. B., Thornley, D. J., Verma, D., Young, R. I., “Building principles for a quality of information specification for sensor information,” in *2009 12th International Conference on Information Fusion*, 2009, pp. 1370–1377.

[142] Eriksson, J., Osterlind, F., Voigt, T., Finne, N., Raza, S., Tsiftes, N., Dunkels, A., "Demo abstract: Accurate power profiling of sensornets with the COOJA/MSPsim simulator," in *2009 IEEE 6th International Conference on Mobile Adhoc and Sensor Systems*, 2009, pp. 1060–1061.

[143] Fay, L., Misra, V., Sarpeshkar, R., "A micropower electrocardiogram amplifier, ECG," *IEEE Trans Biomed Circuits Syst.*, IEEE, vol. 3 (5), pp. 312–320, 2009.

[144] Holland, H.-J., **Tobola, A.,** Neubert, J., "KonMeVit: Kontinuierliches Mess- und Auswertesystem für Vitalparameter zur Präventionsunterstützung für Herz-Kreislauf-Patienten, Teilvorhaben SpO2-Hardware und Software, Abschlussbericht Teilvorhaben Fraunhofer IIS," Tech. Rep., München, 2009. DOI: `10.2314/GBV:646453688`.

[145] Jocke, S. C., Bolus, J. F., Wooters, S. N., Jurik, A., Weaver, A., Blalock, T., Calhoun, B., "A 2.6-$\mu$W sub-threshold mixed-signal ECG SoC, ECG," in *Symposium on VLSI Circuits*, 2009, pp. 60–61.

[146] **Tobola, A.,** "Einsatz agiler Methoden im sicherheitsrelevanten Umfeld am Beispiel Medizintechnik, Master's Thesis," Master's thesis, Technische Hochschule Nürnberg Georg Simon Ohm, 2009.

[147] Arzeno, N. M., Deng, Z.-D., Poon, C.-S., "Analysis of first-derivative based QRS detection algorithms," *IEEE Transactions on Biomedical Engineering*, IEEE, vol. 55 (2), pp. 478–484, 2008.

[148] Cajavilca, C., Varon, J., "Willem Einthoven: The development of the human electrocardiogram," *Resuscitation*, Elsevier, vol. 76 (3), pp. 325–328, 2008.

[149] Douniama, C., **Tobola, A.,** Wentzlaff, H., Benz, M., Norgall, T., Couronne, R., Weigand, C., "Pressure gauge, blood pressure gauge, method of determining pressure values, method of calibrating a pressure gauge, and computer program," US9119536B2, 2008, US Patent App. 12/812,280.

[150] Elgendi, M., Mahalingam, S., Jonkman, M., De Boer, F., "A robust QRS complex detection algorithm using dynamic thresholds," in *International Symposium on Computer Science and its Applications*, IEEE, 2008, pp. 153–158.

[151] Ghaffari, A., Golbayani, H., Ghasemi, M., "A new mathematical based QRS detector using continuous wavelet transform," *Computers & Electrical Engineering*, Elsevier, vol. 34 (2), pp. 81–91, 2008.

[152] Pollak, M., Mateu, L., Spies, P., "Step-up DC-DC-Converter with coupled inductors for low input voltages," *Fraunhofer IIS*, vol. 86, pp. 625–632, 2008.

[153] Tietze, U., Schenk, C., "Electronic Circuits -- Handbook for Design and Applications," 2nd ed. Springer Verlag, 2008.

[154] **Tobola, A.,** Weigand, C., "Garment for detecting respiratory movement," EP1731094B1, 2008, US Patent App. 12/742,260.

[155] Chiarugi, F., Sakkalis, V., Emmanouilidou, D., Krontiris, T., Varanini, M., Tollis, I., "Adaptive threshold QRS detector with best channel selection based on a noise rating system," in *2007 Computers in Cardiology*, IEEE, 2007, pp. 157–160.

[156] Mateu, L., Codrea, C., Lucas, N., Pollak, M., Spies, P., "Human body energy harvesting thermogenerator for sensing applications," in *Proc. SensorComm*, 2007, pp. 366–372.

[157] Spies, P., Pollak, M., Rohmer, G., "Energy harvesting for mobile communication devices," in *Proc. INTELEC*, 2007, pp. 481–488.

[158] "TMP102 Low-Power Digital Temperature Sensor With SMBus and Two-Wire Serial-Interface in SOT563," Datasheet, Texas Instruments, 2007.

[159] "DIN EN 60601-2-27:2006-08; VDE 0750-2-27:2006-08: Medical electrical equipment. Part 2-27. Particular requirements for the basic safety and essential performance of electrocardiographic monitoring equipment," 2006.

[160] Johnston, W. S., Mendelson, Y., "Investigation of signal processing algorithms for an embedded microcontroller-based wearable pulse oximeter.," *Conference proceedings : ... Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE Engineering in Medicine and Biology Society. Annual Conference*, vol. 2006, pp. 5888–91, 2006.

[161] Fischer, G., "Method of analysing a receiver and /or transmitter chain," EP1521378B1, 2005.

[162] Teel, J. C., "Understanding power supply ripple rejection in linear regulators," *Analog Applications Journal*, 2005.

[163] Yao, J., Warren, S., "Applying the ISO/IEEE 11073 standards to wearable home health monitoring systems," *Journal of clinical monitoring and computing*, Springer, vol. 19 (6), pp. 427–436, 2005.

[164] Anliker, U., Ward, J. A., Lukowicz, P., Troster, G., Dolveck, F., Baer, M., Keita, F., Schenker, E. B., Catarsi, F., Coluccini, L., *et al.*, "AMON: a wearable multiparameter medical monitoring and alert system," *Information Technology in Biomedicine, IEEE Transactions on*, IEEE, vol. 8 (4), pp. 415–427, 2004.

[165] Christov, I. I., "Real time electrocardiogram QRS detection using combined adaptive threshold," *Biomedical engineering online*, Springer, vol. 3 (1), p. 28, 2004.

[166] McConnell, S., "Code Complete, Second Edition." Redmond, WA, USA: Microsoft Press, 2004.

[167] "ADA400A Differential Preamplifier," Instruction Manual, version 070-9164-02, Tektronix, 2003.

[168] Kohler, B., Hennig, C., Orglmeister, R., "QRS detection using zero crossing counts," *Applied genomics and proteomics*, OPEN MINDS JOURNAL LTD, vol. 2 (2), pp. 138–145, 2003.

[169] Rao, R., Vrudhula, S., Rakhmatov, D. N., "Battery modeling for energy aware system design," *Computer*, vol. 36 (12), pp. 77–87, 2003.

[170] "Cardiac monitors, heart rate meters, and alarms, American National Standard (ANSI/AAMI EC13:2002)," Arlington, VA, 2002.

[171] Hamilton, P., "Open source ECG analysis," in *Computers in Cardiology*, E.P. Ltd., 2002, pp. 101–104.

[172] Kohler, B., Hennig, C., Orglmeister, R., "The principles of software QRS detection," *IEEE Engineering in Medicine and Biology Magazine*, vol. 21 (1), pp. 42–57, 2002.

[173] Frank, D. J., Dennard, R. H., Nowak, E., Solomon, P. M., Taur, Y., Hon-Sum Philip Wong, "Device scaling limits of Si MOSFETs and their application dependencies," *Proceedings of the IEEE*, vol. 89 (3), pp. 259–288, 2001.

[174] Moyer, B., "Low-power design for embedded processors," *Proceedings of the IEEE*, IEEE, vol. 89 (11), pp. 1576–1587, 2001.

[175] Cetin, A. E., Köymen, H., “Compression of digital biomedical signals,” *The Biomedical Engineering Handbook: Second Edition. Joseph D. Bonzino, Ed. CRC Press LLC*, 2000.

[176] Afonso, V. X., Tompkins, W. J., Nguyen, T. Q., Luo, S., “ECG beat detection using filter banks, ECG, ”Heart rate”, QRS,” *Biomedical Engineering, IEEE Transactions on*, IEEE, vol. 46 (2), pp. 192–202, 1999.

[177] Hahn, R., Reichl, H., “Batteries and power supplies for wearable and ubiquitous computing,” in *Digest of Papers. Third International Symposium on Wearable Computers*, 1999, pp. 168–169.

[178] Kadambe, S., Murray, R., Boudreaux-Bartels, G. F., “Wavelet transform-based QRS complex detector,” *IEEE Transactions on Biomedical Engineering*, vol. 46 (7), pp. 838–848, 1999.

[179] Martin, T. L., Siewiorek, D. P., “Non-ideal battery properties and low power operation in wearable computing,” in *Digest of Papers. Third International Symposium on Wearable Computers*, 1999, pp. 101–106.

[180] Eyre, J., Bier, J., “DSP processors hit the mainstream,” *Computer*, IEEE, vol. 31 (8), pp. 51–59, 1998.

[181] Vijaya, G., Kumar, V., Verma, H., “ANN-based QRS-complex analysis of ECG,” *Journal of medical engineering & technology*, Taylor & Francis, vol. 22 (4), pp. 160–167, 1998.

[182] Bahoura, M., Hassani, M., Hubin, M., “DSP implementation of wavelet transform for real time ECG wave forms detection and heart rate analysis,” *Computer methods and programs in biomedicine*, Elsevier, vol. 52 (1), pp. 35–44, 1997.

[183] Herre, J., Seitzer, D., Brandenburg, K.-H., Eberlein, E., “Process for reducing data in the transmission and/or storage of digital signals from several interdependent channels, Wireless Radio Bluetooth Network Networks IoT, Compression,” US Patent 5,703,999, 1997.

[184] Inoue, H., Miyazaki, A., Iwasaki, S., Shimazu, M., Katsura, T., Teranishi, A., “Detection of QRS Complex in ECG Using a Wavelet Transform,” in *ITC-CSCC: International Technical Conference on Circuits Systems, Computers and Communications*, 1997, pp. 361–364.

[185] Ruha, A., Sallinen, S., Nissila, S., “A real-time microprocessor QRS detector system with a 1-ms timing accuracy for the measurement of ambulatory HRV,” *IEEE Transactions on Biomedical Engineering*, IEEE, vol. 44 (3), pp. 159–167, 1997.

[186] Sahambi, J., Tandon, S., Bhatt, R., “Using wavelet transforms for ECG characterization. An on-line digital signal processing system,” *IEEE Engineering in Medicine and Biology Magazine*, IEEE, vol. 16 (1), pp. 77–83, 1997.

[187] Enz, C. C., Vittoz, E. A., “CMOS low-power analog circuit design,” in *Emerging Technologies: Designing Low Power Digital Systems*, IEEE, 1996, pp. 79–133.

[188] Task Force of the European Society of Cardiology the North American Society of Pacing Electrophysiology, “Heart Rate Variability: Standards of measurement, physiological interpretation, and clinical use.,” *Circulation*, vol. 93, pp. 1043–1065, 1996.

[189] Chandrakasan, A. P., Brodersen, R. W., “Minimizing power consumption in digital CMOS circuits,” *Proceedings of the IEEE*, IEEE, vol. 83 (4), pp. 498–523, 1995.

[190] Li, C., Zheng, C., Tai, C., "Detection of ECG characteristic points using wavelet transforms," *IEEE Transactions on biomedical Engineering*, IEEE, vol. 42 (1), pp. 21–28, 1995.

[191] Poli, R., Cagnoni, S., Valli, G., "Genetic design of optimum linear and nonlinear QRS detectors," *IEEE Transactions on Biomedical Engineering*, IEEE, vol. 42 (11), pp. 1137–1141, 1995.

[192] Suppappola, S., Sun, Y., "Nonlinear transforms of ECG signals for digital QRS detection: a quantitative analysis," *IEEE Transactions on Biomedical Engineering*, IEEE, vol. 41 (4), pp. 397–400, 1994.

[193] GROSSMAN, P., KOLLAI, M., "Respiratory sinus arrhythmia, cardiac vagal tone, and respiration: Within-and between-individual relations, Respiration+Respiratory, ECG," *Psychophysiology*, Wiley Online Library, vol. 30 (5), pp. 486–495, 1993.

[194] Hu, Y. H., Tompkins, W. J., Urrusti, J. L., Afonso, V. X., "Applications of artificial neural networks for ECG signal detection and classification.," *Journal of electrocardiology*, vol. 26, pp. 66–73, 1993.

[195] Trahanias, P., "An approach to QRS complex detection using mathematical morphology," *IEEE Transactions on Biomedical Engineering*, IEEE, vol. 40 (2), pp. 201–205, 1993.

[196] Hosticka, B. J., "Applications of Mixed Analog/Digital Design," in *ESSCIRC '92: Eighteenth European Solid-State Circuits conference*, 1992, pp. 27–34.

[197] Kadambe, S., Murray, R., Boudreaux-Bartels, G., "The dyadic wavelet transform based QRS detector (ECG analysis)," in *Conference Record of the Twenty-Sixth Asilomar Conference on Signals, Systems & Computers*, IEEE, 1992, pp. 130–134.

[198] Sun, Y., Suppappola, S., Wrublewski, T. A., "Microcontroller-based real-time QRS detection.," *Biomedical instrumentation & technology*, vol. 26 (6), p. 477, 1992.

[199] Xue, Q., Hu, Y. H., Tompkins, W. J., "Neural-network-based adaptive matched filtering for QRS detection," *IEEE Transactions on Biomedical Engineering*, IEEE, vol. 39 (4), pp. 317–329, 1992.

[200] Coast, D. A., Stern, R. M., Cano, G. G., Briller, S. A., "An approach to cardiac arrhythmia analysis using hidden Markov models," *IEEE Transactions on biomedical Engineering*, IEEE, vol. 37 (9), pp. 826–836, 1990.

[201] Gritzali, F., "Towards a generalized scheme for QRS detection in ECG waveforms," *Signal processing*, Elsevier, vol. 15 (2), pp. 183–192, 1988.

[202] Hamilton, P. S., Tompkins, W. J., "Quantitative Investigation of QRS Detection Rules Using MIT/BIH Arrhythmia Database, QRS," *IEEE Trans Biomed Eng.*, vol. BME-33 (12), 1986.

[203] Papakonstantinou, G., Skordalakis, E., Gritzali, F., "An attribute grammar for QRS detection," *Pattern recognition*, Elsevier, vol. 19 (4), pp. 297–303, 1986.

[204] Hosticka, B. J., "Performance comparison of analog and digital circuits," *Proceedings of the IEEE*, vol. 73 (1), pp. 25–29, 1985.

[205] Pan, J., Tompkins, W., "A Real-Time QRS Detection Algorithm, ECG, "Heart rate", QRS," *IEEE Transactions on Biomedical Engineering*, vol. BME-32 (3), pp. 230–236, 1985. DOI: 10.1109/TBME.1985.325532.

[206] Weicker, R. P., “Dhrystone: a synthetic systems programming benchmark,” *Communications of the ACM*, ACM New York, NY, USA, vol. 27 (10), pp. 1013–1030, 1984.

[207] Ligtenberg, A., Kunt, M., “A robust-digital QRS-detection algorithm for arrhythmia monitoring,” *Computers and Biomedical Research*, Elsevier, vol. 16 (3), pp. 273–286, 1983.

[208] Thakor, N. V., Webster, J., Tompkins, W., “Optimal QRS detector,” *Medical and Biological Engineering and Computing*, Springer, vol. 21 (3), pp. 343–350, 1983.

[209] Towe, B. C., “Comments on ”Ground-Free ECG Recording with Two Electrodes”,” *IEEE Transactions on Biomedical Engineering*, vol. BME-28 (12), pp. 838–839, 1981.

[210] Hosticka, B. J., “Dynamic CMOS amplifiers,” *IEEE Journal of Solid-State Circuits*, vol. 15 (5), pp. 881–886, 1980.

[211] Thakor, N. V., Webster, J. G., “Ground-Free ECG Recording with Two Electrodes,” *IEEE Transactions on Biomedical Engineering*, vol. BME-27 (12), pp. 699–704, 1980.

[212] Tompkins, W. J., Webster, J. G., “Design of microcomputer-based medical instrumentation, ECG.” Prentice Hall Professional Technical Reference, 1980.

[213] Ziv, J., Lempel, A., “Compression of individual sequences via variable-rate coding,” *IEEE Transactions on Information Theory*, vol. 24 (5), pp. 530–536, Sep. 1978. DOI: 10.1109/TIT.1978.1055934.

[214] Dennard, R. H., Gaensslen, F. H., Yu, H., Rideout, V. L., Bassous, E., LeBlanc, A. R., “Design of ion-implanted MOSFET’s with very small physical dimensions,” *IEEE Journal of Solid-State Circuits*, vol. 9 (5), pp. 256–268, 1974.

[215] Moore, G. E., *et al.*, “Cramming more components onto integrated circuits,” 1965.

[216] Landauer, R., “Irreversibility and Heat Generation in the Computing Process,” *IBM Journal of Research and Development*, vol. 5 (3), pp. 183–191, 1961.

[217] Shannon, C. E., “The mathematics theory of communication,” *Bell Syst. Tech. J*, vol. 27, pp. 379–423, 1948.

[218] Hartley, R. V., “Transmission of information 1,” *Bell System technical journal*, Wiley Online Library, vol. 7 (3), pp. 535–563, 1928.

[219] Johnson, J. B., “Thermal agitation of electricity in conductors,” *Physical review*, APS, vol. 32 (1), p. 97, 1928.

[220] Nyquist, H., “Certain Topics in Telegraph Transmission Theory,” *Transactions of the American Institute of Electrical Engineers*, vol. 47 (2), pp. 617–644, 1928.

[221] Einthoven, W., “Ueber die Form des menschlichen Electrocardiogramms,” *Archiv für die gesamte Physiologie des Menschen und der Tiere*, vol. 60 (3), pp. 101–123, 1895.

[222] Waller, A. D., “A demonstration on man of electromotive changes accompanying the heart’s beat,” *The Journal of physiology*, Wiley-Blackwell, vol. 8 (5), p. 229, 1887.

## List of Tables

## List of Figures

# Appendix A

# Definitions for Signal Flow Diagrams

For this thesis, a custom variant of signal flow diagrams was defined and used.

## A.1. Overview

Most signal flow diagrams use processing blocks, while the signals are passed from processing block to processing block represented by arrows. The arrows are labeled with a letter or a name. The signal flow diagrams are introduced here to take advantage of Signal Blocks in addition to Processing Blocks. With this, processing blocks are consuming signal blocks and generating signal blocks. Processing blocks are represented in rectangular shapes. Signal blocks are represented in hexagonal shapes. The reason for introducing this type of diagram was for additional clarity when analyzing systems for energy consumption.

## A.2. Motivation

Signal flow diagrams were mandatory for understanding and optimizing this work's entire signal processing chain. However, signal flow diagrams exist in different variants, more or less standardized in scientific and engineering domains. A classic diagram in electrical engineering and information theory is the Signal Flow Graph. Signal flow graphs, such as those used by Shannon, are suited for mathematical solutions. Besides this, many less formal signal flow diagrams exist, often with mixed and unclear meanings for the arrows. On the contrary, in software engineering, so-called Activity Diagrams were specified in Unified Modeling Language in version 2 (UML2). Activity diagrams inspired the idea of introducing signal blocks and processing blocks. In activity diagrams, objects can be passed between processing blocks. The objects can be defined by type and supplemented with additional information. However, activity diagrams are limited to software, while signal processing is about hardware and software.

## A.3. Yet Another Signal Flow Diagram

This diagram type's main idea is to introduce signal blocks in addition to processing blocks. Further, signals can be analog or digital. Processing can also be analog or digital. Therefore, processing and signals are represented in two different shapes.

- Processing blocks are represented in rectangular shapes.

- Signal blocks are represented in hexagonal shapes.

Figure A.1 illustrates an abstract example of a signal processing chain. Further, there is no need to define another shape for sensors. Sensors were defined as processing blocks (rectangular shapes) in this type of diagram. Additionally, any shape must contain at least a unique name. Optionally, within any shape, information about the properties should be given. Here is a list of supplementary information useful for system analysis, communication with stakeholders, and development.

- Signal shapes should contain the data rate.
- The bandwidth of a signal processing unit is important information for energy analysis.
- The cut-off frequencies should be given in the case of filters.
- The delay that a signal processing unit may cause should be added if known.
- Power consumption for the signal processing block is mandatory for energy analysis.
- In general, configurable signal processing blocks should include configuration parameters.
- Software signal blocks should provide a unique name referring to the code (module name, function, method).
- Hardware signal blocks may refer to a sub-circuit or integrated circuit.
- For tractability purposes the requirement number should be included if available.

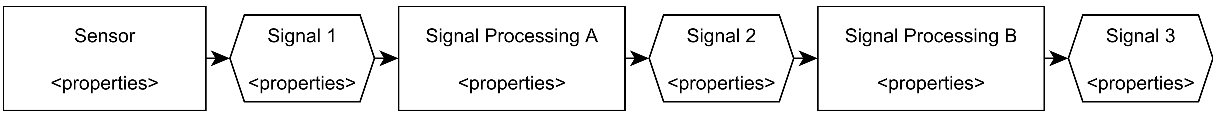


**Figure A.1.:** Signal processing blocks are represented in rectangular shapes. Signals are represented in hexagonal shapes. Sensors are defined as signal processing blocks.

## A.4. Connections

Directed arrows connect both types of blocks. Processing blocks can have one or more inputs of the same or different kinds. Equally, processing blocks can have one or more outputs of the same or different kinds. It is recommended that each output and input of a processing block (rectangular) has to be connected to a signal block (hexagonal). Exceptionally, for abstraction, a processing block can connect to another processing block directly, without signals in between. However, connecting a signal block with a subsequent signal block is not allowed. The connection between any kind of two blocks has to be represented by a directional arrow. The arrows always show the signal flow direction. Arrows with dual directions are not permitted. If the signal flow is bidirectional, this should be represented by separate arrows and additional signals, instead.

## A.5. Summary and Outreach

This diagram type was introduced for analyzing signal processing systems in this work. With the introduction of signal blocks, additional information can be provided within a signal block. Further, with a signal block, it is clear how many times other algorithms reused a signal. The reuse of already computed data is important for energy conservation. With this diagram type, energy saving was identified more clearly. Beyond this work, this diagram type was successfully adapted and extended for professional projects. It turned out to be a powerful specification document when designing and communicating requirements for new sensor systems with other professionals and customers.

# Appendix B

# Algorithm “Apnea”

This appendix provides the apnea detection algorithm used for benchmarks in chapter section 5.2 for those who wish to reproduce the results. The algorithm was developed for the research project *Smart Sensors A* and published in the public project report [77]. The aim of this project was the commercial viability of integrating sensors into clothing fabrics. It involves the examination of technological solutions for assistance systems that will inconspicously and reliably support the elderly or disabled, particularly in their day-to-day activities, and thereby enhance their quality of life.

Figure B.1 shows the feed-forward signal processing chain of the algorithms in four stages. First, a median filter processed the respiratory signal to remove any spikes from the sensor. In the second stage, a low-pass filter removes high-frequency components. In the third stage, the signal peak-to-peak level is estimated. Finally, a decision stage classifies the signal strengths into two classes: *normal breathing* or *apnea*. For comparison, a similar algorithm was published by Almazaydeh [105].

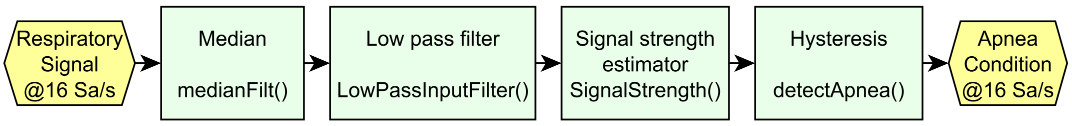


**Figure B.1.:** Abstract signal flow block diagram of the *Apnea Detection Algorithm* according to appendix A. More detailed instructions for this type of signal flow diagram were included in the appendix A.

Finally, this is the corresponding source code used for benchmarking:

**Code B.1:** apnea.c

```
/**
 *  @brief Apnea detection algorithm
 *
 *  This module implements the apne detection algorithm designd to process
 *  respiratory data from RespiSENS-Modules V3 and V4.
 *
 *  @author Andreas Tobola
 *  @since 2013-12-10
 *
 *  Project: Smart Sensors A
 *
 *  Version history:
 *  001   2013-12-10    Apnea detection algorithm according to matlab model
    II
 *  002   2013-12-11    Hysteresis thresold changed
 *  003   2013-12-11    Statistical filter added, filter optimized
 *                      and hysteresis adjustment.
 *  004   2013-12-11    Statistical filter removed. Programm does not fit
    into memory.
 *  005   2014-09-24    Statistical filter put back in.
 */
#include "apnea.h"
#include <stdio.h>
#include <string.h>
#include <stdlib.h>

#ifdef DEBUG_FILE
FILE *tmpFile;
#endif

static volatile uint8_t apneaState = 0;
static float sigStr = 0;

// Internal prototypes
float LowPassInputFilter(const uint16_t);
float SignalStrength(const float);
void detectApnea(const uint16_t);
uint16_t medianFilt(const uint16_t);

/**
 * @author Andreas Tobola
 * Output interface
 * @return Returns signal strength
 */
float getSignalStrength(void)
{
    return sigStr;
}

/**
 * @author Andreas Tobola
 * Input interface
 * @param respiratory signal sample (14 bit, unsigned)
 */
void pushSample(const uint16_t sample)
{
    detectApnea(sample);
}

/**
 * @author Andreas Tobola
 * Output interface
 * @return Returns 1 if apnea detected, otherwise 0
 */
uint8_t isApnea(void)
{
    return apneaState;
```

```
}

/**
 * @author Andreas Tobola
 * Detects apneas by processing an respiratory input signal
 * given by "Smart Sensors A Stramper"
 * @param respiratory signal sample (14 bit, unsigned)
 */
void detectApnea(const uint16_t inSample)
{
    float lpFiltered;
    uint16_t mfSample;
    const float apneaTresholdL = 0.95f;
    const float apneaTresholdU = 1.7f;

    // Median filter
    mfSample = medianFilt(inSample);

    // Step 1: Filter input signal
    lpFiltered = LowPassInputFilter(mfSample);

    // Step 2: Calculate signal strength
    sigStr = SignalStrength(lpFiltered);

    // Step 3: Hysteresis
    if (apneaState == 1)
    {
        if (sigStr > apneaTresholdU)
        {
            apneaState = 0;
        }
    }
    else
    {
        if (sigStr < apneaTresholdL)
        {
            apneaState = 1;
        }
    }

#ifdef DEBUG_FILE
    fprintf(tmpFile, "%u,%f,%f,%u,%u\n",inSample,lpFiltered,sigStr,
        apneaState,mfSample);
#endif

}

/**
 * @author Andreas Tobola
 * Output interface
 * @return Returns 1 if apnea detected, otherwise 0
 */
float LowPassInputFilter(const uint16_t x)
{
    const float alpha = 0.5;
    static float y1 = 0;
    float y;  // Output

    // start condition to force immediately steady state
    if (y1 == 0)
    {
        y1 = x * alpha;
    }

    y = x * alpha - y1 * (alpha - 1);
    y1 = y;

    return y;
}

/**
 * @author Andreas Tobola
 * Calculates the signal strength using extrema detection algorithm
 * @return signal strength
 */
float SignalStrength(const float x)
{
    const float gamma = 1.0f;
    const float beta = 0.017f;
    float alpha;
    float delta;
    static float yu = 1024;  // upper border
    static float yl = 1000;  // lower border

    // calculate upper border
    if (x > yu)
    {
        alpha = gamma;
    }
    else
    {
        alpha = beta;
    }
    yu = yu * (1.0f - alpha) + x * alpha;

    // calculate lower border
    if (x < yl)
    {
        alpha = gamma;
    }
    else
    {
        alpha = beta;
    }
    yl = yl * (1.0f - alpha) + x * alpha;

    // calculate signal strength defined as the difference
    // between upper and lower border
    delta = yu - yl;

    return delta;
}
```

```c

/**
 * @author Andreas Tobola
 * Calculates the median
 * @param input value x
 * @return filtered value
 */
uint16_t medianFilt(const uint16_t x)
{
    const uint8_t Ns = 5;
    const uint8_t km = (Ns - 1) / 2; // Index in der Buffermitte
    uint16_t so[5];         //[Ns]
    static uint16_t ms[5] = { 0, 0, 0, 0, 0 };
    static uint8_t k = 0; // Buffer index
    uint16_t med;
    uint16_t temp;
    uint8_t j;
    uint8_t i;

    //k = (k+1) % Ns;
    k++;
    if (k >= Ns)
    {
        k = 0;
    }

    ms[k] = x;

    memcpy(so, ms, 2 * Ns);

    // Sort
    for (i = 0; i < Ns - 1; i++)
    {
        for (j = i + 1; j < Ns; j++)
        {
            if (so[j] < so[i])
            {
                // swap elements
                temp = so[i];
                so[i] = so[j];
                so[j] = temp;
            }
        }
    }

    med = so[km]; // Median

    return med;
}
```

# Appendix C

# Outreach

## Authored Publications

[PUB1] Scholl, C., Spiegler, M., Ludwig, K., Eskofier, B. M., **Tobola, A.,** Zanca, D., "An Integrated Framework for Data Quality Fusion in Embedded Sensor Systems," *Sensors*, vol. 23 (8), 2023. DOI: 10.3390/S23083798.

[PUB2] **Tobola, A.,** "Introducing PyDCC – a Python module for the DCC," *2nd international DCC-Conference 01 - 03 March 2022 Proceedings*, Physikalisch-Technische Bundesanstalt (PTB), 2022. DOI: 10.7795/820.20220411.

[PUB3] Scholl, C., **Tobola, A.,** Ludwig, K., Zanca, D., Eskofier, B. M., "A Smart Capacitive Sensor Skin with Embedded Data Quality Indication for Enhanced Safety in Human–Robot Interaction," *Sensors*, vol. 21 (21), 2021. DOI: 10.3390/S21217210.

[PUB4] **Tobola, A.,** Leutheuser, H., Pollak, M., Spies, P., Hofmann, C., Weigand, C., Eskofier, B. M., Fischer, G., "Self-Powered Multiparameter Health Sensor," *IEEE Journal of Biomedical and Health Informatics*, vol. 22 (1), pp. 15–22, Jan. 2018. DOI: 10.1109/JBHI.2017.2708041.

[PUB5] Leutheuser, H., Lang, N., Gradl, S., Struck, M., **Tobola, A.,** Hofmann, C., Anneken, Eskofier, B., "Book Chapter: Textile Wearable Technologies for Sports and Medical Applications," in *Smart Textiles: Fundamentals, Design, and Interaction*. Springer, 2017.

[PUB6] **Tobola, A.,** Hofmann, C., "Schlussbericht HE2mT - High-Level Entwurfsmethoden energieoptimierter, mobiler Telemonitoringsysteme, Schlussbericht für das Forschungsvorhaben HE2mT - Teilbericht Fraunhofer IIS, Final report on public funded research project HE2mT," Tech. Rep., Frauhofer IIS, 2017. DOI: 10.2314/GBV:1012240010.

[PUB7] Kindt, P., Yunge, D., **Tobola, A.,** Fischer, G., Chakraborty, S., "Dynamic service switching for the medical IoT," *IEEE 27th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, pp. 1–7, 2016.

[PUB8] **Tobola, A.,** Leutheuser, H., Schmitz, B., Hofmann, C., Struck, M., Weigand, C., Eskofier, B. M., Fischer, G., "Battery Runtime Optimization Toolbox for Wearable Biomedical Sensors," *Proc. IEEE 13th International Conference on Wearable and Implantable Body Sensor Networks (BSN)*, pp. 199–204, Jun. 2016. DOI: 10.1109/BSN.2016.7516259.

[PUB9] Leutheuser, H., Gradl, S., Eskofier, B., **Tobola, A.,** Lang, N. R., Anneken, L., Arnold, M., Achenbach, S., "Arrhythmia classification using RR intervals: Improvement with sinusoidal regression feature," *IEEE 12th International Conference on Wearable and Implantable Body Sensor Networks (BSN)*, pp. 1–5, 2015.

[PUB10] Sauter, C. U., Heinloth, M., **Tobola, A.,** Pensky, N., Weigand, C., "Risk Management for Medical Devices in Research Projects," *Proc. Biomedizinische Technik (BMT)*, 2015.

[PUB11] Schmitz, B., Hofmann, C., Maestre, R., Bleda, A. L., **Tobola, A.,** Melcher, V., Gent, J., van, "Continuous vital monitoring and automated alert message generation for motorbike riders," *Computing in Cardiology Conference (CinC)*, IEEE, pp. 397–400, 2015. DOI: `10.1109/CIC.2015.7408670`.

[PUB12] **Tobola, A.,** Espig, C., Streit, F. J., Korpok, O., *et al.*, "Scalable ECG Hardware and Algorithms for Extended Runtime of Wearable Sensors," *Proc. IEEE International Symposium on Medical Measurements and Applications (MeMeA)*, 2015.

[PUB13] **Tobola, A.,** Rybalko, R., Korpok, O., Streit, F. J., Espig, C., Hofmann, C., Lang, N., Struck, M., Weigand, C., Fischer, G., "ULPSEK -- Ultra-Low-Power Sensor Evaluation Kit," *Proc. Biomedizinische Technik (BMT)*, 2015.

[PUB14] **Tobola, A.,** Streit, F. J., Korpok, O., Espig, C., *et al.*, "Sampling Rate Impact on Energy Consumption of Biomedical Signal Processing Systems," *Proc. IEEE 12th International Conference on Wearable and Implantable Body Sensor Networks (BSN)*, 2015.

[PUB15] **Tobola, A.,** "Intelligente Sensorik : Schlussbericht," Tech. Rep., 2015. DOI: `10.2314/GBV:868717789`.

[PUB16] **Tobola, A.,** Korpok, O., Leutheuser, H., Schmitz, B., Hofmann, C., Struck, M., Weigand, C., Eskofier, B., Heuberger, A., Fischer, G., "System Design Impacts on Battery Runtime of Wearable Medical Sensors," *Proc. 2nd International Conference on Mobile and Information Technologies in Medicine (MobileMed)*, 2014.

[PUB17] Jeleazcov, C., **Tobola, A.,** Weiss, M., Weigand, C., Schüttler, J., "Pharmacodynamic modeling of changes in pulse waveform during induction of propofol anaesthesia in volunteers: Comparison between invasive and continuous non-invasive measurements of puls pressure," *Journal of Clinical Monitoring and Computing*, vol. 26, pp. 227–254, Aug. 2012.

[PUB18] **Tobola, A.,** Hofmann, C., Weigand, C., "Requirement Engineering in Health Care and Telemedicine, FOCUS Session Intelligent Sensors," *Proc. Biomedizinische Technik (BMT)*, 2012.

[PUB19] Holland, H.-J., **Tobola, A.,** Neubert, J., "KonMeVit: Kontinuierliches Mess- und Auswertesystem für Vitalparameter zur Präventionsunterstützung für Herz-Kreislauf-Patienten, Teilvorhaben SpO2-Hardware und Software, Abschlussbericht Teilvorhaben Fraunhofer IIS," Tech. Rep., München, 2009. DOI: `10.2314/GBV:646453688`.

[PUB20] **Tobola, A.,** "Einsatz agiler Methoden im sicherheitsrelevanten Umfeld am Beispiel Medizintechnik, Master's Thesis," Master's thesis, Technische Hochschule Nürnberg Georg Simon Ohm, 2009.

[PUB21] Douniama, C., **Tobola, A.,** "Investigation Of Additional Parameters For The Enhancement Of A Blood Pressure Model Based On Pulse Transit Time," *Proc. 2nd International Conference, Monitoring Sleep and Sleepiness with New Sensors within Medical and Industrial Applications (SENSATION)*, Jul. 2007.

[PUB22] **Tobola, A.,** Douniama, C., "Evaluation of alternative derivation areas for plethysmography and pulse oximetry," *Proc. 2nd International Conference, Monitoring Sleep and*

*Sleepiness with New Sensors within Medical and Industrial Applications (SENSATION)*, 2007.

[PUB23] **Tobola, A.,** "Verarbeitung von Signalen eines Remissionssensors zum Einsatz in der Puloximetrie, Diplomarbeit," 2005.

## List of Patents

[PAT1] Invention disclosure: Digital Twin for Sensor System Energy Optimization and Supervision, German Title: Steuerung eines Sensorsystems, (**Tobola**, 2022)
Patent application describing a system with essential parts: (1) Sensor system extended by a self-monitoring unit for power consumption profiling chapter 4 with the technical solution in section 3.7, (2) a model for the power consumption of the sensor system as described in chapter 7, (3) a real-time communication between a real asset (sensor system) and the energy model to update model parameters and to identify anomalies. This solution aims to detect anomalies in power consumption which can be safety or security related.

[PAT2] Invention disclosure: Data-quality-aware Sensor Communication, (Scholl, **Tobola**, Ludwig, 2021)
This patent described a procedure to transmit data based on the sensor data quality to reduce power consumption to a necessary level.

[PAT3] Patent EP4063795A1: Sensorsystem zur Messung der Variabilität der elektrischen Spannung einer Energieversorgung (**Tobola**, 2021)
System for quality monitoring the sensor's supply voltage. In particular, a circuit to release the load on a microcontroller and thereby save energy.

[PAT4] Patent EP2218395A3: Apparatus and Method for Detecting at Least One Vital Parameter of a Person; Vital Parameter Detection System (Ciancitto, **Tobola**, Hofmann, Couronne, 2010)
An optical sensor apparatus for detecting vital parameters of a person with e.g. a pair of spectacles.

[PAT5] Patent US9119536B2: Pressure gauge, blood pressure gauge, method of determining pressure values, method of calibrating a pressure gauge, and computer program (Douniama, **Tobola**, Wentzlaff, Benz, Norgall, Couronne, Weigand, 2008)
A pressure gauge for determining at least one pressure value describing a pressure of a fluid flowing in a pulsating manner in a phase of the pulsating flow, includes a pulse wave characterizer.

[PAT6] Patent WO2010040452A1: Apparatus for recording and monitoring at least one vital sign of a person in a motor vehicle (Couronne, Moersdorf, **Tobola**, Ershov, Gassmann, Haevescher, RakeJoerg, 2008)
An apparatus for recording at least one vital sign of a person in a motor vehicle with an optoelectronic sensor arrangement.

[PAT7] Patent EP2174590B1: Device and method for sensing respiration of a living being (**Tobola**, Ershov, Wissendheit, Couronne, 2008)
The patent describes a device for sensing the respiration of a living being. The device is integrated into the seat of a vehicle, including an active transmitter configured to generate a magnetic or electromagnetic field. A sensor is arranged on the torso of the living being and configured to provide a signal which depends on the magnetic or electromagnetic field and on a change in the distance caused by the respiration of the living being between the active transmitter and the sensor.

[PAT8] Patent EP2217145B1: Clothing article for detecting breathing movement (**Tobola**, Weigand, 2007)
The invention relates to an article of clothing for detecting a breathing movement of a living being, wherein the clothing article can be pulled over the thorax of the living being and has an electrical conductor which can be integrated into the clothing article at the height of the thorax of the living being.

[PAT9] Patent WO2007096054A2: Adaptive filtering for determining vital parameters more reliably, (**Tobola**, Vogl, Moersdorf, 2006)
The invention relates to a device for reducing the noise component in a time-discrete (biomedical) signal which also has a useful component.

[PAT10] Patent WO2007104390A2: Spread-spectrum method for determining physiological parameters (**Tobola**, Vogl, Moersdorf, 2006)
An optical transmitter sending pseudo-random sequences, a receiver, and an extractor are provided for determining a physiological parameter of a living being. This method provides additional interference suppression in the presence of strong light sources for determining oxygen saturation and heart rate using an optical system like a pulse oximeter.

[PAT11] Patent EP1909189A3: Spectral analysis for a more reliable determination of physiological parameters (**Tobola**, Vogl, Moersdorf, 2006)
This patent describes an apparatus for determining a spectral ratio between a first signal with a first spectrum that depends on a biological quantity and a second signal with a second spectrum that depends on a biological quantity. An algorithm uses a least-squares solution to compute ratios between the harmonics magnitudes from the first spectrum and the harmonic magnitudes from the second spectrum. This ratio is then used to compute oxygen saturation in the arterial blood with high suppression of motion artifacts.

## Academic Lectures on Biomedical Monitoring with Emphasis on Low-Power Design

[AL1] Lectures: Biomedical Signal Processing with Focus on Low-Power Design (Tobola, ongoing since 2013)
The contribution involved preparing and delivering lectures, each totaling two semester weekly hours, during the spring semester for medical engineering

students at the Technische Hochschule Nürnberg. Additionally, each spring and fall semester, the author was responsible for creating a test based on the lecture material and later reviewing the test results, further ensuring the students' comprehension and progress in the subject.

[AL2] Guest Lecture: Introducing the Ultra-Low-Power Sensor Evaluation Kit (Tobola, 2016)
Guest lecture for students at the Institute for Electronics Engineering (LTE) of Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU)
Invited by Dr.-Ing. Heinrich Milosiu

[AL3] Lectures: Biomedical Signal Processing with Focus on Low-Power Design (Tobola, 2012 to 2014)
The contribution was preparing and giving spring semester lectures for medical engineering students at the Hochschule Furtwangen, Hochschulcampus Tuttlingen. Additionally, each semester, the author was responsible for creating a test based on the lecture material and later reviewing the test results.

[AL4] Guest Lecture: Introduction to Photoplethysmography and Pulseoxymetry (Tobola, 2014)
Guest in the lecture of Prof. Dr. Björn Eskofier at the Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU).

## Awards with Relevance to the Topic of this Thesis

[AW1] Second place at the Werner von Siemens Award 2022 in the category Technology with Purpose. The DSP Blackboard mentioned in subsection 3.8.4 and the new signal flow charts defined in Appendix A were contributors to this success.

[AW2] Price for the best talk in the session of Smart Field Devices at the IoT Conference 2021.

[AW3] With the application at Fraunhofer Venture, ULPSEK was awarded two-day entrepreneurship coaching, including hotel and travel costs, in 2016.

## Talks, Workshops, and Exhibitions with Relevance to the Topic of this Thesis

[TWE1] Talk: "State of the Art: Battery-operated sensor nodes at Siemens" at the Siemens Core Technology meeting for Smart Field Devices (Tobola, 2022)
The contribution was to talk about battery-operated sensor nodes, including the contributions from this thesis.

[TWE2] Talk: "State of the Art: Battery-operated sensor nodes at Siemens" for one of the CEOs at Siemens (Tobola, 2022)
The contribution was to talk about battery-operated sensor nodes, including the contributions from this thesis.

[TWE3] Talk: “Quality of Sensing” at the Network and Communication Community at Siemens (Tobola, 2021)
This talk was about novel methods regarding the signal quality of sensor systems.

[TWE4] Talk: “In-field processing” at the Siemens Core Technology meeting for Smart Field Devices (Tobola, 2021)
The contribution was to talk about efficient processing at smart field devices, focusing on implementation and testing processes.

[TWE5] Talk: “Quality of Sensing” at Siemens IoT Conference (Tobola, 2021)
The contribution here was to talk about signal processing on embedded devices, particularly the capabilities of sensor systems for monitoring their own sensing quality. This talk was honored with the price for the best talk in the session of Smart Field Devices.

[TWE6] Talk: “AI on Sensors Systems” at Siemens IoT Conference (Tobola, 2020)
The contribution was to give a talk presenting the energy computing efficiency of a machine learning task on a microcontroller.

[TWE7] Talk: “RTOS Setup for Ultra-low-power IoT Sensors” at the Siemens Core Technology meeting for Smart Field Devices (Tobola, 2019)
The contribution was to give a talk focusing on the configuration of an RTOS for low-power applications.

[TWE8] Talk: “RTOS Setup for Ultra-low-power IoT Sensors” at Siemens IoT Conference (Tobola, 2019)
The contribution was to give a talk focusing on the configuration of an RTOS for low-power applications.

[TWE9] Hackathon: „Health Hackers Erlangen“ on Wearables in Medizintechnisches Test- und Anwendungszentrum (METEAN)
The contribution was to organize the event and support the teams with electrical components and evaluation boards.
An article has been published in Deutsches Ärzteblatt: IT-Sicherheit: Wenn Hacker helfen (Schulz, Liphardt, Tutein, Tobola, Pickert, Hueber, 2017)

[TWE10] Talk: Körpernahe Sensorik für eine älter werdende Bevölkerung (Tobola, 2016)
The contribution of the author was to prepare and give a talk about the challenges and solutions for low-power wearable biomedical sensor systems at the “Sensorik Technologieforum: Mobilität und Sicherheit im Alter” in Regensburg.

[TWE11] Venture Capital Pitch: Wearable Low-Power Sensor System (Tobola, 2016)
The author applied for a venture program with the miniaturized sensor developed in this thesis for *Fraunhofer Netzwert* at *Fraunhofer Venture*. The application has been awarded two coaching days, including hotel and travel costs. The author was giving a pitch at the Fraunhofer Venture.

[TWE12] Exhibition: ULPSEK at Embedded World (Tobola, 2016)
The author was accepted for a dedicated place at the Fraunhofer booth at the Embedded World conference. Embedded World is the leading international fair for embedded systems. ULPSEK was presented live for all three days at the Embedded World exhibition in Nürnberg.

[TWE13] Talk: Pan-Tompkins-Algorithmus zur QRS-Komplex-Detektion (Tobola, 2015)
The author's contribution was to prepare and give a talk about the Pan-Tompkins algorithm with a particular focus on low-power implementation at the Pecha Kucha in Erlangen.

[TWE14] Workshop: Ultra-Low-Power Sensor Evaluation Kit (Tobola, 2015)
The author's contribution is to prepare and give a talk about low-power design methods for wearable biomedical sensor systems with examples achieved with ULPSEK in Medical Technology Test and Demonstration Center, METEAN, Erlangen.

[TWE15] Talk: Energiebedarf tragbarer Sensoren (Tobola, 2015)
The contribution of the author was to prepare and give a talk about the challenges and solutions for low-power wearable biomedical sensor systems at the Health 2.0/Quantified Self Meetup in Erlangen.

[TWE16] Talk: Ultra-Low-Power Biomedical Signal Processing Design (Tobola, 2015) invited by Dr. phil. Heiko Gaßner
The contribution of the author was to prepare and give a talk about the challenges and solutions for low-power wearable biomedical sensor systems for the Embedded Systems Initiative: MotionGroup at the Institut für Sportwissenschaft und Sport (ISS), Erlangen.

[TWE17] Invited Talk: Energiehaushalt mobiler Sensorsysteme (Tobola, 2015)
Keynote about energy management in mobile sensor systems. Abschlusspräsentation Hauptseminar "Power Management in Mobile Devices" at the chair of Real-Time Computer Systems at the Technical University of Munich (TUM), invited by Prof. Dr. Samarjit Chakraborty.

[TWE18] Talk: Textile Integration medizinischer Sensorsysteme (Tobola, 2013)
The contribution of the author was to prepare and give a talk on the integration of wearable sensor systems at the Wissenschaftsrat of the Arbeitsgemeinschaft Messwert-Aufnehmer (AMA) in Erlangen. The video is available online: https://vimeo.com/76310213, last visited in July 2023.

## Supervised Student Theses related to this Ph.D. Thesis

[SST1] Bachelor Thesis: Automatisiertes Abbilden und Verifizieren der Signalverarbeitung eines eingebettetes Systems zur Kollisionserkennung (Heinzius, 2019)

[SST2] Bachelor Thesis: Implementierung und Evaluierung mobiler, medizinischer Ultra-Low-Power-Sensoren (Streit, 2015)

[SST3] Bachelor Thesis: Implementierung und Evaluierung einer Ultra-Low-Power-Software-Architektur für mobile, medizinische Sensorsysteme (Korpok, 2015)

[SST4] Bachelor Thesis: Implementierung und Evaluierung von Ultra-Low-Power-Sensoren für den mobilen Einsatz (Espig, 2015)

[SST5] Bachelor Thesis: Eigenschaften textiler EKG-Elektroden (Endlich, 2015)

[SST6] Master Thesis: Entwicklung einer textilintegrierten Sensorik zur kontinuierlichen Atemaktivitätserfassung (Schmitz, 2014)