Embedded Recording

Tiny, low-power audio solutions for wireless systems

Rebecka Erntell

LUND UNIVERSITY
Department of Automatic Control

MSc Thesis
TFRT-6024
ISSN 0280-5316

Department of Automatic Control
Lund University
Box 118
SE-221 00 LUND
Sweden

Printed in Sweden by Tryckeriet i E-huset
Lund 2016

Abstract

The aim of this thesis was to explore and evaluate current methods for building tiny, low-power audio recording systems for wireless applications. Short audio clips should be recorded, transferred to a smartphone and played back. Optimization of the different parts should be explored and a prototype for usage in battery powered devices with limited space should be built.

System requirements were defined in terms of audio quality, physical size and energy consumption. Additionally, the price should be kept low and the system should be reliable and stable, without notable delays.

The system was divided into six focus areas; microphones, preamplifiers, signal processing, transmission, software design and energy consumption feedback. Each of those was explored theoretically and by practical prototyping, testing and tuning.

Low energy consumption mainly proved to compromise audio quality, system speed and development time. Signal processing and transmission, especially range, speed and protocol overhead, played major roles.

Size constraints did not affect performance very much, except for reduced battery capacity and microphone positioning options. Decent sound quality could be achieved with small size and low cost and signal processing could do much to keep energy consumption down and the perceived sound quality up.

This thesis was supervised by Mats Iderup and Björn Strandmark at Mikrodust AB, Noroz Akhlagi at Verisure Innovation AB and Martina Maggio at the Department of Automatic Control, Lund University.

My deepest thanks for your help and encouragement!

Contents

1

Introduction

The way technology is used has changed rapidly over the last years. The Internet of Things is starting to become a reality and as the number of connected devices increases, constraints on energy consumption and size of the embedded systems are becoming more and more important.

This thesis focuses on optimization of wireless sound recording systems, which have a wide range of applications. Much research has already been conducted in the field, but as technology development is fast, there is a constant need for trying out new combinations, pushing limits and re-evaluating concepts.

The project was performed in cooperation between the Department of Automatic Control at Lund University, Mikrodust, a developer of embedded radio modules based in Lund, and Verisure Innovation, a home security company in Malmö.

1.1 Problem definition

The aim of the thesis is to explore and evaluate current methods for building tiny, low-power audio recording systems for wireless applications. Short audio clips should be recorded in decent quality, transferred to a smartphone and played back. Optimization of the different constituent parts should be explored and, finally, a prototype for usage in battery powered devices with limited space should be built.

The following questions serve as a base for the analysis. The first set divides the specified system into five general focus areas, each of which can be tuned separately or in relation to the other parts. Some of the components are naturally optimized in terms of energy consumption, while others affect size and audio quality more directly. Energy consumption feedback was added as a sixth area, in order to maintain a holistic system view and utilize resources as efficiently as possible.

The second set of questions reflects the complete system evaluation, in respect of energy consumption, size and audio quality, both theoretically and practically.

1. Components
   What microphone types are suitable based on energy, audio quality, size and price?
   How would an efficient microphone preamplifier be constructed?
   What signal processing possibilities are there with respect to low energy consumption?
   How is transmission adapted for saving energy?
   How can software design help to increase efficiency?
   How can energy consumption feedback be used for optimization?

2. System evaluation
   What are the energy consumption limits for the above system with current technology?
   What are the size limits for the above system with current technology?
   What audio quality can be achieved, set against energy consumption and size?

1.2 Contribution

Hopefully, the thesis will help to provide an overview of current possibilities and limitations in the field, serve as starting point for system design and suggest areas and methods for improvement.

Another aim is to test the hardware limits with a few extreme design choices. The system prototype was implemented on a single general purpose radio chip with low energy consumption. It was perhaps not the easiest way to a final product, yet it was not at all impossible with current technology.

1.3 Report structure

The report consists of seven chapters.

1. *Introduction* serves as an introduction to the thesis and to the report.
2. *Researching embedded audio* describes method and previous research.
3. *Design concepts* provides a theoretical background to the system components.
4. *Prototype design* describes the implementation and tuning of a prototype.
5. *Energy consumption, size and audio quality* analyses the conceivable system performance.
6. *Conclusion* evaluates and concludes the previous analysis.
7. *Summary* is a short summary of the thesis.

2

Researching embedded audio

2.1 State of the art

A wireless recording system touches on several research fields, each developing fast and on many fronts simultaneously. Naturally enough, most research tends to focus on aspects of the different components and optimize microphones, preamplifiers, signal processing and other parts separately, albeit with specific purposes and constraints in mind.

Microphone technology is expanding in areas such as materials, construction and fabrication techniques. The main purposes are to improve audio quality and frequency response for different applications and reduce noise, size and energy consumption. A few new sensor types such as laser and optic fibre microphones are being developed. There are also tendencies towards moving amplification and AD conversion closer to or into the microphone package and towards psychoacoustics and sound quality perception [cf. An Sun and Farrell, 2010; Jeng et al., 2011; Homentcovschi and Miles, 2011; Hillenbrand et al., 2013; Madinei et al., 2013; Kendrick et al., 2015; Walser et al., 2016].

In preamplifier design, achieving high amplification while keeping the signal-to-noise ratio high is still a major challenge. With audio devices made compact, portable and battery powered to a higher extent than earlier, focus has also shifted more towards low and adaptive energy consumption [cf. van der Woerd and Serdijn, 1993; Jawed et al., 2009; Starecki, 2010; Rivers, 2010; Du and Odame, 2012; Tu et al., 2013].

As digital signal processing, wireless information transmission and real-time software is present in so many aspects of everyday life, it is almost impossible to sum up current research in those fields in a few lines. There are multiple examples, though, where more or less the entire signal chain from sensors to actuators are taken into account and integrated, in a similar way as is attempted in this thesis.

One of those areas is the design of hearing and communication aid systems, where sound is recorded, transferred and played back. In one study by Wolfe et al [2015], adaptive gain depending on background noise is explored. Another example is the adaptive communication system for motorcycle helmets, with one main microphone and noise cancellation based on an additional microphone array, developed by Cornelius et al [2005].

Microphone arrays, multiple connected microphones positioned to allow for spatial audio processing, are also currently being explored in a number of ways. They are applied in fields as different as hearing aid [Ayllón et al., 2009], non-destructive impact-echo testing of concrete quality [Groschup and Grosse, 2015] and surveillance systems [Crocco et al., 2016; Zhao et al., 2012].

Systems such as the one presented by Zhao et al point to some important general principles for wireless audio construction. In their design, microphones were spread in an energy efficient sensor network covering a surveillance area. In order to maximize usability, they specifically stress the importance of low and flexible energy consumption, monitored in all system stages as continuous low-power operations may prove much more energy-intense than high-power units kept in sleep for most of the time.

Other researchers, such as Girod and Roch [2006] discuss similar issues for the use of remote sensors in general. Girod and Roch also list constraints and possibilities for remote audio systems and identify power management, noise resilience and user-friendly interfaces as the currently most important areas for improvement.

2.2 Method and design approaches

Method

The first step in this project was to define the focus areas in Chapter 1.1, in order to increase clarity and coherence throughout the project. Each of the system components was explored in a general, theoretical background (Chapter 3). This was used as a base for tests, tuning and evaluation, both on a prototype and in theoretical calculations, in the following two chapters.

As the system touched on several fields with different premises and objectives, the method was varied and adapted accordingly. In order to keep methods, design choices and tools in their context, these are described, motivated and discussed more in detail as they appear in each of the report sections.

Delimitations

Some delimitations were naturally derived from the system requirements (Chapter 3.1) and from the fact that the target would be small and battery-powered. Those mainly affected design choices in terms of power consumption, size and price.

Strictly speaking, no firm delimitations were set for what should or should not be included in the system, as long as the basic recording and audio transfer functionality was provided as intended. Still, the components and solutions were kept simple, as the focus of the thesis was to identify and evaluate current possibilities and not, however tempting, to invent completely new solutions.

Limitations

Concerning the problem definition and method in general terms, the lack of delimitations was the most severe limitation. The entire system was more or less six theses in one, and as time was limited to twenty weeks, it was impossible to go deep into all fields. Hopefully, the report may at least serve as a starting point for further exploration.

The optimization concept can not really be applied without reservation. Even if optimization is objectively possible and made with certain constraints on hardware and performance, the definition of constraints and presumptions may well be more essential and complex than the actual analysis. Again, this could easily be subject for a thesis of its own.

One last general remark is that data security and ethical and environmental factors only have been considered on a very basic level. Results and limitations of the specific analysis methods applied on the different system components will be further discussed in the prototype and analysis chapters.

3

Design concepts

3.1 System overview

System requirements

The system requirements were defined by Mikrodust AB with small smart home applications as primary targets. The typical use case would be a safety system, where sound monitoring is added as an alternative or as a complement.

Sound recording should be triggered either remotely from a smartphone or automatically from other devices connected locally. The sound clips recorded should be transferred to a smartphone, where they could be played back. The system requirements were defined within three domains:

Audio quality: Sounds of magnitudes around 50 dB_{SPL} should be clearly audible and possible to recognize and identify. 50 dB_{SPL} is a fairly quiet level, corresponding to soft conversation at a couple of meters distance, calm background music or a dishwasher running in an adjacent room [USDL, 2016].

Size: The physical space required for the system should be minimized as much as sound quality allows for.

Energy consumption: As the target devices typically would be battery powered, the average current needed for the system should be in the μA range. Peak values around 5 mA could be allowed, provided that the peaks were short.

In addition to the above, the price should be kept as low as possible while still meeting the system requirements. Signal processing and transfer should not be subject to noticeable delays and the system should be reliable and stable.

Hardware platform

The system hardware could be divided into a few main components. Following the signal chain, these would be the *microphone*, a *preamplifier*, a *microcontroller* handling AD conversion, recording and transmission and, finally, a *smartphone* with a receiver app (Fig. 3.1). Audio encoding and any signal processing needed could be done either in the microcontroller or in the smartphone app, depending on what proved most efficient. Before AD conversion, a simple anti-aliasing filter would also be needed to suppress frequencies outside the desired bandwidth.

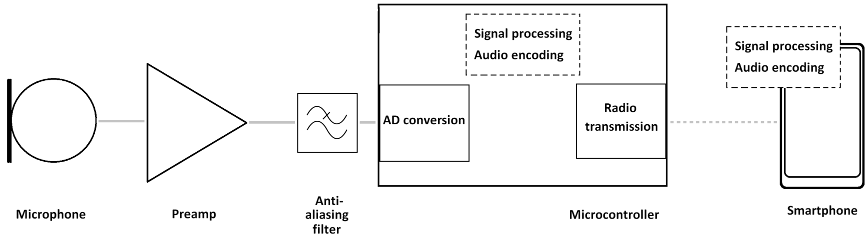

Figure 3.1 An overview of the system.

The only predefined part was the microcontroller platform, a Silicon Labs EFR32MG chip, based on an ARM Cortex-M4 processor, on a Silicon Labs Wireless Starter Kit development board (Fig. 3.2). Several similar solutions would have been possible, but low energy consumption, while still providing large memory and a DSP instruction set, made EFR32MG a suitable choice. A short summary of the specifications can be found in Tables 3.1 and 3.2.

One important point is that the chosen platform was intended for multi-purpose use and not specifically for audio processing. This would facilitate migration and including of recording functionality in already existing systems, with minimal additions in hardware and energy consumption. At the same time, it brought about challenges with respect to audio quality, as for example the integrated AD converter had lower resolution than AD converters normally used in pure audio applications. Additionally, the supported radio protocols were aimed at low-power transfer of occasional, small amounts of data and not really at file transfer.

Processor	ARM Cortex-M4
Clock frequency	40 MHz
Architecture	32 bit Harvard, RISC
Processing efficiency	1.25 Dhrystone MIPS/MHz
Flash program memory	256 kB
RAM memory	32 kB
DSP features	Single cycle 16/32-bit MAC, 8/16-bit SIMD arithmetic

Table 3.1 Specifications for the EFR32MG processor core [ARM, 2015; Silicon Labs, 2016b].

General purpose IO pins	31
Supply voltage	1.85 - 3.8 V
Current consumption	100 μA/MHz in active mode
Sleep current	3.3 μA in deep sleep mode, 1.1 μA in hibernate mode
Radio transceiver	Inverted F antenna, 2.4 GHz, 19.5 dBm Tx power
Supported radio standards	Bluetooth Low Energy, ZigBee, Thread i.a.
AD converter	12 bits successive approximation register (SAR), 1Ms/s
Memory extension	8 Mbit flash memory on SPI connection

Table 3.2 Specifications for the EFR32MG chip used [Silicon Labs, 2016b].

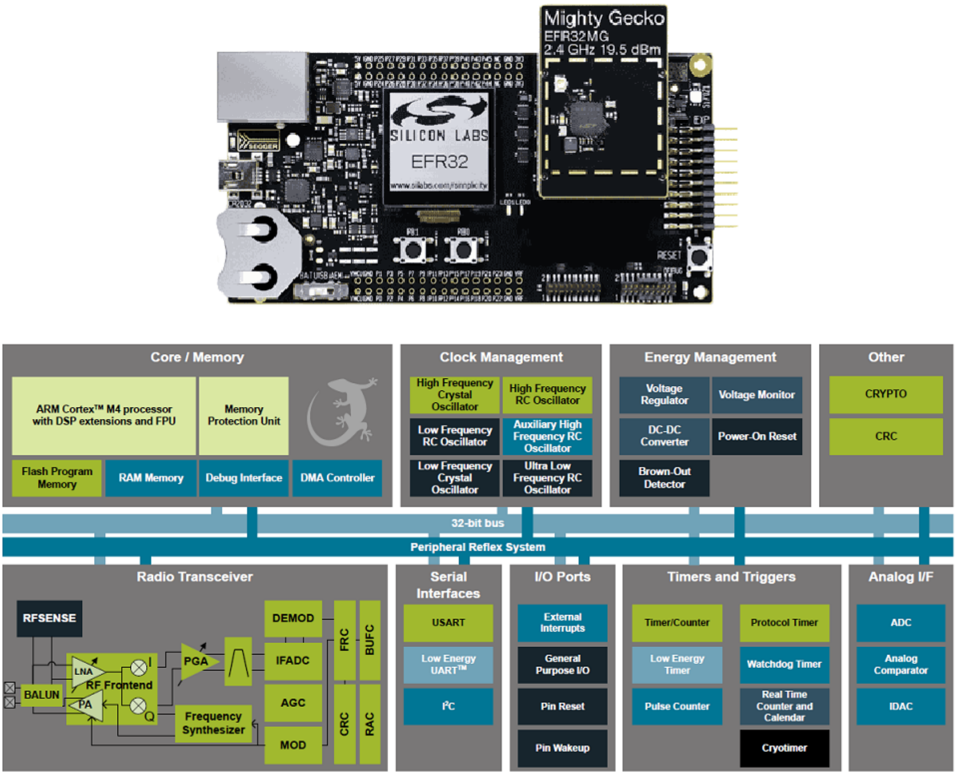

Figure 3.2 The Silicon Labs Wireless Starter Kit, with an EFR32MG card to the upper right, and a block diagram for the EFR32MG chip [Silicon Labs, 2016a; Silicon Labs, 2016b].

System components

The system was broken down into six focus areas, as listed below, which will be explored in the following chapters. These reflect the design step questions from the problem definition and each of them could be optimized both separately and in relation to the other parts.

a. Microphone
b. Preamplifier
c. Signal processing
d. Transmission
e. Software
f. Energy consumption feedback

3.2 Microphone

Microphone properties

Microphone sensitivity measures output voltage as a function of acoustic pressure, commonly in mV/Pa at 1 kHz. 2-8 mV/Pa is enough for close-up recording, while larger distances require 10-50 mV/Pa. Sensitivity increases with membrane diameter and supply voltage, but also varies with frequency [Rayburn, 2012; Czarny, 2015; Ballou et al., 2005].

The signal-to-noise ratio is normally defined as the ratio between a 1 kHz, 1 Pa (94 dB_{SPL}) signal and the noise floor, which sets the lower sensitivity limit for a microphone. The upper limit for the dynamic range is set by the acoustic overload point, where the response becomes nonlinear [Czarny, 2015].

The bandwidth needs to cover most, or chosen parts, of the audible frequency band of 20 Hz to 20 kHz. The lower limit is mainly determined by the microphone encapsulation and the higher by the maximum vibration speed of the membrane. Ideally, the frequency response should be flat for the entire range [Czarny, 2015].

Omni-directional microphones are equally sensitive for all sound directions, with one side of the diaphragm shielded so that sound waves only affect the other side. Bi-directional microphones have eight-shaped sensitivity patterns, created by leaving both sides of the diaphragm open so that only changes in pressure between the sides are detected. By combining omni- and bi-directional types, several other patterns can be formed [Ballou et al., 2005; Czarny, 2015].

Modern microphones often have output impedances from 50 to 2000 Ω. The impedance may change with frequency, but as preamplifier input impedances are normally much higher, impedance matching is seldom a problem [Rayburn, 2012].

Finally, the microphone should be placed close to the preamplifier and away from interference sources in order to minimze noise. The acoustic port hole in the device cover should be short, wide, even, non-leaking and covered with a mesh to avoid echoes, noise and altered frequency responses [Knowles, 2011].

Microphone types

A microphone normally consists of a membrane vibrating with sound pressure waves and a transduction mechanism that generates a current or a voltage proportional to the vibrations (Fig. 3.3). The transduction mechanism may be based on electromagnetic induction, on capacitive or resistive changes or on the piezoelectric effect.

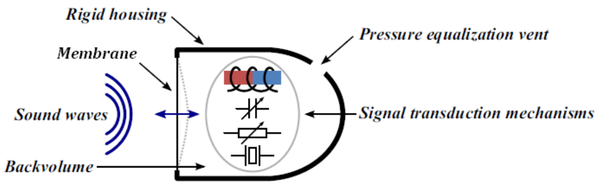

Figure 3.3 General structure for a microphone [Czarny, 2015].

Dynamic microphones have their membranes connected to a coil which moves in a magnet field and generates a current proportional to the sound waves (Fig. 3.4). They are robust and tolerate high sound pressure levels, but are mainly used in close-up recording due to low sensitivity. Ribbon microphones, with a metal ribbon instead of the coil, are more fragile but otherwise have similar properties [Czarny, 2015].

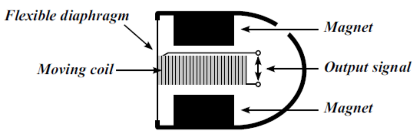

Figure 3.4 Concept sketch for a dynamic microphone [Czarny, 2015].

Condenser microphones use the membrane together with a fixed backplate as a capacitor (Fig. 3.5). Vibrations in the membrane alter the plate distance and thus varies the capacitance. If phantom power, a bias voltage over the plates, is applied, a charge proportional to the sound waves is induced. Condenser microphones have high sensitivity, even frequency responses and large bandwidths and work well for quality and distance recording. For high frequencies, the frequency response may be uneven. At low sound levels, noise can be a problem and at high levels, distortion will occur.

There are also *electret microphones*, in which the phantom power has been replaced by permanently charged membrane materials. These perform almost as well as the phantom powered microphones, but are cheaper and very common in computer and phone applications. Sensitivity is high and bandwidths range from 10 Hz to 30 kHz. The membrane can be made small, only a couple of millimeters in diameter. For condenser type microphones, sensitivity decreases and the low frequency cut-off increases with size as the plate gap gets smaller and capacitance decreases. Signal-to-noise performance deteriorates, but distortion decreases and the lower membrane mass allows for faster vibrations and higher top cut-off frequencies [Shin et al., 2015; Czarny, 2015; OML, 2016].

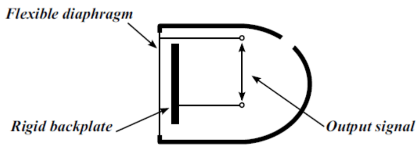

Figure 3.5 Concept sketch for a condenser type microphone [Czarny, 2015].

Piezo microphones have membranes made of or coupled to piezo-electric or piezo-resistive materials. These materials generate a voltage or change resistance proportionally to mechanical stress (Fig. 3.6). Normally they have a certain resonance frequency, but are more or less linear below that frequency. They are relatively cheap, generate high voltage outputs already at low sound pressure levels and are mainly used as contact microphones. Polymer materials are durable and have flat frequency responses and low resonance frequencies. Ceramic materials are cheaper, better shielded, lower in noise, higher in output level and have higher resonance frequencies. Piezo output impedance is high and for efficient transfer, careful impedance matching is required. Good shielding is then essential, in order not to pick up low frequency interference. Also, as large resistances are high in intrinsic noise, the microphone sensitivity is limited by the noise floor [Czarny, 2015; OML, 2016].

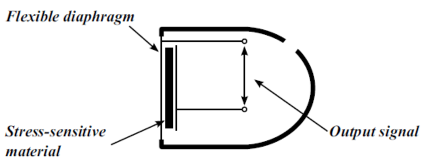

Figure 3.6 Concept sketch for a piezo type microphone [Czarny, 2015].

Micro-electro-mechanical microphones have been commercially available for some fifteen years. The basic construction is similar to the larger microphone types, with a vibrating membrane and a transducer on a silicon wafer. Capacitive transduction is most common and generally reaches higher sensitivity, but piezo-resistive, piezo-electric and optic solutions are also possible [Czarny, 2015].

As already mentioned, sensitivity decreases and low frequencies are cut off when condenser type membranes are made smaller. There are ways to compensate for these effects, for example with bias or preamplifier design, and MEMS microphones can be made with high sound quality. Current models reach signal-to-noise ratios around 65 dBA and sensitivity around -40 dB. In high humidity or high temperatures, they are more reliable than conventional electret microphones. Normally, they are equipped with an integrated preamplifier or a digital readout circuit (Fig. 3.7). Digital output results in larger packages, but minimize the analog signal chain and thus reduce disturbances [Shin et al., 2015; Walser et al., 2016].

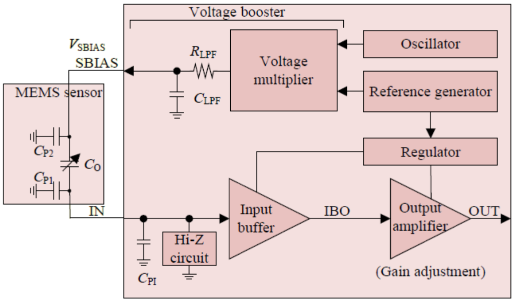

Figure 3.7 A readout circuit for a MEMS microphone [Kim et al., 2016].

In simple terms, both condenser and piezo microphones can be modelled as a voltage source in series with a capacitance (Fig. 3.8).

There are also more specialized microphones. *Laser microphones*, for example, measure variations in reflection angle due to surface vibrations and can be aimed at distant targets. *Fiber optic microphones* sense light intensity changes on a reflective membrane. They are robust and insensitive to electromagnetic interference.

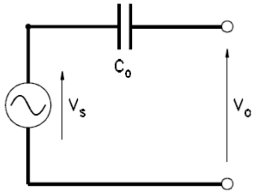

Figure 3.8 An equivalent circuit for a condenser-type microphone [MSI, 1999].

3.3 Preamplifier

Preamplifier properties

The preamplifier is the first stage after the microphone. It is normally a voltage amplifier with the primary function of extracting the weak signal from the sensor and lifting it to line level while adding as little noise as possible. It should be placed close to the microphone in order to minimize noise and interference.

A few basic requirements can be defined for microphone preamplifiers. First of all, the variation span for microphone output voltage is large, as microphone sensitivity as well as source distance and sound pressure vary widely. Input levels are normally low, between -70 dBV and -40 dBV or 0.9-28 mV_{PP}. The desired amplified output is often either a predefined line level, for example -10 dBV, 0 dBV, 0 dBu or 4 dBu (0.9-3.5 V_{PP}), or a suitable level for AD conversion which is normally -2 to 5 dBV (2.2-5 V_{PP}). This calls for a gain between 30 and 65 dB. Typically, the gain would need to be adjustable and highly linear over the audible frequency span, 20-20000 Hz, in order not to distort the sound [Hebert, 2010].

To prevent distortion, the response speed, or slew rate, of the preamplifier needs to be high enough. The top value theoretically needed can be calculated as the product of the double top frequency and the peak-to-peak output voltage. In many applications, energy efficiency is important. The amplifier also needs to be resistant to radio frequency interference and common-mode noise and, crucially when handling weak signals, be very low in noise generation [Hood, 1997; Invensense, 2013].

A complete preamplifier system may contain additional components such as microphone phantom powering, polarity inversion, input attenuation, radio interference protection and input and output impedance switches (Fig. 3.9). In this chapter, though, focus will be on the main gain stage.

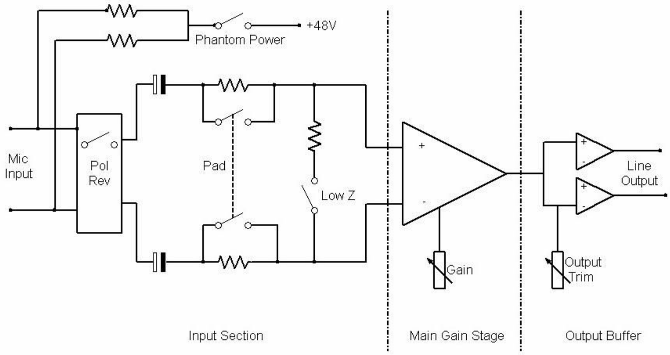

Figure 3.9 A complete microphone preamplifier system [Rivers, 2010].

Amplifier types

The actual amplification can be achieved by vacuum tubes, magnetic saturable reactors or transistors. Vacuum tube technology is more expensive and often used in high-end audio applications when the slight harmonic distortion caused by the tubes is desired. Magnetic amplifiers are sturdy and high in current capacity but low in gain and efficiency and transistor technology is preferred in compact, cheap and durable designs [Hebert, 2010].

Operational or op amplifiers, transistor based amplifiers with very high input impedance and gain and low output impedance, is a common choice for audio preamplification. The op amplifier is normally connected in different feedback configurations in order to increase stability, linearity, robustness and bandwidth, to the cost of lower gain. Feedback can also be used as a way to change input and output impedance and thus improve impedance matching [Coates, 2016].

The two basic feedback connections possible result in inverting or non-inverting amplifiers (Fig. 3.10). If the microphone output is balanced, with the signal applied between two hot terminals instead of between one hot terminal and ground, differential configurations are suitable. For higher input impedance and more exact gain performance, instrumentation amplifiers can be used [Rivers, 2010; Coates, 2016].

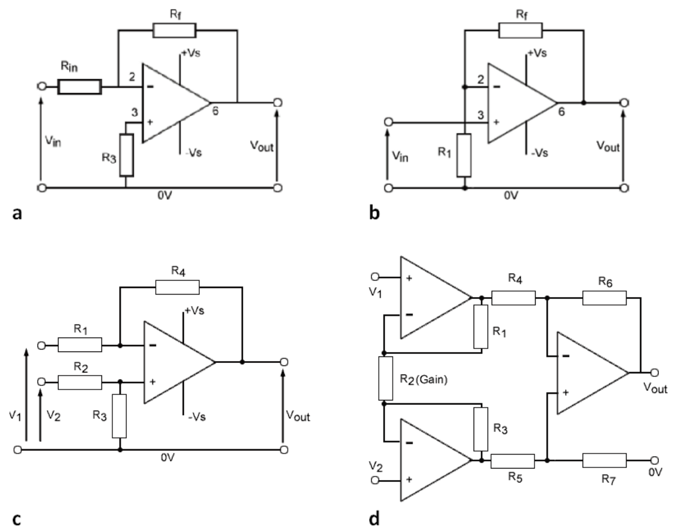

Figure 3.10 Op amplifier circuits. a: Inverting, b: non-inverting, c: differential, d: instrumentation [Coates, 2016].

To drive a large current, which is normally not needed in microphone preamplifiers, a power stage can be added. These are commonly divided into operation classes A, AB, B and C depending on the amount of time that current flows in their output circuits (Fig. 3.11). If sound quality is important and power is low enough to avoid heat dissipation problems, class A is by far most common. In battery powered devices, though, other classes may be a better choice [Rivers, 2010].

Class A output stages conduct constantly, resulting in linear but inefficient systems. Class B only conducts 50 % of the time and reaches efficiencies around 65 %, but introduces crossover distortion when the transistors switch states. In class AB, both transistors are biased to conduct for low signals, combining B efficiency with A sound quality. Class C amplifiers drive a resonant load and are nonlinear but highly efficient in the resonance frequencies [Mellor, 2006].

There are also class D amplifiers, switching between completely open or completely closed and outputting a pulse-width modulated signal. Although the switch frequency may give rise to interference problems, class D is highly efficient and has frequency responses similar to class AB. Amplifiers of classes E and F are mainly used in radio applications. In classes G and H, the voltage supply is adapted to the input amplitude so that the optimum AB efficiency may be reached for all signal levels [Mellor, 2006].

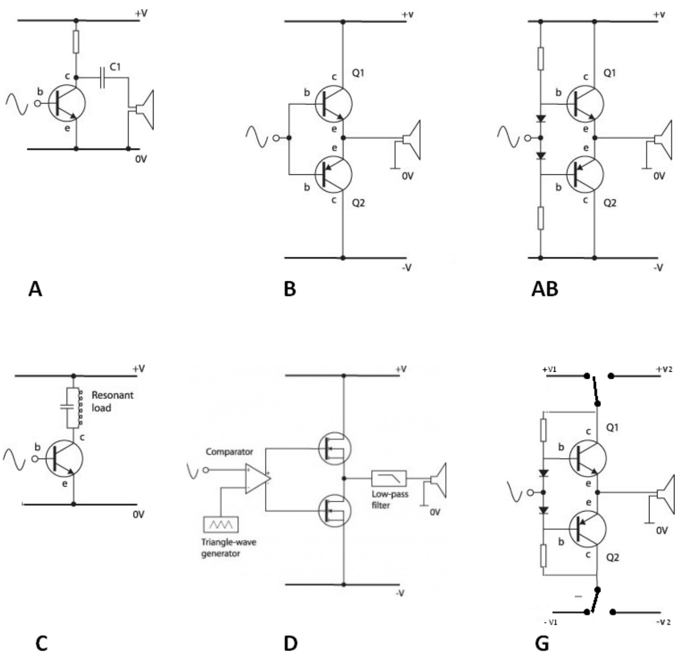

Figure 3.11 Amplifiers of operation classes A, AB, B, D and G with bias circuitry omitted [Mellor, 2006, adapted].

Amplifier coupling

The amplifier input and output impedance needs to be well matched to the microphone and the further stages. Normally, when transferring voltage signals between two cascaded systems, output impedance should be low compared to the input impedance for optimal transmission.

For condenser-type microphones, such as electrets and piezo models with output resistances from 0.2 to 2 kΩ, a 1:10 ratio is normally enough. When working with condenser microphones, though, their output capacitance will create a high-pass filter with the preamplifier input impedance. In order not to lose low frequencies, the amplifier input impedance may need to be much higher. In- and output impedance also vary with gain and frequency, which has to be taken into account in the design [Coates, 2016; Jones, 2010].

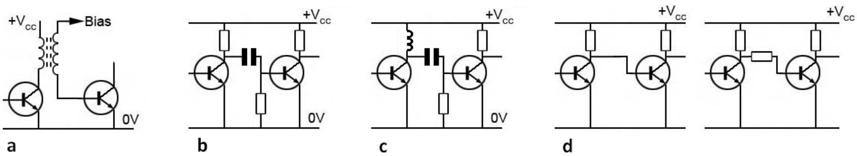

Figure 3.12 Different possibilities for coupling amplifier stages. a: Transformer coupling, b: RC coupling, c: inductive coupling, d: direct coupling [Coates, 2016, adapted].

If multiple amplification stages are needed, these can be coupled in different ways (Fig. 3.12). In transformer-coupled amplifiers, transformers are used to increase the impedance between separate amplifier stages in order to get good impedance matching and thus higher voltage and power gain. RC-coupled amplifiers, where the signal is transferred via a capacitor, are less efficient but have lower prices, less hum and very even frequency responses which make them popular in audio applications. Inductive coupling, introducing an inductor in the RC coupling, increases efficiency but mainly for high frequencies and not so much for the audible register.

For very low frequencies, coupling components have to be large and in some cases they are omitted or replaced with only resistors. This yields direct-coupled amplifiers, which are small, simple and cheap and have even frequency responses, but also temperature sensitive, inefficient and not well suited to high frequencies [Hebert, 2010; Coates, 2016].

3.4 Signal processing

AD conversion

For simplicity, signal processing will be applied as a collective term covering everything that happens to the audio signal from the preamplifier to a file ready for playback. The first step after preamplifier is the analog-to-digital conversion which translates the analog signal to digital samples. A few conversion variables can be set depending on the demands on speed, energy consumption, size, scalability and accuracy.

The two most important digization variables is the bit depth and the sample rate (Fig. 3.13). Multiplied, those yield the bit rate, which should be kept as low as quality allows for in order to save memory and power.

The *bit depth*, together with linearity, jitter, noise floor and signal level matching, decides the dynamic range for the signal. For example, 24 bits per sample (16 777 216 levels) corresponds to a ratio of 144 dB between the lowest and the highest level that could be represented. The dynamic range of the ear is roughly 140 dB, from the hearing threshold defined as 0 dB_{SPL} to the pain threshold at 120-140 dB_{SPL}. Lower bit depth will yield a relatively higher noise floor. Some digital signal processing algorithms require higher resolution. There are also techniques such as dithering or oversampling and averaging, where white noise addition or increased sample speed compensate for lower bit depth [Liu et al., 2015; Harpe et al., 2010].

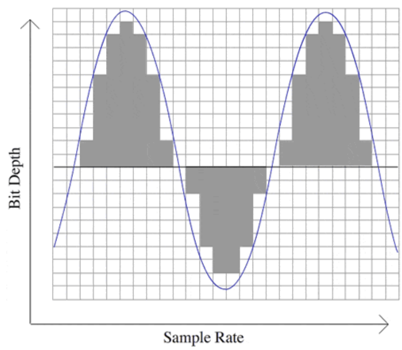

Figure 3.13 Sample rate and bit depth for a sampled wave [Wikiaudio, 2016].

The *sample rate* is decided by the desired bandwidth. The Nyquist sampling theorem gives that the minimum sample speed is twice the highest frequency recorded. This presupposes that the signal can be well approximated by a sum of sinusoidal signals and that sinc interpolation is applied when the signal is reconstructed, though. If supply power and memory size allows for it, a sample rate at least four times the maximum frequency is normally a good idea. An anti-aliasing filter is also needed to suppress frequencies above half the sample rate. For reference, two typical bit rate standards are 8 bits times 8 kHz for land line telephone and 16 bits times 44.1 kHz for CD quality [Liu et al., 2015].

Digital signal processing

For this particular system, any algorithms applied should require little processing power and work on low resolution audio. A first priority could be noise reduction, which is common in most systems and can be made very simple, with a low-pass filter or an amplitude threshold in the frequency domain. If high linearity is important, measuring and compensating for system frequency responses is also good choice.

Automatic gain control would be another option, where the signal energy is monitored and the gain is adjusted so that parts with low sound levels or, alternatively, psychoacoustic importance, get higher amplification. Apart from evening out the perceived sound pressure levels, this reduces static noise, protects from excessive output amplitudes and saves power [Chiu, 2012].

Audio encoding

Either before or after the transmission, the audio samples need to be encoded in a digital audio format. If transmission proves more energy efficient than signal processing, the raw samples should be transferred and otherwise the samples should be encoded in a compressed format before transmission.

The most basic way of encoding audio are the wav and aiff formats. In these, a header is just added to the raw samples, containing information about format, sample rate, bit depth and number of channels.

Lossless compression formats such as flac and alac reduce the file size around 50 % (Table 3.3). Lossy compression, for example mp3 and aac, produce much smaller files. By applying psychoacoustics, compression can be adapted to the characteristics of the human ear so that the perceived quality is affected as little as possible. General compression algorithms, based on correlation between close samples in the time or the frequency domains, could also be applied, as well as higly specialized codecs for example for speech. To minimize time and energy consumption, codecs are often hardware accelerated [Liu et al., 2015; Waggoner, 2010].

Encoder	Application	Bit rate, min. / kbit/s	Bit rate, typ. / kbit/s
Wav	Uncompressed audio	64	706
Flac	Lossless audio compr	24	200
Mp3	Psychoac audio compr	24	96
Opus	Streamed speech compr	6	48
Speex	Streamed speech compr	2	44
ADPCM	General data compr	44	88

Table 3.3 Rough mono channel bit rates for some common encoders. Minimum values are valid for limited bandwidths, for example speech.

3.5 Transmission

Transmission properties

Wireless transmission is most commonly implemented with radio technology. The signal is then transferred as amplitude, frequency or phase modulations on an electromagnetic carrier wave with a frequency between 3 kHz and 300 GHz. Antennas with a resonator tuned for the carrier wave frequency are used as transmitters and receivers, translating between current and electromagnetic waves.

Transmission speed depends on factors such as frequency, digital encoding, interference and collision handling, security, overhead from data packeting, connection establishment and error control [Silicon Labs, 2016d].

Radio energy consumption is determined by a number of factors. Transceivers can be duty cycled and turned off when not needed and transmission windows can be timed in order to maximize sleep time. Protocols can be made more efficient, transmission power can be decreased to just enough for reaching the receiver and the data amount and the signal processing needed can be minimized. There are also more extreme ways to save energy, for example by completely omitting packet overhead or by using pulse frequency modulation [Decuir, 2014; Crepaldi et al., 2015; Ekström et al., 2012].

Transmission protocols

At the moment, there are quite a few different energy efficient, short range radio protocols in use in Internet of Things applications. Five of the most commonly supported ones are summarized in Table 3.4 below.

Wi-Fi is perhaps the most well-known standard and provides long range and high speed but consumes much energy. Bluetooth Low Energy is designed for low cost, low energy and small amounts of data with power consumption around 10-500 mW. ZigBee and Thread are simpler, robust and in average two to three times lower in energy consumption than Bluetooth Smart but often have lower data rates. Z-Wave is somewhat slower, designed for reliability and low latency. There are also sub-GHz wireless standards with even lower throughput but with wave propagation at kilometer range [Silicon Labs, 2016c; Silicon Labs, 2016e; Decuir, 2014].

PROTOCOL	RANGE / m	DATA RATE / kbit/s	FREQUENCY / MHz	TOPOLOGY
Wi-Fi	60-300	$\leq$ 600 000	$\geq$ 2400	Star
Bluetooth Low En.	10-50	10-170	2400	Scatternet
ZigBee	10-100	10-120	$\leq$ 2400	Mesh
Thread	10-100	10-120	$\leq$ 2400	Mesh
Z-Wave	30	$\leq$ 10-40	868	Mesh

Table 3.4 Rough current specifications for some common radio protocols [Silicon Labs, 2016c; Silicon Labs, 2016e; Decuir, 2014].

3.6 Software

Software energy optimization

Software energy consumption can be decreased in a number of ways. Starting with *program code*, algorithms with hardware-friendly operations, few iterations and less data movement are preferred. Libraries, function calls, cache and other memory usage should be well-motivated and data should be kept with just enough accuracy, in efficient representations and storage structures.

Interrupts are preferred to polling, I/O requests can be buffered or batched and threads and resource sharing should be implemented as efficiently as possible. Of course, compromises have to be made with design patterns, development costs, flexibility and code reuse, security and performance [Zhurikhin, 2011; Huang and Ghiasi, 2007; Schulte et al., 2014; Saxe, 2010; Naik and Wei, 2001].

Optimizing compilers are used for instruction and operand ordering, loop optimization, putting hardware in sleep modes, optimization for specific platforms, improved register usage and compaction of stored data. For complex systems, cooperation and resource usage between threads and code units can often be refined. Depending on the situation, it may be relevant to focus either on power and peaks or on the overall energy usage [Schulte et al., 2014; Kandemir et al., 2002; Zendra, 2006].

Peripherals, such as sensors and data transfer units, require duty-cycling, as well as efficient operation order, routing and packaging. Data redundancy should be avoided and hardware should be kept in sleep mode as much as possible, interrupt-driven with either periodic listening or a low-power, separate wake-up channel.

Different ways of omitting processor usage should also be mentioned. The prototype microcontroller in this study, for example, features a peripheral reflex system bus and direct memory access, allowing peripherals to interact in simple, automated tasks without involving the processor [Dutta and Culler, 2005; Silicon Labs, 2016b].

Embedded systems, finally, provide a few more variables. Unused I/O pins, RAM blocks and peripherals should be disabled. As power consumption depends linearly on the clock frequency and quadratically on the supply voltage, keeping those just above what is necessary can make a large difference. This can be integrated in the code, statically or adaptively, made by the compiler or added post compilation. To save processing power, hardware acceleration is also an option [Silicon Labs, 2013].

Software energy analysis

The black box approach to energy analysis would be to run the system under the intended conditions and measure the actual energy consumption with high precision measuring instruments. With a white box approach, somewhat more fine-grained, measurement points could be implemented in different parts of the software.

If the average energy consumption per clock cycle is known, one basic way is to identify instruction blocks produced by the compiler in question, insert timing statements, run several times for various input and calculate the average energy. Theoretical calculations could be useful, identifying possible execution paths and calculating complexity and worst case, best case and average scenarios.

There is also energy and performance analysis software, used with or without the addition of a power meter. Depending on the purpose, simulation and estimation tools could be applied on gate, architecture, bus, instruction, code block or application level [Johann et al., 2012; Roy and Johnson, 1997; Naik and Wei, 2001].

3.7 Energy consumption feedback

Power monitoring and feedback could be implemented either on the entire system or in any of the subcomponents (Fig. 4.16). Starting with the microphone and the preamplifier, dynamic biasing adapted to the input amplitude or bandwidth would help minimizing the energy loss [cf. Du and Odame, 2012].

Sampling, signal processing and transmission may be natural to tune, provided that input and output data, power and performance, can be collected efficiently. Then the software itself can be feedback controlled, so that systems normally dimensioned for worst case scenarios run with less resources [Maggio et al., 2013].

For embedded systems, as already mentioned, clock frequency and supply voltage can be made variable. Dynamic voltage and frequency scaling is one scheme, throttling frequency and voltage whenever full power is not needed. Adaptive body biasing, where the transistor threshold voltage is adjusted, is another. If real-time monitoring is not an option, predefined schemes can still make a difference.

With these methods, efficient mode-switches are important, with fair energy division between tasks and delays short enough not to interfere with real-time requirements. Compromises have to be made with reliability, robustness, availability, flexibility, security and performance [Huang and Ghiasi, 2007; Schulte et al., 2014].

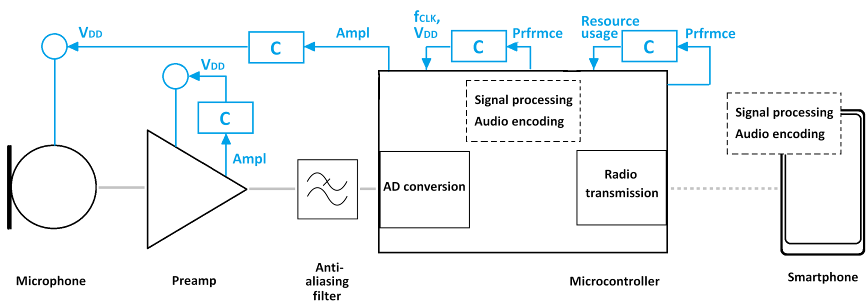

Figure 3.14 Some possible energy feedback loops sketched in the system.

4

Prototype design

4.1 System overview

In order to try out the design concepts in practice, a prototype was built and tuned. The system is depicted in Figures 4.1 and 4.2. Construction, testing, tuning and results are described in detail in the following chapters.

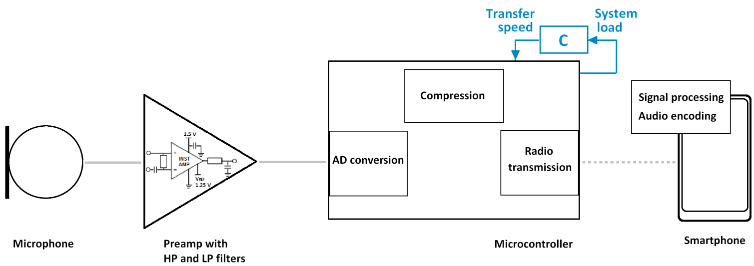

Figure 4.1 Prototype system structure.

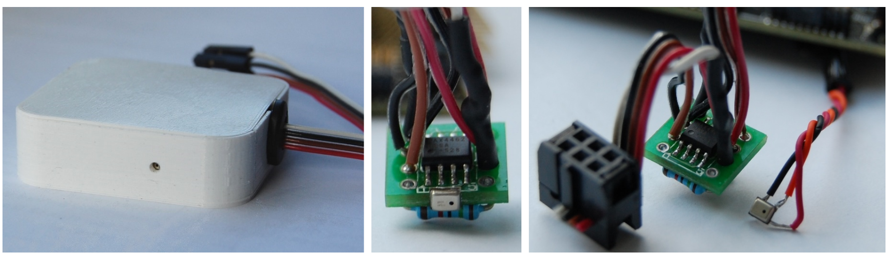

Figure 4.2 The prototype in a printed case. One version was provided with a connector for switching microphones, which were put on input configuration adapters.

4.2 Microphone

Microphone tests

As building a microphone from scratch would typically not be defensible in terms of development cost versus quality, the different microphone options available for purchase were reviewed (Chapter 3.2). For the size, price and sensitivity in question (Chapter 3.1) two main types would be relevant to compare, *electrets* or electret-based *MEMS* microphones. *Piezo elements* were added as the third option in the same price class, although not likely to produce sufficient audio quality at a distance (Fig. 4.3).

Five of the smallest electrets found were selected for evaluation, with different diameters and price. Brands were chosen mainly based on availability. Current consumption, directionality and design did not vary much within the category.

In the MEMS class, four RF shielded models with different current rating from the largest manufacturer were selected. Most were in the same price range. In order to keep preamp and AD conversion parameters separate and changeable, no digital output packages were included, but one model did have an integrated amplifier.

Three piezo elements with different diameters and resonance frequencies were selected. As this is a simple type, other variables did not very much between models.

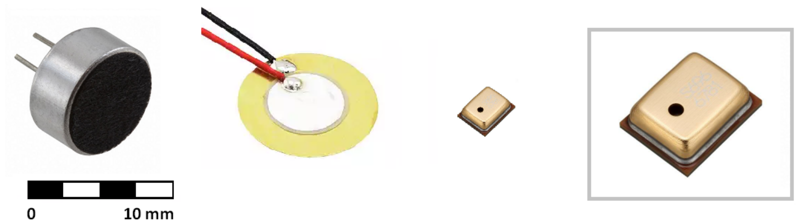

Figure 4.3 Three of the microphones evaluated: Electret (PUI, 10 mm), piezo element (PUI, 10 mm) and MEMS (Knowles, 220 μA) [PUI Audio, 2016; Digikey, 2016; Knowles, 2012].

The test method was adapted from Rayburn [2012] and Nedzelnitsky [1995], based on the IEC standards [IEC, 2010]. The following properties were measured.

1. *Directionality.* Output voltage for different sound source directions.
2. *Linearity.* Output voltage for different frequencies and sound pressure levels.
3. *Sensitivity.* Output voltage at 1 kHz, 94 dB_{SPL}, 1 m.
4. *Power rating.* Output impedance and current at the typical supply voltage.
5. *Noise.* Signal-to-noise ratio.
6. *SPL limits.* Sound pressure levels recordable just above the noise floor and below non-linearity or clipping.

Measurements were carried out in an acoustically isolated room, where sine waves with different frequencies and sound pressure level were played from one loudspeaker. When not varied, the standardized values of 1 kHz, 94 dB_{SPL} at 1 m distance were maintained. A sound level meter was used to keep the sound pressure level well-defined. Two specimens of each model were tested and averaged, but the variation within models proved small.

All microphones were connected to basic, non-amplifying interface circuits according to the recommendations in their respective datasheets and then to an oscilloscope. Signal-to-noise performance was rated as high, mid and low, as it was hard to measure in high precision. For the same reason, minimum SPL levels were not recorded. Results are presented in Figure 4.4.

Model	Size	Price	*Bandw	Direct	Lin, freq	Lin, SPL	Sens	$^*V_{CC}$	Z_{out}	$^*I_{max}$	$I_{typical}$	*S/N	S/N	Max SPL
	mm	SEK	kHz		H, M, L	H, M, L	dBV/Pa	V	kΩ	μA	μA	dB	H, M, L	dB_{SPL}
Electret:														
Kingstate	4	8.90	0.1-20	omni	M	H	-42	2	2.20	500	232	58	M	90
Multicomp	4	11.40	0.1-16	omni	M	M	-40	2	2.20	500	215	60	H	90
Multicomp	6	5.90	0.1-16	omni	M	H	-40	2	2.20	500	204	60	H	90
Kingstate	6	8.20	0.1-20	omni	H	H	-40	2	2.20	500	217	56	H	90
PUI	10	7.50	0.1-20	omni	H	H	-44	3	2.20	500	250	60	H	100
MEMS:														
Knowles	3	14.00	0.1-10	omni	H	H	-38	2-3	0.40	60	50	63	H	90
Knowles	3	8.60	0.1-10	omni	H	H	-42	2-3	0.45	110	79	59	H	90
Knowles	4	15.40	0.1-10	omni	M	H	-38	2-3	0.40	160	115	63	H	90
Knowles	4	15.70	0.1-10	omni	H	H	-38	2-3	0.40	220	180	59	H	90
Piezo:														
PUI	10	14.00	< 7.1	uni	L	L	-54	< 15	≈ 12.0	-2	-0.1		L	100
PUI	12	13.60	< 9.0	uni	L	L	-56	< 30	≈ 12.0	-2	-0.1		L	100
PUI	15	9.70	< 3.6	uni	L	L	-52	< 30	≈ 6.00	-2	-0.1		L	100

Figure 4.4 Microphone test results. Starred columns are datasheet values.

Microphone selection

As a first step, the most suitable microphone of each type was selected. These three were then tested in the complete system context.

The 6 mm Kingstate electret was chosen for its low price and even frequency response up to 10 kHz (Fig. 4.5). Its size was also the second smallest tested. Among the piezo elements, the 12 mm element was selected in order to maximize the resonance frequency and thus the linear bandwidth.

The MEMS microphones were rather similar. Current consumption was small compared to the preamplifier current and would not make much difference in the entire system. The 110 μA model at half the price was chosen, as it was only slightly inferior. The 220 μA amplified model infringed the specification somewhat but would have been superior if it could eliminate the preamplifier completely. However, its gain of 20 dB was somewhat low.

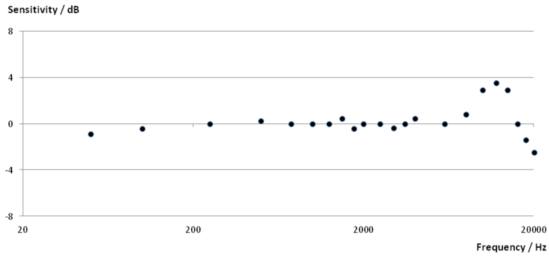

Figure 4.5 The measured frequency reponse for the 6 mm Kingston electret microphone, given as a point of reference. Most of the MEMS models and one other electret preformed similarly, while the piezo element responses were more uneven.

In the prototype, the electret and the MEMS microphones performed comparably, although the latter seemed as the best option based on size, current consumption and impedance. The piezo element was tested both as a single element and in a differential configuration, with two elements connected back to back and sharing ground. It worked well as a contact microphone, but was almost completely insensitive to airborne sound.

4.3 Preamplifier

Preamplifier design

Two different basic op amplifier circuits were built as a starting point; one inverting and one instrumentation amplifier. The inverting amplifier was changed to non-inverting with the piezo element, which required high input impedance (Fig. 4.6).

The amplifiers were made for a voltage gain of 36 dB, which was enough to lift microphone signals between 0.2 and 28 mV_{PP}, the -40 dBV/Pa output at 50 to 96 dB_{SPL}, to a 0-2.5 V AD converter input. The bandwidth was set to 100-8000 Hz.

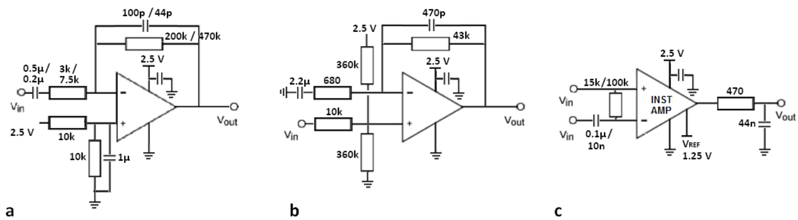

Figure 4.6 Prototype preamplifiers. a: Inverting (low/mid input impedance), b: non-inverting, c: instrumentation (mid/high input impedance).

The active element in the inverting amplifier was a 7 SEK Texas Instruments low-noise op amplifier with high slew rate and 0.1-10 kHz linearity. It was specified for 2.7-12 V but could run on lower supply voltage. The instrumentation amplifier was based on a pre-made 19 SEK low-power, 40 dB Maxim amplifier chip, also high in linearity, slew rate and signal-to-noise ratio and specified down to 2.85 V.

Preamplifier tests

The amplifiers were tested in a similar way as the microphones, as described below. The default input was a 20 mV_{PP}, 1 kHz sine wave from a function generator with 50 Ω output impedance.

1. *Linearity.* Voltage gain for different frequencies and amplitudes.
2. *Step response.* System response on a 0 V to 20 mV step input.
3. *Power rating.* Impedance and current at 2.5 V supply voltage.
4. *Noise.* Signal-to-noise ratio.
5. *Microphone compatibility.* Input test with the three selected microphones, with sine waves at 1 kHz, 50, 70 and 94 dB_{SPL}, 1 m.

Results are presented in Figures 4.7 and 4.8. The inverting and the instrumentation amplifiers proved good enough not to need any additions. There was no notable quality difference with the pre-made chip. The non-inverting amplifier was susceptible to noise and hum due to large resistors and input impedance and would require a more complex design.

When testing microphone compatibility, the microphones generally worked well with lower input impedance than expected, which saved noise and heat loss from large resistors. Energy consumption, though, was high compared to what could be achieved [e.g., van der Woerd and Serdijn, 1993]. As time was limited, this was finally left for future improvement.

The instrumentation amplifier was selected for the system prototype. The inverting amplifier would have worked just as well, though, depending on the constraints on size, complexity or price. After defining the signal processing properties (Chapter 3.4), the amplifier bandwidth was reduced to 0.11-3.85 kHz by changing the output capacitor to 88 nF.

Amplifier	Bandw kHz	Lin, freq H, M, L	Lin, amp H, M, L	Z_{in} kΩ	Z_{out} Ω	$I_{typical}$ µA	I_{BIAS} µA	S/N dB	Mic compatibility
Inverting, low Z_{in}	0.11-7.95	H	H	3	≈ 0	850	125	30.4	(Electret,) mems
Inverting, mid Z_{in}	0.11-7.70	H	H	7.5	≈ 0	856	125	30.4	Electret, mems
Non-inverting	0.11-7.85	H	H	180	≈ 0	845	80	23.5	Elec, mems, piez, diff piez
Instrumentation	0.11-7-70	H	H	15; 100	≈ 0	735	0.5	33.3	Elec, mems, piez, diff piez

Figure 4.7 Results from the preamplifier tests.

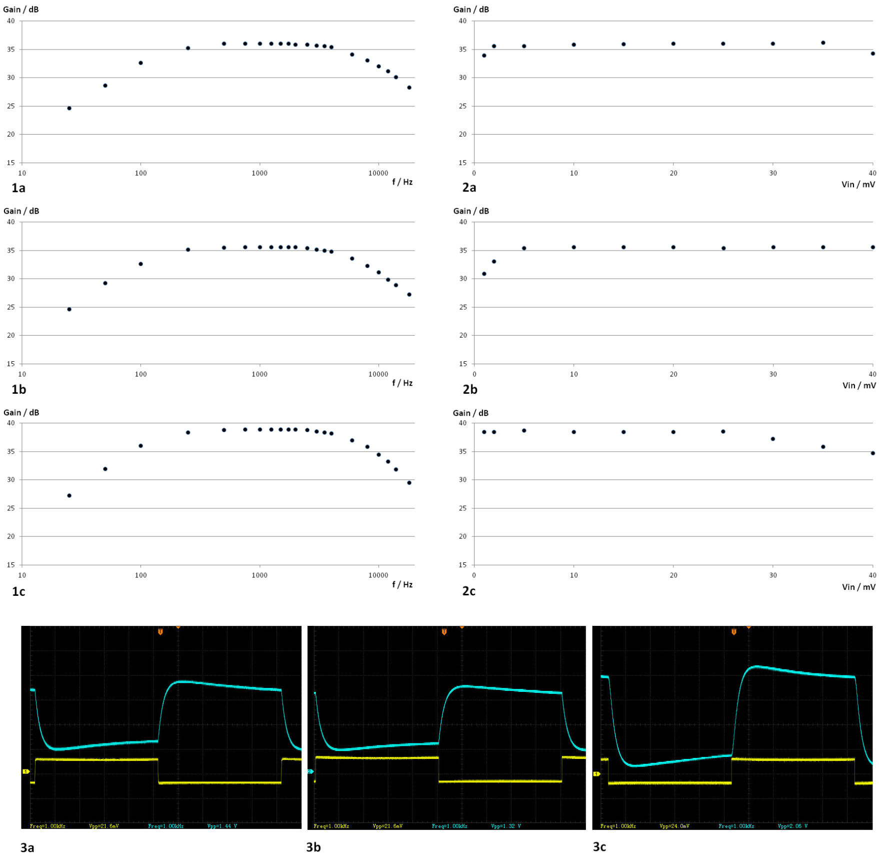

Figure 4.8 Frequency, amplitude and step responses from the best three of the preamplifiers tested. a: Inverting, low impedance. b: Inverting, mid impedance. c: Instrumentation, here 15 kΩ input impedance.

4.4 Signal processing

Sampling settings

To find suitable AD conversion settings, a 16 bits, 44.1 kHz wave file was resampled and listened to. It featured adults and children talking in the foreground, light ambient noise, a bird and talking adults further away and some recording noise.

The *bit depth* available was either 6, 8 or 12 bits. Resolution could be further increased by oversampling; sampling faster and save the average of several values as one sample. When testing different resolutions, 12 and 16 bits were rather similar in quality. 8 bits added a considerable amount of noise, especially to soft sounds.

12 bits could easily be compressed to 8 bits, which was convenient for the flash and transfer registers. 6, 8 or 12 bit resolution took 7, 9 or 13 conversion clock cycles, but with slow sampling and an AD converter wake-up time of 5 cycles, this made no real difference. Provided low-power compression, 12 bits was thus considered as the best option.

The *sampling rate*, maximum 1 MHz, was in practice limited by the AD converter energy consumption and the 1 MB flash memory. 8 kHz, a bandwidth of 4 kHz, proved to be the limit for acceptable audio quality, with adult voices still recognizable to someone who knew the person. Children's voices more obviously lacked harmonics and the bird and other high pitched sounds were subdued. Much of the recording noise, though, was also filtered away.

As AD conversion energy and sample storage space scaled linearly with sampling frequency, the bandwidth would have to be subject to negotiation. The prototype sampling was finally set to 8 kHz, but increasing it would just require one changed code constant and switching of the preamplifier low-pass capacitor.

Compression

After the sampling, the first question was whether to put signal processing in the microcontroller or in the client app. With limited flash memory size and the transmission energy being high compared to other activity (cf. calculations in Chapter 5.1), it was obvious that server-side compression would be useful. If the compression algorithm allowed for it, any other signal processing could be done in the app.

The compression algorithm should preferably be free, light in terms of encoder memory requirements, not too heavy on decoding and suitable for small chunks of low resolution audio. With this in mind, wave and DPCM codecs were written and Flac, Opus and Speex codecs were downloaded and examined (Table 4.1).

CODEC	FLASH / kB	RAM / kB	SAMPLE BUFFER / ms
Wave	3	1	-
Flac	60	20	4-92
Opus	40	10	5-67
Speex	30	7	20
DPCM	2	1	-

Table 4.1 Encoding memory requirements for the codecs considered. Values vary with implementation and with compression settings and are given only as very rough approximations.

DPCM was selected for the prototype. The main advantage was its light encoding, which could be done after each sample and eliminated the need for a raw sample buffer and a separate signal processing interrupt, and also the fact that it was nice to have an excuse for writing a codec.

A DPCM encoder does not save samples, but the difference between the current sample and a prediction [cf. Girod, 2016]. The prediction was in this case derived from the previous sample (Fig. 4.9). Compression was achieved by quantizing the differences into 256 bins, with higher resolution, tighter bin intervals, for the most probable differences. Such an algorithm is well suited to audio, as the correlation between subsequent samples is strong provided sufficient sampling rate.

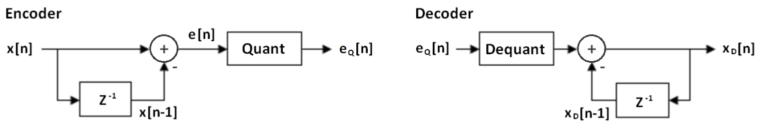

Figure 4.9 DPCM encoding and decoding. x is the sample and e is the prediction error which is quantized and saved.

Encoding was made fast, for each sample a subtraction and an eight comparison binary search in a lookup table to find the correct bin. Decoding before app playback only meant one addition per sample from a lookup table containing the average value for each bin. A wave header was also added.

DPCM compression quality depends on how well the bins match the actual differences between samples. To find suitable bins, 20 s of two persons talking, 1 m and 5 m from the microphone in a room with soft background music, were recorded. The probability mass function for the differences between samples was plotted (Fig. 4.10) and the bins were defined by dividing the integral of the function in 256 equal parts. To set higher resolution for example for soft sounds or low frequencies, the probability mass function could be multiplied with a weight function.

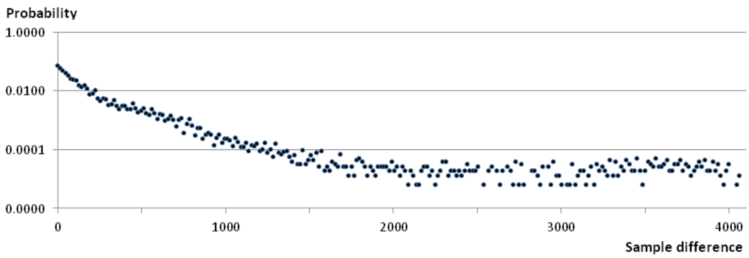

Figure 4.10 The positive half of the sample difference probability mass function.

Finally, the period for writing compressed data to the flash memory was defined. The flash memory wake-up energy was low and the context switch overhead small, so the buffer size was set to a rather arbitrarily picked small value; 300 samples.

Signal processing performance

The bit rate ended at 64 kbit/s on the recorder side and 128 kbit/s decoded to 16 bit wave in the app. A more advanced codec would reach a third of that. Lowering the amount of DPCM bins to 64, 48 kbit/s, produced much noise, especially for soft sounds, but could have worked better with a specifically adapted bin definition.

The prototype amplitude and frequency responses can be seen in Figure 4.11. The same measurements were also repeated without compression, but the results were similar, as the compression did not affect sound quality much.

As time was limited, no additional signal processing was implemented. The bandwidth reduction had already decreased the amount of noise and the microphone, preamplifier and AD converter linearity was sufficient. Improvements could of course be done, but would be easily added on the app side.

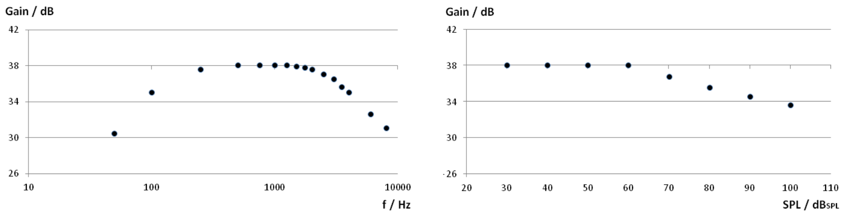

Figure 4.11 The prototype system amplitude and frequency responses.

A short evaluation of hardware codecs was also done. DSP functionality or more advanced compression proved to be beyond budget, but quite a few microphone drivers were conceivable. These cost around SEK 13-20 and contained AD converters, preamplifiers, filters, automatic gain control, noise reduction, DA converters and headphone output amplifiers. Chip size went from 3 mm to 5 mm and the recording current consumption was around 4 mA with high audio resolution.

The extra space requirements would be modest, but the total current would increase by 50 % (cf. Chapter 5.1) and the price would almost double. Nevertheless, in lack of development time or system resources, an external microphone driver could probably improve sound quality drastically.

4.5 Transmission

Transmission implementation

As mentioned earlier, the low-power radio protocols supported by the EFR32MG chip were not intended for large file transfer. One possibility would have been to work around the intended protocol stacks and implement other stacks or send raw data packages without overhead, but it would have required much more time.

For this particular system a long range was not needed, as it would then be more natural to pass through a wired gateway. If transmission time was not an issue, ZigBee and Thread would be preferred for their low energy consumption. Nevertheless, as speed was important and the target would be a smartphone, which typically would support only Bluetooth Low Energy, BLE was chosen and tweaked to maximize throughput.

An audio transfer BLE service was created with four characteristics, or transfer attributes (Fig. 4.12). These were used for recording requests and alerts, data requests and data transfer. The radio link and the BLE profile were set up with the BLE Generic Attribute Profile and Silicon Lab's BLE API. Omitting them would have been possible, but not likely worth the small gain in memory and energy.

Figure 4.12 Transmission tests with a free BLE scanner app.

The theoretical BLE raw data transfer speed, 270 kbit/s, is virtually impossible to achieve due to hardware and software limitations and radio interference. Acknowledged packets normally end up around 8-10 kbit/s [Rowberg, 2012]. In order to maximize transfer speed, the connection interval was set to minimum, 7.5 ms, the packet data size was set to maximum, 20 bytes per packet, and data was sent as repeated notifications, which are non-acknowledged at application level, controlled by a software timer.

Without acknowledgements, it was actually possible to set a timer to send several packages within one connection interval, which was tested. The link layer would still have acknowledgements and retransmit lost packages, which needed to be taken into account. The package rate was then limited by retransmissions, buffer capacity and the hardware transfer speed.

Transmission performance

Transfer speed was measured to average 14.4 kbit/s with acknowledgements, on 5 m distance in an office with 12 employees and a moderately occupied 2.4 GHz band. Sending up to seven 20 bytes packages in one transmission interval worked well for a small number of repeated connection intervals, yielding as much as 149 kbit/s. Transferring 1 s of 64 kbit/s audio would then take 0.43 s.

From an energy point of view, the result was promising. To minimize memory usage, only one connection was allowed. Deep sleep with an interrupt wake-up time of 10 μs was entered between all transmission events. According to the chip datasheet, sleep current would then be 3.3 μA, with 20 bytes transmission bursts at 127 mA for approximately 400 μs [Silicon Labs, 2016b].

4.6 Software

Software design

Except for the user interface app (Chapter 4.8), all software was written in C, compiled with an IAR Systems compiler and executed on the EFR32 ARM Cortex-M4 processor. The application consisted of a main executable, handling the overall system structure and ten drivers for AD converter, flash memory, timing, radio transmission and debug interface. Silicon Labs' pre-compiled Bluetooth Low Energy stack was also used, as well as extracted parts of their low level driver library, EMLIB, and BLE library, BGAPI.

The software was developed with energy efficiency as a main focus. All functionality was interrupt driven, peripherals were deactivated when not in use and the processor was set to deep sleep, at 3.3 μA, with only a low energy timer running between interrupts. Library usage and function calls were kept reasonably low.

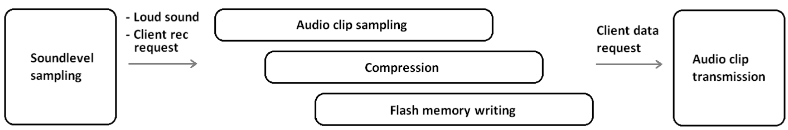

Figure 4.13 Basic recording software flow.

After initialization, the application sampled two seconds of low resolution audio at given intervals and checked the signal energy. Energy was estimated by simply counting the number of samples above an amplitude threshold value. Audio clip recording was started either when loud sounds were detected or when a recording start command was received from the client (Fig. 4.13).

Audio samples were compressed directly and stored in a buffer. Every 300 B, the new buffer data was written to the flash memory. After a completed recording, a notification was sent to the client. The client could request transmission of the latest saved audio at any point. The system flowchart is depicted in Figure 4.14.

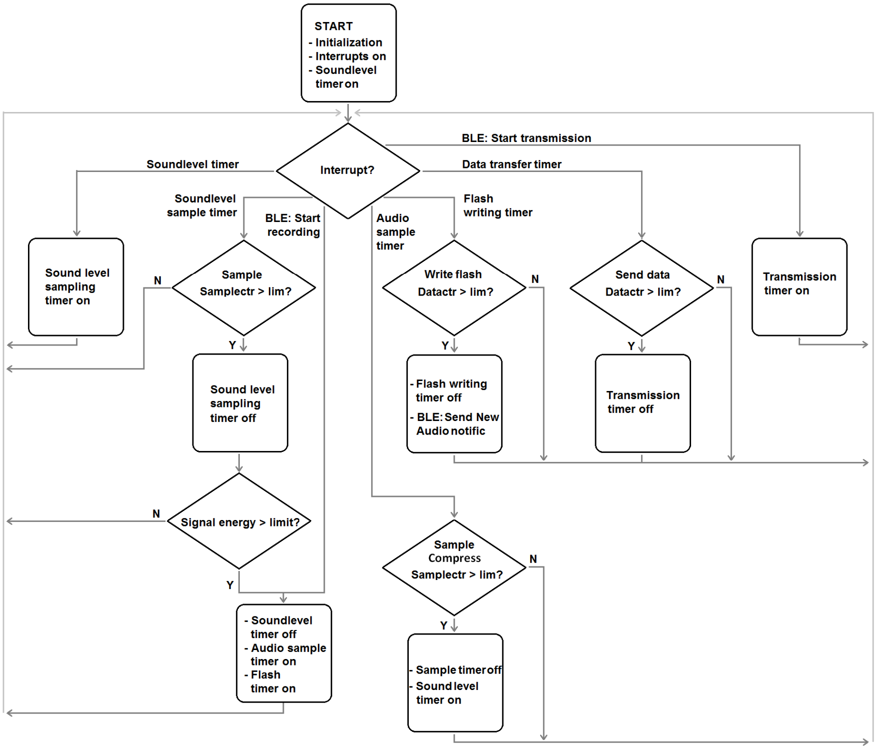

Figure 4.14 Flowchart for the system software.

Software performance

Software energy efficiency was explored by counting clock cycles for the heaviest tasks with a hardware timer, calculating energy consumption according to datasheet values for all active components (Table 4.2) and checking against measured current consumption. Different methods to decrease the energy consumption, adapted from Chapter 3.6, were then implemented one at a time.

COMPONENT	CURRENT / mA	DURATION
Processor, active, 38.4 MHz	3.840	Interrupts
Processor, active, 9.6 MHz	0.960	Interrupts
Processor, deep sleep	0.003	Btw interrupts
AD converter, duty cycled	1.440	20 clk cycles
Ext flash memory, writing	3.500	17 μs / B
BLE hardware, transmission	126.700	20 μs / B

Table 4.2 Current for the active hardware components in the test, with the settings used in the prototype [Silicon Labs, 2016b].

Test results are presented in Figure 4.15. The small difference from compiler optimization is no surprise, as the code was rather hardware oriented. Compiler size optimization only reduced the executable file size, 137 kB, by 2.2 %.

Keeping instructions in the cache memory for fast access could have made a difference with signal processing, but did almost nothing for the cases tested. Decreasing the processor clock frequency would have been a significant improvement for a constantly running application, but did not affect interrupt based tasks.

	Sample 1 s 12 bits, 8 kHz, compr	Wr to ext flash 300 B	Transm fr buffer 140 B
Initial energy cons / mAs @ 2.5 V	0.450	0.024	0.365
Compiler speed optimization	1.00	0.96	1.00
Compiler size optimization	1.00	0.97	1.00
Cache instructions for reusage	1.00	0.98	1.00
Decrease CPU clock, 38.4 to 9.6 MHz	1.00	1.00	0.98

Figure 4.15 Energy consumption comparison for different software improvements, indexed against the initial energy consumption.

Provided that one single client used the system, three tasks would run at the same time at maximum load; sampling, writing to flash memory and data transmission. These were timed, given interrupt priorities and analysed for schedulability (Table 4.3). As execution time was rather constant, the worst case was estimated from ten executions without the optimization tested above.

The priorities needed to be set so that an even sample rate was guaranteed. The sample buffer should be written to the flash memory before overflowing. Transmission had low priority but could be given more or less all the time left.

TASK	WORST CASE EXEC / ms	PERIOD / ms	PRIORITY
Sample once, compr, buffer	0.014	0.125	1
Write 300 B to ext flash	0.916	37.5	2
Transmit 140 B fr ext flash	7.819	x	3

Table 4.3 Concurrent tasks at maximum system load. 12 bit sampling at 8 kHz, compression to 8 bits and no transmission time constraints are assumed.

The priorities matched rate monotonic scheduling, higher for shorter periods, so schedulability was checked against this condition (Equation 4.1). U is processor utilization, n is the number of tasks and C_i and T_i are estimated worst case execution times and periods [Wittenmark et al., 2012]. Ideally, deadlines should be equal to task periods. This compromised an exact sampling timing, but as sampling had the highest priority and would interrupt other tasks, it would not be any major problem.

$$U = \sum_{i=1}^{n} \frac{C_i}{T_i} \leqslant n(2^{1/n} - 1) \tag{4.1}$$

Schedulability was guaranteed for transmission periods down to 13 ms, yielding transfer speed up to 86 kbit/s. 12 bit samples with 16 times oversampling, to 16 bits, was also tested and timed to 0.107 ms/sample. To guarantee schedulability, sampling, compression and flash writing then had to be done in a separate state, without the Bluetooth Low Energy interrupt structure overhead.

4.7 Energy consumption feedback

Supply voltage and clock frequency

Going through the system, a large number of energy and performance factors could be tuned during run-time. Finding feedback loops that would be efficient and actually improve performance compared to constant values was harder, though.

When returning to the feedback loops discussed in Chapter 3.7 (Fig. 4.16), controlling the supply voltage to the microphone, preamplifier and microcontroller would be the first suggestion. For the prototype, this would mean accessing the microcontroller voltage converter, which was not intended to be broken out for external usage. Furthermore, the lowest voltage possible would be generally be a good decision. This was indeed two different levels, 2.5 V when recording and 1.8 V, the supply minimum, otherwise, but switching those two states would not need a control loop.

Adjusting the clock frequency would not make any major difference for interrupts just using the processor and peripherals with variable clock speed, as the processor would go to deep sleep between interrupts. Higher frequency would mean linearly more current but also linearly shorter time in the active mode. When using fixed speed peripherals, though, the processor frequency could be lowered to match the time available. This could have been applicable for the flash memory, but it was fast enough to let the processor run at full speed and get the memory back to sleep as soon as possible.

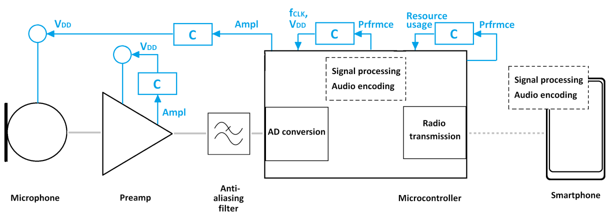

Figure 4.16 The suggested feedback loops from Chapter 3.7 reviewed.

Resource utilization

For the system resources, conceivable control variables could be bit depth, sample rate, compression, buffer size and transmission settings. The interval for checking for loud sounds also had a large impact on energy consumption, but it needed to be predictable and perhaps also set by the user.

AD conversion was most efficient at 8 or 12 bits. Going down to 8 bits saved energy and eliminated compression, but cost much in terms of audio quality, especially as the compression algorithm was very simple and the sound level would often be low. Variable sample rate could be more motivated, but the real-time signal processing was expensive compared to a low constant rate or even to a higher constant rate and harder compression.

The *compression algorithm* could be turned into its adaptive shape, ADPCM. It would also be natural to add automatic gain control, but the low complexity had been the main reason for the choice for algorithm. Still, with decompression to wave files on the app side and all other signal processing done there, the compression algorithm could easily be improved in the future.

The buffer size could be varied, though, to sacrifice some RAM memory for waking the flash memory more seldom. Still, the flash wake-up time was only 65 μs around 10 μA and not really worth the cost for a control loop. With a more advanced compression algorithm, which benefited from larger buffer size, it may have been more motivated.

Transmission was already set up to maximize speed and, secondly, to avoid unnecessarily high energy consumption. What could be done, was to keep track of the active system tasks and adjust the transmission period after the actual situation and not after the worst case scenario. It would not really save energy, but yield maximum transfer performance from a low-power system at any given point.

This feedback loop, though, was implemented by simply letting transmission replace the deep sleep state whenever running. In that way, it would get all time left between the other interrupts.

Separate connection intervals still needed to be started with timer interrupts, as the Bluetooth Low Energy stack had its own internal interrupt structure and constant transmission would block other BLE events, but the transmission period could be set equal to the connection interval so that a new intervals were constantly opened. Starvation would happen, but only affect transmission and causing nothing worse than pauses between packages.

4.8 Interface app

As a user interface for the prototype, an Android app was made. The app listed the latest audio clips, fetched new data and notified when new files were available. The user could play existing audio clips and start recording new ones with a record button (Fig. 4.17).

In terms of software, the app had three main features. Most importantly, it worked as a Bluetooth Low Energy client, connecting to the recorder server, getting notifications about new data, requesting to start recording or to transfer clips and receiving audio samples. Secondly, it collected the received audio samples, decoded them and saved to wave files, and thirdly, it implemented an audio player.

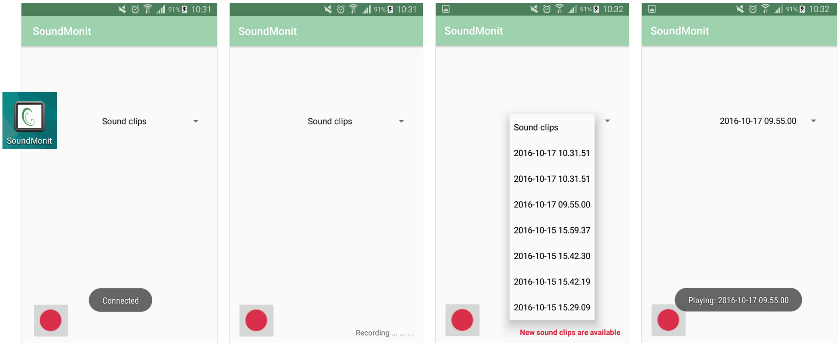

Figure 4.17 The user interface app.

5

Energy consumption, size and audio quality

5.1 Energy consumption

The discussions from previous chapters, with energy consumption limits and trade-off, is summarized in Table 5.1. These are by no means absolute theoretical minimum values, but indications of what is achievable.

Minimum microphone and preamplifier current was estimated from the tested components at 2.5 V. Signal processing was calculated from 8 bit, 8 kHz sampling compressed to 6 bits. File storage and transmission corresponds to 48 kbit/s and, importantly, transmission power lowered to 10.5 dBm. The software calculation was based on a 10 MHz system clock, allowing for concurrent sampling and audio storage but separate transmission after a completed recording.

COMPONENT	CURRENT / mA	DURATION / s	LOW ENERGY TRADE-OFF
Microphone	0.06	1.0	Sensitivity, noise, low freq cut-off
Preamplifier	0.20	1.0	Sensitivity, noise, dynamic range
Sampl, compr	0.67	0.1	Noise, resolution, file size
Audio storage	3.50	occ. erasing + 0.1	Bandwidth, bit depth
Transmission	35.00	0.1	Transfer time, range, audio quality

Table 5.1 Approximate energy consumption limits per recorded second for the given platform and trade-off when lowering energy consumption.

The prototype consumed 24.8 mAs per recorded second, with occasional erasing of the sample storage flash memory included. With the settings from Table 5.1, the system would instead require 5.6 mAs, split on 1 s of recording plus an extra 0.32 s of transmission for each recorded second.

Figure 5.1 gives a picture of the relative energy consumption per recorded second for the different components in the prototype. With the approximated minimum values, the same chart would look like Figure 5.2.

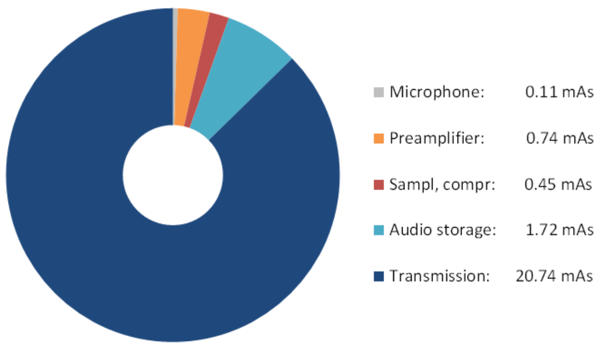

Figure 5.1 Energy consumed with the recorder prototype. The total energy was 24.8 mAs per recorded second.

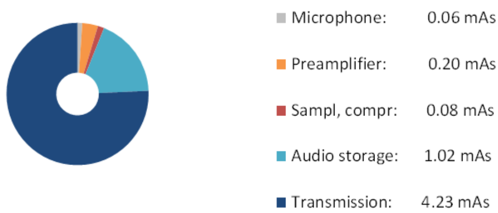

Figure 5.2 Approximate minimum energy consumed with values from Table 5.1. The total energy would be 5.6 mAs per recorded second.

Looking at the above, good microphones and preamplifiers are well worth a small current increase. The preamplifier must provide sufficient dynamic range and signal-to-noise ratio, but should be possible to cut down on quite a lot with different circuit design and low supply power or with a microphone with an integrated preamplifier. This may compromise sensitivity and the signal-to-noise ratio, though.

Transmission showed the largest improvement potential, naturally made in a signal processing equation. Sampling, compression and signal processing was relatively cheap in energy, but could save much of the storage and transmission energy and would be crucial to preserve sound quality. Audio storage in itself was not very energy consuming, but erasing the memory when full drew much current.

In Figure 5.1, some transmission energy could easily be exchanged for sampling and compression to improve audio quality. Transmission schedulability could suffer, though, resulting in a slightly slower system. As reliability and lack of notable delays were two of the basic constraints for initial system definition, any major timing changes would need to be well motivated and evaluated against system usability.

5.2 System size

The question of the minimum or most efficient system size is both simple and complex, depending on the application and the level of detail. If recording functionality is added to an existing system, the main limit would be the size of the microphone membrane for sufficient bandwidth and sensitivity. For most practical purposes, though, the system addition would consist of a MEMS microphone, with a standard length down to around 2.75 mm, and an amplifier which could be integrated and made small compared to the microphone.

For the purposes tested here, MEMS microphones did not prove inferior to the larger electrets. Larger membranes and higher supply voltage would of course make a difference in high end applications and more space could also allow for better shielding. A smaller system would generally run on less power, which is advantageous in most aspects except for sensitivity and noise, and shorter distances would decrease the amount of external noise picked up.

The minimum size for a reasonably uncomplicated system would thus be set by the microphone package size plus power supply, AD converter, processor, sample memory and transmission hardware. The latter can either be borrowed from an already existing main system or estimated to the size of a microcontroller with a couple of peripheral chips and an antenna.

With the rather modest size requirement above, the microphone placement could be a harder spatial constraint. Apart from battery limitations, sample storage would also be an issue. Storage, though, could be traded against extra energy and memory for signal processing or perhaps against space for a hardware codec.

5.3 Audio quality, size and energy consumption

Finally, audio quality should be set against energy and size constraints. For the microphone and the preamplifier, lower energy consumption typically means lower supply voltage and thus lower amplitudes. Energy gain comes to the price of reduced sensitivity, dynamic range and signal-to-noise ratio. In the digital end of the signal chain the effects are similar, but expressed in terms of bit depth, bandwidth and compression noise.

As already discussed, size reductions make the microphone lose sensitivity and low frequency content to some extent and limits positioning options, but also shortens signal paths that could pick up external noise and interference. Nevertheless, decent sound quality may well be achieved with a small system.

Among the digital components, size constraints would probably hit hardest on batteries and sample storage size and force lower audio resolution. Judged by the tests conducted, this could affect sound quality far more than microphone size limitations would, and would be a trade-off depending on the application.

6

Conclusion

Finally, it is time to return to the starting point, to the the problem definition questions and to the system requirements.

Microphone. Finding a small, low-power microphone with sufficient audio quality was not very hard. Electret microphones, either a few millimeters in diameter or, preferably, in MEMS packages, worked well.

Preamplifier. A good quality preamplifier could be made rather simple, for example with single stage op amplifier feedback circuits. Still, with a more complex design, lower supply voltage and efficient noise reduction, much energy could be saved.

Signal processing. Signal processing proved important for reducing energy consumption, by appropriate sampling settings, by data compression and by compensating for distortion and noise caused by energy reductions earlier in the signal chain. Naturally enough, this part was central for the system performance.

Transmission. As this accounted for a large part of the energy consumed, even small improvements could make a difference. The range, and thus the transmission power, proved to be most important, together with transmission speed and protocol overhead. Transmission logic was also crucial for the system timing and speed.

Software. The main purpose for the software was to create a stable and efficient platform for the other parts. Timing and energy analyses were of great help, not taking much time, but providing a valuable picture of the performance and of the areas for improvement.

Energy feedback. For the system being, no really efficient feedback implementation was found. Yet, the concept has great potential and was well worth considering. Also, when going through the system looking for feedback opportunities, a couple of statical ways to improve performance were found.

Energy. Energy consumption was finally a trade-off between audio quality, system speed and development time. The prototype required 24.8 mAs per recorded second, but theoretical estimations indicated that 5.6 mAs per recorded second could be possible to reach.

Size. Except for reduced battery capacity and microphone positioning options, size constraints would not affect performance very much. If an AD converter, a processor and radio hardware already exist, only a few square millimeters would be needed for the microphone, the amplifier and sample storage. In most devices, that would be a rather modest requirement.

Audio quality. Provided enough space for sample storage and a satisfactory microphone position, audio quality was mainly set against energy consumption. Once sufficient AD converter input quality is ensured, a well-designed signal chain could do much to keep the perceived sound quality up.

System requirements. The prototype fulfilled the initial size constraints. The audio quality was acceptable, but the sensitivity was somewhat too low. Energy consumption was higher than aimed for, with sleep current at 3.3 μA, 0.34 mAs when checking for loud sounds and 24.8 mAs per recorded second. Getting below 5 mAs could be possible if a slower system was allowed. Any delays would need to be kept short to ensure system functionality, though.

Rather good performance could be achieved on a low budget. When adding recording functionality to a system, size and energy consumption would be more likely to form obstacles than cost.

The suggested system could of course be improved in many aspects. Building on the already existing components, more advanced signal processing and completely reliable transmission would be a first priority. Measuring transmission energy with different protocols would also be interesting, to see how Bluetooth Low Energy actually compares to other standards when used in a way far from the intended.

Depending on the application, some areas overlooked would also need to be considered. Data security, at least access restriction and encryption, would be needed and cause some overhead. Interaction design and use cases could have been explored further, to identify the most important system aspects and put more effort on those. It would also be natural have the recorder connected to a gateway instead of a smartphone, but that would not require any major system changes.

To sum up, this project was very interesting to work with. It touched upon a vast variety of aspects and fields, from microphone physics and efficient radio transceiver software to calculating encoder quantization and designing an app interface. Hopefully, the results may inspire some further experiments.

7

Summary

The aim of this thesis was to investigate and evaluate current methods for building tiny, low-power audio recording systems for wireless applications. Short audio clips should be recorded in decent quality, transferred to a smartphone and played back. Optimization of the different constituent parts should be explored and, finally, a prototype for usage in battery powered devices with limited space should be built.

The system requirements were defined in terms of audio quality, physical size and energy consumption. Additionally, the price should be kept as low as possible and the system should be reliable and stable, without notable delays. The only predefined part was the microcontroller platform, a low-power, general purpose radio chip with an AD converter and 1 MB of external flash memory.

The system was divided into six focus areas; microphones, preamplifiers, signal processing, transmission, software design and energy consumption feedback. These were explored theoretically and by practical prototyping, testing and tuning.

The prototype was implemented with a MEMS electret microphone, an instrumentation amplifier, 12 bit sampling at 8 kHz and DPCM compression to 64 kbit/s. Samples were transmitted with Bluetooth Low Energy tweaked to maximize transfer speed. A client app was made as a user interface for receiving, decompressing, saving and playing audio clips.

Finally, the complete system was evaluated. Energy consumption was mainly set against audio quality and system speed. Signal processing and transmission, especially range, speed and protocol overhead, played major roles. The prototype required 24.8 mAs per recorded second, but theoretical estimations indicated that 5.6 mAs would be achievable.

Size constraints did not affect performance very much, except for reduced battery capacity and microphone positioning options. This was especially true if the AD converter, processing power and radio hardware from an already existing platform could be used. Decent quality could be achieved with small size and low cost and signal processing could do much to keep energy consumption down and sound quality up.

References

An Sun, Y. S. and G. Farrell (2010). “Low Cost Disposable Reflective Optical Fibre Microphone”. *Microwave and Optical Technology Letters, vol 52 no 7*. Wiley, New York, pp. 1504–1507.

ARM (2015). *ARM Cortex-M4 Processor: Technical Reference Manual, rev r0p1*. ARM Ltd, Cambridge.

Ayllón, D., R. Gil-Pita, and M. Rosa-Zurera (2009). “Design of Microphone Arrays for Hearing Aids Optimized to Unknown Subjects”. *Signal Processing, vol 93*. Focal Press, London, pp. 3239–3250.

Ballou, G., J. Ciaudelli, and V. Schmitt (2005). “Microphones”. *Ballou, Glen (ed): Electroacoustic Devices, Microphones and Loudspeakers*. Newnes, Amsterdam, pp. 3–193.

Chiu, L. K. (2012). *Efficient Audio Signal Processing for Embedded Systems*. Georgia Institute of Technology, Georgia.

Coates, E. (2016). *Amplifiers*. Learn About Electronics, www.learnabout-electronics.org/Amplifiers/amplifiers10.php, accessed 2016-06-28.

Cornelius, P., N. Grbic, and I. Claesson (2005). “Microphone Array System for Speech Enhancement in a Motorcycle Helmet”. *Blekinge Institute of Technology Research Report, vol 2005:5*. Blekinge Institue of Technology, Karlskrona.

Crepaldi, M., M. Stoppa, P. M. Ros, and D. Demarchi (2015). “An Analog-Mode Impulse Radio System for Ultra-Low Power Short-Range Audio Streaming”. *IEEE Transactions on Circuits and Systems I: Regular Papers, vol 62 no 12*. IEEE Circuits and Systems Society, Piscataway, pp. 2886–2897.

Crocco, M., M. Critani, A. Trucco, and V. Murino (2016). “Audio Surveillance: A Systematic Review”. *ACM Computing Surveys, vol 48 no 4, article 52*. Association for Computing Machinery, New York, pp. 52:1–52:46.

Czarny, J. (2015). *Conception, Fabrication and Characterization of a MEMS Microphone*. Institut National des Sciences Appliquées de Lyon, Lyon.

Decuir, J. (2014). “Introducing Bluetooth Smart. Part 1: A Look at Both Classic and New Technologies”. *IEEE Consumer Electronics Magazine, January 2014*. IEEE Consumer Electronics Society, Piscataway, pp. 12–18.

Digikey (2016). *PUI Audio, Inc. AB1541B-LW100-R*. Digi-Key Electronics, www.digikey.se/product-search/en?keywords=668-1448-ND, accessed 2016-10-06.

Du, D. and K. Odame (2012). “An Adaptive Microphone Preamplifier for Low Power Applications”. *Proc. ISCAS ’12 IEEE International Symposium on Circuits and Systems Conference*. IEEE, Piscataway.

Dutta, P. K. and D. Culler (2005). “System Software Techniques for Low-Power Operation in Wireless Sensor Networks”. *Proceedings of the ICCAD-2005*. Association for Computing Machinery, New York, pp. 924–931.

Ekström, M. C., M. Bergblomma, M. Lindén, M. Björkman, and M. Ekström (2012). “A Bluetooth Radio Energy Consumption Model for Low-Duty-Cycle Applications”. *IEEE Transactions on Instrumentation and Measurement, vol 61, no 3*. IEEE Instrumentation and Measurement Society, New York, pp. 609–617.

Girod, B. (2016). *DPCM - Overview*. Stanford University: Image and Video Compression, lectures notes, http://web.stanford.edu/class/ee368b/Handouts/15-DPCM.pdf, accessed 2016-11-04.

Girod, L. and M. A. Roch (2006). “An Overview of the Use of Remote Embedded Sensors for Audio Acquisition and Processing”. *Proceedings of the Eighth IEEE International Symposium on Multimedia*. IEEE Computer Society, Washington.

Groschup, R. and C. U. Grosse (2015). “MEMS Microphone Array Sensor for Air-Coupled Impact-Echo”. *Sensors, vol 2015*. MDPI AG, Basel, pp. 14932–14945.

Harpe, P., H. Hegt, and A. van Roermund (2010). *Smart AD and DA Conversion*. Springer, Dordrecht.

Hebert, G. K. (2010). *Designing Microphone Preamplifiers*. www.thatcorp.com/datashts/AES129_Designing_Mic_Preamps.pdf, accessed 2016-06-28. 129th AES Convention, San Francisco.

Hillenbrand, J, S Haberzettl, and G. M. Sessler (2013). “Electret Microphones with Stiff Diaphragms”. *Journal of the Acoustical Society of America, vol 134*. Acoustical Society of America, Woodbury.

Homentcovschi, D. and R. N. Miles (2011). “An Analytical-Numerical Method for Determining the Mechanical Response of a Condenser Microphone”. *Journal of the Acoustical Society of America, vol 130*. Acoustical Society of America, Woodbury.

Hood, J. L. (1997). *Valve and Transistor Audio Amplifiers*. Elsevier, Amsterdam.

Huang, P.-K. and S. Ghiasi (2007). “Efficient and Scalable Compiler-Directed Energy Optimization for Realtime Applications”. *ACM Transactions on Design Automation of Electronic Systems, vol 12, no 3, article 27*. Association for Computing Machinery, New York.

IEC (2010). *International Standard: Sound system equipment – Part 4: Microphones. IEC 60268-4, Edition 4.0*. International Electrotechnical Commission, Geneva.

Invensense (2013). *Application Note AN-1165: Op Amps for MEMS Microphone Preamp Circuits, rev 1.0*. Invensense Inc, https://store.invensense.com/datasheets/invensense/Op-Amps-for-MEMS-Microphone-Preamp-Circuits7.pdf, accessed 2016-07-06.

Jawed, S. A., J. H. Nielsen, M Gottardi, A Baschirotto, and E Bruun (2009). “A Multifunction Low-Power Preamplifier for MEMS Capacitive Microphones”. *Proceedings of ESSCIRC, vol 2009*. IEEE, Piscataway, pp. 292–295.

Jeng, Y.-N., T.-M. Yang, and S.-Y. Lee (2011). “Response Identification in the Extremely Low Frequency Region of an Electret Condenser Microphone”. *Sensors, vol 2011*. MDPI AG, Basel, pp. 623–637.

Johann, T., M. Dick, S. Naumann, and E. Kern (2012). “How to Measure Energy-Efficiency of Software: Metrics and Measurement Results”. *Proceedings of GREENS 2012*. IEEE, New York, pp. 51–54.

Jones, R. (2010). *Microphones, Preamps and Impedance*. Microphone Data, microphone-data.com/media/filestore/articles/Mic%20impedance-10.pdf, accessed 2016-06-29.

Kandemir, M., N Vijaykrishnan, and M. J. Irwin (2002). “Compiler Optimization for Low Power Systems”. *Robert Graybill and Rami Melhem (eds): Power Aware Computing*. Kluwer Academic Publishers, New York, pp. 191–210.

Kendrick, P., I. R. Jackson, B. M. Fazenda, T. J. Cox, and F. F. Li (2015). “Microphone Handling Noise: Measurements of Perceptual Threshold and Effects on Audio Quality”. *PLoS One, vol 2015*. San Francisco.

Kim, Y.-G., M.-H. Cho, J. Lee, Y.-D. Jeon, T. M. Roh, C.-G. Lyuh, W. S. Yang, and J.-K. Kwon (2016). “Low-Noise MEMS Microphone Readout Integrated Circuit Using Positive Feedback Signal Amplification”. *ETRI Journal, vol 38, no 2*. Electronics and Telecommunications Research Institute, Daejeon, pp. 235–243.

Knowles (2011). *SiSonic Design Guide, Rev 3.0*. Knowles Acoustics, Itasca.

Knowles (2012). *SPU0414HR5H-SB. Amplified SiSonic Microphone Datasheet, Revision: E*. Knowles Acoustics, Itasca.

Liu, H., D. McLachlan, and D. Wang (2015). “Overview of Wireless Microphones—Part I: System and Technologies”. *IEEE Transactions on Broadcasting, vol 61, no 3*. IEEE Broadcast Technology Society, Piscataway, pp. 494–504.

Madinei, H., G. Rezazadeh, and N. Sharafkhani (2013). “Study of Structural Noise Owing to Nonlinear Behavior of Capacitive Microphones”. *Microelectronics Journal, vol 44*. Elsevier, Oxford, pp. 1193–1200.

Maggio, M., H. Hoffmann, M. D. Santambrogio, A. Agarwal, and A. Leva (2013). “Power Optimization in Embedded Systems via Feedback Control of Resource Allocation”. *IEEE Transactions on Control Systems Technology, vol 21, no 1*. IEEE Control Systems Society, New York, pp. 239–246.

Mellor, D. (2006). “What Is Class-D Amplification? The Benefits Explained”. *Sound On Sound, vol June 2016*. www.soundonsound.com/techniques/what-class-d-amplification, accessed 2016-06-28.

MSI (1999). *Piezo Film Sensors: Technical Manual, rev B02*. Measurements Specialties Ltd, Norristown.

Naik, K. and D. S. L. Wei (2001). “Software Implementation Strategies for Power-Conscious Systems”. *ACM Mobile Networks and Applications, vol 6*. Springer, New York, pp. 291–305.

Nedzelnitsky, V. (1995). “Laboratory Microphone Calibration Methods at the National Institute of Standards and Technology, U.S.A”. *G S K Wong and T F W Embleton (eds): AIP Handbook of Condenser Microphones*. American Institute of Physics Press, Woodbury, pp. 145–160.

OML (2016). *Piezos*. Open Music Labs, www.openmusiclabs.com/learning/sensors/piezos/index.html, accessed 2016-07-11.

PUI Audio (2016). *AOM-4544P-R*. PUI AUDIO Inc, www.puiaudio.com/product-detail.aspx?categoryId=4&partnumber=AOM-4544P-R, accessed 2016-10-16.

Rayburn, R. A. (2012). *Eargle’s Microphone Book: From Mono to Stereo to Surround. A Guide to Microphone Design and Application, 3rd ed*. Elsevier, Amsterdam.

Rivers, M. (2010). *Mic Preamps - What You Need To Know*. Mike Rivers Audio, mikeriversaudio.wordpress.com/technical-articles/, accessed 2016-06-29.

Rowberg, J. (2012). *Maximize Throughput with BLE Modules*. Bluegiga Technologies Ltd, https://bluegiga.zendesk.com/entries/22400867–HOW-TO-Maximize-throughput-with-BLE-module, accessed 2016-10-07.

Roy, K. and M. C. Johnson (1997). “Software Design for Low Power”. *Wolfgang Nebel and Jean Mermet (eds): Low Power Design in Deep Submicron Electronics. NATO ASI Series, volume 337*. Advanced Science Institutes, Dartmouth, pp. 433–460.

Saxe, E. (2010). “Power-Efficient Software”. *Communication of the ACM, vol 53, no 2*. Association for Computing Machinery, New York, pp. 44–48.

Schulte, E., J. Dorny, S. Harding, S. Forrest, and W. Weimery (2014). “Post-Compiler Software Optimization for Reducing Energy”. *Proceedings of ASPLOS 2014*. Association for Computing Machinery, New York.

Shin, K., J. Jeon, J. E. West, and W. Moon (2015). "A Micro-Machined Microphone Based on a Combination of Electret and Field-Effect Transistor". *Sensors, vol 2015*. MDPI AG, Basel.

Silicon Labs (2013). *Application Note AN0027: Energy Optimization, rev 1.03*. Silicon Laboratories Inc, www.silabs.com/Support%20Documents/TechnicalDocs/AN0027.pdf, accessed 2016-09-08.

Silicon Labs (2016a). *EFR32 Mighty Gecko Wireless SoC Starter Kit*. Silicon Laboratories Inc, www.silabs.com/products/wireless/mesh-networking/efr32-mighty-gecko/Pages/mighty-gecko-starter-kit.aspx, accessed 2016-06-22.

Silicon Labs (2016b). *EFR32MG1 Mighty Gecko ZigBee & Thread SoC Family Data Sheet, rev 0.951*. Silicon Laboratories Inc, Austin.

Silicon Labs (2016c). *The Wireless Protocols Tying Together the Internet of Things*. Silabs White Papers. Silicon Laboratories Inc, www.silabs.com/products/wireless/Pages/wireless-protocols.aspx, accessed 2016-08-13.

Silicon Labs (2016d). *UG103.1: Application Development Fundamentals: Wireless Networking, Rev 1.0*. Silicon Laboratories Inc, Austin.

Silicon Labs (2016e). *UG103.14: Application Development Fundamentals: Bluetooth Smart Technology, Rev 0.1*. Silicon Laboratories Inc, Austin.

Starecki, T. (2010). "Ultra-Low-Noise Preamplifier for Condenser Microphones". *Review of Scientific Instruments, vol 81*. American Institute of Physics, College Park.

Tu, T.-Y., P. C. P. Chao, and Y.-T. Mai (2013). "A Low-Power Self-Biased Rail-to-Rail Preamplifier as a Readout Circuit for a Capacitor-Type Microphone". *Microsystem Technologies, vol 19*. Springer, Berlin, pp. 1329–1343.

USDL (2016). *OSHA Technical Manual*. United States Department of Labor, www.osha.gov/dts/osta/otm/new_noise/, accessed 2016-06-22.

van der Woerd, A. C. and W. A. Serdijn (1993). "Low-Voltage Low Power Controllable Preamplifier For Electret Microphones". *IEEE Journal of Solid-State Circuits, vol 28, no 10*. IEEE Solid-State Circuits Society, New York, pp. 1052–1055.

Waggoner, B. (2010). *Compression for Great Video and Audio*. Focal Press, Burlington.

Walser, S, C Siegel, M Winter, G Feiertag, M Loibl, and A Leidl (2016). "MEMS Microphones with Narrow Sensitivity Distribution". *Sensors and Actuators A, vol 2016*. Elsevier, Oxford, pp. 423–492.

Wikiaudio (2016). *Audio Sample Rate and Bit Depth*. Wikiaudio, http://en.wikiaudio.org/Audio_sample_rate_and_bit_depth_tutorial, accessed 2016-09-08.

Wittenmark, B., K. J. Åström, and K.-E. Årzén (2012). *Computer Control: An Overview*. IFAC Professional Brief. Department of Automatic Control, Lund University, Lund.

Wolfe, J., M. Morais Duke, E. Schafer, C. Jones, H. E. Mülder, A. John, and M. Hudson (2015). “Evaluation of Performance With an Adaptive Digital Remote Microphone System and a Digital Remote Microphone Audio-Streaming Accessory System”. *American Journal of Audiology, vol 24*. American Speech-Language-Hearing Association, Rockville, pp. 440–450.

Zendra, O. (2006). “Memory and Compiler Optimizations for Low-Power and -Energy”. *Proceedings of ICOOOLPS 2006*. ICOOOLPS 2016, Nantes.

Zhao, G., H. Ma, Y. Sun, and H. Luo (2012). “Design and Implementation of Enhanced Surveillance Platform with Low-Power Wireless Audio Sensor Network”. *International Journal of Distributed Sensor Networks, vol 2012*. Hindawi Publishing Corporation, Cairo.

Zhurikhin, D. M. (2011). “Energy-Saving Compilation for Mobile Systems”. *Programming and Computer Software, vol 37, no 6*. Pleiades Publishing, Moscow, pp. 306–314.

Abbreviations

aac	Advanced audio coding; a standard for lossy digital audio compression.
alac	Apple lossless audio codec; a standard for lossless digital audio encoding.
ADPCM	Adaptive differential pulse-code modulation; a lossy compression standard.
API	Application programming interface; a pre-made low-level code interface.
BLE	Bluetooth Low Energy, also known as Bluetooth Smart; a radio protocol.
CPU	Central processing unit; main processor.
dBA	Sound pressure level unit adapted to the frequency response of the ear.
dB_{SPL}	Sound pressure level unit with reference point 0 $\mathrm{dB_{SPL}}$ = 20 μPa.
dBu	RMS voltage unit with reference point 0 dBu = 0.775 $\mathrm{V_{RMS}}$.
dBV	RMS voltage unit with reference point 0 dBV = 1 $\mathrm{V_{RMS}}$.
DPCM	Differential pulse-code modulation; a lossy compression standard.
DSP	Digital signal processor; a specialized microprocessor.
FFT	Fast Fourier transform; an algorithm finding signal frequency components.
flac	Free lossless audio codec; a standard for lossless audio compression.
MAC	Multiplier-accumulator; a hardware unit for efficient addition of products.
MEMS	Micro-electro-mechanical system; a system designed in microscopic size.
MIPS	Million instructions per second.
mp3	MPEG-1/2 audio layer III; a standard for lossy digital audio compression.
RISC	Reduced instruction set computing; processor design with fewer instructions at higher speed.
RMS	Root mean square; the arithmetic mean of the squares of a set of numbers.
SIMD	Single instruction, multiple data; operation on several parallel data points.
SPI	Serial peripheral interface; a synchronous communication interface.
SPL	Sound pressure level; the pressure change caused by a sound wave.
V_{PP}	Peak-to-peak voltage.

Lund University
Department of Automatic Control
Box 118
SE-221 00 Lund Sweden

Document name
MASTER´S THESIS
Date of issue
December 2016
Document Number
ISRN LUTFD2/TFRT--6024--SE

Author(s)
Rebecka Erntell

Supervisor
Mats Iderup, Mikrodust AB
Björn Strandmark, Mikrodust AB
Noroz Akhlagi, Verisure Innovation AB
Martina Maggio, Dept. of Automatic Control, Lund University, Sweden
Anders Robertsson, Dept. of Automatic Control, Lund University, Sweden (examiner)

Sponsoring organization

Title and subtitle
Embedded Recording - Tiny, low-power audio solutions for wireless systems

Abstract

The aim of this thesis was to explore and evaluate current methods for building tiny, low-power audio recording systems for wireless applications. Short audio clips should be recorded, transferred to a smartphone and played back. Optimization of the different parts should be explored and a prototype for usage in battery powered devices with limited space should be built.
System requirements were defined in terms of audio quality, physical size and energy consumption. Additionally, the price should be kept low and the system should be reliable and stable, without notable delays.

The system was divided into six focus areas; microphones, preamplifiers, signal processing, transmission, software design and energy consumption feedback. Each of those was explored theoretically and by practical prototyping, testing and tuning. Low energy consumption mainly proved to compromise audio quality, system speed and development time. Signal processing and transmission, especially range, speed and protocol overhead, played major roles.
Size constraints did not affect performance very much, except for reduced battery capacity and microphone positioning options. Decent sound quality could be achieved with small size and low cost and signal processing could do much to keep energy consumption down and the perceived sound quality up.

Keywords

Classification system and/or index terms (if any)

Supplementary bibliographical information

ISSN and key title
0280-5316

ISBN

Language
English

Number of pages
1-58

Recipient's notes

Security classification