



Technische Universität Berlin
Fakultät IV - Elektrotechnik und Informatik
Nachrichtenübertragung

Master Thesis

OPTIMIZING LATENCY AND ENERGY EFFICIENCY IN WIRELESS SENSOR NETWORKS THROUGH A SENSOR NODE WITH WAKE-UP RECEIVER

EMPOWERING THE FUTURE OF OPEN-SOURCE INTERNET OF THINGS

Author: Tim Marc Schröder (382094)

First examiner: Prof. Dr. Thomas Sikora

Second examiner: Prof. Dr. Falko Dressler

Supervisor: Sara Reichert

Submitted on: Wednesday, 21. June 2023

Abstract

In an interconnected world of Wireless Sensor Networks (WSNs), striking a balance between latency and energy efficiency poses a significant dilemma. While Wake-up Receiver (WuR) technology presents a promising solution, it has predominantly remained confined to the realm of research. This master's thesis focuses on developing an open-source WuR system to address this limitation. The system was successfully designed, tested, and extensively documented, with all design files made available for reproduction, modification, and improvement. The performance of the system showcased a range of several meters with a wake-up delay under 1ms, using a fully passive front-end. By bridging the gap between research and practical implementation, this work makes a substantial contribution, offering a tangible pathway for the adoption of WuR technology in real-world applications.

All files used in this open source project can be found here:

<https://github.com/timschr/OpenWakeUpReciever>

Zusammenfassung

In einer vernetzten Welt von drahtlosen Sensornetzwerken (WSNs) stellt die Suche nach einem Gleichgewicht zwischen Latenz und Energieeffizienz ein bedeutendes Dilemma dar. Obwohl die Wake-up-Receiver (WuR)-Technologie eine vielversprechende Lösung bietet, ist sie noch überwiegend auf den Bereich der Forschung beschränkt. Diese Masterarbeit konzentriert sich darauf, ein Open-Source-WuR-System zu entwickeln, um diese Hürde zu überwinden. Das System wurde erfolgreich entworfen, getestet und umfassend dokumentiert, wobei alle Designdateien zur Reproduktion, Modifikation und Verbesserung zur Verfügung gestellt wurden. Das Systems ermöglicht eine Reichweite von mehreren Metern bei einem Aufwachverzögerung von weniger als 1ms unter Verwendung eines vollständig passiven Front-Ends. Durch die Überbrückung der Kluft zwischen Forschung und praktischer Umsetzung leistet diese Arbeit einen bedeutenden Beitrag und bietet einen greifbaren Weg für die Verwendung von WuR-Technologie in realen Anwendungen.

Alle Dateien, die in diesem Open-Source-Projekt verwendet werden, finden Sie hier:

<https://github.com/timschr/OpenWakeUpReciever>

Eidesstattliche Erklärung

Hiermit erkläre ich an Eides statt, dass ich die mit meinem Namen gekennzeichneten Teile für die vorliegende *Master Thesis*

OPTIMIZING LATENCY AND ENERGY EFFICIENCY IN WIRELESS SENSOR NETWORKS THROUGH A SENSOR NODE WITH WAKE-UP RECEIVER

selbstständig und eigenhändig sowie ohne unerlaubte fremde Hilfe und ausschließlich unter Verwendung der aufgeführten Quellen und Hilfsmittel angefertigt habe.

Berlin, 21. June 2023



TIM MARC SCHRÖDER (382094)

Contents

| | |
|---|-----------|
| 1 Introduction | 7 |
| 2 Literature review | 9 |
| 2.1 Introduction to Wireless Sensor Networks (WSNs) | 9 |
| 2.2 Improving Energy Efficiency in WSNs | 11 |
| 2.3 Sensor Node Design for WSNs | 12 |
| 2.4 Wake-Up Receiver Hardware | 14 |
| 2.4.1 Components of Wake-up Receivers | 14 |
| 2.4.2 Design Approaches and Hardware Architectures | 16 |
| 2.5 Wake-Up Receiver Software | 20 |
| 2.5.1 Comparator or Address Decoder IC | 20 |
| 2.5.2 Encoding Schemes | 21 |
| 2.5.3 Preamble filtering | 23 |
| 3 Problem identification | 25 |
| 4 Design requirements | 27 |
| 4.1 Soft requirements | 27 |
| 4.1.1 Simplicity in design and usage | 27 |
| 4.1.2 Affordability | 28 |
| 4.1.3 Modularity | 28 |
| 4.1.4 Usability | 29 |
| 4.2 Hard requirements | 29 |
| 4.2.1 Power Efficiency | 29 |
| 4.2.2 Communication range | 30 |
| 4.2.3 Low Latency | 31 |
| 5 Design and Development | 32 |
| 5.1 Reproduction | 32 |
| 5.1.1 Reproduction: Magno et al. (2016) | 33 |
| 5.1.2 Reproduction: Fromm et al. (2022) | 39 |
| 5.2 Design of own system | 42 |
| 5.2.1 Hardware Design | 42 |
| 5.2.2 Software Design | 50 |
| 6 Evaluation | 57 |
| 6.1 Performance | 57 |
| 6.1.1 Sensitivity | 57 |

| | |
|------------------------------------|-----------|
| 6.1.2 Range | 58 |
| 6.1.3 Energy Consumption | 63 |
| 6.1.4 Delay | 64 |
| 6.2 User-friendliness | 65 |
| 7 Conclusion and discussion | 67 |
| 7.1 Outlook | 68 |
| List of Tables | 69 |
| List of Figures | 70 |
| References | 72 |

1 Introduction

Wireless sensor networks (WSNs) consist of a large number of autonomous sensor nodes that work together to perform a variety of sensing, processing and communication tasks. Their capabilities span a wide range of applications, from promoting the development of smart cities, buildings and homes to enabling precision agriculture and health monitoring. Despite the potential of WSNs, their deployment and operation are hampered by certain limitations, most notably power constraints and latency. A typical sensor node in a WSN is powered by a small battery, which requires careful management of energy resources. Duty cycling, a strategy where nodes alternate between active and sleep states, has been widely used to save energy. However, the disadvantage of this energy-saving approach is increased latency or delay within the network. This poses a problem for applications that require real-time or near-real-time data transmission, highlighting the importance of achieving an optimal balance between energy efficiency and latency.

Wake-up receiver (WuR) technology offers a promising solution to this problem. A WuR is a small, ultra low power device that "listens" for certain signals and wakes up the main sensor node from sleep mode when needed. This approach enables a reduction in power consumption and latency, improving the balance between these two critical factors. Despite the potential of WuR technology, its implementation remains largely limited to research environments. While WuR has been the subject of extensive academic research for more than a decade, its transition to practical, real-world applications has been slow. A major obstacle to the widespread adoption of WuRs has been their inaccessibility to those outside the research community.

This work aims to bridge this gap by presenting an open-source sensor node design using WuR technology to make it accessible to a wider audience. An important contribution of this work is to not only explore the WuR technology theoretically and implement it practically, but also to document its design and operation in detail so that it can be easily replicated, modified and improved. The open-source nature of this project encourages community-driven development and paves the way for wider application of WuRs in WSNs. This project includes a comprehensive review of the existing literature, focusing primarily on WuR technology, its practical implementation and its potential to improve WSN performance. Based on this review, this thesis presents an open-source sensor node design using WuR technology, which is then implemented and thoroughly tested in real-world scenarios. By making this design freely available on the project's GitHub page, this work also serves as a valuable resource for hobbyists, developers and researchers interested in this field. All design files and software related to this project are available on this platform, ensuring transparency and easy access.

Ultimately, this work represents a significant contribution to the field of WSNs. By combining theoretical exploration, practical design, rigorous testing and open source

principles, it provides a viable solution to the problem of the trade-off between power and latency, driving further advances in WSN technology.

The remainder of this thesis is organized as follows: Chapter 2 provides an extensive literature review on the hardware and software architectures of wake-up receivers. In Chapter 3, we identify key challenges in open-source WuR development. These challenges guide the design requirements outlined in Chapter 4. Subsequently, Chapter 5 focuses on reproducing two prominent approaches from the literature, and presents our own open-source sensor node design incorporating WuR technology. Chapter 6 evaluates these developments through rigorous testing procedures and compares them with existing approaches. Lastly, Chapter 7 concludes the thesis and suggests future research directions.

2 Literature review

This literature review guides through essential aspects of wake-up receiver (WuR) technology in the context of wireless sensor networks (WSNs). WuRs have significant potential to revolutionise the energy efficiency of WSNs, so a deep understanding of their operation, design considerations and performance metrics is critical for this project. This chapter begins with a brief introduction to WSNs and sets the context for the emergence and benefits of WuR technology. We then move quickly to the core topic of the chapter: exploring the fundamentals of WuR technology and different approaches to implementation from research. Then it provides an overview of the basic components and then explores different design approaches and architectures. The impact of software and coding schemes on the efficiency and robustness of WuRs is also highlighted, considering the inherent interplay between hardware capabilities and software strategies. Towards the end of this chapter, the critical performance metrics that determine the success of a WuR system are discussed. This literature review forms the basis for the later chapters where we use these findings to guide the design and evaluation of a WuR-integrated sensor node.

2.1 Introduction to Wireless Sensor Networks (WSNs)

A wireless sensor network (WSN) is a collection of wireless-enabled sensing, computing, and communication components that enables an operator to measure, monitor, and respond to events and phenomena in a designated environment. These networks offer a flexible infrastructure that can be equipped with various instruments and observation tools, providing real-time data to improve situational awareness and enable swift reactions to changing conditions [1, p. 1–2]. WSNs are a type of network infrastructure that is designed to be low-cost and low-power. They typically consist of a large number of sensor nodes that are distributed throughout an environment and are able to autonomously perform sensing, processing, and communication tasks without human intervention [2]. The protocol stack used in WSNs consists of multiple layers that provide different functionalities such as physical layer (PHY) modulation, medium access control (MAC), network layer routing, and application layer interfaces. Due to the limited processing power and memory of sensor nodes, the protocol stack used in WSNs is designed to be lightweight and optimized for resource-constrained environments. This allows an efficient data transmission and processing, extending the lifespan of the sensor nodes and improving the overall performance of the network [3].

Wireless sensor networks (WSNs) are designed to be self-organizing and adaptive, allowing them to adjust to changes in the network topology or environmental conditions. These networks are typically formed in an ad-hoc manner, without relying on pre-existing

infrastructure or network administration. This flexibility enables WSNs to be rapidly deployed in a wide range of applications and environments, including those that are remote or difficult to access [1, p. 24–25].

The data collected by sensor nodes is usually transmitted to a base station or sink node, which aggregates and processes the information for further analysis. This allows for the implementation of various instruments and observation tools, providing real-time data to improve situational awareness [2]. There are different approaches to how to reduce traffic and thereby energy consumption in self-organised networks. One of the approaches is clustering, organizing nodes into clusters, where one node is designated as the cluster head (cf. fig. [1]).

This allows data to be aggregated and processed locally within a cluster, rather than being transmitted to a central node. This reduces the number of long-distance transmissions, which are often more energy-intensive, and can extend the lifespan of the sensor nodes. Additionally, cluster-based networks can improve the scalability and reliability of the network by reducing the amount of traffic and data that needs to be transmitted over the network, thereby reducing the risk of congestion and network failure [4].

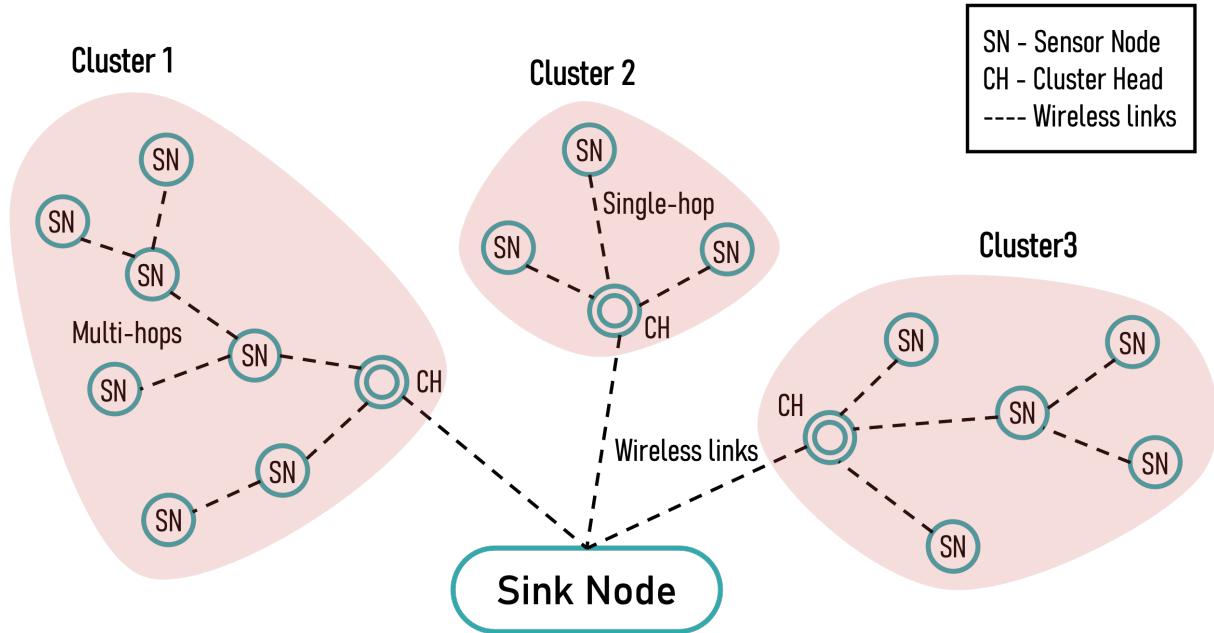


Figure 1: A Typical WSN deployment
[1, p. 16]

The wireless communication capabilities of the sensor nodes eliminate the need for cumbersome wiring, enabling the WSN to be deployed in remote or hard-to-reach locations. This makes WSNs an attractive option for collecting data in a wide range of environments. However, the design and development of WSNs still faces many challenges due to the limited resources of sensor nodes, such as limited battery life and low processing power. Security is also a significant concern, as they operate in a distributed and often hostile

environment. Mechanisms such as encryption, authentication, and intrusion detection are used to protect the network and its data [5].

Additionally, the limited resources of the sensor nodes, including battery life and processing power, make energy efficiency a crucial consideration. Minimizing energy consumption is necessary to prolong the lifespan of the sensor nodes and reduce maintenance costs as most sensor nodes run on batteries. At the same time, low latency is also essential in WSNs, particularly for real-time monitoring and control applications where quick and accurate data transmission is required. However, achieving low latency often comes at the expense of energy efficiency. For instance, if sensor data is transmitted more frequently, the energy used for transmission also increases. Therefore, finding the optimal balance between energy efficiency and latency is a challenging task.

2.2 Improving Energy Efficiency in WSNs

Wireless Sensor Networks (WSNs) operate in an environment where limited energy resources must be optimally utilised to ensure reliable and continuous operation. This problem, known as energy bottleneck, limits the lifetime and performance of the network. This energy bottleneck is particularly challenging because communication, especially wireless transmission, consumes the largest share of a sensor node's energy budget, which is amplified by the often significant distances between nodes in many WSN applications. A major problem in wireless sensor networks is idle listening, a condition in which a node constantly monitors the communication channel and awaits potential data transmissions even when no data is addressed to it. Since low latency is critical in data transmission, nodes must regularly perform this idle monitoring regardless of the amount of data traffic. Consequently, a significant amount of energy is consumed in this state even though the node is not actively processing or transmitting data - a challenge that exists even when traffic is low. Another major challenge arises from the so-called overhearing. This phenomenon occurs when a node accidentally receives packets from neighbouring nodes that are not addressed to it while listening in idle mode. In densely populated networks with high data traffic, overhearing can lead to a significant waste of energy, as nodes spend energy processing irrelevant information [6].

To address efficient power management, strategies have been introduced for example duty cycling. Duty cycling refers to a pattern in which sensor nodes periodically put to a sleep state to conserve energy. During the active state, a node is involved in tasks such as data acquisition, processing and communication, while in the sleep state it significantly reduces its energy consumption by turning off most functions and retaining only the minimal capabilities needed to wake up when needed.

However, duty cycling comes with some challenges, such as decision making regarding

sleep and activity intervals and synchronisation between nodes. The emerging use of WuRs has addressed some of these issues and improved the efficiency of duty cycling in WSNs. A WuR operates in a low-power listening mode and waits for a specific wake-up signal. Upon receiving this signal, the WuR activates the main radio or the processing unit of the sensor node. As illustrated in figure 2 data is exchanged after the WuRs interrupt and the successful reception of data packets is confirmed by the receiver with an acknowledge message [7].

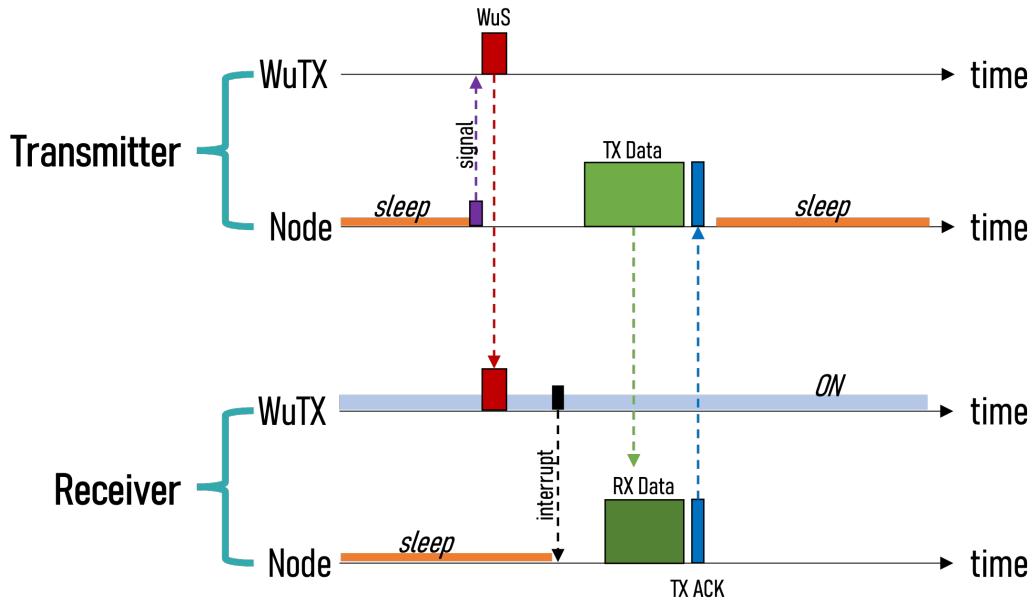


Figure 2: Functional Overview of Wake up Receiver

[6]

In this way, the main communication module of the sensor node can be completely switched off during the idle state, which significantly reduces power consumption and extends the life of the network. The wake-up signal itself usually consist of two parts. First a preamble to activate the decoder part of the wake-up receiver. The second part is a unique code or address that ensures selective wake-up. The main processor is only woken up if the address belongs to the own node to reduce unnecessary wake-ups and save energy.

2.3 Sensor Node Design for WSNs

Wireless Sensor Networks (WSNs) are a type of wireless network that consists of small, low-power, and inexpensive sensor nodes that are distributed in a physical environment to monitor and collect data. The data collected by the sensor nodes can be related to physical, chemical, biological, or environmental conditions, and can be used for a wide

range of applications [1] [8] [9].

The sensor node is a key component of a WSN and consists of several modules that work together to sense the physical environment, process data, communicate wirelessly, and provide power to the node. A typical sensor node can be divided into four main modules:

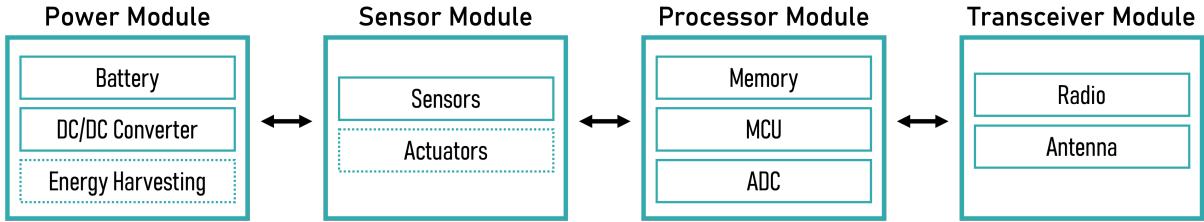


Figure 3: Overview of sensor node modules

[10]

The power source, a pivotal constituent of the sensor node, provides the indispensable energy required for its operational activities. This module, consisting potentially of a battery, solar cells, energy-harvesting modules, or a combination thereof, shapes the longevity and autonomy of the node's functioning [1] [11].

The sensor module interfaces with the surrounding environment, detecting and converting physical, chemical, or biological parameters into electrical signals. The sensor suite may encapsulate a wide range of sensors, including but not limited to, temperature, humidity, pressure, motion, and light detectors.

In the processing module, data derived from the sensor module is managed and processed. This stage involves various operations, such as data filtering, compression, encryption, and decision-making. The typical constituents of a processor module encompass a microcontroller, memory, and supplementary components necessary for task execution.

The transceiver module oversees wireless communication between the sensor node and other constituents of the network or a gateway. Equipped with a radio frequency (RF) transceiver, an antenna, and additional necessary components, the transceiver module facilitates the transmission of sensor-derived data or the reception of data from network peers.

Additional components such as actuators, analog to digital converters (ADC) and monitors may be added to the sensor node depending on the application. Actuators can be used to control and manipulate the physical environment, such as turning on a fan or a pump, while ADCs can be used to convert analog sensor signals to digital form for processing by the processor module. Monitors can be used to provide feedback on the status of the sensor node, such as the remaining battery life or the quality of the wireless connection [12].

2.4 Wake-Up Receiver Hardware

In this thesis, the wake-up receiver (WuR) has been till now explored as a theoretical concept that enables a sensor node to remain in deep sleep mode until a message is available for the corresponding node. The previous chapter delved into the advantages and potential implementations of WuRs within wireless sensor networks (WSNs). The focus now shifts to examining the hardware-level operations of wake-up receivers, encompassing their components and the various research approaches in the field.

In the literature review for this work, some practical boundaries were drawn to achieve the overall objective: the development of an accessible and open-source wake-up receiver. This focus dictates that the investigation concentrates on approaches that use commercial off-the-shelf components (COTS). The review does not extend to methods that use custom CMOS integrated circuits, which would be a significant hurdle for most developers due to advanced manufacturing requirements and costs.

These limitations streamline the focus of the literature review and ensure that the analyses and discussions are relevant and applicable to the open source spirit of the project. However, they also exclude certain solutions that, while innovative or effective, do not align with the overall goals of the project.

2.4.1 Components of Wake-up Receivers

This section provides an overview of the key components that constitute a wake-up receiver. It highlights their functions, discusses commercially available options, and evaluates the trade-offs for each component in terms of performance, power consumption, cost, and complexity.

Antenna

The antenna serves as a crucial component in the wake-up receiver, responsible for receiving the radio frequency (RF) signal transmitted by the wake-up transmitter. It converts the RF signal into an electrical signal to be processed by the subsequent stages of the wake-up receiver. Some approaches share an antenna with the main radio, enabling a more compact and cost-effective design. In such cases, an RF switch or a circulator is required to route the signals between the wake-up receiver and the main radio without interference. Common commercially available antennas include monopole, dipole, and patch antennas, each with their unique characteristics and suitability for specific applications. The choice of antenna is influenced by factors such as operating frequency, gain, radiation pattern, polarization, and form factor. Considerations for antenna selection should also include the environment in which the wireless sensor network operates and potential interference sources.

RF front-end

The RF front-end amplifies and filters the received signal from the antenna, preparing it for further processing. Most approaches use minimal or simplistic RF front-ends to conserve energy, as power consumption is the primary bottleneck in wake-up receivers. The key components in the RF front-end include low noise amplifiers (LNAs), mixers, and filters. In energy-conscious designs, the RF front-end may consist of only a basic LNA or passive components, and in some cases the RF front-end is omitted altogether, as in [13]. Filters, such as bandpass filters, can still be employed to reduce noise and ensure accurate detection of the wake-up signal. The selection of RF front-end components should focus on minimizing power consumption while maintaining adequate sensitivity and selectivity.

Detector

The detector plays a vital role in identifying the presence of a wake-up signal within the received signal. Envelope detectors are widely used in wake-up receivers due to their simplicity and low power consumption. An envelope detector works by extracting the amplitude envelope of the received RF signal, typically using a Schottky diode followed by a low-pass filter. The Schottky diode is crucial for its fast switching characteristics and low forward voltage drop, which contribute to the detector's sensitivity and energy efficiency. Another type of detector is the correlation-based detector, which typically provides better performance at the cost of increased complexity. However, due to the focus on energy efficiency, the envelope detector is more commonly utilized in wake-up receiver designs.

Decoder

Once the wake-up signal is detected, the decoder extracts the address information embedded in the signal to determine if the message is intended for the specific sensor node. This may involve simple thresholding or more complex digital signal processing techniques, such as demodulation and decoding algorithms. Commercially available microcontrollers, field-programmable gate arrays (FPGAs), or application-specific integrated circuits (ASICs) can be employed for decoding tasks. The choice of decoder should consider factors such as processing speed, power consumption, and integration with other components.

Control Logic

The control logic oversees the overall operation of the wake-up receiver, including power management, decision-making, and interfacing with the main sensor node. Commercially available microcontrollers are commonly used for this purpose, as they offer flexibility in programming and can be easily integrated with other components. Advanced power management techniques can be implemented in the control logic to further reduce power consumption, ensuring optimal energy efficiency in the wake-up receiver.

Table 1: Overview of off-the-shelf component Wake-up receiver literature

| Year | Author | Mod. | Freq. | Signal Detection | RF Front-end | Address det. |
|------|--------------------|------|-------|------------------|--------------|--------------|
| 2010 | Gamm et al [14] | OOK | 868 | ANT, MN | ED | AS |
| 2013 | Oller et al [15] | OOK | 868 | ANT, MN | ED | AS |
| 2016 | Bdiri et al [7] | OOK | 868 | ANT, MN | ED, LFA | AS |
| 2016 | Magno et al [13] | OOK | 868 | ANT, MN | ED | COM, ULP |
| 2018 | Woias et al [16] | OOK | 315 | ANT, MN | ED, LFA | AS |
| 2020 | Galante et al [17] | OOK | 868 | ANT, MN | ED | AS |
| 2021 | Fromm et al [18] | OOK | 868 | ANT, MN | ED | - |
| 2022 | Fromm et al [19] | OOK | 868 | ANT, MN, SAW | LNA, ED, LFA | COM |

Data not available is marked with '-'

Mod. Modulation scheme, **Freq.** Carrier Frequency, **ANT** Antenna, **MN** Matching Network, **SAW** Surface Acoustic Wave filter, **ED** envelope detection, **LFA** low-frequency amplifier, **LNA** low-noise amplifier, **AS** AS393x, **COM** comparator, **ULP** ultra low power processor

Table 2: Performance of off-the-shelf component Wake-up receiver literature

| Year | Author | Power [μW] | Sensitivity [dB] | Range [m] |
|------|--------------------|-------------------------|------------------|-----------|
| 2010 | Gamm et al [14] | 8.3 | -53 | 37.8 |
| 2013 | Oller et al [15] | - | - | - |
| 2016 | Bdiri et al [7] | 7.5 | -60 | 82 |
| 2016 | Magno et al [13] | 1.2 | -55 | 50 |
| 2018 | Woias et al [16] | 7.4 | -63 | - |
| 2020 | Galante et al [17] | - | -50 | - |
| 2021 | Fromm et al [18] | 4.2 | -55 | 10 |
| 2022 | Fromm et al [19] | 14.2 | -80 | - |

Data not available is marked with '-'

2.4.2 Design Approaches and Hardware Architectures

The most common WuR structure (cf. Figure 4) consists of an antenna designed for the specific frequency, followed by a matching network that optimizes energy transfer between the antenna and the wake-up receiver stages. Envelope detection is employed to extract the amplitude envelope of the received RF signal, and a low-frequency amplifier boosts the detected signal. Finally, a comparator converts the amplified envelope signal into a digital format for further processing and decision-making. This section will focus on various envelope detection-based approaches to design wake-up receivers, providing an organized structure for a comprehensive understanding of their distinctions, advantages, and limitations.

Antenna switch

When there is a wake-up receiver besides the main radio, there are two traces with data signals received by antennas that propagate through the sensor node. When both radios are based on the same carrier frequency, the same antenna may be used. It must be noted

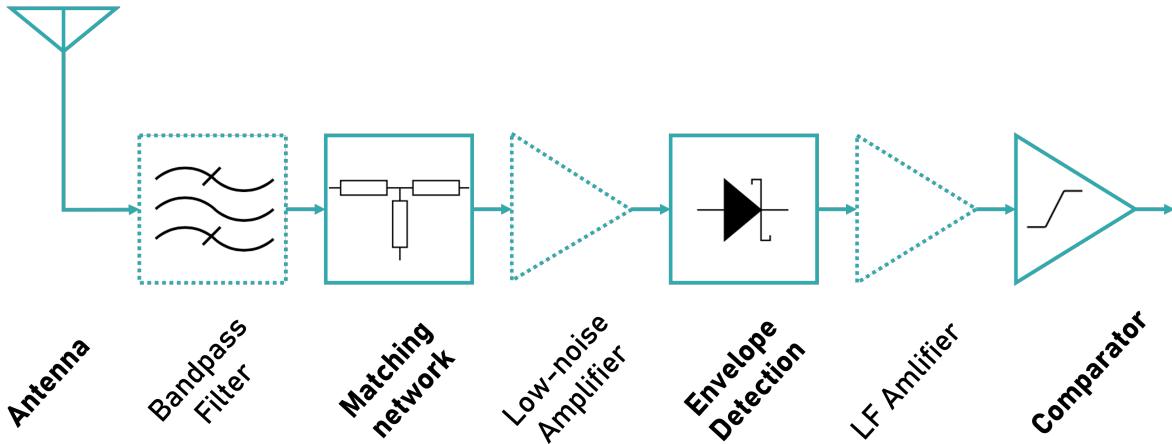


Figure 4: WuR structure

[19]

that these traces must be separated from each other to ensure the best possible sensitivity of the system. For this a HF switching IC can be used like the ADG918 in [14].

Matching Network

Matching networks in wake-up receiver design not only ensure efficient energy transfer between the antenna and subsequent stages but also account for issues like variations in component impedance due to manufacturing conditions and signal-induced changes in parasitic resistance. Careful design of PCB traces and transmission lines, as well as precise impedance measurements using a network analyzer, help in determining an appropriate matching network. A quasi-optimal matching network is designed considering input signal power levels (e.g., less than -40 dBm), and component trimming may be necessary to overcome matching errors. Verifying proper impedance matching can be done by measuring the reflection coefficient (S_{11}) at the wake-up receiver input [7].

The most common matching network setup is an LC-circuit with the capacitor connected to ground. (i.e. [13], [15], [16]). Research [16] demonstrated that having the inductor connected to ground is less efficient. Besides the L-shaped, also T- [17]/and π -shaped [18] matching networks are found in literature. Matching networks in wake-up receiver design must also account for discrepancies between simulations and real-world implementations. In most papers, the components of the matching networks need to be adapted due to imperfect models of diodes and comparators, PCB effects, and other side-effects during board development, such as antenna connectors and soldering. It is crucial to consider these factors when designing and verifying matching networks to achieve optimal performance in real-world applications [13], [20].

Low-noise Amplifier

WuRs that use only passive components consume the least energy, but reach the limits of sensitivity. Even if the passive system is perfectly optimized, the sensitivity is limited e.g.

by the sensitivity of the diodes of the envelope detector. To increase the signal to noise ratio before envelope detection, a low-noise amplifier (LNA) can be utilized. This means that a longer range can be achieved through increased energy expense.

The approaches [19] and [20] have increased the energetic efficiency by duty cycling the WuRs LNA stages, so energy hungry components are less prominent. This means that the WuR is only periodically put into an active state. In [19] this was implemented by the commercial off-the-shelf LNA MAX2640 with a 15dB gain at 17.5mW. The authors of paper [20], on the other hand, implements its own LNA with bipolar junction transistors. This was tuned exactly for the application and reduces consumption to 2.9mW for a 12dB gain. Nevertheless both cases show that duty cycled amplification of the signal can greatly improve sensitivity while keeping energy consumption at a tolerable level.

Envelope detector

Envelope detection is an important stage in wake-up receiver design, responsible for extracting the amplitude envelope of the received RF signal. In recent publications [7][13][14][18][19], the Greinacher voltage doubler configuration has been the most common approach for implementing an envelope detector circuit. This configuration consists of two diodes and two capacitors to alternately charge the capacitors during each half-cycle of the input RF signal (cf. Figure 5). By connecting the capacitors in series, the output voltage is effectively doubled, resulting in a rectified and amplified amplitude envelope of the input signal.

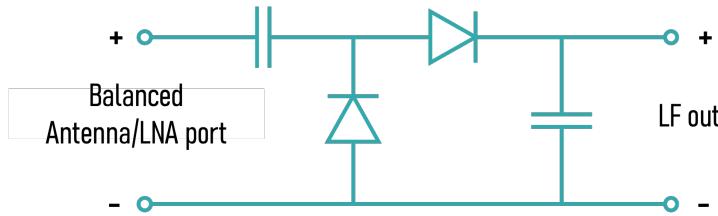


Figure 5: Greinacher Voltage Doubler

[21]

When a low noise amplifier (LNA) is used before the detector, an additional matching network becomes necessary to match the 50 ohm output impedance of the LNA to the envelope detector's load impedance. This ensures efficient energy transfer between the stages and maintains the overall performance of the wake-up receiver. The HSMS-2852 diode has been widely used in many publications [7][13][14] for envelope detection. However, due to its unavailability, the SMS7630-006LF from Skyworks Solutions Inc. has emerged as a viable alternative. With very similar properties to the HSMS-2852, the SMS7630-006LF has been successfully utilized by [18], demonstrating its effectiveness as a suitable replacement in envelope detector designs.

LF Amplifier

When developing wake-up receivers (WuRx), one of the biggest challenges is to achieve adequate sensitivity. This sensitivity can be compromised if a decoder such as AS3933 or a comparator is used for digitisation and connected directly to the detector. As a solution, several researchers, including [7], [16] and [19], propose the installation of an amplifier stage between the detector and the digitising component.

Two notable examples of this approach are described in studies [16] and [19]. The former uses a two-stage low-frequency (LF) transistor amplifier to improve the sensitivity of its WuRx design, while the latter relies on a commercial TSV6391A two-stage amplifier to achieve a similar effect.

In addition, [7] takes a different, more power-conscious approach by discussing the use of duty-cycling amplifiers that operate only during periods of active communication, thereby minimising power consumption during idle channel monitoring periods. Although this is a valid strategy, implementing such an approach is not without problems. Commercially available low-power amplifiers typically have a high input offset voltage VOS, a low slew rate and a low gain-bandwidth product (GBP). These factors make them inadequate for boosting an 18-kHz signal voltage level from the *mV* range to a detectable voltage level, so more powerful devices are required. In response, three stages of COTS MAX4461T amplifiers were successfully used in [7].

Decoding

Decoding is an important step in the operation of a wake-up receiver (WuR). It involves converting the incoming signal from its encoded form back into a format that can be easily understood and used. Depending on the design of the WuR, this process may include aspects such as signal amplification, address detection and the triggering of specific interrupts that cause the main system to 'wake up' from a low power state. This decoding process can be done with an address decoder such as the AS3933 or with a comparator and an ultra-low power (ULP) microcontroller.

Address decoders such as the AS3933 and its predecessor AS3932 are designed for this task and offer high sensitivity and built-in pattern recognition functions. For example, the AS3933 can generate an interrupt to activate the microcontroller unit when it decodes a valid wake-up call. It also has a built-in correlator to detect Manchester encoding, which is a common method of encoding digital data. The AS3933 has a sensitivity limit of $80\mu VRMS$, which is comparable to that of the AS3932. [7]

Comparators, on the other hand, simply compare the incoming signal with a reference voltage and give a binary output. They usually consume less power and have shorter delay times than AS decoders, but require an additional ultra-low power (ULP) microcontroller for address decoding. In the system described by [13], for example, Microchip's PIC12LF1552 microcontroller was used, which consumes only $40nW$ at 2 V when idle. Three different

comparators were tested in their setup, with the LPV7215 comparator having the highest sensitivity, a short delay time of 3us and a power consumption of $1.2\mu W$. The PIC microcontroller does include an inherent comparator, however, it represents a less appealing alternative due to its high consumption of a minimum of $90\mu W$. This is predominantly because it must continuously stay powered on owing to its extended activation duration [22].

2.5 Wake-Up Receiver Software

Following a comprehensive examination of the hardware components and design strategies of Wake-Up Receivers (WuR), the discussion now turns towards the software architecture that drives these systems. Although the software aspects of a WuR are critical in enhancing robustness and energy efficiency, the focus of research to date has clearly been on hardware development. There are only a few publications that deal exclusively with optimising the software of WuR systems. Nevertheless, this section will gather ideas and discuss opportunities in WuR software architecture.

The software is responsible for key functions such as minimizing computational complexity, managing power consumption, and coordinating low-power communication protocols. A WuR system typically utilizes efficient software algorithms, sleep modes, and interrupt-driven designs to manage these functions.

The encoding scheme employed to transmit data to the WuR is a substantial consideration. The scheme chosen significantly affects the reliability and energy efficiency of the communication. This discussion includes an overview and comparison of different encoding schemes. The choice of modulation techniques is a significant aspect, with On-Off Keying (OOK) being the most commonly adopted due to its simplicity. However, OOK's sensitivity to noise and the energy drain caused by the necessity of a prolonged high-signal preamble pose challenges. Furthermore, decoding the received signal is another critical task handled by the software. The decoding methods employed, including the use of comparators and address recognizers, can significantly influence the performance of the WuR. Therefore, exploring various decoding techniques, their energy consumption, and their contribution to the WuR's sensitivity forms an integral part of the software architecture literature review.

2.5.1 Comparator or Address Decoder IC

From the survey of the literature depicted in table 2 it can be observed that the decoder portion of WuR in research papers is predominantly split between the use of the AS3933 chip and designs employing a comparator plus an Ultra-Low-Power microcontroller unit

(ULP-MCU). These two solutions offer their unique benefits and constraints, making them suitable for different scenarios.

A considerable portion of the reviewed papers utilize the AS3933 chip. It operates in a range of 15-150kHz, supports Manchester encoded data, and can work with 16-bit or 32-bit address data (with an option for doubling). It can also support wake-up operations without specific pattern detection. The AS3933 chip offers the advantage of convenience, as it incorporates most functionalities required for WuR operations, leading to energy-efficient designs.

Nevertheless, the use of a specialized address decoder IC like the AS3933 presents certain limitations, notably in the areas of data rate and the chosen coding scheme. These limitations stem from the inherent design specifications of the chip, which may not align with the unique requirements of all applications. An alternative solution involves using a comparator coupled with an ULP-MCU, which offers substantial flexibility. This approach permits developers to implement a coding scheme or preamble detection method that best suits the application's needs. This level of flexibility, however, necessitates a more substantial software development effort, as all features must be manually implemented. This approach might be beneficial if there is a need for greater flexibility in encoding and preamble detection, and the research literature offers numerous innovative concepts that merit consideration in such cases.

Given the scope and aspirations of this project, it becomes critical to identify methods and techniques that not only streamline the implementation process but also improve overall system performance. Hence, the subsequent paragraphs will focus on delving into various software-based strategies to refine and optimize WuR operation, thereby addressing the inherent challenges and exploiting the opportunities presented by this exciting technology.

2.5.2 Encoding Schemes

In the context of wireless communication, an encoding scheme is critical for maintaining data integrity. Errors can occur due to interference, signal attenuation and other factors. By adding redundancy in the form of extra bits to the data, coding schemes allow for more robust communication, enabling error detection and even error correction at the receiving end [23].

Non-Return to Zero (NRZ) is one of the simplest forms of an encoding scheme. With NRZ, a binary "1" is represented by one level (usually a positive voltage) and a binary "0" is represented by another level (usually a negative voltage or zero). (see figure 7)

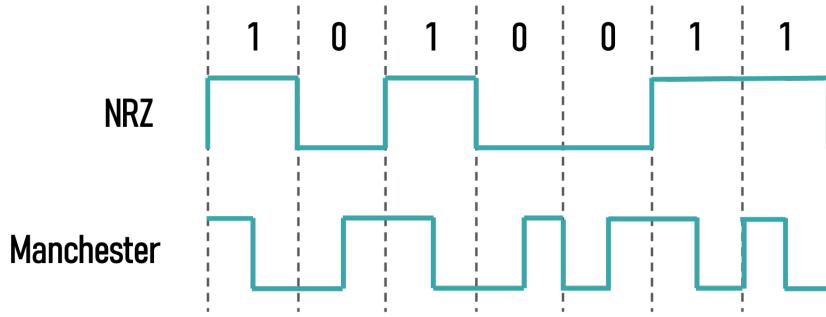


Figure 6: Comparison NRZ to Manchester encoding

Another form of coding scheme is Manchester coding which represents bits by transitions from high to low for a "0" or from low to high for a "1", as shown in figure 7. This method allows for easy clock recovery and error detection, as there is a transition in the middle of each bit period, making it widely used in various forms of data communication. Manchester coding is utilized by the address decoder AS3933 that is widely used among the present papers.

In the context of Wake-up Radio (WuR), channel coding is essential for enhancing sensitivity and promoting energy efficiency. As pointed out in [24], Minimum Energy Coding (MEC) emerges as a highly suitable approach. This technique transforms each k -bit block of data into an n -bit codeword (illustrated in table 3), and it is designed to minimize the energy required to transmit each bit by reducing the time the transmitter stays in the high-power "on" state. As highlighted by [24], employing this method results in a substantial energy saving of 42% compared to transmissions of uncoded signals.

| Source symbols (S_k) | Bit code | Mapped symbols (M_n) | Bit code |
|--------------------------|----------|--------------------------|-----------|
| S_0 | 00..00 | M_0 | 0000..000 |
| S_1 | 00..01 | M_1 | 0000..001 |
| ... | ... | ... | ... |
| S_{n-1} | 11..10 | M_{n-1} | 0100..000 |
| S_n | 11..11 | M_n | 1000..000 |

Table 3: Mapping of source symbols to mapped symbols in Minimum Energy Coding

[?]

Research paper [25] proposes an enhancement to the Minimum Energy Coding called Early Shutdown. In this variant, the ultra-low-power microcontroller (ULP-MCU) is powered off immediately after detecting the first '1' (high) signal, given that only a '1' is anticipated in the sequence under Minimum Energy coding. By reducing the duration that the ULP-MCU remains active, early shutdown further trims the total energy consumption, contributing to more efficient operation.

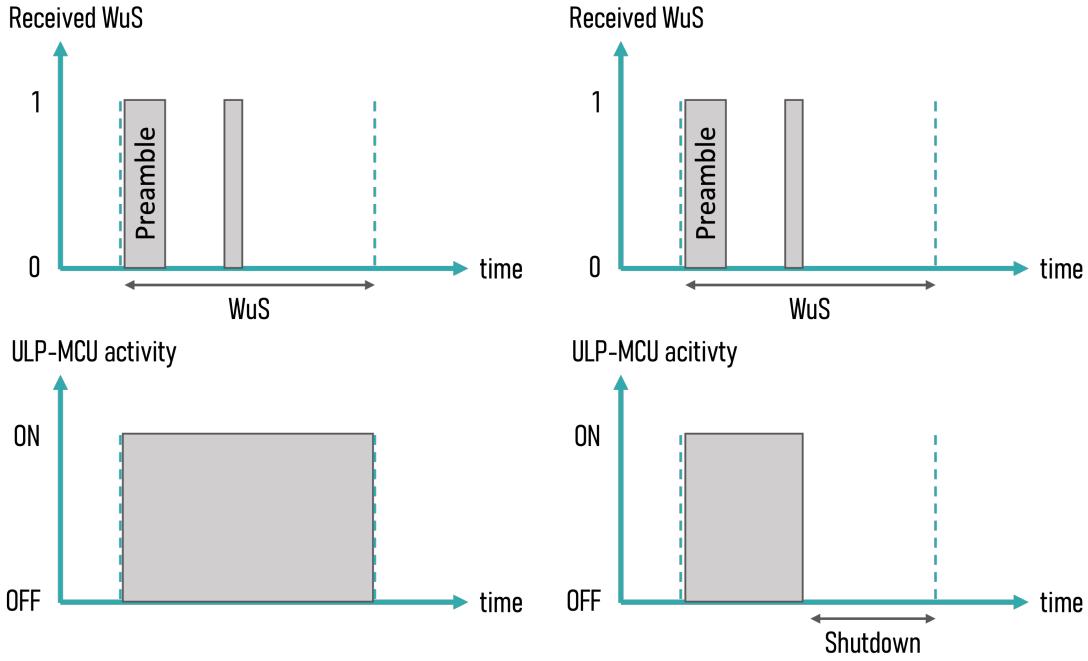


Figure 7: Minimum Energy Coding (left) and Minimum Energy Coding with early shutdown (right)

[25]

2.5.3 Preamble filtering

A Wake-Up Signal (WuS) is typically composed of two fields: the preamble and the destination address field. The preamble primarily serves to wake up the decoder e.g. a ULP-MCU. Once the ULP-MCU receives an address that matches its destination, it triggers an interrupt to awaken the main node of the system.

The novel technique proposed in [24] adds an additional step to this process. Instead of directly proceeding to decode the address upon waking up, the ULP-MCU first filters the preamble that it has received after the wakeup delay of t_d . It measures the duration of the preamble (t_{pr}) and checks whether it falls within a predefined interval, which corresponds to the duration of a valid preamble. If the preamble duration is within this range, the ULP-MCU proceeds to decode the address. If not, the ULP-MCU is switched off (see figure [8]).

[24] states that this approach of preamble filtering significantly reduces the incidence of false wake-ups, which often occur due to the presence of other signals within the 868 MHz band. There could be external signals with different preamble durations, and without the filtering step, these could potentially be mistaken for valid WuS if the address happens to match the destination node. This would lead to unnecessary wake-ups of the main node, resulting in wasted energy. By incorporating the preamble filtering technique, these false wake-ups can be effectively mitigated, enhancing the overall energy efficiency of the system.

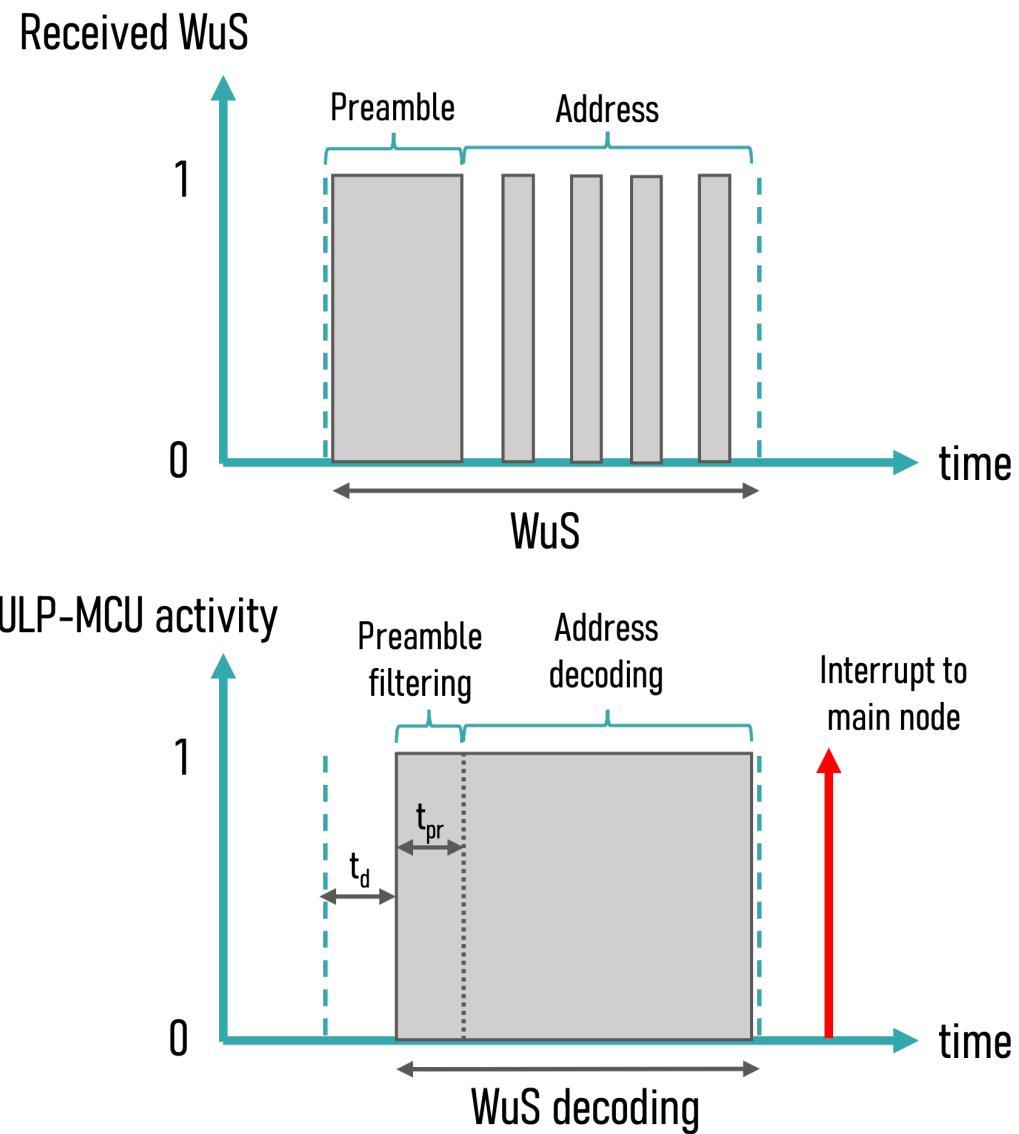


Figure 8: Preamble Filtering

[24]

3 Problem identification

The literature review has shown that there are already many different approaches to WuRs made from off-the-shelf components in research papers. So why is there a need for an open-source project dedicated to this topic? Why does it require a single master's student to delve into a topic that research teams at universities worldwide have been working on for many years?

The necessity of an open-source wake-up receiver project is based on two reasons:

1. Information from research papers is not sufficient to properly reproduce, use, and modify them for individual purposes.
2. Individuals can utilize WuR technology for their own projects, adapt and improve it, potentially making the world slightly better.

Firstly, there are several points that were noticed during the literature research that make the reproduction of the research difficult, in some cases even impossible. These points will be summarized in the following.

Many existing publications on wake-up receivers do not provide comprehensive design information, making it difficult for researchers and developers to replicate or build upon their work. In contrast to open-source projects where all design files are readily available, literature often falls short in providing complete and detailed documentation. Though research papers are usually accessible, they may not include schematics, or may only provide partial schematics with components that are not named or have unspecified values. For example, it is not uncommon for a paper to omit information about specific components, such as the type of diode or the value of a capacitor. Similarly, critical details about the choice of parts for the PCB, the PCB manufacturer, and the PCB material (e.g., FR-4) are often missing. Information about PCB trace width and layout is rarely provided, and in some cases, only photos of the finished PCB are available, which are insufficient for accurate replication.

Replicating existing wake-up receiver designs often requires advanced electronics design knowledge and skills, which can pose a significant barrier to entry for many developers interested in building their own wake-up receivers. One crucial aspect of wake-up receiver design is the need for perfectly tuned matching networks to ensure optimal performance, range, and efficiency. The process of fine-tuning matching networks typically requires a vector network analyzer (VNA), a specialized piece of laboratory equipment that is not easily accessible to many developers.

In addition to the need for specialized equipment, the tuning of matching networks can be sensitive to properties of the printed circuit board (PCB) that may differ between manufacturers. For example, the trace width on one type of PCB may work well for a

specific matching network, but the same trace width might not be suitable for another type of PCB from a different manufacturer. This variability in PCB properties introduces another layer of complexity and increases the difficulty of replication. Furthermore, discrepancies can also arise between the properties of components on the PCB and their respective datasheets. Real-world component behavior can deviate from their specifications to some extent, which can affect the performance of the wake-up receiver design. Additionally, it is crucial to approach simulation results with caution, as there can be significant differences between simulated hardware and real-world implementation. Many literature contributions present designs that use components that are no longer available, difficult to obtain, or too expensive for home-built sensor nodes. This issue makes replication of existing designs difficult and highlights the need for low-cost and readily available components in the development of wake-up receivers.

Unfortunately, these problems are not a special case in the wake-up receiver field, but rather the rule in hardware-based research. Researchers may not publish fully documented hardware and software documents alongside their research papers, similar to open source projects, for a variety of reasons. Intellectual property concerns often play a significant role, as institutions or organizations may want to protect their rights to the technology developed during the research process. Limited resources, such as time and funding, can also prevent researchers from investing in thorough documentation. Additionally, the competitive nature of academia often encourages researchers to prioritize publishing results rather than creating accessible resources.

Lack of incentives is another factor, as academic promotions and funding opportunities usually focus on publications, not on the accessibility of the developed hardware or software. The specialized and complex nature of some research may lead researchers to believe that only a few experts could understand or utilize their work, thus not prioritizing sharing. Confidentiality and security concerns may also prevent open sharing, particularly when research involves sensitive data or has national security implications.

Despite these barriers, there is a growing movement towards increased transparency, reproducibility, and open access in scientific research. This has led to the establishment of open-source repositories, data-sharing platforms, and open access journals that aim to improve collaboration and accessibility within the research community. By addressing these concerns and providing incentives for sharing, the scientific community can work towards a more open and collaborative research environment.

The above problems highlight the importance of developing an open-source, accessible, and well-documented wake-up receiver design. By overcoming these challenges, the research community and developers can benefit from a collaborative and innovative environment. This validates the goal of this master thesis to prepare the topic of wake-up receiver for the general public and to put it into a format that allows many people to use this

technology. The following chapter 4 outlines specific design criteria for the open-source wake-up receiver to overcome the limitations discussed in this chapter and pave the way for a more accessible and cost-effective solution.

4 Design requirements

The next section of this thesis describes the essential criteria for developing an effective, accessible, and practical wake-up receiver (WuR). After carefully identifying the challenges in the previous chapters, this is an opportunity to provide guidelines that could effectively mitigate these problems. An optimal solution must appeal to a wide range of users, from hobbyists to professional researchers and developers. Thus, the goal goes beyond simply developing a functionally successful solution, as the design must be easy to understand, affordable, and practical for a variety of uses. Establishing comprehensive design requirements ensures that the project is consistent with the intended goals and balances multiple, potentially conflicting aspects. These are divided into soft and hard requirements. Soft requirements are elements like simplicity, affordability, and usability, which, while vital for the overall user experience, do not directly impact the technical performance of the WuR. Conversely, hard requirements encompass technical aspects such as energy efficiency, latency, and error rate, which fundamentally affect the WuR's functionality. The following highlights the exact requirements that the open source WuR project should meet and provides a detailed examination of these aspects. With clear, well-defined criteria, the stage is set to create an environment to innovation, collaboration and continuous improvement within the WuR space.

4.1 Soft requirements

4.1.1 Simplicity in design and usage

When developing an effective and accessible wake-up receiver (WuR) system, simplicity is an important consideration in both development and use. A straightforward, intuitive design of the wake-up receiver is desired, consistent with other requirements to accommodate users with varying levels of expertise. Minimizing circuit complexity, clear labeling, and easy-to-understand design principles can improve accessibility and adaptability and support troubleshooting and potential modifications. Also providing open-source design files in conjunction with detailed instructions for fabrication, ordering, and assembly encourages accessibility and ease of modification, which is essential for integrating user-specific elements such as self-designed sensor nodes.

In addition to design, simplicity should also carry through to the use phase. A seamless integration with existing systems, a user-friendly interface, and a manageable learning curve are critical to fostering broader adoption. Providing comprehensive support resources such as documentation, use cases, and an interactive platform for resolving issues, such as GitHub Issues, further simplifies usage and fosters a collaborative problem-solving environment.

4.1.2 Affordability

Affordability is another fundamental design requirement that significantly impacts the accessibility and adaptability of the wake-up receiver (WuR). The affordability commitment spans multiple aspects of the project - from the choice of design and simulation tools to the selection of components and production tools. The choice of design and simulation tools has a direct impact on the overall cost of reproducing the WuR. To ensure affordability, design should not rely on paid programs such as the Advanced Design System (ADS). Instead, free, open-source software tools that provide similar functionality will be used, reducing the costs associated with the design process.

To maintain cost efficiency, priority is given to the use of low-cost, off-the-shelf components. In addition, the design aims to avoid components with potential availability issues or those prone to large price increases due to market fluctuations. Ensuring long-term availability of components is key to maintaining a consistent, affordable production process, especially important for hobbyists or small developers who may not have the resources to procure components in large quantities to compensate for uncertainty. The choice of tooling for WuR fabrication also reflects a commitment to affordability. It is critical to minimize reliance on expensive instrumentation such as Vector Network Analyzers (VNAs) or oscilloscopes. In addition, the design avoids the need for specialized toolkits that are only needed for certain use cases, such as specialized microcontroller programmers. By limiting the toolset to the essentials, the project can maintain a lean production process, further lowering the barriers to entry.

4.1.3 Modularity

The modular system design ensures that the WuR is compatible with a wide range of sensors, actuators and microcontrollers, providing the flexibility to adapt to different applications. This flexibility allows developers to integrate the WuR into different projects. In addition, modularity allows for component interchangeability, which promotes design longevity and adaptability. If a particular component becomes unavailable or obsolete, it can be replaced without requiring a complete redesign, ensuring the sustainability of the WuR. In terms

of form factor, the WuR should be compact and integrate all necessary components on a single board that can serve as an extension for existing microcontrollers. This approach not only contributes to the portability of the end devices, but also complements the concept of modularity and enables easy integration and modification. A small, efficient form factor is consistent with the overall goal of creating an accessible, affordable and versatile WuR design.

4.1.4 Usability

The usability of the wake-up receiver project is an important soft requirement. To improve this aspect, it is important to follow an open source approach, which offers the double benefit of transparency and community participation. By making all design files freely available, interested parties can easily reproduce, modify, and improve the system. This openness not only increases the accessibility of the project, but also allows for collective improvements and sharing of ideas, encouraging innovation. In addition, a fully comprehensive guide to the entire manufacturing, ordering, and assembly process is another critical component of usability.

4.2 Hard requirements

4.2.1 Power Efficiency

Energy efficiency remains the cornerstone of wake-up receiver (WuR) operation. The inherent advantage of WuRs is their significantly lower power consumption compared to the main microcontroller unit (MCU). The essence of the low-power operation of WuRs is that they allow the primary system (often an MCU) to remain in low-power sleep mode for as long as possible. Consequently, unnecessary wake-ups or false alarms - where the MCU is unnecessarily woken up - must be avoided as they consume valuable energy. However, optimising power consumption in WuRs goes beyond managing the primary system's sleep cycle. Within the WuR design, every component selection and system configuration should be reviewed for its impact on power consumption. Active components in particular can contribute to higher power consumption. Certain design strategies, such as the proposal by [26] to perform duty-cycling of active amplifier stages, may prove beneficial. In this approach, the active amplifiers are switched off at regular intervals, effectively reducing their power consumption.

In the pursuit of energy efficiency, it is also important to consider the robustness of the WuR to noise and the error rate. Both false negatives - when the WuR fails to wake up when required - and false positives - when the WuR falsely triggers a wake-up -

have an impact on system performance and power consumption. False-negative messages affect system performance, while false-positive messages contribute to unnecessary energy consumption.

Therefore, the design of the WuR should include measures to minimise both types of errors. This may mean using more robust coding schemes or improving the sensitivity of the receiver to achieve a lower bit error rate (BER). These strategies contribute to a more reliable and energy-efficient WuR system. The energy efficiency of WuR systems therefore depends on a thoughtful balance between design decisions, careful component selection and strategic system configuration to maximise system uptime while minimising unnecessary energy expenditure.

4.2.2 Communication range

Communication range is a key requirement in the development of the wake-up receiver (WuR). Several interrelated factors - transmit power, modulation and receiver sensitivity - essentially determine the extent of the communication range. Transmit power is closely related to energy efficiency. Although one might be tempted to maximize transmit power within regulatory limits to achieve greater range, it is important to remember that higher transmit power inevitably leads to higher energy consumption. Considering the goal of a WuR to consume as little power as possible, an effective balance between transmit power and energy efficiency is required.

In terms of modulation, on-off keying (OOK) modulation has been referred to as the optimal choice for WuRs due to its simplicity (as discussed in section 2.4). Although less complex modulations such as OOK can be prone to errors due to interference, their use simplifies receiver design, making them advantageous for WuR systems. This is in contrast to techniques such as LoRa, where complex modulations allow for greater communication range but also require a more complicated receiver setup.

Receiver sensitivity is another critical factor, influenced primarily by component selection and the presence (or absence) of active amplifier stages in the WuR design. A well thought-out design can make all the difference in optimizing communication range without unduly burdening power consumption. This includes a perfectly tuned matching network or very energy efficient address matching.

Communication range often has a delicate tension with power consumption. The design challenge is to carefully negotiate this trade-off in order to maximize range without compromising energy efficiency or WuR affordability too much.

4.2.3 Low Latency

Latency plays an important role in wireless sensor networks (WSNs) and wake-up receiver (WuR) architectures, directly affecting system responsiveness and efficiency. In WSNs, where countless sensor nodes transmit data to a central processing node, low latency ensures accurate and timely transmission of information. This is essential for real-time applications such as industrial automation, environmental monitoring, or healthcare systems that rely on fast decision-making and response times. In the context of WuRs, latency becomes even more important because of the impact on power management. WuR systems are in mostly power-saving mode and wake up the main system only when a specific trigger, such as wake-up signal with node address, is detected. The wake-up latency describes the time from sending the wake-up signal to the point where the interrupt is sent to the MCU. This latency is caused by the fact that the wake-up signal is first demodulated and digitized and then an ultra low power processor or similar must check whether the wake-up signal is actually intended for its own node (e.g. address correlation). If there is an error in this last step, for example due to interference on the transmission channel, the MCU is unnecessarily woken up and energy is wasted.

Low latency, bandwidth, and power consumption are interrelated variables in wireless communications. Lower latency often requires greater bandwidth to enable faster data transmission. Bandwidth in this context refers to the data transmission capacity of a system in a given time period. Greater bandwidth can result in higher energy consumption, as higher data transfer rates are required. In addition, the coding scheme used for data transmission plays a crucial role. Robust coding, i.e., coding schemes that add redundancy to data to make it resistant to errors during transmission, usually reduces bandwidth because more bits are needed to transmit the same amount of data. When designing a system, the right balance must be struck between these factors, taking into account the specific requirements and constraints of the application.

5 Design and Development

Starting from the comprehensive analysis of wake-up receiver hardware and software architectures in chapter 2, a basic understanding of the existing trends, strategies and their implications was provided. Chapter 3 presented the derived challenges in the development of open source wake-up receivers (WuR) and provided an overview of the potential obstacles that need to be overcome for the effective development of a WuR. This formed the basis for the discussion in chapter 4 on design requirements, which defined the precise objectives for the project that followed.

In this fifth chapter, the aim is to translate these findings and goals into a feasible design for an open-source sensor node with integrated WuR technology. However, before starting to develop a new design, we will reproduce two different WuR approaches from the literature. Reproducing existing designs serves several important purposes. It facilitates a comprehensive understanding of the underlying principles of WuR design and their real-world applications. At the same time, it provides a performance benchmark against which the proposed design can be evaluated. In addition, the reproduction process can identify and overcome obstacles to the reproducibility of such a project. In this way, these hurdles can be overcome in our own development and reproducibility can be significantly increased, which is a central aspect of any open source project.

With these preliminary steps in mind, this chapter first explores the replication of two selected WuR designs, providing a solid foundation for the development of an innovative open-source sensor node design. Then, design decisions and the development process are explained for the development of our own WuR system.

5.1 Reproduction

This section will document the attempts to replicate existing research and verify its results. For this purpose, papers from Magno et al. [13] and Fromm et al. [19] were selected because they had the best documentation among the papers examined (cf. 2.4.2), allowing a comparison of active and passive frontends. This will facilitate the comparison and evaluation of a broad spectrum of wake-up receiver solutions (in the context of WuRs using off-the-shelf components and ISM band).

In Magno et al. (2016) [13], besides the classical WuR front-end of the Greinacher Voltage Doubler (cf. 5), only a sensitive comparator is used to digitize the signal and thus represents a simple setup. For address decoding an additional 8bit ultra low power microcontroller (ULP) of the PIC series is used. With only $1.2\mu W$ the design claims to reach a sensibility of $-55dB$ and a range of $50m$ in open field.

Fromm et al. (2022) [19], on the other hand, chooses a more complex approach for the

frontend with including active components, but promises significantly increased sensitivity and thus range. Compared to [13], a SAW bandpass filter is used to filter the input signal. Before the envelope detector (also Greinacher Voltage Doubler, same as [13]) a double LNA stage amplifies the signal. After that the signal is amplified again by a double amplifier stage before it is digitized by a comparator. The paper states that sensitivity gets as low as -80dB with a power consumption of $14.2\mu\text{W}$.

Due to the different characteristics of the WuR as well as its documentation, the methods of reproduction also differ to some extent. These will be described in the following.

5.1.1 Reproduction: Magno et al. (2016)

Before delving into the practical components, an examination from reproducibility perspective of the paper by Magno et al. [13] is worth discussing. The key takeaways from this study will be summarized and assessed in the following.

First of all, the frontends circuit schematic is provided and the authors describe most of components, only the resistor R1 and the schottky diode D3 were not specified. Additionally the way of hardware testing and design iteration was described, specifically the adjustment of the matching network. A wake-up interrupt is created using passive components, resulting in an adjustable delay that can be fine-tuned by altering the low-pass filter's components. This flexibility allows for customization based on specific requirements like the bit rate.

The sensitive component in the design is the comparator, which must be chosen with care based on three crucial criteria: power consumption, input offset voltage, and propagation delay. The power consumption should be minimal, with a quiescent current on the order of a few nanoamperes. The input offset voltage, V_{IO} , is the voltage difference required at the comparator inputs to toggle the comparator. A total of three comparators are compared with each other, those to WuR with 7, 22 and 50m range. Due to the set design requierments in chapter 4 and for better comparability with the Fromm et al. [19] only the comparator with the highest range will be considered.

Lastly, the assumptions about the difficulty of impedance matching in the transmission from the simulated to the circuit stated in the "Problem identification" section 3 are confirmed here. It is described that the simulated matching network on the prototype PCB had its minimum input reflection (S_{11} parameter) instead of the desired 868 MHz at 3.1GHz. These initial impressions and takeaways provide valuable insights into the design and serve as a foundation for further exploration and reproduction of the work.

Despite the promising hardware prospects, the lack of a detailed software description significantly hampers exact reproduction. Further complicating is the absence of certain critical values such as resistor R1. Basic parameters like the carrier frequency of 868MHz and bit rate of 1kHz are indeed mentioned, however, crucial information, such as the

coding method, is entirely missing. Due to these reasons, reproducing the entire concept only seems plausible through a series of reasoned estimates and thus may not provide significant added value. However, the process of replicating the WuR frontend—comprising of the matching network and envelope detection—offers valuable insights. These insights, derived from the reproduction attempt, will be discussed in the subsequent section.

Simulation

In order to confirm the basic assumptions of the paper, a simulation was first carried out with the program LTSpice. The schematic described in the paper [13] is reproduced in the simulation software (cf. Figure 9). The SPICE parameters of the diode HSMS-285c had to be taken from the data sheet [27]. Since this diode is not produced anymore, this is a good opportunity to compare the diode SMS7630 from Skyworks Solution, Inc. [28]. A SPICE model for the TLV7031 comparator is provided in the datasheet on the Texas Instruments website. The missing specification of R1 and the schottky diode D3 were determined iteratively by the simulation.

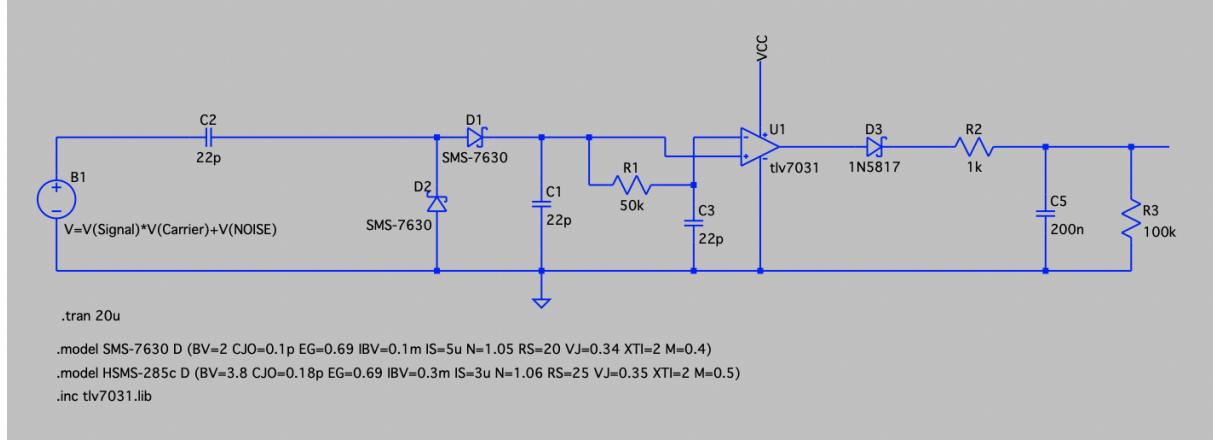


Figure 9: SPICE simulation of WuR frontend circuit of [13]

For the simulation, an 868Mhz carrier ($V(Carrier)$) signal was chosen as the input voltage, which was multiplied by a square wave signal of 125kHz ($V(Signal)$). This voltage simulates an OOK-modeled signal reaching the antenna with 25mV, seen in the top (cyan) part of figures [10] for the SMS7630 diode and [11] for HSMS-285C. In the middle part of the graphics the inverting and non-inverting inputs of the comparator are shown. The inverting input is thresholding the envelope detected signal that is connected to the non-inverting input. To avoid the need of a voltage devider, which would consume energy, a RC-filter is used as an adaptive threshold mechanism (cf. the middle part of figures [10] ND[11]). In the bottom part of the figures the digitilized signal at the comparators output is visualized. Noticeable is the $3\mu s$ operational delay that can be detected, which needs to be taken into account for the overall delay calculations. The digital output signal can be used for wake-up interrupts and address detection. There is barely a difference between

the two diode types, the SMS7630 gets a little more voltage out of the input signal 11mV compared to 10mV in this simulation. It must be said that this simulation was made *only* to understand the operation of the WuR frontend and is not a substitute for real efficiency tests.

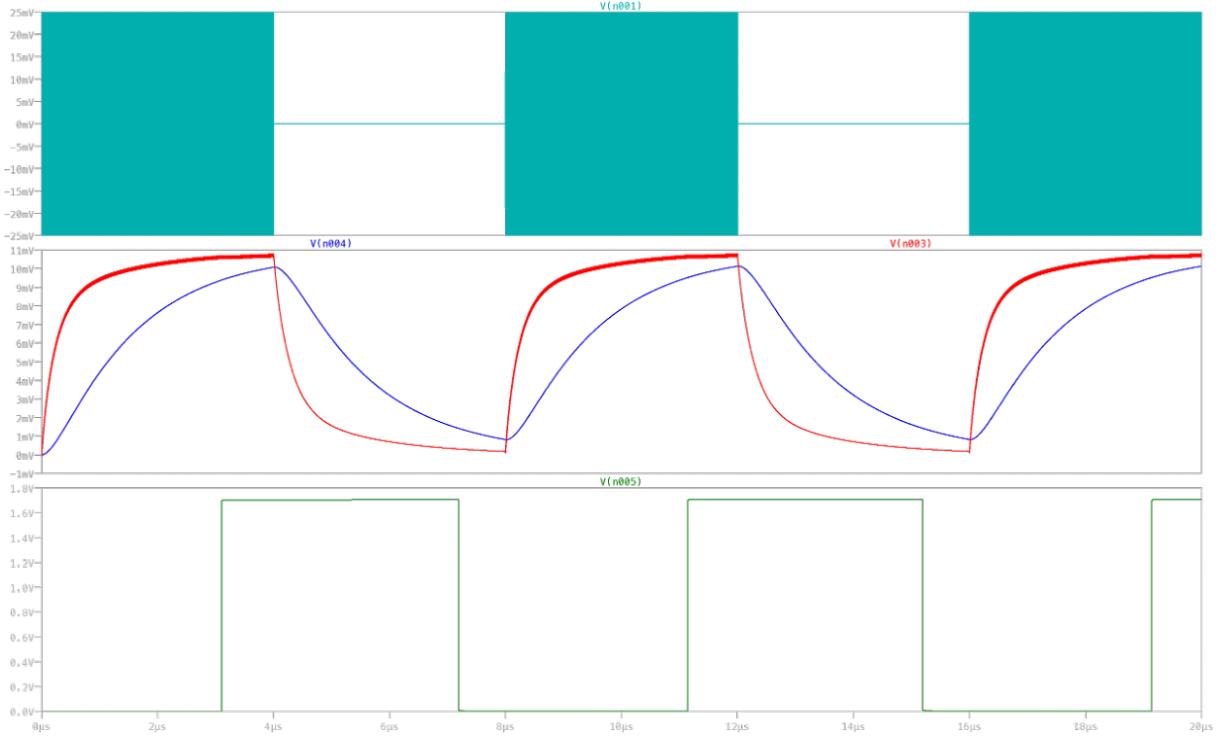


Figure 10: SPICE simulation results for SMS7630 as envelope detector. Comparator output signal (top/cyan), comparator input signals: IN+ (middle/blue) and IN- (middle/red), simulated OOK signal (bottom/green)

Prototyping PCB

The next step is now to produce a prototype board. For this purpose, a matching network must be built to maximize power transfer and minimize signal reflection. It ensures that the impedance of the 50Ω and the envelope detector are matched, which helps prevent signal loss and distortion. Theoretically the optimal matching network can be simulated with a program like ADS, but it is difficult to model diodes at specific frequencies and the impedance at $868MHz$ needs to be known. Therefore a prototyping PCB (see the schematic in figure 12 and the board in figure 13) is built to measure the Z-Parameters of the envelope detector under real-world circumstances. Therefore, the envelope detector circuit was built in the open source electronics design program KiCad and placed on a printed circuit board (PCB). In the process, there are a few things to consider.

First, the transmission lines (traces on the board) must be calibrated to 50Ω . Depending on the width, thickness and characteristics of the board (material and thickness of the dielectric) traces have a different impedance. The properties of the board can usually be found on the website of the board manufacturer, the thickness of the tracks can be

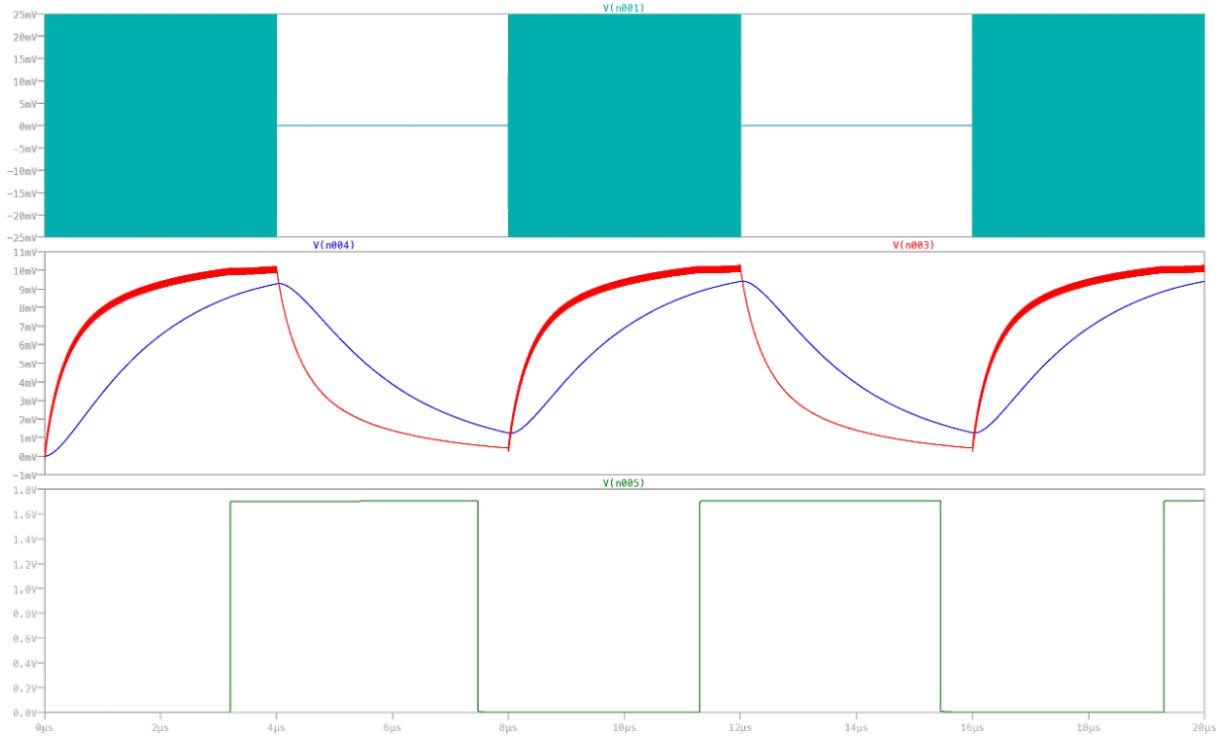


Figure 11: SPICE simulation results for HSMS-285C as envelope detector. Comparator output signal (top/cyan), comparator input signals: IN+ (middle/blue) and IN- (middle/red), simulated OOK signal (bottom/green)

determined with the calculator tools of KiCad, in our case 0.29337mm . This value is valid for the material JLC04161H-7628 of the PCB manufacturer JLCPCB. Furthermore, for optimal impedance control, the circuit must be implemented on a 4-layer PCB, since here the distance between the 1st and 2nd layer is significantly smaller. Finally it has to be ensured that enough vias (connections between the PCB layers) are placed around the traces (also known as via stitching) to minimize signal degradation and interference by providing a low-impedance path between different layers, ensuring proper grounding and reducing electromagnetic radiation. In order to connect the prototyping board to measuring devices like a vector network analyzer, a SMA connector is attached to each end. This board can be used to determine the Z-parameters needed to develop a suitable matching network. In order to test this matching network immediately, a second board is designed, which extends the envelope detector by the matching network. The circuit is shown in figure [14], two row elements and two parallel elements are used to be flexible in the design of the network. The capacitors and inductors are placeholders and have no meaning. Components selected for the matching will be placed on these footprints later, it is also possible that less than four components are needed for this. This can be done by shorting the row elements with 0Ω resistors, parallel elements can be left out as they connect to the ground plane. For the board (cf. [15]) the same characteristic are used as for the Z-parameter board.

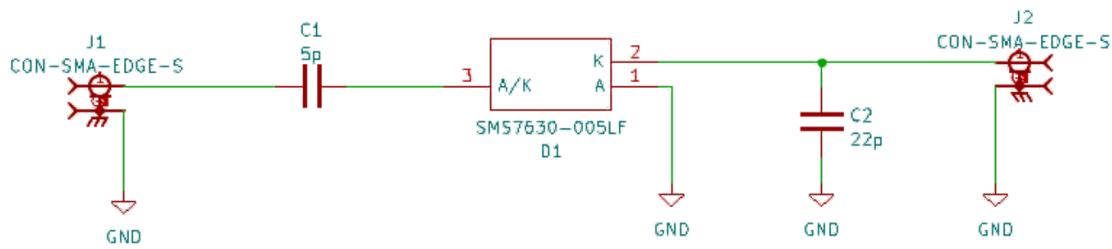


Figure 12: WuR Z-Parameter Prototyping Board Schematic with SMS7630

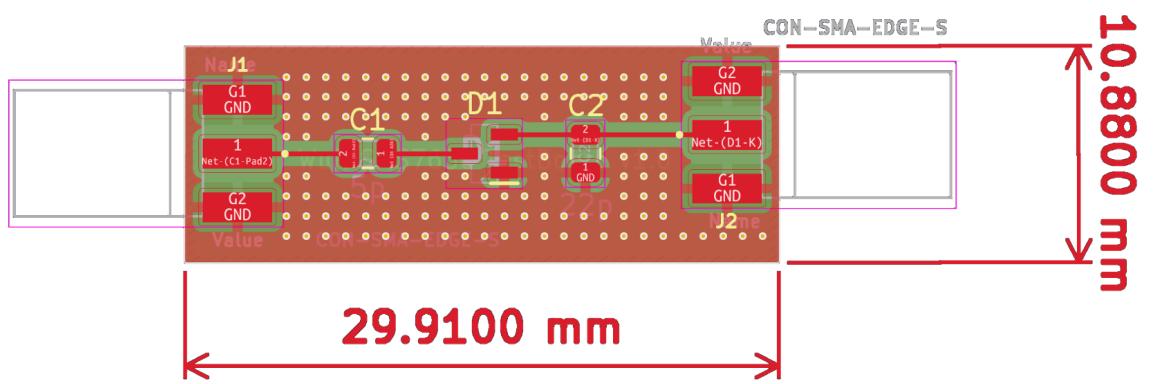


Figure 13: WuR Z-Parameter Prototyping Board Layout with SMS7630

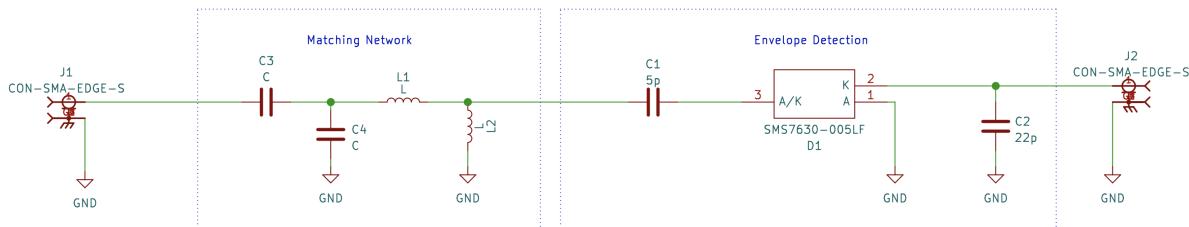


Figure 14: WuR Impedance Prototyping Board Schematic with SMS7630

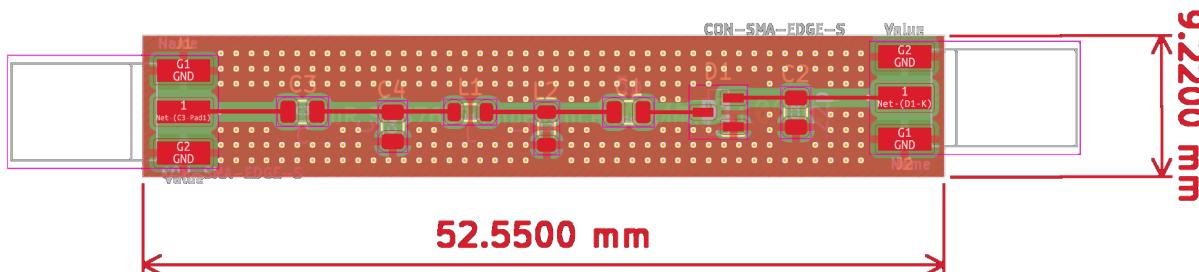


Figure 15: WuR Impedance Prototyping Board Layout with SMS7630

| position | cost [€] |
|---------------|-------------|
| PCB | 0,82 |
| Assembly | 0,00 |
| Parts | 0,80 |
| Shipping | 0,56 |
| Customs + Tax | 0,27 |
| Total | 2,45 |

Table 4: Overview of per board cost of Impedance Matching Prototyping board

Five of each PCBs were produced by JLCPCB for 18.08 USD (13.04.2023) including shipping, customs and taxes. The surface mount parts including the capacitors, the diode and the SMA connectors for were sourced at Digikey. The parts were self assembled. In table 4 a cost overview for the "Impedance Matching Board" from figure 15 can be found.

Impedance Matching

The process of impedance matching is a fundamental aspect of designing any RF circuit, including our wake-up receiver. Figure 16 shows a picture of an unpopulated circuit board that served as the initial test site for measuring the Z-parameters.

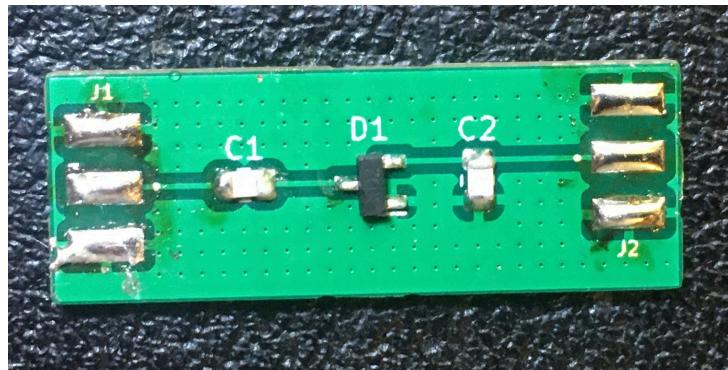


Figure 16: PCB for Z-parameter measurement

The board contained only two diodes and two capacitors. This setup is also known as Greinacher voltage doubler. The plan is to measure the Z-parameters of the board to get a first indication of the matching network. However, when using the VNA, calibration problems were encountered in higher frequency ranges. As Figure Y illustrates, the output power in a "pass-through" arrangement - where ports 1 and 2 are shorted - is significantly lower than expected beyond the 3 GHz mark. In view of this limitation, the frequency range for impedance matching was restricted to 3 GHz, but this led to some uncertainty in the results. Subsequently, the Z-parameters were determined and transferred to the simulation tool Software Advanced Design System (ADS), which provides tools for the design of matching networks.

Further problems arose when comparing the simulation results with the actual VNA

measurements. The simulation and the real results differed significantly. For test purposes, individual inductors and capacitors were implemented in series or as a shunt as a matching network to measure the network response. In the figure X-Z, the Z-parameters expected from the simulation are plotted in comparison to the measured values. Attempts to compensate for these discrepancies by improving the ADS simulation by importing the PCB's Gerber file to simulate microstrips and pads were unsuccessful.

Consequently, a strategic shift was required to overcome these hurdles. A trial-and-error approach was decided upon, in which the VNA measurements were supplemented by checks with the NanoVNA to mitigate the earlier measurement uncertainties. The NanoVNA is a low-cost, open-source device suitable for measuring S-/Z-parameters up to 3GHz. The process involved testing various configurations and making adjustments based on informed intuition. After numerous iterations and constant fine-tuning, a satisfactory result was achieved. The impedance of the board was effectively adjusted to the target frequency of 868 MHz, representing an important milestone in the development process of the wake-up receiver. The success of this hands-on, iterative approach confirms the combination of rigorous theoretical calculations and practical adjustments in overcoming real-world technical challenges. This completes the reproduction of the WuR front end. A detailed description of the tests carried out will be given in the evaluation chapter [6](#).

5.1.2 Reproduction: Fromm et al. (2022)

The procedure for reproducing the design of Fromm et al. [\[19\]](#) has taken a slightly different form than written in the previous section. The WuR frontend is much more complex as explained before and various components are not quantified like the passive components of the matching network. On request, however, we were kindly sent the entire schematic of the setup. Without the circuit diagram provided, reproduction would have been impossible because too much information was missing and the system is too complex to progress on the basis of educated guesswork.

In the first step, the schematic was adapted for our purpose and rebuilt in KiCad. We discarded the microcontroller and power management at first, because only the WuR frontend is chosen for reproduction. For testing proposes additional u.FL coaxial sockets have been mounted. (see figure [17](#)) The hole schematic can be found in figure [18](#). As in [5.1.1](#) the micro strip width was calculated to 0.29337mm and taken into account via stitching in the layout. These boards were also ordered from JLCPCB including part assembly. Since the amplifiers, the diode and bandpass filter were not available at JLCPCB, they were sourced via Mouser and self assembled. The overview of the costs per boards can be found in table [5](#).

| position | cost [€] |
|---------------|--------------|
| PCB | 1,20 |
| Assembly | 2,93 |
| Parts | 11,64 |
| Shipping | 0,59 |
| Customs + Tax | 0,92 |
| Total | 17,28 |

Table 5: Overview of per board cost of WuR Frontend board inspired by [19]

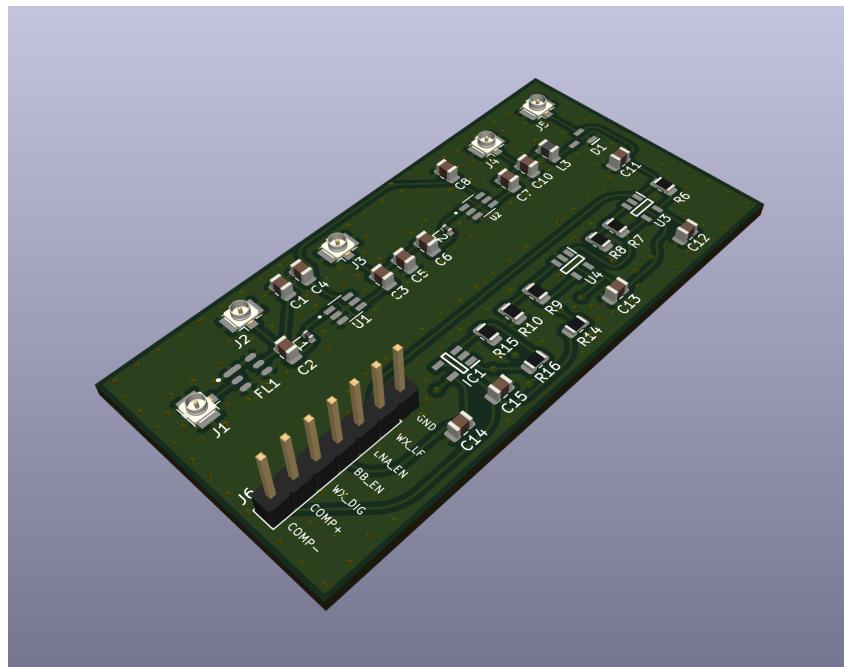


Figure 17: Wake-up Receiver circuit board 3D animation based on Fromm et al. [19]

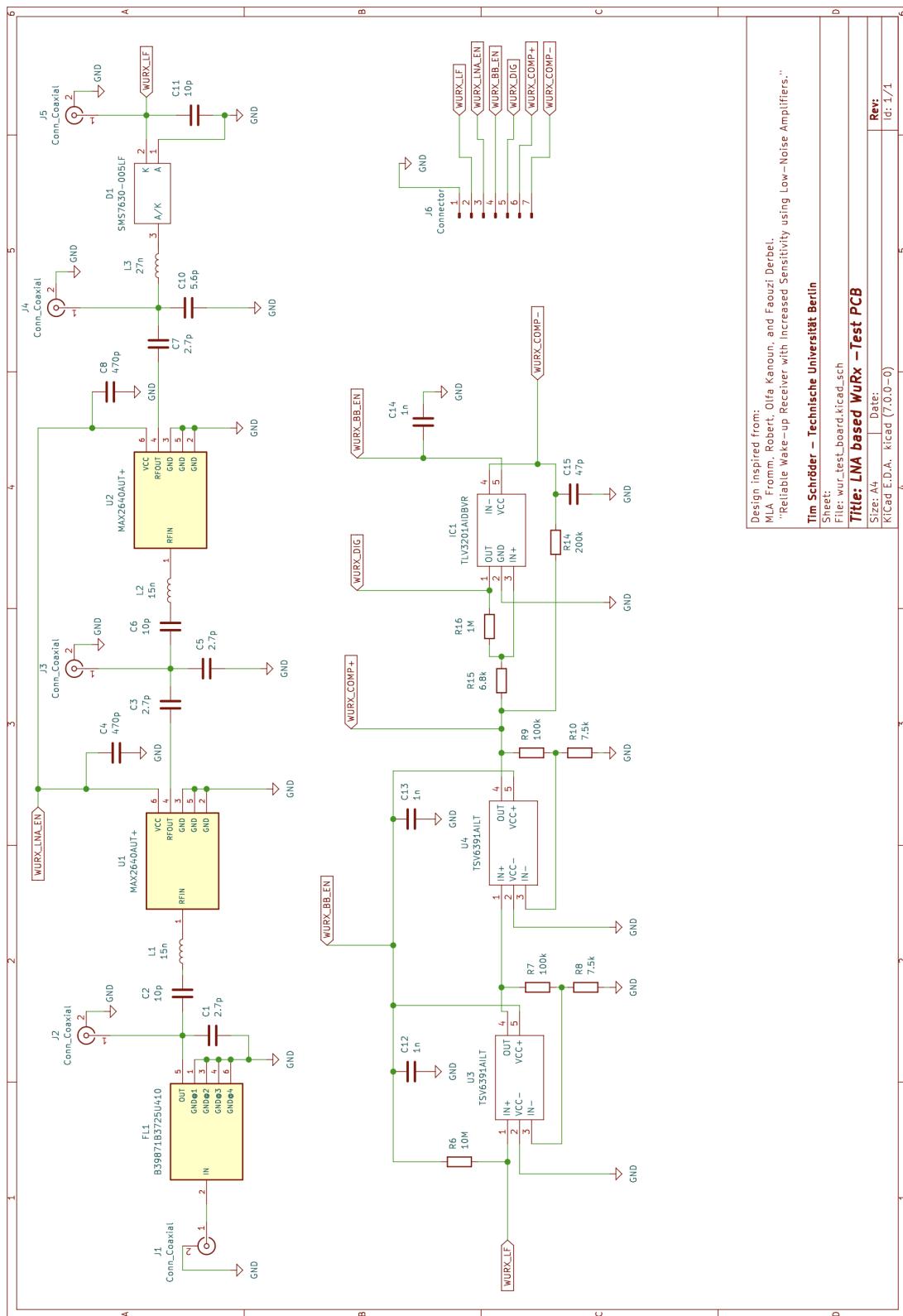


Figure 18: Wake-up Receiver circuit schematic based on Fromm et al. [19]

5.2 Design of own system

This chapter is embarked on for a detailed exploration of the wake-up receiver's hardware design process, taking into account the insights gained from a comprehensive literature review, as well as the challenges and design requirements identified in chapters 3 and 4. Due consideration has been given to the outcomes of the two WuR approaches reproduced from existing literature, with these learnings being applied to the design process. Unlike traditional research, the goal here transcends the confines of the academic community. Being an open-source project, the focus extends to incorporate crucial yet often understated aspects such as usability and reproducibility. Consequently, the design process is shaped to not only enhance the system's performance and energy efficiency but also to ensure that it remains user-friendly and readily reproducible for the broader community. With the groundwork for the hardware development established first, a comprehensive exploration of the software development process will subsequently be delved into.

5.2.1 Hardware Design

In this section, the methodical hardware design process for our WuR is described, from initial signal acquisition to final data decoding and MCU activation. We address the front-end design, digitisation, preamble filtering and address matching mechanisms, all of which play a critical role in improving the performance and energy efficiency of the system.

Frontend

Reproduction of two different frontend developments in the previous section has provided significant learnings about how literature approaches this. However, it is important to take a step back and recall Chapter 4, which discusses how this project differs from the design requirements of classical research projects. Unlike traditional research, the hard design factors such as energy efficiency and range are accompanied by soft factors such as usability and reproducibility. With this understanding, the initiation of the frontend development can now commence.

Simplicity in design and usage is clearly the right path to follow, which for the frontend means: as few complex components as possible, making the design easier to understand, build and modify. Furthermore, the frontend should be easily separated from the rest of the sensor node for the sake of modularity. In this way, the frontend can be used in a wide variety of sensor nodes and can work independently from the utilized digitization and decoding modules. This approach is also in line with the principle of affordability. As shown in 5.1.2, producing a complex frontend comes with significantly higher costs. From the standpoint of usability, there are also many reasons to follow a simple design and only

if that works smoothly enhance efficiency through complexity. However, the starting point must be simple to ensure the entry barrier for developers is not too high, given that wake up receivers and HF systems in general are already quite complex.

This means that the focus shifts towards a simple envelope detector for the WuR frontend. This approach has been validated by much literature, including [13]. This means that the Grainacher voltage doubler described in section 2.4.2 will be used. The diode HSMS-2852 from Agilent Technologies used by [13] is no longer produced, as already mentioned in 5.1.1 the SMS7630-005LF from Skyworks Solution Inc is a suitable replacement. The capacitor value of 22pF for the envelope detector found in literature, has been confirmed by simulation. The PCB that was created during the reproduction process of Magno et al., described in section 5.1.1, can be used as the frontend prototype. This allows to use the solution found for the matching network. Any change made to the PCB, such as the specific microstrip lengths and widths, can shift the match.

The design was tested by connecting an antenna for the 868MHz frequency range with the frontend, as shown in figure 20.



Figure 19: Wake-up Receiver Frontend + 868MHz Antenna

A wake-up signal (WuS) was generated at a data rate of 15.6kHz using an Arduino shield based on the transceiver module SX1276 at 10dB transmission power at approx. 0.2m distance. The wake-up signal includes a 6 bit preamble and minimum energy encoded address. This means that a 868MHz carrier pulse is sent for the time of $T_{\text{preamble}} = \frac{1}{15600\text{Hz}} * 6 = 384,6\mu\text{s}$ followed by one 0 bit to differentiate between preamble and payload. The payload is a high bit with a position that depends on the address value. For now the address is 8 bit long, the nodes address will be defined by the position of the high bit. It is expected that the two pulses - longer preamble and shorter address bit - will be rectified and thus can be observed as square signals on the oscilloscope. For this purpose, the

WuR frontend was placed close to the transmitter and the output signal was connected to the oscilloscope. In Figure 20, the rectified pulses measured by the oscilloscope DZ1104Z from Rigol are visualized. It was also observed how the amplitude decreases as the transmitter-receiver distance increases. This confirms the correct functioning of the WuR frontend.

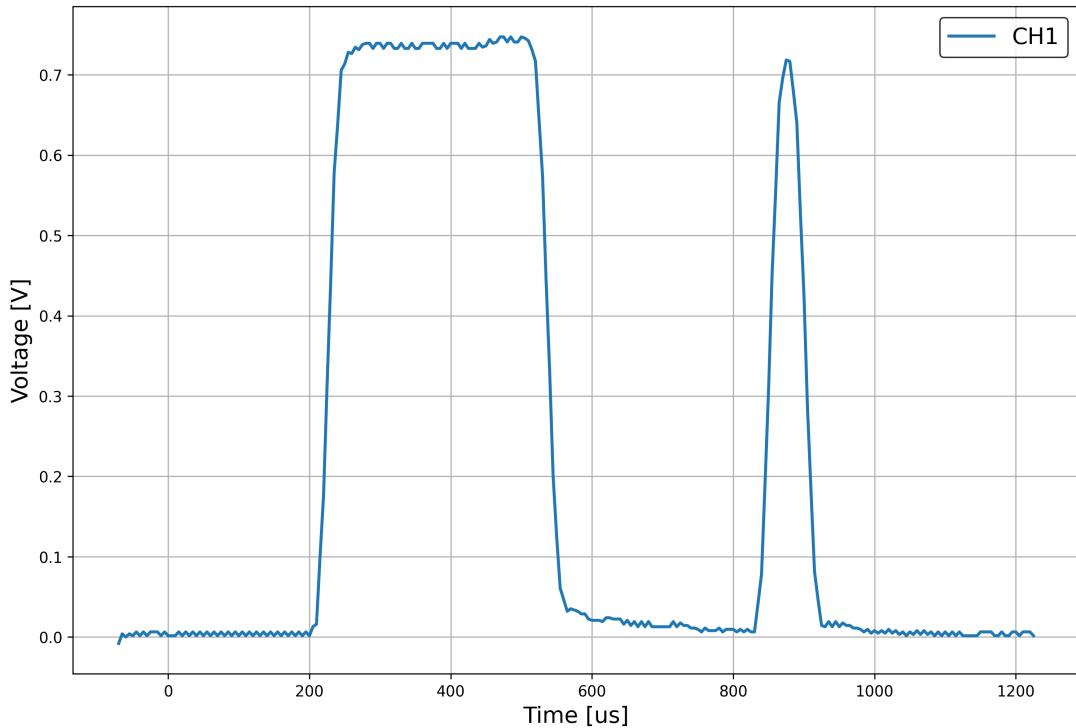


Figure 20: Channel 1 (blue): Envelope of Wake-up Signal

For the complexity reasons mentioned, LNAs and SAW filters will be omitted, even though they promise longer range. However, these could be good iteration steps for future designs. A major hurdle for reproducibility that needs to be addressed is the matching network. During the reproduction process of the frontend from [13], different problems arose that are described in detail in section 5.1.1. To summarize, it should be emphasized here that matching in the entire WuR system is the greatest challenge in terms of reproducibility. This is due to the variance in components and PCBs, which matching is very sensitive to. Specifically, this means that not every 22pF capacitor actually has 22pF, and the behavior profile at higher frequencies can differ greatly from one series and manufacturer to another. The fine-tuning described in 5.1.1 is only possible with a lot of time and access to expensive measuring equipment such as vector network analyzers, a large palette of capacitors and inductances, and simulation programs like ADS. To simplify this process, it is necessary to stick to passive components of very specific series from certain manufacturers that are all available from a very accessible PCB manufacturer, and to provide corresponding production files, so that the frontend just needs to be ordered.

Digitalization

After discussing the frontend development, we now have the rectified OOK signal available (refer to the oscilloscope image). However, the signal strength can vary, leading to a different signal-to-noise ratio. This variability complicates digitization, as we cannot use a fixed threshold to compare the signal, a standard practice when using comparators. Typically, this reference voltage is created by a voltage divider. However, [13] pointed out that this method consumes energy and proposed an adaptive thresholding approach with a low-pass filter, which we have confirmed via simulation. Since the implementation details were missing, we couldn't elaborate on this mechanism during the replication process. However, as part of our own development, we aim to explore this method. The comparator from Magno et al. is a possible candidate for this, but it is outdated. Thankfully, Texas Instruments offers a suitable, more energy-efficient, and readily available successor model, the TLV7031 (comparison table of the comparator from Magno et al. and TLV7031). We will determine the values of the low-pass filter experimentally. Generally the smaller the time constant τ of the filter, the quicker the comparator reacts to changes at the input, but it also becomes more susceptible to noise. [19] suggests that the time constant τ should be about five times the bit time T_{bit} . For a data rate of 15.6kHz, this equates to:

$$\tau = 5 * \frac{1}{15.6} \approx 5 * 64\mu s \approx 320\mu s \quad (5.2.1)$$

Keeping reproducibility in mind when selecting the components' values. It makes sense to use components with standard values. In addition, the experience has been made that 5 bits as time constants delay the system considerably and the value can be selected smaller. Using the standard values of $220k\Omega$ and $1nF$, we obtain a time constant of:

$$\tau = 1nF * 220k\Omega \approx 220\mu s \quad (5.2.2)$$

Unlike the frontend, the rest of the Wake-up receiver (WuR) doesn't need to be on a PCB because, after rectification, the signal loses its high frequent components. This means the circuit can initially be implemented on a breadboard. The developed WuR frontend will be connected to the comparator as shown in Figure 21. The curve on channel 1 (yellow) is the output of the envelope detector and the input of the non-inverting input of the comparator. The rectification of the 3 bit ($156\mu s$) pulse of the 868MHz carrier is clearly visible as a square wave signal. On channel 2 (blue) the inverted input of the comparator is shown, that can be identified as the low-passed filtered version of the rectangle.

The figure 22 in comparison has the comparators output on channel 2 and shows the digitalization of the signals envelope. The $3\mu s$ delay stated in the datasheet can be confirmed.

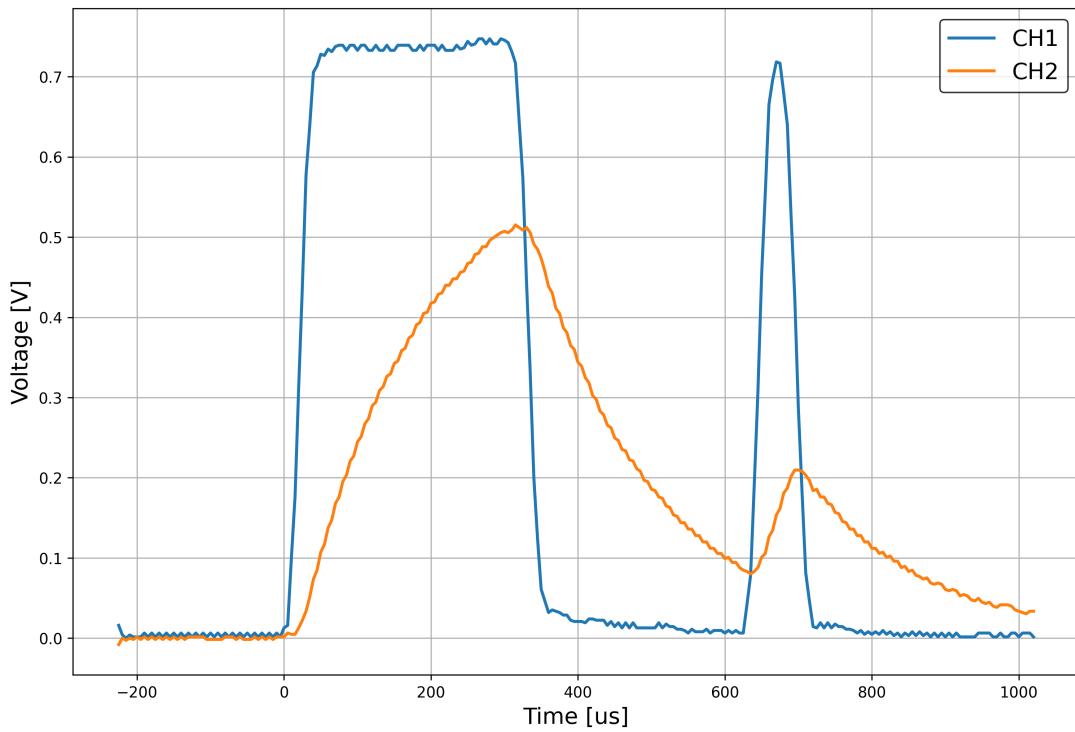


Figure 21: Channel 1: Envelope of WuS (non-inverted comparator input); Channel 2: Low-pass filtered envelope of WuS (inverted comparator input)

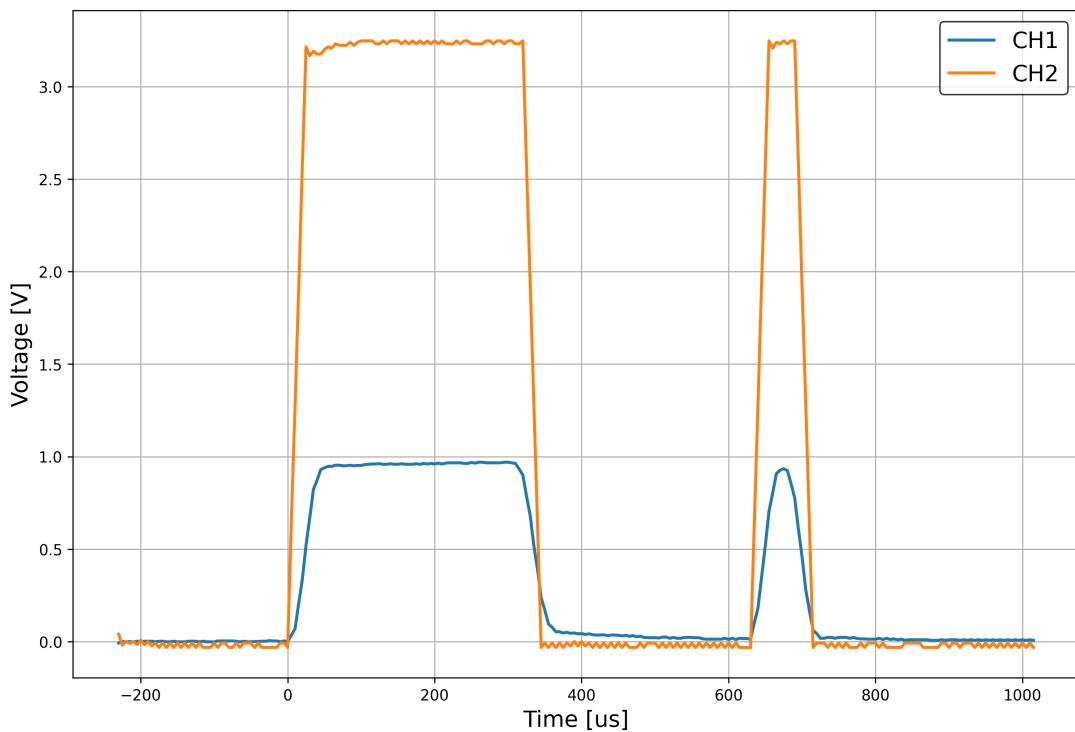


Figure 22: Channel 1: Envelope of WuS (non-inverted comparator input); Channel 2: Comparator output

The oscilloscope images already reflects real transmission conditions via antenna, which is preferable for prototyping than input via cable. It's crucial to consider noise during parameter determination.

A valid argument might be why not use the AS3933, given its lower development effort, availability, and proven use in literature. However, it has some drawbacks: it's relatively expensive, sets specific parameters such as bitrate and data encoding (Manchester coding), and offers less flexibility. Additionally, a comparator circuit allows everyone to understand what's happening, which is more hands-on. This makes the system adaptable and excellent for learning and understanding electronic systems.

Preamble Filter

After successful digitization of the signal, the next step is to process this data signal. At this point, it's possible to connect the system to any microcontroller to receive data. However, the issue that arises is energy efficiency. To decode the signal, the MCU would need to remain active continuously, defeating the purpose of the wake-up receiver. The wake-up receiver is meant to make the entire system more energy-efficient by minimizing the active communication time of the MCU. Therefore, we need an intermediary to determine when a transmission begins. The paper [13] suggested a solution that can be easily implemented with off-the-shelf components: a preamble filter.

A preamble is a signal sent before the payload, the actual data to be transmitted. This allows the transmitter to synchronize with the receiver. The proposed preamble filter is a low-pass filter, where an incoming preamble charges a capacitor. When the capacitor reaches a certain threshold through the input signal, it can be perceived as a rising edge by an interrupt detector. The curve of the low-pass filtered preamble detector can be observed in figure [23] in comparison to the comparator output.

The preamble detector circuit is shown in figure [24] and works as follows: The resistor R3 limits the current flow charging the capacitor C5, determining how quickly the capacitor reaches its maximum voltage. The capacitance of C5 also influences this since the higher the capacitance, the longer the time it takes to fully charge at a constant current. The Schottky diode D3 prevents C5 from discharging through the comparator when it has grounded the output. Instead, the capacitor C5 should discharge in a controlled manner via the resistor R3, the value of which determines how quickly the capacitor can discharge.

These three parameters must be adjusted so that the critical threshold value of the interrupt detection instance is reached by the energy of the preamble. However, the preamble detector should not trigger on individual high bits. As with the setting of the filters for the comparator input, the values were determined experimentally for a bitrate of 15.6kHz, considering the use of standard components.

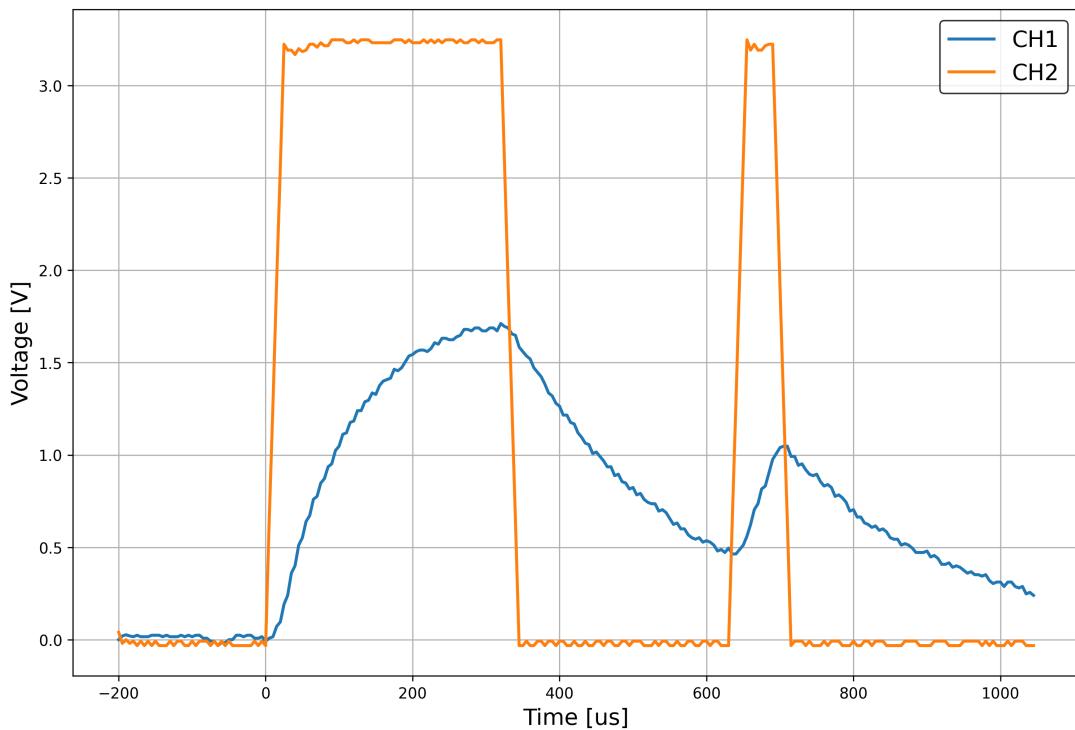


Figure 23: Channel 1: Low-passed comparator output as preamble detector 2: Comparator output

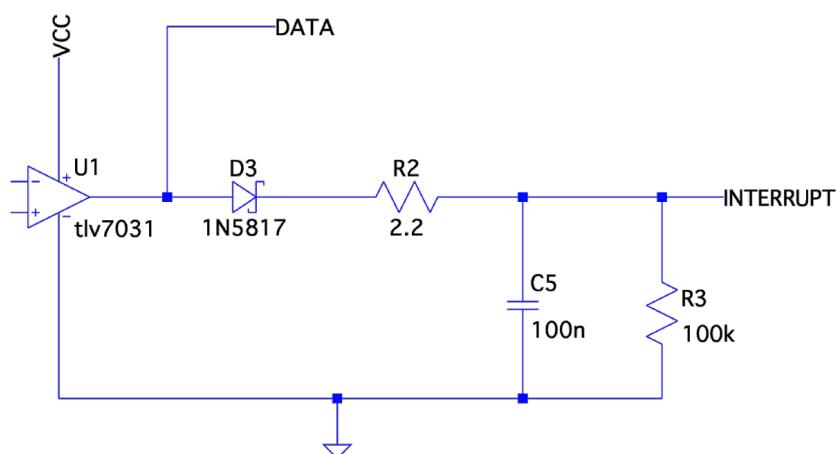


Figure 24: Comparator circuit with preamble filter

As explained before, the preamble filter strongly depends on the component to be interrupted by the preamble signal. Furthermore, besides the preamble, the actual data also needs to be decoded. Therefore, this part will be discussed in the following section.

Decoding

In principle, any microcontroller could be connected after the preamble filter to be woken up by the preamble interrupt. Meaning if no address matching needs to be performed, for instance because there is only one node, the decoding unit can be omitted. However, as soon as multiple nodes are used, an address matching mechanism should be employed. This mechanism helps the nodes determine whether a message is intended for them or another node. The address matching must be carried out by a unit other than the MCU for both energy efficiency and delay reasons. This is because the MCU consumes too much energy as it is overpowered for a simple address correlation. Moreover, standard MCUs take too long to switch from sleep mode to an active mode. This means a very energy-efficient component with a low wake-up delay must be found.

[13] suggests the PIC12LF1552, an ultra-low power 8-bit microcontroller. In addition to the interrupt functionality, the ultra-low power microcontroller should also have a communication interface like I2C or SPI so that it can be configured by the MCU as described. It would be beneficial if there was a suitable ultra-low power 8-bit processor that can be programmed with the Arduino IDE, as C and the PIC architecture can pose a hurdle for developers. Alternative 8-bit microcontrollers for Arduino include:

| MCU | New Designs | Active (uA/MHz) | Sleep (nA) | Arduino Compatible |
|-------------|-------------|-----------------|------------|--------------------|
| STM8 | No | 150 | 200 | No |
| ATTiny25 | Yes | 300 | 100 | Yes |
| PIC12LF1552 | Yes | 30 | 20 | No |

Table 6: Comparison of 8-bit MCUs

These considerations lead us to choose the PIC. Thanks to its convenient DIP8 package, it can be easily added to the breadboard setup. The wiring is shown in figure [25]. To add the PIC to the setup, it must be programmed beforehand. The task here is to fulfill the functionality of preamble detection, address correlation, and interrupt generation. The development of the software will be discussed in the next section.

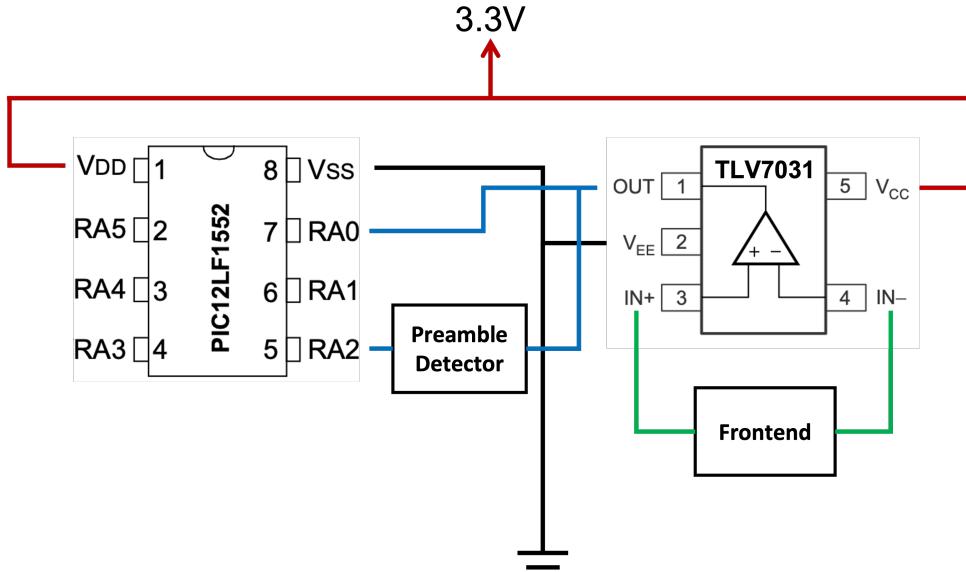


Figure 25: Wiring of PIC12LF1552 with comparator TLV7031

5.2.2 Software Design

The software design chapter addresses the critical role of software in the design of an energy-efficient and reliable wake-up receiver (WuR), which has been heavily neglected in the illuminating literature. The aim of this chapter is to explore the complex interplay between hardware components and software algorithms and how they jointly contribute to the overall performance of the system. This investigation begins with a detailed discussion of the software requirements of a WuR, derived from the knowledge gained in the previous chapters. Furthermore, this chapter explains the development of algorithms that optimise system functionality while reducing energy consumption. Key elements of this discussion include coding schemes, preamble detection strategies and communication protocols, all of which have a profound impact on the robustness and energy efficiency of WuR. The following sections show how these concepts were practically applied in the development of the software for our WuR system. For a better overview, a diagram of the connection between the modules of the WuR system is shown in Figure 26.



Figure 26: Connections between Wake-up Receiver Modules and MCU

Preamble detection and synchronization

The first step in the process of receiving a signal in the wake-up receiver (WuR) system involves the detection of a preamble. The preamble is detected by an external low-pass filter circuit, which charges its output to a level that can trigger an interrupt when the preamble signal is detected. This is directly linked to the PIC12LF1552's pin RA2. RA2 is chosen for this purpose because it is the only pin that supports external interrupt functionality with edge selectivity - it can be set to trigger an interrupt on either a rising edge or a falling edge. This flexibility is critical in the context of the WuR system, as we need to detect both the start (rising edge) and end (falling edge) of the preamble. Once the rising edge of the preamble is detected, the PIC12LF1552 wakes up from its sleep mode. The microcontroller then immediately changes the interrupt sensitivity of the RA2 pin from the rising edge to the falling edge. The purpose of this is to detect the end of the preamble, which is characterized by a falling edge. This moment also marks the synchronization point for the subsequent data reception and decoding. After the falling edge on the RA2 pin is detected, marking the end of the preamble, the PIC resets its Timer0 module and activates the interrupt functionality on the RA0 pin. Before this point, the interrupt on the RA0 pin is kept inactive to avoid any undesired interrupt events that could interfere with the preamble detection. In the PIC MCU the interrupts are handled by the interrupt service routine (ISR). The ISR is a special function in a program that is executed automatically by the microcontroller in response to a specific interrupt condition, such as a timer overflow or a signal from other hardware components.

We noticed that the detection of the end of the preamble was delayed by $80\mu\text{s}$ due to the low-passed characteristics of the filter signal. Since this is more than $T_{\text{Bit}} = 64\mu\text{s}$ an alternative solution must be found. We changed the concept to only using the preamble detector signal for the rising edge detection and then switch to detect the falling edge of the preamble on the non-filtered data channel on pin RA0. This allowed the delay to be reduced to $7\mu\text{s}$. For this purpose, the respective interrupts were only switched on at the respective time.

```

1 void __interrupt() isr(void) {
2     if (INTCONbits.INTF) { // PREAMBLE START DETECTED
3         IOCAFbits.IOCAF0 = 1; // Activate int on RA0
4         INTCONbits.INTF = 0; // Reset the interrupt flag
5     }
6     else if (IOCAFbits.IOCAF0) { // PREAMBLE END DETECTED
7         if (IOCANbits.IOCANO) { // Falling edge
8             // PREAMBLE ENDS
9             INTCONbits.TMROIF = 0; // Clear Timer1 interrupt flag
10        }
11    }
12 }
```

Time Measurement

To effectively measure time within our system, we utilize the Timer0 module in the PIC12LF1552. Timer0 is a part of the PIC MCU that can count up from 0 to 255 (because it's an 8-bit timer), and when it reaches its maximum count, it overflows and starts again from 0. This overflow event generates an interrupt, indicating that a certain amount of time has passed. The timing is determined by the clock source, which in this case is the internal instruction cycle clock with a frequency of 16MHz.

The address information is transmitted following the preamble. The address is represented in a minimal energy encoding where only one bit in an 8-bit address is high. The PIC decodes the position of this one high bit by measuring the time from the end of the preamble to the rising edge on the RA0 pin. The elapsed time results from the following formula, where n_{cycles} is the number of timer overflows, R_{timer} is the value in the timer register TMR0 and $f_{crystal}$ is the frequency of the PICs internal oscillator:

$$T = (n_{cycles} * 256 + R_{timer}) / (f_{crystal}/4) \quad (5.2.3)$$

In figure 27 the interrupts of the timer are visualized by toggling a pin of the PIC microcontroller every time a overflow was registered.

If the position of the high bit matches the PIC's address, the PIC generates an interrupt on the RA5 pin to wake up the main processor and then goes to sleep. This intelligent sleep-wake mechanism facilitates efficient power usage in the WuR system. In our wake-up receiver system, the Timer0 module is started after the end of the preamble, at the falling edge interrupt. Each time the timer overflows, it triggers an interrupt that is handled by the interrupt service routine (ISR). Within the ISR, we increment a counter to keep track of the number of timer overflows, as shown in the following code:

```

1 void __interrupt() isr(void){
2     else if (INTCONbits.TMR0IF) { // Timer1 overflow interrupt
3         timerCycles++;           // count overflows
4         INTCONbits.TMR0IF = 0;    // Clear Timer1 interrupt flag
5     }
6 }
```

In this way, we can measure the elapsed time in terms of the number of Timer0 overflows. This timing information will be necessary for decoding the position of the single high bit in the address data received after the preamble.

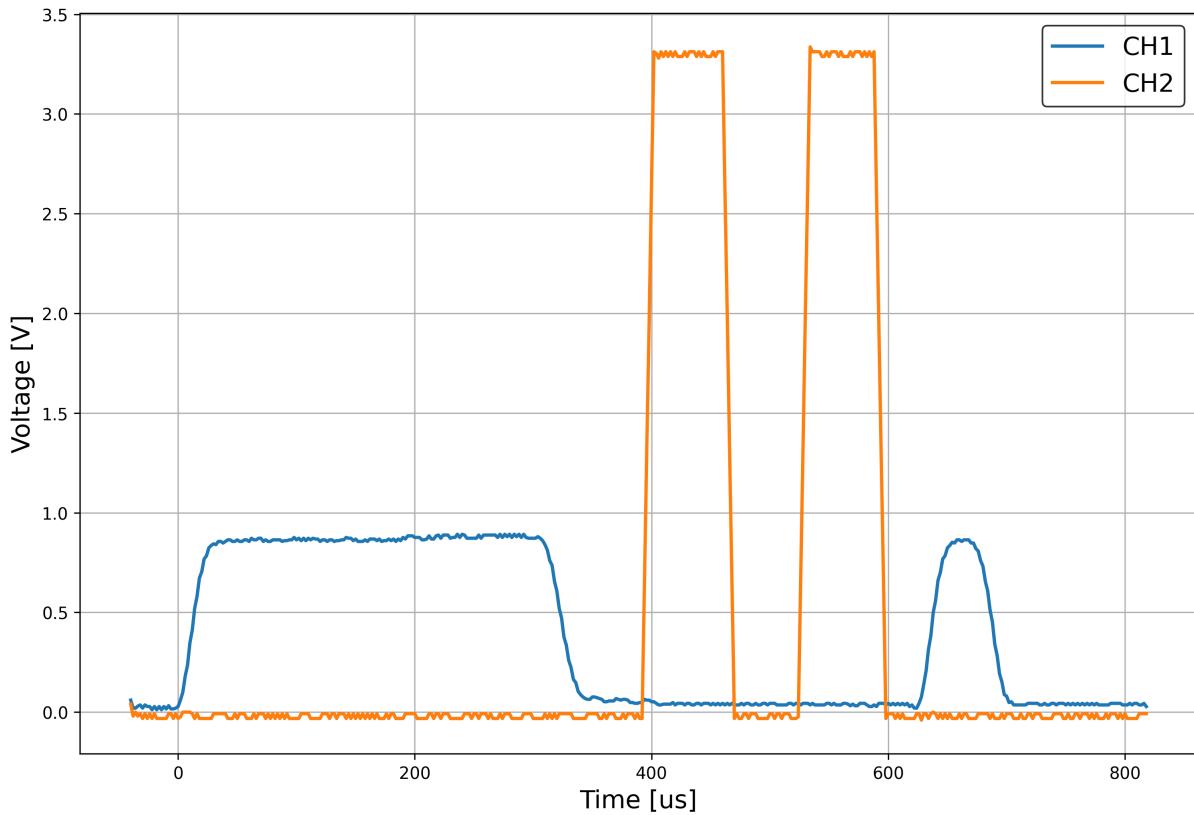


Figure 27: Channel 1: Envelope of the WuS; Channel 2: Timer Interrupts

Address Decoding and MCU Wake Up

In our wake-up receiver system, after the preamble, the address data is transmitted. This address is an 8-bit length data, which uses minimum energy coding - meaning, only one bit in the address is high. The position of this high bit in the address indicates the specific address of a node in the network. In order to decode this address data, our system measures the time between the end of the preamble and the detection of the high bit in the address. The PIC MCU accomplishes this through the utilization of Timer0, by counting the number of overflows. The high bit causes an interrupt on the RA0 pin due to a rising edge. This is part of the ISR handles this interrupt:

```

1 void __interrupt() isr(void) {
2     else if (IOCAFbits.IOCAF0) { // DETECTED PREAMBLE END
3         unsigned int timerValue = TMRO;
4         unsigned int totalTime = (timerCycles * 256 + timerValue)
5             / 4;
6         if (totalTime >= (lowThr) && totalTime <= (highThr)) {
7             interrupt();           // wake the MCU
8         }
9         IOCAFbits.IOCAF0 = 0;    // Reset IOC Interrupt flag
10        IOCAPbits.IOCAPO = 0; // Deactivate INT on RA0
11        SLEEP();

```

| | |
|----|---|
| 11 | } |
| 12 | } |

When the high bit of the address is detected, the MCU calculates the total elapsed time since the end of the preamble by multiplying the number of timer cycles by 256 (since Timer0 is an 8-bit timer), adding the current value of Timer0, and dividing the result by 4 because the timer is incrementing every 4th clock cycle.

The MCU then checks if this elapsed time falls within the expected bit time window, with a tolerance of 10%. If the time corresponds to the expected position of the high bit for the node's address, the main processor is woken up by interrupt. After handling the address bit, the MCU resets the timer cycle count and puts itself back to sleep mode, ready for the next wake-up signal.

In figure 28 the interrupt generated based on the detection of the address bit is shown. With the oscilloscope a delay of $864\mu\text{s}$ from the rising edge of the address bit to the rising edge of the interrupt was determined.

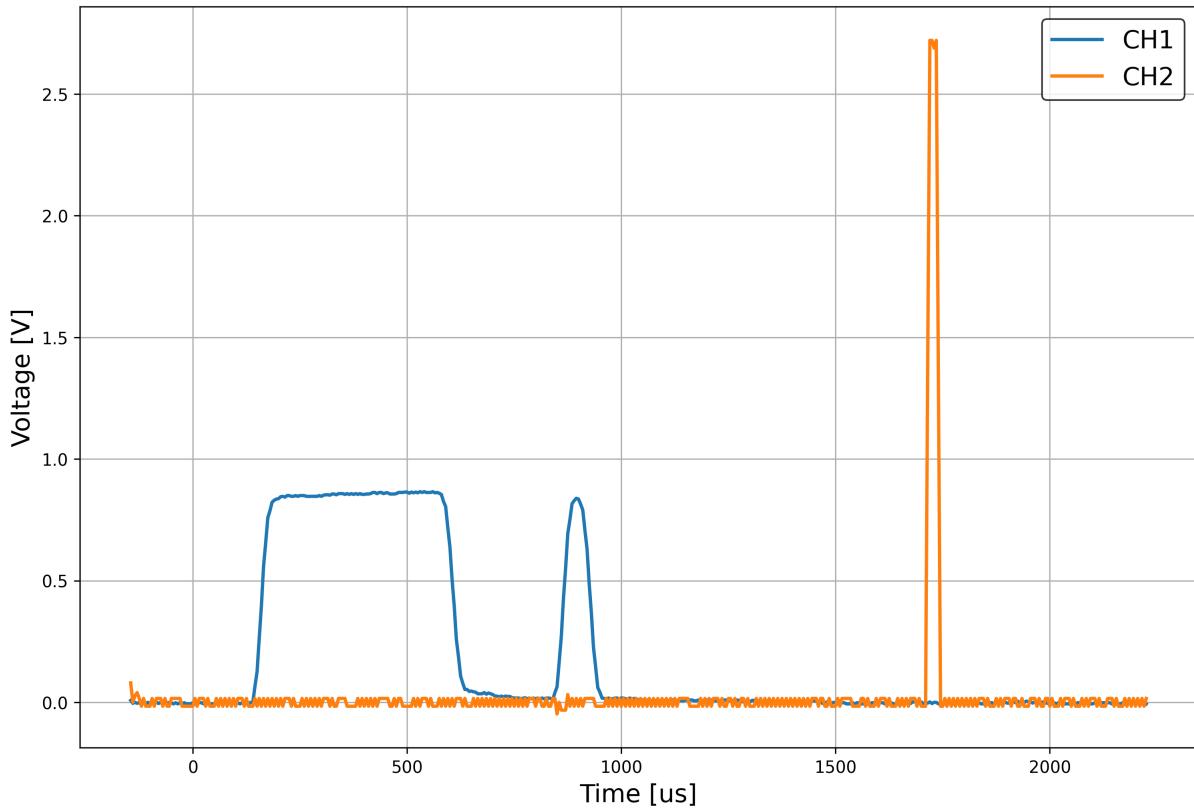


Figure 28: Channel 1: Envelope of WuS; Channel 2: Generated Interrupt

The first obvious assumption is that this strong delay is caused by an incorrect detection of the rising edge at the address bit. But this is not the case. It should be noted that we are working with a small 8-bit microcontroller that needs time for the calculations described in the code. Therefore, a much more efficient method must be used to determine

the position of the address bit. It occurs to us that it would be very practical if the bit time T_{Bit} were synchronised with the timer overflow cycle. This means that for each time T_{bit} the timer overflows exactly once. The timer interrupt frequency is given by

$$f_{\text{cycle}} = \frac{f_{\text{oscillator}}}{4 * 256} = \frac{f_{16\text{MHz}}}{4 * 256} * 256 = 15.625\text{kHz} \quad (5.2.4)$$

This results in the time that elapses until the timer interrupt comes:

$$T_{\text{cycle}} = \frac{1}{f_{\text{cycle}}} = \frac{1}{15.625\text{kHz}} = 64\mu\text{s} \quad (5.2.5)$$

It is no coincidence that this already corresponds quite exactly to the bit time $T_{\text{Bit}}=64\mu\text{s}$. The data rate was only adjusted to 15.6kHz after this realisation. Thus, the transmitted address can be easily determined by the number of timer interrupts.

To do this, the timer register TMR0 must be set to $\frac{T_{\text{Bit}}}{2} = 128$ at the beginning, so that the timer always overflows in the middle of a bit. If this were not done, there would be a danger that a timer cycle would be counted wrong, since it would increase exactly at the beginning or end of T_{Bit} . This situation is illustrated in the following figure 29.

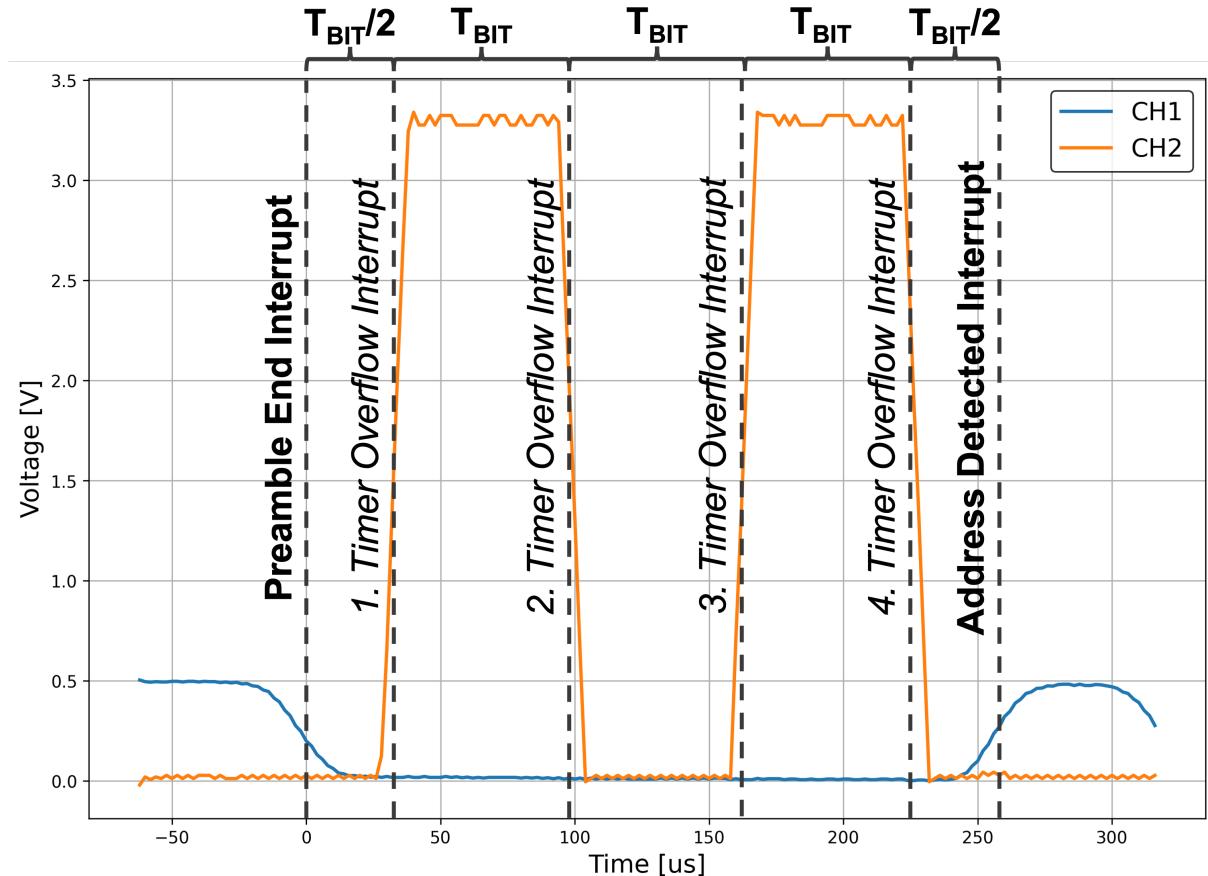


Figure 29: Timing in Address Decoding

With the optimized code the delay from the rising edge of the address bit to the rising edge of the interrupt pulse could be reduced from $864\mu\text{s}$ to $7\mu\text{s}$ which is shown in figure 30.

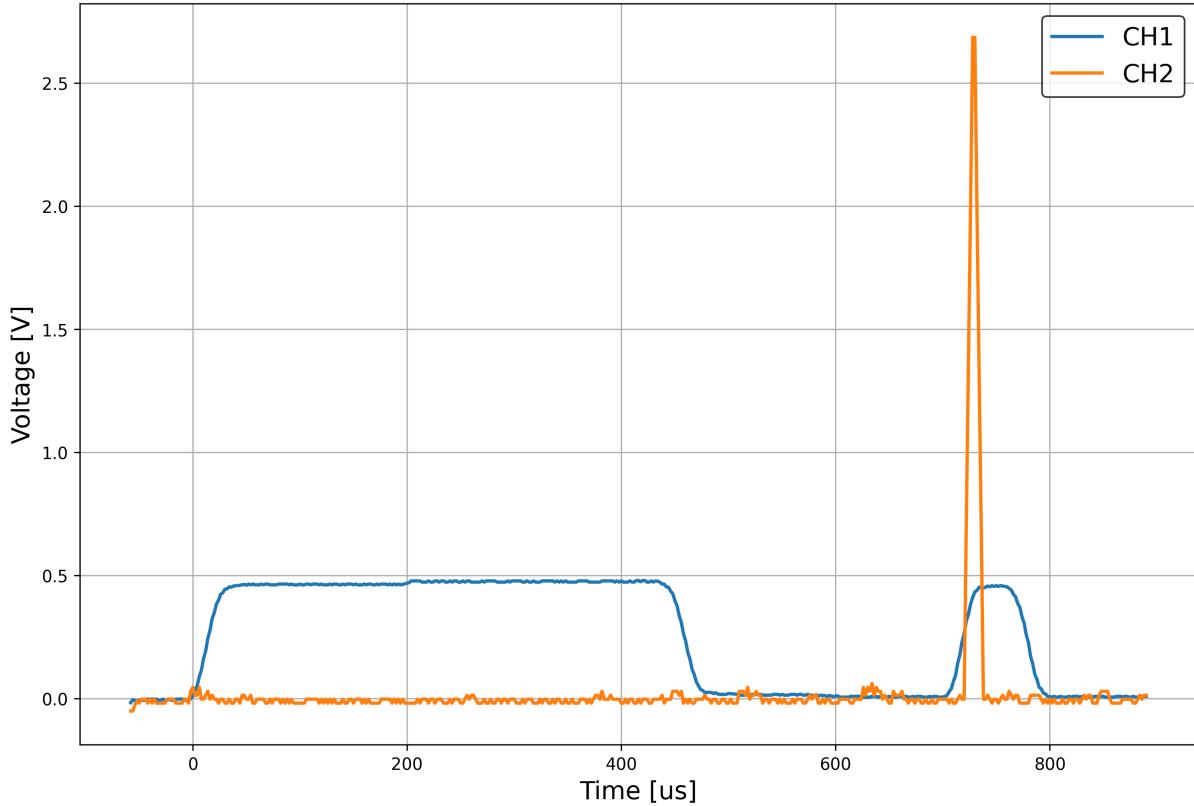


Figure 30: Reduced interrupt time based on optimized code

In conclusion, this chapter illustrated the intertwined nature of hardware and software in developing an efficient wake-up receiver system. Our approach has unveiled the crucial role of a harmonious hardware-software collaboration in achieving optimal performance and efficiency.

As we transition into the next chapter, we will shift our focus towards the evaluation phase. We will delve into the assessment of the reproductions and the wake-up receiver system in depth, considering various metrics and scenarios to gauge their performance. Through a systematic analysis of the results, we will seek to validate our design decisions and to identify any potential areas for further improvements. This comprehensive evaluation will help ensure that our system fulfills its intended objectives while providing reliable and consistent performance.

6 Evaluation

Having followed the complex path of reproducing existing designs and developing our own wake-up receiver system in the previous chapters, we now turn to a comprehensive evaluation of these efforts. In the previous chapters we have worked out the basics, starting with a detailed literature review, which formed the basis for the design and development phase. By analysing existing solutions, we recognised crucial features and identified potential bottlenecks, providing critical insights that shaped our design requirements. In particular, the previous chapter guided us through the iterative process of hardware development. This resulted in two different front-ends: a reproduction of Fromm et al's design [19] and our own design inspired by Magno et al [13]. Both of these can be connected to the wake-up decoders we have developed.

In this evaluation phase, we will examine the efficiency and effectiveness of these designs and compare them with the design requirements described in chapter 4. First, both front-ends will undergo a thorough performance evaluation, including range, energy efficiency and wake-up delay. Then the systems will be assessed for usability, reproducibility, affordability and modularity. This evaluation will provide a holistic view of the strengths and weaknesses of each design. Our goal is to understand the practicality of the replicated and newly developed wake-up receiver, not just to meet theoretical expectations. This process will serve as a crucial checkpoint to ensure that the designs are consistent with the overall goal of this project - to create an efficient, user-friendly, open-source wake-up receiver.

6.1 Performance

6.1.1 Sensitivity

Following the range evaluations, we delve deeper into assessing the performance of our wake-up receiver designs. In this section, we focus on their sensitivity, a key parameter that greatly impacts their operation and efficiency.

To evaluate sensitivity, we employ a Vector Network Analyzer (VNA) to examine the input reflection, represented as S11. The S11 parameter provides us with an understanding of how much of the input energy from the antenna is reflected back or effectively absorbed by the receiver. A smaller reflection indicates a higher proportion of energy absorption, thus translating into a higher sensitivity of the receiver.

A critical area of interest during this evaluation is the behavior of input reflection at the carrier frequency, in our case, 868MHz. This value indicates the effectiveness of the matching network, responsible for maximizing the power transferred from the antenna to the receiver. A lower input reflection at the carrier frequency generally correlates with a

well-designed and effective matching network, and thereby, a higher receiver sensitivity. Figure 31 shows the most important tools to adjust the matching network. The Smith Chart shows the complex input impedance at different frequencies. This shows how the components need to be changed to improve the matching. The S11 return loss represents the input reflection and is a measure for the quality of the matching. The shown matching is with -15.68dB passable but capable of expansion. Since the result of the simulation is not fully accurate, the matching network must still be adjusted manually. This process has already been described in detail in ??, with a return loss of -26.72dB this result is respectable.

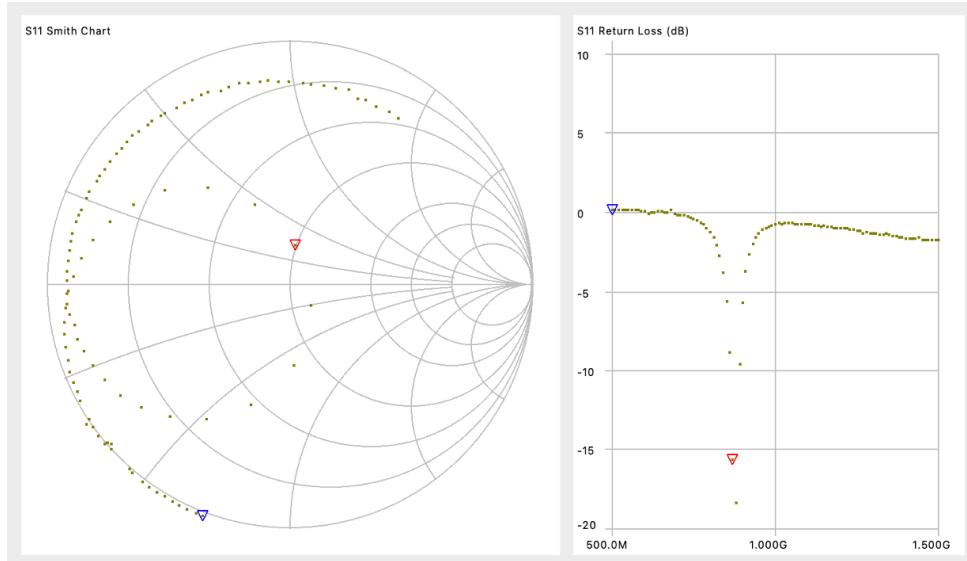


Figure 31: Impedance Matching: Smith chart (left), S11 return loss (right)

6.1.2 Range

An exhaustive range of tests across various environments is crucial for the development of a robust wake-up receiver. This section will concentrate on range evaluations to understand the operational effectiveness of wake-up receivers under different environmental conditions. Both indoor and outdoor scenarios are included in the evaluation to represent as wide a variety of applications as possible.

The indoor scenario involves testing within a typical two-room apartment, reflecting real-world conditions and offering a practical gauge for receiver performance. In contrast, the outdoor scenario aims to minimize interference, thereby determining the maximum achievable range of the receivers.

A pivotal element in these evaluations is the transmitter: an Arduino Shield backed with an SX1276 transceiver. The bit rate is set to 15.6 kHz for efficient decoding with the PIC microcontroller. The wake-up signal consists of a 6-bit preamble, followed by an 8-bit address field. This configuration facilitates up to eight different addresses due to the

implementation of "minimum energy coding".

In the test procedure, wake-up signals are transmitted at 100ms intervals. The wake-up signal (WuS) with the correct address is sent 500 times with a 10ms delay. The transmission power is fixed at 20dBm. The results of these evaluations will yield valuable insights into the real-world performance and robustness of the wake-up receiver designs under investigation.

In the test procedure, three different front ends are compared with one another. Firstly, the front end by Fromm et al. [19] including band-pass filter and low noise amplifier (LNA). The other two frontends are our design according to Magno et al. [13], but one with a quasi-perfect matching network with a return loss of -26.72dB and one with a non-perfect match with a return loss of -15.68dB.

Outdoor scenario

During the outdoor scenario, measurement cycles were conducted at one-meter intervals in an urban green space. Each cycle aimed to discern the number of messages from the total 500 that were accurately detected, incorrectly detected, or missed entirely. The results, depicted in figures [32][34], compare the performance across the three hardware variants under investigation.

The data reveals that our first design, featuring quasi-perfect matching, offers a reliable range of 2m and roughly 10% of the Wake-up Signals (WuS) correctly detected at a 5m distance. Observing the second design, the impact of matching quality becomes more pronounced. Due to its inferior matching compared to the first design, the effective range is shorter. At a 2m distance, only 65% of WuS are correctly detected, and beyond 4m, no WuS detections are recorded. Even though [13] claims a range of 50m for this front-end design, the evaluation conducted does not support this assertion. Despite the alterations undertaken, comparable results were expected. Factors such as the imperfect measurement environment, capable of introducing interference despite the open space, could be responsible for this discrepancy. Additionally, the original paper does not detail the specific measurement conditions employed. Improved results might be achievable through further fine-tuning of the matching network and detector circuits.

The design based on Fromm et al.'s work [19] exhibits the best performance. Reliable detection of wake-up signals is maintained up to 4m, with a significant decrease in accurate detections only beyond 7m. Although the original paper cites a sensitivity of -80dB (implying a higher maximum range), our results fall short of this. Owing to the lack of detailed manufacturing instructions for the circuit, it's plausible that the matching network was detuned during the reproduction process due to differing properties of the PCB and components. Nonetheless, a respectable range of 10m was achieved with this design, highlighting the advantages of SAW filters and Low Noise Amplifiers (LNAs).

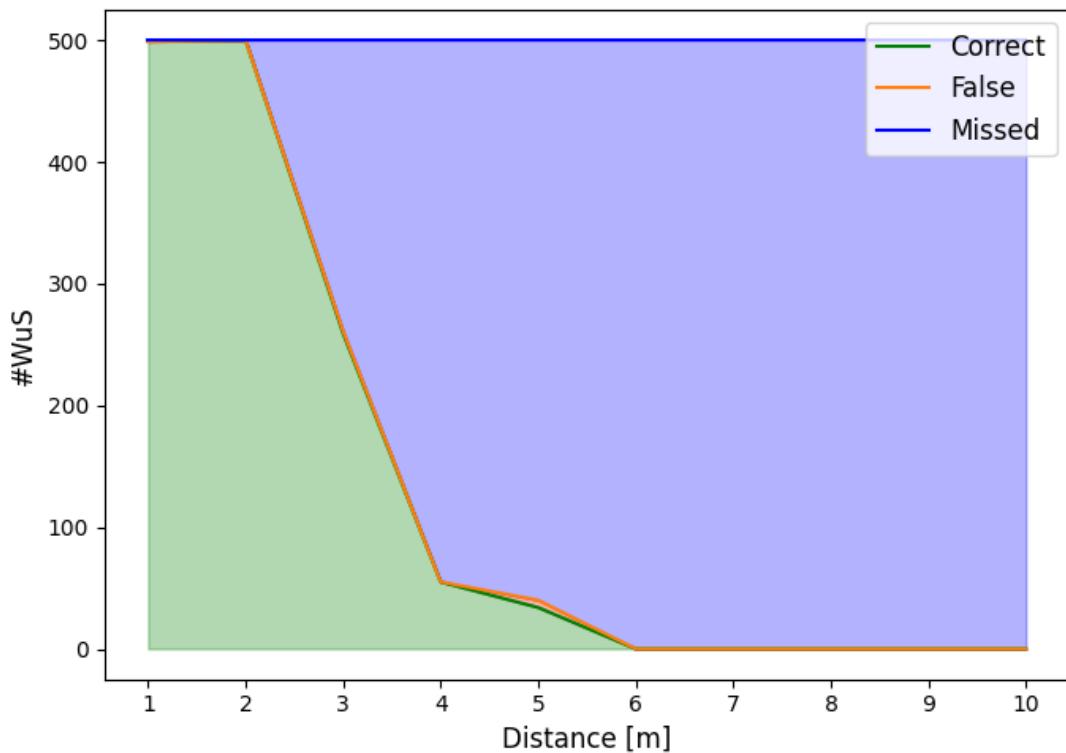


Figure 32: Range test results for our frontend design with quasi-perfect matching

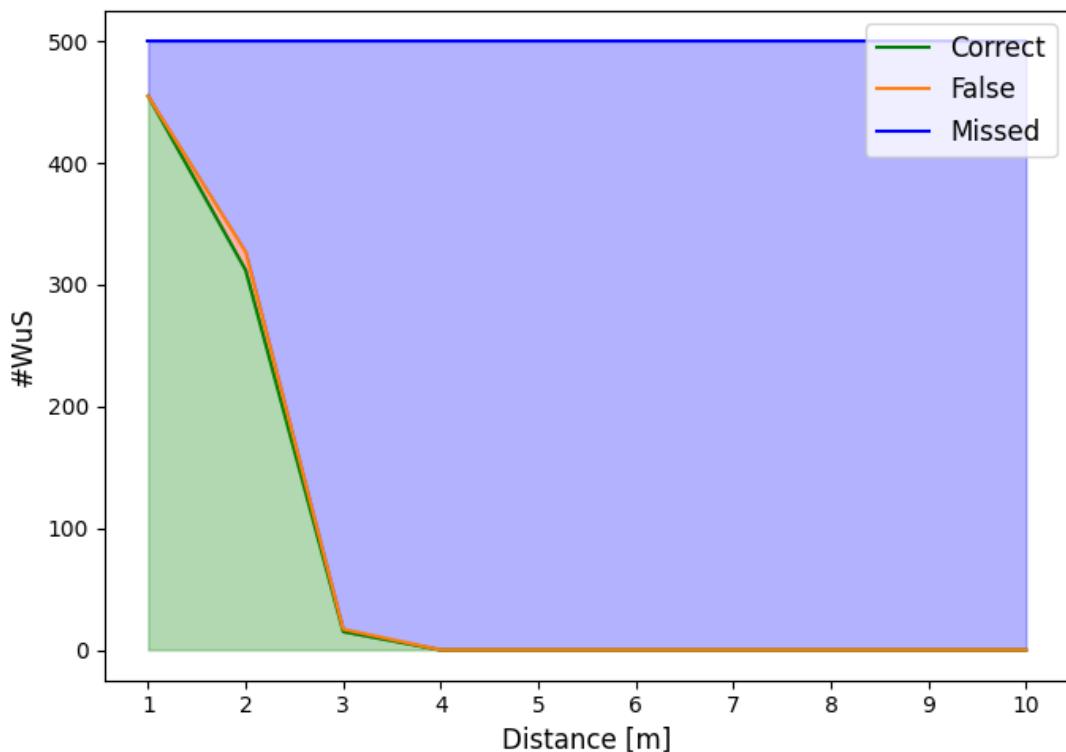


Figure 33: Range test results for our frontend design with non-perfect matching

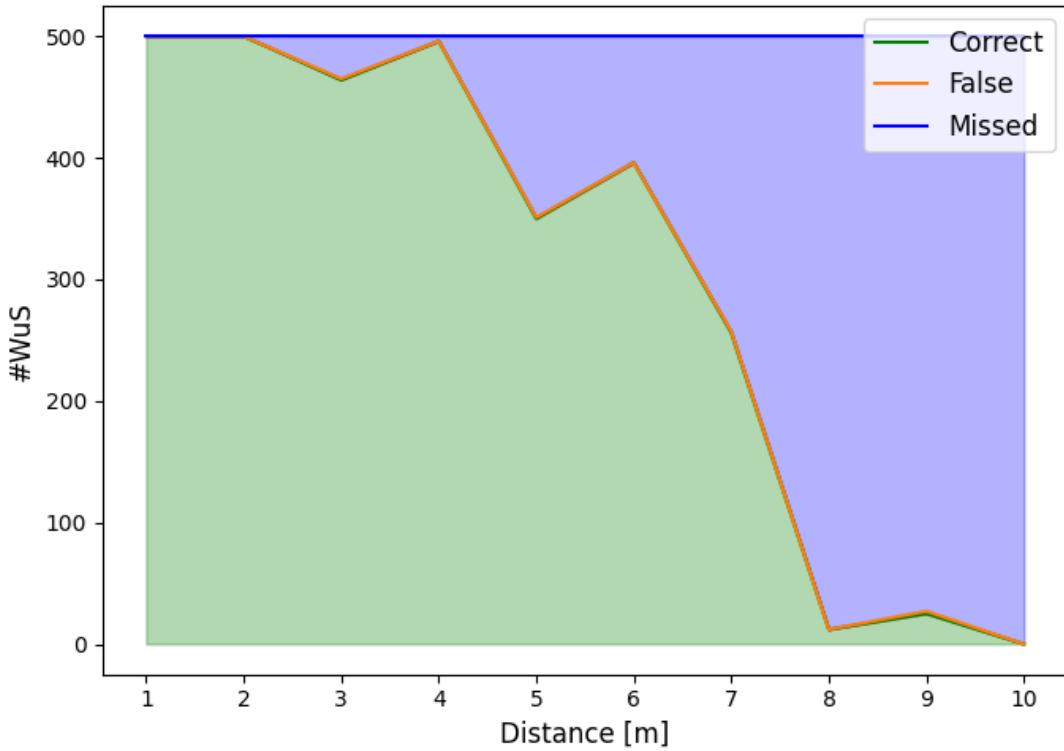


Figure 34: Range test results for Fromm et al. [19] design

Indoor scenario

The indoor scenario was created as a contrast to the outdoor scenario. For this purpose, different installations of receiver and transmitter positions were tested in a two-room flat. These are shown in illustration 35. This indoor scenario simulates how well the use of wake-up receivers works indoors. As in the outdoor scenario, a distinction is made between correct, false and missed detection.

The results are displayed in table 7 and reinforce the findings from the outdoor scenario.

| Position | OWN_1 | OWN_2 | FROMM |
|------------|-----------|-----------|----------|
| Position 1 | 500/0/0 | 235/2/263 | 499/1/0 |
| Position 2 | 248/1/251 | 0/0/500 | 500/0/0 |
| Position 3 | 124/2/474 | 0/0/500 | 500/0/0 |
| Position 4 | 10/0/490 | 0/0/500 | 66/0/434 |
| Position 5 | 0/0/500 | 0/0/500 | 23/1/476 |

Table 7: Signal detection for different positions and hardware. Values are given as Correct/False/Missed.

Situated approximately 3 meters away in the same room, Position 1 serves as the initial performance test for our designs. Our quasi-perfect match design and Fromm et al.'s design [19] detected almost all signals accurately. In contrast, our imperfect match variant decoded 235 messages correctly with minimal false detections.

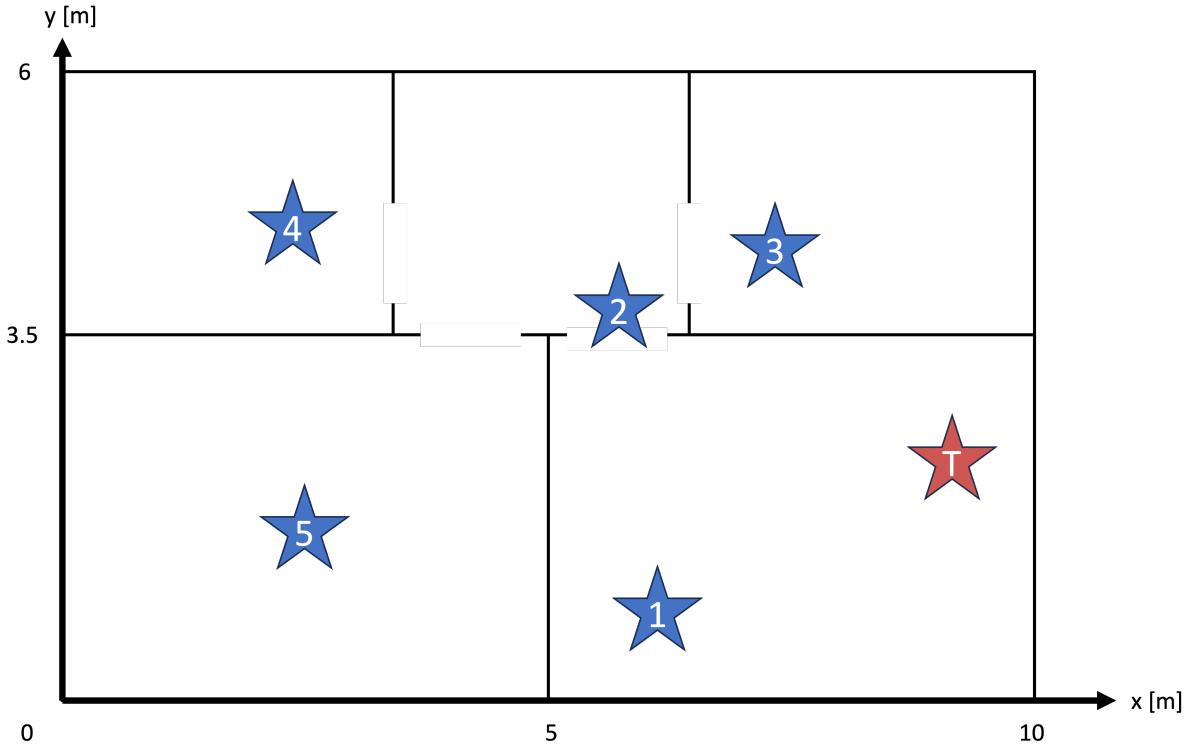


Figure 35: Transmitter positions (blue) relative to the wake-up receiver (red)

For Position 2 in the unobstructed hallway, our imperfect match design fell short, failing to decode any messages. Meanwhile, both the quasi-perfect match and Fromm et al.’s designs [19] managed to identify roughly half of the messages.

Position 3, located just around the corner, brought identical outcomes to Position 2. While our imperfect match design showed zero correct detections, both the quasi-perfect match and Fromm et al.’s designs [19] kept on reliably detecting the majority of the signals.

With increased distance at Position 4, the performance dwindled across the board, especially for our imperfect match design, failing yet again to recognize any signals. The quasi-perfect match design and Fromm et al.’s, although affected, were still able to recognize some signals.

The farthest and most challenging, Position 5, proved to be a stumbling block for all designs, particularly highlighting the deficiency of the imperfect match design, which along with Fromm et al.’s design failed to detect any correct signals. The quasi-perfect match design, though struggling, still managed to detect a few.

In summary, the indoor testing demonstrated the robustness of the Fromm et al. design and our quasi-perfect match design under varying conditions. In contrast, the design with the imperfect match showed substantial difficulty in correctly identifying messages unless conditions were close to ideal. These findings align with the outcomes from the outdoor testing.

Summarily, the indoor testing not only manifested the solid performance of the Fromm

et al. design and our quasi-perfect match design under assorted conditions, but also revealed the substantial challenges our imperfect match design encounters unless operating under near-ideal circumstances. These outcomes reaffirm the trends established during the outdoor testing. The indoor results were surprisingly satisfactory compared to the outdoor scenario. This suggests that despite the complexity and interference-rich environment of indoor settings, they may exert less detrimental impact on the range than initially expected. The primary obstacle observed for signal propagation appears to be solid walls. This insight underscores the fact that the indoor scenario, in spite of its inherent challenges, remains a viable environment for the deployment of wake-up receivers, provided the design considerations are suitably addressed.

6.1.3 Energy Consumption

The energy consumption of a wake-up receiver is a crucial determinant of its efficiency and practicability in real-world deployments. Given the often battery-powered or energy-harvesting nature of wireless sensor nodes, it is vital to ensure minimal power usage while maintaining satisfactory performance levels. This section delves into the energy consumption assessment of the developed wake-up receivers, examining their energy utilization profiles and the associated implications for their long-term operational viability. The evaluation involves a detailed analysis of the power consumption during different operational states, including idle, active and sleep states, and aims to offer insightful conclusions on the overall energy efficiency of the designs. This knowledge will facilitate the optimization of future iterations and enhancements.

The comprehensive analysis of a wake-up receiver's energy consumption involves considering all active components that contribute to the overall power utilization. In our evaluation, we focus on the energy usage of the PIC microcontroller and the comparator, as these are the main active components in our developed system. The front-end, on the other hand, is designed to be entirely passive, contributing no additional power requirements. Figure 36 provides a graphical representation of the current consumption profile over a typical wake-up cycle, including the MCU interrupt. These measurements were obtained under a supply voltage of 3.3V.

The active phase of the wake-up receiver spans $777\mu s$, with an average power consumption of $961.6 \mu A$. Notably, the preamble duration is $384\mu s$. Interrupt-induced peaks are observed, attributable to the comparator during preamble detection, which involves capacitor charging, and the microcontroller during the MCU wake-up interrupt.

As per the datasheet, the PIC12LF1552 consumes $30 \mu A/MHz$ at 16MHz, which calculates to a consumption of $480 \mu A$. When the TLV7031 comparator's output switches high, current flows through the preamble filter's 3.3k resistor, amounting to a current of 1mA. Consequently, the comparator's average current consumption is approximately $520 \mu A$.

In contrast, the sleep mode shows a significantly reduced average current consumption of $0.32 \mu\text{A}$. During this mode, the PIC only consumes 20nA , while the TLV7031 comparator consumes a larger 315nA . Thus, in sleep mode, the current consumption is largely driven by the comparator.

Evidently, energy loss is associated with comparator pulses. This could be mitigated by designing the preamble filter with a larger shunt resistor. Additionally, energy conservation could be achieved by reducing the active times, possibly by increasing the bit rate. However, with the PIC's clock rate already at 16MHz for a 5.6kHz bit rate, this presents a challenge. Lowering the PIC's clock rate could decrease power consumption during the active phase, but would require a reduced bit rate, forming a dilemma that likely already strikes a fair balance.

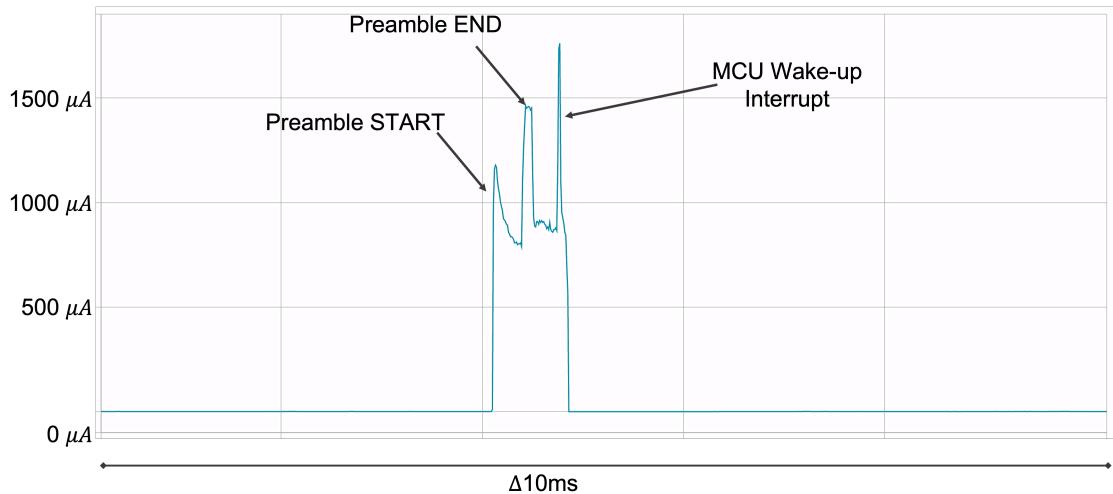


Figure 36: Energy consumption of one wake-up cycle

6.1.4 Delay

As discussed in Section 5.2.2, the delay from the rising edge of the address bit to the MCU wake-up interrupt was reduced to $8\mu\text{s}$. This technique greatly minimizes the wake-up delay, which is primarily dictated by the preamble duration and address transfer.

Given a bit time, T_{Bit} , of $64\mu\text{s}$, a preamble of 6 bits, a wait bit preceding the address, and up to an 8-bit address, the maximum wake-up delay, T_{delay_max} , is calculated as follows:

$$T_{delay_max} = (6 + 1 + 8) * 64\mu\text{s} + 8\mu\text{s} = 968\mu\text{s}$$

If the high bit of the address is not in the last position but on average at position 4.5, the mean wake-up delay, T_{delay_mean} , becomes $744\mu\text{s}$, as the interrupt is instantly triggered upon recognizing the high bit.

Consequently, the wake-up delay is not predictably static, and address assignment should prioritize those with the high bit upfront. The delay can be minimized by a higher bit rate, thereby reducing T_{Bit} , but this leads to challenges discussed in the earlier discourse on energy consumption.

6.2 User-friendliness

The development of an effective wake-up receiver (WuR) system requires careful consideration of several factors such as usability, reproducibility, affordability, and modularity. In our design and development process, we have endeavored to balance these aspects, aiming for a system that is user-friendly, cost-effective, easily reproducible, and modular. Above all, a central aspect of our approach is transparency: all design and development files associated with our system are readily available and comprehensively documented. This is intended to facilitate reproduction of our work and to provide a springboard for further advancements in the field.

Our wake-up receiver system is designed with an emphasis on usability. A key consideration has been to create a system that can operate effectively without the need for expensive equipment once the matching process is completed. Our design is characterized by simplicity, utilizing mainly passive components to reduce complexity and ease the deployment process.

The reproducibility of our wake-up receiver system largely depends on the design of the printed circuit board (PCB). In this iteration, we use a 4-layer PCB as it simplifies the matching process of the microstrip to 50Ω , enhancing the reliability of the system. However, we propose to explore the use of a 2-layer PCB in future iterations to potentially lower the manufacturing cost, while bearing in mind that this could lead to a wider trace on the front-end.

Also affordability of our wake-up receiver system is of utmost importance for its widespread adoption. Our design strategy seeks to balance complexity and cost, primarily employing a minimal number of components to keep the overall price low. With cost of the PCB at 2.45€ and the costs for the PIC and the comparator the over all costs stay below 5€. This approach also contributes to enhanced reproducibility and usability. We have designed our system with a high degree of modularity, aiming for compatibility with any microcontroller capable of being awakened via interrupt. Also the 3.3V supply voltage allows an easy integration with pre-existing sensor nodes. Even within the wake-up receiver itself, the front-end is kept separate from the decoder, which provides flexibility for the user. For example, the decoder can be set up on a breadboard, while the front-end—which necessitates a PCB—remains on the board.

Our system is also designed for quick deployment. The process involves ordering and

soldering the PCB, after which the system can be promptly activated. The programming of the PIC microcontroller, usually requiring a specific programmer, can also be achieved indirectly using simple Arduino boards. The provided PIC code simply needs to be uploaded to the system, streamlining the deployment process.

7 Conclusion and discussion

The essence of this research centers around the imperative challenge of transforming Wake-up Receiver (WuR) technology from a primarily research-based concept to a user-friendly, accessible platform for a broader audience. While WuR technology has been a topic of scholarly interest for over a decade, it remains largely confined to academic exploration, with scarce open-source resources available for developers outside the research community. The primary objective of this work was to bridge this significant gap through the development, testing, documentation, and comparison of an open-source, user-friendly WuR system, empowering a diverse range of users to implement this technology across various wireless sensor networks (WSNs) applications.

A thorough investigation was conducted, leading to the successful design and testing of a unique WuR system. This system not only integrates a fully passive front-end but also demonstrates an efficient method to decode messages through a combination of a comparator and an ultra-low-power 8-bit microcontroller. Exhibiting a range of a few meters and a wake-up delay under 1ms, the system is well-equipped for use in various projects. Importantly, the research provides a clear pathway for enhancing the range of the system in the future.

The outcomes of this research offer significant implications for both the scientific and developer communities. A robust foundation has been laid, demonstrating the potential for carrying out open science projects in complex research areas. This work showcases the transformative power of taking a theoretically complex research topic and translating it into a practical open-source project, bringing WuR technology from laboratories to the desks of hobbyists and professionals alike.

However, like all research, this study is not without its limitations. The current range of the system is confined to only a few meters, and the power consumption could be optimized by using a higher resistor in the preamble filter. Additionally, the bit rate is limited due to the 16MHz constraint of the 8-bit microcontroller. These limitations offer valuable insights and directions for future improvements, paving the way for subsequent research to enhance the system's range, power efficiency, and data rate.

In conclusion, this study has successfully brought WuR technology closer to a broader audience, marking a significant step towards the widespread adoption of this technology in various practical applications. The open-source, user-friendly WuR system designed in this work promises to stimulate innovation, inviting developers to explore, adapt, and enhance this technology for a multitude of Wireless Sensor Networks applications. This work marks a significant contribution to the fields of wireless communication and open science, demonstrating that complex research topics can indeed be made accessible and user-friendly.

7.1 Outlook

The work presented here has opened up new avenues for the application and enhancement of Wake-up Receiver (WuR) technology. However, this research is just the starting point. Here are a few, but by no means all, exciting opportunities for future exploration.

1. **Use Case Development:** Developing concrete use cases for the WuR technology could help to illustrate its potential in real-world applications and encourage its adoption across various domains.
2. **Software Implementation:** Further software development, especially involving widely used microcontrollers like Arduino UNO or ESP32, would increase the accessibility and ease of use of the WuR system.
3. **PCB Manufacturing:** Exploring partnerships with PCB manufacturers like JL-CPCB would make it easier for users to order a pre-manufactured version of the WuR system, significantly reducing the barrier to entry.
4. **Hardware Expansion:** Enhancing the hardware by adding components like Surface Acoustic Wave (SAW) filters and Low Noise Amplifiers (LNA) could lead to improved performance of the WuR system.
5. **Shielding Options:** The exploration of shielding options, such as the use of shield cans, could help to improve signal integrity and reduce interference.
6. **2-Layer PCB:** The use of a 2-layer PCB would not only be more cost-effective but also allow for a more compact design, enabling all components to fit on a single PCB if desired.
7. **Further Simplification:** Implementing configurations through an I2C interface could make the WuR system even more user-friendly. This would allow users to easily adjust settings such as bit rate and node address using an Arduino, eliminating the need to modify the PIC code.

All these and more ideas can be found as todos in the projects [github](#). With continued research, innovation, and community-driven development, it is poised to transform the landscape of wireless sensor networks and beyond. The journey has just begun...

List of Tables

| | | |
|---|---|----|
| 1 | Overview of off-the-shelf component Wake-up receiver literature | 16 |
| 2 | Performance of off-the-shelf component Wake-up receiver literature | 16 |
| 3 | Mapping of source symbols to mapped symbols in Minimum Energy Coding | 22 |
| 4 | Overview of per board cost of Impedance Matching Prototyping board . . . | 38 |
| 5 | Overview of per board cost of WuR Frontend board inspired by [19] | 40 |
| 6 | Comparison of 8-bit MCUs | 49 |
| 7 | Signal detection for different positions and hardware. Values are given as Correct/False/Missed. | 61 |

List of Figures

| | | |
|----|---|----|
| 1 | A Typical WSN deployment | 10 |
| 2 | Functional Overview of Wake up Receiver | 12 |
| 3 | Overview of sensor node modules | 13 |
| 4 | WuR structure | 17 |
| 5 | Greinacher Voltage Doubler | 18 |
| 6 | Comparison NRZ to Manchester encoding | 22 |
| 7 | Minimum Energy Coding (left) and Minimum Energy Coding with early shutdown (right) | 23 |
| 8 | Preamble Filtering | 24 |
| 9 | SPICE simulation of WuR frontend circuit of [13] | 34 |
| 10 | SPICE simulation results for SMS7630 as envelope detector. Comparator output signal (top/cyan), comparator input signals: IN+ (middle/blue) and IN- (middle/red), simulated OOK signal (bottom/green) | 35 |
| 11 | SPICE simulation results for HSMS-285C as envelope detector. Comparator output signal (top/cyan), comparator input signals: IN+ (middle/blue) and IN- (middle/red), simulated OOK signal (bottom/green) | 36 |
| 12 | WuR Z-Parameter Prototyping Board Schematic with SMS7630 | 37 |
| 13 | WuR Z-Parameter Prototyping Board Layout with SMS7630 | 37 |
| 14 | WuR Impedance Prototyping Board Schematic with SMS7630 | 37 |
| 15 | WuR Impedance Prototyping Board Layout with SMS7630 | 37 |
| 16 | PCB for Z-parameter measurement | 38 |
| 17 | Wake-up Receiver circuit board 3D animation based on Fromm et al. [19] | 40 |
| 18 | Wake-up Receiver circuit schematic based on Fromm et al. [19] | 41 |
| 19 | Wake-up Receiver Frontend + 868MHz Antenna | 43 |
| 20 | Channel 1 (blue): Envelope of Wake-up Signal | 44 |
| 21 | Channel 1: Envelope of WuS (non-inverted comparator input); Channel 2: Low-pass filtered envelope of WuS (inverted comparator input) | 46 |
| 22 | Channel 1: Envelope of WuS (non-inverted comparator input); Channel 2: Comparator output | 46 |
| 23 | Channel 1: Low-passed comparator output as preamble detector 2: Comparator output | 48 |
| 24 | Comparator circuit with preamble filter | 48 |
| 25 | Wiring of PIC12LF1552 with comparator TLV7031 | 50 |
| 26 | Connections between Wake-up Receiver Modules and MCU | 50 |
| 27 | Channel 1: Envelope of the WuS; Channel 2: Timer Interrupts | 53 |
| 28 | Channel 1: Envelope of WuS; Channel 2: Generated Interrupt | 54 |
| 29 | Timing in Address Decoding | 55 |

| | | |
|----|---|----|
| 30 | Reduced interrupt time based on optimized code | 56 |
| 31 | Impedance Matching: Smith chart (left), S11 return loss (right) | 58 |
| 32 | Range test results for our frontend design with quasi-perfect matching . . . | 60 |
| 33 | Range test results for our frontend design with non-perfect matching . . . | 60 |
| 34 | Range test results for Fromm et al. [19] design | 61 |
| 35 | Transmitter positions (blue) relative to the wake-up receiver (red) | 62 |
| 36 | Energy consumption of one wake-up cycle | 64 |

References

- [1] K. Sohraby, D. Minoli, and T. Znati, Wireless sensor networks: technology, protocols, and applications. John wiley & sons, 2007. [9], [10], [13]
- [2] A. Deshpande, C. Guestrin, S. R. Madden, J. M. Hellerstein, and W. Hong, “Model-driven data acquisition in sensor networks,” in Proceedings of the Thirtieth international conference on Very large data bases-Volume 30, pp. 588–599, 2004. [9], [10]
- [3] J. Rabaey, “Ultra-low power computation and communication enables ambient intelligence,” in Proceedings of the Smart Objects Conference, Grenoble, 2003. [9]
- [4] L. Xu, R. W. Collier, and G. M. P. O’Hare, “A survey of clustering techniques in wsns and consideration of the challenges of applying such to 5g iot scenarios,” IEEE Internet of Things Journal, vol. 4, pp. 1229–1249, 2017. [10]
- [5] M. Sharma, A. Tandon, S. Narayan, and B. Bhushan, “Classification and analysis of security attacks in wsns and ieee 802.15.4 standards : A survey,” 2017 3rd International Conference on Advances in Computing,Communication & Automation (ICACCA) (Fall), pp. 1–5, 2017. [11]
- [6] R. Piyare, A. L. Murphy, C. Kiraly, P. Tosato, and D. Brunelli, “Ultra low power wake-up radios: A hardware and networking survey,” IEEE Communications Surveys & Tutorials, vol. 19, no. 4, pp. 2117–2157, 2017. [11], [12]
- [7] S. Bdiri, F. Derbel, and O. Kanoun, “An 868 mhz 7.5μ w wake-up receiver with-60 dbm sensitivitiy,” Journal of Sensors and Sensor Systems, vol. 5, no. 2, pp. 433–446, 2016. [12], [16], [17], [18], [19]
- [8] K. Akkaya and M. Younis, “A survey on routing protocols for wireless sensor networks,” Ad hoc networks, vol. 3, no. 3, pp. 325–349, 2005. [13]
- [9] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, “Wireless sensor networks: a survey,” Computer networks, vol. 38, no. 4, pp. 393–422, 2002. [13]
- [10] L. K. Ketshabetswe, A. M. Zungeru, M. Mangwala, J. M. Chuma, and B. Sigweni, “Communication protocols for wireless sensor networks: A survey and comparison,” Heliyon, vol. 5, no. 5, p. e01591, 2019. [13]
- [11] C.-Y. Chong and S. P. Kumar, “Sensor networks: evolution, opportunities, and challenges,” Proceedings of the IEEE, vol. 91, no. 8, pp. 1247–1256, 2003. [13]

- [12] A. M. Zungeru, L.-M. Ang, and K. P. Seng, “Classical and swarm intelligence based routing protocols for wireless sensor networks: A survey and comparison,” *Journal of Network and Computer Applications*, vol. 35, no. 5, pp. 1508–1536, 2012. [13]
- [13] M. Magno, V. Jelicic, B. Srbinovski, V. Bilas, E. Popovici, and L. Benini, “Design, implementation, and performance evaluation of a flexible low-latency nanowatt wake-up radio receiver,” *IEEE Transactions on Industrial Informatics*, vol. 12, no. 2, pp. 633–644, 2016. [15], [16], [17], [18], [19], [32], [33], [34], [43], [44], [45], [47], [49], [57], [59], [70]
- [14] G. U. Gamm, M. Sippel, M. Kostic, and L. M. Reindl, “Low power wake-up receiver for wireless sensor nodes,” in *2010 Sixth International Conference on Intelligent Sensors, Sensor Networks and Information Processing*, pp. 121–126, IEEE, 2010. [16], [17], [18]
- [15] J. Oller, I. Demirkol, J. Casademont, J. Paradells, G. U. Gamm, and L. Reindl, “Performance evaluation and comparative analysis of subcarrier modulation wake-up radio systems for energy-efficient wireless sensor networks,” *Sensors*, vol. 14, no. 1, pp. 22–51, 2013. [16], [17]
- [16] P. Woias, S. Heller, and U. Pelz, “A highly sensitive and ultra-low-power wake-up receiver for energy-autonomous embedded systems,” in *Journal of Physics: Conference Series*, vol. 1052, p. 012024, IOP Publishing, 2018. [16], [17], [19]
- [17] D. Galante-Sempere, D. Ramos-Valido, S. Lalchand Khemchandani, and J. Del Pino, “Low-power rfed wake-up receiver design for low-cost wireless sensor network applications,” *Sensors*, vol. 20, no. 22, p. 6406, 2020. [16], [17]
- [18] R. Fromm, L. Schott, and F. Derbel, “An efficient low-power wake-up receiver architecture for power saving for transmitter and receiver communications.,” in *SENSORNETS*, pp. 61–68, 2021. [16], [17], [18]
- [19] R. Fromm, O. Kanoun, and F. Derbel, “Reliable wake-up receiver with increased sensitivity using low-noise amplifiers,” in *2022 19th International Multi-Conference on Systems, Signals & Devices (SSD)*, pp. 1523–1528, IEEE, 2022. [16], [17], [18], [19], [32], [33], [39], [40], [41], [45], [57], [59], [61], [62], [69], [70], [71]
- [20] I. Ketata, S. Ouerghemmi, A. Fakhfakh, and F. Derbel, “Design and implementation of low noise amplifier operating at 868 mhz for duty cycled wake-up receiver front-end,” *Electronics*, vol. 11, no. 19, p. 3235, 2022. [17], [18]
- [21] A. Galoić, B. Ivšić, and D. Bonefači, “Rectifier for energy harvesting application,” in *2017 International Symposium ELMAR*, pp. 137–140, IEEE, 2017. [18]

- [22] B. Van der Doorn, W. Kavelaars, and K. Langendoen, “A prototype low-cost wakeup radio for the 868 mhz band,” *International Journal of Sensor Networks*, vol. 5, no. 1, pp. 22–32, 2009. [\[20\]](#)
- [23] L. W. Couch, M. Kulkarni, and U. S. Acharya, *Digital and analog communication systems*, vol. 8. Citeseer, 2013. [\[21\]](#)
- [24] N. E. H. Djidi, M. Gautier, A. Courtay, and O. Berder, “Enhancing wake-up receivers reliability through preamble filtering and minimum energy coding,” in *2021 IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 1–5, IEEE, 2021. [\[22\]](#), [\[23\]](#), [\[24\]](#)
- [25] N. E. H. Djidi, M. Gautier, A. Courtay, and O. Berder, “Mees-wur: Minimum energy coding with early shutdown for wake-up receivers,” *IEEE Transactions on Green Communications and Networking*, vol. 5, no. 3, pp. 1502–1513, 2021. [\[22\]](#), [\[23\]](#)
- [26] S. Bdiri, “Wake-up receiver for ultra-low power wireless sensor networks,” 2021. [\[29\]](#)
- [27] Avago Technologies US., *Surface Mount Microwave Schottky Detector Diodes in SOT-323 (SC70)*, 1998. [\[34\]](#)
- [28] Skyworks Solutions, Inc., *SMS7630 - Surface Mount Mixer and Detector Schottky Diodes*, 2013. [\[34\]](#)