Driving and Testing Crystal-free Radios for a Basic End-to-End Transmission

Lorenz Schmid
Master Thesis

Section of Electrical Engineering,
Electronics and Microelectronics
École Polytechnique Fédérale de Lausanne

written at:

The Ubiquitous Swarm Lab,
Berkeley Sensor & Actuator Center
Electrical Engineering and Computer Science
University of California, Berkeley

under the supervision of:

Prof. Kristofer S.J. Pister
Prof. Yusuf Leblebici

August 14, 2015

Abstract

Wireless sensor networks consist of a collection of small, inexpensive, independent nodes, capable of sensing various properties of their environment, and connected together in a wireless network. In order to make them autonomous, non-invasive and cheap, a monolithic integration is envisioned. The renunciation of off-chip components strongly affects the radio design of such a node which usually relies on an external quartz crystal as frequency reference. Using an on-chip frequency reference such as a ring oscillator leads to a high phase noise profile. Little is known about the impact of high phase noise on a continuous radio transmission and its understanding is therefore crucial to be able to design a monolithic sensor node.

In the following work a complete macro-scale test system consisting of a digital transceiver integrated on a FPGA controlled by software, a custom-designed MSK modulation PCB with a PLL, and an RF radio circuit using minicurcuits components is presented. It follows the IEEE 802.15.4 standard and emulates a radio communication at 2.4 GHz between two monolithic sensor nodes relying on frequency references with a high phase noise profile of -80 dBc at 1 MHz frequency offset. Based on the test system a successful transmission with a data rate of 100 kbps has been demonstrated, remotely controlling small MEMS structures. The receiver's demodulation PLL was able to track a 500kHz frequency drift of the radio signal and could keep up the transmission for multiple hours. No significant signal power degenerations as side-effect of high phase noise exposure were observed. This proofs the feasibility of a radio transmission with a high phase noise profile frequency reference and thus the design of monolithic sensor nodes. Furthermore, the demodulation structure based on a PLL revealed the ability to track frequency instabilities. The made test system lays the foundation for further testing and measuring of additional parameters of monolithic sensor nodes.

Acknowledgment

Foremost, I would like to express my sincere gratitude to my supervisor, Prof. Kristofer S.J. Pister, for welcoming me in his research group, for his scientific guidance and excellent inputs during many discussions, and the great insights into his visionary research. I would also like to thank my supervisor, Prof. Yusuf Leblebici, for his help and support.

My sincere thank goes also to Dr. Osama Khan, who gave me scientific guidance, helped me to overcome multiple obstacles and provided excellent explanations for an easier approach on the subject. Furthermore, the many scientific discussions, contributions and friendships made with the entire research group, namely David Burnett, Mike Lorek, Filip Maksimovic, Travis Massey, Sahar Mesri and Brad Wheeler, who I would like to thank as well.

Also, I would like to thank Simone Benatti, who introduced me to the PCB design, and Gylfi Þór Guðmundsson, who helped me correcting this thesis, as well as the Ubiquitous Swarm Lab staff and members for their support.

Finally, I would like to thank wholeheartedly my family, namely Ingrid, Nora and Thomas Schmid, Tatiana Guimarães Ferreira, as well as my friends for their endless support during so many years. Thank you all very much.

Contents

List of Figures

List of Tables

Abbreviations

WSN Wireless Sensor Network

SoC System on Chip

IoT Internet of Things

IC Integrated Circuit

FPGA Field-Programmable Gate Array

PCB Printed Circuit Board

MEMS Microelectromechanical System

PHY Physical Layer

MAC Medium Access Control Layer

TCP Transmission Control Protocol

IP Internet Protocol

WPAN Wireless Personal Area Network

RF Radio Frequency

IF Intermediate Frequency

ISM Industrial, Scientific and Medical Radio Band

FSK Frequency-Shift Keying Modulation

BFSK Binary Frequency-Shift Keying Modulation

MSK Minimum Shift Keying Modulation

PSK Phase-Shift Keying Modulation

QPSK Quadrature Phase-Shift Keying Modulation

O-QPSK Offset Quadrature Phase-Shift Keying Modulation

Tx Radio Transmitter

Rx Radio Receiver

LO Local Oscillator

VCO Voltage Controlled Oscillator

PLL Phase-Locked Loop Control System

PFD Phase Frequency Detector

LPF Low Pass Filter

BPF Band Pass Filter

HPF High Pass Filter

PA Power Amplifier

LNA Low Noise Amplifier

SNR Signal-to-Noise Ratio

PPDU PHY Protocol Data Unit

SHR Synchronization Header

SFD Start-of-Frame Delimiter

PHR PHY Header

PSDU PHY Service Data Unit

CRC Cyclic Redundancy Code

IRQ Interrupt Request

ISR Interrupt Service Routine

UART Universal Asynchronous Receiver/Transmitter Protocol

SPI Serial Peripheral Interface Bus Protocol

ADC Analog-to-Digital Converter

DAC Digital-to-Analog Converter

SAR Successive Approximation Register

DMA Direct Memory Access Controller

FIFO First In, First Out Memory

LDO Low-Dropout Regulator

1 | Introduction

Great progress in the miniaturization of electronics has opened new applications to discover. Combined with the ability to treat large amounts of data and the increasing interest in doing so, sophisticated sensor systems are in high demand.

Wireless Sensor Networks (WSNs) consist of a collection of small, inexpensive, independent nodes, capable of sensing various properties of their environment, and connected together in a wireless network. Each node includes a power source, a processor, some form of sensor, and a radio. In order to keep maintenance cost low, ultimately, the goal is to not need any external cable connection to provide power nor to exchange data. Ongoing miniaturization allows to integrate all these functions on a single chip at a low cost. Because of their small size and low maintenance cost, such sensor nodes can be spread as numerous and as non-invasive as dust. A first proposal for a such a system coning the expression "Smart Dust" was made in [1]. It opens new possibilities of measuring such as given in [2] as "tracking the movements of birds, small animals, and even insects, fingertip accelerometer virtual keyboards, monitoring environmental conditions affecting crops and livestock, and inventory control".

Despite the technology available today, designing such a monolithic system is still challenging. Limited in size and relying on wireless communication naturally leads to large power restrictions. The following section highlights the biggest design challenges. Explications are given on how this work tries to get a better understanding of such limited circuits. The results promise to lead to a better design.

1.1 Motivation

Naturally, ultra low power is the key for long term operation of a wireless sensing node. This puts taxing constrains on the radio units as its power consumption is proportional to its ability to communicate. Especially, as such nodes are deployed in today's crowded radio environment which relays heavily on wireless communication such as for mobile phones or Internet. Next to these high-power radio exchanges it is difficult to maintain a low-power communication without loosing the signal.

Furthermore, by integrating everything into a single chip, there can be no off-chip components such as a quartz oscillator. Missing a stable frequency reference leads to frequency inaccuracy and instability. This makes it hard to tune to a narrow band in busy radio environments. In order to build monolithic low power nodes the design of its radio unit is therefore a key point.

In [3] multiple approaches to replace quartz crystals with on-chip frequency references have already been studied and tuned LC tanks and ring oscillators been proposed as replacement. Generally, such solutions lead to a high phase noise profile which inevitably degrades a radio's performance. Little is known so far about the effect of these degenerations on a continuous radio transmission. Understanding the effect of a high phase noise profile is crucial to make a working radio.

The following work proposes an extendible test system which allows to study the functioning of a crystal-free radio at a macro-scale level. Focus is set on the degeneration effects caused by a high phase noise profile. The feasibility of a sustainable radio communication will be verified.

1.2 Thesis Outline

This thesis is organized in six chapters. The theory behind the design of a crystal-free radio will be summarized and the different levels of the test system used to emulate such a radio will be explained and analyzed.

In Chapter 2, the necessary background knowledge about Wireless Sensor Networks is given and the concepts and challenges of building a monolithic node for such a network are outlined. The main issues of relying on a low quality frequency reference are highlighted.

The remaining chapters explain and analyze a test system which verifies the ability to communicate wirelessly based on a high phase noise profile:

In Chapter 3, the existing digital system, consisting of a digital transceiver, a microprocessor, and its corresponding software interfacing the radio hardware, is described.

In Chapter 4, different modifications of the digital system are proposed. Additional extensions and their implementations to increase control and improve the robustness of the radio communication are presented.

In Chapter 5, the macro-scale crystal-free emulating radio which is connected to the digital transceiver is presented and the feasibility of a sustainable radio communication is analyzed based on different measurements.

In Chapter 6, a conclusion is given followed by a discussion on future work or possible improvements.

2 | Wireless Sensor Network

This chapter gives a necessary background knowledge about Wireless Sensor Networks (WSNs) and the ultimate goal of building a monolithic node for such a network. Following chapters will refer back to the here introduced concepts. Since WSN technology covers multiple areas, the many necessary concepts can only be briefly introduced. A deep analysis is not in the scope of this work and interested readers are encouraged to consult the given references.

WSNs consist of a collection of small, inexpensive, independent nodes, capable of sensing various properties of their environment, and connected together in a wireless network. Its goal is simple: Building small and fully integrated sensing nodes leads to non-invasive deployment with a very low price tag. Maintenance costs are kept low by using energy harvesting and wireless communication between the sensing nodes. Such a low-cost sensing network offers a full range of new possibilities for measuring and sensing. It replaces the current approach of a few, expensive sensing stations and allows instead to measure over large areas and in locations that are difficult access. With the current approach such applications were unrealistic or only feasible at high cost.

In order to be independent and providing multiple functions a wireless sensor node consists out of multiple components:

- Microprocessor with memory,
- Power source such as a battery or a form of energy harvesting,
- Radio transceiver with internal antenna,
- Sensor(s), and
- Actuators such as MEMS.

Its sensors can keep track of different parameters like temperature, humidity

or movement. Autonomous and continuous functioning is guaranteed by energy harvesting or long lasting batteries. Control commands and measured data are collaboratively transfered through the sensor network via radio. A gateway node will link the WSN with the external world.

Since wireless sensing nodes are limited in size and rely on wireless communication they have large power restrictions. The exchange in the network is therefore based on a low-rate and low-power communication. Current wireless network protocols used by mobile phones or for Internet applications, generally focus on high data throughput. Such protocols have a high degree of complexity which reflects in power-hungry designs. Therefore, their use is not suitable for WSNs. The IEEE 802.15.4 standard tries to offer an alternative protocol for low-rate Wireless Personal Area Networks (WPANs) and is further explained in section 2.4 at the end of this chapter.

2.1 State-of-the-Art

WSNs are already in place today. Generally, a node consists of multiple Integrated Circuits (ICs) mounted on a single PCB – an approach which is not area nor power-efficient. With the rise of the Internet of Things (IoT) integrated solutions, combining radio and microprocessor on a single chip or at least into a single packaging, are more and more common. Still, focus is generally on a direct communication with the client device using technologies such as WiFi or Bluetooth. Despite recent efforts to increase power efficiency these protocol still prioritize data throughput over power consumption.

One of the first chips dedicated for WSNs using the efficient IEEE 802.15.4 protocol and integrating radio and microprocessor into a single chip was made by *Dust Networks* (*Linear Technology* today) [4]. With an area of around 1 cm^2 its chips communicate on the 2.4 GHz Industrial, Scientific and Medical (ISM) radio band and require off-chip components such as antenna, passive components, crystal and power source. Its power consumption is around 1.5 mW for transmitting at 0 dBm and it has a receiver sensitivity of -93 dBm.

In academic research approaches for ultra-low power transceiver for WSN have been presented in [5] and [6] respecting the IEEE 802.15.4 standard and keeping consumption as low as 1.12 mW for transmitting at -5 dBm with receiver sensitivities below -94 dBm. Further, their research has been focusing on limiting off-chip components and designing a low-voltage transceiver

which could be powered by a small on-chip solar cell.

More recently a power-harvesting radio without need for external components has been presented in [7]. The proposed radio design has an area of a few square-millimeters consuming as little as 1.5 μW operating in the 24 and 60 GHz ISM band.

2.2 Crystal-Free Ultra Low-Power Radio

As seen in the previous section, current solutions for WSN rely on external components such as batteries, crystals, passive components or antennas. This has a large impact on the area requirements of a single node, increasing manufacturing and deployment costs. Therefore, Prof. Pister's research group, of which the author of this work is part of, is targeting the design of a monolithic sensor platform which envisions zero off-chip components. The beginning of chapter 2 lists the different components contained on a wireless sensor node. The goal of a monolithic node is now to integrate all these function into a single chip:

Microprocessor As microprocessor an *ARM Cortex M0* will be fully synthesized and integrated on the chip. *ARM*'s microprocessors are well known for their versatile application and low-power usage and very common in portable devices such as smartphones. The *Cortex M0* series represents the smallest *ARM* processor available with focus on a low power application. As network stack the use of *OpenWSN* is envisaged and currently under active development by a joint effort of the EVA team of *Inria* in France and Prof. Pister's research group. A first implementation of such a microprocessor on a FPGA with basic interface circuitry and driving low-level software has already been completed and is analyzed in chapter 3.

Power Source The development of high energy microbatteries is ongoing. Currently an areal density of 4.1 mWh cm^{-2} can be achieved as presented in [8]. Ongoing research on energy harvesting with CMOS solar cell by Prof. Pisters research group resulted in solar cells with an areal power density of 15 mW cm^{-2} (unpublished).

Sensing Analog-to-Digital Converters (ADCs) are key components for sensing as they allow to sample measurement signals and thus to convert them

into the digital domain. Generally, the design of ADCs focus on speed and resolution and not, as required by WSN, on minimum energy consumption. Previous work done in [9] by Prof. Pister's research group presents a low-energy 8-bit SAR ADC with a consumption of 3.1 μW at a sample rate of 100 kHz. Development of a low-energy 10-bit SAR ADC is currently ongoing.

Radio As will be presented in the next two subsections, the radio is a key component in the design of a monolithic sensing node due to its relatively large power consumption. Furthermore, seeking a monolithic design the missing external frequency reference has a large impact on the radio's performance.

2.2.1 Power Budget

A closer look at the power consumption of the different components of a wireless sensing node is presented in [10] and shown in figure 2.1. The Radio Transmitter (Tx) and Radio Receiver (Rx) are clearly responsible for the vast majority of the consumption and therefore most crucial for the design of an efficient node.

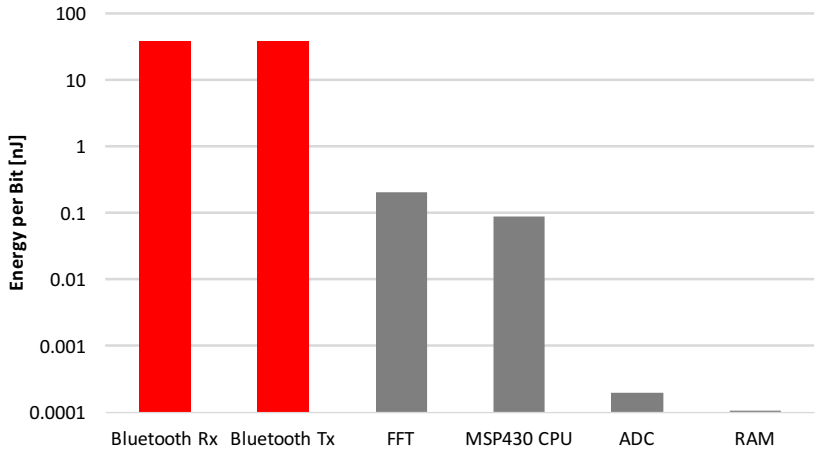


Figure 2.1: Energy consumption partition of different tasks in a wireless sensing node; as given by [10]

Naturally, the power consumption can be reduced by decreasing the transmitting power or the receiver sensitivity which in turn degrades the

transceiver's performance. This leads to multiple trade-offs making the design of low power radios non-trivial. A general overview on low energy radio transceiver design is given in [11].

Focusing on a monolithic sensor platform, its power source must be integrated on the same substrate as the circuit of the node. A long term target of autonomous sensors with energy harvesting leads therefore to a trade-off between size of the node and power consumption. Figure 2.2 shows different ambient sources of energy. Radiant sources such as the sun clearly provide the highest power density and a solar cell of 1 cm^2 generates even indoors a minimum of around 100 μW.

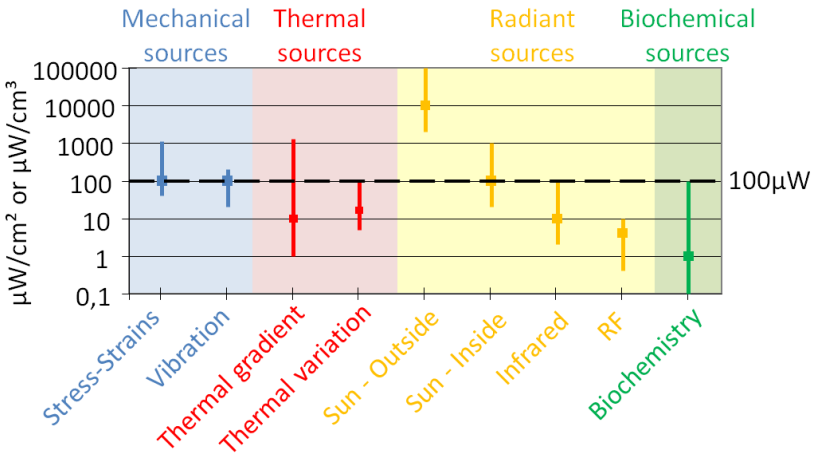


Figure 2.2: Power density of different ambient sources for energy harvesting; as shown in [12]

Based on these insights the ambitious goal of a new radio design consuming less than 100 μW with a Rx sensitivity of -85 dBm was formulated and is currently in development in Prof. Pister's research group. Once completed it will be integrated together with the previously mentioned components. The overall size of the sensing node should be less than a few square-centimeters.

2.2.2 Crystal-Free Radio

Building a monolithic sensing node not only puts large restriction on the power consumption but also envisions the renunciation of any off-chip components. Again, the restrictions affect the radio unit the most:

The most important component in a radio is its frequency reference, an oscillator circuit. Based on this reference the radio generates the necessary frequencies required to transmit or receive a signal on a specified frequency band. Using a unstable or inaccurate reference, the radio is not able to track and tune to the desired bands and therefore unable to communicate. Unfortunately, one of the most stable frequency reference is built using a quartz crystal, a component which can not be integrated on a chip. As seen in section 2.1 so far all WSN solution relay on (external) crystals.

Generally, the quality of an oscillator is measured with the quality factor Q. It describes the relation between an oscillator's center frequency f_c and its bandwidth Δf as shown in the following equation:

$$Q = \frac{f_c}{\Delta f} \tag{2.1}$$

Figure 2.3 gives an visualization of the Q factor showing a typical frequency spectrum as measured at an oscillator's output. As can be seen, the Q factor is an indicator for the sharpness of the generated frequency peak. Generally, the size of electrical components is proportional to the Q factor leading to a trade-off between area and frequency quality.

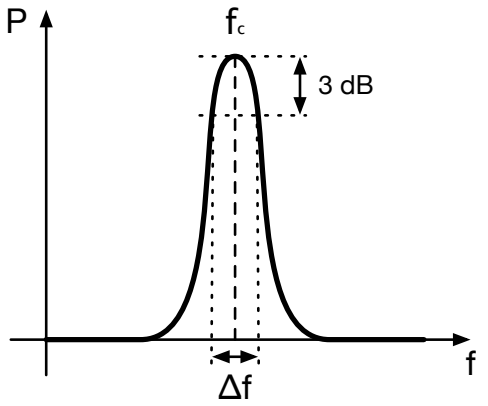


Figure 2.3: Quality factor

Following a list of the most common electrical oscillators. The list is sorted in ascending order of the Q factor. A more detailed analysis of the different oscillators can be found in [3] and [13].

Ring Oscillator is a digital circuit consisting of an odd amount of inverters connected in series forming a loop. Due to the small propagation delay the signal is switching the state of every inverter consecutively creating therefore an oscillation.

Tank Oscillator consists out of an energy restoration and a RLC-circuit. Once a potential is applied, the RLC-circuit starts to oscillate spontaneously shifting energy back and forth between its capacitor and its inductor. During this process some of the energy is lost due to the resistor and the lossy nature of the components. This energy is restored by the energy restoration circuit.

MEMS Resonator are small on-chip structures which oscillate mechanically. These oscillations can be translated into the electrical domain. As of today, MEMS resonator are still in a research phase but promise to be a suitable and power-saving on-chip alternative in the near future.

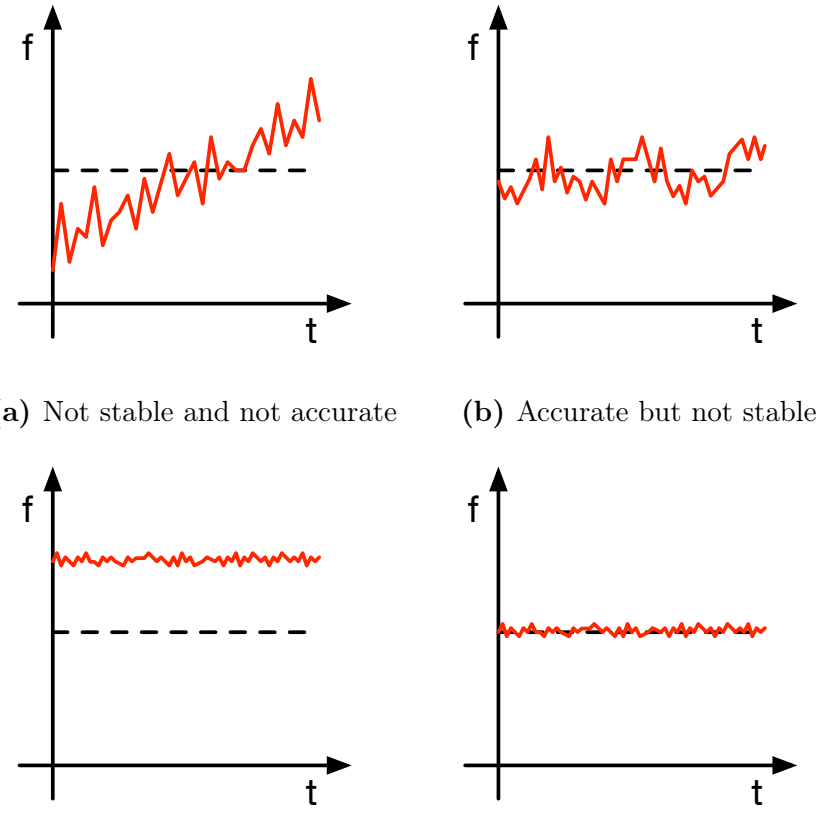
Crystal Oscillator is similar to a tank oscillator. The RLC circuitry is replaced by a quartz crystal resonating based on a piezoelectricity effect. Due to its properties it is very temperature stable.

Definitely, there are alternatives to build on-chip oscillators. Unfortunately, their Q factor is orders of magnitudes lower than what a oscillator based on a quartz crystal has to offer. This lower quality leads to multiple degeneration effects concerning the produced reference frequency. Following are the most important degeneration effects caused by a low Q factor:

Frequency Accuracy describes the ability to tune to a specific radio channel. It is an indicator for the linearity between the input control signal and the output frequency of the oscillator.

Frequency Stability describes the ability to stay tuned to a channel and its frequency over time.

Figure 2.4 shows different examples to outline the difference between frequency accuracy and stability. The dashed line indicates the desired frequency output.



(a) Not stable and not accurate

(b) Accurate but not stable

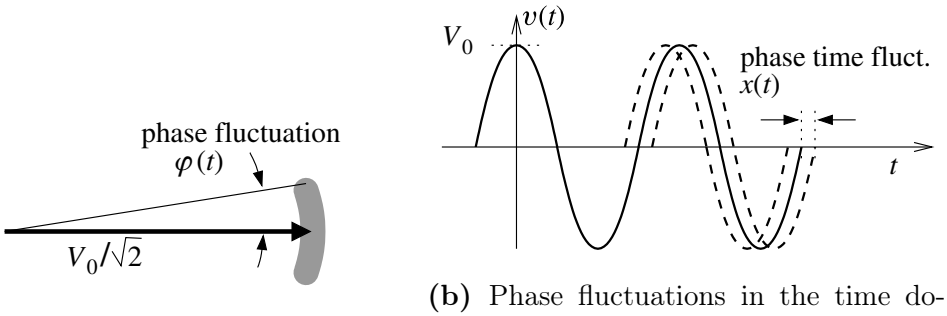
(c) Stable but not accurate

(d) Accurate and stable

Figure 2.4: Frequency accuracy and stability

Phase Noise and Timing Jitter is an effect caused by rapid, random phase fluctuations which translate into time fluctuations [13], so called jitter. A visualization is shown in figure 2.5. The introduced phase noise $\mathcal{L}$ is measured in dBc as the ratio between the power of the signal at the carrier frequency and the noise power. High phase noise in the signal spectrum leads to a higher bit error rate in the transmission and therefore a decrease of performance.

Phase noise in oscillators is a complex phenomenon that has been studied closer in [3]. There, next to maximizing the quality factor, an increase of signal power of the oscillator is recommended. This leads to a trade-off between area/power and phase noise as already outlined in the previous section.



(a) Phase fluctuation

(b) Phase fluctuations in the time domain

Figure 2.5: Phase fluctuation leading to phase noise and time jitter; as given by [13]

Phase noise is a combination of different noise components which is visible in the phase noise profile of an oscillator as shown in figure 2.6. These noise components (f^0, f^{-1}, f^{-2}, ...) are inversely proportional to different exponents of the frequency offset Δf_c. Close-in phase noise is dominated by higher order components such as f^{-3}. As revealed in [3] the corner frequency of these higher order components can be largely reduced by a precise stimulation of the oscillator. In order to emulate a specific phase noise profile the random walk noise f^{-2} is therefore most important noise component.

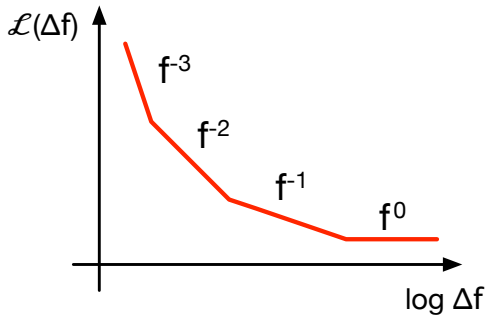


Figure 2.6: Oscillator phase noise profile; white noise f^0, flicker noise f^{-1}, random walk noise f^{-2}; as given by [13]

Timing Synchronization is important in networks with independent nodes. In order to save power the nodes are turned off most of the time.

Only during fixed defined intervals the nodes wake up to reinforce the network. In order to wake up at the right moment, timing is thus crucial. Having an internal frequency reference running too fast or too slow can lead to a misalignment of the wake-up intervals and thus, the node loses connection with its network.

While timing synchronization can be solved from a software side as it is currently implemented in *OpenWSN*, frequency accuracy and stability are very design and environment specific. Valid conclusions can only be drawn once implemented on-chip. Phase noise is probably the degradation effect which is affected the most by a low quality factor. Therefore, it is important to understand its effect on a reliable radio communication in order to be able to build a monolithic sensor platform.

2.3 Crystal-Free Emulating Macro-Scale Radio

The radio circuit is a key component when building a monolithic sensor platform as highlighted in the previous section. Using crystal-free, low quality frequency references leads to a high phase noise profile which degenerates the radio's performance dramatically. In order to understand better the effect of such a degradation, a test system consisting of a macro-scale radio model is presented here. The goal is to verify, if a sustainable radio transmission is possible despite a high phase noise profile frequency reference. The obtained insights can then be used to improve the chip design of the actual radio before passing into manufacturing.

In the next subsections an overview about specification and architecture of the test system is given. The remaining chapters of this work will then explain its different components in detail.

2.3.1 Test System Specification

First simulation results of the phase noise profile of an on-chip ring oscillator are shown in figure 2.7. The test system must therefore have a similar phase noise profile of approximately $\mathcal{L} = -80$ dBc at $\Delta f = 1$ MHz. Since the test system is build out of macro-scale components in order to reduce development time, its power performance will not be comparable to an actual on-chip implementation.

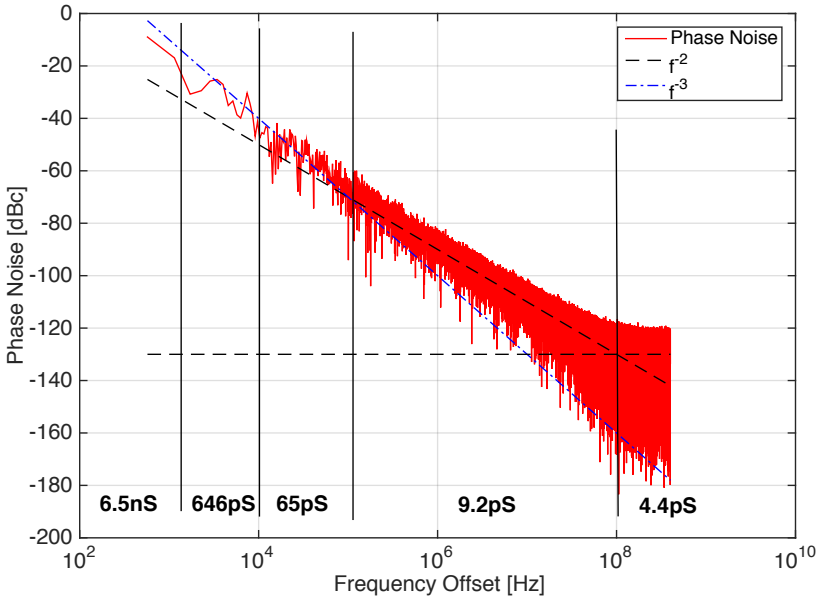


Figure 2.7: Simulated ring oscillator phase noise profile

The final design of the monolithic sensing node should be compatible with existing WSNs using the common IEEE 802.15.4 standard which is explained in more detail in section 2.4. Based on this standard the node and therefore also the test system have to respect following specifications in terms of architecture:

- 2.4 GHz communication on ISM band
- 250 kbps data rate
- Minimum Shift Keying modulation
- Compatible digital Tx and Rx

2.3.2 Test System Architecture Overview

Since the final on-chip design of the radio circuit is not defined yet, a standard radio structure is chosen for the test system. An overview is shown in figure 2.8, a photo of the final hardware system is presented later in the result section in figure 5.10. The test system consists of:

- a digital system interfacing the radio circuit with the controlling computer (explained in chapters 3 and 4),

- a (de)modulation step translating the digital bit stream into a radio signal (section 5.2), and
- the actual radio circuit with a low-quality frequency reference responsible for the wireless transmission (section 5.3).

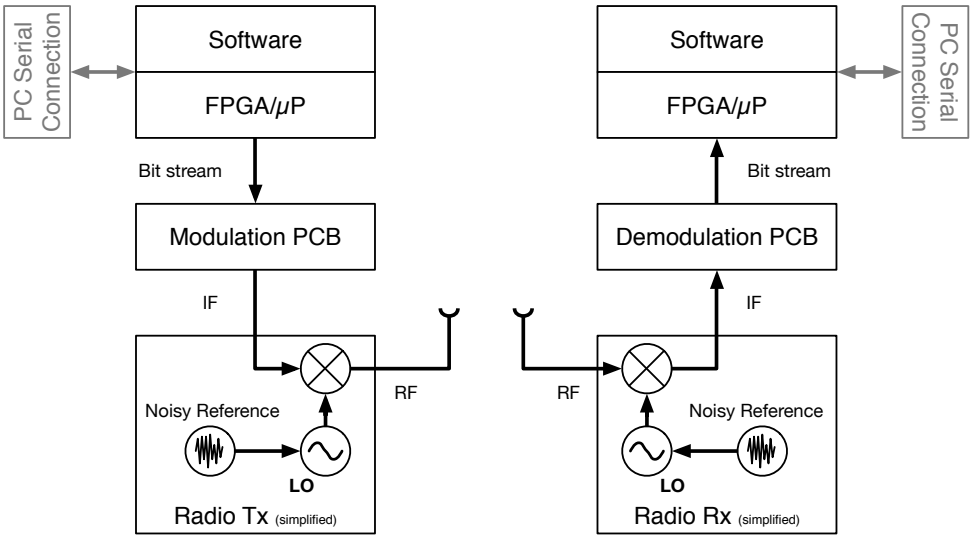


Figure 2.8: Overview on test system architecture

The high phase noise profile of a crystal-free frequency reference will be emulated by injecting noise into a free-running Voltage Controlled Oscillator (VCO) which acts as a Local Oscillator (LO). The relation between white noise on a VCO voltage control line and its resulting phase noise profile is given in [14] as following

$$\mathcal{L}(f) = \frac{\kappa_V^2}{4f^2} S_{V_c}(f) \tag{2.2}$$

where $\mathcal{L}(f)$ is the phase noise density, f the frequency offset of the oscillator reference, κ_V is the sensitivity of the VCO, and $S_{V_c}(f)$ the mean-square input voltage noise density of the voltage control signal.

2.4 IEEE 802.15.4 Standard

Currently the monolithic sensing nodes are in the design phase. One of the goals is to make them compatible to currently existing WSNs which rely on

the IEEE 802.15.4 standard to communicate. This standard is comprised of the lower network layers following the OSI model defining the PHY and data link (MAC) layer of a type of WPAN. Focus is on low-cost and low-rate communication between close-range (< 10 m) low-power nodes. Different well-known standards such as ZigBee or WirelessHART are based on it. An ultimate goal, especially under the rise of the IoT, is the compliance of the MAC layer to the TCP/IP as standardized by IETF.

Following, the most important specification of the Physical Layer (PHY) layer as defined in the IEEE 802.15.4 standard [15] are presented here. Specifications of the MAC layer will be omitted since they target mainly software interfaces and are therefore not part of this work.

2.4.1 Frequency Band

IEEE 802.15.4 defines multiple frequency bands in which a standard compliant device can operate. Targeting a worldwide application the only unlicensed radio band available is the ISM band at 2400–2483.5 MHz. It contains 16 channels with a 5 Mhz channel spacing whereby 2 MHz are occupied per channel. The transmit center frequency tolerance has a maximum of ± 40 ppm. The data rate for this band is specified as 250 kbps.

2.4.2 Modulation

The translation of a binary data stream into a transmittable radio signal is called modulation. This requires a mapping of the binary data into symbols which are then translated into a modulated radio signal. Thereby a binary subsequence of length k is mapped into one of m symbols. Logically, one needs $m = 2^k$ symbols to map all possible combinations of sequences.

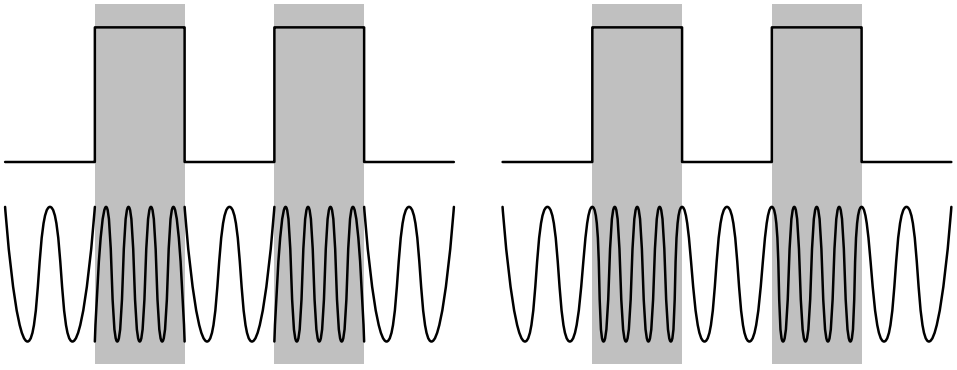
Transmitting a signal naturally leads to degeneration due to the imperfection of the chosen communication channel. The transmitted data can undergo attenuation, distortion, interference, and be exposed to noise. In order to minimize these undesirable effects different types of modulations are available.

Low-data rate signals, as used by the IEEE 802.15.4 standard, naturally require a smaller bandwidth for transmission. Therefore, there is no particular need for a spectral efficient modulation scheme and frequency modulations such as Frequency-Shift Keying (FSK) become attractive due to their simple and low cost implementation requiring less linear components [16].

Following different frequency modulation schemes are presented. Their advantages and disadvantages as well as their relation to each other is highlighted. It will be shown that the simple MSK modulation can produce equal signals as the O-QPSK modulation which is required by the IEEE 802.15.4 standard for the 2.4 GHz band.

Frequency-Shift Keying (FSK) is a frequency modulation, where every data symbol is represented by a different frequency. In its simplest form using only two symbols ($m = 2$), a binary '0' is represented with one frequency (f_0) and a binary '1' with another (f_1) as shown in figure 2.9. f_0 and f_1 are centered around the carrier frequency f_c whereby $|f_1 - f_0| = \Delta f$. This form is also called Binary Frequency-Shift Keying (BFSK) modulation.

Depending on its implementation two independent oscillators can be used to generate the two frequencies. If the phases of the two oscillators are not aligned, switching between the two frequencies results in signal with a non-continuous phase. These sharp transitions require a higher bandwidth and are therefore spectral inefficient. An example of a signal with non-continuous phase is shown in figure 2.9a. A close look at the corner between the white and gray areas indicates these transitions. More commonly used is a single oscillator whose output frequency is controlled via a voltage control signal, a so called VCO. This leads to a continuous phase as shown in figure 2.9b.



(a) Noncontinuous phase **(b)** Continuous phase

Figure 2.9: Binary FSK modulation; digital signal on the top, modulated signal on the bottom

Minimum Shift Keying (MSK) is a type of continuous-phase FSK modulation. A continuous phase is guaranteed between signals with a harmonic frequency, so called orthogonal signals. Defining two signals $s_m(t)$ and $s_n(t)$ with a frequency $f_m = m \cdot R$ and $f_n = n \cdot R$ respectively where R is the data rate, m, n some positive integer and the frequency difference $\Delta f = |f_n - f_m|$, the two signals are orthogonal if

$$\langle s_m(t), s_n(t) \rangle = 0 \tag{2.3}$$

Following equation (3.2.-57) in [17] the scalar product in equation 2.3 is defined as

$$\langle s_m(t), s_n(t) \rangle = 2\varepsilon R \int_0^{\frac{1}{R}} e^{j2\pi(m-n)\Delta f t}\, dt$$

and thus

$$\Re\left\{\langle s_m(t), s_n(t) \rangle\right\} = 2\varepsilon \operatorname{sinc}\left(\frac{2}{R}(m-n)\Delta f\right) \tag{2.4}$$

Equation 2.3 holds therefore true if equation 2.4 is equal zero leading to $(2/R)(m-n)\Delta f = 0$ for all $m \neq n$. This is the case if $\Delta f = (kR)/2$ with k some positive integer. The frequency difference Δf between the two harmonic signals s_m and s_n is thus minimal if $k = 1$ leading to the most spectral efficient FSK modulation. The frequency difference Δf is thereby half the data rate R.

Phase-Shift Keying (PSK) is a phase modulation, where every data symbol is represented by a change in phase.
A simple wave $s(t)$ with a carrier frequency f_c and time-varying phase $\Phi(t)$ can be split into a so called in-phase (I) and quadrature (Q) component:

$$s(t) = \sin\left(2\pi f t + \Phi(t)\right) = \underbrace{\sin(2\pi f t) \cdot \cos\left(\Phi(t)\right)}_{I} + \underbrace{\cos(2\pi f t) \cdot \sin\left(\Phi(t)\right)}_{Q} \tag{2.5}$$

Each of the two components can vary in phase and magnitude. For a PSK modulation the magnitude remains constant. Using different pairs of I and Q values different symbols can be formed and their translation into a radio signal is always defined by equation 2.5. Figure 2.10 shows an example with four symbols ($m = 4$; 00, 01, 10, 11) consisting of two bits each represented in the IQ domain. Every symbol occupies one quadrant. Since the magnitude remains constant, all symbols are placed on a circle and only the angle to the

x-axis (phase offset Φ) defines the current symbol. This modulation is also called Quadrature Phase-Shift Keying (QPSK). Its implementation is more complex compared to a FSK modulation since it requires the generation of the signal's I and Q components.

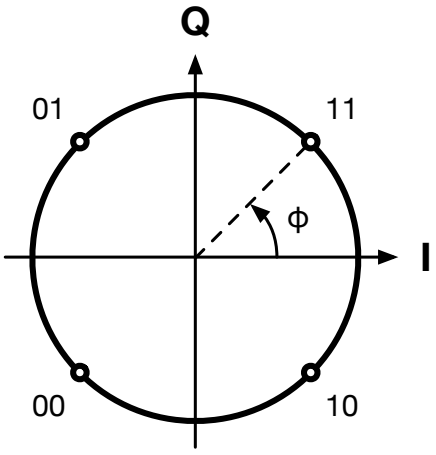
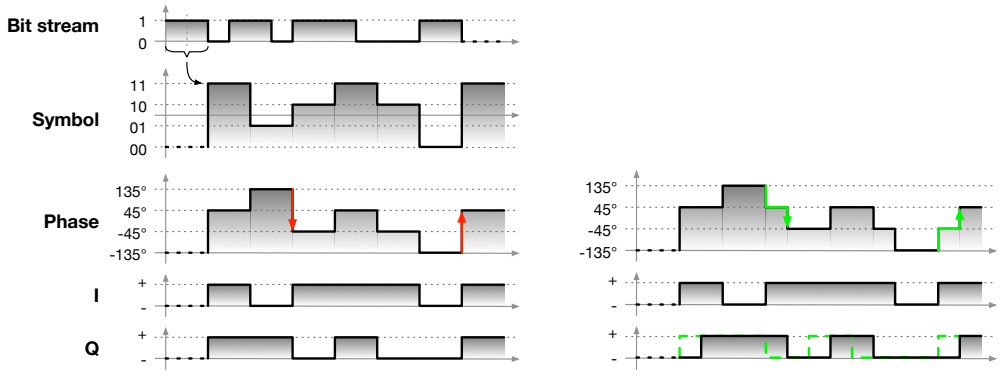


Figure 2.10: QPSK modulation shown in IQ plane

Offset Quadrature Phase-Shift Keying (O-QPSK) is a type of PSK modulation extending the QPSK modulation. It is used by the IEEE 802.15.4 standard for the 2.4 GHz band. In a QPSK modulation two consecutive bits are always translated into a symbol which then defines the output for the duration of the next two bits. Depending on the bit stream this output signal can contain maximum phase changes of 180° as shown in red in the phase signal on figure 2.11a. These rapid amplitude changes are undesirable in radio systems using low pass filters, leading to ringing and instability of the signal.

The QPSK signal's phase remains constant during one symbol (two bits). By offsetting the I and Q wave by half a symbol (one bit), the two added waves never change phase at the same time leading to a maximum phase change of 90° at a time. Such a modulation is called Offset Quadrature Phase-Shift Keying (O-QPSK) and its difference to QPSK is shown in green in figure 2.11b.



(a) QPSK **(b)** O-QPSK

Figure 2.11: Flow and comparison of two modulations

O-QPSK implemented as MSK is demonstrated in [16]: The bit stream signal in figure 2.11a is equivalent to the voltage control signal of a VCO generating a MSK signal as will be explained in section 5.2.1. Every transition indicates a frequency change. Comparing this bit stream and the I and Q signal of the O-QPSK modulation as shown in figure 2.11b the following correlation can be found as shown in [16]:

$$k = I \oplus Q \tag{2.6}$$

with k as current bit value during even clock intervals, and

$$k = \overline{I \oplus Q} \tag{2.7}$$

during odd clock intervals. Therefore, the MSK modulation can, with help of an additional coding step, lead to an equivalent signal as the O-QPSK modulation which is required by the IEEE 802.15.4 standard. Relying on the spectral-efficient and less complex MSK modulation scheme, the implementation can be simplified leading to a lower power consumption.

2.4.3 Data Structure

The PHY layer specification of the IEEE 802.15.4 standard contains next to definitions about the radio signal also definitions on the structure of the actual bit stream. Data is bundled into packets or so called PHY Protocol

Data Units (PPDUs), its structure is shown in figure 2.12. A PPDU contains a preamble for detection of the packet through correlation and a so called Start-of-Frame Delimiter (SFD) for synchronization purposes forming together the Synchronization Header (SHR). The PHY Header (PHR) indicates the length of the payload, the actual data to be transmitted. The payload, also called PHY Service Data Unit (PSDU), can have a maximum length of 127 octets and may contain a Cyclic Redundancy Code (CRC) for verification purposes.

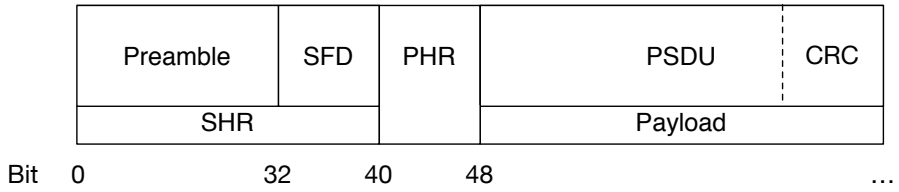


Figure 2.12: PPDU structure

One or multiple PPDUs form the so called bit stream. Before it is passed to the transmitter for transmission the individual bits are first translated into chip sequences, pseudo-random sequences replacing 4 bits by 32. This redundancy allows the final signal to contain a more equilibrated average power and makes it more robust. Figure 2.13 shows the individual translation steps performed in the digital domain before passing the signal to the radio transmitter.

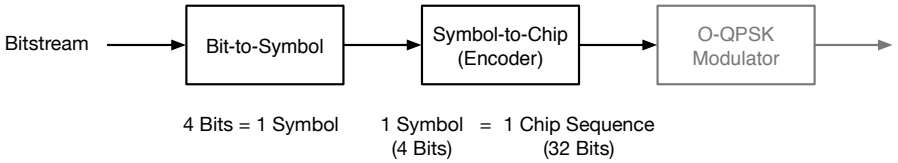


Figure 2.13: Bitstream modulation in IEEE 802.15.4

3 | Digital System Overview

This is the first chapter describing the test system. It explains the digital system which acts as an interface between the external controlling computer and the radio circuit.

The digital part of the test system consists of a *Xilinx Spartan 6* FPGA on which an *ARM Cortex M0* microprocessor is integrated together with multiple digital modules. The modules are written in Verilog, translated into an RTL architecture upon synthesis and loaded into the FPGA. They act as hardware interface between the microprocessor and external circuits such as the radio or the ADC. These external circuits are currently modeled with macro-scale IC components while their full custom chip design is ongoing. Once the design is completed the digital system as it is on the FPGA will be synthesized to complete the chip design. The FPGA offers therefore a flexible way to test the digital system without the need of any chip manufacturing.

The FPGA is placed on a *Digilent Nexys 3* development board containing multiple connectivity options such as USB or GPIOs which simplifies interfacing the external circuits. The system as described in this chapter was set up and tested previously to the start of this project.

This chapter is split into two parts explaining first the hardware structure on the FPGA and how the different digital modules are connected to the microprocessor. Then, in the second part, the software running on the microprocessor and its interaction with the underlaying hardware is briefly presented. Based on this setup, a wired transmission of simple messages between two FPGAs is explained.

3.1 Hardware System

Figure 3.1 gives an overview about the different modules integrated on the FPGA. Since this work focuses on the digital radio, only a fraction of the available functionalities are used. Non-relevant modules and buses are shown hatched. Further explanations concerning the complete system can be found in [18].

The different modules and the microprocessor are connected together via *AMBA* bus, a standard defined by *ARM* for its microprocessor. This simplifies development of additional modules by defining the interface connections every module must have. The microprocessor itself sees only multiple registers at different addresses into which it can write and read. The *AMBA* bus then translates these register operations into an actual access of the addressed module.

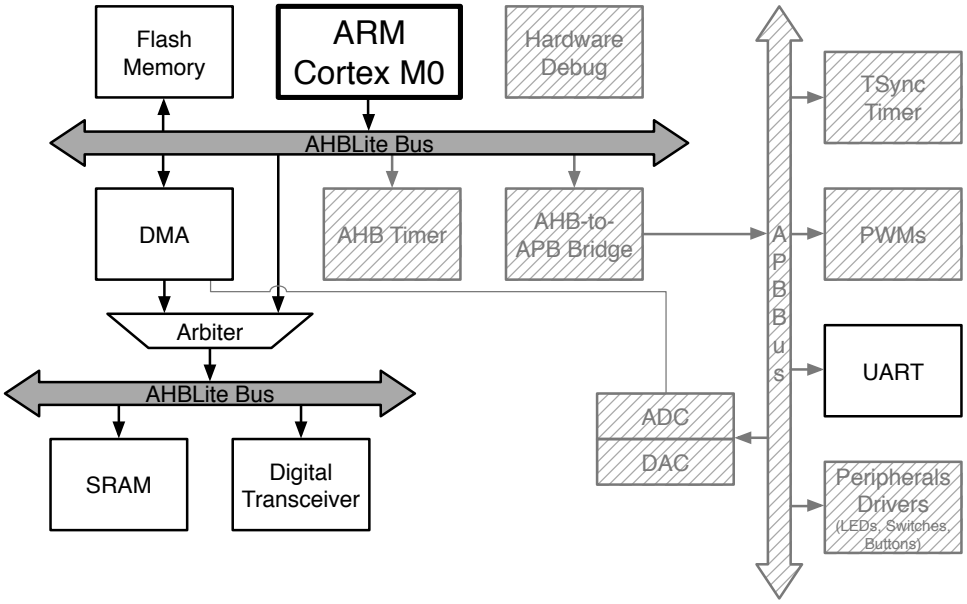


Figure 3.1: Digital hardware structure; non-relevant blocks for this project are shown hatched; based on [18]

Next to the microprocessor and the digital transceiver module two memory units can be found: The Flash memory contains the software code running on the microprocessor which is explained in more details in section 3.2.

The SRAM memory contains space for temporary variables of the software execution and stores PPDUs received or to be transmitted by the digital transceiver module. A Direct Memory Access (DMA) unit allows to offload the microprocessor of shifting data between the digital transceiver and the SRAM memory.

In figure 3.1, arrows connecting modules and buses do not indicate the data flow but show instead which modules act as bus master (arrow outgoing from a module) and which act as slaves (arrow ingoing to a module). The system contains two bus masters, the microprocessor and the DMA unit. In order to be configured by the microprocessor, the DMA unit also contains a slave interface next to its bus master interface indicated with a second arrow. Due to the limitations of the educational *AMBA* bus version used for this project, only one bus master is allowed per bus. Since the system relies on two bus masters, the bus is split up: One bus connects the transceiver and SRAM memory which allows to store received messages quickly. Both bus master can access this bus via an arbiter administrating the access. The other bus connects the remaining modules with the microprocessor. Here, the microprocessor acts as the only bus master.

3.1.1 Digital Transceiver

The digital transceiver is a module linking the microprocessor and the external radio circuit. Connected on the lower bus shown in figure 3.1, it can be accessed by the microprocessor for control purposes and by the DMA in order to shift incoming and outgoing messages from and to the SRAM. The digital transceiver contains both, a Radio Transmitter (Tx) as well as a Radio Receiver (Rx) architecture of which only one can be active at a time (half-duplex communication). Its simplified structure is shown on the bottom in figure 3.2 with the Tx configuration on the left and the Rx configuration on the right.

The Tx is responsible for generating a bit stream out of a simple message and transmit it to the external radio circuitry. The message is originally stored in the SRAM memory by the microprocessor. It then gets transfered via DMA to the Tx which adds a SFD, a PHR and a CRC sequence to the message forming a simplified PPDU structure following the IEEE 802.15.4 standard as described in section 2.4.3. Once the simplified PPDU is formed the resulting bit stream is stored in a FIFO and ready for transmission. It is important to point out that the simplified PPDU does not contain any

preamble nor is it encoded with chip sequences. The bit stream contains therefore no redundancy leading to a higher chance of actual bit errors in transmission and might not be equilibrated by containing long sequences of only '0s' or '1s'. It is planed to add this functionalities in the future with a more complex transceiver.

On the Rx side, once active, it is constantly looking for a SFD sequence indicating an incoming message. Since the Rx contains only a synchronization (start detection) unit, this SFD sequence must match perfectly. If only one bit of the SFD sequence differs, the whole PPDU will be discarded and is lost. This problem will be solved in the future by adding a preamble detection based on correlation with a threshold, which allows a less error-prone reception and completes the requirements of the IEEE 802.15.4 standard. Upon detection, the Rx records the message, stores it in the SRAM memory and notifies the microprocessor.

As can be seen on figure 3.2 the system currently relies on a synchronized clock between Tx and Rx which is transmitted in parallel to the data on a second wire. Obviously, the digital system in this form can not be used for a wireless data transmission. Only changes on the different modules introduced in chapter 4 will make this possible.

3.2 Software System

Once the hardware is deployed on the FPGA, compiled software code can be loaded into the Flash memory which will then be executed by the microprocessor on power-up of the system. Using the hardware address of the different modules as memory address, the software can interact with the underlying hardware presented in the previous section by simple read and write accesses to memory (so called virtual memory). The software is written in standard C code using multiple *ARM* libraries.

The software can be controlled externally by transmitting commands (simple sequence of characters) via the UART module on the FPGA. Upon reception, the software on the microprocessor reads the incoming character sequence and acts according to them. Normally, these commands are send from a computer connected to the UART module. A list of all possible commands is shown in table 3.1.

The structure of the code is simple: After an initialization phase the main routine enters an endless loop which is only interrupted by Interrupt

Requests (IRQs) such as a successful UART reception or an incoming transmission. Every IRQ has its corresponding Interrupt Service Routine (ISR) which reacts upon call. Currently, all executing code is directly integrated in the ISRs making their execution long and unstable. A solution to this problem is presented in section 4.2.2.

For the scope of this project the code allows to send simple sequences of 125 Bytes (= 125 characters) via transceiver between two FPGA boards as illustrated in figure 3.2. Following a typical sequence upon a message transmission between two FPGA boards after initialization is given:

Transmitting FPGA:

1. Computer sends message over serial connection to FPGA
2. UART module on FPGA receives messages and notifies microprocessor via IRQ
3. ISR on microprocessor shifts incoming message from UART module to SRAM memory
4. Upon completion, ISR notifies the Tx module and indicates DMA to transfer the message from memory to the Tx module
5. The Tx generates a bit stream out of the message and transmits

6. Transmission

Receiving FPGA:

7. The start detection unit of the Rx module on the FPGA detects SFD sequence and starts recording
8. The Rx module notifies the microprocessor via ISR about the incoming message
9. ISR indicates the DMA to transfer the message from the Rx module to memory
10. Upon completion ISR notifies the microprocessor and sends the received message via serial connection to the computer

3.2.1 Function Reference

As mentioned previously, the software running on the microprocessor can be controlled via simple commands send via UART connection. The following

table gives an overview about the different possible commands and their function. All commands must be completed with the end-of-line character '`\n`'.

Table 3.1: Microprocessor commands

Cmd	**Argument**	**Function**
`snd`	max. 125 `char`	configures transceiver as Tx and sends message
`rcv`	-	configures transceiver as Rx and listens for incoming messages, messages are printed upon reception and transceiver is put back to sleep
`end`	-	puts transceiver to idle mode
`ctr`	-	reads and prints transceiver control register
`err`	-	reads and prints transceiver status register

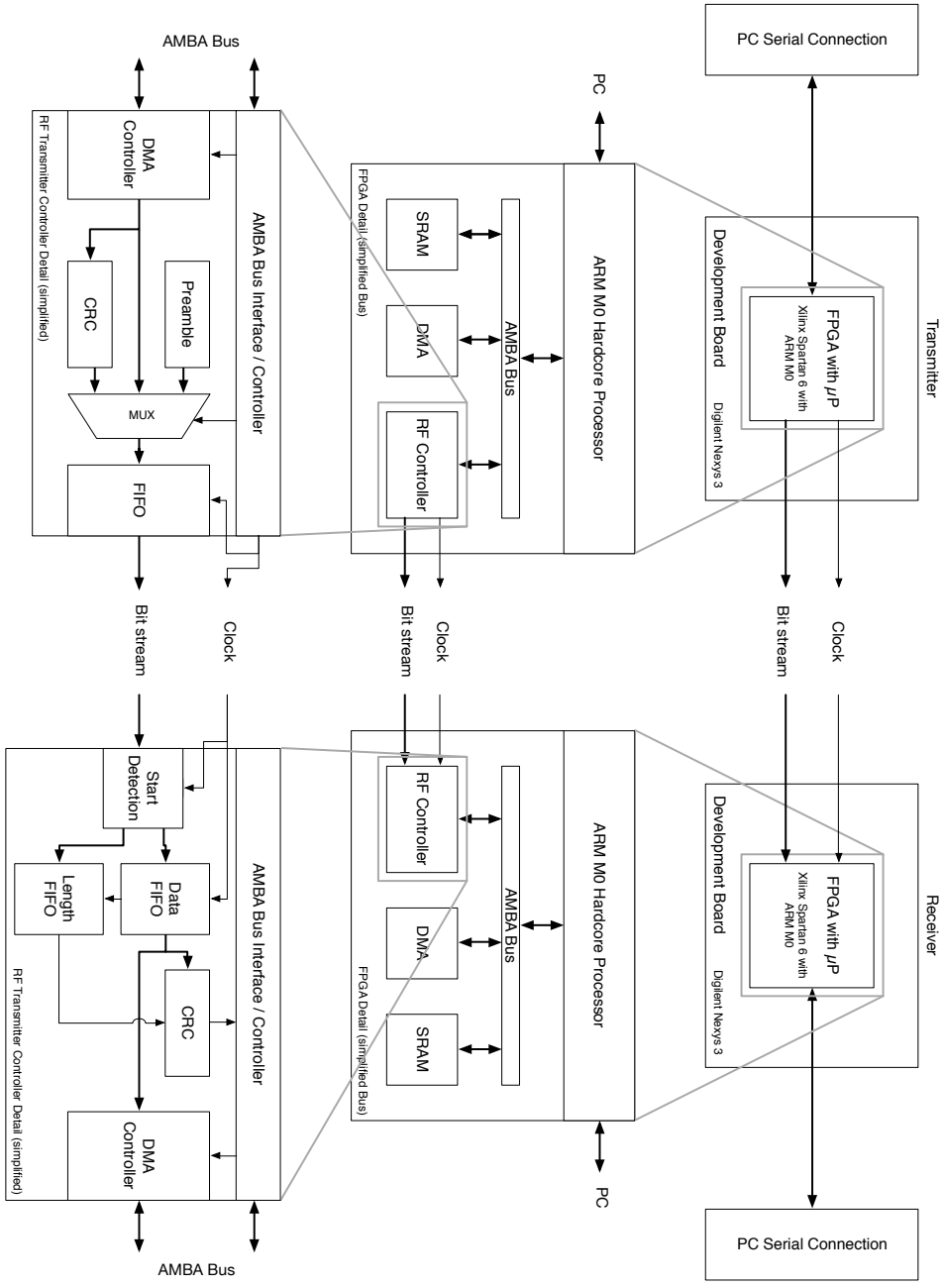


Figure 3.2: Detailed hardware structure of digital transceiver

4 | Digital System Extension

The previous chapter gave an overview about the digital system consisting of a microprocessor and its modules loaded onto a FPGA. A simple message transmission between two FPGAs connected over two wires was presented. As was pointed out, one of the two connections is needed to transmit a clock signal, otherwise the receiving FPGA is not able to sample the incoming bit stream. Obviously, this is not a feasible solution for a wireless communication. Thus, this chapter introduces changes on the existing modules in order to remove the necessity of a second, separate connection for the clock transmission. Furthermore, additional modules are integrated into the system in order to ease the interface to the external radio circuit which will be explained in chapter 5. Additionally, the software is updated to take the new hardware extensions into account. This chapter concludes the description of the digital part of the test system.

4.1 Hardware Extension

This section is split into two parts: First, an approach to remove the separate clock transmission from the existing digital system is presented, thus allowing a wireless communication. Second, additional modules for the digital system are introduced which ease interfacing the radio circuit presented in chapter 5.

4.1.1 Independent Clock Domain

For two digital systems exchanging data between each other it is crucial to operate in the same clock domain. If the receiving system samples incoming data too fast or to slow, the received signal contains some bits multiple

times or misses bits and is therefore not equal anymore to the transmitted signal. In a wireless system transmissions are erroneous. These errors get corrected by redundancy or by discarding and resending the packets on a higher level. Obviously, the clock signal needs to be available at all time and for every received bit, redundancy or resending are no option and it is therefore not possible to transmit the clock signal along with the data. Instead of transmitting the clock separately it can be extracted from the received bit stream. This is done with a dedicated module:

Clock Recovery (RxCRC) Having a similar frequency reference on the transmitting and the receiving system, the sampling frequency is approximately the same on both systems. The problem is to know the exact starting point or offset of sampling. If the receiving system coincidentally samples the bit stream at the bit transition, the extracted signal is very likely to be wrong. Therefore, a clock recovery module is added to the system generating a new sampling clock based on the incoming bit stream allowing a correct sampling.

The method presented here is based on the UART implementation which is described in [19], see section 2.2.2. It makes use of the fact that every bit stream contains bit transitions. These transitions may not happen every clock cycle but allow the clock recovery circuit to align its extracted clock to the bit stream regularly.

It is important to understand the difference between the system clock, driving the modules and the microprocessor, and the sample clock, running at a much lower speed, sampling the incoming bit stream. The clock recovery module generates the sample clock dividing the system clock with the help of a counter. The counter restarts automatically after every sample period with the sample clock signal being high during the first and low during the second half. Now in order to detect transitions in the incoming signal, the bit stream is first oversampled and the duration between a rising and a falling edge measured with a second counter. If the bit stream signal remains high for at least half a sample period the clock recovery module knows that the transition is intended and resets the clock counter independent of its current value. Therefore, the sample clock is always rising in the middle of a sample and the actual sampling of the bit stream does not take place during transitions anymore. Figure 4.1 gives an example of a clock extraction by the clock recovery module in form of a timing diagram. It is only for illustrative

purposes – the clock ratios are not identical with the actual system, but based on a test bench simulation verifying the correct working of the module.

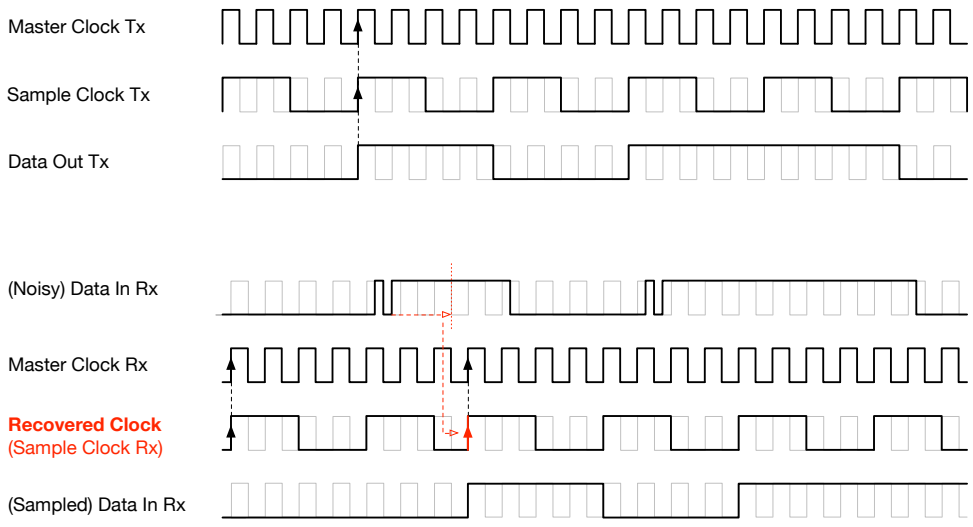


Figure 4.1: Clock recovery timing diagram

FIFO Controller (FifoCtrl) The clock recovery module explained in the previous paragraph allows the removal of the wire connections between the transmitting and the receiving system described in chapter 3. Still, another modification of the original system is necessary to assure a successful wireless transmission: Once the Rx in the receiving system detects a valid transmission, indicated by a SFD, it records the incoming bit stream as long as the sample clock is running. In the original setup this was no problem since the Tx could simply stop the sampling clock once the transmission was over, leading to a sampling stop in the Rx. Unfortunately, the information when a transmission ends can not be extracted by the clock recovery circuit. The problem is solved by adding a so called FIFO controller in front of the FIFO which counts the incoming bits. Since the length of the payload is indicated in the PHR at the beginning of every PPDU, the FIFO controller can compare its bit count and know once a transmission is over. As soon as all bits are passed to the FIFO it therefore stops the forwarding of the sampling clock. With this final modification the digital system is ready for wireless transmissions.

4.1.2 PCB Interface and Control

So far only the digital side of the test system has been presented. Next to it there is also an external radio circuitry which is explained in detail in chapter 5. As will be seen, the radio circuitry consist of multiple IC components which require correct calibration before use. Using different protocols such as SPI, the IC components can be calibrated by communicating over their pins. Dedicated hardware interfaces integrated in the FPGA in form of modules are used to control the IC components via software on the microprocessor. Together with the clock recovery modules explained in the previous subsection these modules are assembled on the FPGA into a single so called *PCB Interface* structure. This allows the microprocessor to interact directly with the external radio circuit, which can be used in the future to change the external component's configuration during an ongoing transmission.

The different hardware interface modules are briefly introduced in the following paragraphs, their Verilog module name is given in parenthesis. An overview on how they are connected is shown in figure 4.2.

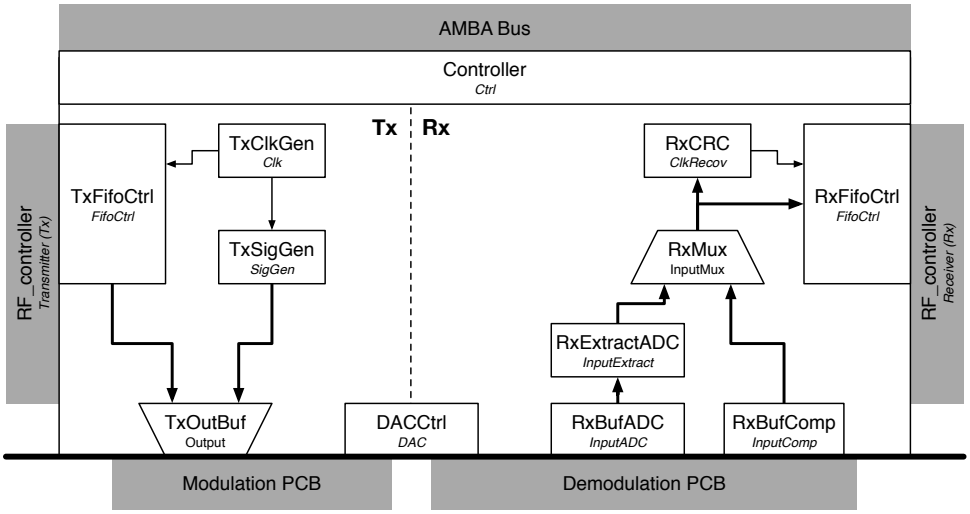


Figure 4.2: Overview on PCB Interface module; connections to the Controller are omitted for visibility, in gray the components the PCB Interface connects to, the (de)modulation PCBs will be explained in section 5.2

Controller (Ctrl) The PCB Interface offers one common *AMBA* bus interface for all its containing modules. Each of the bus interface's many registers is linked to one of the modules and the microprocessor can therefore configure them by a simple write operation to the corresponding register. Table 4.1 gives an overview over the different registers, their configuration and the affiliated modules.

Table 4.1: PCB Interface register mapping

Address	Name/Module	Bit	Function
0x00	Control	0	Enable Transmitter Output
		1	Transmitter Source:
			0: Bit stream
			1: Signal Generator
		2	Enable Receiver Input
		3	Receiver Source:
			0: Comparator
			1: ADC
		4	ADC Threshold Setting:
			0: Moving Average
			1: Fixed Threshold
0x04	DAC 0	11:0	Value
0x08	DAC 1	11:0	Value
0x0c	DAC 2	11:0	Value
0x10	DAC 3	11:0	Value
0x14	ADC Threshold	11:0	Value
0x18	ADC Raw	11:0	Most recent received ADC value
	(read only)	23:12	Current ADC threshold value
0x1c	Signal Generator	15:0	Value
0x20	Version Number	31:0	currently 0x5000
	(read only)		

Sample Clock Generator (TxClkGen) Generates the sample clock on the transmitter side by dividing the system clock.

Signal Generator (TxSigGen) The signal generator is added for test purposes simulating a constant bit stream. It reads the 16 bit value set in

the Controller register `0x1c` bitwise and sets its output pin to the currently read bit. A constant rotation through the 16 bit value leads to a continuous signal. The Tx can choose between transmitting data in from of PPDUs incoming via FIFO controller or the "random" signal generated by the signal generator.

ADC Interface (RxBufADC) The Rx has the possibility of sampling the incoming bit stream via a 1-bit comparator or a 12-bit ADC. More details about the exact implementation of the two components are given in section 5.2. The selection between the two input sources is done in the Controller register `0x00`. This module interfaces the external ADC and reads constantly its most recent sampled 12-bit value passing it on to the ADC Signal Extractor.

ADC Signal Extractor (RxExtractADC) The ADC Signal Extractor tries to extract the actual transmitted bit stream out of the 12-bit values received by the ADC Interface. It can do so via two functions: By comparing the value to a fixed threshold set in the Controller register `0x14` or by comparing it to a moving average. The moving average is calculated out of the most recent 12-bit values and offers therefore more flexibility to react on a high noise floor in the incoming bit stream. Again, the selection between the two methods is done in the Controller register `0x00`. For debug purposes the Controller register `0x18` offers insights into the current active threshold.

DAC Interface Multiple Digital-to-Analog Converter (DAC) units are used to control various components of the external radio circuit explained in section 5.2. Upon change of one of the Controller registers `0x04 - 0x10` the DAC Interface writes the new value to the corresponding external DAC via SPI.

All presented modules have been intensively tested with dedicated testbenches. The system was build with modularity in mind, allowing to quickly replace or remove one of the modules if needed. This is also true for the PCB Interface block itself placed between the digital transceiver module and the output of the FPGA, thus minimizing the impact on the existing system. This is illustrated in figure 4.3, giving an updated view of the bit stream

flow shown previously in figure 3.2. Changes are indicated in green. This concludes the hardware extension of the digital system.

4.2 Software Extension

As presented in section 3.2, software running on the microprocessor allows to control the different modules in the FPGA externally via UART. Since the previous section introduced new hardware modules, the software is updated too with new commands to control the additional modules.

4.2.1 Updated Function Reference

The newly added commands are listed in table 4.2. They extend the functions given in table 3.1 and allow to access the configuration register of the new hardware modules. All commands must be completed with the end-of-line character '`\n`'. Their successful execution has been tested on the actual test system.

Table 4.2: Additional microprocessor commands

Command	**Arg**	**Read/Write register** as shown in table 4.1
`pcb ctrl`	`-/int`	`0x00` (Controller)
`pcb dac0`	`-/int`	`0x04` (DAC0, MSK modulation frequency f_0)
`pcb dac1`	`-/int`	`0x08` (DAC1, MSK modulation frequency f_1)
`pcb dac2`	`-/int`	`0x0c` (DAC2, Rx comparator threshold)
`pcb dac3`	`-/int`	`0x10` (DAC3, RF carrier frequency f_c)
`pcb athh`	`-/int`	`0x14` (ADC threshold)
`pcb araw`	`-/int`	`0x18` (ADC debug data)
`pcb sigg`	`-/int`	`0x1c` (Signal generator value)
`pcb vers`	`-`	`0x20` (Module version number)
`hum`	`-`	prints outputs in a human readable form
`mat`	`-`	prints only reduced outputs

4.2.2 New Software Model

As presented in section 3.2 the code implementation follows a very simple structure of an endless loop in the main routine with ISRs acting on IRQs.

Unfortunately, the ISRs are written in a way containing all executing code, blocking the system upon call for a long period. A new code structure with clearly defined functions and a strict separation into header and source files has thus been implemented. Instead of reacting to the IRQ in the ISR, the ISR simply notifies the main program with the help of flags. Then, it is up to the main routine to call the corresponding function. Thus, the system remains responsive and available for new IRQs. Due to a severe problem discovered in the hardware structure, closer analyzed in section 4.3, a complete test of the new code has not yet been possible.

Based on this new structure a MATLAB class has been implemented as well. By serving the UART module directly, the MATLAB code simplifies testing and interacting with the test system. Instead of sending different commands manually via UART to the microprocessor, one can simply create a transmitter and a receiver object in MATLAB. Implemented methods allow then to directly interact with the objects calibrating the test system and performing multiple transmissions on it in an automated way. This powerful software tool combined with the sophisticated underlying hardware concludes the digital part of the test system.

4.3 Arbitration Issue

Unfortunately, an unexpected behavior of the microprocessor has been found upon testing the new software model introduced in the previous section. Short transmissions are not affected and therefore the demonstration of the test system's functioning is still possible. But in order to measure its performance a larger set of samples with longer messages is necessary and thus, a characterization of the system has so far not been possible.

This section gives a short overview about the encountered problems and tries to outline possible paths to a solution. Still, up to the day of this writing a solution has not been found and further investigations are necessary.

It was previously known that the transmission of long messages (>30 characters) causes erroneous receptions despite stable channels. Using the new software structures a new problem occurred: Once a IRQ is issued and the interrupt handler calls the corresponding ISR the code does not return to the main routine. The exact location where the program counter is lost can be found using GPIOs and shows that the code never returns from the ISR to the interrupt handler. If a new IRQ is made, the interrupt handler is called

again proving that the system is still running. This also explains why the old software system works: Since all code is executed in the ISR the missing jump back to the main routine was never discovered. Simulating the code does not reveal any problems indicating that the issue is in the hardware.

A closer look at the signal exchange on the *AMBA* bus reveals incorrect arbitration upon consecutive access: If a memory access by the DMA is immediately followed by a memory access of the microprocessor, the arbiter does not correctly forward the address signals. This leads to the assumption that, upon completion of the ISR, the program counter receives a corrupted return address and therefore jumps into unknown execution space and not back to the interrupt handler as expected. This is further confirmed by the observed erroneous reception for long messages. Since the problem occurs even if the ISR contains no code to execute, memory corruption can be excluded as cause.

Intense simulations of the arbiter confirm the mentioned issues. Implementing a new arbiter structure did not solve the issue despite its correct functioning in simulation. Further investigation revealed that the DMA unit neither behaves as expected and uses pipelining to transfer data. Pipelining is supported by the *AMBA* bus but not by the two arbiter implementations.

At this point the unusual bus structure should strongly be questioned. Knowing the memory amount required by the digital transceiver a simplification of the system with one bus and a second memory unit directly integrated in the transceiver might be an easier approach, skipping any need for arbiters. Since a demonstration of the test system is possible using the old software structure, further investigations are put on hold for now. Still, this error will hinder most program execution and an it is therefore important to understand the interlude between arbiter and DMA in order to solve the problem, or to pursue a new approach.

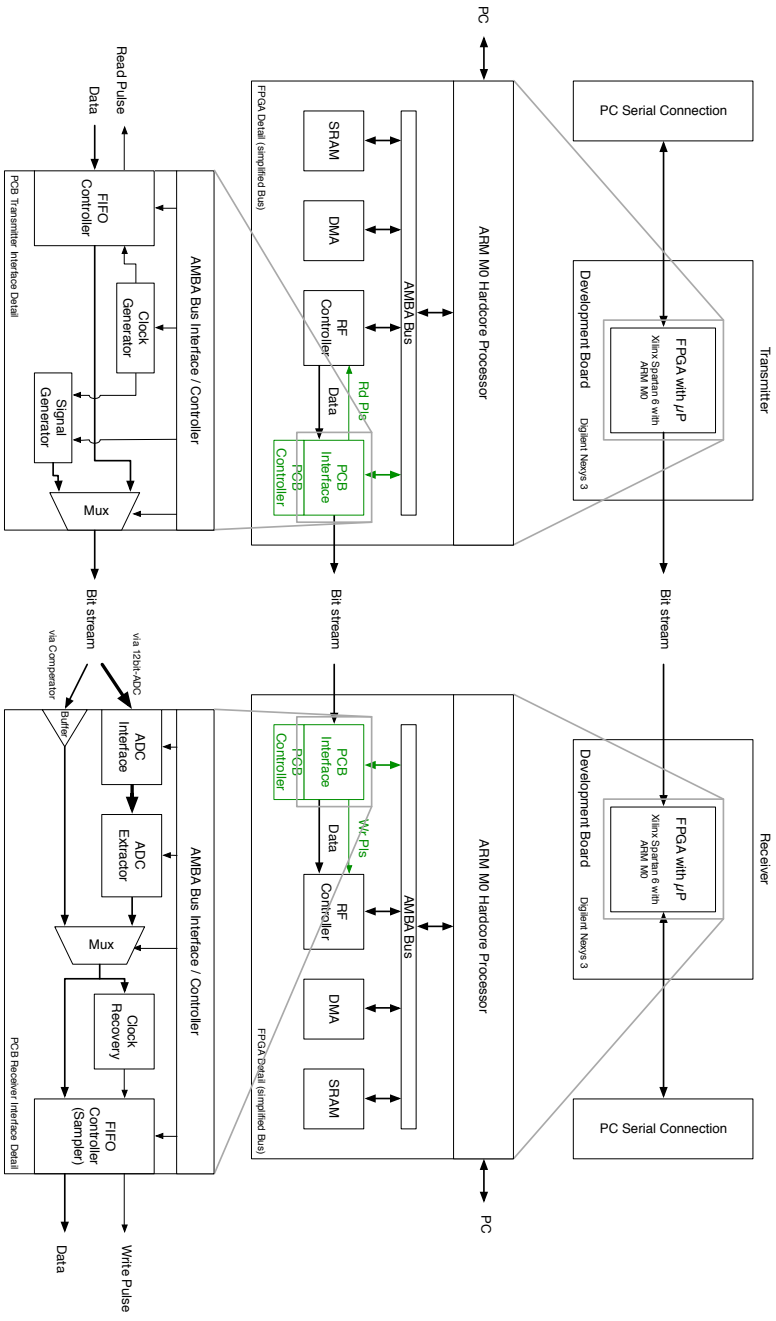


Figure 4.3: Extended detailed hardware structure of digital transceiver; modification to figure 3.2 shown in green

5 | Radio System

The previous chapters explained the structure and functioning of a digital transceiver. But in order to send data wireless, additional radio circuitry is necessary. This chapter presents the architecture and implementation of the radio circuit connected to the digital transceivers, and thus completes the test system. Using an artificially degenerated frequency reference, the radio circuit has similar constraints as a monolithic sensor node. Based on the test system, the feasibility of a stable, sustainable radio transmission between two monolithic sensor nodes is emulated and verified.

As soon as the first on-chip radio components are designed and manufactured, they will replace their macro-scale component in the test system, allowing to test their interaction with the digital transceiver. The test system fulfills therefore two purposes: Verifying a stable radio transmission with a high phase noise profile frequency reference and testing the different on-chip radio components once they are manufactured.

5.1 Overview

The radio circuit consist of two stages: A modulation stage, which translates the bit stream from the digital transmitter into an Intermediate Frequency (IF) radio signal, and a mixing stage, which brings the IF signal up to the actual carrier frequency and forms the so called Radio Frequency (RF) signal. In order to perform the mixing, a frequency reference or so called Local Oscillator (LO) is needed. As outlined previously, in the case of a monolithic sensor node the LO suffers from a high phase noise profile. Once up-mixed, the RF signal is transmitted wirelessly via an antenna before it is received by a second radio system. Naturally, the receiver circuit consists of the opposite stages, first mixing the received RF signal down to an IF and second,

demodulating the IF signal into a bit stream which is then passed to a digital receiver. The overall system is build to be compliant with the IEEE 802.15.4 standard and its specifications as formulated in section 2.3.1. Following the conception and functioning of the radio circuit is explained and analyzed, separated into two sections first presenting the IF stage and then the RF stage.

5.2 MSK Modulation and Demodulation

The modulation stage is responsible of transforming the bit stream of the digital transmitter into an IF radio signal which is then passed on to the up-mixing radio. Similar, on the receiver side the demodulation stage transforms the IF radio signal back to a bit stream which is then forwarded to the digital receiver of the receiving FPGA. Following the IEEE 802.15.4 standard and its specification, the bit stream has to be MSK modulated supporting a data rate of up to 250 kbps.

This section is split into three subsections: First, the architecture of the modulation and demodulation circuit is explained. Second, design flaws as discovered upon realization of the circuits are discussed and solutions are proposed. In the last subsection, results of the working modulation and demodulation circuits are presented in form of signal measurements.

5.2.1 Circuit Conception

Both, the modulation as well as the demodulation circuit are split into two parts: A PCB, transforming the digital signal into a voltage variation, and a Phase-Locked Loop (PLL) development board which transforms the voltage variation into a modulated IF radio signal, or, in case of the demodulator, does the inverse. This separation reduces the development time by relying on a proven PLL system and only designing the voltage variation translation circuit.

Modulation As highlighted in section 2.4.2, the MSK modulation uses two different frequencies to encode its two symbols, a binary ‘0’ and a binary ‘1’. Therefore, an oscillator capable of generating at least two different frequencies is needed. A Voltage Controlled Oscillator (VCO) is such an oscillator where its output frequency is proportional to its input voltage. Thus, using

two distinct voltage values as input of the VCO and switching between them allows to generate a MSK signal with two different frequencies. The two voltage values are set with two DACs, the switching is done via a simple multiplexer whose input is the bit stream signal. An overview of the modulation circuit is given in figure 5.1.
The DACs are configured via PCB Interface as explained in section 4.1.2, the PLL configuration and calibration is done via computer and a dedicated software provided by the PLL development board's manufacturer. During configuration the frequency spectrum of the VCO's output is measured in order to set the two DACs voltage values accordingly.
To avoid undesired effects on the voltage variation signal caused by the rapid switching of the multiplexer (i.e. ringing), two different types of Low Pass Filters (LPFs) are inserted between the switch and the VCO. The 1-pole passive LPF consists of RC components, the 3-pole active LPF contains two operational amplifiers and is designed using the filter design tool by *Analog Devices*. Using a jumper acting as a switch, the desired filter can be selected. Due to the limitations of the PLL gain bandwidth with no suitable alternative, the system was designed to operate at a data rate of 100 kbps instead of 250 kbps as required by the IEEE 802.15.4 standard. The two filters are build for the updated, lower data rate.

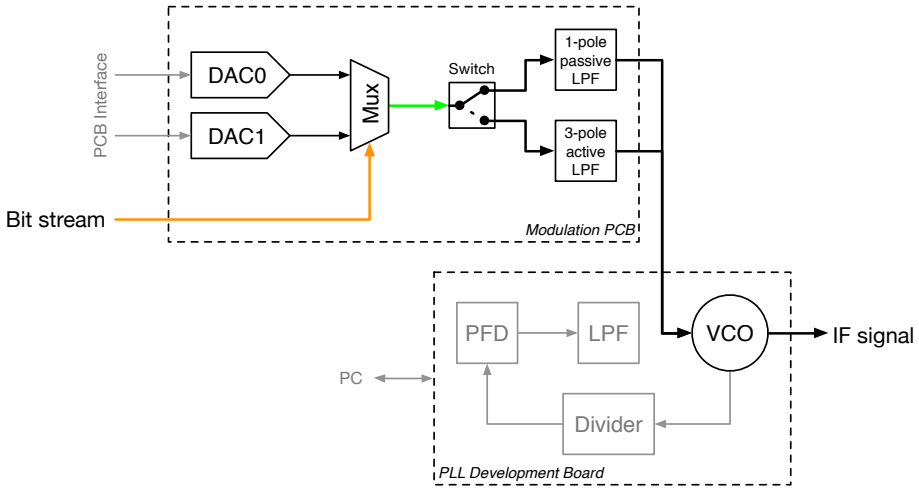


Figure 5.1: Modulation circuit schematic

Demodulation The demodulation stage is more complex using the entire PLL system: The goal of a PLL is to generate a stable frequency by "locking" it to a reference signal. This is done with a VCO whose output phase is compared to the reference signal using a Phase Frequency Detector (PFD). The PFD generates an error signal which describes the difference in phase of its two input signals in form of a voltage value. This error signal is then filtered and fed back into the VCO in order to adjust its output frequency and thus closing the feedback loop.
For the purpose of demodulation, the frequency reference at the PFD's input is replaced by the incoming IF radio signal. Assuming the IF signal contains only a single frequency, the error signal will settle on a constant value. As soon as the IF signal's frequency changes, the error signal follows and after some time stabilizes on a different voltage value. Therefore, the switching between two frequencies in the IF signal, as it is the case for MSK modulated signals, is translated into voltage variations on the error signal. Using an ADC or a simple comparator, these voltage variations can be sampled and translated back into a digital signal, thus demodulating the IF signal. An overview of the demodulation circuit is given in figure 5.2.
Again, the board's configuration is done via the PCB Interface module on the FPGA and the PLL development board is controlled via software on a computer. Like every feedback system, the PLL has a limited gain bandwidth which limits the data rate as highlighted previously.

5.2.2 Conception and Design Flaws

One primary flaw was detected upon manufacturing and testing of the modulation circuit: As can be seen in the circuit's simplified schematic in figure 5.1 the outputs of the two LPFs are connected together. Despite the fact that the unselected filter has no connection to the incoming signal, the selected filter's impedance is affected through the common output connection. This results in a much lower impedance, translating into a lower filter pole and thus, a drastically reduced bandwidth. The problem was solved by connecting the VCO directly to the output of the multiplexer and leaving the switch of the two filters in an intermediated, unconnected state. Despite the missing filtering no heavy ringing is found on the voltage variation signal as is shown in the measurements section in figure 5.2.

A second flaw was discovered in the demodulation system: As stated in the previous section, the error signal of a PLL is used to extract the

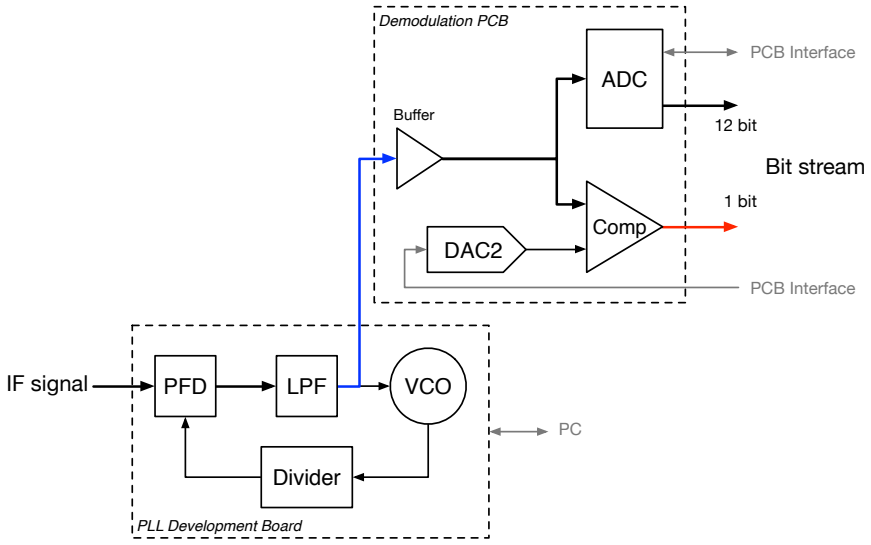


Figure 5.2: Demodulation circuit schematic

incoming data signal. Each of the two frequencies gets thereby translated into two different voltage values. While the difference between these two voltage values remains constant, the offset of the error signal is varying over time. Sampling the error signal with a fixed threshold, the output signal can saturate if the error signal's offset moves above this threshold. Trying to solve this problem in the digital domain, a much easier analog solution was found by adding a decoupling capacitor in series between the PLL and the demodulation PCB removing the offset permanently.

Unexperienced in designing PCBs, little attention was given to the power source of the PCB. Despite the fact that the circuits in their current form are working, connecting the board's ICs directly to a power supply instead of using intermediate Low-Dropout Regulators (LDOs) puts them at a constant risk of voltage spikes. This should be solved in a future design iteration.

5.2.3 Measurements and Results

The successful modulation and demodulation is presented in figure 5.3 showing the same data signal measured at different stages in the circuits. The signal colors correspond to the colored circuit nodes shown in figures 5.1 and 5.2. The measurement was made with a complete test system contain-

ing a wireless transmission using the radio circuit explained in section 5.3. It therefore proves the feasibility of a radio communication based on high phase noise profile frequency references as it will be the case for monolithic sensor nodes.

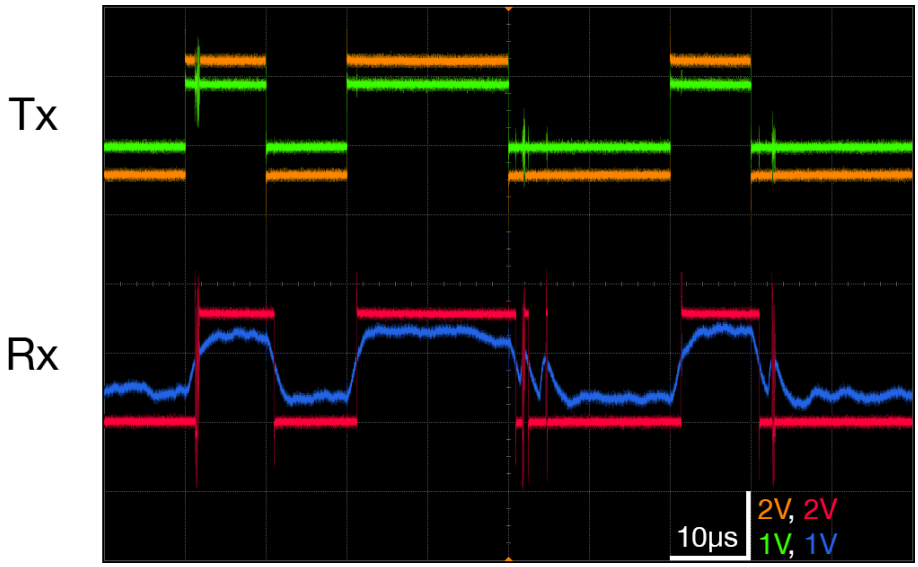


Figure 5.3: Measured (de)modulation signals; $R = 100$ kbps, colors correspond to the colored circuit nodes shown in figures 5.1 and 5.2

The transmission data rate can reach up to 100 kbps and the modulation leads to a radio signal with a center frequency of 50 MHz (IF) and a frequency offset of $\Delta f = \pm 200$ kHz between the two modulation frequencies. A spectrum showing the two distinctive frequency peaks is presented in section 5.3.3 in figure 5.11. For the given data rate the MSK modulation specifies a smaller offset of $\Delta f = \pm 25$ kHz and therefore the presented signal is rather BFSK than MSK modulated. Still, the circuits prove the feasibility of the concept. While the data rate is limited by the PLL architecture as stated previously, a change of the IF is possible. For the test system a IF of 50 MHz is chosen to ease compliance with the other components used in the radio system.

As can be seen on the measurements in figure 5.3, the extracted signal (in red) follows the transmitted signal (in orange) continuously containing little noise and no offset. Despite the missing filters at the output of the modulating switch no ringing can be seen on the voltage variation signal (in green).

The few glitches cause no problems and are likely to originate from coupling with the digital circuit and its much higher system clock. The demodulation PLL is able to track the frequency changes accurately with sharp transitions and little fluctuation on the signal (in blue). Upon sampling in the digital receiver the signal was extracted without any errors. Glitches are ignored by the clock recovery circuit as expected. Obviously, extended testing with a more complex transceiver, as mentioned in the previous chapter, will allow a more detailed analysis of the overall performance of the test system.

5.3 Emulation of Crystal-free Wireless Transmission

The IF is not bound by any specifications and only defined by the characteristics of the operating circuit. In contrast, the frequency band allocation of the radio spectrum is clearly specified and the wireless communication based on a certain standard requires the use of a fixed frequency band. As outlined in section 2.4.1, the IEEE 802.15.4 standard defines this RF band as 2400–2483.5 MHz. The IF of 50 MHz is given by the modulation and demodulation circuit explained in the previous section. This section presents now the circuit responsible of the wireless transmission and the translation of the IF signal into a RF signal and back.

The signal conversion from the IF to the RF is called up-mixing and is done by multiplying the IF signal with a defined frequency reference. Similarly, a multiplication with the same frequency reference is necessary to convert the incoming RF signal down to an IF signal, the so called down-mixing. Thus, communicating at the right RF strongly depends on the quality of the frequency reference. Section 2.2.2 gave an overview on occurring degradation effects using a low quality frequency reference. As further explained, it is not possible to integrate a high quality frequency reference into a monolithic sensor node and it has therefore to rely on a low quality frequency reference with a high phase noise profile. Since the goal of the test system is to emulate the behavior of such a node, the implemented frequency reference is artificially degenerated by injecting noise. Following subsection will explain the architecture of the radio circuit in general and the degenerated frequency reference circuit in detail. A later subsection analyzes design and conception flaws of the circuits and proposes solutions. Eventually, the impact of high

phase noise on a wireless transmission is measured and the obtained results are presented.

5.3.1 Circuit Conception

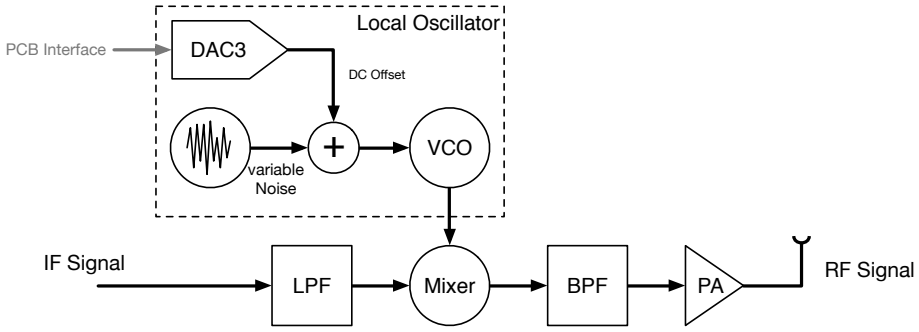
The actual radio architecture is based on a conventional design using different filters and amplifiers to mix, transmit and receive the signal. Two monopole antennas placed at a distance of approximately 30 cm form the wireless channel. An overview of the radio circuit is given in figure 5.4. In order to decrease development time, it is build out of off-the-shelf components made by *minicircuits* connected together with coaxial cables. These components provide a reliable and straight-forward way of creating and testing a radio.

The most important parameters of the test system's radio circuit are listed in table 5.1. The total gain combines the insertion loss and amplification of the individual components, the noise figure is calculated with Friis' Formula. The transmitter output power is limited by the maximum output of its power amplifier. The very high required receiver input power is given by the required minimum input power of the demodulation PLL and the limited receiver gain. Obviously, these requirements as well as the performance of the radio with its very low receiver noise figure are not comparable to the final on-chip radio design of the monolithic sensor node.

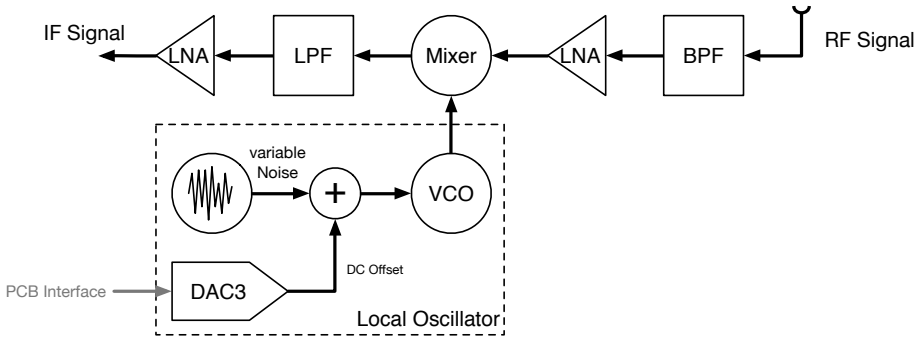
Table 5.1: Calculated radio parameters

	Transmitter	**Receiver**
Gain	19.7 dB	34.1 dB
Noise Figure		0.9 dB
Maximum Given Input Power	7.0 dBm	
Maximum Output Power	22.0 dBm	
Minimum Required Input Power		-40.1 dBm
Minimum Required Output Power		-6.0 dBm

As outlined in the introduction, the frequency reference of the radio circuit, the so called Local Oscillator (LO), is the most crucial component for a reliable radio communication and the only component of the test system's radio circuit which is custom-designed. The following paragraphs explain its design and the artificial degeneration of its quality in order to obtain a



(a) Transmitter



(b) Receiver

Figure 5.4: Radio circuit schematic; details on LO implementation given in figure 5.6

frequency reference with a high phase noise profile as used in a monolithic sensor node.

Local Oscillator circuit The LO consists of a free-running VCO generating a frequency of 2450 MHz which multiplied with the IF signal of 50 MHz leads to the desired center frequency of 2.4 GHz. The VCO and its output frequency are controlled via voltage level, set by a DAC, which in turn is connected to the PCB Interface. This mechanism allows the digital transceivers to directly choose the frequency band on which the radio is communicating. In order to have a low quality LO with a high phase noise profile, white noise is added to the control voltage generated by the DAC. Equation 2.2 in section 2.3.2 explains the relation between the voltage noise on a VCO's input

and the resulting phase noise at its output. The voltage noise is generated with the help of a Zener diode which has the particular property of conducting below a certain negative voltage, as shown in red in the voltage-current diagram in figure 5.5. Applying a negative voltage over the diode moves it into this so called reverse breakdown zone. Small, natural occurring voltage variations of the voltage source lead therefore to larger fluctuations of the diode's current, which can be considered as white current noise. Using an inverting amplifier with a high, variable gain this current noise is amplified and transformed into voltage noise. Using a second operational amplifier in a summing non-inverting configuration the amplified noise is added to the voltage offset of the DAC previously explained. Eventually, the combined voltage is fed into the VCO. A schematic overview of the noisy LO circuit is given in figure 5.6. It is implemented on a custom-designed PCB using IC components. A noise analysis of the circuit is presented in the following paragraph.

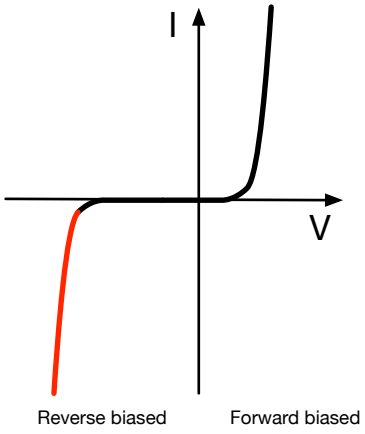


Figure 5.5: Current-voltage diagram of a Zener diode; operating region for noise generation shown in red

Noise Analysis of LO Circuit A monolithic sensor node based on a ring oscillator has a high phase noise profile of around $\mathcal{L} = -80$ dBc at $\Delta f = 1$ MHz as given by the simulation results presented in section 2.3.1. The following noise analysis of the LO circuit is made in order to verify that the test system's frequency reference has a similar (bad) performance. The

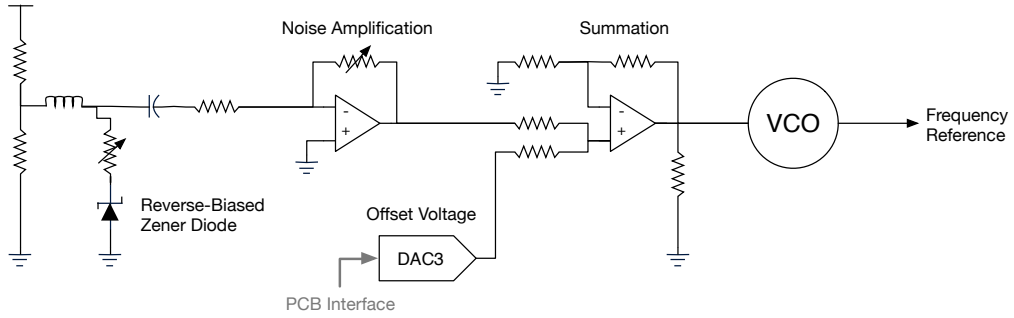


Figure 5.6: Noisy local oscillator circuit schematics

noise analysis is calculated stage-wise, the overall circuit is shown in figure 5.6. Superposition is used to transform the multiple noise sources of a stage into one common noise source at its output.

Zener Diode Noise A diode produces shot and thermal noise and forms therefore a current noise source $S_{I_N,\text{diode}}$ which is defined over the whole frequency spectrum as

$$\begin{aligned} S_{I_N,\text{diode}} &= \overline{i^2_{n,shot}} + \overline{i^2_{n,thermal}} \\ &= 2qI_D + 4qI_D \\ &= 6qI_D \end{aligned} \tag{5.1}$$

as given by equations (11.1) and (11.14) in [20] where q is the quantity of one electric charge defined as $1.602 \cdot 10^{-19}$ C and I_D is the bias current of the diode. I_D is therefore the first variable of the system.

First Stage An overview of the first stage is given in figure 5.7a. It is assumed that the Zener diode's noise is by orders of magnitude larger than the power supply's noise and therefore the only noise source of this stage. This is valid given the fact that the diode is acting as an amplifier for the voltage fluctuations coming form the power supply. Further, R_{P1} can be seen as negligibly small. Since the Zener diode forms a current noise source, the calculated Thévenin equivalent leads to a voltage noise source $S_{V_N,\text{diode}}$ with

a source impedance Z_s and a load impedance Z_L:

$$Z_s = \frac{R_1 R_2}{R_1 + R_2} + sL$$
$$Z_L = \frac{1 + sR_3C}{sC}$$

$$S_{V_N,\text{diode}} = {Z_s}^2 \, S_{I_N,\text{diode}} \tag{5.2}$$

Second Stage The second stage is responsible of the amplification of the diode noise, an overview is given in figure 5.7b. Its containing operational amplifier contributes to the overall noise, characterized with a voltage noise source $S_{V_{Vn},\text{amp, in}}$ in series and a current noise source $S_{V_{In},\text{amp, in}}$ in parallel to its inputs. The transfer function of the second stage is given as

$$H_{2^{\text{nd}}}(s) = \frac{-sR_{P2}C}{1 + sR_3C}$$

and the different output noise sources are

$$\begin{aligned}
S_{V_N,\text{diode, out}} &= {H_{2^{\text{nd}}}}^2(s) \; S_{V_N,\text{diode}} &&\approx \left(-\frac{R_2}{R_1}\right)^2 \; S_{V_N,\text{diode}} \\
S_{V_{Vn},\text{amp, out}} &= {H_{2^{\text{nd}}}}^2(s) \; S_{V_{Vn}\text{amp, in}} &&\approx \left(-\frac{R_2}{R_1}\right)^2 \; S_{V_{Vn},\text{amp, in}} \\
S_{V_{In},\text{amp, out}} &= {R_2}^2 \; S_{V_{In},\text{amp, in}}
\end{aligned}$$

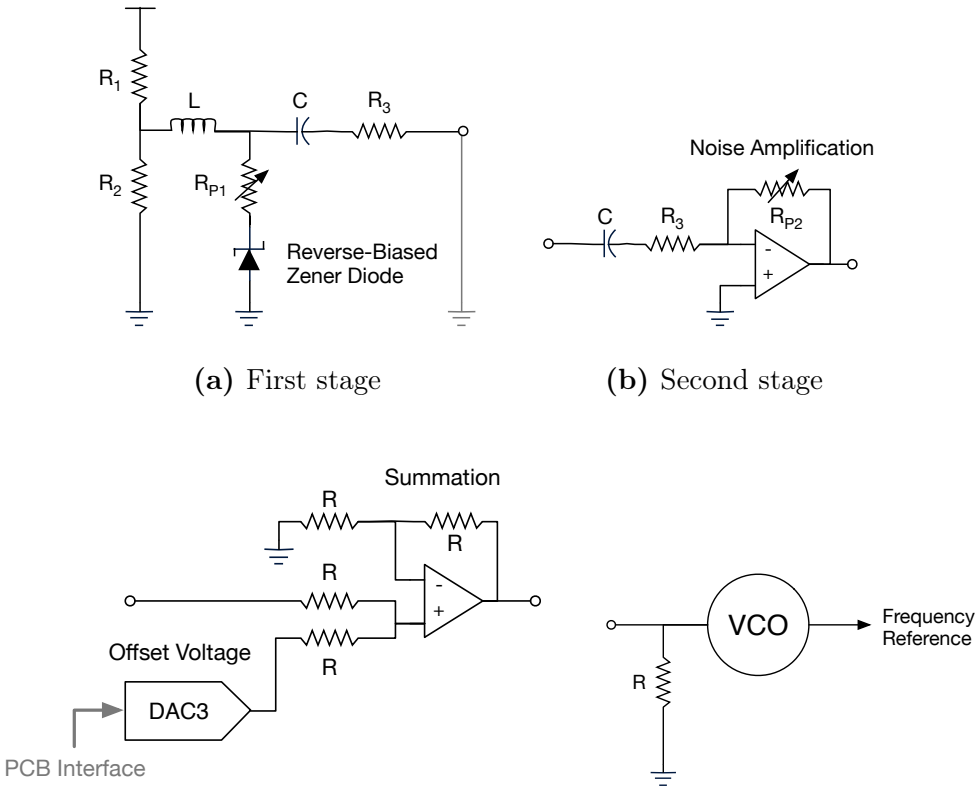
leading to the total noise voltage source at the output of the second stage as

$$\begin{aligned}
S_{V_N,\text{amp}} &= S_{V_{Vn},\text{amp, out}} + S_{V_{In},\text{amp, out}} + S_{V_N,\text{diode, out}} \\
&= G^2 \left({R_1}^2 S_{V_{Vn},\text{amp, out}} + S_{V_{In},\text{amp, out}} + S_{V_N,\text{diode, out}}\right)
\end{aligned} \tag{5.3}$$

where G represents the gain of the stage as $G = -R_2/R_1$. G is therefore the second variable of the system.

Third Stage The third stage sums the amplified noise and the voltage offset given by the DAC using an operational amplifier. An overview of the third stage is given in figure 5.7c. It is assumed

that the second stage noise is by orders of magnitude larger than the noise of the DAC and the operational amplifier. This is valid given that the second stage noise is amplified with a high gain and the purpose of the third stage is a simple addition with unity gain.



(a) First stage **(b)** Second stage

(c) Third stage **(d)** Forth stage

Figure 5.7: Noisy local oscillator circuit stages; complete circuit shown in figure 5.6

Fourth Stage and Application The fourth stage consists of the voltage noise applied to the VCO as shown in figure 5.7d. Choosing an operational amplifier for stage two with noise properties $\overline{i_n} = 4\ \text{pA}/\sqrt{\text{Hz}}$ and $\overline{v_n} = 2.75\ \text{nV}/\sqrt{\text{Hz}}$ and designing the system such as the bias current of the diode is $I_D = 50\ \text{mA}$, the overall noise is given as $S_{V_N,\text{amp}} = G^2 \cdot (214.0\ \text{nV}/\sqrt{\text{Hz}})^2$ and therefore

clearly dominated by the Zener diode's noise. The sensitivity of the selected VCO is $\kappa = 47.5$ MHz/V which based on equation 2.2 leads to the following function

$$\mathcal{L}(f) \;=\; \frac{\kappa_V^2}{4f^2} S_{V_N,\text{amp}} \;\approx\; \frac{3qZ_s{}^2\kappa_V^2}{2f^2}\, G^2 I_D \tag{5.4}$$

with the two variable G and I_D as stated previously. Using a potentiometer for R_{P2}, the gain and thus the intensity of the generated noise can be changed.
Figure 5.8 shows the comparison between the phase noise estimation of the LO circuit for different gains as given by equation 5.4 and the simulated phase noise profile of a ring oscillator presented previously in figure 2.7. As can be seen using a gain of 30 or larger, the estimated phase noise is equal or larger than the simulation results. Furthermore, the generated noise is inverse proportional to the square frequency offset and thus emulates random walk noise. As outlined in section 2.2.2, an oscillator's performance is highly related to this phase noise component. Therefore, the proposed LO structure promises to be a suitable emulation of an on-chip frequency reference as used by a monolithic sensor node. This concludes the noise analysis of the LO circuit.

5.3.2 Conception and Design Flaws

The construction of the radio circuit is straight forward based on the reliable and simple to integrate components from *minicircuits*. Initially, using a signal generator as frequency reference, smaller issues such as a LPF with a too high cutoff frequency have been found and solved.

Replacing the signal generator with the noisy LO did not change the phase noise profile of the radio signal indicating a problem in the noise generation. Measuring the phase noise directly at the LO's output revealed that, even with a maximum gain, the generated noise is much smaller than estimated. A first goal was therefore to measure the noise contribution of the LO circuit's individual components. Surprisingly, the DAC was responsible for the majority of the noise. Lack of basic knowledge in the PCB design led to flaws such as badly placed or missing decoupling capacitors. Since the DAC is controlled via FPGA, noise form the digital circuit propagates to the DACs analog output pin, leading to a higher noise floor. This problem

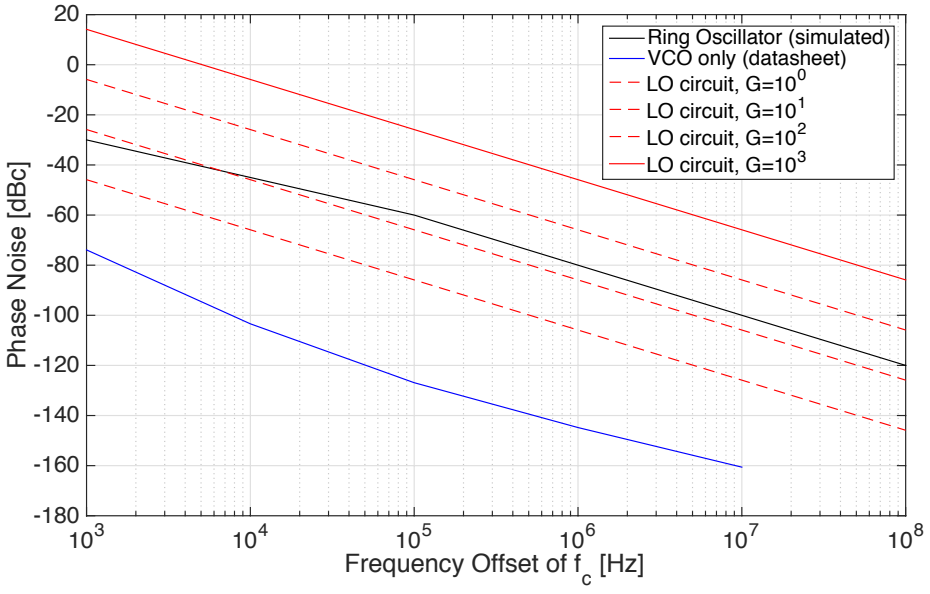


Figure 5.8: Estimated local oscillator phase noise profile

was partially fixed by adding additional capacitors to the circuit. Still, the overall noise performance remained low and the Zener diode's contribution was minimal. A possible explanation for this behavior is given here:
Verifying the value of the passive components revealed that the AC-decoupling inductor between the power supply and the diode (figure 5.7a) had a too low inductance of $L = 1\ \mu$H. Increasing it by a factor 1000, resulted in an increase of the phase noise of up to 10 dB. Unfortunately, modifying the gain via R_{P2} still had no effect. This leads to the conclusion that either the Zener diode does not generate sufficient noise, probably due to a bad biasing, or that too much of its generated noise leaks via inductor into the power supply. Furthermore, having a very high gain, the gain bandwidth product of the amplifier is drastically reduced, possibly degenerating the noise amplification and thus the generated noise even further.

Eventually, a solution for the problem was found in form of a signal generator which is capable of producing white voltage noise. Disconnecting stage one and two from the LO circuit and replacing it with the signal generator allowed to directly inject noise into the voltage offset provided by the DAC. Varying the peak-to-peak amplitude changes the generated white noise intensity. Different amplitude settings are shown in figure 5.9 and compared with

the original state of the LO circuit which did not generate enough noise. As can be seen, a peak-to-peak amplitude of around 50mV leads to a comparable phase noise profile as the targeted simulated ring oscillator profile. All measurements have been performed with a center frequency of $f_c = 2.45$ GHz.

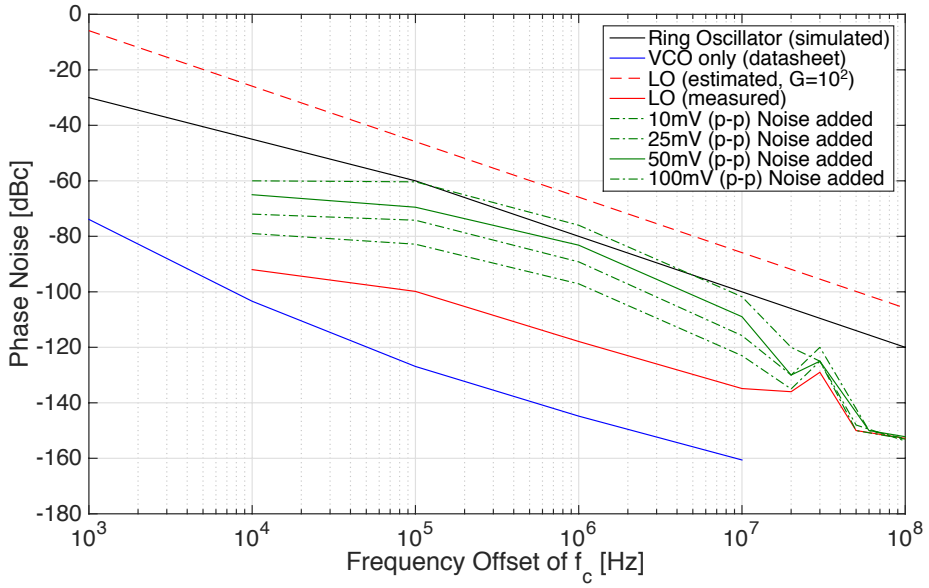


Figure 5.9: Measured local oscillator phase noise profile

For a future LO circuit design, a better integration without an external signal generator could be achieved by generating noise with a resistor instead of a Zener diode. A resistor has the advantage that it does not need any specific voltage biasing and is therefore more flexible. Every resistor generates thermal noise and thus forms a voltage noise source defined by equation (11.1) in [20] as

$$\overline{v_n{}^2} = 4k_BTR$$

where $k_B = 1.381 \cdot 10^{-23}\text{J/K}$ is the Boltzmann constant, $T = 300$ K the ambient temperature and R its resistance. Needing a peak-to-peak voltage noise of 50 mV at the VCO's input as given by the results in figure 5.9, an equivalent voltage noise source of ${S_{V_N}}^{1/2} = (50\text{ mV}/2\sqrt{2}) = 312.5\mu\text{V}/\sqrt{\text{Hz}}$ is required. Setting the second stage gain to $G = 1000$, the resistor should thus have a minimum resistance of

$$R = \sqrt{\frac{1}{4k_BT}\frac{S_{V_N}}{G}} \approx 4.35\ \text{M}\Omega$$

5.3.3 Measurements and Results

Using the test system, a successful wireless transmission based on high phase noise profile frequency references has already been demonstrated in section 5.2.3 with the resulting data signals shown in figure 5.3. Instead of the signal content, this section focuses on the signal's properties and its undergone deteriorations by analyzing and measuring the test system's radio behavior. A photo of the complete test system hardware is shown in figure 5.10.

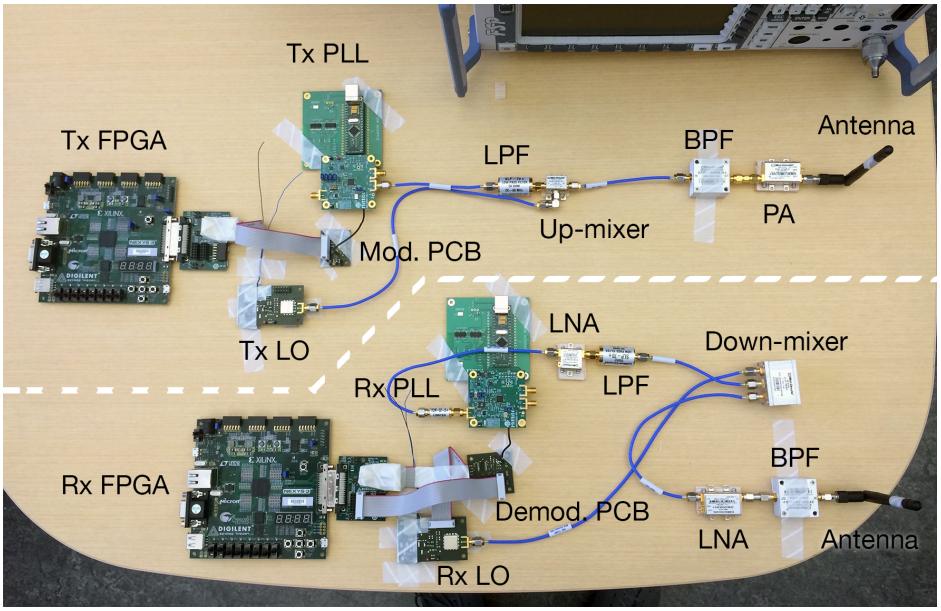
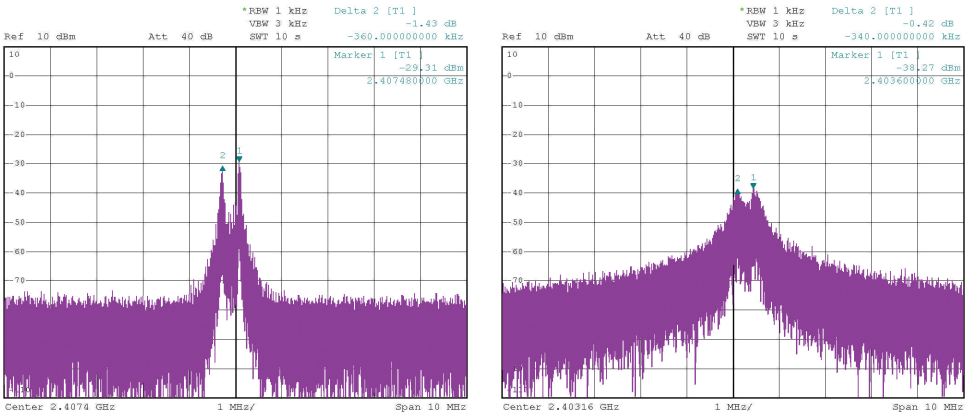


Figure 5.10: Photo of test system hardware; separation between Tx and Rx is indicated with white dashed line, distance between antenna is approximately 30 cm, for visibility no power wires and external connections are shown

Frequency Deterioration by Phase Noise As explained in section 2.2.2, phase noise affects the frequency spectrum of a RF signal. By connecting the test system to two different frequency references, the difference and therefore the impact of high phase noise can be measured. A signal generator is used as frequency reference with a low phase noise profile (figure 5.11a) and the noisy LO circuit presented in the previous subsections forms the high phase noise profile frequency reference (figure 5.11b). Comparing the transmission

frequency spectra of the two frequency references in figure 5.11 reveals an increase of bandwidth, an attenuation of the signal, and a higher noise floor for a transmission based on the noisy LO.



(a) Low phase noise **(b)** High phase noise

Figure 5.11: Receiver frequency spectrum; $f_c = 2.4$ GHz, $\Delta f = 350$ kHz

As outlined previously, an exact quantification of the degeneration effects is not yet possible. Main limitations are the missing preamble detection via correlation (section 3.1.1) and the reduced packet length caused by an arbitration issue of the microprocessor (section 4.3). Due to these limitations it is not possible to measure a meaningful bit error rate and thus, quantify the degeneration effects. Still, the test systems allows the conclusion that a transmission based on high phase noise profile frequency references is possible.

Amplitude Deterioration by Phase Noise As shown in the previous paragraph, high phase noise has an impact on the signal's frequency spectrum. Less is known about its effect on the signal power. So far, the only available results come from simulations and indicate no significant changes of the signal's amplitude. Measuring the Signal-to-Noise Ratio (SNR) of the test system with a low and high phase noise profile frequency reference allows to compare the two and to infer possible deteriorations:

The thermal noise at the input of the receiver is given as

$$P_{\text{noise, in}} = 10 \log_{10} (k_B T B)$$

where k_B is the Boltzmann constant, T the ambient temperature and B the system's bandwidth, chosen here as $B = 1.5 \cdot \Delta f = 600$ kHz. Knowing the receiver's properties the theoretical SNR is calculated as

$$SNR_{\text{calc}} = P^{\text{min}}_{\text{signal, in}} - P_{\text{noise, in}} - NF \tag{5.5}$$

where $P^{\text{min}}_{\text{signal, in}}$ is the calculate required minimum input power of the receiver and NF the noise figure of the receiver, both given in table 5.1. This leads to a very high SNR of 75.0 dB caused by the high minimum input power requirement of -40.1 dBm of the demodulation PLL, the excellent noise properties of the *minicircuits*' components, and the general low bandwidth. Obviously, the high input power requirement leading to a very low receiver sensitivity is not comparable to a real system. The obtained high SNR value reduces the significance of the here presented results.

In order to measure the SNR of the test system, first the output noise of the receiver is calculated as the propagation of the thermal input noise, given by equation 5.3.3, to the receiver's output:

$$P_{\text{noise, out}} = G + NF + P_{\text{noise, in}}$$

where G is the receiver gain. Then, by measuring the output power of the signal, the receiver's SNR is obtained as

$$SNR_{\text{mes}} = P_{\text{signal, out}} - P_{\text{noise, out}} \tag{5.6}$$

where $P_{\text{signal, out}}$ is the measured output power of the receiver with an applied input power $P_{\text{signal, in}}$. Using a minimum input power signal source, the difference between a receiver based on a low and a high phase noise profile frequency reference can be measured and compared. The results are listed in table 5.2 and show a power decrease of 0.9 dB for a receiver operating under high phase noise.

The measurements are consistent with the results of the simulations indicating no significant change of the SNR based on a high phase noise frequency reference. Furthermore, the measured values are in line with the theoretical, calculated value. Having a very high SNR, it is very likely that the results do not contain any interfering thermal noise and thus, are only influenced by the high phase noise. At the same time, with such a high ratio it is uncertain how much noise actually contributes to the measurements. The limited receiver input power range does not allow to measure a tendency, making it

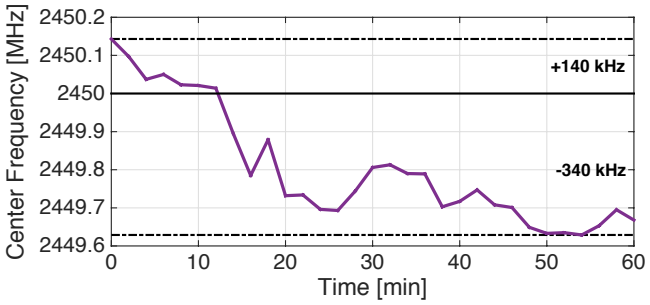
Table 5.2: Measured radio parameters

Power		Calculation	Measurements	Unit
Input Noise	$P_{\text{noise, in}}$	-116.1		[dBm]
Output Noise	$P_{\text{noise, out}}$	-81.0		
Input Signal	$P_{\text{signal, in}}$	(min.) -40.1	-40.1	
Output Signal	$P_{\text{signal, out}}$		-8.0 clean	
			-8.9 noisy	
SNR		75.0	73.1 clean	[dB]
			72.2 noisy	
Difference			0.9	

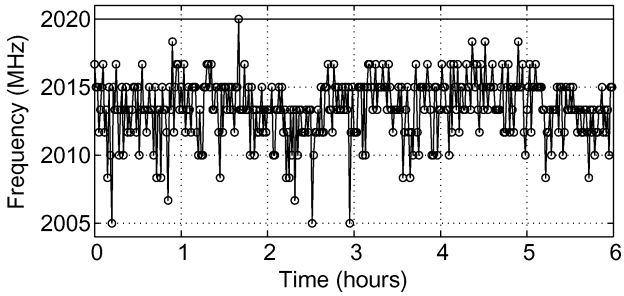
hard to draw a conclusion for the envisioned receiver design with a sensitivity of -85 dBm. Still, the test system shows that a power degradation of the signal due to high phase noise is unlikely.

Frequency Stability A measurement of the frequency stability of the noisy LO is shown in figure 5.12a. It is important to highlight that this measurement is optimistic, based on the strongest frequency component in the signal's frequency spectrum per time unit, and not as it is commonly done by measuring the signal in the time domain and extracting its actual frequency. Still, a clear frequency drift of almost 500 kHz during one hour of measurement is visible, likely to be caused by the free-running property of the LO's VCO. Despite this drift, the receiver PLL was able to continue to track the center frequency of the incoming IF signal and a successful transmission was possible during multiple hours of operation.
Comparing the measured drift of the test system to an implemented ring oscillator as shown in figure 5.12b given by [21] reveals that a real on-chip frequency reference is much less stable with more than ten times the frequency deviations compared to the measurements of the noisy LO. In chapter 6, an approach to further increase the drift of the LO is presented.

Power Consumption As outlined previously, the power values of the radio circuit are not comparable to the final design of the on-chip radio. For completeness, the consumption of the test system's radio is given in table 5.3 with a total consumption of 4.24 W during operation.



(a) Noisy LO circuit



(b) CMOS ring oscillator; as given by [21]

Figure 5.12: Noisy local oscillator frequency stability

Table 5.3: Radio power consumption

		Power [W]
Transmitter	Modulation	1.10
	Radio	1.31
Receiver	Demodulation	0.91
	Radio	0.92
Total		4.24

6 | Conclusion and Future Work

After one semester here I have learned that California is not only about fast-evolving technologies, it is also about open-minded people, beautiful nature – and more recently, a huge water crises. What tremendous effect would a more targeted irrigation have? By how many liters, or gallons, could the water consumption be reduced if a saturated soil is not watered again and again?

Wireless Sensor Networks are a simple and very cost-friendly solution to this problem: Consisting of many small, autonomous sensor nodes connected through a wireless network, multiple parameters such as humidity can be measured and a large area can be mapped through collaborative efforts. With the ongoing development for a monolithic sensor node, deployment and maintenance cost will be even further reduced and such sensor nodes can be spread as numerous and as non-invasive as dust – and might lead to the much needed water savings in the future.

The design of a monolithic sensor node and especially its radio unit is challenging. Being powered only through energy harvesting such as on-chip solar cells, the radio unit, responsible for the majority of the power consumption, is largely constraint. Furthermore, by renouncing any off-chip component such as quartz crystals the radio's performance is further degraded. Relying on a low quality on-chip frequency reference the radio is exposed to high phase noise, leading to a larger bit error rate which forces retransmissions and thus increases the power consumption. So far, it has been unknown if a sustainable radio transmission based on a high phase noise profile frequency reference is realistic.

Presented Work This work presents a complete macro-scale model of two communicating monolithic sensor nodes emulating radio transmissions under high phase noise. Through noise injection the quality of the model's

frequency references is artificially degenerated and the effect on the radio transmission continuously measured. Thus, this test system allows a deeper understanding of the implication of high phase noise and verifies the feasibility of a sustainable communication between two monolithic sensor nodes. In order to complete the test system, achievements on multiple levels have been made:

The presented digital transceiver forms the interface between a microprocessor and radio circuitry by translating simple messages into transmittable packages. Using a clock recovery mechanism and different sampling architectures the digital transceiver is able to reconstruct the message in the receiving system without the need for any wired connection. The main focus of the digital implementation was set on easy extendability and persistent modularity allowing to add new and extend existing transceiver features in a LEGO-like manner. This simplifies the testing and control of on-chip radio circuits once their design is completed, and eases the development of a complete digital transceiver fully compliant with the IEEE 802.15.4 standard. Exhaustive simulations and rigorous testing of all digital systems guarantee a smooth synthesis and thus a straight-forward implementation into an actual chip without the need for adaptations. Powerful software tools interfacing the digital system allow a simple access and an automated way of testing the radio circuitry. With the preparation of a more robust software model and a problem analysis of the current hardware architecture, future improvements of the system are outlined which will ease interconnection with higher software levels such as *OpenWSN* and will increase the overall robustness.

Next, a complete radio setup connected to the digital transceiver is presented. It is able to communicate wirelessly at 2.4 GHz based on high phase noise profile frequency references on the transmitter and receiver side. The carrier frequency as well as the amount of phase noise are variable, leading to a flexible setup for diverse measurements. The radio setup follows the IEEE 802.15.4 standard and thus, allows a profound and adaptable emulation of wireless transmissions between two monolithic sensor nodes. Due to macro-scale of the setup, parameters such as the radio power consumption or the receiver sensitivity are not comparable to a final on-chip radio design.

Based on the test system, a successful, sustainable wireless communication between two nodes is demonstrated, being able to remotely control small MEMS structures. The transmitting and receiving nodes thereby relay on independent frequency references with a phase noise of -80 dBc at a frequency offset of 1 MHz each. The receiver's demodulation PLL was able to

track a 500 kHz frequency drift of the radio signal and could keep up the transmission for multiple hours. No significant signal power degenerations as side-effect of high phase noise exposure were observed. This experiment leads to the conclusion that a wireless communication is possible despite the use of low quality frequency references and therefore, the monolithic design of a sensor node is feasible.

Future Work The presented test setup lays the foundation for future development, allowing to improve measurements and to test additional aspects of the communication between monolithic sensor nodes. Following a few ideas for future improvements are outlined:

Currently, the digital transceiver contains only a reduced implementation of the IEEE 802.15.4 standard which allows no sophisticated evaluation of the radio's performance. Detecting incoming packets with the help of correlation instead of an exact matching as explained in section 2.4.3 would allow to quantify the radio's performance. Furthermore, encoding the bit stream with chip sequences would make transmissions more robust.
As pointed out in section 4.3, the test system is currently limited to the transmissions of short messages. Multiple indications have been presented on how this problem can be solved. The unusual bus structure and its arbiter are very likely to be responsible. For a deeper testing, solving this issue is important.

Next to improvements of the current system, following new features could be implemented: Instead of sampling the received bit stream with a fixed threshold, a dynamic threshold depending on the average power of the digital signal would avoid the problem of saturation. Implementing the translation of bits from and into chip sequences leads to a pseudo-random signal with an almost constant average power. Deviations from this constant value, indicating a saturation, can easily be measured by a dedicated digital module, which then adapts the threshold of the correlator based on the measurements.
As pointed out in section 5.3.3, the carrier frequency drift of the test system is low and thus not comparable to an on-chip ring oscillator. But since the carrier frequency can be controlled via the digital transceiver, a dedicated module could artificially increase the drift by constantly changing the frequency and thus, simulate a more accurate drifting effect.

Personal Comment As shown throughout this work, emulating and designing a monolithic sensor node requires knowledge in many areas of the electronic field. Acting on multiple stages, from high-level software to low-level circuit design, allowed me to gain new insights. Particularly, I discovered the importance of prototyping: Despite the necessity of understanding the underlying theory, prototypes give immediate feedback about the circuit's behavior and reveal misconceptions, especially in the complex field of RF electronics. Going through multiple prototype iterations quicker eventually leads to a better setup. Learning to design PCBs revealed how crucial it is to not only understand circuit theory but also circuit design. A simple decoupling capacitor can remove an offset as efficient as a complex digital module. Relying more on manufacture recommendations such as the use of LDOs will definitely be the goal for future designs.

Doing this master thesis in the Ubiquitous Swarm Lab at UC Berkeley was a great experience. And who knows, up on my next visit in California, the water crises might have already vanished thanks to the rise of monolithic sensor nodes. Although, a behavioral change is likely to be necessary in order to reduce the water consumption, engineering can ease the process, bringing the future a little closer. I am very proud to be part of it.

This Master thesis was written by Lorenz Schmid at the Ubiquitous Swarm Lab of the University of California, Berkeley between February 16, 2015 and August 14, 2015 under the supervision of Professor Kristofer S.J. Pister (UCB) and Professor Yusuf Leblebici (EPFL).

Signature

Place, Date

Bibliography

[1] K. S. Pister, J. Kahn, and B. E. Boser. (1997) Smart dust: BAA97-43 proposal abstract. [Online]. Available: http://www.eecs.berkeley.edu/~pister/SmartDust/SmartDustBAA97-43-Abstract.pdf

[2] B. Warneke, M. D. Scott, B. S. Leibowitz, L. Zhou, C. L. Bellew, J. A. Chediak, J. M. Kahn, B. E. Boser, K. S. Pister, and others, "An autonomous 16 mm3 solar-powered node for distributed wireless sensor networks," in *Sensors, 2002. Proceedings of IEEE*, vol. 2. IEEE, 2002, pp. 1510–1515.

[3] T. H. Lee and A. Hajimiri, "Oscillator phase noise: A tutorial," vol. 35, no. 3, pp. 326–336, 2000.

[4] Linear Technology. LTP5902-WHM SmartMesh WirelessHART node, wireless mote module - datasheet. [Online]. Available: http://cds.linear.com/docs/en/datasheet/59012whmf.pdf

[5] A. Molnar, B. Lu, S. Lanzisera, B. W. Cook, and K. S. Pister, "An ultra-low power 900 MHz RF transceiver for wireless sensor networks," in *Custom Integrated Circuits Conference, 2004. Proceedings of the IEEE 2004*. IEEE, 2004, pp. 401–404.

[6] B. W. Cook, A. Berny, A. Molnar, S. Lanzisera, and K. S. Pister, "Low-power 2.4-GHz transceiver with passive RX front-end and 400-mV supply," vol. 41, no. 12, pp. 2757–2766, 2006.

[7] M. Tabesh, M. Rangwala, A. Niknejad, and A. Arbabian, "A power-harvesting pad-less mm-sized 24/60ghz passive radio with on-chip antennas," in *2014 Symposium on VLSI Circuits Digest of Technical Papers*, 2014, pp. 1–2.

[8] K. T. Braam, S. K. Volkman, and V. Subramanian, "Characterization and optimization of a printed, primary silver–zinc battery," vol. 199, pp. 367–372, 2012.

[9] M. D. Scott, B. Boser, and K. S. Pister, "An ultralow-energy ADC for smart dust," vol. 38, no. 7, pp. 1123–1129, 2003.

[10] B. H. Calhoun, J. Lach, J. Stankovic, D. D. Wentzloff, K. Whitehouse, A. T. Barth, J. K. Brown, Q. Li, S. Oh, N. E. Roberts, and Y. Zhang, "Body sensor networks: A holistic approach from silicon to users," vol. 100, no. 1, pp. 91–106, 2012.

[11] B. W. Cook, "Low energy RF transceiver design (doctoral thesis)," 2007.

[12] M. Lallart, *Small-Scale Energy Harvesting*. InTech, 2012.

[13] E. Rubiola, *Phase Noise and Frequency Stability in Oscillators*. Cambridge University Press, 2008.

[14] A. Abidi, "Phase noise and jitter in CMOS ring oscillators," vol. 41, no. 8, pp. 1803–1816, 2006.

[15] "IEEE standard for local and metropolitan area networks, part 15.4: Low-rate wireless personal area networks (LR-WPANs)," 2011. [Online]. Available: http://standards.ieee.org/getieee802/download/802.15.4-2011.pdf

[16] J. Notor, A. Caviglia, and G. Levy, "CMOS RFIC architectures for IEEE 802.15.4 networks," 2003.

[17] J. G. Proakis and M. Salehi, *Digital Communications*, 5th ed. McGraw-Hill, 2008.

[18] F. Bigazzi, "Digital controller for microrobots (master thesis)," 2014.

[19] J. Axelson, *Serial Port Complete: COM Ports, USB Virtual COM Ports, and Ports for Embedded Systems*, 2nd ed. Lakeview Research, 2007.

[20] T. H. Lee, *The Design of CMOS Radio-Frequency Integrated Circuits*, 2nd ed. Cambridge University Press, 2003.

[21] N. M. Pletcher, S. Gambini, and J. Rabaey, "A 52 uW wake-up receiver with 72 dBm sensitivity using an uncertain-IF architecture," vol. 44, no. 1, pp. 269–280, 2009.